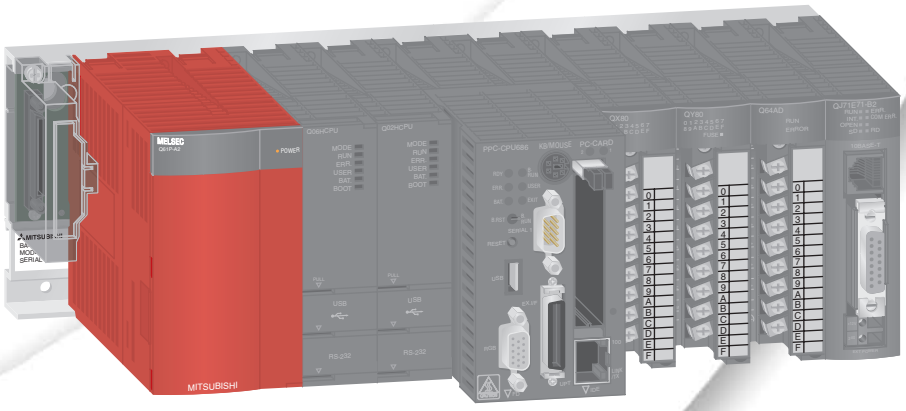
MITSUBISHI ELECTRIC
FACTORY AUTOMATION

MELSEC SYSTEM Q

Programmable Logic Controllers

Training Manual



GX Developer

About this Manual

The texts, illustrations and examples in this manual only explain theinstallation, 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.

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.: 170294

Version			Changes / Additions / Corrections
A	02/2006	pdp	First edition
B	08/2006	pdp-dk	Correction on page 2-5; Changed illustration on page 2-9; Addition of modules in section 2.9.2 on page 2-37 Adjustment of inputs and outputs for the program examples (e. g. X0 -> X10, Y10 -> Y20) to match with the configuration of the training rig

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 documentated 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 1,000V)
  - 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

1 Course Overwiew and Requirements

This course has been specially produced as an introduction to Mitsubishi's Q-Series range of modular PLC's 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 Q-Series PLC's, 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 Mitsubishi Q-Series Training Rig. 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
- 4 Analogue Inputs: Q64AD Located at Head Address 30H
- 4 Analogue Outputs: Q64DA Located at Head Address 40H.

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 constantmaintenance

2.1.4 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.

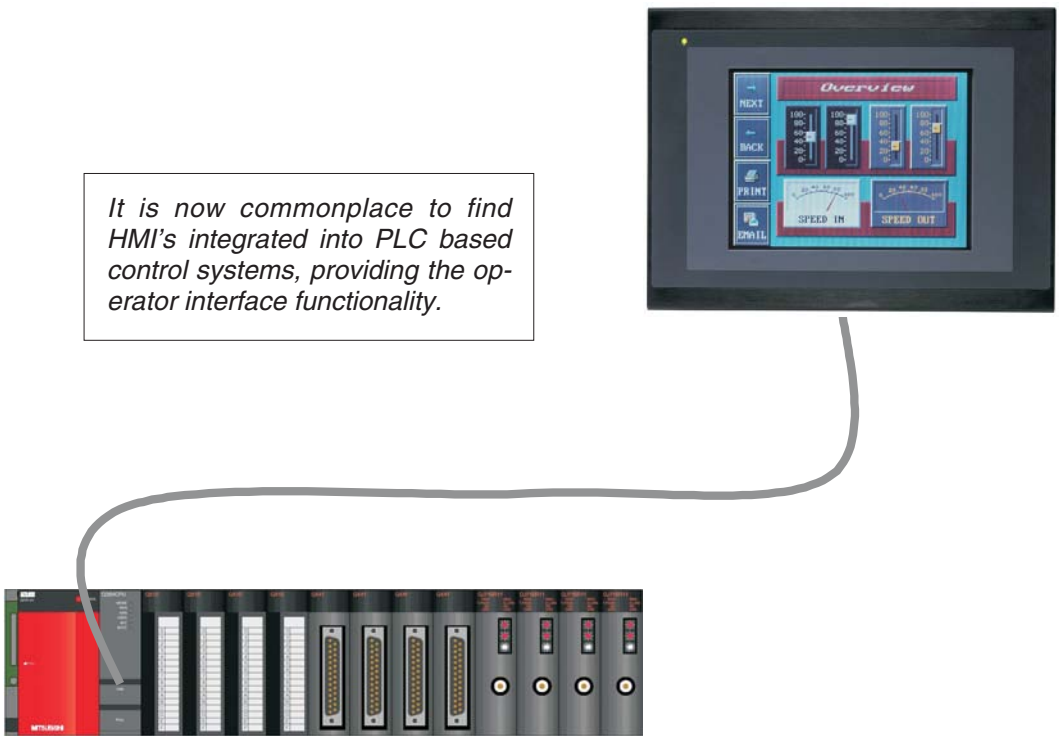
2.1.5 SCADA and HMI

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.

It is now commonplace to find HMI's integrated into PLC based control systems, providing the operator interface functionality.



2.2 Hardware Configuration

This section deals with the design concepts and configuration of a Q-Series system.

2.2.1 Specifying a PLC System

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

External devices, Inputs and Outputs

- Input/Output Requirements
- System Signal Voltage: 24V DC, 110V/240VAC
- If 24V DC inputs then: NPN (Sink) or PNP (Source) devices
- Output Configuration: Transistor (Sink/Source), Triac, Relay or Volt Free Relay contact

Power supply requirements

- Supply voltage: 24VDC, 110V/240VAC

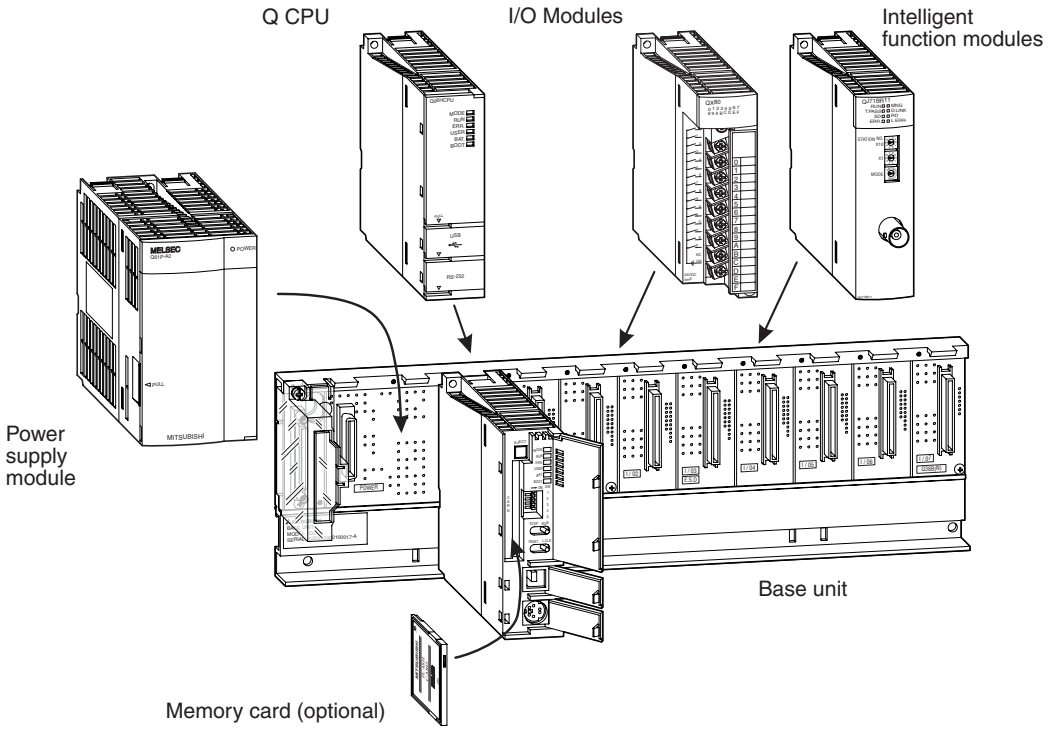
Intelligent Modules

- Number of modules in system
- External power supply requirements

2.3 Qn Series PLC Overview

The following information represents an overview of the configuration and format of the Qn PLC hardware. Data is also provided on the internal and operational specification of the Qn PLC systems.

2.3.1 System Configuration



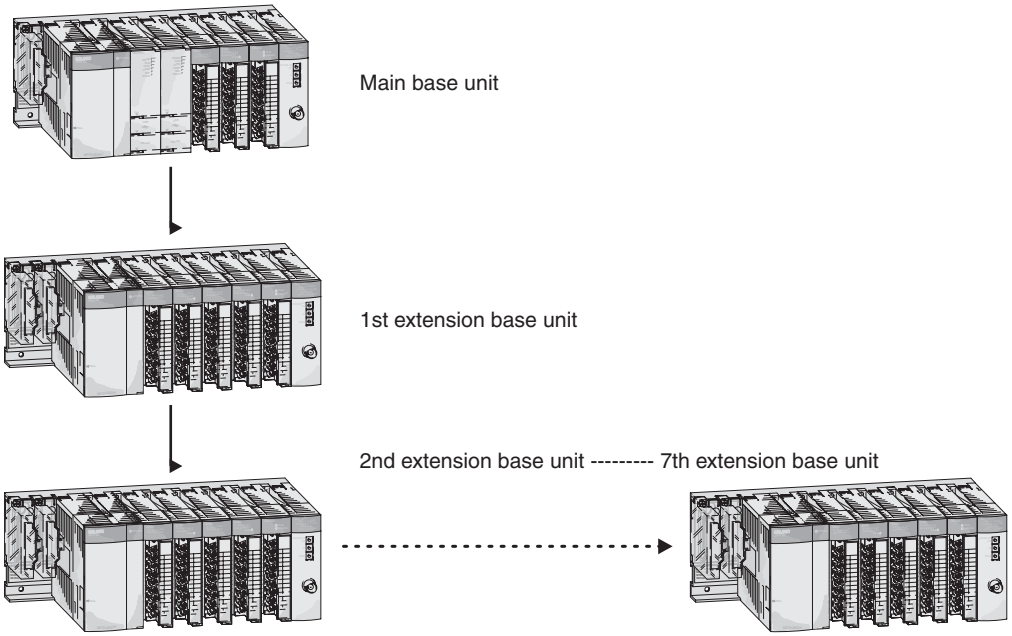
The CPU and modules are held in a base unit which has an internal bus connection for communication between the individual modules and the CPUs. The power supply module which supplies the voltage for the entire system is also installed on this base unit.

The base units are available in 4 different versions with 3 to 12 module slots. Each base unit can be supplemented by means of an extension unit providing additional slots.

If you wish to keep open the option of subsequent extension of your PLC or if you have free slots on your base unit, you can insert dummy modules here. They serve to protect the free slots from soiling or from mechanical effects but can also be used for reserving I/O points.

For cabling larger systems and machines - e.g. in a modular design - the use of remote I/O modules offers additional communications facilities.

Main Base and Extension Base Configuration



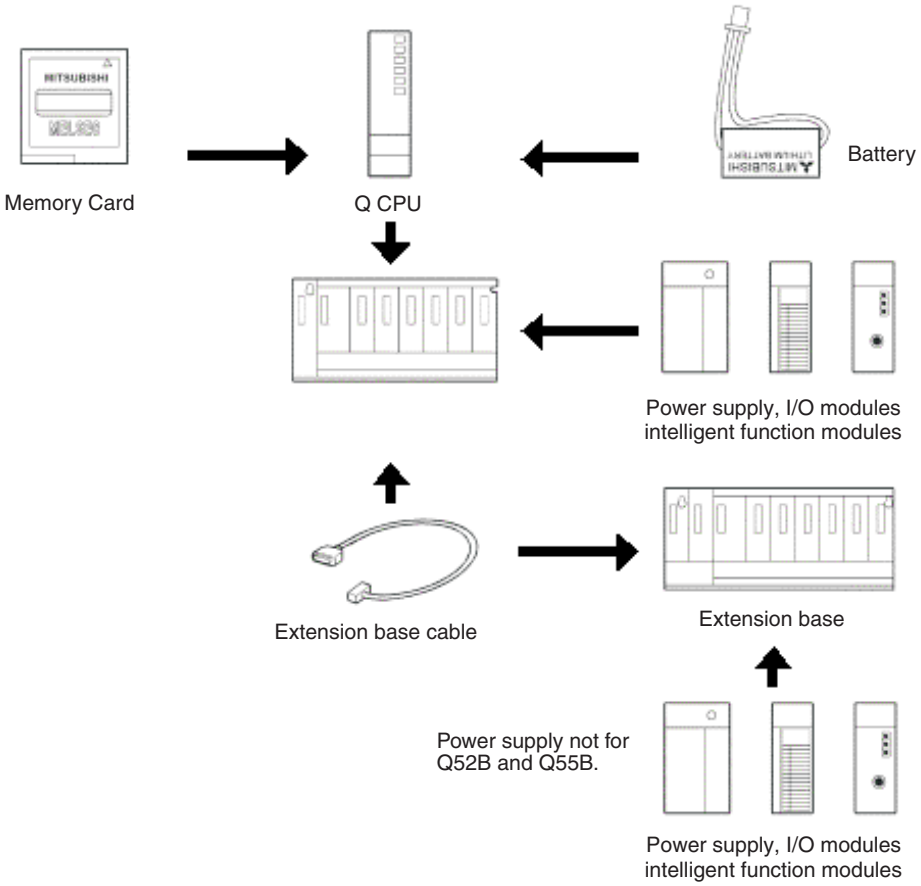
The base unit and extension units are simply connected to one another by extension cables. These connecting cables also supply the extension units with the operating voltage of 5 V DC.

Up to seven extension units with up to 64 modules can be connected to base units or extension base units. The maximum length of the extensions cables is 13.2 m.

When choosing the power supply module, the total power consumption of the I/O modules, of the special function modules and of the peripherals must be taken into account. If necessary, an extension unit with a further power supply module should be used.

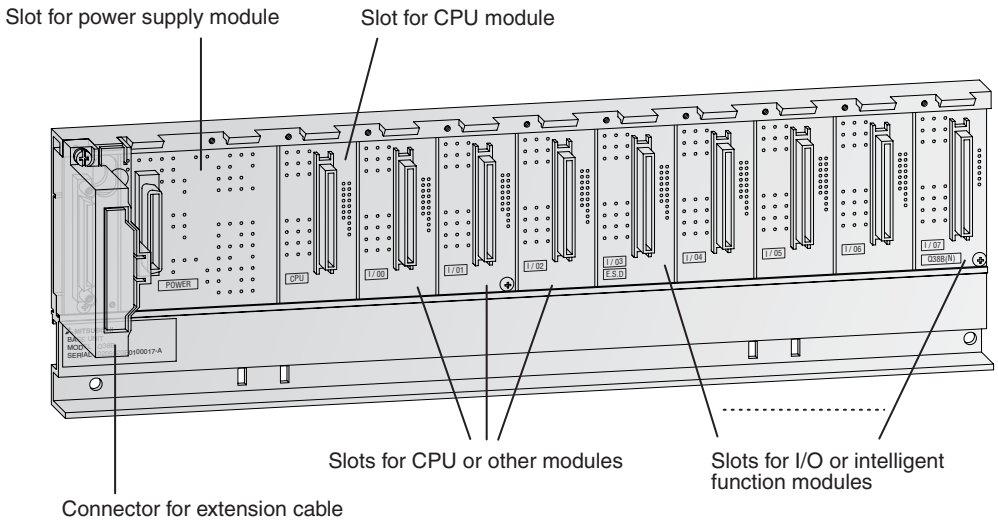
Number of extension base units

- Up to 4 extension base units can be connected to a main base unit in which a Q00CPU or Q01CPU is installed. The maximum number of loadable modules is 24.
- A system using Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU or Q25HCPU can be extented by up to 7 extension base units.The total number of I/O and intelligent function modules in all base units is 64.



2.3.2 Base units

The main base units provide slots for a power supply module, up to four CPU modules,and I/O and intelligent function modules. I/O and intelligent function modules can also be mounted on the extension base units.The base units can be installed directly using screws or on a DIN rail using adapters.



The following table shows the available base units.

Item	Main base units				
	Q33B	Q35B	Q38B	Q38RB	Q312B
Loadable power supply modules	1	1	1	2*	1
Number of slots for I/O or intelligent funktion modules	3	5	8	8	12

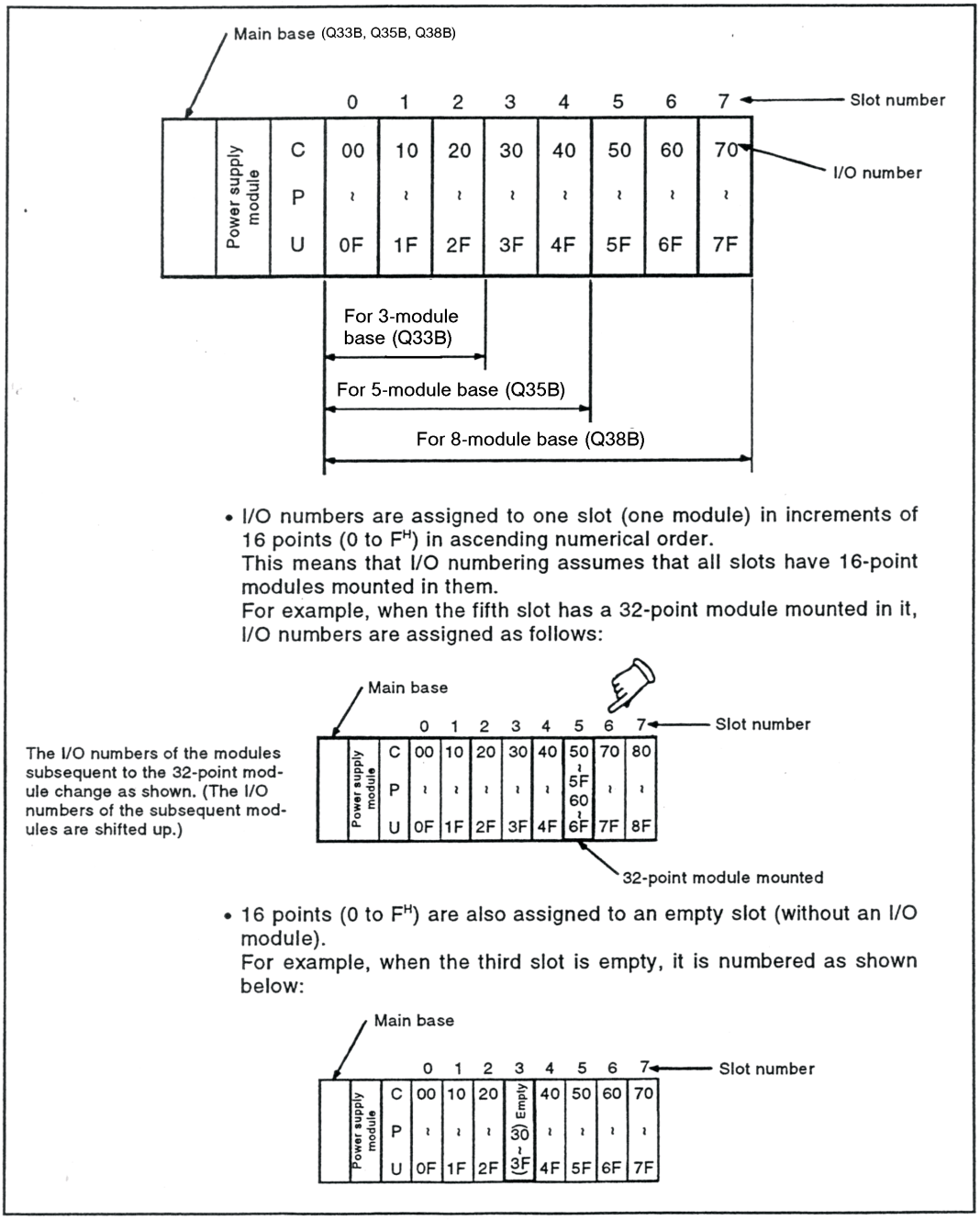
* In this main unit redundant power supply modules can be used.

Item	Extension base units						
	Q52B	Q55B	Q63B	Q65B	Q68B	Q68RB	Q612B
Loadable power supply modules	—	—	1	1	1	2*	1
Number of slots for I/O or intelligent funktion modules	2	5	3	5	8	8	12

* In this extension base unit redundant power supply modules can be used.

2.3.3 Main base I/O numbering

I/O numbers are assigned to the I/O modules mounted on the main base unit as described below. This also applies to special function modules.



- I/O numbers are assigned to one slot (one module) in increments of 16 points (0 to F^H) in ascending numerical order.
  This means that I/O numbering assumes that all slots have 16-point modules mounted in them.
  For example, when the fifth slot has a 32-point module mounted in it, I/O numbers are assigned as follows:

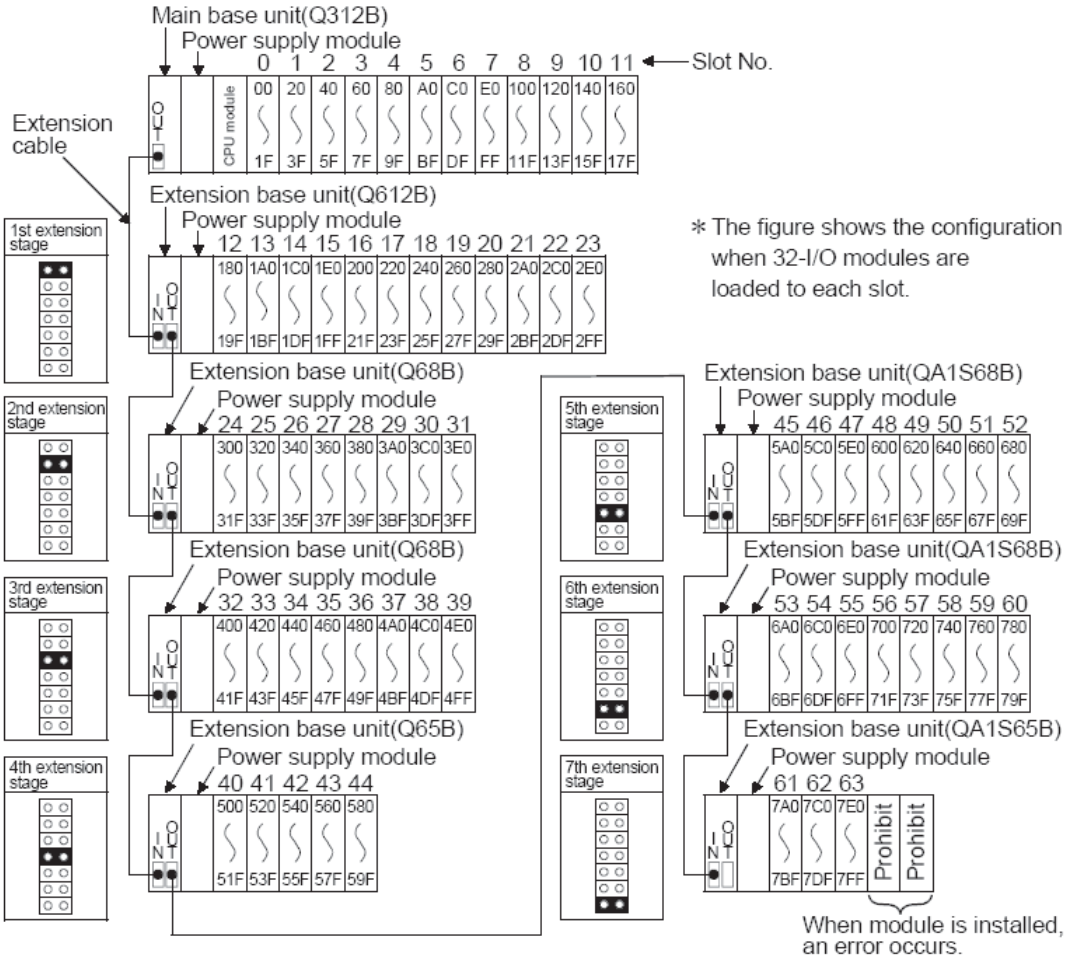
The I/O numbers of the modules subsequent to the 32-point module change as shown. (The I/O numbers of the subsequent modules are shifted up.)

- 16 points (0 to F^H) are also assigned to an empty slot (without an I/O module).
  For example, when the third slot is empty, it is numbered as shown below:

2.3.4 Extension base I/O numbering

The slots of extension bases are also numbered in increments of 16 points in numerical order.

- The first slot of any extension base is numbered following the last number of the main base or preceding extension base.
- An extension base cannot be connected to a 2 slot main base.
- Connect extension bases when more slots are needed in addition to the main base unit. Their I/O numbers are assigned as follows:



2.4 Extensions Base Cables

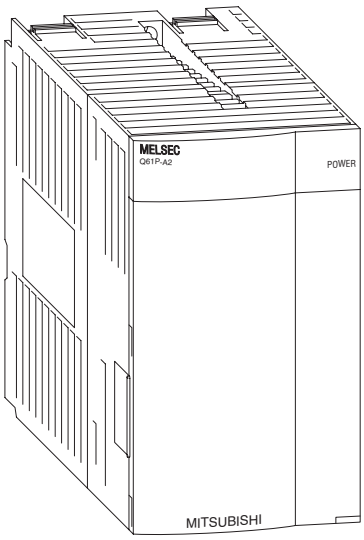
The extension base cables are used for connections between the base units.

Type	QC05B	QC06B	QC12B	QC30B	QC50B	QC100B
Cable length	0.45 m	0.50 m	1.2 m	3.0 m	5.0 m	10,0 m

The overall distance of all extension cables must not exceed 13.2 m.

For connection of the base units without an own power supply (Q52B, Q55B) the cable QC05B is recommended.

2.5 Power Supply Modules



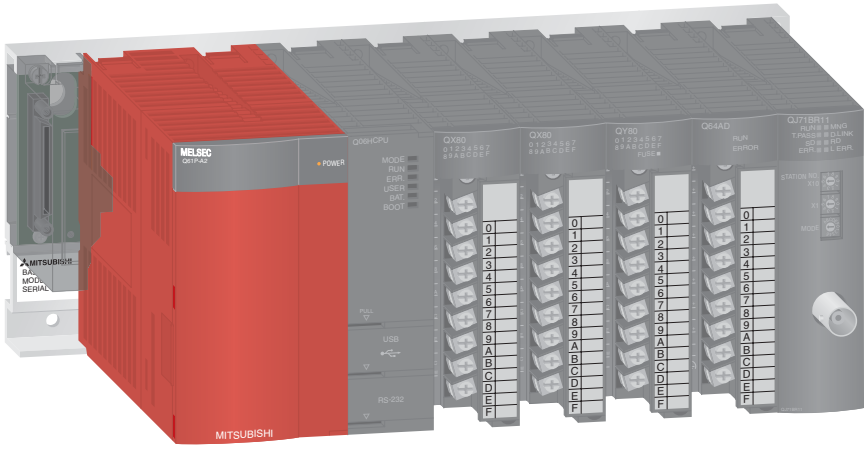
The power supply modules supply 5 V DC to each module on the base unit.
Power supply modules with input voltages of 24 V DC or 240 V AC are available.

Item	Q63P	Q61P-A1	Q61P-A2	Q62P	Q64P
Input voltage	24 V DC	100 – 120 V AC	200 – 220 V AC	100 – 240 V AC	100 – 120 V AC 200 – 240 V AC
Power consumption	45 W	105 VA	105 VA	105 VA	105 VA
Output voltage	5 V DC	5 V DC	5 V DC	5 V DC, 3 A 24 V DC, 0.6 A	5 V DC
Output current	6 A	6 A	6 A		8.5 A

2.5.1 Selection of an appropriate Power Supply

The total current consumption of the installed modules must be smaller than the rated output current of the power supply module. Reduce the number of modules on the base unit, if the current consumption is too high.

Example calculation of the total current consumption

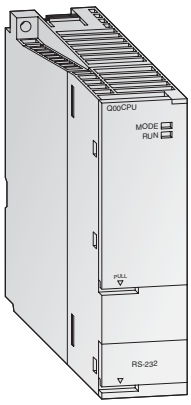


Module	Description	Current consumption
Q06HCPU	CPU module	0.64 A
QX80	Digital input module	0.16 A
QX80	Digital input module	0.16 A
QY80	Digital output module	0.008 A
Q64AD	A/D-converter module	0.63 A
QJ71BR11	MELSECNET/H module	0.75 A
Total current consumption		2.42 A

The total current consumption is 2.42 A. The installed power supply module is able to deliver a current of 6 A. This configuration will work without problems

2.6 CPU Modules

Basic PLC CPUs



The CPU modules of the MELSEC System Q are available as single and multi processor CPUs through which they achieve a wide application range. The performance of the controller here grows with the application by simply replacing the CPU (except Q00J).

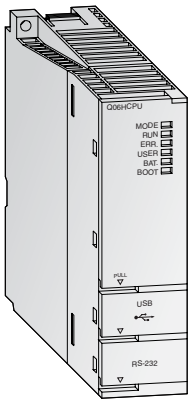
While Q00CPU and Q01CPU are classical separate CPUs, the Q00JCPU forms an inseparable unit consisting of CPU, power supply and base unit and thus enables a low-priced entry into the modular PLC technology.

The standard CPUs were developed especially for applications where a compact system configuration easily to be realized is to the fore.

Special features:

- Every CPU is equipped with an RS232C interface for easy programming and monitoring from a personal computer or operating panel.
- Integrated Flash ROMs for memory operation without additional memory cards.
- Processing the inputs and outputs with refresh mode

High performance CPUs



With the high-performance CPUs a high processing speed and expandability are to the fore. They provide a great variety of functions and an even optimized programming and debugging environment to ensure a flexible response to all systems.

The two process CPU models Q12PHCPU and Q25PHCPU have extended control functions with two degrees of freedom, PID cascading and autotuning. These processors also feature a set of 52 process instructions and support an unlimited number of PID loops

Special features:

- Every multi processor H-CPU is equipped with an USB interface for easy programming and monitoring from a personal computer.
- Processing the inputs and outputs with refresh mode
- Floating point arithmetic according to IEEE 754
- Special statements for processing PID control loops
- Mathematical functions, such as angle/exponential functions and logarithm
- Hot-swap module replacement in RUN mode (with process CPUs)
- Multi processor mode is possible with up to 4 CPU modules.

2.6.1 CPU Specification

Feature		Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Control method		Repeated operation using stored program						
I/O control method		Refresh mode						
Programming language		IEC ladder, lögic symbolic language, list, structured text (ST), SFC						
Processing speed	LD	160 ns	100 ns	79 ns	34 ns			
	MOV	560 ns	350 ns	237 ns	102 ns			
	Mixed instructions per µs	2.0	2.7	4.4	10.3			
	Floating point addition	27 µs*		1.8 µs	0.78 µs			
Number of instrucions (without instructions for intelligent function modules)		249		363				
Processing of floating point numbers		Supported*		Supported				
Processing of character strings		$MOV is supported only		Supported				
PID control		Supported*		Supported				
Special functions (such as trigonometrical functions, extraction of root or logarithm)		Supported*		Supported				

* For Q00/Q01CPU function version B (First 5 digits of serial number are "04122" or later)

Feature		Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Constant scan (program start at given time intervals)		1 to 2000 ms (can be specified in 1 ms increments)		0.5 to 2000 ms (can be specified in 0.5 ms increments)				
Program capacity (number of steps)		8 k	14 k	28 k		60 k	124 k	252 k
Memory capacity	Built-in program memory (drive 0)	94 kbytes		112 kbytes		240 kbytes	496 kbytes	1 MB
	RAM memory card (drive 1)	—		Capacity of loaded memory card (maximum 1 MB)				
	ROM memory card (drive 2)	—		Capacity of loaded memory card (maximum 4 MB for flash cards and 32 MB for ATA cards)				
	Built-in RAM (drive 3)	128 kbytes *		64 kbytes			256 kbytes	
	Built-in ROM (drive 4)	94 kbytes		112 kbytes		240 kbytes	496 kbytes	1 MB
	Common memory for multi processor mode	1 kbytes **		8 kbytes				
I/O points	Total (including remote I/O)	2048		8192				
	Local I/O	1024		4096				

* 64 k bytes for function version A

** For Q00/Q01CPU function version B (First 5 digits of serial number are "04122" or later)

Number of Devices

Device (Device symbol)	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Internal relay (M)	8192		8192				
Latch relay (L)	2048		8192				
Link relay (B)	2048		8195				
Timer (T)	512		2048				
Retentive Timer (ST)	0		0				
Counter (C)	512		1024				
Data register (D)	11136		12288				
Link register (W)	2048		8196				
Annunciator (F)	1024		2048				
Edge relay (V)	1024		2048				

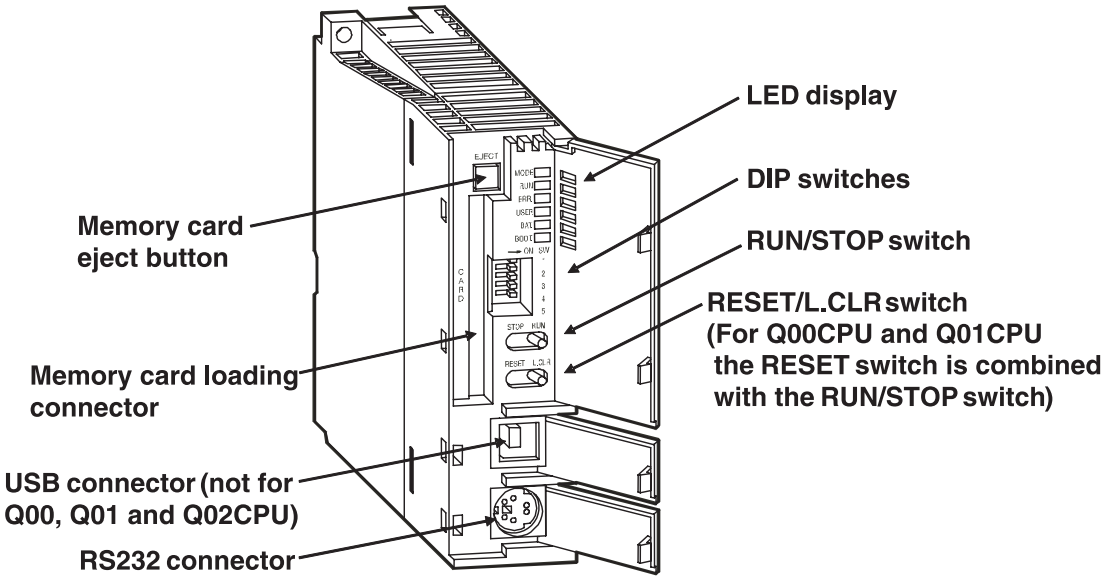
The table above indicates the default number of points. These can be changed in the parameter configuration.

Device (Device symbol)	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
File register (R)	32768		32768 (when the built-in memory is used)			131072 (built-in memory)	
Special link relay (SB)	1024		2048				
Special link register (SW)	1024		2048				
Step relay (S)	2048 (S0 to 127/block)		8192				
Index register (Z)	10		16				
Pointer (P)	300		4096				
Interrupt pointer (D)	128		256				
Special relay (SM)	1024		2048				
Special register (SD)	1024		2048				
Function input	16		16				
Function output	16		16				
Function register	5		5				

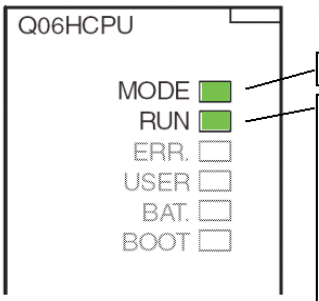
You can increase the number of file register for the Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, and Q25HCPU to up to 1 041 408 points by using a SRAM or flash card.

QnCPU – Operating Items

Feature	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Switch operation	RUN, STOP, RESET		RUN, STOP, RESET, L.CLR (Reset of the latched devices)				
External interfaces	RS232		RS232	RS232, USB			
Memory card	Not available		Available				
LED display	RUN, ERR.		MODE, RUN, ERR., USER, BAT., BOOT, POWER				
Current consumption @ 5 V DC	0.25 A	0.27 A	0.60 A	0.64 A			



LED Display



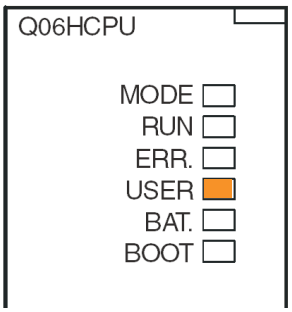
Green: Q mode	
ON:	During operation in "RUN" mode
OFF:	During "STOP" mode or after detection of an error occurrence that stopped the operation
Flicker:	RUN/STOP switch was switched from "STOP" to "RUN" after a program or a parameter was written during "STOP". The CPU is not in "RUN" mode.

Procedure to switch a Q CPU from "STOP" to "RUN" after the program or parameters have been changed during "STOP":

1. Switch the RESET/L.CLR switch to the "RESET" position.
2. Switch the RUN/STOP switch from "STOP" to "RUN".

However, when you want to set the CPU to "RUN" without clearing the device information, switch the RUN/STOP switch from "STOP" to "RUN", then back to "STOP" and finally to "RUN" again.

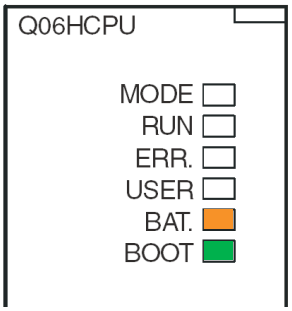
ERR and USER LED



ON:	**After the detection of an error during self-diagnostics. This error will not stop operation.**
OFF:	**Normal operation of the CPU**
Flicker:	**An error that stops the operation has been detected during self-diagnostic.**

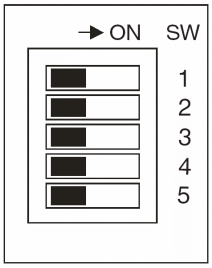
ON:	**An error has been detected by the CHK instruction or an annunciator (F) has been switched ON.**
OFF:	**Normal operation of the CPU**
Flicker:	**Execution of latch clear**

BAT and BOOT LED



ON:	**Voltage of either the battery for the CPU or the memory card is too low.**
OFF:	**Voltage is normal**

Q CPU DIP Switch Functions:



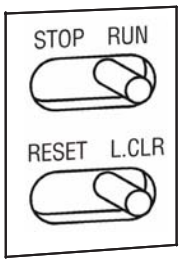
System protection
OFF: No protection
ON: Protection

Used to specify where parameters are stored		
SW2	SW3	Parameter drive
OFF	OFF	Program memory (Drive 0)
ON	OFF	SRAM card (Drive 1)
OFF	ON	Flash card/ATA card (Drive 2)
ON	ON	Built-in ROM (Drive 4)

Parameters can not be stored in the built-in RAM (Drive 3).

All switches are shipped in the OFF position.

RUN/STOP and RESET/L.CLR Switches



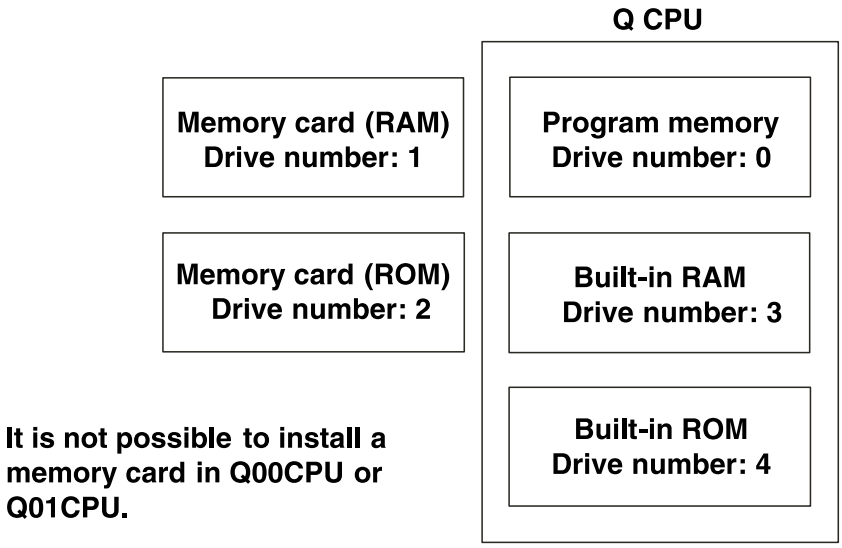
RUN: CPU executes sequence program

STOP: The execution of the sequence program is stopped

RESET: Used to perform hardware reset, operation fault reset, operation initialization etc.
After performing a reset, always return this switch to the neutral position.

L.CLR: Used to clear (turn either OFF or set to zero) all data in the parameter set latch area.
(Not available for Q00CPU and Q01CPU)

Memory Organisation



It is not possible to install a memory card in Q00CPU or Q01CPU.

Organisation of Storage

Q00CPU and Q01CPU

Data	Built-in memory		
	Programm memory (Drive 0)	RAM (Drive 3)	ROM (Drive 4)
Program	●	○	●
Parameters	●	○	●
Intelligent function module parameters	●	○	●
Device comment	●	○	●
File register	○	●	○

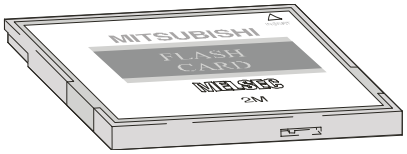
● = Storage is possible
○ = Storage is not possible

Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU and Q25HCPU:

Data	Built-in memory			Memory cards		
	Programm memory (Drive 0)	RAM (Drive 3)	ROM (Drive 4)	RAM (Drive 1)	Flash ROM (Drive 2)	ATA ROM (Drive 2)
Program	●	○	●	●	●	●
Parameters	●	○	●	●	●	●
Intelligent function module parameters	●	○	●	●	●	●
Device comment	●	○	●	●	●	●
Device initial value	●	○	●	●	●	●
File register	○	●	○	●	●	○
Local devices	○	●	○	●	○	○
Debugging data	○	○	○	●	○	○
Failure history	○	○	○	●	○	○
Data file written by a FWRITE instruction	○	○	○	○	○	●

● = Storage is possible
○ = Storage is not possible

Memory Card Specifications

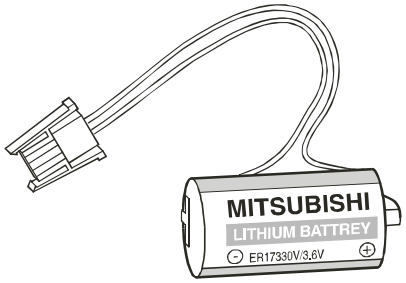


The write protect switch on the card will prevent any unintentional overwriting of stored data. A battery within the RAM memory card will hold the data during an interrupt of the power supply.

Available memory cards

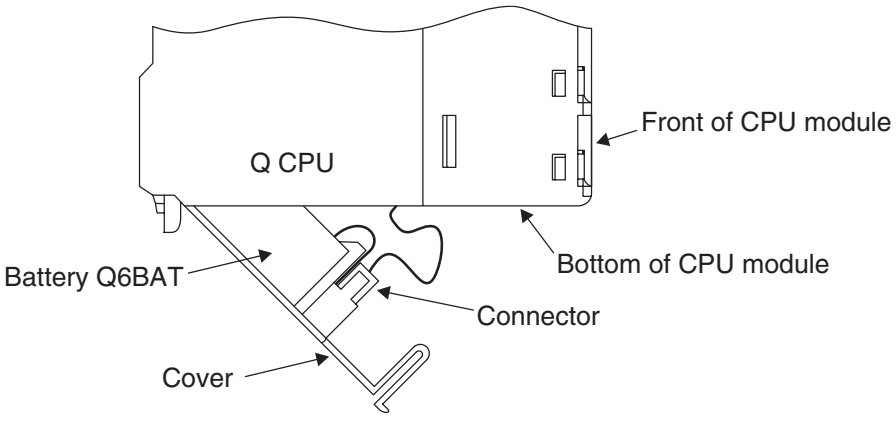
Designation	Type of memory	Memory capacity [Bytes]	Memory capacity [Number of files]	Number of writings
Q2MEM-1MBS	SRAM	1011 k	256	No limitation
Q2MEM-2MBS		2034 k	288	
Q2MEM-2MBF	Flash ROM	2035 k	288	100 000
Q2MEM-4MBF		4079 k		
Q2MEM-8MBA	ATA ROM	7940 k	512	1 000 000
Q2MEM-16MBA		15932 k		
Q2MEM-32MBA		31854 k		

Installation of the Battery for the CPU Module



The battery is installed at the bottom side of the Q CPU. During interruption of the power supply the battery can hold the data of the program memory , the built-in RAM and the clock for several thousand hours. However, this time depends on the type of CPU.

The CPU is shipped with its connector disconnected. Connect the battery before the CPU is used for the first time.



The battery should be changed every 10 years.

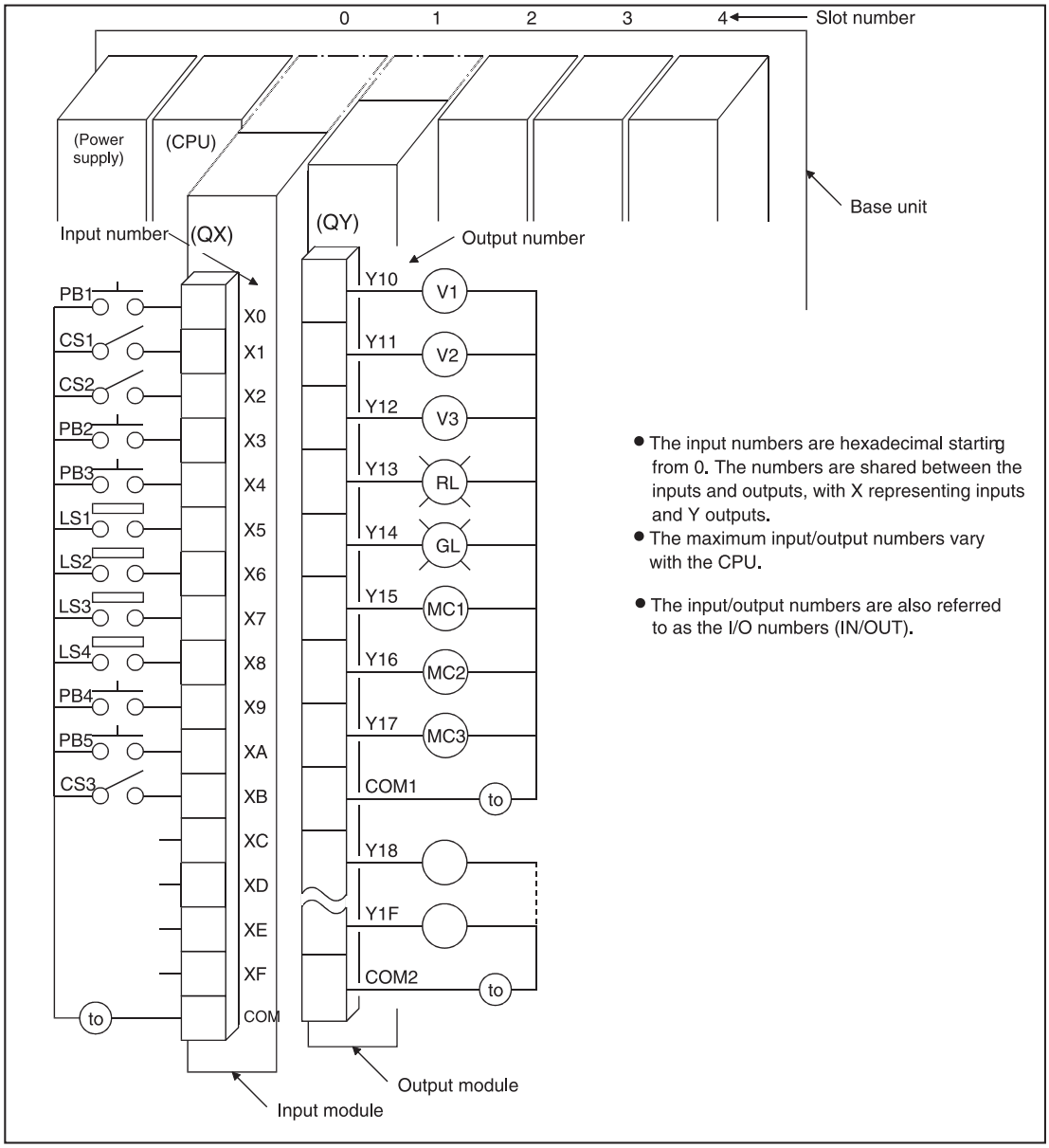
2.7 External I/O Signals and I/O Numbers

2.7.1 I/O device wiring

Signals from external input devices are replaced by input numbers, which are determined by the mounting position and terminal numbers of the input module connected and are handled in the program.

The outputs (coils) of the program operation results use output numbers which are determined by the mounting position and terminal numbers of the output module with which external output devices are connected.

As can be seen in the following examples, the I/O numbering system used is Hexadecimal. This is sensible as the PLC system is based on a 16 bit platform, it therefore follows that the addressing is also in this format.



Inputs & Outputs

The Q-Series range of controllers can be considered to be made up of three parts:

- CPU (Central Processing Unit)
- Input circuit
- Output circuit

The input circuitry provides the PLC CPU with information from a wide variety of input signals.

Typical Input Devices

The Input signals can come from a wide variety of devices i.e.

- Push buttons.
- Rotary switches.
- Key switches.
- Limit switches.
- Level sensors.
- Flow rate sensors
- Photo-electric detectors.
- Proximity detectors (Inductive or Capacitive).

  Proximity detectors usually provide a transistor output which can be either an NPN (Sink) or PNP (Source) transistor.

2.8 Digital Input and Output Modules

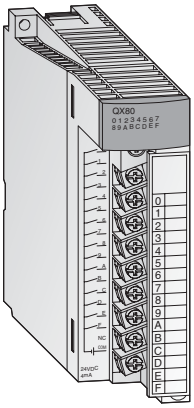
Overview of Digital I/O module types

Type		Number of inputs/outputs			
		8	16	32	64
Input modules	120 V AC	○	●	○	○
	240 V AC	●	○	○	○
	24 V DC	○	●	●	●
	24 V DC (High speed)	●	○	○	○
	5 V DC / 12 V DC	○	●	●	●
Output modules	Relay	●	●	○	○
	Individual relay	●	○	○	○
	Triac output	○	●	○	○
	Transistor output (sink)	●	●	●	●
	Transistor output (source)	○	●	●	○
Combined input/output modules		●	○	●	○

● = Module is available
○ = Module is not available

2.8.1 Digital Input Modules

Input modules are available for various input voltages:



Input voltage	Number of input points			
	8	16	32	64
5 – 12 V DC		QX70	QX71	QX72
24 V DC		QX80	QX81	QX82
24 V DC (Interrupt module)		QI60		
100 – 120 V AC		QX10		
100 – 240 V AC	QX28			

Modules with 8 or 16 connection points provide removable screw terminal blocks. The modules with 32 or 64 connection points are connected via a plug.

General PLC Input - Considerations

All inputs are isolated by Opto-couplers. This prevents the sensitive CPU electronics in the PLC from being affected by electrical noise spikes induced by external equipment.

Another common problem is contact bounce generated by electromechanical switches.

To avoid the PLC from being affected by these parasitic effects, the inputs are filtered so that the On/Off status will register an 'On' state only if the signal is stable for a period exceeding the filter coefficient (see note below).

This filter response time should be taken into account when programming as it will have a direct effect on the way the program will operate.

For the PLC to register a logical change in input condition, it will have to draw a minimum of 3mA; anything less than this will result in the Input not turning on.

The input will accept up to a 7mA signal, anything in excess of this could result in the input being damaged.

If higher speed input functionality is utilised where the input filter coefficient is reduced, care should be taken when using these inputs for digital signalling. Cables should be shielded and run separately to other potential sources of electrical noise!

If very high speed operation is required within the system then use of special modules such as or Interrupt of High Speed Counter should be adopted.

NOTE

A-Series: Standard Input Modules are preset to 10 ms Filter Coefficient.
Q-Series: The Filter Coefficient of the standard Input Modules is preset to 10 ms but may be individually adjusted in the range of 1 ms to 70 ms from within the Parameter setup of the CPU (See individual module specifications).

Source / Sink Inputs

This subject often causes confusion due to differing interpretations of the definition of Sink and Source by different manufacturer's each side of the Atlantic.

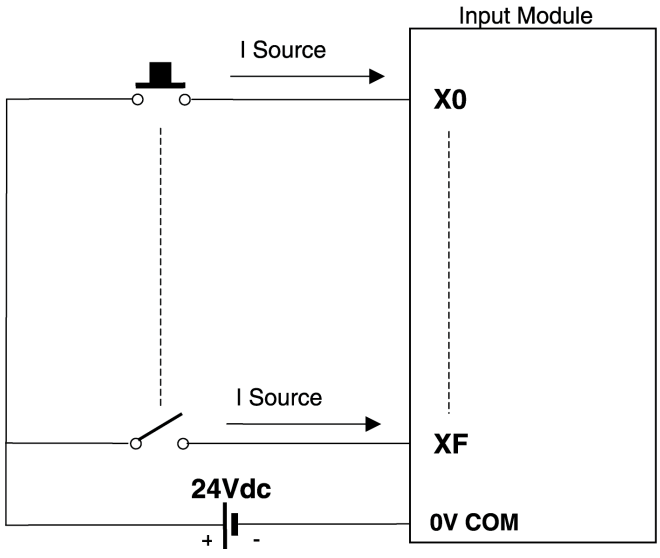
The term Source /Sink refers to the direction of current flow into or out of the input terminals of the PLC.

The following descriptions describe Mitsubishi's interpretation of the subject, which is shared by most other European and Far Eastern PLC manufacturers!

Source Input

When the PLC is connected for Source inputs, then the input signal current flows into the X inputs.

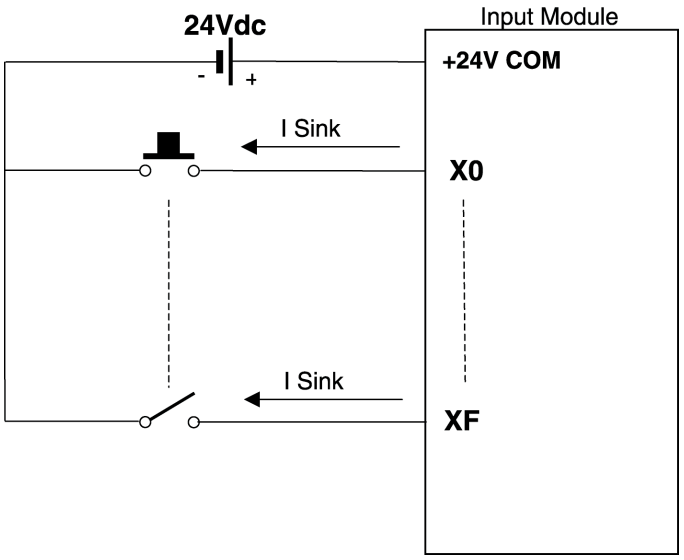
Source Input Configuration



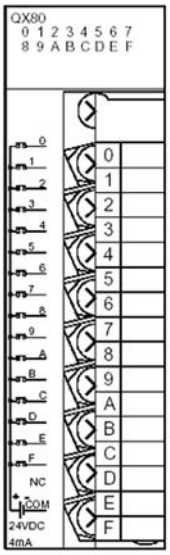
Sink Input

When the PLC is connected for Sink inputs, then the input signal current flows out of the X Inputs.

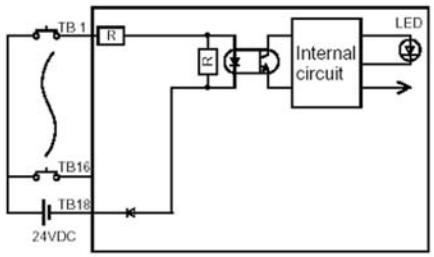
Sink Input Configuration



Source Inputs (Negative Common)- Module Details

Specifications \ Type		DC Input Module (Negative Common Type) QX80	Appearance
Number of input points		16 points	
Isolation method		Photocoupler	
Rated input voltage		24VDC (+20/-15%, ripple ratio within 5%)	
Rated input current		Approx. 4mA	
Input derating		No	
ON voltage/ON current		19V or higher/3mA or higher	
OFF voltage/OFF current		11V or lower/1.7mA or lower	
Input impedance		Approx. 5.6kΩ	
Response time	OFF to ON	1ms/5ms/10ms/20ms/70ms or less (CPU parameter setting) ✱ Initial setting is 10ms.	
	ON to OFF	1ms/5ms/10ms/20ms/70ms or less (CPU parameter setting) ✱ Initial setting is 10ms.	
Dielectric withstand voltage		560VAC rms/3 cycles (altitude 2000m (6557.38ft.))	
Insulation resistance		10MΩ or more by insulation resistance tester	
Noise immunity		By noise simulator of 500Vp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP2X	
Common terminal arrangement		16 points/common (common terminal: TB18)	
Number of I/O points		16 (I/O allocation is set as a 16-points input module)	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Applicable wire size		0.3 to 0.75mm^2 core (2.8mm (0.11in.) OD max.)	
Applicable crimping terminal		R1.25-3 (sleeved crimping terminals cannot be used.)	
5VDC internal current consumption		50mA (TYP. all points ON)	
Weight		0.16kg	

Input Circuit Detail

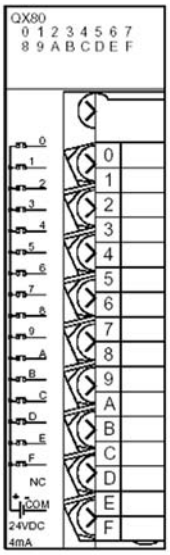
External Connections	Terminal Block Number	Signal Name
	TB1	X00
	TB2	X01
	TB3	X02
	TB4	X03
	TB5	X04
	TB6	X05
	TB7	X06
	TB8	X07
	TB9	X08
	TB10	X09
	TB11	X0A
	TB12	X0B
	TB13	X0C
	TB14	X0D
	TB15	X0E
	TB16	X0F
	TB17	Vacant
	TB18	COM

Direction of Source Current Flow

Referring to the preceding circuit diagram, when the push button is closed, the direction of current flow will be as follows:

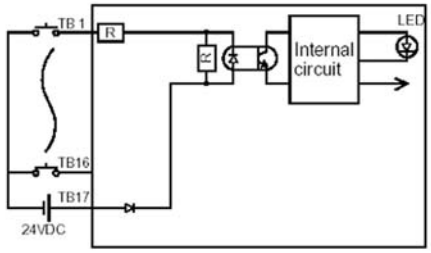
- From the +24 Volt terminal of the external power supply, through the push button and into the TB1 (X0) input terminal i.e. Source Current.
- Through the input resistor network circuit and then through the LED.
- When current flows through the LED it will emit light, which in turn will cause the Photo-Transistor to turn ON.
- The function of the Opto-Isolator is to isolate the plant side 24 Volt input circuit from the sensitive 5 Volt PLC processor logic circuitry. This also provides noise immunity from the input.
- With the Photo-Transistor turning ON, this will cause a signal to be sent to the Input Image Table, to store the information that the input X0 is ON.
- The Input current now flows out of (TB18) COM terminal and then back to the terminal of the External power supply.

Sink Inputs (Positive Common)- Module Details

Specifications \ Type		DC Input Module (Negative Common Type) QX80	Appearance
Number of input points		16 points	
Isolation method		Photocoupler	
Rated input voltage		24VDC (+20/-15%, ripple ratio within 5%)	
Rated input current		Approx. 4mA	
Input derating		No	
ON voltage/ON current		19V or higher/3mA or higher	
OFF voltage/OFF current		11V or lower/1.7mA or lower	
Input impedance		Approx. 5.6kΩ	
Response time	OFF to ON	1ms/5ms/10ms/20ms/70ms or less (CPU parameter setting) ✱ Initial setting is 10ms.	
	ON to OFF	1ms/5ms/10ms/20ms/70ms or less (CPU parameter setting) ✱ Initial setting is 10ms.	
Dielectric withstand voltage		560VAC rms/3 cycles (altitude 2000m (6557.38ft.))	
Insulation resistance		10MΩ or more by insulation resistance tester	
Noise immunity		By noise simulator of 500Vp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP2X	
Common terminal arrangement		16 points/common (common terminal: TB18)	
Number of I/O points		16 (I/O allocation is set as a 16-points input module)	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Applicable wire size		0.3 to 0.75mm^2 core (2.8mm (0.11in.) OD max.)	
Applicable crimping terminal		R1.25-3 (sleeved crimping terminals cannot be used.)	
5VDC internal current consumption		50mA (TYP. all points ON)	
Weight		0.16kg	

Input Circuit Detail

External Connections



Terminal Block Number	Signal Name
TB1	X00
TB2	X01
TB3	X02
TB4	X03
TB5	X04
TB6	X05
TB7	X06
TB8	X07
TB9	X08
TB10	X09
TB11	X0A
TB12	X0B
TB13	X0C
TB14	X0D
TB15	X0E
TB16	X0F
TB17	COM
TB18	Vacant

Direction of Sink Current Flow

In the preceding diagram, when the push button is closed, the direction of current flow will be as follows:

- From the +24 Volt terminal of the external power supply to the Common terminal (TB17) .
- Through the 1st LED and then through the input resistor network circuit to the TB1 (X0) input terminal.
- When current flows through the LED, it will then emit light which in turn will cause the Photo-Transistor to turn ON.
- The Photo-Transistor turning ON causes a signal to be sent to the Input Image Table, to store the information that the input X0 is ON.
- The Input current now flows out of the X0 input terminal i.e. 'Sink Current'.
- It then flows through the push button and then back to the negative (0V) terminal of the external power supply.

Sensors: Proximity and Optical

There are 2 types of proximity sensor; Inductive and Capacitive. There are also many varieties of optical sensors that may be found in Industrial application. The supply voltages to these sensors are commonly 24V DC.

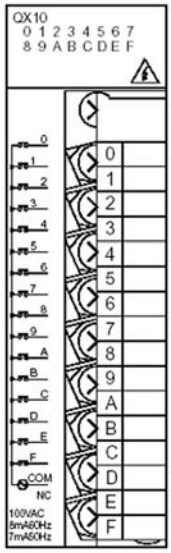
Most Opto and Proximity sensors utilise semiconductor outputs and these are available in two polarities, which are:

- PNP - (SOURCE)
- NPN - (SINK)

NOTE

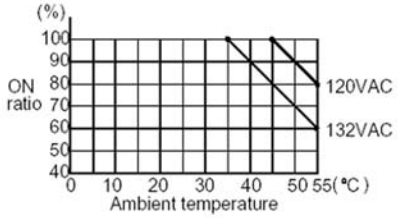
When connecting devices to the physical PLC I/O, think of current flow rather than voltage levels. For example: Input Activated = current flowing. Input Deactivated = No current flowing.

AC Input - Module Details

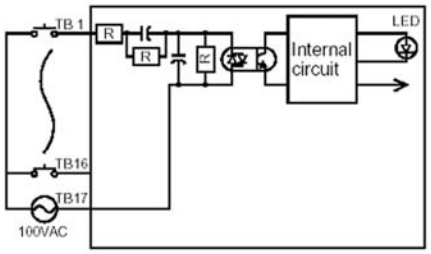
Specifications \ Type		AC Input Module	
		QX10	Appearance
Number of input points		16 points	
Isolation method		Photocoupler	
Rated input voltage, frequency		100-120VAC (+10/-15%) 50/60Hz (±3Hz) (distortion factor within 5%)	
Rated input current		Approx. 8mA (100VAC, 60Hz), approx. 7mA (100VAC, 50Hz)	
Input derating		Refer to the derating chart.	
Inrush current		Max. 200mA within 1ms (at 132VAC)	
ON voltage/ON current		80VAC or higher/5mA or higher (50Hz, 60Hz)	
OFF voltage/OFF current		30VAC or lower/1.7mA or lower (50Hz, 60Hz)	
Input impedance		Approx. 12kΩ (60Hz), approx. 15kΩ (50Hz)	
Response time	OFF to ON	15ms or less (100VAC 50Hz, 60Hz)	
	ON to OFF	20ms or less (100VAC 50Hz, 60Hz)	
Dielectric withstand voltage		1780VAC rms/3 cycles (altitude 2000m (6557.38ft.))	
Insulation resistance		10MΩ or more by insulation resistance tester	
Noise immunity		By noise simulator of 1500Vp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP1X	
Common terminal arrangement		16 points/common (common terminal: TB17)	
Number of I/O points		16 (I/O allocation is set as a 16-points input module)	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Applicable wire size		0.3 to 0.75mm^2 core (2.8mm (0.11in.) OD max.)	
Applicable crimping terminal		R1.25-3 (sleeved crimping terminals cannot be used.)	
5VDC internal current consumption		50mA (TYP. all points ON)	
Weight		0.17kg	

Input Circuit Detail

Derating Chart



External Connections

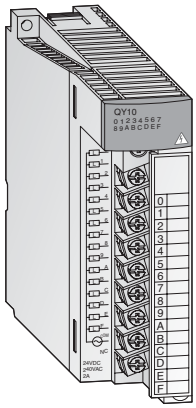


Terminal Block Number	Signal Name
TB1	X00
TB2	X01
TB3	X02
TB4	X03
TB5	X04
TB6	X05
TB7	X06
TB8	X07
TB9	X08
TB10	X09
TB11	X0A
TB12	X0B
TB13	X0C
TB14	X0D
TB15	X0E
TB16	X0F
TB17	COM
TB18	Vacant

With AC Input type modules, it is recommended that the same supply voltage to the PLC is used as for the inputs i.e. (100 - 120VAC). This minimises the possibility of an incorrect voltage being connected to the Inputs.

2.8.2 Digital Output Modules

The output modules of the Q-Series provide different switching elements for adaption to many control tasks:



Output type	Rated output voltage	Number of output points		
		8	16	32
Relay	24 V DC / 240 V AC	QY18A	QY10	
Triac	100 – 240 V AC		QY22	
Transistor	5 / 12 V DC		QY70	QY71
	12 / 24 V DC		QY80	QY81P
	5 – 24 V DC	QY68A		

Modules with 8 or 16 connection points are equipped with removable screw terminal blocks. The modules with 32 or 64 connection points are connected via a plug.

Output Types

Q-Series standard PLC outputs are available in four configurations:

- Relay
- Triac (SSR)
- Transistor (Source Type)
- Transistor (Sink Type)

Type	Advantages	Disadvantages
Relay	● Mixed voltage switching ● Volt-free operation possible ● High current switching capability	● Slow (max. 1 Hz) ● Finite reliability (electromechanical) ● Contact burn ● Noisy (electrical)
Triac	● High reliability ● Higher speed switching ● Suited to high duty switching applications	● AC operation only ● Current limited to 0.6 A /point ● Requires 10 ms to turn ON/OFF at AC 50 Hz
Transistor	● Very high reliability ● Very high speed switching ● Well suited to high duty switching applications	● Low voltage DC operation only ● Current limited to 0.1 A /point

Relay

This interface is more commonly used in the UK.

Electrical Isolation from the internal and external circuitry is achieved by the coils and the contacts of the output relays.

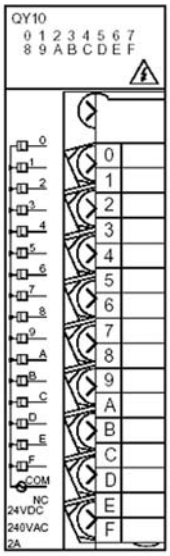
Modules are available as multiple outputs with isolated grouped commons or individually isolated ‘Volt Free’ outputs.

The operation of the output contact is driven by the internal CPU program.

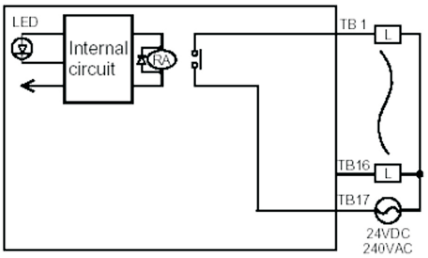
When the "END" instruction is triggered the PLC will REFRESH (update) the outputs from the Output Latch memory, an LED will light and the output contact will close.

The response for the operation of the relay is approximately 10 ms.

Relay Output Circuit Configuration

Specifications \ Type		Contact Output Module	
		QY10	Appearance
Number of output points		16 points	
Isolation method		Relay	
Rated switching voltage, current		24VDC 2A (resistive load) 240VAC 2A ($\cos\phi$ =1) /point, 8A/common	
Minimum switching load		5VDC 1mA	
Maximum switching load		264VAC 125VDC	
Response time	OFF to ON	10ms or less	
	ON to OFF	12ms or less	
Life	Mechanical	20 million times or more	
	Electrical	Rated switching voltage/current load More than 100 thousand times or more	
		200VAC 1.5A, 240VAC 1A ($\cos\phi$ =0.7) 100 thousand times or more 200VAC 0.4A, 240VAC 0.3A ($\cos\phi$ =0.7) 300 thousand times or more	
		200VAC 1A, 240VAC 0.5A ($\cos\phi$ =0.35) 100 thousand times or more 200VAC 0.3A, 240VAC 0.15A ($\cos\phi$ =0.35) 300 thousand times or more	
		24VDC 1A, 100VDC 0.1A (L/R=7ms) 100 thousand times or more 24VDC 0.3A, 100VDC 0.03A (L/R=7ms) 300 thousand times or more	
Maximum switching frequency		3600 times/hour	
Surge suppressor		No	
Fuse		No	
Dielectric withstand voltage		2830VAC rms/3 cycles (altitude 2000m (6557.38ft.))	
Insulation resistance		10MΩ or more by insulation resistance tester	
Noise immunity		By noise simulator of 1500Vp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP1X	
Common terminal arrangement		16 points/common (common terminal: TB17)	
Number of I/O points		16 (I/O allocation is set as a 16-points output module)	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Applicable wire size		0.3 to 0.75mm^2 core (2.8mm (0.11in.) OD max.)	
Applicable crimping terminal		R1.25-3 (sleeved crimping terminals cannot be used.)	
5VDC internal current consumption		430mA (TYP. all points ON)	
Weight		0.22kg	

Output Circuit Detail

External Connections	Terminal Block Number	Signal Name
	TB1	Y00
	TB2	Y01
	TB3	Y02
	TB4	Y03
	TB5	Y04
	TB6	Y05
	TB7	Y06
	TB8	Y07
	TB9	Y08
	TB10	Y09
	TB11	Y0A
	TB12	Y0B
	TB13	Y0C
	TB14	Y0D
	TB15	Y0E
	TB16	Y0F
	TB17	COM
	TB18	Vacant

Triac

Voltages of 240 V AC or 110 V AC can be used on separately commoned blocks.

As with all other output configurations the physical output is isolated by photocoupler.

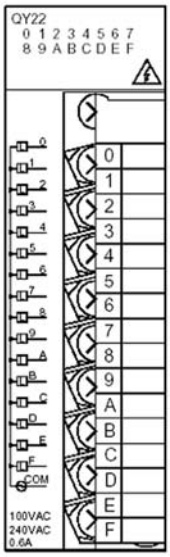
The response of the Triac is obviously faster than the relay with a response time of 1msec to turn ON and 10 ms to turn OFF again.

Care should be taken when configuring your system so as not to overload the output circuitry. Referral to the relevant hardware module manual will give the correct loading.

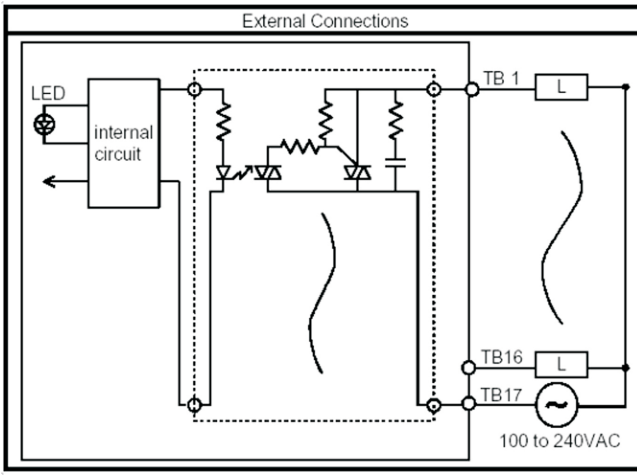
Because the leakage current in a Triac output circuit is greater than that of a relay circuit, care must be taken as this current is enough to cause indicators to illuminated and some miniature relays to hold their operation.
In fact, this is one of the most frequent causes of electric shock when working in cabinets controlled by PLC's.

Special care must be taken when working in live environments with output circuits controlled by Triac devices, even if the outputs are apparently turned off!

Triac Output Circuit Configuration

Specifications \ Type		TRIAC Output Module QY22	Appearance
Number of output points		16 points	
Isolation method		Photocoupler	
Rated load voltage		100-240VDC (+20/-15%)	
Maximum load current		0.6A/point, 4.8A/common	
Minimum load voltage/current		24VAC 100mA, 100VAC 25mA, 240VAC 25mA	
Maximum rush current		20A/cycle or less	
Leakage current at OFF		3mA or lower (for 240V, 60Hz), 1.5mA or lower (for 120V, 60Hz)	
Maximum voltage drop at ON		1.5V or lower	
Response time	OFF to ON	1ms + 0.5Hz or less	
	ON to OFF	1ms + 0.5Hz or less (rated load, resistance load)	
Surge killer		CR absorber	
Fuse		None (Attaching a fuse to external wiring is recommended. Refer to Section 1.2 (14))	
Dielectric maximum voltage		2830VAC rms/3 cycles (altitude 2000m)	
Insulation resistance		10MΩ or higher by insulation resistance meter	
Noise immunity		By noise simulator of 1.5kVp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP1X	
Common terminal arrangement		16 points/common (common terminal: TB18)	
Number of I/O points		16 (I/O allocation is set as a 16-points output module)	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Applicable wire size		Core cable: 0.3 to 0.75mm^2 (Outside diameter: 2.8mm or smaller)	
Applicable connector terminal		R1.25-3 (Terminals with sleeve cannot be used)	
5VDC internal current consumption		250mA (Max., all points ON)	
Weight		0.40kg	

Output Circuit Detail



Terminal Block Number	Signal Name
TB1	Y00
TB2	Y01
TB3	Y02
TB4	Y03
TB5	Y04
TB6	Y05
TB7	Y06
TB8	Y07
TB9	Y08
TB10	Y09
TB11	Y0A
TB12	Y0B
TB13	Y0C
TB14	Y0D
TB15	Y0E
TB16	Y0F
TB17	COM
TB18	Vacant

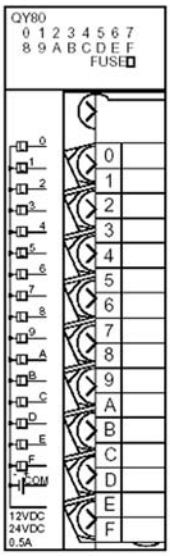
Transistor

As with all other output configurations the physical output is isolated by photocoupler.

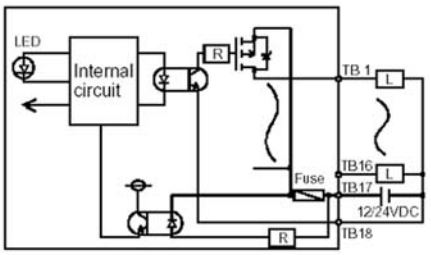
Response of the transistor in either direction is 1 ms at 24 V DC, 200 mA. The exact current handling capacity of each output is specified in the relevant hardware manual.

The Sink and Source Configurations are shown in the following module technical details.

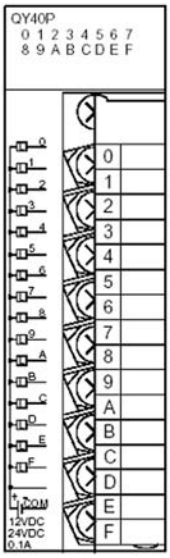
Source Transistor Output Circuit Configuration

Specifications \ Type		Transistor Output Module (Source Type) QY80	Appearance
Number of output points		16 points	
Isolation method		Photocoupler	
Rated load voltage		12-24VDC (+20/-15%)	
Maximum load current		0.5A/point, 4A/common	
Maximum inrush current		4A, 10ms or less	
Leakage current at OFF		0.1mA or less	
Maximum voltage drop at ON		0.2VDC (TYP.) 0.5A, 0.3VDC (MAX.) 0.5A	
Response time	OFF to ON	1ms or less	
	ON to OFF	1ms or less (rated load, resistive load)	
Surge suppressor		Zener diode	
Fuse		6.7A (unchangeable) (fuse blow capacity: 50A)	
Fuse blow indication		Yes (When fuse blows, LED indicates it and signal is output to CPU)	
External supply power	Voltage	12-24VDC (+20/-15%) (ripple ratio within 5%)	
	Current	20mA (at 24VDC)	
Dielectric withstand voltage		560VAC rms/3 cycles (altitude 2000m (6557.38ft.))	
Insulation resistance		10MΩ or more by insulation resistance tester	
Noise immunity		By noise simulator of 500Vp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP2X	
Common terminal arrangement		16 points/common (common terminal: TB17)	
Number of I/O points		16 (I/O allocation is set as a 16-points output module)	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Applicable wire size		0.3 to 0.75mm^2 core (2.8mm (0.11in.) OD max.)	
Applicable crimping terminal		R1.25-3 (sleeved crimping terminals cannot be used.)	
5VDC internal current consumption		80mA (TYP. all points ON)	
Weight		0.17kg	

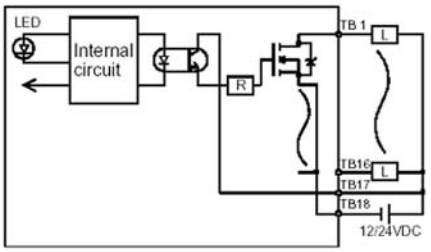
Output Circuit Detail

External Connections	Terminal Block Number	Signal Name
	TB1	Y00
	TB2	Y01
	TB3	Y02
	TB4	Y03
	TB5	Y04
	TB6	Y05
	TB7	Y06
	TB8	Y07
	TB9	Y08
	TB10	Y09
	TB11	Y0A
	TB12	Y0B
	TB13	Y0C
	TB14	Y0D
	TB15	Y0E
	TB16	Y0F
	TB17	COM
	TB18	0V

Sink Transistor Output Circuit Configuration

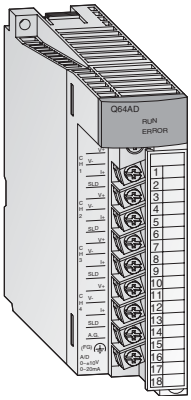
Specifications \ Type		Transistor Output Module (Sink Type)	
		QY40P	Appearance
Number of output points		16 points	
Isolation method		Photocoupler	
Rated load voltage		12-24VDC (+20/-15%)	
Maximum load current		0.1A/point, 1.6A/common	
Maximum inrush current		0.7A, 10ms or less	
Leakage current at OFF		0.1mA or less	
Maximum voltage drop at ON		0.1VDC (TYP.) 0.1A, 0.2VDC (MAX.) 0.1A	
Response time	OFF to ON	1ms or less	
	ON to OFF	1ms or less (rated load, resistive load)	
Surge suppressor		Zener diode	
Fuse		No	
External supply power	Voltage	12-24VDC (+20/-15%) (ripple ratio within 5%)	
	Current	10mA (at 24VDC) (Max. all points ON)	
Dielectric withstand voltage		560VAC rms/3 cycles (altitude 2000m (6557.38ft.))	
Insulation resistance		10MΩ or more by insulation resistance tester	
Noise immunity		By noise simulator of 500Vp-p noise voltage, 1μs noise width and 25 to 60Hz noise frequency	
		First transient noise IEC61000-4-4: 1kV	
Protection of degree		IP2X	
Common terminal arrangement		16 points/common (common terminal: TB18)	
Protection function		Yes (thermal protection, short circuit protection) • Thermal protection is activated in increments of 1 point. • Short circuit protection is activated in increments of 1 point.	
Operation indicator		ON indication (LED)	
External connections		18-point terminal block (M3×6 screws)	
Number of I/O points		16 (I/O allocation is set as a 16-points output module)	
Applicable wire size		0.3 to 0.75mm^2 core (2.8mm (0.11in.) OD max.)	
Applicable crimping terminal		R1.25-3 (sleeved crimping terminals cannot be used.)	
5VDC internal current consumption		65mA (TYP. all points ON)	
Weight		0.16kg	

Output Circuit Detail

External Connections	Terminal Block Number	Signal Name
	TB1	Y00
	TB2	Y01
	TB3	Y02
	TB4	Y03
	TB5	Y04
	TB6	Y05
	TB7	Y06
	TB8	Y07
	TB9	Y08
	TB10	Y09
	TB11	Y0A
	TB12	Y0B
	TB13	Y0C
	TB14	Y0D
	TB15	Y0E
	TB16	Y0F
	TB17	12/24VDC
	TB18	COM

2.9 Special Function Modules

2.9.1 Analog Input Modules

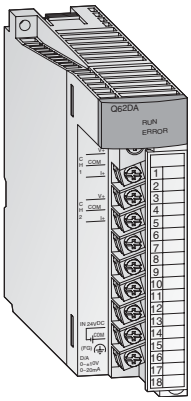


The analog input modules convert analog process signals into digital values which are further processed by the Q CPU. The A/D converter modules combine a high resolution (0.333 mV / 1.33 µA) with a high conversion speed (80 µs per channel).

All modules provide removable screw terminal blocks.

Analog input	Analog input range	Selectable input ranges	Input channels	
			4	8
Voltage	-10 to +10 V	1 to 5 V 0 to 5 V 0 to 10 V -10 to +10 V		Q68ADV
Current	0 to 20 mA	0 to 20 mA 4 to 20 mA		Q68ADI
Voltage or current (can be selected for each channel)	-10 to +10 V 0 to 20 mA	As for Q68ADV and Q68ADI	Q64AD	

2.9.2 Analog Output Modules



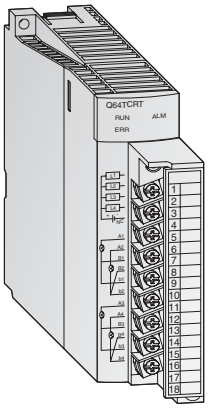
The analog output modules convert digital values into analog current or voltage signals. The resolution of 0.333 mV respectively 0.83 µA and the extremly short conversion time of 80 µs per output channel are only two of the many features of this modules. Isolation between process and control by means of optocouplers is also a standard feature.

All modules provide removable screw terminal blocks.

Analog output	Analog output range	Selectable output ranges	Output channels		
			2	4	8
Voltage or current (can be selected for each channel)	-10 to +10 V 0 to 20 mA	1 to 5 V -10 to +10 V 0 to 20 mA 4 to 20 mA	Q62DA	Q64DA	
Voltage	-10 to +10 V	-10 to +10 V			Q68DAV
Current	0 to 20 mA	0 to 20 mA 4 to 20 mA			Q68DAI

2.9.3 Temperature Control Modules with PID Algorithm

These modules enable PID algorithm temperatue control without placing any load on the Q CPU for the temperature control tasks.

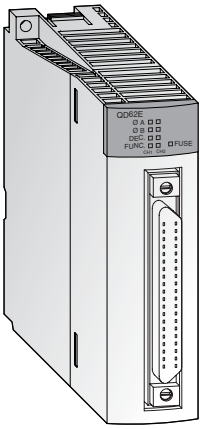


Special features

- 4 temperature input channels and 4 PID control circuits per module
- Input sensor types are either Pt100 temperature-measuring resistors (Q64TCRT and Q64TCRTBW) or thermocouples (Q64TCTT and Q64TCTTBW)
- The modules 64TCRTBW and Q64TCTTBW can detect the disconnection of a heater
- Auto tuning function for the PID control circuits
- Transistor output to drive the actuator in the control circuit

2.9.4 High -Speed Counter Modules

The modules QD62E, QD62, and QD62D detect signals at a frequency too high for normal input modules.



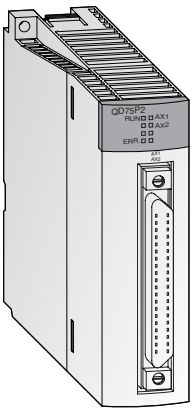
Special features

- Maximum counting frequency up to 500 kHz
- Input for incremental shaft encoder with automatic forward and backward detection
- Preset and selection of counter function via external digital inputs
- 32-bit counting range(-2 147 483 648 to +2 147 483 647)
- Can be used as up, down or ring counter
- All modules offer two counter inputs
- Two digital outputs which are set according to the counter value per counter input

All modules are connected via a plug.

2.9.5 Positioning Modules

In combination with stepper motors or servo amplifiers the modules QD75P1, QD75P2, and QD75P4 can be used for speed or position control.

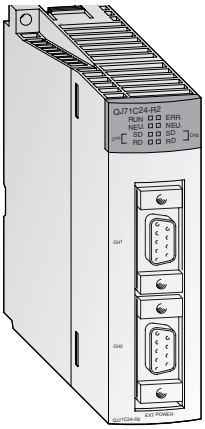


Special features:

- Control of up to four axes with linear interpolation(QD75P4) or two axes with circular interpolation (QD75P2 and QD75P4)
- Storage of up to 600 positional data sets in flash ROM
- Units of travel can be definedin pulses, µm, inches or degrees.
- Configuration and presetting of positional data is carried out by means of the PLC program or with the aid of the Microsoft Windows [TM] software GX Configurator QP.

2.9.6 Serial Communication Modules

The modules QJ71C24 and QJ71C24-R2 enable communications with peripheral devices via a standard serial interface.

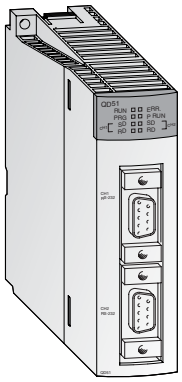


Special features:

- Two RS232C interfaces (QJ71C24-R2) or one RS422/485 and one RS232C interface (QJ71C24)
- Transmission speed up to 115200 bit/s
- Enables PCs connected to the PLC to access the full data set of the Q CPU
- Options for connection of a printer
- Integrated flash ROM memory for logging quality, productivity, or alarm data that can be transmitted when required.
- Support for plain ASCII data exchange. A user frame can be defined
- PLC programming and monitoring through the serial communication line is supported.

2.9.7 Intelligent Communication Modules

The modules QD51S-R24 and QD51 work through their own program(written in BASIC) independently of the Q CPU. Thus, data can be processed and communications can be performed with peripheral devices without imposing an additional load on the PLC CPU.



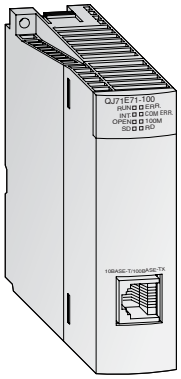
Special features:

- Either two RS232 interfaces (QD51) or one RS422/485 and one RS232 interface (QD51S-R24)
- Transmission speed of up to 38400 bit/s
- Access to devices in the Q CPU and tothe buffer memory of intelligent function modules is supported
- Remote RUN/STOP is supported via the serial communication line

2.9.8 ETHERNET Interface Modules

the modules QJ71E71/E71-100 and QD71E71-B2 are used on the PLC side to connect a host system, e.g. a PC or work station and the System Q via ETHERNET.

Besides the data transfer via TCP/IP or UDP/IP communications the reading and changing of PLC data as well as the monitoring of CPU module operation and control status is supported.

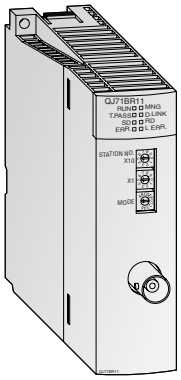


Special features

- Network types: 10BASE5, 10BASE2 or 10BASE-T
- Transfer rate of 10/100Mbit/s
- FTP-server functionality
- The communication function using fixed send and receive buffers is available.
- Up to 16 communication lines can be opened for concurrent data communication.
- PLC programming and monitoring can be performed fromGX Developer or GX IEC Developer on a personal computer via ETHERNET.

2.9.9 MELSECNET Modules

The modules QJ71BR11 and QJ71LP21 are used to connect the System Q to a MELSECNET/10 or MELSECNET/H network. This enables fast and effective communications between PLCs of the Q, QnA and QnAS series.

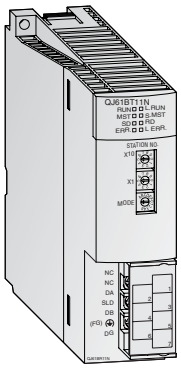


Special features

- Two different topologies are featured: Coaxial bus (QJ71BR11) or redundant optical loop (QJ71LP21).
- High data transfer rates: 10 Mbit/s with coaxial bussystems and optional 10 or 20 Mbit/s with optical loop systems
- Communications with other PLCs, PCs, or remote I/O
- The network system supports data communications between any two stations, no matter how many networks lie between them
- Station separating functionin coaxial bus system and loop back function in optical duplex loop systems in case of a station malfunction
- Control station shifting function and automatic return function

2.9.10 Master/Local Module for CC-Link

The QJ61BT11 is applicable as a master or local station in a CC-Link system and manages the connection of remote inputs and outputs.

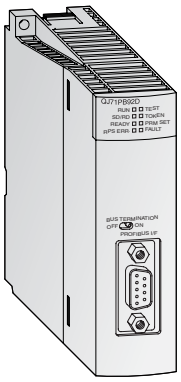


Special features

- The parameters of all modules across the network are set directly via the master module.
- The communications between the remote modules and the master module is performed automatically. The refresh time for 2048 I/O points is 3.3 ms only.
- Transmission speed of up to 10 Mbit/s
- With one master module a system can be extended to up to 2048 remote I/O points.
- An additional stand-by master establishes a duplex system. When an error occurs in the master station the datalink will be continued.
- Automatic CC-Link start without parameter setting
- Interrupt program start via network data command

2.9.11 PROFIBUS-DP Interface Module

The QJ71PB92D PROFIBUS-DP master module and the QJ71PB93D PROFIBUS-DP slave module enables PLCs of the System Q to communicate with other PROFIBUS devices.

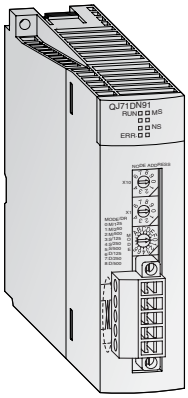


Special features

- The master station can communicate with up to 60 slave units.
- Up to 244 input bytes and 244 output bytes can be processed at a time per slave station.
- Supported functions include SYNC, FREEZE, and specialized diagnostic messages for the specific slave types used.
- Data excange with automatic refresh is supported . Batch transfer can be chosen as an option.

2.9.12 DeviceNet Module

The QJ71DN91 connects a Q series PLC with the DeviceNet. DeviceNet represents a cost-effective solution for the network integration of low-level terminal equipment.

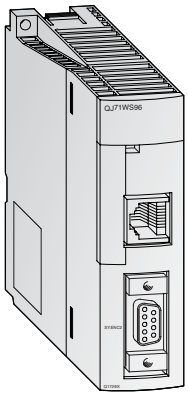


Special features

- The positions of master and slave stations are user selectable.
- Transfer rates of 125, 250 and 500 kBaud
- Transmission distances of up to 500 m
- Communication methods
  - Polling
  - Bit strobe
  - Change of state
  - Cyclic

2.9.13 Web Server Module

The web server module QJ71WS96 enables the remote control monitoring of a Q series PLC.



Special features

- Access to the PLC via the Internet
- Very easy setting functions integrated
- User needs only a Web browser for setting and monitoring
- RS232 interface for modem connection
- Various connections for data exchange are possible:
  ADSL, modem, LAN, etc.
- Sending and receiving data via mail or FTP
- Integration of a self-designed web site and Java applets is possible
- Standard connection via ETHERNET to exchange data between other PLCs or PCs
- Events and CPU data logging functions

2.10 Operation of a PLC

2.10.1 Programming Software

To be able to design a PLC program using a computer, it is essential for the software to have the following facilities:

- Programs can be designed using recognised and understandable conventions i.e. Relay Ladder diagrams and Instruction List formats.
- The functional integrity of programs may be tested prior to use on the chosen PLC.
- Programs can be permanently saved either on a computer's hard disk, or on removable media.
- Programs can be re- loaded from either the hard disk or the removable media.
- Ladder diagrams may be fully annotated.
- Hard copy print-outs can be obtained.
- The program can be transferred to and from the PLC, via a serial link.
- The Program operation can be monitored in 'real time'.
- Modifications can take place, whilst the PLC is On-line.
- Operational Parameters may be altered.
- Data memory areas may be saved and retrieved.
- Programs may be simulated on a PLC software emulator.

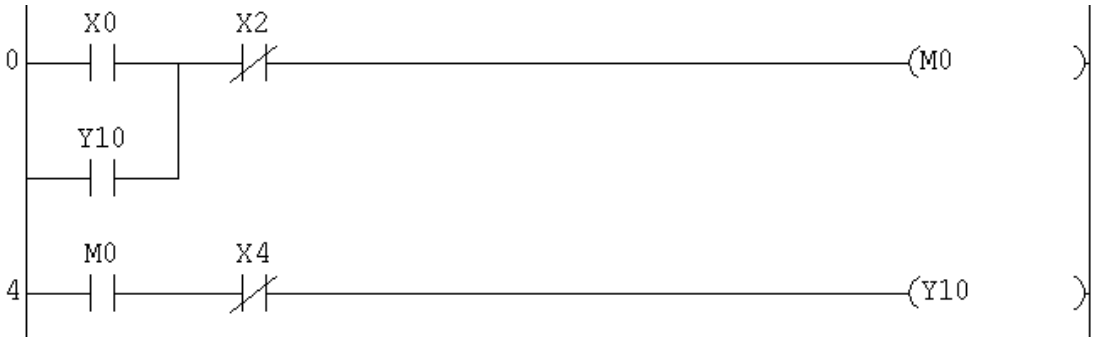
To name but a few!

2.10.2 Basic Operation of the Q-Series PLC System

Devices

PLC's like all computer systems, posses an internal structure. This could be described as a map of locations within the system. Every device in the system has a unique location called an Address. In the Mitsubishi Q-Series range of PLC's this is divided into numerous 'Device Names'.

To explain the basic operation of a PLC system, consider the following 2 networks of Ladder program.



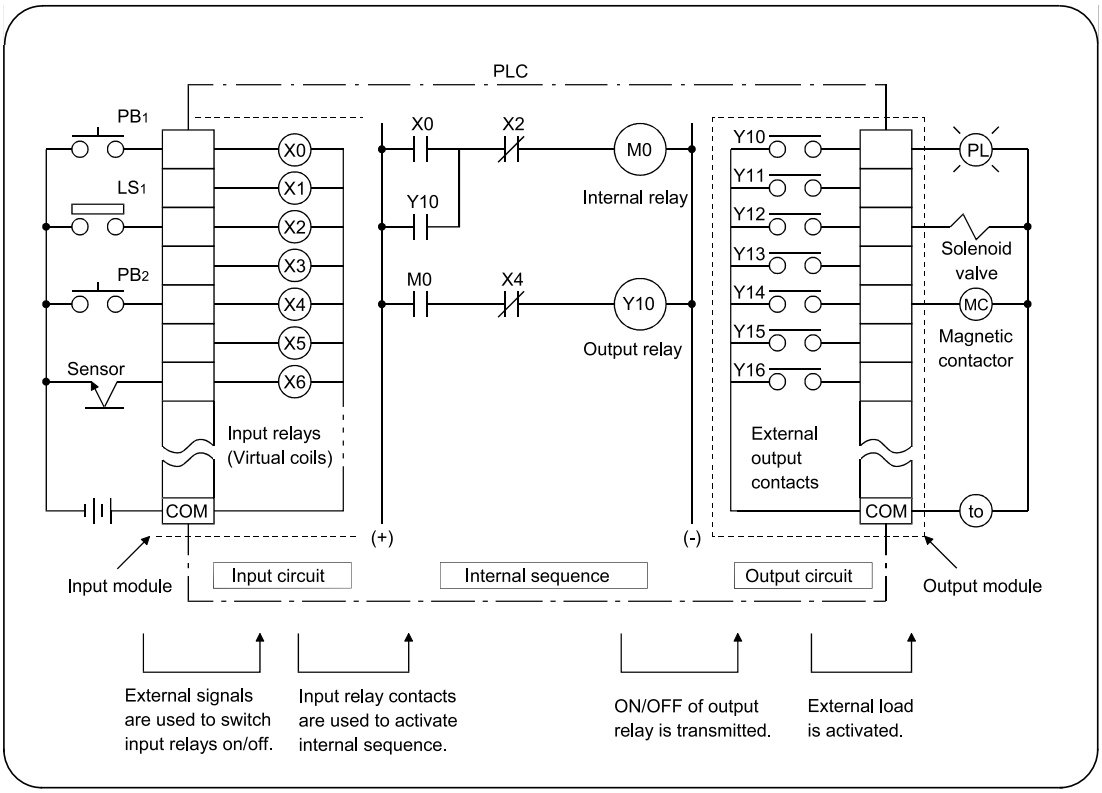
- Network1

  When Input X0 closes, providing X2 has not operated, this drives Internal Memory Coil M0. Y10 is in parallel with input X0 (MELSEC IL – “LD X0 OR Y10”). The condition of Y10 is dependant on the output of Y10 which is driven from Network 2 as described below:

- Network 2

  When the normally open contact of M0 closes and the normally closed contact of X4 has not operated, output Y10 becomes energised. Hence the circuit latches depending on the conditions set in Network1.

Based on the above circuit example, the following diagram helps to illustrate the operation of an Input/Output refresh cycle of a PLC system:



Principle of Operation

As can be seen from this illustration, the I/O PLC refresh cycle can be divided into three primary processes: **Input Processing, Program Processing and Output Processing.**

- Input Processing

  The Programmable Controller (PLC) initially reads the ON/OFF condition of all of the Inputs used in the program. These conditions are then stored into the Input Image Memory.

- Program Processing

  – The PLC then starts at the beginning of the PLC program and for each element of the program; it READS the actual logic state of that element, which is stored in either the Input Image Memory or the Output Image Memory.

  – If the required logic state is correct i.e. X0 is ON and X2 is OFF, the PLC will move on to the next element in the rung, i.e. M0.

- If M0 is ON, then logic 1 will be WRITTEN into the Output Image Memory in the location reserved for M0.
- If X0 is OFF, then logic 0 is WRITTEN into the M0 memory location.
- After an output instruction has been processed, the first element on the next line is executed, which in this example is a normally open contact of M0.
- Hence the logic state of the M0 memory location is this time READ from and if its logic state is at logic 1 indicating that the M0 coil is energised, this effectively means all M0 normally open contacts will now close. When the contact of M0 is closed and X4 open, a Logic 1 will be WRITTEN to the Image Memory Location reserved for the Output Y10.
- However if the contents of the M0 memory location are at logic 0, i.e. M0 is not energised, then a Logic 0 is WRITTEN to the Y10 Memory Location

- Output Processing

  Upon completion of the execution of all instructions, the contents of the Y memory locations within the Output Image Memory are now transferred to the Output Latch Memory and the Output Terminals.

  Hence any output, which is designated to be ON, i.e. Y10, will become energised.

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

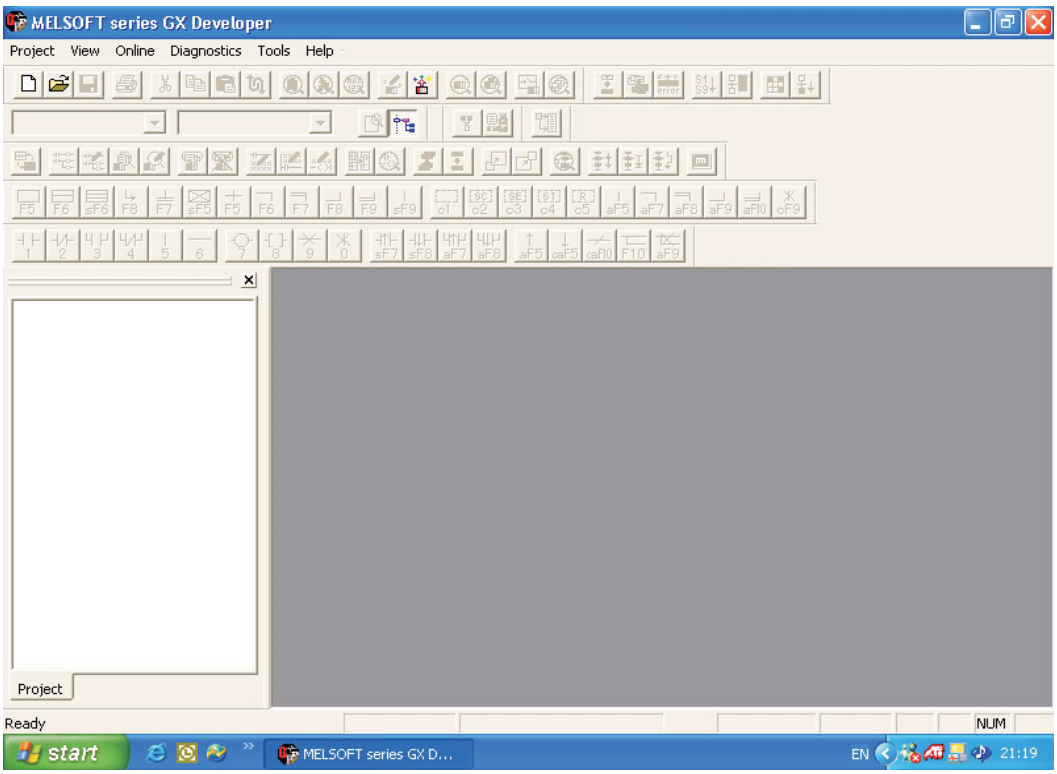
When using GX-Developer for the first time, it is advisable to alter some of the program defaults in order to optimise the working environment.

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

Procedure:

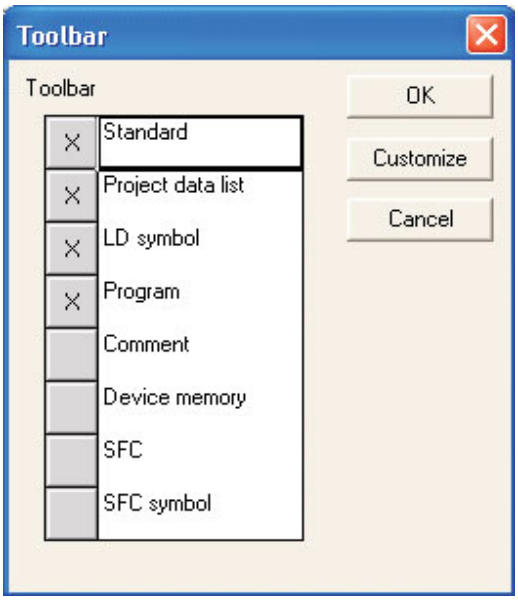
① From Windows Desktop, select ***GX Developer***.

② The display comes up as shown below.

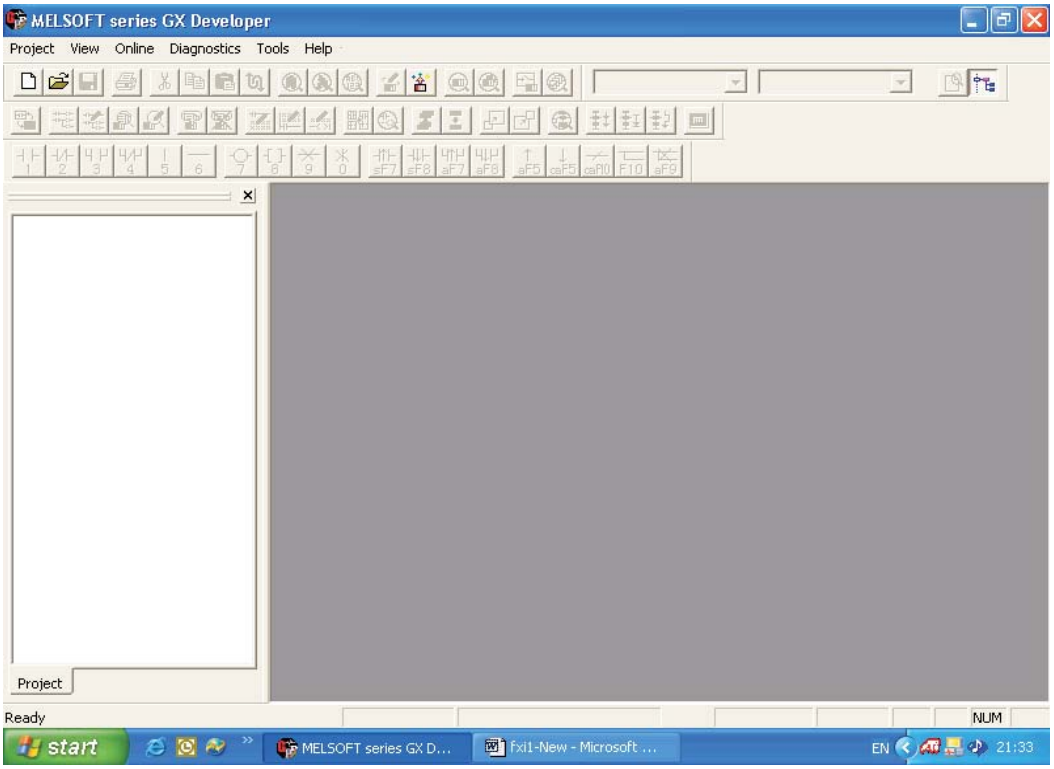


As can be seen from the above display there are large number of icons and this can be confusing to the first time user. Hence initially, it is recommended that only an essential minimum number of icons should be displayed.

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

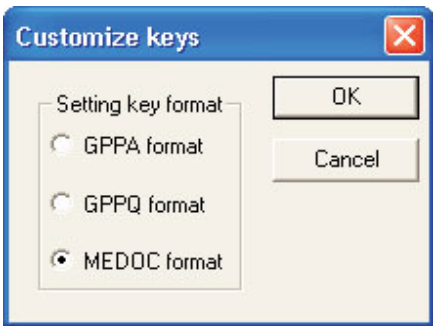


3.3 Setting the Shortcut Key Options

Procedure:

① From the ***Tools*** Menu Select ***Customize Keys***.

② Click on MEDOC from the ***Setting Key Format*** choice as below:



③ Press ***OK*** and the toolbar annotation will now alter to display the MEDOC shortcut Ladder Symbol Key List as shown below:

NOTE | All future references to this toolbar will infer MEDOC style shortcut keys as illustrated above.

4 Creating a Project

The following section describes the procedures required to create a new project with GX Developer using an example program Q-SERIES-PROG1.

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

4.1 Example PLC Program (Q-SERIES-PROG1)

This program enables a PLC output i.e. Y20, to be turned ON/OFF at a controlled rate. In this example the Output Y20 will be ON for 1 second and then OFF for 1 second. When Y20 is OFF then Y21 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 X10, 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 6
  - 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 Y20 will become energised i.e. Output Y20 will turn ON.

    Output Y21 will be de-energised i.e. Output Y21 will turn OFF.

- Lines 0 & 6

  After Timer T1 has been energised for 1 second, it will also operate and it's normally closed contact will open, causing Timer T0 to drop-out.

- With Timer T0 dropping-out, it's normally open contact will now re-open causing:
  - Timer T1 to drop-out.
  - Output Y20 to become de-energised i.e. Output Y20 will turn OFF and Y21 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.

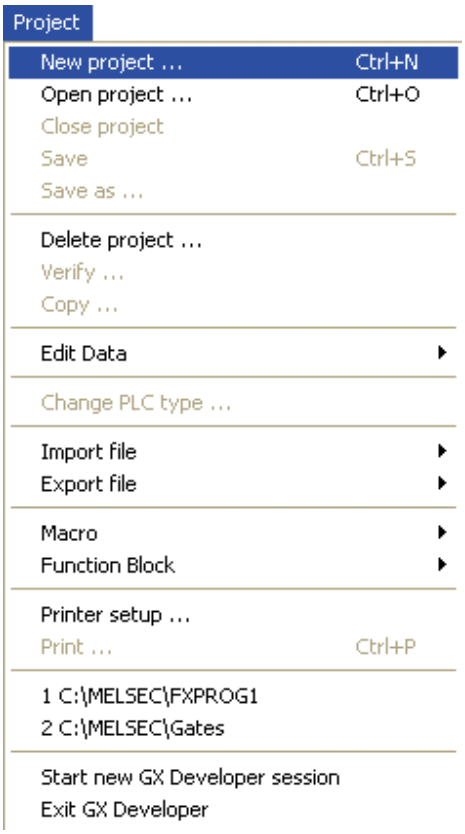
- With Timer T1 dropping out, its normally closed contact will close, and for as long as Input X10 is closed; the operation will be constantly repeated.

- Lines 6 & 12

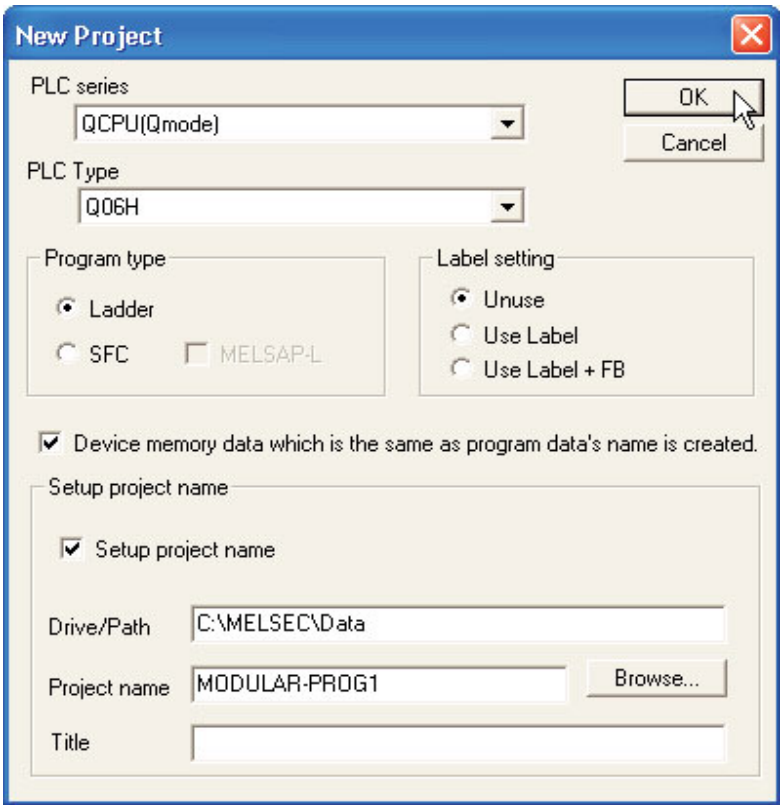
  Hence the Output Y20 will be continuously OFF for 1 second and then ON for 1 second and visa versa for output Y21 (Line12).

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: QCPU (Q Mode) or ACPU depending on type used.

– 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

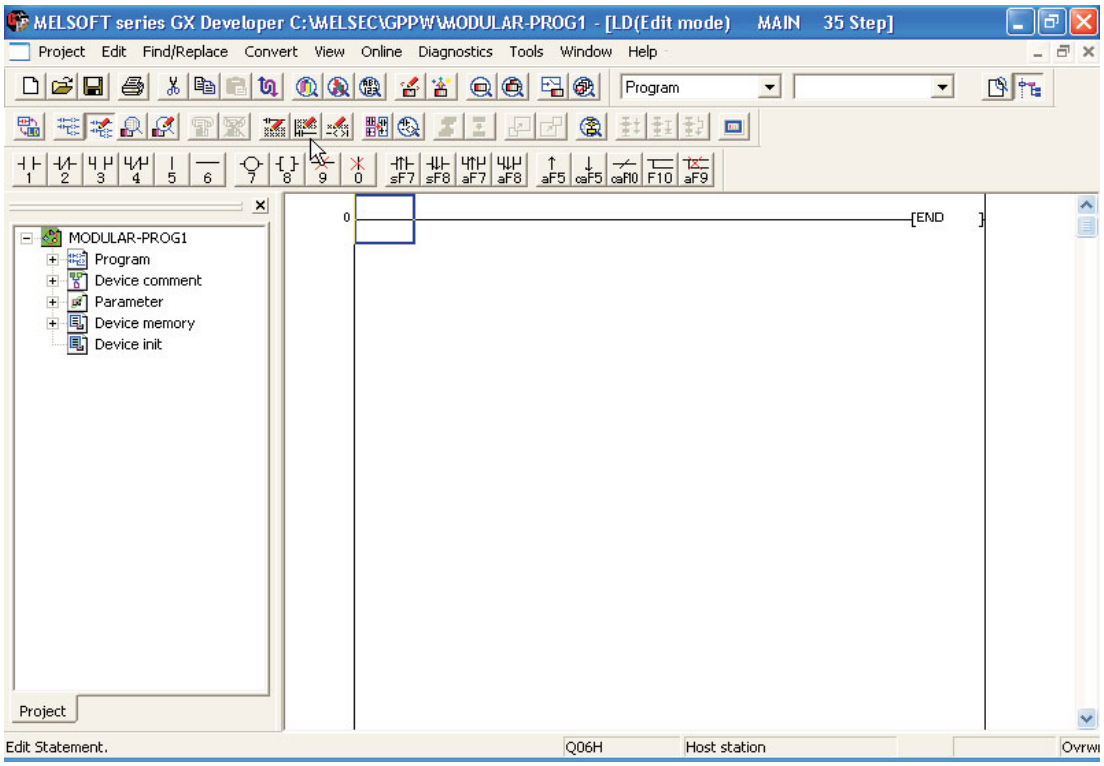
– For this example, use the Project Name: Q-SERIES-PROG1.

– 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.

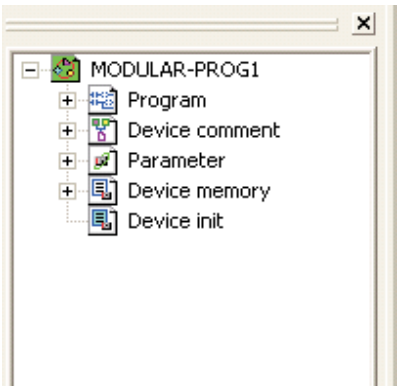
- Normally Open contact 
- Normally Closed contact 
- Normally Open Parallel contact 
- Normally Closed Parallel contact 
- Vertical Line 
- Horizontal Line 
- Output coil 
- Function Command 

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

- Using the mouse and selecting the required element.
- Entering the number 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:

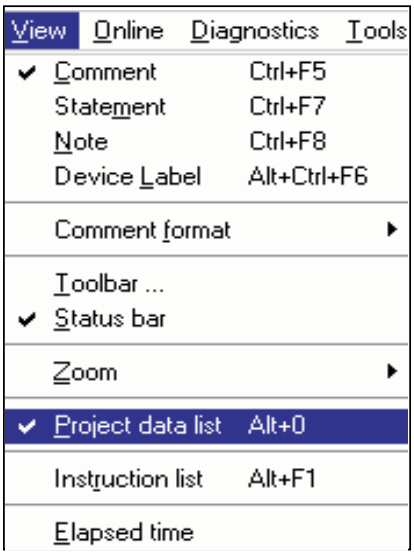


4.5 Toggle Display Project Data List

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.

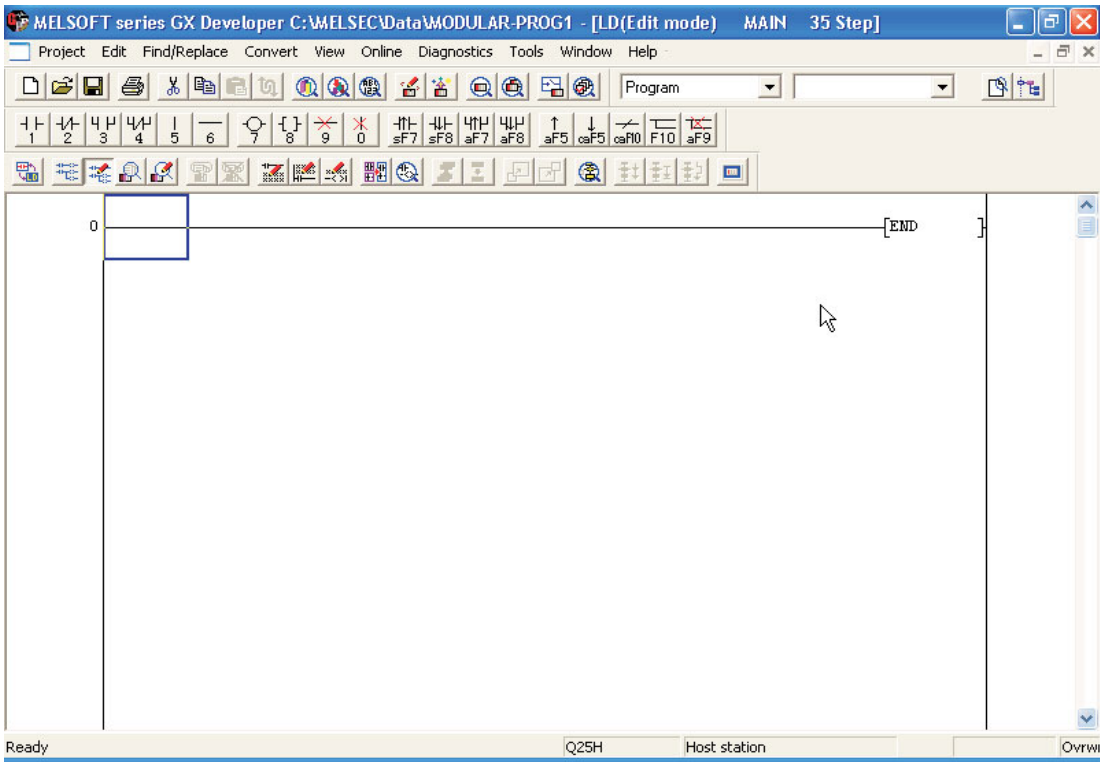
To remove the Project Data List from the displayed area, the following procedure should be adopted.

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:

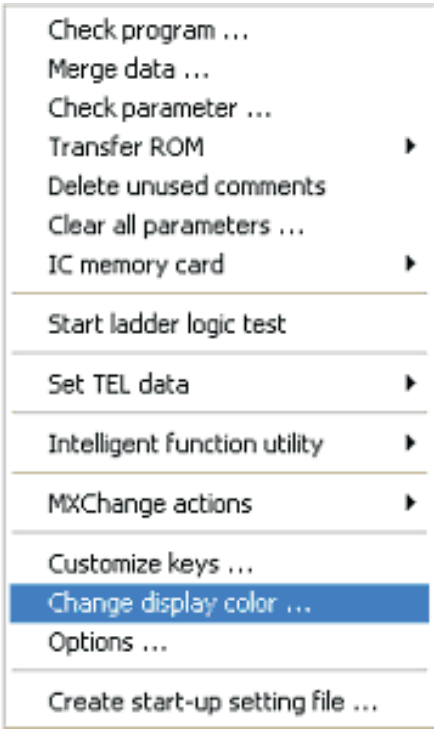


4.6 Changing the Colour Attributes (Optional)

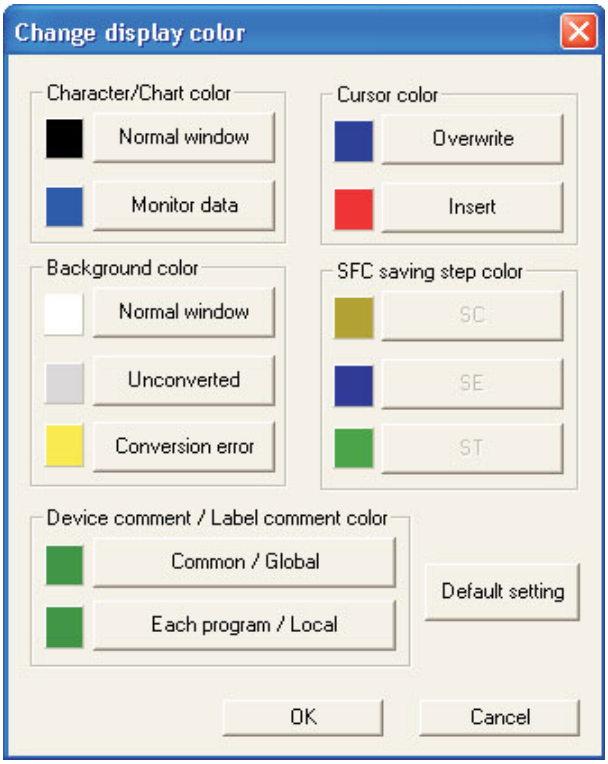
Changing the Colour Attributes is optional. Before Proceeding with the Ladder Program it is recommended to carry out the following procedure:

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 (Q-SERIES-PROG1)

The Ladder Diagram of Q-SERIES-PROG1 as shown at the beginning of this section will now be entered.

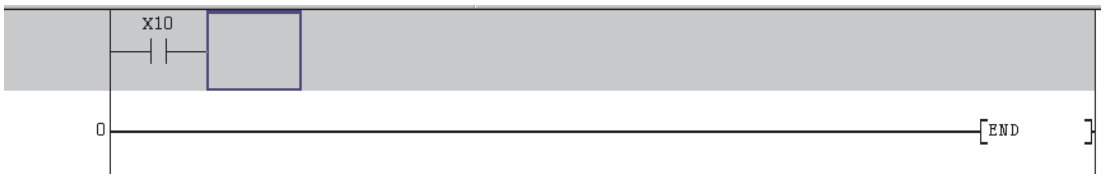
① Entering the first contact, Normally Open X10

– Using the mouse or “1” from keyboard, select the normally open contact.

– Enter the name X10.

– 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:

– 7

– T0

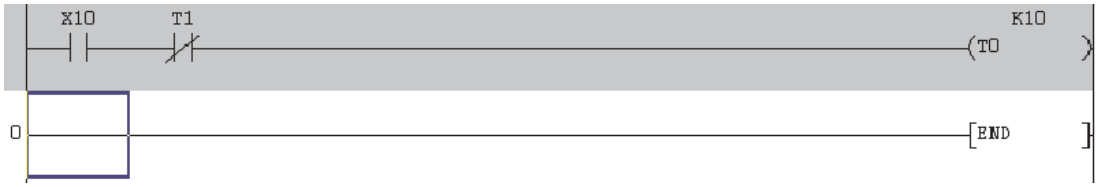
– Space

– K10

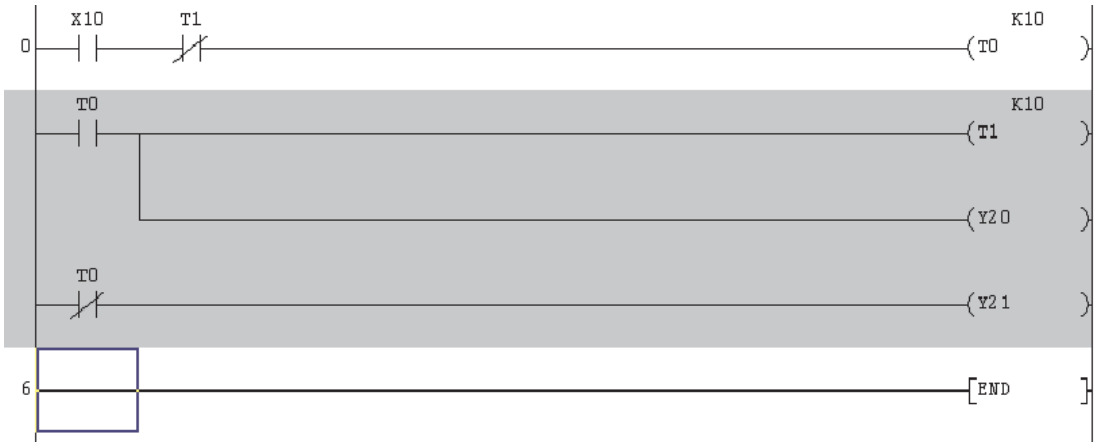
– ***OK***

NOTE Unlike MEDOC, a Space and not <ENTER> is used between the Timer T0 and its time delay value K10.

– 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

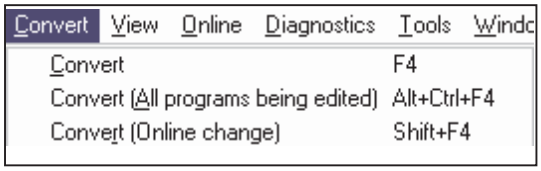
Before the program can be saved, the Ladder Diagram must first of all be converted into a set of MELSEC™ instructions.

(Incidentally, “MELSEC” is the brand name used by Mitsubishi Electric for their PLC products and is derived from the term: “**M**itsubishi **El**ectric **Se**quen**c**ers”)

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

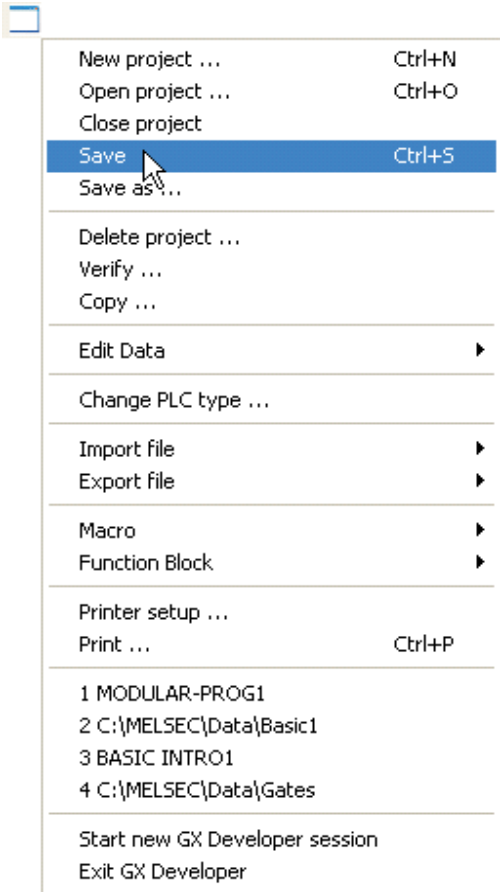
To save the project on the hard drive, carry out the following.

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\Q-SERIES-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.

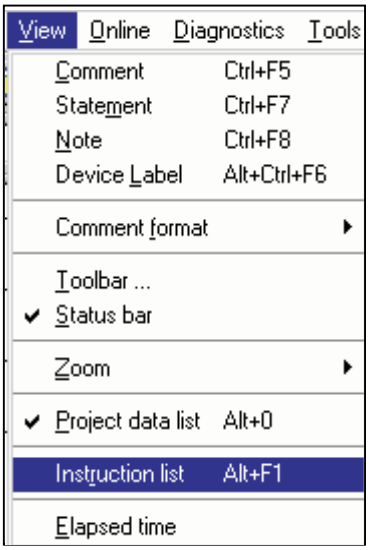
Where GX Developer has been used to produce a Ladder Diagram, then the equivalent Instruction Program can easily be displayed.

5.1 Instruction List Program (Q-SERIES-PROG1)

To obtain the equivalent Instruction Program for Q-SERIES-PROG1, carry out the following.

① From the Main Menu select,

- View
- Instruction List

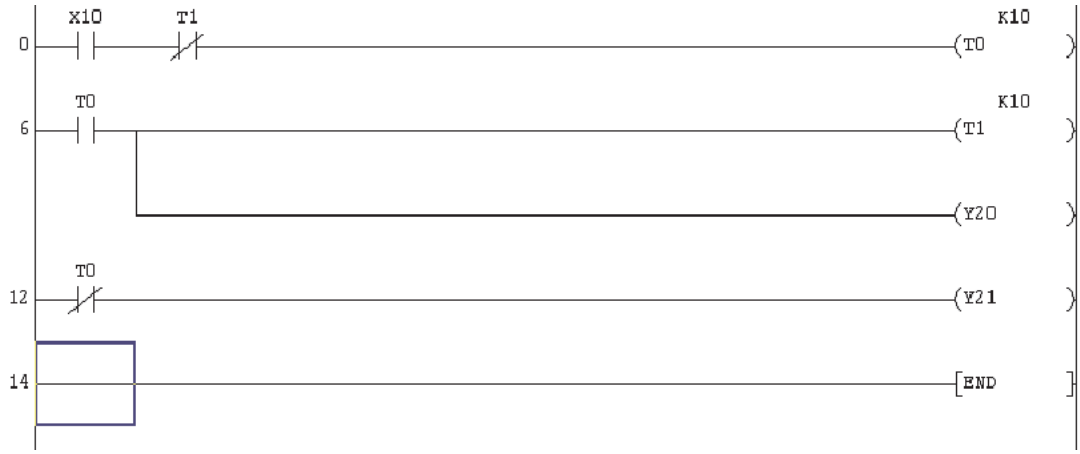


② Displayed on the screen will be the Instruction List representation of the Program for Q-SERIES-PROG1.

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- Q-SERIES-PROG1



Instruction Program – Q-SERIES-PROG1

```
  0    LD        X10
  1    ANI       T1
  2    OUT       T0        K10
  6    LD        T0
  7    OUT       T1        K10
 11    OUT       Y20
 12    LDI       T0
 13    OUT       Y21
 14    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 ‘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:

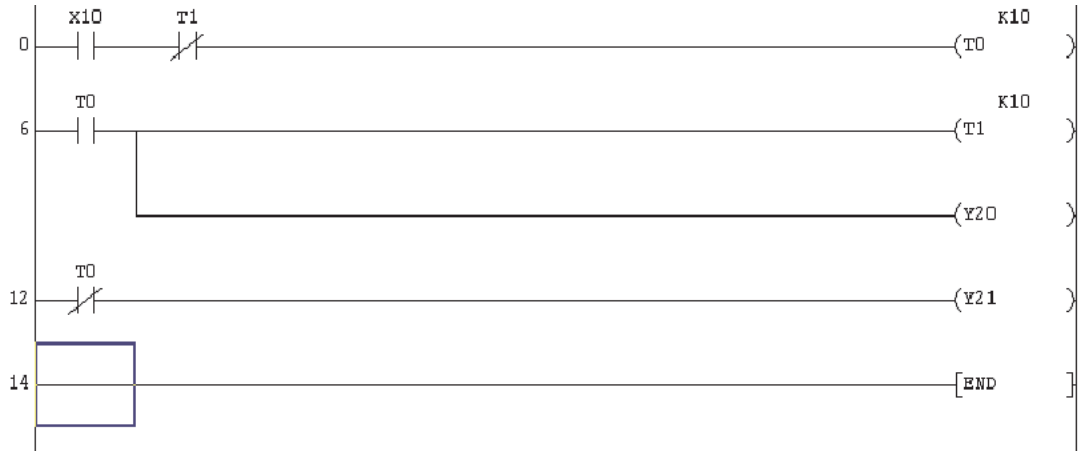
- An immediate jump to a particular Step Number.
- A search for a particular Element.

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.

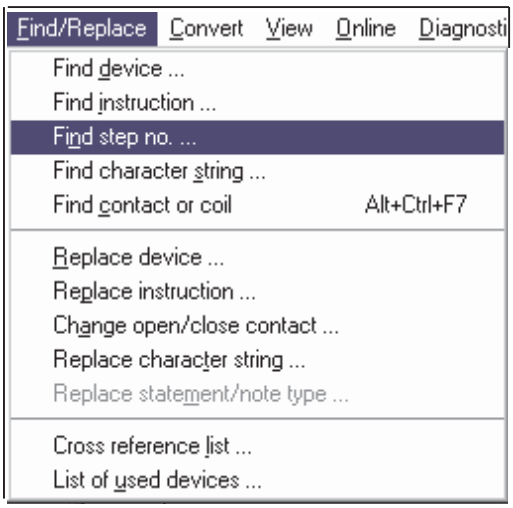
To use this facility, carry out the following:

① Let the Project Q-SERIES-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 6, <OK>.

Note that the program immediately jumps to the start of Line 6.

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.

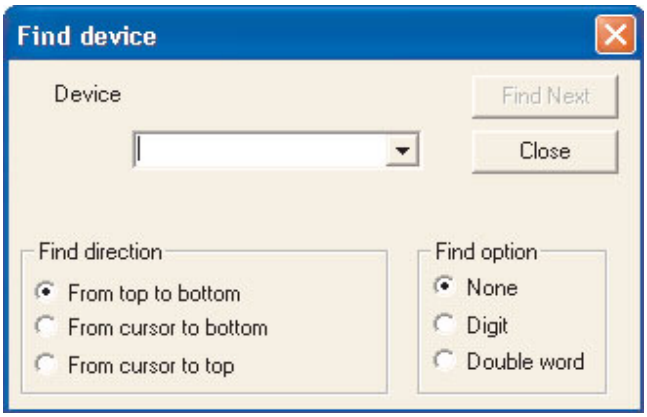
6.2 Find Device

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 Q-SERIES-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 Q-SERIES-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 6.

⑥ Select ***Find Next*** once more and note the next occurrence of T0 at Line 12.

⑦ Continue selecting ***Find Next*** until all of the T0 elements have been found i.e. when the message on the right is displayed. 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 Q-SERIES-PROG1, a search is carried out for the "Normally Closed Contact" of T1. It will be assumed that the Ladder Diagram Q-SERIES-PROG1 is being displayed.

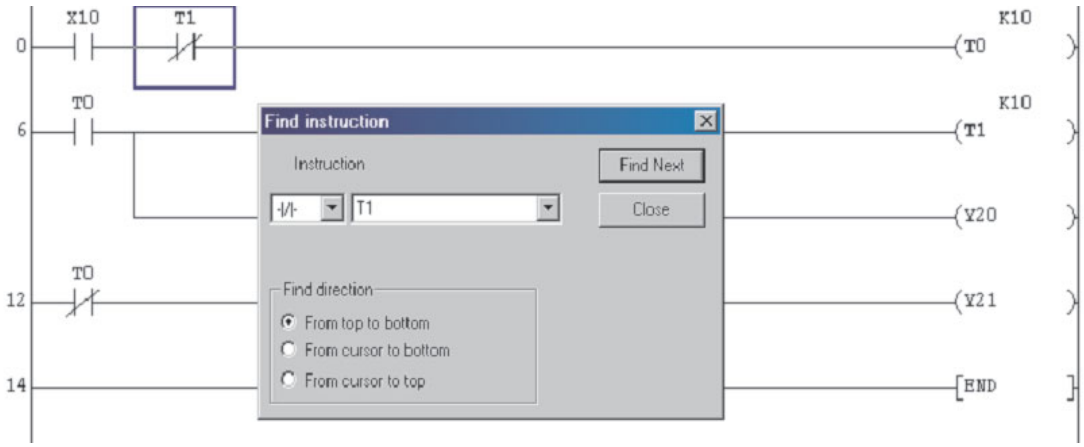
① From the Main Menu select the following.

- ***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.

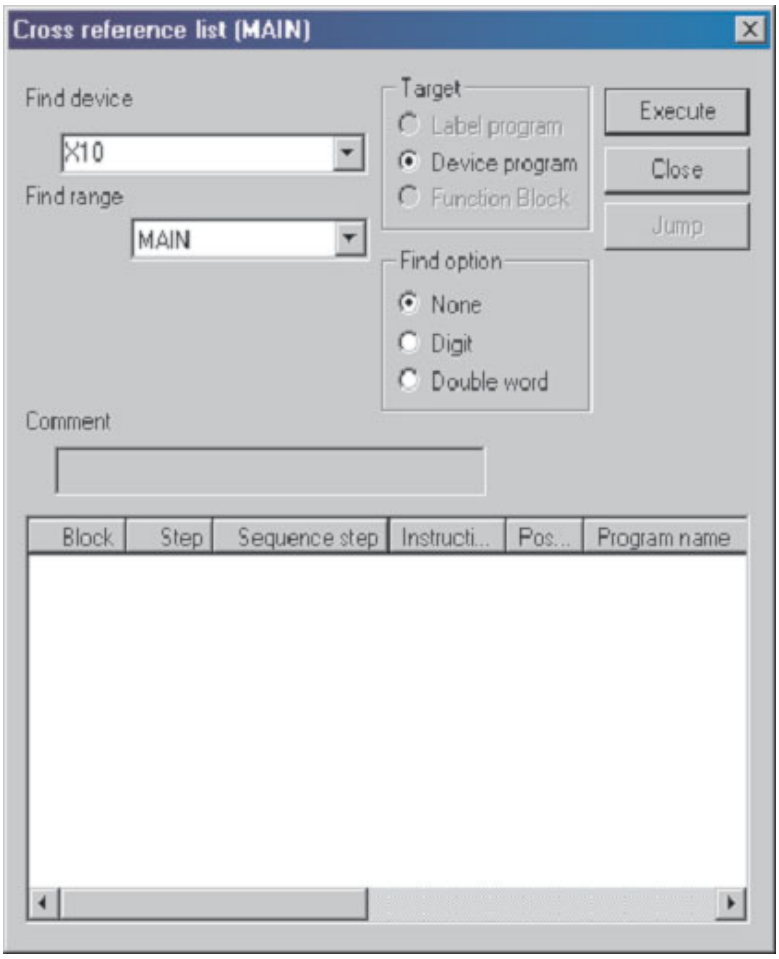
This is very important when fault finding a project and there is a need to track a particular device through the ladder diagram.

The following procedure describes how the Cross Reference details for the Timer T0 in the project Q-SERIES-PROG1 are obtained.

① From the Main Tool Bar select ***Find/Replace***.

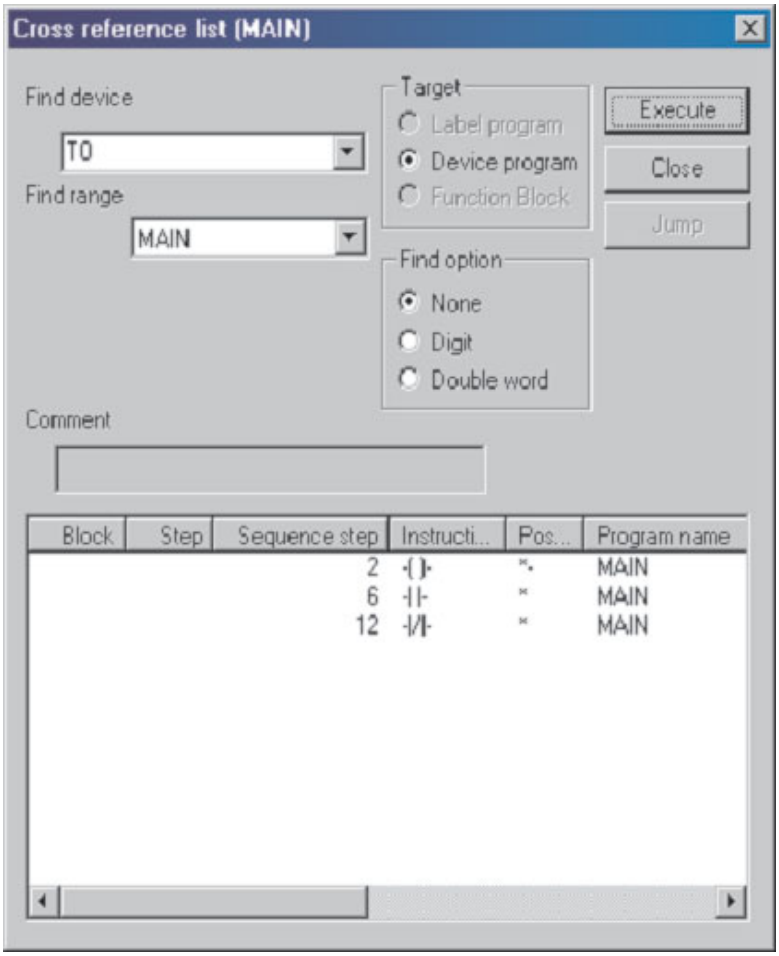
② Select ***Cross reference list***.

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 Q-SERIES-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.

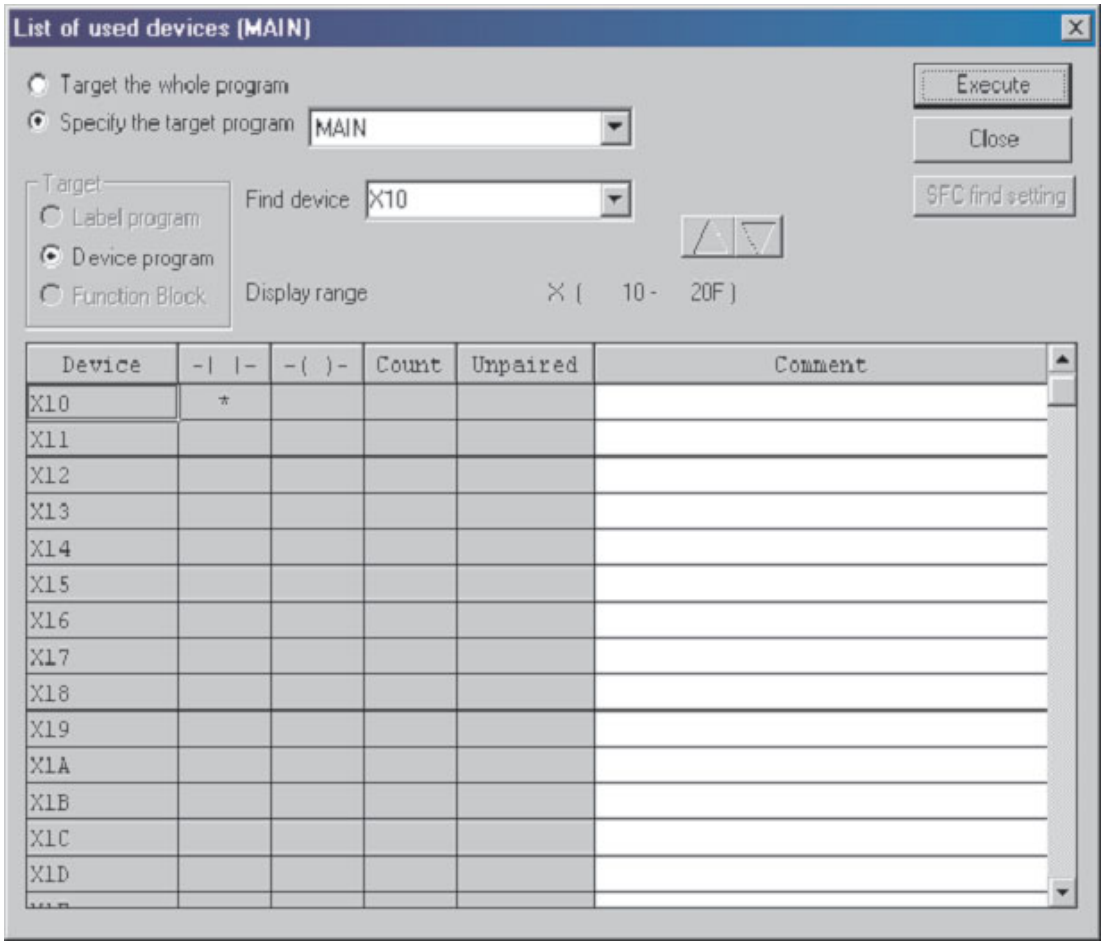
This is very useful when modifications to the ladder diagram are required, as it shows therefore what devices are not being used and hence those devices are available for use in the modification to the program.

The following procedure describes how all of the timers used in the project Q-SERIES-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:



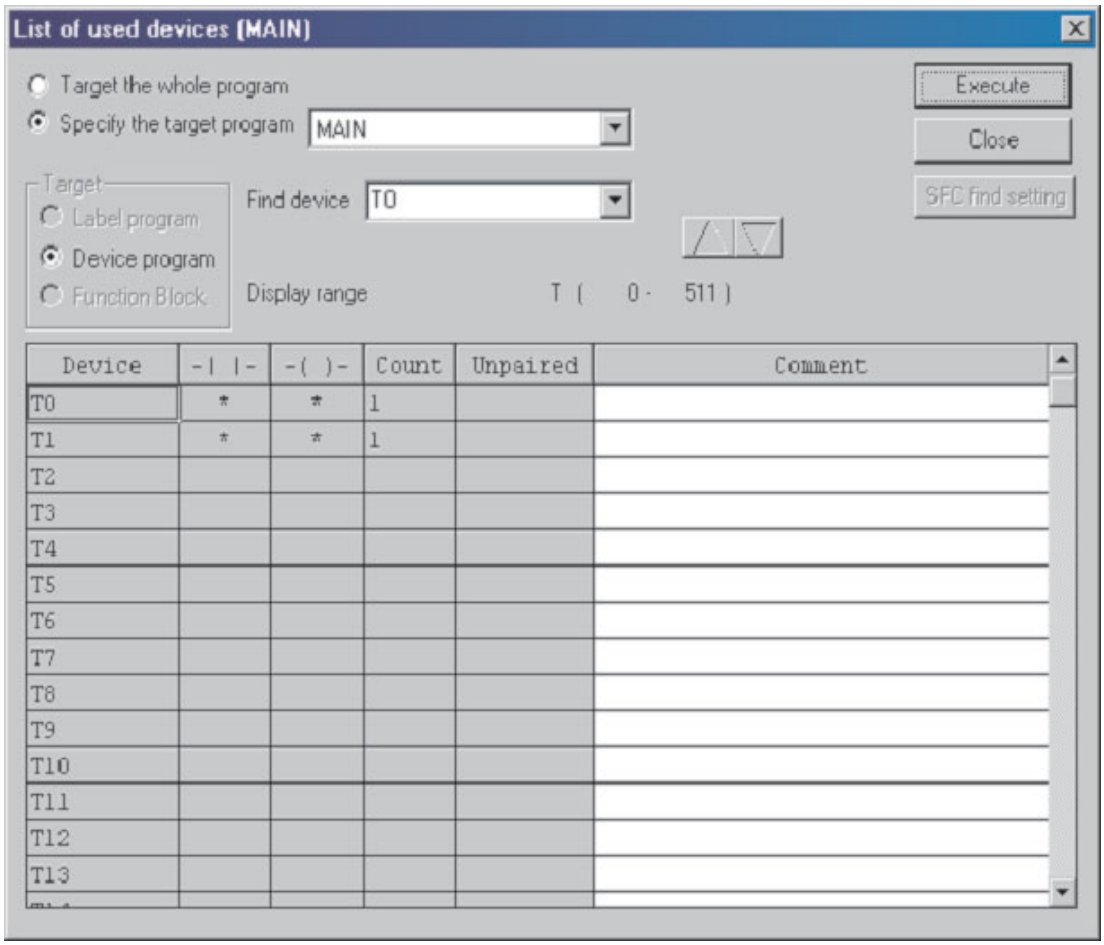
④ As can be seen from the previous display, an entire range of X input devices stating with X0 are being displayed.

⑤ In addition it can be seen there is a '*' in the contact column for X0.This indicates that X0 is used in the project Q-SERIES-PROG1.

⑥ Enter T0 in the ***Find device*** window.

⑦ Select ***Execute*** and the display shows that Timers T0 and T1 are being used in the project Q-SERIES-PROG1.

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



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.

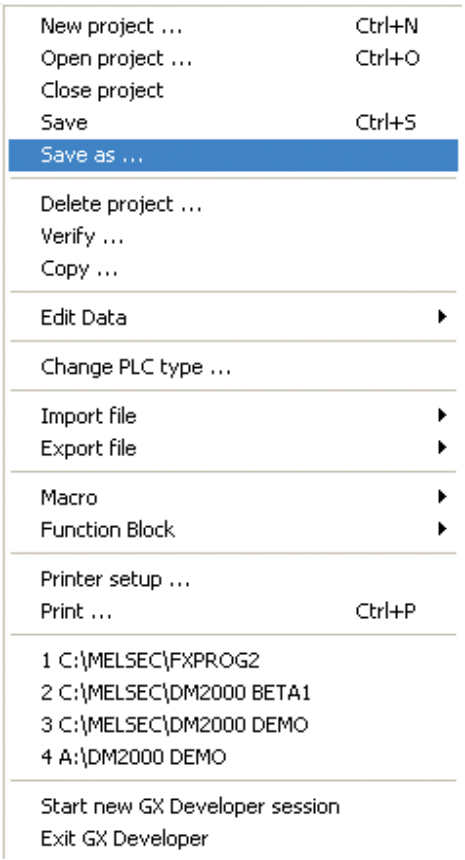
This is necessary in case the modifications do not work as expected and therefore the original project has to be re-loaded into the PLC, so that production can be maintained.

7.1 Copying of the project Q-SERIES-PROG1

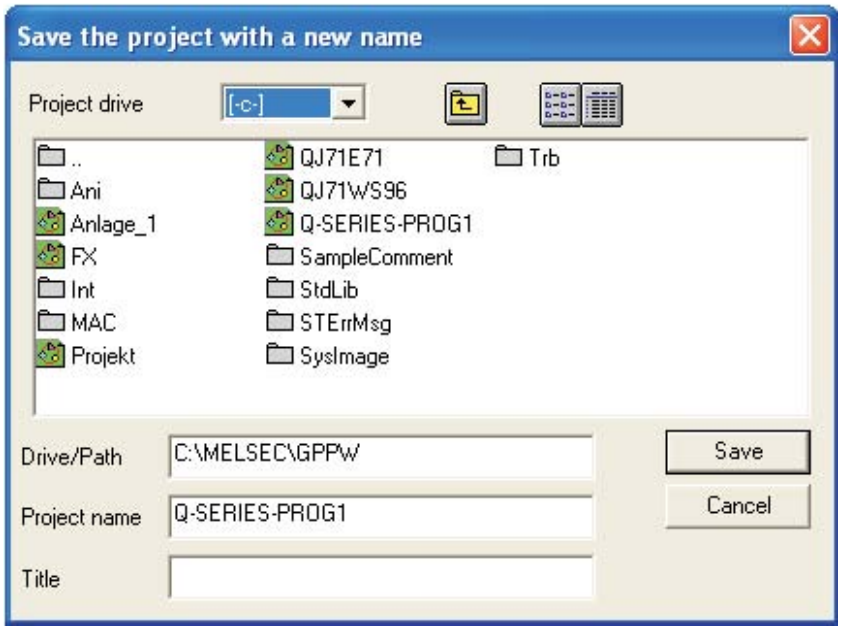
Hence prior to modifying the existing project Q-SERIES-PROG1, it is necessary to copy Q-SERIES-PROG1 to project Q-SERIES-PROG2. This is done as follows:

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

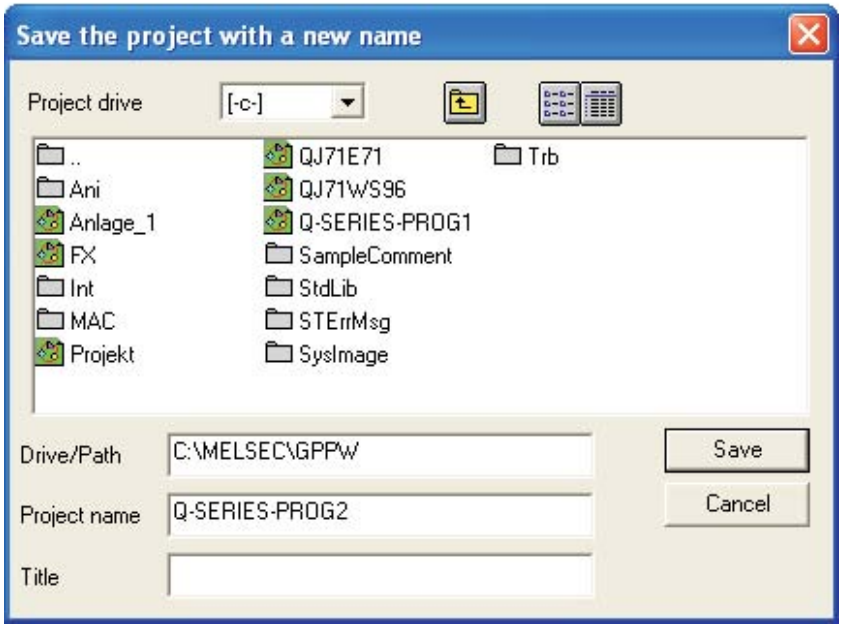
② Select ***Save as***.



③ The display now becomes as shown below:



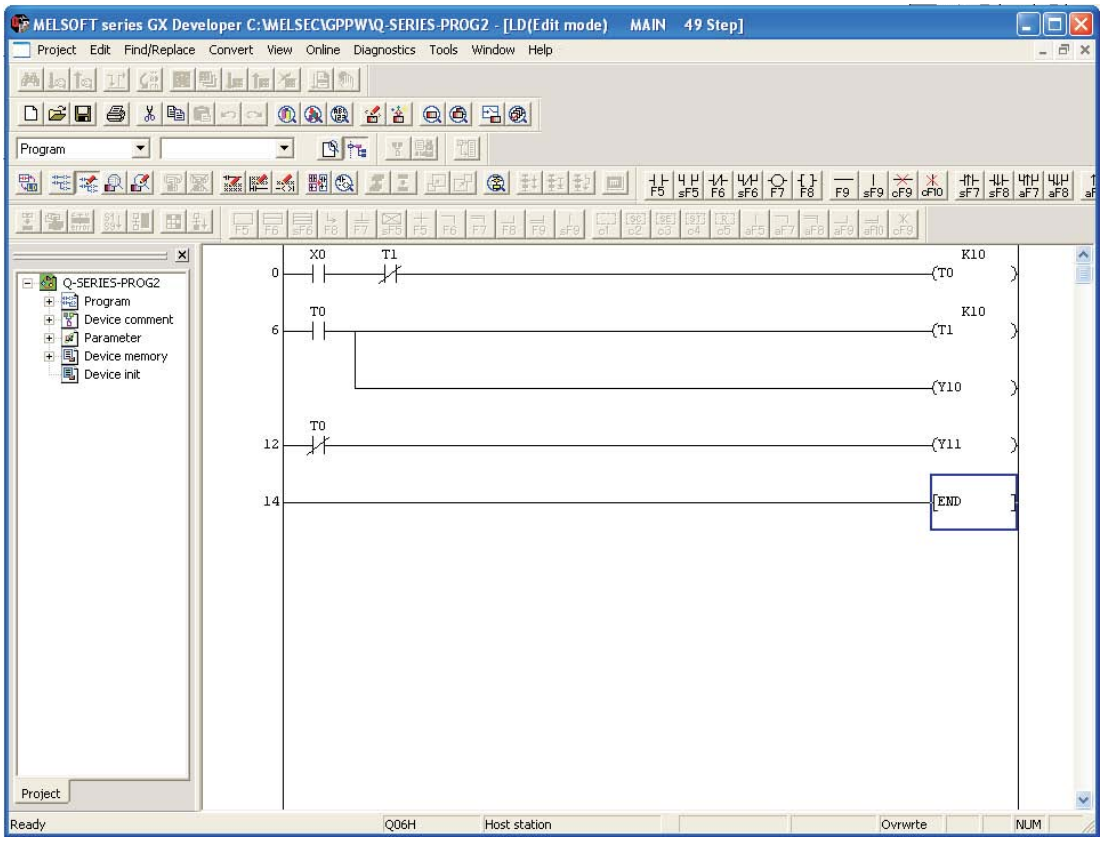
④ Change the ***Project name*** to Q-SERIES-PROG2.



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

⑥ Select ***Yes***, to create the new Project Q-SERIES-PROG2.

⑦ The display now appears as shown below.



NOTE

The Project name has changed to Q-SERIES-PROG2 (see the top Program Information bar). The project Q-SERIES-PROG1 can still be recalled, whenever required.

8 Modification of Ladder Diagrams

8.1 Modification of the project Q-SERIES-PROG2

Before any modifications can be carried out, it is necessary for the Ladder Diagram Q-SERIES-PROG2 to be displayed on the screen.

At the moment Q-SERIES-PROG2 is identical to Q-SERIES-PROG1.

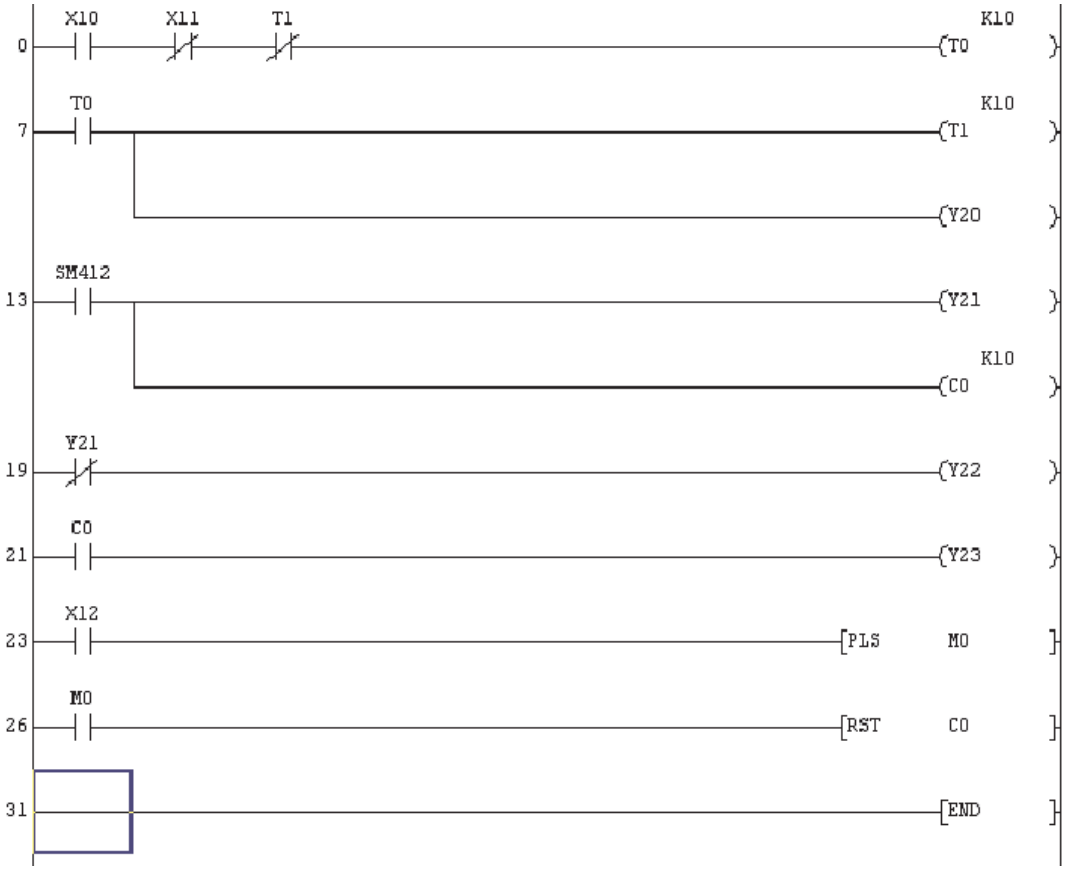
Modification Details

As can be seen from the following altered diagram Q-SERIES-PROG2, the modifications consist of:

- Line 0: The insertion of a normally closed input X11.
- Line 12: Modification of normally open contact from T0 to SM412 *
  Insertion of an additional rung: Output Coil C0 K10
- Insertion of an additional rung: normally open contact of C0 driving output coil Y23
- The insertion of an additional rung: normally open X12 driving a Pulse [PLS M0] instruction.
- The insertion of an additional rung: normally open M0 driving a Reset [RST C0] instruction.

* SM412 is Q-Series special M Relay is one of a number of special devices and equivalent to M9032 in A-Series. SM412 (M9032) 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 Appendix A, for full description and Q-A cross reference list of Special Relays.

Modified Ladder Diagram Q-SERIES-PROG2



8.2 Insertion of a new contact

To insert the normally closed contact X11, between X10 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.

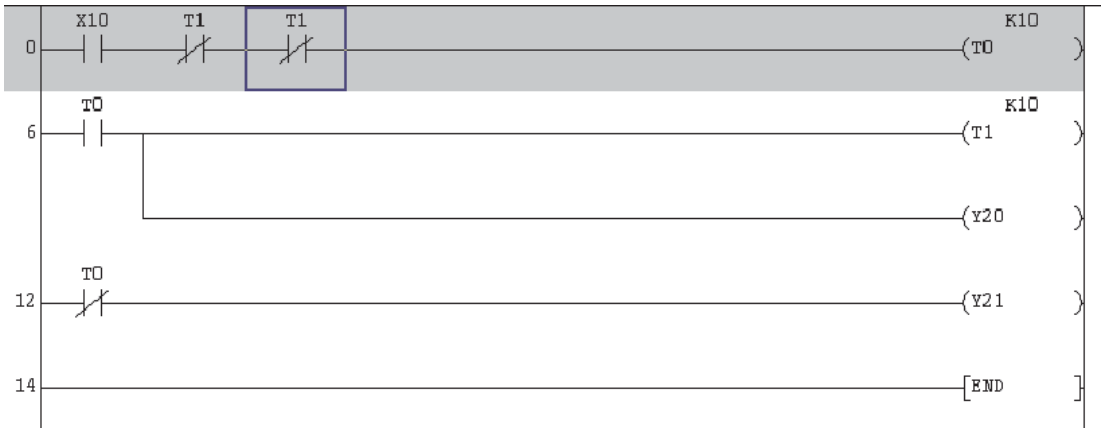
– The word ***Insert*** now appears in the bottom right hand corner of the VDU display:

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 2 for a normally closed contact.

④ Enter the contact name X11 <Enter>.

⑤ Line 0 will now include the normally closed contact X11.

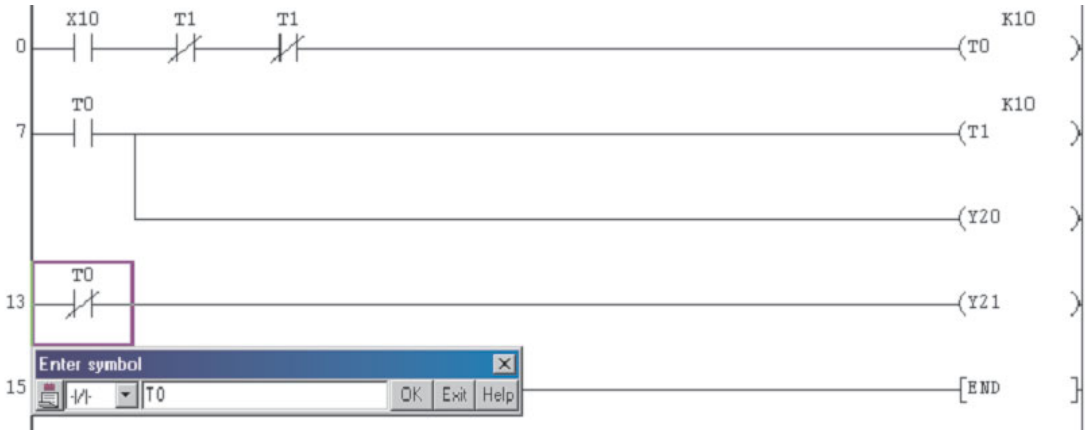


⑥ Press F4 to convert the addition of the normally closed X11.

8.3 Change of Device Detail

① Press the “Insert” button on the keyboard and note the change of mode back to “Overwrite” Ovrwrte (cursor colour changes to blue).

② Move the cursor over to normally closed contact of T0 on line 13. Double click the mouse or press <Enter> and the following will be displayed.



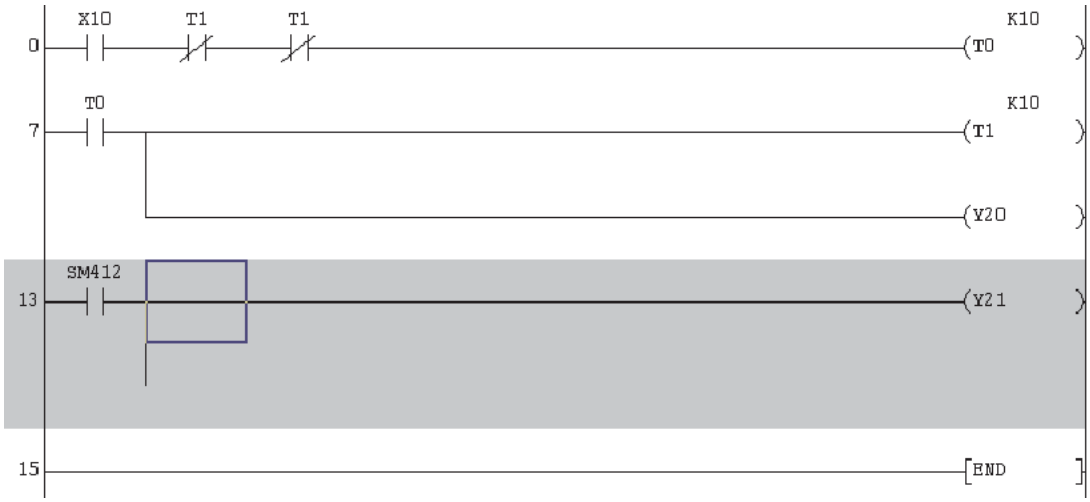
③ Click the small downward triangle in the left of the symbol box and select a normally open contact.

④ Alter T0 to SM412 and press OK. Press F4 or the  buttons to convert the changes and the display will be as follows:

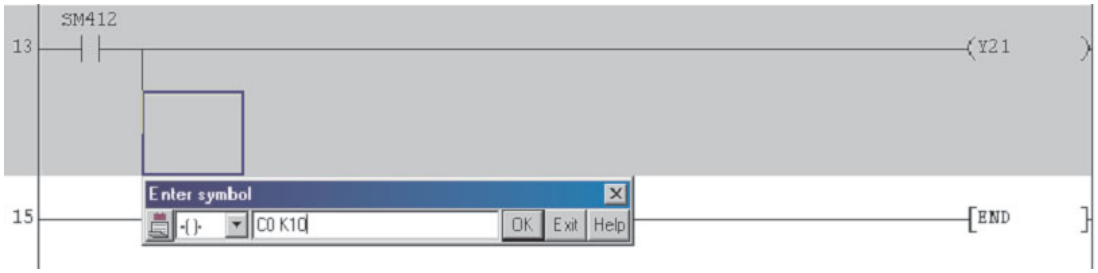
8.4 Inserting a Branch

① To insert the output C0 K10 as a branch to line 13, go to insert mode. The cursor turns red to indicate the change mode.

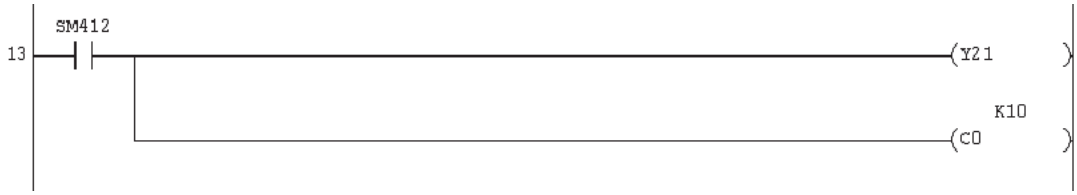
Press the  “branch down” button or 5 on the keyboard and press <Enter>. The display will be presented a follows:



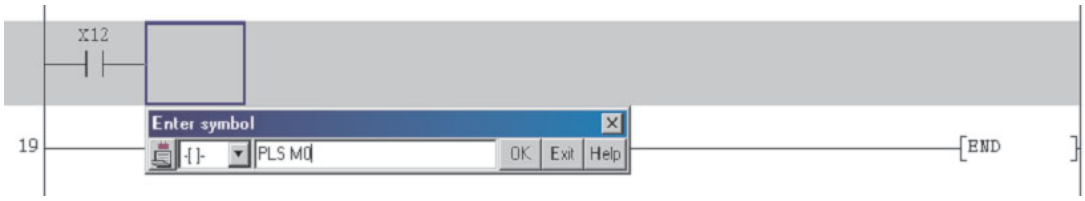
② Move the cursor down one line and press the  “Output coil” button or press 7 on the keyboard. Enter C0 K10 and the display will be as follows:



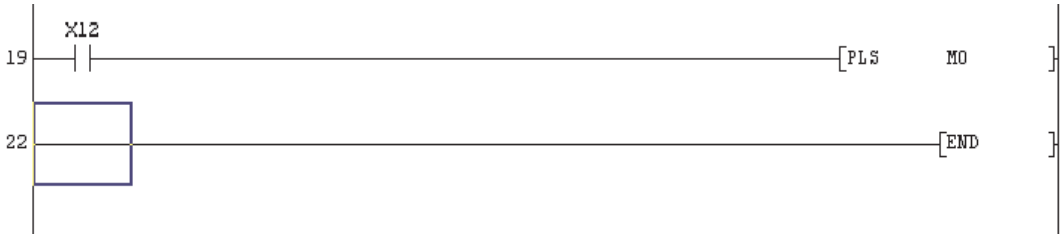
③ Press <Enter> to enter the coil and then press F4 or the  buttons to convert and the display will become thus:



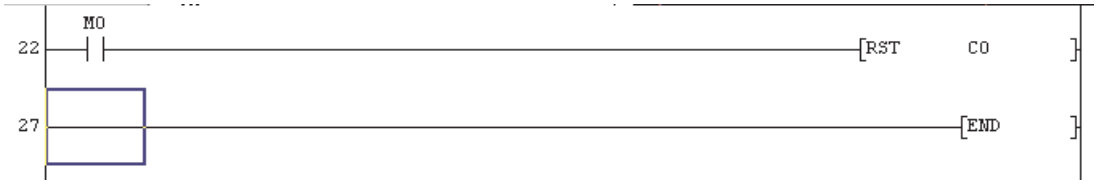
④ With the cursor positioned on the start of line 19, select a normally open contact of X12. To enter the PLS M0 instruction, select 8 -[ ]- from the toolbar and enter PLS M0. The display will be as follows:



⑤ Click ***OK*** or press <Enter> to complete the line. Press F4 or the  buttons to convert and the display will be thus:

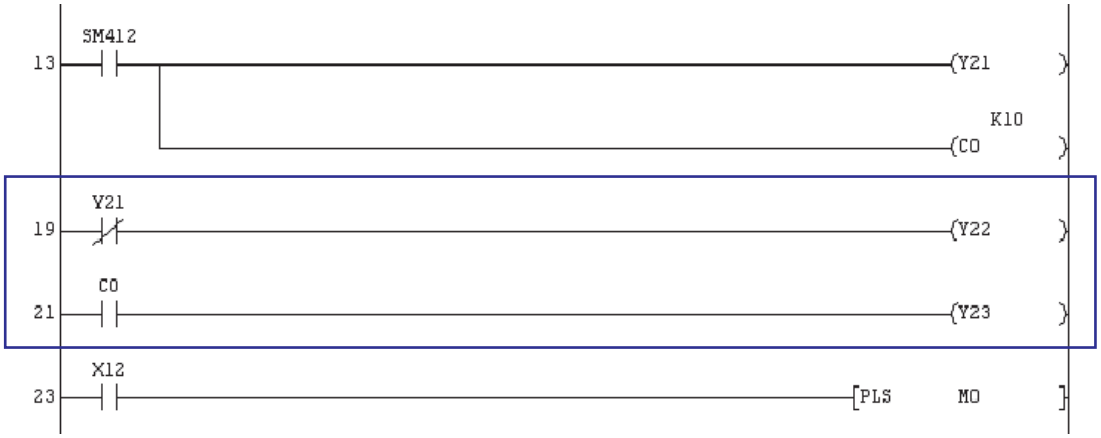


⑥ Repeat the procedure ⑤ above for the next line and the display will be as shown:

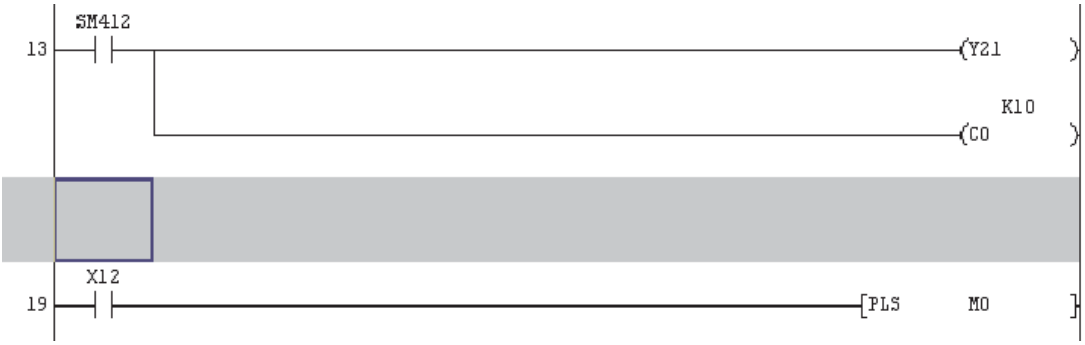


8.5 Insertion of New Program Blocks

The following two (blue framed) further lines will be inserted following line13.



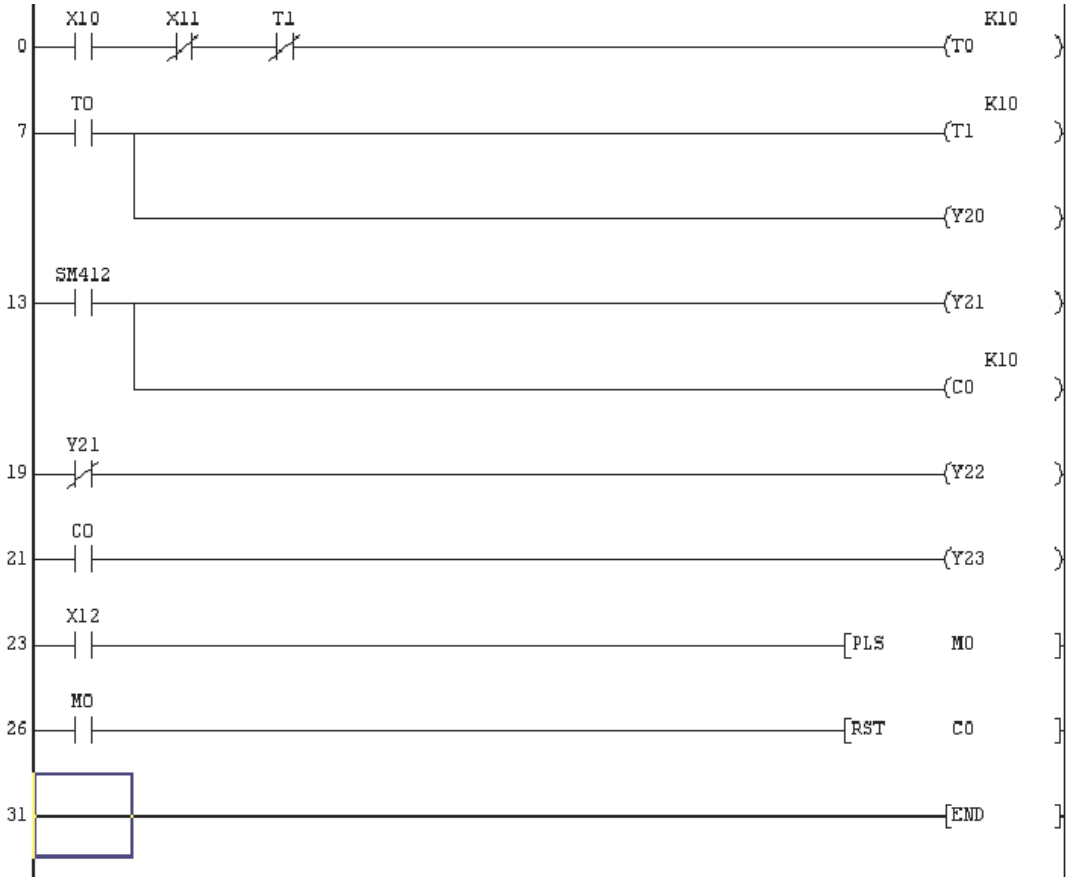
① With the cursor on the start of Line 19, select ***Insert Line*** from the ***Edit*** menu thus:



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 then press F4 or the  buttons.

The final modified Ladder Diagram Q-SERIES-PROG2 will now be as shown on the next page.



④ Save Q-SERIES-PROG2 using the button or Select ***Save*** from the ***Project*** Menu.

9 Delete Functions

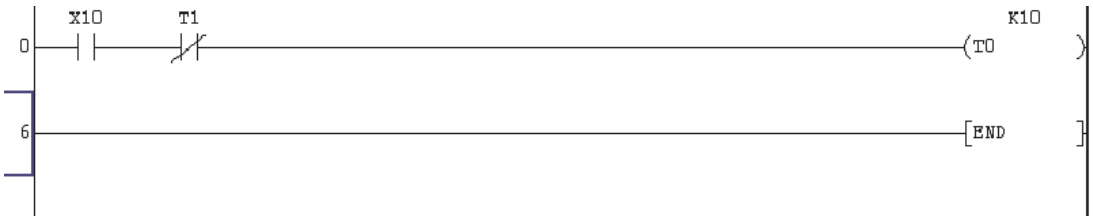
9.1 Overview

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

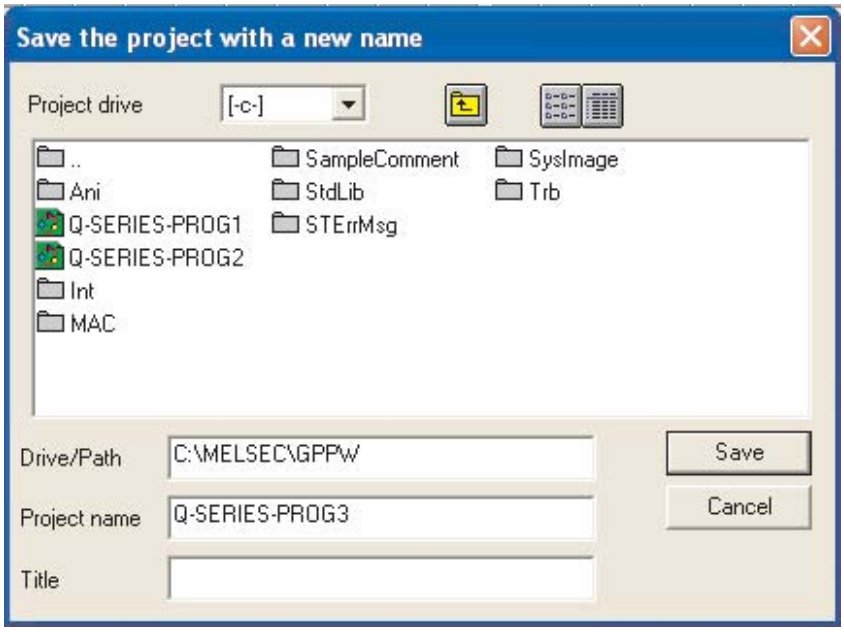
The project Q-SERIES-PROG3 will be used to demonstrate how the following can be deleted:

- 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, Q-SERIES-PROG3 will appear as shown below:



Before carrying out further modifications, save Q-SERIES-PROG2 to Q-SERIES-PROG3, using the ***Save as*** procedure described previously:



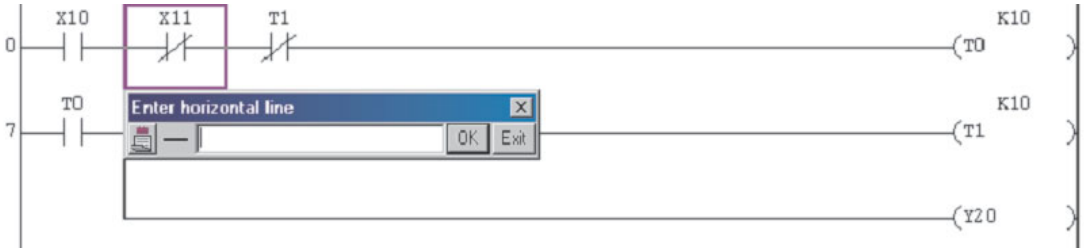
9.2 Deleting an Input Contact

Ensure the project Q-SERIES-PROG3* is displayed and in Overwrite mode

* NB: At this moment in time Q-SERIES-PROG3 will be identical to Q-SERIES-PROG2.

① Move the cursor to the Normally Closed X1 contact.

② Select the horizontal line i.e. key 6 to delete the X1 contact



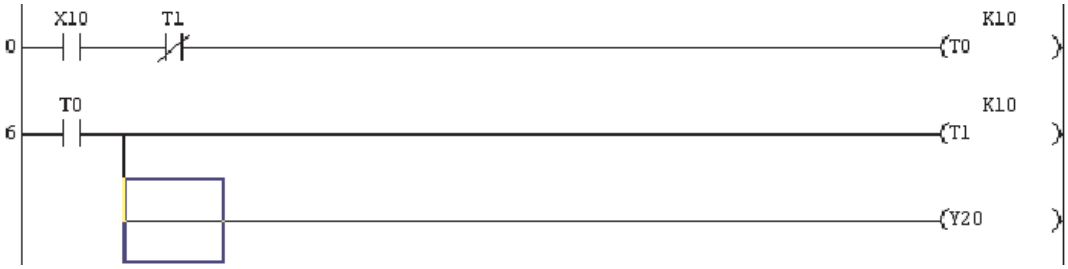
③ Select ***OK*** and the X11 contact will be deleted.

④ Press F4 or the  buttons to convert the modification, the display will be:

9.3 Deleting a Branch

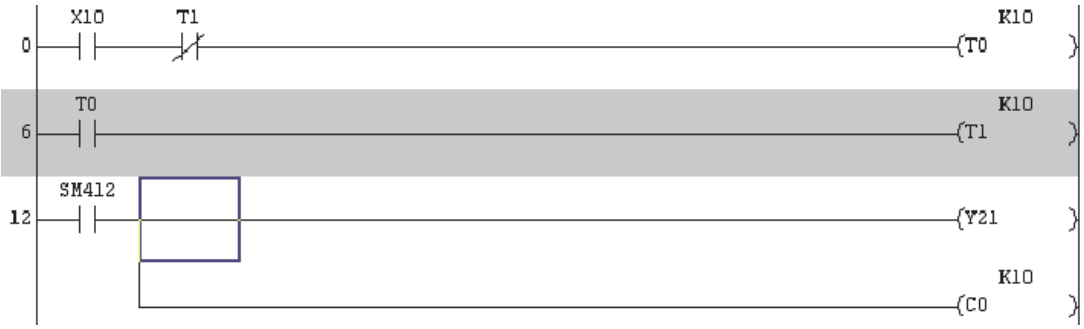
The branch at current Line 6 will now be deleted.

① Move the cursor to the branch at Line 6 as shown below.

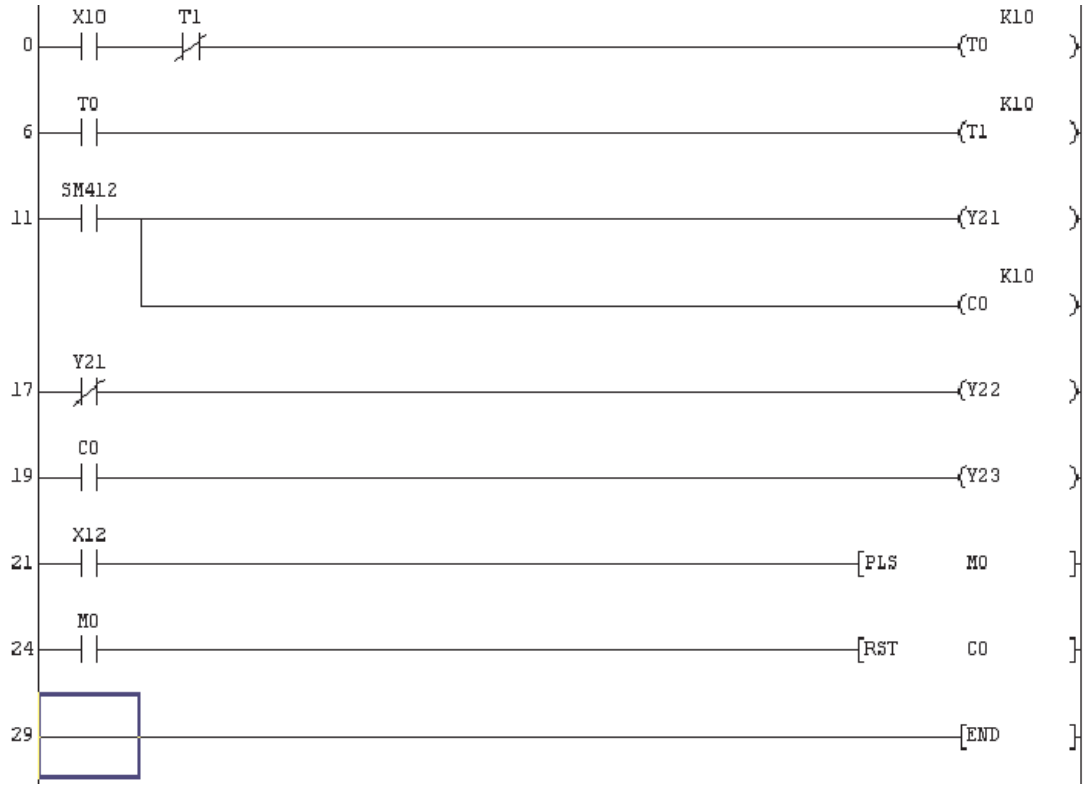


② From the ***Edit*** menu, select ***Delete Line*** or use the shortcut keys "Shift+Delete" together.

③ The display will become:



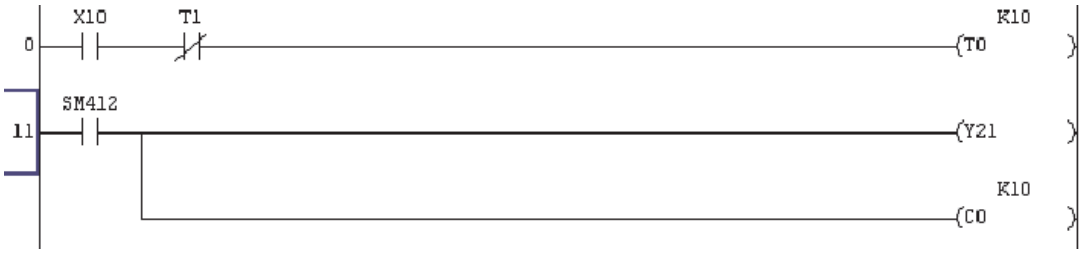
④ Press F4 or the buttons  to convert the changes:



9.4 Deleting a Single Line

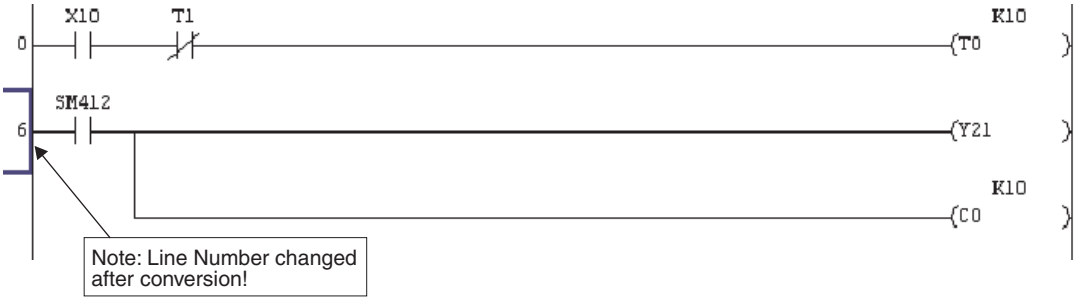
The single line at Line 6 will now be deleted.

① Move the cursor to the start of line 6 (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:



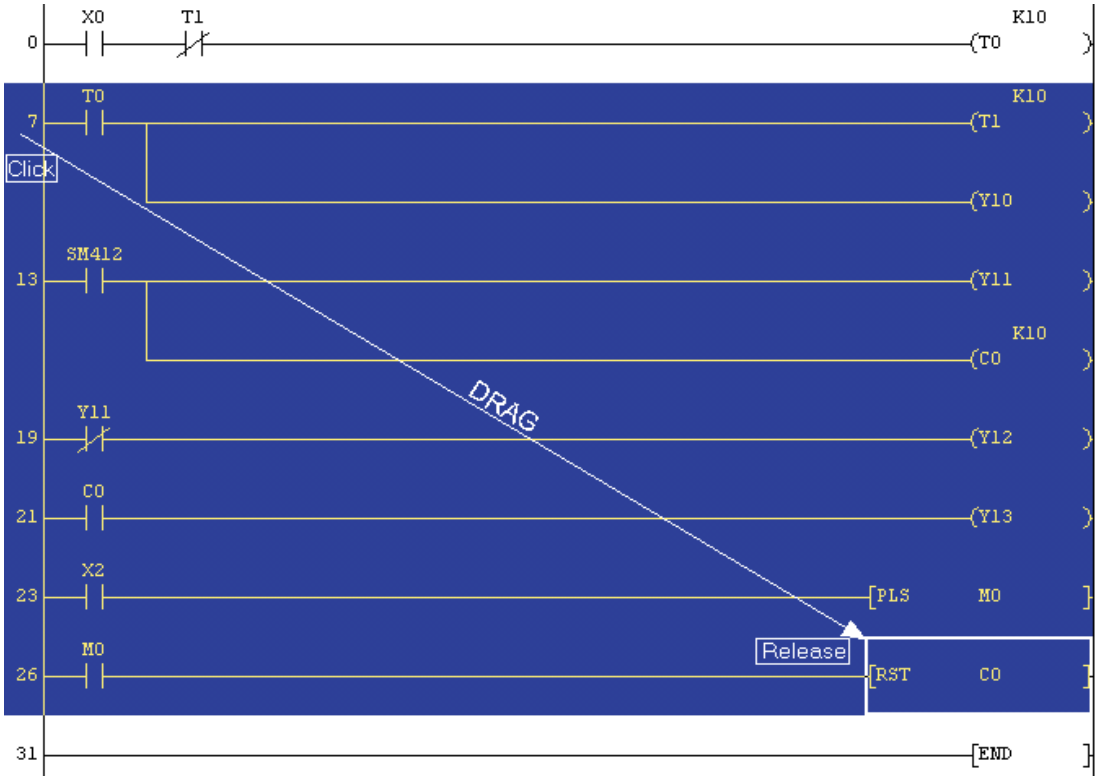
Important: You **MUST** remember to press F4 or click the buttons  to convert the changes following a **line deletion**. In this case, GX-Developer gives **no** indication that there **has been a change** to the code because the changed code has been deleted!

Once converted, note line number changes:



9.5 Deleting multiple lines

① Click and hold the left hand mouse button **to the left hand side of the rung** on line 7. 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 26. Release the mouse button as shown below:



② 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:

③ 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.

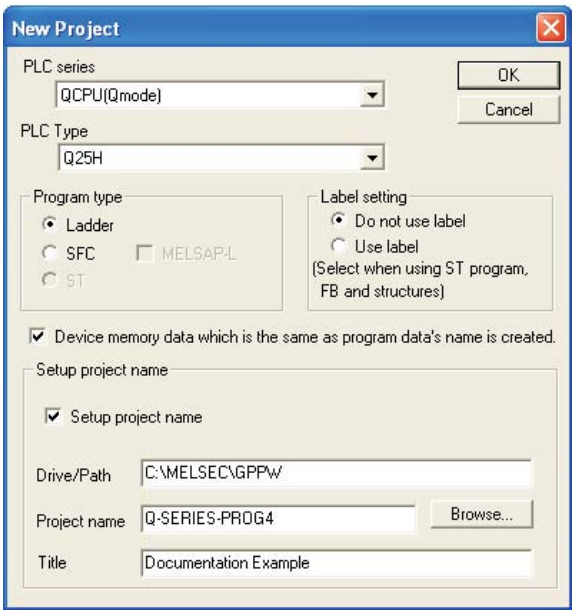
There is really no excuse for poorly documented programs; most PLC programming software provides extensive facilities for the annotation of software. Poorly documented software is totally unacceptable in any situation! Documentation is necessary in order for the program originator to convey programming 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: Q-SERIES-PROG4

A new program Q-SERIES-PROG4 will be constructed in order to demonstrate the use of the documentation and annotation tools provided in GX-Developer.

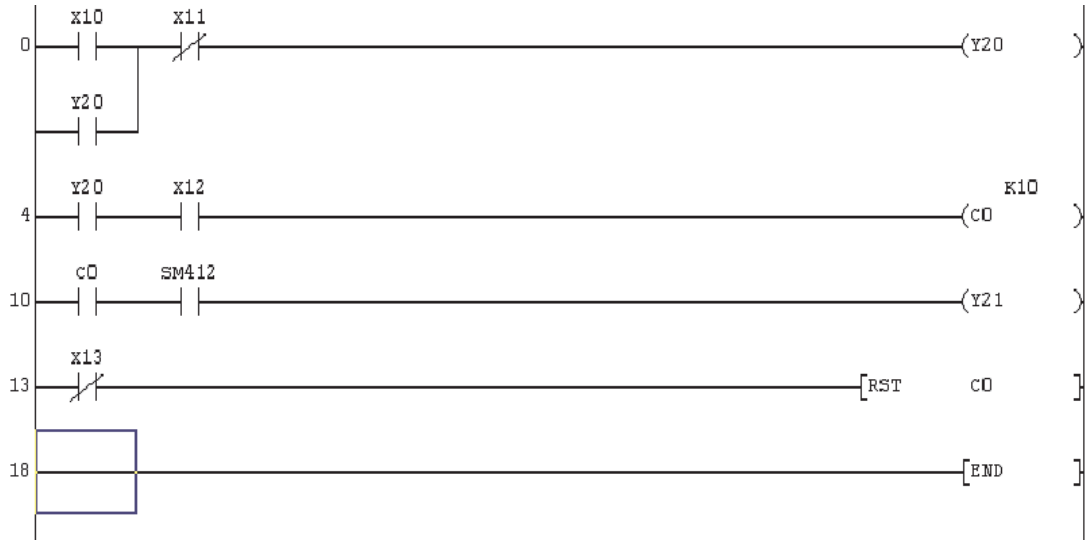
① From the ***Project*** Menu, select ***New Project*** or simply press the Key. The display will be as shown:



Note the addition of the program title "Documentation Example" in the title field.

② Now enter the next Ladder diagram using the methods described in previous sections in this courseware:

Q-SERIES-PROG4



NOTE

It is also possible to enter [function commands] directly instead of using the {} square bracket function first. Simply type the function from the keyboard and GX Developer will automatically accept the entry. This enables quicker data entry to be carried out by reducing the number of keystrokes.

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.

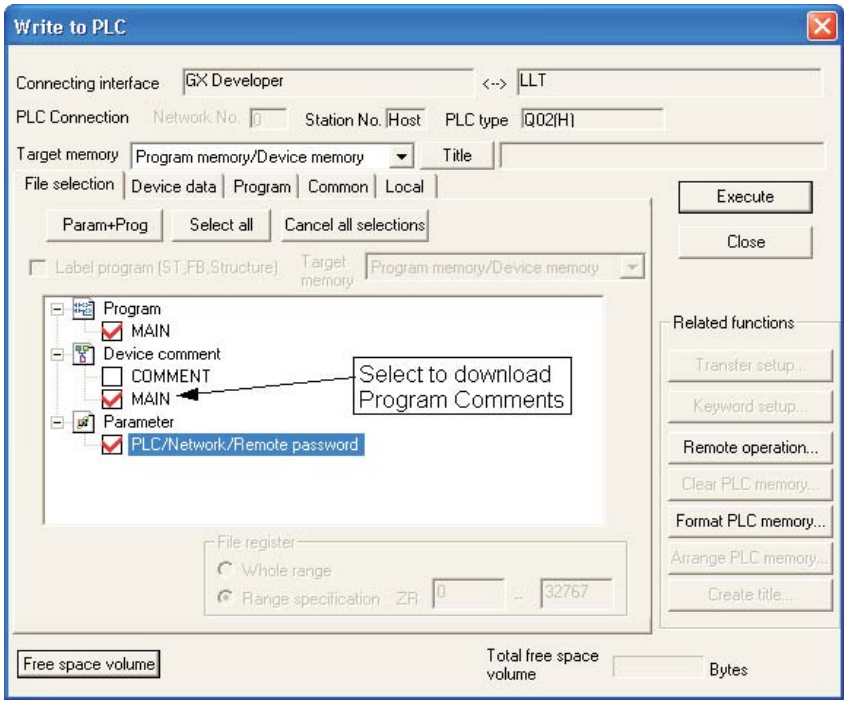
Statements/Notes

GX-Developer offers the facility to embed the program statements and Notes i.e.

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, else it is unavailable on all other PLC models in which case the option to embed line statements will be 'Greyed out'.

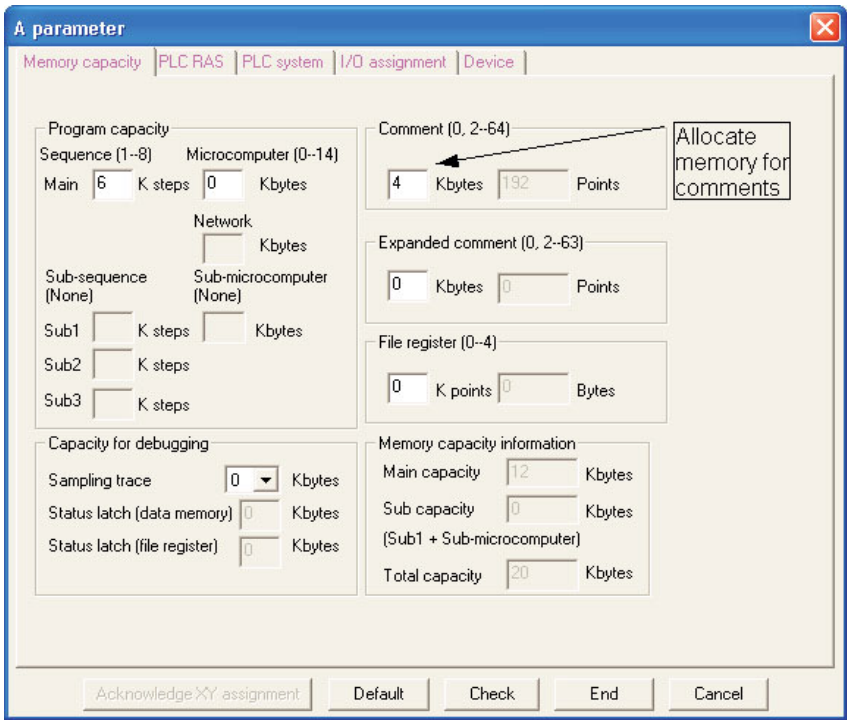
Comments

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



A and FX Series PLC's

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



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.

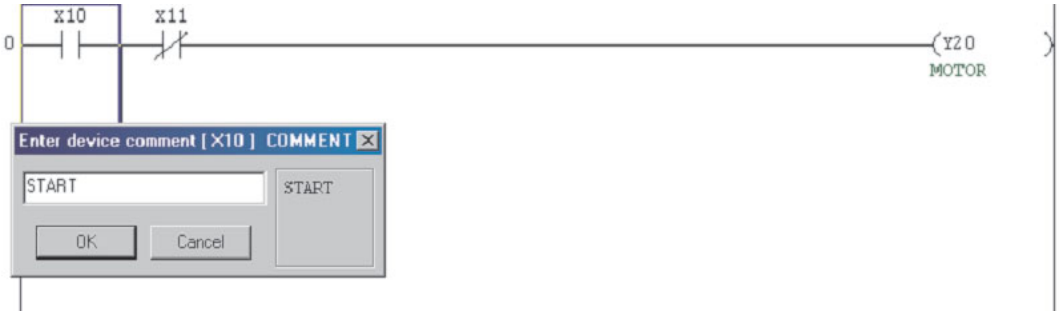
10.3 Comments

Direct onscreen method

You can enter comments direct while programming.

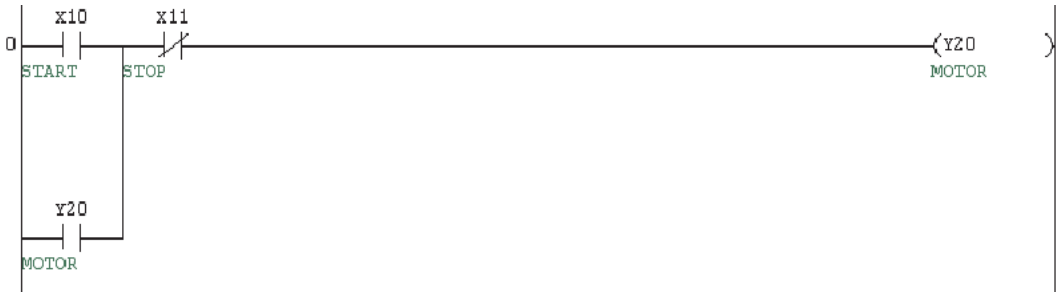
① With the program Q-SERIES-PROG4 displayed on the screen select the Comments mode button: 

For example, to place a comment against the device X10, position the cursor over the X10 contact and press ‘enter’ or double click the mouse over the contact. The following screen is displayed:



② Enter the comment “START” in the text box and press enter or click ***OK***.

③ Move the cursor to X11 and press <enter> or double click the mouse over X11. Repeat for the output Y20 and enter the comments as shown below:



Note that all occurrences of the devices X10, X11 & Y20 will be displayed automatically within the program with the attached comment.

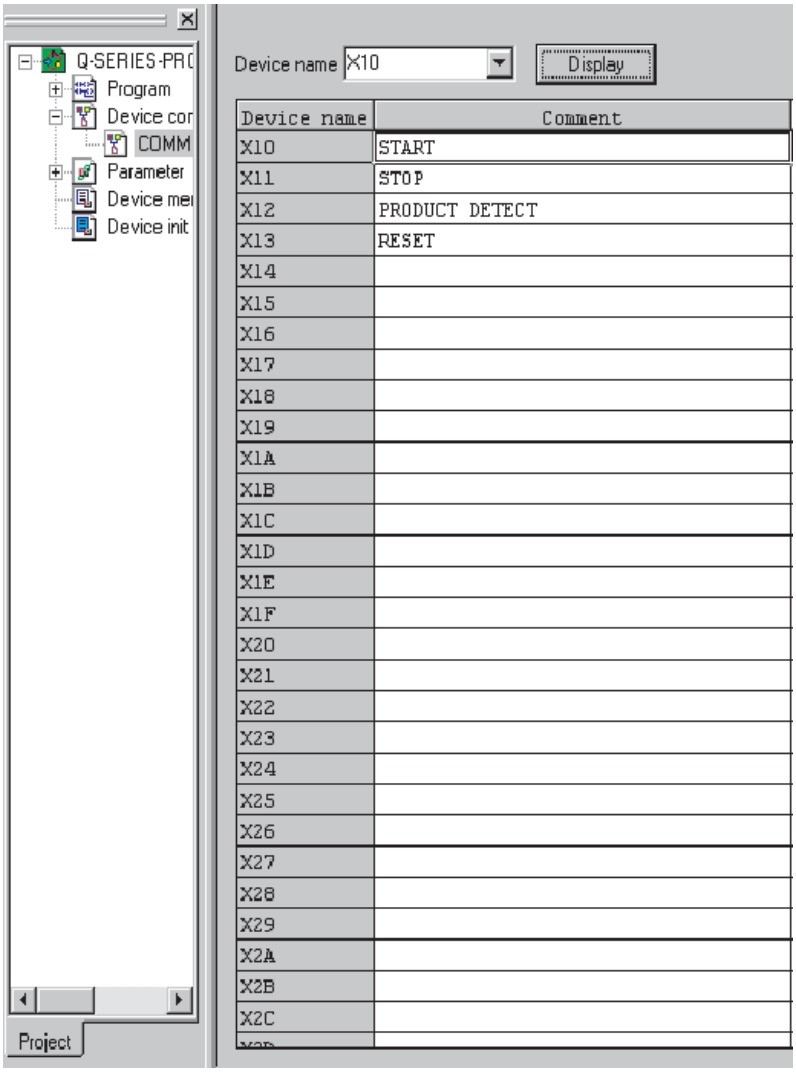
10.4 Project Data List (Navigation Window)

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.

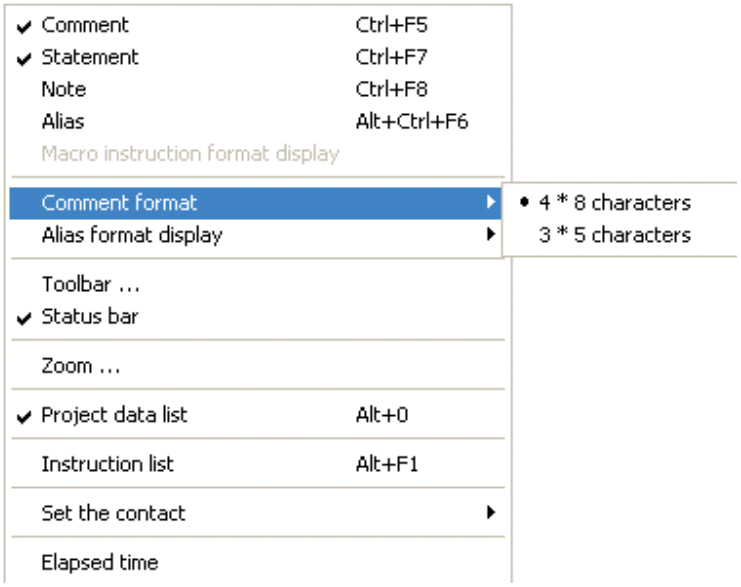
To enter comments into table, double click on the ***Device comment*** folder in the ***Project Data List*** window:



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:

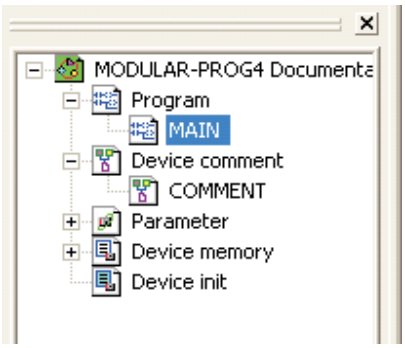


The default format is 4 lines of 8 characters which can be altered using the above menu and advanced system settings which are described later-on and in the advanced course notes.

NOTE

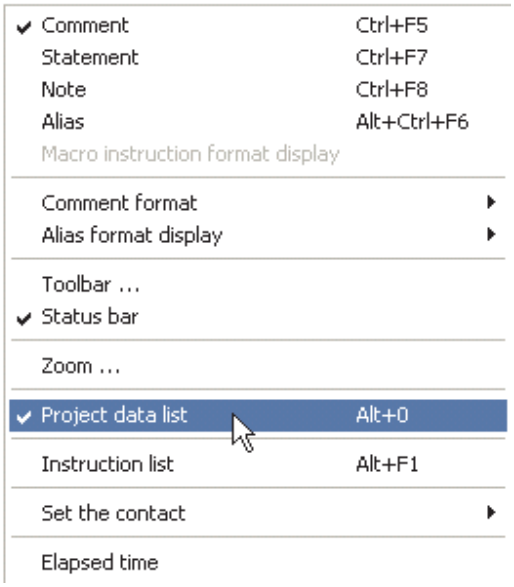
When entering the text in the prompt window, care should be taken to ensure that appropriate padding (spacing) be manually placed in the string so that the comment is displayed correctly. Remember GX-Developer automatically "Word Wraps" the text to the preset format.

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:

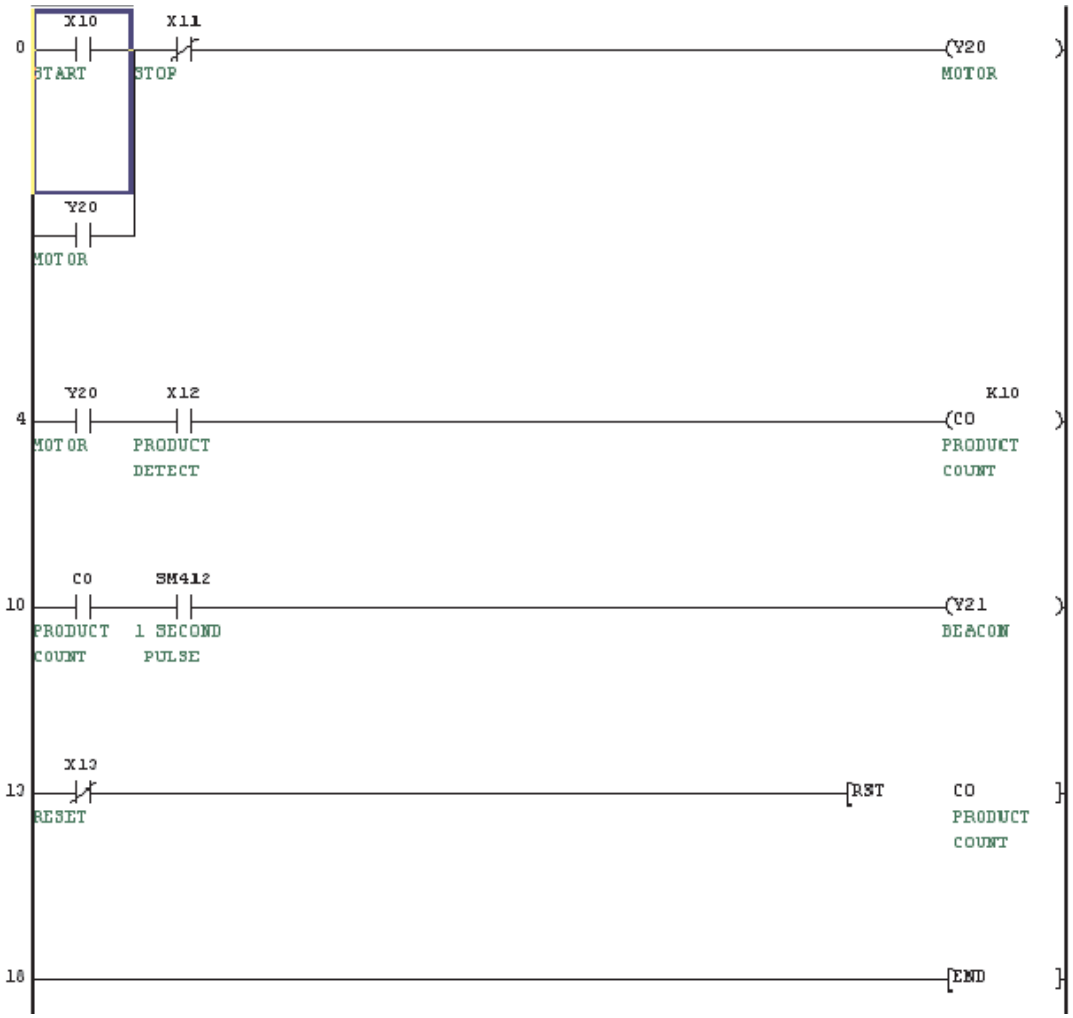


It is convenient to use this Project Data List window in-future to manoeuvre around between displays and editors.

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.



Complete the commenting of the ladder program as follows:



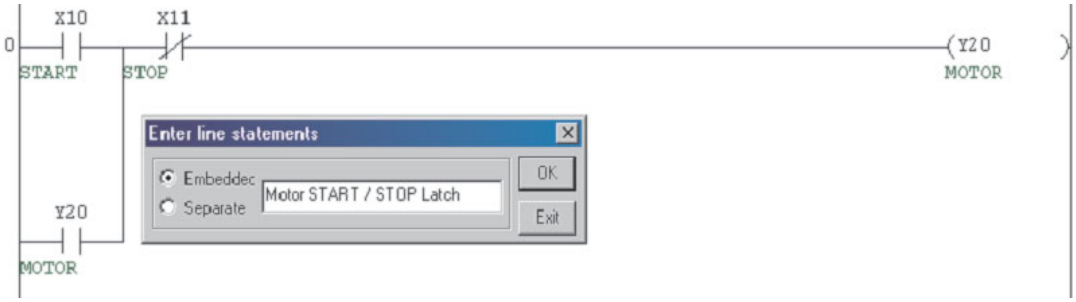
10.6 Statements

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 Q-SERIES-PROG4 displayed on the screen select the Statement mode button: 

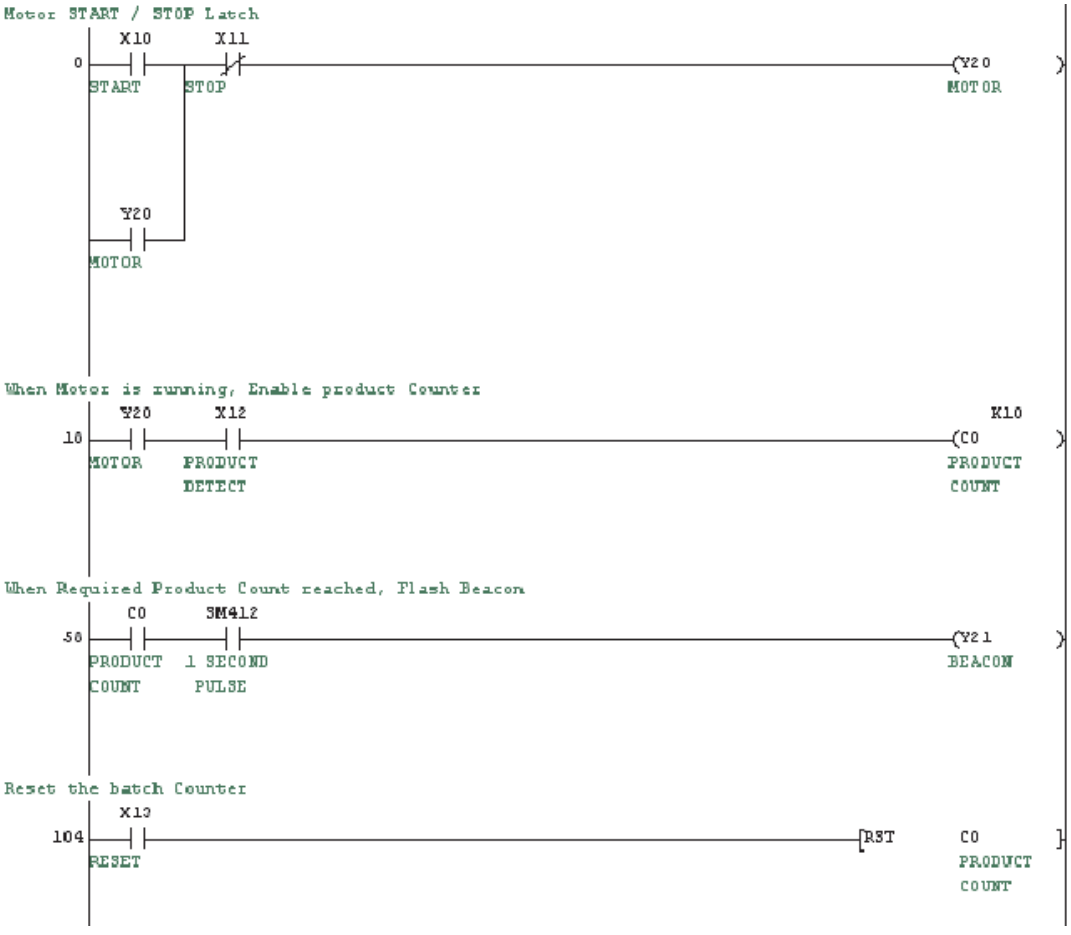
② Position the cursor anywhere on the program block (segment) to which the statement is to be attached. Press ‘enter’ or double click the mouse over the program block.

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:



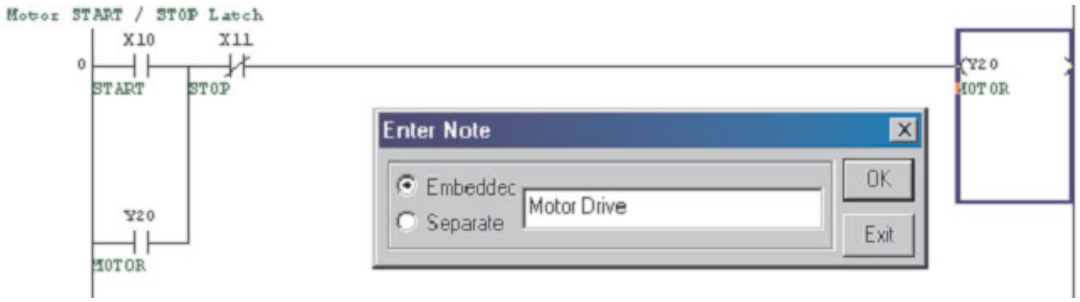
10.7 Notes

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.

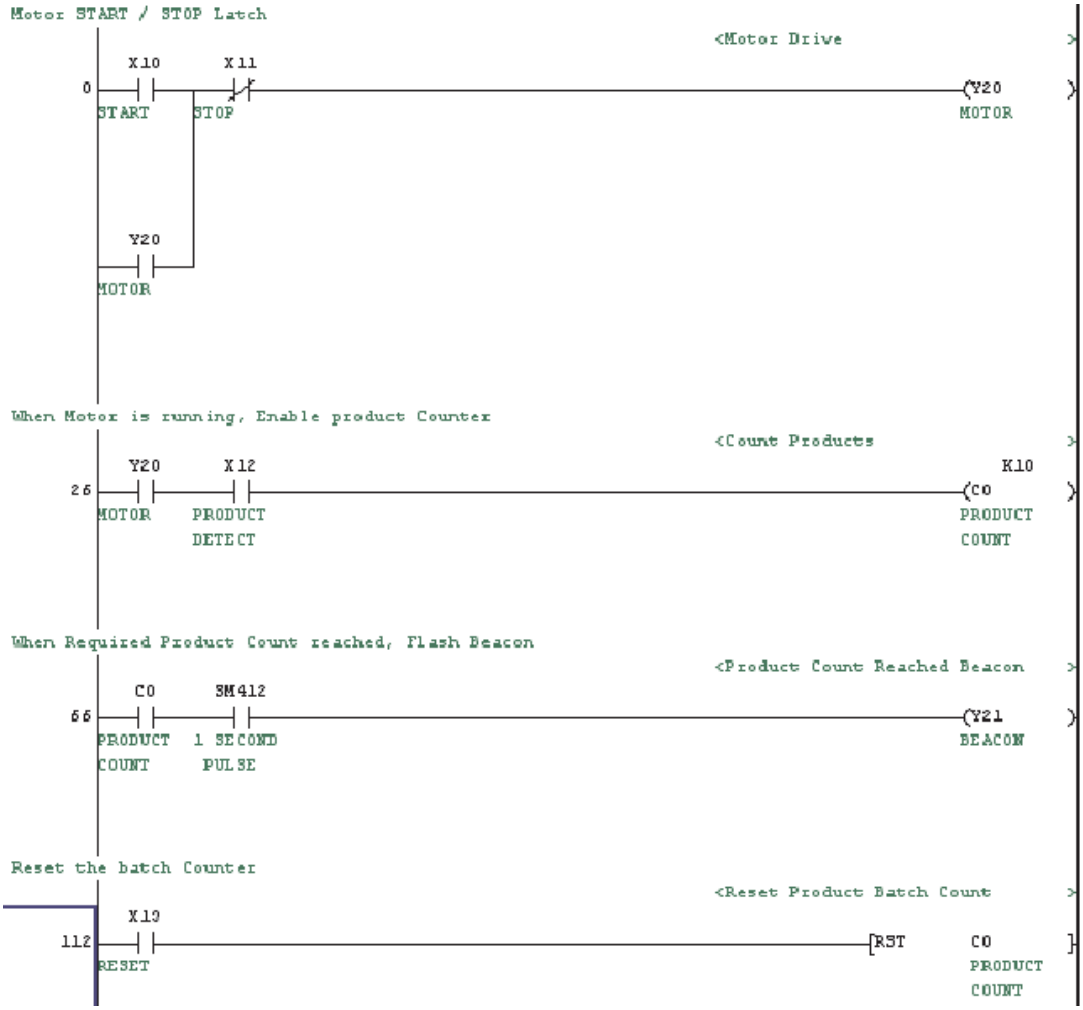
① With the program Q-SERIES-PROG4 displayed on the screen select the Note Entry mode button: 

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

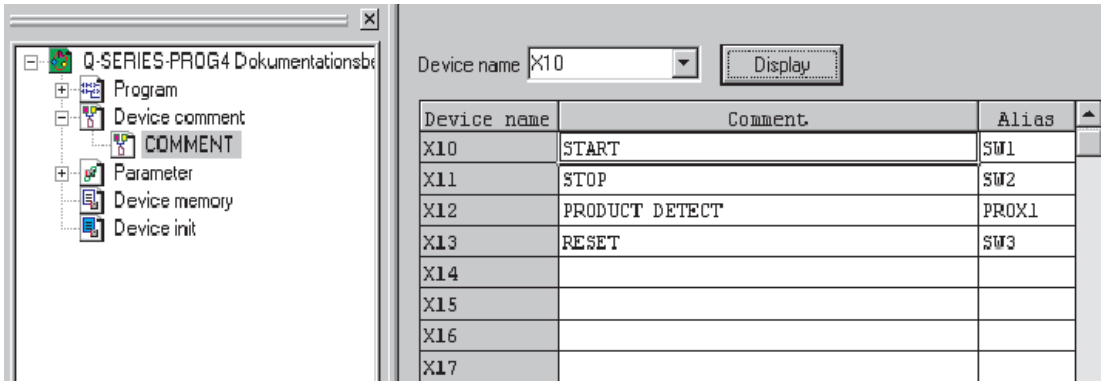
An Alias provides a method of cross-referencing PLC I/O numbers that are connected physically to external system devices. For example; Input X10 may be connected to a Start Button on a machine, whose external circuit diagram device reference is SW1. SW1 may be listed as the Alias to X10 in the Comment list, so as to provide a qualified link with the PLC program listing:

Example:

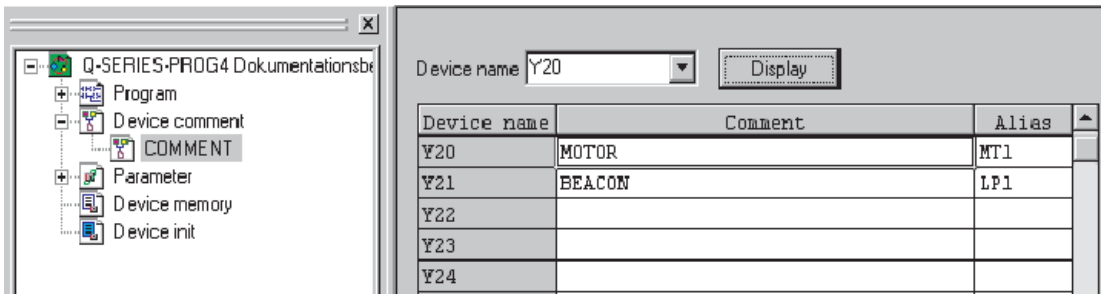
① From the Project data list window, open the Comment list.

② Double click the mouse over the Alias column for X10 and enter the text “SW1”.

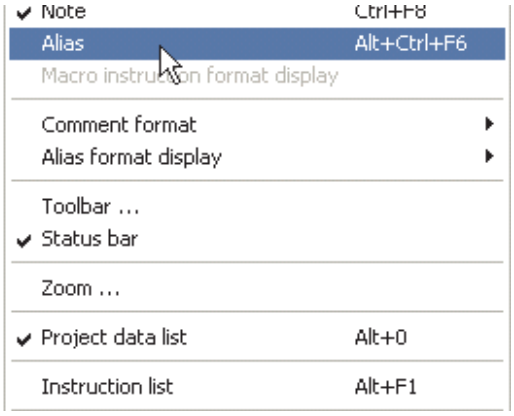
③ Repeat this action for the remainder of X devices as shown:



④ Change the display reference to show devices beginning with Y20. Repeat the action in 4 above for Y20 & Y21 with data as shown:

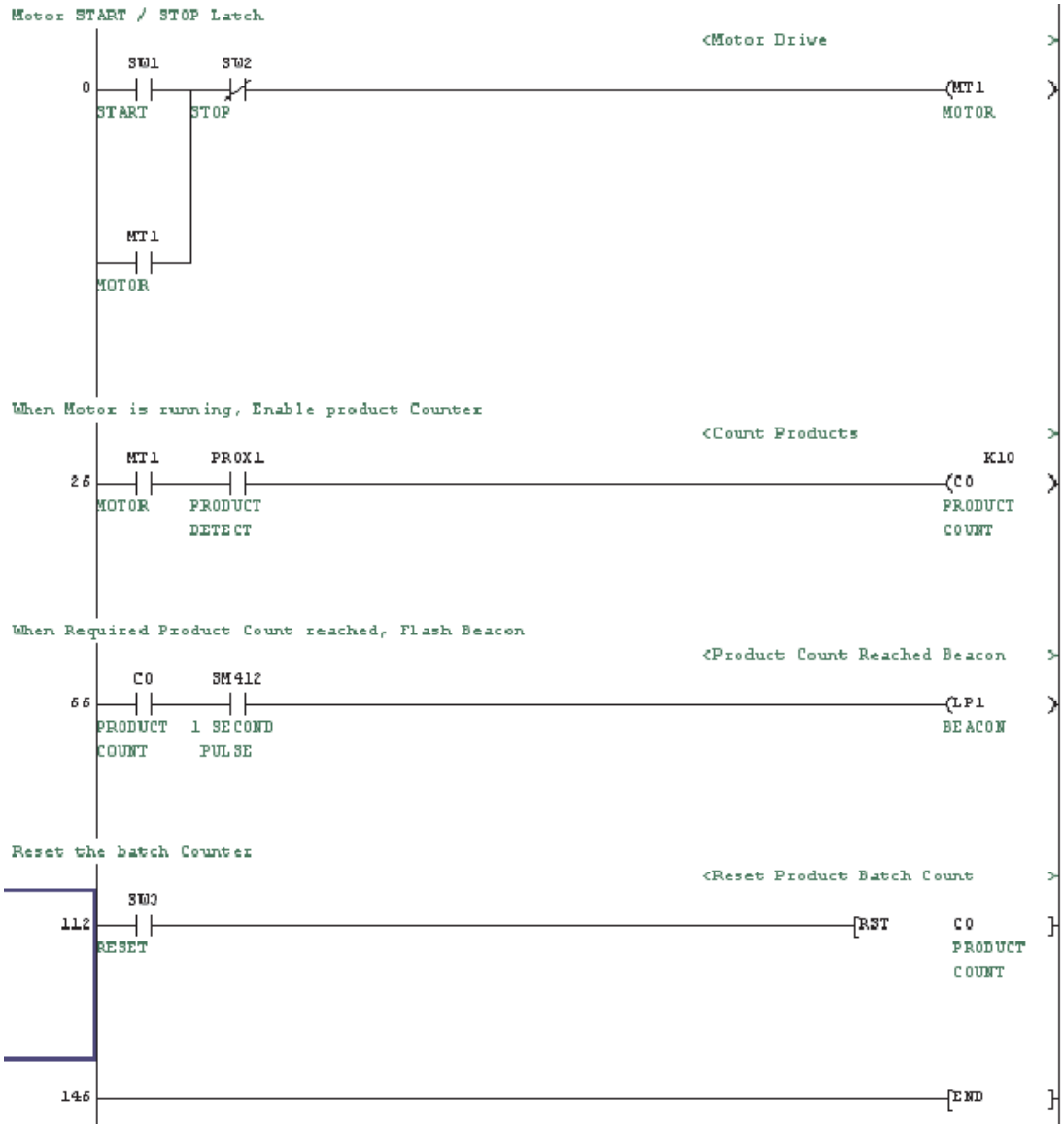


⑤ In the Project Data List Window, Click ***Program*** and ***Main*** to return to the Ladder Display.



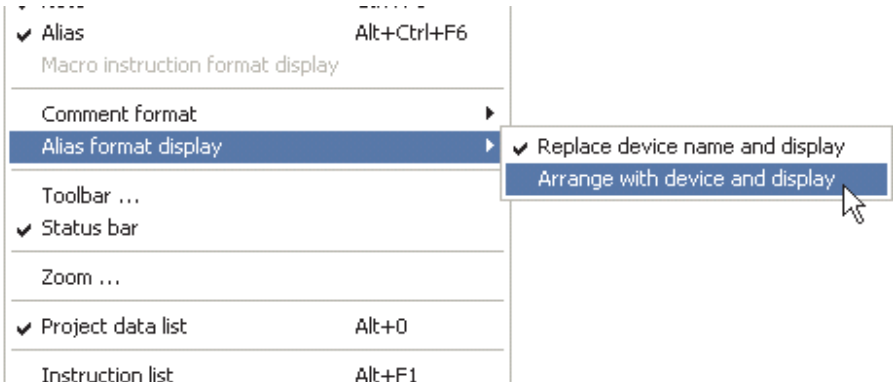
⑥ From the ***View*** menu, click ***Alias***.

The display will be as shown

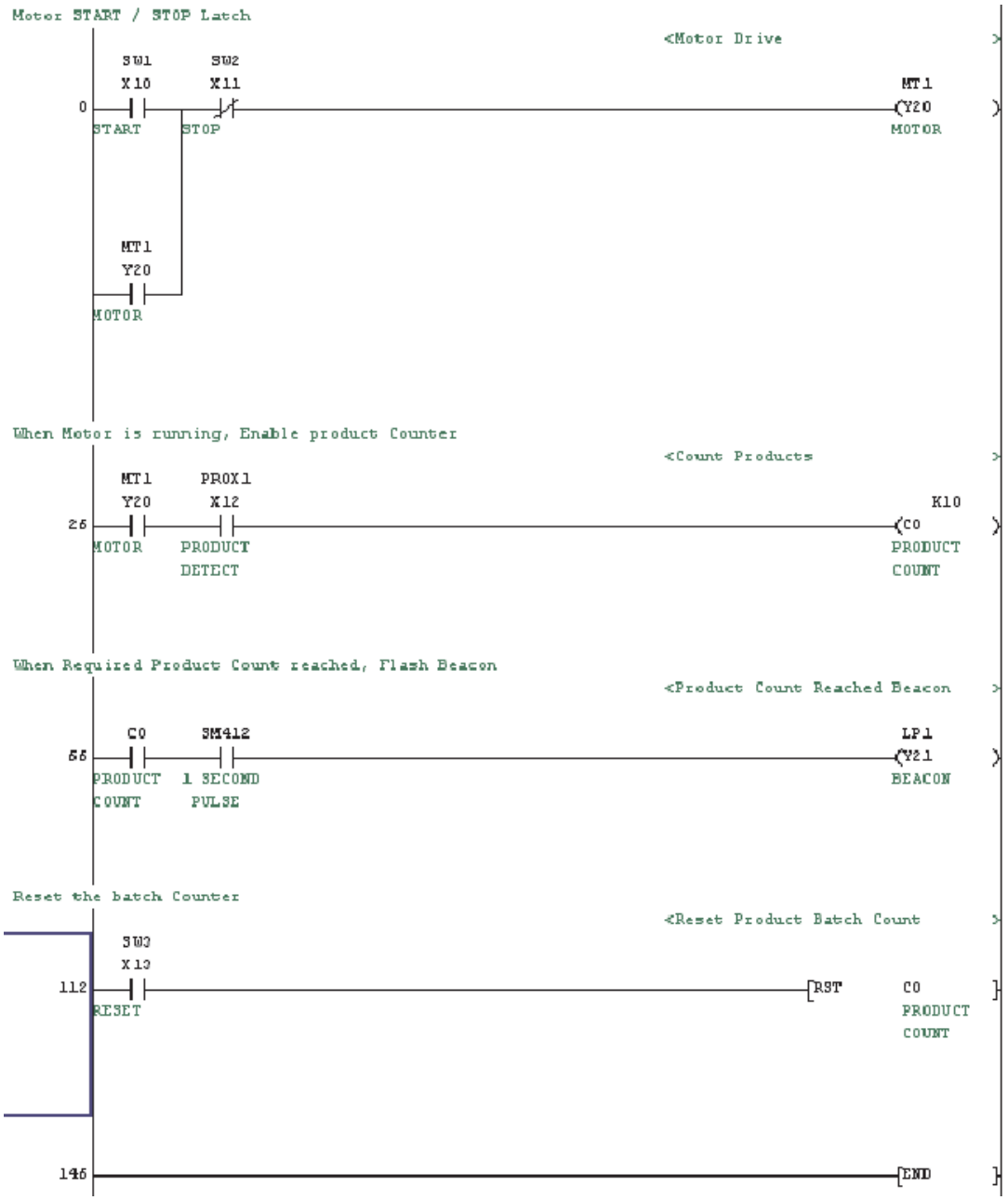


As can be seen, the display has replaced the Device Names with the Alias where appropriate.

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:

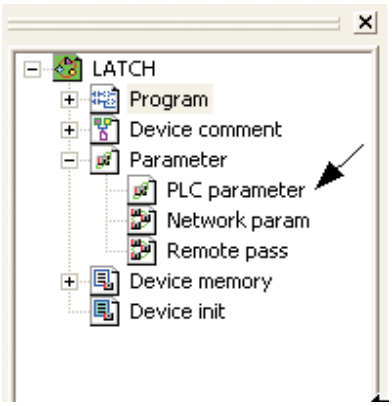


11 I/O Assignment

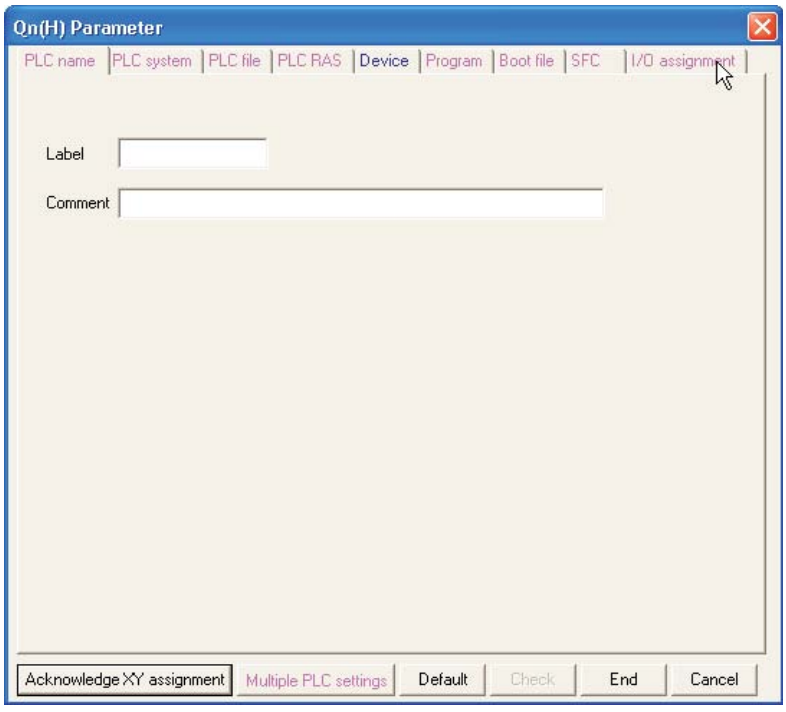
11.1 I/O Assignment for the Q-Series

Before a program can be sent to the PLC CPU, an assignment list of the Inputs and Output module configuration should be built. This is required in order that the CPU knows the correct combination of modules in the rack(s) so that it can interface with each module in the correct manner.

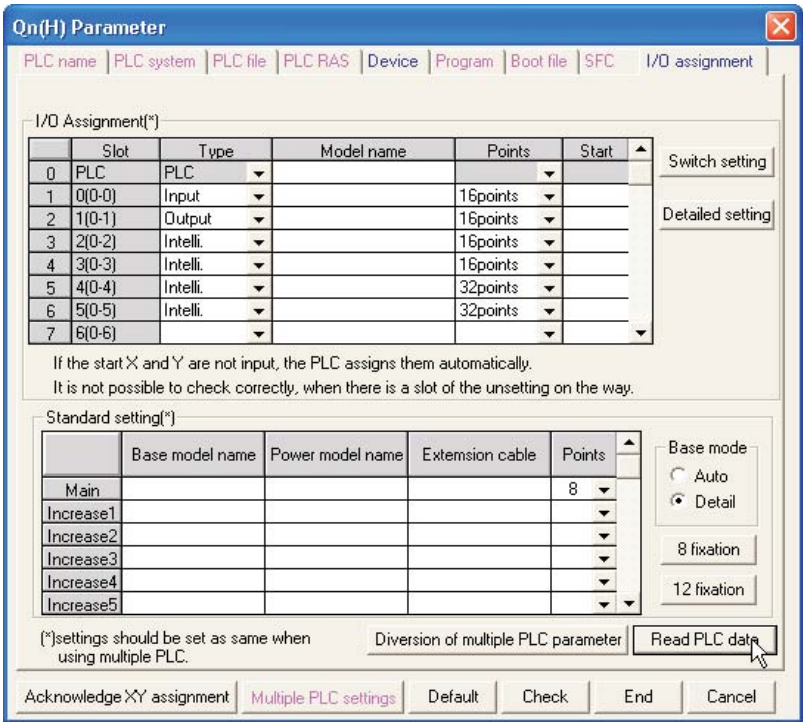
① From the ***Project Data List*** window, open the ***Parameter*** option by double clicking on the folder and file icons:



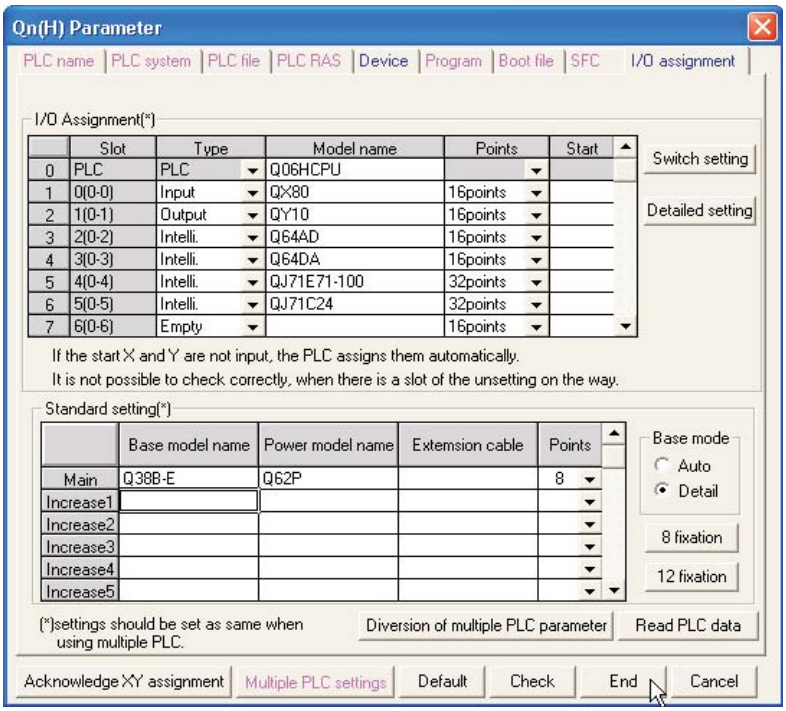
② Once open, click on the 'I/O Assignment' Tab at the top of the Window:



③ Click on the ***Read PLC Data*** button. This causes the PLC module configuration information to be read back to the screen:



④ Now type in the names of the modules in the rack into the table by examining the module information on the front of each unit:



12 Serial Transfer of Programs

12.1 Downloading a Project to a PLC Unit

The following notes describe how the project Q-SERIES-PROG4 is downloaded to a Q-SERIES PLC.

There are a variety of different methods of connecting GX-Developer to a Q-Series PLC:

- A-Series / QnA PLC Programming port

  The SC 09 converter is used, to convert the RS232 common mode serial signals 'to and from' the computer to the RS 422 serial-differential format required by the PLC.

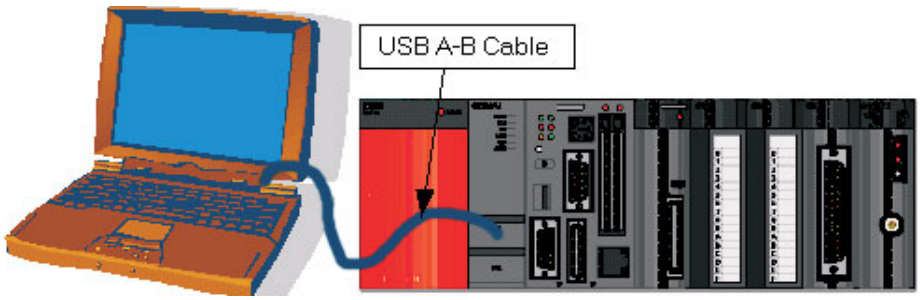
- Q-Series PLC Programming port

  RS232 using special programming cable.

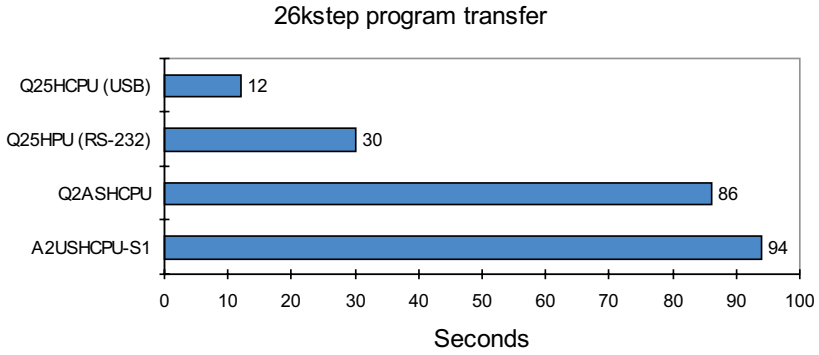
- Q-Series PLC Programming Interface

  USB - Preferred: Standard USB A-B communications cable.

For the Mitsubishi Training Rigs, connect the computer to the Q PLC as shown in the diagram below:

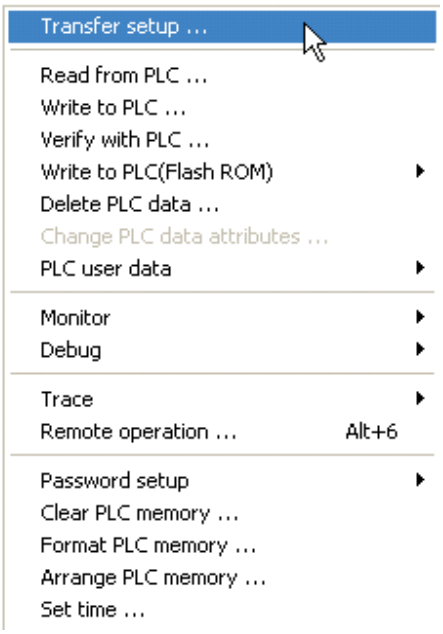


The Table below illustrates the comparison of program transfer times between fastest A-Series CPU with QnA and Q-Series Processors. Note the significant speed of Q- Series increase over A-Series PLC's:

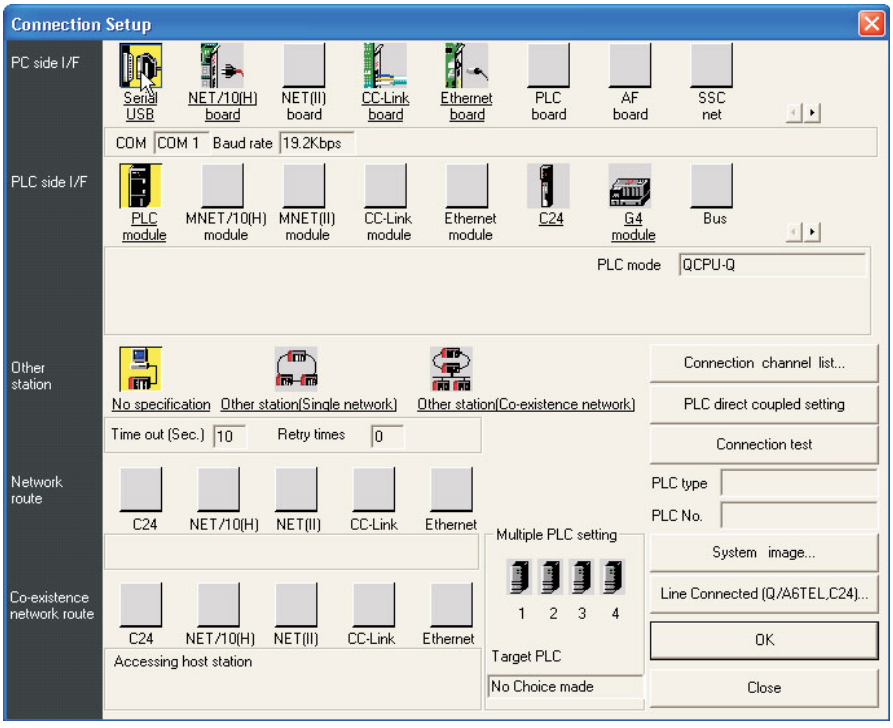


12.1.1 Communications Setup

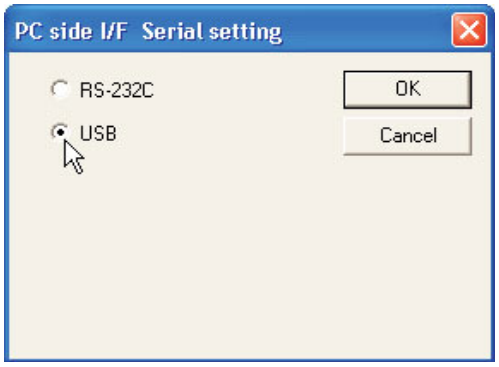
① From ***Online*** Menu, Select ***Transfer Setup***:



The following window will be displayed:

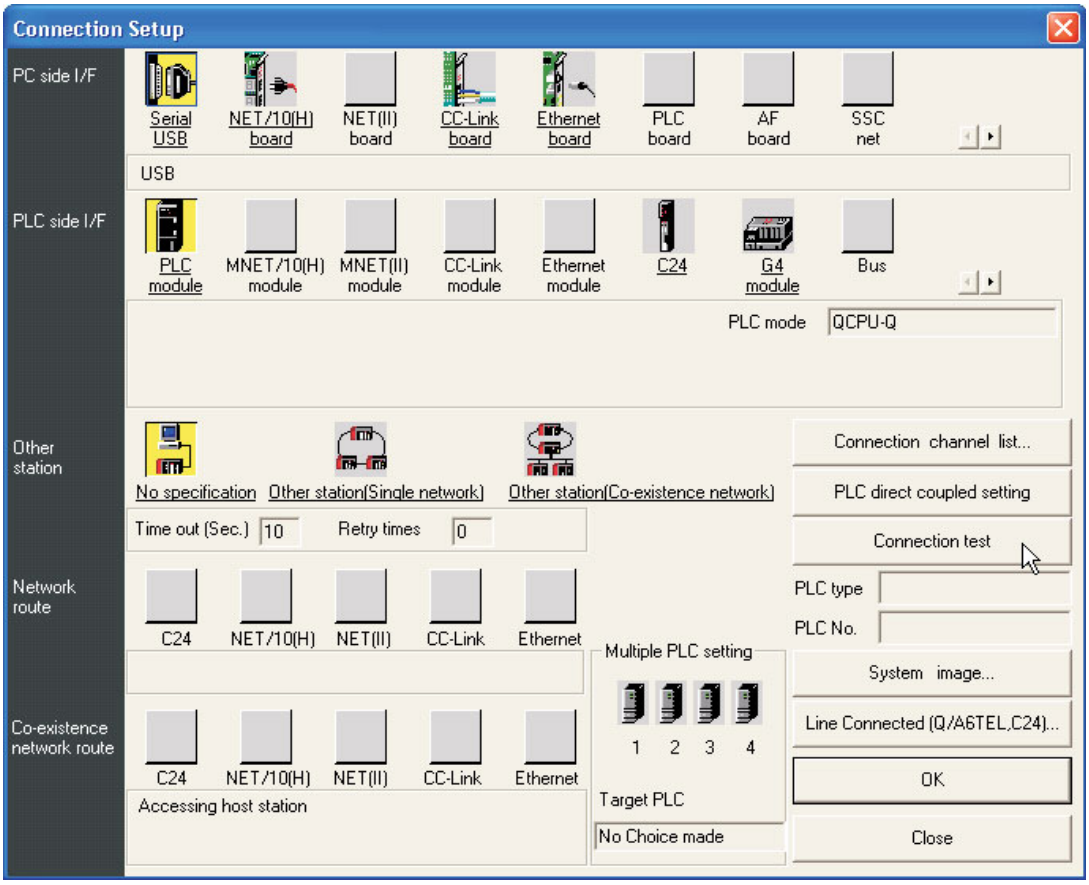


② Double click the mouse on the yellow ***PC side I/F*** – ***Serial*** Button and the following dialogue window is displayed:



③ Select ***USB*** as shown above and click ***OK***.

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



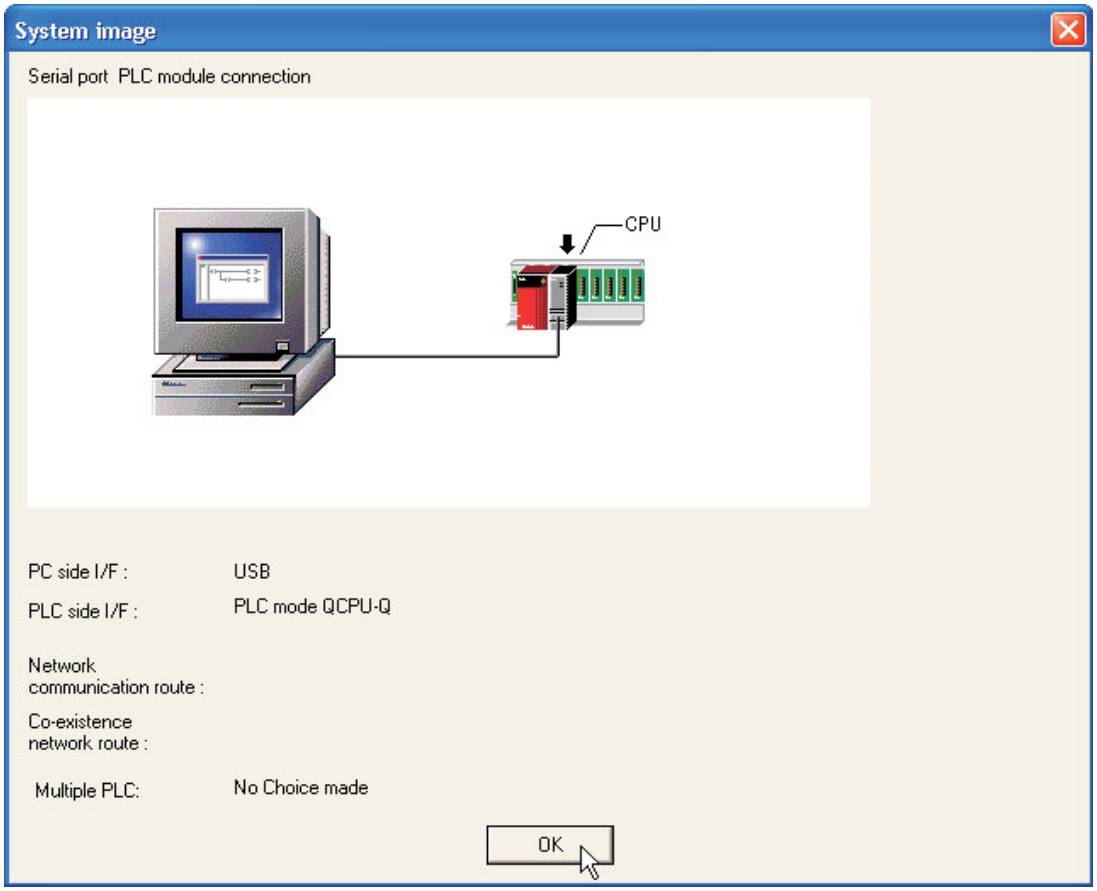
⑤ The following message should be displayed:

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

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

12.1.2 Connection Setup Route

① To obtain a pictorial view of the Connection setup route, select the ***System Image*** button



② Click ***OK*** to clear the display.

As can be seen from the previous display, these particular Connection Setup parameters utilise the USB Interface.

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.

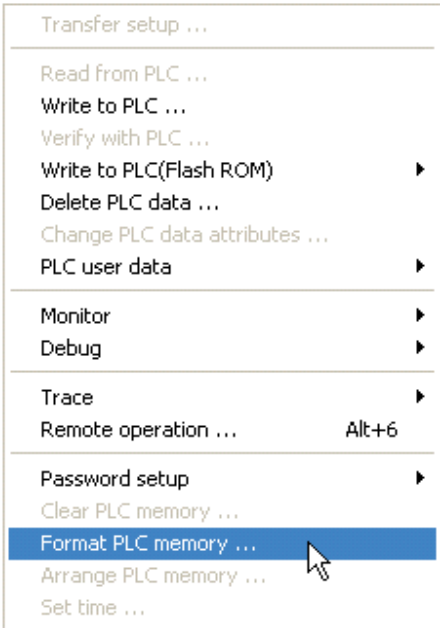
③ Select ***OK*** to close the ***System image*** display and return to the ***Connection setup*** display. Than click the ***OK*** button to close the ***Connection Setup*** window. If you leave the ***Connection Setup*** window using the ***Close*** button, the settings are not saved.

12.2 Formatting the PLC Memory (Q-Series)

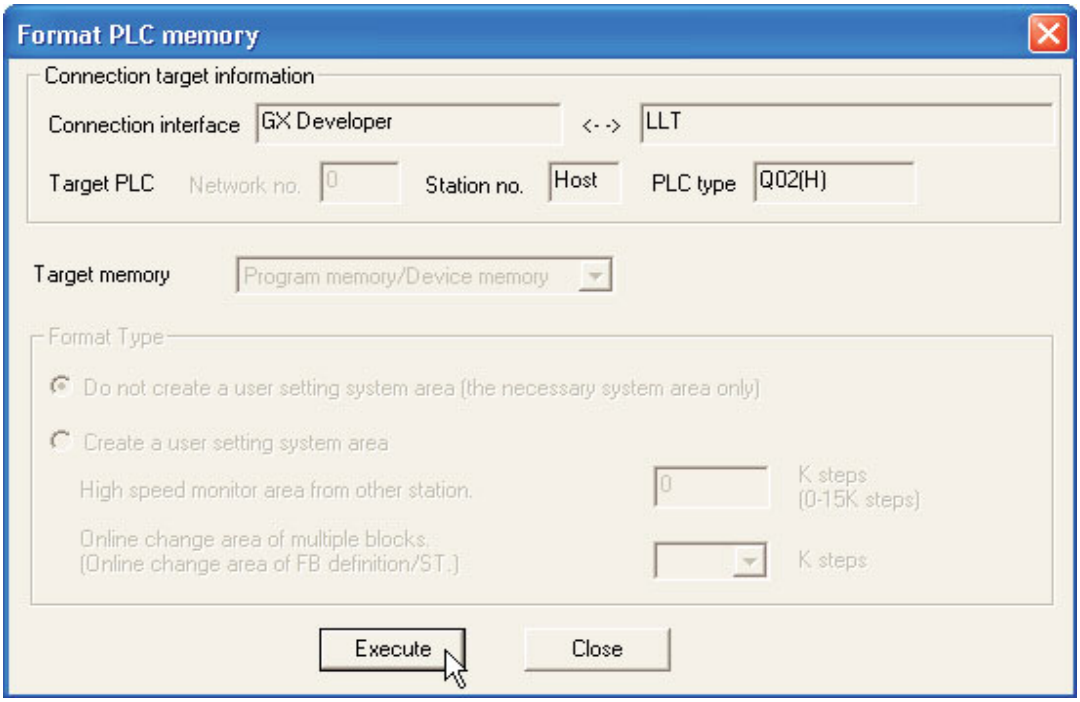
Due to the file method of program storage utilised in the Q-Series PLC's, it is always sensible to initialise the PLC memory before use. This is especially important when the PLC CPU is to be re-used. This ensures that no other programs are present in the CPU before writing new code to the memory.

Procedure:

① Select ***Format PLC Memory*** from the ***Online Menu***:



② When the following window is displayed, click ***Execute***:

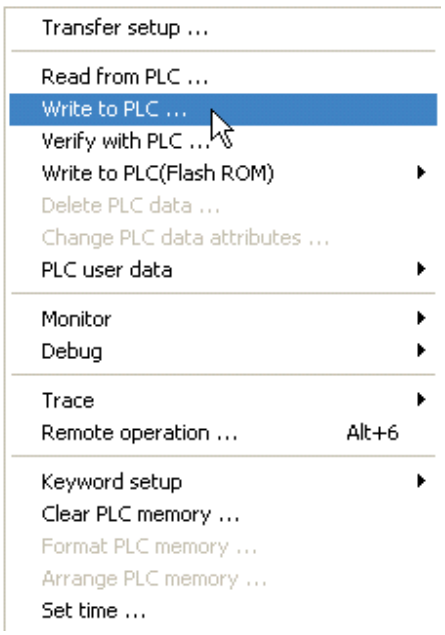


This will ensure that the CPU memory is reformatted and re-initialised ready for use with a new program.

12.3 Write Program to PLC

① From the Main menu, select ***Online***.

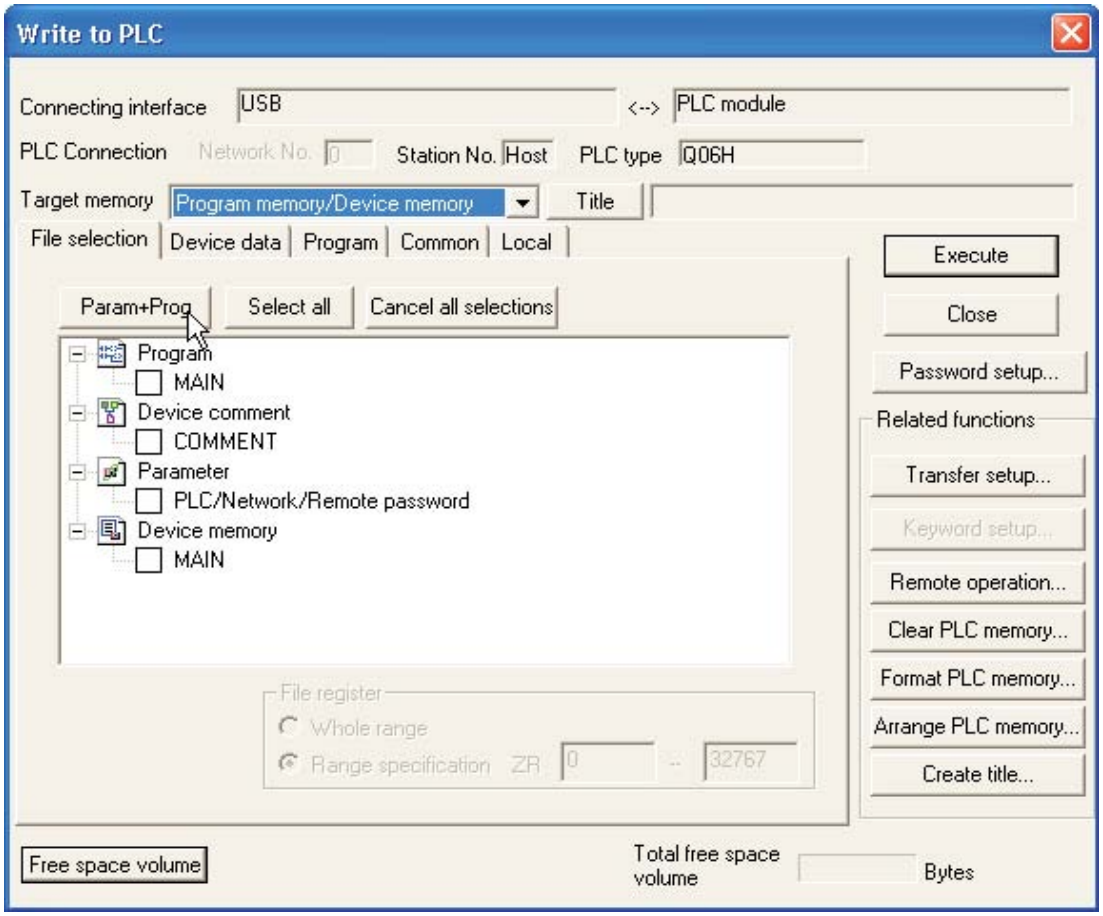
② Select ***Write to PLC***.



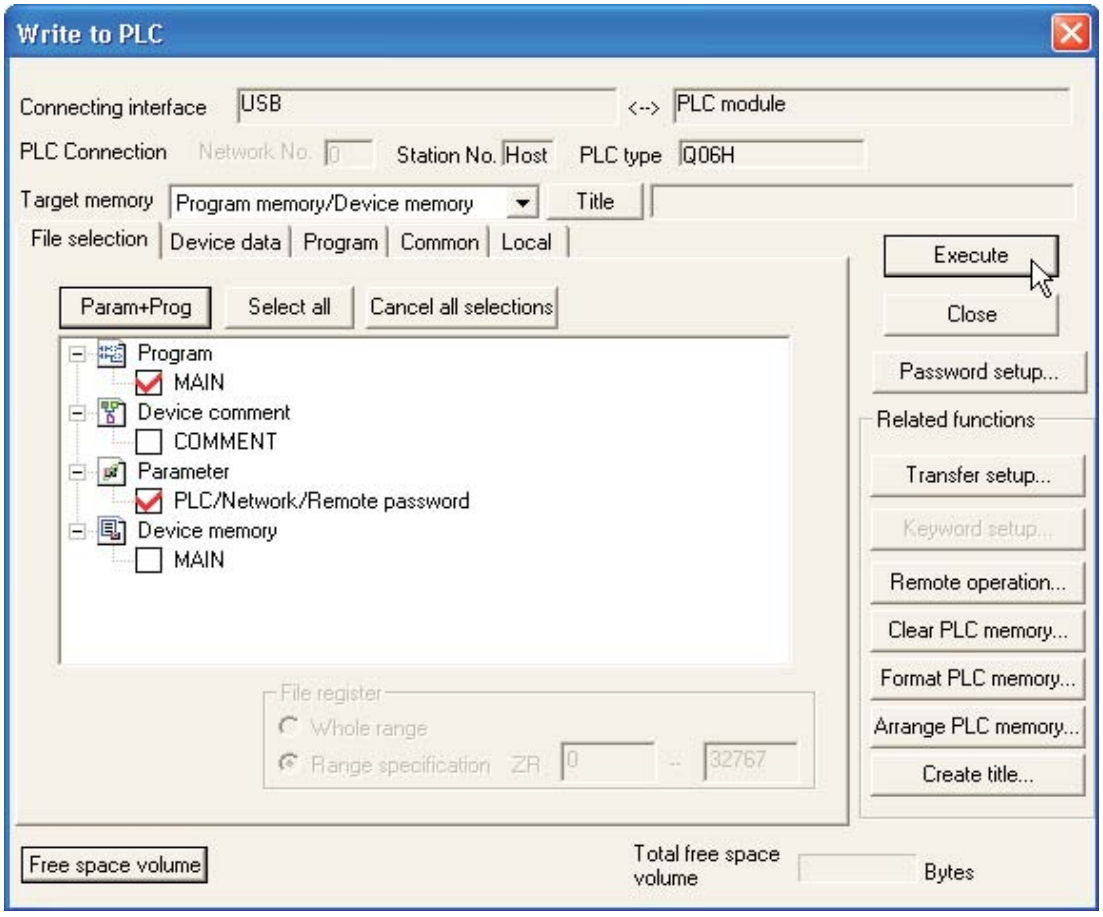
Alternatively, click on the  button from the tool bar.

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

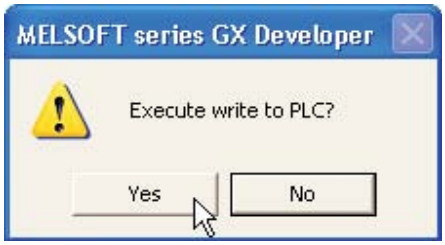
③ The display will be as show below:



④ Select the ***Param+Prog*** button on the display to enable the Program and Parameters for the project Q-SERIES-PROG4 to be downloaded:

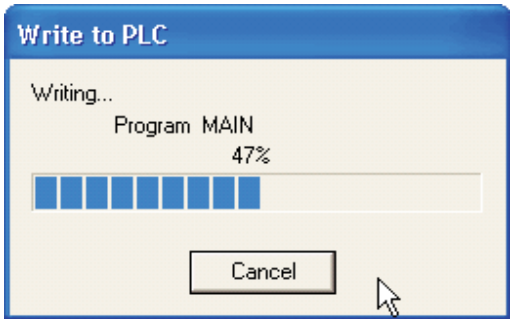


⑤ Select ***Execute*** and the following dialogue window is displayed:



⑥ 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:



⑦ When transfer is complete, the following message is displayed:

⑧ 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 Q-SERIES-PROG4 was downloaded, the default size of the program was actually 8000 steps. However, as Q-SERIES-PROG4 has only 21 steps, this means that the remaining 7979 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.

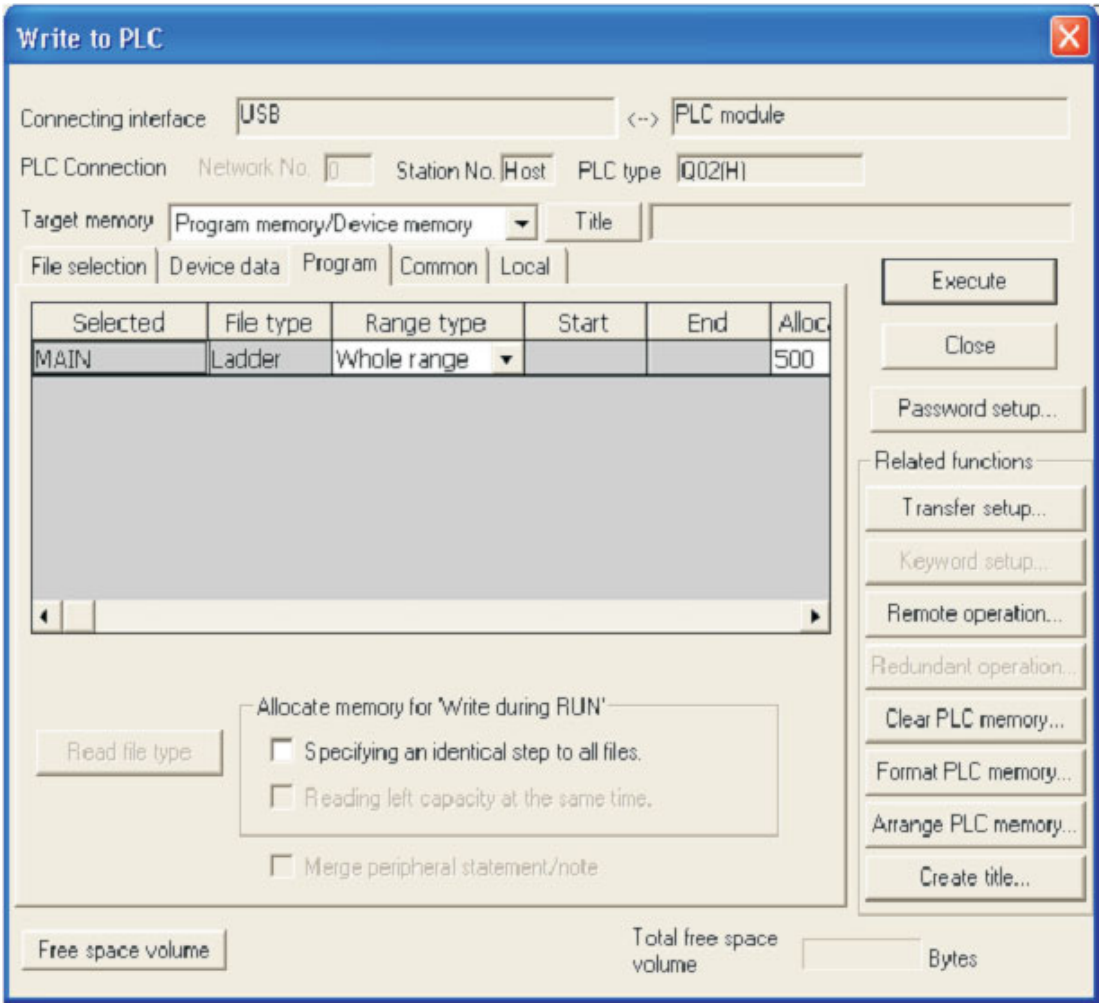
With GX developer versions less than V8.0, the time taken to write a program to the PLC on the A-Series or when using slower communication speeds on Serial Ports can be dramatically reduced by using the following procedure:

① Check that the PLC CPU is switched to Stop.

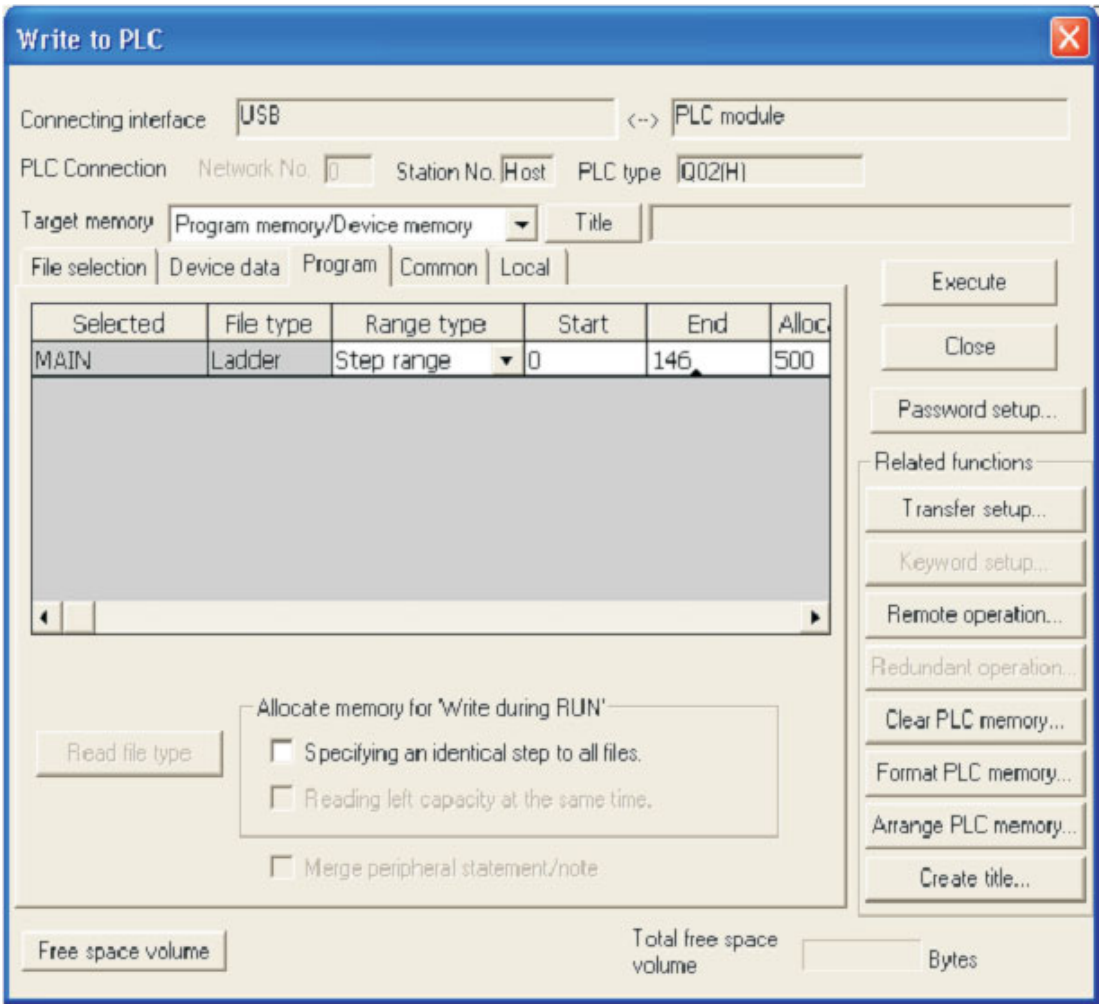
② Select ***Write to PLC***.

③ Select the ***Param+Prog*** button and select the ***Program*** tab.

④ The display now becomes as shown below:



⑤ Click on the ***Step range specification*** 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.

⑥ Select ***Execute*** and answer ***Yes*** to write the Parameters and only the used steps of Q-SERIES-PROG4 to the PLC.

13 Executing the Project

To execute the project Q-SERIES-PROG4, while referring to the Ladder Diagram on GX Developer, carry out the following.

① On the Q-SERIES PLC, switch the mode switch to the RUN position.

② Toggle the switch X10 ON then OFF. Y20 will illuminate and remain ON.

③ Repetitively operate switch X12 and observe that after 10 operations, Y21 indicator in the Training Rig will begin flashing at 1Hz.

④ Momentarily operate the X13 Switch and observe Y21 turns OFF.

⑤ Momentarily operate X11 and note that Y20 turns OFF.

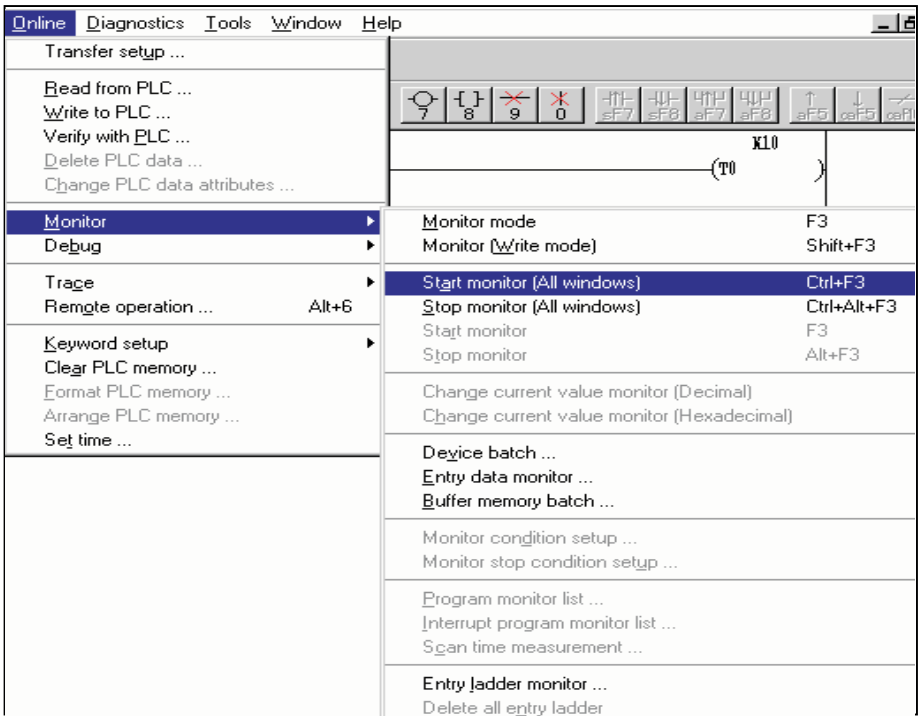
14 Monitoring

14.1 Monitoring the example program Q-SERIES-PROG4

To monitor the Ladder Diagram of Q-SERIES-PROG4, carry out the following.

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

② Select ***Monitor***



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

NOTE

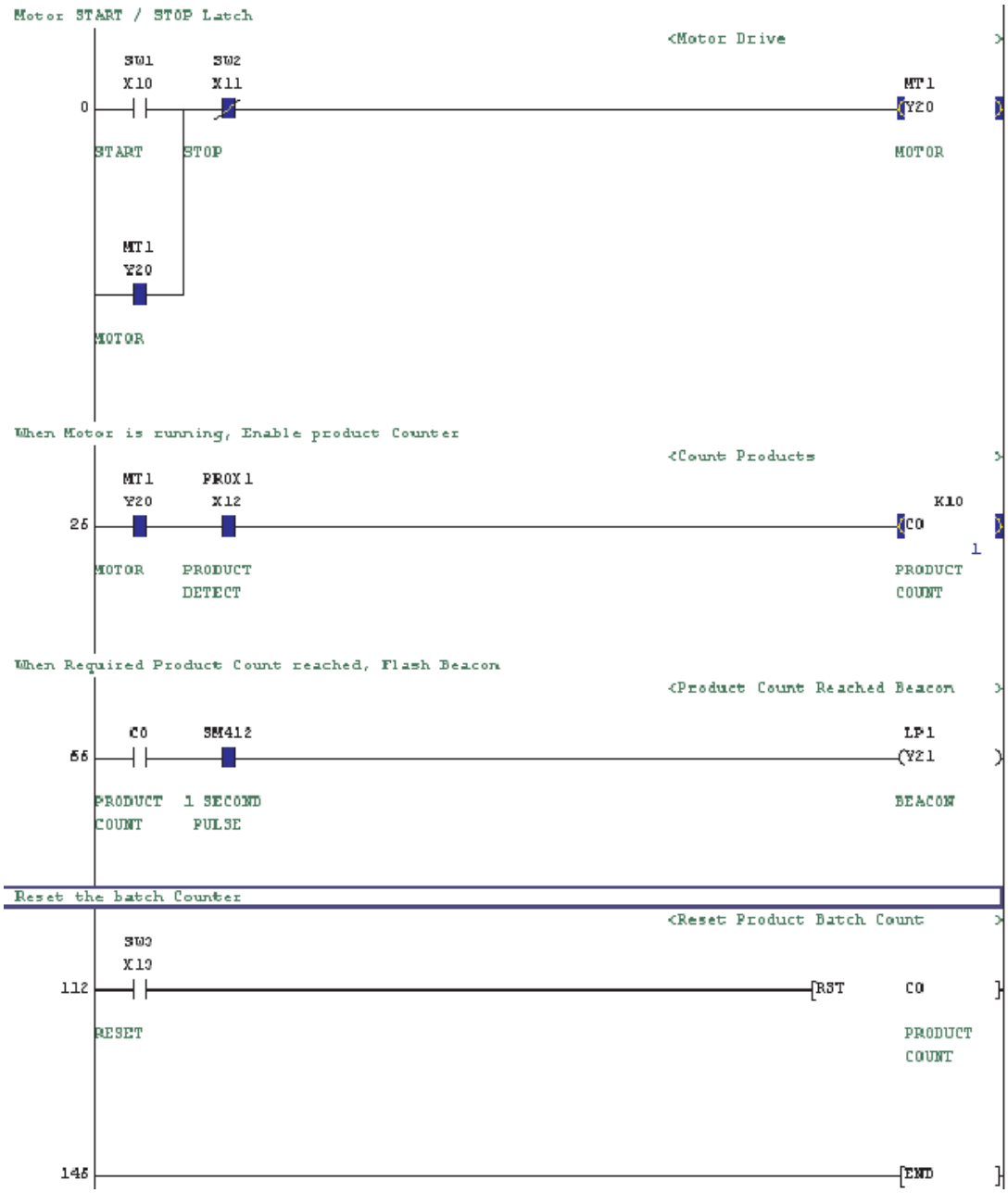
Use of the ***Start Monitoring*** – F3 shortcut key:
As can be see from the display, an alternative to the drop down menus to start monitoring is the F3 key.

Alternatively, use the  Icon to start monitor mode.

14.2 Monitored Display (Q-SERIES-PROG4)

The following display shows the Ladder Diagram Q-SERIES-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:



14.3 Entry Data Monitoring

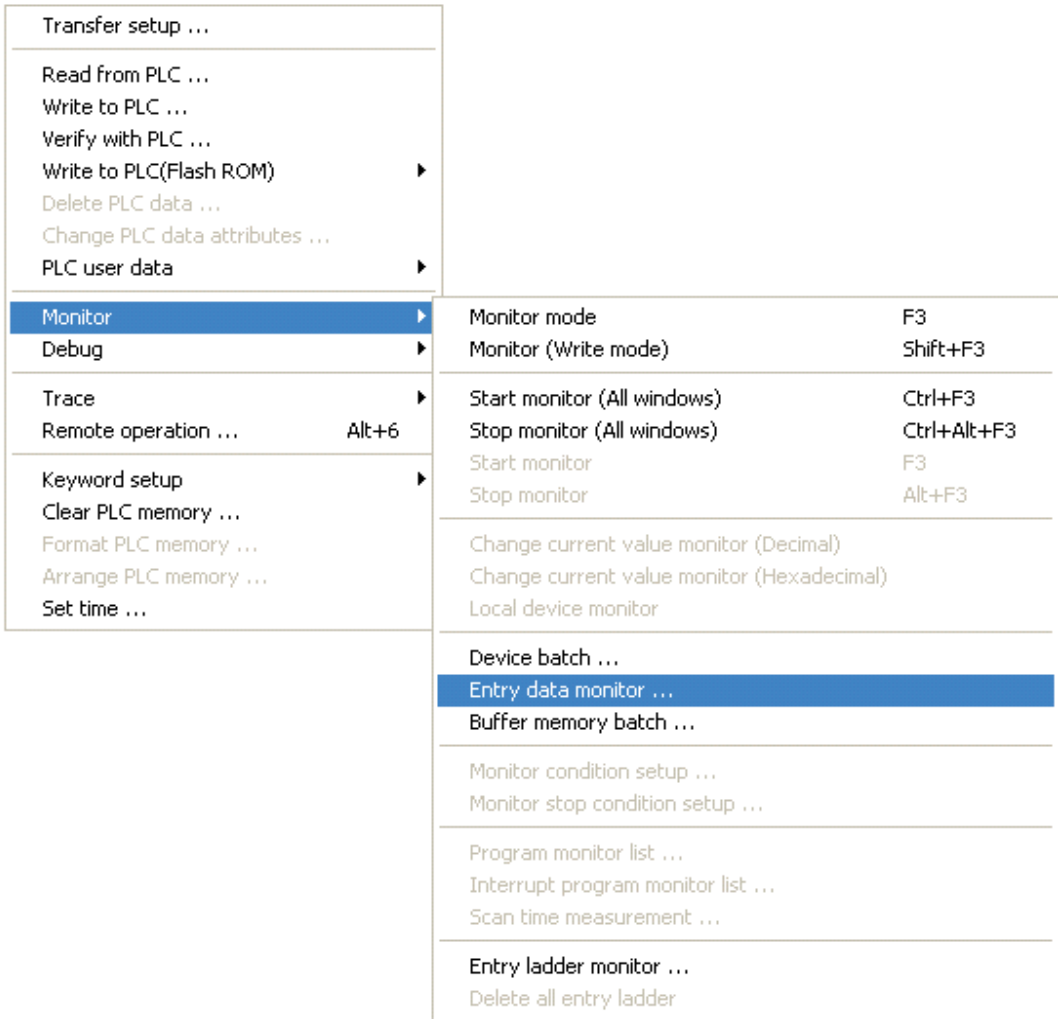
Entry Data Monitoring is an alternative method for monitoring the conditions of the Ladder Diagram elements. It enables the condition of many more devices to be displayed in addition to those on the active ladder monitor window.

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

① From the main select ***Online***.

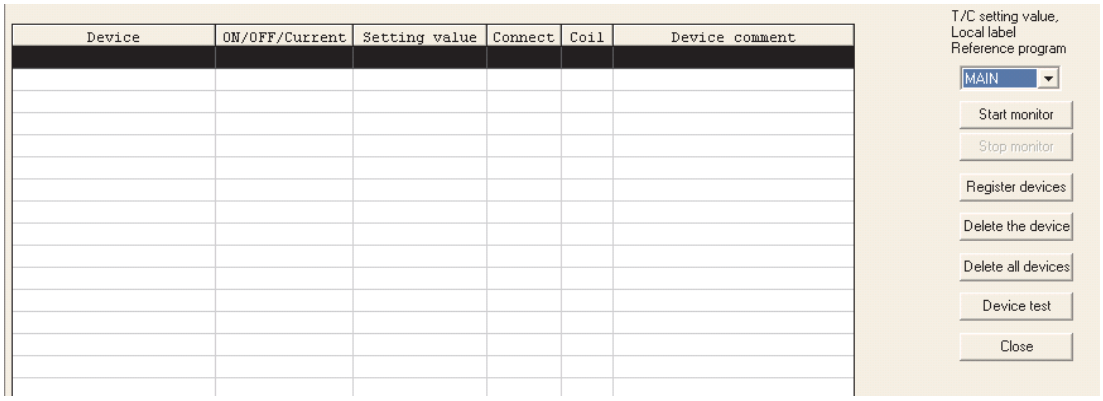
② Select ***Monitor***.

③ Select ***Entry Data Monitor*** thus:

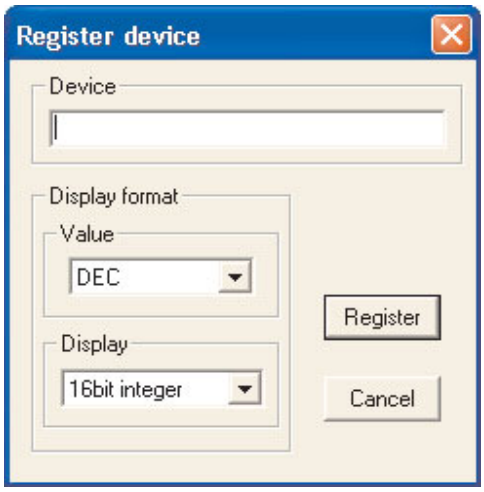


Alternatively, press the  button from the toolbar.

The following window will be presented.



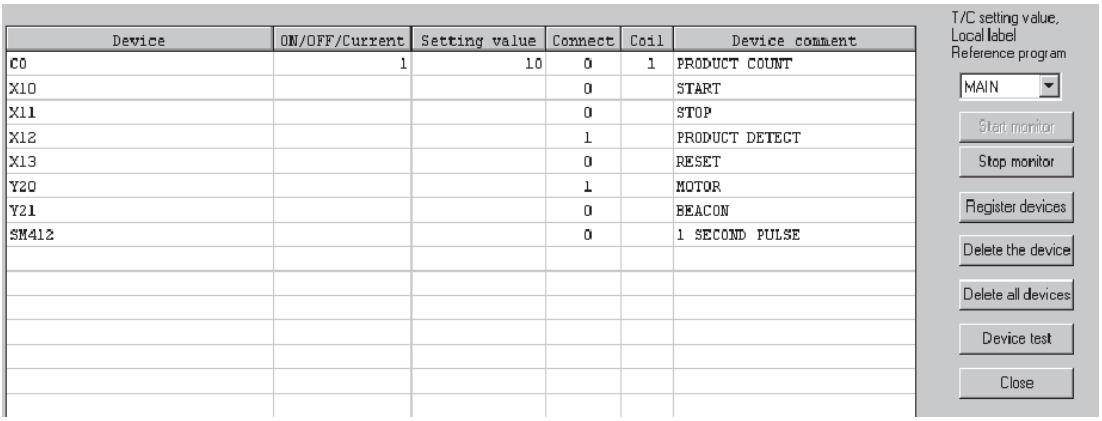
④ Select ***Register Devices*** to obtain the ***Register device*** window.



⑤ Enter the following device names into the window using the register button, press cancel when complete:

- C0
- X10
- X11
- X12
- X13
- Y20
- Y21
- SM412

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).

⑧ 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.

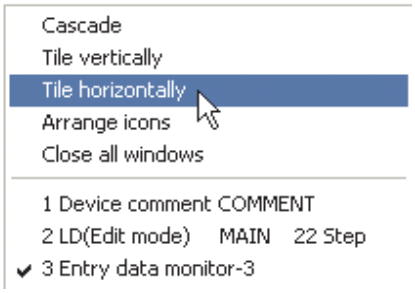
⑨ To clear all devices registered in the monitor window, select the ***Delete All Devices*** button.

14.4 Combined Ladder and Entry Data Monitoring

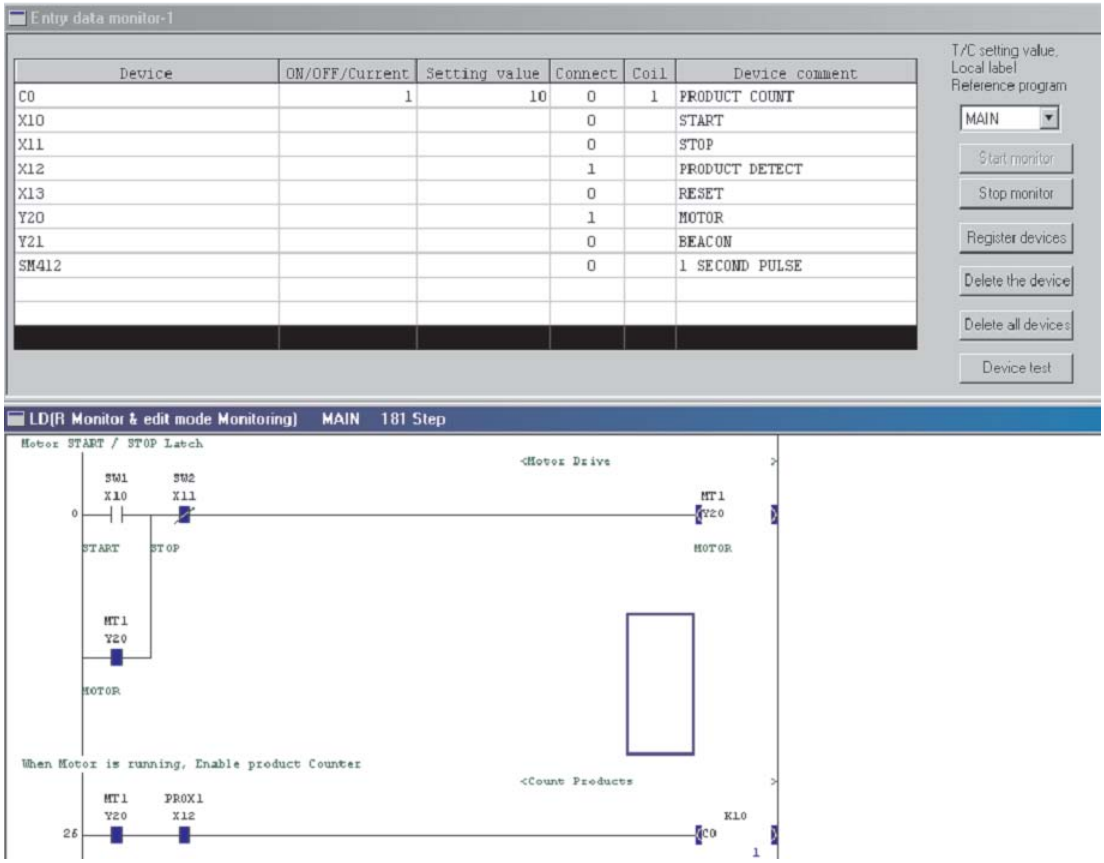
Using Windows, it is possible to monitor both the Ladder Diagram and the Entry Data.

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:



15 Function Block Programming

15.1 What is a Function Block (FB)

The FB is a function designed to convert a ladder block, which is used in a sequence program repeatedly, into a component (FB) to utilize it in the sequence program.

This not only increases the efficiency of program development but also reduces programming mistakes to improve program Quality.

15.1.1 Precautions

- The function block (FB) programming is not available for following Q-CPUs:
  - Q00JCPU
  - Q00CPU
  - Q01CPU
- The FB cannot be used within the FB.
- When the FB definition program has been corrected, online program correction cannot be made to the sequence program that includes the FB.

15.1.2 FB Device Types

Five FB device types exist:

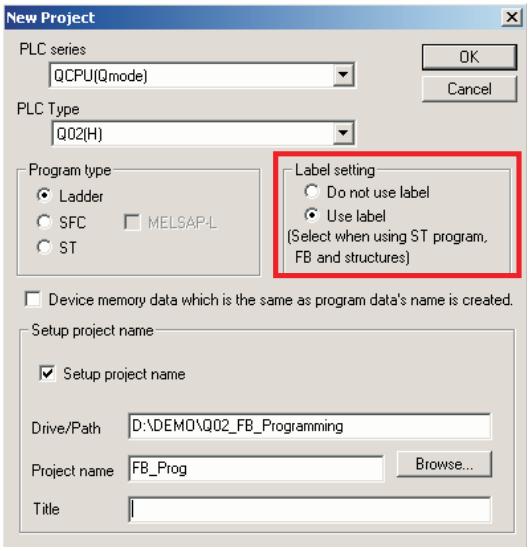
- BOOL: Data represented by ON/OFF.
- INT: Data represented by 16 bits.
- DINT: Data represented by 32 bits.
- REAL: Floating-point data represented by 32 bits.
- STRING: Character string data

The FB definition automatically assigns devices to the labels used in a program. When creating a sequence program, avoid automatically assigned devices. Like local devices, the automatically assigned devices are set in accordance with the automatically assigned device setting, which defaults to the following device ranges.

- Word device : D6144 to D12287
- Bit device : M4096 to M8191
- Timer : T64 to T2047
- Counter : C512 to C1023

15.1.3 Creating a new Project

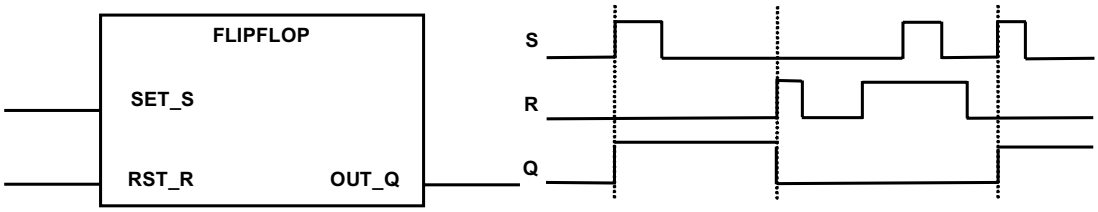
When creating a new project including FB, ***Use label*** has to be selected in the ***Label Setting***.



15.2 Creating a new FB

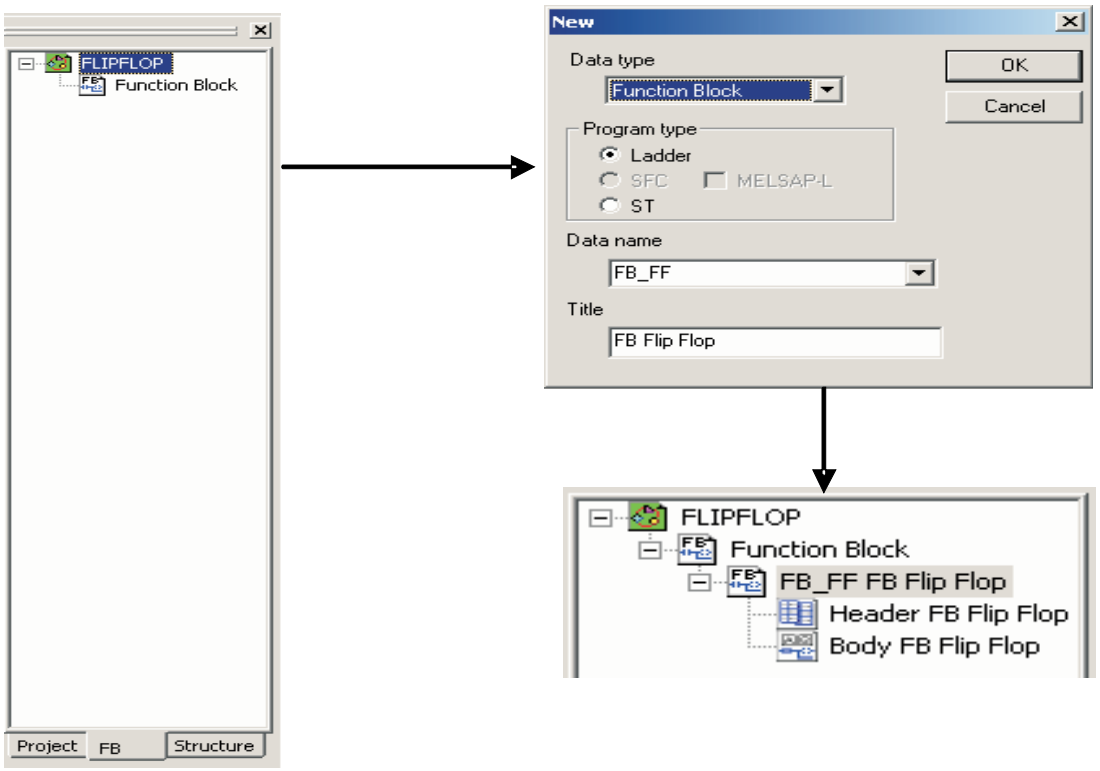
Example

Program a bistabile element (R/S-Flip-Flop), Reset has a higher priority!



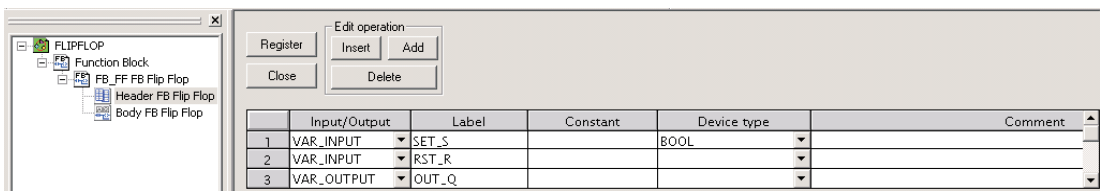
15.2.1 Add a new FB

Open the Folder “FB” in the project list and make a right mouse click in the Project list display. The ***New*** window opens where the data type, program type and the data name have to selected.



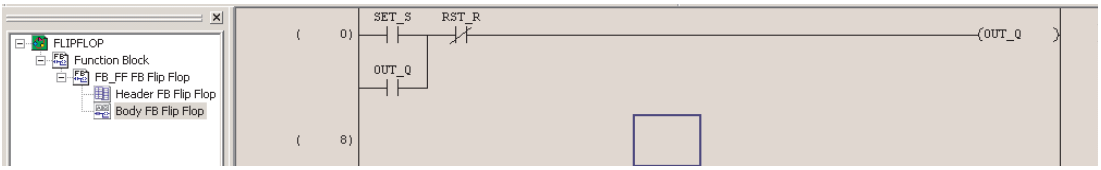
15.2.2 Define Input and Output variables

Define the input and output variables of the header of the FB.



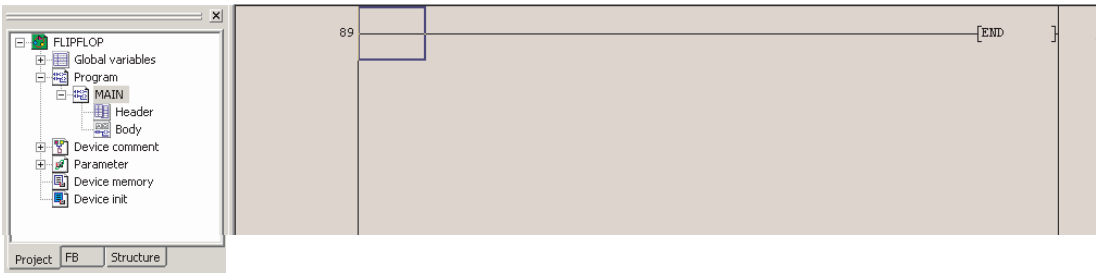
15.2.3 Program the Flip Flop

Program the body of the FB.

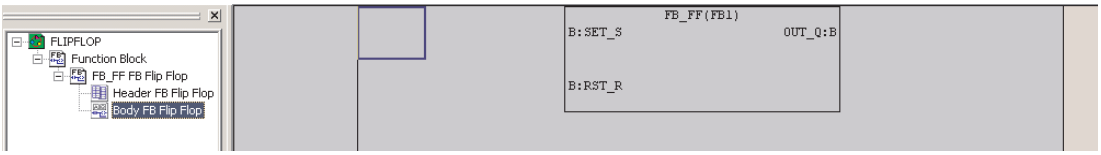


15.2.4 Calling the FB from within the program

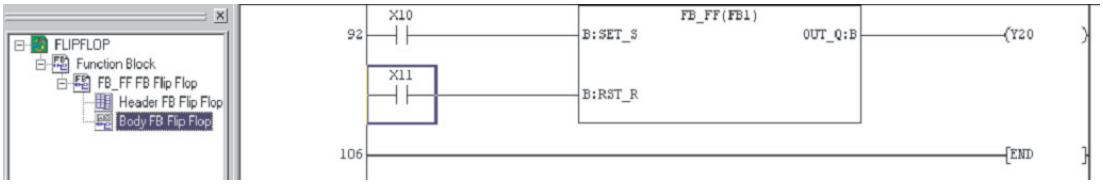
Change to the project tag and open the body of the main block .



Change back to the FB tag. Take the Body of the FB and put it via drag & drop into the project.



Connect the inputs and the output of the FB.



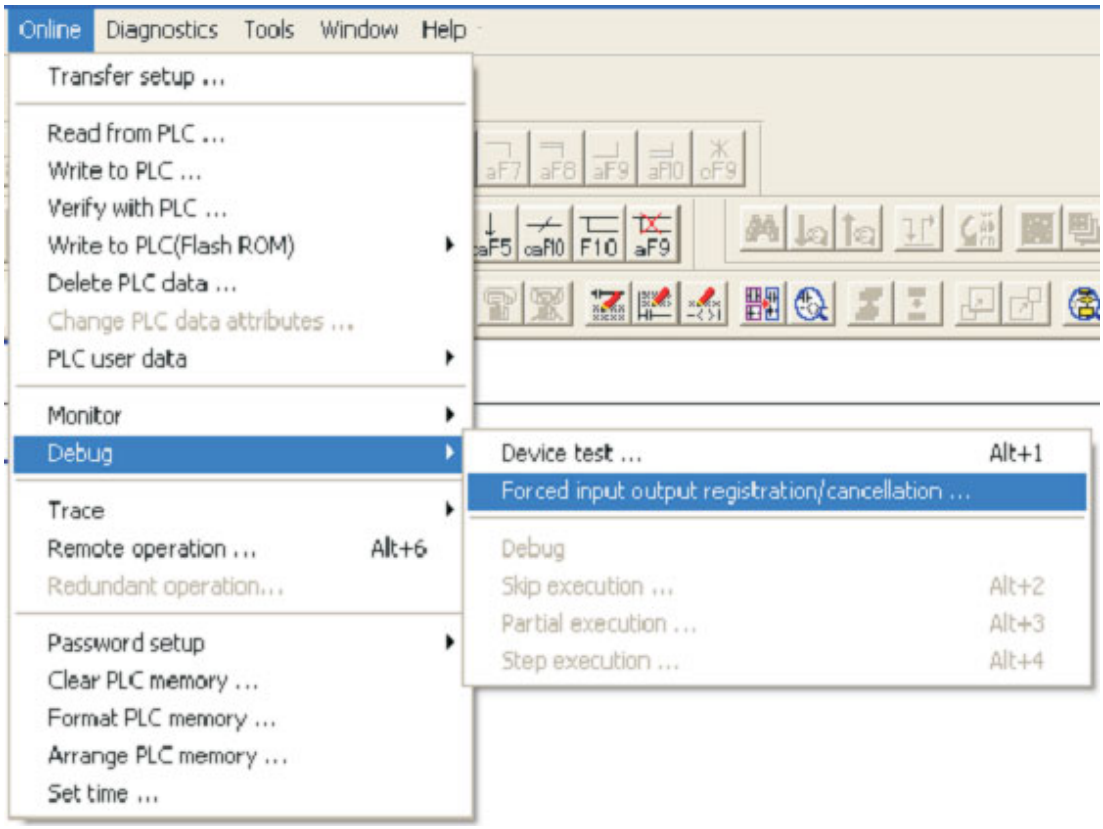
16 Forcing Inputs and Outputs

16.1 Registration/Cancellation of Forced Inputs/Outputs

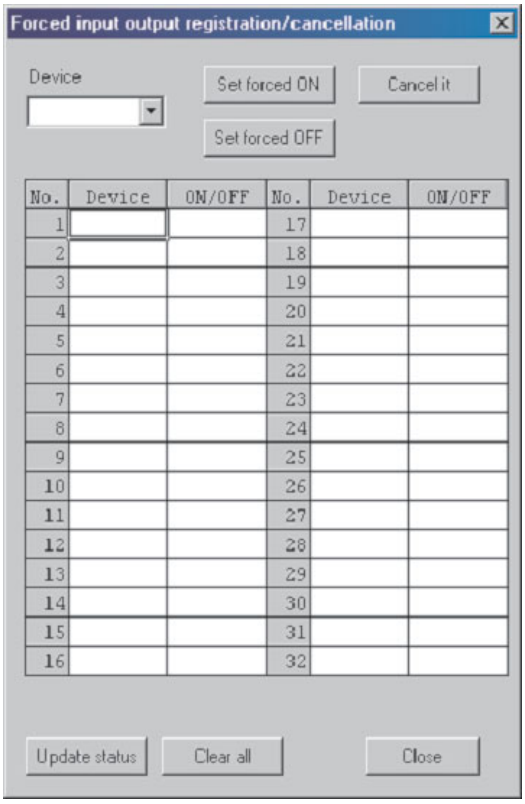
This GX-Developer feature enables both Input and Output registers to be forced independently from the program scan in the Q-Series family only.

This feature is particularly useful as it enables the states of all physical Input and Output devices to be overridden.

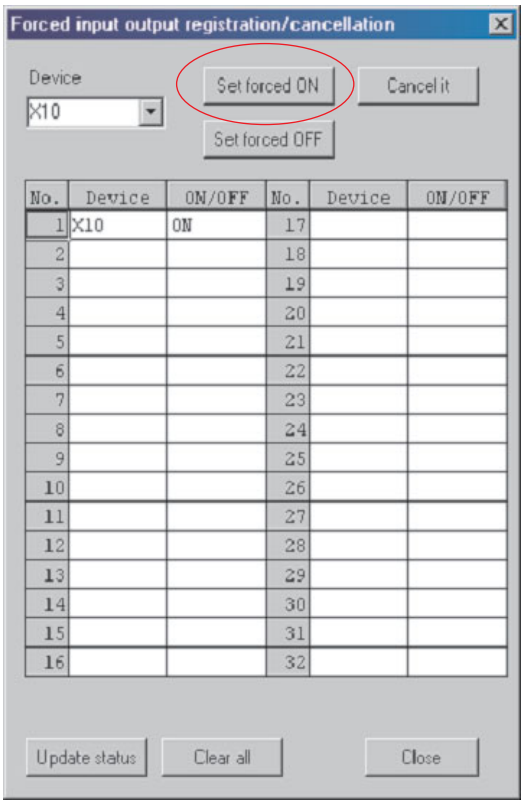
① To activate this function, while in Monitor mode, select ***Debug*** from the ***Online*** menu and than the ***Forced input output registration/cancellation*** option thus:



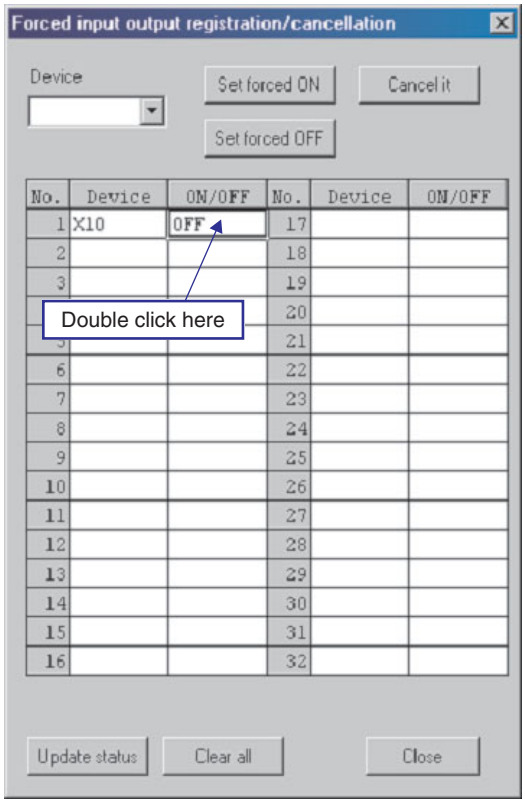
The following window will be displayed:



② Enter X10 into the ***Device*** dialogue box and click on the ***Set Forced ON*** button:

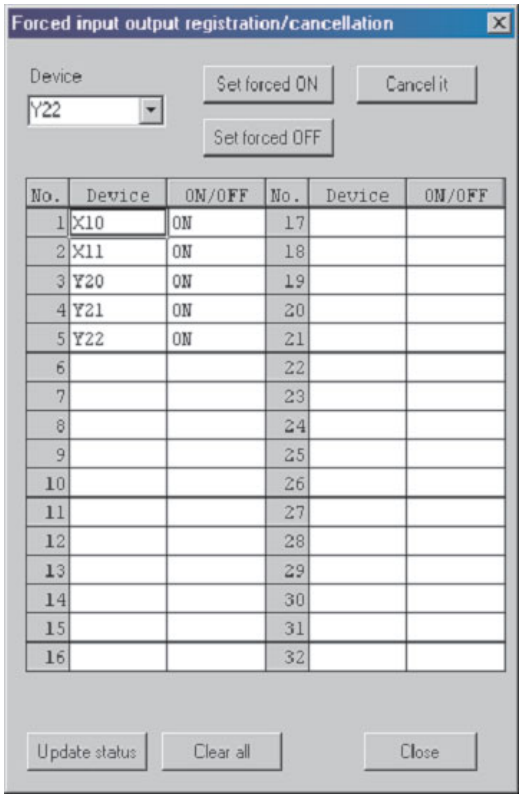


③ To toggle the status of X0, Double Click the left mouse button over the ***ON/OFF*** status cell:

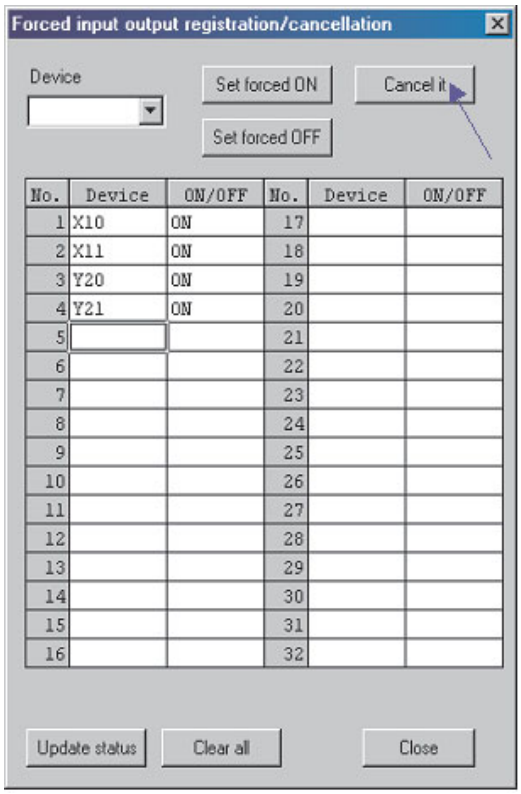


④ Carry out this method of forcing on X11, Y20, Y21 and Y22, noting the effect on the devices.

⑤ To clear a force on an individual device, enter the device then click on the ***Cancel it*** button thus:



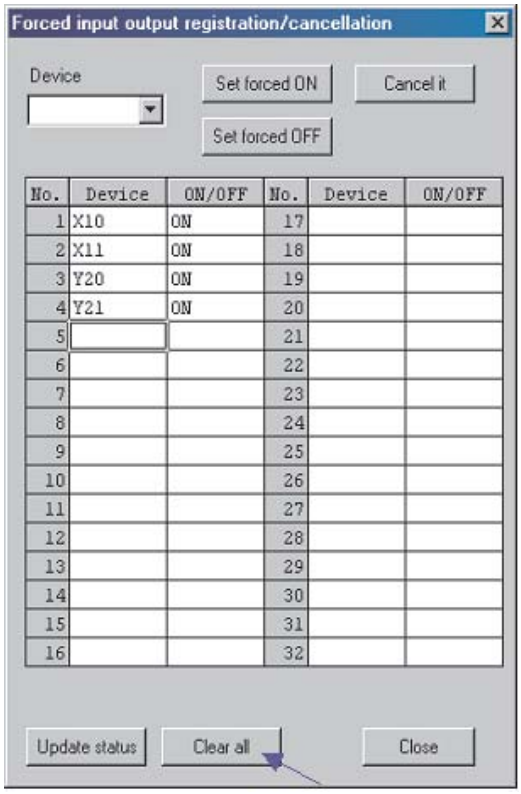
⑥ The following display will result:



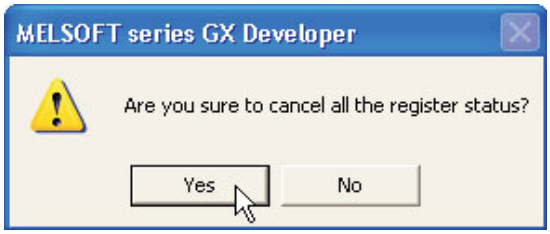
NOTE

When any forces are registered within the PLC, the Mode light on the CPU flashes at about 2Hz.

⑦ To clear all forces registered in the CPU, click the ***Clear All*** button:



⑧ Confirm the cancellation request using the following response:



17 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.

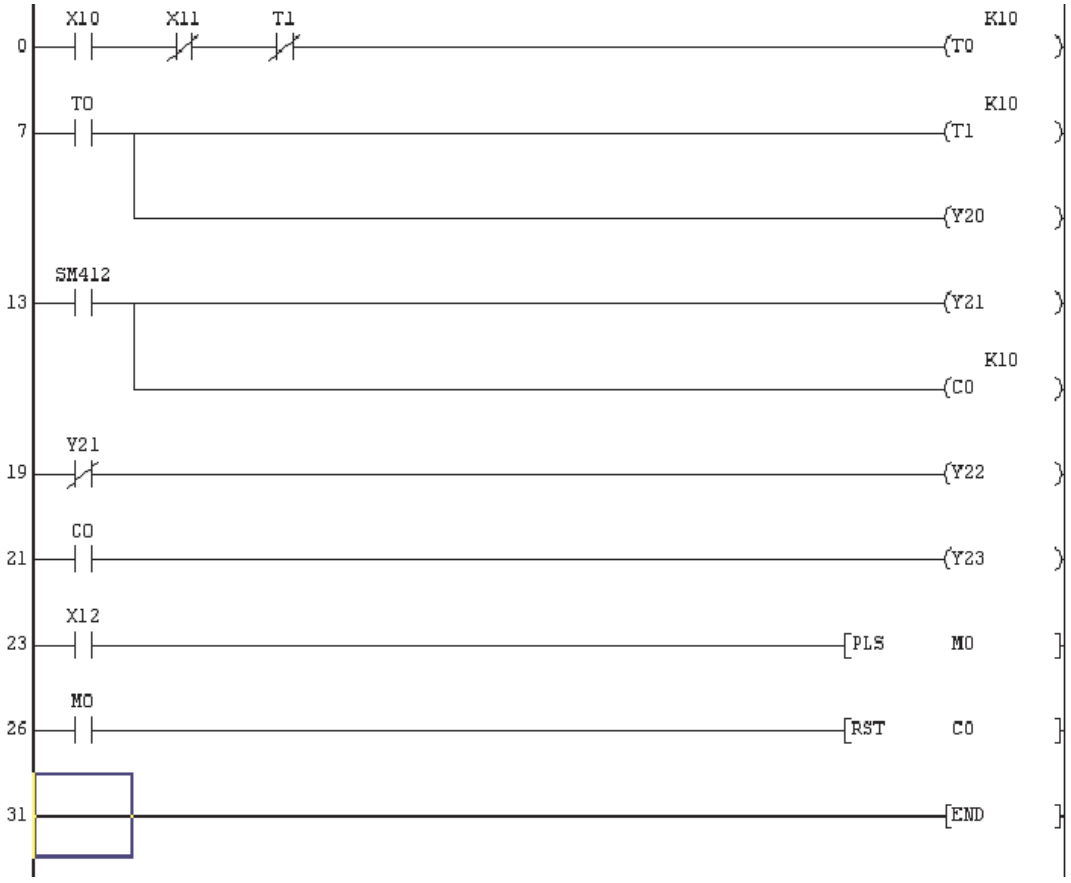
17.1 Verification of Example Programs

To demonstrate the Verify facility, the projects Q-SERIES-PROG4 and Q-SERIES-PROG2 will be used.

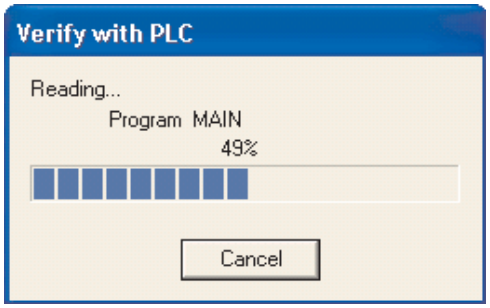
Having previously downloaded it, the project Q-SERIES-PROG4 is currently residing in the PLC.

① Return to the Main Menu, select ***Project*** and ***Open Project*** and open the program Q-SERIES-PROG2.

Ladder Diagram – Q-SERIES-PROG2



② Select ***Online*** and ***Verify with PLC***, then Select ***Param+Prog***. The display will show the following dialogue:



③ When the verification procedure has completed, the following display will be shown illustrating the differences found between the two programs:

```
[Project verify: Program]
 Verify source
   Project name -C:¥MELSEC¥DATA¥Q-SERIES-PROG2
   Data name -MAIN
 Verify destination
   Project name -C:¥MELSEC¥DATA¥Q-SERIES-PROG4
   Data name -MAIN
```

<Memory> Step	Instruction	<File> Step	Instruction
0	LD X10	0	Motor START / STOP Latch
1	ANI X11	14	LD X10
2	ANI T1	15	OR Y20
3	OUT T0 K10	16	ANI X11
7	LD T0	17	OUT Y20
8	OUT T1 K10	18	Motor Drive
12	OUT Y20	26	When Motor is running, Enable produ...
13	LD SM412	51	LD Y20
14	OUT Y21	52	AND X12
19	LDI Y21	57	Count Products
20	OUT Y22	66	When Required Product Count reached...
22	OUT Y23	94	AND SM412
23	LD X12	95	OUT Y21
24	PLS M0	96	Product Count Reached Beacon
26	LD M0	112	Reset the batch Counter
27	RST C0	126	LD X13
31	END	127	RST C0

```
17 items unmatched.
```

④ As can be seen, the two projects Q-SERIES-PROG4 and Q-SERIES-PROG2 are quite different.

⑥ Select ***Main*** to return to the Ladder Diagram Q-SERIES-PROG2.

18 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.

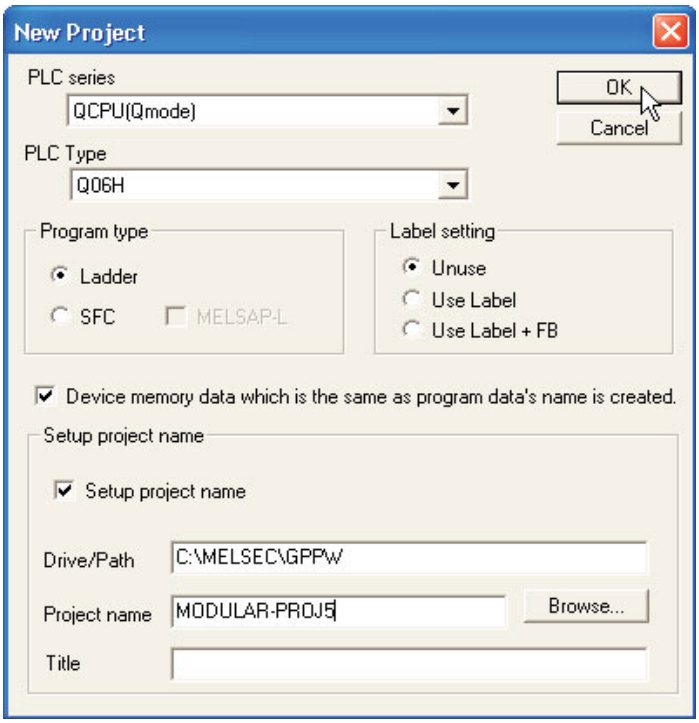
Hence after verifying that the program in the PLC is different to that stored on disk, then the working program within the PLC must be uploaded into GX Developer and stored on the Master Disks.

18.1 Example Program Upload

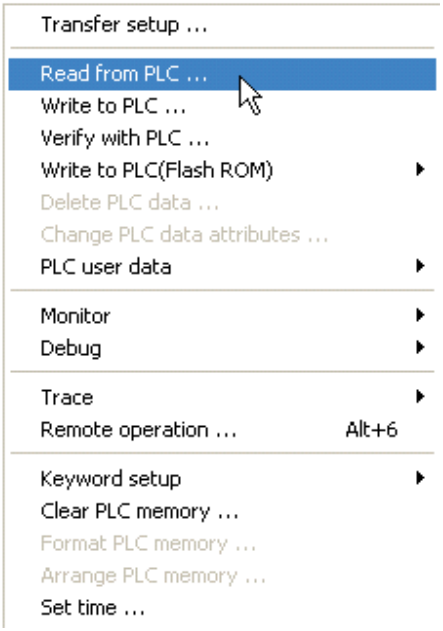
The following describes how the project Q-SERIES-PROG4 is uploaded from the Q-Series PLC and saved as Q-SERIES-PROG5. It is assumed that the program Q-SERIES-PROG4 is still resident in the PLC:

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 Q-SERIES-PROG5:

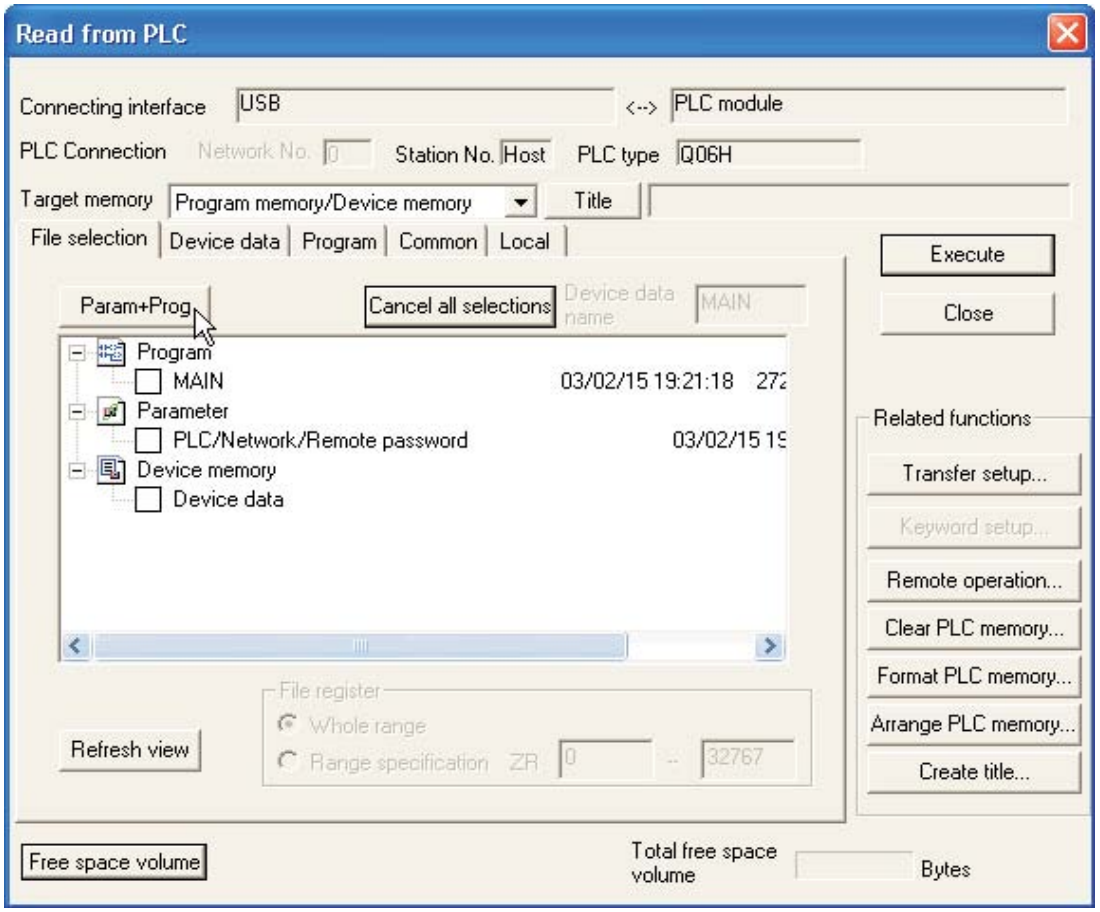


③ Select ***Online*** and ***Read from PLC***.

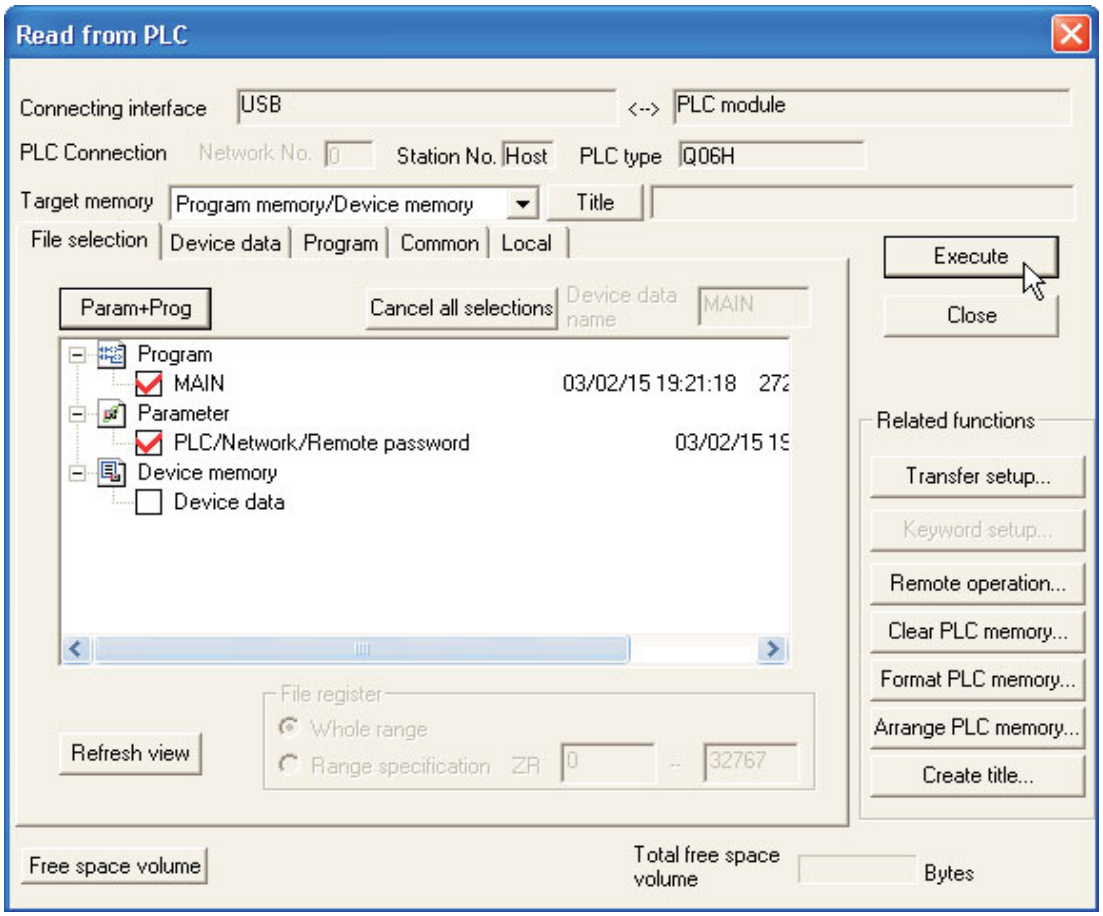


Alternatively click on the ***Read from PLC*** icon: 

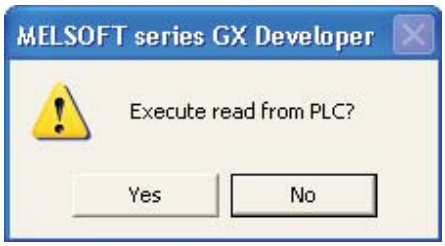
The Display will be as shown below.



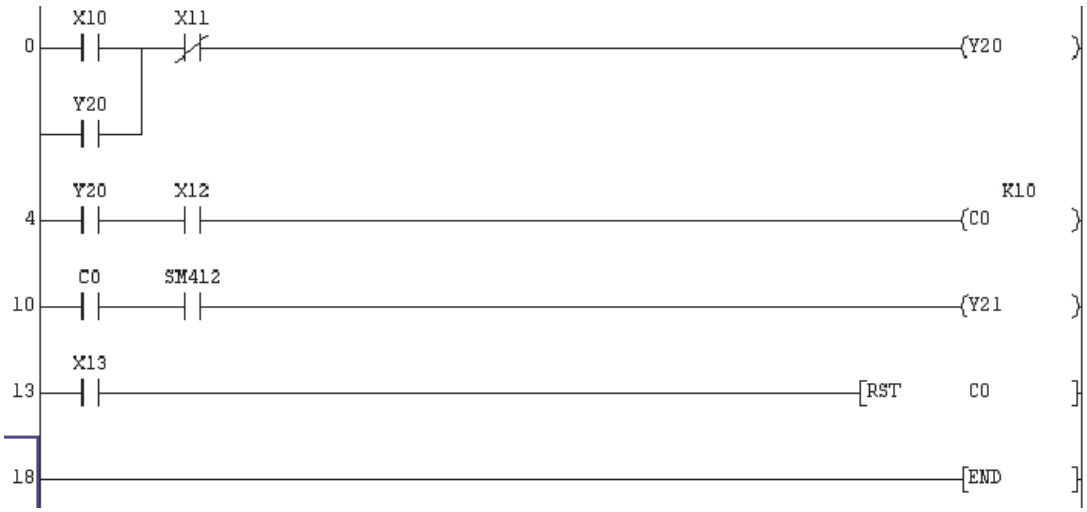
④ Select the ***Param+Prog*** button and click ***Execute*** as shown below:



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



⑥ Click the ***Close*** button on the dialogue window to close the transfer display. The uploaded program Q-SERIES-PROG5 from the PLC will now be displayed. (This was Q-SERIES-PROG4 stored in the PLC.)



⑧ Save Q-SERIES-PROG5.

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

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.

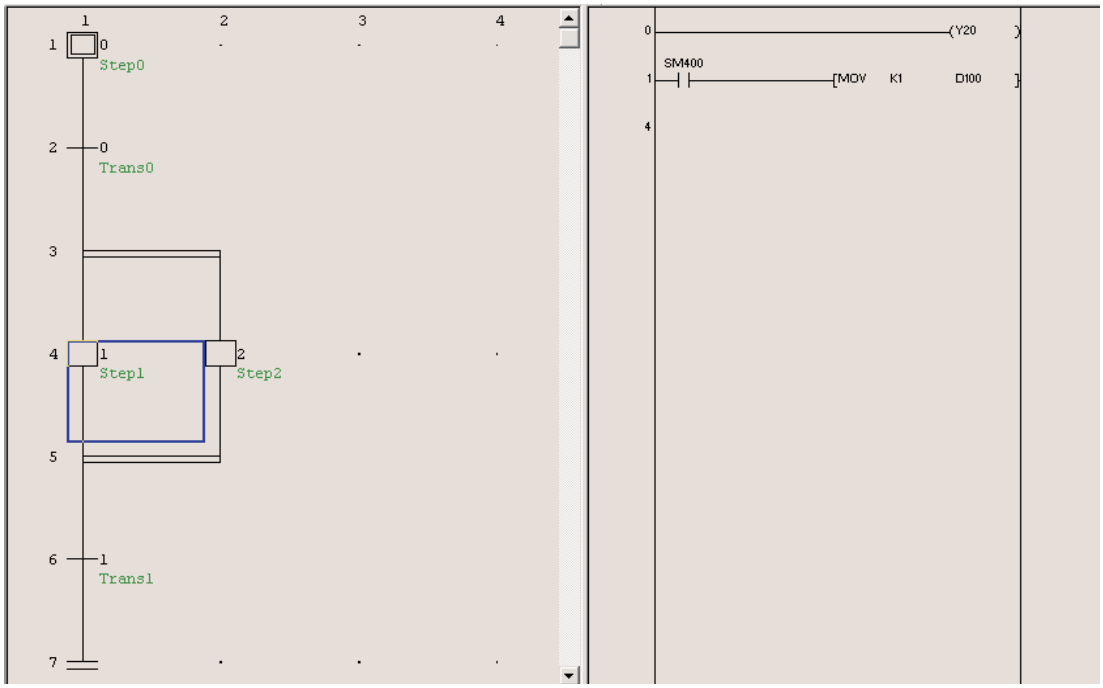
CAUTION:
ONLINE programming can be dangerous, since once the modifications have been entered, they become operative on the next scan of the program.

20 Sequential Function Chart

SFC is one of the graphical methods that can be used for programming the A series, QCPU (Q mode)/QnA series and FX series CPUs. 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.

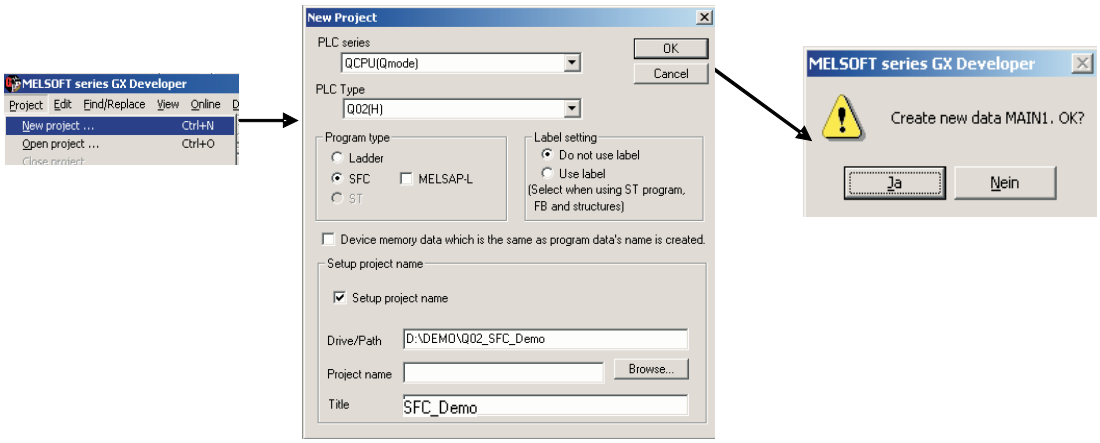
In contract to the case where a program represented by ladders is entirely executed every scan, only the minimum required part of a program may be run if it is written in SFC format.

- 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 Grafcet (IEC 848)
- GX Developer-SFC supports AnA, AnU and QnA, not AnN or AnS
- 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



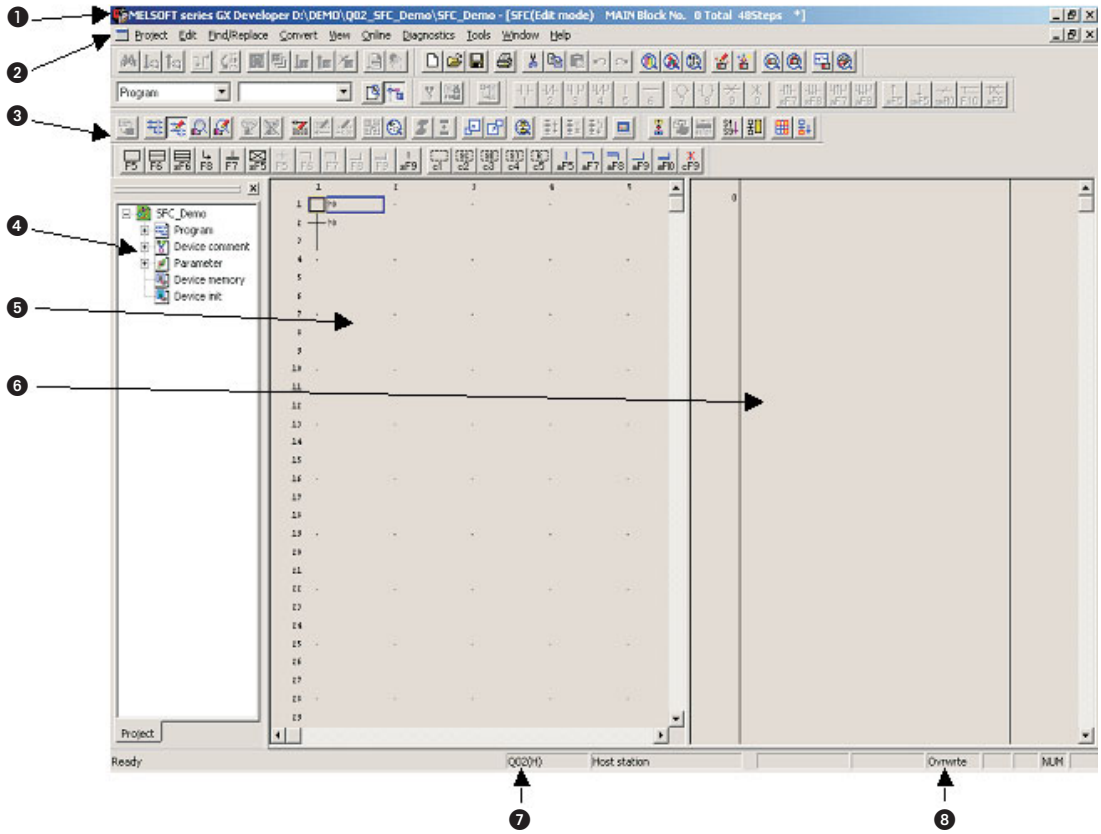
20.1 Creating a SFC-Block

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.



20.1.1 SFC Diagram editing screen

The SFC Diagram editing screen opens.



❶ Area for displaying the project name being edited, the number of steps used, the block number being displayed and so on

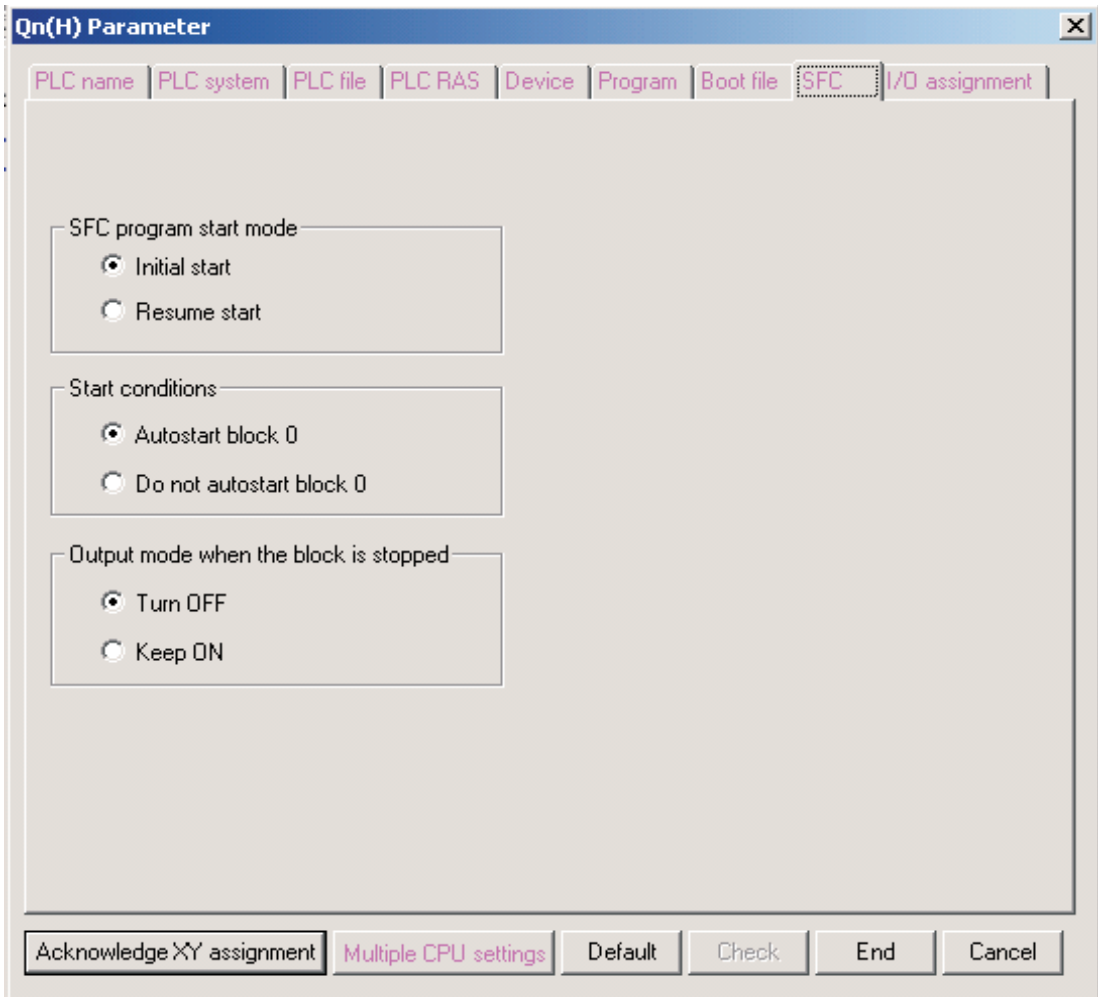
❷ Menu names on the menu bar

❸ Icons on the toolbar

❹ Project list display
❺ SFC diagram editing area
❻ Operation output/transition condition program editing area (Zoom side)
❼ Edited CPU type
❽ Edit mode (overwrite/insert)

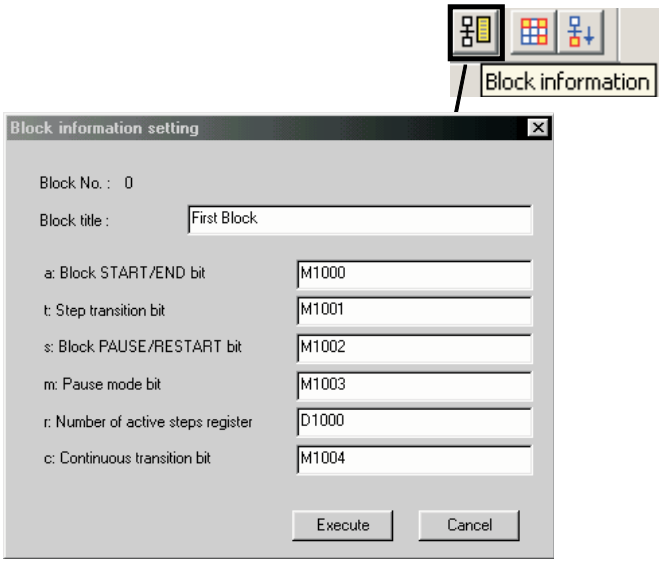
20.1.2 SFC Parameter

Select ***PLC Parameter*** at Project Navigator for the SFC Parameter window.



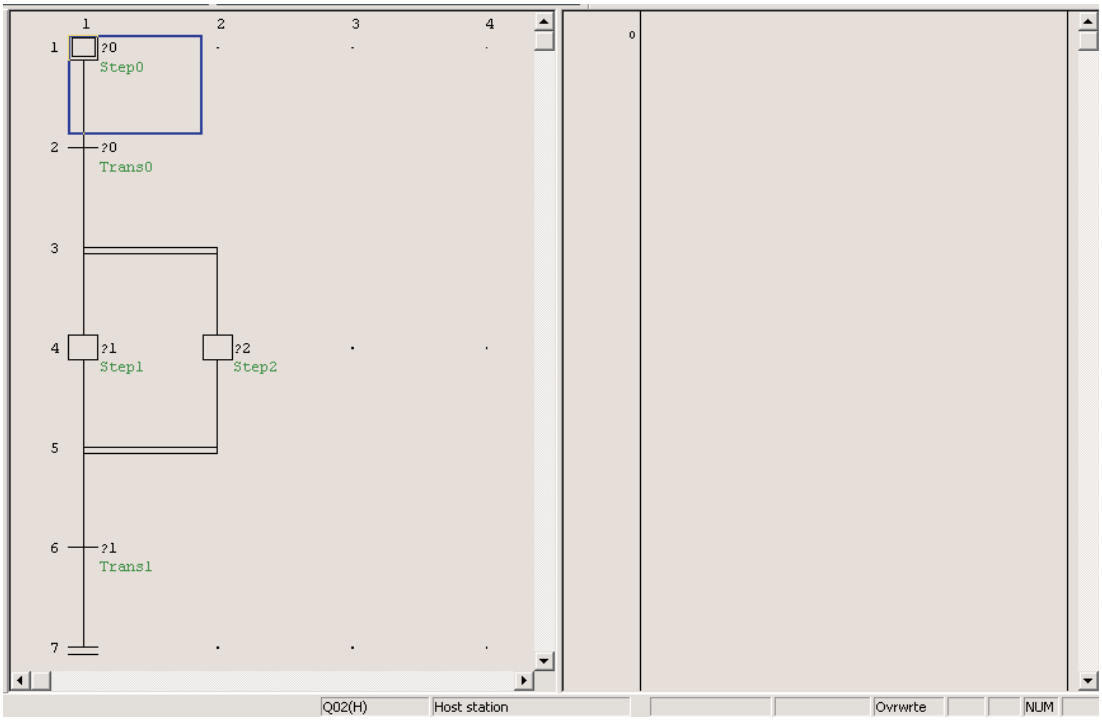
20.1.3 Block information

Select ***Block Information*** at the menu ***Edit*** for setting the block information to the corresponding block.



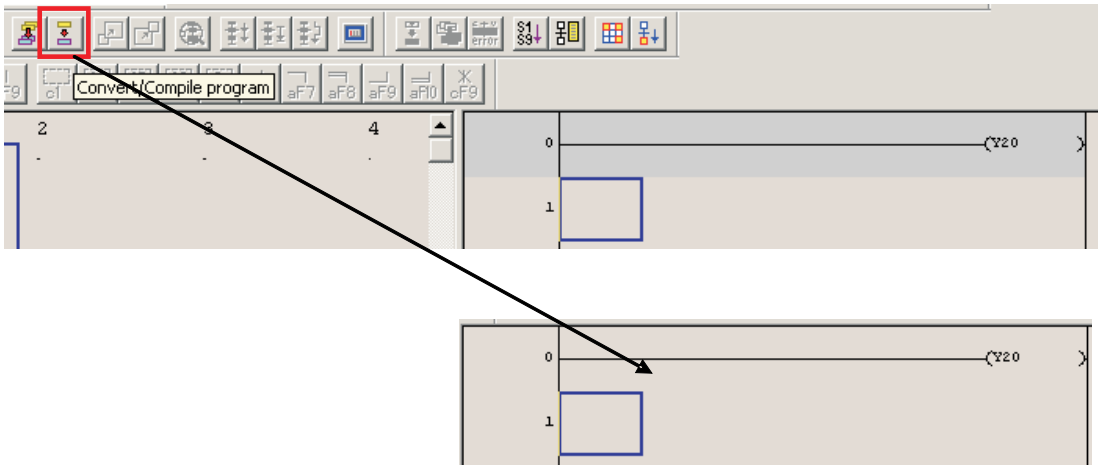
20.1.4 Editing the project

Enter Block

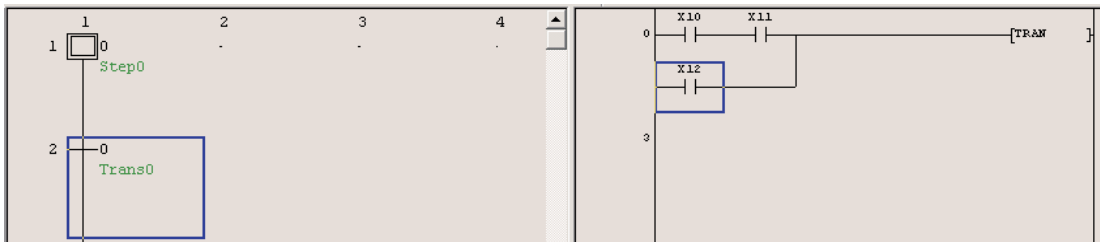


Enter logic for Step

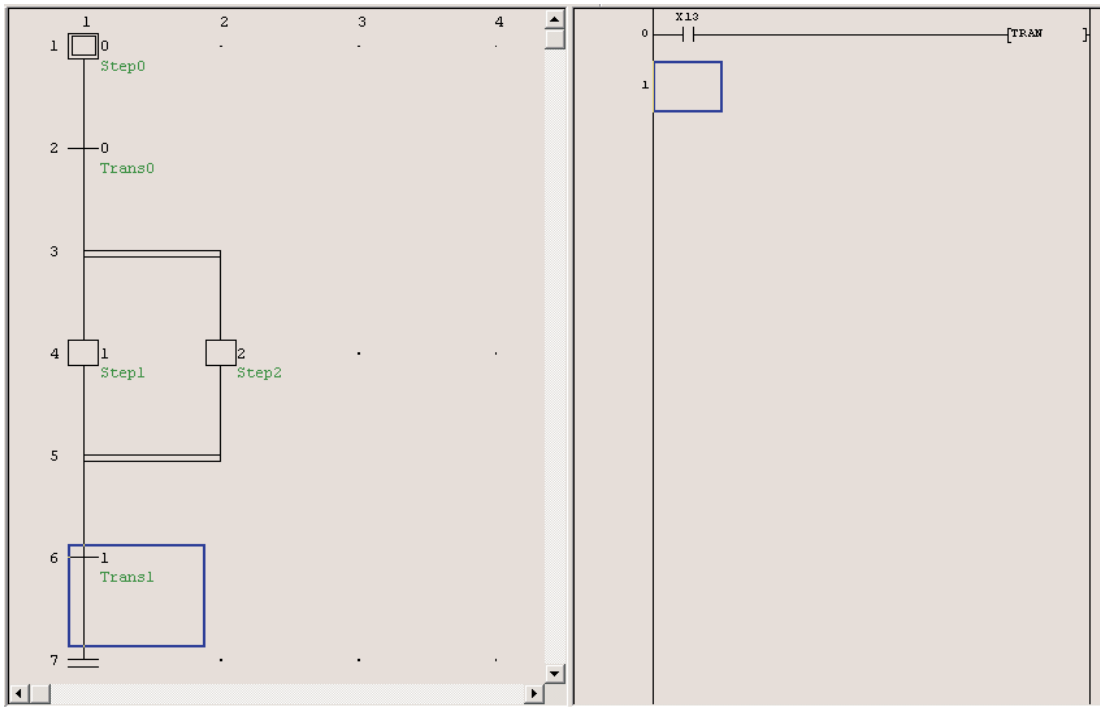
After the condition has been entered, the condition is displayed in dark grey and has to be converted.



Condition of transition



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

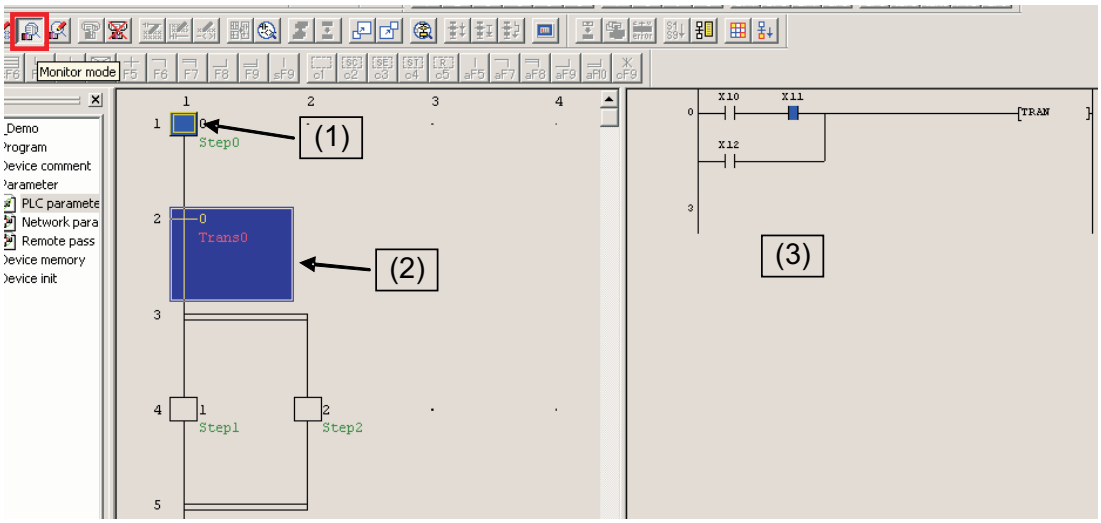


20.1.5 Transfer project

Before the project can be transferred into the PLC, all blocks have to be converted and then they can be written.

20.1.6 Monitor project

The status of the block can be observed by switching on the ***Monitor mode***. The active step (1) is marked with blue background. The condition of the selected transition (2) is shown in the transition edit area (3).



21 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 several instruction steps of program memory in the Q-Series 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 contact Counter Off.

The following example programs illustrate various counter configurations and applications.

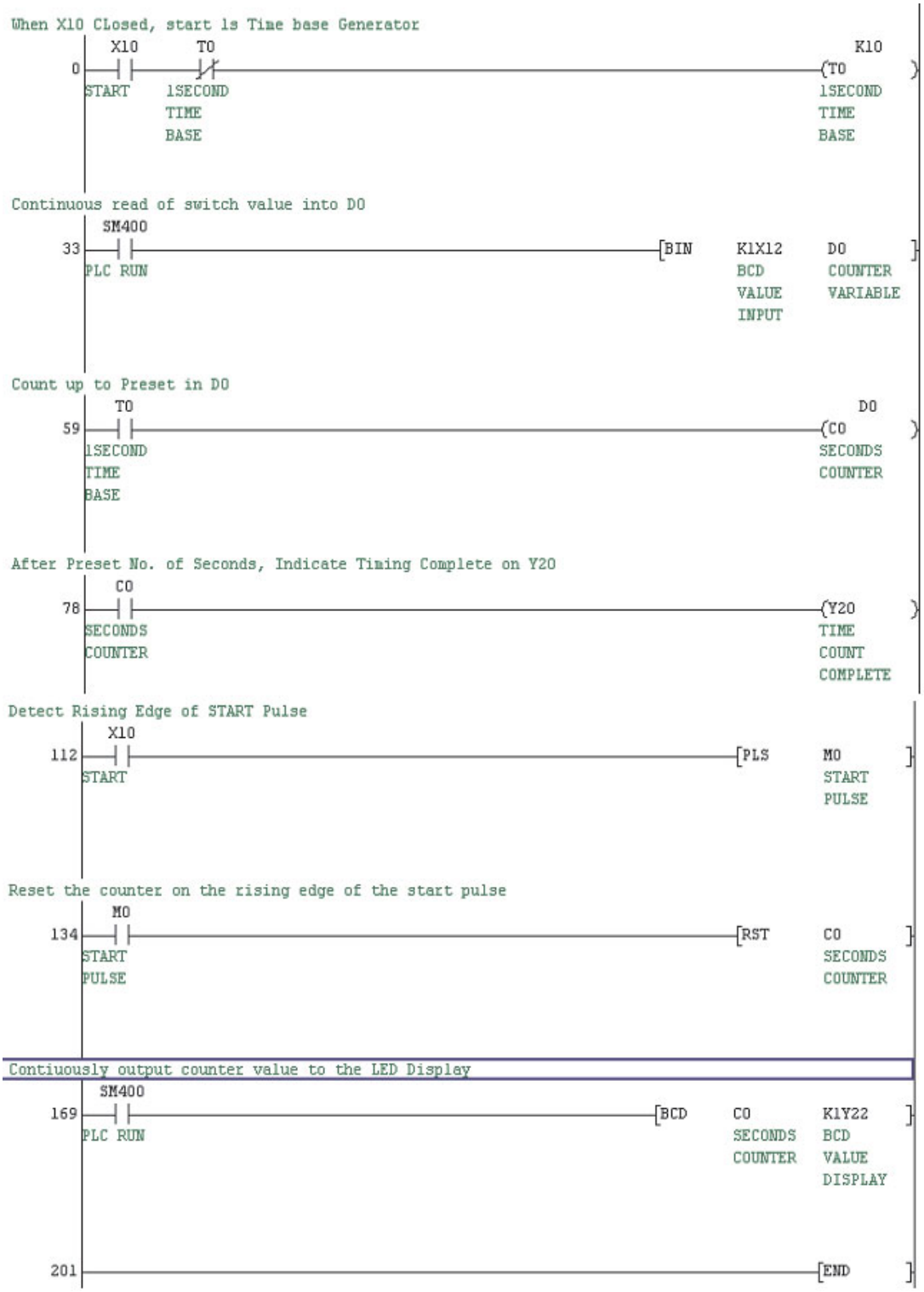
21.1 Programm Example – COUNT DELAY

The following example program - COUNT DELAY, demonstrates how a counter can be used to produce an extended time delay. Entering Special Functional commands the following procedures should be followed:

- To enter -[PLS M0]- enter the following:
  - pls <space>
  - m0 <enter>.
- Use the same procedure for -[RST C0]- i.e.
  - rst <space>.
  - c0 <enter>.
- For the BCD command –[BCD C0 K2Y18]- enter the following:
  - bcd <space>
  - c0 <space>
  - k2y18 <enter>
- For the BIN command –[BIN K2X8 D0]- enter the following:
  - bin <space>
  - k2x8 <space>
  - d0 <enter>

NB: Pressing the [ ] 8 button to open the square brackets before entering the text for the command is optional.

Ladder Diagram – COUNT DELAY



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.

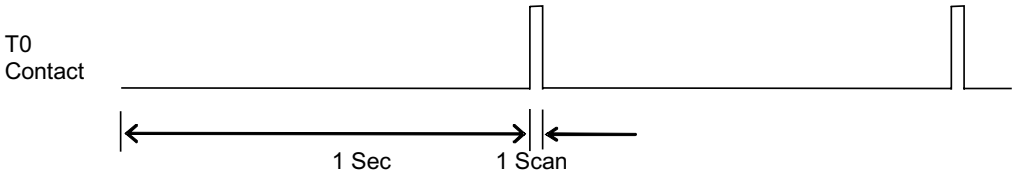
  This 'cut-throat' timer circuit is effectively a pulse oscillator, whose contacts momentarily operate every 1 second.

- Line 33

  The SM400 (Continuously On) contact drives the BIN command. This converts the BCD value represented by 8 Bits (k2); 2 groups of 4 Bits (Nibbles) representing 10s and 1s, starting at X8 into a True Binary Value into Data Register D0.

- Line 55

  With the momentary closure of the normally open contacts of T0, a Count Pulse is sent to Counter C0 every 1 second.



- Line 85

  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 Y10.Therefore, the circuit gives an output signal on Y10, 15 seconds after the Input X0 closes. Hence the circuit can be considered as an extended Timer.

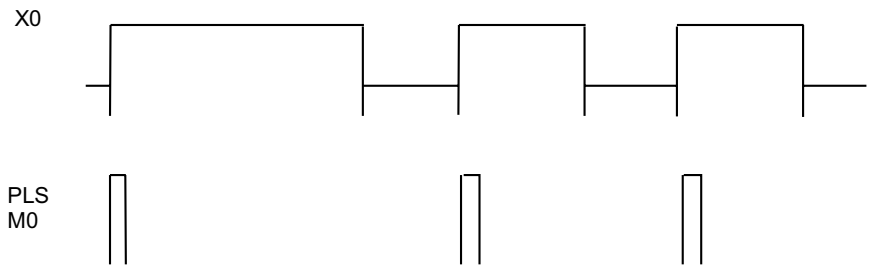
- Line 119

  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.

- Line 141

  The following, are the waveforms associated with the PLS circuit.



  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.

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

- Line 175

  The SM400 (Continuously On) contact drives the BCD command. This converts the True Binary value in C0 into a 2 digit BCD value represented as 2 groups (K2) of 4 Bits (Nibbles) representing 10s and 1s, starting at Y18. This BCD pattern is used to drive a 2 digit – 7 Segment display.

22 FROM / TO INSTRUCTIONS

22.1 Special Function Modules

FROM/TO Instructions are used for Special Function Modules. What are Special Function Modules?

Special function module types

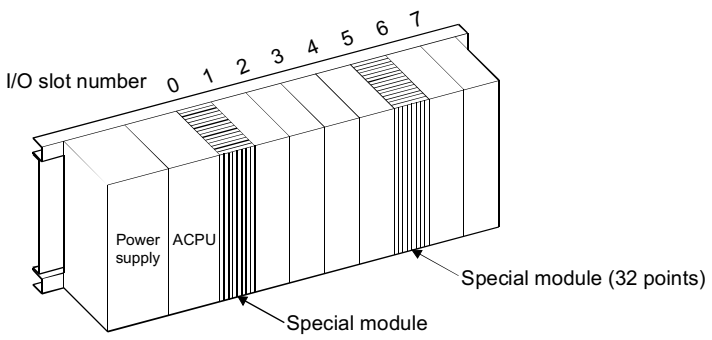
Special function modules perform operations which cannot be implemented by the PLC CPU or functions that are not appropriate to be carried out in the application program code. Therefore, special function modules having required functions are selected and used for different purposes. The following table lists some module examples:

Name	I/O Points Occupied	Function	Current Consumption
Analog-to-digital converter module (Q68AD)	32 points	Input module for converting from 0-20mA to 0-4000 or from 0-±10V to 0-±4000	5VDC 0.63A
Digital-to-analogue converter module (Q62DA)	32 points	Input module for converting from 0-4000 to 0-20mA or from 0-±4000 to 0-±10V	5VDC 0.33A 24VDC 0.12A
High speed counter module (QD61)	32 points	Pulse input, max. 50kpps, 1-phase, 2-phase, 0 to 16,777,215 count (reversible) Allows high speed count which cannot be done by the Q-Series PLC CPU counter C.	5VDC 0.3A

Special function modules used with CPU

Special function modules may be loaded into any I/O slot of the main base and extension base. Note that I/O modules and special function modules consuming higher internal current should be used on the main base or on the extension base (Q65B, Q68B) which requires a power supply module. If special function modules are used on the extension base (Q58B, Q55B) without a power supply module, 5V power is supplied by the power supply module of the main base. Therefore it is necessary to calculate the current consumption and voltage drop (due to extension cable resistance) and select the power supply for the main base.

22.1.1 Loading Special Modules



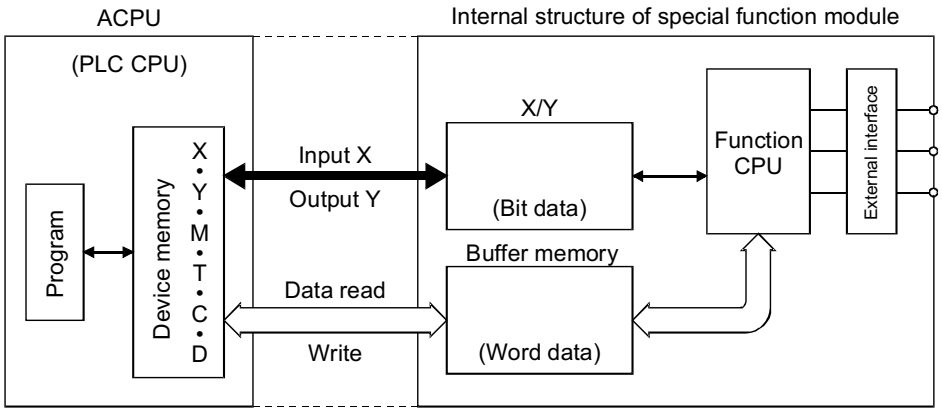
[Numbers used] 32 points from X, Y0 to 1F

22.1.2 Data transfer between Special Function Module and CPU

Roughly classified, two different data are transferred:

- Bit data – Signals using inputs, outputs X/Y
- Word data – 16 or 32 bit data

The following figure shows the internal structure of Special Function Modules:

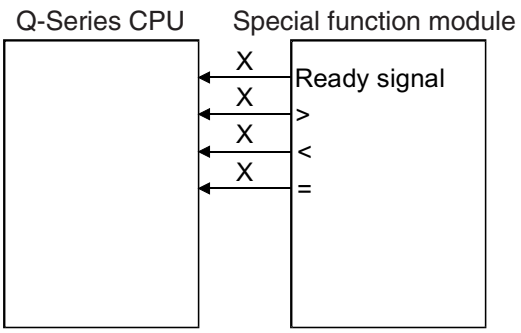


22.1.3 I/O signals 'To and From' the CPU

Among signals transferred between the Q-Series Series CPU and special function module, bit-wise data uses inputs (X) and outputs (Y).

X and Y are different from external inputs and outputs and indicate those used in a sequence program as signals unique to the special function module. The I/O numbers used are numbers assigned according to the special function module physical position in the rack.

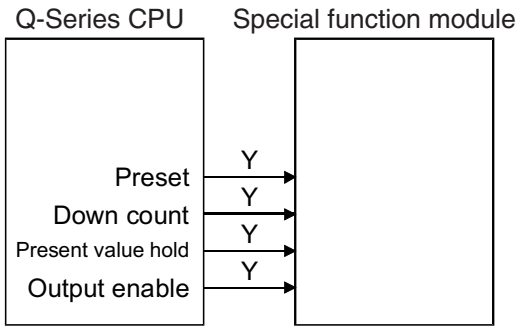
[X]



Xs used in the sequence program are signals generated and input by the special function module to the Q-Series CPU. They are used as contacts in the program, for example:

- Ready signal:
  This signal is input to the Q-Series CPU to indicate that the special function module is ready to operate properly at power on.
- Comparison results:
  The high speed counter module compares a count input with a set value and inputs a magnitude comparison result (>, <, =) to the Q-Series CPU.

[Y]

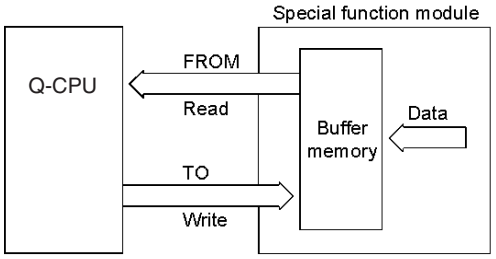


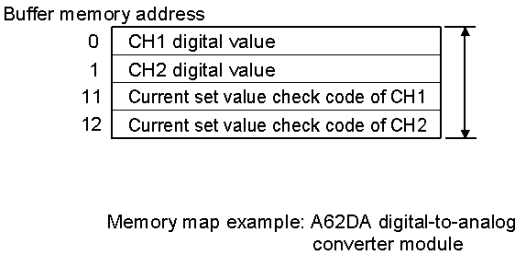
SET, RST and OUT-Y used in the sequence program are signals generated and output by the Q-Series CPU to the special function module. They are used as coils or contacts in the program.

- The high speed counter module outputs set value presetting, down counting and other commands.
- The digital-to-analogue converter module outputs an enable (output enable) command to output a converted analogue value to the outside.

22.1.4 Data transfer between CPU and special function module

Data is transferred in multiples of 16 or 32 bits. Hence, the special function modules have buffer memory to store data.





Memory map example: A62DA digital-to-analog converter module

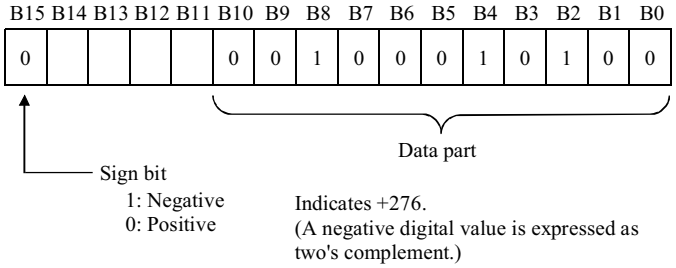
22.1.5 Buffer Memory

The buffer memory can be accessed by the Q-Series CPU. Some modules allow data to be written from a peripheral device to the buffer memory via an interface. The buffer memory is RAM.

The buffer memory of each special function module has its unique addresses in multiples of one word (16 bits).

Addresses start at 0 and are specified to read and write data. The minimum unit is one word. 17 to 32-bit data are handled as two words (32 bits).

Example for Buffer Memory Data:



The above diagram shows an example of a 16-bit digital value written from the Q-Series CPU to the digital-to-analogue converter module. Set a digital value of -2048 to +2047 in 16-bit signed binary.

The FROM (read) and TO (write) instructions are used to access the buffer memory.

NOTE | These instructions may only be used for the buffer memory.

22.2 Buffer Memory Accessing Instructions

CAUTION*:*

Notes on read and write programs:

The buffer memory has a read-only area at some of its addresses. Data must not be written to this area. Otherwise, data is written by the special function module any time and buffer memory data will be corrupted, and sometimes, the special function module may become faulty.

The buffer memory area within the address range should only be accessed. As the area outside the address range may be used by the OS, proper operation cannot be guaranteed if such area is accessed.

The DFRO, DFROP, DTO or DTOP must be used to handle data of 17 or more bits.

Up to 2000 pieces of data may be specified per instruction to read or write 16-bit data or 32-bit data in the range where the watchdog timer error will not occur.

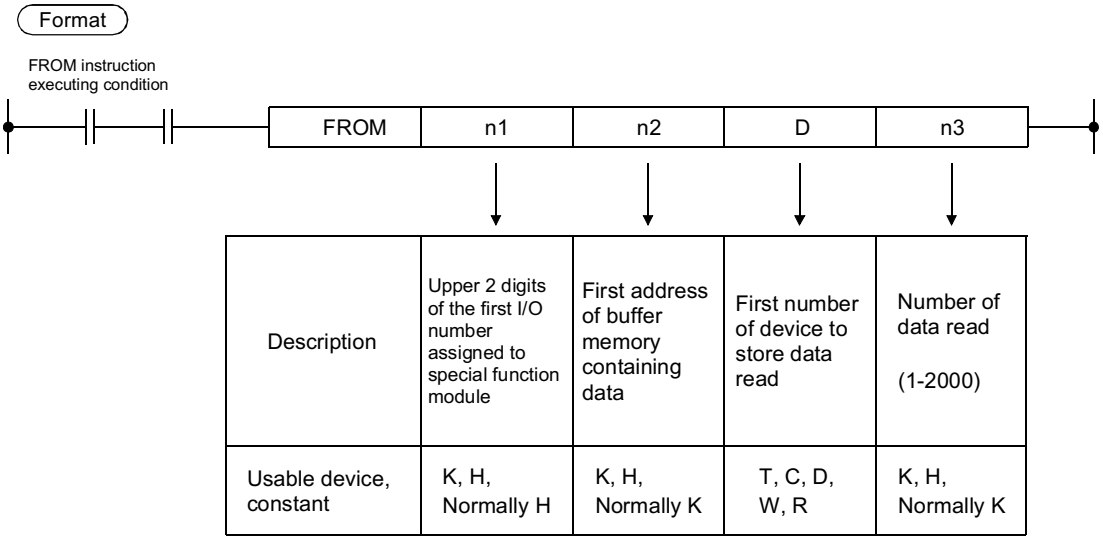
22.2.1 Read Buffer Memory (FROM)

The FROM instruction reads the contents of the special function module buffer memory to the Q-Series CPU. Data read can be stored into any of word devices D, T, C, W, and R of the Q-Series CPU.

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

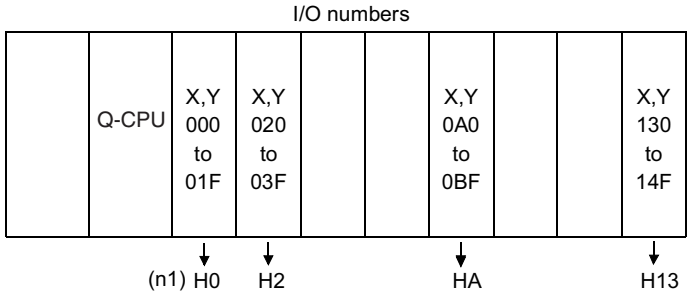
Execution	16-Bit Data (1-Word Data)	32-Bit Data (2-Word Data)
Executed any time when condition switches on	FROM	DFRO
Executed on leading edge of condition	FROMP	DFROP

Read Instruction FROM



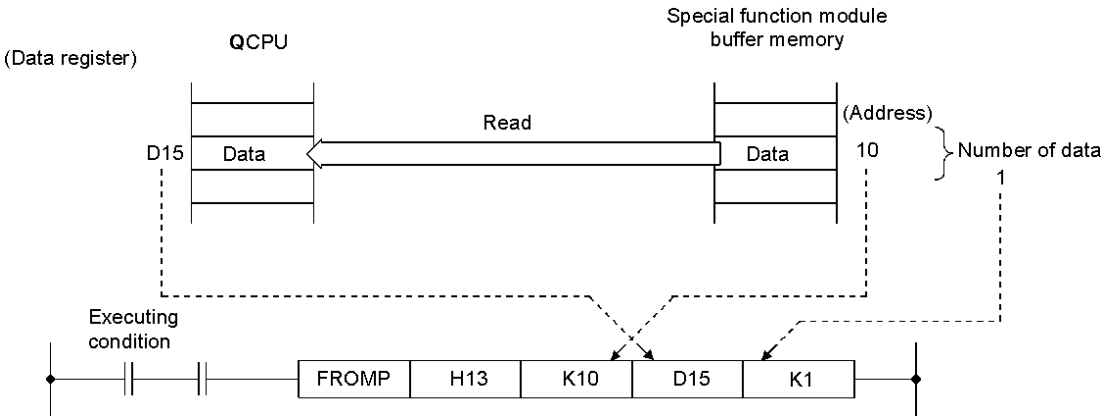
How to Set n1

The following diagram shows how to set the n1 part of the FROM instruction. (This also applies to n1 of TO instruction)

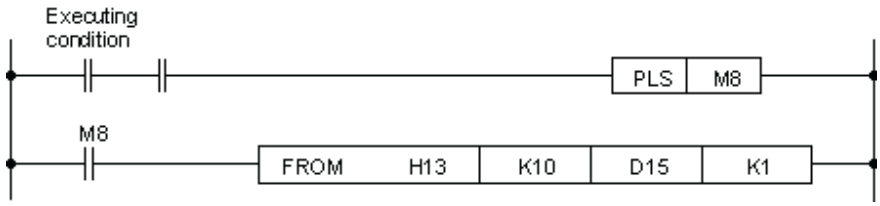


Examples of reading 16-bit data

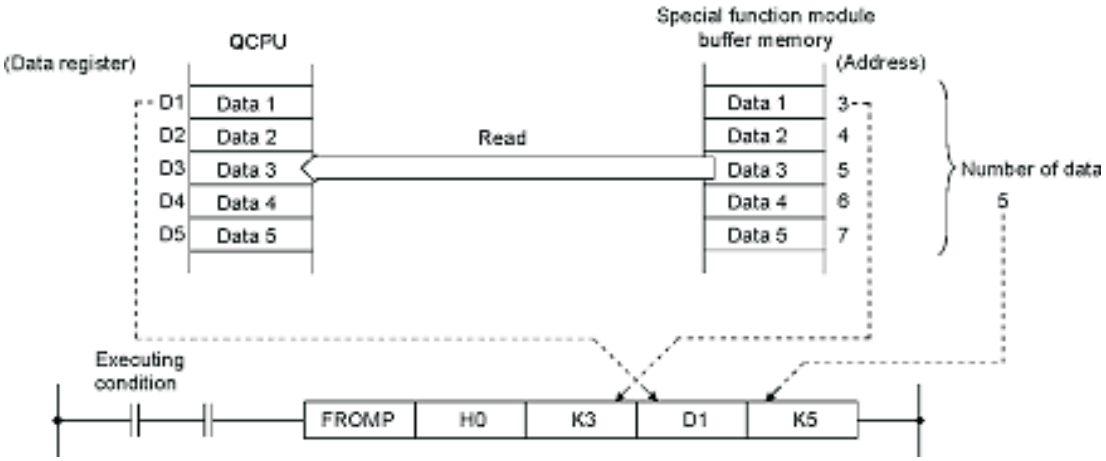
Example 1 To assign a special function module to I/O numbers X130-14F and Y130-14F and read data of only one word from address 10 of the buffer memory to D15.



Example 2 To use a pulsed relay to read the above data.

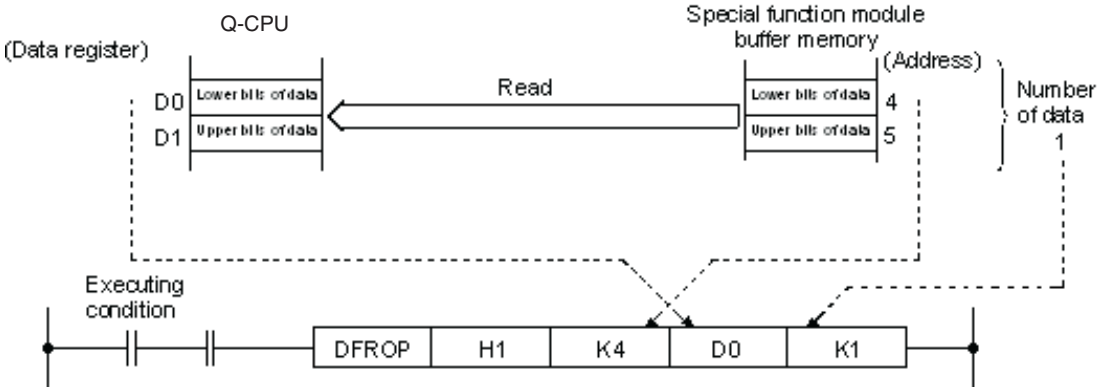


Example 3 To assign a special function module to I/O numbers X00-1F and Y00-1F and read data of five words from address 3 of the buffer memory to D1-D5.

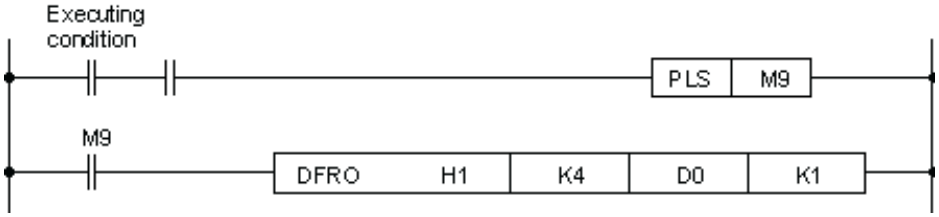


Examples of reading 32-bit data

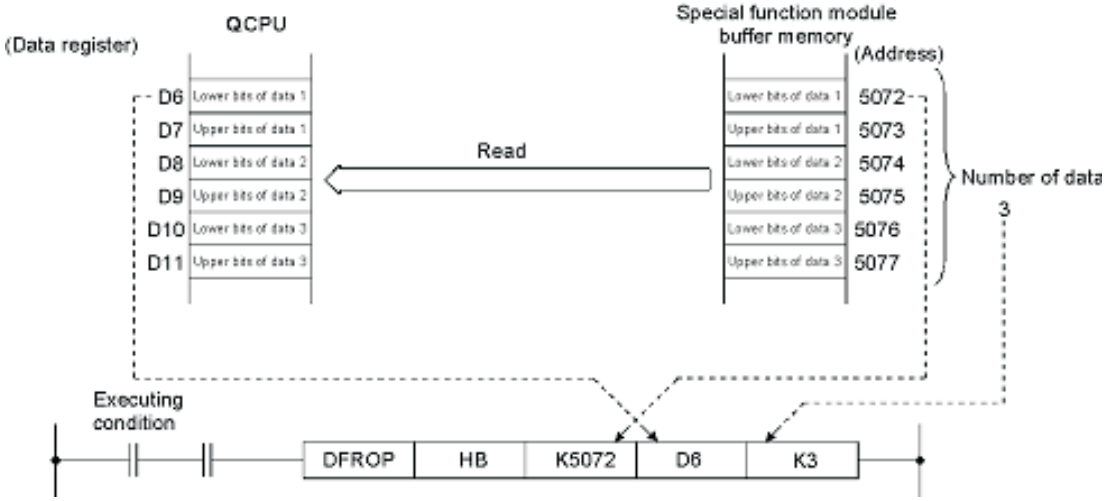
Example 1 To assign a special function module to I/O numbers X10-2F and Y10-2F and read 1 piece of two-word (32-bit) data from address 4 of the buffer memory to D0 and D1.



Example 2 To use a pulsed relay to read the above data.



Example 3 To assign a special function module to I/O numbers XB0-CF and YB0-CF and read 3 pieces of two-word (32-bit) data from addresses 5072-5077 of the buffer memory to D6-D11.



22.2.2 Write to Buffer Memory (TO)

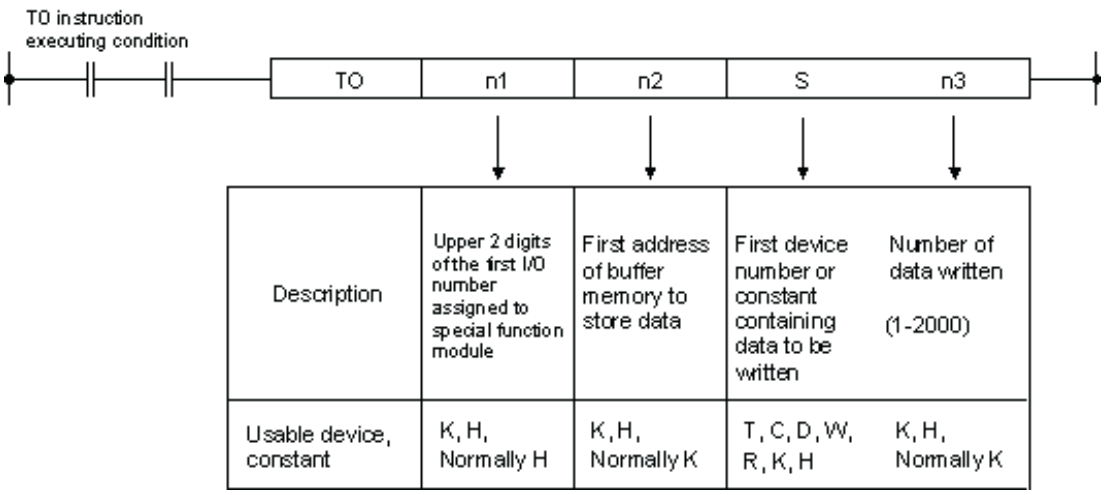
The TO instruction writes data from the Q-Series CPU to the buffer memory of the special function module. Data to be written are those stored in devices D, W, R, T and C of the Q-Series CPU or constant decimals (K□□) and hexadecimals (H□□).

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

Write Instruction TO

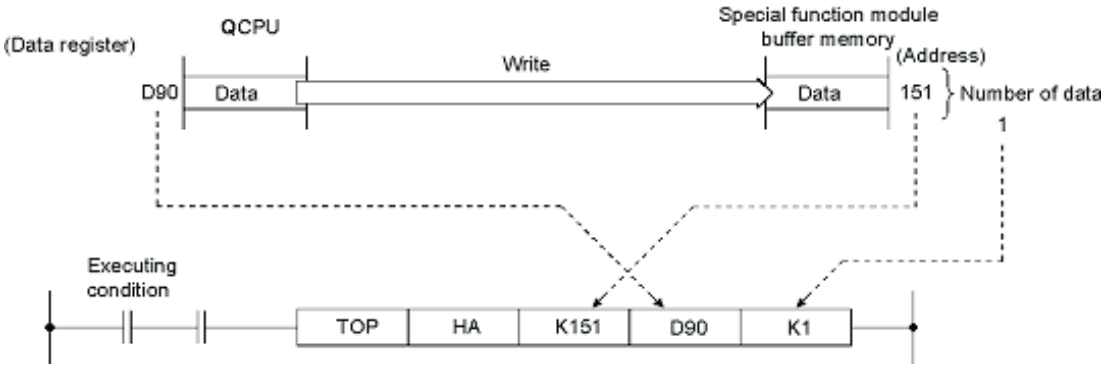
Format



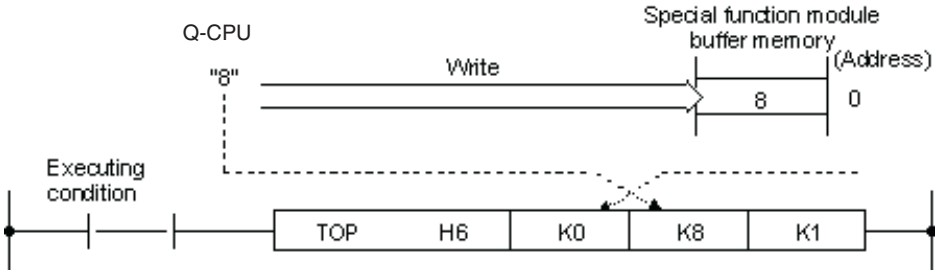
	n1	n2	S	n3
Description	Upper 2 digits of the first I/O number assigned to special function module	First address of buffer memory to store data	First device number or constant containing data to be written	Number of data written (1-2000)
Usable device, constant	K, H, Normally H	K, H, Normally K	T, C, D, W, R, K, H	K, H, Normally K

Examples of writing 16-bit data

Example 1 To assign a special function module to I/O numbers XA0-BF, YA0-BF and write 1-word data stored in data register D90 to address 151 of the buffer memory.

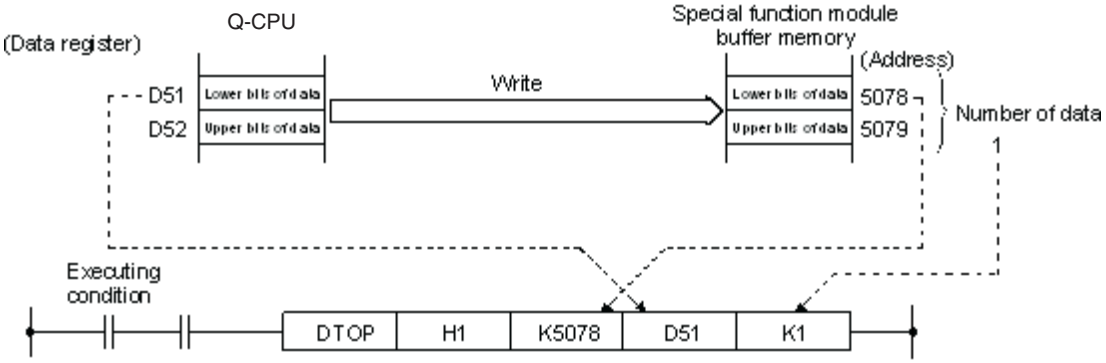


Example 2 To assign a special function module to I/O numbers X60-7F, Y60-7F and write 8 to address 0 of the buffer memory.



Examples of writing 32-bit data

Example 1 To assign a special function module to I/O numbers X10-2F, Y10-2F and write 2-word (32-bit) data stored in data registers D51, 52 to addresses 5078, 5079 of the buffer memory

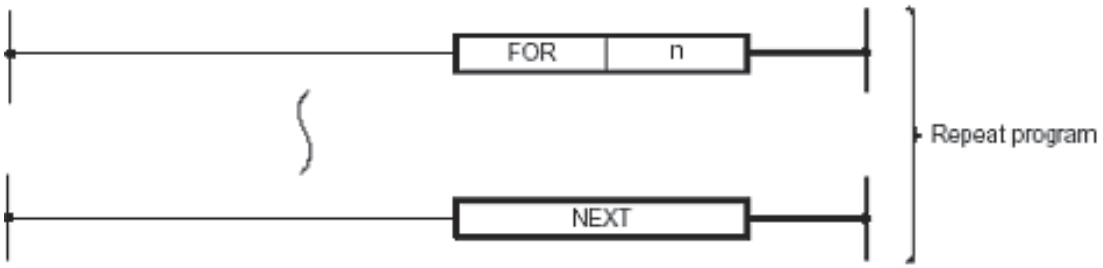


23 FOR – NEXT Loops

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 be 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.

23.1 Operation

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



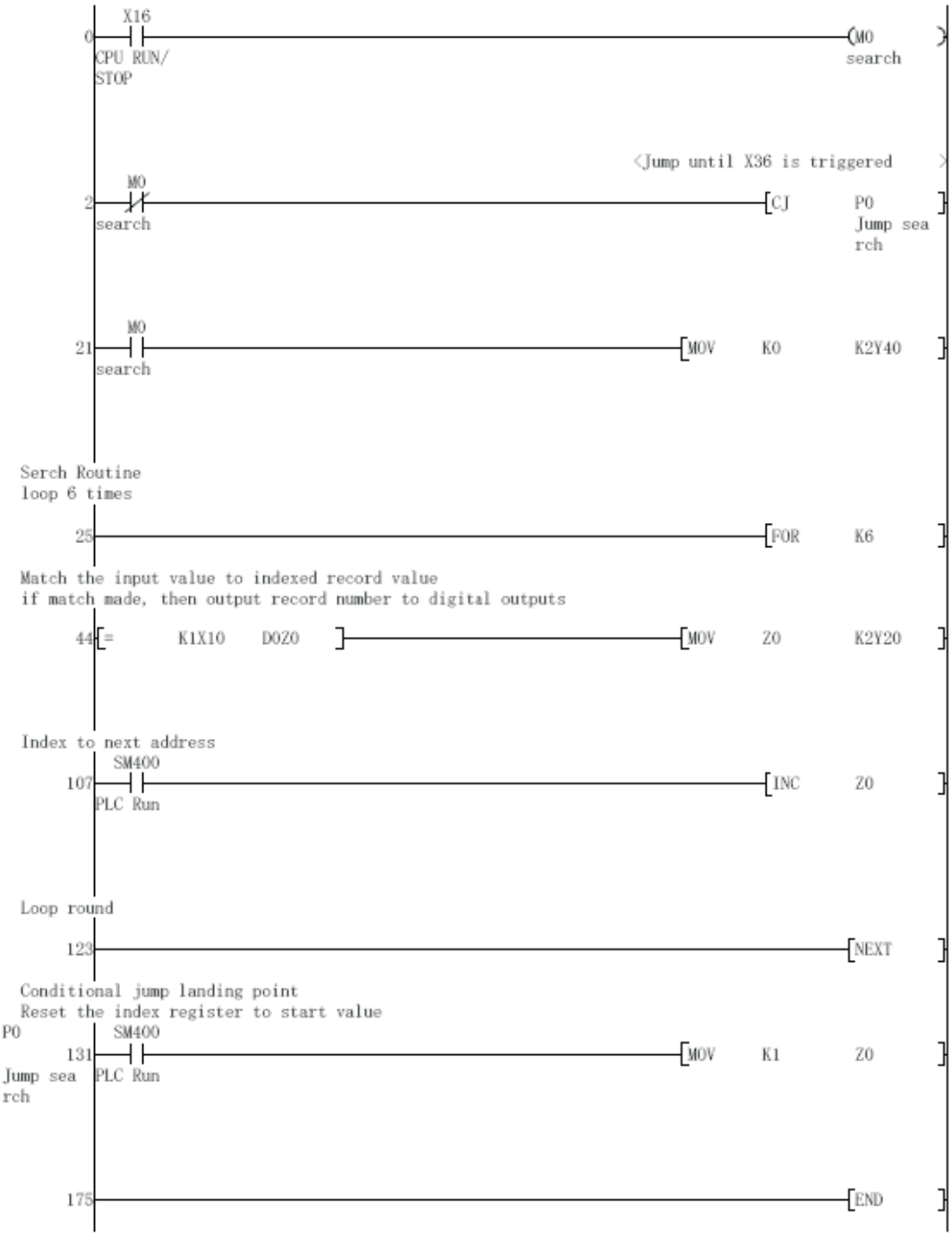
It is common practice to use a Conditional Jump (CJ) to bypass a FOR-NEXT loop if it is not required to be scanned. This will avoid scanning the loop when it is not required, thus minimising the overall program scan time.

23.1.1 Program Example

The following ladder program illustrates the use of a FOR-NEXT loop to search for a specific data value entered BCD coded at the digital inputs from a range of data registers. Input X10 has the value 2^0, X11 has the value 2^1, X12 has the value 2^2 and X13 the value 2^3.

The program returns the record number of the data BCD coded to the digital outputs of the training rig when a match is made.

The program on the next page illustrates the use of ‘FOR-NEXT’, ‘Conditional Jump’, ‘Comparison’ and ‘Index register’ instructions:


X16
0
CPU RUN/
STOP
M0
search
<Jump until X36 is triggered
M0
2
search
CJ P0
Jump sea
rch
M0
21
search
MOV K0 K2Y40
Serch Routine
loop 6 times
25
FOR K6
Match the input value to indexed record value
if match made, then output record number to digital outputs
44
= K1X10 D0Z0
MOV Z0 K2Y20
Index to next address
SM400
107
PLC Run
INC Z0
Loop round
123
NEXT
Conditional jump landing point
Reset the index register to start value
P0
SM400
131
Jump sea
rch
PLC Run
MOV K1 Z0
175
END

23.2 Set up and Monitoring Procedure

Exercise

① Enter the above program example and save it as ‘For-Next1’

② Send the program to the PLC.

③ Before operating the program, fill Data Registers D1-D6 with random values between 1 and 15 decimal, using the ‘Device Test’ feature in GX-Developer (Described in earlier sections).

④ Monitor the ladder program and use ‘Batch Monitor’ in GX Developer to view the contents of the data registers D1-D6.

⑤ Select a valid 2 digit value (corresponding to one of the forced values set in 3. above) between 1 and 15 on with the input switches on the training rig.

⑥ Operate the CPU RUN/STOP 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 – D6, then the readout will produce a value corresponding to the record number (Data register Number) from 1 to 6 of the data match.

23.3 Design Assignment

- Incorporate additional code to the previous program so that an indicator Y25 illuminates for 5 Seconds following a data search to indicate that a data match **has NOT** been made.
- Using one of the special SM registers (Appendix A), further modify the above code to flash the indicator Y25 at a frequency of 1.0Hz (0.5 Second ON/OFF) for the 5 second duration.

24 Ethernet Communications

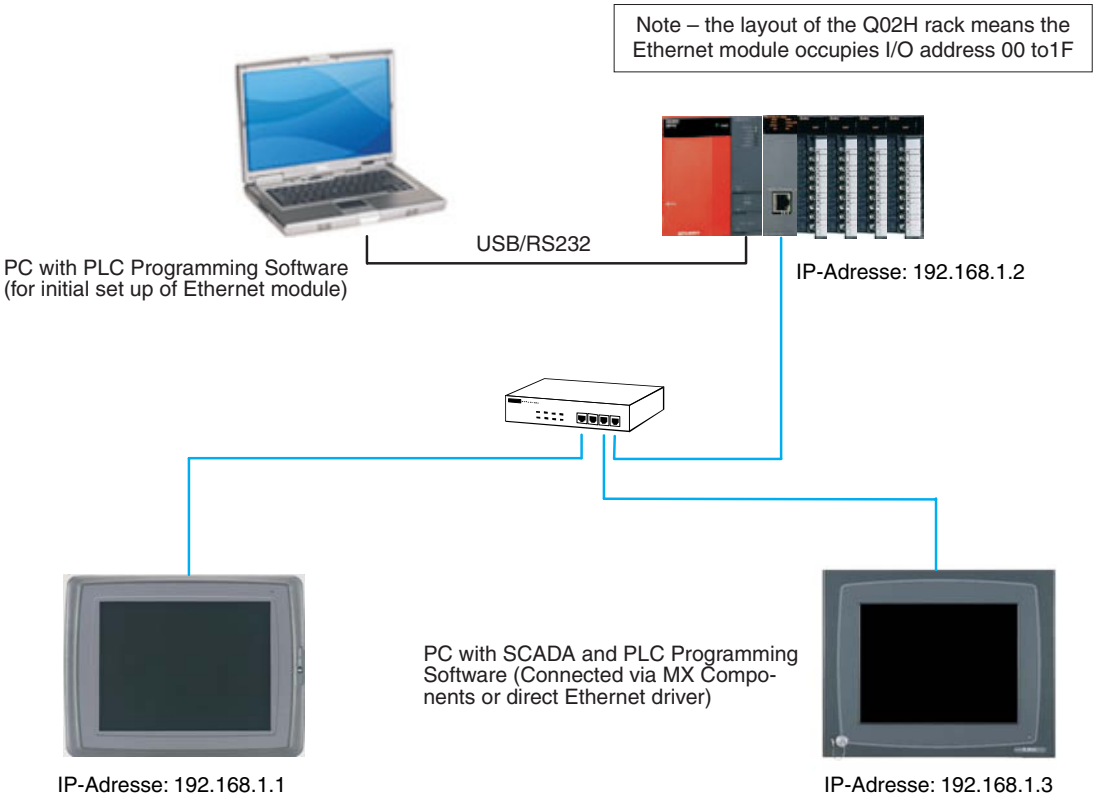
24.1 Configuring Qn Ethernet Module by Parameter

This section provides a step-by-step guide to setting up a QJ71E71 type Ethernet module (to be referred to as 'module' from now on) by parameter setting, GX Developer 8.00 or later.

As an example, this section will show how to set up a module for allowing TCP/IP communications between a Q02H, a SCADA PC and an E1071 HMI. Also shown is how the programming software can be configured to communicate with the Q02H via Ethernet once the settings have been made.

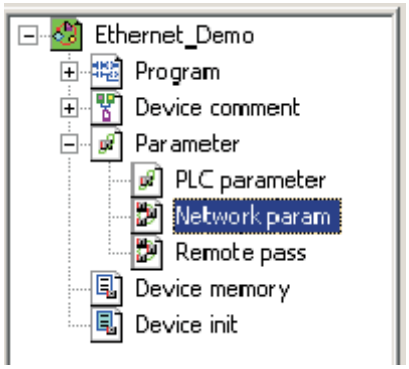
The diagram below shows the layout of the example Ethernet network. Proposed IP addresses are shown in brackets next to the Ethernet nodes.

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.

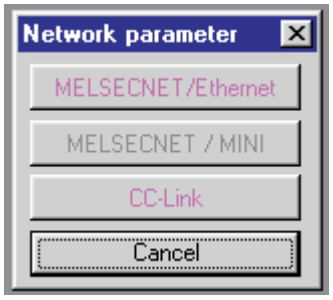


24.1.1 Configuring the PLC (using initial set up PC)

① Using the programming software, call up the ***Network*** Parameter selection box by double clicking on the option highlighted by the arrow.



② When the box has been opened, select ***MELSECNET/Ethernet*** as shown below.

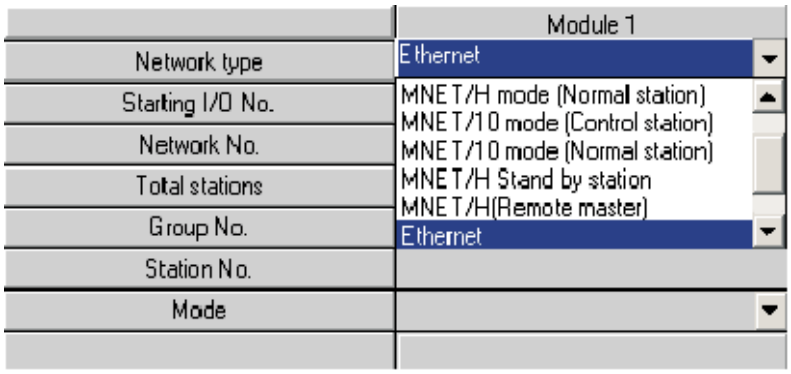


This opens up the dialogue box to allow the Ethernet module to be configured which can be seen below.

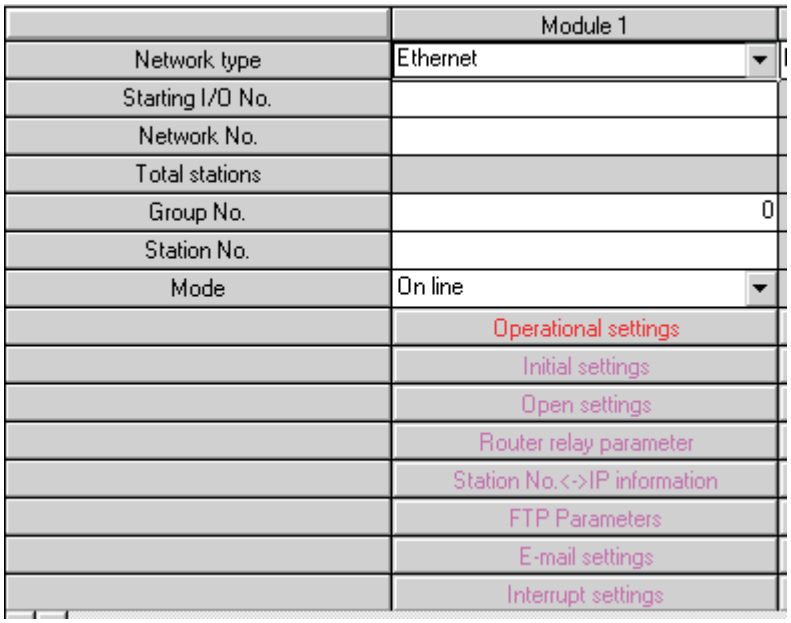
③ In the Network type window, click on the down arrow, to show the available selections:

	Module 1
Network type	None
Starting I/O No.	
Network No.	
Total stations	
Group No.	
Station No.	
Mode	

④ Ethernet is the final option in the list. Select it as shown below:



⑤ The dialogue box now shows the specific setting options for the module. The buttons in the bottom half of the table that are in red are for setting the mandatory parts of the module, those in magenta are optional, and are set as required.



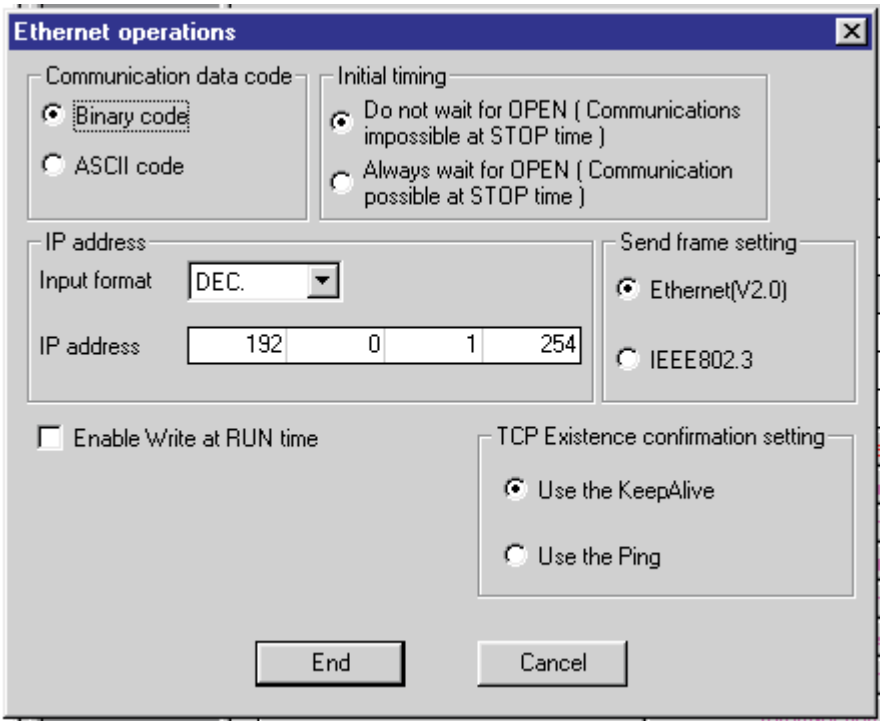
⑥ Click in the boxes in the top half of the table and enter the values as required. The table below shows the settings for the Q02H in the example system described earlier.

	Module 1	
Network type	Ethernet	
Starting I/O No.	0000	
Network No.	1	<— see Note below
Total stations		
Group No.	0	
Station No.	2	<— see Note below
Mode	On line	
	Operational settings	
	Initial settings	
	Open settings	
	Router relay parameter	
	Station No.<->IP information	
	FTP Parameters	
	E-mail settings	
	Interrupt settings	

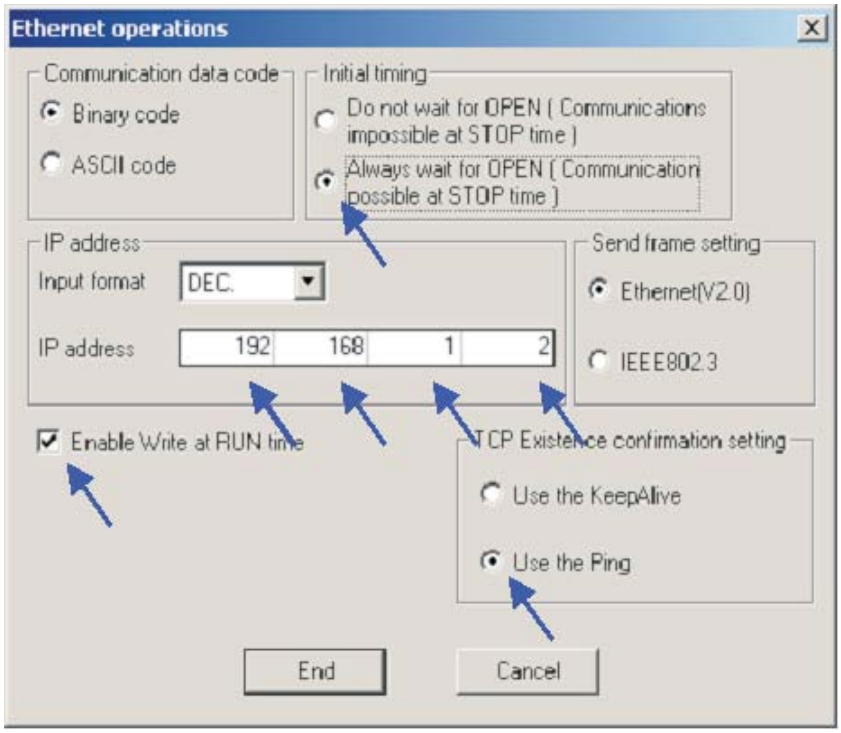
NOTE

The “network number” and “station number” settings are used to identify the module when Qn PLC’s use the Ethernet for Peer-to-Peer communications (not covered in this document). These settings are also used when the programming software is to communicate to the Qn PLC across the Ethernet network. This subject is covered later in the document.

⑦ Next, click on the ***Operational settings*** to bring up the dialogue shown below. The settings already there are the defaults that the programming software applies.



⑧ The dialogue below shows the settings required for the example system described earlier. The arrows highlight the differences for clarity.



⑨ After the settings here are made, click ***End*** to return to the main network parameter setting window. Note that the ***Operational settings*** button has now changed to blue, indicating that changes have been made.

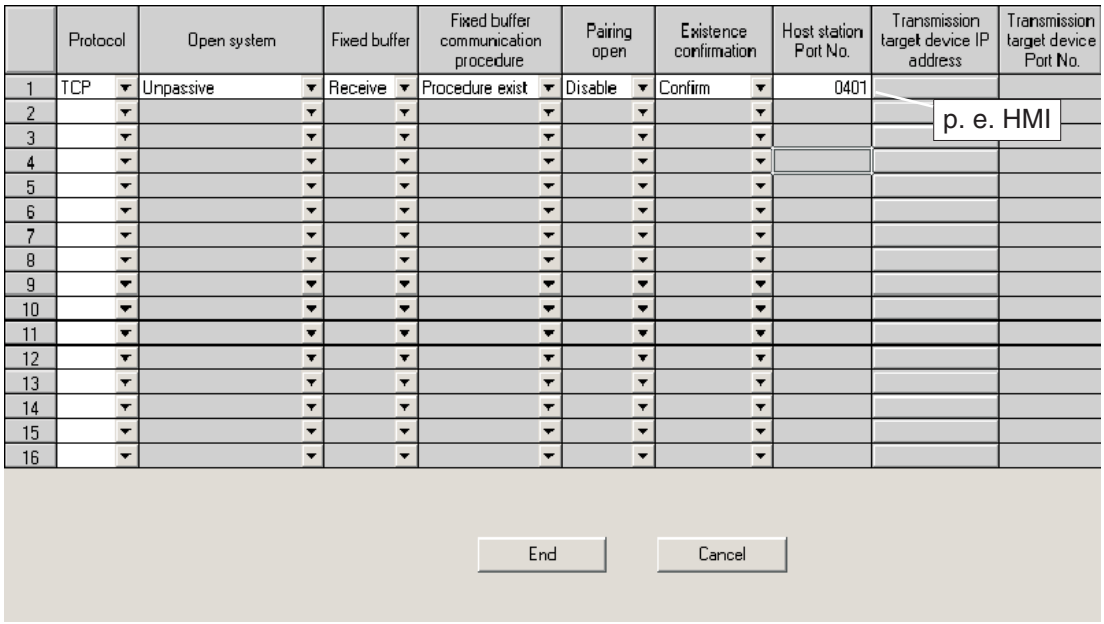
	Module 1
Network type	Ethernet
Starting I/O No.	0000
Network No.	1
Total stations	
Group No.	0
Station No.	2
Mode	On line
	Operational settings
	Initial settings
	Open settings
	Router relay parameter
	Station No.<->IP information
	FTP Parameters
	E-mail settings
	Interrupt settings

Next, click on ***Open settings*** to bring up the following dialogue. This is where the settings for the Scada and HMI will be made.

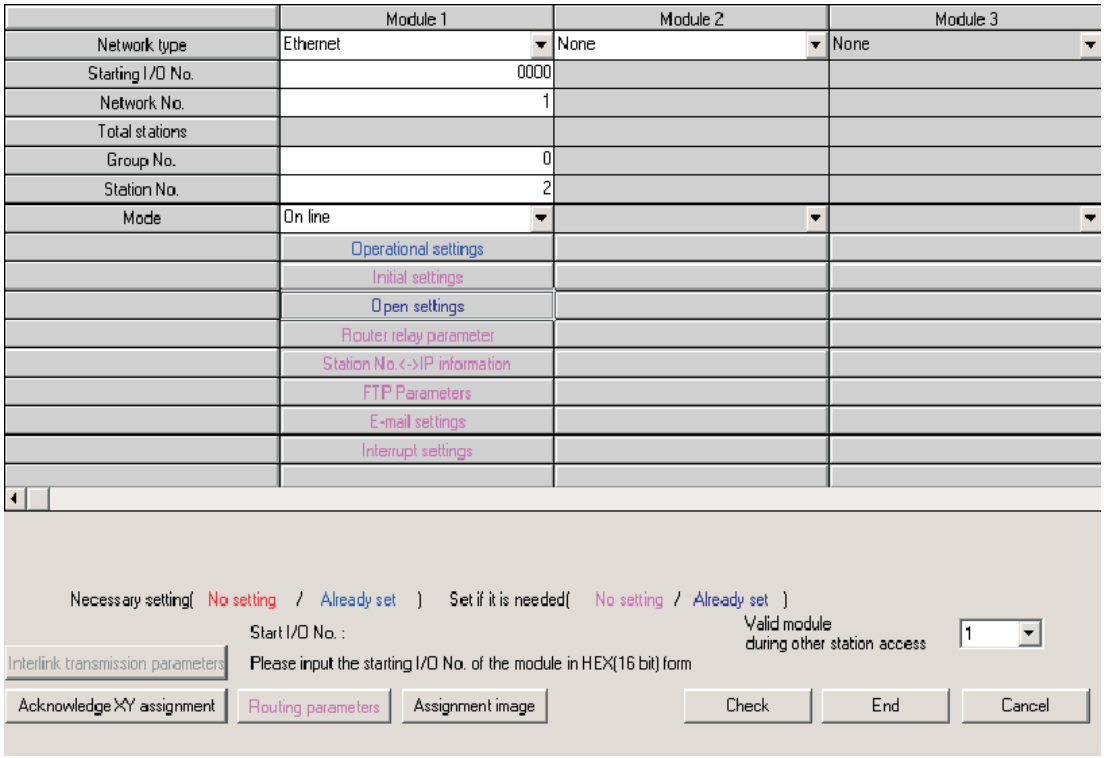
NOTE

There is no need to set anything here, if the Ethernet card is **only** to be used for program monitor/edit using the programming software (as described later).

The dialogue below shows the settings required for communication with both the Scada and the HMI, for the example system described earlier. The settings are made by selecting the required options from the drop-down lists in each window, or typing as required. For a further explanation of these settings, refer to the glossary at the end of this chapter.



When the settings have been made, click ***End*** to return to the main network parameter setting window.

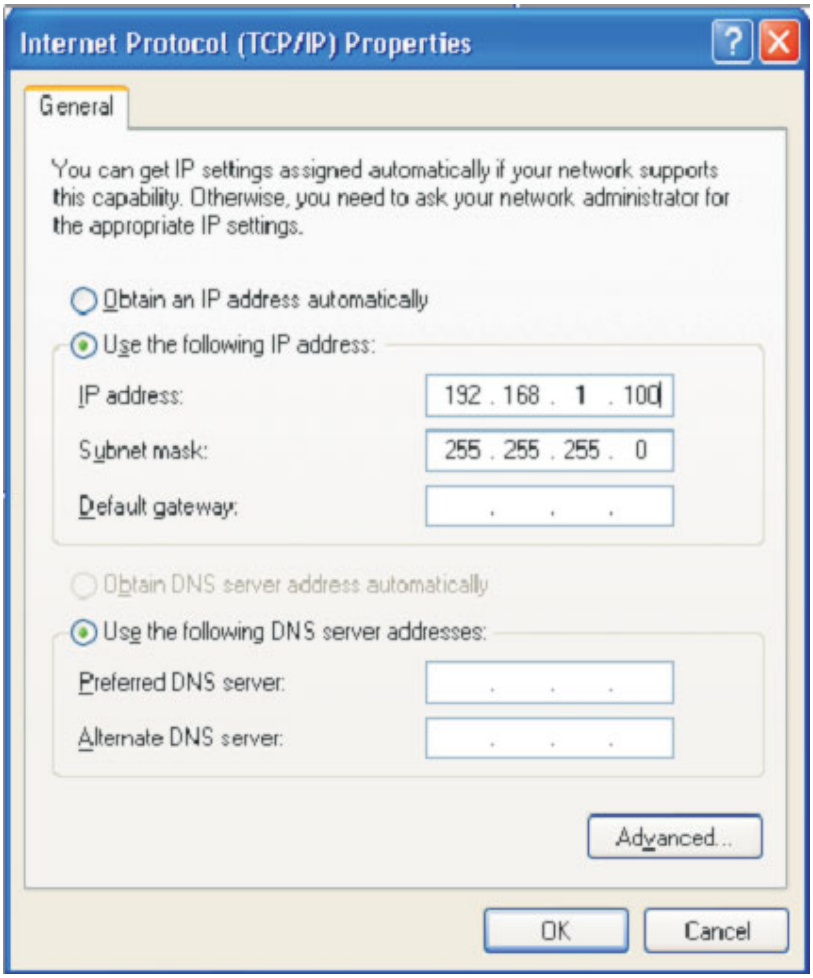


No more setting is required here for communications with the Scada or the HMI.

Click ***End*** to check and close the main network parameter setting dialogue. These settings will be sent to the PLC next time the parameters are downloaded.

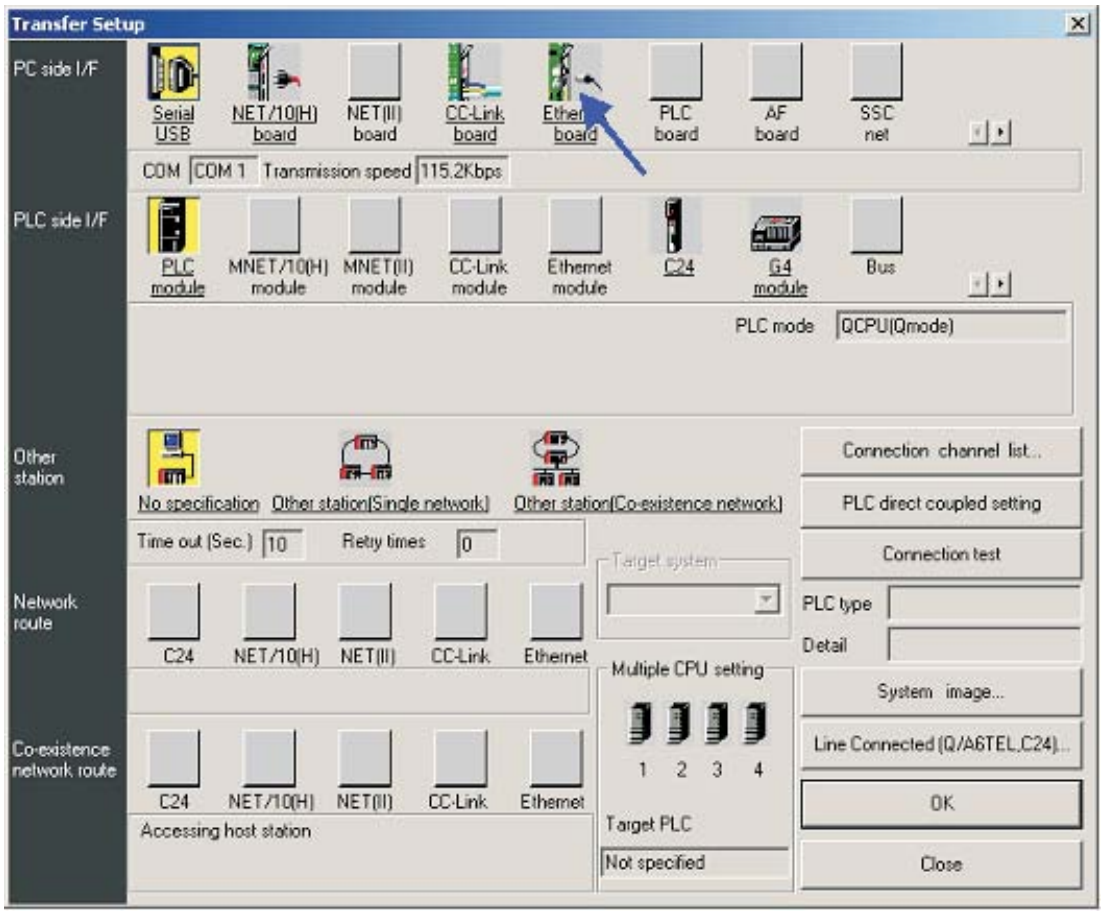
24.2 Configuring the PC on the Ethernet

① 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. Please note that after changing IP address, the PC may require a restart.



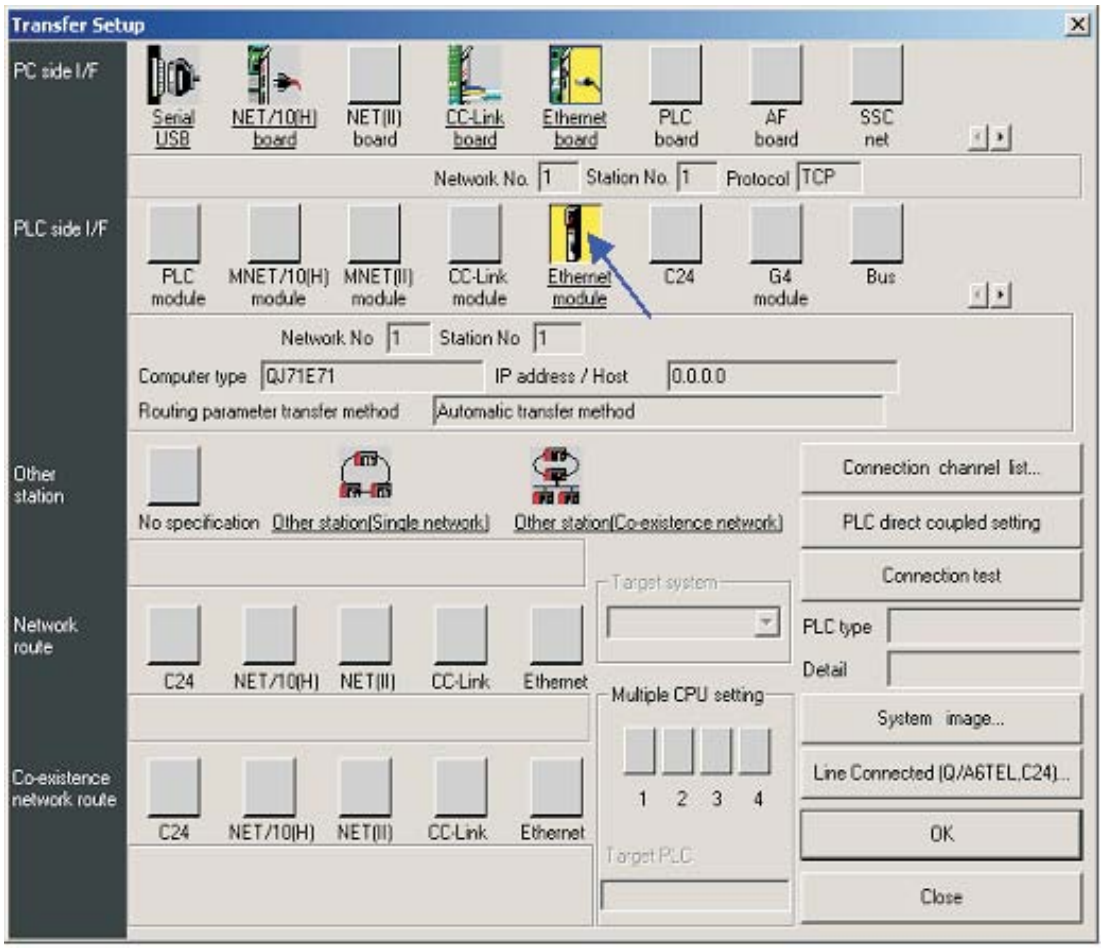
24.3 Configuring GX Developer to access the PLC on Ethernet

① Open the connection settings dialogue as shown



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 The ***PC Side I/F*** should default to Network No. = 1, Station No = 1 and Protocol = TCP as shown above. If it does NOT show this, then double click on ***Ethernet board*** and make these settings in the appropriate places

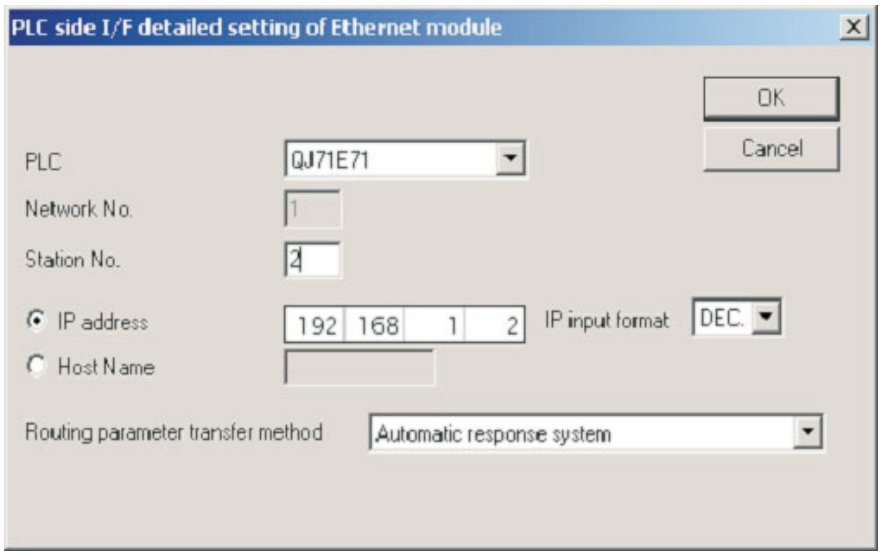


④ Next, double click on ***Ethernet module*** under ***PLC side I/F*** as shown above. This will open up the dialogue to allow the selection of the PLC to be communicated with over the Ethernet. Enter the settings shown, as these were the settings put into the PLC earlier. (refer back to parts 6 and 7 in section 24.1.1)

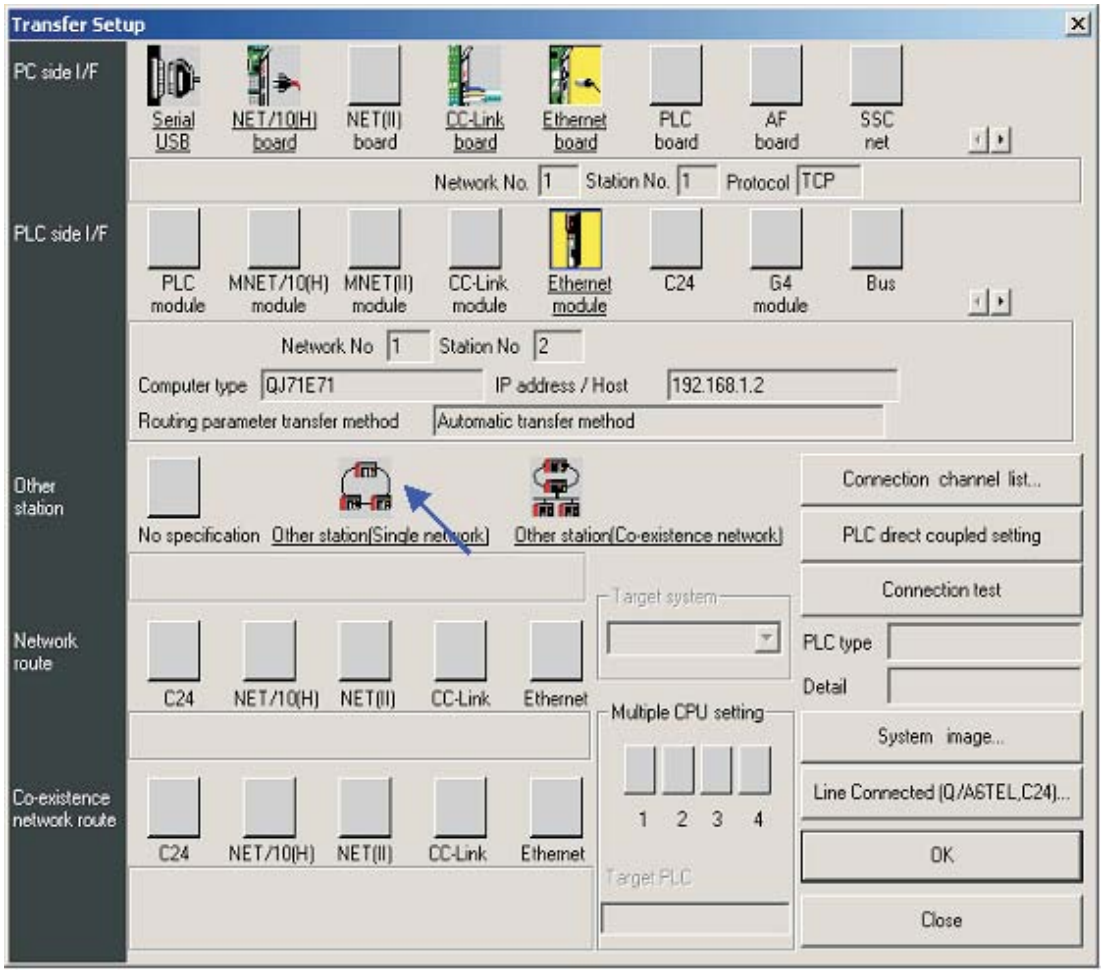
⑤ Click ***OK*** when done.

NOTE

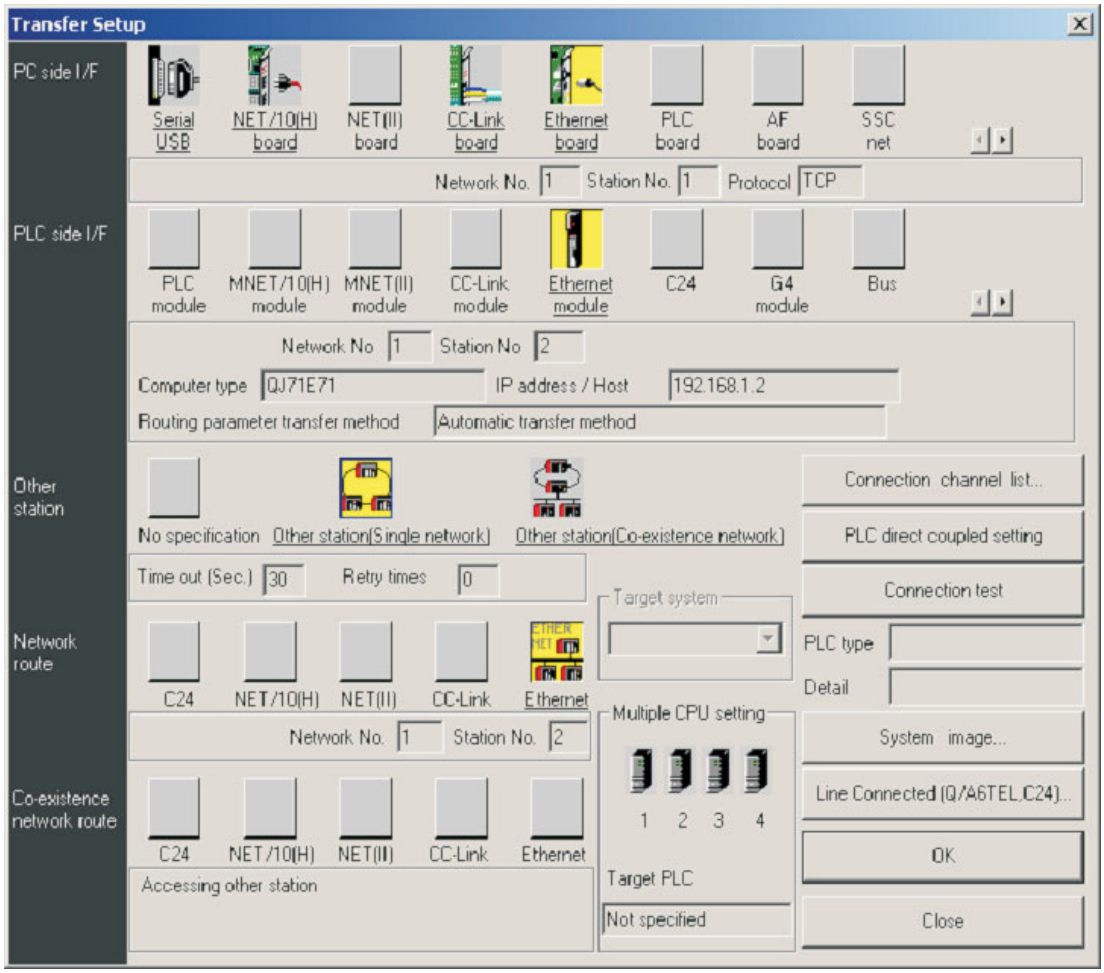
There is no need to specify a port number, as the programming software will use a MELSOFT Protocol dedicated port by default.



⑥ Next, single click on ***Other station (Single network)*** as shown below.

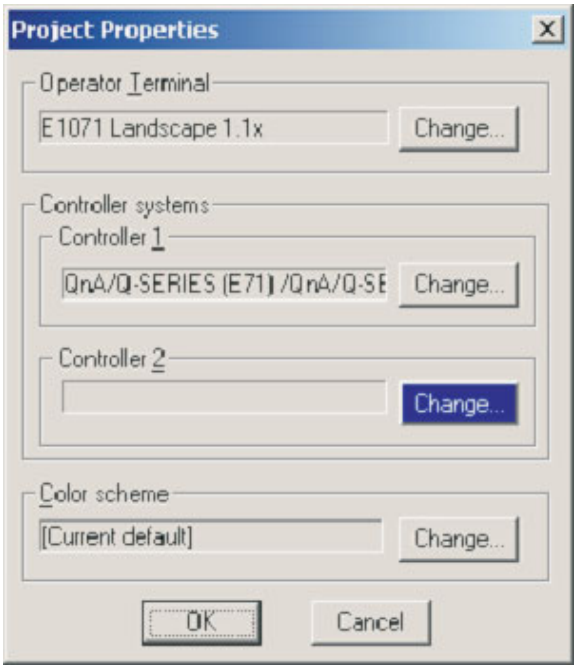


⑦ This will complete the setting, making the dialogue look as shown below. Click Connection test to confirm the settings are correct. Then click OK when finished.

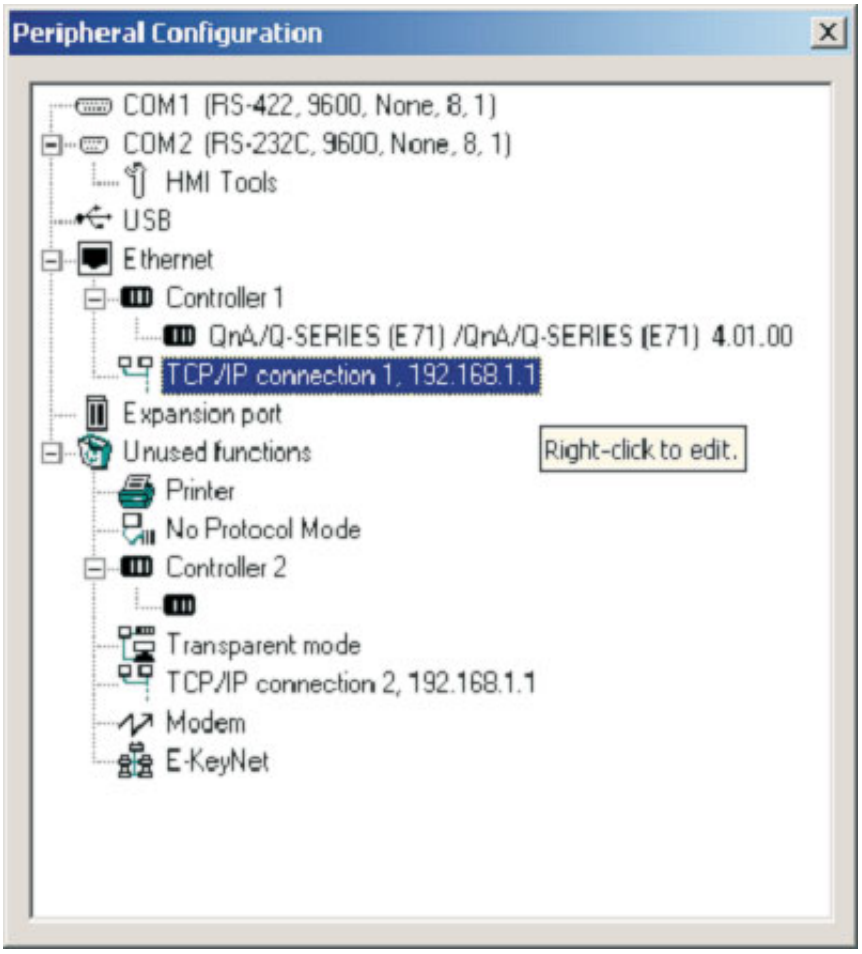


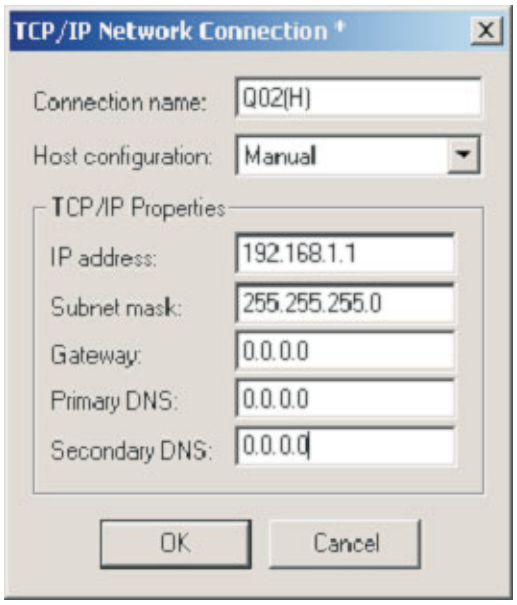
24.4 Setting up the HMI

① The E-Designer project for the example system needs to have the following settings.

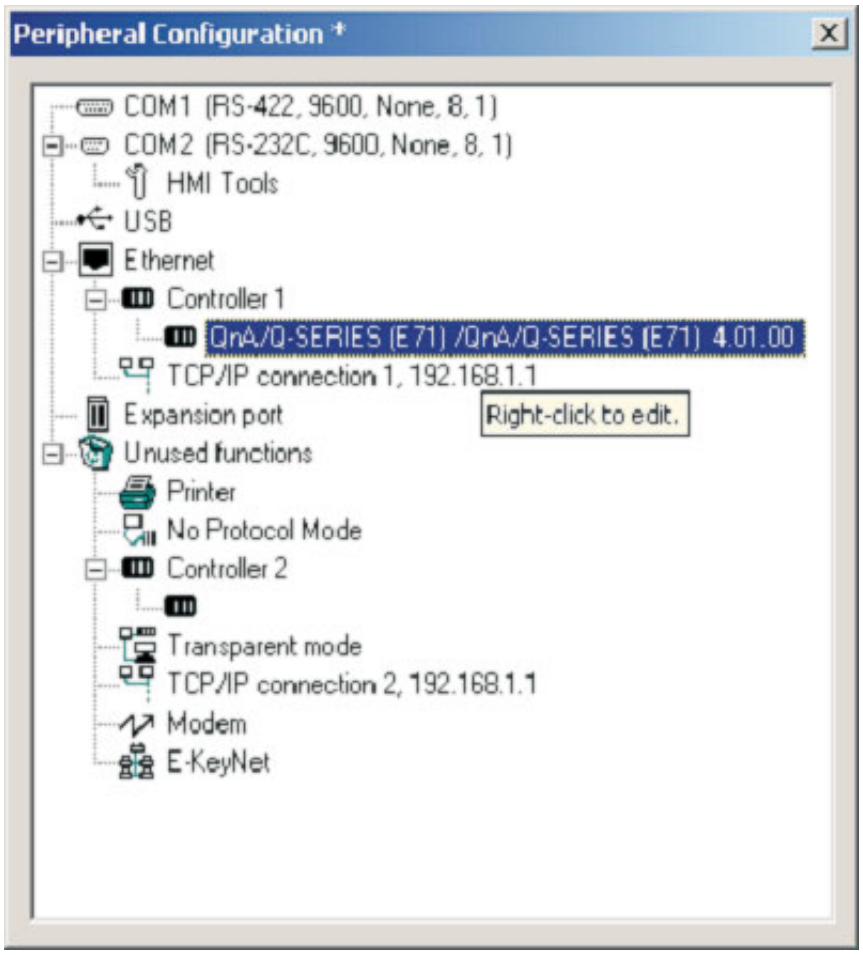


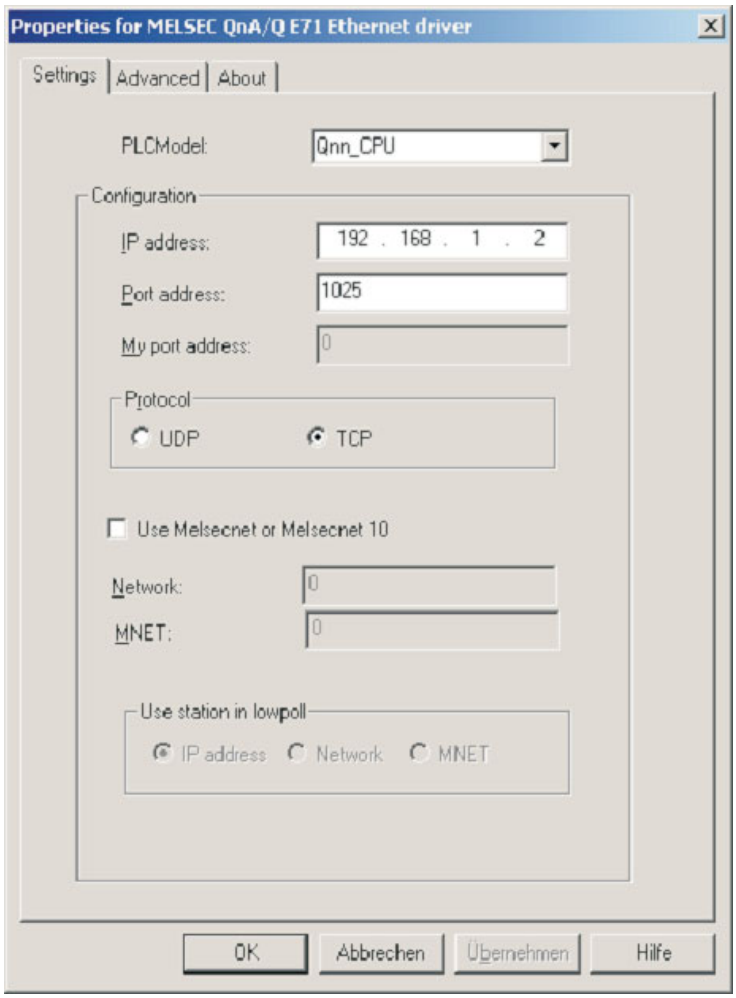
② Next, open up the ***Peripherals*** options under the System menu, and configure the HMI's TCP/IP connection as shown:





③ Then make the following settings for Controller 1 (i.e. the target PLC), according to the settings made in the PLC earlier.





As with the MQE settings earlier, note that E71 port number 1025, decimal 1025 is equal to hex 401 (set in the PLC Local station port number – refer back to part 10 of section 24.1.1).

④ Click ***OK***, exit the Peripheral settings and download these settings with the project.

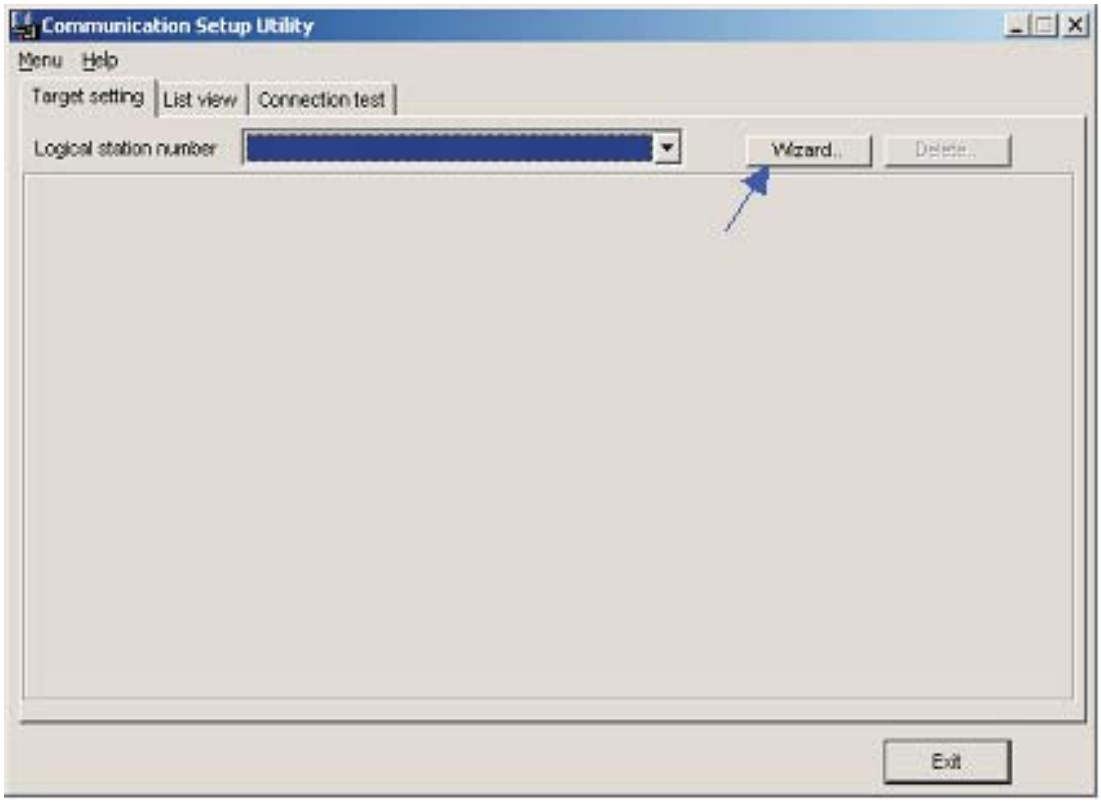
24.5 Communication via MX Component

MX Component is a tool designed to implement communication from PC to the PLC without any knowledge of communication protocols and modules.

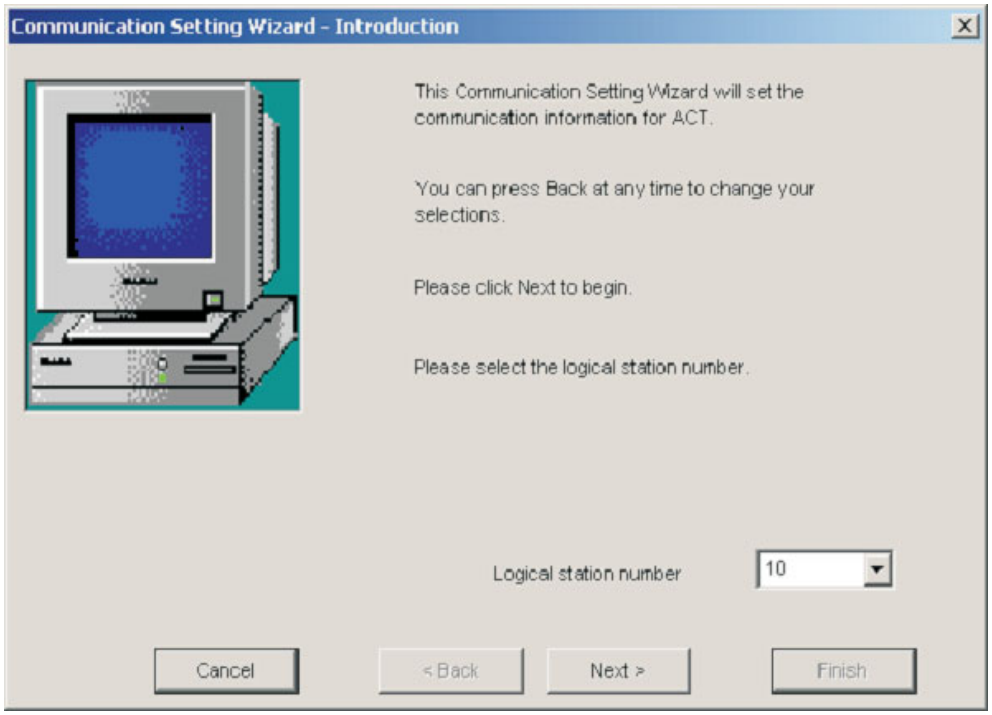
It supports serial CPU port connection, serial computer links (RS232C, RS422), Ethernet, CC-Link and MELSEC networks.

The figure below shows the easy way for creating of communication between a PC and a PLC via MX Component.

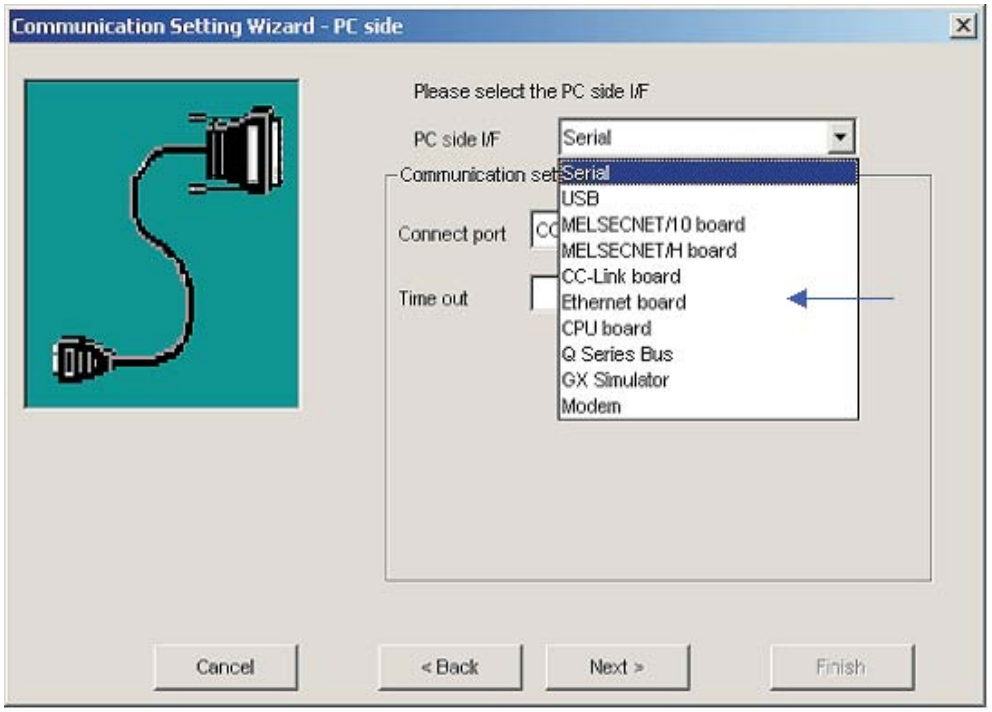
① Start the ***Communication Setting Utility*** and select the ***Wizard***



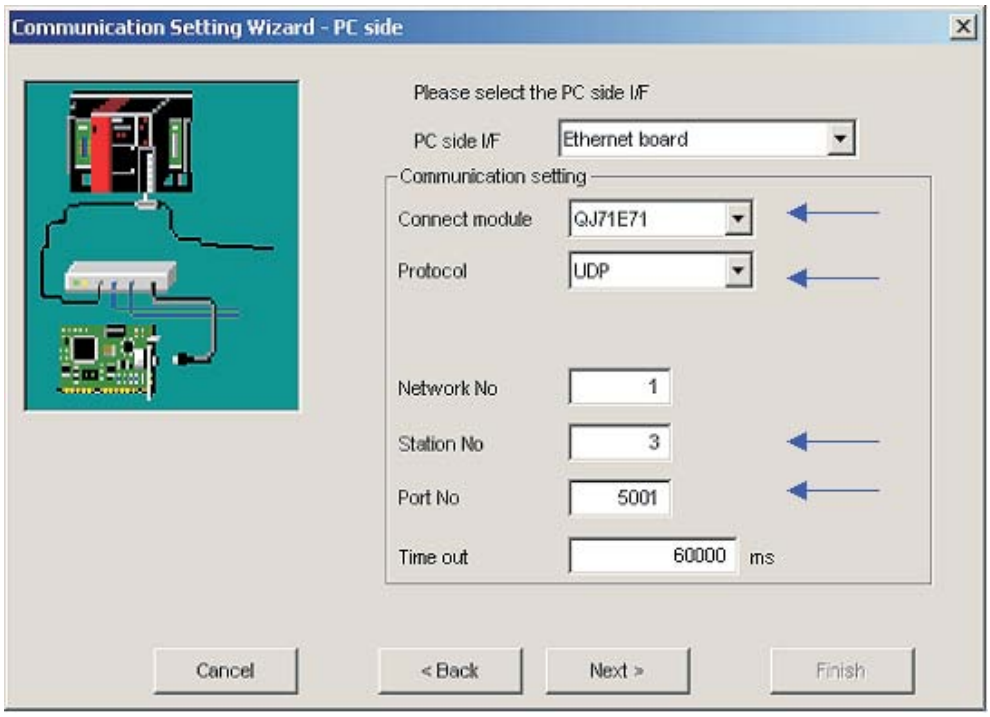
② First you must define the ***Logical station number***



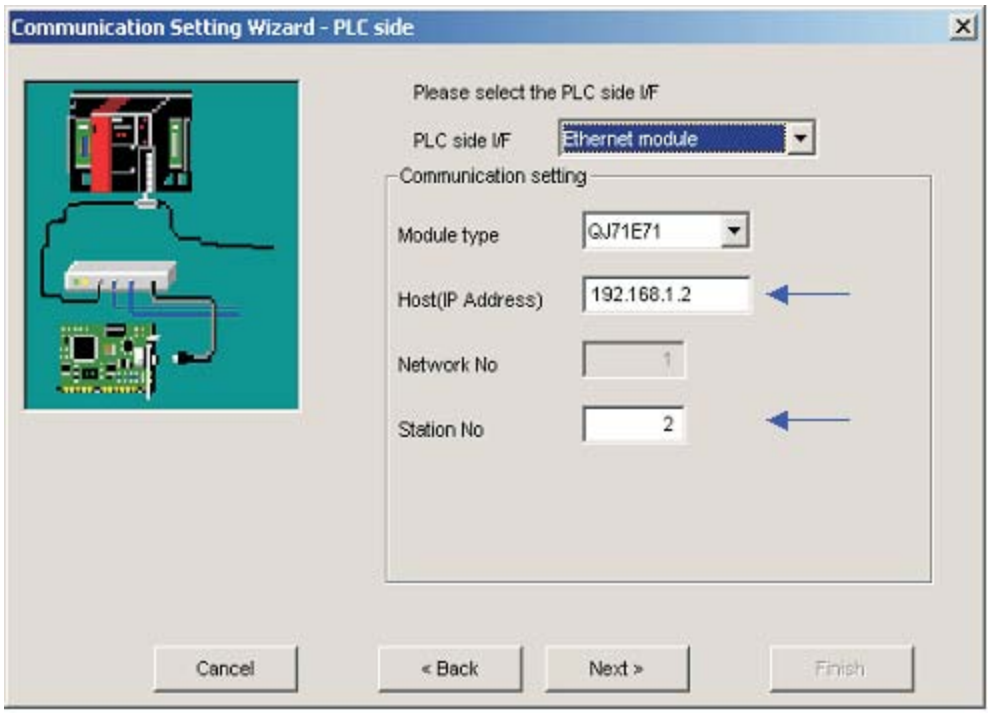
③ Next, configure the ***Communication Settings*** on the PC side



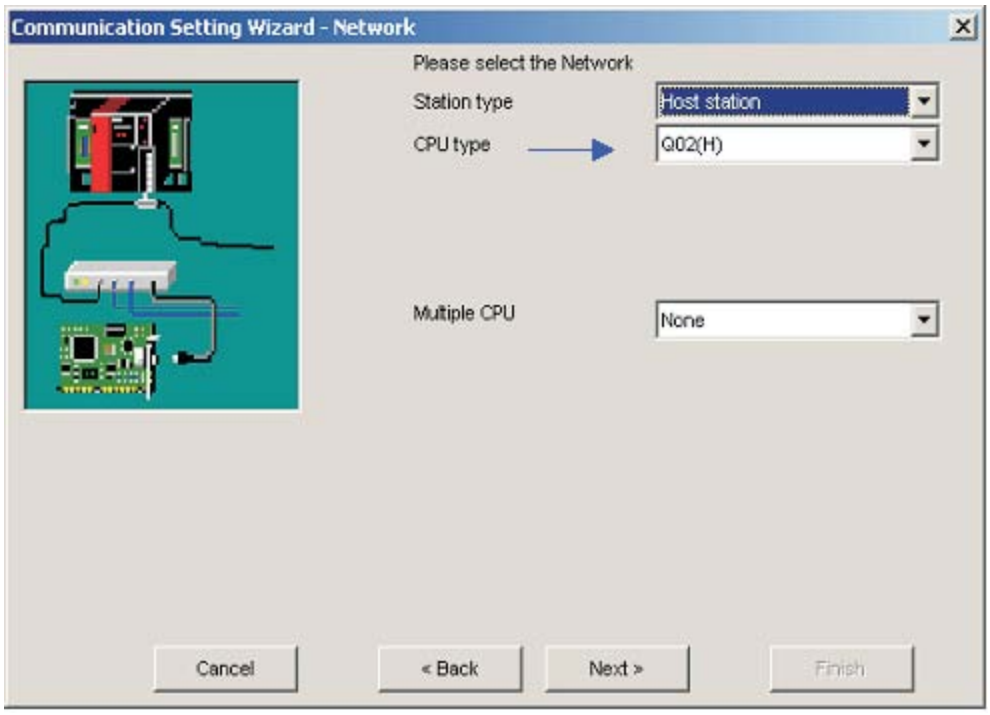
④ Select the UDP protocol and the default Port 5001



⑤ Configure the Communication settings of the PLC side required for the example system described earlier.

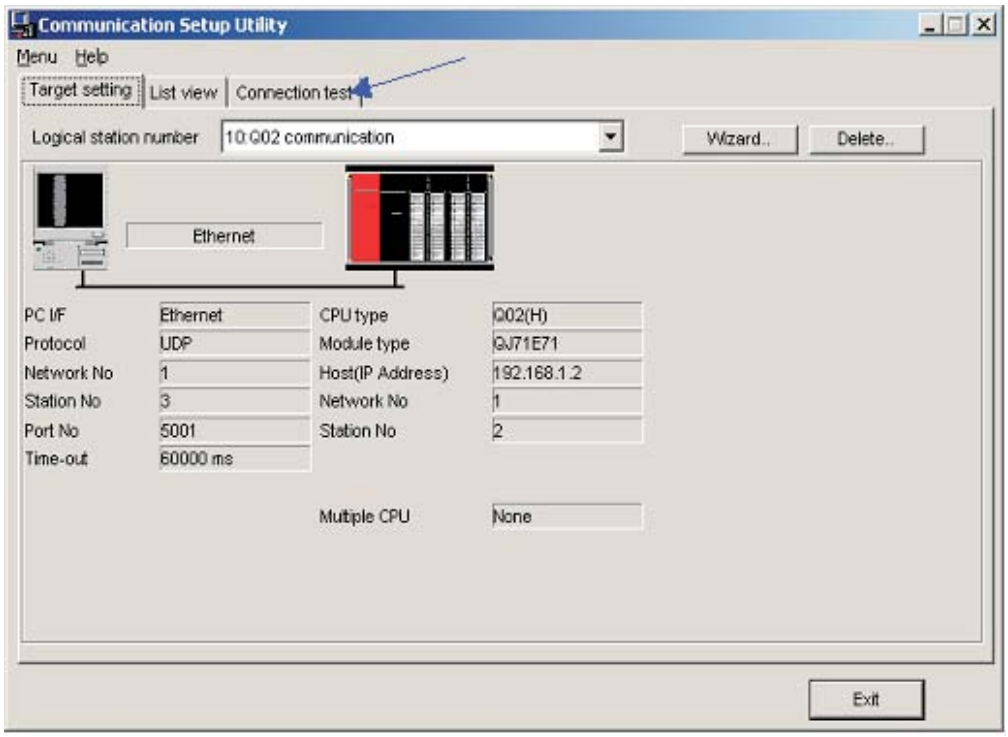


⑥ Select the correct CPU type.

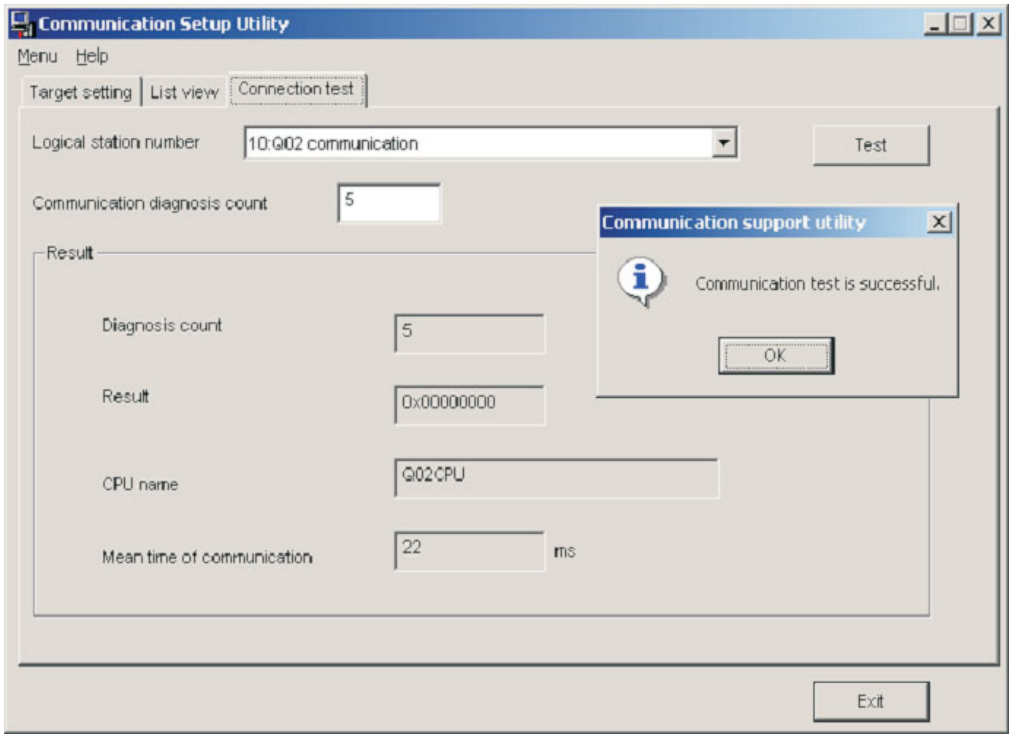


⑦ For the conclusion of the configuration define a name and press the ***Finish*** button

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



Select the ***Logical station number*** for which you want to accomplish the test. The ***Diagnosis count*** shows how many successful connection came. ***Result*** shows the test results. In case of an error an error number is indicated.



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

The Mitsubishi MX components described below are powerful, user-friendly tools that make it very easy to connect your Mitsubishi PLC with the PC world.

A Appendix A

A.1 Special Relay Functionality for A & Q Series PLC´s

Diagnostic special relays (SM) are internal relays the application of which is fixed in the PLC. Therefore, they 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.

NOTES

The special relays SM1200 to SM1255 are used for QnA CPU. These relays are vacant with a Q CPU.

The special relays from SM1500 onward are dedicated for Q4AR CPU

The headings in the table that follows have the following meanings.

Item	Meaning
Number	Indicates the number of the diagnostic special relay.
Name	Indicates the name of the diagnostic special relay.
Meaning	Contains the function of the diagnostic special relay in brief.
Description	Contains a detailed description of the diagnostic special relay.
Set by (if set)	Indicates whether the diagnostic special relay was set by the system or the user. **<Set by>** **S**: Set by the system **U**: Set by the user (via sequence program or a programming terminal in test mode) **S/U**: Set by the system or user Is indicated only if the setting is done by the system. **<if set>** **END processing**: Set during END processing **Initial**: Set during initial processing (Power ON, STOP->RUN) **Status change**: Set after status change **Error**: Set after error **Instruction execution**: Set during instruction execution **Request**: Set for user request (through SM, etc.)
A CPU M9[ ] [ ] [ ]	Indicates special relay M9 [ ] [ ] [ ] corresponding to the A CPU (Change and notation when contents changed). Items indicated as „New“ were newly added to the Q-Series/System Q CPU.
Valid for:	Indicates the corresponding CPU: ●: Can be applied to all types of CPU **Q CPU**: Can be applied to a System Q CPU **QnA CPU**: Can be applied to a CPU of the QnA series and Q2AS series **CPU name**: Can be applied only to the specific CPU (e.g. Q4AR CPU) **Rem**: Can be applied to a remote MELSECNET/H I/O module

Diagnostic Information

Number	Name	Meaning	Description	Set by (if set)	A CPU M9[ ][ ][ ]	Valid for:
SM0	Diagnostic errors	OFF: No error ON: Error	ON if diagnosis results show error occurrence (Includes external diagnosis). Stays ON subsequently even if normal operations restored.	S (Error)	New	
SM1	Self-diagnostic error	OFF: No self diagnosis errors ON: Self-diagnosis	Comes ON when an error occurs as a result of self-diagnosis. Stays ON subsequently even if normal operations restored.	S (Error)	M9008	
SM5	Error common information	OFF: No error common information ON: Error common information	When SM0 is ON, ON if there is error common information.	S (Error)	New	● Rem
SM16	Error individual information	OFF: No error individual information ON: Error individual information	When SM0 is ON, ON if there is error individual information.	S (Error)	New	
SM50	Error reset	OFF -> ON: Error reset	Conducts error reset operation.	U	New	
SM51	Battery low latch	OFF: Normal ON: Battery low	ON if battery voltage at CPU or memory card drops below rated value. Stays ON subsequently even after normal operation is restored. Synchronous with BAT. ALARM LED.	S (Error)	M9007	●
SM52	Battery low	OFF: Normal ON: Battery low	Same as SM51, but goes OFF subsequently when battery voltage returns to normal.	S (Error)	M9006	
SM53	AC DOWN detection	OFF: AC DOWN detected ON: AC DOWN not detected	Comes ON when a AC power supply module is used and a momentary power interruption not exceeding 20 ms has occured; reset by turning the power OFF then ON again.	S (Error)	M9005	●
			Comes ON when a DC power supply module is used and a momentary power interruption not exceeding 10 ms has occured; reset by turning the power OFF then ON again.			Q CPU
			Comes ON when a DC power supply module is used and a momentary power interruption not exceeding 1 ms has occured; reset by turning the power OFF then ON again.			QnA CPU
SM54	MINI link errors	OFF: Normal ON: Error	Goes ON if MINI (S3) link error is detected at even one of the installed AJ71PT32 (S3) modules. Stays ON subsequently even after normal operation is restored.	S (Error)	M9004	QnA CPU
SM56	Operation errors	OFF: Normal ON: Operation error	ON when operation error is generated. Stays ON subsequently even if normal operation is restored.	S (Error)	M9011	●
SM60	Blown fuse detection	OFF: Normal ON: Module with blown fuse	Comes ON even if there is only one output module with a blown fuse and remains ON even after return to normal. Blown fuse state is checked even for remote I/O station output modules.	S (Error)	M9000	● Rem
SM61	I/O module Verification error	OFF: Normal ON: Error	Comes ON if there is a discrepancy between the actual I/O modules and the registered information when the power is turned on. I/O module verification is also conducted for remote I/O station modules.	S (Error)	M9002	
SM62	Annunciator detection	OFF: Not detected ON: Detected	Goes ON if even one annunciator F goes ON.	S (Instruction execution)	M9009	●

Number	Name	Meaning	Description		Set by (if set)	A CPU M9[][][]	Valid for:
SM80	CHK detection	OFF: Not detected ON: Detected	Goes ON if error is detected by CHK instruction. Stays ON subsequently even after normal operation is restored.		S (Instruction execution)	New	QnA CPU, Q CPU (except Q00J, Q00 and Q01CPU)
SM90	Startup of watchdog timer for step transition (Enabled only when SFC program exists)	OFF: Not startet (watchdog timer reset) ON: Started (watchdog timer started)	Corresponds to SD90	Goes ON when measurement of step transition watchdog timer is commenced. Resets watchdog timer when it goes OFF.	U	M9108	
SM91			Corresponds to SD91			M9109	
SM92			Corresponds to SD92			M9110	
SM93			Corresponds to SD93			M9111	
SM94			Corresponds to SD94			M9112	
SM95			Corresponds to SD95			M9113	
SM96			Corresponds to SD96			M9114	
SM97			Corresponds to SD97			New	
SM98			Corresponds to SD98			New	
SM99			Corresponds to SD99			New	

System Information

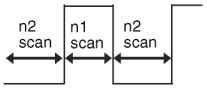
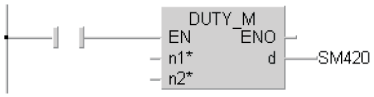
Number	Name	Meaning	Description	Set by (if set)	A CPU M9[][][]	Valid for:
SM202	LED off command	OFF -> ON: LED off	At change from OFF to ON, the LEDs corresponding to the individual bits at SD202 go off.	U	New	● (except Q00J, Q00, Q01CPU)
SM203	STOP contact	STOP state	Goes ON at STOP state.	S (Status change)	M9042	●
SM204	PAUSE contact	PAUSE state	GoesONat PAUSE state.	S (Status change)	M9041	
SM205	STEP-RUN contact	STEP-RUN state	Goes ONat STEP-RUN state.	S (Status change)	M9054	● (except Q00J, Q00 and Q01CPU)
SM206	PAUSE enable coil	OFF: PAUSE disabled ON: PAUSE enabled	PAUSE state is entered if this relay is ON when the remote PAUSE contact goes ON.	U	M9040	●
	Device test request acceptance status	OFF: Device test not yet executed ON: Device test executed	Comes ON when the device test mode is executed on the programming software.	S (Request)	New	Q00J Q00 and Q01 CPU
SM210	Clock data set request	OFF: Ignored ON: Set request	When this relay goes from OFF to ON, clock data being stored from SD210 through SD213 after execution of END instruction for changed scan is written to the clock device.	U	M9025	●
SM211	Clock data error	OFF: No error ON: Error	ON when error is generated in clock data (SD210 through SD213) value and OFF if no error is detected.	S (Request)	M9026	
SM212	Clock data display	OFF: Ignored ON: Display	Displays clock data as month, day, hour, minute and second at the LED display at front of CPU. (Enabled only for Q3A-CPU and Q4A-CPU)	U	M9027	Q3A, Q4A Q4AR CPU
SM213	Clock data read request	OFF: Ignored ON: Read request	When this relay is ON, clock data is read to SD210 through SD213 as BCD values.	U	M9028	● Rem

Number	Name	Meaning	Description	Set by (if set)	A CPU M9[][][]	Valid for:
SM240	No. 1 CPU reset flag	OFF: No reset ON: CPU 1 has been reset	This flag comes ON when the CPU no. 1 has been reset or has been removed from the base. The other CPUs of the multi-CPU system are also put in reset status.	S (Status change)	New	Q02, Q02H, Q06H, Q12H, Q25H CPU with function ver. B or later
SM241	No. 2 CPU reset flag	OFF: No reset ON: CPU 2 has been reset	This flag comes ON when the CPU no. 2 has been reset or has been removed from the base. In the other CPUs of the multi-CPU system the error code 7000 („MULTI CPU DOWN“) will occure.	S (Status change)	New	
SM242	No. 3 CPU reset flag	OFF: No reset ON: CPU 3 has been reset	This flag comes ON when the CPU no. 3 has been reset or has been removed from the base. In the other CPUs of the multi-CPU system the error code 7000 („MULTI CPU DOWN“) will occure.	S (Status change)	New	
SM243	No. 4 CPU reset flag	OFF: No reset ON: CPU 4 has been reset	This flag comes ON when the CPU no. 4 has been reset or has been removed from the base. In the other CPUs of the multi-CPU system the error code 7000 („MULTI CPU DOWN“) will occure.	S (Status change)	New	
SM244	No. 1 CPU error flag	OFF: No error ON: CPU no.1 is stopped due to an error	The set flag indicates that an error has occured which has stopped the CPU. The flag goes OFF when the CPU is normal or when an error occurs which will not stop the CPU.	S (Status change)	New	Q02, Q02H, Q06H, Q12H, Q25H CPU with function ver. B or later
SM245	No. 2 CPU error flag	OFF: No error ON: CPU no.2 is stopped due to an error		S (Status change)	New	
SM246	No. 3 CPU error flag	OFF: No error ON: CPU no.3 is stopped due to an error		S (Status change)	New	
SM247	No. 4 CPU error flag	OFF: No error ON: CPU no.41 is stopped due to an error		S (Status change)	New	

System Clocks

Number	Name	Meaning	Description	Set by (if set)	A CPU M9[ ][ ][ ]	Valid for:
SM400	Always ON	ON OFF	This flag is normally ON	S (Every END processing)	M9036	
SM401	Always ON	ON OFF	This flag is normally OFF	S (Every END processing)	M9037	
SM402	ON for 1 scan only after RUN	ON 1 scan OFF	After RUN, ON for 1 scan only. This connection can be used for scan execution type programs only.	S (Every END processing)	M9038	●
SM403	After RUN, OFF for 1 scan only	ON 1 scan OFF	After RUN, OFF for 1 scan only. This connection can be used for scan execution type programs only.	S (Every END processing)	M9039	
SM404	ON for 1 scan only after RUN	ON 1 scan OFF	After RUN, ON for 1 scan only. This connection can be used for scan execution type programs only.	S (Every END processing)	New	● (except Q00J, Q00 and Q01CPU)
SM405	After RUN, OFF for 1 scan only	ON 1 scan OFF	After RUN, OFF for 1 scan only. This connection can be used for scan execution type programs only.	S (Every END processing)	New	
SM409	0.01 second clock	0.005 s 0.005 s	Repeatedly changes between ON and OFF at 5-ms interval. When power supply is turned OFF, or reset is performed, goes from OFF to start.	S (Status change)	New	Q CPU (except Q00J, Q00 and Q01CPU)
SM410	0.1 second clock	0.05 s 0.05 s	Repeatedly changes between ON and OFF at each designated time interval. Operation continues even during STOP. When power supply is turned OFF, or reset is performed, goes from OFF to start.	S (Status change)	M9030	
SM411	0.2 second clock	0.1 s 0.1 s			M9031	
SM412	1 second clock	0.5 s 0.5 s			M9032	●
SM413	2 second clock	1 s 1 s			M9033	
SM414	2x n second clock	n (s) n (s)	Goes between ON and OFF in accordance with the number of seconds designated by SD414.		M9034 format change	
SM415	2 x n ms clock	n (ms) n (ms)	Goes between ON and OFF in accordance with the number of milliseconds designated by SD415.	S (Status change)	New	Q CPU (except Q00J, Q00 and Q01CPU)

System Clocks (continued)

Number	Name	Meaning	Description	Set by (if set)	A CPU M9[ ] [ ] [ ]	Valid for:
SM420	User timing clock No. 0		Relay repeats ON/OFF switching at fixed scan intervals. When power supply is turned ON, or reset is performed, goes from OFF to start. The ON/OFF intervals are set with the DUTY instruction. 	S (Every END processing)	M9020	
SM421	User timing clock No.1				M9021	
SM422	User timing clock No. 2				M9022	
SM423	User timing clock No. 3				M9023	●
SM424	User timing clock No. 4					
SM430	User timing clock No. 5				M9024	
SM431	User timing clock No. 6					
SM432	User timing clock No. 7		For use with SM420 through SM424 low speed programs.	S (Every END processing)	New	● (except Q00J, Q00 and Q01CPU)
SM433	User timing clock No. 8					
SM434	User timing clock No. 9					

A.2 A to Q series conversion correspondences

For a conversion from the MELSEC A series to the MELSEC Q series the special relays M9000 through M9255 (A series) correspond to the diagnostic relays SM1000 through SM1255 (Q series).

These diagnostic special relays are all set by the system and cannot be changed by a user-program. Users intending to set or reset these relays should alter their programs so that only real Q/QnA series diagnostic special relays are applied. An exception are the special relays M9084 and M9200 through M9255. If a user can set or reset some of these special relays befor conversion, the user can also set and reset the corresponding relays among SM1084 and SM1200 through SM1255 after the conversion.

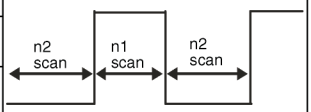
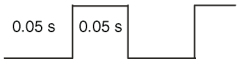
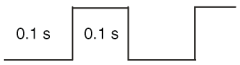
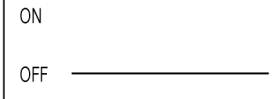
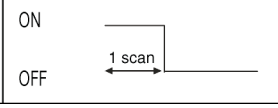
Refer to the manuals of the CPUs and the networks MELSECNET and MELSECNET/B for detailed information on the special relays of the A series.

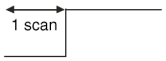
NOTE

The processing time may be longer when converted special relays are used with a Q CPU. Don´t select ***A-PLC: Use special relay/special register from SM/SD 1000*** within the PC system setting in the GX Developer parameters when converted special relays are not used.

When a special relay for modification is provided, the device number should be changed to the provided System Q/QnA CPU special relay. When no special relay for modification is provided, the converted special relay can be used for the device number.

A CPU special relay	Special relay after conversion	Equivalent System Q/QnA diagnostic special relay	Name	Meaning	Valid for:
M9000	SM1000	—	Fuse blown	OFF: Normal ON: Fuse blown module with blown fuse present	System Q/ QnA CPU
M9002	SM1002	—	I/O module verification error	OFF: Normal ON: Error	
M9004	SM1004	—	MINI link error	OFF: Normal ON: Error	QnA CPU
M9005	SM1005	—	AC DOWN detection	OFF: AC DOWN not detected ON: AC DOWN detected	System Q/ QnA CPU
M9006	SM1006	—	Battery low	OFF: Normal ON: Battery low	
M9007	SM1007	—	Battery low (latched)	OFF: Normal ON: Battery low	
M9008	SM1008	SM1	Self-diagnostic error	OFF: No error ON: Error	
M9009	SM1009	SM62	Annunciator detection	OFF: No F number detected ON: F number detected	
M9011	SM1011	SM56	Operation error flag	OFF: No error ON: Error	
M9012	SM1012	SM700	Carry Flag	OFF: Carry OFF ON: Carry ON	
M9016	SM1016	The device does not work with a System Q/QnA CPU	Data memory clear flag	OFF: Ignored ON: Output cleared	
M9017	SM1017	The device does not work with a System Q/QnA CPU	Data memory clear flag	OFF: Ignored ON: Output cleared	

A CPU special relay	Special relay after conversion	Equivalent System Q/QnA diagnostic special relay	Name	Meaning	Valid for:
M9020	SM1020	—	User timing clock No. 0		System Q/ QnA CPU
M9021	SM1021	—	User timing clock No. 1		
M9022	SM1022	—	User timing clock No. 2		
M9023	SM1023	—	User timing clock No. 3		
M9024	SM1024	—	User timing clock No. 4		
M9025	SM1025	—	Clock data set request	OFF: Ignored ON: Set request present used	
M9026	SM1026	—	Clock data error	OFF: No error ON: Error	
M9027	SM1027	—	Clock data display	OFF: Ignored ON: Display	
M9028	SM1028	—	Clock data read request	OFF: Ignored ON: Read request	
M9029	SM1029	The device does not work with a System Q/QnA CPU	Batch processing of data communications request	OFF: Batch processing not conducted ON: Batch processing conducted	
M9030	SM1030	—	0.1 second clock		
M9031	SM1031	—	0.2 second clock		
M9032	SM1032	—	1 second clock		
M9033	SM1033	—	2 second clock		
M9034	SM1034	—	1 minute clock		
M9036	SM1036	—	Always ON		
M9037	SM1037	—	Always OFF		
M9038	SM1038	—	ON for 1 scan only after RUN		

A CPU special relay	Special relay after conversion	Equivalent QnA diagnostic special relay	Name	Meaning	Valid for:
M9039	SM1039	—	RUN flag (After RUN, OFF for 1 scan only)	ON  OFF	System Q/ QnA CPU
M9040	SM1040	SM206	PAUSE enable coil	OFF: PAUSE disabled ON : PAUSE enabled	
M9041	SM1041	SM204	PAUSE status contact	OFF: PAUSE not in effect ON: PAUSE in effect	
M9042	SM1042	SM203	STOP status contact	OFF: STOP not in effect ON: STOP in effect	
M9043	SM1043	SM805	Sampling trace completed	OFF: Sampling trace in progress ON: Sampling trace completed	
M9044	SM1044	SM803	Sampling trace	0 1 STRA Same as execution 1 0 STRAR Same as execution	
M9045	SM1045	The device does not work with a System Q/QnA CPU.	Watchdog timer (WDT) reset	OFF: Does not reset WDT ON: Resets WDT	
M9046	SM1046	SM802	Sampling trace	OFF: Trace not in progress ON: Trace in progress	
M9047	SM1047	SM801	Sampling trace preparations	OFF: Sampling Trace suspended ON: Sampling Trace started	
M9049	SM1049	SM701	Selection of number of characters output	OFF: Output until NUL ON: 16 characters output	
M9051	SM1051	The device does not work with a System Q/QnA CPU.	CHG instruction execution disable	OFF: Enabled ON: Disable	
M9052	SM1052	The device does not work with a System Q/QnA CPU.	SEG instruction switch	OFF: 7 segment display ON: I/O partial refresh	
M9054	SM1054	SM205	STEP RUN flag	OFF: STEP RUN not in effect ON: STEP RUN in effect	
M9055	SM1055	SM808	Status latch completion flag	OFF: Not completed ON: Completed	QnA CPU
M9056	SM1056	These devices do not work with a System Q/QnA CPU.	Main side P, I set request	OFF: Other than when P, I set being requested ON: P, I set being requested	System Q/ QnA CPU
M9057	SM1057		Sub side P, I set request		
M9058	SM1058		Main program P, I set completion	Momentarily ON at P, I set completion	
M9059	SM1059		Sub program P, I set completion	Momentarily ON at P, I set completion	
M9060	SM1060		Sub program 2 P, I set request	OFF: Other than when P, I set being requested ON: P, I set being requested	
M9061	SM1061		Sub program 3 P, I set request		

A CPU special relay	Special relay after conversion	Equivalent QnA diagnostic special relay	Name	Meaning	Valid for:
M9065	SM1065	SM711	Divided processing execution detection	OFF: Divided processing not underway ON: During divided processing	QnA CPU
M9066	SM1066	SM712	Divided processing request flag	OFF: Batch processing ON: Divided processing	
M9070	SM1070	The device does not work with a System Q/QnA CPU.	A8UPU/A8PUJ required search time	OFF: Read time not shortened ON: Read time shortened	System Q/ QnA CPU
M9081	SM1081	SM714	Communication request registration area BUSY signal	OFF: Empty spaces in communication request registration area ON: No empty spaces in communication request registration area	QnA CPU
M9084	SM1084	The device does not work with a System Q/QnA CPU.	Error check	OFF: Error check executed ON: No error check	System Q/ QnA CPU
M9091	SM1091	The device does not work with a System Q/QnA CPU.	Instruction error flag	OFF: No error ON: Error	
M9094	SM1094	SM251	I/O change flag	OFF: Replacement ON: No replacement	QnA CPU
M9100	SM1100	SM320	Presence/absence of SFC program	OFF: SFC programs not used ON: SFC programs used	System Q/ QnA CPU
M9101	SM1101	SM321	Start/stop SFC program	OFF: SFC programs stop ON: SFC programs start	
M9102	SM1102	SM322	SFC program start state	OFF: Initial Start ON: Continue	
M9103	SM1103	SM323	Presence/absence of continuous transition	OFF: Continuous transition not effective ON: Continuous transition effective	
M9104	SM1104	SM324	Continuous transition suspension flag	OFF: When transition is completed ON: When no transition	
M9108	SM1108	SM90	Step transition watchdog timer start (equivalent of D9108)	OFF: Watchdog timer reset ON: Watchdog timer reset start	
M9109	SM1109	SM91	Step transition watchdog timer start (equivalent of D9109)		
M9110	SM1110	SM92	Step transition watchdog timer start (equivalent of D9110)		
M9111	SM1111	SM93	Step transition watchdog timer start (equivalent of D9111)		
M9112	SM1112	SM94	Step transition watchdog timer start (equivalent of D9112)		
M9113	SM1113	SM95	Step transition watchdog timer start (equivalent of D9113)		
M9114	SM1114	SM96	Step transition watchdog timer start (equivalent of D9114)		
M9180	SM1180	SM825	Active step sampling trace execution flag	OFF: Trace will be started ON: Trace completed	
M9181	SM1181	SM822	Active step sampling trace execution flag	OFF: Trace not being executed ON: Trace execution under way	

A CPU special relay	Special relay after conversion	Equivalent QnA diagnostic special relay	Name	Meaning	Valid for:
M9182	SM1182	SM821	Active step sampling trace permission	OFF: Trace disable/suspend ON: Trace enable	System Q/ QnA CPU
M9196	SM1196	SM325	Operation output at block stop	OFF: Coil output OFF ON: Coil output ON	
M9197 . M9198	SM1197 . SM1198	The device does not work with a System Q/QnA CPU	Switch between blown fuse and I/O verification error display	Display is changed depending on combination of M9197 ON/OFF state and M9198 ON/OFF state.	
M9199	SM1199	The device does not work with a System Q/QnA CPU	On-line recovery of sampling trace status latch data	OFF : Does not perform data recovery ON: Performs data recovery	
M9200	SM1200	—	LRDP instruction reception	OFF: Not accepted ON: Accepted	QnA CPU
M9201	SM1201	—	LRDP instruction completion	OFF: Not completed ON: End	
M9202	SM1202	—	LWTP instruction reception	OFF: Not accepted ON: Accepted	
M9203	SM1203	—	LWTP instruction completion	OFF: Not completed ON: End	
M9204	SM1204	—	LRDP instruction completion	OFF: Not completed ON: End	
M9205	SM1205	—	LWTP instruction completion	OFF: Not completed ON: End	
M9206	SM1206	—	Host station link parameter error	OFF: Normal ON: Abnormal	
M9207	SM1207	—	Link parameter check results	OFF: YES ON: NO	
M9208	SM1208	—	Sets master station B and W transmission range (for lower link master stations only).	OFF: Transmits to tier 2 and tier 3 ON: Transmits to tier 2 only	
M9209	SM1209	—	Link parameter check command (for lower link master stations only).	OFF: Executing the check function ON: Check non-execution	
M9210	SM1210	—	Link card error (for local station)	OFF: Normal ON: Abnormal	
M9211	SM1211	—	Link module error (for master station use)	OFF: Normal ON: Abnormal	
M9224	SM1224	—	Link state	OFF: Online ON: Offline, station-to-station test, or self-loopback test	
M9225	SM1225	—	Forward loop error	OFF: Normal ON: Abnormal	
M9226	SM1226	—	Reverse loop error	OFF: Normal ON: Abnormal	
M9227	SM1227	—	Loop test state	OFF: Not being executed ON: Forward or reverse loop test execution underway	

A CPU special relay	Special relay after conversion	Equivalent QnA diagnostic special relay	Name	Meaning	Valid for:
M9232	SM1232	—	Local station operation state	OFF: RUN or STEP RUN state ON: STOP or PAUSE state	QnA CPU
M9233	SM1233	—	Local station error detect state	OFF: No errors ON: Error detection	
M9235	SM1235	—	Local station, remote I/O station parameter error detect state	OFF: No errorsl ON: Error detection	
M9236	SM1236	—	Local station, remote I/O station parameter error detect state	OFF: No communications ON: Communications underway	
M9237	SM1237	—	Local station, remote I/O station error	OFF: Normal ON: Abnormal	
M9238	SM1238	—	Local station, remote I/O station forward or reverse loop error	OFF: Normal ON: Abnormal	
M9240	SM1240	—	Link state	OFF: Online ON: Offline, station-to-station test or self-loopback test	
M9241	SM1241	—	Forward loop line error	OFF: Normal ON: Abnormal	
M9242	SM1242	—	Reverse loop line error	OFF: Normal ON: Abnormal	
M9243	SM1243	—	Loopback implementation	OFF: Loopback not being conducted ON: Loopback implementation	
M9246	SM1246	—	Data not received	OFF: Reception ON: No reception	
M9247	SM1247	—	Data not received	OFF: Reception ON: No reception	
M9250	SM1250	—	Parameters not received	OFF: Reception ON: No reception	
M9251	SM1251	—	Link relay	OFF: Normal ON: Abort	
M9252	SM1252	—	Loop test state	OFF: Not being executed ON: Forward or reverse loop test execution underway	
M9253	SM1253	—	Master station operation state	OFF: RUN or STEP RUN state ON: STOP or PAUSE state	
M9254	SM1254	—	Local station other than host station operation state	OFF: RUN or STEP RUN state ON: STOP or PAUSE state	
M9255	SM1255	—	Local station other than host station error	OFF: Normal ON: Abnormal	

A.3 Special Registers (SD)

The special registers (SD) are internal registers with fixed application in the PLC. Therefore, they cannot be used like other registers in a sequence program. However, some of them can be written as needed in order to control the CPU.

Data stored in special registers are stored as BIN values if no special designation has been made to it.

Represented here are some of the most commonly used devices.

NOTES

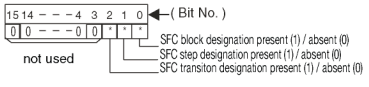
The special registers SD1200 to SD1255 are used for QnA CPU. These relays are vacant with a Q CPU.

The special registers from SM1500 onward are dedicated for Q4AR CPU

The headings in the table that follows have the following meanings.

Item	Meaning
Number	Indicates the number of the special register.
Name	Indicates the name of the special relgister.
Meaning	Contains the function of the special register in brief.
Description	Contains a detailed description of the special register.
Set by (if set)	Indicates whether the diagnostic special relay was set by the system or the user. **<Set by>** **S**: Set by the system **U**: Set by the user (via sequence program or a programming terminal in test mode) **S/U**: Set by the system or user Is indicated only if the setting is done by the system. **<if set>** **END processing**: Set during END processing **Initial**: Set during initial processing (Power ON, STOP->RUN) **Status change**: Set after status change **Error**: Set after error **Instruction execution**: Set during instruction execution **Request**: Set for user request (through SM, etc.)

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9[][][][]</th><th>Valid for:</th></tr>
<tr><td>SD0</td><td>Diagnostic errors</td><td>Diagnosis error code</td><td>Error codes for errors found by diagnosis are stored as BIN data.
Contents identical to latest fault history information.</td><td>S
(Error)</td><td>D9008
format change</td><td rowspan="4">●</td></tr>
<tr><td>SD1</td><td rowspan="3">Clock time for diagnosis error occurrence</td><td rowspan="3">Clock time for diagnosis error occurrence</td><td>Year (last two digits) and month that SD0 data was updated is stored as BCD 2-digit code.
Example: October 1995
H9510
b15 b8 b7 b0
Year (0 to 99) Month (1 to 31)</td><td rowspan="3">S
(Error)</td><td rowspan="3">New</td></tr>
<tr><td>SD2</td><td>The day and hour that SD0 was updated is stored as BCD 2-digit code.
Example: 10 p.m. on 25th
H2510
b15 b8 b7 b0
Day (1 to 31) | Hour (0 to 23)</td></tr>
<tr><td>SD3</td><td>The minute and second that SD0 data was updated is stored as BCD 2-digit code.
Example: 35 min 48s
H3548
b15 b8 b7 b0
Minute (1 to 60) | Second (1 to 60)</td></tr>
<tr><td>SD4</td><td>Error information categories</td><td>Error information category code</td><td>Category codes which help indicate what type of information is being stored in the common information areas (SD5 through SD15) and the individual information areas (SD16 through SD26) are stored here.
b15 b8 b7 b0
Individual error info. | Common error info.
The common information category codes store the following codes:
0: No error
1: Unit/module No.
2: File name/Drive name
3: Time (value set)
4: Program error location
The individual information category codes store the following codes:
0: No error
1: (Open)
2: File name/Drive name
3: Time (value actually measured)
4: Program error location
5: Parameter number
6: Annunciator number
7: Check instruction malfunction number</td><td>S
(Error)</td><td>New</td></tr>
</table>

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for:</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>Error common information</td><td>Error common information</td><td>Common information corresponding to the error codes (SD0) is stored here.
The following four types of information are stored here:
(1) Unit/module No.
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Station / module number</td></tr><tr><td>SD6</td><td>I/O number</td></tr><tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>Vacant</td></tr></table>(2) File name/Drive name
Example:
File name = ABCDEFGH.IJK
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Drive</td></tr><tr><td>SD6
SD7
SD8
SD9</td><td>File name
ASCII code: 8 characters</td></tr><tr><td>SD10</td><td>Extension 2EH(.)</td></tr><tr><td>SD11</td><td>ASCII code: 3 characters</td></tr><tr><td>SD12
SD13
SD14
SD15</td><td>Vacant</td></tr></table><table><tr><th>b15</th><th>b0</th></tr><tr><td>B</td><td>A</td></tr><tr><td>D</td><td>C</td></tr><tr><td>F</td><td>E</td></tr><tr><td>H</td><td>G</td></tr><tr><td>I</td><td>.</td></tr><tr><td>K</td><td>J</td></tr></table>(3) Time (value set)
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5</td><td>Time: 1µs-steps (0 to 999 µs)</td></tr><tr><td>SD6</td><td>Time: 1ms-steps (0 to 999 ms)</td></tr><tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>Vacant</td></tr></table>(4) Program error location
<table><tr><th>Number</th><th>Meaning</th></tr><tr><td>SD5
SD6
SD7
SD8</td><td>File name
(ASCII code: 8 characters)</td></tr><tr><td>SD9</td><td>Extension 2EH(.)</td></tr><tr><td>SD10</td><td>(ASCII code: 3 characters)</td></tr><tr><td>SD11</td><td>Pattern*</td></tr><tr><td>SD12</td><td>Block No.</td></tr><tr><td>SD13</td><td>Step / transition No.</td></tr><tr><td>SD14</td><td>Sequence step No. (L)</td></tr><tr><td>SD15</td><td>Sequence step No. (H)</td></tr></table>* Contents of pattern data
 
</td><td>S
(Error)</td><td>New</td><td>●</td></tr>
</table>

Meaning of extensions

SD10 (SD9)	SD11 (SD10)		Extension name	File type
Higher byte	Lower byte	Higher byte		
51H	50H	41H	QPA	Parameters
51H	50H	47H	QPG	Sequence program
51H	43H	44H	QCD	Device comment
51H	44H	49H	QDI	Device initial value
51H	44H	52H	QDR	File register
51H	44H	53H	QDS	Simulation data
51H	44H	4CH	QDL	Local device
51H	54H	53H	QTS	Sampling trace data (QnA-CPU only)
51H	54H	4CH	QTL	Status latch data (QnA-CPU only)
51H	54H	50H	QTP	Program trace data (QnA-CPU only)
51H	54H	52H	QTR	SFC trace file
51H	46H	44H	QFD	Trouble history data

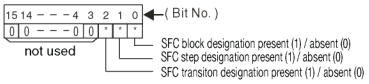
<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for</th></tr>
<tr><td>SD16
SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>Error individual information</td><td>Error individual information</td><td>Individual information corresponding to the error codes (SD0) is stored here.
The following six types of information are stored here:

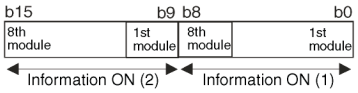
(1) File name/Drive name

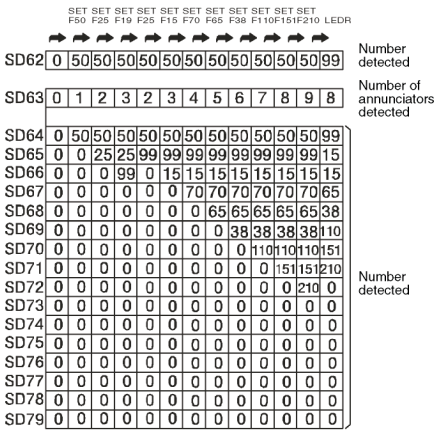
Example:
File name = ABCDEFGH.IJK

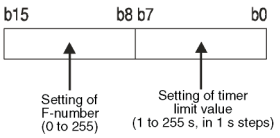
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Drive</td></tr>
<tr><td>SD17
SD18
SD19
SD20</td><td>File name
ASCII code: 8 characters</td></tr>
<tr><td>SD21</td><td>Extension 2EH(.)</td></tr>
<tr><td>SD22</td><td>ASCII code: 3 characters</td></tr>
<tr><td>SD23
SD24
SD25
SD26</td><td>Vacant</td></tr>
</table>
<table>
<tr><th>b15</th><th>b0</th></tr>
<tr><td>B</td><td>A</td></tr>
<tr><td>D</td><td>C</td></tr>
<tr><td>F</td><td>E</td></tr>
<tr><td>H</td><td>G</td></tr>
<tr><td>I</td><td>.</td></tr>
<tr><td>K</td><td>J</td></tr>
</table>
(2) Time (value actually measured)

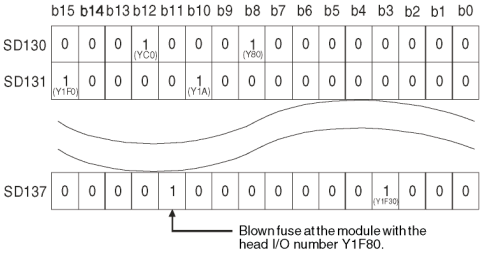
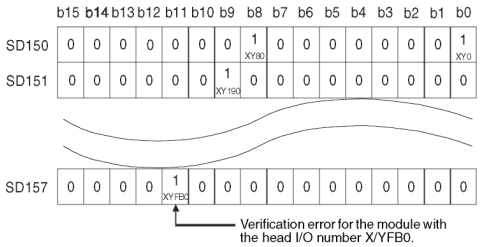
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16</td><td>Time: 1µs-steps (0 to 999 µs)</td></tr>
<tr><td>SD17</td><td>Time: 1ms-steps (0 to 999 ms)</td></tr>
<tr><td>SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>Vacant</td></tr>
</table>
(3) Program error location

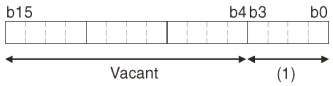
<table>
<tr><th>Number</th><th>Meaning</th></tr>
<tr><td>SD16
SD17
SD18
SD19</td><td>File name
(ASCII code: 8 characters)</td></tr>
<tr><td>SD20</td><td>Extension 2EH (.)</td></tr>
<tr><td>SD21</td><td>(ASCII code: 3 characters)</td></tr>
<tr><td>SD22</td><td>Pattern*</td></tr>
<tr><td>SD23</td><td>Block No.</td></tr>
<tr><td>SD24</td><td>Step / transition No.</td></tr>
<tr><td>SD25</td><td>Sequence step No. (L)</td></tr>
<tr><td>SD26</td><td>Sequence step No. (H)</td></tr>
</table>
* Contents of pattern data

</td><td>S
(Error)</td><td>New</td><td>●</td></tr>
</table>

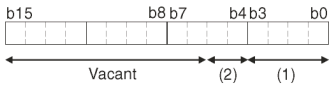
Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD54	MINI link errors	Error detection state	(1) The relevant station bit goes ON when any of the installed MINI (-S3) X(n+0) /X(n+20), X(n+6)/()n+26), X(n+7)/(n+27) or X(n+8)/Xn+28) goes ON. (2) Goes ON when communications between the installed MINI (-S3) and the CPU are not possible. 	S (Error)	D9004 format change	QnA-CPU
SD60	Blown fuse number	Number of module with blown fuse	Value stored here is the lowest station number of the module with the blown fuse, divided by 16.	S (Error)	D9000	● Rem
SD61	I/O module verification error	I/O module verification error module number	The lowest number of the module where the I/O module verification number took place.	S (Error)	D9002	
SD62	Annunciator number	Annunciator number	The first annunciator number to be detected is stored here.	S (Instruction execution)	D9009	●
SD63	Number of annunciators	Number of annunciators	Stores the number of annunciators searched.	S (Instruction execution)	D9124	

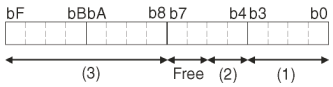
<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for</th></tr>
<tr><td>SD64</td><td rowspan="16">Table of detected annunciator numbers</td><td rowspan="16">Annunciator detection number</td><td rowspan="16">When F goes ON due to OUT F or SET F, the F numbers which go progressively ON from SD64 through SD79 are registered.
F numbers turned OFF by RST F are deleted from SD64 to SD79, and are shifted to the data register following the data register where the deleted F numbers had been stored.
Execution of the LEDR instruction shifts the contents of SD64 to SD79 up by one.
(This can also be done by using the INDICATOR RESET switch on the front of the CPU of the Q3A/Q4ACPUI.)
After 16 annunciators have been detected, detection of the 17th will not be stored from SD64 through SD79.
 
</td><td rowspan="16">S
(Instruction execution)</td><td>D9125</td><td rowspan="16">●</td></tr>
<tr><td>SD65</td><td>D9126</td></tr>
<tr><td>SD66</td><td>D9127</td></tr>
<tr><td>SD67</td><td>D9128</td></tr>
<tr><td>SD68</td><td>D9129</td></tr>
<tr><td>SD69</td><td>D9130</td></tr>
<tr><td>SD70</td><td>D9131</td></tr>
<tr><td>SD71</td><td>D9132</td></tr>
<tr><td>SD72</td><td>New</td></tr>
<tr><td>SD74</td><td>New</td></tr>
<tr><td>SD75</td><td>New</td></tr>
<tr><td>SD76</td><td>New</td></tr>
<tr><td>SD77</td><td>New</td></tr>
<tr><td>SD78</td><td>New</td></tr>
<tr><td>SD79</td><td>New</td></tr>
<tr><td>SD80</td><td>CHK number</td><td>CHK number</td><td>Error codes detected by the CHK instruction are stored as BCD code.</td><td>S
(Instruction execution)</td><td>New</td><td>●
(except Q00J, Q00 and Q01CPU)</td></tr>
</table>

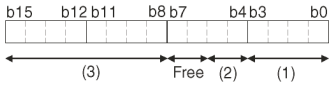
Number	Name	Meaning	Description		Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD90	Step transition watchdog timer setting value (Enabled only when SFC program exists)	F number for timer set value and time over error	Corresponds to SM90	F numbers that are set ON at setting value of step transition watchdog timer and watchdog timer over errors.  Timer is started by turning SM90 through SM99 ON during active step, and if the transition conditions for the relevant steps are not met within the timer limits, the designated annunciator (F) will go ON.	U	D9108	● (except Q00J, Q00 and Q01CPU)
SD91			Corresponds to SM91			D9109	
SD92			Corresponds to SM92			D9110	
SD93			Corresponds to SM93			D9111	
SD94			Corresponds to SM94			D9112	
SD95			Corresponds to SM95			D9113	
SD96			Corresponds to SM96			D9114	
SD97			Corresponds to SM97			New	
SD98			Corresponds to SM98			New	
SD99			Corresponds to SM99			New	
SD100	Transmission speed	Stores the transmission speed specified in the serial communication setting.	K96: 9600 bps, K192: 19.2 kbps, K384: 38.4 kbps, K576: 57.6 kbps, K1152: 115.2 kbps		S (power on or reset)	New	Q00JCPU Q00CPU Q01CPU
SD101	Communication settings	Stores the settings for serial communication	Bit 4 = OFF: Without sumcheck Bit 4 = ON: With sumcheck Bit 5 = OFF: Online program correction disabled Bit 5 = ON: Online program correction enabled The other bits have no function.			New	
SD102	Message waiting time	Stores the waiting time specified in the serial communication setting.	0: No waiting time 1 to F_H: Waiting time (unit: 10 ms) Default: 0			New	
SD105	CH1 transmission speed setting (RS232)	Stores the present transmission speed.	K3: 300 bps, K6: 600 bps, K24: 2400 bps, K48: 4800 bps, K96: 9600 bps, K192: 19.2 kbps, K384: 38.4 kbps, K576: 57.6 kbps, K1152: 115.2 kbps		S	New	Q CPU (except Q00J, Q00 and Q01CPU)
SD110	Data sending result	Stores the data sending result when the serial communication is used.	Stores the error code which occured during transmission using the serial communication.		S (Error)	New	Q00JCPU Q00CPU Q01CPU
SD111	Data receiving result	Stores the data receiving result when the serial communication is used.	Stores the error code which occured when data was received using the serial communication.		S (Error)	New	
SD120	Error number for external power supply OFF	Module number which has external power supply error	Stores the smallest head number of the module whose external power supply is OFF.		S (Error)	New	Q CPU (except Q00J, Q00 and Q01CPU)

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD130 SD131 SD132 SD133 SD134 SD135 SD136 SD137	Modules with blown fuse	The bit pattern (16 Bit) indicates the modules with a blown fuse. 0 : No blown fuse 1 : Blown fuse detected	The number of output modules whose fuses have blown are input as a bit pattern in units of 16 points. If the module numbers are set by parameter, the parameter-set numbers are stored. Blown fuses of remote station output modules will be detected also. A set bit is not automatically cleared when the module with the blown fuse is replaced. The flag is cleared by an error reset operation. 	S (Error)	New	Q00JCPU Q00CPU Q01CPU
SD150 SD151 SD152 SD153 SD154 SD155 SD156 SD157	I/O module verification error	The bit pattern (16 Bit) indicates the modules with verification errors. 0 : No I/O verification error 1 : I/O verification error present	When the power is turned on, the module numbers of the I/O modules whose information differs from the registered I/O module information are set in this register (in units of 16 points). I/O module information is also detected. 	S (Error)	New	

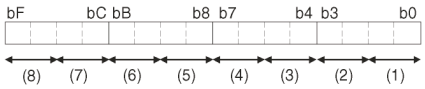
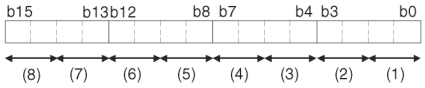
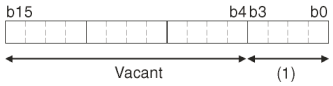
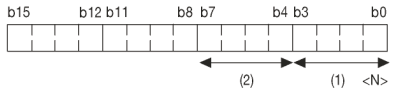
<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for:</th></tr>
<tr><td rowspan="4">SD200</td><td rowspan="4">State of switch</td><td rowspan="4">State of CPU switch</td><td>The status of the remote I/O module is stored in the following format:
 

(1) Remote I/O module switch status Always 1: STOP</td><td>S
(Continous)</td><td>New</td><td>Remote</td></tr>
<tr><td>The CPU switch state is stored in the following format:
 

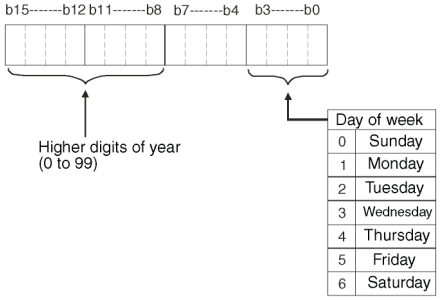
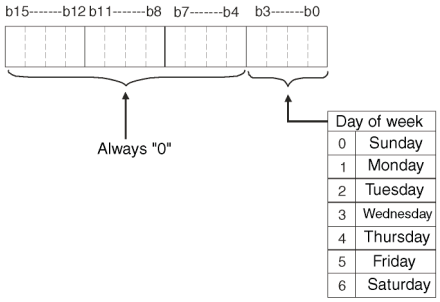
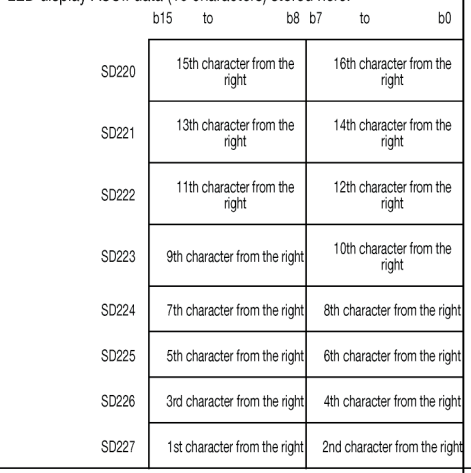
(1) CPU switch status (0): RUN (1): STOP
(2) Memory card switch Always OFF</td><td rowspan="3">S
(Every END processing)</td><td>New</td><td>Q00JCPU
Q00CPU
Q01CPU</td></tr>
<tr><td>The CPU switch state is stored in the following format:
 

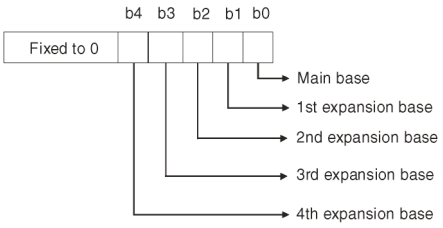
(1) CPU switch status (0): RUN (1): STOP (2): L.CLR
(2) Memory card switch Always OFF
(3) DIP-Switch b8 to bC correspond to SW1 through SW5 of system setting switch 1 . 0: OFF, 1: ON bD,bE and bF are vacant</td><td>New</td><td>Q CPU (except Q00J, Q00 and Q01CPU)</td></tr>
<tr><td>The CPU switch state is stored in the following format:
 

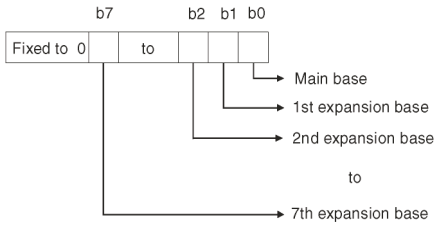
(1) : CPU Status (0) : RUN (1) : STOP (2) : L.CLR
(2) : Memory card switch B4 corresponds to card A, B5 corresponds to card B OFF for 0; ON for 1
(3) : DIP switch B8 to B15 correspond to SW1 to SW8 OFF for 0; ON for 1</td><td>S
(Every END processing)</td><td>New</td><td>QnA CPU</td></tr>
</table>

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD201	LED status	State of CPU-LED	The following bit patterns are used to store the statuses of the LEDs of the CPU:  (1) : RUN (5) : BOOT (2) : ERROR (6) : Vacant (3) : USER (7) : Vacant (4) : BAT.ALARM (8) : MODE Bitpatterns for MODE 0: OFF 1: Green 2: Orange The areas 3 to 8 are not available for a Q00JCPU, Q00CPU or Q01CPU.	S (Status change)	New	System Q CPU
			Information concerning which of the following states the LEDs on the CPU are stored in the following bit patterns: 0 is off, 1 is on, and 2 is flicker  (1) : RUN (5) : BOOT (2) : ERROR (6) : Card A (memory card) (3) : USER (7) : Card B (memory card) (4) : BAT.ALARM (8) : Vacant	S (Status change)	New	QnA CPU
SD202	LED off	Bit pattern of LED that is turned off	Stored bit patterns of LEDs turned off (Only USER and BOOT enabled) Turned off at 1, not turned off at 0	U	New	QnA CPU
SD203	Operating state of CPU	Operating state of CPU	The operating status of the remote I/O module is stored in the following format:  (1) Remote I/O module operating status — Always 2: STOP	S (Continous)	New	Remote
			The CPU operating state is stored as indicated in the following figure:  (1) : Operating state of CPU 0 : RUN 1 : STEP-RUN 2 : STOP 3 : PAUSE (2) : STOP/PAUSE cause 0 : Key switch 1 : Remote contact 2 : Peripheral, computer link, or operation from some other remote source 3 : Internal program instruction 4 : Error Remark: Only the error that occurred first is stored.	S (Every END processing)	D9015 (format change)	●

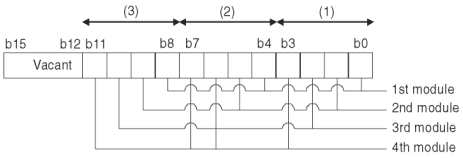
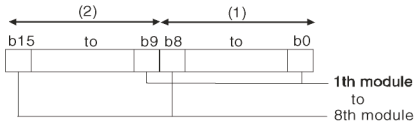
Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD206	Device test execution type	Indicates the kind of device test	When a device test is being executed by a programming device, the contents of this register reflects the state of the test: 0 = Test not yet executed 1 = Test of input devices (X) 2 = Test of output devices (Y) 3 = Test of input and output devices (X/Y)	S (Request)	New	Remote
SD207	LED display priority ranking	Priorities 1 to 4	When error is generated, the LED display (flicker) is made according to the error number setting priorities. The setting areas for priorities are as follows: SD207: Priority 4 | Priority 3 | Priority 2 | Priority 1 SD208: Priority 8 | Priority 7 | Priority 6 | Priority 5 SD209: Priority 10 | Priority 9 (4321H) (8765 H) (00A9 H) No display is made if "0" is set. However, even if "0" has been set, information concerning CPU operation stop (including parameter settings) errors will be indicated by the LEDs without conditions.	U	D9038	● (except Q00J, Q00 and Q01CPU)
SD208		Priorities 5 to 8			D9039 (format change)	
SD209		Priorities 9 to 10			New	
SD210	Clock data	Clock data (year, month)	The year (last two digits) and month are stored as BCD code at SD210 as shown below: b15 b12 b11 b8 b7 b4 b3 b0 Year Month Example: July 1993 = **H9307**	S/U (Request)	D9025	● Rem
SD211	Clock data	Clock data (day, hour)	The day and hour are stored as BCD code at SD211 as shown below: b15 b12 b11 b8 b7 b4 b3 b0 Day Hour Example: 31st, 10 a. m. = **H3110**		D9026	
SD212	Clock data	Clock data (minute, second)	The minutes and seconds (after the hour) are stored as BCD code at SD212 as shown below: b15 b12 b11 b8 b7 b4 b3 b0 Minute Second Example: 35 min, 48 sec. = **H3548**		D9027	

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for:</th></tr>
<tr><td rowspan="2">SD213</td><td rowspan="2">Clock data</td><td rowspan="2">Clock data
(day of the week)</td><td>The day of the week is stored as BCD code at SD213 as shown below:
 
</td><td>S/U
(Request)</td><td rowspan="2">D9028</td><td>Q CPU
Rem</td></tr>
<tr><td>The day of the week is stored as BCD code at SD213 as shown below:
 
</td><td>S/U
(Request)</td><td>QnA CPU</td></tr>
<tr><td>SD220
SD221
SD222
SD223
SD224
SD226
SD227</td><td>LED display data</td><td>Display indicator data</td><td>LED display ASCII data (16 characters) stored here.
 
</td><td>S
(Status change)</td><td>New</td><td>●</td></tr>
<tr><td>SD240</td><td>Base mode</td><td>0: Automatic mode
1: Detail mode</td><td>Stores the base mode</td><td>S
(Initial)</td><td>New</td><td rowspan="2">Q CPU
Rem</td></tr>
<tr><td>SD241</td><td>Number of extension bases</td><td>0: Basic only
1 to 7: Number of extension bases</td><td>Stores the number of extension bases being installed</td><td>S
(Initial)</td><td>New</td></tr>
</table>

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for:</th></tr>
<tr><td rowspan="2">SD242</td><td rowspan="2">A/Q base differentiation</td><td rowspan="2">0: QA[ ] [ ]B is installed (A mode)
1: Q[ ] [ ]B is installed (Q mode)</td><td> 

When no expansion base is installed, the value for b1 to b4 is fixed to "0".</td><td rowspan="2">S
(Initial)</td><td rowspan="2">New</td><td>Q00JCPU
Q00CPU
Q01CPU</td></tr>
<tr><td> 

When no expansion base is installed, the value for b1 to b7 is fixed to "0".</td><td>System Q CPU (except Q00JCPU Q00CPU Q01CPU)</td></tr>
<tr><td>SD243
SD244</td><td>Number of base slots</td><td>Number of base slots
The areas for the 5th to 7th expansion base are fixed to "0" for a Q00JCPU, Q00CPU or Q01CPU</td><td>bF bC bB b8 b7 b4 b3 b0
SM243: 3rd ext. | 2nd ext. | 1th ext. | Basic
SM244: 7th ext | 6th ext. | 5th ext. | 4th ext.
The number of slots being installed is stored in the respective areas for the basic base and the extension bases (ext.).</td><td>S
(Initial)</td><td>New</td><td>System Q CPU</td></tr>
<tr><td>SD250</td><td>Loaded maximum I/O</td><td>Loaded maximum I/O No.</td><td>When SM250 goes from OFF to ON, the upper 2 digits of the final I/O number plus 1 of the modules loaded are stored as BIN values.</td><td>S
(Request END)</td><td>New</td><td>●</td></tr>
<tr><td>SD251</td><td>Head I/O No. for replacement</td><td>Head I/O number for module replacement</td><td>Stores upper two digits of the first I/O number of an I/O module that is removed/replaced in the online status.</td><td>U</td><td>D9094</td><td>Q2A (S1)
Q3A
Q4A
Q4AR</td></tr>
<tr><td>SD253</td><td>RS422 baud rate</td><td>RS422 baud rate</td><td>Stores the baud rate of RS422:
0: 9600 bps, 1: 19,2 bps, 2: 38,4 bps</td><td>S
(When changed)</td><td>New</td><td>QnA CPU</td></tr>
</table>

Number	Name	Meaning		Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD254	MELSECNET/10 information	Number of modules installed		Indicates the number of modules installed on NET/10	S (Initial)	New	●
SD255		Information from 1st module	I/O No.	NET/10 I/O number of first module installed			
SD256			Network No.	NET/10 network number of first module installed			
SD257			Group Number	NET/10 group number of first module installed			
SD258			Station No.	NET/10 station number of first module installed			
SD259			Standby information	In the case of standby stations, the module number of the standby station is stored. (1 to 4)			● (except Q00JCPU Q00CPU Q01CPU)
SD260 — SD264		Information from 2nd module		Configuration is identical to that for the first module.			
SD265 — SD269		Information from 3rd module		Configuration is identical to that for the first module.			
SD270 — SD274		Information from 4th module		Configuration is identical to that for the first module.			
SD280	CC-Link error	Error detection status		 (1) When Xn0 of the installed CC-Link goes ON, the bit corresponding to the station switches ON. (2) When either Xn1 or XnF of the installed CC-Link switch OFF, the bit corresponding to the station switches ON. (3) Switches ON when the CPU cannot communicate with the installed CC-Link.	S (error)	New	Q CPU
				 (1) When Xn0 of the installed CC-Link goes ON, the bit corresponding to the station switches ON. (2) When either Xn1 or XnF of the installed CC-Link switch OFF, the bit corresponding to the station switches ON.	S (error)	New	QnA

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD290	Device allocation (Same as parameter contents)	Number of points allocated for X	Stores the number of points currently set for X	S (Initial)	New	● Rem
SD291		Number of points allocated for Y	Stores the number of points currently set for Y			
SD292		Number of points allocated for M	Stores the number of points currently set for M			
SD293		Number of points allocated for L	Stores the number of points currently set for L			
SD294		Number of points allocated for B	Stores the number of points currently set for B			● Rem
SD295		Number of points allocated for F	Stores the number of points currently set for F			●
SD296		Number of points allocated for SB	Stores the number of points currently set for SB			● Rem
SD297		Number of points allocated for V	Stores the number of points currently set for V			●
SD298		Number of points allocated for S	Stores the number of points currently set for S			
SD299		Number of points allocated for T	Stores the number of points currently set for T			
SD300		Number of points allocated for ST	Stores the number of points currently set for ST			
SD301		Number of points allocated for C	Stores the number of points currently set for C			
SD302		Number of points allocated for D	Stores the number of points currently set for D			
SD303	Device allocation (Same as parameter contents)	Number of points allocated for W	Stores the number of points currently set for W			● Rem
SD304		Number of points allocated for SW	Stores the number of points currently set for SW			
SD315	Time reserved for communication processing	Time reserved for communication processing	Reserves the designated time for communication processing with the GX developer or other units. The greater the value is designated, the shorter the response time for communication with other devices (GX Developer, serial communication units becomes. Setting range: 1 to 100 ms. If the specified value is out of range, it is assumed to no setting. The scan time becomes longer by the specified time.	END processing	New	System Q CPU

System Clocks / Counters

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD412	1 second counter	Number of counts in 1-second units	Following programmable controller CPU RUN, 1 is added each second. Count repeats from 0 to 32767 to -32768 to 0	S (Status change)	D9022	
SD414	n = 1 second steps	2n second clock units	Stores value n of 2n second clock (Default is 30). Setting can be made between 1 and 32767.	U	New	
SD415	n = 1 ms steps	2n ms clock units	Stores value n of 2n ms clock (Default is 30). Setting can be made between 1 and 32767.	U	New	System Q CPU (except Q00JCPU Q00CPU Q01CPU)
SD420	Scan counter	Number of counts in each scan	Incremented by 1 for each scan execution after the PC CPU is set to RUN. Count repeats from 0 to 32767 to -32768 to 0.	S (Every END processing)	New	
SD430	Low speed scan counter	Number of counts in each scan	Incremented by 1 for each scan execution after the PC CPU is set to RUN. Count repeats from 0 to 32767 to -32768 to 0. Used only for low speed execution type programs.	S (Every END processing)	New	(except Q00JCPU Q00CPU Q01CPU)

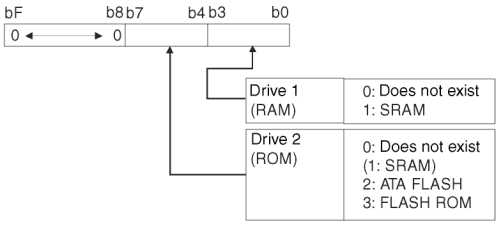
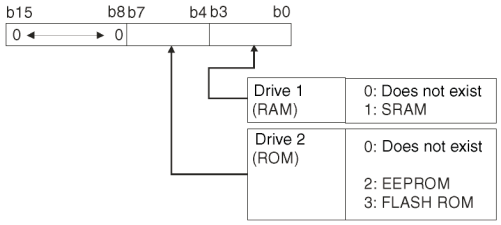
A.3.1 Scan Information

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD500	Execution program No.	Execution type of program being executed	Program number of program currently being executed is stored as BIN value.	S (Status change)	New	● (except Q00JCPU Q00CPU Q01CPU)
SD510	Low speed program No.	File name of low speed execution in progress	Program number of low speed program currently being executed is stored as BIN value. Enabled only when SM510 is ON.	S (Every END processing)	New	
SD520	Current scan time	Current scan time (in 1 ms units)	Stores current scan time (in 1 ms units) Range from 0 to 65535	S (Every END processing)	D9017 (format change)	●
SD521		Current scan time (in 1 s units)	Stores current scan time (in 1 s units) Range from 00000 to 900 (Example) A current scan of 23.6 ms would be stored as follows: D520 = 23 D521 = 600		New	
SD522	Initial scan time	Initial scan time (in 1 ms units)	Stores scan time for first scan (in 1 ms units). Range from 0 to 65535	S (First END processing)	New	● (except Q00JCPU Q00CPU Q01CPU)
SD523		Initial scan time (in 100 s units)	Stores scan time for first scan (in 1 s units). Range of 000 to 900			
SD524	Minimum scan time	Minimum scan time (in 1 ms units)	Stores minimum value of scan time (in 1 ms units). Range from 0 to 65535	S (Every END processing)	D9018 (format change)	●
SD525		Minimum scan time (in 100 s units)	Stores minimum value of scan time (in 100 s units). Range of 000 to 900		New	
SD526	Maximum scan time	Maximum scan time (in 1 ms units)	Stores meximum value of scan time, excepting the first scan. (in 1 ms units). Range from 0 to 65535	S (Every END processing)	D9019 (format change)	
SD527		Maximum scan time (in 100 s units)	Stores maximum value of scan time, excepting the first scan. (in 100 s units). Range of 000 to 900		New	
SD528	For low speed execution type programs current scan time	Current scan time (in 1 ms units)	Stores current scan time for low speed execution type program (in 1 ms units).	S (Every END processing)	New	● (except Q00JCPU Q00CPU Q01CPU)
SD529		Current scan time (in 100 s units)	Stores current scan time for low speed execution type program (in 100 s units). Range of 000 to 900			
SD532	Minimum scan time for low speed execution type programs	Minimum scan time (in 1 ms units)	Stores minimum value of scan time for low speed execution type program (in 1 ms units). Range from 0 to 65535	S (Every END processing)	New	
SD533		Minimum scan time (in 100 s units)	Stores minimum value of scan time for low speed execution type program (in 100 s units). Range of 000 to 900			
SD534	Maximum scan time for low speed execution type programs	Maximum scan time (in 1 ms units)	Stores the maximum scan time for all except low speed execution type program s first scan (in 1 ms units). Range from 0 to 65535	S (Every END processing)	New	
SD535		Maximum scan time (in 100 s units)	Stores the maximum scan time for all except low speed execution type program s first scan (in 100 s units). Range of 000 to 900			

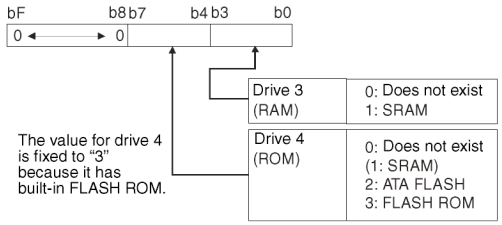
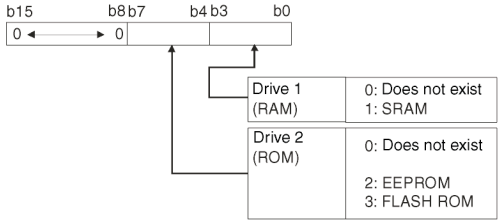
Scan Information (continued)

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD540	END processing time	END processing time (in 1 ms units)	Stores time from completion of scan program to start of next scan (in 1 ms units). Range from 0 to 65535	S (Every END processing)	New	●
SD541		END processing time (in 100 s units)	Stores time from completion of scan program to start of next scan (in 100 s units). Range of 000 to 900			
SD542	Constant scan wait time	Constant scan wait time (in 1 ms units)	Stores wait time when constant scan time has been set (in 1 ms units). Range from 0 to 65535	S (First END processing)	New	
SD543		Constant scan wait time (in 100 s units)	Stores wait time when constant scan time has been set (in 100 s units). Range of 000 to 900			
SD544	Cumulative execution time for low speed execution type programs	Cumulative execution time for low speed execution type programs (in 1 ms units)	Stores cumulative execution time for low speed execution type programs (in 1 ms units). Range from 0 to 65535 Cleared to 0 following 1 low speed scan	S (Every END processing)	New	● (except Q00JCPU Q00CPU Q01CPU)
SD545		Cumulative execution time for low speed execution type programs (in 100 s units)	Stores cumulative execution time for low speed execution type programs (in 100 s units). Range of 000 to 900 Cleared to 0 following 1 low speed scan			
SD546	Execution time for low speed execution type programs	Execution time for low speed execution type programs (in 1 ms units)	Stores low speed program execution time during 1 scan (in 1 ms units). Range from 0 to 65535 Stores each scan	S (Every END processing)	New	
SD547		Execution time for low speed execution type programs (in 100 s units)	Stores low speed program execution time during 1 scan (in 100 s units). Range of 000 to 900 Stores each scan			
SD548	Scan program execution time	Scan program execution time (in 1 ms units)	Stores execution time for scan execution type program during 1 scan (in 1 ms units). Range from 0 to 65535 Stores each scan	S (Every END processing)	New	●
SD549		Scan program execution time (in 100 s units)	Stores execution time for scan execution type program during 1 scan (in 100 s units). Range of 000 to 900 Stores each scan			
SD550	Service interval measurement module	Unit/module No.	Sets I/O number for module that measures service interval.	U	New	
SD551	Service interval time	Module service interval (in 1 ms units)	When SM 551 is ON, stores service interval for module designated by SD 550 (in 1 ms units). Range from 0 to 65535	S (Request)	New	● (except Q00JCPU Q00CPU Q01CPU)
SD552		Module service interval (in 100 s units)	When SM551 is ON, stores service interval for module designated by SD550 (in 1 s units). Range from 000 to 999			

Memory Cards

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD600	Memory card A models	Memory card A models	Indicates memory card A model installed. 	S (Initial and card removal)	New	System Q CPU (except Q00JCPU Q00CPU Q01CPU)
			Indicates memory card A model installed. 	S (Initial and card removal)	New	QnA CPU
SD602	Drive 1 (RAM) capacity	Drive 1 capacity	Drive 1 capacity is stored in 1 k byte units	S (Initial and card removal)	New	● (except Q00JCPU Q00CPU Q01CPU)
SD603	Drive 2 (ROM) capacity	Drive 2 capacity	Drive 2 capacity is stored in 1 k byte units	S (Initial and card removal)	New	
SD604	Memory card A use conditions	Memory card A use conditions	The use conditions for memory card A are stored as bit patterns (in use when ON). The significance of these bit patterns is indicated below: b0 : BOOT operation (QBT) b1 : Parameters (QPT) b2 : Device comments (QCD) b3 : Device initital value (QDI) b4 : File Register (QDR) b5 : Trace (QTS) b6 : b7 : b8 : b9 : CPU fault history (QFD) bA : SFC trace (QTS) bB : Local device (QDL) bC : bD : bE : bF :	S (Status change)	New	System Q CPU (except Q00JCPU Q00CPU Q01CPU)
			The use conditions for memory card A are stored as bit patterns (in use when ON). The significance of these bit patterns is indicated below: b0 : BOOT operation (QBT) b1 : Parameters (QPT) b2 : Device comments (QCD) b3 : Device initital value (QDI) b4 : File Register (QDR) b5 : Sampling trace (QTS) b6 : Status latch (QTL) b7 : Program trace (QTP) b8 : Simulation data (QDS) b9 : CPU fault history (QFD) bA : SFC trace (QTS) bB : Local device (QDL) bC : bD : bE : bF :	S (Status change)	New	QnA CPU

Memory Cards

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD620	Memory card B models	Memory card B models	Indicates memory card B models installed 	S (Initial)	New	System Q CPU
			Indicates memory card B models installed 	S (Initial)	New	QnA CPU
SD622	Drive 3 (RAM) capacity	Drive 3 capacity	Drive 3 capacity is stored in 1k byte units With a Q CPU, this value is fixed to "61" because of the built-in 61k RAM.	S (Initial)	New	System Q CPU
			Drive 3 capacity is stored in 1k byte units	S (Initial)	New	Q2(S1) Q3A Q4A Q4AR CPU
SD623	Drive 4 (ROM) capacity	Drive 4 capacity	Drive 4 capacity is stored in 1k byte units	S (Initial)	New	Q2(S1) Q3A Q4A Q4AR System Q CPU
SD624	Drive 3 use conditions	Drive 3 use conditions	The use condition of drive 3 is indicated by bit 4: b4 = OFF: Drive 3 is not used b4 = ON: Drive 3 is used to store file registers	S (Status change)	New	Q00JCPU Q00CPU Q01CPU
	Drive 3 and 4 use conditions	Drive 3 and 4 use conditions	The use conditions for memory card B are stored as bit patterns (In use when ON) The significance of these bit patterns is indicated below: b0 : BOOT operation (QBT) b1 : Parameters (QPA) b2 : Device comments (QCD) b3 : Device initital value (QDI) b4 : File R (QDR) b5 : Trace (QTS) b6 : b7 : b8 : b9 : CPU fault history (QFD) bA : SFC trace (QTS) bB : Local device (QDL) bC : bD : bE : bF :	S (Status change)	New	System Q CPU (except Q00JCPU Q00CPU Q01CPU)
	Memory card B use conditions	Memory card B use conditions	The use conditions for memory card B are stored as bit patterns (In use when ON) The significance of these bit patterns is indicated below: b0 : BOOT operation (QBT) b1 : Parameters (QPT) b2 : Device comments (QCD) b3 : Device initital value (QDI) b4 : File Register (QDR) b5 : Sampling trace (QTS) b6 : Status latch (QTL) b7 : Program trace (QTP) b8 : Simulation data (QDS) b9 : CPU fault history (QFD) bA : SFC trace (QTS) bB : Local device (QDL) bC : bD : bE : bF :	S (Status change)	New	Q2(S1) Q3A Q4A Q4AR CPU

File Register Information

<table>
<tr><th>Number</th><th>Name</th><th>Meaning</th><th>Description</th><th>Set by (if set)</th><th>ACPU register D9 [ ] [ ] [ ]</th><th>Valid for:</th></tr>
<tr><td>SD640</td><td>File register drive</td><td>Drive number</td><td>Stores drive number being used by file register</td><td>S
(Status change)</td><td>New</td><td rowspan="7"></td></tr>
<tr><td>SD641
SD642
SD643
SD644
SD645
SD646</td><td>File register file name</td><td>File register file name</td><td>Stores file register file name (with extension) selected at parameters or by use of QDRSET instruction as ASCII code.
b15 b8 b7 b0
SD641: 2nd character | 1st character
SD642: 4th character | 3rd character
SD643: 6th character | 5th character
SD644: 8th character | 7th character
SD645: 1st char. of extension | 2EH (.)
SD646: 3rd char. of extension | 2nd char. of extension</td><td>S
(Status change)</td><td>New</td><td>●</td></tr>
<tr><td>SD647</td><td>File register capacity</td><td>File register capacity</td><td>Stores the data capacity of the currently selected file register in 1 K word units.</td><td>S
(Status change)</td><td>New</td></tr>
<tr><td>SD648</td><td>File register block number</td><td>File register block number</td><td>Stores the currently selected file register block number.</td><td>S
(Status change)</td><td>D9035</td></tr>
<tr><td>SD650</td><td>Comment drive</td><td>Comment drive</td><td>Stores the comment drive number selected at the parameters or by the QCDSET instruction.</td><td>S
(Status change)</td><td>New</td><td rowspan="9">●
(except Q00JCPU Q00CPU Q01CPU)</td></tr>
<tr><td>SD651
SD652
SD653
SD654
SD655
SD656</td><td>Comment file name</td><td>Comment file name</td><td>Stores the comment file name (with extension) selected at the parameters or by the QCDSET instruction in ASCII code.
b15 b8 b7 b0
SD651: 2nd character | 1st character
SD652: 4th character | 3rd character
SD653: 6th character | 5th character
SD654: 8th character | 7th character
SD655: 1st char. of extension | 2EH (.)
SD656: 3rd char. of extension | 2nd char. of extension</td><td>S
(Status change)</td><td>New</td></tr>
<tr><td>SD660</td><td rowspan="7">Boot operation designation file</td><td>Boot designation file drive number</td><td>Stores the drive number where the boot designation file (*.QBT) is being stored.</td><td>S
(Initial)</td><td>New</td></tr>
<tr><td>SD661
SD662
SD663
SD664
SD665
SD666</td><td>File name of boot designation file</td><td>Stores the file name of the boot designation file (*.QBT).
b15 b8 b7 b0
SD661: 2nd character | 1st character
SD662: 4th character | 3rd character
SD663: 6th character | 5th character
SD664: 8th character | 7th character
SD665: 1st char. of extension | 2EH (.)
SD666: 3rd char. of extension | 2nd char. of extension</td><td>S
(Initial)</td><td>New</td></tr>
</table>

Instruction related registers

Number	Name	Meaning	Description	Set by (if set)	ACPU register D9 [ ] [ ] [ ]	Valid for:
SD705 SD706	Mask pattern	Mask pattern	During block operations, turning SM705 ON makes it possible to use the mask pattern being stored at SD705 (or at SD705 and SD706 if double words are being used) to operate on all data in the block with the masked values.	U	New	● (except Q00JCPU Q00CPU Q01CPU)
SD714	Number of vacant communication request registration areas	0 to 32	Stores the number of vacant blocks in the communications request area for remote terminal modules connected to the AJ71PT32-S3.	S (During execution)	M9081	QnA CPU
SD715 SD716 SD717	IMASK instruction mask pattern	Mask pattern	Patterns masked by use of the IMASK instruction are stored in the following manner: b15 ..... b0 SD715 I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0 SD716 I31 I30 I29 I28 I27 I26 I25 I24 I23 I22 I21 I20 I19 I18 I17 I16 SD717 I47 I46 I45 I44 I43 I42 I41 I40 I39 I38 I37 I36 I35 I34 I33 I32	S (During execution)	New	●
SD718 SD719	Accumulator	Accumulator	For use as replacement for accumulators used in A-series programs.	S/U	New	
SD720	Program No. destination for PLOAD instruction	Program number destination for PLOAD instruction	Stores the program number of the program to be loaded by the PLOAD instruction when designated. The destination range is from 1 to 124.	U	New	System Q CPU
SD730	No. of vacant registration area for CC-Link communication request	0 to 32	Stores the number of vacant registration areas for the request for communication with the intelligent device station connected to A(1S)J61QBT61.	S (During execution)	New	QnA CPU
SD736	PKEY input	PKEY input	SD that temporarily stores keyboard data input by means of the PKEY instruction.	S (During execution)	New	● (except Q00JCPU Q00CPU Q01CPU)

Index

P

Q

R

S

T

V

Global Partner. Local Friend.

EUROPE
MITSUBISHI ELECTRIC
EUROPE B.V.
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0) 2102 / 486-0
Fax: +49 (0) 2102 / 486-1120
e mail: megfamail@meg.mee.com

FRANCE
MITSUBISHI ELECTRIC EUROPE B.V.
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 1 55 68 55 68
Fax: +33 1 55 68 56 85
e mail: factory.automation@fra.mee.com

IRELAND
MITSUBISHI ELECTRIC EUROPE B.V.
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0) 1 / 419 88 00
Fax: +353 (0) 1 / 419 88 90
e mail: sales.info@meir.mee.com

ITALY
MITSUBISHI ELECTRIC EUROPE B.V.
Italian Branch
Via Paracelso 12
I-20041 Agrate Brianza (MI)
Phone: +39 039 6053 1
Fax: +39 039 6053 312
e mail: factory.automation@it.mee.com

SPAIN
MITSUBISHI ELECTRIC EUROPE B.V.
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés
Phone: +34 9 3 / 565 3160
Fax: +34 9 3 / 589 1579
e mail: industrial@sp.mee.com

UK
MITSUBISHI ELECTRIC EUROPE B.V.
UK Branch
Travellers Lane
GB-Hatfield Herts. AL10 8 XB
Phone: +44 (0) 1707 / 27 61 00
Fax: +44 (0) 1707 / 27 86 95
e mail: automation@meuk.mee.com

JAPAN
MITSUBISHI ELECTRIC CORPORATION
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 6221 6060
Fax: +81 3 6221 6075

USA
MITSUBISHI ELECTRIC AUTOMATION
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 / 478 21 00
Fax: +1 847 / 478 22 83

AUSTRIA
GEVA
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0) 2252 / 85 55 20
Fax: +43 (0) 2252 / 488 60
e mail: office@geva.at

BELARUS
TEHNIKON
Oktjabrskaya 16/5, Ap 704
BY-220030 Minsk
Phone: +375 (0)17 / 210 4626
Fax: +375 (0)17 / 210 4626
e mail: tehnikon@belsonet.net

BELGIUM
Koning & Hartman B.V.
Researchpark Zellik, Pontbeeklaan 43
BE-1731 Brussels
Phone: +32 (0)2 / 467 17 51
Fax: +32 (0)2 / 467 17 45
e mail: info@koningenhartman.com

BULGARIA
AKHNATON
Andrej Ljapchev Lbvd. Pb 21 4
BG-1756 Sofia
Phone: +359 (0) 2 / 97 44 05 8
Fax: +359 (0) 2 / 97 44 06 1
e mail: —

CZECH REPUBLIC
AutoCont
Control Systems s.r.o.
Nemocnicni 12
CZ-702 00 Ostrava 2
Phone: +420 59 / 6152 111
Fax: +420 59 / 6152 562
e mail: consys@autocont.cz

DENMARK
louis poulsen
industri & automation
Geminivej 32
DK-2670 Greve
Phone: +45 (0) 70 / 10 15 35
Fax: +45 (0) 43 / 95 95 91
e mail: lpia@lpmail.com

ESTONIA
UTU Elektrotehnika AS
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0) 6 / 51 72 80
Fax: +372 (0) 6 / 51 72 88
e mail: utu@utu.ee

FINLAND
Beijer Electronics OY
Ansatie 6a
FIN-01740 Vantaa
Phone: +358 (0) 9 / 886 77 500
Fax: +358 (0) 9 / 886 77 555
e mail: info@beijer.fi

GREECE
UTECO A.B.E.E.
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +302 (0) 10 / 42 10 050
Fax: +302 (0) 10 / 42 12 033
e mail: sales@uteco.gr

HUNGARY
Meltrade Ltd.

Fertõ Utca 14.
HU-1107 Budapest
Phone: +36 (0)1 / 431-9726
Fax: +36 (0)1 / 431-9727
e mail: office@meltrade.hu

ISRAEL
TEXEL Electronics Ltd.
Box 6272
IL-42160 Netanya
Phone: +972 (0) 9 / 863 08 91
Fax: +972 (0) 9 / 885 24 30
e mail: texel_me@netvision.net.il

KAZAKHSTAN
Kazpromautomatics Ltd.
2, Scladskaya Str.
KAZ-470046 Karaganda
Phone: +7 3212 50 11 50
Fax: +7 3212 50 11 50
e mail: info@kpakz.com

LATVIA
SIA POWEL
Lienes iela 28
LV-1009 Riga
Phone: +371 784 / 22 80
Fax: +371 784 / 22 81
e mail: utu@utu.lv

LITHUANIA
UAB UTU POWEL
Savanoriu pr. 187
LT-2053 Vilnius
Phone: +370 (0) 52323-101
Fax: +370 (0) 52322-980
e mail: powel@utu.lt

MOLDOVA
INTEHSIS SRL
Cuza-Voda 36/1-81
MD-2061 Chisinau
Phone: +373 (0)2 / 562 263
Fax: +373 (0)2 / 562 263
e mail: intehsis@mdl.net

NETHERLANDS
Koning & Hartman B.V.
Donauweg 2 B
NL-1000 AK Amsterdam
Phone: +31 (0)20 / 587 76 00
Fax: +31 (0)20 / 587 76 05
e mail: info@koningenhartman.com

NORWAY
Beijer Electronics A/S
Teglverksveien 1
N-3002 Drammen
Phone: +47 (0) 32 / 24 30 00
Fax: +47 (0) 32 / 84 85 77
e mail: info@beijer.no

POLAND
MPL Technology Sp. z o.o.
ul. Sliczna 36
PL-31-444 Kraków
Phone: +48 (0) 12 / 632 28 85
Fax: +48 (0) 12 / 632 47 82
e mail: krakow@mpl.pl

ROMANIA
Sirius Trading & Services srl
Str. Biharia No. 67-77
RO-013981 Bucuresti 1
Phone: +40 (0) 21 / 201 1146
Fax: +40 (0) 21 / 201 1148
e mail: sirius@siriustrading.ro

RUSSIA
Avtomatika Sever Ltd.
Lva Tolstogo Str. 7, Off. 311
RU-197376 St Petersburg
Phone: +7 812 1183 238
Fax: +7 812 1183 239
e mail: as@avtsev.spb.ru

RUSSIA
Consys
Promyshlennaya St. 42
RU-198099 St Petersburg
Phone: +7 812 325 3653
Fax: +7 812 147 2055
e mail: consys@consys.spb.ru

RUSSIA
Electrotechnical Systems Siberia
Shetinkina St. 33, Office 116
RU-630088 Novosibirsk
Phone: +7 3832 / 119598
Fax: +7 3832 / 119598
e mail: info@eltechsystems.ru

RUSSIA
Elektrostyle
Poslannikov Per., 9, Str. 1
RU-107005 Moscow
Phone: +7 095 542 4323
Fax: +7 095 956 7526
e mail: info@estl.ru

RUSSIA
Elektrostyle
Krasnij Prospekt 220-1, Office No. 312
RU-630049 Novosibirsk
Phone: +7 3832 / 106618
Fax: +7 3832 / 106626
e mail: info@estl.ru

RUSSIA
ICOS
Industrial Computer Systems Zao
Ryazanskij Prospekt, 8A, Off. 100
RU-109428 Moscow
Phone: +7 095 232 0207
Fax: +7 095 232 0327
e mail: mail@icos.ru

RUSSIA
NPP Uralelektra
Sverdlova 11A
RU-620027 Ekaterinburg
Phone: +7 34 32 / 532745
Fax: +7 34 32 / 532745
e mail: elektra@etel.ru

RUSSIA
STC Drive Technique
Poslannikov Per., 9, Str. 1
RU-107005 Moscow
Phone: +7 095 790 7210
Fax: +7 095 790 7212
e mail: info@privod.ru

SERBIA AND MONTENEGRO
INEA SR d.o.o.
Karadjordjeva 12/260
SCG-113000 Smederevo
Phone: +381 (0)26/ 617 - 163
Fax: +381 (0)26/ 617 - 163
e mail: inea_sr@verat.net

SLOVAKIA
AutoCont Control s.r.o.
Radlinského 47
SK-02601 Dolný Kubín
Phone: +421 435868 210
Fax: +421 435868 210
e mail: info@autocontcontrol.sk

SLOVENIA
INEA d.o.o.
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0) 1-513 8100
Fax: +386 (0) 1-513 8170
e mail: inea@inea.si

SWEDEN
Beijer Electronics AB
Box 426
S-20124 Malmö
Phone: +46 (0) 40 / 35 86 00
Fax: +46 (0) 40 / 35 86 02
e mail: info@beijer.se

SWITZERLAND
ECONOTEC AG
Postfach 282
CH-8309 Nürensdorf
Phone: +41 (0) 1 / 838 48 11
Fax: +41 (0) 1 / 838 48 12
e mail: info@econotec.ch

SOUTH AFRICA
CBI Ltd.
Private Bag 2016
ZA-1600 Isando
Phone: +27 (0) 11/ 928 2000
Fax: +27 (0) 11/ 392 2354
e mail: cbi@cbi.co.za

TURKEY
GTS
Darülaceze Cad. No. 43 Kat. 2
TR-80270 Okmeydani-Istanbul
Phone: +90 (0) 212 / 320 1640
Fax: +90 (0) 212 / 320 1649
e mail: gts@turk.net

UKRAINE
CSC Automation Ltd.
15, M. Raskova St., Fl. 10, Office 1010
UA-02002 Kiev
Phone: +380 (0) 44 / 494 3355
Fax: +380 (0) 44 / 494 3366
e mail: csc-a@csc-a.kiev.ua

MITSUBISHI ELECTRIC
FACTORY AUTOMATION

Mitsubishi Electric Europe B.V. /// FA - European Business Group /// Gothaer Straße 8 /// D-40880 Ratingen /// Germany
Tel.: +49(0)2102-4860 /// Fax: +49(0)2102-486112 /// info@mitsubishi-automation.com /// www.mitsubishi-automation.de

Specifications subject to change /// Art. no. 170294 /// 08.2006