GX IEC Developer 7.01

IEC Programming and
Documentation System

Reference Manual

Art. no.: 43597
04 12 2006
Version K

MITSUBISHI ELECTRIC INDUSTRIAL AUTOMATION

About this Manual

The texts, illustrations, diagrams and examples in this manual are only intended as aids to help explain the functioning, operation, use and programming of the **GX IEC Developer** IEC programming and documentation system.

For using and usage of this software only the user his own is responsible.

If you have any questions regarding the installation and operation of the software described in this manual, please do not hesitate to contact your sales office or one of your Mitsubishi distribution partners.
You can also obtain information and answers to frequently asked questions from our Mitsubishi website under
www.mitsubishi-automation.com.

MITSUBISHI ELECTRIC reserves the right to change the specifications of its products and/or the contents of this manual at any time and without prior notice.

The IEC 61131.1 standard cited in this manual is available from the publishers Beuth Verlag in Berlin (Germany).

Reference Manual for
MELSOFT GX IEC Developer
Art. No.: 43597

Version			Changes / Additions / Corrections
A	03/1995	ME	First issue
B	05/1996	ME	Software update
C	07/1997	ME	Software update
D	01/1998	ME	Software update
E	08/2000	pdp-rs	Update to software version 2.40
F	06/2001	pdp-rs	Update to software version 4.00
G	05/2002	pdp-rs	Update to software version 5.00
H	09/2003	pdp-ow	Update to software version 6.00
I	10/2004	pdp-ow	Update to software version 6.10
J	09/2005	pdp-ow	Update to software version 7.00
K	12/2006	pdp-ow	Update to software version 7.01

Typographic Conventions

Use of notes

Notes containing important information are clearly identified as follows:

NOTE | Note text

Use of examples

Examples containing important information are clearly identified as follows:

Example ▽ Example text

△

Numbering in figures and illustrations

Reference numbers in figures and illustrations are shown with white numbers in a black circle and the corresponding explanations shown beneath the illustrations are identified with the same numbers, like this:

❶ ❷ ❸ ❹

Procedures

In some cases the setup, operation, maintenance and other instructions are explained with numbered procedures. The individual steps of these procedures are numbered in ascending order with black numbers in a white circle, and they must be performed in the exact order shown:

① Text

② Text

③ Text

Footnotes in tables

Footnote characters in tables are printed in superscript and the corresponding footnotes shown beneath the table are identified by the same characters, also in superscript.

If a table contains more than one footnote, they are all listed below the table and numbered in ascending order with black numbers in a white circle, like this:

① Text

② Text

③ Text

Character formatting and orientation aids

Menu names, menu commands, submenu commands, and dialogue box options are printed in **boldface** type. Examples: The menu item **New** in the menu **Project** or the options **PLC interface** and **Computer Link** in the dialogue box **Transfer-Setup**.

Please keep this manual in a place where it is always available for the users.

Contents

1 Introduction

2 Installation

3 The User Interface

4 Projects and Objects

5 Parameters

6 Programming

8 Compiling and Downloading Programs

11 Support of other Applications

A Glossary

B Keyboard Commands

C Tools

D Temporary Files

E Programming Instructions

F Menu structure

G Compliance List

1 Introduction

1.1 This manual …

… is not designed to be read from cover to cover. It is organised so that it can be used as a reference work in which you can find detailed explanations of all the menus and menu commands in GX IEC Developer when you need them. However, the Reference Manual is not based directly on the menu structure of GX IEC Developer, but rather on the central elements of IEC Standard 61131-3 such as Projects, Objects, Variables, Program Organisation Units and Programming Languages. The menus and all menu commands and options are listed in the Appendix F of this manual.

1.2 The Beginner's Manual…

… is the ideal aid for getting acquainted with the basic terms and concepts of the new IEC 61131-3 standard. This manual teaches you how to use GX IEC Developer in easy steps, with a sample program that provides a practical illustration of the most important operations.

1.3 If you are not yet familiar with Windows…

… please at least read the Windows Fundamentals section in the Windows User's Guide or work through the exercises of the online Windows tutorial (accessible in the Program Manager). This will give you the information and skills you need for using the elements of Windows that are the same in all application programs.

1.4 If you are not yet completely familiar with the IEC 61131-3 standard …

… please read the chapter on the IEC 61131-3 standard in the Beginner's Manual. This chapter explains the basic principles of this industry standard, together with the most important new terms you need to know. A glossary of all the terms used is provided at the end of this Reference Manual.

1.5 If you are already acquainted with IEC 61131-3 and want to get to work right away …

… you will find the basic instructions you need to be able to produce immediate results in the ‘Getting Started' chapter in the Beginner's Manual. This chapter tells you how to perform all the basic GX IEC Developer operations, from creating new projects to downloading your finished programs to the PLC (controller).

1.6 If you need more information on programmable logic controllers …

… please refer to the documentation provided with the hardware components of your system.

1.7 If you need help on programming instructions when you are developing your programs …

… use the GX IEC Developer online help system. This provides direct access to brief explanations of all programming instructions, together with their syntax, parameters and operands. A list of all IEC and MELSEC programming instructions is also provided in the Appendix E of this manual. More detailed descriptions and sample programs can be found in the programming manuals of the individual PLCs.

1.8 If you get stuck …

… in what looks like an insoluble problem or if you have questions about GX IEC Developer or the connected PLC configuration, please first refer to the manuals. Use the detailed index provided in this Reference Manual. Answers to many of your questions can also be found in GX IEC Developer‘s context-sensitive online help system, which can always be accessed by pressing F1. Please also make use of the **Search** command in the **Help** menu.

If you fail to find an answer to your particular question in any of these places you can also obtain help from your local MITSUBISHI ELECTRIC representative. Alternatively, you can call us directly at the MITSUBISHI ELECTRIC European headquarters in Ratingen, Germany. The addresses and phone numbers are listed on the rear of all PLC manuals.

2 Installation

2.1 Hardware requirements

2.1.1 Recommended hardware configuration

- Pentium II 350 processor or above
- 32 MB RAM (Microsoft Windows® 98/Me),
  64 MB RAM (Microsoft Windows® NT/2000)
  128 MB RAM (Microsoft Windows® XP)
- Serial interface (RS-232)
- USB port
- Hard disk with at least 200 MB free space
- CD-ROM drive
- 17" (43 cm) VGA monitor (1024 x 768 pixels)

2.1.2 Software Requirements

GX IEC Developer is a 32-bit product. The following operating systems are supported:

- Microsoft Windows® version 98/Me (with ServicePack 1 or higher)
- Microsoft Windows® NT Workstation 4.0 (with ServicePack 6 or higher)
- Microsoft Windows® 2000 Professional (with ServicePack 2 or higher)
- Microsoft Windows® XP Professional (up to ServicePack 2)
- Microsoft Windows® XP Home Edition (up to ServicePack 2)

Versions of Microsoft Windows which are based on double-byte character sets (e. g. Japanese) are not supported.

2.1.3 Compatibility

File based compatibility with former versions

- GX IEC Developer 7.01 can open project databases (*.pro file) of MM+ 2.1, 2.30a, 2.31, 2.31a, 2.32, 2.40, 2.40a, 2.50, 3.00, GX IEC Developer 4.00, 5.00, 5.01, 5.02, 6.00, 6.01, 6.10, 6.11 and 7.00, but these versions can not open GX IEC Developer 7.01 project databases.
  Exception: GX IEC Developer 7.01 can not open project databases (*.pro file) of MM+ 2.1, 2.30a, 2.31, 2.31a, 2.32, 2.40, 2.40a, 2.50 and 3.00 with Microsoft Windows® Me.
  Such projects can be opened and exported under other supported operating systems. Then the projects can be imported under Microsoft Windows® Me.
- GX IEC Developer 7.01 can not open project databases (*.pro file) of MM+ 2.41 and vice versa.
- GX IEC Developer 7.01 can import projects, which have been exported (*.asc file) by MM+ 2.1, 2.30a, 2.31, 2.31a, 2.32, 2.40, 2.40a, 2.41, 2.50, 3.00, GX IEC Developer 4.00, 5.00, 5.01, 5.02, 6.00, 6.01, 6.10, 6.11 and 7.00, but these versions cannot import *.asc exports of GX IEC Developer 7.01.
- GX IEC Developer 7.01 can open library files (*.lib file) of GX IEC Developer 4.00, 5.00, 6.00, 6.10, 6.11 and 7.00, but these versions cannot open GX IEC Developer 7.01 library files.

PLC project based compatibility with former versions

It is possible, that the generated code of GX IEC Developer 7.01 is different to the generated code of a former version of GX IEC Developer. This could result in the following facts:

- An online change can fail, if the project was not downloaded with GX IEC Developer 7.01.
- The functionality to verify the generated code with the program on the PLC can show differences.

The reason for the differences are internal changes in the code generator which result from the implementation of new features in GX IEC Developer 7.01.

NOTE Please also see section 2.8.1, Code Generation.

Compatibility with other programming software for MELSEC PLCs:

- GX IEC Developer 7.01 can import MELSEC MEDOC/DOS print files (*.tmp) of the versions 1.64b/c, 2.20 and 2.30.
- GX IEC Developer 7.01 generally can not upload parameters and programs from Q and QnA/S CPUs, which are generated by GX Developer or GPPQ.
- GX IEC Developer 7.01 can upload only programs from Q and QnA/S CPUs which are generated according to the limitations of GX IEC Developer 5.00, 5.01, 5.02, 6.00, 6.01, 6.10, 6.11 and 7.00 compared to the Q and QnA/S features (only one program file MAIN.QPG).
- GX IEC Developer 7.01 can upload programs from A CPUs which are generated with GX Developer or GPPA.
- GX IEC Developer 7.01 can not open project databases from GX Developer, GPPQ or GPPA.

There is currently no direct possibility to exchange projects between GX IEC Developer 7.01 and GX Developer, GPPQ or GPPA.

2.1.4 Copyright

WARNING:
This software is protected by copyright. By opening the distribution CD-ROM package you automatically accept the terms and conditions of the License Agreement. You are only permitted to make one single copy of the original distribution disk for your own backup and archiving purposes.

2.1.5 Filling out the registration card

You are only entitled to full technical support from MITSUBISHI ELECTRIC if you fill out and return the enclosed registration card – don't forget to include your serial number! You will then also receive information on updates and new versions of GX IEC Developer.

2.2 Installing GX IEC Developer

During this procedure the installation program will create a directory on your hard disk, into which all the GX IEC Developer files will be copied.

2.2.1 Installing GX IEC Developer on your hard disk

① Make sure that the correct Microsoft Windows version is properly installed on your computer. For information on using Microsoft Windows please refer to the Windows User's Guide.

② Start Microsoft Windows.

③ Insert the installation CD-ROM in the CD-ROM drive.
The GX IEC Developer installation program starts automatically
(if not, execute the file SETUP.EXE on the installation CD-ROM).

④ Follow the instructions that appear on the screen.

⑤ Enter the user name, company name, and serial number of the software.

⑥ Follow the instructions that appear on the screen.

⑦ When the installation procedure is finished the program will create a new program group in the Start menu containing the GX IEC Developer program icon.

For further details on the necessary Microsoft Windows procedures please refer to your Microsoft Windows documentation.

2.3 Starting GX IEC Developer

① In the Start menu click on the GX IEC Developer program icon. The icon is located in: Start > Programs > MELSOFT Application > GX IEC Developer. This starts GX IEC Developer and displays the start-up screen.

② Confirm with the Enter key.

2.4 Quitting GX IEC Developer

You can quit GX IEC Developer directly at any point in the program by pressing the key combination Alt F4.

Or:

Double-click on the **Quit** command in the **Project** menu.

2.5 Bringing GX IEC Developer into the Foreground

When GX IEC Developer is active on the screen at the same time as one or more other applications you can bring GX IEC Developer into the foreground by double-clicking on its icon in the Program Manager.

2.6 The MMP701.INI Program Initialisation File

The MMP701.INI initialisation file contains important parameters required by the Windows application GX IEC Developer. The Install program creates this file automatically and stores it in your Windows directory. A copy of the MMP701.INI file will remain in the actual GX IEC Developer directory (→ page 2-10).

2.6.1 Format of the MMP701.INI File

MMP701.INI is arranged in several sections that all have the same basic format. The following list shows the standard and some sample settings. So your own INI file may look different!

Contents of the MMP701.INI File (Standard and sample entries)	**Explanation**

[PROJECTS]
Project0=C:\MMP_PROJ\SILO1
Project1=C:\MMP_PROJ\TEST3
Project2=C:\MMP_PROJ\EXAMPLE
Project3=C:\MMP_PROJ\SILO2
	PROJECT
	List of the last four projects opened in GX IEC Developer
	These projects are also displayed at the bottom of the **Project** menu.
	DO NOT EDIT!

[CompatibleCpu]
A1=A1N
A2=A2N
FX0/FX0S=FX0 / FX0S
FX2=FX / FX2
FX2C/Vup=FXU / FX2U / FX2C
FX2N=FX2N / FX2NC
Q2A=Q2A(H)
Q2A / Q2AS=Q2A(H)
Q2A-S1=Q2A(H)-S1
Q2A-S1 / Q2AS-S1=Q2A(H)-S1
	COMPATIBLE CPU
	List of compatible CPU types
	DO NOT EDIT!

[OPTIONS]
StepZoom=1
FBZoom=1
MacroGraphics=5
AutomaticIOpin=TRUE
PinOverWrite=TRUE
IncremNewDecl=TRUE
CommentCopy=TRUE
ChkDeclareExternal=TRUE
ChkErrors=25
ChkWarnings=100
LRUProjects=4
ArrayElementsToShow=10
CanEditPou=FALSE
MinDiskSpace=20
DefaultStringLength=32
AutoExportSUL=FALSE
AutoImportSUL=FALSE
AutoIMEXSUL=TRUE
ErrorLogFile=C:\MELSEC\
GX IEC Developer 7.01\ERR.TXT
InitNetworkHeight=4
CreateTmpVarForPLS=TRUE
CreateTmpVarForPLF=TRUE
CreateTmpVarForALT=TRUE
AllowFBCallWithVarOut=FALSE
ErrorForOverlappingElementsExceptLines
=FALSE
AutomaticIOPinENO=FALSE
ErrorForOverlappingElementsExceptLines
=FALSE
LRUVarsel=3
AutoUpdate=TRUE
	OPTIONS
	in the EXTRAS menu
	
	Detailed information see page 2-30

Tab. 2-1: *Format of the MMP701.INI file (1)*

Contents of the MMP701.INI File (Standard and sample entries)	Explanation

MonitorStringLen=16
OnlineChangeFreeSpace=500
ActiveSystemTimer=FALSE
AllTimerActive=FALSE
RD_Enabled=FALSE
RD_ExcludeLatched=FALSE
RD_Include_D_Devices=FALSE
RD_Include_M_Devices=FALSE
RD_Include_T_Devices=FALSE
RD_Include_C_Devices=FALSE
RD_Include_S_Devices=FALSE
RD_Include_ST_Devices=FALSE
MinimizeErrlistboxAfterShow=FALSE
MinimizeVarselAfterDeclare=FALSE
MinimizeFunselAfterDeclare=FALSE
DoNotShowFXOnlineChangeMessage
=FALSE
OnlineToolbar=FALSE
ActionOnZoomIntoFbd=2
ActionOnZoomIntoPou=1
ErrorForOverlappingElementsExceptLines
=TRUE
LDAutoConnect=TRUE
LDDisplayVariableNameInLines=1
LDDoNotAllowQuestionMark=TRUE
LDVariableNameMultiLine=TRUE
LDAllowMultilineInstanceNames=TRUE
LDLimitVariableNameDisplay=TRUE
LDVisibleCharactersVar=12
LDVisibleCharactersLine=8
AdjustSFCActiveInfoTable=FALSE
	OPTIONS
	in the EXTRAS menu
	Detailed information see page 2-30

[Online]
OnlinePollrate=250
RunTimeSystem=Generic
Baudrate=19200
Bytesize=8
Parity=0
Stopbits=1
Port=COM1
Downloader=
	ONLINE
	Online settings

[View]
AutoLoadSaveWS=FALSE
ColorBackground=16777215
ColorForeground=12632256
ColorText=0
ColorSelection=8421504
ColorError=255
ColorBPEnabled=65535
ColorBPActive=65280
ColorMonitoring=65535
ColorTextLocked=8421504
ColorTextReadOnly=8421504
ColorShadeLT=16777215
ColorShadeRB=8421504
ColorGrid=0
ColorTextSelected=16777215
ColorBrowsedItem=16776960
ColorShadeMM=12632256
ColorSFCRasterElement=16777215
ColorCursorOvr=255
ColorCursorIns=16711680
ColorCursorLine=65280
ColorFillCursor=16711680
	COLORS
	The color setting (View - Colors) is also stored in the ini-file (Mmp701.ini) of GX IEC Developer 7.01 and no longer in the project. This means, that changes in the color setting have effect to all new or opened projects.
	The entries in the column on the left represent the default values.
	The entries in the ini-file are reset to the default values after activating the menu item *View - System Defaults.*

[LADDER] HLineLength=4	**LADDER DIAGRAM** ***Horizontal length in Ladder diagram*** Length in grid elements Default setting: 4

Tab. 2-1: *Format of the MMP701.INI file (2)*

Contents of the MMP701.INI File (Standard and sample entries)	**Explanation**
VLineLength=3	***Vertical height in Ladder diagram*** Height in grid elements Default setting: 3
[RC_GROUP] 1=1,53,Clock, , 2=2,73,Security Level, , 3=3,150,Editor Info, , 4=4,40,Online, , 5=5,55,CpuType, , 6=6,110,ConnStatus, Not Connected,;Connection Status Box 7=7,110,ErrorStatus, Not Connected,;Error Status Box 8=8,25,GuidedEditing, ,;Guided Editing Mode 9=9,200,Path,, 10=10,30,System Resources,, 11=11,50,Free Memory,, 12=12,110,Redundant Info,, 13=13,150,MXC status,,	**STATUS BAR** **Status bar configuration settings** **DO NOT EDIT!**

[WORK_GROUP]
LEFT=20,2,8,
RIGHT=1,3,
	STATUS BAR
	Field size of the status bar entries
	DO NOT EDIT!

[DEFAULT_GROUP]
LEFT=2,3,
RIGHT=1,
	STATUS BAR
	Field size of the status bar entries
	DO NOT EDIT!

[LIBRARIES] ManufacturerHelpPath= C:\MELSEC\GX IEC Developer 7.01	**USER LIBRARIES** ***Directory for the library help files***
ManufacturerLibPath= C:\MELSEC\GX IEC Developer 7.01\ LIBSMMP	***Directory for the library files***

Manufacturer0=Standard_Lib,
SLIB.lib,FUNCTION.HLP
 	
Manufacturer1=Manufacturer_Lib,
MLIB.lib,FUNCTION.LIB
	Specification of the libraries
	Max. 50 libraries are allowed.
	Input order:
	ManufacturerX=library name, library file, help file
	(help file not necessary)
	DO NOT EDIT!

[PRINT] PrintFrame=1	**PRINTOUT** =1: Printout options are enabled =0: Printout options are disabled
OemBitmap=C:\MELSEC\ GX IEC Developer 7.01\MEGLOGO.BMP	***Directory and file of bitmap*** Bitmap is shown in the upper left corner in the page header
PrBitmap=C:\GX IEC Developer 7.01\ MMPLOGO.BMP	***Directory and file of bitmap*** Bitmap is shown in the upper right corner in the page header

Text1=Company XXX
Text2=Department XXX
Text3=Application XXX
	Text in footer
	3 text lines are shown in the page footer

PrintRaster=10	***Size of printout*** Default setting:10 depends on printer
ScDoc=1	***ScDoc*** ScDoc=1: Activates the interface to the documentation software SCDoc.

[CrossReference]
GenerateSCT=TRUE
GenerateSCX=TRUE
GenerateSCXAfterBuild=FALSE
	CROSS REFERENCE
	Automatic link of the SCT files to the SCX file when creating the cross reference data base
	=TRUE: Function enabled
	=FALSE: Function not enabled

Tab. 2-1: *Format of the MMP701.INI file (3)*

Contents of the MMP701.INI File (Standard and sample entries)	**Explanation**

[ScDoc]
ScPath=
NewBack=FALSE
NewBack1=FALSE
Changed=FALSE
Linkrev=FALSE
CallSc=FALSE
ScdPath=
Link=FALSE
LastPtr=
	SCDOC
	Section containing additional printer settings

[PARAM] COMPORT=0	**PARAMETER** ***Selection of the COM port of Personal Computer*** COMPORT=0 Selection of COM1 COMPORT=1 Selection of COM2 The selection of the COM port will not be saved together with the project. The COM port can be selected in the MMP701.INI.

[PATH]
CNF=C:\MELSEC\GX IEC Developer 7.01\ TYP_CNF
TYP=C:\MELSEC\GX IEC Developer 7.01\ TYP_CNF
EXE=C:\MELSEC\GX IEC Developer 7.01
LIB=C:\MELSEC\GX IEC Developer 7.01\ LIBSMMP
PPR=C:\MELSEC\GX IEC Developer 7.01\ PPR
MODEM=C:\MELSEC\GX IEC Developer 7.01\ MODEMSUP
16BITEXE=C:\MELSEC\GX IEC Developer 7.01\ MMPCNV\SC.EXE
PARAM=C:\MELSEC\GX IEC Developer 7.01\ DEFPARAM
GCDP=C:\MELSEC\GX IEC Developer 7.01\..\
	DIRECTORIES
	CNF Configuration files for all PLC types (special format)
	TYP Configuration files for all PLC types (standard format)
	EXE GX IEC Developer program files
	PPR Project templates for all PLC types
	DO NOT EDIT!

[IEC_Export]
ExportFile=VARLIST.CSV
Separator=;
OverwriteExportFile=TRUE
AutoGenerateExportFile=FALSE
	IEC-EXPORT
	Section containing settings for the IEC export functionality

[WIZARD] MELSEC=TRUE	**WIZARD** ***Automatic POU and Task creation*** MELSEC=TRUE When you create a new project the Wizard is displayed on the screen with a prompt asking if you wish to create a POU and a task automatically. MELSEC=FALSE This setting disables the Wizard.
[SFC] GenerateSfcStartupCode=TRUE	**SEQUENCES** ***Automatic generation of startup code*** ***LD M9039*** ***OUT M9101*** The special relay M9101 starts the execution of the SFC processing. 1) The section [SFC] or the entry GenerateSfcStartupCode does not exist or any of them is spelled incorrectly: ➜ The startup code will be generated automatically. 2) The section [SFC]and the entry GenerateSfcStartupCode exist and the value is set to either TRUE, YES or 1: ➜ The startup code will be generated automatically. 3) The section [SFC]and the entry GenerateSfcStartupCode exist and the value is set to any other value: ➜ The startup code will not be generated. Usually the entry should be set to TRUE, to generate a proper SFC processing.

Tab. 2-1: *Format of the MMP701.INI file (4)*

Contents of the MMP701.INI File (Standard and sample entries)	**Explanation**
[REG] NAME: xxxxx COMPANY: xxxxx DISTRIBUTOR: xxxxx SERIAL: xxxxx	**GX IEC Developer REGISTRATION** Name/Company: information entered when you install GX IEC Developer Distributor/Serial: Distributor name and encoded serial number **DO NOT EDIT!**
[Ccodegen] TabDump=TRUE	**CODE-GENERATOR** When TabDump=TRUE the following files are generated in the project directory when GX IEC Developer is started with the /debug option: - PRECGEN.TAB - POSTCMPR.TAB - POSTCGEN.TAB
[devel] eds=TRUE	
[DEVICE_SETTING] USE_COLORS=1	**DEVICE EDITOR** ***Colors*** Context menu - Options - Use Colors =1: Use of different colors in display mode enabled =0: Use of different colors in display mode disabled
SAFETY_INQUIRY=1	***Safety inquiry*** Context menu - Options - Additional Safety Inquiry =1: Additional safety inquiry prompt is displayed =0: Additional safety inquiry prompt is not displayed

[EDM]
FIELD0=Name
FIELD1=Type
FIELD2=Address (IEC)
FIELD3=Value (dec)
FIELD4=Value (bin)
ObjListInclLibraries=1
	ENTRY DATA MONITOR
	Configuration of Entry Data Monitor table

Tab. 2-1: *Format of the MMP701.INI file (5)*

2.6.2 Editing MMP701.INI Entries

⚠ **WARNING:**
Most MMP701.INI parameters should not be edited by the user! Please observe the **DO NOT EDIT!** ***warnings in the table above.***

How to make changes in the user-editable sections

① Create a backup copy of the file MMP701.INI in another directory or on a floppy disk.

If the changes you make do not work or result in unexpected problems when you run GX IEC Developer you can then restore the original file in the Windows directory from the backup copy.

Important: When you install GX IEC Developer the MMP701.INI file is copied into your Windows directory. Although there is also a copy of MMP701.INI in the GX IEC Developer directory, the program only uses the copy of MMP701.INI in the Windows directory. Before making any editing changes it is thus always advisable to make a backup of the version of MMP701.INI stored in the Windows directory, as it is always possible that the file has already been changed in some way and is no longer identical with the copy stored in the GX IEC Developer directory.

② Load MMP701.INI into a standard text editor such as the Windows Notepad utility.

Do not use an editor or word processor that inserts formatting codes in the text file, as the system would then no longer be able to read the file.

③ Make the necessary editing changes.

④ Save the edited file.

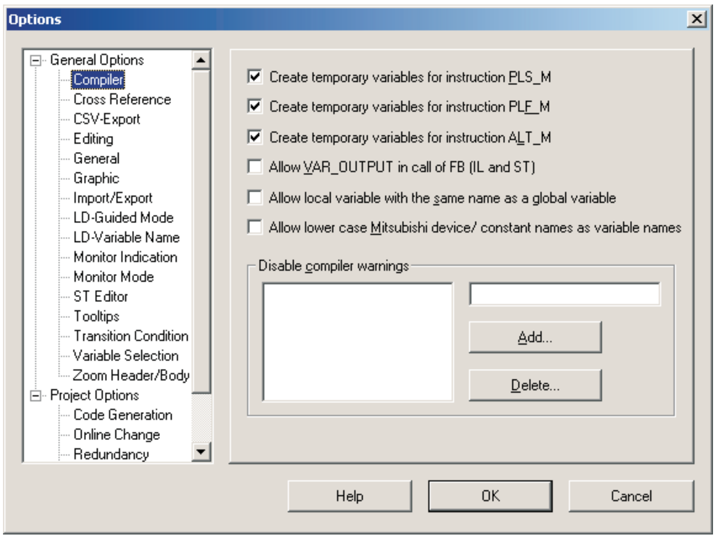
⑤ Close the GX IEC Developer program and restart it.

Changes made to the MMP701.INI file do not take effect until you restart the GX IEC Developer program.

2.7 General Options

The options are configured in the **Extras - Options** menu.

2.7.1 Compiler



☑ Create temporary variables for instruction PLS_M

☑ Create temporary variables for instruction PLF_M

☑ Create temporary variables for instruction ALT_M

☐ Allow VAR_OUTPUT in call of FB (IL and ST)

☐ Allow local variable with the same name as a global variable

☐ Allow lower case Mitsubishi device/ constant names as variable names

Disable compiler warnings

Fig. 2-1:
Options - Compiler

Create temporary variables for instruction PLS_M / PLF_M / ALT_M

System variables are required for internal processing when the output variables of the instructions PLS_M, PLF_M and ALT_M are used to activate additional instructions. If you use many instructions of this kind the maximum limit for number of system variables (system labels) that you can use can be reached very quickly.

If this happens you should disable the option **Create temporary variables for instruction PLS_M / PLF_M / ALT_M**. Your PLC program must then be modified accordingly.

Default: Enabled

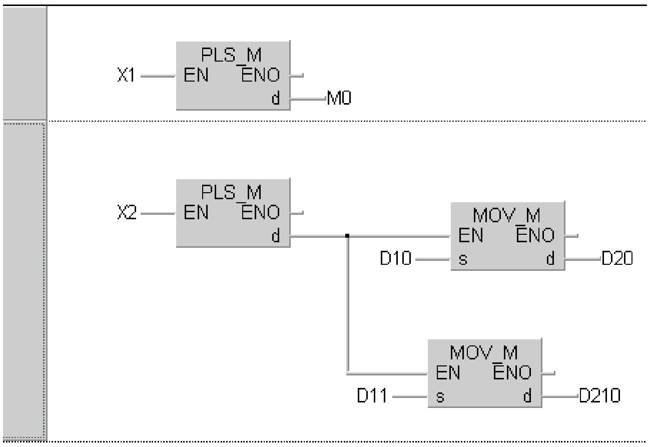


Fig. 2-2:
Output writes directly to the device (no system variable required)
Output activates the MOV instruction (uses 1 system variable)

Allow VAR_OUTPUT in call of FB (IL ans ST)

If checked it is allowed to use a VAR-OUTPUT assignment directly with the call of a FB instance. E.g. CAL FBInst (FBVarOut:=Variable).

Allow local variable with the same name as a global variable

If checked, global variable hiding will not be considered to be an error. This option allows to choose if global variable hiding should be considered to be a programming error or not. By default, if a local variable of a POU has the same name as a global variable, an error will be shown when the POU is checked. If you turn this option on, global variable hiding will be allowed and the POU will be compiled.

Allow lower case Mitsubishi device/constant names as variable names

This option is used to give the user a hint, that a Mitsubishi Device or Constant could be hidden by a variable declaration with lower case letters.

- If the option is checked, a warning is shown when a Mitsubishi device/constant name is used in a lower case format as variable name.
- If the option is unchecked, an error is shown when a Mitsubishi device/constant name is used in a lower case format as variable name.

Disable Compiler Warnings

The display of compiler warnings in the error box during a **Check**, **Build**, or **Rebuild all** execution can be disabled. Default setting: Warnings enabled.

To disable a compiler warning, take the following steps:

① Enter the compiler warning code in the edit field.

② Press the **Add...** button.

The compiler warning is disabled and listed in the list box.

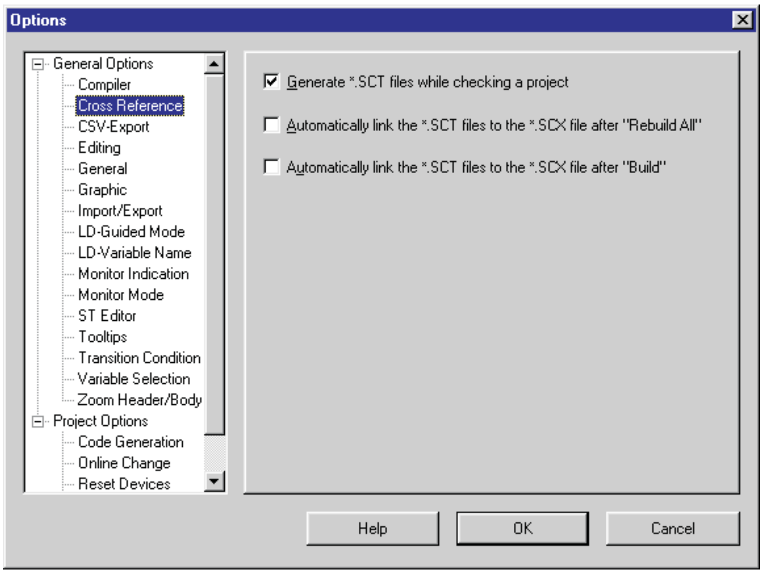
To enable a compiler warning, take the following steps:

① Select a compiler warning code from the list box.

② Press the **Delete...** button.

The compiler warning is enabled and removed from the list box.

2.7.2 Cross Reference



☑ Generate *.SCT files while checking a project

☐ Automatically link the *.SCT files to the *.SCX file after "Rebuild All"

☐ Automatically link the *.SCT files to the *.SCX file after "Build"

Fig. 2-3:
Options - Cross Reference

Cross Reference Options

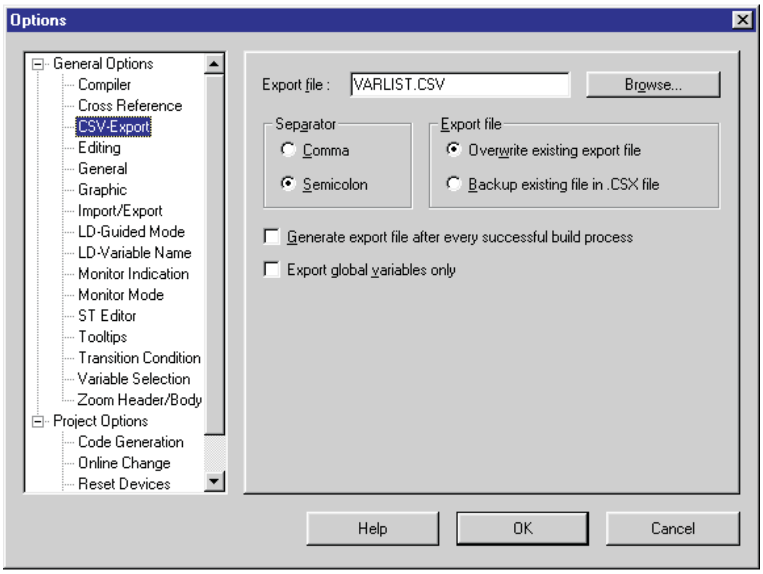
Used to define the options for generation of the cross reference database.

Generate *.SCT files while checking a project: Activates generation of the SCT files during project checks.

Automatically link the *.SCT files to the *.SCX file after ‘Rebuild All': Activates automatic linking of the SCT files to the SCX file after compiling a project.

Automatically link the *.SCT files to the *.SCX file after ‘Build': Activates automatic linking of the SCT files to the SCX file after building a project.

2.7.3 CSV-Export



Export file : VARLIST.CSV

Separator
○ Comma
● Semicolon

Export file
● Overwrite existing export file
○ Backup existing file in .CSX file

□ Generate export file after every successful build process

□ Export global variables only

Fig. 2-4:
Options - CSV-Export

All variables defined in the GVL and the POU header of a project can be exported as ASCII file in Comma Separated Value format (*.csv).

Export File: Enter the path and name of the file to be exported

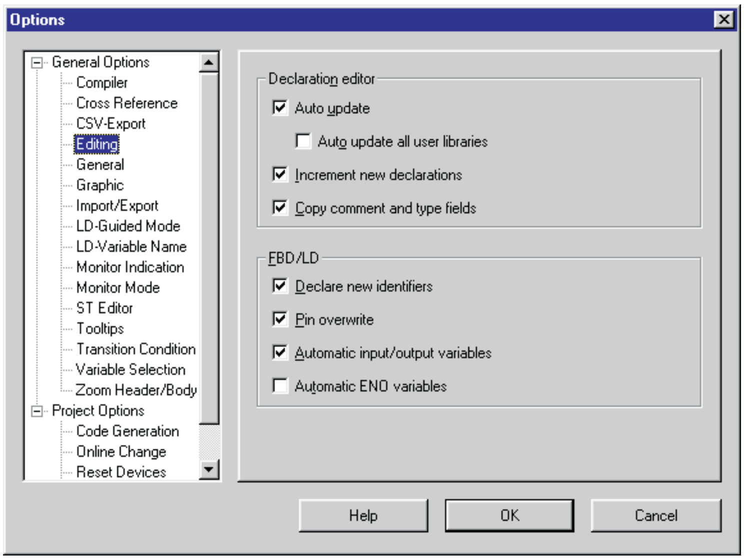
Separator: Select either the “Comma” or the “Semicolon” as separator character between values.

Export file: Select whether to either “Overwrite existing export file” or to “Backup existing file in .CSX file”.

Generate export file after every successful build process: Activate this option, if desired.

Export global variables only: If this option is checked, only global variables are exported by the **Extras** → **Export Variables** functionality. Local variables and type info are omitted.

2.7.4 Editing



Declaration editor

☑ Auto update

☐ Auto update all user libraries

☑ Increment new declarations

☑ Copy comment and type fields

FBD/LD

☑ Declare new identifiers

☑ Pin overwrite

☑ Automatic input/output variables

☐ Automatic ENO variables

Fig. 2-5:
Options - Editing

Declaration Editor

Auto update: Updates global and local variables automatically after changing a variable declaration. The following changes of a variable declaration are updated: Class, Identifier, Address, Type, Initial. All changes made in the GVL will also be executed in the POUs. When you change a POU header, its body will also be changed automatically. The variables are updated after the declaration editor changes have been saved.

Auto update all user libraries: Global variables defined in the project GVL can also be referenced from POUs in a user library, but those variables are not updated by the **Auto update**. Similarly when a global variable defined in a user library GVL is changed, its references are not updated in other open user libraries, only in the user library having the global variable, and in the project POUs. Checking this option automatic updates all open user libraries. If there are no references in the user libraries to global variables of the project or another user library, you should turn this option off (uncheck) to enhance the performance of auto update.

Incremental New Declarations: When this option is enabled the contents of the previous lines are copied automatically when new lines are inserted in the declaration tables. The designators and addresses are automatically incremented by 1. If you also want the system to copy the contents of the type and comment fields you must also enable the *Copy Comment and Type Fields* check box.

Copy Comment and Type Fields: See **Incremental New Declarations**.

FBD/LD

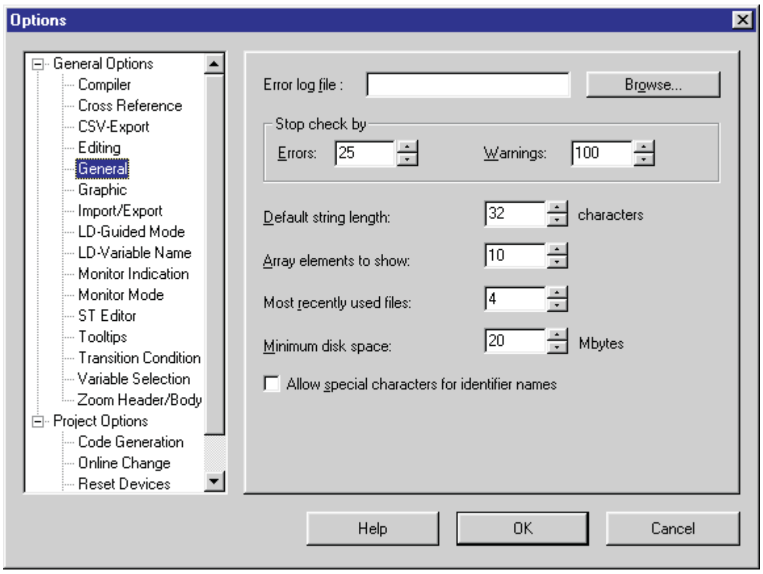
Declare New Identifiers: When you overwrite a dummy variable in a graphical editor with a variable identifier name that has not yet been declared the **Variable Selection (Mode NewVar)** dialogue box will be displayed automatically so that you can enter the declaration for the new variable.

Pin Overwrite: When you create a connection to an empty variable connection in a graphical editor the empty variable will be overwritten.

Automatic Input/Output Variables: The system will automatically create the input and output variables when you insert a function block in a graphical editor.

Automatic ENO Variables: The system will automatically create the ENO output variable when you insert a function block in a graphical editor. If the ENO output variable is not needed this option has to be disabled.

2.7.5 General



Error log file :

Stop check by

Errors: 25 Warnings: 100

Default string length: 32 characters

Array elements to show: 10

Most recently used files: 4

Minimum disk space: 20 Mbytes

☐ Allow special characters for identifier names

Fig. 2-6:
Options - General

Error Log File

Path and filename for the error log file.

Stop Check By

Errors: Stop the check operation when the number of errors found exceeds the maximum limit specified here.
Range: 1 - 9,999

Warnings: Stop the check operation when the number of warnings exceeds the maximum limit specified here.
Range: 1 - 9,999

Default string length (characters): You can set the default length of the STRING data type to any value up to a maximum of 50 characters.

Array elements to show: The Entry Data Monitor can display the individual elements of an array, but this can be impossible if the array is too large. This option enables you to set the number of array elements to display to a maximum of 50 (→ page 9-56).

You can also change this value in the prompt dialogue that is displayed when you open an array in the Entry Data Monitor.

Most recently used files (number): The last projects opened are always displayed at the bottom of the **Project** menu. You can set the number of projects to be displayed to a maximum of 9.

Minimum disk space (Mbytes): GX IEC Developer displays a warning message when the amount of free space available on the current drive falls below the value specified here.

Allow special characters for identifier names

If checked, special characters can be used for identifier names. The following special characters are supported as valid characters:

- ä, Ä, ö, Ö, ü, Ü, ß, _, å, Å

The usage of these characters is valid for the following identifier and all functionalities where these identifiers are used (e.g. Variable selection dialogue):

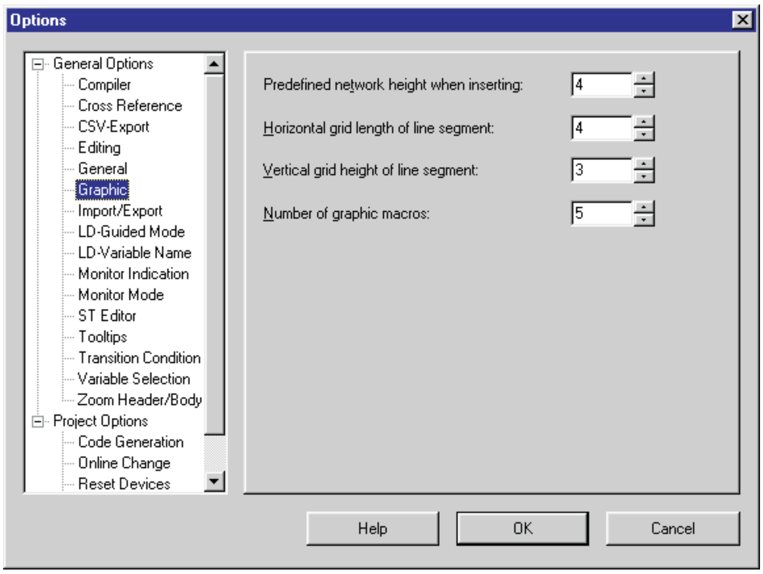
- Library names
- Task names
- DUT names and DUT component names
- Global and local variable names
- POU names (PRG, FUN, FB)
- Network label names
- SFC Transition names
- SFC Action names
- SFC Step names
- SFC Jump label names

The character '_' (underscore) was already allowed in older versions of GX IEC Developer. Due to the restrictions by the IEC naming convention it was not allowed to use a double underscore character at the beginning of an identifier (e.g. "__TestVar"). Additionally there was a restriction for the usage of the underscore character as the last character of an identifier (e.g. "TestVar_"). These restrictions can now be disabled by the user.

These identifiers can be used in every program editor:

- LD
- FBD
- IL
- MELSEC-IL
- SFC
- ST

2.7.6 Graphic



Predefined network height when inserting: 4

Horizontal grid length of line segment: 4

Vertical grid height of line segment: 3

Number of graphic macros: 5

Fig. 2-7:
Options - Graphic

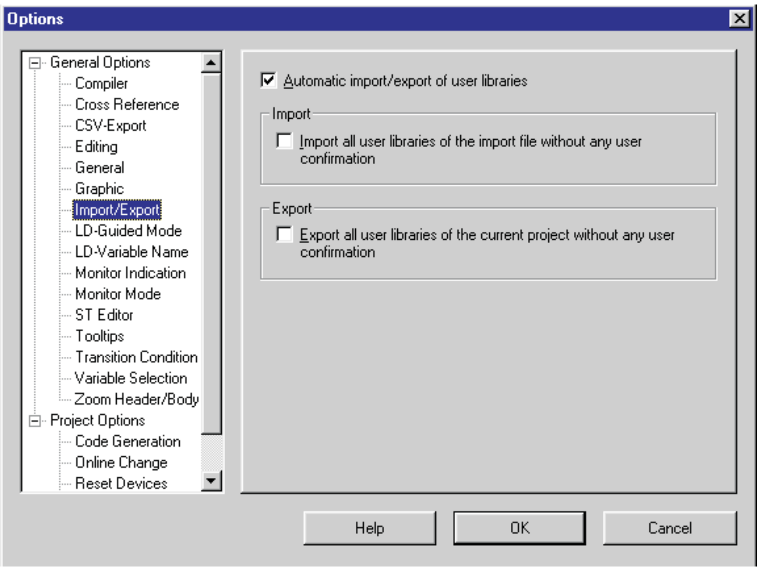
Predefined network height when inserting: The initial height of a network.
Range: 4 - 100 grid points
Default: 4

Horizontal grid length of line segment: Horizontal length of line segments in the ladder diagram editor, in grid points.
Range: 1 - 100 grid points
Default: 4

Vertical grid height of line segment: Vertical length of line segments in the ladder diagram editor, in grid points.
Range: 1 - 100 grid points
Default: 3

Number of graphic macros: Number of additional graphic macros available.
Graphic macros 1 through 5 are always available.
Range: 5 - 25
Default: 5

2.7.7 Import/Export



Options

- General Options
  - Compiler
  - Cross Reference
  - CSV-Export
  - Editing
  - General
  - Graphic
  - Import/Export
  - LD-Guided Mode
  - LD-Variable Name
  - Monitor Indication
  - Monitor Mode
  - ST Editor
  - Tooltips
  - Transition Condition
  - Variable Selection
  - Zoom Header/Body
- Project Options
  - Code Generation
  - Online Change
  - Reset Devices

☑ Automatic import/export of user libraries

Import

□ Import all user libraries of the import file without any user confirmation

Export

□ Export all user libraries of the current project without any user confirmation

Fig. 2-8:
Options - Import/Export

Import/Export options

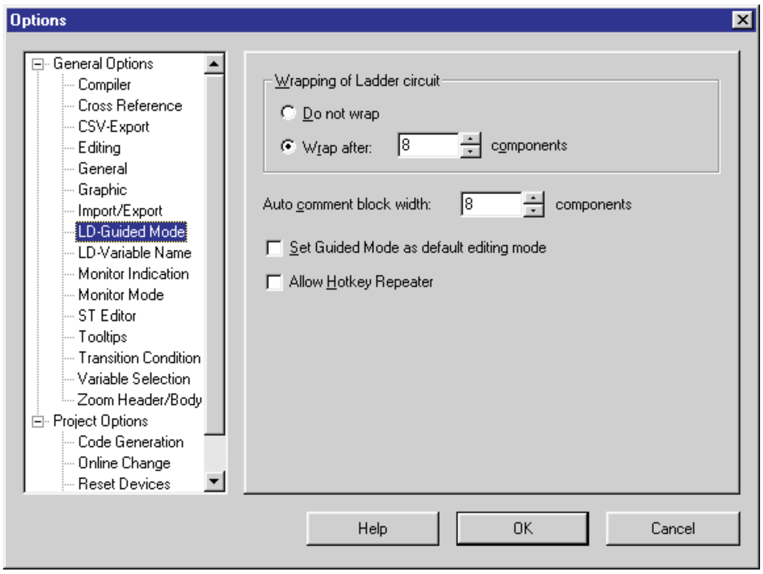
Specifies how the user libraries are handled when a project is imported or exported.

Automatic import/export of user libraries:
Enables the automatic import or export of the user libraries.

Import all user libraries of the import file without any user confirmation:
Imports all user libraries without requesting any confirmation.

Export all user libraries of the current project without any user confirmation:
Exports all user libraries without requesting any confirmation.

2.7.8 LD-Guided Mode



Wrapping of Ladder circuit

○ Do not wrap

● Wrap after: 8 components

Auto comment block width: 8 components

□ Set Guided Mode as default editing mode

□ Allow Hotkey Repeater

Fig. 2-9:
Options - LD-Guided Mode

Wrapping of Ladder circuit

Do not wrap: Do not wrap the ladder circuit. This setting is applicable for the circuit that is going to be edited after making this setting and not for the pre-created circuits.

Wrap after: Wrap if the ladder circuit exceeds the nth component position. This setting is applicable for the circuit that is going to be edited after making this setting and not for the pre-created circuits.

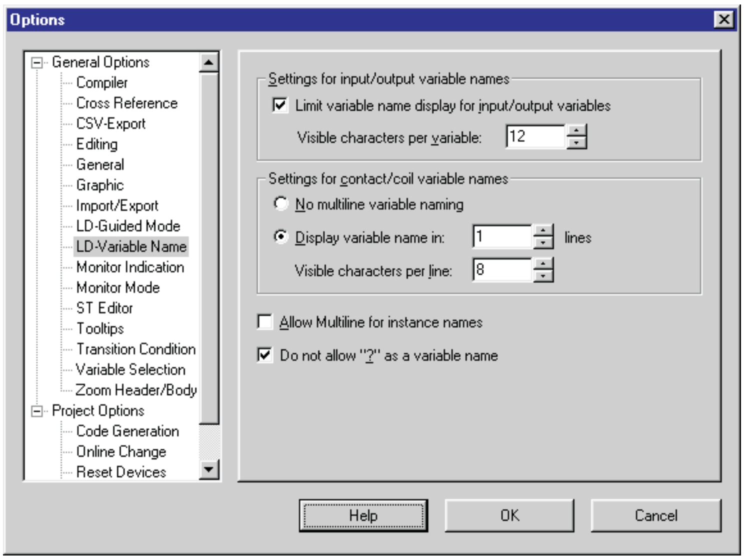
Components: The number of components to wrap after.
Range: 2 - 100
Default: 8

Auto comment Block width / Components: Set the auto comment block with a width of n components in the network.
Range: 1 - 100
Default: 8

Set Guided Mode as default editing mode: The LD editor should be opened in Guided Mode by default.

Allow Hotkey Repeater: The hotkey repeater window will be displayed whenever the Horizontal or Vertical Line hotkey is pressed or accessed through the menu.

2.7.9 LD-Variable Name



Settings for input/output variable names

☑ Limit variable name display for input/output variables

Visible characters per variable: 12

Settings for contact/coil variable names

○ No multiline variable naming

● Display variable name in: 1 lines

Visible characters per line: 8

□ Allow Multiline for instance names

☑ Do not allow "?" as a variable name

Fig. 2-10:
Options -
LD-Variable Name

Settings for input/output variable names

Limit variable name display for input/output variables: When this option is checked, the display of variable names for input/output variables is limited to the specified number of characters.
default: unchecked

Visible characters per variable: Maximum number of characters for displaying input/output variable names
Range: 2–255 characters
default: 12

Settings for contact/coil variable names

No multiline variable naming: When this option is checked, variable names are displayed in one line.

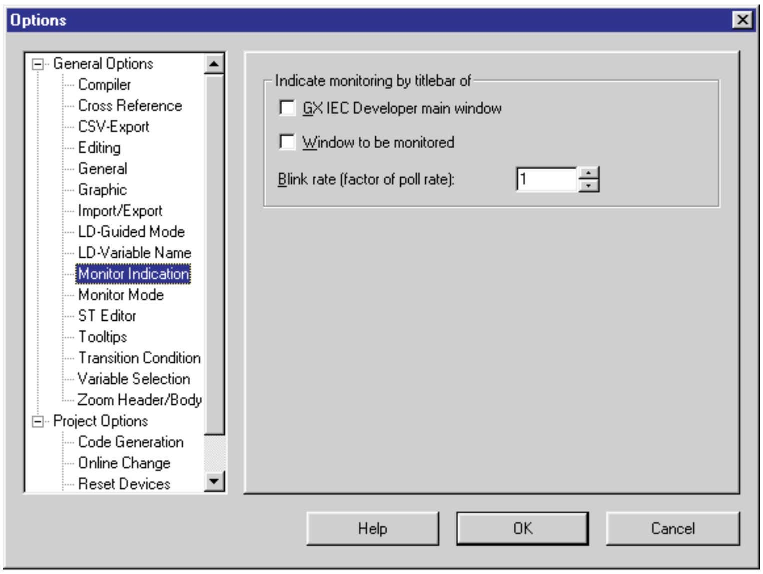
Display variable name in: When this option is checked, a variable name will be wrapped or folded in n lines in the network display.
Range: 1–4 lines
Default: 1

Visible characters per line: Maximum number of characters for displaying multiple line variable names.
Range: 2–255 characters
default: 8

Allow Multiline for instance names: If checked, variable names for function blocks will be displayed in multiple lines. The number of characters per line depends on the function block width.

Do not allow “?” as a variable name: Do not allow the user to give the “?” mark as variable name in Guided Mode. When this option is checked, you must specify a proper variable name whenever you place a new program component in the network.

2.7.10 Monitor Indication



Indicate monitoring by titlebar of

□ GX IEC Developer main window

□ Window to be monitored

Blink rate (factor of poll rate): 1

Fig. 2-11:
Options - Monitor Indication

If the monitor mode is active, the corresponding icon in the toolbar changes its colour from black to red.

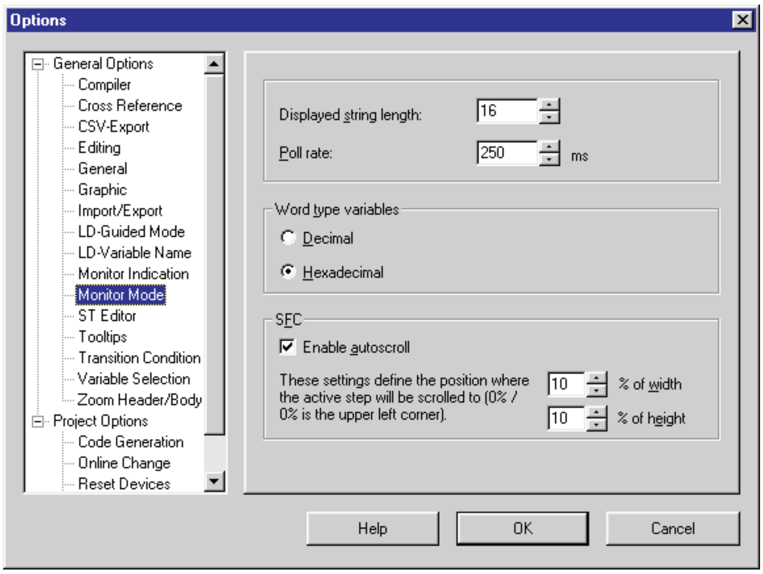
Indicate Monitoring by Title bar of

GX IEC Developer main window: If the monitor mode is active and objects are monitored in one window at least (e.g. body of a POU or Entry Data Monitor), the text "Monitoring" in the title bar of GX IEC Developer is blinking according to the defined blink rate.

Window to be monitored: If the monitoring is started for a window (e.g. body of a POU or Entry Data Monitor), the text "Monitoring" in the title bar of this window is blinking according to the defined blink rate.

Blink Rate: The blink rate can be adjusted depending on the poll rate.
Range: 1 - 10
Default: 1

2.7.11 Monitor Mode



Displayed string length: 16

Poll rate: 250 ms

Word type variables

○ Decimal

● Hexadecimal

SFC

☑ Enable autoscroll

These settings define the position where the active step will be scrolled to (0% / 0% is the upper left corner). 10 % of width 10 % of height

Fig. 2-12: Options - Monitor Mode

Displayed String Length: Displaying long strings significantly reduces performance in monitoring mode. It is thus advisable to limit the number of characters to be displayed.
Range: 2 - 50
Default: 16

Poll rate: Interval at which the monitor polls the CPU, expressed in ms.
Range: 1 - 5,000 ms
Default: 250 ms

Word type variables

Decimal: Displays monitor values of word type variables in decimal format.

Hexadecimal: Displays monitor values of word type variables in hexadecimal format (default setting).

This feature is also available in editors of the type IL, LD, FBD, and MIL via a context menu when the monitor mode is active.

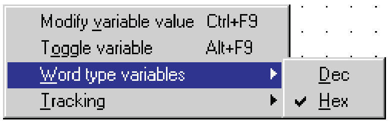


Fig. 2-13: Indication of word type variables

SFC

If the SFC body is observed in the monitor mode, you have to scroll through the window. This can be done manually or by means of the autoscroll function.

The autoscroll function tracks the active steps of the SFC and scrolls the window automatically. The horizontal and vertical position of the active step in the window can be defined.

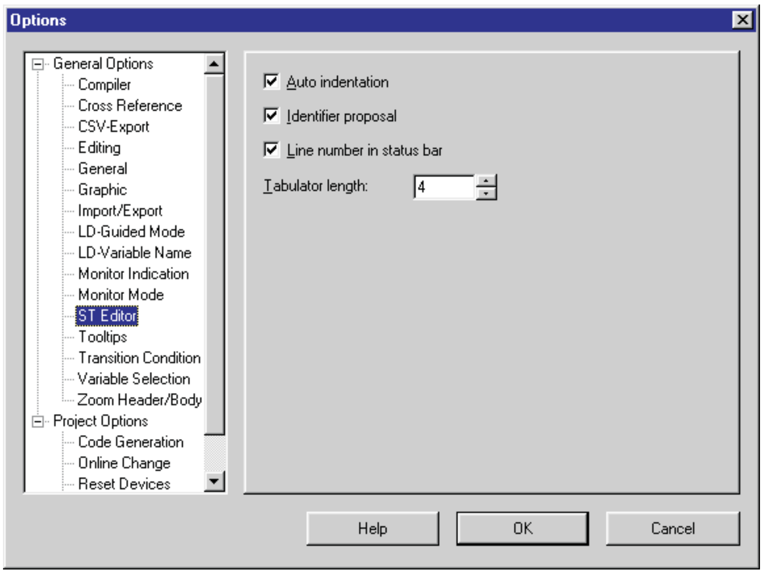
If the SFC includes more than one active step, the upper left active step is used for autoscrolling.

Even an active macro step is opened and the active step inside the macro step is scrolled into view. If all steps get inactive in the SFC macro step, it is closed and the former shown level is shown again.

If you have zoomed into a macro step before you started the monitoring or during monitoring, this level is not left for the next higher level even when no step is active.

Enable autoscroll: These settings define the position where the active step will be scrolled to (0 % / 0 % is the upper left corner).
Range: 0 - 100 %
Default: 10 %

2.7.12 ST Editor



☑ Auto indentation
☑ Identifier proposal
☑ Line number in status bar
Tabulator length: 4

Fig. 2-14:
Options - ST Editor

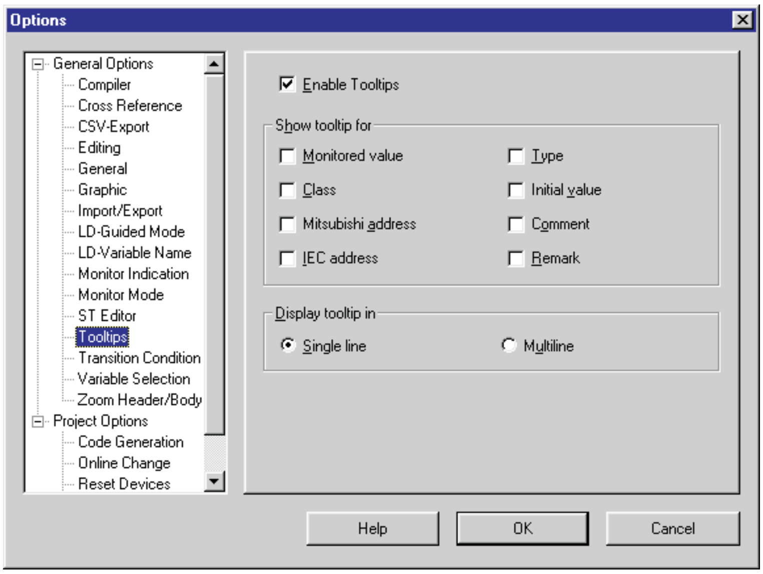
Auto indentation: Auto indentation

Identifier proposal: Identifier proposal

Line number status: Line number status

Tabulator length: Set Tabulator length
Range: 1 - 64
Default: 4

2.7.13 Tooltips



☑ Enable Tooltips

Show tooltip for

□ Monitored value	□ Type
□ Class	□ Initial value
□ Mitsubishi address	□ Comment
□ IEC address	□ Remark

Display tooltip in

● Single line ○ Multiline

Fig. 2-15:
Options - Tooltips

Enable Tooltips: This checkbox when checked enables the Tooltips without resetting the display options.

Show tooltip for

Monitored value: If checked, tooltips are shown for monitored values.

Class: If checked, tooltips are shown for classes.

Mitsubishi address: If checked, tooltips are shown for Mitsubishi addresses.

IEC address: If checked, tooltips are shown for IEC addresses.

Type: If checked, tooltips are shown for types.

Initial value: If checked, tooltips are shown for initial values.

Comment: If checked, tooltips are shown for comments.

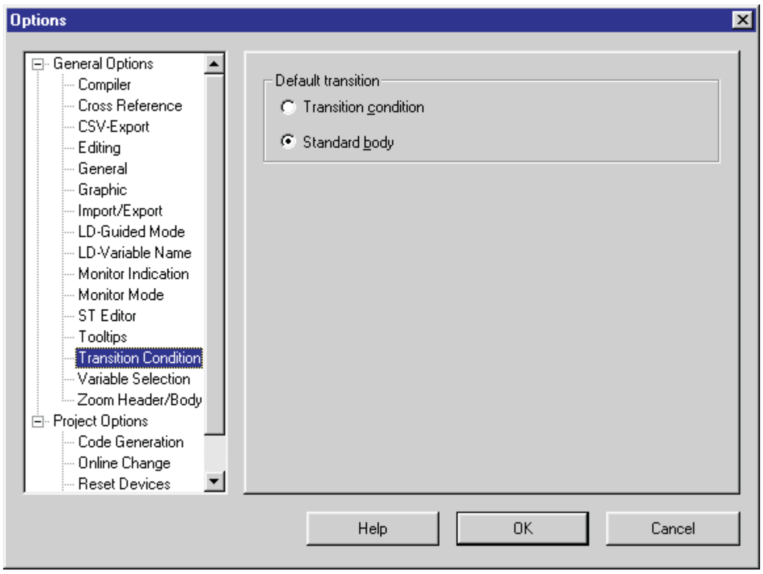
Remark: If checked, tooltips are shown for remarks.

Display tooltip in

Single line: Tooltips are shown in one single line.

Multiline: Tooltips are shown in multiple lines, if necessary.

2.7.14 Transition Condition



Default transition

○ Transition condition

● Standard body

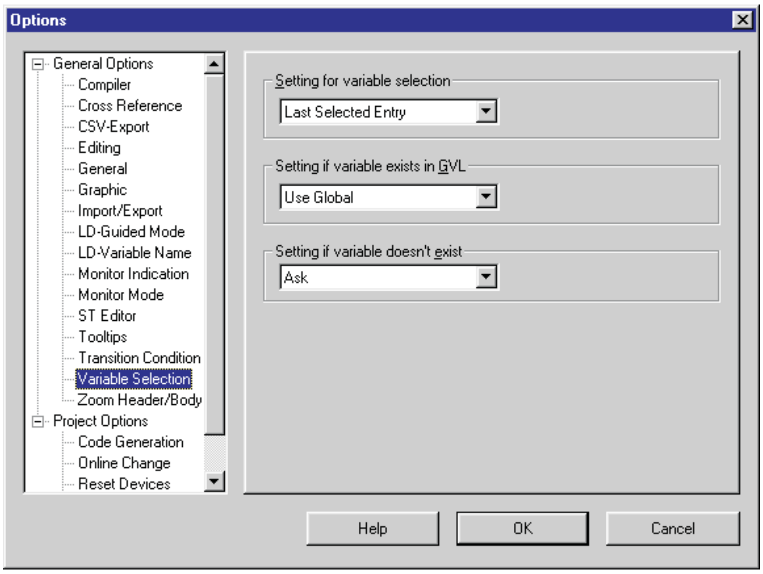
Fig. 2-16: Options - Transition Condition

When opening the transition's rectangle (e.g. by double click or by **Zoom Into**)

Transition Condition: is selected.
The TC editor will appear.

Standard Body: is selected.
The standard body editor will appear for new transitions.

2.7.15 Variable Selection



Options

Setting for variable selection: Last Selected Entry

Setting if variable exists in GVL: Use Global

Setting if variable doesn't exist: Ask

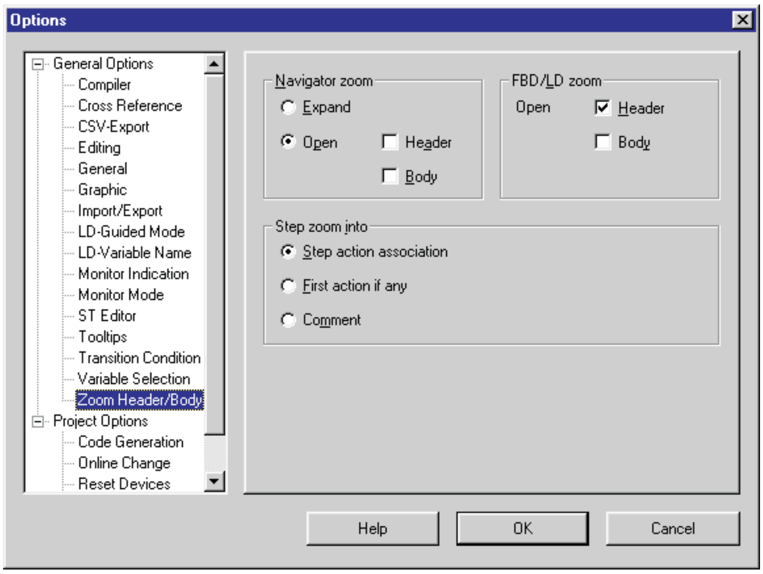
Fig. 2-17: *Options - Variable Selection*

Setting for Variable Selection: Determines which library is pre-initialised when the Variable Selection dialogue is shown.
Range: Header, Global Variable, Last Selected Entry
Default: Last Selected Entry

Setting if variable exists in GVL: Determines which library is pre-initialised when the Variable Selection dialogue is shown from the LD/FBD when you type a variable name that is not defined in the header but exists in the GVL.
Range: Ask, Use Global, Define Local
Default: Use Global

Setting if variable doesn't exists: Determines which library is pre-initialised when the Variable Selection dialogue is shown from the LD/FBD when you type a variable name that is not defined either in the header nor the GVL.
Range: Ask, Define Global, Define Local
Default: Ask

2.7.16 Zoom Header/Body



Navigator zoom

○ Expand

● Open □ Header

□ Body

FBD/LD zoom

Open ☑ Header

□ Body

Step zoom into

● Step action association

○ First action if any

○ Comment

Fig. 2-18:
Options - Zoom Header/Body

Navigator Zoom

Expand: Double-clicking on entries in the POU Pool displays/hides lower levels in the hierarchy (default).

Open Header: Double-clicking on entries in the POU Pool opens the corresponding header.

Open Body: Double-clicking on entries in the POU Pool opens the corresponding body.

FBD/LD Zoom

Open Header: Double-clicking on a function block in a graphical editor opens the corresponding header (default).

Open Body: Double-clicking on a function block in a graphical editor opens the corresponding body.

Step Zoom Into

Step Action Association: Double-clicking on a step opens the **Action Association** dialogue box (association of an action to a step).

First Action If Any: If actions are already associated with the step double-clicking on the step displays the PLC program of the first action in the list.

Comment: Double-clicking on a step displays the **Comment** dialogue box (for entering comment texts).

2.8 Project specific options

The options are configured in the **Extras - Options** menu.

The MM+ versions 2.40 and higher provide the possibility to define project specific settings. The project specific settings will be saved together with the project in the database.

Availability

Due to the fact that the project specific options belong to a specific project and are no general settings, these options in the **Options** dialogue are available only if a project has been loaded.

To indicate that these settings refer only to the project currently loaded, the title of the **Options** dialogue is **Project Specific Options**.

Database

Opening old projects

When opening old projects, i.e. projects created by lower MM+ versions, missing settings will be filled up with the settings included in the **.INI** file. If the entries in the **.INI** file are not available, hard coded settings will be inserted instead.

Creating new projects

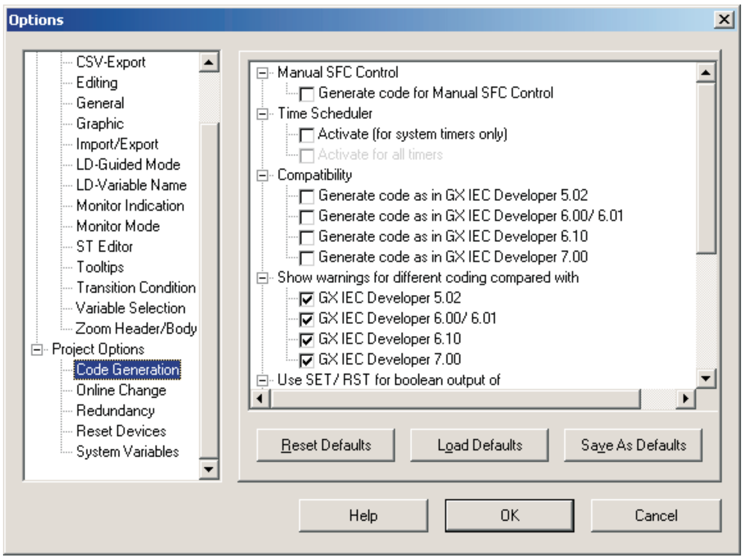
When creating new projects the individual default values included in the **.INI** file are inserted. If the entries do not exist in the **.INI** file, the hard coded default settings are inserted instead. This behaviour is exactly the same as to open the **Options** dialogue and pressing the button **Load Defaults** of both sections.

Compatibility

Due to the necessary changes in the database a project created by means of MM+ 2.40 or higher cannot be opened by lower MM+ versions.

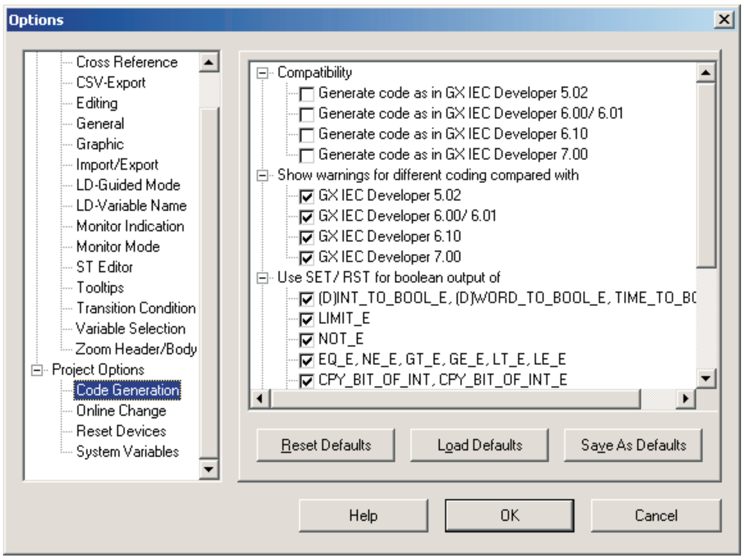
Files with the extensions *.PRO and *.PCD generated by MELSEC MEDOC *plus* 2.50 or lower can not be opened by the current version of GX IEC Developer. In this case we recommend the import/export functions or an external conversion tool (available on request).

2.8.1 Code Generation



- Manual SFC Control
  - □ Generate code for Manual SFC Control
- Time Scheduler
  - □ Activate (for system timers only)
  - □ Activate for all timers
- Compatibility
  - □ Generate code as in GX IEC Developer 5.02
  - □ Generate code as in GX IEC Developer 6.00/ 6.01
  - □ Generate code as in GX IEC Developer 6.10
  - □ Generate code as in GX IEC Developer 7.00
- Show warnings for different coding compared with
  - ☑ GX IEC Developer 5.02
  - ☑ GX IEC Developer 6.00/ 6.01
  - ☑ GX IEC Developer 6.10
  - ☑ GX IEC Developer 7.00
- Use SET/ RST for boolean output of

Fig. 2-19:
Options - Code Generation for QnA and System Q series



- Compatibility
  - □ Generate code as in GX IEC Developer 5.02
  - □ Generate code as in GX IEC Developer 6.00/ 6.01
  - □ Generate code as in GX IEC Developer 6.10
  - □ Generate code as in GX IEC Developer 7.00
- Show warnings for different coding compared with
  - ☑ GX IEC Developer 5.02
  - ☑ GX IEC Developer 6.00/ 6.01
  - ☑ GX IEC Developer 6.10
  - ☑ GX IEC Developer 7.00
- Use SET/ RST for boolean output of
  - ☑ (D)INT_TO_BOOL_E, (D)WORD_TO_BOOL_E, TIME_TO_BC
  - ☑ LIMIT_E
  - ☑ NOT_E
  - ☑ EQ_E, NE_E, GT_E, GE_E, LT_E, LE_E
  - ☑ CPY_BIT_OF_INT, CPY_BIT_OF_INT_E

Fig. 2-20:
Options - Code Generation for A and FX series

The section **Code Generation** is available for all CPU types.

Generate code for Manual SFC Control

This item is available for Q and QnA CPUs only, except Q00, Q00J, Q01, and Q25SS.
If this option is enabled, a special sequence code is generated which includes one SET and one RST instruction for the steps S0–Sn, where n is the maximum step number of all SFC POUs. The code also includes a BRSET instruction at the beginning which sets the current block according to a system word register.

This code helps to influence the step flags of every SFC block by setting the values for the D and M devices. By the SFC editor only on SFC block can be handled at one time. Using LDP allows to set a step in a running SFC without risking to activate the same step again several times.

The following checks are performed during the manual SFC control operation:

- The consistency is checked between PLC and project.
- The PLC must operate in RUN mode.

Generated sample code:

```
LD>   K-1 D2048
BRSET D2048
LDP   M4098
AND   M4100
SET   S0
LDP   M4099
AND   M4101
RST   S0
   :
LDP   M4098
AND   M4120
SET   S10
LDP   M4099
AND   M4121
RST   S10
   :
```

Time Scheduler

This item is available for Q and QnA CPUs only. Timers in the A series CPU are not executed in the same way as those in the Q/QnA series CPU. If a jump is made over a program part containing a timer, the timer behaviour differs as follows:

MESEC A series	MELSEC QnA series
The timer will still be running and the timer value will be incremented.	The timer will be stopped and the timer value and the status will be kept.

Tab. 2-5: *Timer behaviour of A and QnA series*

If an A series project has been converted to a Q/QnA series project, the described timer behaviour may result in a different execution of the complete project, especially in the following cases:

- use of MELSEC function blocks
  (e.g. TIMER_10_FB, TIMER_CONT_FB, TIMER_HIGH_FB)
- use of IEC function blocks (e.g. TOF, TON_HIGH)
- internal use of timers (e.g. timers controlling tasks)
- direct use of timers and retentive timers (via index qualification as well)

In order to avoid the different timer behaviour the timer scheduler function can be used. The timer scheduler generates a special code which ensures the same timer behaviour in the Q/QnA series CPU as in the A series CPU.

If the timer scheduler is activated, the new code generation causes the following disadvantages:

- The project needs more system variables.
- The scan timer will be longer.
- The code size will be bigger.

The timer scheduler can be activated/deactivated in this dialogue.

Generate code as in GX IEC Developer 5.02

The code generation option is available for all CPU types. For A and FX series all entries are disabled except the compatibility flag. The button **Load Defaults** and **Save As Defaults** will work only on the compatibility and different code warning flag of the Code Generation options for A and FX series.

If the compatibility flag is set the same sequence code is generated as in GX IEC Developer 5.02. For old 32bit projects the compatibility flag is automatically set. For old 16bit projects and for new projects the value of the compatibility flag depends on the default values, which were saved by the user. If no default value is found in the INI file for the compatibility flag, FALSE is taken.

If the different code warning flag is set, a warning is shown for every location where a different code is produced in GX IEC Developer 7.01 compared to GX IEC Developer 5.02. The messages of the warnings are different depending on the flag **Generate code as in GX IEC Developer 5.02**.

For old 32bit projects the different code warning flag is automatically set. For old 16bit projects and for new projects the value of the different code warning flag depends on the default values, which were saved by the user. If no default value is found in the INI file for the different code warning flag, TRUE is taken.

In GX IEC Developer 5.02 a boolean destination of some standard functions and of user functions with EN/ ENO is changed even when the EN is FALSE. When the compatibility flag is not set the boolean destinations keep their values and are not changed.

According to the value of the compatibility flag a different code is produced in GX IEC Developer 7.01 for the standard functions with boolean destinations:

- INT_TO_BOOL_E, WORD_TO_BOOL_E
- DINT_TO_BOOL_E, DWORD_TO_BOOL_E, TIME_TO_BOOL_E
- LIMIT_E
- NOT_E
- EQ_E, NE_E, GT_E, GE_E, LT_E, LE_E
- CPY_BIT_OF_INT, CPY_BIT_OF_INT_E
- AND_E, OR_E, XOR_E
- MAX_E, MIN_E

To get the same code produced in GX IEC Developer 7.01 as in GX IEC Developer 5.02 without setting the compatibility flag, the above standard functions could be replaced with the functions without EN/ENO where the boolean result is connected to an AND block with the former EN as other input. The former destination must then be used as output of the AND.

Show warnings for different coding compared with GX IEC Developer 5.02

When enabled a warning is shown for every location where different code is produced in GX IEC Developer 7.01 compared to GX IEC Developer 5.02.

Use SET/RST for Boolean Outputs

If a SET/RST flag is not set, the generated code uses the OUT instruction to write to the bit output for the according EN/ENO functions in LD/FBD.

Allow SFC reset variable as transition

An SFC POU of an FX project can have a “reset variable”, a Boolean variable which – when activated – resets all steps and sets the initial step. Using this variable as transition has no use, because when the transition’s state gets switched on, the control will get back to the initial step, not to the step or steps following the path of the transition.
(Also see: EMERGENCY OFF for FX Series, page 6-18.)

You are able to choose if such a situation should lead to an error or a warning.

The following warning will be produced by the compiler if an SFC reset variable is used as transition and it is allowed in the Options:

> *C9175 SFC reset variable used as transition.*

The following error will be produced by the compiler if an SFC reset variable is used as transition and it is not allowed in the Options:

> *C9175 SFC reset variable used as transition. This error can be converted to a warning in the Options.*

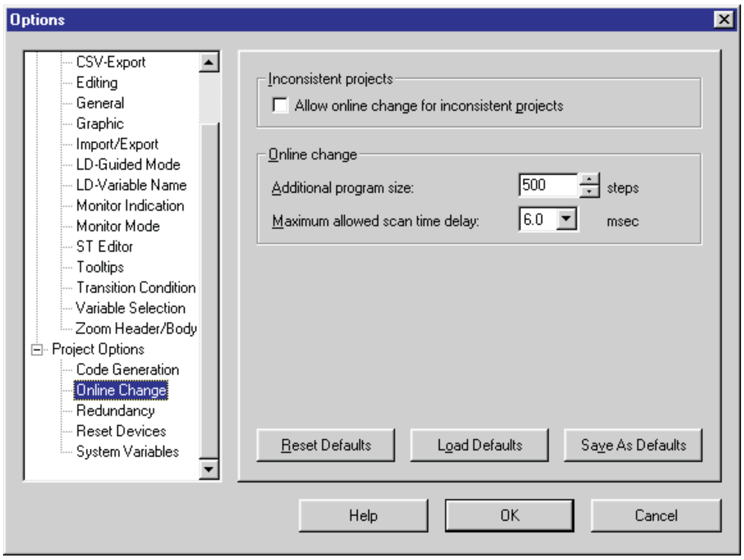
Activate all initial steps in the first scan

This option is available for the FX series only. It allows the control of the activation of the initial step by the user.

The default status is enabled. The compiler will produce a sequence code to activate all initial steps in the first scan. When this option is not checked the sequence code is not produced.

If you want to keep the status of the steps of the SFC POUs for the FX series, you must deactivate the option, latch all step flags of the system variable address range and control the activation of the SFC POUs by their according SFC Reset variables.

2.8.2 Online Change



Inconsistent projects

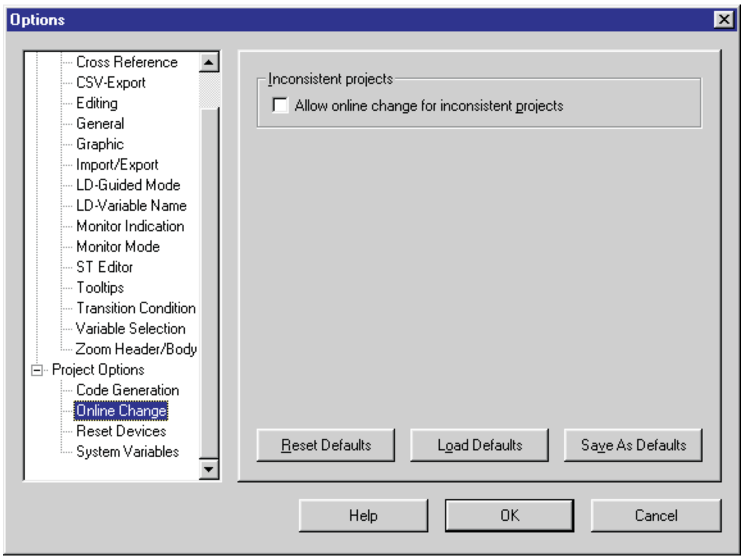
□ Allow online change for inconsistent projects

Online change

Additional program size: 500 steps

Maximum allowed scan time delay: 6.0 msec

Fig. 2-21:
Options - Online Change for QnA and System Q series



Inconsistent projects

□ Allow online change for inconsistent projects

Fig. 2-22:
Options - Online Change for FX series

The section **Online Change** is available for all CPU types except the A series.

Allow online change for inconsistent projects

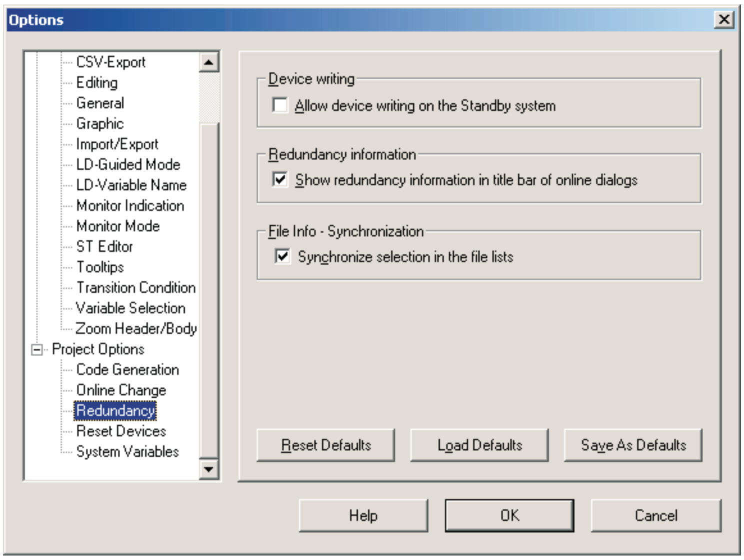
See section 8.7.1 for further details.

Additional program size

This item is available for Q and QnA CPUs only. Due to the file structure of the MELSEC Q and QnA CPUs it is necessary to define an additional program range (in units of steps) for online changes (→ 8.7.2).

After the Additional Program Size has been changed, the value will be checked when leaving the dialogue. This check can be done via the **OK** button or changing the section by clicking on another entry in the **List Options**. In case of an error an error message will appear.

2.8.3 Redundancy



Device writing

☐ Allow device writing on the Standby system

Redundancy information

☑ Show redundancy information in title bar of online dialogs

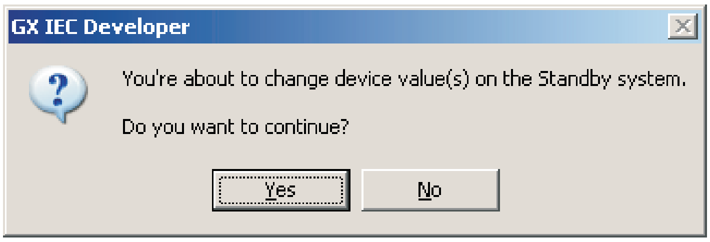
File Info - Synchronization

☑ Synchronize selection in the file lists

Fig. 2-23: *Options - Redundancy for QnPRH PLCs*

The section **Redundancy** is available only for QnPRH CPUs.

Allow device writing on the Standby system: If checked, device writing on the standby system is enabled, and the following confirmation message will be displayed when attempting to write on the standby system.



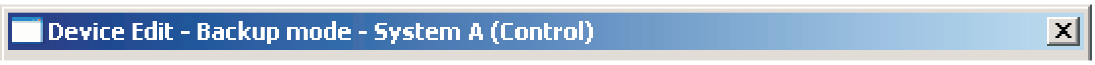
You're about to change device value(s) on the Standby system.

Do you want to continue?

Fig. 2-24: *Confirmation message*

Show redundancy information in title bar of online dialogs: If checked, information about the status of the redundant system will be displayed in the window title for specific functionalities on QnPRH PLCs in the following format:

Original window title - Actual operating mode – Status of connected system



Device Edit - Backup mode - System A (Control)

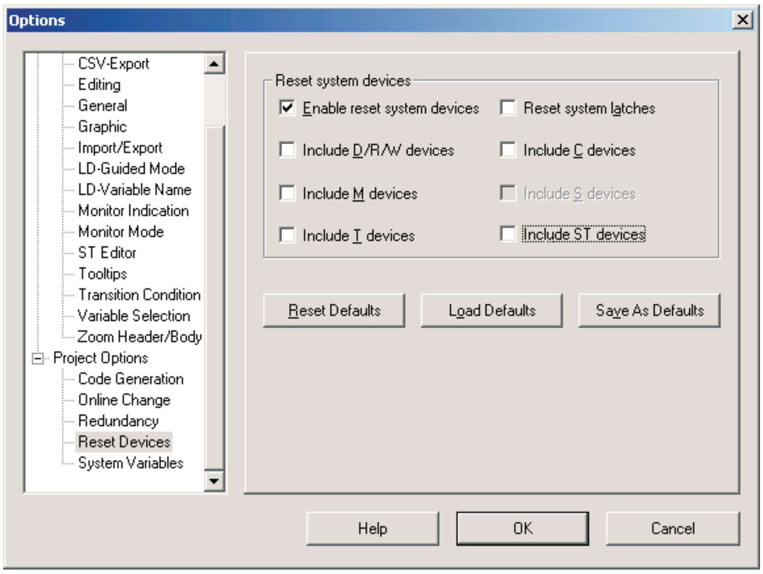
Fig. 2-25: *Title bar with redundancy information*

The following functionalities are included:

- Device Edit
- Buffer Memory Batch Monitor
- File Info
- Format Drive
- Monitoring Indicator
- User Errors
- Hardware Diagnostics
- Scan Time
- Forced I/O Registration/Cancellation

Synchronize file selection if both systems are enabled: This option is relevant if operations on both systems are enabled in the menu **Online - File Info** (→ page 8-92).
The synchronization of selected files concerns the selected files in the file lists of the two systems.
When this option is checked, the selections in the two file lists will be kept synchronized. Any changes in selections in one list will be automatically reflected in the other list (if the selected/unselected file also exists on the other PLC).

2.8.4 Reset Devices



Reset system devices

☑ Enable reset system devices
□ Reset system latches
□ Include D/R/W devices
□ Include C devices
□ Include M devices
□ Include S devices
□ Include T devices
□ Include ST devices

Fig. 2-26:
Options - Reset Devices

If the option **Enable reset system devices** is not selected, all other options are disabled to indicate that the reset devices are not active.

Reset Defaults

When pressing the button **Reset Defaults** the settings of this section are reset to their default values, i.e. the hard coded settings are loaded. This is especially helpful to restore the original settings.

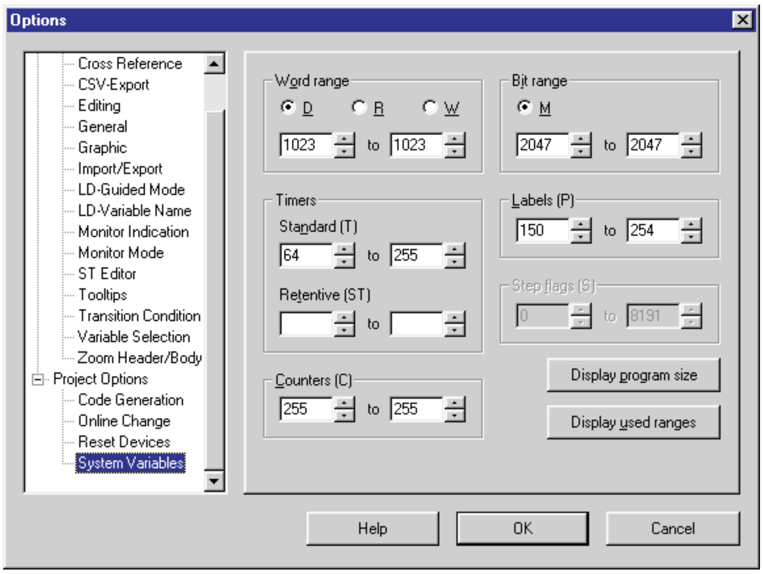
Load Defaults

When pressing the button **Load Defaults** the default settings saved in the **.INI** file are loaded. If the **.INI** file does not include any entries, the hard coded settings are loaded.

Save As Defaults

When pressing the button **Save As Defaults** the current settings are saved as default settings to the **.INI** file. If the **.INI** file does not include any saved default settings yet, they will be created by this operation.

2.8.5 System Variables



Word range: ● D ○ R ○ W
1023 to 1023

Bit range: ● M
2047 to 2047

Timers
Standard (T): 64 to 255
Retentive (ST): to

Labels (P): 150 to 254

Step flags (S): 0 to 8191

Counters (C): 255 to 255

Fig. 2-27:
Options -
System Variables
(for Q/QnA projects)

System variable ranges for the actual project. Available if an Q/QnA project is open.

Word Range

D: D devices are used as word system variables.

R: R devices are used as word system variables.

W: W devices are used as word system variables.

Word range – From/to: PLC type dependant, as defined in the parameters.

Timers

Standard (T) – From/to: PLC type dependant, as defined in the parameters.

Retentive (ST) – From/to: PLC type dependant, as defined in the parameters.

Counters ©)

Counters – From/to: PLC type dependant, as defined in the parameters.

Bit range

M: M devices are used as bit system variables.

Bit range – From/to: PLC type dependant, as defined in the parameters.

Labels (P)

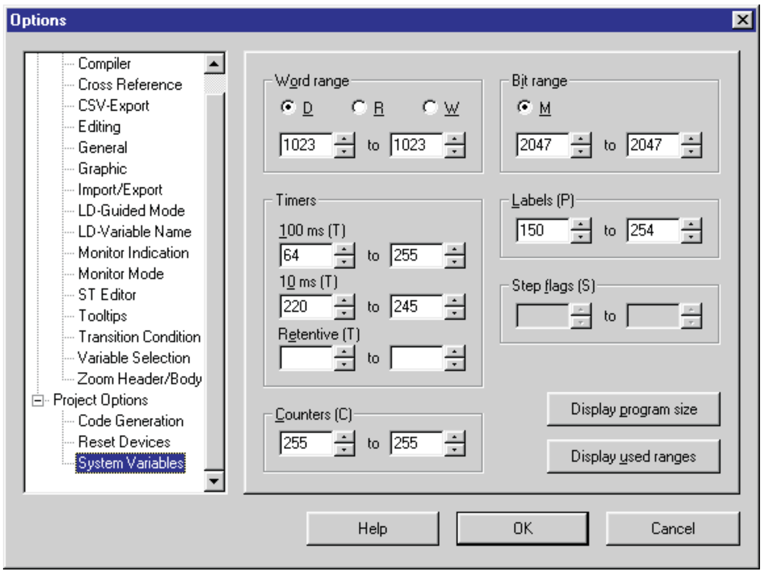
Labels – From/to: PLC type dependant, as defined in the adequate CNF file.

Step flags (S)

Step flags – From/to: PLC type dependant, as defined in the adequate TYP file.

Display Program Size: A summary of the used program size is displayed on a separate dialogue box. If the program is not compiled the dialogue shows a '?' character instead of the program size. If SFC or SUB programs are not available for this CPU, the correspondent line will be grayed.

Display Used Ranges: A summary of the used system variable ranges is displayed on a separate dialogue box.



Word range: ● D ○ R ○ W
1023 to 1023

Bit range: ● M
2047 to 2047

Timers
100 ms (T): 64 to 255
10 ms (T): 220 to 245
Retentive (T): to

Labels (P): 150 to 254

Step flags (S): to

Counters (C): 255 to 255

Fig. 2-28:
Options -
System Variables
(for A/FX projects)

System variable ranges for the actual project. Available if an A/FX project is open.

Word Range

D: D devices are used as word system variables.

R: R devices are used as word system variables.

W: W devices are used as word system variables.

Word range – From/to: PLC type dependant, as defined in the parameters.

Timers

100 ms (T) – From/to: PLC type dependant, as defined in the parameters.

10 ms (T) – From/to: PLC type dependant, as defined in the parameters.

Retentive (ST) – From/to: PLC type dependant, as defined in the parameters.

Counters (C)

Counters – From/to: PLC type dependant, as defined in the parameters.

Bit range

M: M devices are used as bit system variables.

Bit range – From/to: PLC type dependant, as defined in the adequate CNF file.

Labels (P)

Labels – From/to: PLC type dependant, as defined in the adequate CNF file.

Step flags (S) (FX projects only)

Step flags – From/to: PLC type dependant, as defined in the adequate CNF file.

Display Used Ranges: A summary of the used system variable ranges is displayed on a separate dialogue box.

2.9 Connecting the programmable logic controller (PLC) to the computer

In the Project Navigator click on **PLC_Parameter** and in the dialogue box on the button **CPU Type**. You get a list containing all CPU types which are supported by GX IEC Developer. A detailed summary of all CPU types supported by GX IEC Developer is included in section 5.16.

This manual does *not* include instructions for installing your controller hardware. Each controller system is supplied with documentation including comprehensive installation instructions.

There are a number of different options for connecting the PLC to your personal computer:

- Via the CPU interface
- Via the computer link modules
- Via an Ethernet module
- Via the USB interface

The transparent mode enables data transmission by modem between the personal computer, terminal and the PLC.

The options for selecting the connection type and setting the corresponding parameters can be found in the **Online** menu under **Transfer Setup - Ports**
(→ Chapter 8).

The following figure on the next page shows all supported options for connecting the PLC to your personal computer.

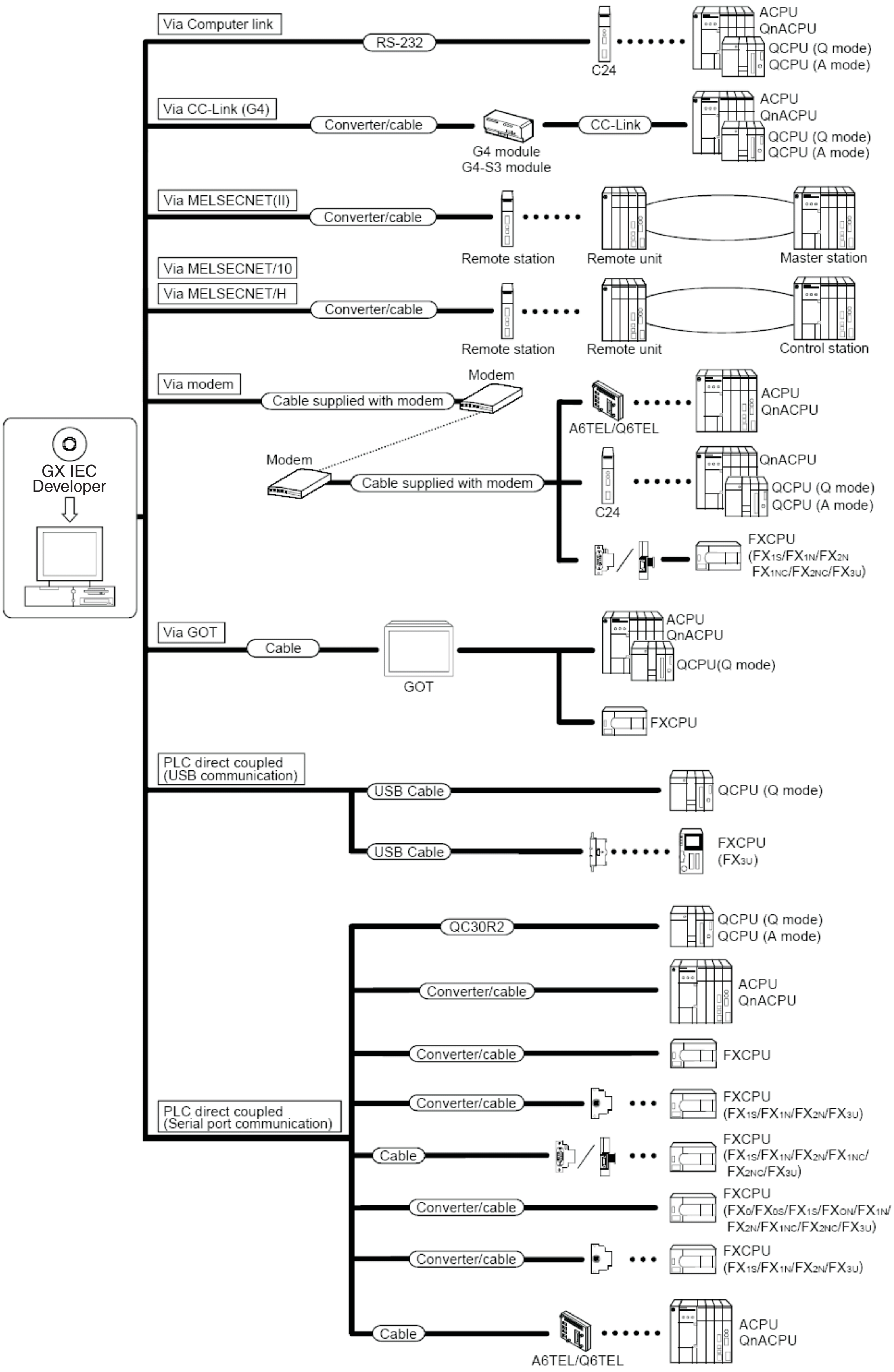


Fig. 2-29: *System configuration*

2.9.1 Connecting FX/FX2/FXU/FX2U/FX2C series controllers

The controllers of the FX series can only be connected to the computer via the CPU interface.

a) Connection via the CPU interface with the SC-09 interface converter

Connect the 9-pin D-shell connector to the computer. If your computer port has a 25-pin connector you will also need a 9/25-pin adapter. Connect the 25-pin D-shell connector to the FX controller.

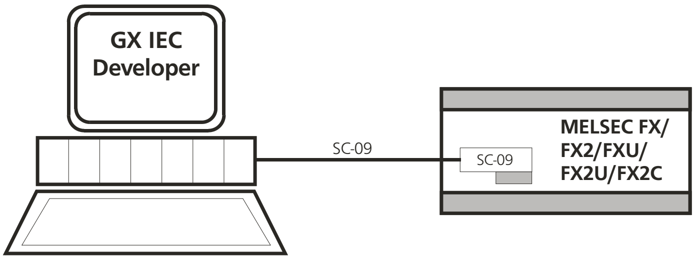


Fig. 2-30:
CPU interface connection

b) Connection via the CPU interface with the FX-232AW serial interface

You will need two additional cables for connecting the FX-232AW module:

- FX-422CAB for connecting the FX controller to the FX-232AW
- F2-232CAB or FX-232CAB for connecting the FX-232AW to the personal computer

If your computer port has a 9-pin connector you will also need a 25/9-pin adapter as both of these cables (F2-232CAB, FX-232CAB) are fitted with 25-pin D-shell connectors at the PC end.

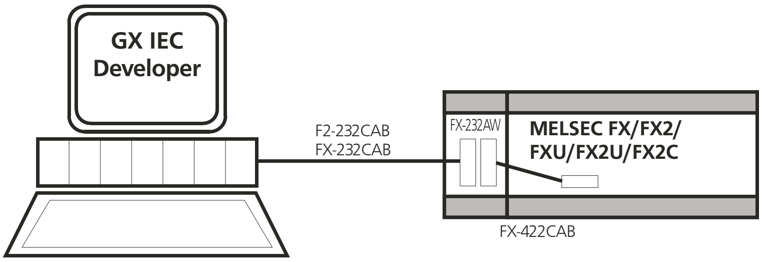


Fig. 2-31:
CPU interface connection

2.9.2 Connecting FX0/FX0N/FX0S/FX1S/FX1N/FX2N/FX2NC series controllers

The controllers of the FX series can only be connected to the computer via the CPU interface.

a) Connection via the CPU interface with the SC-09 interface converter

Connect the 9-pin D-shell connector to the computer. If your computer port has a 25-pin connector you will also need a 9/25-pin adapter. Connect the 8-pin min-DIN connector to the FX controller. This connector is delivered together with the SC-09.

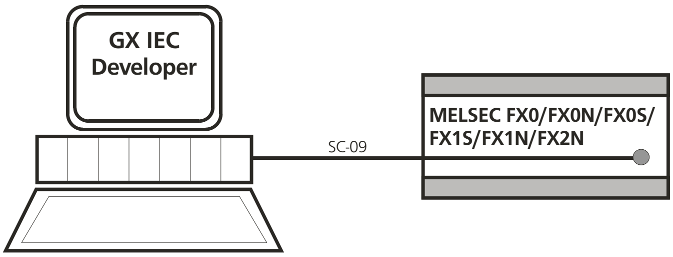


Fig. 2-32:
CPU interface connection

b) Connection via the CPU interface with the FX-232AW serial interface

You need three additional cables for connecting the systems with the FX-232AW module:

- The FX-422CAB and the FX-20PCADP cables for connecting the FX controller to the FX-232AW
- The F2-232CAB or FX-232CAB cable for connecting the FX-232AW to the personal computer

If your computer port has a 9-pin connector you will also need a 25/9-pin adapter, as both these cables (F2-232CAB, FX-232CAB) have 25-pin D-shell connectors at the computer end.

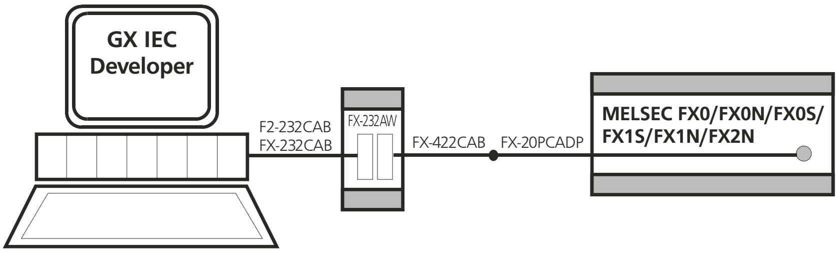


Fig. 2-33:
CPU interface connection

2.9.3 Connecting FX3U series controllers

a) Connection of the CPU interface via the FX-232AWC-H

You can use two additional cables for connecting the CPU and the PC:

- The FX-422CAB0 cable for connecting the FX3U controller to the FX-232AWC-H

  The FX-422CAB0 cable can be connected to the FX3U via the programming port (PP Port) or via the extension board FX3U-422-BD.

- The F2-232CAB-1 cable for connecting the FX-232AWC-H to the personal computer

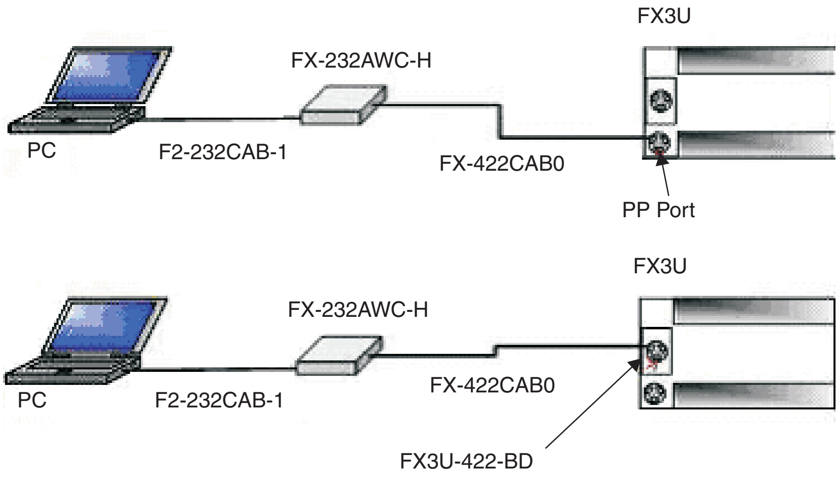


Fig. 2-34:
CPU interface connection

b) Connection of the CPU interface via the FX-232CAB-1 cable

You can use the FX-232CAB-1 cable for connecting the CPU and the PC:

- The FX-232CAB-1 cable can be connected to the FX3U via the extension board FX3U-232-BD.

- The FX-232CAB-1 cable can be connected to the FX3U via the special function module FX3U-232-ADP.

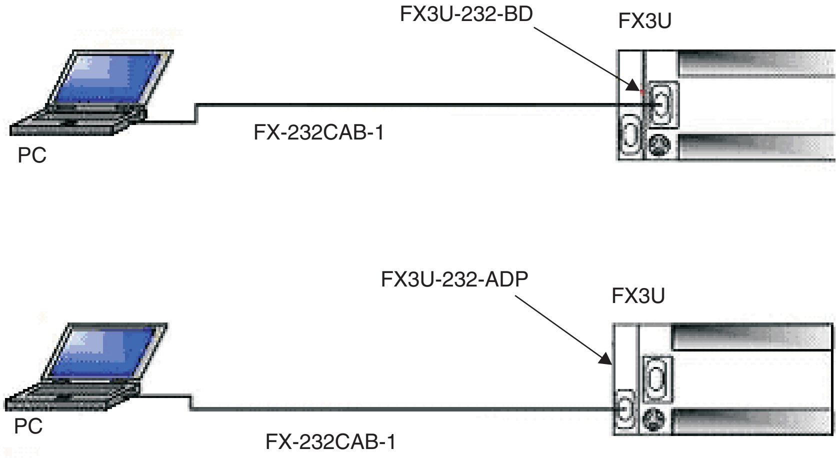


Fig. 2-35:
CPU interface connection

c) Connection of the CPU interface via an USB cable

You can use an USB cable for connecting the CPU and the PC:

- The USB cable can be connected to the FX3U via the extension board FX3U-USB-BD.
- The USB cable can be connected to the FX3U via the connection adapter FX-USB-AW.

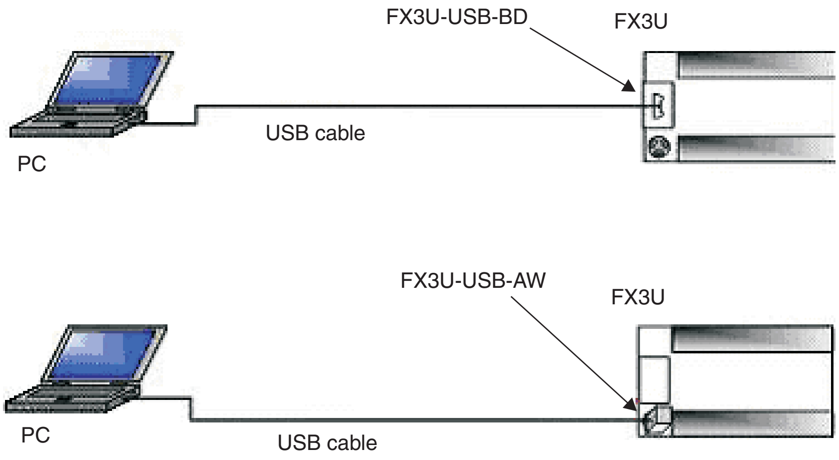


Fig. 2-36:
CPU connection via USB cable

d) Connection of the CPU interface via modem

You can use modems for connecting the CPU and the PC:

- The CPU side modem can be connected to the FX3U via the extension board FX3U-232-BD.
- The CPU side modem can be connected to the FX3U via the special function module FX3U-232-ADP.

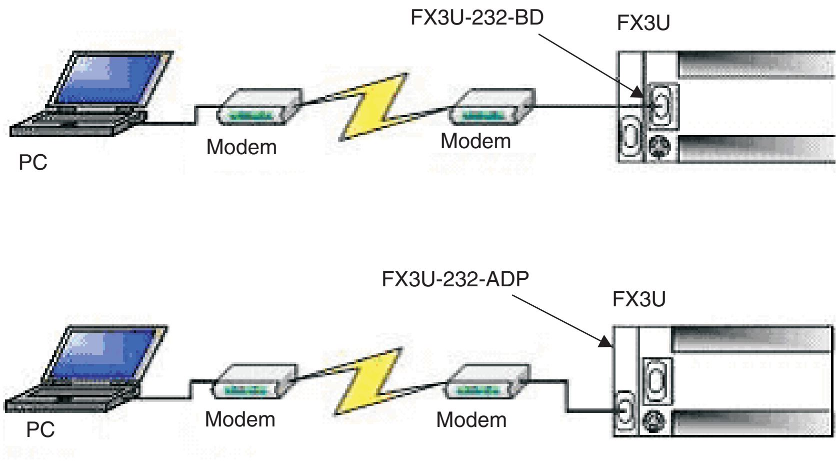


Fig. 2-37:
CPU connection via modem

e) Connection of the CPU interface via HMI module

You can use HMI modules for connecting the CPU and the PC:

- The FX3U can be connected to the PC via a GOT-F900.

  The CPU is connected to the GOT-F900 by the FX-50DU-CAB0 cable via the programming port (PP Port).

  The PC is connected to the GOT-F900 by the FX-232CAB-1 cable.

- The FX3U can be connected to the PC via a GOT-1000.

  The CPU is connected to the GOT-1000 by the GT01-C□R4-8P cable via the programming port (PP Port).

  The PC is connected to the GOT-1000 by the GT01-C30R2-9S cable or the GT08-C20USB-5P cable.

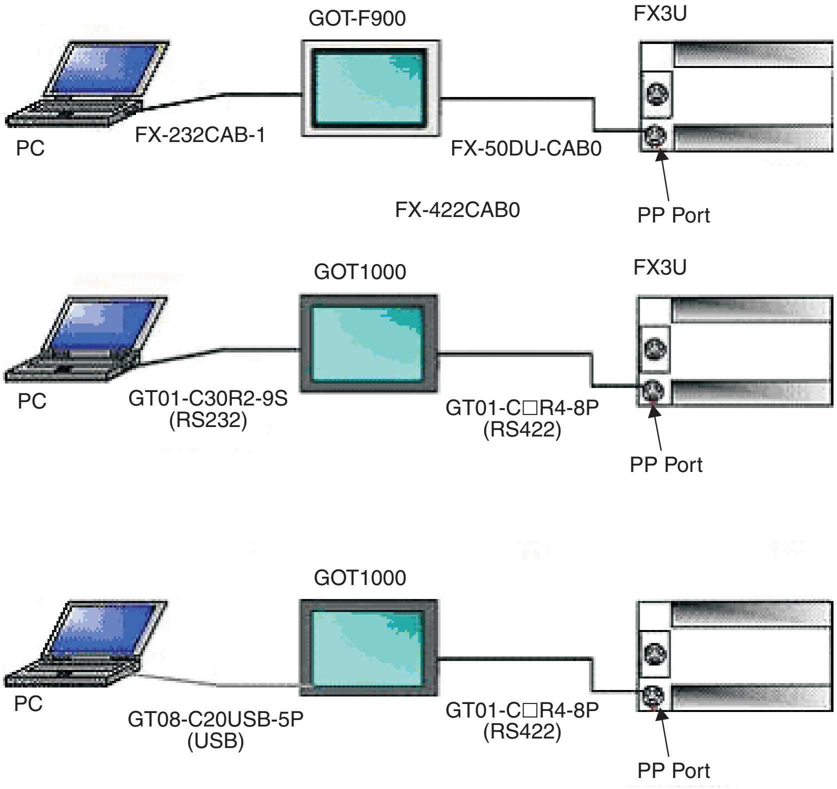


Fig. 2-38:
CPU connection via HMI module

f) Connection of the CPU interface via the ETHERNET

You can connect the CPU to the PC via the ETHERNET:

- The FX3U can be connected to the ETHERNET via the special function module FX3U-ENET.

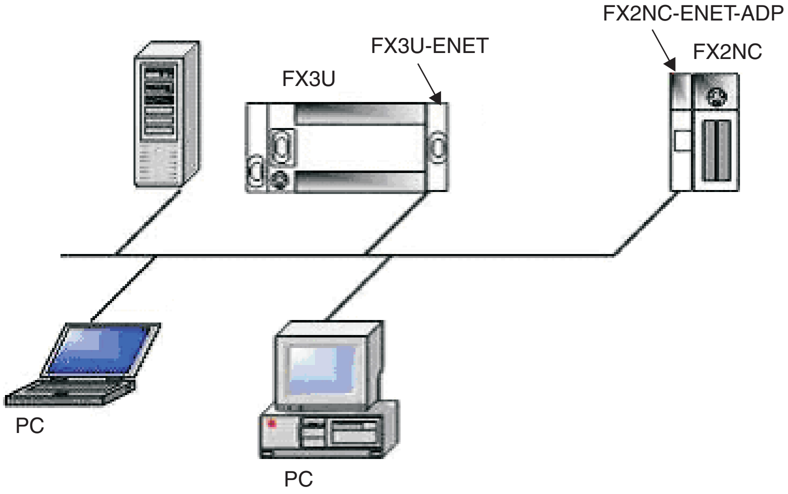


Fig. 2-39: *CPU connection via ETHERNET*

2.9.4 Connecting A/QnA series controllers

The A series controllers can be connected to the computer via either the CPU interface, computer link module, the A7BDE interface card or the Ethernet module.

a) Connection via the CPU interface with the SC-09 interface converter

Connect the 9-pin D-shell connector of the SC-09 to the computer. If your computer's port has a 25-pin connector you will also need a 25/9-pin adapter. Connect the 25-pin D-shell connector to the CPU port.

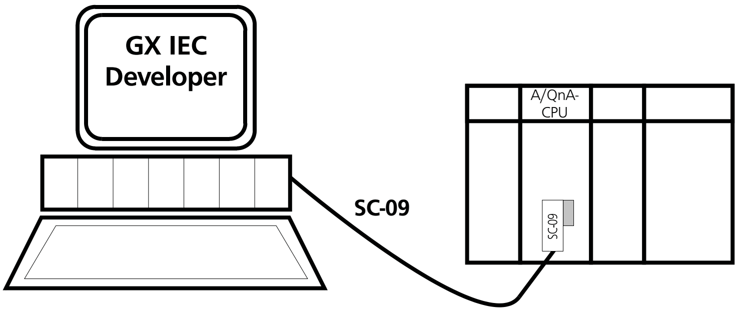


Fig. 2-40:
CPU interface connection

b) Connection via the computer link module

The following computer link modules are available: AJ71UC24, A1SJ71UC24-R2, AJ71QC24(-R), AJ71QC24N(-R2), A1SJ71QC24(-R2), A1SJ71QC24N(-R2). The computer link module is installed in the base unit (module rack). Connect the RS-232 interface of your computer to the RS-232 port of the computer link module via the interface cable. Check the pin assignments of the cable before using it (see below). The communications parameters you set in **Online - Transfer Setup - Ports** (→ Section 8.3.1) must match the DIP switch settings on the computer link modules.

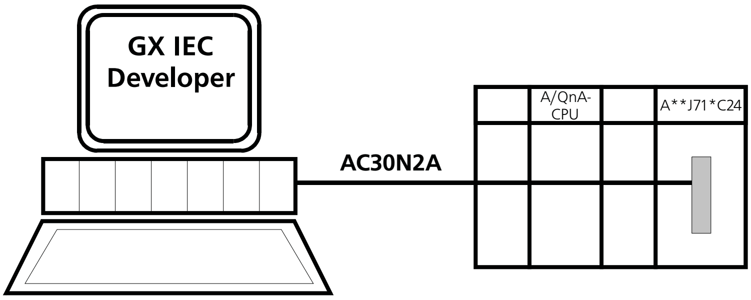


Fig. 2-41:
Computer link connection

NOTE See the computer link module manuals for full details on their settings.

Pin assignments of the RS-232 interface cable

The pin assignments of 25-pin connectors **with** applicable CD signal on pin 8 are as follows:

Computer Link Module		Cable Connection and Signal Direction (Full-/Half-Duplex)	GX IEC Developer	
Signal	Pin		Pin	Signal
FG	1	←→	1	FG
SD (TXD)	2	→	3	RD (RXD)
RD (RXD)	3	←	2	SD (TXD)
RS	4	↓		
CS (CTS)	5	←→	8	CD
DSR (DR)	6	←	20	DTR (ER)
SG	7	←→	7	SG
		↓	4	RS
CD	8	←→	5	CS (CTS)
DTR (ER)	20	→	6	DSR (DR)

Tab. 2-6: *Pin assignments of the RS-232 interface cable (25-pin, with CD signal)*

DC code control or DTR/DSR control is enabled by connecting the computer link module to an external device as shown above.

The pin assignments of 25-pin connectors **without** applicable CD signal on pin 8 are as follows:

Computer Link Module		Cable Connection and Signal Direction (Full-Duplex)	GX IEC Developer	
Signal	Pin		Pin	Signal
FG	1	←→	1	FG
SD (TXD)	2	→	3	RD (RXD)
RD (RXD)	3	←	2	SD (TXD)
RS	4	↓ ↓	4	RS
CS (CTS)	5		5	CS (CTS)
DSR (DR)	6	←	20	DTR (ER)
SG	7	←→	7	SG
CD	8		8	CD
DTR (ER)	20	→	6	DSR (DR)

Tab. 2-7: *Pin assignments of the RS-232 interface cable (25-pin, no CD signal)*

DC code control or DTR/DSR control is enabled by connecting the computer link module to an external device as shown above.

The pin assignments of 9-pin connectors **with** applicable CD signal on pin 1 are as follows:

Computer Link Module		Cable Connection and Signal Direction (Full-/Half-Duplex)	GX IEC Developer	
Signal	Pin		Pin	Signal
		⊤	7	RS (RTS)
CD	1	←→ ↓	8	CS (CTS)
RD (RXD)	2	←	3	SD (TXD)
SD (TXD)	3	→	2	RD (RXD)
DTR (ER)	4	→	6	DSR (DR)
SG	5	←→	5	SG
DSR (DR)	6	←	4	DTR (ER)
RS (RTS)	7	⊤		
CS (CTS)	8	↓ ←→	1	CD

Tab. 2-8: *Pin assignments of the RS-232 interface cable (9-pin, with CD signal)*

DC code control or DTR/DSR control is enabled by connecting the computer link module to an external device as shown above.

The pin assignments of 9-pin connectors **without** applicable CD signal on pin 1 are as follows:

Computer Link Module		Cable Connection and Signal Direction (Full-Duplex)	GX IEC Developer	
Signal	Pin		Pin	Signal
CD	1		1	CD
RD (RXD)	2	←	3	SD (TXD)
SD (TXD)	3	→	2	RD (RXD)
DTR (ER)	4	→	6	DSR (DR)
SG	5	←→	5	SG
DSR (DR)	6	←	4	DTR (ER)
RS	7	⊤ ⊤	7	RS
CS (CTS)	8	↓ ↓	8	CS (CTS)

Tab. 2-9: *Pin assignments of the RS-232 interface cable (9-pin, no CD signal)*

DC code control or DTR/DSR control is enabled by connecting the computer link module to an external device as shown above.

DIP switch settings

The communications parameters you set in **Online - Transfer Setup - Ports** (→ Section 8.3.1) must match the DIP switch settings on the computer link modules.

NOTE

Please see the manual of the computer link module for details on the DIP switch settings.

c) Connection via the Ethernet module

GX IEC Developer enables a MELSEC A series' PLC to be programmed via the Ethernet. The Ethernet module is installed in the base unit. The connection between the Personal Computer and the Ethernet module is realized via the coaxial cable 10BASE5 (Ethernet) or 10BASE2 (Cheapernet).

The following Ethernet modules can be used:

- AJ71E71-S3
- AJ71QE71
- A1SJ71E71-S3
- A1SJ71QE71
- AJ71E71N-B2
- AJ71QE71N-B2
- A1SJ71E71N-B2
- A1SJ71QE71N-B2
- AJ71E71N-B5T
- AJ71QE71N-B5T
- A1SJ71E71N-B5T
- A1SJ71QE71N-B5T

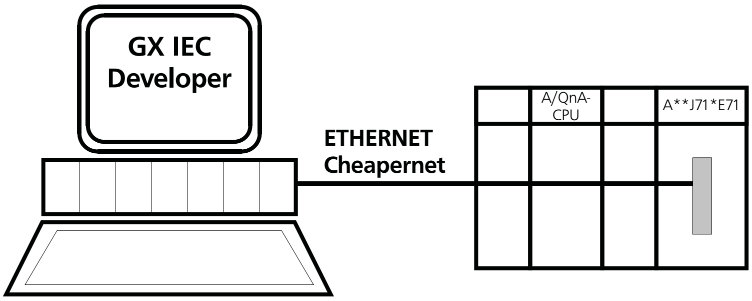


Fig. 2-42:
Ethernet connection

The communications parameters you set in **Online - Transfer Setup - Ports** (→ Section 8.3.1) must match the DIP switch settings on the Ethernet module.

NOTES

See Ethernet manuals for full details on Ethernet module settings.

The Ethernet modules will only work together properly with GX IEC Developer, if its version “E” (or higher) is used.

You will find the version number directly on the Ethernet modules.

When the AnU CPU type has been started on the personal computer, change the PLC type to the AnA CPU.
Communications can only be performed within the same segment.
Communications can not be performed via the router or gateway.

2.9.5 Connecting Q series controllers

The Q series controllers can be connected to the computer via either the CPU interface (RS-232 or USB), computer link module or the Ethernet module.

a) Connection via the CPU interface with the QC30R2 interface cable

Connect the RS-232 interface of your computer to the RS-232 port of the CPU via the interface cable QC30R2. If your computer's port has a 25-pin connector you will also need a 25/9-pin adapter.

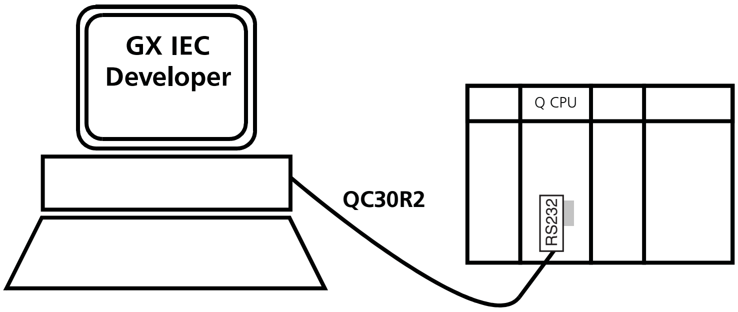


Fig. 2-43:
CPU interface connection

b) Connection via the computer link module

The computer link module QJ71C24(-R2) is installed in the base unit (module rack). Connect the RS-232 interface of your computer to the RS-232 port of the computer link module via the interface cable AC30N2A. Check the pin assignments of the cable before using it (see below). The communications parameters you set in **Online - Transfer Setup - Ports** (→ Section 8.3.1) must match the DIP switch settings on the computer link modules.

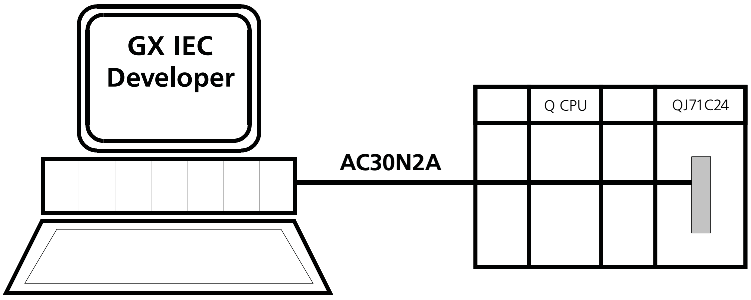


Fig. 2-44:
Computer link connection

NOTE Please see the manual of the computer link module for details on the DIP switch settings.

Pin assignments of the RS-232 interface cable

The connector specifications are indicated below:

Pin	Signal	Description	Signal Direction, Q-compatible C24 <-> External Device
1	CD	Receive carrier detection	←
2	RD (RXD)	Receive data	←
3	SD (TXD)	Send data	→
4	DTR (ER)	Data terminal ready	→
5	SG	Send ground	↔
6	DSR (DR)	Data set ready	←
7	RS	Request to send	→
8	CS (CTS)	Clear to send	←
9	RI (CI)	Call indication	←

Tab. 2-10: *Pin assignments of the RS-232 interface cable*

The pin assignments of 9-pin connectors **with** applicable CD signal on pin 1 are as follows:

Computer Link Module		Cable Connection and Signal Direction (Full-/Half-Duplex)	GX IEC Developer	
Signal	Pin		Pin	Signal
		↓	7	RS (RTS)
CD	1	↔	8	CS (CTS)
RD (RXD)	2	←	3	SD (TXD)
SD (TXD)	3	→	2	RD (RXD)
DTR (ER)	4	→	6	DSR (DR)
SG	5	↔	5	SG
DSR (DR)	6	←	4	DTR (ER)
RS (RTS)	7	↓		
CS (CTS)	8	↔	1	CD

Tab. 2-11: *Pin assignments of the RS-232 interface cable (9-pin, with CD signal)*

DC code control or DTR/DSR control is enabled by connecting the computer link module to an external device as shown above.

The pin assignments of 9-pin connectors **without** applicable CD signal on pin 1 are as follows:

Computer Link Module		Cable Connection and Signal Direction (Full-Duplex)	GX IEC Developer	
Signal	**Pin**		**Pin**	**Signal**
CD	1		1	CD
RD (RXD)	2	←	3	SD (TXD)
SD (TXD)	3	→	2	RD (RXD)
DTR (ER)	4	→	6	DSR (DR)
SG	5	↔	5	SG
DSR (DR)	6	←	4	DTR (ER)
RS	7	┬ ┬	7	RS
CS (CTS)	8	↓ ↓	8	CS (CTS)

Tab. 2-12: *Pin assignments of the RS-232 interface cable (9-pin, no CD signal)*

DC code control or DTR/DSR control is enabled by connecting the computer link module to an external device as shown above.

DIP switch settings

The communications parameters you set in **Online - Transfer Setup - Ports** (→ Section 8.3.1) must match the DIP switch settings on the computer link modules.

NOTE Please see the computer link module manual for details on the DIP switch settings.

c) Connection via the Ethernet module

GX IEC Developer enables a MELSEC Q series' PLC to be programmed via the Ethernet. The Ethernet module QJ71E71(-B2) is installed in the base unit. The connection between the Personal Computer and the Ethernet module is realized via the coaxial cable 10BASE5 (Ethernet) or 10BASE2 (Cheapernet).

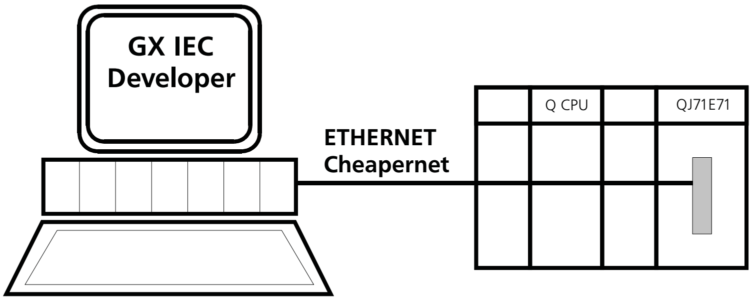


Fig. 2-45:
Ethernet connection

NOTE See Ethernet manuals for full details on Ethernet module settings.

The communications parameters you set in **Online - Transfer Setup - Ports** (→ Section 8.3.1) must match the DIP switch settings on the Ethernet module.

d) Connection via the CPU interface with a QC30-USB cable

The USB port of the computer is connected to the USB port (not on Q00(J), Q01, Q02 CPUs) of the CPU via a QC30-USB connection cable.

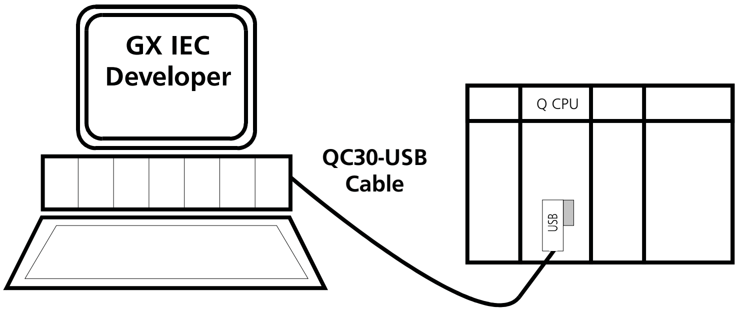


Fig. 2-46:
CPU interface connection

NOTES

Connecting/disconnecting the USB cable, resetting the PLC, or switching power OFF/ON frequently during communications with the PLC can cause a communications error, from which recovery may not be made.
Hence, place GX IEC Developer in an offline status where possible before connecting/disconnecting the USB cable, resetting the PLC, or switching power OFF/ON.
If recovery cannot be made from the communications error, completely disconnect the USB cable once and reconnect it after more than five seconds. (Even after this operation, an error may occur at the initial communication, but communications will be made properly at the second time and later.)
Offline status: Indicates the status other than PLC write/read, monitor, test and PLC diagnostic.

When the PLC CPU is reset while communicating with the PLC CPU via USB connection in the Windows® 2000 environment the message **"Unsafe Removal of Device"** is displayed. In this case restart the communication with the PLC CPU after resetting the PLC CPU.

2.10 Supported utility packages for the Q series

The following utility packages for the Q series are available:

- SWnD5C-QADU-E for A/D Conversion Modules
- SWnD5C-QCTU-E for High Speed Counter Modules
- SWnD5C-QDAU-E for D/A Conversion Modules
- SWnD5C-QSCU-E for Q Corresponding Serial Communication Modules
- SWnD5C-QTCU-E for Temperature Control Modules
- SWnD5C-QDNU-E for DeviceNet Master/Slave Modules
- SWnD5C-QTIU-E for Thermocouple Input Modules
- SWnD5C-QPTU-E for Positioning Control Modules
- SWnD5C-QMBU-E for MODBUS®/TCP Interface Modules
- SWnD5C-QASU-E for AS-i Master Modules
- SWnD5C-QFLU-E for FL-net (OPCN-2) Interface Modules

2.11 Connecting a printer

Install your printer, following the manufacturer's instructions.

The Windows User's Guide contains detailed instructions on installing printers, printer drivers, additional fonts, the print options and using the Windows Print Manager.

In GX IEC Developer, you can access the printer setup options by selecting **Printer Setup** in the **Project** menu.

2.12 Connection between PC and PLC via Modem

The connection between a PC and a PLC via telephone network using GSM and/or analogue modems can be set up as following:

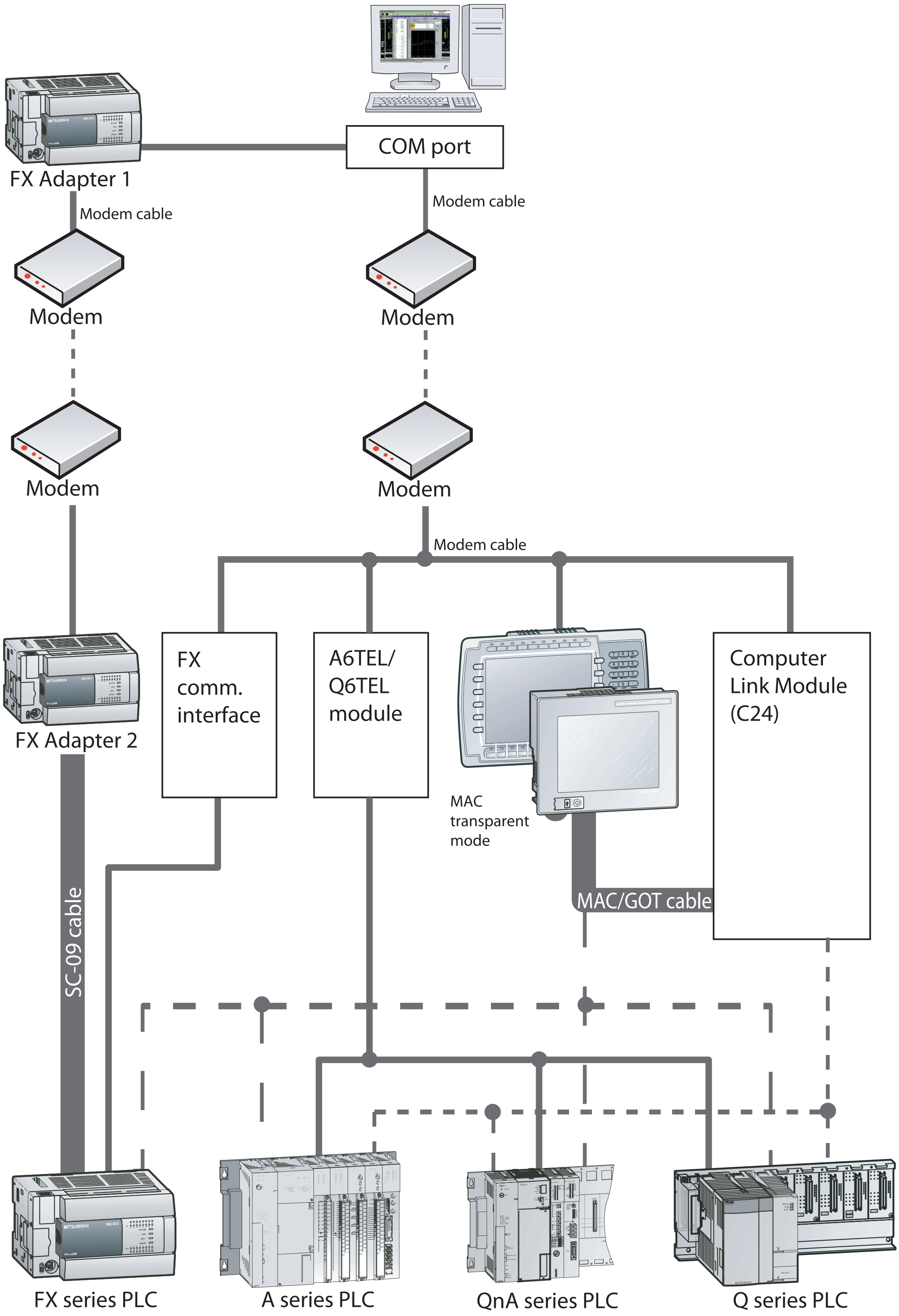


Fig. 2-47: Connection between PC and PLC via modem

2.13 Installing the USB Driver

When Windows® 2000 Professional, Windows® XP Professional or Windows® XP Home Edition is used, the USB driver must be installed to make CPU USB communication. The following is the USB driver installation procedure.

NOTE

If the USB driver cannot be installed, check the following setting.

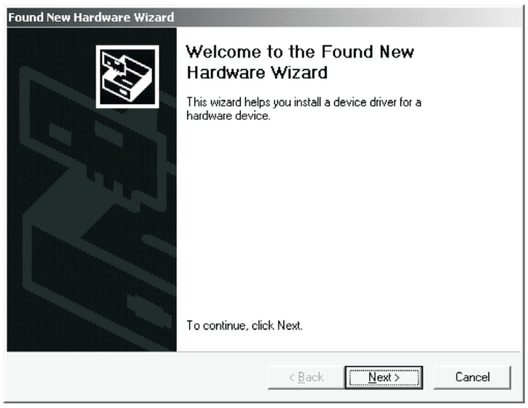
When Windows® 2000 Professional is used
If you have selected "Block-Prevent installation of unsigned files" after [ControlPanel] - [System] - [Hardware] - [Driver Signing], the USB driver may not be installed. Choose "Ignore-Install all files, regardless of file signature" or "Warn-Display a message before installing an unsigned file" for [Driver Signing], and install the USB driver.

When Windows® XP Professional or Windows® XP Home Edition is used
If you have selected "Block-Never install unsigned driver software" after [ControlPanel] - [System] - [Hardware] - [Driver Signing], the USB driver may not be installed. Choose "Ignore-Install the software anyway and don't ask for my approval" or "Warn-Prompt me each time to choose an action" for [Driver Signing], and install the USB driver.

2.13.1 When Windows® 2000 Professional is used

The following indicates the procedure for installing the USB driver when using Windows® 2000 Professional.

① The screen shown below appears when you connect the personal computer and Q CPU (Q mode) by the USB cable. Click the button **Next**.



Found New Hardware Wizard

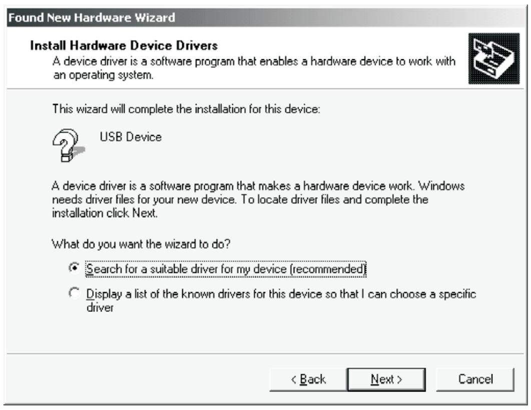
Welcome to the Found New Hardware Wizard

This wizard helps you install a device driver for a hardware device.

To continue, click Next.

Fig. 2-48:
Welcome to the Found New Hardware Wizard

② Choose "Search for a suitable driver for my device (recommended)" and click the button **Next**.



Found New Hardware Wizard

Install Hardware Device Drivers
A device driver is a software program that enables a hardware device to work with an operating system.

This wizard will complete the installation for this device:

USB Device

A device driver is a software program that makes a hardware device work. Windows needs driver files for your new device. To locate driver files and complete the installation click Next.

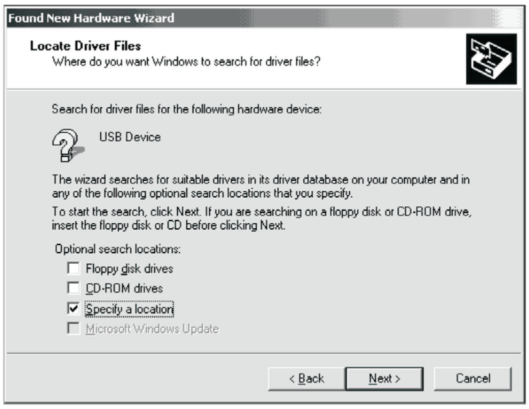
What do you want the wizard to do?

● Search for a suitable driver for my device (recommended)

○ Display a list of the known drivers for this device so that I can choose a specific driver

Fig. 2-49:
Install Hardware Device Drivers

③ Check **Specify a location** and click the button **Next**.



Found New Hardware Wizard

Locate Driver Files

Where do you want Windows to search for driver files?

Search for driver files for the following hardware device:

USB Device

The wizard searches for suitable drivers in its driver database on your computer and in any of the following optional search locations that you specify.

To start the search, click Next. If you are searching on a floppy disk or CD-ROM drive, insert the floppy disk or CD before clicking Next.

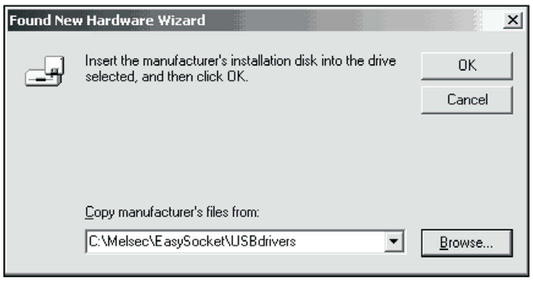
Optional search locations:

- □ Floppy disk drives
- □ CD-ROM drives
- ☑ Specify a location
- □ Microsoft Windows Update

Fig. 2-50:
Locate Driver Files

④ As the screen below appears, set the product installation destination "Easysocket\USBdrivers" and click the button **Next**.

The screen shown below shows the example of setting C:\MELSEC\Easysocket\USBdrivers.



Found New Hardware Wizard

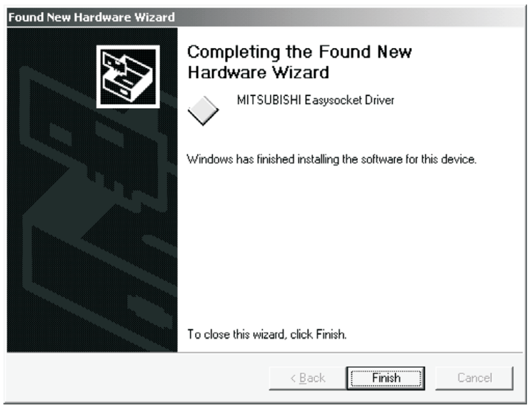
Insert the manufacturer's installation disk into the drive selected, and then click OK.

Copy manufacturer's files from:

C:\Melsec\EasySocket\USBdrivers

Fig. 2-51:
Copy manufacturer's files

⑤ The screen below appears to indicate completion of installation. Click the button **Finish** to terminate installation.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

MITSUBISHI Easysocket Driver

Windows has finished installing the software for this device.

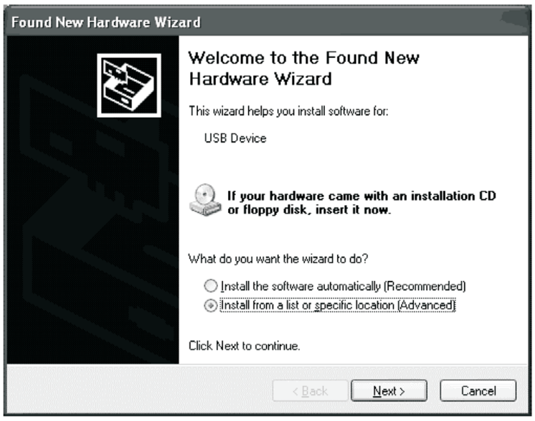
To close this wizard, click Finish.

Fig. 2-52:
Completing the Found New Hardware Wizard

2.13.2 When Windows® XP Professional and Windows® XP Home Edition is used

The following indicates the procedure for installing the USB driver when using Windows® XP Professional or Windows® XP Home Edition.

① The screen below appears when you connect the personal computer and Q CPU (Q mode) by the USB cable.
Choose "Install from a list or specific location [Advanced]" and click the button **Next**.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

USB Device

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)

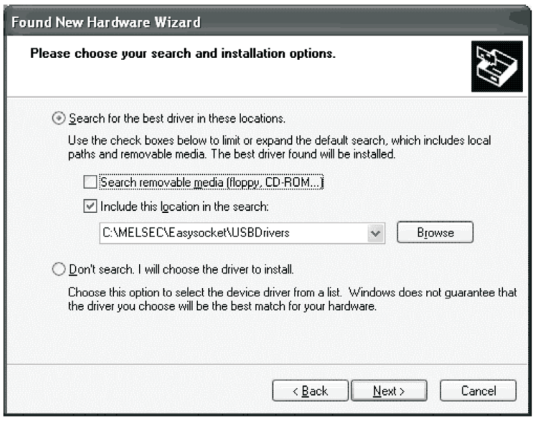
● Install from a list or specific location (Advanced)

Click Next to continue.

Fig. 2-53:
Welcome to the Found New Hardware Wizard

② As the screen below appears, choose "Include this location in the search".
Check "Include this location in the search" and set "Easysocket\USBDrivers" of the folder where GX IEC Developer was installed.

After setting, click the button **Next**. The screen shown below shows the example of setting
C: \MELSEC\Easysocket\USBDrivers.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

C:\MELSEC\Easysocket\USBDrivers

○ Don't search. I will choose the driver to install.

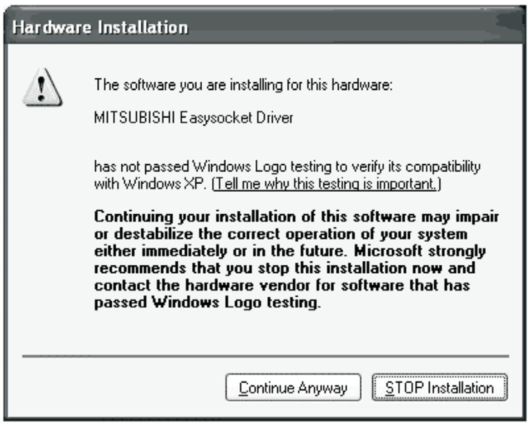
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Fig. 2-54:
Search and installation options

NOTE

If you use Windows® XP Professional and Windows® XP Home Edition with ServicePack 2, please select the option “Don’t search. I will choose the driver to install.” in the dialogue window shown above. The automatic search does not work with ServicePack 2.
Then select the USB driver from your installation root. If this is i.e. drive C, please select: “C: \MELSEC\Easysocket\USBDrivers\ECUsbd.inf”

③ As the screen below appears, click the button **Continue Anyway**.



Hardware Installation

The software you are installing for this hardware:

MITSUBISHI Easysocket Driver

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

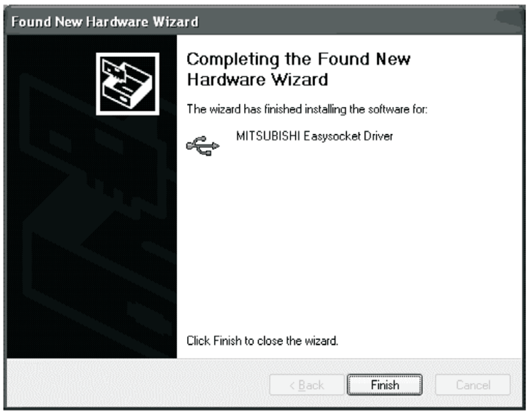
Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Fig. 2-55:
Hardware installation warning
(Please see note on the following page.)

NOTE

Though the screen above appears during installation of the USB driver, we have confirmed that the USB driver operates properly using Windows® XP Professional or Windows® XP Home Edition. (No problem will occur after installation of the USB driver.)
Click **Continue Anyway** to continue the installation of the USB driver.

④ The screen below appears to indicate completion of installation.
Click the button **Finish** to terminate installation.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

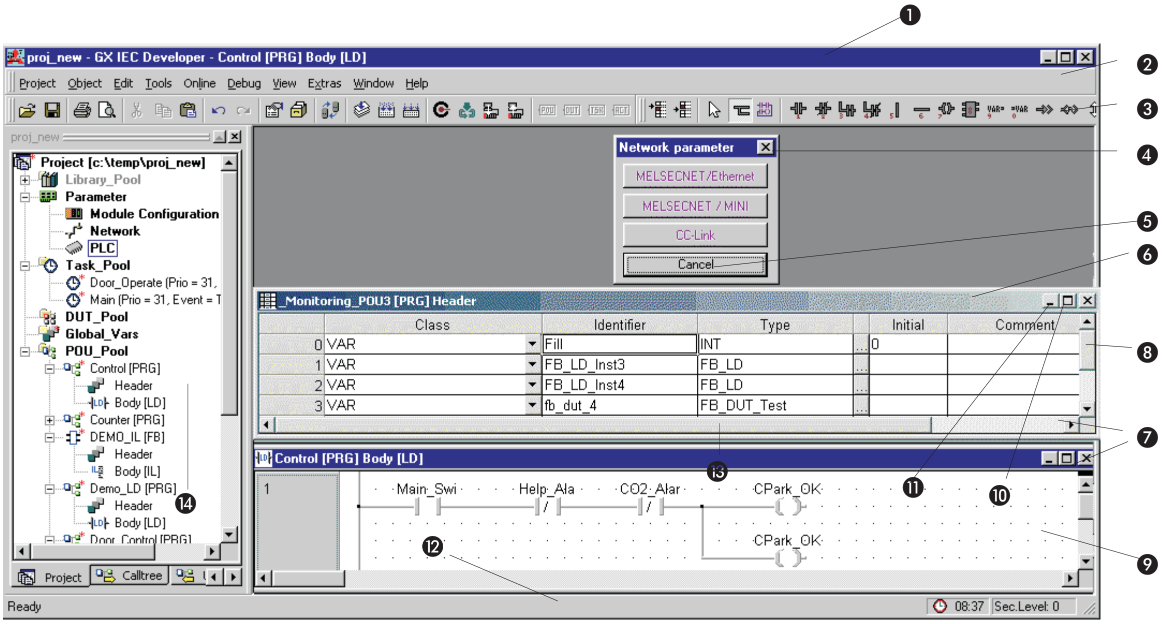
MITSUBISHI Easysocket Driver

Click Finish to close the wizard.

Fig. 2-56:
Completing the Found New Hardware Wizard

3 The User Interface

The illustration below shows the screen after loading a project. In this example, the PLC configuration, a header and a body have all been opened in separate windows.



	Class	Identifier	Type	Initial	Comment
0	VAR	Fill	INT	0	
1	VAR	FB_LD_Inst3	FB_LD		
2	VAR	FB_LD_Inst4	FB_LD		
3	VAR	fb_dut_4	FB_DUT_Test		

Fig. 3-1: *User Interface*

Item	Description
❶	Application title bar
❷	Menu bar
❸	Toolbar
❹	Dialogue box
❺	Button
❻	Declaration table
❼	Object window
❽	Vertical scrollbar
❾	Editor (Body)
❿	Maximise button
⓫	Minimise button
⓬	Status bar
⓭	Horizontal scrollbar
⓮	Project Navigator window

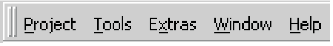
Tab. 3-1: *Explanation for User interface*

3.1 The Title Bar

The title bar contains the name of the project and application, for example:
Garage - GX IEC Developer.

3.2 The Menu Bar

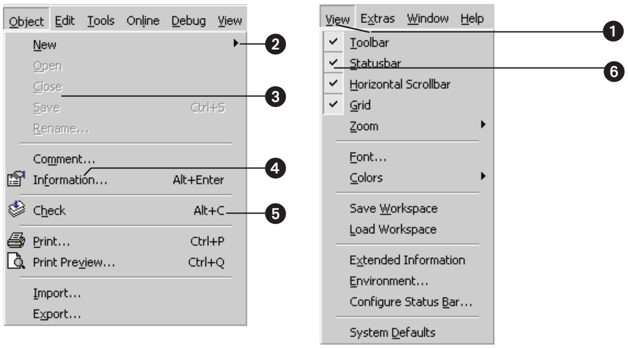
When you start GX IEC Developer only the following menus are displayed:



The complete menu bar with all the GX IEC Developer menus is not displayed until you open or create a project. When you select one of the menu titles a drop-down list is displayed showing the menu commands and options. The menu structure and the available options are context-sensitive, changing depending on what you are currently doing in the program. Options displayed in light grey are not currently available for selection.

NOTE A reference to the complete menu structure is provided at the end of this Reference Manual.

3.2.1 Menu Conventions



Item	Description
❶	Press the key corresponding to the underlined letter in the menu name.
❷	Menu commands followed by a triangle symbol branch to a submenu with additional command options, which is displayed as a pop-up menu box.
❸	Depending on what you are doing, not all menu commands are available at all points in GX IEC Developer. Menu commands displayed in light grey are currently unavailable.
❹	Menu commands followed by three periods open dialogue boxes in which additional entries and selections must be made.
❺	In some cases, key combinations can be used to activate commands directly, without opening the corresponding menu. When available, these key combinations are shown after the command in the menu.
❻	A check mark in front of a menu option shows that this option is currently activated. To deactivate an activated option, simply select it again. The check mark disappears, indicating that the option is now inactive.

Tab. 3-2: *Menus conventions*

3.2.2 Selecting menu commands

Using the mouse

① Point to the menu with the mouse pointer, click with the left mouse button and then click on the desired command in the drop-down list that appears.
Or:
Click on the appropriate tool icon in the toolbar.

Using the keyboard

① Press Alt or F10 to activate the menu bar.

② Move the highlight to the desired menu with the cursor keys → and ←, then select by pressing Enter.
Or:
Press the key corresponding to the underlined letter in the menu name (e.g. **L** for the **Online** menu).

③ Move the highlight to the desired menu option in the drop-down list with the cursor keys ↑ and ↓, then press Enter to select.
Or:
Some menu commands can be selected directly, irrespective of which menu is currently open, by pressing a key or key combination. These key combinations are shown after the menu commands in the drop-down menus (e.g. Alt F4 for **Quit** in the **Project** menu).

NOTE

All the available key combinations for menu commands are listed in the Appendix B to this manual.

3.3 The Toolbar

The toolbar enables you to select the most important menu commands directly by clicking on the corresponding icons. The toolbar is context-sensitive, i.e. different tool icons are displayed depending on what you are currently doing in GX IEC Developer. Tool icons can only be selected with the mouse.

When you start GX IEC Developer, only the icon for opening a project is displayed. You can switch the toolbar display on or off with the **Toolbar** option in the **View** menu.

NOTE | A complete list of the available tools is provided at the end of this manual.

3.4 The Project Navigator Window and the Objects it contains

The Project Navigator window is the ‘control centre’ of GX IEC Developer. It contains all the objects belonging to the project you are working on. All operations on objects are executed through this window.

The Project Navigator window is not displayed until you open a project. Closing the window automatically closes the project it contains.

You can always activate the Project Navigator window via the menu item **Navigator** in the **Window** menu.

Three different views can be selected for the project navigator via tabs below the navigator window:

Project, Calltree, Used by

The Project Navigator window is not displayed until you open a project. Closing the window automatically closes the project it contains.

3.4.1 Navigator views

Project

This view gives a total overview of the project. It contains all elements of the project.

Calltree

For this view, the corresponding root items are tasks or also POUs, if they are not related to a specific task. As subitems all used POUs are shown. Additionally, it can be defined per object, if used global variables should also be shown.

Used by

This view has exactly two root items. The first root item is the POU pool with its POUs as nodes. Subitems of the POUs and global variables are always POUs calling respectively using the corresponding POU or global variable.

3.4.2 Objects in the Project Navigator window

The following standard objects are automatically generated in the Project Navigator window for every newly-created project:

- The Library_Pool
- The PLC Parameters
- The Task_Pool
- The DUT_Pool
- Global Variables
- The POU_Pool

NOTE For a detailed description of these objects please refer to chapter 4.

3.4.3 How objects are displayed in the Project Navigator window

The entries in the Project Navigator window are arranged hierarchically. The first level is the project. The entries for the standard objects are located on the second level, from which you can branch to lower levels where they exist.

The symbol **[+]** before an entry in the Project Navigator window shows that the entry has additional lower levels (e.g. **[+] POU_Pool**).

The symbol **[–]** before an entry in the Project Navigator window indicates that its subordinate levels can be 'collapsed' so that they are no longer visible (e.g. **[–] POU_Pool**). This does not apply for the entry that contains the project name, however.

How to display hidden Navigator hierarchy levels

① Double-click on the entry whose hidden levels you wish to display.
Or:
Select the entry and select the **Expand** command in the **Tools** menu.
Or:
Press the key combination **[CTRL][+]**.
Or:
Select the entry and click on the Expand tool icon in the toolbar:

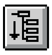
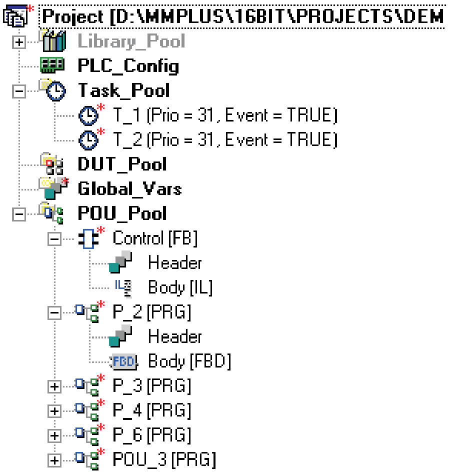


Fig. 3-2: *Tool for expanding collapsed Navigator hierarchy levels*

This expands the next level of the hierarchy in the Project Navigator window. These levels can, in turn, also contain hidden lower levels.



```
Project [D:\MMPLUS\16BIT\PROJECTS\DEM
  Library_Pool
  PLC_Config
  Task_Pool
    T_1 (Prio = 31, Event = TRUE)
    T_2 (Prio = 31, Event = TRUE)
  DUT_Pool
  Global_Vars
  POU_Pool
    Control [FB]
      Header
      Body [IL]
    P_2 [PRG]
      Header
      Body [FBD]
    P_3 [PRG]
    P_4 [PRG]
    P_6 [PRG]
    POU_3 [PRG]
```

Fig. 3-3:
Some of the lower levels in the Project Navigator window are expanded.

How to hide displayed Navigator hierarchy levels

① Double-click on the entry whose lower levels you wish to hide.
Or:
Select the entry and select **Collapse** in the **Tools** menu.
Or:
Press the key combination **[CTRL][-].**
Or:
Select the entry and click on the Collapse tool in the toolbar:

The selected level is collapsed and hidden from view.

NOTE

There are two possible responses when you double-click on a POU name, depending on the settings in the MMP701.INI file and in the menu **Extras - Options - Zoom Header/Body - Navigator Zoom** (→ Chapter 2).
- Show/hide header and body in the Project Navigator window
- Open Header and body

How to display additional object information in the Navigator

When you open a new project the additional object information is displayed by default.

① Select the **Extended Information - Navigator** command in the **View** menu.

Useful additional information is then displayed after each entry in the Project Navigator window.

Information when ‘Extended Information - Navigator’ is not active:

- Tasks: Priority, Event
- POUs: Program class
- Body: Programming language

Additional information when ‘Extended Information - Navigator’ is active:

All entries: Creation date and time

- Task_Pool, DUT_Pool, POU_Pool: Number of entries
- DUT, Global_Vars, Header: Number of variable entries
- POU: Programming language of the program body

NOTE

In the Sequential Function Chart editor you can activate and deactivate additional comments for steps with the menu **View - Extended Information - SFC editor**.

How to toggle between Symbolic Name and Address in LD/FBD

In LD and FBD you can switch between three different view modes:

- Standard Mode
  All variables are shown by their names.
- Mitsubishi Address Mode
  The variables are shown by their Mitsubishi address.
- IEC Address Mode
  The variables are shown by their IEC address.

To toggle the address view mode select **View Mode** in the menu **View**. Here you can select one of the three view modes. Alternatively you can click on **Toggle** or press Ctrl Shift M to toggle between the three view modes.

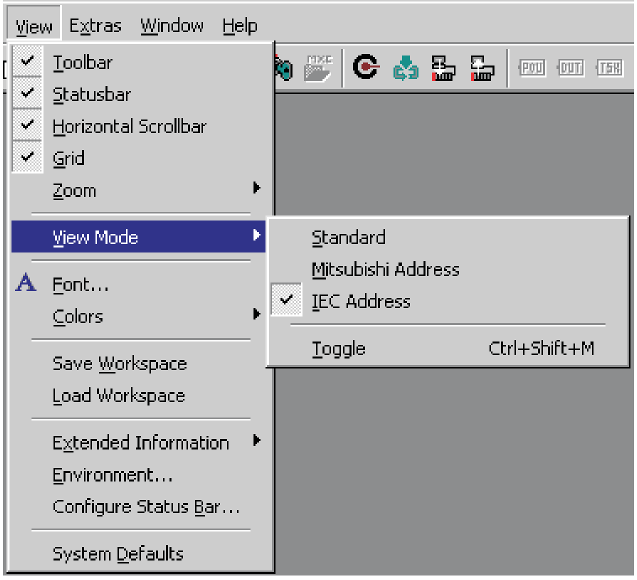


Fig. 3-4: *Menu **View Mode***

NOTES

This function is only for view purposes. All editing/finding/tracking etc. functionalities refer to variable names and not to addresses.

The selected mode is valid for the currently active editor window only and is not persistent. When you close a window and open it again, the editor will be in Standard Mode again.

In **Mitsubishi Address Mode/IEC Address Mode** the address of a variable that is not yet known because the project has not been compiled yet, or the address cannot be evaluated for other reasons, the name is shown preceded by an asterisk (‘*’).

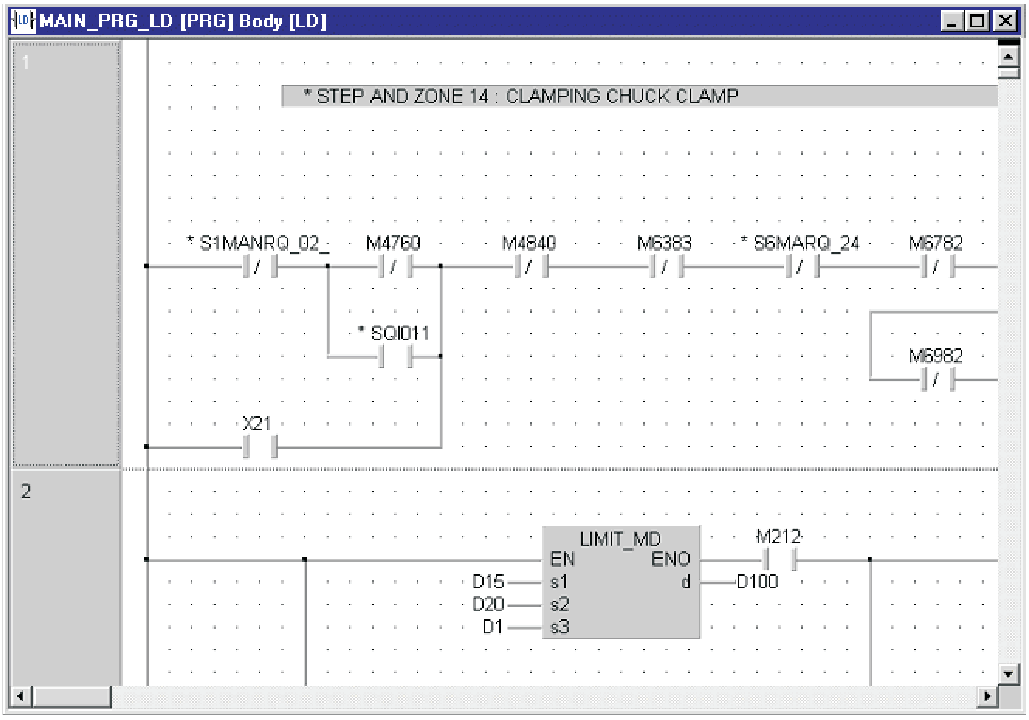


Fig. 3-5: *Example for an address not known*

In the case of an array element with non static index, i.e. the index part contains one or more indices that do not represent a number, the whole array range is shown in the view mode **Mitsubishi Address Mode/IEC Address Mode**.
The address ranges are shown as <start address>–<end address>. To avoid long address range strings, the end address is shortened by the redundant first part, e.g.

- D122-25
- D100-152
- %MW19.3.17.2-3
- %MW19.3.17.2-18.3

NOTES

For DUTs and ARRAY of DUTs no addresses but only symbolic names are shown, preceded by an asterisk (‘*’).

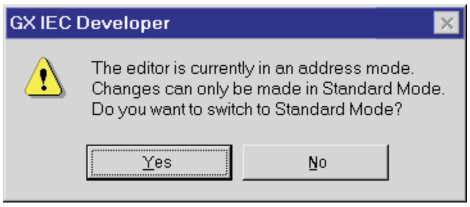
The following elements are not toggled and are shown by their symbolic names:
- Index Registers like „X0Z0"
- Constants (VAR_CONSTANT/VAR_GLOBAL_CONSTANT)
- Labels

The address view mode of the following elements can be toggled by the **View Mode** function:

Element	IEC-Type(s)	Remark
simple variable	BOOL, DINT, DWORD, INT, REAL, STRING[], TIME, WORD	
array variable of simple type	ARRAY of BOOL, DINT, DWORD, INT, REAL, STRING[], TIME, WORD	array and array elements
DUT members	simple variable or array variable of simple type (see above)	
DUT members of Array of DUT element	simple variable or array variable of simple type (see above)	
direct address	—	e.g. M16, K4M0, K4Y12 …

Tab. 3-3: *Elements that can be toggled*

While the editor is in an address mode (Mitsubishi or IEC) it is in the read only mode. If you try to edit a variable the following message box will appear:



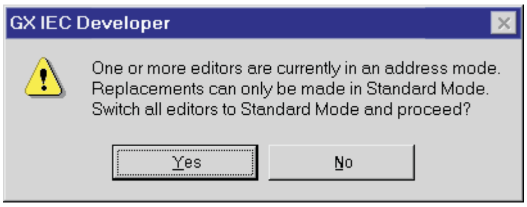
GX IEC Developer

The editor is currently in an address mode.
Changes can only be made in Standard Mode.
Do you want to switch to Standard Mode?

Fig. 3-6:
Message box

This message box is also shown when you perform a **Find** or **Find & Replace** action. If you selects **Yes** in the message box the editor is switched to standard mode and the edit operation is be performed, else the editor remains in the address mode and no change is made.

If you perform a **Replace All** and there are editors in an address mode the following message box will appear. If you select **Yes** in the message box, all editors are switched to standard mode and the **Replace All** operation is performed, else the operation is cancelled.



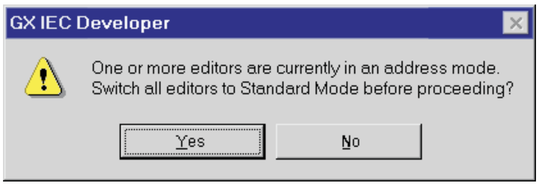
GX IEC Developer

One or more editors are currently in an address mode.
Replacements can only be made in Standard Mode.
Switch all editors to Standard Mode and proceed?

Fig. 3-7:
Message box

During a find operation in the view mode no change is made. The edit control that contains the found text is highlighted. In the address view mode not the found text is shown but the address is highlighted.

If you start tracking of a variable and there are editors in an address mode the following message box will appear.


GX IEC Developer

One or more editors are currently in an address mode.
Switch all editors to Standard Mode before proceeding?

Fig. 3-8:
Message box

If you select **Yes** in the message box, all editors are switched to standard mode before tracking is started, else the view modes are not changed and tracking starts immediately in the selected address view.

In **Online-Mode** you can also toggle between view modes.

3.4.4 Editing objects

The objects at the lowest level of each branch can be opened for editing. These levels are not marked with **[+]** or **[–]** symbols (e.g. PLC_Parameter, Header, Body). Opening them displays either a dialogue box (e.g. PLC_Parameter) or an object window (e.g. Header).

Most operations, such as opening, copying and deleting, can be performed directly in the Project Navigator window with key combinations. The corresponding menu commands are in the **Object** menu. For details on working with objects, see chapter 4.

NOTE

All key combinations used for working with objects and the Project Navigator are listed and explained in the Appendix B.

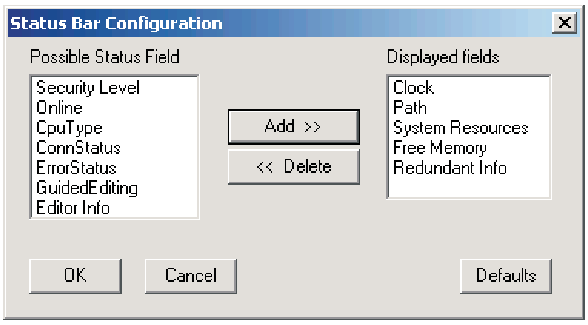
3.5 The Status Bar

The status bar contains information on the current status of your project. You can enable or disable its display and the information it contains is also configurable. The command for switching the status bar display on and off is in the **View** menu.

How to configure the Status Bar

① Double-click in the status bar.
Or:
Select **Configure Status bar** in the **View** menu.

The following dialogue box is displayed:



Status Bar Configuration

Possible Status Field: Security Level, Online, CpuType, ConnStatus, ErrorStatus, GuidedEditing, Editor Info

Displayed fields: Clock, Path, System Resources, Free Memory, Redundant Info

Fig. 3-9:
Status bar configuration

Editor Info: Display the POU name and the number of variable entries in the header

Security Level: Display current security level

Online: Display online status

CpuType: Display the PLC CPU for which the project is configured

Conn Status: Display active connection to the PLC system: PLC state, current scan time. Display depends on selection of monitoring mode and PLC status.

Error Status: Display PLC system error status.
Display depends on selection of monitoring mode and PLC status.

GuidedEditing: Display of the editing mode (OVR/INS)

Clock: Display time

Path: Directory path on which the current project is stored

System Resources: Free Windows system resources in %

Free Memory: Amount of available free memory (in bytes)

Redundant Info: For QnPRH PLCs only. The Status Bar will alternate between displaying the operating mode ("Backup" or "Separate") and the identity of the connected PLC ("Control (System A)" or "Backup (System B)" or any combination thereof) when Monitoring Mode is active. The interval between changes will depend on the poll rate set in **Extras** → **Options** → **Monitor Mode** (see section 2.7.11).

How to add status bar items

① Select the item to be displayed in the **Possible Status Field** list box.

② Click on the **Add** button to display the selected item in the Status bar.

③ Repeat steps ① and ② for each additional item to be displayed in the status bar.

How to delete status bar items

① Select the item to be deleted in the **Displayed fields** list box.

② Click on the **Delete** button to delete the selected item from the Status bar..

③ Repeat steps ① and ② for each item to be cleared from the status bar.

NOTE

You can restore default status bar display settings at any time by clicking on the **Defaults** button.

3.6 Working with Windows

This section only contains a short explanation of the various operations performed with screen windows. For detailed instructions and definitions of the terms used, please refer to the Windows User's Guide. The procedures used with windows in Windows applications programs are standardised and are not specific to GX IEC Developer.

The following window types are used in GX IEC Developer:

- The Project Navigator window
- Object Windows
- Dialogue Boxes

NOTE

Although it is possible to display several different windows on the screen simultaneously, only one window can be active for editing at any one time. The currently active window is identified by a coloured title bar.
Please note that you can only open and edit one project, in one Project Navigator window. But you can start GX IEC Developer several times to open and edit more than one project at a time

3.6.1 Object windows

How to open windows

① In the Project Navigator window, double-click on the name of the object whose window you wish to open.
Or:
Select the object in the Project Navigator window by moving the highlight with the cursor keys ↑ and ↓, and then press Enter or select the **Open** command in the **Object** menu.

The program opens the object window and displays it to the right of the Project Navigator window. You can open several object windows at the same time.

NOTE

When the object window of a body (header) is open you can open its header (body) with the **Open Header (Open Body)** command in the **Object** menu.

How to select an open window

① Click anywhere in the window.
Or:
Select the desired window with the key combination Ctrl Tab.
Or:
Choose the window to be selected from the list of open windows shown in the **Window** menu.

The title bar of a selected window is highlighted in color.

How to close a window

- Double-click on the control menu box of the window you wish to close.
  Or:
  Select the **Close** command in the window's control menu.
  Or:
  Select the window and press Ctrl F4 to activate its control menu **Close** command.
  Or:
  Select the window and select the **Close** command in the **Object** menu.

The program will ask you if you want to save any editing changes before closing the object window.

NOTE

> You can close all opened windows at the same time with the **Close All** command in the **Window** menu.

How to arrange windows without overlaps (tiling)

① Select the **Tile Horizontally** or **Tile Vertically** command in the **Window** menu.

The Project Navigator window is displayed on the left of the screen. The object windows are arranged horizontally or vertically. The size of the object windows depends on how many windows are currently open.

How to arrange windows with overlaps

① Select the **Cascade** command in the **Window** menu.

The Project Navigator window and all open object windows are overlapped across the screen so that all the title bars are visible, starting at the top left corner.

3.6.2 Changing window display size

Buttons

▁ Minimise window

🗗 Restore window

□ Maximise window

Window or Full-size window → Icon

Click on the **Minimise** button in the object window (in the upper right corner of the window title bar), or select **Minimise** in the window's control menu.

You can move and position the icons anywhere on the screen. The **Arrange Icons** command in the **Window** menu arranges all the icons in a row at the lower edge of the screen.

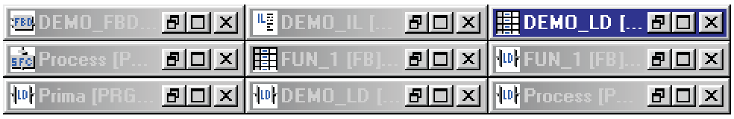


Fig. 3-10: *The object window icons*

Icon → Previous window size

Click on the **Restore** button of the icon you wish to open, or simply double-click on the icon.

Full-size window → Previous window size

Click on the **Restore** button (in the upper right corner of the window title bar), or select the **Restore** command in the control menu.

Window or Icon → Full-size window

Click on the **Maximise** button (in the upper right corner of the window title bar), or select the **Maximise** command in the control menu.

NOTE You can find a list of all the key combinations for working with windows in the Appendix B.

3.6.3 Dialogue boxes

Dialogue boxes are displayed whenever additional user input is required, for example when you select the menu commands that are followed by three periods
(e.g. **Open** … in the **Project** menu). In addition to asking you to make entries, dialogue boxes can also contain important information and warnings.

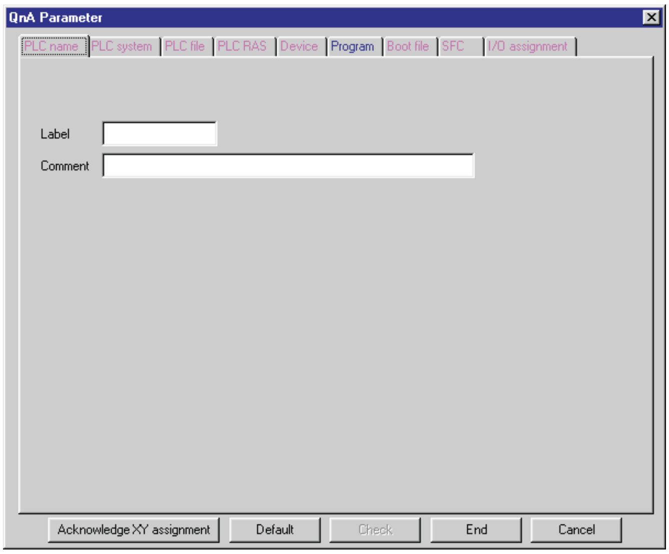


Fig. 3-11: *The PLC Parameter dialogue box*

How to select an open dialogue box

① Click anywhere within the dialogue box.
Or:
Select the desired box with the key combination Alt F6.

The title bar of a selected dialogue box is displayed with color shading.

How to close dialogue boxes

- If you want to discard the entries or changes you have made in the box, double-click on the control menu button of the dialogue box.
  Or:
  Select the **Close** command from the dialogue box's control menu.
  Or:
  Click on the **Cancel** button.
  Or:
  Press the Esc key.
- If you want to accept the entries in the dialogue box, click on the **OK** button.
  Or:
  Select the **OK** button and press the Enter key.

NOTE | You can find a list of all the key combinations for working with windows in the Appendix B.

3.7 Special Functions and Dialogs of Redundant PLC Systems

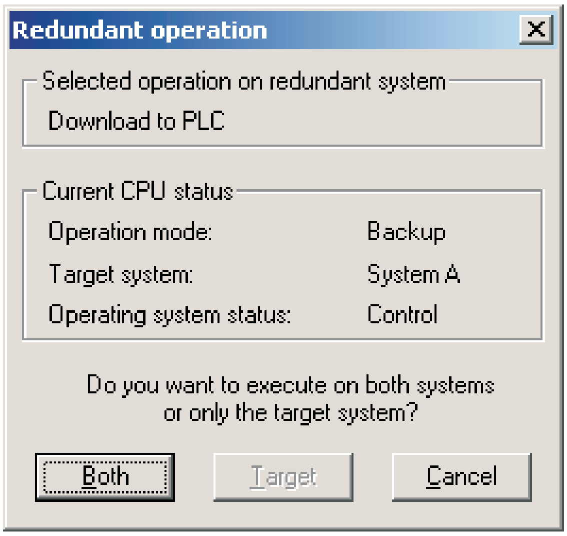
3.7.1 Rules for Redundant PLC Systems in Backup Mode

Depending on the accessibility of the PLCs in a redundant system the following rules apply for online operations.

- If the system, which was selected as the target system in the menu **Transfer Setup** → **Ports** (see section 8.3.1), cannot be accessed:
  - In all modes an error message will be displayed and the operation is aborted.
- If the system, which was selected as the target system in the menu **Transfer Setup** → **Ports** (see section 8.3.1), can be accessed:
  - In Debug mode the operation is executed on the target system.
  - In Separate mode the operation is executed on the target system.
  - In Backup mode, if the other system can also be accessed, a confirmation dialogue (→ 3.7.2) will be displayed in which the execution on the target and/or the other system can be advised.
  - In Backup mode, if the other system cannot be accessed, an error message will be displayed and the operation is aborted only for **Online (Ex)Change**. For other operations a confirmation dialogue (→ 3.7.2) will be displayed in which the execution on the target system can be advised.

3.7.2 Confirmation Dialogue in Backup Mode

Before executing specific operations in backup mode of a redundant system the following dialogue will be displayed. It shows the selected operation and the status of the redundant PLC system.



Redundant operation

Selected operation on redundant system

Download to PLC

Current CPU status

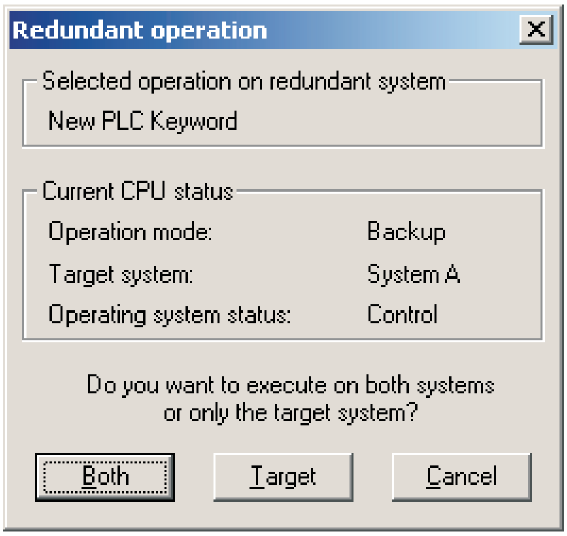
Operation mode: Backup

Target system: System A

Operating system status: Control

Do you want to execute on both systems or only the target system?

Fig. 3-12:
Confirmation Dialogue for allowed on both PLCs only



Redundant operation

Selected operation on redundant system

New PLC Keyword

Current CPU status

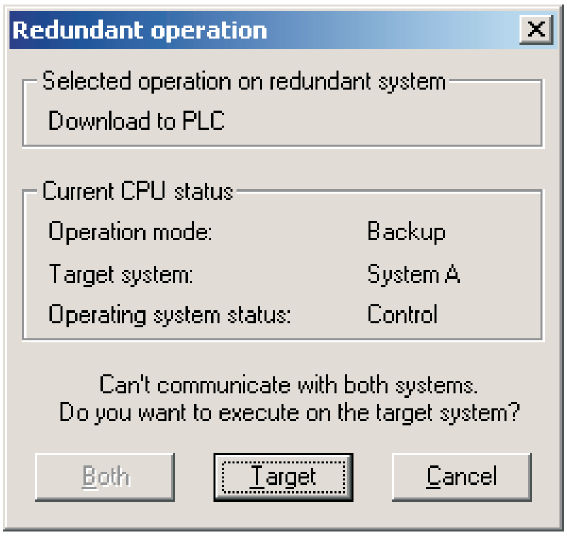
Operation mode: Backup

Target system: System A

Operating system status: Control

Do you want to execute on both systems or only the target system?

Fig. 3-13:
Confirmation Dialogue for allowed on either the target or on both PLCs



Redundant operation

Selected operation on redundant system

Download to PLC

Current CPU status

Operation mode: Backup

Target system: System A

Operating system status: Control

Can't communicate with both systems.
Do you want to execute on the target system?

Fig. 3-14:
Confirmation Dialogue if the other system is not accessible

If you choose **Both** on this dialogue (Fig. 3-12 and Fig. 3-13) the operation will be executed on both systems.
If you choose **Target** (Fig. 3-13 and Fig. 3-14), the operation will only be executed on the system specified in Transfer Setup. (For some functionalities this button will not be available.)
If you choose **Cancel**, the operation will not be executed on either system.

The following functionalities will use this dialogue with the **Target** button disabled (Fig. 3-12):

- Download to PLC
- Online Program (Ex)Change
- Copy Program Memory to Flash ROM
- Download Symbolic Information
- Online Verify
- Format Drive
- Defragment Drive
- PLC Clear → All
- PLC Clear → Latch

The following functionalities will use this dialogue with the **Target** button enabled (Fig. 3-13):

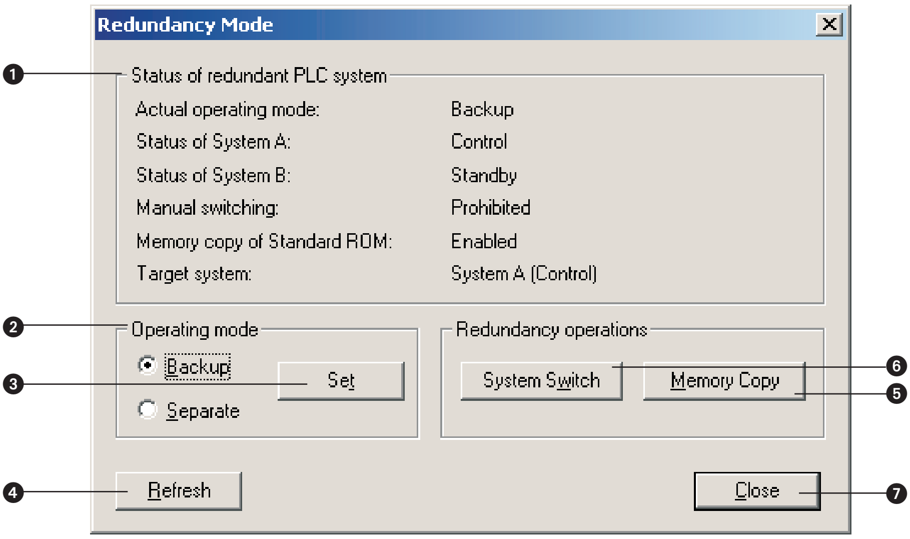
- New PLC Keyword
- Delete PLC Keyword

If the PLC selected in **Transfer Setup** → **Ports** is accessible, but the other system is not, the dialogue shown in Fig. 3-14 will be displayed for the following operations. In this case the **Both** button is disabled and the **Target** button is enabled.

- Download to PLC
- Copy Program Memory to Flash ROM
- Download Symbolic Information
- Online Verify
- Format Drive
- Defragment Drive
- New PLC Keyword
- Delete PLC Keyword
- PLC Clear → All
- PLC Clear → Latch

3.7.3 PLC Redundancy Mode

For the QnPRH PLCs it is possible to switch between Backup and Separate mode remotely. The following dialogue is opened by **Online** → **PLC Redundancy Mode** and shows the current status of both systems, and whether manual switching of the Control and Standby systems is possible. It also provides the functionalities **System Switch** and the **Memory Copy**.



Status of redundant PLC system

Actual operating mode:	Backup
Status of System A:	Control
Status of System B:	Standby
Manual switching:	Prohibited
Memory copy of Standard ROM:	Enabled
Target system:	System A (Control)

Fig. 3-15: Redundancy Mode dialogue

Item	Description
❶	Status information of redundant PLC
❷	Check to set Backup or Separate mode
❸	Click to set desired mode
❹	Click to re-read operating state information from the PLCs
❺	Click to switch Control and Standby systems manually
❻	Click to copy the program memory from the Control system to the Standby system
❼	Click to close the dialogue

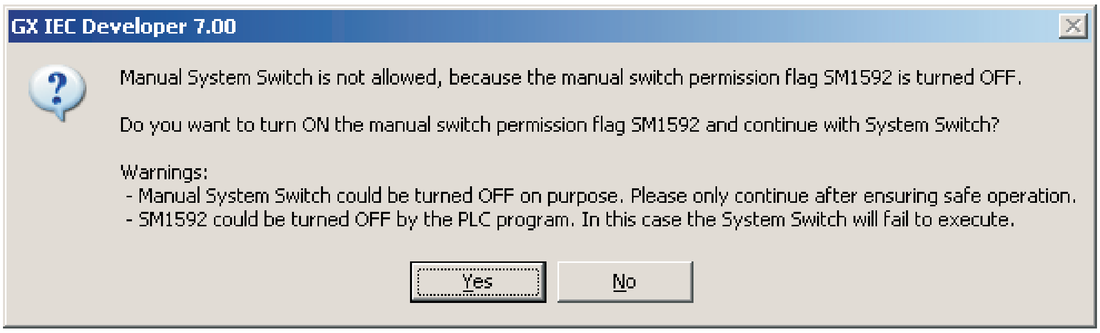
Tab. 3-4: Details of the Redundancy Mode dialogue

NOTE

The **Set Operating Mode**, **System Switch** and **Memory Copy** operations can only be executed if the control system is the currently connected system.

System Switch

The **System Switch** operation manually switches control and standby systems. After successfully executing system switch, the previous control system will become the standby system, and the previous standby system will become the control system. The **System Switch** functionality can only be executed if the appropriate special relay SM1592 is turned ON. If it is OFF, the following dialogue will be displayed:



GX IEC Developer 7.00

Manual System Switch is not allowed, because the manual switch permission flag SM1592 is turned OFF.

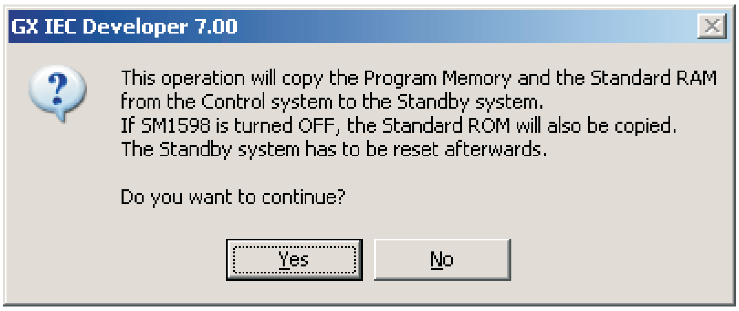
Do you want to turn ON the manual switch permission flag SM1592 and continue with System Switch?

Warnings:
- Manual System Switch could be turned OFF on purpose. Please only continue after ensuring safe operation.
- SM1592 could be turned OFF by the PLC program. In this case the System Switch will fail to execute.

Fig. 3-16: *Confirmation dialogue for system switch*

Memory Copy

The **Memory Copy** operation copies the program memory and the standard RAM from the control system to the standby system. If the special relay SM1598 is turned OFF, it will also copy the standard ROM. The following message will be displayed before executing this operation:



GX IEC Developer 7.00

This operation will copy the Program Memory and the Standard RAM from the Control system to the Standby system.
If SM1598 is turned OFF, the Standard ROM will also be copied.
The Standby system has to be reset afterwards.

Do you want to continue?

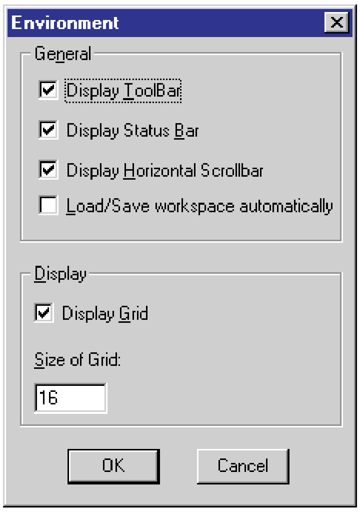
Fig. 3-17: *Confirmation dialogue for memory copy*

3.8 Customising the Screen Display

The **View** menu contains all the commands for customising the display for your personal needs.

Toolbar, Status Bar, Horizontal Scrollbar and Grid

You can switch these display elements on and off with the corresponding options in the **View** menu, or with the check boxes in the dialogue box that is displayed when you select the **Environment** option.



Environment

General

☑ Display ToolBar

☑ Display Status Bar

☑ Display Horizontal Scrollbar

□ Load/Save workspace automatically

Display

☑ Display Grid

Size of Grid: 16

Fig. 3-18:
The toolbar, status bar, horizontal scrollbar and the grid can be activated or deactivated with the check boxes in the Environment dialogue box. Grid display is enabled.

The check mark ✓ in front of the menu command in the menu **View** identifies options that are currently enabled. When you deactivate an option the check mark is no longer displayed.

How to set the screen zoom factor

① Select the **Zoom...** command in the **View** menu.

② Select one of the four standard zoom factors: **50%, 75%, 100%, 150%**.

In addition to these four standard zoom factors you can also enter your own screen display magnification by changing the display grid size setting.

① Select the **Environment...** command in the **View** menu.

② Enter a value between 4 and 128 in the Size of **Grid** field.

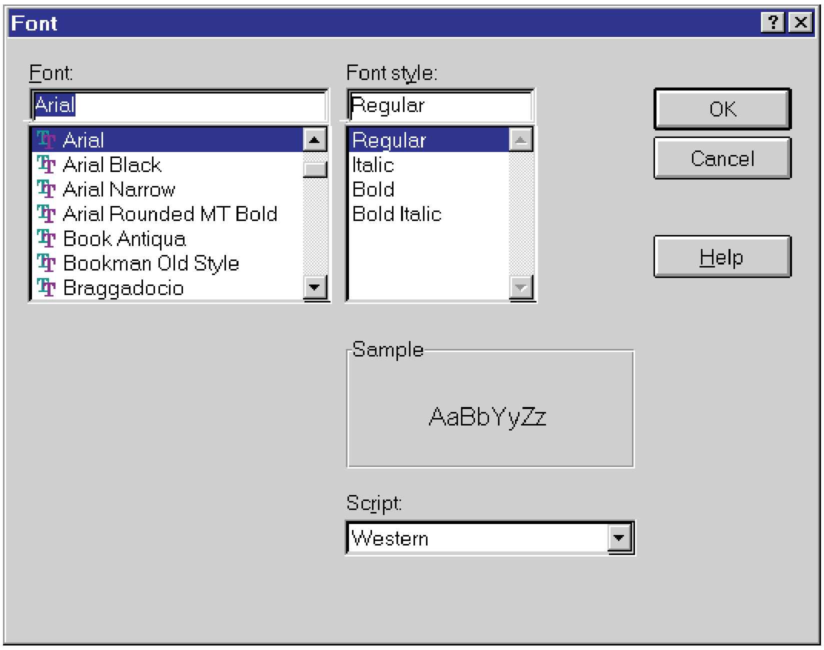
The standard zoom factors correspond to the following grid settings:

Zoom factor:	50%	Grid size:	8
Zoom factor:	75%	Grid size:	12
Zoom factor:	100%	Grid size:	16
Zoom factor:	150%	Grid size:	24

How to select the screen font

Select the **Font** command in the **View** menu.

Choose the font and style you wish to use for the screen display.



Font

Font: Arial

Font style: Regular

Sample AaBbYyZz

Script: Western

Fig. 3-19:
Screen font selection

All GX IEC Developer texts will be displayed in the selected font. You can use all the fonts installed in your Windows configuration.

NOTE | The screen font setting is stored together with the project.

How to set the display colors

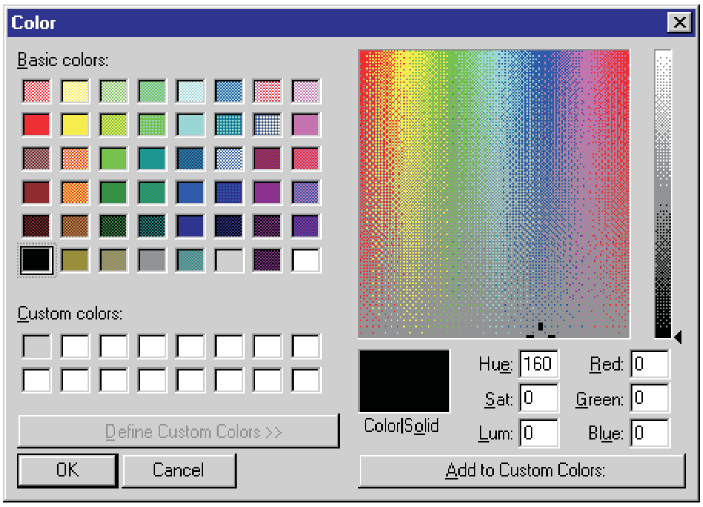
① Select the **Colors** command in the **View** menu.

② Select the screen element whose color you wish to change:

Background
Selection
Text
Selected Text
Read Only Text
Locked Text
Foreground
Monitoring
Errors
Shade Left Top
Shade Right Bottom
Grid
Browser Items
SFC Raster Elements

The **Color** dialogue box is displayed.

③ Click on the color you want in the palette of basic colors.



Hue: 160
Sat: 0
Lum: 0
Red: 0
Green: 0
Blue: 0

Fig. 3-20:
Display color selection

The current color of the selected element is shown in the two-line palette below the basic colors palette.

NOTE You can also define your own custom colors. For details please refer to the Windows User's Guide. The colors settings are stored with the project.

How to save the current Workspace

① Select the **Save Workspace** command in the **View** menu.

NOTE This command only saves the size, arrangement and display of the opened windows - with the exception of the Project Navigator window - in the current workspace, ignoring the font and color settings. The workspace is saved with the project in the file WORKSPACE.SYS.

How to load a previously saved Workspace

① Select the **Load Workspace** command in the **View** menu.

This command automatically loads the settings stored with the **Save Workspace** command by loading the data from the WORKSPACE.SYS file.

How to load the system defaults

① Select the **System Defaults** command in the **View** menu.

When you create a new project the following system defaults are always selected automatically:

- Toolbar display enabled
- Status bar display enabled
- Horizontal scrollbar display enabled
- Grid display enabled
- Zoom factor: 100 %
- Grid size: 16
- Extended information display in Project Navigator enabled
- Background: light grey, Selection: dark grey
- Text: black
- Selected Text: white
- Read Only Text: dark grey
- Locked Text: dark grey
- Foreground: light grey
- Monitoring: blue
- Errors: red
- Shade left top: white
- Shade right bottom: dark grey
- Grid: black

3.9 Mouse and keyboard operation

All operations in GX IEC Developer can be performed both with the mouse and the keyboard. In most cases, a single click with the left mouse button corresponds to the Space key, and a double-click corresponds to the Enter key.

The instructions below are primarily for working with the mouse, but special keyboard key combinations are also included. All the key combinations for working with tables and the editors are listed in the Appendix B of this manual.

NOTE

For full details on using the mouse in Windows please refer to Windows Fundamentals section in the Windows User's Guide.

3.10 General editing functions

3.10.1 Using menu or keyboard operations

Most of the commands in the **Edit** menu are identical in all Windows application programs.

NOTE

You can find detailed explanations of these commands in your Windows User's Guide.

How to cut, copy and delete data

① Select the element you wish to edit (see following pages):

- Cell contents in a Declaration Table: Enable the editing mode in the cell and select the cell text.
- Lines in a Declaration Table: Select one or more lines.
- Entire networks in the editors: Select the network bar to the left of the network you wish to edit.
- Ranges (instruction, operand, lines) in the Instruction List: Select the range to be edited.
- Program elements in the graphical editors: Select the elements to be edited.
- Program elements in the Sequential Function Chart Language: Select the elements to be edited. When you select a Step its corresponding Transition is automatically selected as well.

② Select the editing command you wish to execute in the **Edit** menu.
Or:
Press the corresponding key combination:

Cut	X
Copy	C
Delete	Del

WARNING:
Deletions are performed directly, without prompting for confirmation. If you inadvertently delete something you can restore the deleted data by selecting the Undo command in the Edit menu. However, please note that this only works if you execute Undo directly after performing the deletion!

How to paste data

① Select the position at which you wish to insert an element:

- Cell contents in a Declaration Table: Activate editing mode in the cell in which you wish to insert the text from the clipboard.
- Lines in a Declaration Table: Select a cell in the line before which you wish to insert the lines from the clipboard, or select the lines you wish to overwrite with the data from the clipboard.
- Entire networks in the editors: Select the network bar of the network before which you wish to insert the network stored in the clipboard.
- Ranges (instruction, operand, lines) in the Instruction List: Position the cursor at the point at which you wish to insert the text from the clipboard.
- Sequential Function Chart elements: Select the point in the sequence before which you wish to insert the data from the clipboard.

② Select **Paste** in the **Edit** menu.
Or:
Press the corresponding key combination:

Paste	Ctrl V

③ Elements in the graphical editors: When you select the **Paste** command the mouse pointer changes to a plus sign symbol. Position the pointer at the location at which you wish to insert the elements from the clipboard.

NOTE

Pasting Parameter data into a Q Remote I/O project from a non-Q Remote I/O project is not be supported. Pasting Parameter data into a non-Q Remote I/O project from a Q Remote I/O project is also not supported.

How to undo editing operations

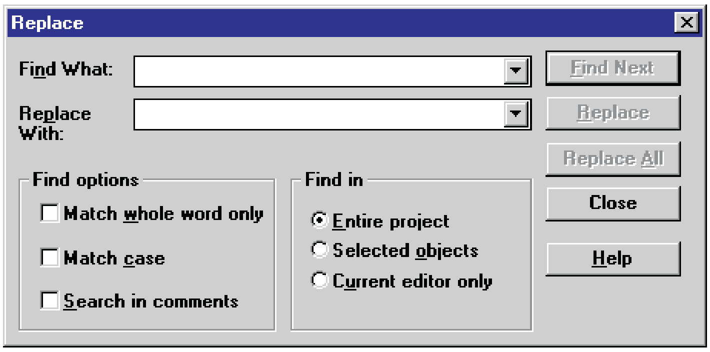
① Select **Undo** or **Redo** in the **Edit** menu.
Or:
Press the corresponding key combination:

Undo	Alt BACKSPACE
Redo	Ctrl BACKSPACE

Undo undoes the effects of the last editing operation.
Redo undoes the effects of the last Undo operation.

Global Search and Replace

The **Search and Replace** functions can now be used globally to search the entire project (**Edit - Find/Replace** menu).



Find What:
Replace With:

Find options
□ Match whole word only
□ Match case
□ Search in comments

Find in
● Entire project
○ Selected objects
○ Current editor only

Fig. 3-21:
Search and replace dialogue

Find in:

This option enables you to search and replace in

the entire project (All objects)

individual objects selected in the Project Navigator

the current editor only

Find What / Replace With:

You can enter the following elements as find/replace arguments:

- Variable identifiers
- Data types
- Addresses
- Sequential function chart language step names
- Sequential function chart language transitions
- POU names

Find options:

Match whole word only: Search the entire word only

Match case: Search and replace operation for text exactly as entered, with upper/or lower case characters

Search in comments: Activating this option extends the search and replace operation to comment texts

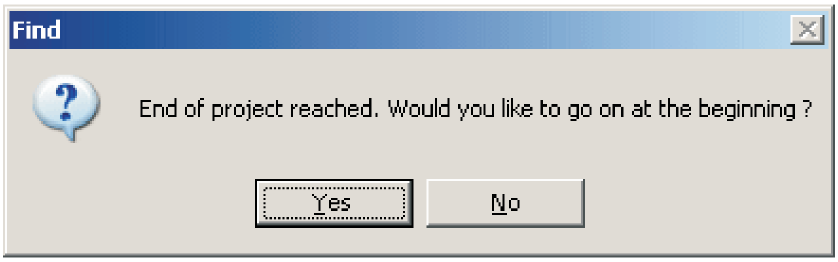
NOTE

You can replace function blocks in the graphical editors but no check is performed. Also, the replace procedure cannot be reversed; the only way to restore the project to its original state is by not saving it and reloading the old version from the disk.

Find Next / Replace:

By pressing the button **Find Next** or **Replace**, the next matching item will be searched and highlighted.

When you selected the option **Current editor only** under **Find in**, and when the last matching item was found and the button **Find Next** or **Replace** is pressed again, the following dialogue will be shown.



Find

End of project reached. Would you like to go on at the beginning ?

Fig. 3-22:
Dialogue box

Confirm by pressing **Yes**. Selecting **No** will close the dialogue box.

3.11 Drag&Drop by mouse operation

Drag & Drop offers a more efficient way of editing which is based on mouse usage in comparison to the equivalent necessary lengthy and various menu commands concerning clipboard handling.
Basically there are two different kind of drag&drop actions which can be performed:

- copy objects
- move objects

3.11.1 Functional description and user interface

In GX IEC Developer, a drag operation is initiated, if the left mouse button is clicked and held down, while the mouse is moved. When the mouse button is released, the drop operation is executed. What happens then is very similar to copy/paste (resp. cut/paste).

We call the object(s), that are dragged (e.g. an object or a selection in the navigator) the dragged item(s). The context where the dragged items come from is called drag source, while the context, where the drop is executed is called drop target. More explanation on this will follow below.

The dragged item and the drop target can be in different instances, but not all contexts can be drag sources and/or drop targets.

When dragging objects over drop targets, the shape of the mouse cursor indicates which operation (copy or move) would be performed, when dropped over this very target:

🚫 drop not possible on actual target

copy

move

For all objects (listed below) the decision between the two possible actions (copy or move) is made by the context, except for those drop targets that don’t support move actions such as the EDM for instance. The action (copy or move), decided by the context is called default action.

The default action can be overruled by the CTRL or the SHIFT key:

- Holding down the CTRL key, while a drop is executed, makes the drop result in a COPY operation.
- Holding down the SHIFT key, while a drop is executed, makes the drop result in a MOVE operation.

Until the moment the object is dropped (releasing of the left mouse button) you can decide between the two methods by pressing or releasing the CTRL or the SHIFT key.

Pressing the ESC key after starting the drag & drop, but before executing the drop (ie. with the mouse button still pressed) aborts the current drag & drop action. CTRL and SHIFT must be released before pressing ESC, because these key combinations may have other effects (e.g. CTRL+ESC brings up the Windows Start menu). When the drop is executed (the mouse button is released), the ESC key has no effect anymore, but the Undo functionality may be used.

After drop, the drop target editor gets the focus in its application instance.

Undo and Redo are window oriented, i.e. if the drag & drop operation involves two editors, the changes made in each editor may be undone by calling Undo in them separately. E.g. after a *move* operation between two editors, calling Undo in the drop target editor deletes the newly inserted object, while calling Undo in the drag source editor undeletes the moved object. Moving an object in its own editor can be undone in one step (because it happened in the same window).

The following section describes the objects which are involved in the drag & drop process resp. which objects can be dragged and which objects serve as valid dropping areas.

Furthermore the valid scope for the dragged objects will be discussed because this is the decisive factor whether a dragged object will be moved or copied on default. The scope is the area of an object, in which he is existing (e.g. POUs exist in the POU pool). If the scope of an object is left during a drag & drop operation (e.g. drag & drop a POU to another project or to a user library in a project), the default drag & drop operation (copy or move) may change.

Dragged objects and their valid scope inside one project

In general: If a dragged objects leaves its valid scope the default operation for the objects will be a COPY operation. For the other case the object is simply MOVED. A MOVE operation is not supported for these items if they are dragged out of the valid scope (SHIFT key has no effect).

Moreover, if multiple objects from different scopes are dragged together (e.g. a POU from the POU pool and the GVL) only the COPY operation is supported. This is then exactly the same operation as if copy/paste would have been made between the same source and target.

The following list gives an overview.

Possible dragged items	Scope
POU	POU pool①
DUT	DUT pool①
Task	Task pool①
Action	SFC POU②
GVL	Project or user library③
Local declarations (selected in a POU header editor)	POU header declaration editor
Global declarations (selected in any GVL editor)	GVL declaration editor
Global declarations (selected in any GVL editor)	POU header declaration editor
IL network	IL body editor
LADDER network	LADDER body editor
FBD network	FBD body editor
SFC elements	SFC body editor
Selected Text	Textual body editor
EDM	EDM declaration editor
IL Text	IL-body editor
MELSEC IL Text	MELSEC IL body editor

Tab. 3-5: Overview of possible dragged items

① POUs, DUTs and tasks cannot be moved in their respective scopes, because they are automatically sorted and therefore changing their order cannot be made by dragging. The copy operation however is supported also for their own scope, by holding down the CTRL key when dropping.

② Actions cannot be moved at all, only the copy operation is supported for them. The copy operation however is supported also for their own scope, by holding down the CTRL key when dropping.

③ GVLs cannot be dropped in their own scope (neither the copy nor the move operation is supported). Outside their scope both operations are possible.

3.11.2 Valid dropping targets inside one project

After describing the items which can be dragged the following section will explain in detail where these items can be dropped to and which preconditions have to exist for dropping.

The following tables give an overview on the drag/drop mechanism inside one project. All drag/drop actions are only defined for the project view of the navigator.

Drag & Drop of objects in the navigator

Dragged item	Dropping target	Action	Precondition
POU	Project	Item will be inserted into project POU pool	User needs WRITE access rights for the project and the POU pool
	POU pool of project	Item will be inserted into project POU pool	User needs WRITE access rights for the project and the POU pool
	POU pool of user library	Item will be inserted into user library POU pool	User needs WRITE access rights for the library and its POU pool
	Task	Item will be inserted into the task. The program is appended at the end of the program list.	Task editor does not have to be opened. User needs WRITE access rights for the task MOVE action is not possible
DUT	Project	Item will be inserted into project DUT pool	WRITE access rights
	DUT pool of project	Item will be inserted into project DUT pool	
	DUT pool of user library	Item will be inserted into user library DUT pool	
TASK	Project	Item will be inserted into project TASK pool	WRITE access rights for the project and TASK pool
	Task pool		
Action	SFC POU	Item will be inserted into the action pool of the associated SFC POU	WRITE access rights to the SFC POU and action pool
	SFC action pool		
GVL (a)	Project	GVL entries of (a) will be inserted into the project GVL	WRITE access rights
	Library	GVL entries of (a) will be inserted into the user library GVL	
	GVL (b)	GVL entries of (a) will be appended into the GVL (b)	
Mixed objects	Project	Items will be inserted into the project	WRITE access rights MOVE action is not possible

Tab. 3-6: *Drag & Drop of objects in the navigator*

Drag & Drop of declarations

Dragged item	Dropping target	Action	Precondition
EDM entry	EDM	Entry will be inserted as new copy	MOVE action is not possible
Local declarations	POU	Declaration will be inserted into the associated header	WRITE access rights
	POU body		
	Header in the navigator	Declaration will be inserted into the header	
	EDM	Entry will be inserted into the EDM (in POUName.VarName format)	MOVE action is not possible[1]
Global declarations	POU	Items will be declared as ~~external~~ local in the associated header with the appropriate class (VAR_GLOBAL => VAR)	WRITE access rights
	POU body		
	POU header (and POU header editor)		
	Any GVL (and GVL editor)	Items will be inserted into the GVL	
	EDM	Entry will be inserted into the EDM	MOVE action is not possible[2]

Tab. 3-7: *Drag & Drop of declarations*

[1] Only those declarations will be inserted into the EDM that existed at the time the GVL was last compiled. If the GVL was not compiled yet, no declarations will be inserted into the EDM.

[2] Only those declarations will be inserted into the EDM that existed at the time the POU was last compiled. If the POU was not compiled yet, no declarations will be inserted into the EDM.

Drag & Drop of body objects

Dragged item	Dropping target	Action	Precondition
IL network	IL POU body editor	Items will be inserted at that position where the insertion cursor will appear.	WRITE access rights
MELSEC IL network	MELSEC IL POU body editor	Items will be inserted at that position where the insertion cursor will appear.	WRITE access rights
Ladder network	Ladder POU body editor	Items will be inserted at that position where the insertion cursor will appear.	WRITE access rights
FBD network	FBD POU body editor	Items will be inserted at that position where the insertion cursor will appear.	WRITE access rights
	Ladder POU body editor		
SFC element	SFC POU body editor	Items will be inserted at that position where the insertion cursor will appear.	WRITE access rights[1]
Selected text	Textual POU body editor	Items will be inserted at that position where the insertion cursor will appear.	WRITE access rights

Tab. 3-8: *Drag & Drop of body objects*

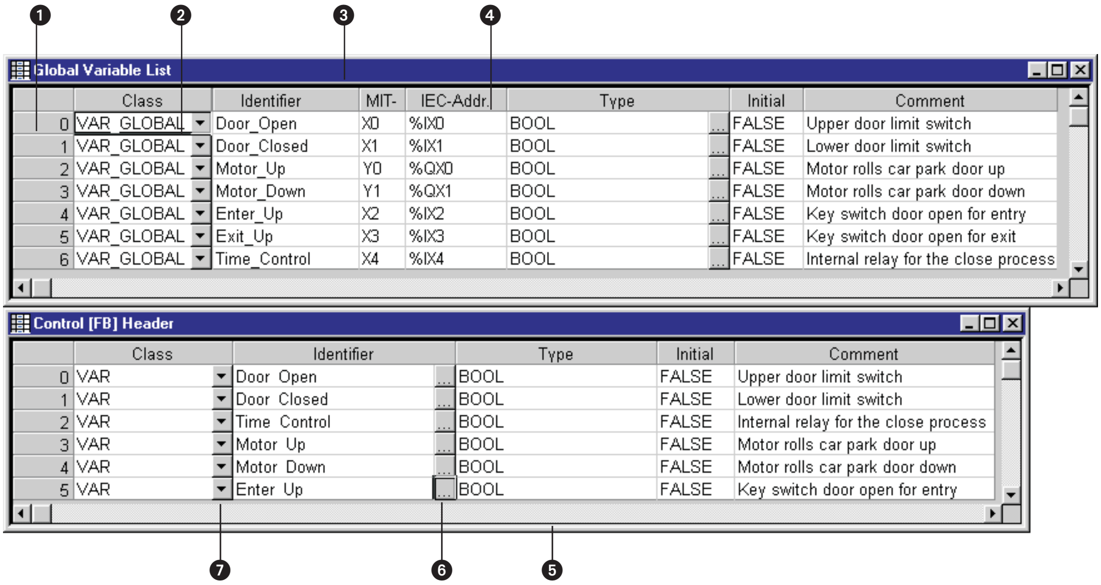
[1] *Single elements can't be dragged; additional element(s) will be selected automatically in the same way as with the Copy operation (i.e. if only a step is selected its related transition(s) and step(s) will be selected and dragged with it automatically).*

3.12 Working in tables

The following objects are managed and defined in tables:

- Global Variables
- Local Variables (Header)
- Data Unit Types (structured data types)
- Data Unit Variable Addresses
- Tasks

To open the table object window, double-click on the corresponding object in the Project Navigator window. If the object window of a body is already open you can open its header by selecting the **Open Header** command in the **Object** menu.



Global Variable List

	Class	Identifier	MIT-	IEC-Addr.	Type	Initial	Comment
0	VAR_GLOBAL	Door_Open	X0	%IX0	BOOL	FALSE	Upper door limit switch
1	VAR_GLOBAL	Door_Closed	X1	%IX1	BOOL	FALSE	Lower door limit switch
2	VAR_GLOBAL	Motor_Up	Y0	%QX0	BOOL	FALSE	Motor rolls car park door up
3	VAR_GLOBAL	Motor_Down	Y1	%QX1	BOOL	FALSE	Motor rolls car park door down
4	VAR_GLOBAL	Enter_Up	X2	%IX2	BOOL	FALSE	Key switch door open for entry
5	VAR_GLOBAL	Exit_Up	X3	%IX3	BOOL	FALSE	Key switch door open for exit
6	VAR_GLOBAL	Time_Control	X4	%IX4	BOOL	FALSE	Internal relay for the close process

Control [FB] Header

	Class	Identifier	Type	Initial	Comment
0	VAR	Door Open	BOOL	FALSE	Upper door limit switch
1	VAR	Door Closed	BOOL	FALSE	Lower door limit switch
2	VAR	Time Control	BOOL	FALSE	Internal relay for the close process
3	VAR	Motor Up	BOOL	FALSE	Motor rolls car park door up
4	VAR	Motor Down	BOOL	FALSE	Motor rolls car park door down
5	VAR	Enter Up	BOOL	FALSE	Key switch door open for entry

Fig. 3-23: *Variables tables*

Item	Description
❶	Cell with the line number
❷	Table cell
❸	Global Variables Table
❹	Table Column Headings
❺	Local Variables Table
❻	Up arrow (dialogue box)
❼	Down arrow (list box)

Tab. 3-9: *Variables table items*

You cannot change the number or the arrangement of the columns. When you activate a new table only line 0 is shown, in which you can enter the first variable.

NOTE

You can also copy all the entries from the global variables list to the header of a POU by using the clipboard. When you do this, the data fields are automatically placed in the correct columns.

3.12.1 Editing in tables

How to select a table cell

① Click on the cell you wish to select.
Or:
Select the cell with the cursor keys.

Selected cells are identified with an outline box.

How to select a table line

① Click on the cell containing the line number.
Or:
Select any cell in the line and press the key combination Home or Shift End.

The selected lines are highlighted in the display.

How to select multiple table lines

① Select the first line in the range to be selected:

Continuous range: Hold down the Shift key and click on the last line of the range to be selected.
Or:
Hold down the Shift key and extend the selection to the lines above or below the first selected line with the ↑ or ↓ keys.

Selecting groups of individual lines: Hold down the Ctrl key and click on the lines you wish to select.

The selected lines are highlighted in the display.

How to activate editing mode

① Double-click on the cell in which you wish to edit.
Or:
Select the cell and press the Space key.
Or:
Select a cell and type the text you wish to enter.

How to deactivate editing mode

① Click on another cell.
Or:
Press the Enter key.

How to open a list box or dialogue box

① Double-click on the symbol on the right hand side of the cell.

② Select the appropriate entry from the list box or dialogue box.

Or:

① Select a cell with a symbol at its right hand side.

② Press the key combination Alt↓.

③ Select the appropriate entry with the ↑ and ↓ keys, then press Enter to transfer the entry to the cell.

When editing mode is activated, the blinking text entry cursor appears in the cell and the background color of the cell changes.

NOTE | All the key combinations for working with tables are listed in the Appendix B.

How to change column widths of the text part

① Position the mouse pointer on the dividing line between the headings of the columns.

The pointer changes to a double-headed arrow symbol.

② Hold down the left mouse button and move the shaded column divider line so that the column has the desired width.

Or:

Double-click the row header near the cell borders to autosize columns to fit the widest text.

How to change column widths of the button part

Position the mouse pointer near the division between the text and the button part. (There is no dividing line between in the header row between the two parts.)

The pointer changes to a double-headed arrow symbol.

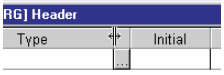


Fig. 3-24:
Double-headed arrow symbol

① Hold down the left mouse button and move the shaded column divider line so that the column has the desired width.

How to insert a new line

① Select **New Declaration** in the **Edit** menu.

② Select the position at which the new line is to be inserted.

Top	Inserts a new line at the top of the table
Before	Inserts a new line directly before the selected line/cell
After	Inserts a new line directly after the selected line/cell
Bottom	Inserts a new line at the end of the table

Or:

① Select any cell in the last line of the table.

② Press the key combination Shift Enter.

The line numbering is automatically updated after the insertion operation.

How to sort table entries

① Select **Sort** in the **Tools** menu.

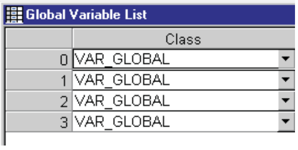
② Select the sort key to be used for the alphabetical sort procedure:

Type	Sort by data type
Class	Sort by class
Identifier	Sort by identifier

Or:

Click on the appropriate column header. The column header contains a sort indicator.

Example ▽



	Class
0	VAR_GLOBAL
1	VAR_GLOBAL
2	VAR_GLOBAL
3	VAR_GLOBAL

Fig. 3-25: *Sort indicator*

△

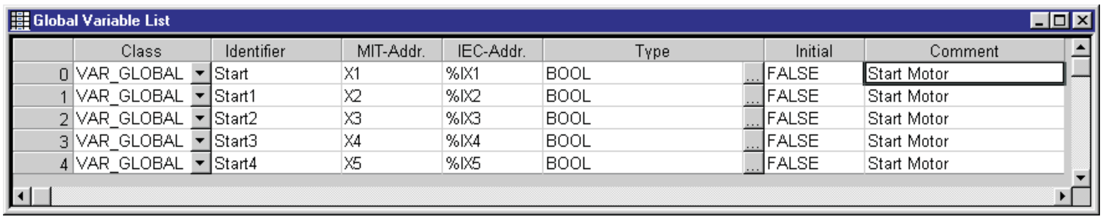
WARNING:
The table should not contain any empty lines.

NOTE

FUN and FB POU headers can't be sorted either using the Tools -> Sort menu or by clicking the appropriate column header. This functionality is disabled intentionally as the order of Input and Output variables for FUN and FB POUs is critical and can therefore not changed automatically.

3.12.2 Automatic copying of the contents of the preceding line into the new line (Autoincrement)

When a new line is entered in a declaration table the contents of the preceding line are automatically copied to the new fields. Identifiers and addresses are automatically incremented by 1.



	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial	Comment
0	VAR_GLOBAL	Start	X1	%IX1	BOOL	FALSE	Start Motor
1	VAR_GLOBAL	Start1	X2	%IX2	BOOL	FALSE	Start Motor
2	VAR_GLOBAL	Start2	X3	%IX3	BOOL	FALSE	Start Motor
3	VAR_GLOBAL	Start3	X4	%IX4	BOOL	FALSE	Start Motor
4	VAR_GLOBAL	Start4	X5	%IX5	BOOL	FALSE	Start Motor

Fig. 3-26: *Automatic copying and incrementing*

This function is only available when the **Incremental New Declarations** option in the **Extras - Options - Editing - Declaration Editor** menu is enabled.

If you also wish to include the contents of the Type and Comment fields you must also activate the **Copy Comment and Type Fields** option.

3.12.3 Export and Import of Excel files

The export and import of a spreadsheet to a file of Excel version, Excel 97, Excel 2000, or Excel 2002 (BIFF8 format), is available. Export will always save the whole spreadsheet. Importing will always append new declarations to the list, preserving any existing declarations.

Export to Excel

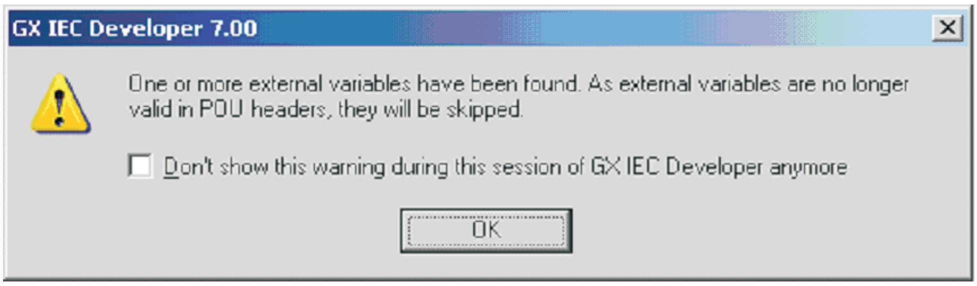
The export to Excel is done by selecting the **Export to Excel** command in the **Tools** menu.

Import from Excel

The import from Excel is done by selecting the **Import from Excel** command in the **Tools** menu.

During the import process the following checks are executed on the imported file:

- The imported file will be checked if it contains the columns required by the current editor. If none of the required columns are present, the import process will be aborted. If some columns are missing, you will be given the choice to continue, but the missing fields will be left empty for all imported declarations.
- If the imported file contains no declarations, the import process will be aborted.
- If the imported file contains empty rows, you will be given a chance to discard the empty rows, or import them anyway.
- All cells that contain longer text than is allowed by the actual field will be automatically truncated.
- The validity of the imported declarations themselves will not be checked (eg. for syntax errors, type mismatches, duplicate names, etc.).
- The imported file will be checked if it contains external variables. These variables will be skipped. A warning is shown to inform you.


GX IEC Developer 7.00

One or more external variables have been found. As external variables are no longer valid in POU headers, they will be skipped.

☐ Don't show this warning during this session of GX IEC Developer anymore

Fig. 3-27: *Import warning when external variables exist*

The affected editors and dialogues are:

- Task editor
- Data Unit Type editor
- Global Variable List editor
- POU Header editor
- Data unit variable addresses dialogue

3.12.4 Selecting data types in the declaration table

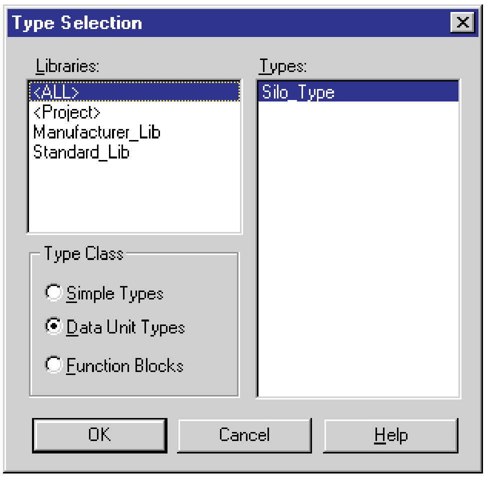


Fig. 3-28:
Data type selection

Libraries

All data types (incl. Libraries)

<Project> All data types in the project

Manufacturer_Lib Reserved for functions to be added later

Standard_Lib Reserved for functions to be added later

Type Class

Data type class

Simple Types: Simple data types

Data Unit Types: Structured data types (DUT)

Function Blocks: Reserved for functions to be added later

Types

All existing data types, depends on the selections in the previous list fields.

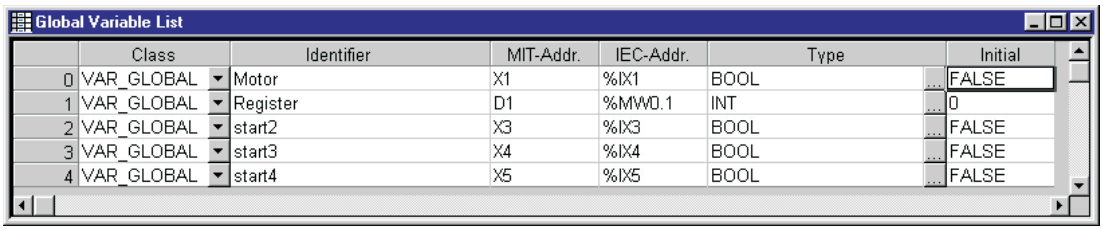
3.12.5 Automatic default data types in the declaration table

As soon as you enter an address the most likely data type for that address is automatically displayed in the Type column.

Example ▽

Entry: X1
Default type: BOOL

Entry: D1
Default type: INT



Global Variable List

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	Motor	X1	%IX1	BOOL	FALSE
1	VAR_GLOBAL	Register	D1	%MW0.1	INT	0
2	VAR_GLOBAL	start2	X3	%IX3	BOOL	FALSE
3	VAR_GLOBAL	start3	X4	%IX4	BOOL	FALSE
4	VAR_GLOBAL	start4	X5	%IX5	BOOL	FALSE

Fig. 3-29: *Default data types in declaration tables*

△

3.13 Working in the editors (Body)

The body contains the actual code of the PLC program. To open the body's object window, double-click on the **Body** entry in the Project Navigator window. If the object window of a header is already open you can open its body with the **Open Body** command in the **Object** window.

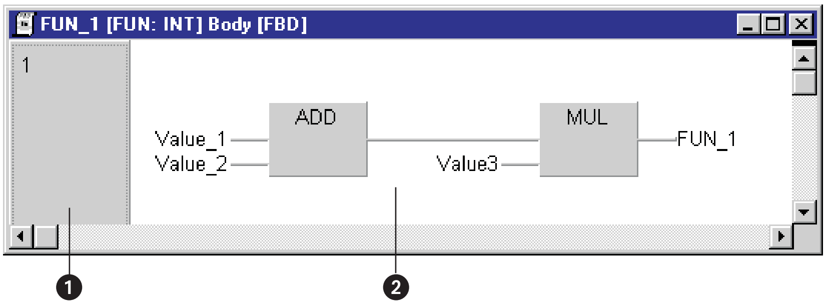


Fig. 3-30: *Program body in editor*

Item	Description
❶	Network bar
❷	Editing area

Tab. 3-10: *Body editor items*

3.13.1 Networks

In the Sequential Function Chart, Ladder Diagram and Function Block Diagram languages, the PLC program is always divided into a number of logical program units referred to as ‘networks’. These networks are numbered consecutively and can be given network labels which can be used as destinations for jump (goto) instructions.

> ***WARNING:***
> ***In the graphical editors (Ladder Diagram and Function Block Diagram) you can only program a maximum of one circuit in each network.***

How to enter a new network

① Select the **New Network** command in the **Edit** menu.

② Choose the position at which you wish to insert the network.

Top	At the beginning of the program
Before	Directly before the selected network
After	Directly after the selected network
Bottom	At the end of the program
MELSEC Before	Insert a MELSEC network directly before the selected network (only possible in Instruction List)

Or:

① Use the following tools:

Tool	Description	Menu Item	Shortcut
→▦	Inserts a new network directly before the selected network.	Edit - New Network - Before	Alt B
→▦	Inserts a new network directly after the selected network.	Edit - New Network - After	Alt A

Tab. 3-11: *Tools for inserting new networks*

Or:

① Insert new networks directly in the Network List.

NOTE

In MELSEC networks you can only use MELSEC programming instructions; you cannot use the IEC instructions. Also, you can only insert MELSEC networks in Instruction List programs (→ Appendix E).

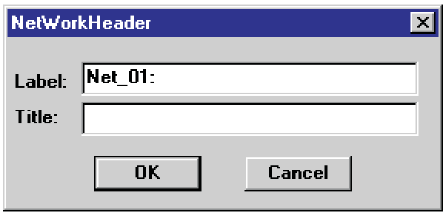
How to assign network labels

① Double-click on the network bar.
Or:
Activate selection mode, position the mouse pointer over the network bar with the cursor keys, and then press the Enter key.

A dialogue box is displayed in which you can enter the network label and title:

Label: The label must be terminated with a colon (:) and must consist of a maximum of 8 characters (including the colon). In MELSEC networks you can also assign absolute labels. An absolute label is a specific pointer address, e.g. P20.

Title: This entry is optional. The title can have up to 20 characters.



NetWorkHeader

Label: Net_01:

Title:

Fig. 3-31:
Assigning network labels and titles

NOTE Network labels and titles can also be entered directly in the Network List.

Network labels can be used as jump destinations in your PLC program, using the JMP operator in the Instruction List language, and with the jump instruction in the Ladder Diagram and Function Block Diagram languages.
Label1Label1Remember that every network label uses one system label and that the total number of system labels is limited to a maximum of 256. Since system labels are also needed for other elements in GX IEC Developer, you should only define labels to networks to which you actually intend to perform a jump.

How to make the network area larger/smaller

① In select mode, position the mouse pointer on the lower horizontal edge of the network bar.

The pointer changes to a double-headed arrow symbol.

② Hold down the left mouse button and drag the dividing line up or down as required.
Or:
Press the Space key. Hold down the Space key and drag the dividing line up or down with the ↑ and ↓ cursor keys.

When you release the mouse button or the Space key the network is displayed in the new size.

NOTE You do not normally have to change network size manually as it is adjusted dynamically; when you enter program elements or text the network is automatically enlarged as required.

Managing networks

The Network List is a powerful tool that gives you a clear overview of each POU’s networks, enabling you to manage them easily and efficiently.

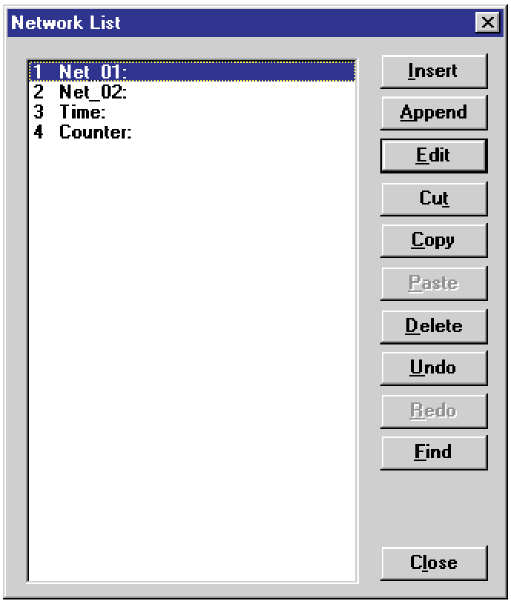


Fig. 3-32:
Network List

Select the **Network List** command in the **Tools** menu.

A list of all the networks with their labels and titles is shown on the left of the dialogue box displayed. The dialogue box contains the following buttons:

Insert: :
You will be prompted for the label and title of the new network. A new network is then inserted in the Network List and the body directly before the currently selected network.

Append:
You will be prompted for the label and title of the new network. A new network is then inserted in the Network List and the body as the last network.

Edit:
Edit the label and title of the currently selected network.

Cut:
Cuts the currently selected network, deleting it from the Network List and the body and inserting it in the clipboard.

Copy:
Copies the currently selected network to the clipboard.

Paste:
The network in the clipboard is inserted in the Network List and the body, directly before the currently selected network.

Delete:
Deletes the currently selected network without copying it to the clipboard. If you accidentally delete a network you can recover it by clicking on the **Undo** button. However, please note that Undo only works if you execute it immediately after performing the Delete command!

Undo:
Undoes the effects of the last command.

Redo:
Undoes the effects of the last Undo command.

Find:
Allows you to find specific network labels in your program. Please note that the search is case-sensitive and wildcard characters (? and *) are not supported.

Close:
Closes the dialogue box, accepting any changes made.

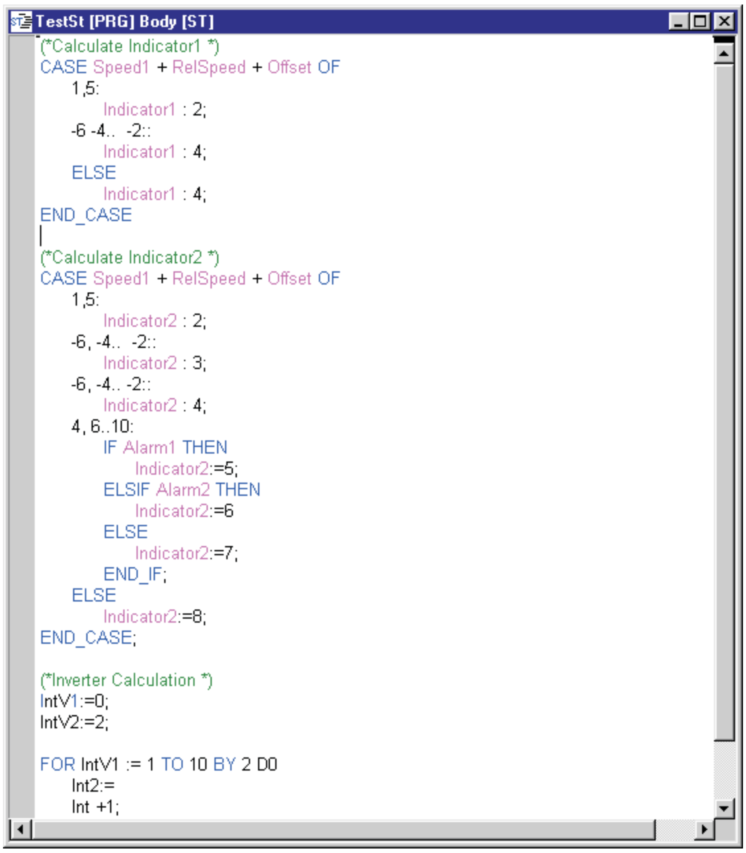
NOTE

The standard Windows key combinations for Delete, Cut, Copy and Paste are not supported in this dialogue box! You can only perform these operations with the buttons in the box. However, the key combinations are supported for editing labels and titles.

3.14 The ST editor

Operation of both the text editors – IEC Instruction List and MELSEC Instruction List – is similar to that of a normal word processor.

The Instruction List consists of a sequence of controller instructions. Each controller instruction must begin on a new line, and consists of an operator (operator, function or function block) and one or more operands/variables. The operator and the operand must be separated by a tab stop.



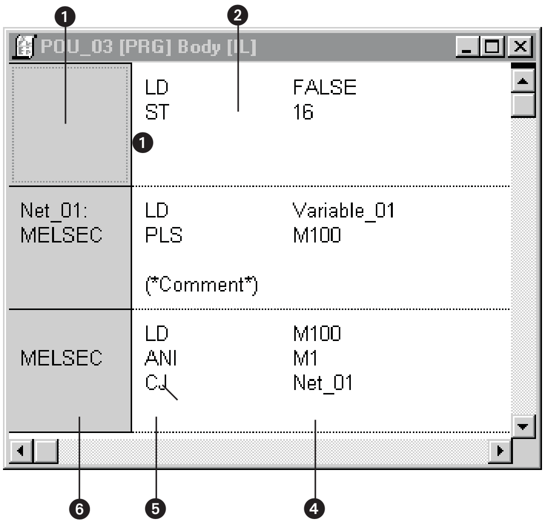
TestSt [PRG] Body [ST]

```
(*Calculate Indicator1 *)
CASE Speed1 + RelSpeed + Offset OF
    1,5:
        Indicator1 : 2;
    -6 -4..  -2::
        Indicator1 : 4;
    ELSE
        Indicator1 : 4;
END_CASE
|
(*Calculate Indicator2 *)
CASE Speed1 + RelSpeed + Offset OF
    1,5:
        Indicator2 : 2;
    -6, -4..  -2::
        Indicator2 : 3;
    -6, -4.. -2::
        Indicator2 : 4;
    4, 6..10:
        IF Alarm1 THEN
            Indicator2:=5;
        ELSIF Alarm2 THEN
            Indicator2:=6
        ELSE
            Indicator2:=7;
        END_IF;
    ELSE
        Indicator2:=8;
END_CASE;

(*Inverter Calculation *)
IntV1:=0;
IntV2:=2;

FOR IntV1 := 1 TO 10 BY 2 D0
    Int2:=
    Int +1;
```

Fig. 3-33: *ST Editor*



POU_03 [PRG] Body [IL]

```
                LD      FALSE
                ST      16

Net_01:         LD      Variable_01
MELSEC          PLS     M100

                (*Comment*)

                LD      M100
MELSEC          ANI     M1
                CJ      Net_01
```

Fig. 3-34:
IEC Instruction List in text editor

Item	Description
❶	Network bar
❷	Editing Area
❸	Comment
❹	Column for operand(s)
❺	Operator column
❻	MELSEC network

Tab. 3-12: *Text editor items*

NOTE

See the chapter 6 for details on programming.
A detailed description on how to insert programming instructions and variables is given at the end of this chapter.

All key combinations for working in Instruction List language are described in the Appendix B.

3.14.1 Syntax colouring

While you are typing text in the ST editor the entered text is coloured depending on the kind of input. The following groups are distinguished.

- Keywords

  AND
  BY
  CASE
  DO
  ELSE
  ELSIF
  END_CASE
  END_FOR
  END_IF
  END_REPEAT
  END_WHILE
  EXIT
  FALSE
  FOR
  IF
  MOD
  NOT
  OF
  OR
  REPEAT
  RETURN
  THEN
  TO
  TRUE
  UNTIL
  WHILE
  XOR

- Operators

  -
&
*
**
/
+
<
<=
=
>
>=
;

- Comments with the following syntax:

  (* <comment> *)

- Constant Numbers

IEC	Mitsubishi	Type
400000, -100	K400000, K-100	decimal
16#FFFF	HFFFF	hexadecimal
8#0745		octal
2#01010101		binary
1.2E-10	E1.2-10	REAL
T#3h10s3ms		TIME

Tab. 3-13: *Constant Numbers*

IEC constants could contain single underscore characters between the digits of a numeric literal (e.g. 1_2_3.4_5 for a REAL constant). IEC typed literals like DINT16#FF are not supported by GX IEC Developer.

NOTE TRUE and FALSE are coloured like other keywords.

- Constant Strings

IEC	Mitsubishi
‘abc’	“abc”

Tab. 3-14: *Constant Strings*

- Variables

Any variable or variable constant which is declared in the POU header.

Each group has a predefined color which can be changed by the user. Text, which does not belong to one of the above groups, is displayed in the editor text color which could be changed under **View** → **Color** → **Text**.

Color setting for syntax colouring

The color settings are valid for the current GX IEC Developer 7.01 project only. They are available as new menu items under **View** → **Colors** with the following menu item text:

- ST Keywords…
- ST Operators…
- ST Comments…
- ST Constant Numbers....
- ST Constant Strings…
- ST Global Variables…
- ST Local Variables…

Default foreground colors:

	RGB	Color
Keywords	(0, 0, 255)	🟦
Operators	(0, 0, 0)	⬛
Comments	(0, 128, 0)	🟩
Constant Numbers	(0, 0, 0)	⬛
Constant Strings	(0, 0, 0)	⬛
Global/Local Variables	(255, 0, 255)	🟪

Tab. 3-15: *Default foreground colors*

If the menu item **View** → **System Defaults** is chosen, all syntax colors, which have the same color as **View** → **Colors** → **Text** are changed to the Window setting for the foreground color of a window.

3.14.2 Options in structured text

By the help of a seperat dialogue the following options for the ST-editor can be configurated:

Auto Indentation

After pressing <RETURN> in the ST-editor (in edit mode), tabulators are automatically inserted at the beginning of the new line, if tabulators are existing at the beginning of the current line.

The number of tabulators is automatically increased, when it is expected (e.g. after keyword FOR).
The number of tabulators is automatically decreased, when it is expected (e.g. after keyword END_FOR).

This behaviour can be enabled or disabled in the options dialogue.

Auto identifier proposal

If you enter an identifier, the system searches in the corresponding header of the POU for a variable or a variable constant which starts with the already entered characters. The compare is done case insensitive.

If only one identifier exists, it is automatically inserted and selected. The selected part can be overwritten or inserted by pressing <RETURN>. If you press <RETURN>, the cursor will be placed behind the inserted identifier.

If several identifiers are existing, a list of all identifiers is shown which match the already entered characters. You then can select one from the list or add additional characters.

If you confirm a suggested identifier with <RETURN> or select one from the list of identifiers the already entered characters are deleted and the selected identifier is inserted. Therefor it is possible to enter characters in any case and, after an identifier is selected, it is inserted as it is written in the POU header.

The auto identifier proposal can be enabled or disabled in the options dialogue.

ToolTip on a variable-identifier

See section 3.19 for details.

Right mouse button

If the right mouse button is pressed, a context sensitive menu is shown.

Line number in the status bar

The current line number is shown in the status bar at the very left side in the format “Line: nnn” where nnn is the current line number. The first line has the line number 1.

This behaviour can be enabled or disabled in the options dialogue.

ST Options

An additional property is available to configure the following options:

Option	Default
Auto indentation	ON
Auto identifier proposal	ON
Line number in the status bar	ON
Tab size (1 - 64)	4

Tab. 3-16:
ST Options

See section 2.7.12 for the design of the dialogue.

See section 3.19 for the tool tip configuration.

Background color

The background color of the editor (in edit mode) is the same as for all editors. It can be changed with **View** → **Colors** → **Background**.

To make it obvious for the user, that the monitoring mode was activated, all colors of the syntax colouring groups are changed to the current text color when the monitoring mode is switched on.

3.14.3 Monitor functionality in ST editor

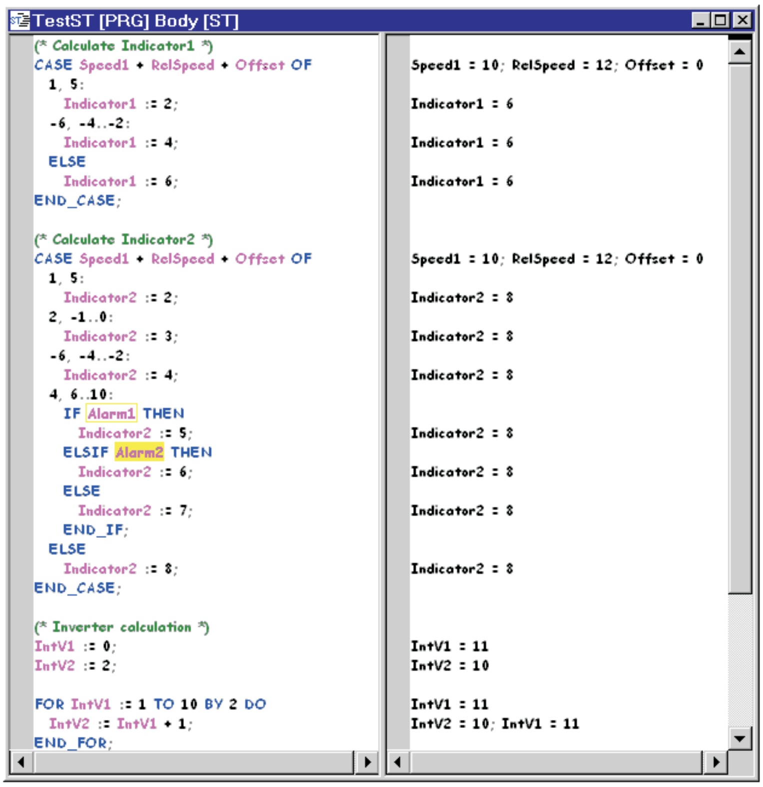
There a three different monitoring possibilities in the ST editor.

ToolTip:
See section 3.19 for details.

Monitor boolean values:
The monitoring of boolean values in the source code window is done like in the IL editor. This item is invoked by **Online** → **Start Monitoring** → **Body**.

A box is painted around a boolean variable or address in the monitor color. If the monitored value is ON the box background is set to the monitoring color, if it is OFF it is shown as the current background color of the editor.

Splitted body:
In addition to the monitoring of boolean values in the source window the ST body could be split into source code window (left part) and variables / values window (right part) with configurable border:



```
(* Calculate Indicator1 *)
CASE Speed1 + RelSpeed + Offset OF
  1, 5:
    Indicator1 := 2;
  -6, -4..-2:
    Indicator1 := 4;
  ELSE
    Indicator1 := 6;
END_CASE;

(* Calculate Indicator2 *)
CASE Speed1 + RelSpeed + Offset OF
  1, 5:
    Indicator2 := 2;
  2, -1..0:
    Indicator2 := 3;
  -6, -4..-2:
    Indicator2 := 4;
  4, 6..10:
    IF Alarm1 THEN
      Indicator2 := 5;
    ELSIF Alarm2 THEN
      Indicator2 := 6;
    ELSE
      Indicator2 := 7;
    END_IF;
  ELSE
    Indicator2 := 8;
END_CASE;

(* Inverter calculation *)
IntV1 := 0;
IntV2 := 2;

FOR IntV1 := 1 TO 10 BY 2 DO
  IntV2 := IntV1 + 1;
END_FOR;
```

```

Speed1 = 10; RelSpeed = 12; Offset = 0

Indicator1 = 6

Indicator1 = 6

Indicator1 = 6



Speed1 = 10; RelSpeed = 12; Offset = 0

Indicator2 = 8

Indicator2 = 8

Indicator2 = 8


Indicator2 = 8

Indicator2 = 8

Indicator2 = 8


Indicator2 = 8



IntV1 = 11
IntV2 = 10

IntV1 = 11
IntV2 = 10; IntV1 = 11
```

Fig. 3-35: *Split Window*

This item is invoked by Online \ Start Monitoring \ Split Window.

Modify Variable

By a double click on a variable name or by pressing the shortcut Ctrl + F9 when the edit cursor is positioned in a variable name the value of the variable could be changed in the source code window or variables / value window like in IL with functionality ‘ Modify Variable Value’. The related menu item is **Online** → **Modify Variable Value**.

By pressing the shortcut Alt + F9 on a boolean variable the value is toggled. The related menu item is **Online** → **Toggle Variable**.

Show MELSEC Code Of Selection

In addition to the menu item ‘Show MELSEC Code Of Network’ the menu item ‘Show MELSEC Code Of Selection’ is available. The MELSEC codes of the current selected lines are shown in an extra window similar to the ‘Show MELSEC Code Of Network’.

The header off the window will show the selected lines. For example:

Example ▽

(* TestST [PRG] *)

(* MELSEC code of TestST, line 4 - 11 *)

(* Code generated at 17.07.02 09:16:56, 7 steps *)

△

It is not possible in the editor to select independent lines. All statements are taken completely, even when only a part of a statement is selected.

Tracking

The tracking functionality is available in monitor mode.

Bookmarks

With the menu item **Edit** → **Toggle Bookmark** (Ctrl F7) a bookmark could be set or reset.

With the menu item **Edit** → **Next Bookmark** (F7) the cursor is positioned at the start of the line of the next bookmark. If the line is currently not visible it is scrolled into view.

With the menu item **Edit** → **Previous Bookmark** (Shift F7) the cursor is positioned at the start of the line of the previous bookmark. If the line is currently not visible it is scrolled into view.

With the menu item **Edit** → **Delete** all Bookmarks all bookmarks are deleted.

Bookmarks are saved to the project database (max. 100 items) when the body is saved. This allows the reuse of set bookmarks at the next edit session.

3.14.4 Limitations

The possible data type for Mitsubishi K and H constants is calculated for all IEC editors depending on their device values:

Device Value			IEC Data Type
K-32768	-	K-1	INT I DINT
K0	-	K32767	INT I WORD I DINT I DWORD
K32768	-	K2147483647	DINT I DWORD
K-2147483648	-	K-32769	DINT
H0	-	HFFFF	INT I WORD I DINT I DWORD
H10000	-	HFFFF FFFF	DINT I DWORD

Tab. 3-17: *Calculation of constants*

Constant Value			IEC Data Type
-2147483648	-	-32769	DINT
-32768	-	-1	INT I DINT
0	-	32767	INT I WORD I DINT I DWORD
32768	-	65535	DINT I WORD I DWORD
65536	-	2147483647	DINT I DWORD
2147483648	-	4294967295	DWORD
16#0	-	16#FFFF	INT I WORD I DINT I DWORD
16#10000	-	16#FFFF FFFF	DINT I DWORD

Tab. 3-18: *Calculation of constant values*

As in former GX IEC Developer versions the Mitsubishi devices FX, FY and FD are not supported in GX IEC Developer 7.01. Indirect devices like @D0 can also not be used except in MELSEC IL networks.

As in former GX IEC Developer versions for the following MELSEC instructions no Manufacturer Library function are defined in GX IEC Developer 7.01:

LD	LDI	AND	ANI	OR	ORI	ANB
ORB	NOP	NOPLF	PAGE	CALL	CALLP	ECALL
ECALLP	EFCALL	EFCALLP	FCALL	FCALLP	XCALL	

To start a timer in GX IEC Developer the functions TIMER_M or TIMER_H_M must be used instead of the MELSEC instructions OUT or OUTH with two operands.

3.14.5 Edit functions

How to activate editing mode

There are two ways to activate editing mode:

- By double-clicking in the editing area, or
- By clicking on the Edit Network tool:

Edit Network tool

The entry cursor is then displayed in the editing area.

How to change the column width

① In select mode, position the mouse pointer on the vertical line between the columns.

The pointer changes to a double-headed arrow.

② Hold down the left mouse button and drag the dividing line to the left or right as required.
Or:

Hold down the Space key and move the dividing line left or right with the ← and → cursor keys.

When you release the mouse button or the Space key the dividing line is redisplayed in the new position.

How to enter comments

① Position the cursor at the point where you wish to enter the comment.

You can insert comments anywhere in the Instruction List, i.e. in the third column or in a new line.

② Comment text must always be enclosed by the characters (* and *).

How to insert a programming instruction and a variable

→ page 3-69 and 3-71

3.15 The graphical editors

Operation of the two graphical editors – Ladder Diagram and Function Block Diagram – is comparable to working in a standard graphics program. The program elements used for writing PLC programs are provided in the toolbar and the **Tools** menu.

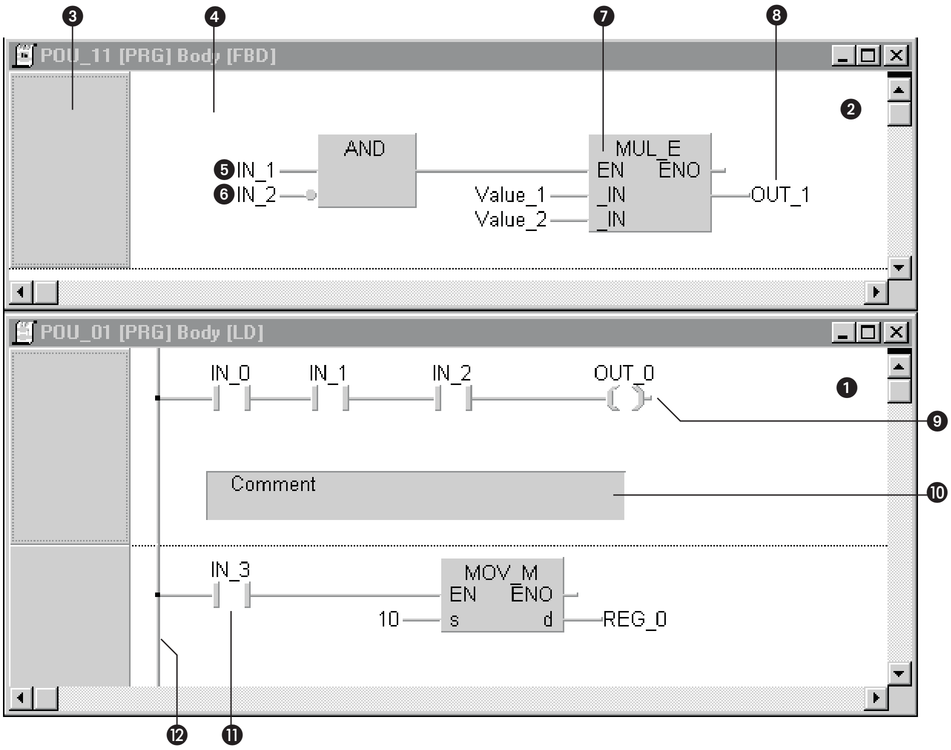


Fig. 3-36: Programs in graphical editor

Item	Description
❶	Ladder Diagram (LD)
❷	Function Block Diagram (FBD)
❸	Network bar
❹	Editing area
❺	Input variable (poll for signal status ‘1’)
❻	Input variable (negated, poll for signal status ‘0’)
❼	Function
❽	Output variable
❾	Output coil (Ladder Diagram only)
❿	Comment
⓫	Input contact (poll for signal status ‘1’) (Ladder Diagram only)
⓬	Power bar (Ladder Diagram only)

Tab. 3-19: Graphical editor items

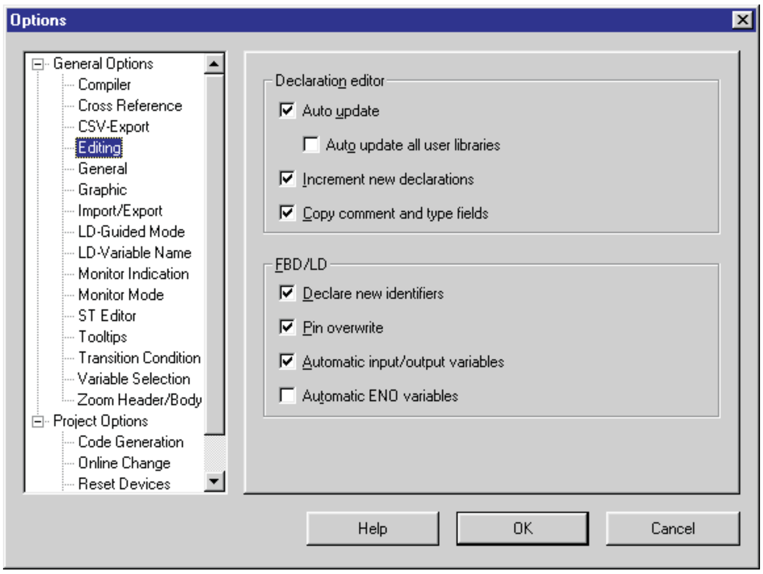
NOTES

See the chapter 6 for details on programming.
A detailed description on how to insert programming instructions and variables is given at the end of this chapter.

All key combinations for working in graphical editors are described in the Appendix B.

3.15.1 Options

In the menu **Extras - Options** you can specify options which make handling operations in the graphic editors easier.



Declaration editor

☑ Auto update

☐ Auto update all user libraries

☑ Increment new declarations

☑ Copy comment and type fields

FBD/LD

☑ Declare new identifiers

☑ Pin overwrite

☑ Automatic input/output variables

☐ Automatic ENO variables

Fig. 3-37: Extras - Options

FBD/LD

Declare New Identifiers: When you overwrite a dummy variable in a graphical editor with a variable identifier name that has not yet been declared the **Variable Selection (Mode NewVar)** dialogue box will be displayed automatically so that you can enter the declaration for the new variable.

Pin Overwrite: When you create a connection to an empty variable connection in a graphical editor the empty variable will be overwritten.

Automatic Input/Output Variables: The system will automatically create the input and output variables when you insert a function block in a graphical editor.

Automatic ENO Variables: The system will automatically create the ENO output variable when you insert a function block in a graphical editor. If the ENO output variable is not needed this option has to be disabled.

3.15.2 Context menu

When edit mode is active you can display the context menu by clicking the right mouse button or pressing F2.

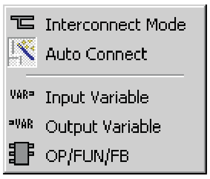


Fig. 3-38:
Editor context menu

Line / Arrow (toggle function): Toggles between select mode and interconnect mode.

AutoConnect: Makes positioning and moving graphical elements simpler. When elements are moved the connection lines are redrawn automatically (→ page 3-59)

Mouse Pointer	Purpose
↖	Select Mode
✎	Interconnect Mode
↖	Select Mode + AutoConnect
✎	Interconnect Mode + AutoConnect
↖	Guided Mode (see chapter 7)

Tab. 3-20: *Editor mouse pointers*

VarInput Pin: Insert input variable (same as **Tools - Input Variable**)

Contact (LD only): Insert input contact (same as **Tools - Contact**)

VarOutput Pin: Insert output variable (same as **Tools - Output Variable**)

Coil (LD only): Insert output coil (same as **Tools - Coil**)

FUN/FB: Opens **Function Block Selection** dialogue box for programming instruction selection

Last 5 function blocks list: The last 5 function blocks inserted in the LD or FBD editors can be reused directly by selecting them from the list at the bottom of the context menu.

3.15.3 Right mouse button or F2

Mouse pointer in editing mode / on block: Activates context menu

Mouse pointer on variable: Opens the **Variable Selection** dialogue box for selection of a variable from the list.

Mouse pointer on a jump instruction: Opens the Label List dialogue box for selection of a network label from the list.

3.15.4 Program Elements

Tool	Program element	Menu Command	Key(s)
	Lines (Interconnect Mode)	Tools - Network Elements - Interconnect Mode	Ctrl T
1	Input contact (Ladder Diagram only)	Tools - Network Elements - Contact	1
2	Input contact negation (Ladder Diagram only)	Tools - Network Elements - Contact Negation	2
3	Input L-contact (Ladder Diagram only)	Tools - Network Elements - L-Connection Contact	3
4	Input L-contact negation (Ladder Diagram only)	Tools - Network Elements - L-Connection Contact Negation	4
5	Vertical line (Ladder Diagram only)	Tools - Network Elements - Vertical Line Segment	5
6	Horizontal line (Ladder Diagram only)	Tools - Network Elements - Horizontal Line Segment	6
7	Output coil (Ladder Diagram only)	Tools - Network Elements - Coil	7
8	Function block (programming instruction)	Tools - Network Elements - Function block	8
VAR= 9	Input variable	Tools - Network Elements - Input Variable	9
=VAR 0	Output variable	Tools - Network Elements - Output Variable	0

Tab. 3-21: Program elements

How to insert new program elements

Icons are provided in the toolbar for all the available program elements.

① Click on the tool for the element you wish to insert.
Or:
Select the menu command from the **Tools** menu, or press the corresponding key combination.

The mouse pointer then changes to the symbol for the selected program element.

② Position the tip of the pointer at the point where you want to insert the element and press the left mouse button or the Enter key.

How to select program elements

① Click on the program element.
Or:
Position the pointer over the program element with the cursor keys and then press the Space key.

Selected program elements are highlighted in a different color.

How to select more than one program element

- Selecting elements out of sequence: Hold down the Shift key and click on all the program elements you wish to select.
  Selecting elements in sequence: Position the mouse pointer outside the area to be selected, hold down the left mouse button and drag the dotted rectangle around the elements you wish to select.

  Or:

- Selecting elements out of sequence: While holding down the Shift key, select the elements one after another by positioning the mouse pointer over them with the cursor keys and pressing Space.
  Selecting elements in sequence: Position the pointer outside the area to be marked. While holding down the Space key, use the cursor keys to drag the dotted rectangle around the program elements you wish to select.

Selected program elements are highlighted in a different color.

NOTE You can only select multiple program elements within the same network.

How to increase and decrease editing area spacing

The **Tools - Open Row/Open Column** commands are used to increase and decrease the horizontal and vertical spacing in the editing area.

① Select the menu command or the corresponding tool in the toolbar.

The pointer changes to a horizontal or vertical double-headed arrow.

② Position the pointer at the location where you wish to increase/decrease spacing.

③ Increase spacing: Click with the mouse button or press the Space key.
Decrease spacing: Hold down the Shift key while clicking or pressing the Space key.

How to enter comments

You can enter comments at any point in the editing area.

① Select **Comment** in the **Tools** menu or the corresponding tool in the toolbar.

The pointer changes to a comment text symbol.

② Move the pointer to the position where you wish to insert the comment.

③ Hold down the left mouse button and drag to create a comment field.
Or:
Hold down the Space key and create a comment field with the cursor keys → and ↓.

When you release the mouse button or the Space key the blinking editing cursor appears in the newly-created comment field.

④ Enter your comment text.

How to insert a programming instruction

→ page 3-69

3.15.5 AutoConnect

The AutoConnect function can be activated in the **Tools** menu or the **context menu**.

The AutoConnect mode makes moving and connecting graphical elements much easier. When you move the elements the connecting lines are stretched and rerouted automatically, so that you do not need to edit the lines by hand. To connect elements in the AutoConnect mode you simply click on the start and end points and the lines are drawn automatically.

The mouse pointer has the following forms in **AutoConnect** mode:

Mouse Pointer	Purpose
↖	Select Mode + AutoConnect
✎	Interconnect Mode + AutoConnect

Tab. 3-22: *Editor mouse pointers*

Moving elements (rubber banding)

① Activate both select mode and AutoConnect.

② Click on the element you wish to move and drag it to the new position, holding the left mouse button down.

③ Release the mouse button when you reach the position at which you wish to place the element.

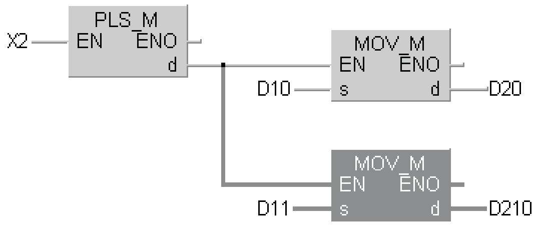


Fig. 3-39:
The lower block MOV_M and its associated connecting lines are selected.

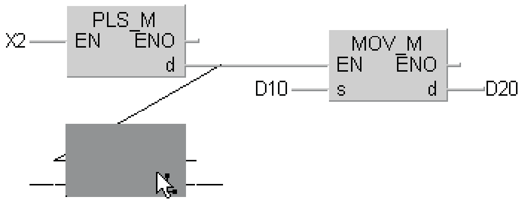


Fig. 3-40:
The block is dragged to its new position. Temporary diagonal connection lines are displayed while you are repositioning the block.

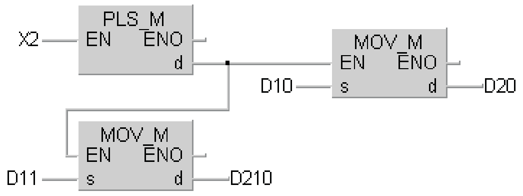


Fig. 3-41:
The block has been moved to its new position and the connecting lines have been redrawn automatically. The variables have also been moved automatically, thus retaining their positions relative to the function block.

If there is an anchor point on the connection line the lines will only be modified up to the position of the first anchor point. The lines on the other side of the anchor point and the anchor point itself all remain unchanged (→ page).

An error message will be displayed if you try to move an element to a position that GX IEC Developer cannot calculate. You can then abort the move operation.

The **Undo** function is available as well.

Inserting connection lines (autorouting)

① Activate both Interconnect mode and AutoConnect.

② Click on the position for the starting point. Possible starting points are indicated by black dots. This activates the connecting line, which follows the mouse pointer as you move it.

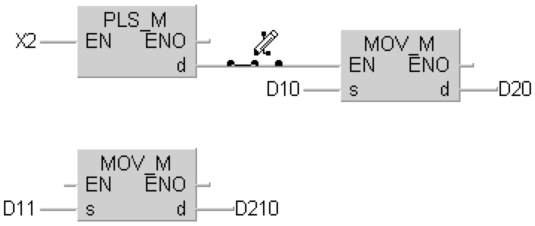


Fig. 3-42:
AutoConnect starting point

③ Position the mouse pointer at the end point.

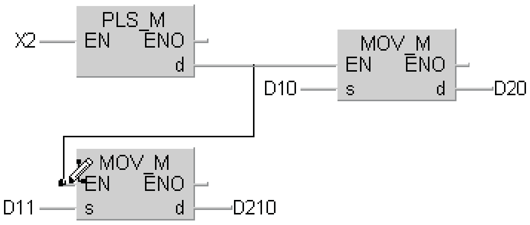


Fig. 3-43:
AutoConnect end point

④ Click on the end point.
The connecting line is drawn automatically.

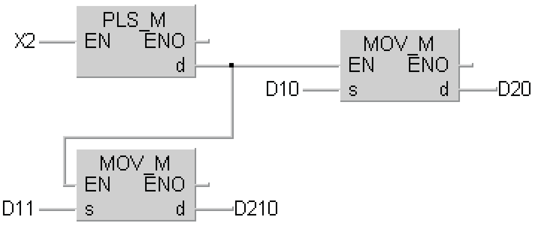


Fig. 3-44:
"AutoConnected" line

Recalculate Line

The menu command **Recalculate Line** in the **Tools** menu helps to recalculate lines which have already been drawn.

In the normal Interconnect Mode (AutoConnect is disabled) a connection with several lines has been created.

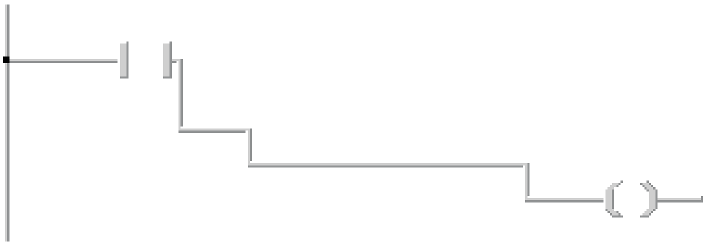


Fig. 3-45:
Connecting line before recalculation

① Activate the Interconnect Mode and AutoConnect.

② Click on the connection.
All lines of the connection will be marked.

③ Activate the command **Recalculate Line** in the **Tools** menu.
The connection is recalculated.

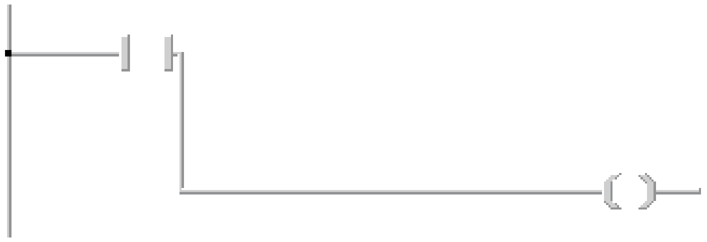


Fig. 3-46:
Recalculated connecting line

Possible starting and end points

Try moving the connecting line over the various elements in the editing window. As you do so black dots are displayed to indicate possible connection points. The connecting line is drawn to the black dot under the tip of the pencil symbol when you click the mouse button.

Examples of possible connection points:

- Variable connections (without variables) on function blocks
- Between variables and variable connections (without variables) on function blocks
- Ends of lines
- Positions on lines

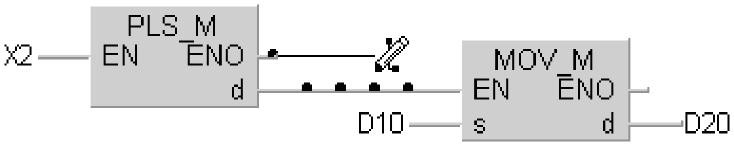


Fig. 3-47:
Starting and end points

The positions of the connection points are dictated by the grid raster.

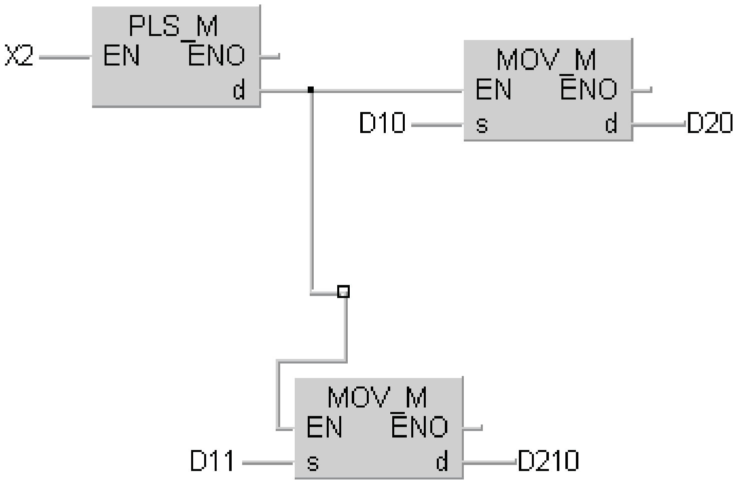
If you click on a position where no black connection dots are displayed the operation will be aborted automatically, thus preventing the inadvertent creation of connection lines without connected ends.

Anchor points

Double-clicking on a potential connection point creates an anchor point. Anchor points are indicated by small black squares.

When you move elements in the editing window the positions of the anchor points remain unchanged.

Fig. 3-48:
Anchor point



Overwriting unassigned variable pins

If the starting or end point of a new line is an unassigned variable connection (pin) the connection is overwritten automatically when the line is inserted. You can activate or deactivate this function with the **Extras - Options - Editing - Pin Overwrite** option.

NOTES

Restrictions for connecting lines
Do not try to insert connecting lines between variable pins for which different variable types have been declared. This will result in an error message when you perform a check on the POU.

Do not try to draw connecting lines through function blocks.

3.15.6 Variables

How to configure input signals

Double-click on an input contact or on the connection of the input variable in a function block.
Or:
Position the pointer with the cursor keys on an input contact or on the connection of the input variable in a function block and press the key combination **[CTRL]+[ALT]+[S]**.

A dialogue box is displayed with the following options:

Normal	Poll for signal status ‘1’
Negation	Poll for signal status ‘0’

Make the appropriate selection in the dialogue box.

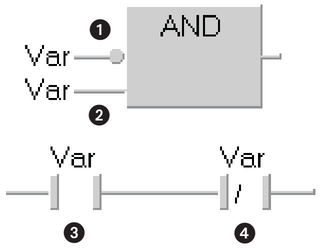


Fig. 3-49:
Input signal configurations:
Normal and negated

Item	Description
❶	Negated input variable
❷	Normal input variable
❸	Normal input contact
❹	Negated input contact

Tab. 3-23: *Signal configurations items*

An input signal can also be configured directly:

① Click on the input contact or a function block to select it.

② Press the key combination **[CTRL]+[ALT]+[S]** to negate the input or press the key combination **[CTRL]+[ALT]+[C]** to undo the negation.

Or:

① Click on the input contact or a function block to select it.

② In the menu **Edit** select **Toggle Signal Configuration** and click on the selected input. The signal configuration of this input will be toggle.
If you select **Edit - Signal Configuration**, after clicking on the selected input the **Signal Configuration** dialogue appears.

How to configure output signals

① Double-click on an output coil or on the connection of the output variable in a function block.
Or:
Position the pointer with the cursor keys on an output coil or on the connection of the output variable in a function block and press the key combination **[CTRL]+[ALT]+[S]**.

A dialogue box is displayed with the following options:

Normal	Poll for signal status ‘1’
Negation	Poll for signal status ‘0’
Set	Set (output coil only)
Reset	Reset (output coil only)

Make the appropriate selection in the dialogue box.

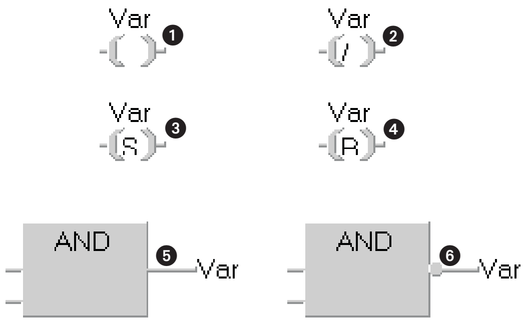


Fig. 3-50:
Output signal configurations: Normal and negated

Item	Description
❶	Normal output coil
❷	Negated output coil
❸	Set output coil
❹	Reset output coil
❺	Normal output variable
❻	Negated output variable

Tab. 3-24: *Signal configurations items*

How to assign input and output variables to programming instructions

① Select **Tools - Input Variable/Output Variable** or the corresponding tool in the toolbar.

The pointer changes to an input variable or output variable symbol.

② Position the tip of the pointer over the connection of the function block.

③ Click on this position.
Or:
Press the Space key.

The variable is inserted. You must now assign the variable name.

NOTE

Most programming instructions have a specific number of input variables. However, in some cases (e.g. ADD_E) the number of input variables can be increased. With these instructions the **Number of Pins** field is activated in the programming instruction dialogue box. Enter the required number (max. 28) in this field. Note that the EN input is not included in the total. For example, if you enter 5 connections for the ADD_E function the resulting Block will have 5 real connections for the addition operation and one EN input.

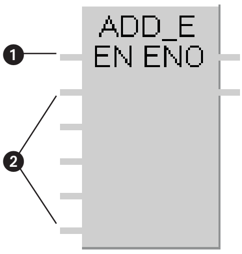


Fig. 3-51:
ADD_E function

Item	Description
❶	One EN input
❷	Five addition connections

Tab. 3-25: *Items of programming instructions*

How to insert a variable

→ page 3-62

Adding and removing input variables

① Select a function block for which the number of input variables can be changed.

There are variety of different ways to add/remove input variable pins:

- **Increment Pins / Decrement Pins** options in the **Edit** menu
- + and - keys in the numeric keypad
- Toolbar icons:

Increment

Decrement

- Position the mouse pointer on the lower border of the function block. When the pointer changes to a double arrow press the left button and drag down (increment) or up (decrement).

Incrementing

If the **Automatic Input/Output Box** option in **Extras - Options - Editing** is activated empty dummy variables are automatically inserted when you increment the number of variable pins.

Decrementing

The input variables will be deleted if they are still empty, i.e. if no variable has yet been assigned to them. However, if variables have already been assigned to them they will not be deleted.

How to insert input and output variables in the graphical editors automatically

① Activate the **Automatic Input/Output Box** option in **Extras - Options - Editing**.

When this option is activated the input and output variables are automatically inserted when you add a function block in the graphical editors. The dummy identifier ? (question mark) is inserted in the variable name field.

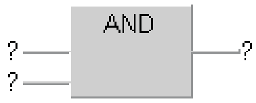


Fig. 3-52:
Function block

Deleting variables

① Select the variable’s connecting line.

② Press Del.

Or

① Select the question mark dummy or the variable name.

② Press Del.

③ Press the Tab key.

NOTE You do not need to delete the variables at the connecting pins before inserting lines in AutoConnect mode. They are overwritten automatically.

Stepping through variables in a network

Pressing Tab and Shift Tab steps through all the variables in the current network forwards or backwards, selecting them one after another.

3.15.7 Jump and Return Instructions

How to configure jump instructions (label and signal)

A jump instruction executes a branch to a specified network, using the network label as the destination.

① Click on the dummy label by the jump instruction symbol.
Or:
Position the pointer over the dummy label and press the Space key.

② Enter the jump destination name.

③ Double-click on the jump instruction symbol.
Or:
Position the pointer over the jump instruction symbol and press the Enter key.

A dialogue box is displayed with the following options:
Normal Poll for signal status ‘1’
Negation Poll for signal status ‘0’

④ Make the appropriate selection in the dialogue box.

NOTE

The jump instruction corresponds to the JMP operator in the Instruction List.

Operator (IL)	Program element (LD/FBD)
JMP	Normal jump (poll for signal status ‘1’)
JMPN	Negated jump (poll for signal status ‘0’)
JMPC	Normal jump with input condition
JMPCN	Negated jump with input condition

How to configure return instructions (signals)

The return instruction returns control from a function block to the program from which the function block instance was called.

① Double-click on the return instruction symbol.
Or:
Position the pointer over the return instruction symbol and press the Enter key.

A dialogue box is displayed with the following options:
NormalPoll for signal status ‘1’
NegationPoll for signal status ‘0’

② Make the appropriate selection in the dialogue box.

NOTE

The return instruction corresponds to the RET operator in Instruction List language (IL).

Operator (IL)	Program element (LD/FBD)
RET	Normal return instruction (Poll for signal status ‘1’)
RETN	Negated return instruction (Poll for signal status ‘0’)
RETC	Normal return instruction with input condition
RETCN	Negated return instruction with input condition

3.15.8 Graphic macros

Setting the number of graphic macros

You can create and store up to 25 graphic macros.

Number of Graphic macros in Extras - Options - Graphic
Enter the number of graphic macros you wish to use.

How to create graphic macros

① Select the area in the editing area that you wish to store as a graphic macro.

The selected area is highlighted.

② Select the **Copy** command in the **Edit** menu.

The selected area is copied to the clipboard.

③ Select **Define Graphic Macros** in the **Tools** menu.

The **Configure Custom Tools** dialogue box appears on the screen.

④ Select the name under which you wish to store the macro.

If a macro is already stored under the selected name its contents are shown in the **Contents** box.

⑤ Click on the **Paste** button.

The contents of the clipboard are pasted into the **Contents** box.

⑥ Click on the **Close** button to confirm your entries and close the dialogue box.

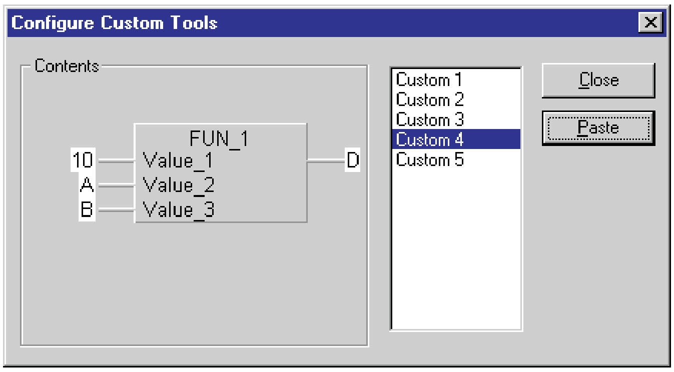


Fig. 3-53:
Creating graphic macros

Selecting graphic macros

Tools - Custom: The names of the first five graphic macros are displayed in the menu and they can also be selected by pressing Alt together with the corresponding number key.

Tools - More Custom: Enables you to select all available graphic macros.

3.16 Selecting Programming Instructions

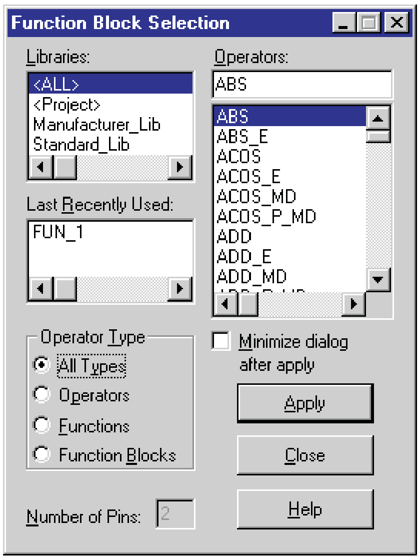
Graphical editors:

- Menu: **Tools - Network Elements - Function Block**
- Toolbar icon
- FUN/FB context menu (right mouse key)

Text editors:

- Menu: **Tools - List Operators/Operands**
- F2 key
- Right mouse key

The **Function Block Selection** dialogue box is displayed. It remains open, active and visible until it is explicitly closed. It is also closed automatically when you close the current body or activate another window on the screen.



Operator Type: ● All Types ○ Operators ○ Functions ○ Function Blocks

□ Minimize dialog after apply

__Fig. 3-54:__
Function Block Selection

Libraries

: All programming instructions available for the PLC currently selected.

<Project>: Functions and function blocks created by the user (depends on the selection in the Operator_Type field)

Manufacturer_Lib: All programming instructions in the manufacturer library

Standard_Lib: All programming instructions in the standard library

Last recently used

All recently-used programming instructions

Operator Type

All Types: IEC operators, functions and function blocks

Operators: IEC operators

Functions: Functions

Function Blocks: Function blocks

Number of Pins

This field is activated when the instruction allows you to change the number of input contacts (input pins).

Operators

All available programming instructions (depends on the selections in the previous fields). If you enter the first letters of an instruction manually the highlight automatically jumps to the first instruction beginning with that letter or letter combination.

APPLY button

Graphical editors: When you move the mouse pointer over the editing window it changes to a pointer with a function block symbol. Simply click at the position where you wish to insert the instruction.

Text editors: Enters the selected programming instruction in the editor window.

Getting help on specific programming instructions

Select the instruction in the Operators list field and then press the Help button.

3.17 Variables

3.17.1 Selecting existing variables and declaring new variables

Before you open the **Variable Selection** dialogue box:

Graphical editors: Highlight the variable/instance dummy name

Text editors: Place the cursor in the variables column

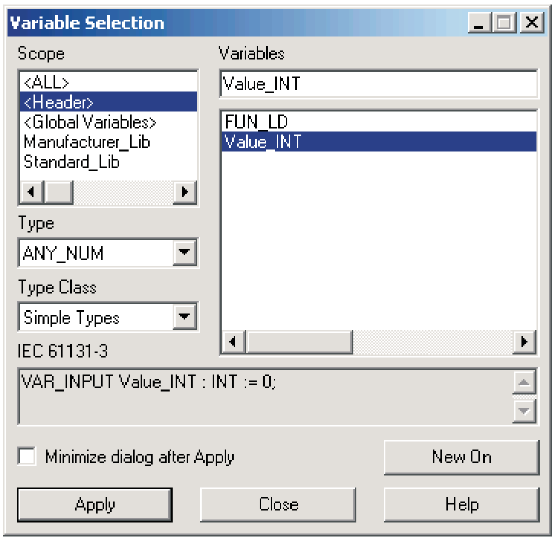
Sequential function chart language: Highlight a transition definition

Then:

In the Task Information dialogue box: Event (F2 key only)

In both graphical and text editors:
- **Tools - List Operators/Operands** menu
- F2 key
- Right mouse button

The **Variable Selection** dialogue box is then displayed. This dialogue remains open and visible on the screen and does not need to be closed when you perform other editing actions in the program body. If you prefer to minimize the dialogue box automatically after applying a declaration, activate the “Minimize Dialogue after apply” checkbox.



Scope: <Header>

Variables: Value_INT

Type: ANY_NUM

Type Class: Simple Types

```
VAR_INPUT Value_INT : INT := 0;
```

□ Minimize dialog after Apply

Fig. 3-55:
Variable Selection

Scope

All of the project’s variables:

<Header>: All local variables in the header of the currently open POU

<Global Vars>: All global variables

Manufacturer_Lib: Reserved for functions to be added later

Standard_Lib: Reserved for functions to be added later

Type

Depends on the selections in the Libraries and Type Class list fields:

Simple Types: Simple data types: ANY, ANY_BIT …

Data Unit Types: Structured data types (DUT)

Function Blocks: Function blocks

For existing variables the currently valid type is displayed.

Default settings:
- Graphic editor: Data type of formal parameter, contact/coil: BOOL
- Transition in SFC: BOOL

NOTE

The variable selection dialogue includes a variable type filter.
The type list only includes variable types valid for the actual project and does not include variable types that are not available for the used PLC type.

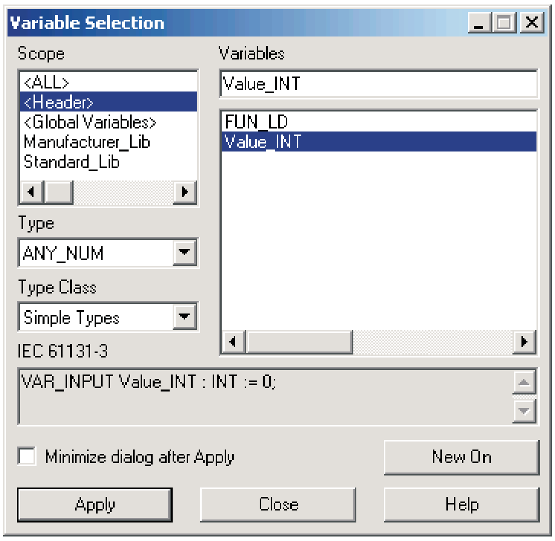
Type Class

Simple Types: Simple data types

Data Unit Types: Structured data types (DUT)

Function Blocks: Function blocks

For existing variables the currently valid type class is displayed.



Variable Selection

Scope

<ALL>
<Header>
<Global Variables>
Manufacturer_Lib
Standard_Lib

Variables: Value_INT

FUN_LD
Value_INT

Type: ANY_NUM

Type Class: Simple Types

IEC 61131-3

VAR_INPUT Value_INT : INT := 0;

□ Minimize dialog after Apply

Fig. 3-56:
Selecting a variable

***Type Class:** Simple Types*
***Type:** BOOL*
***Variables:** The variables are displayed. Doubleclick on the variable to select it.*

Variables

Depends on the selections in the preceding list fields: All existing variables

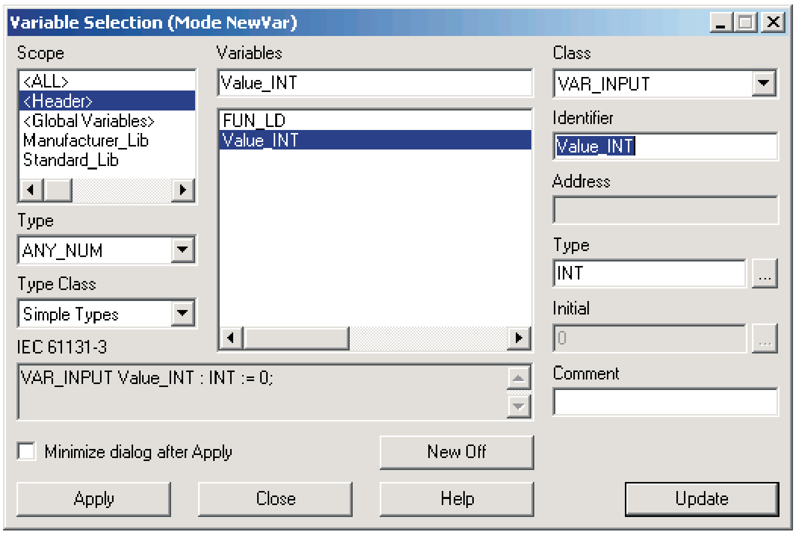
When you enter the first letters of the variable name the highlight will automatically jump to the first variable with this letter or letter combination.

APPLY button

Enters the selected variable into the selected variable dummy name in the editor (same effect as double-clicking).

NEW ON button

Selecting this button expands the dialogue box, displaying an additional area for the declaration of new variables. The dialogue box title changes to **Variable Selection (Mode NewVar)**.



Variable Selection (Mode NewVar)

Scope

<ALL>
<Header>
<Global Variables>
Manufacturer_Lib
Standard_Lib

Variables: Value_INT

FUN_LD
Value_INT

Type: ANY_NUM

Type Class: Simple Types

IEC 61131-3

```
VAR_INPUT Value_INT : INT := 0;
```

Class: VAR_INPUT

Identifier: Value_INT

Address:

Type: INT

Initial: 0

Comment:

☐ Minimize dialog after Apply

Fig. 3-57:
Variable Selection in new variable mode

This dialogue box also appears:

- When you select the **Tools - New Variable** menu option (ALT + N).
- When you overwrite a dummy variable name in a graphical editor with the name of a variable that has not yet been declared. This automatic function can be activated/deactivated with the option **Extras - Options - Editing - Declare New Identifiers**.

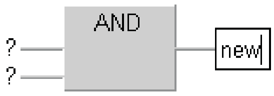


Fig. 3-58:
Dummy variable name

The expanded dialogue box fields enable you to declare new variables of the type defined by the selections in the **Library, Type** and **Type Class** fields. The entries are the same as those in the variable declaration tables.

You can also declare instances, in the same way as in the declaration tables.

DEFINE or UPDATE button

Selecting this button confirms and accepts all your entries for the new or existing variable. The button is inactive (greyed out) if your entries are incorrect or incomplete.

The new variable or the changes on a existing variable are automatically added to the corresponding declaration table.

NEW OFF button

Shrinks the dialogue box back to its original size, hiding the new variable declaration fields again.

IEC 61131-3

This field displays all the available IEC information for the selected variable.

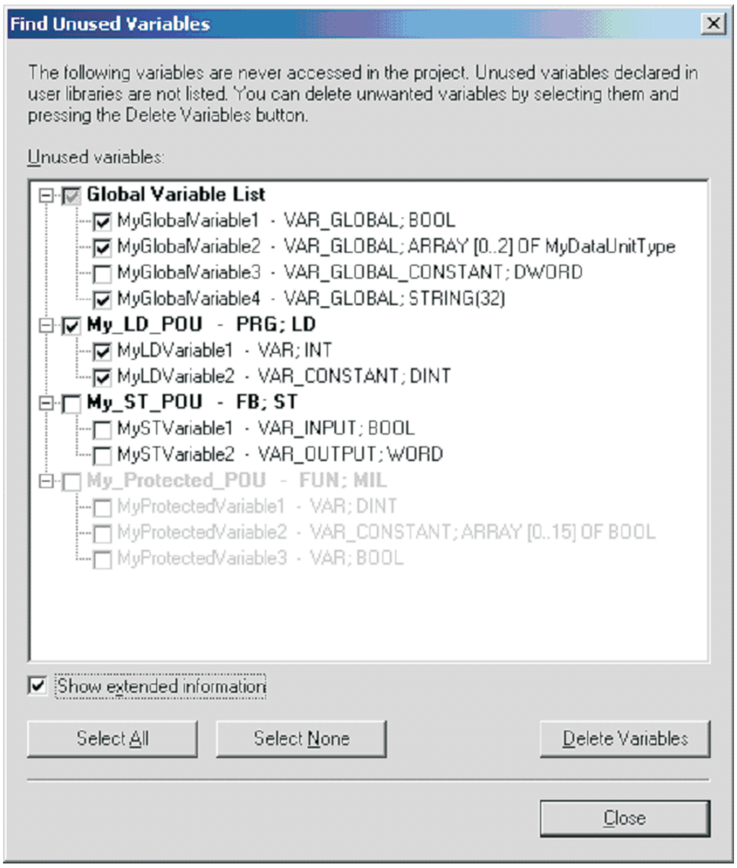
3.17.2 Find unused variables

By using the function **Extras** → **Find Unused Variables** you can find and delete all unused global and local variables that are declared but not used in a project. Unused global and local variables will be detected in the whole project, excluding the user libraries.

NOTE

Finding unused variables can only be performed if the project has been built and was not changed since then. Otherwise a warning message will be displayed.

If the compiled code exists and is up to date, the project will be checked. When there are unused variables in the project, the following dialogue will be displayed. It shows the name, location and – optionally – additional information about the variables. By using this dialogue you can remove the declaration of selected variables, or of all of them.



The following variables are never accessed in the project. Unused variables declared in user libraries are not listed. You can delete unwanted variables by selecting them and pressing the Delete Variables button.

Unused variables:

- ☑ **Global Variable List**
  - ☑ MyGlobalVariable1 - VAR_GLOBAL; BOOL
  - ☑ MyGlobalVariable2 - VAR_GLOBAL; ARRAY [0..2] OF MyDataUnitType
  - □ MyGlobalVariable3 - VAR_GLOBAL_CONSTANT; DWORD
  - ☑ MyGlobalVariable4 - VAR_GLOBAL; STRING(32)
- ☑ **My_LD_POU - PRG; LD**
  - ☑ MyLDVariable1 - VAR; INT
  - ☑ MyLDVariable2 - VAR_CONSTANT; DINT
- □ **My_ST_POU - FB; ST**
  - □ MySTVariable1 - VAR_INPUT; BOOL
  - □ MySTVariable2 - VAR_OUTPUT; WORD
- □ **My_Protected_POU - FUN; MIL**
  - □ MyProtectedVariable1 - VAR; DINT
  - □ MyProtectedVariable2 - VAR_CONSTANT; ARRAY [0..15] OF BOOL
  - □ MyProtectedVariable3 - VAR; BOOL

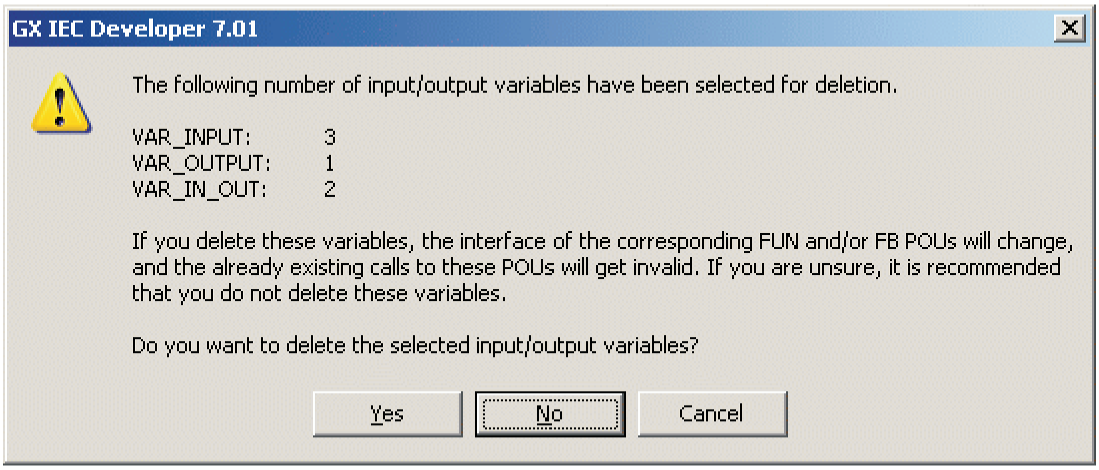
☑ Show extended information

Fig. 3-59:
Find unused variables

Each unused variable is listed under the container of its declaration: the Global Variable List for global variables, or the corresponding POU for local variables. Only those containers are listed where unused variables exist. For example, if there is no unused global variable, the Global Variable List location will not be enlisted. Containers are written in bold text and appear at a higher level than their contained items.

Variables in containers to which no read-access is granted at the current security level are not enlisted. Variables in containers to which no write-access is granted at the current security level are displayed in a disabled state. These variables cannot be selected for deletion.

It is possible that a function (FUN) or function block (FB) POU has an unused variable with the type VAR_INPUT, VAR_OUTPUT or VAR_INOUT. These kinds of variables serve as the interface of the POU. Although an unused interface variable can be deleted, this deletion changes the interface of the POU, and all callers of this POU have to be revised. For this reason, when at least one interface variable is selected for deletion, the following confirmation message will be displayed:



The following number of input/output variables have been selected for deletion.

```
VAR_INPUT:      3
VAR_OUTPUT:     1
VAR_IN_OUT:     2
```

If you delete these variables, the interface of the corresponding FUN and/or FB POUs will change, and the already existing calls to these POUs will get invalid. If you are unsure, it is recommended that you do not delete these variables.

Do you want to delete the selected input/output variables?

Fig. 3-60: *Warning message*

The progress of the deletion will be indicated in the left side of the status bar.
If all unused variables have been deleted, the dialogue (Fig. 3-59) will be closed. Otherwise the dialogue will be kept displayed showing the updated list of unused variables.

3.18 Search and Replace

A new functionality to replace an object in Ladder and FBD is available.

If an operator, a function or a function block is marked in editors of the type LD and FBD and if you select a new operator, function or function block (e.g. from the dialogue **Function Block Selection** or from the pick list with the last recently used objects), the marked object is replaced by the new object.

Example ▽

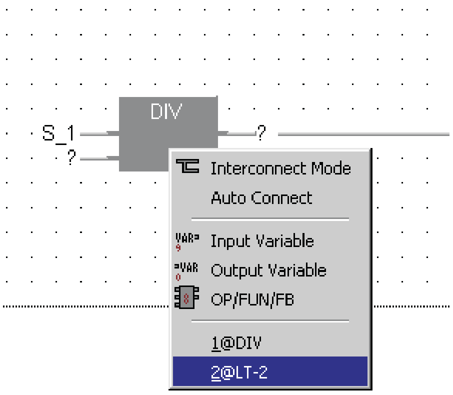


Fig. 3-61: *Before the operation*

After the operation:

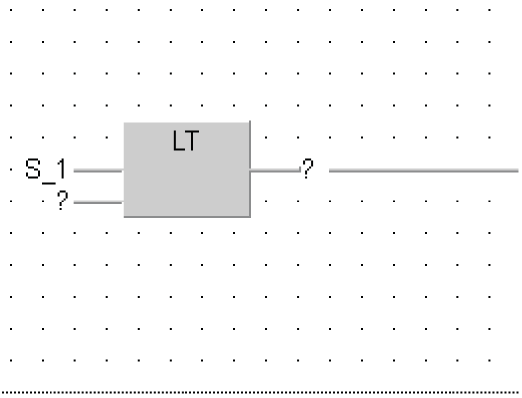


Fig. 3-62: *After the operation*

△

No additional input or output boxes are inserted (regardless of whether the option **Automatic Input/Output Variables** is enabled) and the already available connections to the replaced operator, function or function block are automatically connected to the new one.

If the height of the new object is greater than the old one, the functionality **Open Row** is automatically applied to the lower border of the old object, before the new object is inserted. If the width of the new object is greater than the old one, the functionality **Open Column** is automatically applied to the right border of the old object, before the new object is inserted.

Example ▽

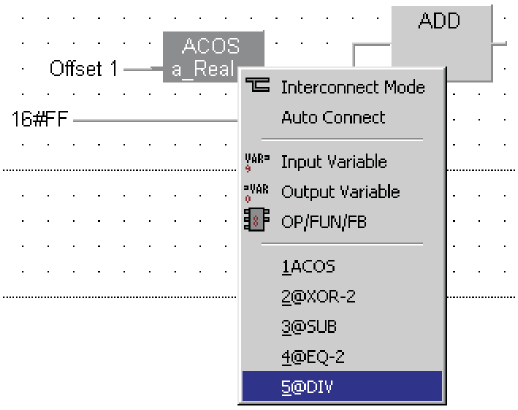


Fig. 3-63: *Before the operation*

After the operation:

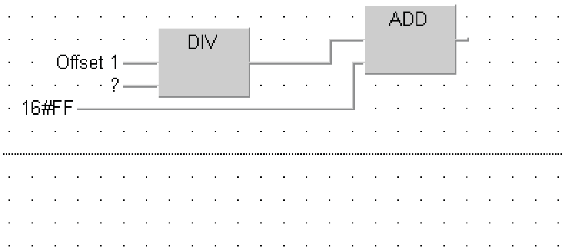
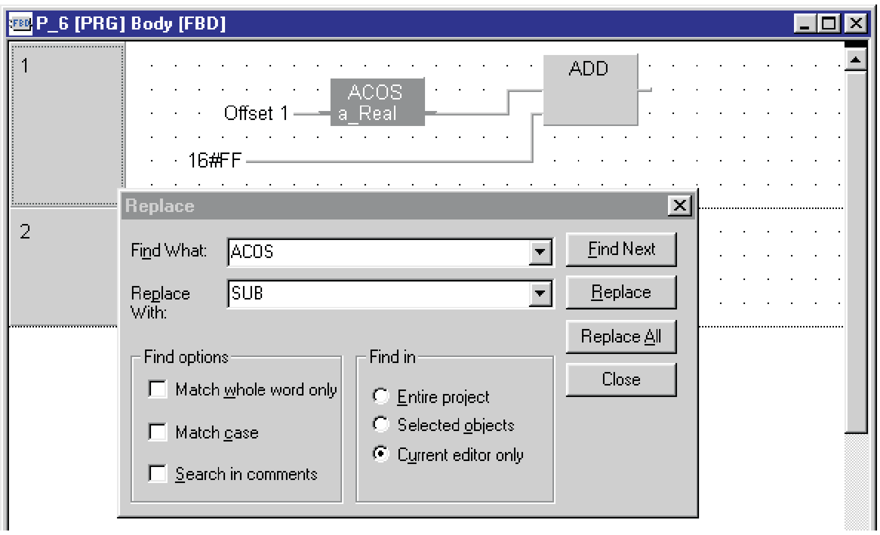


Fig. 3-64: *After the operation*

△

If the size of the new object is equal or smaller than the old one, no change in the layout is done.

The functionality to replace an object in Ladder and FBD is added to the search and replace functionality, so that a manual replace functionality (Button **Replace**) and an automatic replace functionality (Button **Replace All**) is available:


P_6 [PRG] Body [FBD]

Replace

Find What: ACOS

Replace With: SUB

Find options

- ☐ Match whole word only
- ☐ Match case
- ☐ Search in comments

Find in

- ○ Entire project
- ○ Selected objects
- ● Current editor only

Fig. 3-65: *Dialogue* ***Replace***

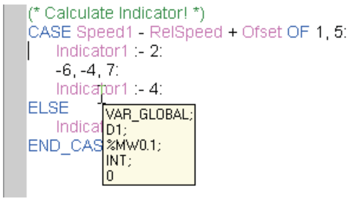
If the autoconnect mode is switched off, it will be switched on automatically in the background during the operation.

If the name in the field **Replace With** is no valid operator, function or function block, the search and replace operation will be aborted.

3.19 Tooltips

Tooltips, which are provided for body editors of the type LD, FBD, ST, IL and MELSEC IL, support the user during the programming and monitor activity.

If the user moves the mouse cursor over a variable, a Tooltip will appear after a short moment:



```
(* Calculate Indicator! *)
CASE Speed1 - RelSpeed + Ofset OF 1, 5:
     Indicator1 :- 2:
     -6, -4, 7:
     Indicator1 :- 4:
ELSE
     Indica
END_CAS
```

```
VAR_GLOBAL;
D1;
%MW0.1;
INT;
0
```

Fig. 3-66:
Appearance of ToolTips

The provided information in the displayed Tooltip can be configured in the options dialogue (see section). The following items are available:

- Monitored value (is only shown in monitor mode)
- Class
- Mitsubishi address
- IEC address
- Type
- Initial value
- Comment (only the first 80 characters are shown)
- Remark (only the first 80 characters are shown)

An additional setting switches between single line (all items are separated by semicolons in one line) and multi line (each item is displayed in a separate line).

4 Projects and Objects

4.1 Projects

The project is the object at the top of the hierarchical structure used in GX IEC Developer. In IEC 61131-3 terms, a project is comparable with a resource. In GX IEC Developer it corresponds to **one** PLC system.

A project consists of considerably more than just one file. When you create a new project, GX IEC Developer creates a separate subdirectory with the name of the project. All the files used by the project are always stored in this directory.

4.1.1 The automatically-generated project files

- ***.PRO**: Project (all objects except libraries)
- ***.BAK**: Backup of the file *.PRO
- ***.SAV**: Extra backup copy of the file *.BAK generated when a restore is performed (→ page 4-15)
- **PLCDOKU**: ASCII PLC configuration files when printing (PLC_Parameter)
- **WORKSPACE.SYS**: Workspace settings (View - Save Workspace)
- **CUSTOM.SYS**: Graphic Macros (Tools - Custom)
- **LRUBOXES.SYS**: All the programming instructions used in the project (Last recently Used list box in the Function Block Selection dialogue box in the graphical editors)
- **SFC.OBJ**: Last program organisation unit created in the Sequential Function Chart language (temporary file stored in a special format)

NOTE

During the compilation and data transfer processes GX IEC Developer generates a number of temporary files that should not be edited or tampered with in any way. See the Appendix D for details.

4.1.2 Files with standard extensions

*.ASC	Import/Export File
*.EDM	Entry Data Monitor Settings
*.SED	Setup for Entry Data Monitor
*.PRN	Print File
*.XLS	Excel File for Device Editor
*.SUL	User Library
*.SCT	Files for the Cross Reference data base
*.SCX	Automatic link of the SCT files to the SCX file when generating the cross reference data base

4.2 What are the components of a project?

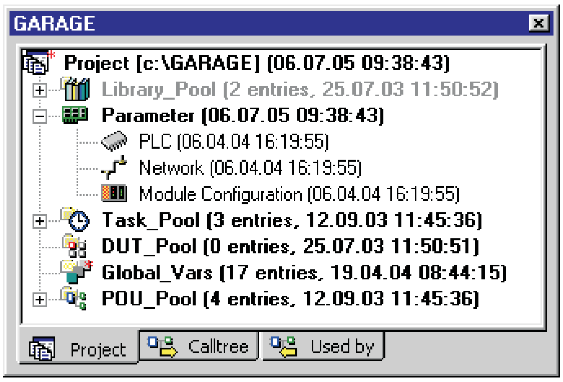
Projects consist of Objects. The following are all objects in GX IEC Developer:

- The libraries in the Library_Pool
- The PLC parameters
- The tasks in the Task_Pool
- The data unit types in the DUT_Pool
- The global variables
- The program organisation units in the POU_Pool, consisting of headers and bodies

These objects are displayed in the **Project Navigator window**, which is automatically displayed when you open a project. The project name is displayed as the first entry in this window.

The Project Navigator window is the ‘control centre’ or ‘starting point’ for all operations performed on these objects. The relationships between the original objects are reflected in the hierarchical structure of the Project Navigator tree.

4.3 The Project Navigator and Objects

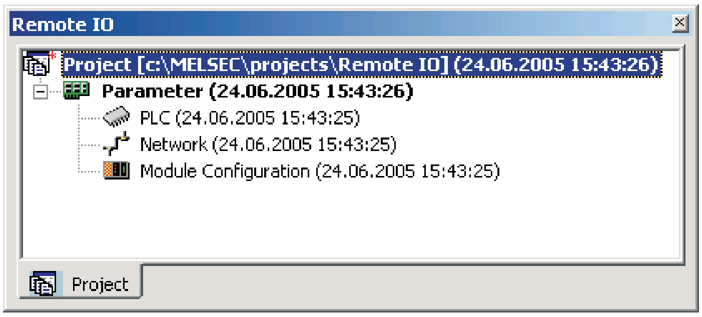


GARAGE

Project [c:\GARAGE] (06.07.05 09:38:43)
Library_Pool (2 entries, 25.07.03 11:50:52)
Parameter (06.07.05 09:38:43)
PLC (06.04.04 16:19:55)
Network (06.04.04 16:19:55)
Module Configuration (06.04.04 16:19:55)
Task_Pool (3 entries, 12.09.03 11:45:36)
DUT_Pool (0 entries, 25.07.03 11:50:51)
Global_Vars (17 entries, 19.04.04 08:44:15)
POU_Pool (4 entries, 12.09.03 11:45:36)

Fig. 4-1:
The Project Navigator window

In the Navigator three different views can be selected via tabs: Project, Calltree, and Used by. For details refer to section 3.4.1.



Remote IO

Project [c:\MELSEC\projects\Remote IO] (24.06.2005 15:43:26)
Parameter (24.06.2005 15:43:26)
PLC (24.06.2005 15:43:25)
Network (24.06.2005 15:43:25)
Module Configuration (24.06.2005 15:43:25)

Fig. 4-2:
The Project Navigator window for a Remote I/O project

Q Remote I/O projects have only parameter data. The Navigator shows only the parameter node and its sub-nodes.

Furthermore, the project navigator window has only the project view, because the Call Tree and Used By views are not applicable for Q Remote I/O projects, as they have no program (and therefore neither libraries, tasks, GVLs nor POUs).

4.3.1 Library Pool

The following libraries contain the programming instructions:

- Standard Library
- Manufacturer Library
- User Library

For more information on the library concept see page 4-8.

4.3.2 Parameter

In the dialogue box PLC parameter you set and store the following PLC system parameters:

- Memory parameters
- Latch operand range
- Timers and Counters range
- I/O configuration
- Network link settings
- PLC setup data
- Special parameters
- System variables
- PLC CPU type selection

NOTE See chapter 5 for details on configuring your system.

4.3.3 Task Pool

Within a project, the task is the highest level in the hierarchy. The task contains the executable program organisation units, which are the subprograms that go to make up a project. Tasks can be either event-triggered or executed at preset intervals. Tasks are also assigned priorities, which determine the execution hierarchy when tasks are executed under the same conditions.

4.3.4 DUT Pool

Data unit types are structured data types consisting of collections of variables that can be of different data types. Data unit types must be declared as global variables.
Values can be assigned to the individual elements (variables) of the data unit type in the body of the POU (→ Chapter 6).

A DUT can also be used as return type for a function. An ARRAY of DUT is not allowed as function result if it is copied to another ARRAY of DUT. But it is possible when the function result is assigned to the parameter of a FB macro.

If the same function is used twice as input parameter of e.g. a FB macro, a temporary variable would be necessary because the second function call would overwrite the values of the first function call. This is not supported for:

- A DUT as function result.
- An ARRAY of DUT as function result.

4.3.5 Global Variables

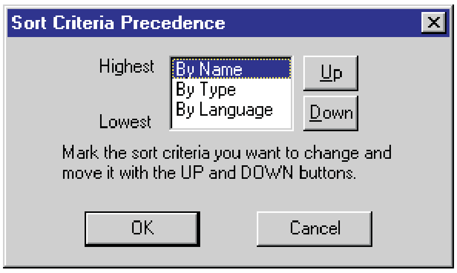
Global variables are declared for the entire project. They make it possible to exchange data between all the individual subprograms that make up the project (→ Chapter 6).

4.3.6 POU Pool

The sorting criteria for the listing of the POUs in the POU pool can be toggled. Three different sorting criteria are available:

- By name (alphabetically)
- By type
- By programming language (alphabetically)

Right-click on the POU pool to open the context menu and select **Sorting Criteria**. The **Sort Criteria Precedence** dialogue will be opened. In this dialogue you can specify the precedence of each sorting criteria via the **Up** and **Down** buttons.



Mark the sort criteria you want to change and move it with the UP and DOWN buttons.

Fig. 4-3:
Sort Criteria Precedence

All program organisation units (POUs) consist of a header and a body.

- Header : The header contains the declarations of the Local Variables used in the POU.
- Body : The body contains the actual PLC program code of the POU.

The entries after the POU name specify the Class:

● [PRG]	:	Program
● [FB]	:	Function Block
● [FUN:INT]	:	Function

The entry before the POU name specifies the compilation:

- * : The asterisk indicates that this POU has not been compiled.

The entries after ‘Body’ specify the programming language in which the PLC program in the body is written:

● [IL]	:	Instruction List
● [LD]	:	Ladder Diagram
● [FBD]	:	Function Block Diagram
● [SFC]	:	Sequential Function Chart
● [MELSEC IL]	:	MELSEC Instruction List
● [ST]	:	Structured Text
● Action_Pool (SFC only)	:	All the POU’s Actions

4.4 The Program Organisation Unit (POU)

The ‘actual’ PLC program code is stored in the body of each POU. However, this no longer consists of just one PLC program with its own subroutines. In GX IEC Developer you use groups of subprograms – the program organisation units. These self-contained subprograms are administrated by tasks; together, they make up the complete PLC program.

A program organisation unit always consists of a header and a body. The header contains the variables, which are only available within this POU. The body contains the executable PLC program code.

NOTE

An introduction to the IEC 61131-3 standard and the basic principles of this new programming concept is provided in the Beginner’s Manual in the chapter ‘The IEC 61131-3 Standard’. For tips on writing PLC sequential control programs in the various programming languages (editors), please refer to chapter 6 in this Reference Manual.

4.5 Preparation Before You Start Programming

Structured programming …

To work really effectively with GX IEC Developer it is very important to analyse and plan the task to be programmed in as much detail as possible before you actually begin to write the code. Divide the entire machine process into small sub-processes, and treat each sub-process as an independent unit. Implement each of these sub-processes as a **Program Organisation Unit**.

… and program management

Develop a sequential structure showing how the individual sub-processes are to be executed. The program organisation units are called via **Tasks** (triggered either by events or at defined intervals).

Global Variables

Draw up a list of your **Global Variables**, i.e. the variables whose hardware addresses must be known, and which must be available to all the components of the program (e.g. inputs, outputs, operands for possible later visual display of the process or operator guidance, analog inputs, etc). The global variables should be declared before you start developing the program so that each sub-program can access them. Of course, you can also add to the list in the course of program development, if necessary.

Reusable subroutines

Program frequently-used processes as **Functions** or **Function Blocks**. The functions and function blocks are stored in a library and can be used at any point in the project where they are needed.

NOTE

To be able to call a function block you must create an ‘instance’ of it, in a process known as ‘instancing’ or ‘instantiation’ (→ Chapter 6).

Data Unit Types

Create **Data Unit Types** for parts of the process where the same constellations of variables must be accessed again and again. The creation of data unit types corresponds to the instantiation of function blocks (→ Chapter 6).

Graphical or textual based programming

The best editor type to use for your project, graphical or textual, depends on the type and scope of the programming task. As a rule of thumb, **Ladder Diagram** and **Function Block Diagram** are best for most applications because of their simple entry options and clear representation of the program. These graphical editors make programming functions and function blocks particularly simple. The **Instruction List** language allows more flexible programming, but it also requires knowledge of the underlying syntax, for example for calling functions and function blocks or for programming Arrays. The **Sequential Function Chart** and the **Structured Text** languages are special cases; strictly speaking, they actually are no programming languages at all, but rather structuring tools. It makes it very easy to see and follow the program sequence. Individual logical sections of the program (sub-processes) can be programmed as macros (→ Chapter 6).

4.6 Library Concept

A distinction is made between the following three types of library:

- Standard Library
- Manufacturer Library
- User Library

Standard Library

The Standard Library is for information purposes only. It contains the following elements:

- **DUT_Pool** : Irrelevant for programming purposes
- **Global_Vars** : Irrelevant for programming purposes
- **POU_Pool** : The POU_Pool contains the standard IEC programming instructions supported in GX IEC Developer. You can view the variable declarations in the header, but you cannot edit them.

NOTE The SLIB.LIB contains the contents of the Standard Library in packed format.

Manufacturer Library

The Manufacturer Library is also for information purposes only. It contains the following elements:

- **DUT_Pool** : Irrelevant for programming purposes
- **Global_Vars** : Irrelevant for programming purposes
- **POU_Pool** : The POU_Pool contains all the ‘Adapted’ MELSEC programming instructions with nomenclature modified for the IEC editors (→ ‘Programming Instructions’ in Appendix E). You can view the variable declarations in the header but you cannot edit them.

NOTE The file MLIB.LIB contains the contents of the Manufacturer Library in packed format.

User Libraries

A user library gives you the possibility of creating functions and program segments which can be used on a supra-project basis. The program segments are grouped in a library and can be used in any project. Following updating, changes made to the library can be applied to other projects.

A user library can contain all the objects that are also available in a project: Global Variables List, Data Unit Types, Arrays and Program Organisation Units. You can assign a password to protect the know-how in the library.

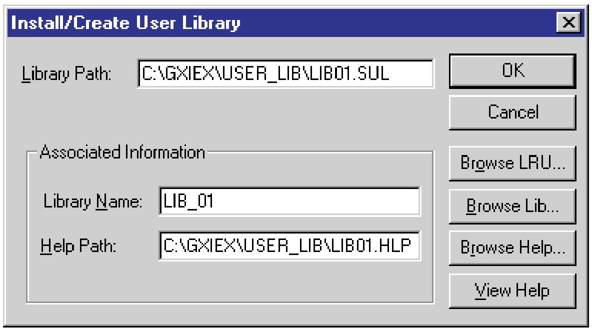
The name of the user library file has the file extension SUL.

NOTE The MMP701.INI initialisation file contains the LIBRARIES section, where you can make directory settings (→ Chapter 2).

How to create a new user library

① Select the **Install/Create User Library** command in the **Edit** menu.

The following dialogue window is displayed:



Library Path: C:\GXIEX\USER_LIB\LIB01.SUL

Library Name: LIB_01

Help Path: C:\GXIEX\USER_LIB\LIB01.HLP

Fig. 4-4:
Install/Create User Library

② Click on the **Browse Lib.** button. The **Select Library Path** dialogue window appears.

③ Choose the directory in which the library is to be stored.

④ Enter a new name for the library.

⑤ Confirm with **OK**.

The entries are transferred to the **Install/Create User Library** dialogue window, where they can also be changed.

⑥ Confirm the library entries with **OK**.

The new library appears in the Navigator, along with the Standard Library and the Manufacturer Library.

The Install/Create User Library dialogue window

Library Path: Directory in which the library file is to be created. This can be any directory, although it is sensible to choose a central directory for the User Libraries.

Library Name: Descriptive name for the library, max. 32 characters.

Help Path: Directory and file name for the Help file. A separate Help file can be created for each User Library.

Browse LRU...: A window is opened which displays the most recently used User Libraries. You can select a library from this window.

Browse Lib...: A window is opened for you to select a directory and enter a library file. A User Library is given the file extension SUL.

Browse Help...: A window is opened for you to select a directory and enter a Help file. A Help file is given the file extension HLP.

View Help: The Help file is opened.

How to protect the User Library with a password

① Select the desired User Library in the Navigator.

② Select the **User Library - Change Password** command in the **Edit** menu.

This opens a dialogue window containing the following items:

Old Password: When changing an existing password, the old password must first be entered here. If you are assigning a password for the first time, you must skip this field by pressing **[ENTER]**.

New Password: Entry of the new password.

Confirmation: Repeat entry of the new password.

Access rights

Allow read access for closed library: If this option is checked, the body for all POUs of the corresponding user library is shown in the navigator and the body can be opened in read-only mode.

NOTES

A protected User Library can only be opened and edited if the password is known. As long as the User Library is closed and the password is not known, it is impossible to make any unauthorised changes. Only the header of the POU can be viewed, the body cannot be opened.

If the option **Allow read access for closed library** is activated, the body (and the header) for all POUs of the corresponding user library is shown in the navigator and the body can be opened in read-only mode:
- from the navigator
- from the header of the POU
- from other POU bodies (with a function or a function block of this user library)
- from the browser window
- from the dialogue ‘Compile/Check Messages’
- from the system error dialogue

Remove/Delete User Library

Remove: The Remove User Library command removes the User Library selected in the Navigator from the project. The library must be open at the time. The file is not deleted by this operation.

Delete: The Delete User Library command physically deletes the User Library selected in the Navigator, along with the associated library file.

Update User Library

Every User Library can be used and modified in various projects. If, however, a change is made in Project_1, this change is not available in Project_2 until the Update Libraries command has been executed in the Edit menu.

WARNING:
It may, however, also be advisable not to update the library in a particular project. Consequently, you should always take a look at the associated Help texts of the libraries before performing an update!

Browse for User Libraries

If User Libraries which are not found in the specified path are accessed (e.g. open User Library, export project), the **Userlib not found** dialogue opens where you can browse for them. From this dialogue you can also create new User Libraries and save the information from the project to them.

This function does not depend on the current status of the User Libraries (opened, closed, changed, not changed, password, no password).

Under the following conditions a new User Library has to be created:

① When opening the User Library (transition from installed to opened status)

② When saving the user-library (transition from changed to opened status)

③ When exporting a project with user-libraries (no transition)

If such a condition is recognised, a dialogue is opened where you either can:

① Create a new User Library at the original place

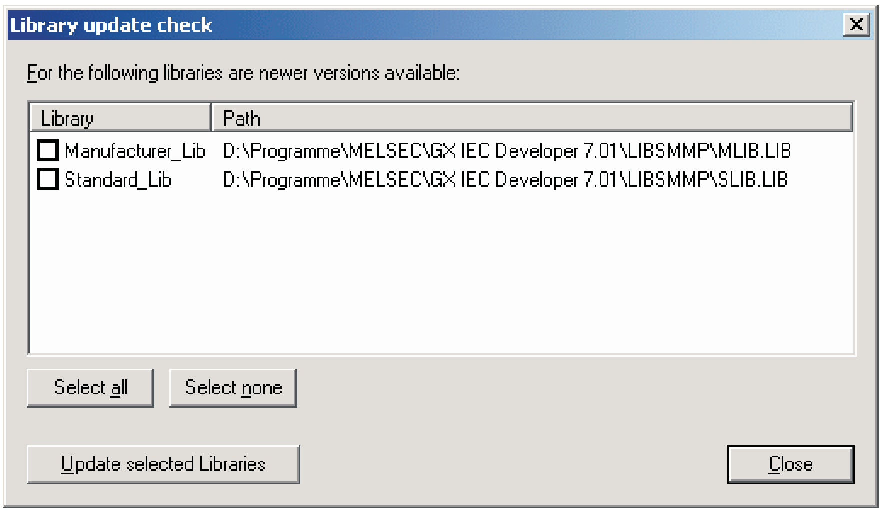
② Browse for the User Library

③ Create the User Library at any proper place

④ Cancel the new creation of the User Library

Check for new library versions when opening a project

When opening a project with libraries, an automatic check for new available versions is executed. If a new version is found, the library and the corresponding file are listed in the following dialogue:



Library update check

For the following libraries are newer versions available:

Library	Path
□ Manufacturer_Lib	D:\Programme\MELSEC\GX IEC Developer 7.01\LIBSMMP\MLIB.LIB
□ Standard_Lib	D:\Programme\MELSEC\GX IEC Developer 7.01\LIBSMMP\SLIB.LIB

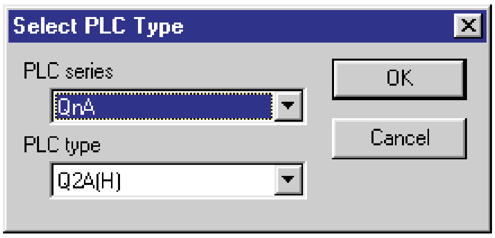
Fig. 4-5: Library update check

When updating the selected libraries, opened libraries are closed and for changed libraries you will be asked to save.
Updated libraries are removed from this list.

4.7 Working with Projects

How to Create a New Project

① Select **New** in the **Project** menu. The dialogue box **Select PLC Type** is displayed.


Select PLC Type

PLC series: QnA

PLC type: Q2A(H)

Fig. 4-6:
At this point,
***CPU Type** selection*
is the only active field.

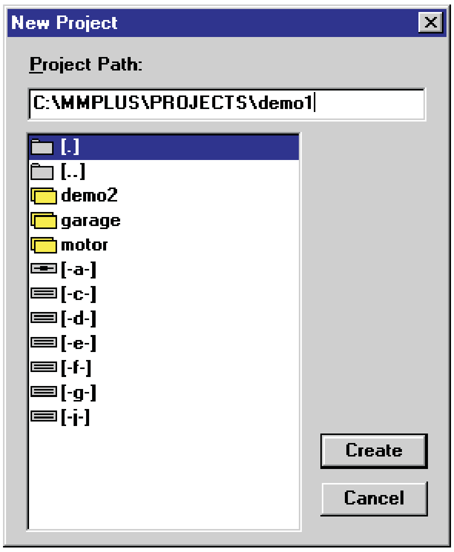
② Select the CPU for which you are programming from the list box.

③ Confirm your selection with **OK**.

The dialogue box **New Project** is displayed.

④ Select a directory for your project or type in the path in the Project Path field. If you wish, you can also enter a directory that does not yet exist; GX IEC Developer will then create it for you automatically.

⑤ Enter the name of the project at the end of the project path.


New Project

Project Path: C:\MMPLUS\PROJECTS\demo1

Fig. 4-7:
The project name can be a maximum of 255 characters long (Microsoft Windows standard for long file names).
The project name does not need an extension as it refers to a directory and not a file.

⑥ Click on the **Create** button.

The program displays the Wizard dialogue box.

⑦ Select the required project option and confirm with **OK**.

The program will then create the project database on the disk.

NOTE

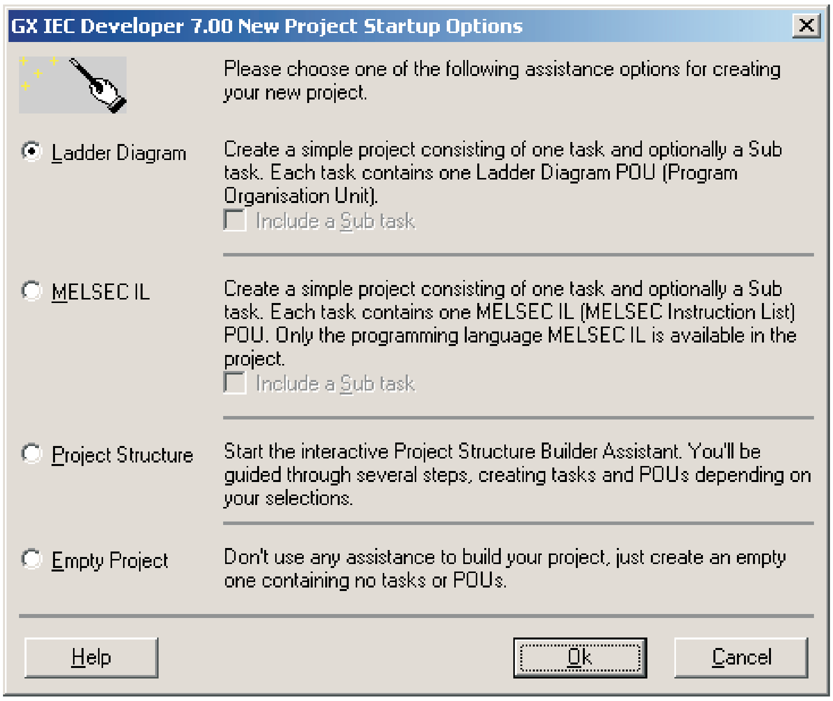
In case of creating a new project for a FX CPU a warning comes up, that warns you of existing limitations concerning the **Online Program Change** function for the FX series (→ 8.7.3). So you should avoid making changes in the program that cannot be carried out by online program change, if you want to use this function later on.

The project creation process is documented in an information box (loading the programming instructions from the libraries, opening the project).

When it generates a new project GX IEC Developer doesn't create a single file but a subdirectory with the project name. This subdirectory contains all the files that make up the project. The two libraries – the Standard Library and the Manufacturer Library – are always loaded with every new project that is created. When you begin, they are displayed in the Project Navigator window with the various pools, which are still 'empty'.

The Wizard

The Wizard is a special tool. When the new project has been generated the following dialogue appears.



GX IEC Developer 7.00 New Project Startup Options

Please choose one of the following assistance options for creating your new project.

● Ladder Diagram — Create a simple project consisting of one task and optionally a Sub task. Each task contains one Ladder Diagram POU (Program Organisation Unit).
□ Include a Sub task

○ MELSEC IL — Create a simple project consisting of one task and optionally a Sub task. Each task contains one MELSEC IL (MELSEC Instruction List) POU. Only the programming language MELSEC IL is available in the project.
□ Include a Sub task

○ Project Structure — Start the interactive Project Structure Builder Assistant. You'll be guided through several steps, creating tasks and POUs depending on your selections.

○ Empty Project — Don't use any assistance to build your project, just create an empty one containing no tasks or POUs.

Fig. 4-8: *Wizard for new projects*

NOTE The project wizard will not show up when creating a Q Remote I/O project, because the project does not have any program.

Select the desired startup option for your project:

Ladder Diagram: Creates a simple project consisting of one task and optionally a Sub task. Each task contains one Ladder Diagram POU (Program Organisation Unit).

MELSEC IL: Creates a simple project consisting of one task and optionally a Sub task. Each task contains one MELSEC IL (MELSEC Instruction List) POU. You enter the MELSEC mode. Only the programming language MELSEC IL is available in the project. IEC POUs can not be added.

Project Structure: Starts the interactive Project Structure Builder Assistant. You are guided through several steps, creating tasks and POUs depending on your selections,

Empty Project: Does not use any assistance. Creates an empty project containing no tasks or POUs.

NOTE If you do not wish to use the Wizard's support you can deactivate it by editing the Wizard setting in the MMP701.INI file (→ Chapter 2).

How to open an existing project

① Select **Open** in the **Project** menu, or click on the corresponding tool in the Toolbar.

Tool for opening an existing project

The **Open Project** dialogue box is displayed.

② Select the drive, the directory and the project name.

③ Click on the **Open** button.

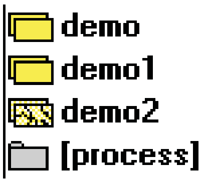


Fig. 4-9:
Projects are easy to identify in the directory list – their path symbols are displayed in yellow.
If the project was not stored properly because of a power failure or because it was not closed correctly a ‘cracked’ yellow path symbol is displayed next to the project name. Usually, such projects can be restored.

NOTE

Please note that you can only open and edit one project, in one Project Navigator window. But you can start GX IEC Developer several times to open and edit more than one project at a time.

How to recover a project

GX IEC Developer has a facility for recovering projects that were not closed properly because of a power failure or some other problem. These projects are identified by a ”cracked” directory path symbol in the Open Project dialogue box.

① Click on the ”cracked” directory symbol.

A dialogue box is displayed with three options: **Recover**, **Use Backup** and **Cancel**

Recover: This option restores all the project’s data so that nothing is lost. The system restores the exact editing status of the project at the point the unplanned termination happened.

Use Backup: Loads the status of the project as it was the last time you loaded it for editing. All changes made between then and the unplanned termination are lost. An additional backup copy of the project is generated and stored in the file *.SAV.

Cancel: Cancels the procedure and closes the dialogue box.

② Click on the appropriate button.

NOTES

When you create a new project the files *.PRO and *.BAK are created automatically. All editing changes and entries you make are stored in the PRO file.

When you open an existing project that was closed correctly GX IEC Developer loads *.PRO and creates *.BAK. Here too, all the entries and editing changes you make are stored in the PRO file. The BAK file stores the status of the project at the point it was loaded.

As you can imagine, when you select **Recover**, the system loads the PRO file in which all the editing changes up to the termination point are stored.

By the same token, selecting **Use Backup** loads the BAK file. This means that if you first created the project in the editing session in which the termination occurred the BAK file will be empty!!! If you loaded an existing project in the last editing session the BAK file contains the status of the project from the point when you began the editing session. Whenever a *.BAK file is loaded an automatic additional backup copy of the project is generated and stored in the file *.SAV.

How to open the SAV file

① Open the File Manager.

② Create a new directory.

③ Copy the SAV file into this new directory.

④ Rename the file *.SAV into *.PRO.

⑤ Open this PRO file in GX IEC Developer.

How to close a project

① Select **Close** in the **Project** menu.
Or:
Close the Project Navigator window with the **Close** command in its control menu (key combination Ctrl F2), or by double-clicking on the control menu button.

A prompt will be displayed asking you if you wish to save the project. If object windows (e.g. headers or bodies) with unsaved editing changes are open a separate prompt will be displayed for each window.

How to Save a Project

① Saving your project with the current name: Select **Save** in the **Project** menu, or click on the corresponding tool in the Toolbar.

💾 Tool for saving the current project

Saving under a new project name: Select **Save As** in the **Project** menu. Select the desired subdirectory and assign a new project name.

When you save the project the backup file is updated automatically.

How to delete a project

① In the **Project** menu open the **Other** submenu and select **Delete**.

The **Delete Project** dialogue box is displayed.

② Select the drive, directory and project name.

GX IEC Developer projects are identified by a yellow path symbol next to the directory name.

③ Click on the **Delete** button.

NOTE

You cannot delete a project while it is open!

How to rename a project

① In the **Project** menu open the **Other** submenu and select **Rename**.

The **Rename Project** dialogue box is displayed.

② Select the drive, directory and project name.

GX IEC Developer projects are identified by a yellow path symbol next to the directory name.

③ Enter the new project name in the **New Project Name** field.

The project name can have a maximum of 8 characters. You can also enter the name of a directory that does not yet exist, and GX IEC Developer will create it for you automatically.

④ Click on the **Rename** button.

NOTE

You cannot rename a project while it is open.

How to copy a project

① In the **Project** menu open the **Other** submenu and select **Copy**.

The **Copy Project** dialogue box is displayed, with two list boxes.

② Select the drive, directory and name of the project to be copied in the left list box.

GX IEC Developer projects are identified by a yellow path symbol next to the directory name.

③ Creating a copy with the same name: Simply select the target drive and directory in the list box on the right. Creating a copy with a new name: Select the target drive and directory in the right hand list box, and then enter a new name at the end of the path. The new project name can have a maximum of 8 characters. You can also enter the name of a directory that does not yet exist; GX IEC Developer will then create it for you automatically. Overwriting an existing project with a copy: Select an existing project in the right hand list box.

④ Click on the **Copy** button.

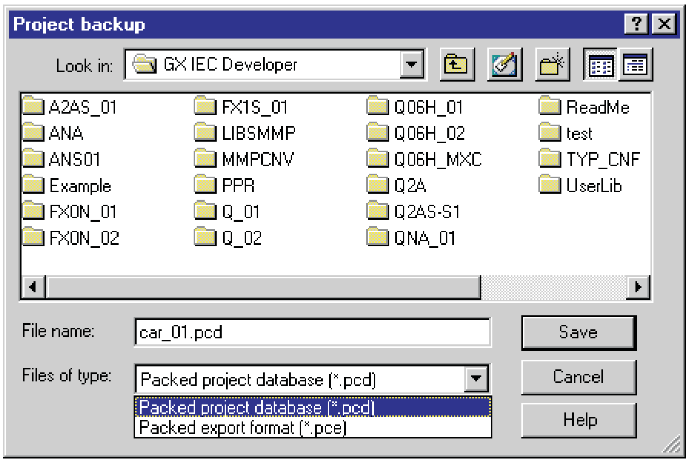
NOTE

You cannot copy or overwrite a project while it is open. Please also have a look at the functions provided by the **Import** and **Export** commands in the **Project - Other** menu (→ Chapter 8).

How to back up a project

The command **Extras - Backup Project** packs the project files and stores them to a medium on the selected drive. External data media such as floppy disks are supported. If the resulting project file is larger than the capacity of the disks it is automatically split across the necessary number of disks.

① Select **Backup Project** in the **Extras** menu. The following dialogue box is displayed.



Look in: GX IEC Developer

File name: car_01.pcd

Files of type: Packed project database (*.pcd)

Packed project database (*.pcd)
Packed export format (*.pce)

Fig. 4-10:
Project backup dialogue

② Select the format for the backup operation:

Packed project database: File format *.PCD: The complete database (already compiled) and the libraries will be packed.

Packed export format: File format: *.PCE: The project data (without compilation, without libraries) will be exported and packed. When restoring a PCE file you have to compile the project again.

③ Select the target drive and directory, then enter a name for the target file.

④ Click on the **OK** button to pack the project and copy it to the selected drive.

⑤ If the target drive is a floppy disk drive you will be prompted to insert the first disk. If the packed file is too large to fit on the disk it will be automatically split across the necessary number of disks. You will be prompted to change disks when necessary.

NOTES

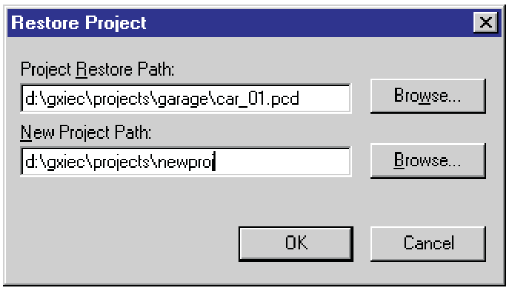
The Backup Project command can only be executed for the project currently opened.

With the **Export** function in the **Project - Other** menu the project data are exported into an ASCII file only, without being compiled or split across the necessary number of disks.

How to restore a project

The command **Extras - Restore Project** is used to restore a backed up project from a packed file. After being unpacked the project is automatically opened.

① In the **Extras** menu select **Project Restore**.
The following dialogue box is displayed.



Restore Project

Project Restore Path: d:\gxiec\projects\garage\car_01.pcd

New Project Path: d:\gxiec\projects\newproj

Fig. 4-11:
Restore project dialogue

② In the upper line select the PCD or PCE file which you wish to unpack and open.

③ In the lower line select the target drive and directory and enter a name for the target project.

④ Click on the **OK** button to unpack and open the project.

NOTE The name for the target project is not a file, but a directory.

How to reorganize a project database

The command **Tools - Reorganize Database...** is used to reorganize a project database. This function fixes possible defragmentation problems in GX IEC Developer 32-bit *.pro project files and repairs damaged database records. The menu item is available only if no project is opened.

① Select **Tools - Reorganize Database...**.
A standard file open dialogue is displayed.

② Select the desired *.pro file to be reorganized and click on the **Open** button.
The project file is reorganized while the progress status is indicated.

4.8 Working with Objects

How to open objects

① Double-click on the object in the Project Navigator window.
Or:
Select the object in the Navigator and then click on **Open** in the **Object** menu.

The window or dialogue box of the selected object will then be opened.

NOTE

> There are two possible responses when you double-click on a POU name, depending on the settings in the MMP701.INI file and in the menu **Extras - Options - Zoom Header/Body - Navigator Zoom** (→ Chapter 2).
> - Show/hide header and body in the Project Navigator window
> - Open Header and body

How to close objects

① Close the object window.

② Before the operation is executed a prompt will be displayed asking you if you wish to save any editing changes. Click on **Yes** if you wish to save the changes.

NOTE

> The various different ways of closing windows are explained in chapter 3.

How to create new objects

① In the Toolbar, select the tool for the object you wish to create.
Or:
Select **New** in the **Object** menu, then select the desired object type from the pop-up list displayed (**POU**, **DUT**, **Task** or **Action**).

② Make the necessary entries in the dialogue box. Enter the name of the object. In the case of program organisation units you must also specify the POU's class (Program, Function Block or Function) and the programming language. In the case of actions you must select the programming language.

The newly-created object is displayed in the Project Navigator window. You can now open the object to edit it, if necessary.

How to save objects

① Select the window of the object you wish to save.

② Select **Save** in the **Object** menu.
Or:
Press the following key combination:
Save Ctrl S

How to rename objects

① Select the object to be renamed in the Project Navigator window.

② Select **Rename** in the **Object** menu.

③ Enter the new name in the dialogue box.

The new name of the object will then be displayed in the Project Navigator window.

NOTE You can only rename closed objects.

How to cut, copy or delete objects

① In the Project Navigator window, select the object that you wish to cut, copy or delete.

② Select the appropriate command in the **Edit** menu.
Or:
Press the corresponding key combination:

Cut	Ctrl X
Copy	Ctrl C
Delete	Del

NOTE The **Cut** operation deletes the data from its original position and stores it in the clipboard.
The **Copy** operation copies the data to the clipboard without changing it in its original location.
The Delete operation deletes the data from its present location without copying it to the **clipboard**. This means that any data that was in the clipboard prior to the delete operation remains unchanged.
Objects can only be deleted if the corresponding object windows are closed (e.g. header **and** body in the case of POUs)

How to paste objects

① Activate the Project Navigator window.

② Select **Paste** in the **Edit** menu.
Or:
Press the following key combination:

Paste	Ctrl V

NOTE Please note that the system does not prompt you for a new name before inserting the data. Objects in the clipboard are inserted in the new location with their original names. You must then rename the new objects manually with **Edit - Rename**, and it may also be necessary to perform editing to eliminate double variable declarations (in the case of POUs).
The **Copy** and **Paste** functions can also be used to copy and paste objects between projects, not just within the current project.

How to copy objects from the current project to another project

① Select the object or objects to be copied in the Project Navigator window.

② Select **Copy** in the **Edit** menu.
Or:
Press the following key combination:
Copy Ctrl C

A copy of the data is now in the clipboard.

③ Close the current project, then open the project into which you wish to copy the objects.

④ Activate the Project Navigator window.

⑤ Select **Paste** in the **Edit** menu.
Or:
Press the following key combination:
Paste Ctrl V

The data from the clipboard are inserted in the Project Navigator window.

NOTE

You can also exchange objects between projects with the Import and Export functions (**Project - Other - Export/Import**).

How to enter comments

① In the Project Navigator window, select the object for which you wish to enter a comment.
Or:
Activate the object window.

② Select **Comment** in the **Object** menu.

③ Type your comment text in the left field.

How to copy, cut and paste comments

① Select the text to copy or cut.

You can use the standard Windows mouse operations and key combinations for selecting text.

② Click on the **Cut**, **Copy** or **Paste** button.

NOTE

You can also use the standard Windows key combinations for these operations:
Cut Ctrl X
Copy Ctrl C
Paste Ctrl V

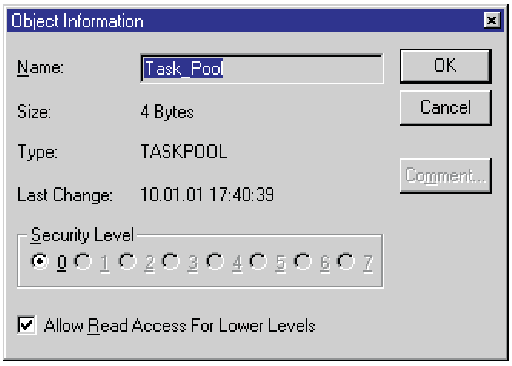
How to view the object information

① In the Project Navigator window, select the object whose information you wish to view.
Or:
Activate the corresponding object window.

② Select **Information** in the **Object** menu.
Or:
Press the key combination Alt Enter.

A dialogue box is opened showing attributes of the selected object.

General Options


Object Information

Name: Task_Pool

Size: 4 Bytes

Type: TASKPOOL

Last Change: 10.01.01 17:40:39

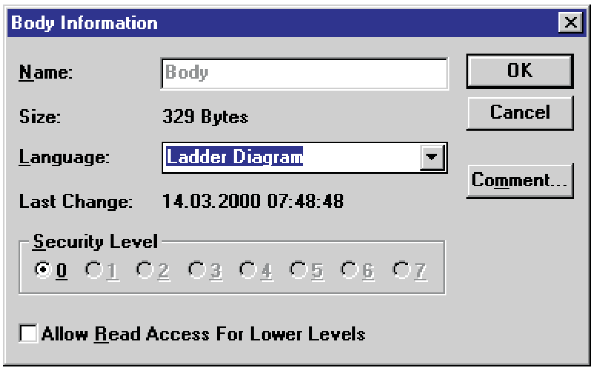
Security Level

● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access For Lower Levels

Fig. 4-12:
Name: *Name of the selected object*
Size: *Size of the selected object in bytes*
Type: *Type of the selected object*
Last Change:
Data and time of the last editing change
Security Level: *Current security level*
Read Access:
Read access for lower security levels
Comment: *Comment text*

Additional Options for Body Objects


Body Information

Name: Body

Size: 329 Bytes

Language: Ladder Diagram

Last Change: 14.03.2000 07:48:48

Security Level

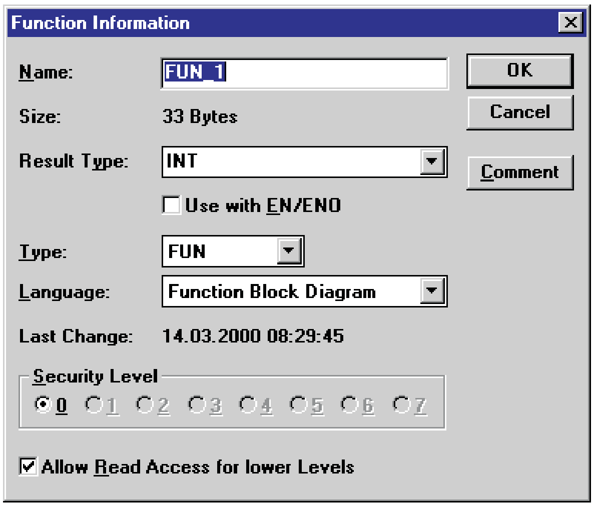
● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

□ Allow Read Access For Lower Levels

Fig. 4-13:
Language: *Programming language*

WARNING:
The system will allow you to change the programming language of a POU, but if you do this you will lose all the codes in the body! A message will be displayed.

Additional Options for Functions



Function Information

Name: FUN_1

Size: 33 Bytes

Result Type: INT

☐ Use with EN/ENO

Type: FUN

Language: Function Block Diagram

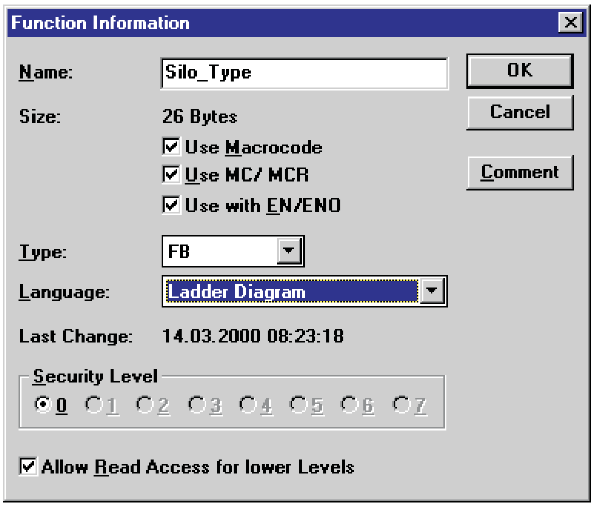
Last Change: 14.03.2000 08:29:45

Security Level: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access for lower Levels

Fig. 4-14:
Result Type*: Result data type (displayed when you change the program class)*
Use with EN/ENO*: If you change the program class to Function this field allows you to specify whether the function is to have a Boolean input and output.*
Type*: Program class*
Language*: Programming language used in the body*

Additional options for Function Blocks



Function Information

Name: Silo_Type

Size: 26 Bytes

☑ Use Macrocode

☑ Use MC/ MCR

☑ Use with EN/ENO

Type: FB

Language: Ladder Diagram

Last Change: 14.03.2000 08:23:18

Security Level: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

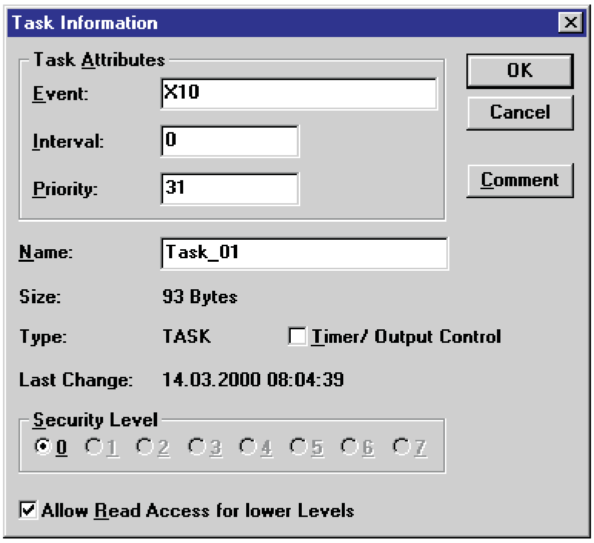
☑ Allow Read Access for lower Levels

Fig. 4-15:
Macrocode*: Macrocode execution of the function block (→ page 6-61)*
MC-MCR*: MC-MCR execution of the function block (→ page 6-61)*
Use with EN/ENO*: If you change the program class to FB (Function Block) this field allows you to specify whether the function block is to have a Boolean input and output.*
Type*: Program class*
Language*: Programming language used in the body*

Additional Options for Task Objects

NOTE

Task attributes can **only** be set in this dialogue box.



Task Information

Task Attributes

Event: X10

Interval: 0

Priority: 31

Name: Task_01

Size: 93 Bytes

Type: TASK □ Timer/ Output Control

Last Change: 14.03.2000 08:04:39

Security Level

● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access for lower Levels

Fig. 4-16:
***Event**: Event-triggered*

***Interval**: Interval-triggered*

***Priority**: Priority level of the task*

***Timer/Output Control**: MC-MCR execution of the task (→ page 6-2)*

How to display an object’s Call Tree (subordinate objects)

① Select the object whose Call Tree you wish to display.

② In the **Object** menu, select **Show** and then select **Call Tree** from the submenu displayed.

An information box is opened showing all the objects used in the selected object.

- Task Call Tree: All associated POUs and all the global variables used in the POUs
- POU, Body and Header Call Trees: All function block instances and all global variables

How to display an object’s Reference List (superordinate objects)

① Select the object whose Reference List you wish to display.

② In the **Object** menu, select **Show** and then select **Reference List** from the submenu displayed.

An information box is opened showing all the objects in which the selected object is used.

- Function Block Reference List: The headers in which the instances of the function block were generated
- POU Reference List: The associated task

4.9 Protecting Your Data

4.9.1 Project security

GX IEC Developer has a eight-level data security system. You can assign a different password to each level, and you can assign security levels to both the entire project and each individual object within the project. The security level assigned to any object will be inherited to all its child objects provided their security level is not higher.

For example, if you assign a security level of 5 to the project you can edit all objects with security levels 0 through five. Objects with security levels 6 and 7 are not editable, but you can view their contents. The names of protected objects are displayed in grey in the Project Navigator window. POUs that are both read and write protected also do not allow you to branch into their headers and bodies in the Project Navigator window.

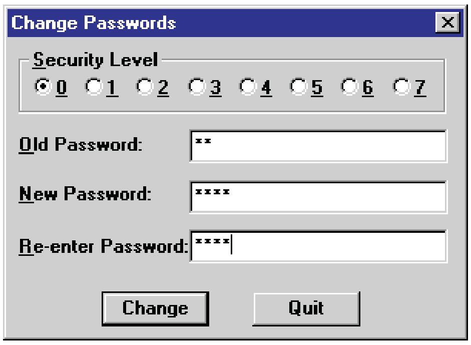
How to assign security level passwords

① Select **Change Passwords** in the **Project** menu.

② Select the security level to which you wish to assign a password.

③ Enter the current password in the **Old Password** field. If you have not yet entered a password for this level, simply leave this field empty.

④ Enter the new password in the **New Password** field.



Change Passwords

Security Level: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Old Password: **

New Password: ****

Re-enter Password: ****

Fig. 4-17:
As an additional security measure asterisks are displayed on the screen instead of the characters you enter.

⑤ Confirm your entry by re-entering the same new password in the **Re-enter Password** field.

⑥ Click on the **Change** button to enter the password.

A message will be displayed telling you whether the system has accepted the old and new passwords.

How to change the project's security level

① Select **Change Security Level** in the **Project** menu.

② Select the desired security level.

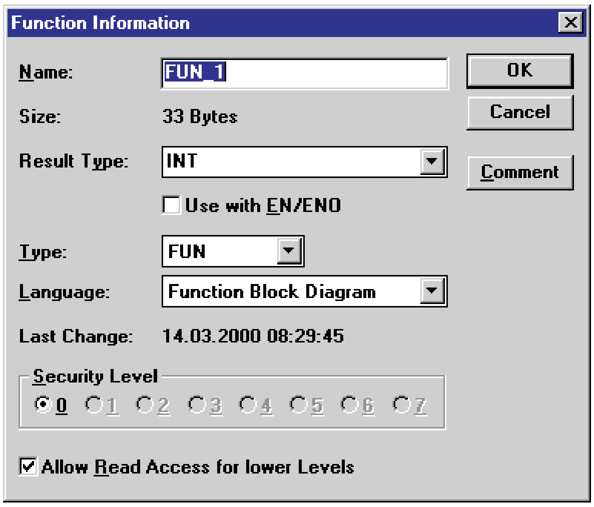
③ Enter the valid password for this security level in the **Password** field.

An error message will be displayed if you enter the password incorrectly.

How to assign read and write protection to objects

① Select the object to be protected in the Project Navigator window.

② Select **Information** in the **Object** menu.
Or:
Press the key combination Alt Enter.



Function Information

Name: FUN_1

Size: 33 Bytes

Result Type: INT

☐ Use with EN/ENO

Type: FUN

Language: Function Block Diagram

Last Change: 14.03.2000 08:29:45

Security Level: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access for lower Levels

Fig. 4-18:
The Object Information dialogue box is displayed.

③ Click on the desired security level.

④ **Write protection**: Click on a security level higher than the current level in order to protect the object against unauthorised editing.

⑤ **Read protection**: Deactivate the **Allow Read Access For Lower Levels** check box to prevent read access to the object.

NOTE

If the project has security level 5 and the object has already been assigned security level 6 you cannot change the security level of the object in this dialogue box. Security levels that cannot be selected are displayed in light grey. To change the security level of such objects you must first change the level of the project; for instance, in this example, you would have to change the project security level to 6. This is only possible if you know the password of this security level, of course. Action: Check and enter PLC keyword for access to the PLC CPU.

How to protect the PLC program

Access to the PLC CPU can be restricted by entering a **PLC keyword** to prevent unauthorised persons from transferring programs from the CPU.

The user is prompted to enter the password once before every Online operation. The password is also checked internally before execution of every subsequent Online operation.

NOTE

In case of QnPRH PLCs in Backup mode the user is asked if he wants to set or delete the keyword for both PLCs. You will be asked to enter the keyword for the Control system first. GX IEC Developer will attempt to use the same keyword for the Standby system as well. If it fails, you will be asked to enter the keyword for the Standby system as well.
If different keywords are used for the control and the standby system, both keywords will be stored and automatically used for any further operations.

① In the **Online** menu select the **PLC keyword** command, then select **Current**, **New** or **Delete**.

② Enter the appropriate PLC keywords in the corresponding fields. You can move from field to field with the [TAB] key. For security, all the characters you enter are displayed as asterisks on the screen.

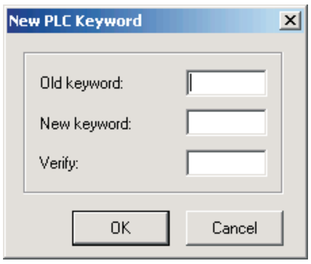
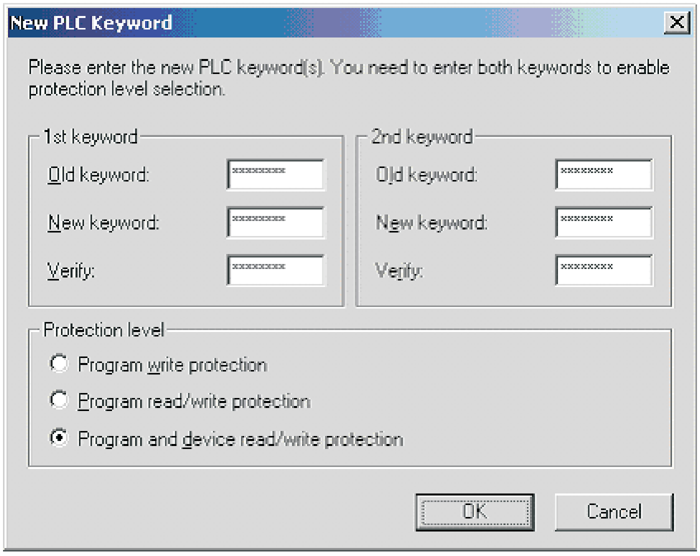


Fig. 4-19:
Entering a new PLC keyword



Please enter the new PLC keyword(s). You need to enter both keywords to enable protection level selection.

1st keyword: Old keyword: *********; New keyword: *********; Verify: *********

2nd keyword: Old keyword: *********; New keyword: *********; Verify: *********

Protection level
- ○ Program write protection
- ○ Program read/write protection
- ● Program and device read/write protection

Fig. 4-20:
Entering a new PLC keyword for a FX3U PLC

1st and 2nd keyword/Protection level (only FX3U)

If the PLC is protected with two keywords, it is possible to modify the protection level without changing the keywords. If both of the old keywords are entered correctly, and neither new keyword is specified, the protection level will be changed to the selected setting, and the keywords will be kept.

The Protection level selection group will be enabled only if both new keywords are specified or both old keywords are specified and neither or both new keywords are specified.

Old keyword: Enter the old keyword (8 characters from range 0 to 9 and A to F).

New keyword: Enter the new keyword (8 characters from range 0 to 9 and A to F).

Verify: Enter the new keyword once again(8 characters from range 0 to 9 and A to F).

③ Confirm your entries with **OK**.

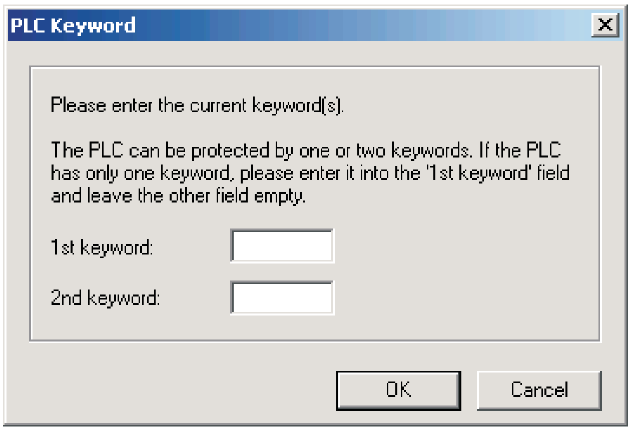
Meaning of the different commands under the submenu PLC keyword

Current: Enter current PLC keyword for access to the Online functions.

New: If an old PLC keyword has already been entered you must first enter it for confirmation. Then enter the new PLC keyword in the next field, after which you must enter it again for confirmation in the last field.

Delete: Delete the current PLC keyword for the PLC CPU.

Disable Protection (only FX3U): You will be offered to specify the first or both keywords in the **PLC Keyword** dialogue.



PLC Keyword

Please enter the current keyword(s).

The PLC can be protected by one or two keywords. If the PLC has only one keyword, please enter it into the '1st keyword' field and leave the other field empty.

1st keyword:

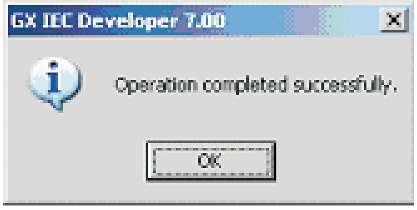
2nd keyword:

Fig. 4-21:
Entering one or both PLC keywords for a FX3U PLC

If only the first keyword is specified, the validity of the keyword will not be verified directly. It will be checked when the first attempt is made to access protected data. In that case, if the currently specified keyword is wrong, it will be asked again.

If both keywords are specified, the validity of the keywords will be verified directly. In case the keywords are incorrect, they will be asked again. In case the keywords are correct, the protection of the PLC will be immediately disabled. A warning will be given that the protection will be enabled again only when it is explicitly enabled or the PLC is reset.

Enable Protection (only FX3U): By calling this menu command the disabled keyword protection can be re-enabled. The following success message will be displayed after performing the operation.



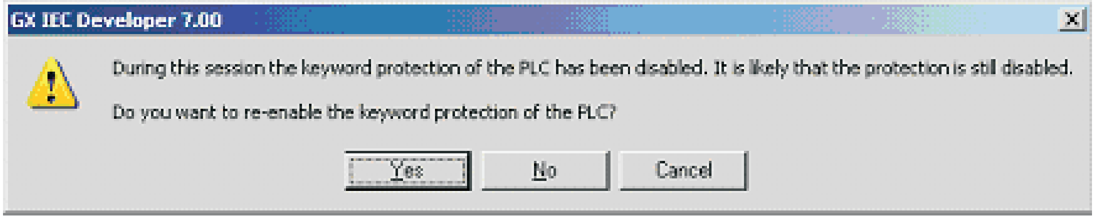
GX IEC Developer 7.00

Operation completed successfully.

Fig. 4-22:
Enabling Protection

When the project is about to be closed, and all of the following conditions are met, you will be warned that the PLC is likely to be in an unprotected state, and you will be offered to re-protect the PLC:

- the PLC is protected with two keywords
- keyword protection was disabled at least once while the project was open
- keyword protection was not re-enabled after the last disablement by:
  - calling the menu item **Online** → **PLC Keyword** → **Enable Protection**
  - registering new keywords



During this session the keyword protection of the PLC has been disabled. It is likely that the protection is still disabled.

Do you want to re-enable the keyword protection of the PLC?

Fig. 4-23: *Warning that the PLC is likely to be in an unprotected state*

4.9.2 Protected Symbolic Information

The symbolic project information of a project for the A/QnA/Q series cannot be protected by the PLC keyword. Using the function of password protection for symbolic information enables the user to prevent access to these data by unauthorized users.

NOTES

The use of password protection for symbolic data increases the time to download/upload of the symbolic information by 30–35 % on an average configuration.

Symbolic information download to a FX PLC is not supported by GX IEC Developer.

When downloading symbolic information to the PLC, the user is asked to enter a password that will be required later to upload the symbolic information from the PLC again. This password may be the same as the PLC keyword, but there is no correspondence between those two passwords. Changing the PLC keyword does not effect the symbolic information password, and vice versa.
The set password even is not related to the project (it is not stored in the project file), it is only related to the symbolic information on the PLC.

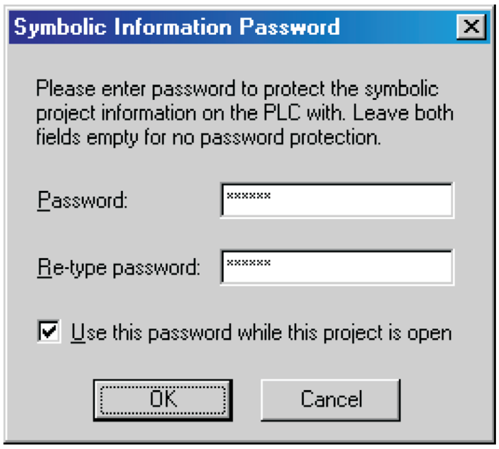
NOTE

Only the read access is password protected. Symbolic information can still be deleted or overwritten on the PLC without a password, since the PLC keyword does not protect those actions. This protection can only be achieved by hardware protection. Of course the symbolic information files can be read from the PLC as well, but they have no meaning to a user not knowing the password and the encryption algorithm.

Symbolic information can be downloaded by the following functionalities:

- Download project (→ section 8.6.6)
- Online change (→ section 8.7.3)
- Download symbolic information to PLC (→ section 8.6.2)

When using any of the three functionalities, the user is asked to enter a password.



Symbolic Information Password

Please enter password to protect the symbolic project information on the PLC with. Leave both fields empty for no password protection.

Password: ******

Re-type password: ******

☑ Use this password while this project is open

Fig. 4-24:
Symbolic Information Password

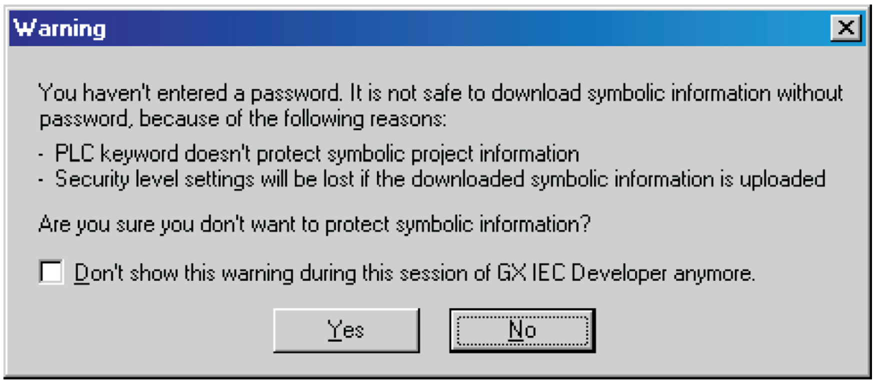
Password: Entry of the new password.

Re-type password: Repeat entry of the new password.

Use this password while this project is open: Select this option to activate the entered password for the time the project is opened.

The user may choose to use the entered password during the project is open, i.e. the password will not be asked for every download until another project is opened.
This remembrance possibility suppresses the password dialogue for every download or online change, so users that download changes to their projects frequently will not have to enter the password every time during one session.
When uploading symbolic information, the password is always asked for.

It is still possible to download symbolic information without protection. If the user does not enter a password, a warning will be displayed to confirm that the user really does not want to protect the symbolic data.



Warning

You haven't entered a password. It is not safe to download symbolic information without password, because of the following reasons:

- PLC keyword doesn't protect symbolic project information
- Security level settings will be lost if the downloaded symbolic information is uploaded

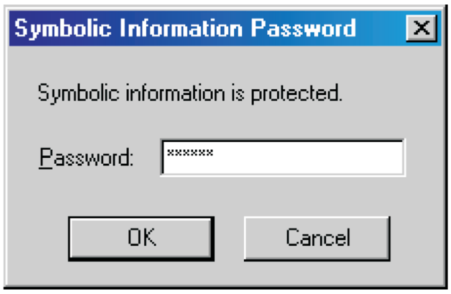
Are you sure you don't want to protect symbolic information?

□ Don't show this warning during this session of GX IEC Developer anymore.

Fig. 4-25:
Warning

There is the option to disable this warning for the rest of the GX IEC Developer session. In that case an empty password will be accepted without requestion.

When uploading protected symbolic information from the PLC, the password has to be entered by the user.



Symbolic Information Password

Symbolic information is protected.

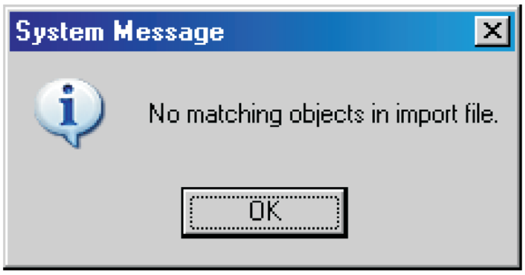
Password: ******

Fig. 4-26:
Dialogbox to query the required password

If the correct password is entered, the symbolic information will be uploaded. Otherwise symbolic upload cannot be executed and binary upload will be offered instead.
If the PLC's program is protected by keyword (PLC keyword) and this keyword has not been input since the project was opened, the user will be prompted to enter the PLC keyword (standard operation for binary upload) (→ page 4-27).

NOTE The protection of the symbolic information files results in changed file formats, which older GX IEC Developer versions cannot handle. The file format of unprotected symbolic information is the same as in older versions.

The following message will be displayed in an older GX IEC Developer version if attempting to upload protected symbolic information:


System Message

No matching objects in import file.

Fig. 4-27:
System Message

Improved Import/Export security

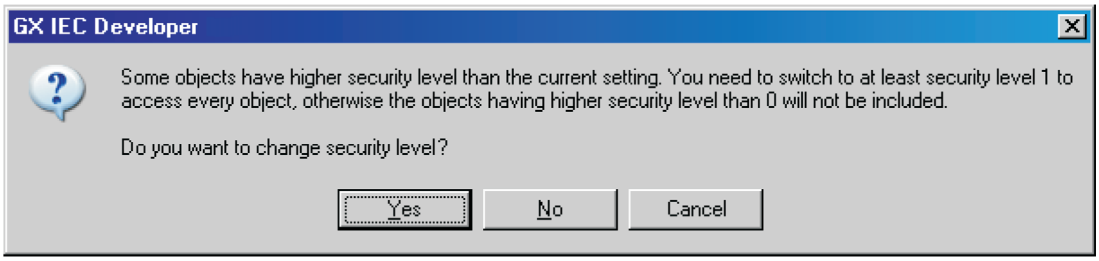
In case a project contains protected POUs and the user wants to make a symbolic download of the project, the following mechanisms are applied.

The dialogue contains two edit fields for entering the passwords, and the two entered passwords have to match to be accepted. Otherwise the user is asked to enter them again.

Normally, when exporting project elements to a file or the clipboard, the objects with a higher security level than the actual state are not included in the export stream and no warning will be given to the user. This behaviour does not break security, but it may lead to misunderstandings. But symbolic information download (which is based on export) would export every needed object, including the ones the user has no access to. This behaviour clearly would break security. A person not having access to some objects could download symbolic project information to a PLC, and then upload it to a new project, making the formerly protected objects accessible.

For this reason file and clipboard export are working as follows:

- If the actual security level is lower than the needed security level to export every object, a warning will be displayed to inform the user about this issue. The user will be asked if he wants to change the security level.

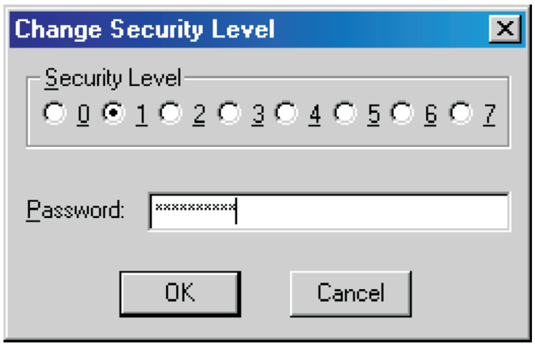

GX IEC Developer

Some objects have higher security level than the current setting. You need to switch to at least security level 1 to access every object, otherwise the objects having higher security level than 0 will not be included.

Do you want to change security level?

Fig. 4-28: *Security level warning*

- When the user chooses **Yes**, the following dialogue will be shown to change the security level.



Change Security Level

Security Level: ○ 0 ● 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Password: **********

Fig. 4-29:
Change Security Level

- If the user chooses **No** in the message box shown in Fig. 4-28, the inaccessible objects will not be exported.
- Choosing **Cancel** in in the message box shown in Fig. 4-28, the process using the export functionality will be cancelled.

If the user decides to change the security level, but then cancels the **Change Security Level** dialogue or sets a security level still not high enough to export every object, the message box shown in Fig. 4-28 will be displayed again, until the needed security level is set or **No** is chosen in this message box.

NOTE

Drag & drop will not display the question message box, because it is a “silent” operation. Showing a message box after starting to drag would interrupt the drag & drop process and the drop could not be executed. Export in this case is based on the actual security level.

4.10 Printing Out Project Data

How to configure the printer

① Select **Printer Setup** in the **Project** menu. This opens the Windows **Print Setup** dialogue box showing the current default printer and its parameters. The changes and settings you can make in this dialogue box depend on the printer connected to your computer.

NOTE For full details, consult the Windows User's Guide and your printer manual.

How to print an object

① Select any object in the Project Navigator or open the window of the object to be printed.

② Select **Print Preview** in the **Object** menu to open the print preview display. The display gives you quite a precise preview of what the printout will look like. If the printout includes more than one page, you can navigate through the pages via the **Page** buttons. You can **Zoom In** (increase) or **Zoom Out** (decrease) the view size via the **Zoom** buttons. You can cancel printing by clicking the **Close** button.

③ Click the **Print** button.
Or:

Press the key combination Alt P.

The **Print** dialogue box is displayed.

④ Select the appropriate print options.

Selecting the **Setup** button opens the **Print Setup** dialogue box, in which you can set the parameters of the default printer or select another printer.

⑤ Start the printout with the **OK** button.

If you do not need to preview the printout, you can directly print any selected or opened object. In the **Object** menu select **Print**.

How to print project data

① Select **Print Preview** in the **Project** menu to open the print preview display. The display gives you quite a precise preview of what the printout will look like. If the printout includes more than one page, you can navigate through the pages via the **Page** buttons. You can **Zoom In** (increase) or **Zoom Out** (decrease) the view size via the **Zoom** buttons. You can cancel printing by clicking the **Close** button.

② Click the **Print** button.
Or:

Press the key combination Alt P.

The **Print** dialogue box is displayed.

③ Select the appropriate print options.

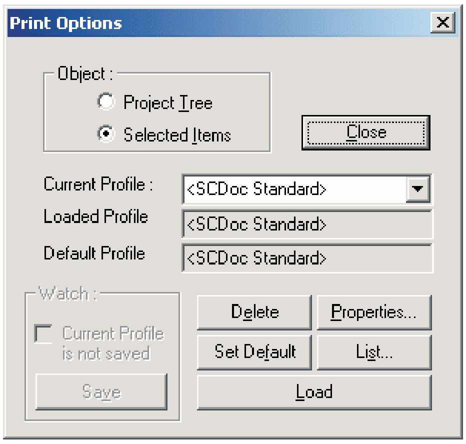
Selecting the **Setup** button opens the **Print Setup** dialogue box, in which you can set the parameters of the default printer or select another printer.

④ Start the printout with the **OK** button.

If you do not need to preview the printout, you can directly print any selected or opened object. In the **Project** menu select **Print**.

4.10.1 Print options for project data

The content and layout of the project data printout can be customised extensively. Different print options settings can be saved as profiles. In the **Project** menu select **Print Options** to open the **Print Options** dialogue.


Print Options

Object :
○ Project Tree
● Selected Items

Current Profile : <SCDoc Standard>
Loaded Profile <SCDoc Standard>
Default Profile <SCDoc Standard>

Watch :
□ Current Profile is not saved

Fig. 4-30:
Print Options

Dialogue items

Object: Select whether only the current **Project Tree** or the currently **Selected Items** according to the current profile are printed.

Current Profile: Type or select (from the combo box) the name of the profile that you want to load, save, set as default, or delete.

Loaded Profile: Displays the currently loaded profile. If this field remains blank, the loaded profile was changed and not yet saved.

Default Profile: Displays the default profile for every printout. The default profile can be specified via the **Set Default** button.

Watch: Indicates whether any property setting of the loaded profile was changed, but the current profile is not saved. In this case the **Loaded Profile** field remains blank.

Save: If any property setting of the loaded profile was changed, this button becomes active. Click the **Save** button to save the current profile under the name entered in the **Current Profile** box.

Delete: Deletes the profile entered in the **Current Profile** box. If the currently entered profile does not exist, an error message is returned.

Set Default: Sets the profile entered in the **Current Profile** box to the default profile. If the currently entered profile does not exist, an error message is returned.

Load: Loads the profile entered in the **Current Profile** box. The loaded profile is indicated in the **Loaded Profile** box.

Properties...: Opens the **Document Configuration** dialogue where you can change the property settings of the loaded profile (for details see below). If any property setting of the loaded profile was changed, this button becomes active and the **Loaded Profile** box is cleared.

List...: Opens the **Checked Properties List** where the specified property settings are summarized in a list.

Document Configuration

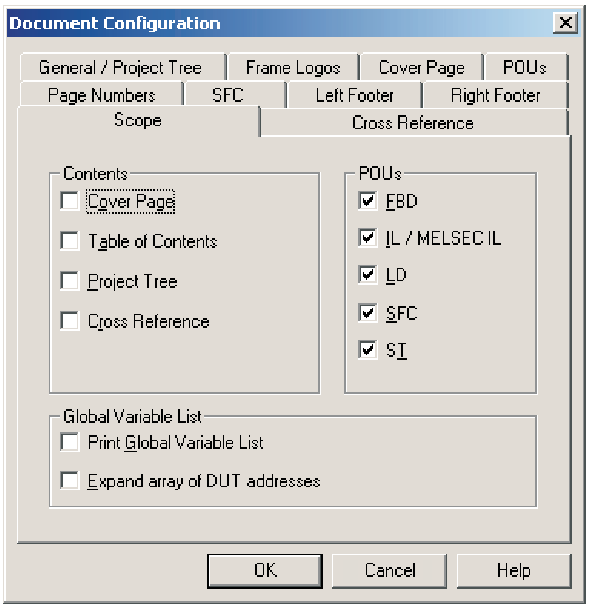
In the **Document Configuration** dialogue all property settings of the loaded profile for the project data printout can be specified.

How to open the Document Configuration dialogue:

① In the **Project** menu select **Print Options....** to open the **Print Options** dialogue.

② In the **Print Options** dialogue click on the **Properties...** button.

The **Document Configuration** dialogue opens:



Contents
- ☐ Cover Page
- ☐ Table of Contents
- ☐ Project Tree
- ☐ Cross Reference

POUs
- ☑ FBD
- ☑ IL / MELSEC IL
- ☑ LD
- ☑ SFC
- ☑ ST

Global Variable List
- ☐ Print Global Variable List
- ☐ Expand array of DUT addresses

Fig. 4-31:
Document Configuration for printouts

The Document Configuration dialogue includes the following property sheet tabs:

Scope

Selects the object class to be documented. This configured scope is related primarily with the selected objects in the main view.

The following objects can be selected for the printout:

- Cover Page
- Table of Contents
- Project Tree
- Cross Reference

POUs (FBD, IL/MELSEC IL, LD, SFC, ST): If FBD, IL, SFC and/or LD are checked, only the selected objects of this type will be documented.

The scope defined by each selection may be restricted through the detailed configuration of the items. The items may be configured via the other tabs.

Global Variable List: Check the option **Print Global Variable List** to print the Global Variable List.

When you choose the option **Print Global Variable List** for printing or previewing of Array of DUT variables, only basic informations about the addresses of the array members are shown:

- WORD start address – address of first array element’s first non-Boolean member
- WORD offset – address difference between two array elements’ WORD start addresses
- Bit designation – whether to store Boolean members mapped onto non-Boolean devices
- BIT start address – address of first array element’s first Boolean member
- BIT offset – address difference between two array elements’ BIT start addresses

Additionally checking the print option **Expand array of DUT addresses** provides the possibility to print the addresses of every member of an Array of DUT variable.
By default the option is turned off, because the listing of all the addresses will make the list very long.

Cross Reference

The following objects can be selected for the printout:

- Call Tree
- Global Variables
  The following selections are possible if **Global Variables** is selected:
  a) Used Read/Write variables (the cross reference for used read/write variables can be documented either for variable names or all addresses or a user-specified address range).
  b) External declarations

General/Project Tree

The following options can be selected for the printout:

- General: Zoom condensed or normal
- Project Tree: Actual State or Expanded
- LD and FBD: Not separated input/output pin directly connected.
  If activated: After the calculation of the layout in FBD and LD an additional calculation will determine all functions and function blocks which have formal input or output pins (only input or output elements directly connected). This option avoids a page break within the function and ensures that the input or output will print the function and function block with their input or output elements on the next page.

Frame Logos

You can specify several bitmap graphics files (*.bmp) and a title for the printout appearing on every page in the header and footer:

- Header: Logo Left, Logo Right, Title
  Enter or browse for a bitmap graphics file to be printed.
- Footer: Logo Bottom
  Enter or browse for a bitmap graphics file to be printed.

Cover Page

You can specify several text entries and a logo (*.bmp) for the printout appearing on the cover page:

- Logo: Enter or browse for a bitmap graphics file (*.bmp) to be printed.
- Project, Comment, Prj. Number, Prj. Manager, PLC Type: Enter text for the respective entry fields.

Left Footer

The following fields can be edited directly for the printout:

- Rev, Change, Date, Name
- Date, Drawn, Appr., Rel.

The captions for the entry fields can be edited by clicking on the respective caption button.

Right Footer

Enter any information into the blank entry fields.

POUs

The following options can be selected for the printout:

- Header
  If selected, the POU header for each selected POU will be printed.
- Local Cross Reference
  If selected the local reference for each selected POU will be printed.
- Body
  If selected, the following objects can be selected from the section Networks: Network Cross Reference, Network Variable Lists, Cross ref./var. list after network
- Either Multiple object on a page or Each object on a page
- Use local variables in Cross Reference
- Show header occurrences in Cross Reference

Page Numbers

The following options can be selected for the printout:

- Show page number in cross reference
  (Variable page reference)
- Show page number in split reference

SFC

The following objects can be selected for the printout:

- Main Network
- Macros
- Steps
- Transitions
- Actions
- Table of Content
- More steps/actions/transitions on one sheet

4.11 Cross Reference

The Cross Reference function generates a database listing of all the project variables. Relationships become clearer, it is easier to find certain variables, and unused variables can be identified.

Default settings in the MMP701.INI initialisation file

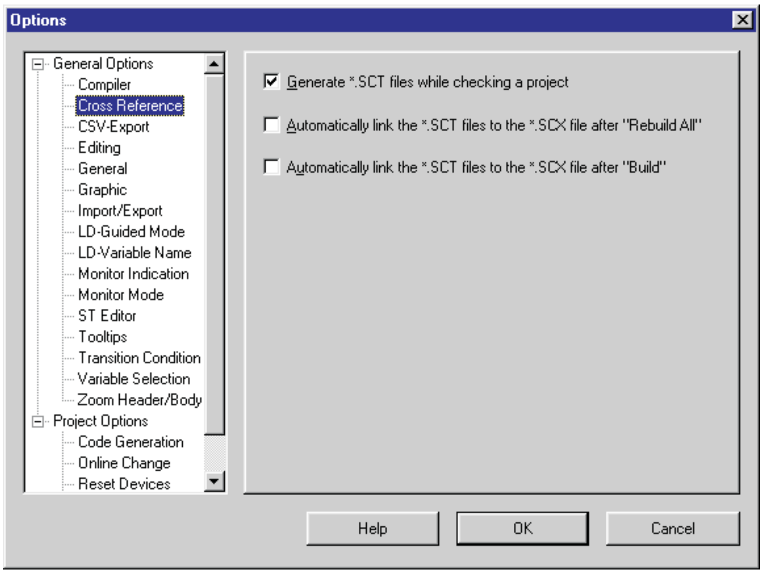
```
[CrossReference]

GenerateSCT=TRUE
```

TRUE: cross reference function enabled.
FALSE: cross reference function disabled.

Cross reference options

In the **Extras - Options - Cross Reference** menu you can set the cross reference options.



☑ Generate *.SCT files while checking a project

☐ Automatically link the *.SCT files to the *.SCX file after "Rebuild All"

☐ Automatically link the *.SCT files to the *.SCX file after "Build"

Fig. 4-32:
Options -
Cross Reference

Generate *.SCT files while checking a project: Activates generation of the SCT files during project checks

Automatically link the *.SCT files to the *.SCX file after 'Rebuild All': Activates automatic linking of the SCT files to the SCX file after compiling the project

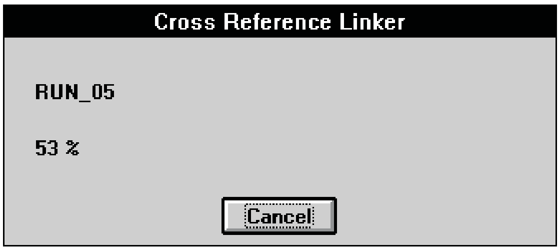
Automatically link the *.SCT files to the .SCX file after 'Build': Activates automatic linking of the SCT files to the SCX file after building the project

To create a cross reference database

① Compile the entire project.

② In the **Project** menu choose the command **Make Cross Reference**.

The program now compiles all the details required for the cross reference database. The procedure is documented in the status window **Cross Reference Linker**. The time needed for this procedure depends on the project length.



Cross Reference Linker

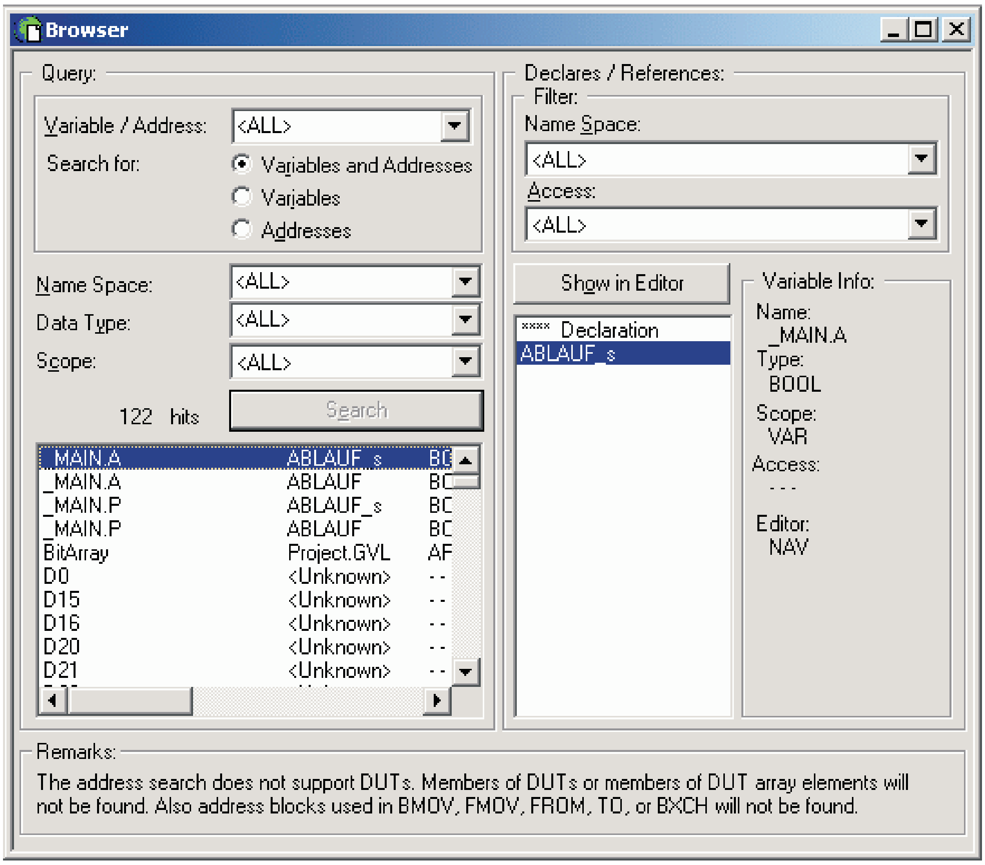
RUN_05

53 %

Fig. 4-33: *Creating the cross reference database*

③ Choose **Browse** in the **Project** menu.

The **Browser** dialogue box is presented, prompting you to enter the search criteria.



Browser

Query:

Variable / Address: <ALL>

Search for: ● Variables and Addresses ○ Variables ○ Addresses

Name Space: <ALL>

Data Type: <ALL>

Scope: <ALL>

122 hits

_MAIN.A	ABLAUF_s	BC
_MAIN.A	ABLAUF	BC
_MAIN.P	ABLAUF_s	BC
_MAIN.P	ABLAUF	BC
BitArray	Project.GVL	AF
D0	<Unknown>	- -
D15	<Unknown>	- -
D16	<Unknown>	- -
D20	<Unknown>	- -
D21	<Unknown>	- -

Declares / References:

Filter:

Name Space: <ALL>

Access: <ALL>

**** Declaration

ABLAUF_s

Variable Info:

Name: _MAIN.A

Type: BOOL

Scope: VAR

Access: - - -

Editor: NAV

Remarks:

The address search does not support DUTs. Members of DUTs or members of DUT array elements will not be found. Also address blocks used in BMOV, FMOV, FROM, TO, or BXCH will not be found.

Fig. 4-34: *Cross reference specification*

Query

Set the search criteria:

Variable / Address: Contains <ALL> if searching for all or input an identifier

Search for: The search for the identifier is for variables and addresses, or only for variables and only for addresses.

Name Space: Object name; e.g. Program Organisation Unit, Global Variable List

Data Type: e.g. BOOL, INT

Scope: Class; e.g. VAR, VAR_GLOBAL

Button: Search: Starts the search. The result of the search is presented in the windows.

hits: Number of matches found

Search result in window

All records that have been found are listed alphabetically.

Declares/References

Depending on the element that is marked in the left-hand window after a search, further details are listed in the right-hand window and in the Variable Info List. You can set further search filters for these details.

Filters

Namespace: Object name

Access: Tells you how the variable is accessed in the projects

Search result in window

Declaration: Program organisation units in which the variable was declared

References: Program organisation units in which the variable was inserted

Search result in the Variable Info List

Name: Name of variable

Type: Data type

Scope: Class; e.g. VAR, VAR_GLOBAL

Access: Access type; e.g. Read, Write, Read and Write

Editor: Programming language, network number in editor

NOTES

The browser functionality regarding search of variables is performed by their hardware address in Mitsubishi format and their address ranges. The search for a hardware address finds all variables that share the searched address.

In the case of an array element with non static index, i.e. the index part contains one or more indices that do not represent a number, the whole array range is used for the address search.

DUT or ARRAY of DUT can only be searched for the start address.

Functions that have address ranges passed by start address and block size like in BMOV, FMOV, FROM, TO, or BXCH can only be searched for the start address.

4.12 Verifying Projects

By using the function of verifying projects you can verify an opened project with an other project. This function is based on the functionality of **Verify** in the menu **Project - Transfer** (see also section 8.7.2).

While the functionality of online verify compares the opened project to the one downloaded to the PLC, the functionality of project verify will compare the currently opened project to another project on the PC, selected by the user.

To verify two projects choose the menu **Project**, the sub menu **Other** and then **Verify**. In the opened dialogue box select the desired project to be compared with the current project.

The following project elements will be verified:

- Boot settings (boot files) (if the PLC is of the Q/QnA series, except Q Remote I/O projects)
- PLC-Parameters
- Main program
- Sub program (if allowed for the PLC type)
- SFC program (if allowed for the PLC type)
- SFC program file header (if SFC is allowed for the PLC type, and it is of the Q/QnA series)
- Timer initial values of main and sub program (if the PLC is of the A series)
- Counter initial values of main and sub program (if the PLC is of the A series)

The opened project will not be closed (only its database file will be closed). Therefore you do not have to save the project before starting the comparison.

The comparison is separated into two parts: the PLC-parameter and the program. Both projects have to include parameters to start the parameter verification, otherwise the parameters are incomparable.

NOTES

The program comparison is not only based on the compiled code, so the two projects do not have to be compiled to start the program verification. The projects will also be considered to have been changed if unsaved changes exist in any open editors. The user will be warned, if either project has been changed since last build.

The for comparison selected project may not be opened in the current or another GX IEC Developer instance. Consequently a project cannot be compared to itself.

An improperly closed project cannot be compared to the opened project. It must be repaired first by opening it in GX IEC Developer normally.

A project last saved with a newer version of GX IEC Developer cannot be compared to the currently opened project.

A project last saved with a version of GX IEC Developer older than 4.00 cannot be compared to the currently opened project. It must be converted first by opening it in GX IEC Developer normally.

Parameters of a project last saved with a version of GX IEC Developer older than 6.00 cannot be compared to the current project's parameters. The project must be converted first by opening it in GX IEC Developer normally.

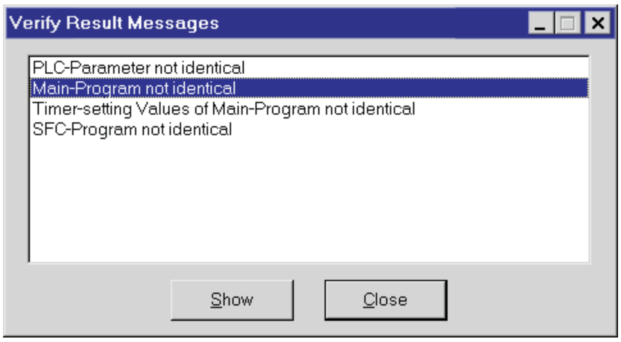
Only projects for the same CPU type can be compared.

NOTE

In case of a Q Remote I/O project, the verify function compares only parameters, because a Q Remote I/O project does not have program data. If the comparison is successful (no error happens), the possible results are:
- PLC-Parameter identical
- PLC-Parameter not identical

If the parameters cannot be compared due to an error, the error will be displayed in a message box and the **Verify Result Messages** dialogue will not be shown.

After the chosen project is read, the comparison results will be displayed in the dialogue **Verify Result Messages** shown below. This is the same dialogue as for online verify (see also section 8.7.2).



PLC-Parameter not identical
Main-Program not identical
Timer-setting Values of Main-Program not identical
SFC-Program not identical

Fig. 4-35: *Dialogue box*

The PLC parameters are compared in whole (having the incomparable blocks removed), so there may be two cases: identical or not identical. The program will be compared part-by-part (e.g. main program, SFC program, etc.), and for each not identical part a message is displayed in the dialogue. If the comparison shows no difference in any of the program's elements, then only the message "Program identical" is displayed. Additionally, there are messages indicating if an element exists in either project but not in the other.

The following table lists the messages related to the various project elements. These messages are the same as for online verify.

Element	“Identical” message	“Not identical” message
Boot file	"Boot file identical"	"Boot file not identical"
Parameter	"PLC-Parameter identical"	"PLC-Parameter not identical"
Main program	"Program identical"	"Main-Program not identical"
Sub program		"Sub-Program not identical"
SFC program		"SFC-Program not identical"
SFC program file header		"SFC-Program file header not identical"
Timer initial values (main program)		"Timer-setting Values of Main-Program not identical"
Timer initial values (sub program)		"Timer-setting Values of Sub-Program not identical"
Counter initial values (main program)		"Counter-setting Values of Main-Program not identical"
Counter initial values (sub program)		"Counter-setting Values of Sub-Program not identical"

Tab. 4-1: *List of Messages related to the command* ***Verify***

The following table lists the messages displayed if an element is missing from either or both projects. These messages are the same as for online verify.

Element	Absence	Message	Remark
Boot file	Missing from either or both projects	"Boot file incomparable"	If the selected project is older than version 6.00 or the boot file of either project was deleted. If an error happens at the reading of one of the boot files because the boot file is corrupted. If the parameter of one of the projects is security protected.
Parameter	Missing from chosen project	"PLC-Parameter incomparable"	It is impossible to open a project without parameters, so the parameters are never missing in the currently open project.
Main program	Missing from either or both projects	"Program incomparable"	If there is no main program in either or both projects a comparison is not possible.
Sub program	Missing from either project	"Sub-Program exists in current, but not in selected project" or "Sub-Program exists in selected, but not in current project"	If there is no sub program in either of the projects a comparison is not possible.
Sub program	Missing from both projects	—	If there is no sub program in both of the projects a comparison will not be performed.
SFC program	Missing from either project	"SFC-Program exists in current, but not in selected project" or "SFC-Program exists in selected, but not in current project"	If there is no SFC program in either of the projects a comparison is not possible.
SFC program	Missing from both projects	—	If there is no SFC program in both of the projects a comparison will not be performed.
SFC program file header	Not covered	—	A SFC program header always exists in Q/QnA projects if a SFC program exists.
Timer/counter initial values	Not covered	—	These initial values always exist in projects for A series CPUs.

Tab. 4-2: *List of Messages displayed if an element is missing*

Detected differences in the program code can be shown in a **Main** or **Sub Program Difference** window via the **Show** button. The verify results are listed as recompiled MELSEC IL, the differences are highlighted in the error color. The left part of the window lists the current project code, the right part of the window lists the selected project code.

When the **Main** or **Sub Program Difference** window is opened the following menu items are available in the **Tools** menu:

Next Difference: **Next Difference** or the tabulator key will position the text in the window so that the first line of the next difference is the top most line. The top most line will be selected. The according calltree, number of network and for IL and ST the line in the network of the selected line are shown in the status bar. If no next difference exists, the menu item is greyed.

Previous Difference: **Previous Difference** or the shift tabulator key will position the text in the window so that the first line of the previous difference is the top most line. The top most line will be selected. The according calltree, number of network and for IL and ST the line in the network of the selected line are shown in the status bar. If no previous difference exists, the menu item is greyed.

Goto Editor: **Goto Editor** or a double click on a selected line will open the according POU body. In IL/ST the according line and in LD/FBD the according network is positioned to the visible area and marked with the browser colour in the network column.
If no POU body relates to the selected line a message window is opened to inform you of this circumstance. The message text is “The selected line belongs to code generated for task or program handling. No editor could be opened.”

Show Step Number: This menu item is checked as default. If it is checked, the step number for each instruction for the code of the current project and the selected project are displayed.
If it is not checked, the step numbers are not displayed. If a line is selected and the step number is not displayed, the line is marked with a small box in front of the instruction.

5 Parameters

5.1 PLC Parameter

The command **Parameter - PLC** in the Project Navigator provides options for configuring your system.

The CPU type can be changed by selecting the command **Change PLC Type** in the **Project** menu or by clicking on the icon.

In the dialogbox **Change PLC Type** you can select a PLC series and a PLC type.

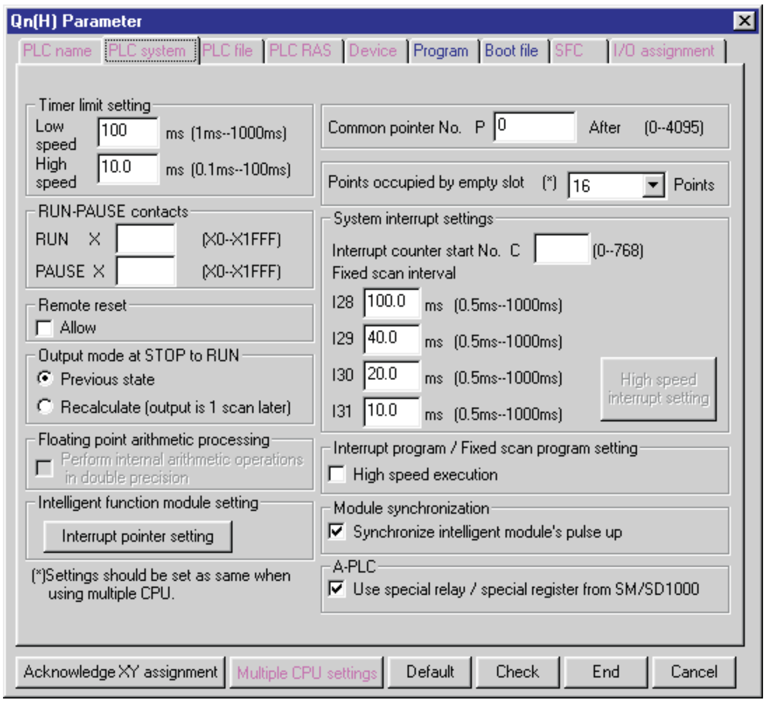
- If you change the PLC series and/or the PLC type and confirm with **OK**, all parameters for the new PLC type will be reset to defaults.
- If you do not change the PLC series/PLC type but want all previous parameter changes to become inactive, confirm with **OK**.
- If you do not want to change anything, click **Cancel**.

The default settings produce trouble-free program performance for most applications. If you are using a network you must configure it with **Parameter - Network** before starting work, however. If the configuration parameter settings lead to problems in your project, GX IEC Developer will display an explanatory warning message.

NOTE

> The PLC parameters are set automatically for the PLC type you select. Some of the parameters explained below are not relevant for all PLC types, and they are thus not displayed when some specific PLC types are selected.

For any information or particular explanation required for parameter setting, refer to the corresponding CPU user's manual and programming manual to pre-design them. The setting items are assigned to the tab screen purpose-by-purpose.



Timer limit setting
Low speed 100 ms (1ms--1000ms)
High speed 10.0 ms (0.1ms--100ms)

RUN-PAUSE contacts
RUN X (X0-X1FFF)
PAUSE X (X0-X1FFF)

Remote reset
□ Allow

Output mode at STOP to RUN
● Previous state
○ Recalculate (output is 1 scan later)

Floating point arithmetic processing
□ Perform internal arithmetic operations in double precision

Intelligent function module setting

(*)Settings should be set as same when using multiple CPU.

Common pointer No. P 0 After (0--4095)

Points occupied by empty slot (*) 16 Points

System interrupt settings
Interrupt counter start No. C (0--768)
Fixed scan interval
I28 100.0 ms (0.5ms--1000ms)
I29 40.0 ms (0.5ms--1000ms)
I30 20.0 ms (0.5ms--1000ms)
I31 10.0 ms (0.5ms--1000ms)

Interrupt program / Fixed scan program setting
□ High speed execution

Module synchronization
☑ Synchronize intelligent module's pulse up

A-PLC
☑ Use special relay / special register from SM/SD1000

Fig. 5-1: *Parameters*

5.1.1 Setting the PLC Parameters

The following table lists the PLC parameter setting items on a series basis. Set the parameters as required.

	A	QnA	System Q					FX
			Q00J/Q00/ Q01	QnH	QnPH	QnPRH	Remote I/O	
PLC name	○	●	●	●	●	●	○	●
PLC system (1)	●	●	●	●	●	●	●	●
PLC system (2)	○	○	○	○	○	○	○	●
Positioning	○	○	○	○	○	○	○	●
PLC file	○	●	●	●	●	●	○	○
PLC RAS (1)	●	●	●	●	●	●	●	○
PLC RAS (2)	○	○	○	●	●	●	○	○
Device	●	●	●	●	●	●	○	●
Program	○	●	●	●	●	●	○	○
Boot file	○	●	●	●	●	●	○	○
SFC	○	●	●	●	●	●	○	○
I/O assignment	●	●	●	●	●	●	●	●
Memory capacity	●	○	○	○	○	○	○	●
Operational settings	○	○	○	○	○	○	●	○
Serial	○	○	●	○	○	○	○	○

Tab. 5-1: PLC parameter setting items

5.1.2 Common Notes on Parameters

Parameter display

This part describes the setting item tabs and network parameter setting items.

Example ▽

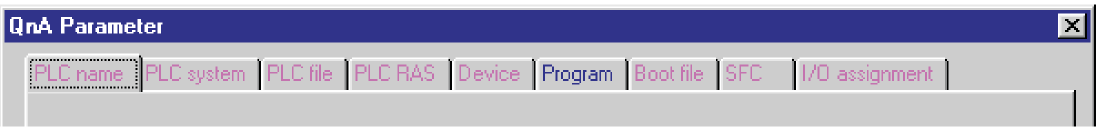


Fig. 5-2: QnA Parameter

△

The meanings of the symbols are the same also when displayed with network parameters.

- Red: The PLC does not operate until data is set. (Data is not set.)
- Blue: Data is set.
- Magenta: The PLC operates without setting data or with the default. (Data is set.)
- Dark blue: Data is set.

Common buttons for parameters

This part describes the settings common to PLC and network parameters.

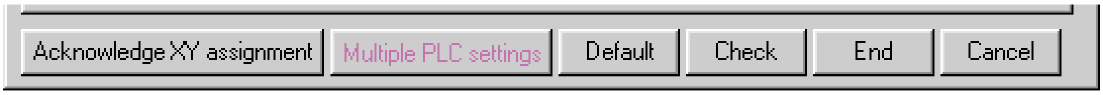


Fig. 5-3: Common buttons for parameters

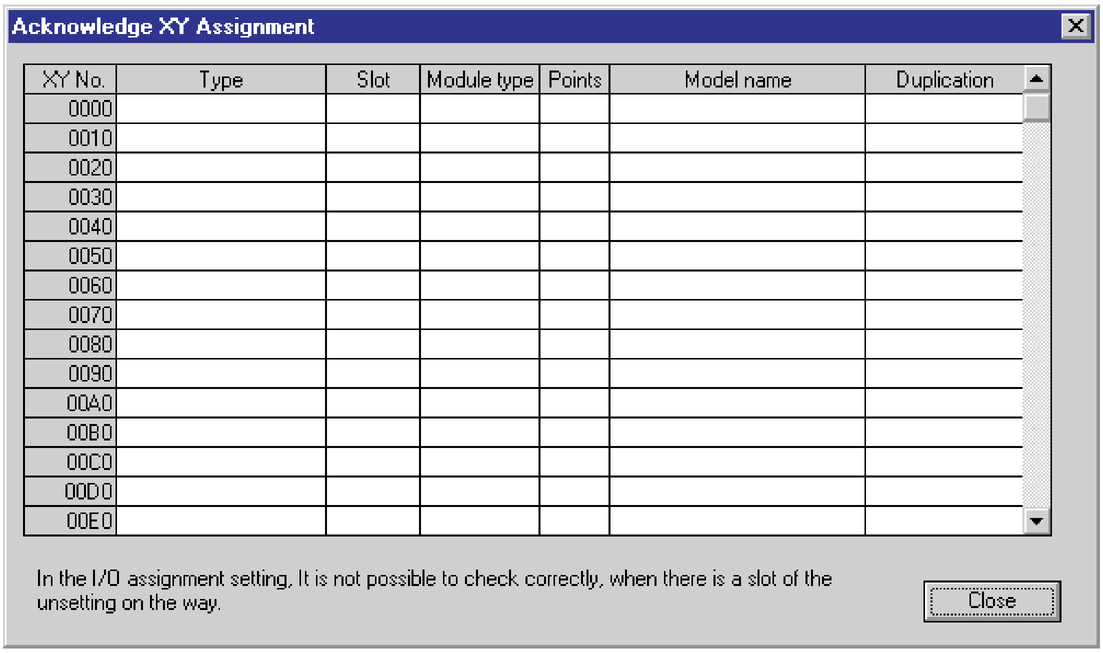
Default button
This button returns all of the set items or values to the previous settings. This button is available only for the currently open tab.

Check button
Use this button to check the set items or values to see if they are correct. This button is available only for the currently open tab.

End setup
The **End** button defines the set items or values and terminates the setting.

Cancel button
Cancels the set items and terminates editing.

Acknowledge X/Y Assignment button
This button is available for the models A2ACPU(S1), A3ACPU, A2AS(S1), A2AS-S30, A2AS-S60, A2UCPU(S1), A2USCPU(S1), A2USHCPU(S1), A3UCPU, and QnACPU, QCPU(Q mode). It is used to confirm the X/Y assignment numbers of the data set in the I/O assignment setting.



XY No.	Type	Slot	Module type	Points	Model name	Duplication
0000						
0010						
0020						
0030						
0040						
0050						
0060						
0070						
0080						
0090						
00A0						
00B0						
00C0						
00D0						
00E0						

In the I/O assignment setting, It is not possible to check correctly, when there is a slot of the unsetting on the way.

Fig. 5-4: XY Assignment

Display	Description	
I/O assignment	Displays I/O assignment information.	
MINI	Displays I/O assignment information and MINI automatic refresh setting information.	
CC-Link	Displays I/O assignment information and CC-Link automatic refresh setting information.	
NET (II)	Master station:	Displays I/O assignment information and refresh parameter information.
	Local Station:	Displays I/O assignment information and refresh parameter information.
NET/10	PLC-to-PLC network, remote I/O network:	Displays network parameter information.

Tab. 5-2: *Display by type*

The priority of display is shown below:

① I/O assignment (AnACPU, AnUCPU, Q/QnA series)

② Fourth MELSECNET network refresh parameter (AnUCPU, Q/QnA series)

③ Third MELSECNET network refresh parameter (AnUCPU, Q/QnA series)

④ Second MELSECNET network refresh parameter (AnUCPU, Q/QnA series)

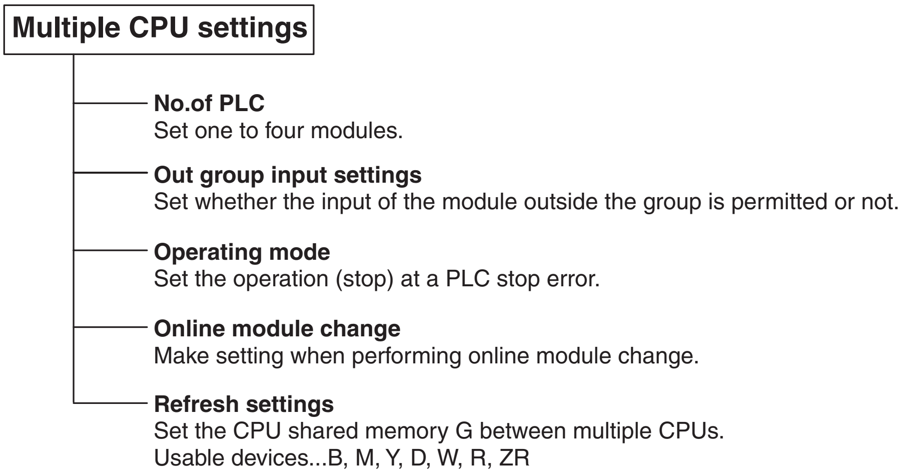
⑤ First MELSECNET network refresh parameter (AnUCPU, Q/QnA series)

⑥ MELSECNET/MINI refresh (AnACPU, AnUCPU, QCPU (A mode)/QnA series)

⑦ CC-Link remote I/O (Q/QnA series)

The Duplicate Error area displays the first duplicate item detected in checking the items according to the priority.

Multiple CPU setting button
Setting cannot be made for Q00J/Q00/Q01CPU and remote I/O projects.



Multiple CPU settings

No.of PLC
Set one to four modules.

Out group input settings
Set whether the input of the module outside the group is permitted or not.

Operating mode
Set the operation (stop) at a PLC stop error.

Online module change
Make setting when performing online module change.

Refresh settings
Set the CPU shared memory G between multiple CPUs.
Usable devices...B, M, Y, D, W, R, ZR

Fig. 5-5: *Multiple PLC setting*

No. of PLC
Clicking the **Default** button in I/O assignment keeps the multiple PLC setting unchanged. When the set number of PLCs is one, a multiple CPU system configuration is not set.

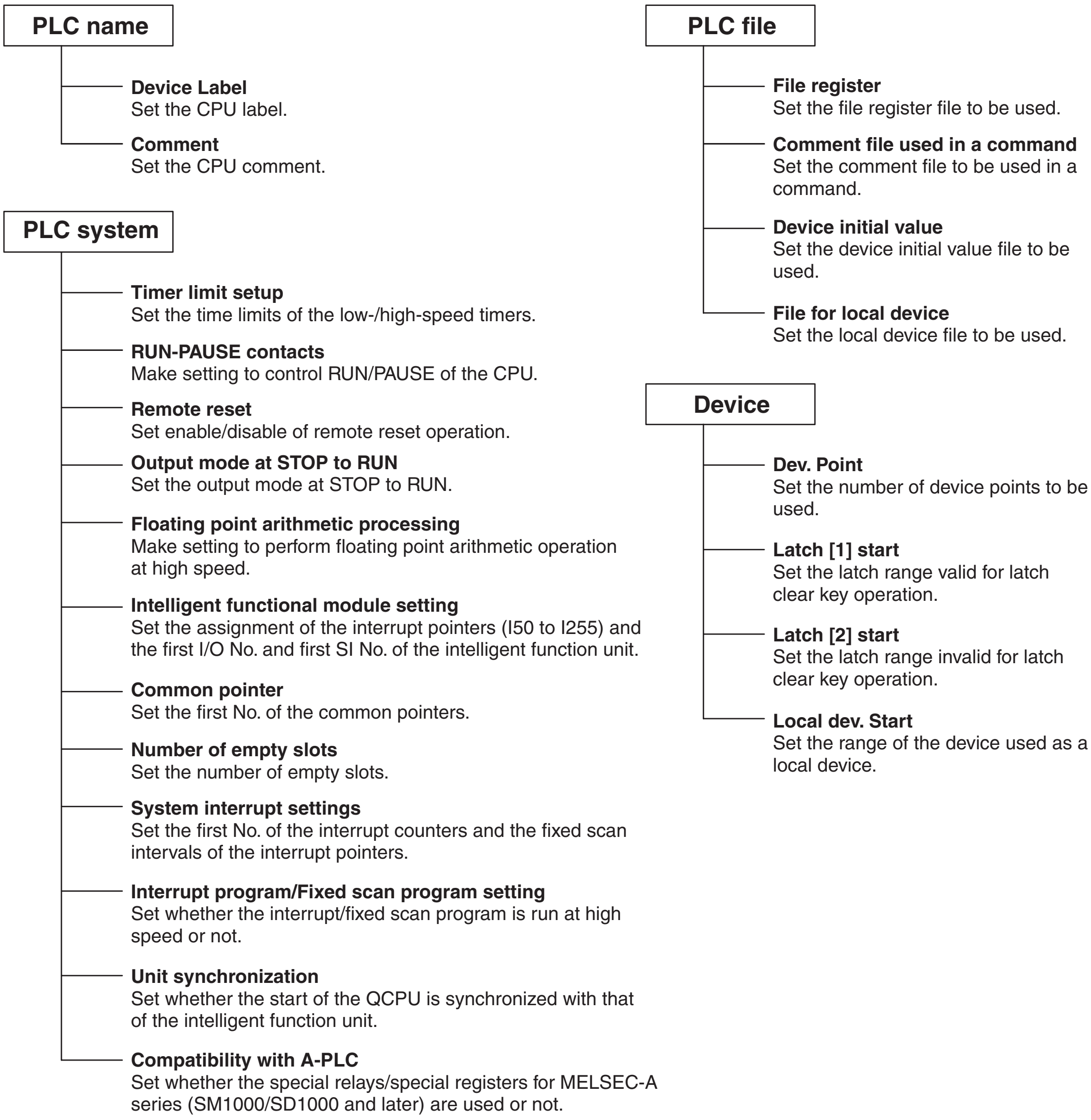
Out group input settings
When the modules are not grouped, they are all put under the control PLC of the PLC No. 1. **OPERATION ERROR** appears if access is made to any module outside the groups.

Refresh settings
The maximum input range of each module is 0 to 2048 points.
Set the total number of all modules to within 4096 points.

Diversion of multiple PLC parameter button
Specify the project to be diverted from the **Open Project** screen. Setting the source project shows the multiple PLC parameter diversion dialogue box. Confirm the message and execute.

5.1.3 PLC Parameter Item Lists

PLC parameter item list of the QnH



PLC name

- **Device Label**
  Set the CPU label.
- **Comment**
  Set the CPU comment.

PLC system

- **Timer limit setup**
  Set the time limits of the low-/high-speed timers.
- **RUN-PAUSE contacts**
  Make setting to control RUN/PAUSE of the CPU.
- **Remote reset**
  Set enable/disable of remote reset operation.
- **Output mode at STOP to RUN**
  Set the output mode at STOP to RUN.
- **Floating point arithmetic processing**
  Make setting to perform floating point arithmetic operation at high speed.
- **Intelligent functional module setting**
  Set the assignment of the interrupt pointers (I50 to I255) and the first I/O No. and first SI No. of the intelligent function unit.
- **Common pointer**
  Set the first No. of the common pointers.
- **Number of empty slots**
  Set the number of empty slots.
- **System interrupt settings**
  Set the first No. of the interrupt counters and the fixed scan intervals of the interrupt pointers.
- **Interrupt program/Fixed scan program setting**
  Set whether the interrupt/fixed scan program is run at high speed or not.
- **Unit synchronization**
  Set whether the start of the QCPU is synchronized with that of the intelligent function unit.
- **Compatibility with A-PLC**
  Set whether the special relays/special registers for MELSEC-A series (SM1000/SD1000 and later) are used or not.

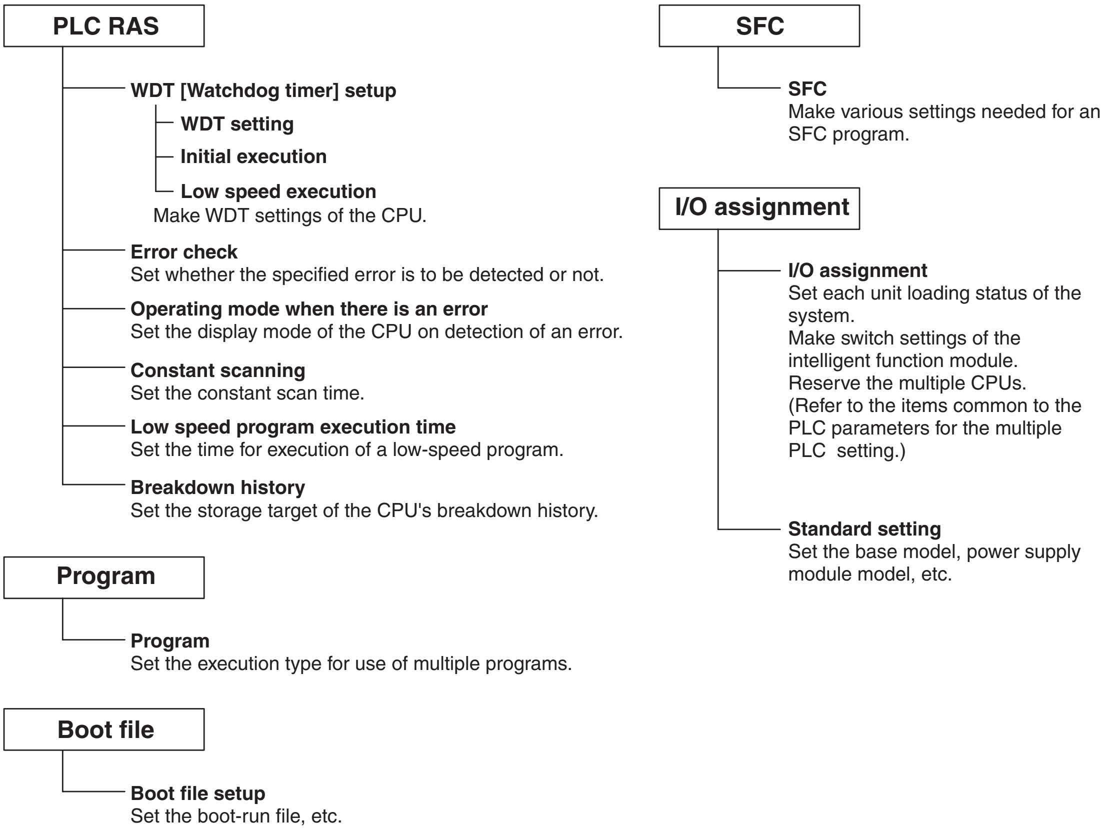
PLC file

- **File register**
  Set the file register file to be used.
- **Comment file used in a command**
  Set the comment file to be used in a command.
- **Device initial value**
  Set the device initial value file to be used.
- **File for local device**
  Set the local device file to be used.

Device

- **Dev. Point**
  Set the number of device points to be used.
- **Latch [1] start**
  Set the latch range valid for latch clear key operation.
- **Latch [2] start**
  Set the latch range invalid for latch clear key operation.
- **Local dev. Start**
  Set the range of the device used as a local device.

Fig. 5-6: *PLC parameter item list of the QnH (1)*



PLC RAS

- **WDT [Watchdog timer] setup**
  - **WDT setting**
  - **Initial execution**
  - **Low speed execution**

  Make WDT settings of the CPU.
- **Error check**
  Set whether the specified error is to be detected or not.
- **Operating mode when there is an error**
  Set the display mode of the CPU on detection of an error.
- **Constant scanning**
  Set the constant scan time.
- **Low speed program execution time**
  Set the time for execution of a low-speed program.
- **Breakdown history**
  Set the storage target of the CPU's breakdown history.

Program

- **Program**
  Set the execution type for use of multiple programs.

Boot file

- **Boot file setup**
  Set the boot-run file, etc.

SFC

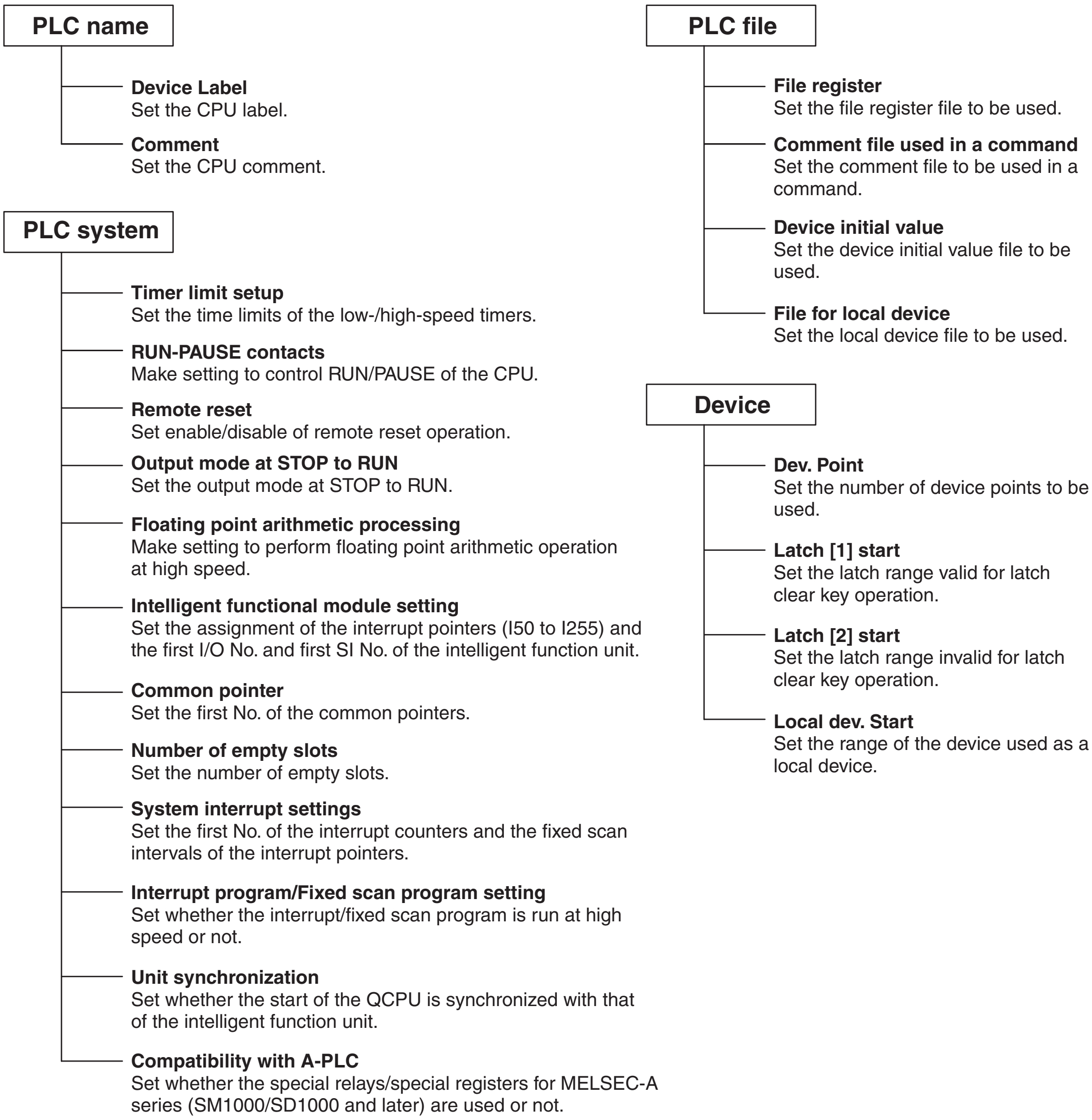
- **SFC**
  Make various settings needed for an SFC program.

I/O assignment

- **I/O assignment**
  Set each unit loading status of the system.
  Make switch settings of the intelligent function module.
  Reserve the multiple CPUs.
  (Refer to the items common to the PLC parameters for the multiple PLC setting.)
- **Standard setting**
  Set the base model, power supply module model, etc.

Fig. 5-6: *PLC parameter item list of the QnH (2)*

PLC parameter item list of the QnPH



PLC name

- **Device Label**
  Set the CPU label.
- **Comment**
  Set the CPU comment.

PLC system

- **Timer limit setup**
  Set the time limits of the low-/high-speed timers.
- **RUN-PAUSE contacts**
  Make setting to control RUN/PAUSE of the CPU.
- **Remote reset**
  Set enable/disable of remote reset operation.
- **Output mode at STOP to RUN**
  Set the output mode at STOP to RUN.
- **Floating point arithmetic processing**
  Make setting to perform floating point arithmetic operation at high speed.
- **Intelligent functional module setting**
  Set the assignment of the interrupt pointers (I50 to I255) and the first I/O No. and first SI No. of the intelligent function unit.
- **Common pointer**
  Set the first No. of the common pointers.
- **Number of empty slots**
  Set the number of empty slots.
- **System interrupt settings**
  Set the first No. of the interrupt counters and the fixed scan intervals of the interrupt pointers.
- **Interrupt program/Fixed scan program setting**
  Set whether the interrupt/fixed scan program is run at high speed or not.
- **Unit synchronization**
  Set whether the start of the QCPU is synchronized with that of the intelligent function unit.
- **Compatibility with A-PLC**
  Set whether the special relays/special registers for MELSEC-A series (SM1000/SD1000 and later) are used or not.

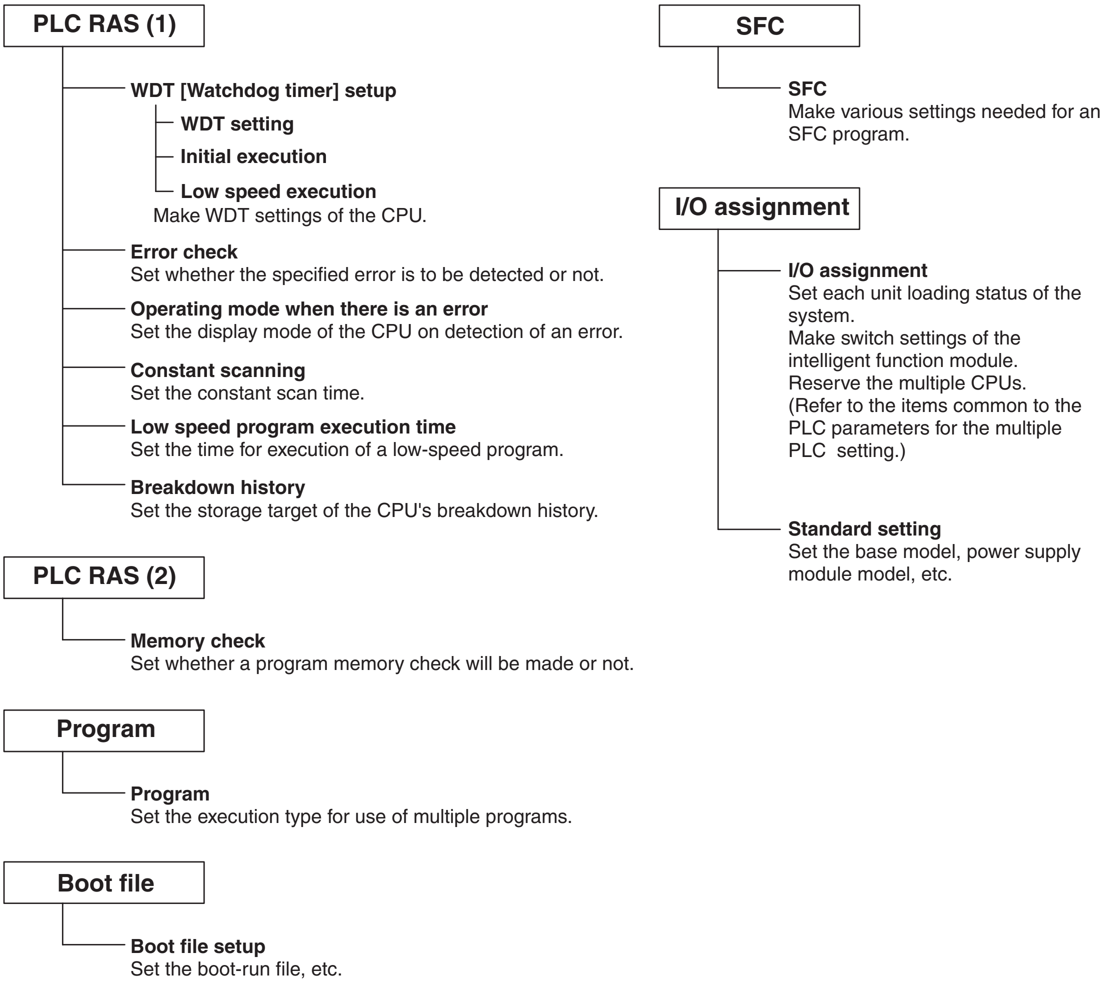
PLC file

- **File register**
  Set the file register file to be used.
- **Comment file used in a command**
  Set the comment file to be used in a command.
- **Device initial value**
  Set the device initial value file to be used.
- **File for local device**
  Set the local device file to be used.

Device

- **Dev. Point**
  Set the number of device points to be used.
- **Latch [1] start**
  Set the latch range valid for latch clear key operation.
- **Latch [2] start**
  Set the latch range invalid for latch clear key operation.
- **Local dev. Start**
  Set the range of the device used as a local device.

Fig. 5-7: *PLC parameter item list of the QnPH (1)*



PLC RAS (1)

- **WDT [Watchdog timer] setup**
  - **WDT setting**
  - **Initial execution**
  - **Low speed execution**

  Make WDT settings of the CPU.
- **Error check**
  Set whether the specified error is to be detected or not.
- **Operating mode when there is an error**
  Set the display mode of the CPU on detection of an error.
- **Constant scanning**
  Set the constant scan time.
- **Low speed program execution time**
  Set the time for execution of a low-speed program.
- **Breakdown history**
  Set the storage target of the CPU's breakdown history.

PLC RAS (2)

- **Memory check**
  Set whether a program memory check will be made or not.

Program

- **Program**
  Set the execution type for use of multiple programs.

Boot file

- **Boot file setup**
  Set the boot-run file, etc.

SFC

- **SFC**
  Make various settings needed for an SFC program.

I/O assignment

- **I/O assignment**
  Set each unit loading status of the system.
  Make switch settings of the intelligent function module.
  Reserve the multiple CPUs.
  (Refer to the items common to the PLC parameters for the multiple PLC setting.)
- **Standard setting**
  Set the base model, power supply module model, etc.

Fig. 5-7: PLC parameter item list of the QnPH (2)

PLC parameter item list of the QnPRH

PLC name

- **Device Label**
  Set the CPU label.
- **Comment**
  Set the CPU comment.

PLC system

- **Timer limit setup**
  Set the time limits of the low-/high-speed timers.
- **RUN-PAUSE contacts**
  Make setting to control RUN/PAUSE of the CPU.
- **Remote reset**
  Set enable/disable of remote reset operation.
- **Output mode at STOP to RUN**
  Set the output mode at STOP to RUN.
- **Floating point arithmetic processing**
  Make setting to perform floating point arithmetic operation at high speed.
- **Intelligent functional module setting**
  Set the assignment of the interrupt pointers (I50 to I255) and the first I/O No. and first SI No. of the intelligent function unit.
- **Common pointer**
  Set the first No. of the common pointers.
- **Number of empty slots**
  Set the number of empty slots.
- **System interrupt settings**
  Set the first No. of the interrupt counters and the fixed scan intervals of the interrupt pointers.
- **Interrupt program/Fixed scan program setting**
  Set whether the interrupt/fixed scan program is run at high speed or not.
- **Unit synchronization**
  Set whether the start of the QCPU is synchronized with that of the intelligent function unit.
- **Compatibility with A-PLC**
  Set whether the special relays/special registers for MELSEC-A series (SM1000/SD1000 and later) are used or not.

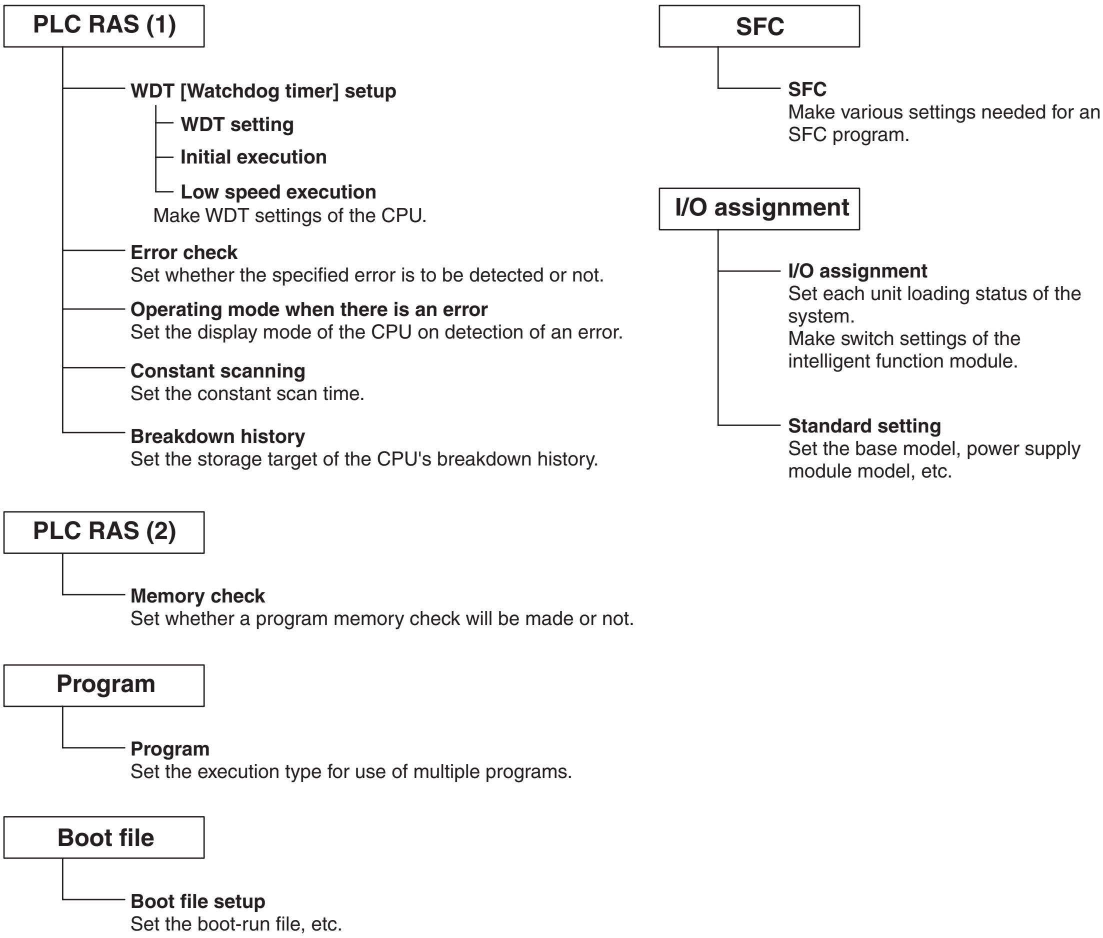
PLC file

- **File register**
  Set the file register file to be used.
- **Comment file used in a command**
  Set the comment file to be used in a command.
- **Device initial value**
  Set the device initial value file to be used.
- **File for local device**
  Set the local device file to be used.

Device

- **Dev. Point**
  Set the number of device points to be used.
- **Latch [1] start**
  Set the latch range valid for latch clear key operation.
- **Latch [2] start**
  Set the latch range invalid for latch clear key operation.
- **Local dev. Start**
  Set the range of the device used as a local device.

Fig. 5-8: *PLC parameter item list of the QnPRH (1)*



PLC RAS (1)

WDT [Watchdog timer] setup
- **WDT setting**
- **Initial execution**
- **Low speed execution**

Make WDT settings of the CPU.

Error check
Set whether the specified error is to be detected or not.

Operating mode when there is an error
Set the display mode of the CPU on detection of an error.

Constant scanning
Set the constant scan time.

Breakdown history
Set the storage target of the CPU's breakdown history.

PLC RAS (2)

Memory check
Set whether a program memory check will be made or not.

Program

Program
Set the execution type for use of multiple programs.

Boot file

Boot file setup
Set the boot-run file, etc.

SFC

SFC
Make various settings needed for an SFC program.

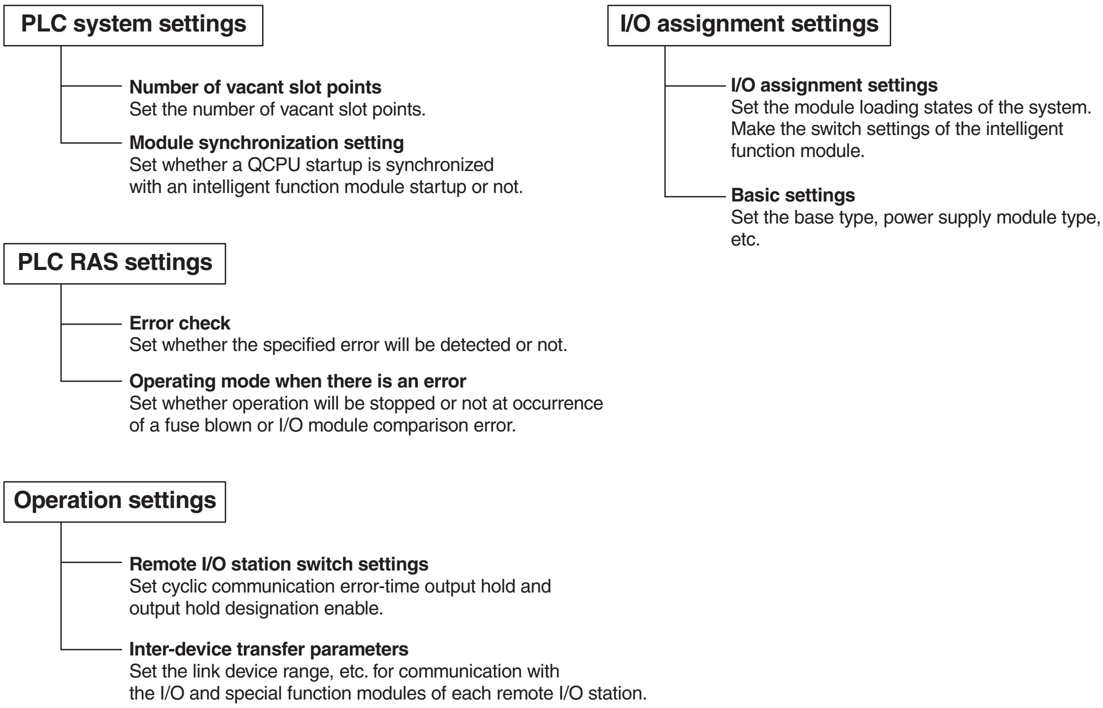
I/O assignment

I/O assignment
Set each unit loading status of the system.
Make switch settings of the intelligent function module.

Standard setting
Set the base model, power supply module model, etc.

Fig. 5-8: *PLC parameter item list of the QnPRH (2)*

PLC parameter item list for Q series remote I/O station



PLC system settings

- **Number of vacant slot points**
  Set the number of vacant slot points.
- **Module synchronization setting**
  Set whether a QCPU startup is synchronized with an intelligent function module startup or not.

PLC RAS settings

- **Error check**
  Set whether the specified error will be detected or not.
- **Operating mode when there is an error**
  Set whether operation will be stopped or not at occurrence of a fuse blown or I/O module comparison error.

Operation settings

- **Remote I/O station switch settings**
  Set cyclic communication error-time output hold and output hold designation enable.
- **Inter-device transfer parameters**
  Set the link device range, etc. for communication with the I/O and special function modules of each remote I/O station.

I/O assignment settings

- **I/O assignment settings**
  Set the module loading states of the system. Make the switch settings of the intelligent function module.
- **Basic settings**
  Set the base type, power supply module type, etc.

Fig. 5-9: *PLC parameter item list for Q series remote I/O station*

PLC parameter item list of the Q00J/Q00/Q01CPU

PLC name

- **Device Label**
  Set the CPU label.
- **Comment**
  Set the CPU comment.

PLC file (Setting is disabled for Q00J)

- **File register**
  Set the file register file to be used.
  (Setting is fixed)

PLC system

- **Timer limit setup**
  Set the time limits of the low-/high-speed timers.
- **RUN-PAUSE contacts**
  Make setting to control RUN/PAUSE of the CPU.
- **Remote reset**
  Set enable/disable of remote reset operation.
- **Output mode at STOP to RUN**
  Set the output mode at STOP to RUN.
- **Number of empty slots**
  Set the number of empty slots.
- **System interrupt settings**
  Set the first No. of the interrupt counters and the fixed scan intervals of the interrupt pointers.
- **Interrupt program/Fixed scan program setting**
  Set whether the interrupt/fixed scan program is run at high speed or not.
- **Unit synchronization**
  Set whether the start of the QCPU is synchronized with that of the intelligent function unit.

Device

- **Dev. Point**
  Set the number of device points to be used.
- **Latch [1] start**
  Set the latch range valid for latch clear key operation.
- **Latch [2] start**

PLC RAS

- **WDT [Watchdog timer] setup**
  - **WDT setting**
    Make WDT settings of the CPU.
- **Error check**
  Set whether the specified error is to be detected or not.
- **Operating mode when there is an error**
  Set the display mode of the CPU on detection of an error.
- **Constant scanning**
  Set the constant scan time.

I/O assignment

- **I/O assignment**
  Set each unit loading status of the system.
  Make switch settings of the intelligent function module.
  Reserve the multiple CPUs.
- **Standard setting**
  Set the base model, power supply module model, etc.

Boot file

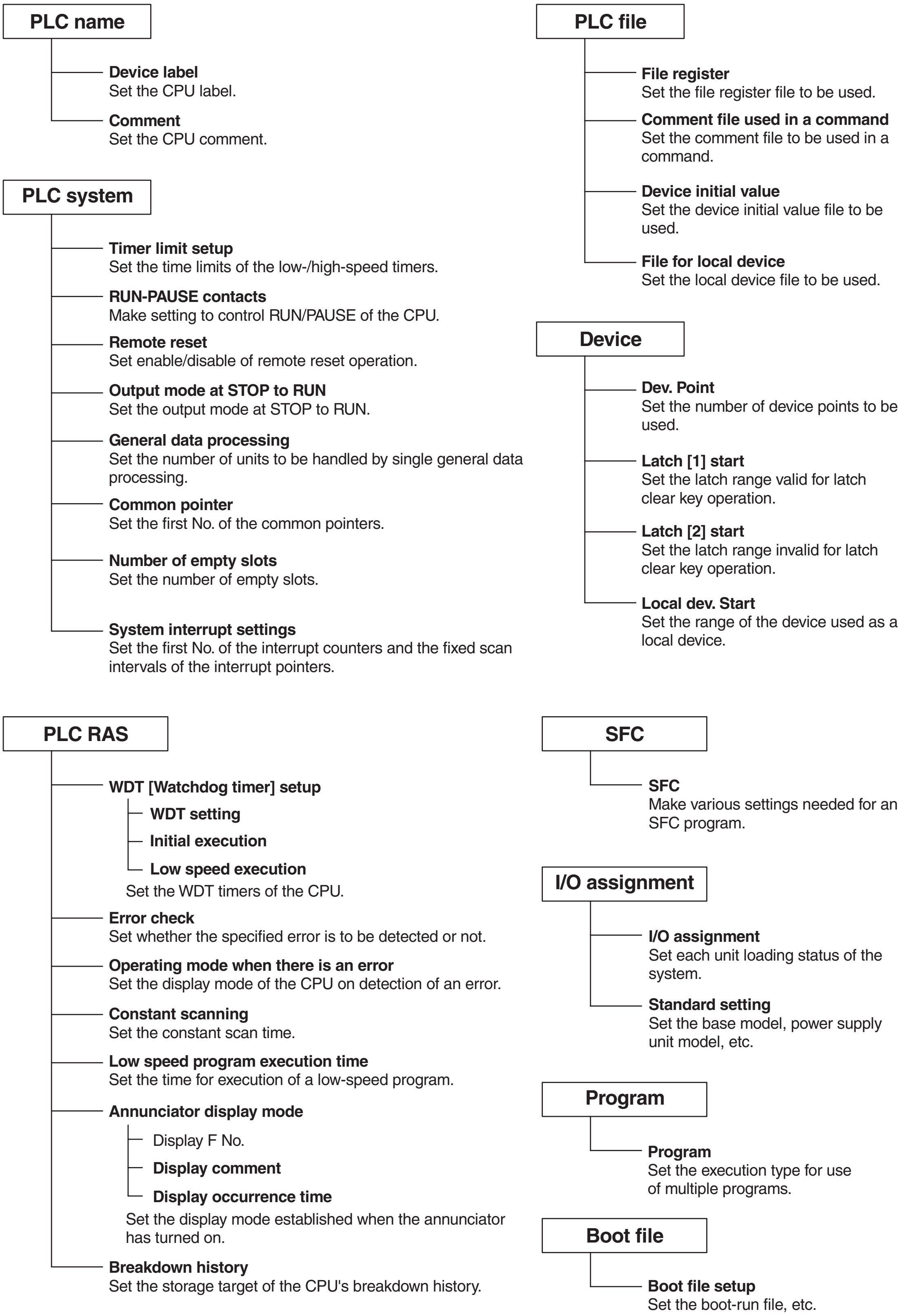
- **Boot file setup**
  Set the boot-run file, etc.

Serial

- **Baud rate**
  Set the baud rate.
- **Sum check**
  Set the sum check.
- **Transmission wait time**
  Set the transmission wait time.
- **RUN write setting**
  Set whether online program change can be made or not.

Fig. 5-10: *PLC parameter item list of the Q00J/Q00/Q01CPU*

PLC parameter item list of the QnA series



PLC name

- **Device label**
  Set the CPU label.
- **Comment**
  Set the CPU comment.

PLC system

- **Timer limit setup**
  Set the time limits of the low-/high-speed timers.
- **RUN-PAUSE contacts**
  Make setting to control RUN/PAUSE of the CPU.
- **Remote reset**
  Set enable/disable of remote reset operation.
- **Output mode at STOP to RUN**
  Set the output mode at STOP to RUN.
- **General data processing**
  Set the number of units to be handled by single general data processing.
- **Common pointer**
  Set the first No. of the common pointers.
- **Number of empty slots**
  Set the number of empty slots.
- **System interrupt settings**
  Set the first No. of the interrupt counters and the fixed scan intervals of the interrupt pointers.

PLC file

- **File register**
  Set the file register file to be used.
- **Comment file used in a command**
  Set the comment file to be used in a command.
- **Device initial value**
  Set the device initial value file to be used.
- **File for local device**
  Set the local device file to be used.

Device

- **Dev. Point**
  Set the number of device points to be used.
- **Latch [1] start**
  Set the latch range valid for latch clear key operation.
- **Latch [2] start**
  Set the latch range invalid for latch clear key operation.
- **Local dev. Start**
  Set the range of the device used as a local device.

PLC RAS

- **WDT [Watchdog timer] setup**
  - **WDT setting**
  - **Initial execution**
  - **Low speed execution**

  Set the WDT timers of the CPU.
- **Error check**
  Set whether the specified error is to be detected or not.
- **Operating mode when there is an error**
  Set the display mode of the CPU on detection of an error.
- **Constant scanning**
  Set the constant scan time.
- **Low speed program execution time**
  Set the time for execution of a low-speed program.
- **Annunciator display mode**
  - Display F No.
  - **Display comment**
  - **Display occurrence time**

  Set the display mode established when the annunciator has turned on.
- **Breakdown history**
  Set the storage target of the CPU's breakdown history.

SFC

- **SFC**
  Make various settings needed for an SFC program.

I/O assignment

- **I/O assignment**
  Set each unit loading status of the system.
- **Standard setting**
  Set the base model, power supply unit model, etc.

Program

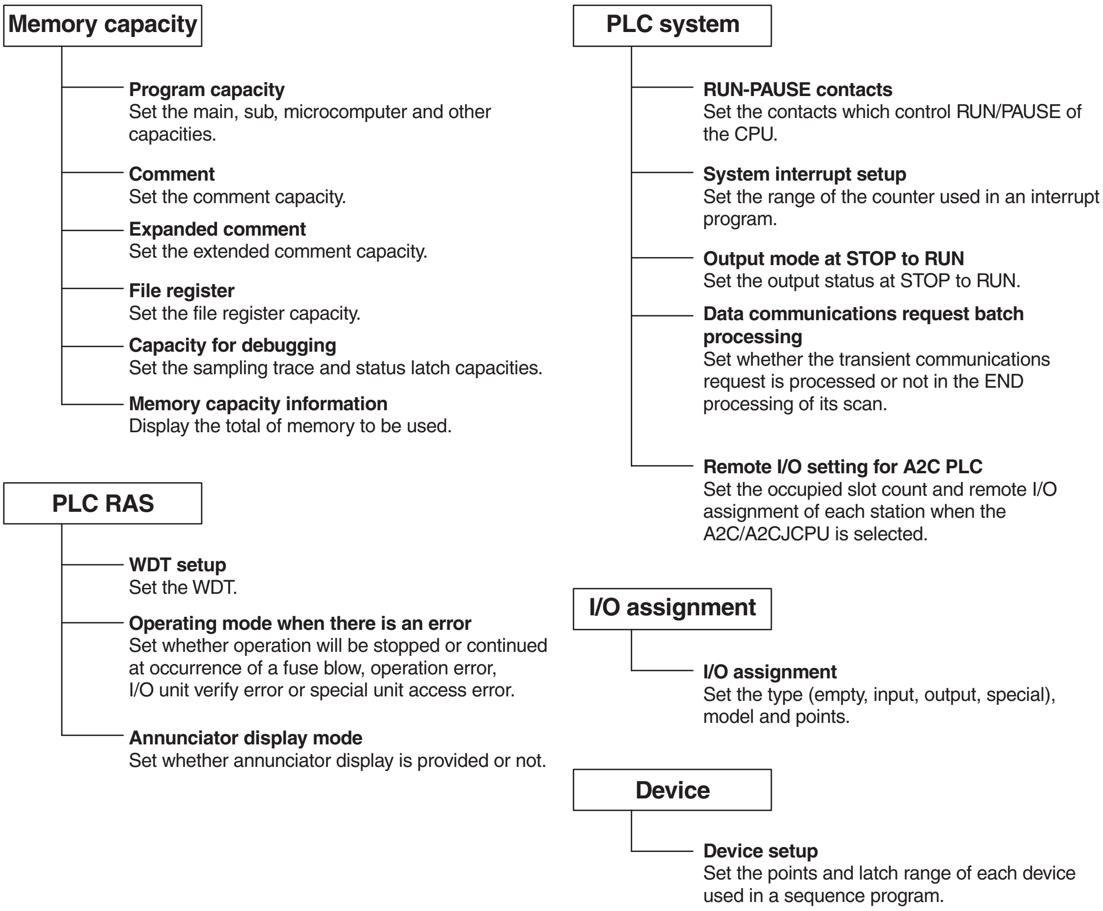
- **Program**
  Set the execution type for use of multiple programs.

Boot file

- **Boot file setup**
  Set the boot-run file, etc.

Fig. 5-11: *PLC parameter item list of the QnA series*

PLC parameter item list of the A series



Memory capacity

- **Program capacity**
  Set the main, sub, microcomputer and other capacities.
- **Comment**
  Set the comment capacity.
- **Expanded comment**
  Set the extended comment capacity.
- **File register**
  Set the file register capacity.
- **Capacity for debugging**
  Set the sampling trace and status latch capacities.
- **Memory capacity information**
  Display the total of memory to be used.

PLC RAS

- **WDT setup**
  Set the WDT.
- **Operating mode when there is an error**
  Set whether operation will be stopped or continued at occurrence of a fuse blow, operation error, I/O unit verify error or special unit access error.
- **Annunciator display mode**
  Set whether annunciator display is provided or not.

PLC system

- **RUN-PAUSE contacts**
  Set the contacts which control RUN/PAUSE of the CPU.
- **System interrupt setup**
  Set the range of the counter used in an interrupt program.
- **Output mode at STOP to RUN**
  Set the output status at STOP to RUN.
- **Data communications request batch processing**
  Set whether the transient communications request is processed or not in the END processing of its scan.
- **Remote I/O setting for A2C PLC**
  Set the occupied slot count and remote I/O assignment of each station when the A2C/A2CJCPU is selected.

I/O assignment

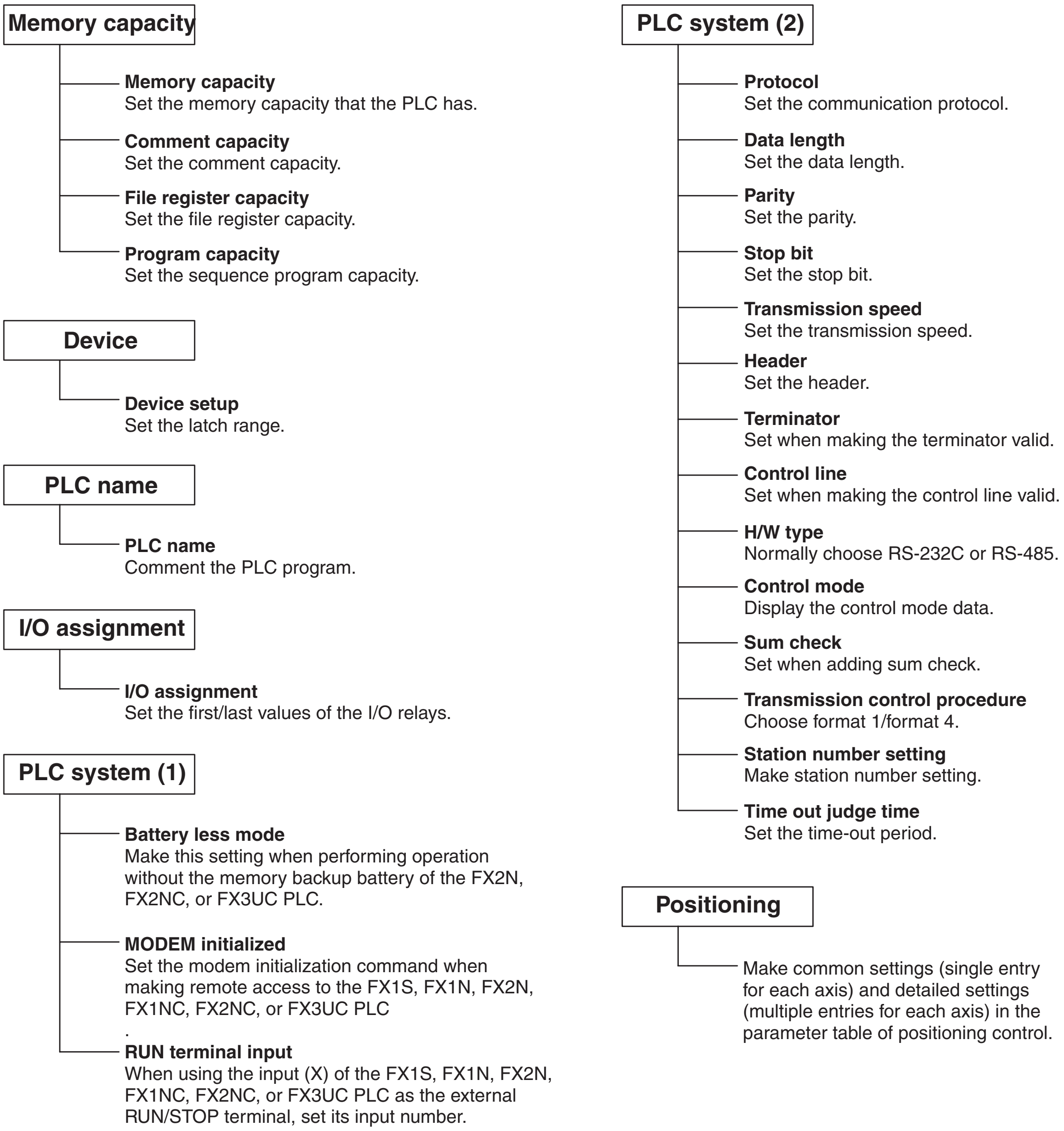
- **I/O assignment**
  Set the type (empty, input, output, special), model and points.

Device

- **Device setup**
  Set the points and latch range of each device used in a sequence program.

Fig. 5-12: *PLC parameter item list of the A series*

PLC parameter item list of the FX series



Memory capacity

- **Memory capacity**
  Set the memory capacity that the PLC has.
- **Comment capacity**
  Set the comment capacity.
- **File register capacity**
  Set the file register capacity.
- **Program capacity**
  Set the sequence program capacity.

Device

- **Device setup**
  Set the latch range.

PLC name

- **PLC name**
  Comment the PLC program.

I/O assignment

- **I/O assignment**
  Set the first/last values of the I/O relays.

PLC system (1)

- **Battery less mode**
  Make this setting when performing operation without the memory backup battery of the FX2N, FX2NC, or FX3UC PLC.
- **MODEM initialized**
  Set the modem initialization command when making remote access to the FX1S, FX1N, FX2N, FX1NC, FX2NC, or FX3UC PLC
  .
- **RUN terminal input**
  When using the input (X) of the FX1S, FX1N, FX2N, FX1NC, FX2NC, or FX3UC PLC as the external RUN/STOP terminal, set its input number.

PLC system (2)

- **Protocol**
  Set the communication protocol.
- **Data length**
  Set the data length.
- **Parity**
  Set the parity.
- **Stop bit**
  Set the stop bit.
- **Transmission speed**
  Set the transmission speed.
- **Header**
  Set the header.
- **Terminator**
  Set when making the terminator valid.
- **Control line**
  Set when making the control line valid.
- **H/W type**
  Normally choose RS-232C or RS-485.
- **Control mode**
  Display the control mode data.
- **Sum check**
  Set when adding sum check.
- **Transmission control procedure**
  Choose format 1/format 4.
- **Station number setting**
  Make station number setting.
- **Time out judge time**
  Set the time-out period.

Positioning

- Make common settings (single entry for each axis) and detailed settings (multiple entries for each axis) in the parameter table of positioning control.

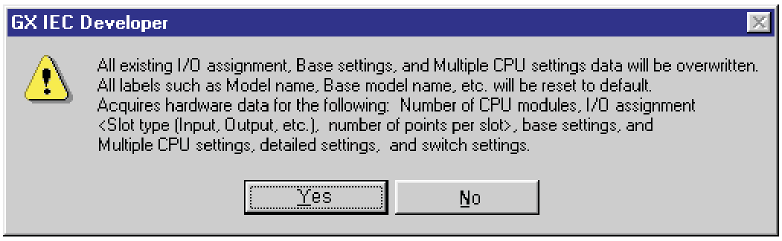
Fig. 5-13: *PLC parameter item list of the FX series*

5.1.4 Reading of PLC Parameter from the PLC

It is also possible to read the PLC parameter from the PLC to the Project.

Read PLC data button
This button is located on the **I/O assignment** tab.

For Q series:
The implementation status is read if the parameter file exists in the PLC. If the unit model, first X/Y, base model, power supply unit model, extension cable model and slot count have been set on the Personal computer, the data will be deleted.
If the parameters have been set on GX IEC Developer, the following dialogue box appears to ask you if the parameter data may be overwritten.



GX IEC Developer

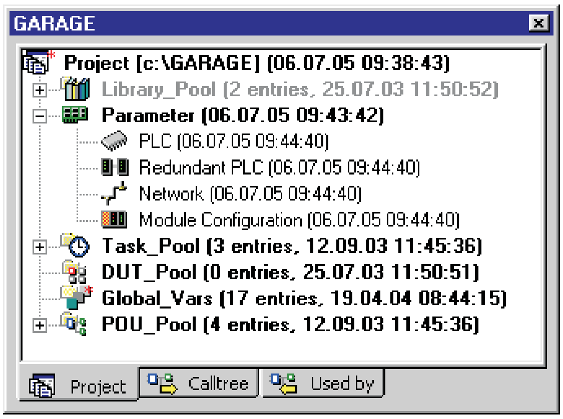
All existing I/O assignment, Base settings, and Multiple CPU settings data will be overwritten.
All labels such as Model name, Base model name, etc. will be reset to default.
Acquires hardware data for the following: Number of CPU modules, I/O assignment <Slot type (Input, Output, etc.), number of points per slot>, base settings, and Multiple CPU settings, detailed settings, and switch settings.

Fig. 5-14: *Note when reading the PLC data*

For QnA series:
When the parameter file exists in the PLC, the parameter file is read.
When the parameter file does not exist in the PLC (implementation read):
After deleting the parameter file in the PLC, set the PLC from RESET to RUN and read the PLC data. If the unit model, first X/Y, base, power supply unit and extension cable have been set on the Personal computer, the data will be deleted.

5.2 Parameters for Redundant PLCs

The command **Parameter - Redundant PLC** in the Project Navigator provides options for configuring a redundant system.



GARAGE

Project [c:\GARAGE] (06.07.05 09:38:43)
Library_Pool (2 entries, 25.07.03 11:50:52)
Parameter (06.07.05 09:43:42)
PLC (06.07.05 09:44:40)
Redundant PLC (06.07.05 09:44:40)
Network (06.07.05 09:44:40)
Module Configuration (06.07.05 09:44:40)
Task_Pool (3 entries, 12.09.03 11:45:36)
DUT_Pool (0 entries, 25.07.03 11:50:51)
Global_Vars (17 entries, 19.04.04 08:44:15)
POU_Pool (4 entries, 12.09.03 11:45:36)

Fig. 5-15:
The Project Navigator window for the QnPRH CPU

Redundant PLC parameter item list of the QnPRH

Redundant Parameters

- **Operation settings**
  - **Start mode setting**
    Set the device status when the system is powered on/reset.
  - **Standby system watch setting**
    Check for a stanby system fault.
    If a fault occurs, error check can be made using the PLC diagnostics.
  - **Debug mode setting**
    Set a start in the debug mode.
  - **Backup mode setting**
    Set the same type of operating status consistency.
- **Tracking settings**
  - **Tracking device settings**
    Set the tracking transfer setting data.
  - **Tracking characteristics settings**
    Set the transfer mode.

Fig. 5-16: *Redundant PLC parameter item list of the QnPRH*

5.3 Network Parameters

The command **Parameter - Network** in the Project Navigator provides options for configuring your networks. Clicking on Parameter - Network opens the following dialogbox.

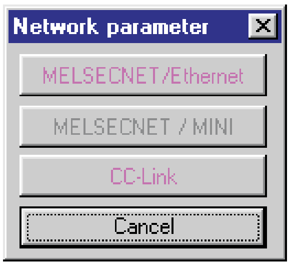


Fig. 5-17:
*Dialogbox **Network parameter***

If the network parameter settings lead to problems in your project, GX IEC Developer will display an explanatory warning message.

NOTE

The PLC parameters are set automatically for the PLC type you select. Some of the parameters explained below are not relevant for all PLC types, and they are thus not displayed when some specific PLC types are selected.

For any information or particular explanation required for network parameter setting, refer to the corresponding CPU user's manual and programming manual to pre-design them. The setting items are assigned to the tab screen purpose-by-purpose.

5.3.1 Setting the Network Parameters

The following table lists the network parameter setting items on a series basis. Set the parameters as required.

Network	A			QnA	System Q	
	AnNCPU	AnACPU	AnUCPU QCPU (A mode)		CPU	Remote I/O
MELSECNET	●	●	●	●	○	○
MELSECNET II	○	●	●	●	○	○
MELSECNET/10	○	○	●	●	●	○
MELSECNET/10H	○	○	○	○	●	○
MELSECNET/MINI	●	○	●	●	○	○
CC-Link	○	○	○	●	●	●
CC-Link Ver. 2	○	○	○	○	●	●
ETHERNET	○	○	○	●	●	●

Tab. 5-3: *Network parameter setting items*

5.3.2 Common Notes on Parameters

About Items Common to the Network Parameters

This section describes operations common to the setting of data link system and network system parameters.

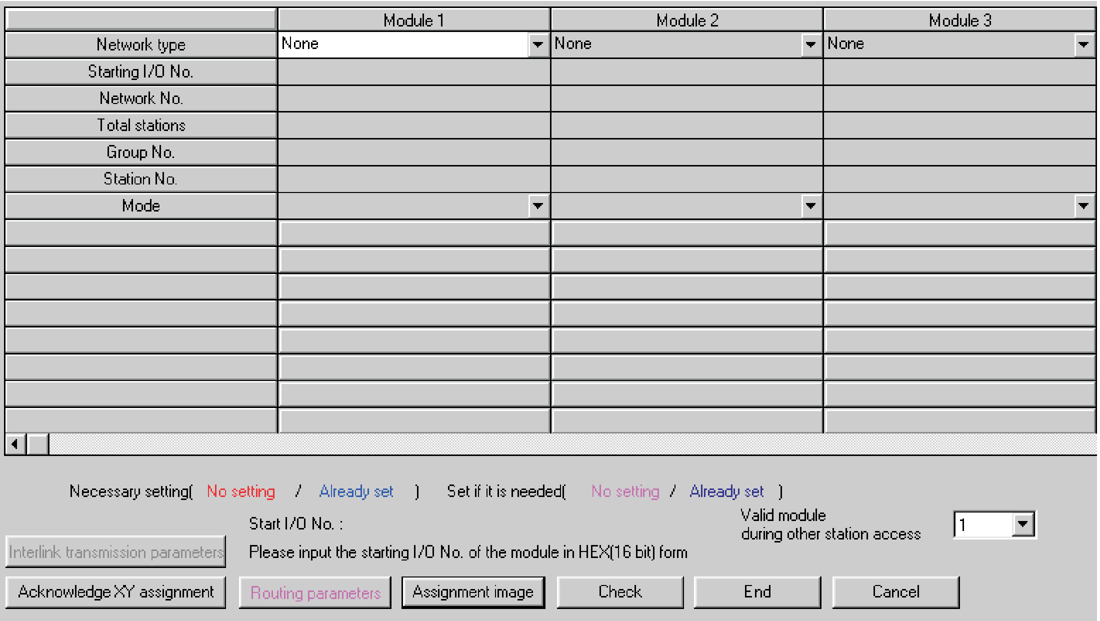
The following operations cannot be performed as those common to the parameters.

- Cut, copy, paste
- **Home** and **End** keys are invalid.

The meanings of the symbols are the same also when displayed with PLC parameters.

- Red: The PLC does not operate until data is set. (Data is not set.)
- Blue: Data is set.
- Magenta: The PLC operates without setting data or with the default. (Data is set.)
- Dark blue: Data is set.

Example ▽ MELSECNET II combined (master station) network range assignment parameter setting window



	Module 1	Module 2	Module 3
Network type	None	None	None
Starting I/O No.			
Network No.			
Total stations			
Group No.			
Station No.			
Mode			

Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set)

Start I/O No. :

Please input the starting I/O No. of the module in HEX(16 bit) form

Valid module during other station access 1

Fig. 5-18: MELSECNET II combined (master station) network range

△

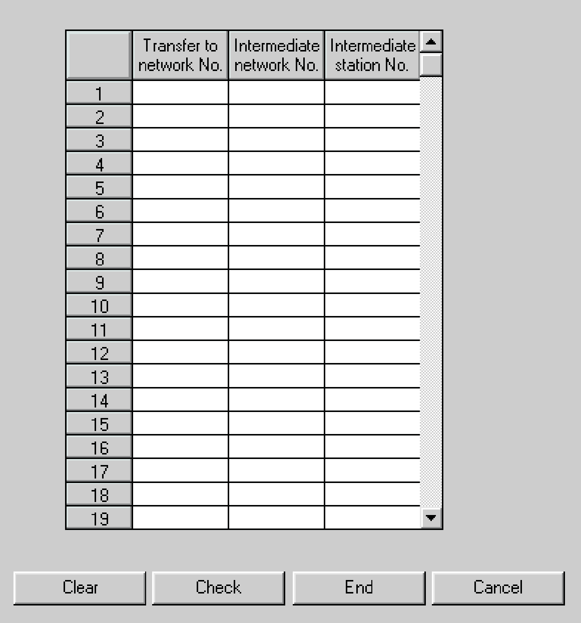
Check button
Use this button to check the set items or values to see if they are correct. This button is available only for the currently open tab.

End setup
The **End** button defines the set items or values and terminates the setting.

Cancel button
Cancels the set items and terminates editing.

Acknowledge X/Y Assignment button
This button is available for the models A2ACPU(S1), A3ACPU, A2AS(S1), A2AS-S30, A2AS-S60, A2UCPU(S1), A2USCPU(S1), A2USHCPU(S1), A3UCPU, and QnACPU, QCPU(Q mode). It is used to confirm the X/Y assignment numbers of the data set in the I/O assignment setting.

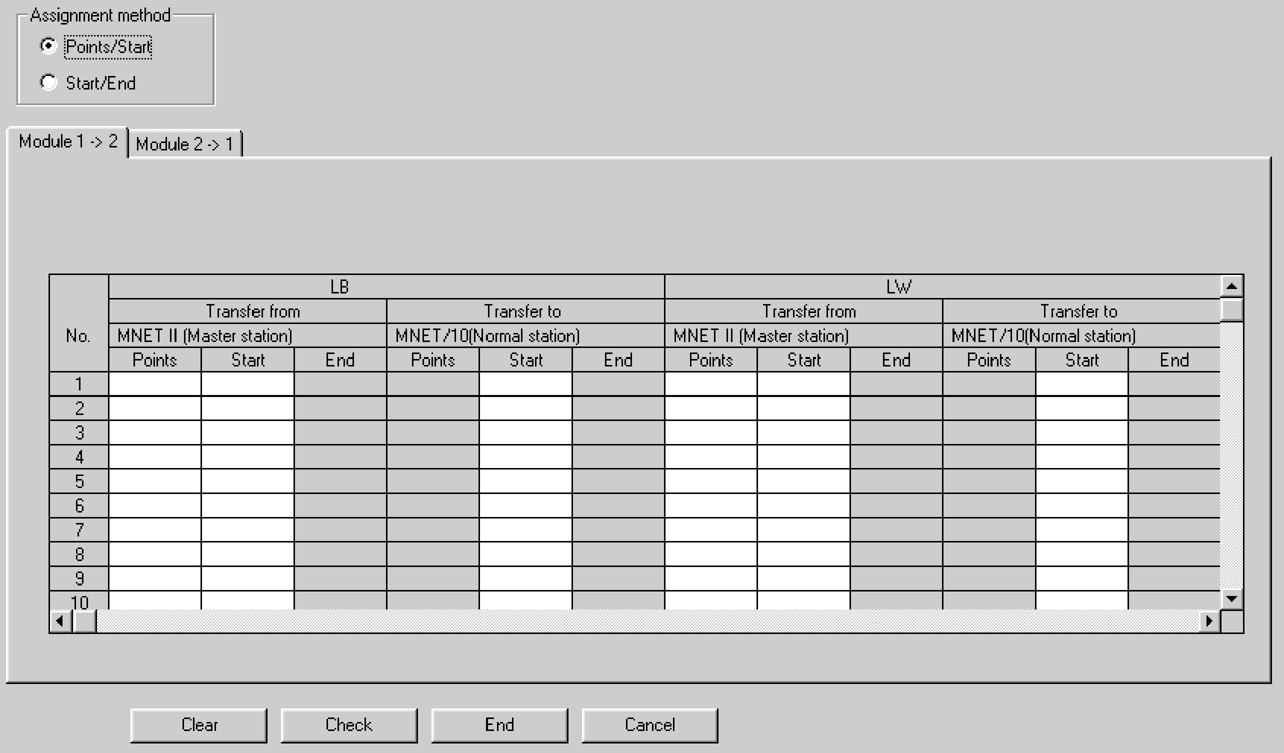
Routing Parameters button
Set the routing parameters in the following dialogbox, that is opened by clicking the button.



	Transfer to network No.	Intermediate network No.	Intermediate station No.
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

Fig. 5-19: *Routing parameters*

Interlink Transmission Parameters button
Set the interlink transmission parameters in the following dialogbox, that is opened by clicking the button.

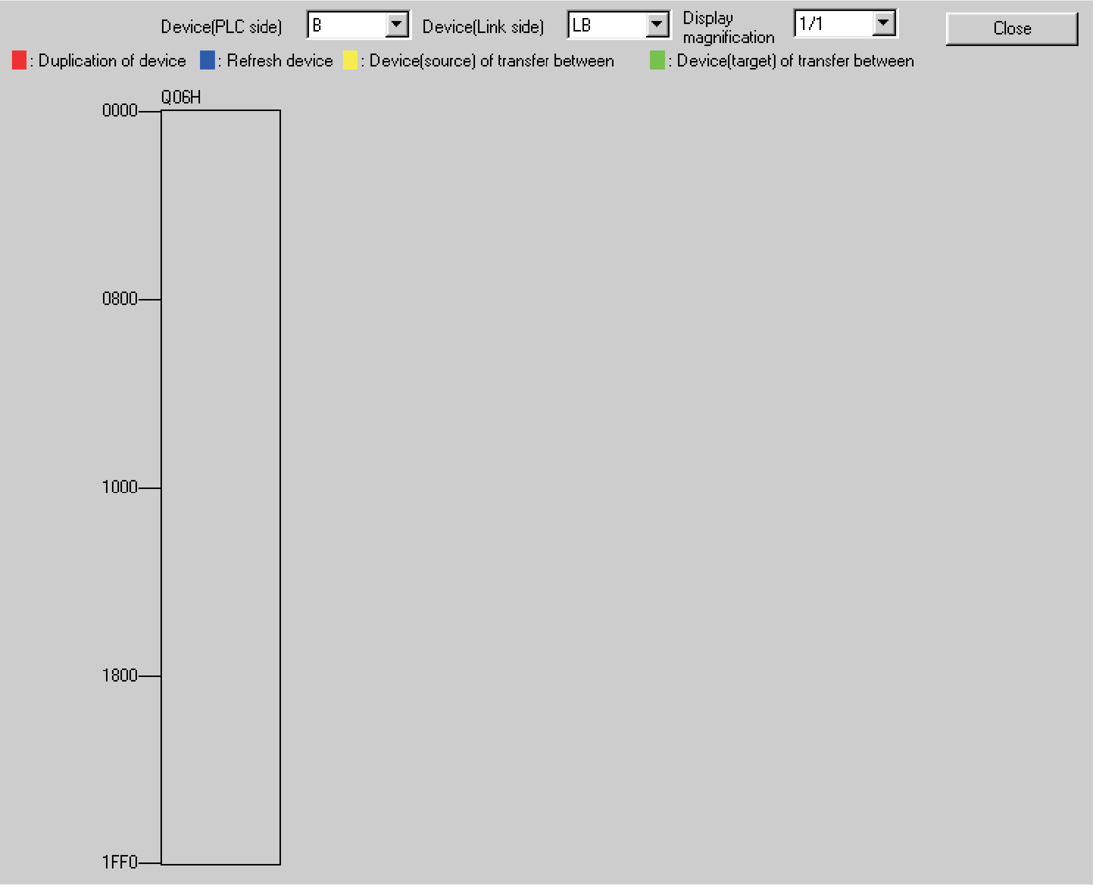


Assignment method
● Points/Start
○ Start/End

	LB						LW					
	Transfer from			Transfer to			Transfer from			Transfer to		
No.	MNET II (Master station)			MNET/10(Normal station)			MNET II (Master station)			MNET/10(Normal station)		
	Points	Start	End	Points	Start	End	Points	Start	End	Points	Start	End
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												

Fig. 5-20: *Interlink Transmission Parameters*

Assignment image button
The following dialogue is opened.



Device(PLC side) B
Device(Link side) LB
Display magnification 1/1

🟥 : Duplication of device 🟦 : Refresh device 🟨 : Device(source) of transfer between 🟩 : Device(target) of transfer between

Fig. 5-21: *Assignment image*

5.3.3 Network Parameter Item Lists

Network parameter items of the Q series

For the remote I/O project, only Ethernet or CC-Link may be set.

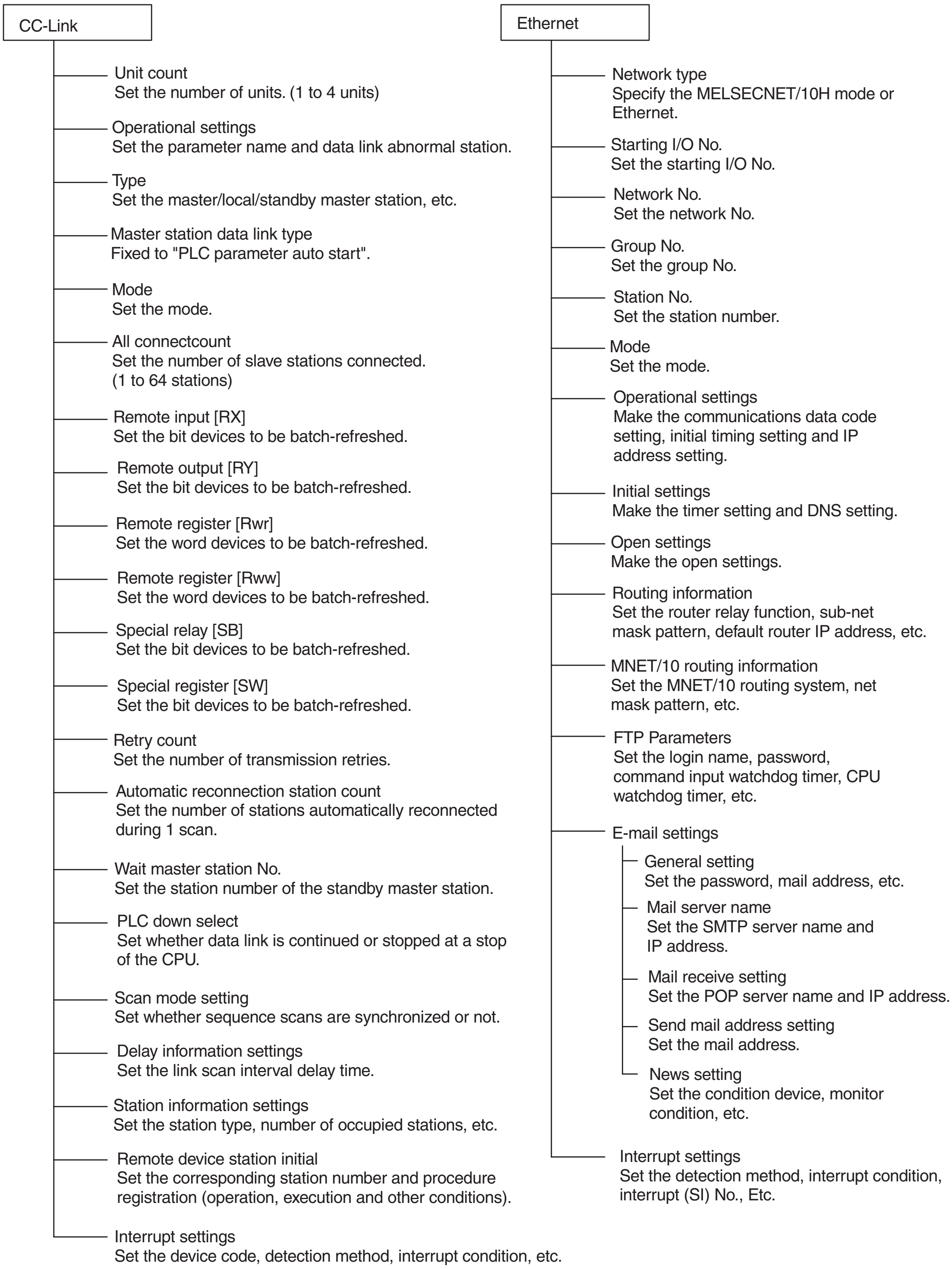
MELSECNET/10
MELSECNET/H

- **Network type**
  Specify the MELSECNET/H·MELSECNET/10 mode,MELSECNET/10 or Ethernet.
- **Starting I/O No.**
  Set the first I/O No.
- **Network No.**
  Set the network No.
- **Total stations**
  Set the total number of (slave) stations.
- **Group No.**
  Set the group No.
- **Mode**
  Set the mode.
- **Network range type (Common parameters)**
  - **I/O Master station specification**
    Set the I/O master station on the LX/LY setting screen.
  - **Reserved station specification**
    Specify the reserved station.
  - **Equal assignment**
    Set when assigning the link device points of the specified stations equally.
  - **Identical point assignment**
    Set the same points according to the preset total number of stations.
  - **Supplementary stings**
    Make the transient, low-etspeed cyclic and other settings.
  - **Station inherent parameters**
    Set the station inherent parameters.
- **Refresh parameters**
  Set the refresh parameters.
- **Interrupt settings**
  Set the device code, detection method, interrupt condition, interrupt (SI) No., Etc.

MELSECNET/H
Remote I/O

- **Network type**
  Specify the MELSECNET/H (remote master).
- **Starting I/O No.**
  Set the starting I/O No.
- **Network No.**
  Set the network No.
- **Total stations**
  Set the total number of (slave) stations.
- **Mode**
  Set the mode.
- **Network range type (Common parameters)**
  - **Reserved station specification**
    Specify the reserved station.
  - **Equal assignment**
    Set when assigning the link device points of the specified stations equally.
  - **Supplementary settings**
    Make the constant scan and the max. Number of return stations in one scan.
- **Refresh parameters**
  Set the refresh parameters.
- **Interrupt settings**
  Set the device code, detection method, interrupt condition, interrupt (SI) No., Etc.

Fig. 5-22: *Network parameter items of the Q series (1)*



CC-Link

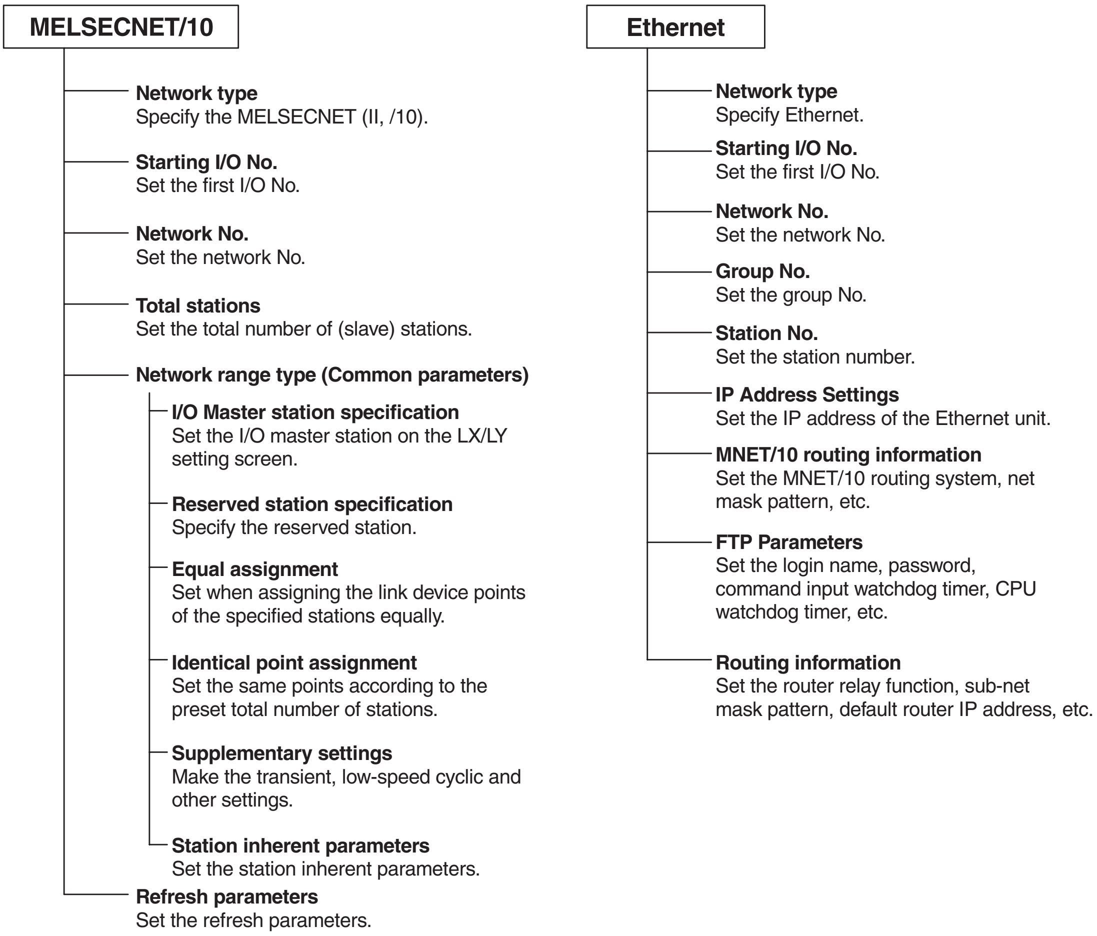
- Unit count
  Set the number of units. (1 to 4 units)
- Operational settings
  Set the parameter name and data link abnormal station.
- Type
  Set the master/local/standby master station, etc.
- Master station data link type
  Fixed to "PLC parameter auto start".
- Mode
  Set the mode.
- All connectcount
  Set the number of slave stations connected.
  (1 to 64 stations)
- Remote input [RX]
  Set the bit devices to be batch-refreshed.
- Remote output [RY]
  Set the bit devices to be batch-refreshed.
- Remote register [Rwr]
  Set the word devices to be batch-refreshed.
- Remote register [Rww]
  Set the word devices to be batch-refreshed.
- Special relay [SB]
  Set the bit devices to be batch-refreshed.
- Special register [SW]
  Set the bit devices to be batch-refreshed.
- Retry count
  Set the number of transmission retries.
- Automatic reconnection station count
  Set the number of stations automatically reconnected during 1 scan.
- Wait master station No.
  Set the station number of the standby master station.
- PLC down select
  Set whether data link is continued or stopped at a stop of the CPU.
- Scan mode setting
  Set whether sequence scans are synchronized or not.
- Delay information settings
  Set the link scan interval delay time.
- Station information settings
  Set the station type, number of occupied stations, etc.
- Remote device station initial
  Set the corresponding station number and procedure registration (operation, execution and other conditions).
- Interrupt settings
  Set the device code, detection method, interrupt condition, etc.

Ethernet

- Network type
  Specify the MELSECNET/10H mode or Ethernet.
- Starting I/O No.
  Set the starting I/O No.
- Network No.
  Set the network No.
- Group No.
  Set the group No.
- Station No.
  Set the station number.
- Mode
  Set the mode.
- Operational settings
  Make the communications data code setting, initial timing setting and IP address setting.
- Initial settings
  Make the timer setting and DNS setting.
- Open settings
  Make the open settings.
- Routing information
  Set the router relay function, sub-net mask pattern, default router IP address, etc.
- MNET/10 routing information
  Set the MNET/10 routing system, net mask pattern, etc.
- FTP Parameters
  Set the login name, password, command input watchdog timer, CPU watchdog timer, etc.
- E-mail settings
  - General setting
    Set the password, mail address, etc.
  - Mail server name
    Set the SMTP server name and IP address.
  - Mail receive setting
    Set the POP server name and IP address.
  - Send mail address setting
    Set the mail address.
  - News setting
    Set the condition device, monitor condition, etc.
- Interrupt settings
  Set the detection method, interrupt condition, interrupt (SI) No., Etc.

Fig. 5-22: *Network parameter items of the Q series (2)*

Network parameter items of the QnA series



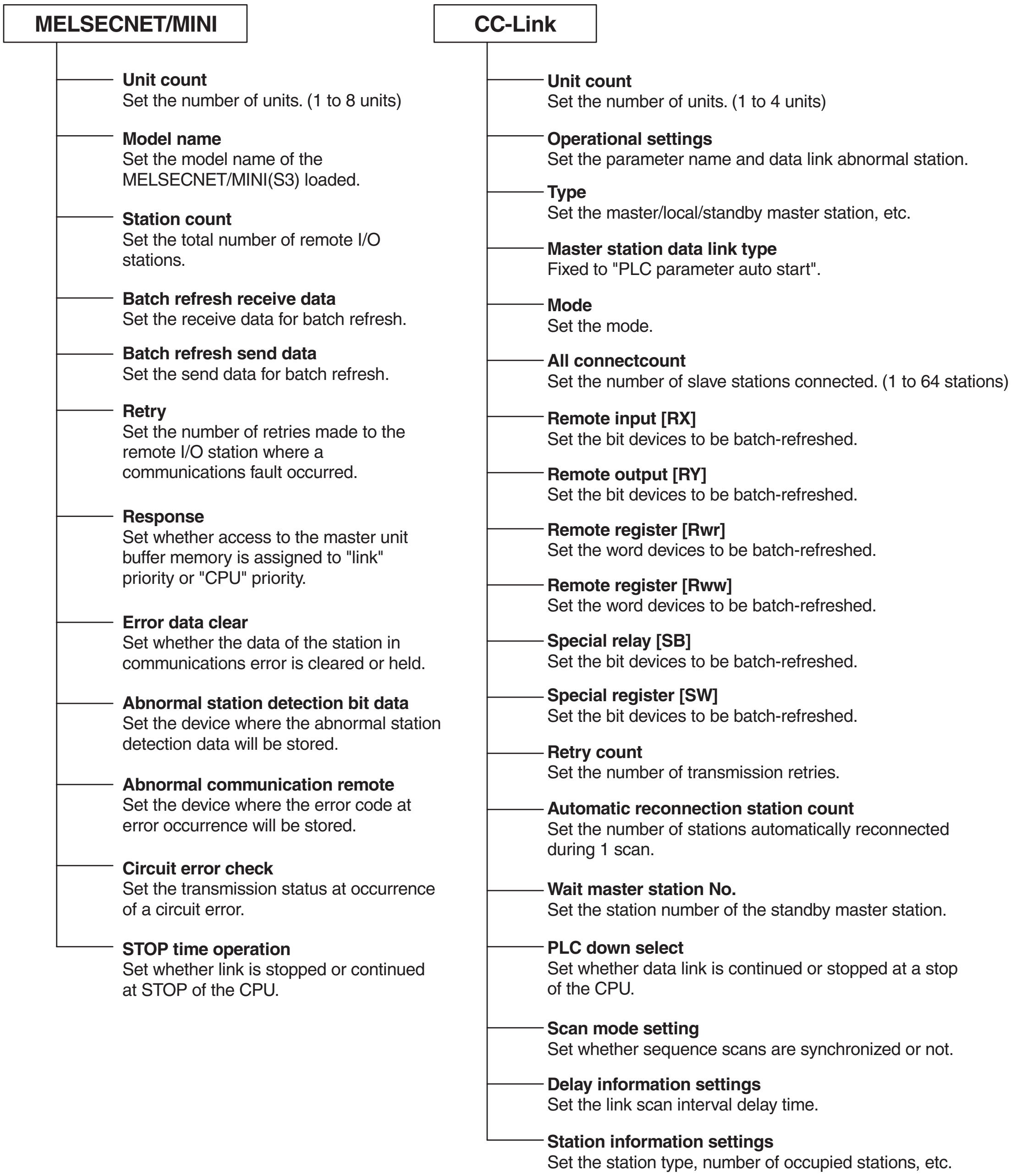
MELSECNET/10

- **Network type**
  Specify the MELSECNET (II, /10).
- **Starting I/O No.**
  Set the first I/O No.
- **Network No.**
  Set the network No.
- **Total stations**
  Set the total number of (slave) stations.
- **Network range type (Common parameters)**
  - **I/O Master station specification**
    Set the I/O master station on the LX/LY setting screen.
  - **Reserved station specification**
    Specify the reserved station.
  - **Equal assignment**
    Set when assigning the link device points of the specified stations equally.
  - **Identical point assignment**
    Set the same points according to the preset total number of stations.
  - **Supplementary settings**
    Make the transient, low-speed cyclic and other settings.
  - **Station inherent parameters**
    Set the station inherent parameters.
- **Refresh parameters**
  Set the refresh parameters.

Ethernet

- **Network type**
  Specify Ethernet.
- **Starting I/O No.**
  Set the first I/O No.
- **Network No.**
  Set the network No.
- **Group No.**
  Set the group No.
- **Station No.**
  Set the station number.
- **IP Address Settings**
  Set the IP address of the Ethernet unit.
- **MNET/10 routing information**
  Set the MNET/10 routing system, net mask pattern, etc.
- **FTP Parameters**
  Set the login name, password, command input watchdog timer, CPU watchdog timer, etc.
- **Routing information**
  Set the router relay function, sub-net mask pattern, default router IP address, etc.

Fig. 5-23: *Network parameter items of the QnA series (1)*



MELSECNET/MINI

- **Unit count**
  Set the number of units. (1 to 8 units)
- **Model name**
  Set the model name of the MELSECNET/MINI(S3) loaded.
- **Station count**
  Set the total number of remote I/O stations.
- **Batch refresh receive data**
  Set the receive data for batch refresh.
- **Batch refresh send data**
  Set the send data for batch refresh.
- **Retry**
  Set the number of retries made to the remote I/O station where a communications fault occurred.
- **Response**
  Set whether access to the master unit buffer memory is assigned to "link" priority or "CPU" priority.
- **Error data clear**
  Set whether the data of the station in communications error is cleared or held.
- **Abnormal station detection bit data**
  Set the device where the abnormal station detection data will be stored.
- **Abnormal communication remote**
  Set the device where the error code at error occurrence will be stored.
- **Circuit error check**
  Set the transmission status at occurrence of a circuit error.
- **STOP time operation**
  Set whether link is stopped or continued at STOP of the CPU.

CC-Link

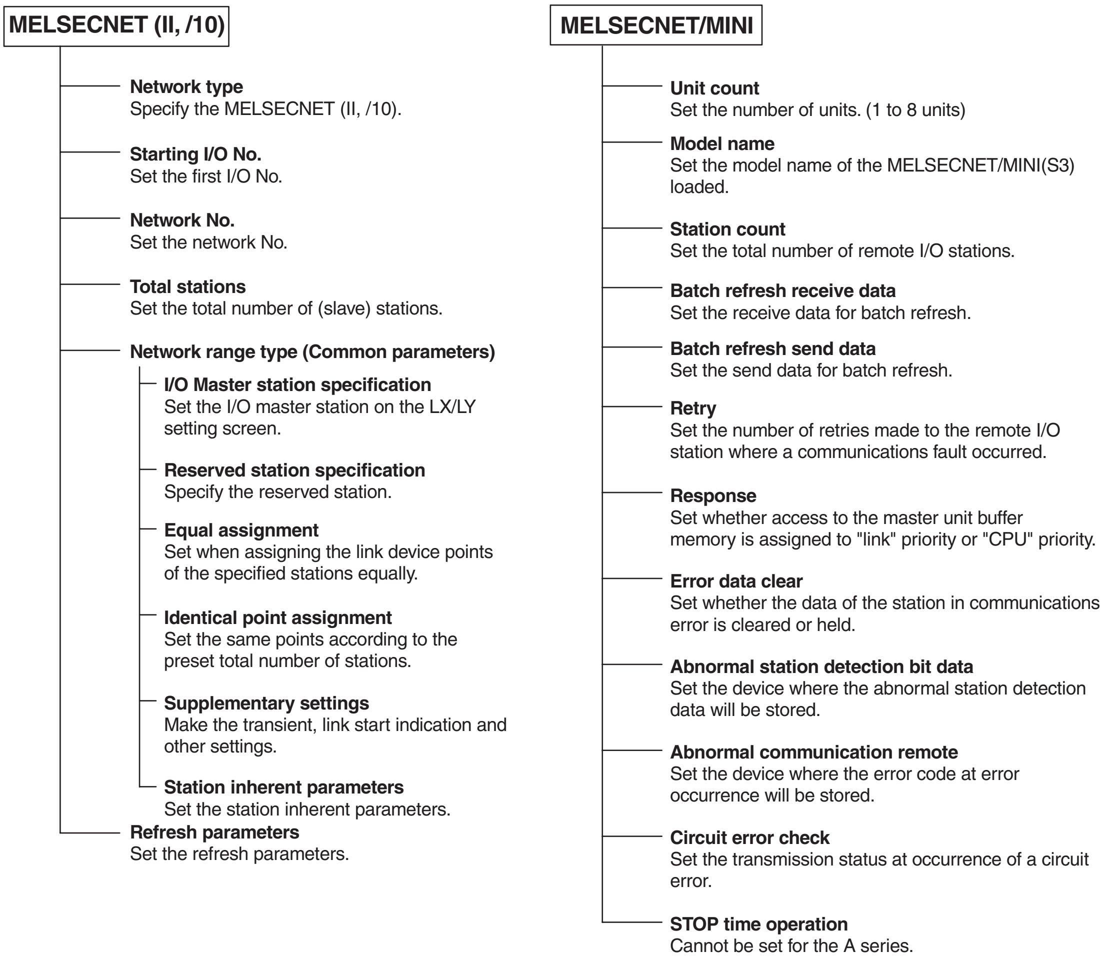
- **Unit count**
  Set the number of units. (1 to 4 units)
- **Operational settings**
  Set the parameter name and data link abnormal station.
- **Type**
  Set the master/local/standby master station, etc.
- **Master station data link type**
  Fixed to "PLC parameter auto start".
- **Mode**
  Set the mode.
- **All connectcount**
  Set the number of slave stations connected. (1 to 64 stations)
- **Remote input [RX]**
  Set the bit devices to be batch-refreshed.
- **Remote output [RY]**
  Set the bit devices to be batch-refreshed.
- **Remote register [Rwr]**
  Set the word devices to be batch-refreshed.
- **Remote register [Rww]**
  Set the word devices to be batch-refreshed.
- **Special relay [SB]**
  Set the bit devices to be batch-refreshed.
- **Special register [SW]**
  Set the bit devices to be batch-refreshed.
- **Retry count**
  Set the number of transmission retries.
- **Automatic reconnection station count**
  Set the number of stations automatically reconnected during 1 scan.
- **Wait master station No.**
  Set the station number of the standby master station.
- **PLC down select**
  Set whether data link is continued or stopped at a stop of the CPU.
- **Scan mode setting**
  Set whether sequence scans are synchronized or not.
- **Delay information settings**
  Set the link scan interval delay time.
- **Station information settings**
  Set the station type, number of occupied stations, etc.

Fig. 5-23: *Network parameter items of the QnA series (2)*

NOTE

CC-Link parameters cannot be set to the Q4ARCPU.

Network parameter items of the A series



MELSECNET (II, /10)

- **Network type**
  Specify the MELSECNET (II, /10).
- **Starting I/O No.**
  Set the first I/O No.
- **Network No.**
  Set the network No.
- **Total stations**
  Set the total number of (slave) stations.
- **Network range type (Common parameters)**
  - **I/O Master station specification**
    Set the I/O master station on the LX/LY setting screen.
  - **Reserved station specification**
    Specify the reserved station.
  - **Equal assignment**
    Set when assigning the link device points of the specified stations equally.
  - **Identical point assignment**
    Set the same points according to the preset total number of stations.
  - **Supplementary settings**
    Make the transient, link start indication and other settings.
  - **Station inherent parameters**
    Set the station inherent parameters.
- **Refresh parameters**
  Set the refresh parameters.

MELSECNET/MINI

- **Unit count**
  Set the number of units. (1 to 8 units)
- **Model name**
  Set the model name of the MELSECNET/MINI(S3) loaded.
- **Station count**
  Set the total number of remote I/O stations.
- **Batch refresh receive data**
  Set the receive data for batch refresh.
- **Batch refresh send data**
  Set the send data for batch refresh.
- **Retry**
  Set the number of retries made to the remote I/O station where a communications fault occurred.
- **Response**
  Set whether access to the master unit buffer memory is assigned to "link" priority or "CPU" priority.
- **Error data clear**
  Set whether the data of the station in communications error is cleared or held.
- **Abnormal station detection bit data**
  Set the device where the abnormal station detection data will be stored.
- **Abnormal communication remote**
  Set the device where the error code at error occurrence will be stored.
- **Circuit error check**
  Set the transmission status at occurrence of a circuit error.
- **STOP time operation**
  Cannot be set for the A series.

Fig. 5-24: *Network parameter items of the A series*

5.3.4 Explanations for Network Parameter Setting Screen

The following items are related to network parameter setting.

- Unit valid for other station access

  This setting is made valid when there are multiple units which are used without specifying the network No. or when the network No. has not been set for access to the other station.

- **Read PLC data** button

  Reads the implementation status.

  – For QnA series
  Always delete the parameter file of the PLC before reading the implementation status. The parameter file is read if the parameter file exists in the PLC.

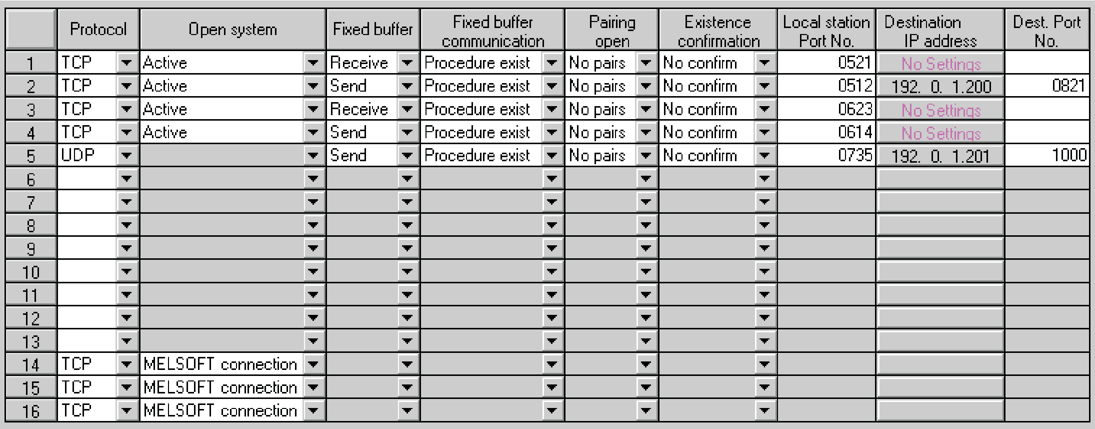
- Remote I/O project

  You can set up to four Ethernet modules and CC-Link modules, respectively.

- MELSOFT connection on Ethernet connection (Q Series only)

  This setting is used when connecting multiple MELSOFT products such as GX IEC Developer (a maximum of 17 units) using the TCP/IP communication system.

  – Setting procedure:
  Ethernet → Open settings → Protocol (TCP) → Open system (MELSOFT connection)
  The screen below gives a setting example when user connections 14 to 16 are used to connect MELSOFT products, while connections 1 to 5 are used to communicate with non-MELSOFT products via a MC protocol communication, fixed-buffer communication, etc.



Used for various communications

Used to connect MELSOFT products

	Protocol	Open system	Fixed buffer	Fixed buffer communication	Pairing open	Existence confirmation	Local station Port No.	Destination IP address	Dest. Port No.
1	TCP	Active	Receive	Procedure exist	No pairs	No confirm	0521	No Settings	
2	TCP	Active	Send	Procedure exist	No pairs	No confirm	0512	192. 0. 1.200	0821
3	TCP	Active	Receive	Procedure exist	No pairs	No confirm	0623	No Settings	
4	TCP	Active	Send	Procedure exist	No pairs	No confirm	0614	No Settings	
5	UDP		Send	Procedure exist	No pairs	No confirm	0735	192. 0. 1.201	1000
6									
7									
8									
9									
10									
11									
12									
13									
14	TCP	MELSOFT connection							
15	TCP	MELSOFT connection							
16	TCP	MELSOFT connection							

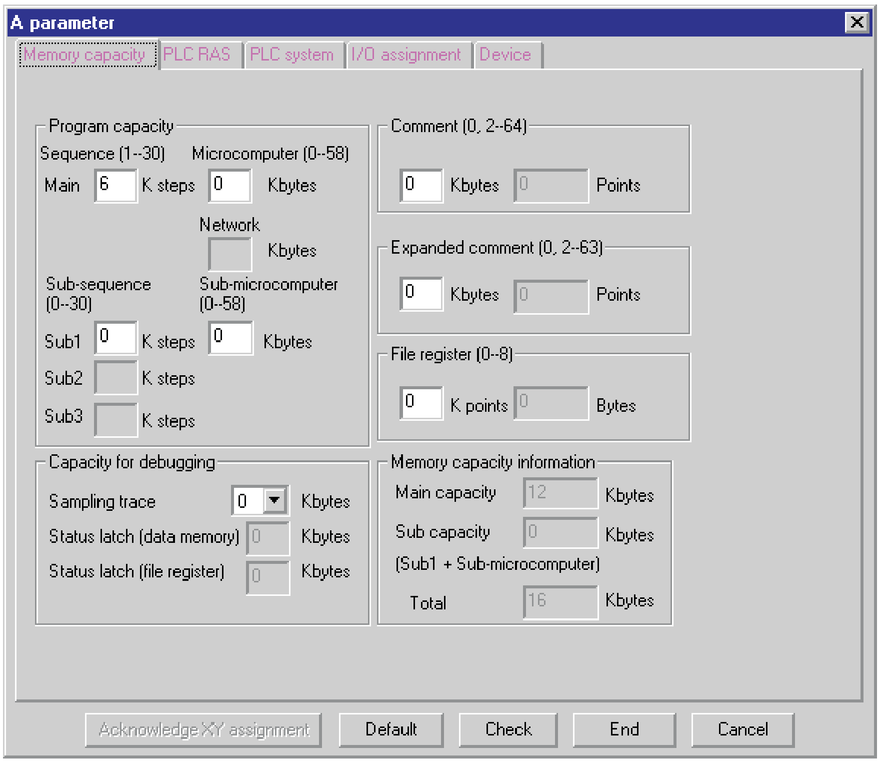
Fig. 5-25: Device Settings

NOTE For the Q series, Read PLC data is not available for the network parameters.

5.4 Memory Parameters

5.4.1 Memory Parameters (MELSEC A)

This dialogue box contains all the settings for the assignment and allocation of memory. In addition, you can also set further parameters.



Program capacity
Sequence (1--30) Microcomputer (0--58)
Main 6 K steps 0 Kbytes
Network Kbytes
Sub-sequence (0--30) Sub-microcomputer (0--58)
Sub1 0 K steps 0 Kbytes
Sub2 K steps
Sub3 K steps

Comment (0, 2--64)
0 Kbytes 0 Points

Expanded comment (0, 2--63)
0 Kbytes 0 Points

File register (0--8)
0 K points 0 Bytes

Capacity for debugging
Sampling trace 0 Kbytes
Status latch (data memory) 0 Kbytes
Status latch (file register) 0 Kbytes

Memory capacity information
Main capacity 12 Kbytes
Sub capacity 0 Kbytes
(Sub1 + Sub-microcomputer)
Total 16 Kbytes

Fig. 5-26: Memory parameters of the A series

Program capacity: Allocation of MAIN program sequence memory
1 step = 2 bytes
The program checks the maximum value automatically. After compilation the memory actually required is shown in **Memory capacity information**.

Sub sequence: Allocation of SUB program sequence memory
1 step = 2 bytes
The program checks the maximum value automatically. After compilation the memory actually required is shown in parentheses after this entry.

Microcomputer: Allocation of memory for microcomputer programs
This option is only relevant for Sequential Function Chart programs for the PLC CPUs of the AnN series. The memory for these programs must be allocated in steps of 2 kByte.

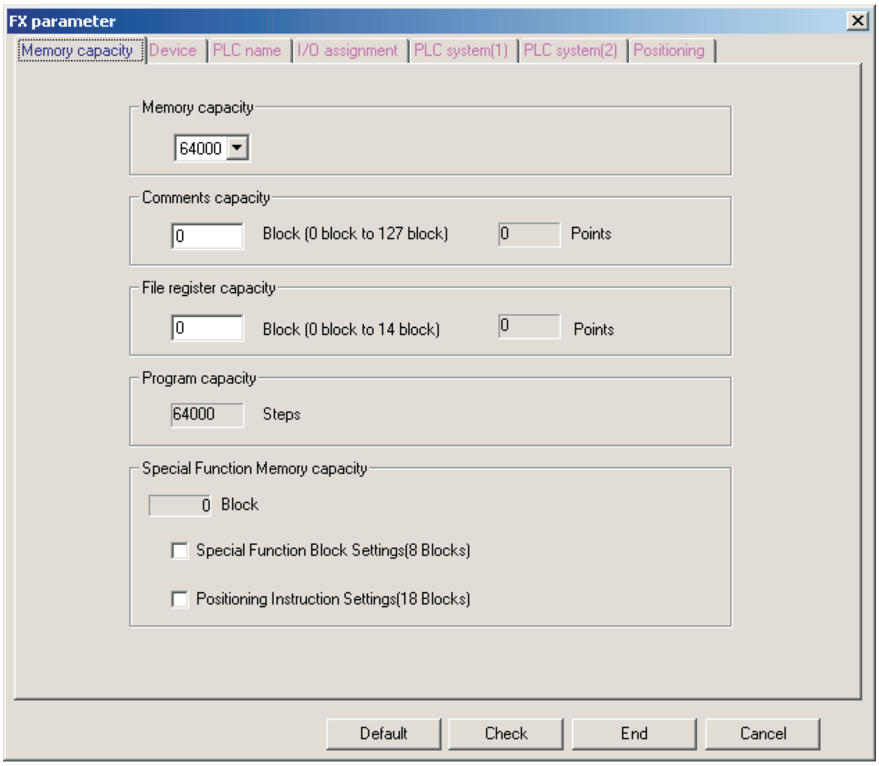
File register: Allocation of memory for the file registers
The available memory depends on the type of memory cassette installed.

Comment/Extended comment: Allocation of memory for the comments

Capacity for debugging: Allocation of memory for debugging

5.4.2 Memory Parameters (MELSEC FX)

This dialogue box contains all the settings for the assignment and allocation of memory. In addition, you can also set further parameters.



Memory capacity: 64000

Comments capacity: 0 Block (0 block to 127 block) 0 Points

File register capacity: 0 Block (0 block to 14 block) 0 Points

Program capacity: 64000 Steps

Special Function Memory capacity: 0 Block

□ Special Function Block Settings(8 Blocks)

□ Positioning Instruction Settings(18 Blocks)

Fig. 5-27: *Memory parameters of the FX3U*

Memory capacity: Allocation of program memory. Set the memory capacity of the PLC.

Comments capacity: Allocation of memory for the comments.

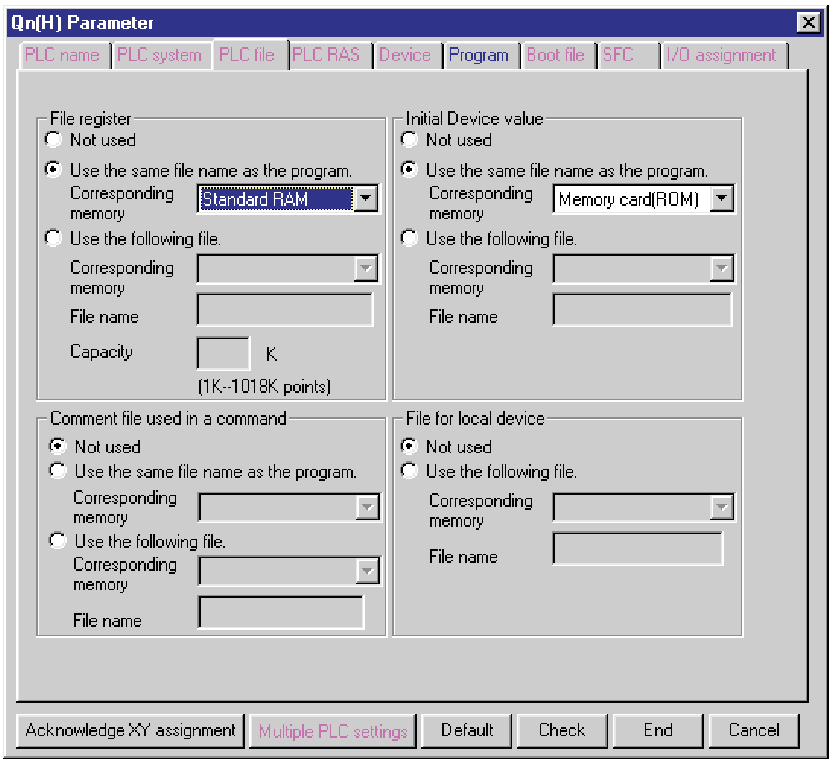
File register capacity: Allocation of memory for the file registers.

Program capacity: Allocation of sequence program memory.

Special Function Memory capacity (FX3U only): Select the special extension module/block initial value setting or positioning setting.

5.4.3 Memory Parameters (MELSEC Q/QnA)

In addition to showing the program size parameters for the selected CPU this dialogue box is also used to define the number of file registers you wish to use.



File register
○ Not used
● Use the same file name as the program.
Corresponding memory: Standard RAM
○ Use the following file.
Corresponding memory:
File name:
Capacity: K
(1K--1018K points)

Initial Device value
○ Not used
● Use the same file name as the program.
Corresponding memory: Memory card(ROM)
○ Use the following file.
Corresponding memory:
File name:

Comment file used in a command
● Not used
○ Use the same file name as the program.
Corresponding memory:
○ Use the following file.
Corresponding memory:
File name:

File for local device
● Not used
○ Use the following file.
Corresponding memory:
File name:

Fig. 5-28: *Specifying program size and file registers*

File Register

If you wish to use file registers in Q/QnA series CPUs you have to define the number you wish to use here first. The Q00J CPU does not support file registers.

Capacity: Specify the number of file registers you wish to use (units: k points). The Q/QnA CPU can handle a maximum of 1024 addresses (i.e. 1MB). However, since a few addresses are also used for making the file registers available you can only actually use a maximum of 1018 addresses. The Q00 and Q01 CPUs are fixed to 32 kPoints.

Corresponding memory: Specify the number of the drive corresponding to the external memory card you wish to use for the file registers (not available for Q00(J), Q01 CPUs). Valid entries in the pull-down list:

IC Card A(RAM): RAM in the card in the first slot
IC Card A(ROM): ROM in the card in the first slot (QnA only)
IC Card B(RAM): RAM in the card in the second slot
IC Card B(ROM): ROM in the card in the second slot (QnA only)

NOTE

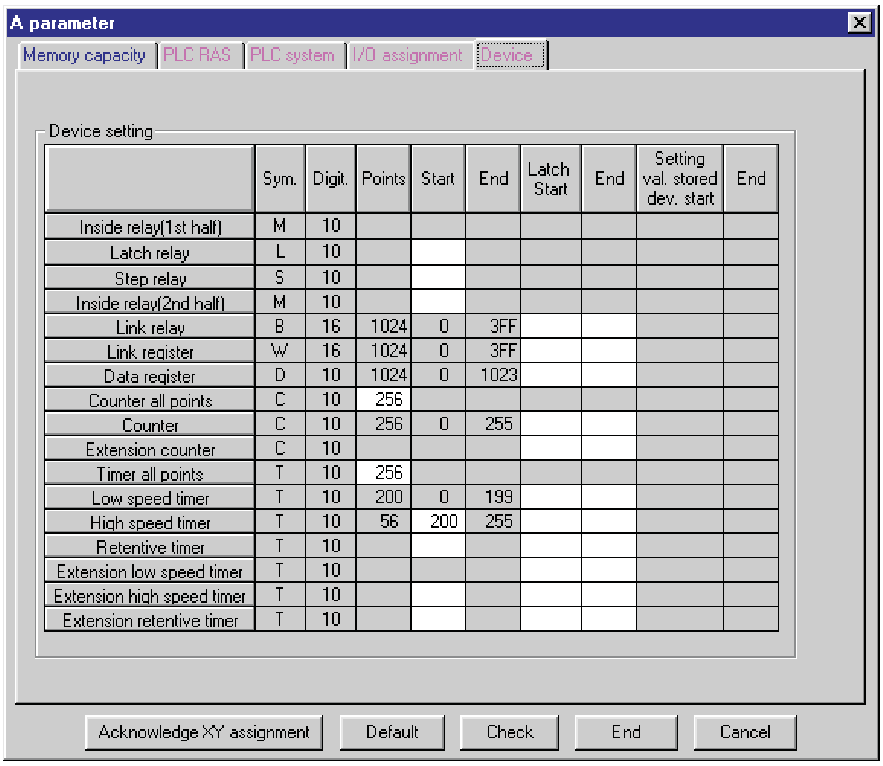
File registers can not be created in the flash card ROM of a Q CPU under GX IEC Developer.

5.5 Latch Ranges

The PLC Parameter settings include a check mechanism. If a setting in the System Variables overlaps with the corresponding setting in the Latch Range, an overlap warning will be displayed. You can ignore this warning and continue or modify the settings.

5.5.1 Latch Ranges (MELSEC A)

This dialogue box includes all the settings available for the division and allocation of the retentive, battery-powered memory ranges. The data in these ranges is then stored in the event of a power failure or a system reset. Please note that the settings you make here have nothing to do with the Status Latch settings in the **Debug** menu!



	Sym.	Digit.	Points	Start	End	Latch Start	End	Setting val. stored dev. start	End
Inside relay(1st half)	M	10							
Latch relay	L	10							
Step relay	S	10							
Inside relay(2nd half)	M	10							
Link relay	B	16	1024	0	3FF				
Link register	W	16	1024	0	3FF				
Data register	D	10	1024	0	1023				
Counter all points	C	10	256						
Counter	C	10	256	0	255				
Extension counter	C	10							
Timer all points	T	10	256						
Low speed timer	T	10	200	0	199				
High speed timer	T	10	56	200	255				
Retentive timer	T	10							
Extension low speed timer	T	10							
Extension high speed timer	T	10							
Extension retentive timer	T	10							

Fig. 5-29: *The values for parameters followed by an H in parentheses (H) must be entered in hexadecimal format.*

Latch Relays / Link Relays: Device range for the memory-resident latch and link relays. The parameter values must be entered in hexadecimal format.

Timers: Device range for the 10 ms and 100 ms timers

Counters: Counter device range

Data/Link Registers: Register device range

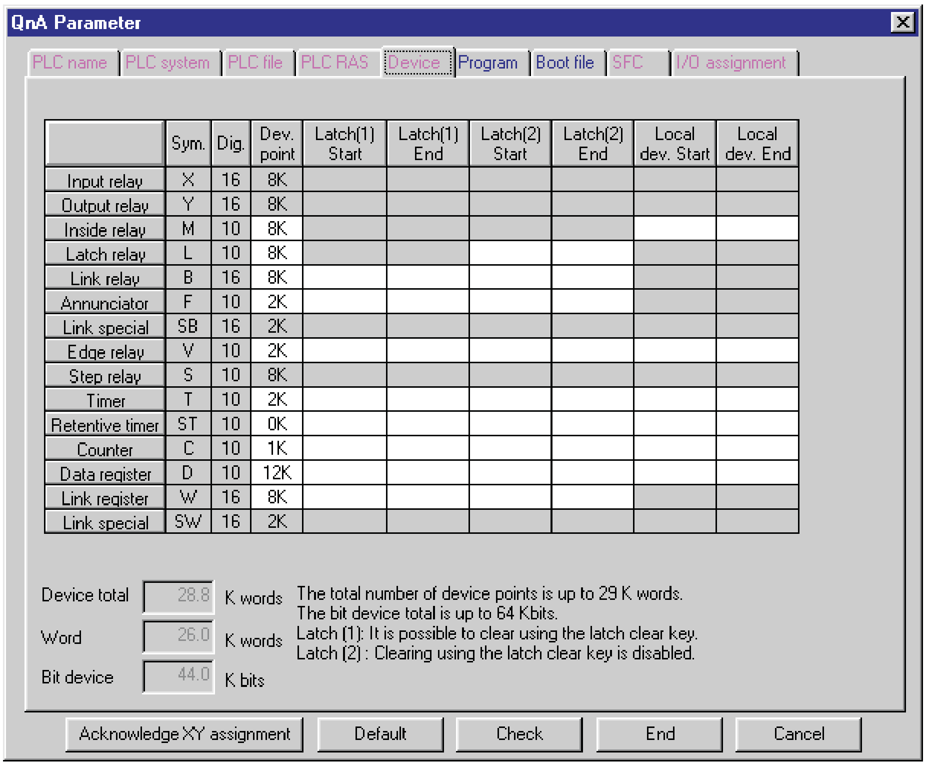
Extended Timers: Device range for extended timers

Extended Counters: Device range for extended counters

5.6 Device Settings

5.6.1 Device Settings (MELSEC Q/QnA)

This dialogue box is used to define the devices to be used in the project. You can also specify whether the latch operation is to be performed or not for the non-volatile device ranges with battery backup.



	Sym.	Dig.	Dev. point	Latch(1) Start	Latch(1) End	Latch(2) Start	Latch(2) End	Local dev. Start	Local dev. End
Input relay	X	16	8K						
Output relay	Y	16	8K						
Inside relay	M	10	8K						
Latch relay	L	10	8K						
Link relay	B	16	8K						
Annunciator	F	10	2K						
Link special	SB	16	2K						
Edge relay	V	10	2K						
Step relay	S	10	8K						
Timer	T	10	2K						
Retentive timer	ST	10	0K						
Counter	C	10	1K						
Data register	D	10	12K						
Link register	W	16	8K						
Link special	SW	16	2K						

Device total 28.8 K words
Word 26.0 K words
Bit device 44.0 K bits

The total number of device points is up to 29 K words.
The bit device total is up to 64 Kbits.
Latch (1): It is possible to clear using the latch clear key.
Latch (2) : Clearing using the latch clear key is disabled.

Fig. 5-30: *Device Settings*

The following table shows the correspondences between the devices in the A series and the Q/QnA series:

MELSEC A Devices	MELSEC Q/QnA Devices
Accumulators A0 and A1	Diagnostics link registers SD718 and SD719
Link registers B	
Counters C	
Data registers D0 - D8191	
Special registers D9000 - D9255	Diagnostics link registers SD1000 - SD1255
Annunciators F	
Hexadecimal constants H	
Interrupt pointers I	
Constants K	
Latch relays L0 - L8191	
Internal Relays M0 - M8191	
Special relays M9000 - M9255	Diagnostics link relays SM1000 - SM1255
Nesting N	
Pointers P	
File registers R	
Step relays S0 - S8191	
Timers T (low-speed 100ms)	
Timer T (high-speed 10ms)	
Accumulator (retentive) timers T	Accumulator timers ST
Index registers V, V1, V2, ...	Index registers Z7, Z8, Z9 …
Link registers W	
Input relays X	
Output relays Y	
Index registers Z, Z1, Z2, ...	Index registers Z0, Z1, Z2, ...

Tab. 5-4: Correspondences between A series and Q series devices

RAD

Input format (10 = decimal, 16 = hexadecimal)

Devices

Enter the number of devices you wish to use in your project in this column. The device types with no entry fields have a fixed range which cannot be changed by the user:

- Input relays X
  Q00J, Q00, Q01 = 2 K, Others = 8 K
- Output relays Y
  Q00J, Q00, Q01 = 2 K, Others = 8K
- Diagnostics link relays SB
  Q00J, Q00, Q01 = 1 K, Others = 2K
- Step relays S
  Q00J, Q00, Q01 = 2 K, Others = 8K
- Diagnostics link registers SW
  Q00J, Q00, Q01 = 1 K, Others = 2K

Maximum Input Limits
Maximum value for bit devices: 32K address points
Maximum number of all bit devices: 64K address points
Maximum total number of devices: 29K words

Number of bits per word/kWord
1 word = 16 bits
1Kword = 1,024 words = 16,384 bits

Number of bits per address
X, Y, M, L, B, F, SB, V, S : 1 address = 1 bit
D, W, SW: 1 address = 16 bits = 1 word
T, ST, C: 1 address = 18 bits = 1 word + 2 bits

5.6.2 Devices in a Redundant System

A feature called Tracking makes it possible, that in a redundant system devices have the same values on both PLCs. This is vital for consistent continuation after a system switch. The following devices are synchronized (tracked) between the Control and Standby systems automatically at every scan by default.

Device	Default tracking range
Input	X0 to X1FFF
Output	Y0 to Y1FFF
Internal Relay	M0 to M8191
Latch Relay	L0 to L8191
Step Relay	S0 to S8191
Edge Relay	V0 to V2047
Link Relay	B0 to B1FFF
Timer	T0 to T2047
Counter	C0 to C1023
Data Register	D0 to D 12287
Link Register	W0 to W1FFF
Index Register	Z0 to Z15

Tab. 5-5: *List of devices being synchronized in a redundant system*

The tracking settings can also be set manually (for max 100 kW devices) in the Parameters. This may be helpful to

- reduce the number of tracked devices, reducing the delay in the scan time caused by the tracking, or
- track devices that are not tracked by default (e.g. File Register).

Modifying device values on the Standby system does not have the same effect as on a regular system, since any modification can be overwritten with the device value from the Control system by the tracking functionality. Therefore modifying device values on the Standby system is not supported by default.

This affects the following functionalities:

- Monitoring in a POU body
- Monitoring in the EDM
- The **Online** → **Modify Variable Value** functionality
- The EDM’s **Write to PLC** functionality
- The Device Edit’s **Write to PLC** functionality
- Monitoring in the Buffer Memory Batch Monitor

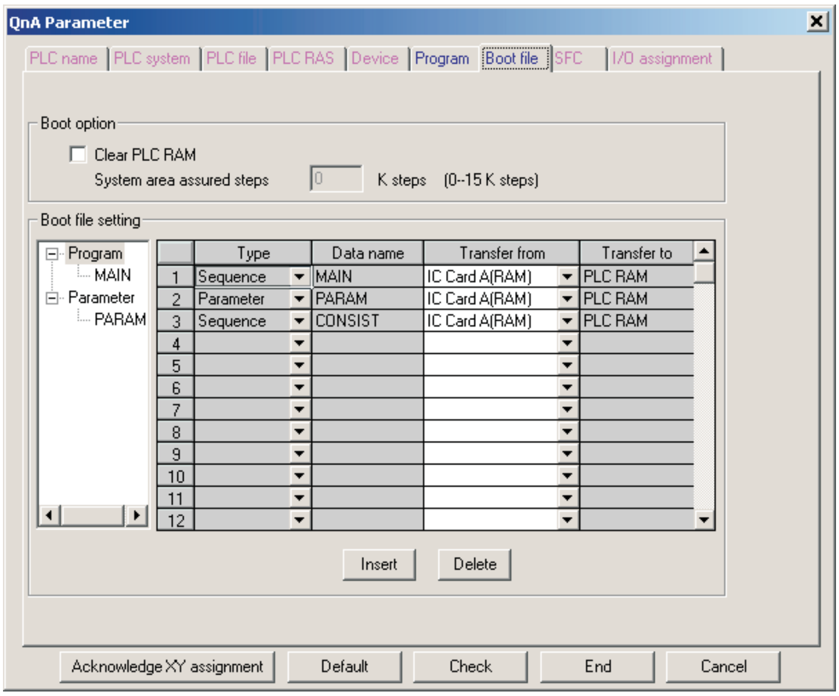
When attempting to execute any of these operations on the Standby system, an error message will be displayed.

NOTE

It is possible to enable modifying devices on the Standby system on the **Extras** → **Options** → **Redundancy option page**. Please see sec. 2.8.3 for details.

5.7 Boot Settings (MELSEC Q/QnA (except Q25SS (SX Controller))

In the Boot settings PLC behaviour on booting are specified. Here a system area can be created to speed up monitoring or for Multiple Block Online Change (MBOC).



Boot option

☐ Clear PLC RAM

System area assured steps 0 K steps (0--15 K steps)

Boot file setting

- Program
  - MAIN
- Parameter
  - PARAM

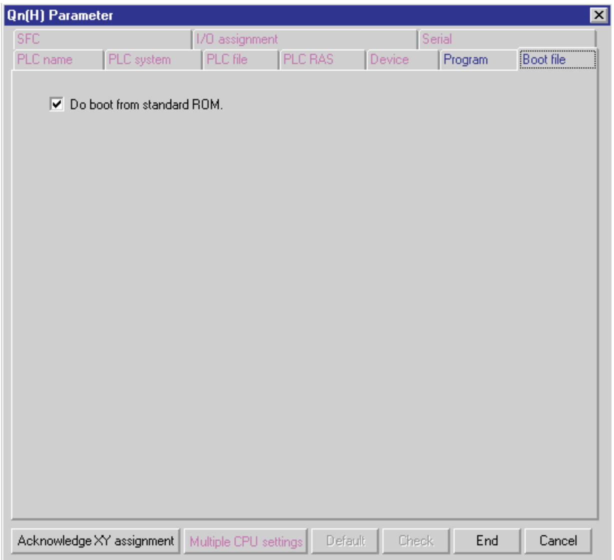
	Type	Data name	Transfer from	Transfer to
1	Sequence	MAIN	IC Card A(RAM)	PLC RAM
2	Parameter	PARAM	IC Card A(RAM)	PLC RAM
3	Sequence	CONSIST	IC Card A(RAM)	PLC RAM
4				
5				
6				
7				
8				
9				
10				
11				
12				

Fig. 5-31: *Boot settings (except Q00(J), Q01 CPU),*

Clear PLC RAM: Clears the program memory before booting, if selected.

System area assumed steps: Select size of system area for high-speed monitoring from other stations. Range: 0 – 15 kSteps. Only available, if **Clear program memory** is selected.

Boot file setting: Set the boot-run file, etc.



☑ Do boot from standard ROM.

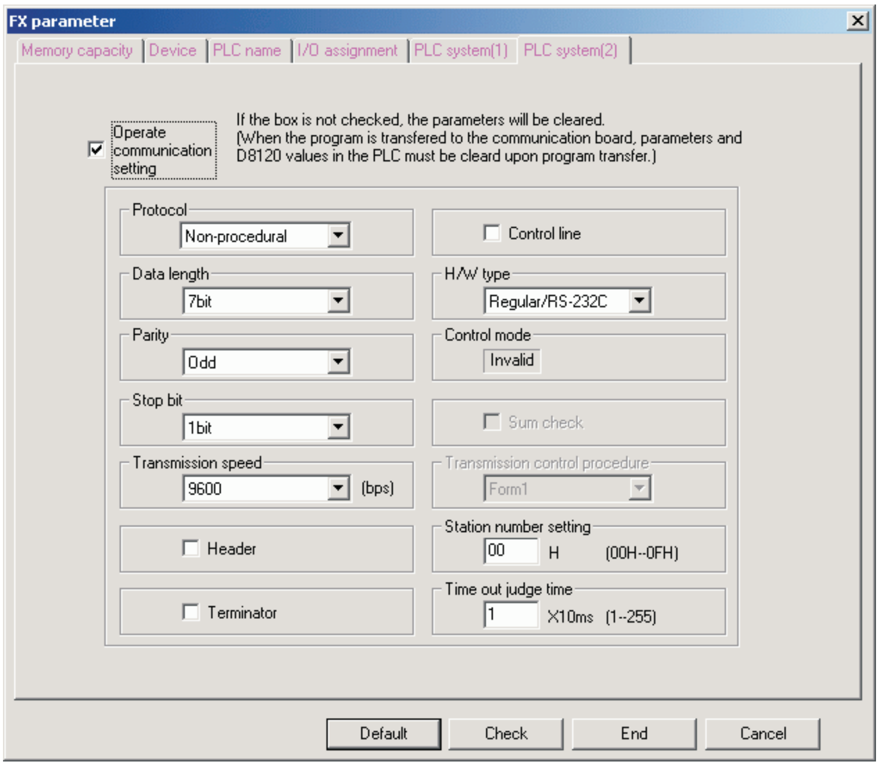
Fig. 5-32:
Boot setting (Q00(J), Q01 CPU)

Do boot from Standard ROM: Do boot from Standard RAM, if selected.

NOTE | See sec. 8.6.1 for Boot Settings concerning the Intelligent Parameter File of the Q series.

5.8 PLC System (FX2N)

In this dialogue box the communication settings between computer link and protocol type is enabled, disabled, and setup. The settings are stored in the project database.



☑ Operate communication setting

If the box is not checked, the parameters will be cleared.
(When the program is transfered to the communication board, parameters and D8120 values in the PLC must be cleard upon program transfer.)

Protocol: Non-procedural

Data length: 7bit

Parity: Odd

Stop bit: 1bit

Transmission speed: 9600 (bps)

□ Header

□ Terminator

□ Control line

H/W type: Regular/RS-232C

Control mode: Invalid

□ Sum check

Transmission control procedure: Form1

Station number setting: 00 H (00H--0FH)

Time out judge time: 1 X10ms (1--255)

Fig. 5-33: PLC system settings

Operate communication setting: Enables or disables the use of communication settings.

Protocol: Select the protocol for communication. Only available, if **Operate communication setting** is selected.

Data length: Select the data length for communication. Only available, if **Operate communication setting** is selected.

Parity: Select the parity for communication. Only available, if **Operate communication setting** is selected.

Stop bit: Select the number of stop bits for communication. Only available, if **Operate communication setting** is selected.

Baud rate: Select the baud rate for communication. Only available, if **Operate communication setting** is selected.

Header: Enables or disables the use of a header in the transmission. Only available, if **Operate communication setting** is selected.

Terminator: Enables or disables the use of a terminator in the transmission. Only available, if **Operate communication setting** is selected.

Control line: Use handshaking recognized by the **Non procedural protocol**. Only available, if **Operate communication setting** and **Non procedural protocol** are selected.

H/W type: Select the hardware type for the transmission. Only available, if **Operate communication setting** is selected.

Control mode: Indication of control mode.

Sum check: Enables or disables the sum check for the transmission using **Dedicated protocol**. Only available, if **Operate communication setting** and **Dedicated protocol** are selected.

Transmission control procedure: Select the format for **Dedicated protocol**. Only available, if **Operate communication setting** and **Dedicated protocol** are selected.

Station number setting: Specify a station number to use within networks. Only available, if **Operate communication setting** is selected.

Time out judge time: Specify a time when to judge a time out of the transmission. Only available, if **Operate communication setting** is selected.

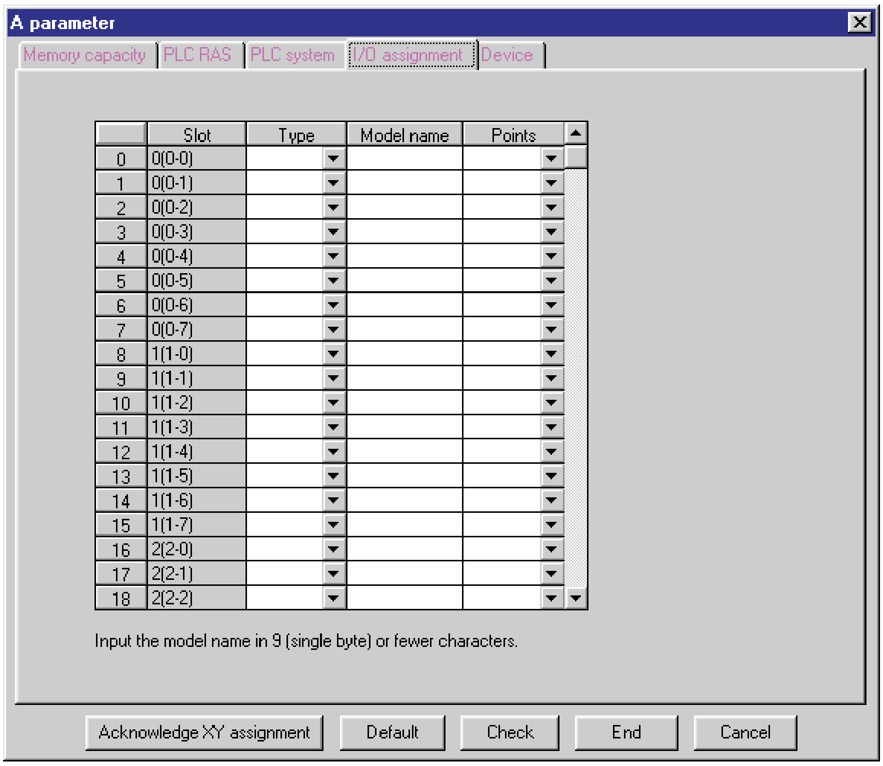
5.9 I/O Configuration

5.9.1 I/O Configuration (MELSEC A)

All the modules, their slot numbers and I/O address number ranges are entered in this dialogue box. A PLC system can consist of up to 8 base units with 8 slots each, thus providing up to a maximum of 64 slot numbers.

Entering the configuration data is particularly important for:

- AnA and AnU systems. With these systems certain commands for specific special function modules can only be executed if the configuration has been explicitly defined.
- Systems for which address ranges different from the current configuration need to be defined to reserve addresses for future expansions.



	Slot	Type	Model name	Points
0	0(0-0)			
1	0(0-1)			
2	0(0-2)			
3	0(0-3)			
4	0(0-4)			
5	0(0-5)			
6	0(0-6)			
7	0(0-7)			
8	1(1-0)			
9	1(1-1)			
10	1(1-2)			
11	1(1-3)			
12	1(1-4)			
13	1(1-5)			
14	1(1-6)			
15	1(1-7)			
16	2(2-0)			
17	2(2-1)			
18	2(2-2)			

Input the model name in 9 (single byte) or fewer characters.

Fig. 5-34: *Input and output configuration*

I/O Slot: Number of the base unit slot in which the module is installed

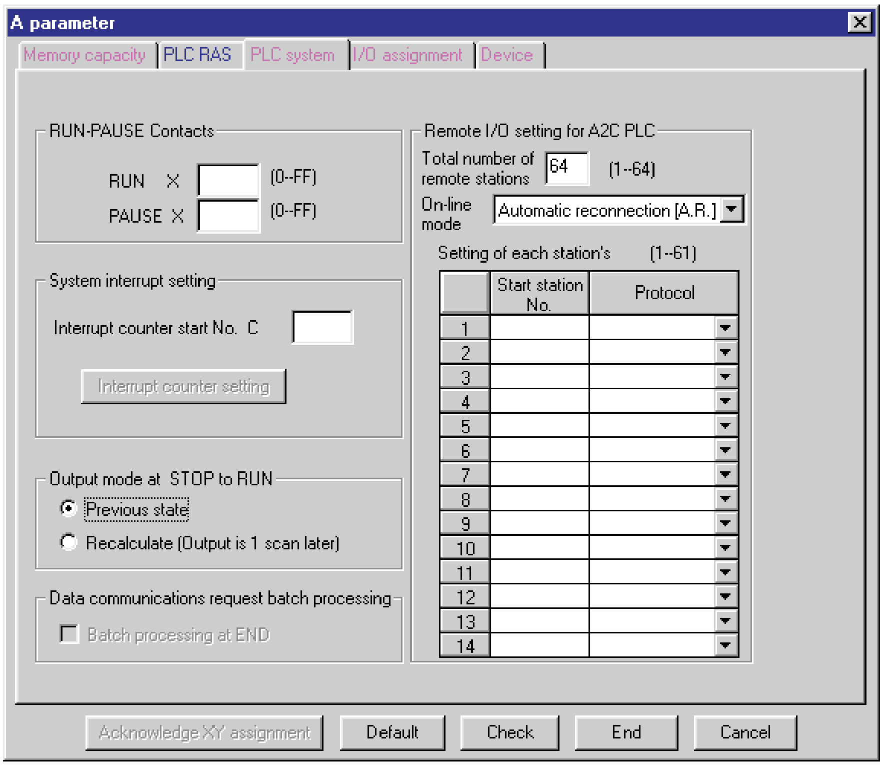
I/O Type: Module type: Input, Output or Special Function Module

I/O Range: Number of addresses allocated in the base unit

Mode name: List of the MITSUBISHI modules that can be used. Depends on the settings made in I/O Type and I/O Range.

5.9.2 I/O Configuration (MELSEC A2C)

The A2C is configured in MINI NET mode. You can install a maximum of 14 stations in the A2C network. The station numbers to be allocated depend on the address numbers of the modules used in the configuration.



RUN-PAUSE Contacts

RUN X (0--FF)

PAUSE X (0--FF)

System interrupt setting

Interrupt counter start No. C

Output mode at STOP to RUN

● Previous state

○ Recalculate (Output is 1 scan later)

Data communications request batch processing

□ Batch processing at END

Remote I/O setting for A2C PLC

Total number of remote stations 64 (1--64)

On-line mode Automatic reconnection [A.R.]

Setting of each station's (1--61)

	Start station No.	Protocol
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

Fig. 5-35: *Input and output configuration*

Remote I/O setting for A2C PLC

Total number of remote stations (1–64): Set the occupied slot count.

Online Mode The Online Mode parameter defines the CPUs operating mode.

Setting of each station

Start station No.: Starting address of the station. The calculation of the number must be performed by the user. The first address is always station number 1. For example, you must add 4 station numbers for each module occupying 32 addresses; the starting address for module 2 would thus be 5.

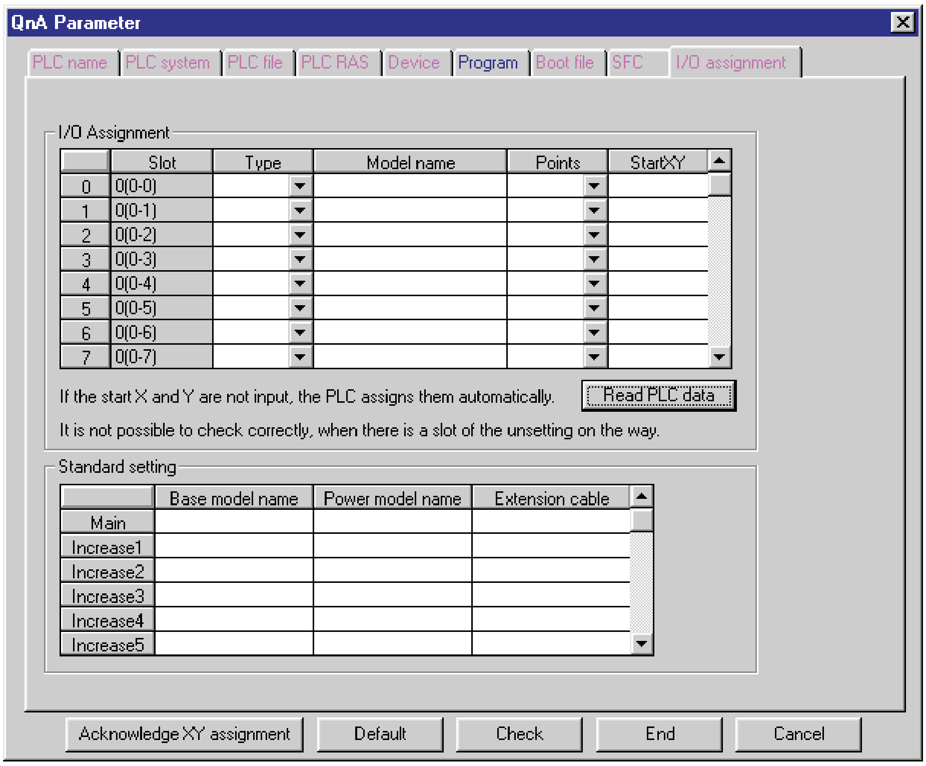
Protocol

MINIstandard: MINI NET protocol, normally used in most situations.

Non procedural: If you are using the AJ35PTF RS-232 module select the non procedural protocol to transfer freely-defined protocols (texts).

5.9.3 I/O Configuration (MELSEC QnA)

This dialogue box is used to enter all the modules and extension units installed in the system.



I/O Assignment

	Slot	Type	Model name	Points	StartXY
0	0(0-0)				
1	0(0-1)				
2	0(0-2)				
3	0(0-3)				
4	0(0-4)				
5	0(0-5)				
6	0(0-6)				
7	0(0-7)				

If the start X and Y are not input, the PLC assigns them automatically.

It is not possible to check correctly, when there is a slot of the unsetting on the way.

Standard setting

	Base model name	Power model name	Extension cable
Main			
Increase1			
Increase2			
Increase3			
Increase4			
Increase5			

Fig. 5-36: *Input and output configuration*

I/O Assignment

Slot List: All slots (max. 64)

Slot: Selected slot

Type: Input module, output module, special function module or spare slot

Points: Number of addresses occupied by the slot

StartXY: Starting address of the module in units of 16 addresses

Model name: List of supported MITSUBISHI modules

Standard setting

Extension List: 1 base unit and up to a maximum of 7 extension units

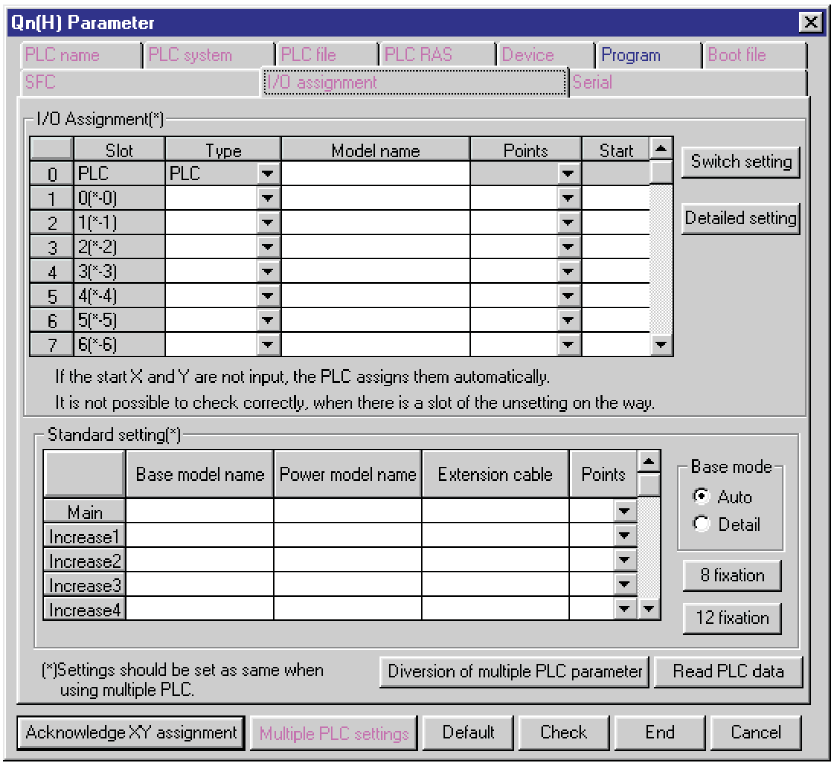
Base model name: Designation of the base unit

Power model name: Designation of the power supply

Extension cable: Designation of the extension cable

5.9.4 I/O Configuration (MELSEC Q)

This dialogue box is used to enter all the modules and extension units installed in the system.



I/O Assignment(*)

	Slot	Type	Model name	Points	Start
0	PLC	PLC			
1	0(*-0)				
2	1(*-1)				
3	2(*-2)				
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If the start X and Y are not input, the PLC assigns them automatically.
It is not possible to check correctly, when there is a slot of the unsetting on the way.

Standard setting(*)

	Base model name	Power model name	Extension cable	Points
Main				
Increase1				
Increase2				
Increase3				
Increase4				

Base mode: ● Auto ○ Detail

(*)Settings should be set as same when using multiple PLC.

Fig. 5-37: Input and output configuration

I/O Assignment

Slot List: All slots (max. 64)

Slot: Selected slot

Type: Input module, output module, special function module or spare slot

Points: Number of addresses occupied by the slot

StartXY: Starting address of the module in units of 16 addresses

Model name: List of supported MITSUBISHI modules

Standard setting

Extension List: 1 base unit and up to a maximum of 7 extension units

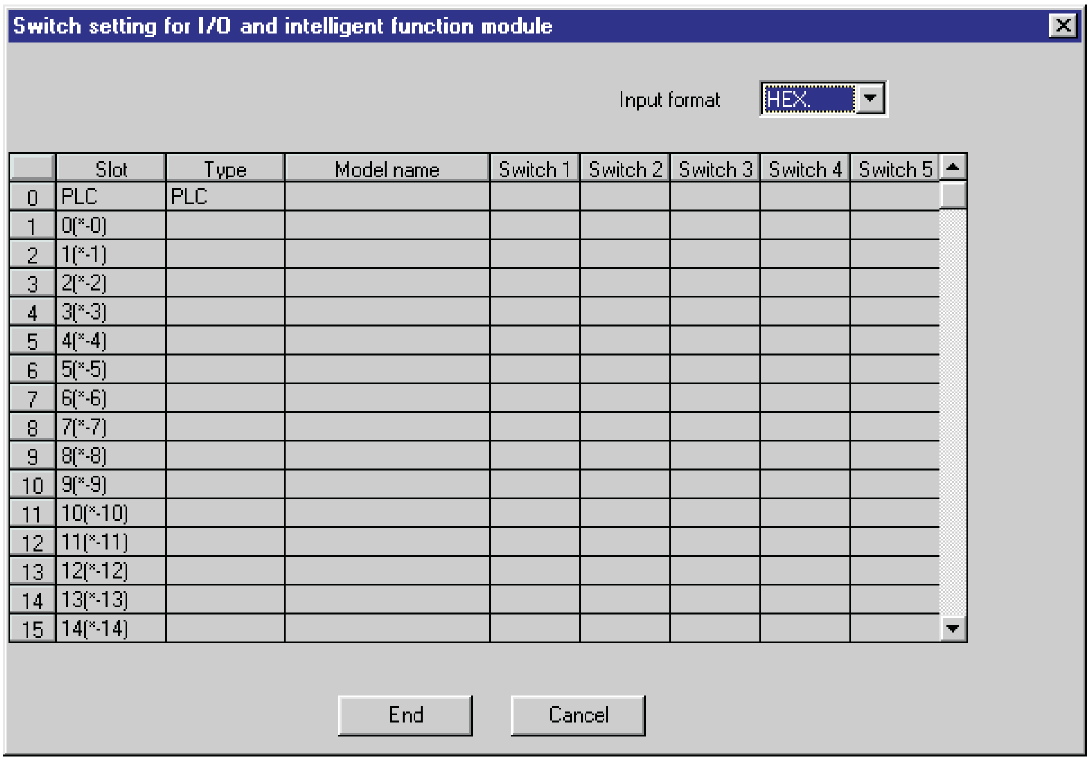
Base model name: Designation of the base unit

Power model name: Designation of the power supply

Extension cable: Designation of the extension cable

Switch setting

This button opens the following dialogue:



Input format HEX.

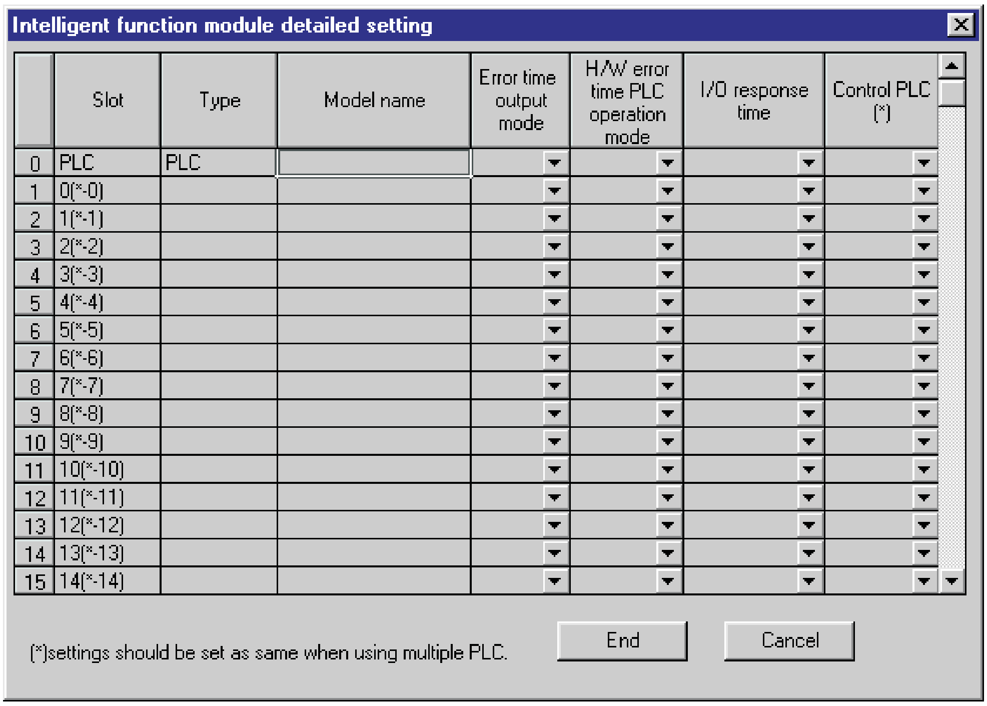
	Slot	Type	Model name	Switch 1	Switch 2	Switch 3	Switch 4	Switch 5
0	PLC	PLC						
1	0(*-0)							
2	1(*-1)							
3	2(*-2)							
4	3(*-3)							
5	4(*-4)							
6	5(*-5)							
7	6(*-6)							
8	7(*-7)							
9	8(*-8)							
10	9(*-9)							
11	10(*-10)							
12	11(*-11)							
13	12(*-12)							
14	13(*-13)							
15	14(*-14)							

Fig. 5-38: *Switch settings for Intelligent Function Modules*

Switch 1 – 5: Settings for switching operations. Device range: 0000h – FFFFh.

Detailed setting

This button opens the following dialogue:



Intelligent function module detailed setting

	Slot	Type	Model name	Error time output mode	H/W error time PLC operation mode	I/O response time	Control PLC (*)
0	PLC	PLC					
1	0(*-0)						
2	1(*-1)						
3	2(*-2)						
4	3(*-3)						
5	4(*-4)						
6	5(*-5)						
7	6(*-6)						
8	7(*-7)						
9	8(*-8)						
10	9(*-9)						
11	10(*-10)						
12	11(*-11)						
13	12(*-12)						
14	13(*-13)						
15	14(*-14)						

(*)settings should be set as same when using multiple PLC.

Fig. 5-39: Detailed settings for Intelligent Function Modules

Error time output mode: Select whether to clear or hold the output value in case of an error.

H/W error time PLC operation mode: Select whether to stop or continue the operation in case of an error.

I/O response time: Select the response time after which the I/O values have reached a valid value.

Control PLC: For Multiple PLC operation (see section 5.14). Assign a selected control PLC from the drop down list to the selected Intelligent Function Module. The control PLC stores the program to control the Intelligent Function Module.

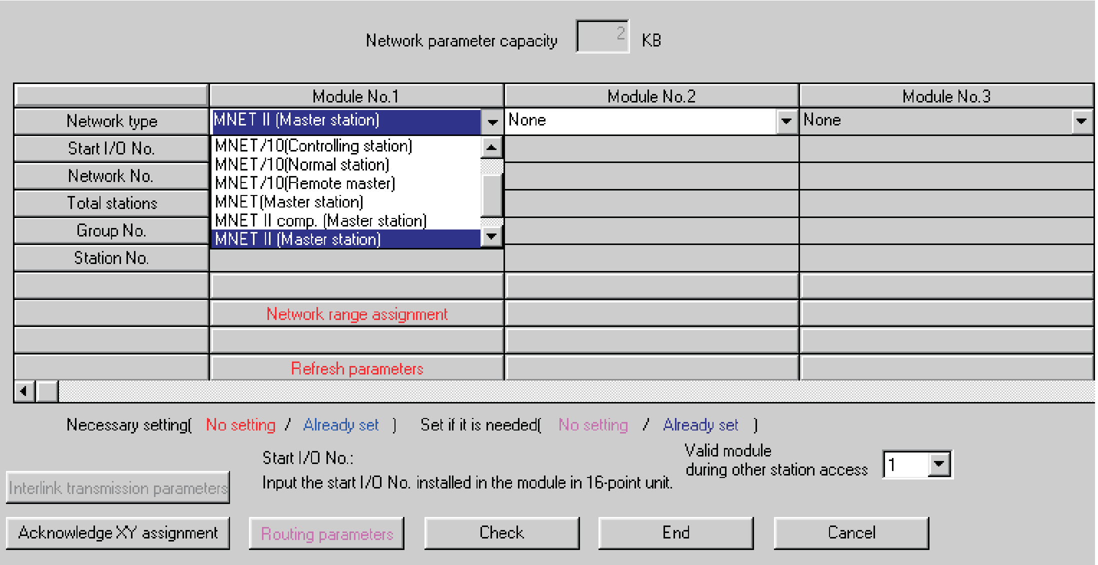
5.10 MELSECNET / Ethernet

The MELSECNET/Ethernet settings define the hardware, communications addresses and paths in a network. The dialogues differ depending on the selected CPU module. The FX family CPUs do not support network communications. For details refer to the manuals of the network modules.

5.10.1 A series

GX IEC Developer supports MELSECNET/10, MELSECNET (II) and MELSECNET/MINI connections to the AnU CPU and A2US CPU modules.

MELSECNET/10, MELSECNET (II)



Network parameter capacity 2 KB

	Module No.1	Module No.2	Module No.3
Network type	MNET II (Master station)	None	None
Start I/O No.			
Network No.			
Total stations			
Group No.			
Station No.			
	Network range assignment		
	Refresh parameters		

MNET/10(Controlling station)
MNET/10(Normal station)
MNET/10(Remote master)
MNET(Master station)
MNET II comp. (Master station)
MNET II (Master station)

Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set)

Start I/O No.:
Input the start I/O No. installed in the module in 16-point unit.

Valid module
during other station access 1

Fig. 5-40: MELSECNET parameters

Module No. 1...4: Here you select the connection type for the network module.

Network type: Select the desired network type.

Start I/O No.: Enter the first I/O number for the network module in hexadecimal format.

Network No.: Enter the network number for MELSECNET/10.

Total stations: Enter the total number of stations.

Group No.: Enter the group number.

Station No.: Enter the station number.

Network range assignment: Opens a dialogue to enter and edit the network range assignment.

Station inherent parameters: Opens a dialogue to enter and edit the station inherent parameters.

Refresh parameters: Opens a dialogue to enter and edit the refresh parameters of the network module for the MELSEC network.

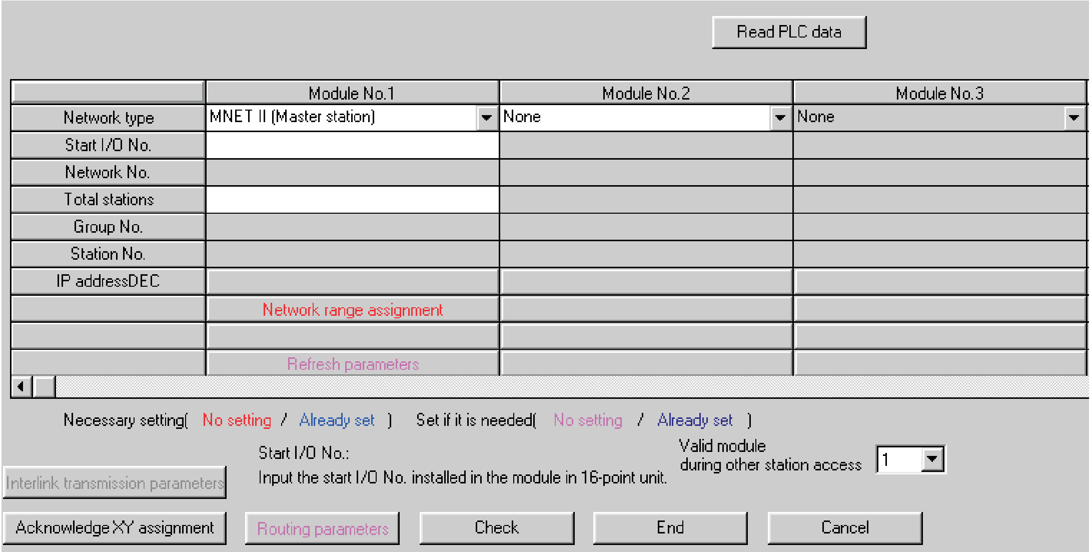
5.10.2 QnA series

GX IEC Developer supports MELSECNET/10, MELSECNET (II), and Ethernet connections to the QnA CPU module.

Restrictions:

- The total number of network modules (MELSECNET modules + Ethernet modules) must not exceed four.
- The communication via Ethernet to the QnA series is fixed to the binary protocol.
- The communication via a Computer Link module to the QnA series is fixed to the format 1.
- The communication via Ethernet and MELSECNET (II) to the QnA series is not supported.

MELSECNET/10, MELSECNET (II)



	Module No.1	Module No.2	Module No.3
Network type	MNET II (Master station)	None	None
Start I/O No.			
Network No.			
Total stations			
Group No.			
Station No.			
IP addressDEC			
	Network range assignment		
	Refresh parameters		

Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set)

Start I/O No.:
Input the start I/O No. installed in the module in 16-point unit.

Valid module during other station access 1

Fig. 5-41: Network parameters

Module No. 1...4: Here you select the connection type for the network module.

Network type: Select the desired network type.

Start I/O No.: Enter the first I/O number for the network module in hexadecimal format.

Network No.: Enter the network number for MELSECNET/10.

Total stations: Enter the total number of stations.

Group No.: Enter the group number.

Station No.: Enter the station number.

Network range assignment: Opens a dialogue to enter and edit the network range assignment.

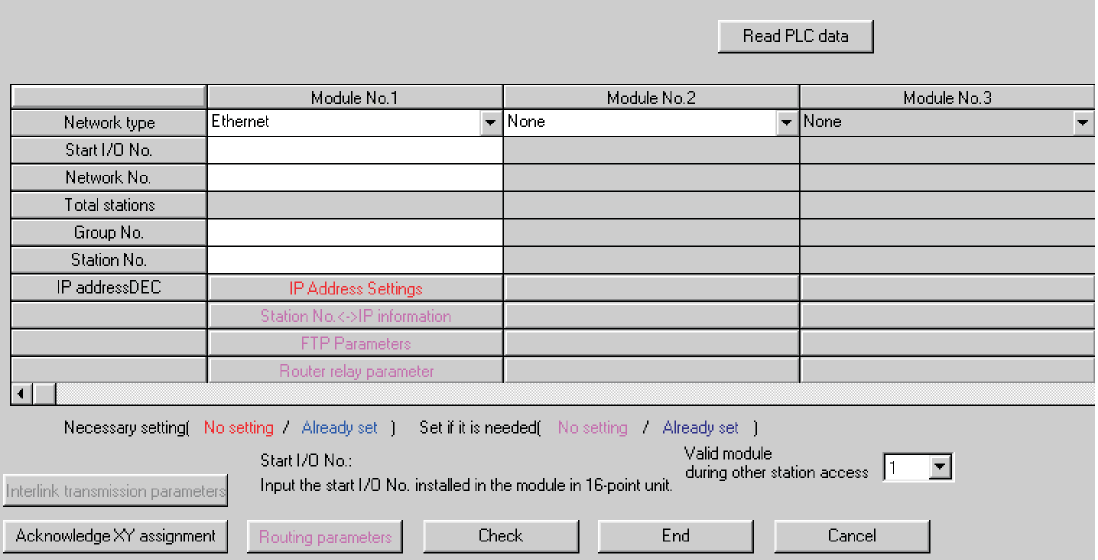
Station inherent parameters: Opens a dialogue to enter and edit the station inherent parameters.

Refresh parameters: Opens a dialogue to enter and edit the refresh parameters of the network module for the MELSEC network.

Stand by station compatible module: Opens a dialogue to enter the standby station compatible module.

Remote station compatible module: Opens a dialogue to enter the remote station compatible module.

Ethernet Settings



	Module No.1	Module No.2	Module No.3
Network type	Ethernet	None	None
Start I/O No.			
Network No.			
Total stations			
Group No.			
Station No.			
IP addressDEC	IP Address Settings		
	Station No.<->IP information		
	FTP Parameters		
	Router relay parameter		

Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set)

Start I/O No.:
Input the start I/O No. installed in the module in 16-point unit.

Valid module during other station access: 1

Fig. 5-42: Ethernet settings

Module No. 1...4: Here you select the connection type for the network module.

Network type: Select the desired network type.

Start I/O No.: Enter the first I/O number for the network module in hexadecimal format.

Network No.: Enter the network number for Ethernet.

Total stations: Enter the total number of stations.

Group No.: Enter the group number.

Station No.: Enter the station number.

IP address DEC: Opens a dialogue to enter the IP address in decimal coding. Each part may contain a value from 0 – 255.

Station No. <-> IP information: Opens a dialogue to enter and edit system for the relation between station No. and IP information.

FTP parameters: Opens a dialogue to enter and edit the FTP parameters.

Router relays parameters: Opens a dialogue to enter and edit router relay parameters.

After the Ethernet settings have been set up successfully the modules will be sorted according to their first I/O numbers.

NOTES

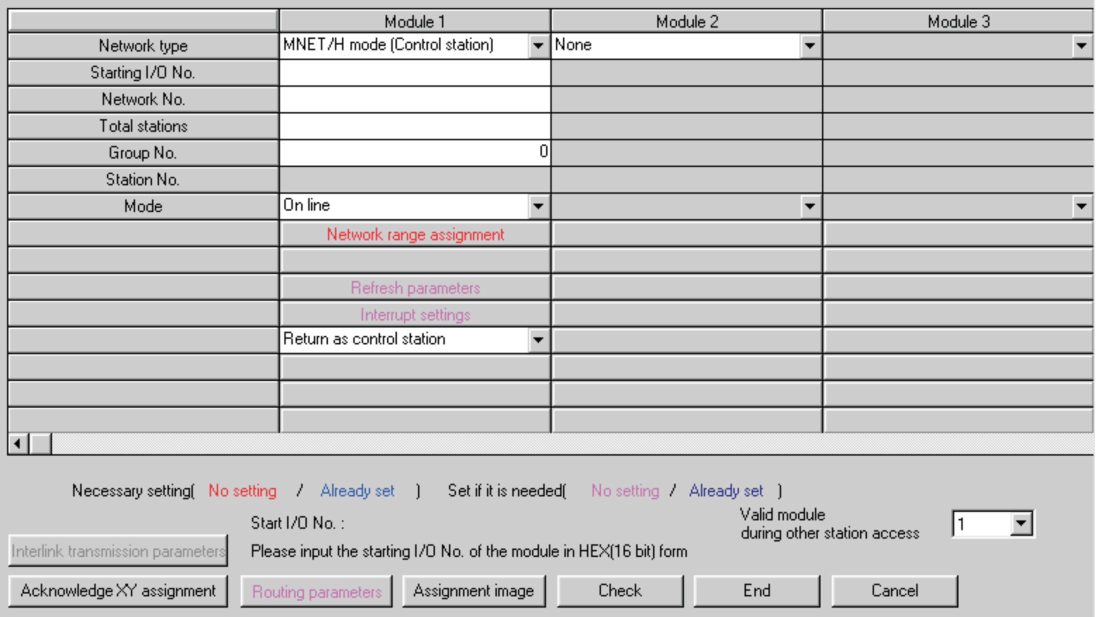
The connection to the PLC via Ethernet is only supported for the AJ71QE71/A1SJ71QE71 from version B or higher and the QnA or QnAS CPU. The product version is printed on the device label. For details refer to the hardware manual of the device.

Do not apply a Windows Forced Close Operation (CTRL + ALT + DEL) during the communication between GX IEC Developer and the PLC or an communication error will occur when starting GX IEC Developer the next time. In this case restart Microsoft Windows.

5.10.3 Q series

GX IEC Developer supports MELSECNET/H, MELSECNET/10, Ethernet and CC-Link connections to the Q CPU module.

MELSECNET/H / MELSECNET/10 parameters



	Module 1	Module 2	Module 3
Network type	MNET/H mode (Control station)	None	
Starting I/O No.			
Network No.			
Total stations			
Group No.	0		
Station No.			
Mode	On line		
	Network range assignment		
	Refresh parameters		
	Interrupt settings		
	Return as control station		

Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set)

Start I/O No. :

Please input the starting I/O No. of the module in HEX(16 bit) form

Valid module during other station access 1

Fig. 5-43: MELSECNET/H / MELSECNET/10 parameters

Module No. 1...4: Here you select the connection type for the network module.

Network type: Select the desired network type.

Start I/O No.: Range: 0 – FFF. Enter the starting I/O number for the network module in hexadecimal format. The starting I/O number must be a multiple of 16 (10h). The starting I/O numbers must not overlap even with defined Ethernet modules and keep a gap of at least 20h.

Network No.: Range: 1 – 239. Enter the network number. The network number must not overlap with other defined modules.

Total stations: Enter the total number of stations.

Group No.: Range: 0 – 32. Enter the group number.

Station No.: Enter the station number.

Mode: Select the desired mode.

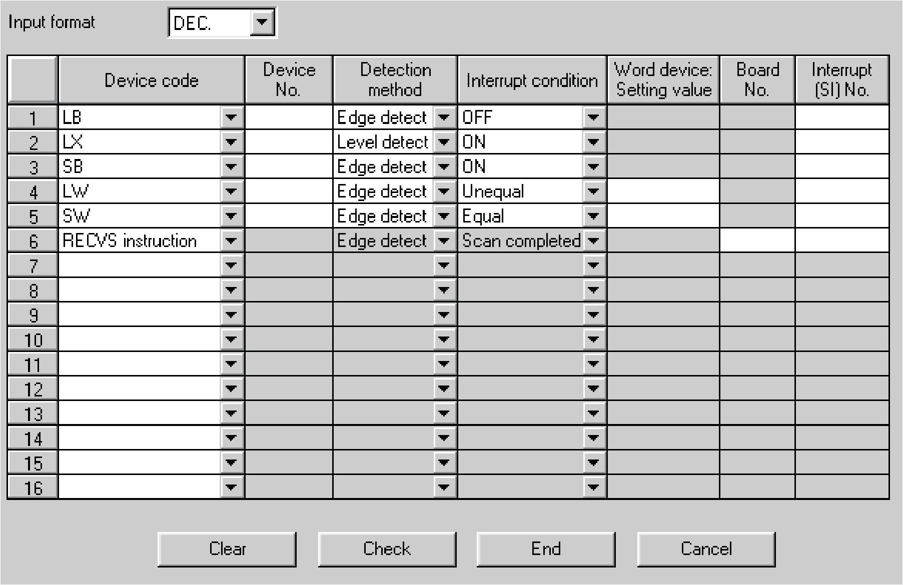
Network range assignment: Opens a dialogue to enter and edit the network range assignment.

Refresh parameters: Opens a dialogue to enter and edit the refresh parameters of the network module for the MELSEC network.

Interrupt settings: Opens a dialogue to enter and edit the interrupt settings for the selected module (not for Q00(J), Q01).

Interrupt settings (not for Q00(J), Q01)

In this dialogue you edit the MELSECNET interrupt settings.



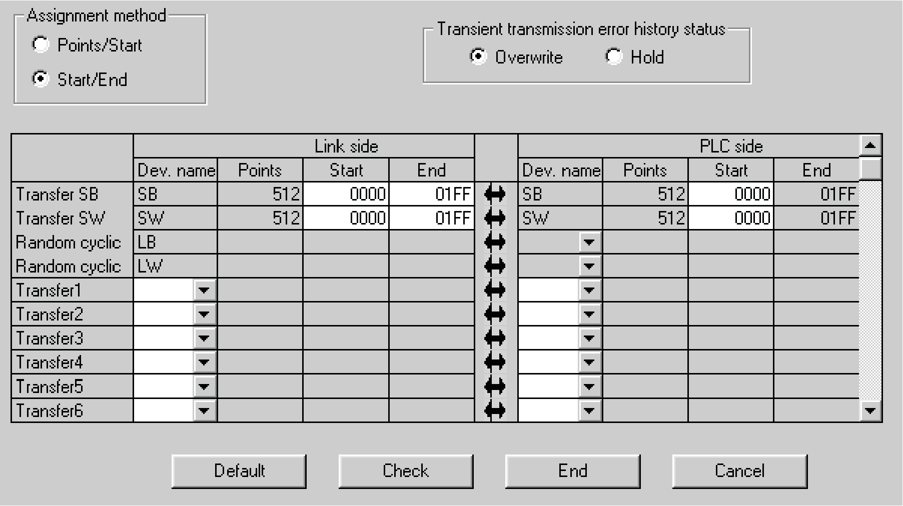
Input format: DEC.

	Device code	Device No.	Detection method	Interrupt condition	Word device: Setting value	Board No.	Interrupt (SI) No.
1	LB		Edge detect	OFF			
2	LX		Level detect	ON			
3	SB		Edge detect	ON			
4	LW		Edge detect	Unequal			
5	SW		Edge detect	Equal			
6	RECVS instruction		Edge detect	Scan completed			
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Fig. 5-44: *MELSECNET interrupt settings*

Refresh parameters

In this dialogue you edit the refresh parameters for a selected Q module.



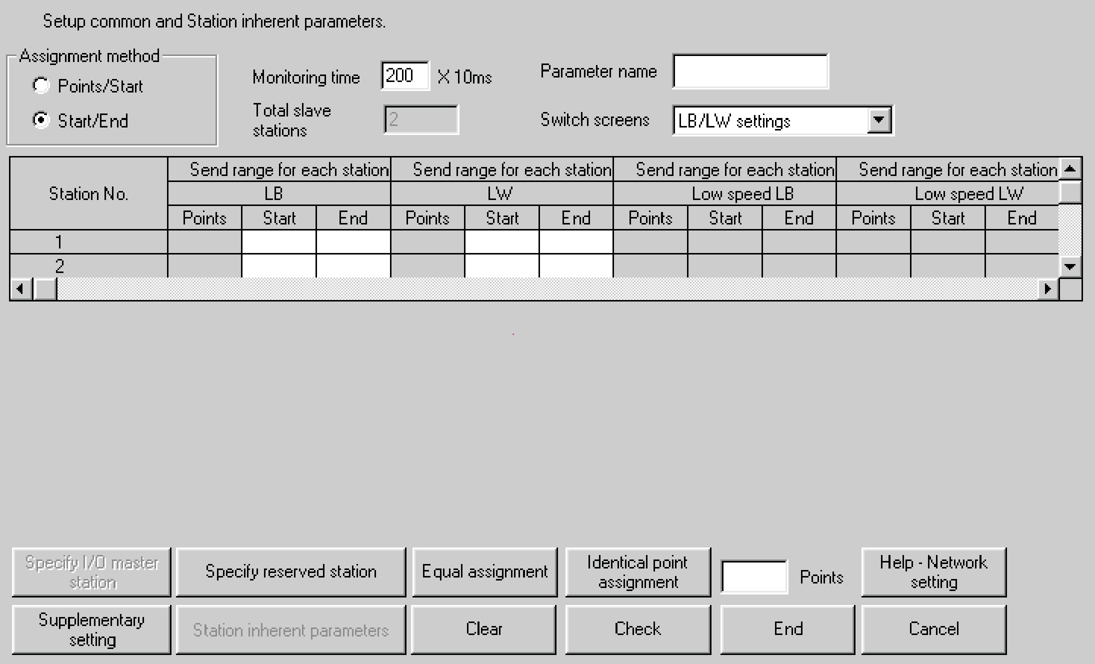
Assignment method
○ Points/Start
● Start/End

Transient transmission error history status
● Overwrite ○ Hold

	Link side					PLC side			
	Dev. name	Points	Start	End		Dev. name	Points	Start	End
Transfer SB	SB	512	0000	01FF	↔	SB	512	0000	01FF
Transfer SW	SW	512	0000	01FF	↔	SW	512	0000	01FF
Random cyclic	LB				↔				
Random cyclic	LW				↔				
Transfer1					↔				
Transfer2					↔				
Transfer3					↔				
Transfer4					↔				
Transfer5					↔				
Transfer6					↔				

Fig. 5-45: *Refresh parameters*

Common parameters



Setup common and Station inherent parameters.

Assignment method
○ Points/Start
● Start/End

Monitoring time 200 X 10ms

Total slave stations 2

Parameter name

Switch screens LB/LW settings

Station No.	Send range for each station			Send range for each station			Send range for each station			Send range for each station		
	LB			LW			Low speed LB			Low speed LW		
	Points	Start	End	Points	Start	End	Points	Start	End	Points	Start	End
1												
2												

Fig. 5-46: *Common network parameters*

I/O Master station specification: Using Switch screens to choose LX/LY settings enables the I/O master station to be specified. To set the I/O master station, specify the station number with the cursor and click **I/O Master station specification**.

Reserved station specification: Specify the reserved station No. and click this button.

Equal assignment: Used to assign the link device points of all stations equally. The first and last stations can be set within the number of stations between the first and last equally assigned station numbers (total number of link stations - (first station number - 1)).

Identical point assignment: Used to make simple assignment with the same points according to the preset total number of stations.

Supplementary settings: Used to set the constant scan, max. number of reconnected stations during 1 scan, multiple transmission, communications error setting and transient setting.

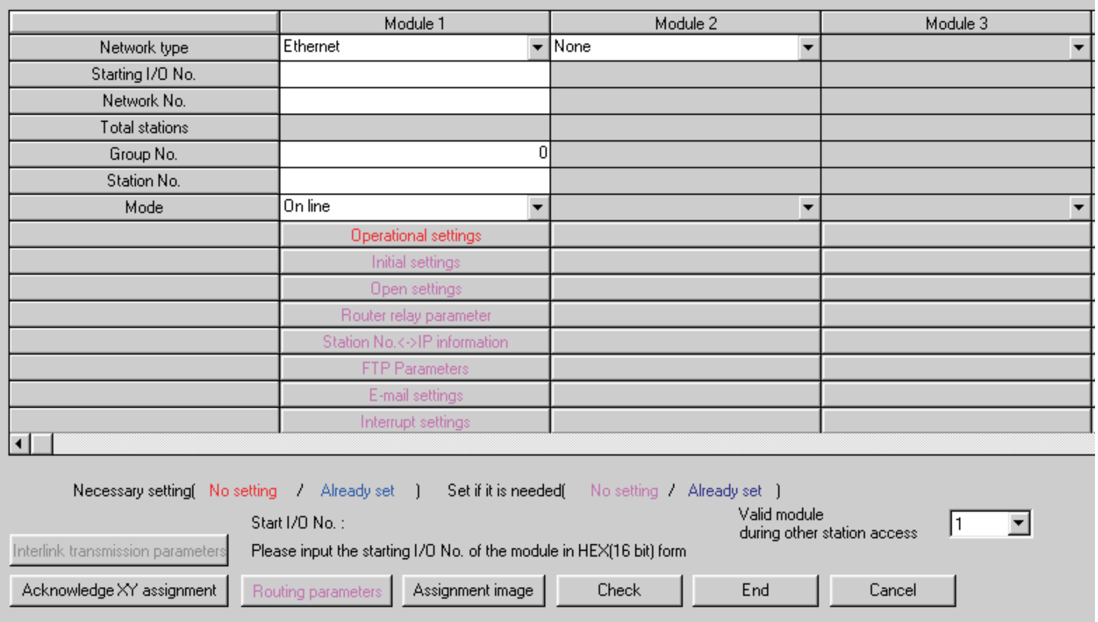
Station inherent parameters: Used to change the network unit storage locations of the link devices assigned to each station in the common parameters.

Clear: Used to default the parameter settings.

Check: Used to check the preset parameter data.

End setup: Clicking this button at the end of data setting returns to the network setting screen.

Ethernet parameters



	Module 1	Module 2	Module 3
Network type	Ethernet	None	
Starting I/O No.			
Network No.			
Total stations			
Group No.	0		
Station No.			
Mode	On line		
	Operational settings		
	Initial settings		
	Open settings		
	Router relay parameter		
	Station No.<->IP information		
	FTP Parameters		
	E-mail settings		
	Interrupt settings		

Necessary setting(No setting / Already set) Set if it is needed(No setting / Already set)

Start I/O No. :

Please input the starting I/O No. of the module in HEX(16 bit) form

Valid module during other station access: 1

__Fig. 5-47:__ Ethernet parameters

Module No.1...4: Here you select the connection type for the network module.

Network type: Select the desired network type from the drop-down list. Select “**None**” to clear the parameter set.

Start I/O No.: Range: 0 – FFF. Enter the starting I/O number for the network module in hexadecimal format. The starting I/O number must be a multiple of 16 (10h). The starting I/O numbers must not overlap even with defined MELSECNET modules and keep a gap of at least 20h.

Network No.: Range: 1 – 239. Enter the network number. The network number must not overlap with other defined modules.

Total stations: Enter the total number of stations.

Group No.: Range: 0 – 32. Enter the group number.

Station No.: Range: 0 – 64. Enter the station number.

Operational settings: Opens a dialogue to enter the Ethernet operations like the IP address in decimal coding.

Initial settings: Opens a dialogue to enter the Timer and DNS settings.

Open settings: Opens a dialogue to enter settings when connecting multiple MELSOFT products such as GX IEC Developer (a maximum of 17 units) using the TCP/IP communication system. See also section 5.3.4.

Router relay settings: Opens a dialogue to enter the router relay settings.

Station No. <-> IP information: Opens a dialogue to enter and edit system for the relation between station No. and IP information.

FTP parameters: Opens a dialogue to enter and edit the FTP parameters.

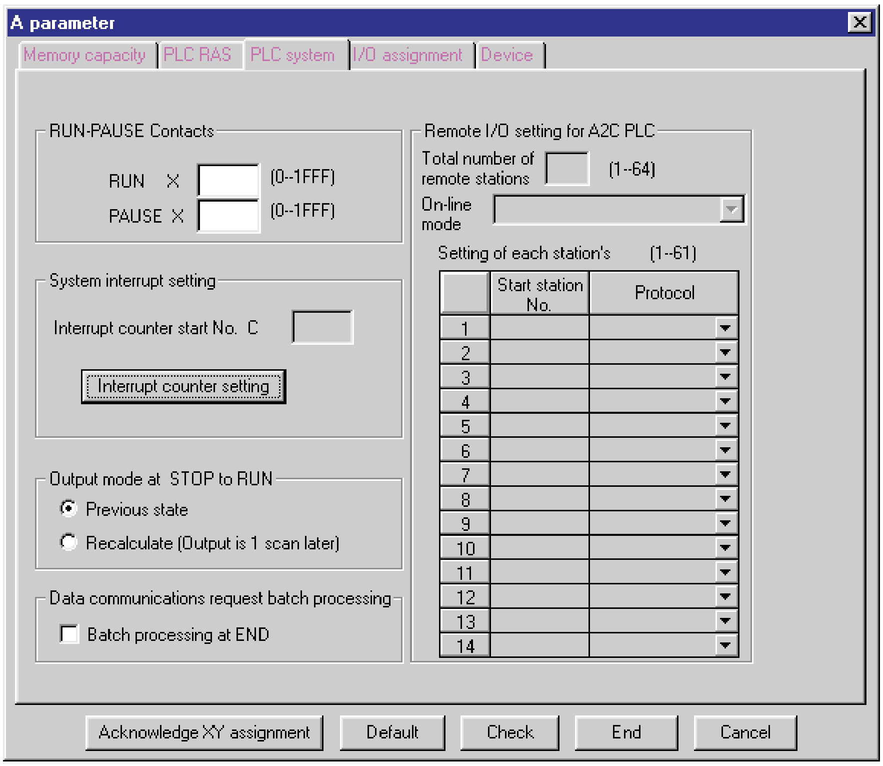
Router relays parameters: Opens a dialogue to enter and edit router relay parameters.

E-mail settings: Opens a dialogue to enter the settings for sending and receiving e-mails.

Interrupt settings: Opens a dialogue to edit the Ethernet interrupt settings (not for Q00(J), Q01).

5.11 PLC Setup

5.11.1 PLC Setup (MELSEC A)



RUN-PAUSE Contacts

RUN X (0--1FFF)

PAUSE X (0--1FFF)

System interrupt setting

Interrupt counter start No. C

Output mode at STOP to RUN

● Previous state

○ Recalculate (Output is 1 scan later)

Data communications request batch processing

□ Batch processing at END

Remote I/O setting for A2C PLC

Total number of remote stations (1--64)

On-line mode

Setting of each station's (1--61)

	Start station No.	Protocol
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

Fig. 5-48: PLC Setup

RUN-PAUSE Contacts: These fields are used for entering the inputs (X) used for controlling the PLC’s operating mode.

Run The PLC is set to RUN mode as soon as this input is on.

Pause When this input is on the PLC is switched to PAUSE mode.
Relay M9040 must be set for this to be possible.

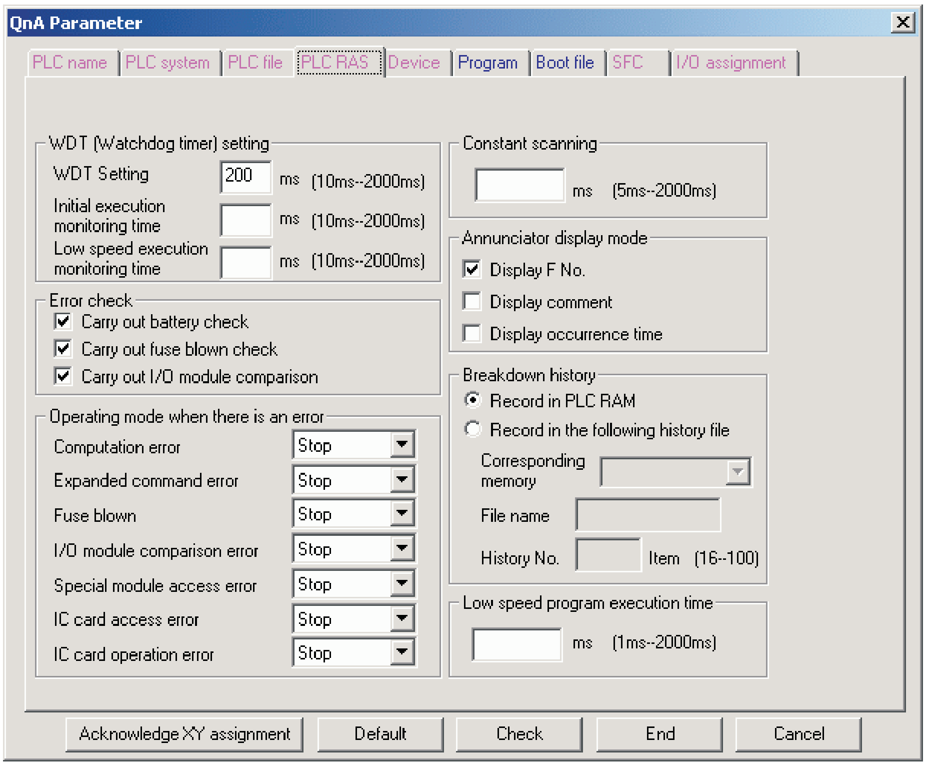
System interrupt setting: Set the range of the counter used in an interrupt program.

Output mode at STOP to RUN: Behaviour of the PLC CPU after a system stop and restart of the process.

Previous State	Resume process
Recalculate State	Recalculate

Data communications request batch processing: Set whether the transient communications request is processed or not in the END processing of its scan.

5.11.2 PLC Setup (MELSEC Q/QnA)



QnA Parameter

PLC name | PLC system | PLC file | PLC RAS | Device | Program | Boot file | SFC | I/O assignment

WDT (Watchdog timer) setting
- WDT Setting: 200 ms (10ms--2000ms)
- Initial execution monitoring time: ms (10ms--2000ms)
- Low speed execution monitoring time: ms (10ms--2000ms)

Error check
- ☑ Carry out battery check
- ☑ Carry out fuse blown check
- ☑ Carry out I/O module comparison

Operating mode when there is an error
- Computation error: Stop
- Expanded command error: Stop
- Fuse blown: Stop
- I/O module comparison error: Stop
- Special module access error: Stop
- IC card access error: Stop
- IC card operation error: Stop

Constant scanning
- ms (5ms--2000ms)

Annunciator display mode
- ☑ Display F No.
- □ Display comment
- □ Display occurrence time

Breakdown history
- ● Record in PLC RAM
- ○ Record in the following history file
- Corresponding memory
- File name
- History No. Item (16--100)

Low speed program execution time
- ms (1ms--2000ms)

Acknowledge XY assignment | Default | Check | End | Cancel

Fig. 5-49: *PLC RAS*

WDT Setting: Setting for the watchdog timer.
Range: 10 - 2,000 ms in units of 10 ms
(only for Q25SS (SX Controller)
Range: 10 - 5,000 ms in units of 10 ms)

Error Check: Enables you to activate/deactivate the following error checking functions:

- Battery
- Fuse
- I/O modules

Operation Mode at Error: These options enable you to specify whether the CPU is to pause or continue working when the following error conditions are registered:

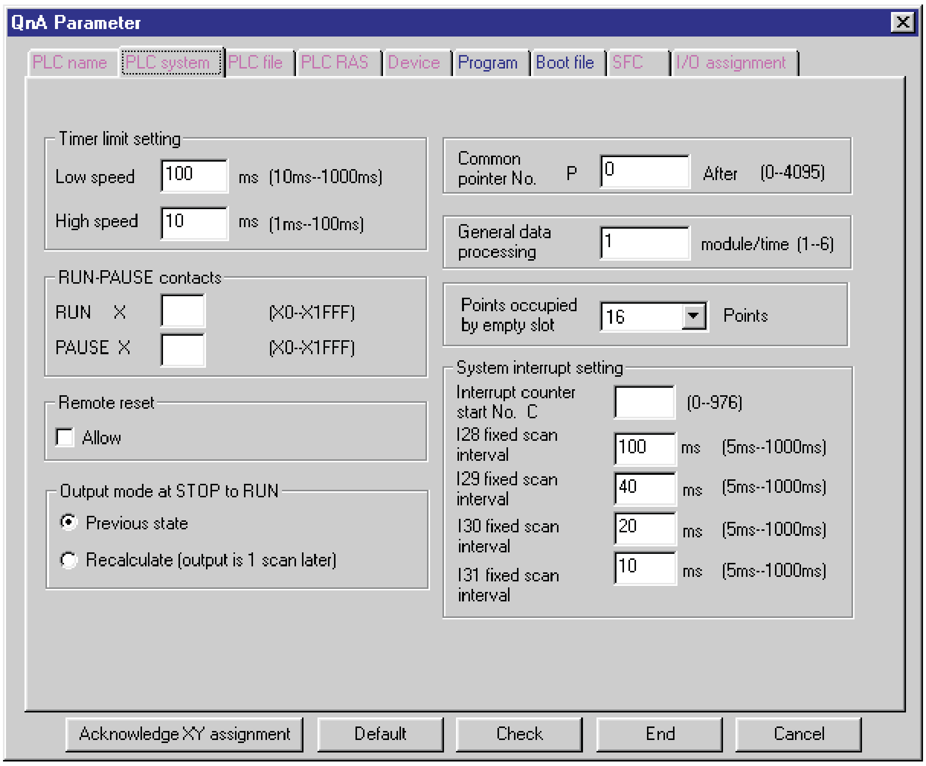
- Calculation error
- Extended Ins
- Fuse blown
- I/O unit compare error
- Special function unit access error
- IC card access error (not for Q00(J), Q01)
- IC card operation error (not for Q00(J), Q01)

Constant Scanning: Enter the constant scan cycle period here.
Range: 5 - 2,000 ms in units of 5 ms

Number of free slots: Enter the number of addresses occupied by a free slot in this field.
Range: 0 - 1024 in units of 16

Annunciator display mode: Set the display mode established when the annunciator has turned on.

Breakdown history: Set the storage target of the CPU's breakdown history.



Timer limit setting
Low speed 100 ms (10ms--1000ms)
High speed 10 ms (1ms--100ms)

RUN-PAUSE contacts
RUN X (X0--X1FFF)
PAUSE X (X0--X1FFF)

Remote reset
□ Allow

Output mode at STOP to RUN
● Previous state
○ Recalculate (output is 1 scan later)

Common pointer No. P 0 After (0--4095)

General data processing 1 module/time (1--6)

Points occupied by empty slot 16 Points

System interrupt setting
Interrupt counter start No. C (0--976)
I28 fixed scan interval 100 ms (5ms--1000ms)
I29 fixed scan interval 40 ms (5ms--1000ms)
I30 fixed scan interval 20 ms (5ms--1000ms)
I31 fixed scan interval 10 ms (5ms--1000ms)

Fig. 5-50: *PLC system*

Timer limit setting: Used to set the intervals for the high-speed and low-speed timers. The timers will then act as high-speed or low-speed timers when they are activated by the appropriate instructions in the PLC program.

Low-speed timer instructions: TIMER (IEC editor), OUT (MELSEC editor)

High-speed timer instructions: TIMER_H (IEC editor), OUT_H (MELSEC editor)

Low-speed timer range: 10 - 1,000ms in units of 10ms

High-speed timer range: 1 - 100ms in units of 1ms

RUN-PAUSE Contacts: Enter the inputs to be used to control the operating mode of the PLC here.

Range: X0 - X1FFF (QnA series)

The PLC will be switched to RUN or PAUSE mode as soon as the corresponding input is set. When using the PAUSE contact the diagnostics link relay SM206 (QnA series) must also be set.

Remote reset: Set enable/disable of remote reset operation.

Output mode at STOP to RUN: Behaviour of the PLC CPU after a system stop and restart of the process.

Previous State	Resume process
Recalculate State	Recalculate

Common pointer: Set the first No. of the common pointers.

Number of empty slots: Set the number of empty slots.

System interrupt setting: These fields enable you to specify the counters you wish to allocate to the system interrupt signals. A total of 32 interrupts are available (I0 - I31).

1st Int Counter: The first interrupt counter can be an address from the following counter range:
C0 - [sum of all counters minus 48]

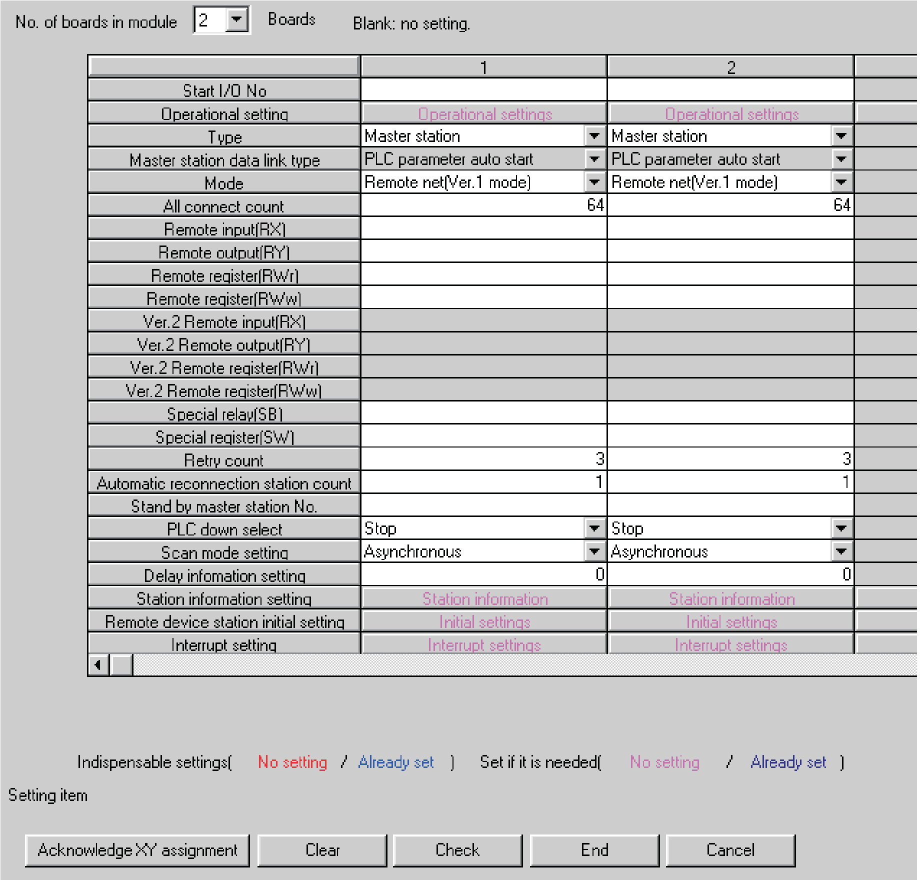
Example:
Assuming you have set 1K of addresses for the counters in the Device Settings dialogue box you have 1,024 counter addresses available (C0 - C1023).

A possible entry for the first interrupt counter would thus be:
C0 - 1024 minus 48, which gives you
C0 - C976

I28 / I29 / I30 / I31 Const Interval: You can set constant interrupt intervals for interrupts I28 - I31.
Range: 5 - 1,000ms in units of 5ms
(only for Q25SS (SX Controller)
Range: 100 - 1,000 ms in units of 10 ms)

5.12 CC-Link (MELSEC Q/QnA)

The CC-Link settings define the hardware, communications addresses and paths in a network. For details refer to the manuals of the network modules.



No. of boards in module 2 Boards Blank: no setting.

	1	2
Start I/O No		
Operational setting	Operational settings	Operational settings
Type	Master station	Master station
Master station data link type	PLC parameter auto start	PLC parameter auto start
Mode	Remote net(Ver.1 mode)	Remote net(Ver.1 mode)
All connect count	64	64
Remote input(RX)		
Remote output(RY)		
Remote register(RWr)		
Remote register(RWw)		
Ver.2 Remote input(RX)		
Ver.2 Remote output(RY)		
Ver.2 Remote register(RWr)		
Ver.2 Remote register(RWw)		
Special relay(SB)		
Special register(SW)		
Retry count	3	3
Automatic reconnection station count	1	1
Stand by master station No.		
PLC down select	Stop	Stop
Scan mode setting	Asynchronous	Asynchronous
Delay infomation setting	0	0
Station information setting	Station information	Station information
Remote device station initial setting	Initial settings	Initial settings
Interrupt setting	Interrupt settings	Interrupt settings

Indispensable settings(No setting / Already set) Set if it is needed(No setting / Already set)

Setting item

Fig. 5-51: CC-Link settings

Number of boards within modules: Before you can edit the CC-Link settings you have to select the number of CC-Link boards from the drop-down list. Several boards can be edited in individual dialogue boxes. Each dialogue box can be selected via tabs.

Start I/O No.: Enter the start I/O number in multiples of 10h. The start I/O number of the individual boards must not overlap. Value Range: 0000 – 0FE0.

Operational setting: Enter the parameter name and data link abnormal station.

Type: Enter the master/local/standby master station, etc.

Master station data link type: Fixed to "PLC parameter auto start".

Mode: Set the mode.

All connect count: Enter the number of slave stations connected. (1 to 64 stations)

Remote input [RX]: Enter the bit devices to be batch-refreshed.

Remote output [RY]: Enter the bit devices to be batch-refreshed.

Remote register [Rwr]: Enter the word devices to be batch-refreshed.

Remote register [Rww]: Enter the word devices to be batch-refreshed.

Special relay [SB]: Enter the word devices to be batch-refreshed.

Special register [SW]: Enter the word devices to be batch-refreshed.

Retry count: Enter the number of transmission retries.

Automatic reconnection station count: Enter the number of stations automatically reconnected during 1 scan.

Standby master station No.: Enter the station number of the standby master station.

PLC down select: Select **Stop** (default) or **Continue** for CC-Link communications in case the PLC has stopped.

Scan mode setting: Enter whether sequence scans are synchronized or not. Select **Asynchronous** (default) or **Synchronous** scan mode.

Delay information settings: Enter the link scan interval delay time.

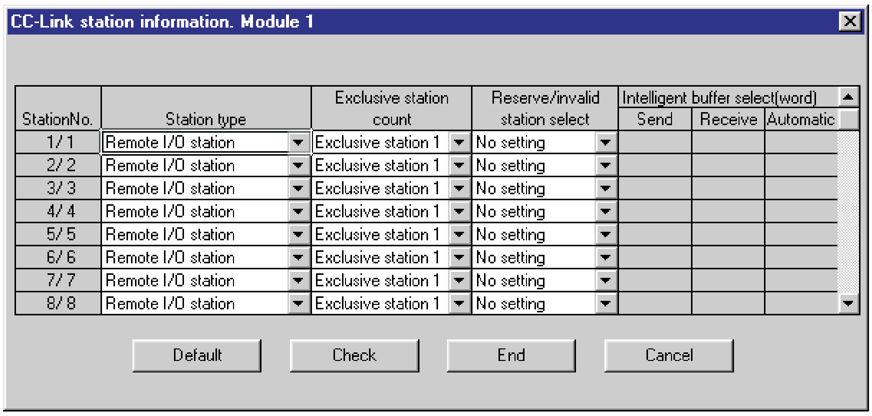
Station information settings: This button opens the **CC-Link station information** dialogue (see below). Enter the station type, number of occupied stations, etc.

Remote device station initial setting: This button opens the **Remote device station initial setting** dialogue (see below). Enter the corresponding station number and procedure registration (operation, execution and other conditions).

Interrupt settings: Opens the **Interrupt settings** dialogue (see below) to edit the Ethernet interrupt settings (not for Q00(J), Q01).

5.12.1 CC-Link station information

In the **CC-Link setting** dialogue click on the **Station information** button to open the following dialogue.



CC-Link station information. Module 1

StationNo.	Station type	Exclusive station count	Reserve/invalid station select	Intelligent buffer select(word)		
				Send	Receive	Automatic
1/1	Remote I/O station	Exclusive station 1	No setting			
2/2	Remote I/O station	Exclusive station 1	No setting			
3/3	Remote I/O station	Exclusive station 1	No setting			
4/4	Remote I/O station	Exclusive station 1	No setting			
5/5	Remote I/O station	Exclusive station 1	No setting			
6/6	Remote I/O station	Exclusive station 1	No setting			
7/7	Remote I/O station	Exclusive station 1	No setting			
8/8	Remote I/O station	Exclusive station 1	No setting			

Fig. 5-52: CC-Link station information

The total number of CC-Link stations you specified in the **CC-Link setting** dialogue in the field **All connect count** are listed. Click on any station number to select a station entry and edit it via the drop-down lists and entry fields on top.

Station type: Select the station type.

Exclusive station count: Select a number for the exclusive station.

Reserve/invalid station select: Enter the station status.

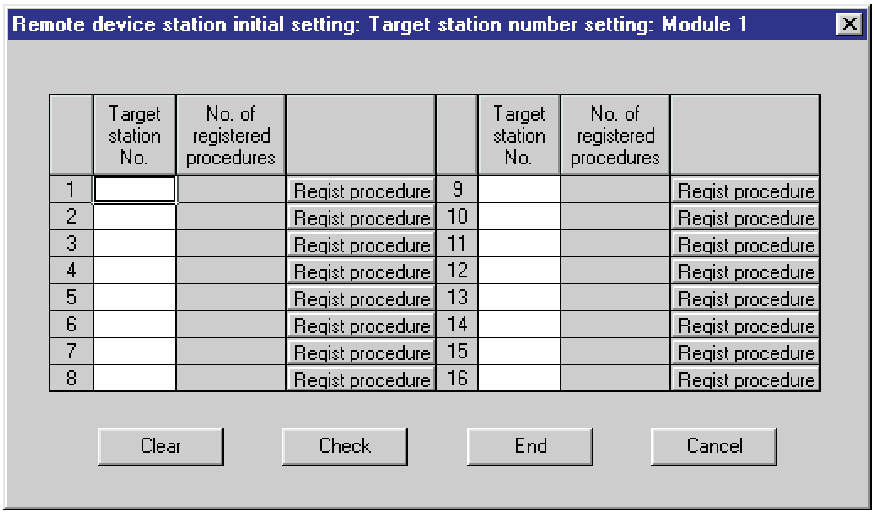
Intelligent buffer select: Enter the buffer sizes. Value ranges: 0 – 4096

Default: This button resets the selected station number to default values.

Check: This button checks the settings of the selected station number for errors.

5.12.2 Remote device station initial setting

In the **CC-Link setting** dialogue click on the **Initial settings** button to open the following dialogue.



Remote device station initial setting: Target station number setting: Module 1

	Target station No.	No. of registered procedures			Target station No.	No. of registered procedures	
1			Regist procedure	9			Regist procedure
2			Regist procedure	10			Regist procedure
3			Regist procedure	11			Regist procedure
4			Regist procedure	12			Regist procedure
5			Regist procedure	13			Regist procedure
6			Regist procedure	14			Regist procedure
7			Regist procedure	15			Regist procedure
8			Regist procedure	16			Regist procedure

Fig. 5-53: *Initial settings of remote device stations*

Target number: Lists the number of the registered target station.

Target station No.: Enter a number for the target station to be registered or edited. Value range: 1– 64.

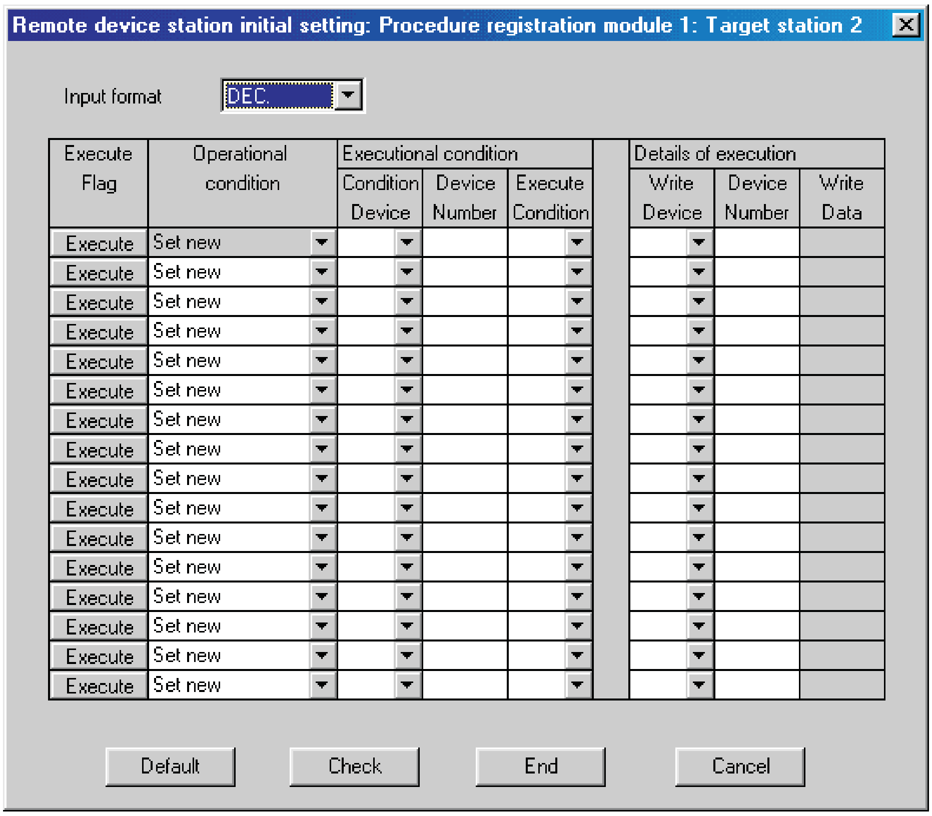
Regist procedure: This button opens the procedure registration dialogue (see below) for a target station either selected from the list or entered in the **Register target station** field.

Clear: This button deletes all entered data.

Check: This button checks a selected entry for setting errors.

Procedure registration dialogue

In the **Remote device station initial setting** dialogue click on the **Regist procedure** button to open the following dialogue.



Remote device station initial setting: Procedure registration module 1: Target station 2

Input format DEC.

Execute Flag	Operational condition	Executional condition				Details of execution		
		Condition Device	Device Number	Execute Condition		Write Device	Device Number	Write Data
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							
Execute	Set new							

Fig. 5-54: Procedure registration for modules

The procedures for the target station are listed. Click on any procedure entry to select and edit it via the drop-down lists and entry fields on top.

Execute flag: Select to set the execute flag.

Operational condition: Select **Same as prev. set** or **Set new**.

Condition device: Select a condition device from the drop-down list.

Device number: Enter address of the condition device.

Execute condition: Select from drop-down list.

Write device: Select from drop-down list.

Device number: Enter address of the write device.

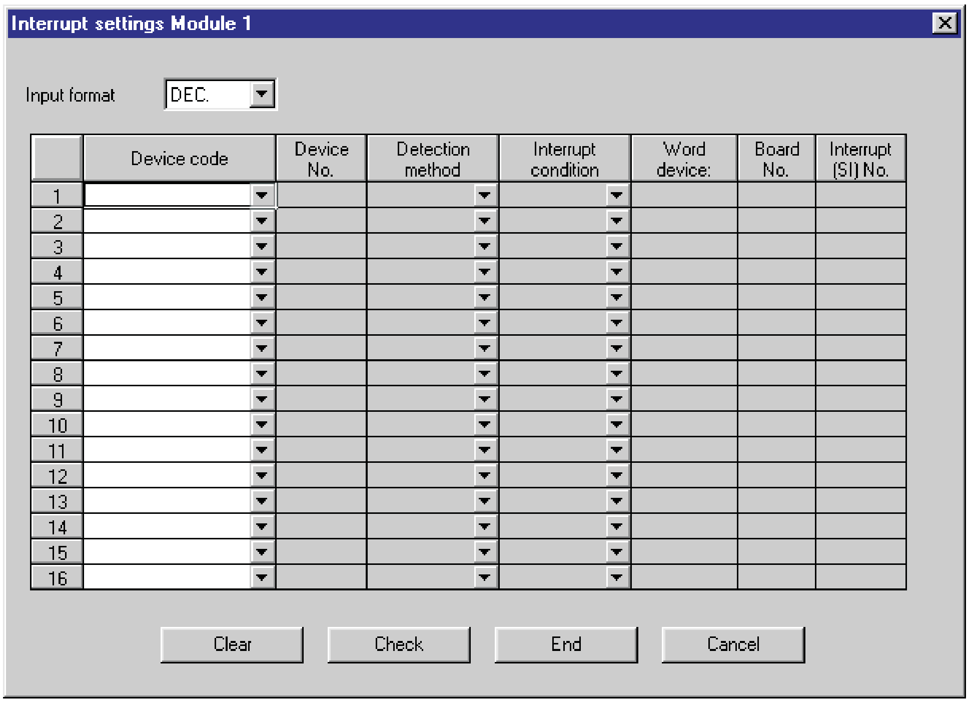
Write data: Select ON or OFF from the drop-down list.

Default: This button resets a selected entry to default values.

Check: This button checks a selected entry for setting errors.

5.12.3 Interrupt settings

In the **CC-Link setting** dialogue click on the **Interrupt settings** button to open the following dialogue.



Interrupt settings Module 1

Input format: DEC.

	Device code	Device No.	Detection method	Interrupt condition	Word device:	Board No.	Interrupt (SI) No.
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Fig. 5-55: Interrupt settings

The interrupt settings for the selected CC-Link board are listed. Click on any entry to select and edit it via the drop-down lists and entry fields on top.

Device code: Select a device code from the drop-down list.

Device No.: Enter a device number. Value range: 0000 – 07FF.

Detection method: Select a detection method from the drop-down list.

Interrupt condition: Select an interrupt condition from the drop-down list.

Word device: Setting value: Enter a setting value for word devices.
Value range: 0 – 65535

Board No.: Enter the board number.

Interrupt (SI) No.: Enter value. Value range: 0 – 15.

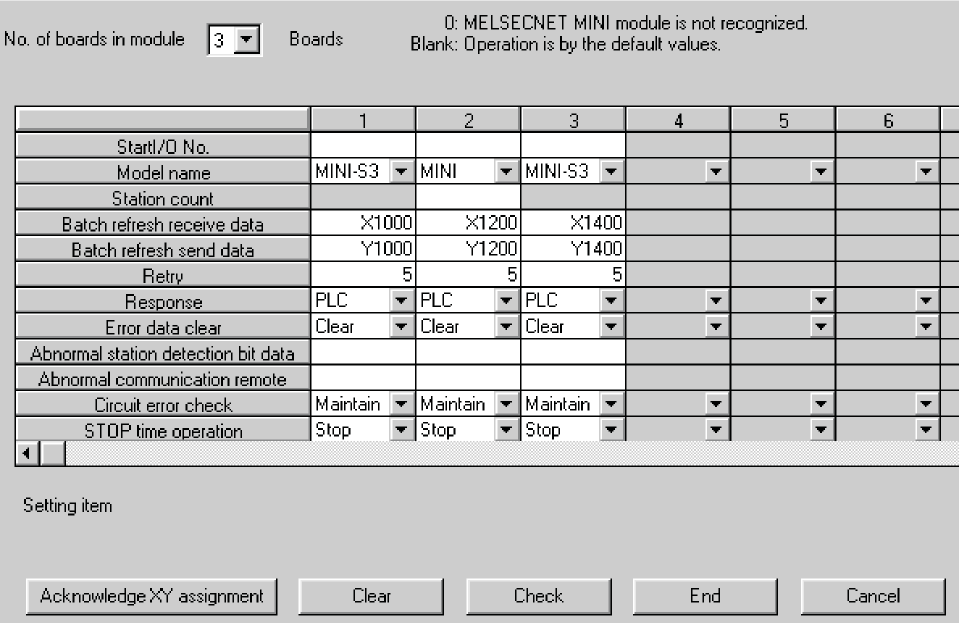
Clear: This button deletes all entered data.

Check: This button checks a selected entry for setting errors.

5.13 MELSECNET/MINI

5.13.1 MELSECNET/MINI (MELSEC A/QnA)

The settings for the MiniNet are made in the table. The line above the buttons displays either a note on correct entry or an error message, depending on the cell selected in the table.



No. of boards in module 3 Boards

0: MELSECNET MINI module is not recognized.
Blank: Operation is by the default values.

	1	2	3	4	5	6
StartI/O No.						
Model name	MINI-S3	MINI	MINI-S3			
Station count						
Batch refresh receive data	X1000	X1200	X1400			
Batch refresh send data	Y1000	Y1200	Y1400			
Retry	5	5	5			
Response	PLC	PLC	PLC			
Error data clear	Clear	Clear	Clear			
Abnormal station detection bit data						
Abnormal communication remote						
Circuit error check	Maintain	Maintain	Maintain			
STOP time operation	Stop	Stop	Stop			

Setting item

Fig. 5-56: *MELSECNET/MINI settings*

Start I/O No.: Enter the first I/O number for the network module in hexadecimal format.

Model name: Selection of the network type: MiniNet or MiniNet-S3

Batch refresh receive data: Definition of the operands for refreshing receive data. The following operands are valid: X, M, L, S, B, T, C, D, W, R

Batch refresh send data: Definition of the operands for refreshing send data. The following operands are valid: X, M, L, S, B, T, C, D, W, R

Retry: Enter the number of communications error retries within the range of 0–32.

Response: Select which device has FROM/TO command buffer memory access priority rights, the link or the PLC.

Error data clear: The status of data transmission in the event of a communication error. The following entries are possible:
TEST Output of a test string
OFF Transmission of a 00 string to the remote I/O station
HOLD Sends data to the remote I/O station shortly before communication

Abnormal station detection bit data: Enter the M, L, S, B, T, C, D, W, or R device name.

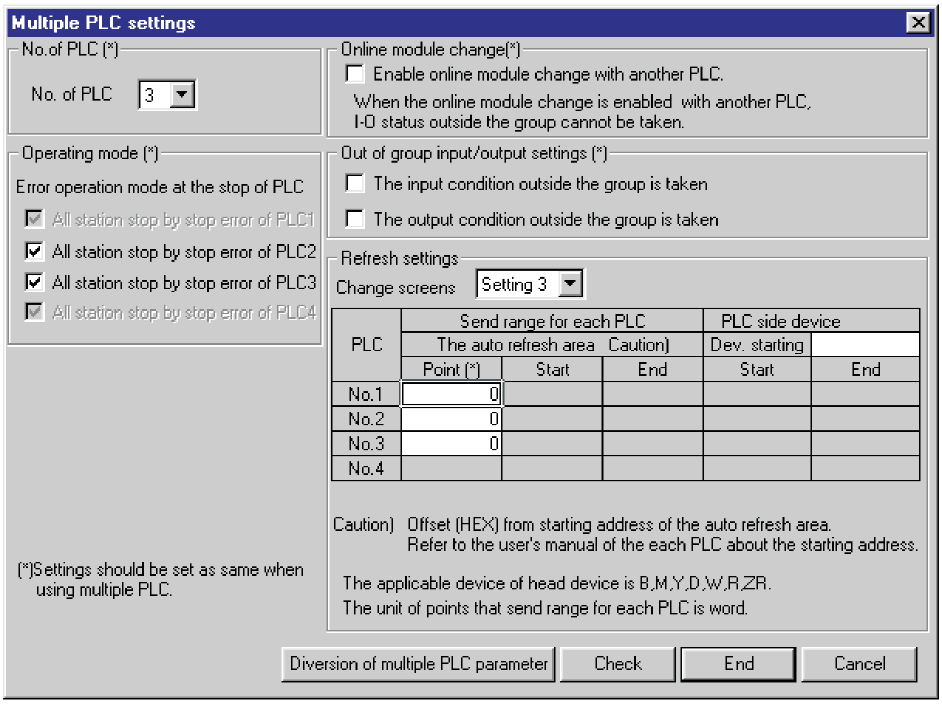
Abnormal communication remote: Enter the T, C, D, W, or R device name.

Circuit error check: Select either line error check test memo, OFF data or hold data.

STOP time operation: Specifies the operating status of the MELSECNET/MINI when the PLC was stopped:
RESUME: MELSECNET/MINI continues operation
PAUSE: MELSECNET/MINI suspends operation
Valid only for QnA.

5.14 Multiple PLC settings (Q series only, except Q00(J))

Up to four Q series CPUs can be combined on one base unit to increase the overall PLC performance by sharing different controller tasks to different CPUs.


Multiple PLC settings

No.of PLC (*)

No. of PLC 3

Online module change(*)

□ Enable online module change with another PLC.

When the online module change is enabled with another PLC, I-O status outside the group cannot be taken.

Operating mode (*)

Error operation mode at the stop of PLC

☑ All station stop by stop error of PLC1

☑ All station stop by stop error of PLC2

☑ All station stop by stop error of PLC3

☑ All station stop by stop error of PLC4

Out of group input/output settings (*)

□ The input condition outside the group is taken

□ The output condition outside the group is taken

Refresh settings

Change screens Setting 3

PLC	Send range for each PLC			PLC side device	
	The auto refresh area Caution)			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	0				
No.2	0				
No.3	0				
No.4					

Caution) Offset (HEX) from starting address of the auto refresh area.
Refer to the user's manual of the each PLC about the starting address.

The applicable device of head device is B,M,Y,D,W,R,ZR.
The unit of points that send range for each PLC is word.

(*)Settings should be set as same when using multiple PLC.

Fig. 5-57: *Multiple PLC settings*

No. of PLC
Clicking the **Default** button in I/O assignment keeps the multiple PLC setting unchanged. When the set number of PLCs is one, a multiple CPU system configuration is not set.

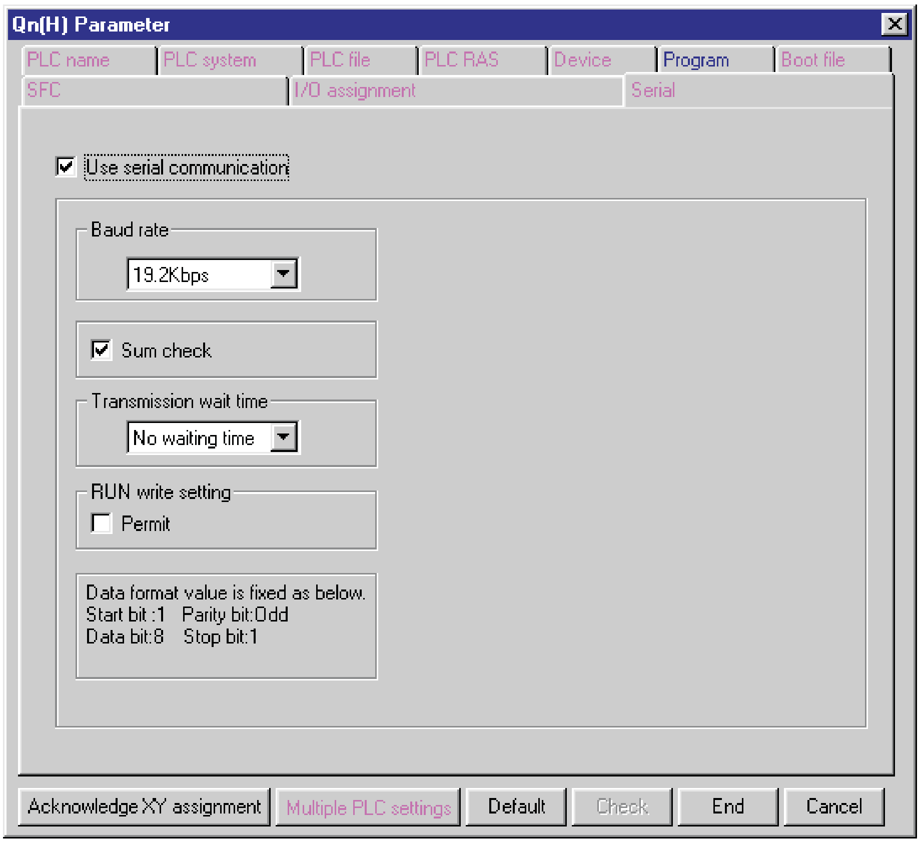
Out of group input/output settings
When the modules are not grouped, they are all put under the control PLC of the PLC No. 1. **OPERATION ERROR** appears if access is made to any module outside the groups.

Refresh settings
The maximum input range of each module is 0 to 2048 points.
Set the total number of all modules to within 4096 points.

Diversion of multiple PLC parameter button
Specify the project to be diverted from the **Open Project** screen. Setting the source project shows the multiple PLC parameter diversion dialogue box. Confirm the message and execute.

5.15 Serial (Q00 and Q01 only)

The dialogue **Internal serial communication** is used to enable, disable, and setup the communications with the PLC as Computer Link Module via the serial port. The settings are stored in the project database.



Qn(H) Parameter

☑ Use serial communication

Baud rate: 19.2Kbps

☑ Sum check

Transmission wait time: No waiting time

RUN write setting: □ Permit

Data format value is fixed as below.
Start bit :1 Parity bit:Odd
Data bit:8 Stop bit:1

Fig. 5-58: Internal serial communication settings for computer link connections

Use serial communication: Enables or disables the use of the serial communication settings for Computer Link communications. Default: Disabled.

Baud Rate: Selects a baud rate. Default: 19200. Setting range: 9600 – 115200.

Sum Check: Enables or disables a sum check for the transmission. Default: Enabled.

Transmission wait time: Selects a wait time for the transmission.
Default: 0. Setting range: 0 – 150.

RUN write setting: Enables or disables online changes during the transmission.
Default: Disabled.

5.16 CPU Type Selection

The CPU type can be changed by selecting the command **Change PLC Type** in the **Project** menu or by clicking on the icon.

In the dialogbox **Change PLC Type** you can select a PLC series and a PLC type. To change the CPU type, open the list box, select the new CPU type and confirm with OK.

NOTE

When you change the CPU GX IEC Developer deletes the old libraries and regenerates the Standard and Manufacturer libraries for the new PLC type. This process can take a while as all the instructions for the new CPU must be integrated in the libraries.

The following table lists all supported CPU types.

PLC Series	CPU Type
AnN	A1N, A2C, A2N, A3N
AnS	A1S, A1S-S1, A2S, A2S-S1
AnSH	A1SH / A1SJH, A2SH, A2SH-S1
AnA / AnU	A2A, A2A-S1, A3A, A2U, A2U-S1, A3U
A2AS / A2US	A2AS, A2AS-S1, A2US, A2US-S1, A2AS-S30, A2US-S30, A2AS-S60, A2US-S60, A2USH-S1
FX	FX / FX2, FX0 / FX0S, FX0N, FX1S, FX1N, FX2N / FX2NC, FXU / FX2U / FX2C, FX3U
QnA	Q2A(H), Q2AS(H), Q2A(H)-S1, Q2AS(H)-S1, Q3A, Q4A, Q4AR (from version 0012B)
Q	Q00, Q00J, Q01, Q02(H), Q06H, Q12H, Q25H, Q25SS (SX Controller), Q Remote I/O
QnPH	Q12PH, Q25PH
QnPRH	Q12PRH, Q25PRH

Tab. 5-6: *Supported CPU-types*

6 Programming

6.1 The Structure of GX IEC Developer Programs

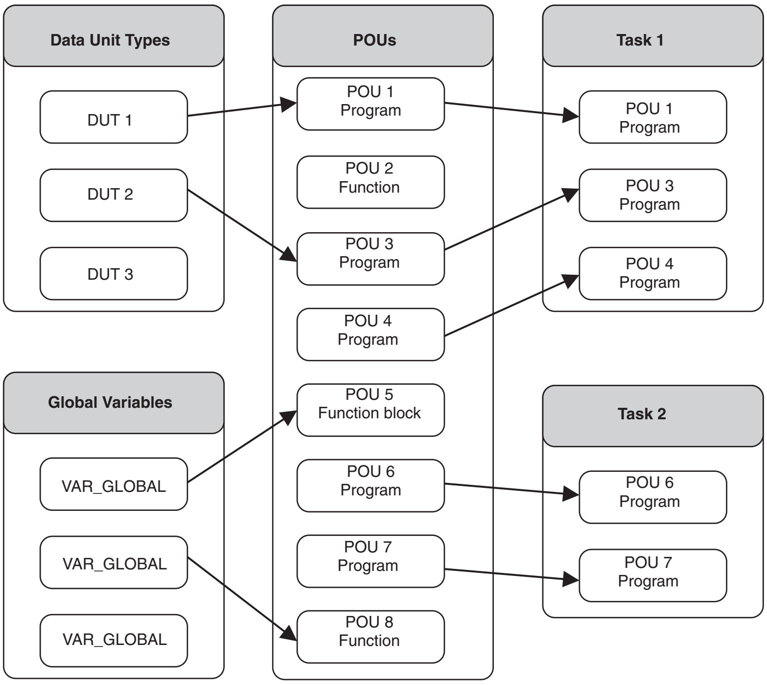


Fig. 6-1: *PLC program structure*

6.2 Tasks

The Task Pool is the ‘control centre’ for the execution of your programs. You must define at least one task for the project in the Task Pool. In turn, each task contains one or more PRG (program) type POUs. Each POU can only be defined once in any one task in the Task Pool. The tasks in the Task Pool are polled cyclically.

NOTE You cannot directly manage either function blocks or their instances in tasks.

The execution of tasks is controlled by the following three parameters:

- Event
- Interval
- Priority

All tasks are either event-triggered or interval-triggered.

NOTE In order to address the SUB memory range with the A3 CPUs you must create a special task called **MELSEC_SUB** or use the prefix "SUB_" (not case sensitive).

6.2.1 Event-triggered Tasks

Events can have the following Boolean values:

- **TRUE/FALSE**: Tasks with the event value TRUE are always executed. The TRUE or FALSE parameter is automatically converted to UPPER CASE letters!
- **Interrupts**: Interrupt-triggered execution control is only possible with input modules that support this function. The task will be executed when the interrupt polled is set. The interrupt number (e.g.: `I20`) is entered after **Event** in the task attributes in the **Task Information** dialogue box.

Series	A	FX	Q	QnA
Usable interrupts	I0–I31	I000–I899	I0–I255	I0–I31

Tab. 6-1: *Interrupt ranges for each CPU series*

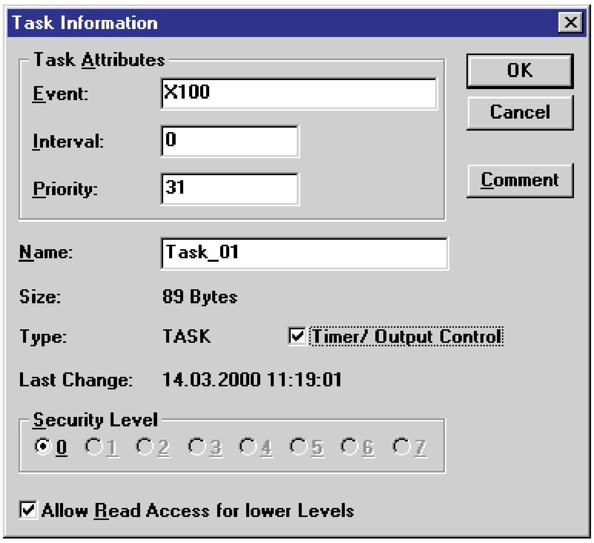
NOTE Remember that the EI instruction (Enable Interrupt) must be programmed in a MELSEC network or the function EI_M must be used within the project when using interrupt-triggered tasks.
POUs assigned to interrupt triggered tasks must not be programmed in SFC language.

- **Relays**: The task is executed when the corresponding relay is set (e.g.: `M100`).
- **I/Os**: The task is executed if the status conditions of the corresponding inputs or outputs are satisfied (e.g.: `X0`).
- **Global variables**: All global variables of data type BOOL could be used to control a task. Elements of arrays or components of data unit variables could not be used.

NOTE For every event or interval-triggered task one system label is allocated (→ page 2-39).

6.2.2 Event-triggered tasks with Timer/Output Control (MC-MCR execution)

If you enter a Boolean device as an event or a time interval you can then select the **Timer / Output Control** option.



Task Information

Task Attributes

Event: X100

Interval: 0

Priority: 31

Name: Task_01

Size: 89 Bytes

Type: TASK ☑ Timer/ Output Control

Last Change: 14.03.2000 11:19:01

Security Level

● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access for lower Levels

Fig. 6-2: *Task Information*

The following table explains how this option influences the processing of the task.

	Timer / Output Control not activated	**Timer / Output Control activated**
Execution mode	Normal execution	MC-MCR execution
Task event	X10	X100
Execution sequence	LDI X10 CJ End of Task POU Code End of Task	LDI X100 ANI MC Relay CJ End of Task LD X100 MC N0 MC Relay POU Code MCR N0 End of Task
Behaviour when event is active	If X10 is set the POU associated with the task is executed. If X10 is not set the PLC program jumps to the end of the task.	If X100 is set execution of the program sequence between the MC and MCR instructions begins.
Behaviour when event is not active	The program jumps to the end of the task but some of the device statuses are preserved: - Timers: continue to run (A, FX) or with Timer scheduler for Q, QnA - Timers: stop running for Q, QnA without Timer scheduler - Retentive/accumulator timers and counters: status preserved - Devices in OUT instructions: status preserved	If X100 is not set the program sequence between the MC and MCR instructions is executed once with the MC relay set to FALSE. If the MC relay is also not set the PLC program jumps to the end of the task. The statuses of the devices are reset: - Timer: Count value reset to 0, contacts remain off - Retentive timers and counters: count value and input contact status preserved, output contact is reset - Devices in OUT instructions: outputs are reset - Devices in SET, RST, SFT instructions: current status preserved

Tab. 6-2: *Effect of Timer / Output Control option*

6.2.3 Interval-triggered Tasks

The intervals are entered as time constants. If a task is to be executed at intervals you must enter FALSE for the **Event** parameter in the **Task Information** dialogue box.

Syntax:	`T#[Time]`
[Time]	This is the value for the interval duration. Example: 12h30m, 5s, 100ms

You can also use a global variable of data type TIME as interval of a task. Elements of arrays or components of data unit variables could not be used.

WARNING:
You must consider the program cycle period when defining the interval duration! Defining intervals shorter than the program cycle period can lead to unpredictable results. When in doubt, always calculate the cycle period of the task using the instruction execution times listed in the controller manual.

6.2.4 Priority-controlled Tasks

The task with the lowest priority value has priority and it will call its associated programs first. You can assign priority values between 0 and 31.

How to define task execution attributes

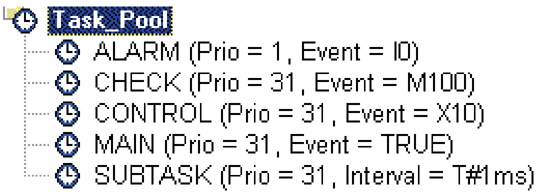
① In the Project Navigator window, select the name of the task you wish to edit.

② Press the key combination Alt Enter to open the **Task Information** dialogue box.

③ Enter your values for the task attributes.

④ Confirm with **OK**.

The attribute settings are shown in parentheses behind the name of the task in the Project Navigator window.



```
Task_Pool
  ALARM (Prio = 1, Event = I0)
  CHECK (Prio = 31, Event = M100)
  CONTROL (Prio = 31, Event = X10)
  MAIN (Prio = 31, Event = TRUE)
  SUBTASK (Prio = 31, Interval = T#1ms)
```

Fig. 6-3:
Tasks in order of their execution priority:
– Event-triggered via Interrupt I0
– Event-triggered via M100
– Event-triggered via X10
– Always executed
– Interval-triggered, every 10 ms

NOTE The procedure for creating a new task is described in chapter 4.

6.2.5 Special task MELSEC_FIRST

If you name a task *MELSEC_FIRST* (not case sensitive), it is handled with its assigned POUs in a special way:

- The assigned POUs will be the first ones in the MAIN sequence program.
- All internal code parts from the code generator which must be positioned at the beginning of the MAIN sequence program will be positioned behind this task.

The following checks are performed during the code generation and will cause an error on occurrence:

- Interrupt events are not possible for this task.
- Because SFC POUs must be initialised in the first scan before they are used, they could not be assigned to this task.
- The SFC control functions SFC_CTRL and SFC_PAUSE use a semaphore which is reset at the beginning of the MAIN sequence (now after MELSEC_FIRST). Therefore this functions could not be used directly by programs assigned to MELSEC_FIRST or within their calltree.

The task priority is ignored for MELSEC_FIRST – you are informed of this circumstance by a warning. All errors or warnings are displayed in the error listbox.

6.3 Variables

6.3.1 Global and Local Variables

Variables are similar to operands. They contain the values of inputs, outputs or internal memory locations in the PLC. There are two types of variables:

- Global Variables
- Local Variables

Global Variables

Global variables are defined for the entire project. They are accessible from any POU body, action or transition, and they make it possible to exchange data between all the sub-programs that go to make up the project.

Local Variables

Local variables (VAR) are only accessible within the one specific program organisation unit (POU) in which they are declared. Thus, they cannot be used for exchanging data between POUs.

NOTE

> When there is a local variable declared in a POU with the same name as a global variable, the global variable is hidden, i.e. it is not possible to read or write the value of that global variable – always the local variable is referenced instead. (See also section 2.7.1.)

6.3.2 Variable declarations

Each variable declaration has the following elements:

- Class,
- Identifier,
- Data type,
- Initial value (automatically),
- Comment (optional),
- Remark (optional).

6.3.3 Class

The class keyword assigns the variable a specific property that defines how it is to be used in the project.

Class	Use in POUs:			Meaning
	PRG	FUN	FB	
VAR	X	X	X	Variable that is only used within the POU
VAR_CONSTANT	X	X	X	Local variable with unchangeable initial value used within the POU
VAR_INPUT	—	X	X	Variable passed from outside that cannot be altered within the POU
VAR_OUTPUT	—	—	X	Variable passed (output) by the POU
VAR_IN_OUT	—	—	X	Local variable passed from outside and passes (output) by the POU, can be altered within the POU
VAR_GLOBAL	X	—	X	Global variable declared in the Global Variable List
VAR_GLOBAL_CONSTANT	X	—	X	Global variable with unchangeable initial value declared in the Global Variable List

Tab. 6-3: *Use of classes in POUs*

6.3.4 Identifiers and Absolute Addresses

Each variable is given a symbolic address, i.e. a name. This is referred to as the **identifier**; it consists of a string of alphanumeric characters and underline characters. The identifier must always begin with a letter or an underline character. Spaces and mathematical operator characters (e.g. +, -, *) are not permitted.

Examples of identifiers: FAULT
ZEROSIG
LIM_SW_5

When global variables are declared **absolute addresses** can be assigned that reference the memory location of the variable in the CPU or a physical input or output.

When local variables are declared in the header of the POU they are automatically assigned a suitable memory location in the CPU.

You can use either the IEC syntax (IEC-Addr.) or the MITSUBISHI syntax (MIT-Addr.) to assign the absolute addresses. Two address columns are available.

As soon as you have entered an address in one of these columns, the other address also appears. You can enter either of the two address formats in both columns. If, for instance, you enter a MITSUBISHI address in the IEC column, GX IEC Developer identifies it immediately, places it in the correct column and produces the matching IEC address in the other column.

Use upper case letters only and no spaces or mathematical operator characters
(e.g. +, -, *) in addresses.

When you define global variables you must specify the hardware addresses for each individual variable. The address declarations are shown in the table below.

MITSUBISHI Address	IEC Address	Size ⑤	Range	Example
X	%I	X		X0 ↔ %IX0 X1F ↔ %IX31 ③ X37 ↔ %IX31 ④ X7FFF ↔ %IX32767 ⑤
Y	%Q	X		Y0 ↔ %QX0 Y1F ↔ %QX31 ③ Y37 ↔ %QX31 ④ Y7FFF ↔ %QX32767 ⑤
D	%M	W, D	0	D450 ↔ %MW0.450 D501+ D502 ↔ %MD0.501
	%M	X ⑤	0	D0.5 ↔ %MX0.0.5 D100.F ↔ %MX0.100.15
W ③⑤	%M	W, D	1	W35F ↔ %MW1.863 W301+ W302 ↔ %MD1.769
	%M	X ⑤	1	W35F.5 ↔ %MX1.863.5 W35F.F ↔ %MX1.863.15
R ③⑤	%M	W, D	2	R450 ↔ %MW2.450 R501+ R502 ↔ %MD2.501
	%M	X 5	2	R450.5 ↔ %MX2.450.5 R450.F ↔ %MX2.450.15
M / L / S ①②	%M	X	0	M0 ↔ %MX0.0 M1024 ↔ %MX0.1024
B ③⑤	%M	X	1	B3F ↔ %MX1.63
S ②	%M	X	2	S10 ↔ %MX2.10
T Contact ⑦	%M	X	3	TS10 ↔ %MX3.10
T Value	%M	W	3	TN40 ↔ %MW3.40
C Contact	%M	X	4	CS10 ↔ %MX4.10
C Value	%M	W, D	4	CN10 ↔ %MW4.10 CN210 ↔ %MD4.210 ④
T Coil	%M	X	5	TC20 ↔ %MX5.20
C Coil	%M	X	6	CC20 ↔ %MX6.20
A ③	%M	W,D	5	A0 ↔ %MW5.0 A0+ A1 ↔ %MD5.0
V ②③	%M	W,D	6	V1 ↔ %MW6.1 V2+ Z2 ↔ %MD6.2
Z	%M	W	7	Z2 ↔ %MW7.2
	%M	X ⑤	7	Z2.5 ↔ %MX7.2.5 Z2.F ↔ %MX7.2.15
F	%M	X	7	F20 ↔ %MX7.20
L ⑤	%M	X	8	L30 ↔ %MX8.30
V ⑤	%M	X	9	V47 ↔ %MX9.47
N				Not permitted in IEC
P				Cf. network label
I				Only allowed in event-triggered tasks
K				K0 ↔ 0

Tab. 6-4: *Address declarations for variables (1)*

MITSUBISHI Address	IEC Address	Size ⑤	Range	Example		
H	16#			HFF	↔	16#FF
SM ⑤	%M	X	10	SM1024	↔	%MX10.1024
SD ⑤	%M	W, D	10	SD450	↔	%MW10.450
	%M	X	10	SD45.5 SD45.F	↔ ↔	%MX10.45.5 %MX10.45.15
SB ⑤	%M	X	11	SB3F	↔	%MX11.63
SW ⑤	%M	W, D	11	SW35F	↔	%MW11.863
	%M	X	11	SW5F.5 SW5F.F	↔ ↔	%MX11.95.5 %MX11.95.15
ZR ⑤	%M	W, D	12	ZR327 ZR40501 + ZR40502	↔ ↔	%MW12.327 %MD12.40501
	%M	X	12	ZR27.5 ZR27.F	↔ ↔	%MX12.27.5 %MX12.27.15
ST Contact ⑤⑦	%M	X	13	STS10	↔	%MX8.10
ST Value ⑤	%M	W	13	STN40	↔	%MW8.40
U \G ⑤	%M	W, D	14	U1\G8 U2\G10 + U2\G11	↔ ↔	%MW14.1.8 %MD14.2.10
	%M	X	14	U1\G8.5 U1\G8.F	↔ ↔	%MX14.1.8.5 %MX14.1.8.15
ST Coil ⑤	%M	X	15	STC20	↔	%MX9.20
J \X ⑤	%I	X	16	J1\X1F	↔	%IX16.1.31
J \Y ⑤	%Q	X	16	J1\Y1F	↔	%QX16.1.31
J \B ⑤	%M	X	16. .1	J2\B1F	↔	%MX16.2.1.31
J \SB ⑤	%M	X	16. .1	J2\SB1F	↔	%MX16.2.11.31
J \W ⑤	%M	W, D	16. .1	J1\W35F J1\W301 + J1\W302	↔ ↔	%MW16.1.1.863 %MD16.1.1.769
	%M	X	16. .1	J1\W5F.5 J1\W5F.F	↔ ↔	%MX16.1.1.95.5%MX16.1.1.95.15
J \SW ⑤	%M	W, D	16. .11	J1\SW35F J1\SW301 + J1\SW302	↔ ↔	%MW16.1.11.863 %MD16.1.11.769
	%M	X	16. .11	J1\SW5F.5 J1\SW5F.F	↔ ↔	%MX16.1.11.95.5 %MX16.1.11.95.15
BL ⑤	%M	W	17	BL319	↔	%MW17.319
TR ⑤	%M	X	18	TR511	↔	%MX18.511
DX ⑤	%I	X	1	DX1F	↔	%IX1.31
DY ⑤	%Q	X	1	DY1F	↔	%QX1.31
FX ⑤				Not needed in IEC corresponds to VAR_INPUT		
FY ⑤				Not needed in IEC corresponds to VAR_OUTPUT or function result		
FD ⑤				Not needed in IEC corresponds to VAR_OUTPUT, VAR_INOUT or function result		
J ⑤				Not permitted in IEC		
U ⑤				Not permitted in IEC		

Tab. 6-4: *Address declarations for variables (2)*

Table notes

① M, L and S devices are not compatible to old MEDOC projects. The system does not check whether latch relay devices L (e.g. L100) are within the latch range of the relay devices M. Memory-buffered M relays are not displayed as L latch relays.
L and S relays are handled differently in the Q/QnA series.

② In the A series CPUs the step status devices S are treated in the same way as M relay devices. For example, this means that M100 and S100 are identical.
In the F and Q/QnA series CPUs the S devices are treated as step status devices
The devices S can not be used by the user program.

③ Applies for A series controllers only

④ Applies for F series controllers only

⑤ Applies for Q/QnA series controllers only

⑥ The size prefix is optional, i.e. the entries %IX10 and %I10 are effectively identical.
Prefixes:

X	Bit device
W	Word device
D	Double word device

⑦ The following types of MITSUBISHI devices are used for timers and counters:

STS, TS, CS	Contact (Timer, Counter)
STC, TC, CC	Coil (Timer, Counter)
STN, TN, CN, T, C	Value (Timer, Counter)

6.3.5 Initial Value

The initial values are set automatically by the system and cannot be changed by the user.

6.3.6 Comment

You can add a comment up to 64 k characters long for each variable.

6.3.7 Remark

You can add additional user information.

6.3.8 Data Types

The data type of a variable defines the number of bits it contains, how they are processed and the variable's value range. The following data types are available.

Data type		Value range	Size
BOOL	Boolean	0 (FALSE), 1 (TRUE)	1 Bit
INT	Integer	-32768 to 32 767	16 Bit
DINT	Double integer	-2147483648 to 2147483647	32 Bit
WORD	Bit string 16	0 to 65535	16 Bit
DWORD	Bit string 32	0 to 4294967295	32 Bit
REAL	Floating-point value	3.4 E +/- 38 (7 digits)	32 Bit
TIME	Time value	T#-24d-0h31m23s648.00ms to T#24d20h31m23s647.00ms	32 Bit
STRING ①	Character string	max. 50 characters	

Tab. 6-5: Data types

① You can set the default value for the number of characters in string variables with the option **Extras - Options - General - Default String Length (characters)**.

NOTES

The data type STRING with a maximum length of 50 characters is only available for MELSEC Q/QnA series.

When using the data type STRING with the A series or the FX series (only FX2, FX2C, and FX2N(C)) a maximum length of 8 characters is alowed.
With these CPU series the string cannot be manipulated using dedicated instructions.

Data unit types (DUT) are structured, derived data types containing a collection of variables which can be of different data types (→ page 6-66).

Data types hierarchy ANY

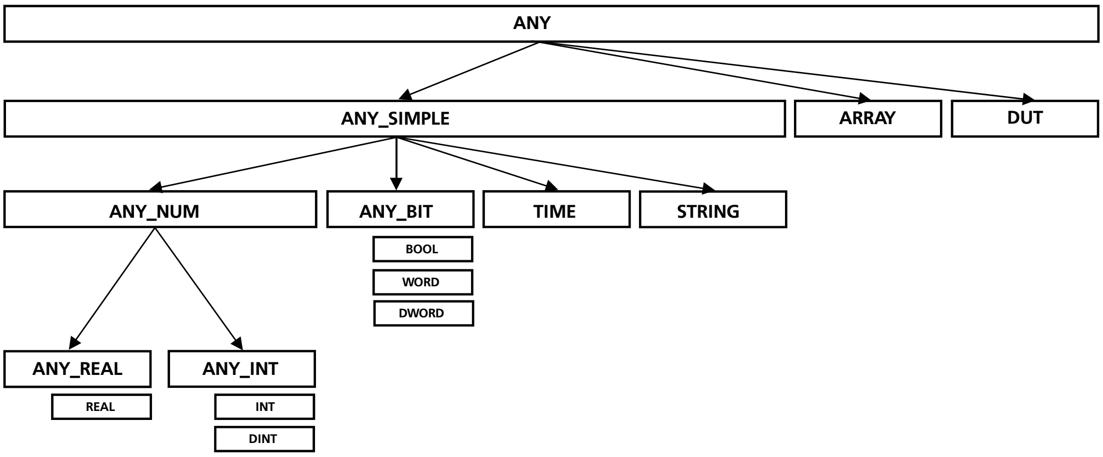


Fig. 6-4: *Data types hierarchy ANY*

Data types hierarchy ANY16 and ANY32

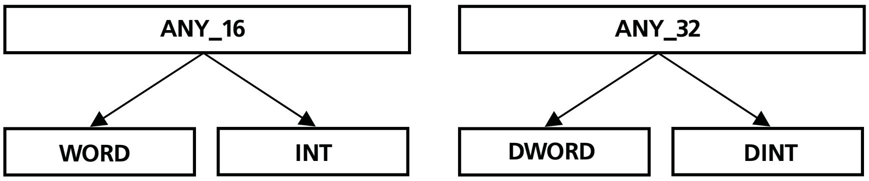


Fig. 6-5:
Data types hierarchy ANY16 and ANY32

6.3.9 Address Batch Replace

The Address Batch Replace function is available for the global variable list of the project and in the global variable lists of the included libraries.

This function replaces an existing address range by a new address range.

In the **Tools** menu select **Address Batch Replace** to open the following dialogue.

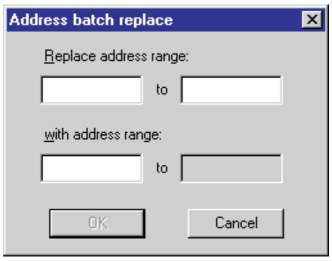
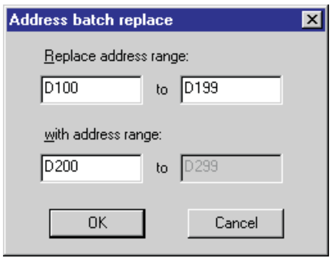


Fig. 6-6:
Address batch replace

The address range can be entered in either IEC or Mitsubishi format. Both formats must not be combined.

The greyed last entry field is calculated automatically.

Example ▽



Replace address range: D100 to D199

with address range: D200 to D299

Fig. 6-7:
Address batch replace settings

The variables with the addresses D100 – D199 are replaced by the addresses D200 – D299. A variable with the address D100.0 will be changed to D200.0.

△

NOTE The Undo function is not supported for the Address Batch Replace.

6.3.10 Auto Update

Global and local variables can be updated automatically by the Auto Update function. This function can be activated via the menu **Extras - Options - Editing** (refer to 2.7.4).

The following changes of a variable declaration are updated: Class, Identifier, Address, Type, Initial. All changes made in the GVL will also be executed in the POUs. When you change a POU header, its body will also be changed automatically. The variables are updated after the declaration editor changes have been saved.

Via the menu **Extras - Update Variables** all references of the **selected** local variables are updated, if the name of the local variable has been changed. For every reference to be updated a message box appears where you can decide to update the reference or not.

6.4 Programming Languages

6.4.1 Networks

Programming is done in ‘networks’ in all programming languages except Sequential Function Chart (SFC) and Structured Text (ST). You can define multiple networks, each of which is identified with a network label. It is then possible to switch from one network to another within the PLC program by using jump instructions in the program code. For a detailed description of this please refer to chapter 3.

Network labels can have a maximum of eight characters and must always be terminated with a colon (:). You can also assign a title to each network. These network titles can be descriptive names up to 22 characters long, and using them makes handling large projects with many networks a great deal easier.

NOTE

Remember that every network label uses one system label and that the total number of system labels is limited to a maximum of 256. Since system labels are also needed for other elements in GX IEC Developer (→ page 2-39), you should only define labels to networks to which you actually intend to perform a jump.

The Network List

The networks in your program organisation units are managed in the Network List, which you can only open when the body window of the corresponding POU is open. The list window contains all the POUs networks with their labels and titles. In the window you can add more networks and edit existing ones. A search function (**Find**) is provided for finding specific networks quickly (→ Chapter 3).

WARNING:
For all practical purposes it is not possible to change the programming language of a POU once you have selected it. The system will allow you to make the switch, but if you do this you will lose all the code in the body. You should thus only switch the programming language of bodies that do not contain any code!

6.4.2 Text Editors

Instruction List (IL)

Programming in Instruction List is performed with a simple text editor.

An Instruction List program consists of a sequence of controller instructions. Each controller instruction must contain an operator or a function or function block and one or more operands/variables. Each controller instruction must begin in a new line. You can also add optional comments to each instruction. All the programming elements in the individual instructions must be separated by tab stops.

MELSEC Instruction List

The user interface for programming in MELSEC IL is identical to that for IEC IL. However, only the ‘Pure’ MELSEC instruction set is supported (→ Appendix E); you cannot write IEC standard programs in the MELSEC IL editor. The functionality of the MELSEC IL language corresponds to that of the MELSEC MEDOC programming and documentation system. This enables you to easily import programs developed with MELSEC MEDOC and integrate them into your GX IEC Developer programs
(→ Chapter 8).

IEC Instruction List

The user interface for programming in IEC IL is identical to that for MELSEC IL. There are just a few differences in programming:

- **MELSEC Networks in IEC IL**
  You can also integrate MELSEC networks in IEC IL programs, thus effectively giving you access to the ‘Pure’ MELSEC programming instructions.
- **The Accumulator**
  The accumulator is a result management system often used in high-level languages. The result of every operation is stored in the accumulator directly after execution of the instruction. The accumulator thus always contains the operation result of the last instruction executed. You do not need to program any input conditions (execution conditions) for the operations; execution always depends on the contents of the accumulator.

NOTE

The difference between programming in the MELSEC editor (with execution condition) and in the IEC editor (with accumulator) is described in the Beginner’s Manual (→ Chapter 3).

Structured Text (ST)

ST is a text-oriented editor (programming language), similar to PASCAL and supports mathematical functions and a simple creation of loops.

ST body does not contain a network list because it always consists of only one network.

ST is an editor from the IEC 61131 programming standard. The Structured Text editor is compatible to the IEC 61131-3.

All IEC 61131 (IEC 61131-3: PART3-1992) standard functions are supported.

All MELSEC instructions are supported.

6.4.3 Graphical Editors

Ladder Diagram (LD)

In Ladder Diagram you can use both IEC and MELSEC programming instructions in every network. A Ladder Diagram program consists of contacts (break and make), coils, function blocks and functions. These elements are connected with horizontal and vertical lines called interconnects. The circuits always begin at the power bar on the left.

Functions and function blocks are displayed as graphical blocks. In addition to their input and output parameters, some blocks also have a Boolean input (EN = ENable) and a Boolean output (ENO = ENable Out). The status of the output always corresponds to that at the input.

⚠ **WARNING:**
You must normally create a separate network for each circuit in a Ladder Diagram program. However, you can also create an additional power bar parallel to the standard one provided by the editor, connecting the two at one point. This option makes it possible to perform programming in just one network where this is more convenient.

Function Block Diagram (FBD)

In Function Block Diagram language, as in Ladder Diagram language, you can use both IEC and MELSEC programming instructions in every network (→ Appendix E). All instructions are implemented using graphical blocks, which are connected with one another with horizontal and vertical connecting lines. There are no power bars (rails) in Function Block Diagram language.

Functions and function blocks are displayed as graphical blocks. In addition to their input and output parameters, some blocks also have a Boolean input (EN = ENable) and a Boolean output (ENO = ENable Out). The status of the output always corresponds to that at the input.

⚠ **WARNING:**
In Function Block Diagram language you must create a separate network for each circuit.

6.4.4 Sequential Function Chart (SFC)

Sequential Function Chart is a graphical language. Strictly speaking it is not really a genuine programming language but rather a structural tool which allows a clear representation of processes.

The program (PRG) is the only program organisation unit supported in SFC.

SFC language has two basic elements, Steps and Transitions (continue conditions). A SFC program sequence consists of a series of steps separated by transitions. Only one step can be active at any one time in such a program sequence except in parallel branches. The next step in the sequence is activated when the preceding step/steps is/are active and the transition condition is satisfied.

Steps and Actions

Several actions can be assigned to one step. An action can be a Boolean output variable or a piece of program, which can be written in any of the editors, including Sequential Function Chart itself. All actions are listed in the Action Pool in the Project Navigator window. A step to which no action is assigned is treated as a wait loop that stops execution until the following transition condition is fulfilled. A step is activated when the transition condition (continue condition) immediately preceding it is fulfilled.

Steps are represented graphically as rectangles containing a name (descriptor).

Macro Steps

Each step can be declared as a macro step, which in turn consists of a sequential structure. The only limitation on the nesting depth of such structures is the memory capacity of the controller used.

Macro steps are identified by two additional horizontal lines within the step rectangle.

Transitions

Each transition has a transition condition. Transition conditions can be programmed in the Instruction List, Ladder Diagram, Function Block Diagram, MELSEC IL and ST language. It is also possible to use a Boolean variable, direct address, or constant.

Like the steps, the transitions are also listed in the Project Navigator window.

Transitions are displayed in the editor with a small box lying on the vertical connecting line between the steps.

Initial Step

An SFC sequence always begins with an initial step. The initial step does not have to be at the beginning, however – it can also be placed at another point in the sequence.

NOTE

> The initial step is always the first step that is executed in an SFC program, even if it is not placed at the physical beginning of the sequence.

The initial step is identified by a double outline.

Branching

SFC programs support both selective and parallel branching. Selective branches divide into two or more transition conditions. Parallel branches are always inserted after transitions, and they can activate actions or jumps.

Selective branching

In a selectively-branched sequence not more than one step can be active simultaneously. If the transition conditions of two or more transitions in a selectively-branched sequence are active at the same time the execution priority is determined on the basis of the order of the sequences from left to right. This means that only the sequence that is furthest to the left will be executed. The sequences to the right of this sequence will not be executed, even if their transition conditions evaluate true.

Selective branches are displayed as single horizontal lines.

Parallel branching

Parallel-branched sequences allow the activation of multiple steps simultaneously. The steps leading off the parallel branch become active as soon as the preceding step and the transition condition are satisfied.

Several processes can be executed simultaneously in parallel branches.

Parallel branches are displayed as double horizontal lines.

pJumps

Each jump destination is always directly followed by a step. You can thus only insert a jump destination when the step which is to follow it is selected. The jump must always be programmed after a transition, as it replaces the step that would normally come in this position. This means that no further steps or transitions can follow a jump, which is thus also the final instance within its SFC sequence. The relationship between the jump and jump destination is established by using the same label (name) for both of them.

When you program a jump instruction pressing F2 displays a list of all the labels (i.e. potential jump destinations) that have already been defined. You can then simply select the appropriate label from the list and confirm to enter it.

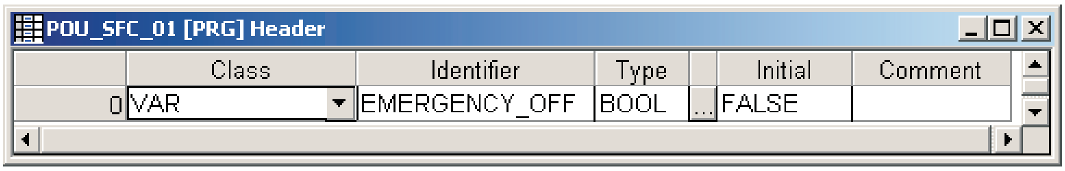
NOTE

You will find examples of the sequencing rules described here on the following pages.

EMERGENCY OFF for FX Series

For the FX series PLCs it is possible to integrate an EMERGENCY OFF switch in an SFC program. All step relays in the sequence are reset as soon as the EMERGENCY OFF switch is activated. The calculated values in the steps are not lost when this happens. The program process branches back to the initial step. This does not mean that the complete sequence is re-initialized, however – the sequence is only interrupted. You should always exercise caution when restarting processes.

The SFC reset variable (in the following example it is a local variable called EMERGENC_OFF) can be either a local or a global variable with the type BOOL or ARRAY OF BOOL. When an ARRAY OF BOOL variable is given, its first element will be used as the reset variable.

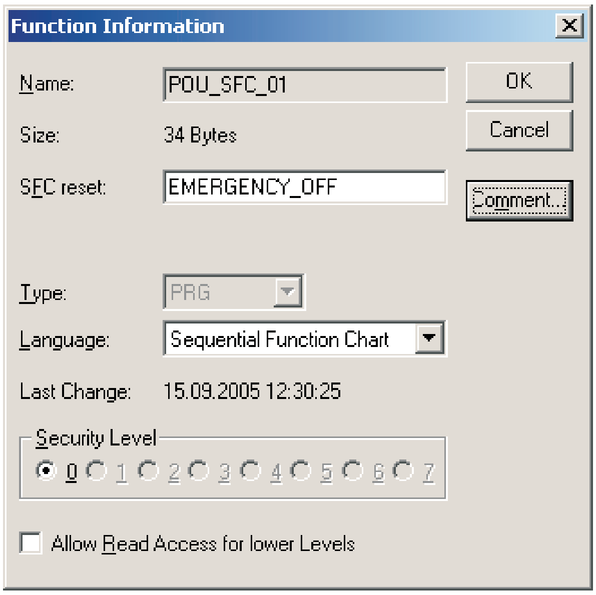


POU_SFC_01 [PRG] Header

	Class	Identifier	Type		Initial	Comment
0	VAR	EMERGENCY_OFF	BOOL	...	FALSE	

Fig. 6-8: *EMERGENCY OFF*

The assignment of the SFC reset variable to the SFC POU is introduced in the POU information dialogue. When you call **Object** → **Information** for the marked SFC POU (in this example called POU_SFC_01), the following dialogue provides the setting for the SFC reset variable. Input the identifier of the required local or global variable in the field **SFC reset**.



Function Information

Name: POU_SFC_01

Size: 34 Bytes

SFC reset: EMERGENCY_OFF

Type: PRG

Language: Sequential Function Chart

Last Change: 15.09.2005 12:30:25

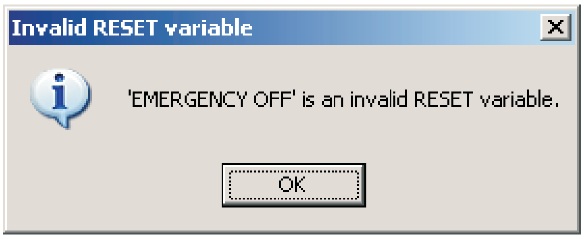
Security Level

● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☐ Allow Read Access for lower Levels

Fig. 6-9: *Entering the variable identifier for the SFC reset variable*

When pressing **OK** in this dialogue, the syntax of the entered name will be checked, but it will not be verified whether the entered name specifies an existing variable. When the POU or the project is checked (or built), and the SFC reset setting does not specify a valid variable, an error will be added to the list of errors and warnings in the **Compile/check messages** dialogue.



Invalid RESET variable

'EMERGENCY OFF' is an invalid RESET variable.

Fig. 6-10: *Warning message in case of a syntax error*

6.5 Programming in Sequential Function Chart Language

Programming in SFC language calls for systematic analysis of the control sequence before you start. The algorithm for implementing the control task should be drawn up in a systematic step-by-step procedure. SFC programs are planned on the basis of statuses and status transitions. One starts by identifying all the statuses in the process and connecting them with status transitions. The general statuses formulated in the first program development phase can then be gradually made more precise. In the final phase each status is implemented in a step and each status transition in a transition.

For the use of SFC programs you can set some CPU specific parameters (→ Chapter 5).

The PLC holds an active information table to be able to activate the active steps of the SFC POUs after a switch from Stop to Run. For this the SFC program start mode has to be set to **Resume start** (on the page **SFC** in the PLC's parameters).

When the entry **AdjustSFCActiveInfoTable** in the **[OPTIONS]** section of the MMP701.INI file (→ Section 2.6) is set to TRUE, the active step information table is updated during download for QnA and Q PLCs. If it is set to FALSE in the MMP701.INI file the active step information table is not updated and the PLC will raise an error when the offset of the active steps have been changed or a new SFC POU has been added or one has been deleted.

If **AdjustSFCActiveInfoTable** is set to TRUE the following updates are done during download:

- The offsets of the active steps are adjusted when the SFC program has been changed.
- The information of a SFC block, which corresponds to a SFC POU, is deleted from the active step information table of the PLC when the SFC POU has been deleted.

If a SFC POU has been added, the active step information table is expanded. The initial step of the new SFC POU is automatically activated.

NOTES

In the Beginner’s Manual (→ Chapter 6) you will learn step by step how to create an SFC program and to use the tools in the SFC editor.

The Q25SS (SX Controller) CPUs do not support the SFC language.

6.5.1 Sequential Function Chart Language Program Structure

The body of a newly-generated SFC POU always consists of the initial step, a transition and the final step.

NOTE

When you are first trying out the SFC editor it's a good idea to keep an eye on the active tools in the Toolbar. Only the elements whose tool icons are active can be inserted at the selected position. Create a test program to experiment with all the available ways of using SFC language elements.

The following basic rules of thumb will help you with writing your first SFC programs:

- **Each step must always be followed by a transition, each transition by a step**.
The only exceptions to this rule are exit and entry steps (jumps and jump destinations). Both of them 'replace' steps, and they must thus always be programmed directly after transitions. Entry steps (jump destinations) must logically always be followed by a step that can be executed. An exit step (jump) must always be programmed after a transition that is not followed by a step.
- **Selective branches can only be inserted before transitions**.
Since a selective branch polls a variety of conditions before executing a step these conditions must be placed in transitions directly after the branch.
- **Parallel branches can only be inserted after transitions.**
This is because a parallel branch has the effect that two or more steps are performed simultaneously after a common transition.
- **You must assign a transition condition to every transition in the program.**
A transition condition is a program, Boolean variable, address or constant that defines the conditions under which the transition to the next step is to be enabled.
- **You can assign an action to every step in the program.**
An action can be either a program or a Boolean output variable. Please note that an action's program is not a POU, however!

6.5.2 Sequence rules and examples

Simple sequences

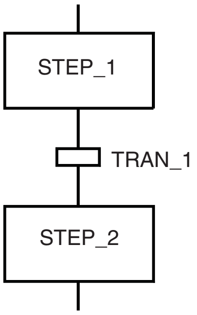


Fig. 6-11:
A simple sequence consists of a series of alternating ‘Step – Transition’ pairs.

Example:
The transition from STEP_1 to STEP_2 is only executed if STEP_1 has been executed and transition condition TRAN_1 evaluates true.

Divergence into selective sequences

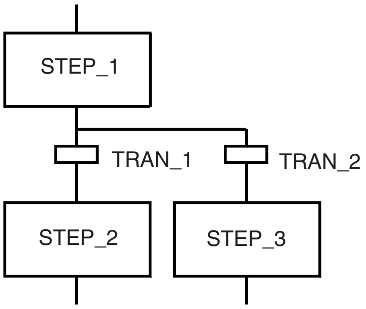


Fig. 6-12:
Branching to a selection of alternative sequences (divergence) is programmed by programming one transition beneath the horizontal line for each alternative sequence to be included in the selection. Note that if several transition conditions evaluate true at the same time only the sequence that is furthest to the left will actually be executed! Example:
The transition from STEP_1 to STEP_2 only takes place when STEP_1 has been executed and transition condition TRAN_1 evaluates true. The transition from STEP_1 to STEP_3 only takes place when STEP_1 has been executed, TRAN_2 evaluates true and TRAN_1 evaluates false.

Convergence from selective sequences

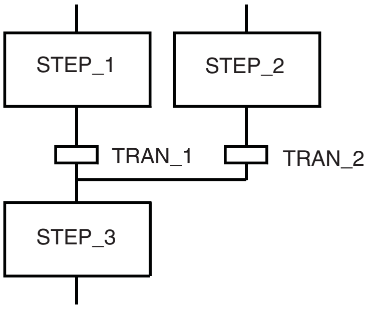


Fig. 6-13:
The end of a selectively branched sequence is converged again by programming one transition above the horizontal line for each branch sequence to be terminated.
Example:
The transition from STEP_1 to STEP_3 only takes place when STEP_1 has been executed and transition condition TRAN_1 evaluates true. The transition from STEP_2 to STEP_3 only takes place when STEP_2 has been executed and TRAN_2 evaluates true.

Divergence into parallel sequences

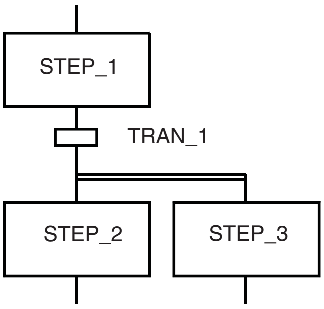


Fig. 6-14:
You can only program one transition directly above the double horizontal synchronisation line.
Example:
The transition from STEP_1 to STEP_2 and STEP_3 only takes place when STEP_1 has been executed and transition condition TRAN_1 evaluates true. Following simultaneous activation of STEP_2 and STEP_3 the two sequences are executed in parallel, independently of one another.

Convergence of parallel sequences

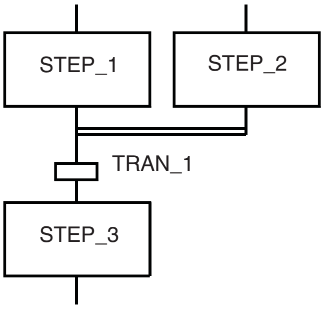


Fig. 6-15:
You can only program one common transition beneath the double horizontal synchronisation line.

Example:
The transition from STEP_1 and STEP_2 to STEP_3 only takes place when both STEP_1 and STEP_2 have been executed and are connected by the double horizontal line above the transition. The transition condition TRAN_1 must also evaluate true.

Empty branches

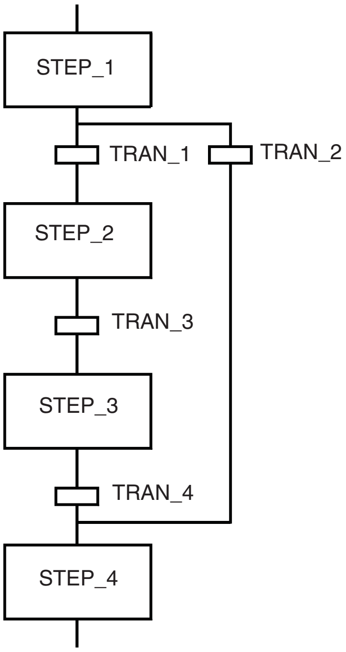


Fig. 6-16:
This is a special form of selective branching in which one or more alternative branches contain no steps.

Example:
If and only if TRAN_1 is false and TRAN_2 is true the transition runs from STEP_1 to STEP_4, skipping STEP_2 and STEP_3.

Exit and Entry Steps (Jumps)

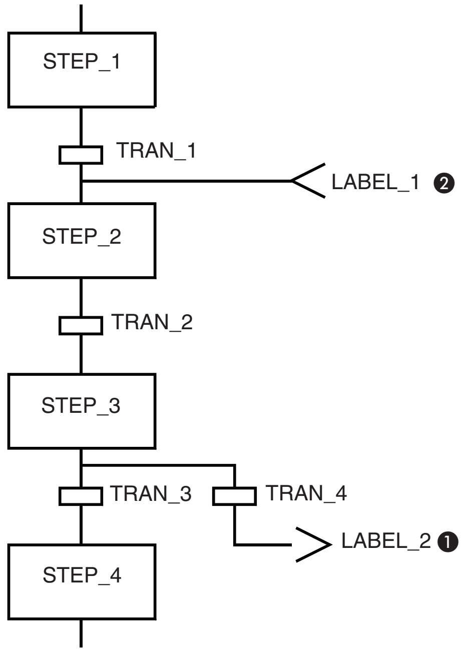


Fig. 6-17:
A transition condition can be followed by an exit step ❶ instead of a step. The label of the exit step references the corresponding entry step ❷, which can be placed anywhere in a sequence directly before an executable step. The exit and entry steps must have the same designators (names). This structure can be used to program loop constructs.

Example:
The sequence is processed up to STEP_3. Then, if STEP_3 has been executed, TRAN_3 evaluates false and TRAN_4 evaluates true, execution performs a jump back to STEP_2 via LABEL_1.

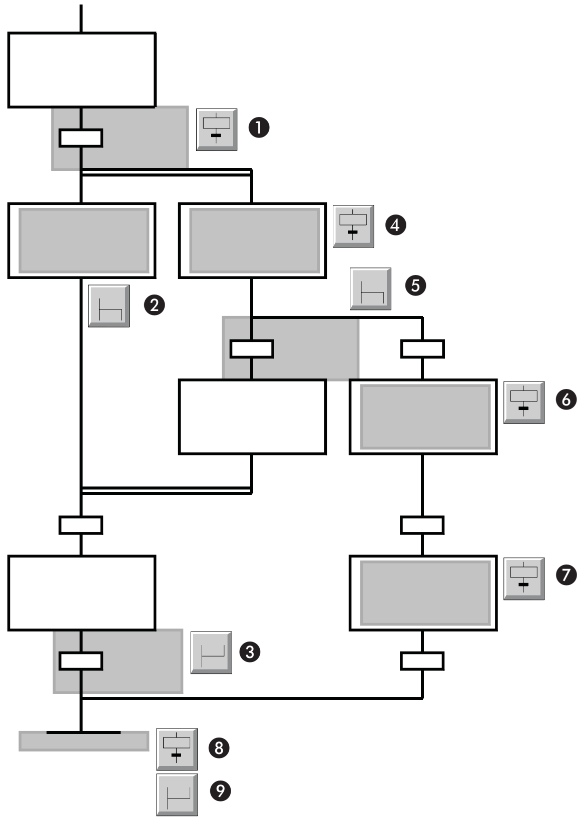


Fig. 6-18:
Creating a Sample Program

Operations and Order of Procedure:

❶ *Step and Transition*

❷ *Parallel Branch (Divergence)*

❸ *Convergence*

❹ *Step and Transition*

❺ *Selective Branch (Divergence)*

❻ *Step and Transition*

❼ *Step and Transition*

❽ *Step and Transition*

❾ *Convergence*

The illustration shows the creation of an example program in nine steps. The shaded areas show the position of the selection highlight for insertion of the corresponding element.

6.5.3 Editing elements in the SFC language

You can edit all the elements in an SFC sequence. The commands for editing elements are accessed in the **Edit** menu under **Modify**. Here too, only the commands that can be performed for the current selection are active:

- Initial step
- Final step
- Macro step
- Entry step
- Exit step

6.5.4 Content indication in SFC editor

The content of a body or transition in an SFC can be indicated by a filling in a user-definable color. The color can be defined in the menu **View - Colors - SFC Raster Elements**.

The filling indicates that

- A step includes an assigned action
- A transition includes an assigned condition

6.5.5 Zoom functions

Zoom functions for transitions

If you have not yet created a transition double-clicking on the transition symbol opens the New Transition dialogue box for the definition of a new transition.

If the transition has already been defined double-clicking on the transition symbol opens the editor for the creation of the corresponding PLC program.

Zoom functions for steps

The **Step Zoom Into** option in **Extras - Options - Zoom Header/Body** enables you to define which dialogue box is to be opened when you double-click on a step symbol.

Step Action Association: Action Association dialogue box: Association of an action to a step

First Action If Any: If at least one action is already associated to the step the PLC program of the first action in the list is opened for editing.

Comment: Dialogue box **Comment**: Enables you to enter a comment text. If you want the comment text to be displayed on the screen in the step box you must activate the Extended Information feature in the View menu. This will also display additional information in the Project Navigator window (→ page 3-7).

6.5.6 Assigning Actions to Steps

You can assign actions to every step. An action is a program that can be written in Instruction List (IEC or MELSEC), Ladder Diagram, Function Block Diagram, or MELSEC ST language. These programs are all listed in the Action Pool in the Project Navigator window.

How to associate actions to steps

① Select the step to which you wish to assign an action.

② Select **Edit Action Association** in the Tools menu to open the **Action Association** dialogue box.

③ Press F2 to open the **Action Name List**, which shows all the programs in the Action Pool.

NOTE The procedure for creating a new action is described in chapter 4 ('How to create new objects').

④ Select the required action and confirm with **OK**. The name is then displayed in the **Action Association** field.

⑤ Press Esc to close the dialogue box.

The selection of actions from the **Action Name List** as described in step ③ is for the association of programs already stored in the Action Pool. If no actions have yet been created a message is displayed when you press F2.

The action associated with a step does not have to be a program; you can also associate steps with direct operands, effectively defining the operand as an action.

NOTE Steps that have already been associated with actions are shown shaded in the editor.
If the background colour and the SFC grid element colour are the same, which is set as the default, it is not visible!

6.5.7 Assigning Transition Conditions

Each transition must be associated with a transition condition. A transition condition can be

- a program written in the IL, ST, LD or FBD languages
- a ST expression written in a transition condition window
- a constant TRUE or FALSE
- a Boolean variable with the value of FALSE or TRUE

How to assign transition conditions

① Open the body of the SFC POU.

② Select the transition symbol to which you wish to assign a condition.

③ Enter the name of the condition.

④ Select **Zoom Body** from the **Tools** menu. The New **Transition** dialogue box is displayed with the new transition name.

⑤ Select the programming language you wish to use for the transition program.

⑥ Confirm your selection with **OK**.

The editing window of the selected language editor is then opened, and the name of the new transition is displayed below the Body entry of the SFC POU in the Project Navigator window. You can now write the program for the transition condition.

Transitions that have been assigned a name are shown shaded in the sequence chart, even if the associated program has not yet been created. This means that the identification feature that applies for steps does not apply for transitions; you can only check whether the associated program has been created by looking in the Project Navigator window. If the program has been created the name of the transition will appear under the Body entry in the Navigator.

Programming Transition Conditions

Programming a transition condition is the same as programming any other POU. However, the program must be written in a network. Also, you are only allowed to use one output variable, and this variable must have the name of the transition or the name TRAN.

6.5.8 Enhanced Display

If you need information about the actions contained in a step or programs contained in a transition you can automatically open and arrange the respective windows via the **Enhanced Display** function. If an SFC body is monitored, the monitor mode is automatically activated for the opened bodies.

How to activate the enhanced display

① In an SFC body click on a step or transition to select it.

② To open the **Enhanced Display** context menu
right-click on the SFC body or press [Shift]+[F10].

③ From the context menu select **Enhanced Display - Open action / transition**.

④ The action and the transition of the selected item are opened and arranged as shown below. The monitor mode is automatically activated.

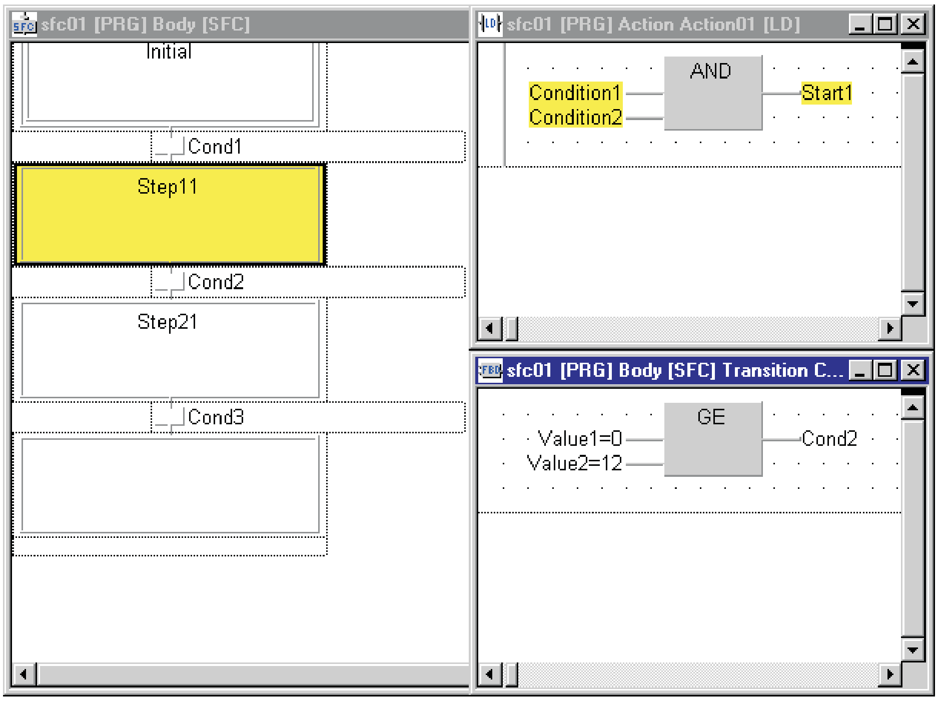


Fig. 6-19: *Enhanced display*

The body of the action is displayed in the upper right section of the SFC body.
The body of the transition is displayed in the lower right section of the SFC body.

If a step was selected, the first action of this step and the body of the following transition are opened.

If a transition was selected, the body of the first action of the previous step and the body of this transition are opened. If more than one previous step is available (parallel sequences) the leftmost located step is displayed.

Actions and transitions which were already opened will be closed before the **Enhanced Display**.

The menu item **Enhanced Display - Close windows** closes all windows which were opened and arranged automatically.

6.5.9 Comments in Sequential Function Chart Language

You can now enter an additional comment for each step in the SFC editor. Each comment can be up to 256 characters long.

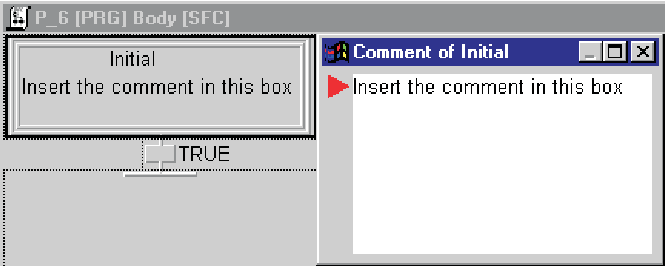
Entering comments

① Optionally activate the **Extended Information - SFC editor** option in the **View** menu to increase the size of the step display in SFC language.

② Select the step in the sequence for which you wish to enter a comment. Take care to only mark the step itself and not its name.

③ Activate the **Edit Step Comment** option in the **Tools** menu to display the comment entry dialogue box.

④ Enter the text of your comment.



P_6 [PRG] Body [SFC]
Initial
Insert the comment in this box
TRUE
Comment of Initial
Insert the comment in this box

Fig. 6-20:
Entering comment text

⑤ Press Ctrl Enter to insert the comment text into the box representing the step.

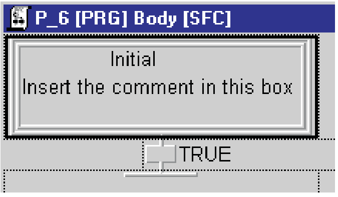


Fig. 6-21:
Inserted comment text

⑥ You can suppress the comment text display by deactivating the **Extended Information - SFC editor** option in the **View** menu.

NOTE

See also the **Comment** option in **Extras - Options - Zoom Header/Body - Step Zoom Into** (→ Chapter 2).

6.5.10 Manual SFC Control

Via the manual SFC control a machine can be controlled by manual operation of the SFC sequence.

By mouse or menu operation the selected steps can either be set or reset. Alternatively, all active steps are reset and all selected steps are set.

Before the corresponding command will be executed, a message box appears including one of the following warnings:

- All selected steps are set.
- All selected steps are reset.
- All active steps are reset and all selected steps are set.

You can execute or cancel the selected command.

Step selection

It is possible to select a step either via the mouse or the keyboard. If you want to select more than one step, you can click on the corresponding steps while pressing the [CTRL] key. Multiple selections via the keyboard are not possible.

Availability

If an SFC body is opened, the sub menu of **Debug - SFC Control** is enabled.

The submenu **SFC Control** is also available in a context menu which can be opened either via the key [SHIFT]+[F10] or a right-click on the SFC body.

The submenu **SFC Control** includes the following menu items:

Set selected steps...: All selected steps are set.

Reset selected steps...: All selected steps are reset.

Replace active by selected steps...: All active steps are reset and all selected steps are set.

The submenu **SFC Control** is available only if the monitor mode is enabled. If the monitor mode is disabled, the submenu is greyed, I. e. not available.

Limitations

If you select more than one step at time and then call the set or replace function, it will not be checked whether the SFC configuration is valid or executable.

The manual SFC control is not available for selected macro steps.

Code generation for manual SFC control

The step flags of every SFC block can be influenced by automatically generated code.
For details refer to 2.8.1.

6.5.11 Sequence errors

The following examples illustrate two typical errors in SFC programs.

An ‘unsafe’ sequence

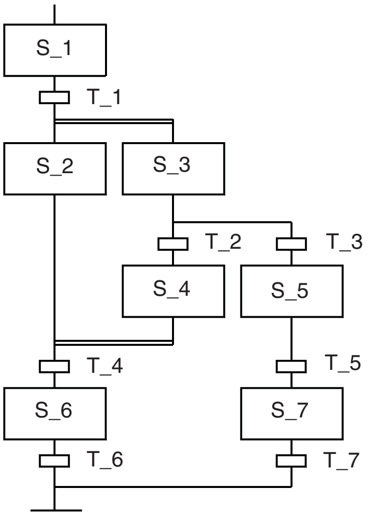


Fig. 6-22:
Transition T_4 can only be polled when both S_2 and S_4 have been executed. However, as soon as T_2 is false and T_3 is true, sequence execution continues with S_5. The result is that it is possible for the process to restart without S_4 being active and without it being possible to execute S_6.
This type of process is referred to as ‘unsafe’.

An ‘inaccessible’ sequence

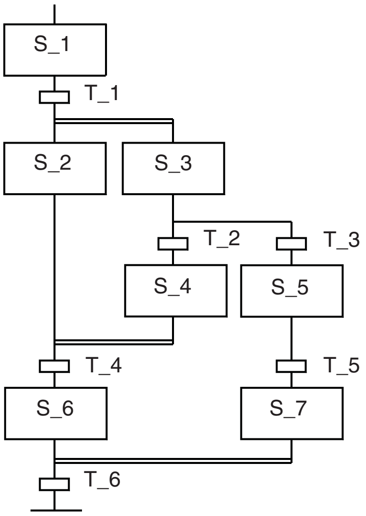


Fig. 6-23:
This sequence suffers from a similar problem. In this case, T_6 is completely inaccessible when the sequence containing steps S_5 and S_7 is executed. T_6 can only be polled if both S_6 and S_7 are active. S_6 can never be active, however, because T_4 can only be polled and execute its transition when S_2 and S_4 are active.

6.5.12 Start, stop, and initialise SFC for QnA and System Q CPUs

To control SFCs in a project from other program parts, GX IEC Developer provides the following functions: Start, stop, and initialise.

Additionally, a function including inverse logic is available.

SFC control functions

All SFC control functions are declared as manufacturer functions in the Manufacturer Library (MLIB). These SFC control functions can be used for QnA and System Q CPUs. They are realised as functions which are not called, but directly expanded by the code generator at the locations where they are used.

Functions:

SFC_START
SFC_PAUSE, SFC_PAUSED
SFC_STOP, SFC_STOPPED
SFC_CTRL

The names used for SFC POUs and SFC steps must be valid. If they are invalid, an error message appears.

Additionally, the used SFC POUs must be assigned to a task. If they are not assigned to a task, an error message appears.

NOTE

Never use **SFC_START after SFC_STOP** for the same POU. Always use **SFC_START before SFC_STOP** or interlock the execution of **SFC_START** so that both functions can not be executed within the same program scan.
Otherwise, errors like multiple active steps might occur.

SFC_START

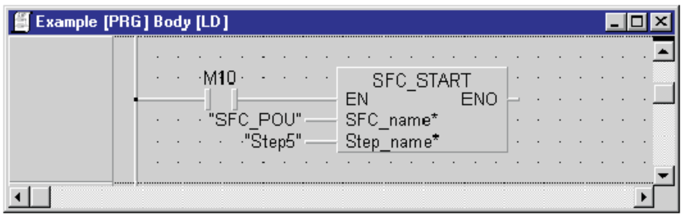


Fig. 6-24:
SFC_START

Via SFC_START a paused SFC POU is started again. The step name is not used for paused SFC POUs.

Via SFC_START also a stopped SFC POU can be started again. If a Step_name is given, the POU is started at the specified step. If several steps should be activated, several SFC_START functions have to be used.

If a paused SFC POU should continue with the last active step, the empty string must be used. A stopped SFC POU will be started with its initial step.

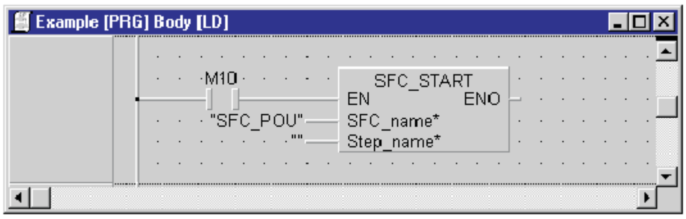


Fig. 6-25:
SFC_START

SFC_PAUSE

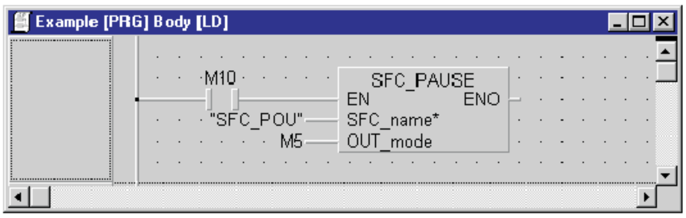


Fig. 6-26:
SFC_PAUSE

When the input EN is TRUE the SFC POU given with SFC_name is paused when it is not already paused.

When the input parameter OUT_mode is TRUE the output of all coils, which are used with an OUT instruction at the step being executed when the SFC POU will be paused, are maintained. When it is FALSE they are switched off.

The OUT_mode behaviour is realised with the special relay SM325. The SFC POU is paused when its according block is not executed. The value of SM325 is used to decide how to handle the coils.

To avoid the influence of other SFC_PAUSE functions with a different OUT_mode set, an additional “Pause-is-Currently-Executed-Flag” is used to pause in one scan all SFC POUs with the same OUT_mode set. At the next scan all SFC POUs will be effected where the OUT_mode has the opposite setting. Therefore for the EN and OUT_mode inputs of the SFC_PAUSE no pulsed devices should be used.

SFC_PAUSED

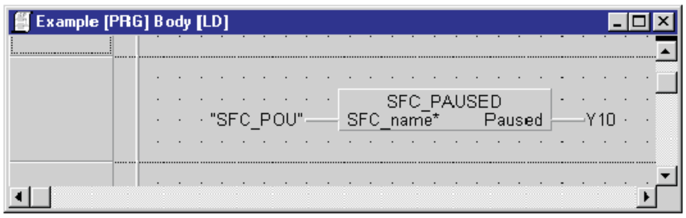


Fig. 6-27:
SFC_PAUSED

The status of the given SFC POU is assigned to the output parameter Paused. If the SFC POU is paused, the output parameter Paused is TRUE. If the SFC POU is not paused, the output parameter Paused is FALSE.

SFC_STOP

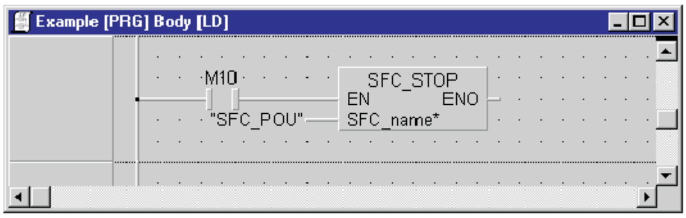


Fig. 6-28:
SFC_STOP

Via SFC_STOP the specified SFC POU is stopped. All steps are reset and the status of the output of all coils are switched off.

SFC_STOPPED

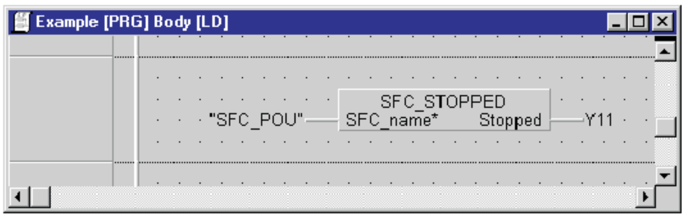


Fig. 6-29:
SFC_STOPPED

Via SFC_STOPPED the status of the given SFC POU is assigned to the output parameter Stopped. The output parameter Stopped is TRUE when the SFC POU is stopped. The output parameter Stopped is FALSE when the SFC POU is not stopped.

SFC_CTRL

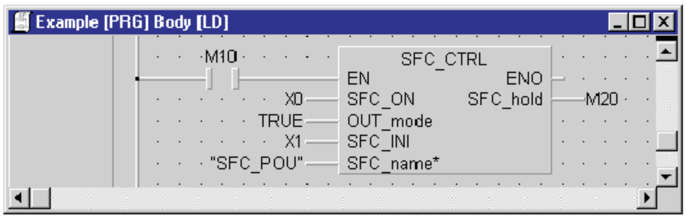


Fig. 6-30:
SFC_CTRL

The function SFC_CTRL is activated via the EN input. If EN is TRUE, the SFC POU is influenced.

If EN is FALSE, the SFC POU is not influenced.

As long as SFC_ON is TRUE the specified SFC POU will be executed.

If SFC_ON is FALSE, the SFC POU is paused. The output of all coils depends on the input parameter OUT_mode at the step being executed when the SFC POU will be paused.

If SFC_INI is TRUE, the specified SFC POU is stopped. All steps are reset and the output of all coils is switched off at the step being executed when the SFC POU will be stopped.

If SFC_INI is FALSE again and SFC_ON is TRUE, the SFC POU is started with the initial step.

If SFC_ON and SFC_INI are both active, the SFC POU is paused and reset.

The SFC_CTRL can be used within the SFC POU which is controlled by the SFC_CTRL. In this case the SFC_CTRL is executed only when the step containing the action including the SFC_CTRL function is active. When the SFC POU is paused or reset the SFC_CTRL must be started outside of the SFC POU.

Conditions of using SFC control functions

The control functions SFC_PAUSE and SFC_CTRL, which include “OUT_Mode” influence the relay SM325. This special relay controls the OUT instruction behaviour of SFC POUs.

Since SFC POUs are also paused when they are assigned to a conditioned task, only one of both methods to control the OUT instruction behaviour is allowed to be active at the same time. This is not checked or guaranteed by the code generator, but must be done by the user. Therefore, you should use only one method at a time to avoid problems.

NOTE

The SFC_STOP function resets a SFC POU to its initial step. The reset of all active steps of a SFC POU is done by resetting the according block device of the SFC POU. The reset of the active steps is then done by the SFC program handler, when the SFC block should be executed.

This could lead to a problem, when after a SFC_STOP the SFC_START function is activated for the same SFC POU in the same scan. The SFC_START function sets the according block device again and continues or starts the SFC POU. It is also possible to parameterise a step name of the SFC POU, which is activated in addition.

While the block device is set again, the reset of the active steps is not done by the SFC program handler anymore, because the block device has not changed from ON to OFF. Furthermore SFC_START could activate a step in addition to the currently active steps.

To avoid this problems, in MM+ 3.00 SP2.5 and GX IEC Developer 4.00 and up:

- SFC_START should not be used after SFC_STOP for the same SFC POU.
- Or the execution of SFC_START should be interlocked, so that both functions could not be executed in the same scan.

6.5.13 ST for Boolean transitions

The permanently visible direct variable name or transition name has been expanded by a window, which is shown below the conditions. In this window IEC 61131-3 ST Boolean arithmetic for the Transition Condition can be edited in textual form. These Boolean expressions meet the IEC 61131-3 standard. The transition is executed according to the Boolean expression.

Normally this window will show one line consisting of approx. 25 characters (depending on the selected font, scrollable). A line not completely visible can be indicated by dots. If the transition or the step before is selected (or if it is active in monitor mode), a window of appropriate size is displayed. If text is added into the window, slider bars might appear. The window is not shown when the transition or the step is selected in offline or monitoring mode.

Tools

The command **Edit Transition Condition**

The command **Edit Transition Condition** is included in the SFC toolbar and in the menu **Tools**.

Edit Transition Condition is available if the SFC editor is active and a transition is selected. The SFC editor is active if a transition rectangle is marked at the moment. Otherwise the SFC editor is disabled.

The function of this command depends on the current status of the transition or on the current content of the transition's rectangle respectively.

a) Transition rectangle is empty (no transition defined yet)

- Create a new transition: name="", language = ST.
- Open the Transition Condition editor.

b) Transition rectangle contains a name which is no variable, no transition name, and not TRUE or FALSE

- Create a new transition: name="", language = ST.
- Set transition rectangle to “”.
- Open the Transition Condition editor.

c) Transition is a Boolean variable whose direct address is TRUE or FALSE

- You are asked whether you want to convert the direct variable into a Transition Condition:

  “Do you really want to replace ‘varX’ by a Transition Condition?”

  – **NO**: nothing is done, exit function

  – **YES**: continues
- Create a new transition: name=""; language = ST.
- Clear the old transition's rectangle name.
- Transform the Boolean variable or direct address into the ST format.
- Open the Transition Condition editor.

d) Transition is already a Transition Condition

Open the Transition Condition editor.

e) Transition is a standard transition including a body in any language

- You are asked:

  “There is a body defined for this transition! Do you want to delete this body?”

  - **NO**: nothing is done, exit function
  - **YES**: continues
- Delete the existing transition body.
- Create a new transition: name=""; language = ST.
- Open the Transition Condition editor.

The command Zoom Body

The command **Zoom Body** opens the Transition Condition editor if the corresponding transition is already a Transition Condition. Otherwise the appropriate editor will be opened (as before).

- If the current transition is a valid device or variable, nothing will happen.
- If the transition is empty (new definition), it depends on settings of the Transition Conditions (**Extras - Options - General Options - Transition Condition**).

The command Zoom Into

The command **Zoom Out** works also in the Transition Condition editor. Changes will be stored to the database.

The command Zoom Out

The command **Zoom Out** works also in the Transition Condition editor. Changes will be stored to the database.

The command End Edit

The command **End Edit** behaves like **Zoom Out.**

The command List Operands

The command **List Operands** opens the dialogue **Variable Selection**, which can be called by means of the menu command, the toolbar icon, or by pressing the key **F2** within the TC editor.

The command List Operators

The command **List Operators** opens the dialogue **Function Block Selection**, which can be called by means of the menu command, the toolbar icon, or by pressing the key **SHIFT+F2** within the TC editor.

The command New variable

The command **New Variable** opens the dialogue **Variable Selection** in the **NewVar** mode. This dialogue can also be called by pressing **ALT+N**.

Transition rectangle

Display

The transition rectangle display contains the beginning (if there is not enough space to display the complete Transition Condition) of the Transition Condition in brackets with a leading '='.

'...' shows that there is not enough space to display the complete condition.

Example ▽

TC:	(X10 AND X11) OR (aBool1 OR aBool2)
Transition Display:	=[(X10 AND X11) OR ((aBoo...]

In the display carriage returns, line feed, and tabs will be replaced by spaces.

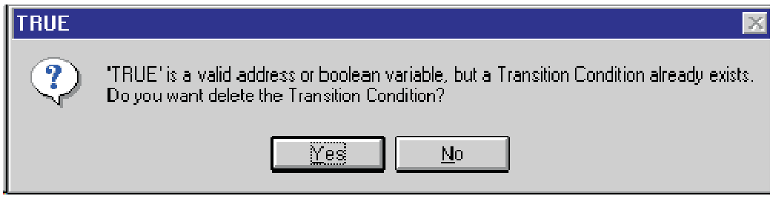
△

Edit field of the Transition name

When leaving the edit field of the Transition name, additional checks will be performed. There are the following situations:

a) A Transition Condition exists and has been replaced by a Boolean variable or address.

- You are asked whether you want to delete the existing Transition Condition.



TRUE

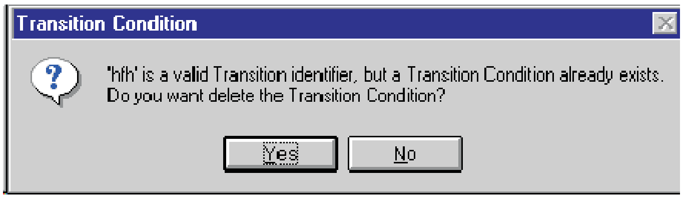
'TRUE' is a valid address or boolean variable, but a Transition Condition already exists.
Do you want delete the Transition Condition?

Fig. 6-31: Confirmation of Transition Condition deletion

- **Yes:** Delete the corresponding ST Transition
  Set the given variable into the Transition's rectangle
- **No:** Nothing will be done, exit function

b) A Transition Condition exists and will be replaced by an identifier not yet defined

- You are asked whether you want to delete the existing Transition Condition.



Transition Condition

'hfh' is a valid Transition identifier, but a Transition Condition already exists.
Do you want delete the Transition Condition?

Fig. 6-32: Confirmation of Transition Condition deletion

- **Yes:** Delete the corresponding ST Transition
  Set the given identifier into the Transition's rectangle
- **No:** Nothing will be done, exit function

Double-click behaviour

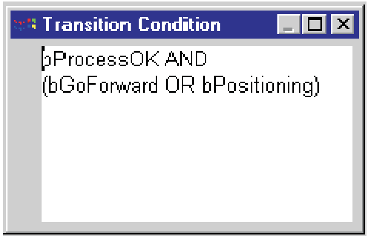
A double click on the Transition’s rectangle has the same effect as using the commands **Zoom Into** or **Zoom Body**.

The double click opens the Transition Condition editor if the corresponding Transition is already a Transition Condition. Otherwise the appropriate editor will be opened.

If the current transition is a valid device or variable, nothing will happen.

If the transition is empty (new definition), it depends on settings of the Transition Conditions (**Extras - Options - General Options - Transition Condition**).

Edit window



```
bProcessOK AND
(bGoForward OR bPositioning)
```

Fig. 6-33:
Edit window for Transition Conditions

The left upper corner of the edit window is located at the right upper corner of the transition rectangle.

The edit window has a maximum and a minimum size. Within this range the window is adapted to the current size of its content.

Syntax

ST Boolean expressions can be entered. All types of IEC 61131-3 ST expressions resulting in a Boolean value can be entered.

The following table shows the IEC 61131-3 ST operators and to their priority level.

Priority level	Operation	Operator symbol	Comments
1	Brackets	( )	
2	Function call	Fun( )	
3	Exponentiation	**	
4	Negation Complement	- NOT	Unary minus
5	Multiplication Division Modulo	* / MOD	Remainder of integer division
6	Addition Subtraction	+ -	Binary minus
7	Comparison	<, >, <=, >=	
8	Equality Inequality	= < >	
9	Logical AND	AND, &	
10	Logical Exclusive OR	XOR	
11	Logical OR	OR	

Tab. 6-6: *IEC 61131-3 ST operators*

It is also possible to enter only the pure expression in the SFC transition (ST Boolean expressions):

X1 AND X2

ST Boolean expressions are nameless. Therefore they are not displayed in the project navigator.

Navigator

Cut, Copy, Paste

Since the TC bodies are not visible for the user, there is no possibility to cut, copy, paste these bodies explicitly.

Cut, copy, and paste the complete SFC POU also include the TC bodies.

SFC UNDO

The **SFC UNDO** function handles the Transition Conditions in a similar way as standard transitions.

Transition body

a) Blank transition before defining a new TC

Take the following steps:

① Delete the created ST body.

② Delete the internal SFC transition body.

b) Direct transition before defining a new TC

A direct transition can be TRUE or FALSE, a direct Boolean variable, or a direct Boolean address.

Take the following steps:

① Delete the created ST body.

② Delete the internal SFC transition body.

③ Restore the old direct transition name.

c) Standard transition before defining a new TC

Take the following steps:

① Delete the created ST body.

② Delete the internal SFC transition body.

③ Restore the old transition name.

Replace an existing TC

a) Direct transition

To replace an existing TC by a direct transition,

① clear the name of the direct transition.

Remark

Step 1 is not implemented because this feature is not available when replacing a standard transition body.

b) Transition name

To replace an existing TC by a transition name,

① clear the name of the transition body name.

Remark

Step 1 is not implemented because this feature is not available when replacing a standard transition body.

Unused transitions

When leaving the SFC editor without saving the changes problems may occur.

In the following situations unused ST bodies will be left:

- Replace a direct transition by a Transition Condition
- Leave the SFC editor and cancel the saving of the changes
- Open the SFC editor. The direct transition is displayed again, but the corresponding transition body still exists in the database

The situations described above are also valid for standard transition bodies, but standard transition bodies are visible for the user. You can delete unused transitions by yourself.

Since Transition Conditions are hidden in the navigator, it is not allowed to create unused ST bodies.

Monitoring

What is monitored?

GX IEC Developer provides a monitoring feature splitting the Transition Conditions into several Boolean expression parts monitored separately. The splitting is done by the Boolean operators, except for **NOT** (see **evaluate** in the following table). If a Boolean operator is included in an expression, the expression is split into two expression parts (one part on the left and one part on the right of the operator). For all parts this algorithm is used recursively, until no more Boolean operators can be found in any part. Brackets will be handled according to the arithmetic rules.

The Boolean operator **NOT** will be handled uniformly. **NOT** will be evaluated with the inverted expression if this is a single monitoring expression (it will be monitored separately). If the inverted expression cannot be monitored because it includes ignored operators for example, the complete expression part including **NOT** will not be monitored neither.

All parts will be searched for comparison operators (see **evaluate** in the following table). If a comparison operator is found, it will be compared with a Boolean value and monitored.

If an expression part includes any other operator, this expression part will be ignored and not monitored (see **ignored** in the following table).

Priority level	Operation	Operator symbol	When monitoring
1	Brackets	( )	**handled**
2	Function call	Fun( )	ignored
3	Exponentiation	**	ignored
4	Negation Complement	- NOT	ignored **evaluate**
5	Multiplication Division Modulo	* / MOD	ignored ignored ignored
6	Addition Subtraction	+ -	ignored ignored
7	Comparison	<, >, <=, >=	**evaluate**
8	Equality Inequality	= < >	**evaluate** **evaluate**
9	Logical AND	AND, &	**separator**
10	Logical Exclusive OR	XOR	**separator**
11	Logical OR	OR	**separator**

Tab. 6-7: *Priority levels of monitored operations*

The following examples show how transitions are monitored. The underlined parts will be monitored as one Boolean value.

Example 1 X1 AND X2 OR X3

Example 2 (D1 < 0) OR ((D2 > 0) AND (D3 > 0))

Example 3 X1 AND (X2 OR X3) AND ((D0 0) OR X3)

Example 4 NOT (D1 < D2)

Example 5 NOT ((D5 > 0) AND (D3 < 0))

Example 6 NOT (D1 < D2) AND (D3 > D4 + D5)

Example 7 (M1 AND M2) OR (M3 AND M4)

Example 8 NOT M1 AND M2 OR M3 XOR M4

Display

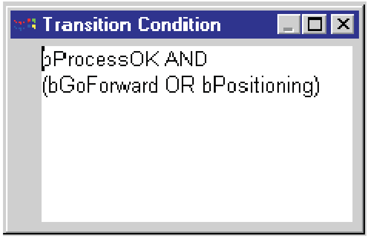
There are three possible display modes: the edit mode, the online mode, and the monitoring mode.

a) Edit mode

Edit control

The edit control opens when the online mode is not active. When the online mode is closed, the edit control opens automatically.

See sections above

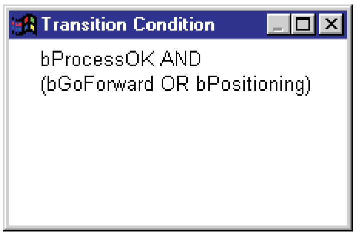


```
bProcessOK AND
(bGoForward OR bPositioning)
```

Fig. 6-34:
Edit mode

b) Online mode

The edit mode is closed. The content of the Transition Conditions is displayed using the standard background colour of the editor.



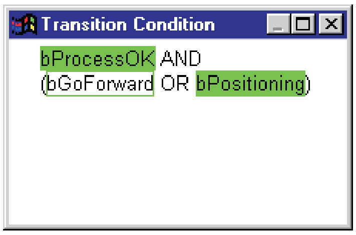
```
bProcessOK AND
(bGoForward OR bPositioning)
```

Fig. 6-35:
Online mode

c) Monitoring mode

The edit mode is closed. The content of the Transition Conditions is displayed using the standard background colour of the editor. The state of the Boolean variables and direct addresses is displayed additionally.

Values which cannot be indicated completely within the window will not be monitored.



Transition Condition

```
bProcessOK AND
(bGoForward OR bPositioning)
```

Fig. 6-36:
Monitoring mode

Starting or stopping the Transition Condition monitor automatically

Starting the Transition Condition monitor

The Transition Condition monitor starts automatically when the SFC body is monitored.

Stopping the Transition Condition monitor

The Transition Condition monitor stops when

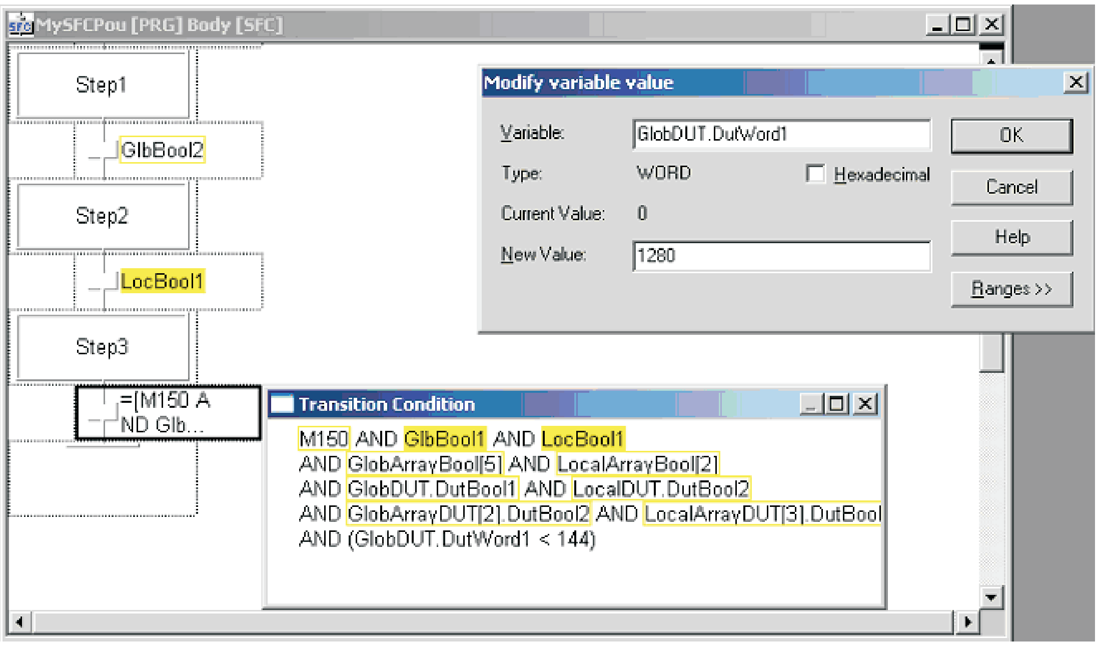
- the Transition Condition editor has been closed.
- the SFC body monitor has been stopped.
- the online mode has been left.

Multiple monitors

Only one Transition Condition monitor can be active per SFC editor. However, it is possible that more TCs can be monitored at the same time if they are included in different SFCs.

Value modification in the SFC TC editor

When an address or variable is double-clicked in a transition condition editor during monitoring, then the **Modify Variable Value** dialogue is opened, and it initializes with the name under the cursor.



Variable: GlobDUT.DutWord1

Type: WORD □ Hexadecimal

Current Value: 0

New Value: 1280

```
M150 AND GlbBool1 AND LocBool1
AND GlobArrayBool[5] AND LocalArrayBool[2]
AND GlobDUT.DutBool1 AND LocalDUT.DutBool2
AND GlobArrayDUT[2].DutBool2 AND LocalArrayDUT[3].DutBool
AND (GlobDUT.DutWord1 < 144)
```

Fig. 6-37: *Modify Variable Value*

6.5.14 SV instruction for QnA

Definitions

SV program	PLC program for supervision (SV)
SV table	Table containing inputs for the SV program

General concept

An array of help memories includes the information when to start an SV and where the information is located in an SV table. If a Help Memory Cell is set to ON, a supervised output is set to ON and the SV starts. All necessary supervision parameters are included in a relative place of the memory cell in the array. The index indicates the reference to an element in an SV table (see below).

If the acknowledge Stop input does not get its expected value within the supervision time, a TimeOut memory cell is activated.

M-cell			Time	Stop	Out	TimeOut	Margin
0 M1000		0	100	X56	Y200	F600	12
1 M1001		1	80	XE2	Y10F	M1700	25
2 M1002	ON	2	140	X10	Y100	M1504	25
3 M1003		3	160	X8F	Y135	F601	50
		4					
		5					
.			.	.	.	.	.
.			.	.	.	.	.
.			.	.	.	.	.
		998					
		999					

Help Memory Cells SV table in the PLC

Example: If the Run input of an OUT_SV function block is active, the according Help Memory Cell (here M1002) is switched to ON, i.e. that the SV program fetches all data from line 2 of the SV table. If the SV times out, the TimeOut memory cell (here M1504), is switched to ON. The values in the SV table are not coded correctly.

Besides F devices the TimeOut column can contain also the device values of other M memory cells, which can be used as trigger for other user program parts.

The Margin fields are reserved for future use.

Each register of the SV table contains only a value, i.e. no memory cell or pointer can be included. The possibilities to code SV table elements are described in section “Coding of the SV table elements” (see page 6-49).

SV library

GX IEC Developer includes an SV library containing all necessary items for supervising:

- **OUT_SV, OUT_SV2 – OUT_SV6**
  Function blocks to activate outputs and start supervising stop inputs
- **SV**
  Supervising program

To enable the SV functionality in GX IEC Developer, the new library must be specified in the file **MMP701.INI** and loaded project into an existing GX IEC Developer via **Project - Other - Update Libraries**.

When creating a new project, use pre-packed projects already containing the User and the Standard libraries for the specified CPU type. After the menu command **Project - New ...** has been executed, the file **MMP701.INI** is scanned automatically in order to check whether additional libraries must be included into the new project. You have the possibility to update the libraries of your new project.

Code generation for OUT_SV

At each step where an **OUT_SV** is used, the code generator of GX IEC Developer sets the according Help Memory Cell to ON to enable the supervision.

The GX IEC Developer code generator maintains an internal code generator SV table including information about all used supervisions. This table collects all static SV parameters during compiling. From the code generator SV table the SV table of the PLC is created and downloaded to the PLC when the program is transferred to the PLC. During online changes the internal code generator SV table and the PLC SV table are updated.

For the **OUT_SV** a predefined function block instance must be used. This is necessary because **OUT_SV** has three outputs which cannot be used together with a function. The same predefined function block instances should be used for different TimeOuts. It is also possible to define only one global OUT_SV function block instance and to use it at several locations.

The following table shows the possible parameters.

Name	Type	Input range
Run	BOOL	Any Boolean expression
Stop	BOOL	X, M, B
_Time	INT	0 – 32767 (0,0 – 3276,7 sec) or a VAR_INPUT if it is used in an FB
Margin	INT (constant)	0 (to disable teach in) / 12/ 25/ 50/ 100 %
Out	BOOL	Y, M, B
T_O (* TimeOut *)	BOOL	Any M or F address

Tab. 6-8: *Possible parameters for OUT_SV*

For all parameters except *Run* only a direct connection to a constant or a variable is allowed. The parameter *Stop* can also be used negatively. The parameters *Out* or *TimeOut* cannot be used negatively.

For the parameter *_Time* only a direct constant or a **VAR_INPUT** of a function block can be used.

VAR_CONSTANT or **VAR_GLOBAL_CONSTANT** are not allowed to be constants. For a **VAR_INPUT** code is produced, which writes the value to the SV table. For each variable a new line is inserted into the SV table to be independent of other entries using constants for the parameter *_Time*.

For the parameter *Margin* only direct constants are allowed. To be able to use **OUT_SV** in a macro FB with **VAR_INPUT** for the parameter *Margin*, the code generator checks whether a constant is used.

The parameter of **OUT_SV** cannot be used directly in other instructions, e.g. *ST InstOUT_SV.Margin* is not allowed.

For the parameters Stop and Out all local variables, global variables, or direct addresses can be used if they meet the address range restrictions described in the section "*Coding of the SV table elements*" (see page 6-49).

For the output parameter *TimeOut* only M or F devices can be used. In this case direct addresses, global variables, or local variables are allowed. If any other device type is used for the *TimeOut,* an error is generated during compiling.

The following figure shows a code which has been programmed manually in an editor.

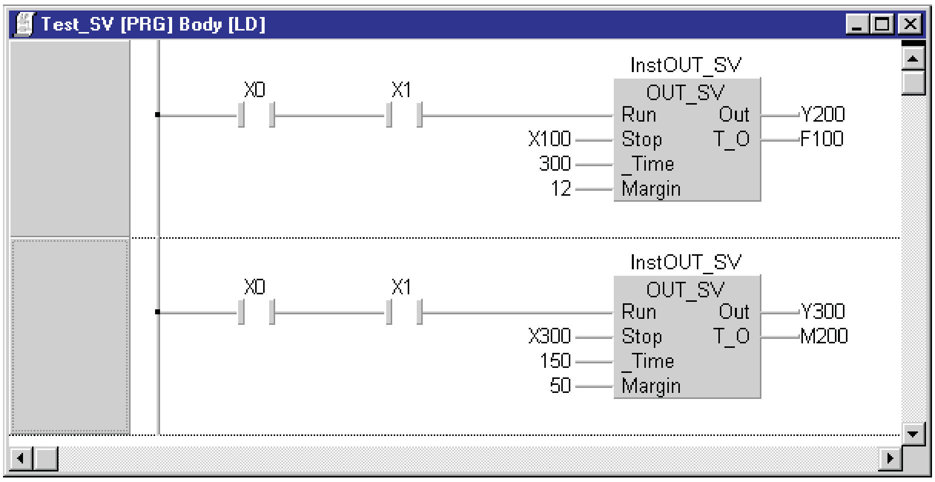


Fig. 6-38: *Sample programs for OUT_SV*

The second network shows an example in which stop has been negated (set with a double click in the **OUT_SV** box near the stop parameter).

Codes generated for the first network:

```
LD              X0
AND             X1
OUT             Help Memory Cell
ANI             X100
OUT             Y200
```

No interlocking is made when the Help Memory Cell is already activated.

It is not necessary to set a Help Memory Cell, because the Help Memory Cell is activated by the SV program. After the activation the value of the Help Memory Cell can change without influencing the SV program.

Codes generated for the second network:

```
LD              X0
AND             X1
OUT             Help Memory Cell
AND             X300
OUT             Y300
```

The SV parameters in the code generation SV table are updated for all networks during compiling.

For each Stop/ Out/ TimeOut combination a new line is added to the internal code generation SV table. If a Stop/ Out/ TimeOut combination is already included in this table, *Time* and *Margin* must be the same. Otherwise an error is generated during compiling.

To allow several Stop/ TimeOut combinations to be used with the same *Time* and *Margin*, additional **OUT_SV** function blocks can be used. They must be defined as new function blocks including their own names, because GX IEC Developer does not support extended output parameters. **OUT_SV2** allows two Stop/TimeOut combinations, **OUT_SV3** allows three combinations, etc.

For each Stop/ TimeOut combination the code generator does the same as for **OUT_SV**, but it uses the same Time and Margin for the SV table.

Example for **OUT_SV6**

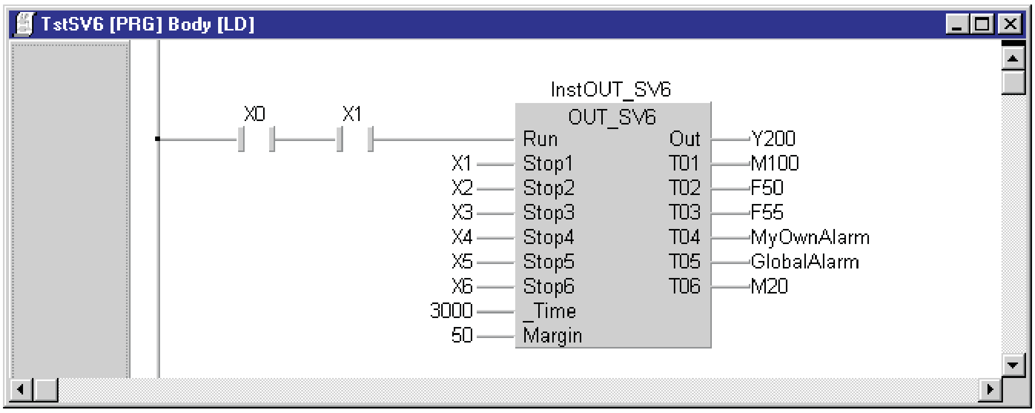


Fig. 6-39: *Example for **OUT_SV6***

Generated codes:

```
LD          X0
AND         X1
OUT         Help Memory Cell 1
  :                   :
OUT         Help Memory Cell 6
LDI         X100
ORI         X20
ORI         M10
ORI         B2F
ORI         MotorOff
ORI         M20
ANB
OUT         Y200
```

For the parameter TimeOut output M and F devices can be mixed.

F devices and User Error file

If F devices are used as TimeOut parameters for **OUT_SV**, the SV will activate the F devices if the Stop input does not get its specified value within the specified time. Via **Debug - User Errors** a list box can be opened in GX IEC Developer displaying the active F devices.

If the project directory includes the file **USER_ERR.TXT**, the project directory is scanned for user defined error texts. For this, the text lines in the file must have the following format:

[F Device Value]   [Space or ',' or Tab]1-n   [Error text]

Example ▽

1 Error text for user error 1

10   Go to Motor 12 and check whether it is running

11, Robot 10 did not reach the target B

△

A line can include 260 characters maximum and the error text 255 characters maximum.

Interface of the SV program

To enable the supervision, assign the SV program to an task always running. It is not possible to include a predefined task in a library in GX IEC Developer. If you program some **OUT_SV** without assigning the SV program to a task, a warning will be generated.

The SV program uses fixed addresses. The fixed addresses will be defined as global variables in the SV library as a reference for the user that these addresses are reserved. They cannot be changed by the user. Internally these addresses will be used directly due to some restrictions of the MELSEC networks concerning symbolic programming and in order to have a quick base for testing. In a second step the SV program will be changed to symbolic programming.

Coding of the SV table elements

Name	Type	Offset (hex)	Input range	Expected status	Number
Stop	M	0	0 - 2FFF	1	12288
	B	3000	3000 - 5FFF	1	12288
	X	6000	6000 - 7FFF	1	8192
	M	8000	8000 - AFFF	0	12288
	B	B000	B000 - DFFF	0	12288
	X	E000	E000 - FFFF	0	8192
Supervision Time	K		0 - 32767	-	
Teach In Margin	K		0 12/ 25/ 50/ 100	no Teach In -	
Out	Y	0	0 - 1FFF	1	8192
	M	2000	2000 - 8FFF	1	32768
	B	A000	A000 - FFFF	1	24576
TimeOut Memory Cell	M	0	0 - 7FFF	-	32768
	F	8000	8000 - FFFF	-	32768

Tab. 6-9: *Coding of the SV table elements*

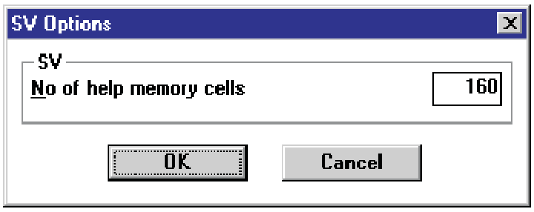
Additional menu commands

The additional menu commands

Extras - SV - Update Time
Extras - SV - Options

are enabled only when the SV library is present, i.e. when the SV library is part of the project.

The values which can be edited by the user are set in the dialogue **Option**.



SV Options

SV

No of help memory cells 160

Fig. 6-40:
SV Options

If the number of the Help Memory Cells is not a multiple of 32, it is adjusted to the next lower value which can be divided by 32. If a value for the number of the SV Help Memory Cells is entered, which is too high, it is adjusted automatically. The maximum value will result from the length of the SV table. The default value will be 160 = 5 x 32 blocks.

When compiling the project, the addresses of the specified Help Memory Cells are allocated and reserved in the system address range of the project. By this, these addresses will not be used directly anywhere else in the project.

The values of the **SV Options** are exported or imported respectively together with the PLC configuration.

All necessary inputs for the SV program will be set by GX IEC Developer via symbolic VAR_GLOBAL_CONSTANTs in the SV library. The user cannot change the values of the constants.

Symbolic name	Content set by GX IEC Developer
SV_HELP_START	Start of SV Help Memory Cells (e.g. 1234 to start at M1234) Automatically calculated by GX IEC Developer.
SV_HELP_BLOCKS	Number of 32 bit blocks of SV Help Memory Cells

Tab. 6-10: *Contents of symbolic constants*

The constants will be updated when the user has changed the value in the dialogue **SV Options** or when new addresses have been allocated by the compiler.

When **SV_HELP_START** or **SV_HELP_BLOCKS** are changed, the SV program and all GX IEC Developer objects, which use **OUT_SV – OUT_SV6**, are set to **Must be compiled**.

6.6 Programming Functions

Programming the header and body of a function is not different from programming a program POU. You can program functions in the following editors:

- Instruction List (IL)
- Ladder Diagram (LD)
- Structured Text (ST)
- Function Block Diagram (FBD)

The two main characteristics of functions are that they output exactly one data element when they are processed and that they always deliver the same output values in response to the same input values. Functions do not have any memory, and although they can contain more than one data element (arrays) they can only output one data element. Functions are thus independent program blocks; they have the same functionality as normal programming instructions, and can be replaced and used in other program sections at will.

NOTE

Each user-defined function uses one system label, irrespective of how often the function is called (→ page 2-39).

6.6.1 Using timers in functions

If you are not careful, programming timers in subprograms, functions and function blocks can lead to problems. You should always remember that once a timer has been triggered it just goes on counting independently, irrespective of whether the condition that triggered the branch to the subprogram, function or function block is still true or not. It is thus better to avoid using timers in this way if possible, and if you can't avoid it you should make allowances for this behaviour in the code of your program.

6.6.2 Declarations

The first step in creating a function is to declare it as a program organisation unit.

Header

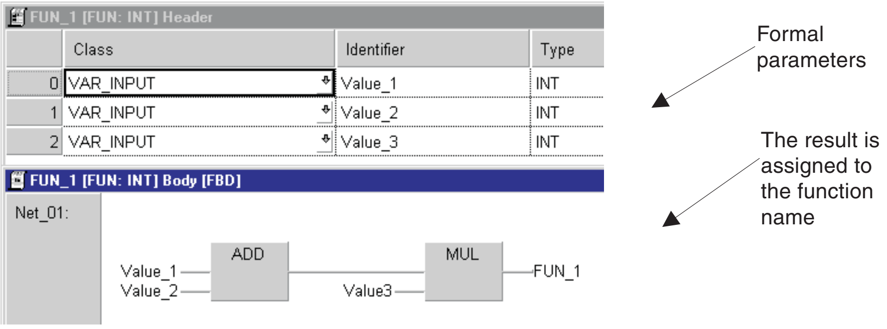
In the header you must declare all the variables used in the function. Function variables can be either of the type VAR, VAR_INPUT or VAR_CONSTANT. The function itself can be one of the data types (BOOL, INT, DINT, WORD, DWORD, TIME, REAL).

The variables declared in the header are also referred to as formal parameters.

Body

The body contains the program code to be performed on the variables declared in the header. The result of the function must be explicitly assigned to the function name.

NOTE The procedure for creating a new function is described in section 4.8, ‘How to create new objects’.



FUN_1 [FUN: INT] Header

	Class	Identifier	Type
0	VAR_INPUT	Value_1	INT
1	VAR_INPUT	Value_2	INT
2	VAR_INPUT	Value_3	INT

FUN_1 [FUN: INT] Body [FBD]

Formal parameters

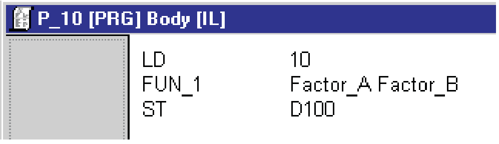
The result is assigned to the function name

Fig. 6-41: User created function block

6.6.3 Calling functions in the text editor

NOTE All variables must be assigned when a function is called; the parameter list must be complete.

When you call a function in Instruction List language you must first load the first parameter into the accumulator with the Load instruction (LD). The next program line then contains the references to the function's name, as the operator, and to the other variables declared in the header. When this statement is executed actual parameters – i.e. specific values – are passed to the remaining formal parameters defined in the header. The result of the function is stored in the accumulator.



```
LD       10
FUN_1    Factor_A Factor_B
ST       D100
```

Fig. 6-42:
Function call in text editor

The value 10 is loaded into the accumulator. This is then followed by the function call FUN_1. The value in the accumulator is added to Factor_A and multiplied by Factor_B. The result is stored both in D100 and the accumulator.

NOTE Please note that you must always assign the values to the variables in the correct order as dictated by the syntax!

Using the editing support function

① Position the cursor in the first column and enter the name of the function.

```
INSERT
```

② Press Ctrl and F1. The parameters are inserted automatically, with dummy variable question marks and the data type in the actual parameter positions.

```
LD       ?STRING
INSERT   ?STRING,
         ?INT
ST       ?STRING
```

③ Overwrite the dummy variable names with a direct address or a variable name.

6.6.4 Calling functions in the graphical editors

The same procedure is used for calling functions in Ladder Diagram and Function Block Diagram.

How to call functions in the graphical editors

① Open the POU body

② Select **Function Block** in the **Tools** menu. This opens the **Function Block Selection** dialogue box.

③ Select the required function's name.

④ Confirm with **OK**.

⑤ The mouse pointer changes to a function 'block' symbol. Position it at the required location in the editing area and press Enter. The graphical block representing the function is inserted at the pointer location. The names of the variables are shown at the function, inside the block representing the function. The name of the function itself is shown at the output, which is also inside the function 'block' symbol.

⑥ Allocate actual parameters (outside) to the formal parameters (inside).

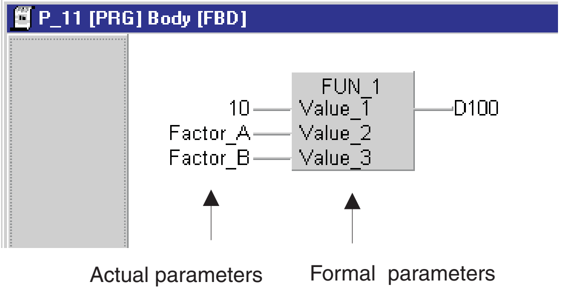


Fig. 6-43:
Function call in graphical editor

6.7 Programming Function Blocks

Programming the headers and bodies of function blocks is no different from programming normal program POUs. You can program function blocks with the following languages:

- Instruction List (IL)
- Ladder Diagram (LD)
- Structured Text (ST)
- Function Block Diagram (FBD)

Function blocks can have one or more outputs and can produce one or more different output values for the same input values. The results are stored after the function block has been processed, and these values are used the next time the function block is called. Function blocks must be instantiated before they can be called in a POU.

6.7.1 Using timers in function blocks

If you are not careful, programming timers in subprograms, functions and function blocks can lead to problems. You should always remember that once a timer has been triggered it just goes on counting independently, irrespective of whether the condition that triggered the branch to the subprogram, function or function block is still true or not. It is thus better to avoid using timers in this way if possible, and if you can’t avoid it you should make allowances for this behaviour in the code of your program.

NOTE The instruction Timer_FB_M uses one Timer for each instance.

6.7.2 Declaration

The first step in creating a function block is to declare it as a program organisation unit.

Header

All the variables used in the function block's code must be declared in the header. Variables in function blocks can be of the types VAR, VAR_IN_OUT, VAR_INPUT, VAR_OUTPUT, VAR_CONSTANT, VAR_GLOBAL or VAR_GLOBAL_CONSTANT. The result of the function can be one of the data types (BOOL, INT, DINT, WORD, DWORD, TIME, REAL).

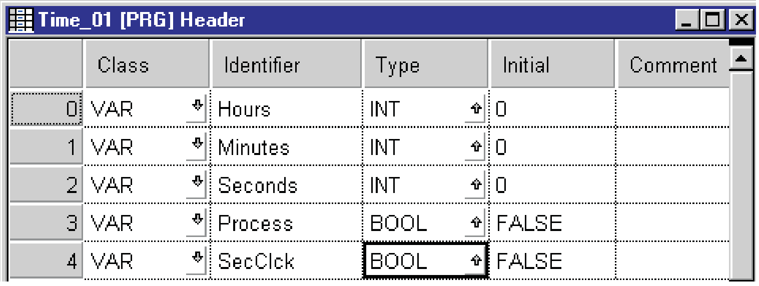
Body

The body contains the program code to be performed on the variables declared in the header. Direct programming of hardware addresses is not permitted.

Example ▽ The function block COUNTER stores the duration of a process. The header contains declarations of three output variables (Hours, Minutes, Seconds), one input variable (Process) and one global variable (SecClck) that addresses relay M9032, which switches once per second.

NOTE The procedure for creating a new function block is described in section 4.8, 'How to create new objects'.

Header of the function block


Time_01 [PRG] Header

	Class	Identifier	Type	Initial	Comment
0	VAR	Hours	INT	0	
1	VAR	Minutes	INT	0	
2	VAR	Seconds	INT	0	
3	VAR	Process	BOOL	FALSE	
4	VAR	SecClck	BOOL	FALSE	

Fig. 6-44:
Function block header

Body of the function block

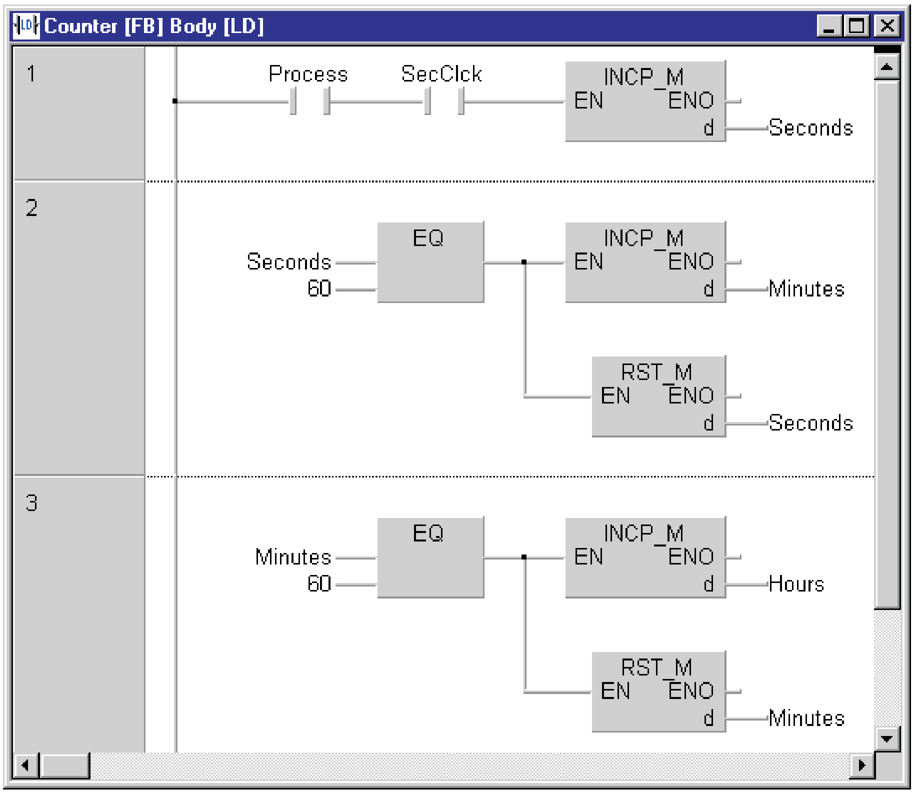


Fig. 6-45: *Function block body*

As soon as an input signal is registered the variable ‘Seconds’ is incremented once per second.
When the value of ‘Seconds’ reaches 60 ‘Minutes’ is incremented by 1 and ‘Seconds’ is reset to 0.
When the value of ‘Minutes’ reaches 60 ‘Hours’ is incremented by 1 and ‘Minutes’ is reset to 0.

△

6.7.3 Instantiation

Function blocks cannot be called directly, they must be called as ‘instances’. When a function block is ‘instantiated’, an instance or copy of it is created in the header declaration table of the POU under consideration. The instance is declared as a variable VAR, for which any identifier can be used. The ‘Type’ is the name of the function block.

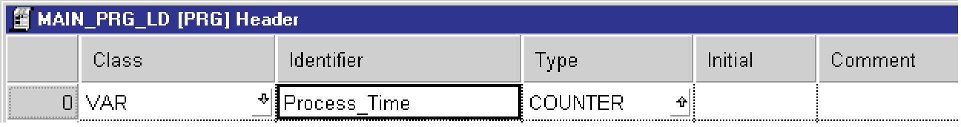
NOTES

You can make multiple instances of one function block. However, remember that each instance you create uses up one system label (→ section 2.8.5).

When using the macro code no label is required.

Example ▽

In this example, the COUNTER function block programmed above is instantiated in the MAIN program POU as PROCESS_TIME.


MAIN_PRG_LD [PRG] Header

	Class	Identifier	Type	Initial	Comment
0	VAR	Process_Time	COUNTER		

Fig. 6-46: *Instantiation example*

△

6.7.4 Calling function blocks in the text editor

You do not have to assign all the variables in the parameter list when you call a function block. The old register values are used for those parameters not assigned in the call.

Function blocks are called with the following instructions

CAL	(Call)
CALC	(CallConditional: conditional call)
CALCN	(CallConditionalNot: negated conditional call)

Calling with a list of input parameters:

Syntax:

```
CAL    Instance Name ( Formal parameter := Actual parameter , ...)
```

Example ▽

The instance name of the function block is PROCESS_TIME. In this example, the actual parameters X0, D1, D2 and D3 are assigned to the formal parameters ‘Process’, ‘Seconds’, ‘Minutes’ and ‘Hours’.

```
CAL      Process_Time(Process:=X0, Seconds:=D1, Minutes:=D2, Hours:=D3)
```

△

Call with input parameter load/store:

Syntax:

```
LD    Actual parameter
ST    Formal parameter
CAL   FBName()
```

Example ▽ The instance name of the function block is PROCESS_TIME. In this example, the actual parameters X0, D100, D200 and D300 are assigned to the formal parameters ‘Process’, ‘Seconds’, ‘Minutes’ and ‘Hours’.

```
LD       X0
ST       Process
LD       D100
ST       Seconds
LD       D200
ST       Minutes
LD       D300
ST       Hours
CAL      Process_Time()
```

△

Using the editing support function

① Position the cursor in the first column and enter the CAL instruction.

② Position the cursor in the second column and press F2.

③ Select the **function block instance (FB_INST in the example) from the** Variable Selection dialogue box.

```
CAL   FB_INST
```

④ Press Ctrl and F1. The parameters are inserted automatically. The names of the internal function block variables are entered as the formal parameters. Dummy variable question marks and the data type are displayed in the actual parameter positions.

```
CAL    FB_INST(IN1:=?BOOL,
       IN2:=?BOOL,
       IN3:=?BOOL)
```

⑤ Overwrite the dummy names with either a direct address or a variable name. Tip: If you select the dummy name (?BOOL) and press F2 you can select the variable directly from the list displayed. Select **Apply** to enter the variable.

```
CAL    FB_INST(IN1:=X1,
       IN2:=VAR_EIN1,
       IN3:=VAR_EIN2)
```

6.7.5 Calling function blocks in the graphical editors

How to call function blocks in the graphical editors

① Open the body of the program POU in which you wish to call the function block.

② Select **Function Block** in the **Tools** menu. This opens the **Function Block Selection** dialogue box.

③ Select the required function block's name and confirm with **OK**.

④ Move the mouse pointer to the desired position in the editing window and press the Space Bar to insert the function block.

⑤ Press F2 for a list of the instance names declared in the header.

⑥ Select the appropriate instance name and confirm with **OK**. The name is then used as the identifier for the function block instance.

⑦ Terminate the procedure by pressing Enter.

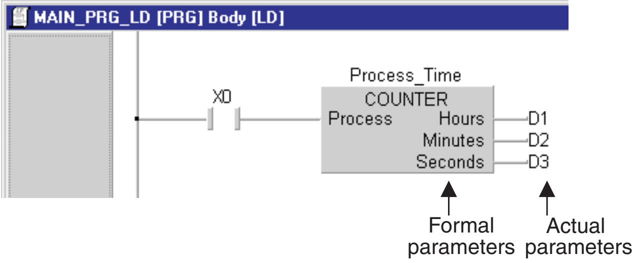


Fig. 6-47:
The names of the formal parameters are shown in the function block. The function block type is COUNTER, the instance name is 'Process_Time' (→ Header).

⑧ Assign the actual parameters (outside) to the formal parameters (inside).

6.7.6 Multiple use of a single instance

You can use the same instance of a function block in several different places within a POU, or, if it is defined globally, in several different places within the same PLC program.

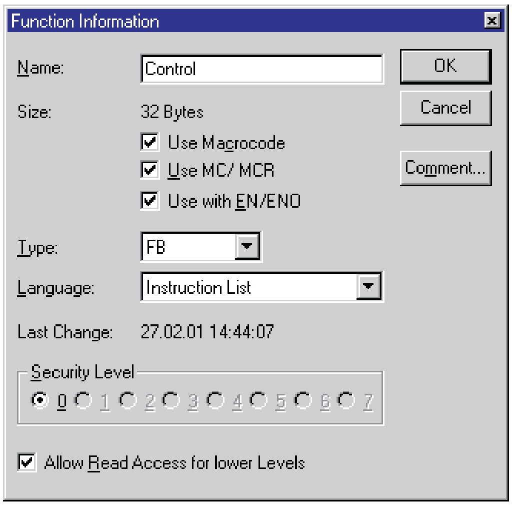
NOTE If the EN of an FB or an FUN is zero, the output is not defined.

6.7.7 Execution of Function Blocks

Function blocks can be executed in different ways:

- Macrocode execution
- MC-MCR execution
- Use with EN/ENO

The execution mode is selected in the **Function Information** dialogue box.



Function Information

Name: Control

Size: 32 Bytes

☑ Use Macrocode
☑ Use MC/ MCR
☑ Use with EN/ENO

Type: FB

Language: Instruction List

Last Change: 27.02.01 14:44:07

Security Level
● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access for lower Levels

Fig. 6-48:
Function block information

How to set the execution option

① Select the function block in the Project Navigator window.

② Display the Function Information dialogue box by pressing Alt Enter.

③ Activate the check box. The **Use MC-MCR** option can only be activated when the other two options have already been activated.

This does not make any changes to instantiation and the programming of instances in the various programming languages.

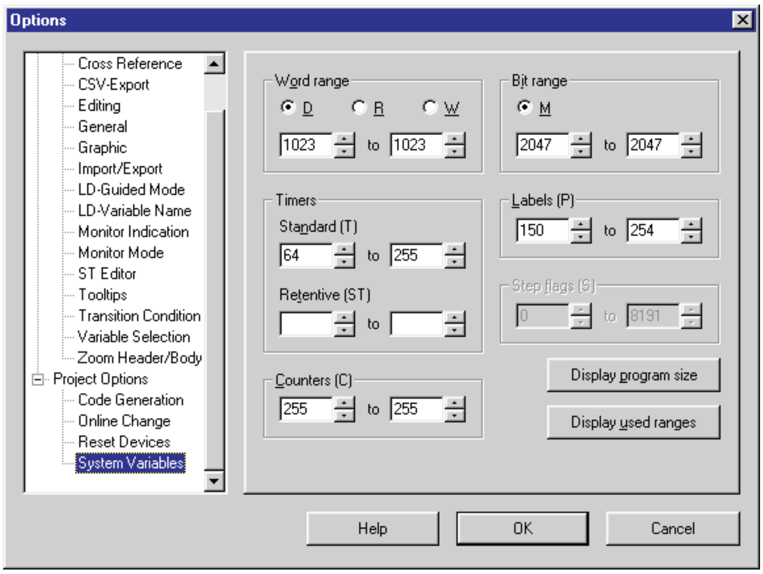
Macrocode execution

- Standard execution: The function block is called via a system label
- Macrocode execution: The function block is expanded internally

With Macro Code	Without Macro Code (standard execution)
No internal system labels are needed to execute a function block instance.	Each instance uses internal system labels (pointers). [1]
Consequence: The number of function blocks you can use is only limited by the size of the PLC memory as function blocks are independent of system labels.	Consequence: Since the number of available system labels is limited (FX: 128, A: 256, Q: 1024) you cannot use more than a theoretical limited number of function blocks. In practice this number is even smaller as system labels are also required for other internal processes.
User-oriented execution of the function block	Implementation of the function block construct in conformity with the IEC 61131-3 standard
No restrictions on the handling of timers and coils within the function block.	Restrictions on the handling of timers and coils within the function block (subroutines).

Tab. 6-11: *Macrocode execution*

[1] To set the system labels parameter select **Extras - Options** and then click on **System Variables** (→ page 2-39).


Options

Word range: ● D ○ R ○ W — 1023 to 1023

Bit range: ● M — 2047 to 2047

Timers — Standard (T): 64 to 255

Retentive (ST): to

Labels (P): 150 to 254

Step flags (S): 0 to 8191

Counters (C): 255 to 255

Fig. 6-49:
System Variables

The maximum permitted number of system labels (internal pointers) is limited by the used PLC CPU.

Macrocode execution, MC-MCR execution, Use with EN/ENO

The function blocks FB_1 through FB_5 and the corresponding instances i_FB_1 through i_FB_5 show the different execution possibilities.

The function block

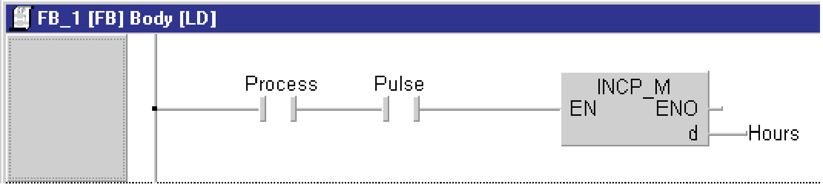


Fig. 6-50:
Function block for macrocode execution

Use of the function block

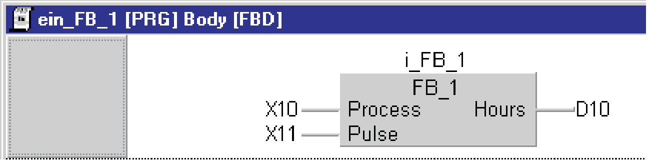


Fig. 6-51:
Function block instance

Macrocode execution options

	FB_1	FB_2	FB_3	FB_4	FB_5
Macro	**NO**	**YES**	**NO**	**YES**	**YES**
MC-MCR	**NO**	**NO**	**NO**	**NO**	**YES**
EN-ENO	**NO**	**NO**	**YES**	**YES**	**YES**
Generated Code	LD X10 OUT M4096 LD X11 OUT M4097 LD SM400 CALL P2048 LD SM400 MOV D6144 D10	LD X20 AND X21 INCP D20	LD M3 OUT M4098 AND X30 SET M4099 LD M4098 ANI X30 RST M4099 LD M4098 AND X31 SET M4100 LD M4098 ANI X31 RST M4100 LD M4098 OUT M4101 CALL P2049 LD M4101 OUT M31 LD SM400 MOV D6145 D30	LD M4 OUT M41 LDI M4 CJ P2050 LD X40 AND X41 INCP D40P2050	LD M5 OUT M51 MC N0 M4103 LD X50 AND X51 INCP D50 MCR N0
Instance (no macro)	LD M4096 AND M4097 INCP D6144 RET		LD M4099 AND M4100 INCP D6145 RET		
System labels	1 System label (no macro)	No system labels (macro)	1 System label (no macro)	1 System label (EN/ENO and jump)	No system labels (macro and MC/MCR)
Behaviour of: **- Timers** **- Counters** **- Outputs**	The function block is always executed. The timers, counters and outputs are thus controlled by the program.	The function block is always executed. The timers, counters and outputs are thus controlled by the program.	The function block is only executed if the EN input is set. This means that the timers, counters and outputs retain their status from the last execution. Timers continue to run. The status of counters, accumulator/retentive timers and outputs in OUT instructions is preserved.	The function block is only executed if the EN input is set. This means that the timers, counters and outputs retain their status from the last execution. Timers continue to run. The status of counters, accumulator/retentive timers and outputs in OUT instructions is preserved.	The statuses of the devices are reset: - Timer: Count value reset to 0, contacts remain off - Retentive timers and counters: count value and input contact status preserved, output contact is reset - Devices in OUT instructions: outputs are reset - Devices in SET, RST, SFT instructions: current status preserved

Tab. 6-12: Macrocode execution options

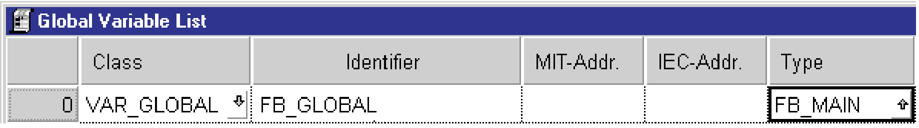
6.7.8 Global Function Blocks

Function blocks allow global instancing, so that the same instance can be used in different program organisation units.

How to create a global instance

① Create a program organisation unit as a function block (in our example: MAIN).

② Create an instance (Main_Instance1) of this function block (MAIN) in the Global Variable List.



Global Variable List

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type
0	VAR_GLOBAL	FB_GLOBAL			FB_MAIN

Fig. 6-52: Created instance

③ Call up the function block in the body, and allocate to it the name of the instance.

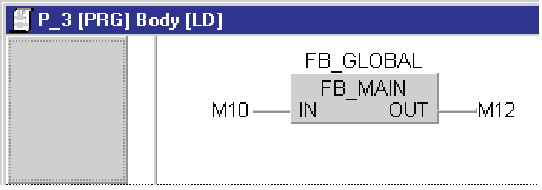


Fig. 6-53: Function block allocated to instance

6.8 Programming Data Unit Types

Data unit types (DUT) are structured, derived data types containing a collection of variables which can be of different data types. Data unit types can be declared as global or as local variables.

NOTE

Data unit types can only be used in the IEC editors. DUT components or arrays can not be used in MELSEC IL

6.8.1 Declaration

Data unit types must be defined in the DUT_Pool.

All the variables in the DUT structure must be declared in a special declaration table. After this you must then create one or more instances of the data unit type in the global variables declaration table.

How to declare a data unit type

① In the **Object** menu select **New**, then select **Data Unit Type**.

② Enter a name for the new DUT (max. 32 characters) and confirm with **OK**. The name of the new DUT is displayed in the DUT_Pool of the Project Navigator window.

③ Double-click on the name of the new DUT in the Project Navigator window to open its special declaration table.

④ Enter a name (identifier) for the variable in the first field.

⑤ Select the data type for this variable in the second field.

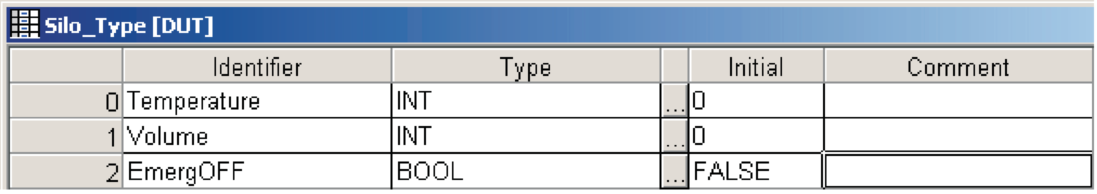
⑥ The default initial value (Initial) is added automatically. If you wish, you can also enter an optional comment for the variable.

⑦ Repeat steps ① – ⑥ until all the variables of the data unit type are defined.

Example ▽

The following example shows the creation of a data unit type called `SiloType`. The variable collection of `SiloType` contains two variables of the type INT and one variable of the type BOOL.

① Enter the following lines in the declaration table for the data unit type:



Silo_Type [DUT]

	Identifier	Type		Initial	Comment
0	Temperature	INT	...	0	
1	Volume	INT	...	0	
2	EmergOFF	BOOL	...	FALSE	

Fig. 6-54: Declaring variables of a data unit type

② Save your entries.

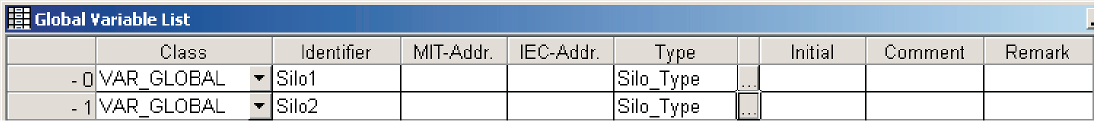
△

6.8.2 Use of the DUT as global variable

How to declare the DUT

① Double-click on **Global_Vars** in the Project Navigator window.

② Enter the following lines in the global variables declaration table:



Global Variable List

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type		Initial	Comment	Remark
- 0	VAR_GLOBAL	Silo1			Silo_Type	...			
- 1	VAR_GLOBAL	Silo2			Silo_Type	...			

Fig. 6-55: *Declaring the data unit type as global variable*

`Silo1` and `Silo2` are the names of the variable of data type `Silo_Type`. Since the elements in type `SiloType` can have different data types the address assignment is handled automatically by the PLC, and no default initial values are shown.

Example ▽

You have created a Data Unit type called `SiloType`. Two variables of the type `SiloType`, `Silo1` and `Silo2`, have now been declared in the global variables list. The structure of both is identical, so to reference the individual variables of each DUT you only needs to prefix their names with the name of the respective global variable.

In `Silo1` the values for the temperature and volume variables are determined by the contents of registers D0 and D1, in `Silo2` by D10 and D11. In `Silo1` the Emergency Off (EmergOFF variable) switch is controlled by the status of input X0, in `Silo2` by input X1.

The variables are stored in the Global Variable List. If you apply an element of an instance of a DUT or FB (e. g. Motor.Status) from the Global Variable List to an editor via the **Variable Selection** dialogue, the corresponding DUT or FB is automatically inserted in the header of the editor. Like for simple data types the library <Global Variables> must be selected in the **Variable Selection** dialogue.

Instruction List:

SILO1

```
LD      D0
ST      Silo1.Temperature
LD      D1
ST      Silo1.Volume
LD      X0
ST      Silo1.EmergOff
```

SILO2

```
LD      D10
ST      Silo2.Temperature
LD      D11
ST      Silo2.Volume
LD      X1
ST      Silo2.EmergOff
```

Ladder Diagram and Function Block Diagram:

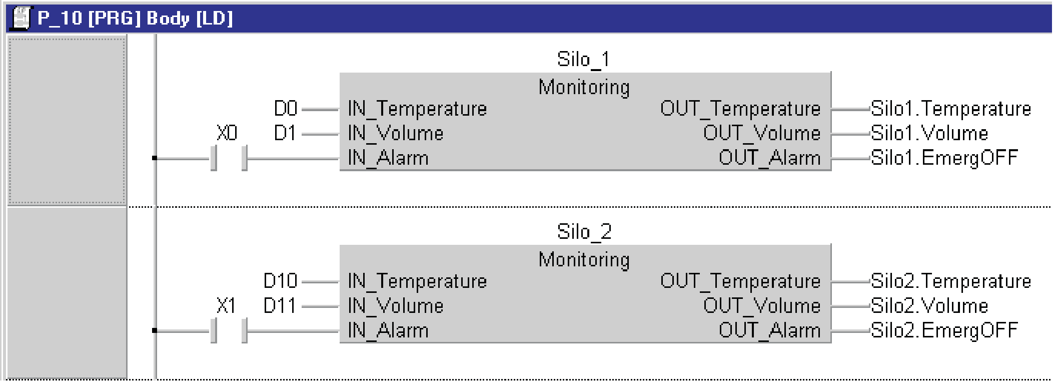


Fig. 6-56: *Ladder Diagram and Function Block Diagram*

In the Ladder Diagram version of the example a function block of the type ‘Monitoring’ has been programmed for assigning the register values and the Boolean input to the elements of the DUTs. Two separate instances (`Silo_1` and `Silo_2`) of this function block were then created for the two silos.

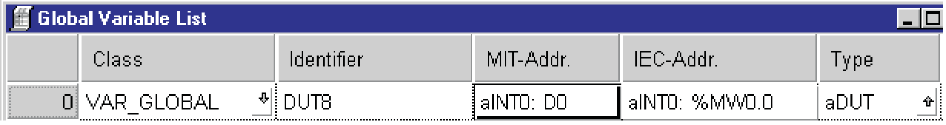
△

Define direct addresses for data unit variables

The GVL has been extended to define addresses for all elements of a data unit variable. Not defined addresses are handled by the system.

Appearance and handling

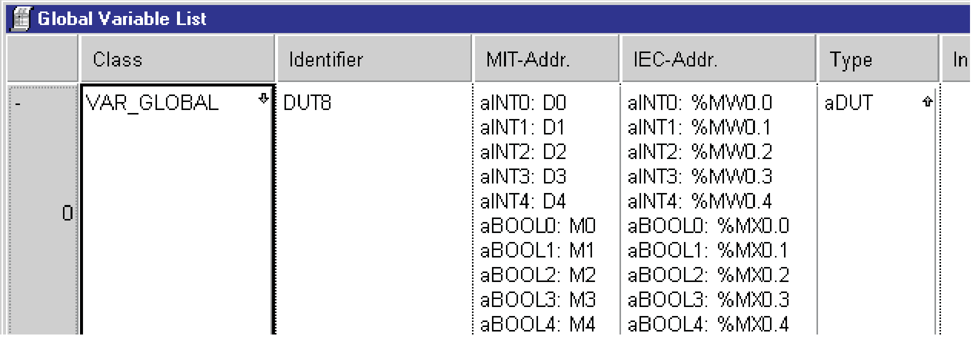
In the GVL data unit variables will display their IEC and Mitsubishi address fields in controls similar to push buttons. On these controls the elements for which an address is defined will be displayed in the form ‘element: address’.


Global Variable List

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type
0	VAR_GLOBAL	DUT8	aINT0: D0	aINT0: %MW0.0	aDUT

Fig. 6-57: *Control elements of data unit types*

To view all definitions at once (if more than one definition is available), DUT entries in the GVL can be expanded by double-clicking the row number field.



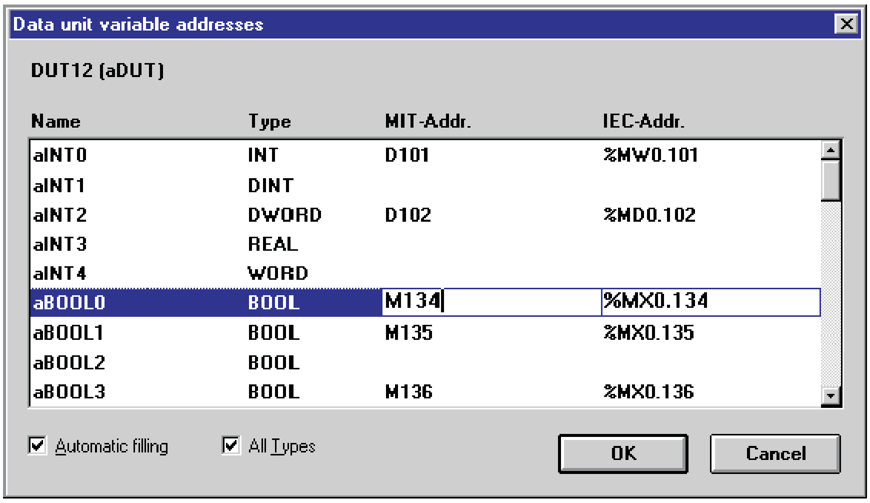
Global Variable List

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	In
- 0	VAR_GLOBAL	DUT8	aINT0: D0 aINT1: D1 aINT2: D2 aINT3: D3 aINT4: D4 aBOOL0: M0 aBOOL1: M1 aBOOL2: M2 aBOOL3: M3 aBOOL4: M4	aINT0: %MW0.0 aINT1: %MW0.1 aINT2: %MW0.2 aINT3: %MW0.3 aINT4: %MW0.4 aBOOL0: %MX0.0 aBOOL1: %MX0.1 aBOOL2: %MX0.2 aBOOL3: %MX0.3 aBOOL4: %MX0.4	aDUT	

Fig. 6-58: Expanded DUT entries in the GVL

Editing

It is not possible to edit the addresses directly in the GVL. To edit the EC and Mitsubishi addresses, click on a DUV's IEC or Mitsubishi address field. Then a separate dialogue (application modal) will appear and you can edit the addresses for each DUV element.



Data unit variable addresses

DUT12 (aDUT)

Name	Type	MIT-Addr.	IEC-Addr.
aINT0	INT	D101	%MW0.101
aINT1	DINT		
aINT2	DWORD	D102	%MD0.102
aINT3	REAL		
aINT4	WORD		
aBOOL0	BOOL	M134	%MX0.134
aBOOL1	BOOL	M135	%MX0.135
aBOOL2	BOOL		
aBOOL3	BOOL	M136	%MX0.136

☑ Automatic filling ☑ All Types

Fig. 6-59: Opened editing dialogue for data unit variables

After you have selected an element you can edit this element like in other address fields in the GVL (e.g. by. double clicking or pressing the space key).

When the two edit fields are displayed, the focus can be moved between them by clicking on one of the fields or pressing the **TAB** key. By pressing **ENTER** or clicking outside of the edit fields the editing will be finished and the modifications accepted. By pressing **ESC** the editing will also be finished, but the modifications are cancelled.

If you activate the **Automatic filling** checkbox, when entering a address all variables of the same type that are not yet addressed will be addressed automatically.

MELSEC Q/QnA only:

If the function **Automatic filling** is activated the additional function **All types** is available.

If you activate the **All types** function, when entering an address all addresses of all variable types are filled automatically.

For timers and counters the **All types** function is not available, addresses of these types are not filled automatically neither.

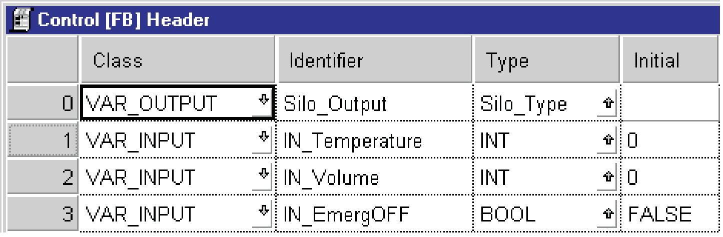
6.8.3 Use of the DUT as local variable

Data unit types can also be passed directly as variables:

- As input and output parameters in function blocks
- As input parameters in functions

Create the Control function block

The function block processes the input variables and passes the results to the output variables. The input variables `IN_Temperature, IN_Volume` and `IN_EmergOff` are declared in the header, along with the output variable `Silo_Output. Silo_Output` is declared as the data unit type `Silo_Type`.



Control [FB] Header

	Class	Identifier	Type	Initial
0	VAR_OUTPUT	Silo_Output	Silo_Type	
1	VAR_INPUT	IN_Temperature	INT	0
2	VAR_INPUT	IN_Volume	INT	0
3	VAR_INPUT	IN_EmergOFF	BOOL	FALSE

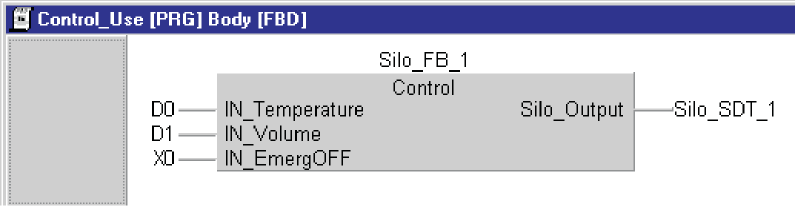
Fig. 6-60:
Declaring variables for the control function block

Calling the `Control`_Use function block and passing the data unit type directly as an output parameter

The function block `Control` is called in the program organisation unit `Control_Use`, creating the instance `Silo_FB_1`.

The input variables of the function block obtain their values from devices `D0, D1` and `X0`.

The output variables of `Silo_Output` are passed to the data unit type `Silo_DUT_1`.



Control_Use [PRG] Body [FBD]

Fig. 6-61:
Control function block

6.9 Programming Arrays

An array is a collection of variables of the same data type. GX IEC Developer supports arrays with up to a maximum of three dimensions.

NOTE Arrays are only supported in the IEC editors.

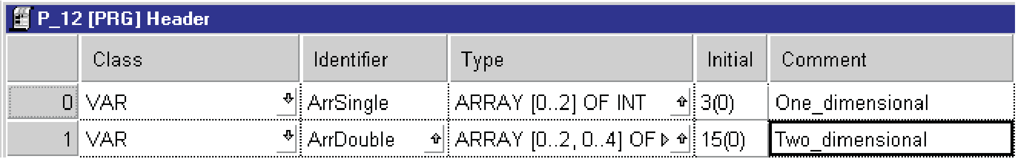
6.9.1 Declaration

Arrays are declared in the header of the program POU. They can be declared as local or global variables.

Example ▽ This example shows the definition of a one-dimensional array called `ArrSingle` with three elements of the type INT, and a two-dimensional array called `ArrDouble` with three elements of the type INT in the first dimension and five elements of the type INT in the second dimension.

① Open the header of a POU.

② Add a new line to the table with the following entries:



P_12 [PRG] Header

	Class	Identifier	Type	Initial	Comment
0	VAR	ArrSingle	ARRAY [0..2] OF INT	3(0)	One_dimensional
1	VAR	ArrDouble	ARRAY [0..2, 0..4] OF ▸	15(0)	Two_dimensional

Fig. 6-62: Declaring arrays in the program header

VAR: Definition of the array as a local variable in the program organisation unit ‘P_12’.

ArrSingle/ArrayDouble: Names of the arrays.

ARRAY [0..2] OF INT: The one-dimensional array has three elements (0 – 2) of the type integer (INT)

ARRAY [0..2, 0..4] OF INT: This array is two-dimensional. The second dimension is separated by a comma between the braces. In this example the first dimension of ArrDouble has three elements (0 – 2) and the second dimension has five elements (0 – 4).

3(0): Number of elements (3) with the default value 0 (0)

15(0): Number of elements (3 x 5 = 15) with the default value 0 (0)

△

6.9.2 Calling

Instruction List:

- One-dimensional arrays

```
LD    Actual parameter
ST    Array Name [ Element ]
```

- Two-dimensional arrays

```
LD    Actual parameter
ST    Array Name [ Element of 1st Dimension ,
      Element of 2nd Dimension ]
```

Example ▽

A two-dimensional array called `ArrDouble` has been defined with three elements in the first dimension and five elements in the second dimension.

The data registers D0 and D1 are assigned to the following elements:

D0: Position 1 in the first dimension I, position 2 in the second dimension II
D1: Position 2 in the first dimension I, position 3 in the second dimension II

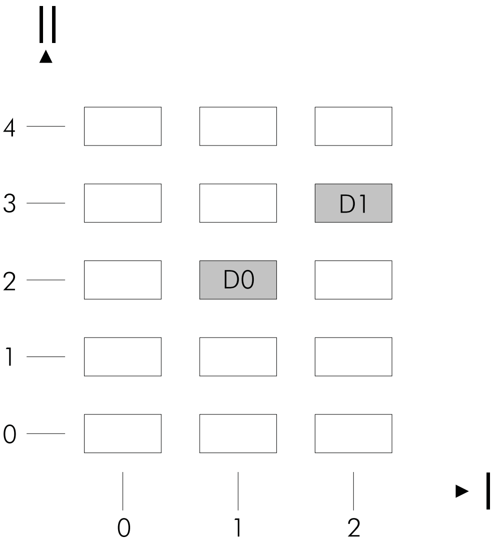
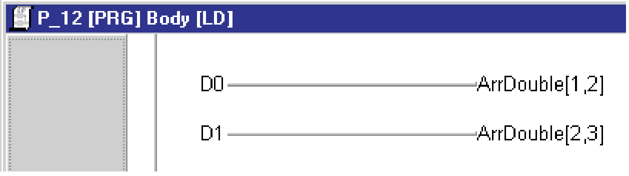


Fig. 6-63:
Two-dimensional array: "ArrDouble"

Instruction List:

```
LD     D0
ST     ArrDouble[1,2]
LD     D1
ST     ArrDouble[2,3]
```

Ladder Diagram and Function Block Diagram:



```
D0 ───────────── ArrDouble[1,2]
D1 ───────────── ArrDouble[2,3]
```

Fig. 6-64:
Ladder and Function Block Diagram for arrays

△

6.10 Programming Timers and Counters

6.10.1 TIMER and COUNTER Functions

The timer and counter functions are stored in the manufacturer library.

Timer Functions	Counter Functions	Operands	Action
TIMER_M	COUNTER_M	Normal timers/counters and extended timers/counters	Pass set point value and call (start) timer/counter
TIMER_VALUE_M	COUNTER_VALUE_M	Extended timers/ counters only	Pass set point value
TIMER_START_M	COUNTER_START_M		Call (start) timer/counter
TIMER_H_M	-	MELSEC Q/QnA series' high-speed timer	Pass set point value and call (start) timer Set timer intervals by parameters

Tab. 6-13: *TIMER and COUNTER functions*

Timers and counters always have the following four elements:

	Timer	Counter
Set point value	TValue	CValue
Actual value	TN	CN
Output coil	TC	CC
Input contact (status)	TS	CS

Tab. 6-14: *TIMER and COUNTER elements*

NOTE The use of the various functions is illustrated below using a timer as an example. Counters are used in exactly the same way.

The TIMER_M function

NOTE A detailed example of the use of the TIMER_M function is provided in the Beginner's Manual.

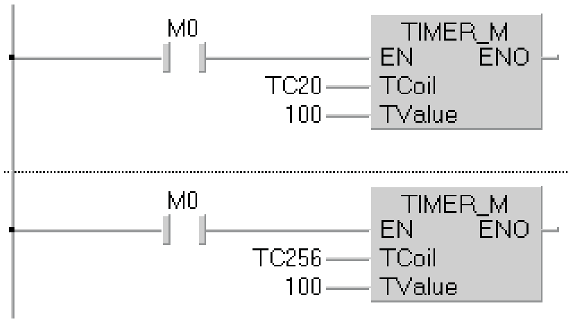


Fig. 6-65:
TIMER_M
Use of the normal timer TC20

Use of the extended timer TC256

The TIMER_M function both passes the set point value and starts the timer TC (TCoil). The set point value is passed to TValue. With normal timers the set point value is passed directly. With extended timers the passed set point value is assigned to the timer and the set point value register (see below).

The TIMER_VALUE_M and TIMER_START_M functions

Using TIMER_VALUE_M and TIMER_START_M together provides exactly the same functionality as TIMER_M. The only difference is that the parameter passing and timer start operations are handled separately, which makes it possible to use them at different points in the PLC program. TIMER_VALUE_M and TIMER_START_M can only be used with the extended timers, and you must define the set point values registers for the extended timers.

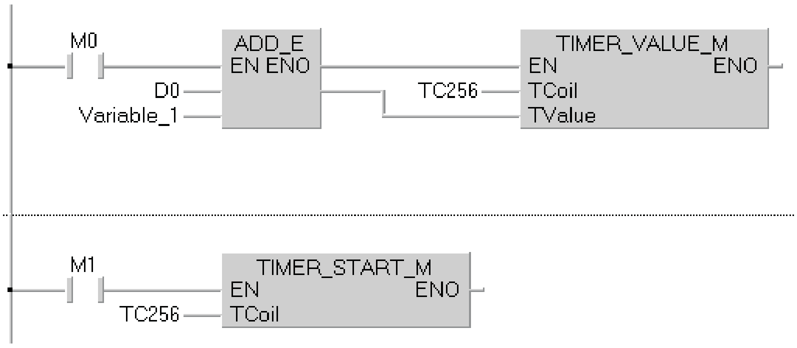


Fig. 6-66:
TIMER_VALUE_M
Passes the reference value to the extended timer TC256

TIMER_START_M
Starts timer TC256

Defining the reference value registers for extended timers and counters

You must always define and set a number of parameters when you use timers or counters. See Section 2.8.5 System Variables for the setting of Timer and Counter.

6.10.2 TIMER and COUNTER Function Blocks

A selection of new standard function blocks are provided for programming local timers and counters. The system automatically uses free timers and counters from the system range.

Advantages: No direct timer and counter addresses are required. You can create any number of instances – depends on the number of system timers – with different timers from just one function block.

The following function blocks are provided:

Function Blocks	Meaning	Comment
TIMER_10_FB_M	10-ms-Timer	
TIMER_100_FB_M	100-ms-Timer	
TIMER_CONT_FB_M	retentive timer	
TIMER_HIGH_FB_M	high-speed timer	MELSEC Q/QnA series only
TIMER_LOW_FB_M	low-speed timer	MELSEC Q/QnA series only
TIMER_CONTHFB_M	retentive high-speed timer	MELSEC Q/QnA series only
COUNTER_FB_M	counter	

Tab. 6-15: *TIMER and COUNTER Function Blocks*

MELSEC Q/QnA series' timers

The timers will act as high-speed or low-speed timers when they are activated by the appropriate instructions in the PLC program.

The timer intervals for the high-speed and low-speed range can be set in the PLC Setup dialogue box (→ page 5-50). In order to be compatible to A series projects, the function blocks TIMER_10_FB_M and TIMER_100_FB_M are available in the MELSEC Q/QnA series as well. In a MELSEC Q/QnA series project they are the same as the function blocks TIMER_HIGH_FB_M and TIMER_LOW_FB_M. But in this case, they are no longer 10-ms and 100-ms timers. Instead the timer interval set by the parameters is valid.

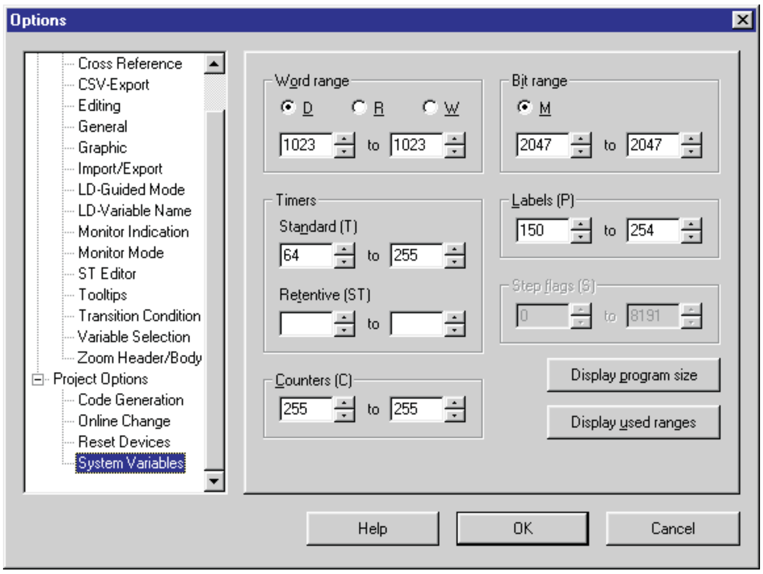
MELSEC A/AnS	MELSEC Q/QnA	Comment
TIMER_10_FB_M	TIMER_HIGH_FB_M	For the MELSEC Q/QnA series, it is not the intervals 10 ms and 100 ms which are valid, but the timer interval set by the parameters.
TIMER_100_FB_M	TIMER_LOW_FB_M	

Tab. 6-16: MELSEC Q/QnA timers

Defining the system range

You must define the system range before attempting to use local timers or counters. The system then automatically selects a free timer or counter from this range. A compile error will be displayed if you try to use these new function blocks without previously defining the corresponding system range.

① Open the System Variables dialogue box (**Extras – Options – System Variables**).



Word range: ● D ○ R ○ W — 1023 to 1023

Bit range: ● M — 2047 to 2047

Timers — Standard (T): 64 to 255; Retentive (ST): to

Labels (P): 150 to 254

Step flags (S): 0 to 8191

Counters (C): 255 to 255

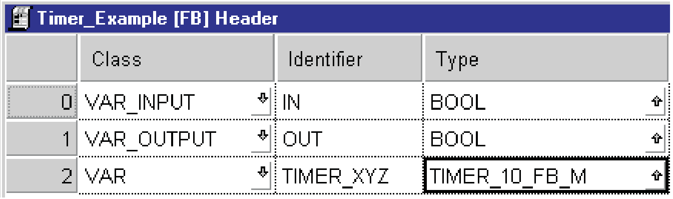
Fig. 6-67: *System timer System counter*

② Define the necessary system range in the System Timers or System Counters fields. This disables access to these addresses by other applications using direct addressing.

Instantiation

The use of these new function blocks is illustrated below using a 10 ms timer as an example.

① Declare the following variables in the header:

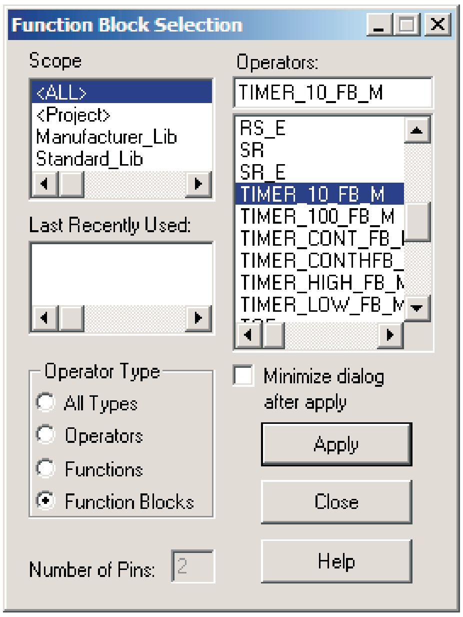


Timer_Example [FB] Header

	Class	Identifier	Type
0	VAR_INPUT	IN	BOOL
1	VAR_OUTPUT	OUT	BOOL
2	VAR	TIMER_XYZ	TIMER_10_FB_M

Fig. 6-68:
TIMER_XYZ is the instance of function block TIMER_10_FB_M Timer start address

② Select function block TIMER_10_FB_M in the body.



Operators: TIMER_10_FB_M

Operator Type: ○ All Types ○ Operators ○ Functions ● Function Blocks

□ Minimize dialog after apply

Number of Pins: 2

Fig. 6-69:
Selecting the timer function block

The timer function block has the following input and output variables:

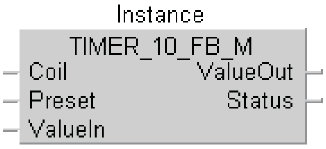


Fig. 6-70:
The timer function block TIMER_10_FB_M for 10 msec timers
TIMER_10_FB_M for 10-ms-Timer

Designation in the Function	Meaning	
Coil	Coil	Activation condition for the timer process
Preset	Set point value	
ValueIn	Initial value	(Normally: 0) ➜ Note
ValueOut	Actual value	
Status	Output contact	The output contact is set when execution of the function block is complete.

Tab. 6-17: *Timer function block designations*

③ Assign the instance names defined in the header to the function block and allocate the necessary input and output variables.

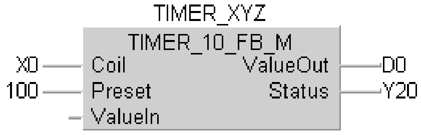


Fig. 6-71:
Instance TIMER_XYZ of function block TIMER_10_FB_M

NOTES

If you do not need one of the parameters for your application, e.g. ValueIn in the example, you can simply leave the contact unassigned. In this case, no program code will be generated.

For MELSEC FX series controllers the ValueIn parameter **must** be left unassigned!

6.11 The EN input and ENO output

Some programming instructions have an EN input and an ENO output in addition to the normal input and output variables. These instructions are identified in the standard library by the suffix _E, e.g. ABS_E, BCD_TO_DINT_E . The functions in the manufacturer library do not have this suffix as they are all automatically provided with an EN input and an ENO output.

Functionality

The EN input controls the execution of the instruction.

There are several different ways to connect the EN input:

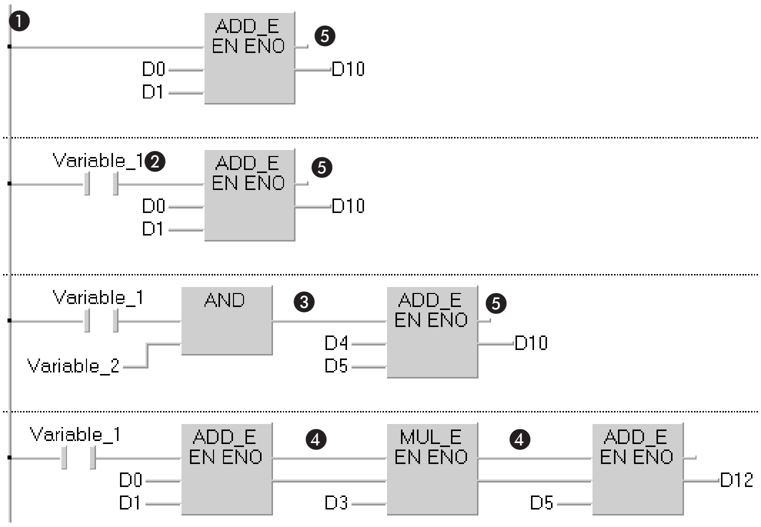


Fig. 6-72: Different EN input connections

Item	Description
❶	Direct connection of the EN input to the power bar at the left (LD only). The EN input is then always on and the instruction is always executed. When ADD_E is used like this it will produce exactly the same result as ADD without the EN input.
❷	Link the EN input to a Boolean input variable. The instruction is then only executed when Variable_1 is active.
❸	Connect the Boolean result of a an operation to the EN input. The instruction is then only executed when the result of the operation is TRUE.
❹	Connect the EN input to the ENO output of the preceding instruction. The three instructions are then only executed when Variable_1 is active.
❺	The ENO output does not have to be connected; it can be left unconnected if it is not needed.

Tab. 6-18: EN input items

6.11.1 Execution with and without EN and ENO

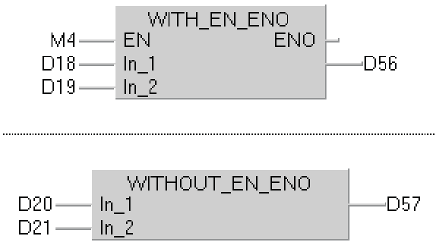


Fig. 6-73:
User-written function WITH_EN_ENO:

With EN input and ENO output

User-written function WITHOUT_EN_ENO:

Without EN input and ENO output

User function WITH_EN_ENO

Execution of the function is controlled by relay M4 (when the input EN is processed).

M4 = 1 (TRUE):

- The function is executed.
- The result of the operation is written to D56.
- A new value is always written to D56.

M4 = 0 (FALSE):

- The function is not executed.
- No new value is written to D56. The value stored in D56 remains unchanged, it is not reset.

User function WITHOUT_EN_ENO

- The function is executed continuously.
- The result of the operation is written to D57.
- A new value is always written to D57.

7 Guided Ladder Editor

7.1 Main Features

The following are the main features of the Guided Ladder Editor:

Implementation of an additional programming mode, **Guided Mode**.

- Guided ladder circuit editing
- **Character Mode** programming
- **Auto Comment** feature
- **Ladder Editing** Settings

Enhancement for the ease of ladder programming

- Additional functionalities
- User-friendly hotkeys
- Enhanced program component variable name display

Each item of the main features is explained in detail in the following sections.

7.1.1 Guided Mode

The **Guided Mode** is an additional ladder-programming mode considering the possibility of programming mainly using the keyboard, and to reduce the mouse operations.

Guided Editing (Guided ladder circuit editing), **Character Mode** (Character programming mode) and **Auto comment** functions are collectively known as **Guided Mode** and Guided Mode is just a name for the new ladder circuit-editing mode.

Guided Mode can be accessed from the following menu:

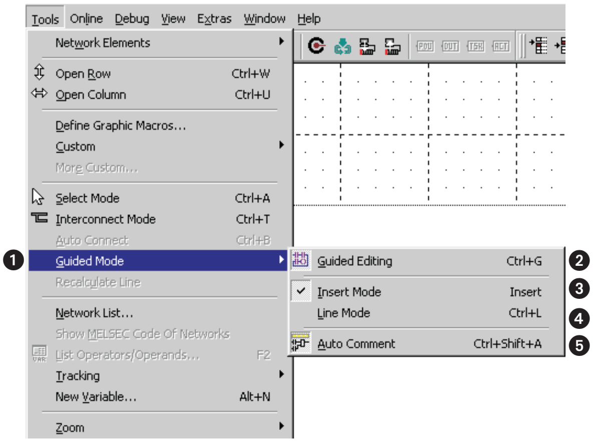


Fig. 7-1: *Menu **Guided mode***

No.	Explanation
❶	Guided Mode
❷	Guided editing
❸	Insert Mode
❹	Line Mode
❺	Auto Comment

Tab. 7-1: *Explanation for Fig. 7-1*

NOTE

The circuits created in **Select Mode or Inter connect Mode** are not affected when the user changes the current editing mode to **Guided Mode**.
The circuit created in **Select Mode** or **Inter connect Mode** can be edited in **Guided Mode**.
Character Mode is an internal functionality of **Guided Mode** and it cannot be accessed directly from the menu. But, **Guided Editing** and **Auto Comment** can be accessed from the user menu items.
Guided Editing and **Auto Comment** can also be accessed using their hotkeys and menu short cuts.

The following are the menu items and hotkeys related to **Guided Mode**.

No.	Menu Item	Hotkey
1	[Tools]-[Guided Mode]- [Guided Editing]	CTRL+G
2	[Tools]- [Guided Mode]- [Insert Mode]	INSERT
3	[Tools]- [Guided Mode]-[Line Mode]	CTRL+L
4	[Tools]-[Guided Mode]-[Auto Comment]	CTRL+SHIFT+A
5	[View]-[Colors]-[Overwrite Mode Cursor]	—
6	[View]-[Colors]-[Insert Mode Cursor]	—
7	[View]-[Colors]-[Line Mode Cursor]	—
8	[View]-[Colors]-[Cursor Fill]	—
9	[Tools]-[Network Elements]-[Delete Line]	CTRL + SHIFT + DEL

Tab. 7- 2: *Menu items for Guided mode*

7.1.2 Guided Editing Modes & Cursors

There are 3 types of Guided Editing Modes.

Insert Mode

Inserts a program component when **Insert** key is activated.

Overwrite Mode

Overwrites an existing program component when **Insert** key is not activated.

Line Mode

Draws connecting line between program components and handles the function block placement in a network.

The Guided Editing Modes and their associated cursors are as follows:

a) Normal cursor or Overwrite Mode cursor

The **Normal cursor** is a red (RGB = 255, 0, 0 by default) rectangle of 5 Grids of width and 4 Grids of height. The color of the cursor can be changed using the **View - Colors - Overwrite cursor** menu. The **View - Colors - Overwrite Cursor** will display a color selection dialogue same as the dialogue for changing the color of the Grid or Background in the previous versions of GX IEC Developer.

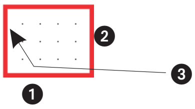


Fig. 7-2: *Normal cursor*

No.	Explanation
❶	5 Grids
❷	4 Grids
❸	Thickness of the cursor line is 4 Pts.

Tab. 7-3: *Explanation for Fig. 7-2*

b) Insertion cursor or Insert Mode cursor

The **Insertion cursor** is a blue (RGB = 0, 0, 255 by default) rectangle of 5 Grids of width and 4 Grids of height. The color of the cursor can be changed using the **View - Colors - Insert cursor** menu. The **View - Colors - Insert Cursor** will display a color selection dialogue same as the dialogue for changing the color of the Grid or Background in the previous versions of GX IEC Developer.

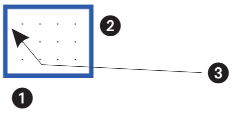


Fig. 7-3: *Insertion cursor*

No.	Explanation
❶	5 Grids
❷	4 Grids
❸	Thickness of the cursor line is 4 Pts.

Tab. 7-4: *Explanation for Fig. 7-3*

The **Grid cursor** is a minimized form of the **Normal cursor** or **Overwrite cursor**.

When the **Tools - Guided Mode - Line Mode** menu item is selected, the **Grid cursor** will appear at the top left corner of the current cursor position of the **Normal cursor**.

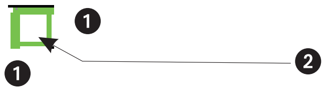


Fig. 7-4: *Grid cursor*

No.	Explanation
❶	1 Grid
❷	Thickness of the cursor line is 2 Pts.

Tab. 7-5: *Explanation for Fig. 7-4*

c) Grid cursor or Line Mode cursor

The **Grid cursor** is a green (RGB = 0, 255, 0 by default) square of 1 Grid size on each side. The color of the cursor can be changed using the **View - Colors - Grid cursor** menu. The **View - Colors - Grid Cursor** will display a color selection dialogue same as the dialogue for changing the color of the Grid or Background in the previous versions of GX IEC Developer.

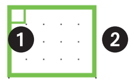


Fig. 7-5: *Grid cursor in Normal cursor*

No.	Explanation
❶	Grid cursor
❷	Normal cursor area

Tab. 7-6: *Explanation for Fig.7-5*

NOTE

The **Normal cursor** or **Overwrite cursor** will appear on the screen but it cannot be used when **Grid cursor** is displayed.
In the following sections the word “Cursor” implies that the **Normal** or **Insertion** cursor is used. It is not applicable to **Grid Cursor** if not mentioned explicitly.
Clicking a different location in the network can also move the cursor and it is applicable to all kind of cursors used in Guided Mode.
Displaying and hiding the cursor raster is possible using the **View - Grid** Menu.
The **Grid** menu can enable or disable both the Grid and Cursor raster.

Behaviour of Grid Cursor

Key	Operation
Arrow keys	Moves the **Grid cursor** according to the **Up, Down, Left, Right** arrow key pressed.
5	Draws a vertical line at the left side of the **Grid cursor**.
6	Draws a horizontal line at the bottom of the **Grid cursor**.
9	Places an input variable of a FB at the left side of the **Grid cursor**.
0	Places an input variable of a FB at the right side of the **Grid cursor**.
Delete	Deletes horizontal or vertical line that exactly fits with the size of Grid cursor

Tab. 7-7: *Behaviour of Grid Cursor*

Cursor movements & Keyboard shortcuts

The **normal cursor** or **insertion cursor** can be moved in 4 directions, left, right, top and bottom using the left, right, top and bottom arrow keys of the standard keyboard.

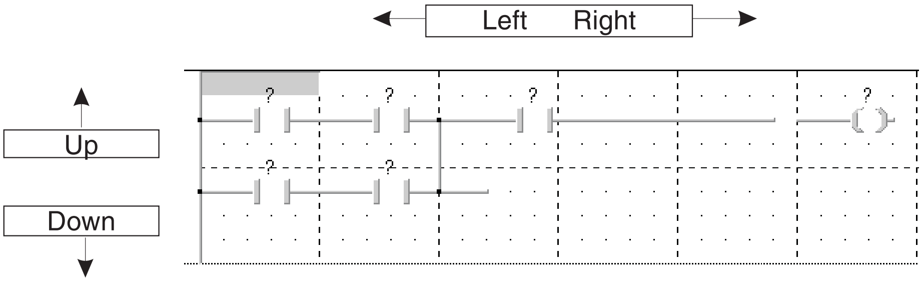


Fig. 7-6: *Cursor movements*

NOTE After selecting a function block, if the user presses the Space key and an arrow key, it will move the Function block by 1 grid position respectively to the arrow key pressed.

The other key operations with the **normal cursor** or **insertion cursor** are as follows:

Key	Operation
HOME	Move the cursor to the first element position of the row
END	Move the cursor to the last element position of the row, only if circuit wrapping is set
CTRL + HOME	Move the cursor to the first element position of the first row
CTRL + END	Move the cursor to the first element position of the last row
SHIFT+HOME	Select all grids from the current position to the first element position of the row.
SHIFT+END	Select all grids from the current position to the last element position of the row, only if circuit wrapping is set. This has no effect when circuit-wrapping position is not set.
SHIFT + <Arrow key>	Selects the consecutive program components from the current cursor location in the direction of cursor movement until the arrow key is released.
ENTER	Displays the instruction and the device name/variable name in the **Character Mode Input** dialogue.
CTRL + C or CTRL + INSERT	Copies the program components in the cursor selected area
CTRL + V or SHIFT + INSERT	Pastes the copied program components at the current cursor location
INSERT	Toggles between **Overwrite Mode** and **Insert Mode**
CTRL+L	Activates **Line Mode**
SPACE	Selects/deselects the function block, if the cursor is over FB.
SPACE + <Arrow key>	Moves the function block by 1 grid position in arrow key direction.
DELETE	Deletes the program component at the current cursor location, except function block or deletes the function block only if it is selected with SPACE key.
CTRL+DOWN	Adds a new row to the end of the current network
CTRL+UP	Deletes the last empty row of the current network

Tab. 7-8: *Keyboard shortcuts*

Selection of normal cursor or insertion cursor area

When some consecutive cursor areas are selected with SHIFT Key, the selected area will appear as shown below:

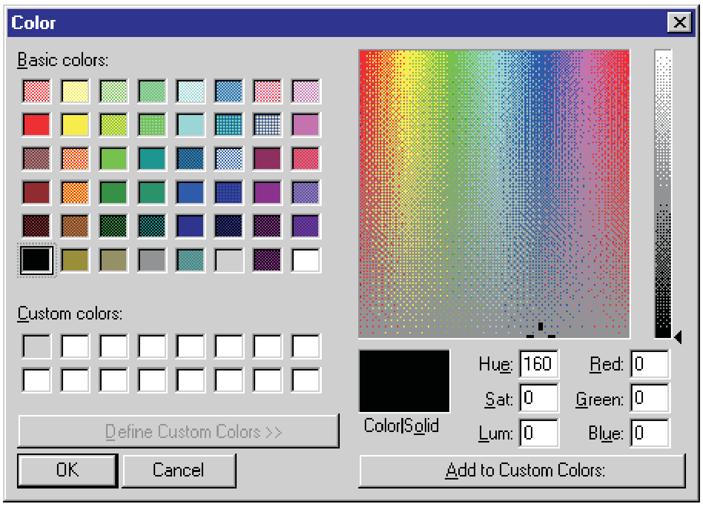
By default, solid fill with Blue color (RGB= 0,0,255),

Fig. 7-7: *Selected cursor area*

The user can change the color of the cursor using the **View - Colors - Cursor Fill** menu. The **View - Colors - Cursor Fill** will display a color selection dialogue same as the dialogue for changing the color of the Grid or Background in the previous version of GX IEC Developer.

Color Selection Dialogue

The Color selection dialogue same as the following will be displayed for all the cursor related color selection menu items. The user can change or set the color using this dialogue.



Hue: 160
Sat: 0
Lum: 0
Red: 0
Green: 0
Blue: 0

Fig. 7-8: *Color Selection Dialogue*

Cursor and the program components

Contact:

Horizontal lines will be added left and right of the contact, to fit the size of the cursor.

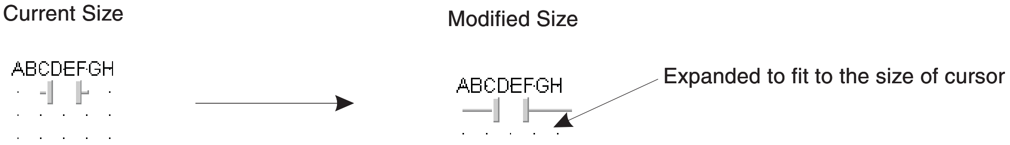


Fig. 7-9: *Contact*

Coil:

Horizontal lines will be added only at the left of the coil, to fit the size of the cursor.

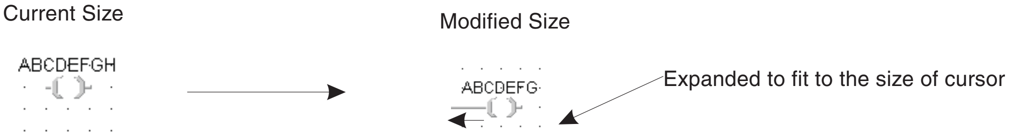


Fig. 7-10: *Coil*

NOTE

There is no horizontal line at the right of the coil to avoid compiler errors.

Function block:

There is no change in the size of the function block.

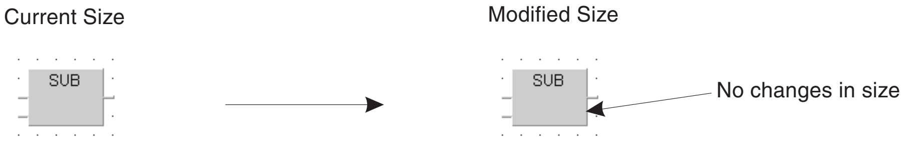


Fig. 7-11: Function block

Horizontal line:

A line with a width of a grid will be added at the right side of the horizontal line, to fit the size of the cursor.

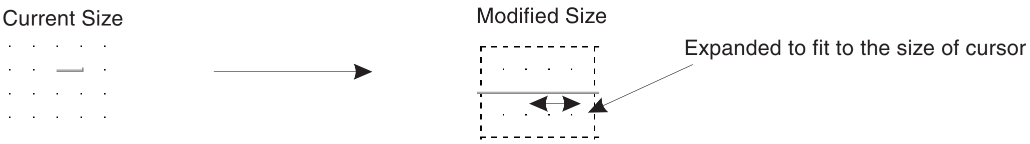


Fig. 7-12: Horizontal line

Vertical line:

A line of grid width will be added at the bottom of the vertical line, to fit the size of the cursor.

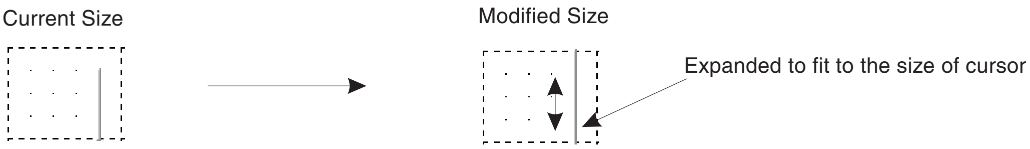


Fig. 7-13: Vertical line

NOTES

The above mentioned changes in the size of the program components do not change the sizes of the program components of other modes.

The expansion of the size of program components should be made as the method of adding lines with the program components in **Interconnect** or moving the program components when **Auto connect** feature is enabled, so that there is no difference between the current sizes of the program components, internally.

7.1.3 Guided editing

Guided Editing allows the user to edit a program using simple keyboard operations.

The main features of **Guided Editing** are:

- Programming using keyboard
- Automatic ladder circuit wrapping at the user defined location
- Selection of program components with keyboard
- Insertion and deletion of program components
- Positioning of function blocks

Programming using keyboard

In **Guided Mode**, a ladder circuit can be created, simply by pressing the hotkeys of the program components in a sequence. The program component will be placed at the current cursor location and after placing the program component the cursor will be moved to the next position, horizontally. If the next position is exceeding the user defined wrapping location, the cursor will be shifted to the position where the next program component can be placed.

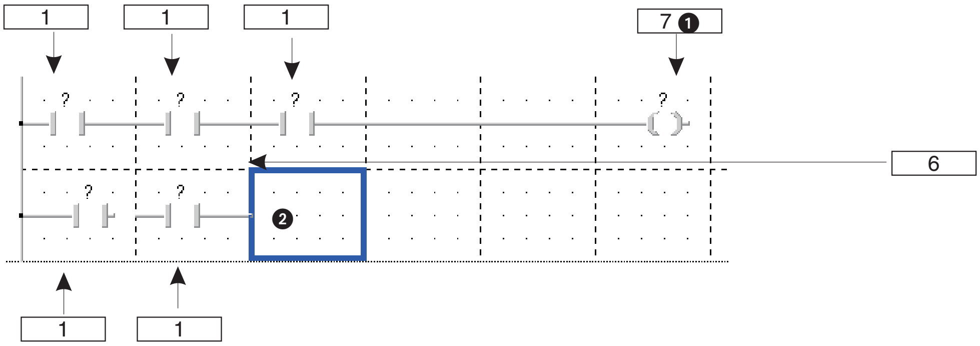


Fig. 7-14: Programming using keyboard

❶ Short cut key

❷ Current cursor position

The above circuit can be created in 3 steps as given below:

- Step 1: Press the keys 1-1-1-7 and 1-1 in sequence.
- Step 2: Move the cursor to the top of the vertical connecting line position.
- Step 3: Press the key 6.

NOTE

The above method is applicable when the user does not check the **Do not allow "?" as a variable name of Ladder Editing** settings. If it is checked, the user has to set the variable name using the ***V*ariable Selection** dialogue.

Automatic ladder circuit wrapping at the user defined location

The following operations will be carried out, if the ladder circuit wrapping is active.

- Wrapping of circuit:

If the ladder circuit exceeds the user-defined location, it will be wrapped. Refer to **Wrap after** setting in the **Ladder option** window.

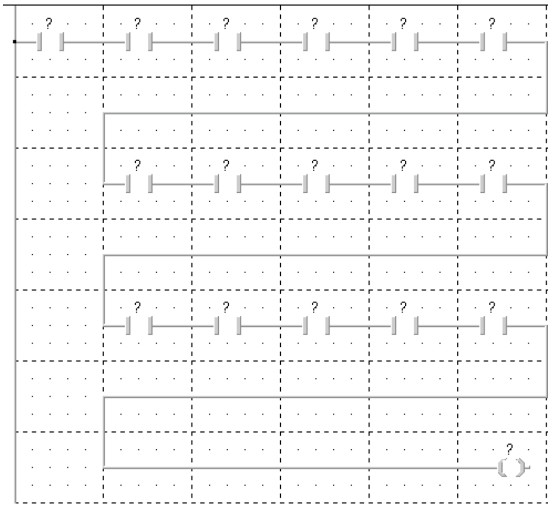


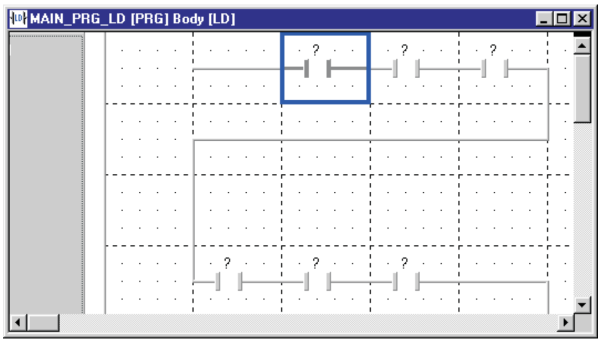
Fig. 7-15: Wrapping of circuit

NOTES

The wrapping is adapted from the current Inter connect mode or moving the program components when auto connect feature be enabled.

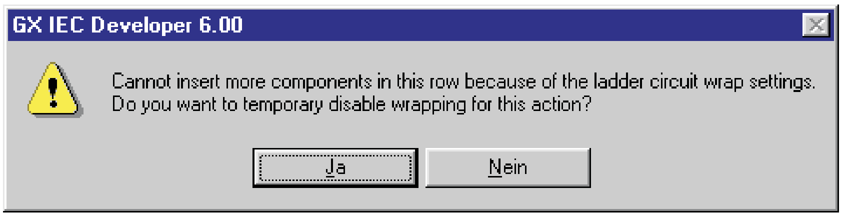
The user-defined wrapping location is valid only for the program component that is going to be placed in the network and it will not have any effect on pre-created networks or already positioned program components.

In active wrapping mode it is possible to shift the elements beyond the wrapping position. In the following example, a new contact is inserted. The user will be prompted to disable the wrapping for this action.



MAIN_PRG_LD [PRG] Body [LD]

Fig. 7-16: A new contact is inserted in an already filled line



GX IEC Developer 6.00

Cannot insert more components in this row because of the ladder circuit wrap settings.
Do you want to temporary disable wrapping for this action?

Fig. 7-17: Displayed warning

If the ladder circuit wrapping is temporarily disabled, the contact will be inserted although it exceeds the user-defined location:

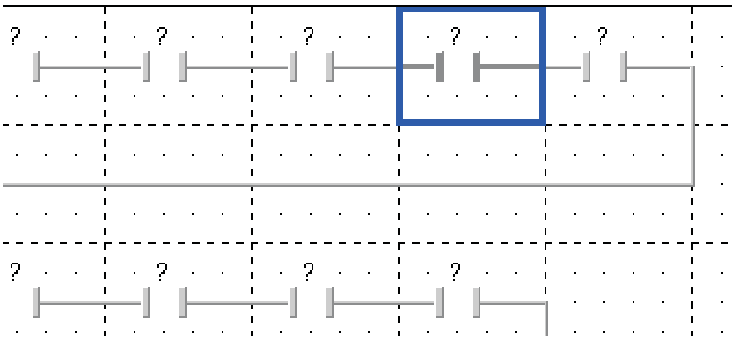


Fig. 7-18: *Insertion of a contact*

- Positioning of coil

If the ladder circuit does not exceed the user-defined location and it has a coil at the end, the coil will be positioned and placed according to the user-defined wrapping location.

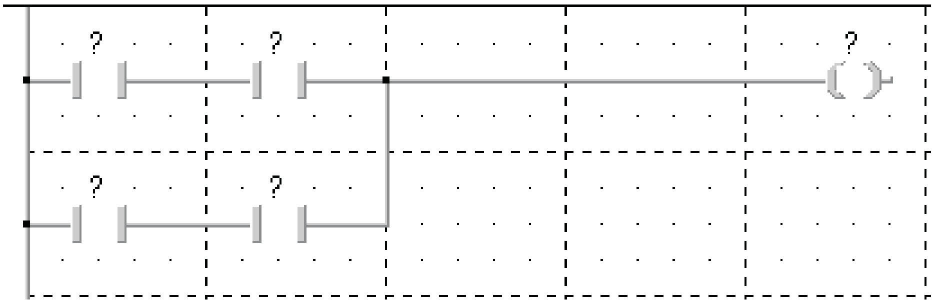


Fig. 7-19: *Positioning of coil*

Insertion and deletion of program components

A program component can be inserted or deleted after selecting it. Insertion and deletion of program components can be done as an Overwrite or as an Insertion according to the activation of the **Insert** key of the standard keyboard.

According to the activation of the **Insert** key, the menu item **Insert Mode** or **Overwrite Mode** of the Guided Mode menu will be checked or not checked.

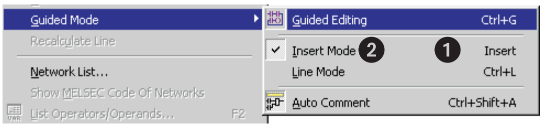


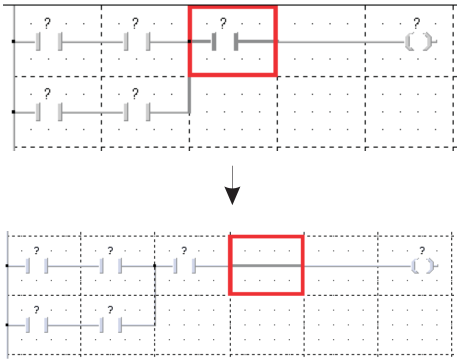
Fig. 7-20: *Guided Mode menu*

No.	Explanation
❶	No check mark on **Insert Mode** means that **Overwrite Mode** is selected
❷	Insert Mode – Activated by **Insert** key.

Tab. 7-9: *Explanation for Fig. 7-20*

Insertion and deletion with no Insert key activation - Overwrite Mode

- Insertion

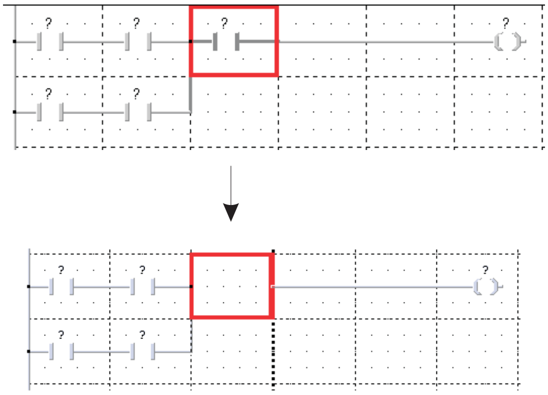


The cursor is here and 1 key is pressed to insert new contact.

The contact is overwritten.
The vertical line is not deleted.

Fig. 7-21: Insertion

- Deletion



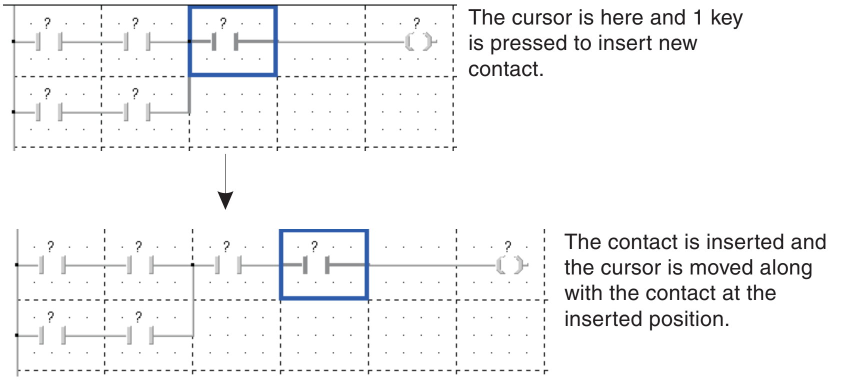
The cursor is here and **Delete** key is pressed to delete this contact.

The contact at this position is deleted.

Fig. 7-22: Deletion

Insertion and deletion with Insert key activation - Insert Mode

- Insertion

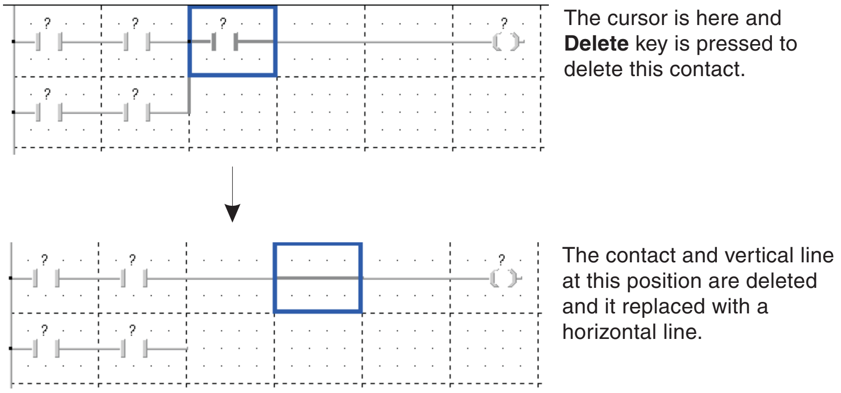


The cursor is here and 1 key is pressed to insert new contact.

The contact is inserted and the cursor is moved along with the contact at the inserted position.

Fig. 7-23: Insertion

- Deletion



The cursor is here and **Delete** key is pressed to delete this contact.

The contact and vertical line at this position are deleted and it replaced with a horizontal line.

Fig. 7-24: Deletion

Deleting single selected elements

All elements inside the cursor rectangle (except FB) are selected automatically. Specific selection of single elements is possible by pressing the SPACE-key. With each SPACE, the selection changes to the next element which is completely or partially inside the grid.

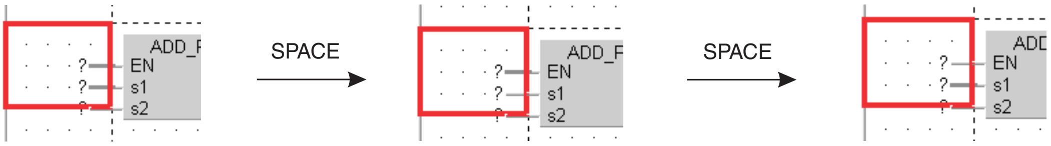


Fig. 7-25: Selection of a single element

Deletion of connecting lines

Deletion of only connecting lines is also possible using the **Delete Line** menu item under the menu **Tools - Network Elements**.

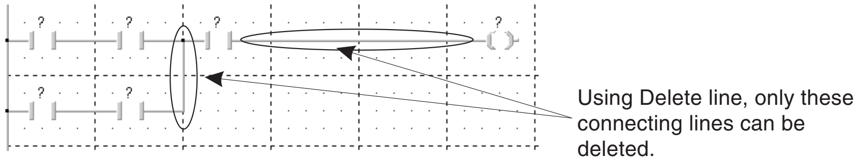

?
Using Delete line, only these connecting lines can be deleted.

Fig. 7-26: *Deletion of connecting lines*

After selecting the Delete Line menu item (not the delete key), if the cursor is moved over the network, the connecting lines will be deleted. Delete Line will not delete any program components.

Open Row and Open Column in Guided Mode

Open Row and **Open Column** menu items under the **Tools** menu can be used to add a new cursor row, cursor column respectively at the cursor position.

- Open Row:

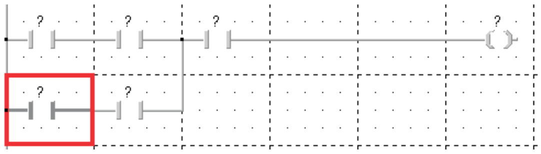


Fig. 7-27: *Open Row*

If **Open Row** is executed at the above cursor position, the circuit will be changed as shown below.

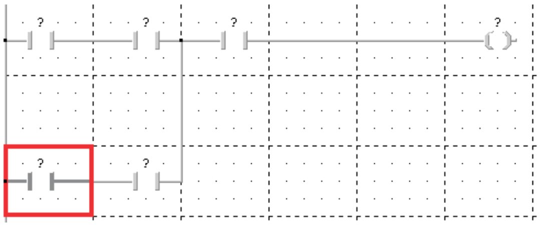


Fig. 7-28: *Open Row*

- **Open Column:**

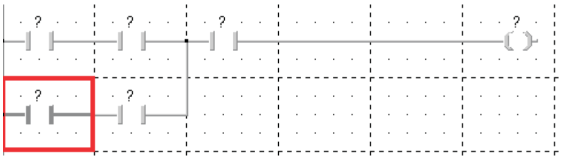


Fig. 7-29: Open Column

If **Open column** is executed at the above cursor position, the circuit will be changed as shown below.

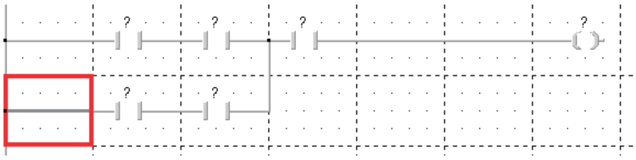


Fig. 7-30: Open Column

NOTES

During insertion and deletion with no **Insert** key activation, **Normal cursor** will be displayed. (Ref. Cursor for Guided Mode)

During insertion and deletion with **Insert** key activation, **Insertion cursor** will be displayed. (Ref. Cursor for Guided Mode)

The status of activation of the **Insert** key will be displayed on status bar. “**OVR**” will be displayed when **Insert** key is not pressed, “**INS**” will be displayed when **Insert** key is pressed and it will be empty for all other modes. User can add or delete the status panel of the OVR or INS display from the status bar, using the **View - Configure Status Bar** menu.

Deleting of multiple program components is also possible, if they are selected with the cursor.

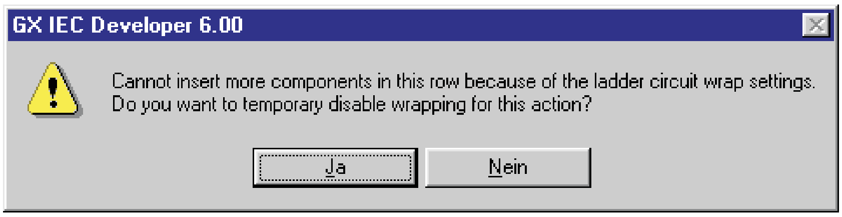
The insertion of program components at the last cursor position from the point of user defined wrapping location is not allowed. But the overwrite or the deletion of program component at the last cursor position from the point of user defined wrapping location is allowed.

Insertion & User-defined circuit wrapping :
In general, when a program component is inserted, the circuit will be expanded horizontally. But, the circuit cannot be expanded or wrapped after the user-defined wrapping settings.

The insert key activation and deactivation is common to all ladder networks in the project.

NOTES

Wrapping is not possible when inserting a program component, if the circuit has some user-defined location and there is no more space to insert a new component. In that case, a dialogue will be displayed to choose if the wrapping should be temporarily disabled.



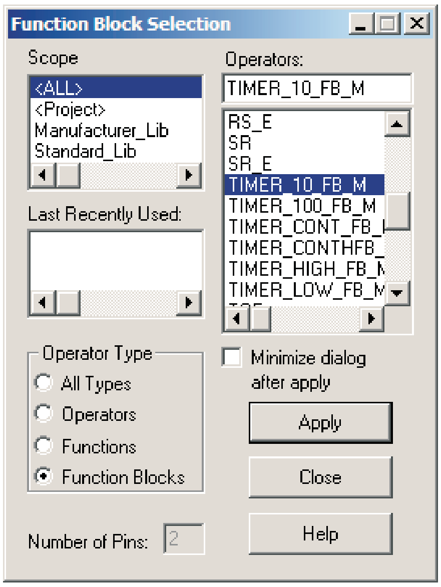
Cannot insert more components in this row because of the ladder circuit wrap settings.
Do you want to temporary disable wrapping for this action?

Open column is also not possible if the inserted column exceeds the circuit wrapping setting.

Refer to the **Wrap after** setting in the **Ladder option** window.

Positioning of function blocks

If the user tries to place a function block in the **Guided Mode**, the Function Block Selection dialogue will be displayed. The user has to select the function block from the **operators** list box and then press the **Apply** button to place a function block in the network.



Function Block Selection

Scope:
- <ALL>
- <Project>
- Manufacturer_Lib
- Standard_Lib

Last Recently Used:

Operators: TIMER_10_FB_M
- RS_E
- SR
- SR_E
- TIMER_10_FB_M
- TIMER_100_FB_M
- TIMER_CONT_FB_I
- TIMER_CONTHFB_
- TIMER_HIGH_FB_N
- TIMER_LOW_FB_M

Operator Type
- ○ All Types
- ○ Operators
- ○ Functions
- ● Function Blocks

□ Minimize dialog after apply

Apply | Close | Help

Number of Pins: 2

Fig. 7-31:
Function Block Selection

Minimum cursor area needed for a function block

For placing a function block, a minimum cursor area is needed. It is calculated as follow:

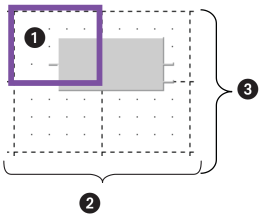


Fig. 7-32: *Minimum cursor area needed for a function block*

No.	Explanation
❶	Current Cursor location
❷	Required Width of FB (RWFB)
❸	Required Height of FB (RHFB)

Tab. 7-10: *Explanation for Fig. 7-32*

In general,

Required Height of Function Block = Actual Height of Function Block

+

cursor area at the bottom of function Block

Required Width of Function Block = Actual Width of Function Block

+

1 cursor area on both left and right of the Function Block

More precise, it can be calculated as given below:

Calculating Required Height of FB (RHFB)

If (AHFB Mod 5) not equal to 0

RHFB = (((AHFB – (AHFB Mod 5)) /5)+ 2) * 5

If (AHFB Mod 5) equal to 0

RHFB = ((AHFB /5)+ 1) * 5

Where,

AHFB = Actual Height of Function Block

RHFB = Required Height of Function Block

Calculating Required Width of FB (RWFB)

If (AWFB Mod 5) not equal to 0

RWFB = (((AWFB – (AWFB Mod 5)) /5)+ 3) * 5

If (AWFB Mod 5) equal to 0

RWFB = ((AWFB /5)+ 2) * 5

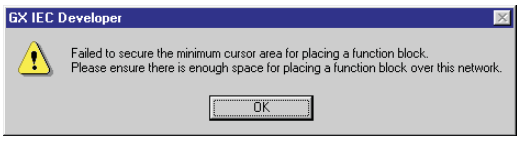
Where,

AWFB = Actual Width of Function Block

RWFB = Required Width of Function Block

NOTES

If there is no minimum cursor area in the circuit, the function block cannot be placed in a network. An error message as shown below will be displayed if there is no minimum cursor area.



Failed to secure the minimum cursor area for placing a function block.
Please ensure there is enough space for placing a function block over this network.

Consecutive placement of FB/FUN is possible, if the FB/FUN to be placed does not hide any element or exceeds the user defined network wrap settings.

Variable name of function block

The first 8 characters of the variable name of the function block will be displayed in the circuit. While placing the function block, it will be positioned automatically to secure the space for a 8 characters length variable name. The variable name should be scrolled to view the characters after the 8th position. The variable name will be displayed with “>”mark at the end (as shown below) if it exceeds 8 characters.

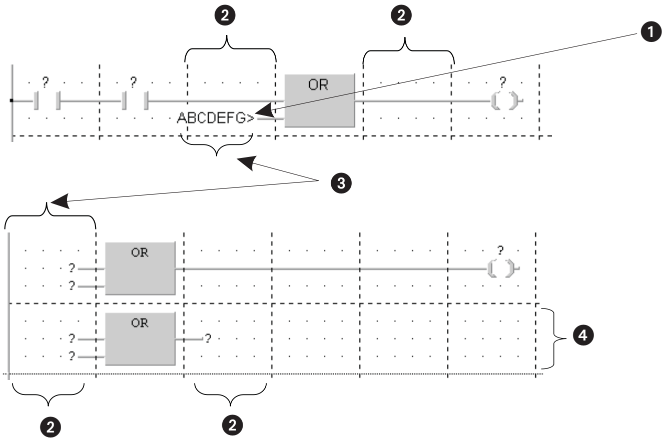


Fig. 7-33: *Variable name of function block*

No.	Explanation
❶	Exceeding 8 characters
❷	User has to create this part after placing FB
❸	Space for 8 characters size variable name
❹	A cursor row is deleted here to make FB in adjacent rows.

Tab. 7-11: *Explanation for Fig. 7-33*

NOTES

If there are no program components at the left or right side of a function block, the input variable or output variable will be added to the function block.

The input or output variable of a function block will be removed automatically, if the user places some connecting lines between the function block and other program component.

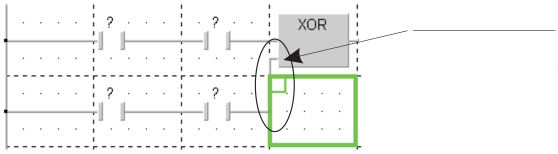
The horizontal connecting lines of circuit may get disconnected because of placing a function block in an existing circuit.

Function block related operations

No.	Operation	Implementation Details	
1	**Placement of FB**		
	At the Top Left corner of a network	1	It will be treated same as placing a function block by clicking at the Top Left grid of the network.
	At the Top Right corner of a network	1	It will be treated same as placing a function block by clicking at the Top grid row of the network.
		2	The current cursor location should be at 2 cursor locations prior to the Right end of the current cursor row.
	At the Bottom Left corner of a network	1	It will be treated same as placing a function block by clicking at the Bottom Left grid of the network.
	At the Bottom Right corner of a network	1	It will be treated same as placing a function block by clicking at the Bottom grid row of the network.
		2	The current cursor location should be at 2 cursor locations prior to the Right end of the current cursor row.
	At the left end or the first cursor column of the network	1	It will be treated same as placing a function block by clicking at the Left most grid of the network.
2	Moving a function block	1	The function block should be selected with **Space Key**.
		2	After selecting the function block, it can be moved using arrow keys while Space key is pressed. Each pressing of the arrow key will move the function block by one grid position.
3	Inserting a function block or placing FB next to a contact	1	It will be treated same as placing a function block by clicking at the Left most grid of the network.
4	Deleting a function block	1	The function block should be selected with **Space Key**.
		2	After selecting the function block, it can be deleted by pressing the **Delete key**.

Tab. 7-12: Function block related operations

Connecting Function block pins with some other program component



Connecting FB pins with the other part of a program component can be drawn using Grid cursor

Fig. 7-34: Connecting Function block pins

Placing the input and output variables of a function block

In **Line Mode**, the user can place the input or output variable of a function block using menu items or the hotkeys. The input or output variables of a function block cannot be placed in **Insert** and **Overwrite** modes.

Creating connected and unconnected crossing

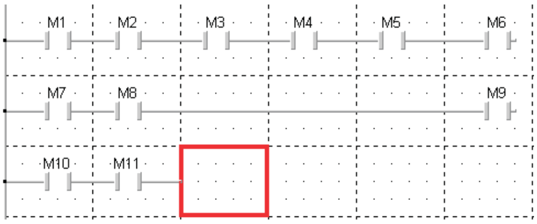
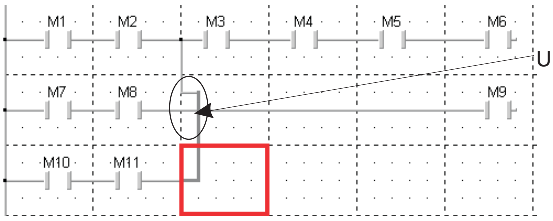


Fig. 7-35: *Start*

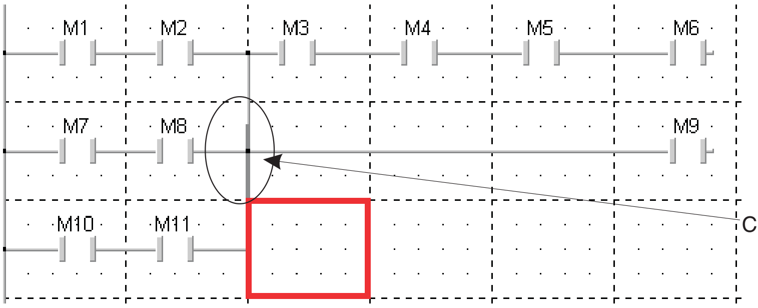
To make an unconnected crossing after M2 and M11, the **Vertical line** key (Hotkey 5) should be pressed after selecting the cursor area of contact M3, cursor area next to contact M8 and the cursor area next to contact M11 using the **Shift** and **Arrow** keys or dragging the cursor using the mouse.



Unconnected crossing

Fig. 7-36: *Unconnected crossing*

To make a connected crossing, the **Vertical line** key (Hotkey 5) should be pressed at two locations, at cursor area of contact M3 and cursor area next to contact M8.



Connected crossing

Fig. 7-37: *Connected crossing*

7.1.4 Character mode

The **Character Mode** is a function implemented inside the **Guided Mode**. It can be accessed by typing the direct instructions for the placement of program components in the **Guided Mode** and it cannot be accessed directly from the menu.

The placement of program components in a network is possible by typing the direct instructions. The program component will be inserted at the current cursor position.

The following is the list of instructions that can be used to access **Character Mode** in **Guided Mode**. (The items in bold letters in the following table are the instructions.)

No.	Instruction	Program Component	Remarks
1	**LD** Device/Variable name	Contact	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
2	**DI** Device/Variable name	Contact negation	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
3	**OUT** Device/Variable name	Coil	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
4	**FB** [Function block name]	Function Block	If a valid ‘Function block name’ is entered, the function block will be placed. Otherwise, the **Function Block Selection** dialogue will be displayed.
5	**HL** Count	Horizontal Line	Draws horizontal lines according to the value set in Count.
6	**VL** Count	Vertical Line	Draws vertical lines according to the value set in Count.
7	**LDL** Device/Variable name	L Connection Contact	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
8	**LDI L** Device/Variable name	L Connection Contact negation	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
9	**JMP** *Label*	Jump	
10	**IV** Device/Variable name	Input Variable	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
11	**OV** Device/Variable name	Output Variable	Program component with valid device/variable name can be entered directly. In case of unknown variable name, the **Variable Selection** dialogue will be displayed.
12	**RET**	Return	

Tab. 7-13: *Character mode*

Character Mode input

In **Guided Mode**, if the user presses any of the alphabet keys or just the **Enter** key, the **Character Mode Input** dialogue will be displayed.

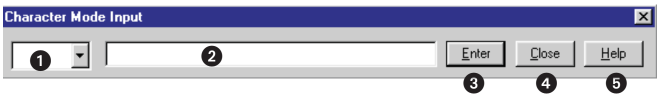


Fig. 7-38: *Character Mode input*

No.	Explanation
❶	Combo box for selecting the instruction
❷	Space for typing the instruction
❸	Update the contents
❹	Close the dialogue
❺	Instruction help information

Tab. 7-14: *Explanation for Fig. 7-38*

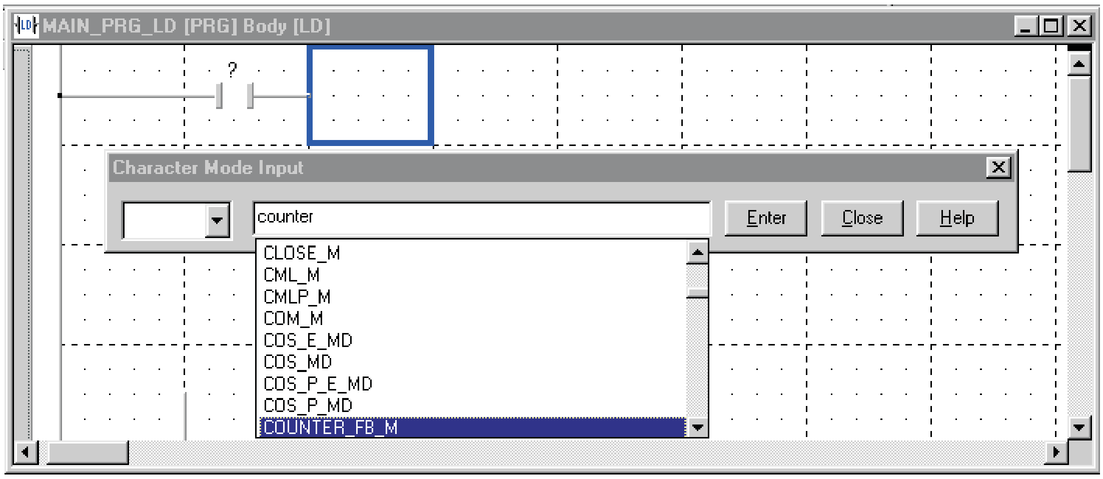
In the **Character Mode** the user can directly input instructions in any of the 2 methods:

- Select the instruction from the combo box and type the device/variable name.
- Type the instruction and device/variable name in the Edit box itself.

After inputting the direct instruction, the user has to press the **Enter** button or **Enter** key of the keyboard to update it to the network. To close the **Character mode input** dialogue, the **Close** button or ESC key of the keyboard should be pressed. Some help information to type the instruction will be displayed if the user presses the [?] button or F1 key of the keyboard.

The character mode input dialogue will be closed after every successful direct command input and it will appear again if the user presses any of the alphabet keys or just the **Enter** key.

If there is no instruction or ‘FB’ selected in the combo box and characters are typed into the edit field, a list opens automatically at the bottom of the edit field. This list selects the first function block, which matches the entered character combination. By pressing the cursor down key, the list can be entered as the proper function block and can be selected to the edit field by the ENTER key.



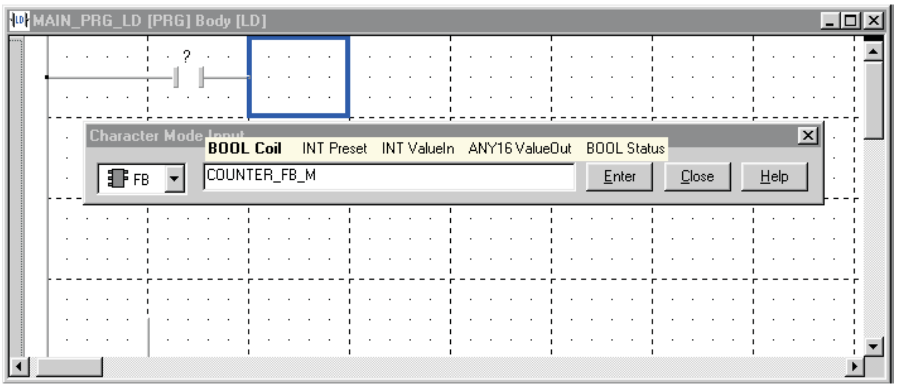
counter

CLOSE_M
CML_M
CMLP_M
COM_M
COS_E_MD
COS_MD
COS_P_E_MD
COS_P_MD
COUNTER_FB_M

Fig. 7-39: *Character mode input*

The function block name can be extended with all valid variable names, separated by SPACE. These variables are automatically added to the function block. If there are more input/output pins left than variables and 'Automatic Input/Output Variables' is active, '?' is selected as variable name. If the function block is placed next to a horizontal line, the corresponding input pin is connected to this line and an assigned variable is ignored.

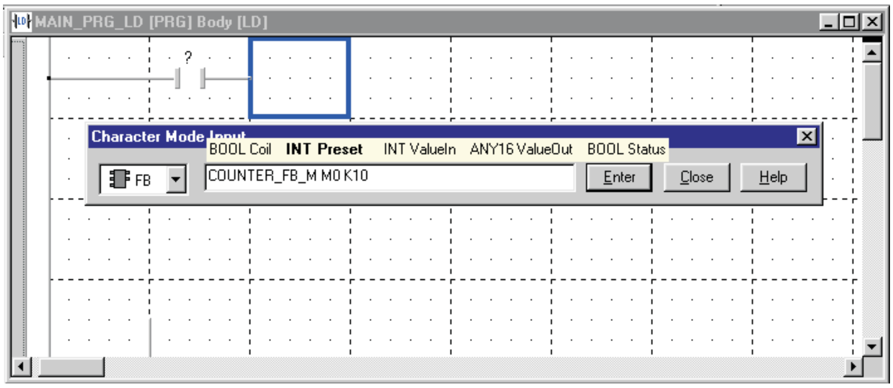
As soon as a valid function is recognised by a following SPACE, a ToolTip window appears, showing the variable definition from the corresponding header. The current edited variable will be highlighted in the ToolTip window.



BOOL Coil INT Preset INT ValueIn ANY16 ValueOut BOOL Status

COUNTER_FB_M

Fig. 7-40: Function block name



BOOL Coil **INT Preset** INT ValueIn ANY16 ValueOut BOOL Status

COUNTER_FB_M M0 K10

Fig. 7-41: Input of a variable

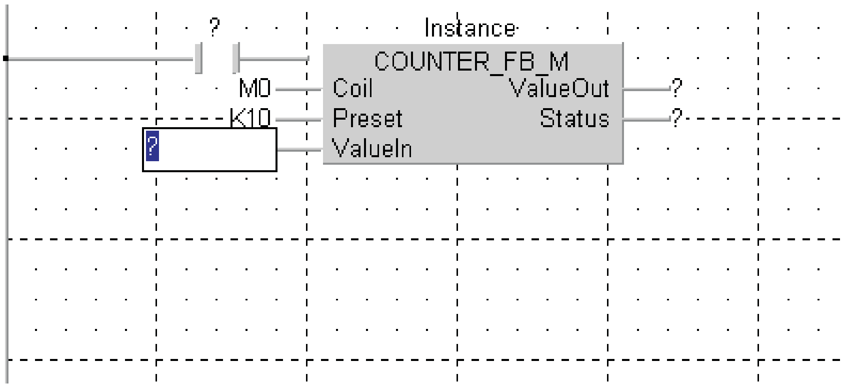
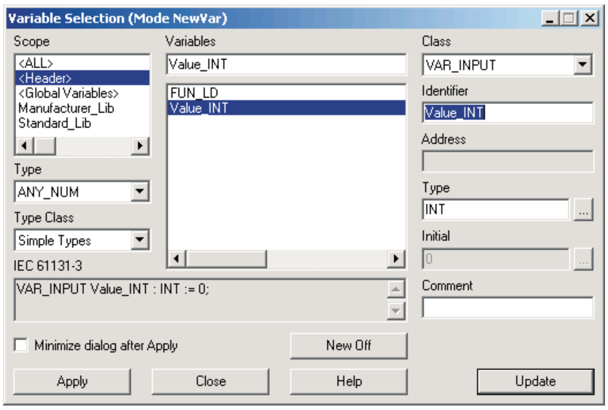


Fig. 7-42: Inserted function block

NOTE

If the variable name of the user typed instruction is not present in the Global, Local header settings, **Variable Selection** dialogue will be displayed automatically.
The user has to set the variable name using the **Variable Selection** dialogue before the program component is placed in the network.



Variable Selection (Mode NewVar)

Scope
<ALL>
<Header>
<Global Variables>
Manufacturer_Lib
Standard_Lib

Variables
Value_INT
FUN_LD
Value_INT

Type: ANY_NUM

Type Class: Simple Types

IEC 61131-3: VAR_INPUT Value_INT : INT := 0;

Class: VAR_INPUT

Identifier: Value_INT

Address:

Type: INT

Initial: 0

Comment:

□ Minimize dialog after Apply

Fig. 7-43:
Variable selection

NOTES

An error will be displayed if there is any error in the instruction, the device, the variable, or function block name typed in the **Character Mode Input** dialogue.

Variable selection dialogue will be displayed, if the user presses the F2 key after clicking the space for typing instructions of the Character Mode Input dialogue. The variable selected in the Variable Selection dialogue will appear in the Character Mode Input dialogue, if the user presses the Apply button of the Variable Selection dialogue.

If **Enter** key is pressed and there is some program component at the location of the **Guided mode** cursor, the Character Mode Input dialogue will be opened and the direct instruction for the program component will be displayed. (IV & OV instructions are available only in **Line Mode**.)

The user can overwrite or insert an element using Character Mode Input dialogue, but incase of FB it is possible only if there is a **Minimum cursor area** for a function block.

7.1.5 Auto comment

Auto Comment is an option available in **Guided Mode**. The purpose of Auto comment is to induce the user to insert a comment into the network that is created in **Guided Mode**.

The properties of **Auto Comment** are:

- It will insert a comment block at the beginning of a new network created using **Edit - New Network** menu item in **Guided Mode**.
- It will also insert a comment block at the beginning of the network if the placement of the very first program component is made in **Guided Mode**, even if network is created newly in **Select Mode** or **Interconnect Mode**.
- **Auto Comment** option does not have any effect on the networks created by placing the very first program component in **Select Mode** or **Interconnect Mode**. The comment blocks will not be added for those networks.
- The number of characters per Comment line depends on the **Auto Comment block width** value in the **Ladder Option settings**. The height of the comment block will be grown or expanded, if the number characters in a comment line exceed the **Auto Comment block width.**

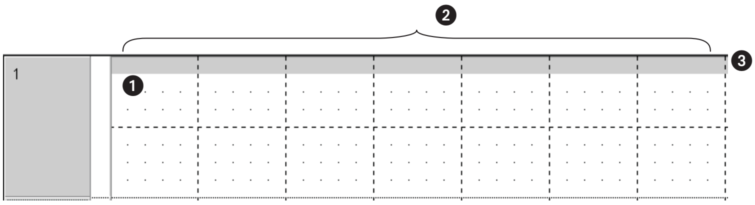
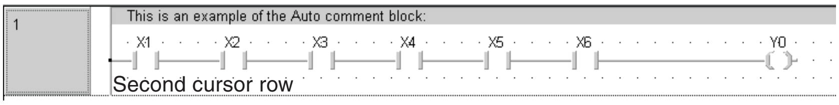


Fig. 7-44: *Comment block*

No.	Explanation
❶	Comment block
❷	Equal to the value set in the **Comment block width** settings
❸	Grow with Comment

Tab. 7-15: *Explanation for Fig. 7-44*

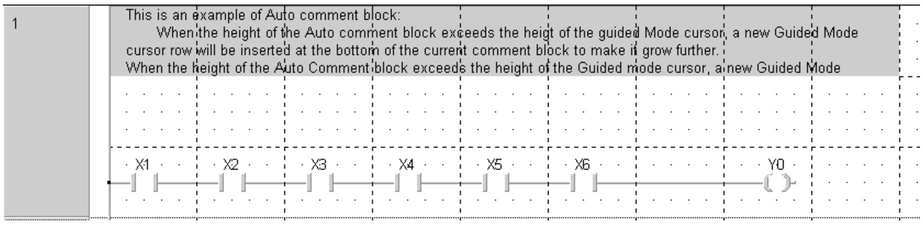
- **Auto comment** will be disabled if **Esc** key is pressed and no new auto comment blocks will be added for the new network created using **Edit-New Network** menu item or the placement of the very first program component is made in **Guided Mode**. It is effective only for the active project.
- **Auto Comment** will be enabled when the user selects **Guided Mode** in a new project.



This is an example of the Auto comment block:

Fig 7-45: *Second cursor row*

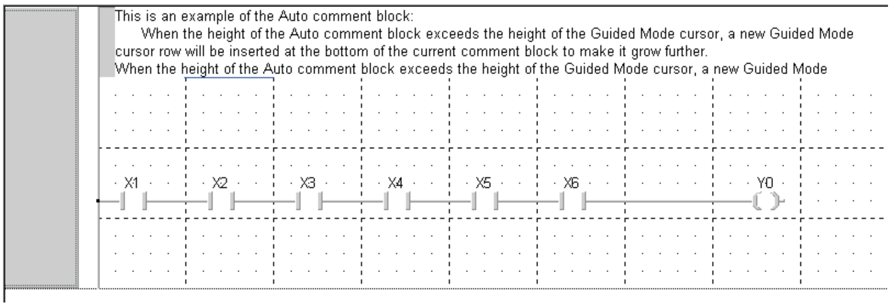
- When **Auto Comment** is enabled, the very first program component will be placed at the second **Guided Mode** cursor row from the top of the active network.
- When the height of the **Auto Comment** block exceeds the height of the **Guided Mode** cursor, a new Guided Mode cursor row will be inserted at the bottom of the current comment block to make it grow further.



This is an example of Auto comment block:
When the height of the Auto comment block exceeds the heigt of the guided Mode cursor, a new Guided Mode cursor row will be inserted at the bottom of the current comment block to make it grow further.
When the height of the Auto Comment block exceeds the height of the Guided mode cursor, a new Guided Mode

Fig. 7-46: *Height of the Auto Comment block*

- In **Guided Mode**, the user can edit an **Auto Comment** block by placing the **Guided Mode** cursor over it and pressing the **Enter** key.



This is an example of the Auto comment block:
When the height of the Auto comment block exceeds the height of the Guided Mode cursor, a new Guided Mode cursor row will be inserted at the bottom of the current comment block to make it grow further.
When the height of the Auto comment block exceeds the height of the Guided Mode cursor, a new Guided Mode

Fig. 7-47: *Placing the Guided Mode cursor*

- The maximum number of characters in a comment block is that of **Select Mode** and there is no change in **Guided Mode**.

7.1.6 Characteristics of guided mode

Changing in & out of Guided Mode

Changing into **Guided Mode** or changing out of **Guided Mode** to some other mode can be done easily with the mouse click on toolbar buttons or using the menu or menu short cuts.

Toolbar Button:

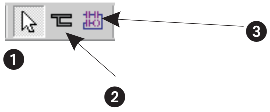


Fig. 7-48: *Icons*

No.	Explanation
❶	Select mode
❷	Interconnect Mode
❸	Guided editing

Tab. 7-16: *Explanation for Fig. 7-48*

Menu:

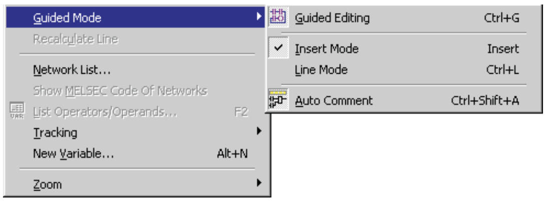


Fig. 7-49: *Menu*

Please refer to section 7.3 **User-friendly hotkeys** for the proper hotkeys of the above menu items.

Masked functions in Guided Mode

The functions available in other mode but not needed in **Guided Mode** will be masked and they cannot be used in **Guided Mode.**

The following functions will be masked in **Guided Mode** (in all **Overwrite**, **Insert** and **Line Mode**):

- [Tools]-[Auto Connect]
- [Tools]-[Recalculate Line]

The following functions will be masked in **Overwrite** and **Insert Mode** but available in **Line Mode**:

- [Tools]-[Network Elements]-[Input Variable]
- [Tools]-[Network Elements]-[Output Variable]

Functions specific to Guided Mode

The following menu items are specific to **Guided Mode**:

[Guided Mode]- [Insert Mode]

[Guided Mode]-[Line Mode]

[Guided Mode]-[Auto Comment]

[Tools]-[Network Elements]-[Delete Line]

Right Click Menu items

In Guided Mode, the user can access the following menu items using right click on the ladder editor window. The right click menu items are dependent of the current user selection Mode and will change accordingly.

Right click menu item	Remarks
Select Mode	Changes to the Select Mode
Interconnect Mode	Changes to the Interconnect Mode
Insert Mode	Toggles Insert and Overwrite Mode
Line Mode	Changes to Line Mode
Auto Comment	Toggles Auto Comment
——————————————-	A separator
Open Row	Inserts a new cursor row at the current cursor location
Open Column	Inserts a new cursor column at the current cursor location
Delete Line	Activate/deactivate the delete line function
Disable Circuit wrapping	Disables the circuit wrapping settings. (Displayed only when the user has made the circuit wrapping settings.)
Input Variable	Places an input variable of a function block. (Displayed only in **Line Mode**.)
Output Variable	Places an output variable of a function block. (Displayed only in **Line Mode**.)

Tab. 7-17: *Right Click Menu items*

NOTE

The menu item **Guided Editing** will be added with the current right click menu of the **Select Mode** or **Interconnect Mode**. The user can activate the **Guided Mode** from **Select** or **Interconnect Mode**, by clicking the **Guided Editing** right click menu item.

7.1.7 Editing a pre-created circuit in Guided Mode

The following circuit locations cannot be edited in **Guided Mode**. The **Guided mode** cursor can scroll over the location but does not have any effect on it.

A part of a multiline variable name

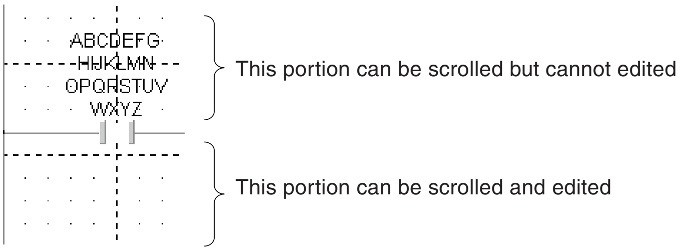

ABCDEFG
HIJKLMN
OPQRSTUV
WXYZ
This portion can be scrolled but cannot edited
This portion can be scrolled and edited

Fig. 7-50: *Editing a pre-created circuit*

The cursor can scroll over the variable name but the variable name can be edited only if the cursor is over a program component and not over a part of a variable name.

A part of a function block

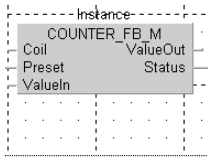


Fig. 7-51: *Scrolled function block*

NOTES The following operations can be done at the part of a function block.

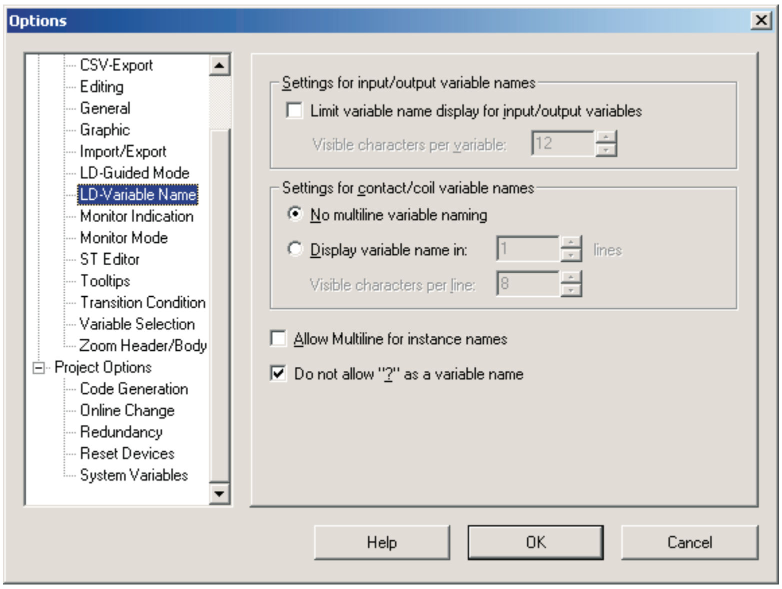
- The function block can be deleted, if it is already selected.
- Horizontal and vertical lines can be drawn using **Grid cursor**.

7.1.8 Ladder option settings

Ladder Editing Option settings can be accessed using **Extras - Options** menu.

LD-Variable Name option

The options related to Ladder variable name can be changed using the **LD-Variable Name** option.



Settings for input/output variable names

☐ Limit variable name display for input/output variables

Visible characters per variable: 12

Settings for contact/coil variable names

● No multiline variable naming

○ Display variable name in: 1 lines

Visible characters per line: 8

☐ Allow Multiline for instance names

☑ Do not allow "?" as a variable name

Fig. 7-52:
Options menu

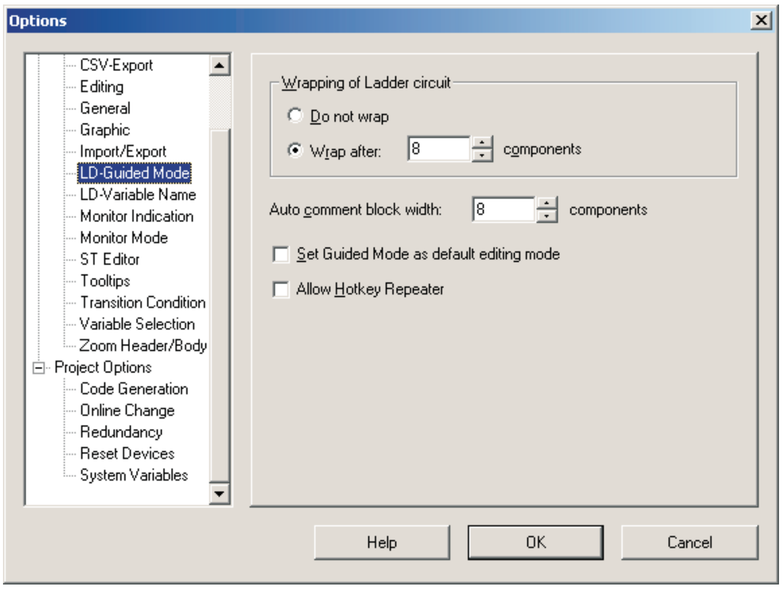
No Multiline variable naming: No multilane variable naming.

Display the variable name in n Lines: Wrap or fold the variable name in **n** (n=1 to 4 lines (maximum allowable variable length is 32 (4 x 8 characters)) lines in the network display.

Do not allow "?" as a variable name: Do not allow the user to give the "?" mark as variable name in **Guided Mode**. When this option is checked, the user must specify a proper variable name whenever a new program component is placed in the network.

LD-Guided Mode Option

The options related to Ladder Guided Mode can be changed using **LD-Guided Mode** option.



Wrapping of Ladder circuit

○ Do not wrap

● Wrap after: 8 components

Auto comment block width: 8 components

□ Set Guided Mode as default editing mode

□ Allow Hotkey Repeater

Fig. 7-53:
Options menu

Do not wrap: Do not wrap the ladder circuit. This setting is applicable for the circuit that is going to be edited after making this setting and not for the pre-created circuits.

Wrap after: Wrap if the ladder circuit exceeds the **n**th (n=2 to 100) component position. This setting is applicable for the circuit that is going to be edited after making this setting and not for the pre-created circuits.

Auto comment block width: Set the auto comment block with a width of **n** (n=1 to 100) components in the network.

Set Guided Mode as a default-editing mode: The project should be always opened in Guided Mode.

Allow Hotkey repeater: Hotkey repeater window will be displayed whenever the Horizontal or Vertical Line hotkey is pressed or accessed through the menu.

NOTE The above mentioned setting information will be stored in the .ini file.

7.2 Enhanced functionalities

7.2.1 Additional functionalities

The following are the additional functionalities:

Contact Negation

To place a negated Contact, so far the user had to change the **Signal Configuration** of the contact after placing it in the network. This additional functionality allows the direct placement of a negated Contact in the network.

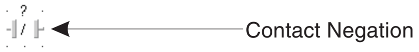


Fig. 7-54: *Contact Negation*

L-Connection Contact

To place a L-Connection Contact, so far the user had to draw the connections after placing the contact in the network. This additional functionality allows the direct placement of a L-Connection Contact in the network. In case of Guided Mode, the size of the vertical line depends on the variable name wrapping settings.

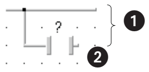


Fig. 7-55: *L-Connection Contact*

No.	Explanation
❶	This height will change with the multiline var. settings
❷	L-Connection Contact

Tab. 7-18: *Explanation for Fig. 7-55*

L-Connection Contact Negation

Similar to L-Connection Contact, this functionality is used to place a direct L-Connection Contact Negation in the network. In case of Guided Mode, the size of the vertical line depends on the variable name wrapping settings.

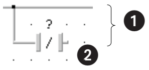


Fig. 7-56: *L-Connection Contact Negation*

No.	Explanation
❶	This height will change with the multiline var. settings
❷	L-Connection Contact

Tab. 7-19: *Explanation for Fig. 7-56*

7.2.2 Modified functionalities

The following functionalities that had been available in the last version of GX IEC Developer were modified as shown below.

Hotkey repeater for Horizontal line

When the user pressed the hotkey or menu item of a horizontal line, a horizontal line was placed at the mouse click location. Now the following dialogue is displayed before placing the horizontal line in a network.

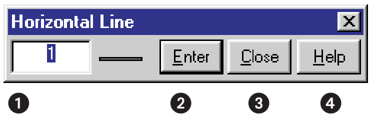


Fig. 7-57: Hotkey repeater for Horizontal line

No.	Explanation
❶	Count for Horizontal line
❷	Enter Horizontal line
❸	Close or cancel the dialogue
❹	Display some help information

Tab. 7-20: Explanation for Fig. 7-57

By default the count for Horizontal line is 1. But the user can draw more than 1 horizontal line from a cursor location, just by increasing the count value.

Hotkey repeater for Vertical line

When the user pressed the hotkey or menu item of a vertical line, a vertical line was placed at the mouse click location. Now the following dialogue is displayed before placing the vertical line in a network.

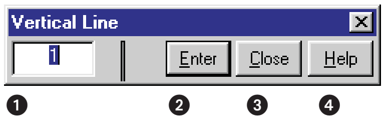


Fig. 7-58: Hotkey repeater for Vertical line

No.	Explanation
❶	Count for Vertical line
❷	Enter Vertical line
❸	Close or cancel the dialogue
❹	Display some help information

Tab. 7-21: Explanation for Fig. 7-58

By default the count for Vertical line is 1. But the user can draw more than 1 vertical line from a cursor location, just by increasing the count value.

7.3 User-friendly hotkeys

The following are the user-friendly hotkeys new to this version of GX IEC Developer.

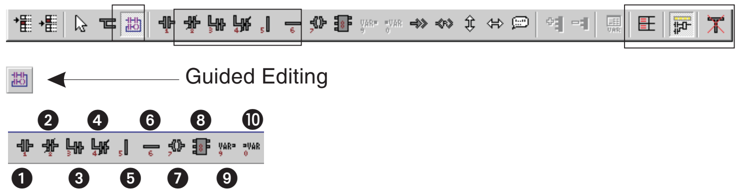


Fig. 7-59: Hotkey icons

No.	Symbol
❶	Contact (Hot Key = 1)
❷	Contact Negation (Hot Key = 2)
❸	L-Connection Contact (Hot Key = 3)
❹	L-Connection Contact Negation (Hot Key = 4)
❺	Vertical line (Hot Key = 5)
❻	Horizontal line (Hot Key = 6)
❼	Coil (Hot Key = 7)
❽	Function Block (Hot Key = 8)
❾	Input Variable (Hot Key = 9)
❿	Output Variable (Hot Key = 0)

Tab. 7-22: Explanation for Fig. 7-59

Location of the new toolbar buttons in the tool bar

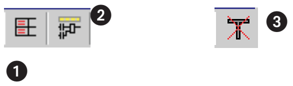


Fig. 7-60: Icons

The above hotkeys for program components are revised or allocated according to the MEDOC standards.

No.	Description
❶	Auto comment
❷	Label of the network
❸	Delete Line

Tab. 7-23: Explanation for Fig. 7-60

The **Label/Title** dialogue that was available in the last version of GX IEC Developer is now displayed if the user presses the **Label of the network** button.

Menu and the Hotkeys

Tools - Network Elements

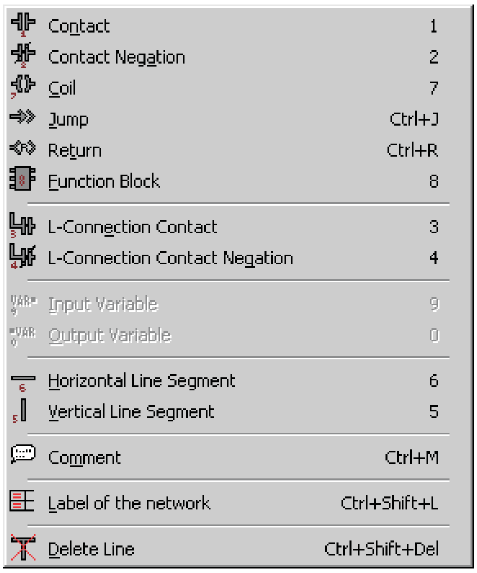


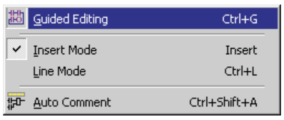
Fig. 7-61: *Menu*

The following menu items can be accessed with both the old and new hotkeys:

Menu Item	Old Hotkey	New Hotkey
Function Block	Ctrl + F	8

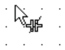
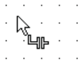
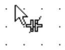
Tab. 7-24: *New hotkeys*

Tools - Guided Mode



New Toolbar buttons and Mouse cursors

The following mouse cursors are displayed when the new toolbar buttons or the new menu items are accessed. But there is no mouse cursor displayed for the functions related to **Guided Mode** and for functions that do not need a specialized mouse cursor for their operations (e.g. Label of a network).

No.	Function Name	Mouse cursor	Remarks
1	Contact Negation		
2	L-Connection Contact		
3	L-Connection Contact Negation		
4	Horizontal Line Segment	Same as GID Ver. 5	Only the tool button is added
5	Vertical Line Segment	Same as GID Ver. 5	Only the tool button is added
6	Guided Editing	No mouse cursor needed	
7	Label of a network	No mouse cursor needed	
8	Auto Comment	No mouse cursor needed	
9	Delete Line	No mouse cursor needed	

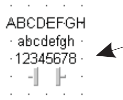
Tab. 7-25: *New Toolbar buttons and Mouse cursors*

7.3.1 Enhanced program component variable name display

For a proper display the variable name of the program components in a network had to be adjusted manually. Now the variable names will be arranged automatically before they are displayed.

Multiline variable naming

The variable name displayed above a program component can be wrapped/folded according to the **Display variable name in** ** **lines** setting of the Ladder Editing settings.



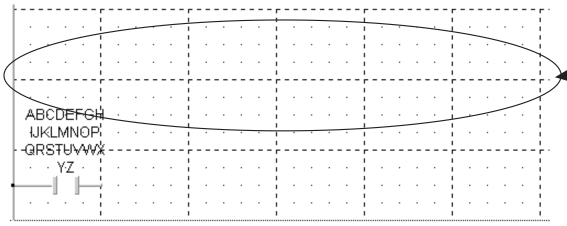
A lengthy variable name,
Example:
[ABCDEFGHabcdefgh12345678XYZ]
will be displayed like this, if it is made to be
displayed in 3 lines

Fig. 7-62: *Multiline variable naming*

The rules for variable name wrapping are:

- Each line in the variable name can accept a maximum of 8 characters
- The remaining part of the variable name that does not appear in the display can be viewed by scrolling the variable name. (In the above example the last 3 characters [XYZ] of [ABCDEFGHabcdefgh12345678XYZ] can be viewed by scrolling it.)

Multiline variable naming and the Behaviour of Guided Mode cursor



Cursor row

Fig. 7-63: *Cursor row*

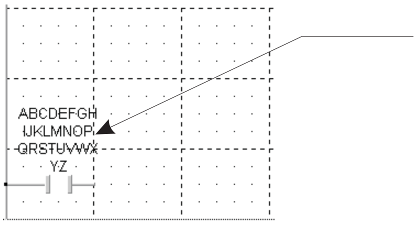
The cursor can scroll over the variable name but the variable name can be edited only if the cursor is over a program component and not over a part of a variable name. But the user can edit the part of a multiline variable name in **Select** or **Inter connect** mode.

NOTE

If multiline variable name is allowed/selected and the variable name has more than 8 characters, a new cursor row will be added above the current cursor.

Alignment of variable name in Guided Mode

All variable name displays are centre aligned in each line and the variable name is bottom aligned in the variable name box. If the variable name is 27 characters long and the „Display variable name in **n** lines“ is set to 8, the variable name will be displayed as shown below.



Space for 4 lines of variable name and no program component is allowed to place over this row.

Fig. 7-64: *Alignment of variable name*

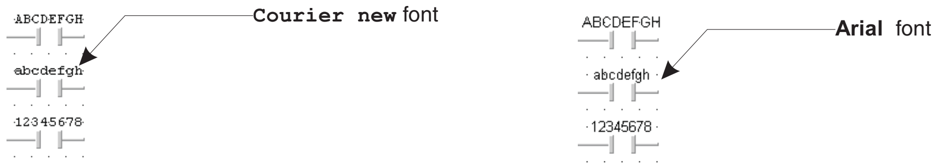
NOTES

Regardless of the mode of ladder editing, the variable name display is centre aligned in each line and the variable name is bottom aligned in the variable name box.

The width of the variable name display changes with the type of the font set for the Body. So to implement the enhanced display of the variable name, the font for the Body must be set with a fixed size font like **Courier new** font always.

Example ▽

The width of the variable name changes with the font.



Width of the variable name display is same for all cases

Width of the variable name display is not same for all cases

Fig. 7-65: *Example*

△

About scrolling a multiline variable name:
No scrollbars appear inside the Edit control. The user can scroll down until reaching the end of the variable name. Behind the end of scrolling region the cursor inside the edit box will not move down any further. Since the default settings for the variable name wrapping is 3 lines (3 x 8 characters), even if the user sets a variable name of 32 characters length, scrolling down only two lines is possible.

7.3.2 Ladder circuit printout

In general, there are no changes in the printing settings of the current GX IEC Developer.

Regarding the printing of a ladder circuit, the following point should be taken care of:

Since the **Courier New** font is used for creating the ladder circuit in **Guided Mode**, there is a necessity to maintain the same font in printing also. In **Guided Mode**, if the user changes the default font settings from Courier New font to some other font, the variable names in the ladder circuit may get overlapped on both the computer screen as well as in the printout.

8 Compiling and Downloading Programs

8.1 Error Check

Before compiling your project it is advisable to perform an error check on the project or individual objects. The following checks are carried out:

Header

Checks of all the variable declarations (unique descriptors, addressing etc).

Body

The following checks are performed in the editors:

Graphical editors

- Validity of the data types
- Variable names (declarations)
- Validity of the IEC addresses
- Validity of the MITSUBISHI addresses
- Definition of the input variables
- Recursions
- Interconnects
- Label syntax

Text editors

- Syntax of the operators and operands
- Number of operands
- Recursions
- Jumps
- Comments

Errors and warnings are displayed in the **Compiler/Check Messages** status window with explanations.

NOTE

When you double-click on an error message or warning in the status window GX IEC Developer automatically opens the corresponding object, moves the point in the program where the error was found and highlights the error in a different colour. The jumped to position in for example an editor of IL, MELSEC IL or ST may not be the exact position of the error cause. Look for the error cause in the jumped to line or in the lines above.

How to perform an error check on an object

① Select **Check** in the **Object** menu.
Or:
Click with the left mouse button on the Check tool:

NOTE

> You do not have to save your editing changes in open objects before selecting the Check option. If changes have been made the object is saved automatically before Check is executed.

How to specify the max. number of errors and warnings

① Select the option **Stop Check By** in the **Extras - Options - General** menu.

② Enter the maximum number of errors and warnings. The check operation will be stopped when the number of errors and warnings exceeds the limit specified here. Range: 0 - 2,000.

8.1.1 Execution order

In some cases the execution order in a ladder or FBD network is not clear, for example if complex connections between different elements are established.

The scanner for ladder / FBD networks recognises these critical networks. In this case a warning is indicated in the dialogue **Compile / Check Messages**.

On demand the generated MELSEC IL code of the respective network will be displayed if the POU has been compiled. To do so take the following steps:

① Select one or multiple networks.

② Select the menu item **Tools** → **Show MELSEC Code Of Network**.

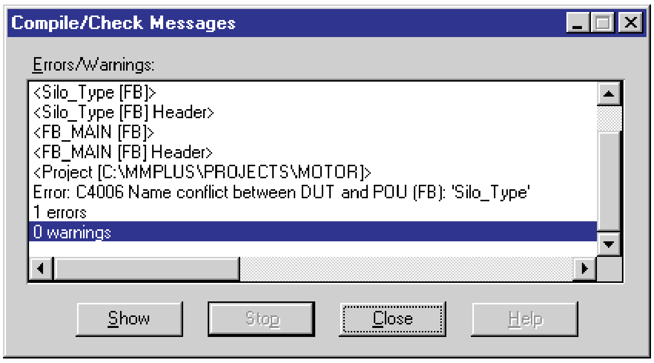
8.2 Compiling

The compilation process translates the project written in GX IEC Developer into code that the PLC CPU can execute. This translation process must be performed before downloading the program to the PLC system.

How to compile the complete project

Select **Rebuild All** in the **Project** menu.

GX IEC Developer first performs checks on all the objects in the project and displays any errors or warnings in the **Compiler/Check Messages** status window. If no errors are displayed this means that the program code has been compiled without errors.



```
<Silo_Type [FB]>
<Silo_Type [FB] Header>
<FB_MAIN [FB]>
<FB_MAIN [FB] Header>
<Project [C:\MMPLUS\PROJECTS\MOTOR]>
Error: C4006 Name conflict between DUT and POU (FB): 'Silo_Type'
1 errors
0 warnings
```

Fig. 8-1:
Compiler/Check Messages

Once the project has been compiled, the **Compiler/Check Messages** status window can be opened any time via the menu item **Window - Show Compile/Check Messages**.

In the status window occurred errors and warnings can be browsed via the following **Window** menu items:

- **Show Next Error** Hotkey: **[F4]**
- **Show Previous Error** Hotkey: **[SHIFT] + [F4]**
- **Show Next Warning**
- **Show Previous Warning**

How to compile only modified objects

POUs that have not yet been compiled are identified in the Project Navigator with an asterisk. POUs marked in this way were either created after the last compilation or are not part of a task.

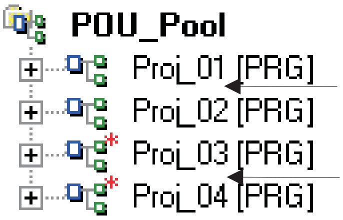


Fig. 8-2:

Already compiled

Not yet compiled

① Select **Build** in the **Project** menu.

Only the new objects which have not yet been compiled and the project elements which depend on the changes will be compiled.

NOTE

When you double-click on an error message or warning in the status window GX IEC Developer automatically opens the corresponding object, moves the point in the program where the error was found and highlights the error in a different colour.

WARNING:
Please note that only those Program POUs that you have included in your tasks are checked and compiled!

Earlier MM+ versions

You must completely recompile projects generated with earlier versions of MM+ (**Compile Project**) before you can use the **Build** function on them to compile only changes made in the editing session.

8.3 Downloading Projects to the CPU

8.3.1 Transfer Setup - Ports

You can only download compiled projects to the CPU. Before making your first download, check that the port and its parameters are configured correctly.

NOTE See chapter 2 and the manuals of your MELSEC hardware for details on connecting the controller to your computer.

How to configure the communications port

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

The **Transfer Setup** menu has moved to the top of the **Online** menu.

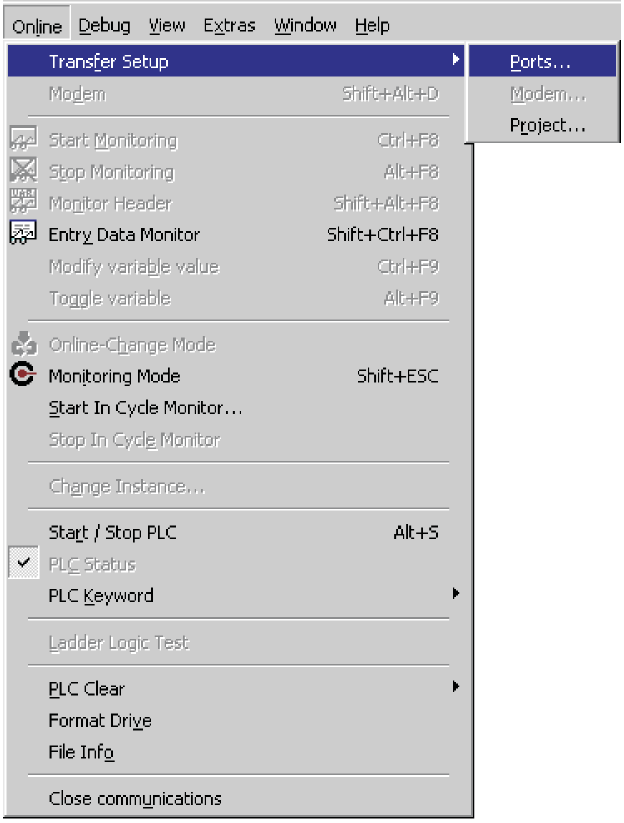
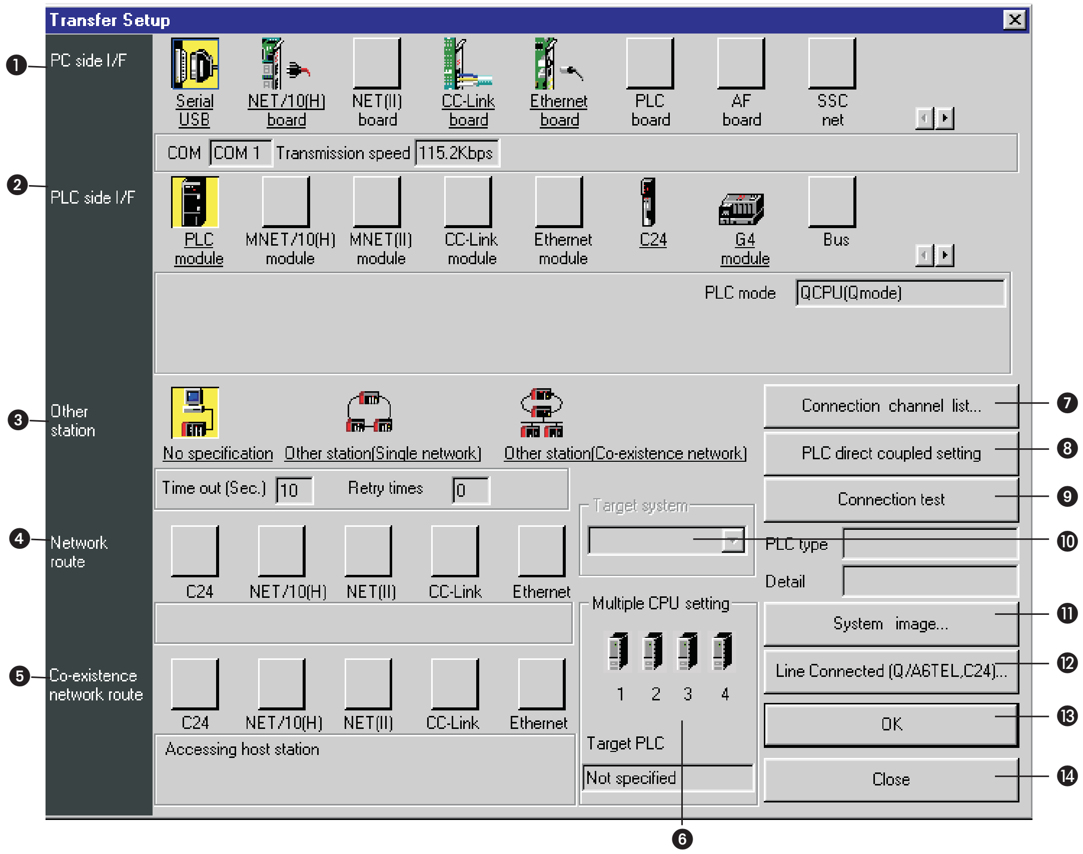


Fig. 8-3: *Transfer Setup*

A dialogue box will be opened with the following options:



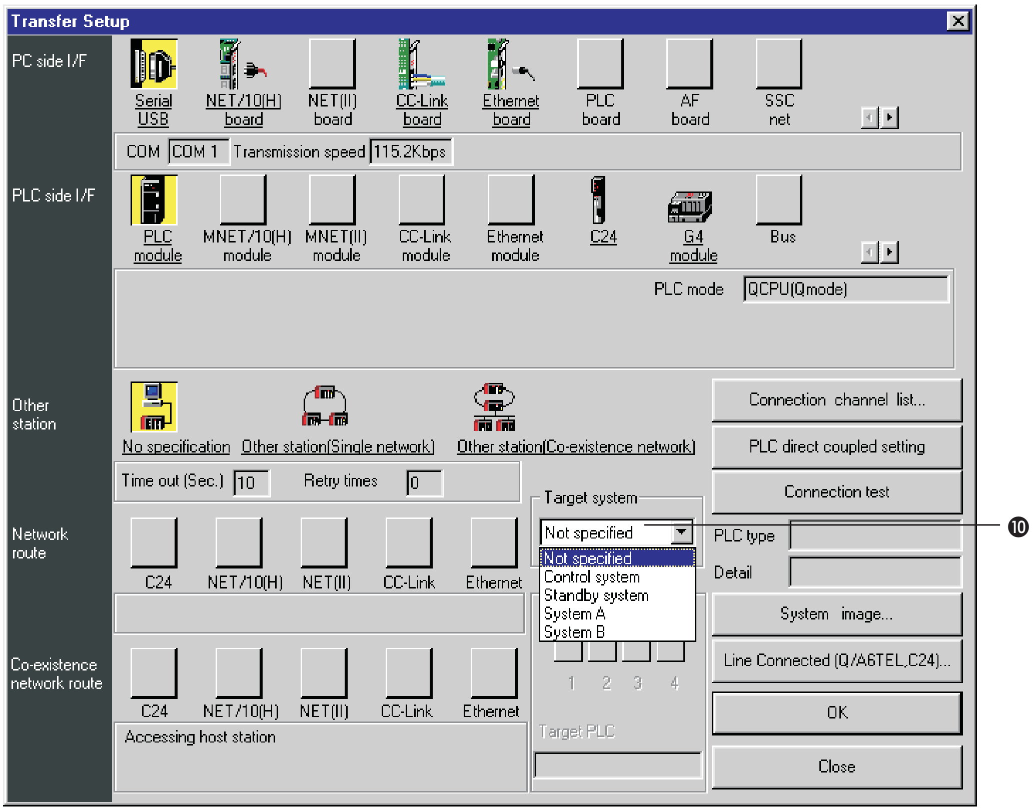
COM: COM 1 Transmission speed: 115.2Kbps

PLC mode: QCPU(Qmode)

Time out (Sec.): 10 Retry times: 0

Target PLC: Not specified

Fig. 8-4: *Transfer Setup (1)*



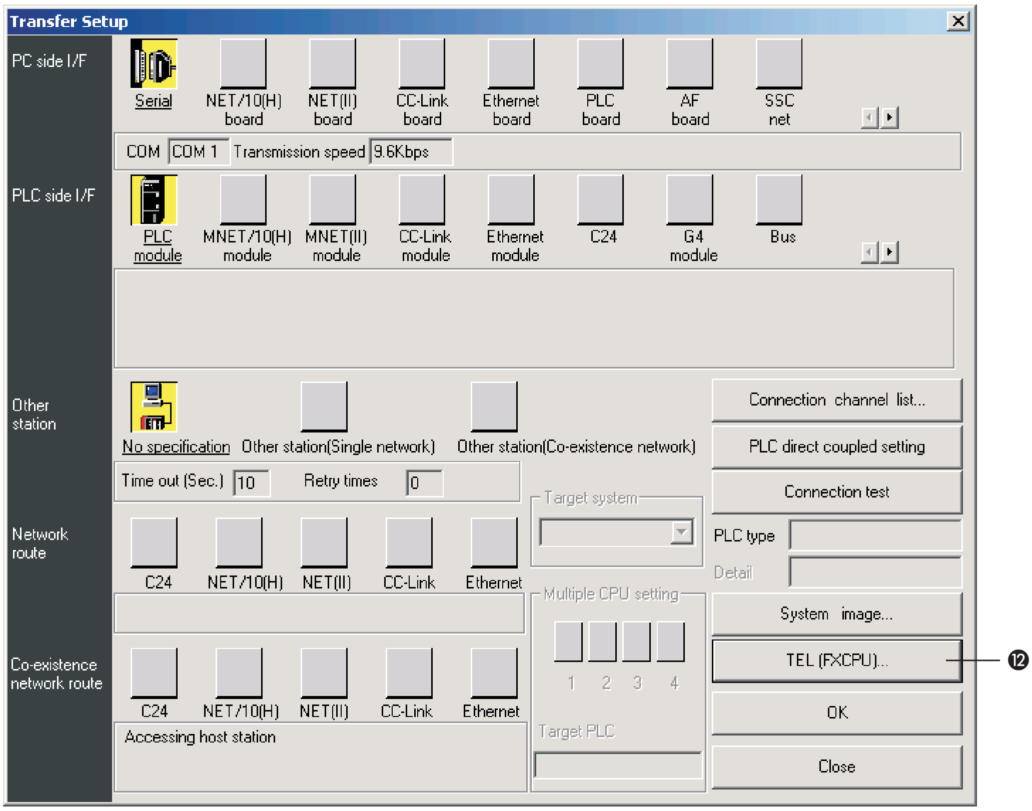
COM: COM 1 Transmission speed: 115.2Kbps

PLC mode: QCPU(Qmode)

Time out (Sec.): 10 Retry times: 0

Target system: Not specified

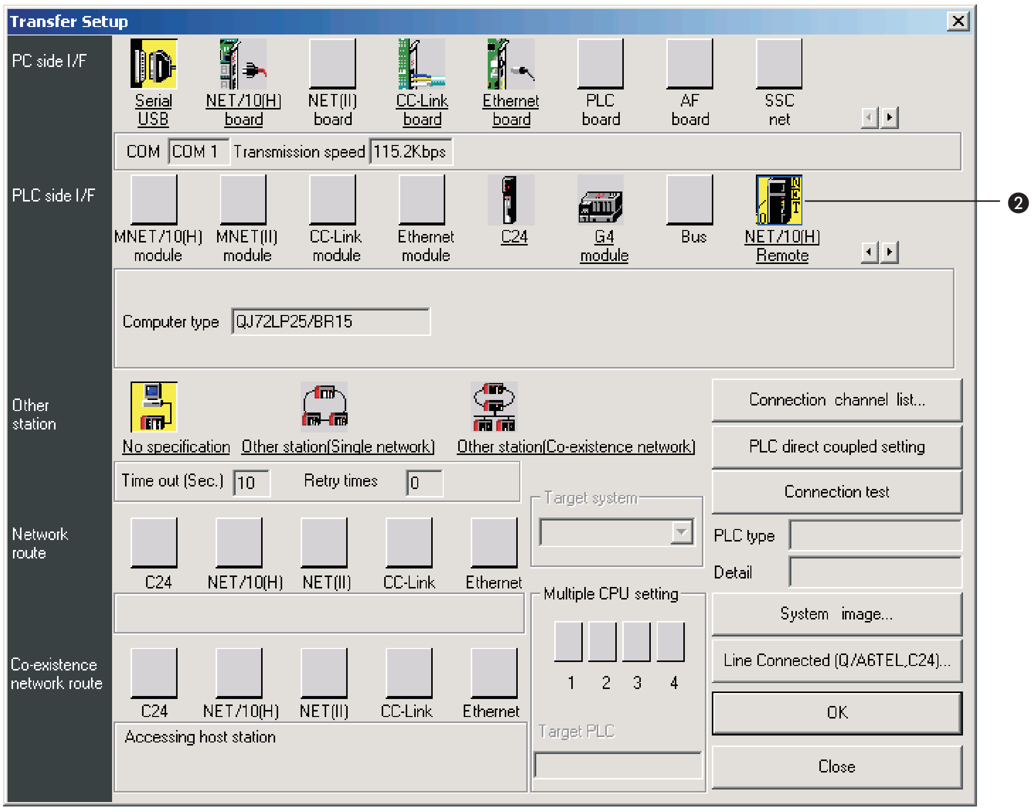
Fig. 8-4: *Transfer Setup for QnPRH CPU (2)*


Transfer Setup

COM COM 1 Transmission speed 9.6Kbps

Time out (Sec.) 10 Retry times 0

Fig. 8-4: *Transfer Setup for FX3U CPU (3)*


Transfer Setup

COM COM 1 Transmission speed 115.2Kbps

Computer type QJ72LP25/BR15

Time out (Sec.) 10 Retry times 0

Fig. 8-4: *Transfer Setup for Q Remote I/O (4)*

<table>
<tr><th>Number</th><th>Description</th></tr>
<tr><td>❶</td><td><b>PC side I/F</b>
Choose the I/F for the connection of the PC to the PLC.</td></tr>
<tr><td>❷</td><td><b>PLC side I/F</b>
Choose the unit to be connected with the personal computer.</td></tr>
<tr><td>❸</td><td><b>Other station</b>
Choose no network or one of the specified network types.</td></tr>
<tr><td>❹</td><td><b>Network route</b>
Choose the network type, network No., station number and first I/O No. to be accessed. The setting items depend on the network type that has been set.</td></tr>
<tr><td>❺</td><td><b>Coexistence network route</b>
Choose this when making access to the network different from the one where the personal computer is connected. Choose the network type, network No., station number and first I/O No. to be accessed. The setting items depend on the network type that has been set.</td></tr>
<tr><td>❻</td><td><b>Multiple PLC setting</b>
Specify when the access target is multiple CPUs. You can connect up to four PLC CPUs. In this option you decide which CPU is to be connected.</td></tr>
<tr><td>❼</td><td><b>Connection channel list…</b>
Lists possible connection modes and their image. You can set the connection target while looking at the Connection channel list.</td></tr>
<tr><td>❽</td><td><b>PLC direct coupled setting</b>
By clicking this button you change from Other station to the own station.</td></tr>
<tr><td>❾</td><td><b>Connection test</b>
Tests whether proper access can be made to the PLC set as the access target on the Connection Setup screen. If proper access can be made, the model name of the PLC as the access target appears in the CPU type field.</td></tr>
<tr><td>❿</td><td><b>Target System</b> [1]
Specifies the connection destination for redundant PLC systems.
<table>
<tr><th>Choice</th><th>Connection destination</th></tr>
<tr><td>Not specified</td><td>When a PLC is directly connected:
The PLC directly connected to the personal computer.
Via network:
The PLC at the station where the network module of the specified station No. is installed in the network communication path.</td></tr>
<tr><td>Control system</td><td>The PLC whose system type is the control system.</td></tr>
<tr><td>Standby system</td><td>The PLC whose system type is the standby system.</td></tr>
<tr><td>System A</td><td>The PLC connected to the A side connector of the tracking cable.</td></tr>
<tr><td>System B</td><td>The PLC connected to the B side connector of the tracking cable.</td></tr>
</table></td></tr>
<tr><td>⓫</td><td><b>System image</b>
Here you see an image of the setup system.</td></tr>
<tr><td>⓬</td><td><b>Line connected (Q/A6TEL, C24)... / TEL (FXCPU)...</b>
Clicking on this button opens a dialogue to setup a modem connection.</td></tr>
<tr><td>⓭</td><td><b>OK</b>
Closes the dialogue and saves the settings.</td></tr>
<tr><td>⓮</td><td><b>Cancel</b>
Closes the dialogue without saving.</td></tr>
</table>

Tab. 8-1: *Description for Fig. 8-4*

[1] For a redundant PLC system some functionalities are not available with all of these settings. For the following functionalities it is necessary to select **Not specified**, **System A** or **System B** to function properly:
- In Cycle Monitoring (→ 9.6)
- Online Module Change (→ 9.1.6)
- Forced Input Output Registration / Cancellation (→ 9.4.1)

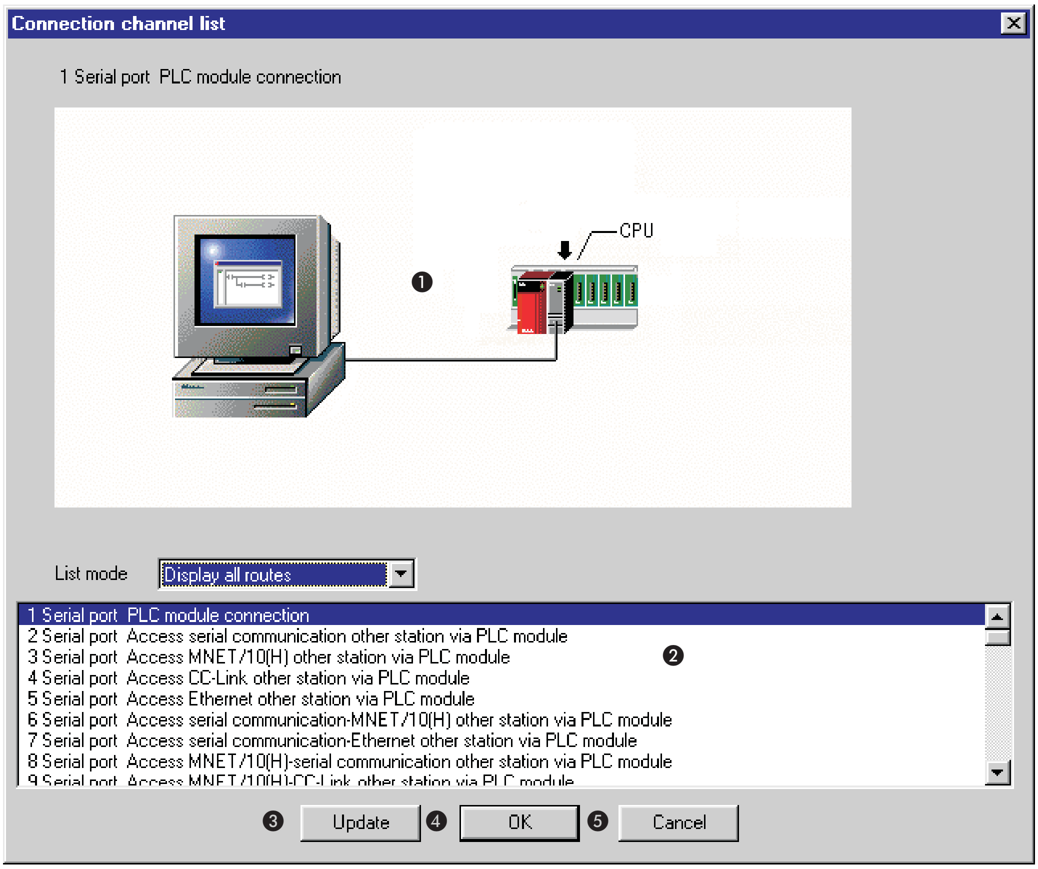
Setup of the connection to the PLC

To setup the connection follow the different setup option in Fig. 8-5. Therefore use the manuals of the PLC CPU and the network type you are using.

GX IEC Developer provides the possibility to setup the PLC and network parameter by using graphical images.

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.
The in Fig. 8-5 shown dialogue is opened.

② Click on the button **Connection channel list**. The following dialogue window is opened.



1 Serial port PLC module connection

List mode: Display all routes

1 Serial port PLC module connection
2 Serial port Access serial communication other station via PLC module
3 Serial port Access MNET/10(H) other station via PLC module
4 Serial port Access CC-Link other station via PLC module
5 Serial port Access Ethernet other station via PLC module
6 Serial port Access serial communication-MNET/10(H) other station via PLC module
7 Serial port Access serial communication-Ethernet other station via PLC module
8 Serial port Access MNET/10(H)-serial communication other station via PLC module
9 Serial port Access MNET/10(H)-CC-Link other station via PLC module

Fig. 8-5: *Dialogue* ***Connection channel list***

❶ Graphical image of the in ❷ selected network type.

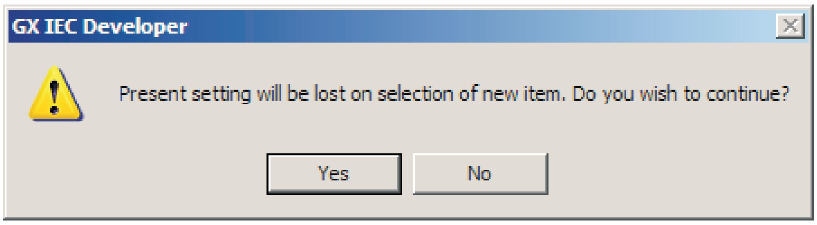
❷ Listing of the possible network configurations for the selected PLC .

❸ **Update**
Click this button to confirm the selected network configuration without closing the dialogue.

❹ **OK**
Confirm the set network configuration and close the dialogue.

❺ **Cancel**
Closes the dialogue without saving. When the button Update was clicked the dialogue is closed but the network configuration is already saved.

③ Use the scollingbar to scroll through the network configurations and select a network configuration corresponding to your network type by clicking in the list. Click on the button **Update**. Confirm the following security note with **OK**.

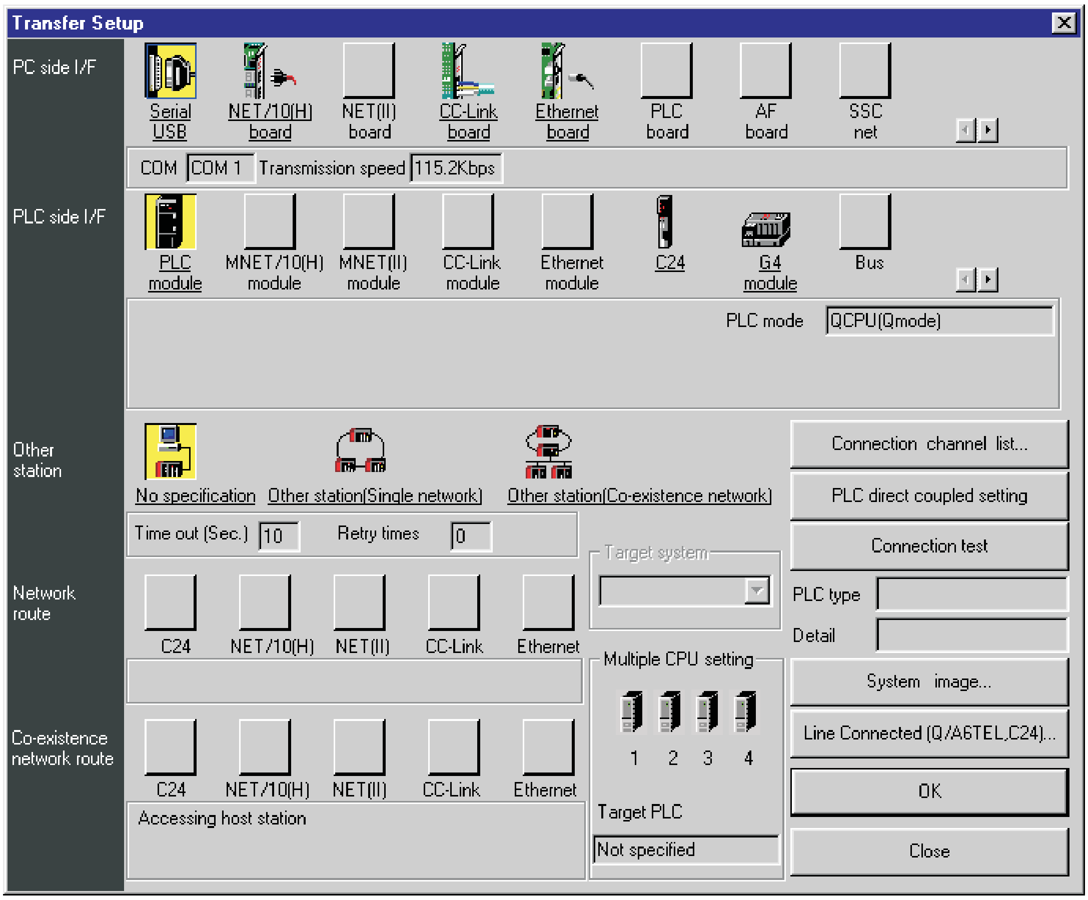


Present setting will be lost on selection of new item. Do you wish to continue?

Fig. 8-6: *Security note*

④ Test the selected network configuration by clicking on the button **Connection test**. If the connection between the PC and the network is possible a positive note will be shown on the screen. If no connection is possible an error message is shown. In this case you have to check and correct the connection between the PC and the network.

⑤ Close the dialogue by clicking on **OK**. The selected network configuration will be saved and shown in the **Transfer Setup**.



COM COM 1 Transmission speed 115.2Kbps

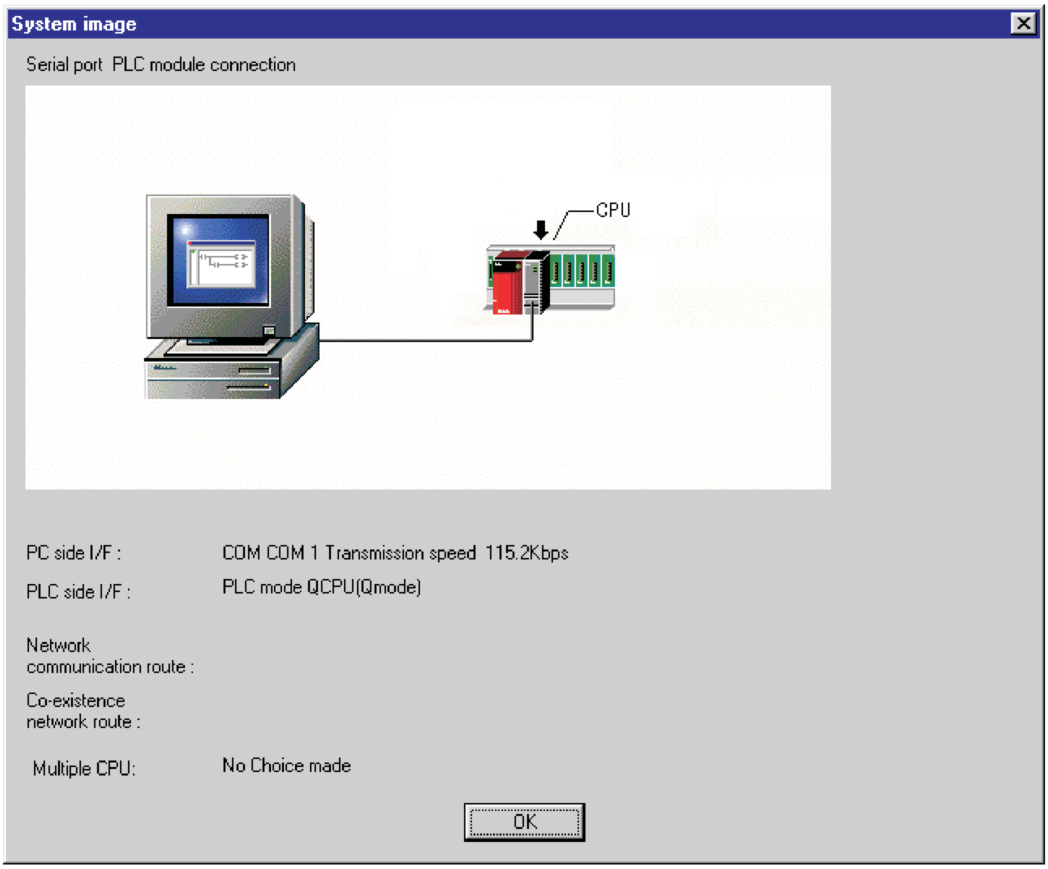
PLC mode QCPU(Qmode)

Time out (Sec.) 10 Retry times 0

Target PLC Not specified

Fig. 8-7: *Dialogue* ***Transfer Setup***

⑥ Now click on **System image**. An information window is opened in which an image of the setup system is shown.
In comparison to the **Connection channel list** dialogue in this dialogue the system parameter are also listed.
So you can change single settings and check the settings by the help of this dialogue.



Serial port PLC module connection

PC side I/F : COM COM 1 Transmission speed 115.2Kbps

PLC side I/F : PLC mode QCPU(Qmode)

Network communication route :

Co-existence network route :

Multiple CPU: No Choice made

Fig. 8-8: *Dialogue* ***System image***

NOTES

The COM and USB status is saved to the INI file every time you close the **Transfer Setup** dialogue with the **OK** button. The saved values are then used as defaults when you create a new project.

The USB status is only saved for Q PLCs. The selected COM port is only saved if USB is not selected. For new projects, the USB status is used as default for Q PLCs, except the Q00(J)/01 PLCs.

The following table shows the old **Transfer Setup** settings (Version 5.0 and before) and their equivalents in the new **Transfer Setup** (Version 6.0 and later).

Example ▽

Example for reading the table: If you selected the CPU-Port interface with MELSECNET/10 network in the old Transfer Setup, you will have to do the following in the new Transfer Setup to achieve the same result:

① Select the Serial / USB icon in the PC side I/F group.

② Double-click this icon to configure the settings such as port, baud rate, etc.

③ Select the PLC module icon in the PLC side I/F group.

④ Double-click this icon to configure the PLC mode (if the PLC supports this).

⑤ Select the Other station (Single network) icon in the Other Station group.

⑥ Double-click this icon to configure timeout and retries settings.

⑦ Select the NET/10(H) icon from the Network Route group.

⑧ Double-click this icon to configure the Network and Station numbers.

Old setting	New setting	
	Click to select…	**Double-click to configure…**
CPU-Port Host	PC side: Serial / USB PLC side: PLC module Other stat: No spec	COM port, Baud rate PLC mode Timeout, retries
CPU-Port Net/NetII	PC side: Serial / USB PLC side: PLC module Other stat: Single network Network: NET(II)	COM port, Baud rate PLC mode Timeout, retries Station no.
CPU-Port Net10	PC side: Serial / USB PLC side: PLC module Other stat: Single network Network: NET/10(H)	COM port, Baud rate PLC mode Timeout, retries Network no., station no.
CPU-Port CC-Link	PC side: Serial / USB PLC side: PLC module Other stat: Single network Network: CC-Link	COM port, Baud rate PLC mode Timeout, retries Head I/O, station no.
Computer Link Host	PC side: Serial / USB PLC side: C24 Other stat: No spec	COM port, Baud rate PLC type, Station no., parity Timeout, retries
Computer Link Net/NetII	PC side: Serial / USB PLC side: C24 Other stat: Single network Network: NET(II)	COM port, Baud rate PLC type, Station no., parity Timeout, retries Station no.
Computer Link Net10	PC side: Serial / USB PLC side: C24 Other stat: Single network Network: NET/10(H)	COM port, Baud rate PLC type, Station no., parity Timeout, retries Network no., station no.
Computer Link CC-Link	PC side: Serial / USB PLC side: C24 Other stat: Single network Network: CC-Link	COM port, Baud rate PLC type, Station no., parity Timeout, retries Head I/O, station no.
Ethernet Host	PC side: Ethernet board PLC side: Ethernet module Other stat: Single network Network: Ethernet	Network no., station no., protocol PLC type, IP address, host name Timeout, retries Network no., station no.
Ethernet Net/NetII	PC side: Ethernet board PLC side: Ethernet module Other stat: Co-existence Network: Ethernet Co-existence: NET(II)	Network no., station no., protocol PLC type, IP address, host name Timeout, retries — Station no.
Ethernet Net10	PC side: Ethernet board PLC side: Ethernet module Other stat: Single network Network: Ethernet	Network no., station no., protocol PLC type, IP address, host name Timeout, retries Network no., station no.
Ethernet CC-Link	PC side: Ethernet board PLC side: Ethernet module Other stat: Co-existence Network: Ethernet Co-existence: CC-Link	Network no., station no., protocol PLC type, IP address, host name Timeout, retries Network no., station no. Head I/O, station no.
USB Port Host	PC side: Serial / USB PLC side: PLC module Other stat: No spec	Select USB PLC mode Timeout, retries
USB Port Net10	PC side: Serial / USB PLC side: PLC module Other stat: Single network Network: NET/10(H)	Select USB PLC mode Timeout, retries Network no., station no.
USB Port CC-Link	PC side: Serial / USB PLC side: PLC module Other stat: Single network Network: CC-Link	Select USB PLC mode Timeout, retries Head I/O, station no.

Tab. 8-2: *Comparison of old and new setting*

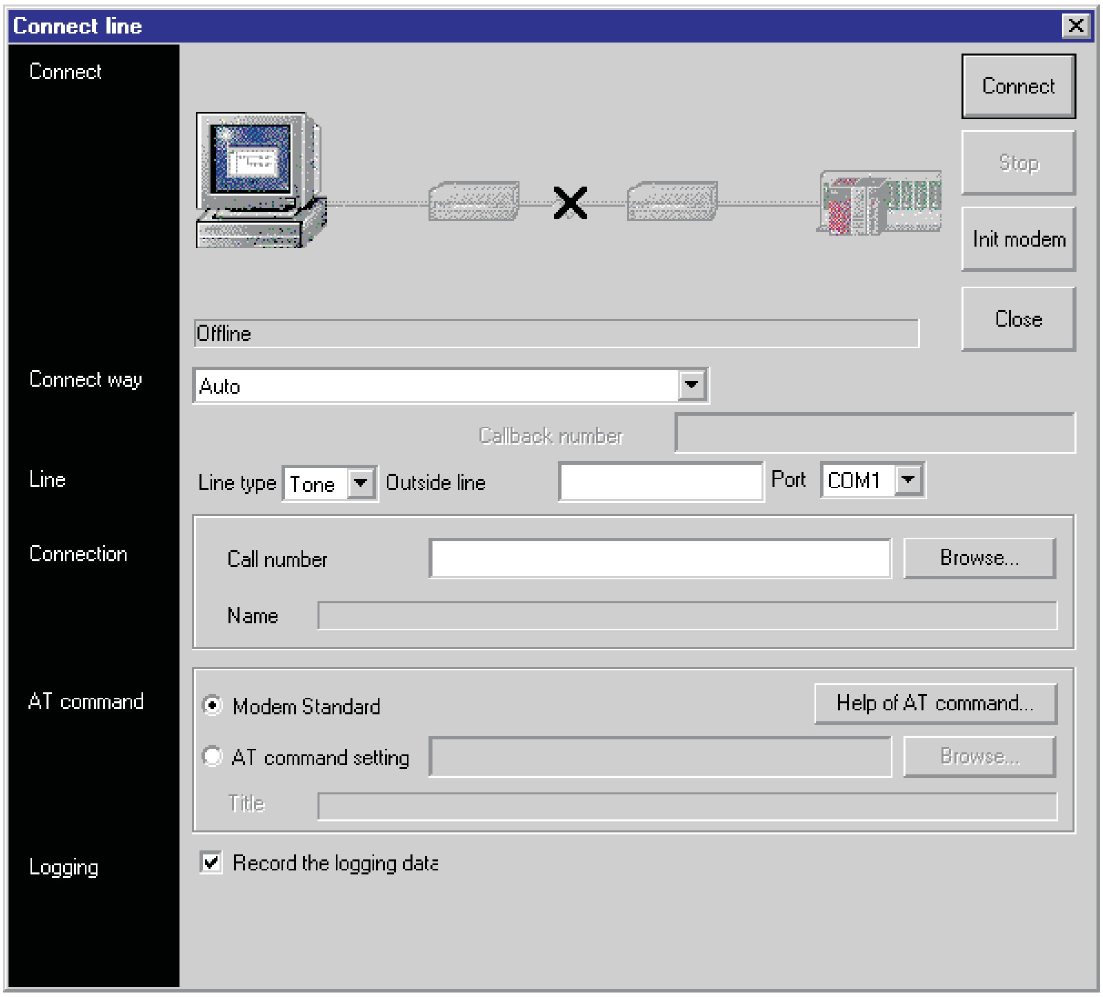
8.4 Modem settings for analog Modems

When using an analog modem make the following settings.

① Set MAC transparent mode with CPU port or C24 in the menu **Transfer Setup -> Ports**.

② Set the desired data transmission settings on the sending and receiving modems. You can use any terminal program (e.g. Windows Hyperterminal) to configure the modem settings.

③ Click on the button **Line connected** in the menu **Transfer Setup -> Ports**.



Connect line

Connect: Offline

Connect way: Auto

Line: Line type: Tone; Outside line: ; Port: COM1

AT command: ● Modem Standard ○ AT command setting

Logging: ☑ Record the logging date

Fig. 8-9: *Modem settings*

Connection: Enter the telephone number here. Or select one out of the **Call book** by clicking on the Browse button.

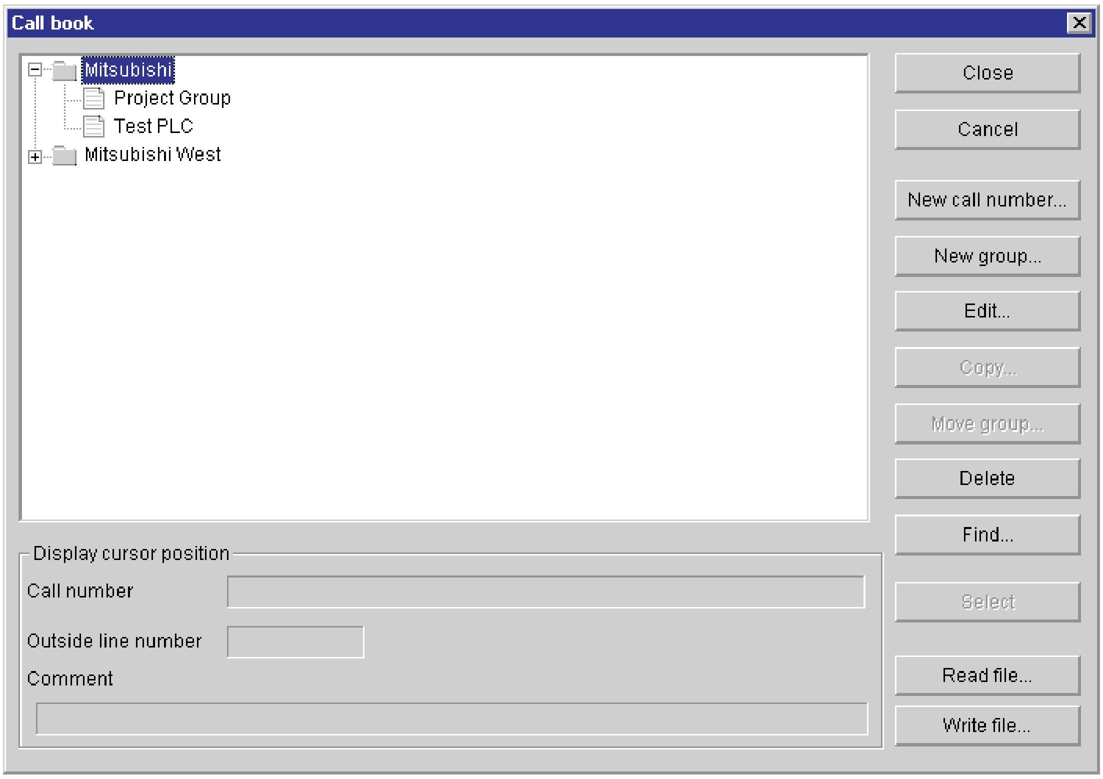


Fig. 8-10: Call book

AT command: The modem settings are set here. Or select one out of the **AT command registration** by clicking on the Browse button.

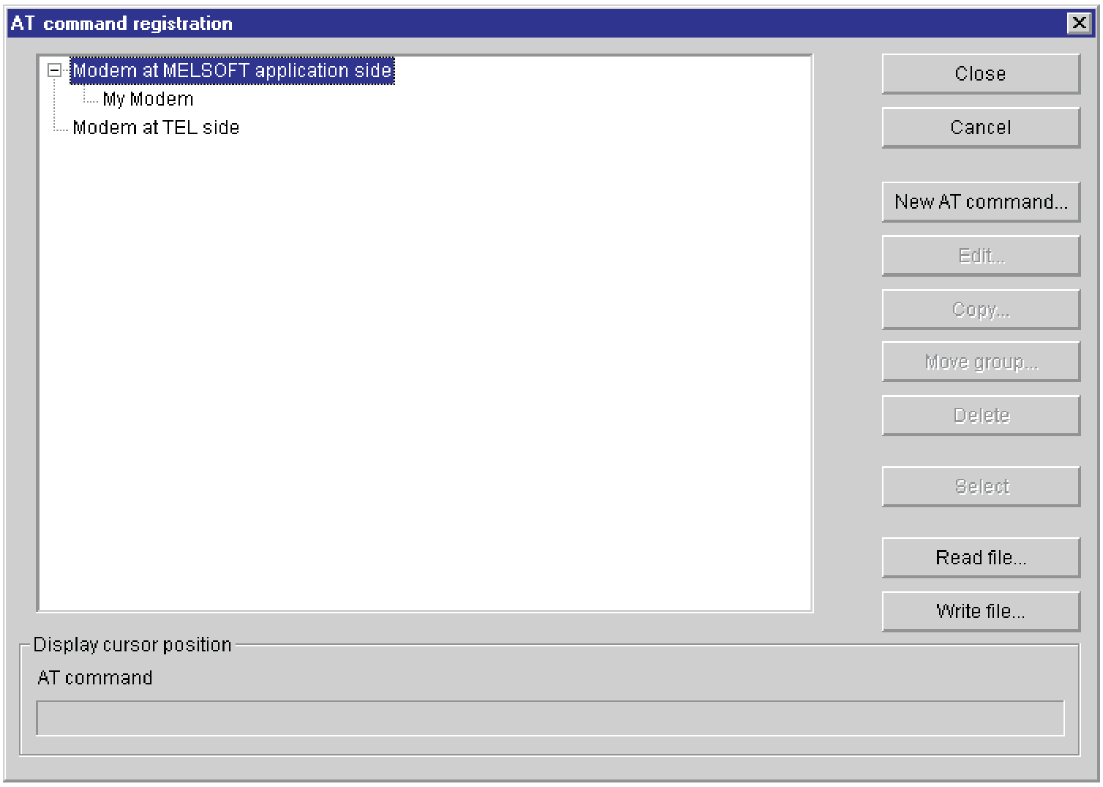


Fig. 8-11: AT command registration

Data transmission procedure

① Set data transmission settings in GX IEC Developer and on MAC.

② Receiving modem: set “Auto Answer On”.

③ Set data transmission settings on receiving and sending modems.

④ Start connection: dial receiving modem.

⑤ Quit terminal software, though without hanging up.

⑥ Take the necessary steps in GX IEC Developer.

Close communications

When the PLC and other devices (e.g. the MAC terminal in transparent mode) are connected to the Personal Computer via the same communication port, the communications must be closed to release the COM port when switching between the two software applications GX IEC Developer and MAC Programmer.

① Click on the command **Close communications** in the **Online** menu.

NOTES

The communications can not be closed when GX IEC Developer is running in monitor mode.

Follow the following precautions for access to the other station in the following Q, QnA, and A CPU mixed system are as follows:

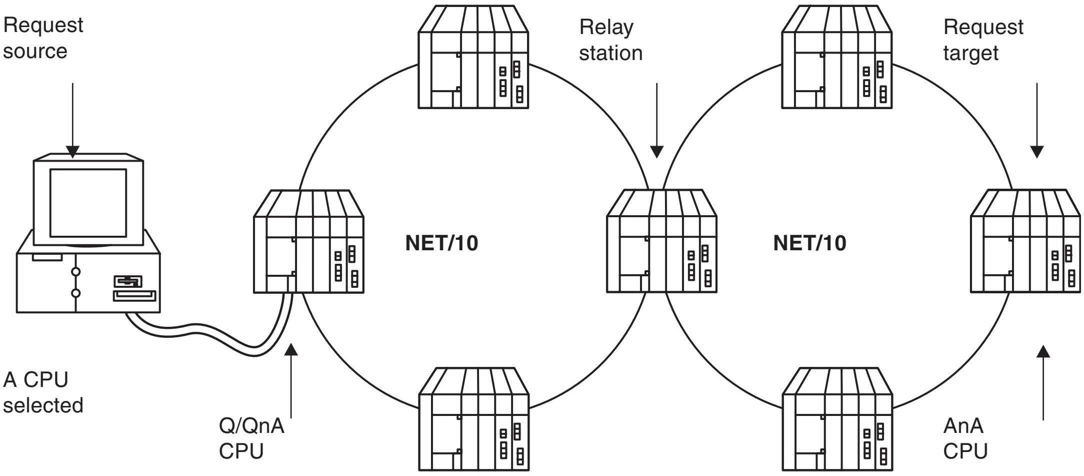


Fig. 8-12: *Hardware configuration for Q, QnA, and A CPU mixed systems.*

① Change the peripheral device to the PLC type of the request target. Communication can not be performed, if the PLC type differs.

② The relay station used should be the Q CPU or QnA CPU.

③ If GX IEC Developer is connected to the QnA CPU and the request target is the A CPU GX IEC Developer can not communicate with the request target (A CPU).

④ If GX IEC Developer is connected to the Q CPU and the request target is the Q CPU, GX IEC Developer can not communicate with the request target (Q CPU) via the relay station (QnA/A CPU).

8.5 Modem settings for GSM Modems

8.5.1 FX-Series – Communication via Programming port

Environment: Analogue Modem -> GSM Modem -> Programming port

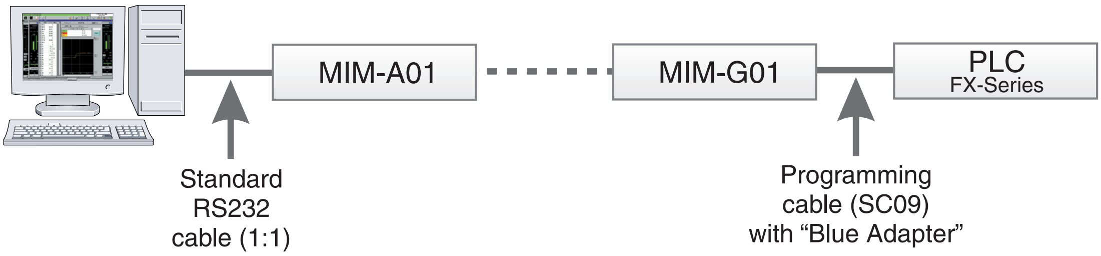


Fig. 8-13: *Environment*

Analogue Modem settings:

AT	E0
	&S0
	&D0
	&K0

When the analogue modem is used at the PC side, the parity settings have to be modified:

Parity:	Even
Data Bits:	7bits
Stop Bits:	1bit
AT	+TFORMAT=”7E1”

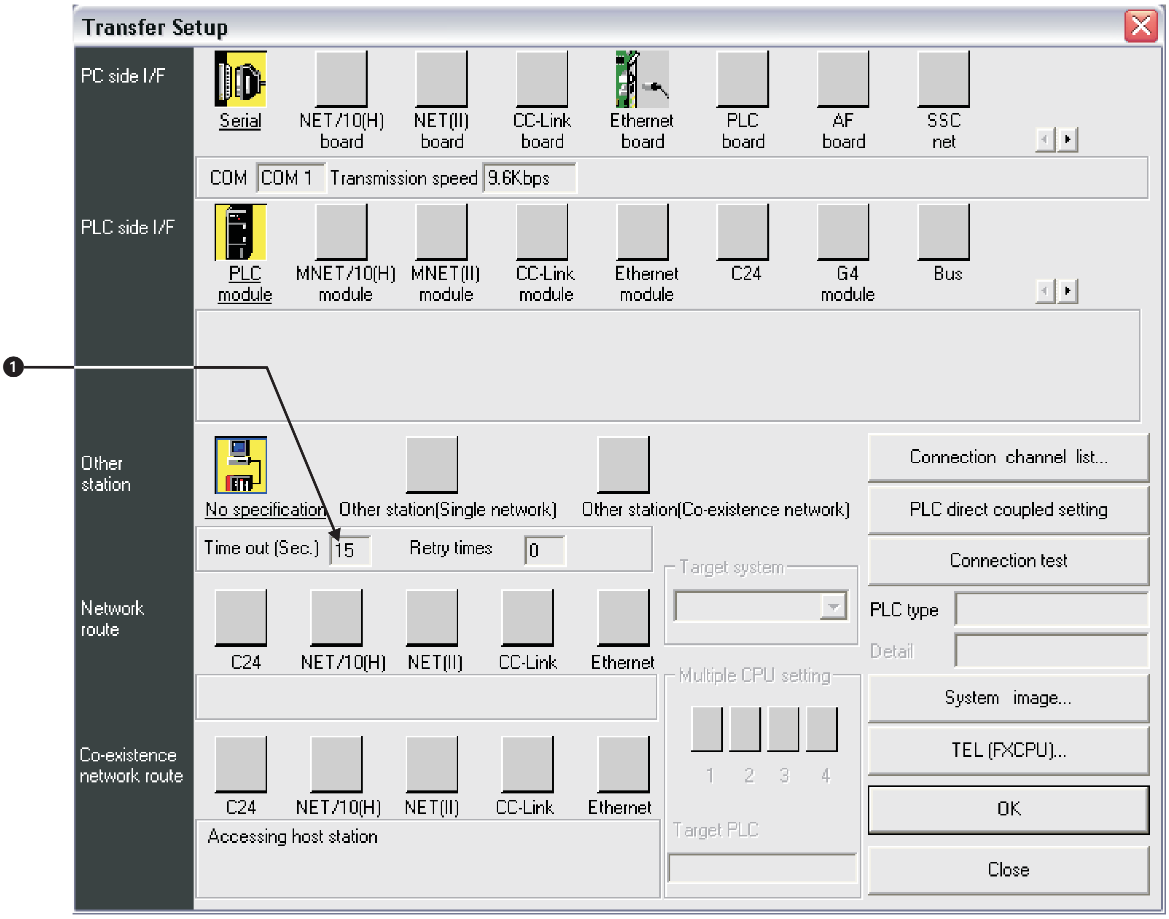
GSM Modem settings:

AT	E0
	S0=2
	&S0
	&D0
	+CICB=0
	+IFC=0,0
	+CMEE=1
	+IPR=9600
	+CPIN=____ (PIN No. of SIM card)

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

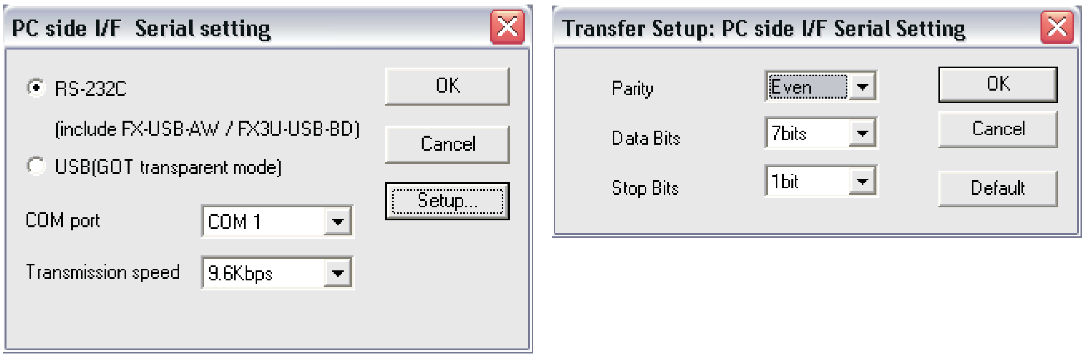
COM COM 1 Transmission speed 9.6Kbps

Time out (Sec.) 15 Retry times 0

Fig. 8-14: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

● RS-232C
(include FX-USB-AW / FX3U-USB-BD)
○ USB(GOT transparent mode)

COM port COM 1

Transmission speed 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

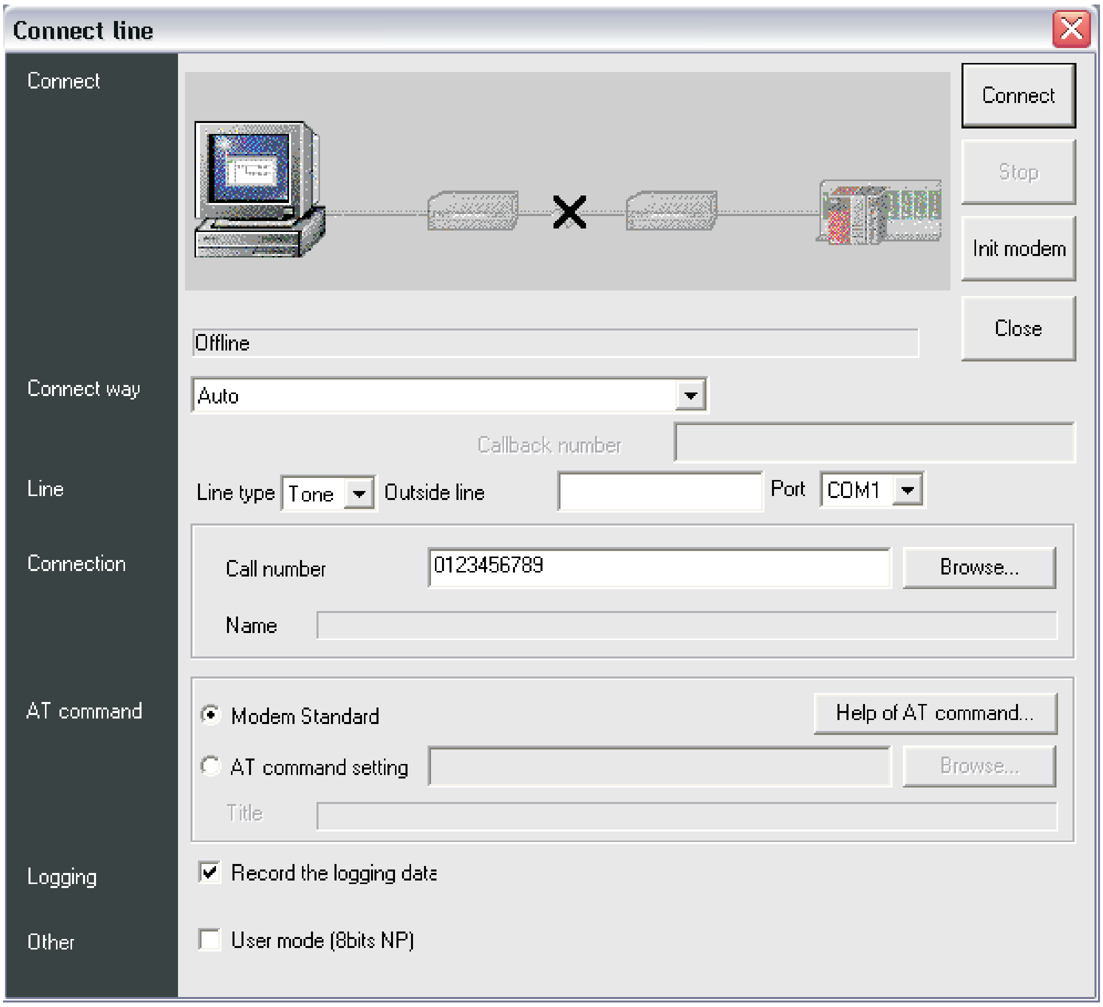
Parity Even

Data Bits 7bits

Stop Bits 1bit

Fig. 8-15: *Setting the communication details*

⑤ Select **TEL (FXCPU)**, enter the **Call number** of the remote modem and click **Connect**.



Connect: Offline

Connect way: Auto

Callback number

Line: Line type Tone, Outside line, Port COM1

Connection: Call number 0123456789, Name

AT command: ● Modem Standard ○ AT command setting, Title

Logging: ☑ Record the logging date

Other: □ User mode (8bits NP)

Fig. 8-16: *Set up the connection*

⑥ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Modem -> Programming port

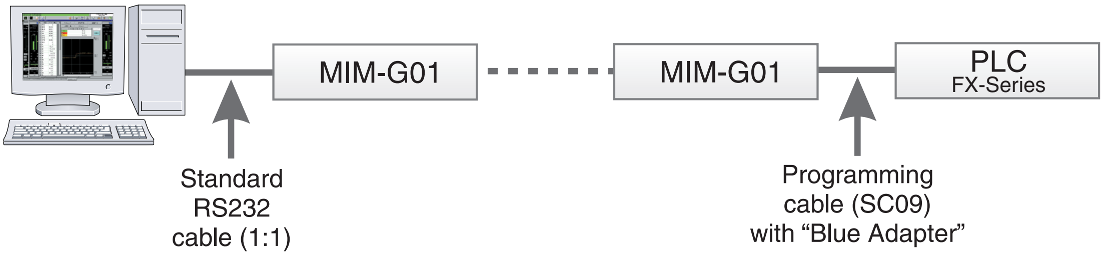


Fig. 8-17: *Environment*

GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        &W
        +CPIN=____  (PIN No. of SIM card)
```

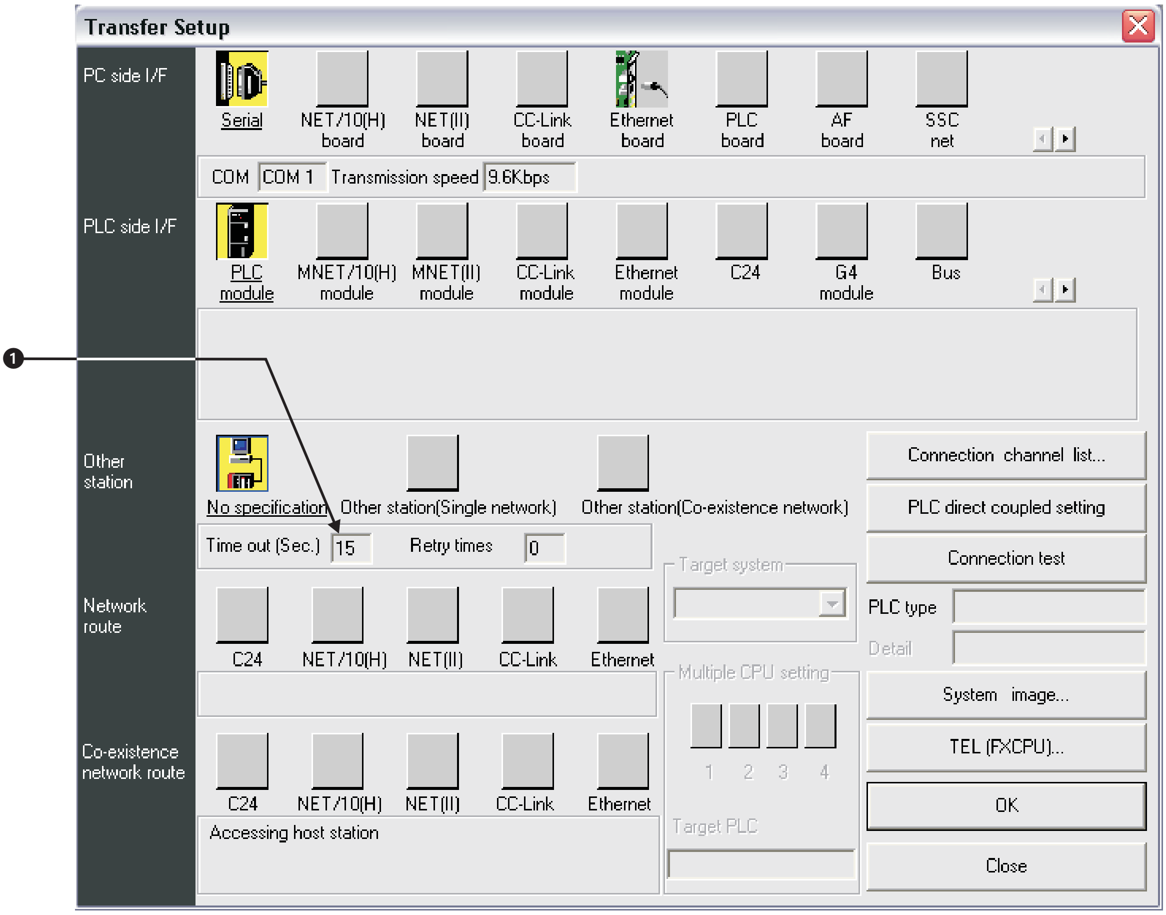
The modem which is used at the PC side has to be modified regarding the parity settings:

Parity:	Even
Data Bits:	7bits
Stop Bits:	1bit
AT	+ICF=5,1

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

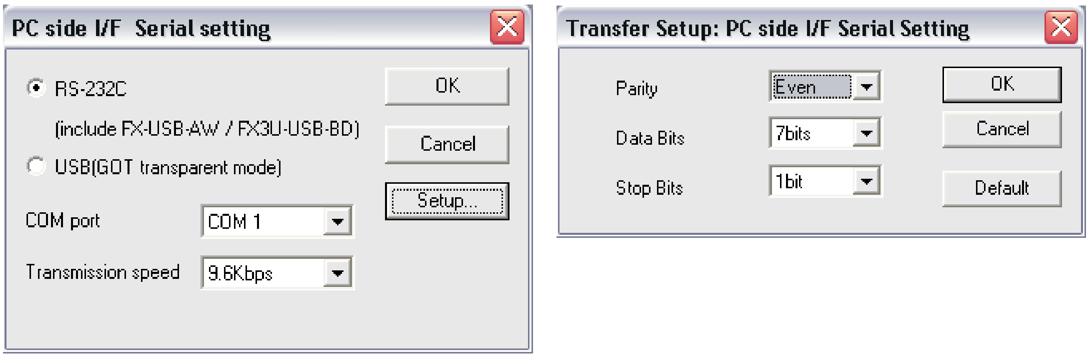
COM: COM 1 Transmission speed: 9.6Kbps

Time out (Sec.): 15 Retry times: 0

Fig. 8-18: *Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

● RS-232C (include FX-USB-AW / FX3U-USB-BD)

○ USB(GOT transparent mode)

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

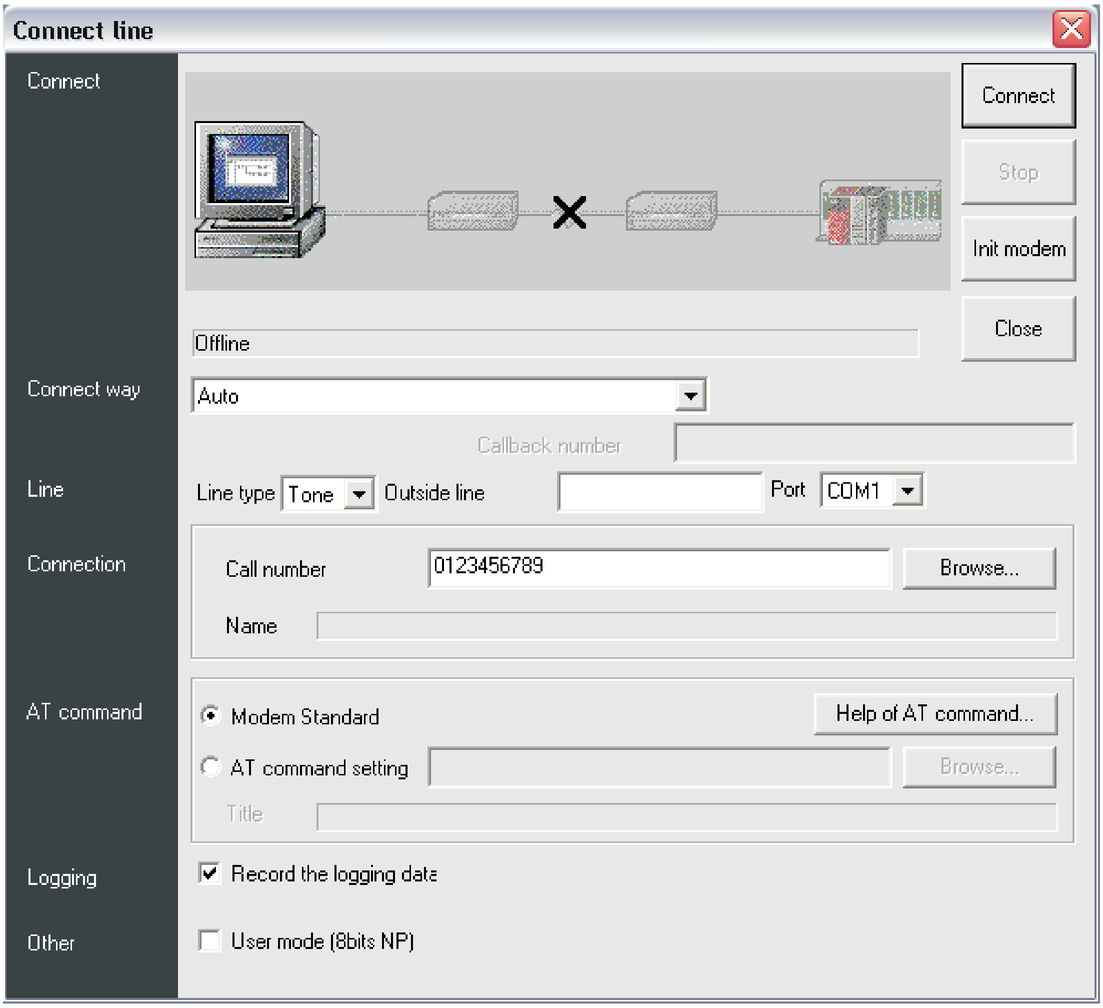
Parity: Even

Data Bits: 7bits

Stop Bits: 1bit

Fig. 8-19: *Setting the communication details*

⑤ Select **TEL (FXCPU)**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Offline

Connect way: Auto

Callback number:

Line: Line type Tone, Outside line, Port COM1

Connection: Call number 0123456789

Name

AT command: ● Modem Standard ○ AT command setting

Title

Logging: ☑ Record the logging date

Other: ☐ User mode (8bits NP)

Fig. 8-20: *Set up the connection*

⑥ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

8.5.2 FX-Series – Communication with RS 232 BD

Environment: Analogue Modem -> GSM Modem -> RS 232 BD

NOTE A modification of the PLC settings for the RS 232 BD is not necessary.

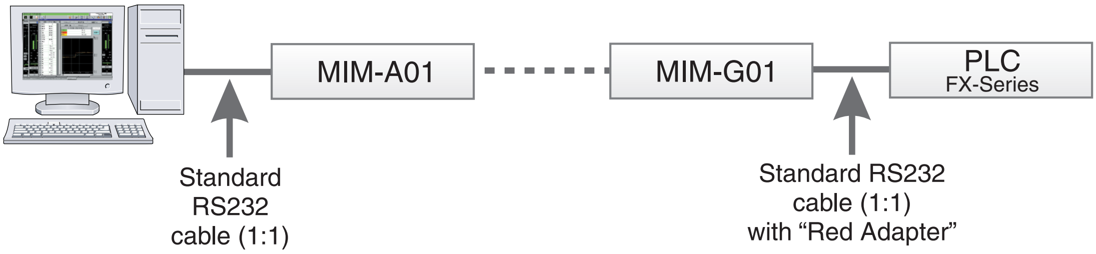

MIM-A01
MIM-G01
PLC
FX-Series
Standard RS232 cable (1:1)
Standard RS232 cable (1:1) with “Red Adapter”

Fig. 8-21: *Environment*

Analogue Modem settings:

```
AT        E0
          &S0
          &D0
          &K0
```

When the analogue modem is used at the PC side, the parity settings have to be modified:

Parity:	Even
Data Bits:	7bits
Stop Bits:	1bit
AT	+TFORMAT=”7E1”

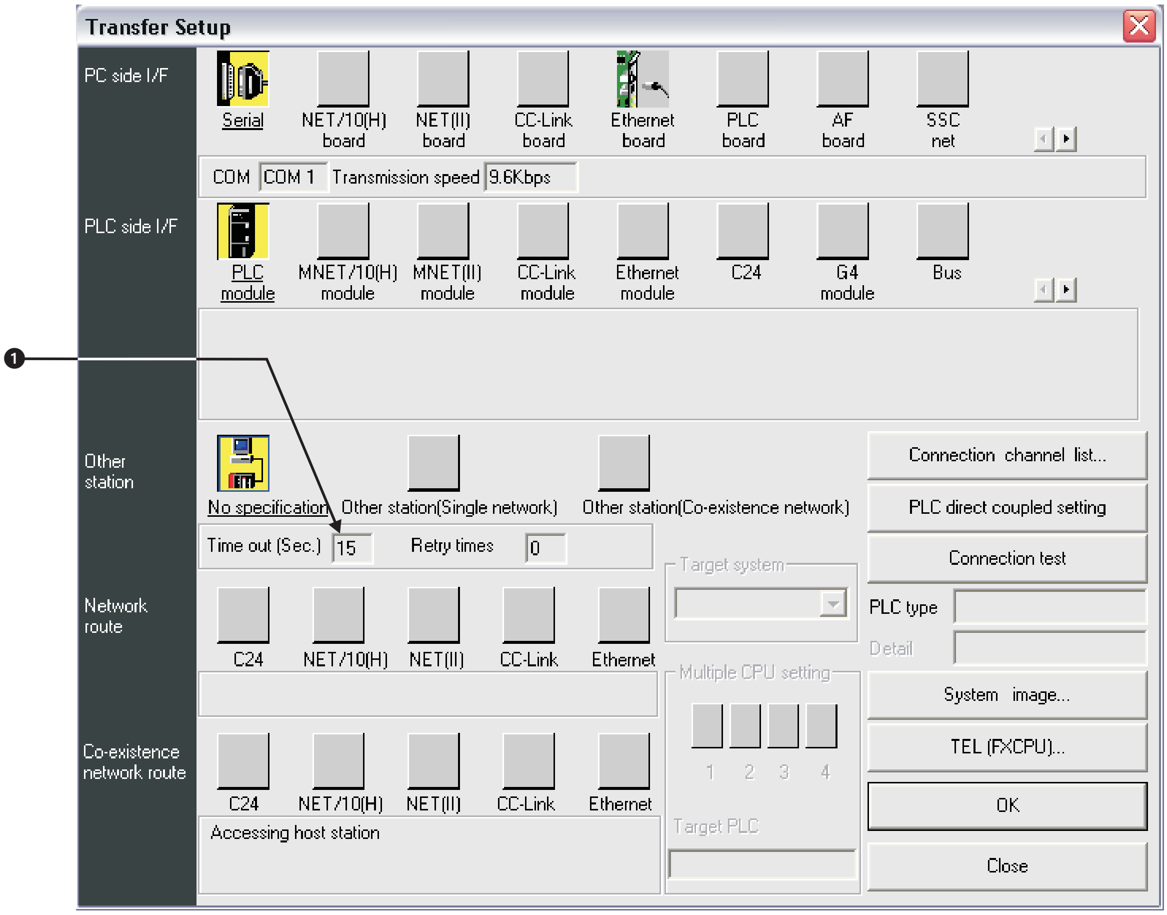
GSM Modem settings:

```
AT        E0
          S0=2
          &S0
          &D0
          +CICB=0
          +IFC=0,0
          +CMEE=1
          +IPR=9600
          +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



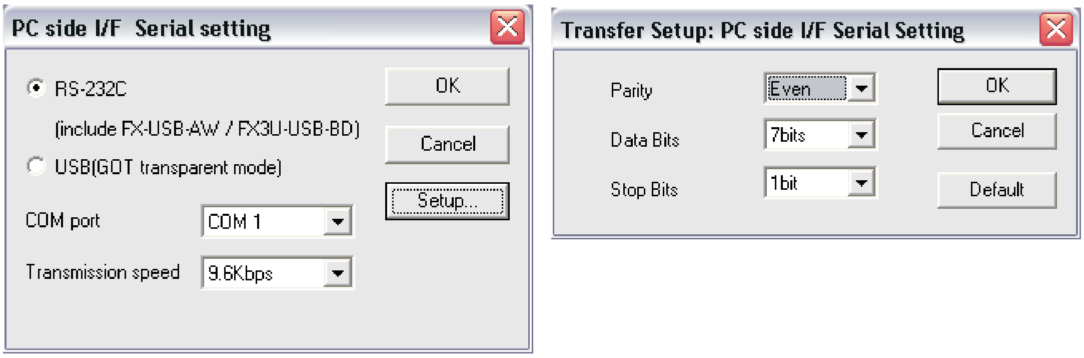
COM COM 1 Transmission speed 9.6Kbps

Time out (Sec.) 15 Retry times 0

Fig. 8-22: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



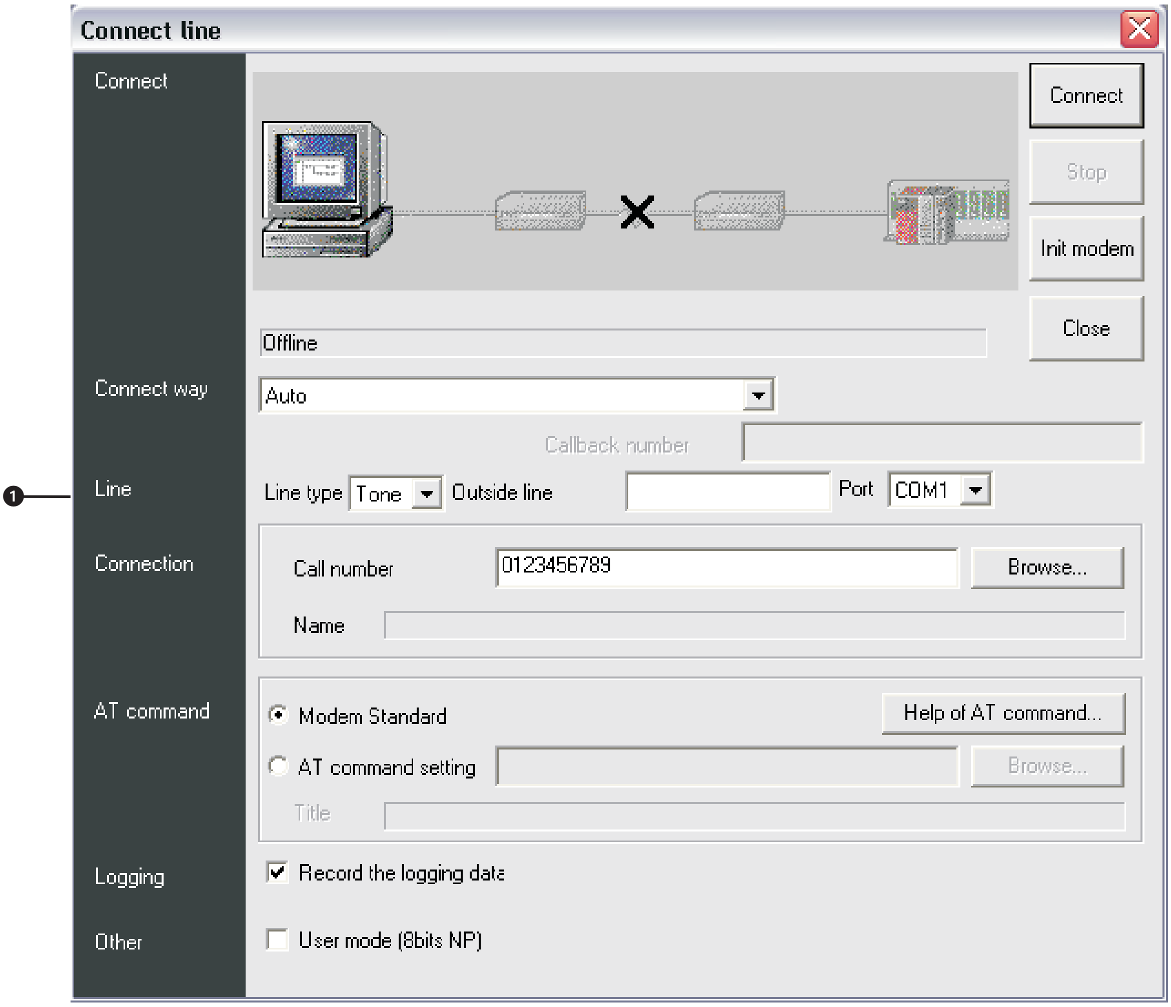
PC side I/F Serial setting
OK
Cancel
Setup...
Transfer Setup: PC side I/F Serial Setting
OK
Cancel
Default

● RS-232C
(include FX-USB-AW / FX3U-USB-BD)
○ USB(GOT transparent mode)
COM port COM 1
Transmission speed 9.6Kbps

Parity Even
Data Bits 7bits
Stop Bits 1bit

Fig. 8-23: *Setting the communication details*

⑤ Select **TEL (FXCPU)**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Connect

Offline

Connect way: Auto

Callback number:

Line: Line type: Tone; Outside line: ; Port: COM1

Connection: Call number: 0123456789; Name:

AT command: ● Modem Standard; ○ AT command setting; Title:

Logging: ☑ Record the logging date

Other: ☐ User mode (8bits NP)

Fig. 8-24: *Set up the connection*

⑥ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Modem -> RS 232 BD

NOTE A modification of the PLC settings for the RS 232 BD is not necessary.

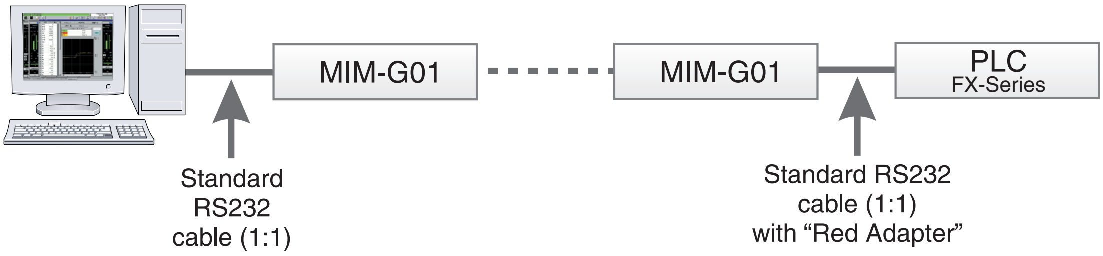

MIM-G01
MIM-G01
PLC
FX-Series
Standard RS232 cable (1:1)
Standard RS232 cable (1:1) with “Red Adapter”

Fig. 8-25: *Environment*

GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        +CPIN=____  (PIN No. of SIM card)
```

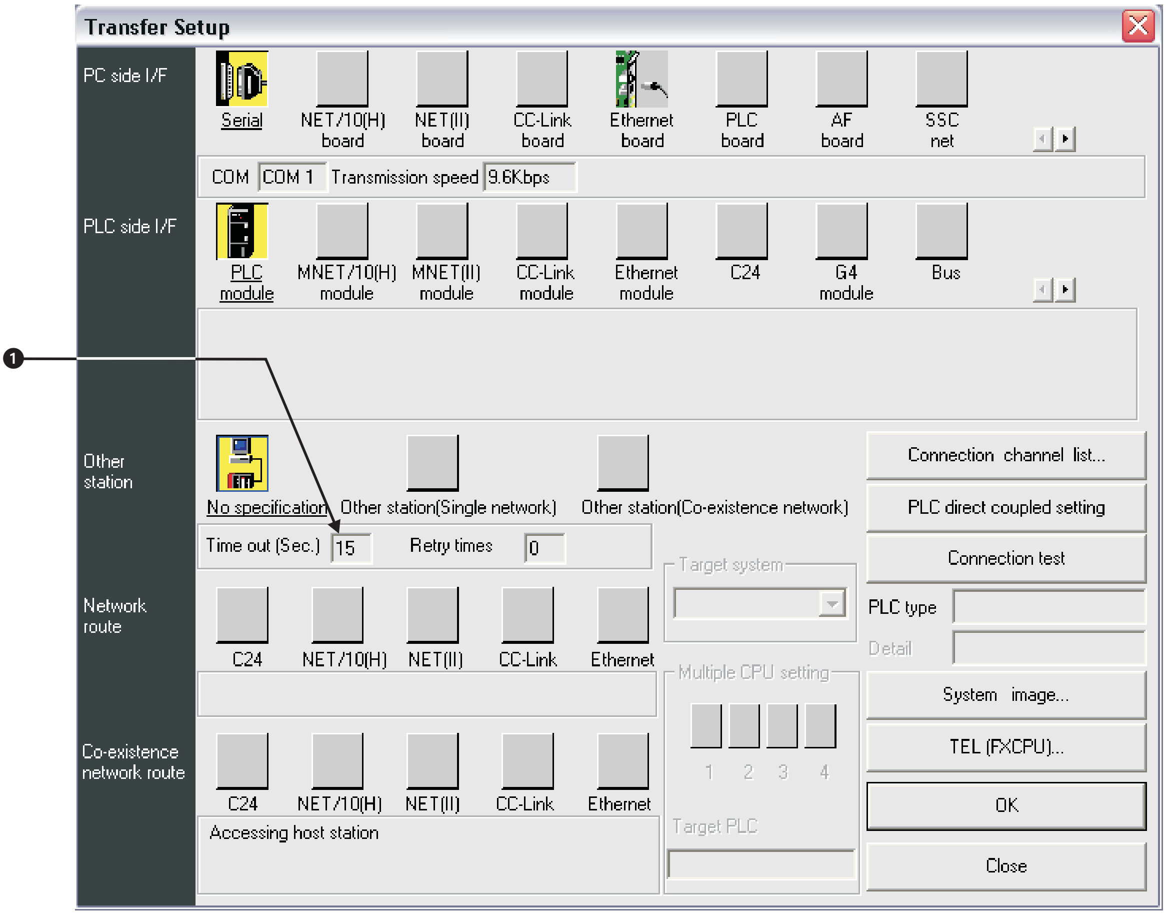
The modem which is used at the PC side has to be modified regarding the parity settings:

Parity:	Even
Data Bits:	7bits
Stop Bits:	1bit
AT	+ICF=5,1

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

PC side I/F: Serial, NET/10(H) board, NET(II) board, CC-Link board, Ethernet board, PLC board, AF board, SSC net

COM: COM 1 Transmission speed: 9.6Kbps

PLC side I/F: PLC module, MNET/10(H) module, MNET(II) module, CC-Link module, Ethernet module, C24, G4 module, Bus

Other station: No specification, Other station(Single network), Other station(Co-existence network)

Time out (Sec.): 15 Retry times: 0

Network route: C24, NET/10(H), NET(II), CC-Link, Ethernet

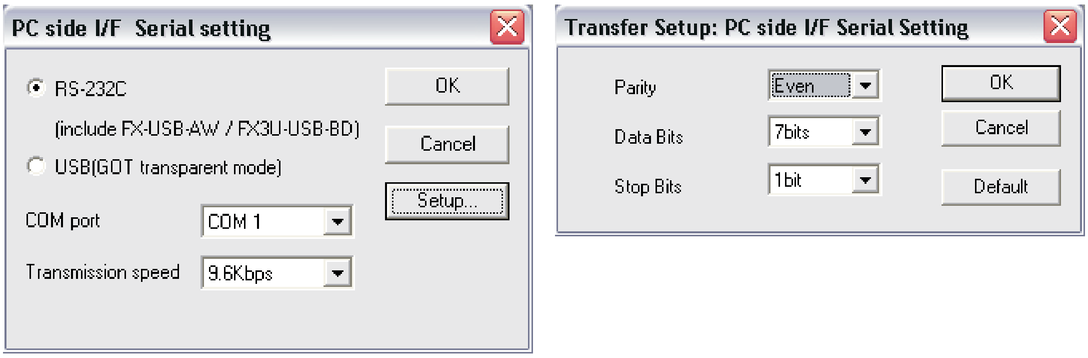
Co-existence network route: C24, NET/10(H), NET(II), CC-Link, Ethernet

Accessing host station

*Fig. 8-26: Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

● RS-232C (include FX-USB-AW / FX3U-USB-BD)

○ USB(GOT transparent mode)

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

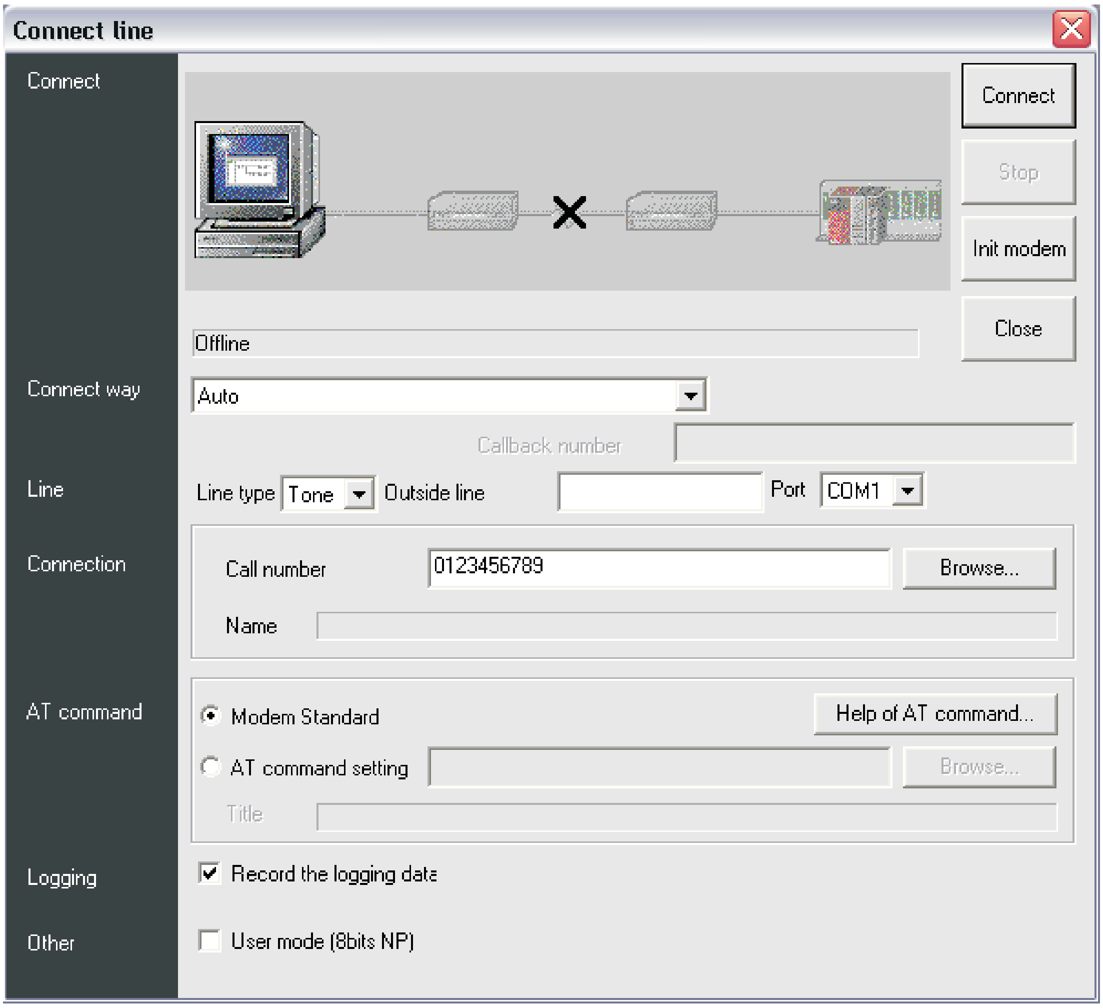
Parity: Even

Data Bits: 7bits

Stop Bits: 1bit

Fig. 8-27: Setting the communication details

⑤ Select **TEL (FXCPU)**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Offline

Connect way: Auto

Line type: Tone   Outside line: Port: COM1

Call number: 0123456789

● Modem Standard

○ AT command setting

☑ Record the logging date

☐ User mode (8bits NP)

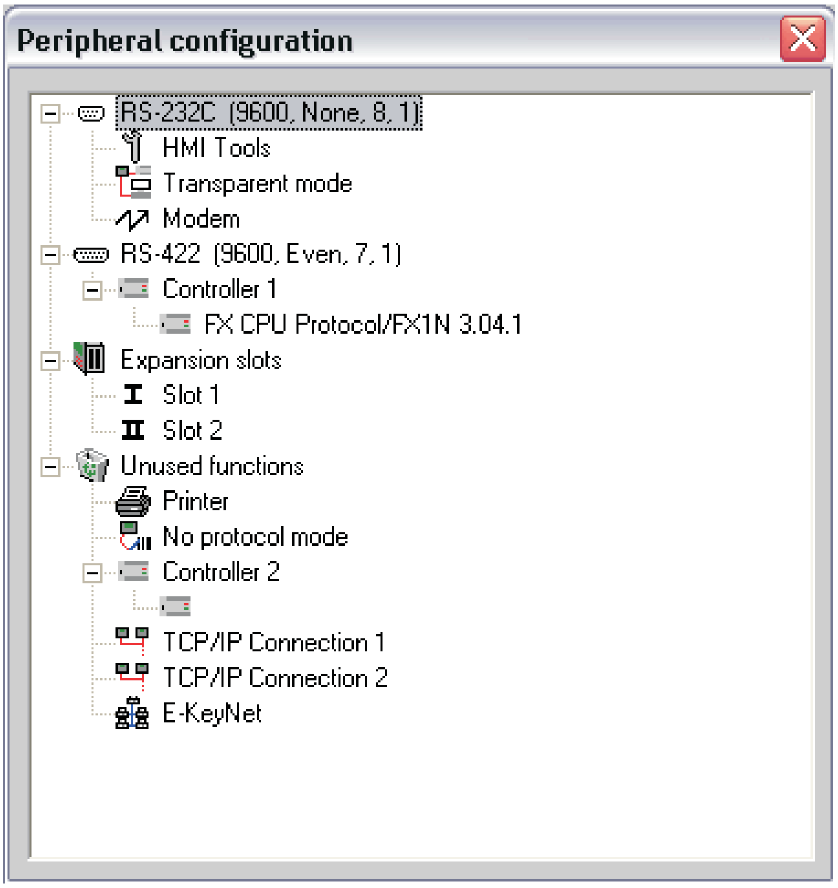
Fig. 8-28: *Set up the connection*

⑥ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

8.5.3 FX-Series – Communication via MAC (Transparent Mode)

Settings for MAC Terminal:



Peripheral configuration

- RS-232C (9600, None, 8, 1)
  - HMI Tools
  - Transparent mode
  - Modem
- RS-422 (9600, Even, 7, 1)
  - Controller 1
    - FX CPU Protocol/FX1N 3.04.1
- Expansion slots
  - Slot 1
  - Slot 2
- Unused functions
  - Printer
  - No protocol mode
  - Controller 2
  - TCP/IP Connection 1
  - TCP/IP Connection 2
  - E-KeyNet

Fig. 8-29: *Setting the peripheral configuration in the E-Designer*

Environment: Analogue Modem -> GSM Modem -> MAC -> FX Programming Port

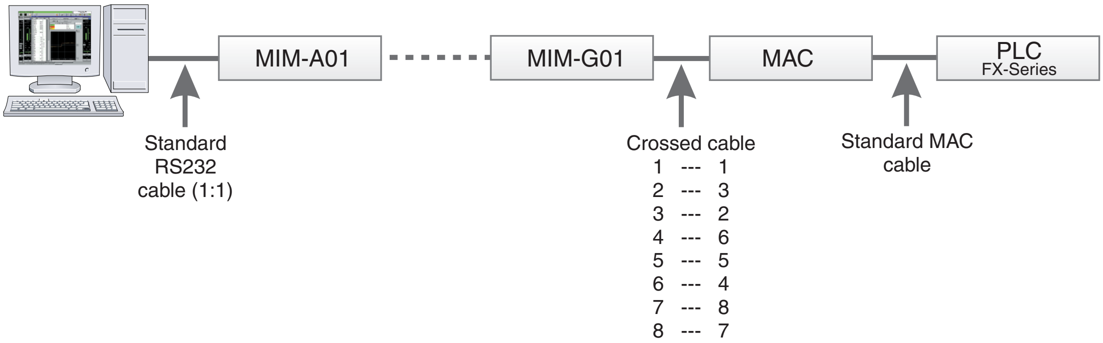


Fig. 8-30: *Environment*

GSM Modem settings:

```
AT          E0
            S0=2
            &S0
            &D0
            +CICB=0
            +IFC=0,0
            +CMEE=1
            +IPR=9600
            +CPIN=____  (PIN No. of SIM card)
```

Analogue Modem settings:

```
AT          E0
            &S0
            &D0
            &K0
```

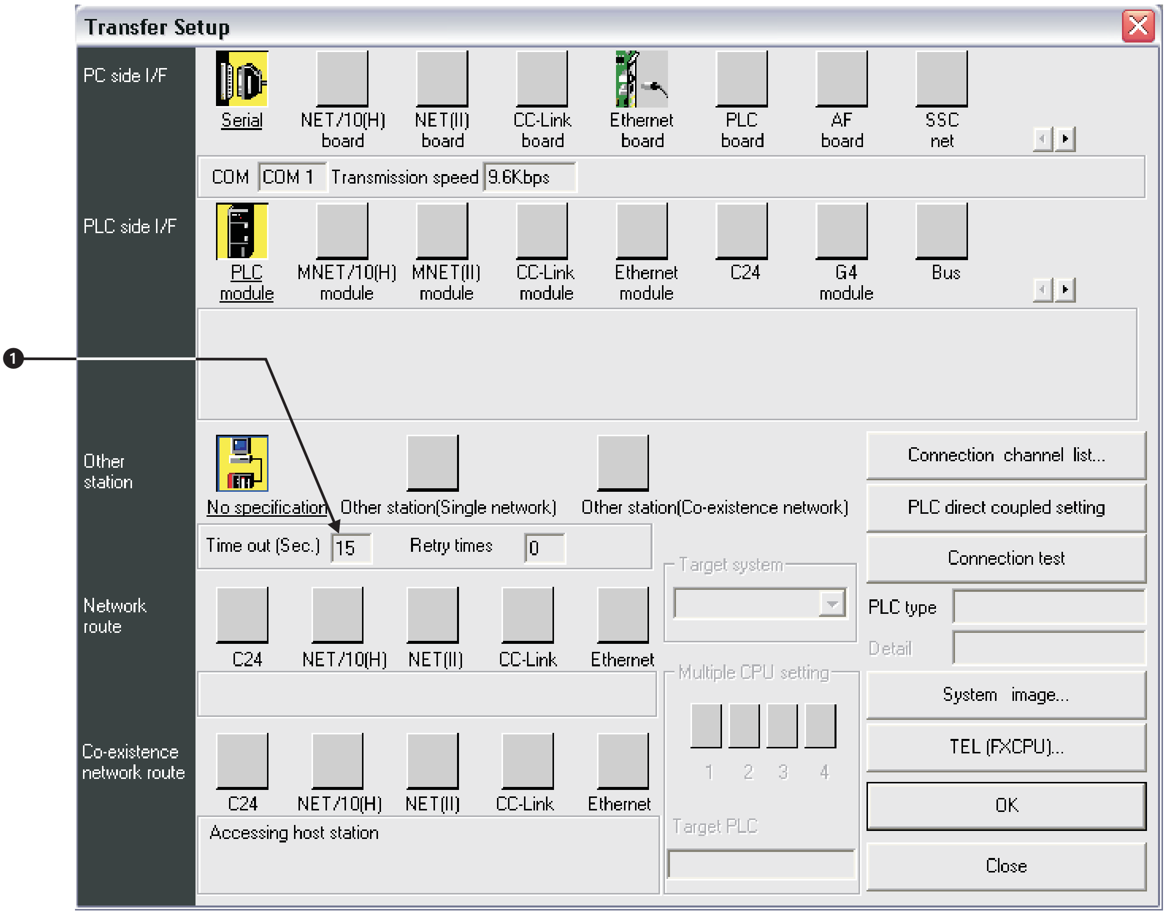
When the analogue modem is used at the PC side, the parity settings have to be modified:

Parity:	Even
Data Bits:	7bits
Stop Bits:	1bit
AT	+TFORMAT=”7E1”

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

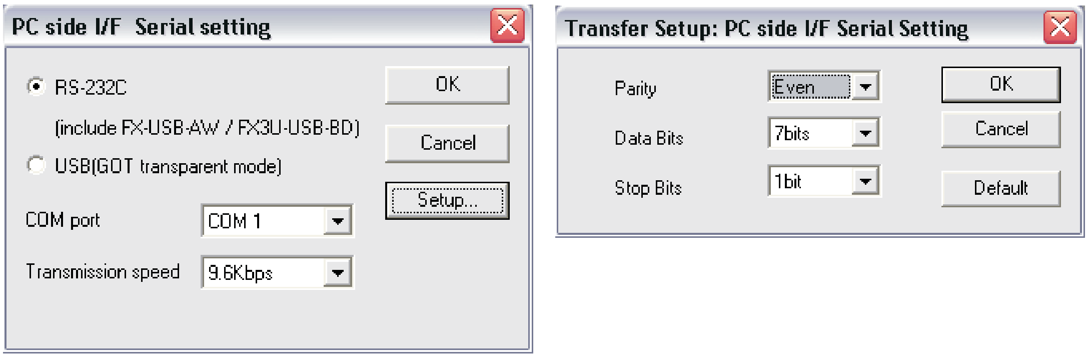
COM: COM 1 Transmission speed: 9.6Kbps

Time out (Sec.): 15 Retry times: 0

Fig. 8-31: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

● RS-232C (include FX-USB-AW / FX3U-USB-BD)
○ USB(GOT transparent mode)

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

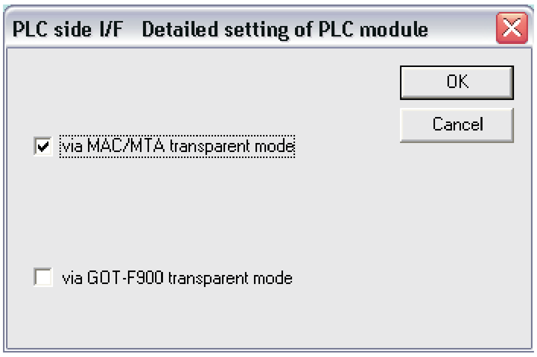
Parity: Even

Data Bits: 7bits

Stop Bits: 1bit

Fig. 8-32: *Setting the communication details*

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.

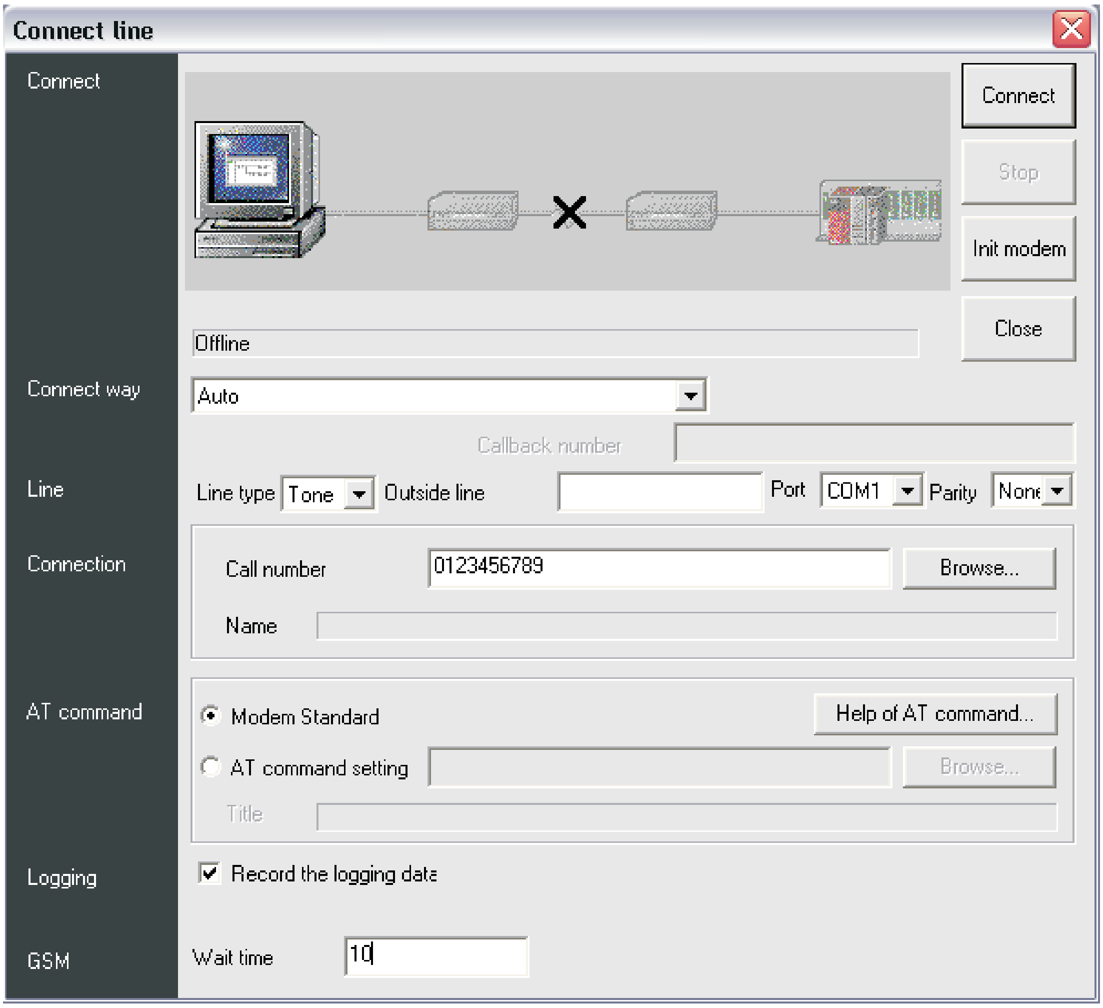

PLC side I/F Detailed setting of PLC module

☑ via MAC/MTA transparent mode

☐ via GOT-F900 transparent mode

Fig. 8-33:
Setting the communication mode

⑥ Select **TEL (FXCPU)**, enter the **Call number** of the remote modem and click **Connect**.


Connect line

Offline

Connect way: Auto

Callback number:

Line: Line type: Tone; Outside line: ; Port: COM1; Parity: None

Connection: Call number: 0123456789; Name:

AT command: ● Modem Standard; ○ AT command setting; Title:

Logging: ☑ Record the logging date

GSM: Wait time: 10

Fig. 8-34: *Set up the connection*

⑦ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Modem -> MAC -> FX Progr. Port

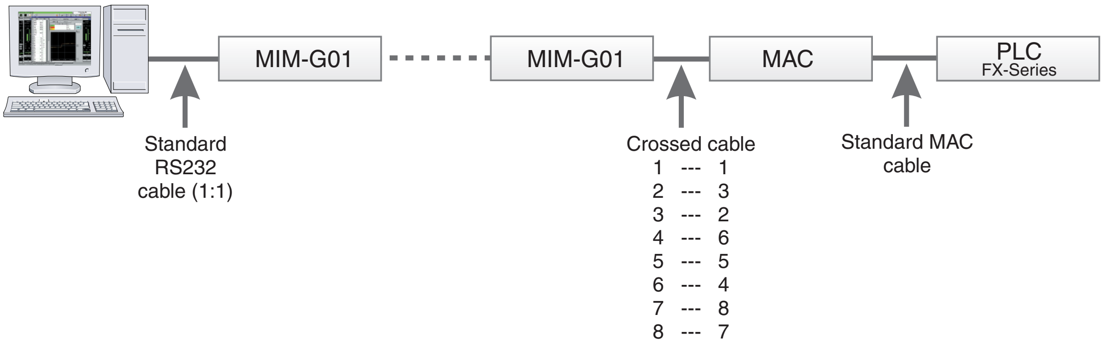


Fig. 8-35: *Environment*

GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        &W
        +CPIN=____  (PIN No. of SIM card)
```

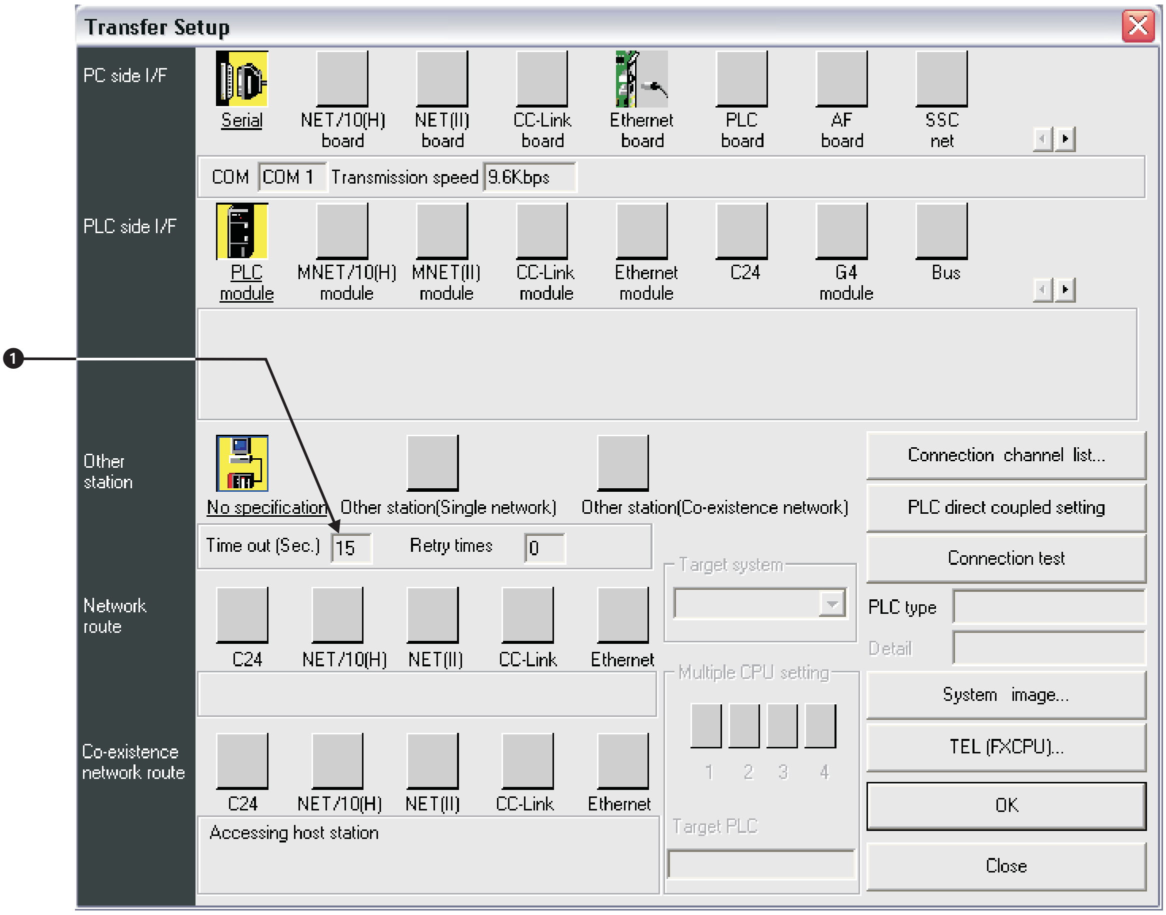
The modem which is used at the PC side has to be modified regarding the parity settings:

Parity:	Even
Data Bits:	7bits
Stop Bits:	1bit
AT	+ICF=5,1

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



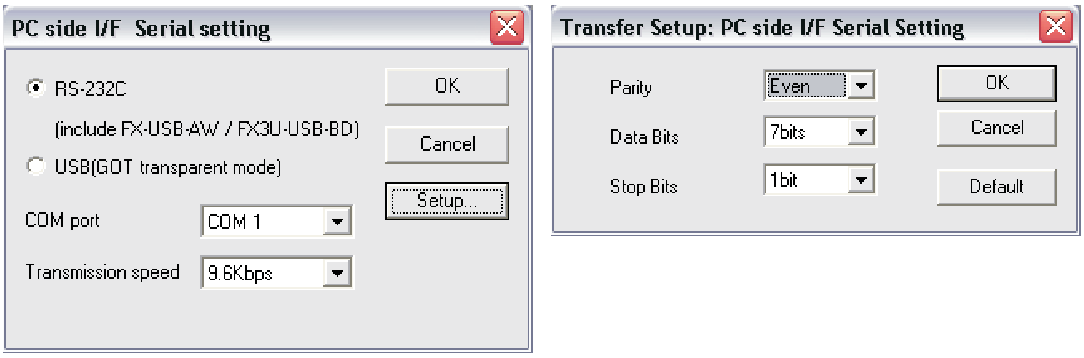
COM COM 1 Transmission speed 9.6Kbps

Time out (Sec.) 15 Retry times 0

Fig. 8-36: *Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.


PC side I/F Serial setting
OK
Cancel
Setup...
Transfer Setup: PC side I/F Serial Setting
OK
Cancel
Default

● RS-232C
(include FX-USB-AW / FX3U-USB-BD)
○ USB(GOT transparent mode)

COM port: COM 1

Transmission speed: 9.6Kbps

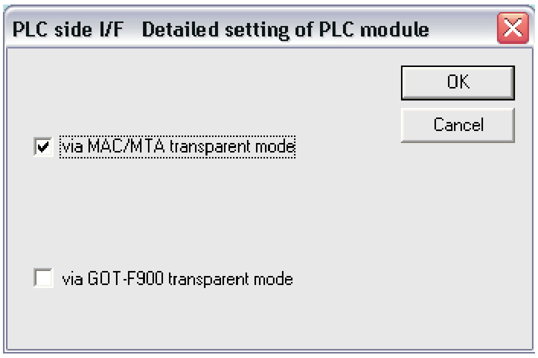
Parity: Even

Data Bits: 7bits

Stop Bits: 1bit

Fig. 8-37: *Setting the communication details*

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.

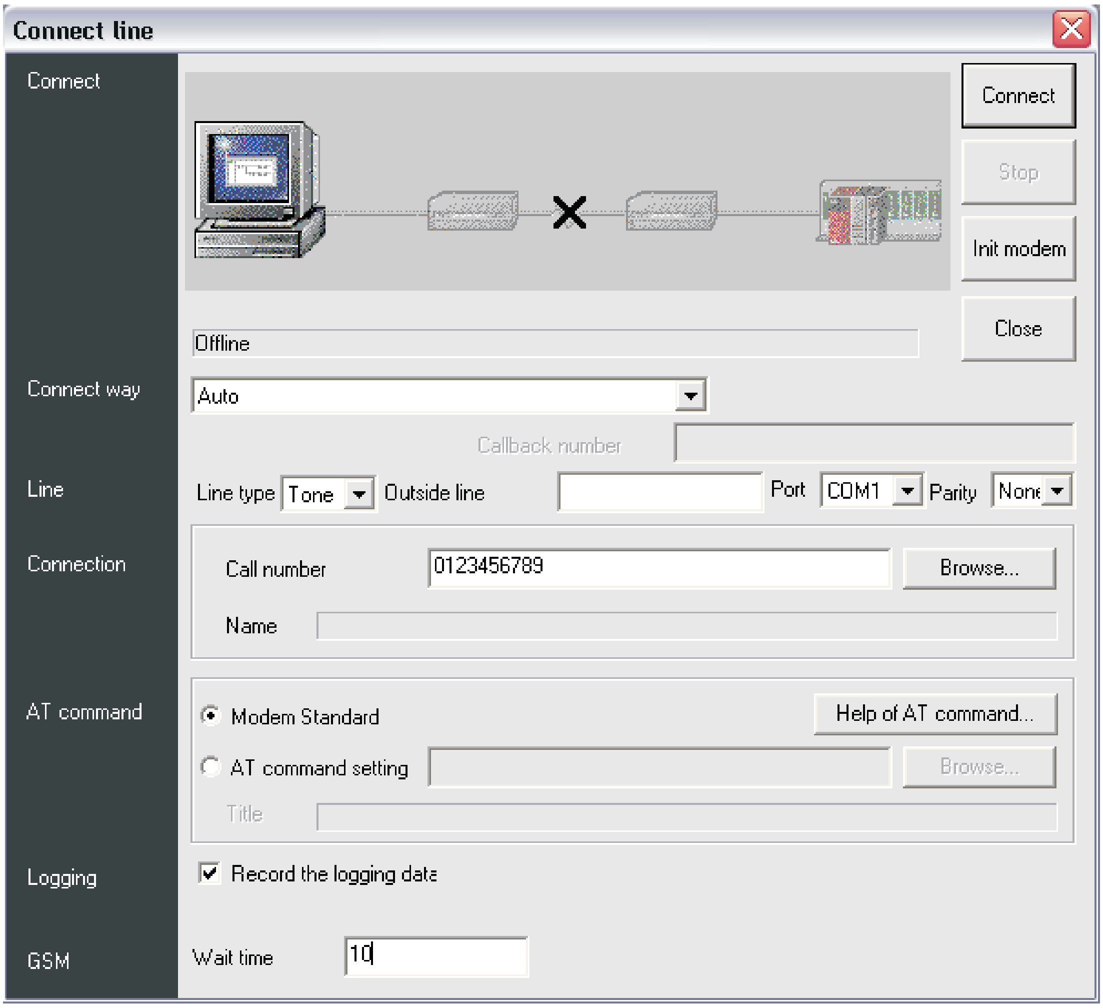

PLC side I/F Detailed setting of PLC module

☑ via MAC/MTA transparent mode

□ via GOT-F900 transparent mode

Fig. 8-38:
Setting the communication mode

⑥ Select **TEL (FXCPU)**, enter the **Call number** of the remote modem and click **Connect**.


Connect line

Connect: Offline

Connect way: Auto

Callback number

Line: Line type Tone Outside line Port COM1 Parity None

Connection: Call number 0123456789 Name

AT command: ● Modem Standard ○ AT command setting Title

Logging: ☑ Record the logging date

GSM: Wait time 10

Fig. 8-39: *Set up the connection*

⑦ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Alarm Modem -> FX Progr. Port

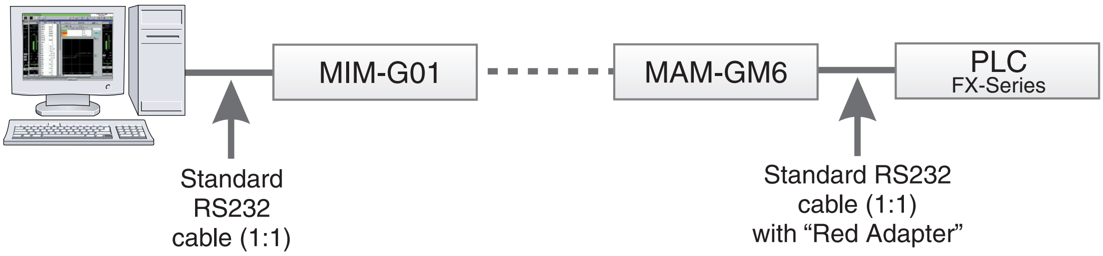

MIM-G01
MAM-GM6
PLC
FX-Series
Standard RS232 cable (1:1)
Standard RS232 cable (1:1) with “Red Adapter”

Fig. 8-40: *Test environment*

MIM-G01 settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        %C0
        \N0
        &W
        +CPIN=____  (PIN No. of SIM card)
```

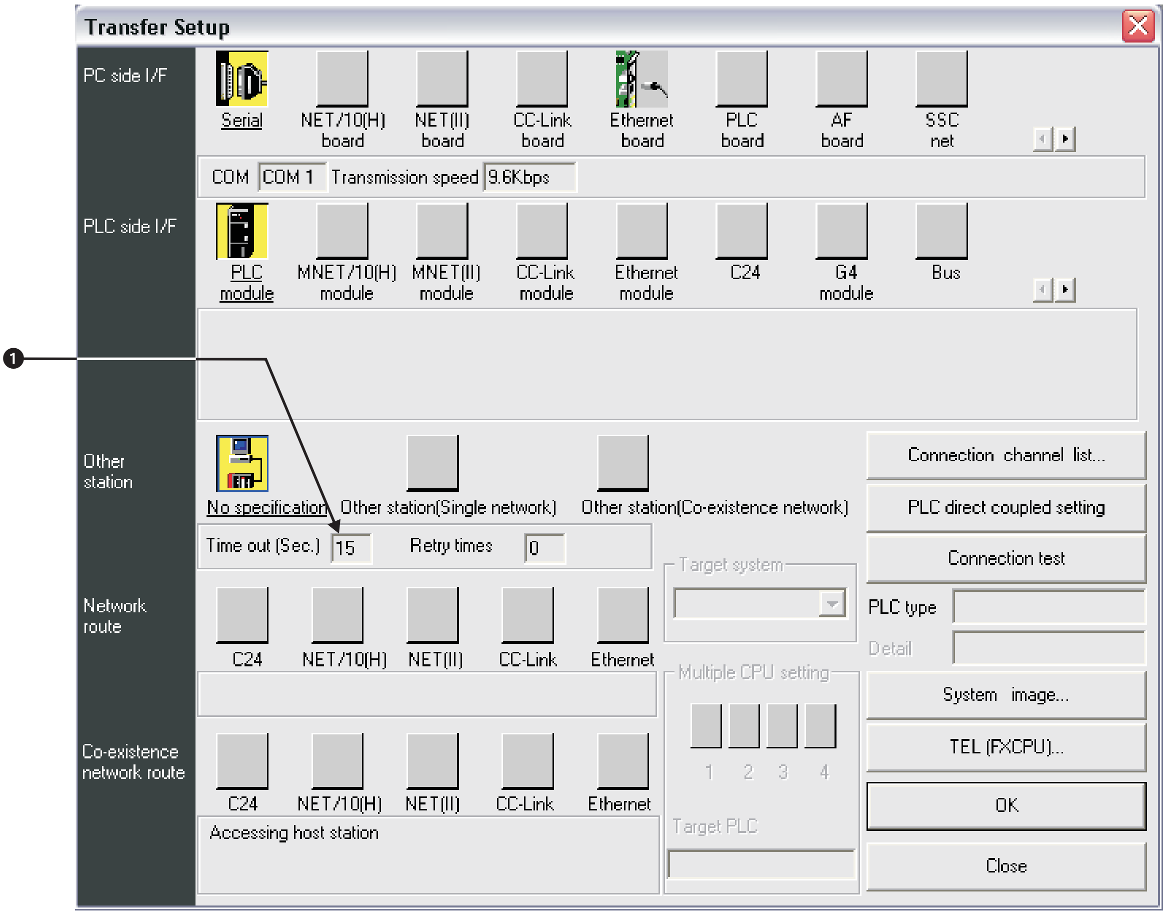
MAM-GM6 settings:

For the MAM-GM6 settings refer to MX Mitsubishi Alarm Editor

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

PC side I/F: Serial — COM: COM 1, Transmission speed: 9.6Kbps

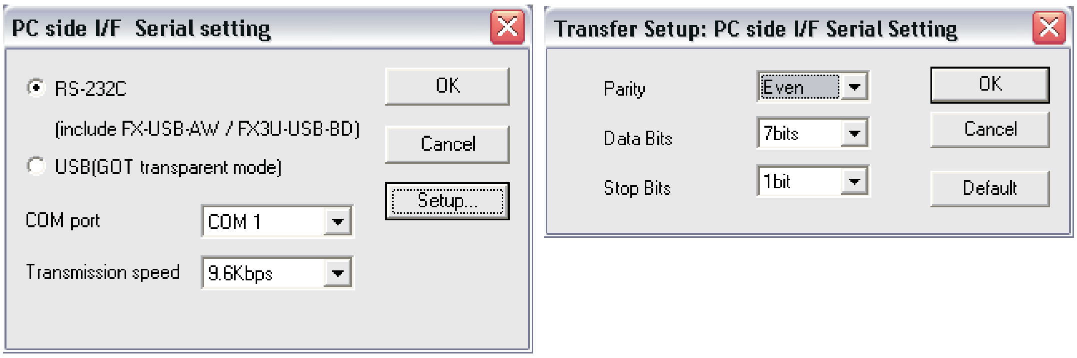
PLC side I/F: PLC module

Other station: No specification — Time out (Sec.): 15, Retry times: 0

Fig. 8-41: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

● RS-232C (include FX-USB-AW / FX3U-USB-BD)

○ USB(GOT transparent mode)

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

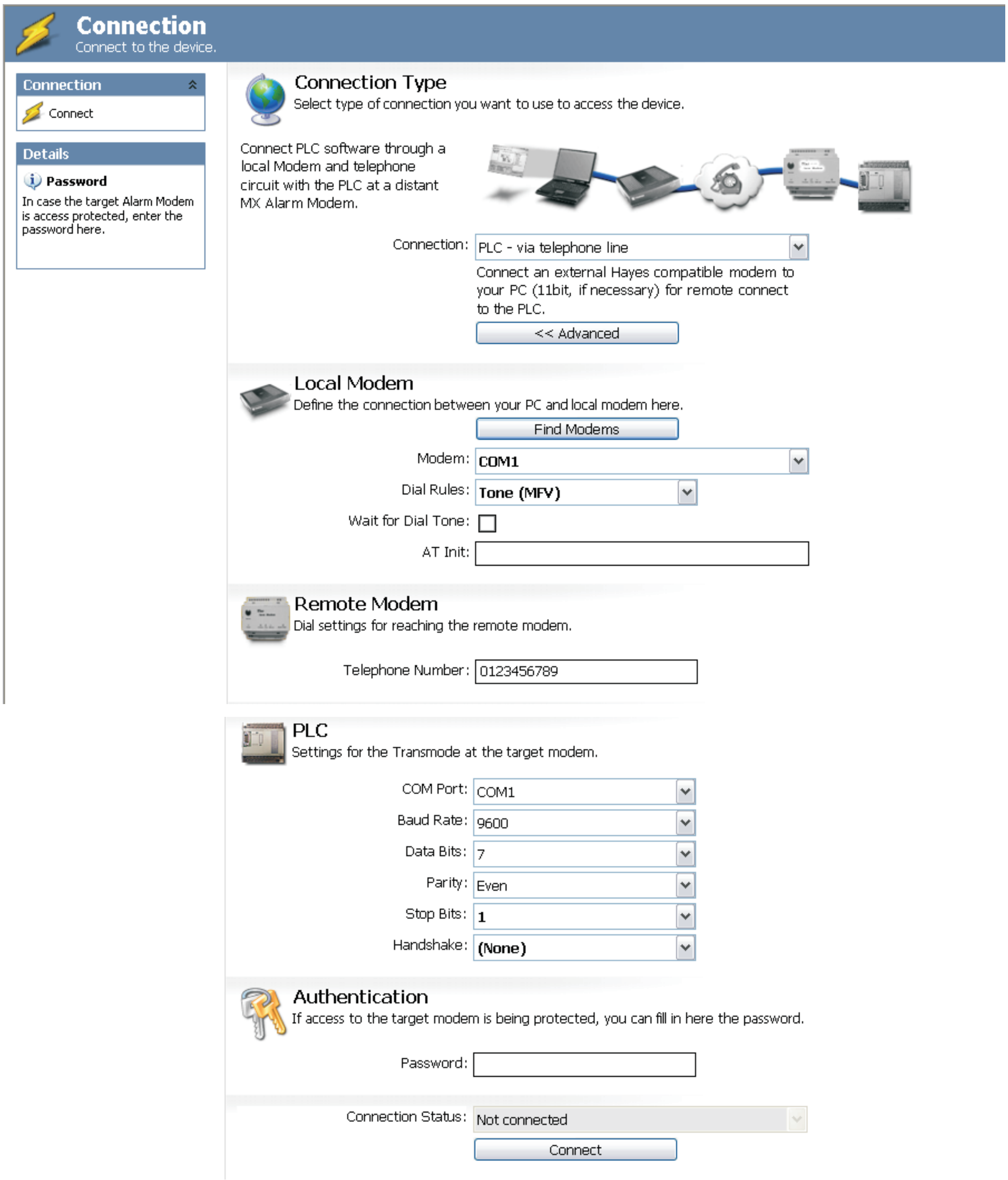
Parity: Even

Data Bits: 7bits

Stop Bits: 1bit

Fig. 8-42: *Setting the communication details*

⑤ To connect the modems you have to use the MX Mitsubishi Alarm Editor.
The following screenshot shows the settings.



Connection
Connect to the device.

Connection
Connect

Details
Password
In case the target Alarm Modem is access protected, enter the password here.

Connection Type
Select type of connection you want to use to access the device.

Connect PLC software through a local Modem and telephone circuit with the PLC at a distant MX Alarm Modem.

Connection: PLC - via telephone line
Connect an external Hayes compatible modem to your PC (11bit, if necessary) for remote connect to the PLC.
<< Advanced

Local Modem
Define the connection between your PC and local modem here.
Find Modems
Modem: COM1
Dial Rules: Tone (MFV)
Wait for Dial Tone: □
AT Init:

Remote Modem
Dial settings for reaching the remote modem.
Telephone Number: 0123456789

PLC
Settings for the Transmode at the target modem.
COM Port: COM1
Baud Rate: 9600
Data Bits: 7
Parity: Even
Stop Bits: 1
Handshake: (None)

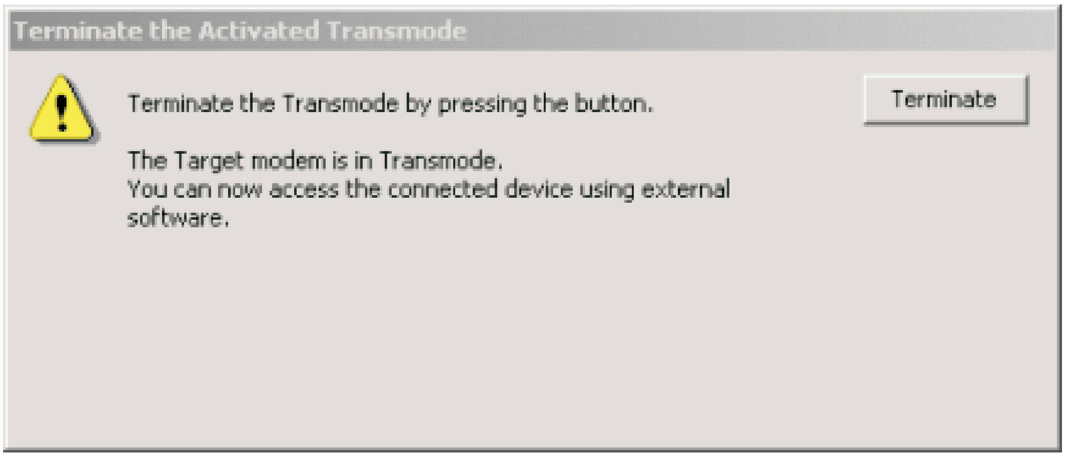
Authentication
If access to the target modem is being protected, you can fill in here the password.
Password:

Connection Status: Not connected
Connect

Fig. 8-43: *Settings in the MX Mitsubishi Alarm Editor*

⑥ To connect the modems click on the button **Connect**.

⑦ After the connection has been established the following message is displayed.



Terminate the Activated Transmode

Terminate the Transmode by pressing the button.

The Target modem is in Transmode.
You can now access the connected device using external software.

Fig. 8-44: *Connection message*

8.5.4 A-Series – Communication with C24

Settings for AJ71C24

Mode switch: 1 = Protocol 1

SW11 … SW24

SW11	OFF	=	RS232
SW12	ON	=	Data length = 8 bits
SW13	ON	=	Baud rate = 9600
SW14	OFF	=	
SW15	ON	=	
SW16	OFF	=	Parity check disabled
SW17	OFF	=	Not used
SW18	OFF	=	1 Stop bit
SW21	ON	=	Sum check enabled
SW22	ON	=	Write during RUN enabled
SW23	OFF	=	Not used
SW24	OFF	=	Not used

Settings for A1SJ71C24-R2

Mode switch: 1 = Protocol 1

SW03 … SW12

SW03	OFF	=	Not used
SW04	ON	=	Write during RUN enabled
SW05	ON	=	Baud rate = 9600
SW06	OFF	=	
SW07	ON	=	
SW08	ON	=	Data length = 8 bits
SW09	OFF	=	Parity check disabled
SW10	OFF	=	Not used
SW11	OFF	=	1 Stop bit
SW12	ON	=	Sum check enabled

Environment: Analogue Modem -> GSM Modem -> A(1S)J71(U)C24

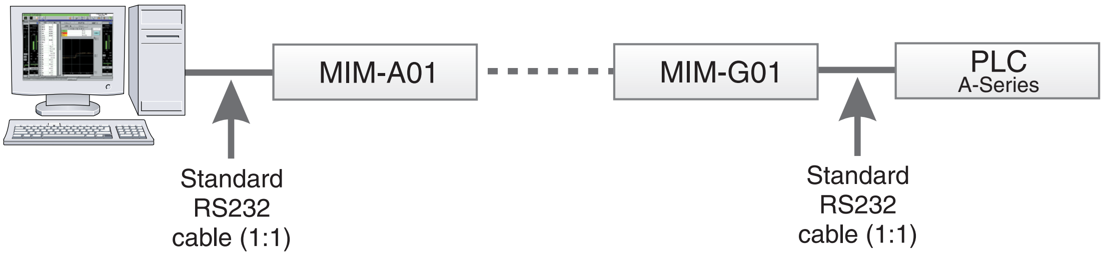


Fig. 8-45: *Environment*

Analogue Modem settings:

```
AT      E0
        &S0
        &D0
        &K0
```

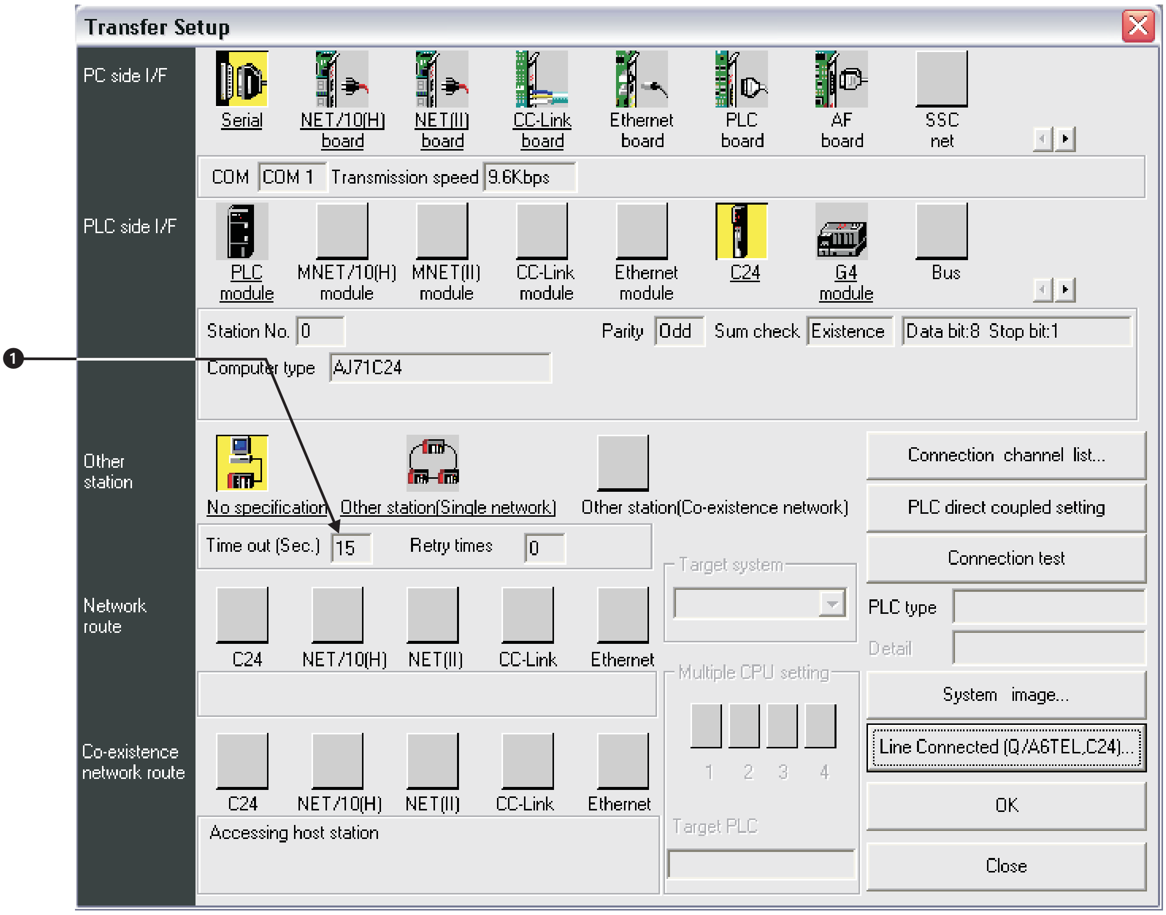
GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        +CPIN=____ (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



COM COM 1 Transmission speed 9.6Kbps

Station No. 0 Parity Odd Sum check Existence Data bit:8 Stop bit:1

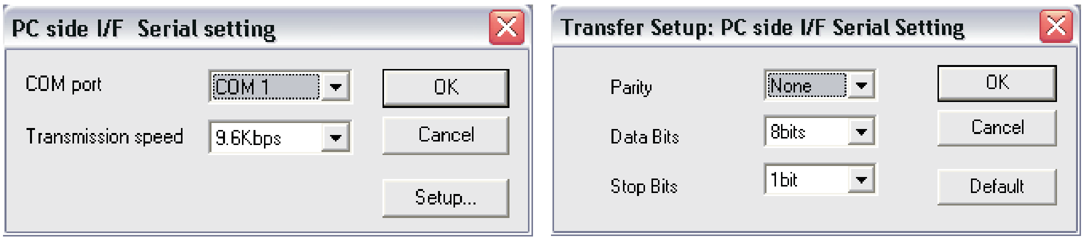
Computer type AJ71C24

Time out (Sec.) 15 Retry times 0

Fig. 8-46: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.


PC side I/F Serial setting
OK
Cancel
Setup...
Transfer Setup: PC side I/F Serial Setting
OK
Cancel
Default

COM port COM 1

Transmission speed 9.6Kbps

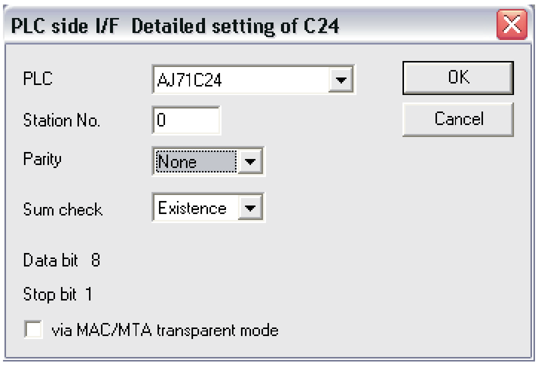
Parity None

Data Bits 8bits

Stop Bits 1bit

Fig. 8-47: *Setting the communication details*

⑤ Double click on **PLC side I/F - C24** to open the dialogue box **PLC side I/F Detailed setting of C24** and set the details.



PLC side I/F Detailed setting of C24

PLC: AJ71C24

Station No.: 0

Parity: None

Sum check: Existence

Data bit 8

Stop bit 1

□ via MAC/MTA transparent mode

Fig. 8-48:
Detailed setting for the C24

NOTE

To connect the modems, the WINDOWS Hyperterminal has to be used.
The GX IEC Developer function **Connect Line** does not work in this case.

Environment: GSM Modem -> GSM Modem -> A(1S)J71(U)C24

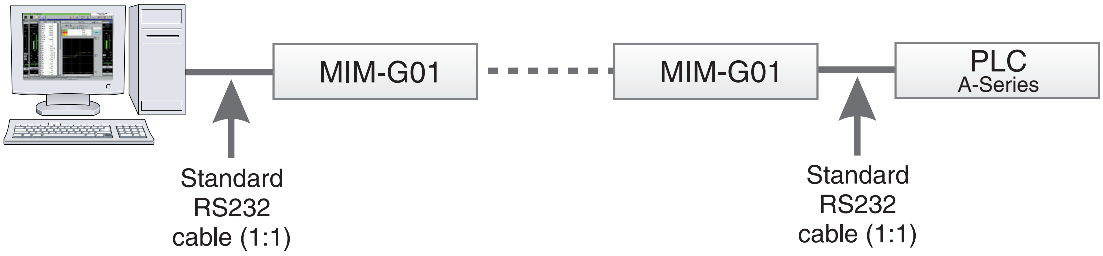


Fig. 8-49: Environment

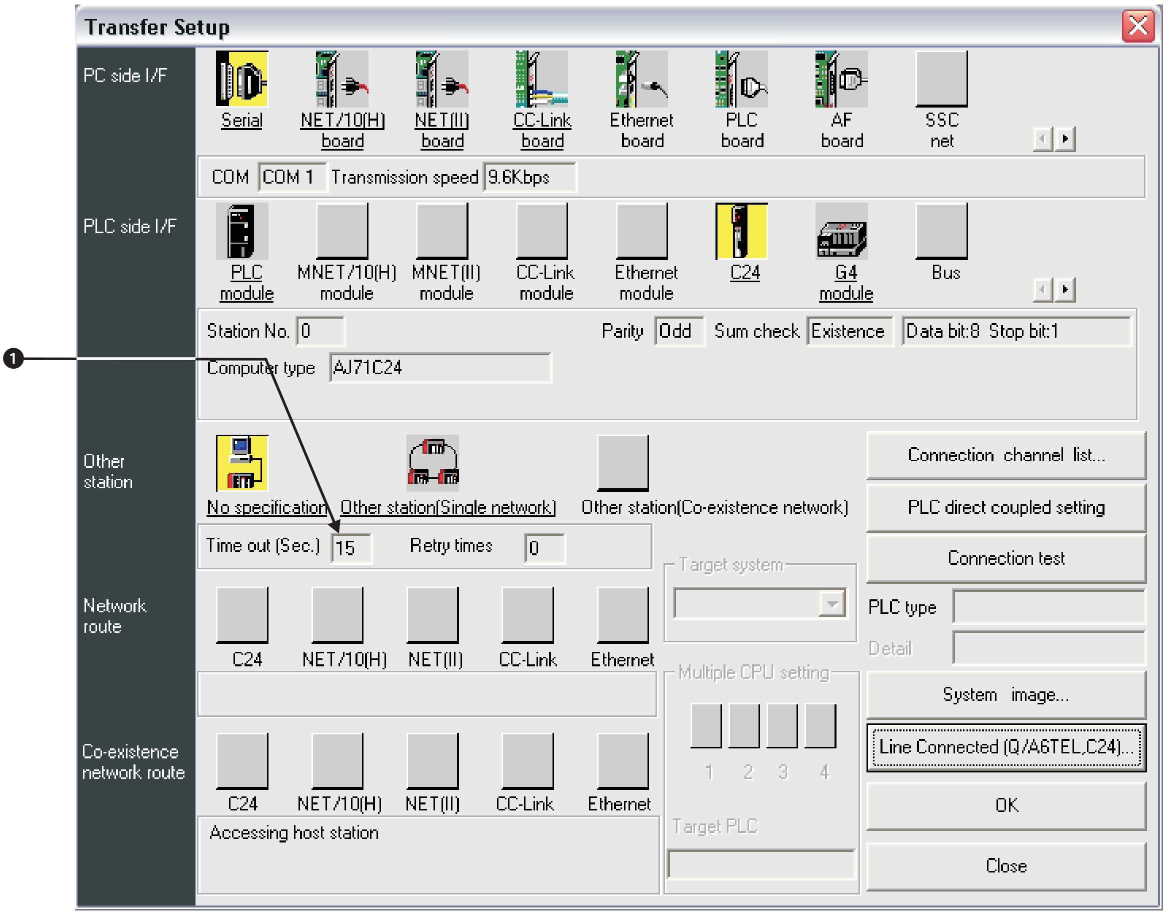
GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



COM COM 1 Transmission speed 9.6Kbps

Station No. 0 Parity Odd Sum check Existence Data bit:8 Stop bit:1

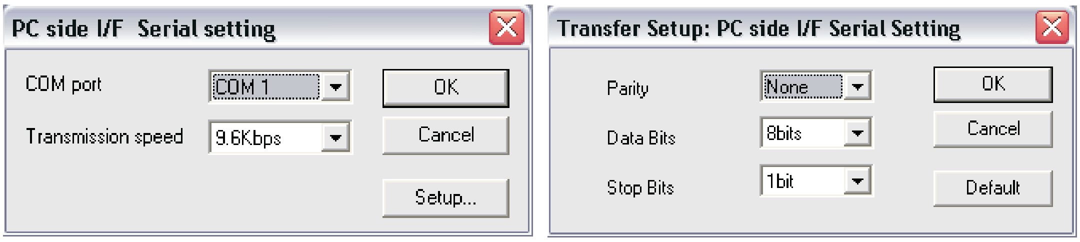
Computer type AJ71C24

Time out (Sec.) 15 Retry times 0

Fig. 8-50: *Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.


PC side I/F Serial setting
OK
Cancel
Setup...
Transfer Setup: PC side I/F Serial Setting
OK
Cancel
Default

COM port COM 1

Transmission speed 9.6Kbps

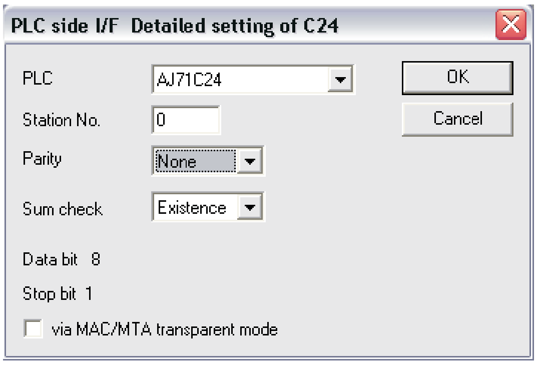
Parity None

Data Bits 8bits

Stop Bits 1bit

Fig: 8-51: *Setting the communication details*

⑤ Double click on **PLC side I/F - C24** to open the dialogue box **PLC side I/F Detailed setting of C24** and set the details.



PLC side I/F Detailed setting of C24

PLC: AJ71C24

Station No.: 0

Parity: None

Sum check: Existence

Data bit 8

Stop bit 1

□ via MAC/MTA transparent mode

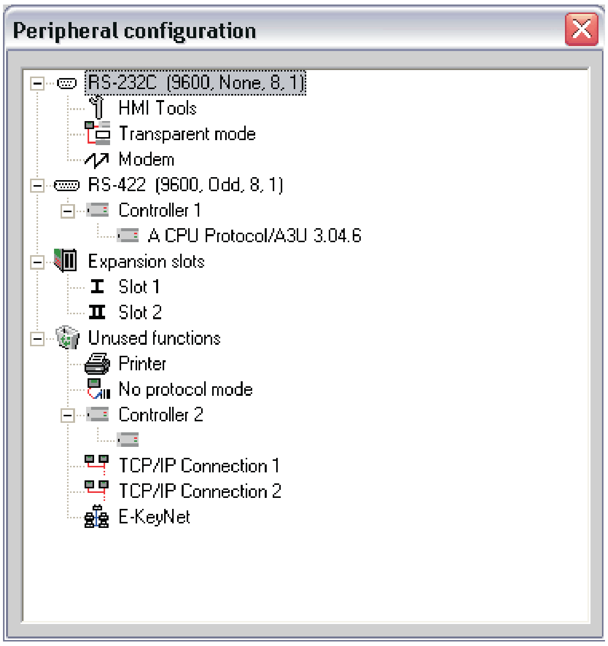
Fig. 8-52:
Detailed setting for the C24

NOTE

To connect the modems, the WINDOWS Hyperterminal has to be used.
The GX IEC Developer function **Connect Line** does not work in this case.

8.5.5 A-Series – Communication via MAC (Transparent Mode)

Settings for MAC Terminal:



Peripheral configuration

```
RS-232C (9600, None, 8, 1)
    HMI Tools
    Transparent mode
    Modem
RS-422 (9600, Odd, 8, 1)
    Controller 1
        A CPU Protocol/A3U 3.04.6
Expansion slots
    Slot 1
    Slot 2
Unused functions
    Printer
    No protocol mode
    Controller 2
    TCP/IP Connection 1
    TCP/IP Connection 2
    E-KeyNet
```

Fig. 8-53:
Setting the peripheral configuration in the E-Designer

Environment: Analogue Modem -> GSM Modem -> MAC -> A-Series Progr. Port

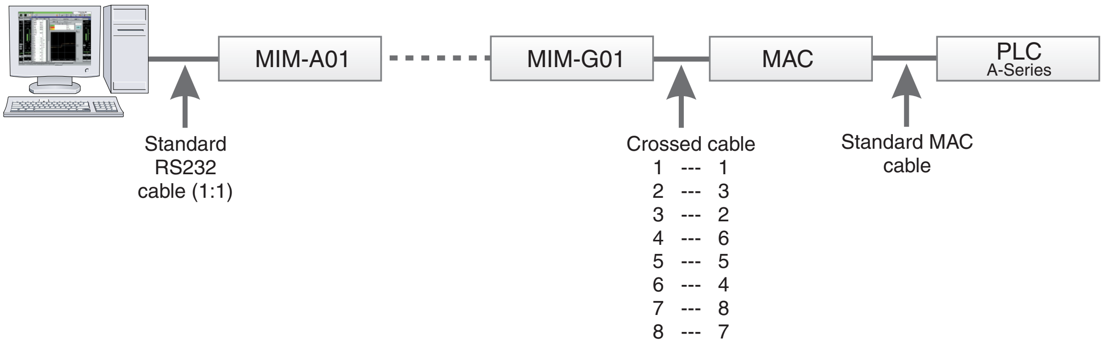


Fig. 8-54: *Environment*

Analogue Modem settings:

```
AT      E0
        &S0
        &D0
        &K0
```

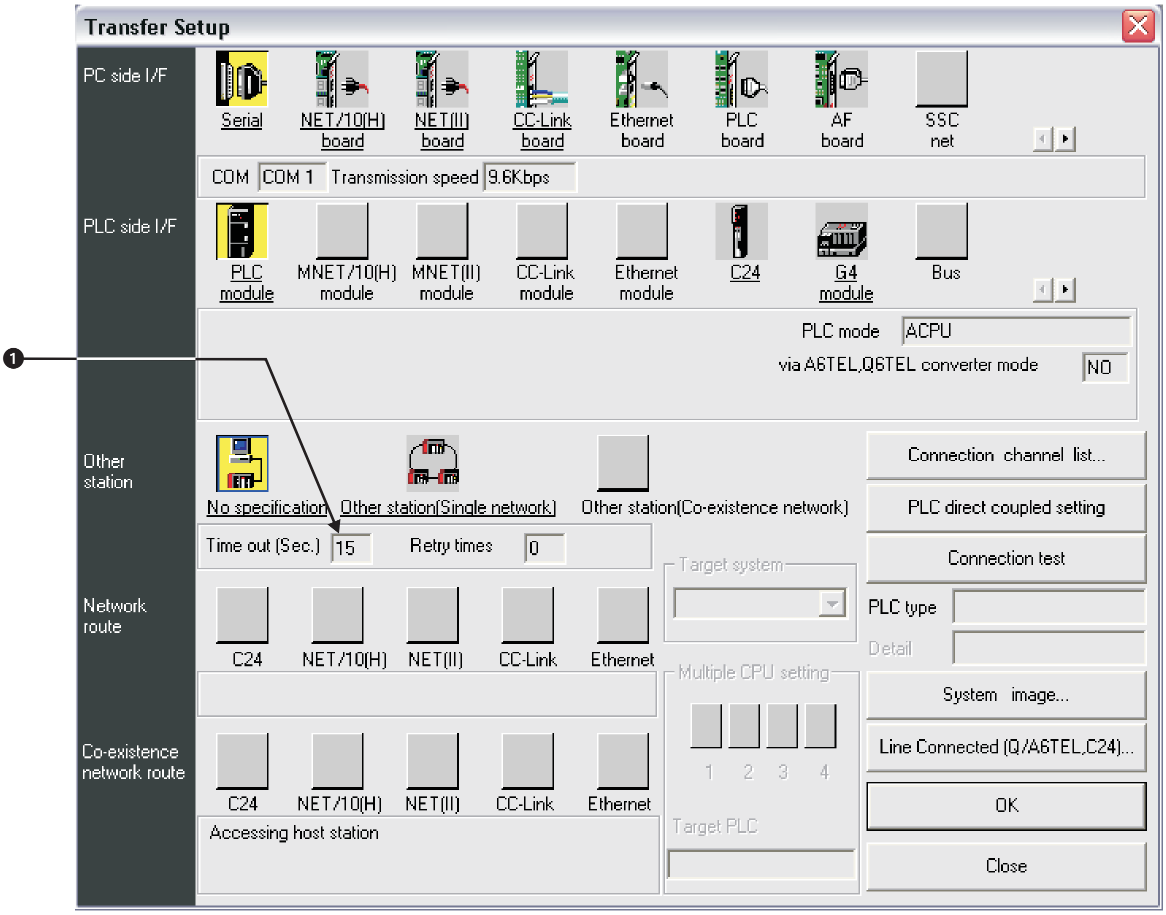
GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        &W
        +CPIN=____ (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

COM: COM 1 Transmission speed: 9.6Kbps

PLC mode: ACPU

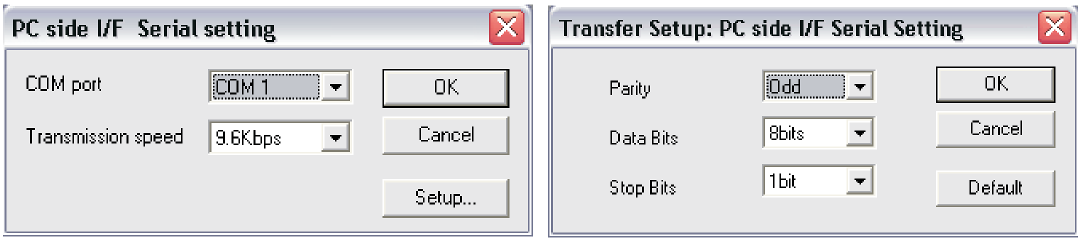
via A6TEL,Q6TEL converter mode: NO

Time out (Sec.): 15 Retry times: 0

*Fig. 8-55: Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

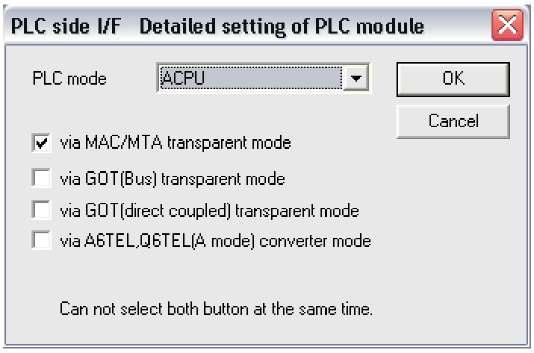
Parity: Odd

Data Bits: 8bits

Stop Bits: 1bit

Fig. 8-56: Setting the communication details

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.



PLC side I/F Detailed setting of PLC module

PLC mode: ACPU

☑ via MAC/MTA transparent mode

☐ via GOT(Bus) transparent mode

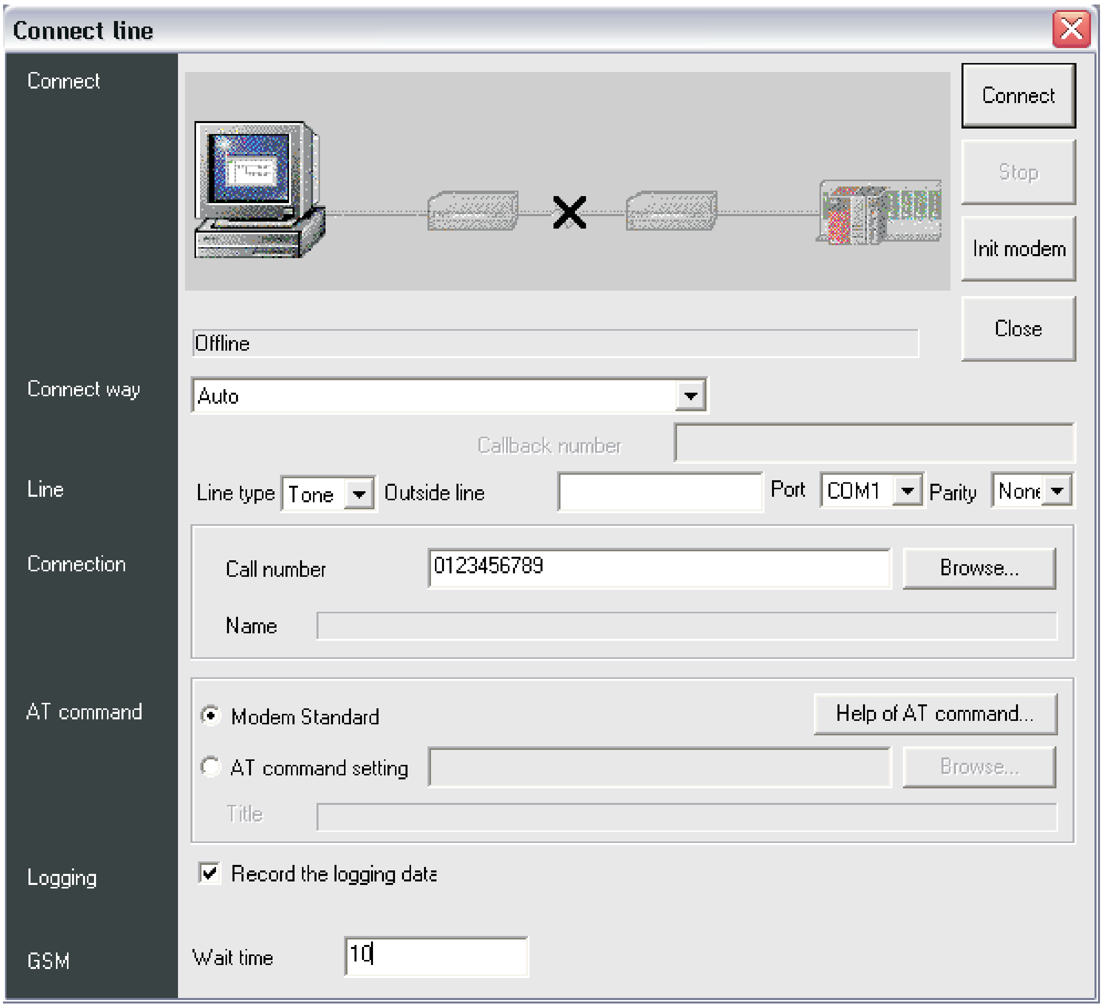
☐ via GOT(direct coupled) transparent mode

☐ via A6TEL,Q6TEL(A mode) converter mode

Can not select both button at the same time.

Fig. 8-57:
Setting the communication mode

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Connect: Offline

Connect way: Auto

Callback number

Line: Line type Tone, Outside line, Port COM1, Parity None

Connection: Call number 0123456789, Name

AT command: ● Modem Standard ○ AT command setting, Title

Logging: ☑ Record the logging date

GSM: Wait time 10

Fig. 8-58: *Set up the connection*

⑦ If the connection was successful, a message box shows:

– Connect completed

– Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Modem -> MAC -> A-Series Progr. Port

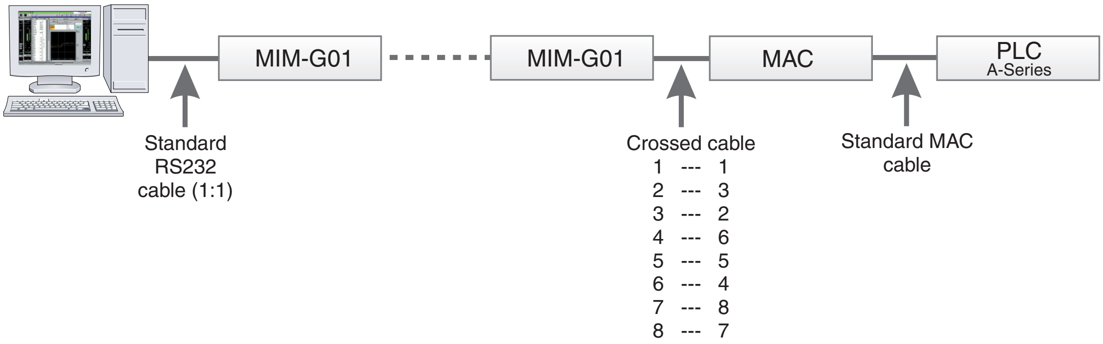


Fig. 8-59: *Environment*

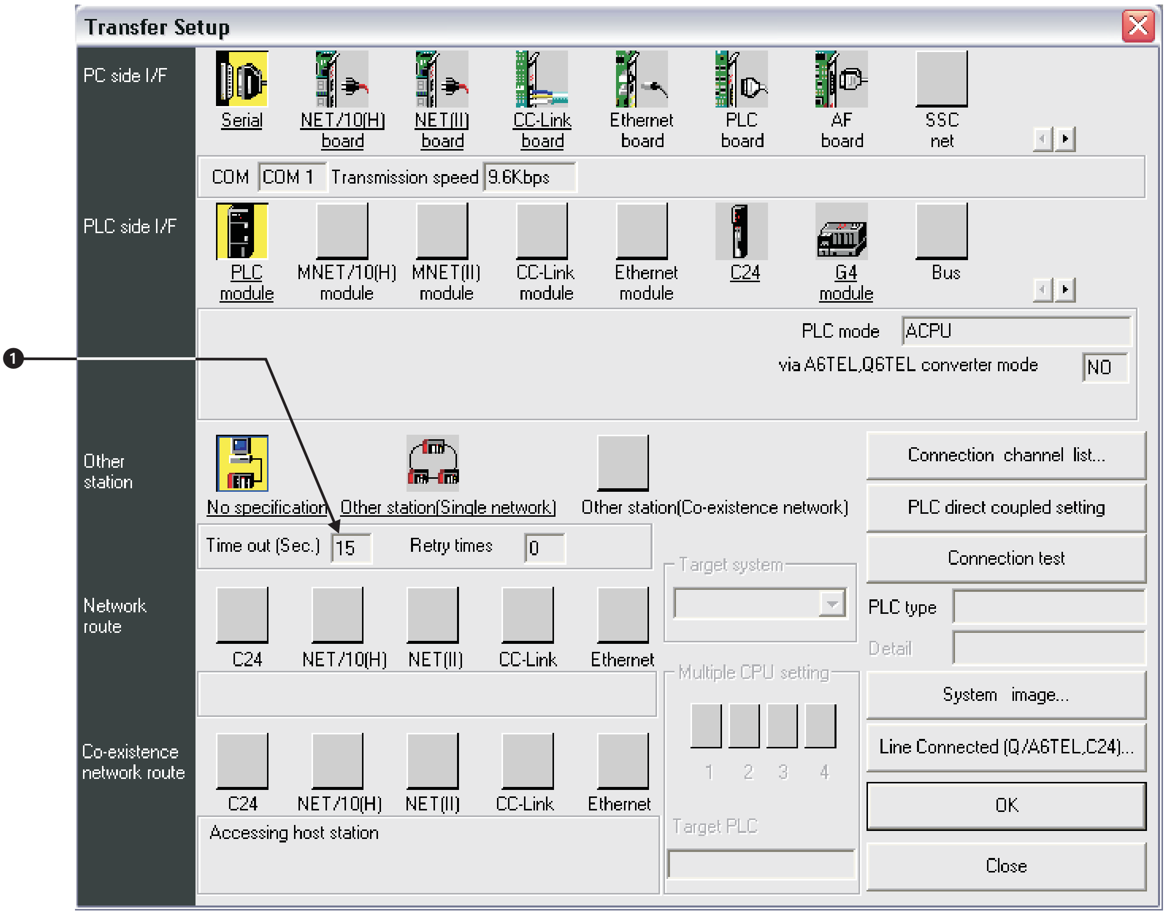
GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        &W
        +CPIN=____ (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

COM COM 1 Transmission speed 9.6Kbps

PLC mode ACPU

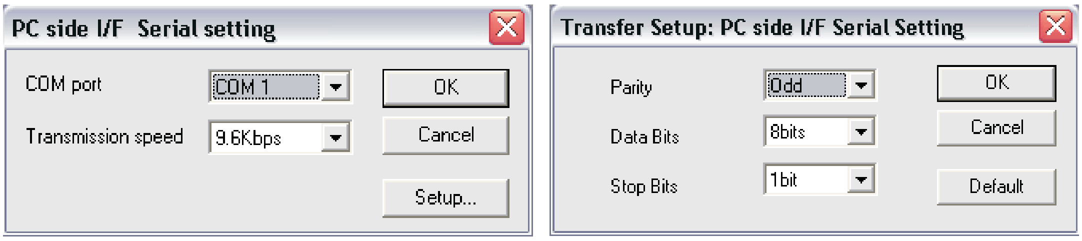
via A6TEL,Q6TEL converter mode NO

Time out (Sec.) 15 Retry times 0

Fig. 8-60: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

COM port COM 1

Transmission speed 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

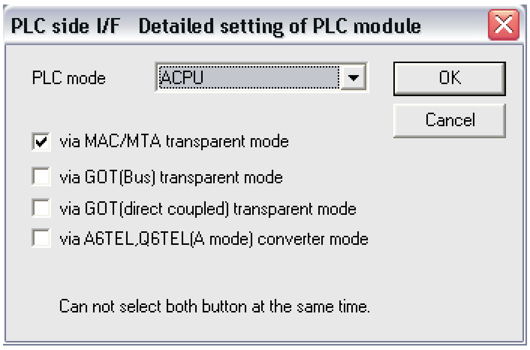
Parity Odd

Data Bits 8bits

Stop Bits 1bit

Fig. 8-61: *Setting the communication details*

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.



PLC side I/F Detailed setting of PLC module

PLC mode ACPU

☑ via MAC/MTA transparent mode

□ via GOT(Bus) transparent mode

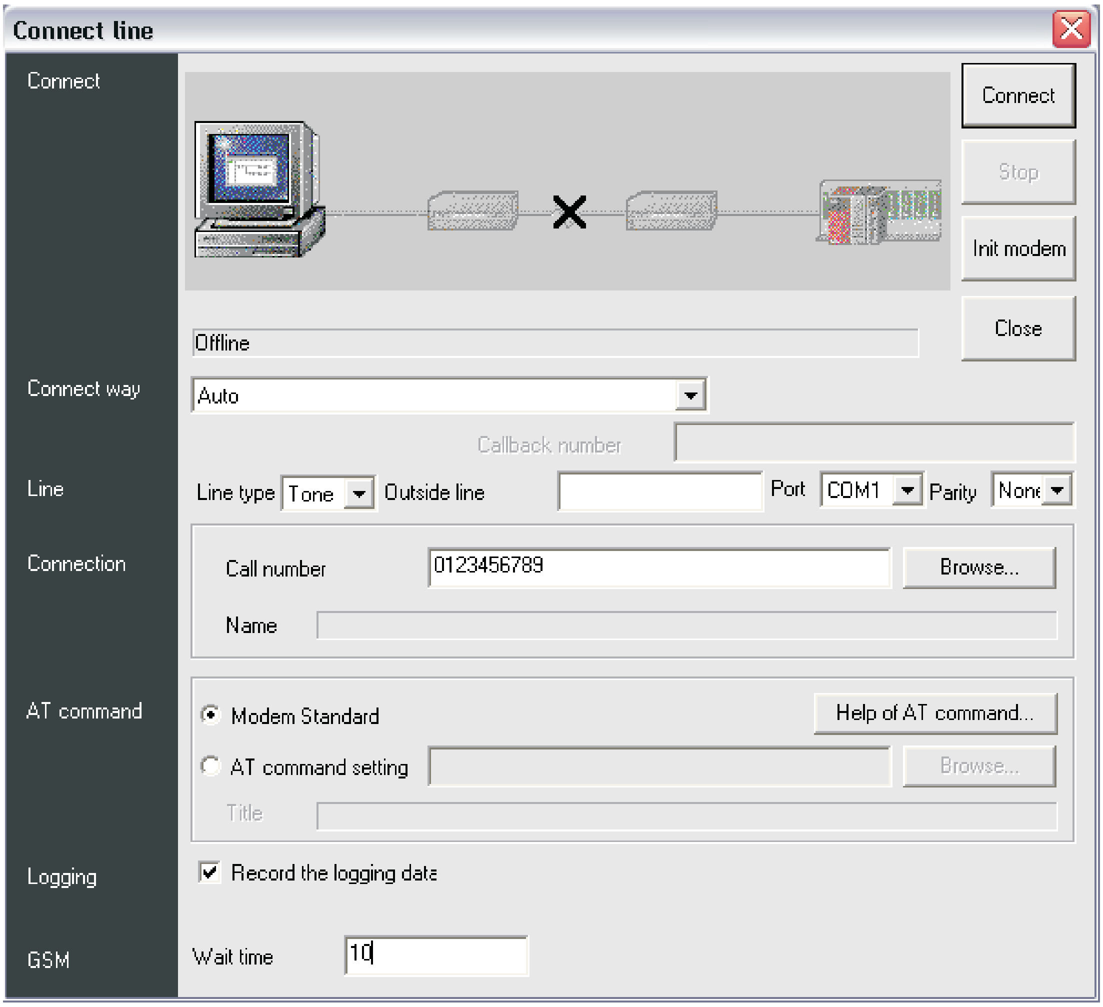
□ via GOT(direct coupled) transparent mode

□ via A6TEL,Q6TEL(A mode) converter mode

Can not select both button at the same time.

Fig. 8-62:
Setting the communication mode

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Connect: Offline

Connect way: Auto

Callback number

Line: Line type Tone Outside line Port COM1 Parity None

Connection: Call number 0123456789 Name

AT command: ● Modem Standard ○ AT command setting Title

Logging: ☑ Record the logging data

GSM: Wait time 10

Fig. 8-63: *Set up the connection*

⑦ If the connection was successful, a message box shows:

– Connect completed

– Destination – 0123456789 ← Call number of your remote modem

8.5.6 QnA-Series – Communication with C24

Settings for AJ71QC24N or A1SJ71QC24-R2

Mode switch: 5 = Binary mode Format 5

SW01 … SW12

SW01	OFF	=	always OFF
SW02	ON	=	Data length = 8 bits
SW03	OFF	=	Parity check disabled
SW04	OFF	=	Not used
SW05	OFF	=	1 Stop bit
SW06	ON	=	Sum check enabled
SW07	ON	=	Write during RUN enabled
SW08	ON	=	Changes enabled (allow)
SW09	ON	=	Baud rate = 9600
SW10	OFF	=	
SW11	ON	=	
SW12	OFF	=	

Environment: Analogue Modem -> GSM Modem -> A(1S)J71QC24

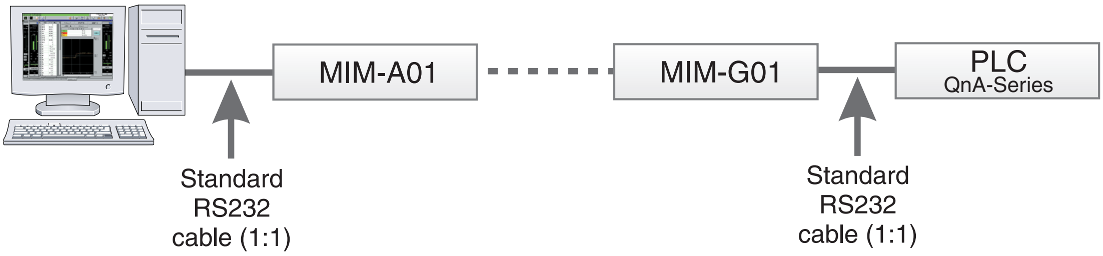


Fig. 8-64: *Environment*

Analogue Modem settings:

```
AT    E0
      &S0
      &D0
      &K0
```

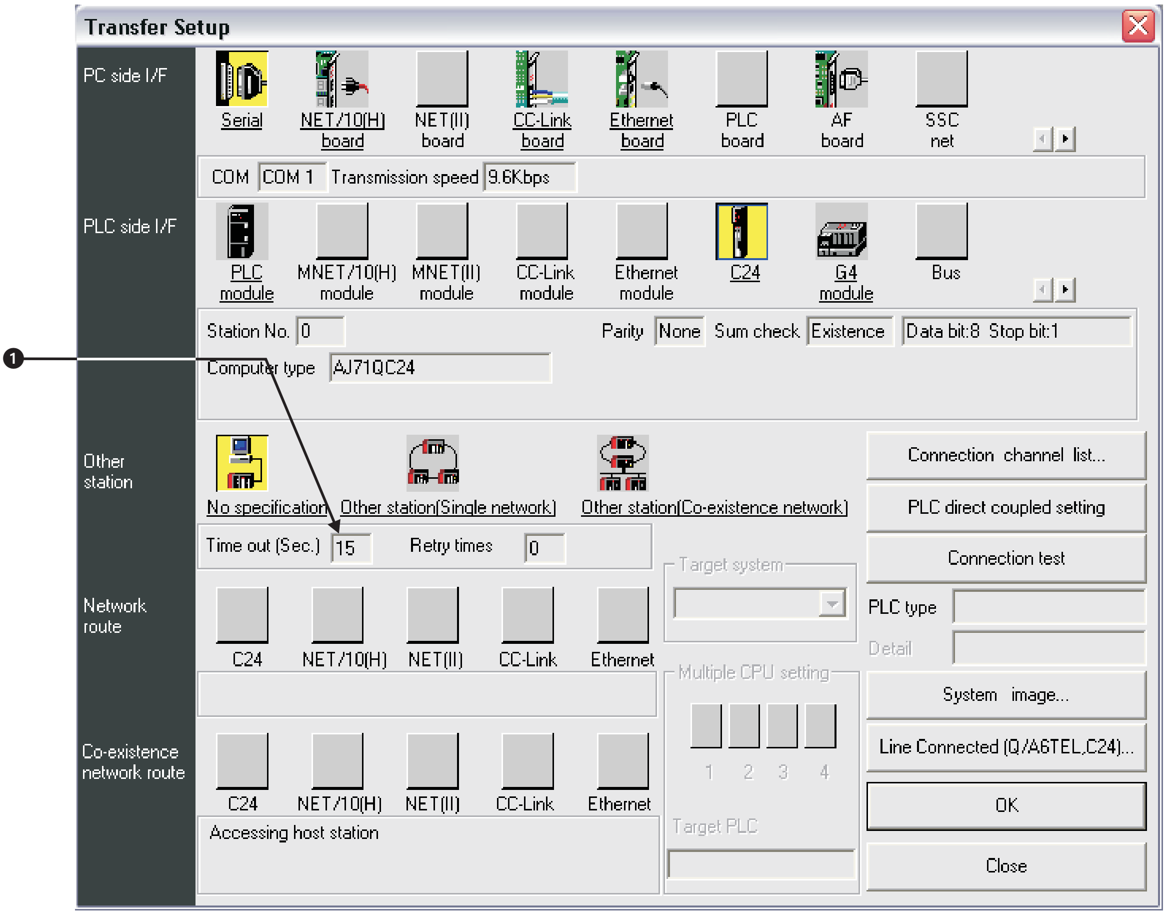
GSM Modem settings:

```
AT    E0
      S0=2
      &S0
      &D0
      +CICB=0
      +IFC=0,0
      +CMEE=1
      +IPR=9600
      +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

COM: COM 1 Transmission speed: 9.6Kbps

Station No.: 0 Parity: None Sum check: Existence Data bit:8 Stop bit:1

Computer type: AJ71QC24

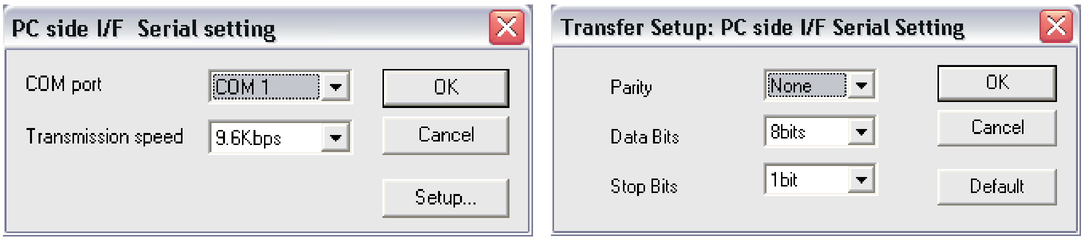
Time out (Sec.): 15 Retry times: 0

Accessing host station

Fig. 8-65: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

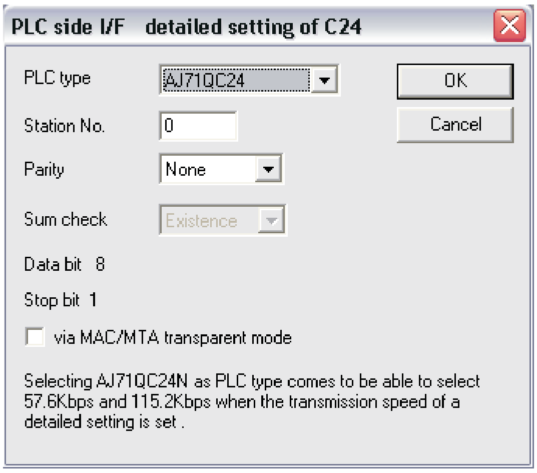
Parity: None

Data Bits: 8bits

Stop Bits: 1bit

Fig. 8-66: *Setting the communication details*

⑤ Double click on **PLC side I/F - C24** to open the dialogue box **PLC side I/F Detailed setting of C24** and set the details.



PLC side I/F detailed setting of C24

PLC type: AJ71QC24

Station No.: 0

Parity: None

Sum check: Existence

Data bit 8

Stop bit 1

□ via MAC/MTA transparent mode

Selecting AJ71QC24N as PLC type comes to be able to select 57.6Kbps and 115.2Kbps when the transmission speed of a detailed setting is set .

Fig. 8-67:
Detailed setting for the C24

NOTE

To connect the modems, the WINDOWS Hyperterminal has to be used.
The GX IEC Developer function **Connect Line** does not work in this case.

Environment: GSM Modem -> GSM Modem -> A(1S)J71QC24

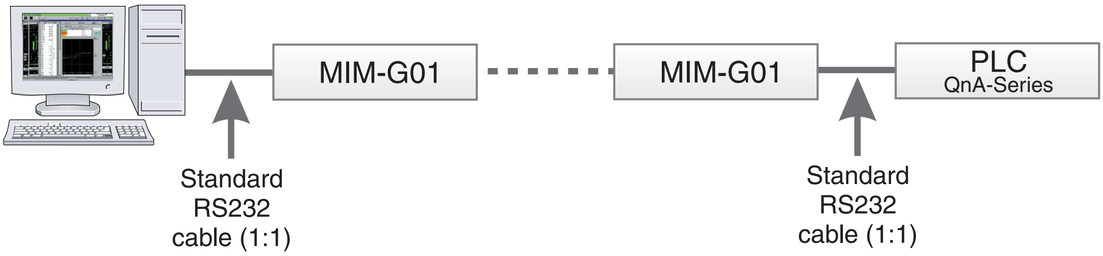


Fig. 8-68: *Environment*

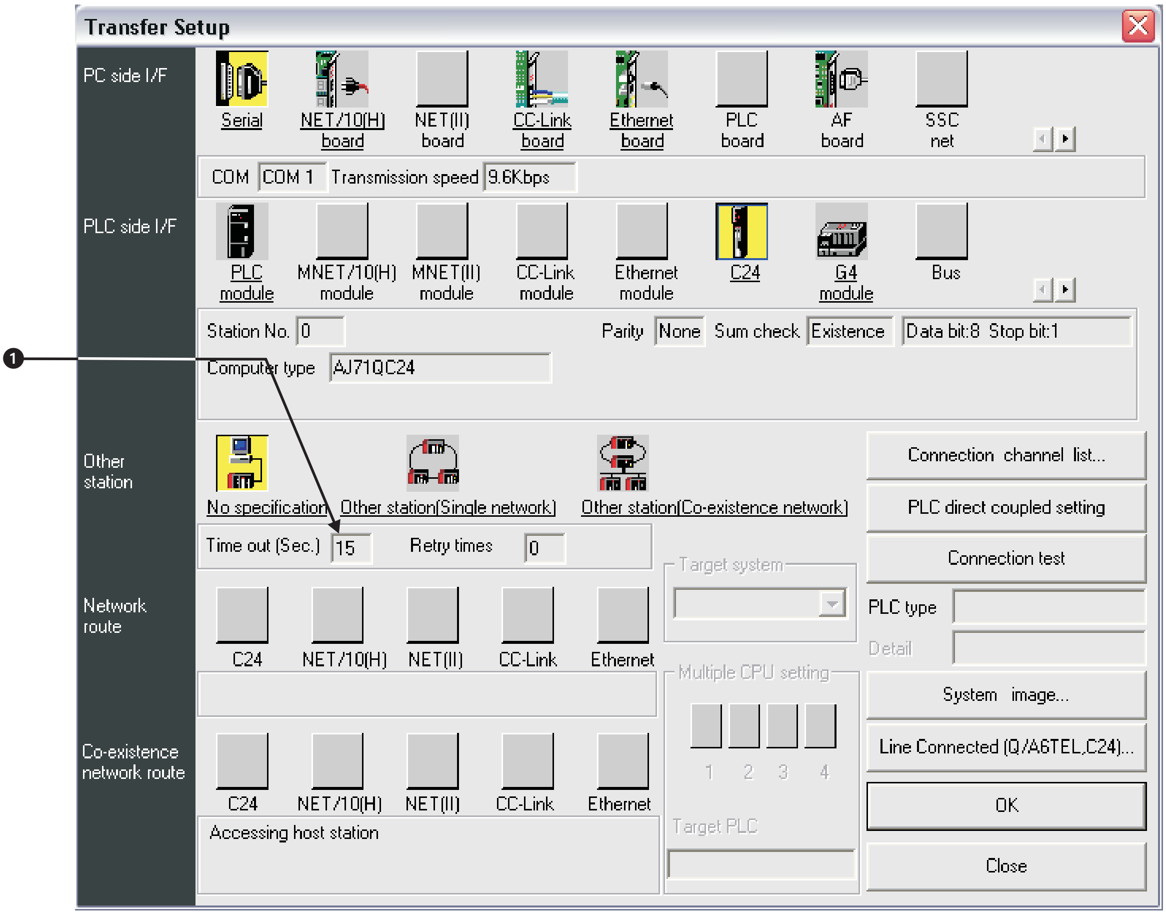
GSM Modem settings:

```
AT        E0
          S0=2
          &S0
          &D0
          +CICB=0
          +IFC=0,0
          +CMEE=1
          +IPR=9600
          +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

PC side I/F: Serial; COM: COM 1; Transmission speed: 9.6Kbps

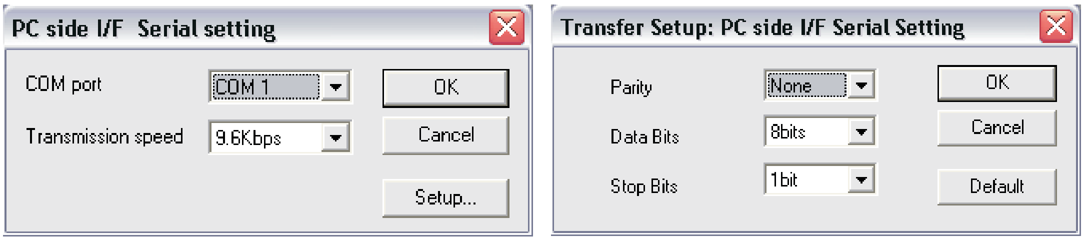
PLC side I/F: C24; Station No.: 0; Parity: None; Sum check: Existence; Data bit:8 Stop bit:1; Computer type: AJ71QC24

Other station: No specification; Time out (Sec.): 15; Retry times: 0

Fig. 8-69: *Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

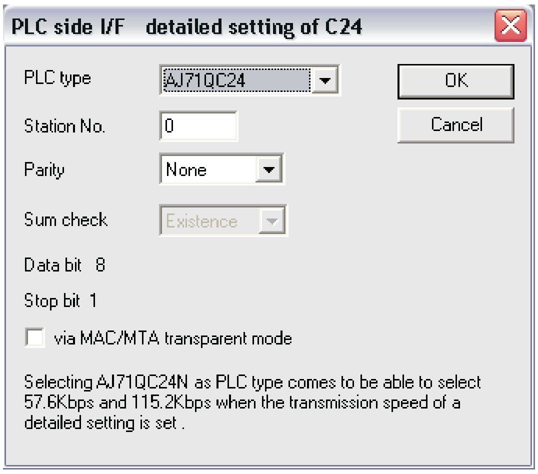
Parity: None

Data Bits: 8bits

Stop Bits: 1bit

Fig. 8-70: *Setting the communication details*

⑤ Double click on **PLC side I/F - C24** to open the dialogue box **PLC side I/F Detailed setting of C24** and set the details.



PLC side I/F detailed setting of C24

PLC type: AJ71QC24

Station No.: 0

Parity: None

Sum check: Existence

Data bit 8

Stop bit 1

□ via MAC/MTA transparent mode

Selecting AJ71QC24N as PLC type comes to be able to select 57.6Kbps and 115.2Kbps when the transmission speed of a detailed setting is set .

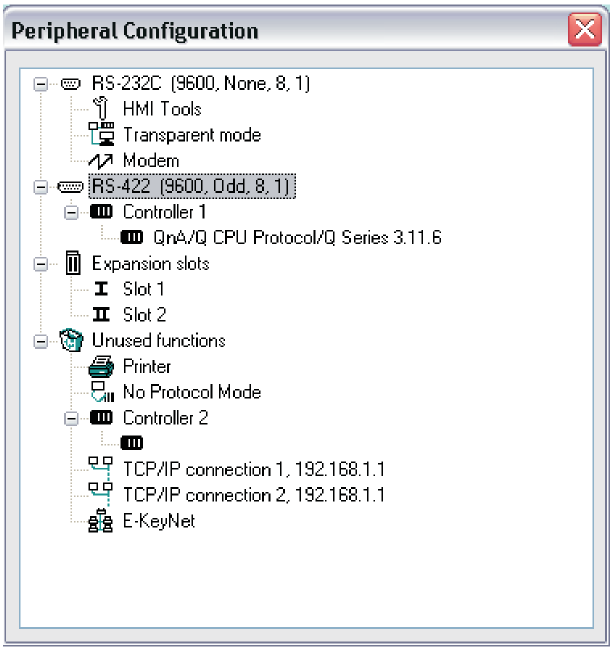
Fig. 8-71:
Detailed setting for the C24

NOTE

To connect the modems, the WINDOWS Hyperterminal has to be used.
The GX IEC Developer function **Connect Line** does not work in this case.

8.5.7 QnA-Series – Communication via MAC (Transparent Mode)

Settings for MAC Terminal:



Peripheral Configuration

```
RS-232C (9600, None, 8, 1)
    HMI Tools
    Transparent mode
    Modem
RS-422 (9600, Odd, 8, 1)
    Controller 1
        QnA/Q CPU Protocol/Q Series 3.11.6
Expansion slots
    Slot 1
    Slot 2
Unused functions
    Printer
    No Protocol Mode
    Controller 2

    TCP/IP connection 1, 192.168.1.1
    TCP/IP connection 2, 192.168.1.1
    E-KeyNet
```

Fig. 8-72:
Setting the peripheral configuration in the E-Designer

Environment: Analogue Modem -> GSM Modem -> MAC -> QnA-Series Progr. Port

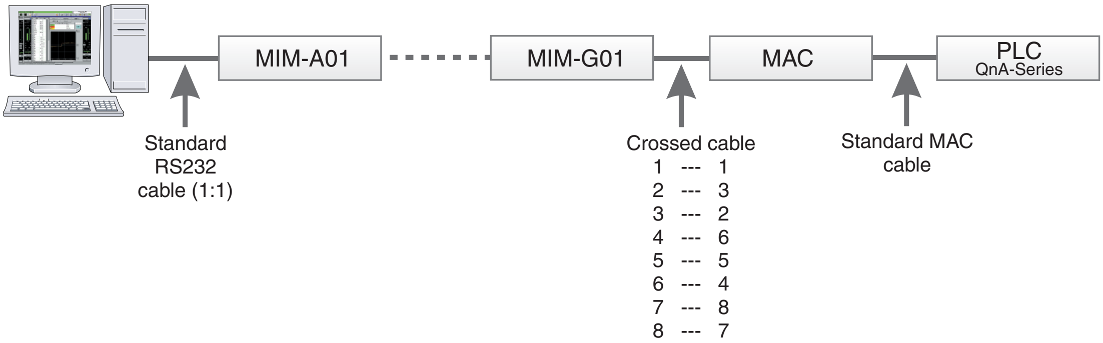


Fig. 8-73: Environment

Analogue Modem settings:

```
AT      E0
        &S0
        &D0
        &K0
```

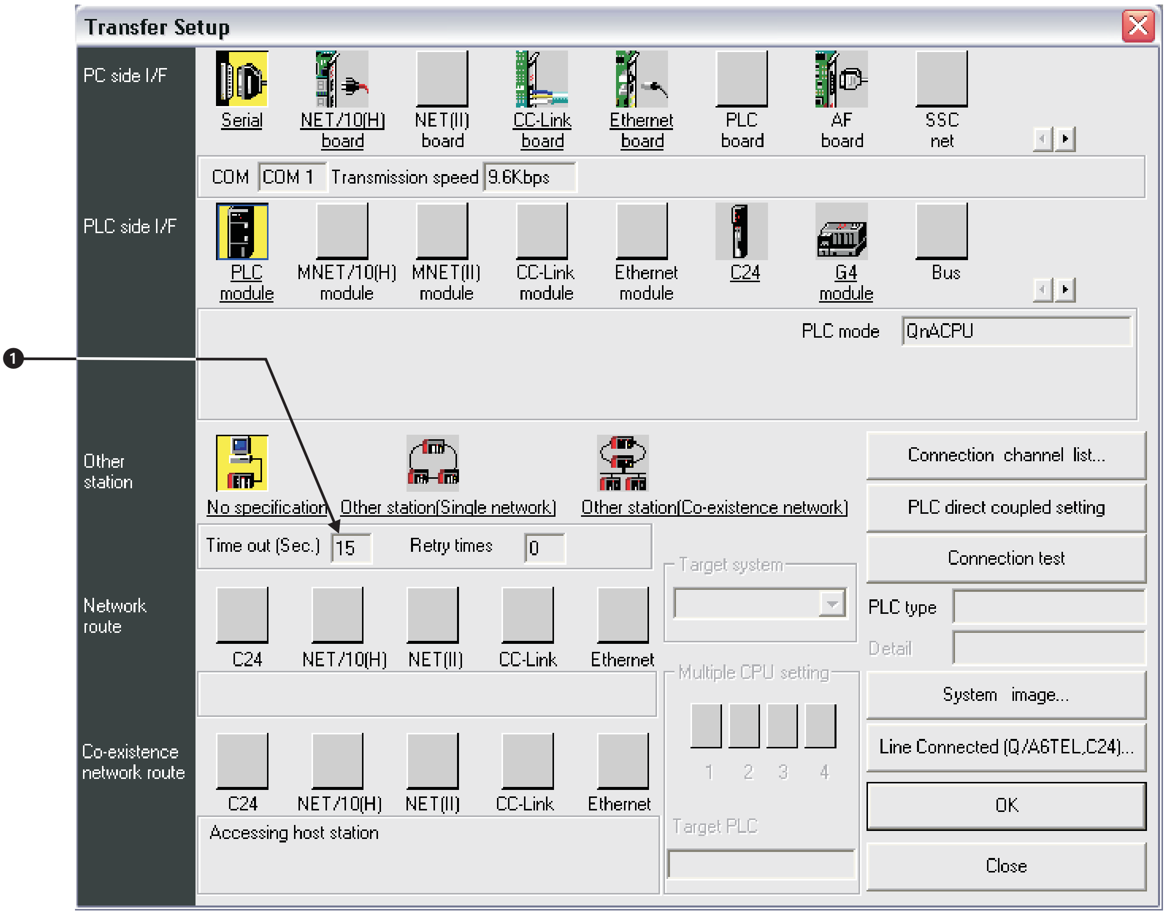
GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        &W
        +CPIN=____      (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

COM COM 1 Transmission speed 9.6Kbps

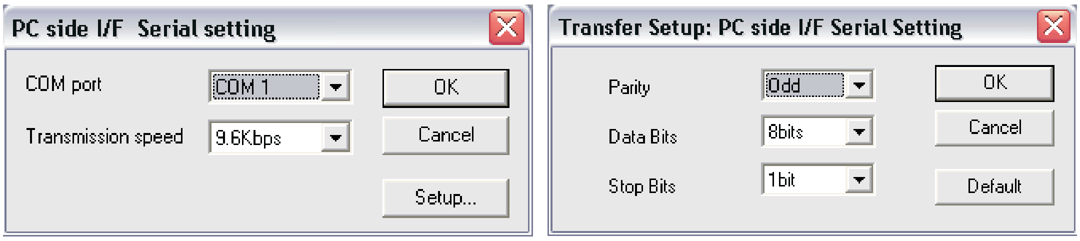
PLC mode QnACPU

Time out (Sec.) 15 Retry times 0

Fig. 8-74: *Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

COM port COM 1

Transmission speed 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

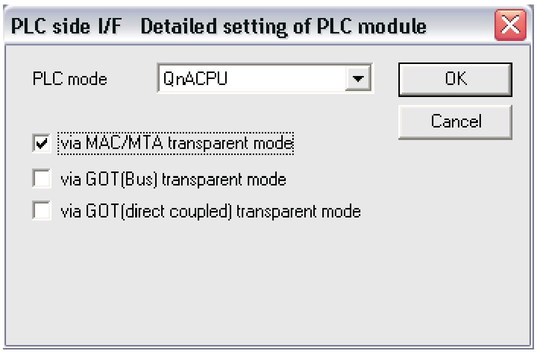
Parity Odd

Data Bits 8bits

Stop Bits 1bit

Fig. 8-75: *Setting the communication details*

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.



PLC side I/F Detailed setting of PLC module

PLC mode QnACPU

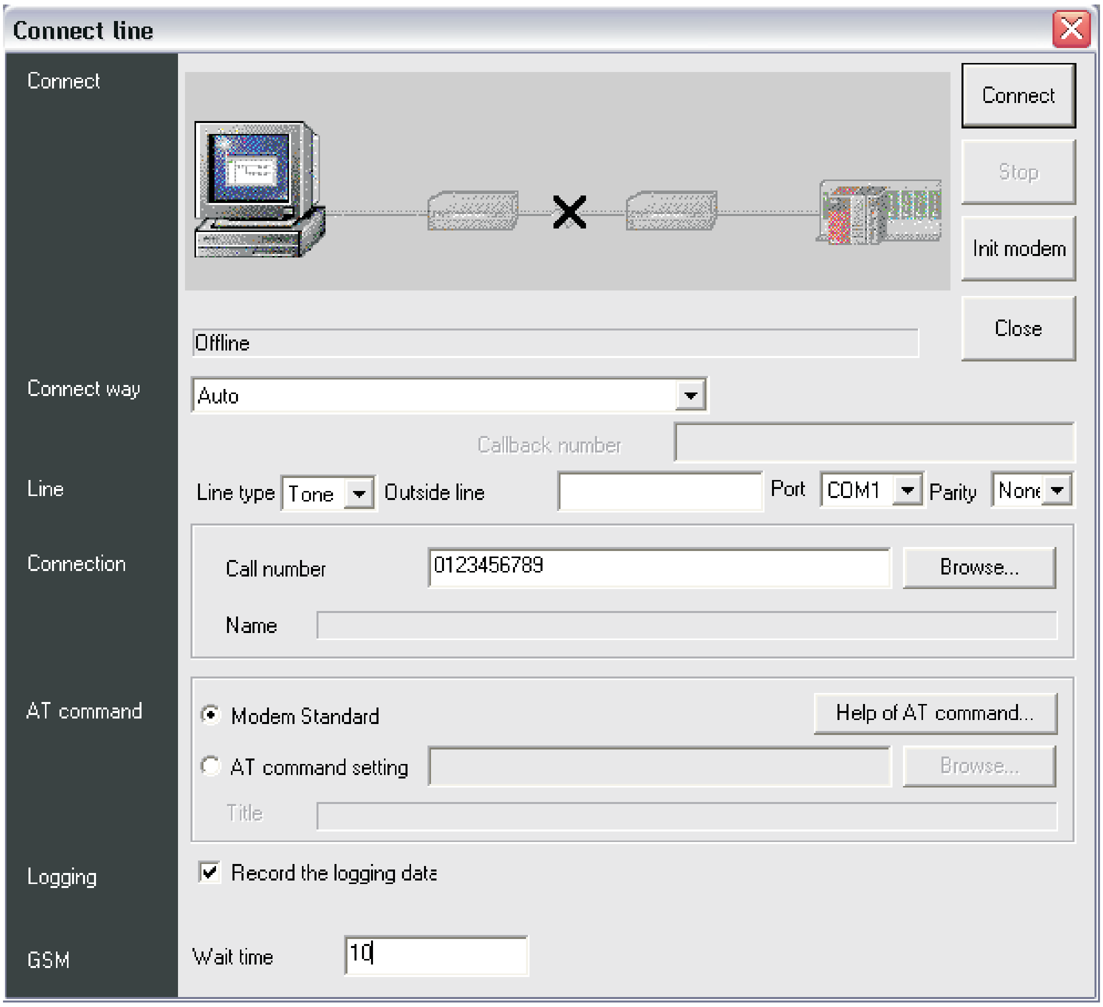
☑ via MAC/MTA transparent mode

□ via GOT(Bus) transparent mode

□ via GOT(direct coupled) transparent mode

Fig. 8-76:
Setting the communication mode

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Offline

Connect way: Auto

Callback number

Line: Line type Tone Outside line Port COM1 Parity None

Connection: Call number 0123456789

Name

AT command: ● Modem Standard

○ AT command setting

Title

Logging: ☑ Record the logging date

GSM: Wait time 10

Fig. 8-77: *Set up the connection*

⑦ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

Environment: GSM -> GSM -> MAC -> QnA-Series Progr. Port

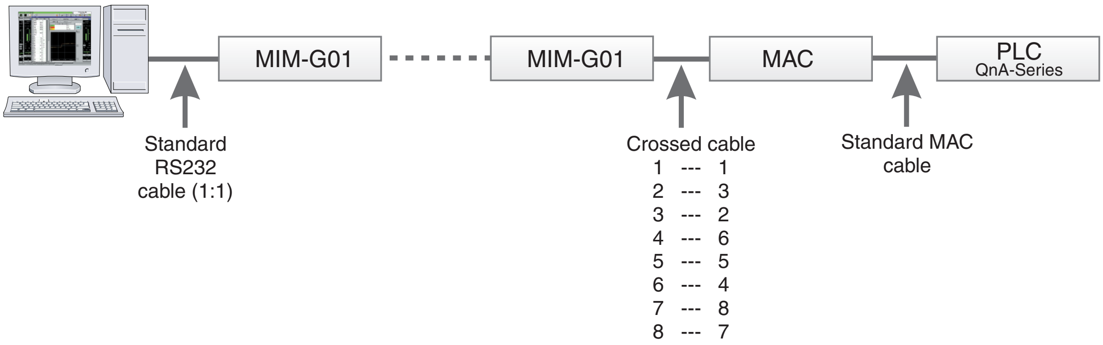


Fig. 8-78: *Environment*

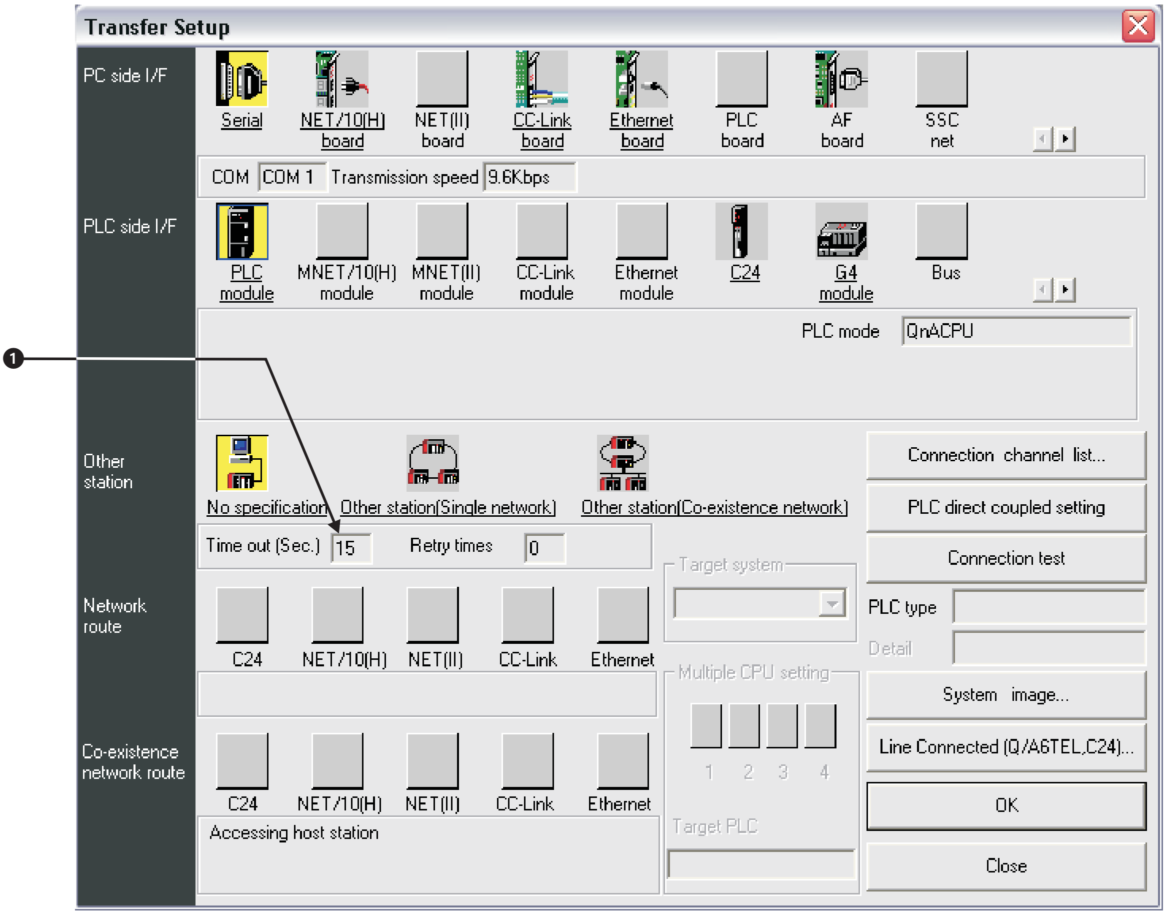
GSM Modem settings:

```
AT          E0
            S0=2
            &S0
            &D0
            +CICB=0
            +IFC=0,0
            +CMEE=1
            +IPR=9600
            &W
            +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

COM: COM 1 Transmission speed: 9.6Kbps

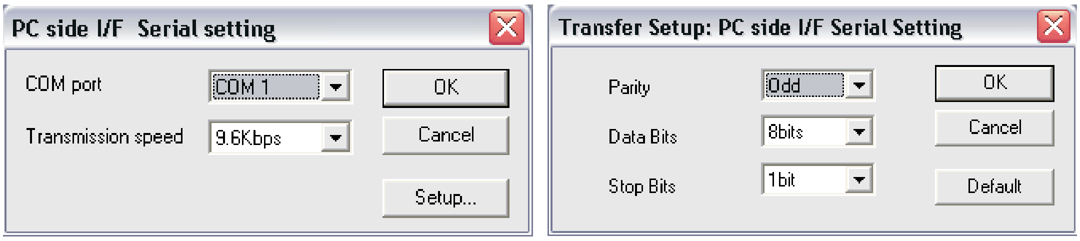
PLC mode: QnACPU

Time out (Sec.): 15 Retry times: 0

Fig. 8-79: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

COM port: COM 1

Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

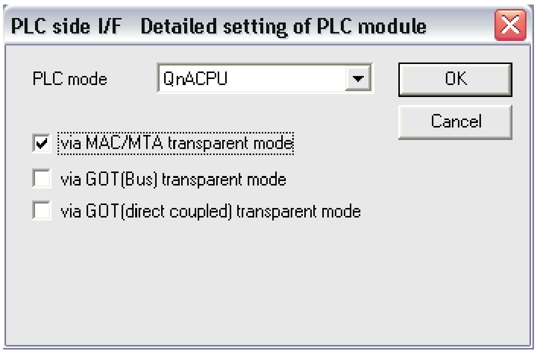
Parity: Odd

Data Bits: 8bits

Stop Bits: 1bit

Fig. 8-80: *Setting the communication details*

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.



PLC side I/F Detailed setting of PLC module

PLC mode: QnACPU

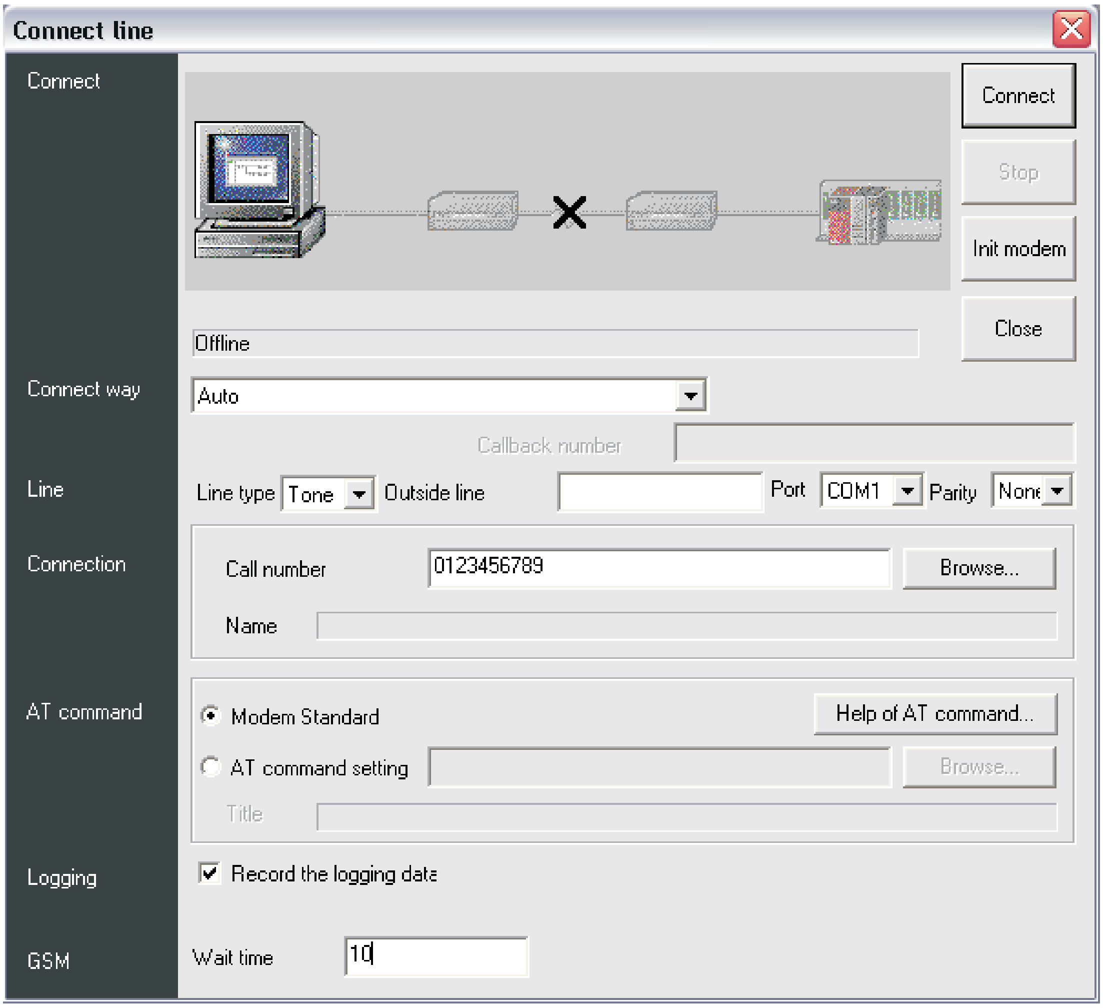
☑ via MAC/MTA transparent mode

□ via GOT(Bus) transparent mode

□ via GOT(direct coupled) transparent mode

Fig. 8-81:
Setting the communication mode

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Connect: Offline

Connect way: Auto

Callback number

Line: Line type: Tone; Outside line: ; Port: COM1; Parity: None

Connection: Call number: 0123456789; Name:

AT command: ● Modem Standard; ○ AT command setting; Title

Logging: ☑ Record the logging data

GSM: Wait time: 10

Fig. 8-82: *Set up the connection*

⑦ If the connection was successful, a message box shows:

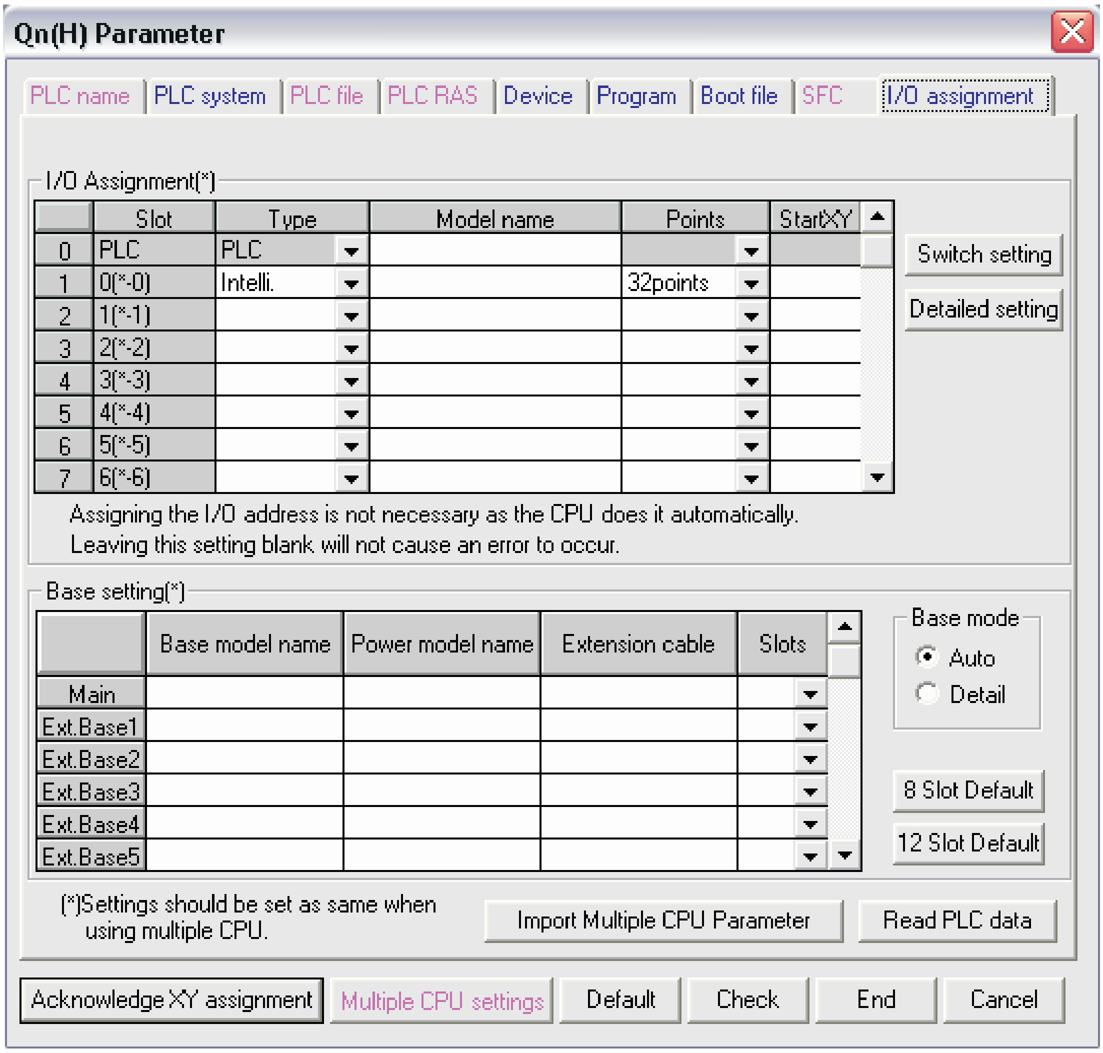
- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

8.5.8 Q-Series – Communication with C24

Settings for QJ71C24-R2

When using the QJ71C24-R2 you first have to do the following settings:

① Open the **Parameter** dialogue and select the tab **I/O assignment**.



I/O Assignment(*)

	Slot	Type	Model name	Points	StartXY
0	PLC	PLC			
1	0(*-0)	Intelli.		32points	
2	1(*-1)				
3	2(*-2)				
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

Base setting(*)

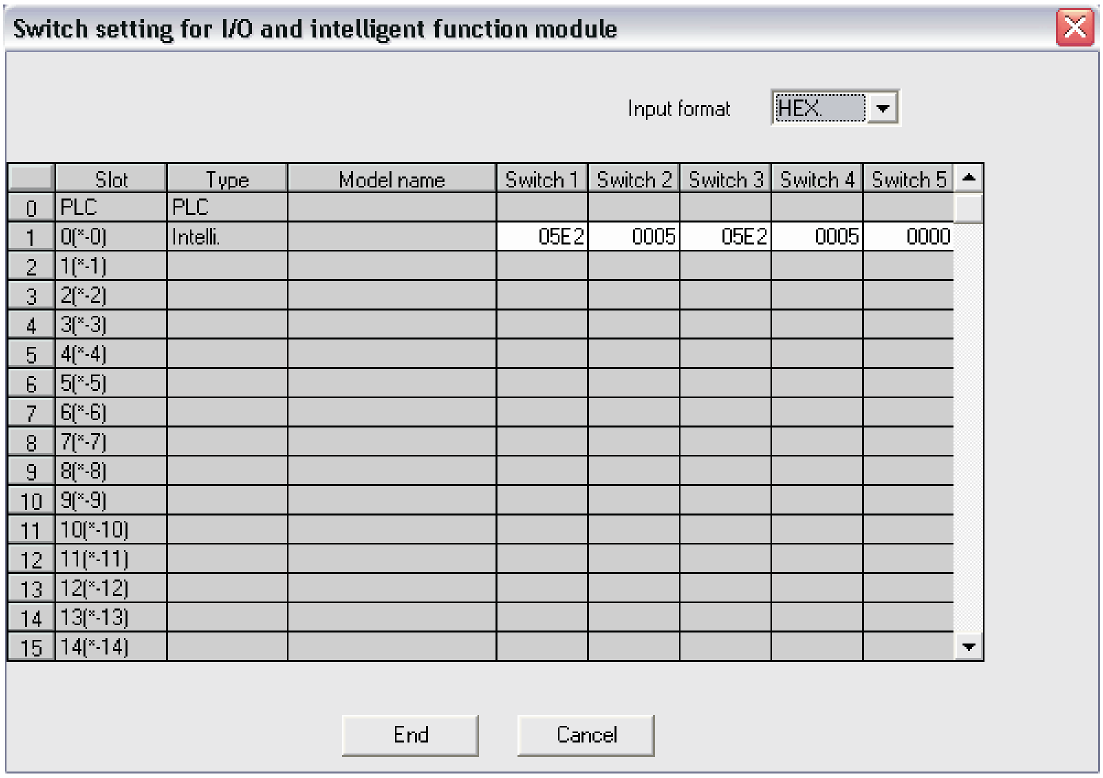
	Base model name	Power model name	Extension cable	Slots
Main				
Ext.Base1				
Ext.Base2				
Ext.Base3				
Ext.Base4				
Ext.Base5				

Base mode
● Auto
○ Detail

(*)Settings should be set as same when
using multiple CPU.

Fig. 8-83: *Tab **I/O assignment** of the **PLC Parameter** dialogue*

② Click on the button **Switch setting** to open the following dialogue.



Switch setting for I/O and intelligent function module

Input format: HEX

	Slot	Type	Model name	Switch 1	Switch 2	Switch 3	Switch 4	Switch 5
0	PLC	PLC						
1	0(*-0)	Intelli.		05E2	0005	05E2	0005	0000
2	1(*-1)							
3	2(*-2)							
4	3(*-3)							
5	4(*-4)							
6	5(*-5)							
7	6(*-6)							
8	7(*-7)							
9	8(*-8)							
10	9(*-9)							
11	10(*-10)							
12	11(*-11)							
13	12(*-12)							
14	13(*-13)							
15	14(*-14)							

Fig. 8-84: *Input the switch settings for the module*

③ Set the following switches:

Switch 1: CH1 Communication and Transmission setting

Switch 2: CH1 Communication protocol setting

Switch 3: CH2 Communication and Transmission setting

Switch 4: CH2 Communication protocol setting

Switch 5: Station number setting

④ Click on the button **End** to save the settings.

Environment: Analogue Modem -> GSM Modem -> QJ71C24-R2

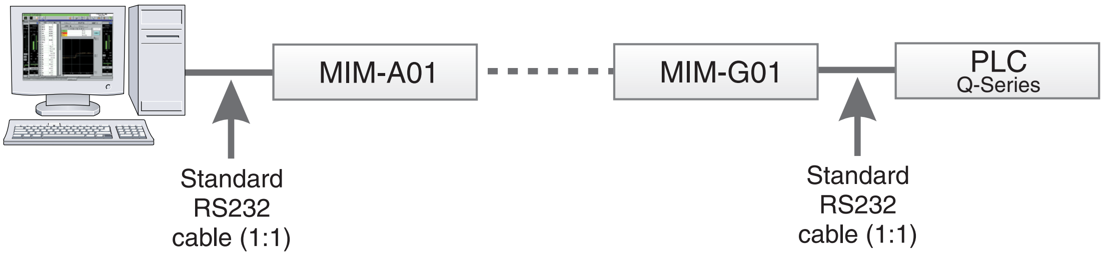


Fig. 8-85: *Environment*

Analogue Modem settings:

AT	E0
	&S0
	&D0
	&K0

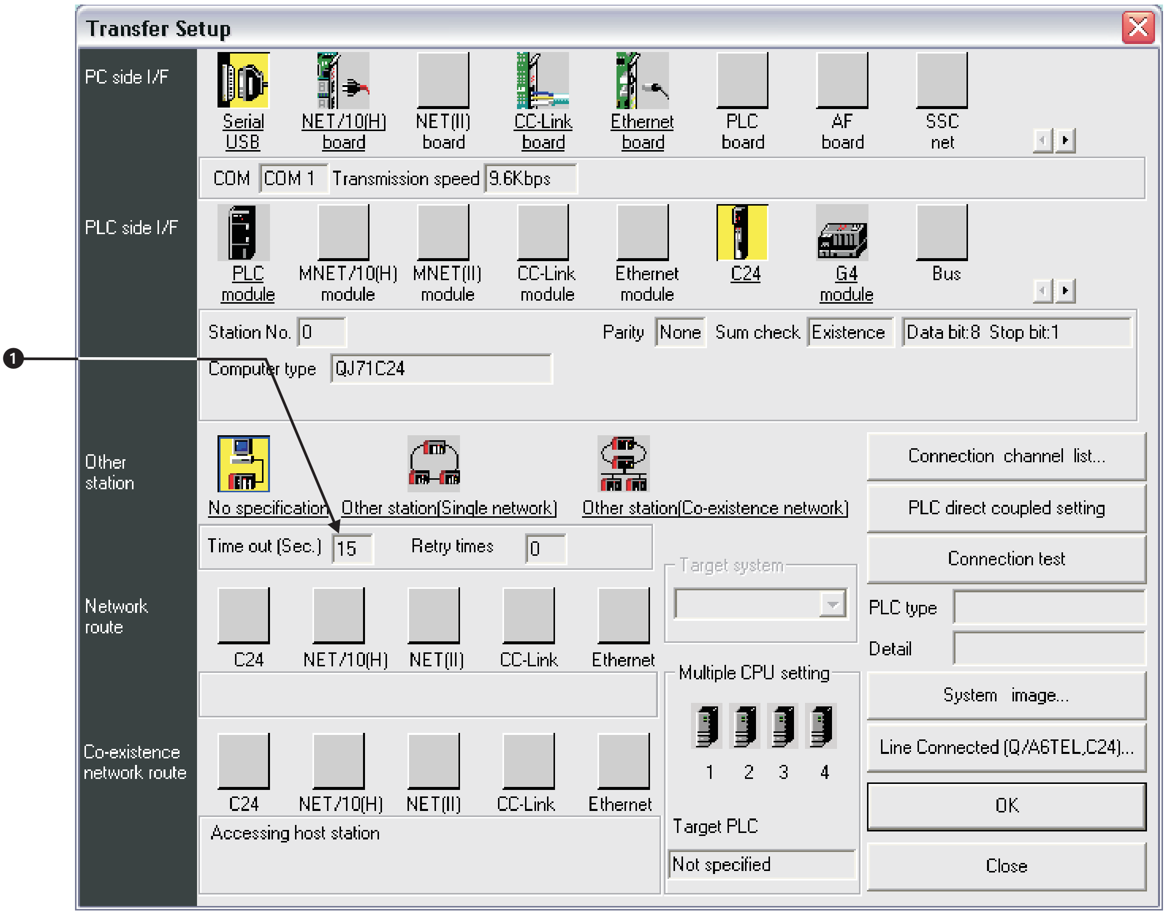
GSM Modem settings:

AT	E0
	S0=2
	&S0
	&D0
	+CICB=0
	+IFC=0,0
	+CMEE=1
	+IPR=9600
	+CPIN=____ (PIN No. of SIM card)

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

PC side I/F: COM COM 1 Transmission speed 9.6Kbps

PLC side I/F: Station No. 0 Parity None Sum check Existence Data bit:8 Stop bit:1

Computer type QJ71C24

Other station: Time out (Sec.) 15 Retry times 0

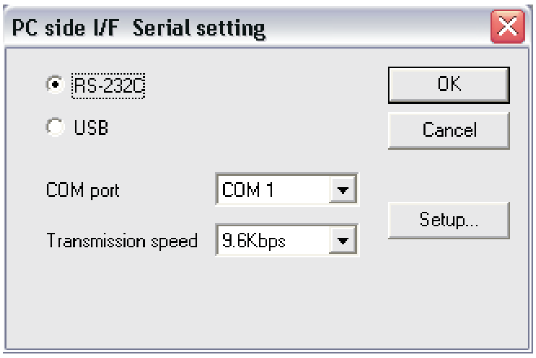
Target PLC: Not specified

Accessing host station

Fig. 8-86: *Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.

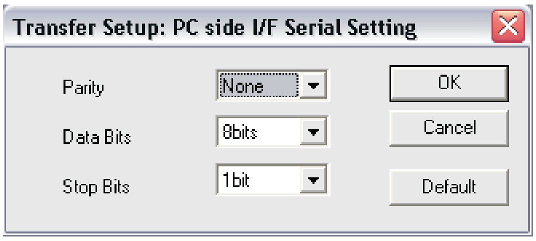


PC side I/F Serial setting

● RS-232C
○ USB

COM port: COM 1

Transmission speed: 9.6Kbps



Transfer Setup: PC side I/F Serial Setting

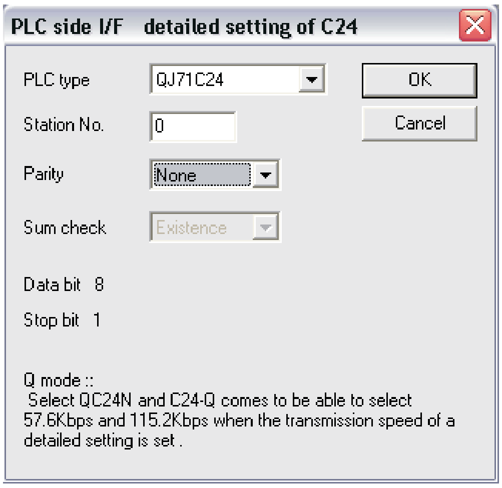
Parity: None

Data Bits: 8bits

Stop Bits: 1bit

Fig. 8-87: *Setting the communication details*

⑤ Double click on **PLC side I/F - C24** to open the dialogue box **PLC side I/F Detailed setting of C24** and set the details.



PLC side I/F detailed setting of C24

PLC type QJ71C24

Station No. 0

Parity None

Sum check Existence

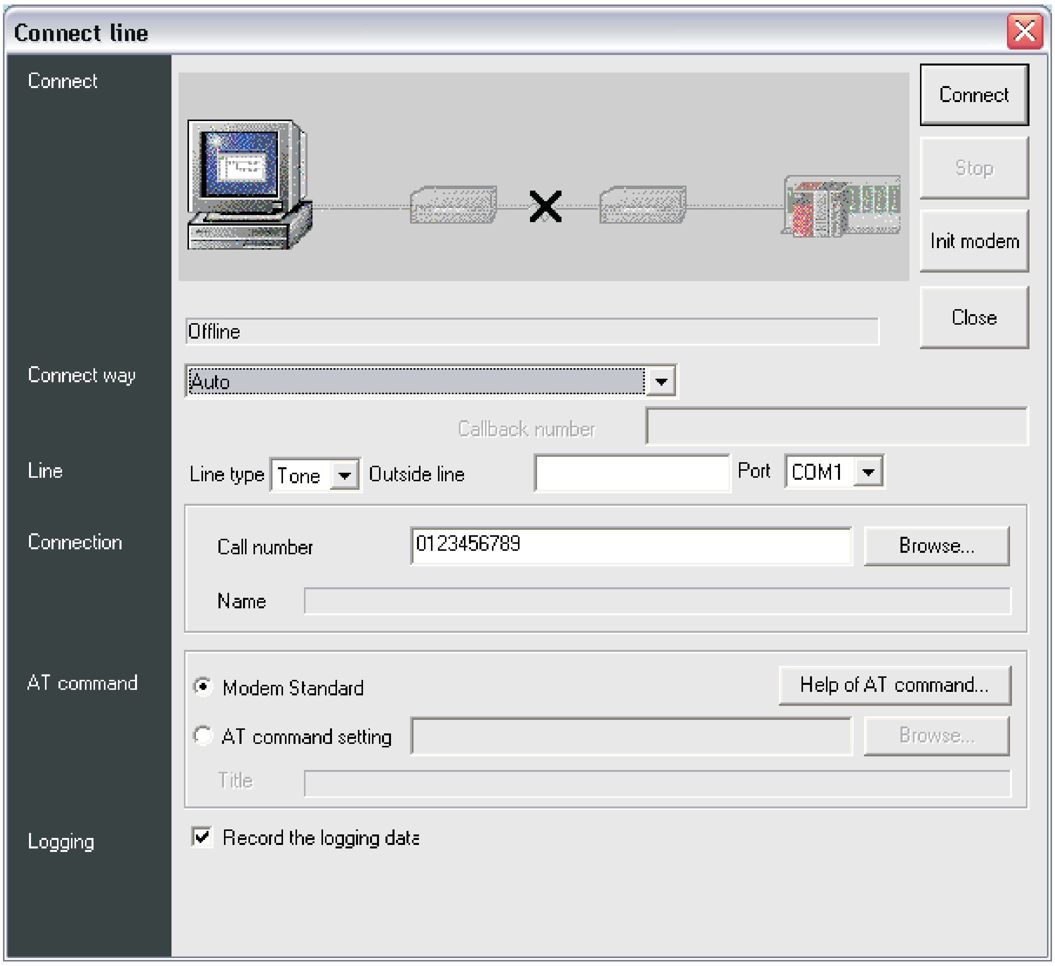
Data bit 8

Stop bit 1

Q mode ::
Select QC24N and C24-Q comes to be able to select 57.6Kbps and 115.2Kbps when the transmission speed of a detailed setting is set .

Fig. 8-88:
Detailed setting for the C24

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Offline

Connect way Auto

Callback number

Line: Line type Tone Outside line Port COM1

Connection: Call number 0123456789 Name

AT command: ● Modem Standard ○ AT command setting Title

Logging: ☑ Record the logging date

Fig. 8-89: *Set up the connection*

⑦ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Modem -> QJ71C24-R2

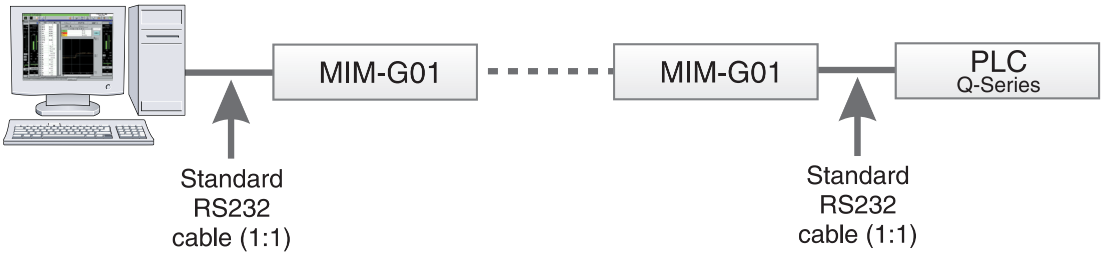


Fig. 8-90: *Environment*

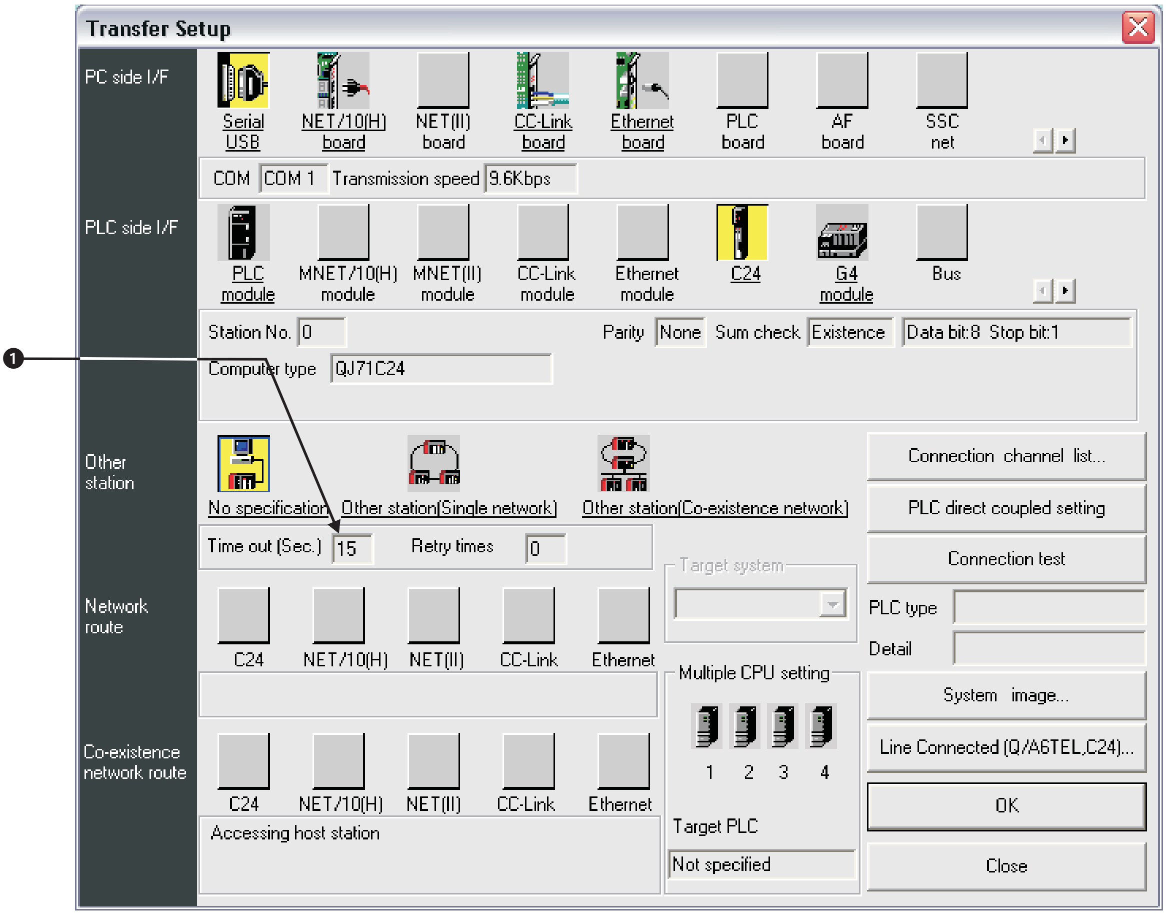
GSM Modem settings:

```
AT      E0
        S0=2
        &S0
        &D0
        +CICB=0
        +IFC=0,0
        +CMEE=1
        +IPR=9600
        +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



COM COM 1 Transmission speed 9.6Kbps

Station No. 0 Parity None Sum check Existence Data bit:8 Stop bit:1

Computer type QJ71C24

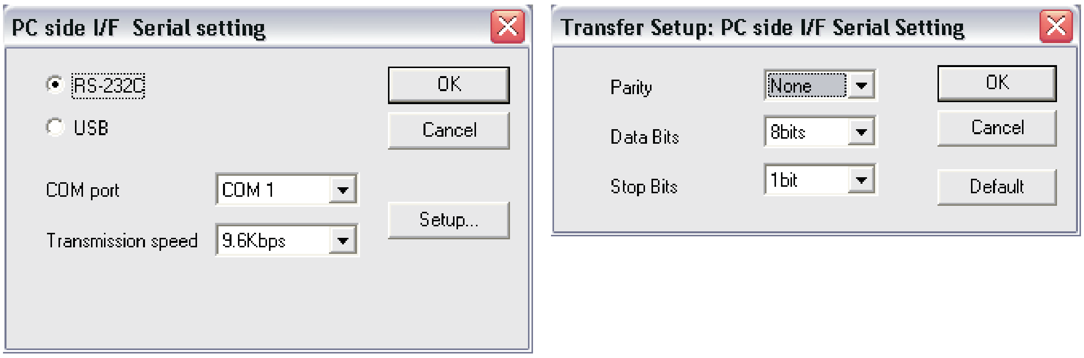
Time out (Sec.) 15 Retry times 0

Target PLC Not specified

*Fig. 8-91: Setting in the menu **Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.


PC side I/F Serial setting
OK
Cancel
Setup...
Transfer Setup: PC side I/F Serial Setting
OK
Cancel
Default

● RS-232C
○ USB

COM port COM 1

Transmission speed 9.6Kbps

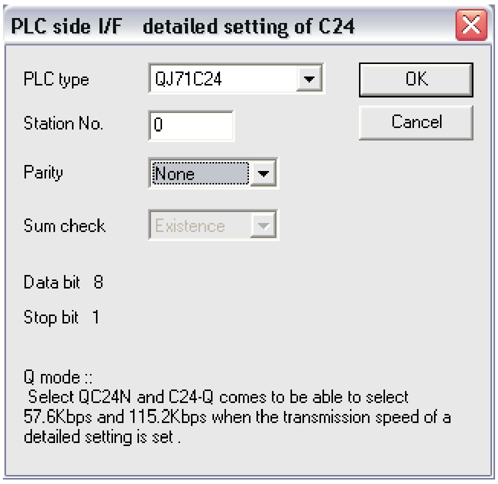
Parity None

Data Bits 8bits

Stop Bits 1bit

Fig. 8-92: Setting the communication details

⑤ Double click on **PLC side I/F - C24** to open the dialogue box **PLC side I/F Detailed setting of C24** and set the details.



PLC side I/F detailed setting of C24

PLC type: QJ71C24

Station No.: 0

Parity: None

Sum check: Existence

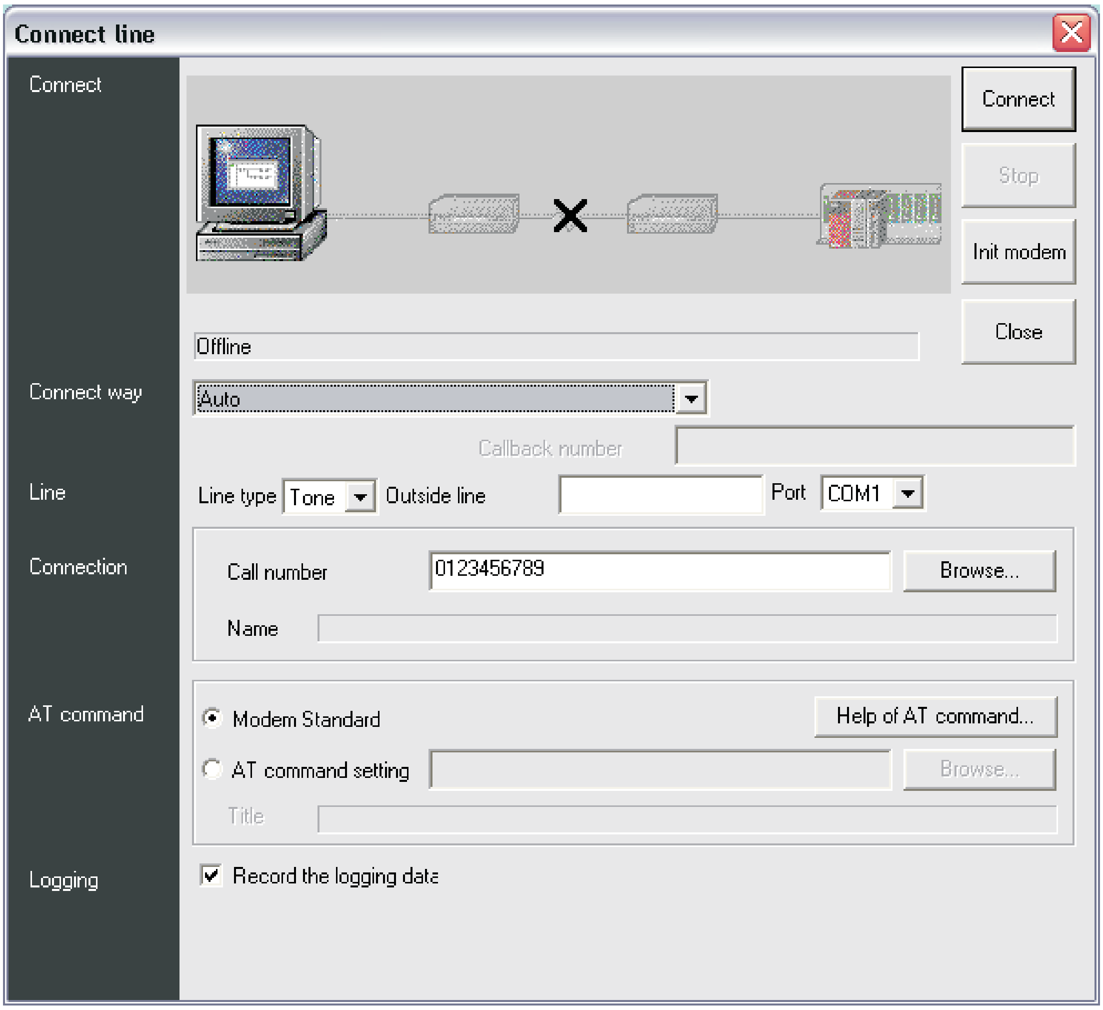
Data bit 8

Stop bit 1

Q mode ::
Select QC24N and C24-Q comes to be able to select 57.6Kbps and 115.2Kbps when the transmission speed of a detailed setting is set .

Fig. 8-93:
Detailed setting for the C24

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Offline

Connect way: Auto

Callback number:

Line type: Tone Outside line: Port: COM1

Call number: 0123456789

Name:

AT command: ● Modem Standard ○ AT command setting

Title:

Logging: ☑ Record the logging data

Fig. 8-94: *Set up the connection*

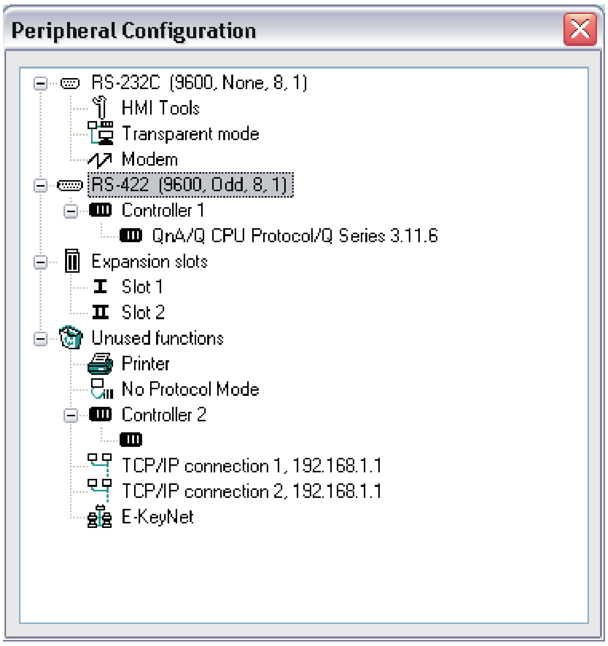
⑦ If the connection was successful, a message box shows:

– Connect completed

– Destination – 0123456789 ← Call number of your remote modem

8.5.9 Q-Series – Communication via MAC (Transparent Mode)

Settings for MAC Terminal:



Peripheral Configuration

```
RS-232C (9600, None, 8, 1)
    HMI Tools
    Transparent mode
    Modem
RS-422 (9600, Odd, 8, 1)
    Controller 1
        QnA/Q CPU Protocol/Q Series 3.11.6
Expansion slots
    Slot 1
    Slot 2
Unused functions
    Printer
    No Protocol Mode
    Controller 2
    TCP/IP connection 1, 192.168.1.1
    TCP/IP connection 2, 192.168.1.1
    E-KeyNet
```

Fig. 8-95:
Setting the peripheral configuration in the E-Designer

Environment: Analogue Modem -> GSM Modem -> MAC -> Q-Series Progr. Port

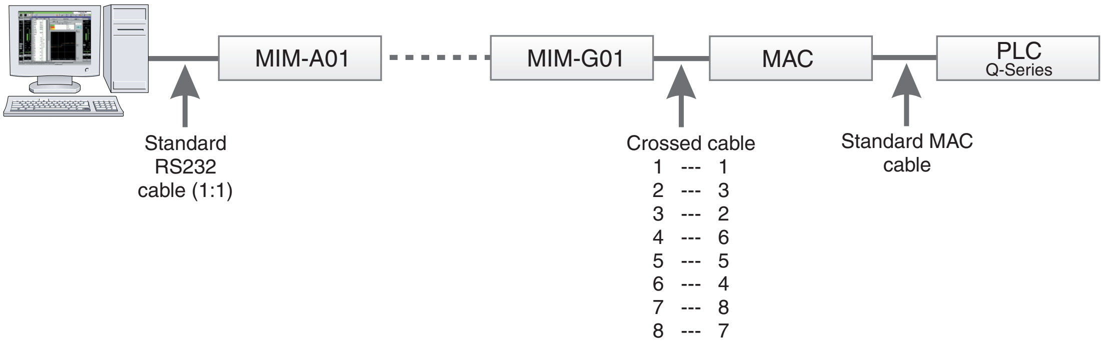


Fig. 8-96: *Environment*

Analogue Modem settings:

```
AT          E0
            &S0
            &D0
            &K0
```

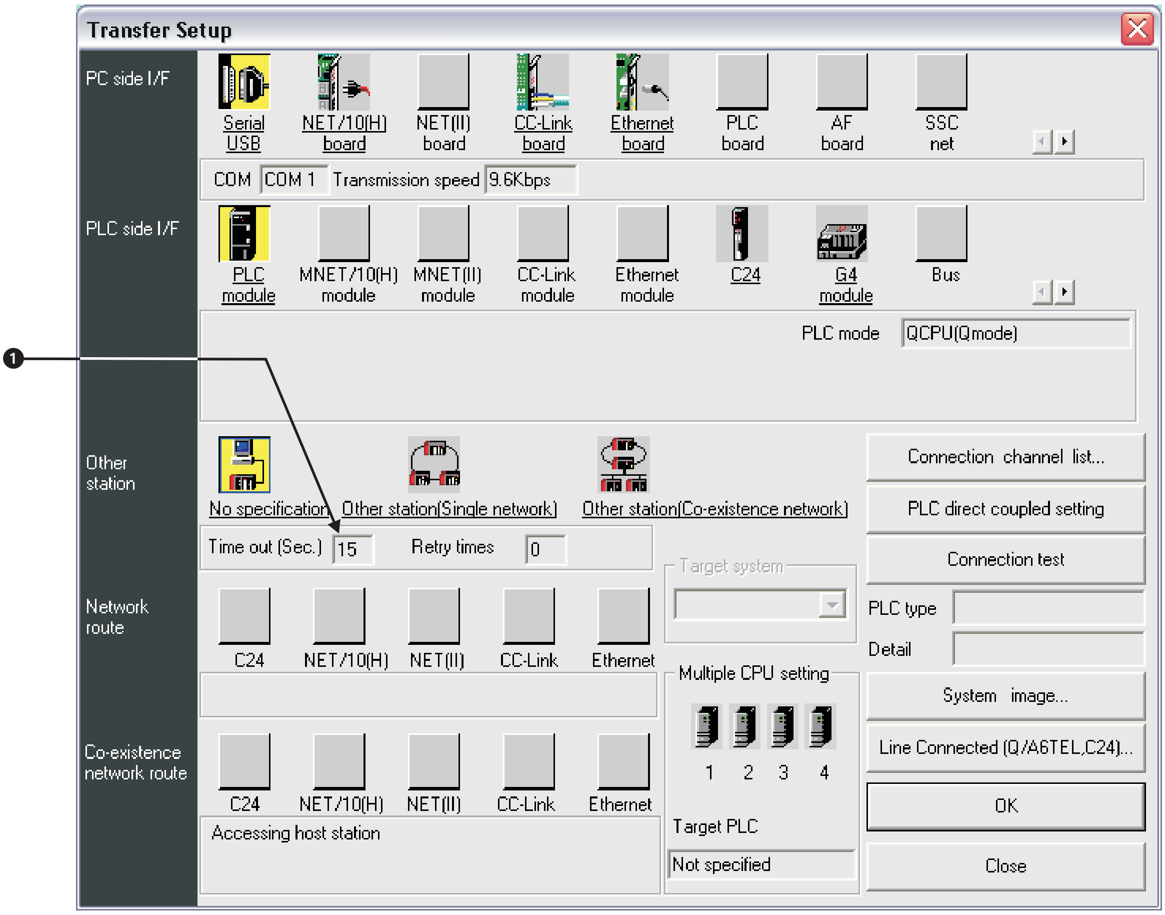
GSM Modem settings:

```
AT          E0
            S0=2
            &S0
            &D0
            +CICB=0
            +IFC=0,0
            +CMEE=1
            +IPR=9600
            &W
            +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

COM COM 1 Transmission speed 9.6Kbps

PLC mode QCPU(Qmode)

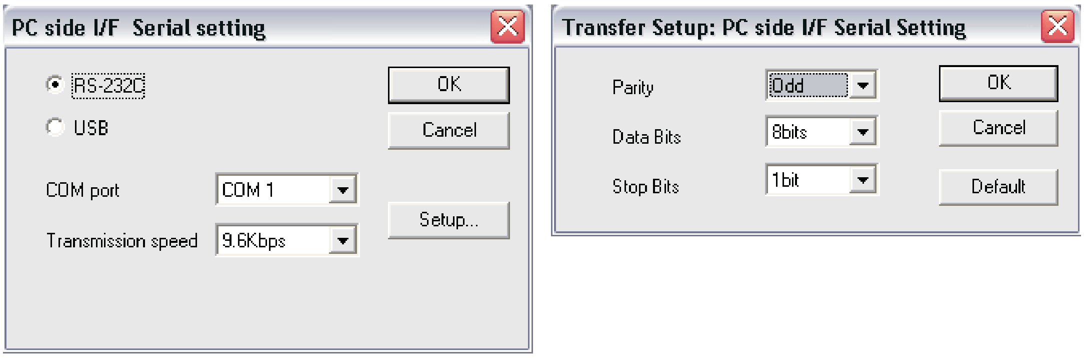
Time out (Sec.) 15 Retry times 0

Target PLC: Not specified

Fig. 8-97: *Setting in the menu* ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

◉ RS-232C
○ USB

COM port COM 1

Transmission speed 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

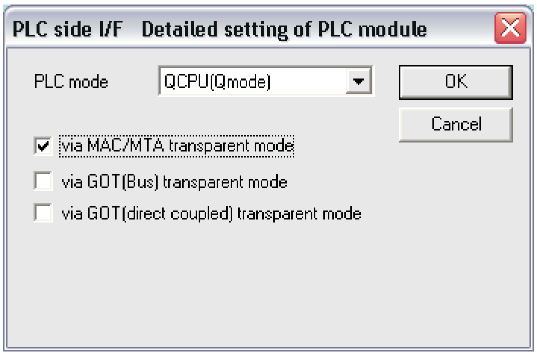
Parity Odd

Data Bits 8bits

Stop Bits 1bit

Fig. 8-98: *Setting the communication details*

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.



PLC side I/F Detailed setting of PLC module

PLC mode: QCPU(Qmode)

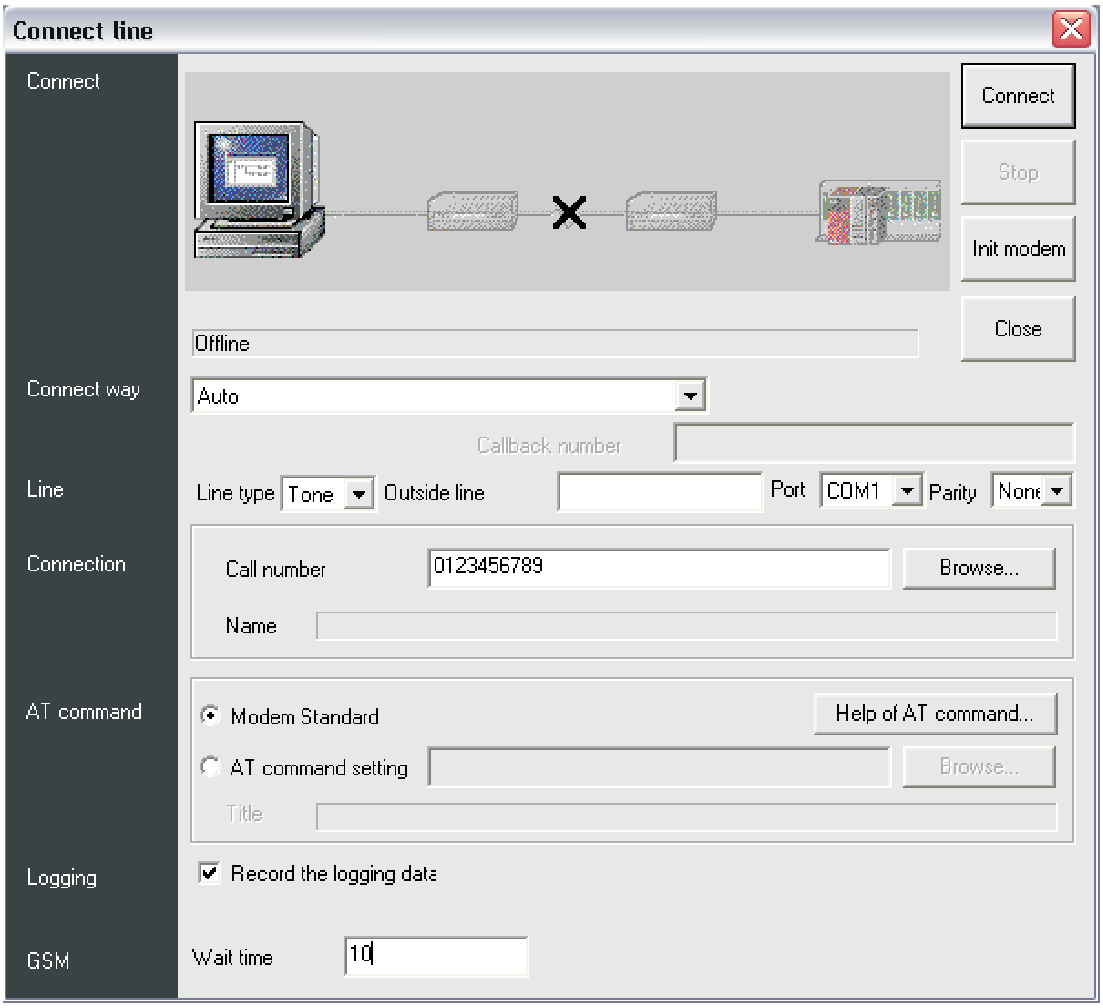
☑ via MAC/MTA transparent mode

□ via GOT(Bus) transparent mode

□ via GOT(direct coupled) transparent mode

Fig. 8-99:
Setting the communication mode

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Connect: Offline

Connect way: Auto

Callback number:

Line: Line type Tone, Outside line, Port COM1, Parity None

Connection: Call number 0123456789, Name

AT command: ● Modem Standard ○ AT command setting, Title

Logging: ☑ Record the logging date

GSM: Wait time 10

Fig. 8-100: *Set up the connection*

⑦ If the connection was successful, a message box shows:

– Connect completed

– Destination – 0123456789 ← Call number of your remote modem

Environment: GSM Modem -> GSM Modem -> MAC -> Q-Series Progr. Port

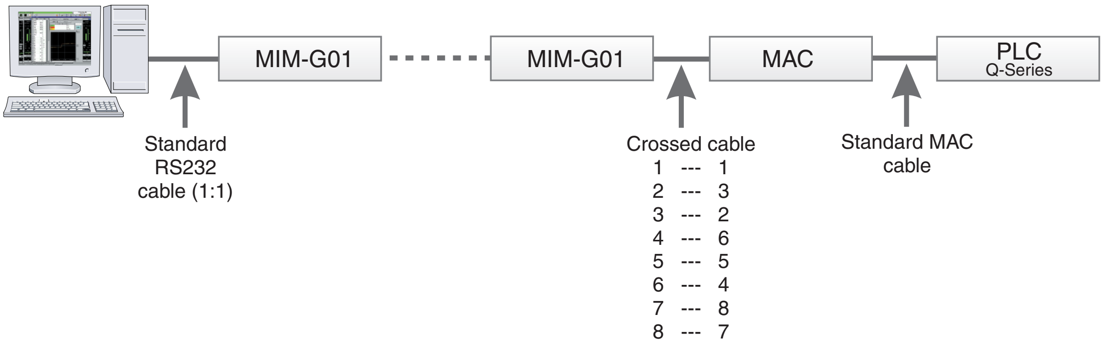


Fig. 8-101: *Environment*

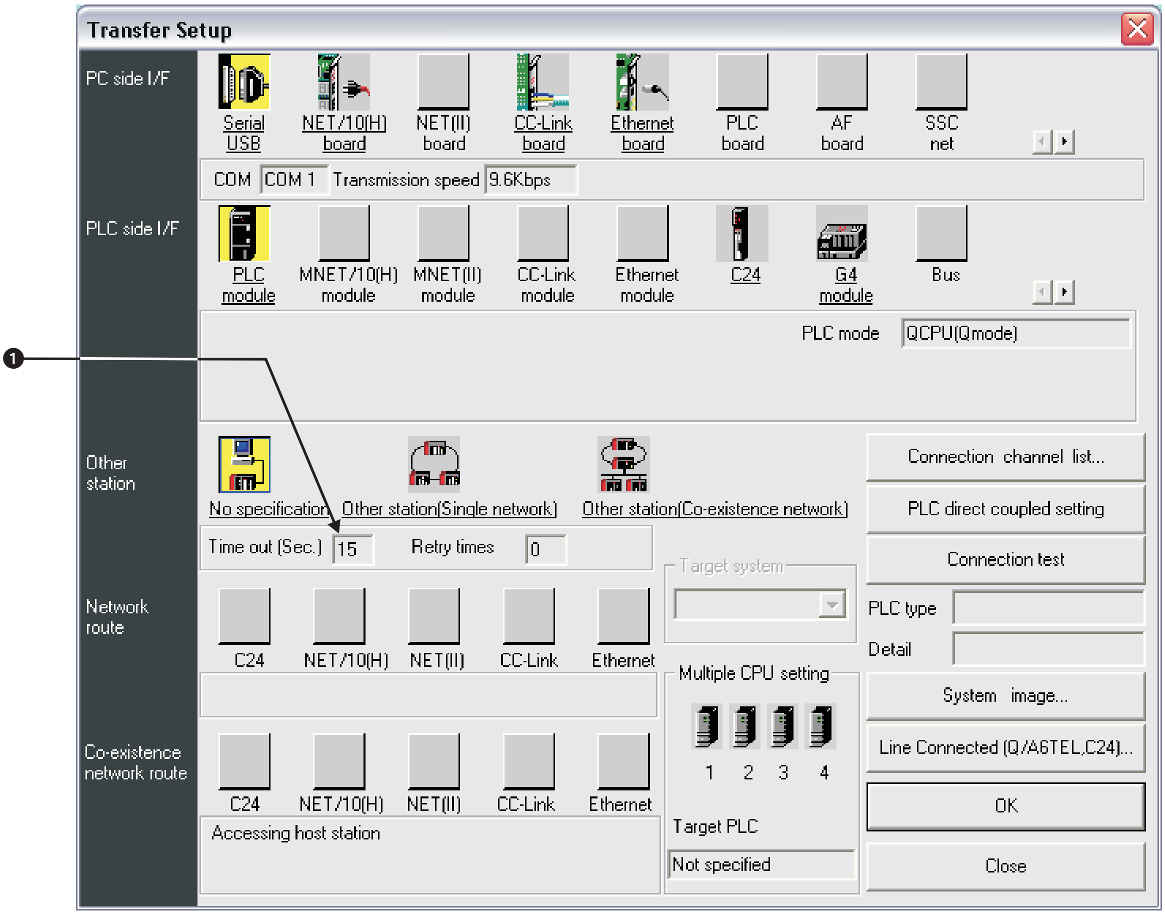
GSM Modem settings:

```
AT        E0
          S0=2
          &S0
          &D0
          +CICB=0
          +IFC=0,0
          +CMEE=1
          +IPR=9600
          &W
          +CPIN=____  (PIN No. of SIM card)
```

GX IEC Developer settings:

① Select **Transfer Setup** in the **Online** menu, then select **Ports** from the submenu displayed.

② Set the value for the option **Time out** (❶ in the following dialogue window) to ≥ 15 sec.



Transfer Setup

PC side I/F: Serial USB — COM: COM 1, Transmission speed: 9.6Kbps

PLC side I/F: PLC module — PLC mode: QCPU(Qmode)

Other station: No specification — Time out (Sec.): 15, Retry times: 0

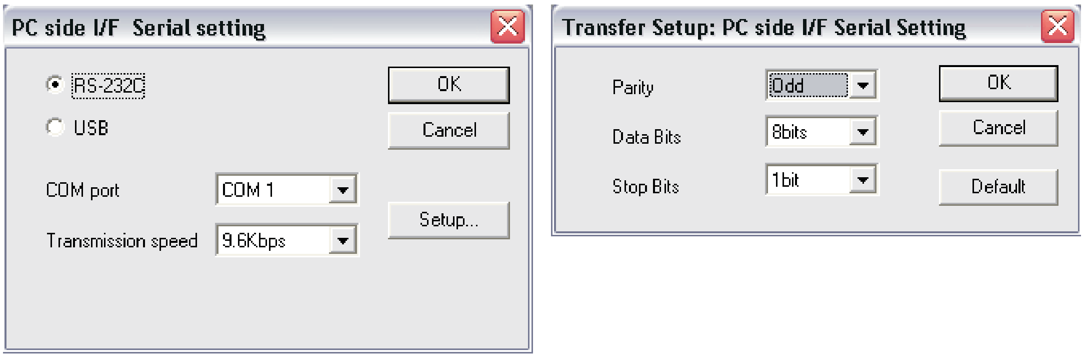
Co-existence network route: Accessing host station

Target PLC: Not specified

Fig. 8-102: Setting in the menu ***Transfer Setup***

③ Double click on **PC side I/F - Serial** to open the dialogue box **PC side I/F Serial setting** and set the connection details.

④ Select Setup to open the dialogue **Transfer Setup: PC side I/F Serial Setting** to set up further communication details.



PC side I/F Serial setting

● RS-232C
○ USB

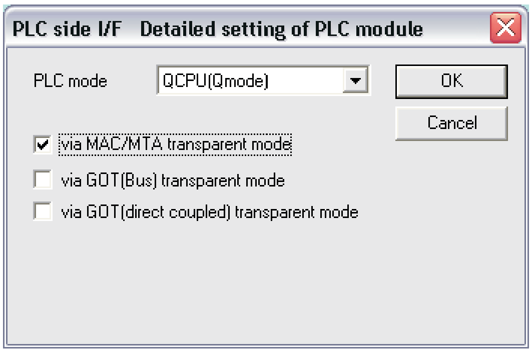
COM port: COM 1
Transmission speed: 9.6Kbps

Transfer Setup: PC side I/F Serial Setting

Parity: Odd
Data Bits: 8bits
Stop Bits: 1bit

Fig. 8-103: Setting the communication details

⑤ Double click on **PLC side I/F - PLC Modul** to open the dialogue box **PLC side I/F Detailed setting of PLC module** and set the communication mode.



PLC side I/F Detailed setting of PLC module

PLC mode: QCPU(Qmode)

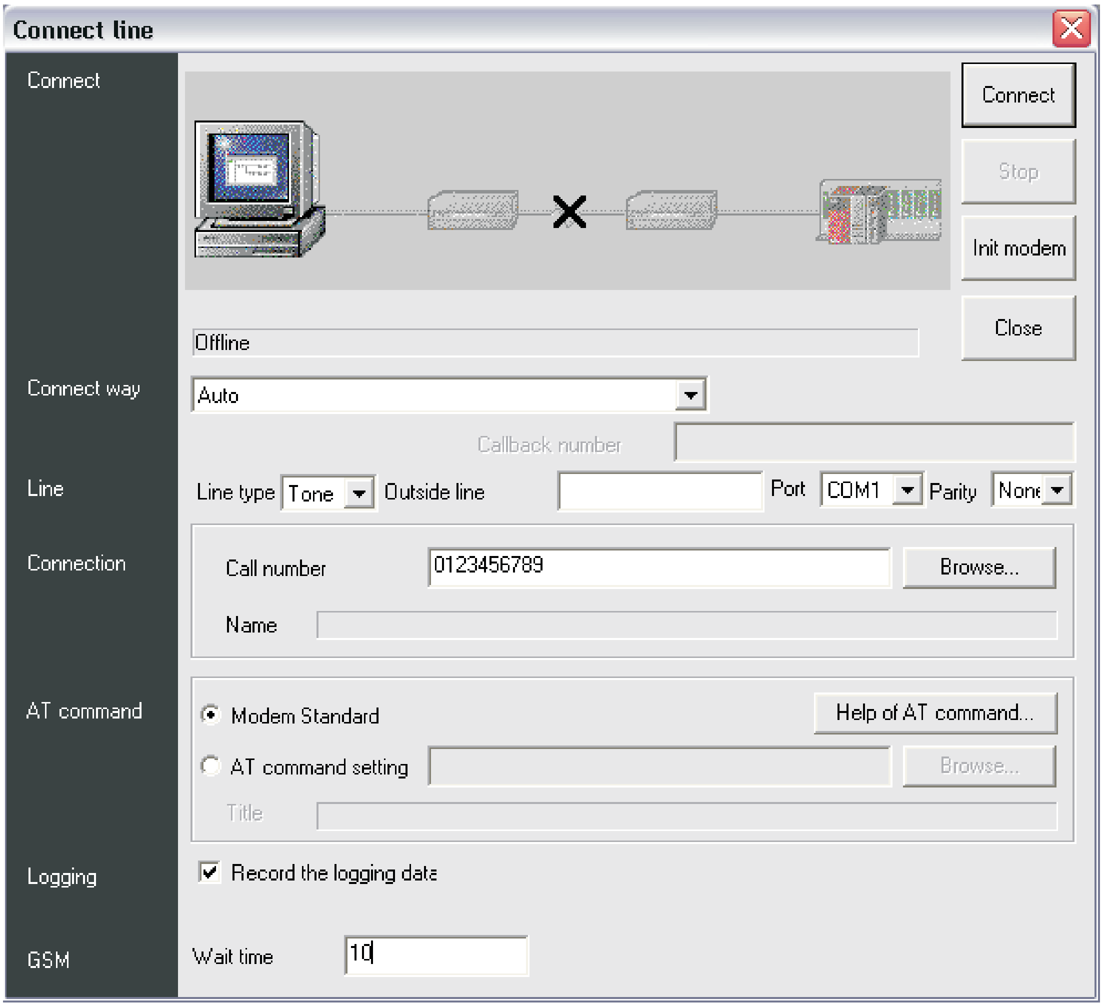
☑ via MAC/MTA transparent mode

□ via GOT(Bus) transparent mode

□ via GOT(direct coupled) transparent mode

Fig. 8-104:
Setting the communication mode

⑥ Select **Line Connected**, enter the **Call number** of the remote modem and click **Connect**.



Connect line

Connect: Offline

Connect way: Auto

Callback number:

Line: Line type Tone, Outside line, Port COM1, Parity None

Connection: Call number 0123456789; Name

AT command: ● Modem Standard; ○ AT command setting; Title

Logging: ☑ Record the logging dat

GSM: Wait time 10

Fig. 8-105: *Set up the connection*

⑦ If the connection was successful, a message box shows:

- Connect completed
- Destination – 0123456789 ← Call number of your remote modem

8.5.10 Special Remarks on using GSM modems

GSM modem synchronisation

Some modems, mainly GSM modems, do not work with the standard communication settings. In such cases the (GSM) modem often needs some time between retries while communicating. This time to wait between retries we call modem synchronisation time, which can be enabled and set in this functionality.

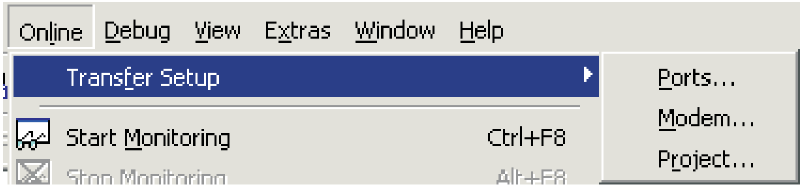
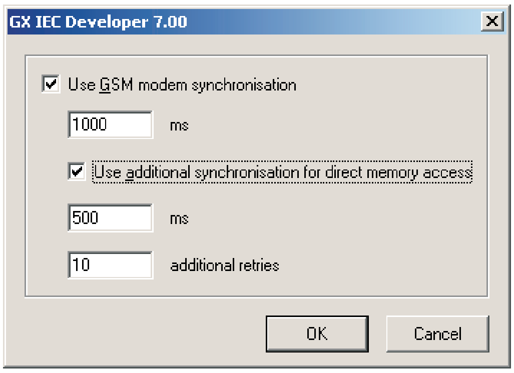


Fig. 8-106: *Menu item under **Online -> Transfer Setup***

The menu item is available if a project is open and either MAC transparent mode with CPU port or C24 is selected in the **Transfer Setup -> Ports**.



GX IEC Developer 7.00

☑ Use GSM modem synchronisation

1000 ms

☑ Use additional synchronisation for direct memory access

500 ms

10 additional retries

Fig. 8-107:
Dialogue

Use GSM modem synchronization: If checked, the synchronization is used. Enter a value in the range of 1 to 60000 ms for synchronization.

Timeout setting for communication initialization

When initializing the connection with the PLC, the timeout setting entered in the **Transfer Setup** does not apply to the communication. The timeout value set in the **Transfer Setup** is used only after the communication has been initialized.

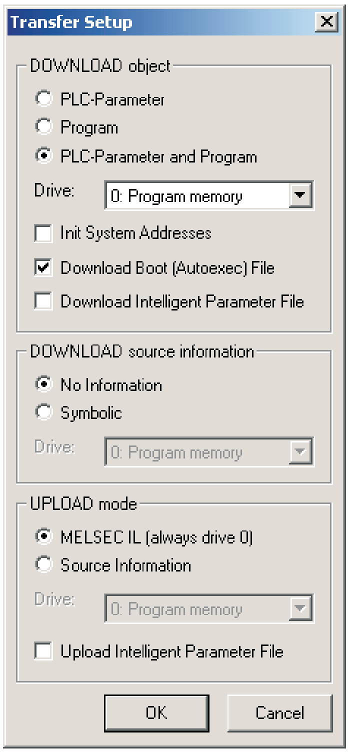
The pre-defined timeout setting is not suitable for GSM communication under specific circumstances, which causes communication failures.

To avoid communication failures when using GSM or other connection types with occasional high delays, it is possible to increase the timeout value used at communication initialization. When you set a timeout value of 15 seconds or more in the **Transfer Setup**, this value will be used also when initializing the communication with the PLC. If the timeout value is lower than 15 seconds, the original pre-defined value will be used for communication initialization.

- If the timeout setting is less than 15 seconds, then the performance is not affected.
- If the timeout setting is 15 seconds or greater, then:
  - If the connected PLC can be accessed, the performance of the communication initialization is not affected.
  - If there is no connected PLC, or the connected PLC cannot be accessed, then at least the time set for timeout must elapse until the communication initialization attempt can finish.

8.6 Transfer Setup - Project

You can specify precisely which objects and source information in your project you want to download.



Transfer Setup

DOWNLOAD object
- ○ PLC-Parameter
- ○ Program
- ● PLC-Parameter and Program

Drive: 0: Program memory
- □ Init System Addresses
- ☑ Download Boot (Autoexec) File
- □ Download Intelligent Parameter File

DOWNLOAD source information
- ● No Information
- ○ Symbolic

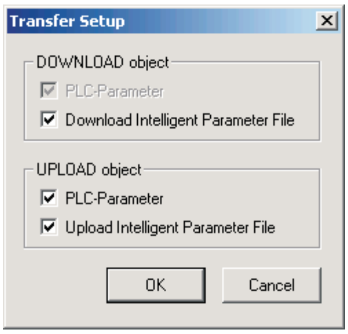
Drive: 0: Program memory

UPLOAD mode
- ● MELSEC IL (always drive 0)
- ○ Source Information

Drive: 0: Program memory
- □ Upload Intelligent Parameter File

Fig. 8-108:
Project transfer setup

NOTE Since the Q Remote I/O module has a Flash ROM drive only, the download function transfers only the parameter and (if selected) the intelligent parameter to drive 4 of the PLC.



Transfer Setup

DOWNLOAD object
- ☑ PLC-Parameter
- ☑ Download Intelligent Parameter File

UPLOAD object
- ☑ PLC-Parameter
- ☑ Upload Intelligent Parameter File

Fig. 8-109:
Project transfer setup for a Q Remote I/O module

How to set the transfer settings

① In the **Online** menu select **Transfer Setup**, then select **Project**.

② Select the desired options in the **Download Object** and **Source Information** fields of the dialogue box.

PLC Parameter : Downloads only the PLC configuration data to the PLC CPU.

Program: Downloads only the compiled program code to the PLC.

PLC Parameter and Program: Downloads both the configuration data and the compiled program code to the PLC.

Download Intelligent Parameter File (only Q Remote I/O): Downloads the intelligent parameter file to the Remote I/O module.

Drive: Select the PLC CPU drive for the transfer.

Init System Addresses: Initialises all system variables during download. This is done by a direct memory write.

Download Boot (Autoexec) File: Controls the download of the boot file to the PLC. However, if this check box is unchecked, the boot file (if it already exists on the PLC) will be deleted from the PLC during download. For Q00(J)/Q01 PLCs this check box is disabled. To download the boot file to these PLCs, the corresponding check box **Do boot from standard ROM** in the PLC Parameters option **Boot File** has to be checked.

Download Intelligent Parameter File: By checking this option the Intelligent Parameter File (IPARAM) for Q projects will be downloaded to the PLC.

Symbolic: Downloads the project together with all information, including the POUs not defined in tasks. If a project, which includes the setting *structure* (used in lower MM+ versions), is downloaded the setting will be converted automatically to *symbolic*. This means that the information in the project database will not be changed, but when the information is queried, *structure* will be handled as *symbolic*.
The symbolic download may be encrypted with a password. In the password only alphanumeric characters (a..z, A..Z, 0..9) are accepted.

No Information: Downloads only the binary program code to the CPU.

Drive: Select the PLC CPU drive for the transfer.

MELSEC IL (always drive 0): Reads the binary code from the PLC CPU and creates a POU in MELSEC IL language with the standard name PRG_MAIN and a task with the standard name MELSEC_MAIN.

Source information: Reads all the graphical and structural information from the PLC CPU in addition to the binary code.

Drive: Select the PLC CPU drive for the transfer.

Upload Intelligent Parameter File: By checking this option the Intelligent Parameter File (IPARAM) for Q projects will be uploaded from the PLC.

③ Confirm your selections with **OK**.

WARNING:
The first time you download a new project it is important to ensure that the option PLC Parameter and Program is selected.

Available drives

The following table lists all drives and their respective availability for the following functions in the menu **Project - Transfer:**
Upload from PLC, Download to PLC, Copy Program Memory to Flash ROM.

Drive		Q	Q00(J)/Q01	QnA	Q Remote I/O
0	Type	Program memory	Program memory	Internal RAM	—
	Upload	●	●	●	—
	Download	●	●	●	—
1	Type	Memory card RAM	not available	IC card A RAM	—
	Upload	●		●	—
	Download	●		●	—
	Copy destination	—		—	—
2	Type	Memory card ROM	not available	IC card A ROM	—
	Upload	●		●	—
	Download	●		●	—
	Copy destination	● (flash ROM card only)		—	—
3	Type	Standard RAM	Standard RAM (used for automatically defined file registers)	IC card B RAM	—
	Upload	—	—	●	—
	Download	—	—	●	—
	Copy destination	—	—	—	—
4	Type	Standard ROM	Standard ROM	IC card B ROM	Flash ROM
	Upload	●	●	●	●
	Download	●	— [1]	●	●
	Copy destination	●	●	—	●

Tab. 8-3: *Available PLC CPU drives*

● Supported

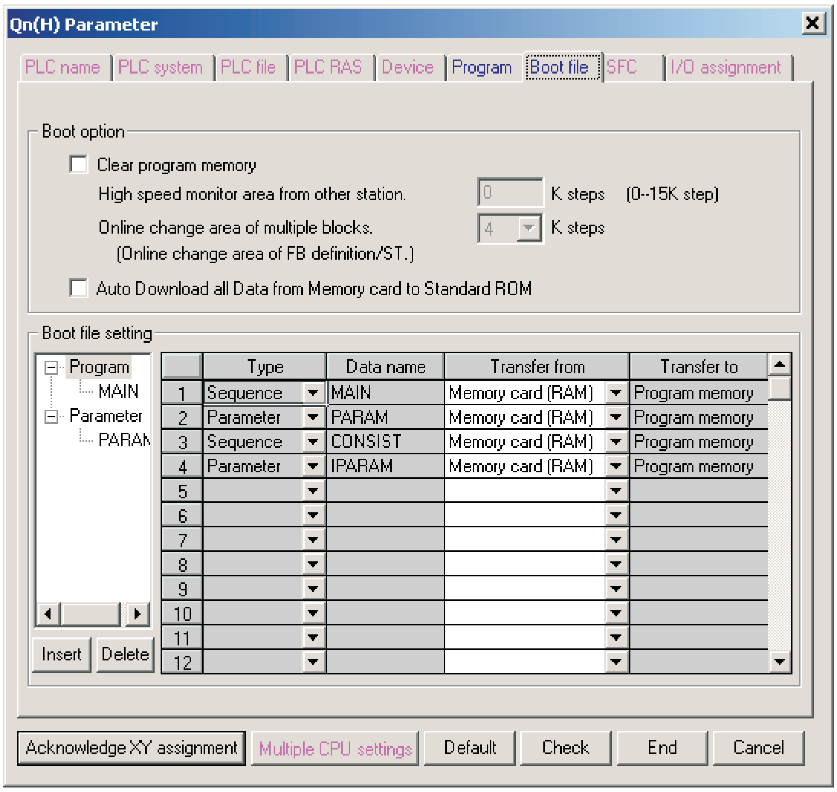
— Not supported

[1] Direct download of the project to a flash ROM is not supported for the Q00(J)/Q01 CPUs. A project can be transferred to drive 4 by downloading it to the program memory and then copying the program memory to the flash ROM.

8.6.1 Downloading the Intelligent Parameter file for Q projects

The download of the Intelligent Parameter (IPARAM) file for a Q project is handled by the dialogues **PLC Parameter** → **Boot Settings** and **Project Transfer Setup**. The file IPARAM is part of the GX IEC Developer project. It is located in the subfolder **Resource** of the project folder.

PLC Parameter



Boot option

☐ Clear program memory

High speed monitor area from other station. 0 K steps (0--15K step)

Online change area of multiple blocks. 4 K steps

(Online change area of FB definition/ST.)

☐ Auto Download all Data from Memory card to Standard ROM

Boot file setting

Program
- MAIN

Parameter
- PARAM

	Type	Data name	Transfer from	Transfer to
1	Sequence	MAIN	Memory card (RAM)	Program memory
2	Parameter	PARAM	Memory card (RAM)	Program memory
3	Sequence	CONSIST	Memory card (RAM)	Program memory
4	Parameter	IPARAM	Memory card (RAM)	Program memory
5				
6				
7				
8				
9				
10				
11				
12				

Fig. 8-110: *Boot file setting*

The Intelligent Parameter file is added to the **Boot file setting** list automatically. Additionally, editing the “Boot file setting” list is enabled. This causes the following behaviour:

- The column **Transfer from** is editable for all entries, allowing you to select the drive where the PLC files are transferred from.
- The buttons **Insert** and **Delete** are used to insert or delete an entry in the **Boot file setting** list.

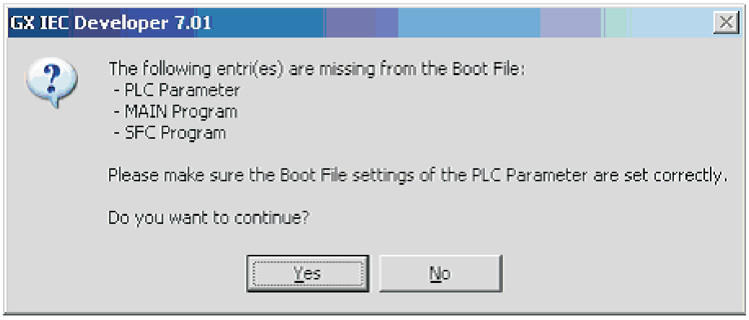
NOTE

However, in order to preserve consistency in the project, if CONSIST.MMP is removed from the **Boot file setting** list, it will be re-added automatically as soon as the dialogue is closed. The **Transfer from** drive of CONSIST.MMP will be set to the same drive as for the MAIN program.

- An entry for a SFC program will be added or removed automatically during compile, depending on the existence of a SFC program in the project. The **Transfer from** drive of the SFC program will be set to the same drive as for the MAIN program.

Checking the Boot File before download

If the Boot File is selected for download, its entries will be checked during the initialization of the Download process. The following warning will be displayed if either the Parameter, the MAIN or (if it exists in the project) the SFC program is missing from the boot File:



GX IEC Developer 7.01

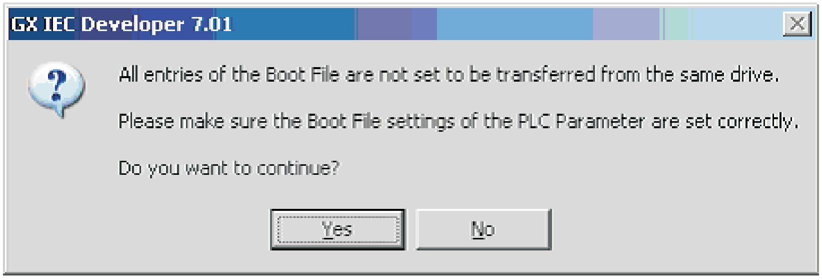
The following entri(es) are missing from the Boot File:
- PLC Parameter
- MAIN Program
- SFC Program

Please make sure the Boot File settings of the PLC Parameter are set correctly.

Do you want to continue?

Fig. 8-111:
Message

The following warning will be displayed if the **Transfer from** drive of all entries is not set to the same drive:



GX IEC Developer 7.01

All entries of the Boot File are not set to be transferred from the same drive.

Please make sure the Boot File settings of the PLC Parameter are set correctly.

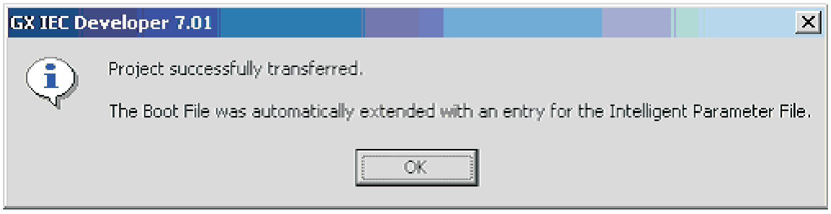
Do you want to continue?

Fig. 8-112:
Warning

When the PLC Parameter dialogue is opened, it is automatically checked if the Intelligent Parameter File exists in the project. If it does, the display of the **Boot File setting** is modified, so it includes an entry for the Intelligent Parameter File. The **Transfer from** drive of the Intelligent Parameter File is set to the same drive as for the PLC Parameter.

At the beginning of the Download process, it is checked if the Boot File and the Intelligent Parameter File are selected to be transferred. If both files were selected, the Boot File will be modified before download to include an entry for the Intelligent Parameter File (if this entry does not exist already).
The **Transfer from** drive of the Intelligent Parameter File is set to the same drive as for the PLC Parameter. The existence of the IPARAM file in the project will be checked before download. If no IPARAM file exists in the project, no entry will be added to the Boot File of the Intelligent Parameter File. If the entry already exists, it will be removed.

The message **Download success** will be shown, in order to inform you:



GX IEC Developer 7.01

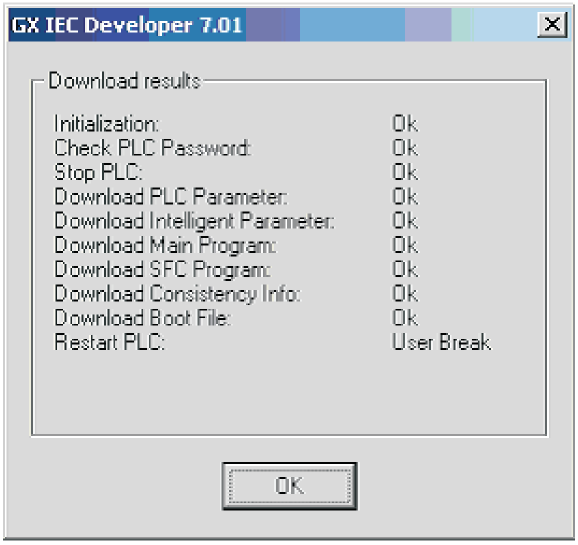
Project successfully transferred.

The Boot File was automatically extended with an entry for the Intelligent Parameter File.

Fig. 8-113:
Download success

Download and Upload of the Intelligent Parameter

The dialogue showing the **Download results** also includes the status of the Intelligent Parameter. If the download of IPARAM was successful, but the download process fails or is aborted later, the following dialogue will be displayed:


GX IEC Developer 7.01

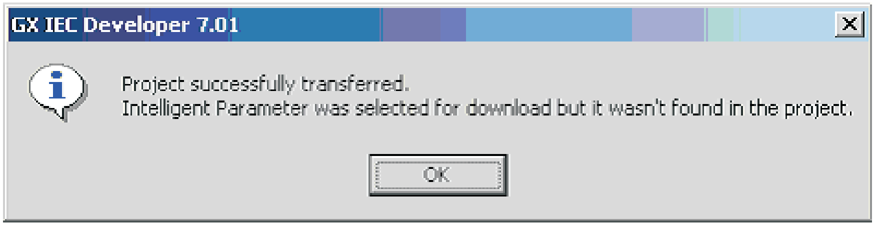
Download results

```
Initialization:                    Ok
Check PLC Password:                Ok
Stop PLC:                          Ok
Download PLC Parameter:            Ok
Download Intelligent Parameter:    Ok
Download Main Program:             Ok
Download SFC Program:              Ok
Download Consistency Info:         Ok
Download Boot File:                Ok
Restart PLC:                       User Break
```

Fig. 8-114:
Download results

NOTE When downloading to both systems of a QnPRH PLC, the redundant version of this dialogue will be used. In this case, the same file is downloaded to both systems (as with all other files during a redundant download).

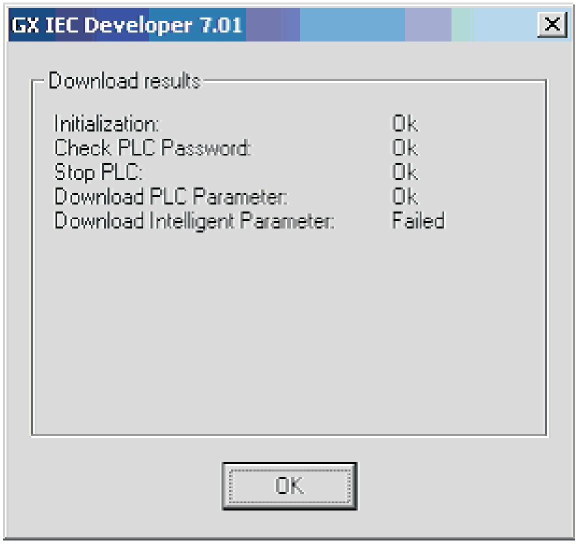
If the download of the Intelligent Parameter fails because the project does not contain the Intelligent Parameter (IPARAM.WPA) file, the download process will not be interrupted. In this case, the following dialogue will be shown:


GX IEC Developer 7.01

Project successfully transferred.
Intelligent Parameter was selected for download but it wasn't found in the project.

Fig. 8-115:
Message

If the download of the Intelligent Parameter fails because of any other reason, the download process will be aborted and the dialogue showing the **Download Results** will be displayed:

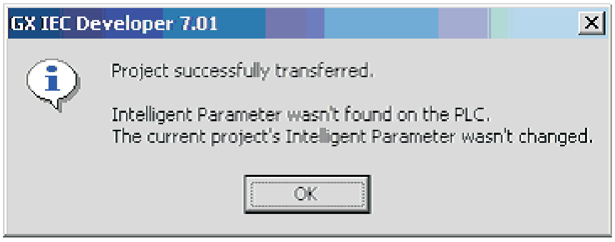

GX IEC Developer 7.01

Download results

```
Initialization:                    Ok
Check PLC Password:                Ok
Stop PLC:                          Ok
Download PLC Parameter:            Ok
Download Intelligent Parameter:    Failed
```

Fig. 8-116:
Download results

If the upload of the Intelligent Parameter fails because the PLC drive does not contain the Intelligent Parameter (IPARAM.QPA) file, the upload process will not be interrupted. In this case, the following dialogue will be displayed:

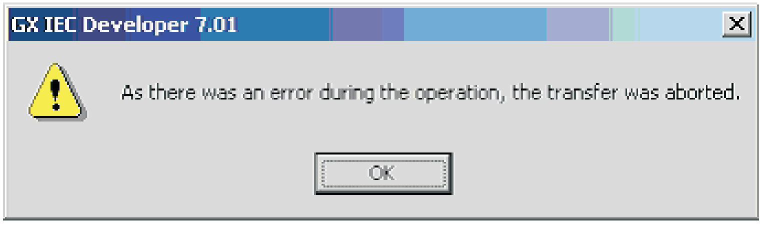


Project successfully transferred.

Intelligent Parameter wasn't found on the PLC.
The current project's Intelligent Parameter wasn't changed.

Fig. 8-117:
Message

If the upload of the Intelligent Parameter fails because of any other reason, the upload process will be aborted and the same dialogue will be displayed as for any other error occurring during the upload:



As there was an error during the operation, the transfer was aborted.

Fig. 8-118:
Warning

8.6.2 Download symbolic information by menu command

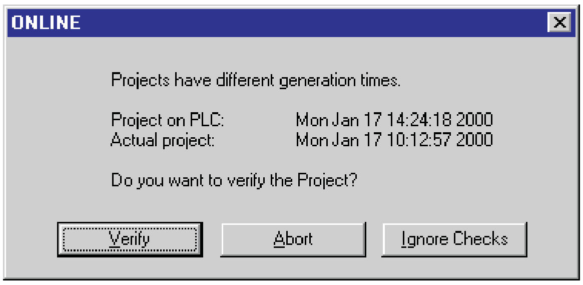
The menu **Project - Transfer** includes a menu command to download symbolic information of the current project to the PLC. This is independent of the current settings concerning source information in the transfer setup.

This menu command will be shown only when the current PLC type is A/QnA/Q.

In a redundant PLC system in backup mode, the user will be asked (via a confirmation dialogue) if he wants to download symbolic information of the current project to both PLCs. (Please refer to section 3.7.2 for details of the confirmation dialogue.)

8.6.3 Ignore checks

If the project currently open in GX IEC Developer and the project on the connected PLC have different time stamps, there is the possibility to continue the current action, although the projects are not equal.



ONLINE

Projects have different generation times.

Project on PLC: Mon Jan 17 14:24:18 2000
Actual project: Mon Jan 17 10:12:57 2000

Do you want to verify the Project?

Fig. 8-119:
Different project generation times are notified.

This dialogue is used for the following actions:

- Project online program change
- Online change mode
- Download symbolic information

During monitoring the button **Ignore Checks** is enabled. In this non-critical case the user is allowed to monitor the projects although the projects are not equal. In all other cases the button **Ignore Checks** is disabled.

It is not possible to ignore checks when starting a *project online program change* or entering the *online change mode*.

8.6.4 The Q/QnA's File Structure

How to format and defragment the Q/QnA series drive

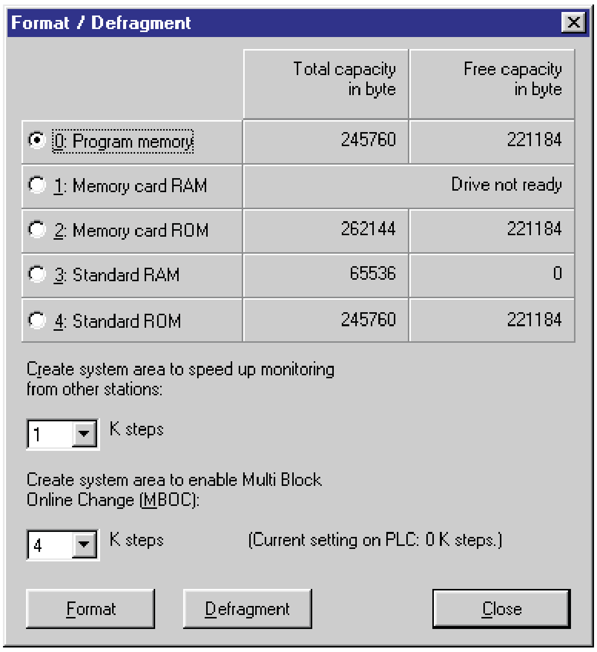
Before you can use the memory in the MELSEC Q/QnA series CPUs you must first format the corresponding drive. This applies both for the internal RAM and the external memory cards.

For details on the available drives refer to the table on page 8-85.

① Select **Format Drive** in the Online menu. The **Format / Defragment** dialogue box is displayed.

In a redundant PLC system in backup mode, the user will be asked (via a confirmation dialogue) if he wants to format the specified drive of both PLCs. (Please refer to section 3.7.2 for details of the confirmation dialogue.)

Similarly, before defragmenting any of the drives in Backup mode, the user will be asked (via a confirmation dialogue) if he wants to defragment the specified drive of both systems. (Please refer to section 3.7.2 for details of the confirmation dialogue.)



Format / Defragment

	Total capacity in byte	Free capacity in byte
● 0: Program memory	245760	221184
○ 1: Memory card RAM		Drive not ready
○ 2: Memory card ROM	262144	221184
○ 3: Standard RAM	65536	0
○ 4: Standard ROM	245760	221184

Create system area to speed up monitoring from other stations:

1 K steps

Create system area to enable Multi Block Online Change (MBOC):

4 K steps (Current setting on PLC: 0 K steps.)

Fig. 8-120: *Format / Defragment dialogue*

0 … 4: Drive to be formatted or defragmented.

Create system area to speed up monitoring...: If a Q/QnA is connected to GX IEC Developer and a remote Q/QnA is to be monitored, both PLCs (host and remote) have to be formatted with at least 1-K steps system area available in the dialogue Format / Defragment.
If one of both PLCs is not formatted with this system area, the remote Q/QnA cannot be monitored.

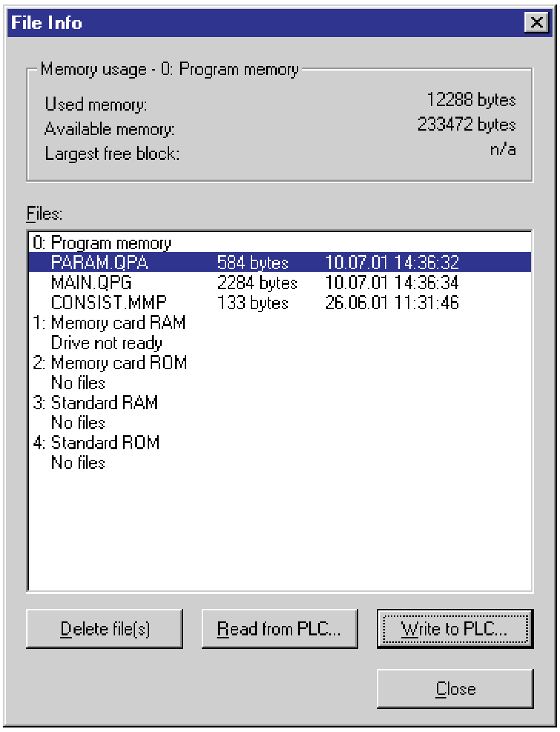
Create system area to enable Multi Block Online Changes… : In the CPUs supporting the MBOC function the number of steps which can be exchanged in the online mode has been expanded to 1024 steps. These 1024 steps do not have to be within one block. It is also possible to make changes in several small blocks. The maximum number of blocks is fixed to 64. However, the number of 1024 steps must not be exceeded (Q series only except Q00(J), Q01 CPUs).

Format: Selecting this button starts the format procedure.

Defragment: Because of the file structure used in the Q/QnA series CPUs it is possible for the drives to become fragmented after data have been written to the CPU, just as with a normal hard drive. Selecting the **Defragment** button executes a procedure that defragments the contents of the drive for better performance.

How to get File Information

① Select **File Info** in the **Online** menu. The **File Info** dialogue box displays information on the memory and the files in the CPU.



File Info

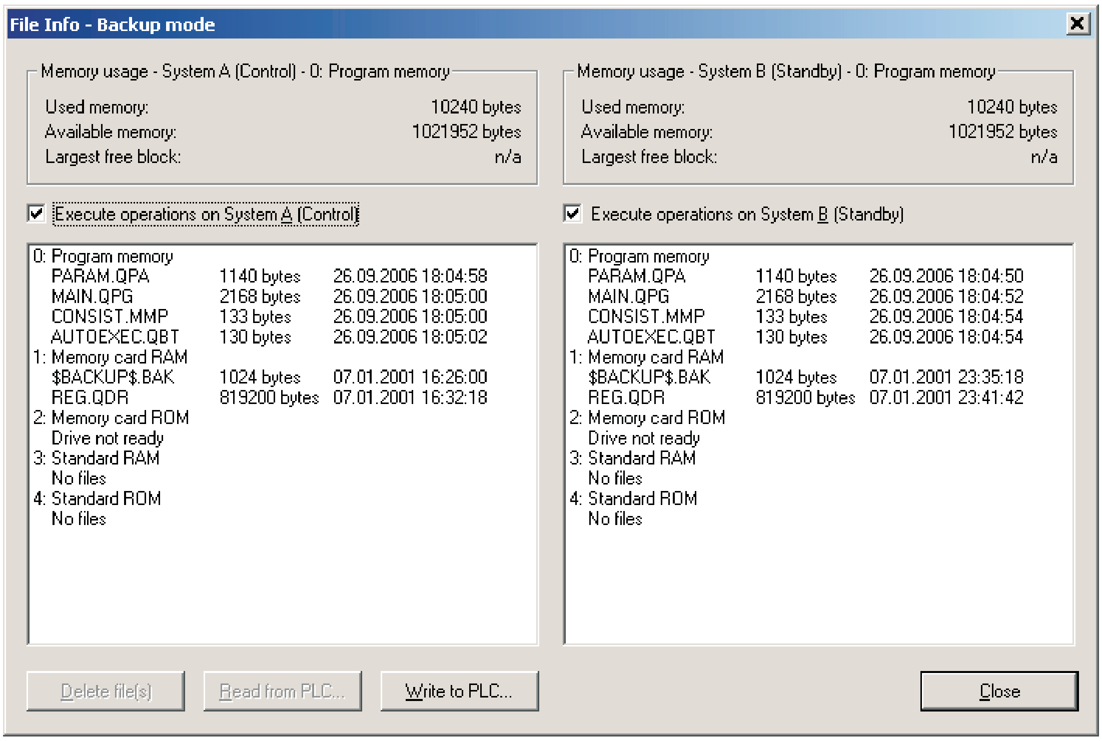
Memory usage - 0: Program memory

```
Used memory:              12288 bytes
Available memory:        233472 bytes
Largest free block:               n/a
```

Files:

```
0: Program memory
   PARAM.QPA      584 bytes    10.07.01 14:36:32
   MAIN.QPG       2284 bytes   10.07.01 14:36:34
   CONSIST.MMP    133 bytes    26.06.01 11:31:46
1: Memory card RAM
   Drive not ready
2: Memory card ROM
   No files
3: Standard RAM
   No files
4: Standard ROM
   No files
```

Fig. 8-121:
File information



File Info - Backup mode

Memory usage - System A (Control) - 0: Program memory

```
Used memory:              10240 bytes
Available memory:       1021952 bytes
Largest free block:               n/a
```

Memory usage - System B (Standby) - 0: Program memory

```
Used memory:              10240 bytes
Available memory:       1021952 bytes
Largest free block:               n/a
```

☑ Execute operations on System A (Control)

☑ Execute operations on System B (Standby)

```
0: Program memory
   PARAM.QPA      1140 bytes     26.09.2006 18:04:58
   MAIN.QPG       2168 bytes     26.09.2006 18:05:00
   CONSIST.MMP    133 bytes      26.09.2006 18:05:00
   AUTOEXEC.QBT   130 bytes      26.09.2006 18:05:02
1: Memory card RAM
   $BACKUP$.BAK   1024 bytes     07.01.2001 16:26:00
   REG.QDR        819200 bytes   07.01.2001 16:32:18
2: Memory card ROM
   Drive not ready
3: Standard RAM
   No files
4: Standard ROM
   No files
```

```
0: Program memory
   PARAM.QPA      1140 bytes     26.09.2006 18:04:50
   MAIN.QPG       2168 bytes     26.09.2006 18:04:52
   CONSIST.MMP    133 bytes      26.09.2006 18:04:54
   AUTOEXEC.QBT   130 bytes      26.09.2006 18:04:54
1: Memory card RAM
   $BACKUP$.BAK   1024 bytes     07.01.2001 23:35:18
   REG.QDR        819200 bytes   07.01.2001 23:41:42
2: Memory card ROM
   Drive not ready
3: Standard RAM
   No files
4: Standard ROM
   No files
```

Fig. 8-122: *File information for a redundant system*

For details on the available drives refer to the table on page 8-85.

Memory usage - ***drive***

Used memory: Amount of downloaded data (unit: bytes)

Available memory: Amount of memory available for use (unit: bytes)

Largest free block: Largest free memory block available (unit: bytes)
In case this block is not large enough, use the Defrag option.

Execute operations on System ...: If one of the **Execute operations** check boxes of the extended dialog for the redundant system is unchecked, the corresponding file list will be disabled and any selections removed.
If both of the **Execute operations** check boxes are unchecked, the **Delete**, **Read** and **Write** buttons will become disabled.

Files

0 … 4: Files stored in the program memory (Drive 0), in the RAM or ROM of the memory card (Drives 1 and 2), and in the standard RAM or standard ROM (Drives 3 and 4).

NOTES

Full details on the file structure used by the MELSEC Q/QnA series CPUs can be found in the CPU manuals.

The extended dialog for the redundant system is used in Backup and Separate mode if both systems are accessible. In Debug mode, or if the other system is not accessible, the standard version of this dialog is used.

Transfer files between personal computer and PLC

From the **File Info** dialogue selected PLC files can be transferred from the PLC drives to the hard disk of your personal computer and vice versa. They can also be deleted. Multiple files can be selected via mouse-click in combination with the **[CTRL]** or **[SHIFT]** key.

This function is for example useful for storing the documentation of the currently running process on a PLC drive to make it always available.

Delete file(s)

Deletes the selected files from the PLC. Multiple files can only be deleted from one single drive. Only files can be deleted (not drives etc.).

Read from PLC

Reads the selected files from the PLC. A standard file dialogue is opened where you specify the destination folder for the selected PLC file(s) on your hard disk. Multiple files can only be read from one single drive.

Write to PLC

Writes the selected files to the selected PLC drive. A standard file dialogue is opened where you specify the source file(s) on your hard disk. Multiple files can only be written from one single directory. Writing to flash ROM drives is not supported.

NOTE

Please also see the remarks on the following page!

Remarks on stopping and restarting the PLC during the "Write to PLC" operation

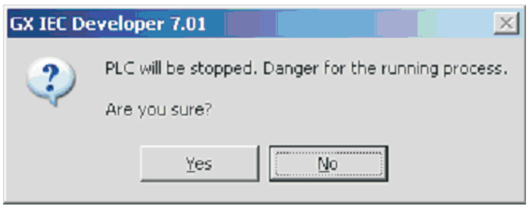
Some files (e.g. Program and Parameter files) cannot be written to Q and QnA PLCs via the **File Info** dialogue when the PLC is in RUN mode. In previous versions of GX IEC Developer, attempting to write such files during RUN failed with an EasySocket error message.

The dialogues **Stop PLC** and **Restart PLC** (normaly used to stop and restart the PLC e.g. during a **Delete file** operation) can also be used to stop and restart the PLC during a **Write to PLC** operation if any system files (i.e. files used by the GX IEC Developer project) are selected.

The following files will be considered to be system files:

- MAIN.QPG
- SFC.QPG or MAIN-SFC.QPG
- PARAM.QPA
- IPARAM.QPA
- SYMPARAM.MMP
- CONSIST.MMP
- SYMBOLIC.MMP
- LIBRARY.MMP
- AUTOEXEC.QBT
- MBOC.QMB

When the first such file is encountered during **Write to PLC**, the following dialogue will be displayed if the PLC is in RUN mode:

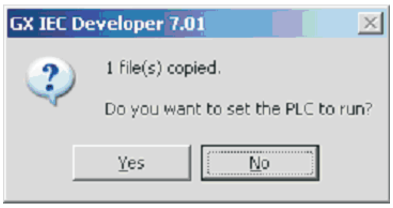


PLC will be stopped. Danger for the running process.

Are you sure?

Fig. 8-123:
Warning that PLC will be stopped

When the Write to PLC operation is finished, the following dialogue will be displayed:

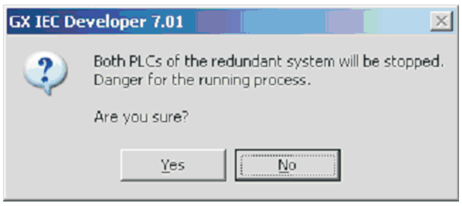


1 file(s) copied.

Do you want to set the PLC to run?

Fig. 8-124:
Message about copying result

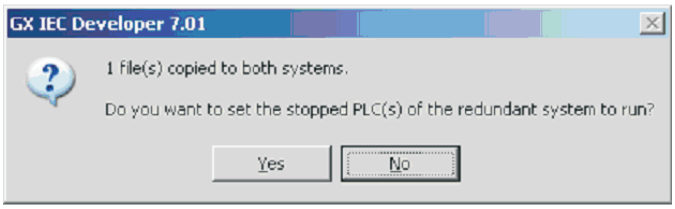
For QnPRH PLCs in Backup mode, or if both systems are selected in Separate mode, the redundant version of these dialogues will be used:



Both PLCs of the redundant system will be stopped.
Danger for the running process.

Are you sure?

Fig. 8-125:
Warning that both PLCs of a redundant system will be stopped



1 file(s) copied to both systems.

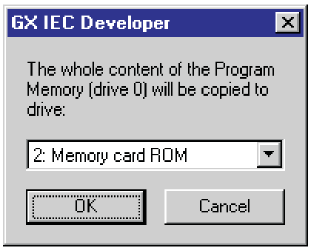
Do you want to set the stopped PLC(s) of the redundant system to run?

Fig. 8-126:
Message about copying result for a redundant system

8.6.5 Copy program memory to flash ROM (MELSEC Q only)

The complete PLC program memory content can be copied to the selected flash ROM drive. The flash ROM drive can even be unformatted. All files stored on the flash ROM drive are overwritten.

Select the menu item **Project - Transfer - Copy Program Memory to Flash ROM** to open the following dialogue.



GX IEC Developer

The whole content of the Program Memory (drive 0) will be copied to drive:

2: Memory card ROM

Fig. 8-127:
Copy Program Memory to Flash ROM dialogue

Select the desired flash ROM drive and confirm with **OK**.

In a redundant PLC system in backup mode, the user will be asked (via a confirmation dialogue) if he wants to copy the program memory to the Flash ROMs of both PLCs. (Please refer to section 3.7.2 for details of the confirmation dialogue.)

Supported flash ROM drives

Only the following drives are supported as copy destinations.

Drive	Q	Q00(J), Q01
2	Memory card ROM (only flash type supported)	not available
4	Standard ROM	Standard ROM

Tab. 8-4: *Available copy destination drives*

8.6.6 Downloading the Project

When you have checked the port settings and project parameters you can download your project to the controller.

How to download the project to the PLC

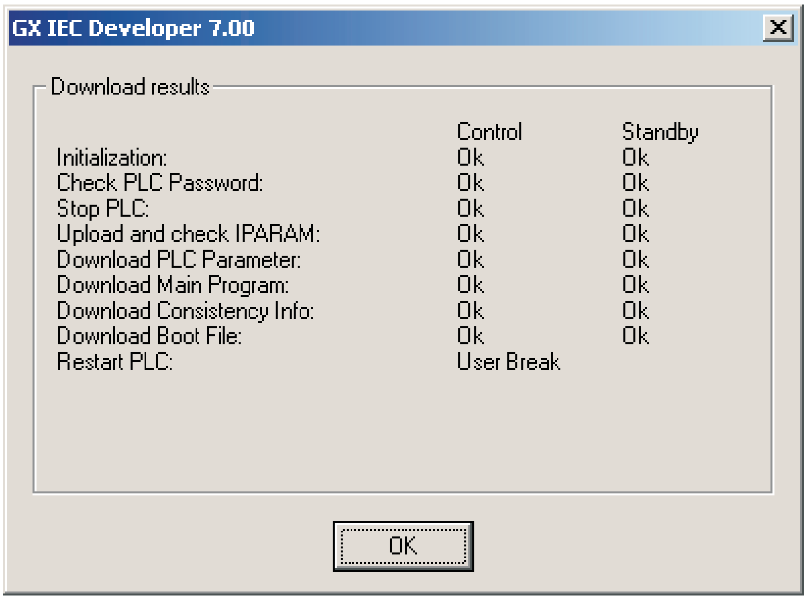
① Select **Transfer** in the **Project** menu, then select **Download to PLC**.

NOTES

In a redundant PLC system in backup mode, the user will be asked (via a confirmation dialogue) if he wants to download the project to both PLCs. (Please refer to section 3.7.2 for details of the confirmation dialogue.)

In case of a Q Remote I/O module the download function transfers only the parameter and (if selected) the intelligent parameter to drive 4 of the PLC.

The progress of the download process will be documented in a status window.



Download results

	Control	Standby
Initialization:	Ok	Ok
Check PLC Password:	Ok	Ok
Stop PLC:	Ok	Ok
Upload and check IPARAM:	Ok	Ok
Download PLC Parameter:	Ok	Ok
Download Main Program:	Ok	Ok
Download Consistency Info:	Ok	Ok
Download Boot File:	Ok	Ok
Restart PLC:	User Break	

Fig. 8-128:
Status window of the download process

The successful transfer to the PLC is confirmed by a message box.

Check list before downloading

- Has the project been compiled without errors?
- Are the port parameters set properly for the transfer?
- Are the project parameters set correctly?

8.7 Comparison and Online Change of Project Data

8.7.1 Consistency

To verify the consistency between the data in the project and on the PLC, the necessary data must be uploaded from the PLC, which can take a long time with large projects.
To provide a quick way to check if the parameter and program data in the project and on the PLC are identical, consistency information is maintained on the PLC for projects downloaded by GX IEC Developer. The project consistency check is based on the timestamp stored in the consistency information, which represents the time when the program codes were generated. If there is no consistency information on the PLC, or if it is incorrect (not downloaded by GX IEC Developer), consistency cannot be determined by this quick method.

Hereafter the exact verification of consistency will be referred to as *verification of consistency by the long method*. The quick consistency check, based on the consistency information downloaded to the PLC, will be referred to as *checking consistency by the quick method* or *quick consistency check*.

Parameter and program data on the PLC are considered to be consistent with that in the project, if the following conditions are met:

For all PLC types:

- The project is compiled without any error.
- Code generation options and PLC parameter have not been changed in the project since last compilation.

Additionally for Q/QnA series PLC types:

- The parameter in the project and the parameter on the PLC are identical, excluding the parts of the parameter which are modified by the PLC after switching it to RUN mode.
- The main program in the project and the main program on the PLC are identical.
- The SFC program in the project and the SFC program on the PLC are identical (if SFC program exists in the project), considering SFC program file headers as well.

NOTE

When online verification is called to confirm consistency, the boot settings will not be compared. The reason is that differences in the boot settings do not affect the functionalities that require consistency.

Additionally for A series PLC types:

- The parameter in the project and the parameter on the PLC are identical, excluding the parts of the parameter which are modified by the PLC after switching it to RUN mode.
- The main program in the project and the main program on the PLC are identical.
- The timer initial values of the main program are identical in the project and on the PLC.
- The counter initial values of the main program are identical in the project and on the PLC.
- The sub program in the project and the sub program on the PLC are identical (if sub program exists in the project).
- The timer initial values of the sub program are identical in the project and on the PLC (if sub program exists in the project).
- The counter initial values of the sub program are identical in the project and on the PLC (if sub program exists in the project).
- The SFC program in the project and the SFC program on the PLC are identical (if SFC program exists in the project).

Additionally for FX series PLC types:

- The parameter in the project and the parameter on the PLC are identical, excluding the parts of the parameter which are modified by the PLC after switching it to RUN mode.
- The program in the project and the program on the PLC are identical.

There are situations when the consistency information is wrong, i.e. inconsistency is indicated although the parameter and the program codes are identical, or consistency is indicated although the parameter and/or the program codes are different.
The first case is inconvenient for the user, because he must then verify the consistency by the long method, which can take much longer than the quick consistency check.
The second case is dangerous, because functionalities assuming consistency from the quick check can cause erroneous operations when the data in the project and on the PLC are actually inconsistent.
Solutions for both of these issues are available.

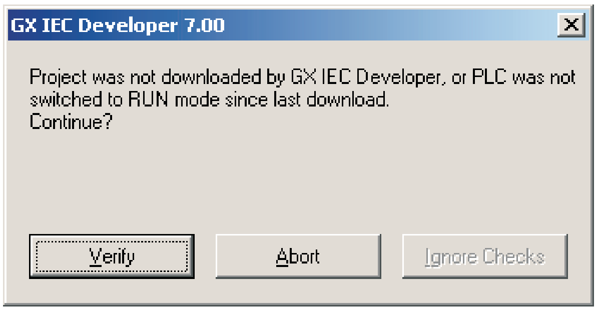
False indication of inconsistency – due to missing or invalid consistency information

It is possible that the consistency information in the PLC's memory is invalid, or there is no consistency information at all on the PLC. Invalid consistency information means that the data existing in the location of supposed GX IEC Developer consistency information is actually not GX IEC Developer consistency information.

This can happen, because...

- the project was not downloaded by GX IEC Developer.
- the consistency information on the PLC has been (accidentally) overwritten or deleted by the user.
- the data on the PLC has been changed in an unsupported way (i.e. not by GX IEC Developer).
- for FX and A PLCs: the PLC has not been switched to RUN mode since last download.

In this case, when a functionality that requires consistency is performed, the user will be presented a dialogue where he is noticed about inconsistency – based on the validity of the consistency information –, and he is offered to verify consistency by the long method. Verifying consistency can take long, so it is desirable to perform it as infrequently as possible.

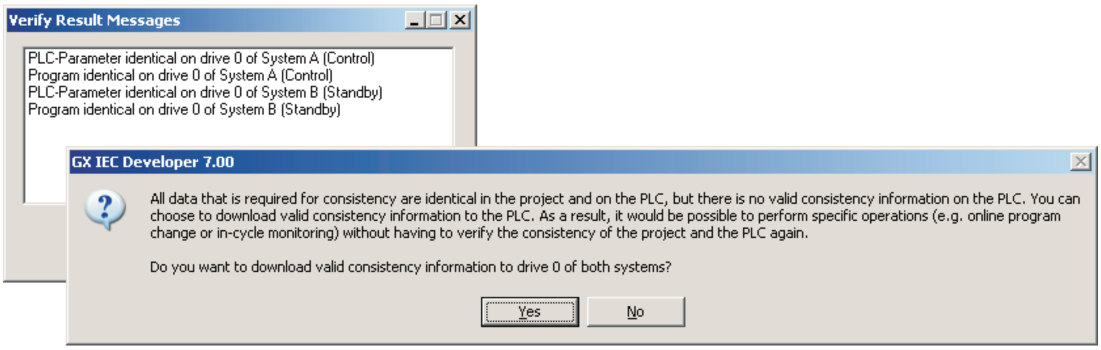


GX IEC Developer 7.00

Project was not downloaded by GX IEC Developer, or PLC was not switched to RUN mode since last download.
Continue?

Fig. 8-129:
Message box

The **Online Verify** functionality is extended, so that when **Project** → **Transfer** → **Verify** is called, it will determine if the consistency information on the PLC is valid. If the verification finds that all required data in the project is identical with that of the PLC, and there is no valid consistency information on the PLC, the user will be asked whether he wants to download valid consistency information to the PLC. The question will be shown after displaying all the verification results in the **Verify Result Messages** window (please see Tab. 4-1 and Tab. 4-2 for details on the messages).



PLC-Parameter identical on drive 0 of System A (Control)
Program identical on drive 0 of System A (Control)
PLC-Parameter identical on drive 0 of System B (Standby)
Program identical on drive 0 of System B (Standby)

All data that is required for consistency are identical in the project and on the PLC, but there is no valid consistency information on the PLC. You can choose to download valid consistency information to the PLC. As a result, it would be possible to perform specific operations (e.g. online program change or in-cycle monitoring) without having to verify the consistency of the project and the PLC again.

Do you want to download valid consistency information to drive 0 of both systems?

Fig. 8-130: *Message box*

("Drive 0" in the question will be replaced by the drive set in the project transfer setup for Q- and QnA series PLCs. It will be replaced by "the PLC" for FX- and A series PLCs.)

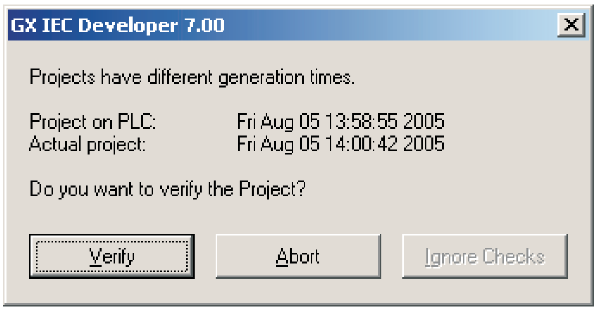
If the user clicks on **Yes**, consistency information will be generated and downloaded to the PLC. As a result, next time when an operation requiring consistency is called, it will be able to detect consistency by the quick method.

NOTE

This feature is not available when the drive selected for download in the project transfer setup (**Online** → **Transfer Setup** → **Project**) of a Q or QnA series PLC is Flash ROM type.

False indication of inconsistency – due to different program code generation times

When there is valid consistency information on the PLC, it is possible that the program code generation times differ even with identical parameters and program codes. For example if the project is changed and compiled, then changed back and compiled again. In this case, when a functionality that requires consistency is performed, the user will be presented a dialogue where he is noticed about inconsistency – based on code generation times –, and he is offered to verify consistency by the long method. Verifying consistency can take long, so it is desirable to perform it as infrequently as possible.



Projects have different generation times.

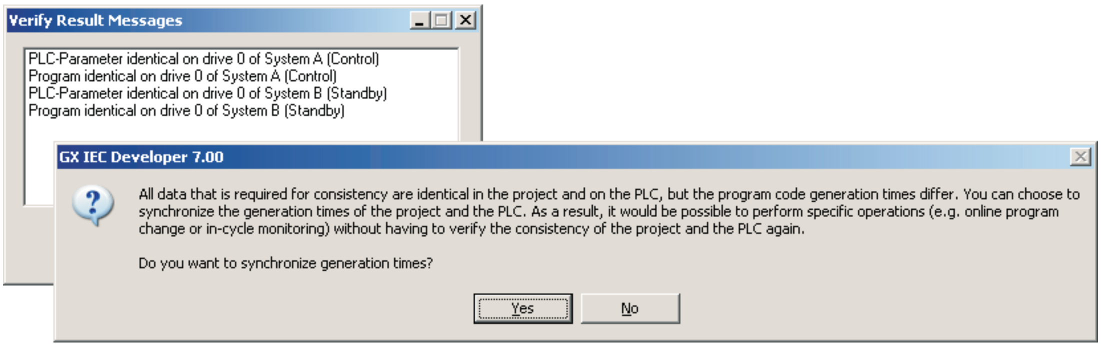
Project on PLC: Fri Aug 05 13:58:55 2005
Actual project: Fri Aug 05 14:00:42 2005

Do you want to verify the Project?

Fig. 8-131: *Message box*

In the current version, when consistency verification is invoked from this dialogue, and it is proved that the parameters and the program codes are identical, the generation times in the consistency information of the project and the PLC will be synchronized, so that next time the quick consistency check will find consistency.

On the other hand, when **Project** → **Transfer** → **Verify** was called, it did not synchronize generation times in the same situation. So the user will be asked whether the consistency information should be updated. The question will be shown after displaying all the verification results in the **Verify Result Messages** window.



PLC-Parameter identical on drive 0 of System A (Control)
Program identical on drive 0 of System A (Control)
PLC-Parameter identical on drive 0 of System B (Standby)
Program identical on drive 0 of System B (Standby)

All data that is required for consistency are identical in the project and on the PLC, but the program code generation times differ. You can choose to synchronize the generation times of the project and the PLC. As a result, it would be possible to perform specific operations (e.g. online program change or in-cycle monitoring) without having to verify the consistency of the project and the PLC again.

Do you want to synchronize generation times?

Fig. 8-132: *Message box*

If the user clicks on **Yes**, the generation time of the project will be updated to be the same as the generation time on the PLC. The project has to be saved to keep this change.

False indication of consistency

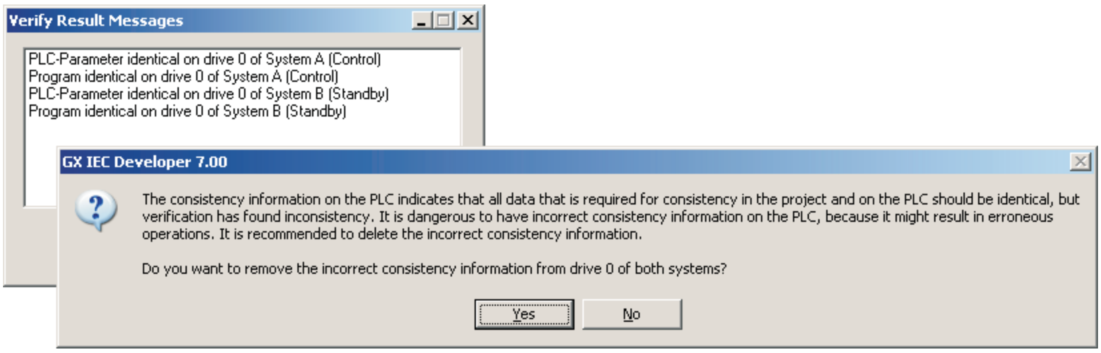
It is possible that the program code generation times are equal even with different parameters and/or program codes.

It is a rather rare situation, but it can happen if...

- a software other than GX IEC Developer was used to download or change data on the PLC, when the PLC has project data downloaded by GX IEC Developer**.**
- system files were directly manipulated on the PLC with GX IEC Developer or another software**.**

It is dangerous to have incorrect consistency information on the PLC, when the parameters and/or the program codes in the project and on the PLC are different. It can easily result in malfunction when an operation – e.g. online program change – falsely assumes consistency by the quick method.

The **Online Verify** functionality is extended, so that when the user calls **Project** → **Transfer** → **Verify**, such a false indication of consistency will be detected and the user will be asked if the incorrect consistency information should be deleted.



PLC-Parameter identical on drive 0 of System A (Control)
Program identical on drive 0 of System A (Control)
PLC-Parameter identical on drive 0 of System B (Standby)
Program identical on drive 0 of System B (Standby)

The consistency information on the PLC indicates that all data that is required for consistency in the project and on the PLC should be identical, but verification has found inconsistency. It is dangerous to have incorrect consistency information on the PLC, because it might result in erroneous operations. It is recommended to delete the incorrect consistency information.

Do you want to remove the incorrect consistency information from drive 0 of both systems?

Fig. 8-133: *Message box*

("Drive 0" in the question will be replaced by the drive set in the project transfer setup for Q- and QnA series PLCs. It will be replaced by "the PLC" for FX- and A series PLCs.)

If the user clicks on **Yes**, the consistency information will be removed from the PLC.

NOTE

This feature is not available when the drive selected for download in the project transfer setup (**Online** → **Transfer Setup** → **Project**) of a Q or QnA series PLC is Flash ROM type.

NOTES

The FX0 and FX0S PLCs have no consistency information, therefore this functionality is not applicable to FX0 and FX0S PLCs.

Downloading or deleting consistency information in Flash ROM type memories is not supported. This feature is not available when the drive selected for download in the project transfer setup (**Online** → **Transfer Setup** → **Project**) of a Q or QnA series PLC is Flash ROM type.

8.7.2 Comparing the programs in the PLC CPU and GX IEC Developer

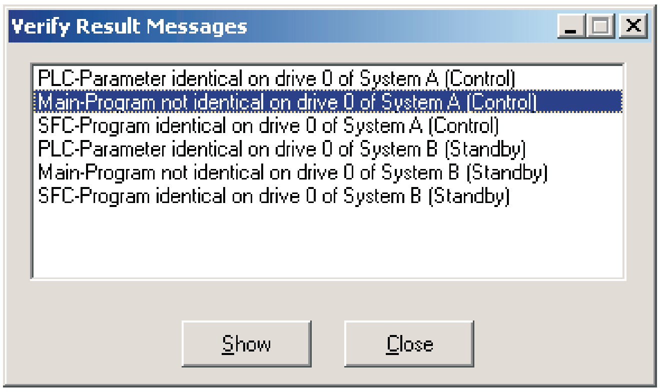
The function **Online verify** in the menu **Project - Transfer** compares the program code stored in the PLC CPU with the code of the current project compiled by GX IEC Developer.

NOTE

In a redundant PLC system in backup mode, the user will be asked (via a confirmation dialogue) if he wants to verify both PLCs. (Please refer to section 3.7.2 for details of the confirmation dialogue.)

Only sequence code differences are shown. Differences for SFC programs of A series CPUs which will be transferred to the main micro computer area will not be shown.

For a redundant PLC system in backup mode the verify process will display items for both PLCs in the **Verify Result Messages** window.



PLC-Parameter identical on drive 0 of System A (Control)
Main-Program not identical on drive 0 of System A (Control)
SFC-Program identical on drive 0 of System A (Control)
PLC-Parameter identical on drive 0 of System B (Standby)
Main-Program not identical on drive 0 of System B (Standby)
SFC-Program identical on drive 0 of System B (Standby)

Fig. 8-134:
Verify Result Messages

NOTES

Please see Tab. 4-1 and Tab. 4-2 for details on these messages.

For **Online Verify** the message **Boot file not found** is displayed, if there is no boot file in the project on the PLC compared with.

Detected differences in the program code can be shown in a **Main** or **Sub Program Difference** window via the **Show** button. The verify results are listed as recompiled MELSEC IL, the differences are highlighted in the error color. The left part of the window lists the project code, the right part of the window lists the uploaded PLC code.

When the **Main** or **Sub Program Difference** window is opened the following menu items are available in the **Tools** menu:

Next Difference: **Next Difference** or the tabulator key will position the text in the window so that the first line of the next difference is the top most line. The top most line will be selected. The according calltree, number of network and for IL and ST the line in the network of the selected line are shown in the status bar. If no next difference exists, the menu item is greyed.

Previous Difference: **Previous Difference** or the shift tabulator key will position the text in the window so that the first line of the previous difference is the top most line. The top most line will be selected. The according calltree, number of network and for IL and ST the line in the network of the selected line are shown in the status bar. If no previous difference exists, the menu item is greyed.

Goto Editor: **Goto Editor** or a double click on a selected line will open the according POU body. In IL/ ST the according line and in LD/ FBD the according network is positioned to the visible area and marked with the browser colour in the network column.
If no POU body relates to the selected line a message window is opened to inform you of this circumstance. The message text will be “The selected line belongs to code generated for task or program handling. No editor could be opened.”

Show Step Number: This menu item is checked as default. If it is checked, the step number for each instruction for the code of the project and of the PLC are displayed.
If it is not checked, the step number are not displayed. If a line is selected and the step number are not displayed, the line is marked with a small box in front of the instruction.

8.7.3 Online Changes

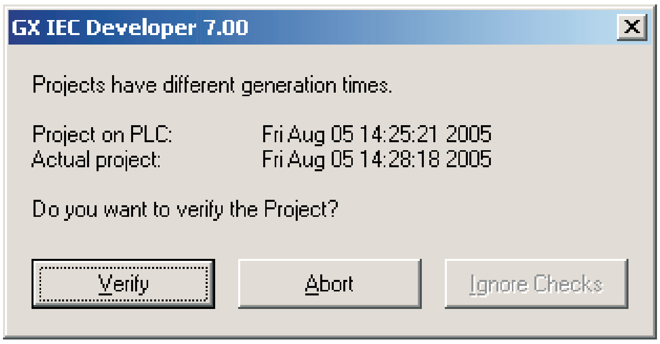
There are three different ways to make changes to programs when the controller is in RUN mode:

Project – Online Program Change and **Online Program Exchange** respectively:
These options compile the whole modified program and transfer it to the PLC.

Online – Online Change Mode:
Selecting this option compiles changes made in a program network and directly transfers them to the PLC.

Tool for the menu command **Online - Online Change Mode**

In general Online Change only works if the compiled program codes in the project and the PLC are identical. After the project is changed and compiled, online change cannot be performed, because the compiled program codes are different in the actual project and on the PLC. In this case the projects in GX IEC Developer and on the PLC are called inconsistent (→ 8.7.1). You will be shown the following dialogue when an online change is attempted:



Projects have different generation times.

Project on PLC: Fri Aug 05 14:25:21 2005
Actual project: Fri Aug 05 14:28:18 2005

Do you want to verify the Project?

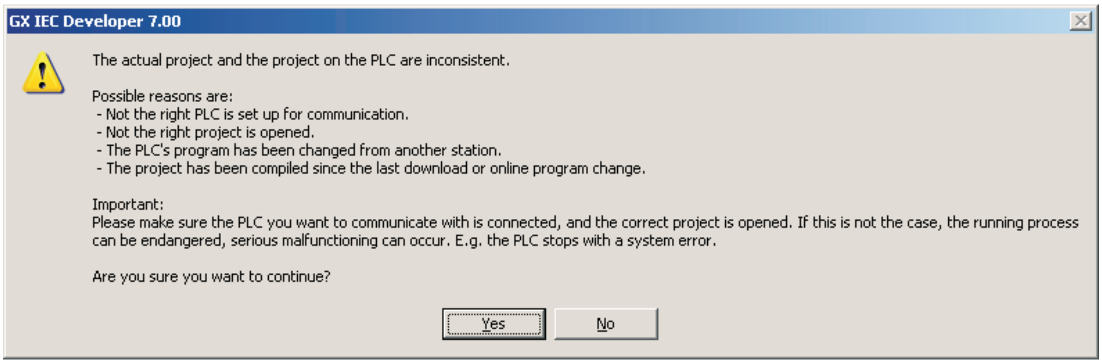
Fig. 8-135:
Message window

This result is based on the comparison of the generation times of the program codes in the project and on the PLC (→ 8.7.1).
The generation times can differ even if the program codes are identical (e.g. when the project is changed, compiled and then changed back to its original state and compiled again). The **Verify** button can be used to check if the program codes and the parameter are really different. If the parameter and the program codes turn out to be identical, online change will be performed.

Online program change needs to compare the differences between the old and the new program codes. If the two projects are inconsistent, the old program code exists only on the PLC, not in the project, so the new program code cannot be compared with the old one.

When the parameters in the actual project and on the PLC don’t differ, the program code on the PLC can be uploaded and used as “old code” to be compared with the new code when online program change is prepared. This way it will be possible to perform an online change even if (an accidental) Rebuild was called before.

After pressing the **Verify** button in the consistency dialogue shown above, if the inconsistency is confirmed, you will be warned and asked whether you still want to perform online program change.



GX IEC Developer 7.00

The actual project and the project on the PLC are inconsistent.

Possible reasons are:
- Not the right PLC is set up for communication.
- Not the right project is opened.
- The PLC's program has been changed from another station.
- The project has been compiled since the last download or online program change.

Important:
Please make sure the PLC you want to communicate with is connected, and the correct project is opened. If this is not the case, the running process can be endangered, serious malfunctioning can occur. E.g. the PLC stops with a system error.

Are you sure you want to continue?

Fig. 8-136: *Dialogue box to inform about inconsistency*

If you click on **Yes**, online change will be performed by comparing the newly compiled code with the old code that is uploaded from the PLC before – provided that the other usual checks preceding an online change are passed (see below). Depending on the circumstances (e.g. too big code parts to be changed), it is possible that online exchange will be applied – just like in the case of a consistent project. And in specific cases (e.g. when changing both SFC and Main programs with step deletion in the SFC program or when the connected PLC does not support online exchange), it will be possible that online change will be denied – just like in the case of a consistent project.

In the case that online change (or online exchange) is executed, the actual project and the project on the PLC will be consistent after that. Consecutive online changes can be completed normally, without the above described handling.

You can set if online change should be available when inconsistency is found. Please see section 2.8.2 for the related Options setting.

NOTES

In Backup mode only the control system has to be in RUN mode for Online Change. The standby system may be in either RUN or STOP mode. However, if the standby system is in FILE VERIFY ERROR (6000 or 6001), an error message will be displayed and Online Change will be aborted.

In backup mode Online Program Change will only be executed if the registered keyword (if any) is the same on both PLCs. If there's a mismatch, an error message will be displayed, and Online Change will be aborted.

In Backup mode Online Program Change will only be executed if the size of the additional program size area is the same on both PLCs. (For details on setting the additional program size please refer to section 2.8.2.) If there's a mismatch, an error message will be displayed, and Online Change will be aborted.

If in a redundant system the current additional program size area is exceeded during online program change, the number of necessary additional steps will be calculated. If either of the PLCs has not enough free space in Program Memory for the necessary additional steps, an error message will be displayed, and Online Change will be aborted.

Project – Online Program Change/Online Program Exchange

This function enables you to make changes to the program while the system is in RUN mode. The edited program is recompiled automatically and compared with the code in the PLC CPU. Only the altered part of the program code is downloaded to the CPU.

NOTE

Online changes are possible using the **Single Block Online Change (SBOC)** function or the **Multiple Block Online Change (MBOC)** function which is available for MELSEC Q/QnA series' CPUs (from release date 03.1999 on). The MBOC function is not available in the Q4ARCPU.

Limitations for FX CPUs

For the FX family, there are the following restrictions regarding the kind of changes you can apply on the program with **Online Program Change**:

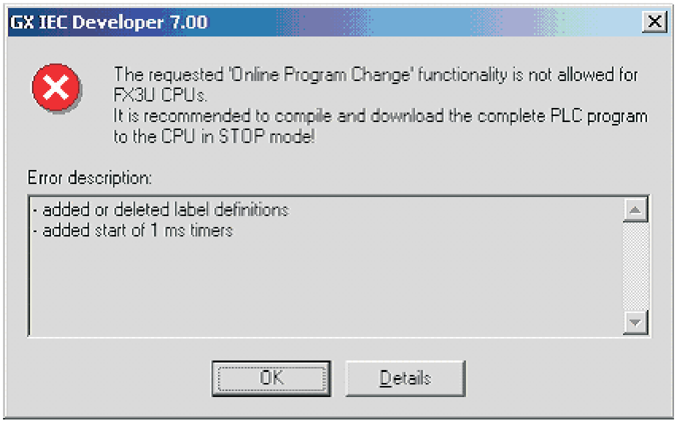
- No newly added, deleted or changed (replaced) pointer or interrupt labels (P and I) are allowed.
- No used pointer or interrupt label (P and I) may be shifted up or down due to a change in the size of the program code.
- No pointer or interrupt label (P and I) may be included in the changed part of a program.
- No newly added or deleted start or reset of high speed counters is allowed.
- No newly added start of 1 ms timers is allowed.
- No newly added or deleted instructions MNET(P), ANRD(P), ANWR(P), RMST, (D)RMWR(P), (D)RMRD(P), RMMN(P), BLK(P) or MCDE(P) are allowed.

Limitations for FX3U CPUs

For the FX3U there are fewer limitations than for other FX PLCs. The restrictions for FX3U are:

- No newly added, deleted or changed (replaced) pointer or interrupt labels (P and I) are allowed.
- No newly added or deleted start or reset of high speed counters is allowed.
- No newly added start of 1 ms timers is allowed.
- No newly added or deleted instructions DTBL, WBFM, RBFM or (D)SORT2 are allowed.

When online program change is not possible because one or more restricted changes have been made to the program, the reason(s) will be displayed in the following dialogue and online program change will be aborted.


GX IEC Developer 7.00

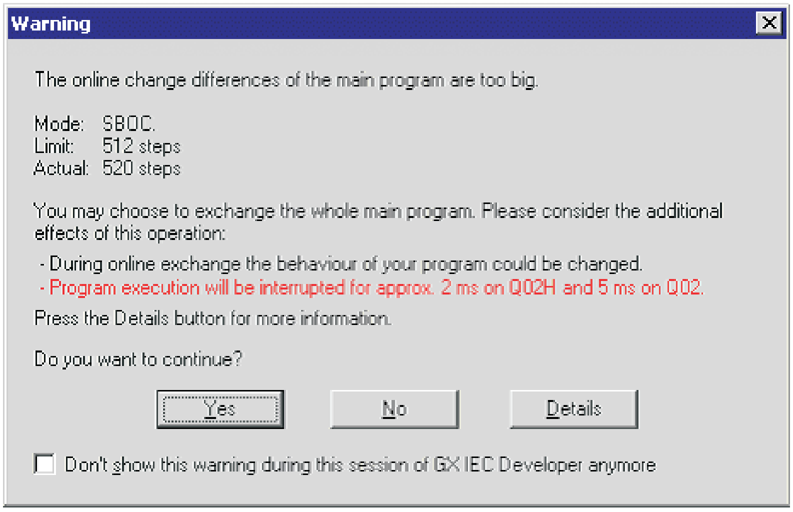
The requested 'Online Program Change' functionality is not allowed for FX3U CPUs.
It is recommended to compile and download the complete PLC program to the CPU in STOP mode!

Error description:

- added or deleted label definitions
- added start of 1 ms timers

Fig. 8-137:
Message window in case of a FX3U CPU

Online Program Exchange is an extension to **Online Program Change** and **Online Change Mode.** If the difference between the old and the new main/SFC program exceeds the step-limit of normal online change (both SBOC and MBOC if available), you will be offered to use **Online Program Exchange** to perform the change on the respective program type.
If you then chooses not to do so, online change will recover, as it does when the step-limit is exceeded. The **Online Program Change** will not be executed then.


Warning

The online change differences of the main program are too big.

Mode: SBOC.
Limit: 512 steps
Actual: 520 steps

You may choose to exchange the whole main program. Please consider the additional effects of this operation:

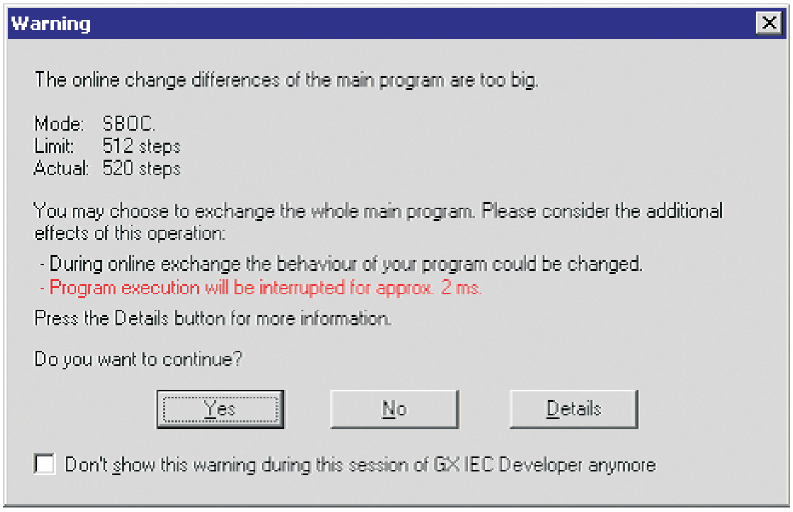
- During online exchange the behaviour of your program could be changed.
- Program execution will be interrupted for approx. 2 ms on Q02H and 5 ms on Q02.

Press the Details button for more information.

Do you want to continue?

□ Don't show this warning during this session of GX IEC Developer anymore

Fig. 8-138: *Dialogue box to inform about program exchange (for a QnH CPU other than Q02H)*


Warning

The online change differences of the main program are too big.

Mode: SBOC.
Limit: 512 steps
Actual: 520 steps

You may choose to exchange the whole main program. Please consider the additional effects of this operation:

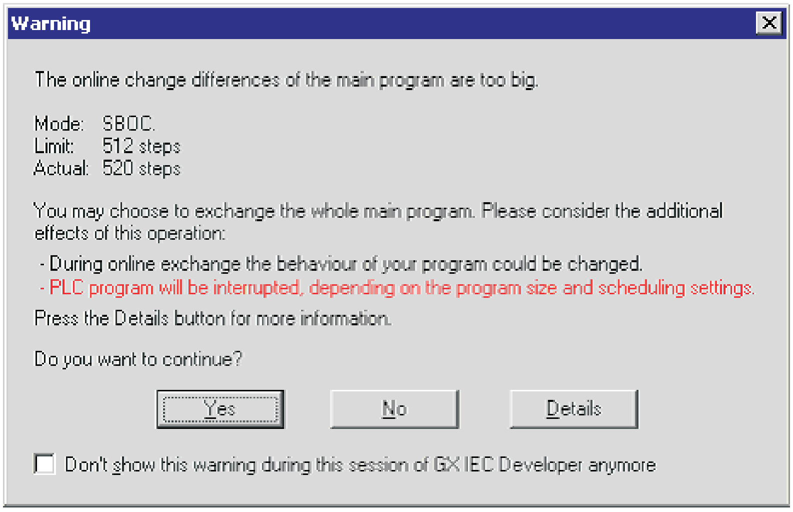
- During online exchange the behaviour of your program could be changed.
- Program execution will be interrupted for approx. 2 ms.

Press the Details button for more information.

Do you want to continue?

□ Don't show this warning during this session of GX IEC Developer anymore

Fig. 8-139: *Dialogue box to inform about program exchange (for a Q02H CPU)*



Warning

The online change differences of the main program are too big.

Mode: SBOC.
Limit: 512 steps
Actual: 520 steps

You may choose to exchange the whole main program. Please consider the additional effects of this operation:

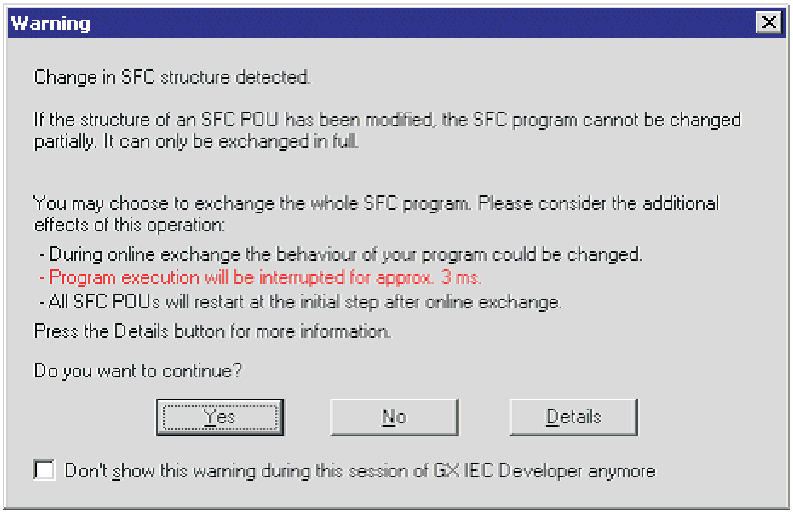
- During online exchange the behaviour of your program could be changed.
- PLC program will be interrupted, depending on the program size and scheduling settings.

Press the Details button for more information.

Do you want to continue?

□ Don't show this warning during this session of GX IEC Developer anymore

Fig. 8-140: *Dialogue box to inform about program exchange (for a Q25SS CPU)*



Warning

Change in SFC structure detected.

If the structure of an SFC POU has been modified, the SFC program cannot be changed partially. It can only be exchanged in full.

You may choose to exchange the whole SFC program. Please consider the additional effects of this operation:

- During online exchange the behaviour of your program could be changed.
- Program execution will be interrupted for approx. 3 ms.
- All SFC POUs will restart at the initial step after online exchange.

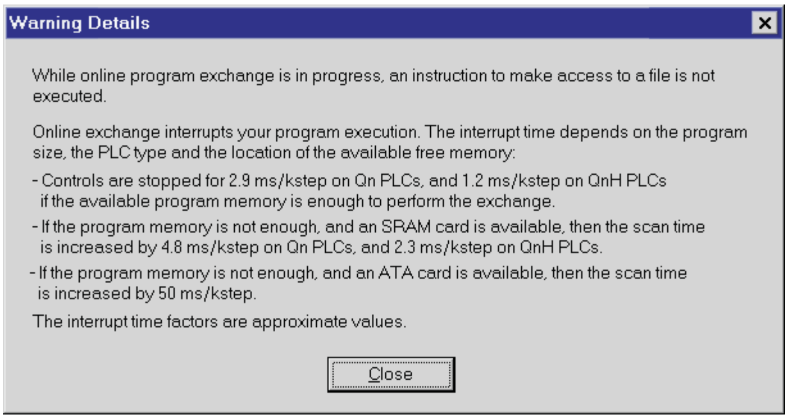
Press the Details button for more information.

Do you want to continue?

□ Don't show this warning during this session of GX IEC Developer anymore

Fig. 8-141: *Dialogue box to inform about program exchange (if the SFC structure has been changed)*

Pressing the **Details** button shows the following dialogue box:



Warning Details

While online program exchange is in progress, an instruction to make access to a file is not executed.

Online exchange interrupts your program execution. The interrupt time depends on the program size, the PLC type and the location of the available free memory:

- Controls are stopped for 2.9 ms/kstep on Qn PLCs, and 1.2 ms/kstep on QnH PLCs if the available program memory is enough to perform the exchange.
- If the program memory is not enough, and an SRAM card is available, then the scan time is increased by 4.8 ms/kstep on Qn PLCs, and 2.3 ms/kstep on QnH PLCs.
- If the program memory is not enough, and an ATA card is available, then the scan time is increased by 50 ms/kstep.

The interrupt time factors are approximate values.

Fig. 8-142: *Dialogue box **Warning Details***

PLCs of the MELSEC Q series support **Online Program Exchange**. This means that while the PLC is in RUN mode, the main program on drive 0 (program memory) of the PLC will completely be replaced with the one you are working on. The advantage of this functionality compared to the normal **Online Program Change** is that it allows a significantly larger number of steps to be changed at once than using the **Online Program Change**.
The advantage of the project download is that there is no need to stop the PLC.

Nevertheless it is recommended to use **Online Program Change** instead of **Online Program Exchange** if the applicable differences can be handled by **Online Program Change**, as it needs less free memory and time.

NOTE

Qn(H) CPUs from version B4 onward support **Online Program Change** of the SFC program. **Online Program Exchange** is supported as well. SFC online change with older CPU versions will work just like before: if normal **Online Program Change** cannot be performed, the online change process will be aborted.

The following table shows the availability of main program and SFC program online exchange on specific CPUs of the Q-series:

CPU	Support of main program exchange	Support of SFC program exchange
Q02(H), Q06H, Q12H, Q25H	✔	from version B4 on
Q12PH, Q25PH	✔	—
Q25SS	✔	—
Q00(J), Q01	—	—

Tab. 8-5: *Support of* ***Online Program Exchange***

So in the case of a SFC program when online change fails, online exchange can be performed. If the structure of at least one SFC POU was modified, online exchange will be offered – provided that the version of the CPU is detected to be at least B4.

When executing **Online Program Exchange** first the compiled main/SFC program is downloaded to the PLC's memory, then MAIN.QPG/SFC.QPG in the program memory is overwritten with the downloaded file. For this program exchange there must be enough free space for the new program code while the old MAIN.QPG/SFC.QPG is still in memory. This free space can be either in the program memory or in a SRAM or ATA memory card (provided that the size of the downloaded file is not greater than the size of the exchanged file and the free memory space in the program memory altogether).

If there is not enough free space on drive 0, an installed memory card is automatically used instead.

If both programs have to be exchanged, the main and the SFC program are exchanged separately, so there must be enough free space for the bigger program for a successful online exchange.

After **Online Program Exchange** a SFC program is not automatically restarted by the PLC (except for block 0, if the “Autostart block 0” option is set in the SFC Parameters).
Therefore, for Q Series PLCs, the code generated when SFC POUs are used in a project, the last SFC block (BL319) is generated as an SFC control block, which is used to restart all SFC blocks after an Online Exchange.
If this control block could not be generated (because the project already used SFC block 319), a warning will be displayed during SFC Online Exchange.

CAUTION
- Performing an Online Program Exchange affects the behaviour of the PLC.
- Controls are stopped during the program exchange interval and outputs are not changed, which may be dangerous in some situations.

NOTES

The scan time of the first cycle after finishing the program exchange will be increased by approximately
- 1.2 ms/kSteps on QnH CPU and
- 2.9 ms/kSteps on Qn CPU
if there is enough space in the program memory to perform the exchange.

If a SRAM card is needed for the operation, the scan time will be increased by approximately
- 2.3 ms/kSteps on QnH CPU and
- 4.8 ms/kSteps on Qn CPU.

If an ATA card is needed for the operation, the scan time will be increased by approximately 50 ms/kSteps.

While **Online Program Exchange** is in progress, an instruction to make access to a file is not executed.

The following conditions have to be fulfilled to start the **Online Program Exchange**:

- **Online Program Change** must fail because of exceeding the step-limit of both SBOC and MBOC (if available), or because of the SFC structure change.
- There must be enough space on drive 0 for the new program(s) (with the old program(s) considered as free space, since they will be replaced)
- There must be enough space on drive 0 or a memory card for the new program file (with the old program file still in memory). If both main and SFC programs get exchanged, there must be enough space for the bigger one of them.

Online Program Exchange is not available in the following situations:

- Too many changed labels and/or interrupt labels
- The CHK CHKST, CHKCIR or CHKEND instruction could not be changed, added or deleted.

Changes between the CHKST and CHK, or between the CHKCIR and CHKEND instructions are not allowed.

Online – Online Change Mode

Online Change Mode is different to the options for compiling the project or **Online Program Change**. In this mode, as soon as you make a change to a network a program verify is performed and the change is then written directly to the PLC. This ensures maximum program consistency and security.

After an online change has been made GX IEC Developer checks whether symbolic source information is selected in the transfer setup. If this is the case, you will be asked whether you want to download symbolic information. This dialogue can be disabled for the current GX IEC Developer session.

If you do not want to download any symbolic information or the option *no information* is selected in the transfer setup, symbolic information is deleted on the PLC.

NOTES

You should only ever make changes to ***one*** contiguous program range in the PLC at a time, and the range should not be too large. If you want to be absolutely sure that you are not modifying excessively large sections it is advisable to use ***Online Change Mode***, as this mode only permits you to make changes to one network at a time. After every change made to a network the system automatically performs a program compare and transfers the modification directly to the PLC. This approach ensures maximum program consistency and security.

In FX0 CPUs you can only change counters and timers when the PLC is in RUN mode.

Add new network

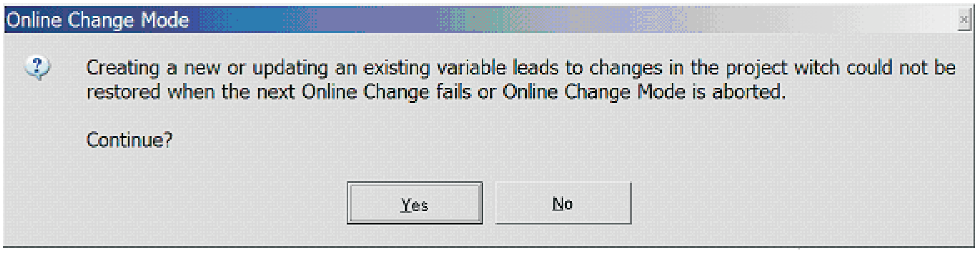
In the Online Change Mode new networks can be added to programs of the type IL, LD, FBD, and MIL in the editors. To add a new network, in the **Edit** menu select **New Network**. From the opening submenu select where to insert the new network: **Top**, **Before**, **After**, **Bottom**.

In the ST editor (Structured Text) online changes can be performed in the network.

Creation or Update of a Variable

In Online Change Mode the **Variable Selection Dialog** can be opened and a new variable can be created or an existing one can be changed. The according POU or a GVL is saved in the database immediately.

But these changes can not be undone when the Online Change Mode is left. Because of this a confirmation message is displayed when you create a new variable or update an existing variable in the POU header or a GVL in Online Change Mode.


Online Change Mode

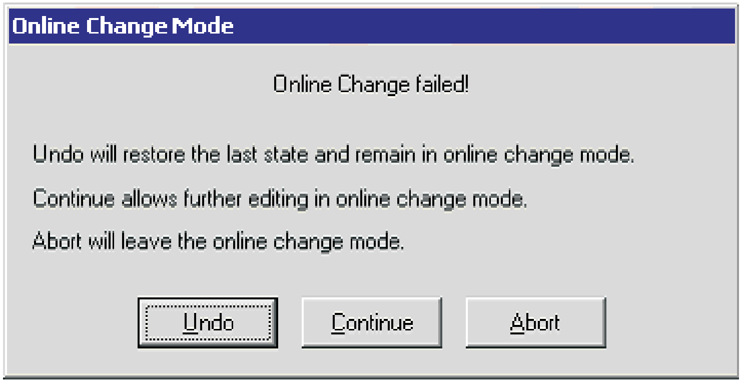
Creating a new or updating an existing variable leads to changes in the project witch could not be restored when the next Online Change fails or Online Change Mode is aborted.

Continue?

Fig. 8-143: *Dialogue box to warn you*

Recover Online Changes

If the online change fails or it is cancelled by the user, a message box appears giving three options. In case a variable was created or changed an Undo is not possible. The **Undo** button is grayed then.


Online Change Mode

Online Change failed!

Undo will restore the last state and remain in online change mode.

Continue allows further editing in online change mode.

Abort will leave the online change mode.

Fig. 8-144: *Online Change failed*

Undo

The changes recently made are rejected and the unmodified network saved at the beginning is recovered. GX IEC Developer remains in online change mode. Additional **Undo/Redo** in the menu **Edit** is not possible.

Continue

The changes recently made are kept and you have the possibility to correct the faulty network manually. Recent changes can be restored by means of the functions **Edit - Undo** and **Edit - Redo**.

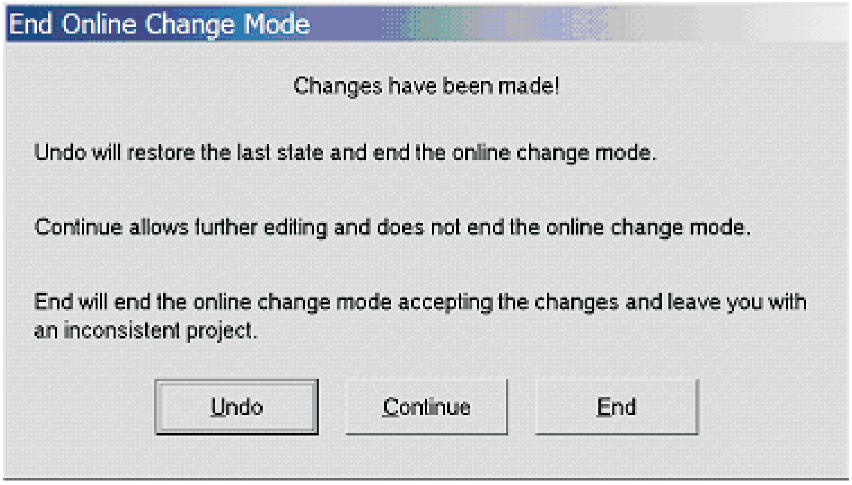
GX IEC Developer remains in online change mode.

Abort

The changes recently made are kept and the online change mode will be left. Recent changes can be restored by means of the functions **Edit - Undo** and **Edit - Redo.**

Via mouse clicks to the navigator area the online change mode will be left.

If you have made changes, you are asked whether you want to remain in the online change mode. Additionally you have the option to undo the changes which have already been done. In case a variable was created or changed an Undo is not possible. The **Undo** button is grayed then.


End Online Change Mode

Changes have been made!

Undo will restore the last state and end the online change mode.

Continue allows further editing and does not end the online change mode.

End will end the online change mode accepting the changes and leave you with an inconsistent project.

Fig. 8-145: *Changes notification*

Code Generation Options

In the **Extras - Options** menu the **Code Generation** command is available (refer to 2.8.1).

The file structure used in the CPUs of the MELSEC Q series makes it necessary to define an additional program range (in units of steps) for online changes when you are using these controllers.

8.8 Exporting

When you perform an import/export both the PLC program data and all the parameter settings are now imported and exported.

8.8.1 Exporting an GX IEC Developer project into an ASCII file

You can export and save the contents of your projects in ASCII format for archiving or to provide the information to others. The ASCII file contains all the project data and has the added advantage that it takes up significantly less storage space than the contents of a complete project directory.

How to export the entire project

① Select the command **Other** in the **Project** menu, then select **Export**.

A dialogue box is displayed.

② Enter the name for the ASCII file.

③ Click on the **OK** button.

The progress of the export process is documented in a status window.

How to export individual objects

① Select the objects in Project Navigator you wish to export.

② Select the command **Export** in the **Object** menu.

A dialogue box is displayed.

③ Enter the name for the ASCII file.

④ Click on the **OK** button.

The progress of the export process is documented in a status window.

8.8.2 Export to EPROM

This function stores an ASCII file in Motorola format which can then be used to program PLC EPROMS with a standard EPROM programmer. Every line in the ASCII file contains the corresponding hardware address and values. The **Export to EPROM** command is in the **Other** submenu of the **Project** menu.

8.8.3 CSV variable export

The menu item **Extras - Export Variables** exports all variables defined in the GVL and the POU header of a project in the CSV (Comma Separated Variables) format into a text file. This file can be opened in external tools, e.g. process visualisation or Microsoft Excel.

Before the function is available, the project must be compiled. Besides the variable declarations (including class, identifier, address, type, initial value, comment), the exported file contains general project information (e.g. project name and GX IEC Developer version number).

The options for the CSV variable export can be set under **Extras - Options - CSV-Export** (refer to section 2.7.3).

8.9 Importing

When you perform an import/export both the PLC program data and all the parameter settings are now imported and exported.

GX IEC Developer provides a number of data import functions, which are listed briefly below.

Importing Programs from Files

- Import a GX IEC Developer ASCII file
  (**Project - Other - Import**) (**Object - Import**)
- Import a GX IEC Developer ASCII file
  (**Tools - Import - GX Developer Network**)
- Import a MELSEC print file
  (**Tools - Import MEDOC Network**)

Importing Programs from the PLC CPU

- Symbolic Upload: Upload the project from the PLC CPU, together with all graphical information (**Project - Transfer - PLC to MEDOC (Symbolic)**)
- MELSEC Upload: Upload the project from the PLC CPU in a MELSEC network
  (**Project - Transfer - PLC to MEDOC (MELSEC)**)

8.9.1 Importing GX IEC Developer ASCII files

A project saved as an ASCII file can be imported selectively. If you do not create a new project before performing the import the selected objects are added to those in the current project. This applies for all POUs, the tasks, DUTs and the global variables. The import function only works on ASCII files saved with the GX IEC Developer Export function!

NOTES

You must first create an export file before you can perform an import.
Importing a large ASCII file may take a while without any indication.

The ASCII import functionality is restricted for the Q Remote I/O modules. Files exported from a non-Q Remote I/O project are not allowed to be imported into a Q Remote I/O project. And files exported from a Q Remote I/O project are not allowed to be imported into a non-Q Remote I/O project.

How to import the entire project

① Select the command **Other** in the **Project** menu, then select **Import**.

A dialogue box is displayed.

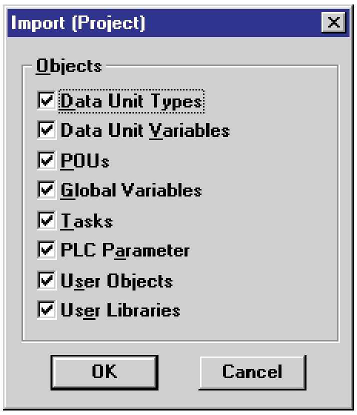
② Select the desired ASCII file.

③ Click on the **OK** button.

The progress of the import process is documented in a status window.

How to import individual objects

① Select the command **Import** in the **Object** menu.
A dialogue box is displayed in which you can select the objects that you wish to import into the new or existing project.



Import (Project)

Objects

☑ Data Unit Types
☑ Data Unit Variables
☑ POUs
☑ Global Variables
☑ Tasks
☑ PLC Parameter
☑ User Objects
☑ User Libraries

Fig. 8-146:
Import items selection

NOTE

In case of a Q Remote I/O module the Import functionality will import only the PLC Parameter, since Q Remote I/O projects have no other objects. Therefore in the object selection dialogue of the Import functionality all objects except the PLC Parameter will be disabled.

② Mark the check boxes of the objects you wish to import.

- Data Unit Types (DUTs)
- Data Units Variables
- POUs
- Global Variables
- Tasks
- PLC Parameter
- User Objects
- User Libraries

③ Click on the **OK** button..

A dialogue box is displayed.

④ Select the desired ASCII file.

⑤ Click on the **OK** button.

The progress of the import process is documented in a status window.

NOTES

When importing data from an ASCII export file that includes the Global Variable List (e.g. a complete project export), external variables will be automatically removed from POU headers (including POUs in user libraries).

When importing data from an ASCII export file that has no Global Variable List in it, each imported POU header will be checked for external variables. If one or more variables with the class VAR_EXTERNAL or VAR_EXTERNAL_CONSTANT are found, you will be warned that these variables will be skipped.

8.9.2 Importing GX Developer ASCII files

GX Developer (former GPP/WIN) ASCII files can be imported. The ASCII file must have been exported from GX Developer by the separate software package GX Converter.

Open any editor of the type IL or MELSEC IL and select a network. In the **Tools** menu select the item **Import GX Developer Network**. Select the required file from GX Developer (text or CSV) to be imported into the selected network.

NOTE

Functions not supported by GX IEC Developer are ignored during the import (e. g. device comments, initial values, program execution types).
SFC programs can not be imported.

The data to be imported must comply with the following requirements:

Common

The maximum length of an import line is limited to 512 characters.

The width of the different data fields is limited to the following values:

● Step number:	7
● Line statement:	64
● Instruction:	15
● P/I-Statement:	64
● Note:	32
● Device:	15

Text files:

- Text files have to start with a step number followed by a blank (‘ ‘); otherwise the text file is not recognized as a text file
- Line statements have to start with a step number, followed by one or more blanks, followed by a ‘;’ and followed directly with the line statement.
  Sample:
  7 ;I am a line statement
- Notes may not start with a step number and can be located wherever in the import line; they have to be enclosed by ‘<’ and ‘>’
  Sample:
  <I am a Note>
- P/I-statements may not start with a step number and can be located wherever in the import line; they have to be enclosed by ‘[’ and ‘]’. Separate Statements must start with a ‘*’ before the ‘[‘.
  Sample:
  *[I am a separate Statement]
  [I am an embedded Statement]
- If a line only contains device data and no statement, the device information must start in column 25, columns 9 to 24 must be blank (‘ ‘)

CSV-Files:

- CSV-files have to start with a step number followed by a comma (‘,’), otherwise CSV-Files are not recognised as CSV-Files
- Step numbers must be located in column 1
- Line statements must be located in column 2
- Instructions must be located in column 3
- Devices must be located in column 4
- Column 5 must be empty
- P/I statements must be located in column 6
- Notes must be located in column 7
- Columns 8 and 9 must be empty
- Each line must contain exactly 8 separators
- Notes, line statements and P/I statements, which contain one or more ‘,’ must be enclosed with quotes “”’
- Quotes within notes, line statements and P/I statements must be masked with another quote; the whole statements must be enclosed with quotes
- Separate P/I-statements must start with ‘*’

8.9.3 Importing MELSEC MEDOC print files

Print files generated by MELSEC MEDOC (DOS version) can be loaded into IEC or MELSEC Instruction Lists as MELSEC networks. The POU into which you wish to load the file must be open and the network must be selected. Please note that performing this import procedure automatically overwrites the contents of the selected network in the body!

How to import MELSEC MEDOC print files

In MELSEC MEDOC (DOS version):

① Select a file name as the printer port. The extension TMP is added to the file name automatically.

② Select only the Instruction List and Name List in the program listing. Make sure that the header is switched off!

③ Start the print procedure.

In GX IEC Developer:

④ Open the body of a MELSEC Instruction List or create a new MELSEC Instruction List POU. This new POU must be a Program (PRG).

⑤ Select **New Network** in the **Edit** menu, then select **MELSEC Before** from the submenu displayed.

⑥ Select **Import MELSEC Network** in the **Tools** menu.

⑦ In the dialogue box displayed specify the drive and directory where the import file is stored and select the import file (`TMP`). Confirm your selection with **OK**. An option box is opened.
GX IEC Developer checks whether the specified file is a valid import file, and whether it contains program code and symbolic names. The default settings displayed in the option box depend on the results of this check. For example, if no symbolic names are found this option will be displayed in light grey and is not selectable.

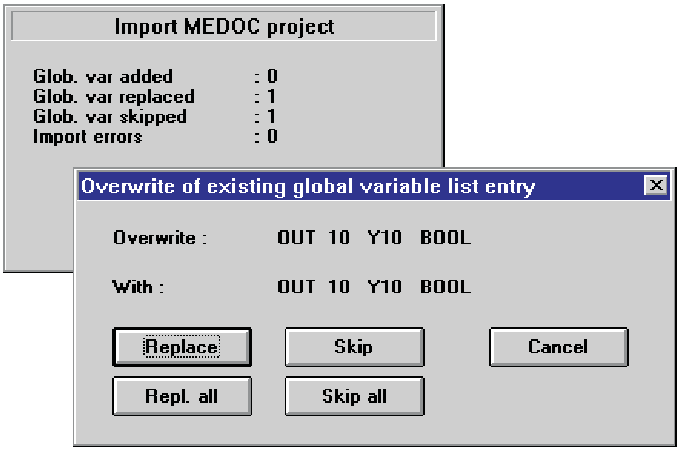
⑧ Confirm the selections with **OK**.

The progress of the import process is documented in a status window.

MELSEC Program: Import only the program code lines of the Instruction List.

MELSEC Symbolic Names: Import only the Name List. The entries from the list are inserted both in the global variables declaration table and in the header of the POU. Comments after relays and data registers are converted into symbolic names and imported as comments. When this conversion is performed spaces, quotes and periods are either deleted or replaced with underline characters in accordance with the IEC standard.

The translation of the symbolic names can result in duplicate names for some variables. If this happens the following dialogue box is displayed:



Import MEDOC project

Glob. var added : 0
Glob. var replaced : 1
Glob. var skipped : 1
Import errors : 0

Overwrite of existing global variable list entry

Overwrite : OUT 10 Y10 BOOL

With : OUT 10 Y10 BOOL

Fig. 8-147:
Duplicate variable names notification

The first two entries show the name of the existing variable and the name that the program proposes to use to overwrite this variable.

Replace: Replace the old variable name with the new one.

Repl. all: Replace all existing variable names with new names. If you select this button the program will replace all subsequent variables without prompting.

Skip: Skip this variable name without replacing it. The name is also not added to the Variable List.

Skip all: Skip all variables with duplicate names and do not add any of them to the Variable List. If you select this button the program will skip all subsequent duplicate variables without prompting.

8.9.4 Upload PLC to MEDOC

The upload function reads the binary code from the PLC CPU and creates a POU in MELSEC IL language with the standard name PRG_MAIN and a task with the standard name MELSEC_MAIN. In addition to the binary code all the graphical and structural information from the PLC CPU can be uploaded.

How to upload project code from the CPU

① In the **Project** menu select **Transfer.** Then select **Upload from PLC**.

② Select the PLC type and confirm with **OK**.

③ The upload must be stored in a new project. Enter the name of the new project to be created in the dialogue box displayed.

④ Confirm your entry with **OK**.

⑤ A dialogue box is displayed showing a selection of ports for the transfer procedure:

- CPU Port
- Computer Link
- Ethernet

⑥ Select the appropriate port for your configuration and confirm with **OK**.

⑦ A dialogue box is displayed showing a selection of UPLOAD modes:

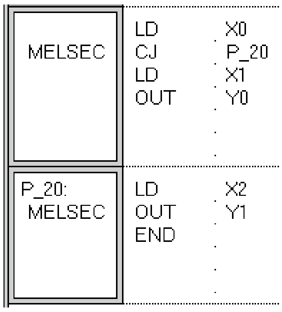
- MELSEC IL
- Source Information

⑧ Select the required UPLOAD mode and confirm with **OK**.
When you select symbolic source information to be uploaded from the PLC, GX IEC Developer will check whether symbolic information is available on the PLC.

The progress of the import procedure is documented in a status window.

Absolute Label

Absolute labels are used as jump destinations when PLC programs are uploaded in GX IEC Developer. Absolute labels are specific pointer addresses (e.g. P20). Every program section identified with a pointer in the original PLC program is loaded to a separate MELSEC network.



```
MELSEC        LD     X0
              CJ     P_20
              LD     X1
              OUT    Y0

P_20:         LD     X2
MELSEC        OUT    Y1
              END
```

Fig. 8-148:
The absolute label is P_20

Absolute labels can also be assigned by the user, but you must be very careful to use the correct syntax. Only addresses with the format P – underline – address are interpreted as absolute labels.

P_20	Reference to pointer P20 (absolute label)
P20	Reference to a free pointer P** (not an absolute label)

NOTE The labels with the format LAB_20 generated when you import MELSEC MEDOC print files are not absolute labels!

9 Diagnostics Functions and Online Operation

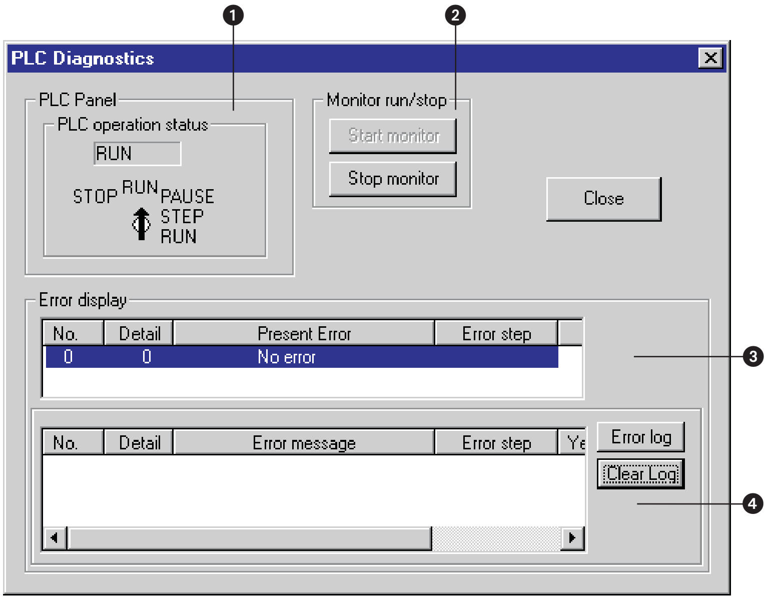
9.1 Debugging

The functions in the **Debug** menu allow to perform precise troubleshooting and error analysis on your progress.

9.1.1 PLC diagnostics

A CPU diagnostics

The status and the error codes of the CPU can be displayed. In the **Debug** menu select **PLC Diagnostics**. The PLC will be checked and the dialogue below is opened.



PLC operation status: RUN

No.	Detail	Present Error	Error step
0	0	No error	

No.	Detail	Error message	Error step	Ye

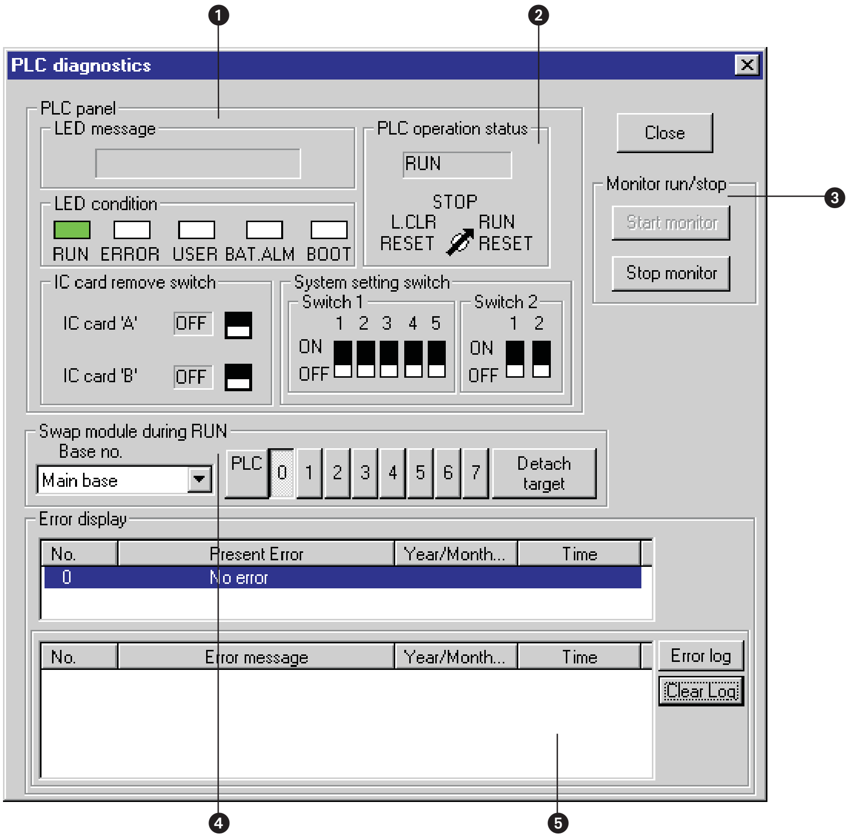
Fig. 9-1: A CPU diagnostics

Item	Description
❶	PLC Panel Shows the operation status of the target PLC.
❷	Monitor The **Start monitor** button starts the communications with the PLC and the error display is updated. The **Stop monitor** button stops the communication with the PLC and the error display. Clicking on **Error log** or **Clear Log** automatically stops monitoring.
❸	Error display (Present Error) This section displays errors currently occurring in the connected CPU.
❹	Error display (Error message) This section lists the detected errors. Click on the **Error log** button to display recent error messages from an error log. The **Clear Log** button clears this error log.

Tab. 9-1: A CPU diagnostics items

QnA CPU diagnostics

The status and the error codes of the CPU can be displayed. In the **Debug** menu select **PLC Diagnostics**. The PLC will be checked and the dialogue below is opened.



No.	Present Error	Year/Month...	Time
0	No error		

No.	Error message	Year/Month...	Time

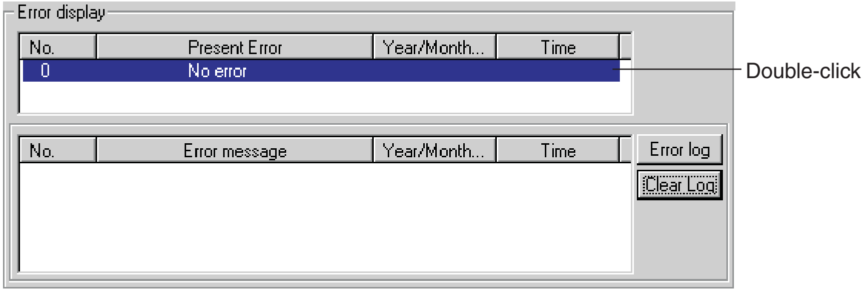
Fig. 9-2: QnA CPU diagnostics

Item	Description
❶	PLC panel Shows the PLC panel status of the target PLC.
❷	PLC operation status Shows the PLC operation status of the target PLC.
❸	Monitor The **Start monitor** button starts the communications with the PLC and the error display is updated. The **Stop monitor** button stops the communication with the PLC and the error display. Clicking on **Error log** or **Clear Log** automatically stops monitoring.
❹	Swap module during run Select the base number and the slot number for the module to be swapped and click on **Detach target**.
❺	Error display Click on the **Present Error** button to display errors currently occurring in the connected CPU. Click on the **Error log** button to display recent error messages from an error log. The **Clear Log** button clears this error log.

Tab. 9-2: QnA CPU diagnostics items

NOTE

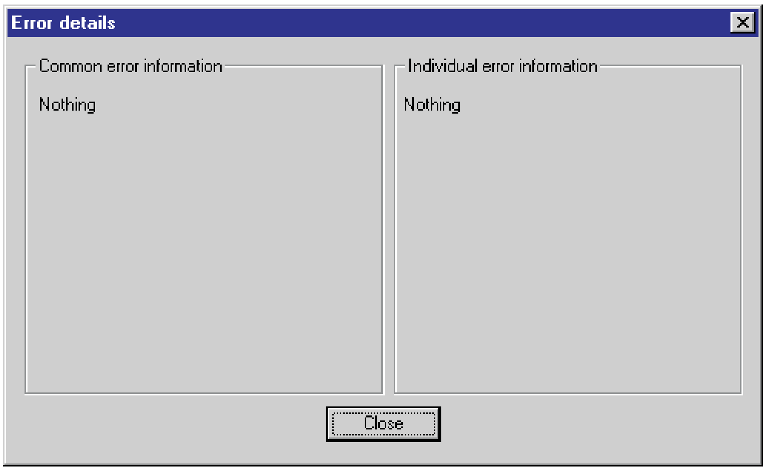
Double-click on a present/current error shown in the list to open the **Error details** dialogue. The Error details dialogue shows the content of SD5 – SD15 (common error information) and SD16 – SD26 (individual error information).



No.	Present Error	Year/Month...	Time
0	No error		

No.	Error message	Year/Month...	Time

Fig. 9-3: *Double-click on error opens error details*



Common error information

Nothing

Individual error information

Nothing

Fig. 9-4: *Error details*

Q CPU/Q Remote I/O Module diagnostics

The status and the error codes of the CPU can be displayed. In the **Debug** menu select **PLC Diagnostics**. The PLC will be checked and the dialogue below is opened.

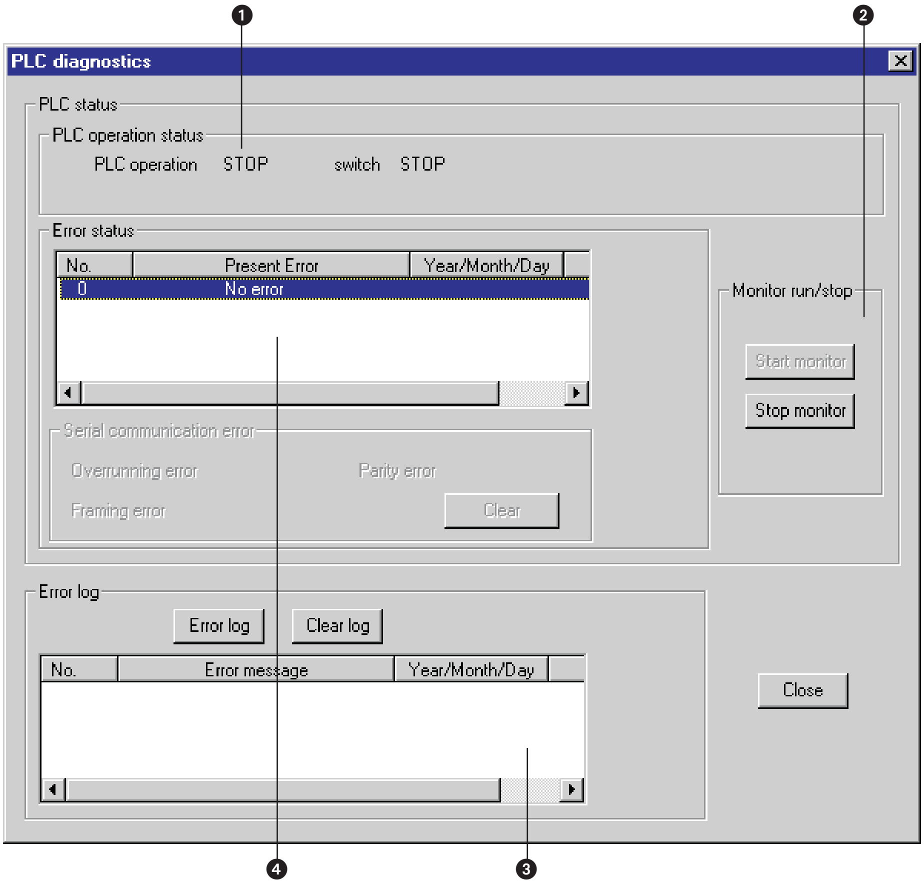
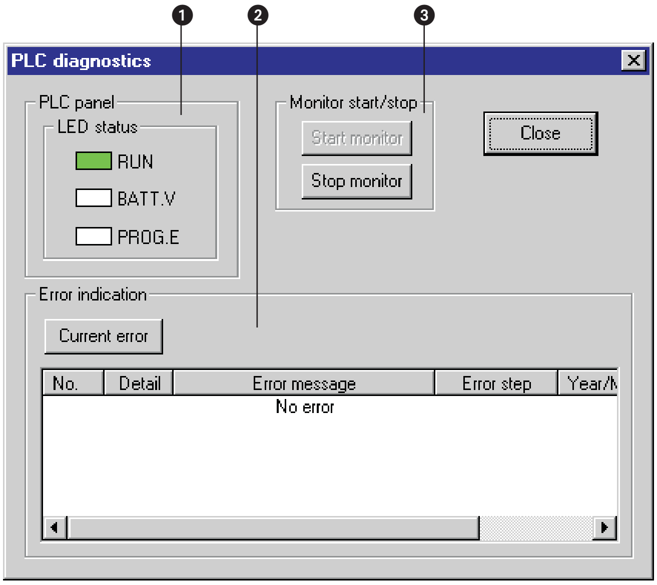


Fig. 9-5: Q CPU diagnostics

Item	Description
❶	PLC operation status Shows the PLC operation status of the target PLC.
❷	Monitor run/stop The **Start monitor** button starts the communications with the PLC and the error display is updated. The **Stop monitor** button stops the communication with the PLC and the error display. Clicking on **Error log** or **Clear Log** automatically stops monitoring.
❸	Error log This section lists the detected errors.
❹	Error status This section shows whether the displayed error is a present error or an error from the error log. Click on the **Present Error** button to display errors currently occurring in the connected CPU. Click on the **Error log** button to display recent error messages from an error log. The **Clear Log** button clears this error log.

Tab. 9-3: Q CPU diagnostics items

FX diagnostics



No.	Detail	Error message	Error step	Year/M
		No error		

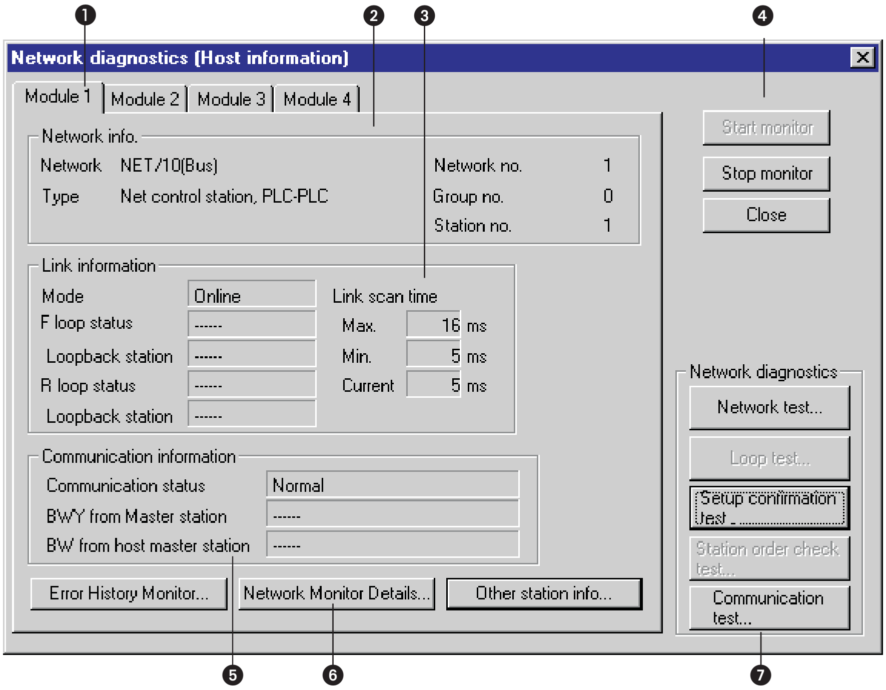
Fig. 9-6: *FX diagnostics*

Item	Description
❶	PLC panel Shows the PLC panel status of the target PLC.
❷	Error indication Click on the **Current Error** button to display errors currently occurring in the connected CPU. The date and time values refer to the moment when the **Current Error** button was activated but not when the error occurred. The currently occurring errors are listed including the error code and error message.
❸	Monitor start/stop The **Start monitor** button starts the communications with the PLC and the error display is updated. The **Stop monitor** button stops the communication with the PLC and the error display.

Tab. 9-4: *FX diagnostics items*

9.1.2 Network Diagnostics

From a selectable target host station the status of a MELSECNET (II) or MELSECNET/10 network can be checked.



Network info.
Network NET/10(Bus)
Type Net control station, PLC-PLC
Network no. 1
Group no. 0
Station no. 1

Link information
Mode: Online
F loop status: ------
Loopback station: ------
R loop status: ------
Loopback station: ------
Link scan time
Max. 16 ms
Min. 5 ms
Current 5 ms

Communication information
Communication status: Normal
BWY from Master station: ------
BW from host master station: ------

Fig. 9-7: Network diagnostics

Item	Description
❶	Tabs for link modules Select the tab for the link module for displaying the diagnostics results (AnN or AnA CPUs only support 2 modules).
❷	Network information Shows information on the network the selected target host station belongs to (for MELSECNET (II) the network number and group address are not displayed).
❸	Link information Shows the network status.
❹	Monitoring The **Start monitor** button starts monitoring. In RUN mode the display will be updated. The **Stop monitor** button stops monitoring.
❺	Communication information Shows the communications status of the selected network (for MELSECNET/10 only the communications status is displayed).
❻	Error History Monitor (see below) Network Monitor Details (see below) Other station information (see below)
❼	Network diagnostics (see below) Network test Setup confirmation test Station order check test Communication test

Tab. 9-5: Network diagnostics items

NOTES

In a MELSECNET (II) network at maximum 2 link modules can be installed.
In a MELSECNET/10 network at maximum 4 link modules can be installed.

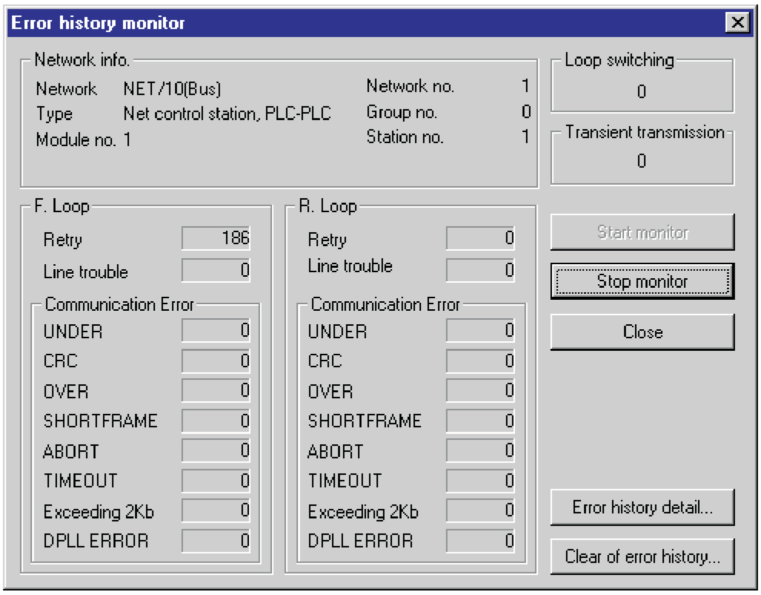
The network diagnostics check the network the connected host station belongs to. To change the network simply change the connection to the host station and select the connection to another host station.

For the connection to remote I/O modules the network diagnostics are not supported.

If the connection is established via a E71 unit the monitoring of an AnU CPU is performed like for an AnA CPU. MELSECNET (II) information can be monitored but no MELSECNET/10 information.

Error history Monitor

This button in the **Network diagnostics** dialogue opens a dialogue that monitors and lists communication errors in an error history. Communication errors are listed separately for the forward and the reverse loop. Error history details can be monitored and the error history can be cleared.



Error history monitor

Network info.

Network	NET/10(Bus)	Network no.	1
Type	Net control station, PLC-PLC	Group no.	0
Module no. 1		Station no.	1

Loop switching: 0

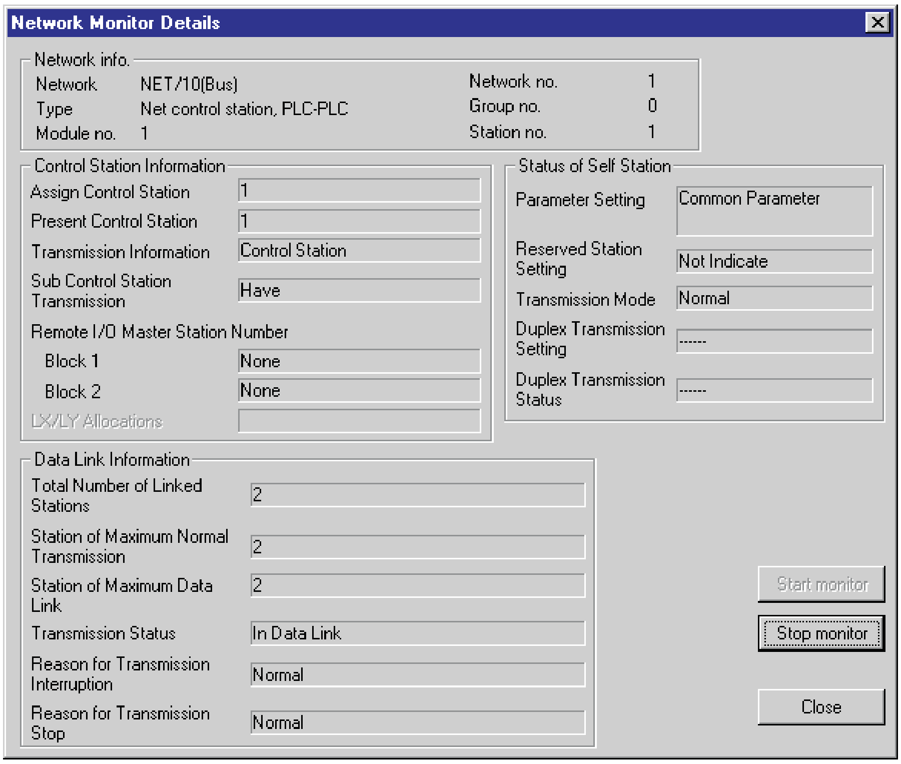
Transient transmission: 0

	F. Loop	R. Loop
Retry	186	0
Line trouble	0	0
Communication Error		
UNDER	0	0
CRC	0	0
OVER	0	0
SHORTFRAME	0	0
ABORT	0	0
TIMEOUT	0	0
Exceeding 2Kb	0	0
DPLL ERROR	0	0

Fig. 9-8:
Error history monitor

Network Monitor Details

This button in the **Network diagnostics** dialogue opens a dialogue that displays network monitor details.



Network Monitor Details

Network info.

Network	NET/10(Bus)	Network no.	1
Type	Net control station, PLC-PLC	Group no.	0
Module no.	1	Station no.	1

Control Station Information

Assign Control Station	1
Present Control Station	1
Transmission Information	Control Station
Sub Control Station Transmission	Have
Remote I/O Master Station Number	
Block 1	None
Block 2	None
LX/LY Allocations	

Status of Self Station

Parameter Setting	Common Parameter
Reserved Station Setting	Not Indicate
Transmission Mode	Normal
Duplex Transmission Setting	------
Duplex Transmission Status	------

Data Link Information

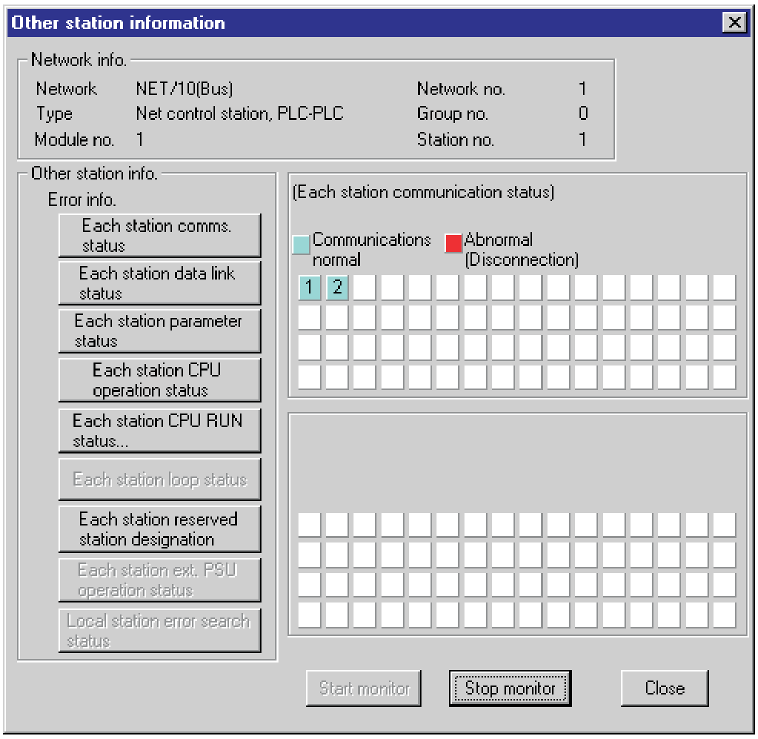
Total Number of Linked Stations	2
Station of Maximum Normal Transmission	2
Station of Maximum Data Link	2
Transmission Status	In Data Link
Reason for Transmission Interruption	Normal
Reason for Transmission Stop	Normal

Fig. 9-9: Network Monitor Details

Other station information

This button in the **Network diagnostics** dialogue opens a dialogue that monitors other stations connected to the MELSECNET (II) or 10.

- **Network information**: Shows information on the network of the selected target host station.
- **Other station information/Error information**: If an error in a station, a stopped station, a reserved station, or a station with external power supply the error information field reads *ERR*. Click on the respective button to select an object for the error information to show detailed error information.
- **Display field for detailed error information**: Shows the status of the relevant station for the selected item.
- Start monitor: Starts monitoring.
- Stop monitor: Stops monitoring.



Other station information

Network info.

Network	NET/10(Bus)	Network no.	1
Type	Net control station, PLC-PLC	Group no.	0
Module no.	1	Station no.	1

Other station info.

(Each station communication status)

🟦 Communications normal 🟥 Abnormal (Disconnection)

Error info.
Each station comms. status
Each station data link status
Each station parameter status
Each station CPU operation status
Each station CPU RUN status...
Each station loop status
Each station reserved station designation
Each station ext. PSU operation status
Local station error search status
1
2
Start monitor
Stop monitor
Close

Fig. 9-10:
Other station information

The following information on other stations is available for the A series:

Information on each Station		MELSECNET /10						MELSECNET (II)	
		MELSECNET /10				MELSECNET (II)		—	
		Control Station		Normal Station		Master Station		Master	Local
		Loop	Branch	Loop	Branch	Loop	Branch	—	—
1	Communications Status	●	●	●	●	●	●	●	—
2	Data Link Status	●	●	●	●	●	●	—	—
3	Parameter Status	●	●	(—)	(—)	●	●	●	—
4	CPU Operation Status	●	●	●	●	—	—	—	—
5	CPU RUN Status	●	●	●	●	—	—	●	●
6	Loop Status	●	—	●	—	●	—	●	—
7	Res. Station Designation	●	●	●	●	●	●	—	—
8	External Power Supply	—	—	—	—	—	—	—	—
9	Error Search Status	—	—	—	—	—	—	●	—

Tab. 9-6: *Information on other stations, A series*

● = available, – = not available

The following information on other stations is available for the Q/QnA:

Information on each Station		MELSECNET /10						MELSECNET (II)	
		MELSECNET /10				MELSECNET (II)		—	
		Control Station		Normal Station		Master Station		Master	Local
		Loop	Branch	Loop	Branch	Loop	Branch	—	—
1	Communications Status	●	●	●	●	●	●	●	—
2	Data Link Status	●	●	●	●	●	●	—	—
3	Parameter Status	●	●	(—)	(—)	●	●	●	—
4	CPU Operation Status	●	●	●	●	—	—	—	—
5	CPU RUN Status	●	●	●	●	—	—	●	●
6	Loop Status	●	—	●	—	●	—	●	—
7	Res, Station Designation	●	●	●	●	●	●	—	—
8	External Power Supply	●	(—)	●	(—)	●	(—)	—	—
9	Error Search Status	—	—	—	—	—	—	●	—

Tab. 9-7: *Information on other stations, Q/QnA*

● = available, – = not available

Network diagnostics/Network test

This button in the **Network diagnostics** dialogue opens a dialogue where you can start or stop the link operation for the MELSECNET/10 host station, selected stations, or all stations.

The following items are available:

- **Network information**: Shows information on the network of the selected target host station.
- **All stations operating status**: Displays the operating status of the tested modules.
- **Link start/stop**:
  Link start: Stations that were stopped by the host station can be started. Stations that were stopped by another station can not be started.

  Link stop: Stations can be stopped by the host station or another station.

  Force link start: Stations can be started regardless of whether they have been stopped by a host station or another station.

  However, the start can not be forced, if all stations were stopped. A start can not be forced neither, if for the link target another station is the target.

  By monitoring SB and SW can be determined whether a forced start of a peripheral device can be performed. By monitoring SW0000 – SW0004 can be determined whether the execution was started or stopped by a host station.

  A start can be forced, if the connection is established via C24, UC24, or QC24.

  Via a MELSECNET (II) or 10 interface board the start can not be forced.

  Via E71/QE71 the start can not be forced.

Network diagnostics/Setup confirmation test

This button in the **Network diagnostics** dialogue opens a dialogue where you can check the status of the station number, network address, and group address.

The following items are available:

- **Network information**: Shows information on the network of the selected target host station.
- **Setting check test:**

  Parameter settings: All stations that are setup in the parameters except the reserved station are tested (If no network settings are setup, all stations are tested).

  All stations: All stations are tested. No specified station is addressed.

  Target module: The module by default selected for the for the line monitor is specified. If no MELSECNET/10 module is connected, an error message is returned.
- **Execution results:**

  Station number: Shows the station number up to the specified station (64 station numbers max.).

  Loopback station: If the same loopback station was assigned to more than one station a "*" is displayed.

  Network number: If the same network number was assigned to more than one station, "*" is displayed.

  Group number: Shows the group addresses of the stations for which the setup confirmation test is performed. For a remote I/O network faulty stations are indicated by a blank.

  Reserved station: For stations specified as reserved stations "*" is displayed (if parameter confirmation was selected as test object).

  Error: If a station is reserved in the parameter settings or if a module in all selected stations is faulty "*" is displayed.

  Error network type (Q series only): Stations for which the parameter settings and current link do not match are indicated as "*".

  Double multiplex remote master station (Q series only): Stations with more than one sub master station within the same network are indicated by "*".

  Double parallel remote master stationXXX (Q series only): Stations with more than one sub master station within the same network are indicated by "*".

NOTE

The tests can not be performed for more than one station at one time.
During the test execution the scan execution is stopped.

Network diagnostics/Station order check test

This button in the **Network diagnostics** dialogue opens a dialogue where you can check the station order in the forward and reverse loops of a MELSECNET/10.

- **Network information**: Shows the information on the network of the selected target host station.
  The loop status is displayed for forward loops, reverse loops, forward/reverse loops, or loopback. The displayed total number of stations includes also the reserved-by station.
- **Loop test**: All stations specified in the parameter settings except the reserved stations are tested (If no network settings were specified, all stations are tested).

  All stations: All stations are tested. No specific station is addressed.

  Target module: The module by default selected for the for the connection monitor is specified. If no MELSECNET/10 module is connected, an error message is returned.
- **Execution results**: The station order is displayed in forward loop and reverse loop direction beginning from the host station.
  For loopback only the station order in forward loop direction is displayed.

NOTE

The tests can not be performed for more than one station at one time.
During the test execution the scan execution is stopped.

Network diagnostics/Communication test

This button in the **Network diagnostics** dialogue opens a dialogue that displays general communication test results.

9.1.3 Ethernet Diagnostics

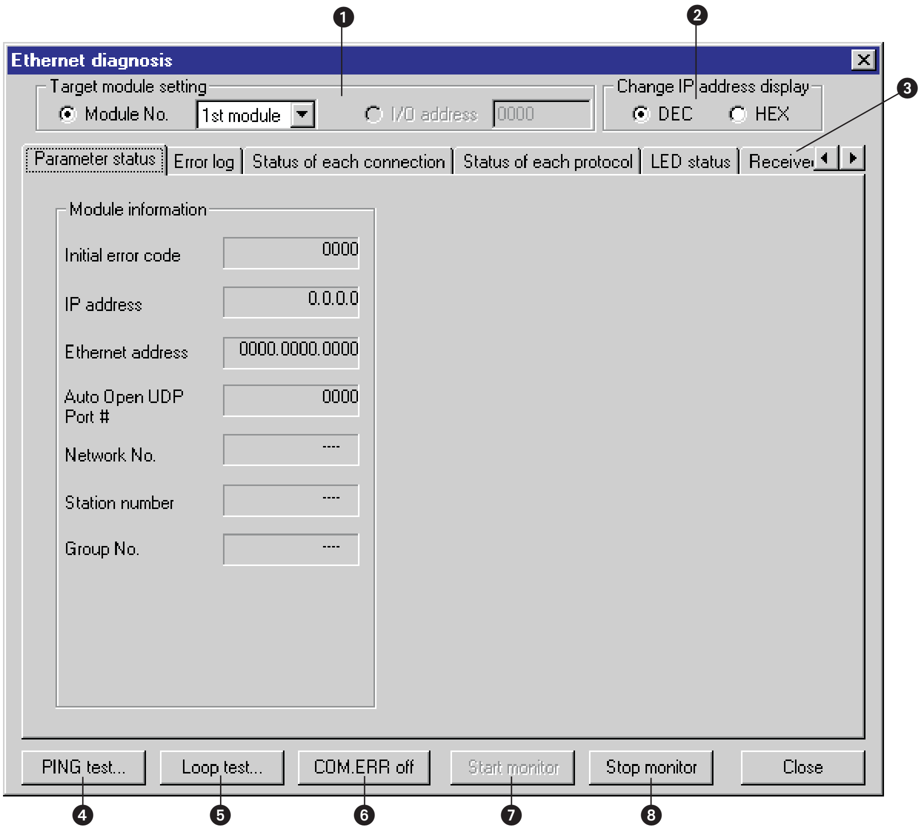
The Ethernet diagnostics test several settings of the connected Ethernet modules.

For details on error messages and the buffer memory refer to the manuals of the corresponding modules:

- Ethernet modules for the Q series (User's manual)
- Ethernet modules for the QnA series (User's manual)
- Ethernet modules AJ71E71-S3, A1SJ71E71-B2-S3, and A1SJ71E71-B5-S3 (User's Manual)

You can check the status of the following items:

- Parameters
- Errors
- Individual connections
- Individual protocols
- LEDs
- Received and sent e-mail information



Target module setting: ● Module No. 1st module ○ I/O address 0000

Change IP address display: ● DEC ○ HEX

Module information

Initial error code	0000
IP address	0.0.0.0
Ethernet address	0000.0000.0000
Auto Open UDP Port #	0000
Network No.	----
Station number	----
Group No.	----

Fig. 9-11: *Ethernet diagnostics*

Item	Description
❶	**Target module setting** Here you select the Ethernet module to be checked. The Q series modules No. 1 – 4 can be selected. A or QnA series modules are selected by their I/O address.
❷	**Change IP address display** Here the display of the IP address can be toggled between decimal or hexadecimal format.
❸	**Tabs for Ethernet information** Here different information types for the Ethernet modules are displayed.
❹	**PING test** Tests an Ethernet module that has completed its initialization or checks for any existing IP address. The PING test can either be executed via an Ethernet board or via a direct connection to the PLC CPU.
❺	**Loop test** This function is intended for Ethernet modules that are compatible with the Q series function version B or higher. A loop instruction tests whether all modules under a specified network number or station number have completed their initialization. If this test is executed for Ethernet modules that are compatible with the Q series function version A the response can not be evaluated and is processed as not responding. The loop test can either be executed via an Ethernet board or via a direct connection to the PLC CPU.
❻	**COM. ERR. off** Turns off the communications error LED.
❼	**Start monitor** Starts the Ethernet diagnostics.

Tab. 9-8: *Ethernet diagnostics items*

9.1.4 CC-Link Diagnostics

The CC-Link diagnostics monitor the network information of each station and test the network status.

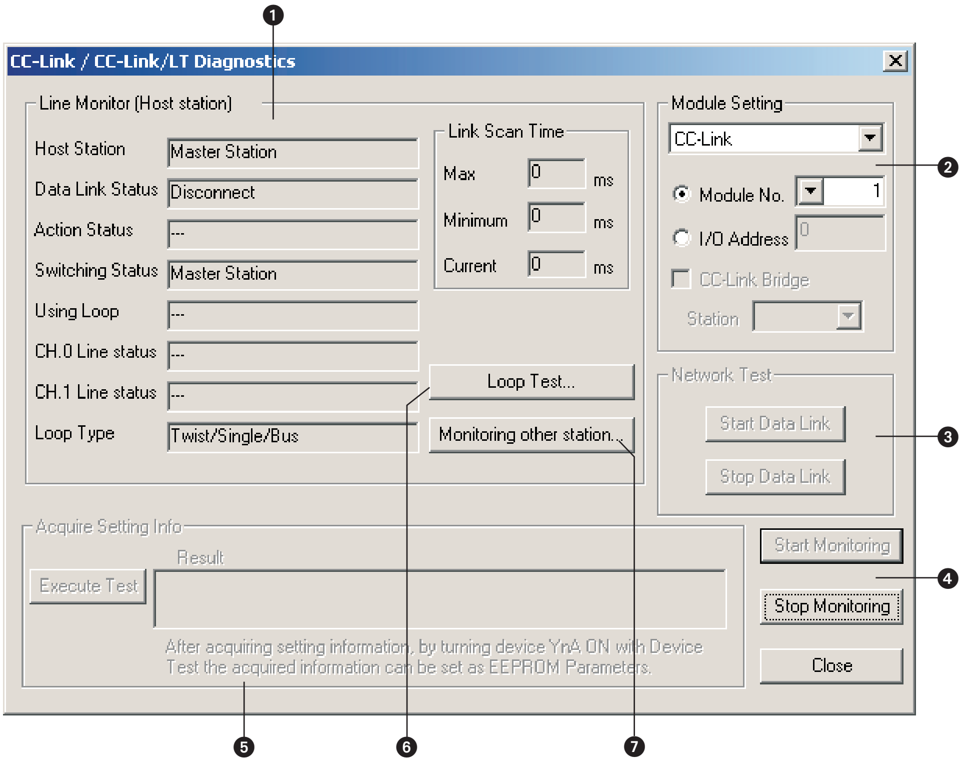
The CC-Link diagnostics can be started via the PLC CPU independently from the CC-Link station running as master, stand-by master, or local station.

NOTE

For the diagnostics via a Q CPU in Q mode the module QJ61BT11 is required.
For the diagnostics via a Q CPU in A mode the module AJ61BT11 or A1SJ61BT11 is required. If a program for the A series includes refresh parameters, the network test is disabled.

Monitoring a CC-Link line (host station)

The CC-Link diagnostics monitor the CC-Link line connected to the PLC CPU via the host station.



CC-Link / CC-Link/LT Diagnostics

Line Monitor (Host station)

Host Station: Master Station
Data Link Status: Disconnect
Action Status: ---
Switching Status: Master Station
Using Loop: ---
CH.0 Line status: ---
CH.1 Line status: ---
Loop Type: Twist/Single/Bus

Link Scan Time
Max: 0 ms
Minimum: 0 ms
Current: 0 ms

Module Setting: CC-Link
● Module No. 1
○ I/O Address 0
□ CC-Link Bridge
Station

Acquire Setting Info
Result:
After acquiring setting information, by turning device YnA ON with Device Test the acquired information can be set as EEPROM Parameters.

Fig. 9-12: CC-Link Diagnostics

Item	Description
❶	**Line Monitor (Host station)** Lists all connection relevant CC-Link information. For details on the indication of special relays (SB) and special registers (SW) refer to: **Loop test**: See section below **Monitoring other station**: See section below
❷	**Module setting** (Q series only): Specify the module the CC-Link diagnostics have to be executed for. Before executing make sure that your Q CPU and QJ61BT11 are of function version B or higher.
❸	**Network test** The data link is started or stopped for the specified CC-Link module.
❹	**Monitor start/stop** The monitoring of the host station is started or stopped.
❺	**Acquire the setting information** The setting information of the CC-Link module is acquired for the CC-Link status. After execution the acquired setting information can be set as EEPROM parameters by turning ON the device YnA via the **Device Test**.
❻	**Loop test:** See section below
❼	**Monitoring other station:** See section below

Tab. 9-9: *CC-Link Diagnostics items*

Monitoring other station

The CC-Link diagnostics monitor the CC-Link line connected to the PLC CPU via other stations.

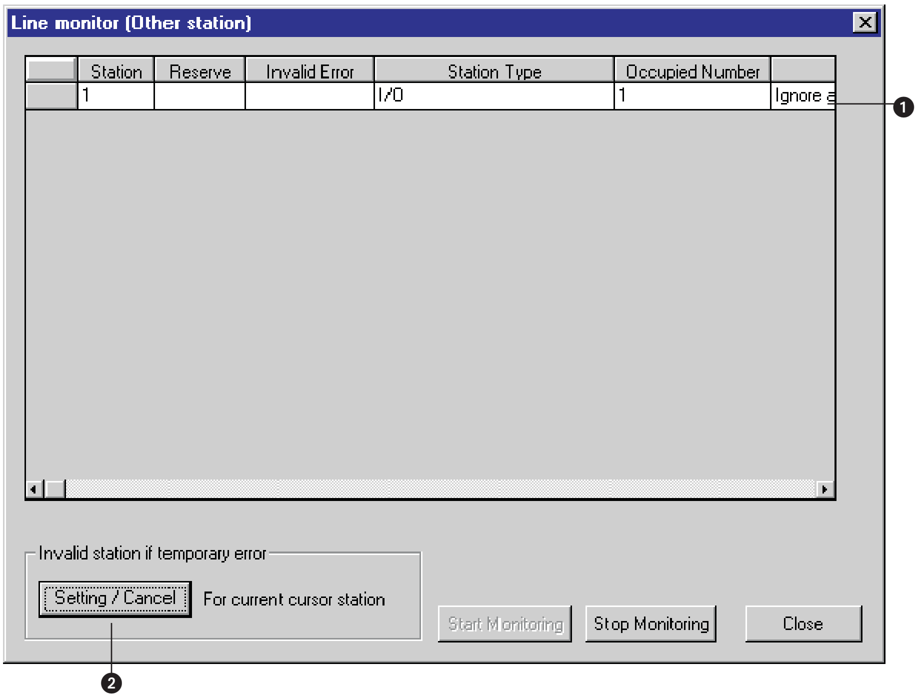


Fig. 9-13: Line monitor (Other station)

Item	Description
❶	**Information on other station** Acquires and displays information on reserve stations, settings for invalid errors, station types, and occupied station numbers from the Q CPU.
❷	**Invalid station if temporary error** Modules that have been replaced without corresponding remote station are detected in online operation mode.

Tab. 9-10: Line monitor (Other station) items

Maker Code
Displays the manufacturer's name and identification of the modules connected via CC-Link. The manufacturer identification is also displayed, in case the manufacturer name has not been entered.
The remote I/O modules are represented as blank entry because they do not provide a manufacturer name or identification.
The list of manufacturers is saved as file (PARTNER.DAT) in the GX IEC Developer installation directory. It can be edited with a text (ASCII) editor.
When editing the list the following formatting must be kept:

Description	Maker Code	Company name
Maximum number of characters	5	255
Allowed characters ①	0 – 9	Alphanumeric, symbols
Remarks	Manufacturer identification not consisting of 5 digits is ignored.	Display varies depending on screen size. More than 75 characters can not be displayed in one window.

Tab. 9-11: PARTNER.DAT formatting

① The maker's name should be completely entered in " ". Example: "##### Co., Ltd."

Loop test

The loop test tests all stations or a selected station.

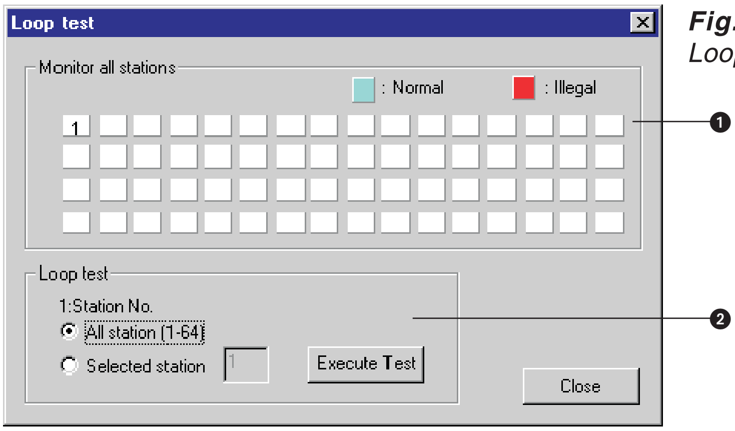


Fig. 9-14:
Loop test

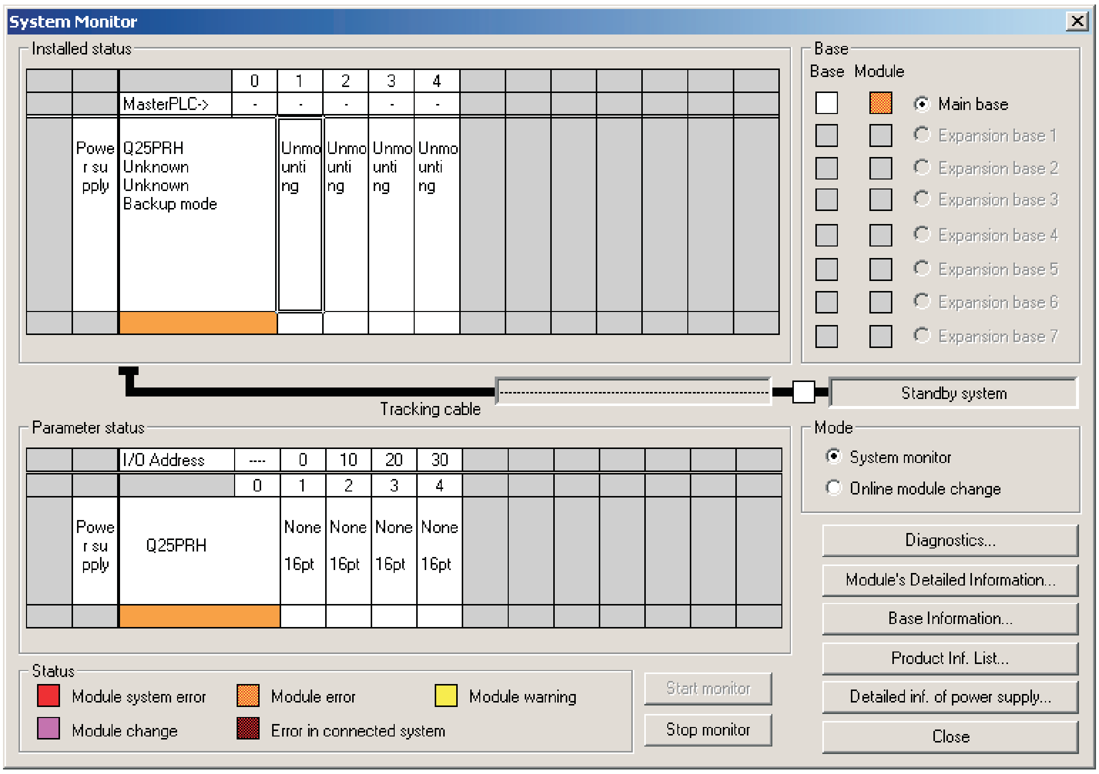
Item	Description
❶	**Monitor all stations** Monitors all CC-Link stations Illegal stations are indicated in red.
❷	**Loop test** Specify whether to test **All stations** or a **Selected station**. Click **Execute Test** to execute the loop test.

Tab. 9-12: Loop test items

9.1.5 System Monitor

The System Monitor displays comprehensive information on the entire connected PLC system. The System Monitor supports Q series PLCs only.

In the **Debug** menu select **System Monitor** to open the following dialogue:



System Monitor

Installed status

			0	1	2	3	4
		MasterPLC->	-	-	-	-	-
	Powe r su pply	Q25PRH Unknown Unknown Backup mode	Unmo unti ng	Unmo unti ng	Unmo unti ng	Unmo unti ng

Base

Base	Module	
□	▨	● Main base
□	□	○ Expansion base 1
□	□	○ Expansion base 2
□	□	○ Expansion base 3
□	□	○ Expansion base 4
□	□	○ Expansion base 5
□	□	○ Expansion base 6
□	□	○ Expansion base 7

Parameter status

		I/O Address	----	0	10	20	30
			0	1	2	3	4
	Powe r su pply	Q25PRH		None 16pt	None 16pt	None 16pt	None 16pt

Mode

● System monitor

○ Online module change

Status

🟥 Module system error ▨ Module error 🟨 Module warning

🟪 Module change ▨ Error in connected system

Fig. 9-15: Mode: System Monitor

Installed status

This section illustrates the selected base and shows the installed modules and vacant slots (”not mounted”).

Base

This section lists the connected bases and displays the error status coded by colors (see section “Status”). If several bases are connected, the base to be displayed can be selected via radio buttons.

Parameter status

This section gives additional information on the installed modules and their addresses.

Status

Indicates the color codes for the status displayed in the “Base” section.

Start monitor/Stop monitor

Starts or stops monitoring.

Diagnostics

Also available in the context menu as **Diagnosis feature execution**.

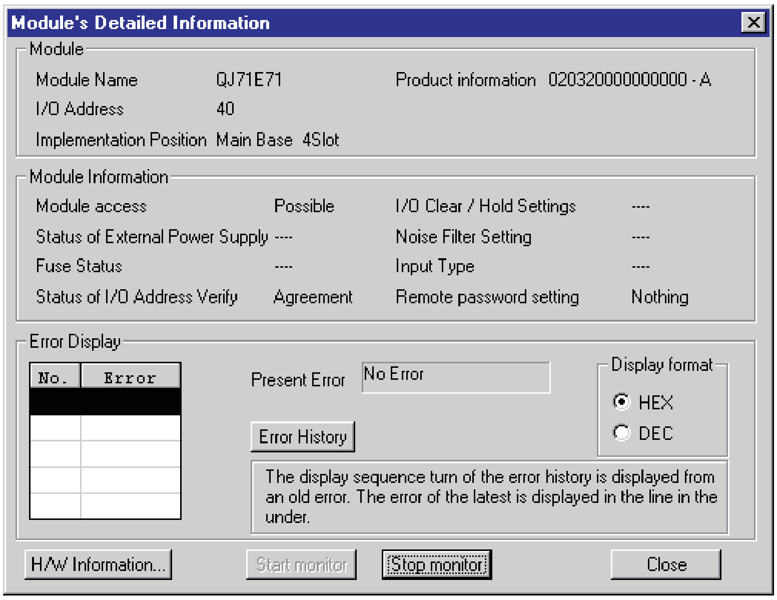
Opens appropriate diagnostics functions for the selected module:

- For a CPU module the **PLC diagnostics** dialogue (see section 9.1.1) is opened.
- For an Intelligent Function Module the **Intelligent Function Module Utility** (see section 9.9) is opened.
- For network modules the **Networks Diagnostics** dialogue (see section 9.1.2) is opened.

Module’s Detailed Information

Also available in the context menu as **module’s detailed information** or via double-click on a module.

Opens the following dialogue that displays detailed information on the selected module.



Module's Detailed Information

Module

Module Name QJ71E71

Product information 020320000000000 - A

I/O Address 40

Implementation Position Main Base 4Slot

Module Information

Module access	Possible	I/O Clear / Hold Settings	----
Status of External Power Supply	----	Noise Filter Setting	----
Fuse Status	----	Input Type	----
Status of I/O Address Verify	Agreement	Remote password setting	Nothing

Error Display

No.	Error

Present Error No Error

Display format: ◉ HEX ○ DEC

The display sequence turn of the error history is displayed from an old error. The error of the latest is displayed in the line in the under.

Fig. 9-16:
Module’s Detailed Information

NOTE It may take a few seconds until all details are displayed, especially those for the error display.

Error History: Displays the stored error history.

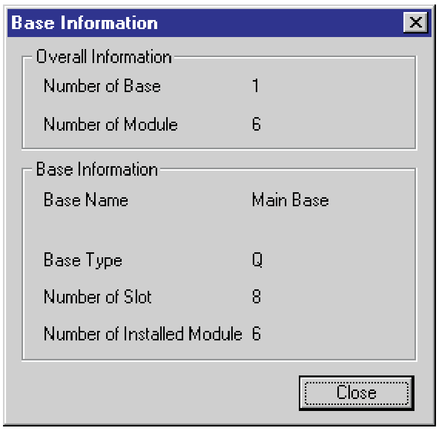
H/W Information: Opens a dialogue that displays more details on the LED and SW status.

Start monitor: Starts monitoring.

Stop monitor: Stops monitoring

Base Information

Opens the following dialogue that displays information on the selected base.



Base Information

Overall Information

Number of Base	1
Number of Module	6

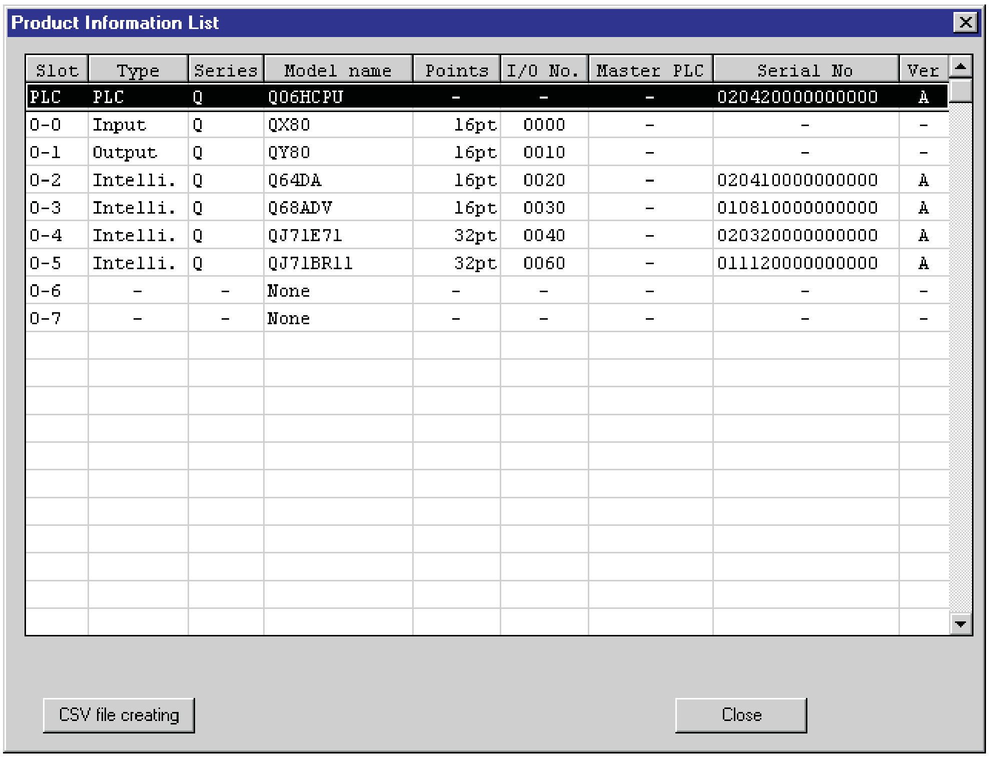
Base Information

Base Name	Main Base
Base Type	Q
Number of Slot	8
Number of Installed Module	6

Fig. 9-17:
Base Information

Product Inf. List

Opens the **Product Information List** that gives an overview of the connected PLC system. The **Product Information List** can be exported into a CSV file (comma separated value) via the **CSV file creating** button. The CSV format can be imported by many other applications.



Product Information List

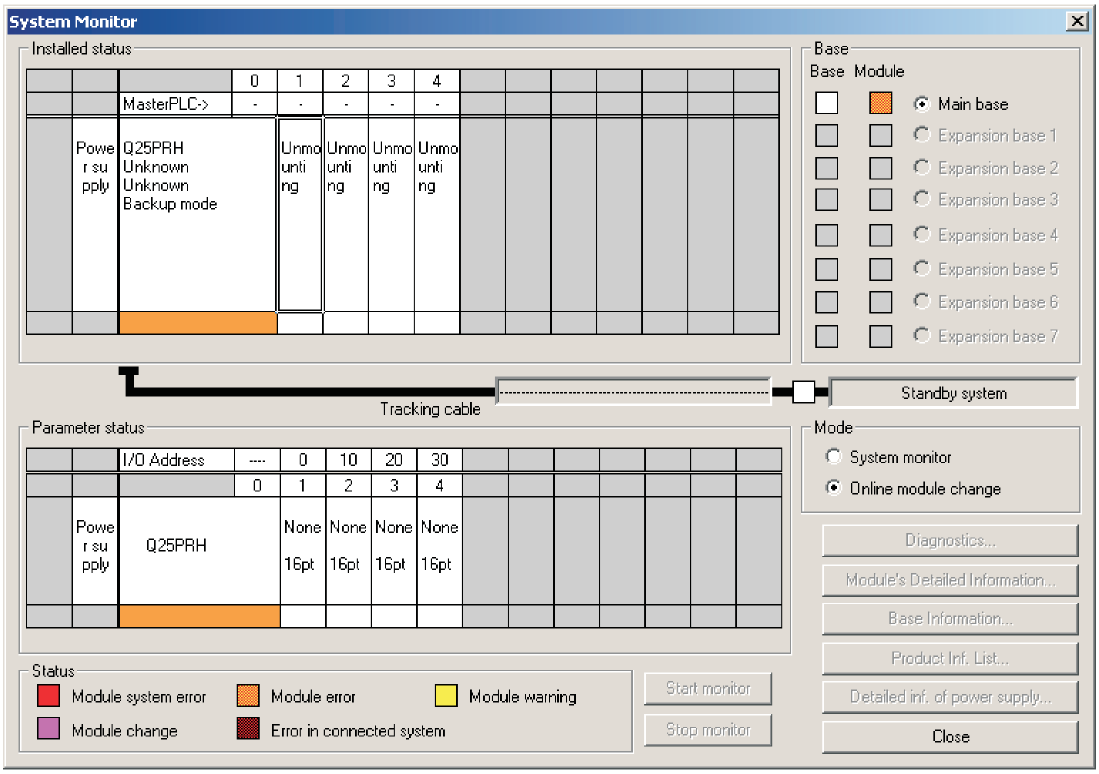
Slot	Type	Series	Model name	Points	I/O No.	Master PLC	Serial No	Ver
PLC	PLC	Q	Q06HCPU	-	-	-	020420000000000	A
0-0	Input	Q	QX80	16pt	0000	-	-	-
0-1	Output	Q	QY80	16pt	0010	-	-	-
0-2	Intelli.	Q	Q64DA	16pt	0020	-	020410000000000	A
0-3	Intelli.	Q	Q68ADV	16pt	0030	-	010810000000000	A
0-4	Intelli.	Q	QJ71E71	32pt	0040	-	020320000000000	A
0-5	Intelli.	Q	QJ71BR11	32pt	0060	-	011120000000000	A
0-6	-	-	None	-	-	-	-	-
0-7	-	-	None	-	-	-	-	-

Fig. 9-18: *Product Information List*

9.1.6 Online Module Change

The **System Monitor** display also provides the option of online module change (hot swapping) with the QnPH, QnPRH and Q Remote I/O PLCs.

In the **Debug** menu select **Online Module Change** to open the following dialogue or select the different mode, if it was opened via **System Monitor** already:



System Monitor

Installed status

			0	1	2	3	4
		MasterPLC->	-	-	-	-	-
	Powe r su pply	Q25PRH Unknown Unknown Backup mode	Unmo unti ng	Unmo unti ng	Unmo unti ng	Unmo unti ng	

Base

Base	Module	
□	▨	● Main base
□	□	○ Expansion base 1
□	□	○ Expansion base 2
□	□	○ Expansion base 3
□	□	○ Expansion base 4
□	□	○ Expansion base 5
□	□	○ Expansion base 6
□	□	○ Expansion base 7

Tracking cable

Standby system

Parameter status

		I/O Address	----	0	10	20	30
			0	1	2	3	4
	Powe r su pply	Q25PRH		None 16pt	None 16pt	None 16pt	None 16pt

Mode

○ System monitor

● Online module change

Status

🟥 Module system error ▨ Module error 🟨 Module warning

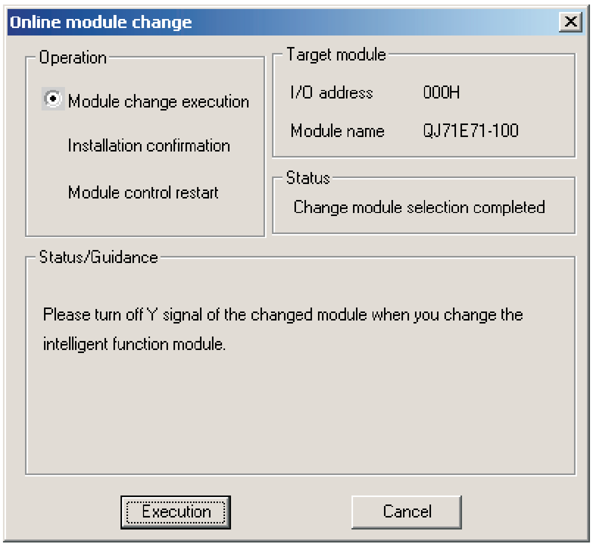
🟪 Module change ▨ Error in connected system

Fig. 9-19: *Mode: Online module change*

The following explains a sequence of operations for making online module change from **System monitor**.

① Choose the module you want to change from Mounting status on the System monitor screen. (Module selection waiting status)

② After the Online module change dialogue box has appeared, click the **Execution** button. (Online module change execution confirmation)



Online module change

Operation
- ● Module change execution
- Installation confirmation
- Module control restart

Target module
- I/O address 000H
- Module name QJ71E71-100

Status: Change module selection completed

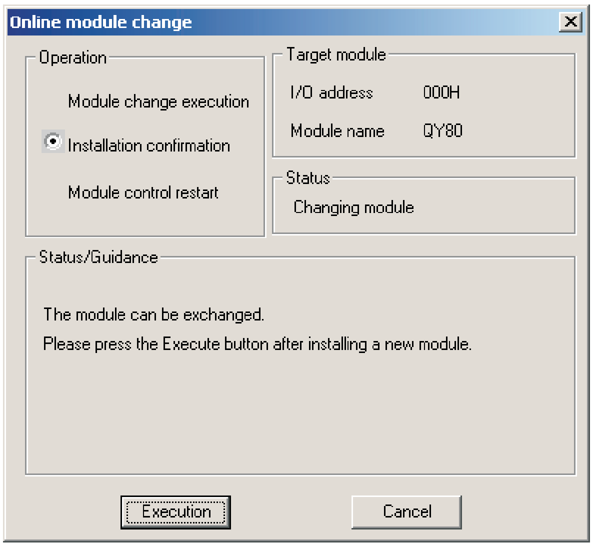
Status/Guidance: Please turn off Y signal of the changed module when you change the intelligent function module.

Fig. 9-20:
Online module change – execution confirmation

NOTE

Depending on the faulty condition of the module to be changed, the type of the module may not be acquired. In that case, "Unknown" appears in the Target module name display field.

③ The dialogue box indicating that online module change can be made appears. (Online module change completion confirmation)



Online module change

Operation
- Module change execution
- ● Installation confirmation
- Module control restart

Target module
- I/O address 000H
- Module name QY80

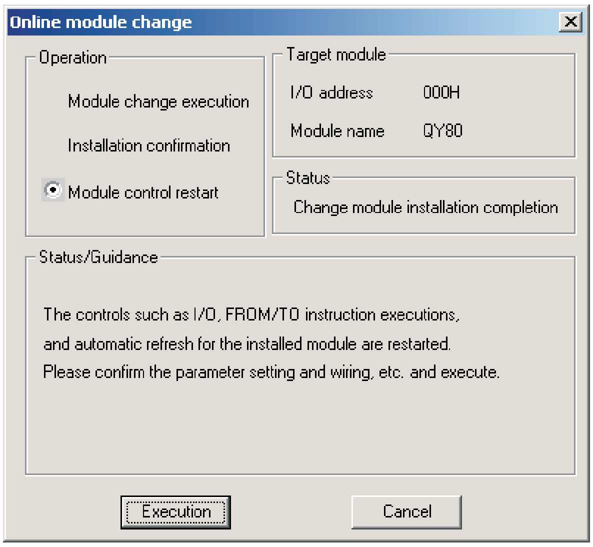
Status: Changing module

Status/Guidance: The module can be exchanged.
Please press the Execute button after installing a new module.

Fig. 9-21:
Online module change – completion confirmation

④ After changing the module, click the **Execution** button.

⑤ If online module change is completed properly, the following dialogue box appears. (Module control resumption confirmation)



○ Module change execution

○ Installation confirmation

● Module control restart

I/O address 000H

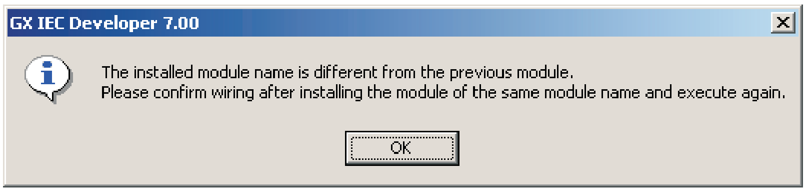
Module name QY80

Change module installation completion

The controls such as I/O, FROM/TO instruction executions, and automatic refresh for the installed module are restarted. Please confirm the parameter setting and wiring, etc. and execute.

Fig. 9-22: *Online module change – resumption confirmation*

If it is not completed properly, the following dialogue box appears.



The installed module name is different from the previous module.
Please confirm wiring after installing the module of the same module name and execute again.

Fig. 9-23: *Online module change – error*

Click the **OK** button. As clicking that button the **Online module change** dialogue box is displayed again, and you can restart the procedure from step ②.

NOTES

GX IEC Developer cannot perform the online module change of multiple modules simultaneously.

Note that if the Process CPU/redundant CPU is in either of the following statuses, online module change cannot be performed.

- PLC CPU's RESET/L.CLR is reset.
- PLC CPU is stopped due to stop error.

If online module change is performed from two personal computers to one PLC CPU, an error message appears on the personal computer where online module change was performed later.

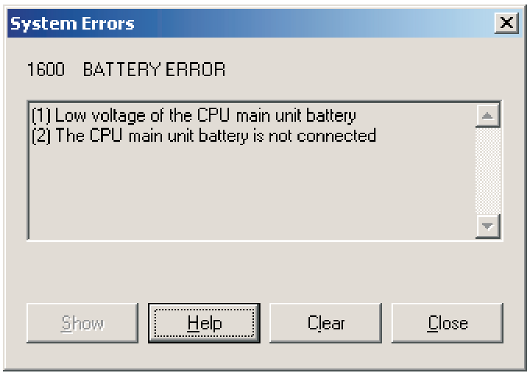
If the PLC CPU is reset or powered off while the Online module change execution dialogue box is being displayed, online module change cannot be continued and an error message appears.

9.2 PLC System Error Check

9.2.1 System errors

If a system error occurs, GX IEC Developer automatically outputs the corresponding error code in the **System Errors** dialogue.

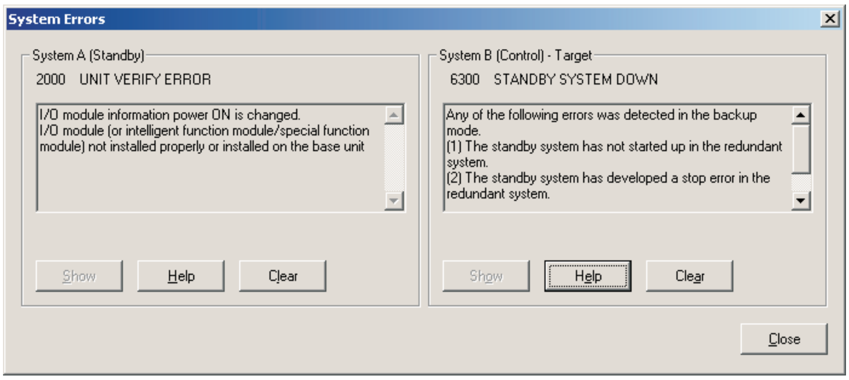
The **System Errors** dialogue can be opened by **Debug** → **System errors**.



1600 BATTERY ERROR

(1) Low voltage of the CPU main unit battery
(2) The CPU main unit battery is not connected

Fig. 9-24:
***System Errors** dialogue*



2000 UNIT VERIFY ERROR

I/O module information power ON is changed.
I/O module (or intelligent function module/special function module) not installed properly or installed on the base unit

6300 STANDBY SYSTEM DOWN

Any of the following errors was detected in the backup mode.
(1) The standby system has not started up in the redundant system.
(2) The standby system has developed a stop error in the redundant system.

Fig. 9-25:
***System Errors** dialogue for QnPRH*

The button **Show** opens the editor window containing the error location of the selected error if it is possible to show it. You can obtain additional information on the error message by selecting the **Help** button.

A system error can be cleared by selecting the button **Clear**.

NOTES

Clearing the system error is only possible for errors that allow the PLC to continue operation. If the cause of the error is still present, it will be re-registered in the failure history.

The **Clear** error functionality is available for all Q and QnA series PLCs only.

9.2.2 User-defined errors

These functions are not available for the FX series.

Error flags combined with an error location program make it possible to display errors that occur in the course of the process. If an error is encountered, an error flag (F) is set, together with a special relay (M/SM) to indicate the presence of an error condition. The error flags F0 – F127 can be programmed for example in the MELSEC network with the OUT function.

The text for the error messages must be stored in the error file USER_ERR.TXT in the current project's directory.

How to create an error messages file

① Open the Windows Notepad editor (you can also use any other editor that saves files in ASCII format without formatting codes).

② Create a new file.

③ Enter the error messages, one message per line.
Syntax:

```
0, Text1
1, Text2
...
```

You must enter a space between the comma and the text of the error message.

Meaning:

In the example above Text1 will be displayed, if F0is set and Text2 is displayed, if F1 is set.

④ Store the file as USER_ERR.TXT in the current project's directory.

9.2.3 Hardware Diagnostics

Selecting **HW Diagnosis** in the **Debug** menu shows all hardware errors registered
(not available for FX series).

M9000 (A) / SM60 (Q/QnA): Fuse blown. This relay is set as soon as a blown fuse is registered in one or more modules in the system. The relay will remain set even after you switch back to normal status. (Does not apply for A2C CPUs.)

M9002 (A) / SM61 (Q/QnA): I/O module verify error.

M9005 (A) / SM53 (Q/QnA): Mains power drop. The relay is set after a power drop of no more than 20 ms. You can reset the relay after switching the power supply off and on again.

M9006 (A) / SM52 (Q/QnA): Battery low. The voltage of the retentive memory battery has dropped below the minimum level. This flag is reset when the battery is replaced.

M9007 (A) / SM51 (Q/QnA): Battery low (latch relay). The voltage of the retentive memory battery has dropped below the minimum level. This flag is reset when the battery is replaced.

M9026 (A) / SM211 (Q/QnA): Clock data error. Set in the event of incorrect clock data (registers D9025 – D9028). (Only for AnN/AnS CPUs.)

Hardware errors are indicated by cross symbols instead of minus signs on the corresponding buttons. Clicking on a button with a cross displays context-sensitive help for that error.

NOTE

You will find detailed descriptions and information on special relays and special registers in the controller manuals.

9.2.4 Scan

The **Scan** command in the **Debug** menu provides a statistical display of the scan time performance of the program in the PLC CPU.

The program execution cycle periods can vary depending on status conditions, e.g. whether inputs are polled during the cycle or not. The PLC CPU logs the following scan times:

Actual Scan Time: The current program cycle time (special register D9018 in the A series; D8010 in the FX series)

Minimal Scan Time: The last minimum scan time saved (special register D9017 in the A series; D8011 in the FX series)

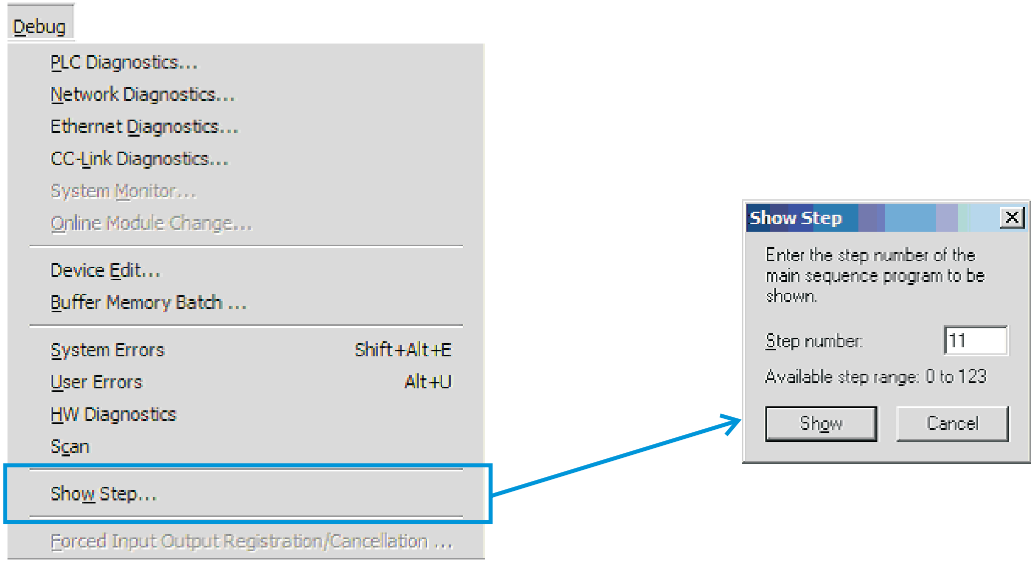
Maximal Scan Time: The last maximum scan time saved (special register D9019 in the A series; D8012 in the FX series)

9.2.5 Show Step

To show the location of a given PLC Main program step in the source code you use the **Show Step** command in the **Debug** menu.

When you call the menu command **Debug** → **Show Step**, a dialog will be displayed to enter the step number. The maximum number of characters in the **Step number** edit box of the **Show Step** dialogue is limited to the number of characters in the highest step number of the current main program. E.g. if the main program has 7490 steps, then the edit box is limited to a maximum of 4 characters.

After pressing **Show**, the body where the selected step is located will be opened, and the line or object corresponding to that step will be marked.



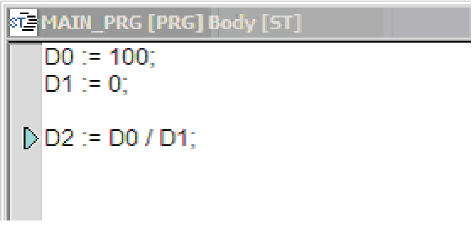
Enter the step number of the main sequence program to be shown.

Step number: 11

Available step range: 0 to 123

Fig. 9-26: *Menu command* ***Debug*** *→* ***Show Step***

If the selected step number is valid, the location of the step will be marked:

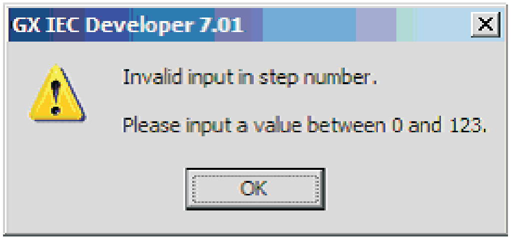


```
D0 := 100;
D1 := 0;

D2 := D0 / D1;
```

Fig. 9-27:
The desired step is marked and displayed

If the selected step number is invalid, the following error message is shown:



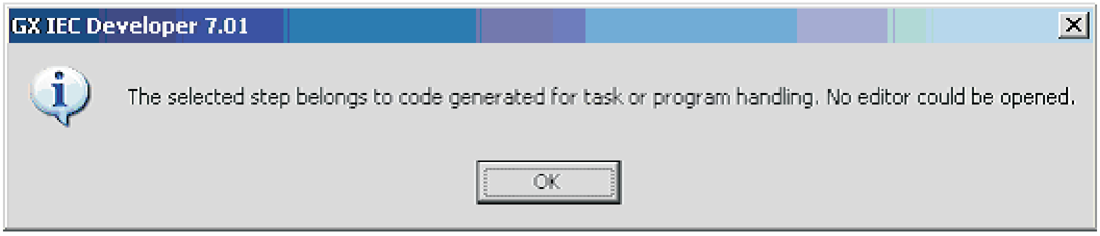
Invalid input in step number.

Please input a value between 0 and 123.

Fig. 9-28:
Error message

In the message the actual step range of the program is shown.

If the selected step number is valid, but the code does not belong to any of the editors, the following message is displayed:


GX IEC Developer 7.01

The selected step belongs to code generated for task or program handling. No editor could be opened.

Fig. 9-29: *Message*

NOTES

This command is available, if a compiled project is open. It is disabled for Remote I/O projects.

This functionality can only show steps of the Main program. Showing step locations in the Sub and SFC programs is not supported.
Since the SFC POUs of FX series projects are compiled into the Main program, it is possible that a step of a FX-series Main program is located in a SFC body. If you specify a step that originates from a SFC POU, the appropriate body will be opened, but no source code position will be marked.

9.3 The Device Editor

The device editor is a tool for displaying and editing the contents of the PLC CPU's word and bit devices. The main element of the device editor is a table that is very similar to a Microsoft Excel worksheet, both in its appearance and in the way it is used.

You can read the values from the CPU to the table, and after editing them you can then write the new values back to the CPU. Additionally, you can also write the table contents to an external file and load external files for editing. These files are stored in Excel 4.0 format, which makes it possible to edit them with Microsoft's Excel spreadsheet program as well.

Maximum table size is 16,383 lines and 255 columns, which means it can handle a total of 4,177,665 cells. Effectively, the number of bit and word devices you can load to the table is only restricted by the amount of available memory on your computer.

Warning prompts are displayed before all operations that alter table ranges, before critical operations (e.g. writing to CPU inputs) and before you close the device editor.

9.3.1 The worksheet

The **Device Editor** option in the **Debug** menu opens the **Device Edit** window, which contains the worksheet table in which all editing actions are performed.

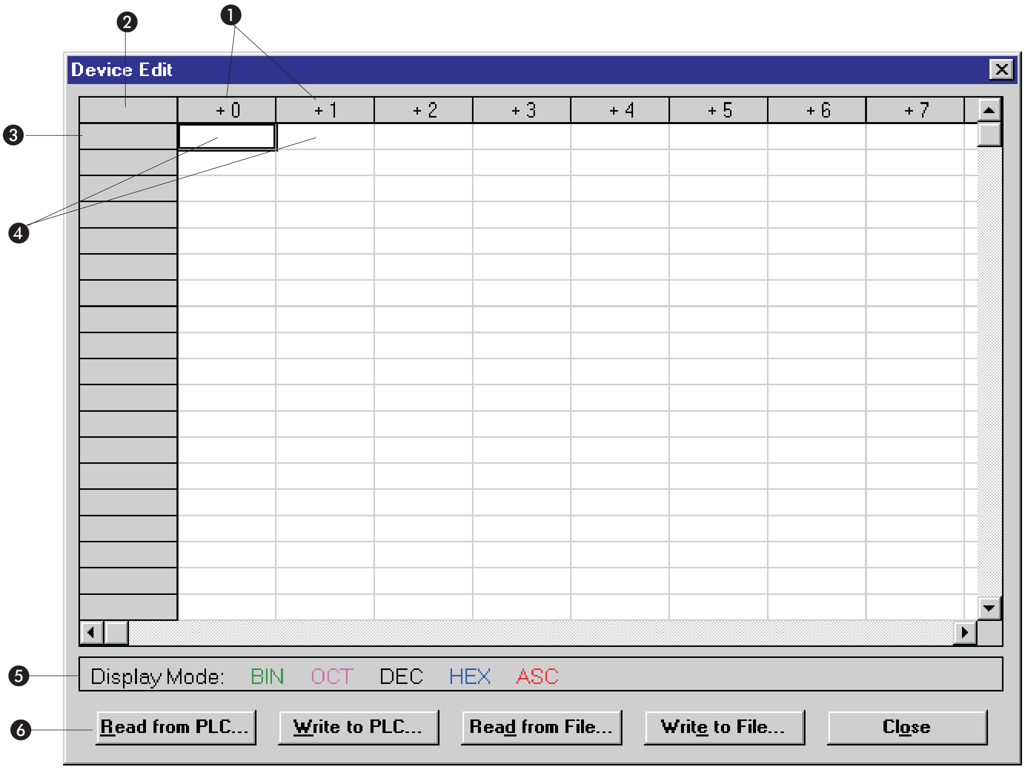


Fig. 9-30: *Device editor*

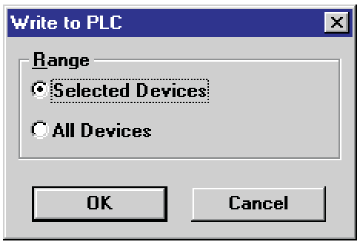
Item	Description
❶	The numbers of the devices following the first device are obtained by adding the values shown at the heads of the column.
❷	The leftmost column of the table contains the first of a consecutive sequence of devices.
❸	Each series of devices is displayed across a table line.
❹	The values of the devices are contained in the individual table cells.
❺	The current display mode of the selected cell is shown underlined. The display mode is set in the pop-up menu (see below).
❻	Commands can be accessed by clicking on the buttons (see below)or with the pop-up menu (see below).

Tab. 9-13: *Device editor items*

9.3.2 Buttons

Read from PLC: Reads the values of word and bit devices (registers) from the PLC. You can choose to read either the entire table or a selected table range.

Write to PLC: Writes the table values to the PLC's word and bit devices (registers). You can write either the entire table or a selected table range.



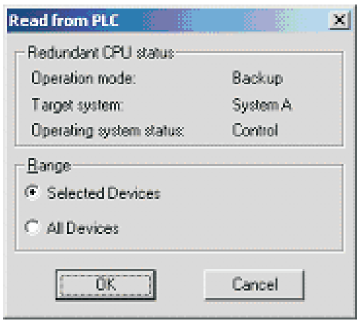
Write to PLC

Range

● Selected Devices

○ All Devices

Fig. 9-31:
Selected Devices: Read/write only the devices selected in the table.
All Devices: Read/write all the registers in the table.



Read from PLC

Redundant CPU status

Operation mode: Backup

Target system: System A

Operating system status: Control

Range

● Selected Devices

○ All Devices

Fig. 9-32:
For QnPRH:
Selected Devices: Read/write only the devices selected in the table.
All Devices: Read/write all the registers in the table.

Read from File: Loads an external file stored in Excel 4.0 format.

Write to File: Writes the GX IEC Developer device table data to an external file in Excel 4.0 format.

WARNING:
The device table settings are* not *stored with the GX IEC Developer project. They are lost irrevocably when you close the* Device Edit *window. If you want to access the data again later you should always save the table to a file before closing the edit window.

9.3.3 The pop-up menu

To display the pop-up menu click with the right mouse button or press Shift F10.

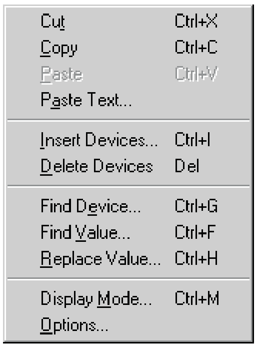


Fig. 9-33:
Device editor pop-up menu

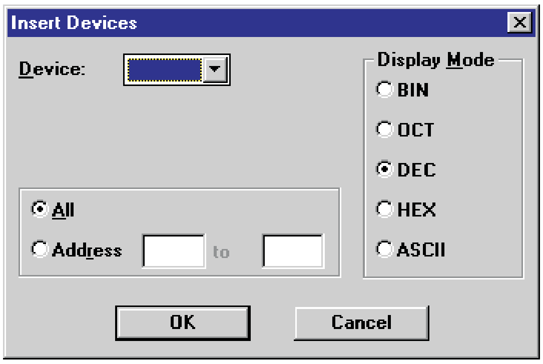
Cut: Copies the contents of selected cells to the clipboard and deletes them from the table (Ctrl X)

Copy: Copies the contents of selected cells to the clipboard (Ctrl C)

Paste: Pastes the content of the clipboard to the table (Ctrl V)

Paste Text: Opens a dialogue box in which you can enter ASCII text to paste to the table. ASCII characters are only displayed in the table in the ASC display mode. You cannot enter more characters than the selected device can store. A data register can only store two ASCII characters, for example. You can enter more characters by selecting several cells (i.e. a series of registers) and entering ASCII text.

Insert Devices: Displays this dialogue box:



Insert Devices

Device:

Display Mode
○ BIN
○ OCT
● DEC
○ HEX
○ ASCII

● All
○ Address to

OK Cancel

Fig. 9-34:
Device*: List of the available devices for selection.*
All*: Selects the entire range of devices for insertion in the table.*
Addresses: Allows specification of a specific address range.
Display Mode*: Specifies the display mode for the new registers*

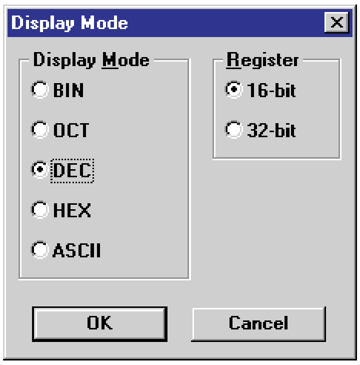
Delete Devices: Deletes the selected device or device range. A prompt is displayed asking for confirmation before the delete operation is executed.

Find Device: Searches for specific devices. You can select the search direction.

Find Value: You can also search for a specific value in a device register. Again, you can specify the search direction.

Replace Value: Search and replace function for replacing device register values.

Display Mode: Displays this dialogue box:



Display Mode

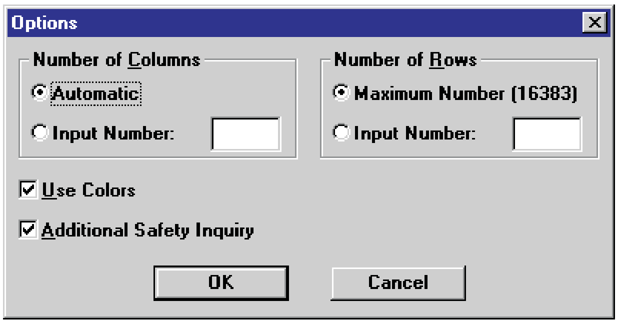
Display Mode
○ BIN
○ OCT
● DEC
○ HEX
○ ASCII

Register
● 16-bit
○ 32-bit

OK Cancel

Fig. 9-35:
Display Mode: *Mode for displaying the selected devices*
Register*: Selects 16 or 32 bit register display.*

Options: Displays the following dialogue box in which you can set the basic options for the table structure:



Options

Number of Columns
● Automatic
○ Input Number:

Number of Rows
● Maximum Number (16383)
○ Input Number:

☑ Use Colors
☑ Additional Safety Inquiry

OK Cancel

Fig. 9-36:
Device editor options

Number of Columns
Automatic: 8 rows for octal registers, 10 rows for decimal registers, 16 rows for hex registers
Input Number: You can enter a value between 1 an 255.

Number of Rows
Maximum Number (16383): Sets the rows to the maximum of 16,383 rows.
Input Number: You can enter a value between 1 and 16,383.

Use Colors: When this check box is selected the table is displayed using different colors.

Additional Safety Inquiry: When this check box is selected the system displays an additional prompt asking for confirmation for some critical operations.

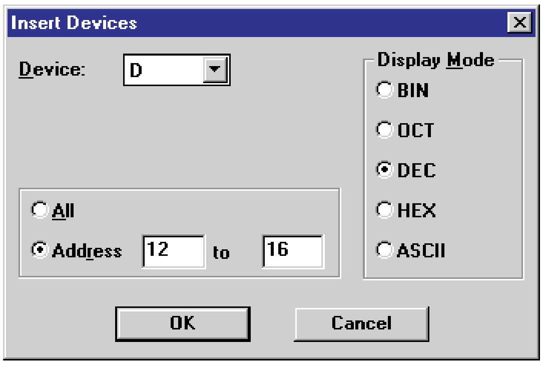
Inserting devices with the Insert Devices dialogue box

Open the **Device Edit** window and select the top left cell of the table.

① Click with the right mouse button to open the pop-up menu.

② Click on **Insert Devices** with the left mouse button to display the **Insert Devices** dialogue box.

③ Select the device type, address range and display mode, then click on **OK**.



Insert Devices

Device: D

Display Mode
○ BIN
○ OCT
● DEC
○ HEX
○ ASCII

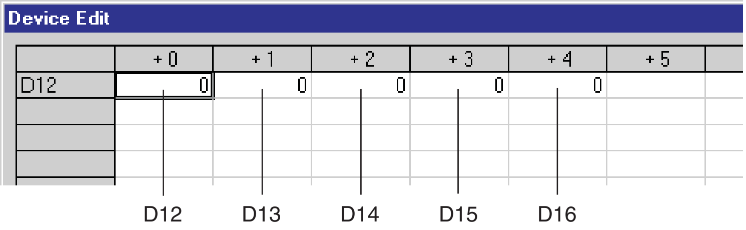
○ All
● Address 12 to 16

Fig. 9-37:
Data registers D

Address range: D12 – D16

Display mode: Decimal

The devices are inserted in the table. Initially, all the devices are given a default value of 0.



Device Edit

	+0	+1	+2	+3	+4	+5
D12	0	0	0	0	0	

Fig. 9-38:
Devices D12 – D16 have been inserted in the table.

Inserting devices directly

Open the **Device Edit** window.

① Click in any cell in a free line. All the cells in the line must be empty!

② Enter the device. The system automatically switches to editing mode when you make an entry. GX IEC Developer only accepts valid entries. If you make an incorrect entry editing mode will remain active.

The inserted device is then displayed in the leftmost table column.

NOTE

When you enter registers directly you can only store one register in a line; the remaining cells in the line are disabled and cannot be selected.

Entering device values

① Select the cell where you want to enter the value.

② Enter the new value. The system switches to editing mode automatically when you make an entry. GX IEC Developer only accepts valid entries. If you make an incorrect entry editing mode will remain active.

NOTE Double-clicking on a bit device (or pressing Space) toggles the value of the device between 0 and 1. Double-clicking on a word device activates editing mode.

Entering 32-bit values

A 32-bit value is stored in two 16-bit registers.

① Select the first of the two table cells.

② Hold down the Shift key and extend the selection to include the second cell using the mouse or cursor key. Both cells now have a bold outline and the second cell is shaded.

③ Press Ctrl M to select the **Display Mode** dialogue box from the pop-up menu.

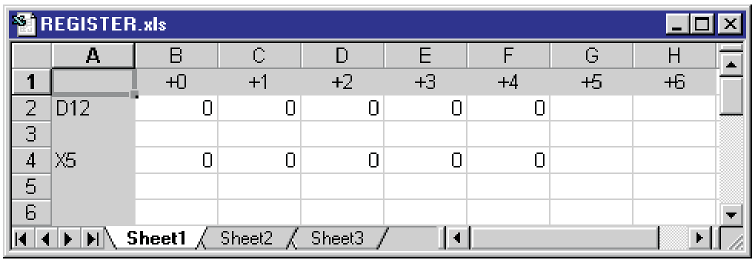
④ Select the **32-bit** option in the **Registers** field.

⑤ Click on **OK** to link the two selected cells.

⑥ Enter a 32-bit value in the cell. GX IEC Developer will only accept valid entries.

NOTE You can use the standard Windows mouse actions and key combinations in the device table. Special key combinations are listed in the Appendix B of this manual.

9.3.4 Editing tables in Microsoft Excel



	A	B	C	D	E	F	G	H
1		+0	+1	+2	+3	+4	+5	+6
2	D12	0	0	0	0	0		
3								
4	X5	0	0	0	0	0		
5								
6								

Fig. 9-39:
Entries in Excel

D12 – D16

X5 – X9

Changing device designations

① Overwrite the entry in D12 in leftmost column, e.g. by entering D24.

The line now contains the five data registers D24 – D28.

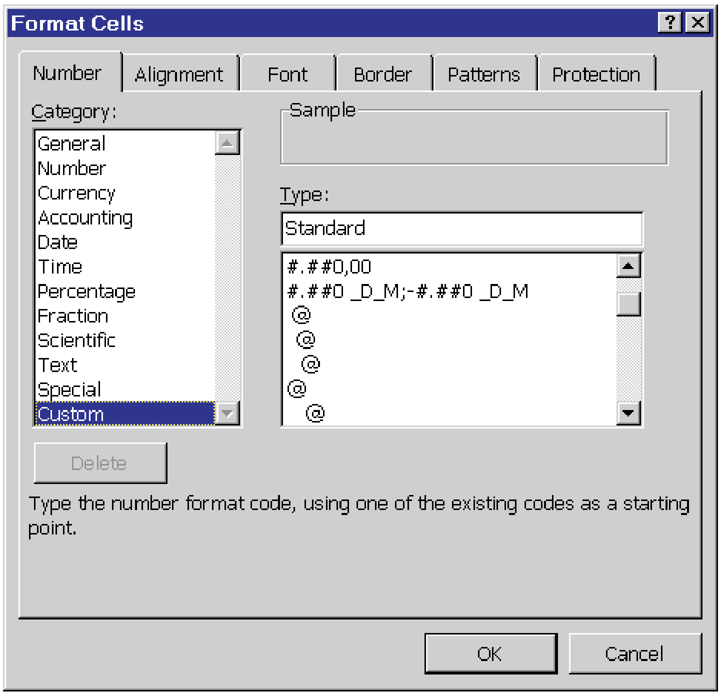
Adding devices to a line

① Select the cell in column +5 and enter the value to be contained by the new register D17.

② Click with the right mouse button to open the pop-up menu and select the **Number Format** option.

③ Select the appropriate number format from the list in the dialogue box.

The number formats in Excel correspond to the display mode in GX IEC Developer. These formats are represented by a fixed number of spaces and the @ character:



Category:
General
Number
Currency
Accounting
Date
Time
Percentage
Fraction
Scientific
Text
Special
Custom

Type:
Standard

```
#.##0,00
#.##0 _D_M;-#.##0 _D_M
 @
  @
   @
@
    @
```

Type the number format code, using one of the existing codes as a starting point.

Fig. 9-40:

BIN: 1 space
HEX: 2 spaces
ASC: 3 spaces
DEC: No spaces
OCT: 4 spaces

NOTE

Setting the correct number format (i.e. the right display mode) is absolutely essential when you add new devices in an Excel table that you plan to reload in GX IEC Developer later!

9.4 Forced Settings

9.4.1 Forced Input/Output

By using the function **Forced Input Output Registration/Cancellation** it is possible to forcefully set or reset an input (X device) and output (Y device) of a PLC. The user can easily debug a program that is under execution in the PLC by intentionally setting or resetting the X/Y devices. The function facilitates a detailed study of input and output changes.

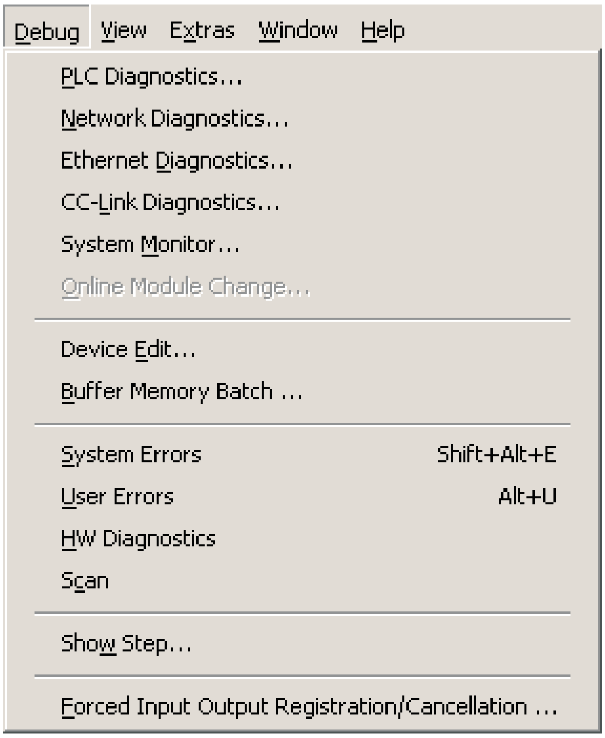


Fig. 9-41:
*Menu **Debug***

NOTES

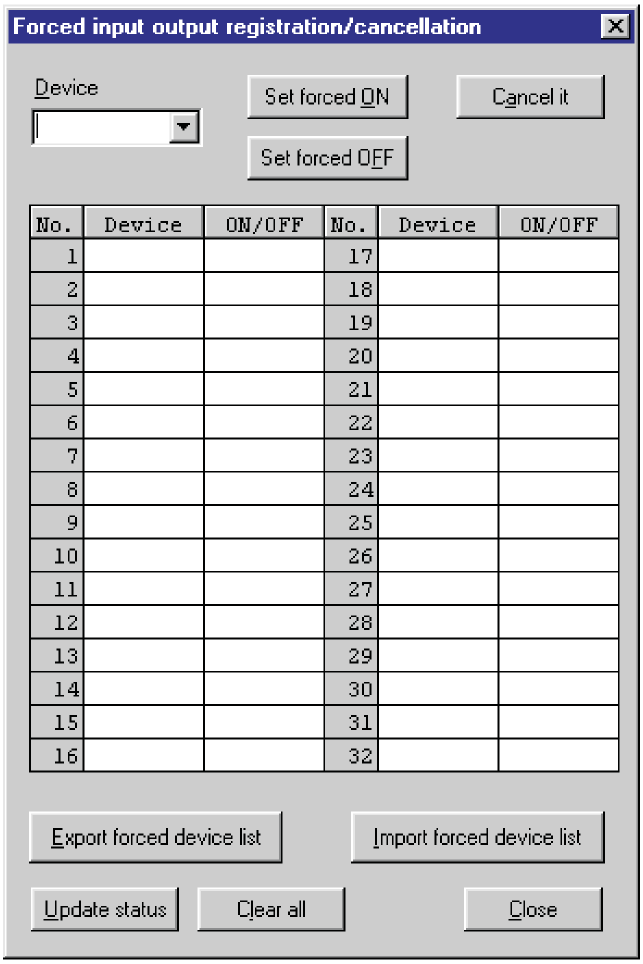
Once a device X/Y is forced to turn ON/OFF, its status is not affected by the input (X) from an external device and the output (Y) of the program. The forced input/output status can be cancelled only through the **Forced Input Output Registration/Cancellation** function.

Since the exclusive control is not exercised on the PLC CPU side, forcefully setting or resetting a device may cause the registration to be changed from multiple GX IEC Developer programs or from the equipment which can register/cancel forced devices.

The **Forced input/output** function is available only in Q series PLCs (except Q00(J)/01) and Q Remote I/O modules, but is not supported by the A, QnA and FX series.

The **Forced input/output** function is not supported by a standby system of a QnPRH PLC.

Select **Forced Input Output Registration/Cancellation** in the menu **Debug** to open the following dialogue box.



Forced input output registration/cancellation
Device
Set forced ON
Cancel it
Set forced OFF
Export forced device list
Import forced device list
Update status
Clear all
Close

No.	Device	ON/OFF	No.	Device	ON/OFF
1			17		
2			18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Fig. 9-42:
*Dialogue box **Forced Input Output…***

The forced input output registration and cancellation can be executed from the **Forced input output registration/cancellation** dialogue. The device name entered in the dialogue can be forced to turn ON/OFF using the respective command button.
Most recently used devices can be automatically added by using the selection list **Device**. The maximum number of devices that can be stored in the **Device** list is 5. When the 6th device is entered in the device name list, the least recently used or the very first entry in the list is deleted from it.

By pressing the button **Set forced ON**, the X/Y device entered/selected in the Devices list is set to ON regardless of the current device status.
When the **Set forced OFF** button is pressed, the X/Y device entered/selected in the Devices list is set to OFF regardless of the current device status.

The devices that are currently under forced input/output are displayed in a list in the dialogue so that the status of forced registration/cancellation can easily be checked.

NOTE

Up to 32 devices can be set to forced ON/OFF.

When the **Cancel it** button is pressed, the forced ON/OFF status of the X/Y device entered/selected from the devices list **Device** is cancelled and the device name is also removed from the forced I/O devices display list.

To export the listed devices under forced ON/OFF press the **Export forced device list** button. This opens a file save dialogue to select the path for saving the file. The default path is the project path. The default file name is “ForcedDeviceList.xls”. The file format is the Excel format.

For importing a saved device list press the **Import forced device list** button. A file open dialogue is opened to select the list file to be loaded. The default path is the project path. The default file name is “ForcedDeviceList.xls”.

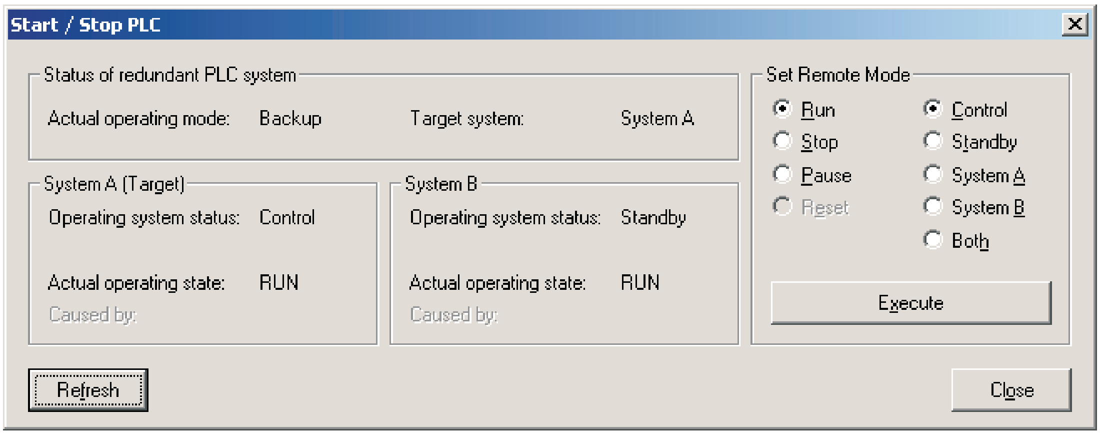
By clicking on the button **Update status** it is possible to update the current status of the forced input/output devices in the list. All forced I/Os registered to the PLC CPU can be cancelled by clicking on the button **Clear all** .

NOTE

The status of forced input or output devices is not modified until it is modified through this dialogue by the user.

9.4.2 Starting/Stopping a PLC

This function toggles the PLC CPU's RUN/STOP mode. To control SFCs in a project from other program parts, GX IEC Developer provides the following functions for the System Q/QnA CPUs: Start, stop, and initialise.



Start / Stop PLC

Status of redundant PLC system
Actual operating mode: Backup
Target system: System A

System A (Target)
Operating system status: Control
Actual operating state: RUN
Caused by:

System B
Operating system status: Standby
Actual operating state: RUN
Caused by:

Set Remote Mode
● Run
○ Stop
○ Pause
○ Reset
● Control
○ Standby
○ System A
○ System B
○ Both

Fig. 9-43: PLC status display options in case of a redundant PLC system

Status of redundant PLC system: Shows the status of the redundant PLC system.

System A: Shows the status information of System A.

System B: Shows the status information of System B.

Set Remote Mode

- Select to set the operating state of the PLC:
  Run, Stop, Pause, Reset
- Select to choose the target PLC for the remote mode change:
  Control, Standby, System A, System B, Both

Execute: Sets the operating state of the PLC

Refresh: Re-reads the operating state information from the PLCs.

Close: Closes the dialogue box.

How to start/stop a PLC

① Select the **Start/Stop PLC** command in the **Online** menu.

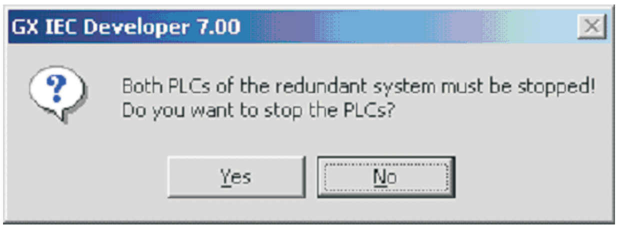
② Click on the desired mode in the right hand section of the dialogue window displayed.

③ In a redundant system click on the desired target PLC in the right hand section of the dialogue window displayed.

④ Confirm your selection with **Execute**.
You will be prompted for a confirmation when you attempt to switch from RUN to STOP mode. The current status of the PLC is shown in the left hand section of the dialogue box.

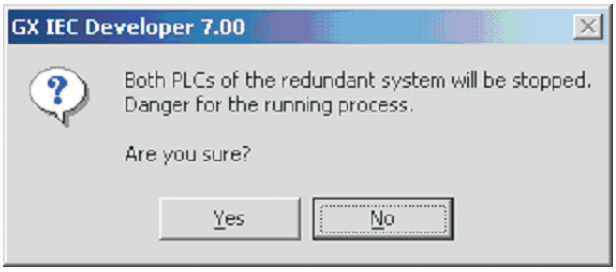
Stop PLC and Restart PLC during RUN mode

A Stop PLC dialogue is displayed when you attempt an operation in RUN mode that is only possible in STOP mode (e.g. Download to PLC). For the QnPRH PLCs in Backup mode, when both PLCs are involved in the operation, a dialogue is shown to inform you that both PLC need to be stopped.



Both PLCs of the redundant system must be stopped!
Do you want to stop the PLCs?

Fig. 9-44:
STOP PLC dialogue when downloading to PLC

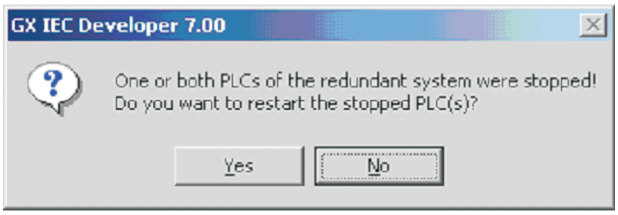


Both PLCs of the redundant system will be stopped.
Danger for the running process.

Are you sure?

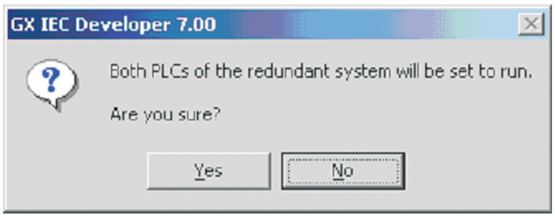
Fig. 9-45:
STOP PLC dialogue for all other functions

A Restart PLC dialogue is displayed after the PLC was stopped because of an operation that was only possible in STOP mode (e.g. Download to PLC). For the QnPRH PLCs in Backup mode, when both PLCs were involved in the operation, a dialogue is shown to inform you that both PLC will be restarted.



One or both PLCs of the redundant system were stopped!
Do you want to restart the stopped PLC(s)?

Fig. 9-46:
Restart PLC dialogue after Download to PLC



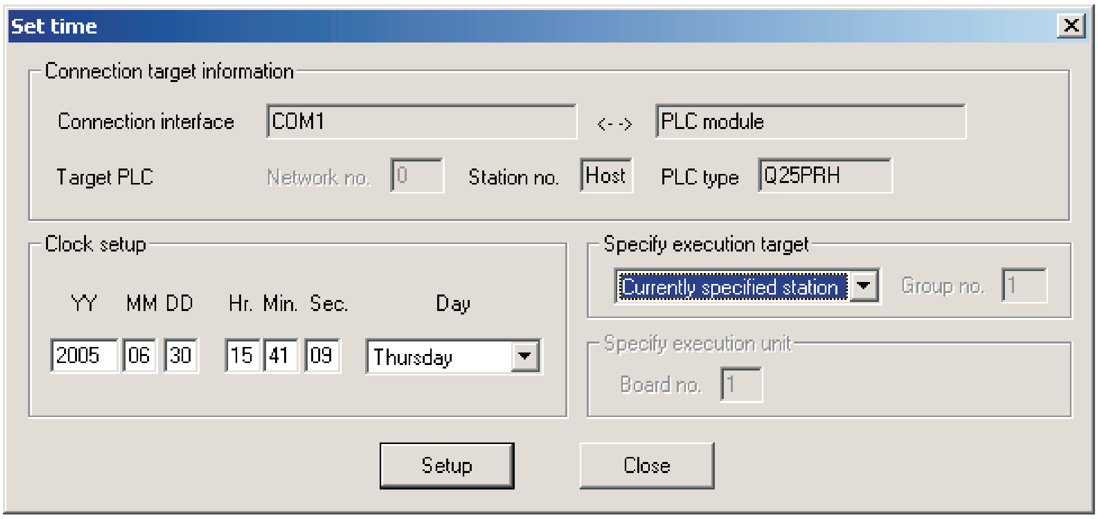
Both PLCs of the redundant system will be set to run.

Are you sure?

Fig. 9-47:
Restart PLC dialogue after other function

9.4.3 Setting the PLC Time

This function sets the time on the connected PLC or on all PLCs in a network.



Set time

Connection target information

Connection interface: COM1 <--> PLC module

Target PLC: Network no. 0, Station no. Host, PLC type Q25PRH

Clock setup

YY	MM	DD	Hr.	Min.	Sec.	Day
2005	06	30	15	41	09	Thursday

Specify execution target: Currently specified station, Group no. 1

Specify execution unit: Board no. 1

Fig. 9-48: *Set time dialogue*

Connection target information: Shows the actual connection.

Clock setup: Set date, time, and day.

Specify execution target: You can select between **Currently specified station**, **Specify all stations**, and **Specify group** in connection with the **Group no.** as the target for setting the internal clocks.

Specify execution unit: Set the board no.

Setup: Sets up the set time information.

Close: Closes the dialogue box.

9.5 The Monitor Function

The monitor function enables you to follow program execution and to check the status of operands/devices in runtime. The data for monitor mode is read from the PLC itself.

GX IEC Developer supports different monitoring modes:

- Program Monitor
- Entry Data Monitor
- Header Monitor
- Link Monitor (see page 5-11)

NOTE When the monitoring mode is activated all GX IEC Developer editing functions are disabled.

9.5.1 Inheritance

The monitor mode is inherited to objects that are opened by other objects they depend on. Therefore, the monitor mode does not need to be enabled for depending objects manually.

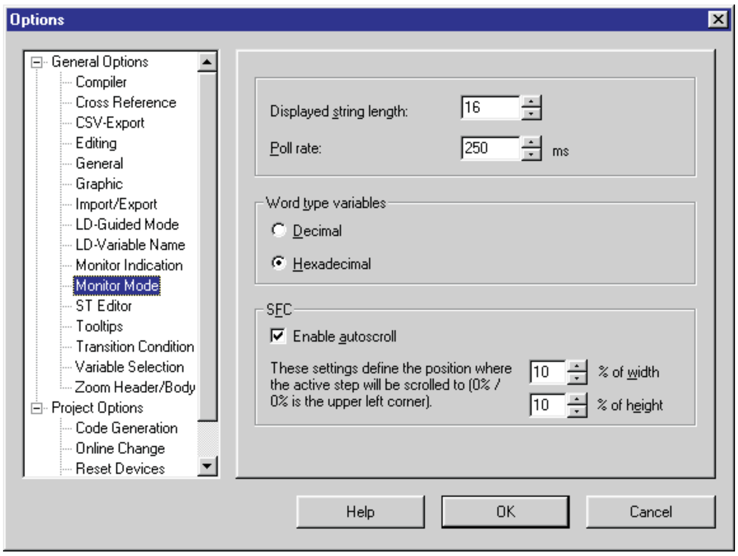
For the following actions the monitor mode will be inherited:

Action	Monitor mode inherited to
Change instance of a function block	New instance
Zoom function block	Body of function block
Zoom function	Body of function
Open transition from SFC body	Transition ● Transition elements of the body, if the transition is a variable or address ● Transition condition editor
Open action from SFC body	Action ● Action association list editor
Open new windows during tracking	New window

Tab. 9-14: Inheritance of monitor mode

9.5.2 Monitor Mode Options

In the **Extras** - **Options** menu you can specify options for the monitor mode.



Displayed string length: 16

Poll rate: 250 ms

Word type variables

○ Decimal

● Hexadecimal

SFC

☑ Enable autoscroll

These settings define the position where the active step will be scrolled to (0% / 0% is the upper left corner). 10 % of width 10 % of height

Fig. 9-49: Monitor mode options

Displayed String Length: Displaying long strings significantly reduces performance in monitoring mode. It is thus advisable to limit the number of characters to be displayed.

Pollrate: Interval at which the monitor polls the CPU, expressed in ms.

NOTE

While monitoring at a low poll rate and rapidly changing activation of actions/transitions the system may be slowed down.

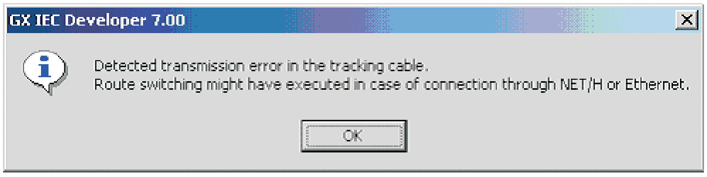
9.5.3 System Switching during Monitoring a Redundant System

If the control and standby systems are switched during monitoring, the monitoring continues automatically with the system selected in **Transfer Setup** → **Ports** (see section 8.3.1).

Transfer Setup setting	Physically connected	Monitoring before switch	Monitoring after switch
Not specified	System A (Control)	System A (Control)	System A (Standby)
	System B (Control)	System B (Control)	System B (Standby)
	System A (Standby)	System A (Standby)	System A (Control)
	System B (Standby)	System B (Standby)	System B (Control)
Control system	not relevant	System A (Control)	System B (Control)
		System B (Control)	System A (Control)
Standby system	not relevant	System A (Standby)	System B (Standby)
		System B (Standby)	System A (Standby)
System A	not relevant	System A (Control)	System A (Standby)
		System A (Standby)	System A (Control)
System B	not relevant	System B (Control)	System B (Standby)
		System B (Standby)	System B (Control)

Tab. 9-15: Switching between control and standby system

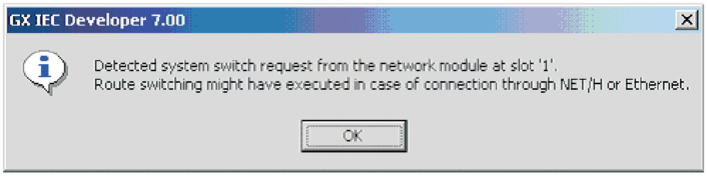
If the switch was caused by a transmission error in the tracking cable, the following message will be displayed:



Detected transmission error in the tracking cable.
Route switching might have executed in case of connection through NET/H or Ethernet.

Fig. 9-50: *Switching notification*

If the switch was caused by a request from a network module, the following message will be displayed:



Detected system switch request from the network module at slot '1'.
Route switching might have executed in case of connection through NET/H or Ethernet.

Fig. 9-51: *Switching notification*

9.5.4 The Program Monitor

In the text and graphical editors the values of all visible variables are displayed in the editing window, and they are updated after every program scan. The variables shown are always those lying within the current editing window pane. If you scroll the contents of the window or change its size the system automatically shows the values of all new variables that come into the window, and stops displaying the values of all variables that move out of the window.

Inputs and outputs that are set are displayed in the colours corresponding to the colour parameter settings.

When you start monitoring in the Sequential Function Chart editor the active Steps in the current window are shown in the colours set in the parameters. This enables you to follow which Step is currently active.

How to open the Program Monitor window

① Switch to Online mode by activating **Monitor Mode** in the **Online** menu or clicking on the monitor mode icon:

Tool for activating the monitor mode

② Open and select the body of the POU that you wish to switch to monitoring mode.

③ Select **Start Monitoring** in the **Online** menu.

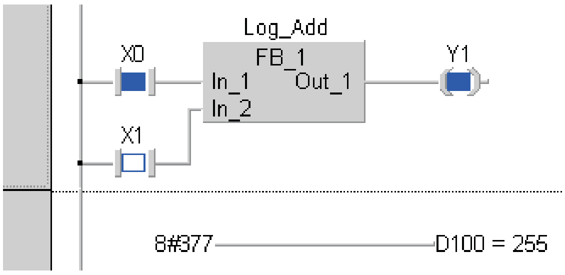


Fig. 9-52:
Active inputs and outputs are identified in colour.

Register D100 takes the value 255 of the base-8 literal.

In addition to Program POUs you can also use the Program Monitor for function blocks and functions. When you activate monitoring for the operand status of function blocks GX IEC Developer first prompts you to specify the Instance of the function block for which you wish to activate the Program Monitor.

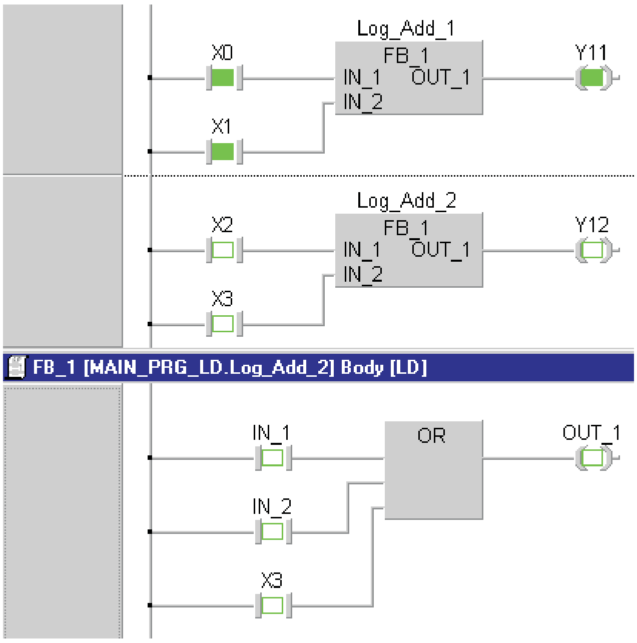


Fig. 9-53:
In the body of the program there is a signal on the inputs and outputs of Log_Add_1, which is an instance of function block FB_1.

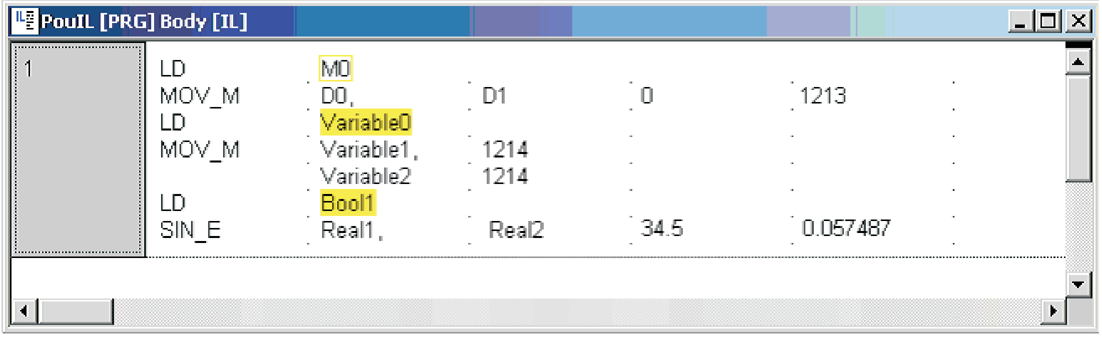
Log_Add_2 (an instance of function block FB_1) has no signals on its I/Os …

... as a result, no signals are displayed for the devices of function block FB_1 in Monitor mode.
In Monitor mode, there would be signals in the devices In_1, In_2 and Out_1 of FB_1.

How to start monitoring of an IL editor in a normal window

① Activate the command **Start Monitoring** → **Body** in the **Online** menu.

When Body Window monitoring is started, Boolean and all other types of variables will be monitored in the un-split body window. If more than one variable or device is used in one line, monitored values will be displayed at subsequent tab stops.



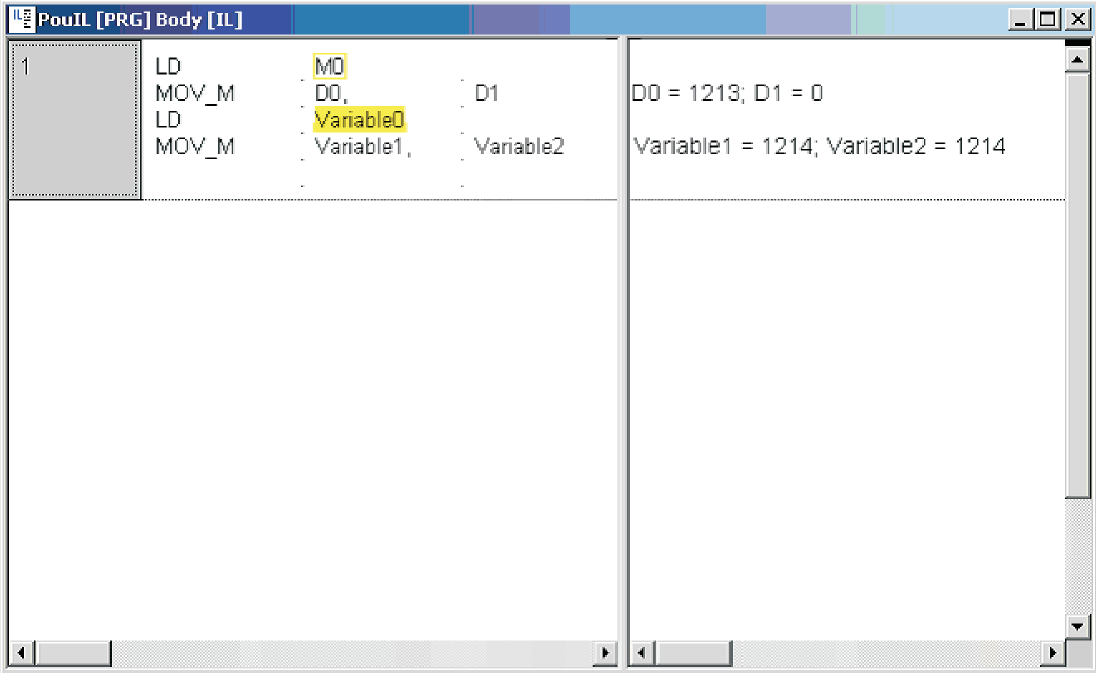
```
1    LD       M0
     MOV_M    D0,         D1       0       1213
     LD       Variable0
     MOV_M    Variable1,  1214
              Variable2   1214
     LD       Bool1
     SIN_E    Real1,      Real2    34.5    0.057487
```

Fig. 9-54: *Monitoring in an unsplit window*

How to start monitoring of an IL editor in a split window

① Activate the command **Start Monitoring** → **Split Window** in the **Online** menu.

When Split Window monitoring is started, the editor will be split into two vertical columns. Boolean values will be monitored as usual in the left split window. All other devices and variable types will be monitored in the right split window, as in the ST editor Split Window monitoring.



```
LD        M0
MOV_M     D0,         D1          D0 = 1213; D1 = 0
LD        Variable0
MOV_M     Variable1,  Variable2   Variable1 = 1214; Variable2 = 1214
```

Fig. 9-55: *Monitoring in a split window*

NOTE

When the monitoring of an IL body is not started by one of the above mentioned menu items (but e.g. with **Online** → **Monitoring Mode,** or via zooming into a FB body), **Split Window Monitoring** will be used.

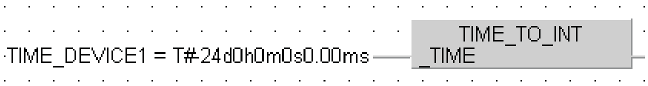
How to stop the Program Monitor window

① Activate the command **Stop Monitoring** in the **Online** menu.

NOTE

If you experience difficulties in distinguishing symbols that are closely spaced e. g. the “**-**” and the “**#**” symbols in the example below in the default **Arial** font setting, switch to another font such as **Courier New** or **Fixedsys** for better viewing.

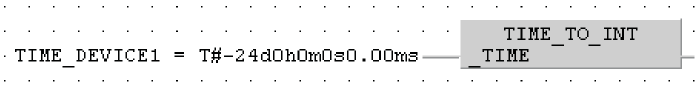
Display in Arial font



```
TIME_DEVICE1 = T#24d0h0m0s0.00ms —— TIME_TO_INT
                                    _TIME
```

Fig. 9-56:
*Display in **Arial** font*

Display in Courier New font



```
TIME_DEVICE1 = T#-24d0h0m0s0.00ms —— TIME_TO_INT
                                     _TIME
```

Fig. 9-57:
*Display in **Courier New** font*

Refer to section for details on the font selection.

9.5.5 The Entry Data Monitor (EDM) and the Header Monitor

The Entry Data Monitor can be used to group objects from the open project in order to obtain a display of their current status. Several windows can be opened at once. The Header Monitor essentially offers all the functions available in the Entry Data Monitor. The only difference, although an important one, is that calling up the Header Monitor automatically displays the objects of the active POU.

The status of the following objects can be monitored in the Entry Data Monitor: Global and Local Variables, POUs and Tasks.

The central element of the Data Entry Monitor is a table which the user can configure himself.

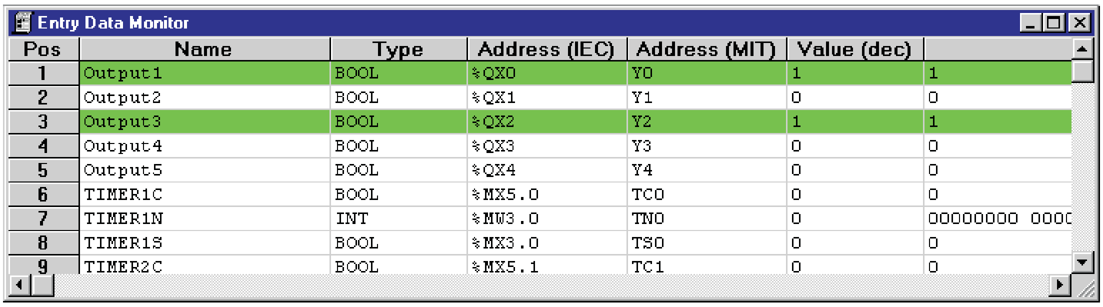
The values in the table can be read from the CPU and also written back to the CPU after editing. There is also the possibility of writing the table, i.e. the grouped objects, and the configuration of the table into an external file, or of reading the data from an external file.

The table is restricted to a maximum of 16,383 rows and 255 columns. This means that a total of 4,177,655 cells can be managed. The number of objects in the table is limited only by the available memory on your computer.

A security function is available to prevent data being written into the CPU.

The worksheet

In the **Online** menu, the **Entry Data Monitor** command calls up the EDM window.



Entry Data Monitor

Pos	Name	Type	Address (IEC)	Address (MIT)	Value (dec)	
1	Output1	BOOL	%QX0	Y0	1	1
2	Output2	BOOL	%QX1	Y1	0	0
3	Output3	BOOL	%QX2	Y2	1	1
4	Output4	BOOL	%QX3	Y3	0	0
5	Output5	BOOL	%QX4	Y4	0	0
6	TIMER1C	BOOL	%MX5.0	TC0	0	0
7	TIMER1N	INT	%MW3.0	TN0	0	00000000 0000
8	TIMER1S	BOOL	%MX3.0	TS0	0	0
9	TIMER2C	BOOL	%MX5.1	TC1	0	0

Fig. 9-58: *Entry Data Monitor worksheet*

If a direct address (in the IEC and Mitsubishi Editor) was entered in lower-case letters, these characters will automatically be converted into upper-case letters.

Rows including Boolean objects are highlighted if the respective object is active. The colour is defined in the project settings.

By means of this option it is possible to monitor an object in the Mitsubishi *Kn* notation.

Example ▽ K1M10 will monitor the relays M10 to M13.

The result is displayed in the result columns offering a suitable style, e.g. Value (bin), Value (dec), Value (hex). The objects will be displayed in their respective representations.

△

Example ▽ Bit pattern of M0 to M7 is displayed as 1000 1101.

In the column Value (bin) it will be displayed as 1000 1101.
In the column Value (dec) it will be displayed as 141.
In the column Value (hex) it will be displayed as 8D.

In the Mitsubishi *Kn* notation object values cannot be changed.

△

NOTE The commands in the **Edit** and **Tools** menus can also be called up via the pop-up menu. The pop-up menu appears if you press the right-hand mouse button or press the key combination Shift F10.

The Edit menu

The following commands are available in the Edit menu when the window of the Data Entry Monitor is active:

Insert Object: The Object List window appears containing all the objects available in the current project. The objects can be selected via a group filter and a type ahead input field.

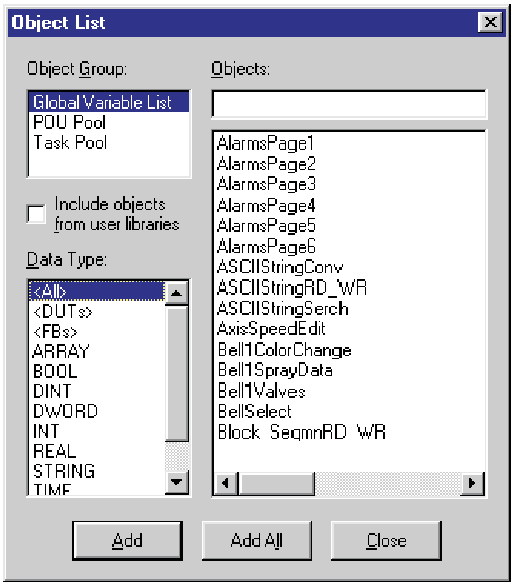


Fig. 9-59:
Inserting an object from the Object List
*If the checkbox **Include objects from user libraries** is checked, global variables and POUs in any installed user libraries are shown in the object list.*

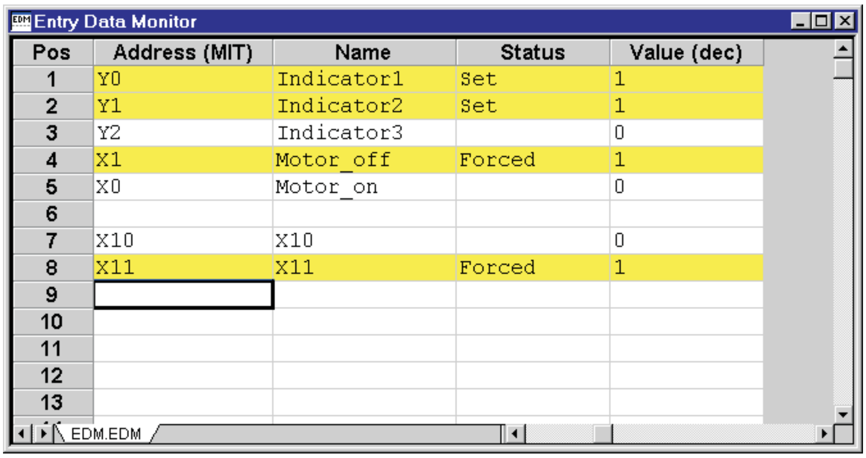
Next Object: The object following the currently selected object is inserted. If the selection marker is positioned on an Array entry, the next element of this Array is inserted. If the selection marker is positioned on a numbered entry, the element with the next number is inserted. In all other cases, the object with the next consecutive address is inserted. This command is only available if the selected row contains an elementary object and the selected object is not a sub-object.

Insert Forced Inputs (not for FX family and Q00(J), Q01 CPUs): Inserts all user-forced input devices into the EDM. The "**Status**" column indicates that they are "**Forced**" (see below). Forced inputs can be cleared via **Online - PLC Clear - Forced Inputs**.

Insert Set Inputs (FX family and Q00(J), Q01 CPUs only): Inserts all user-set input devices into the EDM. The "**Status**" column indicates that they are "**Set**" (see below).

]

Insert Set Outputs: Inserts all user-set outputs into the EDM. The “**Status**” column indicates that they are “**Set**” (see below).



Pos	Address (MIT)	Name	Status	Value (dec)
1	Y0	Indicator1	Set	1
2	Y1	Indicator2	Set	1
3	Y2	Indicator3		0
4	X1	Motor_off	Forced	1
5	X0	Motor_on		0
6				
7	X10	X10		0
8	X11	X11	Forced	1
9				
10				
11				
12				
13				

Fig. 9-60: *Indication of the output status in the Entry Data Monitor*

Clear Device File: All forced or set inputs and outputs are stored in the *DeviceFile.MON* in the project path. This file is cleared.

Insert Row: A new row is inserted in front of the selected row. As the EDM does not offer copy and paste operations, you cannot insert several rows at once.

Delete Rows: The selected row is deleted. You can also delete several rows at once. This command is only available if the first object selected and the object after the last object selected are not sub-objects.

Delete All: All entries in the table and the table configuration are deleted. You are asked to confirm the command before it is executed.

Close: The EDM window is closed. All information is automatically saved. The information concerning the current settings is stored in the MMP701.INI file and is loaded as the default when the EDM is next called up.

The Tools menu

The following commands are available in the **Tools** menu when the window of the Entry Data Monitor is active:

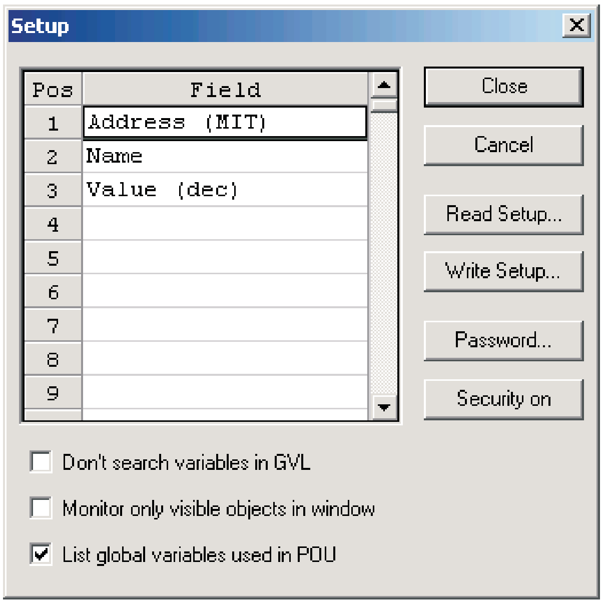
Read from PLC: The data are read from the PLC and displayed in the EDM table. This command is only available in Offline mode.

Write to PLC: The data from the EDM table are written to the PLC. A dialogue window appears which allows you to choose whether only the selected objects or all objects are to be transmitted. This command is only available in Offline mode.

Read from File: The file previously created with the Write to File command is called up. This command is only available in Offline mode.

Write to File: Both the grouped objects and the configuration of the table can be saved in an external file. This file has the file extension EDM. This command is only available in Offline mode.

Setup: The Setup dialogue window is opened, in which the EDM table can be configured and the security function activated.



Pos	Field
1	Address (MIT)
2	Name
3	Value (dec)
4	
5	
6	
7	
8	
9	

□ Don't search variables in GVL

□ Monitor only visible objects in window

☑ List global variables used in POU

Fig. 9-61: ***

The columns for the EDM table are compiled in consecutive rows.

Disable automatic search

If you enter a direct address into the Entry Data Monitor, the system searches automatically for a corresponding variable in the GVL of the project. If a variable has been found, the variable name is added to the Entry Data Monitor.

This feature can cause long delays if the projects include many global variables. To increase the performance of GX IEC Developer, disable the automatic search in the setup of the Entry Data Monitor.

Monitor only visible objects in window

In the monitor mode all objects entered in the **Entry Data Monitor** are monitored, even if they are not visible. This feature can cause long delays if the window contains many EDM elements. To increase the performance of GX IEC Developer, it is possible to monitor only visible objects in the Entry Data Monitor. This option is realised in the setup of the Entry Data Monitor:

The disadvantage of this option is that short delays may be caused during scrolling in the Entry Data Monitor. These delays are caused by the registration, which is changed every time when the visible part of the window has been changed.

Always on top: Keeps the **Entry Data Monitor** always on top of other windows. This option can be enabled or disabled in the menu **Tools** or **Context**. If this option is active, a check mark is placed next to the menu item. Only one Entry Data Monitor can be on top.

List global variables used in POU

This function in the EDM Setup is used to switch the listing of global variables used in a POU on and off. When the setting is off, only local variables (those declared in the POU header) will be listed under the POU. When the setting is on, the list will also contain all global variables used in the POU’s body, actions and transitions.

The pop-up menu for configuring the table

The Shortcut menu appears if you select the desired row and click on the right-hand mouse button or press the key combination Shift F10.

Field list: The Field List contains the possible column entries for the EDM table.

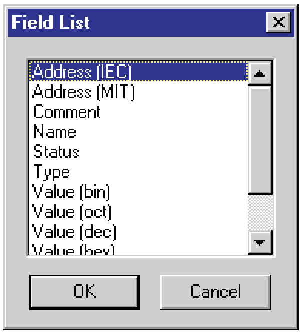


Fig. 9-62:
Field list

Insert row: A new row is inserted in front of the selected row.

Delete row: The selected row is deleted.

Delete All: All entries are deleted. You are asked to confirm your choice before the command is executed.

The buttons for configuring the table

Close: The dialogue window is closed. The settings are saved.

Cancel: The operation is aborted. The settings are not saved.

Read Setup: The file previously created with the Write Setup is called up.

Write Setup: The table configuration can be stored in an external file. This file is given the extension SED.

Password: The password for the security function is defined here.

Security on/off: A security function is available to prevent data being written into the CPU. The security function can be switched on or off. You are prompted for the password.

How to open the Entry Data Monitor or the Header Monitor

① Activate the **Monitor Mode** command in the **Online** menu.

② Select the POU to be switched to Monitor Mode.

③ Select the **Entry Data Monitor** or **Header Monitor** command in the **Online** menu. The dialogue window containing the EDM table appears.

How to insert an object

① Select any cell in the desired row. A selected cell is indicated by the double border. (If the **Name** or **Address** column is selected, you can now enter the name or address directly.)

② Select the **Insert Object** command from the **Edit** menu or from the pop-up menu.

③ Select the desired object and click on **OK**. You can also select several objects in the list - consecutively or selectively. You also have the possibility of transferring the entire list of Global Variables, as well as the entire POU and Task pool.

or

- Copy global declarations from a GVL into an EDM window (Cut, Copy and Paste or Ctrl+X, Ctrl+C and Ctrl+V).
- Copy entries within an EDM window (Cut, Copy and Paste or Ctrl+X, Ctrl+C and Ctrl+V).
- Copy entries between two EDM windows (Cut, Copy and Paste or Ctrl+X, Ctrl+C and Ctrl+V).

How to insert a 32-bit register

① As an IEC address: Enter the corresponding address in the Address (IEC) column, e.g. %MD0.0.

As a MITSUBISHI address: Enter the two contiguous MITSUBISHI addresses in the Name column, e.g. D3+D4.

How to change the values of the objects

① Select the desired values column.

② In the case of bit operands: You can switch between the two possible entries (0 and 1) by double-clicking or pressing the SPACE key.

In the case of word operands: Activate the Edit Mode (mouse click or m). Enter the desired value. GX IEC Developer only accepts valid entries. Pay attention to the correct format (decimal, octal, etc.). The Edit Mode remains active in the event of an incorrect entry.

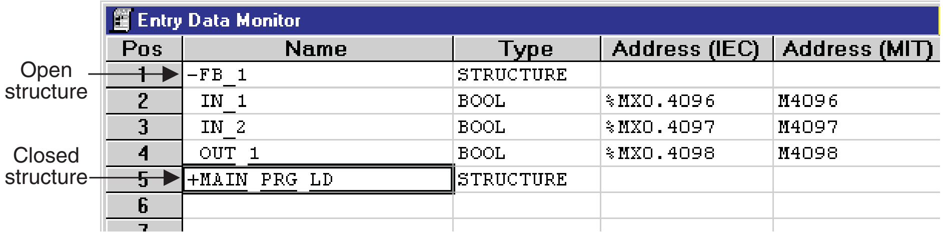
How to open and close a structure

The word **Structure** in the **Type** column indicates a tree structure. If a tree structure is open, all the associated variables of the POU are displayed.

① Select the Name cell in the row containing a structure.

② Closing an open structure: The open structure is indicated by a minus sign in front of the name. Double-click on the **Name** cell or press the Ctrl - key combination. This closes the tree structure.

Opening a closed structure: The closed structure is indicated by a plus sign in front of the name. Double-click on the **Name** cell or press the Ctrl + key combination. This opens the tree structure.



Pos	Name	Type	Address (IEC)	Address (MIT)
1	-FB_1	STRUCTURE		
2	IN_1	BOOL	%MX0.4096	M4096
3	IN_2	BOOL	%MX0.4097	M4097
4	OUT_1	BOOL	%MX0.4098	M4098
5	+MAIN_PRG_LD	STRUCTURE		
6				
7				

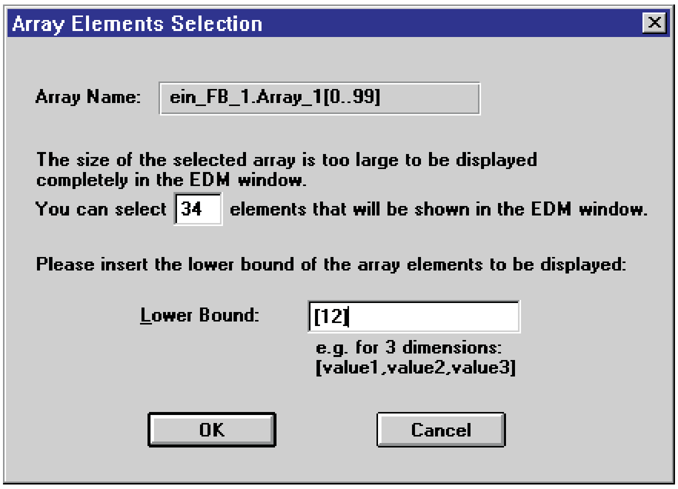
Fig. 9-63: Open and closed structures in the Entry Data Monitor

Arrays in the Entry Data Monitor

You can enter arrays in the Entry Data Monitor. All elements of the array can be read, written and set.

If the array is large it may not be possible to display all its elements on the screen at the same time. You can set the default number of elements to display to a maximum of 50 with the option **Extras** - **Options** - **General** - **Array Elements To Show**.

You can also change this value in the prompt dialogue that is displayed when you open an array in the Entry Data Monitor.



Array Name: ein_FB_1.Array_1[0..99]

The size of the selected array is too large to be displayed completely in the EDM window.
You can select 34 elements that will be shown in the EDM window.

Please insert the lower bound of the array elements to be displayed:

Lower Bound: [12]
e.g. for 3 dimensions:
[value1,value2,value3]

Fig. 9-64: Array Elements Selection

The entry in the **Lower Bound** field specifies the first array element to be displayed.

9.5.6 Changing variable values

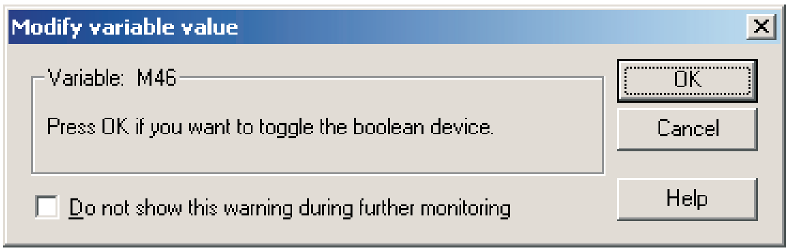
The value of a variable can be changed in the Entry Data Monitor (EDM). Additionally the variable value can be changed directly in an editor (IL, MELSEC IL, Ladder, FBD, and SFC). This is realised by means of new dialogues for simple data types, which can be opened in the monitor mode. The restriction to simple data types means that it is not possible to change the values of structured data types (e.g. DUTs).

Dialogue layouts

Three different dialogue layouts are available for changing a variable value in an editor:

a) Dialogue to toggle Boolean variables or devices

This dialogue appears, if you want to change a Boolean variable or device. This dialogue can be switched off by selecting the correspondent option in the dialogue. This setting is valid until you leave the monitor mode.


Modify variable value

Variable: M46

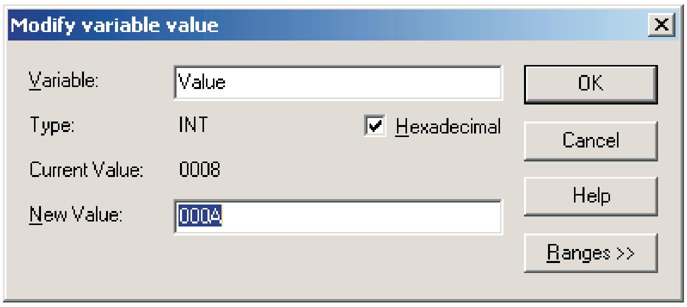
Press OK if you want to toggle the boolean device.

□ Do not show this warning during further monitoring

Fig. 9-65: *Modifying boolean devices*

b) Dialogue to change numerical variables or devices

This dialogue appears, if you want to change a numerical variable or device. It contains the type and the current value of the variable or device. The new value can be entered either as a decimal or as a hexadecimal value.


Modify variable value

Variable: Value

Type: INT

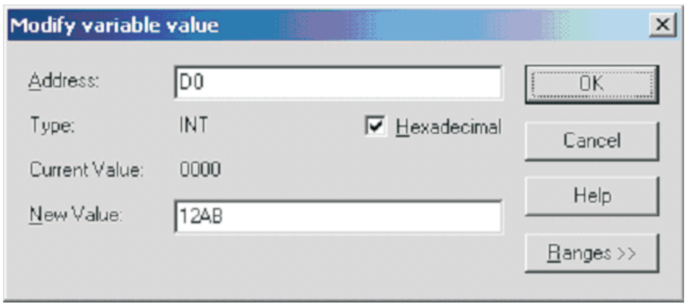
☑ Hexadecimal

Current Value: 0008

New Value: 000A

Fig. 9-66: *Modifying numerical variables*

For a Q Remote I/O module the Modify variable value functionality can only be used to modify device values, since a Q Remote I/O project does not have symbolic names (there are neither GVLs nor POU headers). Because of this, the dialogue is different for the Q Remote I/O module.



Address: D0

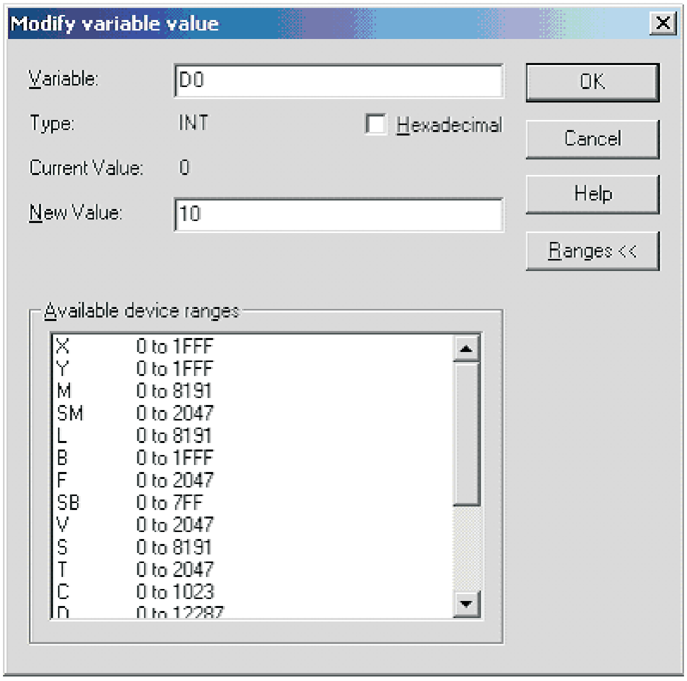
Type: INT ☑ Hexadecimal

Current Value: 0000

New Value: 12AB

Fig. 9-67:
Modifying numerical variables for a Q Remote I/O module

By clicking on **Ranges** you get shown a list of valid device ranges for the current PLC.



Variable: D0

Type: INT □ Hexadecimal

Current Value: 0

New Value: 10

Available device ranges

X	0 to 1FFF
Y	0 to 1FFF
M	0 to 8191
SM	0 to 2047
L	0 to 8191
B	0 to 1FFF
F	0 to 2047
SB	0 to 7FF
V	0 to 2047
S	0 to 8191
T	0 to 2047
C	0 to 1023
D	0 to 12287

Fig. 9-68:
Modifying numerical variables - Ranges

NOTE

The devices LB, LW, LX, LY cannot be monitored in GX IEC Developer, therefore they will not be displayed in the **Available device ranges** list of the **Modify Variable Value** dialogue.

Dialogue operations

There are the following dialogue operations:

- Changing the value of a variable or device

  After you have entered the new value in the edit control, press **Enter** or **OK** to leave the dialogue. A message will appear asking whether the value should be written to the PLC. If the new value has a wrong format, a message in the dialogue appears and the value in the edit control is selected. In this case the value is not written to the PLC. For numerical data types (INT, WORD, DWORD, DINT), the value can be entered in the hexadecimal format.

- Changing the name of a variable or device

  After you have entered a new variable or device and left the corresponding edit control, the dialogue will be updated. If the variable can be monitored, the new name and the current value are displayed. The cursor is set to the edit control **New Value**.

  The following objects are valid:

  – global variables

  – local variables which are declared in the POU of the active body

  – direct addresses

- Changing the display type

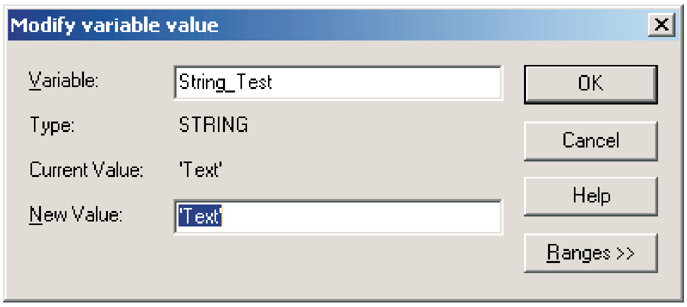
  For numerical data types it is possible to switch the display format of the current value and the new value between decimal or hexadecimal. This option is disabled for Boolean variables or devices.

- Updating the current value

  The current value of the variable or device will be updated according to the poll rate of the monitor mode.

c) Dialogue to change string variables

This dialogue appears, if you want to change a string variable. It contains the type and the current value (character string) of the variable or device. The new value (character string) can be entered in the field **New Value**.



Modify variable value

Variable: String_Test

Type: STRING

Current Value: 'Text'

New Value: 'Text'

Fig. 9-69: *Modifying numerical variables*

Opening the dialogues

The dialogues to change the variable or device values can be opened in the monitor mode in the following ways for the corresponding windows:

a) SFC Body

① Choose **Modify variable** in the menu **Online** or use the shortcut **[CTRL] + [F9]**.
If a transition with a variable or device is selected in the POU body, the corresponding dialogue with this variable or device is opened. If no transition with a variable or device is selected, the dialogue is empty and you can enter a variable or device.

② Double-click with the left mouse button on a transition with a variable or device. The corresponding dialogue with this variable or device is opened.

Example ▽

For a selected transition with the variable *OperationDone*:

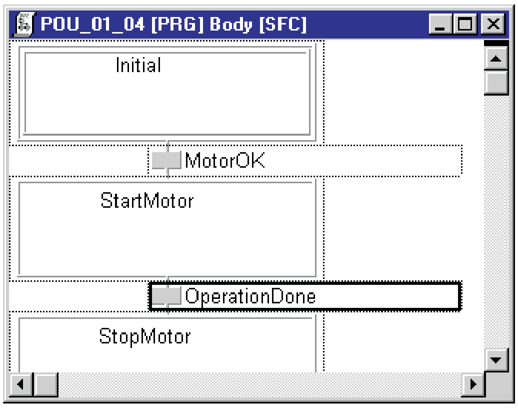


Fig. 9-70:
Opened dialogue indicating respective variable or device

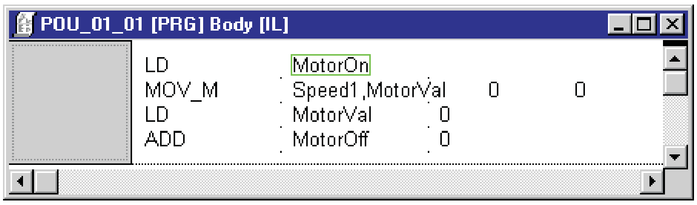
△

b) MELSEC IL / IL Body

① Choose **Modify variable i**n the menu **Online** or use the shortcut **[CTRL] + [F9]**.
If a variable or device is selected in the POU body, the corresponding dialogue with this variable or device is opened. If no variable or device is selected, the dialogue is empty and you can enter a variable or device.

② Double-click with the left mouse button on a variable or device. The corresponding dialogue with this variable or device is opened.

Example ▽ In IL for the selected variable *MotorVal*:



POU_01_01 [PRG] Body [IL]

```
LD          MotorOn
MOV_M       Speed1 ,MotorVal    0       0
LD          MotorVal      0
ADD         MotorOff      0
```

Fig. 9-71:
Opened dialogue indicating respective variable or device

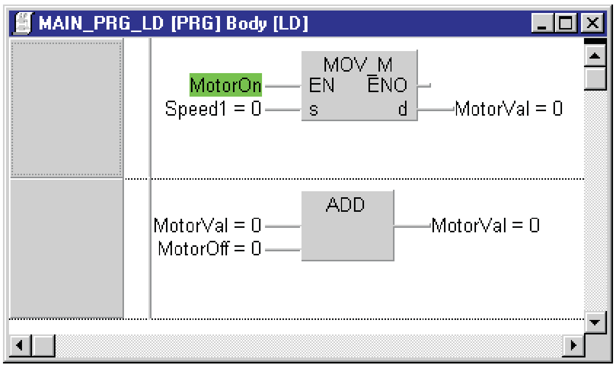
△

c) Ladder / FBD Body

① Choose **Modify variable** in the menu **Online** or use the shortcut **[CTRL] + [F9]**.
If a variable or device is selected in the body of the POU, the corresponding dialogue with this variable or device is opened. If no variable or device is selected, the dialogue is empty and you can enter a variable or device.

② Double-click with the left mouse button on a variable or device. The corresponding dialogue is opened with this variable or device.

Example ▽ Example in Ladder for the selected variable *MotorVal*:



MAIN_PRG_LD [PRG] Body [LD]

Fig. 9-72:
Opened dialogue indicating respective variable or device

△

d) Navigator

① To open the dialogue select **Modify variable** in the menu **Online** or press the shortcut **[CTRL] + [F9]. The dialogue is empty and you can enter a variable or device.**

9.5.7 Display direct address of a variable

In the monitor mode the direct address instead of the symbolic name of global variables can be displayed. This is possible for global variables of simple data types in Ladder and FBD.

Toggle direct address and symbolic name

In the monitor mode direct address and symbolic name of a variable can be toggled.

① Select the symbolic name or direct address of a global variable and choose **Toggle variable** in the menu **Online** or press the shortcut **[ALT] + [F9]**.

② Press **[CTRL]** and double-click with the left mouse button on the symbolic name or direct address of the global variable.

Example ▽ Example in Ladder

A) Global variable *MotorVal* before calling the toggle functionality:

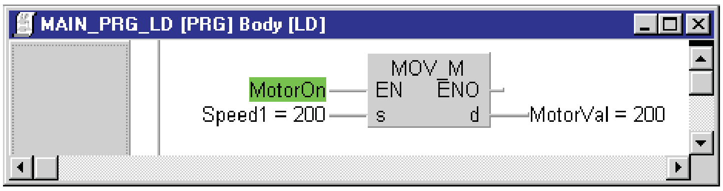

MAIN_PRG_LD [PRG] Body [LD]

Fig. 9-73:
Symbolic name is indicated

B) Global variable *MotorVal* after the toggle functionality has been called:

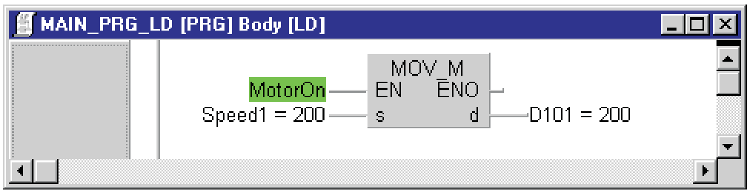

MAIN_PRG_LD [PRG] Body [LD]

Fig. 9-74:
After toggling direct address is indicated

If the monitoring is stopped for a window, the direct addresses of the toggled global variables are still available. If the monitor mode is finished, the symbolic names of the toggled global variables are restored.

△

9.5.8 Tracking variables

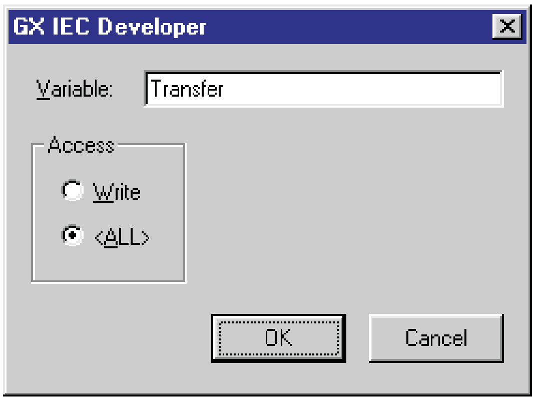
The write access to variables can be tracked during monitoring. The tracking of variables is available within the IL, MELSEC IL, ST, LD, and FBD editors for the following objects:

- POU bodies
- SFC transitions
- SFC actions

The tracking functions are available from

- the menu **Tools - Tracking**
- the editor context menu (right-click on variable or keys **[SHIFT]+[F12]**)

The tracking function is available if the monitor mode is online or offline. If the tracking function is started, a dialogue to enter the variable name and the access type (to search only for variables with write access or for all variables) appears. The tracking function is started, when the dialogue is closed with the OK button.



GX IEC Developer

Variable: Transfer

Access

○ Write

● <ALL>

Fig. 9-75:
Tracking

The tracking function can be started in the menu bar, if the monitor functionality is switched off or switched on (not context sensitive).
If the monitor mode is switched on, the tracking functionality is also available in the context menu. The name of the variable, where the right mouse click was done, is used as the default variable name in the appearing dialogue.

The **Start Tracking** function creates an internal list (from the browser information). This internal list indicates where the selected variable name is used for write access. If this list already exists, it will be overwritten. A window will be opened showing where the variable name is used for the first time. In this window the corresponding variable is highlighted. If the corresponding variable name can not be found or if the browser information is not available, an appropriate message box will appear.

If the variable name could be found, the **next**, **previous**, **first**, and **last** occurrence of the variable can be tracked via menu commands. Each occurrence is shown in a window where the variable is highlighted. If the corresponding variable name can not be found or if the browser information is not available, an appropriate message box will appear.

Tracking menu items and functions

Start Tracking:

- Open the browser.
- The code location selected is the first entry in the browser list box.
- Open the code location of the occurrence.
- Mark the code location of the variable occurrence.
- Start monitoring of the newly opened editor.
- If no ‚First‘ is found, the following message box is displayed: ‘Start Tracking: No reference found’.

Next:

- According to the browser information the selection is set to the next occurrence.
- Open the code location of the occurrence.
- Mark the code location of the variable occurrence.
- Start monitoring of the newly opened editor.
- If no *Next* is found, the following message box is displayed: “No reference found”.

Previous:

- According to the browser information the selection is set to the previous occurrence.
- Open the code location of the occurrence.
- Mark the code location of the variable occurrence.
- Start monitoring of the newly opened editor.
- If no ‚Previous‘ is found, the following message box is displayed: “Previous: No reference found”.

First:

- According to the browser information the selection is set to the first occurrence within the browser.
- Open the code location of the occurrence.
- Mark the code location of the variable occurrence.
- Start monitoring of the newly opened editor.
- If no *First* is found, the following message box is displayed: “Already placed on desired location”.

Last:

- According to the browser information the selection is set to the last occurrence within the browser.
- Open the code location of the occurrence.
- Mark the code location of the variable occurrence.
- Start monitoring of the newly opened editor.
- If no *Last* is found, the following message box is displayed: “Already placed on desired location”.

NOTES

If the .scx file is not available and the .sct files are also not available, a dialogue will appear with the information, that the project must be checked or compiled with activated cross reference options. A button to call the corresponding options dialogue is also available.

If the .sct files are not available, a dialogue will appear with the information, that the project must be checked or compiled with activated cross reference options. A button to call the corresponding options dialogue is also available.

If the .scx file is not available and the .sct files are available, a dialogue will appear with the information, that the cross reference must be generated.

Limitations

If a variable is used in the MELSEC IL, the variable can not be displayed by the tracking function.

9.5.9 Tracking Variables in a Redundant System

If default tracking settings are used, most system variables will automatically be tracked. There is, however, a problem if the device R is selected as the Word range of system variables because File Registers are not automatically tracked.

It is possible to set tracking ranges manually in the **Redundant PLC Parameter** dialogue (also see sec. 5.2). In this case only the devices set by the user are tracked. Therefore the system variables will not be tracked if the user does not specifically set all system variable ranges to be tracked.

The following table describes the default tracking ranges and the possible tracking ranges that may be set in the **Redundant PLC Parameter**.

Device	Default tracking range	Possible tracking range	Can be used as system variable
Input	X0 to X1FFF		○
Output	Y0 to Y1FFF		○
Internal Relay	M0 to M8191 [1]		●
Latch Relay	L0 to L8191 [1]		○
Step Relay	S0 to S8191		●
Annunciator	—	F0 to F2047 or the range set in **PLC Parameter** → **Devices**	○
Edge Relay	V0 to V2047 [1]		○
Link Relay	B0 to B1FFF [1]		○
Link Special Relay	—	SB0 to SB7FF	○
Timer	T0 to T2047 [1]		●
Retentive Timer	—		●
Counter	C0 to C1023 [1]		●
Data Register	D0 to D12287 [1]		●
Link Register	W0 to W1FFF [1]		●
Link Special Register	—	SW0 to SW7FF	○
Index Register	Z0 to Z15		○
Special Relay	See QnPRH user manual chapter 5.5.3 (2) (c)	Not necessary; default range is always tracked in backup mode	○
Special Register	See QnPRH user manual chapter 5.5.3 (2) (d)	Not necessary; default range is always tracked in backup mode	○
File Register	—	The range set in **PLC Parameter** → **PLC File**, if any	●

Tab. 9-16: *Default tracking ranges*

●: Can be used as a System Variable

○: Can not be used as a System Variable

[1] If the number of devices is modified in **PLC Parameter** → **Devices**, both the default and the possible tracking ranges change accordingly.

The following checks will be carried out, depending on the settings of the Redundant Parameter and the System Variables options:

If default tracking settings are used, the whole range of the following devices that may be used in system variables are tracked by default: D, W, T, ST, C, M, S.
It is not necessary to track P devices.

- A warning dialogue is displayed if R devices are used as system variables.

If tracking ranges are set manually, then it is necessary to check all used system variable ranges against the set tracking ranges.

- A warning dialogue is displayed if not all devices used as system variables are tracked.
- A warning dialogue is displayed if all devices used as system variables are tracked, but in several different tracking blocks.
- A warning dialogue is displayed if all devices used as system variables are tracked in one tracking block, to inform the user how to make sure that the block is actually tracked.

These checks will be executed during project check. If a problem is found, a warning will be displayed in the **Compile/Check Messages** window. Please refer to section 8.2 for details.

If the situation is not corrected, a warning dialogue will be displayed when attempting to download the project.

9.5.10 Monitoring Boolean I/Os of function blocks

Boolean inputs and outputs of function blocks are monitored even if they are connected to a network. The function block itself does not need to be opened and monitored individually.

All Boolean inputs and outputs of a function block are monitored in the function block box, if the monitor mode is active. The Boolean variables of the value FALSE are indicated by a highlighted frame and the Boolean variables of the value TRUE are indicated by a completely highlighted rectangle.

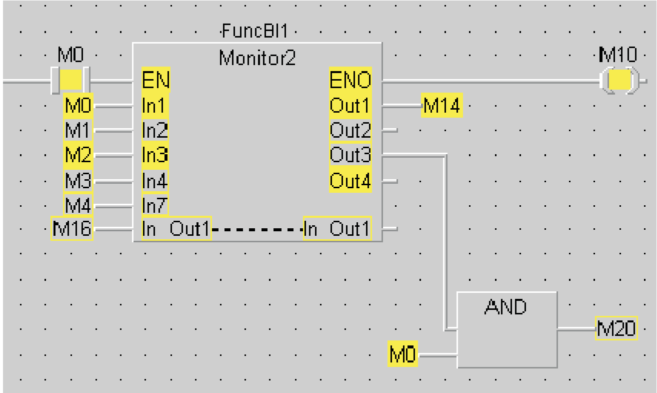


Fig. 9-76:
Monitored inputs and outputs

9.6 In Cycle Monitor

In GX IEC Developer monitoring can be done at any valid step of the sequence program.

When doing „normal“ monitoring, the monitor values reflect the status of the monitored devices at the end of the scan cycle. When doing in cycle monitoring, the monitor values reflect the status of the monitored devices **previous** of executing the designated step.

9.6.1 Running the In Cycle Monitor

Two entries in the Online-menu allow starting and stopping of the In Cycle Monitor.

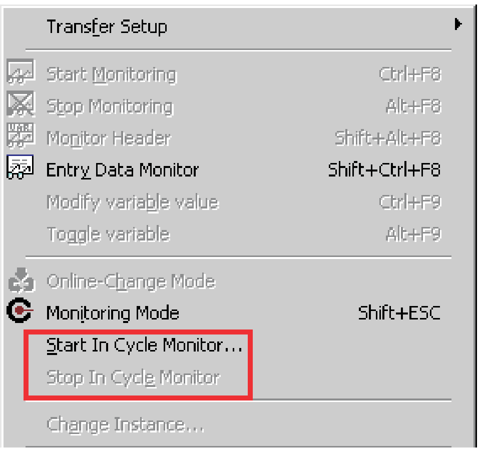


Fig. 9-77:
Menu command for starting the In Cycle Monitor

The following conditions must be fulfilled to start the In Cycle Monitor:

- A project must be open
- The project’s PLC type is Q or QnA series
- The PLC must be in RUN mode
- Transfer Setup designates PLC Module or Net/10(H) Module as the PLC side I/F, and in the Network route there is no specification or Net/10(H) selected
- The monitor mode is off
- The online change mode is off
- The GX Simulator is not running

NOTES

In Cycle Monitor is not available for SFC POUs (only the main program file MAIN.QPG is supported).

In Cycle Monitor is not available for the standby system of a redundant system in Backup mode. A corresponding message will be shown if an attempt for monitoring is made.

The In Cycle Monitor is not available for the A and FX series.

It is not possible to use “normal” monitoring for one window and in cycle monitoring for another. Once monitoring is started in In Cycle Monitor mode all monitoring is done at the designated step and vice versa.

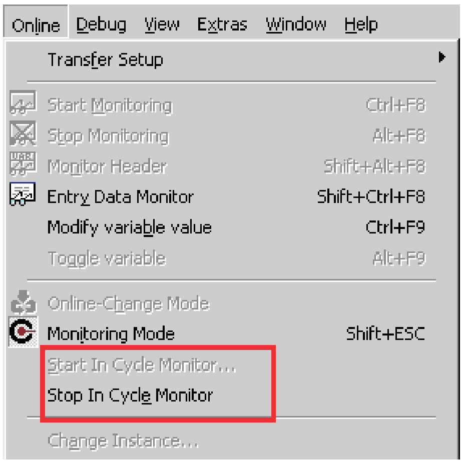
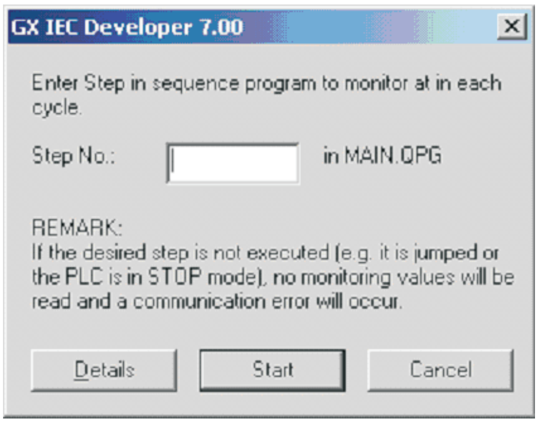

Online Debug View Extras Window Help
Transfer Setup
Start Monitoring Ctrl+F8
Stop Monitoring Alt+F8
Monitor Header Shift+Alt+F8
Entry Data Monitor Shift+Ctrl+F8
Modify variable value Ctrl+F9
Toggle variable Alt+F9
Online-Change Mode
Monitoring Mode Shift+ESC
Start In Cycle Monitor...
Stop In Cycle Monitor
Change Instance...

Fig. 9-78:
Menu command for stopping the In Cycle Monitor

The „Stop In Cycle Monitor“ menu command will also stop monitor mode and vice-versa, if the command „Stop Monitoring“ mode is issued during In Cycle Monitoring, In Cycle Monitoring is stopped automatically.

If you are in In Cycle Monitor mode, but would like to monitor at another step, then first you will have to stop monitoring and start In Cycle Monitor again.

After starting the In Cycle Monitor, a dialogue box for input of the step number will be displayed:


GX IEC Developer 7.00

Enter Step in sequence program to monitor at in each cycle.

Step No.: in MAIN.QPG

REMARK:
If the desired step is not executed (e.g. it is jumped or the PLC is in STOP mode), no monitoring values will be read and a communication error will occur.

Fig. 9-79:
Entry of the desired step number

The step number is pre-initialised, if the In Cycle Monitor is started from one of the IL, MELSEC IL or ST body editors and from the current editor position a step can be calculated. If for the current line no step number can be calculated, the previous lines are tried, until a step number can be calculated or the beginning of the network is reached.

The step number is also pre-initialised, if the In Cycle Monitor is started from one of the IL, MELSEC IL, LD or FBD body editors and a network is selected. In this case the first step after the network is used for pre-initialisation. This means that the status after the network is executed is monitored.

If a step number cannot be calculated the dialogue is opened with the Step No. field empty.

The Step No. field cannot be initialised if

- the actual window is not an IL, MELSEC IL, LD, FBD or ST body editor
- the body is not compiled
- an empty network is selected
- more than one network is selected

NOTE

If the designated step is not executed (e.g. the step is jumped or the PLC is in STOP mode), no monitoring values can be read and a communication error will occur.

How to calculate the step number to input?

With In Cycle Monitoring, the monitor values shows the status of the monitored devices previous of executing the designated step. This means, that the actions of the step set as the In Cycle Monitoring step, do not effect the monitor values.

If you would like to monitor the results of a step, for In Cycle Monitoring you must set the result of the addition
„starting step number of the desired instruction“ + „length of the desired instruction“.

For convenience, the Tools -> Show MELSEC Code Of Networks functionality can be used, where the start step of each instruction is displayed (in brackets at the end of each line).

Example ▽

The following example shows 12 steps of code, generated for a Q06H PLC. The code snippet occupies step 14 through 26. After the code lines the start step number is written in brackets.

Every LD instruction for the Q06H PLC occupies 1 step each, whereas a MOV instruction occupies 2 steps.

```
LD       SM400      (*   15 *)
MOV      K1111      (*   16 *)
         D1

LD       SM400      (*   18 *)
MOV      K2222      (*   19 *)
         D1

LD       SM400      (*   21 *)
MOV      K3333      (*   22 *)
         D1

LD       SM400      (*   24 *)
MOV      K4444      (*   25 *)
         D1
```

Fig. 9-80:
Program sequence for the example

In these 12 steps of PLC program the value of the device D1 is changed four times, by four different MOV instructions.

The value of the device D1 is not changed elsewhere in the PLC program.

If you would monitor the device D1 in „normal“ monitor mode, the value would be 4444, as this is the last setting.

As an example, if you would like to monitor the result of the MOV instruction at step 22, you must add 2 (length of MOV instructions) to the start step (22). The result is 24, which is also the start step of the following instruction, of course.

For the in cycle monitoring mode the following table shows the monitored values corresponding to the set in cycle monitor step:

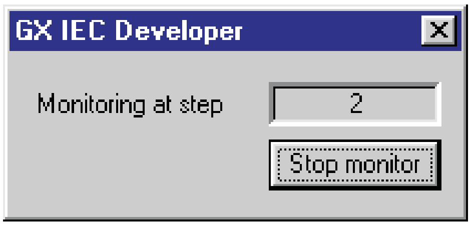
Step No.	Monitored value of D1
… 14	4444
15 or 16	4444
18 or 19	1111
21 or 22	2222
24 or 25	3333
27 …	4444

Tab. 9-17:
Monitored value for the example

△

Before in cycle monitoring is started the validity of the step is checked both in the project and on the PLC. If the step is found invalid or the program on the PLC does not have the same instruction at the same step, in cycle monitoring is aborted. A step is considered valid for in cycle monitoring, if it is in the range of the program and it is the start step of any instruction. In the above example, steps 17, 20, 23 and 26 would not be accepted.

During the operation of the In Cycle Monitor a dialogue box informs you about the selected step number:

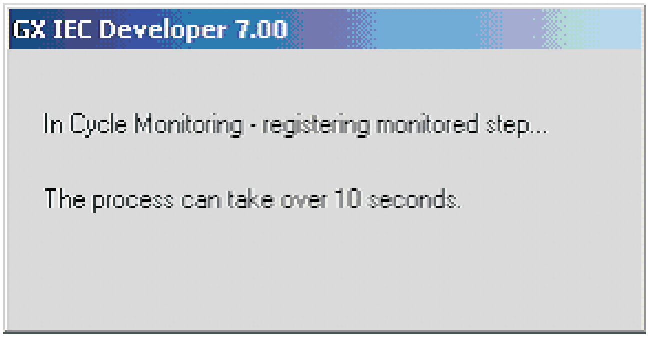


Monitoring at step 2

Fig. 9-81:
Display of the monitored step no.

If the step selected for In Cycle Monitoring is invalid or not executed by the PLC, it can take longer than the communication time-out setting for the PLC to return an error, and afterwards to terminate In Cycle Monitoring.

A status dialogue is displayed when In Cycle Monitoring is started or terminated, until the process is finished. The estimated time depends on the communication time-out setting.



In Cycle Monitoring - registering monitored step...

The process can take over 10 seconds.

Fig. 9-82:
Status dialogue

9.7 Buffer Memory Monitoring

Using the function of **Buffer Memory Batch** in the menu **Debug** you can monitor the current values of the buffer memory of a special function module attached with the active PLC. The monitoring can be started and stopped from the buffer memory monitor window. For displaying the monitored values you can select different user defined monitor, display and value formats. The formats can be changed by clicking the radio buttons in the function dialogue box.

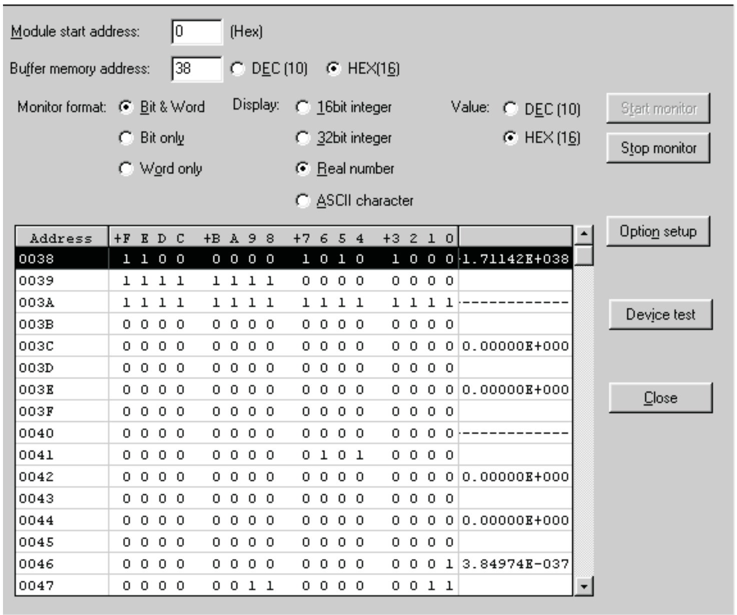
NOTES

The function of **Buffer Memory Batch** is not supported by the PLCs of the FX series.

GX Simulator (section 9.8) does not support buffer memory monitoring.

When using the function **Buffer Memory Batch** some menu items like **Object**, **Edit** and **Tools** are not available.

When selecting **Buffer Memory Batch** in the **Debug** menu the following dialogue box will be displayed.



Module start address: 0 (Hex)

Buffer memory address: 38 ○ DEC (10) ● HEX(16)

Monitor format: ● Bit & Word ○ Bit only ○ Word only

Display: ○ 16bit integer ○ 32bit integer ● Real number ○ ASCII character

Value: ○ DEC (10) ● HEX (16)

Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0038	1 1 0 0	0 0 0 0	1 0 1 0	1 0 0 0	1.71142E+038
0039	1 1 1 1	1 1 1 1	0 0 0 0	0 0 0 0	
003A	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	------------
003B	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
003C	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0.00000E+000
003D	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
003E	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0.00000E+000
003F	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0040	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	------------
0041	0 0 0 0	0 0 0 0	0 1 0 1	0 0 0 0	
0042	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0.00000E+000
0043	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0044	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0.00000E+000
0045	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0046	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	3.84974E-037
0047	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1	

Fig. 9-83:
*Dialogue box **Buffer memory batch monitor***

Module start address: Enter here the I/O address (lowest digit not required) of the special function module whose buffer memory you want to be monitored. The module start address should be input in hexadecimal format.

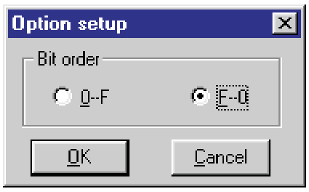
Buffer memory address: Here the start address of the buffer memory block that is to be monitored has to be set. The address can be in either decimal or hexadecimal format according to the value format option selected (see section 9.7.3).

Monitor format: Please see sections 9.7.1 to 9.7.3.

Start monitor: Click here to start the buffer monitor function.

Stop monitor: Click here to stop the buffer monitor function.

Option setup: Clicking on this button opens a dialogue box to select the bit order for displaying the buffer memory monitor values.

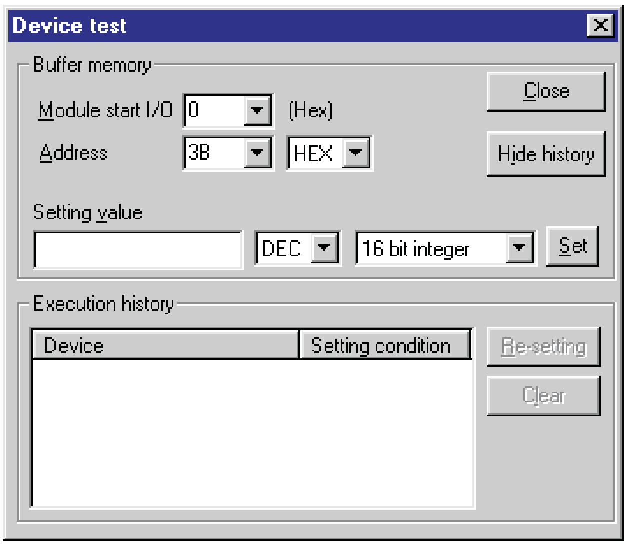


Option setup

Bit order: ○ 0--F ● F--0

Fig. 9-84:
*Dialogue box **Option setup***

Device test: Clicking on this button opens a dialogue box for device test.



Device test

Buffer memory

Module start I/O: 0 (Hex)

Address: 3B HEX

Setting value: DEC, 16 bit integer

Execution history

Device	Setting condition

*Fig. 9-85: Dialogue box **Device test***

Module start I/O: In this pull-down list select the start I/O address of the module in which the buffer memory device for testing is available.

Address: Select a buffer memory address for testing from this pull-down list.

Close: Clicking on this button closes the dialogue.

Setting value: Here you can input the value to be set in the selected buffer memory address. By clicking on the **Set** button the input value is set to the selected buffer memory address.

Hide history: Clicking on this button shows/hides the device test history panel in the Device test dialogue.

Execution history: This history list shows the device tests made using the **Device test** dialogue. Here the devices and their setting values are displayed.

Re-setting: Sets the selected device and its associated value from the history panel to the **Setting value** box. To finally set the value in buffer memory you have to click on the **Set** button.

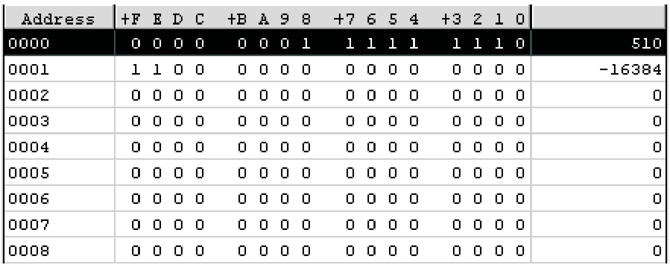
Clear: This button deletes all data of the history.

9.7.1 Monitor Formats

You can select one of the following formats for buffer memory monitoring.

- Bit & Word

  The buffer memory can be monitored by having displayed both Bits and Words as shown below.

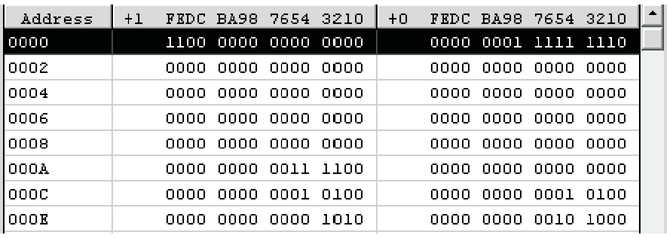


Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 1	1 1 1 1	1 1 1 0	510
0001	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	-16384
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0005	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0006	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0007	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0008	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0

Fig. 9-86: *Monitoring option **Bit & Word***

- Bit only

  The buffer memory can be monitored by having displayed only Bits as shown below.

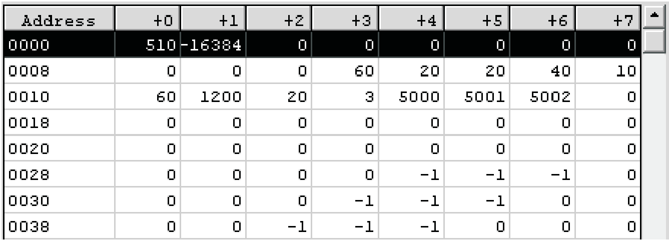


Address	+1 FEDC BA98 7654 3210	+0 FEDC BA98 7654 3210
0000	1100 0000 0000 0000	0000 0001 1111 1110
0002	0000 0000 0000 0000	0000 0000 0000 0000
0004	0000 0000 0000 0000	0000 0000 0000 0000
0006	0000 0000 0000 0000	0000 0000 0000 0000
0008	0000 0000 0000 0000	0000 0000 0000 0000
000A	0000 0000 0011 1100	0000 0000 0000 0000
000C	0000 0000 0001 0100	0000 0000 0001 0100
000E	0000 0000 0000 1010	0000 0000 0010 1000

Fig. 9-87: *Monitoring option **Bit only***

- Word only

  The buffer memory can be monitored by having displayed only Words as shown below.



Address	+0	+1	+2	+3	+4	+5	+6	+7
0000	510	-16384	0	0	0	0	0	0
0008	0	0	0	60	20	20	40	10
0010	60	1200	20	3	5000	5001	5002	0
0018	0	0	0	0	0	0	0	0
0020	0	0	0	0	0	0	0	0
0028	0	0	0	0	-1	-1	-1	0
0030	0	0	0	-1	-1	-1	0	0
0038	0	0	-1	-1	-1	0	0	0

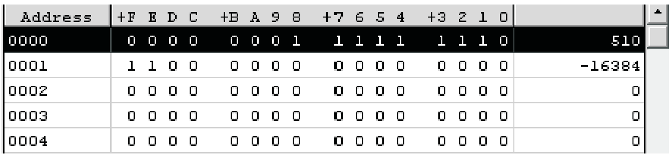
Fig. 9-88: *Monitoring option **Word only***

9.7.2 Monitor Display

When you have chosen one of the monitor format options **Bit & Word** or **Word only** you can additionally select one of the following display options for the buffer memory monitoring value. This option only applies to the monitoring values in Word format.

- 16-Bit-Integer

  The monitoring values will be displayed in 16-Bit-Integer.

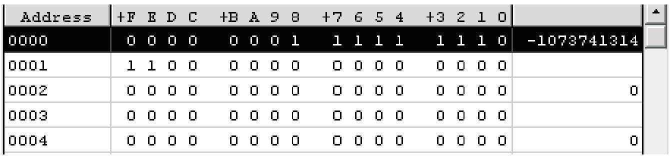


Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 1	1 1 1 1	1 1 1 0	510
0001	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	-16384
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0

Fig. 9-89: *Monitoring option **16bit integer***

- 32-Bit-Integer

  The monitoring values will be displayed in 32-Bit-Integer.

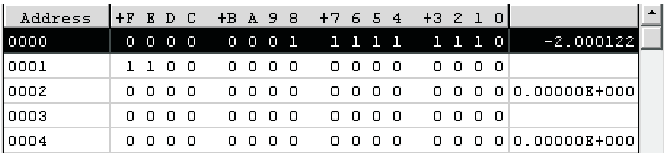


Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 1	1 1 1 1	1 1 1 0	-1073741314
0001	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0

Fig. 9-90: *Monitoring option **32bit integer***

- Real number

  The monitoring values will be displayed as Real numbers.

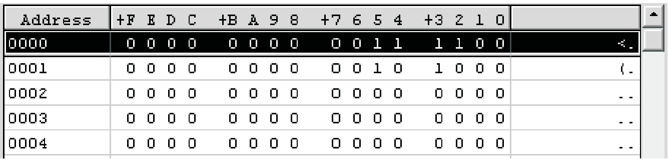


Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 1	1 1 1 1	1 1 1 0	-2.000122
0001	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0.00000E+000
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0.00000E+000

Fig. 9-91: *Monitoring option **Real number***

- ASCII character

  The monitoring values will be displayed as ASCII characters.



Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 0	0 0 1 1	1 1 0 0	<.
0001	0 0 0 0	0 0 0 0	0 0 1 0	1 0 0 0	(.
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	..
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	..
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	..

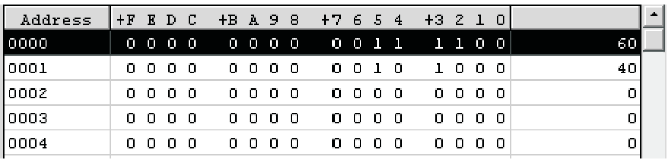
Fig. 9-92: *Monitoring option **ASCII character***

9.7.3 Monitor Value

When you have chosen one of the monitor format options **Bit & Word** or **Word only** you can additionally select one of the following value options for the buffer memory monitoring value. This option only applies to the monitoring values in Word format.

- Decimal

  The monitoring values will be displayed in decimal format.

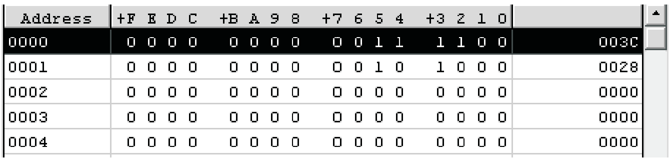


Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 0	0 0 1 1	1 1 0 0	60
0001	0 0 0 0	0 0 0 0	0 0 1 0	1 0 0 0	40
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0

Fig. 9-93: *Monitoring option **DEC (10)***

- Hexadecimal

  The monitoring values will be displayed in hexadecimal format.



Address	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
0000	0 0 0 0	0 0 0 0	0 0 1 1	1 1 0 0	003C
0001	0 0 0 0	0 0 0 0	0 0 1 0	1 0 0 0	0028
0002	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000
0003	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000
0004	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000

Fig. 9-94: *Monitoring option **HEX (16)***

9.8 GX Simulator

The GX Simulator is a utility for offline simulation and debugging of PLC programs. It provides the simulation of external devices, load and save buffer memory contents, timing charts and much more.

NOTE Step execution is also supported even for an FX CPU, except the FX3U CPU.

GX Simulator can only be started, if the tool is installed, the project is compiled, and GX IEC Developer is not in online-mode.

In the **Online** menu select **GX Simulator**. The PLC parameters are transferred to GX Simulator and the main GX Simulator window appears. It shows the PLC CPU condition and allows to start different tools.

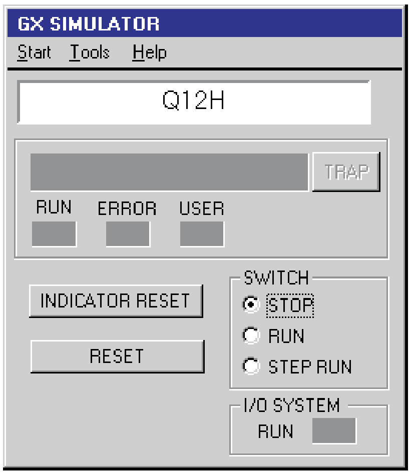
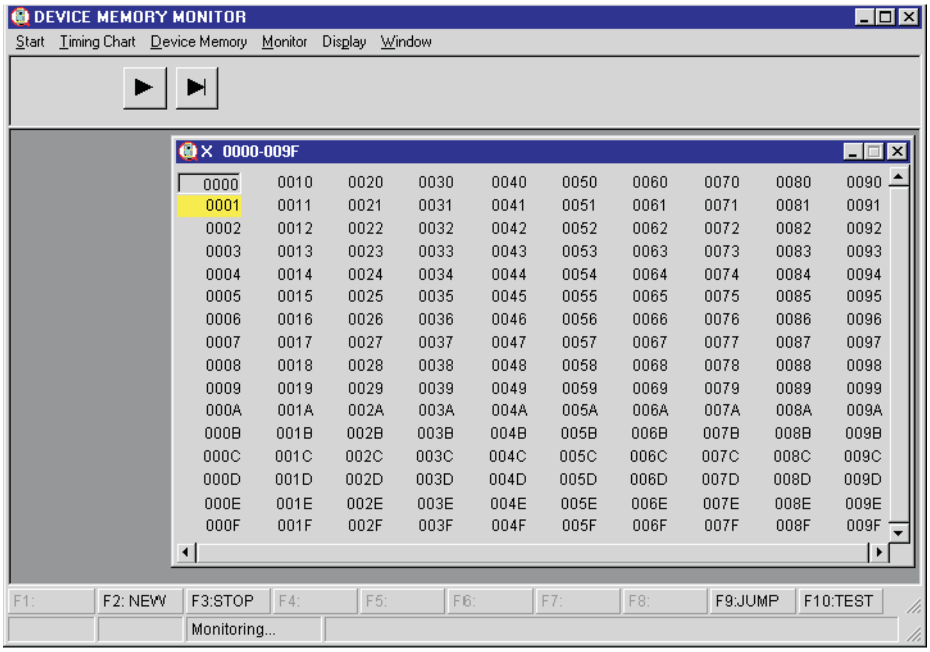


Fig. 9-95:
GX Simulator main window

The device memory monitor of the GX Simulator shows the values of the selected devices and allows to change them.



0000	0010	0020	0030	0040	0050	0060	0070	0080	0090
0001	0011	0021	0031	0041	0051	0061	0071	0081	0091
0002	0012	0022	0032	0042	0052	0062	0072	0082	0092
0003	0013	0023	0033	0043	0053	0063	0073	0083	0093
0004	0014	0024	0034	0044	0054	0064	0074	0084	0094
0005	0015	0025	0035	0045	0055	0065	0075	0085	0095
0006	0016	0026	0036	0046	0056	0066	0076	0086	0096
0007	0017	0027	0037	0047	0057	0067	0077	0087	0097
0008	0018	0028	0038	0048	0058	0068	0078	0088	0098
0009	0019	0029	0039	0049	0059	0069	0079	0089	0099
000A	001A	002A	003A	004A	005A	006A	007A	008A	009A
000B	001B	002B	003B	004B	005B	006B	007B	008B	009B
000C	001C	002C	003C	004C	005C	006C	007C	008C	009C
000D	001D	002D	003D	004D	005D	006D	007D	008D	009D
000E	001E	002E	003E	004E	005E	006E	007E	008E	009E
000F	001F	002F	003F	004F	005F	006F	007F	008F	009F

Fig. 9-96: *Device Memory Monitor*

The timing chart shows the condition of the selected devices within a selected timing range.

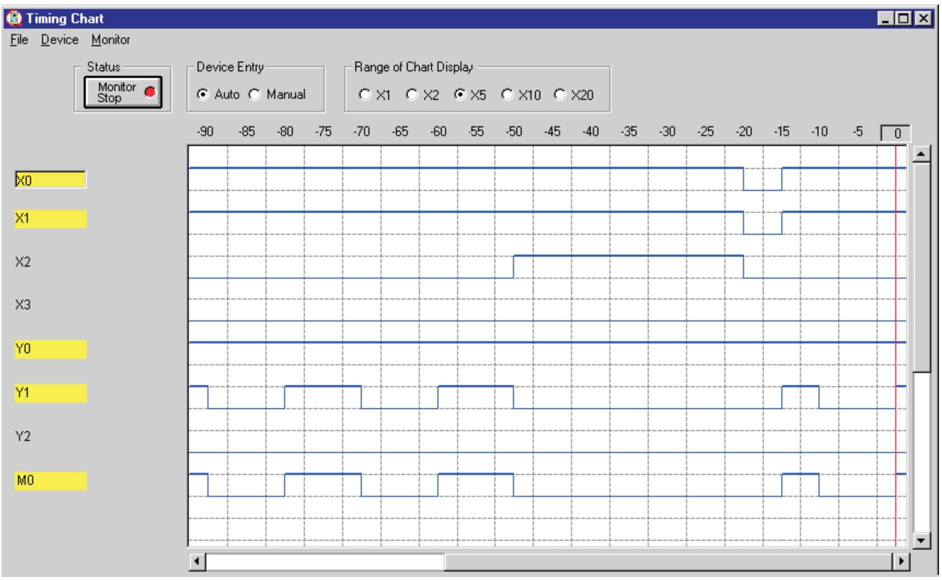


Fig. 9-97: *Timing Monitor*

It is not possible to perform online changes. Rounding errors may occur in the results of instructions using the floating decimal point.

NOTE

After the error correction by the GX Simulator program module the program must be transferred to the specified PLC and checked again before it is executed.

The following functions are not supported:

- Physical special function modules
- Physical I/O modules
- Networks
- Interrupt programs
- Read from PLC, Compare with PLC
- Comments
- SFC programs

There are other CPU related restrictions in device ranges, PLC memory format, etc. For more details refer to the manual for the GX Simulator.

9.9 Intelligent Function Module Utility

The included intelligent function module utility has to be installed separately. It supports an easy configuration of intelligent function modules by dialogue based input instead of PLC program code needed to access the module buffer memory setup area.

The utility is available only if a Q series project is active and the utility is installed. The utility supports the Q series only. For details refer to the documentation of the utility.

GX IEC Developer supports the call of the intelligent function module utility and a check for address collisions between refresh parameters for the modules and GX IEC Developer system addresses. The configuration file has to be written by the intelligent function utility to the PLC and to the hard disk. A corresponding message is displayed by GX IEC Developer if the utility is started.

If initial setting and automatic refresh setting are performed, the intelligent function module parameters require a maximum of 76 bytes per module.

If the project is saved with a different name, GX IEC Developer informs the utility about a closed project because the iparam.wpa file of the old project path is not saved with the new project! The iparam.wpa file is the configuration file, created by the utility. It can be found in the subdirectory *Resource* of the current project. You have to close the utility, and copy the iparam.wpa file from the old project to the new one.

NOTES

After configuration of the intelligent function module, please save the parameters to hard disk!

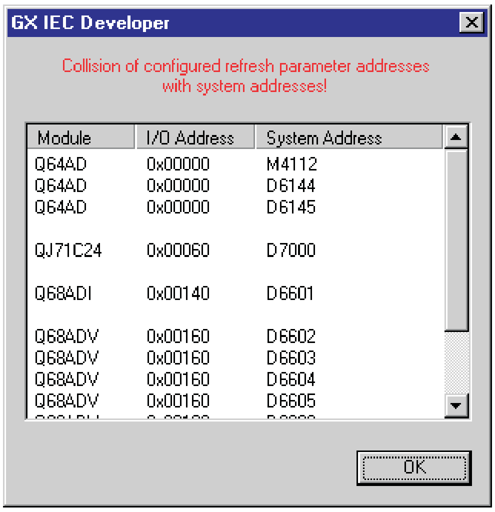
The intelligent function module utility will be closed, if the current project is closed or changed. Please remember to save the configuration BEFORE closing or changing the project.

9.9.1 Starting the utility

From the **Extras** menu select **Intelligent Function Utility - Start** to start the external utilities and navigate via the utility's menus.

From the Extras menu select **Intelligent Function Utility - Verify System Variables** to start the external utilities and directly verify the system variables.

Address collisions between refresh addresses and GX IEC Developer system addresses.



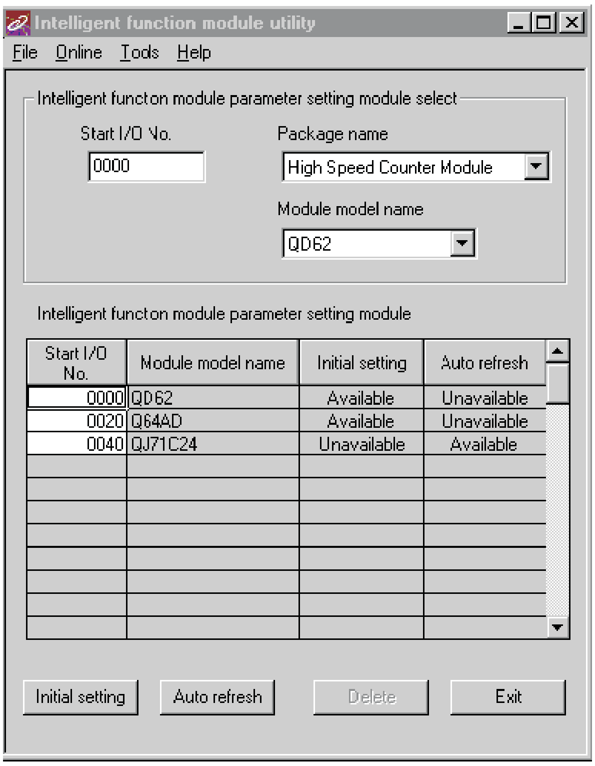
GX IEC Developer

Collision of configured refresh parameter addresses with system addresses!

Module	I/O Address	System Address
Q64AD	0x00000	M4112
Q64AD	0x00000	D6144
Q64AD	0x00000	D6145
QJ71C24	0x00060	D7000
Q68ADI	0x00140	D6601
Q68ADV	0x00160	D6602
Q68ADV	0x00160	D6603
Q68ADV	0x00160	D6604
Q68ADV	0x00160	D6605

Fig. 9-98:
Notified address collisions

Intelligent function module utility with configuration for three different modules.



Intelligent function module utility

Intelligent functon module parameter setting module select

Start I/O No.: 0000

Package name: High Speed Counter Module

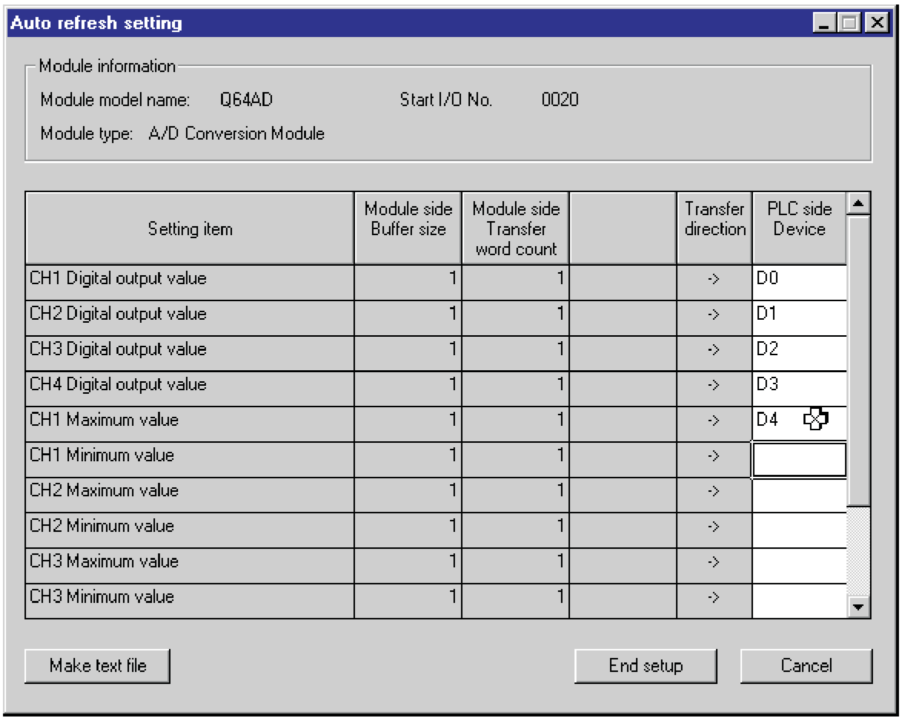
Module model name: QD62

Intelligent functon module parameter setting module

Start I/O No.	Module model name	Initial setting	Auto refresh
0000	QD62	Available	Unavailable
0020	Q64AD	Available	Unavailable
0040	QJ71C24	Unavailable	Available

Fig. 9-99:
Intelligent function module utility

Example ▽ Example for the auto refresh configuration of the A/D conversion Module Q64AD.



Auto refresh setting

Module information

Module model name: Q64AD Start I/O No. 0020

Module type: A/D Conversion Module

Setting item	Module side Buffer size	Module side Transfer word count		Transfer direction	PLC side Device
CH1 Digital output value	1	1		->	D0
CH2 Digital output value	1	1		->	D1
CH3 Digital output value	1	1		->	D2
CH4 Digital output value	1	1		->	D3
CH1 Maximum value	1	1		->	D4
CH1 Minimum value	1	1		->	
CH2 Maximum value	1	1		->	
CH2 Minimum value	1	1		->	
CH3 Maximum value	1	1		->	
CH3 Minimum value	1	1		->	

Fig. 9-100: *Sample auto refresh settings*

△

9.9.2 Troubleshooting

In very rare cases problems might occur which will be solved in future software versions.

① **Problem**: If GX IEC Developer 7.01 is already installed before installing GX Developer (Version < 7.01), the Intelligent Function Utility can not be started anymore from GX IEC Developer 7.01.
Solution: Install GX IEC Developer 7.01 again.

② **Problem**: If GX Developer (Version < 7.01) is already installed before installing GX IEC Developer 7.01, GX Developer can not be started anymore.
Solution: Install GX Developer again in a newer version (Version ≥ 7.01).

③ **Problem**: If GX Developer as well as GX IEC Developer 7.01 are installed and GX Developer will be uninstalled afterwards, the Intelligent Function Utility can not be started anymore from GX IEC Developer 7.01.
Solution: Install GX IEC Developer 7.01 again.

10 Working with Projects of Version 6.11 and before

This chapter describes major changes in the new version of the GX IEC Developer compared to earlier versions. So reading this chapter is important for users who want to use projects with the GX IEC Developer 7.01 that have been created by using the GX IEC Developer 6.11 or an earlier version. Please also see section 10.4.

10.1 External Variables are no longer used

Instead of using external variables, every global variable now can be directly accessed from any POU in the new version 7.01 of GX IEC Developer.
This means that in a POU body, action or transition every variable will be accessible that is declared either global or local in the respective scope. A global variable declaration can either take place in the Global Variable List of the project or in any user library.

10.1.1 Using Global Variables in Bodies

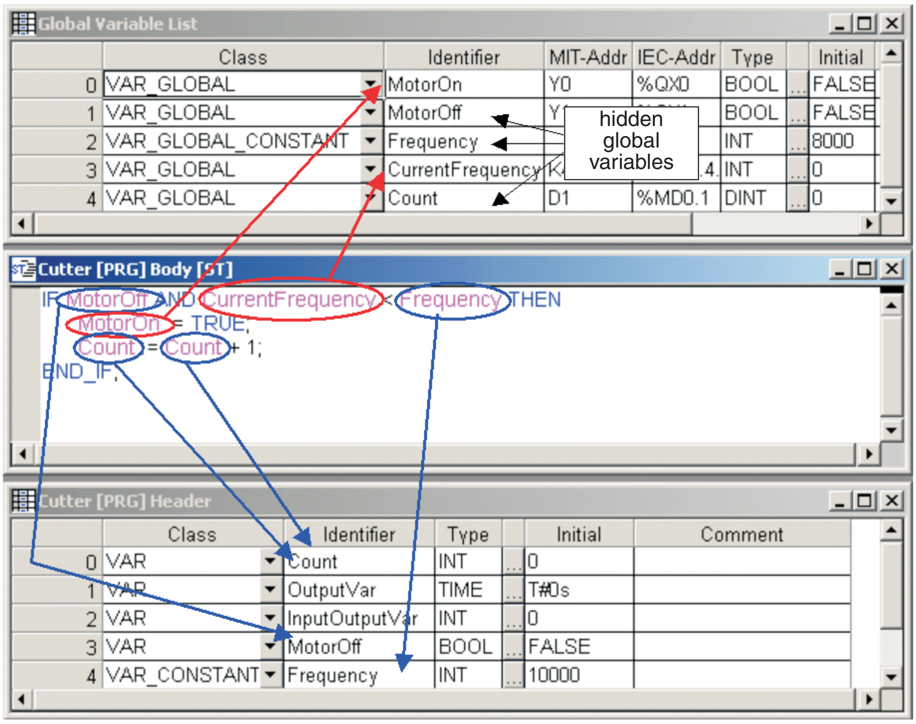
Every global variable is now accessible from any POU body, action or transition, without declaring it as an external variable in the POU scope. Global variables will be usable the same way as any local variable.

When there is a local variable declared in a POU with the same name as a global variable, the global variable is hidden, i.e. it is not possible to read or write the value of that global variable – always the local variable is referenced instead.

In the example below there are both global and local variables used in a POU body.

Example ▽

There are three variables in the POU’s header whose names are identical with global variable names: *Count*, *MotorOff* and *Frequency*. These local variables hide the corresponding global variables, so when e.g. *Count* is used in the POU’s body, it means the local variable *Count*, not the global variable *Count*.



	Class	Identifier	MIT-Addr	IEC-Addr	Type		Initial
0	VAR_GLOBAL	MotorOn	Y0	%QX0	BOOL	...	FALSE
1	VAR_GLOBAL	MotorOff	Y		BOOL	...	FALSE
2	VAR_GLOBAL_CONSTANT	Frequency			INT	...	8000
3	VAR_GLOBAL	CurrentFrequency	K	.4	INT	...	0
4	VAR_GLOBAL	Count	D1	%MD0.1	DINT	...	0

```
IF MotorOff AND CurrentFrequency < Frequency THEN
    MotorOn := TRUE;
    Count := Count + 1;
END_IF;
```

	Class	Identifier	Type		Initial	Comment
0	VAR	Count	INT	...	0	
1	VAR	OutputVar	TIME	...	T#0s	
2	VAR	InputOutputVar	INT	...	0	
3	VAR	MotorOff	BOOL	...	FALSE	
4	VAR_CONSTANT	Frequency	INT	...	10000	

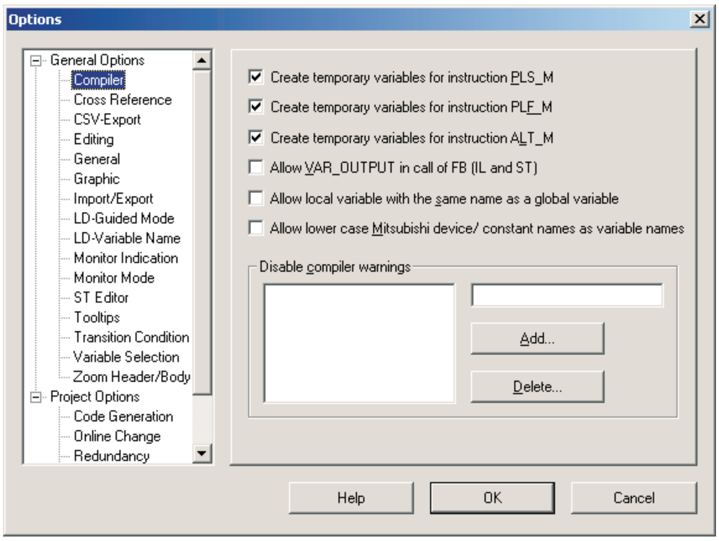
Fig. 10-1: *Global and local variables with the same name*

△

10.1.2 Compiler Options

Two new options are introduced on the **Compiler** page of the **Options**. One is to allow you to choose if global variable hiding should be considered as a programming error or not, the other one is to allow you to choose if using lower case device/constant names used as variable names should be considered as a programming error or not.

Please refer to section 2.7.1 for additional information.



☑ Create temporary variables for instruction PLS_M
☑ Create temporary variables for instruction PLF_M
☑ Create temporary variables for instruction ALT_M
□ Allow VAR_OUTPUT in call of FB (IL and ST)
□ Allow local variable with the same name as a global variable
□ Allow lower case Mitsubishi device/ constant names as variable names

Disable compiler warnings

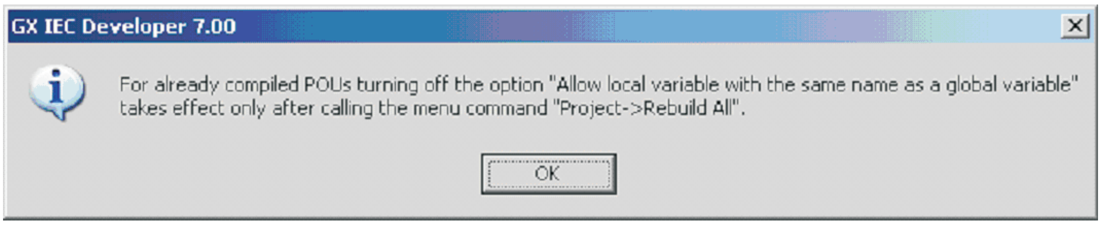
Fig. 10-2:
Options - Compiler

Hiding global variables

By default, if a local variable of a POU has the same name as a global variable, an error will be shown when the POU is checked. If you turn this option on, hiding of global variables will be allowed and the POU will be compiled.

POUs that have been already compiled without error and have not been changed since the compilation are not checked or built again when the corresponding menu command **Object** → **Check** or **Project** → **Build** is called.
Consequently, for already compiled POUs changing the option “Allow local variable with the same name as a global variable” from checked to unchecked has no effect, unless the whole project is recompiled by calling the menu item **Project** → **Rebuild All**.

You will be notified about this issue when you change the option from checked to unchecked and presses OK in the **Options** dialogue.



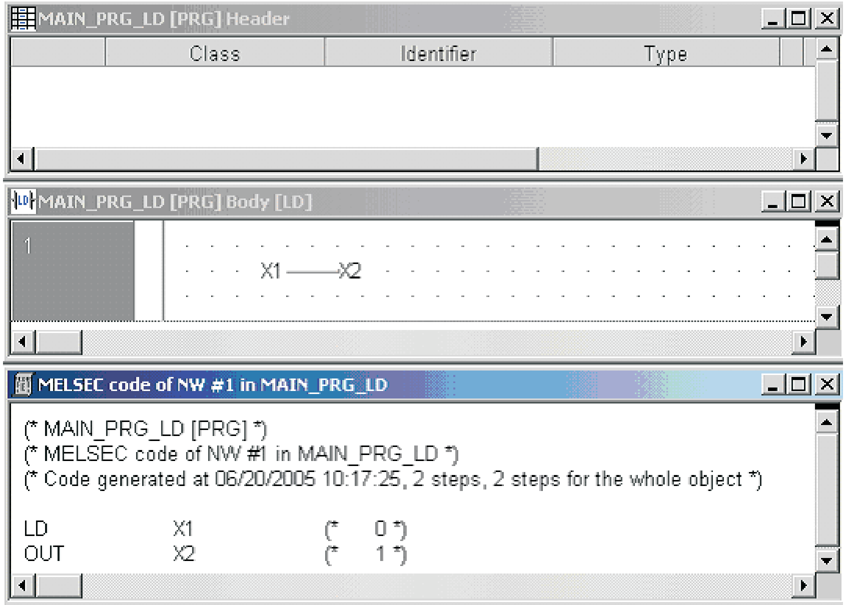
For already compiled POUs turning off the option "Allow local variable with the same name as a global variable" takes effect only after calling the menu command "Project->Rebuild All".

Fig. 10-3: *Warning message*

Using lower case device/constant names

The following programming is possible in GX IEC Developer:

- Using X1 and X2 variables in a body (either typed with caps, or automatically converted by the editor) with no declarations in the header or GVL.
  Result: Devices are used in the generated code.

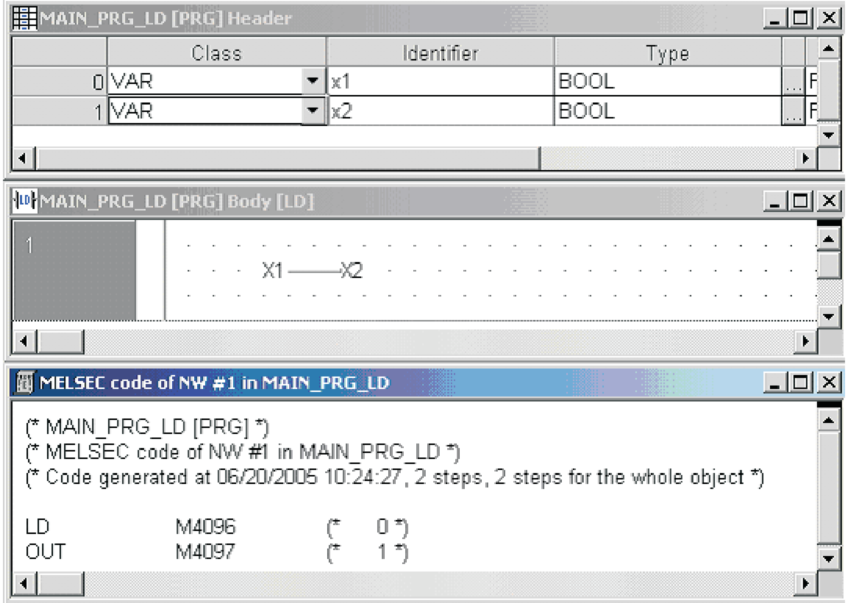

MAIN_PRG_LD [PRG] Header
Class
Identifier
Type
MAIN_PRG_LD [PRG] Body [LD]
1
X1
X2
MELSEC code of NW #1 in MAIN_PRG_LD

```
(* MAIN_PRG_LD [PRG] *)
(* MELSEC code of NW #1 in MAIN_PRG_LD *)
(* Code generated at 06/20/2005 10:17:25, 2 steps, 2 steps for the whole object *)

LD           X1           (*    0 *)
OUT          X2           (*    1 *)
```

Fig. 10-4: *Using X1 and X2 variables in a body with no declarations*

- Using X1 and X2 variables in a body (either typed with caps, or automatically converted by the editor) with x1 and x2 declarations in the header or GVL.
  Result: Variables are used in the generated code.


MAIN_PRG_LD [PRG] Header
MAIN_PRG_LD [PRG] Body [LD]
1
X1
X2
MELSEC code of NW #1 in MAIN_PRG_LD

	Class	Identifier	Type	
0	VAR	x1	BOOL	F
1	VAR	x2	BOOL	F

```
(* MAIN_PRG_LD [PRG] *)
(* MELSEC code of NW #1 in MAIN_PRG_LD *)
(* Code generated at 06/20/2005 10:24:27, 2 steps, 2 steps for the whole object *)

LD           M4096        (*    0 *)
OUT          M4097        (*    1 *)
```

Fig. 10-5: *Using X1 and X2 variables in a body with declarations*

To give the user a hint, that a Mitsubishi device or constant could be hidden by a variable declaration with lower case letters the new option is added to the **General Options\Compiler** option.

If the option is checked, the following warning is shown when a Mitsubishi device/ constant name is used as variable name in a lower case format:

> *Warning: C9173 The variable '%s' could hide a Mitsubishi device/ constant. Make sure you want to use the variable instead.*

If the option is unchecked, the following error is shown when a Mitsubishi device/ constant name is used in a lower case format as variable name:

> *Error: C9172 The variable '%s' has the same name in lower case format as a Mitsubishi device/ constant. This error can be converted to a warning in General Options\ Compiler.*

In the LD/FBD and ST editor Mitsubishi devices are already converted to upper case when they are entered in lower case format. In GX IEC Developer 7.01 this is extended to convert Mitsubishi constants, too.

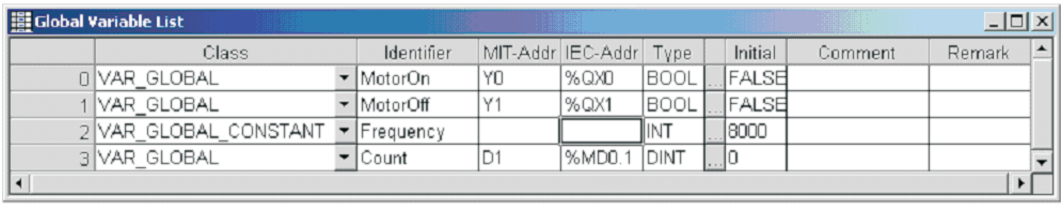
10.1.3 Variable Declaration

Variable Declaration in the Declaration Editors

The global variable list and POU header editors were modified to comply with the new convention. In the GVL editor the **Autoextern** checkbox column was removed, because every global variable now is accessible from any POU.

In the POU header editor VAR_EXTERNAL and VAR_EXTERNAL_CONSTANT were removed from the list of available classes. Moreover, the button column on the right of the Identifier column also is removed, because it was used to access the dialogue to select the name of an external variable from the list of global variable names.

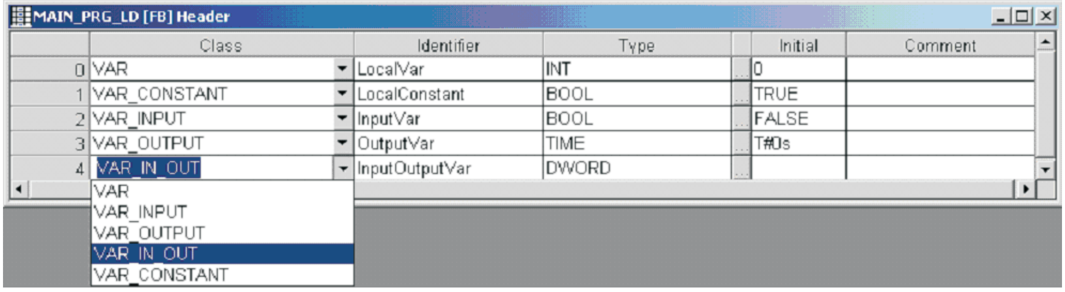
Example▽ Example of the changed Global Variable List editor:



	Class	Identifier	MIT-Addr	IEC-Addr	Type		Initial	Comment	Remark
0	VAR_GLOBAL	MotorOn	Y0	%QX0	BOOL	...	FALSE		
1	VAR_GLOBAL	MotorOff	Y1	%QX1	BOOL	...	FALSE		
2	VAR_GLOBAL_CONSTANT	Frequency			INT	...	8000		
3	VAR_GLOBAL	Count	D1	%MD0.1	DINT	...	0		

Fig. 10-6: *Sample of GVL*

Example to the changed POU header editor:

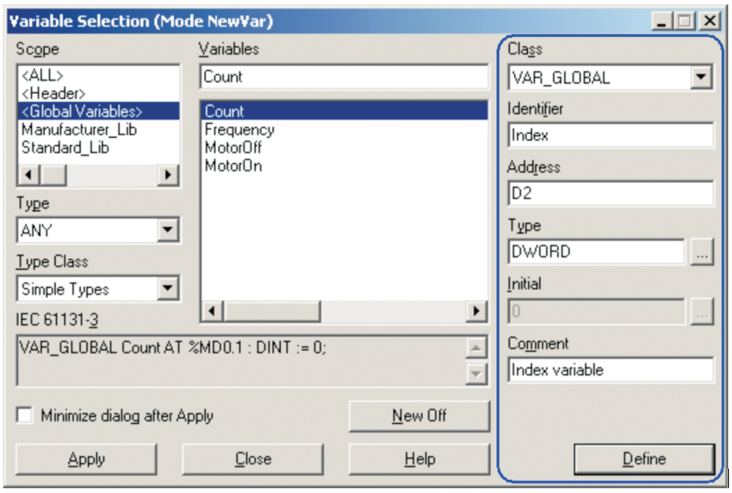


	Class	Identifier	Type		Initial	Comment
0	VAR	LocalVar	INT	...	0	
1	VAR_CONSTANT	LocalConstant	BOOL	...	TRUE	
2	VAR_INPUT	InputVar	BOOL	...	FALSE	
3	VAR_OUTPUT	OutputVar	TIME	...	T#0s	
4	VAR_IN_OUT	InputOutputVar	DWORD	...		

Fig. 10-7: *Sample of POU Header*

Variable Declaration in the Variable Selection Dialogue

The variable declaration part of the **Variable Selection** dialogue has also been modified to comply with the changed variable scope convention.
The **Autoextern** checkbox was removed from the dialogue, and the external variable classes (VAR_EXTERNAL and VAR_EXTERNAL_CONSTANT) are no longer available in the Class list when the scope of the **Variable Selection** dialogue is set to **<Header>**.


Variable Selection (Mode NewVar)

Scope: <ALL>, <Header>, <Global Variables>, Manufacturer_Lib, Standard_Lib

Type: ANY

Type Class: Simple Types

Variables: Count

Count, Frequency, MotorOff, MotorOn

Class: VAR_GLOBAL

Identifier: Index

Address: D2

Type: DWORD

Initial: 0

Comment: Index variable

IEC 61131-3: VAR_GLOBAL Count AT %MD0.1 : DINT := 0;

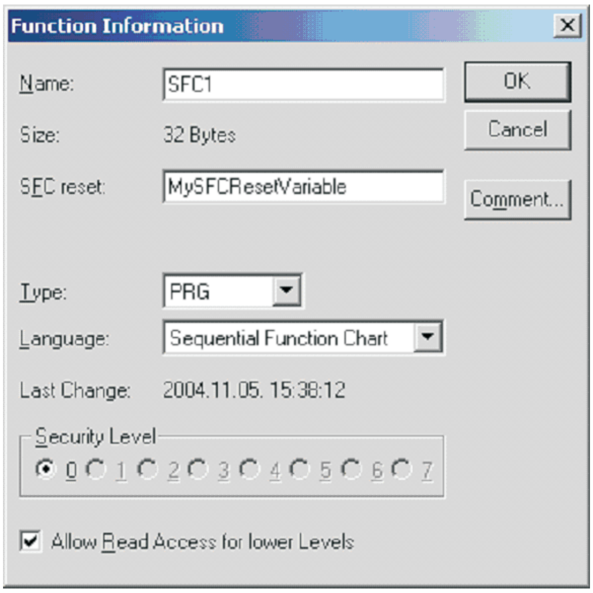
□ Minimize dialog after Apply

Fig. 10-8:
Variable Selection Dialogue

Declaration of a SFC Reset Variable

In each SFC POU of an FX project a Boolean variable can be specified to be used to trigger the reset of all steps and the set of the initial step. This variable can be either local or global (external in previous versions).
Formerly a variable was specified to be such a SFC reset variable by setting its comment field in the POU header to “!RESET!” (the first variable in the POU header with the comment “!RESET!” was used as SFC reset variable).
With the removal of external variables, this is not possible anymore, because a global variable can not be specified to be the reset variable for a POU, as it is not in the header of the POU.

The assignment of a SFC reset variable to a SFC POU of an FX project is introduced in the POU information dialogue. When calling **Object** → **Information** for a SFC POU in an FX project, the dialogue will provide the setting for the SFC reset variable.


Function Information

Name: SFC1

Size: 32 Bytes

SFC reset: MySFCResetVariable

Type: PRG

Language: Sequential Function Chart

Last Change: 2004.11.05. 15:38:12

Security Level: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

☑ Allow Read Access for lower Levels

Fig. 10-9:
Function Information Dialogue

NOTES

The SFC reset variable can be a local or global variable with the type BOOL or ARRAY OF BOOL. When an ARRAY OF BOOL variable is given, its first element will be used as the reset variable.

By pressing the function key F2 the dialogue **Variable Selection** can be called to select and apply a variable for SFC reset.

When pressing OK in the dialogue, the syntax of the entered name will be checked, but it will not be verified whether the entered name specifies an existing variable. When the POU or the project is checked (or built), and the SFC reset setting does not specify a valid variable, an error will be added to the list of errors and warnings in the **Compile/Check Messages** dialogue. See section 8.1 for details.

10.1.4 Update Variables

The function of updating variables is enabled when at least one declaration is selected in an active GVL or POU header editor; is disabled in Monitoring mode. It is disabled in POU header editors if the POU does not have a write-accessible body and in GVL editors if the GVL is in the Standard or the Manufacturer library, or if it is in a read-only user library.

Formerly when the functionality **Extras** → **Update Variables** was called in a GVL, every reference in the project to the selected variable(s) – including all user libraries – was updated. Variable update meant that in all POU headers the external variable(s) referring the selected global variable(s) were synchronized with the global variables, so that their type, address and other properties matched.
Furthermore, if the name of a global variable was changed, not only the corresponding external variables were updated in the POU headers, but also all occurrences of those external variables in the POU bodies.

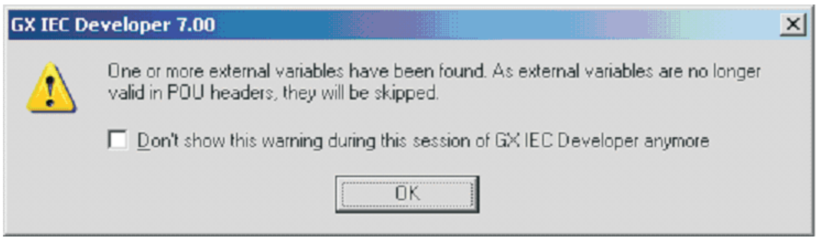
With the removal of external variables, POU headers do not have to be updated when a global variable is updated. If the name of a global variable is changed, **Update Variables** will change all occurrences of that global variable in each POU.

When the **Auto Update** feature is switched on in the Editing page of the Options (→ 2.7.4), the update of changed variables will be performed automatically each time the GVL is saved.

The behaviour of **Extras** → **Update Variables** remains the same as in former versions when called in a POU header: all occurrences of selected local variables whose name has been changed will be updated in the POU.

10.1.5 Import from Excel

When importing a Microsoft Excel file into a POU header, the file is checked for external variables in it.
If one or more variables with the class VAR_EXTERNAL or VAR_EXTERNAL_CONSTANT are found, you will be warned that these variables will be skipped.



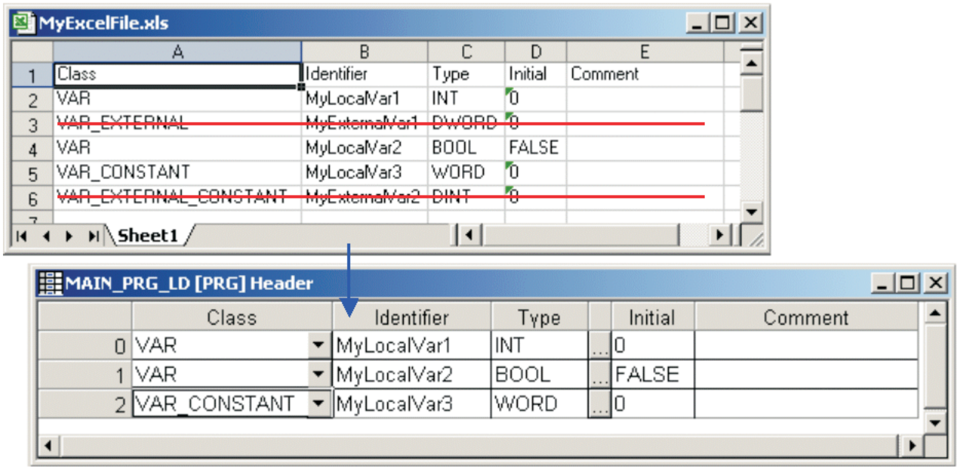
GX IEC Developer 7.00

One or more external variables have been found. As external variables are no longer valid in POU headers, they will be skipped.

□ Don't show this warning during this session of GX IEC Developer anymore

Fig. 10-10: *Warning message*

Example ▽



MyExcelFile.xls

	A	B	C	D	E
1	Class	Identifier	Type	Initial	Comment
2	VAR	MyLocalVar1	INT	0	
3	~~VAR_EXTERNAL~~	~~MyExternalVar1~~	~~DWORD~~	~~0~~	
4	VAR	MyLocalVar2	BOOL	FALSE	
5	VAR_CONSTANT	MyLocalVar3	WORD	0	
6	~~VAR_EXTERNAL_CONSTANT~~	~~MyExternalVar2~~	~~DINT~~	~~0~~	

MAIN_PRG_LD [PRG] Header

	Class	Identifier	Type	Initial	Comment
0	VAR	MyLocalVar1	INT	0	
1	VAR	MyLocalVar2	BOOL	FALSE	
2	VAR_CONSTANT	MyLocalVar3	WORD	0	

Fig. 10-11: *External variables are skipped*

△

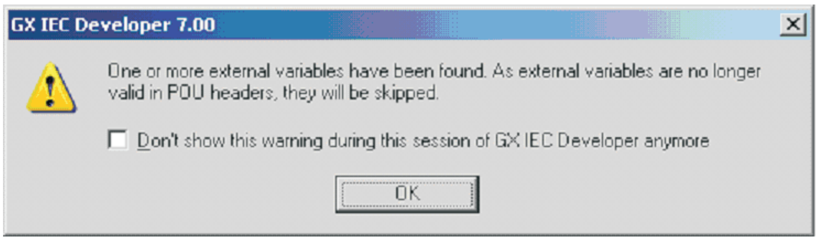
10.1.6 ASCII Import and Export

ASCII Import

When you import data from an ASCII export file that includes the **Global Variable List** (e.g. a complete project exported), the external variables will be automatically removed from the POU headers (including the POUs in user libraries).

When you import data from an ASCII export file that contains no **Global Variable List**, each imported POU header will be checked for external variables. If one or more variables with the class VAR_EXTERNAL or VAR_EXTERNAL_CONSTANT are found, you will be warned that the variable(s) will be skipped.

This message will not be displayed when you are importing an ASCII export file that was created by GX IEC Developer version 7.01 or newer.


GX IEC Developer 7.00

One or more external variables have been found. As external variables are no longer valid in POU headers, they will be skipped.

□ Don't show this warning during this session of GX IEC Developer anymore

Fig. 10-12: *Warning message*

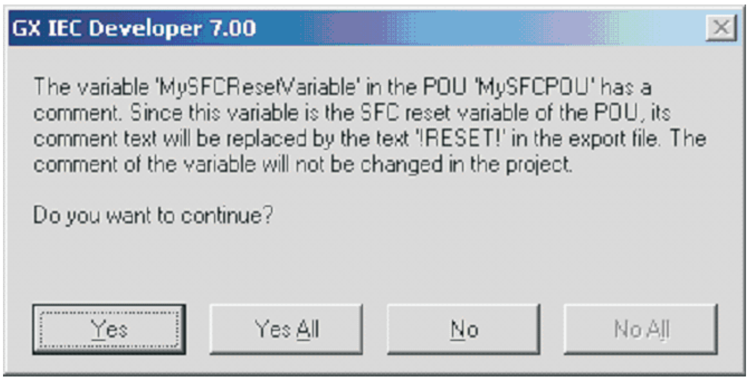
In FX projects before version 7.00, SFC reset variables were assigned to the POUs based on declaration comments, i.e. in each POU the first variable having the comment “!RESET!” was set to be the SFC reset variable of the POU. During ASCII import the SFC reset variable will be assigned to the POU information. When you call **Object** → **Information** for the SFC POU of the FX project, you can check the setting of the SFC reset variable.

ASCII Export

To keep compliance with the IEC standard, when exporting the project or any part of the project that includes at least one POU, the global variables used in each POU will be automatically declared as external variables of those POUs in the exported ASCII file. If such an exported file is imported in a new GID version working without external variables, the external variables will be removed from the headers. If it is imported in a previous version of GID, external variables will be imported as usual.

For FX projects, to keep compliance with the IEC standard, the SFC reset variable of a POU will be exported by setting the variable declaration’s comment field to “!RESET!”. This is because the SFC reset variable is a special GX IEC Developer feature, for which there is no standard IEC definition. When a global variable is used as a reset variable for a POU, the automatically declared external variable (please see "Declaration of a SFC Reset Variable" in section 10.1.3) will get the “!RESET!” comment.

When the comment field of a variable that is used as the SFC reset variable is not empty or is not equal to “!RESET!”, you will be warned that the comment field of that variable will be overwritten in the export file.



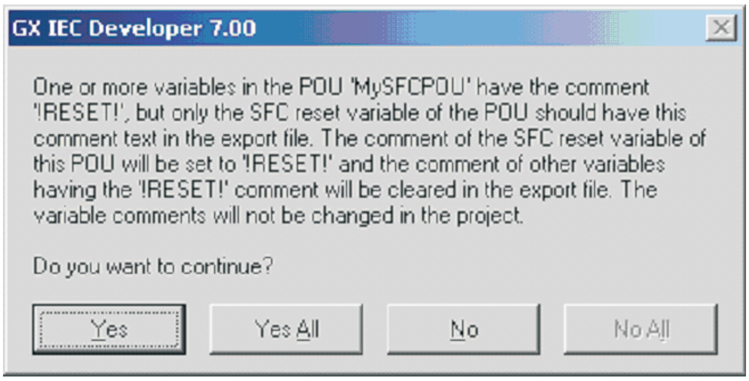
GX IEC Developer 7.00

The variable 'MySFCResetVariable' in the POU 'MySFCPOU' has a comment. Since this variable is the SFC reset variable of the POU, its comment text will be replaced by the text '!RESET!' in the export file. The comment of the variable will not be changed in the project.

Do you want to continue?

Fig. 10-13:
Warning message

If there is already another variable in the POU header with the comment “!RESET!”, you will be offered to set the comment of the actual SFC reset variable to “!RESET!” and clear the comment field of other variables having the comment text “!RESET!”.
This change affects only the ASCII export file, not the project.



GX IEC Developer 7.00

One or more variables in the POU 'MySFCPOU' have the comment '!RESET!', but only the SFC reset variable of the POU should have this comment text in the export file. The comment of the SFC reset variable of this POU will be set to '!RESET!' and the comment of other variables having the '!RESET!' comment will be cleared in the export file. The variable comments will not be changed in the project.

Do you want to continue?

Fig. 10-14:
Warning message

10.1.7 User Libraries

Installing Libraries

When installing user libraries, external variables will be removed from the header of each POU installed into the project from the user library. The removal affects only the POUs installed into the project, the user library file will be kept unchanged.

In FX projects, SFC reset variables will be assigned to the POUs based on declaration comments, i.e. in each POU the first variable having the comment “!RESET!” will be the SFC reset variable of the POU. Since version 7.00 the SFC reset variable is assigned to the POU information.

Saving Libraries

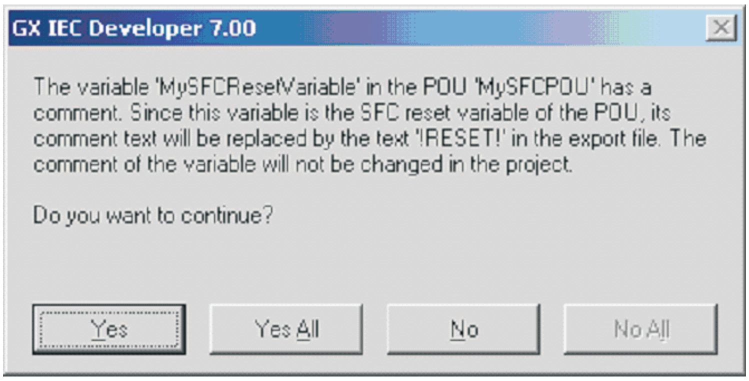
To keep compliance with the IEC standard, user libraries will be saved to the corresponding user library file, so that for each POU the global variables used in the POU will be automatically declared as external variables in the header of that POU.

This ensures that user libraries saved in GID 7.01 are compatible with former GID versions, starting from version 4.00.

If the user library is installed in a new GID version working without external variables, the external variables will be removed from the headers.
If a user library saved in the new GID version is installed in a former GID version, it will be able to use it without any problems because the necessary external variables will be declared.

For FX projects, to keep compliance with the IEC standard, the SFC reset variable of a POU will be saved so that it will work also when the user library is installed in a former GX IEC Developer version. Because of this the variable's comment will be set to "!RESET!".

When the comment field of a variable that is used as the SFC reset variable is not empty or is not equal to "!RESET!", you will be warned that the comment field of that variable will be overwritten in the user library file.



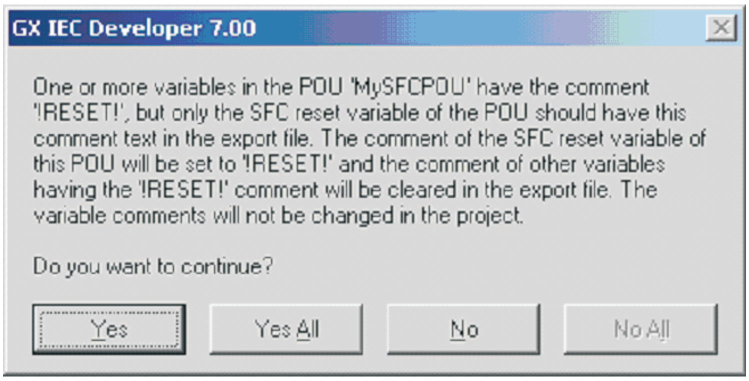
GX IEC Developer 7.00

The variable 'MySFCResetVariable' in the POU 'MySFCPOU' has a comment. Since this variable is the SFC reset variable of the POU, its comment text will be replaced by the text '!RESET!' in the export file. The comment of the variable will not be changed in the project.

Do you want to continue?

Fig. 10-15:
Warning message

If there is already another variable in the POU header with the comment "!RESET!", you will be offered to set the comment of the actual SFC reset variable to "!RESET!" and clear the comment field of other variables having the comment text "!RESET!".
This change affects only the user library file, not the project.



GX IEC Developer 7.00

One or more variables in the POU 'MySFCPOU' have the comment '!RESET!', but only the SFC reset variable of the POU should have this comment text in the export file. The comment of the SFC reset variable of this POU will be set to '!RESET!' and the comment of other variables having the '!RESET!' comment will be cleared in the export file. The variable comments will not be changed in the project.

Do you want to continue?

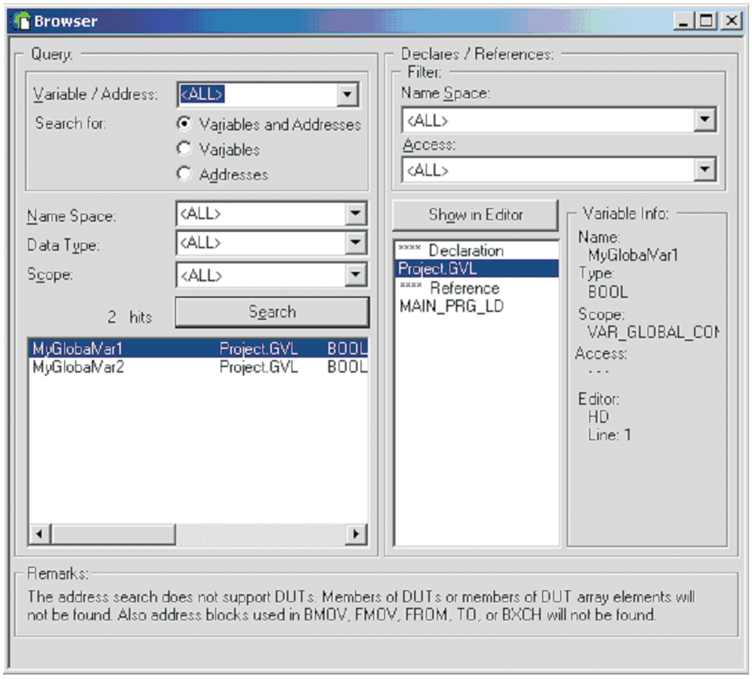
Fig. 10-16:
Warning message

10.1.8 Browser

In the Browser window the VAR_EXTERNAL and VAR_EXTERNAL_CONSTANT classes are no longer available in the **Scope** list.

When a global variable is selected in the left hand side list, the location of its declaration (the GVL of the project or a user library) will be shown on the right hand side list, together with the references to it.
Formerly there could be several items in the Declaration part of the right hand side list for a global variable, because external variable declarations were listed as well.

With the removal of external variables only the GVL declaration is listed for a global variable.



Variable / Address: <ALL>
Search for: ● Variables and Addresses ○ Variables ○ Addresses
Name Space: <ALL>
Data Type: <ALL>
Scope: <ALL>
2 hits

MyGlobalVar1	Project.GVL	BOOL
MyGlobalVar2	Project.GVL	BOOL

Name Space: <ALL>
Access: <ALL>

**** Declaration
Project.GVL
**** Reference
MAIN_PRG_LD

Name:
MyGlobalVar1
Type:
BOOL
Scope:
VAR_GLOBAL_CON
Access:
. . .
Editor:
HD
Line: 1

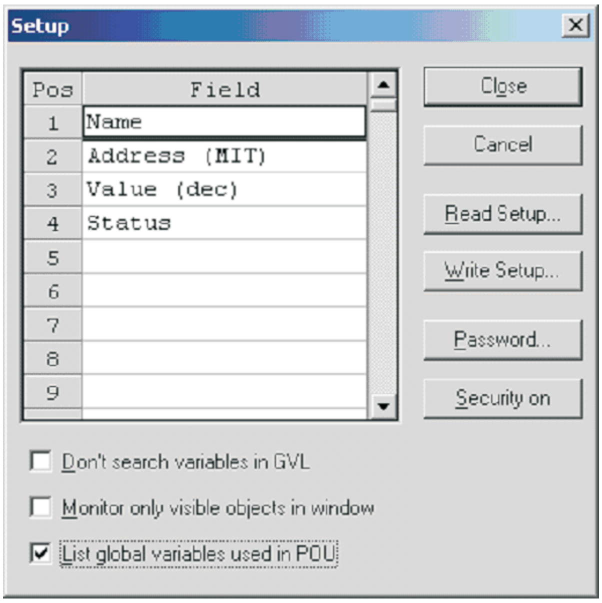
The address search does not support DUTs. Members of DUTs or members of DUT array elements will not be found. Also address blocks used in BMOV, FMOV, FROM, TO, or BXCH will not be found.

Fig. 10-17:
Browser dialogue

10.1.9 Entry Data Monitor

Formerly when a POU entry has been inserted into the Entry Data Monitor and expanded, all local and external variables were listed under the POU name.

In the new version of GID a new setting is introduced in the **EDM Setup** to switch the listing of global variables used in a POU on and off.
When the setting is off, only local variables (those declared in the POU header) will be listed under the POU.
When the setting is on, the list will also contain all global variables used in the POU's body, actions and transitions.



Setup

Pos	Field
1	Name
2	Address (MIT)
3	Value (dec)
4	Status
5	
6	
7	
8	
9	

☐ Don't search variables in GVL

☐ Monitor only visible objects in window

☑ List global variables used in POU

Fig. 10-18:
Setup dialogue

10.2 Opening Projects saved by former Versions

10.2.1 Opening older projects

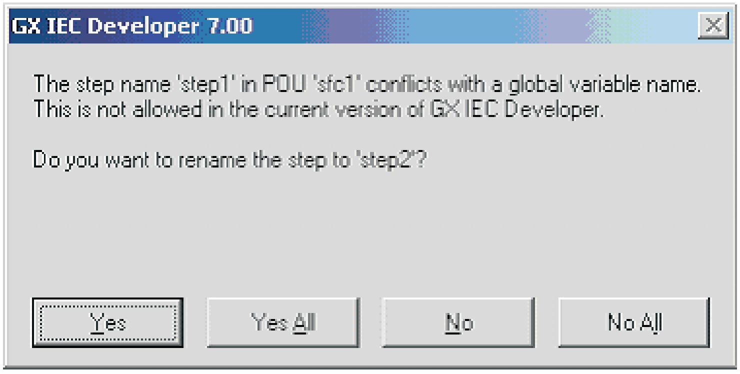
When opening a projects saved by a former version of GX IEC Developer, the project will be converted according to the new variable scope convention:

- External variables will be removed from each POU's header (including POUs in user libraries) (also see section 10.2.2).
- SFC reset variables in FX projects will be assigned to the POUs based on declaration comments, i.e. the first variable having the comment "!RESET!" will be the SFC reset variable of the POU (also see section 10.2.3).
- The objects of an SFC POU (steps, actions and transitions) conflicting with global variable names will be offered to be automatically renamed.

The objects of an SFC POU (steps, actions and transitions) are not allowed to be named the same as any local or global variable accessible from the POU. Before GX IEC Developer Version 7.01 a global variable could be used in a body only if it had been declared in the POU as an external variable, therefore an SFC object could have the same name as a global but non-external variable.

In GID 7.01 each global variable is accessible in every POU, therefore the name of an SFC object may be the same as the name of any global variable. This leads to the problem that a project that could be compiled in a former version could cause an error in GID 7.01 when building the project, if it had one or more SFC objects sharing names with global but non-external variables.

If a project last saved with a GX IEC Developer version preceding 7.01 is opened, or a project is imported from an .ASC file using File → Other → Import, SFC objects conflicting with global variable names will be offered to be automatically renamed.



The step name 'step1' in POU 'sfc1' conflicts with a global variable name.
This is not allowed in the current version of GX IEC Developer.

Do you want to rename the step to 'step2'?

Fig. 10-19:
Auto-renaming SFC POU objects

Yes: This renames the mentioned SFC object.

Yes All: Renames the mentioned and following problematic SFC objects without showing this dialogue again.

No: Does not rename the mentioned SFC object.

No All: Does not rename the mentioned nor the following SFC objects.

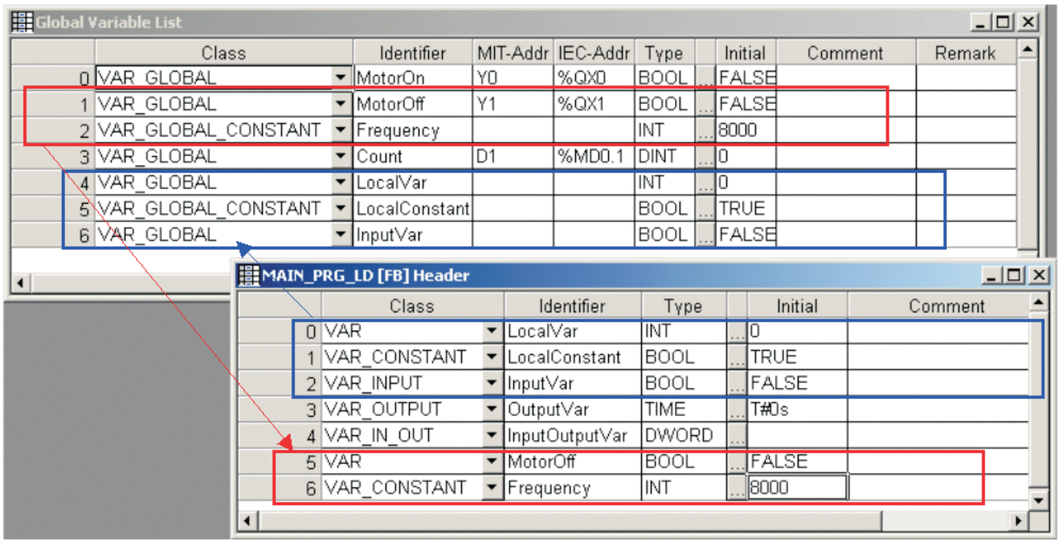
Automatic renaming sets an index number at the end of the name, or increases it if there is one. If the name is too long, the last character will be cut before numbering. If the new name would conflict with a device name, the index number will be separated with an underscore character (e.g. X_1 instead of X1).

If the conflicting SFC object is not write-accessible, no renaming will be offered. Any unresolved conflict will cause an error during build, and can then be fixed manually by the user.

10.2.2 Copying and Moving Variables

When inserting variables into the POU header of GX IEC Developer 7.01 from a POU header editor of a former GX IEC Developer version via clipboard operations (Copy/Cut/Paste) or Drag & Drop, variables with the class VAR_EXTERNAL or VAR_EXTERNAL_CONSTANT will be skipped. This applies also to the case when variables are dropped from a declaration list of a former GX IEC Developer version onto a POU header in the project navigator window.

To easily create a local variable with the same properties as a global one, when a variable with the class VAR_GLOBAL or VAR_GLOBAL_CONSTANT is copied to a POU header, its class will be changed to VAR or VAR_CONSTANT, respectively. This is also possible in the opposite direction: VAR_CONSTANT converts to VAR_GLOBAL_CONSTANT, while all other classes convert to VAR_GLOBAL.



	Class	Identifier	MIT-Addr	IEC-Addr	Type	Initial	Comment	Remark
0	VAR_GLOBAL	MotorOn	Y0	%QX0	BOOL	FALSE		
1	VAR_GLOBAL	MotorOff	Y1	%QX1	BOOL	FALSE		
2	VAR_GLOBAL_CONSTANT	Frequency			INT	8000		
3	VAR_GLOBAL	Count	D1	%MD0.1	DINT	0		
4	VAR_GLOBAL	LocalVar			INT	0		
5	VAR_GLOBAL_CONSTANT	LocalConstant			BOOL	TRUE		
6	VAR_GLOBAL	InputVar			BOOL	FALSE		

	Class	Identifier	Type	Initial	Comment
0	VAR	LocalVar	INT	0	
1	VAR_CONSTANT	LocalConstant	BOOL	TRUE	
2	VAR_INPUT	InputVar	BOOL	FALSE	
3	VAR_OUTPUT	OutputVar	TIME	T#0s	
4	VAR_IN_OUT	InputOutputVar	DWORD		
5	VAR	MotorOff	BOOL	FALSE	
6	VAR_CONSTANT	Frequency	INT	8000	

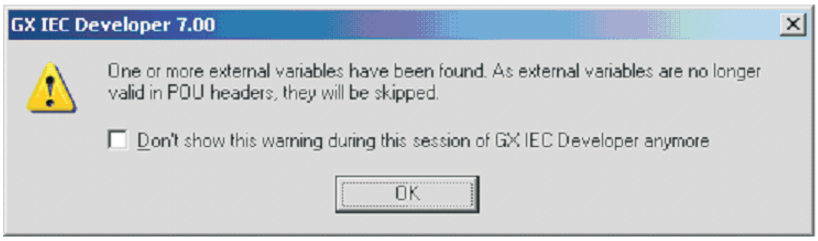
Fig. 10-20: *Sample for copying and moving variables*

10.2.3 Clipboard Operations and Drag & Drop in the Project Navigator

When a selection including one or more POUs and the Global Variable List is copied or moved from a project opened in a former GID version to a project opened in the new GID version by clipboard operations (Copy/Cut/Paste) or Drag & Drop, external variables will be removed from POU headers.

If the copied or moved selection includes one or more POUs but no Global Variable List, the header of each selected POU will be checked for external variables. If one or more variables with the class VAR_EXTERNAL or VAR_EXTERNAL_CONSTANT are found, you will be warned that the variables will be skipped.

This message will be displayed only when copying or moving data from a project opened in a GX IEC Developer version preceding version 7.01.



GX IEC Developer 7.00

One or more external variables have been found. As external variables are no longer valid in POU headers, they will be skipped.

□ Don't show this warning during this session of GX IEC Developer anymore

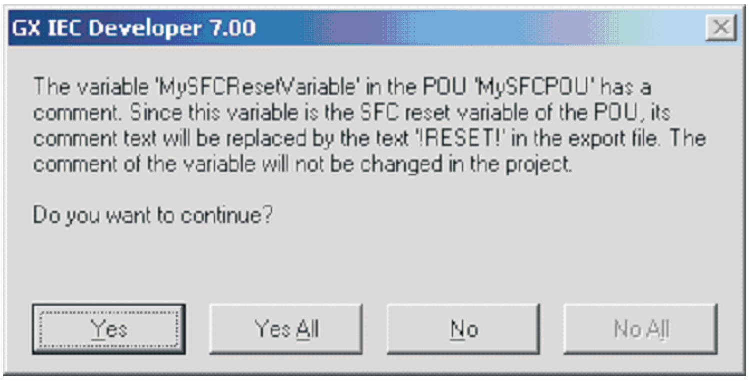
Fig. 10-21:
Warning message

In FX projects, SFC reset variables will be assigned to the POUs based on declaration comments, i.e. in each POU the first variable having the comment "!RESET!" will be the SFC reset variable of the POU.

To keep compliance with the IEC standard, when a selection including one or more POUs is copied or moved by clipboard operations (Copy/Cut/Paste) or Drag & Drop from a project opened in the new GID version to a project opened in a former GID version, the global variables used in each POU will be automatically declared as external variables in the POU's header in the destination project. As a result, former GID versions – working with external variables – will be able to compile a copied POU without any problems.

For FX projects, to keep compliance with the IEC standard, the SFC reset variable of a POU will be copied by setting the variable declaration's comment field to "!RESET!". When a global variable is used as the reset variable of a POU, the automatically declared external variable (see previous paragraph) will get the "!RESET!" comment.

When the comment field of a variable that is used as the SFC reset variable of the POU is not empty or is not equal to "!RESET!", you will be warned that the comment field of that variable will be overwritten when copied. This warning will be displayed only for clipboard operations, not for drag & drop. For drag & drop "Yes All" is assumed as the user's answer.



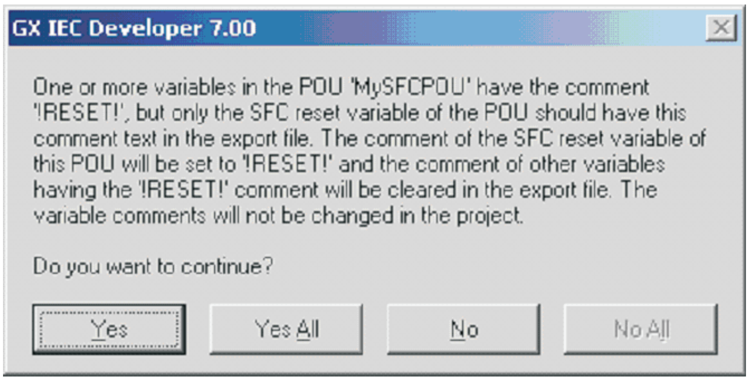
GX IEC Developer 7.00

The variable 'MySFCResetVariable' in the POU 'MySFCPOU' has a comment. Since this variable is the SFC reset variable of the POU, its comment text will be replaced by the text '!RESET!' in the export file. The comment of the variable will not be changed in the project.

Do you want to continue?

Fig. 10-22:
Warning message

If there is already another variable in the POU header with the comment "!RESET!", you will be offered to set the comment of the actual SFC reset variable to "!RESET!" and clear the comment field of other variables having the comment text "!RESET!".
This change affects only the declaration copied to the clipboard, not the declaration in the project. This warning will be displayed only for clipboard operations, not for drag & drop. For drag & drop "Yes All" is assumed as the user's answer.



GX IEC Developer 7.00

One or more variables in the POU 'MySFCPOU' have the comment '!RESET!', but only the SFC reset variable of the POU should have this comment text in the export file. The comment of the SFC reset variable of this POU will be set to '!RESET!' and the comment of other variables having the '!RESET!' comment will be cleared in the export file. The variable comments will not be changed in the project.

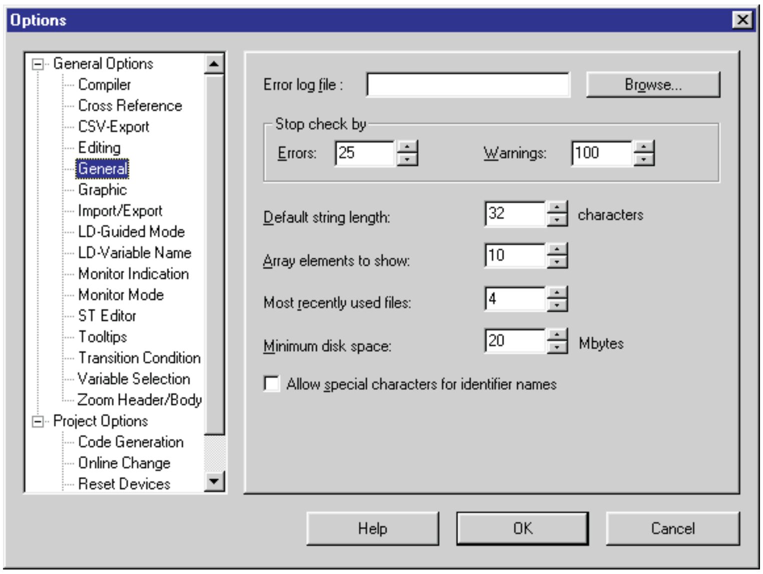
Do you want to continue?

Fig. 10-23:
Warning message

10.3 Special characters for identifier

In former versions of GX IEC Developer it was not possible to have identifier names which include special characters like e.g. ‘ä’, ‘ö’, ‘ü’. This was not possible due to the IEC naming convention.
GX IEC Developer does now support the usage of special characters.

When the option "Allow special characters for identifier names" in the **Options-General** dialogue is checked, you can use special characters for identifier names.



Error log file :

Stop check by
Errors: 25
Warnings: 100

Default string length: 32 characters
Array elements to show: 10
Most recently used files: 4
Minimum disk space: 20 Mbytes

☐ Allow special characters for identifier names

Fig. 10-24:
Options - General

The following list shows all newly supported valid characters:

Valid characters: ‘ä’, ‘Ä’, ‘ö’, ‘Ö’, ‘ü’, ‘Ü’, ‘ß’, ‘_’, ‘å’, ‘Å’

The usage of these characters is valid for the following identifier and all functionalities where these identifiers are used (e.g. Variable selection dialogue):

- Library names
- Task names
- DUT names and DUT component names
- Global and local variable names
- POU names (PRG, FUN, FB)
- Network label names
- SFC Transition names
- SFC Action names
- SFC Step names
- SFC Jump label names

The character ‘_’ (underscore) was already allowed in older versions of GX IEC Developer. Due to the restrictions by the IEC naming convention it was not allowed to use a double underscore character at the beginning of an identifier (e.g. “__TestVar”). Additionally there was a restriction for the usage of the underscore character as the last character of an identifier (e.g. “TestVar_”). These restrictions can now be disabled by the user.

These identifiers can be used in every program editor:

- LD
- FBD
- IL
- MELSEC-IL
- SFC
- ST

NOTES

The usage of numbers for the first character for identifier names is not possible.

Identifiers including special characters should only be used in the same way as they were listed in the local header or GVL (upper/lower case).
Otherwise the compiler will show an error message after compiling.

10.4 Compatibility

10.4.1 File based compatibility with former versions

- GX IEC Developer 7.01 can open project databases (*.pro file) of MM+ 2.1, 2.30a, 2.31, 2.31a, 2.32, 2.40, 2.40a, 2.50, 3.00, GX IEC Developer 4.00, 5.00, 5.01, 5.02, 6.00, 6.01, 6.10, 6.11 and 7.00, but these versions can not open GX IEC Developer 7.01 project databases.
  Exception: GX IEC Developer 7.01 can not open project databases (*.pro file) of MM+ 2.1, 2.30a, 2.31, 2.31a, 2.32, 2.40, 2.40a, 2.50 and 3.00 with Microsoft Windows® Me.
  Such projects can be opened and exported under other supported operating systems. Then the projects can be imported under Microsoft Windows® Me.
- GX IEC Developer 7.01 can not open project databases (*.pro file) of MM+ 2.41 and vice versa.
- GX IEC Developer 7.01 can import projects, which have been exported (*.asc file) by MM+ 2.1, 2.30a, 2.31, 2.31a, 2.32, 2.40, 2.40a, 2.41, 2.50, 3.00, GX IEC Developer 4.00, 5.00, 5.01, 5.02, 6.00, 6.01, 6.10, 6.11 and 7.00, but these versions cannot import *.asc exports of GX IEC Developer 7.01.
- GX IEC Developer 7.01 can open library files (*.lib file) of GX IEC Developer 4.00, 5.00, 6.00, 6.10, 6.11 and 7.00, but these versions cannot open GX IEC Developer 7.01 library files.

10.4.2 PLC project based compatibility with former versions

It is possible, that the generated code of GX IEC Developer 7.01 is different to the generated code of a former version of GX IEC Developer. This could result in the following facts:

- An online change can fail, if the project was not downloaded with GX IEC Developer 7.01.
- The functionality to verify the generated code with the program on the PLC can show differences.

The reason for the differences are internal changes in the code generator which result from the implementation of new features in GX IEC Developer 7.01.

NOTE

Please also see section 2.8.1, Code Generation.

10.4.3 Compatibility with other programming software for MELSEC PLCs:

- GX IEC Developer 7.01 can import MELSEC MEDOC/DOS print files (*.tmp) of the versions 1.64b/c, 2.20 and 2.30.
- GX IEC Developer 7.01 generally can not upload parameters and programs from Q and QnA/S CPUs, which are generated by GX Developer or GPPQ.
- GX IEC Developer 7.01 can upload only programs from Q and QnA/S CPUs which are generated according to the limitations of GX IEC Developer 5.00, 5.01, 5.02, 6.00, 6.01, 6.10, and 6.11 compared to the Q and QnA/S features (only one program file MAIN.QPG).
- GX IEC Developer 7.01 can upload programs from A CPUs which are generated with GX Developer or GPPA.
- GX IEC Developer 7.01 can not open project databases from GX Developer, GPPQ or GPPA.

There is currently no direct possibility to exchange projects between GX IEC Developer 7.01 and GX Developer, GPPQ or GPPA.

11 Support of other Applications

11.1 GX Configurator-DP Support

GX IEC Developer supports the configuration of Profibus Modules. A entry called **Module Configuration** is implemented in the Parameters section of the Navigator.

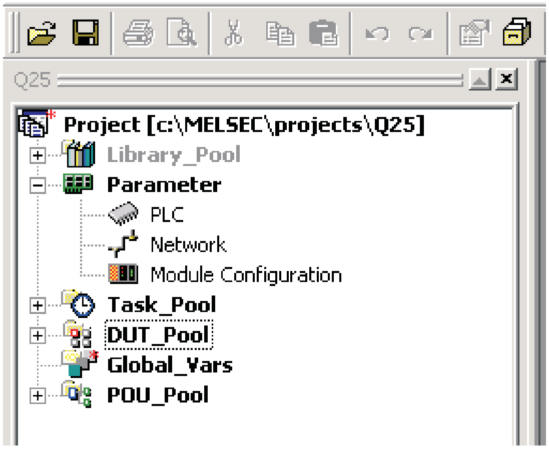
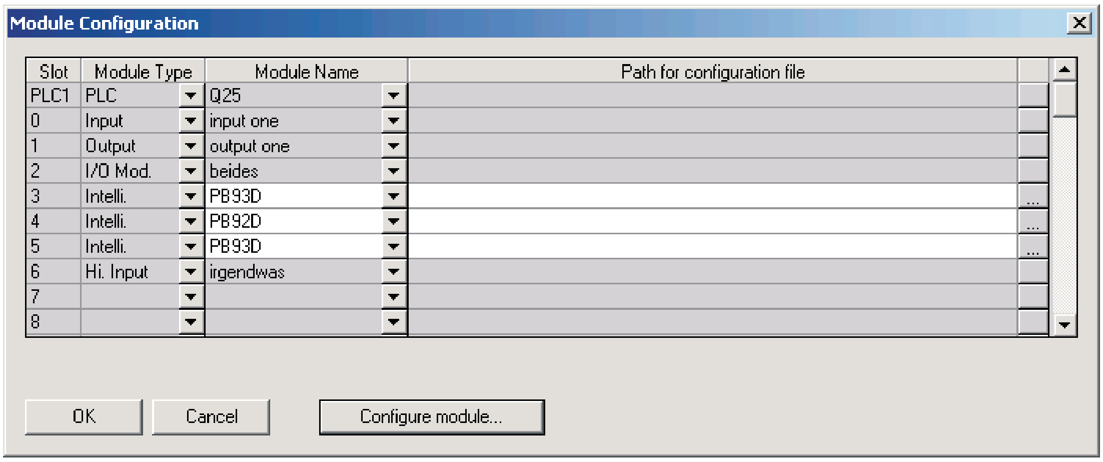


Fig. 11-1:
Project Navigator

The module configuration invokes a dialogue with a list of all configured modules, which are currently available. This information is initially copied from the I/O settings in the PLC parameter configuration. Furthermore it is possible to change the module name. In this case, the I/O settings in the PLC parameter are updated.

For all intelligent function modules a configuration editor can be called by **Configure module**.



Slot	Module Type	Module Name	Path for configuration file
PLC1	PLC	Q25	
0	Input	input one	
1	Output	output one	
2	I/O Mod.	beides	
3	Intelli.	PB93D	
4	Intelli.	PB92D	
5	Intelli.	PB93D	
6	Hi. Input	irgendwas	
7			
8			

Fig. 11-2: *Module Configuration*

Configure module: Opens the corresponding module configuration program (only for Profibus and intelligent function modules).

- Profibus modules

  Profibus modules are identified by the module name and the module type **intelli.**. Valid module names for Profibus modules are **PB92D** (for QJ71PB92D or A(1S)J71PB92D), **PB93D** (for QJ71PB93D) and **FX3U-64DP-M** (for FX3U-64DP-M). In case of Profibus Modules, each module can be assigned to a Profibus configuration by either selecting an existing configuration path or by creating a new configuration. By clicking on **Configure module**, the optional application GX Configurator-DP will be called for a more comfortable configuration of the selected Profibus module.

- Intelligent function modules:

  If the module type is **intelli.** and the module name does not match the Profibus description, **Configure module** calls the intelligent function utilities.

  For all other modules, the button **Configure module** is disabled.

NOTE

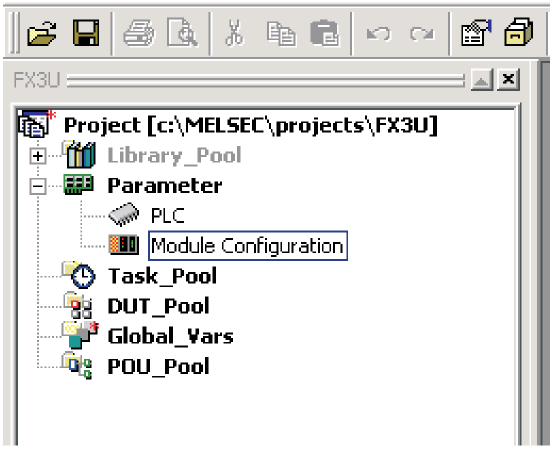
At the first time GX Configurator-DP is called from the dialogue, the user will be prompted to specify the location of the application.

11.2 FX Configurator-EN Support

With the introduction of the new Ethernet unit (FX3U-ENET), GX IEC Developer is now able to communicate with the FX3U via Ethernet. The parameters of the new Ethernet unit are not be part of the PLC parameter, they are stored in the Ethernet unit.

An external application, FX Configurator-EN was introduced to set the parameters and diagnose the FX3U-ENET module.

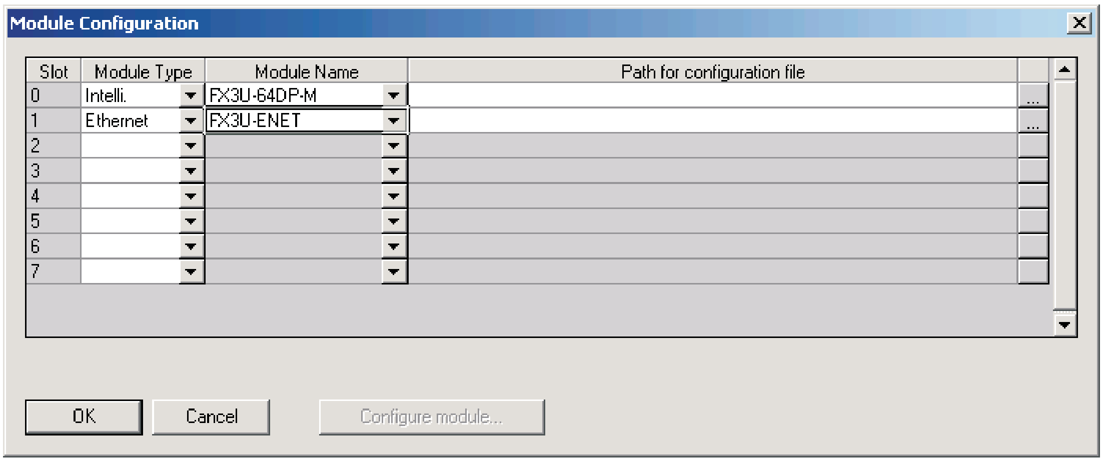
Open the **Module Configuration** implemented in the Parameters section of the Navigator.



Project [c:\MELSEC\projects\FX3U]
- Library_Pool
- Parameter
  - PLC
  - Module Configuration
- Task_Pool
- DUT_Pool
- Global_Vars
- POU_Pool

Fig. 11-3: *Project Navigator in case of a FX3U CPU*

For the FX3U eight module positions are available. For each position the type and name of the module can be selected. If the selected module in a row is either a Profibus module or a Ethernet module, for each of which a Configurator is available, the field to specify or browse for the path of the configuration file will be enabled.



Module Configuration

Slot	Module Type	Module Name	Path for configuration file
0	Intelli.	FX3U-64DP-M	
1	Ethernet	FX3U-ENET	
2			
3			
4			
5			
6			
7			

Fig. 11-4: *Module Configuration for a FX3U CPU*

The button **Configure module…** will be enabled, if ...

- FX3U-ENET is selected in the active row and FX Configurator-EN (optional) is installed. Clicking on **Configure module** will start the FX Configurator-EN.
- FX3U-64DP-M is selected in the active row and GX Configurator-DP (optional) is installed. Clicking on **Configure module** will start the GX Configurator-DP.
  At the first time GX Configurator-DP is called from the GX IEC Developer, the user will be prompted to specify the location of this application's executive file to be able to start the application.

A Glossary

Absolute address
A variable’s absolute address references a fixed memory location in the CPU or a physical input or output. This absolute address can be assigned using either IEC syntax (IEC address) or MITSUBISHI syntax (MITSUBISHI address). The user must only assign absolute addresses for the global variables. The program assigns suitable memory locations for the local variables automatically.

Absolute Label
Absolute labels are the addresses of specific pointers in the PLC program (e.g. P20). When PLC programs are uploaded to GX IEC Developer absolute labels are assigned as jump instruction destinations. Every program section identified by a pointer in the original PLC program is loaded in a separate MELSEC network. You can also use absolute labels (Syntax: P_20) when you assign your own names to networks.

Accumulator
The accumulator is a result handling facility used in IEC Instruction List. You may be familiar with it from other high-level programming languages. The result of every operation performed is written to the accumulator directly after execution. The accumulator thus always contains the result of the last instruction programmed. This means that you do not have to program any input conditions (execution conditions) for the operations, as execution is always dependent on the contents of the accumulator.

Action
An action is allocated to every step in an SFC sequence. These actions can be Boolean output variables or PLC programs.

Action Pool
When you write a POU in Sequential Function Chart language the action pool is displayed in the Project Navigator window as an entry in that POU. The Action Pool contains all the actions belonging to the program organisation unit.

Actual parameters
Functions and function blocks have two kinds of parameters: Formal parameters and actual parameters. The formal parameters are the variables used when you create a function or a function block. The actual parameters are the variables or values passed to the function or function block instance when they are used in another program organisation unit.

’Adapted’ MELSEC instructions
‘Adapted’ MELSEC instructions are MITSUBISHI ELECTRIC instructions with a special modified nomenclature (e.g. BMULTI_M, PLUS_M, FLOAT_MD) for use in the IEC editor. They always have the suffix _M for ‘MELSEC’ or _MD for ‘MELSEC Dedicated instruction’. All the ‘adapted’ MELSEC instructions are stored in the Manufacturer Library. These instructions can only be used in the IEC editor (→ Appendix E).

Array
Arrays are derived data types, consisting of fields of one kind of variable. You can define arrays with up to three dimensions.

AutoConnect mode
The Autoconnect mode is a special editing mode used in the graphical editors (→ Chapter 3).

Body
Program organisation units (POUs) always consist of a header and a body. The body contains the actual PLC program code.

Breakpoint
Breakpoints are a debugging aid. You can set a breakpoint at any point in a program organisation unit. The execution of the PLC program will then automatically stop at this point.

Class
The class is specified in the declaration table to define the way variables are to be used in the project. The following classes are supported: VAR, VAR_CONSTANT, VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR_GLOBAL and VAR_GLOBAL_CONSTANT.

Comment
You can enter and store explanatory comment texts for all objects used in GX IEC Developer. National characters can be used.

Compiling
When you compile a project GX IEC Developer translates the program into code that can be read and executed directly by the PLC.

Computer link module
One of the several ways of connecting the PLC to your personal computer is with a MITSUBISHI ELECTRIC computer link module. Computer link modules available from MITSUBISHI ELECTRIC include the following models: AJ71C21, AJ71UC24, AJ71QC24, A1SJ71QC24, QJ71C24.

Continue condition
→ ‘Transition condition’

Controller instruction
PLC programs written in Instruction List language consist of a series of controller instructions. Each controller instruction contains an operator, i.e. the programming instruction, and an operand, i.e. the variable.

CPU interface
You can connect the PLC to your personal computer via the CPU interface, which is located on the PLC unit.

Data type
The data type of a variable defines the number of bits it contains, how they are processed and the variable’s value range. The following data types are available: BOOL, INT, DINT, WORD, DWORD, TIME, REAL, ARRAY, STRING.

Data Unit Type
The Data Unit Type (DUT) is a structured data type. It contains a collection of variables.

Declaration table
Declaration tables are lists used for managing and declaring both global and local variables.

Device Editor
The device editor allows you to display and edit the contents of the PLC CPU’s bit and word devices (registers). The central element of the device editor is a table that is very similar to a Microsoft Excel worksheet, both in its appearance and in the way it is used.

Dialogue box
A dialogue box is a special kind of window used for handling dialogues with the user. They can contain a number of different elements, including fields for entering data, check boxes or radio buttons for selecting options and messages containing important warnings or information. Dialogue boxes are displayed at many different points in the program, for example when you select menu commands followed by three periods (e.g. Open...).

Download
Transfer process from GX IEC Developer to the PLC

DUT Pool
The DUT Pool is one of the standard entries in the Project Navigator window (DUT_Pool). It contains all the Data Unit Types used in the project.

Edit mode
When you are editing the body of a POU you can choose between Select mode and Edit mode. In Edit mode you can perform actions on selected program elements (e.g. move, overwrite, delete etc).

Editing area
The editing window for working on the body of a POU is divided into two main areas, the network bar and the editing area. The editing area is where you enter the component elements of your program.

Editor
The GX IEC Developer editors are program modules used for writing and editing programs in the supported programming languages. The editors are displayed as separate windows on the screen. GX IEC Developer includes text, graphical, IEC and MELSEC editors.

EN input
In addition to their input and output variables, some function blocks also have a Boolean input (EN = ENable) and a Boolean output (ENO = ENable Out). The status of the ENO output always corresponds to the current status of the EN input.

ENO output
→ ‘EN input’

Entry Data Monitor
GX IEC Developer supports a variety of different monitoring modes: Program Monitor, Entry Data Monitor and Header Monitor. In the Entry Data Monitor mode the program displays the status of a user-defined list of variables. The variables in this list can be displayed in any order.

Ethernet card
One of the several ways of connecting the PLC to your personal computer is via an Ethernet network. To do this, you must then have an Ethernet card in your PC and connect it to the MITSUBISHI ELECTRIC Ethernet module.

Final step
The final step is the last step in an SFC sequence.

Formal parameters
→ ‘Actual parameters’

Function
The function is a program organisation unit (POU) with the class ‘Function’ (FUN). It can be used in the same way as a normal programming instruction. When they are executed, functions deliver precisely one output variable. Functions do not store any internal status data. This means that all a function’s internal variables are always set to 0. When a function is called with the same input parameters it thus always delivers the same output value.

Function block
The function block is a program organisation unit (POU) with the class ‘Function Block’ (FB). It can be used in the same way as a normal programming instruction. Function blocks deliver one or more output variables when they are executed. All the values of the output variables and the internal variables are stored in the function block following execution. This means that function blocks do not necessarily always deliver the same output values when they are called again with the same input parameters. Function blocks can only be called in the form of instances. Be careful not to confuse these function blocks with the graphical function blocks used in the Function Block and Ladder Diagram language!!

Function Block Diagram
Function Block Diagram (FBD) is a programming language. Programming instructions are displayed as graphical blocks which are connected with horizontal and vertical interconnect lines. Be careful not to confuse these graphical function blocks with the Function Block (FB) program organisation units!!

Function blocks in the graphical editors
In the graphical editors most programming instructions are displayed as graphical blocks in the editing screen. These graphical blocks are sometimes referred to as 'function blocks'. Be very careful not to confuse these blocks with the Function Block program organisation units!!

Global variable
Global variables have fixed hardware addresses (absolute addresses). They are declared 'globally' for the entire project and can be referenced from all function block (FB) and program (PRG) program organisation units. They make it possible to exchange data between the individual program modules that make up the project. The global variables are also included as a standard entry in the Project Navigator window, Global_Vars.

Graphic macro
You can define any selected section of a PLC program as a graphic macro. Macros are a useful construct when you need to use the same piece of code several times at different points in your PLC program. Graphic macros are only supported in the graphical editors.

Graphical editor
The following programming languages use graphical editors, which are used in the same way as a standard graphics drawing program: Ladder Diagram (LD), Function Block Diagram (FBD) and Sequential Function Chart (SFC).

Header
Program organisation units (POUs) always consist of a header and a body. The header contains the declarations of all the variables used in the POU, which are organised in a declaration table.

Header Monitor
GX IEC Developer supports a variety of different monitoring modes: Program Monitor, Entry Data Monitor and Header Monitor. Header Monitor mode is fundamentally the same as Entry Data Monitor mode. However, in addition to the user-defined list of variables the monitor also always displays the variables of the active program organisation unit.

Identifier
The identifier is the name allocated to a variable that serves as its symbolic address.

IEC address
The IEC address assigns a fixed memory location to a global variable. The syntax of IEC addressing is defined in the IEC 61131-3 standard. Examples: %IX0 (input X0) , %MW0.450 (data register D450).

IEC editor
The IEC editor is used for programming in the IEC 61131-3 programming languages.

IEC function
→ 'Programming instructions'

IEC function block
→ 'Programming instructions'

IEC Instruction List
GX IEC Developer supports both IEC Instruction List language and MELSEC Instruction List. In IEC Instruction List the program is written according to the IEC 61131-3 standard, using only the instructions from the IEC standard library. However, if you need to do so you can insert MELSEC networks in programs written in IEC Instruction List.

IEC operator
→ ‘Programming instructions’

IEC programming instruction
→ ‘Programming instructions’

Initial step
The initial step is the first step in an SFC sequence.

Initialisation file
MMP701.INI is the ‘initialisation file’ for GX IEC Developer. It is stored in the Windows directory and contains all the parameters necessary for initialising the GX IEC Developer program when it is started.

Input variable
→ ‘Output variable’

Installation program
The installation program INSTALL.EXE copies the contents of the GX IEC Developer program distribution disks to your hard disk and sets the program up for your hardware configuration.

Instance
An instance is a copy of a function block. Function blocks can only be called in the form of instances. Instances are created by declaring them in the headers of program organisation units, and they are called in the body of the POU. Special programming instructions are provided for calling instances.
It is also possible to create global function block instances. These global instances can be used in different program organisation units.

Instructions
→ ‘Programming instructions’, ‘Controller instructions’

Instruction List
Instruction List is a programming language. GX IEC Developer supports two Instruction List variants, IEC Instruction List and MELSEC Instruction List. An Instruction List program consists of a list of controller instructions.

Interconnect mode
Interconnect mode is a special editing mode in the Ladder Diagram editor, with functionality similar to a line-drawing tool in a graphics program. In Interconnect mode you connect programming elements by drawing horizontal and vertical lines between them.

Interface card
One of the several ways of connecting the PLC to your personal computer is with the A7BDE interface card from MITSUBISHI ELECTRIC and a MITSUBISHI network. This is an expansion card which must be installed in your computer. (Not yet supported)

Jump entry point
The jump entry point is the point in an SFC sequence to which a jump is to be performed (i.e. the jump label).

Jump exit point
The jump exit point is the point in an SFC sequence from which a jump is to be performed (i.e. the position of the jump instruction).

Jump instruction
Jump instructions are used to execute a branch from the current POU to another network. In the graphical editors the jump instruction is implemented as a tool. In IEC Instruction List jumps are programmed with the JMP operator.

Label
The word 'Label' is automatically displayed as a dummy variable when you are programming jump instructions in the graphical editors. It must be overwritten with the designator of the actual jump entry point or destination (→ absolute label, network label, system label)

Ladder Diagram
Ladder Diagram is a programming language. The programming elements are contacts (inputs), coils (outputs) and programming instructions which are displayed as graphical blocks. These elements are connected to make circuits using horizontal and vertical interconnect lines. The circuits always start at the power bar on the left.

Library
There are three types of libraries: Standard Library, Manufacturer Library and User Library.

Local variables
Local variables are only accessible within the program organisation units in which they are declared. They are declared in the POU header.

Macro step
In an SFC sequence a macro step is a step that in turn contains another complete SFC sequence.

Manufacturer Library
The Manufacturer Library (Manufacturer_Lib) contains the 'adapted' MELSEC programming instructions, with a special nomenclature for use in the IEC editor (→ 'Programming Instructions' in the Appendix E).

MELSEC editor
The MELSEC editor is used for editing MELSEC Instruction Lists and MELSEC networks.

MELSEC Instruction List
GX IEC Developer supports both IEC Instruction List language and MELSEC Instruction List language. MELSEC Instruction List programs only contain MELSEC networks.

MELSEC network
MELSEC networks are programmed according to the DIN 19239 standard, using only the 'pure' MELSEC instructions.

MELSEC programming instructions
→ 'Programming instructions'

MELSEC upload
The MELSEC upload procedure loads a program into GX IEC Developer from the PLC unit. Only the binary data is read in when this process is carried out.

MITSUBISHI address
The MITSUBISHI address is used to assign a fixed memory location to a global variable. Examples: X0, M1024.

Monitoring mode
GX IEC Developer supports a variety of different monitoring modes: Program Monitor, Entry Data Monitor and Header Monitor.

Network
A network is a logically continuous program section within a program organisation unit.

Network bar
The editing window in which you work on the body of a POU is divided into the network bar and the editing area. The network bar is used for making the network label and title assignments.

Network label
The network label is the name assigned to a network. It can be used as the destination address for jump instructions in PLC programs.

Network list
The network list contains all the networks in a program organisation unit, along with their network labels and titles.

Network title
The network title is a brief description of the network. It is optional, and it is not displayed in the network bar.

Null modem cable
→ ‘RS-232 cable’

Object
GX IEC Developer uses the following objects: The IEC Standard Library (Standard_Lib), the Manufacturer Library (Manufacturer_Lib), the PLC Parameter (PLC_Parameter), the Task Pool (Task_Pool), the Data Unit Type Pool (DUT_Pool), the Global Variables (Global_Vars) and the POU Pool (POU_Pool). These objects are all displayed as standard entries in the Project Navigator window.

Object window
Object windows contain the data of selected objects (e.g. the declaration table for the global variables or a task’s table).

Output variable
Function blocks have input and output variables. The input variables give the function block the values it uses to perform its programmed operations. The output variables pass the results of these operations out of the function block.

PLC Parameter
The PLC parameter (PLC_Parameter) is one of the standard entries in the Project Navigator window. It contains the user-editable parameters of the connected PLC system.

POU Pool
The POU Pool is one of the standard entries displayed in the Project Navigator window (POU_Pool). It contains all the program organisation units used in the project.

Power bar
In the Ladder Diagram graphical editor all circuits start at the vertical power bar on the left of the editing area. This bar is sometimes also referred to as the rail.

Program
1) Program = An application program (e.g. GX IEC Developer);
2) Program = A PLC program, consisting of a series of programming instructions;
3) Program = A program organisation unit classed as a ‘Program’ (PRG). Program POUs can only be executed in tasks.

Program class
The program class of a POU defines its functionality. You can declare POUs as Programs (PRG), Function Blocks (FB) or Functions (FUN).

Program group
The Windows Program Manager organises programs in groups called ‘program groups’. Each group can contain a large number of individual programs (i.e. application programs). The GX IEC Developer installation program automatically creates a program group for GX IEC Developer containing the GX IEC Developer program icon.

Program Monitor
GX IEC Developer supports a variety of different monitoring modes: Program Monitor, Entry Data Monitor and Header Monitor. You can activate the Program Monitor mode in all the editors. This monitor displays the values and switching statuses of the variables visible in the current editor window.

Program organisation unit
Program organisation units (POUs) can be regarded as subprograms or program modules. There are three different POU classes, defined by their functionality: Programs (PRG), Function Blocks (FB) and Functions (FUN). Every POU consists of a header and a body.

Programming elements
Examples of programming elements in the graphical editors include contacts, function blocks, interconnect lines and variable connections.

Programming instructions
GX IEC Developer supports the following groups of programming instructions: IEC programming instructions, MELSEC programming instructions and user programming instructions (→ 'Programming Instructions' in the Appendix E).

Programming language
GX IEC Developer supports the following programming languages for writing PLC programs: Instruction List (IL), Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC).

Project
The project is the element at the top of the GX IEC Developer object hierarchy. It is equivalent to a PLC system.

Project name
The project name is actually the name of a subdirectory. This subdirectory contains all the individual files that form the components of a project.

Project Navigator
The Project Navigator is the control centre of GX IEC Developer. It is used to administrate all the objects that go to make up a project. It is displayed on the screen as a separate window, the Project Navigator window.

Project Navigator tree
The entries in the Project Navigator window are arranged using a hierarchical tree structure. Some of the levels can be expanded and collapsed to display or hide further levels of the tree hierarchy.

Project Navigator window
All the objects contained in a project are listed in the Project Navigator window. This window is opened on the left hand side of the screen as soon as you create or open a project. It always contains the following standard entries: Standard_Lib, Manufacturer_Lib, PLC_Parameter, Task_Pool, DUT_Pool, Global_Vars and POU_Pool.

'Pure' MELSEC Instructions
The 'pure' MELSEC instructions are the original MITSUBISHI MELSEC instructions with the standard nomenclature (e.g. AND=, +, FLOAT). These instructions are only supported in the MELSEC editor (→ 'Programming Instructions' in the Appendix E).

Return instruction
The Return instruction effects a return from a function block to the point in the program from where the function block was called. In the graphical editors the Return instruction is implemented as a tool. In the IEC Instruction List editor returns are implemented with the RET operator.

RS-232 cable
An RS-232 cable wired as a null modem cable is required for connecting the personal computer to a computer link module.

Select mode
When you are working in the programming editors you can choose between Select mode and Edit mode. In Select mode the mouse pointer is displayed as an arrow with which you can select the various programming elements.

Sequence
Sequences are written with the Sequential Function Chart language (SFC). An SFC sequence consists of steps, actions, transitions and transition conditions. Both parallel and selective branching are supported in sequences.

Sequential Function Chart Language
Sequential Function Chart is a programming language in which steps are linked together with conditions referred to as continue conditions. The elements of the SFC language are steps, actions, transitions and transition conditions.

SFC sequence
→ 'Sequence'

Signal configuration
You can set the following signal configuration parameters for the input and output variables of a function block and the input contacts and output coils used in the Ladder Diagram language: Poll for signal status 0 or 1, set and reset.

Standard Library
The Standard Library (Standard_Lib) contains the standard IEC functions (e.g. INT_TO_BOOL) and function blocks (e.g. R_TRIG).

Status bar
The status bar displayed at the bottom of the screen shows information on the current status of the project. You can configure the status bar to suit your personal requirements.

Step
Sequences in the SFC language always consist of several steps connected by transitions. An action is assigned to each step in the sequence.

Step mode
Step mode execution is a debugging aid. In this mode, the program is executed one step at a time. Please note that the connected PLC system must be switched to step run mode when you use this option.

STRING
A STRING data type contains characters.

System Label
The system labels (pointers) are system variables that GX IEC Developer needs for internal management of the project. The number of available system labels depends on the PLC used. One system label is needed for each of the following elements: every instance of a function block, every user-defined function, every event-triggered task, every network label.

Task
A task contains one or more program organisation units of the class 'Program' (PRG). The task controls the execution of its component POUs. Tasks can be event-triggered, interval-triggered or priority-triggered.

Task Pool
The Task Pool (Task_Pool) is one of the standard entries in the Project Navigator window. It contains all the tasks defined in the project.

Text editor
The text editor is used for writing and editing Instruction List programs. Its functionality is similar to that of a standard word processor.

Toolbar
The toolbar displays icons with which the most-used menu commands can be selected directly by clicking with the mouse.

Transition
Transitions connect steps in SFC language sequences. Each transition is allocated a transition condition, i.e. a continue condition that controls how and when control is passed to the next step in the sequence.

Transition condition
A transition condition is a PLC program. It is executed when the preceding SFC step is completed, and if the condition is satisfied control is passed to the next step in the sequence.

Upload
Transfer process from the PLC to GX IEC Developer

User library
A user library gives you the possibility of creating functions and program segments which can be used on a supra-project basis. The program segments are grouped in a library and can be used in any project. Following updating, changes made to the library can be applied to other projects. A user library can contain all the objects that are also available in a project: Global Variables List, Structured Data Types, Arrays and Program Organisation Units. You can assign a password to protect the know-how in the library. The name of the user library file has the file extension SUL.

User programming instructions
Programming instructions written by the user. → ‘Programming instructions’.

Variable
Variables can be compared with operands. They contain the values of inputs, outputs or internal memory locations in the PLC. GX IEC Developer makes a distinction between global and local variables. Variables are declared and administered in declaration tables; they are defined with a class, an identifier, an absolute address (global variables only) and a data type.

Wizard
The Wizard is a special aid that can be activated and deactivated by editing the initialisation file MMP701.INI. When the Wizard is activated the program can automatically generate a program organisation unit and a task for you when you create a new project.

B Keyboard Commands

B.1 GX IEC Developer Menu Keys

Key(s)	Functions
Alt	Activate/deactivate the Menu Bar
F10	Activate/deactivate the Menu Bar
→	Move through Menu Bar to the right (GX IEC Developer menus and Windows control menu)
←	Move through Menu Bar to the left (GX IEC Developer menus and Windows control menu)
A - Z	When Menu Bar or menu activated: Select corresponding menu or menu option/command (e.g. L for Online)
Enter	Activate highlighted menu or menu option/command
↑	Move highlight up through menu options
↓	Move highlight down through menu options
Esc	Exit menu/selection
Alt F4	Quit GX IEC Developer (**Project - Quit**)
F1	Activate Help system
Alt Tab	Open Task List (list of all application programs currently running) and select program
Ctrl Esc	Open Windows Start menu

Tab. B-1: *GX IEC Developer menu keys*

B.2 Common Keys

Key(s)	Function
F4	Show next compile/check error **(Window - Show Next Error)**
Alt F4	Quit the application **(Project - Quit)**
Alt 0	Activate the navigator window **(Window - Navigator)**
Ctrl Alt W	Download the current project to the PLC **(Project - Transfer - Download to PLC...)**
Alt S	Start, stop or reset the PLC or view the PLC operating status **(Online - Start/Stop PLC)**
Shift Alt B	Build the project **(Project - Build)**
Shift Alt C	Rebuild the project **(Project - Rebuild All)**
Shift Alt D	Setup modem for communication **(Online - Modem)**
Shift Alt E	View the system errors **(Debug - System Errors)**
Alt U	View the user defined errors **(Debug - User Errors)**
Ctrl N	Create a new project (not LD editor)
Ctrl N	Insert a contact **(Tools - Network Elements - Contact)** (LD editor only)
Ctrl O	Open a project (not LD and FBD editor)
Ctrl O	Insert an output variable field **(Tools - Network Elements - Output Variable)** (LD and FBD editor only)
Shift Ctrl F8	Open an Entry Data Monitor window **(Online - Entry Data Monitor)**
Shift Ctrl D	Change the program online without stopping the PLC **(Project - Online Program Change)**
Shift Esc	Enable or disable the monitoring mode **(Online - Monitoring Mode)**
Shift F4	Show previous compile/check error **(Window - Show Previous Error)**

Tab. B-2: Common keys

B.3 Project Navigator Keys

Key(s)	Function
↑	Move highlight up through Project Navigator Window
↓	Move highlight down through Project Navigator Window
→	Scroll window contents to the right
←	Scroll window contents to the left
PgUp	Page window contents up one screen
PgDn	Page window contents down one screen
Ctrl + +: num. keypad	Display lower tree levels of selected entry (**Tools** - **Expand**)
Ctrl - -: num. keypad	Hide lower tree levels of selected entry (**Tools** - **Collapse**)
Alt Back space	Undo the last action **(Edit - Undo)**
Ctrl Z	Undo the last action
Ctrl X	Cut selected entry and copy to the clipboard (not available for all entries!) (**Edit** - **Cut**)
Shift Del	Cut selected entry and copy to the clipboard
Ctrl C	Copy selected entry to clipboard (not available for all entries!) (**Edit** - **Copy**)
Ctrl V	Paste data from the clipboard to highlight position (**Edit** - **Paste**)
Shift Ins	Paste data from the clipboard to highlight position
Del	Delete selected entry (not available for all entries!) (**Edit** - **Delete**)
Alt F3	Find the specified text **(Edit - Find)**
F3	Find the next occurrence of the specified text
Shift F3	Find the previous occurrence of the specified text
Alt Ctrl F3	Replace specific text with different text **(Edit - Replace)**
Ctrl F3	Replace the next occurrence of the specified text
Ctrl Shift F3	Replace all occurrences of the specified text
Ctrl Ins	Insert copied contents
Ctrl F8	Start monitoring of the active body **(Online - Start Monitoring)**
Alt F8	Stop monitoring of the active body **(Online - Stop Monitoring)**
Ctrl F9	Modify the variable value **(Online - Modify variable value)**
Enter	Open object window or dialogue box of selected entry (**Object** - **Open**)
Alt Enter	Edit the properties of the current object **(Object - Information...)**

Tab. B-3: *Project Navigator keys*

Key(s)	Function
Alt C	Check the selected object **(Object - Check)**
Alt K	Create a new task (**Object - New - Task)**
Alt N	Create a new Program Organisation Unit (**Object - New - POU)**
Shift Alt A	Create a new Sequential Function Chart action **(Object - New - Action)**
Ctrl P	Print the active object **(Object - Print)**
Ctrl Q	Show the print preview of the active object **(Object - Print Preview)**
Ctrl S	Save the selected object **(Object - Save)**
Ctrl F4	Close Project Navigator Window (i.e. close project) (**Project- Close**)

Tab. B-3: *Project Navigator keys*

B.4 Object Window Keys

Key(s)	Function
Ctrl Tab	Step through all opened windows
Ctrl F6	Switch to next window
Ctrl Shift F6	Switch to previous window
PgUp	Page window contents one screen up
PgDn	Page window contents one screen down
Ctrl PgUp	Scroll window contents to the left
Ctrl PgDn	Scroll window contents to the right
Alt -	Open the Windows control menu of the active window
Ctrl F4	Close the active window (**Object - Close**)
Ctrl Shift T	Arrange all opened windows side by side (**Window - Tile**)
↑	1) Move pointer up in window 2) Following selection of **Move** (Size) in the control menu: Move window up (resize window)
↓	1) Move pointer down in window 2) Following selection of **Move** (Size) in control menu: Move window down (resize window)
→	1) Move pointer right in window 2) Following selection of **Move** (Size) in control menu: Move window right (resize window)
←	1) Move pointer left in window 2) Following selection of Move (Size) in control menu: Move window left (resize window)
Enter	1) Storing the new window position or window size 2) In Project Navigator Window: Open the object window or dialogue box of the selected entry
Ctrl S	Save object (**Object** - **Save**)
Alt C	Check object (**Object** - **Check**)
Ctrl P	Print object (**Object** - **Print**)
Alt Enter	View object information (**Object** - **Information**)

Tab. B-4: *Object window keys*

B.5 Dialogue Box Keys

Key(s)	Function
Alt F6	Move to next dialogue box
Alt Shift F6	Move to previous dialogue box
Alt	Activate/deactivate control menu button
F10	Activate/deactivate control menu button
Tab	Step through options from top to bottom / from left to right
Shift Tab	Step through options from bottom to top / from right to left
Alt A - Z	Execute option (e.g. L for Latch Range)
Enter	1) Control menu button selected: Open control menu 2) Option selected: Execute option (often the same as clicking the **OK**! button) 3) List box: Activate selected element
Alt F4	Close dialogue box (often the same as selecting the **Cancel**! button)
Esc	Close dialogue box (often the same as selecting the **Cancel**! button)
→ ←	Step through options
↑ ↓	1) Step through options 2) List box: Page list up/down one screen at a time
Home	1) List box: Move to first list element 2) Text box: Move to first character
End	1) List box: Move to last list element 2) Text box: Move to last character
Space	1) List box: Activate/deactivate list element 2) Activate/deactivate check boxes

Tab. B-5: Dialogue box keys

B.6 Declaration Table Keys

Key(s)	Function
↑	Move up one line at a time
↓	Move down one line at a time
→	Move right one cell at a time
←	Move left one cell at a time
PgUp	Page window contents up one screen at a time
PgDn	Page window contents down one screen at a time
Home	1) Move to cell in the first column 2) Editing mode: Move to first character in the cell
End	1) Move to cell in the last column 2) Editing mode: Move to last character in the cell
Tab	Step through all cells from left to right
Shift Tab	Step through all cells from right to left
Ctrl Home	Move highlight to beginning of table
Ctrl End	Move highlight to end of table
Shift ↑	Extend or reverse continuous selection up one line at a time
Shift ↓	Extend or reverse continuous selection down one line at a time
Shift Home	Select current line
Shift End	Select current line
Shift Ctrl Home	Select all lines from the first line to the current line
Shift Ctrl End	Select all lines from the current line to the last line
Alt ↓	Open dialogue box or list box of a selected cell containing an arrow symbol
Space	Activate editing mode in the current cell
Enter	Deactivate editing mode in the current cell
Alt Enter	Edit the properties of the current object **(Object - Information...)**
Alt C	Check the selected object **(Object - Check)**
Ctrl P	Print the active object **(Object - Print)**
Ctrl Q	Show the print preview of the active object **(Object - Print Preview)**
Ctrl S	Save the selected object **(Object - Save)**
Del	1) Delete selected range 2) Editing mode: Delete character to right of cursor (**Edit** - **Delete**)
Back space	1) Delete selected range 2) Editing mode: Delete character to left of cursor
Ctrl X	Cut selected range and copy to clipboard (**Edit** - **Cut**)
Shift Del	Cut selected entry and copy to the clipboard

Tab. B-6: *Declaration table keys*

Key(s)	Function
Ctrl C	Copy selected range to clipboard (**Edit** - **Copy**)
Ctrl V	Paste data from clipboard to current position (**Edit** - **Paste**)
Shift Ins	Paste data from the clipboard to highlight position
Ctrl Z	Undo the last action
Ctrl Ins	Insert copied contents
Shift Enter	1) From any cell: Move to first column of the next line 2) From any cell in the last line: Insert a new line
Alt Back space	Undo the last command (not supported for all operations!) (**Edit** - **Undo**)
Ctrl Back space	Undo the last Undo command (not supported for all operations!) (**Edit - Redo**)
Alt F3	Find (**Edit - Find**)
F3	Find Next (**Edit** - **Find Next**)
Shift F3	Find the previous occurrence of the specified text
Alt Ctrl F3	Replace specific text with different text **(Edit - Replace)**
Ctrl F3	Replace the next occurrence of the specified text
Ctrl Shift F3	Replace all occurrences of the specified text
Ctrl F8	Start monitoring of the active body **(Online - Start Monitoring)**
Alt F8	Stop monitoring of the active body **(Online - Stop Monitoring)**
Shift Alt F8	Monitor the header of the active body **(Online - Monitor Header)**
Ctrl + +: num. keypad	First cell in the line is marked by a - sign: Display all lines in the comment column (**Tools** - **Expand declaration**)
Ctrl - -: num. keypad	First cell in the line is marked by a + sign: Display only one line in the comment column (**Tools** - **Collapse declaration**)

Tab. B-6: *Declaration table keys*

B.7 Entry Data Monitor Keys

Key(s)	Function
Del	Delete rows from the EDM **(Edit - Delete Rows)**
F2	Insert objects into the EDM **(Edit - Insert Objects)**
F3	Insert next object into the EDM **(Edit - Next Object)**
Ins	Insert empty row into the EDM **(Edit - Insert Row)**
Shift F10	Show context menu
Ctrl F8	Start monitoring of the active body **(Online - Start Monitoring)**
Alt F8	Stop monitoring of the active body **(Online - Stop Monitoring)**
Ctrl + +: num. keypad	First cell in the line is marked by a - sign: Display all lines in the comment column (**Tools** - **Expand declaration**)
Ctrl - -: num. keypad	First cell in the line is marked by a + sign: Display only one line in the comment column (**Tools** - **Collapse declaration**)

Tab. B-7: *Entry Data Monitor keys*

B.8 Show Generated Code Keys

Key(s)	Function
Ctrl Ins	Copy the whole editor content to the clipboard **(Edit - Copy All)**
Ctrl C	Copy the whole editor content to the clipboard

Tab. B-8: *Show generated code keys*

B.9 Show Verify Differences Keys

Key(s)	Function
Tab	Go to the next difference **(Tools - Next Difference)**
Shift Tab	Go to the previous difference **(Tools - Previous Difference)**
Ctrl +	Open the according body and go to the related position **(Tools - Go to Editor)**
Ctrl C	Copy the whole content of the window to the clipboard in plain text format **(Edit - Copy All)**

Tab. B-9: *Show verify differences keys*

B.10 Device Editor Keys

Key(s)	Function
↑	Move up one cell
↓	Move down one cell
→	Move right one cell
←	Move left one cell
Ctrl ↑	Move to top of current data range
Ctrl ↓	Move to bottom of current data range
Ctrl →	Move to right edge of current data range
Ctrl ←	Move to left edge of current data range
PgUp	Page up
PgDn	Page down
Ctrl PgUp	Page left
Ctrl PgDn	Page right
Home	Select cell in first column of current row
End	Select cell in last column of current row
Ctrl Home	Select cell in first column of first row (top left)
Ctrl End	Select cell in last column of last row (bottom right)
Tab	Step through all cells from left to right
Shift Tab	Step through all cells from right to left
Enter	Step down through all cells
Shift Enter	Step up through all cells
Shift ↑	Extend or cancel selection upwards one line at a time
Shift ↓	Extend or cancel selection downwards one line at a time
Shift →	Extend or cancel selection right one cell at a time
Shift ←	Extend or cancel selection left one cell at a time
Ctrl Shift ↑	Extend or cancel selection to top of current data range
Ctrl Shift ↓	Extend or cancel selection to bottom of current data range
Ctrl Shift →	Extend or cancel selection to right edge of current data range
Ctrl Shift ←	Extend or cancel selection to left edge of current data range
Shift PgUp	Extend or cancel selection up one page
Shift PgDn	Extend or cancel selection down one page
Ctrl Shift PgUp	Extend or cancel selection left one page
Ctrl Shift PgDn	Extend or cancel selection right one page
Shift Home	Extend or cancel selection to beginning of current line

Tab. B-10: *Device Editor keys*

Key(s)	Function
Shift End	Extend or cancel selection to end of current line
Ctrl Shift Home	Extend or cancel selection to first column in first line (top left)
Ctrl Shift End	Extend or cancel selection to last column in last line (bottom right)
Shift Space	Select current line
Ctrl Space	Select current column
Ctrl Shift Space	Select entire table
Ctrl A	Select entire table
Del	Delete selected range
Ctrl X	Cut selected range to clipboard
Ctrl C	Copy selected range to clipboard
Ctrl V	Paste contents of clipboard
Ctrl Ins	Copy selected range to clipboard
Shift Ins	Paste contents of clipboard
Shift Del	Cut selected range to clipboard
F2	Activate editing mode for selected cell
Back space	Clear contents of selected cell and activate editing mode
Ctrl Enter	Terminate editing mode
Esc	Abort editing mode
Enter	Toggle value of selected bit device between 0 and 1

Tab. B-10: *Device Editor keys*

B.11 Text Editor Keys (IL, MIL, ST)

Key(s)	Function
Alt T E	Activate editing mode (**Tools - Edit Network**)
Alt T D	Activate Select mode (**Tools - End Edit**)
Alt N	Display the New Variable dialogue **(Tools - New Variable)**
Shift F12	Start the variable tracking **(Tools - Tracking - Start Tracking)**
Ctrl F12	Next variable occurrence **(Tools - Tracking - Next)**
Ctrl F11	Previous variable occurrence **(Tools - Tracking - Previous)**
F2	Display a list of available operands **(Tools - List Operands)** (ST only)
Shift F2	Display a list of available operators **(Tools - List Operators)** (ST only)
↑ ↓ → ←	1) Select mode: Move cursor/pointer in window 2) Editing mode: Scroll through Instruction List
Enter	1) Select mode - Pointer on network bar: Open dialogue box for entry of network labels and titles 2) Editing mode: Select/enter new line
Space **(2x)**	Select mode - pointer on network bar: Open dialogue box for assigning network label and title
Space	1) Select mode - pointer on network bar: Activate network 2) Select mode - cursor/pointer in editing window: Activate editing mode
Space ↑ ↓	Pointer on bottom border line of network bar: Change size of network area
Space → ←	Pointer on column dividing line: Change column width
Tab	1) Line without text: Select next column 2) Line with text: Move text to next column
F2	1) Cursor/pointer in first column: Open programming instructions list 2) Cursor/pointer in second column: Open variables list
PgUp	Page window contents up one screen
PgDn	Page window contents down one screen
Ctrl PgUp	Scroll window contents to the right
Ctrl PgDn	Scroll window contents to the left
Home	Move to first character in the line
End	Move to last character in the line
Ctrl Home	Move to beginning of Instruction List
Ctrl End	Move to end of Instruction List
Shift →	Extend or reverse selection to the right one character at a time
Shift ←	Extend or reverse selection to the left one character at a time
Shift ↑	Extend or reverse selection up one line at a time
Shift ↓	Extend or reverse selection down one line at a time
Shift Ctrl Home	Select all lines from the first line to the current line

Tab. B-11: *Text editor keys (IL, MIL, ST)*

Key(s)	Function
Shift Ctrl End	Select all lines from the current line to the last line
Alt Enter	Edit the properties of the current object **(Object - Information...)**
Alt C	Check the selected object **(Object - Check)**
Ctrl P	Print the active object **(Object - Print)**
Ctrl Q	Show the print preview of the active object **(Object - Print Preview)**
Ctrl S	Save the selected object **(Object - Save)**
Del Back space	1) Delete selected range 2) Delete character to right of cursor (**Edit** - **Delete**)
	1) Delete selected range 2) Delete character left of cursor
Ctrl X	Cut selected range and copy to clipboard (**Edit** - **Cut**)
Shift Del	Cut selected entry and copy to the clipboard
Ctrl C	Copy selected range to clipboard (**Edit** - **Copy**)
Ctrl V	Paste data from clipboard to current position (**Edit** - **Paste**)
Shift Ins	Paste data from the clipboard to highlight position
Ctrl Ins	Insert copied contents
Alt F3	Find (**Edit** - **Find**)
F3	Find Next (**Edit** - **Find Next**)
Shift F3	Find the previous occurrence of the specified text
Alt Ctrl F3	Replace specific text with different text **(Edit - Replace)**
Ctrl F7	(Edit - Toggle Bookmark)
F7	(Edit - (Next Bookmark)
Shift F7	(Edit - Previous Bookmark)
Ctrl F3	Replace the next occurrence of the specified text
Ctrl Back space Shift F3	Replace all occurrences of the specified text
Alt	Undo last command (not supported for all operations!) (**Edit** - **Undo**)
Ctrl Back space	Undo the last action
Ctrl	Undo last Undo command (not supported for all operations!) (**Edit** - **Redo**)
Alt A	Insert a new network after the current network (**Edit** - **New Network** - **After**)
Alt B	Insert new network before the current network (**Edit** - **New Network** - **Before**)
Ctrl F8	Start monitoring of the active body **(Online - Start Monitoring)**
Shift F8	Start program monitoring **(Online - Start Monitoring - Body)** (ST only)
Ctrl F8	Start program monitoring **(Online - Start Monitoring - Split Window)** (ST only)
Alt F8	Stop monitoring of the active body **(Online - Stop Monitoring)**
Shift Alt F8	Monitor the header of the active body **(Online - Monitor Header)**
Ctrl F9	Modify the variable value **(Online - Modify variable value)**

Tab. B-11: *Text editor keys (IL, MIL, ST)*

B.12 Graphical Editor Keys (FBD, LD)

Key(s)	Function
Ctrl A	Activate Select mode (**Tools** - **Select Mode**)
Ctrl T	Activate Interconnect mode (draw lines) (**Tools** - **Interconnect Mode**)
Ctrl B	Activate the auto connect mode **(Tools - Auto Connect)**
Ctrl M	Select the comment placement mode **(Tools - Network Elements - Comment)**
Shift F12	Start the variable tracking **(Tools - Tracking - Start Tracking)**
Ctrl F12	Next variable occurrence **(Tools - Tracking - Next)**
Ctrl F11	Previous variable occurrence **(Tools - Tracking - Previous)**
↑ ↓ → ←	Move pointer
Space **(2x)**	1) Pointer on network bar: Open dialogue box for entering network labels and titles 2) Pointer on a contact: Open dialogue box for signal configuration F"Helvetica"P10
Enter	Pointer on network bar: Open dialogue box to assign network label and title
Space ↑ ↓	Pointer on lower border of network bar: Change size of network area
PgUp	Page window contents up by one screen
PgDn	Page window contents down by one screen
Ctrl PgUp	Scroll window contents to the right
Ctrl PgDn	Scroll window contents to the left
Space	1) Select programming element 2) Select network bar 3) Activate editing mode to overwrite the dummy names Var and Label
F2	1) Editing mode for overwriting dummy variable name Var: Open list of variables 2) Open list of programming instructions
Space ↑ → ↓ ←	1) Move selected programming element 2) Following selection of **Tools** - **Open Row/Column**: Increase row/column spacing 3) Following selection of **Tools** - **Comment:** Draw a comment box 4) Interconnect mode: Draw interconnect lines
Space ↑ → ↓ ←	Select a continuous group of program elements
Shift ↑ → ↓ ←	Select several program elements individually
Alt Enter	Edit the properties of the current object **(Object - Information...)**
Alt C	Check the selected object **(Object - Check)**
Ctrl P	Print the active object **(Object - Print)**
Ctrl Q	Show the print preview of the active object **(Object - Print Preview)**
Ctrl S	Save the selected object **(Object - Save)**
Del	1) Delete selected range 2) Delete character right of cursor (**Edit** - **Delete**)
Back space	Delete character left of cursor
Ctrl X	Cut selected range and copy to clipboard (**Edit** - **Cut**)
Shift Del	Cut selected entry and copy to the clipboard

Tab. B-12: *Graphical editor keys (FBD, LD)*

Key(s)	Function
Ctrl C	Copy selected range to clipboard (**Edit** - **Copy**)
Ctrl V	Paste data from clipboard to current position (**Edit** - **Paste**)
Shift Ins	Paste data from the clipboard to highlight position
Ctrl Ins	Insert copied contents
Alt F3	Find (**Edit** - **Find**)
F3	Find Next (**Edit** - **Find Next**)
Shift F3	Find the previous occurrence of the specified text
Alt Ctrl F3	Replace specific text with different text **(Edit - Replace)**
Ctrl F3	Replace the next occurrence of the specified text
Ctrl Shift F3	Replace all occurrences of the specified text
Alt Ctrl C	Toggle the signal configuration **(Edit - Toggle Signal Configuration)**
Alt Back space	Undo last command (not supported for all operations!) (**Edit** - **Undo**)
Ctrl Z	Undo the last action
Ctrl Back space	Undo last Undo command (not supported for all operations!) (**Edit** - **Redo**)
1	Insert input contact (**Tools** - **Contact**) (LD only)
2	Insert input contact negation (**Tools** - **Contact Negation**) (LD only)
3	Insert input L-connect contact (**Tools** - **L-Connect Contact**) (LD only)
4	Insert input L-connect contact negation (**Tools** - **L-Connect Contact Negation**) (LD only)
7	Insert output coil **(Tools - Network Elements - Coil)** (LD only)
Ctrl J	Insert jump instruction **(Tools - Network Elements - Jump)**
Ctrl R	Insert return instruction **(Tools - Network Elements - Return)**
9	Insert input variable **(Tools - Network Elements - Input Variable)**
0	Insert output variable **(Tools - Network Elements - Output Variable)**
6	Draw horizontal line segment **(Tools - Network Elements - Horizontal Line Segment)**
5	Draw vertical line segment **(Tools - Network Elements - Vertical Line Segment)**
8	Insert programming instruction (Function Block) (**Tools** - **Function Block**)
Ctrl W	Increase horizontal spacing (**Tools** - **Open Row**)
Ctrl U	Increase vertical spacing (**Tools** - **Open Column**)
Ctrl Shift L	Insert network label **(Tools - Network Elements - Label of the Network)**
Ctrl Shift Del	Delete line mode **(Tools - Network Elements - Delete Line Mode)**
Alt 1	Insert graphic macro 1 (**Tools** - **Custom 1**)
Alt 2	Insert graphic macro 2 (**Tools - Custom 2**)
Alt 3	Insert graphic macro 3 (**Tools - Custom 3**)
Alt 4	Insert graphic macro 4 (**Tools - Custom 4**)

Tab. B-12: *Graphical editor keys (FBD, LD)*

Key(s)	Function
Alt 5	Insert graphic macro 5 (**Tools - Custom 5**)
Alt N	Display the New Variable dialogue **(Tools - New Variable)**
Alt A	Insert new network after the current network (**Edit - New Network - After**)
Alt B	Insert new network before the current network (**Edit - New Network - Before**)
Ctrl F8	Start monitoring of the active body **(Online - Start Monitoring)**
Alt F8	Stop monitoring of the active body **(Online - Stop Monitoring)**
Alt F9	Toggle the selected variable **(Online - Toggle variable)**
Shift Alt F8	Monitor the header of the active body **(Online - Monitor Header)**
Ctrl F9	Modify the variable value **(Online - Modify variable value)**
Tab	Step through all the variable objects in the current network forwards, selecting them one after another.
Shift Tab	Step through all the variable objects in the current network backwards, selecting them one after another.
+ (+: num. keypad)	Function block is activated: Increment number of input variables (pins) (**Edit - Increment Pins**)
- (-: num. keypad)	Function block is activated: Decrement number of input variables (pins) (**Edit - Decrement Pins**)
Shift Ctrl M	Toggle the view mode for the addresses **(View Mode - Toggle)**

Tab. B-12: *Graphical editor keys (FBD, LD)*

B.13 Sequential Function Chart Editor Keys

Key(s)	Function
↑ ↓ → ←	Move highlight
Ctrl ←	Highlight on a connection line: Move highlight to the leftmost branch in the row
Ctrl →	Highlight on a connection line: Move highlight to the rightmost branch in the row
Ctrl ↑	Move highlight to the next branch above or to the Initial Step
Ctrl ↓	Move highlight to next branch below or to the Final Step
Shift ←	Extend selection to the element to the left of the highlight
Shift →	Extend the selection to the element to the right of the highlight
Shift ↑	Extend the selection up one element
Shift ↓	Extend the selection down one element
Shift Ctrl ←	Highlight on a connection line: Extend the selection to the leftmost element in the row
Shift Ctrl →	Highlight on a connection line: Extend the selection to the rightmost element in the row
Shift Ctrl ↑	Extend the selection to the next element above or to the Initial Step
Shift Ctrl ↓	Extend the selection to the next element below or to the Final Step
Tab	Move the highlight to the next element
Shift Tab	Extend the selection to the next element
Home	Move the highlight to the first column
End	Move the highlight to the last column
Ctrl Home	Move the highlight to the beginning of the program
Ctrl End	Move the highlight to the end of the program
Shift Home	Select all elements between the current highlight position and the left column
Shift End	Select all elements between the current highlight position and the right column
Shift Ctrl Home	Select all elements between the current highlight position and the beginning of the program
Shift Ctrl End	Select all elements between the current highlight position and the end of the program
Enter	1) Highlight on an unnamed Transition or an unnamed Step: Switch to Editing mode 2) Highlight on a named Step: Open window of associated POU 3) Highlight on a Macro Step: Open Macro Program window
Esc	1) Undo all selections
Del	Delete selected elements
PgUp	Page window contents up by one screen
PgDn	Page window contents down by one screen
F2	1) Cursor/pointer in first column: Open programming instructions list 2) Cursor/pointer in second column: Open variables list
Ctrl S	Save the selected object **(Object - Save)**
Alt Enter	Edit the properties of the current object **(Object - Information...)**
Alt C	Check the selected object **(Object - Check)**
Ctrl P	Print the active object **(Object - Print)**
Ctrl Q	Show the print preview of the active object **(Object - Print Preview)**
Alt Back space	Undo last command (not supported for all operations!) (**Edit** - **Undo**)
Ctrl Z	Undo the last action
Ctrl Back space	Undo last Undo command (not supported for all operations!) (**Edit** - **Redo**)
Ctrl X	Cut selected range and copy to clipboard (**Edit** - **Cut**)

Tab. B-13: *Sequential Function Chart editor keys*

Key(s)	Function
Shift Del	Cut selected entry and copy to the clipboard
Ctrl C	Copy selected range to clipboard (**Edit** - **Copy**)
Ctrl V	Paste data from clipboard to current position (**Edit** - **Paste**)
Ctrl Ins	Insert copied contents
Shift Ins	Paste data from the clipboard to highlight position
Alt F3	Find (**Edit** - **Find**)
F3	Find Next (**Edit** - **Find Next**)
Shift F3	Find the previous occurrence of the specified text
Alt Ctrl F3	Replace specific text with different text **(Edit - Replace)**
Ctrl F3	Replace the next occurrence of the specified text
Ctrl Shift F3	Replace all occurrences of the specified text
Ctrl U	Insert Step/Transition unit at the current position (**Edit** - **Insert** - **Step/Trans Unit**)
Ctrl F	Insert Step at the current position (**Edit - Insert - Step**)
Ctrl T	Insert Transition at the current position (**Edit - Insert - Transition**)
Ctrl L	Insert left divergence from the current position (**Edit - Insert - Left Divergence**)
Ctrl R	Insert right divergence from the current position (**Edit - Insert - Right Divergence**)
Shift Ctrl L	Insert left convergence to the current position (**Edit - Insert - Left Convergence**)
Shift Ctrl R	Insert right convergence to the current position (**Edit - Insert - Right Convergence**)
Ctrl Shift J	Insert Label (jump destination/entry step) (**Edit - Insert - Label**)
Ctrl J	Insert Jump (jump/exit step) (**Edit - Insert - Jump**)
Alt N	Display the New Variable dialogue **(Tools - New Variable)**
Ctrl I	Open (zoom) Step, Transition or Macro Program window (**Tools - Zoom Into**)
Ctrl B	Undo zoom operation (**Tools - Zoom Back**)
Ctrl F8	Start monitoring of the active body **(Online - Start Monitoring)**
Shift Alt F8	Monitor the header of the active body **(Online - Monitor Header)**
Ctrl F9	Modify the variable value **(Online - Modify variable value)**

Tab. B-13: *Sequential Function Chart editor keys*

B.14 Keys for Checking and Downloading Programs

Key(s)	Function
Shift Alt C	Compile project (**Project - Rebuild All**)
Ctrl Alt W	Download project to CPU (**Project - Transfer - Medoc -> PLC**)
Shift Ctrl D	Change program in Online mode operation (**Project - Online Program Change**)
Alt C	Check object (**Object - Check**)
Ctrl F8	Start program monitoring (**Online - Start Monitoring**)
Alt F8	Stop program monitoring (**Online - Stop Monitoring**)
Shift Alt F8	Open monitoring window for variable status of current Header **(Online - Monitor Header**)
Shift Ctrl F8	Open Entry Data Monitor (**Online - Entry Data Monitor**)
Shift Esc	Activate Monitoring mode (**Online - Monitoring Mode**)
Alt S	Switch PLC CPU to Start or Stop mode (**Online - Start/Stop PLC**)
F5	Go to breakpoint (**Debug - Go To Breakpoint**)
F7	Execute program in stop mode, stopping after each step (**Debug - Step**)
Shift Alt E	Display contents of system error registers (**Debug - System Errors**)
Alt U	Associate error messages to error flags (**Debug - User Errors**)

Tab. B-14: *Keys for checking and downloading programs*

B.15 Online Help System Keys

Key(s)	Function
F1	Activate Help system
I	**Contents** button
S	**Search** button
Z	**Back** button
R	**History** button
<	<< button (previous topic)
>	>> button (next topic)
Alt	Activate Menu Bar
Tab	Select one of the icons/buttons on the screen Step through the icons/buttons clockwise
Shift Tab	Select one of the icons/buttons on the screen Step through the icons/buttons anticlockwise
Ctrl Tab	Select all the icons/buttons on the screen
Ctrl Enter	Copy the current help topic text to the clipboard
Shift Enter	Copy the clipboard contents to the **Annotate** dialogue box
Alt F4	Close the Help window

Tab. B-15: *Online Help System Keys*

For more details on using the online Help system select **How to use help** in the **Help** menu.

C Tools

C.1 General Tool Icons

Tool	Function	Menu command	Key(s)
📂	Open an existing project	**Project - Open**	
💾	Save an opened project	**Project - Save**	
🖨	Print the active object	**Project - Print**	
🔍	Shows a print preview of the active object	**Project - Print Preview**	
✂	Cut the selected object to the clipboard	**Edit - Cut**	Ctrl X
	Copy the selected object to the clipboard	**Edit - Copy**	Ctrl C
📋	Paste object from the clipboard	**Edit - Paste**	Ctrl V
↶	Undo the last action	**Edit - Undo**	Alt ←
↷	Redo the previously undone action	**Edit - Redo**	Ctrl ←
	Edit the properties of current selection	**Object - Information**	Alt Enter
	Open the cross reference browser	**Project - Browse**	
	Start the check routine for the currently selected object	**Object - Check**	Alt C
	Build the current project (compile changes)	**Project - Build**	Shift Alt B
	Rebuild the current project (compile project)	**Project - Rebuild All**	Shift Alt C
	Enable or disable the monitoring mode	**Online - Monitoring Mode**	Shift Esc
	Activate Online Change mode	**Online - Online-Change Mode**	
	Download the current project to the PLC	**Project - Transfer - Download to PLC**	Ctrl Alt W
	Upload the project from the PLC	**Project - Transfer - Upload from PLC**	

Tab. C-1: *General tool icons*

C.2 Instruction List Editor Tool Icons

Tool	Function	Menu command	Key(s)
	Insert a new network before the current network	**Edit - New Network - Before**	Alt B
	Insert a new network after the current network	**Edit - New Network - After**	Alt A
	Activate Select mode, End Editing mode	**Tools - End Edit**	
	Activate Editing mode, End Select mode	**Tools - Edit Network**	

Tab. C-2: *Instruction List Editor Tool Icons*

C.3 Ladder Diagram and Function Block Diagram Editor Tool Icons

Tool	Function	Menu command	Key(s)
	Insert a new network before the current network	**Edit - New Network - Before**	Alt B
	Insert a new network after the current network	**Edit - New Network - After**	Alt A
	Activate Select mode, End Editing mode (toggles with Interconnect mode)	**Tools - Select Mode**	Ctrl A
	Activate Interconnect mode, Line tool (toggles with Select mode)	**Tools - Network Elements - Interconnect Mode**	Ctrl T
	Insert input contact (Ladder Diagram only)	**Tools - Network Elements - Contact**	1
	Insert input contact negation (Ladder Diagram only)	**Tools - Network Elements - Contact Negation**	2
	Insert input L-contact (Ladder Diagram only)	**Tools - Network Elements - L-Connection Contact**	3
	Insert input L-contact negation (Ladder Diagram only)	**Tools - Network Elements - L-Connection Contact Negation**	4
	Insert vertical line (Ladder Diagram only)	**Tools - Network Elements - Vertical Line Segment**	5
	Insert horizontal line (Ladder Diagram only)	**Tools - Network Elements - Horizontal Line Segment**	6
	Insert output coil (Ladder Diagram only)	**Tools - Network Elements - Coil**	7
	Insert programming instruction (Function Block)	**Tools - Network Elements - Function block**	8
VAR=	Insert input variable	**Tools - Network Elements - Input Variable**	9
=VAR	Insert output variable	**Tools - Network Elements - Output Variable**	0
	Activate Auto Connect mode	**Tool - Auto Connect**	Ctrl B
	Insert Jump instruction	**Tools - Network Elements - Jump**	Ctrl J
	Insert Return instruction	**Tools - Network Elements - Return**	Ctrl R
	Increase vertical spacing	**Tools - Open Row**	Ctrl U
	Increase horizontal spacing	**Tools - Open Column**	Ctrl W
	Insert comment	**Tools - Network Elements - Comment**	Ctrl M

Tab. C-3: *Ladder Diagram and Function Block Diagram editor tool icons*

C.4 Sequential Function Chart Editor Tool Icons

Tool	Function	Menu command	Key(s)
	Insert a Step/Transition unit at the current position	**Edit - Insert - Step/Trans Unit**	Ctrl U
	Insert a Step at the current position	**Edit - Insert - Step**	Ctrl F
	Insert a Transition at the current position	**Edit - Insert - Transition**	Ctrl T
	Insert a left divergence from the current position	**Edit - Insert - Left Divergence**	Ctrl L
	Insert a right divergence from the current position	**Edit - Insert - Right Divergence**	Ctrl R
	Insert a left convergence to the current position	**Edit - Insert - Left Convergence**	Shift Ctrl L
	Insert a right convergence to the current position	**Edit - Insert - Right Convergence**	Shift Ctrl R
	Insert a Label (entry step/jump destination)	**Edit - Insert - Label**	Ctrl Shift J
	Insert a Jump (exit step jump instruction)	**Edit - Insert - Jump**	Ctrl J
	Open (zoom) a Macro Program window	**Tools - Zoom Macro**	
	Open (zoom) the Action Association window for the selected Step	**Tools - Edit Action Association**	
	Edit the Step Comment	**Tools - Edit Step Comment**	
OR AND TRAN	Edit the Transition Condition	**Tools - Edit Transition Condition**	
	Open (zoom) the Body of a Step or Transition	**Tools - Zoom Body**	
	Open (zoom) a Step, Transition, or Macro Program window	**Tools - Zoom Into**	Ctrl I
	Undo zoom operation (i.e. close window)	**Tools - Zoom Back**	Ctrl B
VAR	List all available variables or operands	**Tools - List Operands...**	F2

Tab. C-4: Sequential Function Chart Editor Tool Icons

C.5 Project Navigator Window Tool Icons

Tool	Function	Menu command	Key(s)
	Display lower levels of the Project Navigator tree	Tools - Expand	Alt +
	Hide displayed levels of the Project Navigator	Tools - Collapse	Alt -
POU	Create a new Program Organisation Unit	Object - New - POU	Alt N
DUT	Create a new Data Unit Type	Object - New - DUT	
TSK	Create a new Task	Object - New - Task	Alt K
ACT	Create a new Action (this icon is only enabled when a POU written in Sequential Function Chart language is selected)	Object - New - Action	Shift Alt A

Tab. C-5: *Project Navigator Window Tool Icons*

D Temporary Files

D.1 Temporary Files Created when Projects are Compiled

A number of temporary files are created during compilation. These files are normally deleted when the program is finished; if you don't want them to be deleted you must start GX IEC Developer with the */debug* command line option. To do this, add this option to the GX IEC Developer command line in the Program Item Properties dialogue box in the Windows Program Manager:
E.g.: *c:\melsec\gx iec developer\sc.exe* /debug.

- **_tskmain.sic** Data of all tasks allocated to the MAIN memory range (stored in a special format)
  _tskmain.out Data of all tasks allocated to the MAIN memory range (special MELSEC code)
  _tskmain.bin Data of all tasks allocated to the MAIN memory range (the compiled code that is downloaded to the controller)
- **_tsksub.sic** Data of the Task *MELSEC_SUB*, allocated to the SUB memory range (stored in a special format)
  _tsksub.out Data of the Task *MELSEC_SUB*, allocated to the SUB memory range (special MELSEC code)
  _tsksub.bin Data of the Task *MELSEC_SUB*, allocated to the SUB memory range (the compiled code that is downloaded to the controller)
- **POUname.sic** Data of a specific Program Organisation Unit (stored in a special format)
  POU-Name.out Data of a specific Program Organisation Unit (special MELSEC code)
  POU-Name.bin Data of a specific Program Organisation Unit (the compiled code that is downloaded to the controller)
- **_end.sic** Data of the END instruction (stored in a special format)
  _end.out Data of the END instruction (special MELSEC code)
  _end.bin Data of the END instruction (the compiled code that is downloaded to the controller)
- **main.map** All the data of the program allocated to the MAIN memory range (compiled structural tree)
- **sub.map** All the data of the program allocated to the SUB memory range (compiled structural tree)
- **sfc.map** All the data of the program written in Sequential Function Chart Language (compiled structural tree)
- **sfccode.cod** Data of the Program Organisation Units that were written in Sequential Function Chart Language (compiled code)
- **mmicro.bin** Data of the Program Organisation Units that were written in Sequential Function Chart Language - MELSEC A-Series (compiled code)
- **build.out, tab.out, rebuild.out, comp.out**
  Compiling data

D.2 Temporary files Created During Downloads

- dwl_sym.asc
- upl_sym.pac
- upl_main.bin
- upl_sub.bin
- fnw_main.dis
- fnw_sub.dis
- fnw_imp.asc
- task_imp.asc

E Programming Instructions

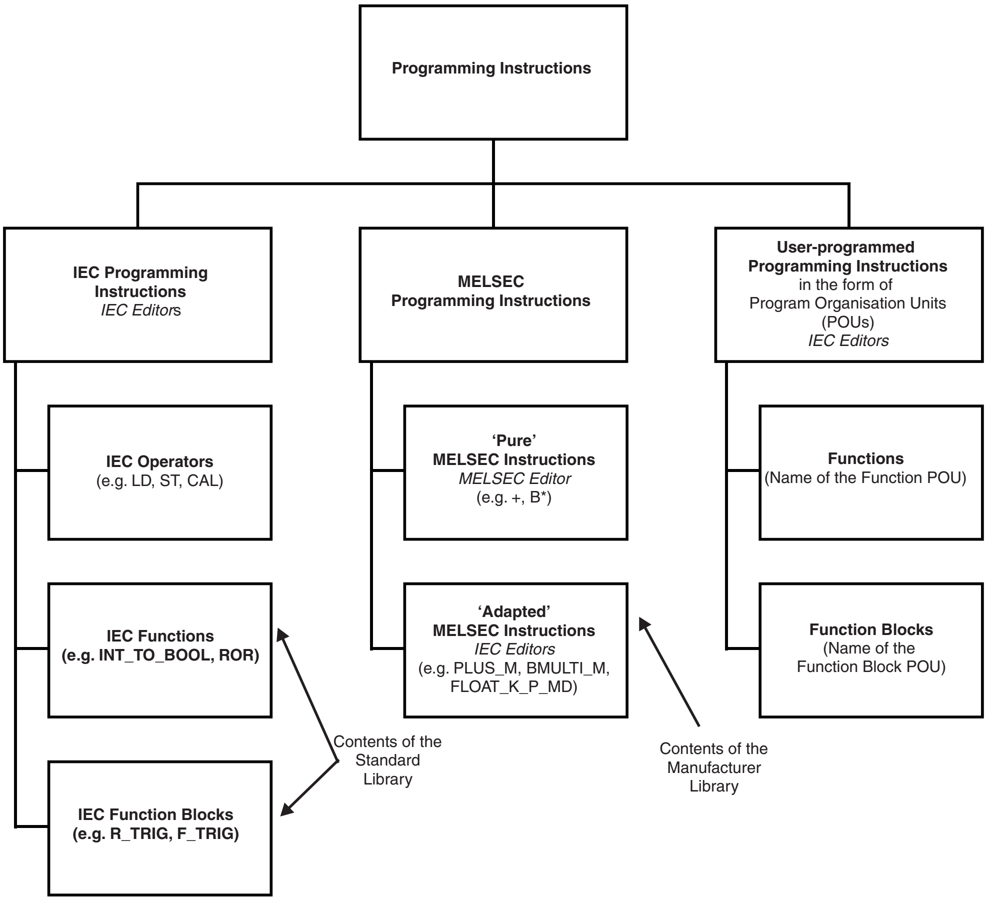

Programming Instructions
IEC Programming Instructions
IEC Editors
MELSEC Programming Instructions
User-programmed Programming Instructions in the form of Program Organisation Units (POUs)
IEC Editors
IEC Operators
(e.g. LD, ST, CAL)
IEC Functions
(e.g. INT_TO_BOOL, ROR)
IEC Function Blocks
(e.g. R_TRIG, F_TRIG)
'Pure' MELSEC Instructions
MELSEC Editor
(e.g. +, B*)
'Adapted' MELSEC Instructions
IEC Editors
(e.g. PLUS_M, BMULTI_M, FLOAT_K_P_MD)
Functions
(Name of the Function POU)
Function Blocks
(Name of the Function Block POU)
Contents of the Standard Library
Contents of the Manufacturer Library

Fig. E-1: *Programming instructions*

Standard Library

The standard library contains all the IEC functions and function blocks. Every available function has a FUN (function) POU, the headers of which you can open and view on the screen.

Manufacturer Library

The Manufacturer Library contains the 'adapted' MELSEC instructions, with a nomenclature adapted for use in IEC. Every function has a FUN (function) POU, the headers and bodies of which you can open and view on the screen. With the help of a special tool you can also add your own POUs to the Manufacturer Library. These POUs can be declared as functions (FUN), function blocks (FB) or programs (PRG).

MELSEC Instructions

The functionality of the ‘adapted’ and ‘pure’ MELSEC functions is identical. The only difference between the two versions is the way they are written and how they are used in the various editors.

The ‘pure’ MELSEC instructions use the familiar MELSEC nomenclature (e.g. AND=, +, FLOAT). These instructions can only be used in the MELSEC editor.

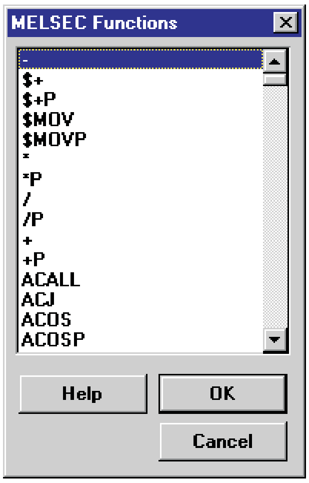
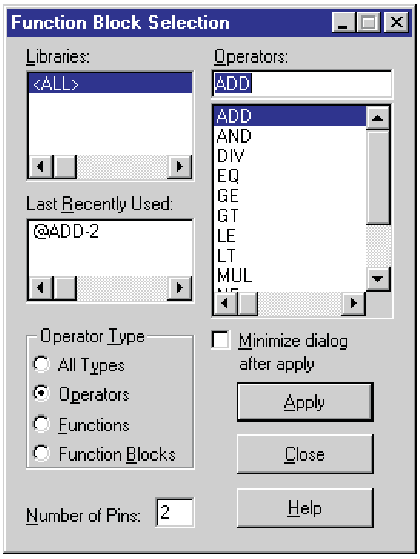


Fig. E-2:
This dialogue box is displayed when you are working in the MELSEC editor. The list contains the ‘pure’ MELSEC instructions.

The ‘adapted’ MELSEC instructions use a special nomenclature. Their names always end with _M for MELSEC or _MD for MELSEC Dedicated Instructions (e.g. PLUS_M, FLOAT_K_P_MD). You can use these instructions in the IEC editor.



Libraries: <ALL>

Operators: ADD

Last Recently Used: @ADD-2

Operator Type
○ All Types
● Operators
○ Functions
○ Function Blocks

□ Minimize dialog after apply

Number of Pins: 2

Fig. E-3:
This dialogue box is displayed when you are working in the IEC editor.
The list of the Manufacturer Library contains the ‘adapted’ MELSEC instructions.

The suffixes **_P_MD** and **_K_MD** indicate that the instruction under consideration is executed on a rising edge or with a constant.

MELSEC Instructions

MELSEC Editor	IEC Editor	Explanation
LEDA FLOAT LEDC (S) LEDC (D) LEDR	FLOAT_MD	Normal execution (16 Bit)
LEDB FLOAT LEDC (S) LEDC (D) LEDR	FLOAT_P_MD	Pulse-triggered execution (16 Bit)
LEDA FLOAT SUB (S) LEDC (D) LEDR	FLOAT_K_MD	Normal execution (16 Bit) Constant used in operand (S)
LEDB FLOAT SUB (S) LEDC (D) LEDR	FLOAT_K_P_MD	Pulse-triggered execution (16 Bit) Constant used in operand (S)
LEDA DFLOAT LEDC (S) LEDC (D) LEDR	DFLOAT_MD	Normal execution (32 Bit)
LEDB DFLOAT LEDC (S) LEDC (D) LEDR	DFLOAT_P_MD	Pulse-triggered execution (32 Bit)
LEDA DFLOAT DXNR (S) LEDC (D) LEDR	DFLOAT_K_MD	Normal execution (32 Bit) Constant used in operand (S)
LEDB DFLOAT DXNR (S) LEDC (D) LEDR	DFLOAT_K_P_MD	Pulse-triggered execution (32 Bit) Constant used in operand (S)

Tab. E-1: *MELSEC Instructions*

Meaning of extensions in the IEC editor

The following table lists the extensions of MELSEC instructions in the IEC editor. The nomenclature is as follows:

- **_instruction_**
- **_instruction**

Extension in IEC Editor	Meaning
_M	MELSEC instruction
_P_M	Pulse triggered execution of MELSEC instruction
_MD	Dedicated MELSEC Instruction
_P_MD	Pulse triggered execution of dedicated MELSEC Instruction
_K_MD	Constant used in dedicated MELSEC Instruction
_K_P_MD	Constant used in and pulse triggered execution of dedicated MELSEC Instruction
_E	Floating point number operands are used
_S, _STRING S_, STRING_	String operands are used
_J_M, _JP_M	Network number of HOST station is used in operand
_U_M, _UP_M	Head I/O address of network in HOST station is used in operand

Tab. E-2: *Meaning of extensions in the IEC editor*

EN input and ENO output

In the Standard Library, most of the programming instructions are displayed in two similar forms. The only difference between these programming instructions is the suffix _E. This suffix specifies functions and function blocks which have an EN input and an ENO output (➜ page 6-79).

The functions in the Manufacturer Library do not have this suffix although they all have an EN input and an ENO output.

E.1 IEC Programming Instructions

Instruction	Modi-fier	Description	No. of input variables	Data types	
				Input variables	Output variables
)		IEC Instruction List operator Right parenthesis The right parenthesis (close parenthesis) is used to evaluate a deferred operation. The beginning of the deferred operation is marked with a left parenthesis	e.g. AND(	MUL(	NE(. The left parenthesis is added to the operator itself as a modifier.
ABS ABS_E		Absolute value Writes the input variable data to the output variable with a positive sign.	1	INT, DINT	INT, DINT
ADD ADD_E	(	Addition	n (2 – 28)	(IN1) – (INn): INT, DINT, Constant	INT, DINT
ADD_TIME ADD_TIME_E		Addition	2	(IN1), (IN2): TIME	TIME
AND AND_E	N, (	Logical AND operation	n (2 – 28)	WORD, DWORD, BOOL, Constant	WORD, DWORD, BOOL
BCD_TO_DINT BCD_TO_DINT_E		Converts BCD format to binary format (DINT)	1	BOOL, WORD, DWORD	DINT
BCD_TO_INT BCD_TO_INT_E		Converts BCD format to binary format(INT)	1	BOOL, WORD, DWORD	INT
BOOL_TO_DINT BOOL_TO_DINT_E		Converts BOOL data type to DINT data type	1	BOOL	DINT
BOOL_TO_DWORD BOOL_TO_DWORD_E		Converts BOOL data type to DWORD data type	1	BOOL	DWORD
BOOL_TO_INT BOOL_TO_INT_E		Converts BOOL data type to INT data type	1	BOOL	INT
BOOL_TO_TIME BOOL_TO_TIME_E		Converts BOOL data type to TIME data type	1	BOOL	TIME
BOOL_TO_WORD BOOL_TO_WORD_E		Converts BOOL data type to WORD data type	1	BOOL	WORD
CAL	C, N	IEC Instruction List operator Calls a Function Block instance. In the Instruction List language Function Block instances are called with the CAL operator. The operator must be followed by the instance name, which in turn must be followed by the argument list in parentheses, containing the actual parameters to be passed to the formal parameters.	-	-	-
DINT_TO_BCD DINT_TO_BCD_E		Converts binary format (DINT) to BCD format	1	DINT, Constant	BOOL, WORD, DWORD
DINT_TO_BOOL DINT_TO_BOOL_E		Converts DINT data type to BOOL data type	1	DINT	BOOL
DINT_TO_DWORD DINT_TO_DWORD_E		Converts DINT data type to DWORD data type	1	DINT	DWORD
DINT_TO_INT DINT_TO_INT_E		Converts DINT data type to INT data type	1	DINT	INT
DINT_TO_REAL DINT_TO_REAL_E		Converts DINT data type to REAL data type	1	DINT	REAL

Tab. E-3: *IEC programming instructions*

Instruction	Modi-fier	Description	No. of input variables	Data types	
				Input variables	Output variables
DINT_TO_TIME DINT_TO_TIME_E		Converts DINT data type to TIME data type	1	DINT	TIME
DINT_TO_WORD DINT_TO_WORD_E		Converts DINT data type to WORD data type	1	DINT	WORD
DIV DIV_E	(	Division; (IN1) : (IN2)	2	(IN1), (IN2): INT, DINT, Constant	INT, DINT
DIV_TIME DIV_TIME_E		Division; (IN1) : (IN2)	2	(IN1), (IN2): TIME	TIME
DWORD_TO_BOOL DWORD_TO_BOOL_E		Converts DWORD data type to BOOL data type	1	DWORD	BOOL
DWORD_TO_DINT DWORD_TO_DINT_E		Converts DWORD data type to DINT data type	1	DWORD	DINT
DWORD_TO_INT DWORD_TO_INT_E		Converts DWORD data type to INT data type	1	DWORD	INT
DWORD_TO_TIME DWORD_TO_TIME_E		Converts DWORD data type to TIME data type	1	DWORD	TIME
DWORD_TO_WORD DWORD_TO_WORD_E		Converts DWORD data type to WORD data type	1	DWORD	WORD

EQ
EQ_E
	(	Equality (=);
		[(IN1 = IN2)] & [(IN2) = (IN3)] & ... & [(INn-1) = (INn)];
		If the comparison result is true, then 1 to output.
		If the comparison result is false, then 0 to output.
			n
			(2 – 28)
				(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL, Constant	BOOL

F_TRIG F_TRIG_E		Function Block; Falling edge detector; the current signal status on the output variable (Q) changes in response to a falling edge on the input variable (CLK).	1	(CLK): BOOL	Q: BOOL

GE
GE_E
	(	Greater than/equals sequential comparison (>=);
		[(IN1) >= (IN2)] & [(IN2) >= (IN3)] & ... & [(INn-1) >= (INn)];
		If the comparison result is true, then 1 to output.
		If the comparison result is false, then 0 to output.
			n
			(2 – 28)
				(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL, Constant	BOOL

GT GT_E	(	Greater than sequential comparison (>) [(IN1) > (IN2)] & [(IN2) > (IN3)] & ... & [(INn-1) > (INn)] If the comparison result is true, then 1 to output. If the comparison result is false, then 0 to output.	n (2 – 28)	(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL, Constant	BOOL
INT_TO_BCD INT_TO_BCD_E		Converts binary format (INT) to BCD format	1	INT	BOOL, WORD, DWORD
INT_TO_BOOL INT_TO_BOOL_E		Converts INT data type to BOOL data type	1	INT	BOOL
INT_TO_DINT INT_TO_DINT_E		Converts INT data type to DINT data type	1	INT	DINT
INT_TO_DWORD INT_TO_DWORD_E		Converts INT data type to DWORD data type	1	INT	DWORD
INT_TO_REAL INT_TO_REAL_E		Converts INT data type to REAL data type		INT	REAL
INT_TO_TIME INT_TO_TIME_E		Converts INT data type to TIME data type		INT	TIME

Tab. E-3: *IEC programming instructions*

Instruction	Modi-fier	Description	No. of input variables	Data types	
				Input variables	Output variables
INT_TO_WORD INT_TO_WORD_E		Converts INT data type to WORD data type	1	INT	WORD
JMP	C, N	IEC Instruction List operator Jump to network label Performs a jump from one program section to a label in another network. You must specify the network label name after the operator.	-	-	-
LD	N	IEC Instruction List operator Load Loads the value of the specified operand into the accumulator.	1	BOOL, WORD, DWORD, INT, DINT	-
LE LE_E	(	Less than/equals sequential comparison (<=) [(IN1) <= (IN2)] & [(IN2) <= (IN3)] & ... & [(INn-1) <= (Inn)] If the comparison result is true, then 1 to output. If the comparison result is false, then 0 to output.	n (2 – 28)	(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL, Constant	BOOL
LIMIT LIMIT_E		Value limitation If (IN) > (MX), then (MX) to output. If (IN) < (MN), then (MN) to output. If (MN) < (IN) > (MX), then (IN) to output	3	(MN), (IN), (MX): INT, DINT, WORD, DWORD, BOOL	INT, DINT, WORD, DWORD, BOOL
LT LT_E	(	Less than sequential comparison (<) [(IN1)< (IN2)] & [(IN2)< (IN3)] & ... & [(INn-1) < (INn)] If the comparison result is true, then 1 to output. If the comparison result is false, then 0 to output.	n (2 – 28)	(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL	BOOL
MAX MAX_E		Select maximum value The largest value in (IN1) – (INn) as written to the output.	n (2 - 28)	(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL	INT, DINT, WORD, DWORD, BOOL
MIN MIN_E		Select minimum value The smallest value in (IN1) – (INn) is written to the output.	n (2 - 28)	(IN1) – (INn): INT, DINT, WORD, DWORD, BOOL	INT, DINT, WORD, DWORD, BOOL
MOD MOD_E		Remainder (IN1) - [(IN1) : (IN2)] x (IN2)	2	(IN1), (IN2): INT, DINT	INT
MOVE MOVE_E		Move	1	INT, DINT	INT, DINT
MUL MUL_E	(	Multiplication	n (2 – 28)	(IN1) – (INn): INT, DINT	INT, DINT
MUL_TIME MUL_TIME_E		Multiplication	2	(IN1), (IN2): TIME	TIME
MUX MUX_E		Multiplexer Writes the data of the input variable between (INO) and (Inn) to the output. The input variable is specified in (K).	n (2 – 28)	(K): INT, DINT; (IN0) – (INn): INT, DINT, WORD, DWORD, BOOL	INT, DINT, WORD, DWORD, BOOL

Tab. E-3: *IEC programming instructions*

Instruction	Modi-fier	Description	No. of input variables	Data types	
				Input variables	Output variables
NE NE_E	(	Not equal (<>) [(IN1) <> (IN2)] If the comparison result is true, then 1 to output. If the comparison result is false, then 0 to output.	2	(IN1), (IN2): INT, DINT, WORD, DWORD, BOOL	BOOL
NOT NOT_E		Negation	1	WORD, DWORD, BOOL	WORD, DWORD, BOOL
OR OR_E	N, (	Logical OR operation	n (2 – 28)	WORD, DWORD, BOOL	WORD, DWORD, BOOL
R		IEC Instruction List operator Reset Resets the BOOL variable to 0. This operation is only performed if the value in the accumulator is 1.	1	BOOL	-
R_TRIG R_TRIG_E		Function Block Rising edge detector The status of the output variable (Q) is changed in accordance with the presence of a rising edge on the input variable (CLK).	1	(CLK): BOOL	(Q): BOOL
REAL_TO_DINT REAL_TO_DINT_E		Converts REAL data type to DINT data type		REAL	DINT
REAL_TO_INT REAL_TO_INT_E		Converts REAL data type to INT data type		REAL	INT
RET	C, N	IEC Instruction List operator Return from a Function Block Executes a return from a Function Block to the program in which the Function Block instance was called.	-	-	-
ROL ROL_E		Rotate left Rotates the data of the input variable (IN) to the left by (N) bits. No bits are lost in this process.	2	(IN): BOOL, WORD, DWORD; (N): Constant	WORD, DWORD, BOOL
ROR ROR_E		Rotate right Rotates the data of the input variable (IN) to the right by (N) bits. No bits are lost in this process.	2	(IN): BOOL, WORD, DWORD; (N): Constant	WORD, DWORD, BOOL
S		IEC Instruction List operator Set Sets the value of the BOOL variable to 1. This operation is only performed when the value in the accumulator is 1.	1	BOOL	-
SEL SEL_E		Binary selection When (G) = 0 then (IN0) to output. When (G) = 1 then (IN1) to output.	3	(G): BOOL; (IN0), (IN1): INT, DINT, WORD, DWORD, BOOL	INT, DINT, WORD, DWORD, BOOL
SHL SHL_E		Left shift Shifts the data of the input variable (IN) to the left by (N) bits. The upper (N) bits of (IN) are lost in this process.	2	(IN): BOOL, WORD, DWORD; (N): Constant	WORD, DWORD, BOOL
SHR SHR_E		Right shift Shifts the data of the input variable (IN) to the right by (N) bits. The lower (N) bits of (IN) are lost in this process.	2	(IN): BOOL, WORD, DWORD; (N): Constant	WORD, DWORD, BOOL

Tab. E-3: *IEC programming instructions*

Instruction	Modi-fier	Description	No. of input variables	Data types	
				Input variables	Output variables
ST	N	IEC Instruction List operator Store Stores the value of the specified operand to the accumulator.	1	BOOL, WORD, DWORD, INT, DINT	-
SUB SUB_E	(	Subtraction; (IN1) - (IN2)	2	(IN1), (IN2): INT, DINT	INT, DINT
SUB_TIME SUB_TIME_E		Subtraction; (IN1) - (IN2)	2	(IN1), (IN2): TIME	TIME
TIME_TO_BOOL TIME_TO_BOOL_E		Converts TIME data type to BOOL data type	1	TIME	BOOL
TIME_TO_DINT TIME_TO_DINT_E		Converts TIME data type to DINT data type	1	TIME	DINT
TIME_TO_DWORD TIME_TO_DWORD_E		Converts TIME data type to DWORD data type	1	TIME	DWORD
TIME_TO_INT TIME_TO_INT_E		Converts TIME data type to INT data type	1	TIME	INT
TIME_TO_WORD TIME_TO_WORD_E		Converts TIME data type to WORD data type	1	TIME	WORD
WORD_TO_BOOL WORD_TO_BOOL_E		Converts WORD data type to BOOL data type	1	WORD	BOOL
WORD_TO_DINT WORD_TO_DINT_E		Converts WORD data type to DINT data type	1	WORD	DINT
WORD_TO_DWORD WORD_TO_DWORD_E		Converts WORD data type to DWORD data type	1	WORD	DWORD
WORD_TO_INT WORD_TO_INT_E		Converts WORD data type to INT data type	1	WORD	INT
WORD_TO_TIME WORD_TO_TIME_E		Converts WORD data type to TIME data type	1	WORD	TIME
XOR XOR_E	N, (	Logical exclusive OR operation	n (2 – 28)	WORD, DWORD, BOOL	WORD, DWORD, BOOL

Tab. E-3: *IEC programming instructions*

E.2 IEC – MELSEC

The following tables list the most common IEC programming instructions with their corresponding MELSEC programming instructions. For a complete overview of instructions refer to the programming manuals for the A/Q and FX series.

E.2.1 Addition

Programming instruction		**Description**	**No. of input variables**	**Series**	**Notes**
IEC ***IEC Editor***					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. The number of input variables is fixed using the MELSEC functions and variable using the IEC functions (min. 2, max. 28). All the MELSEC functions have an EN input and an ENO output.
ADD ADD_E Modifier: (		Addition of numerical data/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
ADD_M DADD_M	ADD DADD	Addition of numerical data (16/32 bit)	3	Q, QnA, FX	
PLUS_M DPLUS_M	+ D+	Addition of numerical data (16/32 bit)	2	Q, QnA, A	
PLUS_3_M DPLUS_3_M		Addition of numerical data (16/32 bit)	3	Q, QnA, A	
ADD_MD	ADD (with LED* !)①	Addition of floating-point numbers	2	Q, QnA, AnA	For floating-point numbers only
BPLUS_M DBPLUS_M	B+ DB+	BCD addition (16/32 bit)	2	Q, QnA, A	For BCD data only
BPLUS_3_M DBPLUS_3_M		BCD addition (16/32 bit)	3	Q, QnA, A	
SADD_MD	SADD (with LED* !)①	Concatenation of two ASCII strings	2	Q, QnA, AnA	**WARNING!** **No addition!**

Tab. E-4: *Addition*

IEC		**MELSEC in the IEC Editor**
ADD_E with 3 input variables of data type INT	→	ADD_M for the FX series PLUS_3_M for the A series
ADD_E with 2 input variables of data type DINT	→	DPLUS_M for the A series
Advantage: Enables addition of up to 28 input variables.		

① Dedicated instructions with LEDA/B

E.2.2 Subtraction

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. All the MELSEC functions have an EN input and an ENO output.
SUB SUB_E Modifier: (		Subtraction of numerical data/with EN input and ENO output	2	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
SUB_3_M DSUB_3_M	SUB DSUB	Subtraction of numerical data (16/32 bit)	3	FX	
MINUS_M DMINUS_M	- D-	Subtraction of numerical data (16/32 bit)	2	Q, QnA, A	
MINUS_3_M DMINUS_3_M		Subtraction of numerical data (16/32 bit)	3	Q, QnA, A	
SUB_MD	SUB (with LED* !)①	Subtraction of floating-point numbers	2	Q, QnA, AnA	For floating-point numbers only
BMINUS_M DBMINUS_M	B- DB-	BCD subtraction (16/32 bit)	2	Q, QnA, A	For BCD data only
BMINUS_3_M DBMINUS_3_M		BCD subtraction (16/32 bit)	3	Q, QnA, A	
②	SUB	Execute a microcomputer program	1	Q, QnA, A	**WARNING!** **No subtraction!**

Tab. E-5: *Subtraction*

IEC		**MELSEC in the IEC Editor**
SUB_E with input variables of data type INT	→	MINUS_M for the A series
SUB_E with input variables of data type DINT	→	DMINUS_M for the A series

① Dedicated instructions with LEDA/B

② These instructions are not supported by the IEC editor.

E.2.3 Multiplication

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. The number of input variables is fixed using the MELSEC functions and variable using the IEC functions (min. 2, max. 28). All the MELSEC functions have an EN input and an ENO output.
MUL MUL_E Modifier: (		Multiplication of numerical data/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
MUL_M DMUL_M	MUL DMUL	Multiplication of numerical data (16/32 bit)	3	FX	
MULTI_M DMULTI_M	* D*	Multiplication of numerical data (16/32 bit)	2	A	
MULTI_3_M DMULTI_3_M		Multiplication of numerical data (16/32 bit)	3	Q, QnA, A	
MUL_MD	MUL (with LED* !)[1]	Multiplication of floating-point numbers	2	Q, QnA, AnA	For floating-point numbers only
BMULTI_M DBMULTI_M	B* DB*	BCD multiplication (16/32 bit)	2	Q, QnA, A	For BCD data only
BMULTI_3_M DBMULTI_3_M		BCD multiplication (16/32 bit)	3	A	

Tab. E-6: *Multiplication*

IEC		**MELSEC in the IEC Editor**
MUL_E with 3 input variables of data type INT	→	MUL_M for the FX series MULTI_3_M for the A series
MUL_E with 2 input variables of data type DINT	→	DMULTI_M for the A series

Advantage: Enables multiplication of up to 28 input variables.

[1] Dedicated instructions with LEDA/B

E.2.4 Division

Programming instruction		**Description**	**No. of input variables**	**Series**	**Notes**
IEC ***IEC Editor***					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. All the MELSEC functions have an EN input and an ENO output.
DIV DIV_E Modifier: (		Division of numerical data/with EN input and ENO output	2	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
DIV_M DDIV_M	DIV DDIV	Division of numerical data (16/32 bit)	3	FX	
DIVID_M DDIVID_M	/ D/	Division of numerical data (16/32 bit)	2	A	
DIVID_3_M DDIVID_3_M		Division of numerical data (16/32 bit)	3	Q, QnA, A	
DIV_MD	DIV (with LED* !)[1]	Division of floating-point numbers	2	Q, QnA, AnA	For floating-point numbers only
BDIVID_M DBDIVID_M	B/ DB/	BCD division (16/32 bit)	2	Q, QnA, A	For BCD data only
BDIVID_3_M DBDIVID_3_M		BCD division (16/32 bit)	3	A	

Tab. E-7: *Division*

IEC		**MELSEC in the IEC Editor**
DIV_E with input variables of data type INT	→	DIVID_M for the A series
DIV_E with input variables of data type DINT	→	DDIVID_M for the A series

[1] Dedicated instructions with LEDA/B

E.2.5 Move

Programming instruction		Description	No. of input variables	Series	Notes
IEC IEC Editor					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. All the MELSEC functions have an EN input and an ENO output.
MOVE MOVE_E		Moves numerical data/ with EN input and ENO output	1	Q, QnA, A, FX, FX0	
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
MOV_M DMOV_M	MOV DMOV	Moves numerical data (16/32 bit)	1	Q, QnA, A, FX, FX0	
BMOV_M	BMOV	Block data move	2	Q, QnA, A, FX, FX0	Special move operations
BMOVR_MD	BMOVR (with LED* !)[1]	Moves the data from one block to another block with the extension file registers	2	AnA	
FMOV_M	FMOV	Fill move	2	Q, QnA, A, FX, FX0	
SMOV_M	SMOV	Shift move	4	FX	
SMOV_MD	SMOV (with LED* !)[1]	Character string move	1	Q, QnA, AnA	

Tab. E-8: *Move*

IEC		MELSEC in the IEC Editor
MOVE_E with 1 input variable of data type INT	→	MOV_M for the A and FX series
MOVE_E with 1 input variable of data type DINT	→	DMOV_M for the A and FX series

[1] Dedicated instructions with LEDA/B

E.2.6 BCD – BIN conversion

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. All the MELSEC functions have an EN input and an ENO output.
INT_TO_BCD INT_TO_BCD_E		Conversion of binary format (INT) to BCD format/with EN input and ENO output	1	Q, QnA, A, FX, FX0	
DINT_TO_BCD DINT_TO_BCD_E		Conversion of binary format (DINT) to BCD format/with EN input and ENO output	1	Q, QnA, A, FX, FX0	
BCD_TO_INT BCD_TO_INT_E		Conversion of BCD format to binary format (INT)/with EN input and ENO output	1	Q, QnA, A, FX, FX0	
BCD_TO_DINT BCD_TO_DINT_E		Conversion of BCD format to binary format (DINT)/with EN input and ENO output	1	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
BCD_M DBCD_M	BCD DBCD	Conversion of binary format to BCD format (16/32 bit)	1	Q, QnA, A, FX, FX0	
BIN_M DBIN_M	BIN DBIN	Conversion of BCD format to binary format (16/32 bit)	1	Q, QnA, A, FX, FX0	
BINDA_MD DBINDA_MD	BINDA DBINDA (with LED* !)①	Conversion of binary data (16/32 bit) to an ASCII string (decimal)	1	Q, QnA, AnA	For ASCII strings
BINHA_MD DBINHA_MD	BINHA DBINHA (with LED* !)①	Conversion of binary data (16/32 bit) to an ASCII string (hexadecimal)	1	Q, QnA, AnA	
DABIN_MD DDABIN_MD	DABIN DDABIN (with LED* !)①	Conversion of an ASCII string (decimal) to binary data (16/32 bit)	1	Q, QnA, AnA	
HABIN_MD DHABIN_MD	HABIN DHABIN (with LED* !)①	Conversion of an ASCII string (hexadecimal) to binary data (16/32 bit)	1	Q, QnA, AnA	
BCDDA_MD DBCDDA_MD	BCDDA DBCDDA (with LED* !)①	Conversion of BCD data (16/32 bit) to an ASCII string (decimal)	1	Q, QnA, AnA	
DABCD_MD DDABCD_MD	DABCD DDABCD (with LED* !)①	Conversion of an ASCII string (decimal) to BCD coded data (16/32 bit)	1	Q, QnA, AnA	

Tab. E-9: *BCD – BIN conversion*

IEC		**MELSEC in the IEC Editor**
INT_TO_BCD_E	→	BCD_M for the A and FX series
BCD_TO_INT_E	→	BIN_M for the A and FX series

① Dedicated instructions with LEDA/B

E.2.7 Logic Operations

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					The IEC functions with input variables of the INT (DINT) data type correspond to the MELSEC functions with 16 (32) bit data. The number of input variables is fixed using the MELSEC functions and variable using the IEC functions (min. 2, max. 28). All the MELSEC functions have an EN input and an ENO output.
AND AND_E Modifier: N, C		Logical AND operation/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
OR OR_E Modifier: N, C		Logical OR operation/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
XOR XOR_E Modifier: N, C		Logical exclusive OR operation/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
WAND_M DAND_M	WAND DAND	Logical AND operation (16/32 bit)	2	Q, QnA, A	
WAND_3_M DAND_3_M		Logical AND operation (16/32 bit)	3	Q, QnA, A, FX, FX0	
WOR_M DOR_M	WOR DOR	Logical OR operation (16/32 bit)	2	Q, QnA, A	
WOR_3_M DOR_3_M		Logical OR operation (16/32 bit)	3	A, FX, FX0	
WXOR_M DXOR_M	WXOR DXOR	Logical exclusive OR operation (16/32 bit)	2	Q, QnA, A	
WXOR_3_M DXOR_3_M		Logical exclusive OR operation (16/32 bit)	3	Q, QnA, A, FX, FX0	
WXNR_M DXNR_M	WXNR	Logical exclusive NOR operation (16/32 bit)	2	Q, QnA, A	Exclusive NOR is not available as an IEC function.
WXNR_3_M DXNR_3_M		Logical exclusive NOR operation (16/32 bit)	3	Q, QnA, A	
②	AND	AND operation with poll for signal status 1	1	Q, QnA, A, FX, FX0	Input contacts
②	ANI	AND operation with poll for signal status 0	1	A, FX, FX0	
②	OR	OR operation with poll for signal status 0	1	Q, QnA, A, FX, FX0	
②	ORI	OR operation with poll for signal status 0	1	A, FX, FX0	
②	AND** ANDD**	AND comparison operations (16/32 bit)	2	Q, QnA, A	Comparison (→ Comparisons table)
②	OR** ORD**	OR comparison operations (16/32 bit)	2	Q, QnA, A	

Tab. E-10: *Logic Operations*

IEC		**MELSEC in the IEC Editor**
AND_E with 3 input variables of data type WORD	→	WAND_3_M for the A and FX series
AND_E with 2 input variables of data type DWORD	→	DAND_M for the A and FX series

Advantage: Enables operation performance of up to 28 input variables.

② These instructions are not supported by the IEC editor.

E.2.8 Start of an Operation and Output of a Result

Programming instruction		**Description**	**No. of input variables**	**Series**	**Notes**
IEC **IEC Editor**					The IEC function LD does not function as an input condition. This function loads a value into the accumulator. The MELSEC LD and LDI functions are used as the input condition for the subsequent operations. The IEC functions LD and ST are operators that are only used in the IEC Instruction List language (→ Chapter 6).
LD Modifier: N		IEC Instruction List operator Load specified value/variables into the accumulator	1	Q, QnA, A, FX, FX0	
ST Modifier: N		IEC Instruction List operator Load specified value/variables into the accumulator	1	Q, QnA, A, FX, FX0	
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
②	LD	Start of a logic operation with poll for signal status 1 (input contact)	1	Q, QnA, A, FX, FX0	
②	LDI	Start of a logic operation with poll for signal status 0 (input contact)	1	Q, QnA, A, FX, FX0	
②	LD** LDD**	Start of comparison operations (16/32 bit)	1	Q, QnA, A	
②	OUT	Output, Assignment of a logic operation result (output coil)	1	Q, QnA, A, FX, FX0	

Tab. E-11: *Start of an Operation and Output of a Result*

② These instructions are not supported by the IEC editor.

E.2.9 Comparisons

Programming instruction		Description	No. of input variables	Series	Notes
IEC *IEC-Editor*					The number of input variables is fixed using the MELSEC functions and variable using the IEC functions (min. 2, max. 28).
GT GT_E		Decreasing sequence >/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
GE GT_E		Monotonic sequence >=/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
EQ EQ_E		Equality =/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
LE LE_E		Monotonic sequence <=/with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
LT LT_E		Increasing sequence </with EN input and ENO output	n (2 – 28)	Q, QnA, A, FX, FX0	
NE NE_E		Inequality <>/with EN input and ENO output	2	Q, QnA, A, FX, FX0	
MELSEC *IEC-Editor*	*MELSEC* *MELSEC-Editor*				
②	LD> LDD>	Beginning of a comparison, Sequential comparison > (16/32 bit)	2	Q, QnA, A	
②	LD>= LDD>=	Beginning of a comparison, Sequential comparison >= (16/32 bit)	2	Q, QnA, A	
②	LD= LDD=	Beginning of a comparison, Equality = (16/32 bit)	2	Q, QnA, A	
②	LD<= LDD<=	Beginning of a comparison, Sequential comparison <= (16/32 bit)	2	Q, QnA, A	
②	LD< LDD<	Beginning of a comparison, Sequential comparison < (16/32 bit)	2	Q, QnA, A	
②	LD<> LDD<>	Beginning of a comparison, Inequality <> (16/32 bit)	2	Q, QnA, A	
②	AND> ANDD>	AND comparison, Sequential comparison > (16/32 bit)	2	Q, QnA, A	
②	AND>= ANDD>=	AND comparison, Sequential comparison >= (16/32 bit)	2	Q, QnA, A	
②	AND= ANDD=	AND comparison, Equality = (16/32 bit)	2	Q, QnA, A	
②	AND<= ANDD<=	AND comparison, Sequential comparison <= (16/32 bit)	2	Q, QnA, A	
②	AND< ANDD<	AND comparison, Sequential comparison < (16/32 bit)	2	Q, QnA, A	
②	AND<> ANDD<>	AND comparison, Inequality <> (16/32 bit)	2	Q, QnA, A	
②	OR> ORD>	OR comparison, Sequential comparison > (16/32 bit)	2	Q, QnA, A	
②	OR>= ORD>=	OR comparison, Sequential comparison >= (16/32 bit)	2	Q, QnA, A	
②	OR= ORD=	OR comparison, Equality = (16/32 bit)	2	Q, QnA, A	
②	OR<= ORD<=	OR comparison, Sequential comparison <= (16/32 bit)	2	Q, QnA, A	
②	OR< ORD<	OR comparison, Sequential comparison < (16/32 bit)	2	Q, QnA, A	
②	OR<> ORD<>	OR comparison, Inequality <> (16/32 bit)	2	Q, QnA, A	

Tab. E-12: *Start of an Operation and Output of a Result*

IEC		MELSEC in the IEC Editor
GT_E with 2 input variables of data type INT	→	LD_GT_M for the A series
GT_E with 2 input variables of data type DINT	→	LDD_GT_M for the A series
Advantage: Enables comparison of up to 28 input variables.		

② These instructions are not supported by the IEC editor.

E.2.10 Shift

Programming instruction		Description	No. of input variables	Series	Notes
IEC IEC Editor					The upper bits are lost in the IEC functions. With the MELSEC functions the upper bits are also lost
SHL SHL_E		Left shift Shifts the data of the input variable (IN1) to the left by (N) bits. The upper (N) bits of (IN1) are lost in this process/with EN input and ENO output	2	Q, QnA, A, FX, FX0	
SHR SHR_E		Right shift Shifts the data of the input variable (IN1) to the right by (N) bits. The upper (N) bits of (IN1) are lost in this process /with EN input and ENO output	2	Q, QnA, A, FX, FX0	
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
SFL_M	SFL	Shift a 16-bit data word to the left by (n) bits (Carry Flag)	2	Q, QnA, A	
SFR_M	SFR	Shift a 16-bit data word to the right by (n) bits (Carry Flag)	2	Q, QnA, A	
SFTL_M	SFTL	Shift bit devices (n) bits to the left	4	FX	
SFTR_M	SFTR	Shift bit devices (n) bits to the right	4	FX	
WSFL_M	WSFL	Shift word devices (n) places to the left	4	FX	
WSFR_M	WSFR	Shift word devices (n) places to the right	4	FX	
SFT_M	SFT	Shift bit devices by 1 bit	1	Q, QnA, A	
BSFL_M	BSFL	Shift bit devices one place to the left (Carry Flag)	2	Q, QnA, A	
BSFR_M	BSFR	Shift bit devices one place to the right (Carry Flag)	2	Q, QnA, A	
DSFL_M	DSFL	Shift word devices one place to the left	2	Q, QnA, A	
DSFR_M	DSFR	Shift word devices one place to the right	2	Q, QnA, A	

Tab. E-13: *Shift*

IEC		**MELSEC in the IEC Editor**
SHL_E with input variables of data type WORD	→	SFL_M for the A series (Difference: Carry Flag)
SHR_E with input variables of data type WORD	→	SFR_M for the A series (Difference: Carry Flag)

E.2.11 Rotation

Programming instruction		**Description**	**No. of input variables**	**Series**	**Notes**
IEC IEC Editor					None of the bits are lost.
ROL ROL_E		Rotate left The data in (IN) are rotated left by (N) bits; no bits are lost/with EN input and ENO output	2	Q, QnA, A, FX, FX0	
ROR ROR_E		Rotate right The data in (IN) are rotated right by (N) bits; no bits are lost /with EN input and ENO output	2	Q, QnA, A, FX, FX0	
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
ROL_M DROL_M	ROL DROL	Rotate left in A0 and A1 (16/32 bit)	1	Q, QnA, A	
ROR_M DROR_M	ROR DROR	Rotate right in A0 and A1 (16/32 bit)	1	Q, QnA, A	
ROL_2_M DROL_2_M	ROL DROL	Rotate left (16/32 bit)	2	FX	Special rotation
ROR_2_M DROR_2_M	ROR DROR	Rotate right (16/32 bit)	2	FX	
RCL_2_M DRCL_2_M	RCL DRCL	Rotate left with Carry Flag (16/32 bit)	2	FX	
RCR_2_M DRCR_2_M	RCR DRCR	Rotate right with Carry Flag (16/32 bit)	2	FX	
RCL_M DRCL_M	RCL DRCL	Rotate left in A0 and A1 with Carry Flag (16/32 bit)	1	Q, QnA, A	
RCR_M DRCR_M	RCR DRCR	Rotate right in A0 and A1 with Carry Flag (16/32 bit)	1	Q, QnA, A	

Tab. E-14: *Rotation*

IEC		**MELSEC in the IEC Editor**
ROL_E with input variables of data type WORD	→	ROL_M for the A series (Difference: Carry Flag, Fixed device A0)
ROR_E with input variables of data type DWORD	→	DROR_M for the A series (Difference: Carry Flag, Fixed devices A0 and A1)

E.2.12 Set and Reset Operations

Programming instruction		**Description**	**No. of input variables**	**Series**	**Notes**
IEC ***IEC Editor***					In the IEC functions the contents of the accumulator have an effect on the execution of the functions (→ Chapter 6)
S		IEC Instruction List operator Set Sets the value of the BOOL variable to 1. This operation is only executed when the value in the accumulator is 1.	1	Q, QnA, A, FX, FX0	
R		IEC Instruction List operator Reset Sets the value of the BOOL variable to 0. This operation is only executed when the value in the accumulator is 1.	1	Q, QnA, A, FX, FX0	
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
SET_M	SET	Device set operation	1	Q, QnA, A, FX, FX0	
RST_M	RST	Device reset operation	1	Q, QnA, A, FX, FX0	
BSET_M	BSET	Used to set individual bits in word devices	2	Q, QnA, A	Special set and reset operations
BRST_M	BRST	Used to reset individual bits in word devices	2	Q, QnA, A	
DSET_MD	DSET (with LED!)[1]	Direct set	1	Q, QnA, AnA	
DRST_MD	DRST (with LED!)[1]	Direct reset	1	Q, QnA, AnA	
ZRST_M	ZRST	Reset device zones	2	FX	

Tab. E-15: *Set and Reset Operations*

[1] Dedicated instructions with LEDA/B

E.2.13 Rising/Falling Edge

Programming instruction		Description	No. of input variables	Series	Notes
IEC IEC Editor					R_TRIG/E_R_TRIG and F_TRIG/E_F_TRIG are Function Blocks. This means that you must create instances of them before you can use them in your program (→ Chapter 6).
R_TRIG R_TRIG_E		Function block Rising edge detect The current signal status of the output variable (Q) changes on the rising edge on the input variable (CLK)/with EN input and ENO output	1	Q, QnA, A, FX	
F_TRIG F_TRIG_E		Function block Falling edge detect The current signal status of the output variable (Q) changes on the falling edge on the input variable (CLK)/with EN input and ENO output	1	Q, QnA, A, FX	
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
PLS_M	PLS	Generates a single pulse on rising edge; the current signal status changes on a rising edge.	1	Q, QnA, A, FX, FX0	
PLF_M	PLF	Generates a single pulse on a falling edge; the current signal status changes on a falling edge.	1	Q, QnA, A, FX, FX0	

Tab. E-16: *Rising/Falling Edge*

IEC		**MELSEC in the IEC Editor**
R_TRIG_E	→	PLS_M
F_TRIG_E	→	PLF_M

E.2.14 Negation

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					
NOT NOT_E		Negation/with EN input and ENO output	1	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
CML_M DCML_M	CML DCML	Data negation (16/32 bit) Negation of the individual bits	2	Q, QnA, A, FX, FX0	
NEG_M DNEG_M	NEG DNEG	Data negation (16/32 bit) Formation of the two's complement	0	Q, QnA, A, FX, FX0	

Tab. E-17: *Negation*

IEC		**MELSEC in the IEC Editor**
NOT_E with an input variable of data type WORD	→	CML_M
NOT_E with an input variable of data type DWORD	→	DCML_M

E.2.15 Jumps/Program Branching

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					The IEC JMP function is normally executed without any input condition. You must use the modifier to program a conditional jump. Network labels are used as jump destinations. (→ Chapter 6).
JMP Modifiers: N, C		IEC Instruction List operator. Jump to a network label. You can execute a jump to another network within a program section. Specify the name (label) of the network after the operator.	0	Q, QnA, A, FX, FX0	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
②	JMP	Unconditional jump within a program	1	Q, QnA, A	
②	CJ	Conditional jump within a program	1	Q, QnA, A, FX	
②	SCJ	Conditional jump within a program, executed in the next program cycle	1	Q, QnA, A	

Tab. E-18: *Jumps/Program Branching*

Meaning of the modifiers for the jump function JMP:

JMP (Jump):
The jump is always executed unconditionally.

JMPC (Jump Conditional):
The jump is only executed when the value in the accumulator is 1.

JMPCN (Jump Conditional Not):
The jump is only executed when the value in the accumulator is 0.

② These instructions are not supported by the IEC editor.

E.2.16 Calling Subprograms and Function Blocks

Programming instruction		Description	No. of input variables	Series	Notes
IEC ***IEC Editor***					The IEC functions are used for jumps to and returns from Function Blocks. The MELSEC functions are used for jumps to and returns from the normal subroutines/subprograms in the PLC sequence program.
CAL Modifiers: C, N		IEC Instruction List operator Call to a Function Block instance. In Instruction List the instances of Function Blocks can only be called via the CAL operator. Specify the instance name after the operator, then pass the actual parameters to the formal parameters.	0	A, FX	
RET Modifiers: C, N		IEC Instruction List operator Return from a Function Block. From a Function Block, you can return to the program from which the Function Block instance was called.	0	Q, QnA, A, FX	
MELSEC ***IEC Editor***	***MELSEC*** ***MELSEC Editor***				
②	CALL	Call a subprogram	1	Q, QnA, A, FX	
②	SRET	End/return from a subprogram	0	FX	
②	RET		0	Q, QnA, A	
		Terminate Step Ladder	0	FX	

Tab. E-19: *Calling Subprograms and Function Blocks*

② These instructions are not supported by the IEC editor.

E.2.17 Exclusive Q Series Instructions

Programming instruction		Description	No. of input variables	Series	Notes
MELSEC IEC Editor	***MELSEC MELSEC Editor***				
UNIRD_M UNIRDP_M	UNIRD UNIRDP	Read the module information stored in the area starting from the designated I/O No. and store it in the area starting from the designated device.	3	Q	
TRACE_M	TRACE	Store trace data set at a peripheral device to trace file in IC memory card by the designated number when SM800, SM801, and SM802 turn ON.	0	Q	
TRACER_M	TRACER	Reset the data set by the TRACE instruction.	0	Q	
SP.FWRITE_M	SP.FWRITE	Write data to a designated file.	6	Q	
SP.FREAD_M	SP.FREAD	Read data from a designated file.	6	Q	
PLOADP_M	PLOADP	Transfer the program stored in a memory card or standard memory card (other than drive 0) to drive 0 and place the program in stand-by status.	2	Q	
PUNLOADP_M	PUNLOADP	Delete the stand-by program stored in standard memory (drive 0).	2	Q	
PSWAPP_M	PSWAPP	Delete stand-by program stored in designated standard memory (drive 0). Then the program stored in a memory card or standard memory (other than drive 0) is transferred to drive 0 and placed in stand-by status.	3	Q	
RBMOV_M RBMOVP_M	RBMOV RBMOVP	Block data move	3	Q	

Tab. E-20: *Exclusive Q Series Instructions*

F Menu structure

The following pages show the menu structure of GX IEC Developer. The menu structure and the available commands are context-sensitive, changing depending on which CPU series you are using and what you are currently doing in the program.

This is marked by the following footnotes:

① no Remote I/O

② no FX0 / FX0S

③ no A- and FX-series

④ no FX-series

⑤ no Q00(J)/Q01

⑥ only FX-series

⑦ only Q00(J)/Q01

⑧ only A- and FX-series

⑨ only FX3U

⑩ only Q- and QnA-series

⑪ only Q-series

⑫ only Q-series except Q25SS

⑬ only QnPRH

⑭ only with MXFastLinx PAK code

⑮ only available in the IEC Instruction List (Body IL)

⑯ only available in Ladder Diagram

⑰ only available in Global Variable List

⑱ only available in SFC

⑲ only available in Ladder Diagram and Function Block Diagram

Project

all windows

New…	
Open...	
Close	
Save	
Save as…	
Other ▶	Rename…
	Copy…
	Import…
	Export…
	Export to Eprom ⑧
	Update Libraries ①
	Verify
Change PLC Type… ①	
Build ①	
Rebuild All ①	
Transfer ▶	Upload from PLC…
	Download to PLC…
	Verify
	Online verify ⑬
	Download Symbolic Information ① ⑩
	Copy Program Memory to Flash ROM ① ⑪
Online Program Change ① ②	
Browse… ① ⑫	
Make Cross Reference ①	
FastLinx ▶	FastLinx New ① ⑭
	FastLinx Open ① ⑭
MXChange Database Actions ▶	New ① ⑭
	Open ① ⑭
	Save as ① ⑭
	Delete ① ⑭
	Compact ① ⑭
	Change Password ① ⑭
	Close ① ⑭
MXChange Project Actions ▶	Load from MXChange ① ⑭
	Show Changes ① ⑭
	Global Adjust ① ⑭
	Delete ① ⑭
Printer Setup…	
Print Options…	
Print Preview…	
Print…	
Change Security Level…	
Change Passwords…	
Recent Files	
Quit	

Object
in Navigator

New ▶	POU… ① Task… ① Action… ① Data Unit Type… ①
Open ①	
Close ①	
Save ①	
Rename… ①	
Comment…	
Information…	
Check	
Print…	
Print Preview…	
Import…	
Export…	

Object
in Body IL, MELSEC IL, LD, FBD, ST, SFC

Close
Save
Comment…
Information…
Check
Print…
Print Preview…
Open Header

Object
in Declaration editors

Close
Save
Comment…
Information…
Check
Print…
Print Preview…
Open Body

Edit

in Navigator

Cut ①	
Copy	
Paste	
Delete ①	
Find… ①	
Replace… ①	
Update Library ①	
User Library ▶	Install/Create… ①
	Deinstall ①
	Modify… ①
	Open… ①
	Close ①
	Save ①
	Delete ①
	Change Password… ①

Edit

in Body IL, MELSEC IL

Undo	
Redo	
Cut	
Copy	
Paste	
Delete	
Find…	
Replace…	
New Network ▶	Top
	Before
	After
	Bottom
	MELSEC Before ⑮

Edit

in Body LD, FBD

Undo	
Redo	
Cut	
Copy	
Paste	
Delete	
Find…	
Replace…	
New Network ▶	Top
	Before
	After
	Bottom
Increment Pins	
Decrement Pins	
Signal Configuration…	
Toggle Signal Config…	

Edit

in Body ST

Undo
Redo
Cut
Copy
Paste
Delete
Find…
Replace…
Toggle Bookmark
Next Bookmark
Previous Bookmark
Delete all Bookmarks

Edit

in Body SFC

Undo	
Redo	
Cut	
Copy	
Paste	
Delete	
Find…	
Replace…	
Mark	
Macro	
Expand Macro	
Modify ▶	Name
	Initial Step
	Final Step
	Macro Step
	Entry Step
	Exit Step
Insert ▶	Step/Transition Unit
	Step
	Transition
	Left Divergence
	Right Divergence
	Left Convergence
	Right Convergence
	Label
	Jump

Edit

in Declaration editors

- Undo
- Redo
- Cut
- Copy
- Paste
- Delete
- Find…
- Replace…
- New Declaration ▶
  - Top
  - Before
  - After
  - Bottom

Edit

in Entry Data Monitor (EDM)

- Insert Objects… ①
- Next Object
- Inert Forced Inputs ④ ⑤
- Insert Set Inputs ⑥ ⑦
- Insert Set Outputs
- Clear Device File
- Cut
- Copy
- Paste
- Insert Row
- Delete
- Delete All
- Close

Edit

in Show Generated Code

- Copy All

Tools

in Navigator

Expand
Collapse

Tools

in Body IL, MELSEC IL

Edit Network	
End Edit	
Network List…	
Show MELSEC Code Of Networks	
Import GX Developer Network	
Import MEDOC Network	
List Operators/Operands	
Create Template	
Tracking ▶	Start Tracking
	Next
	Previous
	First
	Last
New Variable…	
Zoom ▶	Header
	Body

Tools

in Body ST

Show MELSEC Code Of Networks	
Show MELSEC Code Of Selection	
List Operands	
List Operators	
Create Template	
Tracking ▶	Start Tracking
	Next
	Previous
	First
	Last
New Variable…	
Zoom ▶	Header
	Body

Tools

in Body LD

Network Elements ▶	Contact ⑯
	Contact Negation
	Coil ⑯
	Jump
	Return
	Function Block
	L-Connection Contact
	L-Connection Contact Negation
	Input Variable
	Output Variable
	Horizontal Line Segment
	Vertical Line Segment
	Comment
	Lable of the Network
	Delete Line
Open Row	
Open Column	
Define Graphic Macros…	
Custom ▶	Custom 1
	Custom 2
	Custom 3
	Custom 4
	Custom 5
More Custom…	
Select Mode	
Interconnect Mode	
Auto Connect	
Guided Mode ▶	Guided Editing
	Overwrite Mode
	Insert Mode
	Line Mode
	Auto Comment
Recalculate Line	
Network List…	
Show MELSEC Code Of Networks	
List Operators/Operands	
Tracking ▶	Start Tracking
	Next
	Previous
	First
	Last
New Variable…	
Zoom ▶	Header
	Body

Tools
in FBD

Network Elements ▶	Jump
	Return
	Function Block
	Input Variable
	Output Variable
	Horizontal Line Segment
	Vertical Line Segment
	Comment
Open Row	
Open Column	
Define Graphic Macros…	
Custom ▶	Custom 1
	Custom 2
	Custom 3
	Custom 4
	Custom 5
More Custom…	
Select Mode	
Interconnect Mode	
Auto Connect	
Recalculate Line	
Network List…	
Show MELSEC Code Of Networks	
List Operators/Operands	
Tracking ▶	Start Tracking
	Next
	Previous
	First
	Last
New Variable…	
Zoom ▶	Header
	Body

Tools
in Body SFC

Edit Action Association
Edit Step Comment
Edit Transition Condition
End Edit
List Operands
List Operators
New Variable
Zoom Macro
Zoom Body
Zoom Into
Zoom Back
Find Error

Tools

in Declaration editors

- Expand Declaration
- Collapse Declaration
- Sort ▶
  - Class
  - Identifier
  - Address ⑰
  - Type
- Find Overlapping Addresses
- Address Batch Replace ①
- Export to Excel file
- Import from Excel file

Tools

in the Entry Data Monitor (EDM)

- Expand Declaration
- Collapse Declaration
- Read from PLC
- Write to PLC
- Read from File
- Write to File
- Setup…
- Always on top

Tools

when no project is opened

- Reorganize Database

Tools

in Show Verify Differences

- Next Difference
- Previous Difference
- Goto Editor
- Show Step Number

Online

Transfer Setup ▶	Ports
	Modem
	Project
Start Monitoring (⑳, see below)	
Stop Monitoring	
Monitor Header ①	
Entry Data Monitor	
Modify variable value	
Online-Change Mode ① ②	
Monitoring Mode	
Start In Cycle Monitor ① ③	
Stop In Cycle Monitor ① ③	
Change Instance… ①	
Start/Stop PLC ①	
PLC Redundancy Mode ⑬	
PLC Status	
PLC Keyword ▶	Current ①
	New ①
	Delete ①
	Disable Protection ⑨
	Enable Protection ⑨
Set PLC Time ⑬	
GX Simulator ①	
PLC Clear ▶	All ①
	Devices ⑧
	Latch ① ⑩
	Forced Inputs ④ ⑤
Format Drive ① ⑩	
File Info ⑩	
Close communications	

⑳ for ST / IL see below

Online

specific items: ST / IL

Start Monitoring ▶	Body
	Split Window

Debug

SFC Control ⑱	Set selected steps ⑩
	Reset Selected steps ⑩
	Replace active by selected steps ⑩
PLC Diagnostics	
Network Diagnostics ④	
Ethernet Diagnostics ④	
CC-Link Diagnostics ④	
System Monitor ⑪	
Online Module Change ⑬	
Device Edit	
Buffer memory batch	
System Errors	
User Errors ① ④	
HW Diagnostics ① ④	
Scan ①	
Show Step	
Forced input output registration/cancellation…	

View

Toolbar		
Statusbar		
Horizontal Scrollbar		
Grid		
Zoom	▶	50%
		75%
		100%
		150%
View mode ⑲	▶	Standard
		Mitsubishi Address
		IEC Address
		Toggle
Font…		
Colors	▶	Background…
		Selection…
		Text…
		Selected Text…
		Read Only Text…
		Locked Text…
		Foreground…
		Monitoring…
		Errors…
		Shade Left Top…
		Shade Right Bottom…
		Grid…
		Browser Items…
		SFC Raster Elements…
		Overwrite Mode Cursor
		Insert Mode Cursor
		Line Mode Cursor
		Cursor Fill
		ST Keywords…
		ST Operators…
		ST Comments…
		ST Constant Numbers…
		ST Constant Strings…
		ST Variables…
Save Workspace		
Load Workspace		
Extended Information ⑲	▶	Navigator
		SFC editor
Environment		
Configure Status Bar		
System Defaults		

Extras

Project Backup…	
Project Restore…	
Declare MXChange Tags ①	
Delete MXChange Tags ①	
Update Variables ①	
Find Unused Variables	
Export Variables ①	
Intelligent Function Utility ▶	Start ⑪
	Verify System Variables ① ⑪
Ethernet Configurator ⑨	
Options…	
SV ▶	Export Alarm Info… ⑩
	Update Time ⑩
	Options…

Window

- Tile Horizontally
- Tile Vertically
- Cascade
- Arrange Icons
- Close All
- Navigator
- Show Compile/Check Messages
- Show Next Error
- Show Previous Error
- Show Next Warning
- Show Previous Warning
- 1 Window 1
- 2 Window 2
- 3 Window 3
- 4 Window 4

Help

- Index
- Overview
- Glossary
- Error Messages
- Search…
- How to use Help
- About…

G Compliance List

This list provides a description of all the features of the GX IEC Developer programming and documentation system that comply with the specifications of IEC standard 61131, Part 3.

The format of the list corresponds to the tables in IEC 61131-3. This makes it possible to obtain additional information quickly by referring to the corresponding tables and the accompanying texts in the IEC standard publication.

Character set features

No.	Description	Note
1	ISO/IEC 646 character set “A-Z,_”	Supported
2	Lower case characters “a-z”	Supported
3a	Number sign “#”	Supported
3b	Pound sign “£”	Not supported
4a	Dollar sign “$”	Supported
4b	Currency sign	Not supported
5a	Vertical bar “I”	Not supported
5b	Exclamation mark “!”	Not supported
6a	Left and right brackets “[]”	Supported (Arrays)
6b	Left and right parentheses “()”	Supported

Tab. G-1: *Character set features*

Identifier features

No.	Description	Note
1	Upper case and numbers	Supported
2	Upper and lower case, numbers, embedded underlines	Supported
3	Upper and lower case, numbers, leading or embedded underlines	Supported

Tab. G-2: *Identifier features*

Comment feature

No.	Description	Note
1	(*This is a comment*)	Supported

Tab. G-3: *Comment feature*

Numeric literals

No.	Description	Note
1	Integer literals	Supported
2	Real literals	Supported
3	Real literals with exponents	Supported
4	Base 2 literals	Supported
5	Base 8 literals	Supported
6	Base 16 literals	Supported
7	Boolean zero (0) and one (1)	Supported
8	Boolean FALSE and TRUE	Supported

Tab. G-4: *Numeric literals*

Character string literals

No.	Description	Note
1	‘’, ‘A’, ‘ ‘, ‘$", ‘$R$L", ‘$0D$0A’, ‘$$1.00’	Supported

Tab. G-5: *Character string literals*

Two-character combinations in character strings

No.	Description	Note
2	$$	Not supported
3	$’	Not supported
4	$L or $l	Not supported
5	$N or $n	Not supported
6	$P or $p	Not supported
7	$R or $r	Not supported
8	$T or $t	Not supported

Tab. G-6: *Two-character combinations in character strings*

Duration literal features

No.	Description	Note
1a	Duration literals without underlines: Short prefix	Supported
1b	Long prefix	Supported
2a	Duration literals with underlines: Short prefix	Supported
2b	Long prefix	Supported

Tab. G-7: *Duration literal features*

Date and time of day literals

No.	Description	Note
1	Date literals (long prefix)	Not supported
2	Date literals (long prefix)	Not supported
3	Time of day literals (long prefix)	Not supported
4	Time of day literals (short prefix)	Not supported
5	Date and time literals (long prefix)	Not supported
6	Date and time literals (short prefix)	Not supported

Tab. G-8: *Date and time of day literals*

Date and time of day literals notation

No.	Description	Note
1	Time literals notation	Not supported

Tab. G-9: *Date and time of day literals notation*

Elementary data types

No.	Description	Note
1	BOOL	Supported
2	SINT	Not supported
3	INT	Supported
4	DINT	Supported
5	LINT	Not supported
6	USINT	Not supported
7	UINT	Not supported
8	UDINT	Not supported
9	ULINT	Not supported
10	REAL	Supported
11	LREAL	Not supported
12	TIME	Supported
13	DATE	Not supported
14	TIME_OF_DAY or TOD	Not supported
15	DATE_AND_TIME or DT	Not supported
16	STRING	Supported
17	BYTE	Not supported
18	WORD	Supported
19	DWORD	Supported
20	LWORD Plus: EDGE, CHANGE, SGRAY, GRAY, DGRAY, BCD, WFIX, DFIX	Not supported

Tab. G-10: *Elementary data types*

Hierarchy of generic data types

No.	Description	Note

	ANY
	      ANY_NUM
	          ANY_REAL
	                        REAL
	          ANY_INT
	                        DINT
	                        INT
	
	ANY _BIT
	      DWORD
	      WORD
	      BOOL
	
	TIME	

Tab. G-11: *Hierarchy of generic data types*

Data type declaration feature

No.	Description	Note
1	Direct derivation of elementary data types	Not supported
2	Enumerated data types	Not supported
3	Subrange data types	Not supported
4	Array data types	Supported
5	Structured data types	Supported

Tab. G-12: *Data type declaration feature*

Default initial values

No.	Description	Note
	BOOL = FALSE	Supported
	INT, DINT = 0	Supported
	WORD, DWORD = 0	Supported
	REAL = 0.0	Supported
	TIME = T#0s	Supported

Tab. G-13: *Default initial values*

Data type initial value declaration features

No.	Description	Note
1	Initialisation of directly derived types	Not supported
2	Initialisation of enumerated data types	Not supported
3	Initialisation of subrange data types	Not supported
4	Initialisation of array data types	Not supported
5	Initialisation of structured data type elements	Not supported
6	Initialisation of derived structure data types	Not supported

Tab. G-14: *Data type initial value declaration features*

Location and size prefix features for directly represented variables

No.	Description	Note
1	I, Input location	Supported
2	Q, Output location	Supported
3	M, Memory location	Supported
4	X, Single bit size	Supported
5	None, Single bit size	Not supported
6	B, Byte (8 bits) size	Not supported
7	W, Word (16 bits) size	Supported
8	D, Double word (32 bits) size	Supported
9	L, Long (quad) word (64 bits) size	Not supported
	* See table on page 6-7	

Tab. G-15: *Location and size prefix features for directly represented variables*

Variable declaration keywords

No.	Description	Note
	VAR / VAR_CONSTANT	Supported
	VAR_INPUT	Supported
	VAR_OUTPUT	Supported
	VAR_IN_OUT	Supported
	VAR_GLOBAL / VAR_GLOBAL_CONSTANT	Supported
	VAR_ACCESS	Not supported
	RETAIN	Not supported
	CONSTANT	Not supported
	AT	Supported

Tab. G-16: *Variable declaration keywords*

Variable type assignment features

No.	Description	Note
1	Declaration of directly represented, non-retentive variables	Supported
2	Declaration of directly represented retentive variables	Not supported
3	Declaration of locations of symbolic variables	Supported
4	Array location assignment	Supported
5	Automatic memory allocation of symbolic variables	Supported
6	Array declaration	Supported
7	Retentive array declaration	Not supported
8	Declaration of structured variables	Supported

Tab. G-17: *Variable type assignment features*

Variable initial value assignment features

No.	Description	Note
1	Initialisation of directly represented, non-retentive variables	Not supported
2	Initialisation of directly represented retentive variables	Not supported
3	Location and initial value assignment to symbolic variables	Not supported
4	Array location assignment and initialisation	Not supported
5	Initialisation of symbolic variables	Not supported
6	Array initialisation	Not supported
7	Retentive array declaration and initialisation	Not supported
8	Initialisation of structured variables	Not supported
9	Initialisation of constants	Not supported

Tab. G-18: *Variable initial value assignment features*

Graphical negation of Boolean signals

No.	Description	Note
1	Negated input	Supported
2	Negated output	Supported

Tab. G-19: *Graphical negation of Boolean signals*

Use of EN input and ENO output

No.	Description	Note
1	Use of EN and ENO Required for Ladder Diagram (LD) language	Supported
2	Use of EN and ENO Optional for Function Block Diagram (FBD) language	Supported
3	FBD without EN and ENO	Supported

Tab. G-20: *Use of EN input and ENO output*

Typed and overloaded functions

No.	Description	Note
1	Overloaded functions (untyped)	Supported*
2	Typed functions	Not supported
	* Implicit typing through passing of the actual parameters	

Tab. G-21: *Typed and overloaded functions*

Type conversion function features

No.	Description	Note
1	*_TO_**	Supported
2	TRUNC	Not supported
3	BCD_TO_**	Supported
4	*_TO_BCD	Supported

Tab. G-22: *Type conversion function features*

Standard functions with one numeric variable

No.	Description	Note
1	ABS	Supported
2	SQRT	Supported*
3	LN	Supported*
4	LOG	Not supported
5	EXP	Supported*
6	SIN	Supported*
7	COS	Supported*
8	TAN	Supported*
9	ASIN	Supported*
10	ACOS	Supported*
11	ATAN	Supported*
	* Only for PLC systems that support floating point values: Q, QnA, AnA, AnU	

Tab. G-23: *Standard functions with one numeric variable*

Standard arithmetic functions

No.	Description	Note
12	ADD	Supported
13	MUL	Supported
14	SUB	Supported
15	DIV	Supported
16	MOD	Supported
17	EXPT	Supported
18	MOVE	Supported*
	* Only for PLC systems that support floating point values: Q, QnA, AnA, AnU	Supported

Tab. G-24: *Standard arithmetic functions*

Standard bit-shift functions

No.	Description	Note
1	SHL	Supported
2	SHR	Supported
3	ROR	Supported
4	ROL	Supported

Tab. G-25: *Standard bit-shift functions*

Standard bitwise Boolean functions

No.	Description	Note
5	AND	Supported
6	OR	Supported
7	XOR	Supported
8	NOT	Supported

Tab. G-26: *Standard bitwise Boolean functions*

Standard selection functions

No.	Description	Note
1	SEL	Supported
2a	MAX	Supported
2b	MIN	Supported
3	LIMIT	Supported
4	MUX	Supported

Tab. G-27: *Standard selection functions*

Standard comparison functions

No.	Description	Note
5	GT	Supported
6	GE	Supported
7	EQ	Supported
8	LE	Supported
9	LT	Supported
10	NE	Supported

Tab. G-28: *Standard comparison functions*

Standard character string functions

No.	Description	Note
1	LEN	Supported *
2	LEFT	Supported *
3	RIGHT	Supported *
4	MID	Supported *
5	CONCAT	Supported *
6	INSERT	Supported *
7	DELETE	Supported *
8	REPLACE	Supported *
9	FIND	Supported *
	* Only for PLC systems that support character strings: Q, QnA	

Tab. G-29: *Standard character string functions*

Functions of time data types

No.	Description	Note
1	ADD (TIME, TIME, TIME)	Not supported
2	ADD (TOD, TIME, TOD)	Not supported
3	ADD (DT, TIME, DT)	Not supported
4	SUB (TIME, TIME, TIME)	Not supported
5	SUB (DATE, DATE, TIME)	Not supported
6	SUB (TOD, TIME, TOD)	Not supported
7	SUB (TOD, TOD, TIME)	Not supported
8	SUB (DT, TIME, DT)	Not supported
9	SUB (DT, DT, TIME)	Not supported
10	MUL (TIME, ANY_NUM, TIME)	Not supported
11	DIV (TIME, ANY_NUM, TIME)	Not supported
12	CONCAT (DATE, TOD, DT)	Not supported
13	DT_TO_TOD	Not supported
14	DT_TO_D	Not supported

Tab. G-30: *Functions of time data types*

Functions of enumerated data types

No.	Description	Note
1	SEL	Not supported
2	VIUX	Not supported
3	EQ	Not supported
4	NE	Not supported

Tab. G-31: *Functions of enumerated data types*

Function block I/O parameter usage features

No.	Description	Note
1	Input read	Supported
2	Input write	Supported
3	Output read	Supported
4	Output write	Supported

Tab. G-32: *Function block I/O parameter usage features*

Function block declaration features

No.	Description	Note
1	RETAIN qualifier on internal variables	Not supported
2	RETAIN qualifier on output variables	Not supported
3	RETAIN qualifier on internal function blocks	Not supported
4a	Input/output declaration (textual)	Supported
4b	Input/output declaration (graphical)	Not supported
5a	Function block instance name as input (textual)	Not supported
5b	Function block instance name as input (graphical)	Not supported
6a	Function block instance name as input/output (textual)	Not supported
6b	Function block instance name as input/output (graphical)	Not supported
8a	Declaration of rising edge inputs (textual)	Not supported
8b	Declaration of falling edge inputs (textual)	Not supported
9a	Declaration of rising edge inputs (graphical)	Not supported
9b	Declaration of falling edge inputs (graphical)	Not supported

Tab. G-33: *Function block declaration features*

Standard bistable function blocks

No.	Description	Note
1	Bistable function block (set dominant)	Supported
2	Bistable function block (reset dominant)	Supported
3	Semaphore with non-interruptible “Test and Set”	Not supported

Tab. G-34: *Standard bistable function blocks*

Standard edge detection function blocks

No.	Description	Note
1	Rising edge detector	Supported
2	Falling edge detector	Supported

Tab. G-35: *Standard edge detection function blocks*

Standard counter function blocks

No.	Description	Note
1	Up-counter	Supported
2	Down-counter	Supported
3	Up/Down-counter	Supported

Tab. G-36: *Standard counter function blocks*

Standard timer function blocks

No.	Description	Note
1	TP	Supported
2a	TON	Supported
2b	T—-0	Not supported
3a	TOF	Supported
3b	0—-T	Not supported

Tab. G-37: *Standard timer function blocks*

Standard timer function blocks — timing diagrams

No.	Description	Note
1	TP	Supported
2	TON	Supported
3	TOF	Supported

Tab. G-38: *Standard timer function blocks — timing diagrams*

Program declaration features

No.	Description	Note
1-9b	Features 1 to 9b of table 23	
10	Formal input and output parameters	Not supported
11-14	Features 1-4 of table 17	
15-18	Features 1-4 of table 18	
19	Use of directly represented variables	Supported
20	VAR_GLOBAL...END_VAR declaration within a program	Supported
21	VAR_ACCESS...END_VAR declaration within a program	Not supported

Tab. G-39: *Program declaration features*

Step features

No.	Description	Note
1	Step - graphical form with directed links Initial step - graphical form with directed links	Supported
2	Step - textual form without directed links Initial step - textual form without directed links	Not supported
3a	Step flag - general form: name.X = a variable of type BOOL	Not supported
3b	Step flag - direct connection of Boolean variable name.X to right side of step	Not supported
4	Step elapsed time - general form name.T = a variable of type TIME	Not supported

Tab. G-40: *Step features*

Transitions and transition conditions

No.	Description	Note
1	Transition condition using ST language	Supported
2	Transition condition using LD language	*
3	Transition condition using FBD language	*
4	Use of transition connector	Supported
4a	Transition condition in LD language with transition connector	Supported
4b	Transition condition in FBD language with transition connector	Supported
5	Transition condition in ST in textual form	Not supported
6	Transition condition in IL in textual form	Not supported
7	Transition with transition name	Supported
7a	Transition condition using LD language	Supported
7b	Transition condition using FBD language	Supported
7c	Transition condition using IL language	Supported
7d	Transition condition using ST language	Supported
	* Supported through connectors	

Tab. G-41: *Transitions and transition conditions*

Declaration of actions

No.	Description	Note
1	Any Boolean variable declared in a VAR or VAR_OUTPUT block, or their graphical equivalents, can be an action.	Supported
2l	Graphical declaration of an action in LD language	Supported
2s	Inclusion of SFC elements in an action	Not supported
2f	Graphical declaration of an action in FBD language	Supported
3s	Textual declaration of an action in ST language	Supported
3i	Textual declaration of an action in IL language	Supported

Tab. G-42: *Declaration of actions*

Step/action association

No.	Description	Note
1	Action block	*
2	Concatenated action blocks	*
3	Textual step body with action calls	Not supported
4	Action block “d” field	Not supported
	* Implemented through the editors	

Tab. G-43: *Step/action association*

Action block features

No.	Description	Note
1	“a”: Qualifier	Not supported
2	“b”: Action name	Not supported
3	“c”: Boolean “indicator” variables	Not supported
4	“d”: Action using IL language	Not supported
5	“d”: Action using ST language	Not supported
6	“d”: Action using LD language	Not supported
7	“d”: Action using FBD language	Not supported
8	Use of action blocks in ladder diagrams	Not supported
9	Use of action blocks in function block diagrams	Not supported

Tab. G-44: *Action block features*

Action qualifiers

No.	Description	Note
1	None: Non-stored (mull qualifier)	Supported
2	N: Non-stored	Not supported
3	R: overriding Reset	Not supported
4	S: Set (stored)	Not supported
5	L: time Limited	Not supported
6	D: time Delayed	Not supported
7	P: Pulse (edge)	Not supported
8	SD: Stored and time Delayed	Not supported
9	DS: Delayed and Stored	Not supported
10	SL: Stored and time Limited	Not supported

Tab. G-45: *Action qualifiers*

Sequence evolution

No.	Description	Note
1	Single sequence	Supported
2a	Divergence of sequence selection, priority of evaluation from left to right	Supported
2b	Divergence of sequence selection, user-defined priority of evaluation (numbers)	Not supported
2c	Divergence of sequence selection with user-defined mutual exclusion of transition conditions as specified by IEC 848.	Not supported
3	Convergence of sequence selection	Supported
4	Simultaneous sequences — divergence Simultaneous sequences — convergence	Supported
5abc	Sequence skip (corresponding to 2a, b, c)	Supported (2a only)
6abc	Sequence loop (corresponding to 2a, b, c)	Not supported
7	Directional arrows in SFC charts	Not supported

Tab. G-46: *Sequence evolution*

Compatible SFC features (graphical only)

No.	Description	Note
40	1, 3a, 3b, 4	Not supported
41	1, 2, 3, 4, 4a, 4b, 7, 7a, 7b	Not supported
42	1, 2l, 2s, 2f	Not supported
43	1, 2, 4	Not supported
44	1 to 9	Not supported
45	1 to 10	Not supported
46	1 to 7	Not supported
57	all	Not supported

Tab. G-47: *Compatible SFC features (graphical only)*

SFC minimal compliance requirements (graphical only)

No.	Description	Note
40	1	Supported
41	1 or 2 or 3 or (4 and (4a or 4b)) or (7 and (7a or 7b or 7c or 7d))	Supported
42	1 or 2l or 2f	Supported
43	1 or 2 or 4	*
45	1 or 2	Supported
46	1 and (2a or 2b or 2c) and 3 and 4	Supported
47	(1 or 2) and (3 or 4) and (5 or 6) and (7 or 8) and (9 or 10) and (11 or 12)	Not supported
	* Implemented through the editors	

Tab. G-48: *SFC minimal compliance requirements (graphical only)*

Configuration and resource declaration features

No.	Description	Note
1	CONFIGURATION...END_CONFIGURATION construction	Not supported
2	VAR_GLOBAL...END_VAR construction within CONFIGURATION	Not supported
3	RESOURCE...ON...END_RESOURCE construction	Not supported
4	VAR_GLOBAL...END_VAR construction within RESOURCE	Not supported
5a	Periodic TASK construction within RESOURCE	Not supported
5b	Non-periodic TASK construction within RESOURCE	Not supported
6a	PROGRAM declaration PROGRAM-to-TASK association	Not supported
6b	PROGRAM declaration with FUNCTION BLOCK-to-TASK association	Not supported
6c	PROGRAM declaration with no TASK association	Not supported
7	Declaration of directly represented variables in VAR_GLOBAL	Not supported
8a	Connection of directly represented variables to PROGRAM inputs	Not supported
8b	Connection of GLOBAL variables to PROGRAM inputs	Not supported
9a	Connection of PROGRAM outputs to directly represented variables	Not supported
9b	Connection of PROGRAM outputs to GLOBAL variables	Not supported
10a	VAR_ACCESS...END_VAR construction	Not supported
10b	Access paths to directly represented variables	Not supported
10c	Access paths to PROGRAM inputs	Not supported
10d	Access paths to GLOBAL variables in RESOURCES	Not supported
10e	Access paths to GLOBAL variables in CONFIGURATIONS	Not supported
10f	Access paths to PROGRAM outputs	Not supported

Tab. G-49: *Configuration and resource declaration features*

Task features

No.	Description	Note
1a	Textual declaration of periodic TASKS according to feature 5a of table 49	Not supported
1b	Textual declaration of non-periodic TASKS according to feature 5b of table 49	Not supported
2a	Graphical representation of periodic TASKS	Supported
2b	Graphical representation of non-periodic TASKS	Supported
3a	Textual association with PROGRAMS according to feature 6a of table 49	Not supported
3b	Textual association with FUNCTION BLOCKS according to feature 6b of table 49	Not supported
4a	Graphical association with PROGRAMS (within RESOURCES)	Supported
4b	Graphical association with FUNCTION BLOCKS (within PROGRAMS in RESOURCES)	Supported
5a	Non-pre-emptive scheduling	Supported
5b	Pre-emptive scheduling	Not supported

Tab. G-50: *Task features*

Examples of instruction fields

No.	Description	Note
1	Label, Operator, Operand	Supported

Tab. G-51: Examples of instruction fields

Instruction List (IL) operators

No.	Description	Note
1	LD	Supported
2	ST	Supported
3	S, R	Supported
4	AND, N, (	Supported
5	&, N, (	Not supported
6	OR, N, (	Supported
7	XOR, N, (	Supported
8	ADD, (	Supported
9	SUB, (	Supported
10	MUL, (	Supported
11	DIV, (	Supported
12	GT, (	Supported
13	GE, (	Supported
14	EQ, (	Supported
15	NE, (	Supported
16	LE, (	Supported
17	LT, (	Supported
18	JMP, C, N	Supported
19	CAL, C, N	Supported
20	RET, C, N	Supported
21	)	Supported

Tab. G-52: Instruction List (IL) operators

Function block invocation features for IL language

No.	Description	Note
1	CALL with input parameter list	Supported
2	CALL with load/store of input parameters	Supported
3	Use of input operators	Not supported

Tab. G-53: Function block invocation features for IL language

Standard function block input operators for IL language

No.	Description	Note
	!!!Only relevant when supported in 53.3!!!	

Tab. G-54: *Standard function block input operators for IL language*

Operators of the ST language

No.	Description	Note
1	(Expression) parenthesisation	Supported
2	Function call (argument list)	Supported
3	**	Supported
4	- (Negation)	Supported
5	NOT	Supported
6	*	Supported
7	/	Supported
8	MOD	Supported
9	+	Supported
10	- (Subtraction)	Supported
11	<, >, <=, >=	Supported
12	=	Supported
13	<>	Supported
14	&	Supported
15	AND	Supported
16	XOR	Supported
17	OR	Supported

Tab. G-55: *Operators of the ST language*

ST language statements

No.	Description	Note
1	Assignment :=	Supported
2	Function block invocation and FB output usage	Supported
3	RETURN	Supported
4	IF	Supported
5	CASE	Supported
6	FOR	Supported
7	WHILE	Supported
8	REPEAT	Supported
9	EXIT	Supported
10	Empty statement	Supported

Tab. G-56: *ST language statements*

Representation of lines and blocks

No.	Description	Note
1	Horizontal lines: ISO 646 "–" character, semigraphic	Not supported
2	Horizontal lines, graphic	Supported
3	Vertical lines ISO 646 "+" character, semigraphic	Not supported
4	Vertical lines, graphic	Supported
5	Horizontal/vertical connection ISO 646 "+" character, semigraphic	Not supported
6	Horizontal/vertical connection, graphic	Supported
7	Line crossing without connection ISO 646 characters, semigraphic	Not supported
8	Line crossing without connection, graphic	Supported
9	Connected and non-connected corners ISO 646 "+" character, semigraphic	Not supported
10	Connected and non-connected corners, graphic	Supported
11	Blocks with connecting lines, ISO 646 character, semigraphic	Not supported
12	Blocks with connecting lines, graphic	Supported
13	Connectors using ISO 646 characters, semigraphic	Not supported
14	Connectors, graphic	Supported

Tab. G-57: *Representation of lines and blocks*

Graphic execution control elements, examples

No.	Description	Note
1	Unconditional jump in FBD language	Supported
2	Unconditional jump in LD language	Supported
3	Conditional jump in FBD language	Supported
4	Conditional jump in LD language	Supported
5	Conditional return in LD language	Supported
6	Conditional return in FBD language	Supported
7	Unconditional return from functions and function blocks	Supported
8	Alternative representation in LD language	Supported

Tab. G-58: *Graphic execution control elements, examples*

Power bar

No.	Description	Note
1	Left power bar with attached horizontal link	Supported
2	Right power bar with attached horizontal link	Not supported

Tab. G-59: *Power bar*

Link elements

No.	Description	Note
1	Horizontal link	Supported
2	Vertical link (with attached horizontal links)	Supported

Tab. G-60: *Link elements*

Contacts

No.	Description	Note
1	Normally open contact (make) —| |—	Supported *
2	Normally open contact (make) —! !—	Not supported *
3	Normally closed contact (break) —| / |—	Supported *
4	Normally closed contact (break) —! / !—	Not supported *
5	Positive transition-sensing contact —| P |—	Not supported *
6	Positive transition-sensing contact —! P !—	Not supported *
7	Negative transition-sensing contact —| N |— Negative transition-sensing contact —! N !—	Not supported * Not supported *
	* Graphical	

Tab. G-61: *Contacts*

Coils (terminating operations)

No.	Description	Note
1	Coil —( )—	Supported *
2	Negated coil —(/)—	Supported *
3	SET (latch) coil —(S)—	Supported *
4	RESET (unlatch) coil —(R)—	Supported *
5	Retentive (memory) coil —(M)—	Not supported
6	SET retentive (memory) coil —(SM)—	Not supported
7	RESET retentive (memory) coil —(RM)—	Not supported
8	Positive transition-sensing coil —(P)—	Not supported
9	Negative transition-sensing coil —(N)—	Not supported
	* Graphical	

Tab. G-62: *Coils (terminating operations)*

Index

MITSUBISHI ELECTRIC

HEADQUARTERS

MITSUBISHI ELECTRIC **EUROPE**
EUROPE B.V.
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0) 2102 / 486-0
Fax: +49 (0) 2102 / 486-1120
e mail: megfamail@meg.mee.com

MITSUBISHI ELECTRIC **FRANCE**
EUROPE B.V.
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 1 55 68 55 68
Fax: +33 1 55 68 56 85
e mail: factory.automation@fra.mee.com

MITSUBISHI ELECTRIC **IRELAND**
EUROPE B.V.
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0) 1 / 419 88 00
Fax: +353 (0) 1 / 419 88 90
e mail: sales.info@meir.mee.com

MITSUBISHI ELECTRIC **ITALY**
EUROPE B.V.
Italian Branch
Via Paracelso 12
I-20041 Agrate Brianza (MI)
Phone: +39 039 6053 1
Fax: +39 039 6053 312
e mail: factory.automation@it.mee.com

MITSUBISHI ELECTRIC **SPAIN**
EUROPE B.V.
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés
Phone: +34 9 3 / 565 3131
Fax: +34 9 3 / 589 2948
e mail: industrial@sp.mee.com

MITSUBISHI ELECTRIC **UK**
EUROPE B.V.
UK Branch
Travellers Lane
GB-Hatfield Herts. AL10 8 XB
Phone: +44 (0) 1707 / 27 61 00
Fax: +44 (0) 1707 / 27 86 95
e mail: automation@meuk.mee.com

MITSUBISHI ELECTRIC **JAPAN**
CORPORATION
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 6221 6060
Fax: +81 3 6221 6075

MITSUBISHI ELECTRIC **USA**
AUTOMATION
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 / 478 21 00
Fax: +1 847 / 478 22 83

MIDDLE EAST REPRESENTATIVE

TEXEL Electronics Ltd. **ISRAEL**
Box 6272
IL-42160 Netanya
Phone: +972 (0) 9 / 863 08 91
Fax: +972 (0) 9 / 885 24 30
e mail: texel_me@netvision.net.il

EUROPEAN REPRESENTATIVES

GEVA **AUSTRIA**
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0) 2252 / 85 55 20
Fax: +43 (0) 2252 / 488 60
e mail: office@geva.at

TEHNIKON **BELARUS**
Oktjabrskaya 16/5, Ap 704
BY-220030 Minsk
Phone: +375 (0)17 / 22 75 704
Fax: +375 (0)17 / 22 76 669
e mail: tehnikon@belsonet.net

Getronics b.v. **BELGIUM**
Control Systems
Pontbeeklaan 43
B-1731 Asse-Zellik
Phone: +32 (0) 2 / 467 17 51
Fax: +32 (0) 2 / 467 17 45
e mail: infoautomation@getronics.com

TELECON CO. **BULGARIA**
4, A. Ljapchev Blvd.
BG-1756 Sofia
Phone: +359 (0) 2 / 97 44 05 8
Fax: +359 (0) 2 / 97 44 06 1
e mail: —

INEA CR d.o.o. **CROATIA**
Losinjska 4 a
HR-10000 Zagreb
Phone: +385 (0) 1 / 36 940-01
Fax: +385 (0) 1 / 36 940-03
e mail: inea@inea.hr

AutoCont **CZECH REPUBLIC**
Control Systems s.r.o.
Nemocnicni 12
CZ-702 00 Ostrava 2
Phone: +420 59 / 6152 111
Fax: +420 59 / 6152 562
e mail: consys@autocont.cz

louis poulsen **DENMARK**
industri & automation
Geminivej 32
DK-2670 Greve
Phone: +45 (0) 70 / 10 15 35
Fax: +45 (0) 43 / 95 95 91
e mail: lpia@lpmail.com

UTU Elektrotehnika AS **ESTONIA**
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0) 6 / 51 72 80
Fax: +372 (0) 6 / 51 72 88
e mail: utu@utu.ee

Beijer Electronics OY **FINLAND**
Ansatie 6a
FIN-01740 Vantaa
Phone: +358 (0) 9 / 886 77 500
Fax: +358 (0) 9 / 886 77 555
e mail: info@beijer.fi

UTECO A.B.E.E. **GREECE**
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +302 (0) 10 / 42 10 050
Fax: +302 (0) 10 / 42 12 033
e mail: sales@uteco.gr

Meltrade Automatika Kft. **HUNGARY**
55, Harmat St.
HU-1105 Budapest
Phone: +36 (0)1 / 2605 602
Fax: +36 (0)1 / 2605 602
e mail: office@meltrade.hu

SIA POWEL **LATVIA**
Lienes iela 28
LV-1009 Riga
Phone: +371 784 / 22 80
Fax: +371 784 / 22 81
e mail: utu@utu.lv

EUROPEAN REPRESENTATIVES

UAB UTU POWEL **LITHUANIA**
Savanoriu pr. 187
LT-2053 Vilnius
Phone: +370 (0) 52323-101
Fax: +370 (0) 52322-980
e mail: powel@utu.lt

INTEHSIS SRL **MOLDOVA**
Cuza-Voda 36/1-81
MD-2061 Chisinau
Phone: +373 (0)2 / 562 263
Fax: +373 (0)2 / 562 263
e mail: intehsis@mdl.net

Getronics b.v. **NETHERLANDS**
Control Systems
Donauweg 2 B
NL-1043 AJ Amsterdam
Phone: +31 (0) 20 / 587 67 00
Fax: +31 (0) 20 / 587 68 39
e mail: info.gia@getronics.com

Beijer Electronics AS **NORWAY**
Teglverksveien 1
N-3002 Drammen
Phone: +47 (0) 32 / 24 30 00
Fax: +47 (0) 32 / 84 85 77
e mail: info@beijer.no

MPL Technology Sp. z o.o. **POLAND**
ul. Sliczna 36
PL-31-444 Kraków
Phone: +48 (0) 12 / 632 28 85
Fax: +48 (0) 12 / 632 47 82
e mail: krakow@mpl.pl

Sirius Trading & Services srl **ROMANIA**
Str. Biharia No. 67-77
RO-013981 Bucuresti 1
Phone: +40 (0) 21 / 201 1146
Fax: +40 (0) 21 / 201 1148
e mail: sirius@siriustrading.ro

INEA d.o.o. **SLOVENIA**
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0) 1-513 8100
Fax: +386 (0) 1-513 8170
e mail: inea@inea.si

Beijer Electronics AB **SWEDEN**
Box 426
S-20124 Malmö
Phone: +46 (0) 40 / 35 86 00
Fax: +46 (0) 40 / 35 86 02
e mail: info@beijer.se

ECONOTEC AG **SWITZERLAND**
Postfach 282
CH-8309 Nürensdorf
Phone: +41 (0) 1 / 838 48 11
Fax: +41 (0) 1 / 838 48 12
e mail: info@econotec.ch

GTS **TURKEY**
Darülaceze Cad. No. 43 Kat. 2
TR-80270 Okmeydani-Istanbul
Phone: +90 (0) 212 / 320 1640
Fax: +90 (0) 212 / 320 1649
e mail: gts@turk.net

CSC Automation Ltd. **UKRAINE**
15, M. Raskova St., Fl. 10, Office 1010
UA-02002 Kiev
Phone: +380 (0) 44 / 238-83-16
Fax: +380 (0) 44 / 238-83-17
e mail: csc-a@csc-a.kiev.ua

EURASIAN REPRESENTATIVES

Avtomatika Sever Ltd. **RUSSIA**
Lva Tolstogo St. 7, Off. 311
RU-197376 St Petersburg
Phone: +7 812 / 11 83 238
Fax: +7 812 / 11 83 239
e mail: as@avtsev.spb.ru

CONSYS **RUSSIA**
Promyshlennaya St. 42
RU-198099 St Petersburg
Phone: +7 812 / 325 36 53
Fax: +7 812 / 147 20 55
e mail: consys@consys.spb.ru

Electrotechnical **RUSSIA**
Systems Siberia
Partizanskaya St. 27, Office 306
RU-121355 Moscow
Phone: +7 095/ 416-4321
Fax: +7 095/ 416-4321
e mail: info@eltechsystems.ru

Electrotechnical **RUSSIA**
Systems Siberia
Shetinkina St. 33, Office 116
RU-630088 Novosibirsk
Phone: +7 3832 / 22-03-05
Fax: +7 3832 / 22-03-05
e mail: info@eltechsystems.ru

Elektrostyle **RUSSIA**
ul. Garschina 11
RU-140070 Moscow
Phone: +7 095 / 514 9316
Fax: +7 095 / 514 9317
e mail: info@estl.ru

Elektrostyle **RUSSIA**
Krasnij Prospekt 220-1
Office No. 312
RU-630049 Novosibirsk
Phone: +7 3832 / 10 66 18
Fax: +7 3832 / 10 66 26
e mail: info@estl.ru

ICOS **RUSSIA**
Industrial Computer Systems Zao
Ryazanskij Prospekt 8a, Office 100
RU-109428 Moscow
Phone: +7 095 / 232 - 0207
Fax: +7 095 / 232 - 0327
e mail: mail@icos.ru

NPP Uralelektra **RUSSIA**
ul. Sverdlova 11a
RU-620027 Ekaterinburg
Phone: +7 34 32 / 53 27 45
Fax: +7 34 32 / 53 27 45
e mail: elektra@etel.ru

SSMP Rosgidromontazh Ltd. **RUSSIA**
23, Lesoparkovaya Str.
RU-344041 Rostov On Don
Phone: +7 8632 / 36 00 22
Fax: +7 8632 / 36 00 26
e mail: —

STC Drive Technique **RUSSIA**
ul. Bajkalskaja 239, Office 2 - 23
RU-664075 Irkutsk
Phone: +7 3952 / 24 38 16
Fax: +7 3952 / 23 02 98
e mail: privod@irk.ru

STC Drive Technique **RUSSIA**
Poslannikov Per. 9, str.1
RU-107005 Moscow
Phone: +7 095 / 790-72-10
Fax: +7 095 / 790-72-12
e mail: info@privod.ru

AFRICAN REPRESENTATIVE

CBI Ltd. **SOUTH AFRICA**
Private Bag 2016
ZA-1600 Isando
Phone: +27 (0) 11/ 928 2000
Fax: +27 (0) 11/ 392 2354
e mail: cbi@cbi.co.za

MITSUBISHI ELECTRIC INDUSTRIAL AUTOMATION

Gothaer Straße 8
D-40880 Ratingen

Phone: +49 2102 486-0
Hotline: +49 1805 000-765

Fax: +49 2102 486-7170
megfa-mail@meg.mee.com

www.mitsubishi-automation.de
www.mitsubishi-automation.com