Learning by Relay Sequences...

Your First PLC

Introduction
(Be sure to read this before the training.)

Before designing a system, be sure to read this manual and pay close attention to safety. During the training, pay attention to the following points to ensure correct handling.

[Precautions for Training]

⚠️ WARNING

- To prevent electric shock, do not touch the terminals while they are powered ON.
- Before removing safety covers, either turn the power supply OFF or confirm safety.
- Do not put your hand into moving parts.

⚠️ CAUTION

- Proceed with the training under the guidance of a teacher.
- Do not remove the training machine module or change the wiring without permission. Doing so may result in malfunction, misoperation, injury or fire.
- Before attaching or detaching the module, turn the power OFF. Attaching or detaching the module while it is still ON may cause the module to malfunction or cause an electric shock.
- If unusual older is detected with the training machine (X/Y table, etc.) immediately turn the power switch to OFF.
- If an abnormality event occurs, immediately contact your teacher.
This text book introduces basic knowledge you should be aware of for sequence control together with simple examples for first-time users of PLCs. Descriptions in this text book are primarily for the teaching material FX-I/O demonstration model that uses the micro PLC FX3G-14MR/ES type.

The following shows related materials:

- **FX3G Series main unit**
  - FX3G Series Hardware Manual JY997D33401

- **Programming**

- **Handy programming panel**
  - FX-30P Installation Manual JY997D34201
  - FX-30P Operation Manual JY997D34401

*Registration*

- Windows, Windows 7, Windows 8 are registered trademarks of Microsoft Corporation in the United States and other countries.
- All other company names and product names used in this document are trademarks or registered trademarks of their respective companies.

This document does not guarantee the implementation of industrial copyright and other rights, nor authorizes rights of implementation. Also, MITSUBISHI ELECTRIC CORPORATION cannot assume any responsibility whatsoever for problems in terms of industrial copyright that may arise by use of the content described in this document.

© 2006 MITSUBISHI ELECTRIC CORPORATION
## Contents

### Chapter 1 WHAT IS "SEQUENCE CONTROL?"
- 1.1 What Is "Sequence Control?" .............................................. 4
- 1.2 Devices Associated with Sequence Control ......................... 6
- 1.3 Items Required for Sequence Control ................................ 8
- 1.4 Performing Wiring Based on Sequences ................................. 12
- 1.5 Let's Remember Sequence Symbols ................................. 17

### Chapter 2 WHAT IS A "PLC?"
- 2.1 What is a "PLC?" .......................................................... 20
- 2.2 How PLCs Work ............................................................ 21
- 2.3 Wiring and Programs ..................................................... 27
- 2.4 Advantages of Using PLCs .............................................. 28

### Chapter 3 THE OPERATION OF GX Works2
- 3.1 Basic knowledge for operating GX Works2 ....................... 32
- 3.2 Starting GX Works2 and creating a new project .................. 36
- 3.3 Creating a program ..................................................... 39
- 3.4 Writing programs to the PLC ........................................ 45
- 3.5 Editing a program ....................................................... 50
- 3.6 Saving the program ..................................................... 60
- 3.7 Necessary operation for debugging a program .................. 62
- 3.8 Inputting comments .................................................... 69

### Chapter 4 SEQUENCE INSTRUCTIONS
- 4.1 Let's Remember PLC Instructions .................................... 74
  - LD, LDI, OUT, END
  - AND, ANI
  - OR, ORI
  - SET, RST
- 4.2 About Timer Circuits ................................................... 82
- 4.3 About Counter Circuits .............................................. 83
- 4.4 Order of Programs .................................................... 84

### Chapter 5 PROGRAM TRAINING
- 5.1 Introductory Example 1 (Control of Escalators) .................. 86
- 5.2 Introductory Example 2 (Control of Tea Dispenser) ............. 88
- 5.3 Introductory Example 3 (Control of Fan) .......................... 90
- 5.4 Introductory Example 4 (Control of Press Machine) ........... 92
- 5.5 Introductory Example 5 (Timing Chart) ............................ 94
- 5.6 Introductory example answers ..................................... 96

### Appendices
- Appendix 1 The OPERATION OF THE GX Developer ................. 97
- Appendix 2 Introduction example list programs ...................... 145
- Appendix 3 Handy Programming Panel ................................ 149
- Appendix 4 Training Machine wiring .................................. 153
Chapter 1
WHAT IS "SEQUENCE CONTROL?"

Let's learn about sequence control
Sequence control is a word that we ordinarily do not hear often. Yet, it exists all around us and we have contact with it in our day today lives.
For example, a fully automatic washing machine is an excellent example of "sequence control."
In this chapter, we will consider what "sequence control" is and the affect it has on our daily lives.
1.1 What Is "Sequence Control?"

What does "sequence" mean?

Though the word "sequence control" may generally be unfamiliar to us, it is used very often around us, and everyone is likely to have seen or have had contact with something that is controlled by sequence control.

Dictionaries describe the word "sequence" as follows:

1. State or fact of being sequent or consequent
2. Succession
3. Order of succession
4. A series of things following in order, etc.

From this, we can understand that "sequence" refers to a succession or order in which events occur.

Examples of things you are familiar with

Let's consider an example of a car wash you frequently catch sight of at a gasoline stand.

- Put your money in and press the START button.
- The car is first washed with water.
- Dirt and mud is then removed from the car with detergent.
- Brushed with a big rotating brush.
- And then washed again with water.
- Water is brushed off the car and the car wash is complete.
Though operation of the car wash on the previous page was described in approximate terms, this order is the basic way of thinking behind sequences, and making this sequence operate correctly and automatically countless times as intended becomes sequence control.

Sequence control is used extensively and is a vital approach in all situations and all fields.

Sequence control is used not just in complex applications but also in applications very familiar to us.

- **FA (factory automation) equipment in factories**
  - Control of conveyors, various processing machines, various assembling machines and others

- **Food manufacturing/processing equipment**
  - Control of various manufacturing machines for material treatment, forming, heating, cutting, packaging and others

**Sequence control is used extensively in various fields.**

- **Various equipment for professional use**
  - Large washing machine
  - Ticketing machine
  - Individually specified equipment such as cold storage and freezing

- **For control of various automation**
  - Control of parking lots
  - Control of physical distribution conveyors
  - Control of floodgates
  - Control of vinyl greenhouses
  - Control of signals and electric spectaculars
  - Control of shutter doors and others

Sequence control is used not just in complex applications but also in applications very familiar to us.
1.2 Devices Associated with Sequence Control

Component devices in sequence control

The following devices are used to perform sequence control. These devices can be broadly classified as follows:
- "devices operated by personnel"
- "devices that notify personnel of machine states"
- "devices that detect machine states"
- "devices for making machines move"

Various devices are also combined to perform sequence control in a car wash, for example.

- Devices operated by personnel
  - Switches, etc.
  - Start/stop buttons, etc.

- Devices for notifying personnel of machine states
  - Lamps, buzzers, etc.
  - In-operation lamps, etc.

- Devices for making machines move
  - Pumps for discharging detergent or water, and motors, etc. for rotating brushes

- Devices for detecting machine states
  - Switches, etc. for detecting arrival of vehicle
The devices shown in this figure are just examples and only a few of many such devices.
In sequence control, combinations of devices such as these are made to operate in accordance with a work procedure.
Of these devices, "devices operated by personnel" and "devices for detecting machine states" become the **conditions** for making things move in sequence control, while "devices that notify personnel of machine states" and "devices for making machines move" are the devices that are operated in accordance with these conditions.

**Operation panel** ----- A panel on which "devices operated by personnel" (pushbutton switches, selector switches, etc.) and "devices for notifying personnel of machine states" (lamps, digital displays, etc.) are installed.

**Control panel** -------- A panel on which devices, such as electromagnetic contactors, relays and PLCs, for controlling machine movement, are installed.

Though small-load devices such as small-size solenoid valves and pilot lamps can be driven directly by a PLC, large-load devices such as large-size solenoid valves must be driven via an electromagnetic contact or relays.
Let's actually consider sequence control.

Now, let's try actually considering sequence control based on the connection diagram below. The following also describes the terms you need to know in learning sequence control.

Example 1  The electrical circuit below is wired using a pushbutton switch and lamps (blue, red).

**Content of sequence control**

1. When the pushbutton switch is not pressed, electricity flows along route B and the blue lamp is lit.
2. When the pushbutton switch is pressed, electricity flows along route A and the red lamp lights.
3. When the pushbutton switch is released, the blue lamp lights again as in (1) above.

The operations in (1) through (3) are part of sequence control.

Connection diagram
Let's learn some new terms.

- **About contacts**
  Contacts perform a switching operation to allow or block the flow of electricity. The two basic contacts are N.O. contacts and N.C. contacts. Switches, relays, timers, counters and other devices have contacts.

**N.O. contacts**

A "N.O. contact" is a "normally open" contact, and closes when the contact is actuated.

Note 1 "Actuate" here means to force operation from normal state of rest. The operation of pressing a pushbutton corresponds to actuation.

**Operation: In the case of a pushbutton switch**

When the pushbutton switch is not pressed, the contact is open.
When it is pressed, the contact closes.

**N.C. contacts**

A "N.C. contact" is a "normally closed" contact, and opens when actuated.

**Operation: In the case of a pushbutton switch**

When the pushbutton switch is not pressed, the contact is closed.
When it is pressed, the contact opens.
Relays

Relays incorporate an electromagnet. By action of the electromagnet, pulls a moveable contact to open or close the contact*.

*Contacts: A contact refers to the contact area which executes the switching operation, and conducts or blocks electricity. Not only relays but also switches, timers, counters and other devices incorporate contacts. Contacts are mainly classified into N.O. (normally open) contacts and N.C. (normally closed) contacts (as described on the previous page).
Where are relays used?

(1) Relays can make large motors and lamps operate by using small signals.

(2) Relays can make motors and lamps in remote sites operate.

12 V DC (direct current)  
100 V AC (Alternate current)
1.4 Performing Wiring Based on Sequences

Now that you've learned some new words

Let's train using Example 1 on page 8 that we studied for relay contacts.

© Outline of relay-wired training demonstration model

Product configuration

- Power cable .......... 1 pc
- Pushbutton .......... 1 pc
- Blue lamp .......... 1 pc
- Red lamp .......... 1 pc
- Relay .......... 1 pc
- Wires (brown, red, orange, yellow, green, blue, purple, gray) .......... 1 each

Let's wire the training demonstration model.

1. Make sure that the training demonstration model is turned OFF.
2. Wire the blue lamp in Figure 1 by connecting the wires as follows:
   - Purple wire to No.7 terminal
   - Yellow wire to No.4 terminal
   - Gray wire to No.8 terminal
3. Wire the red lamp in Figure 2 by connecting the wires as follows:
   - Green wire to No.5 terminal
   - Blue wire to No.6 terminal
   - Brown wire to No.1 terminal
   - Red wire to No.2 terminal
   - Orange wire to No.3 terminal

Note) In this example, power supply terminals No.4 and No.5 are common terminals.

*1 What is an “actual wiring diagram?”

This refers to a diagram that is as close as possible to the real thing and that shows circuit connections and devices used for the circuit. As wiring and the structure of devices can be accurately seen in this diagram, it is handy when actually manufacturing devices or performing maintenance.
Now that you've learned some new terms and phrases

- Let's check operation
  
  Let swap the content of sequence control explanation on page 8 with the new terms and phrases you've learned.

1. When the power is turned ON, the N.C. contact and route B is made, and the blue lamp is turned on.
2. When the pushbutton switch (N.O. contact) is pressed, the action of the relay causes the N.O. contact to close. Route A is made, and the red lamp is turned on.
3. When the pushbutton switch is released, the blue lamp is turned on again as in (1) above.

**Sequence expressed as a flow chart**

What is a "flow chart?"

With sequence control, various devices are combined to comprise a circuit. To explain, the control of these devices can become quite difficult using based methods. Therefore a flow chart is a preferred method for explaining a control sequence.

A flow chart uses rectangular symbols and arrows to more simply express the overall order of related operations.

**Sequence expressed as a time chart**

What is a "time chart?"

A time chart expresses changes in the operation sequence over time. The devices to be controlled are drawn on the vertical axis, while changes over time are drawn on the horizontal axis. Dotted lines with arrow heads are used to show the relationship between respective devices and resulting operations.

Time charts sometimes express changes over time without the use of arrows.

You can also learn the content of sequence control from flow charts and time charts.
An actual wiring diagram is a diagrammatic means of making the structure of devices and wiring easy to understand. However, the operation sequence is harder to follow in actual wiring diagrams for complex electrical circuits. Let's try swapping an actual wiring diagram with a sequence diagram to illustrate this point.

**Wiring Diagram**

![Wiring Diagram](image)

**Sequence Diagram**

![Sequence Diagram](image)

Sequence diagrams are connection diagrams intended to make the content of operation of many circuits easier to understand. Unified standards enable this diagrammatic method to be easily understood by third parties.

Note) With the FX-I/O demonstration training machine, both relays and lamps are driven by a 24 V DC power supply. However, in general, relays are driven by a 24 V DC power supply and lamps are driven by a 100 V AC power supply. This is shown in the figure below.
Let's consider another example

**Example 2** Controlling the level of water in a tank

**Details of sequence control**

1. When the operation switch is closed, the float switch limit switch 1 closes if the tank is empty, and the magnet switch MC is operated to drive the transfer pump motor. Relay MC is designed to hold its own state even if the water level is at the mid position.
2. When the tank becomes full, float switch limit switch 2 opens, hold operation of the relay MC is canceled, and the transfer pump motor stops.
3. When the water level reaches the empty level, the motor starts to operate automatically again.

![Diagram of sequence control system](image-url)
**Sequence diagram**

---

**1** The sequence in this section is generally replaced by a sequence program on a PLC.

**2** About self-hold circuits

A "self-hold circuit" refers to an operation circuit that bypasses signals arriving from an external relay or other device by means of the contact of the relay itself.

Self-hold circuits can also be said to have a memory function that allows them to operate continuously even if the circuit is cut by releasing a pressed pushbutton.

In this circuit, the contact of the relay (MC *2) for driving the motor is connected in parallel to float switch limit 1 (LS1). This provides an example of a "self-hold switch."

**3** Though limit switch 2 (LS2) functions to cut the self-hold circuit to stop the transfer pump motor, when limit switch 2 (LS2) is used with a N.C. contact, the switch can also be turned OFF to stop transfer of water during a switch contact failure or wiring break.

---

**Operation of self-holding circuit (time chart)**

<table>
<thead>
<tr>
<th>Start switch</th>
<th>Full</th>
<th>Water level mid position</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float switch LS2</td>
<td>Water level mid position</td>
<td>Empty</td>
<td>Water level mid position</td>
</tr>
<tr>
<td>Float switch LS1</td>
<td>Operation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Even when the float switch LS1 turns OFF, operation of the magnet switch continues due to the self-holding circuit.
## 1.5 Let's Remember Sequence Symbols

### List of main sequence symbols

<table>
<thead>
<tr>
<th>Product Category</th>
<th>Contact Category</th>
<th>N.O. Contact</th>
<th>N.C. Contact</th>
<th>Drive Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Horizontal format</td>
<td>Vertical format</td>
<td>Horizontal format</td>
</tr>
<tr>
<td>Pushbutton switch (auto-return type)</td>
<td><img src="image" alt="Pushbutton switch (auto-return type)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Pushbutton switch (latched operation type)</td>
<td><img src="image" alt="Pushbutton switch (latched operation type)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Thermal relay (OCR)</td>
<td><img src="image" alt="Thermal relay (OCR)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Switch (general)</td>
<td><img src="image" alt="Switch (general)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Limit switch (mechanical operation)</td>
<td><img src="image" alt="Limit switch (mechanical operation)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Auxiliary switch relay contact</td>
<td><img src="image" alt="Auxiliary switch relay contact" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Magnetic contactor</td>
<td><img src="image" alt="Magnetic contactor" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Timer (ON delay)</td>
<td><img src="image" alt="Timer (ON delay)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
<tr>
<td>Timer (OFF delay)</td>
<td><img src="image" alt="Timer (OFF delay)" /></td>
<td><img src="image" alt="N.O. Contact Horizontal" /></td>
<td><img src="image" alt="N.O. Contact Vertical" /></td>
<td><img src="image" alt="N.C. Contact Horizontal" /></td>
</tr>
</tbody>
</table>
Introducing You to PLCs

Chapter 2
WHAT IS A "PLC?"

What is a "PLC?"
PLC stands for "programmable controller" and is sometimes known as a "sequence controller" or SC. These are defined as electronic devices that control various devices via I/O sections and have built-in memory for storing programmable instructions.

Actually...
So far, we have performed "sequence control" by physically wiring relays and timers. This chapter considers the use of a simple PLC program to control electronic devices.
2.1 What is a "PLC?"
What do PLCs do?

The devices that are operated by personnel or that detect machine states in sequence control are called "instruction signals" or "condition signals." Devices that notify personnel of machine states or that move machines are called "loads."

PLCs are responsible for carrying out "sequence control." The diagram above illustrates the control of these devices. Where the condition signals and loads in the devices are connected to the PLC.
2.2 How PLCs Work

How does sequence control work?

PLCs are microcomputer-controlled devices for industry.

Devices connected to the input side of the PLC are called "input devices," while devices connected to the output side of the PLC are called "output devices." One device is connected to each terminal block.

One input device and one output device may merely be connected to a PLC in this way, and the connection for performing sequence control is performed electronically inside the PLC.

PLC internal connections are programmed using dedicated sequence language (instructions), and a combination of these instructions is called a "sequence program." Sequence control is performed in accordance with this program, so there is no need to wire externally.
Actually consider a PLC as a group of relays and timers

The figure above shows how the input devices, output devices and the sequence program are configured. The input devices is connected to the PLC’s input relays, and the output device is controlled via contacts for external outputs.

- **Input relays**
  The input relays convert the signals from an external device to signals for the PLC. In the above figure, the input device is designed to operate merely by connecting across the input terminal and the COM terminal. However it should be noted that PLCs have a limited number of contacts, even though, an infinite number of contacts is provided on the sequence program.

  Operation

  ![](image)

  The PLC has a built-in power supply for the input relays. When the contact of the input device is conducting, current flows along the dotted line to drive the input relay.

  Regardless of whether the external input device is a N.O. contact or a N.C. contact, the contact on the PLC that closes when the COM and input terminals are conducting is an N.O. contact, and the contact that opens is a N.C. contact.

- **Output contacts**
  Output contacts are otherwise known as just outputs. These are controlled by the sequence program, and are necessary for driving an external load. Devices having a different power supply (AC or DC) can be connected to the contacts for external output.

  Operation

  ![](image)

  When the output relays on the sequence program are driven, the contacts for external output close. Provide a power supply for driving the external device outside the PLC.
An I/O number is assigned to each terminal to make the I/O devices connected to the input terminals and output terminals correspond to the input relays and output relays of the sequence program.

Timers and counters held internally by the PLC are also assigned to devices numbers in addition to each terminal number.

☆ **Element numbers** ---- Element numbers comprise a symbol that shows what the devices is and a number that categorizes individual devices.

**Input relays** : X000~
Input relays act as the "point of contact" for receiving signals, for example, from an input switch outside of the PLC. X is used to identify their device number. PLCs have a built-in number of input relays corresponding to the number of inputs (number of terminals).

**Output relays** : Y000~
Output relays act as the "point of contact" for driving loads outside the PLC. Y is used to identify their device numbers. PLCs have a built-in number of output relays corresponding to the number of outputs (number of terminals).

**Auxiliary relays** : M0~
Auxiliary relays are held internally by the PLC and are also called "internal relays."

☆ The number of inputs relays, output relays, timers, counters, and other devices varies according to the model of PLC.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Decimal, Octal and Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>As shown in the following table, octal and hexadecimal numbers are assigned as devices numbers in addition to decimal numbers.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input relays, output relays</th>
<th>Auxiliary timers, timers, counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro PLC FX Series</td>
<td>Octal</td>
</tr>
<tr>
<td>General-purpose Q/A Series</td>
<td>Hexadecimal</td>
</tr>
</tbody>
</table>

"**Decimal**" The decimal number system is used most often, and counts up numbers in units of ten in the format 0 to 9, 10 to 19, 20 to 29 and so forth.

"**Octal**" The octal number system counts up numbers in units of eight in the format 0 to 7, 10 to 17, 20 to 27 and so forth.

"**Hexadecimal**" The hexadecimal number system counts up numbers in units of 16 in the format 0 to 9, 0A, 0B, 0C, 0D, 0E, 0F, 10 to 19, 1A, 1B, 1C, 1D, 1E, 1F and so forth.
Expressions used in sequence diagrams and sequence programs

The sequence program on a PLC replaces the input device and output device connected to each of the external terminals with a circuit for performing sequence control. For details on instruction words, see Chapter 4.

The following explains how to replace a sequence diagram with a sequence program. This example shows the tank water level control described on page 15 replaced with a sequence program.

The following describes the various differences when a relay sequence program is replaced with a sequence program.

(1) How N.O. contacts and N.C. contacts are expressed

Relay sequence diagram

<table>
<thead>
<tr>
<th>N.O. contact</th>
<th>N.C. contact</th>
</tr>
</thead>
</table>

Sequence program

<table>
<thead>
<tr>
<th>N.O. contact</th>
<th>N.C. contact</th>
</tr>
</thead>
</table>

(2) The power supply circuit is not expressed.

(3) The sequence program is assigned devices numbers explained in the previous page.

Important

(4) Limit switch 2 in the sequence diagram is a N.C. contact but becomes a N.O. contact in a sequence program.

This is because the role of limit switch 2 in the above sequence diagram is to break the self-hold circuit of the MC and stop the MC. So, we should consider that it is in a conducting state at all times on the circuit.

Next, remember the operation of the input relay on page 22. In operation of the N.O. contacts and N.C. contacts in the sequence program, **N.O. contacts turn ON** and N.C. contacts turn OFF when the input relay is electrically conducting due to a change in state of the contact on the external wiring. Due to this fact, a N.O. contact must be used on the sequence program to obtain the same operating state.

In this way, when a PLC is used, N.O. contacts and N.C. contacts can be used for each of the contacts on the sequence program.
The following describes the operation of N.O contact and N.C. contact in a PLC program.

• When no current is flowing in the input devices, a N.O. contact appears as OFF in PLC program. A N.C. contact appears as ON.
• When current is flowing in the input devices, a N.O. contact appears as ON in PLC program. A N.C. contact appears as OFF.

A contact in the PLC program does not replace the physical switch. It represents the input coming into the PLC. Therefore, we use N.O. contact in the PLC program for the signal of LS2.

For details, please refer to (1) and (2) below.

[1] Reason for using limit switch 2 (X2) as an N.O. contact in the PLC program
   (1) When limit switch 2 is used as N.C. contact in the same way as the sequence diagram
   (2) When the limit switch 2 is used as an N.O. contact

Program concept
N.O. contacts become conductive and N.C. contacts become nonconductive when a signal is input from the outside.

- The pump is OFF while the water is in the mid level.

- The pump is OFF.

- The limit switch 1 turns ON, and the pump turns ON.

- The limit switch 1 turns OFF, but the pump remains ON by the self-holding function.

- While the pump is ON and water is supplied

- The limit switch 1 turns OFF, but the pump remains ON by the self-holding function.

- When the water reaches the full level

- The limit switch 2 turns ON, and the pump turns OFF.

- The pump is OFF.
2.3 Wiring and Programs

2.3.1 What are PLC wiring and programs like?

The figure below shows the circuit on page 16.

The wiring of a PLC wiring can be divided up into I/O wiring and internal wiring.

I/O wiring must be handled in the conventional way using cutting pliers and a screwdriver.

Complicated internal wiring in a PLC (sequence program), can be handled easily by operating the pushbuttons (keys) on the programming panel.

The connections between input terminals and input relay coils and between output contacts of output relays and output terminals were already established at the delivery from the factory.
2.4 Advantages of Using PLCs
2.4.1 Effective use of PLCs

1. Economical
   In general, PLC is more inexpensive than conventional control box where 10 or more relays are required.

2. Less manpower for design
   Large amount of labor-saving can be achieved through simplified parts layout drawings, sequence design and pre-start-up and test procedures.

3. Short turn-around time
   The turn-around time can be significantly reduced through reduction in quantity of procured parts, parallel procurement of the machinery and control box, flexibility in specification change, simplified wiring work and so forth.

4. Downsizing and standardization
   Significantly reduced in size than relay boards and mass-production is possible by reusing the program.

5. Improved reliability
   Troubles due to relays and timers can be reduced and the component with PLC can be used without any concern once the initial set-up of the PLC is completed.

6. Improved maintainability
   Maintenance can be done in an easy manner, since few parts with life limitation are used and the PLC has a self diagnosis function.
### 2.4.2 Comparison with relay control

<table>
<thead>
<tr>
<th>Item</th>
<th>Method</th>
<th>Relay control</th>
<th>PLC control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>Complex control is possible by using many relays.</td>
<td>Control can be programmed however complex it is.</td>
<td></td>
</tr>
<tr>
<td>Changeability of control content</td>
<td>No other option but to change the wiring</td>
<td>Only the program needs to be changed, and control can be freely changed.</td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>Though there is no problem in regular use, there are restrictions in faulty contacts and service life of parts.</td>
<td>Reliability is higher as the center of operations is made up entirely of semiconductors.</td>
<td></td>
</tr>
<tr>
<td>Versatility</td>
<td>Once made, devices cannot be used for other applications.</td>
<td>PLCs can be used for any kind of control depending on the program.</td>
<td></td>
</tr>
<tr>
<td>Device expandability</td>
<td>If additions and modifications are required they are difficult.</td>
<td>PLCs can be freely expanded up to their full potential.</td>
<td></td>
</tr>
<tr>
<td>Ease of maintenance</td>
<td>Periodic inspection and limited service life parts must be replaced.</td>
<td>PLCs can be repaired merely by replacing modules.</td>
<td></td>
</tr>
<tr>
<td>Range of functions</td>
<td>Relay control only is supported.</td>
<td>Analog and positioning control also can be performed in addition to sequence programs.</td>
<td></td>
</tr>
<tr>
<td>Device size</td>
<td>Generally, large</td>
<td>PLCs do not increase in size even in complex, advanced control applications.</td>
<td></td>
</tr>
<tr>
<td>Design, production period</td>
<td>Many drawings are required, and it takes time to arrange parts and test assemblies.</td>
<td>Design is simple even for complex control, and it does not take time to manufacture PLCs.</td>
<td></td>
</tr>
</tbody>
</table>

### Reference

**A Short History of PLCs**

The device "PLC" came into being in 1968 as a result of commissioned development by General Motors in the United States, and the general market for PLCs started in the United States in the following year. The first domestically produced PLC made its appearance in Japan in 1970, yet it took six years for the first general-purpose PLC to appear in 1976. MITSUBISHI ELECTRIC CORPORATION initiated the market for general-purpose PLCs in the year 1977, and has since made one-board PLC modules a familiar name on the market. Since then, it has developed the general-purpose PLC K series with integrated numeric value processing functions, which was followed by the micro PLC F series with built-in programmer. Today a variety of PLC products are available world wide that can suite an array of control applications.
Let's use a personal computer to create programs!

GX Works2 software provides an efficient and easy way to create and edit sequence programs for PLCs. Once the basic operations are mastered, programming often involves straightforward repetition. Beginning with the most necessary operations, let's learn how to program from the beginning.

Smoothly begin new projects and update them with ease...

It is easy to debug programs with GX Works2 and update them as necessary. The operation status of the PLC and program can be monitored with a personal computer, so if some parts are not working as planned, changes and updates can quickly made.

Make programs easier to read

There is a "comment input function" in GX Works2 to make sequence programs easier to read. Comments can improve the efficiency of creating and debugging ladder programs.
3.1 Basic knowledge for operating GX Works2
3.1.1 Layout of the GX Works2 screen

1) Title bar
The name of the opened project and the window operation icons are displayed.

- Changing the size of GX Works2 and closing it
- Maximizing and restoring GX Works2
- Display of the project name and its path
- Minimizing GX Works2
- Closing GX Works2
2) Menu bar

Drop down menu items are displayed when a menu is selected.

3) Toolbar*

*: The contents of the toolbar can be moved, added, and removed. Therefore, the layout and items displayed depend on saved environment.

Frequently used functions are displayed with icon buttons for quick execution.

4) Status bar

The status of the operation and keyboard settings are displayed.
5) Navigation window

The sequence program is displayed when [POU] → [Program] → [MAIN] is clicked.

The tree structure indicating the contents of “Project” is displayed in the above area when this button is clicked.

The personal computer connection destination (PLC connection) is displayed in the above area when this button is clicked.

6) Edit screen

Switch tab
3.1.2 Workspace and project

- **Workspace**
  GX Works2 can control multiple projects under one name. Do not change the workspace configuration using Windows® Explorer or other software.

- **Project**
  The project consists of “Program”, “Device comment”, “Parameter” and “Device memory”. Altogether, this data is called “Project”, and saved in a folder having the workspace name.

![Diagram of Workspace and Project](image)

When saved in workspace format
3.2 Starting GX Works2 and creating a new project

3.2.1 Starting GX Works2

1) Start from the Start button of Windows®, and select the application as follows:

   [Programs]
   [MELSOFT Application]
   [GX Works2]

2) GX Works2 is started.
3.2.2 Creating a new project

1) Select \[\text{Project} \rightarrow \text{New} (\text{Ctrl} + \text{N})\] from the menu.

2) Click the \[\text{▼}\] button of \[\text{Series}\].

3) Select "FXCPU".

(Note) Select the series name that is actually used.

4) Click the \[\text{▼}\] button of \[\text{Type}\].

5) Select "FX3G/3GC".

(Note) Select the series name that is actually used.
6) Click **OK**.

(Note) Choose **Simple Project** for the "Project Type". Do not check "Use Label". Select "Ladder" in "Language".

7) A new project screen is displayed for project data to be input.
3.3 Creating a program
3.3.1 Creating a program by using the function keys

[Program to be created]

POINT
In this book, the input and output relay numbers are displayed with three digits, such as "X000," and "Y000." When using GX Works2, however, "X0," "Y0," etc. may be input.

POINT
The keyboard shortcuts for ladder elements are displayed on the buttons of the toolbar.

Main key operations

• "NO contact (+→)" and "Coil (←), (←)" can be input directly without using the keyboard shortcuts.
1) Press the $F5$ (←-→) key.
Input "X2".

2) Confirm by pressing the Enter key or [OK].

3) The ladder input (←↑-) is displayed.
4) Press the $F6$ (←-→) key.
Input "X0".

5) Confirm by pressing the Enter key or [OK].

6) The ladder input (←X0) is displayed.
7) Press the $F7$ (←←-) key.
Input "Y0".

8) Confirm by pressing the Enter key or [OK].

9) The ladder input (←Y0←) is displayed.
10) Press the $\text{Shift} + F5$ (←-→) key.
Input "Y0".

11) Confirm by pressing the Enter key or [OK].
12) The ladder input \(Y_0\) is displayed.

13) Move the cursor to the beginning of the next line.

14) Press the \(F_5\) key.
   Input "X3".

15) Confirm by pressing the Enter key or [OK].

16) The ladder input \(X_3\) is displayed.

17) Press the \(F_7\) key.
   Input "Y1".

18) Confirm by pressing the Enter key or [OK].

19) The ladder input \(X_3 \to Y_1\) is displayed.

20) Compile Operation [Important].
    Perform the "Compile" operation to finalize ladder that not been compiled (the gray part).

    Press the \(F_4\) (Build) key.
    Or select [Compile] \(\to\) [Build] from the menu.
3.3.2 Creating a program by using the toolbar buttons

[Program to be created]

Click the toolbar buttons to input ladder elements.

POINT

In this book, the input and output relay numbers are displayed with three digits, such as "X000," and "Y000." When using GX Works2, however, "X0," "Y0," etc. may be input.
1) Click the toolbar button. Input "X2".

2) Confirm by pressing the Enter key or [OK].

3) The ladder input (X2) is displayed.

4) Press the toolbar button. Input "X0".

5) Confirm by pressing the Enter key or [OK].

6) The ladder element is displayed.

7) Click the toolbar button. Input "Y0".

8) Confirm by pressing the Enter key or [OK].

9) The ladder element is displayed.

10) Click the toolbar button. Input "Y0".

11) Confirm by pressing the Enter key or [OK].
12) The ladder element is displayed.

13) Move the cursor to the beginning of the next line.

14) Click the toolbar button $\frac{\text{as}}{\text{as}}$. Input "X3".

15) Confirm by pressing the Enter key or [OK].

16) The ladder element $\text{ş}$ is displayed.

17) Click the toolbar button $\frac{\text{ç}}{\text{ç}}$. Input "Y1".

18) Confirm by pressing the Enter key or [OK].

19) The ladder element $(-Y1-)$ is displayed.

20) Compile Operation [Important].

Perform the "Compile" operation to finalize the ladder that has not been compiled (gray part).

Press the $\text{F4}$ (Build) key.

Or select $\text{Build}$ from the toolbar, or select [Compile] $\rightarrow$ [Build] from the menu.

The gray shade disappears and the program is compiled.
If an error occurs, the cursor moves to the problematic part of the program. Correct the program.
3.4 Writing programs to the PLC

Write the created sequence program to the FX PLC.

3.4.1 Connecting PC to PLC

Connecting the FX3G PLC (with a USB cable)

Preparation on the personal computer side]

For connecting the FX3G PLC to the personal computer with a USB cable, it is necessary to install the USB driver software in the personal computer.

Install the USB driver software in accordance with the procedure described in the “GX Works2 Operating Manual (Common)”. 

Point

USB Driver installation

<When using Windows® 7, Windows® 8>

1. Connect the personal computer and the programmable controller CPU with a USB cable, and then turn on the programmable controller CPU.

2. From Windows® Control Panel, select [System and Security] - [Device Manager]. Right-click “Unknown device” and click “Update Driver Software”.

3. The Update Driver Software screen is displayed. Select "Browse my computer for driver software" and specify "Easysocket\USBdrivers" in the folder where GX Works2 has been installed on the displayed screen. If multiple MELSOFT products are installed previously, refer to their installed location.

* The USB driver installation is different when using FX3U-USB-BD and FX-USB-AW. Installation method also differs depending on the personal computer, please refer to “GX Works2 operating manual (Common)” for details.
3.4.2 "Transfer Setup" in GX Works2 and "Writing programs"

Configure the settings of GX Works2 to communicate with the PLC.

1) Click "Connection Destination".
2) Double-click "Connection1".
3) Transfer Setup is displayed.
4) Double-click the icon.
5) Set the communication port of the personal computer side.
   - Choose "USB" for connecting to the built-in USB port of the FX3G PLC.
6) Click [OK] after the setting is completed.
7) Click [Connection test], to check communication with the PLC.
8) After checking, click [OK] to confirm the configured setting.
9) Set the "RUN/STOP" switch of the PLC to "STOP". [Supplementary note: Automatic RUN/STOP function from the programming software]

- When attempting to write program while the PLC is in the RUN state, the message “The CPU status must be STOP to perform a write operation. Would you like to perform write operation?” appears after step (4).
- Click [Yes (Y)] to execute writing.
- When writing is finished, the message “The CPU status is currently STOP. Would you like to perform a remote-RUN?” appears.
- Click [Yes (Y)] to set the PLC to RUN state.

10) Select [ ] from the toolbar, or select [Online] → [Write to PLC] from the menu.

11) Click [Param + Prog].
12) Click [Execute].
(Refer to “Supplementary note” in 9 above also).

A dialog box showing the write progress is displayed.

13) Click [Close] after it is completed.
### 3.4.3 Monitoring PLC operation

1) Set the "RUN/STOP" switch of the PLC to "RUN".

2) Perform either of the following operations:
   - Click the F3 (Monitor Mode) key.
   - Click on the toolbar.
   - Select [Online] → [Monitor] → [Start Monitoring (All Windows)] from the menu.

---

#### Check operation by monitor mode

1) Set [Switch X002 is "ON"] and [Switch X000 is "OFF"], and then check [Output Y000 is "ON"].

2) Check [Output Y000 is "ON"] while [Switch X002 is "OFF"].

3) Set [Switch X000 is "ON"] and then check [Output Y000 is "OFF"].

4) Check [Output Y001 is "ON/OFF"] in accordance with [Switch X003 is "ON/OFF"].
### Reference

1) **The display of the monitor status dialog**

- **Scan time**
  - The maximum scan time of the sequence program is displayed.

- **PLC status**
  - The status of the PLC is displayed. "▲" is displayed when PLC is in RUN, and "■" is displayed when PLC is in STOP.

- **Monitor execution status.**
  - This icon is flashing during monitor mode.

- **Memory type**
  - The memory type of the PLC is displayed.

2) **Interpretation of the ladder monitor display**

#### 1. Contact Instruction

<table>
<thead>
<tr>
<th>Type</th>
<th>X0 : OFF</th>
<th>X0 : ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO contact</td>
<td><img src="image" alt="Contact open" /></td>
<td><img src="image" alt="Contact close" /></td>
</tr>
<tr>
<td>NC contact</td>
<td><img src="image" alt="Contact close" /></td>
<td><img src="image" alt="Contact open" /></td>
</tr>
</tbody>
</table>

#### 2. Out Instruction

<table>
<thead>
<tr>
<th>Type</th>
<th>Non-execution/Non-drive</th>
<th>Execution/Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT instruction</td>
<td><img src="image" alt="Y000" /></td>
<td><img src="image" alt="Y000" /></td>
</tr>
<tr>
<td>SET instruction, etc.</td>
<td><img src="image" alt="SET M0" /></td>
<td><img src="image" alt="SET M0" /></td>
</tr>
</tbody>
</table>

The ON/OFF status of the device to be reset is displayed during monitor mode using the RST instruction.

<table>
<thead>
<tr>
<th>Type</th>
<th>Device status</th>
<th>When device to be reset is OFF</th>
<th>When device to be reset is ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST instruction</td>
<td><img src="image" alt="RST M0" /></td>
<td><img src="image" alt="RST M0" /></td>
<td><img src="image" alt="RST M0" /></td>
</tr>
</tbody>
</table>
3.5 Editing a program
3.5.1 Correcting a program

Point

Switch between "Ovrwrte" and "Insert"
- Set to "Ovrwrte" when correcting and overwriting ladder.
- New ladder will be inserted when the "Insert" mode is on.

Switches between "Ovrwrte" and "Insert" by pressing the Insert key.
[1] Changing the OUT coils and contacts

[Program to be corrected]  

1) Double-click the part that needs to be corrected.

2) Change “Y001” to “Y002”.

3) Confirm by pressing the Enter key or [OK].

4) The corrected result is displayed and the ladder block is displayed in gray.

5) Confirm the changes by pressing the F4 (Build) key.
[2] Adding lines

[Program to be corrected]

1) Click  on the toolbar.

2) Locate the cursor on the upper right of the desired vertical line to be added, and then drag it until it reaches the desired position, and then drop it.

3) The line reaching the dropped position is added.

4) Locate the cursor on the position where the OUT coil is to be added and click  on the toolbar.

5) Input "Y3".

6) Confirm by pressing the Enter key or [OK].
7) The added ladder is finished and the ladder block is displayed in gray.

8) Confirm the changes by pressing the F4 (Build) key.

- Click on the toolbar again to finish the operation.

**Reference**

Adding lines using keys

In GX Works2, lines can be added using the Ctrl + , , , keys.
[3] Deleting lines

[Program where lines are to be deleted]

1) Click (Alt+F9) on the toolbar.

Locate the cursor on the upper right of vertical line to be deleted.

2) Locate the cursor on the upper right of the desired vertical line to be deleted, and then drag it until it reaches the desired position, and then drop it.

3) The lines are deleted.

4) Delete the OUT coil by pressing the Delete key.
5) The deleted ladder block is displayed in gray.

6) Confirm the changes by pressing the F4 (Build) key.

- Click on the toolbar again to finish the operation.

---

**Reference**

**Deleting lines using keys**

In the GX Works2, lines can be deleted using the Ctrl + →, ←, ↑, ↓ keys.
3.5.2 Inserting and deleting rows

[1] Adding rows

**[Program where a rung is to be inserted]**

A row is inserted above the row where the cursor is located.

1) Locate the cursor on the row below the one to be inserted.

2) Right click the mouse at any place, and select [Edit] → [Insert Row].

3) A row is inserted.

4) Add ladder in the inserted row.

5) Confirm the changes by pressing the **Build** (F4) key.
[2] Deleting rows

1) Move to the row to be deleted.

2) Right click the mouse at any place, and select [Edit] → [Delete Row].

3) The row is deleted.

Point

Confirm it by pressing Build (F4).
3.5.3 Cutting and copying (pasting) ladder

[1] Cut

[Program to be edited]

1) Move the cursor to the beginning of the ladder to be cut.

2) Drag the cursor until it reaches the desired position, and then drop it.

3) Execute cut

4) The selected area is cut.

A gray part remains when a smaller portion of the ladder is cut. After amending the ladder, confirm the changes by pressing the F4 (Build) key.
[2] Copy (pasting)

[Ladder to be copied (pasting)]

1) Move the cursor to the beginning of the copy area.

2) Drag the cursor until it reaches the desired position, and then drop it.

3) Select 📐 from the toolbar or select [Edit] → [Copy] (Ctrl+C) from the menu.

4) Move the cursor to the position where to paste.

- Using the Insert key
  - "Ovwrte" mode: Pastes by overwriting data from the cursor position.
  - "Insert" mode: Pastes it by inserting data at the cursor position.

5) Select 📐 from the toolbar or select [Edit] → [Paste] (Ctrl+V) from the menu.

A gray part remains when a smaller portion of the ladder is pasted. After amending the ladder, confirm the changes by pressing the F4 (Build) key.
3.6 Saving the program
3.6.1 Save and Save as

Point
If there is ladder that has not been compiled, press **Build** (F4).

1) Select ![icon] from the toolbar or select [Project] → [Save] (Ctrl+S) or [Save As] from the menu.

(When it is newly saved)

1) Click

(When it is saved by overwriting)

**Project saving is finished.**

2) Specify the storage destination for the file.
3) Specify the file name.
4) Specify the title describing the project (optional).
5) Click ![icon]

6) Click ![icon] in the confirmation dialog to finish.
3.6.2 Reading a project

Reference

If another project is open when reading/opening a new file, the current project is closed. If there is uncompiled ladder in the project or the project is not saved, a warning message is displayed.

1) Select \(\text{Open} \) from the toolbar or select [Project] → [Open] (Ctrl+O) from the menu.

2) Select the storage destination of the file.

3) Select the project to be read.

4) Click [Open] to read the project.
3.7 Necessary operation for debugging a program

For connecting to the PLC and writing a program to the PLC, see "3.4 Writing programs to the PLC".

3.7.1 Ladder monitor

Display the ladder, and monitor the conduction status of the contacts and the driving status of the coils.
(See section "3.4 Writing programs to the PLC") .

1) Select from the toolbar or select [Online] → [Monitor] → [Start Monitoring(All Windows)] from the menu.

2) The ON/OFF status of the ladder and the current value of word devices (timer, counter and data register) are displayed in the ladder monitor window.

Reference

In GX Works2, stop monitoring by selecting the icon (Stop Monitoring) on the toolbar or [Online] → [Monitor] → [Stop Monitoring] from the menu. Ladder can be edited when monitoring is stopped.
3.7.2 Device registration monitor

Specify one device or a range on the ladder monitor window, and register corresponding device(s) on the watch window.

1) Set the ladder monitor status. (See “Ladder monitor” above).

2) • When registering one device, select it by clicking it.
   • When registering multiple devices, click and drag to select the corresponding range.

3) Display a watch window with [View]→[Docking Window]→[Watch 1].

4) When the mouse icon changes to , drag and drop to the watch window.

5) The selected device(s) is (are) registered in the watch window.

6) It is possible to directly input device names such as “X0”, “M0” and “D0” to blank spaces of the “Device/Label” column instead of dragging & dropping device(s).

7) Select [Online] → [Monitor] → [Start Monitoring] from the menu.

63
3.7.3 Device batch monitor

Specify a device and monitor a continuous range of devices that follow it.

1) Set to monitor mode. (See Section “3.7.1. Ladder monitor”).

2) Select [Online] → [Monitor] → [Device/Buffer Memory Batch] from the menu. Or right click the ladder window and select [Device/Buffer Memory Batch].

3) Input the first device of the devices to be monitored in the “Device/Buffer Memory Batch” window and then pressing the Enter key to begin monitoring.

(Note) When specifying a timer or counter, click [Reference], and specify the program “MAIN”.

4) The operation state is displayed in accordance with the device type.
   - Bit devices (X, Y, M and S): 1 (= ON) or 0 (= OFF)
   - Timers and counters: ON/OFF status, set value and current value of contact/coil
   - Data registers: Current value

5) A bit device can be forcibly turned ON or OFF or the current value of a word device can be changed by selecting the corresponding displayed monitor data and clicking [Modify Value].
3.7.4 Device test

[1] Force ON/OFF

Using the device test screen, forcibly turn ON/OFF the bit devices of the PLC (M, Y, T, C and so on). (The forced ON/OFF function for X is not available).

When the PLC is running, the forced ON/OFF function can turn on or off specific devices.

1) Set to monitor mode. (See Section “3.7.1. Ladder monitor”).

2) Select [Debug] → [Modify Value] from the menu.

3) Input the device number to be forcibly turned on/off.

4) • [ON]: Forces device to ON.
   • [OFF]: Forces device to OFF.
   • [Switch ON/OFF]: Forces the device to toggle ON/OFF each time it is pressed.

Reference

Forced ON/OFF (Ladder monitor window)

Any bit device (contact and coil) can be forcibly turned on/off by double-clicking the device in the [Ladder monitor window] while pressing the [Shift] key.
[2] Changing the current value of a word device

Change the current value of the PLC’s word device (T, C, D and so on).

1) Set to monitor mode. (See Section “3.7.1. Ladder monitor”).

2) Select [Debug] → [Modify Value] from the menu.

3) Input the device number to be changed.

4) Input a new value.

5) Click [Set].
[3] Force ON/OFF and current value change using the watch window

It is possible to forcibly turn ON or OFF a bit device (such as M and Y) or change the current value of a word device (such as T, C and D) from the watch window.

When an input (X) is forcibly turned ON or OFF, it remains ON or OFF only in 1 scan cycle while the PLC is in the RUN state.

When checking the operation of outputs (Y), set the PLC to the STOP state.

1) Enter the device number to the “Device/Label” column, and register it to the watch window.

(Note) After registering a device, select [Online] → [Watch] → [Start Watching] from the menu.

2) Enter a new value.

- Bit devices (such as X and Y): Enter “1” for “ON”, or “0” for “OFF”.

- Word devices (such as T, C and D): Enter a numeric value within the available range of the corresponding device.
3.7.5 Writing a program to the PLC during RUN

Write the corrected part of the ladder to the PLC when the PLC is running. Less time is needed for writing during RUN since the entire program is not transferred.

1) A contact will be added to the ladder on the left as an example. Within the ladder view, set the mode to write mode.

2) Add a contact. The ladder block is displayed in gray.

3) Press [Shift] and [F4] together, or select [Compile] → [Online Program Change] from the menu.

4) Click [Yes] to confirm the warning message about PLC safety regarding online changes.

5) The message "Online change has completed." is displayed. Click [OK].

Caution

It is impossible to write the program to the PLC if the program in the PLC is different from the one in GX Works2. Verify in advance, or transfer the program first by using [Write to PLC].
3.8 Inputting comments

3.8.1 Types of comments

The following 3 types of comments can be input.

<table>
<thead>
<tr>
<th>Type</th>
<th>Purpose</th>
<th>The number of characters (full-width)</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Device comment</td>
<td>A comment describing the role and function of each device</td>
<td>16</td>
<td>It is necessary to set the &quot;Comments capacity&quot; in the parameter setting when writing to the PLC. The &quot;Comment range setting&quot; must also be set.</td>
</tr>
<tr>
<td>2) Statement</td>
<td>A comment describing the role and function of ladder blocks</td>
<td>32</td>
<td>This is a comment on GX Works2 side. (It is not downloaded to the PLC).</td>
</tr>
<tr>
<td>3) Note</td>
<td>A comment describing the role and function of output instructions</td>
<td>16</td>
<td>This is a comment on GX Works2 side. (It is not downloaded to the PLC).</td>
</tr>
</tbody>
</table>

[Comment Examples]

Global device comment and local device comment in the GX Works2

- Global device comment: This type of comment can be written to PLCs.
- Local device comment: By setting local device comments, it is possible to create multiple comments different from global device comments.

Local device comments can be written to Q Series PLCs, but cannot be written to the FX Series PLCs. In general, use “Global device comment”.

Point

How to display comments

Select [View] → [Comment] from the menu and then the comments are displayed. Repeat the operation above to stop displaying comments.
3.8.2 Operation for creating device comments

[1] How to input device comments using a list

1) Double click [Global Device comment] in the project list.

2) Input the start device of the devices to be commented in "Device Name", and press the Enter key.

3) Input comments in the "Comment" column.

• When inputting comments for another device, input the device number again in step 2.

[2] How to input device comments in the ladder

1) Click from the toolbar and double-click the ladder element to be commented.

2) Input the comment in the "Enter symbol" window and click [OK].

• Click on the toolbar again to finish the operation.
Setting for writing the device comments to the PLC

It is necessary to set "Parameter setting" and "Comment range setting" in order to write the device comments to the PLC.

1) Parameter setting

1) Double click [Parameter] → [PLC parameter] in the project list.

2) Comment capacity setting

2) Set the "number of blocks" in the "Comments capacity" setting. One block is equivalent to 50 comments, occupying 500-steps of program memory.

2) Comment range setting

1) Choose [Online] → [Write to PLC] from the menu.

2) Check "global device comment".

3) Click [details].

4) Set the device type and range in the Device Comment Detail Setting dialog box.

Reference

How to input comments when creating a ladder

While inputting a ladder element, click the icon shown on the left to switch the input method.
3.8.3 Operation for creating statements

1) Click \( \text{ } \) from the toolbar, and double-click anywhere on the ladder block where the statement is to be written.

2) Input the statement in the "Enter line statement" window and click [OK].

• Click \( \text{ } \) on the toolbar again to finish the operation.

3.8.4 Operation for creating notes

1) Click \( \text{ } \) from the toolbar, and double-click the output instruction symbol where the note is to be written.

2) Input the note in the "Enter Note" window and click [OK].

• Click \( \text{ } \) on the toolbar again to finish the operation.
What you have learned up till now

So far we have explained that PLCs are a collection of many relays, timers and counters, and that their internal sequences can be wired by key operation on a programming panel. When performing this wiring work, rules matched to contact and coil connection methods and types of coils are required. These rules are the instructions. Instructions are programmed in the format "instruction word + devices number" or by individual instruction words.

This chapter introduces several of the basic instructions for programming a PLC.
4.1 Let's Remember PLC Instructions

4.1.1 Commands and Programs

There are two types of peripheral devices (programming devices) for entering Sequence programs: devices for drawing sequence program on screen and devices for entering programs using instruction words (that is, devices for creating a list program). These devices differ only by program input method, and the program itself is the same.

The following table summarizes the instructions held by a PLC, the meanings of each instruction and how the ladder is indicated in a sequence program.

<table>
<thead>
<tr>
<th>Notation, Name</th>
<th>Function</th>
<th>Ladder representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD Load</td>
<td>Bus connection instruction, N.O. contact</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>LDI Load inverse</td>
<td>Bus connection instruction, N.C. contact</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>AND</td>
<td>Series connection, N.O. contact</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>ANI AND inverse</td>
<td>Series connection, N.C. contact</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>OR ORI</td>
<td>Parallel connection, N.O. contact</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>OR inverse</td>
<td>Parallel connection, N.C. contact</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>ANB AND block</td>
<td>Series connection between blocks</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>ORB OR block</td>
<td>Parallel connection between blocks</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>OUT</td>
<td>Coil drive instruction</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>SET Set</td>
<td>Latch operation, coil instruction</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>RST Reset</td>
<td>Cancel latch operation, coil instruction</td>
<td>![Ladder Diagram]</td>
</tr>
<tr>
<td>NOP No-operation</td>
<td>No operation</td>
<td>For deleting program or space</td>
</tr>
<tr>
<td>END End</td>
<td>End of program</td>
<td>Return to 0 step</td>
</tr>
</tbody>
</table>

- - and - used in ladder indications indicate contacts. Contacts are in two states, conducting and non-conducting, depending on the ON/OFF state of input relays or output relays, auxiliary relays, timers, and counters.

- - and - show driving of coils.

- - and - used in ladder indications indicate contacts. Contacts are in two states, conducting and non-conducting, depending on the ON/OFF state of input relays or output relays, auxiliary relays, timers, and counters.

- - and - show driving of coils.
4.1.2 Program mechanics

The internal sequence for the sequence controlling is created as the sequence program with the format of circuit diagram (ladder diagram) and instruction list.

A program is comprised of multiple instruction codes and device numbers (operand). These instructions are numbered in turn. This number is referred to as **step number**. (Step numbers are automatically controlled).

Each "**instruction**" is comprised of "**instruction code + device number**". However, there are some instructions without devices. Also in some cases, instruction codes are just referred to as instructions.

The max steps that can be programmed depend on the "program memory capacity" of the PLC that is used. For example, there is a program memory with the capacity of "2000" steps in FX1s PLC, "8000" steps in FX1N and FX2N, and "64000" steps in FX3U.

PLC repeatedly performs the instruction from step 0 to the END instruction. This operation is referred to as **cyclic operation**, and the time required to perform this one cycle is referred to as **operation cycle (scan time)**. Operation cycle will be changed according to the contents of the programs and the actual operating orders, ranging from several msec to several tens of msec.

The PLC program created by the format of circuit diagram (ladder diagram) is also stored in the program memory of the PLC with the format of instruction list (program list). The conversion between instruction list (program list) and circuit diagram (ladder diagram) can be done by using the programming software in personal computer.
4.1.3 N.O. contact, N.C. contact, OUR instruction, END instruction

**Circuit program**

The rail is called a “bus”.

- **LD** (Load) instruction is used for the first N.O. contact used on the bus, and the **LDI** (Load inverse) instruction is used for N.C. contacts. Contact instructions such as LD and LDI are used for devices such as input relays X, output relays Y, timers T, counters C, and auxiliary relays M.
- **Coil drive instruction** such as **OUT** are used for devices other than input relays X.
- **Instruction used for the end of a program**

By the above program...

- **Input X000**
  - OFF
  - ON
  - OFF
- **Output Y000**
  - OFF
  - ON
  - OFF
- **Output Y003**
  - ON
  - OFF
  - ON

Output Y000 turns ON when input X000 is ON. Output Y003 turns ON when input X003 is OFF.

**Reference**

**Programming by instruction list**

- **LD** (Load)
- **LDI** (Load inverse)
- **OUT** (Coil drive)
- **END** (Instruction used for the end of a program)

**List program**

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD</td>
</tr>
<tr>
<td>1</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>2</td>
<td>LDI X000</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>4</td>
<td>END</td>
</tr>
</tbody>
</table>

The LD (Load) instruction is used for the first N.O. contact used on the bus, and the LDI (Load inverse) instruction is used for N.C. contacts. Contact instructions such as LD and LDI are used for devices such as input relays X, output relays Y, timers T, counters C, and auxiliary relays M.

Coil drive instructions such as OUT are used for devices other than input relays X.
4.1.4 Series connection

Circuit program

By the above program...

Input X000
Input X001
Input X002
Input X003
Output Y000
Output Y003

Output Y000 turns ON when input X000 is ON, X001 is ON and X002 is OFF.
Output Y003 turns ON when input X000 is ON, X001 is ON, X002 is OFF, and X003 ON.

Programming by instruction list

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X000</td>
</tr>
<tr>
<td>1</td>
<td>AND X001</td>
</tr>
<tr>
<td>2</td>
<td>ANI X002</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>4</td>
<td>AND X003</td>
</tr>
<tr>
<td>5</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>6</td>
<td>END</td>
</tr>
</tbody>
</table>

Y000 and Y003 operate when all contacts serially connected that are driving AND and ANI are conducting.

The AND (AND) instruction is used for N.O. contacts and the ANI (AND Inverse) instruction is used for N.C. contacts connected serially following the LD and LDI instructions.
4.1.5 Parallel connection

Circuit program

By the above program...

Programming by instruction list

<table>
<thead>
<tr>
<th>OR</th>
<th>ORI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel connection instruction for N.O. contacts</td>
<td>Parallel connection instruction for N.C. contacts</td>
</tr>
</tbody>
</table>

List program

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X000</td>
</tr>
<tr>
<td>1</td>
<td>OR X001</td>
</tr>
<tr>
<td>2</td>
<td>ORI X002</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>4</td>
<td>END</td>
</tr>
</tbody>
</table>

The OR (OR) instruction is used for N.O. contacts and the ORI (OR Inverse) instruction is used for N.C. contacts connected in parallel following the LD and LDI instructions. Output Y000 in the figure on the left operates when even one of the contacts connected in parallel that is driving OR and ORI is conducting.
4.1.6 Series and parallel connections

**Circuit program**

Example program 1

![Circuit Diagram]

By the above program...

When input X000 or X001, and X002 or X003 turn on simultaneously, Y000 turns on.

Example program 2

![Circuit Diagram]

By the above program...

When the inputs X000 and X001 turn on simultaneously or X002 and X003 turn on simultaneously, the output Y000 turns on.

**Programming by instruction list**

**List program**

**Example program 1**

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X000</td>
</tr>
<tr>
<td>1</td>
<td>OR X001</td>
</tr>
<tr>
<td>2</td>
<td>LD X002</td>
</tr>
<tr>
<td>3</td>
<td>OR X003</td>
</tr>
<tr>
<td>4</td>
<td>ANB</td>
</tr>
<tr>
<td>5</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>6</td>
<td>END</td>
</tr>
</tbody>
</table>

**Example program 2**

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X000</td>
</tr>
<tr>
<td>1</td>
<td>AND X001</td>
</tr>
<tr>
<td>2</td>
<td>LD X002</td>
</tr>
<tr>
<td>3</td>
<td>AND X003</td>
</tr>
<tr>
<td>4</td>
<td>ORB</td>
</tr>
<tr>
<td>5</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>6</td>
<td>END</td>
</tr>
</tbody>
</table>

Refer to the appendix for the details of an ANB and an ORB command.
4.1.7. SET instruction, RST instruction

The SET/RST instructions are used for output relays Y and auxiliary relays M, for example. The RST instruction is also used for counters and retentive timers.

Circuit program

These instructions drive coils like the OUT instruction. With the OUT instruction, coils programmed with the OUT instruction also turn OFF when the contact that drives the coil changes state from ON to OFF. However, when the SET (Set) instruction is used, the operation state of the coil is held at ON even if the state of the contact changes from ON to OFF. The RST (Reset) instruction is used to change the state of the coil driven by the SET (Set) instruction from ON to OFF.

By the above program...

Output Y000 turns ON when input X001 is turned ON. After this, the operation state of Y000 is held at ON even if the state of input X001 changes from ON to OFF. Output Y000 turns OFF when input X003 is turned ON.

Reference

Programming by instruction list

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X001</td>
</tr>
<tr>
<td>1</td>
<td>SET Y000</td>
</tr>
<tr>
<td>2</td>
<td>LD X003</td>
</tr>
<tr>
<td>3</td>
<td>RST Y000</td>
</tr>
<tr>
<td>4</td>
<td>END</td>
</tr>
</tbody>
</table>
4.1.8 About Latch Circuits

Output operation can be held or cancelled by programming a self-hold sequence.

Circuit program

By the above program...

- Y003 turns ON when X003 is ON and X004 is OFF.
- Y003 continues to operate even if X003 is turned OFF. (This is "self-hold operation." ) Y003 turns OFF when X004 is turned ON.

Both circuits perform the same operation.

- [Output latch circuit]
- [Output SET/RST circuit]

When the SET instruction is used, the output will stay ON even when the contact driving the coil turns OFF. Also, it can be used multiple times for the same output without being treated as a double coil. It useful for directly controlling an output from multiple points in the program.

Reference

Programming by instruction list

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X001</td>
</tr>
<tr>
<td>1</td>
<td>OR Y003</td>
</tr>
<tr>
<td>2</td>
<td>ANI X003</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>4</td>
<td>END</td>
</tr>
</tbody>
</table>
4.2 About Timer Circuits

Timers

Of the various timers available, this section describes how to program a digital timer that is held internally on a micro PLC.

Circuit program

![Circuit diagram]

By the above program...

- The timer contact operates after a preset delay has elapsed after the coil has been made conductive. (ON delay timer)
- This preset time is called the "set value" and is expressed as K which can be set in the range 1 to 32,767.
- For example, K100 is a timer of ten seconds.
- When X000 is turned OFF while the timer is being driven, the present value of the timer returns to "0" and the timer contact also turns OFF.

- Input method

  Input: X000
  Output: Y000, T0 (coil)
  Output: T0 (N.O. contact)
  Output: Y003 (coil)

Reference

Programming by instruction list

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X000</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>OUT Y000</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>OUT T0 K100</td>
<td>7</td>
</tr>
</tbody>
</table>

*: When a 100 ms (0.1 second) timer is used
4.3 About Counter Circuits

Counters

Of the various counters available, this section describes how to program a general-purpose counter that is held internally on a micro PLC.

By the above program...

- Counters count the number of times a contact (X003) changes state from OFF to ON. This contact (X003) is called the "count input," and the value counted by the counter is called the "present value." The counter operates when the present value reaches the specified number (set value). The set value can be set a value within the range 1 to 32,767.
- After the counter has counted up, the present value of the counter does not change, and the output contact stays in an operating state.
- When reset input X001 turns ON, the present value of the counter returns to "0" and the counter contact also turns OFF.

Input method

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X001</td>
</tr>
<tr>
<td>1</td>
<td>RST C0</td>
</tr>
<tr>
<td>3</td>
<td>LD X003</td>
</tr>
<tr>
<td>4</td>
<td>OUT C0 K10</td>
</tr>
</tbody>
</table>

Reference

Programming by instruction list

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>LD C0</td>
</tr>
<tr>
<td>8</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>9</td>
<td>END</td>
</tr>
</tbody>
</table>
4.4 Order of Programs

Program order

Circuit diagrams are programmed from left to right and from top to bottom.

A program like the one above is made using 16 program steps, (1) to (16). The following shows the above program rewritten as a list program.

Reference

Programming by a command list

<table>
<thead>
<tr>
<th>Step</th>
<th>Instruction</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD</td>
<td>X000</td>
</tr>
<tr>
<td>1</td>
<td>OUT</td>
<td>Y000</td>
</tr>
<tr>
<td>2</td>
<td>LD</td>
<td>X001</td>
</tr>
<tr>
<td>3</td>
<td>OR</td>
<td>X002</td>
</tr>
<tr>
<td>4</td>
<td>AND</td>
<td>X003</td>
</tr>
<tr>
<td>5</td>
<td>OUT</td>
<td>Y001</td>
</tr>
<tr>
<td>6</td>
<td>OUT</td>
<td>Y002</td>
</tr>
<tr>
<td>7</td>
<td>LD</td>
<td>X004</td>
</tr>
<tr>
<td>8</td>
<td>AND</td>
<td>X005</td>
</tr>
<tr>
<td>9</td>
<td>LD</td>
<td>X006</td>
</tr>
<tr>
<td>10</td>
<td>AND</td>
<td>X007</td>
</tr>
<tr>
<td>11</td>
<td>ORB</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>OUT</td>
<td>Y003</td>
</tr>
<tr>
<td>13</td>
<td>OUT</td>
<td>Y004</td>
</tr>
<tr>
<td>14</td>
<td>OUT</td>
<td>Y005</td>
</tr>
<tr>
<td>15</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5
PROGRAM TRAINING

Enough for theory, and now for practice
In this chapter, you study all about handling and operation such as programming and monitoring based on some simple examples.
5.1 Introductory Example 1  
(Control of Escalators)

Let's consider a PLC program for an escalator.

**I/O assignments**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass detection switch: 0  X001</td>
<td>Illumination lamp  Y000</td>
</tr>
<tr>
<td>Pass detection switch: 1  X003</td>
<td>Escalator drive output  Y003</td>
</tr>
</tbody>
</table>

**Operation**

(1) Let's assume that the escalator does not move until someone approaches it.

(2) When someone approaches the escalator and turns on pass detection switch: 0, the illumination lamp lights and the escalator starts to move. (Escalator movement is up only).

(3) The person is carried up by the escalator, and when that person moves off the escalator, pass detection switch: 1 turns ON, and the illumination lamp and escalator return to a non-operating state.

In this example, consider both pass detection switches: 0 and 1 as non-hold type switches. Another assumption in this example is that people do not continue to get on this escalator.
The program is configured as follows

**〈Circuit program : Example 1〉**

![Circuit diagram](image)

**〈Circuit program : Example 2〉**

![Circuit diagram](image)

**〈Let's check operation〉**

Enter the above program on the PLC, and check program operation.

When input X001 is turned ON, outputs Y000 and Y003 operate. Next, when input X003 is turned ON, outputs Y000 and Y003 become non-operational.

Refer to the appendix 2 for a list program.
5.2 Introductory Example 2 (Control of Tea Dispenser)

Let's consider a PLC program for a tea dispenser

**I/O assignments**

**Input**
- Cup detection X000 (ON when cup is present)
- Dispense tea button X001
- Check tea leaves button X002
- Replenish water tank lower limit switch X003
- Replenish water tank upper limit switch X004

**Output**
- Change tea leaves indicator lamp Y000
- Dispense tea output Y001
- Replenish water output Y003

**Explanation of operation**

1. When cup detection X000 is ON, and dispense tea button X001 is pressed (X001 turns ON), dispense tea output Y001 operates, and hot water is poured into the cup. Hot water is poured only for the duration that the button is pressed, and stops being poured when you release your hand from the button. When cup detection X000 is OFF, hot water is not poured even by pressing dispense tea button X001.

2. When water in the replenish water tank gets low, lower limit switch X003 turns ON and replenish water output Y003 operates. When replenish water output Y003 operates and water is poured into the tank, upper limit switch X004 finally turns ON, and replenish water output Y003 becomes non-operational.

3. When water is replenished five times, the change tea leaves indicator lamp lights.

4. When the check button is pressed, the change tea leaves indicator lamp goes out.
The program is configured as follows

(The Circuit program)

- **X000** Cup detection
- **X001** Dispense tea button
- **Y003** Lower limit switch
- **Y004** Upper limit switch
- **Y001** Self hold
- **C0** Replenish water output
- **K5** Replenish water count
- **Y000** Change tea leaves indicator lamp
- **X002** Check button
- **RST** Counter reset

Let's check operation

Enter the above program on the PLC, and check program operation.

1. Output Y001 operates when both inputs X000 or X001 turns ON.
2. When input X003 turns ON, output Y003 operates, and when input X004 turns ON, output Y003 becomes non-operational.
3. Each time that output Y003 operates, the value of counter C0 is incremented by one. When C0 reaches "5", Y000 operates.
4. When input X002 turns ON, the value of counter C0 returns to "0" and output Y000 becomes non-operational.

Refer to Appendix 2 for the list program.
5.3 Introductory Example 3 (Control of Fan)

Let's consider a PLC program for a Fan.

**I/O assignments**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X000</td>
<td>Y000 [Ready indicator lamp]</td>
</tr>
<tr>
<td>X001</td>
<td>Y001 Weak wind output</td>
</tr>
<tr>
<td>X003</td>
<td>Y002 Electromagnetic contactor for fan</td>
</tr>
<tr>
<td>X005</td>
<td>Y003 Strong wind output</td>
</tr>
<tr>
<td>X006</td>
<td>X005 [Weak wind] selection</td>
</tr>
<tr>
<td>X006</td>
<td>X006 [Strong wind] selection</td>
</tr>
</tbody>
</table>

**Operation**

1. When the [Preparation] switch (X001) is turned ON, the [Ready indicator lamp] (Y000) turns ON and latched. When the [Preparation cancel] switch (X003) is turned ON, the [Ready indicator lamp] (Y000) turns OFF and unlatched.
2. When the [Fan start/stop] switch is pressed while the [Ready] indicator lamp (Y000) is ON, the electromagnetic contactor for the fan (Y002) is actuated and the fan starts or stops. The wind velocity of the fan can be selected by the selector switch (X005/X006).
   - When X005 is ON: The weak wind output (Y001) is ON.
   - When X006 is ON: The strong wind output (Y003) is ON.
Input required devices in blank areas, and complete the program.

• Refer to the last page of this chapter for the solution.
• Refer to the appendix for the list program.
Let’s consider a PLC program for a press machine.

### I/O assignments

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X001 Conveyer switch</td>
<td>Y001 Press start buzzer</td>
</tr>
<tr>
<td>X004 Workpiece counting sensor</td>
<td>Y002 Press output</td>
</tr>
<tr>
<td>Y003 Conveyer</td>
<td>Y003 Conveyer</td>
</tr>
</tbody>
</table>

### Operation

1. When the conveyer switch [Start] (X001) is turned ON, the conveyer (Y003) turns ON and carries workpieces to the pressing position.
2. When 4 workpieces are detected by the workpiece counting sensor (X004), the workpiece carrying conveyer (Y003) turns OFF and stops carrying workpieces.
3. When the conveyer (Y003) stops, the press start buzzer (Y001) sounds.
   - After 1 second, the press machine will perform pressing (Y002).
4. The press machine completes press (Y002) in 2 seconds, the start buzzer (Y001) sounds, and the press machine stops pressing at the same time.
5. After pressing is completed, the press workpiece count is reset, and operation is repeated from step (1).
《Circuit program》

Input required devices in blank areas, and complete the program.

* Refer to the last page of this chapter for the solution.
* Refer to the appendix for the list program.
5.5 Introductory Example 5 (Timing Chart)

Let's consider a PLC program for the timing chart below.

![Timing Chart Diagram]

**I/O assignments**

<table>
<thead>
<tr>
<th>Input</th>
<th></th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X001</td>
<td>Start switch</td>
<td>Y000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y004</td>
</tr>
</tbody>
</table>

**Operation**

When the start switch (X001) is turned ON, the indicator lamp (Y000) turns ON and the timer (T0) starts measuring the time at the same time.

On condition that the timer (T0) reaches the set value 2 seconds later, the buzzer (Y001) turns ON and the timer (T1) starts measuring the time at the same time.

After that, the electromagnetic contactor (Y002), motor (Y003) and solenoid valve (Y004) turn ON at intervals of 2 seconds in the same way. After the last solenoid valve (Y004) remains ON for 2 seconds, all outputs turn OFF.
《Circuit program》

Input required devices in blank areas, and complete the program.

• Refer to the last page of this chapter for the solution.
• Refer to the appendix for the list program.
## 5.6 Introduction example answers

<table>
<thead>
<tr>
<th>Introductory Example 3</th>
<th>Introductory Example 4</th>
<th>Introductory Example 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>① X001</td>
<td>① Y003</td>
<td>① Y000</td>
</tr>
<tr>
<td>② Y000</td>
<td>② C0 K4</td>
<td>② Y001</td>
</tr>
<tr>
<td>③ X003</td>
<td>③ T0</td>
<td>③ T1 K20</td>
</tr>
<tr>
<td>④ X000</td>
<td>④ T1</td>
<td>④ Y002</td>
</tr>
<tr>
<td>⑤ X005</td>
<td></td>
<td>⑤ T2 K20</td>
</tr>
<tr>
<td>⑥ X006</td>
<td></td>
<td>⑥ Y003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>⑦ T3 K20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>⑧ Y004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>⑨ T4 K20</td>
</tr>
</tbody>
</table>
Appendix 1
THE OPERATION OF THE GX Developer

Using a personal computer, sequence becomes easier...
If a personal computer software GX Developer is used to create and edit the sequence program, it becomes easier just like illustration making.
If the basic operations are mastered, it is just the repetition of them.
There are many functions easy to use in the software. Let's master them sequentially starting from the necessary operations.

Smoothly starting up and adjusting...
The program is accompanied by the debug.
The operation status of the PLC and program can be monitored on the personal computer screen, so if some parts are not working as planned, confirmation and adjustment can be conducted at once.

Make the program easy to read...
There is a "comment input function" in GX Developer to make the sequence program easy to read.
It can improve the efficiency to create and debug the program if a comment is input.
Appendix 1.1 Basic knowledge for operating GX Developer

Appendix 1.1.1 Construction of the GX Developer's screen

1) Title bar
The name of the opened project and the operation icon of windows are displayed.

2) Menu bar

3) Toolbar

4) Project data list

5) Edit screen

6) Status bar

Changing the size of GX Developer and quitting it
Maximizing and restoring down GX Developer

Display of the project name and its path
Minimizing GX Developer
Quitting GX Developer
2) Menu bar

Drop down menu will be displayed after menu is selected.

3) Toolbar

The description of the function will be displayed when the mouse cursor stops at the buttons.

*: The contents of the toolbar can be moved, added and removed, therefore the displayed items and the layouts depend on the various environments.

The frequently used functions are configured by icon buttons. Compared to selecting from the menu, the desired function can be directly executed.

4) Project data list

Directly specify the items displayed by mouse clicking.

Ladder creating window, parameter setting screen and so on are displayed by tree structure.
5) Edit screen

Ladder creating screen, monitor screen and so on are multiply displayed by windows.

6) Status bar
The status of the operation and setting is displayed.
Appendix 11.2 About the "Project"

"Project" consists of Program, Comment, Parameter and Device Memory. An aggregate of a series of data in GX Developer is called "Project", and stored as a folded of Windows®.

- **Editing multiple projects**
  Start multiple GX Developer when more than one projects are to be edited.
Appendix 1.2 Starting GX Developer and creating a new project
Appendix 1.2.1 Starting GX Developer

1) Start from the Start button of Windows®, and select in the following order. [Programs] ↓ [MELSOFT Application] ↓ [GX Developer]

2) GX Developer is started.
Appendix 1.2.2 Creating a new project

1) Select ![image](image1.png) from the toolbar, or select [Project] → [New project] (Ctrl + N) from the menu.

2) Click the ![image](image2.png) button of [PLC series].

3) Select "FXCPU".

4) Click the ![image](image3.png) button of [PLC type].

5) Select "FX3U(C)".

Note: Select the series name that is actually used.
6) Click **OK**.

7) New project screen is displayed and the project can be input.
Appendix 1.3 Creating a program

Appendix 1.3.1 Creating a program by using the function keys

[Program to be created]

Point

- The keyboard shortcuts for ladder elements are displayed on the buttons of the toolbar.

Main key operations

- When creating a program, make sure to set to "Write Mode".

Select from the toolbar. Select from the menu ([Edit] → [Write mode]).

- Use half-width characters when inputting. Full-width characters are not allowed.

Point

In this book, the number of the input relay and output relay is displayed as "X000", "Y000" by three digits. "X0", "Y1" and the "0" on the left front can be omitted when the input is performed from GX Developer.
1) Press the **F5** (→) key. Input "X2".

2) Confirm by pressing the **Enter** key or [OK].

3) The ladder input $$X_2$$ is displayed.

4) Press the **F6** (←) key. Input "X0".

5) Confirm by pressing the **Enter** key or [OK].

6) The ladder input $$X_0$$ is displayed.

7) Press the **F7** (←) key. Input "Y0".

8) Confirm by pressing the **Enter** key or [OK].

9) The ladder input $$Y_0$$ is displayed.

10) Press the **Shift + F5** (←) key. Input "Y0".

11) Confirm by pressing the **Enter** key or [OK].
12) The ladder element is displayed.
13) Move the cursor to the beginning of the next line.
14) Press the F5 (-→) key. Input "X3".
15) Confirm by pressing the Enter key or [OK].

16) The ladder element is displayed. 
17) Press the F7 (←→) key. Input "Y1".
18) Confirm by pressing the Enter key or [OK].

19) The ladder input (-Y1→) is displayed.

20) Convert Operation [Important]
Perform the "Conversion" operation to finalize ladder that not been compiled (the gray part).

Press the F4 (Convert) key.
Or select from the toolbar, or select [Convert] → [Convert] from the menu.
Appendix 1.3.2 Creating a program by using the toolbar buttons

[Program to be created]

- Click the toolbar buttons to input ladder elements.

- When creating a program, make sure to set to "Write Mode".

Select from the toolbar.

- Use half-width characters when inputting. Full-width characters are not allowed.

Point

In this book, the number of the input relay and output relay is displayed as "X000", "Y000" by three digits. "X0", "Y1" and the "0" in the left front can be omitted when the input is performed from GX Developer.

Main toolbar buttons
1) Click the tool button 
   Input “X2”.

6) The ladder element is displayed.

9) The ladder element is displayed.

1) Click the tool button 
   Input “X2”.

2) Confirm by pressing the Enter key or [OK].

3) The ladder input \( X_2 \) is displayed.

4) Press the tool button 
   Input “X0”.

5) Confirm by pressing the Enter key or [OK].

7) Input “Y0” 
   8) Click

3) The ladder element is displayed.

4) Input “X0” 
   5) Click

6) The ladder element is displayed.

7) Input “Y0” 
   8) Click

9) The ladder input \( Y_0 \) is displayed.

10) Input “Y0” 
    11) Click

Cancel it by ESC or [Exit].
12) The ladder input $Y_0$ is displayed.
13) Move the cursor to the beginning of the next line.
14) Click the tool button $F_4$.
   Input "X3".
15) Confirm by pressing the [Enter] key or [OK].

16) The ladder input $X_3$ is displayed.
17) Click the tool button $F_4$.
   Input "Y1".
18) Confirm by pressing the [Enter] key or [OK].

19) The ladder input $Y_1$ is displayed.

20) The gray shade disappears.

Convert Operation [Important]
Perform the "Conversion" operation to finalize ladder that not been compiled (the gray part).
Press the $F_4$ (Convert) key.
Or select $F_4$ from the toolbar, or select [Convert] → [Convert] from the menu.

The gray shade disappears and the program is compiled.
If an error occurs, the cursor moves to the problematic part of the program. Correct the program.
Appendix 1.4 Writing programs to PLC

Write the created sequence program to FX PLC.

Appendix 1.4.1 Connecting PC to PLC

Connecting the personal computer to an FX3G PLC (with a USB cable)

[Preparation on the personal computer side]

For connecting the personal computer to an FX3G PLC with a USB cable, it is necessary to install the USB driver software in the personal computer. Install the USB driver software in accordance with the procedure described in the “GX Developer Operating Manual (Startup)”.

[Connection diagram]
Appendix 1.4.2 "Transfer Setup" in GX Developer

Configure the settings of GX Developer to communicate with PLC.

1) Select [Online] → [Transfer Setup].
2) Double-click the icon.

3) Set the communication port of personal computer side. Choose "USB(GOT transparent mode)" for connecting to the built-in USB port of the FX3G PLC.
4) Click [OK] after the setting is completed.

5) Click [Connection test], to check communication with PLC.
6) After checking, click [OK] to confirm the setting configured.
Appendix 1.4.3 Writing a program to the PLC

1) Set the "RUN/STOP" switch of PLC to "STOP".

2) Select from the toolbar or select [Online] → [Write to PLC] from the menu.

3) Click [Param + Prog].
4) Click [Execute].

The dialog box of the progressing rate is displayed.

5) Click [OK] after it is completed.
Appendix 1.4.4 Monitoring PLC operation

1) Set the "RUN / STOP" switch of PLC to "RUN".

2) Select from the toolbar or select [Online] → [Monitor] → [Monitor mode] from the menu.

---

Check operation by monitor mode

1) Set [Switch X002 is "ON"] and [Switch X000 is "OFF"], and then check [Output Y000 is "ON"].
2) Check [Output Y000 is "ON"] while [Switch X002 is "OFF"].
3) Set [Switch X000 is "ON"] and then check [Output Y000 is "OFF"].
4) Check [Output Y001 is "ON/OFF"] in accordance with [Switch X003 is "ON/OFF"].
(1) The display of the monitor status dialog

<table>
<thead>
<tr>
<th>Monitor status</th>
<th>1ms</th>
<th>RUN</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Scan time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The maximum scan time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>of the sequence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>program is displayed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2) PLC status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The status of the PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>is displayed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3) Execution status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>of the monitor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This icon flashing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>when the monitor is</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>being executed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4) Memory type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The memory type of the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLC is displayed.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(2) Interpretation of the ladder monitor display

1) Contact Instruction

<table>
<thead>
<tr>
<th>Type</th>
<th>Input contact</th>
<th>X0: OFF</th>
<th>X0: ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO contact</td>
<td>X000</td>
<td>Contact open</td>
<td>X000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact close</td>
<td></td>
</tr>
<tr>
<td>NC contact</td>
<td>X000</td>
<td>Contact close</td>
<td>X000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact open</td>
<td></td>
</tr>
</tbody>
</table>

2) Out Instruction

<table>
<thead>
<tr>
<th>Type</th>
<th>Driving status</th>
<th>Non-execution/Non-drive</th>
<th>Execution/Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT instruction</td>
<td>(Y000)</td>
<td></td>
<td>Y000</td>
</tr>
<tr>
<td>SET instruction, etc.</td>
<td>[SET M0]</td>
<td></td>
<td>[SET M0]</td>
</tr>
</tbody>
</table>

The ON/OFF status of the device to be reset is displayed by the monitor using RST instruction.

<table>
<thead>
<tr>
<th>Type</th>
<th>Device status</th>
<th>When device to be reset is OFF</th>
<th>When device to be reset is ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST instruction</td>
<td>RST M0</td>
<td>--[RST M0]</td>
<td>--[RST M0]</td>
</tr>
</tbody>
</table>
Appendix 1.5 Editing a program

Appendix 1.5.1 Correcting a program

- Make sure to set to "Write Mode" when amending the circuit. Select from the toolbar. Select from the menu ([Edit]→[Write mode]).

- Use half-width characters when inputting. Full-width characters are not allowed.

- Switch between "Ovrwrte" and "Insert"
  - Set to "Ovrwrte" when correcting and overwriting ladder.
  - New ladder will be inserted when the "Insert" mode is on.

Switches between "Ovrwrte" and "Insert" by pressing the [Insert] key.
1) Changing the OUT coils and contacts

1) Double-click the part need to be corrected.

2) Change “Y001” to “Y002”.

3) Confirm by pressing the Enter key or [OK].
4) The corrected result is displayed and the ladder block is displayed in gray.

5) Confirm the changes by pressing the F4 (Convert) key.
2) Adding lines

[Program to be corrected]

1) Click (F10) on the toolbar.

2) Locate the cursor on the upper right of the desired vertical line to be added, and then drag it until it reaches the desired position, and then drop it.

3) The line reaching the dropped position is added.

4) Locate the cursor on the position where the OUT coil is to be added to and click (F10) on the toolbar.

5) Input "Y3".

6) Confirm by pressing the Enter key or [OK].
7) The added ladder is completed.

8) Confirm the changes by pressing the F4 (Convert) key

• Click on the toolbar again to finish the operation.

7) The added ladder is finished and the ladder block is displayed in gray.
3) Deleting lines

1) Click $\text{Alt} + F9$ on the toolbar.

2) Locate the cursor on the upper right of the desired vertical line to be deleted, and then drag it until it reaches the desired position, and then drop it.

3) The lines are deleted.
4) Delete the OUT coil by the Delete key.
5) The deleted ladder block is displayed in gray.

6) Confirm the changes by pressing the **F4** (Convert) key.

   • Click **F4** on the toolbar again to finish the operation.
Appendix 1.5.2 Inserting and deleting rows

1) Adding rows

[Program where a rung is to be inserted]

A row is inserted above the row where the cursor is located

1) Locate the cursor on the row below the one to be inserted.

2) Right click the mouse at any place, and select [Insert line].

3) A row is inserted.

4) Add ladder in the inserted row.

5) Confirm the changes by pressing the Convert (F4) key.
2) Deleting rows

1) Move to the row to be deleted

2) Right click the mouse at any place, and select [Delete line]

3) The row is deleted

Point

The line to be deleted is displayed in gray. Confirm it by pressing Convert (F4).
Appendix 1.5.3 Cutting and copying (pasting) ladder

1) Cut

[Program be edited]

1) Move the cursor to the beginning of the ladder to be cut.

2) Drag it until it reaches the desired position, and then drop it.

3) Select [X] from the toolbar or select [Edit] → [Cut] (Ctrl + X) from the menu, and execute cut.

4) The selected area is cut
   A gray part remains when a smaller portion of the ladder is cut. After amending the ladder, confirm the changes by pressing the F4 (Convert) key.
2) Copy(Pasting)

1) Move the cursor to the beginning of ladder to be copied.

2) Drag it until it reaches the desired position, and then drop it.

3) Select \( \text{from the toolbar or select [Edit] } \Rightarrow \text{[Copy] (Ctrl+C)} \) from the menu.

4) Move the cursor to the position where to paste.

Point

Switch by the \( \text{Insert} \) key

"Ovwrte" mode : Pastes by overwriting data from the cursor position.

"Insert" mode : Pastes it by inserting data at the cursor position.

5) Select \( \text{from the toolbar or select [Edit] } \Rightarrow \text{[Paste] (Ctrl+V)} \) from the menu.

A gray part remains when a smaller portion of the ladder is pasted. After amending the ladder, confirm the changes by pressing the \( \text{F4} \) (Convert) key.
Appendix 1.6 Saving the program
Appendix 1.6.1 Save as and save

Point
If there is ladder that has not been compiled, press [Convert] (F4).

1) Select 🎨 from the toolbar or select [Project] → [Save] (Ctrl + S) from the menu.

(When it is saved by overwriting)
Project saving is finished.

2) Specify the storage destination for the project.
3) Specify the project name.
4) Specify the title describing the project (optional).
5) Click Save.

6) Click Yes in the conformation dialog to finish.

If there is not sufficient space in the floppy disk to save the project, temporarily save it to the hard disk and then move it to the other floppy disk.

Reference
• The following characters cannot be used in the project name.
/ \ , > , < , , , ; , , , are only used to specify the driver
Also, do not use "." (period) at the end of the project name.
• When the project name is specified with 8 or more characters by GX Developer (later than SW6D5-GPPW), characters past the 8th character will not be displayed if read by the old versions (older than SW2D5-GPPW) of GX Developer.
• The project path plus its name is within 150 half-width characters (75 full-width characters).
• The title is within 32 half-width characters (16 full-width characters).
• If there are spaces in the project path and project name, GX Developer cannot start normally even if GPPW.gpj, ***.gps is double-clicked in the Explorer window.
If there are spaces in the project path and project name, open the project by starting GX Developer → selecting [Project] → [Open project] from the menu.
Appendix 1.6.2 Saving the project as a new one

**Point**
If there is ladder that has not been compiled, press `Convert` (F4).

1) Select `[Project] → [Save as]` from the menu.

2) Specify the storage destination for the project.
3) Specify the project name.
4) Specify the title describing the project (optional).
5) Click `Save`.

6) Click `Yes` in the confirmation dialog and finish.

For the way to name the driver/path and the project, see the previous page.

If there is not sufficient space in the floppy disk to save the project, temporarily save it to the hard disk and then move it to the other floppy disk.
Appendix 1.6.3 Reading the project

Reference

If another project is open when reading/opening a new file, the current project is closed. If there is uncompiled ladder in the project or the project is not saved, a warning message is displayed.

1) Select \( \text{from the toolbar or select [Project]} \) → [Open project] (\( \text{Ctrl} + \text{O} \)) from the menu.

2) Select the storage destination of the project.
3) Select the project to be read.
4) Click [Open] and read the project.
Appendix 1.7 Necessary operation for the debugging program

For connecting to the PLC and the writing a program to the PLC, see "Appendix 1.4 Writing programs to PLC".

Appendix 1.7.1 Ladder monitor

Display the ladder, and monitor the conduction status of the contacts and the driving status of the coils.

1) Select $\text{ }$ from the toolbar or select [Online] → [Monitor] → [Monitor mode] from the menu.

2) The ON/OFF status of the ladder and the current value of word devices (timer, counter and data register) are displayed in the ladder monitor window.

3) Right click the window, select [Stop monitor] to quit the ladder monitor.

4) In order to correct and write the program, select $\text{ }$ from the toolbar or select [Edit] → [Write mode] from the menu.
(1) the display of the monitor status dialog

1) Scan time
The maximum scan time of the sequence program is displayed.

2) PLC status
The status of the PLC is displayed.

3) The execution status of the monitor
This icon flashing when the monitor is being executed.

4) Memory type
The memory type of the PLC is displayed.

(2) Interpretation of the ladder monitor display

1) Contact Instruction

<table>
<thead>
<tr>
<th>Type</th>
<th>X0: OFF</th>
<th>X0: ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO contact</td>
<td>X000</td>
<td>X000</td>
</tr>
<tr>
<td></td>
<td>Contact open</td>
<td>Contact close</td>
</tr>
<tr>
<td>NC contact</td>
<td>X000</td>
<td>X000</td>
</tr>
<tr>
<td></td>
<td>Contact close</td>
<td>Contact open</td>
</tr>
</tbody>
</table>

2) Out Instruction

<table>
<thead>
<tr>
<th>Type</th>
<th>Non-execution/ Non-drive</th>
<th>Execution/Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT instruction</td>
<td>—(Y000) —</td>
<td>—Y000—</td>
</tr>
<tr>
<td>SET instruction, etc.</td>
<td>—[SET M0]—</td>
<td>—SET M0—</td>
</tr>
</tbody>
</table>

The ON/OFF status of the device to be reset is displayed by the monitor using RST instruction.

<table>
<thead>
<tr>
<th>Type</th>
<th>When device to be reset is OFF</th>
<th>When device to be reset is ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST instruction</td>
<td>—RST M0—</td>
<td>—[RST M0]—</td>
</tr>
</tbody>
</table>
Appendix 1.7.2 Device registration monitor

1) Registering optional devices

Register the optional devices in the monitor window and monitor the necessary part only.

1) Set the ladder monitor status. (See Section Appendix 1.7.1)

2) Select [Online] → [Monitor] → [Entry data monitor] from the menu. Or right click the ladder window and select [Entry data monitor].

3) Click [Register devices] in the "Entry data monitor" window.

4) Input the device number to be registered in the Register device window.

5) Click [Register].

6) The device is registered in the monitor window.

7) Click [Start monitor], and the value showing the device action and the ON/OFF status of the contacts and coils are displayed.
2) Registering the devices displayed in the ladder monitor

Specify the area of the ladder in the ladder monitor window and register all of the devices in the area.

1) Set to monitor mode. (See Section to Appendix 1.7.1)

2) Select [Online] → [Monitor] → [Entry data monitor] from the menu. Or right-click the ladder monitor window and select [Entry data monitor]. (See the previous page)

3) Select [Window] → [Tile horizontally] from the menu to display the "Ladder monitor window" and "Entry data monitor window" apposed together. (Set "Entry data monitor window" to the status of stop monitoring).

4) The "Ladder monitor window" and "Entry data monitor window" are displayed horizontally.

5) Click the start point of the ladder.

6) Click the end point while pressing the [Shift] key to select the area.

7) Drag the selected area to the "Entry data monitor window".

8) The devices are registered to the monitor window.

9) Click [Start monitor], and the value showing the device action and the ON/OFF status of the contacts and coils are displayed.
Appendix 1.7.3 Device batch monitor

Specify a device and monitor a continuous range of devices that follow it.

1) Set to monitor mode. (See Section Appendix 1.7.1)

2) Select [Online] → [Monitor] → [Device batch] from the menu. Or right click the ladder window and select [Device batch].

3) Input the first device of the devices to be monitored in the "Device batch monitor" window and then pressing the Enter key, and then click [Start monitor].

4) The value showing the device action and the ON/OFF status of the contacts and coils are displayed.
Appendix 1.7.4 Device test

1) Force ON/OFF

Forcedly turn ON/OFF the bit devices of PLC (M, Y, T, C and so on). (The forced ON/OFF function of X is not available).

When PLC is running, the forced ON/OFF function can turn on or off the devices only for one scan and the operations according to the sequence program are given priority. When checking the output, set PLC to the STOP status.

1) Set to monitor mode. (See Section Appendix 1.7.1)

2) Select [Online] → [Debug] → [Device test] from the menu. Or right click the ladder window and select [Device test].

3) Input the device number to be forcibly turned on/off.

4) • [FORCE ON]: Forces device to ON.
   • [FORCE OFF]: Forces device to OFF.
   • [Toggle force]: Forces the device to toggle ON/FF each time it is pressed.

Reference

Forced ON/OFF (Ladder monitor window)

Any bit device (contact and coil) can be forcibly turned on/off by double-clicking the device in the [Ladder monitor window] while pressing the [Shift] key.
2) Changing the current value of the word device

Change the current value of the PLC’s word device (T, C, D and so on).

1) Set to monitor mode. (See Section Appendix 1.7.1.)

2) Select [Online] → [Debug] → [Device test] from the menu. Or right click the ladder window and select [Device test].

3) Input the device number to be changed.

4) Input a new value.

5) Click [Set].
Appendix 1.7.5 Writing a program to the PLC during RUN

Write the corrected part of the ladder to the PLC when the PLC is running. Less time is needed for writing during RUN since the entire program is not transferred.

1) A contact will be added to the ladder on the left as an example. Within the ladder view, set the mode to write mode (写入).

2) Add the contact. The ladder block is displayed in gray.

3) Press [Shift] and [F4] together, or select [Convert] → [Convert (Write during RUN)] from the menu.

4) Click [Yes] to confirm the warning message about PLC safety regarding online changes.

5) The message "RUN write processing has completed." is displayed. Click [OK].

Caution

It is impossible to write the program to the PLC if the program in the PLC is different from the one in GX Developer. Verify in advance, or transfer the program first by using [Write to PLC].
## Appendix 1.8 Inputting comment

### Appendix 1.8.1 Types of the comment

The following 3 types of the comments can be input.

<table>
<thead>
<tr>
<th>Type</th>
<th>Purpose</th>
<th>The number of characters (full-width)</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Device comment</td>
<td>A comment describing the role and function of each device</td>
<td>16</td>
<td>It is necessary to set the &quot;Comments capacity&quot; in the parameter setting when writing to the PLC. The &quot;Comment range setting&quot; must also be set.</td>
</tr>
<tr>
<td>2) Statement</td>
<td>A comment describing the role and function of ladder blocks</td>
<td>32</td>
<td>This is a comment on GX Developer side. (It is not downloaded to the PLC).</td>
</tr>
<tr>
<td>3) Note</td>
<td>A comment describing the role and function of output instructions</td>
<td>16</td>
<td>This is a comment on GX Developer side. (It is not downloaded to the PLC).</td>
</tr>
</tbody>
</table>

### [Comment Examples]

- **1) Comment**
  
- **2) Statement**
  
- **3) Note**

### Point

**How to display the comment**

- Select [Display] → [Display comment] from the menu and then the comments are displayed.
- Repeat the operation above to stop displaying comments.
Appendix 1.8.2 Operation for creating device comments

1) How to input device comments using a list

1) Double click [Device comment] → [COMMENT] in the project list.

2) Double-click [Device comment] → [COMMENT] in the project list.

1) Input the start device of the devices to be commented in "Device name", and click [Display].

2) Input the device number

3) Input comments in the "Comment" column.

• When inputting comments for another device, Input the device number again in step 2.

2) How to input the device comments in the ladder

1) Click from the toolbar and double-click the ladder element to be commented.

2) Input the comment in the "Enter symbol" window and click [OK].

• Click on the toolbar again to finish the operation.
139

Point

Setting for writing the device comments to the PLC

It is necessary to set "Parameter setting" and "Comment range setting" in order to write the device comments to the PLC.

1) Parameter setting
   • Select [Parameter] → [PLC parameter] in the project list.
   • Set the "number of blocks" in the "Comments capacity" setting.
     One block is equivalent to 50 comments, occupying 500-steps of program memory.

2) Comment range setting
   • Select [Device comment] → [Comment], and then the comment input screen is displayed.
   • Select [Edit] → [Setup comment range] from the menu.
   • Set the type and the range of the devices to be written to PLC in the Setup comment range dialog.

Reference

Select [Tool] → [Options] from the menu, Check "Continues during command write" of the [Comment input] box in the [Program common] tab.
After configuring the settings as above, the operation of the ladder input continues and the "Enter symbol" window described in step 2) is displayed, when the program is being created.
Appendix 1.8.3 Operation for creating statements

1) Click \(\text{Double-click}\) from the toolbar, and double-click anywhere on the ladder block where the statement is to be written.

2) Input the statement in the "Enter line statements" window and click [OK].

   • Click \(\text{Double-click}\) on the toolbar again to finish the operation.

Appendix 1.8.4 Operation for creating notes

1) Click \(\text{Double-click}\) from the toolbar, and double-click the output instruction symbol where the note is to be written.

2) Input the note in the "Enter Note" window and click [OK].

   • Click \(\text{Double-click}\) on the toolbar again to finish the operation.
Appendix 1.9 Operation for creating the program list

In GX Developer, the program can also be created by list.

Appendix 1.9.1 Displaying the list edit screen

1) Create a new project (See Section Appendix 1.2.2) or display the ladder of the existing project.

2) Select from the toolbar or select [View] → [Instruction list] from the menu.

3) The list edit screen is displayed. Click on the toolbar again or select [View] → [Ladder] from the menu to return to the ladder display.

Appendix 1.9.2 How to input instructions

1) Input the instruction language from step 0 sequentially. The step number is added automatically when instruction is input. (For the input procedures, see the next page).
• How to input the basic instructions and the applied instructions
A "space" is input between the instruction language, device number and operand.

[Examples for the basic instructions]
LD  X0      Enter
OUT Y0     Enter
LDI X0      Enter
AND Y0     Enter
OUT M0     Enter
LD M0     Enter
OUT T0 K10 Enter
OUT C0 K5 Enter

Connection and OUT instructions
Coil instructions for the timer and counter

[Examples for the applied instructions]
MOV K1 D0   Enter
CMP K20 D3 M10 Enter

Reference
The key operation when inputting/editing
• "Ovrwrte"/"Insert" mode switches by pressing the Insert key.
• An instruction can be deleted by the Delete key.
• [Insert line] and [Delete line] operations can be done by the right click of the mouse.
Appendix 1.9.3 Checking the content of the list input

Confirm that there are no errors in the program created by list input in the ladder display.

1) Select from the toolbar, or select [View] → [Ladder] from the menu.

2) Check whether the program created by list input is displayed.

3) Select [Tools] → [Check program] to execute the program check, and check whether errors occur and the error steps.
Appendix 2
Introduction example list programs
Appendix 2.1 List programs

Introduction example 1

List program: Example 1

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X001</td>
</tr>
<tr>
<td>1</td>
<td>OR Y000</td>
</tr>
<tr>
<td>2</td>
<td>ANI X003</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>4</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>5</td>
<td>END</td>
</tr>
</tbody>
</table>

Introduction example 2

List program

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X000</td>
</tr>
<tr>
<td>1</td>
<td>AND X001</td>
</tr>
<tr>
<td>2</td>
<td>OUT Y001</td>
</tr>
<tr>
<td>3</td>
<td>LD X003</td>
</tr>
<tr>
<td>4</td>
<td>OR Y003</td>
</tr>
<tr>
<td>5</td>
<td>ANI X004</td>
</tr>
<tr>
<td>6</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>7</td>
<td>LD Y003</td>
</tr>
<tr>
<td>8</td>
<td>OUT C0 K5</td>
</tr>
<tr>
<td>11</td>
<td>LD C0</td>
</tr>
<tr>
<td>12</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>13</td>
<td>LD X002</td>
</tr>
<tr>
<td>14</td>
<td>RST C0</td>
</tr>
<tr>
<td>16</td>
<td>END</td>
</tr>
</tbody>
</table>

Introduction example 3

List program

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X001</td>
</tr>
<tr>
<td>1</td>
<td>OR Y000</td>
</tr>
<tr>
<td>2</td>
<td>ANI X003</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y000</td>
</tr>
<tr>
<td>4</td>
<td>LD Y000</td>
</tr>
<tr>
<td>5</td>
<td>AND X000</td>
</tr>
<tr>
<td>6</td>
<td>OUT Y002</td>
</tr>
<tr>
<td>7</td>
<td>AND Y002</td>
</tr>
<tr>
<td>8</td>
<td>MPS</td>
</tr>
<tr>
<td>9</td>
<td>AND X005</td>
</tr>
<tr>
<td>10</td>
<td>ANI Y003</td>
</tr>
<tr>
<td>11</td>
<td>OUT Y001</td>
</tr>
<tr>
<td>12</td>
<td>MPP</td>
</tr>
<tr>
<td>13</td>
<td>AND X006</td>
</tr>
<tr>
<td>14</td>
<td>ANI Y001</td>
</tr>
<tr>
<td>15</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>16</td>
<td>END</td>
</tr>
</tbody>
</table>

Introduction example 4

List program

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD X001</td>
</tr>
<tr>
<td>1</td>
<td>OR Y003</td>
</tr>
<tr>
<td>2</td>
<td>ANI C0</td>
</tr>
<tr>
<td>3</td>
<td>OUT Y003</td>
</tr>
<tr>
<td>4</td>
<td>LD X004</td>
</tr>
<tr>
<td>5</td>
<td>OUT C0 K4</td>
</tr>
<tr>
<td>8</td>
<td>LD C0</td>
</tr>
<tr>
<td>9</td>
<td>OUT Y001</td>
</tr>
<tr>
<td>10</td>
<td>OUT T0 K10</td>
</tr>
<tr>
<td>13</td>
<td>AND T0</td>
</tr>
<tr>
<td>14</td>
<td>OUT Y002</td>
</tr>
<tr>
<td>15</td>
<td>OUT T1 K20</td>
</tr>
<tr>
<td>18</td>
<td>LD T1</td>
</tr>
<tr>
<td>19</td>
<td>RST C0</td>
</tr>
<tr>
<td>21</td>
<td>END</td>
</tr>
</tbody>
</table>
Introduction example 5

**List program**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Command</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LD</td>
<td>X001</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>OR</td>
<td>Y000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ANI</td>
<td>T4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>OUT</td>
<td>Y000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>OUT</td>
<td>T0 K20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>LD</td>
<td>T0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>OUT</td>
<td>Y001</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>OUT</td>
<td>T1 K20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>LD</td>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>OUT</td>
<td>Y002</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>OUT</td>
<td>T2 K20</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>LD</td>
<td>T2</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>OUT</td>
<td>Y003</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>OUT</td>
<td>T3 K20</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>LD</td>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>OUT</td>
<td>Y004</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>OUT</td>
<td>T4 K20</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>END</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 3
Handy Programming Panel

By using the handy programming panel (HPP) ...
You can easily edit programs in list format using the HPP. For creating programs and debugging equipment on sites where personal computers cannot be always installed, the HPP is convenient to give simple modifications to programs and changing constants of timers and counters.

For checking the contents of errors ...
You can easily check the contents of errors using the monitor function and program check function of the HPP.

For storing programs and transferring programs to PLCs ...
The HPP FX-30P can store up to 15 sequence programs (or up to 7 sequence programs if their capacity exceeds 32000 steps). The personal computer is not required to update PLCs far way from the personal computer.

For export equipment ...
Because the HPP FX-30P offers display in Japanese, English and Chinese, it can be included with export machines.
Appendix 3.1 Handy programming panel (HPP)

You can easily edit programs in list format using the HPP FX-30P. For creating programs and debugging equipment on sites where personal computers cannot be always installed, the HPP is convenient for making simple modifications to programs and changing constants of timers and counters.

1) Functions convenient for setting up equipment
   • Editing programs using instruction list
   • Monitoring and forcing ON/OFF of devices

2) Functions convenient for troubleshooting
   • Program check function
   • PLC diagnosis function

3) Functions convenient for maintenance
   • Change of timer set values and counter set values
   • Transfer of updated sequence programs
     (The HPP can store up to 15 sequence programs whose capacity is 32000 steps or less).

4) Functions convenient for export equipment
   • Operation messages can be switched between English, simplified Chinese and Japanese.
## FX-30P Function List

### Online mode (Functions to directly access the memory inside the PLC)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming</strong></td>
<td></td>
</tr>
<tr>
<td>Read</td>
<td>Reads a sequence program (from the program memory inside the PLC to the screen on the FX-30P).</td>
</tr>
<tr>
<td>Write</td>
<td>Writes a sequence program (by key inputs in the FX-30P to the program memory inside the PLC).</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts instructions into a sequence program (by key inputs in the FX-30P into the program memory inside the PLC).</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes instructions from a sequence program (by key inputs in the FX-30P from the program memory inside the PLC).</td>
</tr>
<tr>
<td><strong>Monitor</strong></td>
<td></td>
</tr>
<tr>
<td>Offline select</td>
<td>Switches to offline mode.</td>
</tr>
<tr>
<td>PLC diagnosis</td>
<td>Diagnoses the PLC.</td>
</tr>
<tr>
<td>Memory cassette transfer</td>
<td>Transfers data to the memory cassette.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Sets parameters.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Sets keywords.</td>
</tr>
<tr>
<td>Device conversion</td>
<td>Executes device conversion.</td>
</tr>
<tr>
<td>Latch clear</td>
<td>Executes latch clear.</td>
</tr>
<tr>
<td>Device batch monitor</td>
<td>Executes device batch monitor.</td>
</tr>
<tr>
<td>BFM batch monitor</td>
<td>Executes buffer memory batch monitor.</td>
</tr>
<tr>
<td>Baud rate</td>
<td>Changes the baud rate.</td>
</tr>
<tr>
<td>PLC memory clear</td>
<td>Clears the memory inside the PLC.</td>
</tr>
<tr>
<td>Remote RUN/STOP</td>
<td>Changes over the PLC state between &quot;RUN&quot; and &quot;STOP&quot;.</td>
</tr>
<tr>
<td>PLC clock setting</td>
<td>Sets the clock in the PLC.</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td></td>
</tr>
<tr>
<td>Offline select</td>
<td>Switches to offline mode.</td>
</tr>
<tr>
<td>PLC diagnosis</td>
<td>Diagnoses the PLC.</td>
</tr>
<tr>
<td>Memory cassette transfer</td>
<td>Transfers data to the memory cassette.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Sets parameters.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Sets keywords.</td>
</tr>
<tr>
<td>Device conversion</td>
<td>Executes device conversion.</td>
</tr>
<tr>
<td>Latch clear</td>
<td>Executes latch clear.</td>
</tr>
<tr>
<td>Device batch monitor</td>
<td>Executes device batch monitor.</td>
</tr>
<tr>
<td>BFM batch monitor</td>
<td>Executes buffer memory batch monitor.</td>
</tr>
<tr>
<td>Baud rate</td>
<td>Changes the baud rate.</td>
</tr>
<tr>
<td>PLC memory clear</td>
<td>Clears the memory inside the PLC.</td>
</tr>
<tr>
<td>Remote RUN/STOP</td>
<td>Changes over the PLC state between &quot;RUN&quot; and &quot;STOP&quot;.</td>
</tr>
<tr>
<td>PLC clock setting</td>
<td>Sets the clock in the PLC.</td>
</tr>
</tbody>
</table>

### Offline mode (Functions to access the RAM memory inside the FX-30P)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming</strong></td>
<td></td>
</tr>
<tr>
<td>Read</td>
<td>Reads a sequence program (from the RAM built in the FX-30P to the screen on the FX-30P).</td>
</tr>
<tr>
<td>Write</td>
<td>Writes a sequence program (by key inputs in the FX-30P to the RAM built in the FX-30P).</td>
</tr>
<tr>
<td>Insert</td>
<td>Inserts instructions into a sequence program (by key inputs in the FX-30P into the RAM built in the FX-30P).</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes instructions from a sequence program (by key inputs in the FX-30P from the RAM built in the FX-30P).</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td></td>
</tr>
<tr>
<td>Online select</td>
<td>Switches to online mode.</td>
</tr>
<tr>
<td>Program check</td>
<td>Checks programs.</td>
</tr>
<tr>
<td>HPP-FX transfer</td>
<td>Transfers data between the RAM built in the FX-30P and the FX PLC.</td>
</tr>
<tr>
<td>HPP-PC transfer*</td>
<td>Transfers data between the RAM built in the FX-30P and the personal computer.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Sets parameters.</td>
</tr>
<tr>
<td>Device conversion</td>
<td>Executes device conversion.</td>
</tr>
<tr>
<td>PLC type</td>
<td>Changes the PLC type.</td>
</tr>
<tr>
<td>HPP memory clear</td>
<td>Clears the memory inside the FX-30P.</td>
</tr>
<tr>
<td>Program management</td>
<td>Manages programs stored in the RAM and flash memory (15 blocks) built in the FX-30P.</td>
</tr>
<tr>
<td><strong>HPP setting</strong></td>
<td></td>
</tr>
<tr>
<td>Buzzer sound volume</td>
<td>Adjusts the buzzer sound volume.</td>
</tr>
<tr>
<td>LCD contrast</td>
<td>Adjusts the LCD contrast.</td>
</tr>
<tr>
<td>Backlight brightness</td>
<td>Adjusts the backlight brightness.</td>
</tr>
<tr>
<td>Screen saver</td>
<td>Sets the screen saver.</td>
</tr>
<tr>
<td>HPP protect</td>
<td>Sets protection to programs stored in the FX-30P.</td>
</tr>
<tr>
<td>HPP initialization</td>
<td>Returns the FX-30P to factory default settings.</td>
</tr>
<tr>
<td>HPP F/W update</td>
<td>Updates the firmware stored in the FX-30P.</td>
</tr>
</tbody>
</table>

*: The firmware Ver. 1.10 or later supports transfer between the HPP and the personal computer.

### HPP setting (Setting functions for the FX-30P main unit)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
Appendix 4
Training Machine Wiring
Appendix 4.1 I/O Wiring Diagram for Training Machine

FX-I/O-DEMO2 Type Training Machine
# Revised History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/2006</td>
<td>A</td>
<td>First Edition</td>
</tr>
<tr>
<td>3/2007</td>
<td>B</td>
<td>Adding and revising the other descriptions.</td>
</tr>
<tr>
<td>2/2014</td>
<td>C</td>
<td>Adding and revising the other descriptions.</td>
</tr>
<tr>
<td>9/2014</td>
<td>D</td>
<td>Adding and revising the other descriptions.</td>
</tr>
<tr>
<td>5/2015</td>
<td>E</td>
<td>A part of the cover design is changed.</td>
</tr>
</tbody>
</table>
Effective May 2015
Specifications are subject to change without notice.