

PLC Applications of Programming

This course is for participants who have completed the MELSEC-Q Series Basics Course and is ready to learn the next step of programming.

Introduction Purpose of the Course

This course is intended for users who have completed the Basic Course or who have enough knowledge to learn about more details on the functions and usage of the MELSEC-Q Series programmable controllers.

By taking this course, you will learn about usage of different devices of the Q Series programmable controllers, the CPU system configuration and diagnostics, and usage of fundamental functions of the Q Series programmable controllers.

Introduction **Course Structure**



The contents of this course are as follows.
We recommend that you start from Chapter 1.

Chapter 1 Device Setting and Modification

Learn how to set and change the device settings, and the latch function.

Chapter 2 How to Use Devices with Various Functions

Learn how to use the retentive timer, index register, special relay, and special register.

Chapter 3 Memory for the CPU Module and File Register

Learn about the types of memory that can be used with the CPU module and how to use the file register.

Chapter 4 Programs with Real Numbers

Learn about the handling of real numbers and operations using real numbers.

Chapter 5 Concept of I/O Numbers and How to Use the I/O Assignment Function

Learn about the concept of I/O numbers and how to use the I/O assignment function.

Final Test

Passing grade: 60% or higher.

Introduction How to Use This e-Learning Tool



| | | |
|---------------------------|--|---|
| Go to the next page | | Go to the next page. |
| Back to the previous page | | Back to the previous page. |
| Move to the desired page | | "Table of Contents" will be displayed, enabling you to navigate to the desired page. |
| Exit the learning | | Exit the learning. Window such as "Contents" screen and the learning will be closed. |

Introduction **Cautions for Use**

Safety precautions

When you learn by using actual products, please carefully read the safety precautions in the corresponding manuals.

Precautions in this course

- The displayed screens of the software version that you use may differ from those in this course.

This course uses the following software version:

- GX Works2 Version 1.91V

Chapter 1 Device Setting and Modification

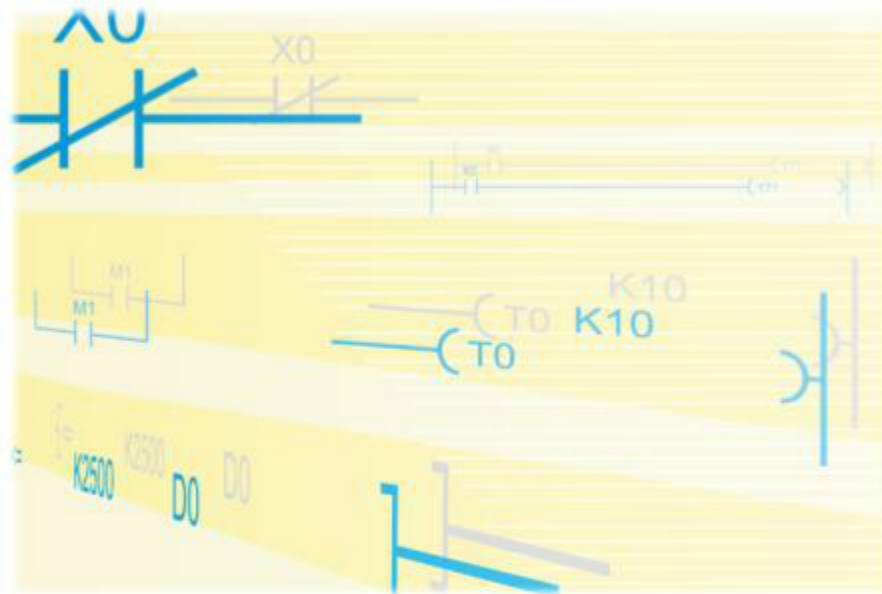
This chapter explains how to change the device settings.

Section 1.1: Specifying the Devices

Section 1.2: How to Customize the Number of Device Points

Section 1.3: Saving the Device Status at Power-Off or Reset

Section 1.4: Summary



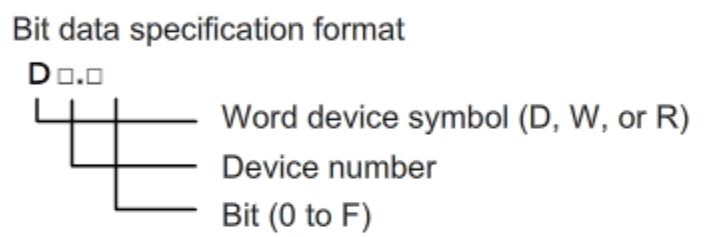
1.1 Specifying the Devices

1.1.1 Bit specification of word device

Word devices are normally specified using word data, but they can also be specified using bit data (such as integer, etc.). Bit data can be used in word devices such as a data register (D) and file register (R).

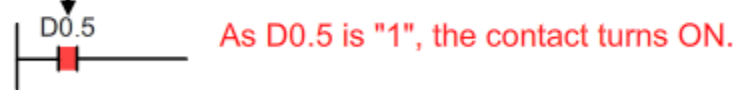
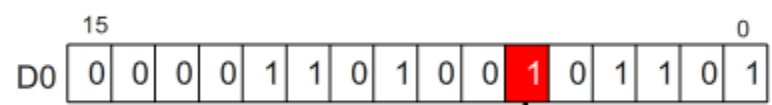
Example: Data register (D)

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

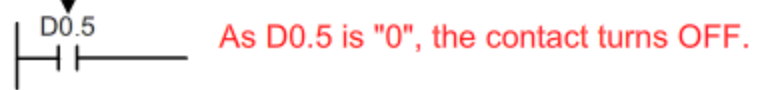
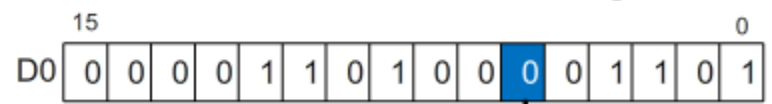


Program example

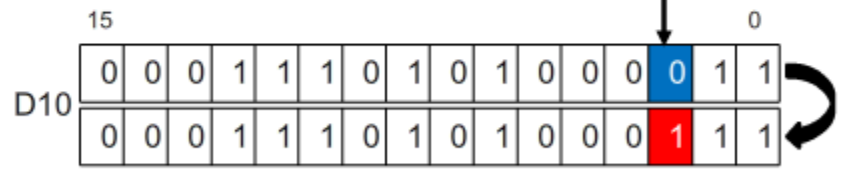
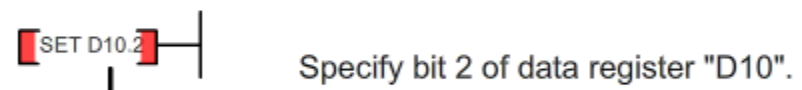
Example 1 When the bit 5 of data register "D0" is 1.



When the bit 5 of the data register "D0" is 0.



Example 2

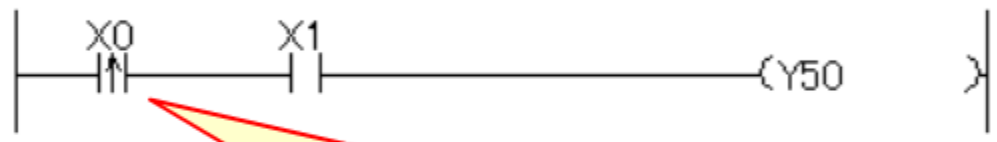


As D10.2 is "0", the value is inverted to "1 (ON)".

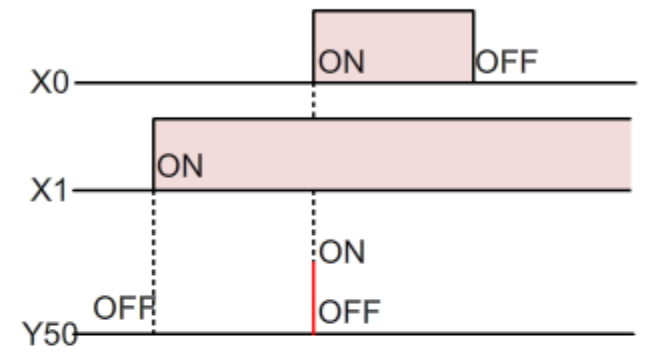
1.1.2 Leading or trailing edge specification for contacts

For ON/OFF operation of contacts, a signal can be set to turn ON for only 1 scan at the leading or trailing edge of a contact. This is useful to program a leading-edge or trailing-edge signal input condition.

Program example for the leading-edge contact



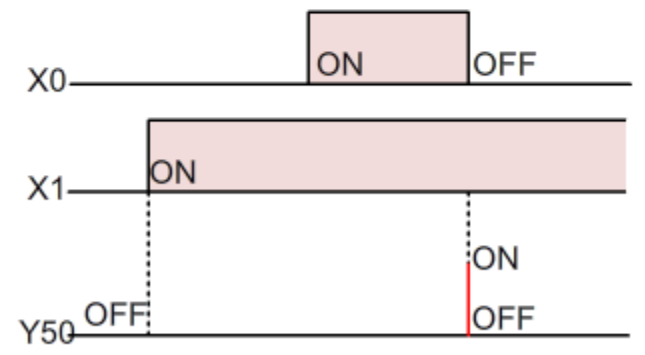
When the contact "X0" turns from OFF to ON, stays ON for one scan.



Program example for the trailing-edge contact



When the contact "X0" turns from ON to OFF, it stays ON for one scan.



1.2 How to Customize the Number of Device Points

Table of Contents

Different CPU modules have a different number of device points with their device numbers initially allocated, corresponding to the capacity of the used CPU module.

When the number of points allocated to a frequently-used device is insufficient, reduce the points allocated to other devices and use those points for the frequently-used device. Go to the Device tab of the PLC Parameter window to change the setting.

Example of device setting screen

| | Sym. | Dig. | Device Points | Local Device Start | Local Device End |
|-----------------|------|------|---------------|--------------------|------------------|
| Input Relay | X | 16 | 8K | | |
| Output Relay | Y | 16 | 8K | | |
| Internal Relay | M | 10 | 8K | | |
| Latch Relay | L | 10 | 8K | | |
| Link Relay | B | 16 | 8K | | |
| Annunciator | F | 10 | 2K | | |
| Link Special | SB | 16 | 2K | | |
| Edge Relay | V | 10 | 2K | | |
| Step Relay | S | 10 | 8K | | |
| Timer | T | 10 | 2K | | |
| Detective Timer | ST | 10 | 0K | | |
| | | | 1K | | |
| | | | 12K | | |
| | | | 8K | | |
| | | | 2K | | |
| Index | Z | 10 | 20 | | |

Device Points:

- At default, initial values are set.
- The values in the white cells are changeable.
- Set device points in 16 point units.
- 1K point means 1024 actual points of data.

If the capacity of the set device points exceeds the capacity of the CPU module, a message indicating to modify the setting appears.

Total Device Points:
Automatically converted in word units.

 Please set the total number of devices used in the sequence program so that it is 29 K words or less.

| | | |
|--------------|------|---------|
| Device Total | 28.8 | K Words |
| Word Device | 25.0 | K Words |
| Bit Device | 44.0 | K Bits |

The total number of device points is up to 29 K words.
Latch(1) : Able to clear the value by using a latch clear

Maximum number of device points = capacity of the CPU module
For example, the capacity of the CPU module Q06UDEHCPU is 29K words.
When using the local devices, please do the file setting at PLC file setting parameter.

Latch function

By using the latch function, the CPU module retains device values when it stops operation.

For example, if a momentary power failure longer than the allowable limit occurs, the CPU module retains data at the operation stop and uses them when restarting the sequence control.

If the latch function is not used, device values are reset to the default values (bit devices to OFF and word devices to "0") in the following events:

- (1) Power-Off
- (2) Reset by the "RUN/STOP/RESET" switch
- (3) Momentary power failure longer than the allowable limit in the power supply module

Setting of the latch range

Select the Device tab in the PLC Parameter window of GX Works2 to set the latch range. Below is a setting example to latch the latch relays L0 to L1024 and data registers D0 to D128.

| | | | | A | B | C | D | |
|-----------------|------|------|---------------|-----------------|---------------|-----------------|---------------|-----|
| | Sym. | Dig. | Device Points | Latch (1) Start | Latch (1) End | Latch (2) Start | Latch (2) End | Loc |
| Input Relay | X | 16 | 8K | | | | | |
| Output Relay | Y | 16 | 8K | | | | | |
| Internal Relay | M | 10 | 8K | | | | | |
| Latch Relay | L | 10 | 8K | | | 0 | 1024 | |
| Link Relay | B | 16 | 8K | | | | | |
| Annunciator | F | 10 | 2K | | | | | |
| Link Special | SB | 16 | 2K | | | | | |
| Edge Relay | V | 10 | 2K | | | | | |
| Step Relay | S | 10 | 8K | | | | | |
| Timer | T | 10 | 2K | | | | | |
| Retentive Timer | ST | 10 | 0K | | | | | |
| Counter | C | 10 | 1K | | | | | |
| Data Register | D | 10 | 12K | | | 0 | 128 | |
| Link Register | W | 16 | 8K | | | | | |
| Link Special | SW | 16 | 2K | | | | | |
| Index | Z | 10 | 20 | | | | | |

| | | |
|---|-----------------|--|
| A | Latch (1) Start | Specify the start number of the latch range to be set. |
| C | Latch (2) Start | |
| B | Latch (1) End | Specify the end number of the latch range to be set. |
| D | Latch (2) End | |

* See the next page for the difference between latches (1) and (2).

1.3 Saving the Device Status at Power-Off or Reset

How to clear the latch data

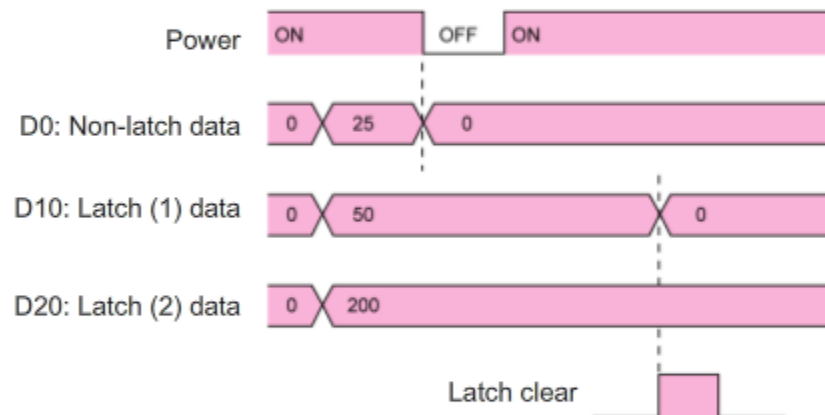
The clear method for latches (1) and (2) is different.

Latch (1): Clears the latched data from the Remote Operation window of GX Works2. Use latch 1 when latched data needs to be cleared at the installation site.

Latch (2): Clears the latched data by a dedicated instruction of the program. Use latch 2 when the latched data do not have to be cleared at the installation site.

| | Sym. | Dig. | Device Points | Latch (1) Start | Latch (1) End | Latch (2) Start | Latch (2) End | Loca |
|-----------------|------|------|---------------|-----------------|---------------|-----------------|---------------|------|
| Input Relay | X | 16 | 8K | | | | | |
| Output Relay | Y | 16 | 8K | | | | | |
| Internal Relay | M | 10 | 8K | | | | | |
| Latch Relay | L | 10 | 8K | | | 0 | 1024 | |
| Link Relay | B | 16 | 8K | | | | | |
| Annunciator | F | 10 | 2K | | | | | |
| Link Special | SB | 16 | 2K | | | | | |
| Edge Relay | V | 10 | 2K | | | | | |
| Step Relay | S | 10 | 8K | | | | | |
| Timer | T | 10 | 2K | | | | | |
| Retentive Timer | ST | 10 | 0K | | | | | |
| Counter | C | 10 | 1K | | | | | |
| Data Register | D | 10 | 12K | 0 | 128 | | | |
| Link Register | W | 16 | 8K | | | | | |
| Link Special | SW | 16 | 2K | | | | | |
| Index | Z | 10 | 20 | | | | | |

Timing chart



How to clear latched data via remote operation

Select Online on the menu bar of GX Works2, then select Remote Operation.



In this chapter, you have learned:

- Specifying the devices
- How to customize the number of device points
- Saving the device status at power-off or reset

Point

| | |
|--------------------------------------|---|
| Changing the number of device points | <ul style="list-style-type: none">• Different CPU modules have a different number of device points, and their device numbers are initially allocated in accordance with the CPU module capacity.• When the number of points allocated to a frequently-used device is insufficient, reduce the points allocated to other devices and use those points for the frequently-used device. |
| Latch function | The latch function of the CPU module retains device values at power-off or reset and uses the retained data when the operation restarts. The retained values are cleared by the latch clear. |

Chapter 2 How to Use Devices with Various Functions

This chapter explains about the devices with various embedded functions.

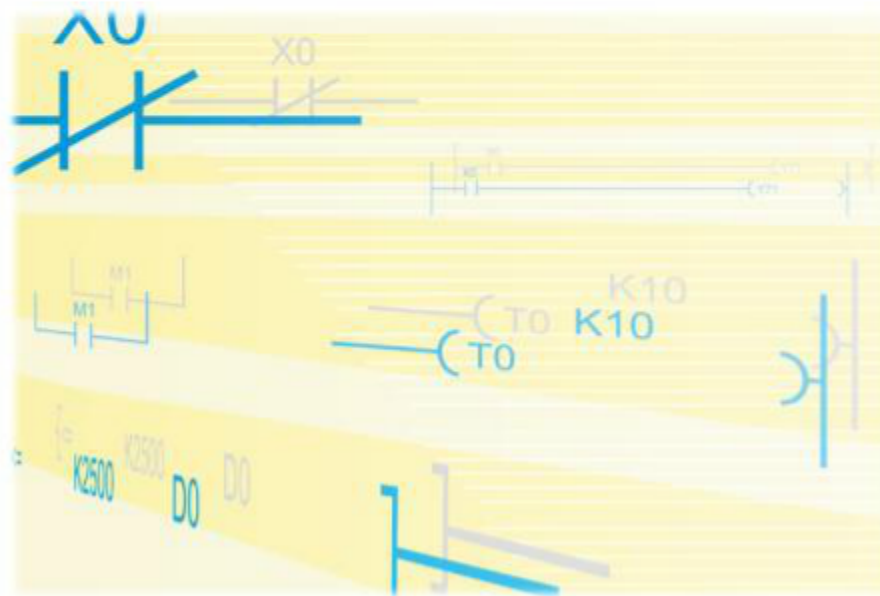
Unlike the devices such as data registers, which can only store values, devices such as the retentive timer or index register have functions of their own.

Section 2.1: How to Use the Retentive Timer

Section 2.2: How to Use the Index Register

Section 2.3: How to Use the Special Relay and Special Register

Section 2.4: Summary



2.1 How to Use the Retentive Timer

2.1.1 Difference between timers and retentive timers

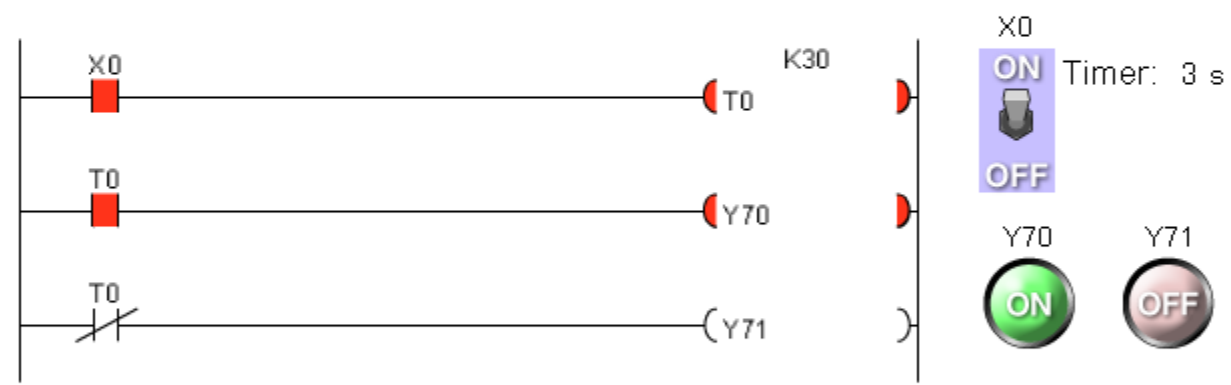
A timer and a retentive timer are both used in sequence programs for an operation involving time measurement.
 * Details of the timers are explained in the MELSEC-Q Series Basics Course.

(a) Timer

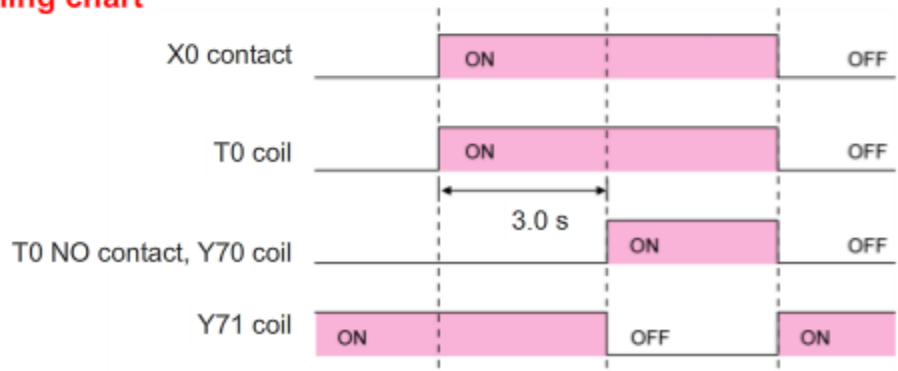
A timer turns ON a contact at certain timing after turn-ON of a coil. When the coil turns OFF, the timer's value is reset to "0". The device symbol for the timer is "T".

Ladder program and operation

Turn the switch ON/OFF to see how the timer operates.
 At 3 seconds after X0 changes ON, Y70 will also changes to ON and Y71 will change to OFF state.



Timing chart



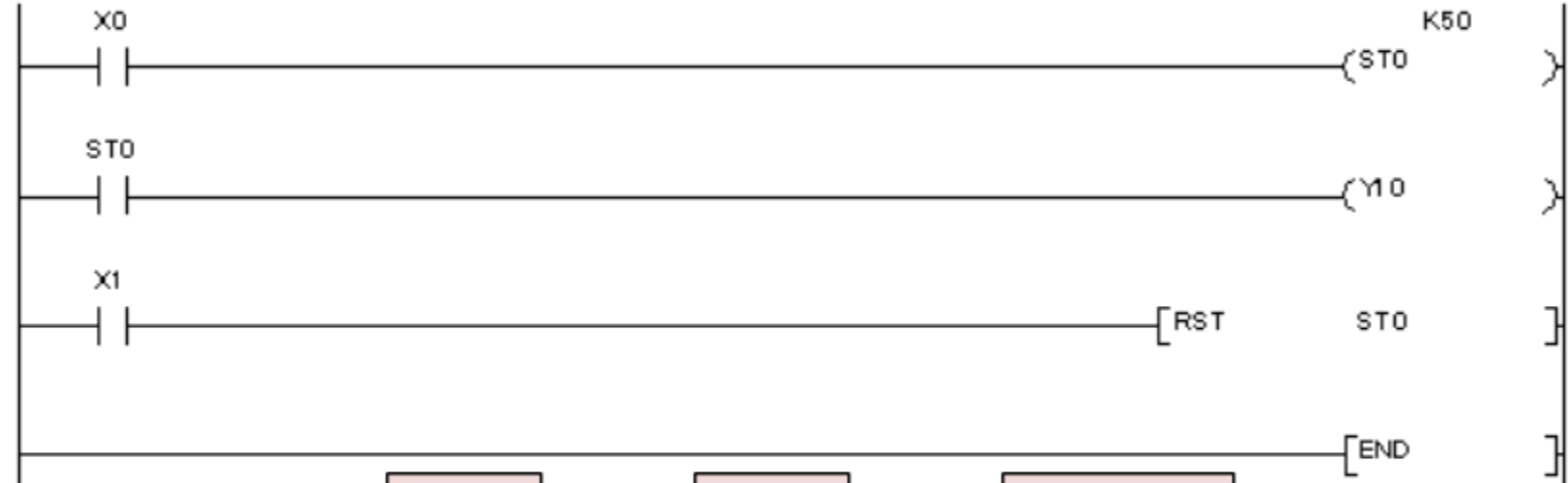
2.1.1 Difference between timers and retentive timers

(b) Retentive timer

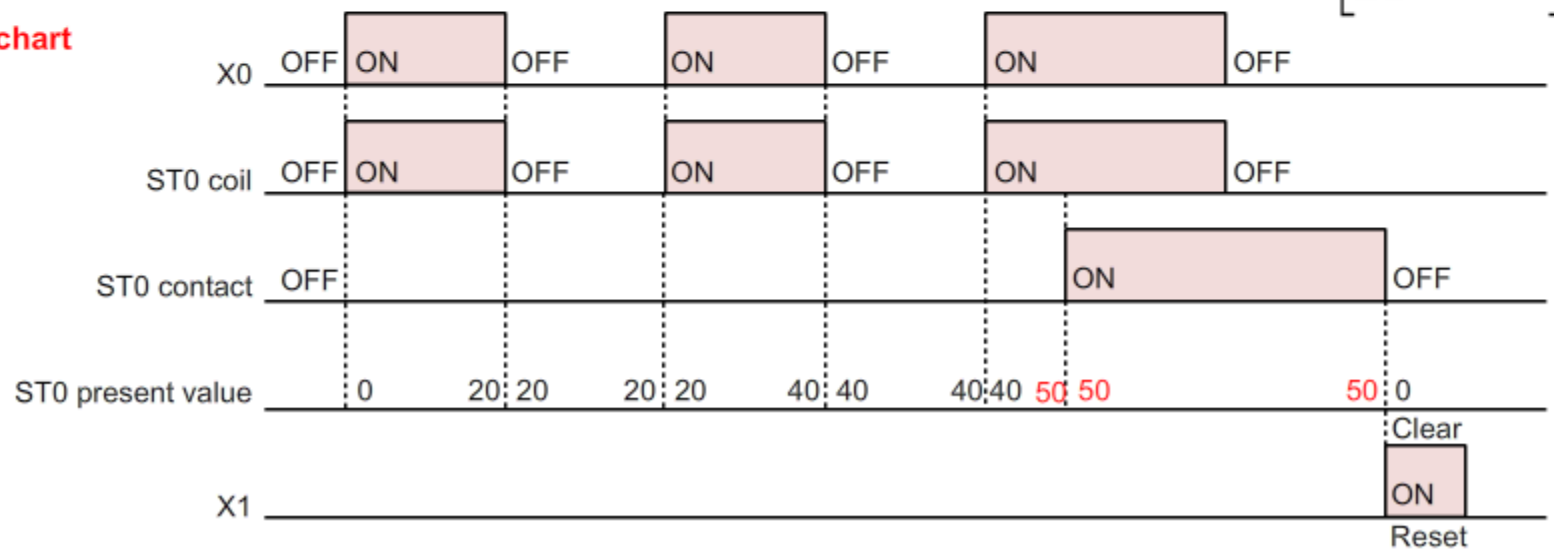
A retentive timer is useful in measuring the total operation time.

A retentive timer turns ON a contact (OFF to ON) at certain timing after turn-ON of a coil. When the coil turns OFF, the timer's value is not reset and is retained. When the coil turns ON again, the timer restarts measuring from the retained value. The device symbol of a retentive timer is "ST".

Ladder program



Timing chart

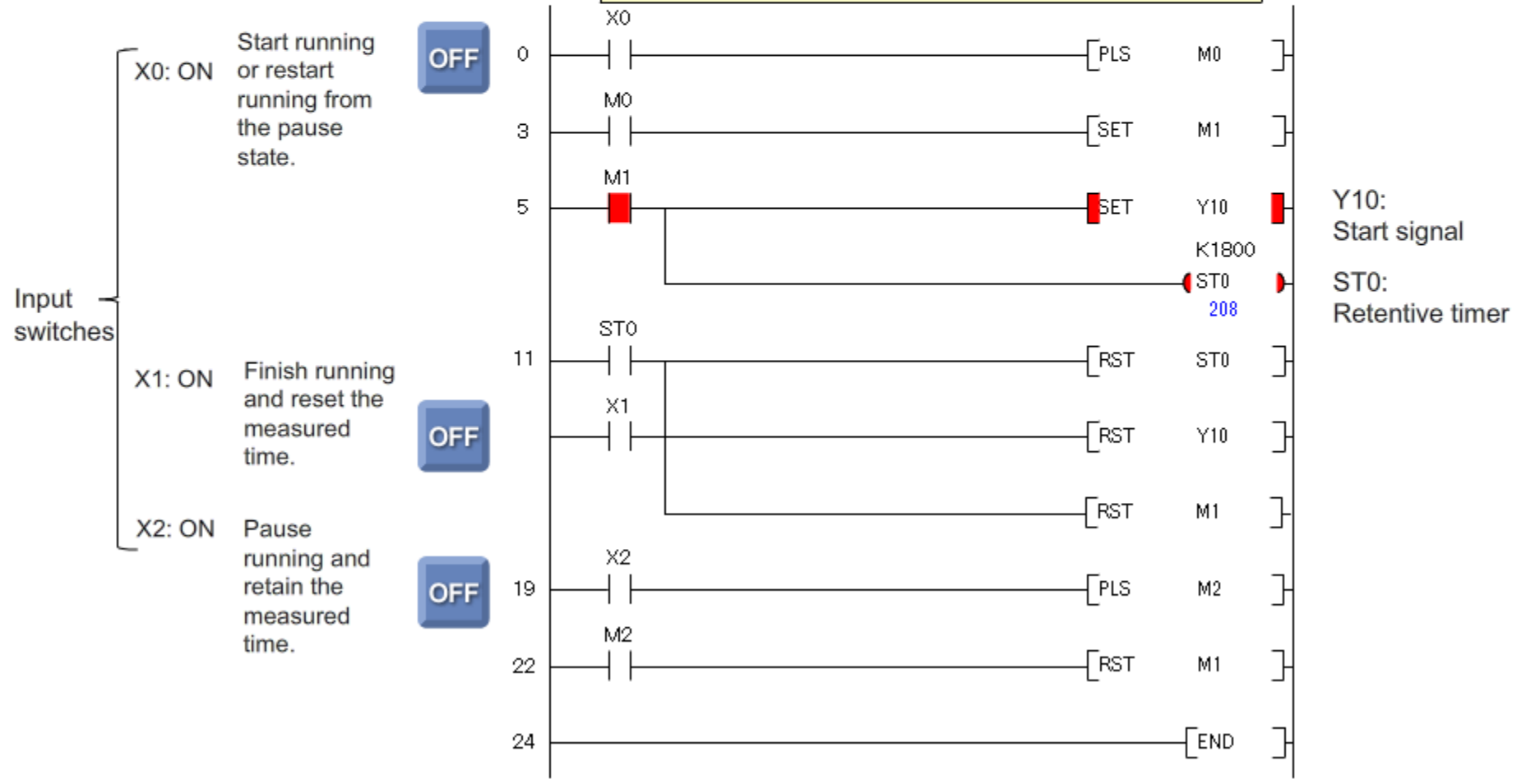


2.1.2 Retentive timer operation

Let's look at how a retentive timer operates on a running machine that uses input switches (X0 to X2).
 * The retentive timer (ST0) is set in 100 ms unit.



Timer ST0 is set to K1800 = 180,000 ms (3 min.)/100 ms
Running time
 (Time measured by the timer) 20 sec.



2.1.3

Preparation to use the retentive timer

The number of points used by a retentive timer is initially "0". To use a retentive timer, some points need to be allocated. Open the PLC Parameter window of GX Works2, select the Device tab, and set the number of device points used by the retentive timer.

Below is a setting example for using ST0 to ST63 (64 points) for the retentive timer.

| | Sym. | Dig. | Device Points | Latch (1) Start | Latch (1) End | Latch (2) Start | Latch (2) End | Local Device Start | Local Device End |
|-----------------|------|------|---------------|-----------------|---------------|-----------------|---------------|--------------------|------------------|
| Input Relay | X | 16 | 8K | | | | | | |
| Output Relay | Y | 16 | 8K | | | | | | |
| Internal Relay | M | 10 | 8K | | | | | | |
| Latch Relay | L | 10 | 8K | | | | | | |
| Link Relay | B | 16 | 8K | | | | | | |
| Annunciator | F | 10 | 2K | | | | | | |
| Link Special | SB | 16 | 2K | | | | | | |
| Edge Relay | V | 10 | 2K | | | | | | |
| Step Relay | S | 10 | 8K | | | | | | |
| Timer | T | 10 | 2K | | | | | | |
| Retentive Timer | ST | 10 | 64 | | | | | | |
| Counter | C | 10 | 1K | | | | | | |
| Data Register | D | 10 | 12K | | | | | | |
| Link Register | W | 16 | 8K | | | | | | |
| Link Special | SW | 16 | 2K | | | | | | |
| Index | Z | 10 | 20 | | | | | | |

Device Total K Words

Word Device K Words

Bit Device K Bits

The total number of device points is up to 29 K words.

Latch(1) : Able to clear the value by using a latch clear.

Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program. Scan time is extended by the latch range setting (including L).

If the latch is necessary, please set the required minimum latch range.

When using the local devices, please do the file setting at PLC file setting parameter.

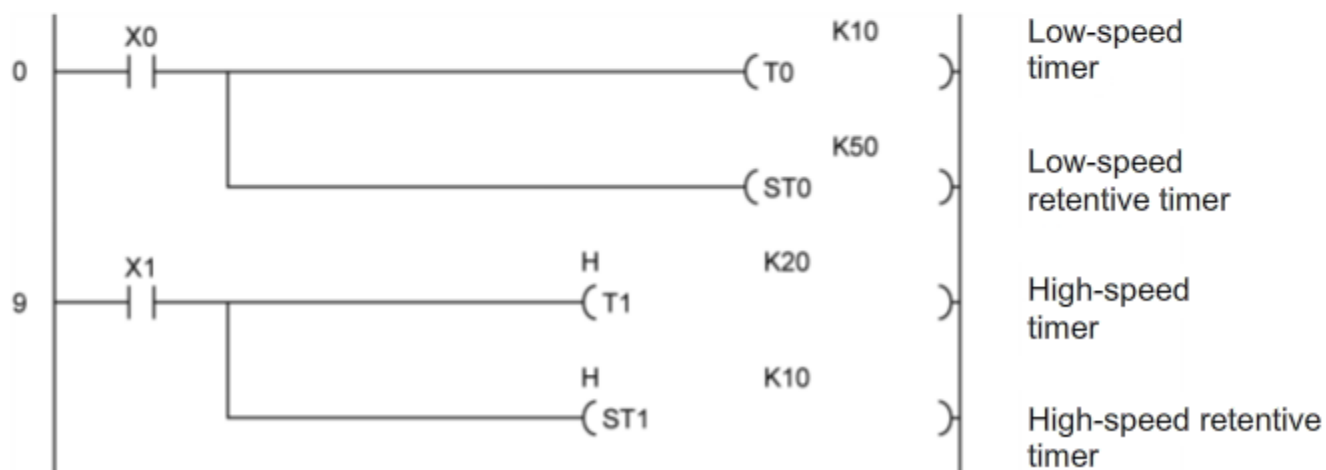
2.1.4

Difference between the low-speed timer and high-speed timer

| | Unit | Program example | Operation |
|----------------------------|---------------------|-------------------------------|--|
| Low-speed timer | 100 ms (default) | $\{ T0 \quad K50 \}$ | The low-speed timer T0 counts 5 sec. |
| High-speed timer | 10 ms (default) | $\{ T1 \quad H \quad K50 \}$ | The high-speed timer T1 counts 0.5 sec. |
| Low-speed retentive timer | 100 ms (default) | $\{ ST0 \quad K50 \}$ | The low-speed retentive timer ST0 counts 5 sec. |
| High-speed retentive timer | 10 ms (default) | $\{ ST1 \quad H \quad K50 \}$ | The high-speed retentive timer ST1 counts 0.5 sec. |

The initial unit for the time measurement is 100 ms for the low-speed timer and 10 ms for the high-speed timer. See the next page for how to change the unit.

Below is a sample ladder program including timers.



2.1.4

Difference between low-speed timers and high-speed timers

How to change the unit of the timer

Change the Timer Limit Setting on the PLC System tab of the PLC Parameter window. Below is a setting example on the PLC System screen.

| Timer Limit Setting | | |
|---------------------|------------------------------------|--------------------|
| Low Speed | <input type="text" value="100"/> | ms (1ms--1000ms) |
| High Speed | <input type="text" value="10.00"/> | ms (0.01ms--100ms) |

Unit for the low-speed timer

Unit for the high-speed timer

An index register "Z", in combination with another device, specifies (indexes) a device number of a device to be controlled. An index register is useful to simplify programs because it can describe multiple devices in a batch.

- When an index register is used, it is written after a device symbol and a device number as shown below to indicate an actual control target device.

Actual control target device = device symbol (device number + index register)

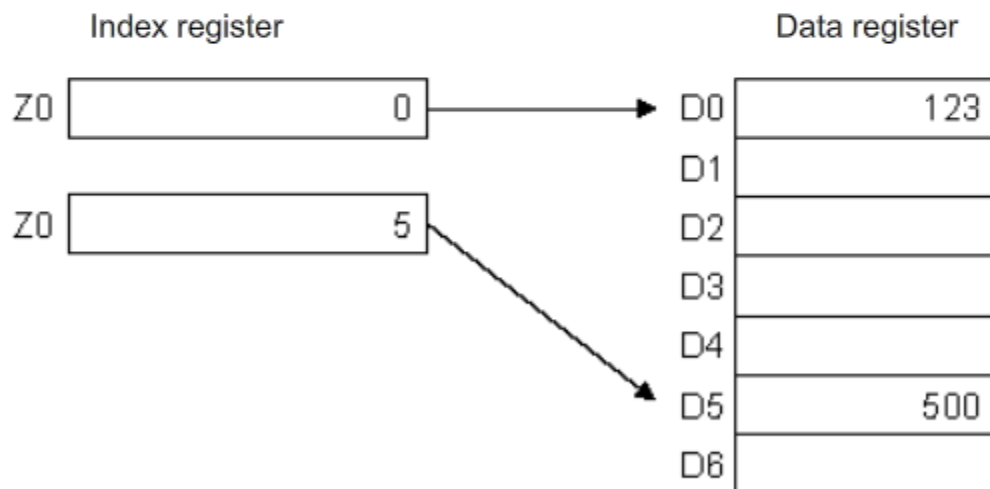
- 16 points, from Z0 to Z15, can be used for an index register.

Example of the index register

When a device is stated as "D0Z0", it means D (0 + Z0), so the device number is "0 + (value of Z0)".

Example: When Z0 = 0, the device number is D0.

When Z0 = 5, the device number is D5.

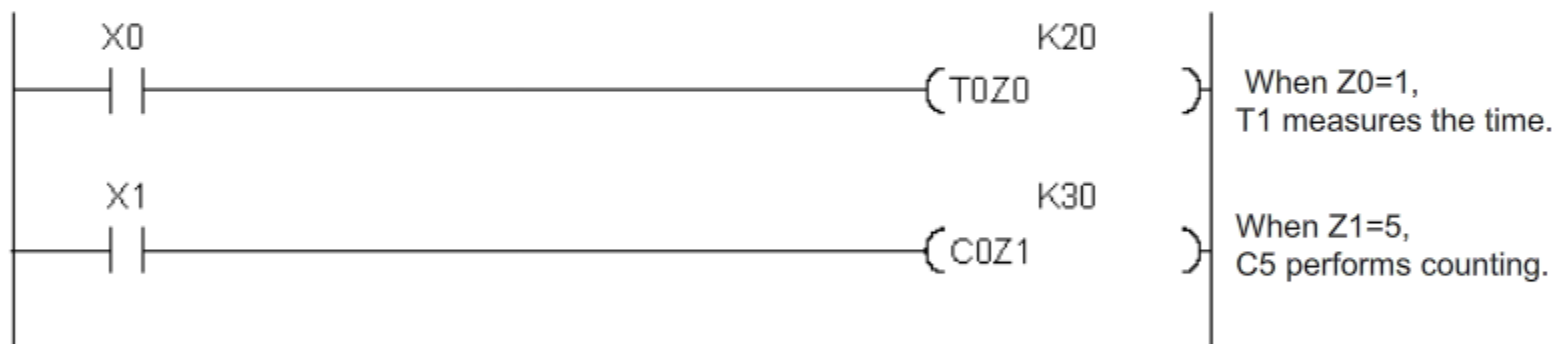


Devices that can be indexed by index registers

The following devices can be indexed using index registers:

| | |
|-------------|---------------------|
| Bit device | X, Y, M, L, S, B, F |
| Word device | T, C, D, R, W |
| Constant | K, H |
| Pointer | P |

Note: For the contacts and coils used in timers and counters, only the index registers Z0 and Z1 are available.



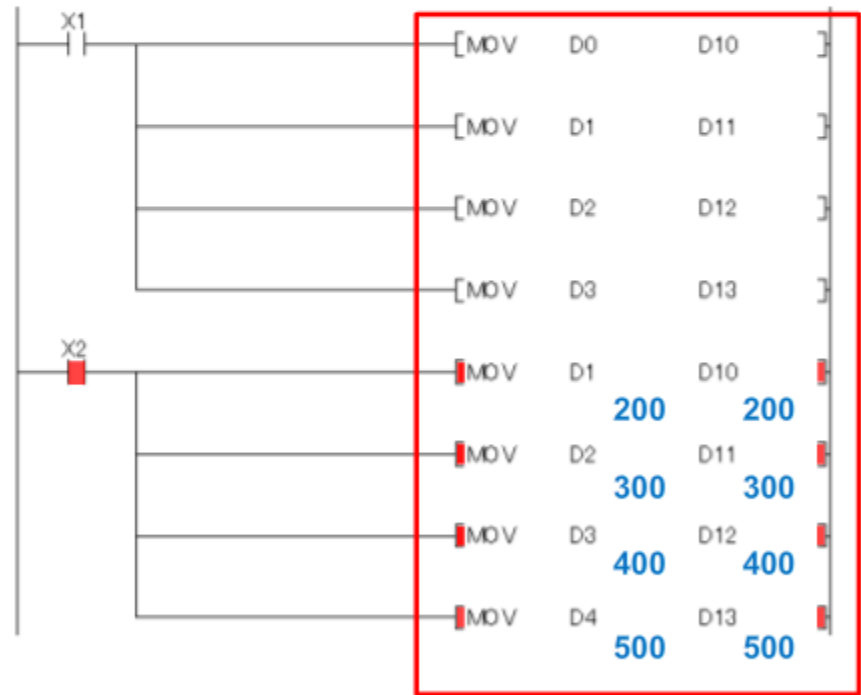
2.2 How to Use the Index Register

Simplification of programs using index registers

The programs shown below transfer the values in "D0 to D4" to "D10 to D13" when X1 or X2 turns ON. The programs (1) and (2) will bring the same result.
 In the program (1), the data are transferred directly.
 In the program (2), the data are transferred via the index register.

Initial values
 D0=100
 D1=200
 D2=300
 D3=400
 D4=500

(1) Example without index registers



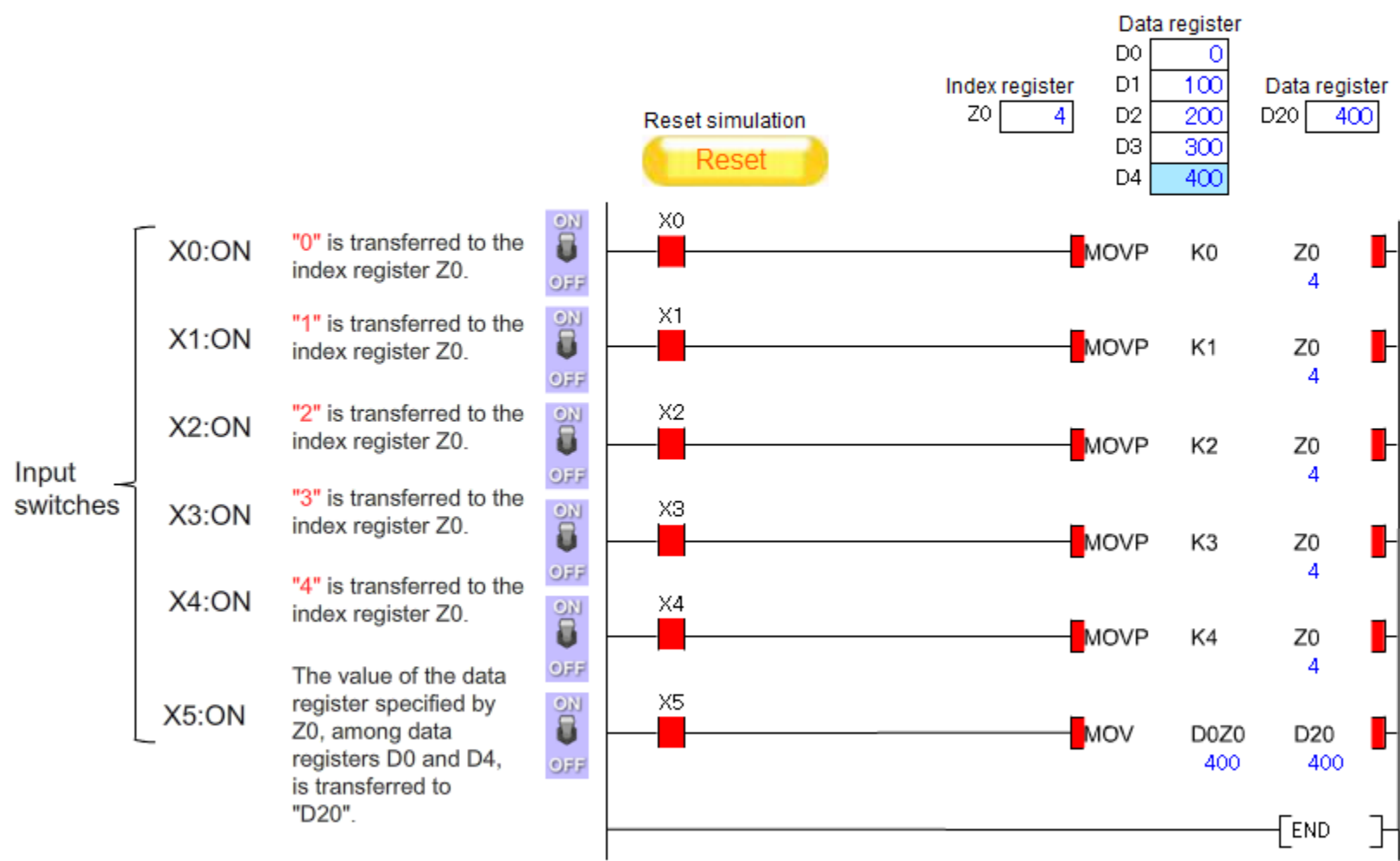
(2) Example using index registers



Simplification of the programs

2.2.1 Index register operation

Click the input switches X0 to X5 to see how the index register Z0 operates.
 *K0 to K400 are already stored in the data registers D0 to D4.



The special relays and special registers used in the CPU module have predetermined functions and operations. The internal relays used for bit information (ON/OFF) are called "special relays (SM)", and the internal registers used for word information are called "special registers (SD)".

In programs, they are used as judgment conditions for operations. They are also used as monitoring conditions, which can be specified on the device monitor of GX Works2.

The special relays and special registers are categorized by their types as below.

Diagnostic information: SM0 to 199, SD0 to 199

Stores the diagnostic results of the CPU module.

Various diagnostic errors and error codes

System information: SM200 to 399, SD200 to 399

Stores the system information of the CPU module.

CPU module information, clock data, etc.

System clock/counter: SM400 to 499, SD400 to 499

Stores clock signals and count values that are used as basic timing elements.

Various clock signals

Scan information: SM500 to 599, SD500 to 599

Stores scan execution information of the programs.

Various scan time information

Memory card information: SM600 to 699, SD600 to 699

Stores card information such as usage of memory cards and file registers.

Memory card enabled/disabled

Instruction information: SM700 to 799, SD700 to 799

Stores execution status and control information regarding special instructions.

Instruction execution flags

Debug information: SM800 to 899, SD800 to 899

Stores the information concerning debugging.

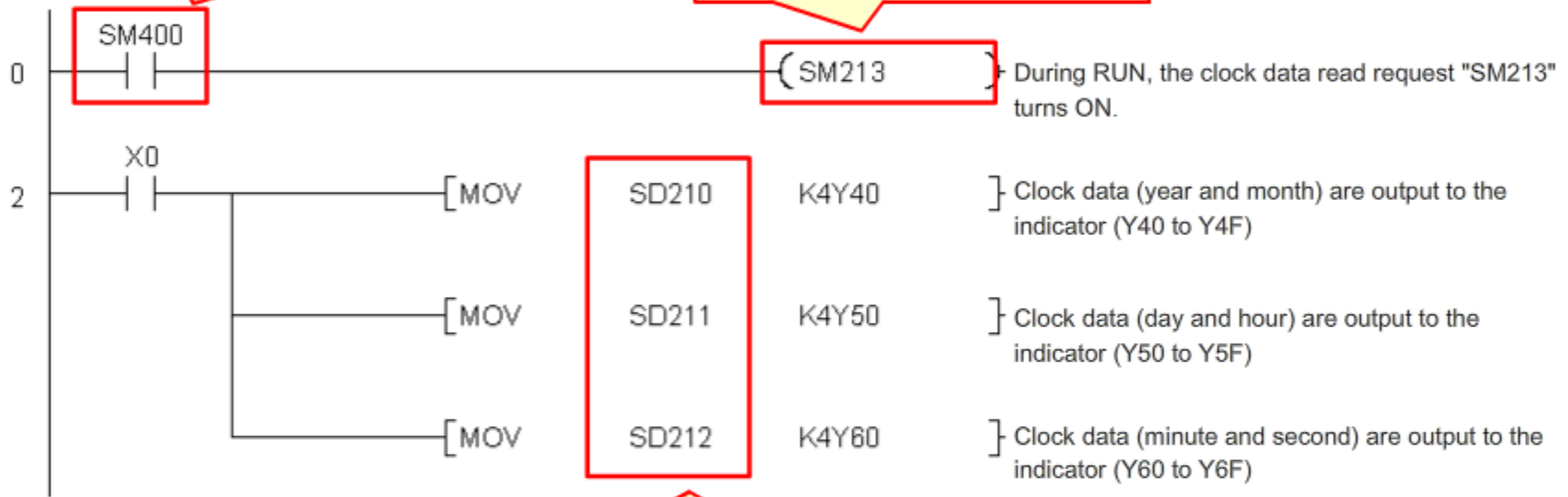
Trace status monitoring

2.3.1 Sample program using special relays/registers

Sample program to request to read the clock data of the CPU module.

Special relay (normally ON)

Relay that requests to read the clock data of the CPU module



Clock data are normally stored in the special registers (SD210 to SD212).

In this chapter, you have learned:

- How to use the retentive timer
- How to use the index register
- How to use the special relay and special register

Point

| | |
|---|---|
| Usage of the retentive timer | <ul style="list-style-type: none">• To use a retentive timer, some points must be allocated in the PLC Parameter window.• The measured time (present value) and the contact status (ON/OFF) of the retentive timer are not cleared even if the condition changes to dissatisfy the input condition after a timeout.• The program requires a ladder to reset the retentive timer. (The RST instruction is used.) |
| Usage of the index register | <ul style="list-style-type: none">• The index register "Z" is indicated following a device used in the program. For example, "D0Z5".• 16 points, from Z0 to Z15, are available for index registers. |
| Functions of the special relays and special registers | The special relays and special registers are used to indicate the internal condition of the CPU module, including diagnostic information and system information. |

Chapter 3 Memory for the CPU Module and the File Register

This chapter explains about memory available for the CPU module and how to use the file register.

Section 3.1: Memory for the CPU module

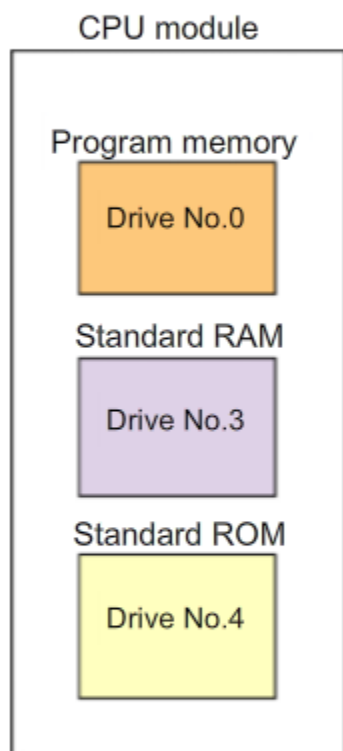
Section 3.2: How to Use the File Register

Section 3.3: Summary

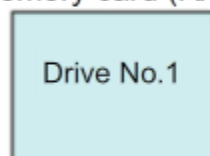


3.1 Memory for the CPU Module

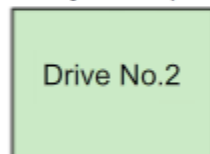
A CPU module can use two types of memory; one that is built-in into the CPU module, and a memory card that can be inserted into a slot of the CPU module. To ensure CPU module access by GX Works2, drive numbers, which indicate target memory types, must be specified correctly.



Memory card (RAM)



Memory card (ROM)



| Memory type | | Stored data type | Data condition at power-off | Memory formatting for the first-time use |
|-------------|----------------|---|--|--|
| CPU module | Program memory | •Parameter •Program | Retained by using the battery of the CPU module | Required (Use GX Works2) |
| | Standard RAM | •File register •Local device | | |
| | Standard ROM | •Parameter •Program | Retained without using a battery | Not required |
| Memory card | RAM | •Parameter •Program •File register •Local device | Retained by using the battery of the memory card | Required (Use GX Works2) |
| | ROM | •Parameter •Program •File register | Retained without using a battery | No flash card required. ATA card is required (Use GX Works2). |

- Programs stored in the standard ROM or a memory card is booted (loaded) to the program memory of the CPU module and executed when the CPU module is started.
- When a file register is saved in the standard RAM, the access speed to the file register is as fast as accessing the data register (D).
- When the standard RAM is used, turning the power off without a backup battery clears the data stored in the RAM.
- In general, the high-speed read/write RAM is used for starting up the system and the ROM is used for continuous system operations.

3.2 How to Use the File Register

Overview of the file register

- A file register is a word device used to extend data registers (D).
- Compared to a data register, a file register can hold large amounts of data.
- A file register is stored in the standard RAM of the CPU module or in a memory card (RAM).
- The data stored in a file register will not be cleared even when the power is turned off or the CPU module is reset.
- The device symbol is "ZR".

Operation of the ladder program

Turn the power switch and input switches ON/OFF to simulate the operation of file registers.

Toggle the power supply switch from ON to OFF, and then back ON again to check that the data in the file registers ZR0 and ZR1 are retained.

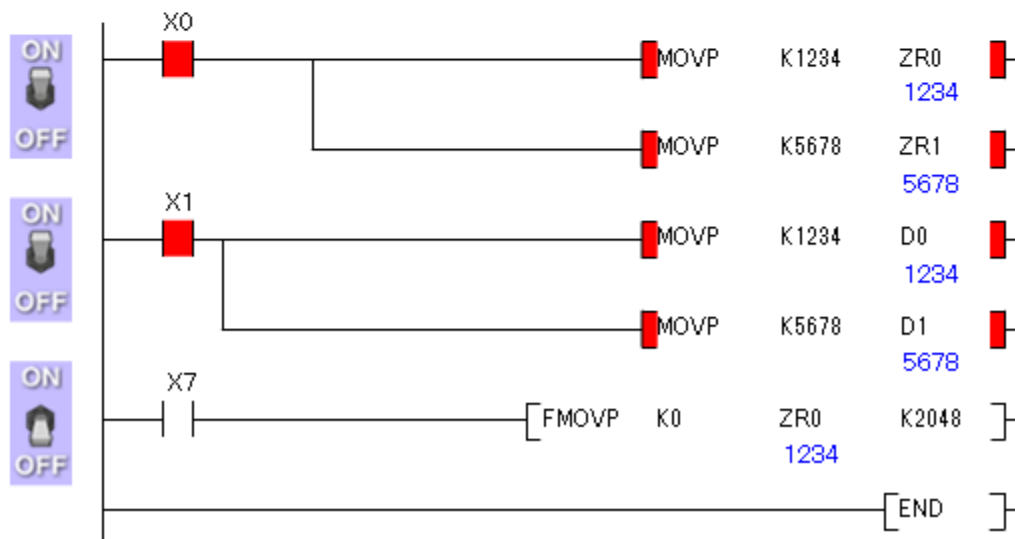


Reset simulation



| | File register | Data register |
|-----|---------------|---------------|
| ZR0 | 1234 | D0 1234 |
| ZR1 | 5678 | D1 5678 |

- Input switches
- X0: ON The data are written to the file registers ZR0 and ZR1.
 - X1: ON The data are written to the data registers D0 and D1.
 - X7: ON The data in the file register ZR0 and ZR1 are set back to "0".



3.2 How to Use the File Register

This section explains about the setting that designates a local file register as the storage destination. In the PLC Parameter window, select the PLC File tab. Then, select "Use the same file name as the program" for the File Register and specify the storage destination memory.

Note that a memory card is required for this setting. (The standard RAM can store only one file register.)

File Register

Not Used

Use the same file name as the program

Corresponding Memory

Use the following file

Corresponding Memory

File Name

Capacity K Points
(1K--4086K Points)

Transfer to Standard ROM at Latch data backup operation.

Following settings are available in device setting when select "Use the following file" and specify capacity.

- Change of latch(2) of file register.
- Assignment to expanded data register/expanded link register of part of file register area.

For "Corresponding Memory", select "Memory card (RAM)".

The file register setting must be made for each program. The settings will be written to the CPU module at PLC write.

3.3

Summary

In this chapter, you have learned:

- Memory for the CPU module
- How to use the file register

Point

| | |
|----------------------------|--|
| Usage of the file register | To use a file register, either the standard RAM of the CPU module or a memory card must be selected as the data storage destination. For the setting, go to the PLC File tab of the PLC Parameter window. The file register retains data even when the power is OFF. |
|----------------------------|--|

Chapter 4 Programs with Real Numbers

This chapter explains how programs handle real numbers and operation instructions.

Section 4.1: Application and Notation of Real Numbers

Section 4.2: Real Number Operation Instruction

Section 4.3: Conversion Instructions between Integers and Real Numbers

Section 4.4: Summary



Application of real numbers

- "Real numbers" are numeric values including decimal points.
- Sequence programs are normally configured using integers. However, real numbers with decimal points are required in programs for advanced arithmetic operations such as for trigonometric function and exponent operations.
- Numeric data of real numbers are referred to as "floating-point data".

Precautions

- One real number **always uses two consecutive word devices** (occupy 32-bit memory space), regardless of the size of the number.
- In sequence programs, **dedicated operation instructions** (addition, subtraction, multiplication, division, special functions, etc.) that handle real numbers are available. Conversion instructions, for example between integers and real numbers, are also available.

Notation for real numbers

"E" is used to represent a real number.

(1) Expressing a constant with real numbers

To write a constant, start with "E".

| | |
|------------------------|---|
| Normal expression | Write a numeric value as is. (Example) 10.2345 as "E10.2345". |
| Exponential expression | Write a numeric value as "(numeric value) × 10 ⁿ ". (Example) 1234.0 as "E1.234+3". |

(2) Instruction with a real number

Add "E" in front of an instruction.

For instance, a transfer instruction is "EMOV", and the addition or subtraction instruction is "E+" or "E-".

4.2 Real Number Operation Instruction

4.2.1 Addition and subtraction instructions

| Instruction code | Ladder example | |
|------------------|--|--|
| E+ (Addition) | Real number operation "D + S = D" is executed. | Real number operation "S1 + S2 = D" is executed. |
| E- (Subtraction) | Real number operation "D - S = D" is executed. | Real number operation "S1 - S2 = D" is executed. |

S (source): Data before operation (constant, device number)

D (destination): Destination of data after operation (device number)

P: Instruction to be executed on leading edge

S1 and S2: Two data items to be operated.

Note:

In real number operations, S1, S2 and D in the ladder must all be real numbers.

Integers and real numbers cannot be mixed for an operation.

4.2.1 Addition and subtraction instructions

Program example with the addition instruction

[E+(P) D0 D10]

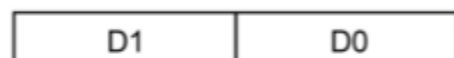
Floating-point real number (32 bits)



2.54

+

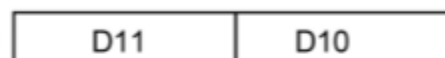
Floating-point real number (32 bits)



10.55

=

Floating-point real number (32 bits)



13.09

[E+(P) D0 D10 D20]

Floating-point real number (32 bits)



1000.000

+

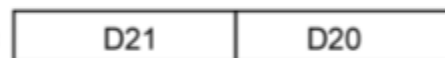
Floating-point real number (32 bits)



3.140

=

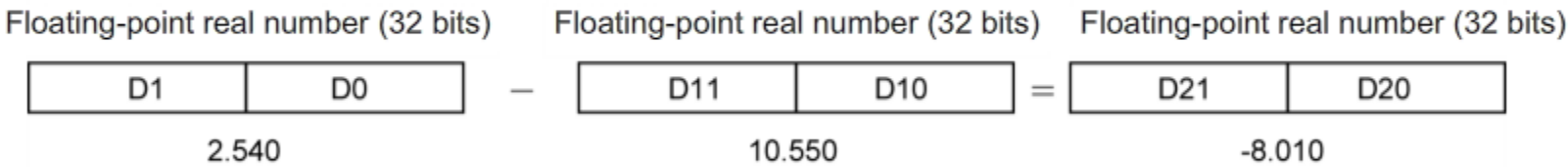
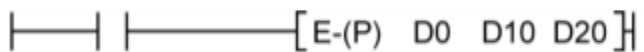
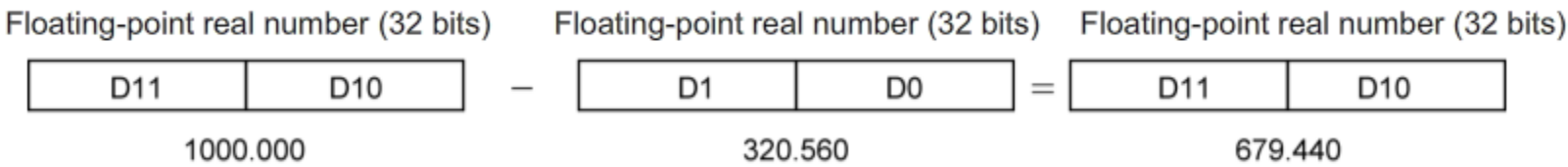
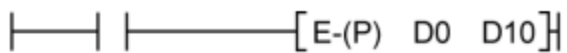
Floating-point real number (32 bits)



1003.140

4.2.1 Addition and subtraction instructions

Program example with the subtraction instruction



4.2.2

Multiplication and division instructions



| Instruction code | Ladder example |
|---------------------|--|
| E* (Multiplication) | Real number operation "S1 * S2 = D" is executed. |
| E/ (Division) | Real number operation "S1 / S2 = D" is executed. |

S1, S2 (source): Two data items to be operated

D (destination): Destination of data after operation (device number)

P: Instruction to be executed on leading edge

Note:

In real number operations, S1, S2 and D in the ladder must all be real numbers. Integers and real numbers cannot be mixed for an operation.

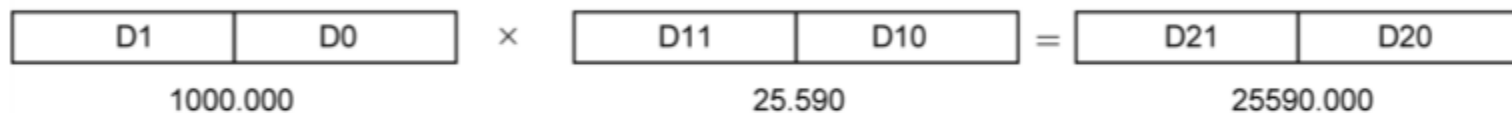
4.2.2

Multiplication and division instructions

Program example with the multiplication instruction

Floating-point real number (32 bits)

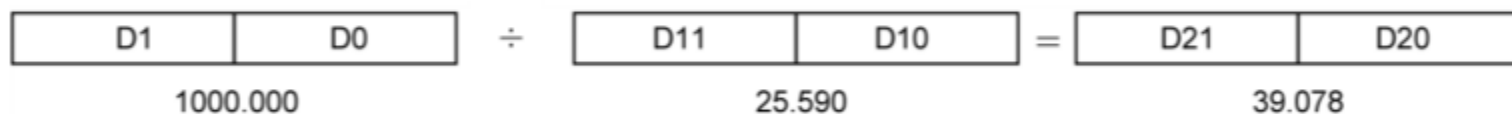
Floating-point real number (32 bits) Floating-point real number (32 bits)



Program example with the division instruction

Floating-point real number (32 bits)

Floating-point real number (32 bits) Floating-point real number (32 bits)



4.3

Conversion Instructions between Integers and Real Numbers

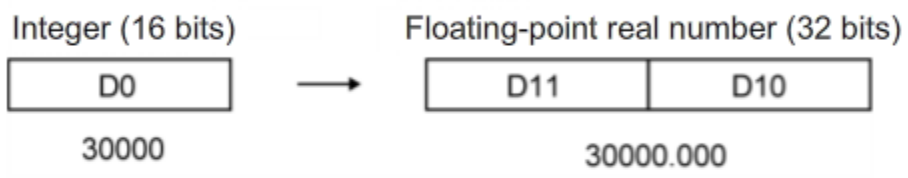
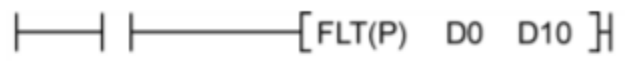
| Instruction code | Ladder example | |
|---|--|--|
| FLT (Integer to real number conversion) | <p>An integer (16 bits) is converted to a real number (32 bits).</p> | <p>An integer (32 bits) is converted to a real number (32 bits).</p> |
| INT (Real number to integer conversion) | <p>A real number (32 bits) is converted to an integer (16 bits).</p> | <p>A real number (32 bits) is converted to an integer (32 bits).</p> |

S (source): Data before operation (constant, device number)

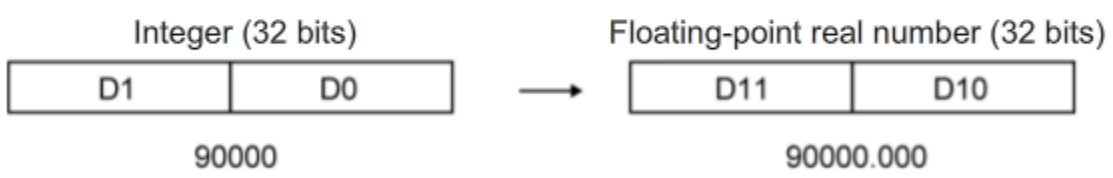
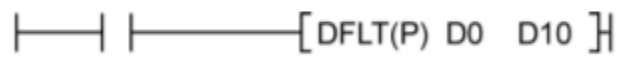
D (destination): Destination of data after operation (device number)

4.3 Conversion Instructions between Integers and Real Numbers

Program example with the integer (16 bits) / real number (32 bits) conversion instruction



Program example with the integer (32 bits) / real number (32 bits) conversion instruction



4.4

Summary

In this chapter, you have learned:

- Application and notation of real numbers
- Real number operation instruction
- Conversion instructions between integers and real numbers

Point

Real number
operation

- Real number data uses 2-word (32-bit) memory.
- Add E in the front of a real number operation instruction, such as E* (multiplication).
- An integer and a real number cannot be processed together. An integer must be converted into a real number before processing an arithmetic operation.

Chapter 5 Concept of I/O Numbers and How to Use The I/O Assignment Function

This chapter explains about the concept of I/O numbers and how to use the I/O assignment function.

Section 5.1: Concept of I/O Numbers

Section 5.2: I/O Numbers for the Extension Base Unit

Section 5.3: I/O Number Assignment Check on the System Monitor

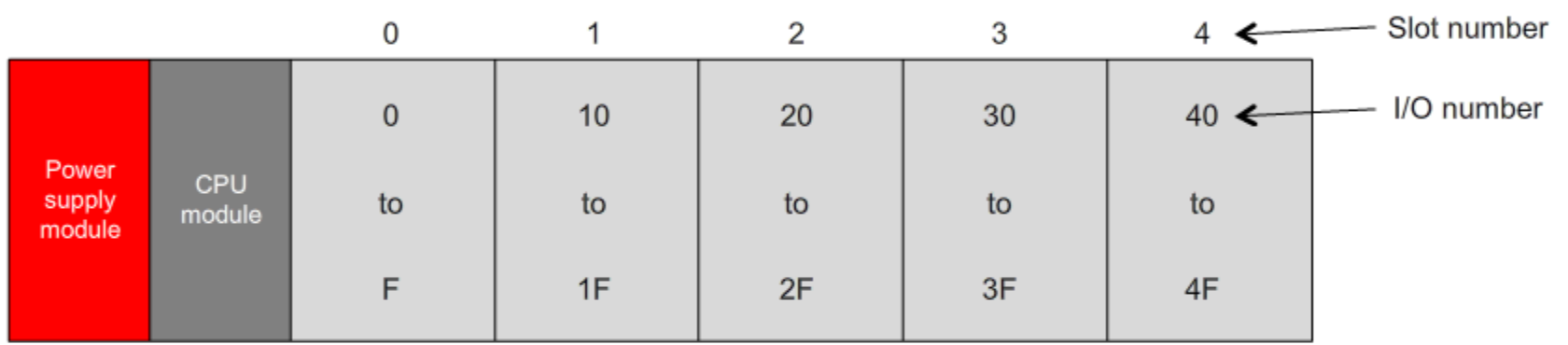
Section 5.4: How to Use the I/O Assignment Function

Section 5.5: Summary



5.1 Concept of I/O Numbers

I/O numbers are assigned to I/O modules on a base unit as shown below.
 (There are three types of I/O modules: 16, 32, and 64 point types. The example shown below uses 16 point type I/O modules.)



(Example) Q35B base unit with five I/O slots

I/O numbers (hexadecimal 0 to F) are assigned to each slot (module) sequentially, starting from the slot closest to the CPU module. Each slot (module) is assigned by default with 16 I/O numbers.

5.1 Concept of I/O Numbers

When 16, 32, and 64 point I/O modules are used together, the I/O numbers are assigned as follows:

| | | 0 | 1 | 2 | 3 | 4 | ← Slot number |
|---------------------|------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Power Supply module | CPU module | 16 point type | 32 point type | 64 point type | 32 point type | 16 point type | |
| | | 0 | 10 | 30 | 70 | 90 | ← I/O number |
| | | to | to | to | to | to | |
| | | F | 2F | 6F | 8F | 9F | |

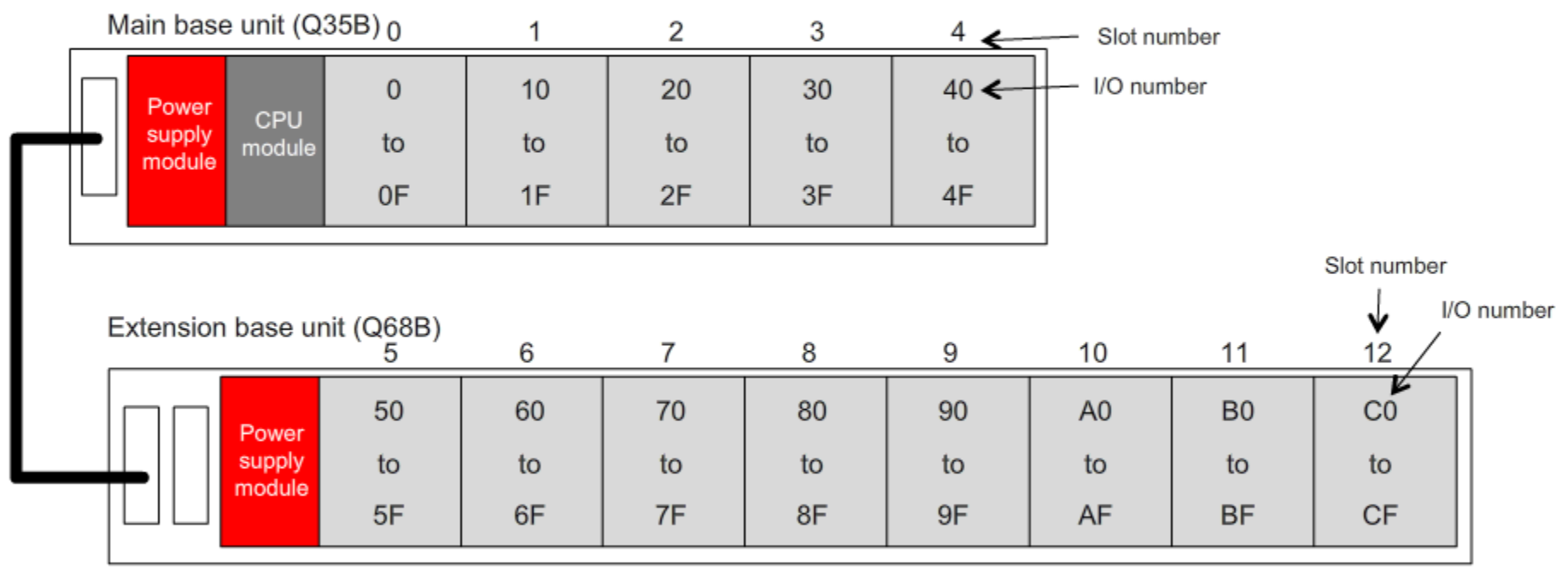
If there is an empty slot in the middle of the base unit, I/O numbers are also assigned to the slot. (In the initial setting.)

| | | 0 | 1 | 2 | 3 | 4 | ← Slot number |
|---------------------|------------|---------------|---------------|---------------|------------|---------------|---------------|
| Power Supply module | CPU module | 16 point type | 32 point type | 64 point type | Empty slot | 16 point type | |
| | | 0 | 10 | 30 | 70 | 80 | ← I/O number |
| | | to | to | to | to | to | |
| | | F | 2F | 6F | 7F | 8F | |

NOTE: 16 I/O numbers (hexadecimal) are assigned to an empty slot by default. However, the setting can be changed, and I/O numbers in the range of 0 to 64 can be set in 16 point units.

5.2 I/O Numbers for the Extension Base Unit

Each module's I/O numbers, which correspond with the input (X)/output (Y) relays of the CPU module, are automatically assigned by detecting the modules on the base unit.
 The I/O numbers of modules on the extension base unit are also automatically assigned, following the last I/O number of the main base unit.
 The following figure shows how I/O numbers are assigned using 16 point modules.



5.3 I/O Number Assignment Check on the System Monitor

To check the I/O number assignment, go to the menu of GX Works2, select Diagnostics, then select System Monitor.

(2) Check the start I/O numbers of the modules on the selected base unit.

(1) Select a base unit you want to check.

(3) Check the start I/O numbers of the modules on the selected base unit.

Base Information List

| Base | Module | Base Model Name | Power Supply | Base Type | Slots | Installed Modules |
|---------|--------|-----------------|--------------|-----------|-------|-------------------|
| | | Main Base | Exist | Q | 8 | 4 |
| | | Extension Base1 | | | | |
| | | Extension Base2 | | | | |
| | | Extension Base3 | | | | |
| | | Extension Base4 | | | | |
| | | Extension Base5 | | | | |
| | | Extension Base6 | | | | |
| | | Extension Base7 | | | | |
| Overall | 1Base | | | | | 4Module |

Module Information List (Main Base)

| Status | Base-Slot | Series | Model Name | Point | Parameter | | I/O Address | Network No. Station No. | Master PLC |
|--------|-----------|--------|------------|---------|-----------|---------|-------------|-------------------------|------------|
| | | | | | Type | Point | | | |
| | - | - | Power | - | Power | - | - | - | - |
| | | Q | Q06UDEHCPU | - | CPU | - | - | - | - |
| | 0-0 | - | Empty | - | Empty | 16Point | 0000 | - | - |
| | 0-1 | Q | QX42 | 64Point | Input | 64Point | 0010 | - | - |
| | 0-2 | Q | QY42P | 64Point | Output | 64Point | 0050 | - | - |
| | 0-3 | Q | Q64AD | 16Point | Intelli. | 16Point | 0090 | - | - |
| | 0-4 | Q | Q62DAN | 16Point | Intelli. | 16Point | 00A0 | - | - |
| | 0-5 | - | Empty | - | Empty | 16Point | 00B0 | - | - |
| | 0-6 | - | Empty | - | Empty | 16Point | 00C0 | - | - |
| | 0-7 | - | Empty | - | Empty | 16Point | 00D0 | - | - |

Legend

- Error (X)
- Minor Error (A)
- Major Error (E)
- Assignment Error (I)
- Moderate Error (T)
- Assignment Incorrect (F)

5.4 How to Use the I/O Assignment Function

The I/O assignment function assigns fixed I/O numbers to the base unit slots instead of installed modules. This means that reassignment of I/O numbers is no longer required for existing modules even when the system setup changes (for example, when new modules are added).

(1) Without the I/O assignment function

System setup without new modules

| | | | | | |
|---------------------|------------|---------------------------|----------------------------|--|--|
| Power supply module | CPU module | Input module 64 points | Output module 64 points | Intelligent function module 16 points | |
| | | X00 to X3F | Y40 to Y7F | X/Y80 to X/Y8F | |

System setup with new modules (a 32 point input module and a 16 point output module added)

| | | | | | | | |
|---------------------|------------|---------------------------|---------------------------|----------------------------|----------------------------|----------------|--|
| Power supply module | CPU module | New modules | | | | | Intelligent function module 16 points |
| | | Input module 64 points | Input module 32 points | Output module 64 points | Output module 16 points | | |
| | | X00 to X3F | X40 to X5F | Y60 to Y9F | YA0 to YAF | X/YB0 to X/YBF | |

I/O number needs to be reassigned because of new modules.

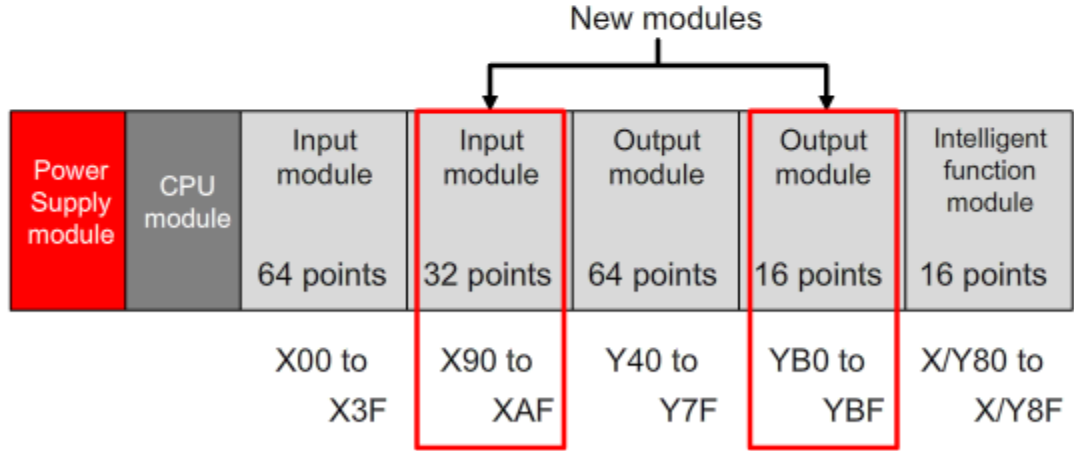
5.4 How to Use the I/O Assignment Function

(2) With the I/O assignment function

System setup without new modules

| | | | | | |
|---------------------|------------|---|--|--|--|
| Power Supply module | CPU module | Input module 64 points X00 to X3F | Output module 64 points Y40 to Y7F | Intelligent function module 16 points X/Y80 to X/Y8F | |
|---------------------|------------|---|--|--|--|

System setup with new modules (a 32 point input module and a 16 point output module added)



Because the I/O numbers of the existing modules remain unchanged, only the programs of the added modules need to be modified.

5.4

How to Use the I/O Assignment Function

The I/O assignment setting can be configured from GX Works2. Open the PLC Parameter window, then select the I/O Assignment tab.

Any I/O number is assignable for each slot regardless of the physical order of the slots.

I/O Assignment(*1)

| No. | Slot | Type | Model Name | Points | Start XY |
|-----|--------|-------------|------------|----------|----------|
| 0 | PLC | PLC | | | |
| 1 | 0(*-0) | Input | QX42 | 64Points | 0000 |
| 2 | 1(*-1) | Input | QX41 | 32Points | 0090 |
| 3 | 2(*-2) | Output | QY42 | 64Points | 0040 |
| 4 | 3(*-3) | Output | QY50 | 16Points | 00B0 |
| 5 | 4(*-4) | Intelligent | Q62DA | 16Points | 0080 |
| 6 | 5(*-5) | | | | |
| 7 | 6(*-6) | | | | |

Set the start I/O number of a module.

Assigning the I/O address is not necessary as the CPU does it automatically.

Leaving this setting blank will not cause an error to occur.

I/O numbers do not have to be continuous numbers. Some numbers can be skipped.

If the system is expected to be extended in the future, maybe some numbers should be reserved.

Module Selection

Module Type: Input module

Module Name: QX41

Mount Position

Base No.: - Mounted Slot No.: 1 Acknowledge I/O Assignment

Specify start XY address: 0090 (H) 1 Slot Occupy [32 points]

Title setting

Title: _____

Click New Module to open this window.
Here, you can select and register a module type and a module name using dropdown lists.

5.4.1 Base unit slot setting

Each slot of a base unit also has a number called the slot number, which can be assigned from within the I/O assignment settings. The slot numbers are automatically assigned (in most cases). They can also be set manually using the detail mode. This detail mode is useful to reserve some slot numbers for future extension of

Auto mode (default)

The slot numbers are automatically set according to the (main or extension) base's physical slot quantity. When an extension base unit is connected to a main base unit, the extension base's slot numbers are assigned following the last slot number of the main base unit.

(Example) When the main base unit has five slots (slot numbers 0 to 4), the slots of the connected extension base unit are numbered starting with 5.

Detail mode

Set the slot quantity for each base unit. Any number can be set. When using the detail mode, this setting is required for all the base units in use. To make a setting, open the PLC Parameter window, and select the I/O Assignment tab.

I/O Assignment(*1)

| No. | Slot | Type | Model Name | Points | Start No. | Switch Setting |
|-----|--------|------|------------|--------|-----------|----------------|
| 0 | PLC | | | | | |
| 1 | 0(0-0) | | | | | ailed Setting |
| 2 | 1(0-1) | | | | | ct PLC type |
| 3 | 2(0-2) | | | | | w Module |
| 4 | 3(0-3) | | | | | |
| 5 | 4(0-4) | | | | | |
| 6 | 5(1-0) | | | | | |
| 7 | 6(1-1) | | | | | |

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

Base Setting(*1)

| | Model Name | Cable | Slots | Base Mode |
|-----------|------------|-------|-------|-----------|
| Main | Q33B | | 5 | Auto |
| Ext.Base1 | Q65B | | 8 | Detail |
| Ext.Base2 | | | | |
| Ext.Base3 | | | | |
| Ext.Base4 | | | | |
| Ext.Base5 | | | | |
| Ext.Base6 | | | | |
| Ext.Base7 | | | | |

A (B-C)
 A: Consecutive slot number
 (number of main base unit slots + extension base unit slot number)
 B: Base unit number
 C: Slot number

Setting example

- Assign 5 slots to the main base unit (Q33B) that has 3 physical slots (to have 2 empty slots).
- Assign 8 slots to the extension base unit (Q65B) that has 5 physical slots (to allow 3 more slots).

Base unit setting mode

In this chapter, you have learned:

- Concept of I/O numbers
- I/O numbers for the extension base unit
- I/O number assignment check on the system monitor
- How to use the I/O assignment function

Point

Concept of I/O numbers and how to use the I/O assignment function

- The I/O numbers of each input/output module are sequentially assigned in 16 point units (0 to F), from the slot closest to the CPU module.
- If there is an empty slot in the middle of a base unit, I/O numbers are also assigned to the empty slot.
- The I/O numbers of modules on an extension base unit are automatically assigned, following the last I/O number of the main base unit.
- For the I/O assignment function, I/O numbers are assignable regardless of the physical order of the slots on a base unit.

Test**Final Test**

Now that you have completed all of the lessons of the **PLC Applications of Programming** course, you are ready to take the final test. If you are unclear on any of the topics covered, please take this opportunity to review those topics.

There are a total of 6 questions (29 items) in this Final Test.

You can take the final test as many times as you like.

How to score the test

After selecting the answer, make sure to click the **Answer** button. Your answer will be lost if you proceed without clicking the Answer button. (Regarded as unanswered question.)

Score results

The number of correct answers, the number of questions, the percentage of correct answers, and the pass/fail result will appear on the score page.

Correct Answers : **2**

Total Questions : **9**

Percentage : **22%**

To pass the test, you have to answer **60%** of the questions correct.

Proceed

Review

Retry

- Click the **Proceed** button to exit the test.
- Click the **Review** button to review the test. (Correct answer check)
- Click the **Retry** button to retake the test again.

The following sentences describe the retentive timer. Select the appropriate words for the respective blanks to complete the sentences.

When the is satisfied, the coil turns and the retentive timer starts measuring the time.

A retentive timer's value is retained even when the condition changes to the input condition during measurement.

When the coil turns again, the timer restarts measuring from the retained value.

When the measurement value reaches the setting value, a timeout occurs and turns on.

Even when the coil turns OFF after the timeout, the measurement value is not cleared and the contact stays ON.

The instruction is used to clear the measurement value and to turn OFF the contact.

[Answer](#)[Back](#)

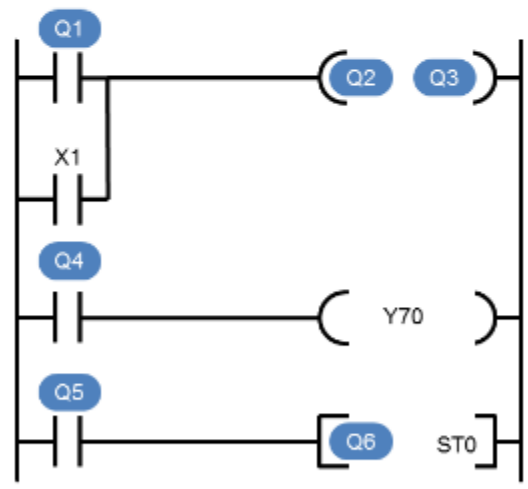
Test Final Test 2

Complete the sequence program that executes the following operation with a retentive timer:

Operation details:

- 1) The retentive timer (ST0) measures how long the input signal X0 or X1 stays ON.
- 2) When the turn-ON time period of X0 or X1 reaches 30 seconds, the coil (Y70) turns ON to turn on the timeout lamp.
- 3) When X2 turns ON, the contact of the retentive timer (ST0) turns OFF and the measurement value (present value) is reset.

Q1 Q2 Q3 Q4
 Q5 Q6



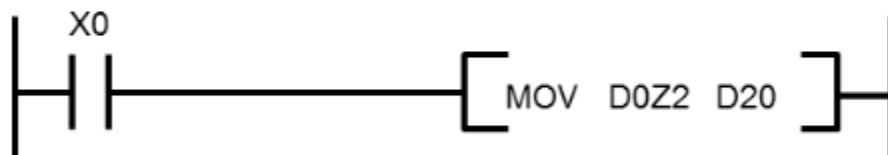
Test

Final Test 3



Below is a program using the index register "Z2". Select the value to be stored in the data register (D20) when X0 is turned ON under each condition:

- 1) When the stored value in Z2 is 0, is stored in D20.
- 2) When the stored value in Z2 is 1, is stored in D20.
- 3) When the stored value in X0 is 2, is stored in D20.
- 4) When the stored value in Z2 is 3, is stored in D20.



Stored values in
the data registers

| | |
|----|-----|
| D0 | 100 |
| D1 | 200 |
| D2 | 400 |
| D3 | 500 |

[Answer](#)[Back](#)

Test

Final Test 4



The following sentences describe the file register in QCPU. Select the appropriate words for the respective blanks to complete the sentences.

- 1) A file register is a word device used to extend data registers (D) extension and is represented by the device symbol .
- 2) Unlike the data register, the data stored in a file register are not even when the power is turned off or the CPU module is reset.
- 3) Normally, the file register is stored as a file in the memory card (RAM) or in the in the CPU module.
- 4) To use the file register, you must make the required settings in the tab of the PLC parameter window.

[Answer](#)[Back](#)

Among the numeric values used in the programmable controller, a numeric value without a decimal point is referred to as an integer and that having a decimal point is referred to as a real number.

Select the appropriate words for the respective blanks to complete the following text explaining real numbers.

- 1) One real number uses word device(s) and occupies bit memory space.
- 2) A numeric value handling a real number is referred to as . For instance, the numeric value 2.035 is stated as in the sequence program.
- 3) An instruction that handles a real number is prefixed by .
- 4) An arithmetic instruction handling a real number contain an integer and a real number at the same time for operation.

[Answer](#)[Back](#)

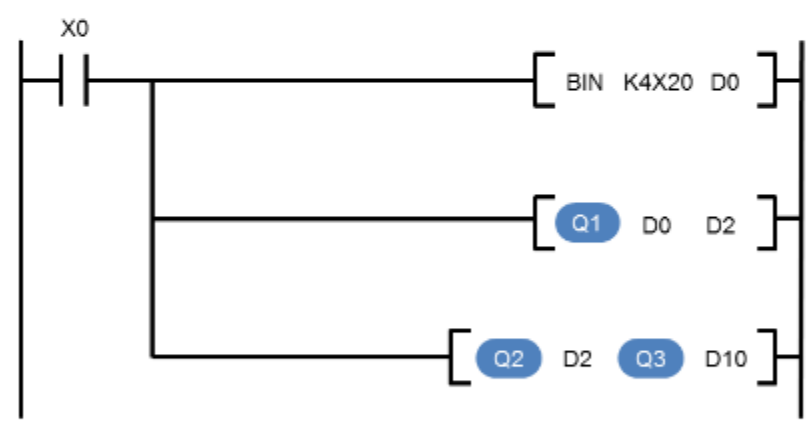
Test **Final Test 6**

Complete the following sequence program using real numbers.

Program details

- 1) When X0 is ON, operation data in X20 to X2F (BCD data) are read and stored in D0.
- 2) The value in D0 is converted to a real number and stored in D2.
- 3) The value in D2 is multiplied by 3.14 and the product is stored in D10.

Q1 Q2 Q3



Answer

Back

Test**Test Score**

You have completed the Final Test. Your results are as follows.
To end the Final Test, proceed to the next page.

Correct answers : 0

Total questions : 6

Percentage : 0%

[Proceed](#)[Review](#)[Retry](#)

You failed the test.

You have completed the **PLC Applications of Programming** Course.

Thank you for taking this course.

We hope you enjoyed the lessons and the information you acquired in this course will be useful in the future.

You can review the course as many times as you want.

Review

Close