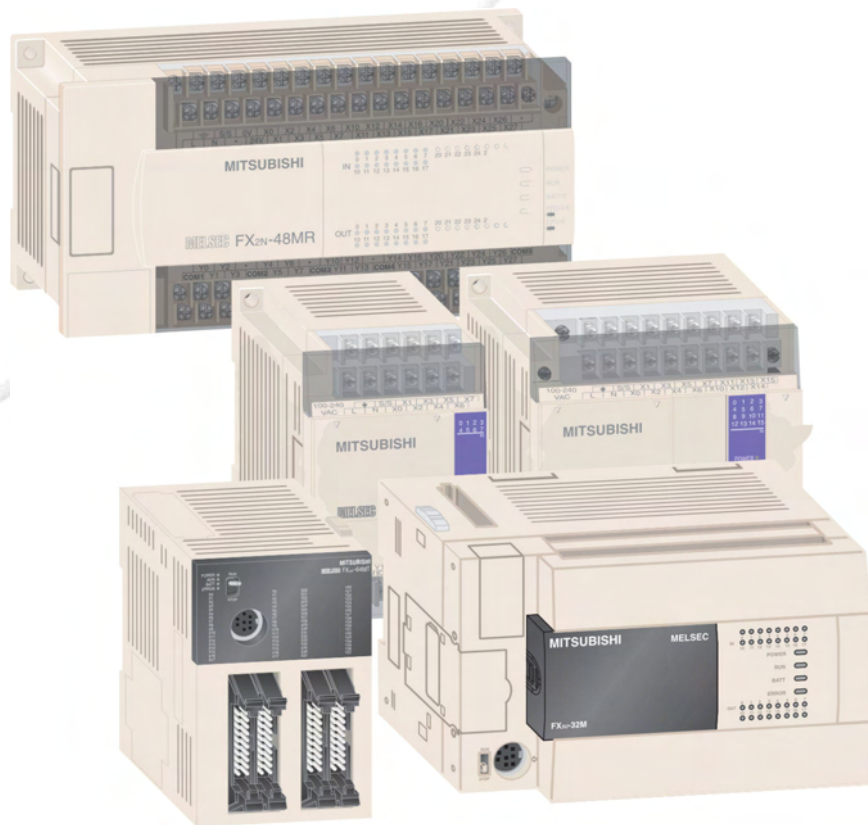


# FX Family

## Programmable Logic Controllers

## Training Manual



**GX Developer**



## About this Manual

The texts, illustrations and examples in this manual only explain the installation, operation and use of the *GX Developer* programming package.

If you have questions about the programming and operation of the programmable logic controllers mentioned in this manual please contact your dealer or one of our distributors (see back cover). Up-to-date information and answers to frequently-asked questions can be found on the Mitsubishi website at [www.mitsubishi-automation.com](http://www.mitsubishi-automation.com).

MITSUBISHI ELECTRIC EUROPE B.V. reserves the right to make changes to this manual or the technical specifications of its products at any time without notice.





**Training Manual**  
**GX Developer Programming Software Package**  
**Art.-no.:**

<b>Version</b>			<b>Changes / Additions / Corrections</b>
A	05/2007	pdp-dk	First edition
B	04/2010	pdp-dk	Chapter 2 and appendix: Addition of the base units of the FX3G and FX3UC series and of new modules for the FX3U series (FX3U-2HC, FX3U-4LC and FX3U-3A-ADP). Changes in chapter 22 (Ethernet communications)



---

# Safety Information

## For qualified staff only

This manual is only intended for use by properly trained and qualified electrical technicians who are fully acquainted with automation technology safety standards. All work with the hardware described, including system design, installation, setup, maintenance, service and testing, may only be performed by trained electrical technicians with approved qualifications who are fully acquainted with the applicable automation technology safety standards and regulations.

## Proper use of equipment

The programmable logic controllers are only intended for the specific applications explicitly described in this manual. Please take care to observe all the installation and operating parameters specified in the manual. All products are designed, manufactured, tested and documented in agreement with the safety regulations. Any modification of the hardware or software or disregarding of the safety warnings given in this manual or printed on the product can cause injury to persons or damage to equipment or other property. Only accessories and peripherals specifically approved by MITSUBISHI ELECTRIC may be used. Any other use or application of the products is deemed to be improper.

## Relevant safety regulations

All safety and accident prevention regulations relevant to your specific application must be observed in the system design, installation, setup, maintenance, servicing and testing of these products. The regulations listed below are particularly important. This list does not claim to be complete; however, you are responsible for knowing and applying the regulations applicable to you.

- VDE Standards
  - VDE 0100  
(Regulations for electrical installations with rated voltages up to 1,000 V)
  - VDE 0105  
(Operation of electrical installations)
  - VDE 0113  
(Electrical systems with electronic equipment)
  - VDE 0160  
(Configuration of electrical systems and electrical equipment)
  - VDE 0550/0551  
(Regulations for transformers)
  - VDE 0700  
(Safety of electrical appliances for household use and similar applications)
  - VDE 0860  
(Safety regulations for mains-powered electronic appliances and their accessories for household use and similar applications)
- Fire prevention regulations
- Accident prevention regulations
  - VBG No. 4 (Electrical systems and equipment)

---

### Safety warnings in this manual

In this manual special warnings that are important for the proper and safe use of the products are clearly identified as follows:



**DANGER:**

*Personnel health and injury warnings. Failure to observe the precautions described here can result in serious health and injury hazards.*



**CAUTION:**

*Equipment and property damage warnings. Failure to observe the precautions described here can result in serious damage to the equipment or other property.*

## General safety information and precautions

The following safety precautions are intended as a general guideline for using the PLC together with other equipment. These precautions must always be observed in the design, installation and operation of all control systems.



### CAUTION:

- **Observe all safety and accident prevention regulations applicable to your specific application. Installation, wiring and opening of the assemblies, components and devices may only be performed with all power supplies disconnected.**
- **Assemblies, components and devices must always be installed in a shockproof housing fitted with a proper cover and protective equipment.**
- **Devices with a permanent connection to the mains power supply must be integrated in the building installations with an all-pole disconnection switch and a suitable fuse.**
- **Check power cables and lines connected to the equipment regularly for breaks and insulation damage. If cable damage is found, immediately disconnect the equipment and the cables from the power supply and replace the defective cabling.**
- **Before using the equipment for the first time check that the power supply rating matches that of the local mains power.**
- **Residual current protective devices pursuant to DIN VDE Standard 0641 Parts 1-3 are not adequate on their own as protection against indirect contact for installations with positioning drive systems. Additional and/or other protection facilities are essential for such installations.**
- **EMERGENCY OFF facilities pursuant to EN 60204/IEC 204 VDE 0113 must remain fully operative at all times and in all control system operating modes. The EMERGENCY OFF facility reset function must be designed so that it cannot cause an uncontrolled or undefined restart.**
- **You must also implement hardware and software safety precautions to prevent the possibility of undefined control system states caused by signal line cable or core breaks.**
- **All relevant electrical and physical specifications must be strictly observed and maintained for all the modules in the installation.**



# Table of Contents

<b>1</b>	<b>Course Overview and Requirements</b>	
1.1	Modular PLC Training Hardware . . . . .	1-1
<b>2</b>	<b>The Hardware</b>	
2.1	General Introduction to PLCs . . . . .	2-1
2.1.1	History & Development . . . . .	2-1
2.1.2	The initial specification for the PLC . . . . .	2-1
2.1.3	Comparison of PLC and Relay Systems . . . . .	2-1
2.1.4	Programming . . . . .	2-2
2.1.5	Human Machine Interfaces . . . . .	2-2
2.2	What is a PLC? . . . . .	2-3
2.3	How PLCs Process Programs . . . . .	2-4
2.4	The MELSEC FX Family . . . . .	2-6
2.5	Selecting the Right Controller . . . . .	2-7
2.6	Controller Design . . . . .	2-9
2.6.1	Input and output circuits . . . . .	2-9
2.6.2	Layout of the MELSEC FX1S base units . . . . .	2-9
2.6.3	Layout of the MELSEC FX1N base units . . . . .	2-10
2.6.4	Layout of the MELSEC FX2N base units . . . . .	2-10
2.6.5	Layout of the MELSEC FX2NC base units . . . . .	2-11
2.6.6	Layout of the MELSEC FX3G base units . . . . .	2-11
2.6.7	Layout of the MELSEC FX3U base units . . . . .	2-12
2.6.8	Layout of the MELSEC FX3UC base units . . . . .	2-12
2.7	Wiring . . . . .	2-13
2.7.1	Power Supply . . . . .	2-13
2.7.2	Wiring of Inputs . . . . .	2-14
2.7.3	Wiring of Outputs . . . . .	2-15
2.8	Extending the Range of Digital Inputs/Outputs . . . . .	2-17
2.8.1	Extension Boards . . . . .	2-17
2.8.2	Compact Extension Units . . . . .	2-17
2.8.3	Modular Extension Blocks . . . . .	2-18
2.9	Extending for Special Functions . . . . .	2-19
2.9.1	Analog Modules . . . . .	2-20
2.9.2	High-Speed Counter Modules and Adapters . . . . .	2-23
2.9.3	Positioning Modules . . . . .	2-24
2.9.4	Network Modules for ETHERNET . . . . .	2-25

2.9.5	Network Modules for Profibus/DP	2-26
2.9.6	Network Modules for CC-Link	2-28
2.9.7	Network Module for DeviceNet	2-29
2.9.8	Network Module for CANopen	2-29
2.9.9	Network Module for AS-Interface	2-30
2.9.10	Interface Modules and Adapters	2-31
2.9.11	Communication Adapters	2-32
2.9.12	Setpoint Adapter Boards	2-33
2.10	System Configuration	2-34
2.10.1	Connection of Special Adapters	2-36
2.10.2	Basic Rules for System Configuration	2-38
2.10.3	Quick Reference Matrixes	2-39
2.11	I/O Assignment	2-41
2.11.1	Concept of assigning	2-41
2.11.2	Special function module address	2-42

### **3 GX Developer**

3.1	Advantages of GX-Developer	3-1
3.2	Programming Software Initialisation	3-2

### **4 Creating a Project**

4.1	Example PLC Program (COMPACT_PROG1)	4-1
4.1.1	Line Numbers	4-1
4.1.2	Principle of Operation	4-2
4.2	Start Up Procedure	4-3
4.3	Ladder Diagram Elements	4-5
4.4	Project Data List	4-5
4.5	Toggle Display Project Data List	4-6
4.6	Changing the Colour Attributes (Optional)	4-7
4.7	Entering the Ladder Diagram (COMPACT_PROG1)	4-9
4.8	Conversion to an Instruction Program	4-11
4.9	Saving the Project	4-12

### **5 Instruction List Programming**

5.1	Instruction List Program (COMPACT_PROG1)	5-1
5.2	Explanation - Instruction List Programming	5-3



<b>6</b>	<b>Find</b>	
6.1	Find Step Numbers	6-1
6.2	Find Device	6-3
6.3	Instruction Search	6-4
6.4	Cross Reference List	6-5
6.5	List of Used Devices	6-7
<b>7</b>	<b>Copying Projects</b>	
7.1	Copying of the project COMPACT_PROG1	7-1
<b>8</b>	<b>Modification of Ladder Diagrams</b>	
8.1	Modification of the project COMPACT_PROG2	8-1
8.2	Insertion of a new contact	8-3
8.3	Change of Device Detail	8-4
8.4	Inserting a Branch	8-5
8.5	Appention of New Program Blocks	8-6
8.6	Insertion of New Program Blocks	8-7
<b>9</b>	<b>Delete Functions</b>	
9.1	Overview	9-1
9.2	Deleting an Input Contact	9-2
9.3	Deleting a Branch	9-3
9.4	Deleting a Single Line	9-4
9.5	Deleting multiple lines	9-5
<b>10</b>	<b>Program Documentation</b>	
10.1	New Program Example: COMPACT_PROG4	10-1
10.2	Annotating the Program	10-3
10.3	Comments	10-5
10.4	Project Data List (Navigation Window)	10-6
10.5	Comment Format	10-7
10.6	Statements	10-9
10.7	Notes	10-10
10.8	Alias	10-11

<b>11</b>	<b>I/O Assignment</b>	
11.1	Checking the Input/Output Range . . . . .	11-1
<b>12</b>	<b>Downloading a Project to a PLC</b>	
12.1	Communications Setup . . . . .	12-1
12.1.1	Connection Setup Route . . . . .	12-3
12.2	Clearing of the PLC Memory . . . . .	12-4
12.3	Write Program to PLC . . . . .	12-5
12.4	Reducing the Number of Steps Transferred to the PLC . . . . .	12-8
<b>13</b>	<b>Executing the Project</b>	
13	Executing the Project . . . . .	13-1
<b>14</b>	<b>Monitoring</b>	
14.1	Monitoring the example program COMPACT_PROG4 . . . . .	14-1
14.2	Entry Data Monitoring . . . . .	14-3
14.3	Combined Ladder and Entry Data Monitoring . . . . .	14-6
14.4	Device Test Function . . . . .	14-7
<b>15</b>	<b>Program Verification</b>	
15.1	Verification of Example Programs . . . . .	15-1
<b>16</b>	<b>Serial Transfer – Upload</b>	
16.1	Example Program Upload . . . . .	16-1
<b>17</b>	<b>Sequential Function Chart (SFC)</b>	
17.1	SFC Elements . . . . .	17-2
17.1.1	Steps . . . . .	17-2
17.1.2	Transitions . . . . .	17-2
17.1.3	Initialization Step . . . . .	17-3
17.2	Rules for Sequences . . . . .	17-4
17.2.1	Divergence into Parallel Sequences . . . . .	17-4
17.2.2	Convergence of Parallel Sequences . . . . .	17-4
17.2.3	Divergence into Selective Sequences . . . . .	17-4
17.2.4	Convergence of Selective Sequences . . . . .	17-5

17.2.5	Jumps .....	17-5
17.2.6	Exit and Entry Steps .....	17-6
17.3	Example for Programming in SFC .....	17-7
17.4	Creating a SFC-Block .....	17-8
17.4.1	SFC Diagram editing screen .....	17-9
17.4.2	Block information .....	17-10
17.4.3	Editing the project .....	17-10
17.4.4	Transfer project .....	17-12
17.4.5	Monitor project .....	17-12
<b>18 Counters</b>		
18.1	Programm Example – COUNT DELAY .....	18-2
18.2	Programm Example – Batch Counter .....	18-5
18.2.1	BATCH1 .....	18-5
18.2.2	Modification of Program BATCH2 .....	18-6
<b>19 On Line Programming</b>		
19.1	Modification of the program COUNT DELAY .....	19-1
<b>20 FROM / TO INSTRUCTIONS</b>		
20.1	Exchanging Data with Special Function Modules .....	20-1
20.2	Buffer Memory Accessing Instructions .....	20-2
20.2.1	Read Buffer Memory (FROM) .....	20-4
20.2.2	Write to Buffer Memory (TO) .....	20-5
<b>21 FOR – NEXT Loops</b>		
21.1	Operation .....	21-1
21.1.1	Program Example .....	21-4
<b>22 Ethernet Communications</b>		
22.1	Configuring the PC on the Ethernet .....	22-2
22.2	Configuring the FX3U-ENET by FX Configurator-EN .....	22-3
22.3	Configuring GX Developer to access the PLC on Ethernet .....	22-7
22.4	Setting up a HMI of the GOT1000 Series (GT12, GT15 or GT16) .....	22-11
22.5	Setting up a HMI of the E1000 series .....	22-14
22.6	Communication via MX Component .....	22-18

<b>A</b>	<b>Appendix</b>	
A.1	Special Relays	A-1
A.1.1	PLC Status Diagnostic Information (M8000 to M8009)	A-2
A.1.2	Clock Devices and Real Time Clock (M8011 to M8019)	A-2
A.1.3	PLC Operation Mode (M8030 to M8039)	A-3
A.1.4	Error Detection (M8060 to M8069)	A-4
A.1.5	Extension Boards (Dedicated to FX1S and FX1N)	A-4
A.1.6	Analog Special Adapter and Adapter Boards for FX3G	A-5
A.2	Special Registers	A-6
A.2.1	PLC Status Diagnostic Information (D8000 to D8009)	A-6
A.2.2	Scan Information and Real Time Clock (D8010 to D8019)	A-7
A.2.3	PLC Operation Mode (D8030 to D8039)	A-7
A.2.4	Error Codes (D8060 to D8069)	A-8
A.2.5	Extension Boards (Dedicated to FX1S and FX1N)	A-8
A.2.6	Analog Special Adapter and Adapter Boards for FX3G	A-9
A.3	Error Code List	A-10
A.3.1	Error codes 6101 to 6409	A-10
A.3.2	Error codes 6501 to 6510	A-11
A.3.3	Error codes 6610 to 6632	A-12
A.3.4	Error codes 6701 to 6710	A-13
A.4	Number of Occupied Input/Output Points and Current Consumption	A-14
A.4.1	Interface Adapter Boards and Communication Adapter Boards	A-14
A.4.2	Special Adapters	A-15
A.4.3	Extension Blocks	A-15
A.4.4	Special Function Modules	A-16
A.5	PLC Components Glossary	A-17

# 1 Course Overview and Requirements

This course has been specially produced as an introduction to the PLC's of Mitsubishi's FX family utilising the GX Developer Version 8 software package.

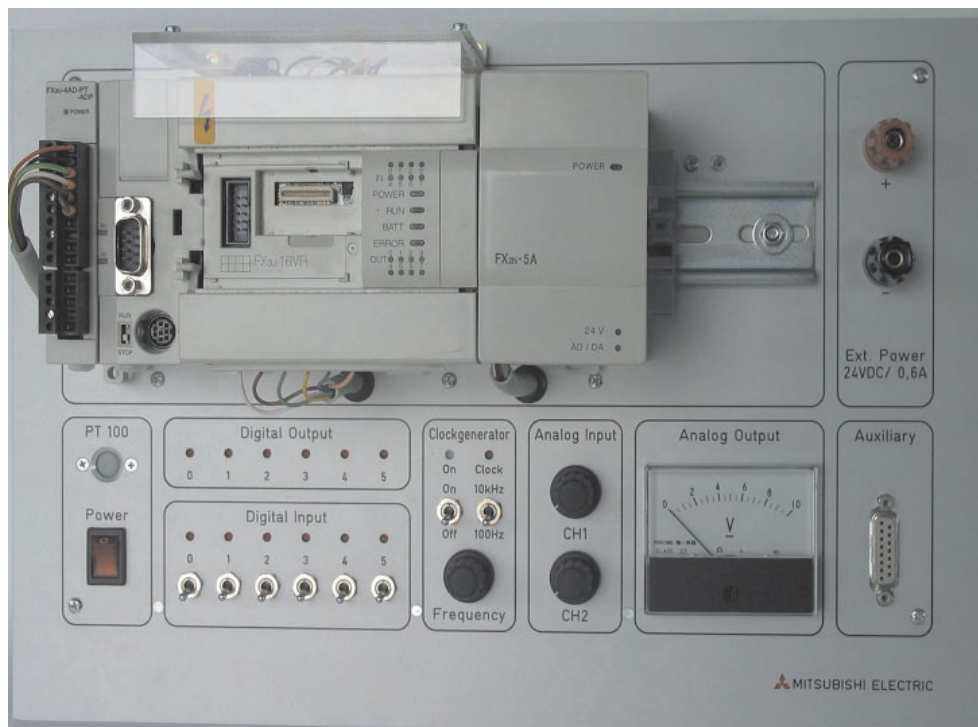
The course content has been selectively produced to provide an introduction into the functionality of the Mitsubishi range of the FX family, together with the GX Developer programming system. The first section deals with the PLC hardware configuration and operation, whilst the remainder covers the use of Mitsubishi's programming system, which is illustrated using worked examples.

It is assumed that student will have will have a sound working knowledge of the Microsoft Windows operating environment.

## 1.1 Modular PLC Training Hardware

There are various models of training rigs for Mitsubishi's FX family. Most exercises within this training manual are based around use of the facilities offered in these training systems. The examples used in these course notes assume the following configuration:

- 6 Digital input simulator switches: X0-X5
- Variable clock input (1–100 Hz and 0.1– 10 kHz): X7
- 6 Digital output LED indicators: Y0-Y5
- 1 Special function block FX2N-5A with 4 analog inputs and 1 analog output
- 1 Temperature acquisition special adapter FX3U-4AD-PT-ADP



Thus, adjustments according to other training simulators may be accommodated with appropriate address alterations to the example code provided this training document.



## 2 The Hardware

### 2.1 General Introduction to PLCs

#### 2.1.1 History & Development

Bedford Associates, founded by Richard Morley introduced the first Programmable Logic Controller in 1968. This PLC was known as the Modular Digital Controller from which the MODICON Company derived its name.

Programmable Logic Controllers were developed to provide a replacement for large relay based control panels. These systems were inflexible requiring major rewiring or replacement whenever the control sequence was to be changed.

The development of the Microprocessor from the mid 1970's have allowed Programmable Logic Controllers to take on more complex tasks and larger functions as the speed of the processor increased. It is now common for PLC's to provide the heart of the control functions within a system often integrated with SCADA (Supervisory Control And Data Acquisition), HMI (Human Machine Interfaces), Expert Systems and Graphical User Interfaces (GUI). The requirements of the PLC have expanded to providing control, data processing and management functionality.

#### 2.1.2 The initial specification for the PLC

- Easily programmed and reprogrammed in plant to enable its sequence of operations, to be altered.
- Easily maintained and repaired – preferably using 'plug-in' cards or modules.
- Able to withstand the rigorous Environmental, Mechanical and Electrical conditions, found in plant environments.
- Smaller than its relay and “discrete solid state” equivalents.
- Cost effective in comparison with “discrete solid state” and relay based systems.

#### 2.1.3 Comparison of PLC and Relay Systems

Characteristic	PLC	Relay
Price per function	Low	Low - If equivalent relay program uses more than 10 relays
Physical size	Very compact	Bulky
Operating speed	Fast	Slow
Electrical noise immunity	Good	Excellent
Construction	Easy to program	Wiring - time consuming
Advanced instructions	Yes	No
Changing the control sequence	Very simple	Very difficult – requires changes to wiring
Maintenance	Excellent PLC's rarely fail	Poor - relays require constant maintenance

### 2.1.4 Programming

#### Ladder Logic

PLC's had to be maintainable by technicians and electrical personnel. To support this, the programming language of Ladder Logic was developed. Ladder Logic is based on the relay and contact symbols technicians were used to through wiring diagrams of electrical control panels.

The documentation for early PLC Programs was either non existent or very poor, just providing simple addressing or basic comments, making large programs difficult to follow. This has been greatly improved with the development of PLC Programming packages such as Mitsubishi's Windows based, **GX Developer** (covered in detail later in this document).

Until recently there has been no formal programming standard for PLC's. The introduction of the **IEC 61131-3** Standard in 1998 provides a more formal approach to coding. Mitsubishi Electric has developed a programming package, "**GX IEC Developer**". This enables IEC compliant coding to be adopted.

### 2.1.5 Human Machine Interfaces

The early programmable logic controllers interfaced with the operator in much the same way as the relay control panel, via push-buttons and switches for control and lamps for indication.

The introduction of the Personal Computer (PC) in the 1980's allowed for the development of a computer based interface to the operator, these where initially via simple Supervisory Control And Data Acquisition (SCADA) systems and more recently via Dedicated Operator Control Panels, known as Human Machine Interfaces (HMI). It is now common place to see PLC's heavily integrated with these products to form user friendly control system solutions.

Mitsubishi offer a very wide range of HMI and SCADA products to suit a variety of operator interface applications.



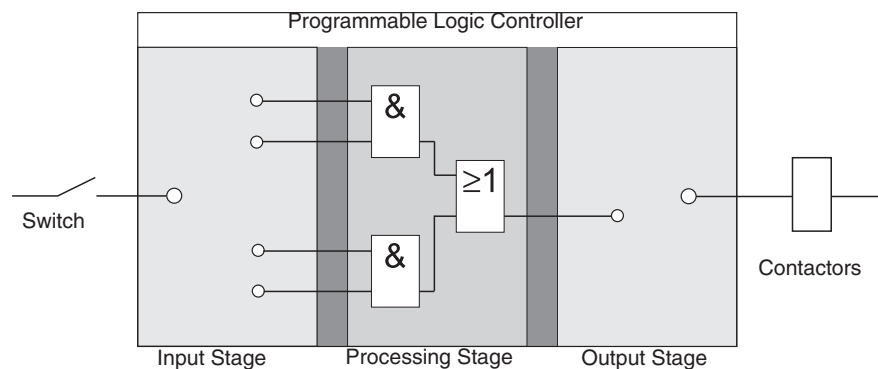


## 2.2 What is a PLC?

In contrast to conventional controllers with functions determined by their physical wiring the functions of programmable logic controllers or PLCs are defined by a program. PLCs also have to be connected to the outside world with cables, but the contents of their program memory can be changed at any time to adapt their programs to different control tasks.

Programmable logic controllers input data, process it and then output the results. This process is performed in three stages:

- an input stage,
  - a processing stage
- and
- an output stage



### The input stage

The input stage passes control signals from switches, buttons or sensors on to the processing stage.

The signals from these components are generated as part of the control process and are fed to the inputs as logical states. The input stage passes them on to the processing stage in a pre-processed format.

### The processing stage

In the processing stage the pre-processed signals from the input stage are processed and combined with the help of logical operations and other functions. The program memory of the processing stage is fully programmable. The processing sequence can be changed at any time by modifying or replacing the stored program.

### The output stage

The results of the processing of the input signals by the program are fed to the output stage where they control connected switchable elements such as contactors, signal lamps, solenoid valves and so on.

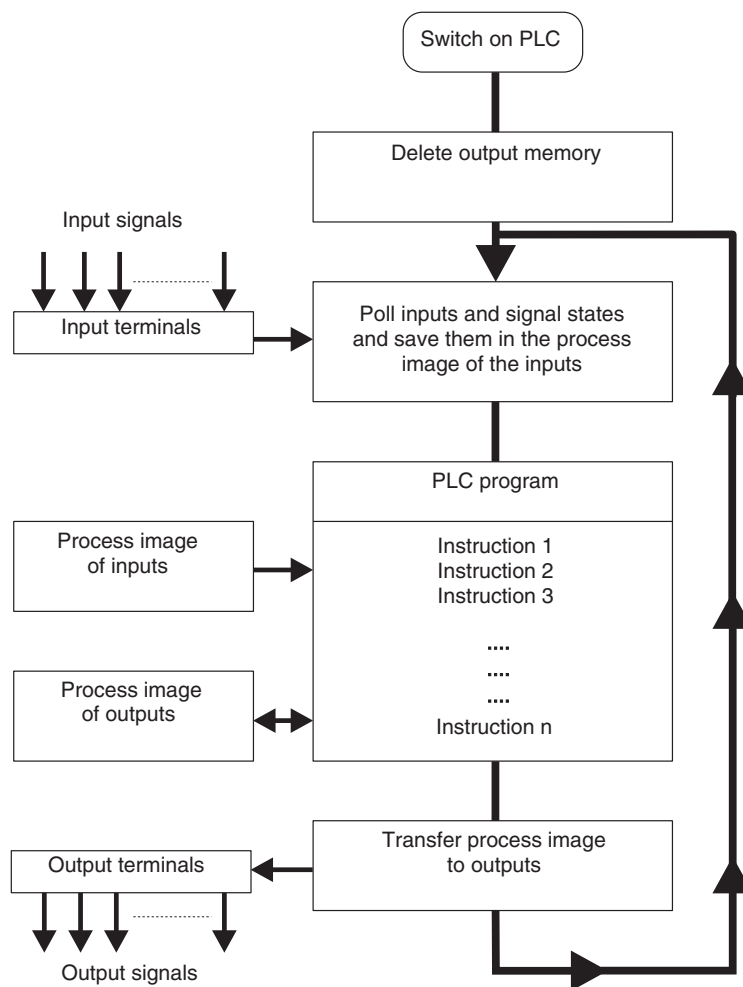
## 2.3 How PLCs Process Programs

A PLC performs its tasks by executing a program that is usually developed outside the controller and then transferred to the controller’s program memory. Before you start programming it is useful to have a basic understanding of how PLCs process these programs.

A PLC program consists of a sequence of instructions that control the functions of the controller. The PLC executes these control instructions sequentially, i.e. one after another. The entire program sequence is cyclical, which means that it is repeated in a continuous loop. The time required for one program repetition is referred to as the program cycle time or period.

### Process image processing

The program in the PLC is not executed directly on the inputs and outputs, but on a “process image” of the inputs and outputs:



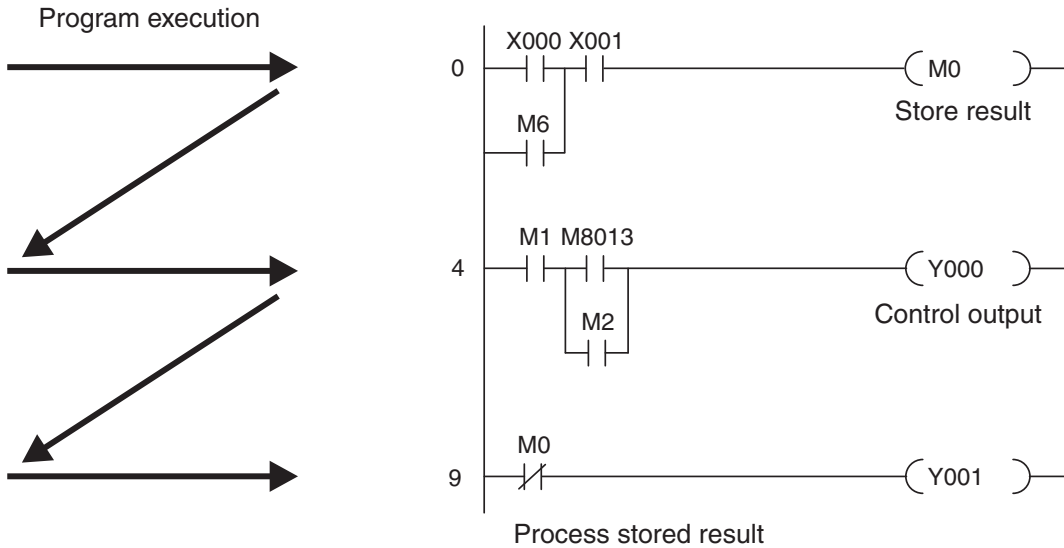
### Input process image

At the beginning of each program cycle the system polls the signal states of the inputs and stores them in a buffer, creating a “process image” of the inputs.

**Program execution**

After this the program is executed, during which the PLC accesses the stored states of the inputs in the process image. This means that any subsequent changes in the input states will not be registered until the **next** program cycle!

The program is executed from top to bottom, in the order in which the instructions were programmed. Results of individual programming steps are stored and can be used during the current program cycle.



**Output process image**

Results of logical operations that are relevant for the outputs are stored in an output buffer – the output process image. The output process image is stored in the output buffer until the buffer is rewritten. After the values have been written to the outputs the program cycle is repeated.

**Differences between signal processing in the PLC and in hard-wired controllers**

In hard-wired controllers the program is defined by the functional elements and their connections (the wiring). All control operations are performed simultaneously (parallel execution). Every change in an input signal state causes an instantaneous change in the corresponding output signal state.

In a PLC it is not possible to respond to changes in input signal states until the next program cycle after the change. Nowadays this disadvantage is largely compensated by very short program cycle periods. The duration of the program cycle period depends on the number and type of instructions executed.

## 2.4 The MELSEC FX Family

MELSEC means **MITSUBISHI ELECTRIC SEQUENCER**. The compact micro-controllers of the MELSEC FX series provide the foundation for building economical solutions for small to medium-sized control and positioning tasks requiring 10 to 256 integrated inputs and outputs in applications in industry and building services.

With the exception of the FX1S all the controllers of the FX series can be expanded to keep pace with the changes in the application and the user's growing requirements.

Network connections are also supported. This makes it possible for the controllers of the FX family to communicate with other PLCs and controller systems and HMIs (Human-Machine Interfaces and control panels). The PLC systems can be integrated both in MITSUBISHI networks as local stations and as slave stations in open networks like PROFIBUS/DP.

In addition to this you can also build multi-drop and peer-to-peer networks with the controllers of the MELSEC FX family.

The FX1N, FX2N, FX3G, FX3UC and FX3U have modular expansion capabilities, making them the right choice for complex applications and tasks requiring special functions like analog-digital and digital-analog conversion and network capabilities.

All the controllers in the series are part of the larger MELSEC FX family and are fully compatible with one another.

Specifications	FX1S	FX1N	FX2N	FX2NC	FX3G	FX3U	FX3UC
Max integrated I/O points	30	60	128	96	60	128	96
Expansion capability (max. possible I/Os)	34	132	256	256	256	384	384
Program memory (steps)	2000	8000	16000	16000	32000	64000	64000
Cycle time per logical instruction ( $\mu$ s)	0.55–0.7	0.55–0.7	0.08	0.08	0.21/0.42	0.065	0.065
No. of instructions (standard / step ladder / special function)	27 / 2 / 85	27 / 2 / 89	27 / 2 / 107	27 / 2 / 107	29 / 2 / 123	27 / 2 / 209	29 / 2 / 209
Max. special function modules connectable	—	2	8	4	4 <sup>①</sup> / 8 <sup>②</sup>	10 <sup>①</sup> / 8 <sup>②</sup>	6 <sup>①</sup> / 8 <sup>②</sup>

① Connectable to the left side

② Connectable to the right side

## 2.5 Selecting the Right Controller

The base units of the MELSEC FX family are available in a number of different versions with different power supply options and output technologies. You can choose between units designed for power supplies of 100–240 V AC, 24 V DC or 12–24 V DC, and between relay and transistor outputs.

Series	I/Os	Type	No. of inputs	No. of outputs	Power supply	Output type
FX1S	10	FX1S-10 M□-□□	6	4	24 V DC or 100–240 V AC	Transistor or relay
	14	FX1S-14 M□-□□	8	6		
	20	FX1S-20 M□-□□	12	8		
	30	FX1S-30 M□-□□	16	14		
FX1N	14	FX1N-14 M□-□□	8	6	12–24 V DC or 100–240 V AC	Transistor or relay
	24	FX1N-24 M□-□□	14	10		
	40	FX1N-40 M□-□□	24	16		
	60	FX1N-60 M□-□□	36	24		
FX2N	16	FX2N-16 M□-□□	8	8	24 V DC or 100–240 V AC	Transistor or relay
	32	FX2N-32 M□-□□	16	16		
	48	FX2N-48 M□-□□	24	24		
	64	FX2N-64 M□-□□	32	32		
	80	FX2N-80 M□-□□	40	40		
FX2NC	16	FX2NC-16 M□-□□	8	8	24 V DC	Transistor or relay
	32	FX2NC-32 M□-□□	16	16		
	64	FX2NC-64 M□-□□	32	32		
	96	FX2NC-96 M□-□□	48	48		
FX3G	14	FX3G-14M□/□□□	8	6	24 V DC or 100–240 V AC	Transistor or relay
	24	FX3G-24M□/□□□	14	10		
	40	FX3G-40M□/□□□	24	16		
	60	FX3G-60M□/□□□	36	24		
FX3U	16	FX3U-16 M□-□□	8	8	24 V DC or 100–240 V AC	Transistor or relay
	32	FX3U-32 M□-□□	16	16		
	48	FX3U-48 M□-□□	24	24		
	64	FX3U-64 M□-□□	32	32		
	80	FX3U-80 M□-□□	40	40		
	128	FX3U-128 M□-□□	64	64	100–240 V AC	Transistor or relay
FX3UC	16	FX3UC-16M□/□□□	8	8	24 V DC	Transistor
	32	FX3UC-32M□/□□□	16	16		
	64	FX3UC-64M□/□□□	32	32		
	96	FX3UC-96M□/□□□	48	48		

Here are some considerations that should be taken into account when configuring a system:

- Power supply requirements

Supply voltage: 24 V DC or 100–240 V AC

- Input/Output requirements

- How many signals (external switch contacts, buttons and sensors) do you need to input?
- What types of functions do you need to switch, and how many of them are there?
- How high are the loads that the outputs need to switch?

Choose relay outputs for switching high loads and transistor outputs for switching fast, trigger-free switching operations.

- **Special Function Modules**

- Number of modules in system
- External power supply requirements

## 2.6 Controller Design

All the controllers in the series have the same basic design. The main functional elements and assemblies are described in the glossary in the appendix.

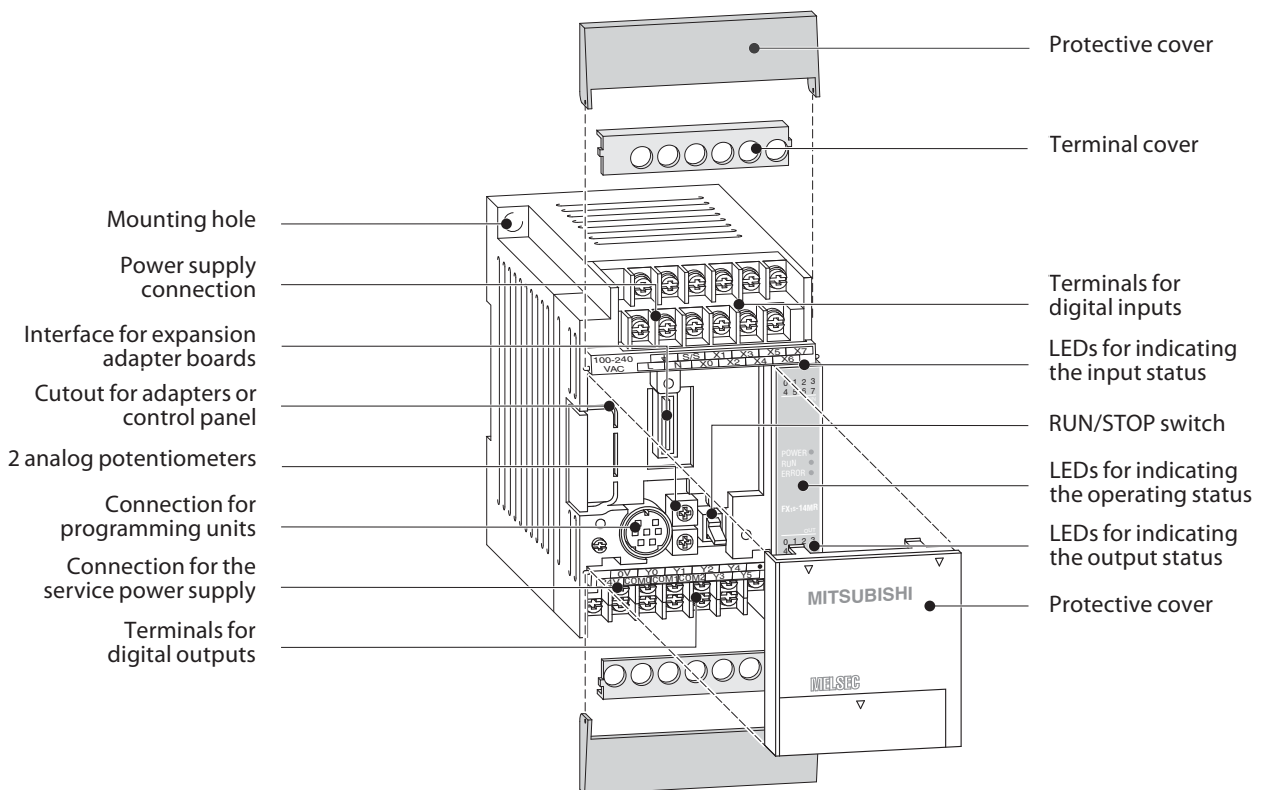
### 2.6.1 Input and output circuits

The **input circuits** use floating inputs. They are electrically isolated from the other circuits of the PLC with optical couplers. The **output circuits** use either relay or transistor output technology. The transistor outputs are also electrically isolated from the other PLC circuits with optical couplers.

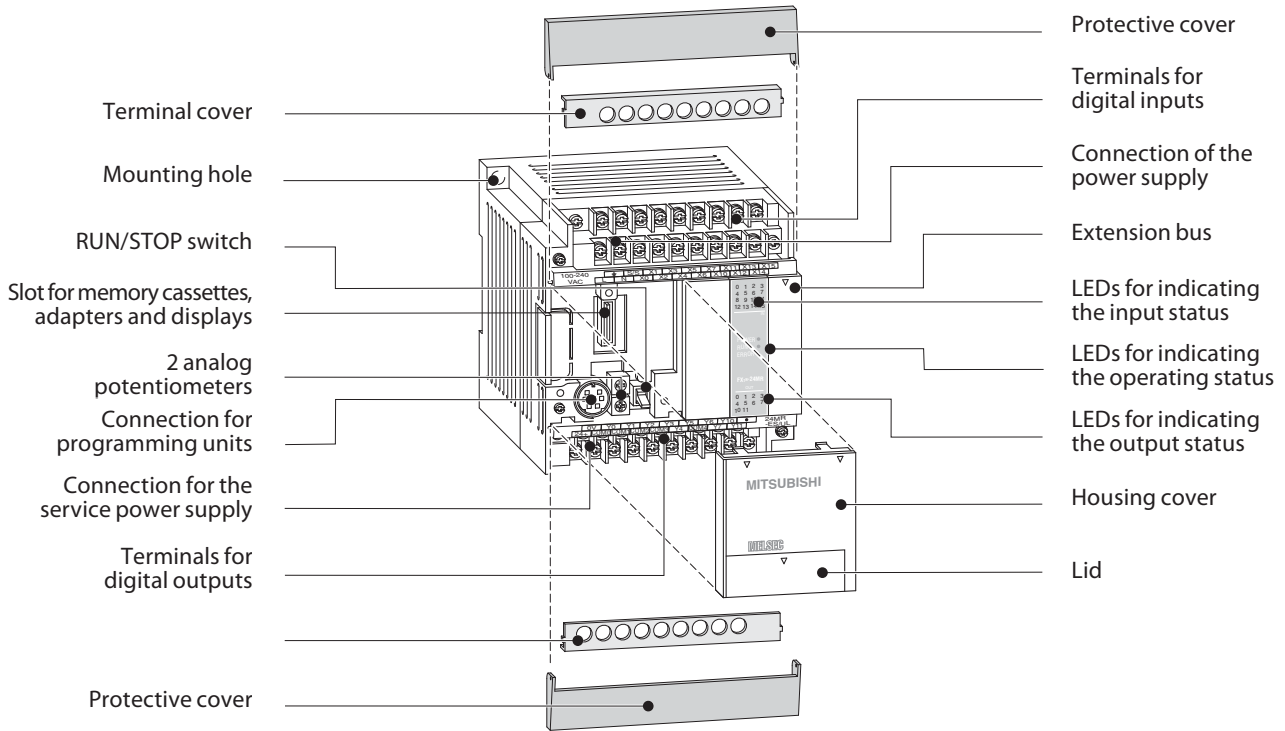
The switching voltage at all the digital inputs must have a certain value (e.g. 24 V DC). This voltage can be taken from the PLC's integrated power supply unit. If the switching voltage at the inputs is less than the rated value (e.g. <24 V DC) then the input will not be processed.

The maximum output currents are 2 A on 250 V three-phase AC and non-reactive loads with relay outputs and 0.5 A on 24 V DC and non-reactive loads.

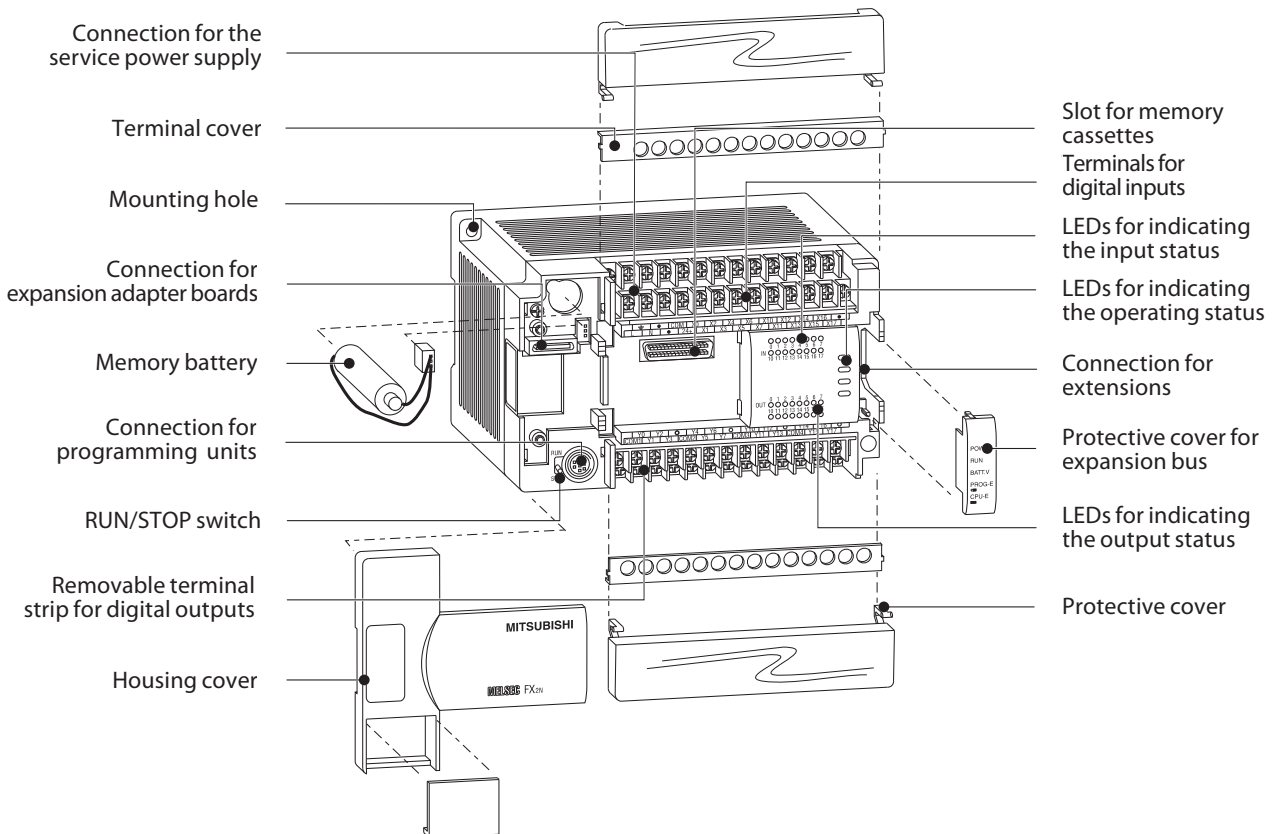
### 2.6.2 Layout of the MELSEC FX1s base units



### 2.6.3 Layout of the MELSEC FX1N base units

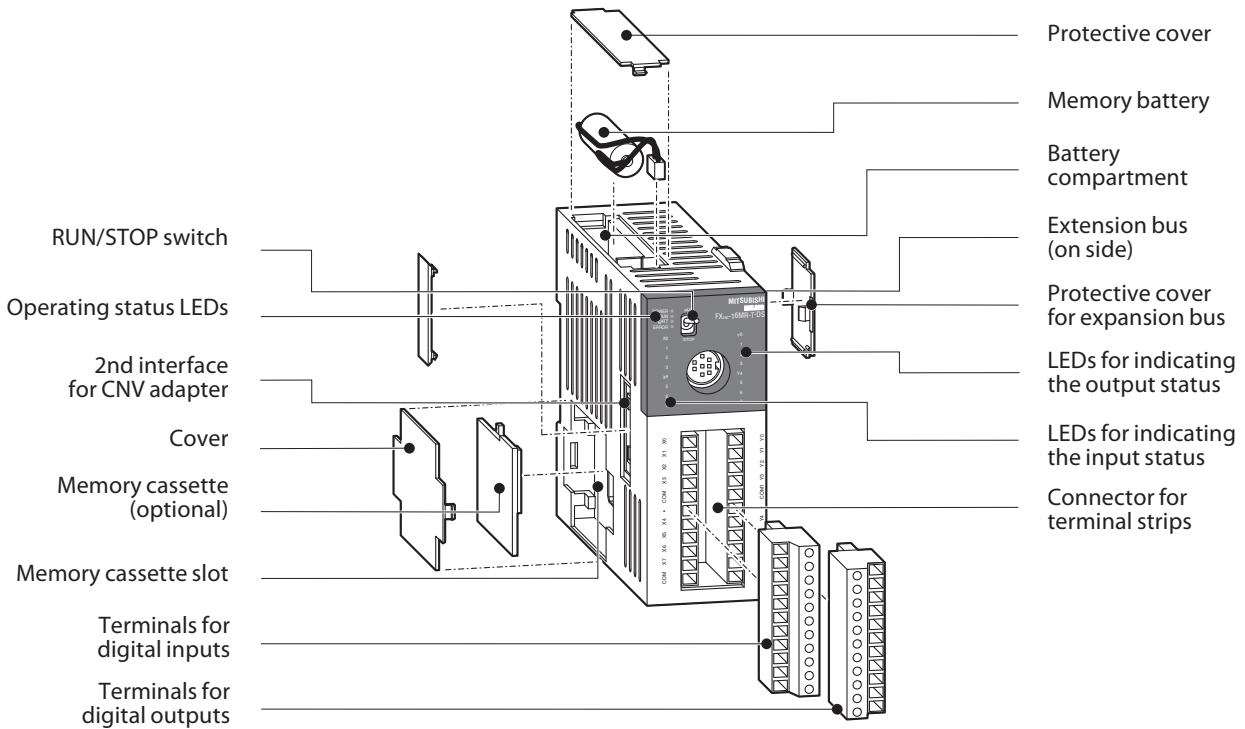


### 2.6.4 Layout of the MELSEC FX2N base units

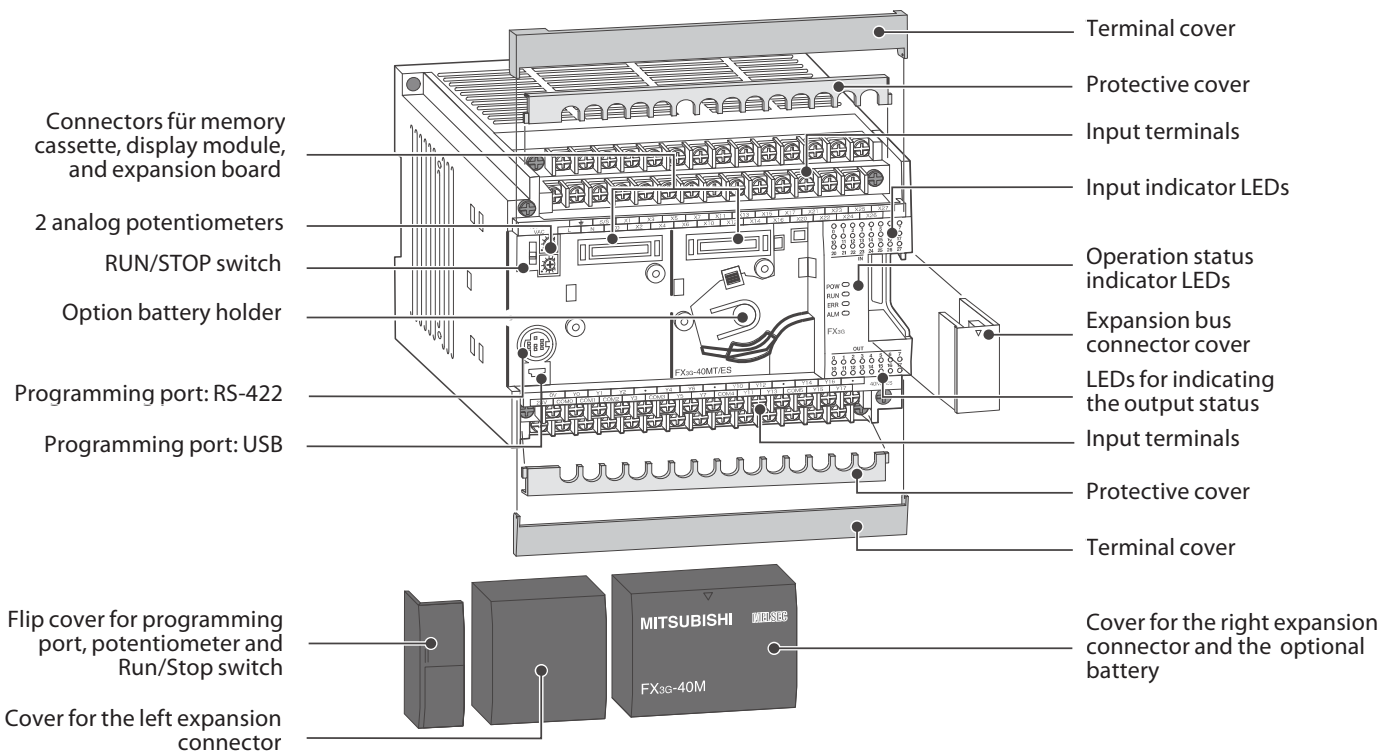




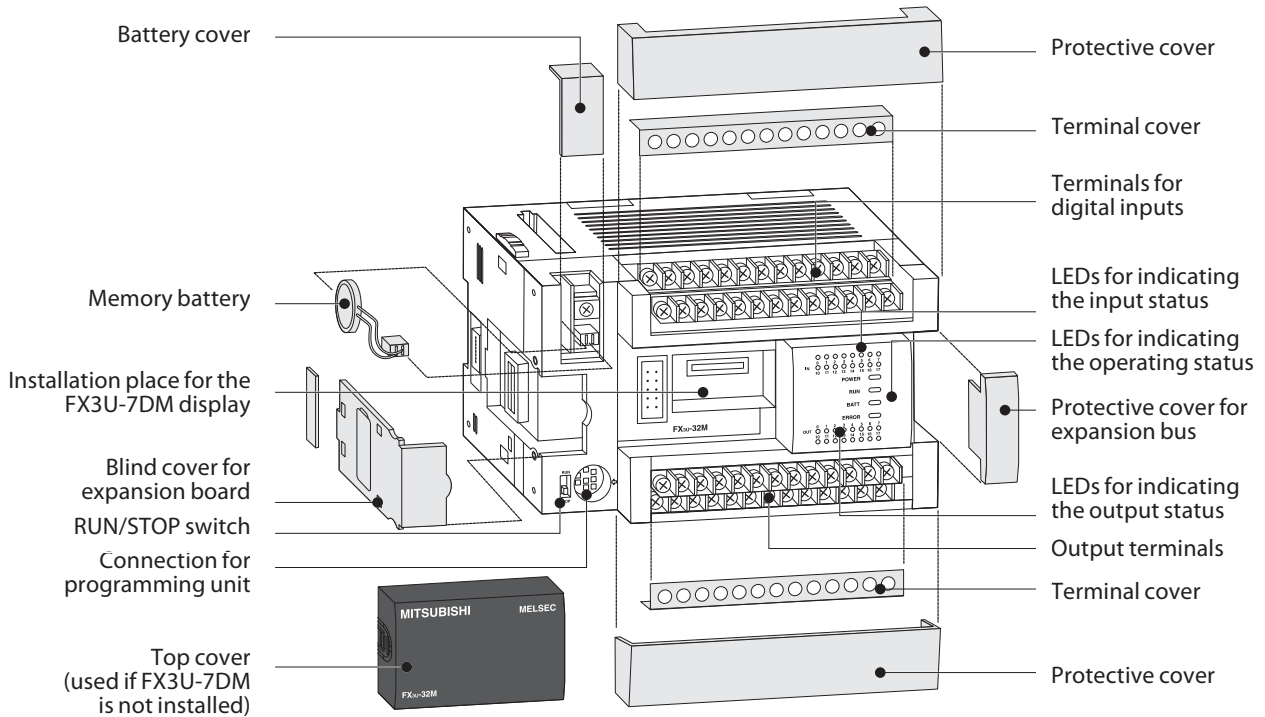
### 2.6.5 Layout of the MELSEC FX2NC base units



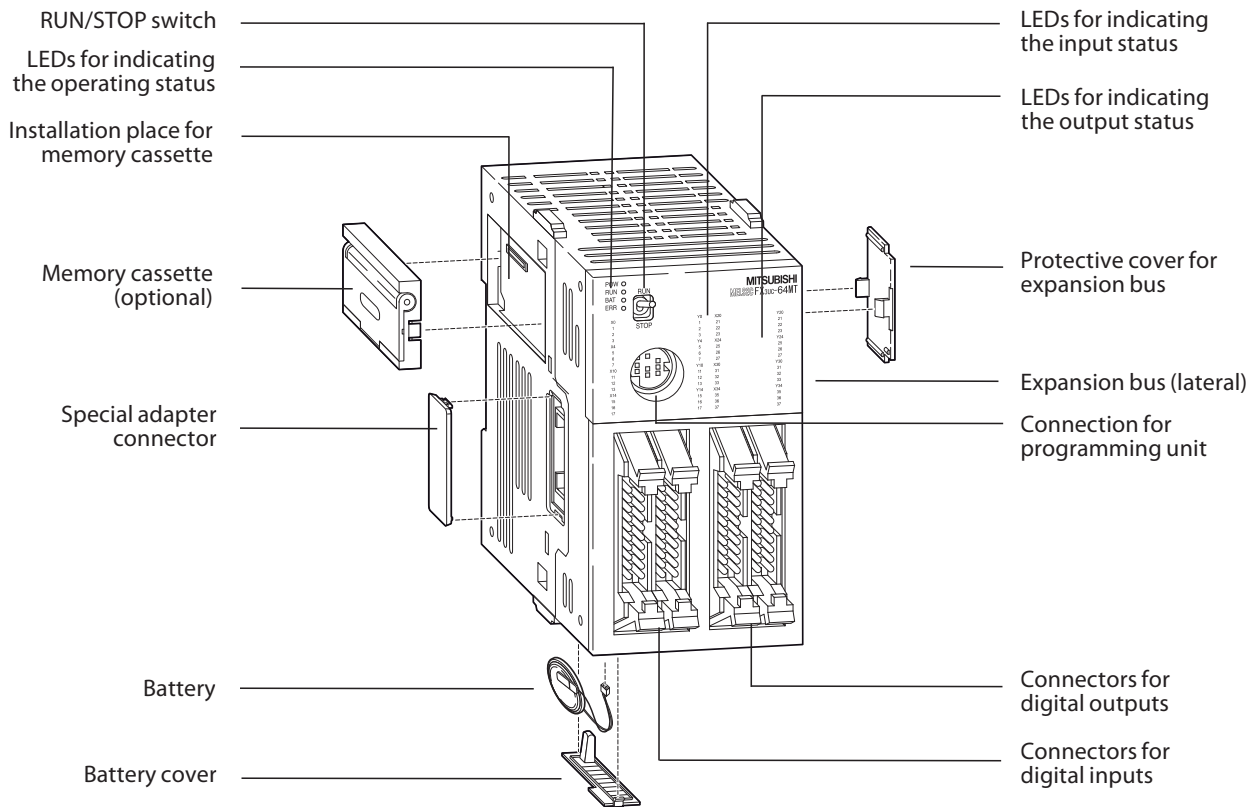
### 2.6.6 Layout of the MELSEC FX3G base units



### 2.6.7 Layout of the MELSEC FX3U base units



### 2.6.8 Layout of the MELSEC FX3UC base units



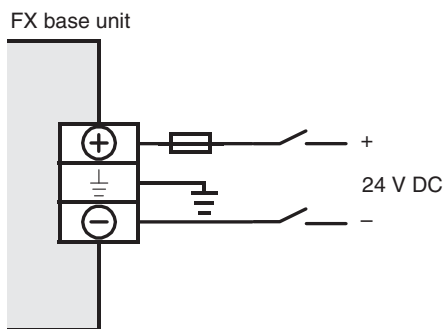
## 2.7 Wiring

### 2.7.1 Power Supply

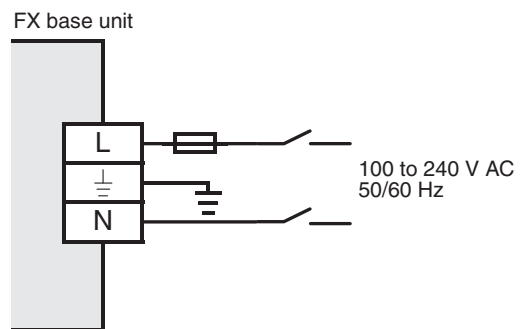
#### Power Supply Specifications

Specification	Units for DC Power Supply		Units for AC Power Supply
Rated voltage	12 to 24 V DC	24 V DC	100 to 240 V AC
Voltage range	10.2 to 26.4 V DC	20.4 to 26.4 V DC	85 to 264 V AC
Allowable momentary power failure time	5 ms		20 ms

#### Connection of units with DC power supply



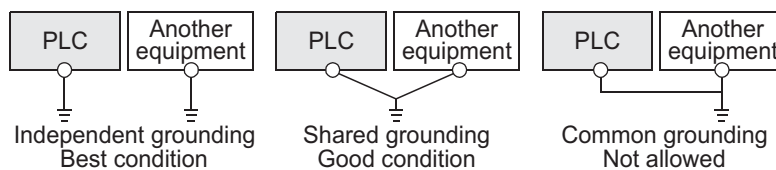
#### Connection of units with AC power supply



#### Grounding

The PLC should be grounded.

- The grounding resistance should be 100 Ω or less.
- The grounding point should be close to the PLC. Keep the grounding wires as short as possible.
- Independent grounding should be performed for best results. When independent grounding is not performed, perform "shared grounding" of the following figure.

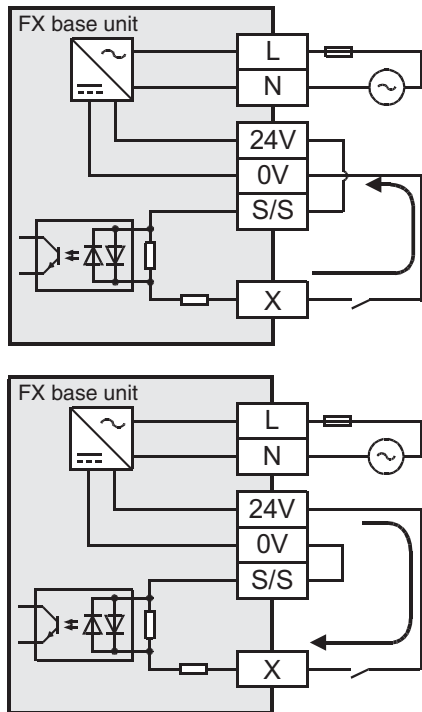


- The ground wire size should be at least 2 mm<sup>2</sup>.

## 2.7.2 Wiring of Inputs

### Connecting sink or source devices

The base units of the FX family series can be used with sink or source switching devices. The decision is made by the different connections of the "S/S" terminal.



In the case of the **sink input type**, the S/S terminal is connected to the 24V terminal of the service power supply or, when a DC powered base unit is used, to the positive pole of the power supply.

Sink input means that a contact wired to the input (X) or a sensor with NPN open collector transistor output connects the input of the PLC with the **negative** pole of a power supply.

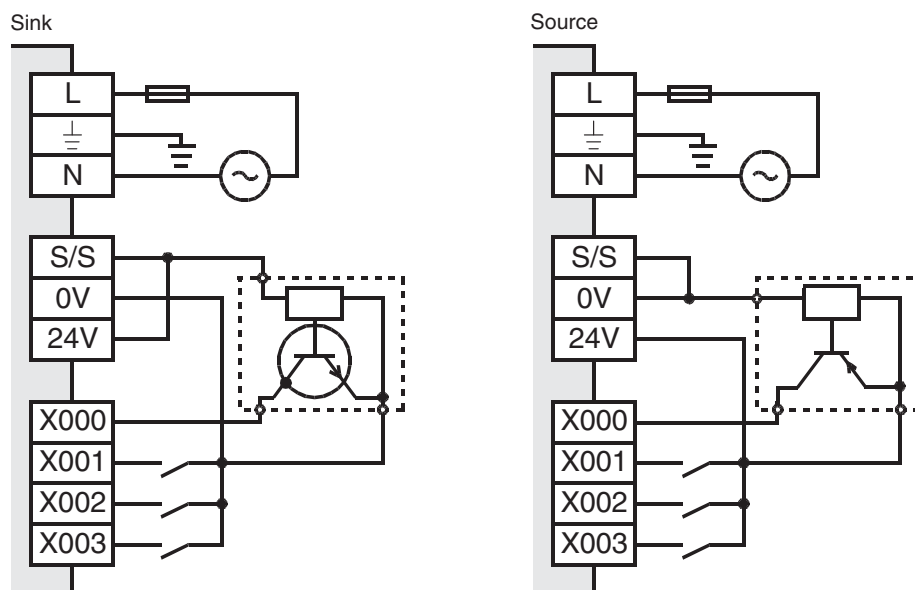
In the case of the **source input type**, the S/S terminal is connected to the 0V terminal of the service power supply or, when a DC powered base unit is used, to the negative pole of the power supply.

Source input means that a contact wired to the input (X) or a sensor with PNP open collector transistor output connects the input of the PLC with the **positive** pole of a power supply.

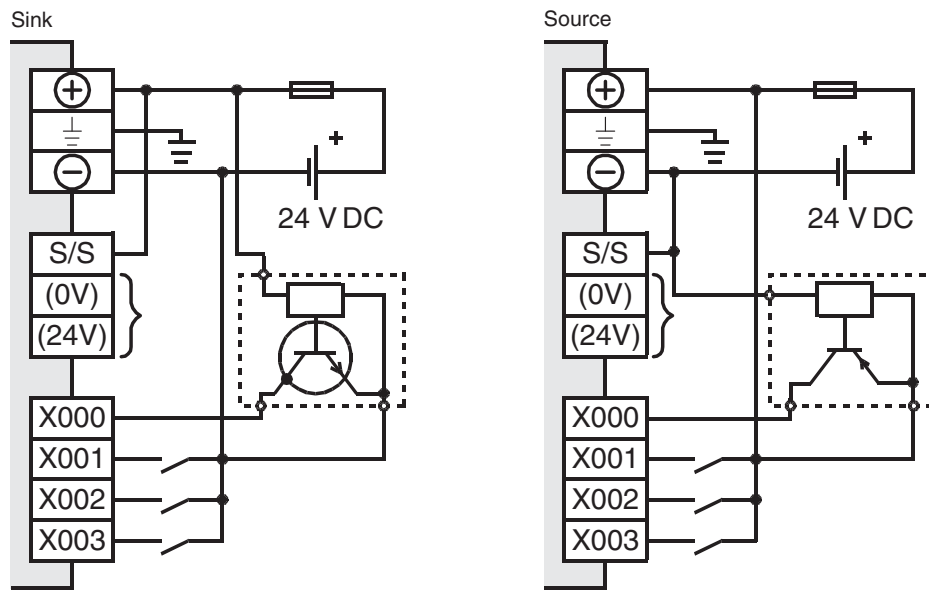
All inputs of a base unit or an extension unit can be either used as sink or source inputs, but it is not possible to mix sink and source inputs in one unit. Separate units in one PLC however can be set as sink or source inputs types, since the base unit and input/output powered extension units are individually set to sink or source input mode.

### Examples for input types

AC powered base units



DC powered base units



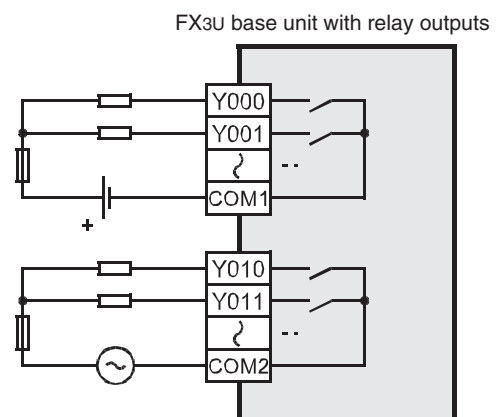
2.7.3 Wiring of Outputs

In case of base units with only few outputs (e. g. FX3G-14M□ or FX3U-16M□) each output can be connected separately. In case of the base units with more outputs, the outputs are pooled into groups of 2, 3, 4, 8 or 16 outputs. Each group has a common contact for the load voltage. These terminals are marked "COM□" for base units with relays outputs or transistor outputs of the sink type and "+V□" for base units with source transistor outputs. "□" stands for the number of the output group e. g. "COM1".

Because the outputs groups are isolated against each other, one base unit can switch several voltages with different potentials. base units with relay outputs can even switch AC and DC voltages.

The first group of outputs is used to switch a DC voltage.

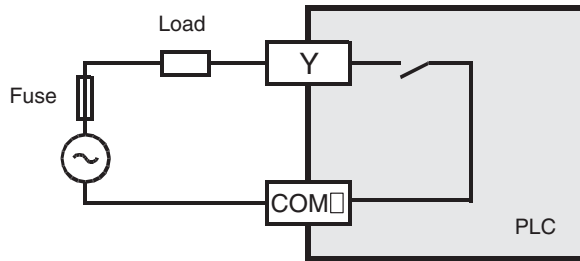
The second group of relays controls AC powered loads.



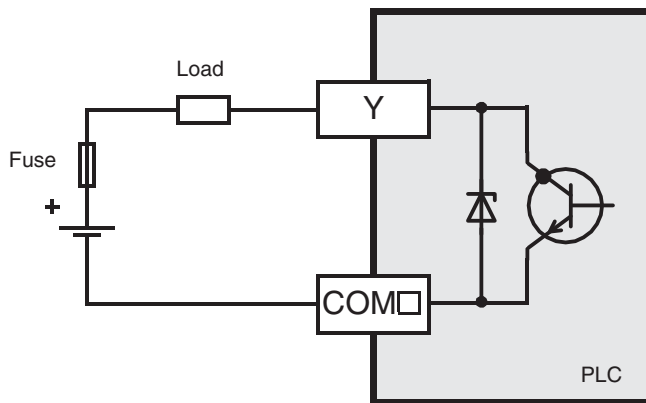
The selection of sink and source output type is done by the selection of a correspondent base unit. Both types are available with DC or AC power supply. The output type is given in the model designation code: base units with the code "MT/□S" provide transistor sink type outputs (e. g. FX3U-16MT/ES) while base units with the code "MT/□SS" provide transistor source type outputs (e. g. FX3U-16MT/ESS).

### Examples of output wiring

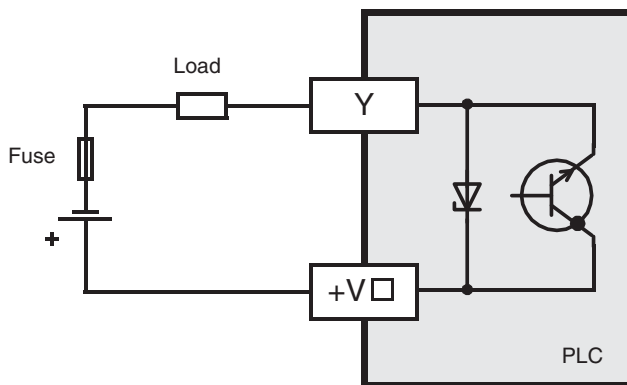
Relay output



Transistor output (sink)



Transistor output (source)

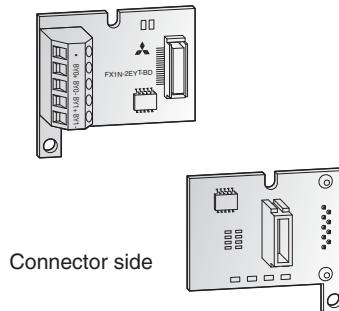


## 2.8 Extending the Range of Digital Inputs/Outputs

For the MELSEC FX family of PLCs several ways and means are available to provide a base unit with additional inputs and outputs.

### 2.8.1 Extension Boards

FX1N-2EYT-BD with two digital outputs



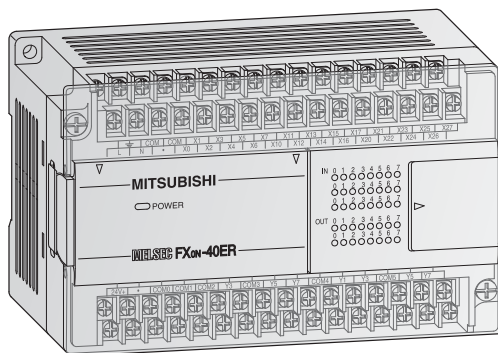
For a small number of I/O (2 to 4) an extension adapter board can be installed directly in a FX1S or FX1N base unit. Extension boards therefore do not require any additional installation space.

The state of the additional input and outputs is reflected in special relays in the PLC (see section A.1.5). In the program these relays are used instead of X and Y devices.

Designation	Number of I/O			Output type	Power supply	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
	Total	No. of inputs	No. of outputs							
FX1N-4EX-BD	4	4	—	—	From base unit	●	●	○	○	○
FX1N-2EYT-BD	2	—	2	Transistor						

- : The extension board can be used with a base unit of this series.
- : The extension board cannot be used with this series.

### 2.8.2 Compact Extension Units



The powered compact input/output extension units have their own power supply. The integrated service power supply (24 V DC) of AC powered extension units can be used for the supply of external devices.

It is possible to choose between relay and transistor (source) output type.

#### Compact Extension Units of the FX0N Series

Designation	Number of I/O			Output type	Power supply	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
	Total	No. of inputs	No. of outputs							
FX0N-40ER/ES-UL	40	24	16	Relay	100–240 V AC	○	●	○	○	○
FX0N-40ER/DS	40	24	16	Relay	24 V DC					
FX0N-40ET/DSS	40	24	16	Transistor						

- : The extension unit can be used with a base unit of this series.
- : The extension unit cannot be used with this series.

### Compact Extension Units of the FX2N Series

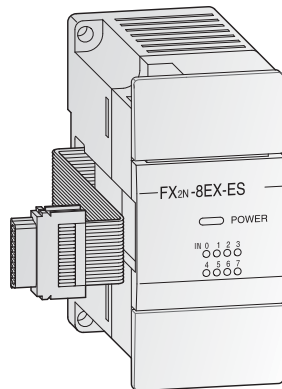
Designation	Number of I/O			Output type	Power supply	FX1S	FX1N	FX2N FX2NC	FX3U	FX3U FX3UC
	Total	No. of inputs	No. of outputs							
FX2N-32ER-ES/UL	32	16	16	Relay	100–240 V AC	○	●	●*	●	●*
FX2N-32ET-ESS/UL	32	16	16	Transistor						
FX2N-48ER-ES/UL	48	16	16	Relay						
FX2N-48ET-ESS/UL	48	24	24	Transistor						
FX2N-48ER-DS	48	24	24	Relay	24 V DC					
FX2N-48ET-DSS	48	24	24	Transistor						

● : The extension unit can be used with a base unit of this series.

○ : The extension unit cannot be used with this series.

\* These extension units cannot be connected to a base unit of the FX2NC or FX3UC series.

### 2.8.3 Modular Extension Blocks



Modular extension blocks have no build-in power supply but very compact dimensions. The FX2N series modular extension blocks are available with 8 or 16 input/output points. It is possible to choose between relay and transistor (source) output type.

The FX2NC series extension blocks are available with 16 or 32 integrated I/O with selectable relay or transistor 16-output models (source type).

Designation	Number of I/O			Output type	Power supply	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
	Total	No. of inputs	No. of outputs							
FX2N-8ER-ES/UL	16 <sup>①</sup>	4	4	Relay	100–240 V AC	○	●	●		●
FX2N-8EX-ES/UL	8	8	—	—						
FX2N-16EX-ES/UL	16	16	—	—						
FX2N-8EYR-ES/UL	8	—	8	Relay						
FX2N-8EYT-ESS/UL	8	—	8	Transistor	24 V DC					
FX2N-16EYR-ES/UL	16	—	16	Relay						
FX2N-16EYT-ESS/UL	16	—	16	Transistor						
FX2NC-16EX-DS	16	16	—	—	From base unit	○	○	● <sup>②</sup>	○	● <sup>②</sup>
FX2NC-16EX-T-DS	16	16	—	—						
FX2NC-32EX-DS	32	32	—	—						
FX2NC-16EYT-DSS	16	—	16	Transistor	From base unit					
FX2NC-16EYR-T-DS	16	—	16	Relay						
FX2NC-32EYT-DSS	32	—	32	Transistor						

① The extension block FX2N-8ER-ES/UL occupies 16 input/output points of the PLC. Four inputs and four outputs are occupied but cannot be used.

② The FX2NC series extension blocks can only be connected to a base unit of the FX2NC or FX3UC series.

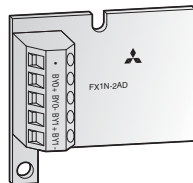


## 2.9 Extending for Special Functions

A variety of hardware for special functions are available for the MELSEC FX family.

### Adapter Boards

Adapter boards are small circuit boards that are installed directly in the FX1S, FX1N or FX3G controllers, which means that they don't take up any extra space in the switchgear cabinet.

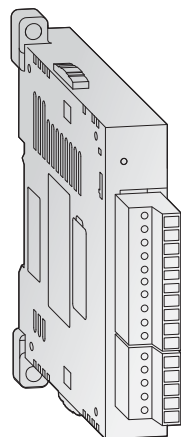


In the case of analog adapter boards, the digital values generated from the signals coming from the analog input adapter's two input channels are written directly to special registers, which makes it particularly easy to process them.

The output value for the analog output adapter is written by the program also to a special register and then converted by the adapter and sent to the output.

### Special Adapter

Special adapters can only be connected on the left side of a base unit of the MELSEC FX3G, FX3U and FX3UC series.

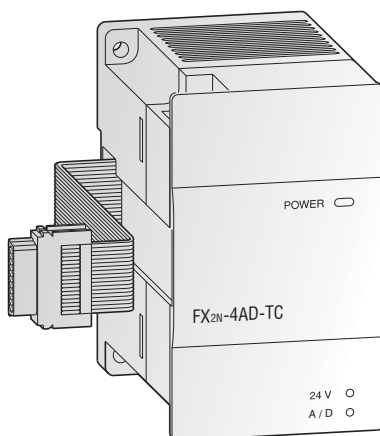


You can install one analog special adapter to a FX3G base unit with 14 or 24 inputs and outputs. Up to two analog special adapter can be mounted to a FX3G base unit with 40 or 60 inputs and outputs. To a FX3U or FX3UC base unit up to four analog special adapters can be connected.

Special adapters do not use any input or output points in the base unit. They communicate directly with the base unit via special relays and registers. Because of this, no instructions for communication with special function modules are needed in the program.

### Special function modules

Up to eight special function modules can be connected on the right side of a single base unit of the MELSEC FX family.



In addition to analog modules the available special function modules include communication modules, positioning modules and other types. Each special function module occupies eight input points and eight output points in the base unit.

Communication between the special function module and the PLC base unit is carried out via the memory buffer of the special function module with the help of FROM and TO instructions.

### 2.9.1 Analog Modules

Without additional modules the base units of the MELSEC FX family can only process digital input and output signals (i.e. ON/OFF data). Additional analog modules are thus required for inputting and outputting analog signals.

Modul Type	Designation	No. of channels	Range	Resolution	FX1s	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC	
Analog Input Modules	Adapter Board	FX1N-2AD-BD	2	Voltage: 0 V to 10 V DC	2.5 mV (12 Bit)	●	●	○	○	○
				Current: 4 mA to 20 mA DC	8 μA (11 Bit)					
	Adapter Board	FX3G-2AD-BD	2	Voltage: 0 V to 10 V DC	2.5 mV (12 Bit)	○	○	○	●	○
				Current: 4 mA to 20 mA DC	8 μA (11 Bit)					
	Special Adapter	FX3U-4AD-ADP	4	Voltage: 0 V to 10 V DC	2.5 mV (12 Bit)	○	○	○	●	●
				Current: 4 mA to 20 mA DC	10 μA (11 Bit)					
	Special Function Modules	FX2N-2AD	2	Voltage: 0 V to 5 V DC 0 V to 10 V DC	2.5 mV (12 Bit)	○	●	●	●	●
				Current: 4 mA to 20 mA DC	4 μA (12 Bit)					
		FX2N-4AD	4	Voltage: -10 V to 10 V DC	5 mV (with sign, 12 bits)	○	●	●	●	●
				Current: 4 mA to 20 mA DC -20 mA to 20 mA DC	10 μA (with sign, 11 bits)					
		FX2N-8AD <sup>①</sup>	8	Voltage: -10 V to 10 V DC	0.63 mV (with sign, 15 bits)	○	●	●	●	●
				Current: 4 mA to 20 mA DC -20 mA to 20 mA DC	2.50 μA (with sign, 14 bits)					
FX3U-4AD FX3UC-4AD	4	Voltage: -10 V to 10 V DC	0.32 mV (with sign, 16 bits)	○	○	○	● <sup>②</sup>	● <sup>②</sup>		
		Current: 4 mA to 20 mA DC -20 mA to 20 mA DC	1.25 μA (with sign, 15 bits)							
Analog Output Modules	Adapter Board	FX1N-1DA-BD	1	Voltage: 0 V to 10 V DC	2.5 mV (12 Bit)	●	●	○	○	○
				Current: 4 mA to 20 mA DC	8 μA (11 Bit)					
	Adapter Board	FX3G-1DA-BD	1	Voltage: 0 V to 10 V DC	2.5 mV (12 Bit)	○	○	○	●	○
				Current: 4 mA to 20 mA DC	8 μA (11 Bit)					
	Special Adapter	FX3U-4DA-ADP	4	Voltage: 0 V to 10 V DC	2.5 mV (12 Bit)	○	○	○	●	●
				Current: 4 mA to 20 mA DC	4 μA (12 Bit)					
	Special Function Modules	FX2N-2DA	2	Voltage: 0 V to 5 V DC 0 V to 10 V DC	2.5 mV (12 Bit)	○	●	●	●	●
				Current: 4 mA to 20 mA DC	4 μA (12 Bit)					

① The special function block FX2N-8AD is able to measure voltage, current and temperature.

② The FX3UC-4AD can be connected to base units of the FX3UC series only.

Modul Type		Designation	No. of channels	Range	Resolution	FX1s	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
Analog Output Modules	Special Function Modules	FX2N-4DA	4	Voltage: -10 V to 10 V DC	5 mV (with sign, 12 bits)	○	●	●	●	●
				Current: 0 mA to 20 mA DC 4 mA to 20 mA DC	20 μA (10 Bit)					
	FX3U-4DA	4	Voltage: -10 V to 10 V DC	0.32 mV (with sign, 16 bits)	○	○	○	●	●	
			Current: 0 mA to 20 mA DC 4 mA to 20 mA DC	0.63 μA (15 Bit)						
Combined Analog Input & Output Modules	Special Function Modules	FX0N-3A	2 inputs	Voltage: 0 V to 5 V DC 0 V to 10 V DC	40 mV (8 Bit)	○	●	●	○	● <sup>①</sup>
				Current: 4 mA to 20 mA DC	64 μA (8 Bit)					
			1 output	Voltage: 0 V to 5 V DC 0 V to 10 V DC	40 mV (8 Bit)					
				Current: 4 mA to 20 mA DC	64 μA (8 Bit)					
	FX2N-5A	4 inputs	Voltage: -100 mV to 100 mV DC -10 V to 10 V DC	50 μV (with sign, 12 bits) 0.312 mV (with sign, 16 bits)	○	●	●	●	●	
			Current: 4 mA to 20 mA DC -20 mA to 20 mA DC	10 μA/1,25 μA (with sign, 15 bits)						
		1 output	Voltage: -10 V to 10 V DC	5 mV (with sign, 12 bits)						
			Current: 0 mA to 20 mA DC	20 μA (10 Bit)						
	Special Adapter	FX3U-3A-ADP	2 inputs	Voltage: 0 V to 10 V DC	2,5 mV (12 Bit)	○	○	○	●	●
				Current: 4 mA to 20 mA DC	5 μA (12 Bit)					
1 output			Voltage: 0 V to 10 V DC	2,5 mV (12 Bit)						
			Current: 4 mA to 20 mA DC	4 μA (12 Bit)						
Temperature Acquisition Modules	Special Adapter	FX3U-4AD-PT-ADP	4	Pt100 resistance thermometer: -50 °C to 250 °C	0.1 °C	○	○	○	●	●
		FX3U-4AD-PTW-ADP	4	Pt100 resistance thermometer: -100 °C to 600 °C	0.2 °C to 0.3 °C	○	○	○	●	●
		FX3U-4AD-PNK-ADP	4	Pt100 resistance thermometer: -50 °C to 250 °C	0.1 °C	○	○	○	●	●
				Ni1000 resistance thermometer: -40 °C to 110 °C	0.1 °C	○	○	○	●	●
		FX3U-4AD-TC-ADP	4	Thermocouple type K: -100 °C to 1000 °C	0.4 °C	○	○	○	●	●
				Thermocouple type J: -100 °C to 600 °C	0.3 °C					

① A FX0N-3A can not be connected to base units of the FX3UC series.

Modul Type		Designation	No. of channels	Range	Resolution	FX1s	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
Temperature Acquisition Modules	Special Function Modules	FX2N-8AD <sup>①</sup>	8	Thermocouple type K: -100 °C to 1200 °C	0.1 °C	○	●	●	●	●
				Thermocouple type J: -100 °C to 600 °C	0.1 °C					
				Thermocouple type T: -100 °C to 350 °C	0.1 °C					
		FX2N-4AD-PT	4	Pt100 resistance thermometer: -100 °C to 600 °C	0.2 °C to 0.3 °C	○	●	●	●	●
		FX2N-4AD-TC	4	Thermocouple type K: -100 °C to 1200 °C	0.4 °C	○	●	●	●	●
Thermocouple type J: -100 °C to 600 °C	0.3 °C									
Temperature Control Modules (Special Function Modules)		FX2N-2LC	2	For example with a thermocouple type K: -100 °C to 1300 °C  Pt100 resistance thermometer: -200 °C to 600 °C	0.1 °C or 1 °C (depends on temperature probe used)	○	●	●	●	●
		FX3U-4LC	4			○	○	○	●	●

\* The special function block FX2N-8AD is able to measure voltage, current and temperature.

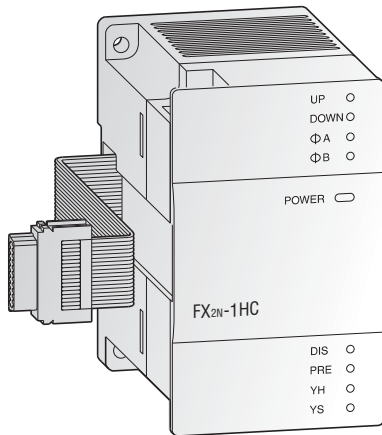
● The adapter board, special adapter or special function module can be used with a base unit or expansion unit of this series.

○ The adapter board, special adapter or special function module cannot be used with this series.

## 2.9.2 High-Speed Counter Modules and Adapters

### FX2N-1HC, FX2NC-1HC and FX3U-2HC

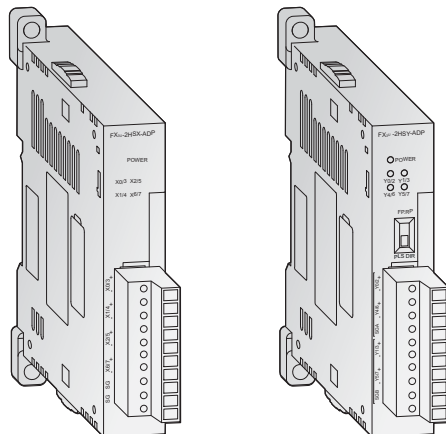
In addition to the internal high-speed MELSEC FX counters, the high-speed counter modules FX2N-1HC, FX2NC-1HC and FX3U-2HC provide the user with an external counter. They count 1- or 2-phase pulses up to a frequency of 50 kHz resp. 200 kHz for the FX3U-2HC. The counting range covers either 16 or 32 bit.



The two integrated transistor outputs can be switched independently of one another by means of internal comparison functions. Hence, simple positioning tasks can also be realized economically. In addition, the high-speed counter modules can be used as ring counters.

### FX3U-4HSX-ADP and FX3U-2HSY-ADP

These adapter modules allow direct processing of positioning application data.



The FX3U-4HSX-ADP (far left) provides four high speed counter inputs up to 200 kHz while the FX3U-2HSY-ADP (left) delivers two channels of pulse train outputs up to 200 kHz.

**CAUTION:**

*When connecting these special adapters, same input resp. output numbers are allocated to the base unit and the special adapter. Wire either one of the input or output terminals*

### Overview of High-Speed Counter Modules/Adapters

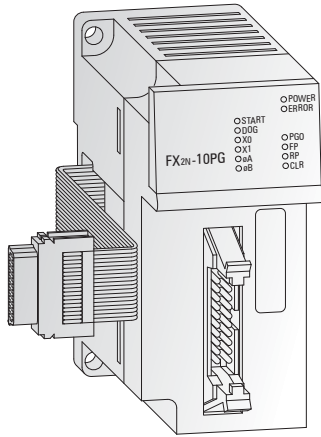
Module type	Designation	Description	FX1S	FX1N	FX2N	FX2NC	FX3G	FX3U	FX3UC
Special function module	FX2N-1HC	1-ch high speed counter	○	○	●	●	○	●	●
	FX2NC-1HC		○	○	○	●	○	○	●
	FX3U-2HC	2-ch high speed counter	○	○	○		○	●	●
Special adapter	FX3U-4HSX-ADP	Differential line driver input (high-speed counter)	○	○	○	○	○	●	○
	FX3U-2HSY-ADP	Differential line driver input (positioning output)							

- The special adapter or special function module can be used with a base unit or expansion unit of this series.
- The special adapter or special function module cannot be used with this series.

### 2.9.3 Positioning Modules

#### FX2N-1PG-E, FX2N-10PG

The positioning modules FX2N-1PG-E and FX2N-10PG are extremely efficient single-axis positioning modules for controlling either step drives or servo drives (by external regulator) with a pulse chain.



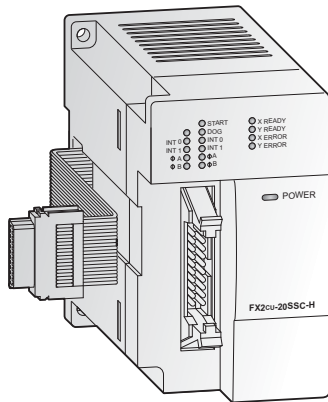
They are very suitable for achieving accurate positioning in combination with the MELSEC FX series. The configuration and allocation of the position data are carried out directly via the PLC program.

The FX2N-1PG-E provides an 100 kHz open collector output while the FX2N-10PG is equipped with a 1 MHz differential line driver output.

A very wide range of manual and automatic functions are available to the user.

#### FX3U-20SSC-H

The SSCNET\* module FX3U-20SSC-H can be used in combination with a FX3U or FX3UC programmable controller to achieve a cost effective solution for high precision, high speed positioning. The plug-and-play fiber optic SSCNET cabling reduces setup time and increases control distance for positioning operations in a wide range of applications.



Servo parameters and positioning information for the FX3U-20SSC-H are easily set up with an FX3U or FX3UC base unit and a personal computer. For parameter setting, monitoring and testing the easy programming software FX Configurator-FP is available.

\* SSCNET: Servo System Controller Network

#### Overview of Positioning Modules

Module type	Designation	Description	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
Special function modules	FX2N-1PG-E	Pulse output for independent 1-axis control	○	○	●	○	●
	FX2N-10PG						
	FX3U-20SSC-H	Simultaneous 2-axis (independent 2-axis) control (Applicable to SSCNET III)	○	○	○	○	●

- The special function module can be used with a base unit or expansion unit of this series.
- The special function module cannot be used with this series.

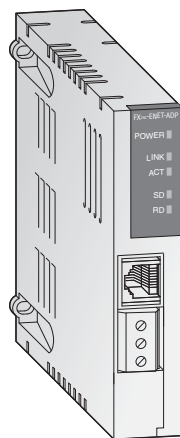
## 2.9.4 Network Modules for ETHERNET

ETHERNET is the most widespread network for connection of information processors such as personal computers and work stations. By loading an ETHERNET interface into the PLC, production-related management information can be transmitted rapidly to personal computers or work stations. ETHERNET is a platform for a very wide range of data communications protocols. The combination of ETHERNET and the extremely widespread TCP/IP protocol enables high-speed data communications between process supervision systems and the MELSEC PLC series. TCP/IP provides logical point-to-point links between two ETHERNET stations.

The programming software GX Developer provides setup routines for the PLCs, making the configuration of one or more TCP/IP links a quick and easy process.

### FX2NC-ENET-ADP

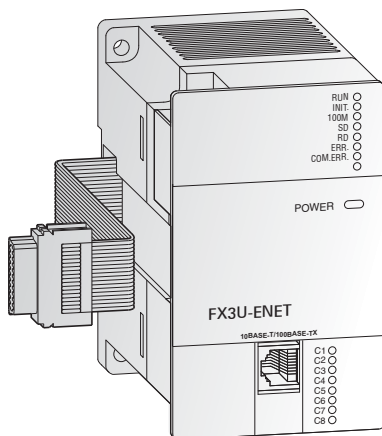
The FX2NC-ENET-ADP communications adapter is an Ethernet interface with 10BASE-T specifications for the FX1S, FX1N, FX2NC and FX2N series\*.



PCs with GX Developer or MX Component and the virtual COM port driver installed are enabled for program upload/download through this module.

\* When connecting this special adapter to a FX1S or FX1N PLC the communications adapter FX1N-CNV-BD is required. When connecting this adapter to a FX2N PLC the communications adapter FX2N-CNV-BD is required.

### FX3U-ENET



The FX3U-ENET communications module provides the FX3G, FX3U or FX3UC with a direct connection to an Ethernet network.

The FX3U-ENET enables 8 ports of simultaneous Ethernet communication with features such as peer-to-peer communication, extensive e-mail send/receive options, and program upload/download. The FX3U-ENET is also used to communicate with GOTs via the Ethernet. Easy communication parameter setup and module troubleshooting is also possible using the dedicated software, FX Configurator-EN.

### Overview of Network Modules for ETHERNET

Module type	Designation	Description	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
Special function modules	FX2NC-ENET-ADP	ETHERNET network modules	●	●	●	○	○
	FX3U-ENET		○	○	○	●	●

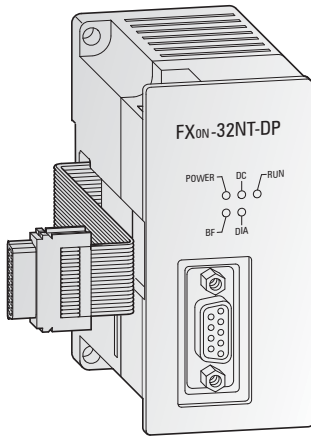
### 2.9.5 Network Modules for Profibus/DP

The Profibus/DP network enables communication between a master module and decentralised slave modules, with data transfer rates of up to 12 Mbps. With a MELSEC PLC as master, PROFIBUS/DP allows quick and simple connection of sensors and actuators, even from different manufacturers.

A MELSEC PLC, serving as slave in a PROFIBUS/DP network, can execute decentralised control tasks and simultaneously exchange data with the PROFIBUS/DP master.

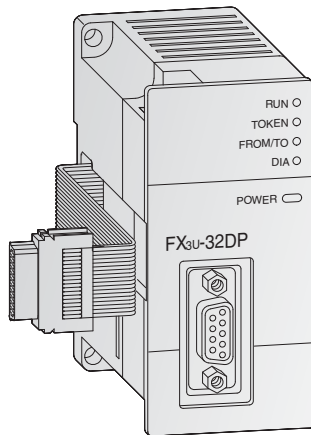
To help reduce costs PROFIBUS/DP uses RS485 technology with shielded 2-wire cabling.

#### FX0N-32NT-DP



The FX0N-32NT-DP PROFIBUS DP slave module enables the attached FX base unit to be a slave station on a PROFIBUS DP network. Transfer of up to 40 bytes of data per cycle is supported at up to 12 Mbps.

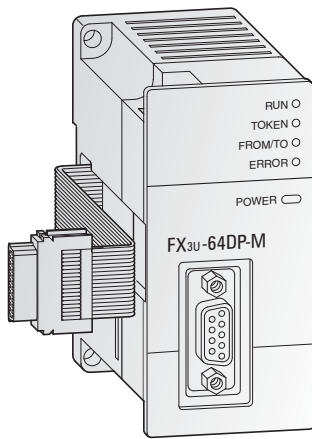
#### FX3U-32DP



Like the FX0N-32NT-DP, the FX3U-32DP is a PROFIBUS/DP slave module. It allows the integration of a FX3G, FX3U, or FX3UC PLC into a PROFIBUS/DP network.



**FX3U-64DP-M**

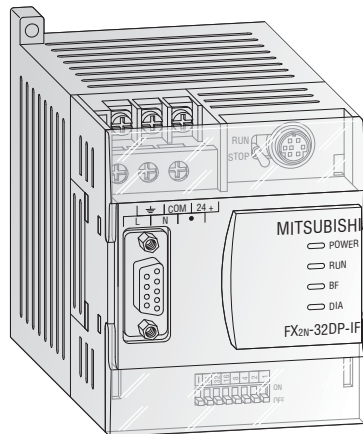


The FX3U-64DP-M PROFIBUS DP master module is available for the FX3U and FX3UC base units and enables the attached FX base unit to be a master station on a PROFIBUS DP-V1 network. PROFIBUS DP allows for the implementation of decentralized control with comprehensive data and alarm processing capabilities.

Easy setup is available by using the GX Configurator-DP software package.

**FX2N-32DP-IF**

The remote I/O station FX2N-32DP-IF forms an extremely compact communication unit and provides a connection of I/O modules with up to 256 I/O points and/or up to 8 special function modules as an alternative.



It is not necessary to install an FX base unit to a remote I/O station. The FX2N-32DP-IF connects the connected I/O modules or special function modules to the master station of a PROFIBUS/DP network.

Used in combination with a FX3U or FX3UC base unit and a FX3U-64DP-M master station, the creation of a high performance remote I/O system, consisting entirely of MELSEC FX modules, is possible.

PROFIBUS data such as the baud rate or I/O data can be monitored directly with the programming software or on the hand-held programming unit FX-20P-E. This facilitates an easy error diagnosis directly on the remote I/O station.

**Overview of Profibus/DP modules**

Module type	Designation	Description	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
Special function modules	FX0N-32NT-DP	PROFIBUS/DP slave	●	●	●	○	●
	FX3U-32DP		○	○	○	●	●
	FX3U-64DP-M	PROFIBUS/DP master	○	○	○	○	●
—	FX2N-32DP-IF	PROFIBUS/DP remote I/O station	Power supply: 100–240 V AC		Compatible with PROFIBUS/DP masters		
	FX2N-32DP-IF-D		Power supply: 24 V DC				

● The special function module can be used with a base unit or expansion unit of this series.

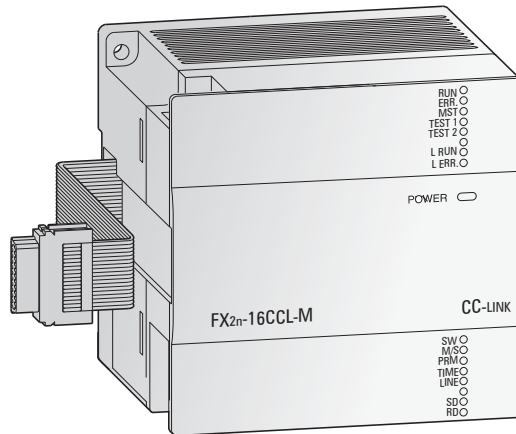
○ The special function module cannot be used with this series.

## 2.9.6 Network Modules for CC-Link

### CC-Link Master Module FX2N-16CCL-M

The CC-Link network enables the controlling and monitoring of decentralized I/O modules at the machine.

The CC-Link master module FX2N-16CCL-M is a special extension block which assigns an FX series PLC as the master station of the CC-Link system.



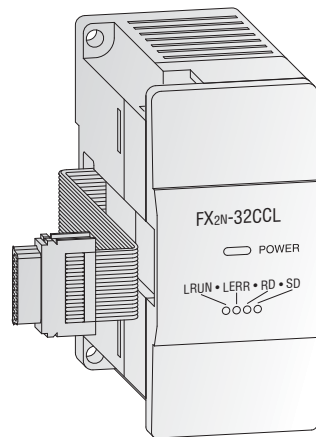
The setting of all modules within the network is handled directly via the master module.

Up to 15 remote stations (7 remote I/O stations and up to 8 remote device stations) can be connected to the master station. Two master modules can be connected to one base unit.

The maximum communications distance is 1200 m without repeater.

### CC-Link Communication Modules FX2N-32CCL and FX3U-64CCL

The communication modules FX2N-32CCL and FX3U-64CCL enable the user to connect to the CC-Link network with a superior PLC system as master CPU. This gives him access to the network of all MELSEC PLC systems and frequency inverters and to additional products from other suppliers.



Thus the network is expandable via the digital inputs/outputs of the FX modules to a maximum of 256 I/Os.

### Overview of Network Modules for CC-Link

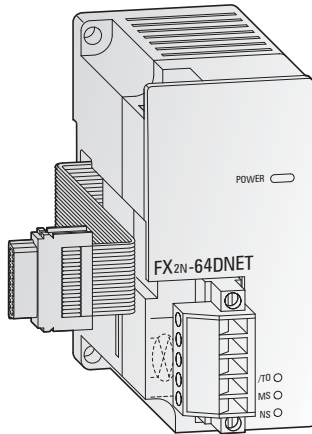
Module type	Designation	Description	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U FX3UC
Special function modules	FX2N-16CCL-M	Master for CC-Link	○	●	●	●	●
	FX2N-32CCL	Remote device station for CC-Link	○	●	●	●	●
	FX3U-64CCL		○	○	○	●	●

● The special function module can be used with a base unit or expansion unit of this series.

○ The special function module cannot be used with this series.

### 2.9.7 Network Module for DeviceNet

DeviceNet represents a cost-effective solution for the network integration of low-level terminal equipment. Up to 64 devices including a master can be integrated in one network. For the data exchange a cable with two shielded twisted-pair cables is used.



The DeviceNet slave module FX2N-64DNET can be used to connect FX2N, FX2NC and FX3U programmable controllers to a DeviceNet network.

The FX2N-64DNET can communicate to the master by the master/slave communication (using the master/slave I/O connection), and to other nodes supporting the UCMM connection by client/server communication.

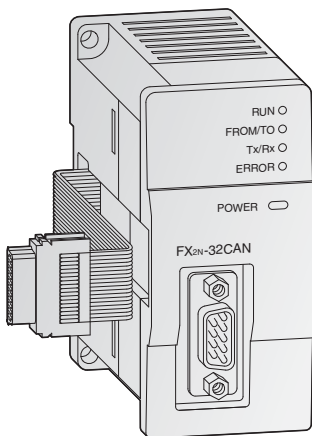
The communication between the programmable controller and the internal buffer memory of the FX2N-64DNET is handled by FROM/TO instructions.

Module type	Designation	Description	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U	FX3UC
Special function module	FX2N-64DNET	DeviceNet slave module	○	○	●	○	●	○

- The special function module can be used with a base unit or expansion unit of this series.
- The special function module cannot be used with this series.

### 2.9.8 Network Module for CANopen

CANopen is an “open” implementation of the Controller Area Network (CAN), which is defined in the EN50325-4 standard. CANopen offers cost effective network communications with fault-resistant network structure where components of different manufacturers can be integrated quickly and easily. CANopen networks are used for connecting sensors, actuators and controllers in a variety of applications. The bus uses inexpensive twisted-pair cabling.



The FX2N-32CAN communications module makes it possible to connect an FX2N, FX3G, FX3U or FX3UC PLC to an existing CANopen network.

In addition to real-time capabilities and high-speed data transfer at rates of up to 1 Mbps the CANopen module also shines with high transfer reliability and simple network configuration. Up to 120 words of data can be sent and received as process data objects (30 PDOs).

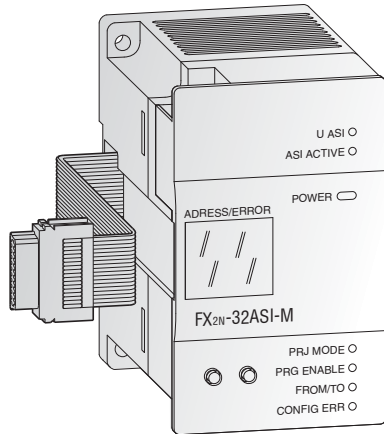
Communication with the module’s memory buffer is performed with simple FROM/TO instructions

Module type	Designation	Description	FX1S	FX1N	FX2N	FX2NC	FX3G	FX3U FX3UC
Special function module	FX2N-32CAN	CANopen module	○	○	●	○	●	●

- The special function module can be used with a base unit or expansion unit of this series.
- The special function module cannot be used with this series.

### 2.9.9 Network Module for AS-Interface

The Actuator Sensor interface (AS interface or ASi) is an international standard for the lowest field bus level. The network suits versatile demands, is very flexible and particularly easy to install. The ASi is suitable for controlling sensors, actuators and I/O units.



The FX2N-32ASI-M serves as master module for the connection of the FX1N/FX2N and FX3U/FX3UC PLC to the AS-Interface system. Up to 31 slave units with up to 4 inputs and 4 outputs can be controlled.

For status and diagnosis messages a 7-segment display is integrated.

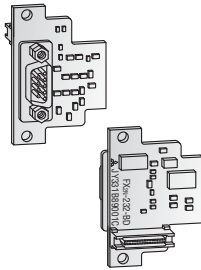
Module type	Designation	Description	FX1s	FX1N	FX2N	FX2NC	FX3G	FX3U
Special function module	FX2N-32ASI-M	Master for AS-i system	○	●	●	○	○	●

- The special function module can be used with a base unit or expansion unit of this series.
- The special function module cannot be used with this series.

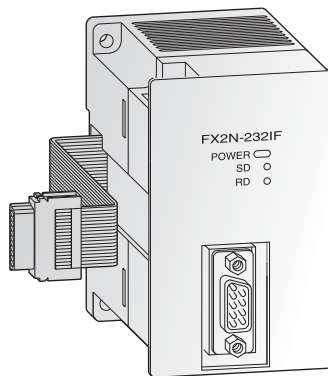
### 2.9.10 Interface Modules and Adapters

For serial data communication a large range of interface modules/adapters is available. Shown below are only some examples, but the following table covers all available interfaces.

RS232C interface adapter board FX2N-232-BD



Communication special adapter FX3U-232ADP (RS232C interface)



Interface Module FX2N-232IF

The interface module FX2N-232IF provides an RS232C interface for serial data communications with the MELSEC FX2N, FX2NC, FX3U and FX3UC.

Communication with PCs, printers, modems, barcode readers etc. is handled by the PLC program. The send and receive data are stored in the FX2N-232IF's own buffer memory.

#### Overview of Interface Modules and Adapters

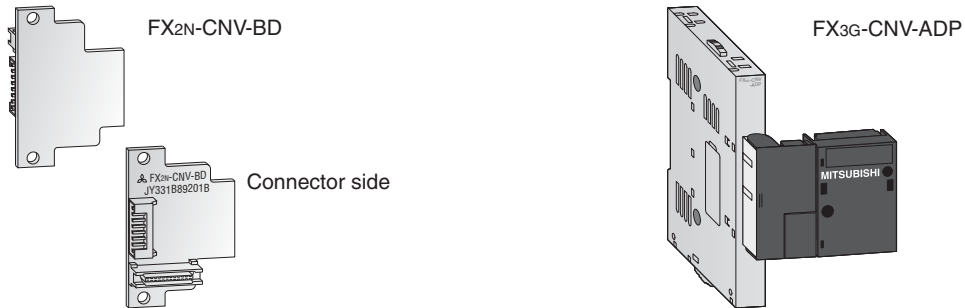
Module type	Designation	Description	FX1S	FX1N	FX2N FX2NC	FX3G	FX3U	FX3UC
Adapter boards	FX1N-232-BD	RS232C inter- faces	●	●	○	○	○	○
	FX2N-232-BD		○	○	●	○	○	○
	FX3G-232-BD		○	○	○	●	○	○
	FX3U-232-BD		○	○	○	○	●	○
Special adapter	FX2NC-232ADP*		●	●	●	○	○	○
	FX3U-232ADP-MB		○	○	○	●	●	●
Special function module	FX2N-232IF	●	●	●	○	●	●	
Adapter boards	FX1N-422-BD	RS422 interfaces	●	●	○	○	○	○
	FX2N-422-BD		○	○	●	○	○	○
	FX3G-422-BD		○	○	○	●	○	○
	FX3U-422-BD		○	○	○	○	●	○
Adapter boards	FX1N-485-BD	RS485 interfaces	●	●	○	○	○	○
	FX2N-485-BD		○	○	●	○	○	○
	FX3G-485-BD		○	○	○	●	○	○
	FX3U-485-BD		○	○	○	○	●	○
Special adapter	FX2NC-485ADP*		●	●	●	○	○	○
	FX3U-485ADP-MB		○	○	○	●	●	●
Adapter board	FX3U-USB-BD	USB interface	○	○	○	○	●	○

\* The FX2NC-232ADP and the FX2NC-485ADP require a FX2N-CNV-BD or FX1N-CNV-BD interface adapter when connecting to a FX1S, FX1N or FX2N base unit.

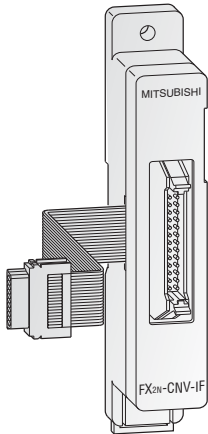
## 2.9.11 Communication Adapters

### Communication adapters boards

Communication adapters boards (product code FX□□-CNV-□□) are installed directly in a base unit. They are needed to connect special adapters (FX□□-□□□ADP) to the left-hand side of base units of the FX1N, FX2N, FX3G or FX3U series.



### FX2N-CNV-IF



The FX2N-CNV-IF interface allows special function modules of the old FX series to be connected to the base units of the FX family.

### Overview of Communication Adapters

Module type	Designation	Description	FX1S	FX1N	FX2N	FX2NC	FX3G	FX3U	FX3UC
Adapter boards	FX1N-CNV-BD	Communication adapters for connection of special adapters	●	●	○	○	○	○	○
	FX2N-CNV-BD		○	○	●	○	○	○	○
	FX2NC-CNV-IF		○	○	○	●	○	○	●
	FX3G-CNV-ADP		○	○	○	○	●	○	○
	FX3U-CNV-BD		○	○	○	○	○	●	○
Adapter	FX2N-CNV-IF	Communication adapter for connection of FX series modules	●	●	●	○	○	●	○

● The adapter can be used with a base unit of this series.

○ The adapter cannot be used with this series.

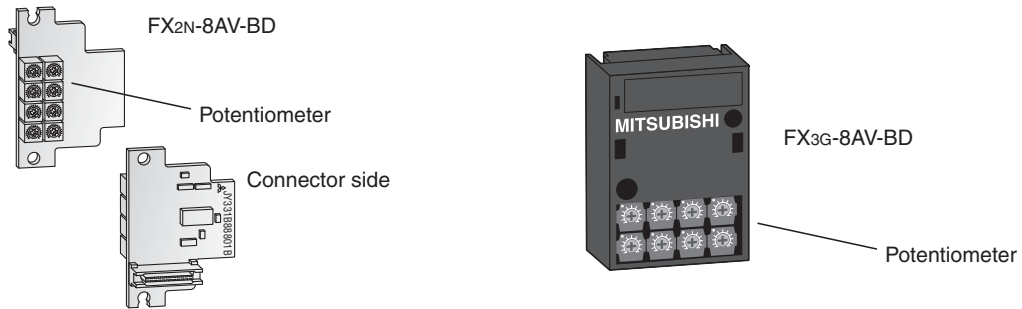
### 2.9.12 Setpoint Adapter Boards

These analog setpoint adapters enable the user to set 8 analog setpoint values. The analog values (0 to 255) of the potentiometers are read into the controller and used as default setpoint values for timers, counters and data registers by the user's PLC programs.

Each potentiometer value can also be read as an 11 position rotary switch (positions 0 to 10).

Setpoint value polling is performed in the PLC program using the dedicated instruction VRRD. The position of an rotary switch is read using the VRSC instruction.

The analog setpoint adapters are installed in the expansion slot of the base unit. No additional power supply is required for operation.



Module type	Designation	Description	FX1S	FX1N	FX2N	FX2NC	FX3G	FX3U FX3UC
Adapter boards	FX1N-8AV-BD	Analog setpoint adapters	●	●	○	○	○	○
	FX2N-8AV-BD		○	○	●	○	○	○
	FX3G-8AV-BD		○	○	○	○	●	○

- The adapter board can be used with a base unit or expansion unit of this series.
- The adapter board cannot be used with this series.

## 2.10 System Configuration

A basic FX PLC system can consist of a stand alone base unit, with the functionality and I/O range increased by adding extension I/O and special function modules. An overview of available options is given in sections 2.8 and 2.9.

### Base Units

Base units are available with different I/O configurations from 10 to 128 points but can be expanded to 384 points depending upon the FX range selected.

### Extension Boards

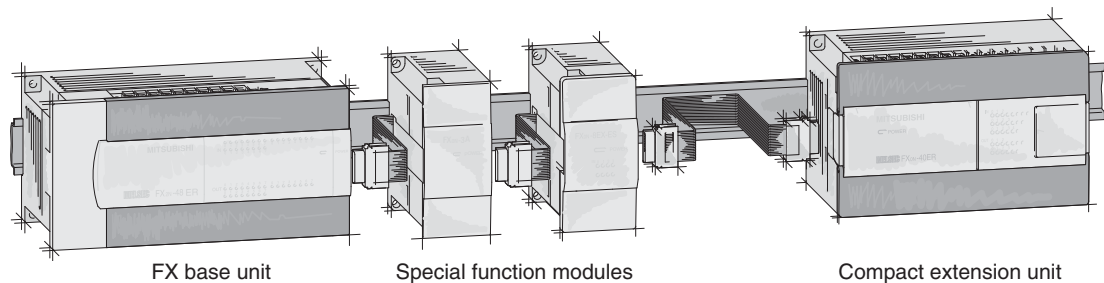
Extension adapter boards can be installed directly into the base unit and therefore do not require any additional installation space. For a small number of I/O (2 to 4) an extension adapter boards can be installed directly into the FX1S or FX1N controller. Interface adapter boards can also provide the FX PLC with additional RS232 or RS485 interfaces.

### Extension I/O Modules

With the exception of the FX1S series, unpowered modular extension blocks and powered compact extension units modules can be added to all base units of the FX family. For modular extension blocks powered by the base unit, the power consumption has to be calculated as the 5 V DC bus can only support a limited number of expansion I/O.

### Special Function Modules / Special Adapters

A wide variety of special function modules are available for all FX PLCs, again the exception is the FX1S. They cover networking functionality, analog control, pulse train outputs and temperature inputs (for further details please refer to section 2.9).





### Expansion Options

PLC	Number of modules on the left side of base unit	Number of boards in expansion board port of base unit	Number of modules on the right side of base unit
FX1S	The modules FX <sub>0N</sub> -485ADP and FX <sub>0N</sub> -232ADP can be mounted in combination with a communication adapter FX <sub>1N</sub> -CNV-BD.	1 (product code FX□□-□□□-BD)	—
FX1N			Up to 2 special function modules of the FX <sub>2N</sub> series.
FX2N			Up to 8 special function modules of the FX <sub>2N</sub> series.
FX2NC	The modules FX <sub>0N</sub> -485ADP and FX <sub>0N</sub> -232ADP can be mounted on the left side directly. An adapter is not required.	—	Up to 4 special function modules of the FX <sub>2N</sub> series.
FX3G	Up to 4 special adapters of the FX <sub>3U</sub> series can be mounted on the left side of the base unit in combination with an adapter board FX <sub>3G</sub> -CNV-BD.	Up to 2 (dependent on the type of base unit) (product code FX <sub>3G</sub> -□□□-BD)	Up to 8 special function modules of the FX <sub>2N</sub> or FX <sub>3U</sub> series.
FX3U	Up to 10 special adapters of the FX <sub>3U</sub> series can be mounted on the left side of the base unit directly or in combination with an interface/communication adapter FX <sub>3U</sub> -□□□-BD.	1 (product code FX <sub>3U</sub> -□□□-BD)	
FX3UC	Up to 6 special adapters of the FX <sub>3U</sub> series can be directly mounted on the left side of the base unit.	—	

The difference between a base unit, extension unit and extension block is described as follows:

- A base unit is made up of 4 components i.e. power supply (for base units with AC power supply only), inputs, outputs and CPU.
- An extension unit is made up of 3 components i.e. power supply, inputs and outputs.
- An extension block is made up of 1 or 2 components i.e. inputs and/or outputs.

It can be seen that the extension block does not have a power supply. It therefore obtains its power requirement from either the base unit or extension unit.

Hence it is necessary to determine how many of these unpowered units can be connected before the "On Board" power supply capacity is exceeded.

### 2.10.1 Connection of Special Adapters

Special adapters of the FX3U series can be mounted on the left side of the base unit of the FX3G, FX3U, and FX3UC series.

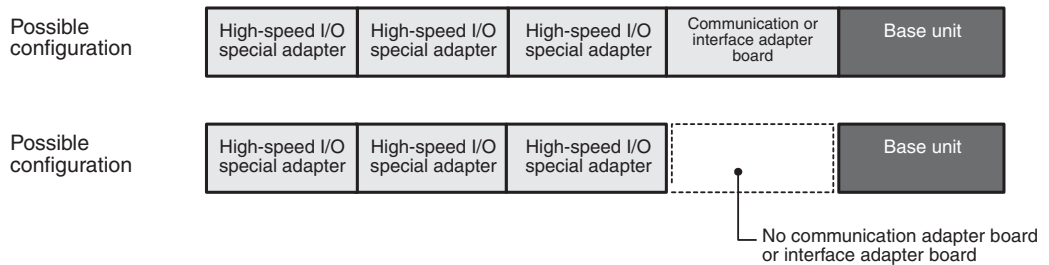
**NOTE** | The following rules apply to FX3U base units. For the rules of system configuration for the FX3G or FX3UC series, please refer to the appropriate manual.

#### High-speed input/output special adapters

Up to two high-speed input special adapters FX3U-4HSX-ADP and up to two high-speed output special adapters FX3U-2HSY-ADP can be connected to a base unit.

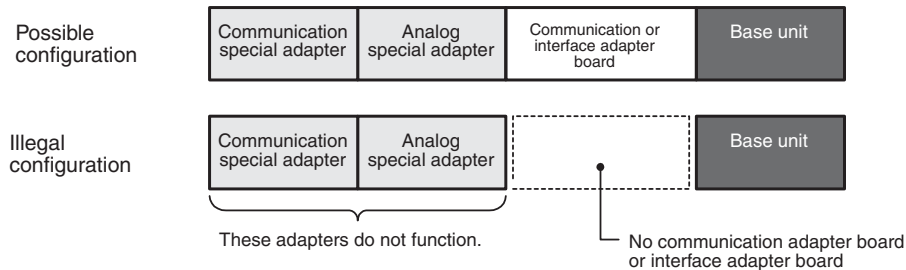
Connect all high-speed I/O special adapters before connecting other special adapters when they are used in combination. A high-speed I/O special adapter can not be mounted on the left side of a communication or analog special adapter.

When only high-speed input/output special adapters are connected, the adapters can be used without a communication or interface adapter board installed in the base unit.



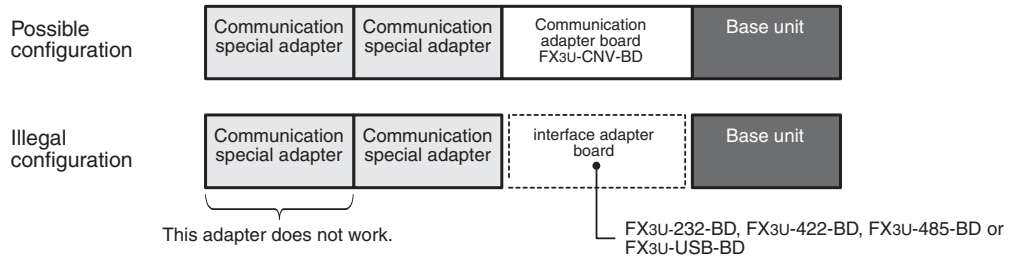
#### Combination of analog and communication special adapters

Analog and communication special adapters must be used with a communication adapter board or an interface adapter board installed in the base unit.



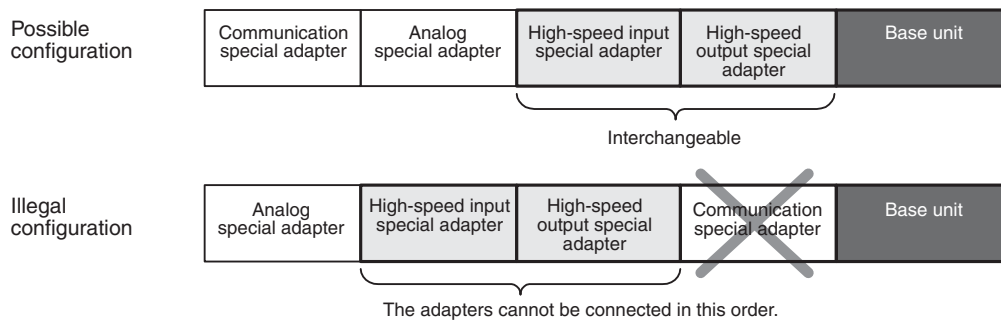
### Combination of communication special adapters and an interface adapter board

When instead of a communication adapter board FX3U-CNV-BD an interface adapter board FX3U-232-BD, FX3U-422-BD, FX3U-485-BD, or FX3U-USB-BD is mounted, one communication special adapter FX3U-232ADP or FX3U-485ADP may be used.



### Combination of high-speed input/output, analog and communication special adapters

When these adapters are used, connect the high-speed input/output special adapters on the left side of the base unit. The high-speed input/output special adapters cannot be connected on the downstream side of any communication/analog special adapter.



### Summary

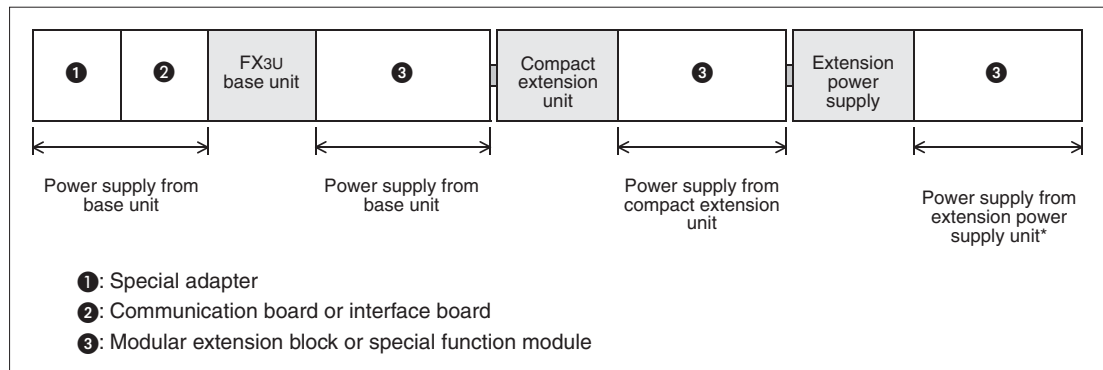
Mounted communication adapter board or interface adapter board	Number of connectable special adapters			
	Communication special adapter	Analog special adapter	High-speed input special adapter	High-speed output special adapter
No adapter board installed	These special adapters cannot be connected.			2
FX3U-CNV-BD	2	4	2	2
FX3U-232-BD FX3U-422-BD FX3U-485-BD FX3U-USB-BD	1	4	2	2

### 2.10.2 Basic Rules for System Configuration

The following considerations should be taken into account when configuring a system with extension units or special function modules:

- Current consumption from 5 V DC backplane bus
- 24 V DC current consumption
- The total number of inputs and outputs point must be smaller than the number of max. I/Os.

The following figure shows the distribution of the power supply in case of an FX3U.



\* When connecting an **input** extension block on the downstream side of an extension power supply unit, this input extension block is supplied from the base unit or from an input/output powered extension unit which is mounted between base unit and extension power supply unit.

#### Calculation of current consumption

The power is supplied to each connected device from the built-in power supply of the base unit, the input/output powered extension unit or – for FX3U and FX3UC only– the extension power supply unit.

There are three types of built-in power supplies

- 5V DC
- 24V DC (for internal use)
- 24V DC service power supply (only in AC powered base units).

The following table shows the capacities of the built-in power supplies:

Model		5 V DC built-in power supply	24 V DC built-in power supply (internal / service power supply)
Base units	FX1N	Suitable to power all connected modules	400 mA
	FX2N	290 mA	250 mA (FX2N-16M□, FX2N-32M□) 460 mA (all other base units)
	FX3G	Sufficient for 2 special function modules or 32 additional I/O	—
	FX3U	500 mA	400 mA (FX3U-16M□, FX3U-32M□) 600 mA (all other base units)
	FX3UC	400 / 480 / 560/ 600 mA	—
Compact extension unit	FX2N	690 mA	250 mA (FX2N-32E□) 460 mA (FX2N-48E□)

When only input/output extension blocks are added, a quick reference matrix can be used.

When also special function modules are added, calculate the current consumption to ensure that the total current to be consumed by the additional modules can be supplied by the built-in power supply. For details of the power consumption please refer to the appendix (section A.4).

### 2.10.3 Quick Reference Matrixes

When only input/output extension blocks without a built-in power supply are added to a base unit, a quick reference matrix can be used. The following examples are valid for base units of the FX3U series.

#### AC powered base units

In the following quick reference matrixes, the value at the intersection of the number of input points to be added (horizontal axis) with the number of output points to be added (vertical axis) indicates the remaining power supply capacity.

For FX3U-16MR/ES, FX3U-16MT/ES, FX3U-16MT/ESS, FX3U-32MR/ES, FX3U-32MT/ES or FX3U-32MT/ESS:

		see example										
Number of additional outputs	40	25										
	32	100	50	0								
	24	175	125	75	25							
	16	250	200	150	100	50	0					
	8	325	275	225	175	125	75	25				
	0	400	350	300	250	200	150	100	50	0		
			0	8	16	24	32	40	48	56	64	
		Number of additional inputs										

- Example

When a 16-input and a 16-output point extension block are connected to a base unit FX3U-16M□ or FX3U-32M□, the residual current of the 24V DC service power supply is 150 mA.

For FX3U-48MR/ES, FX3U-48MT/ES, FX3U-48MT/ESS, FX3U-64MR/ES, FX3U-64MT/ES, FX3U-64MT/ESS, FX3U-80MR/ES, FX3U-80MT/ES, FX3U-80MT/ESS, FX3U-128MR/ES, FX3U-128MT/ES or FX3U-128MT/ESS:

		see example															
Number of additional outputs	64	0															
	56	75	25														
	48	150	100	50	0												
	40	225	175	125	75	25											
	32	300	250	200	150	100	50	0									
	24	375	325	275	225	175	125	75	25								
	16	450	400	350	300	250	200	150	100	50	0						
	8	525	475	425	375	325	275	225	175	125	75	25					
	0	600	550	500	450	400	350	300	250	200	150	100	50	0			
		0	8	16	24	32	40	48	56	64	72	80	88	96			
		Number of additional inputs															

- Example

When a 32-input and a 16-output point extension block are connected to an AC powered base unit with 48, 64, 80 or 128 I/Os, the 24 V DC service power supply can still deliver a maximum current of 250 mA to other devices.

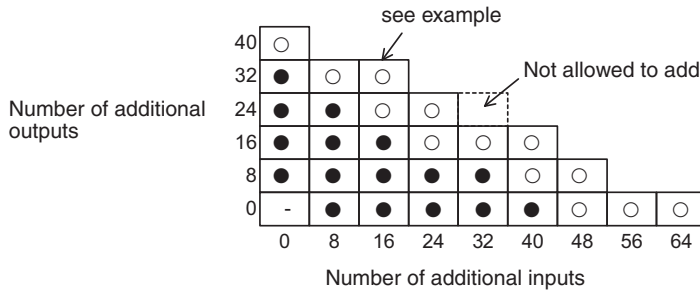
Confirm the current capacity of 24 V DC service power supply from the value shown in the quick reference matrix. This remaining power supply capacity (current) can be used as a power supply to external loads (sensors or the like) by the user. When special function modules are connected, it is necessary to consider whether they can be powered by the remaining power supply capacity.

**DC powered base units**

The DC power type base units have restrictions in expandable I/O points since they lack a built-in service power supply.

The following matrixes show the expandable units up to the ○ mark, where the desired inputs (horizontal axis) and outputs (vertical axis) intersect. System are expandable up to the ● mark when the supply voltage is 16.8 V to 19.2 V.

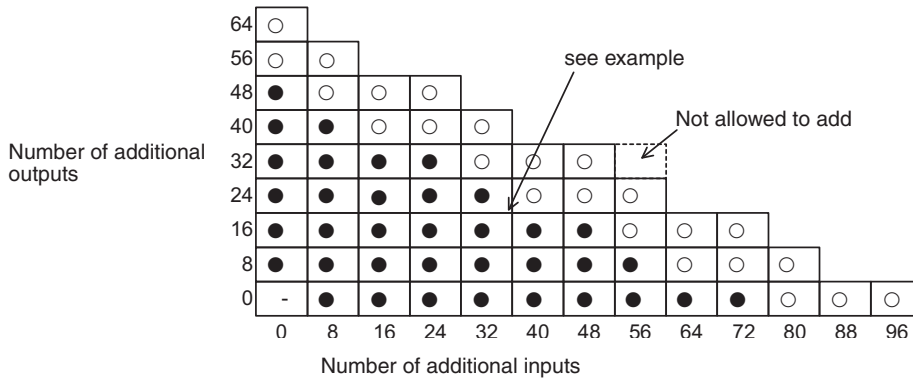
For FX3U-16MR/DS, FX3U-16MT/DS, FX3U-16MT/DSS, FX3U-32MR/DS, FX3U-32MT/DS or FX3U-32MT/DSS:



● Example

When adding 16 inputs to a DC powered base unit with 16 or 32 I/O, a maximum of 32 outputs are expandable. When adding 16 inputs under the supply voltage 16.8 V to 19.2 V, a maximum of 16 outputs are expandable.

For FX3U-48MR/DS, FX3U-48MT/DS, FX3U-48MT/DSS, FX3U-64MR/DS, FX3U-64MT/DS, FX3U-64MT/DSS, FX3U-80MR/DS, FX3U-80MT/DS or FX3U-80MT/DSS:



● Example

When adding 32 inputs to a DC powered base unit with 48, 64, or 80 I/Os, a maximum of 40 outputs are expandable. But when adding 32 inputs under the supply voltage 16.8 V to 19.2 V, a maximum of 24 outputs are expandable.

## 2.11 I/O Assignment

The assignment of the inputs and outputs in a PLC of the MELSEC FX family is fixed and can not be altered.

When power is turned on after input/output powered extension units/blocks have been connected, the base unit automatically assigns the input/output numbers (X/Y) to the units/blocks.

Therefore, it is unnecessary to specify the input/output numbers with parameters.

Input/output numbers are not assigned to special function units/blocks.

### 2.11.1 Concept of assigning

#### Input/output numbers (X/Y) are octal

The inputs and outputs of a PLC of the MELSEC FX family are counted in the octal numeral system. This is a base-8 number system and uses the digits 0 to 7.

The following table shows a comparison between some decimal and some octal numbers:

Decimal	Octal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20
:	:

Octal numbers are assigned as input/output numbers (X/Y) as shown below.

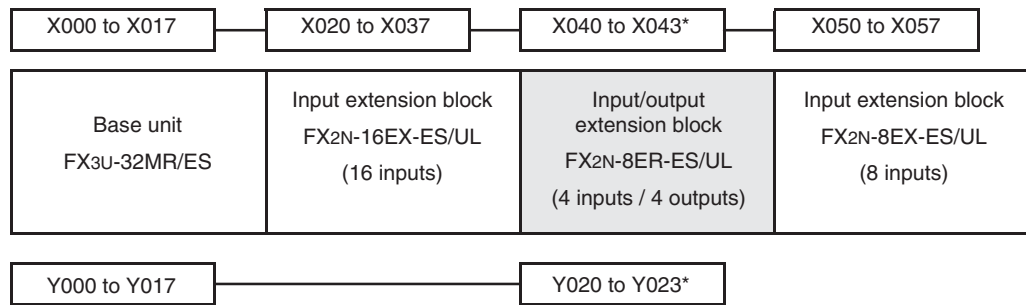
- X000 to X007, X010 to X017, X020 to X027....., X070 to X077, X100 to X107...
- Y000 to Y007, Y010 to Y017, Y020 to Y027....., Y070 to Y077, Y100 to Y107...

#### Numbers for added input/output unit/block

To an added input/output powered extension unit/block, input numbers and output numbers following the input numbers and output numbers given to the preceding device are assigned.

The last digit of the assigned numbers must begin with 0.

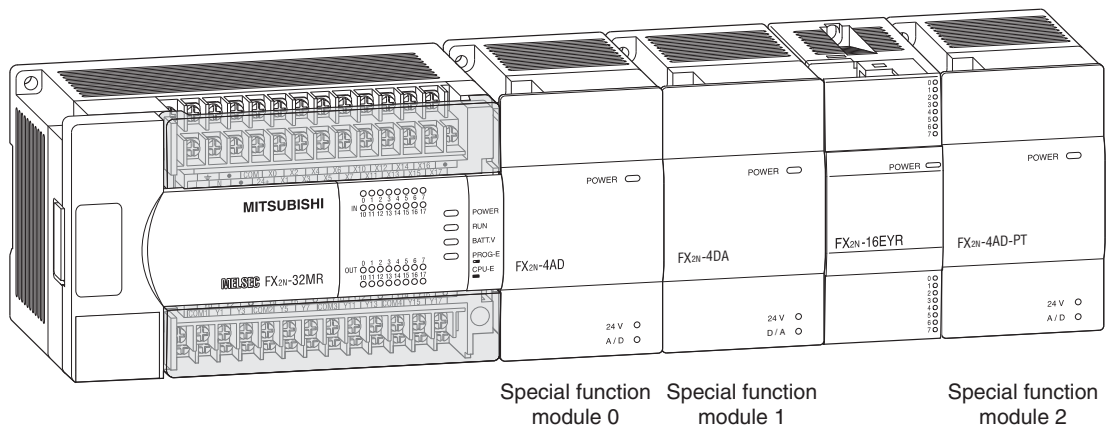
For example, when the last number on the preceding device is Y43, the output numbers are assigned to the next device starting from Y50.



\* The inputs from X044 to X047 and the outputs from Y024 to Y027 are occupied by the FX2N-8ER-ES/UL, but they can not be used.

## 2.11.2 Special function module address

Since you can attach multiple special function modules to a single base unit each module needs to have a unique identifier so that you can address it to transfer data to and from it. Each module is automatically assigned a numerical ID in the range from 0 to 7 (you can connect a maximum of 8 special function modules). The numbers are assigned consecutively, in the order in which the modules are connected to the PLC.



Special function module addresses are **not** assigned to the following products:

- Input/output powered extension units (e. g. FX2N-32ER-ES/UL or FX2N-48ET-ESS/UL)
- Input/output extension blocks (e. g. FX2N-16EX-ES/UL or FX2N-16EYR-ES/UL)
- Communication adapter (e.g. FX3U-CNV-BD)
- Interface adapter (e. g. FX3U-232-BD)
- Special adapter (e. g. FX3U-232ADP)
- Extension power supply unit FX3U-1PSU-5V



## 3 GX Developer

This course utilises Mitsubishi's GX Developer programming and monitoring software package.

The GX-Developer software is a Windows based package, which enables users to produce Ladder Diagram projects for use with the Mitsubishi range of PLC's.

It has been produced by Mitsubishi Electric to replace the popular DOS based package "MEDOC".

### 3.1 Advantages of GX Developer

The GX Developer software is windows based and thus offers many advanced features including:

- All program functions can be accessed using icons from tool bars on the console, as well as dropdown menus and shortcut keys.
- Ladder diagrams can be entered rapidly using fast entry key sequences or point and click tools.
- Program modifications can be easily carried out either "on or off-line". Changes may also be written to the program in the PLC while in Run mode.
- Unlimited use of the Windows clipboard enables program editing to be carried out quickly and efficiently.
- Superior monitoring facilities are offered including batch, entry data and direct monitoring of the contents of the buffer memory areas of special function modules. Different elements of a ladder diagram may also be monitored simultaneously.
- Advanced Fault Finding and diagnostic features are offered.
- Improved documentation and context sensitive help
- Various program structuring tools are provided that improve program readability and viability, particularly operation sequencing.
- Extensive program documentation tools are offered.

Full program simulation may be carried out without the need for any PLC hardware.

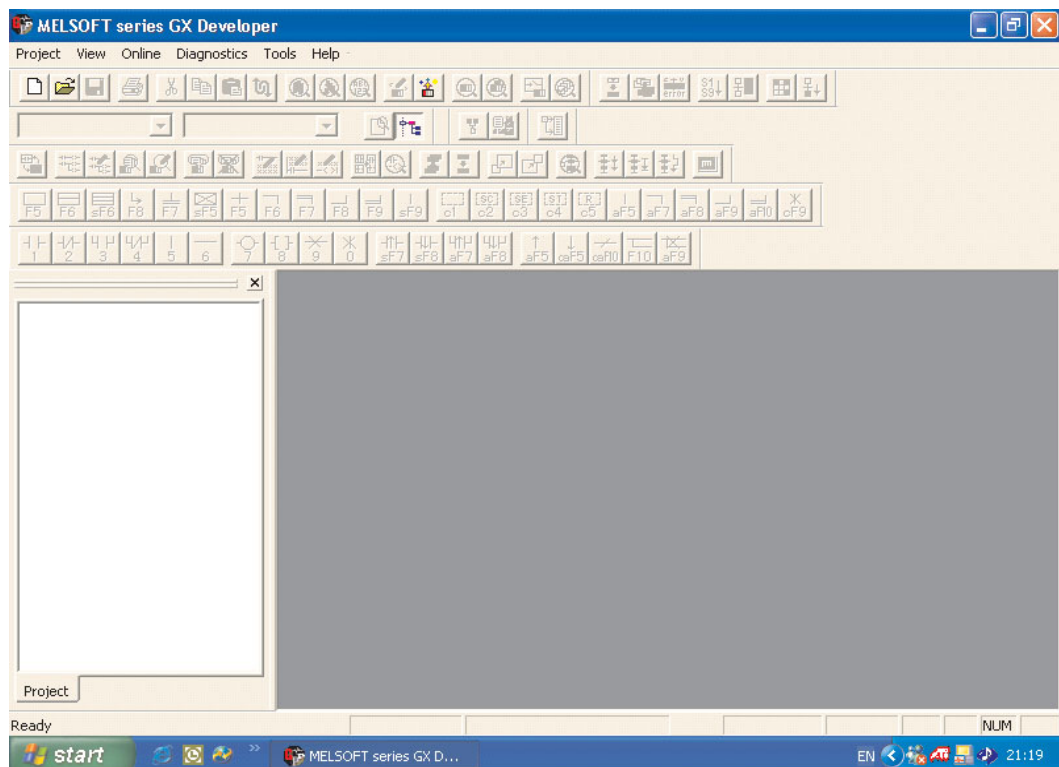
## 3.2 Programming Software Initialisation

When using GX Developer for the first time, it is advisable to alter some of the program defaults in order to optimise the working environment.

The following procedures customise GX Developer for optimised operation for the remainder of this course.

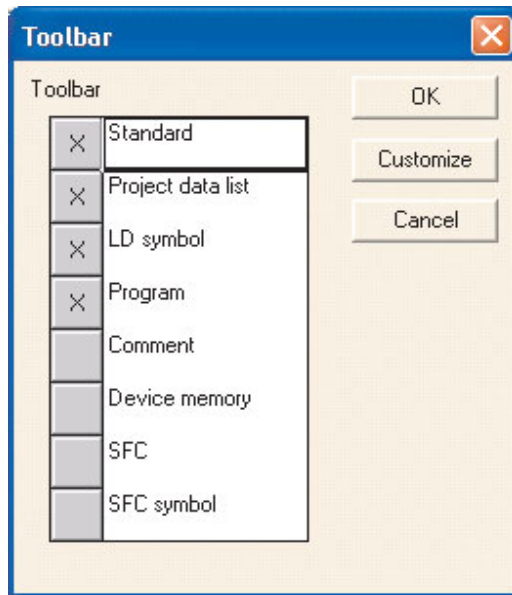
### Procedure:

- ① From Windows Desktop, select ***GX Developer***.
- ② The display comes up as shown below.

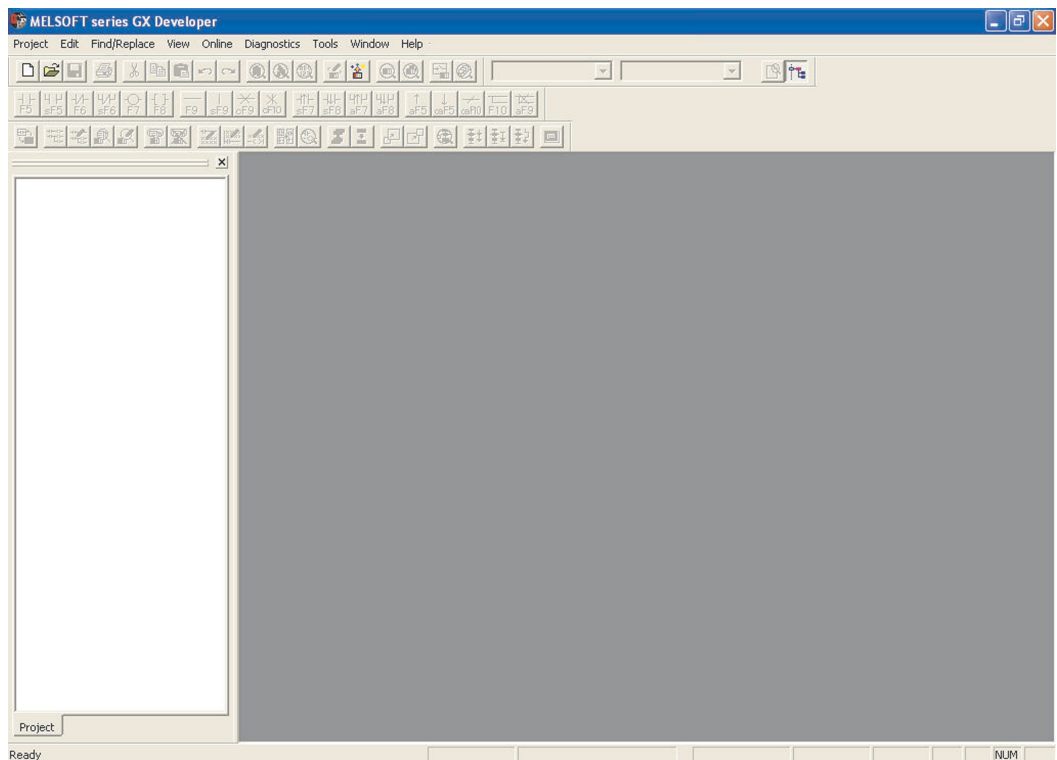


As can be seen from the above display there are large number of icons and this can be confusing to the first time user. Hence initially, it is recommended that only an essential minimum number of icons should be displayed.

- ③ From the Main Menu, select **View** and then **Toolbar**. Deselect the items which no longer are identified by an X, so that the display appears as shown below.



- ⑤ Select **OK** and the display will now be as shown below





# 4 Creating a Project

The following section describes the procedures required to create a new project with GX Developer using an example program COMPACT\_PROG1.

The Program will be used to illustrate how a PLC Ladder diagram can be produced, modified and tested. Then using a Mitsubishi PLC of the FX family, the program will be downloaded, run and monitored.

## 4.1 Example PLC Program (COMPACT\_PROG1)

This program enables a PLC output i.e. Y0, to be turned ON/OFF at a controlled rate. In this example the Output Y0 will be ON for 1 second and then OFF for 1 second. When Y0 is OFF then Y1 is ON and visa versa.

### PLC Ladder Diagram



### 4.1.1 Line Numbers

In the descriptions that follow, references will be made to Line Numbers.

A Line Number is the Step Number of the first element for that particular line.

Therefore Line Numbers will not increase by one from one line to the next, but will depend on the number of steps used by the elements, for each line. The usage of program steps varies between different PLC types.

## 4.1.2 Principle of Operation

- Line 0
  - On closing the Input switch X0, the timer T0 will be enabled via the normally closed contact of Timer T1.
  - Timer T0 will now start timing out and after 1 second, the Timer will operate. This means:
    - Any T0 normally open contacts -| |-, will close.
    - Any T0 normally closed contacts -| / |-, will open.
  
- Line 5
  - The normally open contact of T0 will close, and the normally closed contact will open causing the following to occur:
    - Timer T1 will become enabled and start timing out.
    - Output Y0 will become energised i.e. Output Y0 will turn ON.
    - Output Y1 will be de-energised i.e. Output Y1 will turn OFF.
  
- Lines 0 & 5

After Timer T1 has been energised for 1 second, it will also operate and its normally closed contact will open, causing Timer T0 to drop-out.
  
- With Timer T0 dropping-out, its normally open contact will now re-open causing:
  - Timer T1 to drop-out.
  - Output Y0 to become de-energised i.e. Output Y0 will turn OFF and Y1 will turn ON.
  
- Hence it can be seen that Timer T1 is part of a 'cut-throat' circuit, in that its operation immediately causes itself to drop-out. This operation must be viewed in conjunction with the PLC program scan cycle process.
  
- With Timer T1 dropping out, its normally closed contact will close, and for as long as Input X0 is closed; the operation will be constantly repeated.
  
- Lines 5 & 10

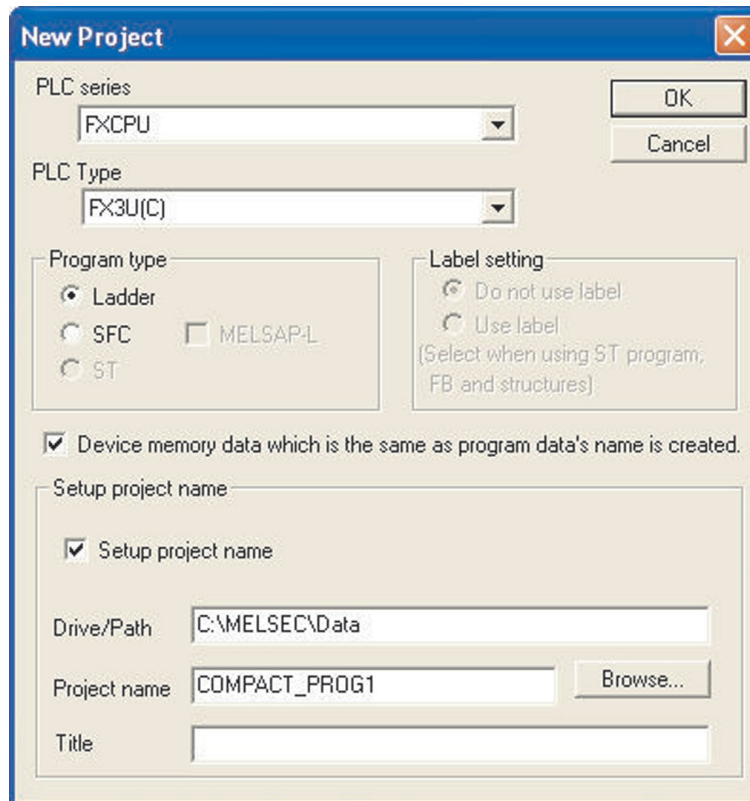
Hence the Output Y0 will be continuously OFF for 1 second and then ON for 1 second and visa versa for output Y1 (Line 10).

## 4.2 Start Up Procedure

① From the **Project** menu, select **New Project** as:



② Enter the details in the **New Project** selection window as illustrated below:



- PLC series: **FXCPU**
- PLC Type: Selection depends on the CPU used. Read the description on front of CPU and make your selection here.
- Program type: **Ladder**
- Device memory data...: **Ticked**
- Setup project name: **Ticked**
- Drive/Path: C:\MELSEC (The actual Drive/Path varies with computer configuration).

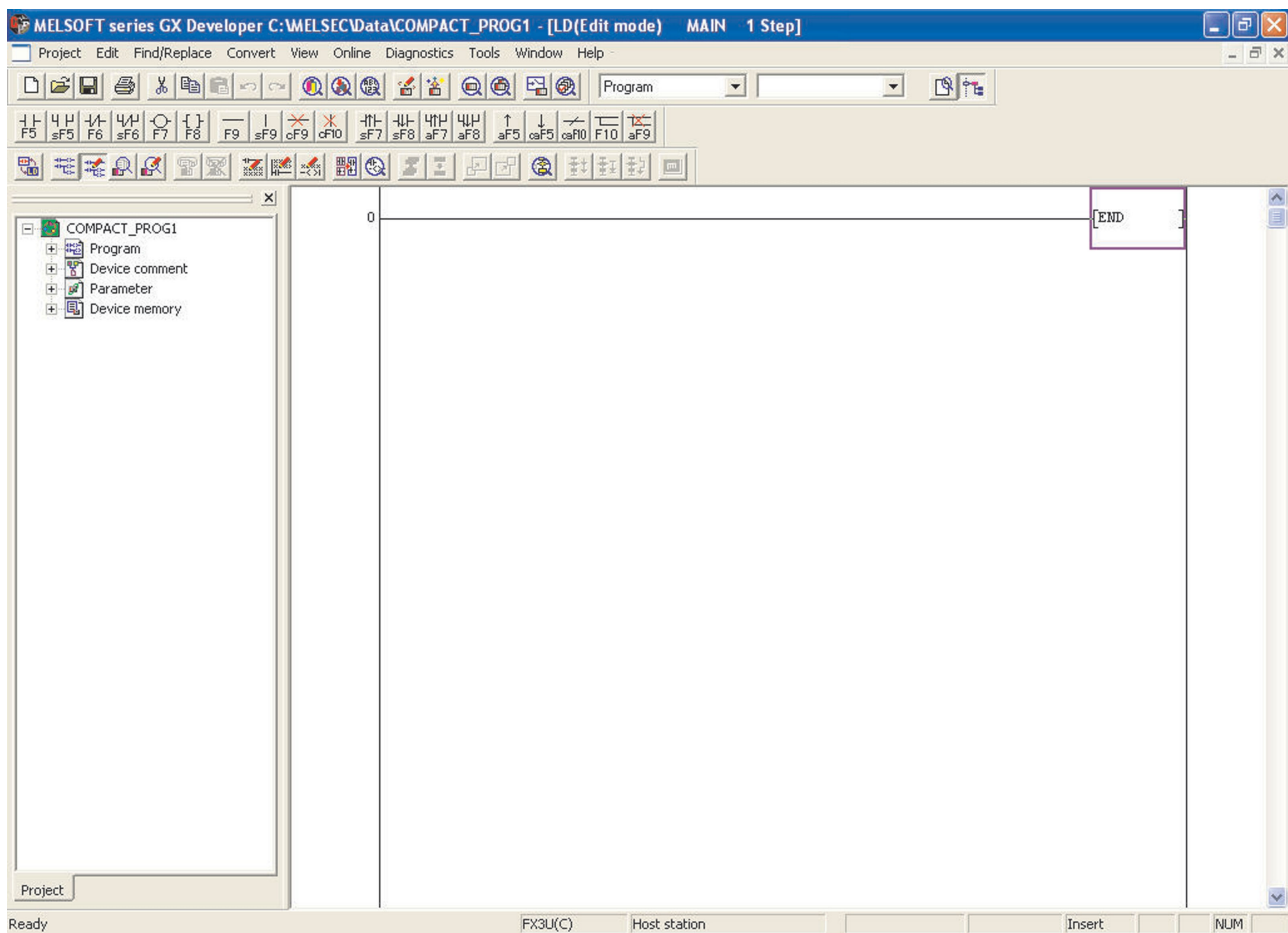
**NB:** You may consider using the following pathname in order to keep your programs separate from the others that may be present on the hard disk of the computer:  
**C:\MELSEC\Your Company name\Project name**

- For this example, use the Project Name: **COMPACT\_PROG1**.
- The **Title** is optional. Any description could be entered here.

- ③ Select the **OK** button. The following message will appear:











- ④ Select the **Yes** button.
- ⑤ The display will now be as shown in the following screen shot.





### 4.3 Ladder Diagram Elements

Previously the Ladder Diagram Elements in the tool bar have optionally been set with exactly the same numbers as for MEDOC i.e.

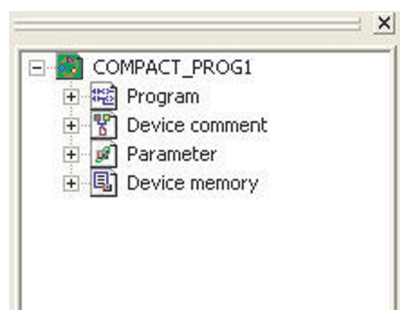
- Normally Open contact 
- Normally Closed contact 
- Normally Open Parallel contact 
- Normally Closed Parallel contact 
- Vertical Line 
- Horizontal Line 
- Output coil 
- Function Command 

This means that the Ladder Diagram can be constructed by either:

- Using the mouse and selecting the required element.
- Entering the Function key corresponding to the required element.

### 4.4 Project Data List

The Project Data List is displayed on the left hand side of the Ladder Diagram as shown below. This window displays the directory structure of the displayed project. It is used to ease navigation between various elements of the program. This list varies dependent on the PLC CPU model specified:

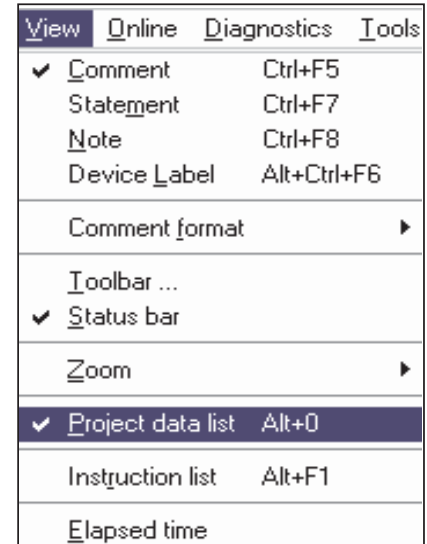




## 4.5 Toggle Display Project Data List

To improve the clarity of the Ladder Diagram, the Project Data List can be removed from the display. This is useful particularly for smaller video displays for example Laptop and LCD's.

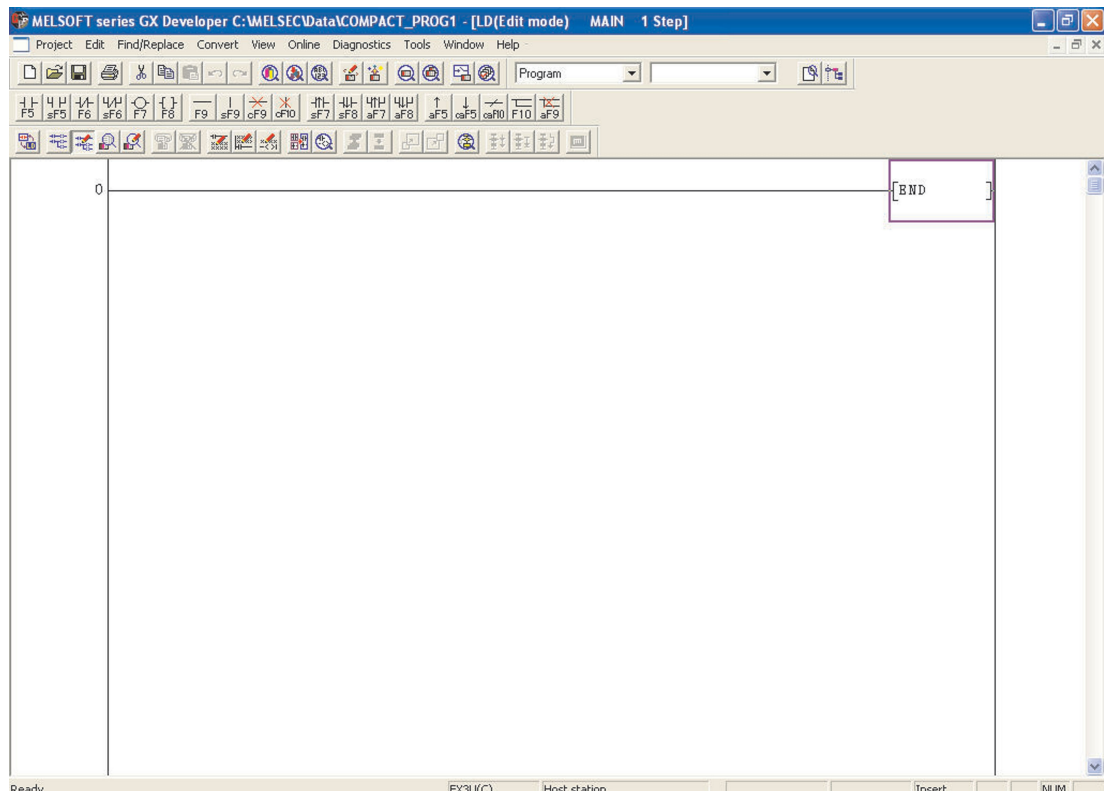
To remove the Project Data List from the displayed area, the following procedure should be adopted.

From the Main Menu select **View** and click 'Toggle' (de-select) the **Project data list**.



- Alternatively 'Toggle' click the button from the  tool bar to select / Deselect the Project Data List display window.
- The Project Data list can also be removed by clicking on the  "Close Window" on the top right of the Project Data List Window.

The altered display is shown below:

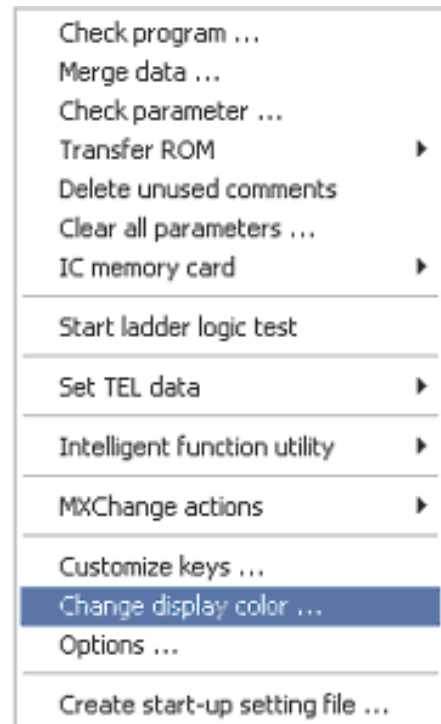


## 4.6 Changing the Colour Attributes (Optional)

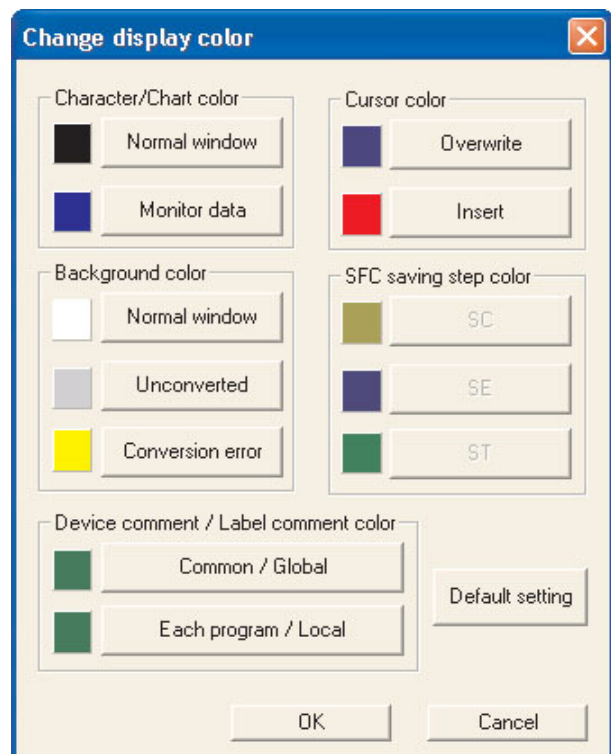
Changing the Colour Attributes is optional. Before proceeding with the Ladder Program it is recommended to carry out the following procedure:

Due to the poor choice of **default** colours used in the editing functions, it is recommended that colour attributes should be altered for the “**Insert**” cursor function in order to provide improved visibility. The colour attributes will be stored from now-on by GX Developer but it is a requirement that a project is first opened in order to alter this particular group of settings. These modified settings will therefore be used for the remainder of this course:

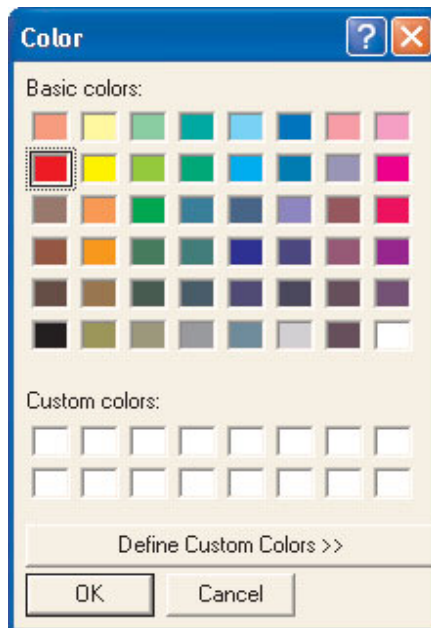
- ① From the **Tools** menu, select the **Change Display Colour** option, thus:



- ② The display colour attribute window is then displayed.



- ③ Click on the **Insert** button for the **Cursor colour** function. The following colour palate window will be displayed:



- ④ Click on the bright red box in the above window and then click OK. This alters the colour attribute for the cursor in “Insert” mode from Purple to bright Red.
- ⑤ Having carried out these operations, this is the configuration that best suits the format of the following training notes.

## 4.7 Entering the Ladder Diagram (COMPACT\_PROG1)

The Ladder Diagram of COMPACT\_PROG1 as shown at the beginning of this section will now be entered.

① Entering the first contact, Normally Open X0

- Using the mouse or "F5" from keyboard, select the normally open contact.



- Enter the name X0.



- Select **OK**.
- The Ladder Diagram now becomes as shown below.



② Entering the second contact, Normally Closed T1.

Using the keyboard, enter:

- T1
- Select **OK**



- The Ladder Diagram now becomes as shown below.



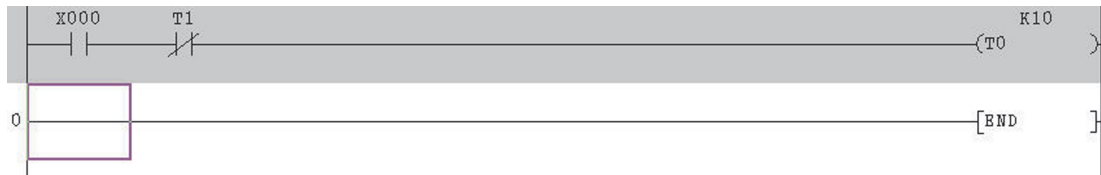
③ Output, Timer T0.

Enter the following:

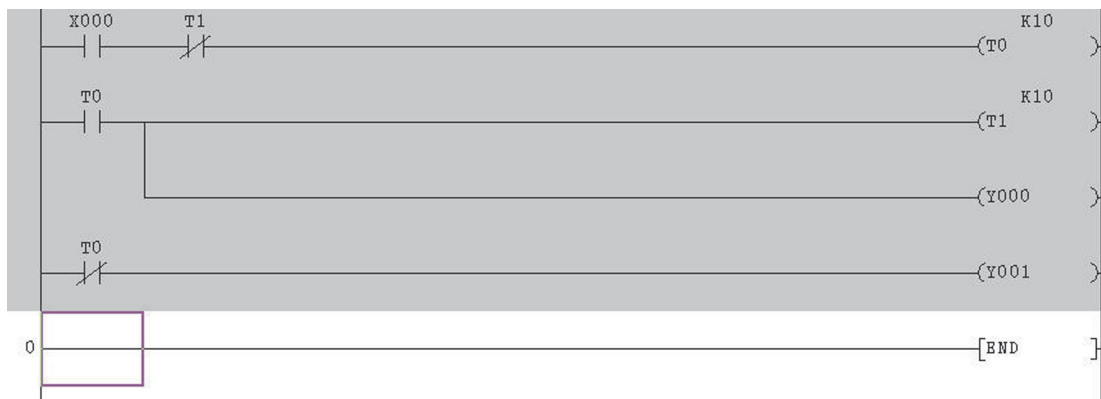
- Function key "F7"
- T0
- Space
- K10
- **OK**



- The Ladder diagram will be as displayed below:



④ Complete the Ladder diagram as shown below:



**NOTE**

There is no need to enter the instruction END as it is always on the last line of the Ladder Diagram and is created automatically by GX Developer.


## 4.8 Conversion to an Instruction Program

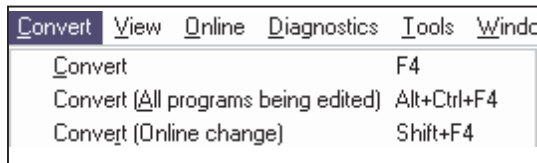
Before the program can be saved, the Ladder Diagram must first of all be converted into a set of MELSEC™ instructions.

(Incidentally, “MELSEC” is the brand name used by Mitsubishi Electric for their PLC products and is derived from the term: “Mitsubishi Electric Sequencers”)

To execute the conversion process, carry out the following:

- ① From the Main Menu, select **Convert**.

- ② Select the **Convert** function. Alternatively click on either of the  buttons or simply press the F4 key.



The Ladder diagram will now be converted to instruction code for the PLC and the resulting display will be as shown below.



**NOTE**

The grey unconverted background area becomes clear and line numbers appear at the start of each line.

## 4.9 Saving the Project

To save the project on the hard drive, carry out the following.

① From the Main Menu, select **Project**.

② Select **Save**.

Alternatively press the  button on the tool bar.

The project will now be saved to C:\MELSEC\COMPACT\_PROG1 on the computer hard disk drive. (Dependent on individual computer preferences)





## 5 Instruction List Programming

An Instruction List program is an alternative method for producing PLC Programs.

The instruction program is the actual command instructions that the PLC executes when running a program.

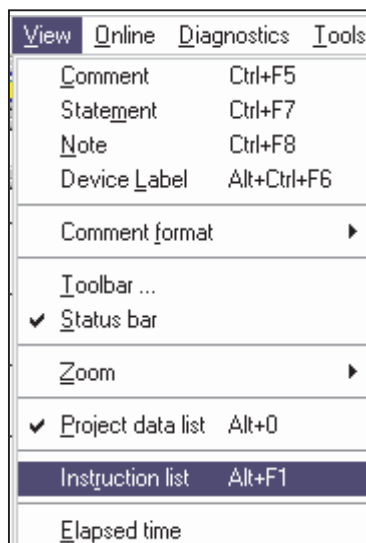
However, unless a programmer is very skilled at producing such programs, it is usually preferred that the program be produced by the ladder method.

Where GX Developer has been used to produce a Ladder Diagram, then the equivalent Instruction Program can easily be displayed.

### 5.1 Instruction List Program (COMPACT\_PROG1)


To obtain the equivalent Instruction Program for COMPACT\_PROG1, carry out the following.

- ① From the Main Menu select,
  - **View**
  - **Instruction List**



- ② Displayed on the screen will be the Instruction List representation of the program for COMPACT\_PROG1.

#### NOTE

By toggling the keys <Alt> F1 or by clicking the  button on the toolbar, the Ladder Diagram or equivalent Instruction Program can be displayed.

**Ladder Diagram- COMPACT\_PROG1**




**Instruction Program – COMPACT\_PROG1**

```

0  LD      X000
1  ANI     T1
2  OUT     T0      K10
5  LD      T0
6  OUT     T1      K10
9  OUT     Y000
10 LDI     T0
11 OUT     Y001
12 END
    
```

**NOTES**

It may be necessary to cursor up the Instruction List display in order to view the entire program.

To further improve the viewability of the Instruction list, use the Zoom Up / Down buttons on the toolbar thus: 

## 5.2 Explanation - Instruction List Programming

### Start of a Rung

Where the first contact on each rung is a normally open contact, then the equivalent Instruction will always be:

- LD (Load).

Where the first contact on each rung is a normally closed contact, then the equivalent Instruction will always be:

- LDI (Load Inverse)

### Contacts in Series

Where there is more than one contact connected in series, then to obtain an Output, all of the contacts must be correctly operated.

- i.e. X0 ON, T1 OFF

Hence for the timer coil T0 to be energised, input X0 is operated AND the input T1 is not operated. This is written in an instruction program as

- LD X0
- ANI T0

Hence after the first contact on each rung, any additional series connected contacts, will be preceded by the following:

- AND for all normally open contacts
- ANI for all normally closed contacts

### Outputs

Each rung must be terminated by one or more outputs i.e.

- Output solenoid 'Y'
- Timer coil 'T'
- Counter 'C'
- Internal memory bit (relay) 'M'
- Special instructions i.e.
  - Pulse (One shot on rising edge) 'PLS'
  - Master Control Contact 'MC'
  - End of program 'END'
- An Applied / Functional Instruction i.e.
  - Block move 'BMOV'
  - Addition 'ADD'
  - Multiplication 'MUL'

All output solenoid (coil) instructions are preceded with the instruction OUT, followed by the output number and if appropriate, a constant K value i.e.

OUT T0 K10

This indicates that Timer T0 has been programmed to give an ON time delay of (10 x 0.1 ms) = 1.0 second.



## 6 Find

The Find option is an extremely useful facility in that it enables:

- An immediate jump to a particular step number.
- A search for a particular element.

### 6.1 Find Step Numbers

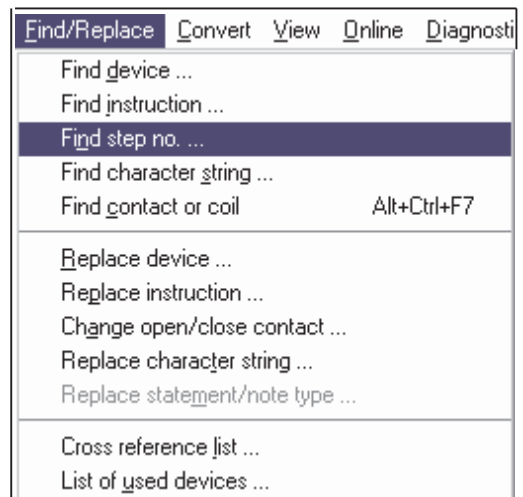
Where a project contains a large number of steps, it is advantageous to be able to jump to a known part of the program, than have to cursor down from Step 0.

To use this facility, carry out the following:

- ① Let the project COMPACT\_PROG1 be displayed as shown below.



- ② From the Main Menu, select **Find/Replace**.



- ③ Select **Find step no.**

The **Find step no.** window now appears as shown below.



- ④ Enter 5, <OK>.

Note that the program immediately jumps to the start of Line 5.

Hence using this method, any part of the program can be quickly accessed. Repeat the procedure to jump back to the start of the Ladder Diagram.

## 6.2 Find Device

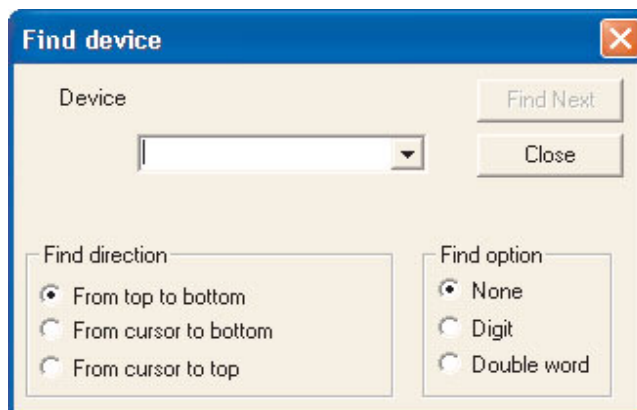
This facility enables a search for an I/O device and GX Developer will search for this device and stop at the first match.

- ① Let the Project COMPACT\_PROG1 be displayed as shown below.



- ② From the **Find/Replace** menu, select **Find device**.

The display now changes to:



- ③ Enter T0.

- ④ Select **Find Next**.

On the Ladder Diagram of COMPACT\_PROG1, it can be seen that the coil of T0 is highlighted.

- ⑤ Selecting **Find Next** again, will cause the next occurrence of T0 to become highlighted, i.e. the normally open contact of T0 at Line 5.

- ⑥ Select **Find Next** once more and note the next occurrence of T0 at Line 10.

- ⑦ Continue selecting **Find Next** until all of the T0 elements have been found i.e. when the message on the right is displayed.

- ⑧ Select OK and then close the **Find device** window.



## 6.3 Instruction Search

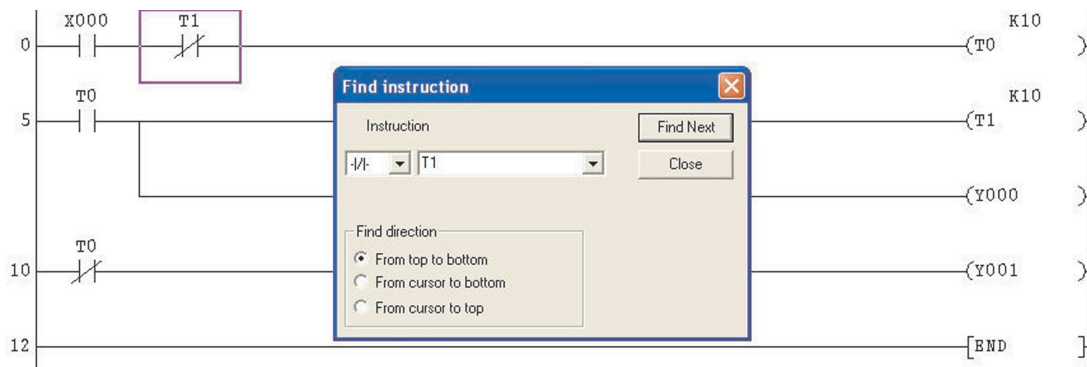
Instruction Search is an extremely useful facility which enables a search to be carried out for a particular program instruction.

Hence where a Ladder Diagram contains a large number of steps and it is difficult to determine if a particular Instruction is being used, then the Instruction Search facility can confirm whether or not it is in the program.

The following describes how using the project COMPACT\_PROG1, a search is carried out for the “Normally Closed Contact” of T1. It will be assumed that the ladder diagram COMPACT\_PROG1 is being displayed.

- ① From the Main Menu select the following.
  - **Find/Replace.**
  - **Find instruction.**
- ② Using the triangular symbol on the left hand drop down box, select the symbol or ‘Normally Closed Input’ and enter T1 in the right hand box (See Below).
- ③ Click the **Find Next** button

The display will now appear as shown below, with the first normally closed contact of T1 enclosed within the blue cursor square.



- ④ Repeatedly select **Find Next**, until all of the matching input instructions have been found.  
When there are no more items found matching the search criteria then the following message is displayed:



- ⑤ Select **OK** and then close the **Find instruction** window.

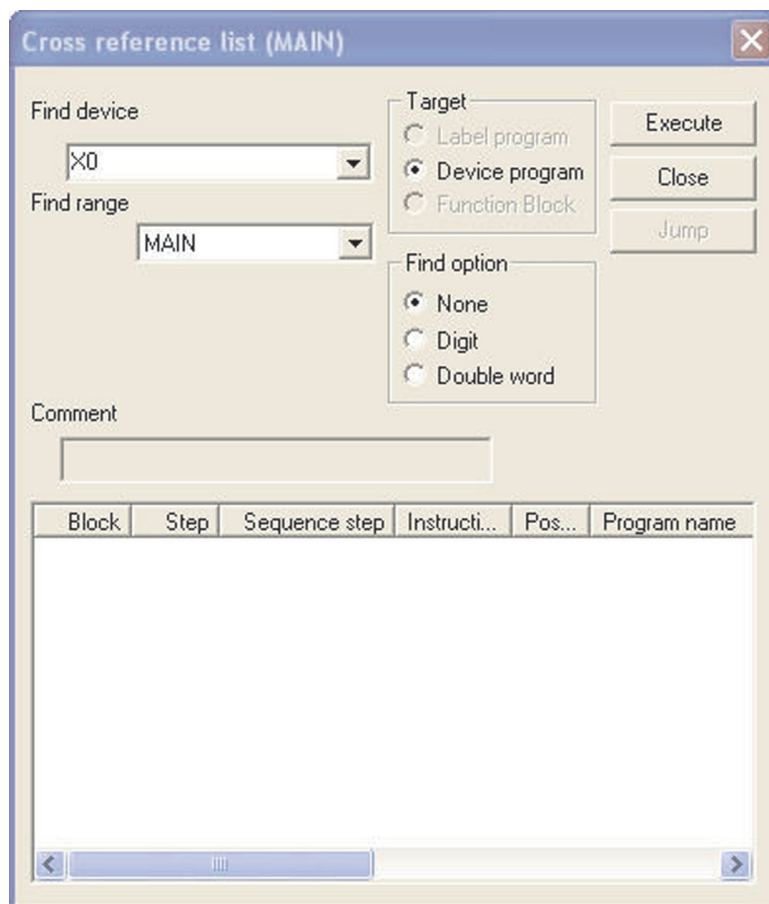


## 6.4 Cross Reference List

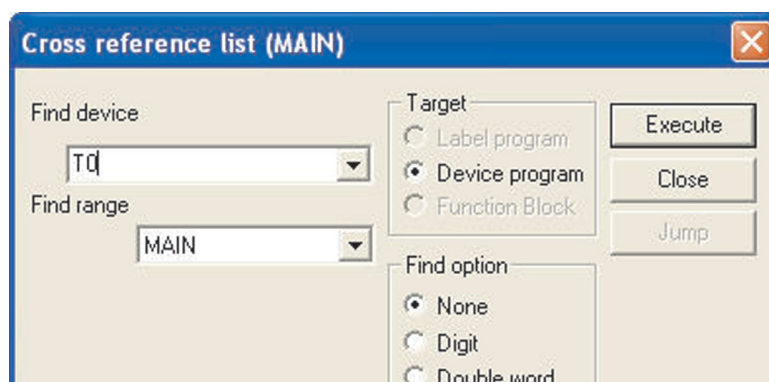
The **Cross Reference List** produces a display of the step numbers for both the coil and contacts of the selected device where they appear on the ladder diagram. This is very important when fault finding a project and there is a need to track a particular device through the ladder diagram.

The following procedure describes how the cross reference details for the Timer T0 in the project COMPACT\_PROG1 are obtained.

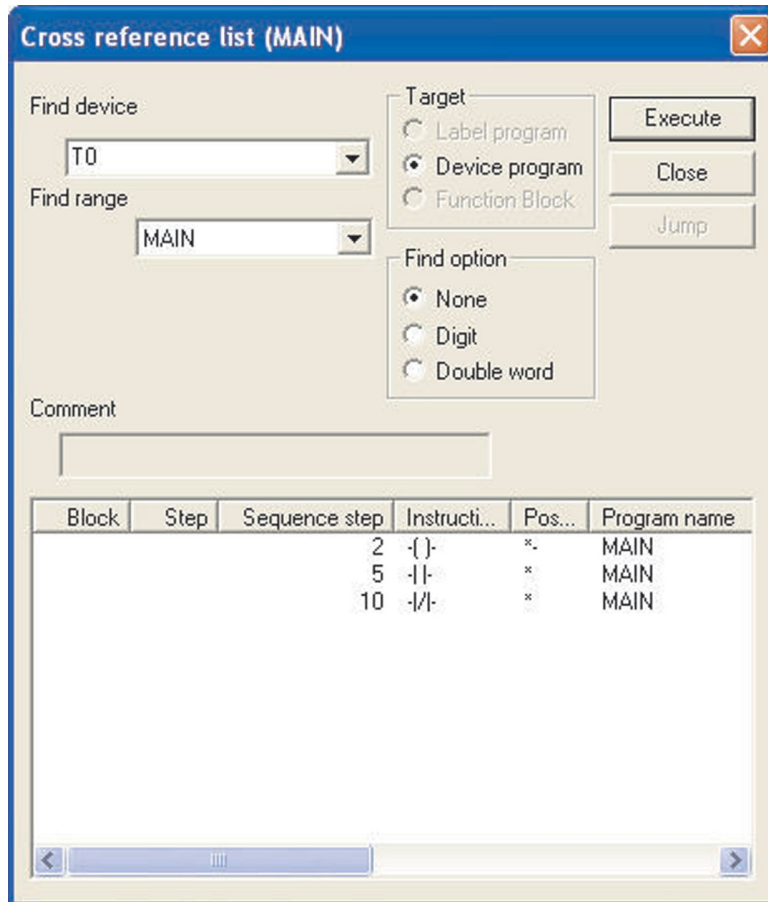
- ① From the Main Tool Bar select **Find/Replace**.
- ② Select **Cross reference list**.
- ③ The following window is displayed:



- ④ Enter T0 in the **Find device** window.



- ⑤ Select **Execute** and all the step numbers of where T0 occurs in the project COM-PACT\_PROG1, will be displayed.



- ⑥ Select **Close** to return to the ladder diagram.

## 6.5 List of Used Devices

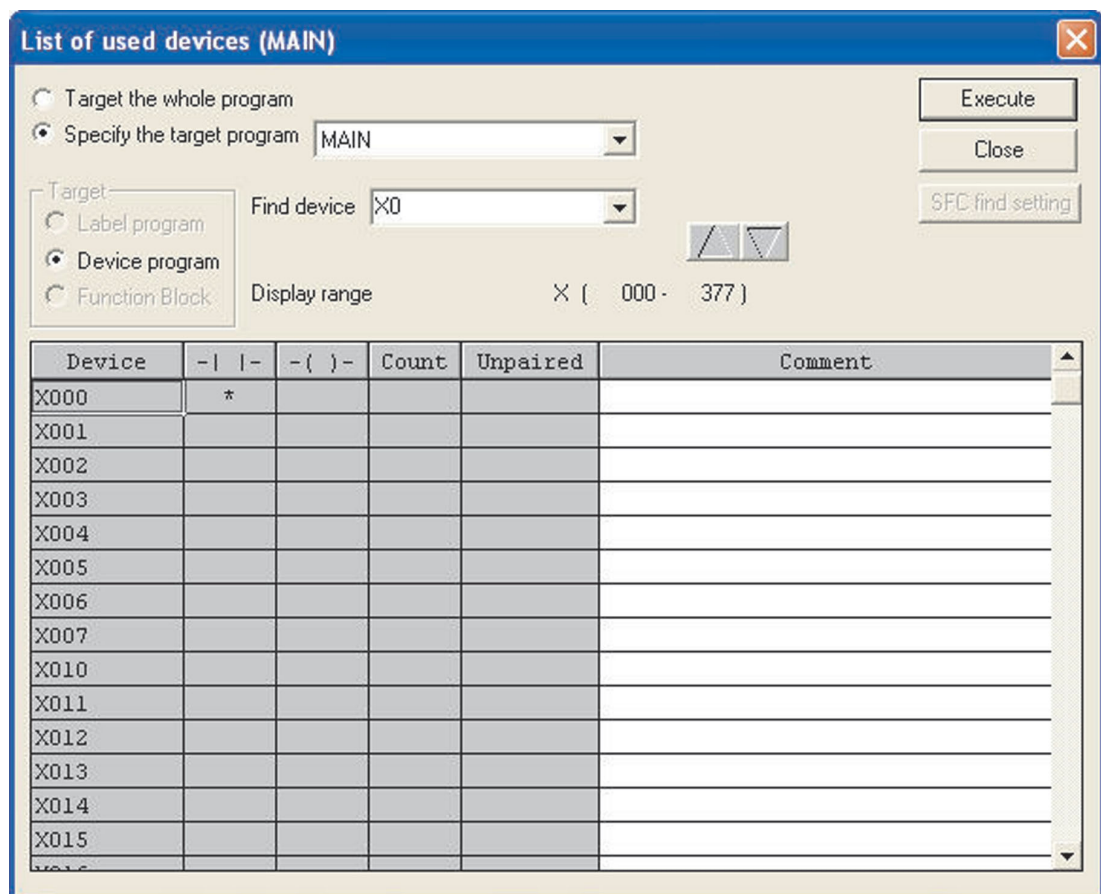
Another useful facility, which is in the **Find/Replace** menu, is the **List of Used Devices** function.

The list enables the user to see what devices are being used in the project.

This is very useful when modifications to the ladder diagram are required, as it shows therefore what devices are not being used and hence those devices are available for use in the modification to the program.

The following procedure describes how all of the timers used in the project COMPACT\_PROG1 are listed.

- ① From the Main Toolbar select **Find/Replace**.
- ② Select **List of used devices**, as shown in the display below.
- ③ The display now becomes as shown below:



- ④ As can be seen from the previous display, an entire range of X input devices stating with X0 are being displayed.

In addition it can be seen there is a "\*" in the contact column for X0. This indicates that X0 is used in the project COMPACT\_PROG1.

- ⑤ Enter T0 in the **Find device** window.

- ⑥ Select **Execute** and the display shows that Timers T0 and T1 are being used in the project COMPACT\_PROG1.

Hence the next available timer which can be used is T2.

The screenshot shows a software dialog box titled "List of used devices (MAIN)". It contains several controls:
 

- Radio buttons for "Target the whole program" (unselected) and "Specify the target program" (selected). A dropdown menu next to "Specify the target program" shows "MAIN".
- Buttons for "Execute", "Close", and "SFC find setting".
- A "Target" section with radio buttons for "Label program" (unselected), "Device program" (selected), and "Function Block" (unselected).
- A "Find device" dropdown menu showing "T0".
- A "Display range" field showing "T ( 0 - 199 )".
- A table with columns: Device, -| |-, -( )-, Count, Unpaired, and Comment.

 The table contains the following data:
 

Device	-   -	-( )-	Count	Unpaired	Comment
T0	*	*	1		
T1	*	*	1		
T2					
T3					
T4					
T5					
T6					
T7					
T8					
T9					
T10					
T11					
T12					
T13					

# 7 Copying Projects

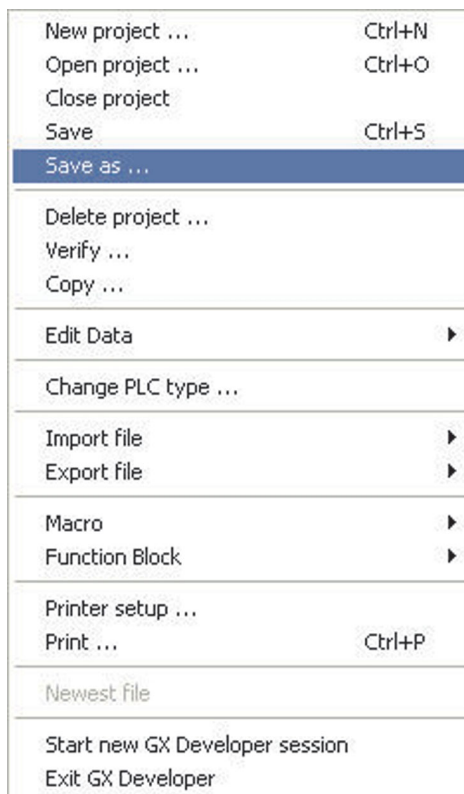
This section describes how an existing project can be copied to a second project, which has a different filename. This is required when modifying an existing project and yet still retains a copy of the original Ladder Diagram.

This is necessary in case the modifications do not work as expected and therefore the original project has to be re-loaded into the PLC, so that production can be maintained.

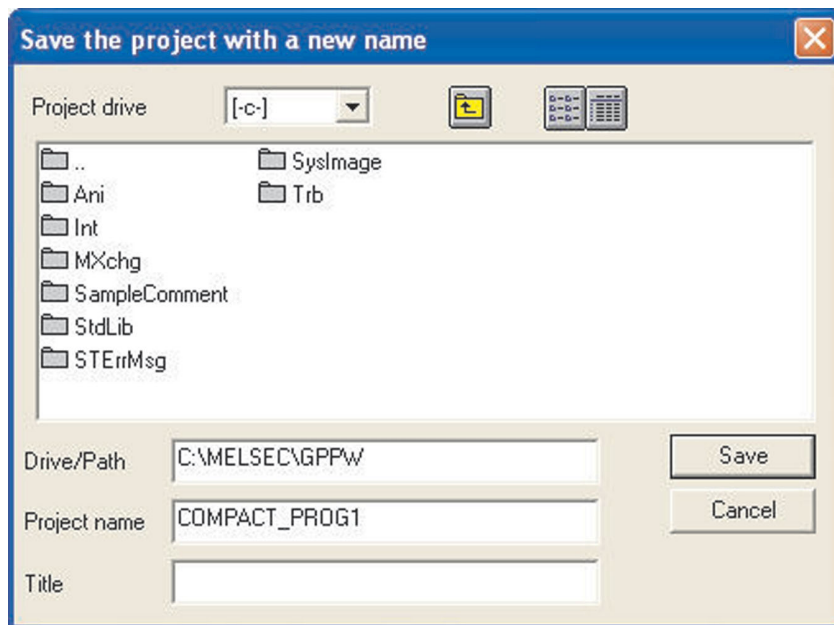
## 7.1 Copying of the project COMPACT\_PROG1

Hence prior to modifying the existing project COMPACT\_PROG1, it is necessary to copy COMPACT\_PROG1 to project COMPACT\_PROG2. This is done as follows:

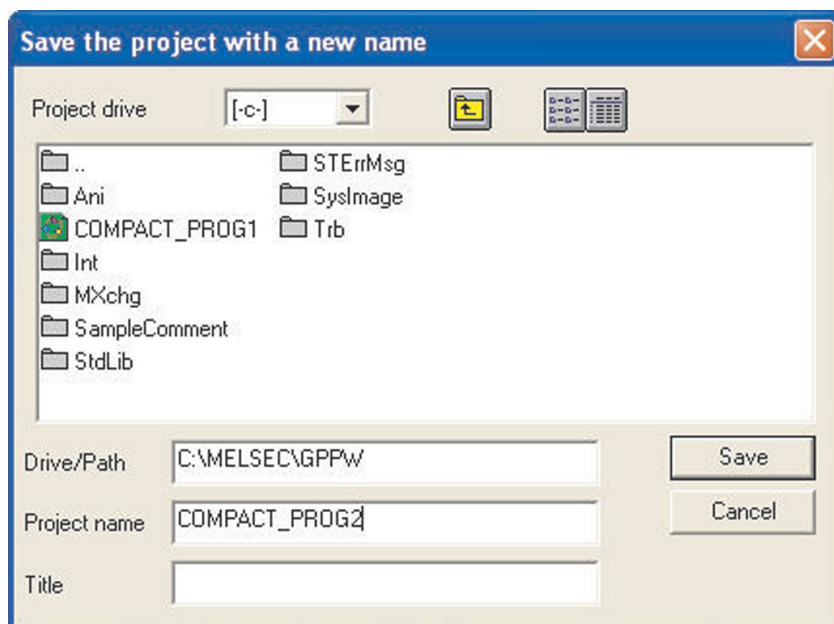
- ① From the Main Menu, select **Project**.
- ② Select **Save as**.



- ③ The display now becomes as shown below:



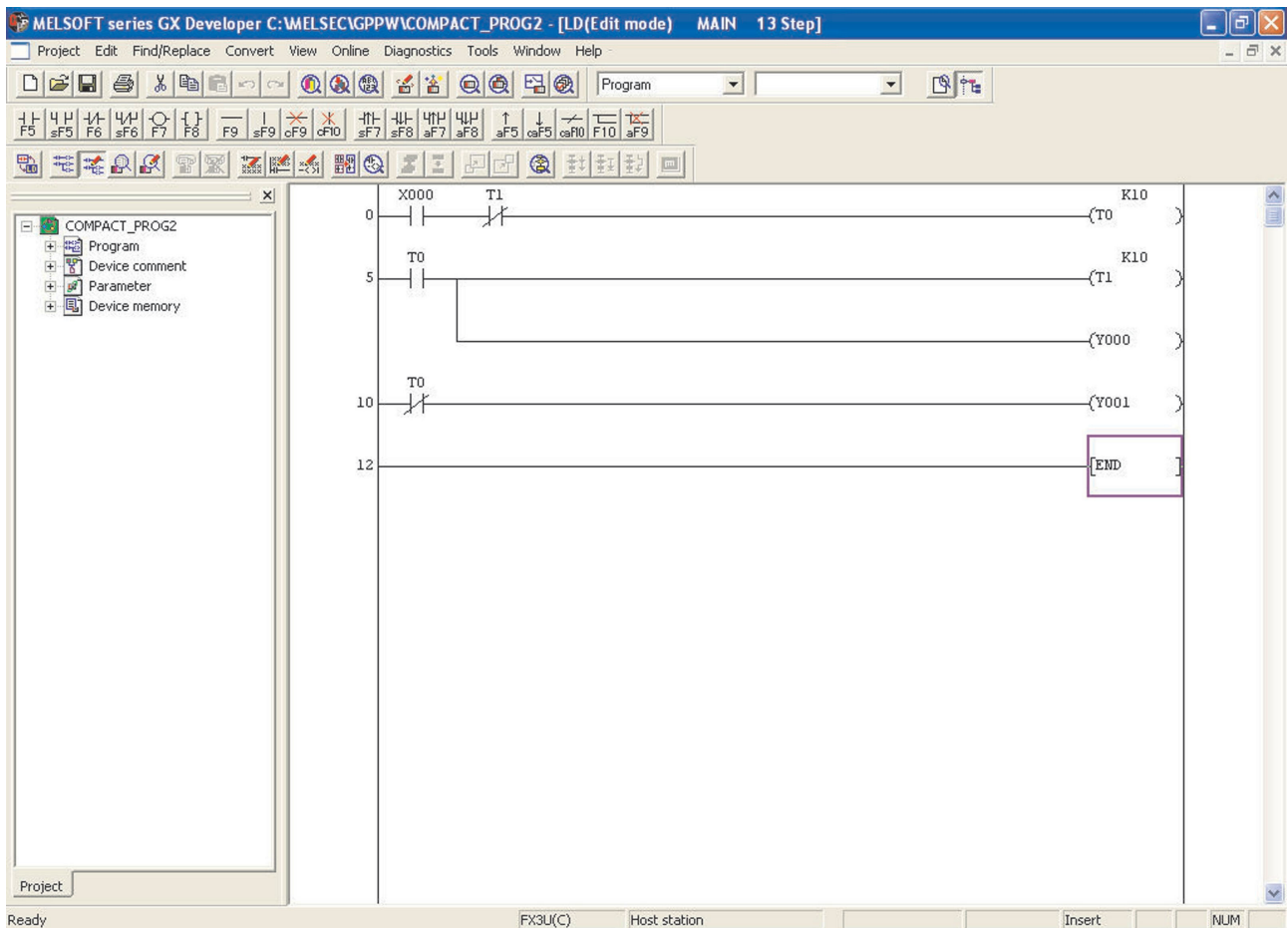
- ④ Change the **Project name** to COMPACT\_PROG2.



- ⑤ Select **Save** and the following message box is displayed:

- ⑥ Select **Yes**, to create the new project COMPACT\_PROG2.

⑦ The display now appears as shown below.



**NOTE**

The Project name has changed to COMPACT\_PROG2 (see the top program information bar). The project COMPACT\_PROG1 can still be recalled, whenever required.





## 8 Modification of Ladder Diagrams

### 8.1 Modification of the project COMPACT\_PROG2

Before any modifications can be carried out, it is necessary for the Ladder Diagram COMPACT\_PROG2 to be displayed on the screen.

At the moment COMPACT\_PROG2 is identical to COMPACT\_PROG1.



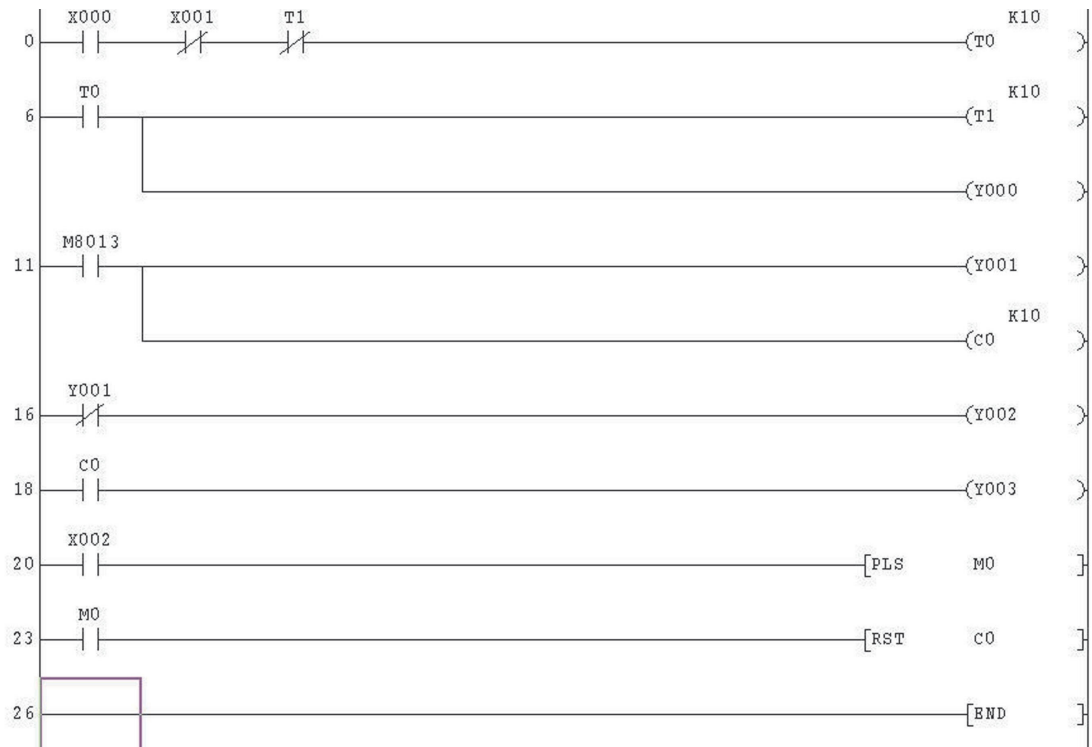
#### Modification Details

As can be seen from the following altered diagram COMPACT\_PROG2, the modifications consist of:

- Line 0: The insertion of a normally closed input X1.
- Line 11: Modification of normally open contact from T0 to M8013 \*  
Insertion of an additional rung: Output coil C0 K10
- Insertion of an additional rung: normally open contact of C0 driving output coil Y3
- Insertion of an additional rung: normally open X2 driving a Pulse [PLS M0] instruction.
- Insertion of an additional rung: normally open M0 driving a Reset [RST C0] instruction.

\* M8013 is one of many special relays in the PLCs of the MELSEC FX family. M8013 switches at a frequency of 1Hz and is derived from the internal crystal based clock. It is internally driven by the CPU which makes it ideal for accurate timing applications. Refer to the appendix for a full description of special relays.

**Modified Ladder Diagram COMPACT\_PROG2**



## 8.2 Insertion of a new contact


To insert the normally closed contact X1, between X0 and T1, it will be necessary to change from OVERWRITE mode to INSERT mode.

- ① This is done, by pressing the <Insert> key on the keyboard. Note the lower right mode box changes to **Insert**.

Note:

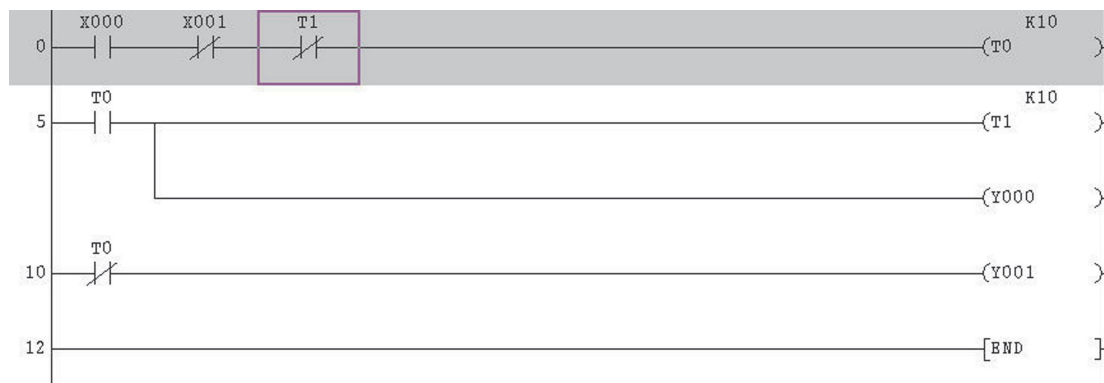
- The colour of the edging around the square changes to bright red.
- The word **Insert** now appears in the bottom right hand corner of the VDU display.

- ② Move the cursor over the normally closed T1 contact using the cursor keys on the keyboard or by left double clicking the mouse key over the contact.

- ③ Click on  or enter F6 for a normally closed contact.

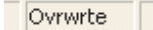
- ④ Enter the contact name X1 <Enter>.

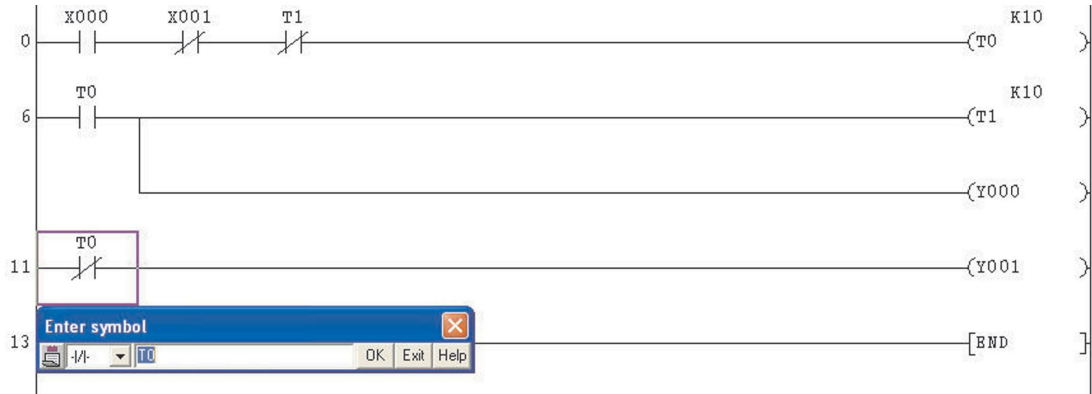
- ⑤ Line 0 will now include the normally closed contact X1.





- ⑥ Press F4 to convert the addition of the normally closed X1.

### 8.3 Change of Device Detail

- ① Press the “Insert” button on the keyboard and note the change of mode back to “Overwrite”  (cursor colour changes to blue).
- ② Move the cursor over to normally closed contact of T0 on line 11. Double click the mouse or press <Enter> and the following will be displayed.




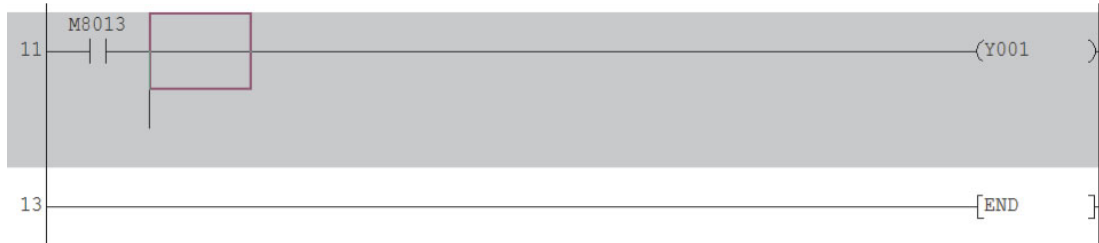
- ③ Click the small downward triangle in the left of the symbol box and select a normally open contact.
- ④ Alter T0 to M8013 and press **OK**. Press F4 or the   buttons to convert the changes and the display will be as follows:




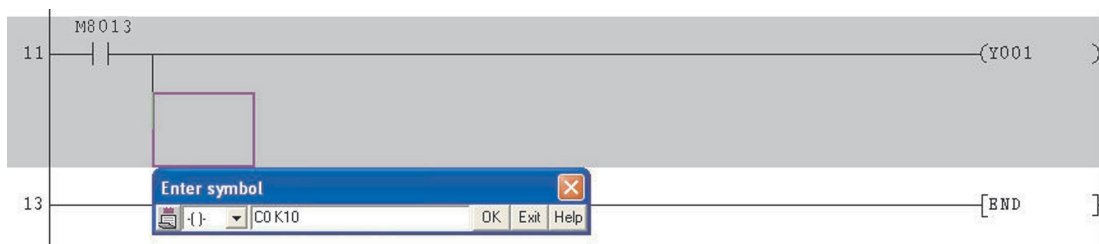
## 8.4 Inserting a Branch

- ① To insert the output C0 K10 as a branch to line 11, go to insert mode. The cursor turns red to indicate the changed mode.

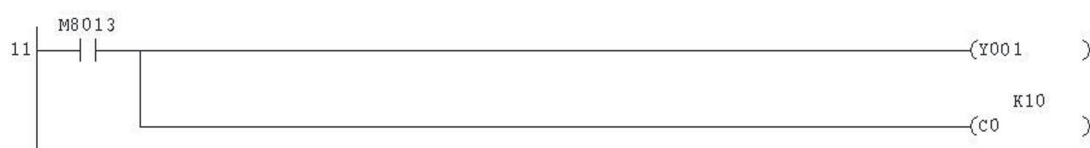
Press the  "branch down" button or, on the keyboard, the SHIFT key together with the function key F9 and press <Enter>. The display will be presented as follows:




- ② Move the cursor down one line and press the  "Output coil" button or press F7 on the keyboard. Enter C0 K10 and the display will be as follows:

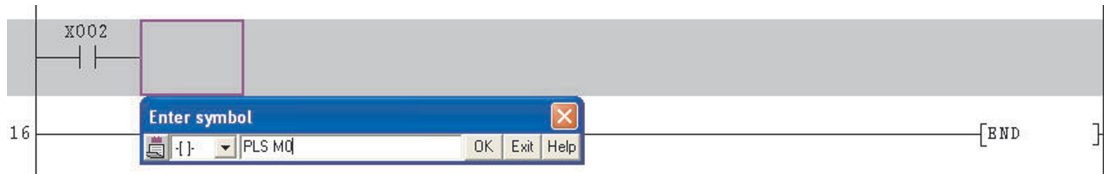




- ③ Press <Enter> to enter the coil and then press F4 or the  buttons to convert and the display will become thus:

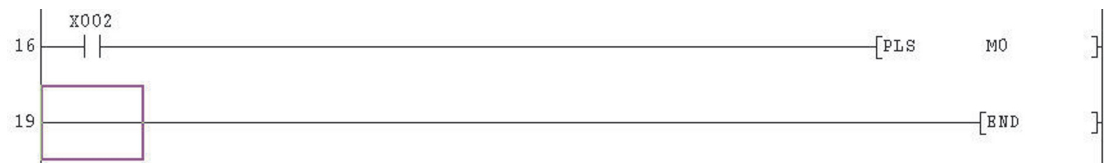


## 8.5 Appention of New Program Blocks

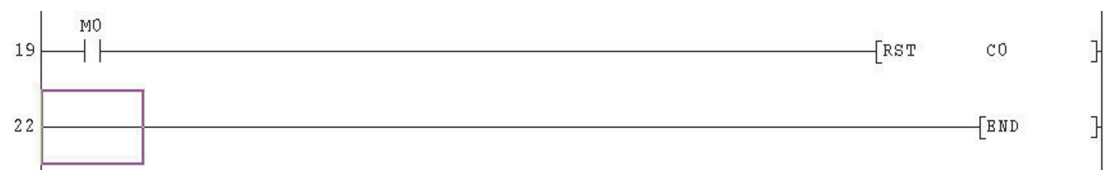
- ① With the cursor positioned on the start of line 16, select a normally open contact of X2. To enter the PLS M0 instruction, select  from the toolbar and enter PLS M0. The display will be as follows:



- ② Click **OK** or press <Enter> to complete the line. Press F4 or the   buttons to convert and the display will be thus:



- ③ Repeat the procedure ⑤ above for the next line with the RESET instruction for C0 (RST C0) and the display will be as shown:

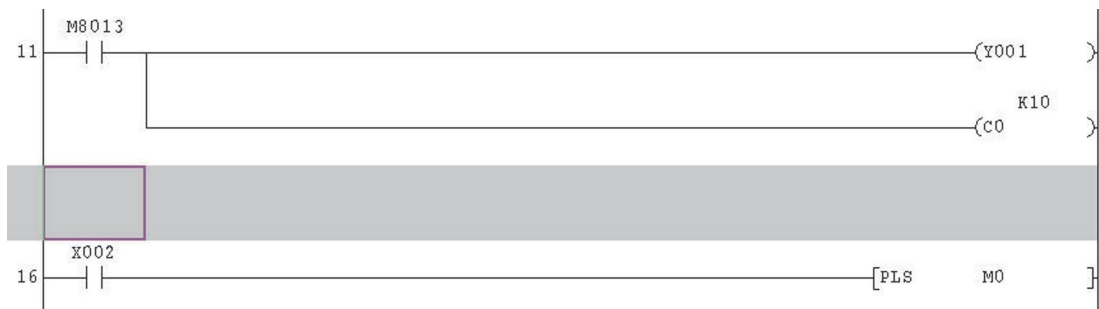


## 8.6 Insertion of New Program Blocks


The following two (red framed) further lines will be inserted following line11.



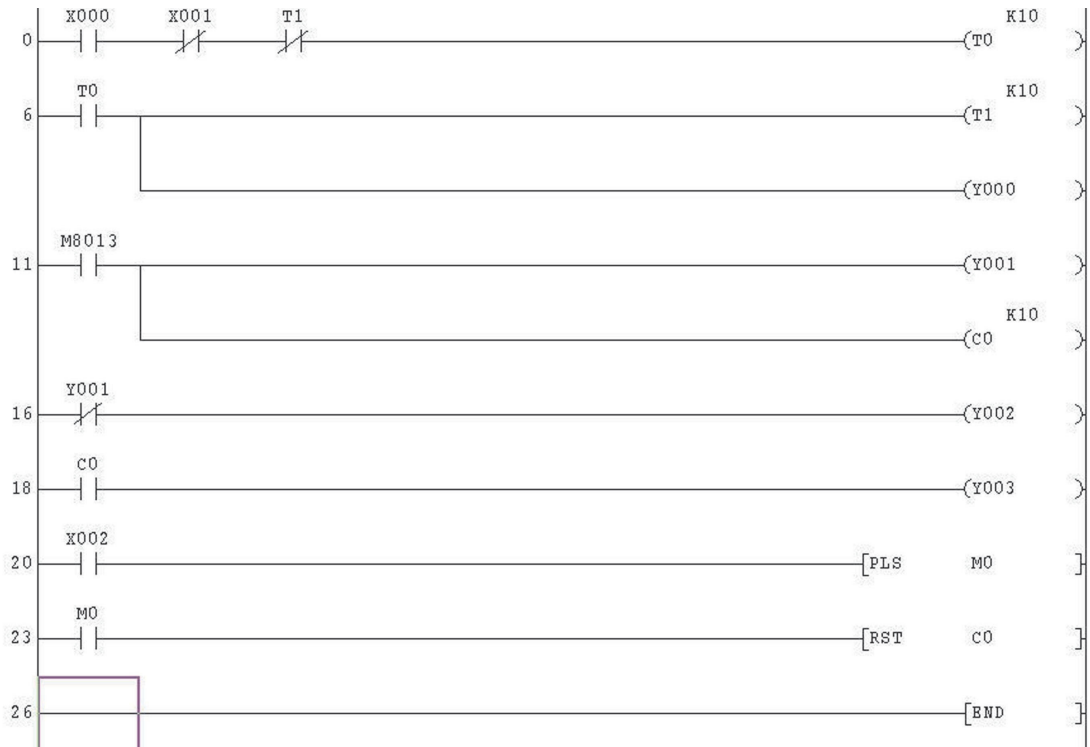
① With the cursor on the start of line 16, select **Insert Line** from the **Edit** menu thus:




② Enter the first new line as shown above and press F4 or the  buttons to convert.

③ Repeat the procedures in steps ① & ② above for the second new line. Then press F4 or one of the  buttons.

The final modified Ladder Diagram COMPACT\_PROG2 will now be as shown on the next page.



④ Save COMPACT\_PROG2 using the  button or select **Save** from the **Project** Menu.



# 9 Delete Functions

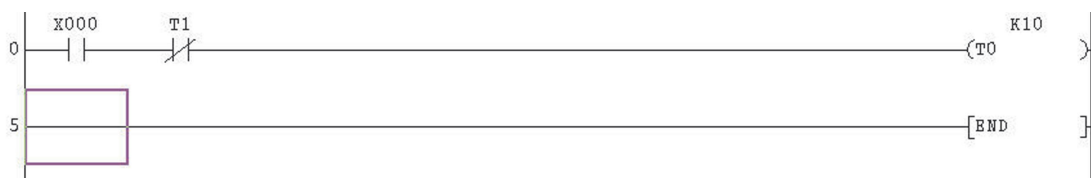
## 9.1 Overview

When modifying a Ladder Diagram, it may be necessary not only to make additions to the program but also to delete parts of it.

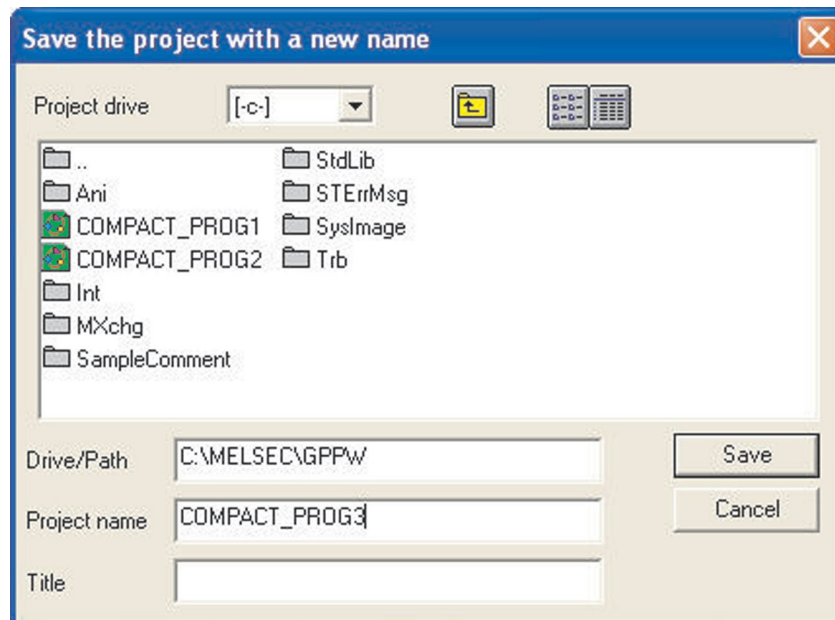
The project COMPACT\_PROG3 will be used to demonstrate how the following can be deleted:

- An input contact.
- Part of a line.
- A complete line.
- More than one line simultaneously.

After all of the delete modifications have been carried out, COMPACT\_PROG3 will appear as shown below:



Before carrying out further modifications, save COMPACT\_PROG2 to COMPACT\_PROG3, using the **Save as** procedure described previously:

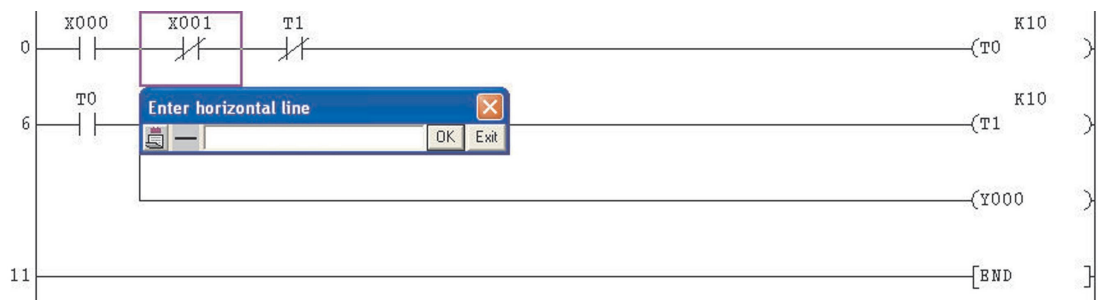




## 9.2 Deleting an Input Contact

Ensure the project COMPACT\_PROG3\* is displayed and in overwrite mode

\* NB: At this moment in time COMPACT\_PROG3 will be identical to COMPACT\_PROG2.

- ① Move the cursor to the normally closed X1 contact.
- ② Select the horizontal line i.e. function key F9 to delete the X1 contact



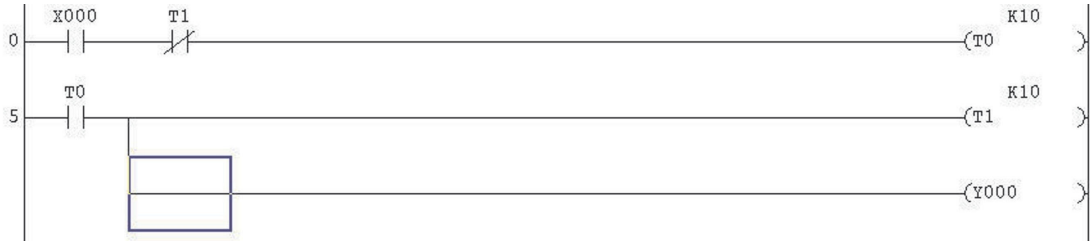
- ③ Select **OK** and the X1 contact will be deleted.
- ④ Press F4 or the   buttons to convert the modification, the display will be:



### 9.3 Deleting a Branch

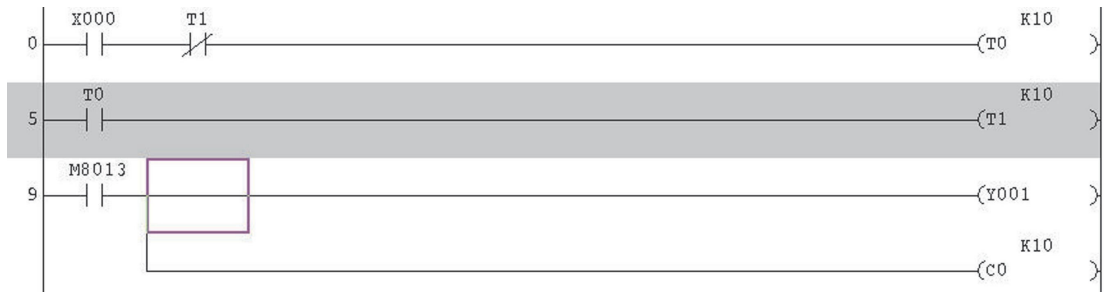
The branch at current line 5 will now be deleted.

- ① Move the cursor to the branch at line 5 as shown below.



- ② From the **Edit** menu, select **Delete Line** or use the shortcut keys "Shift+Delete" together.

- ③ The display will become:



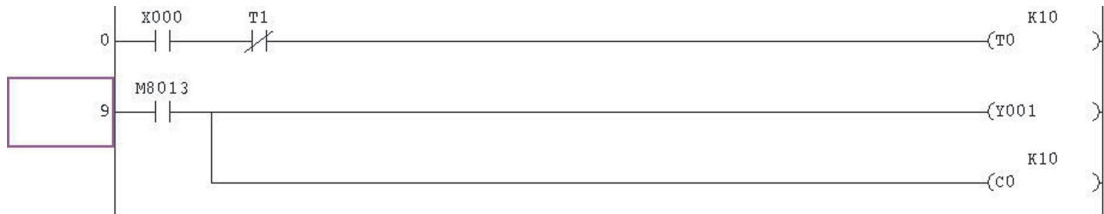
- ④ Press F4 or one of the buttons  to convert the changes:





## 9.4 Deleting a Single Line

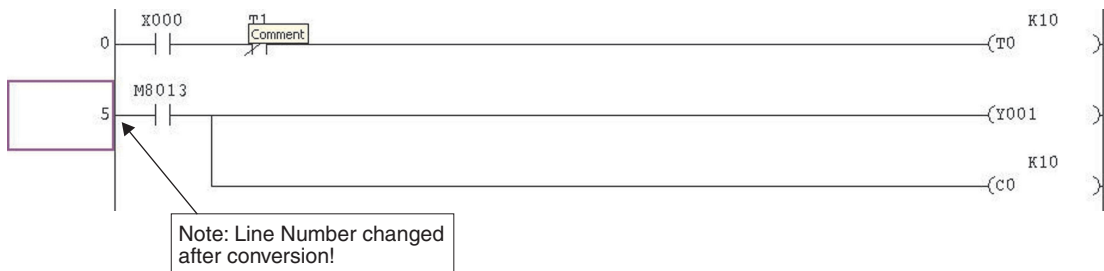
The single line at Line 5 will now be deleted.

- ① Move the cursor to the start of line 5 (Right hand side of ladder rung). Select **Edit** and then **Delete line** or you may find it easier to use "Shift+Delete" keys together. The line will be deleted immediately and the display will be as shown below:



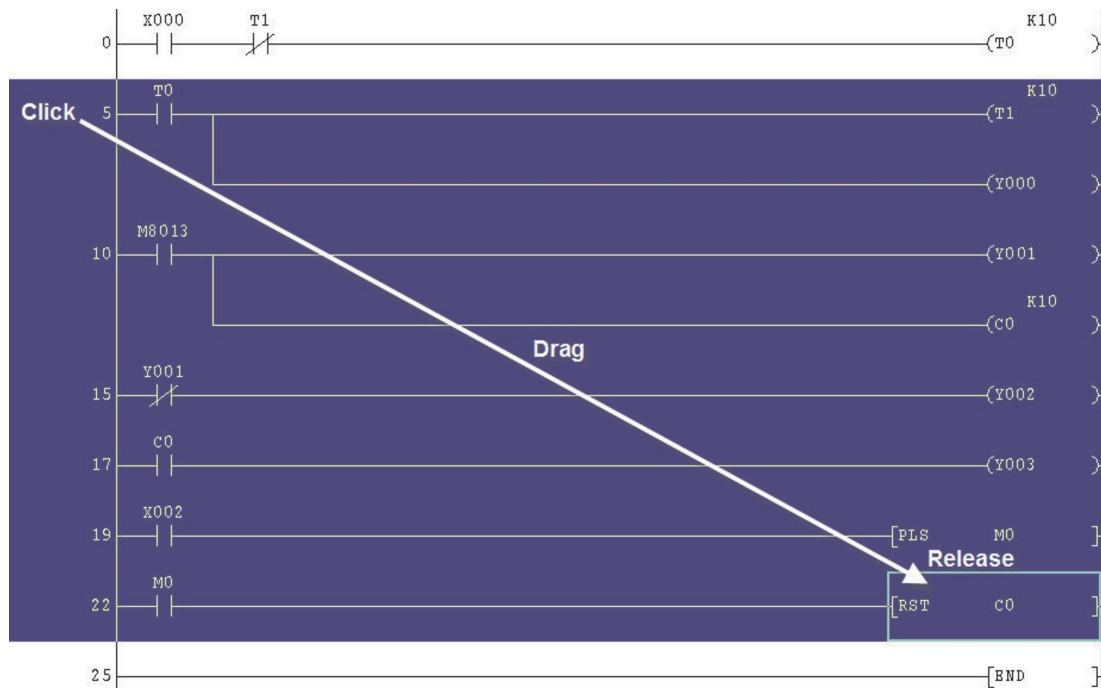
**Important:** You **MUST** remember to press F4 or click the buttons   to convert the changes following a **line deletion**. In this case, GX Developer gives **no** indication that there **has been a change** to the code because the changed code has been deleted!

Once converted, note line number changes:

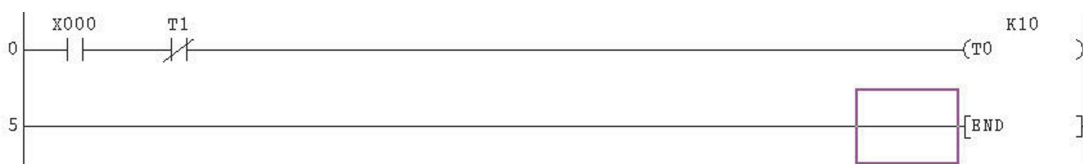



## 9.5 Deleting multiple lines

- Click and hold the left hand mouse button **to the left hand side of the rung** on line 5. While holding the left hand button down, drag the mouse diagonally to the right and down until over the RST C0 function on the far right of line 22. Release the mouse button as shown below:



- From the Edit menu, press the "DEL" key on the keyboard. All of the selected instructions will be deleted and the display will be thus:



- Finally save the file using the  button.



# 10 Program Documentation

Perhaps one of the most commonly encountered difficulties for maintenance engineers and technicians working on plant is often the total lack of adequately documented PLC program listings.

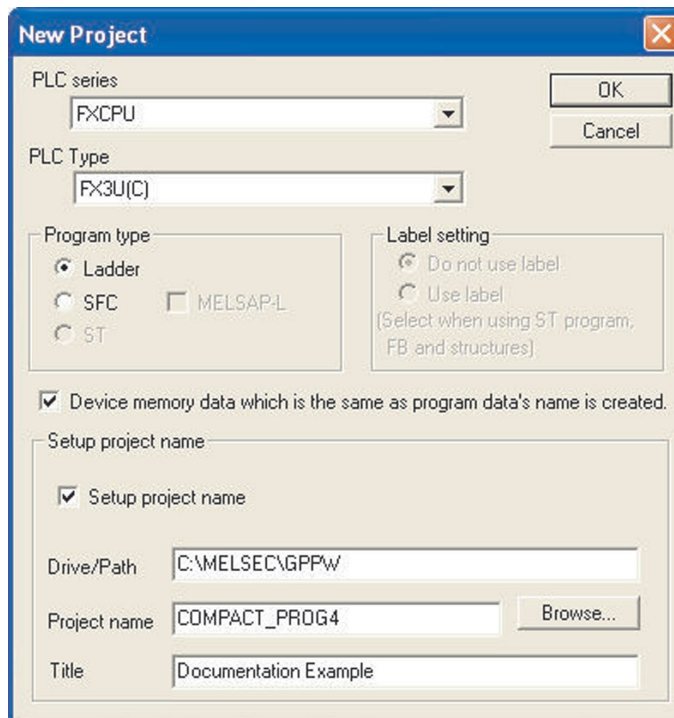
There is really no excuse for poorly documented programs; most PLC programming software provides extensive facilities for the annotation of software. Poorly documented software is totally unacceptable in any situation! Documentation is necessary in order for the program originator to convey programming methods, structures and layouts used within the code to others who may have to perform maintenance or modification tasks.

GX Developer offers a wide range of documentation tools to enable the code to be fully readable and decipherable by other programmers, maintenance engineers or various third parties who may become for example, involved with the operation, fault finding or maintenance of a particular system.

## 10.1 New Program Example: COMPACT\_PROG4

A new program COMPACT\_PROG4 will be constructed in order to demonstrate the use of the documentation and annotation tools provided in GX Developer.

- ① From the **Project** menu, select **New Project** or simply press the  Key. The display will be as shown:



Note the addition of the program title “Documentation Example” in the title field.





## 10.2 Annotating the Program

### General Points

The following section deals with the various methods and facilities offered in GX Developer for program annotation. Before proceeding with the description of these procedures, it will be necessary to clarify one or two points regarding options for the embedding of 'Statements' and 'Notes' into the source code and the downloading of annotation elements into the PLC CPU with the program.

### Differences

The following settings differ depending on the PLC series selected.

### Statements/Notes

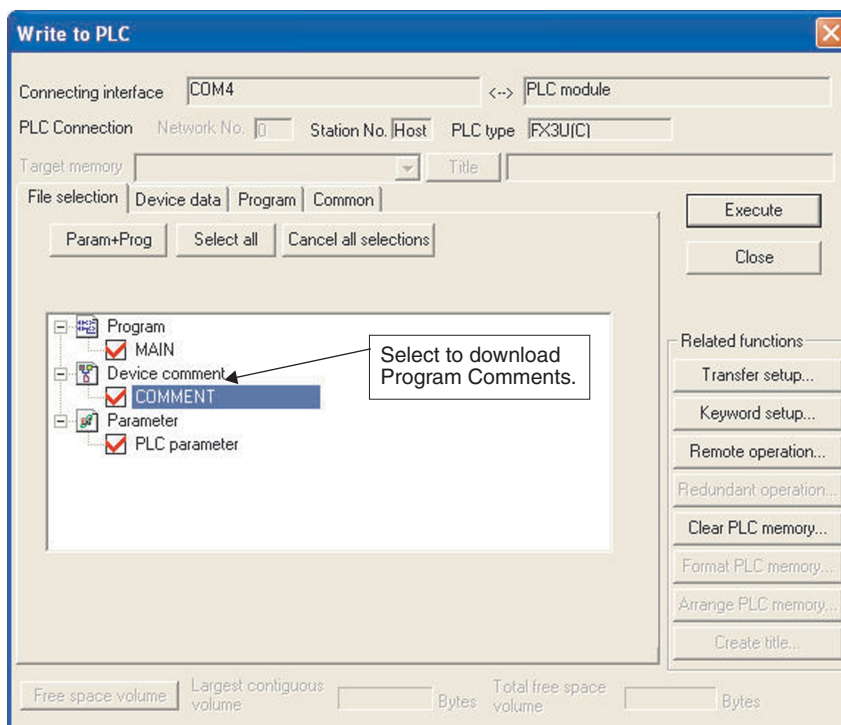
**Separate** means that statements and notes are stored in the project directory. When you download a program from the PLC this information will only be displayed if there is a corresponding project containing this data on the PC.



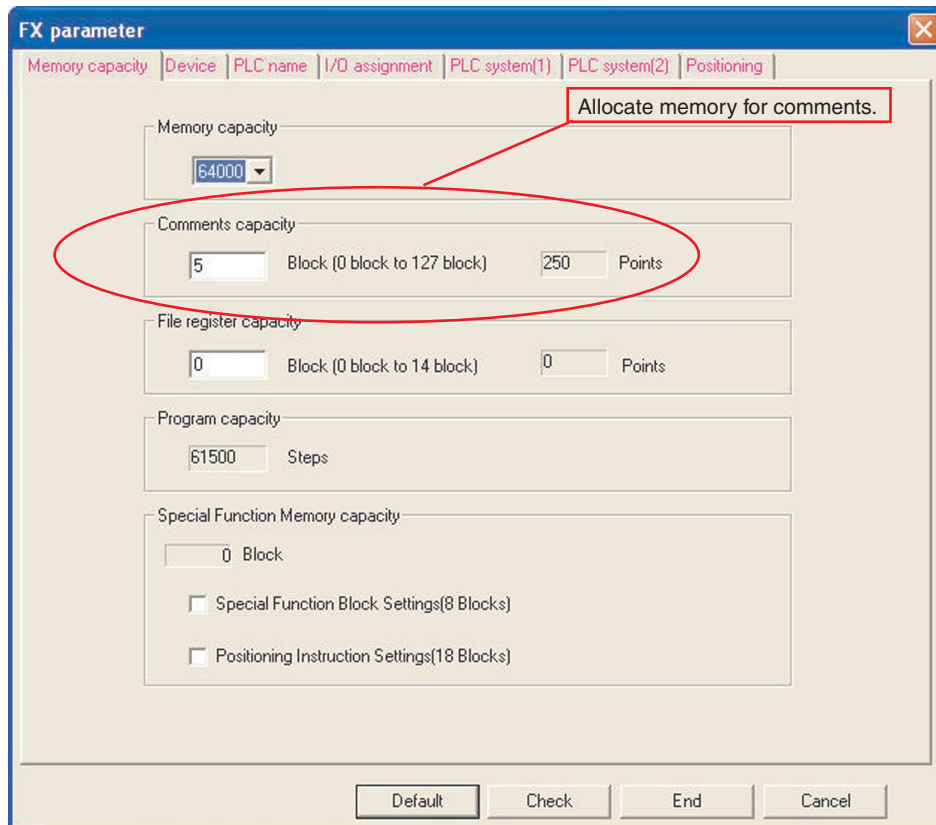
Note that you cannot store statements and notes in the PLCs of the MELSEC FX family. This is why the **Embedded** option is always disabled when you are working with these PLCs. (With **Embedded** selected, the Statement/Note is embedded into the program source code and will be sent to the PLC on download. This is the default setting for Q-Series PLC's.)

### Comments

May be sent to the PLC with the program source code when selected in the transfer options menu:



If comments are to be sent to the PLC CPU then an area of memory must be pre-allocated from the PLC Parameter editor thus:



This reserved memory is then no longer available for program code.

**Annotation Toolbar Buttons:**

Three buttons are used to select different annotation options:



These buttons are used while the program is in 'Write' mode and operate with a toggle action – click to activate – click again to deactivate.

## 10.3 Comments

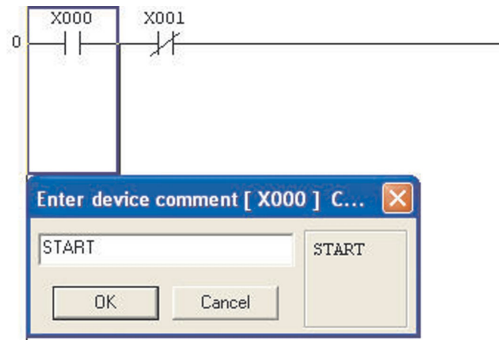
### Direct onscreen method

You can enter comments direct while programming.

- ① With the program COMPACT\_PROG4 displayed on the screen select the Comments mode button:



For example, to place a comment against the device X0, position the cursor over the X0 contact and press 'enter' or double click the mouse over the contact. The screen shown on the right is displayed:



- ② Enter the comment "START" in the text box and press enter or click **OK**.
- ③ Move the cursor to X1 and press <enter> or double click the mouse over X1. Repeat for the output Y0 and enter the comments as shown below:



Note that all occurrences of the devices X0, X1 and Y0 will be displayed automatically within the program with the attached comment.

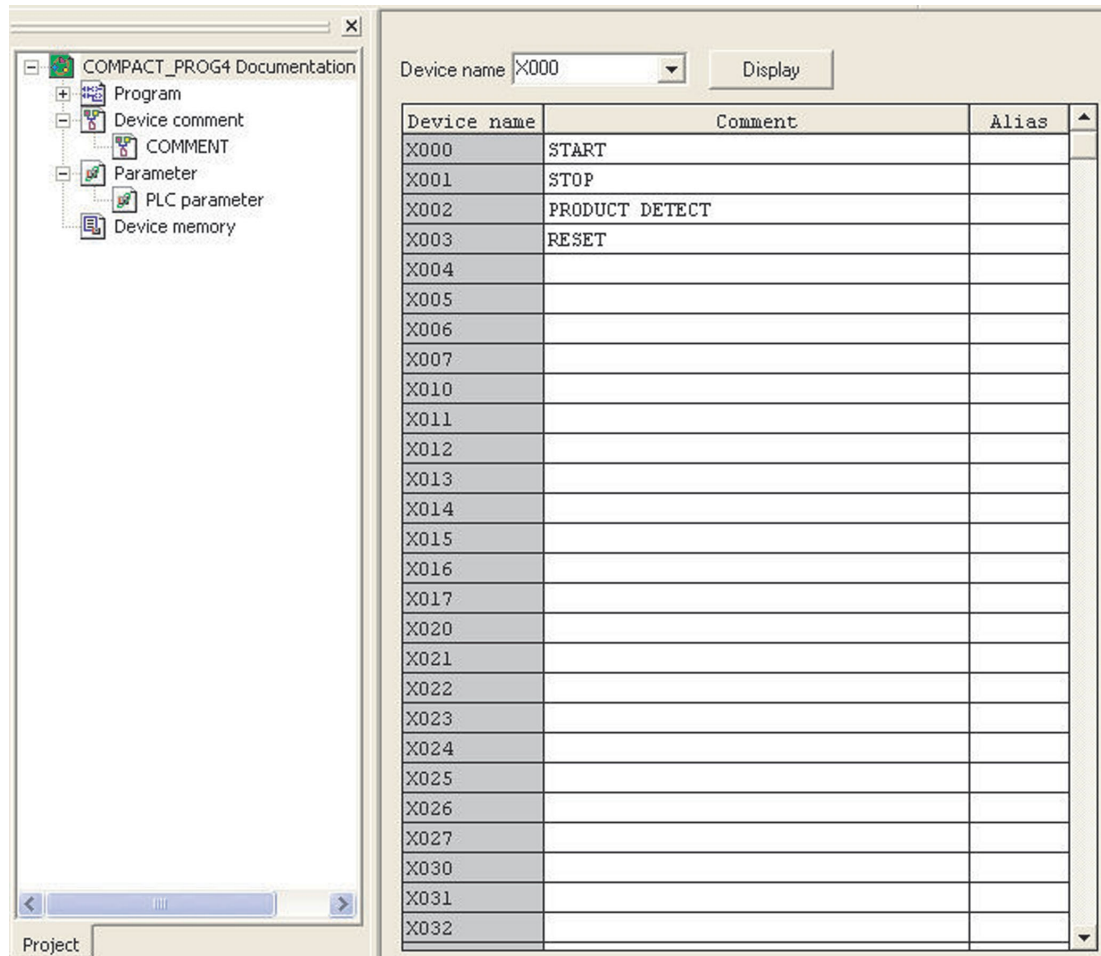
## 10.4 Project Data List (Navigation Window)

### Comment input, table entry method.

Comments can also be input using the table entry method.

Where batches of devices are to be commented, for example all inputs or outputs, it is preferable to be able to enter the comments into a table. GX Developer offers this method of data entry through the **Device Comment** file option on the navigation window.

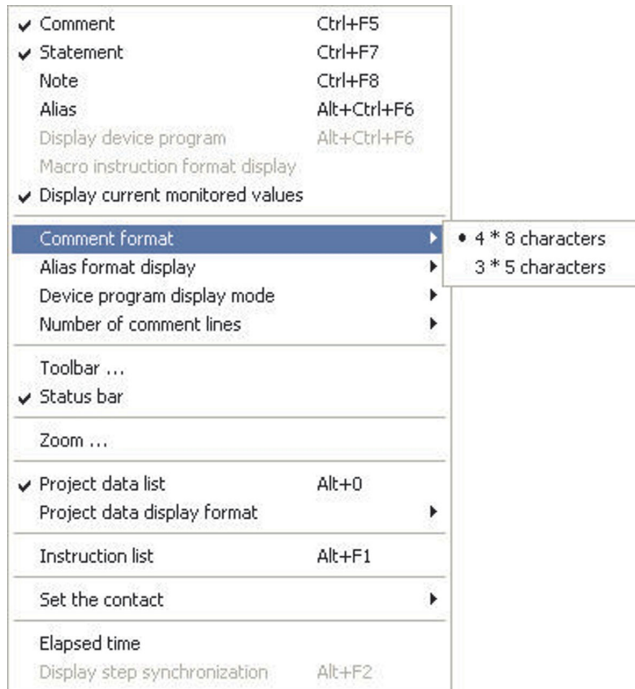
To enter comments into table, double click on the **Device comment** folder in the **Project Data List** window:



## 10.5 Comment Format

**NOTE**

GX Developer will word wrap the text to a preset format as set in the **Comment format** function from within the **View** menu:

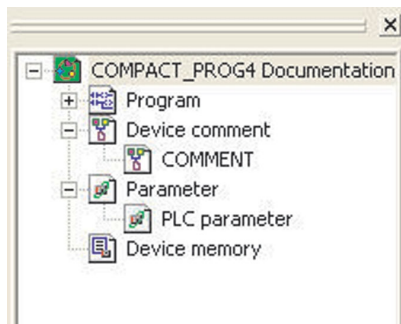


The default format is 4 lines of 8 characters which can be altered using the above menu and advanced system settings which are described later-on and in the advanced course notes.


**NOTE**

When entering the text in the prompt window, care should be taken to ensure that appropriate padding (spacing) be manually placed in the string so that the comment is displayed correctly. Remember GX Developer automatically “Word Wraps” the text to the preset format.

Return to the main ladder editor by double clicking on the **Main** file selection using the Project Data List window on the left of the screen thus:

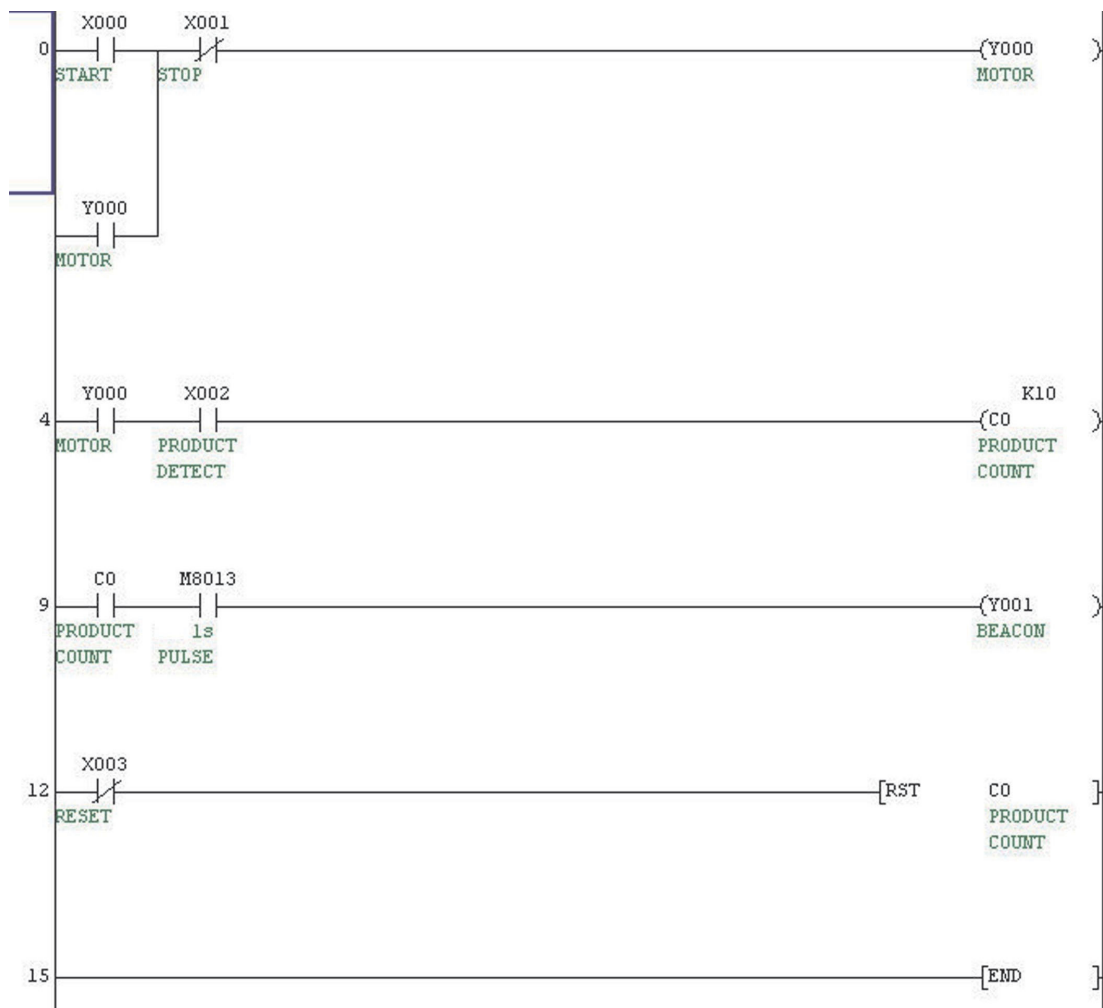


It is convenient to use this Project Data List window in-future to manoeuvre around between displays and editors.

Remember that this window may be turned on or off with the  button or tick/un-tick the **Project Data List** selection under the **View** menu.


✓ Comment	Ctrl+F5
✓ Statement	Ctrl+F7
Note	Ctrl+F8
Alias	Alt+Ctrl+F6
Display device program	Alt+Ctrl+F6
Macro instruction format display	
✓ Display current monitored values	
Comment format	▶
Alias format display	▶
Device program display mode	▶
Number of comment lines	▶
Toolbar ...	
✓ Status bar	
Zoom ...	
✓ Project data list	Alt+0
Project data display format	▶
Instruction list	Alt+F1
Set the contact	▶
Elapsed time	
Display step synchronization	Alt+F2

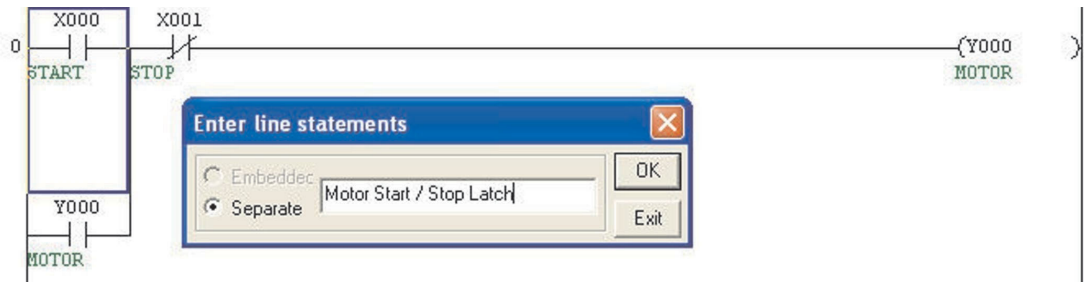
Complete the commenting of the ladder program as follows:




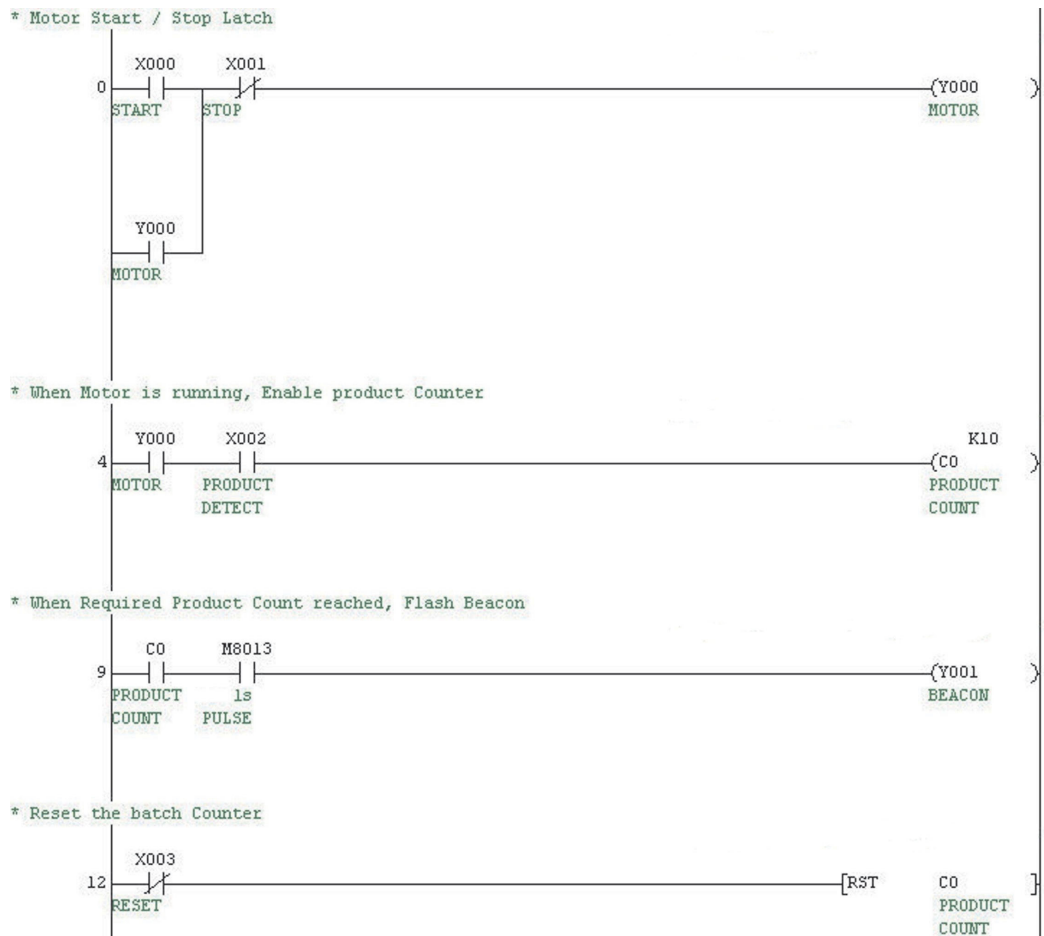
## 10.6 Statements

Statements enable detailed descriptions to be added above the program blocks in order to describe the operation or functionality. Statements can also be used to provide an overall description or title to the program or a routine.

- ① With the program COMPACT\_PROG4 displayed on the screen select the Statement mode button: 
- ② Position the cursor anywhere on the program block (segment) to which the statement is to be attached. Press 'enter' or double click the mouse over the program block.
- ③ Enter the statement text into the prompt box:




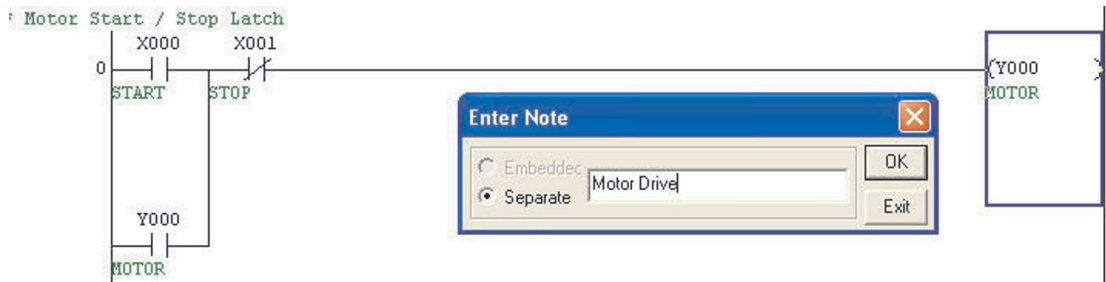
- ④ Once the statement has been entered, it will be necessary to press F4 or click the buttons  to convert the changes to the source code of the program.
- ⑤ Place statements on the ladder program as follows:



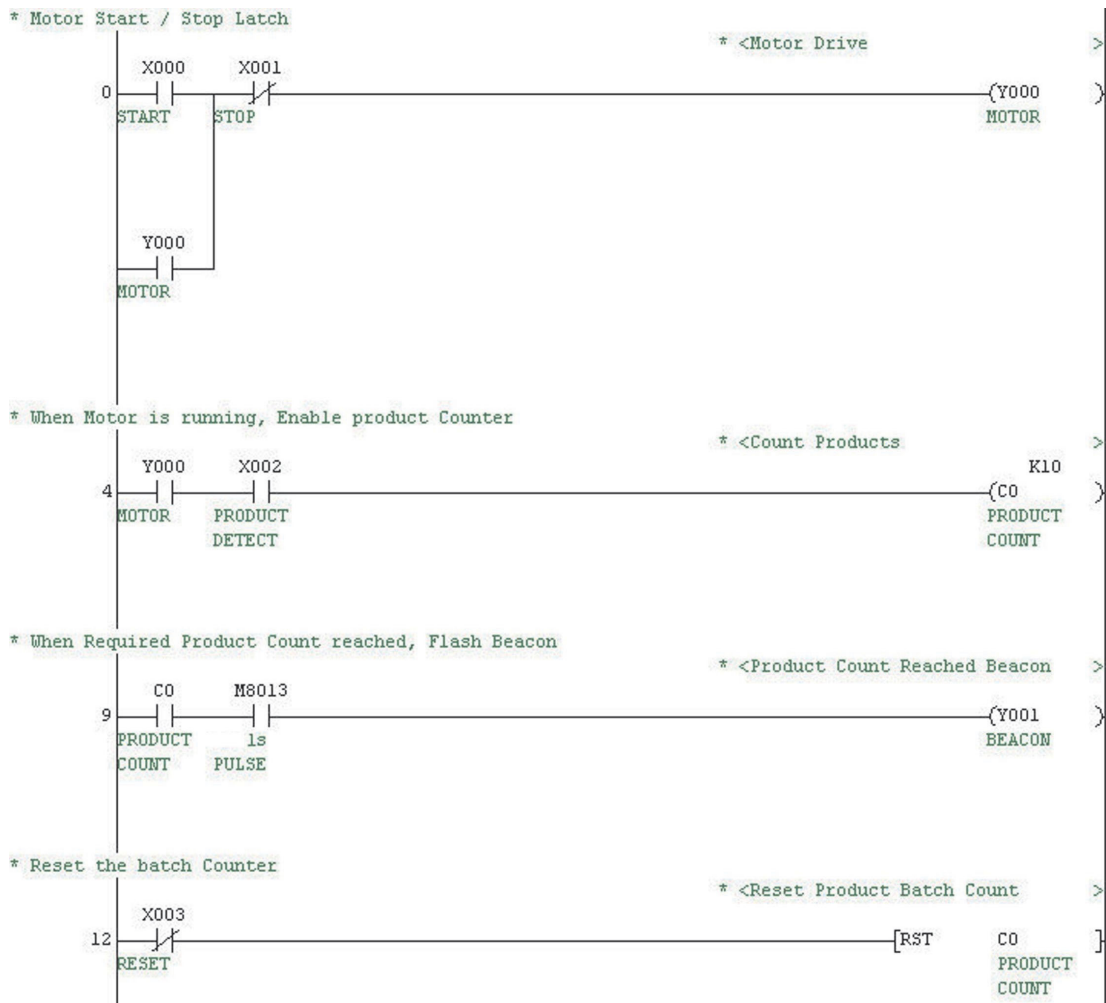
## 10.7 Notes

Notes enable textual descriptions to be added to the output functions of a ladder program. This helps describe the operation of individual output lines or functions in the program. 'Notes' are justified to the right hand of the ladder program layout.

- ① With the program COMPACT\_PROG4 displayed on the screen select the Note Entry mode button: 
- ② Position the cursor over the output coil or function in the program block (Segment) to which the 'Note' is to be attached. Press <Enter> or double click the mouse over the program block.
- ③ Enter the 'Note' text into the prompt box:



- ④ Complete the ladder diagram as follows:



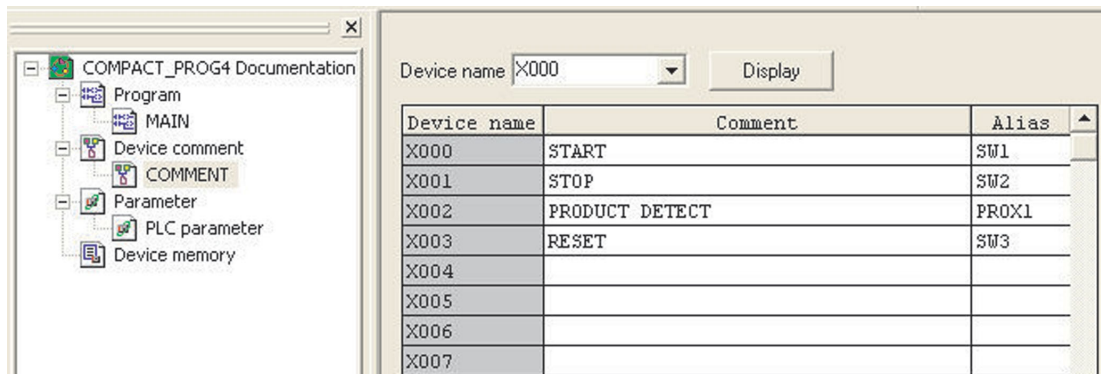


## 10.8 Alias

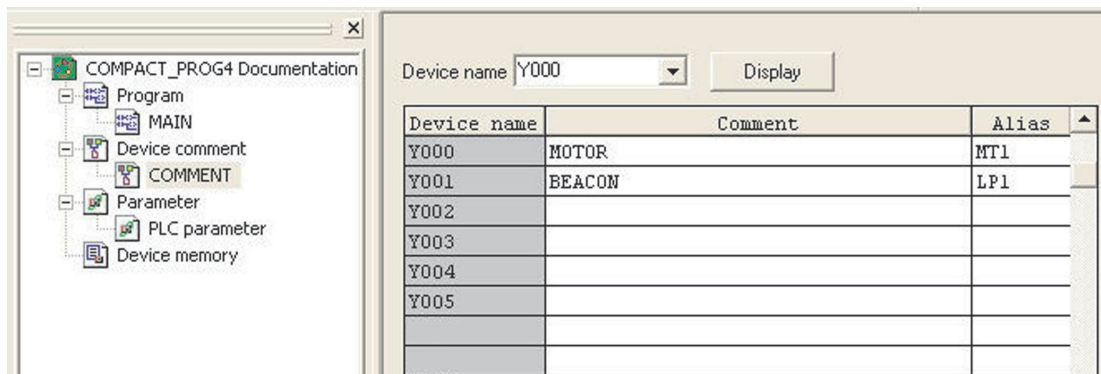
An Alias provides a method of cross-referencing PLC I/O numbers that are connected physically to external system devices. For example; Input X0 may be connected to a Start Button on a machine, whose external circuit diagram device reference is SW1. SW1 may be listed as the Alias to X0 in the Comment list, so as to provide a qualified link with the PLC program listing.

### Example:

- ① From the Project data list window, open the comment list.
- ② Double click the mouse over the alias column for X0 and enter the text “SW1”.
- ③ Repeat this action for the remainder of X devices as shown:



- ④ Change the display reference to show devices beginning with Y0. Repeat the action in ③ above for Y0 & Y1 with data as shown:

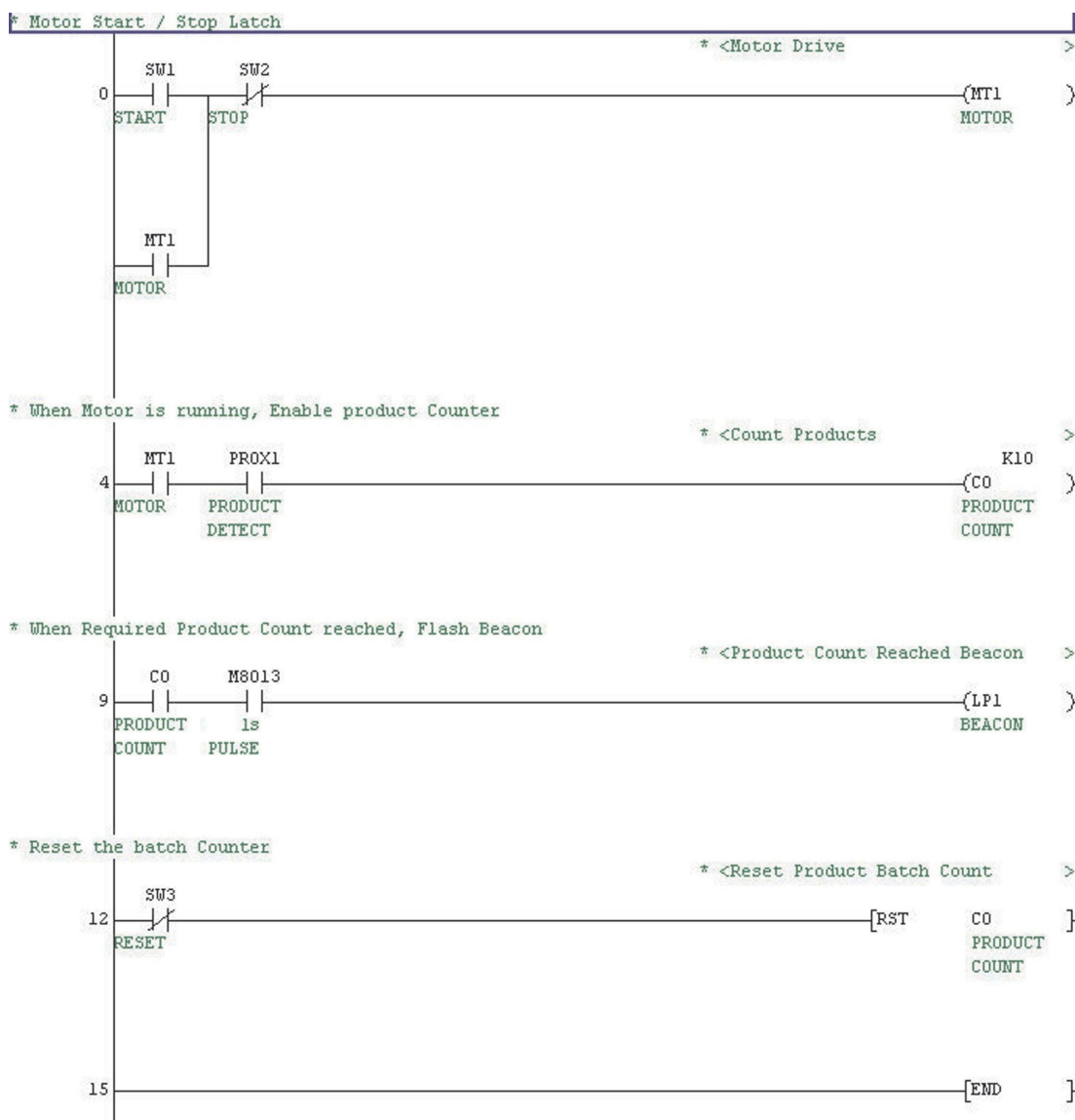


- ⑤ In the Project Data List Window, Click **Program** and **Main** to return to the Ladder display.

✓ Comment	Ctrl+F5
✓ Statement	Ctrl+F7
✓ Note	Ctrl+F8
Alias	Alt+Ctrl+F6
Display device program	Alt+Ctrl+F6
Macro instruction format display	
✓ Display current monitored values	
Comment format	▶
Alias format display	▶
Device program display mode	▶
Number of comment lines	▶

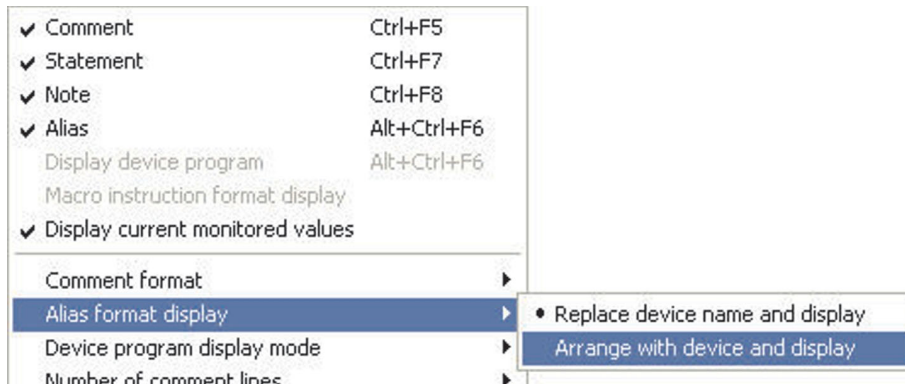
⑥ From the **View** menu, click **Alias**.

The display will be as shown

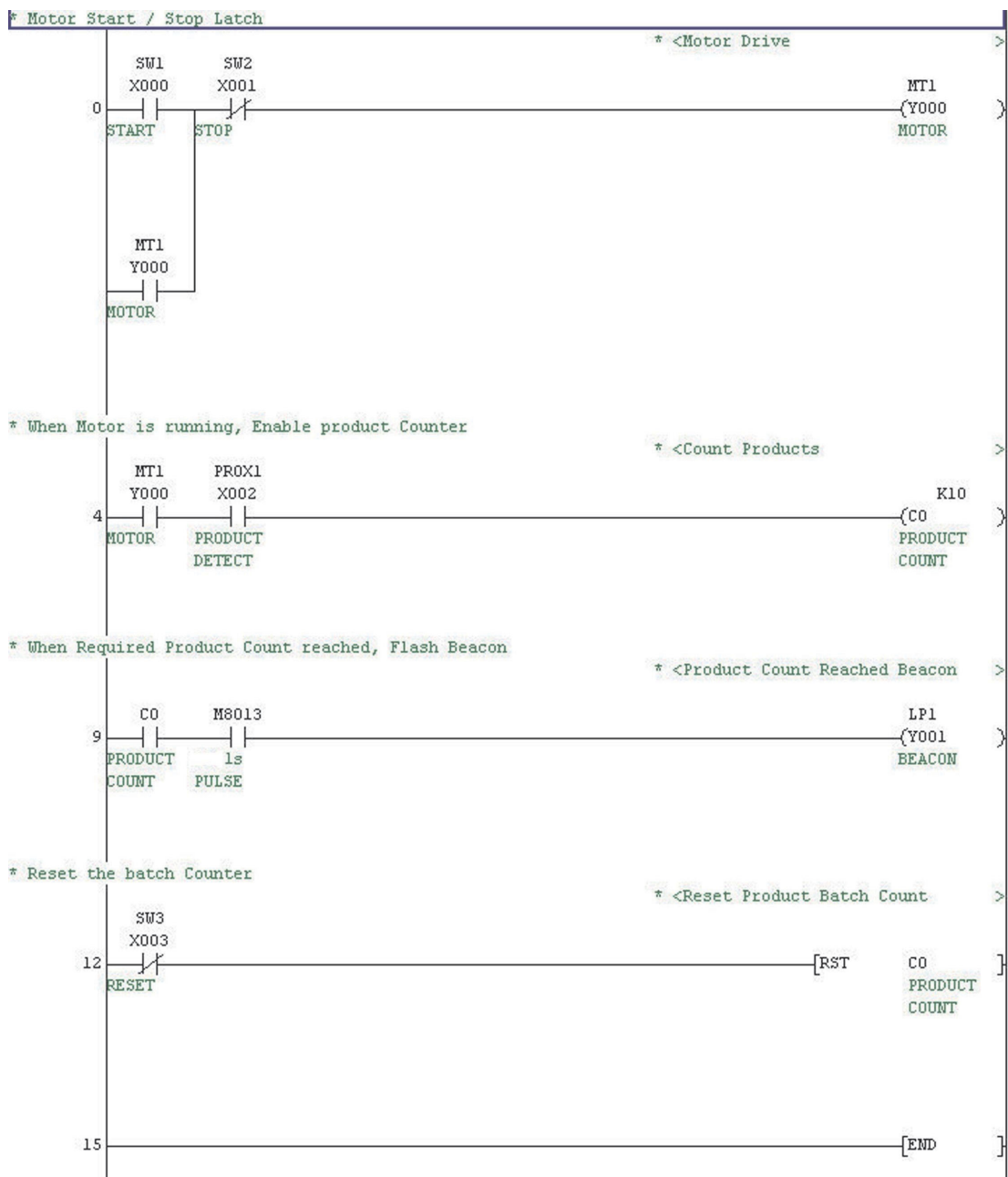


As can be seen, the display has replaced the device names with the alias where appropriate.

If desired, the alias and the device names may be displayed together. To achieve this, click the **View** menu and select **Alias Format Display**. Select **Arrange With Device And Display** thus:



In which case, the Ladder display becomes:



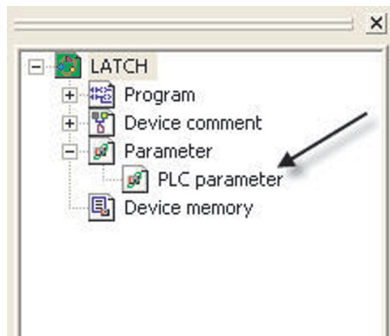


# 11 I/O Assignment

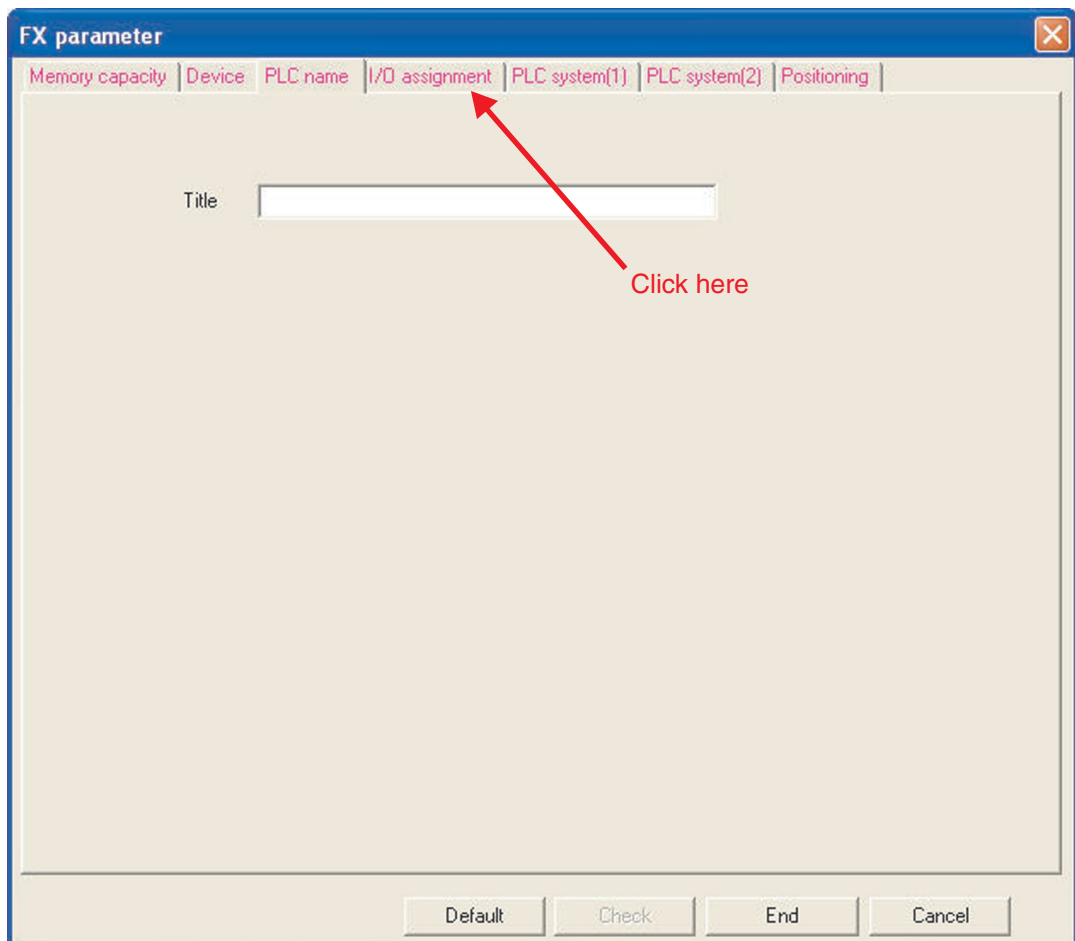
## 11.1 Checking the Input/Output Range

The available input/output range of the used FX PLC can be checked in the PLC parameter.

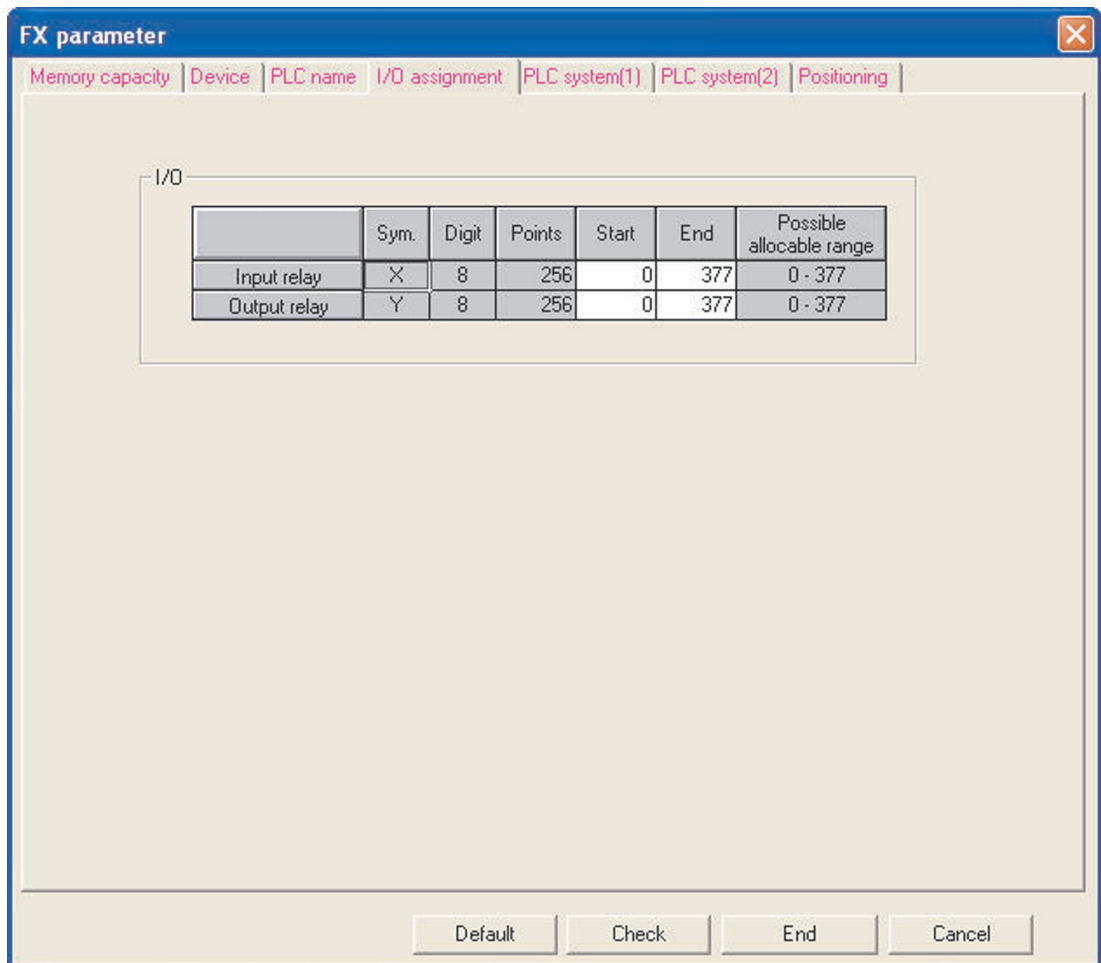
- ① From the **Project Data List** window, open the **Parameter** option by double clicking on the folder and file icons:



- ② Once open, click on the 'I/O Assignment' tab at the top of the Window:



- ③ The possible allocable range is shown in the right hand column. Depending on the actual used configuration, the start and the end of the ranges can be altered. Use the **Check** button to check the edited ranges afterwards.



# 12 Downloading a Project to a PLC

## 12.1 Communications Setup

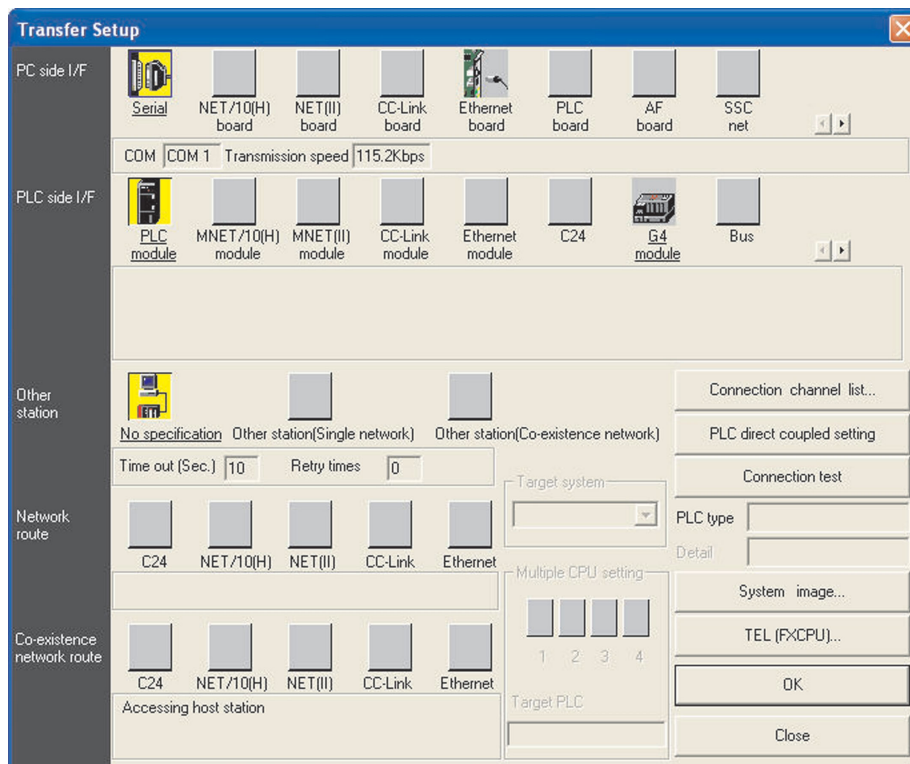
The following notes describe how the project COMPACT\_PROG4 is downloaded to a FX PLC.

Before you can transfer your program to the PLC the PLC must be connected to the programming device and the controller's power supply must be switched on.

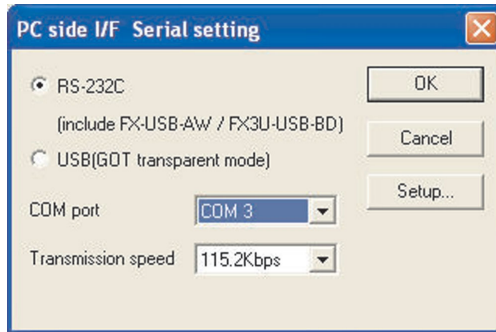
① From **Online** Menu, Select **Transfer Setup**:



The following window will be displayed:

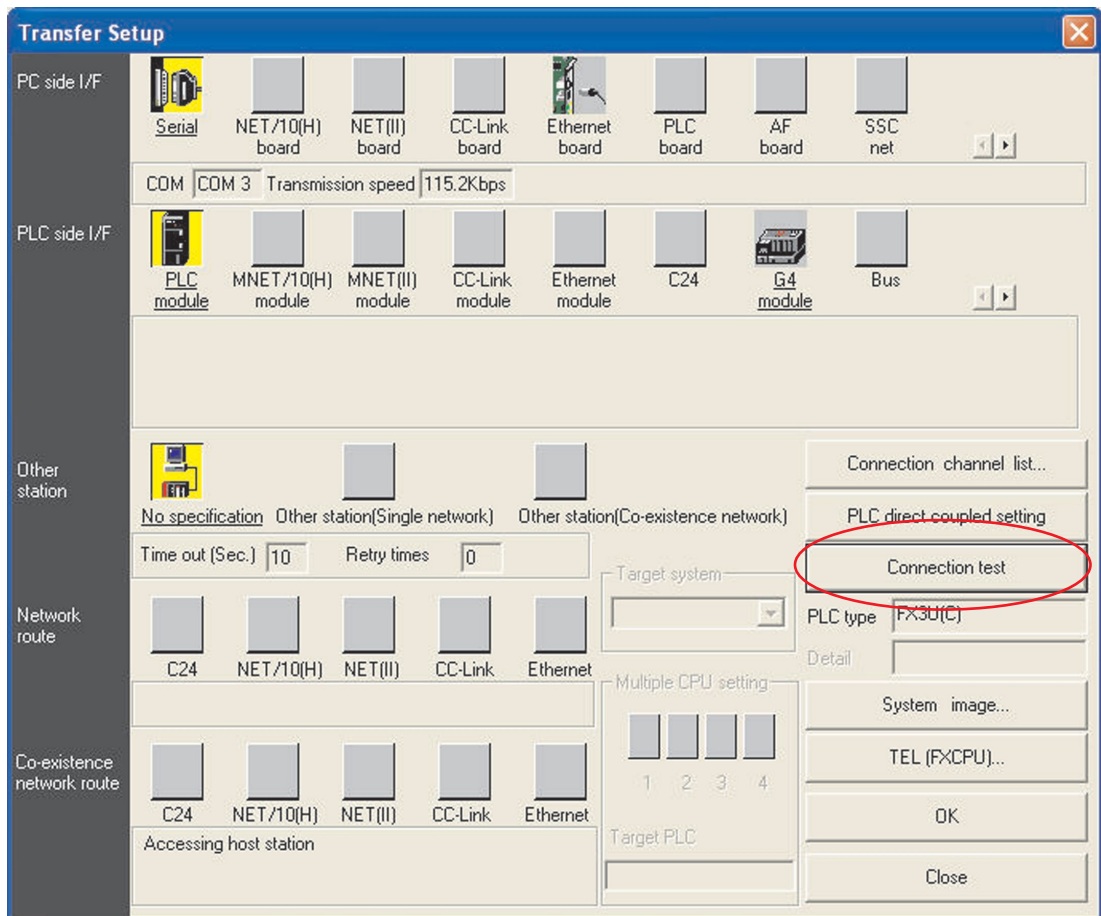


- ② Double click the mouse on the yellow **PC side I/F – Serial** Button and the following dialogue window is displayed:



- ③ Select **RS-232C** and an appropriate COM port as shown and click **OK**.

- ④ Click on the **Connection Test** button to check PC-PLC communications are ok:



- ⑤ The following message should be displayed:



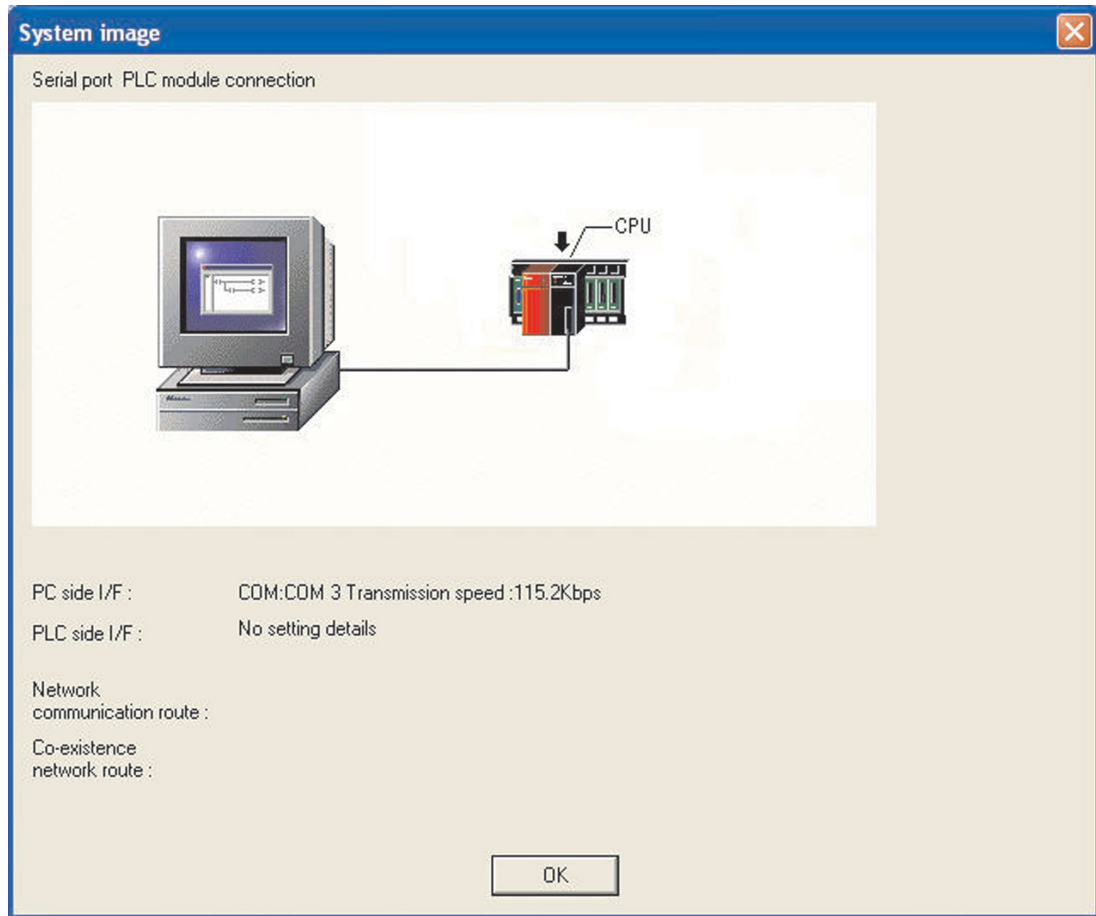
- ⑥ Click **OK** to close this message.

If an error message is displayed, check connections and settings with the PLC.



### 12.1.1 Connection Setup Route

- ① To obtain a pictorial view of the connection setup route, select the **System image** button



**NOTE**

When using a standard RS232 Serial Port to communicate with the PLC, if another device is already connected to the selected COM (n) interface, for example a serial mouse; Select another free serial port.

- ② Select **OK** to close the **System image** display and return to the **Connection setup** display.
- ③ Then click the **OK** button to close the **Connection Setup** window. If you leave the **Connection Setup** window using the **Close** button, the settings are not saved.

## 12.2 Clearing of the PLC Memory

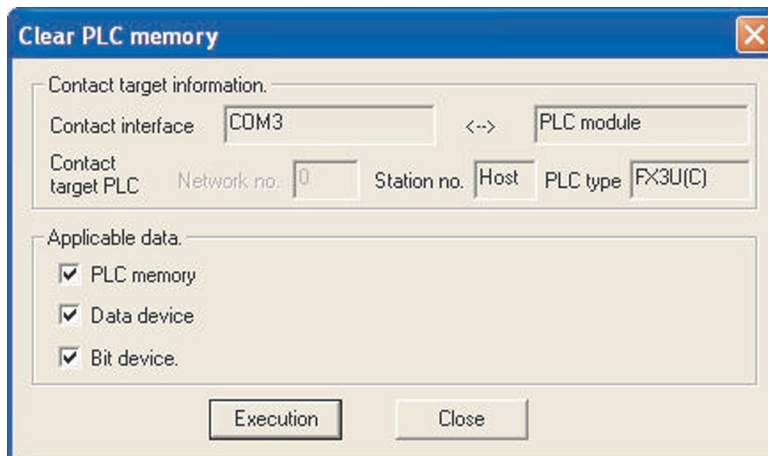
It is always sensible to clear the PLC memory before use. This is especially important when the main unit is to be re-used. This ensures that no other programs are present in the PLC before writing new code to the memory.

**Procedure:**

- ① Select **Clear PLC Memory** from the **Online Menu**:

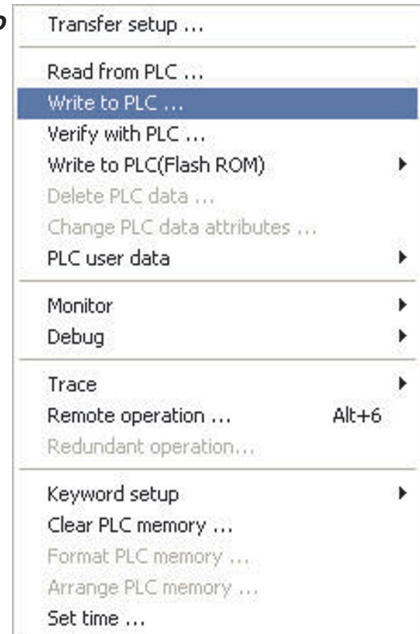


- ② When the following window is displayed, click **Execution**:



## 12.3 Write Program to PLC

- ① From the Main menu, select **Online** and than **Write to PLC**.

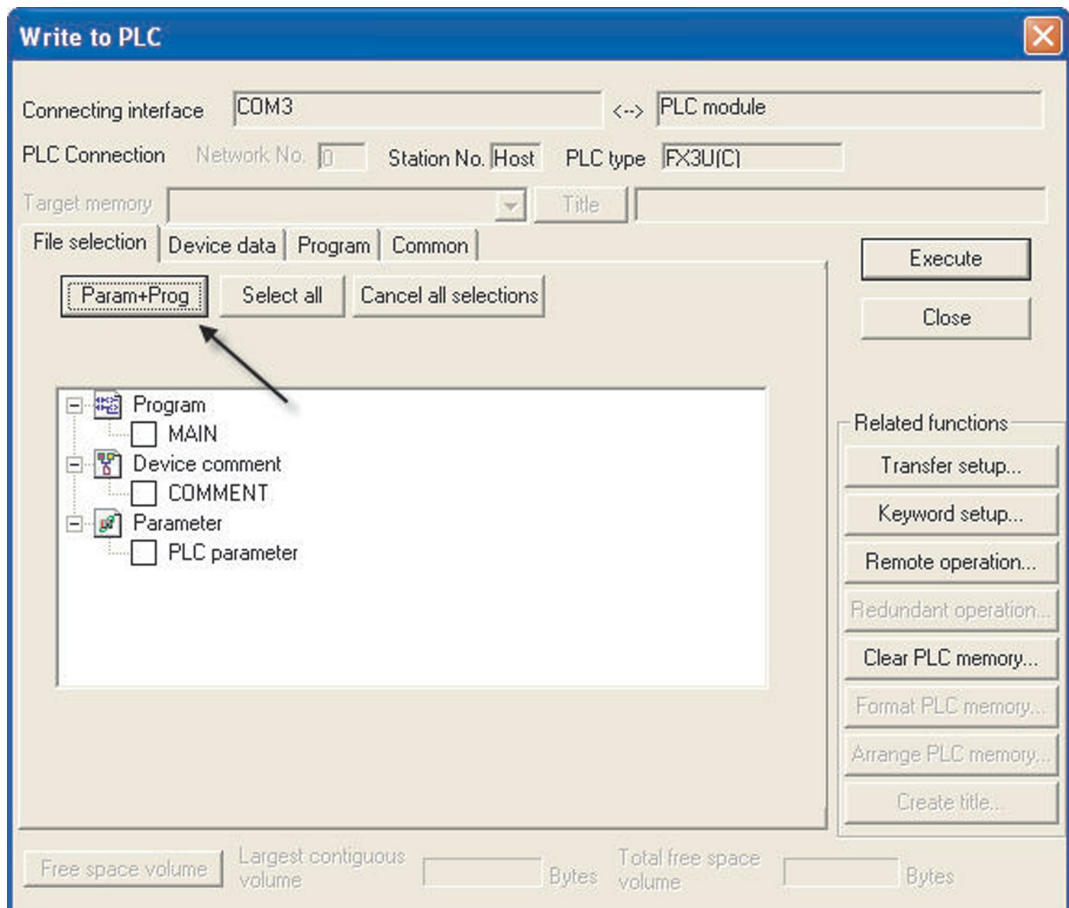


Alternatively, click on the  button from the tool bar.

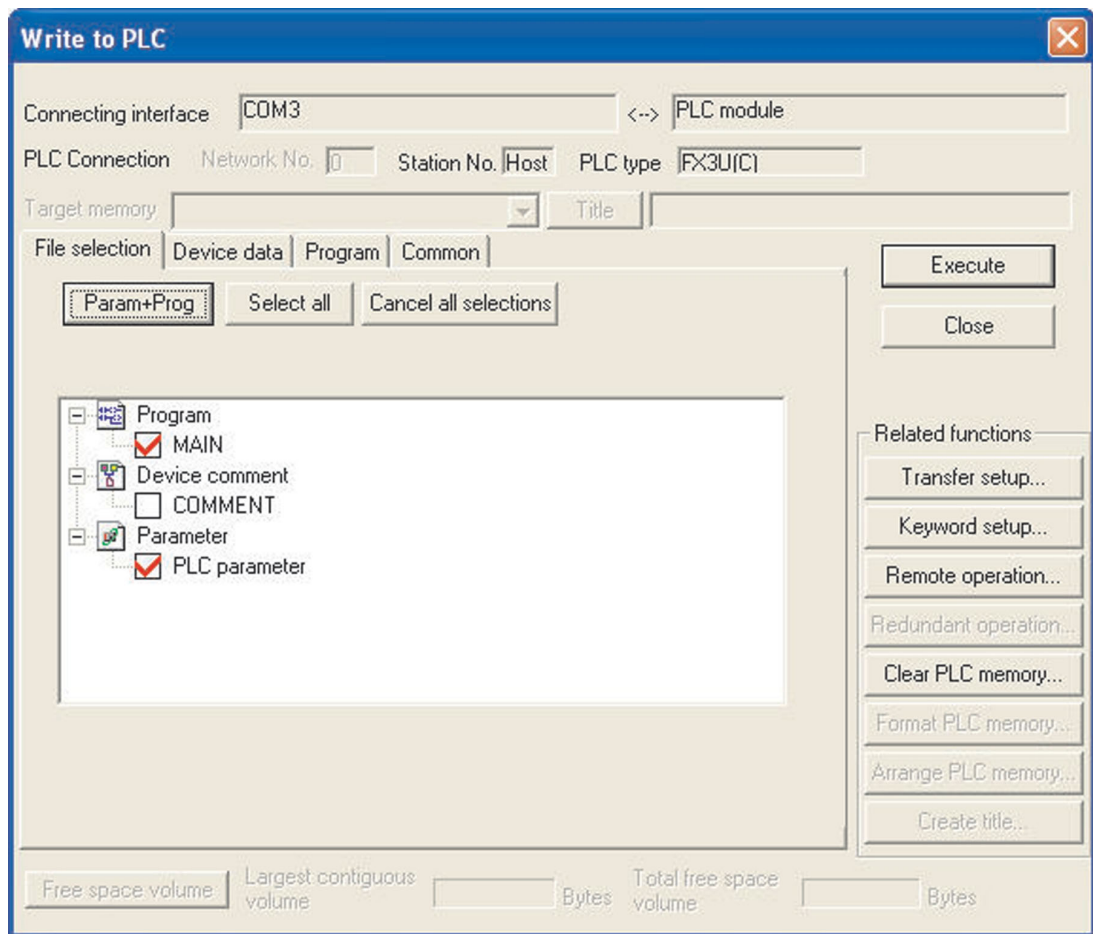
**TIP**

| Get into the habit of using the tool buttons, they save a great deal of time!

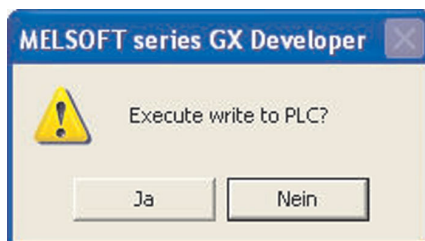
The display will be as show below:



- ② Select the **Param+Prog** button on the display to enable the program and parameters for the project COMPACT\_PROG4 to be downloaded:

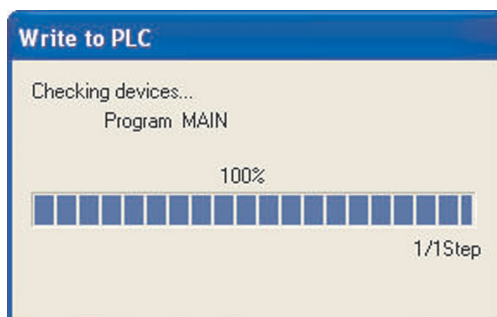


- ③ Select **Execute** and the following dialogue window is displayed:



- ④ Select **Yes** and the Parameters and the Main program will be downloaded to the PLC.

During program transfer, the progress is displayed on the screen:



When transfer is complete, the following message is displayed:



⑤ Click **OK** to clear the dialogue box.

Program transfer has now successfully completed.

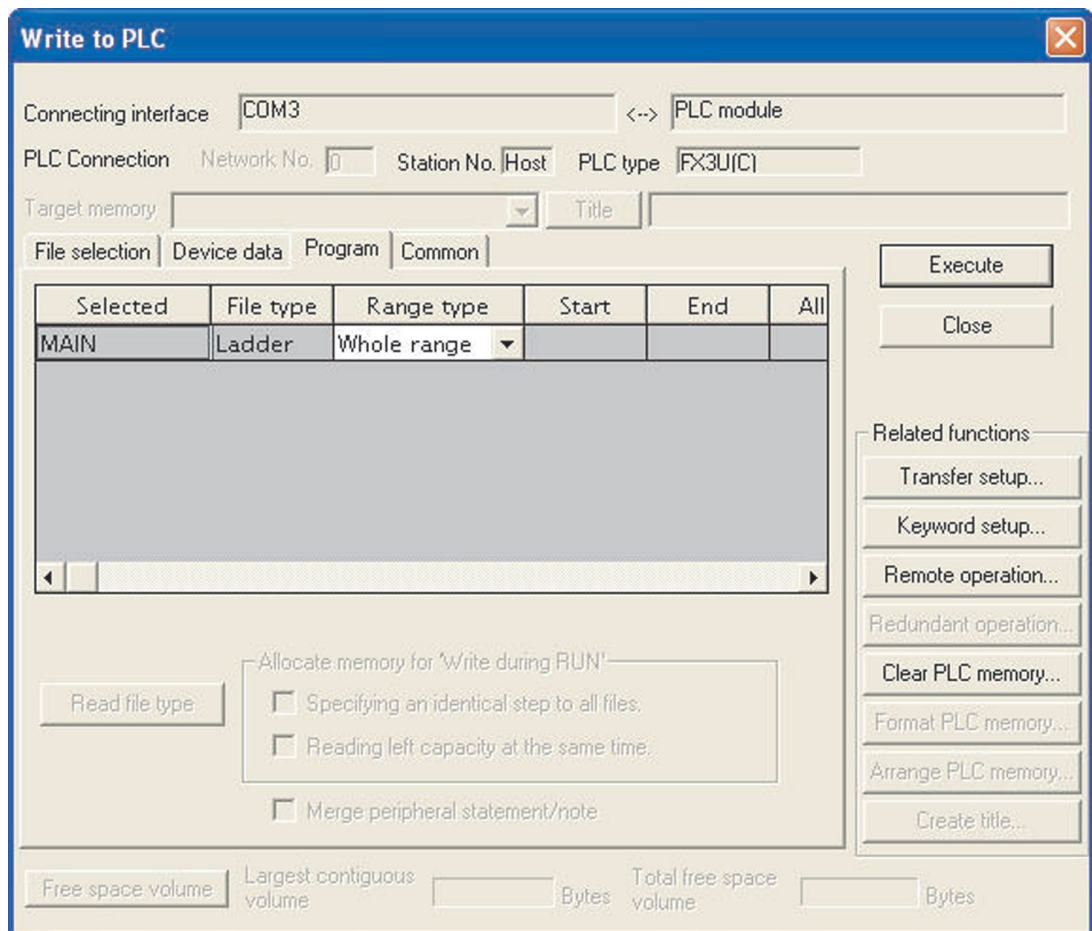
## 12.4 Reducing the Number of Steps Transferred to the PLC

When the project COMPACT\_PROG4 was downloaded, the default size of the program was actually 8000 steps. However, as COMPACT\_PROG4 has only 15 steps, this means that the remaining 7985 steps will all contain NOP (No Operation) instructions. This is used to clear (Null) the contents of the unused memory area. GX Developer from version 8.0 upward will automatically download only the used program steps up to the END statement. However previous versions of the software treat this as an option.

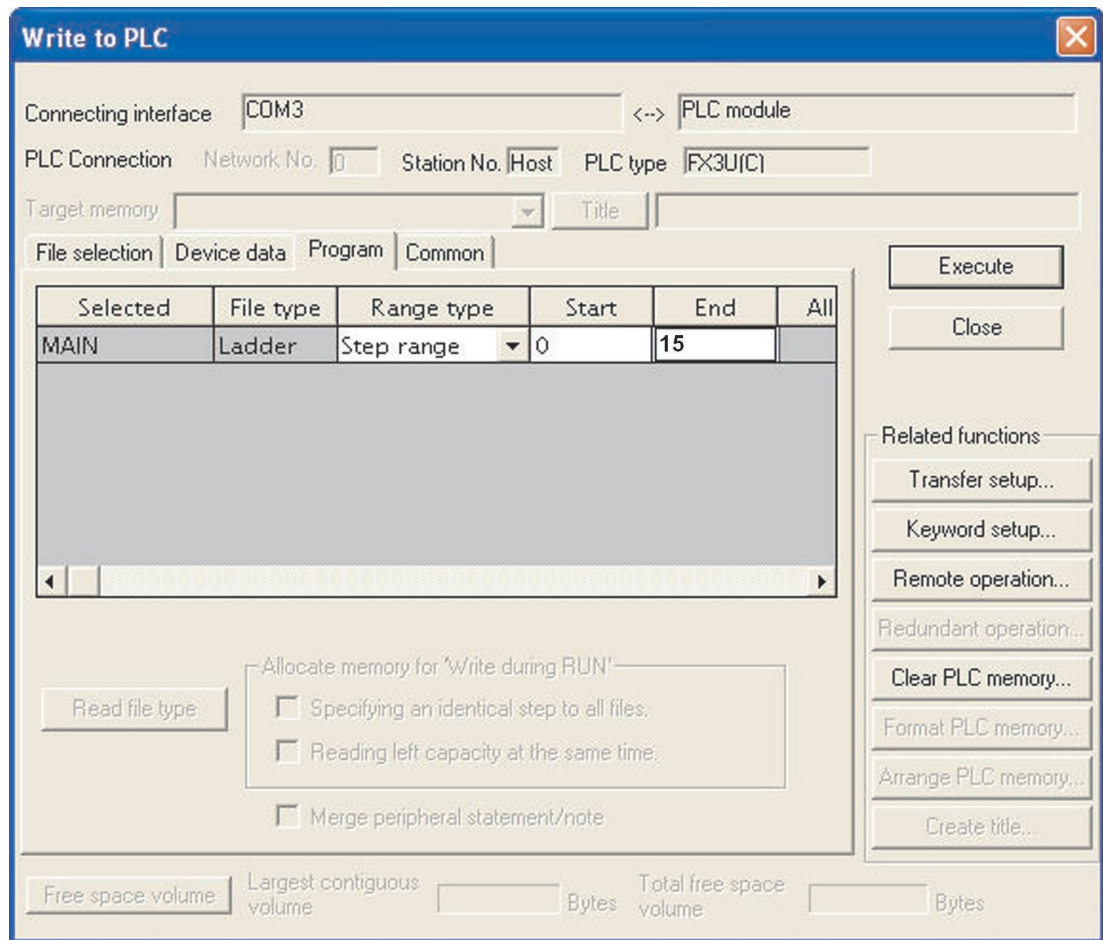
With GX Developer versions less than V8.0, the time taken to write a program to the PLC or when using slower communication speeds on serial ports can be dramatically reduced by using the following procedure:

- ① Check that the main unit is switched to "Stop".
- ② Select **Write to PLC**.
- ③ Select the **Param+Prog** button and select the **Program** tab.

The display now becomes as shown below:



- ④ In column **Range type**, select **Step range** and enter the last step in the program (Step no. of end command). The display should be as follows:



**NOTES**

The Specified **End** step number above must be identical to the last step number of the Ladder Diagram i.e. the step corresponding to the 'END' Command.

Depending on the PLC used and the memory usage, the total number of steps used in the program will differ.

- ⑤ Select **Execute** and answer **Yes** to write the Parameters and only the used steps of COMPACT\_PROG4 to the PLC.





## 13 Executing the Project

To execute the project COMPACT\_PROG4, while referring to the ladder Diagram on GX Developer, carry out the following.

- ① On the FX main unit, switch the mode switch to the RUN position.
- ② Toggle the switch X0 ON then OFF. Y0 will remain ON.
- ③ Repetitively operate switch X2 and observe that after 10 operations, Y1 indicator in the training rig will begin flashing at 1Hz.
- ④ Momentarily operate the X3 Switch and observe Y1 turns OFF. (Please note that a normally closed contact is used for RESET and that for counting X3 has to be switched ON.)
- ⑤ Momentarily operate X1 and note that Y0 turns OFF.

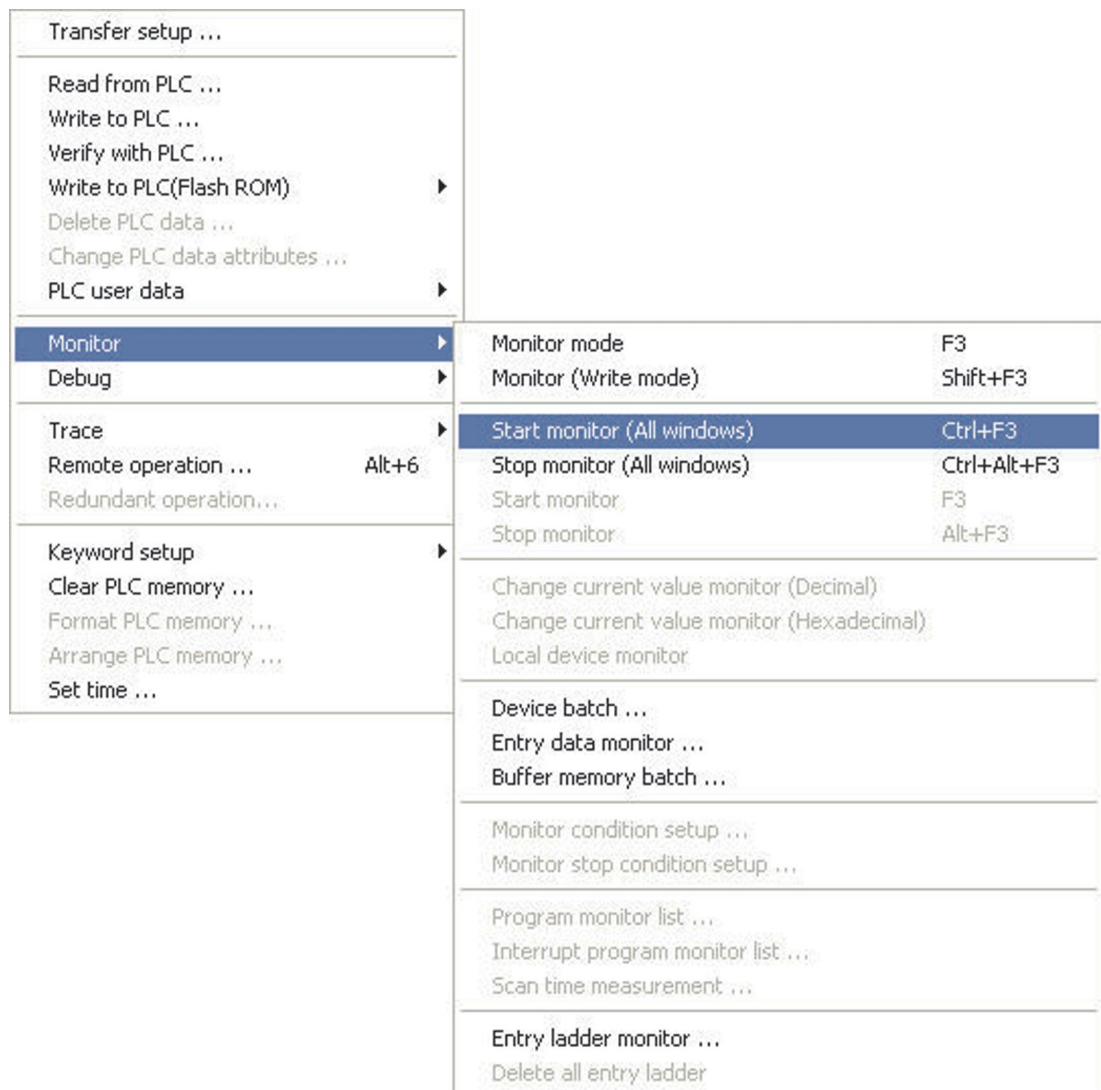


# 14 Monitoring

## 14.1 Monitoring the example program COMPACT\_PROG4

To monitor the Ladder Diagram of COMPACT\_PROG4, carry out the following.

- ① From the Main Menu, select **Online**.
- ② Select **Monitor**



- ③ Select **Start Monitoring (All Windows)**

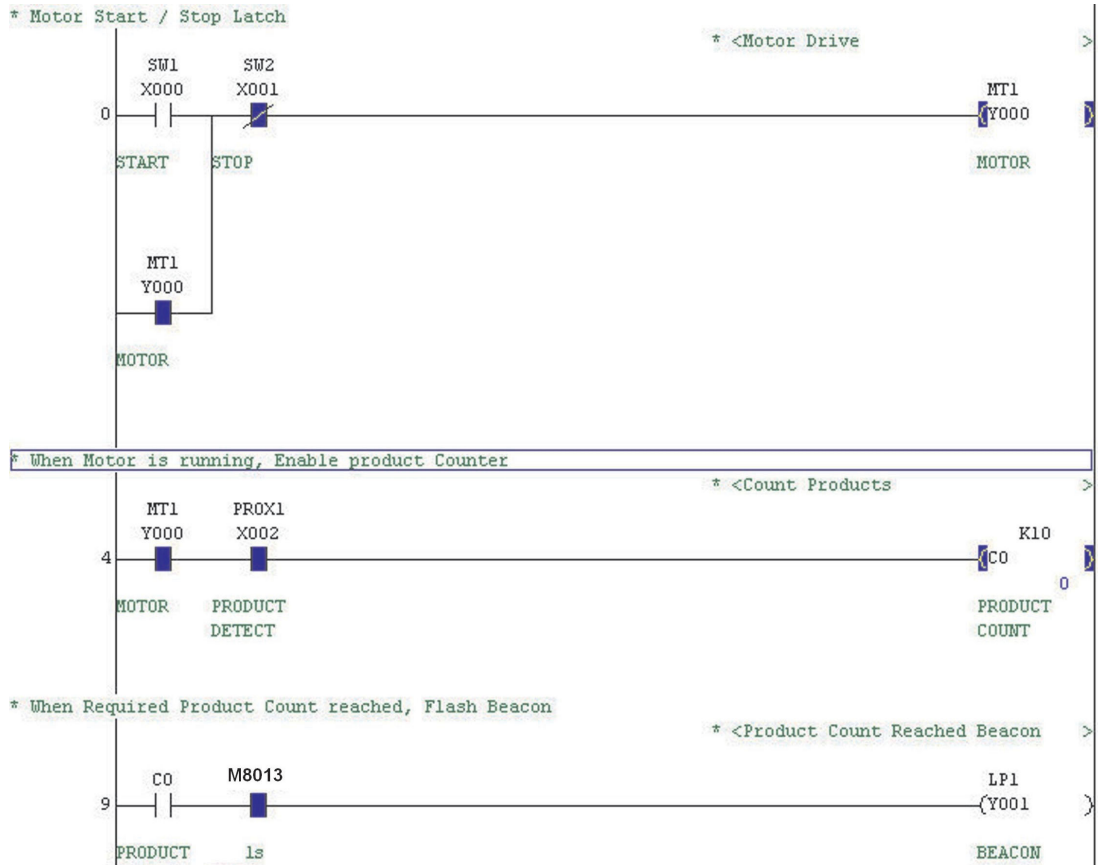
### NOTE

Use of the **Start Monitoring** – F3 shortcut key:  
As can be see from the display, an alternative to the drop down menus to start monitoring is the F3 key.

Alternatively, use the  icon to start monitor mode.

The following display shows the Ladder Diagram COMPACT\_PROG4, whilst in Monitor mode.

Repeat the operations described in the previous chapter. Current count values can be seen beneath the counter references. All contacts and coils in the true (ON) condition can be seen in blue:

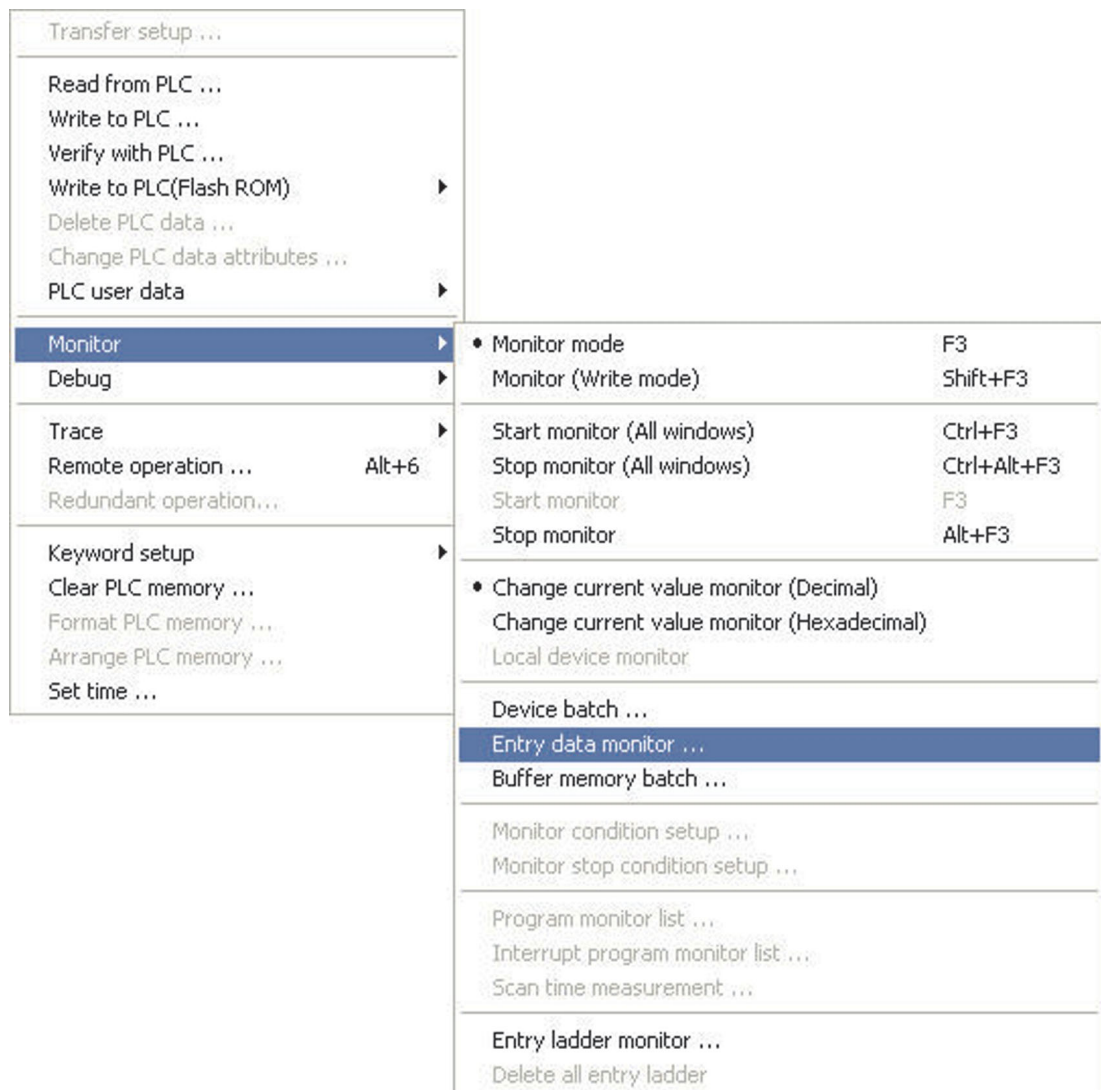


## 14.2 Entry Data Monitoring

Entry Data Monitoring is an alternative method for monitoring the conditions of the Ladder Diagram elements. It enables the condition of many more devices to be displayed in addition to those on the active ladder monitor window.

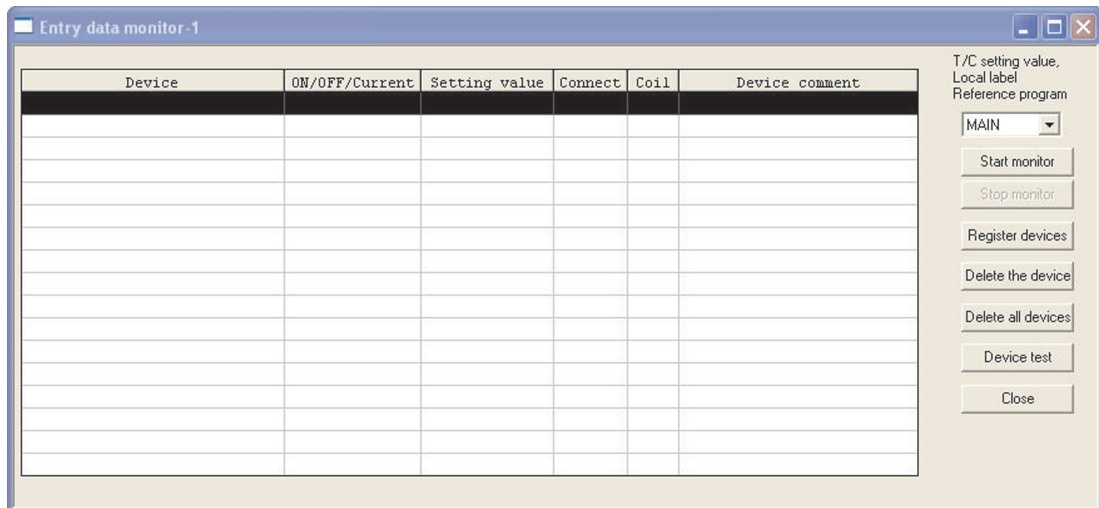
To monitor using Entry Data Monitoring, carry out the following:

- ① From the main menu select **Online**.
- ② Select **Monitor**.
- ③ Select **Entry Data Monitor** thus:



Alternatively, press the  button from the toolbar.

The following window will be presented.

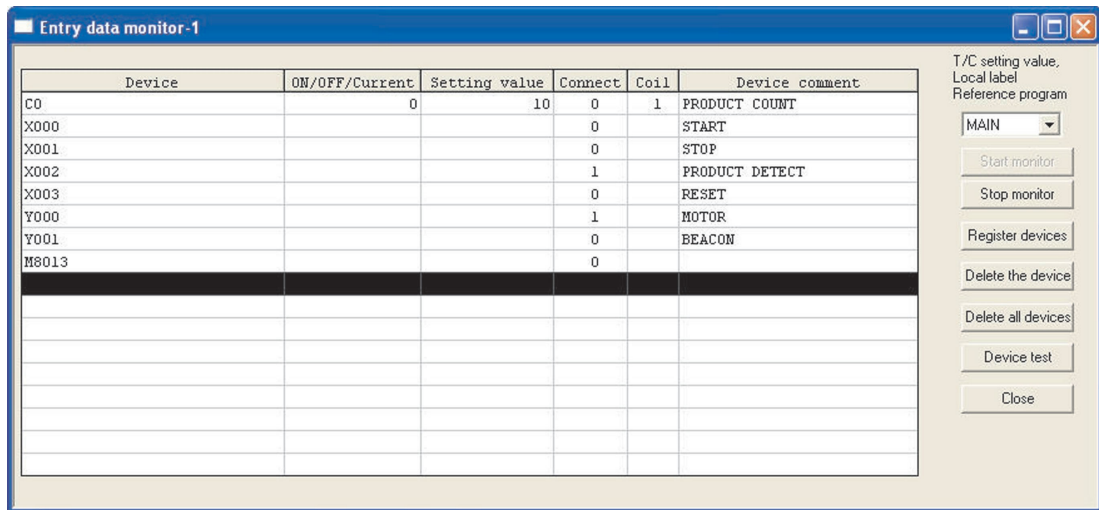


- ④ Select **Register devices** to obtain the **Register device** window.



- ⑤ Enter the following device names into the window using the register button, press cancel when complete:
- C0
  - X0
  - X1
  - X2
  - X3
  - Y0
  - Y1
  - M8013
- ⑥ To close the window after entering the last device, click on **Cancel**.

- ⑦ Click the **Start Monitor** button and the following window provides a live monitor of the values in the listed items:



The above display shows all attributes of the displayed devices.

#### Column Descriptions:

- **Device**  
The MELSEC device name being monitored.
- **ON/OFF/Current**  
Accumulator value of device (Running Value)
- **Setting Value**  
Constant / Preset Setting Value (Where relevant)
- **Connect**  
The Digital Contact condition.
- **Coil**  
The digital coil status (Where relevant)
- **Device Comment**  
The comment for the specific device (Where used).

#### NOTES

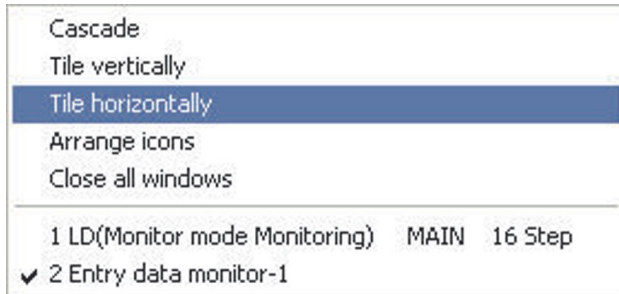
To delete devices in the monitor window, use the 'Up Arrow' and 'Down Arrow' keys on the keyboard to highlight the appropriate device and press the **Delete the device** button.

To clear all devices registered in the monitor window, select the **Delete all devices** button.

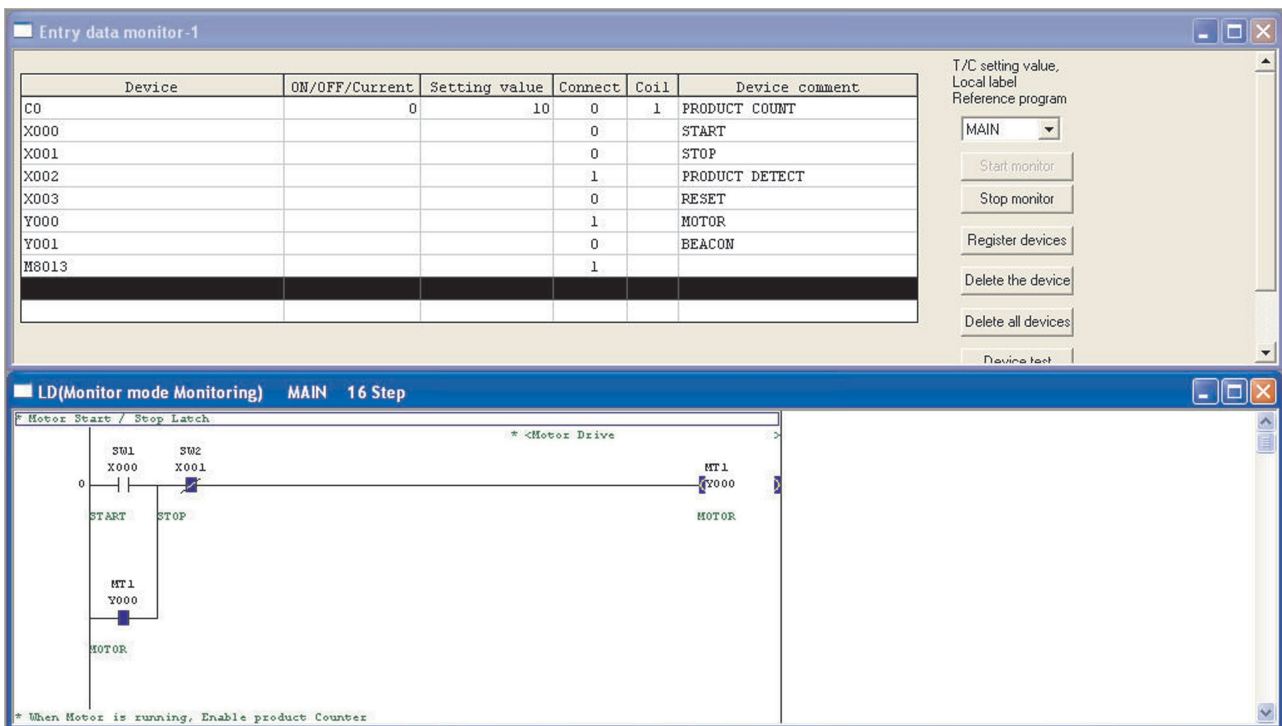
### 14.3 Combined Ladder and Entry Data Monitoring

Using Windows, it is possible to monitor both the Ladder Diagram and the Entry Data.

- ① From the Main Menu, select **Window**.
- ② Select **Tile horizontally**:



- ③ The Ladder diagram window will now be displayed together with the Entry Data monitor window:





## 14.4 Device Test Function

You can also use the programming device (in this case the PC) change the values of devices directly from while you are testing your program. For example, if you need the input signal from a certain switch to initiate a process you can activate the switch from the PC and continue to watch the resulting program execution.



**CAUTION:**

*Be extremely careful when using this function! Changing the states of devices independently of the program can cause potentially dangerous situations for both personnel and equipment!*

Execution by the program has higher priority for the control of devices used by output instructions, such as outputs. The Device Test function only changes these devices very briefly for the test, after which they are returned to the status assigned to them by the program.

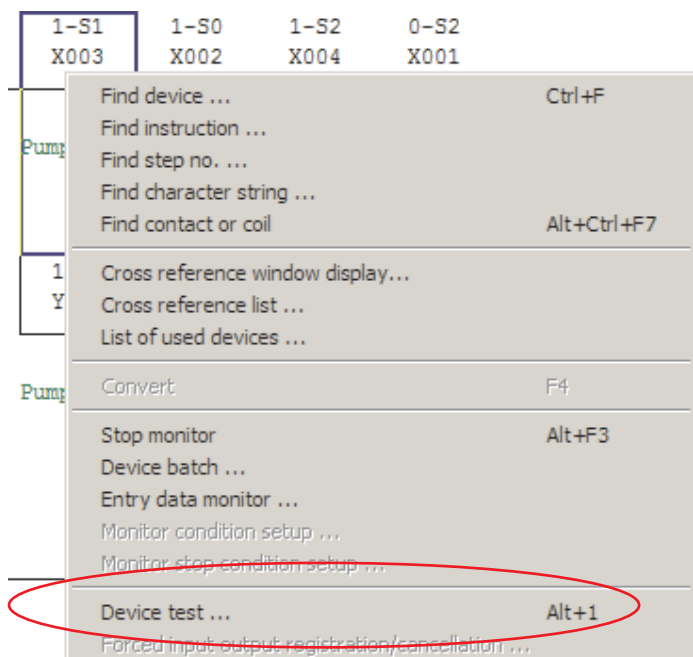
There are several ways to start the Device Test function.

- In the **Online** menu, click on **Debug** and then on **Device test**.



- In **Monitor** mode you can activate the Device Test function by selecting a device in the editor and then right-clicking on it.

With some other test functions the Device Test can be selected in the dialog box of the function.



- Or click on the  tool in the toolbar.

All actions described above open the Device Test dialog:

Bit devices (inputs, outputs, relays etc):  
Set (**Force ON**), reset (**Force OFF**)  
or change state (**Toggle Force**).

Select word devices: On some PLCs you can  
alter the contents of the buffer memory in  
special function modules. To do this you must  
specify the both the start I/O address of the  
module and the memory buffer address.

Value to be written to the word device.  
Click on **Set** to write the value.

History of device tests performed. You can  
search for selected devices with the buttons on  
the right. **Clear** removes all the current entries  
from the history list.

**Device test** ✕

Bit device

Device:  Close

Hide history

---

Word device/buffer memory

Device

Buffer memory Module start I/O

Address:

---

Setting value

---

Program


Label reference program:

---

Execution history

Device	Setting condition	
		Find
		Find next
		Re-setting
		Clear

14 - 8



MITSUBISHI ELECTRIC

# 15 Program Verification

Situations may arise, when due to extensive modifications to a PLC project, the program in the PLC may be different to that stored on disk.

However, it is possible to verify whether or not that the programs stored in the PLC and on disk are identical and if not identical, what those differences are.

Also, when a program is to be monitored, then it is very useful if the documented ladder diagram can be displayed, whilst it is being monitored.

The difficulty though, due to the relatively large amounts of memory required to store this data, it is not always practical to store the documentation i.e. comments, statements and notes in the PLC itself.

However, by monitoring the program using the program stored on a disk, which also contains the documentation, the project can be more effectively monitored.

Hence it is essential before the project is monitored, that it can be verified that the project stored on the disk is identical to that stored in the PLC.

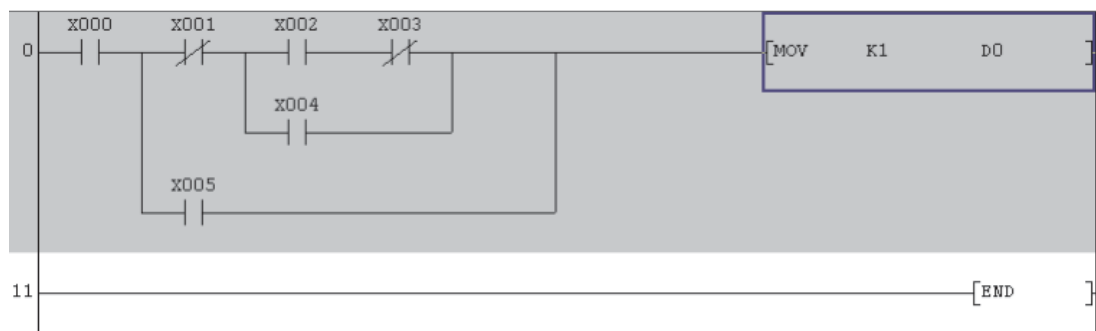
## 15.1 Verification of Example Programs

To demonstrate the Verify facility, a new short program will be used.

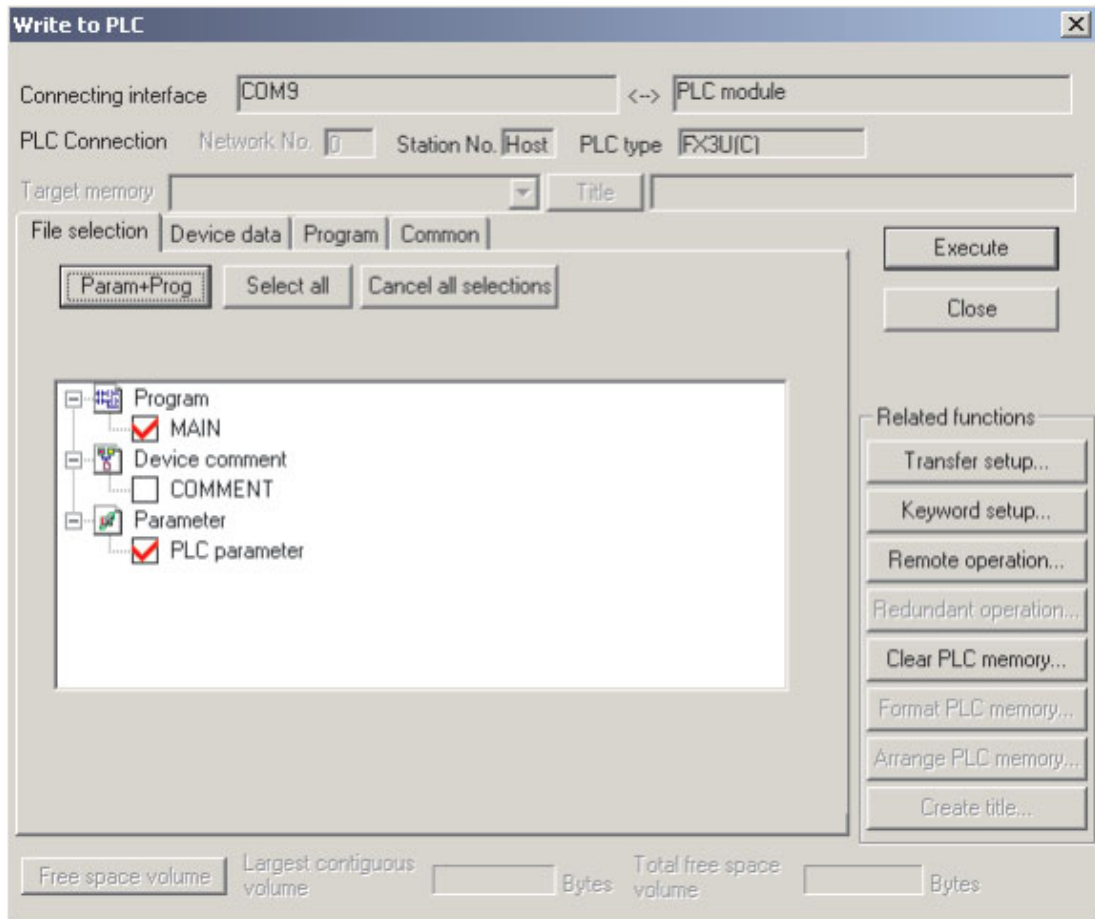
- ① From the **Project** menu, select **New Project**.



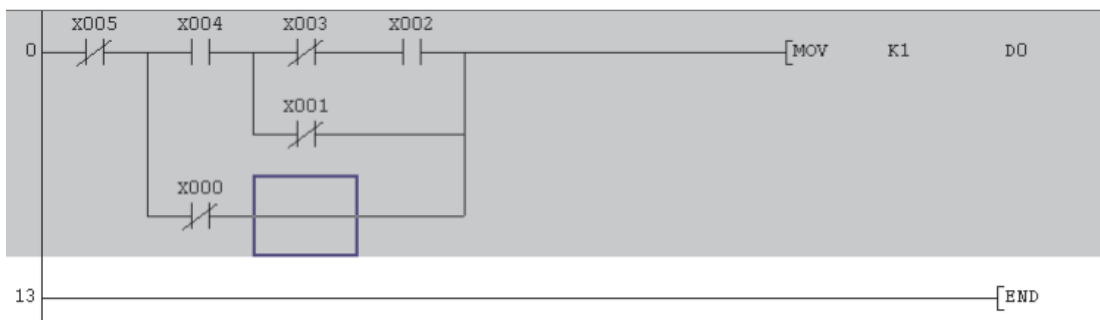
- ② Enter the following program.



- ③ Convert and transfer the program to the PLC as shown on the next page.



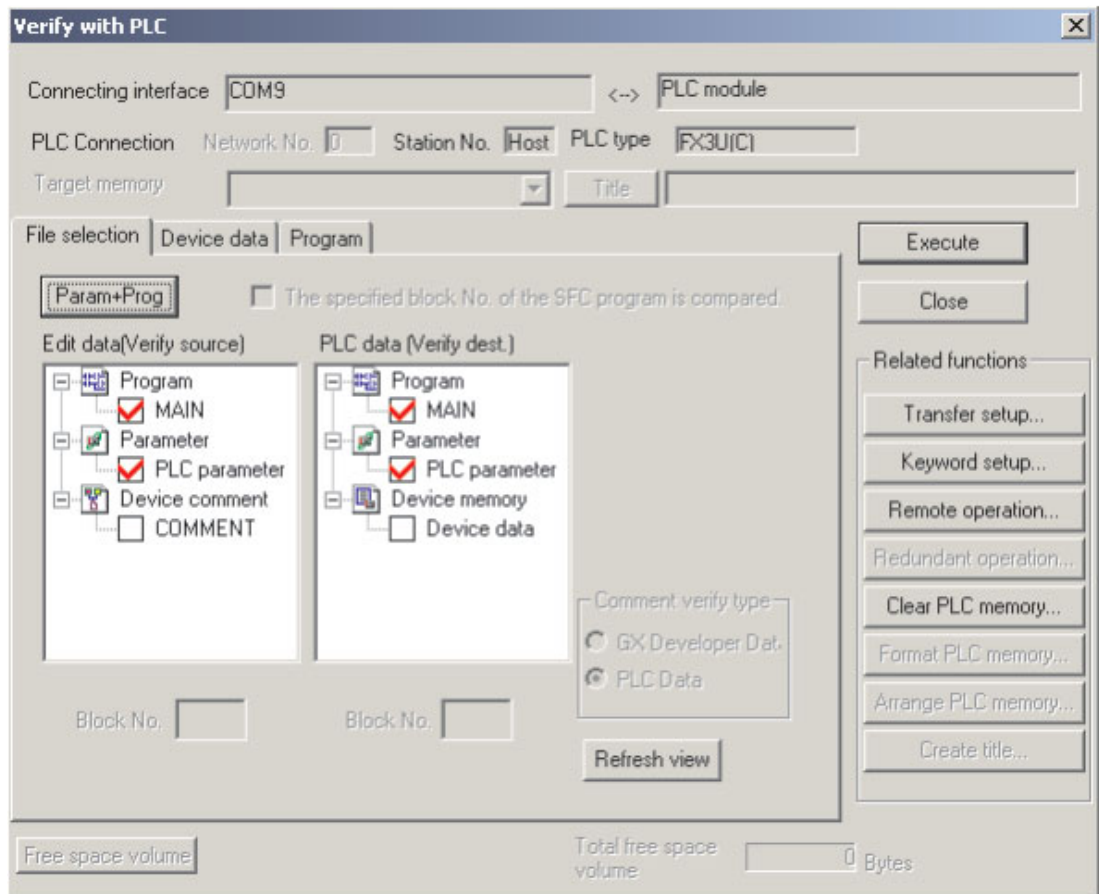
- ④ Modify the program on the hard disk as shown below. On a quick glance the two programs look the same, but nearly all instructions and devices have been changed. Therefore the function is completely different to the program stored in the PLC.



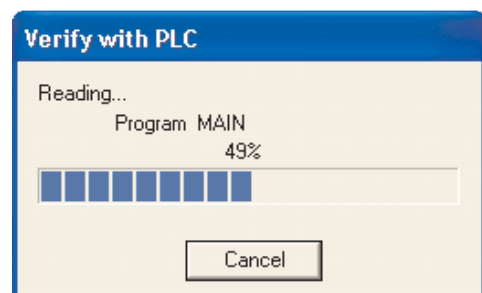
- ⑤ In the **Online** menu, select **Verify with PLC**.



The following window will be displayed:



- ⑥ Select **Param+Prog** and click **Execute** to start the verification.



- ⑦ When the verification procedure has completed, the following display will be shown illustrating the differences found between the two programs:

```
[PLC verify: Program]
Verify source
  Project name -
  Data name -MAIN
Verify destination
  Project name - none
  Data name -MAIN
```

<Memory>		<PLC>	
Step	Instruction	Step	Instruction
0	LDI X005	0	LD X000
1	LD X004	1	LDI X001
2	LDI X003	2	LD X002
3	AND X002	3	ANI X003
4	ORI X001	4	OR X004
6	ORI X000	6	OR X005

6 items unmatched.

# 16 Serial Transfer – Upload

Here are two possible scenarios where it is necessary to transfer the program from the PLC into GX Developer:

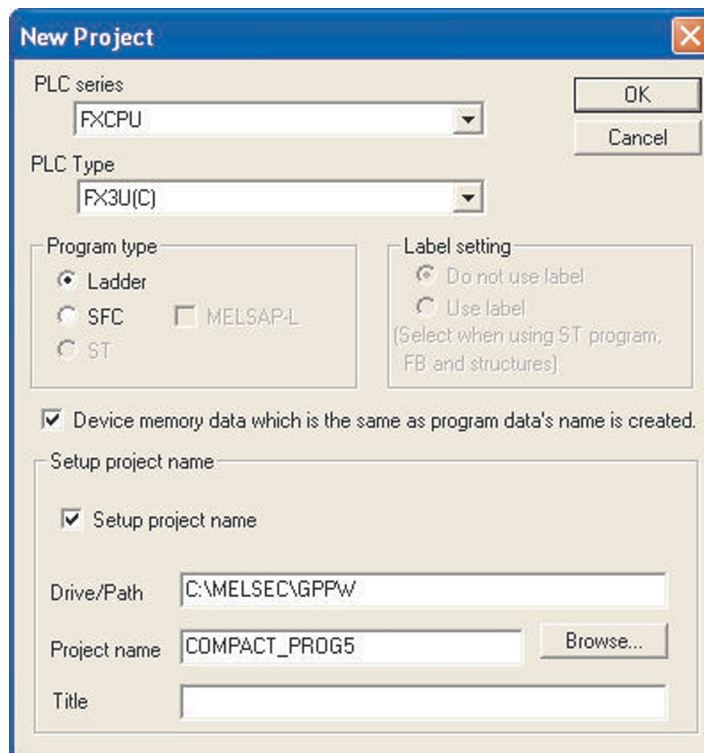
- Where the GX Developer source files are not available, it will be required that the program in the PLC be uploaded and subsequently saved into GX Developer in order that a backup of the raw PLC code is created. The program may then be documented using information from circuit diagrams and reverse engineering techniques.
- Circumstances can arise, when it is necessary to know what program is stored in the PLC. This may be due to a number of modifications being made to the original program and those changes have not been fully documented and saved on the Master Disks.

Hence after verifying that the program in the PLC is different to that stored on disk, then the working program within the PLC must be uploaded into GX Developer and stored on the Master Disks.

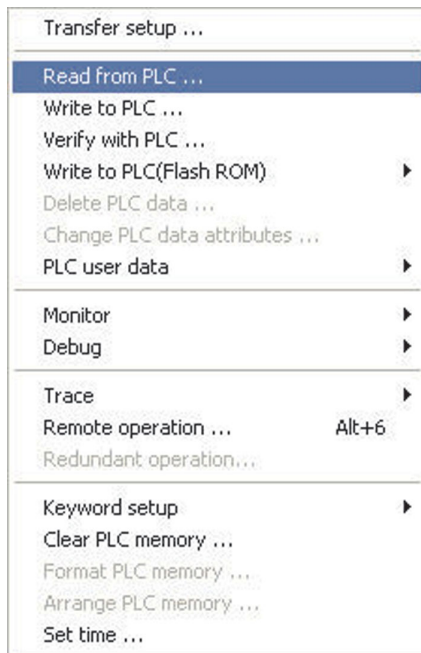
## 16.1 Example Program Upload

The following describes how the project COMPACT\_PROG4 is uploaded from the FX base unit and saved as COMPACT\_PROG5. It is assumed that the program COMPACT\_PROG4 is still resident in the PLC:

- ① Close the currently loaded project by selecting **Close project** from the **Project** Menu. (This is optional, as GX developer will prompt to close an already open project when a new one is created)
- ② Select **Project** and open a new project with the name COMPACT\_PROG5:

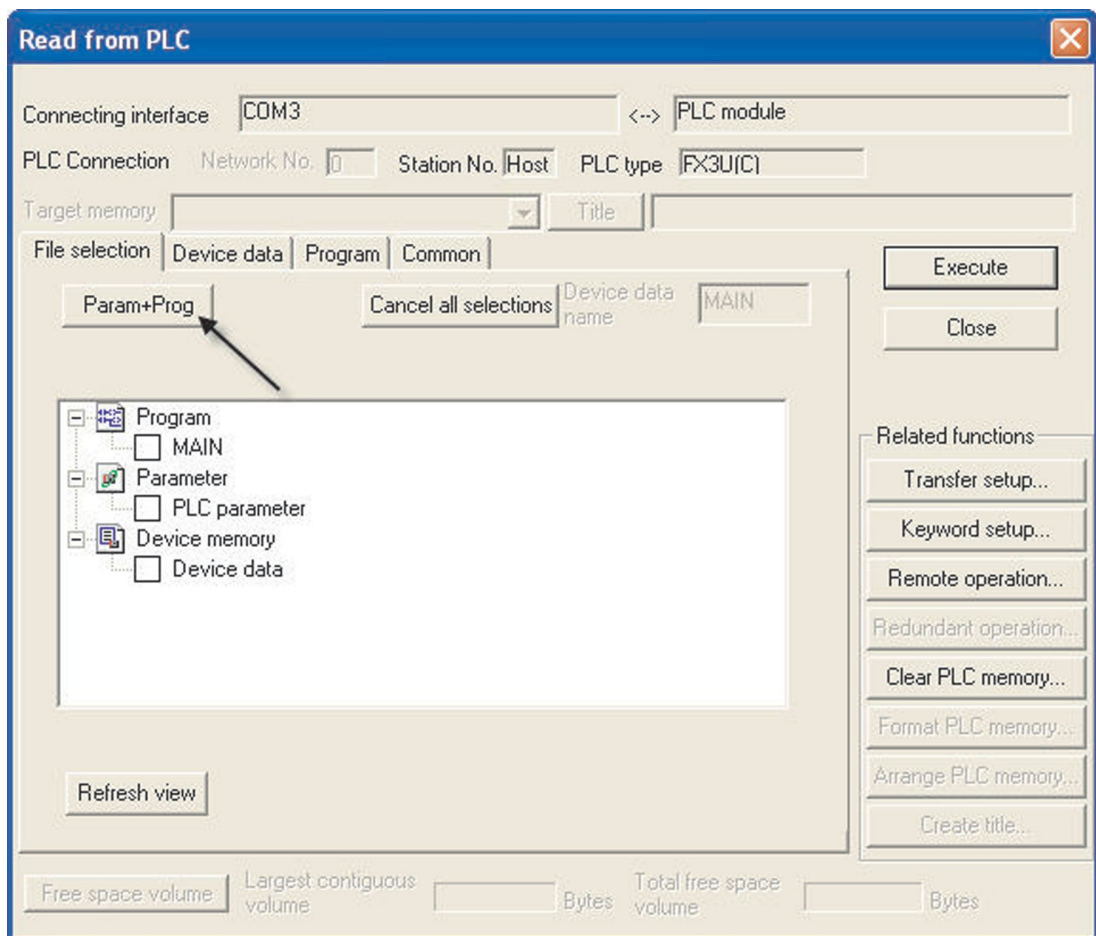


③ Select **Online** and **Read from PLC**.



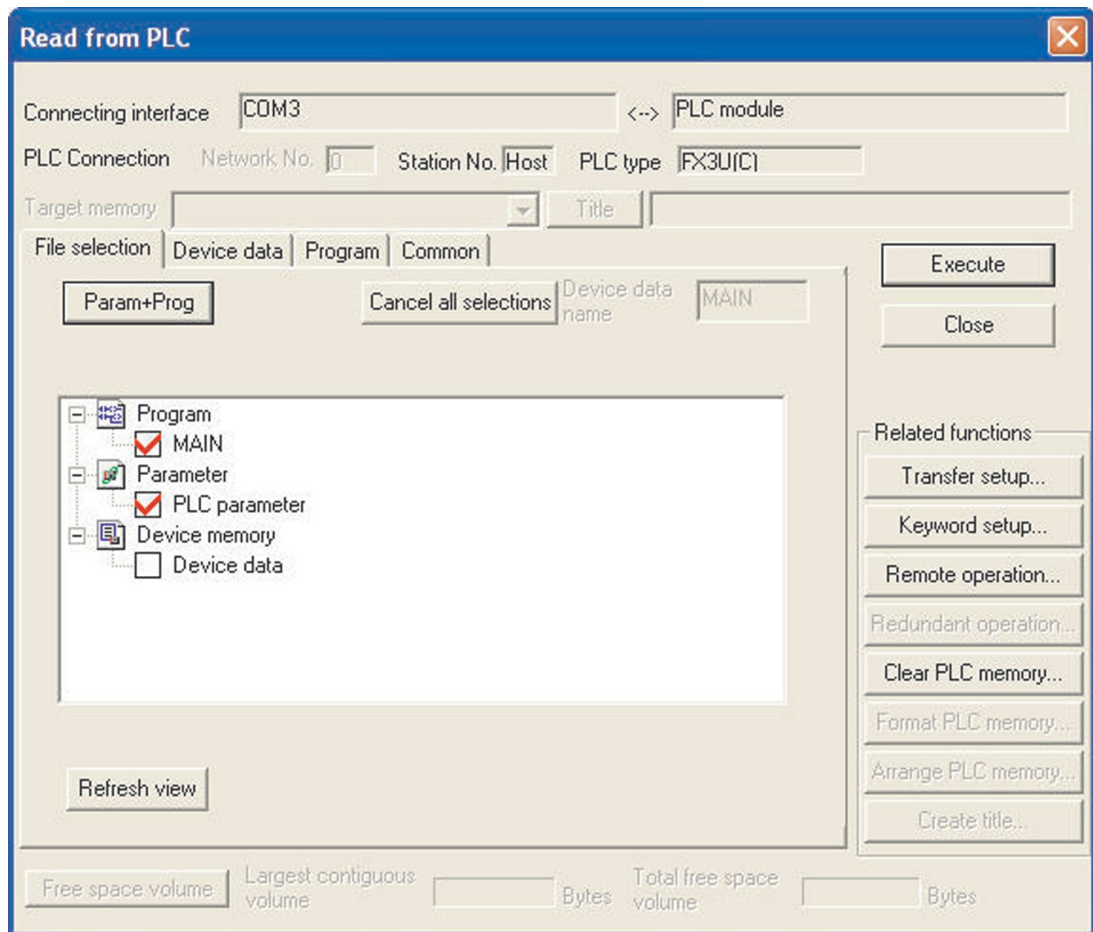
Alternatively click on the **Read from PLC** icon: 

The display will be as shown below.





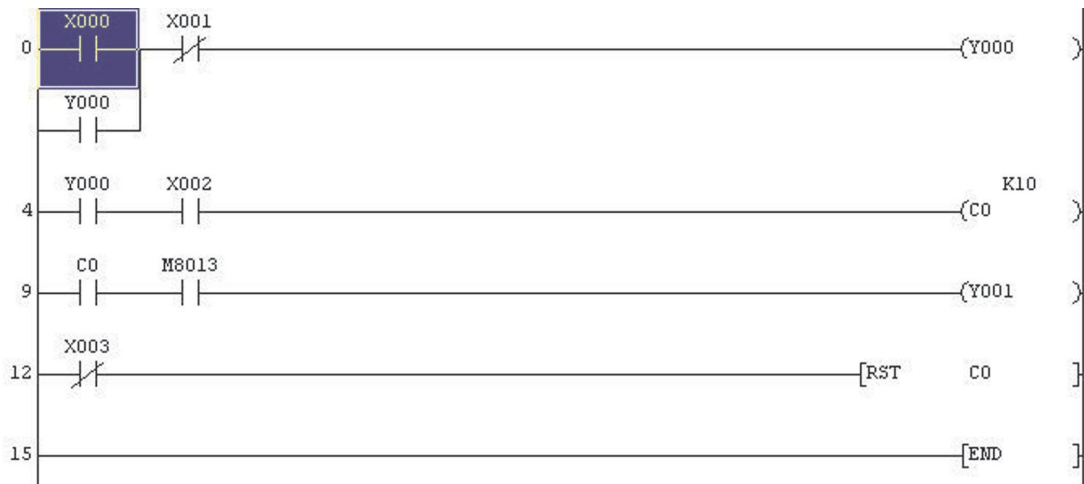
- ④ Select the **Param+Prog** button as shown above:



- ⑤ Select **Execute** and the following prompt dialogue will be displayed:



- ⑥ Click **Yes**.
- ⑦ After completion of the transfer click the **Close** button on the dialogue window to close the transfer display. The uploaded program COMPACT\_PROG5 from the PLC will now be displayed. (This was COMPACT\_PROG4 stored in the PLC.)



⑧ Save COMPACT\_PROG5.

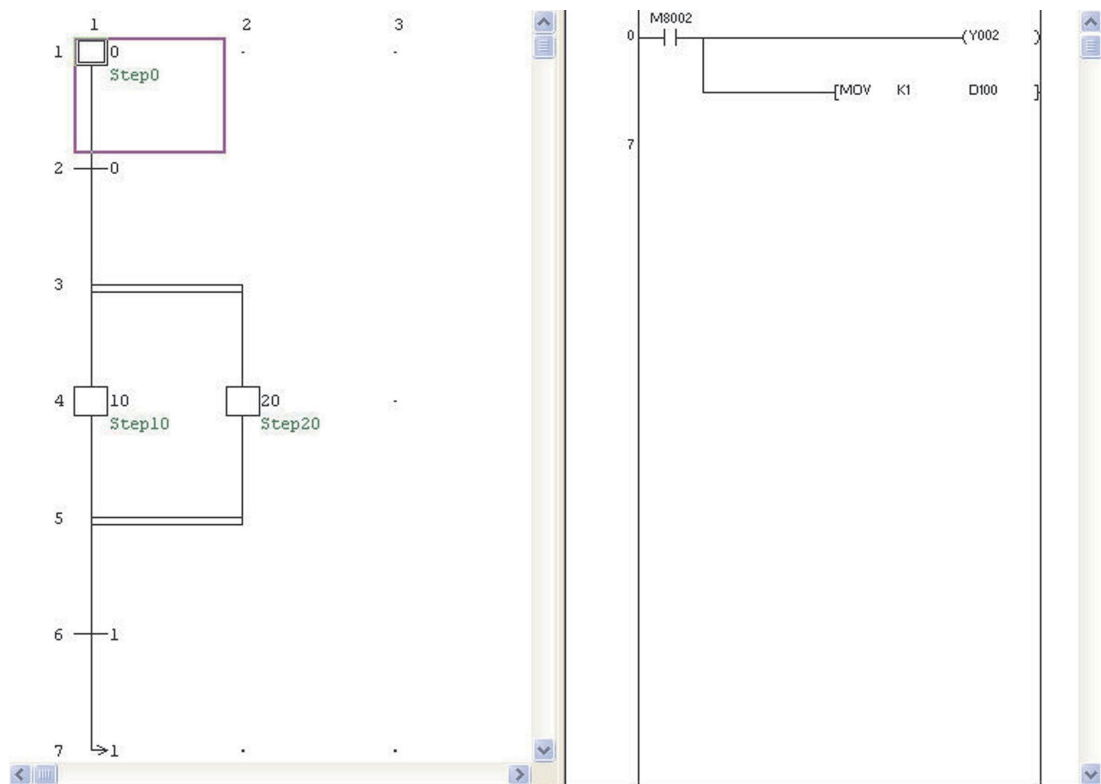
**NOTE** | Remember, use the toolbar button, it is quicker.

# 17 Sequential Function Chart (SFC)

Sequential Function Chart Language (SFC) is one of the graphical methods that can be used for programming the MELSEC range of PLCs. By clearly representing the operating sequence of machinery/equipment controlled by the CPU, this language makes it easy to grasp the system as a whole and makes programming easier.

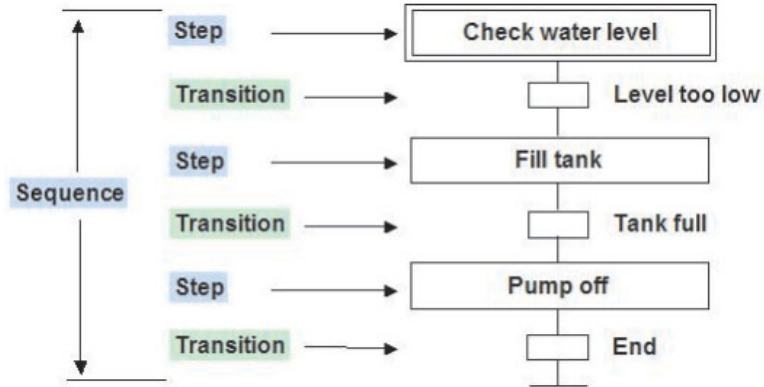
In contrast to the case where a program represented by ladders is entirely executed every scan, only the minimum required part of a program may be run if it is written in SFC format.

- SFC is a graphical language which provides a diagrammatic representation of program sequences → flowchart
- GX Developer Sequential Function Chart is IEC 1131.3 compatible, based on the French Grafset (IEC 848)
- Main structure and suitable for rapid diagnostics
- The basic elements are steps with action blocks and transitions
- Steps consist of a piece of program that is carried out until a condition specified in the transition is met.
- Easy programming of complex tasks by dividing in smaller parts
- Each element can be programmed in ladder or instruction list
- GX Developer-SFC steps can be switched between IL and Ladder



## 17.1 SFC Elements

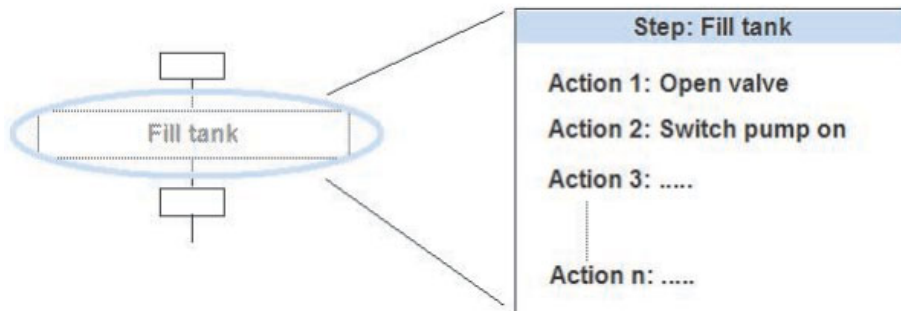
SFC is a structured language which allows clear representation of complex processes. A process is divided into steps and transitions.



### 17.1.1 Steps

A step represents either no, one, or several actions. An action for instance can be the setting of a boolean device or the start of a PLC program.

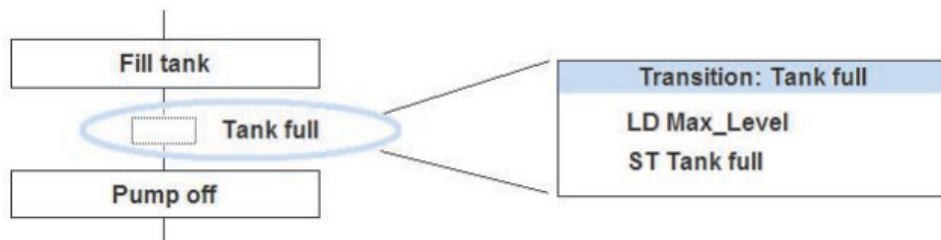
The called program can be edited in every language, even in SFC.



### 17.1.2 Transitions

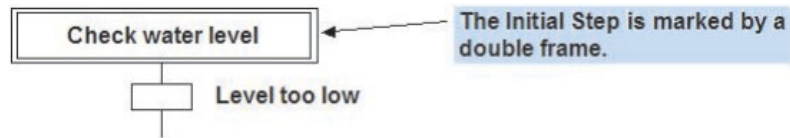
To each transition a transition *condition* is assigned. When the transition condition is matched the next step will be activated.

Transition conditions can be written using any of the editors - except Sequential Function Chart language itself.



### 17.1.3 Initialization Step

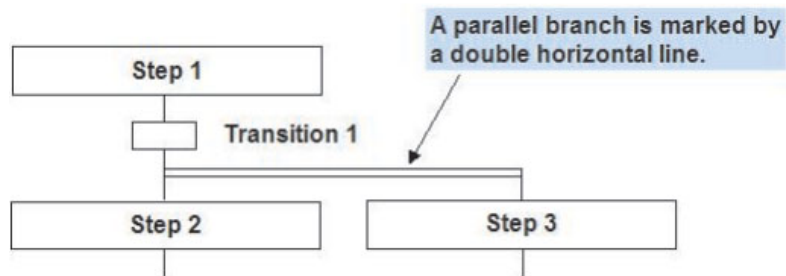
Every SFC program starts with the Initial Step.



The Initial Step is treated like a regular step. Actions can be added to the Initial Step like to other steps.

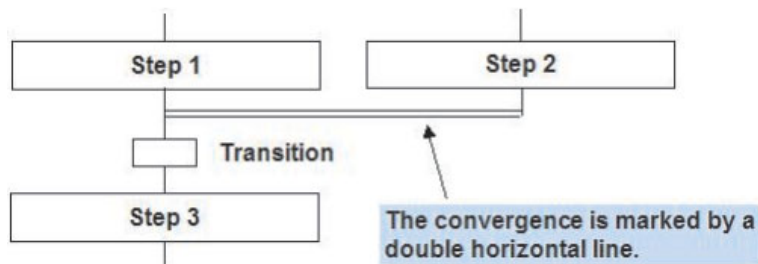
## 17.2 Rules for Sequences

### 17.2.1 Divergence into Parallel Sequences



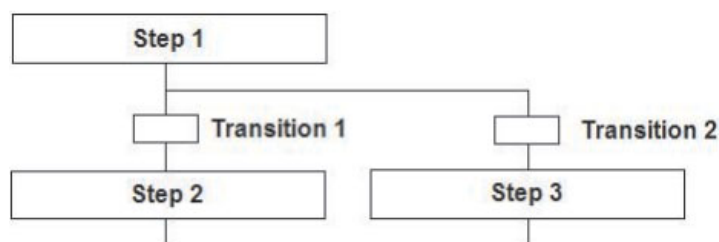
- A parallel branch is allowed after a transition only.
- Only one transition can exist before a parallel branch.
- When the transition condition before the parallel branch is matched, both following steps are executed independently of each another.

### 17.2.2 Convergence of Parallel Sequences



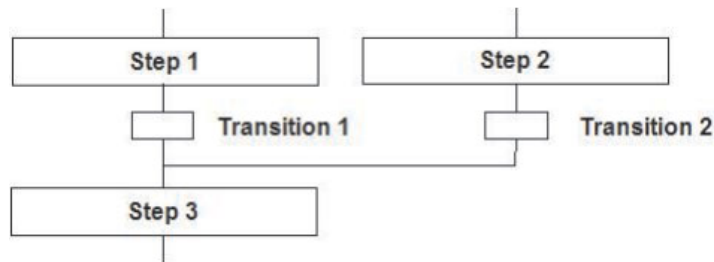
- Only when both Step 1 and Step 2 have been executed and the transition condition is evaluated TRUE the next step will be executed.
- The convergence must be followed by a transition.

### 17.2.3 Divergence into Selective Sequences



- Selective branching is possible before transitions only. It depends on these transitions which step is executed.
- If several transition conditions are evaluated TRUE at the same time, the execution priority is given from the arrangement of the sequences. The leftmost sequence will be executed first.

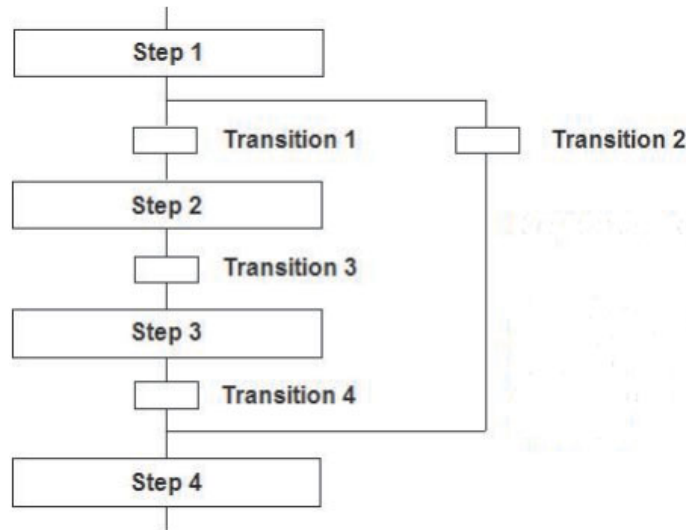
### 17.2.4 Convergence of Selective Sequences



When the respective transition condition becomes TRUE the next common step is executed.

### 17.2.5 Jumps

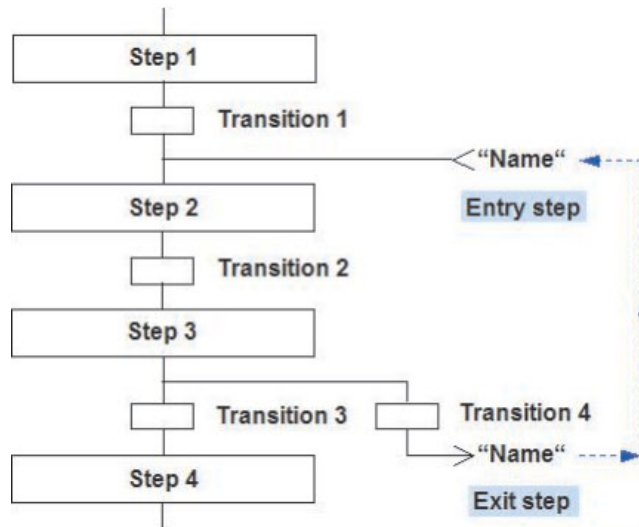
It is possible to jump in a sequence.



When in the example above the transition condition (transition 2) for the jump is evaluated TRUE, the steps 2 and 3 are not executed but the step 4 is activated.

### 17.2.6 Exit and Entry Steps

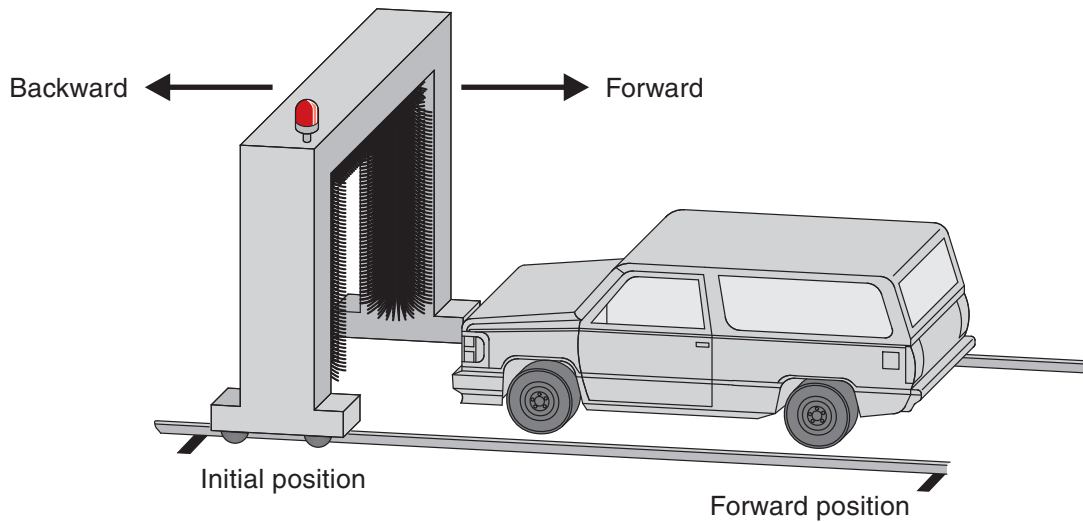
Exit and Entry steps can be used to jump forward and backward. The name of the exit step and the corresponding entry step must be the same.





## 17.3 Example for Programming in SFC

### Carwash



The washing process is divided into steps:

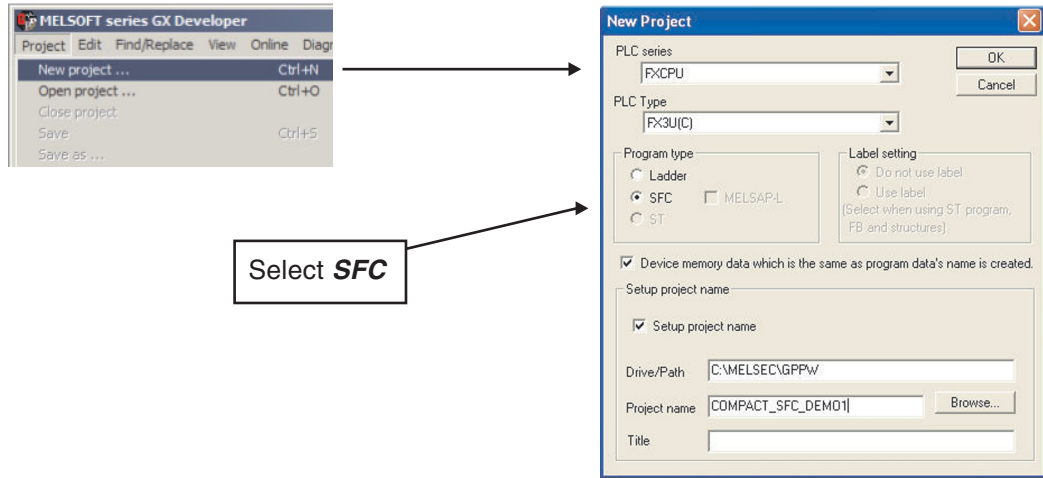
1. A lamp indicates the readiness of the carwash.
2. After pressing the start button the frame with the brushes moves forward and backward. Water is sprayed on the car to soak the dirt. The brushes are switched off.
3. The frame moves forward and backward with rotating brushes.
4. To dry the car the frame moves again forward and backward but this time compressed air is blown onto the car. The brushes are switched off again.
5. The end of the washing process is indicated by a flashing light.

#### List of used devices

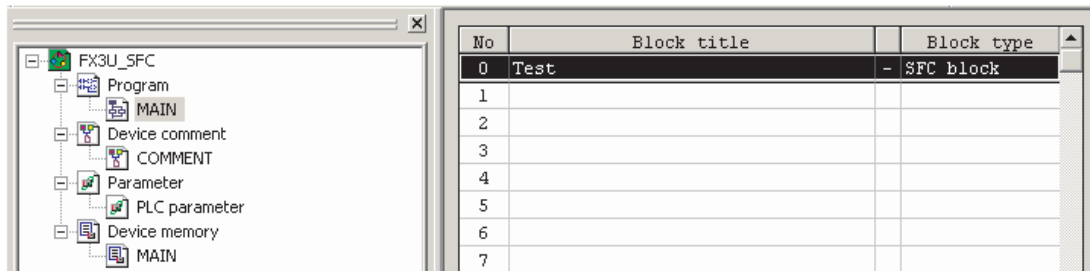
Inputs		Outputs	
X0	Frame in initial position	Y0	Frame backward
X1	Frame in forward position	Y1	Frame forward
X2	—	Y2	Water ON
X3	—	Y3	Brushes ON
X4	—	Y4	Air ON
X5	Start button	Y5	Lamp "Finished/Ready"

## 17.4 Creating a SFC-Block

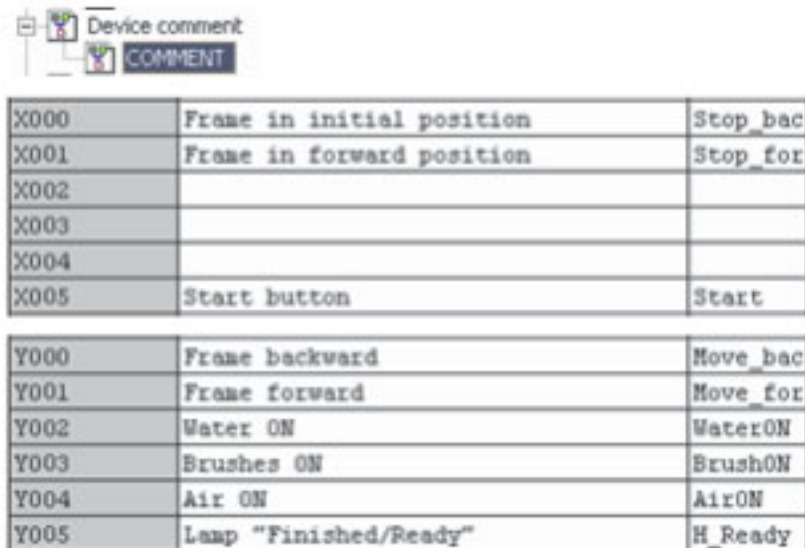
Select **Project -> New project** in the main screen of the GX Developer. Select the connected PLC type, the program type **SFC** and the project name.



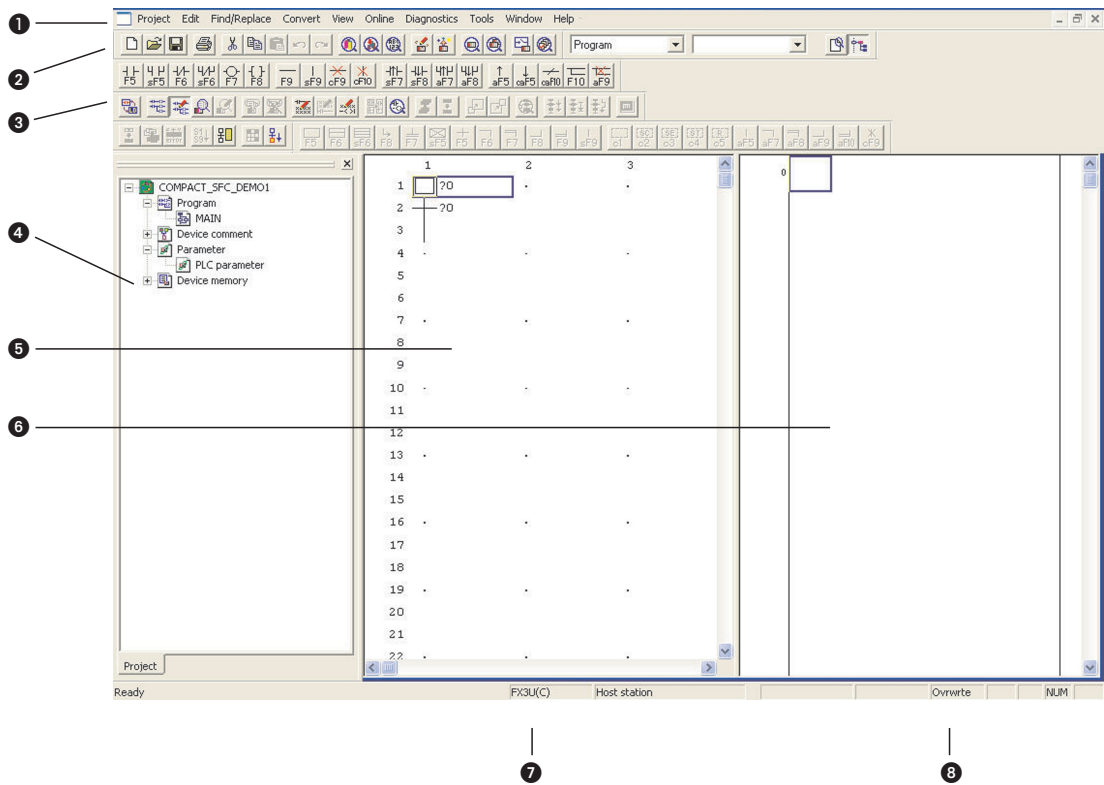
Select **OK** and the following screen will be displayed:



Enter the comments for the devices:




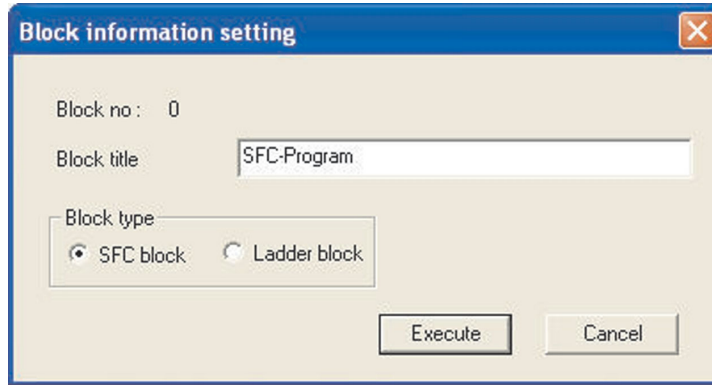
### 17.4.1 SFC Diagram editing screen



- ① Area for displaying the project name being edited, the number of steps used, the block number being displayed and so on
- ② Menu names on the menu bar
- ③ Icons on the toolbar
- ④ Project list display
- ⑤ SFC diagram editing area
- ⑥ Operation output/transition condition program editing area (Zoom side)
- ⑦ Edited CPU type
- ⑧ Edit mode (overwrite/insert)

### 17.4.2 Block information

Select **Block Information** at the menu **Edit** or click on the  icon in the SFC tool bar for setting the block information to the corresponding block.

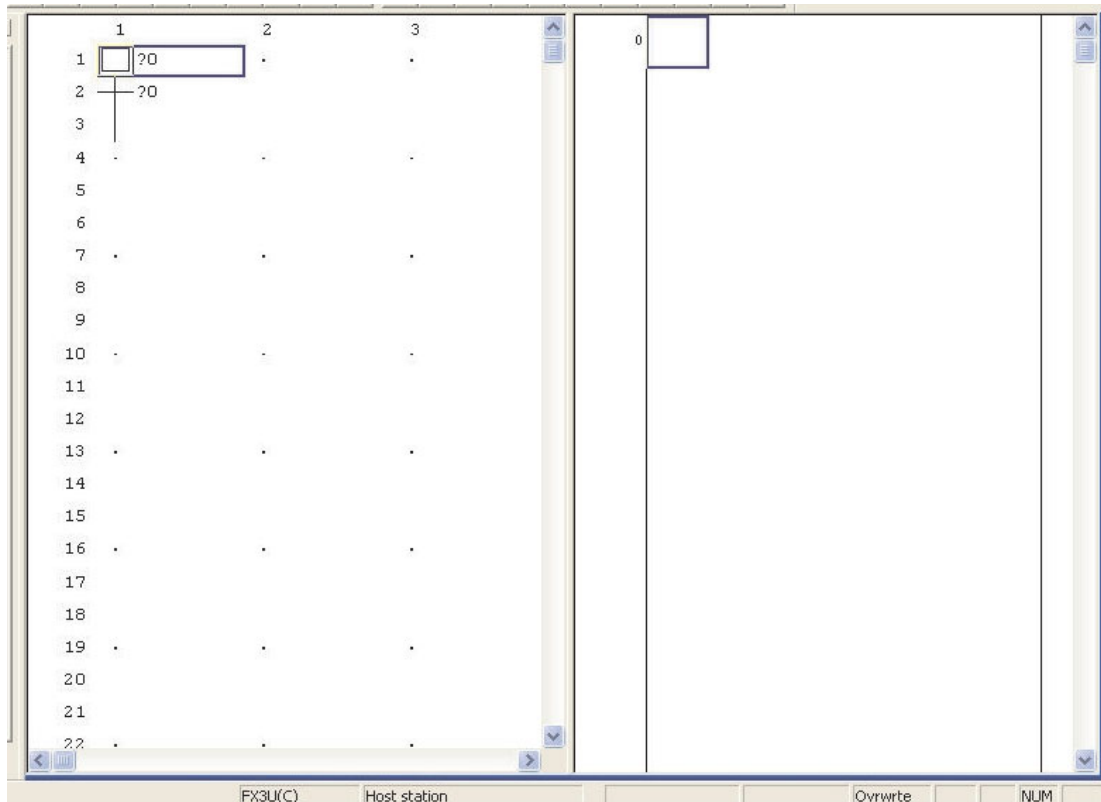


### 17.4.3 Editing the project

No	Block title	Block type
0	Test	- SFC block
1		
2		
3		
4		
5		

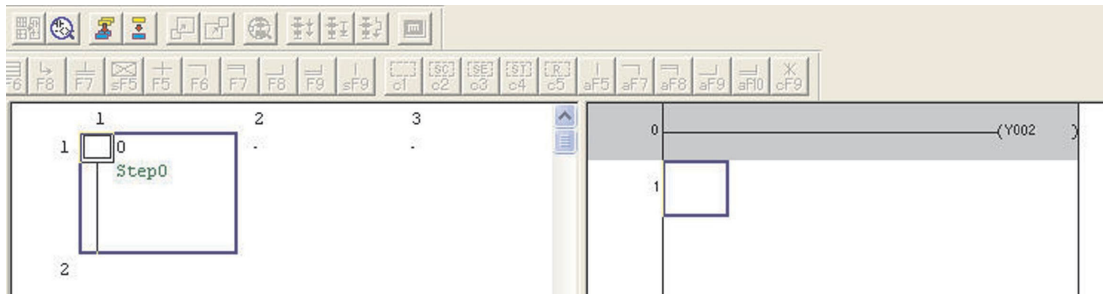
Double click in row number 0.

#### Enter Block

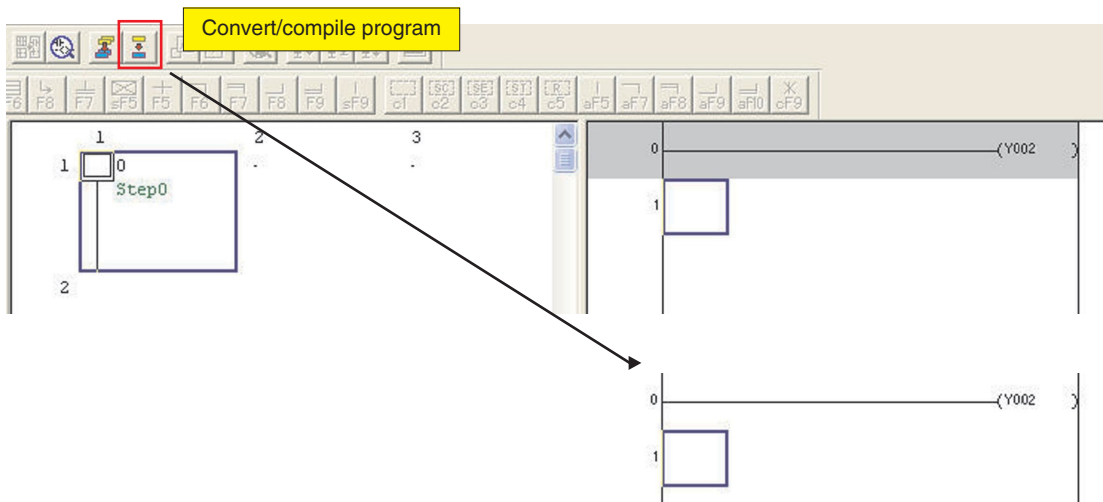


### Enter logic for Step

Click on the step and then in the "operation output program editing area". Enter the required instructions as in any ladder diagram.



After the condition has been entered, the condition is displayed in dark grey and has to be converted.

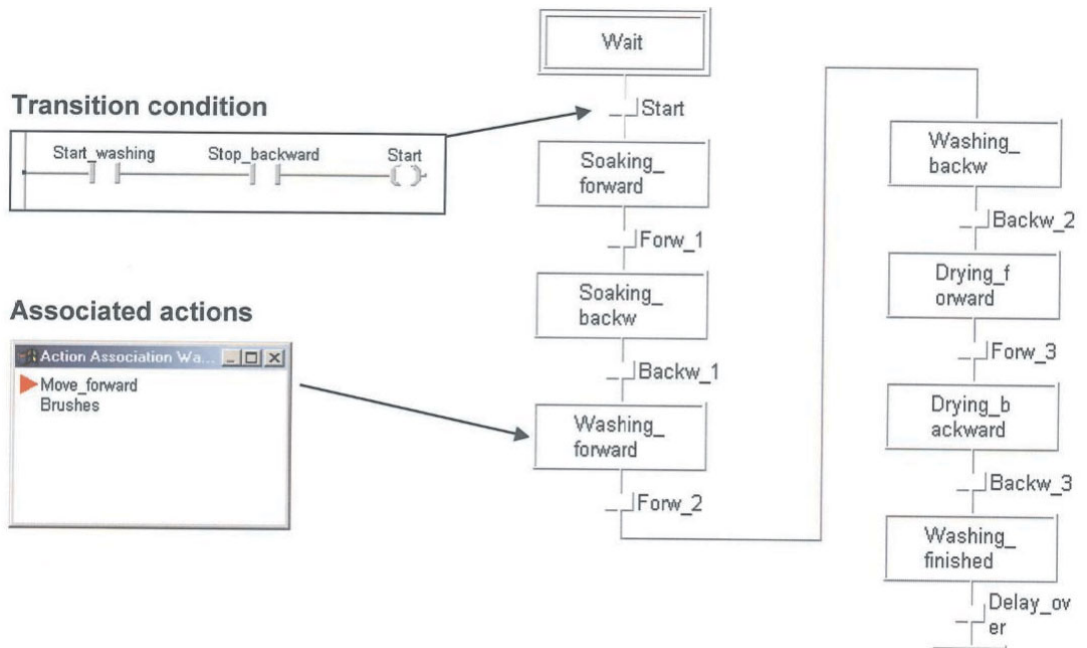


### Condition of transition

Click on the transition and enter the instructions in the "Transition condition program editing area".



After completion of all steps and transitions, the project looks like following.



### 17.4.4 Transfer project

Before the project can be transferred into the PLC, all blocks have to be converted. Use the icon shown on the left. Then click the **Write to PLC** icon to transfer the program to the FX base unit.



### 17.4.5 Monitor project

The status of the block can be observed by switching on the **Monitor mode**. The active step is marked with blue background. The condition of the selected transition is shown in the transition edit area.

# 18 Counters

Counters are a very important part of a sequence control system.

They can be used for example:

- To ensure a particular part of a sequence, is repeated a known number of times.
- To count the number of items, being loaded into a carton.
- To count the number of items passing along a conveyor belt, in a given time.
- To position a component, prior to it being machined.

## Counter Configuration

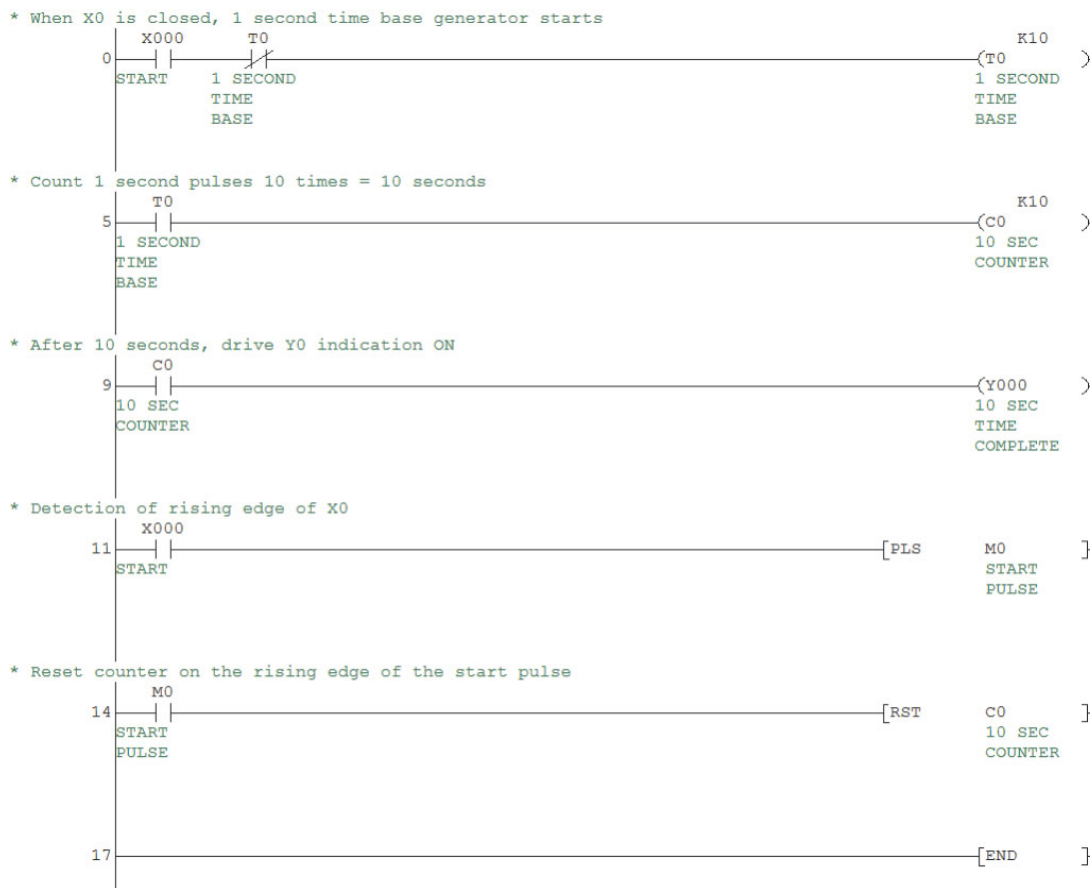
- Counters occupy 3 instruction steps of program memory in the FX range of PLC's.
- Driving the counter coil causes a count to be registered on the rising edge of the driving input.
- When the counter register is equal to the preset, the counter contact closes.
- In order to re-start the counter a separate RESET [RST] instruction is required which zeros the counter register and turns the counter contact OFF.

The following example programs illustrate various counter configurations and applications.

## 18.1 Programm Example – COUNT DELAY


The following example program COUNT DELAY, demonstrates how a counter can be used to produce an extended time delay.

### Ladder Diagram – COUNT DELAY



#### NOTES

- To enter -[ PLS M0 ]- enter the following:
  - pls <space>
  - m0 <enter>.
- Use the same procedure for -[ RST C0 ]- i.e.
  - rst <space>.
  - c0 <enter>.

NB: Pressing the  button to open the square brackets before entering the text for the command is optional.



### Principle of Operation

#### ● Line 0

The closing of input X0 and the normally closed timer contact T0 will provide a path to enable the coil of timer T0 to be energised.

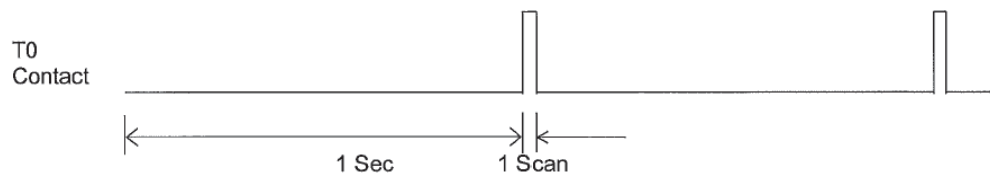
After 1 second, timer T0 times out and its normally closed contact will open, causing the timer to become de-energised for a time equal to one scan period.

With the timer dropping out, its contact re-closes causing the timer to be re-energised once more.

This 'cut-throat' timer circuit is effectively a pulse oscillator, whose contacts momentarily operate every 1 second.

#### ● Line 5

With the momentary closure of the normally open contacts of T0, a count pulse is sent to counter C0 every 1 second.



#### ● Line 9

Counter C0 counts the incoming pulses and when the number of pulses equals the preset K value of 10, all the C0 contacts operate as follows:

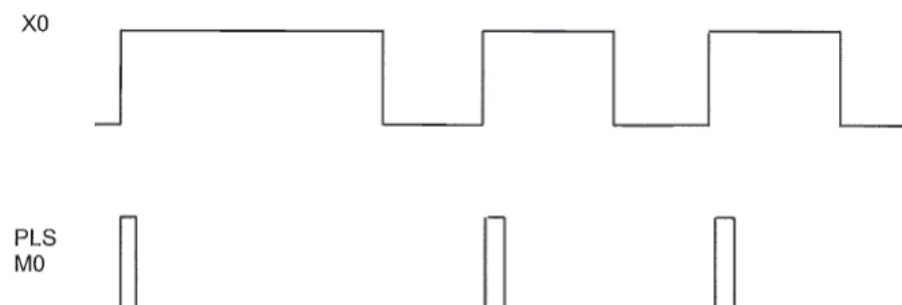
- All normally open contacts CLOSE.
- All normally closed contacts OPEN.

The normally open contact C0 closes, hence energising the Output Coil Y0. Therefore, the circuit gives an output signal on Y0, 10 seconds after the Input X0 closes. Hence the circuit can be considered as an extended timer.

#### ● Line 11

Whenever Input X0 closes, this energises a special function, which is known as a Rising Edge Pulse, PLS.

A pulse circuit only operates on the closing of an input and when energised, the pulse circuit will cause its associated output, the internal memory M0, to energise for a time equal to 1 scan time for the program. The following are the waveforms associated with the PLS circuit.




Even though Input X0 remains closed, the pulse circuit will not re-operate until Input X0 re-opens and closes again.

- Line 14

From the above waveforms, it can be seen that each time input X0 operates, the instruction PLS M0 will be executed and the normally open contact of M0 will momentarily close, hence causing the Counter C0 to be reset to zero. Hence with the operation of Input X0 and the resetting of Counter C0, the cycle will repeat itself.

### Monitoring

Carry out the following:

- Open a new project and give it the name COUNT DELAY.
- Enter the ladder diagram shown on page 18-2.
- Save and convert the program.
- Download the program to the FX PLC.
- Monitor the ladder diagram COUNT DELAY. (To start monitoring, press the F3 or click the  icon.)

## 18.2 Programm Example – Batch Counter

### 18.2.1 BATCH1

BATCH1 is a batch counter program which has the following specification:

- Use input X0 to reset counter C0.
- Use input X1 to input pulses to the counter.
- After 10 input pulses the output Y0 is to come ON.
- Resetting the counter will turn the output Y0 OFF and enable counting to be repeated.

#### Ladder Diagram – BATCH1



#### Principle of Operation

- Line 0  
The momentary operation of input X0 will reset the counter C0. This in turn will cause output Y0 to turn OFF (line 7).
- Line 3  
Each time input X1 closes, this will increment the contents of C0.
- Line 7  
After input X1 has been pulsed 10 times, the C0 contacts will operate to enable the output Y0 to energise.

### 18.2.2 Modification of Program BATCH2

Modify the program BATCH1 so that the following sequence can be obtained.

- After the count value of 10 has been reached output Y0 will be energised and stays ON for just 5 seconds.
- After the 5 second delay the following is to occur:
  - Output Y0 will turn OFF.
  - Counter C0 will automatically be reset.
- Counter C0 than count up once more to 10 and output Y0 will again be energised and stay ON for 5 seconds.

# 19 On Line Programming

Using the ONLINE programming facility of GX Developer, it is possible to modify one line at a time of the project, even though the PLC is in RUN.

In a continuous process, which cannot be stopped i.e. in a steel works, ONLINE programming may be the only way that changes to the program can be carried out.



**CAUTION:**  
**ONLINE programming can be dangerous, since once the modifications have been entered, they become operative on the next scan of the program.**

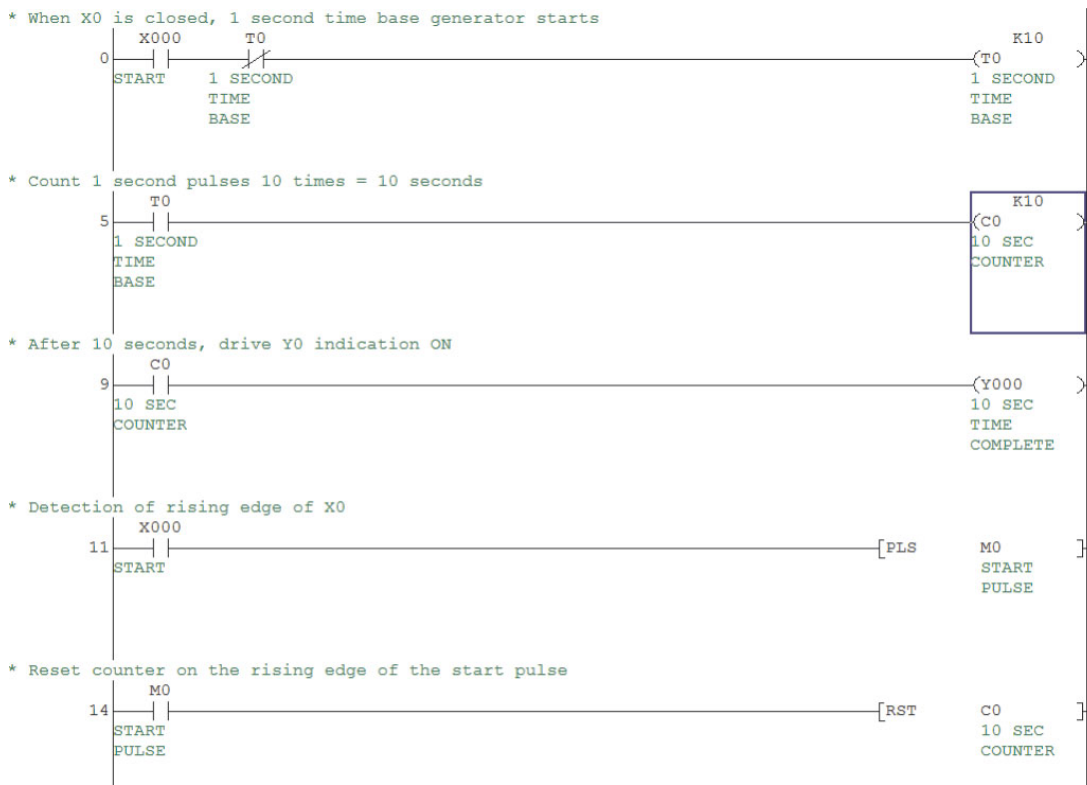
The project COUNT DELAY from the previous chapter is used to demonstrate the use of the online programming facility.

## 19.1 Modification of the program COUNT DELAY

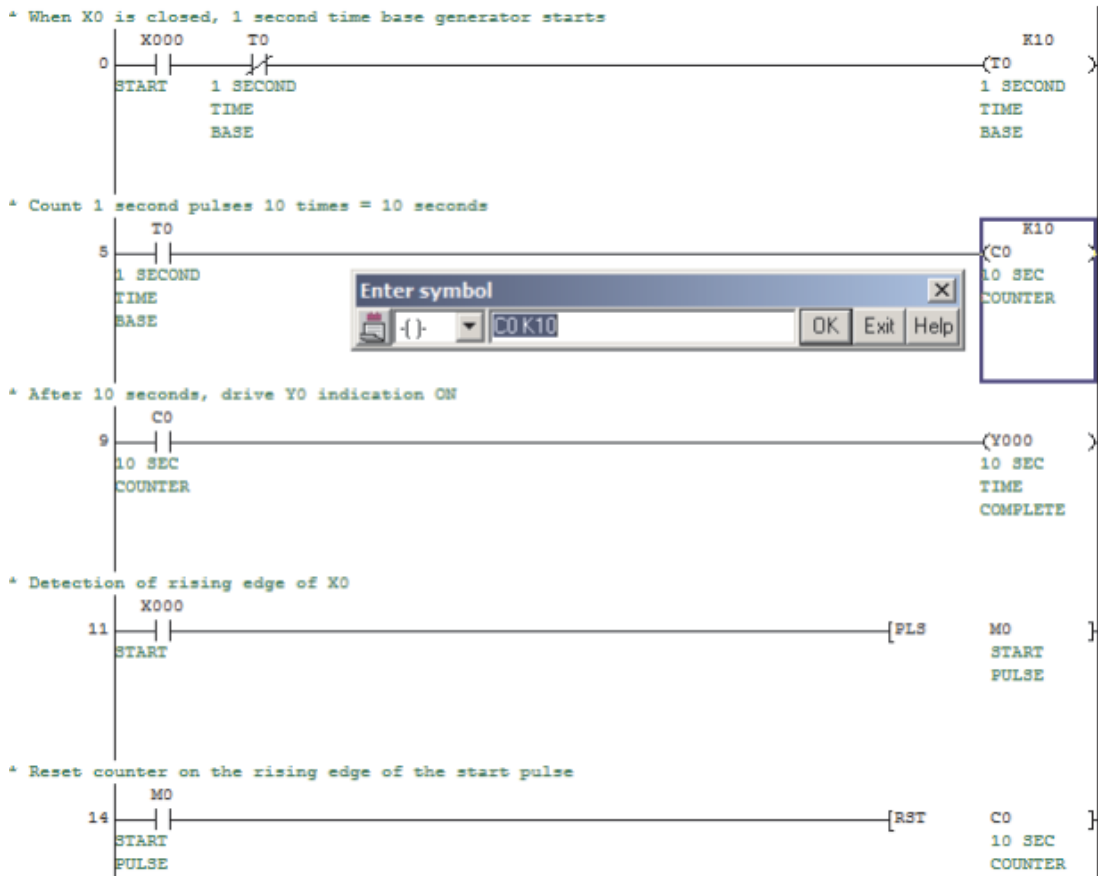
### Counter Value Change

The counter C0 will now have its value changed from "10" (K10) to "20" (K20) whilst the PLC is in RUN mode i.e. an online programming change.

- ① Save COUNT DELAY as COUNT MON.
- ② Open the ladder diagram COUNT MON.
- ③ Ensure the PLC is in RUN.
- ④ Move the cursor to line 5 and over the output -C0 K10- as shown below.



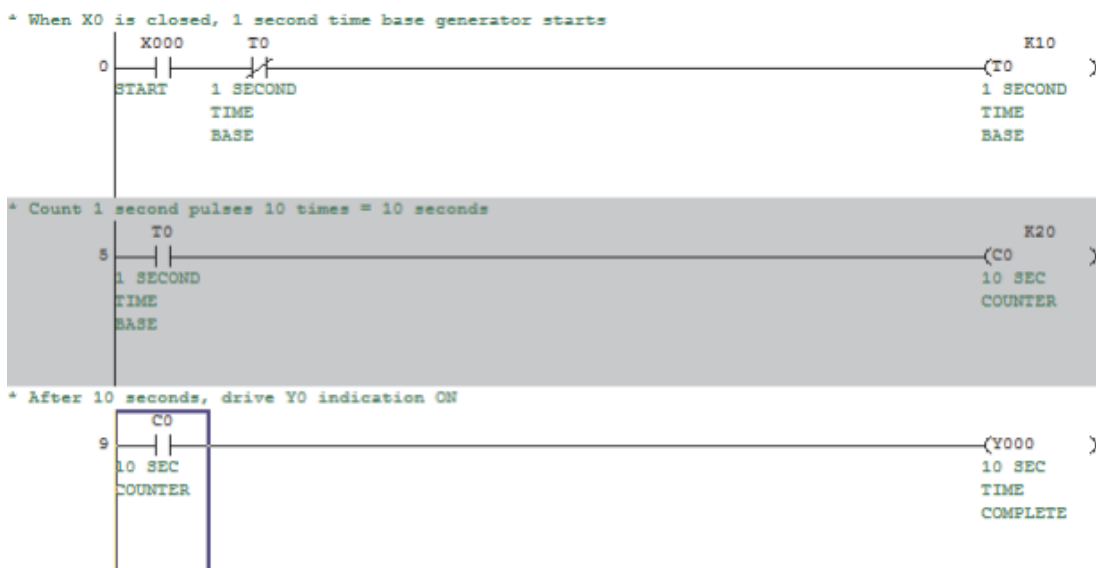
⑤ Double click the left hand mouse button to obtain the output information.



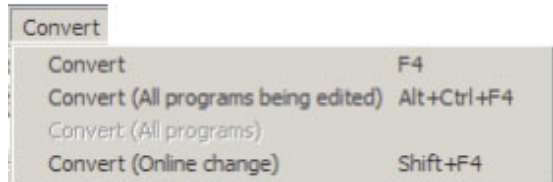
⑥ Change the output details to:



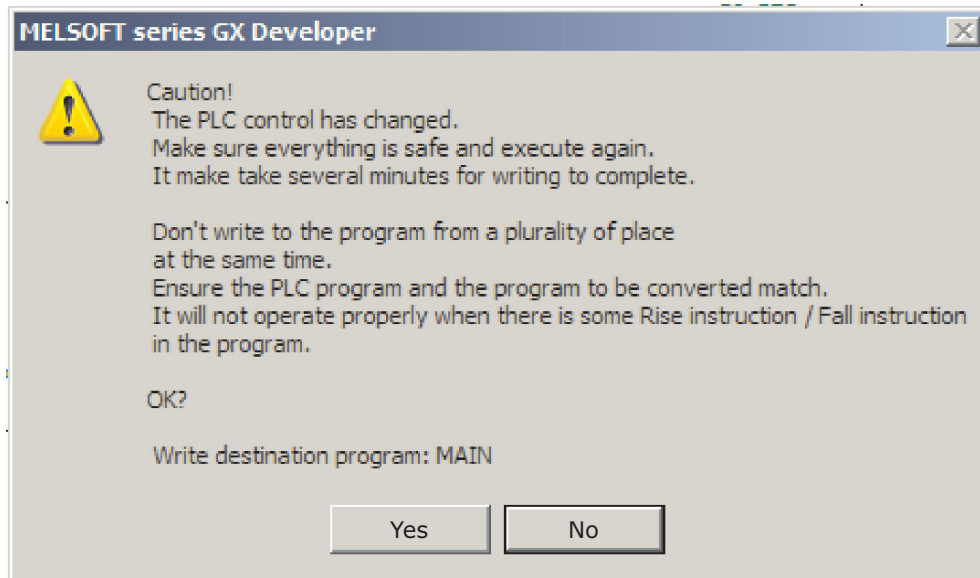
⑦ Select **OK** and the display will appear with line 5 greyed out.



- ⑧ From the menu **Convert**, select **Convert (Online change)**.



- ⑨ The following message is now displayed:



- ⑩ Select **Yes** to enable the program changes to be downloaded to the PLC. The following dialogue will be displayed.

- ⑪ After the online change has taken place the following message is displayed:



- ⑫ Selekt **OK**.
- ⑬ Execute the project COUNT MON and note that the time from the operation of input X0 to when Y0 turns ON will now be 20 s.
- ⑭ Save COUNT MON.





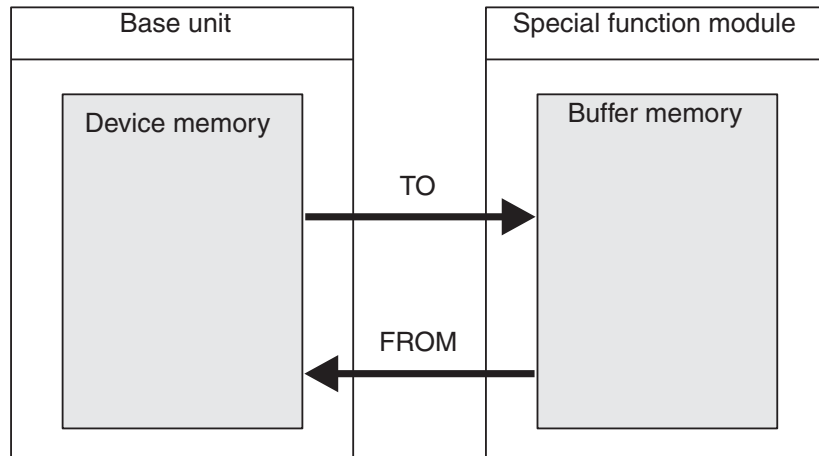
# 20 FROM / TO INSTRUCTIONS

## 20.1 Exchanging Data with Special Function Modules

You can supplement the controller’s functions by adding so-called “special function modules” – for example for reading analog signals for currents and voltages, for controlling temperatures and for communicating with external equipment (see section 2.9).

Communication between the base unit and special function modules is performed with two special applied instructions: the FROM and TO instructions.

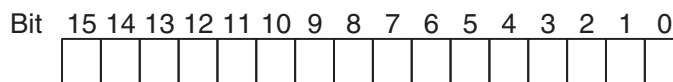
Each special function module has a memory range assigned as a buffer for temporary storage of data, such as analog measurement values or received data. The base unit can access this buffer and both read the stored values from it and write new values to it, which the module can then process (settings for the module’s functions, data for transmission etc).



The buffer memory can have up to 32,767 individual addressable memory cells. The functions of the buffer memory cells depends on the individual special function module – see the module’s documentation for details.

Buffer memory address 0
Buffer memory address 1
Buffer memory address 2
:
:
Buffer memory address n-1
Buffer memory address n

Each buffer memory cell can store 16 bits of data.



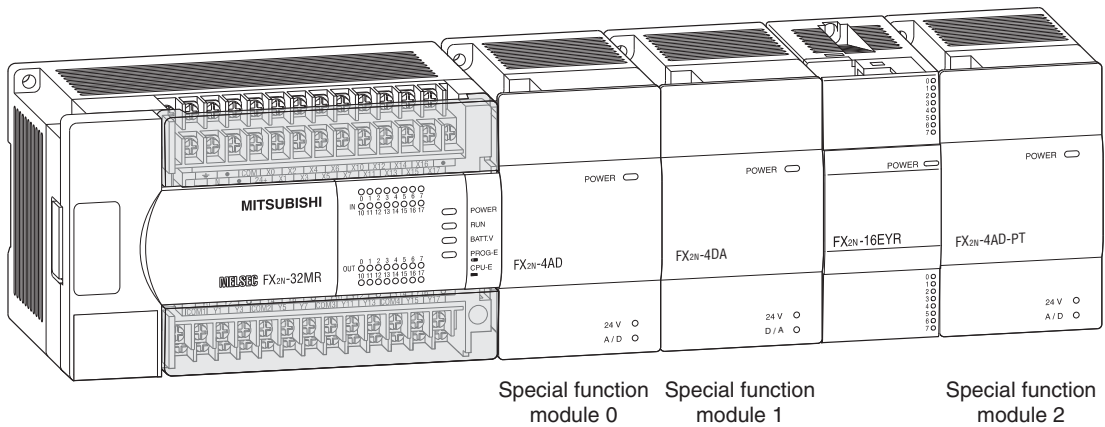
## 20.2 Buffer Memory Accessing Instructions

The following information is required when you use the FROM and TO instructions:

- The special function module to be read from or written to
- The address of the first buffer memory cell to be read from or written to
- The number of buffer memory cells to be read from or written to
- The location in the base unit where the data from the module is to be stored or containing the data to be written to the module

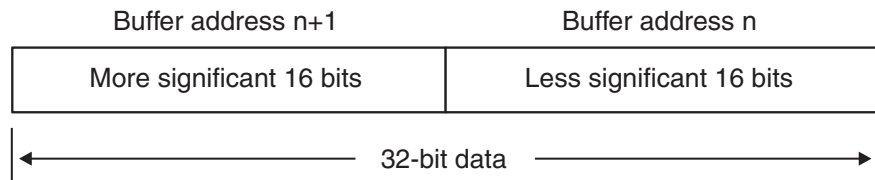
### Special function module address

Since you can attach multiple special function modules to a single controller each module needs to have a unique identifier so that you can address it to transfer data to and from it. Each module is automatically assigned a numerical ID in the range from 0 – 7 (you can connect a maximum of 8 special function modules). The numbers are assigned consecutively, in the order in which the modules are connected to the PLC.



### Starting address in the buffer memory

Every single one of the 32,767 buffer addresses can be addressed directly in decimal notation in the range from 0 – 32,767 (FX1N: 0 – 31). When you access 32-bit data you need to know that the memory cell with the lower address stores the less significant 16 bits and the cell with the higher address stores the more significant bits.



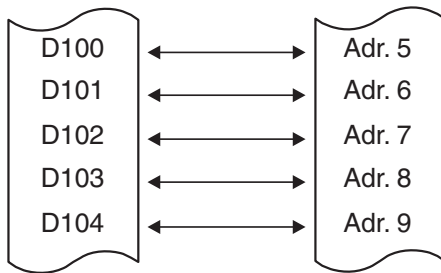
This means that the starting address for 32-bit data is always the address containing the less significant 16 bits of the double word.

### Number of data units to be transferred

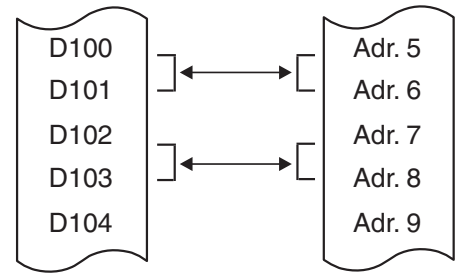
The quantity of data is defined by the number of data units to be transferred. When you execute a FROM or TO instruction as a 16-bit instruction this parameter is the number of words to be trans-

ferred. In the case of the 32-bit versions DFROM and DTO the parameter specifies the number of double words to be transferred.

16-bit instruction  
Units of data: 5



32-bit instruction  
Units of data: 2



The value you can enter for the number of data units depends on the PLC model you are using and whether you are using the 16-bit or 32-bit version of the FROM instruction:

PLC Model	Valid range for no. of data units to be transferred	
	16-Bit Instruction (FROM, TO)	32-Bit Instruction (DFROM, DTO)
FX2N	1 to 32	1 to 16
FX2NC	1 to 32	1 to 16
FX3U	1 to 32767	1 to 16383

**Data destination or source in the base unit**

In most cases you will read data from registers and write it to a special function module, or copy data from the module’s buffer to data registers in the base unit. However, you can also use outputs, relays and the current values of timers and counters as data sources and destinations.

**Pulse-triggered execution of the instructions**

If you add a P suffix to the instructions the data transfer is initiated by pulse trigger. In some applications it is better if the value is written to the destination or read from the source in one program cycle only. For example, you will want to do this if other instructions in the program also write to the same destination or if the FROM or TO operation must be performed at a defined time.

If you add a “P” to the FROM or TO instruction (FROMP, TOP) it will only be executed **once**, on the rising edge of the signal pulse generated by the input condition.



**CAUTION:**

*Most buffer memories have a read-only area at some of its addresses. Data must not be written to this area. Otherwise buffer memory data will be corrupted, and sometimes the special function module may become faulty.*

### 20.2.1 Read Buffer Memory (FROM)

The FROM instruction is used to transfer data from the buffer of a special function module to the controller base unit. Note that this is a copy operation – the contents of the data in the module buffer are not changed.

Four different instructions in the following table are available as read instructions:

Execution	16-Bit Data (1-Word Data)	32-Bit Data (2-Word Data)
Executed any time when condition switches on	FROM	DFRO
Executed on leading edge of condition	FROMP	DFROP

#### Ladder Diagram



#### Instruction List

0 FROM K0 K9 D0 K1

①      ②      ③      ④

- ① Special function module address (0 to 7)
- ② Starting address in buffer (FX1N: 0 – 31, FX2N, FX2NC and FX3U: 0 – 32,766). You can use a constant or a data register containing the value.
- ③ Data destination in the controller base unit
- ④ Number of data units to be transferred

The example above uses FROM to transfer data from an FX2N-4AD analog/digital converter module with the address 0. The instruction reads the current value of channel 1 from buffer address 9 and writes it to data register D0.

The next example shows how the 32-bit version of the instruction is used to read data from address 2 in the special function module. The instruction reads 4 double words starting at buffer address 8 and writes them to data registers D8 – D15.



The next example illustrates the use of the pulse triggered version, FROMP. Here the contents of the four buffer addresses 0 – 3 are only transferred to data registers D10 – D13 when the signal state of the input condition changes from “0” to “1”.



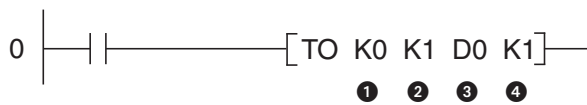
## 20.2.2 Write to Buffer Memory (TO)

The TO instruction transfers (writes) data from the controller base unit to the buffer of a special function module. Note that this is a copy operation, it does not change the data in the source location.

Four different instructions shown in the following table are available as write instructions:

Execution	16-Bit Data (1-Word Data)	32-Bit Data (2-Word Data)
Executed any time when condition switches on	TO	DTO
Executed on leading edge of condition	TOP	DTOP

### Ladder Diagram



### Instruction List

```
0 TO      K0   K1   D0   K1
          ①   ②   ③   ④
```

- ① Special function module address (0 – 7)
- ② Starting address in buffer (FX1N: 0 – 31, FX2N, FX2NC and FX3U: 0 – 32,766). You can use a constant or a data register containing the value.
- ③ Data source in the controller base unit
- ④ Number of data units to be transferred

In the example above the contents of data register D0 are copied to the buffer address 1 of special function module number 0.



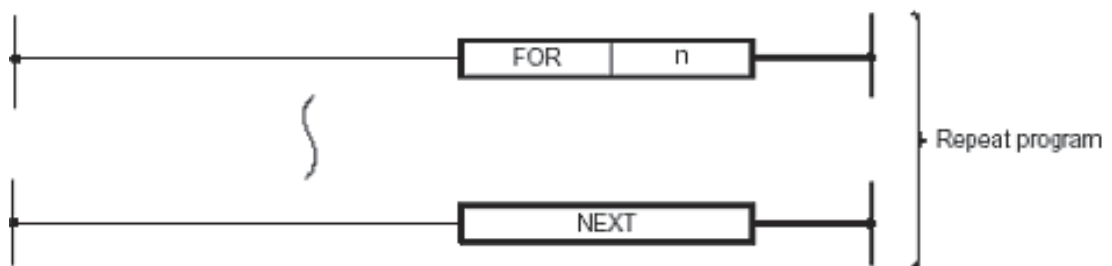
## 21 FOR – NEXT Loops

FOR-NEXT loops have many uses and are often used to enable multiple processing of a common algorithm or process on different address points.

FOR-NEXT processing may also be used in search routines to enable specific information to be retrieved from data tables and files stored in the PLC; actions may subsequently be carried out on the results obtained from the lookup process.

### 21.1 Operation

FOR - NEXT Loops operate by interrupting the program flow by holding the scan process in a loop that executes n number of times:



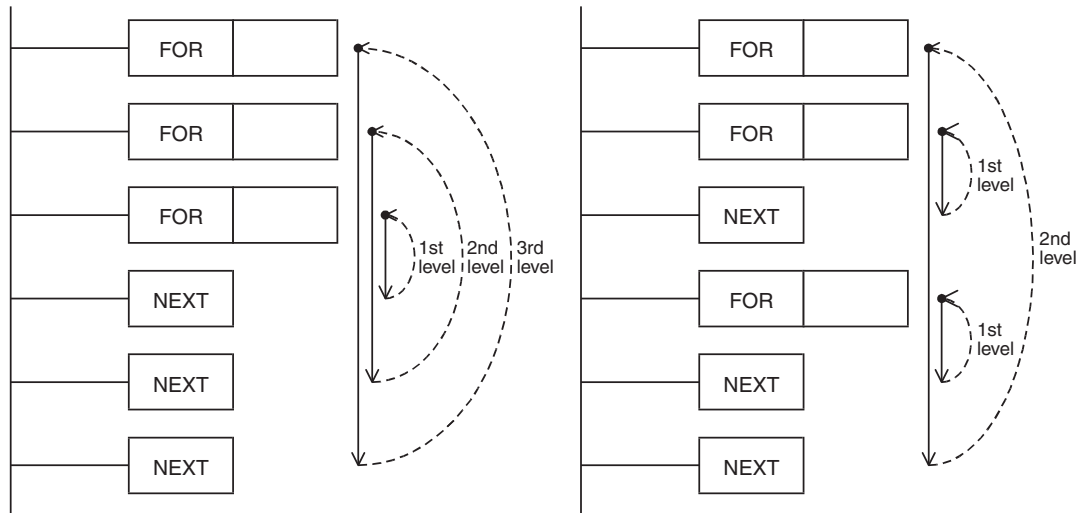
It is common practice to use a Conditional Jump (CJ) to bypass a FOR-NEXT loop if it is not required to be scanned. This will avoid scanning the loop when it is not required, thus minimising the overall program scan time.

#### Points to note

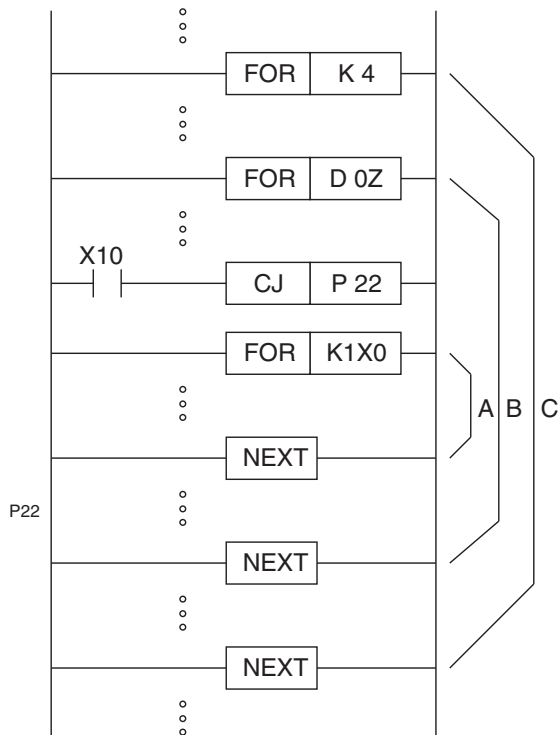
- The FOR instruction operates in a 16 bit mode hence, the value of the operand n may be within the range of 1 to 32,767. If a number between the range -32,768 and 0 (zero) is specified it is automatically replaced by the value 1, i.e. the FOR-NEXT loop would execute once.
- The NEXT instruction has NO operand.
- The FOR-NEXT instructions must be programmed as a pair e.g. for every FOR instruction there **must** be an associated NEXT instruction. The same applies to the NEXT instructions, there **must** be an associated FOR instruction. The FOR-NEXT instructions must also be programmed in the correct order. This means that programming a loop as a NEXT-FOR (the paired NEXT instruction proceeds the associated FOR instruction) is **not** allowed.  
Inserting an FEND instruction between the FOR-NEXT instructions, i.e. FOR-FEND-NEXT, is **not** allowed. This would have the same effect as programming a FOR without a NEXT instruction, followed by the FEND instruction and a loop with a NEXT and no associated FOR instruction.
- A FOR-NEXT loop operates for its set number of times **before** the main program is allowed to finish the current program scan.
- When using FOR-NEXT loops care should be taken not to exceed the PLC's watchdog timer setting. The use of the WDT instruction and/or increasing the watchdog timer value is recommended.

### Nested FOR-NEXT loops

FOR-NEXT instructions can be nested for 5 levels. This means that 5 FOR-NEXT loops can be sequentially programmed within each other.



In the example shown below a 3 level nest has been programmed. As each new FOR-NEXT nest level is encountered the number of times that loop is repeated is increased by the multiplication of all of the surrounding/previous loops.



For example, loop C operates 4 times. But within this loop there is a nested loop, B. For every completed cycle of loop C, loop B will be completely executed, i.e. it will loop the number of times written in D0Z. This again applies between loops B and A.

The total number of times that loop A will operate for **one** scan of the program will equal:



- Actual number of loop A operations: (No. of loop C operations) multiplied by  
(No. of loop B operations) multiplied by  
(No. of loop A operations)

The total number of times that loop B will operate for **one** scan of the program will equal:

- Actual number of loop B operations: (No. of loop C operations) multiplied by  
(No. of loop B operations)

If values were associated to loops A, B and C, e.g. 7, 6 and 4 respectively, the following number of operations would take place in **one** program scan:

Number of loop C operations = 4 times

Actual number of loop B operations = 24 times (C x B, 4 x 6)

Actual number of loop A operations = 168 times (C x B x A, 4 x 6 x 7)

In this example the use of the CJ programming feature, causing the jump to label P22 allows the 'selection' of which loop will be processed and when, i.e. if X10 was switched ON, loop A would no longer operate.



## Set up and Monitoring Procedure

### Exercise

- ① Enter the above program example and save it as 'For-Next1'
- ② Transfer the program to the PLC.
- ③ Before operating the program, fill data registers D1-D5 with random values between 0 and 32767 decimal, using the **Device Test** feature in GX Developer (described in section 14.4).
- ④ Monitor the ladder program and use **Batch Monitor** in GX Developer to view the contents of the data registers D1-D5.
- ⑤ Using the **Device Test** feature, input a value (corresponding to one of the forced values set in ③ above) between 0 and 32767 decimal to D10.
- ⑥ Operate the X0 switch momentarily and observe the signals of the digital outputs of the training rig. If a data match is made within the data register file D1 – D5, then the readout will produce a value corresponding to the record number (Data register number) from 1 to 5 of the data match.

Example: If the value entered in D10 corresponds to a value stored in D2 (record #2), output Y2 will be switched ON.

The corresponding output will stay ON until a new search is executed. If no match is found Y0 is switched ON.



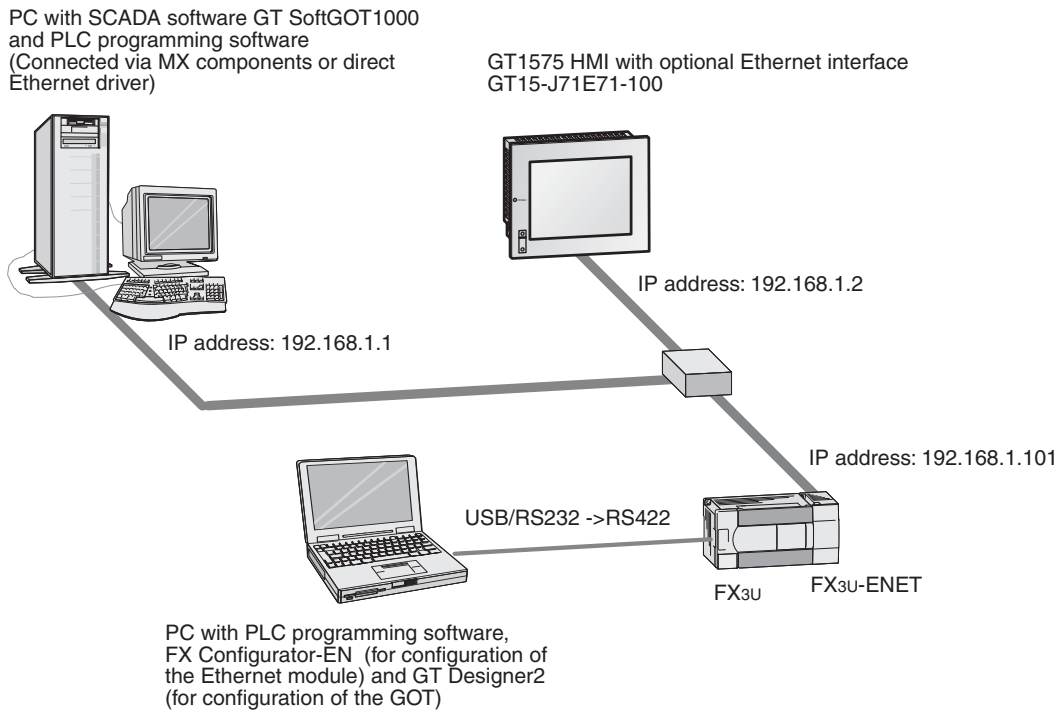
# 22 Ethernet Communications

This section provides a step-by-step guide to setting up a Ethernet module FX3U-ENET using FX Configurator-EN.

As an example, this section will show how to set up a module for allowing TCP/IP communications between a FX3U, a SCADA PC and a GT15 HMI\*. Also shown is how the programming software can be configured to communicate with the FX3U via Ethernet once the settings have been made.

The diagram below shows the layout of the example Ethernet network. Proposed IP addresses are shown next to the Ethernet nodes.

Please note that more attention is given to the set up of the PLC than the PC or HMI, as the user may require more specific settings than this section covers.

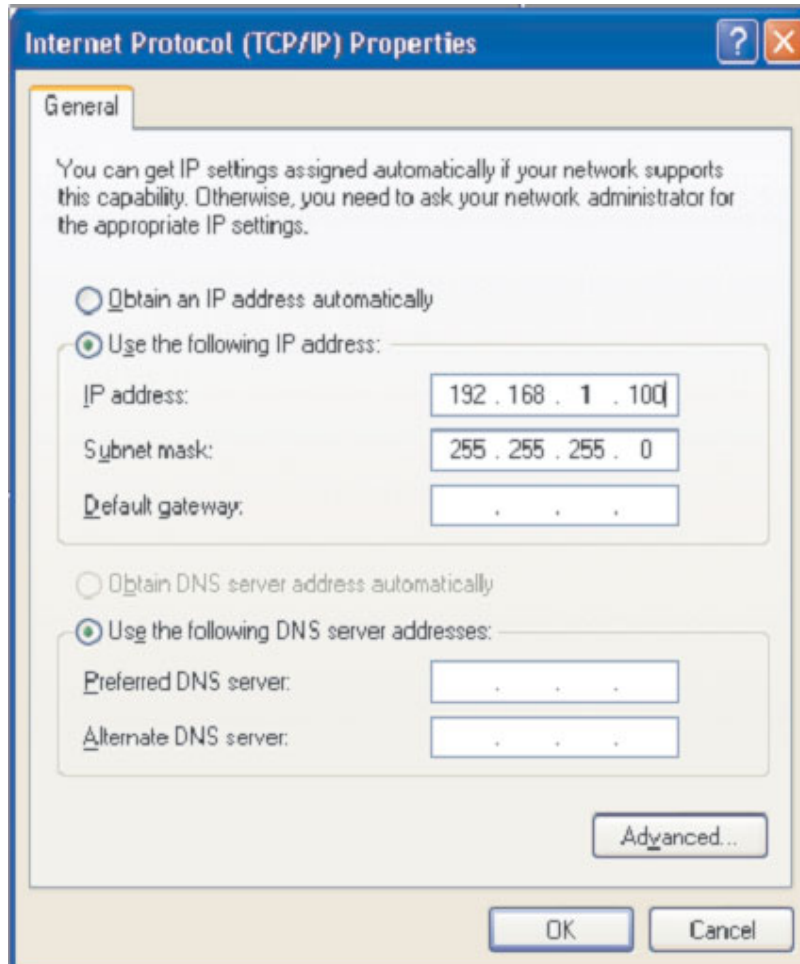


\* For the case that a HMI of the E1000 series is used instead of a GOT, the settings in the software E-Designer are shown in section 22.5.

## 22.1 Configuring the PC on the Ethernet

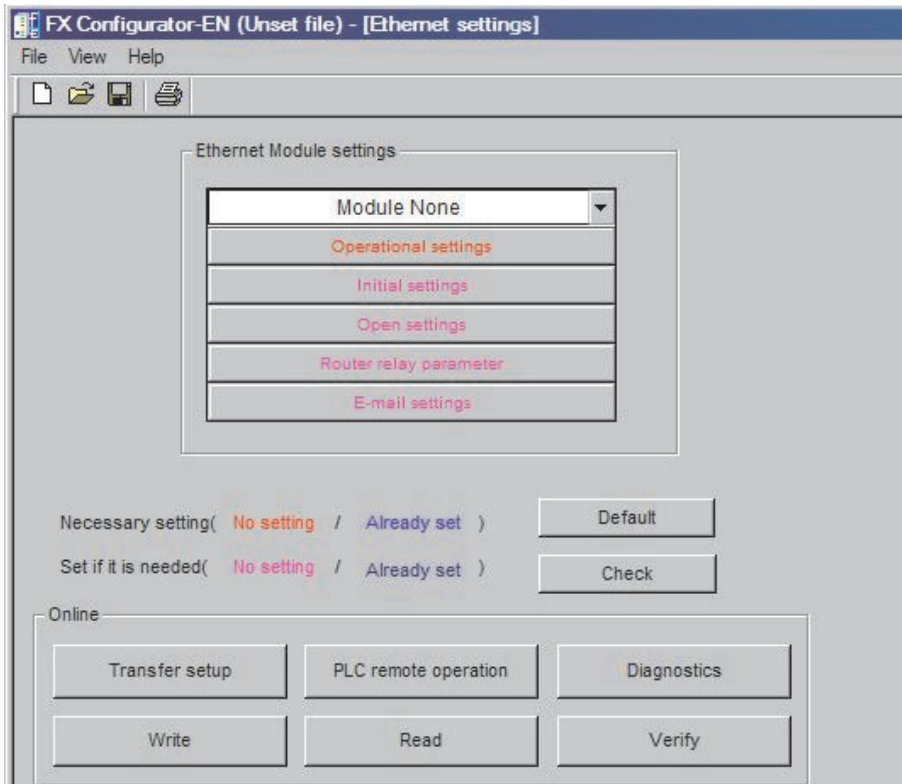
Open the Network properties of Windows, and assign an IP address and subnet mask in the TCP/IP properties dialogue for the Ethernet network adapter to be used in the PC.

Please note that after changing IP address, the PC may require a restart.

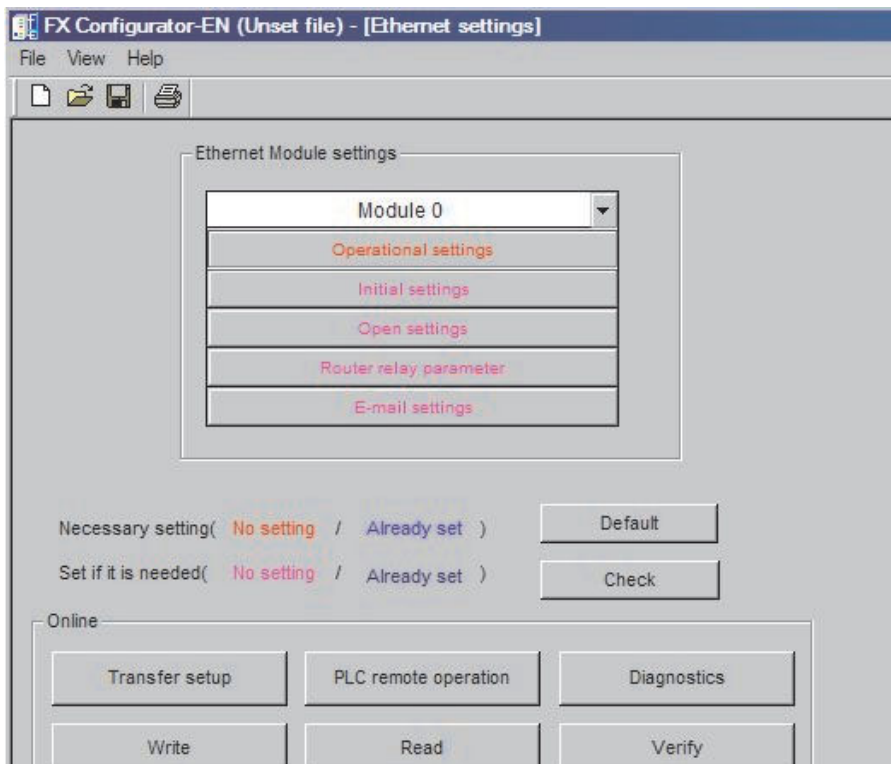


## 22.2 Configuring the FX3U-ENET by FX Configurator-EN

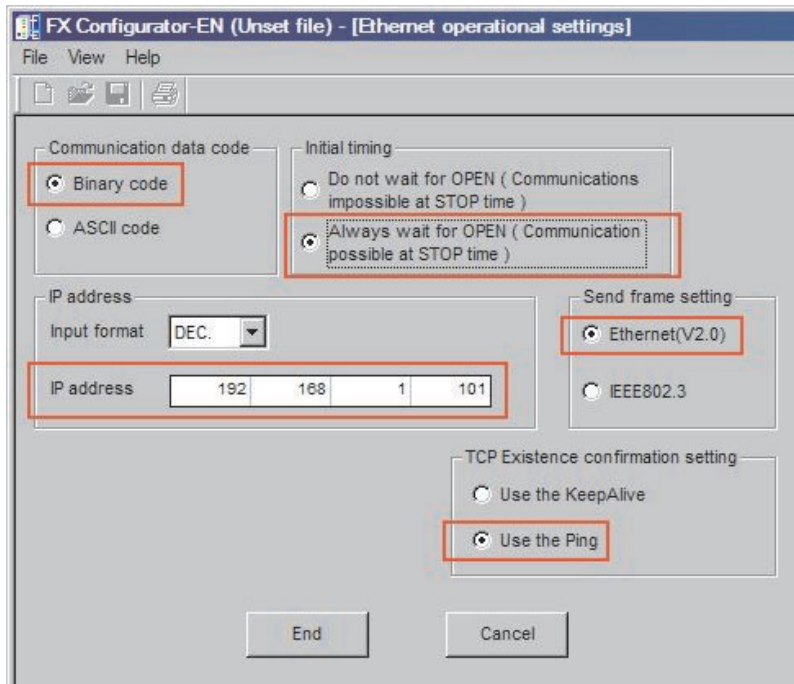
- ① Open the FX Configurator-EN and start the setting of the ETHERNET module FX3U-ENET.



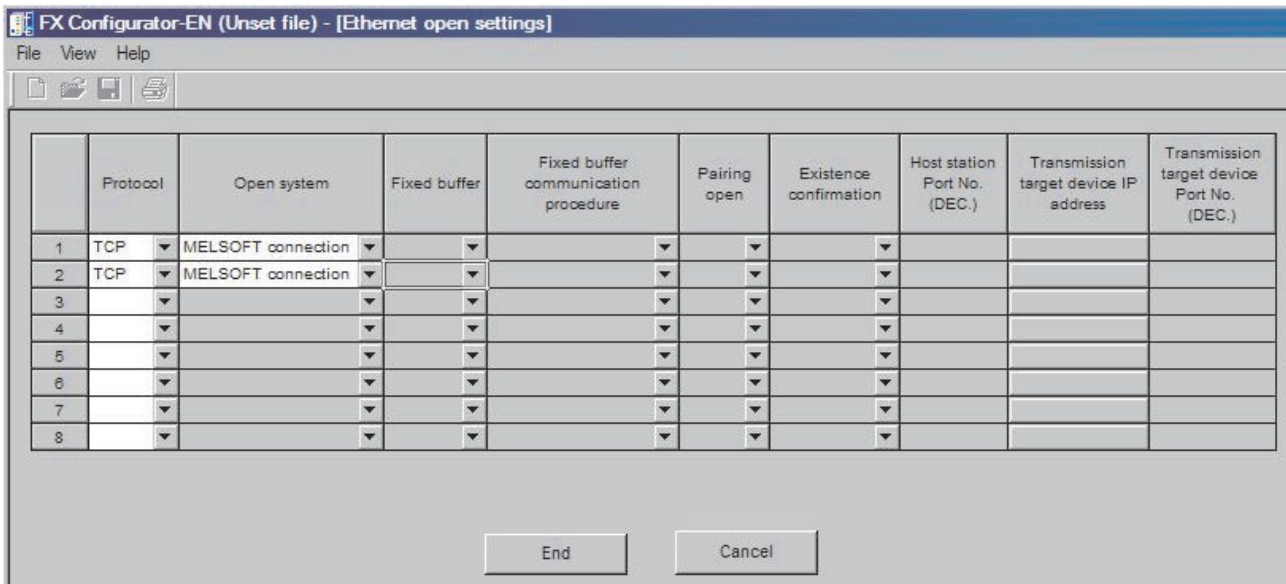
- ② Now select the special function module address of the FX3U-ENET. Special function modules connected to the right side of the base unit are counted from left to right. If the FX3U-ENET is the first special function module select **Module 0**.



- ③ Open the **Operational Settings** and take over the settings shown below in red frames. The IP address 192.168.1.101 of the FX3U-ENET corresponds to the requirements of your network if your network IP is 192.168.1.1.

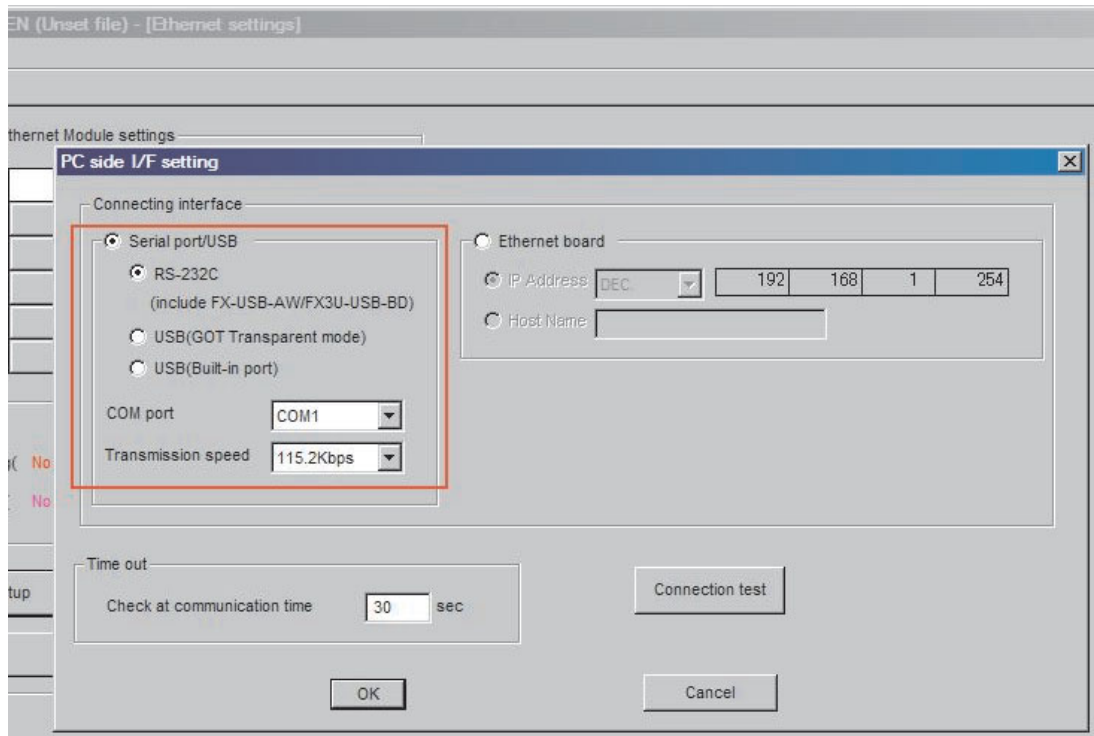


- ④ Next, open the **Open Settings** and take over the following settings.



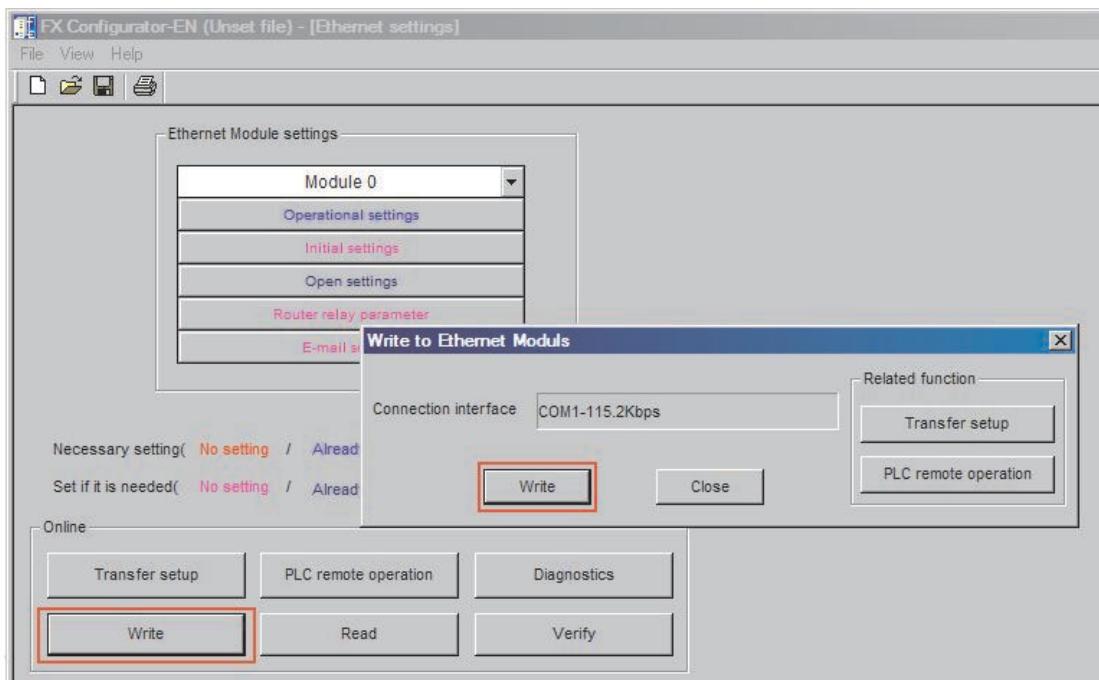


- ⑤ In the **Ethernet Module settings**, click on **Transfer Setup** and take over the settings shown below in the red frame.



- ⑥ Click on **Write** in the **Ethernet Module settings**. As you see, the transfer speed for COM1 is set to 115.2 Kbps.

In the dialogue window **Write to Ethernet Module** click on **Write** and transfer your settings to the PLC. Confirm displayed messages with **YES** resp. **OK**.



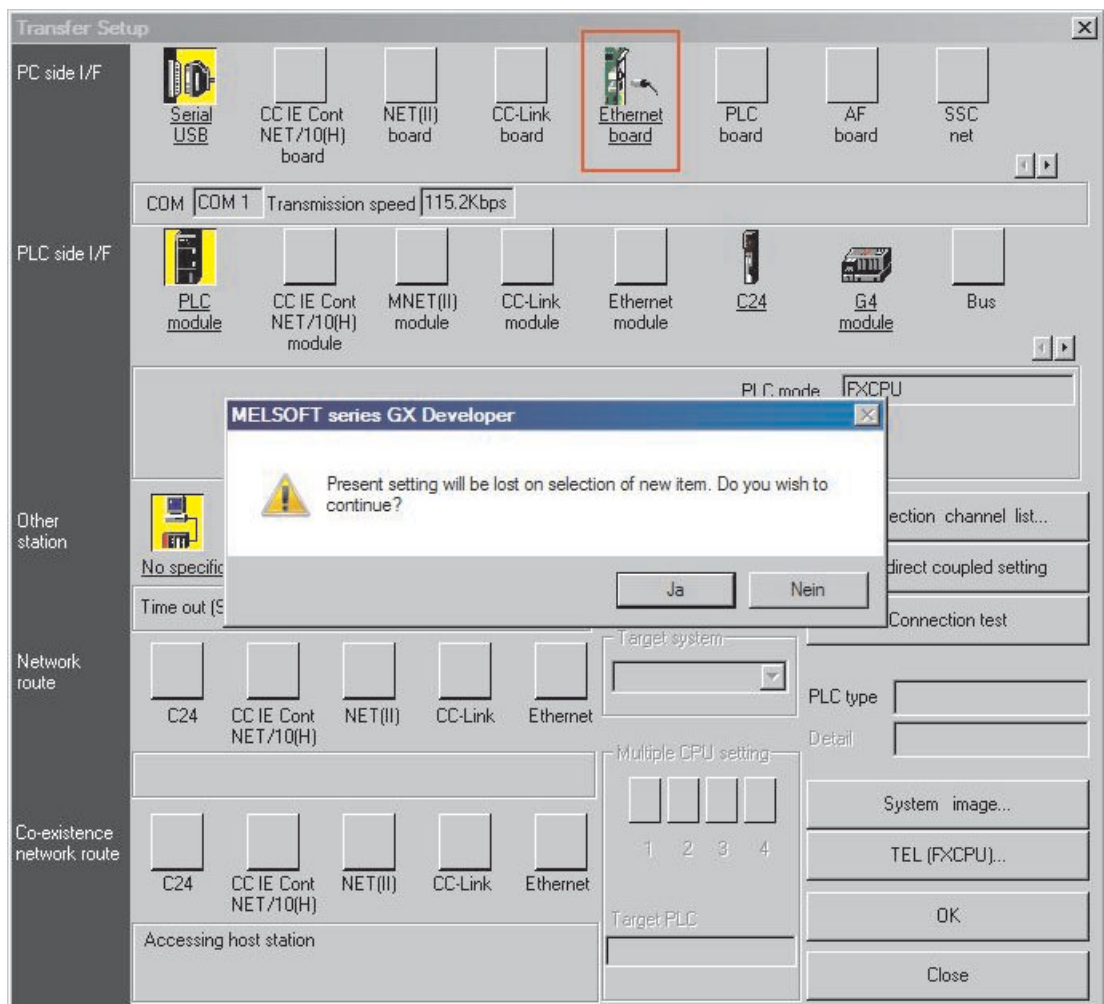
Now you can confirm the completion of the initial processing by issuing a PING command to the FX3U-ENET. The ping command is provided by Microsoft® Windows. Shown below is an example for normal completion.

```
C:\>ping 192.0.1.254...Remark: Execute the ping command  
Pinging 192.168.1.101 with 32 bytes of data:  
Reply from 192.168.1.101: bytes=32 time=1ms TTL=250  
Reply from 192.168.1.101: bytes=32 time=1ms TTL=250  
Reply from 192.168.1.101: bytes=32 time=1ms TTL=250  
Reply from 192.168.1.101: bytes=32 time=1ms TTL=250  
Ping statistics for 192.168.1.101:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)  
Approximate round trip times in milli-seconds:  
Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

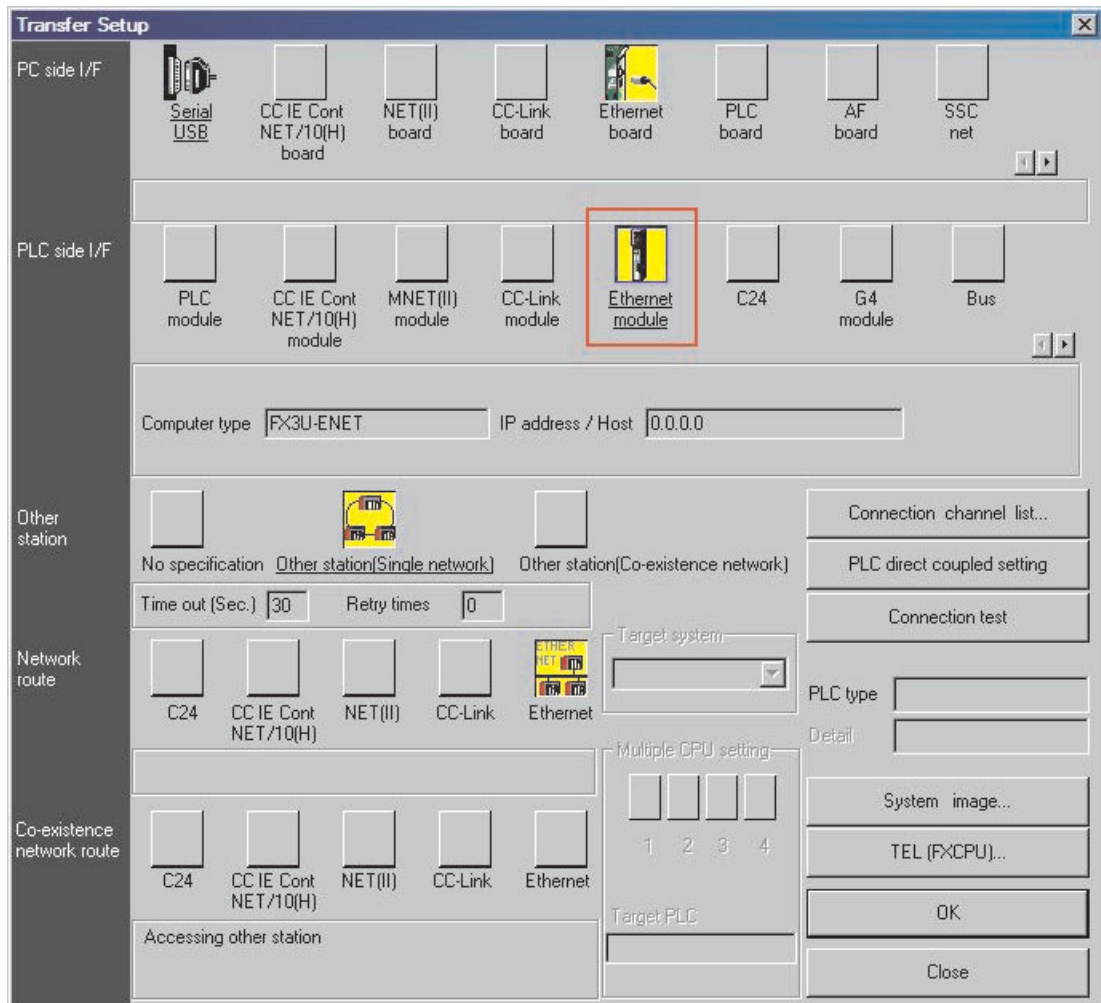
## 22.3 Configuring GX Developer to access the PLC on Ethernet

Please make the following settings in order to access the PLC via an Ethernet network and an Ethernet interface module.

- ① From the **Online** menu, select **Transfer Setup**:



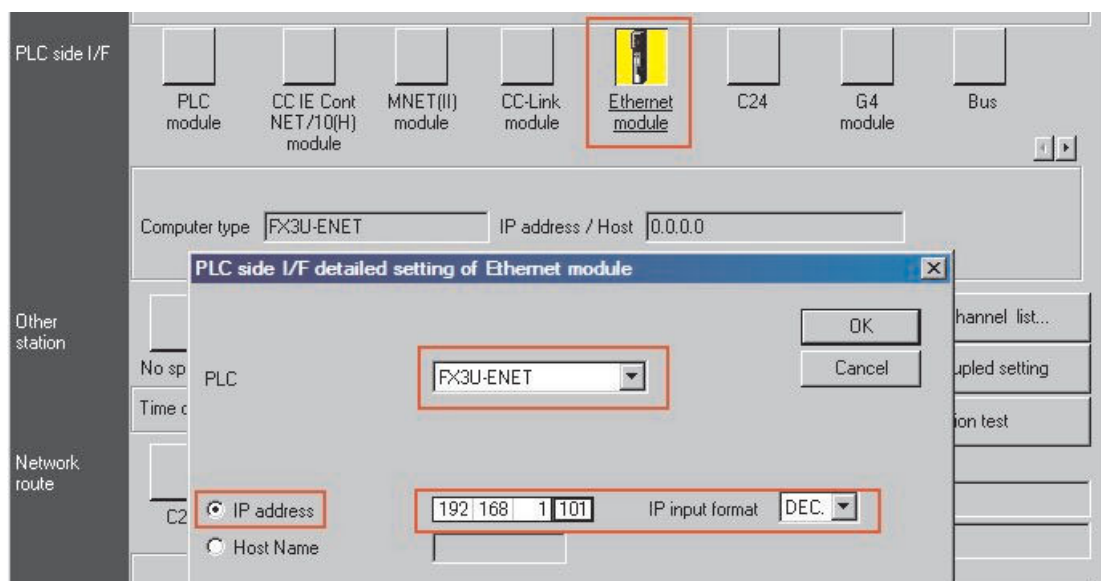
- ② The default connection is for the **PC Side I/F** to use serial connection to the PLC CPU module. Change the **PC Side I/F** to **Ethernet board** by clicking on it as shown above, and saying **Yes** to the question about present setting will be lost (i.e. the setting of serial to CPU).



③ Next, double click on **Ethernet module** under **PLC side I/F** as shown above. This will open up the dialogue to allow the settings for the Ethernet interface module used.

**NOTE**

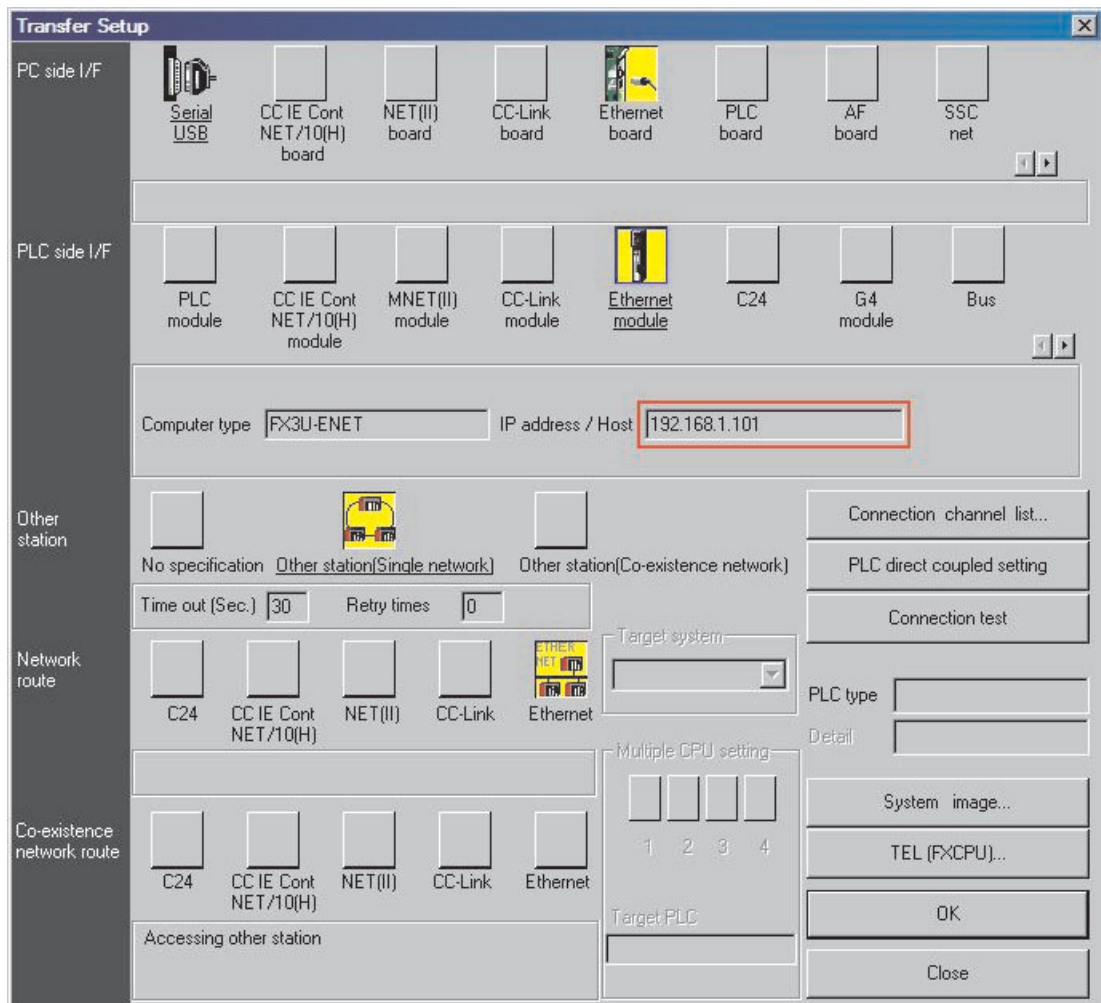
There is no need to specify a port number, as the programming software will use a MELSOFT Protocol dedicated port by default.



④ Click **OK** when done.

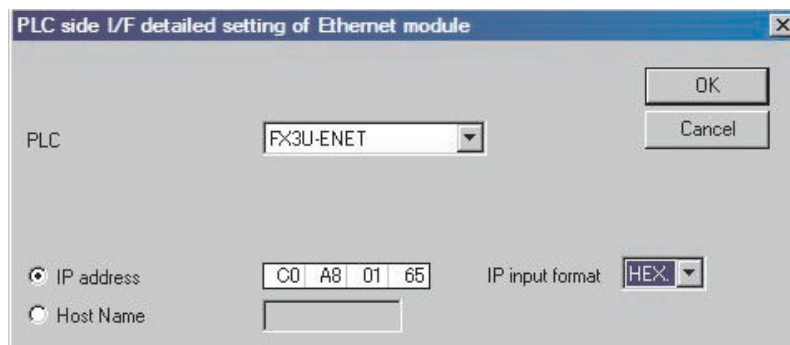
This will complete the setting, making the dialogue look as shown below.

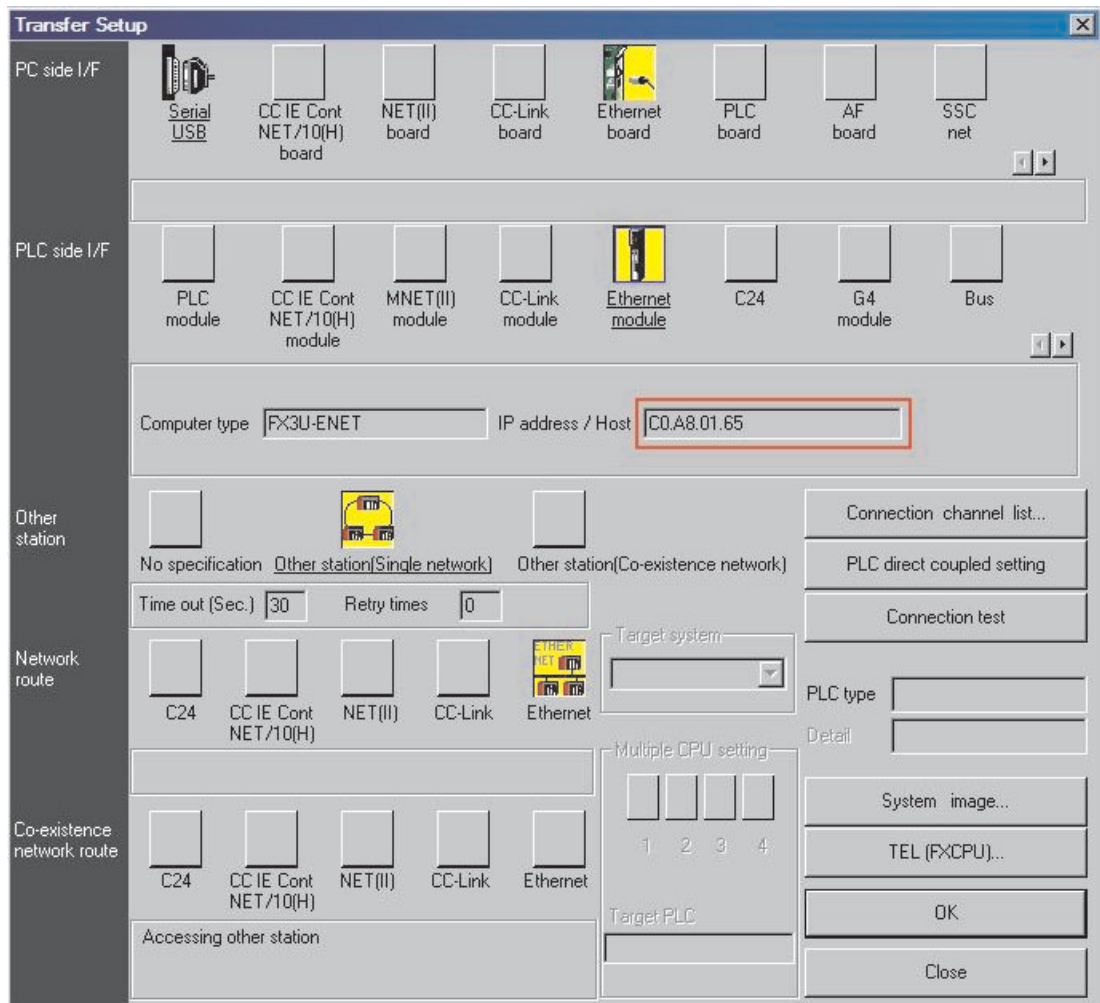
- ⑤ Click **Connection test** to confirm the settings are correct. Then click **OK** when finished.



**NOTE**

The IP address can be entered also in hexadecimal format. This option is shown in the following two figures.

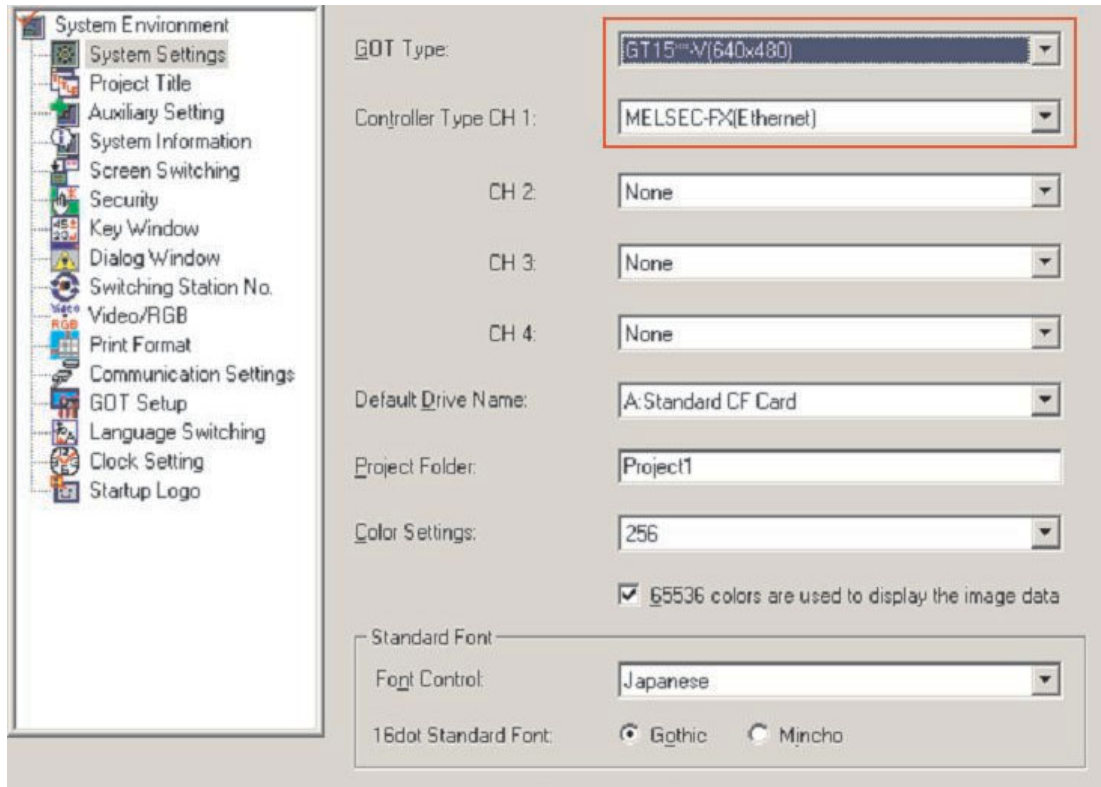




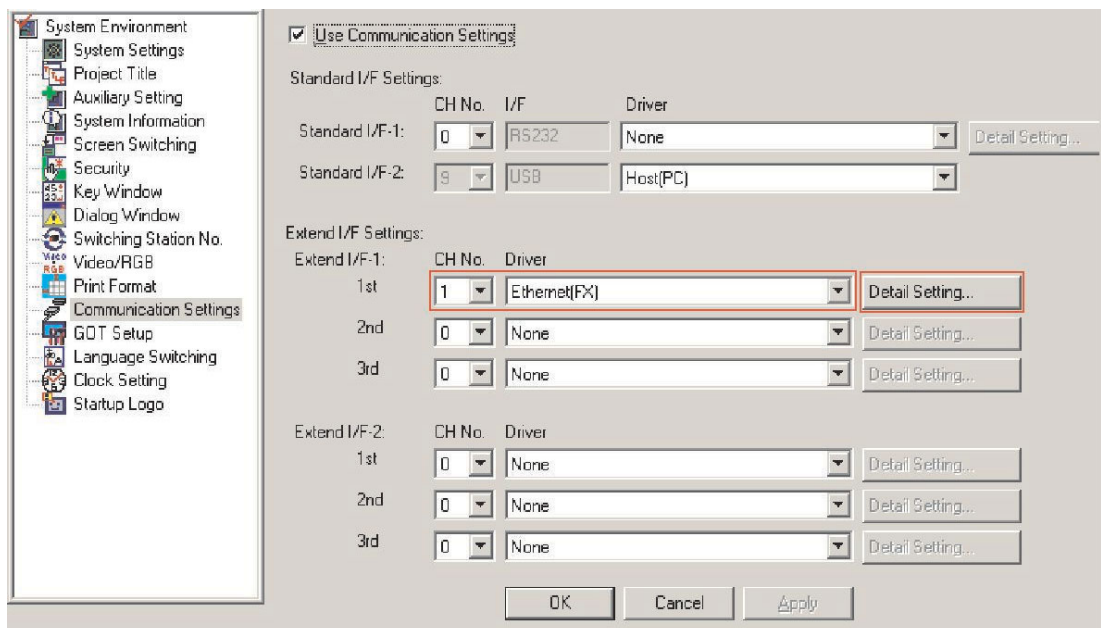


## 22.4 Setting up a HMI of the GOT1000 Series (GT12, GT15 or GT16)

- 1 Please start GT-Designer2 and open a new project. In the Project Navigator window, double click on **System Settings** and make the settings shown below.

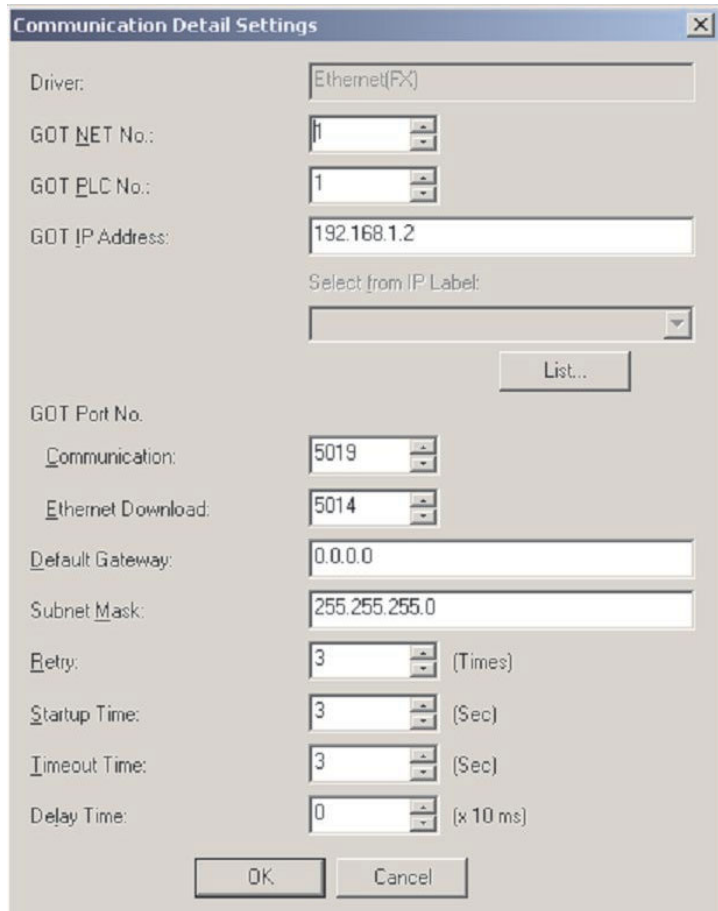


- 2 In the **System Environment**, double click on **Communication Settings** and make the settings shown below.

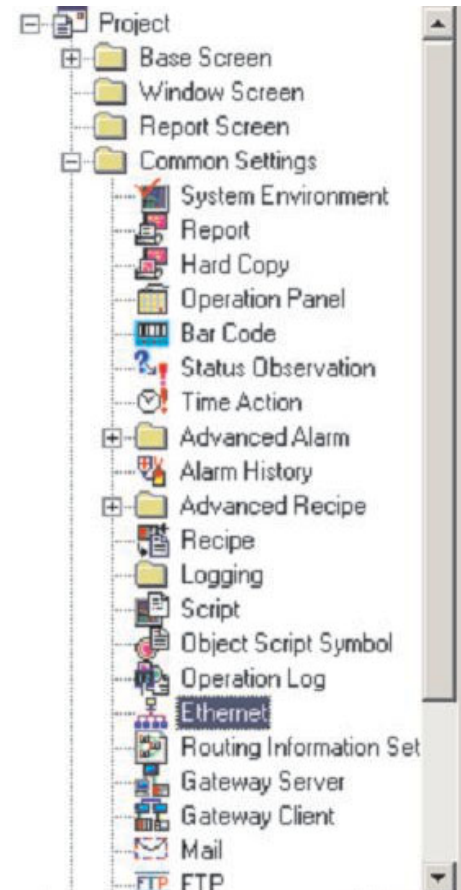


- 3 Then, in the **Communication Settings** window, click on **Detail Setting**.

- ④ As shown on the right, enter the details of the network in use and the IP address of the GOT.

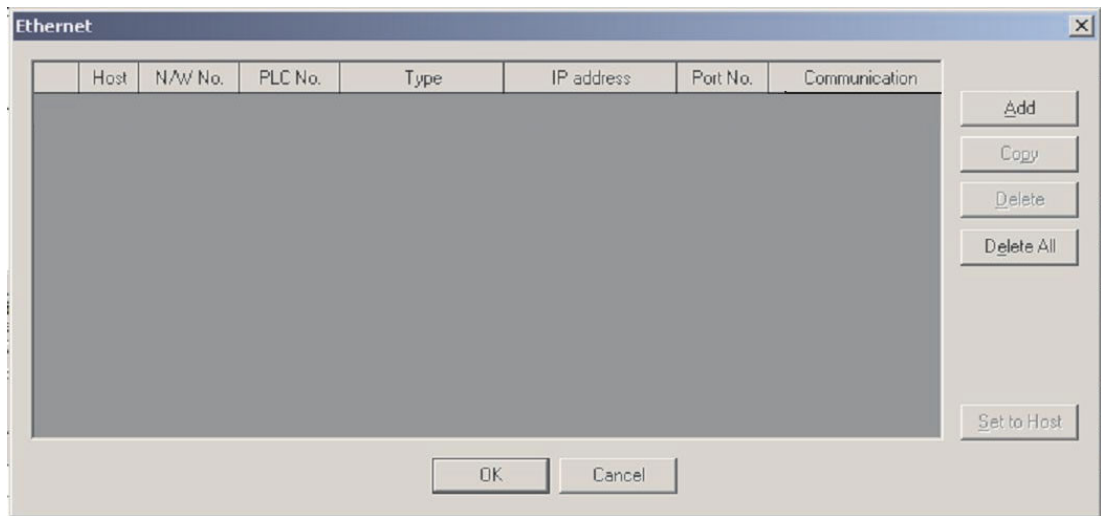


- ⑤ In the Project Navigator, select: **Common Settings -> Ethernet** to set up the connected PLC and the associated IP address.





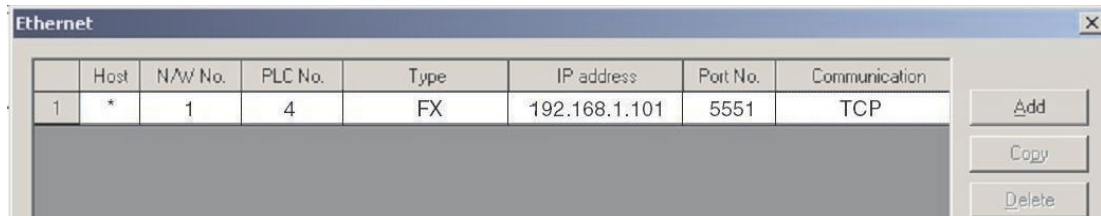
- ⑥ The following dialogue window will be displayed. Click on **Add**.



- ⑦ Click on the **Type** column and select the type of PLC.



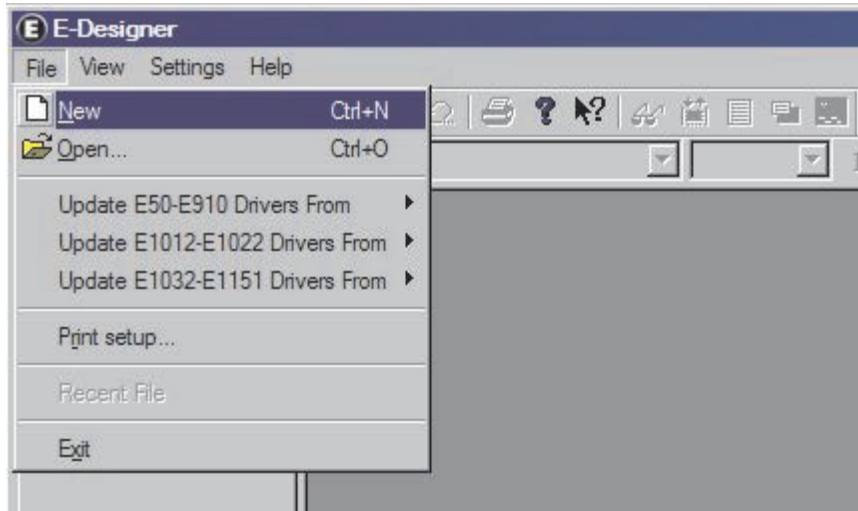
- ⑧ When selecting the PLC, certain settings (e.g. the Port No.) are taken over as defaults. Please make the remaining settings.



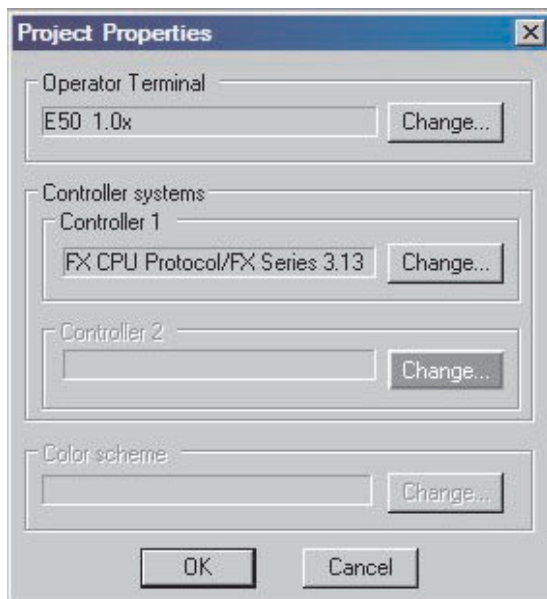
## 22.5 Setting up a HMI of the E1000 series

The settings shown in this section are only necessary when a HMI of the E1000 series is connected to the network instead of a GOT series HMI.

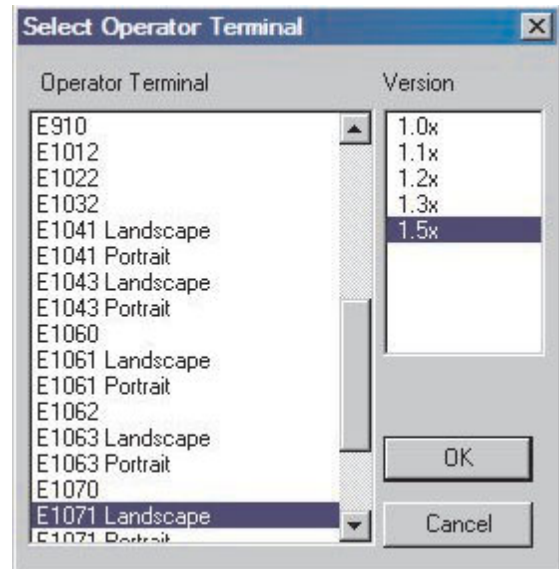
- ① Please open a new E-Designer project.



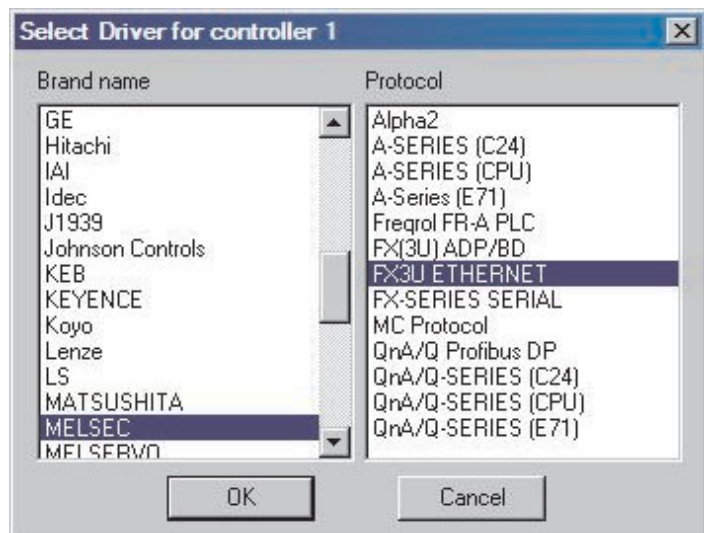
- ② Next, a dialogue window allowing the settings for the HMI used and the connected PLC is opened.



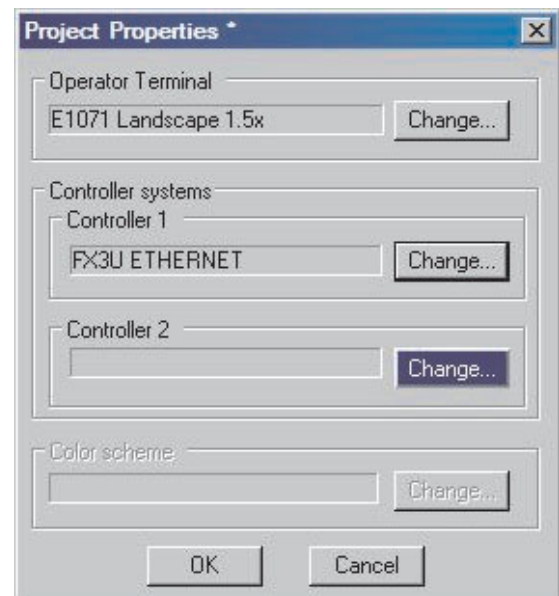
③ Select the operator terminal.



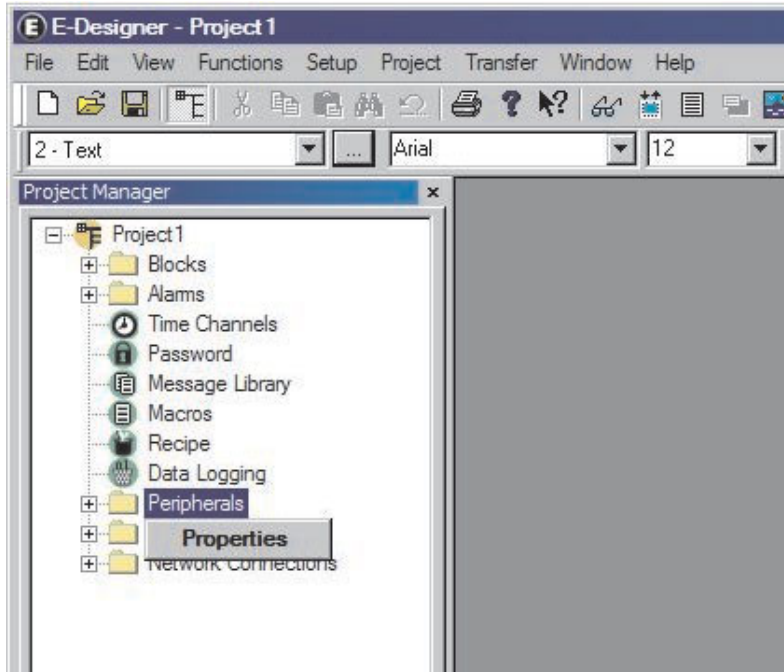
④ Select the PLC



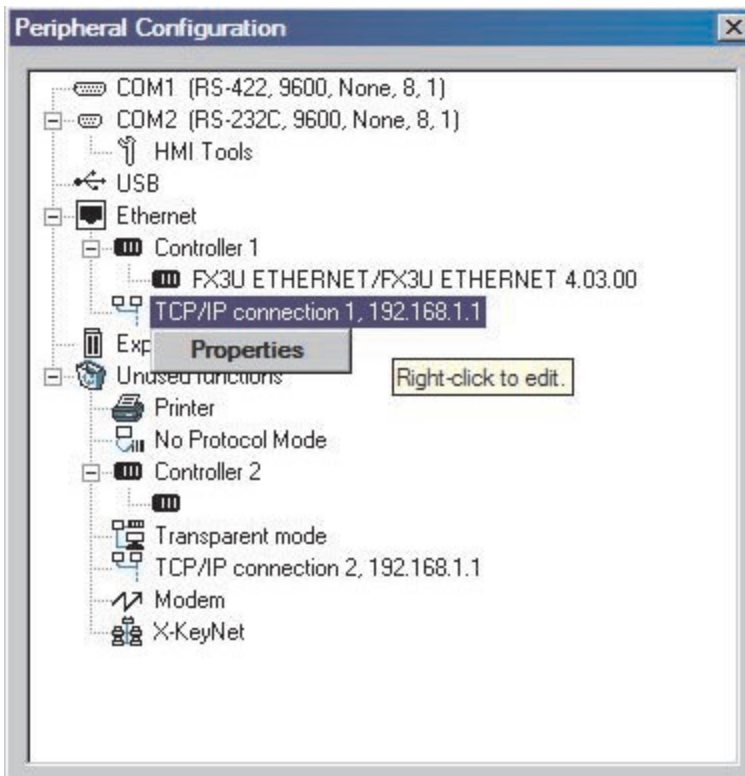
⑤ Click **OK** to confirm the selection.



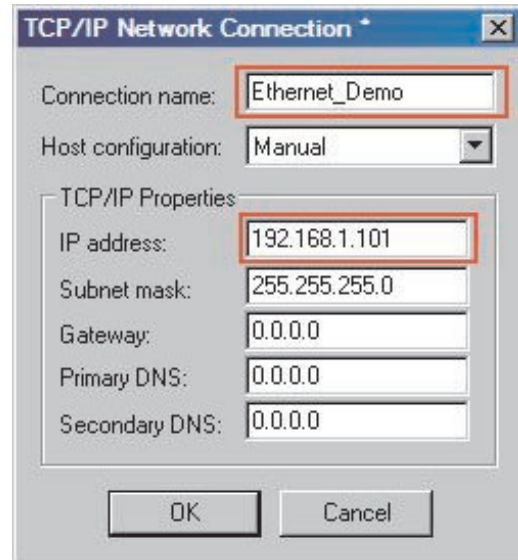
- ⑥ Open the properties of the peripherals. (Right click on **peripherals**, than click on **properties**.)



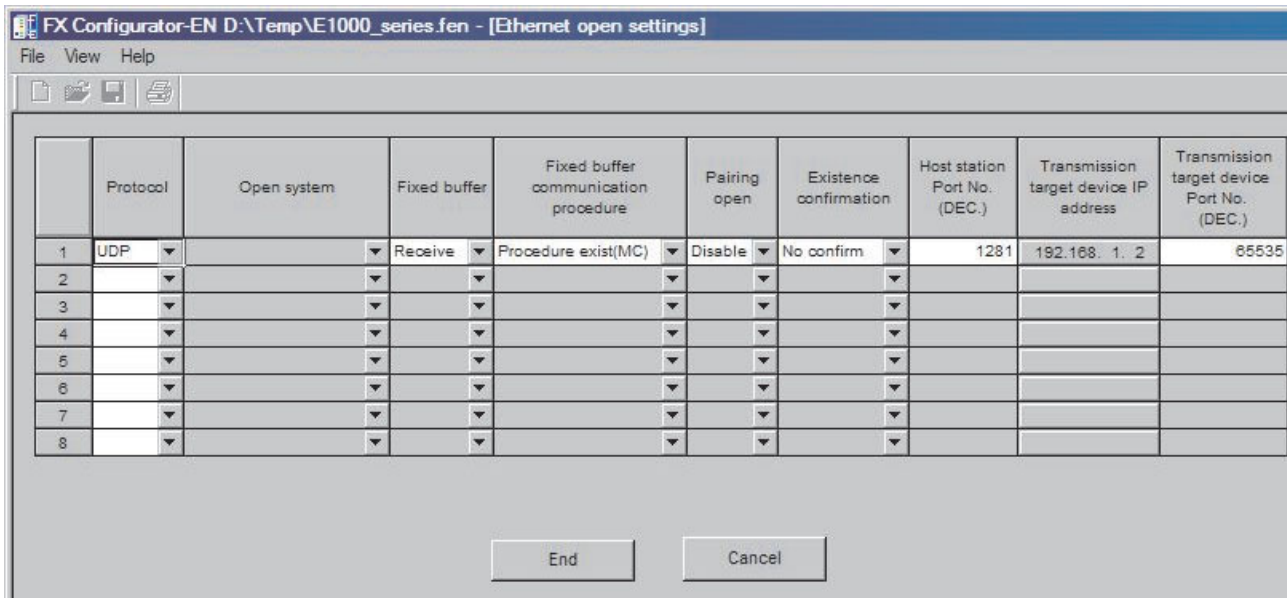
- ⑦ Open the properties of the TCP/IP connection.



- ⑧ Please enter a name for the connection and the IP address of the FX3U-ENET used for the connection.



When a HMI of the E1000 series is connected to the network, the following setting is required for the FX3U-ENET (refer to section 22.2, step ④):



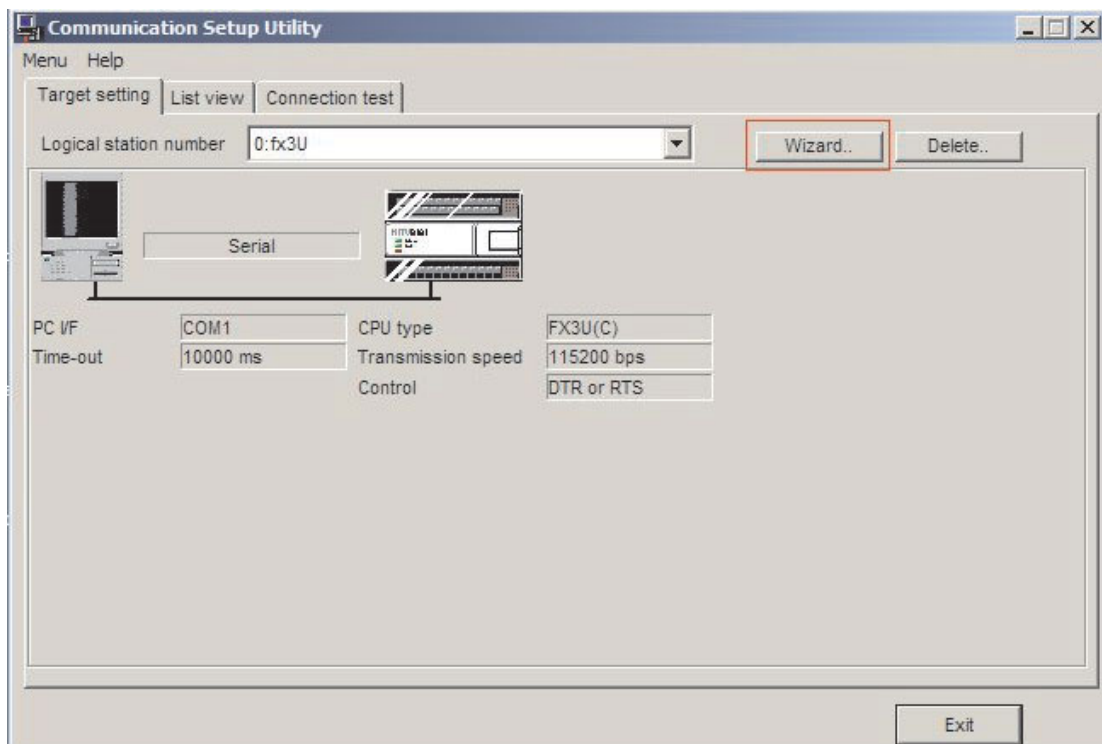
## 22.6 Communication via MX Component

MX Component is a tool designed to implement communication from a Personal Computer to the PLC without any knowledge of communication protocols and modules. MX Component is a powerful, user-friendly tools that make it very easy to connect your Mitsubishi PLC with the PC world.

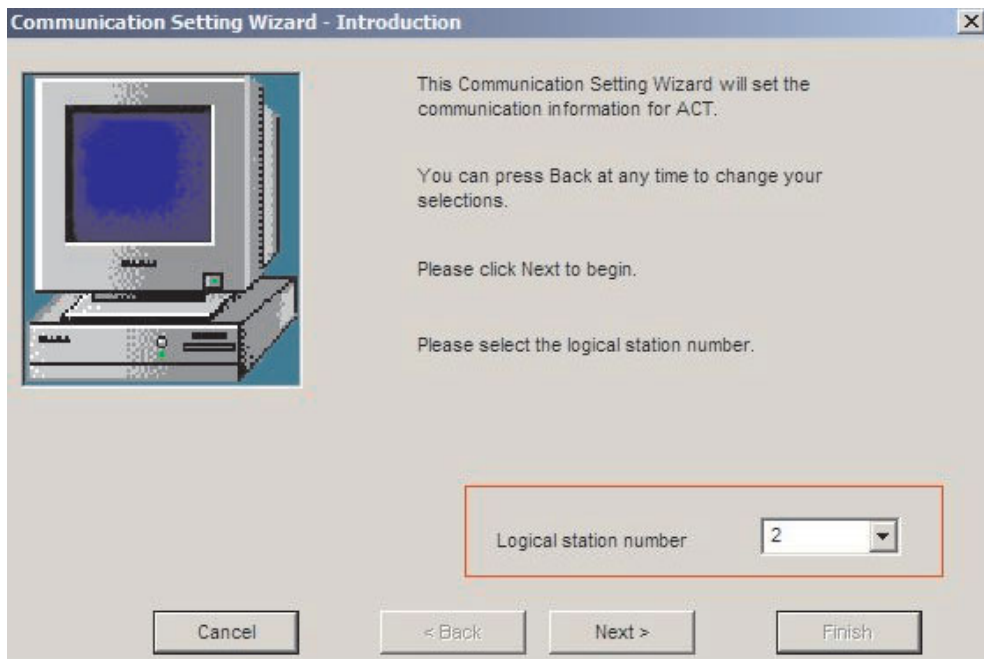
MX Component supports serial CPU port connection, serial computer links (RS232C, RS422) and networks (Ethernet, CC-Link, MELSEC).

The following figures show the easy way for creating of communication between a PC and a PLC via MX Component.

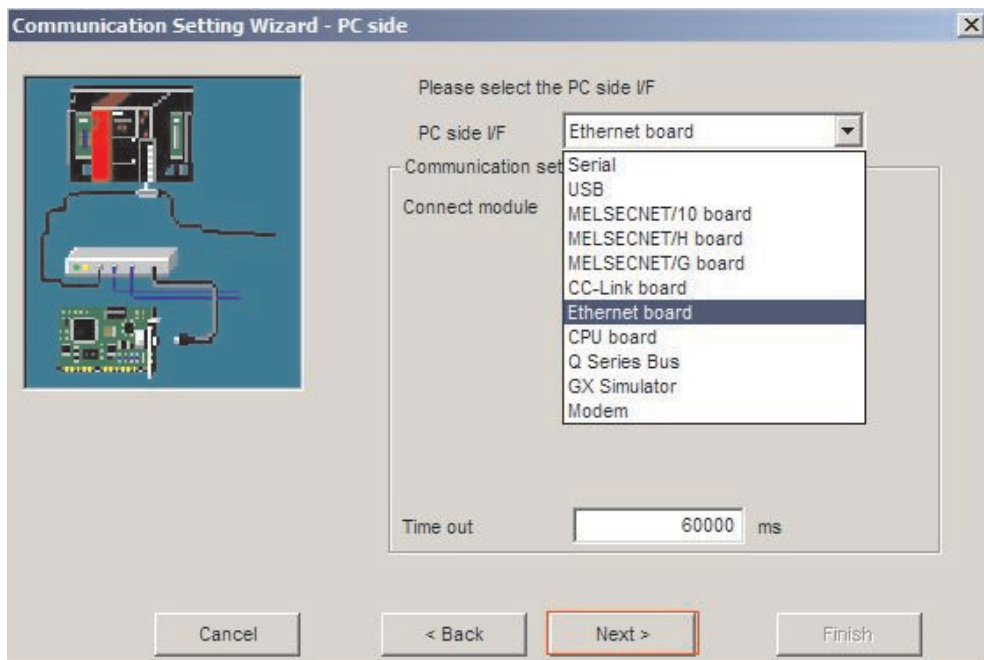
- ① Start the **Communication Setting Utility** and select the **Wizard**.



② First you must define the **Logical station number**.

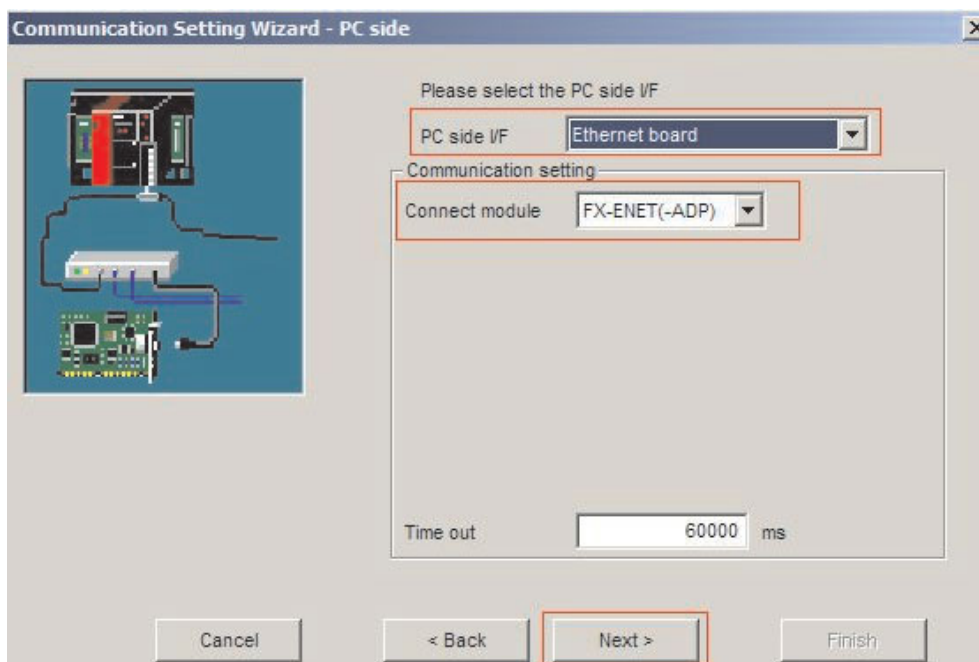


③ Next, configure the **Communication Settings** on the PC side. (Select the **Ethernet board**.)

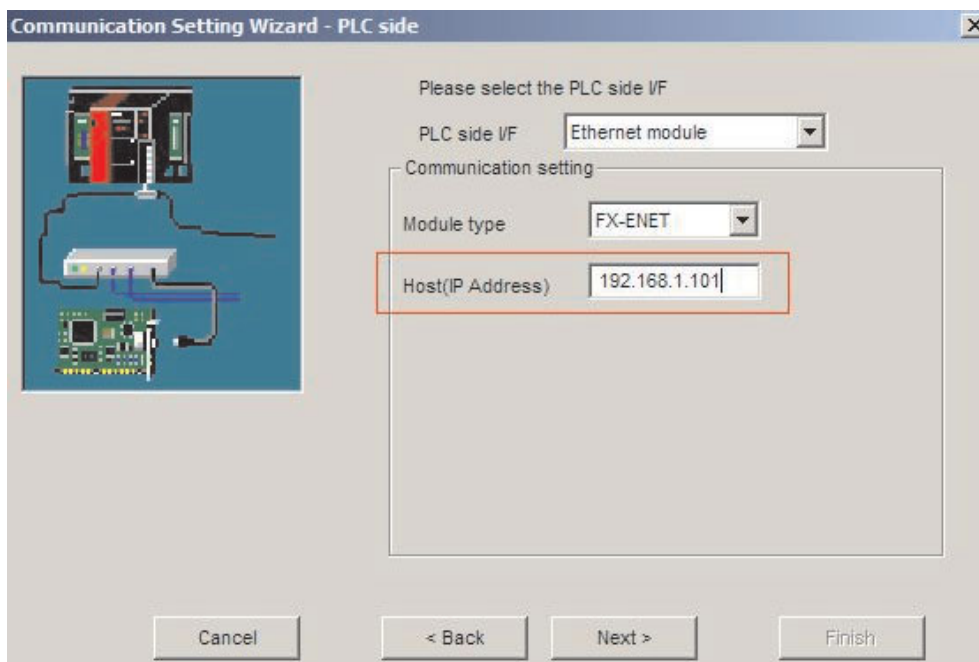




- ④ Select the FX-ENET(-ADP).

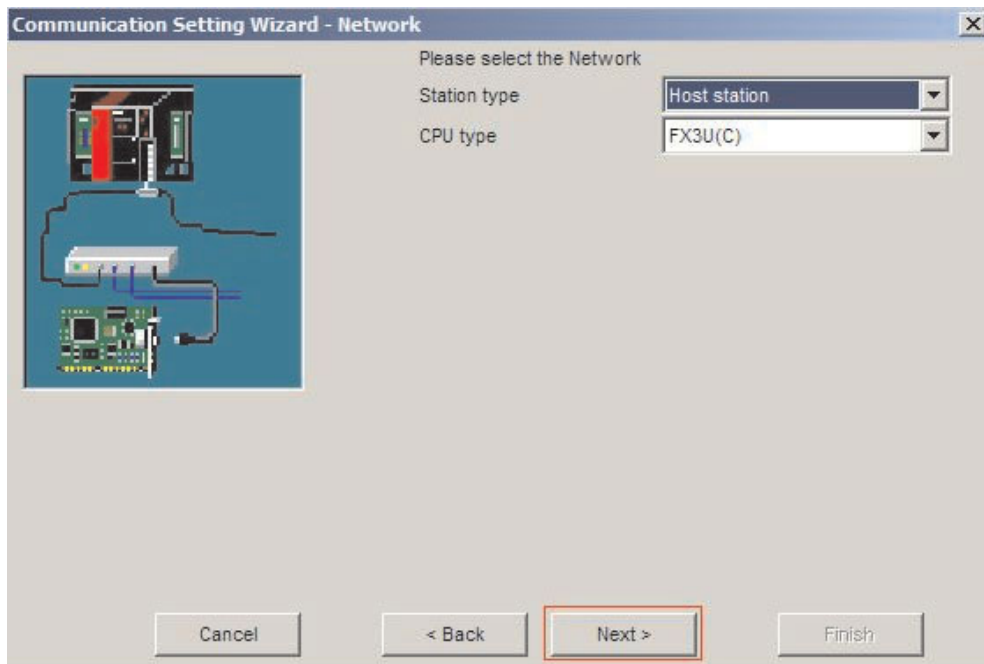


- ⑤ Enter the IP address of the Ethernet interface module.

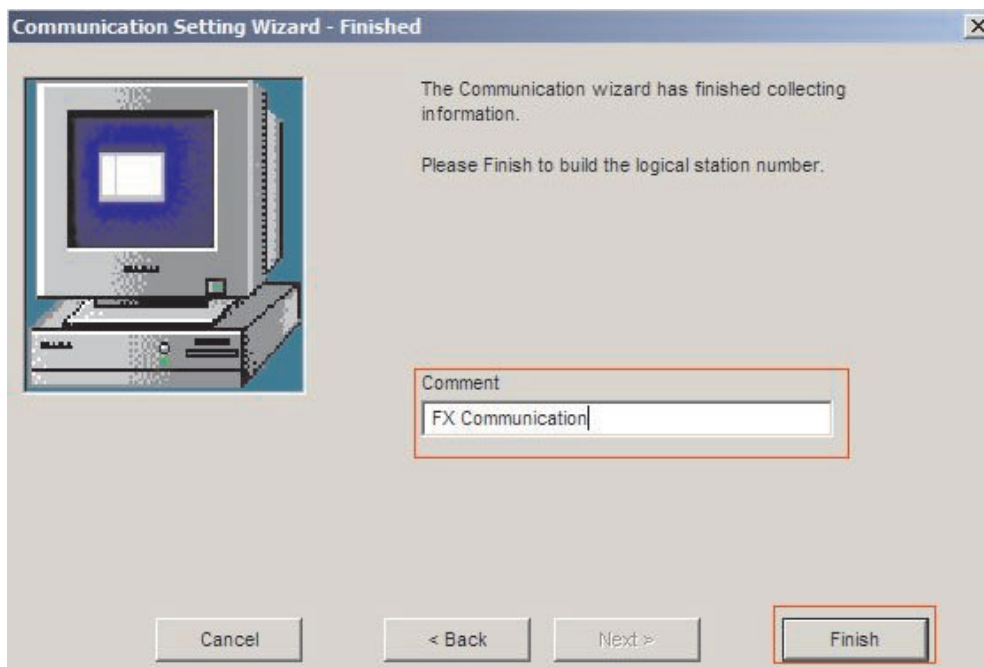




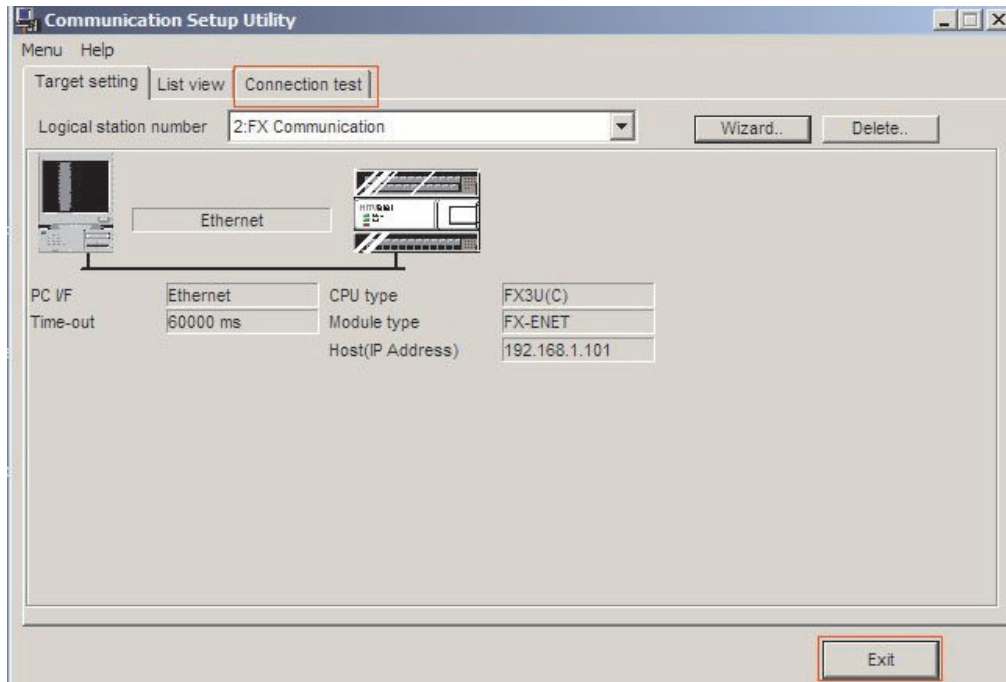
⑥ Select the correct CPU type.



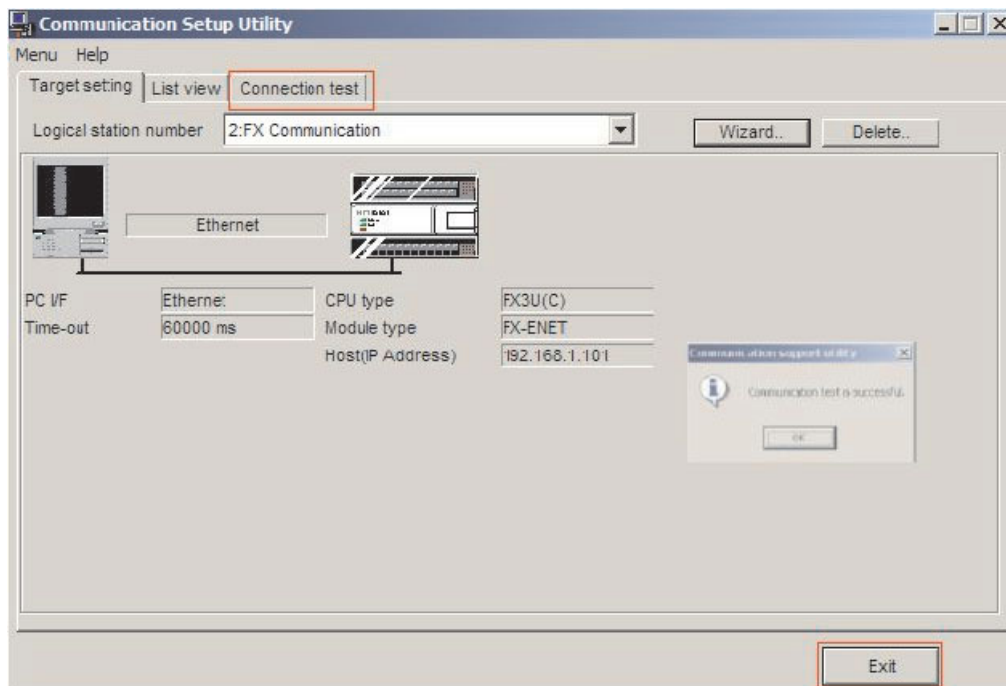
⑦ For the conclusion of the configuration define a name and press the **Finish** button.



Now the definition of communication is finished. Under the folder **Connection test** the connection can be examined.



The message **Communication test is successful** indicates that your configuration is correct.



After configuring the communication paths you can access all controller devices (read/write) with Microsoft programming languages like MS Visual Basic, MS C++ etc.

# A Appendix

## A.1 Special Relays

In addition to the relays that you can switch on and off with the PLC program there is also another class of relays known as special or diagnostic relays. These relays use the address range starting with M8000. Some contain information on system status and others can be used to influence program execution. Special relays cannot be used like other internal relays in a sequence program. However, some of them can be set ON or OFF in order to control the CPU. Represented here are some of the most commonly used devices.

Special relays can be divided in two groups:

- Special relays whose signal state can only be read by the program (for instance using a LD or LDI instruction).
- Special relays whose signal state can be read and written (set or reset) by the program.

The following tables feature a "Read" and a "Write" column. If the symbol "●" is shown in one of these columns, the corresponding action is possible. The symbol "—" means that the corresponding action is not allowed.

There are also special registers for word information in a FX CPU. They are described in the next section.

### A.1.1 PLC Status Diagnostic Information (M8000 to M8009)

Special Relay	Read	Write	CPU	Function
M8000	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	RUN monitor (NO contact)
M8001	●	—		RUN monitor (NC contact)
M8002	●	—		Initial pulse (NO contact)
M8003	●	—		Initial pulse (NC contact)
M8004	●	—		Error occurrence
M8005	●	—	FX2N FX2NC FX3G FX3U FX3UC	Battery voltage low (ON when battery voltage is below the value set in D8006)
M8006	●	—		Battery error latch (M8006 is set when battery voltage low is detected)
M8007	●	—	FX2N FX2NC	Momentary power failure
M8008	●	—	FX3U FX3UC	Power failure detected
M8009	●	—	FX2N FX2NC FX3G FX3U FX3UC	24V DC down (service power supply)

### A.1.2 Clock Devices and Real Time Clock (M8011 to M8019)

Special Relay	Read	Write	CPU	Function
M8010	—	—	—	Not used
M8011	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	10 ms clock pulse ON and OFF in 10 ms cycle (ON: 5 ms, OFF: 5 ms)
M8012	●	—		100 ms clock pulse ON and OFF in 100 ms cycle (ON: 50 ms, OFF: 50 ms)
M8013	●	—		1 s clock pulse ON and OFF in 1 s cycle (ON: 500 ms, OFF: 500 ms)
M8014	●	—		1 min clock pulse ON and OFF in 1 min cycle (ON: 30 s, OFF: 30 s)
M8015	●	●		Clock stop and preset (For real time clock)
M8016	●	—		Time read display is stopped (For real time clock) The contents of D8013 to D8019 is frozen, but the clock is still running.
M8017	●	●		±30 seconds correction (For real time clock)
M8018	●	—		Real time clock installation detection (Always ON) For an FX2NC a memory card with integrated RTC must be installed.
M8019	●	—		Real time clock (RTC) setting error

### A.1.3 PLC Operation Mode (M8030 to M8039)

Special relay	Read	Write	CPU	Function
M8030	●	—	FX2N FX2NC FX3G FX3U FX3UC	Battery LED OFF When M8030 set to ON, LED on PLC is not lit even if battery voltage low is detected.
M8031	●	●	FX1S FX1N FX2N FX2NC, FX3G FX3U FX3UC	Non-latch memory all clear
M8032	●	●		Latch memory all clear
M8033	●	●		Memory hold STOP When PLC is switched from RUN to STOP, image memory and data memory are retained.
M8034	●	●		All outputs disable All external output contacts of the PLC are turned OFF. The program however is still executed.
M8035	●	●		Forced RUN mode
M8036	●	●		Forced RUN signal
M8037	●	●		Forced STOP signal
M8038	—	●		FX1S FX1N FX2N (V2.0 or later) FX2NC FX3G FX3U FX3UC
M8039	●	●	FX1S FX1N FX2N FX2NC FX3G, FX3U FX3UC	Constant scan mode When M8039 is ON, PLC waits until scan time specified in D8039 and then executes cyclic operation.

### A.1.4 Error Detection (M8060 to M8069)

Special relay	Read	Write	CPU	Function	
M8060	●	—	FX2N FX2NC FX3G FX3U FX3UC	I/O configuration error	
M8061	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	PLC hardware error	
M8062	●	—	FX2N FX2NC	PLC/Programming device communication error	
			FX3G	Serial communication error [ch0]	
M8063 <sup>①</sup>	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	Serial communication error 1 [ch1]	
M8064	●	—		Parameter error	
M8065	●	—		Syntax error	
M8066	●	—		Ladder error	
M8067 <sup>②</sup>	●	—		Operation error	
M8068	—	●		Operation error latch	
M8069	—	●		FX2N FX2NC	I/O bus check <sup>③</sup>
				FX3G FX3U FX3UC	

① The operation varies according to a PLC: Cleared in an FX1S, FX1N, FX2N, FX1NC, or FX2NC when PLC switches from STOP to RUN. Cleared in an FX3G, FX3U, and FX3UC PLC when the power supply is switched on. Serial communication error 2 [ch2] in FX3G, FX3U, and FX3UC PLCs is detected by M8438.

② Cleared when PLC switches from STOP to RUN.

③ When M8069 is ON, I/O bus check is executed. If an error is detected, the error code 6130 is written to special register D8069 and the special relay M8061 is set.

### A.1.5 Extension Boards (Dedicated to FX1S and FX1N)

Special relay	Read	Write	CPU	Function
M8112	●	●	FX1S FX1N	Extension board FX1N-4EX-BD: Input BX0
				Extension board FX1N-2AD-BD: ch1 input mode change
				Extension board FX1N-1DA-BD: output mode change
M8113	●	●		Extension board FX1N-4EX-BD: Input BX1
				Extension board FX1N-2AD-BD: ch2 input mode change
M8114	●	●		Extension board FX1N-4EX-BD: Input BX2
M8115	●	●		Extension board FX1N-4EX-BD: Input BX3
M8116	●	●		Extension board FX1N-2EYT-BD: Output BY0
M8117	●	●	Extension board FX1N-2EYT-BD: Output BY1	

### A.1.6 Analog Special Adapter and Adapter Boards for FX3G

Special relay	Read	Write	CPU	Function
M8260 to M8269	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special relays for the 1st analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub> <sup>⑤</sup>	Special relays for the 1st analog adapter board <sup>②</sup>
M8270 to M8279	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special relays for the 2nd analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub> <sup>⑤</sup>	Special relays for the 2nd analog adapter board <sup>③</sup>
M8280 to M8289	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special relays for the 3rd analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub>	Special relays for the 1st analog special adapter
M8290 to M8299	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special relays for the 4th analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub>	Special relays for the 2nd analog special adapter (FX <sub>3G</sub> -40M□/□ and FX <sub>3G</sub> -60M□/□) only

- ① The unit number of the analog special adapter is counted from the base units side.
- ② Mounted to the expansion board connector of the base units FX<sub>3G</sub>-14M□/□ or FX<sub>3G</sub>-24M□/□ or to the left expansion board connector (BD1) of the base units FX<sub>3G</sub>-40□/□ or FX<sub>3G</sub>-60M□/□.
- ③ Mounted to the right expansion board connector (BD2) of the base units FX<sub>3G</sub>-40□/□ or FX<sub>3G</sub>-60M□/□.
- ④ Available in Version 2.00 or later
- ⑤ Available in Version 1.10 or later

## A.2 Special Registers

Just like the special relays (section A.1) starting at address M8000 the FX controllers also have special or diagnostic registers, whose addresses start at D8000. Often there is also a direct connection between the special relays and special registers. For example, special relay M8005 shows that the voltage of the PLC's battery is too low, and the corresponding voltage value is stored in special register D8005. The following tables shows a small selection of the available special registers as examples.

Special registers can be divided in two groups:

- Special registers whose value can only be read by the program
- Special relays whose value can be read and written by the program.

The following tables feature a "Read" and a "Write" column. If the symbol "●" is shown in a one of these columns, the corresponding action is possible. The symbol "—" means that the corresponding action is not allowed.

### A.2.1 PLC Status Diagnostic Information (D8000 to D8009)

Special Register	Read	Write	CPU	Function
D8000	●	●		Watchdog timer setting (in 1 ms steps). (Writes from system ROM at power ON) Value overwritten by program is valid after END or WDT instruction execution. The setting must be larger than the maximum scan time (stored in D8012). Default value is 200 ms.
D8001	●	—		PLC type and system version FX1S: 22V <sub>VV</sub> FX1N/FX3G: 26V <sub>VV</sub> FX2N/FX2NC/FX3U/FX3UC: 24V <sub>VV</sub> (e. g. FX1N Version 1.00 → 26100)
D8002	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	Memory capacity 0002 → 2k steps (FX1S only) 0004 → 4k steps (FX2N/FX2NC only) 0008 → 8k steps or more (not for FX1S) If 16K steps or more "K8" is written to D8002 and "16", "32" or "64" is written to D8102.
D8003	●	—		Memory typ: 00H → RAM (Memory cassette) 01H → EPROM (Memory cassette) 02H → EEPROM (Memory cassette or flash memory) 0AH → EEPROM (Memory cassette or flash memory, write-protected) 10H → Built-in memory in PLC
D8004	●	—		Error number (M) If D8004 contains e.g. the value 8060, special relay M8060 is set.
D8005	—	—	FX2N	Battery voltage (Example: "36" → 3.6 V)
D8006	—	—	FX2NC FX3G FX3U FX3UC	Low battery voltage detection level. Default settings: FX2N/FX2NC: 3.0 V ("30") FX3G/FX3U/FX3UC: 2.7 V ("27")
D8007	—	—	FX2N FX2NC FX3U FX3UC	Momentary power failure count Operation frequency of M8007 is stored. Cleared at power-off.
D8008	—	—	FX2N FX2NC FX3U FX3UC	Power failure detection Default settings: FX2N/FX3U: 10 ms (AC power supply) FX2NC/FX3UC: 5 ms (DC power supply)
D8009	—	—	FX2N FX2NC FX3G FX3U FX3UC	24V DC failed device Minimum input device number of extension units and extension power units in which 24V DC has failed.



### A.2.2 Scan Information and Real Time Clock (D8010 to D8019)

Special Register	Read	Write	CPU	Function
D8010	●	—	FX1S FX1N FX2N	Present scan time (in units of 0.1 ms)
D8011	●	—		Minimum value of scan time (in units of 0.1 ms)
D8012	●	—		Maximum value of scan time (in units of 0.1 ms)
D8013	●	●	FX1S FX1N FX2N FX2NC* FX3G FX3U FX3UC	Real time clock: Seconds (0 to 59)
D8014	●	●		Real time clock: Minutes (0 to 59)
D8015	●	●		Real time clock: Hours (0 to 23)
D8016	●	●		Real time clock: Date (Day, 1 to 31)
D8017	●	●		Real time clock: Date (Month, 1 to 12)
D8018	●	●		Real time clock: Date (Year, 0 to 99)
D8019	●	●		Real time clock: Day of the week (0 (Sunday) to 6 (Saturday))

\* For an FX2NC a memory card with integrated RTC must be installed.

### A.2.3 PLC Operation Mode (D8030 to D8039)

Special Register	Read	Write	CPU	Function
D8030	●	—	FX1S FX1N FX3G	Value of analog volume VR1 (Integer from 0 to 255)
D8031	●	—		Value of analog volume VR2 (Integer from 0 to 255)
D8032 – D8038	—	—	—	Not used
D8039	—	●	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	Constant scan duration Default: 0 ms (in 1 ms steps) (Writes from system ROM at power ON) Can be overwritten by program

### A.2.4 Error Codes (D8060 to D8069)

Special Register	Read	Write	CPU	Function
D8060	●	—	FX2N FX2NC FX3G FX3U FX3UC	If the unit or block corresponding to a programmed I/O number is not actually loaded, M8060 is set to ON and the first device number of the erroneous block is written to D8060 Meaning of the four digit code: 1st digit: 0 = Output, 1 = Input 2nd to 4th digit: First device number of the erroneous block
D8061	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	Error code for PLC hardware error
D8062	●	—	FX2N, FX2NC FX3G FX3U FX3UC	Error code for PLC/PP communication error
			FX3G	Error code for serial communication error [ch0]
D8063	●	—	FX1S FX1N FX2N FX2NC FX3G FX3U FX3UC	Error code for serial communication error 1 [ch1]
D8064	●	—		Error code for parameter error
D8065	●	—		Error code for syntax error
D8066	●	—		Error code for ladder error
D8067	●	—		Error code for operation error
D8068*	—	●		Operation error step number latched In case of 32K steps or more, the step number is stored in [D8313, D8312].
D8069*	●	—		Error step number of M8065 to M8067 In case of 32K steps or more, the step number is stored in [D8315, D8314].

\* Cleared when PLC switches from STOP to RUN.

### A.2.5 Extension Boards (Dedicated to FX1S and FX1N)

Special Register	Read	Write	CPU	Function
D8112	●	—	FX1S FX1N	Adapter FX1N-2AD-BD: Digital input value ch.1
D8113	●	—		Adapter FX1N-2AD-BD: Digital input value ch.2
D8114	●	●		Adapter FX1N-1DA-BD: Digital output value ch.1

## A.2.6 Analog Special Adapter and Adapter Boards for FX3G

Special Register	Read	Write	CPU	Function
D8260 to D8269	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special registers for the 1st analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub> <sup>⑤</sup>	Special registers for the 1st analog adapter board <sup>②</sup>
D8270 to D8279	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special registers for the 2nd analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub> <sup>⑤</sup>	Special registers for the 2nd analog adapter board <sup>③</sup>
D8280 to D8289	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special registers for the 3rd analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub>	Special registers for the 1st analog special adapter
D8290 to D8299	●	—	FX <sub>3U</sub> FX <sub>3UC</sub> <sup>④</sup>	Special registers for the 4th analog special adapter <sup>①</sup>
	●	—	FX <sub>3G</sub>	Special registers for the 2nd analog special adapter (FX <sub>3G</sub> -40M□/□ and FX <sub>3G</sub> -60M□/□) only

- ① The unit number of the analog special adapter is counted from the base units side.
- ② Mounted to the expansion board connector of the base units FX<sub>3G</sub>-14M□/□ or FX<sub>3G</sub>-24M□/□ or to the left expansion board connector (BD1) of the base units FX<sub>3G</sub>-40□/□ or FX<sub>3G</sub>-60M□/□.
- ③ Mounted to the right expansion board connector (BD2) of the base units FX<sub>3G</sub>-40□/□ or FX<sub>3G</sub>-60M□/□.
- ④ Available in Version 2.00 or later
- ⑤ Available in Version 1.10 or later

## A.3 Error Code List

When an error has been detected in the PLC, the error code is stored in special registers D8060 to D8067 and D8438. The following actions should be followed for diagnostic errors.

Represented here are some of the most common error codes.

### A.3.1 Error codes 6101 to 6409

Error	Special Register	Error Code	Description	Corrective Action
PLC hardware error	D8061	0000	No error	—
		6101	RAM error	
		6102	Operation circuit error	
		6103	I/O bus error (M8069 = ON)	Check for the correct connection of extension cables.
		6104	Powered extension unit 24 V failure (M8069 = ON)	
		6105	Watchdog timer error	Check user program. The scan time exceeds the value stored in D8000.
		6106	I/O table creation error (CPU error) When turning the power ON to the baseunit, a 24V power failure occurs in a powered extension unit. (The error occurs if the 24V power is not supplied for 10 seconds or more after main power turn ON.)	Check the power supply for the powered extension units.
6107	System configuration error	Check the number of the connected special function units/blocks. A few special function units/blocks are limited the number to connect.		
Communication error between PLC and programming device (FX2N and FX2NC only)	D8062	0000	No error	—
		6201	Parity, overrun or framing error	Check the cable connection between the programming device and the PLC. This error may occur when a cable is disconnected and reconnected during PLC monitoring.
		6202	Communication character error	
		6203	Communication data sum check error	
		6204	Data format error	
6205	Command error			
Serial communication error	D8063	0000	No error	<ul style="list-style-type: none"> <li>● Inverter communication, computer link and programming: Ensure the communication parameters are correctly set according to their applications.</li> <li>● N:N network, parallel link, etc.: Check programs according to applications.</li> <li>● Remote maintenance: Ensure modem power is ON and check the settings of the AT commands.</li> <li>● Wiring: Check the communication cables for correct wiring.</li> </ul>
		6301	Parity, overrun or framing error	
		6302	Communication character error	
		6303	Communication data sum check error	
		6304	Communication data format error	
		6305	Command error	
		6306	Communication time-out detected	
		6307	Modem initialization error	
		6308	N:N network parameter error	
		6312	Parallel link character error	
		6313	Parallel link sum error	
		6314	Parallel link format error	
6320	Inverter communication error			

Error	Special Register	Error Code	Description	Corrective Action
Parameter error	D8064	0000	No error	STOP the PLC, and correctly set the parameters.
		6401	Program sum check error	
		6402	Memory capacity setting error	
		6403	Latched device area setting error	
		6404	Comment area setting error	
		6405	File register area setting error	
		6406	Special unit (BFM) initial value setting, positioning instruction setting sum check error	
		6407	Special unit (BFM) initial value setting, positioning instruction setting error	
		6409	Other setting error	

### A.3.2 Error codes 6501 to 6510

Error	Special Register	Error Code	Description	Corrective Action
Syntax error	D8065	0000	Kein Fehler	During programming, each instruction is checked. If a syntax error is detected, modify the instruction correctly.
		6501	Incorrect combination of instruction, device symbol and device number	
		6502	No OUT T or OUT C before setting value	
		6503	– No OUT T or OUT C before setting value – Insufficient number of operands for an applied instruction	
		6504	– Same label number is used more than once. – Same interrupt input or high speed counter input is used more than once.	
		6505	Device number is out of allowable range.	
		6506	Invalid instruction	
		6507	Invalid label number [P]	
		6508	Invalid interrupt input [I]	
		6509	Other error	
		6510	MC nesting number error	

### A.3.3 Error codes 6610 to 6632

Error	Special Register	Error Code	Description	Corrective Action
Circuit error	D8066	0000	No error	—
		6610	LD, LDI is continuously used 9 times or more.	This error occurs when a combination of instructions is incorrect in the entire circuit block or when the relationship between a pair of instructions is incorrect. Modify the instructions in the program mode so that their mutual relationship becomes correct.
		6611	More ANB/ORB instructions than LD/LDI instructions	
		6612	Less ANB/ORB instructions than LD/LDI instructions	
		6613	MPS is continuously used 12 times or more.	
		6614	No MPS instruction	
		6615	No MPP instruction	
		6616	No coil between MPS, MRD and MPP, or incorrect combination	
		6617	Instruction below is not connected to bus line: STL, RET, MCR, P, I, DI, EI, FOR, NEXT, SRET, IRET, FEND or END	
		6618	STL, MC or MCR can be used only in main program, but it is used elsewhere (e.g. in interrupt routine or subroutine).	
		6619	Invalid instruction is used in FOR-NEXT loop: STL, RET, MC, MCR, I (interrupt pointer) or IRET.	
		6620	FOR-NEXT instruction nesting level exceeded	
		6621	Numbers of FOR and NEXT instructions do not match.	
		6622	No NEXT instruction	
		6623	No MC instruction	
		6624	No MCR instruction	
		6625	STL instruction is continuously used 9 times or more.	
		6626	Invalid instruction is programmed within STL-RET loop: MC, MCR, I (interrupt pointer), SRET or IRET.	
		6627	No RET instruction	
		6628	Invalid instruction is used in main program: I (interrupt pointer), SRET or IRET	
		6629	No P or I (interrupt pointer)	
6630	No SRET or IRET instruction			
6631	SRET programmed in invalid location			
6632	FEND programmed in invalid location			

### A.3.4 Error codes 6701 to 6710

Error	Special Register	Error Code	Description	Corrective Action
Operation error	D8067	0000	No error	—
		6701	<ul style="list-style-type: none"> <li>– No jump destination (pointer) for CJ or CALL instruction</li> <li>– Label is undefined or out of P0 to P4095 due to indexing</li> <li>– Label P63 is executed in CALL instruction; cannot be used in CALL instruction as P63 is for jumping to END instruction.</li> </ul>	This error occurs in the execution of operation. Review the program, or check the contents of the operands used in the applied instructions.*
		6702	CALL instruction nesting level is 6 or more	
		6703	Interrupt nesting level is 3 or more	
		6704	FOR-NEXT instruction nesting level is 6 or more.	
		6705	Operand of applied instruction is inapplicable device.	
		6706	Device number range or data value for operand of applied instruction exceeds limit.	
		6707	File register is accessed without parameter setting of file register.	
		6708	FROM/TO instruction error	This error occurs in the execution of operation. Review the program, or check the contents of the operands used in the applied instructions. Check whether the specified buffer memories exist in the equipment. Check whether the extension cables are correctly connected.
		6709	Other (e.g. improper branching)	This error occurs in the execution of operation. Review the program, or check the contents of the operands used in the applied instructions.*
6710	Mismatch among parameters	This error occurs when the same device is used within the source and destination in a shift instruction, etc.		

\* Even if the syntax or circuit design is correct, an operation error may still occur. For example: "T200Z" itself is not an error. But if Z had a value of 400, the timer T600 would be attempted to be accessed. This would cause an operation error since there is no T600 device available.

## A.4 Number of Occupied Input/Output Points and Current Consumption

The following tables show how many input/output points are occupied in a base unit by a certain unit, along with the power supply type and current consumption values needed for selecting a product.

The current consumption is determined differently in the following cases.

5V DC and internal 24V DC are supplied to the products through an extension cable, and the current consumption must be calculated

Subtract the current consumption at the internal 24V DC as follows.

- For the AC power type base unit, subtract the current consumption at the internal 24V DC from the 24V DC service power supply.
- For the DC power type base unit, subtract the current consumption at the internal 24V DC from the power supply for the internal 24V DC.
- Some special function modules need "external 24 V DC". Include this current in the calculation of current consumption when the current is supplied by the 24V DC service power supply. When the current is supplied by an external power supply, the current is not included in the calculation of current consumption.

### A.4.1 Interface Adapter Boards and Communication Adapter Boards

Type	Number of occupied I/O points	Current consumption [mA]		
		5 V DC	24 V DC (internal)	24 V DC (external)
FX1N-232-BD	—	20	—	—
FX2N-232-BD	—			
FX3G-232-BD	—			
FX3U-232-BD	—			
FX1N-422-BD	—	60*	—	—
FX2N-422-BD	—			
FX3G-422-BD	—			
FX3U-422-BD	—			
FX1N-485-BD	—	60	—	—
FX2N-485-BD	—			
FX3G-485-BD	—			
FX3U-485-BD	—			
FX3U-USB-BD	—	15	—	—
FX1N-CNV-BD	—	—	—	—
FX2N-CNV-BD				
FX3G-CNV-BD				
FX3U-CNV-BD				
FX3G-2AD-BD	—	—	—	—
FX3G-1DA-BD				
FX3G-8AV-BD				

\* When a programming tool or GOT is connected, add the current consumed by this unit (see next page)



**Programming Tool, Interface Converter, Display Module and GOT**

Type	Number of occupied I/O points	Current consumption [mA]		
		5 V DC	24 V DC (internal)	24 V DC (external)
FX-20P(-E)	—	150	—	—
FX-232AWC-H	—	120	—	—
FX-USB-AW	—	15	—	—
FX3U-7DM	—	20	—	—
FX10DM-E	—	220	—	—
F920GOT-BBD5-K-E	—	220	—	—

**A.4.2 Special Adapters**

Type	Number of occupied I/O points	Current consumption [mA]			
		5 V DC	24 V DC (internal)	24 V DC (external)	At start up
FX3U-4HSX-ADP	—	30	30	0	30*
FX3U-2HSY-ADP	—	30	60	0	120*
FX3U-4AD-ADP	—	15	0	40	—
FX3U-4DA-ADP	—	15	0	150	—
FX3U-4AD-PNK-ADP	—	15	0	50	—
FX3U-4AD-PT-ADP	—	15	0	50	—
FX3U-4AD-PTW-ADP	—	15	0	50	—
FX3U-4AD-TC-ADP	—	15	0	45	—
FX3U-3A-ADP	—	20	0	90	—
FX2NC-232ADP	—	100	0	0	—
FX3U-232ADP	—	30	0	0	—
FX3U-485ADP	—	20	0	0	—

\* The current consumption at start up must be considered when connected to a DC powered base unit.

**A.4.3 Extension Blocks**

Type	Number of occupied I/O points	Current consumption [mA]		
		5 V DC	24 V DC (internal)	24 V DC (external)
FX2N-8ER-ES/UL	16	—	125	0
FX2N-8EX-ES/UL	8	—	50	0
FX2N-16EX-ES/UL	16	—	100	0
FX2N-8EYR-ES/UL	8	—	75	0
FX2N-8EYT-ESS/UL	8	—	75	0
FX2N-16EYR-ES/UL	16	—	150	0
FX2N-16EYT-ESS/UL	16	—	150	0

### A.4.4 Special Function Modules

Type	Number of occupied I/O points	Current consumption [mA]			
		5 V DC	24 V DC (internal)	24 V DC (external)	At start up
FX3U-2HC	8	245	0	0	—
FX3U-4AD	8	110	0	90	—
FX3U-4DA	8	120	0	160	—
FX3U-4LC	8	160	0	50	—
FX3U-20SSC-H	8	100	0	220	—
FX2N-2AD	8	20	50 <sup>①</sup>	0	170
FX2N-2DA	8	30	85 <sup>①</sup>	0	190
FX2N-4AD	8	30	0	55	—
FX2N-4DA	8	30	0	200	—
FX2N-4AD-TC	8	30	0	50	—
FX2N-4AD-PT	8	30	0	50	—
FX2N-8AD	8	50	0	80	—
FX2N-5A	8	70	0	90	—
FX2N-2LC	8	70	0	55	—
FX2N-1HC	8	90	0	0	—
FX2N-1PG-E	8	55	0	40	—
FX2N-10PG	8	120	0	70 <sup>②</sup>	—
FX2N-232IF	8	40	0	80	—
FX2N-16CCL-M	8 <sup>③</sup>	0	0	150	—
FX2N-32CCL-M	8	130	0	50	—
FX2N-32ASI-M	8 <sup>④</sup>	150	0	70	—
FX0N-3A	8	30	90 <sup>①</sup>	0	165
FX2N-10GM	8	—	—	5	—
FX2N-20GM	8	—	—	10	—

- ① When analog special function blocks (FX0N-3A, FX2N-2AD and FX2N-2DA) are connected to an input/ output powered extension unit (FX2N-32E□ or FX2N-48E□), the following limitation must be taken into consideration. (When the blocks are connected to the base unit, this limitation is not applied.)

The total current consumption of the analog special function blocks (FX0N-3A, FX2N-2AD and FX2N-2DA) should be less than the following current values.

- When connected to FX2N-32E□: 190 mA or less
- When connected to FX2N-48E□: 300 mA or less.

- ② When the voltage of the external DC power supply is 5 V DC, the current is 100 mA.
- ③ A FX2N-16CCL-M cannot be used together with a FX2N-32ASI-M. The following number of points is added according to the products connected to the network: (Number of remote I/O stations) x 32 points.
- ④ A FX2N-32ASI-M cannot be used together with a FX2N-16CCL-M. Only one unit can be added to the whole system. The following number of points is added according to the products connected to the network: (Number of active slaves) x 8 points.

#### NOTE

When applying a DC power type base unit, calculate the current consumption at startup.

## A.5 PLC Components Glossary

The following table describes the meaning and functionality of the single components and parts of a Mitsubishi PLC.

Component	Description
Connection for expansion adapter boards	Optional expansion adapter boards can be connected to this interface. A variety of different adapters are available for all FX lines (except the FX2NC). These adapters extend the capabilities of the controllers with additional functions or communications interfaces. The adapter boards are plugged directly into the slot.
Connection for programming units	This connection can be used for connecting the FX-20P-E hand-held programming unit or an external PC or notebook with a programming software package (e.g. GX Developer).
EEPROM	Read/write memory in which the PLC program can be stored and read with the programming software. This solid-state memory retains its contents without power, even in the event of a power failure, and does not need a battery.
Memory cassette slot	Slot for optional memory cassettes. Inserting a memory cassette disables the controller's internal memory – the controller will then only execute the program stored in the cassette.
Extension bus	Both additional I/O expansion modules and special function modules that add additional capabilities to the PLC system can be connected here. See Chapter 6 for an overview of the available modules.
Analog potentiometers	The analog potentiometers are used for setting analog setpoint values. The setting can be polled by the PLC program and used for timers, pulse outputs and other functions.
Service power supply	The service power supply (not for FX2NC and FX3UC) provides a regulated 24V DC power supply source for the input signals and the sensors. The capacity of this power supply depends on the controller model (e.g. FX1S, FX1N and FX3G: 400mA; FX2N-16M□-□□ through FX2N-32M□-□□: 250 mA, FX2N-48M□-□□ through FX2N-64M□-□□: 460 mA)
Digital inputs	The digital inputs are used for inputting control signals from the connected switches, buttons or sensors. These inputs can read the values ON (power signal on) and OFF (no power signal).
Digital outputs	You can connect a variety of different actuators and other devices to these outputs, depending on the nature of your application and the output type.
LEDs for indicating the input status	These LEDs show which inputs are currently connected to a power signal, i.e. a defined voltage. When a signal is applied to an input the corresponding LED lights up, indicating that the state of the input is ON.
LEDs for indicating the output status	These LEDs show the current ON/OFF states of the digital outputs. These outputs can switch a variety of different voltages and currents depending on the model and output type.
LEDs for indicating the operating status	The LEDs RUN, POWER and ERROR show the current status of the controller. POWER shows that the power is switched on, RUN lights up when the PLC program is being executed and ERROR lights up when an error or malfunction is registered.
Memory battery	The battery protects the contents of the MELSEC PLC's volatile RAM memory in the event of a power failure (FX2N, FX2NC and FX3U only). It protects the latched ranges for timers, counters and relays. In addition to this it also provides power for the integrated real-time clock when the PLC's power supply is switched off.
RUN/STOP switch	MELSEC PLCs have two operating modes, RUN and STOP. The RUN/STOP switch allows you to switch between these two modes manually. In RUN mode the PLC executes the program stored in its memory. In STOP mode program execution is stopped and it is possible to program the controller.



# Index

## A

AS interface . . . . . 2 - 30  
 Adapter boards . . . . . 2 - 19  
 Alias . . . . . 10 - 11  
 Analog modules . . . . . 2 - 20

## B

Base unit  
   FX1N . . . . . 2 - 10  
   FX1S . . . . . 2 - 9  
   FX2N . . . . . 2 - 10  
   FX2NC . . . . . 2 - 11  
   FX3G . . . . . 2 - 11  
   FX3U . . . . . 2 - 12  
   FX3UC . . . . . 2 - 12  
   Overview . . . . . 2 - 6  
   Power supply . . . . . 2 - 13  
   S/S terminal . . . . . 2 - 14  
 Branch  
   delete . . . . . 9 - 3  
   insert . . . . . 8 - 5  
 Buffer memory . . . . . 20 - 1

## C

CANopen  
   Network module . . . . . 2 - 29  
 CC-Link  
   Network modules . . . . . 2 - 28  
 Change Display Color (Tools menu) . . . . . 4 - 7  
 Comments . . . . . 10 - 5  
 Communication adapters . . . . . 2 - 32  
 Connection Test . . . . . 12 - 2  
 Contact  
   change details . . . . . 8 - 4  
   delete . . . . . 9 - 2  
   insert . . . . . 8 - 3  
 Counter modules . . . . . 2 - 23  
 Counters . . . . . 18 - 1  
 Cross Reference List . . . . . 6 - 5

## D

Device test . . . . . 14 - 7  
 DeviceNet  
   Network modules . . . . . 2 - 29  
 Documentation . . . . . 10 - 1  
 Download . . . . . 12 - 1

## E

E-Designer . . . . . 22 - 14  
 EEPROM . . . . . A - 17  
 ETHERNET  
   Configuration . . . . . 22 - 1  
   FX Configurator-EN . . . . . 22 - 3  
   Network modules . . . . . 2 - 25  
 Edit menu  
   Delete Line . . . . . 9 - 3  
   Insert line . . . . . 8 - 7  
 Entry Data Monitor . . . . . 14 - 3  
 Error codes . . . . . A - 10  
 Extension  
   blocks . . . . . 2 - 18  
   boards . . . . . 2 - 17  
   units . . . . . 2 - 17

## F

FOR instruction . . . . . 21 - 1  
 FROM instruction . . . . . 20 - 4  
 FX Configurator-EN . . . . . 22 - 3  
 FX family  
   Current consumption . . . . . A - 14  
   Overview . . . . . 2 - 6  
   Power supply . . . . . 2 - 13  
   occupied I/O points . . . . . A - 14  
 FX0N-32NT-DP . . . . . 2 - 26  
 FX1N-8AV-BD . . . . . 2 - 33  
 FX1N-CNV-BD . . . . . 2 - 32  
 FX2N-10PG . . . . . 2 - 24  
 FX2N-16CCL-M . . . . . 2 - 28  
 FX2N-1HC . . . . . 2 - 23  
 FX2N-1PG-E . . . . . 2 - 24  
 FX2N-232IF . . . . . 2 - 31  
 FX2N-32ASI-M . . . . . 2 - 30  
 FX2N-32CAN . . . . . 2 - 29  
 FX2N-32CCL . . . . . 2 - 28  
 FX2N-32DP-IF . . . . . 2 - 27

FX2N-64DNET	2 - 29
FX2N-8AV-BD	2 - 33
FX2N-CNV-BD	2 - 32
FX2N-CNV-IF	2 - 32
FX2NC-1HC	2 - 23
FX2NC-CNV-IF	2 - 32
FX2NC-ENET-ADP	2 - 25
FX3G-8AV-BD	2 - 33
FX3G-CNV-ADP	2 - 32
FX3U-20SSC-H	2 - 24
FX3U-2HC	2 - 23
FX3U-2HSY-ADP	2 - 23
FX3U-32DP	2 - 26
FX3U-4HSX-ADP	2 - 23
FX3U-64CCL	2 - 28
FX3U-64DP-M	2 - 27
FX3U-CNV-BD	2 - 32
FX3U-ENET	2 - 25
Configuration	22 - 3
Find device	
for Cross Reference	6 - 5
for List of used devices	6 - 7
in Find/Replace menu	6 - 3
Find/Replace menu	
Cross Reference List	6 - 5
Find device	6 - 3
Find instruction	6 - 4
Find step no.	6 - 1
List of used devices	6 - 7

**G**

GT-Designer2	22 - 11
GX Developer	
customizing	3 - 3
installation	3 - 2
Glossary	A - 17
Grounding	2 - 13

**H**

HMI	
Ethernet communication	22 - 1
Overview	2 - 2

**I**

Inputs	
Assignment	2 - 41
wiring	2 - 14
Instruction	
List	5 - 1
in Ladder Diagram	4 - 5
search	6 - 4
Instruction List	5 - 1
Interface	
adapters	2 - 31
modules	2 - 31

**L**

Ladder Diagram	
Counters	18 - 1
Deletion	9 - 1
Documentation	10 - 1
Elements (instructions)	4 - 5
Line numbers	4 - 1
entering	4 - 9
modify	8 - 1

**M**

MELSEC	2 - 6
Memory battery	A - 17
Monitoring	14 - 1

**N**

NEXT instruction	21 - 1
Network modules	
AS interface	2 - 30
CANOpen	2 - 29
CC-Link	2 - 28
DeviceNet	2 - 29
ETHERNET	2 - 25
PROFIBUS	2 - 26
Notes	10 - 10

**O**

- Online menu
  - Clear PLC Memory . . . . . 12 - 4
  - Device test . . . . . 14 - 7
  - Monitor . . . . . 14 - 1
  - Read from PLC . . . . . 16 - 2
  - Transfer Setup . . . . . 12 - 1
  - Verify with PLC . . . . . 15 - 3
  - Write to PLC . . . . . 12 - 5
- Optical couplers . . . . . 2 - 9
- Outputs
  - Assignment . . . . . 2 - 41
  - wiring . . . . . 2 - 15

**P**

- PLC
  - History . . . . . 2 - 1
  - comparison with relay systems . . . . . 2 - 1
- PROFIBUS/DP
  - Network module . . . . . 2 - 26
- Positioning modules . . . . . 2 - 24
- Process image processing . . . . . 2 - 4
- Program
  - Conversion . . . . . 4 - 11
  - New . . . . . 4 - 3
  - Online programming . . . . . 19 - 1
  - see also Project
  - Verification . . . . . 15 - 1
  - documentation . . . . . 10 - 1
  - download to PLC . . . . . 12 - 1
  - monitor . . . . . 14 - 1
  - transfer to PLC . . . . . 12 - 5
  - upload from PLC . . . . . 16 - 1
- Programmable Logic Controller
  - see PLC
- Project
  - I/O Assignment . . . . . 2 - 41
  - New . . . . . 4 - 3
  - Save . . . . . 4 - 12
  - copy . . . . . 7 - 1
- Project Data List . . . . . 4 - 6
- Project menu
  - New Project . . . . . 4 - 3
  - Save . . . . . 4 - 12
  - Save as . . . . . 7 - 1

**R**

- RUN/STOP switch . . . . . A - 17
- Relay
  - comparison with PLC systems . . . . . 2 - 1

**S**

- SCADA . . . . . 2 - 2
- SFC . . . . . 17 - 1
- Service power supply . . . . . A - 17
- Sink
  - inputs . . . . . 2 - 14
  - outputs . . . . . 2 - 16
- Source
  - inputs . . . . . 2 - 14
  - outputs . . . . . 2 - 16
- Special Register
  - Diagnostic information . . . . . A - 7
  - Error codes . . . . . A - 8
  - PLC operation mode . . . . . A - 7
  - Real time clock . . . . . A - 7
  - described . . . . . A - 6
- Special Relays
  - Diagnostic information . . . . . A - 2
  - Error detection . . . . . A - 4
  - PLC operation mode . . . . . A - 3
  - Real time clock . . . . . A - 2
  - described . . . . . A - 1
- Special adapter
  - connection . . . . . 2 - 36
  - described . . . . . 2 - 19
- Special function modules
  - Buffer Memory . . . . . 20 - 1
  - address . . . . . 2 - 42
  - described . . . . . 2 - 19
- Statements . . . . . 10 - 9
- System Image . . . . . 12 - 3

**T**

- TO instruction . . . . . 20 - 5
- Temperature
  - acquisition modules . . . . . 2 - 21
  - control modules . . . . . 2 - 22
- Tools menu
  - Change Display Color . . . . . 4 - 7
- Trouble shooting
  - Error codes . . . . . A - 10
  - Special registers . . . . . A - 8
  - Special relays . . . . . A - 4

**U**

Upload . . . . . 16 - 1

**V**

View menu  
Alias . . . . . 10 - 12  
Alias Format Display . . . . . 10 - 13  
Comment format . . . . . 10 - 7  
Instruction List . . . . . 5 - 1  
Project Data List . . . . . 4 - 6





**HEADQUARTERS**

MITSUBISHI ELECTRIC EUROPE B.V. **EUROPE**  
 German Branch  
 Gothaer Straße 8  
**D-40880 Ratingen**  
 Phone: +49 (0)2102 / 486-0  
 Fax: +49 (0)2102 / 486-1120

MITSUBISHI ELECTRIC EUROPE B.V. **CZECH REPUBLIC**  
 Czech Branch  
 Avenir Business Park, Radlická 714/113a  
**CZ-158 00 Praha 5**  
 Phone: +420 - 251 551 470  
 Fax: +420 - 251-551-471

MITSUBISHI ELECTRIC EUROPE B.V. **FRANCE**  
 French Branch  
 25, Boulevard des Bouvets  
**F-92741 Nanterre Cedex**  
 Phone: +33 (0)1 / 55 68 55 68  
 Fax: +33 (0)1 / 55 68 57 57

MITSUBISHI ELECTRIC EUROPE B.V. **IRELAND**  
 Irish Branch  
 Westgate Business Park, Ballymount  
**IRL-Dublin 24**  
 Phone: +353 (0)1 4198800  
 Fax: +353 (0)1 4198890

MITSUBISHI ELECTRIC EUROPE B.V. **ITALY**  
 Italian Branch  
 Viale Colleoni 7  
**I-20041 Agrate Brianza (MB)**  
 Phone: +39 039 / 60 53 1  
 Fax: +39 039 / 60 53 312

MITSUBISHI ELECTRIC EUROPE B.V. **POLAND**  
 Poland Branch  
 Krakowska 50  
**PL-32-083 Balice**  
 Phone: +48 (0)12 / 630 47 00  
 Fax: +48 (0)12 / 630 47 01

MITSUBISHI ELECTRIC EUROPE B.V. **SPAIN**  
 Spanish Branch  
 Carretera de Rubí 76-80  
**E-08190 Sant Cugat del Vallés (Barcelona)**  
 Phone: 902 131121 // +34 935653131  
 Fax: +34 935891579

MITSUBISHI ELECTRIC EUROPE B.V. **UK**  
 UK Branch  
 Travellers Lane  
**UK-Hatfield, Herts. AL10 8XB**  
 Phone: +44 (0)1707 / 27 61 00  
 Fax: +44 (0)1707 / 27 86 95

MITSUBISHI ELECTRIC CORPORATION **JAPAN**  
 Office Tower "Z" 14 F  
 8-12, 1 chome, Harumi Chuo-Ku  
**Tokyo 104-6212**  
 Phone: +81 3 622 160 60  
 Fax: +81 3 622 160 75

MITSUBISHI ELECTRIC AUTOMATION, Inc. **USA**  
 500 Corporate Woods Parkway  
**Vernon Hills, IL 60061**  
 Phone: +1 847 478 21 00  
 Fax: +1 847 478 22 53

**EUROPEAN REPRESENTATIVES**

GEVA **AUSTRIA**  
 Wiener Straße 89  
**AT-2500 Baden**  
 Phone: +43 (0)2252 / 85 55 20  
 Fax: +43 (0)2252 / 488 60

TEHNIKON **BELARUS**  
 Oktyabrskaya 16/5, Off. 703-711  
**BY-220030 Minsk**  
 Phone: +375 (0)17 / 210 46 26  
 Fax: +375 (0)17 / 210 46 26

ESCO DRIVES & AUTOMATION **BELGIUM**  
 Culliganlaan 3  
**BE-1831 Diegem**  
 Phone: +32 (0)2 / 717 64 30  
 Fax: +32 (0)2 / 717 64 31

Koning & Hartman b.v. **BELGIUM**  
 Woluwelaan 31  
**BE-1800 Vilvoorde**  
 Phone: +32 (0)2 / 257 02 40  
 Fax: +32 (0)2 / 257 02 49

INEA BH d.o.o. **BOSNIA AND HERZEGOVINA**  
 Aleja Lipa 56  
**BA-71000 Sarajevo**  
 Phone: +387 (0)33 / 921 164  
 Fax: +387 (0)33 / 524 539

AKHNATON **BULGARIA**  
 4 Andrej Ljapchev Blvd. Pb 21  
**BG-1756 Sofia**  
 Phone: +359 (0)2 / 817 6004  
 Fax: +359 (0)2 / 97 44 06 1

INEA CR d.o.o. **CROATIA**  
 Losinjska 4 a  
**HR-10000 Zagreb**  
 Phone: +385 (0)1 / 36 940 -01 / -02 / -03  
 Fax: +385 (0)1 / 36 940 -03

AutoCont C.S. s.r.o. **CZECH REPUBLIC**  
 Technologická 374/6  
**CZ-708 00 Ostrava-Pustkovce**  
 Phone: +420 595 691 150  
 Fax: +420 595 691 199

B:ELECTRIC, s.r.o. **CZECH REPUBLIC**  
 Mladoboleslavská 812  
**CZ-197 00 Praha 19 - Kbely**  
 Phone: +420 286 850 848, +420 724 317 975  
 Fax: +420 286 850 850

Beijer Electronics A/S **DENMARK**  
 Lykkegårdsvej 17, 1.  
**DK-4000 Roskilde**  
 Phone: +45 (0)46 / 75 76 66  
 Fax: +45 (0)46 / 75 56 26

Beijer Electronics Eesti OÜ **ESTONIA**  
 Pärnu mnt.160i  
**EE-11317 Tallinn**  
 Phone: +372 (0)6 / 51 81 40  
 Fax: +372 (0)6 / 51 81 49

Beijer Electronics OY **FINLAND**  
 Jaakonkatu 2  
**FIN-01620 Vantaa**  
 Phone: +358 (0)207 / 463 500  
 Fax: +358 (0)207 / 463 501

UTEKO A.B.E.E. **GREECE**  
 5, Mavrogenous Str.  
**GR-18542 Piraeus**  
 Phone: +30 211 / 1206 900  
 Fax: +30 211 / 1206 999

MELTRADE Ltd. **HUNGARY**  
 Fertő utca 14.  
**HU-1107 Budapest**  
 Phone: +36 (0)1 / 431-9726  
 Fax: +36 (0)1 / 431-9727

Beijer Electronics SIA **LATVIA**  
 Vestienas iela 2  
**LV-1035 Riga**  
 Phone: +371 (0)784 / 2280  
 Fax: +371 (0)784 / 2281

Beijer Electronics UAB **LITHUANIA**  
 Savanorių Pr. 187  
**LT-02300 Vilnius**  
 Phone: +370 (0)5 / 232 3101  
 Fax: +370 (0)5 / 232 2980

**EUROPEAN REPRESENTATIVES**

ALFATRADE Ltd. **MALTA**  
 99, Paola Hill  
**Malta- Paola PLA 1702**  
 Phone: +356 (0)21 / 697 816  
 Fax: +356 (0)21 / 697 817

INTEHSIS srl **MOLDOVA**  
 bld. Traian 23/1  
**MD-2060 Kishinev**  
 Phone: +373 (0)22 / 66 4242  
 Fax: +373 (0)22 / 66 4280

HIFLEX AUTOM.TECHNIEK B.V. **NETHERLANDS**  
 Wolweverstraat 22  
**NL-2984 CD Ridderkerk**  
 Phone: +31 (0)180 - 46 60 04  
 Fax: +31 (0)180 - 44 23 55

Koning & Hartman b.v. **NETHERLANDS**  
 Haarlerbergweg 21-23  
**NL-1101 CH Amsterdam**  
 Phone: +31 (0)20 / 587 76 00  
 Fax: +31 (0)20 / 587 76 05

Beijer Electronics AS **NORWAY**  
 Postboks 487  
**NO-3002 Drammen**  
 Phone: +47 (0)32 / 24 30 00  
 Fax: +47 (0)32 / 84 85 77

Sirius Trading & Services srl **ROMANIA**  
 Aleea Lacul Morii Nr. 3  
**RO-060841 Bucuresti, Sector 6**  
 Phone: +40 (0)21 / 430 40 06  
 Fax: +40 (0)21 / 430 40 02

Craft Con. & Engineering d.o.o. **SERBIA**  
 Bulevar Svetog Cara Konstantina 80-86  
**SER-18106 Nis**  
 Phone: +381 (0)18 / 292-24-4/5  
 Fax: +381 (0)18 / 292-24-4/5

INEA SR d.o.o. **SERBIA**  
 Izletnicka 10  
**SER-113000 Smederevo**  
 Phone: +381 (0)26 / 617 163  
 Fax: +381 (0)26 / 617 163

AutoCont Control s.r.o. **SLOVAKIA**  
 Radlinského 47  
**SK-02601 Dolny Kubin**  
 Phone: +421 (0)43 / 5868210  
 Fax: +421 (0)43 / 5868210

CS MTrade Slovensko, s.r.o. **SLOVAKIA**  
 Vajanskeho 58  
**SK-92101 Piestany**  
 Phone: +421 (0)33 / 7742 760  
 Fax: +421 (0)33 / 7735 144

INEA d.o.o. **SLOVENIA**  
 Stegne 11  
**SI-1000 Ljubljana**  
 Phone: +386 (0)1 / 513 8100  
 Fax: +386 (0)1 / 513 8170

Beijer Electronics AB **SWEDEN**  
 Box 426  
**SE-20124 Malmö**  
 Phone: +46 (0)40 / 35 86 00  
 Fax: +46 (0)40 / 35 86 02

Omni Ray AG **SWITZERLAND**  
 Im Schörl 5  
**CH-8600 Dübendorf**  
 Phone: +41 (0)44 / 802 28 80  
 Fax: +41 (0)44 / 802 28 28

GTS **TURKEY**  
 Bayraktar Bulvari Nutuk Sok. No:5  
**TR-34775 Yukari Dudullu-Umraniye-ISTANBUL**  
 Phone: +90 (0)216 526 39 90  
 Fax: +90 (0)216 526 39 95

CSC Automation Ltd. **UKRAINE**  
 4-B, M. Raskovoyi St.  
**UA-02660 Kiev**  
 Phone: +380 (0)44 / 494 33 55  
 Fax: +380 (0)44 / 494-33-66

**EURASIAN REPRESENTATIVES**

Kazpromautomatiks Ltd. **KAZAKHSTAN**  
 Mustafina Str. 7/2  
**KAZ-470046 Karaganda**  
 Phone: +7 7212 / 50 11 50  
 Fax: +7 7212 / 50 11 50

**MIDDLE EAST REPRESENTATIVES**

TEXEL ELECTRONICS Ltd. **ISRAEL**  
 2 Ha'umanut, P.O.B. 6272  
**IL-42160 Netanya**  
 Phone: +972 (0)9 / 863 39 80  
 Fax: +972 (0)9 / 885 24 30

CEG INTERNATIONAL **LEBANON**  
 Cebaco Center/Block A Autostrade DORA  
**Lebanon - Beirut**  
 Phone: +961 (0)1 / 240 430  
 Fax: +961 (0)1 / 240 438

**AFRICAN REPRESENTATIVE**

CBI Ltd. **SOUTH AFRICA**  
 Private Bag 2016  
**ZA-1600 Isando**  
 Phone: +27 (0)11 / 977 0770  
 Fax: +27 (0)11 / 977 0761