This manual explains the programming and monitoring procedures of the micro PC MELSEC-FX series using the FX-10P-E handy programming panel.

Read this manual carefully and thoroughly and make sure the specifications for handy programming panel are correct.

See the corresponding FX series PC manual for details about PC handing and instructions.

CONTENTS
1. Introduction ........................................... P1
2. Sample program operation ............................ P11
3. Programming .......................................... P21
4. Monitoring ............................................. P56
5. Test ..................................................... P65
6. Other functions ........................................ P75
7. Detailed procedure for starting up the system.... P89
8. Appendices ............................................. P96
Guidelines for the Safety of the User and Protection of the FX-10P-E HANDY PROGRAMMING PANEL.

This manual provides information for the use of the FX-10P-E HANDY PROGRAMMING PANEL. The manual has been written to be used by trained and competent personnel. The definition of such a person or persons is as follows:

a) Any engineer who is responsible for the planning, design and construction of automatic equipment using the product associated with this manual, should be of a competent nature, trained and qualified to the local and national standards required to fulfill that role. These engineers should be fully aware of all aspects of safety with regards to automated equipment.

b) Any commissioning or service engineer must be of a competent nature, trained and qualified to the local and national standards required to fulfill that job. These engineers should also be trained in the use and maintenance of the completed product. This includes being completely familiar with all associated documentation for said product. All maintenance should be carried out in accordance with established safety practices.

c) All operators of the completed equipment (See Note) should be trained to use this product in a safe manner in compliance to established safety practices. The operators should also be familiar with documentation which is connected with the actual operation of the completed equipment.

Note: The term ‘completed equipment’ refers to a third party constructed device which contains or uses the product associated with this manual.
Note’s on the Symbols Used in this Manual

At various times throughout this manual certain symbols will be used to highlight points which are intended to ensure the users personal safety and protect the integrity of equipment. Whenever any of the following symbols are encountered its associated note must be read and understood. Each of the symbols used will now be listed with a brief description of its meaning.

**Hardware Warnings**

1) Indicates that the identified danger **WILL** cause physical and property damage.

2) Indicates that the identified danger could **POSSIBLY** cause physical and property damage.

3) Indicates a point of further interest or further explanation.

**Software Warnings**

4) Indicates special care must be taken when using this element of software.

5) Indicates a special point which the user of the associate software element should be aware.

6) Indicates a point of interest or further explanation.
Under no circumstances will Mitsubishi Electric be liable responsible for any consequential damage that may arise as a result of the installation or use of this equipment.

All examples and diagrams shown in this manual are intended only as an aid to understanding the text, not to guarantee operation. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.

Please contact a Mitsubishi distributor for more information concerning applications in life critical situations or high reliability.
Caution on operation

Thoroughly read the manual and sufficiently assure safety before executing the operation to forcibly set devices to ON/OFF or the operation to change present values and set values of word devices in the test mode.
Otherwise, the machine may be damaged and accidents may occur by erroneous operations.
Version upgrade to support FX1S, FX1N, FX2N and FX2NC Series Programmable controllers. (ver 4.10 or later)

The FX-10P-E (ver 4.10 or later) fully supports all items and procedures used for the FX1S, FX1N, FX2N and FX2NC PLCs. Read section 1-3 in addition to the contents of the text when using the FX-10P-E (ver 4.10 or later).

1. **New Features**

<table>
<thead>
<tr>
<th>Table 1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Feature</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

○: Already compatible  ✗: Not yet compatible
2. List of Additional Devices and Instructions

- For details of each instruction and devices, refer to the FX programming manual.

### Table 2.1

<table>
<thead>
<tr>
<th>Devices</th>
<th>M0 to M3071</th>
<th>3072 points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auxiliary relays</td>
<td>M0 to M3071</td>
<td>3072 points</td>
</tr>
<tr>
<td>Data registers</td>
<td>D0 to D7999</td>
<td>8000 points</td>
</tr>
<tr>
<td>Index registers</td>
<td>V0 to V7, Z0 to Z7</td>
<td>16 points</td>
</tr>
</tbody>
</table>

### Table 2.2

<table>
<thead>
<tr>
<th>Basic Instructions</th>
<th>LDP</th>
<th>ORP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANDP</td>
<td></td>
<td>INV</td>
</tr>
<tr>
<td>INV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANDF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2.3

<table>
<thead>
<tr>
<th>Applied Instructions</th>
<th>FNC No.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNC No.</td>
<td>Instruction</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>PLSR</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>ECMP</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>EZCP</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>EBCCD</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>EBIN</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>EADD</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>ESUB</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>EMUL</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>EDIV</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Applied Instructions</th>
<th>FNC No.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNC No.</td>
<td>Instruction</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>ESQR</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>INT</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>SIN</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>COS</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>TAN</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>SWAP</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>DABS</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>ZRN</td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>PLSV</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Applied Instructions</th>
<th>FNC No.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNC No.</td>
<td>Instruction</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>DRVI</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>DRVA</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>TCMP</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>TZCP</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>TADD</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>TSUB</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>TRD</td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>TWR</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>HOUR</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Applied Instructions</th>
<th>FNC No.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNC No.</td>
<td>Instruction</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>GRY</td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>GBIN</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>RD3A</td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>WR3A</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>EXTR</td>
<td></td>
</tr>
<tr>
<td>224 to 246</td>
<td>In line Compare</td>
<td></td>
</tr>
</tbody>
</table>
3. **New Operations for FX2N, FX2NC, FX1N, FX1S Support**

3.1 **Basic and Applied instructions are added**

Symbol and input operation of additional basic instructions

*Table 3.1*

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Mnemonic</th>
<th>Key Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDP</td>
<td>Initial leading edge pulse contact</td>
<td>LDP</td>
<td>LD P/I Device Number GO</td>
</tr>
<tr>
<td>LDF</td>
<td>Initial trailing edge pulse contact</td>
<td>LDF</td>
<td>LD F Device Number GO</td>
</tr>
<tr>
<td>ANDP</td>
<td>Serial connection leading edge pulse contact</td>
<td>ANP</td>
<td>AND P/I Device Number GO</td>
</tr>
<tr>
<td>ANDF</td>
<td>Serial connection trailing edge pulse contact</td>
<td>ANF</td>
<td>AND F Device Number GO</td>
</tr>
<tr>
<td>ORP</td>
<td>Parallel connection leading edge pulse contact</td>
<td>ORP</td>
<td>OR P/I Device Number GO</td>
</tr>
<tr>
<td>ORF</td>
<td>Parallel connection trailing edge pulse contact</td>
<td>ORF</td>
<td>OR F Device Number GO</td>
</tr>
<tr>
<td>INV</td>
<td>Invert current logical status</td>
<td>INV</td>
<td>NOP P/I GO</td>
</tr>
</tbody>
</table>


Applied instruction input operation

<table>
<thead>
<tr>
<th>FNC</th>
<th>Applied Instruction No.</th>
<th>SP</th>
<th>Operand 1</th>
<th>SP</th>
<th>Operand 2</th>
<th>.....</th>
<th>GO</th>
</tr>
</thead>
</table>

The number of operands vary depending on the instruction.

- It is the responsibility of the operator to ensure that the instruction entered is valid for the current unit.

Application instruction input operation (by using the HELP function)

0 PROGRAM FLOW
1 TRANSFERS, COMP

Example: 0 1

10 CMP 12 MOV
11 ZCP 14 CML

- When using the HELP function with a model other than FX2N be careful not to select an invalid instruction.
3.2 Expansion of Device Range

For a list of the new device ranges see section 2.

Handling index registers

“V0” and “Z0” index registers are equivalent to registers “V” and “Z” respectively, so either one can be used for programming. When entering “V1” to “V7” or “Z1” to “Z7” add a numeral 1 to 7 to the “V” or “Z” before pressing \( \text{SP} \) or \( \text{GO} \).

3.3 Parameter setting

Addition of 16K steps

<table>
<thead>
<tr>
<th>MEMORY SETTING</th>
<th>2K</th>
<th>4K</th>
<th>8K</th>
<th>16K</th>
</tr>
</thead>
</table>

Setting of external RUN input

<table>
<thead>
<tr>
<th>RUN INPUT</th>
<th>USE</th>
<th>X002</th>
</tr>
</thead>
<tbody>
<tr>
<td>DON'T USE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When a RUN terminal is needed on the PC, select “USE”, and specify an input number from X000 to X017 (X000 to X007 only for FX2N-16M☆ and FX2NC-16M☆).

Number of file register blocks

0 to 14 blocks can be specified for file registers. These occupy program space, 1 block = 500 program steps, and are stored with the program. File registers start at D1000.
3.4 Others

Device conversion

Device number conversion is not possible between the following devices because the number of program steps required is different.

- M0 to M1535 ↔ M1536 to M3071 (FX2N and FX2NC only)
- 16-bit counter ↔ 32-bit counter (all FX models)

Writing the program to the FX-EEPROM-16

• When an instruction is inserted to a 16K-step program saved in the FX-EEPROM-16 in online mode, it may take 5 to 50 seconds depending on the total program size and the instruction insertion position.
The FX-10P-E handy programming panel (hereafter called the HPP) is a hand-held programming and monitoring panel used to write programs (sequence programs and parameters) to a PC and to monitor PC operations when connected to a MELSEC-FX series PC.

**List of Functions**

- **Read**
  - Reads existing sequence programs .................................................. P26
- **Write**
  - Overwrites and modifies sequence programs and writes new sequence programs. ... P31
- **Insert**
  - Inserts and adds instructions to sequence programs .......................... P50
- **Delete**
  - Deletes instructions from sequence programs ..................................... P52
- **Monitor**
  - Confirms the operating and control states of the PC. .......................... P58
- **Test**
  - Changes the current state or value of monitored devices ..................... P67
- **Others**
  - Displays selection screens, confirms and changes program states, and checks programs .................................................. P76
  - **Program check**
  - **Memory cassette transfer**
  - **Parameter**
  - **Device conversion**
  - **Latch clear**
  - **Buzzer control**
1. INTRODUCTION

This section explains the configuration of the HPP.

The HPP is composed of a liquid crystal display (2 lines x 16 characters) and a dedicated rubber key sheet (function, instruction, device symbol and numeric keys).

The HPP uses an FX-20P-CAB0 program cable (1.5 m) for communications with an FX0 PC. Use an FX-20P-CAB program cable for connections with an FX PC. Always make sure to use the proper cable for the corresponding PC.

FX-20P-CAB program cable 1.5m (4.92ft) for connection to FX.
FX-20P-CAB0 program cable 1.5m (4.92ft) for connection to FX0.
(Both cables are supplied separately).
1. INTRODUCTION

Panel Configuration

- LCD display unit (2 lines x 16 characters)
- HPP unit
- Connector (to the PC)

(Function keys)
- Dedicated keysheet
- Instruction keys
- Device symbol keys
- Number keys

- Other key
- Clear key
- Help key
- Space key
- Step key
- Cursor key
- Execution key

MITSUBISHI

FX-10P

W ▶ 234 LD M 55
235 ANI X 007
# 1. INTRODUCTION

## Key use

This section explains how each key is used.

<table>
<thead>
<tr>
<th><strong>Key</strong></th>
<th><strong>Use</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Function keys</td>
<td>The function keys are toggle keys. Pressing the key once calls the first function. Pressing it again calls the second function.</td>
</tr>
<tr>
<td>- Read/write</td>
<td></td>
</tr>
<tr>
<td>- Insert/delete</td>
<td></td>
</tr>
<tr>
<td>- Monitor/test</td>
<td></td>
</tr>
<tr>
<td>Other key</td>
<td>Even when other functions are in use, the mode menu selection screen is displayed.</td>
</tr>
<tr>
<td>Clear key</td>
<td>Cancels key operations before the [GO] key is pressed, clears error messages or returns to the previous screen.</td>
</tr>
<tr>
<td>Help key</td>
<td>Displays the applied instruction menu and switches the display between decimal and hexadecimal when the monitoring function is selected.</td>
</tr>
<tr>
<td>Space key</td>
<td>Used whenever advice or a constant is to be input.</td>
</tr>
<tr>
<td>Step key</td>
<td>Sets step numbers.</td>
</tr>
<tr>
<td>Cursor keys</td>
<td>Move the line cursor and prompt, scroll lines, and designate the numbers of devices preceding and following a designated device.</td>
</tr>
<tr>
<td>Execution key</td>
<td>Confirms and executes commands, scrolls information screen and continues a search.</td>
</tr>
<tr>
<td>Instruction keys</td>
<td>Each of these keys has two functions:</td>
</tr>
<tr>
<td>Device symbol keys</td>
<td>instructions (upper part) and device symbols or numbers (lower part). Which key function is effective during each operational step is automatically determined according to the currently executed operation.</td>
</tr>
<tr>
<td>Number keys</td>
<td>The following device symbol keys are toggled: [Z/V] key, [K/I] key, and [P/I] key.</td>
</tr>
</tbody>
</table>
This section explains how to connect the HPP to an FX PC.

When using an FX PC

Connector (to the HPP)

FX PC

FX-20P-CAB program cable
1.5 m (4.92 ft)

HPP unit

When using an FX0 PC

Connector (to the HPP)

FX0 PC

FX-20P-CAB0 program cable
1.5 m (4.92 ft)

HPP unit
1. INTRODUCTION

This section explains the procedure for connecting the HPP and a PC.

PC side
- Open the cover of the HPP connector on the PC base unit.
- Install the FX-20P-CAB0 program cable to the program connector of the PC base unit.

HPP side
- Install the FX-20P-CAB0 program cable to the HPP unit.
- After connecting the HPP and PC, turn ON the power supply to the PC.

---

Initial screen
- COPYRIGHT(C) 1992 MITSUBISHI
- MELSEC FX-10P-E V1.00

---

*1 Use FX-20P-CAB cable (optional) for connections with an FX PC. Now, tighten the FX-20P-CAB cable screws.

---

IMPORTANT

Do not touch the HPP connection ports. Doing so could cause the internal electronic circuits to be destroyed by static electricity.
1. INTRODUCTION

HPP outside dimensions

Hole Ø12 (0.47"

The hook slides to the position indicated by the dashed lines. The hook is used for hanging the HPP.

Weight: Approx. 0.25 kg (0.55 lb)

Units: mm (inches)
# 1. INTRODUCTION

## General specifications

<table>
<thead>
<tr>
<th>Items</th>
<th>General Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating ambient temperature</td>
<td>0 to 40 °C</td>
</tr>
<tr>
<td>Operating ambient humidity</td>
<td>35 to 85 % RH (no condensation)</td>
</tr>
<tr>
<td>Vibration resistance</td>
<td>Conforms to JIS C0911.</td>
</tr>
<tr>
<td>Vibration frequency</td>
<td>10 to 55 Hz</td>
</tr>
<tr>
<td>Acceleration</td>
<td>1 G</td>
</tr>
<tr>
<td>Amplitude</td>
<td>0.1 mm</td>
</tr>
<tr>
<td>Shock resistance</td>
<td>Conforms to JIS C0912. (10G, 3 times in 3 (X, Y, and Z) directions)</td>
</tr>
<tr>
<td>Operating ambience</td>
<td>Free from corrosive gases. Dust should be minimal.</td>
</tr>
</tbody>
</table>

## Performance specifications

<table>
<thead>
<tr>
<th>Items</th>
<th>Performance Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power supply voltage</td>
<td>5 V DC ± 5 % electric power is supplied from the PC.</td>
</tr>
<tr>
<td>Current consumption</td>
<td>120 mA</td>
</tr>
<tr>
<td>Display contents</td>
<td>Graphic display</td>
</tr>
<tr>
<td></td>
<td>1 character: 8 x 5 = 40 dots; 1 x 5 dots at the bottom is for the prompt.</td>
</tr>
<tr>
<td>Number of displayed characters</td>
<td>16 characters x 2 lines = 32 characters</td>
</tr>
<tr>
<td>Character types</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>Keyboard</td>
<td>35 keys</td>
</tr>
<tr>
<td>Built-in interface</td>
<td>Conforms to EIA RS422. FX-20P-CAB cable, FX-20P-CAB0 cable</td>
</tr>
<tr>
<td>Outside dimensions</td>
<td>160 x 85 x 27 (mm) (6.30 x 3.35 x 1.06 (in))</td>
</tr>
<tr>
<td>Weight</td>
<td>0.25 kg (0.55 lb)</td>
</tr>
</tbody>
</table>
*1 "PC state" shows the valid PC RUN/STOP states for this key sequence (shaded areas are valid).

*2 "Valid memory" shows the PC program memory areas for which this key sequence is valid (shaded areas are valid). However, RAM indicates either PC RAM or a RAM cassette.
1. INTRODUCTION

Explanation of the key sequences used in this manual

A \rightarrow B : Press either the [A] or [B] key.
A \rightarrow B : Press the [A] key, and then press the [B] key.
A \rightarrow B : Press the [A] key several times, and then press the [B] key.
A \rightarrow \text{N times} : Press the [A] key n times.
RST + GO : Press and hold the [RST] key, then press the [GO] key (for resetting the HPP).

Abbreviations and their meanings

The following abbreviations are used in this manual:
- FX-10P-E handy programming panel ................................................................. HPP
- FX series PC (Programmable Controller) ............................................................ PC
- ▶ symbol that is displayed on the left side of the display screen (Execution line) .......... Line cursor
- ▣ symbol that blinks on the left side of the display screen (Item selection line) ............ Cursor
- Underline _ symbol displayed under a character on the display screen (Key input wait place) .......... Prompt
This section explains the actual operation of the HPP and how to create programs using a simple training example. Operations are confirmed by using the monitoring and test functions after creating the program.

Start by operating the HPP. Since only the basic operation of the HPP is explained in this section, see subsequent sections for details about the different modes and operations.

Prepare the following training devices beforehand.
- PC (PC base unit: FX0-14MR-ES)
- FX-20P-CAB0 programming cable
- HPP (FX-10P-E handy programming panel)
- Simulation switches

**Training procedure**

The training procedure is as follows:

1. Connect the HPP and PC
2. Mode selection
3. Program creation
4. Monitoring operation
5. Test operation
6. Training completed

---

Page 6 gives details about connecting the HPP and PC.

Operating the HPP. To prepare program creation, select [Write].

Erase the program in the PC's RAM and write the program following the example.

Confirm whether or not the program is operating correctly by monitoring the program's designated device.

Turn the designated device ON/OFF by using the HPP keys.

After completing the steps above the basic operations of the HPP will be understood.
Mode selection determines which function will be operated by the HPP keys.

After connecting the HPP and PC, turn the PC power supply ON.

Display screen

COPYRIGHT(C) 1992 MITSUBISHI

MELSEC FX-10P-E V1.00

Mode setting

SELECT FUNCTION FX [2K FXRAM]

RD

WR

Begin program creation
Write the sample program, following the examples below, after executing an NOP batch write (all programs erased) to the PC's RAM.

Confirm that RUN/STOP switch of the PC is set to STOP before continuing.

**NOP batch write**

When writing a new program, clearing the PC RAM (NOP batch write designating "all area") is done by the following operation:

```
RD → WR → NOP → A → GO → GO → W
```

**Screen configuration**

The screen configuration during HPP program creation is indicated below.

The following functions are displayed using alphabetical abbreviations:
- R: Read
- W: Write
- I: Insert
- D: Delete
- M: Monitor
- T: Test

- Line cursor: This shows the execution line of the function.
- Device symbol: Indicates the HPP is waiting for input at this position.
- Device numbers: If an application instruction is used, the application instruction number is displayed.
- Instruction
- Step number

2 lines x 16 characters
The sample program used is shown below.

```
0  LD  X000
1  AND X001
2  OUT Y000
3  LD  X002
4  OR  X003
5  OUT Y001
6  LD  X004
7  ANI  T 1
8  OUT  T 0  K10
11  LD  T 0
12  OUT  T 1  K10
15  OUT Y002
16  LD  X005
17  RST  C 0
19  LD  X006
20  OUT  C 0  K10
23  LD  C 0
24  OUT Y003
25  END
```
Write the sample program using the following key operations:

### Key Operations

1. **LD** X 0 GO AND X
   - X
   - GO
   - OUT Y 0 GO
   - 1

2. **LD** X 2 GO OR X
   - X
   - GO
   - OUT Y 1 GO
   - 3

3. **LD** X 4 GO ANI T
   - T
   - GO
   - OUT T 0 SP
   - 1

### Display Screens

- W 0 LD X 000
  - 1 AND X 1
- W 1 AND X 001
  - 2 OUT Y 0
- W 2 OUT Y 000
  - 3 LD X 2
- W 4 OR X 003
  - 5 OUT Y 1
- W 6 LD X 004
  - 7 ANI T 1
- W 8 OUT T 0
  - K 10
* After writing the program, use the following key operations to read and check the program:
Confirm the operation of the created program by using simulation switches. Connect the simulation switches and the PC as shown below. First turn OFF the power supply and then make the connection.

Simulation switch
(Simulation input switch: FX0-14SW-E)

Programmable Controller
(PC base unit: FX0-14MR-E)
2. SAMPLE PROGRAM OPERATION

Confirm the operation of the sample program by using the monitoring function.

Turn ON the power supply and set the RUN/STOP switch of the PC to RUN.

(1) Read from Y0 to Y3 individually on the display screen of the HPP.

```
MNT ➔ SP ➔ Y ➔ 0 ➔ GO ➔ ↓ ➔ ↓ ➔ T ➔ Y 0 0 0 ➔ Y 0 0 1
       ↑ ➔ Y 0 0 2 ➔ Y 0 0 3
```

* The display screen on the left appears when the [↓] key is pressed three times.

(Example when Y1 is ON)

(2) Confirm whether Y0 to Y3 operate as follows when simulation switches [X0] to [X6] are operated on the display screen of the HPP. (Begin operations after turning [X0] to [X7] OFF.)

<table>
<thead>
<tr>
<th>Simulation Switch Operations</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Turn [X0] and [X1] ON.</td>
<td>[Y0] goes ON.</td>
</tr>
<tr>
<td>(b) Turn either [X2] or [X3] ON.</td>
<td>[Y1] goes ON.</td>
</tr>
<tr>
<td>(c) Turn [X4] ON.</td>
<td>[Y2] flashes.</td>
</tr>
<tr>
<td>(d) Turn [X6] ON then OFF ten times.</td>
<td>[Y3] goes ON.</td>
</tr>
<tr>
<td>(e) After [Y3] goes ON, turn [X5] ON.</td>
<td>[Y3] goes OFF.</td>
</tr>
</tbody>
</table>

If (a) to (e) do not give the expected results the ladder is faulty. Use the read function to read and recheck the program.
2. SAMPLE PROGRAM OPERATION

Using the test function to turn designated devices of an already created program ON and OFF with the HPP keys.

Make sure the RUN/STOP switch is set to STOP.

(1) Read Y0 to Y2 individually on the HPP display screen.
   (This procedure is the same as the monitoring function read procedure.)

   MNT → SP → Y → 0 → GO → ↓ → ↓ → M Y 0 0 0 Y 0 0 1
   * The display screen on the left appears when the [↓] key is pressed twice.

(2) After reading Y0 to Y2 on the display screen, try to turn ON and OFF Y1 forcibly as a test.

   M Y 0 0 0 Y 0 0 1
   TEST → ↑
   * This is the state for reading Y0 to Y2 on the screen.
   * Press the [TEST] key to switch to the test function. Then, designate the target device number by using the [↑] / [↓] keys.

   → T Y 0 0 0 Y 0 0 1
   SET
   * The function mode display changes from M to T line cursor points to Y1.
   * Press the [SET] key to turn Y1 ON forcibly.

   → T Y 0 0 0 Y 0 0 1
   RST
   * The ■ symbol is displayed before Y1, and Y1 goes ON.
   * Press the [RST] key to turn Y1 OFF forcibly.

   → T Y 0 0 0 Y 0 0 1
   * The ■ symbol before Y1 disappears and Y1 goes OFF.

Repeat the above operation for Y0 to Y2, and confirm whether or not they all turn ON and OFF.
Programming consists of creating, modifying, and reading the sequence program in the list format. Connect the HPP to the PC and directly access the PC program memory by using the HPP.

The program memory is composed of parameters and programs.

When the FX₀ series is used, programs are written to the EEPROM built in the PC.

When the FX series is used, if a memory cassette has not been installed in the PC, programs are written to the RAM built in the PC. If a memory cassette has been installed in the PC, programs are written to the memory cassette.

When the FX series is used, transfers between the PC built-in RAM and the memory cassette installed in the PC can be done with the HPP (see page 78).

* Writing and transfer cannot be done if the memory cassette is an EPROM or the memory-protect switch of the EEPROM is set to ON.

[FX₀ series]

HPP → PC → Built-in EEPROM → HPP

[FX series]

HPP → PC → Built-in RAM → Transfer → Memory Cassette
3. PROGRAMMING

**General function description**

Programming functions:

- **Read**
  - Reads existing programs from the program memory.
  - There are four ways of reading - step, instruction, device and pointer.

- **Write**
  - Creates new programs, overwrites and adds to existing programs.

- **Insert**
  - Inserts an instruction in a created program and sequentially increases the step numbers following the insertion.

- **Delete**
  - Deletes a designated instruction of a program and sequentially reduces the step numbers after the deletion.
  - Includes the deletion of instructions and pointers, NOP batch deletion and deletion by designating a range.

---

**POINT**

1. Usable program steps display
   - The usable program steps display can be displayed by pressing the [HELP] key during programming.
   - Pressing the [HELP] key again clears this display.

2. Applied instructions
   - The applied instruction menus can be displayed and searched by pressing the [FNC] and [HELP] keys.
3. PROGRAMMING

Programming outline

The programming procedure outline is shown below.

Connect the HPP and PC.

Turn ON the PC power supply.  Reset the HPP ([RST] + [GO])

The mode selection screen is displayed.

Read  Write  Insert  Delete

Program creation

(See page 6.)

(See page 89.)
3. PROGRAMMING

Programming operations

There are two programming methods, writing a new program and editing an existing program.

After starting up the HPP, all functions can be selected from the mode selection screen.

Key sequences and programming operations are shown below.

(1) Writing a new program.
   
   RD → WR  Write the program from step 0

(2) Editing an existing program.

   RD  Reading the corresponding instruction
   WR
   INS
   INS  DEL

   The existing program will be overwritten.
   The required instructions can be inserted.
   Unnecessary instructions can be deleted.
3. PROGRAMMING

How to create a program

Creating a program using an instruction list.
Instructions consist of basic and applied instructions (see pages 96 to 98).

The basic operation for creating a program is shown below.

(1) Basic instructions (including step ladder instructions)
   Directly input instructions to be set.

(2) Applied instruction
   Input [FNC] <applied instruction number>
   (The list of applied instructions can be displayed by pressing the [FNC] and [HELP] keys.)

Completion

Programs are directly written to the PC program memory.

While programming, switching to other functions and menus is possible by pressing a function key or the [OTHER] key.
3. PROGRAMMING

**Reading by step number**

Designate a step number to read and display a program.

**Basic operation**

![Diagram of Basic Operation]

**Sample operation 1**

To read step 55 of the program

**Sample display**

1. Press the [STEP] key and enter the step number 55.
2. Press the [GO] key to read the desired program lines.

- Two lines of the program are read and displayed beginning with the designated step number.
- If the designated step number corresponds to an operand, such as a T/C setting value, two lines of the program are read and displayed beginning with the corresponding instruction.
- The following instructions are displayed sequentially by pressing the [GO] key repeatedly.
- Programs can be displayed by scrolling one line at a time using the cursor key.
3. PROGRAMMING

Reading by instruction

Designate an instruction to read and display a program.

Basic operation

![Diagram showing the process of reading by instruction]

PC STATE: [RUN] [STOP] [VALID MEMORY: [RAM] [EEPROM] [EPROM]

Sample operation 1
To read PLS M104

Sample display

R 100 OUT Y 002
INSTR. PLS M 104
R ▶ 350 PLS M 104
351 LD M 103
R ▶ 370 PLS M 104
LD M 116

(1) Press an instruction key and enter the device number, if applicable.
[In the case of an application instruction]
Input [FNC], [D], [1], [2], [GO], or [FNC], [1], [2], [GO]. Both instructions are searched whether or not pulse symbol [P] is pressed.

* The designated instruction is searched for starting from step 0. And then, two lines of the program are displayed on the screen beginning with the first detected instruction.
* The next and following occurrences of this instruction are found by pressing the [GO] key repeatedly.
* When the search of all designated instructions has been completed, or the designated instruction is not found, the "NOT FOUND" message will be displayed (Instructions after the END instruction are not read).
* Pressing the cursor key switches the reading mode to step order. This is the same for a pointer and a device.
### 3. PROGRAMMING

#### Basic sequence instructions and applicable devices

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Functions</th>
<th>Devices</th>
<th>Instructions</th>
<th>Functions</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD LOAD</td>
<td>Logical operation start (NO contact)</td>
<td>X, Y, M, S, T, C Special M</td>
<td>LDI LOAD INVERSE</td>
<td>Logical operation start (NC contact)</td>
<td>X, Y, M, S, T, C Special M</td>
</tr>
<tr>
<td>AND AND</td>
<td>Logical product (NO contact serial connection)</td>
<td>X, Y, M, S, T, C Special M</td>
<td>ANI AND INVERSE</td>
<td>Negative AND (NC contact serial connection)</td>
<td>X, Y, M, S, T, C Special M</td>
</tr>
<tr>
<td>OR OR</td>
<td>Logical OR (NO contact parallel connection)</td>
<td>X, Y, M, S, T, C Special M</td>
<td>ORI OR INVERSE</td>
<td>Negative OR (NC contact serial connection)</td>
<td>X, Y, M, S, T, C Special M</td>
</tr>
<tr>
<td>ANB AND BLOCK</td>
<td>Serial connection between blocks</td>
<td>*</td>
<td>ORB OR BLOCK</td>
<td>Parallel connection between blocks</td>
<td>*</td>
</tr>
<tr>
<td>OUT OUT</td>
<td>Coil drive</td>
<td>Y, M, S, T, C Special M</td>
<td>NOP NOP</td>
<td>No processing</td>
<td>For erasing programs or a space *</td>
</tr>
<tr>
<td>SET SET</td>
<td>Latch a device ON.</td>
<td>Y, M, S Special M</td>
<td>RST RESET</td>
<td>Latch a device OFF.</td>
<td>Y, M, S, T, C, V, Z, D, Special M, Special D</td>
</tr>
<tr>
<td>MC MASTER CONTROL</td>
<td>Common serial contact</td>
<td>N, Y, M</td>
<td>MCR MASTER CONTROL RESET</td>
<td>Common serial contact cancellation</td>
<td>N</td>
</tr>
<tr>
<td>PLS PULSE</td>
<td>A one-operation cycle pulse is generated at the leading edge of an input signal (turning on).</td>
<td>Y, M</td>
<td>PLF PULSE</td>
<td>A one-operation cycle pulse is generated at the fall of an input signal (turning off).</td>
<td>Y, M</td>
</tr>
<tr>
<td>MPS PUSH</td>
<td>Operation memory</td>
<td>*</td>
<td>MRD READ</td>
<td>Memory read</td>
<td>*</td>
</tr>
<tr>
<td>MPP POP</td>
<td>Memory read and reset</td>
<td>*</td>
<td>END END</td>
<td>Program completion</td>
<td>The final instruction at the end of a program. The program is returned to step 0.</td>
</tr>
<tr>
<td>STL STEP LADDER</td>
<td>Start of the step ladder</td>
<td>S</td>
<td>RET RETURN</td>
<td>Completion of the step ladder</td>
<td>*</td>
</tr>
</tbody>
</table>

- Instructions with an * symbol do not need an applicable device.
- Special M: Special Auxiliary Relay
- Special D: Special Data Register
3. PROGRAMMING

Read by a pointer

Designate a pointer to read and display the program.

Basic operation

![Diagram of pointer operation]

PC STATE: RUN STOP VALID MEMORY: RAM EEPROM EPROM

Sample operation 1

To read the label of pointer number 3

Sample display

![Sample display diagram]

(1) Press the [P] key and enter the pointer number, then press the [GO] key. The designated pointer number is read.

* Two lines of the program are read and displayed beginning with the designated label.

* If the designated label is not found, the "NOT FOUND" message will be displayed. (Instructions after the END instruction are not read.)

* This pointer search only looks for a pointer as a label. A pointer designated as an operand in an applied instruction will not be searched.

POINT

(1) P (pointer): The device that identifies the label to jump for a branch instruction.

(2) Label: The number that identifies the first step of branch program.

(3) I (interrupt pointer): The label for an interrupt program.

An interrupt program must end with an IRET (interrupt return) instruction.
3. PROGRAMMING

Read by a device

Designate a device symbol and a device number to read and display a program.

Basic operation

<table>
<thead>
<tr>
<th>RD</th>
<th>SP</th>
<th>Device symbol</th>
<th>Device number</th>
<th>GO</th>
<th>↑</th>
<th>↓</th>
</tr>
</thead>
</table>

PC STATE: [RUN] [STOP] [VALID MEMORY: RAM EEPROM EPROM]

Sample operation 1

Sample display

1. Press the [SP] key and enter a device symbol and a device number and then press the [GO] key.

* The designated device is searched for beginning with step 0. Two lines of the program are displayed beginning with the first detected instruction that contains the device.

* The next and following instructions containing this device are found by pressing the [GO] key repeatedly.

* When the search of all designated devices has been completed, or the designated device is not found, the "NOT FOUND" message will be displayed (Instructions after the END instruction are not read).

POINT

1. Precautions when reading using a device

Only the X, Y, M, S, T, C, D, V, and Z basic instruction device can be searched by a device read (V and Z as indices are not searched).

Only a D device used with the OUT instruction of a timer and a counter and the RST D instruction will be searched.
3. PROGRAMMING

Input of a basic instruction

There are three ways of entering a basic instruction and a step ladder instruction:

- Instruction only, instruction and device, instruction and two devices.

These three input methods are explained below.

Basic operation

(1) Instructions that do not require any devices:
The ANB, ORB, MPS, MRD, MPP, RET, END and NOP instructions are written without designating a device.

Sample operation 1

To input (write) the ORB instruction

Sample display

* Press the [ORB] key and then press the [GO] key. Writing has now been completed.
3. PROGRAMMING

(2) Instructions that are written by a device are entered as instructions and devices.

The LD, LDI, AND, ANI, OR, ORI, SET, RST, PLS, PLF, MCR, STL and OUT (except T and C) instructions are written by inputting an instruction and a device.

Sample operation 2

To input (write) the LD instruction

Sample display (Before confirmation)

(W) 0 LD X 0
1 NOP

(GO)

Write function

* Press the [LD], [X] and [0] keys.
* When the HPP is waiting for a device symbol or a device number key to be input the prompt appears at the end of the line.
* Press the [GO] key. Writing has now been completed.

Sample operation 3

To input (write) the MCR N1 instruction

Sample display

(W) 1 2 MCR N 1
1 4 NOP

Write function

* The nesting level device symbol "N" is automatically displayed when the [MCR] key is pressed. Then press the [1] key and the [GO] key to complete the writing.

The program step numbers are determined automatically. See page 96 for the step numbers occupied by each instruction.
3. PROGRAMMING

(3) Instructions that are written by entering an instruction, and two devices: MC and OUT (T,C) instructions.

Sample operation 4
To input (write) the OUT T10, K19 instruction

Sample display (Before confirmation)

<table>
<thead>
<tr>
<th>W</th>
<th>1 0 0</th>
<th>OUT</th>
<th>T 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>K 1 9</td>
</tr>
</tbody>
</table>

(After confirmation)

<table>
<thead>
<tr>
<th>W</th>
<th>K 1 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 0 3</td>
</tr>
</tbody>
</table>

* Press the [OUT], [T], [1], [0], [SP], [K], [1] and [9] keys and then press the [GO] key. Writing has now been completed.

* When the HPP is waiting for a device symbol or a device number key to be input the prompt appears.

Sample operation 5
To input (write) the MC N2 M3 instruction

Sample display

<table>
<thead>
<tr>
<th>W</th>
<th>M 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 4</td>
</tr>
</tbody>
</table>

* The nesting level device symbol "N" is automatically displayed when the [MC] key is pressed. Press the [2], [SP], [M] and [3] keys and then press the [GO] key. Writing has now been completed.
3. PROGRAMMING

Modifying before confirming:

This section explains how to correct keyed-in data while entering an instruction.

**Basic operation**

(1) Correcting an instruction with an operand (before confirming)

**Sample operation 1**
To correct the instruction OUT T0, K10 to be OUT T0, D9 before confirmation with the [GO] key

**Sample display**

<table>
<thead>
<tr>
<th>W</th>
<th>7</th>
<th>OUT</th>
<th>T</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>K</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W</th>
<th>D</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(1) Enter an instruction, the first device and the second device.
(2) To cancel the second device press the [CLEAR] key once.
   * If the [CLEAR] key is pressed twice the modification is executed from step 7.
(3) Input the correct second device.
(4) Press the [GO] key. Writing of the instruction has now been completed.
(2) Correcting an instruction with an operand (after confirming)

Sample operation 2
To correct the instruction OUT T0, K10 to be OUT T0, D9 after confirmation with the [GO] key

Sample display

1. Enter an instruction, the first device, and the second device.
2. When the [GO] key is pressed, and the input (writing) of (1) has been completed, the line cursor moves to the following step.
3. Move the line cursor to the K10 position.
4. Enter the correct second device.
5. Press the [GO] key.
   Writing of the instruction has now been completed.

---

POINT

(1) Operand
An operand is a device used by an instruction.
Always make sure to input [SP] and the operand(s) when inputting an operand.
For example, in the case of the MOV instruction
[MOV D0 D1]
3. PROGRAMMING

Inputting an applied instruction

When inputting an applied instruction use the [FNC] key and the applied instruction number.

The applied instructions do not have an instruction symbol key. There are two ways to input instruction numbers:
(1) Inputting the number directly
(2) Searching the applied instruction menu with the HELP function.

Basic operation

(1) Instruction by inputting a number directly

* When the applied instruction number has been decided press the [FNC] key and then input the number directly
(See the instruction lists on pages 97 and 98).

Sample operation 1

To write the DMOVP

D0, D2 instruction

[Diagram showing the process of inputting the applied instruction number and device number]
3. PROGRAMMING

Sample display

1. Press the [FNC] key.
2. When designating 32-bit (D: double) instructions, before or after the applied instruction number is input, press the [D] key.
3. Input the applied instruction number.
4. To designate a P (pulse) instruction press the [P] key after entering the applied instruction number.
5. To enter a device input [SP], a device symbol and a device number.
   * The instruction symbol is also displayed on the screen along with the instruction number.
   Writing of the instruction has now been completed.

POINT

1. Designation of D and P prefix/suffix:
   D and P can be input in the order displayed (Sample operation 1) or input by keying in after the instruction number.
   D and P themselves can be input in any order.
   If you are not sure whether the instruction in question is effective as a D and/or P instruction, press the [HELP] key to display the list of instructions while keying in the instructions. See the point on Page 39.
   FX0 has D instructions, but not P instructions.
2. An ASC instruction can be read and displayed in ASCII characters.
   However, an ASC instruction cannot be written by the HPP.
3. If an applied instruction is input which the PC model does not have an error occurs.
3. PROGRAMMING

(2) Inputting an instruction by referring to the instruction menu displayed with the HELP function
When an applied instruction number is not known, press the [FNC] key and press the [HELP] key.
There are 10 applied instruction classifications:
The screen displays the applied instruction classifications and then the instructions symbols menu.
Select the applied instruction number from these screen guides.

0  Program flow
1  Transfer and comparison
2  Four arithmetical operations and logical operation
3  Rotation and shift
4  Data processing
5  High-speed processing
6  Convenient instruction
7  External device FX I/O
8  External device FX SER
9  External device F2

Sample operation 2  D0 D2 instruction

Point

(1) Applied instructions are often made up of an instruction (FNC number) and several operands.
When the HPP is used, a number (1, 2, 3, 4, 5) is displayed to the left of each operand to indicate which operand is currently displayed on the screen.

Sample display

The first operand

The second operand
Sample display

(1) Press the [FNC] key.
(2) Press the [HELP] key to display the applied instruction classifications.
    Screens can be scrolled by using the cursor key.
(3) Select the required applied instruction group by its number.
    The screen changes to display the menu of instruction numbers and
    instructions corresponding to that selected group (The group number comes
    from the ten's digit of the instruction number).
    The instructions of [ ]0 to [ ]9 are displayed on three screens.
    Screens can be scrolled by using the cursor key.
    Only valid applied instructions are displayed on the list in accordance with the
    model selection.
(4) Select the instruction symbol by its unit's digit.
    An applied instruction number is designated with the key operations in steps
    (3) and (4).
(5) Follow steps (4) to (6) of sample operation 1.

---

POINT

(1) Contents of the applied instruction menu.

The dot(s) appearing between the instruction number and instruction symbol show the validity of the D/P prefix/suffix. The upper part indicates a valid D (double) instruction and the lower part indicates a valid P (pulse) instruction.
3. PROGRAMMING

Inputting a device
This section explains how to input a device.

Basic operation

![Diagram showing the process of inputting a device]

- Write function → Instruction (basic instruction) → Instruction number (applied instruction)
- Basic instruction
  - Device symbol → Device number
- SP
- Device input
- First device input
- Second device input
- GO

Device input Details

- Device input
  - K → Numeral
  - Designation of constants
  - For MC, MCR instructions
  - When the device symbol is Z or V
  - Z → V
  - Index modifier
Sample operation 1  K1X010Z, D1 instruction

(1) Following the writing procedures, input an instruction and an instruction number.
(2) Designate the necessary number of digits.
   K1 to K4 (16-bit instruction) and K1 to K8 (32-bit instruction) are valid for the
digit designation. K1 indicates 4 bits.
(3) Input the device symbol.
   For the MC and MCR instructions the nesting level device symbol "N" is
automatically displayed.
(4) Input a device number.
   An index modifier Z of V, can be appended to a device number.
Label inputting (P, I)

When using P (pointer) and I (interrupt pointer) in a sequence program as labels, input them in the same way as an instruction.

Basic operation

Sample operation 1

To write label number 3

Sample display

W 100 P 3

1 Press the [P] key or [P] and [I] keys and then input the label number. P and I used as labels are handled in the same manner as instructions.

2 Press the [GO] key to write the keyed in pointer or interrupt pointer.
3. PROGRAMMING

Inputting numerical values
When writing a program, many different numbers are used (step numbers, device numbers, pointer numbers, and applied instruction numbers). These numbers may contain up to four digits. A keyed-in digit is displayed at the far-right column and the previous input is shifted to the left.

Sample operation 1

To input 1, 2, 3, 4, and 5 to the four-digit display area

* Each time a number key is pressed, the keyed-in number is displayed in the far-right column and the previously keyed-in numbers are shifted to the left. Therefore, when a number with more than the allowed number of digits is keyed in, the excess digits disappear from the screen beginning with the first keyed-in digit.

If only three or less digits are keyed-in for X and Y (which accept up to four digits) the other digit places are filled with leading zeros.

* Only the number displayed on the screen is written to memory.

Therefore, pay attention to the screen display when inputting.
3. PROGRAMMING

Writing of instructions and pointers

This section explains how to edit a program by overwriting an instruction.

Basic operation

To overwrite the OUT instruction T50 K123 to step 100

Sample display

W 100 OUT T 50
101 LDI X 013

W 100 OUT T 50
K 123

(1) Read the program from step number 100 and select [WR] function. Enter the instruction, a device symbol and a device number.
(2) To write the second device press the [SP] key and then enter a device symbol and a device number.
(3) * The set value of timers and counters can be changed using the monitoring function (See page 74).
3. PROGRAMMING

Overwriting Existing Instructions

(3) Press the [GO] key to write the new instruction.
* To overwrite instructions in steps near the currently displayed step number, move the line cursor to the required step number.

POINT
(1) Processing step numbers when overwriting
If the number of steps used by the instructions before and after overwriting differ from each other, the ensuring instructions are shifted as illustrated.
When overwriting is made over an NOP instruction, the NOP is deleted and the new instruction is inserted.

When overwriting OUT T0 K100 to step 11

<table>
<thead>
<tr>
<th>11</th>
<th>OUT</th>
<th>Y001</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>LD</td>
<td>X010</td>
</tr>
<tr>
<td>13</td>
<td>AND</td>
<td>M100</td>
</tr>
</tbody>
</table>

When overwriting OUT T0 K100 to step 20

<table>
<thead>
<tr>
<th>20</th>
<th>NOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>NOP</td>
</tr>
<tr>
<td>22</td>
<td>LD</td>
</tr>
<tr>
<td>23</td>
<td>ANI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20</th>
<th>OUT</th>
<th>T0</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>LD</td>
<td>X010</td>
</tr>
</tbody>
</table>

45
3. PROGRAMMING

Device writing
This section explains how to edit an instruction operand.

Basic operation

PC STATE: RUN STOP VALID MEMORY: RAM EEPROM EPROM

Sample operation 1 To change the operand of the MOVP instruction at step 100 from K2X1 to K1X0

Sample display

W 100 MOVP 12  
►1 K 1 X 0  

(1) Read the program from step number 100 and select the [WR] function. And then, move the line cursor to the position of the operand to be changed.
(2) When designating a digit, press the [K] key, and input the numerical value.
(3) Input the device symbol and device number and press the [GO] key. Then the designated device is altered.

* Lines that can be altered in this manner are lines without a step number. (When changing a line with a step number follow the instruction writing operation.)
3. PROGRAMMING

POINT

(1) K-number of grouped bit devices:
   K1 to K8 can be designated. Multiples of 4 bits are allocated. Therefore, K2 allocates 8 bits.

(2) Z and V devices:
   Z and V devices represent an index register. The index register is used as a modifier to the device number.
3. PROGRAMMING

NOP batch writing
This section explains how to perform NOP batch write operation to a specific range.

Basic operation

Sample operation 1 To batch write NOP from step 114 to step 124

Sample display
(1) Move the line cursor to the start step where NOP is written after pressing the [RD] and [WR] keys.
Writing cannot begin on lines without step numbers.
(2) Input [NOP], [K] and the end step number.
(3) NOP is batch written to the designated range by pressing the [GO] key.
   If the designated end step number is in the middle of an instruction the whole of this instruction is cleared with NOP's.

IMPORTANT
Writing of each instruction is valid for RAM/EEPROM. Follow the procedure for transferring to EEPROM after writing to the RAM built in the PC. However, when FX0 is used, each instruction is written directly to the EEPROM built in the PC.
3. PROGRAMMING

Sample operation 2
To write NOP instructions throughout the entire program memory

Sample display

![Diagram showing the sequence of steps]

1. Press the [RD], [WR], [NOP] and [A] keys.
   * In this case the position of the line cursor is unrelated to the writing range.

2. Press the [GO] key. The "ALL-CLEAR?" confirmation message is displayed.

3. To respond to the "ALL CLEAR?" message, press the [GO] key. The line cursor moves to step number 0 and the entire program is cleared (i.e., NOP instruction is written to every program step).

IMPORTANT

When a NOP instruction is written to every program step, all the parameters values are reset to the default settings. Latch elements are also reset. Therefore, the comment allocation will be 0 blocks, file register allocation will be 0 blocks, and memory capacity will be 2K steps (in the offline mode or online mode when there is not any memory cassette installed). If a memory cassette is installed, the memory capacity is defaulted to the maximum memory cassette capacity.

Page 109 gives details about parameter default values.

POINT

1. Devices that can be latched in the FX series (For details, see page 109):
   - M (internal relay)
   - C (16-bit counter)
   - D (data register)
   - S (state)
   - C (32-bit counter)
   - D (file registers)
   - T (timer)
This section explains how to insert an instruction at a designated position in a program.

Basic operation

Sample operation 1 To insert the AND M5 instruction before step 200

Sample display

1 ► 200 AND T 11
2 0 1. OR C 2 5

1 ► 200 AND M 5
2 0 0 AND T 11

(1) Read the program and press the [INS] key. Then insert the required instruction at the step designated by the line cursor. Lines that do not display a step number cannot be selected for the insert function.

(2) Key in the instruction, a device symbol and a device number or pointer symbol and pointer number.
3. PROGRAMMING

(3) Press the [GO] key to complete the insertion. Step numbers of the ensuing instructions are reassigned automatically.

* To insert instructions in steps near the currently displayed step number move the line cursor to the required step number.

POINT

(1) Insertion is not possible when the current program has filled the whole memory area. If insertion is attempted when the memory area is full the "STEP OVERFLOW" error message will be displayed.
Deleting an instruction and a pointer
This section explains how to delete an instruction designated by the cursor in one step.

Basic operation

```
[Program read] → INS → DEL → GO
```

Delete function

<table>
<thead>
<tr>
<th>PC STATE:</th>
<th>RUN</th>
<th>STOP</th>
<th>VALID MEMORY:</th>
<th>RAM</th>
<th>EEPROM</th>
<th>EPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample operation 1
To delete the AND M10 instruction at step 100

Sample display

```
D> 100 AND M10
101 OUT Y 001
```

```
D> 100 OUT Y 001
101 OUT Y 002
```

(1) Read the program and then press the [INS] and [DEL] keys.

(2) The instruction pointed to by the line cursor is deleted by pressing the [GO] key. The subsequent step numbers are reassigned automatically.
* To delete instructions in steps that are near the currently displayed step number, move the line cursor to the required step number.
* The instruction to be deleted should be designed with the line cursor. However, the delete function cannot be used for a line that does not have a step number displayed.
NOP batch deletion
This function allows all the NOP’s between program instructions to be removed simultaneously.
This section explains how to batch delete all NOP instructions from a program.

Basic operation

Sample To batch delete NOP operation 1

(1) Press the [INS], [DEL] and [NOP] keys and then press the [GO] key. The screen shown on the left is displayed. All NOP instructions are deleted from the program.

* After all NOP instruction steps have been deleted the step numbers are reassigned automatically.

* After the completion of batch-delete the first line of the program (step number 0) is displayed on the first line of the screen.
3. PROGRAMMING

Range deletion
This section explains how to batch-delete program steps within a specified range.

Basic operation

![Diagram of deletion process]

**Sample operation 1**
To delete program steps from step number 10 to step number 40

**Sample display**

1. Press the [INS], [DEL] and [STEP] keys and then input the start step number of the deletion range.
   If the entered step number lies inside an instruction the beginning of the instructions becomes the delete start step.

2. Press the [SP] key.
   The range is displayed on the screen.

3. Press the [STEP] key and then input the end step number of the deletion range.
   If the entered step number lies inside an instruction the last step number of the instructions becomes the final deleted step.
(4) Press the [GO] key to delete the program steps in the designated range. The step numbers of the ensuing instructions are reassigned automatically. The program is displayed with the deletion start step number as the first line.

POINT

(1) How to cancel this setting
Press the [CLEAR] key before pressing the [GO] key. The first time the [CLEAR] key is pressed, the end step number is cleared. The second time the [CLEAR] key is pressed, the start step number is cleared.
The monitoring function is used to display the operating and control states of PC on the HPP screen.

**General function description**

Monitoring consists of the following functions:

- **Device monitoring**
  - ON/OFF monitoring of a designated device, display of the set value and the current value of T, C, D, and a file registers.

- **Continuity check**
  - The program is read to monitor contact continuity and coil operations.

- **Active state monitoring**
  - Transfer of state elements can be checked by automatically displaying active states in priority from low to high state numbers.
Outline procedure

This section shows an outline of the monitoring procedure.

Connect the HPP and PC. (See page 6.)

Turn ON the power supply to the PC.  
Reset the HPP. ([RST] + [GO]) (See page 89.)

The mode selection screen is displayed.

Monitoring

Monitoring operations
4. MONITORING

This section explains how to monitor the ON/OFF state and setting and current values for a designated device.

Basic operation

Sample operation 1
To monitor X0 and succeeding devices in order.
[Allowed devices: X, Y, M and S]

Sample display

<table>
<thead>
<tr>
<th>M</th>
<th>X</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>X</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&gt;</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

(1) After pressing the [MNT] key, press the [SP] key and enter the device symbol and device number.
(2) Press the [GO] key. The ON/OFF state of the designated device is displayed on the screen using the ■ symbol (the ■ symbol indicates the ON state.)
(3) Use the [↑] and [↓] keys to monitor the ON/OFF state of preceding and succeeding devices.
The screen can display the ON/OFF state for up to 4 devices. Different devices can be monitored on one screen.
4. MONITORING

Sample operation 2
To monitor D25 and succeeding devices in order.
[Allowed devices: D, Z, and V (16 bits)]

Sample display

* The key operation is the same as for steps (1) to (3) of sample operation 1.
The current value is monitored on the display screen.
A maximum of two devices can be monitored on one screen.

Sample operation 3
To monitor D19 and D20 and succeeding devices as a pair.
[Allowed devices: D, Z (32 bits)]

Sample display

* Before inputting the device press the D key to designate 32 bits monitoring.
The other key operations are the same as for (1) to (3) of sample operation 1.
* Because the monitored device is a 32-bit device, the designated device number and the following device number are monitored in pair. For example, D19 is monitored with D20 and D20 is monitored with D21.
* Note: To step on a full 32 bit register press a cursor key twice.
4. MONITORING

Sample operation 4 To monitor T10 and then C9.
[Allowed Devices: T, C (16 bits)]

Sample display (Set value = K)

* The key operation is the same as for steps (1) to (3) of sample operation 1. When the devices are T (timer) or C (counter) the current and set values are monitored.

The ON/OFF state of the output contact and the reset coils are also monitored and indicated with the ■ symbol.

Sample display (Set value = D)

* If the set value of T or C is designated indirectly with a data register (D) the value of the data register can be monitored by using the [ ↓ ] key.
4. MONITORING

Sample operation 5
To monitor C251
(32-bit counter).

Sample display (Set value = D)

* The key operation is the same as for steps (1) to (3) of sample operation 1. The current value and the set value are monitored as 32 bits.
* The display is similar to that in sample operation 4. In addition, the up/down mode of the counter is also monitored (□ indicates up).
* The display screen is scrolled by using the [↑]/[↓] keys.
* The monitored current value can be toggled between decimal display and hexadecimal display pressing the [HELP] key.

POINT

(1) Direct designation.
   Numerical values designated by using constant K (decimal) are dealt with as setting values.
   Only decimal values for K can be set.
   The allowable ranges are as follows.
   16 bit: -32768 to 32767
   32 bit: -2147483648 to 2147483647

(2) Indirect designation.
   The setting values is designated by a data register (D).
   For example, if D10 is designated and contents of D10 are 123, it is the same as setting K123.
   However, even if the contents of D are latched, if the battery voltage falls below the allowable limit, the data will be lost.
This section explains how to read a program by a step number or an instruction and monitor the contact continuity of a device and coil operation.

**Basic operation**

![Diagram of monitoring process]

**Sample operation 1**
To read step 127 and to monitor a contact continuity and coil operation.

**Sample display**

```
M: 127 OR IM 10
128 OUT MY 005
```

* Press the [MNT] key.
The following operation is the same as the read operation.

Reading by a step:
Press the [STEP] key and enter the required step number, then press the [GO] key.

Reading by an instruction:
Press the instruction key to be designated.
With an instruction which needs a device entering, designate the necessary conditions before pressing the [GO] key.
4. MONITORING

* Two lines of instructions are displayed from the designated step number. The contact continuity and coil operation states are displayed and monitored on the left side of the device by using the symbol.
* Line scrolling can be done in the monitoring state by using the [↑]/[↓] keys.
* If the program is displayed before monitoring, the contact continuity and coil operation are monitored for the current display state by pressing the [MNT] key.
This section explains how to monitor up to a maximum of six points of active states beginning with the lowest active state number when using the step ladder instruction.

**Basic operation**

```
MNT  →  STL  →  GO
```

<table>
<thead>
<tr>
<th>PC STATE:</th>
<th>RUN</th>
<th>STOP</th>
<th>VALID MEMORY:</th>
<th>RAM</th>
<th>EEPROM</th>
<th>EPROM</th>
</tr>
</thead>
</table>

**Sample operation 1** To monitor an active state.

**Sample display**

```
M ACTIVE STATE → GO
```

```
M S 30 S 40 S 234
S 300 S 301 S 302
```

* Press the [MNT] and [STL] keys and then press the [GO] key. A maximum of six currently operating states are monitored, beginning with the lowest number, on the screen.

* When a state element changes, the displayed state numbers change accordingly. And so, the machine's operating processes can be monitored.

* The states in the range from S0 to S899 can be monitored. The state of S900 and after (for the annunciator) will be ignored.

* When seven points or more are turned ON, "/>" will be displayed on the lower left side of the screen. The state of the seventh point onwards can be displayed by pressing the [↑]/[↓] keys.

* Active state monitoring is disabled unless special auxiliary relay M8047 is turned ON.

**POINT**

(1) When monitor M8094 of the annunciator is turned ON, the minimum active state number of S900 to S999 can be monitored by the monitoring operation given above.
The test function is used to forcibly turn devices ON/OFF and to change the constant value of data devices.

**General function description**

Test consists of the following functions:

- **Forcible ON/OFF**
  - A designated device can be turned ON/OFF forcibly by using the HPP. (Changing the contents of the image memory)

- **Changing the current values of T, C, D, Z, and V**
  - The current values of devices T, C, D, Z and V can be changed. (Changing the contents of the image memory)

- **Changing the set values of T and C**
  - The set values of devices T and C can be changed. (Changing the contents of the program memory)

- **Writing of a file register**
  - Numerical values can be written to file registers in the program memory (FX only).

**POINT**

1. *1 When the PC is in the RUN state this function is valid for RAM memory.

   When the PC is in the STOP state this function is valid for RAM and EEPROM (with the memory protect switch is OFF). EEPROM is invalid.

   However, with FXo, a change to the set values of T and C is valid when the PC is in the RUN state (program memory: EEPROM).
Outline procedure

This section shows an outline of the test procedure.

Connect the HPP and PC. → → → → → → → → → → → → (See page 6.)

Turn ON the power supply to the PC.  
Reset the HPP. ([RST] + [GO]) → → → → → → (See page 89.)

The mode selection screen is displayed.

Test

Test operation
This section explains how to forcibly turn a device ON/OFF. After device monitoring, switch to the test function.

The forced ON/OFF operation is only valid for one operating cycle. Therefore, turning ON/OFF forcibly when the PC is in the RUN state will have an effect on timers, counters, set/reset circuits, latched circuits and auxiliary relays.

**Basic operation**

```
<table>
<thead>
<tr>
<th>Device monitoring</th>
<th>TEST</th>
<th>SET</th>
<th>RST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Forcibly ON

Forcibly OFF

**PC STATE:**

<table>
<thead>
<tr>
<th>RUN</th>
<th>STOP</th>
<th>VALID MEMORY:</th>
<th>RAM</th>
<th>EEFROW</th>
<th>EPROM</th>
</tr>
</thead>
</table>

**Sample operation 1** To turn device Y1 ON/OFF forcibly.

**Sample display**

```
T ▲ Y 0 0 1
```

(1) Monitor the device using the monitoring function.

(2) Press the [TEST] key.

If the monitored device is in the OFF state press the [SET] key to turn the device ON forcibly.

If the monitored device is in the ON state press the [RST] key to turn the device OFF forcibly.
5. TEST

Forcible ON/OFF

POINT

(1) Sample operation when the PC is in the STOP state.

Even if there is not a latch circuit, when the PC is in the STOP state, Y1 will remain in the ON state until the [RST] key is pressed again.
This section explains how to change the current values of devices T, C, D, Z, and V.

Switch to the test function after monitoring the device.

### Basic operation

![Diagram showing the test function and changing the current value]

### Sample operation 1

To change the current value of the 32-bit counter setting register (D1, D0) from K-12345 to K10.

![Sample display]

1. Monitor the device using the monitoring function.
2. Press the [TEST] and [SP] keys, then press the [K] key or [K] and [H] keys and input the new current value (Constant K is for decimal setting and H is for hexadecimal setting).
3. Press the [GO] key to change the current value to the new value.

* The display of the current value can be switched between a hexadecimal display and a decimal display by pressing the [HELP] key.
5. TEST

Changing of the Current Values of T, C, D, Z, and V

--- POINT ---

(1) Writing to a file register (FX only).

- Data writing to a file register is done in this mode.

- File register changes are possible with RAM memory (PC in the RUN state), or with RAM and EEPROM memory (PC in the STOP state).

- A maximum of 2000 file register points is allocated in the program memory (RAM, EEPROM, EPROM) in 500-point units by the parameter setting. (See page 85).

- A file register can be utilized by designating data registers D1000 to 2999.

- Independently of the PC state (RUN/STOP) and the program memory, the current value of T, C, D, Z and V (other than a file register) can be changed.
5. TEST

This section explains how to change the set value of devices T and C after device monitoring or a continuity check. Changing the set value is possible with RAM memory (PC in the RUN state), or with RAM and EEPROM memory (PC in the STOP state). With FX, it is possible when the PC is in the RUN state.

**Basic operation**

![Diagram](attachment:image.png)

<table>
<thead>
<tr>
<th>Test function</th>
<th>Function</th>
<th>Designation of a set value</th>
<th>New register number</th>
<th>GO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device monitoring</td>
<td>TEST</td>
<td>SP</td>
<td>SP</td>
<td>D</td>
</tr>
<tr>
<td>Continuity check</td>
<td>1</td>
<td>↓</td>
<td>TEST</td>
<td>K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PC STATE:</th>
<th>RUN</th>
<th>STOP</th>
<th>VALID MEMORY:</th>
<th>RAM</th>
<th>EEPROM</th>
<th>EPROM</th>
</tr>
</thead>
</table>

**Sample operation 1**

To change T5's set value from K300 to K500.

1. T5 device monitoring
2. Press the [TEST] key and then press the [SP] key once. The prompt appears at the position where the current value is displayed.

**Sample display**

```
M ➤ T 5 K 200
P R K 300

T ➤ T 5 K 200
P R K 300
```

1. Monitor the device using the monitoring function.

2. Press the [TEST] key and then press the [SP] key once. The prompt appears at the position where the current value is displayed.
5. TEST

Changing the Set Values of T and C

(3) The prompt can be moved to the position where the set value is displayed by pressing the [SP] key again.

(4) Input the new set value and then press the [GO] key. Changing the set value is now complete.

* The changed set value becomes valid for the OUT T or C instruction that appears first in the program.

If there are many out coils for the same device changing the set value for a specific OUT T or C instruction should be made by using the continuity check operation (See page 74).

---

POINT

(1) Changing a constant when the PC is in the RUN state.

Even if the program memory is EEPROM, including FX0, a constant (set value of a timer and a counter and the current value of a data register) can be changed when the PC is in the RUN state.

However, when the constant is changed the PC scan time increases 20 to 30 msec and the response to the input interrupts of I00[ ] to I30[ ] is delayed by 20 to 30 msec.
Sample operation 2: To change T100's set value from D123 to D120

Sample display

(1) Monitor the device using the monitoring function.
(2) The prompt can be moved to the position for the data register number where the set value is stored by pressing the [SP] key twice.
Input the new data register number.
(3) Press the [GO] key to change the setting value register to the new register number.

To change D123's contents from K300 to K400

Sample display

* When changing the contents of D follow the operating procedure for changing the current value of D (See page 69).
Sample operation 3
To change the set value K1234 of the OUT T50 instruction at step 251 to K123

Sample display

<table>
<thead>
<tr>
<th>M 251 OUT T 50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>K 1 2 3 4</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>T K 1 2 3</td>
</tr>
<tr>
<td>2 5 4 LD I M 3 5</td>
</tr>
</tbody>
</table>

(1) Execute the continuity check for required device using the monitoring function.
(2) Move the line cursor to the line with the set value.
(3) Press the [TEST] key and input a new set value. Then press the [GO] key to change present setting to the new set value.

* With a timer and a counter the change can be made with the monitoring function without needing to switch to the write function.

* The line cursor can be moved after pressing the [TEST] key to switching to the test function.
6. OTHER FUNCTIONS

The mode menu can be displayed by pressing the [OTHER] key during programming.

The mode menu consists of a program check, memory cassette transfer, parameter, device conversion, latch clear and buzzer control.

Outline procedure

The outline procedure for the mode menu is shown below.

1. Program check
2. Memory cassette transfer
3. Parameter
4. Device conversion
5. Latch clear
6. Buzzer control

Operation of each function

Connect the HPP and PC.

Turn ON the power supply of the PC.

Reset the HPP. [RST] + [GO] ... (See page 89.)

The mode selection screen is displayed.

OTHER

(See page 6.)
6. OTHER FUNCTIONS

This section explains how to select each menu item from the mode menu.

Basic operation

Mode menu display

- The menu screen is displayed by pressing the [OTHER] key.
  Press the target item number or move the cursor to the target item by using the [↑]/[↓] keys and then press the [GO] key.
  Then, the selected menu item is displayed.

- The screen can be scrolled by using the cursor keys.

POINT

(1) Transferring between Function Screen and Mode Menu Screen
  Press the [OTHER] key during a function screen operation and the mode menu screen will be displayed.
  Conversely, pressing a function key from a screen accessed by the [OTHER] key permits the entry to that function operation.
6. OTHER FUNCTIONS

This section explains how to check the PC's internal program.

Basic operation

<table>
<thead>
<tr>
<th>Mode menu display</th>
<th>1</th>
</tr>
</thead>
</table>

PC STATE: RUN STOP VALID MEMORY: RAM EEPROM EPROM

Display

1. PROGRAM CHECK EXECUTING

* Display the menu screen and then press the [1] key. A program check is executed and the check result is displayed.

<When there is not an error>

1. PROGRAM CHECK NO ERROR

* If there are no errors the screen shown on the left is displayed.

<When there is an error>

1. PROGRAM CHECK

* If there is an error the program step number with the error, the error message and error code are displayed. For each operation, only the first error is displayed. The error code is displayed by pressing the [↓] key.

* Press the [CLEAR] key or the [OTHER] key to return to the mode menu.

* Clearing any errors found will automatically reset special auxiliary relay M8068 and special data register D8068. If other errors remain the next error code is stored.
6. OTHER FUNCTIONS

This section explains how to transfer the program and parameters between the RAM built in the PC and a memory cassette installed in the PC.

Menu item displays for transfers between the RAM built in the PC and a memory cassette will differ depending on the type (RAM, EPROM, EEPROM) of memory cassette (invalid for FX0).

Basic operation

![Diagram of basic operation]

**Memory cassette transfer screen**

1. **DATA TRANSFER**
   - FXRAM → CSRAM (1)
   - FXRAM → CSRAM (2)
   - FXRAM : CSRAM (3)

2. **DATA TRANSFER**
   - FXRAM → EPROM (1)
   - FXRAM : EPROM (2)

(1) Transfer from the RAM built in the PC to a CSRAM cassette.
(2) Transfer from a CSRAM cassette to the RAM built in the PC.
(3) Verification of the RAM built in the PC and a CSRAM cassette.

**POINTS**

1. **CSRAM**
   - CSRAM is the RAM cassette memory installed in the PC.
   - This name is used to distinguish it from the RAM (FXRAM) built in the PC.

2. **Precautions when transferring from the FXRAM to an EEPROM**
   - When data is transferred from the FXRAM to an EEPROM cassette, set the EEPROM cassette memory protect switch to OFF.
6. OTHER FUNCTIONS

Sample operation 1
To transfer a program and parameters from the RAM built in the PC to a CSRAM cassette.

Sample display

2. DATA TRANSFER
   ✓ FXRAM → CSRAM
   ← FXRAM → CSRAM
   OK → [GO] NO → [CLR]
   EXECUTING

* Display the menu screen and then press the [2] key. The memory cassette transfer screen is displayed.

(1) Use the cursor key to select the item to be transferred and then press the [GO] key.

(2) To transfer press the [GO] key. To cancel the transfer press the [CLEAR] key. The screen returns to the memory cassette transfer screen.

* When the transfer has been completed, the "COMPLETED" message will be displayed.
   If the transfer is not performed normally, an error message will be displayed.

---

POINTS

(1) Program memory capacity.
   If a program whose memory capacity parameter is set to 4K or 8K steps is transferred from the memory cassette to the RAM built in the PC, the "PC PARA. ERROR" error message will be displayed and the program will not be transferred.
   Set the memory capacity to 2K steps by parameter setting (See page 81).

(2) When a parameter is mismatched.
   When verification is selected, if the execution result contains a mismatch, the "VERIFY ERROR" message will be displayed, as well as the mismatched location.
6. OTHER FUNCTIONS

Effective Usage of memory cassettes

<Considerations when using EEPROM>
If an EEPROM cassette is installed in the FX and programming is done, writing can be done to the EEPROM. Since the writing time to an EEPROM is longer than that to the RAM, it takes much longer to insert and delete programs.
More importantly, you can only write to the EEPROM 10,000 times.
Therefore, if the program is less than 2K steps, it is better to transfer it to an EEPROM cassette after writing it using the RAM built in the PC.

<Masters and copies>
If a program which was transferred to the EEPROM cassette is kept as a master, that program can be (a) transferred to a PC and used as is, or (b) some parts of the program can be changed after by transferring the program from this master to another PC.
Parameters are the values that specify the various functions and ranges of a device. Parameters are stored independently in the parameter area of the PC’s memory. Setting of default values, memory capacity, entry code registration, latch ranges and file registration of PC program memory are all done by parameter setting.

**Basic operation**

```
[ Mode menu display ] → 3 → [ Default value setting ] → [ Memory capacity setting ] → [ Entry code registration ] → [ Latch range setting ] → [ File register setting ]
```

<table>
<thead>
<tr>
<th>PC STATE:</th>
<th>RUN</th>
<th>STOP</th>
<th>VALID MEMORY:</th>
<th>RAM</th>
<th>EEPROM</th>
<th>EPROM</th>
</tr>
</thead>
</table>

**Operation procedure**

- Select an item and then press the [GO] key to move to the next setting item. When not changing the current set value, press the [GO] key.

- Press the [CLEAR] key to return to the previous setting.

- Press the [OTHER] key to return to the mode menu during parameter setting.
POINT

(1) Changing parameter settings
  Parameter settings can be changed during programming.
  Programming can begin when the parameters are the default values and then altered later.
6. OTHER FUNCTIONS

Default value setting

Display

* Display the menu screen and then press the [3] key.
The parameter setting screen is displayed.
When setting the default values (initial values), select "YES" by using the cursor key
and then press the [GO] key (The default option is "NO").
When not setting the default values, press the [GO] key.
Page 109 gives details about parameter default values.

Memory capacity setting

Display

* The memory capacity default value is set to 2K steps.
When changing the memory capacity, select the steps to be changed by using the
cursor key and then press the [GO] key.

* Types of PCs and memory capacities that can be set

<table>
<thead>
<tr>
<th>Memory capacities</th>
<th>2K steps</th>
<th>4K steps</th>
<th>8K steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FX0</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FX</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

o : Valid
x : Not valid
6. OTHER FUNCTIONS

**Entry code registration**

Display

| ENTRY CODE | ENTER | DELETE |

* When registering an entry code, select "ENTER" by using the cursor key, input the entry code and then press the [GO] key (The default is set to "Not Registered").

* When not changing the entry code press the [GO] key.

* When deleting an entry code, select "DELETE" with the cursor key, input that entry code and then press the [GO] key.

* Page 90 gives details about entry code registration.

**Latch range setting**

Display

- LATCH RANGE M 500 - M 1023
  - LATCH RANGE S 500 - S 999
  - LATCH RANGE (16) C 100 - C 199
  - LATCH RANGE (32) C 200 - C 255
  - LATCH RANGE D 200 - D 511

  2times GO

* When changing the latch range, input the device number and then press the [GO] key. Change the settings in order; the start device then the end device. When not changing a latch range, press the [GO] key twice.

* The latch range end device can be set to a value that is below the default value (See page 109).

* When using FX, select the default values.

**POINT**

(1) Latch

Latch is the function which maintains the state of a device when the power supply is turned OFF and it keeps the device from being cleared when the power is turned back ON. Therefore, the latch function will store the state just before a power failure and restart operations from that state when the power goes ON again.
6. OTHER FUNCTIONS

**File register setting (FX)**

**Display**

- **FILE REGISTER**
  - **0 BLOCK (500/B)**
  - **GO**

* When setting a file register, input the number of blocks (0 to 4 blocks) of memory to be allocated and then press the [GO] key.
  The maximum range that can be set is four blocks.

* Since the FX0 does not have the file register function, set 0 blocks.
  Pages 105, 108, and 109 give details.

**Parameter setting completed**

**Display**

- **PARAMETER VALUES**
  - **COMPLETE? [YES, NO]**
  - **GO**

* To end parameter setting press the [GO] key.
  The mode menu will be displayed.
  The default option is "YES".
  To continue parameter setting, select "NO" with the cursor key and then press the [GO] key.
  The screen returns to the initial parameter setting screen.
6. OTHER FUNCTIONS

This section explains how to change a device number to a new device number in the entire program memory (even instructions written after the end instruction will be converted).

Basic operation

Sample operation 1
To convert device X0 to device X3

Sample display

* Display the menu screen, and then press the [4] key. The device conversion screen is displayed.

(1) Input the source device.
Press the [X] and [0] keys and then press the [GO] key.

(2) Input the target device.
Press the [X] and [3] keys and then press the [GO] key.

* When canceling the target or source device press the [CLEAR] key. However, do so before confirming the target device (before pressing the [GO] key).
6. OTHER FUNCTIONS

This section explains how to clear the device latch state.

Basic operation

<table>
<thead>
<tr>
<th>Mode menu display</th>
<th>5</th>
<th>↑</th>
<th>↓</th>
<th>GO</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC STATE:</td>
<td>RUN</td>
<td>STOP</td>
<td>VALID MEMORY:</td>
<td>RAM</td>
</tr>
</tbody>
</table>

Display

| 5. LATCH CLEAR | S | C (16-BIT) | C (32-BIT) | D | D (FILE) |

* Display the menu screen and then press the [5] key. The latch clear screen is displayed. The devices that can be latched are displayed on the screen.

* Line scrolling can be done with the cursor keys.

* Select the device for which the latched state is to be cleared by using the cursor key and then press the [GO] key. Clearing has now been executed. The latch clear operation needs to be done for each device.

* The latch state of all devices except for the file register can be cleared even if the program memory is in RAM, EEPROM, or EPROM.

* If the program memory is in EPROM the file register cannot be cleared.

If the program memory is in EEPROM set the memory protect switch to OFF to enable clearing of the file register.

---

POINT

(1) File registers and EPROM

If the program memory is in EPROM, the file registers cannot be cleared because file registers are allocated to normal program memory.
6. OTHER FUNCTIONS

This section explains how to control the HPP buzzer.

Basic operations

<table>
<thead>
<tr>
<th>Mode menu display</th>
<th>6</th>
<th>↑</th>
<th>↓</th>
<th>GO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PC STATE:</th>
<th>RUN</th>
<th>STOP</th>
<th>VALID MEMORY:</th>
<th>RAM</th>
<th>EEPROM</th>
<th>EPROM</th>
</tr>
</thead>
</table>

Display

| 6. BUZZER | ON | OFF |

* Display the menu screen and then press the [6] key. The buzzer control screen is displayed.

* Select ON or OFF using the cursor keys and then press the [GO] key. The display returns to the menu screen and the buzzer status is adjusted.
The procedure for starting up the system is shown below.

The PC will be put into the STOP state when the power to the PC is turned on if the [RD/WR] key is held down, even if the PC RUN input is set to the ON position.

The PC will be restored to the RUN state when the power is turned ON again.

The HPP is reset by pressing the [RST] + [GO] keys and the screen is switched to the initial power ON screen.

Mode setting screen

- **SELECT FUNCTION**
  - FX2
  - [2K FXRAM]

  - Type of memory: FXRAM, CSRAM, EPROM, EEPROM
  - Memory capacity (parameter set values): 2K, 4K, 8K
  - PC series name

---

**Connect the HPP and PC, and turn ON the power supply to the PC.**

**COPYRIGHT(C) 1992 MITSUBISHI**

**MELSEC FX-10P-E V1.00**

- An entry code is registered.
  - To entry code handling (see page 90)
- An entry code is not registered.
  - To the parameter error processing (see page 94)
- Parameter error
  - (see page 93)
This section explains how to register entry codes. Input the correct entry code enables all HPP operations. If the input entry code does not match, the functions of the HPP are limited according to the following table.

If the entry code for a PC is unknown operations can be stored by keying in the special entry code (8 [SP] keys). In this case, however, all of the instructions in the PC will be cleared (see page 92).

The entry code is composed of eight hexadecimal digits. The protection level of the program is set by the first of these eight digits.

The program protection level consists of (A) the all-operations prohibition level, (B) the theft prevention level and (C) the miswriting prevention level.

### [Valid HPP functions for protection levels]

<table>
<thead>
<tr>
<th>Function</th>
<th>Protection Levels</th>
<th>All-Operations Prohibition</th>
<th>Theft Prevention</th>
<th>Miswriting Prevention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>Read</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Write</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Insert</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Device monitoring</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Continuity check</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Active state monitoring</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Test</td>
<td>Forcible ON/OFF</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Current value change</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Set value change</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Mode menu</td>
<td>Parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Program check</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Device conversion</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Transfer</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Latch clear</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

All-operations prohibition: A [ ] [ ] [ ] [ ] [ ] [ ] [ ]

Theft prevention: B [ ] [ ] [ ] [ ] [ ] [ ] [ ]

Miswriting prevention: C [ ] [ ] [ ] [ ] [ ] [ ] [ ]

☆ If the most significant digit is not A, B, or C, the all-operations prohibition level is set.

O: Available
X: Unavailable
This section explains how to register an entry code.

To the start-up procedure (See page 89.)

*1 All-operations prohibition level [A]
*2 Theft prevention level [B]
*3 Miswriting prevention level [C]
Special entry code input]

If the entry code is unknown operations can be started by inputting the special entry code.

The procedure for inputting the special entry code is shown below.

---

*1 If "YES" is selected, the program and all parameters (which include the entry code) are cleared.

To the start-up procedure (See page 89).
This section explains how to start up all functions.

The PC model is determined and the parameter set values of the PC program memory capacity are displayed. (If an FX0 PC is selected, FX is displayed.)

1. PROGRAM CHECK
2. DATA TRANSFER
3. PARAMETERS
4. XYM. NO.CONV.
5. LATCH CLEAR
6. BUZZER
The section explains how to take corrective action if a PC parameter error occurs when the system is started up.

If a PC is used which has been left for a long time with the battery removed, a parameter error occurs.

*1 An FX0 cannot be set to 4K or 8K steps.

To the start-up procedure (See page 89).
POINTS

(1) File register setting
   From 0 to 2000 steps of program memory can be used for file registers by setting 0 to 4 block (One file register point corresponds to one step).
   When an FX0 is used no file registers can be set.

(2) Comment area setting
   From 0 to 4000 steps of program memory can be used for the comment area by setting 0 to 8 blocks
   (The fifteen characters of one comment are equivalent to ten steps).
   When an FX0 is used no comment area can be set (See page 110).
### Basic instructions and step ladder instructions

<table>
<thead>
<tr>
<th>Group</th>
<th>Instructions</th>
<th>Devices</th>
<th>Number of Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>LD</td>
<td>X, Y, M, S, T, C, Special M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>LDI</td>
<td>X, Y, M, S, T, C, Special M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>AND</td>
<td>X, Y, M, S, T, C, Special M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ANI</td>
<td>X, Y, M, S, T, C, Special M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>OR</td>
<td>X, Y, M, S, T, C, Special M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ORI</td>
<td>X, Y, M, S, T, C, Special M</td>
<td>1</td>
</tr>
<tr>
<td>Joint</td>
<td>ANB</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ORB</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MRD</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MPP</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td>Other</td>
<td>MC</td>
<td>N, Y, M</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>MCR</td>
<td>N (nesting)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>NOP</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>END</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td>Step ladder</td>
<td>STL</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>RET</td>
<td>None</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>Instructions</th>
<th>Devices</th>
<th>Number of Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OUT</td>
<td>Y, M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special M</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T-K, D</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C-K, D (16-BIT)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C-K, D (32-BIT)</td>
<td>5</td>
</tr>
<tr>
<td>Output</td>
<td>SET</td>
<td>Y, M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special M</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>RST</td>
<td>Y, M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special M</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T, C</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D, V, Z, Special D</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>PLS</td>
<td>Y, M</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>PLF</td>
<td>Y, M</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>Instructions</th>
<th>Pointer Number</th>
<th>Number of Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>P</td>
<td>0 to 63</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0[ ] to 8[ ]</td>
<td>1</td>
</tr>
</tbody>
</table>
8. APPENDICES

**Applied instructions**
(The O symbol indicates an available instruction.)

The FX0 series PCs do not have pulse execution-type application instructions.

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FX0</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program flow</td>
<td>00</td>
<td>CJ</td>
<td>Conditional jump</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>01</td>
<td>CALL</td>
<td>Subroutine call</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>SRET</td>
<td>Subroutine return</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>IRET</td>
<td>Interruption return</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>EI</td>
<td>Interruption permission</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>DI</td>
<td>Interrupt disable</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>FEND</td>
<td>Main program end</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>07</td>
<td>WDT</td>
<td>Watchdog timer</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>08</td>
<td>FOR</td>
<td>Repeat range start</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>09</td>
<td>NEXT</td>
<td>Repeat range completion</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FX0</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algebraic and logical operations</td>
<td>20</td>
<td>ADD</td>
<td>Addition (binary) (S1) + (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>SUB</td>
<td>Subtraction (binary) (S1) − (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>MUL</td>
<td>Multiplication (binary) (S1) × (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>DIV</td>
<td>Division (binary) (S1) / (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>INC</td>
<td>Increment (binary) (D) + 1 → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>DEC</td>
<td>Decrement (binary) (D) − 1 → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>WAND</td>
<td>Logical AND (S1) ∧ (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>WOR</td>
<td>Logical OR (S1) ∨ (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>WXOR</td>
<td>Exclusive Logical OR (S1) ⊕ (S2) → (D)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>NEG</td>
<td>Negation (D) + 1 → (D)</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FX0</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation and shift</td>
<td>30</td>
<td>ROR</td>
<td>Rotation (right)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>ROL</td>
<td>Rotation (left)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>RCR</td>
<td>Rotation (right) with CY</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>RCL</td>
<td>Rotation (left) with CY</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>SFTTR</td>
<td>Right shift bit</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>SFTL</td>
<td>Left shift bit</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>WSFR</td>
<td>Right shift word</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>WSFL</td>
<td>Left shift word</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>38</td>
<td>SFWR</td>
<td>Shift register write</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>SFRD</td>
<td>Shift register read</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

* Number of steps of an applied instruction

Instruction... One step is composed of the sum of [FNC] and the succeeding instruction number.

Operand...... Sum of the instruction number and the succeeding devices—

16-bit instruction : Two steps per operand
32-bit instruction : Four steps per operand
### 8. APPENDICES

**List of Instructions**

**(The O symbol indicates an available instruction).**

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FXg</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data processing</td>
<td>40</td>
<td>ZFRST</td>
<td>Batch reset</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>DECO</td>
<td>Decode</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>ENCO</td>
<td>Encode</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>SUM</td>
<td>Bit check (number of <em>1</em> bits)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>BON</td>
<td>Bit ON/OFF judgment</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>MEAN</td>
<td>Mean</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>ANS</td>
<td>Annunciator set</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>ANR</td>
<td>Annunciator reset</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>REF</td>
<td>I/O refresh</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>REFF</td>
<td>Filter adjustment</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>MTR</td>
<td>Matrix input</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>HSICS</td>
<td>Comparison set (high-speed counter)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>HSCR</td>
<td>Comparison reset (high-speed counter)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>HSIZ</td>
<td>Zone comparison (high-speed counter)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>SPD</td>
<td>Pulse density</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>57</td>
<td>PSLY</td>
<td>Pulse output</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>58</td>
<td>PWM</td>
<td>Pulse-width modulation</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>59</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-speed processing</td>
<td>60</td>
<td>IST</td>
<td>Initial state (step ladder)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>ABSD</td>
<td>Drum sequence (absolute value)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>63</td>
<td>INCD</td>
<td>Drum sequence (incremental value)</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>TIMR</td>
<td>Teaching timer</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>STMR</td>
<td>OFF delay, one shot, and flicker timer</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>ALT</td>
<td>Alternation output</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>RAMP</td>
<td>Ramp signal</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>ROTC</td>
<td>Rotary table control</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Handy instructions</td>
<td>69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FXg</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>External device I/O</td>
<td>70</td>
<td>TKY</td>
<td>10-key pad input</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>HKY</td>
<td>16-key pad input</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>72</td>
<td>DSW</td>
<td>Digital switch time-sharing interruption</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>SEGD</td>
<td>7-Segment decode</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>74</td>
<td>SEGL</td>
<td>7-Segment time-sharing display</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>ARWS</td>
<td>Arrow switch control</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>76</td>
<td>ASC</td>
<td>ASCII conversion</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>PR</td>
<td>ASCII code print</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>FROM</td>
<td>Reading from buffer</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>TO</td>
<td>Writing to buffer</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FXg</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>External device SER</td>
<td>80</td>
<td>PRUN</td>
<td>For FX-40AP/AW</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>83</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>84</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>VRRD</td>
<td>FX-8AV read</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>VRSC</td>
<td>FX-8AV graduation read</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>88</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>FNC Numbers</th>
<th>Instruction Symbols</th>
<th>Instruction Name</th>
<th>FXg</th>
<th>FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>External devices F2</td>
<td>90</td>
<td>MNET</td>
<td>For F-16NP/NT</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>ANRD</td>
<td>FZ-6A read</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>92</td>
<td>ANWR</td>
<td>FZ-6A write</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>RMST</td>
<td>FZ-32RM start</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>94</td>
<td>RMWR</td>
<td>FZ-32RM write</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>RMRD</td>
<td>FZ-32RM read</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>RMNN</td>
<td>FZ-32RM monitor</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>BLK</td>
<td>FZ-30GM block designation</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>98</td>
<td>MCDE</td>
<td>FZ-30GM M code read</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 8. APPENDICES

#### Device Number list

**Common to FX series**

<table>
<thead>
<tr>
<th>Basic Units</th>
<th>Input Relay Numbers</th>
<th>Output Relay Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>FX0 - 30M</td>
<td>X0 to X17 16 points</td>
<td>Y0 to Y15 14 points</td>
</tr>
<tr>
<td>FX0 - 20M</td>
<td>X0 to X13 12 points</td>
<td>Y0 to Y7 8 points</td>
</tr>
<tr>
<td>FX0 - 14M</td>
<td>X0 to X7 8 points</td>
<td>Y0 to Y5 6 points</td>
</tr>
<tr>
<td>FX - 128M</td>
<td>X0 to X77 64 points</td>
<td>Y0 to Y77 64 points</td>
</tr>
<tr>
<td>FX - 80M</td>
<td>X0 to X47 40 points</td>
<td>Y0 to Y47 40 points</td>
</tr>
<tr>
<td>FX - 64M</td>
<td>X0 to X37 32 points</td>
<td>Y0 to Y37 32 points</td>
</tr>
<tr>
<td>FX - 48M</td>
<td>X0 to X27 24 points</td>
<td>Y0 to Y27 24 points</td>
</tr>
<tr>
<td>FX - 32M</td>
<td>X0 to X17 16 points</td>
<td>Y0 to Y17 16 points</td>
</tr>
<tr>
<td>FX - 24M</td>
<td>X0 to X13 12 points</td>
<td>Y0 to Y13 12 points</td>
</tr>
<tr>
<td>FX - 16M</td>
<td>X0 to X7 8 points</td>
<td>Y0 to Y7 8 points</td>
</tr>
</tbody>
</table>

- The total maximum number of I/O points is 256 points for an FX using an extension units and extension blocks in combination.
## FX series device numbers

<table>
<thead>
<tr>
<th>Items</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Auxiliary relays</strong></td>
<td></td>
</tr>
<tr>
<td>General use</td>
<td>496 points (M0 to M495)</td>
</tr>
<tr>
<td>Latched</td>
<td>16 points (M496 to M511)★</td>
</tr>
<tr>
<td>Special use</td>
<td>56 points (in the range of M8000 to M8254)</td>
</tr>
<tr>
<td>Initial</td>
<td>10 points (S0 to S9)</td>
</tr>
<tr>
<td>General</td>
<td>54 points (S10 to S63)</td>
</tr>
<tr>
<td>Keep</td>
<td>—</td>
</tr>
<tr>
<td>Annunciator</td>
<td>—</td>
</tr>
<tr>
<td><strong>States</strong></td>
<td></td>
</tr>
<tr>
<td>100 msec</td>
<td>56 points (T0 to T55) 0.1 to 3276.7 sec</td>
</tr>
<tr>
<td>10 msec</td>
<td>24 points (T32 to T55) 0.01 to 3276.7 sec *1</td>
</tr>
<tr>
<td>1 msec additional</td>
<td>—</td>
</tr>
<tr>
<td>100 msec additional</td>
<td>—</td>
</tr>
<tr>
<td>Analog</td>
<td>1 point (indirect designation of special D8013) 0.1 to 25.5 sec</td>
</tr>
<tr>
<td><strong>Timers</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Counters</strong></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>14 points (C0 to C13) 16-BIT</td>
</tr>
<tr>
<td>Up/down</td>
<td>2 points (C14, C15) 16-BIT ★</td>
</tr>
<tr>
<td>High speed</td>
<td>4 points (C235 to C254) 32-BIT ★</td>
</tr>
<tr>
<td><strong>Registers</strong></td>
<td></td>
</tr>
<tr>
<td>General-purpose register</td>
<td>30 points (D0 to D29) 16-BIT</td>
</tr>
<tr>
<td>Special register</td>
<td>2 points (D30, D31) 16-BIT ★</td>
</tr>
<tr>
<td>Index use</td>
<td>2 points (V, Z) 16-BIT</td>
</tr>
<tr>
<td>File use</td>
<td>—</td>
</tr>
<tr>
<td><strong>Pointers</strong></td>
<td></td>
</tr>
<tr>
<td>Branch use</td>
<td>64 points (see P0 to P63) JUMP instruction</td>
</tr>
<tr>
<td>Interrupt use</td>
<td>4 points (input interrupt by I0 to I3)</td>
</tr>
<tr>
<td>Nesting</td>
<td>8 points (N0 to N7) for master control</td>
</tr>
</tbody>
</table>

★ backed up by built-in EEPROM.

☆ battery-backed up.

*1 T32 to T55 become 10 msec times when the special auxiliary relay, M8028, is turned ON.
### Error messages

<table>
<thead>
<tr>
<th>Messages</th>
<th>Causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMS. ERROR</td>
<td>PC communication error</td>
<td>Check the PC and cable.</td>
</tr>
<tr>
<td>WRITE FORBIDDEN</td>
<td>An attempt was made to write data to EPROM.</td>
<td>Change the destination memory.</td>
</tr>
<tr>
<td></td>
<td>The EEPROM cassette memory protect switch is set in the ON position</td>
<td>Set the memory protect switch to the OFF position before writing data to EEPROM.</td>
</tr>
<tr>
<td></td>
<td>when an attempt was made to write to EEPROM.</td>
<td></td>
</tr>
<tr>
<td>NOT FOUND</td>
<td>The designated instruction was not found.</td>
<td>Proceed to next step.</td>
</tr>
<tr>
<td>ENTRY CODE ERROR</td>
<td>An operation was attempted that is not allowed with the Keyed-in entry code.</td>
<td>Attempt only the operations that are allowed for the set protection level.</td>
</tr>
<tr>
<td>NOT USABLE</td>
<td>The selected function cannot be used under the current conditions.</td>
<td>Select a usable function.</td>
</tr>
<tr>
<td>VERIFY ERROR</td>
<td>Mismatched step data was found.</td>
<td>Correct the mismatch.</td>
</tr>
<tr>
<td>STEP OVERFLOW</td>
<td>The designated step number is greater than the allowable maximum step number.</td>
<td>Change the step number.</td>
</tr>
<tr>
<td>SETTING ERROR</td>
<td>The set value or data is improper.</td>
<td>Key in proper value or data.</td>
</tr>
<tr>
<td>PC PARA. ERROR</td>
<td>The set PC parameter is incorrect.</td>
<td>Set a correct PC parameter.</td>
</tr>
<tr>
<td>PC RUNNING</td>
<td>A write operation is attempted while the PC is in the RUN state.</td>
<td>Set the PC to the STOP state.</td>
</tr>
<tr>
<td>NO PROGRAM SPACE</td>
<td>There is no more program storage area.</td>
<td>Change the parameter settings.</td>
</tr>
<tr>
<td>PROGRAM OVERFLOW</td>
<td>No more memory space for inserts.</td>
<td>Delete all NOP instructions from the program. If the program is still larger than the available memory area, revise the program.</td>
</tr>
<tr>
<td>COMMAND ERROR</td>
<td>The instruction is incorrect.</td>
<td>Set a correct instruction.</td>
</tr>
<tr>
<td>NO MEM. CASSETTE</td>
<td>A memory cassette is not installed to the PC.</td>
<td>Install the memory cassette.</td>
</tr>
<tr>
<td>DEVICE ERROR</td>
<td>The designated device or pointer is incorrect.</td>
<td>Input the correct device or pointer.</td>
</tr>
</tbody>
</table>
## 8. APPENDICES

### Error messages (program check)

<table>
<thead>
<tr>
<th>Error Messages</th>
<th>Error Codes</th>
<th>Error Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PC H/W ERROR</strong></td>
<td>6101</td>
<td>RAM error</td>
</tr>
<tr>
<td></td>
<td>6102</td>
<td>Operation circuit error</td>
</tr>
<tr>
<td></td>
<td>6103</td>
<td>I/O bus error (M8069 driven)</td>
</tr>
<tr>
<td><strong>COMMS. ERR.</strong></td>
<td>6201</td>
<td>Parity error, overrun error, frame error</td>
</tr>
<tr>
<td></td>
<td>6202</td>
<td>Communications character error</td>
</tr>
<tr>
<td></td>
<td>6203</td>
<td>Communications data sum check error</td>
</tr>
<tr>
<td></td>
<td>6204</td>
<td>Data format error</td>
</tr>
<tr>
<td></td>
<td>6205</td>
<td>Command error</td>
</tr>
<tr>
<td><strong>LINK ERROR</strong></td>
<td>6301</td>
<td>Parity error, overrun error and frame error</td>
</tr>
<tr>
<td></td>
<td>6302</td>
<td>Communications character error</td>
</tr>
<tr>
<td></td>
<td>6303</td>
<td>Communications data sum check error</td>
</tr>
<tr>
<td></td>
<td>6304</td>
<td>Data format error</td>
</tr>
<tr>
<td></td>
<td>6305</td>
<td>Command error</td>
</tr>
<tr>
<td></td>
<td>6306</td>
<td>Watchdog timer error</td>
</tr>
<tr>
<td><strong>PARA. ERROR</strong></td>
<td>6401</td>
<td>Program sum check</td>
</tr>
<tr>
<td></td>
<td>6402</td>
<td>Memory capacity setting error</td>
</tr>
<tr>
<td></td>
<td>6403</td>
<td>Keep area setting error</td>
</tr>
<tr>
<td></td>
<td>6404</td>
<td>Comment area setting error</td>
</tr>
<tr>
<td></td>
<td>6405</td>
<td>File register area setting error</td>
</tr>
<tr>
<td></td>
<td>6409</td>
<td>Other setting error</td>
</tr>
<tr>
<td><strong>GRAMMAR ERROR</strong></td>
<td>6501</td>
<td>Instruction, a device symbol, device number combination error</td>
</tr>
<tr>
<td></td>
<td>6502</td>
<td>There is no OUT T or C before the setting value.</td>
</tr>
<tr>
<td></td>
<td>6503</td>
<td>There is no setting value after OUT T and C. Insufficient operands with applied instructions</td>
</tr>
<tr>
<td></td>
<td>6504</td>
<td>Duplicate label number used; Overlapping designation of interrupt input and/or high-speed counter input</td>
</tr>
<tr>
<td></td>
<td>6505</td>
<td>Device number range over</td>
</tr>
<tr>
<td></td>
<td>6509</td>
<td>Other</td>
</tr>
<tr>
<td>Error Messages</td>
<td>Error Codes</td>
<td>Error Descriptions</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>LADDER ERR.</td>
<td>6601</td>
<td>LD or LDI used more than 8 times continuously.</td>
</tr>
<tr>
<td></td>
<td>6602</td>
<td>1) No LD or LDI instruction. No coil. Incorrect relationship of LD/LDI and ANB/ORB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) One of the following is not connected to the bus line:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STL, RET, MCR, P (pointer), I (interrupt), EI, DI, SRET, IRET, FOR, NEXT, FEND, END</td>
</tr>
<tr>
<td></td>
<td>6603</td>
<td>MPS used more than 11 times continuously.</td>
</tr>
<tr>
<td></td>
<td>6604</td>
<td>Incorrect relationship between MPS and MRD/MPP</td>
</tr>
<tr>
<td></td>
<td>6605</td>
<td>1) STL used more than 8 times continuously</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) MC, MCR, I (interrupt), or SRET in STL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3) RET outside STL. No STL</td>
</tr>
<tr>
<td></td>
<td>6606</td>
<td>1) No P or I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) No SRET or IRET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3) I, SRET, or IRET designated in the main program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4) STL, RET, MC, and MCR designated in the subroutine or interrupt routine.</td>
</tr>
<tr>
<td></td>
<td>6607</td>
<td>1) Illegal FOR and NEXT designation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nesting level deeper than 5 levels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) One of the following commands is designated in the FOR-NEXT loop.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STL, RET, MC, MCR, IRET, SRET, FEND, END</td>
</tr>
<tr>
<td></td>
<td>6608</td>
<td>1) Illegal MC and MCR designation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) No MCR N0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3) One of the following commands is designated in the MC-MCR loop.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRET, IRET, I</td>
</tr>
<tr>
<td></td>
<td>6609</td>
<td>Other</td>
</tr>
</tbody>
</table>
### Error Messages

<table>
<thead>
<tr>
<th>Error Codes</th>
<th>Error Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>6701</td>
<td>No jump destination of CJ or CALL. A label follows after the END instruction. An independent label is in the FOR-NEXT loop or a subroutine program.</td>
</tr>
<tr>
<td>6702</td>
<td>CALL nesting level deeper than 5 levels</td>
</tr>
<tr>
<td>6703</td>
<td>Interrupt nesting level deeper than 2 levels</td>
</tr>
<tr>
<td>6704</td>
<td>FOR-NEXT loop nesting level deeper than 5 levels</td>
</tr>
<tr>
<td>6705</td>
<td>A device other than a valid is used for an applied instruction operand.</td>
</tr>
<tr>
<td>6706</td>
<td>The device number or data designated as an applied instruction operand is outside the allowable designation range.</td>
</tr>
<tr>
<td>6707</td>
<td>File register operation is accessed without allocating the file register area. Other (No IRET or SRET or illegal FOR-NEXT loop)</td>
</tr>
<tr>
<td>6709</td>
<td></td>
</tr>
</tbody>
</table>

### I/O ERROR

(Example) 1020

The I/O number is not supported by the current hardware.

**An explanation of the error code.**

```
1 0 2 0
```

- In the case of X20
- Device number
- 1: Input X
- 0: Output Y (leading '0's' will not be displayed)
Contents of Program Memory

(1) A sequence program may contain constants for the setting [K] for timers and counters.

(2) It is necessary to allocate a block by a parameter setting for file register.

   Block 0: No file register
   Block 1: D1000 to D1499 500 points/500 steps
   Block 2: D1000 to D1999 1000 points/1000 steps
   Block 3: D1000 to D2499 1500 points/1500 steps
   Block 4: D1000 to D2999 2000 points/2000 steps

(3) The HPP cannot read or write comments.

   However, if a program containing comments is transferred to the HPP RAM, the
   comments will be written to the RAM area.

(4) The parameter area sets the program memory capacity, latch memory device
   number range, file register block number, entry code, etc.

(5) Pages 110 and 111 give details about the FX0 PC.
Program Memory Types

There are three program memory capacities available:
2K steps, 4K steps and 8K steps.

The FX memory capacity, steps 0 to 1999/3999/7999, depends on the parameter setting and the area available from the RAM, EEPROM, and EPROM cassettes. The program memory capacity of the FX0 PC is fixed to 800 steps. This is stored on internal EEPROM.

(1) RAM

Battery backup is necessary.
A RAM (2K steps) area is incorporated in the PC and an FX-RAM-8 RAM cassette (8K steps) is also available.
If the RAM cassette is removed from the PC, the program in the cassette will be lost.

(2) EEPROM

Programs cannot be written unless the memory protect switch is set to the OFF position.
The following EEPROM cassettes are available:
  FX-EEPROM-4 (4K steps), FX-EEPROM-8 (8K steps).

(3) EPROM

Use a ROM writer module to write the program to the EPROM cassette.
The FX-EPROM-8 (8K steps) EPROM cassette is available.

- In addition, there are an FX-RAM-8C cassette and the FX-EEPROM-4C EEPROM cassette with a real-time clock (clock function).
However, these cassettes need battery backup.
(1) Bit device memory

The bit device memory stores the ON/OFF state of a contact or coil operation; the status of input relay X, output relay Y, auxiliary relay M, state S, timer T and counter C.

Some M, S, T, and C, device data is backed up by the battery. For example, M0 to M499 are general use, while M500 to M1023 can be battery backed. The boundary point M499/M500 can be changed by a parameter setting.

(2) Word device memory

The word device memory stores the current value of timers (T), counters (C), data registers (D) and other devices. The word device memory area is divided into the two parts; the general use area are the battery back up area. The area size can be changed by changing the parameter setting.

(3) Since the FX0 PC holds backup data in its built-in EEPROM, a battery is unnecessary. The backup range (latch range) is fixed.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Types of Program Memory</th>
<th>RAM built in the PC</th>
<th>EX-EEPROM:4</th>
<th>FX-RAM-8</th>
<th>FX-EPROM-8</th>
<th>FX-EEPROM-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>FX</td>
<td>Sequence program</td>
<td>0 to 2K</td>
<td>0 to 4K</td>
<td>0 to 8K</td>
<td>0 to 2K</td>
<td>0 to 2K</td>
</tr>
<tr>
<td></td>
<td>File register</td>
<td>0 to 2K</td>
<td>0 to 2K</td>
<td>0 to 4K</td>
<td>0 to 2K</td>
<td>0 to 4K</td>
</tr>
<tr>
<td></td>
<td>Comment</td>
<td>0 to 2K</td>
<td>0 to 4K</td>
<td>0 to 2K</td>
<td>0 to 4K</td>
<td>0 to 4K</td>
</tr>
<tr>
<td></td>
<td>Total (available memory)</td>
<td>2K</td>
<td>2K, 4K</td>
<td>2K, 4K</td>
<td>2K, 4K, 8K</td>
<td></td>
</tr>
</tbody>
</table>

- Each file register or comment block is the equivalent of 500 programming steps.
- The FX0's built in program capacity is 800 steps. However, file registers and comments cannot be registered.
<table>
<thead>
<tr>
<th>Items</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory capacities</td>
<td>2K steps</td>
</tr>
<tr>
<td></td>
<td>Same as the memory cassette contents</td>
</tr>
<tr>
<td>File registers</td>
<td>None available</td>
</tr>
<tr>
<td></td>
<td>0 without allocation</td>
</tr>
<tr>
<td>Comment areas</td>
<td>0 without allocation</td>
</tr>
<tr>
<td></td>
<td>0 without allocation</td>
</tr>
<tr>
<td>Latch ranges</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>496 to 511</td>
</tr>
<tr>
<td></td>
<td>500 to 1023</td>
</tr>
<tr>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>500 to 999</td>
</tr>
<tr>
<td>C (16)</td>
<td>14 to 15</td>
</tr>
<tr>
<td></td>
<td>100 to 199</td>
</tr>
<tr>
<td>C (32)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>220 to 255</td>
</tr>
<tr>
<td>D</td>
<td>30 to 31</td>
</tr>
<tr>
<td></td>
<td>200 to 511</td>
</tr>
<tr>
<td>Entry codes</td>
<td>Unregistered</td>
</tr>
<tr>
<td></td>
<td>Unregistered</td>
</tr>
<tr>
<td>Comments</td>
<td>Unregistered</td>
</tr>
<tr>
<td></td>
<td>Unregistered</td>
</tr>
</tbody>
</table>

- The parameter is rewritten to the default value by batch NOP writing to the entire range, also by the default value setting operation of a parameter.
- Comment areas cannot be allocated and comments cannot be registered by the HPP.
- The default value of the parameter of FX₀ is the same as the 2K steps of FX. The memory capacity is fixed to 800 steps and the latch range for actual operation is fixed to the range indicated by the ★ symbol in the table on page 100. Although file registers and comment areas can only be set to 0, the entry codes and titles can be registered.
8. APPENDICES

This appendix explains the precautions to take when handling peripheral devices while using an FX0 series PC.

1. Parameter settings: Latch ranges, memory capacities, comment capacities and file register capacities must be default values.

(1) Memory capacity settings
(a) Program capacity: Select 2K step. Do not set a value other than 2K step.

If a value more than 2K step is set:

If writing is executed to an FX0 series PC, the "PC MEMORY TOO SMALL" message is displayed and writing cannot be executed.

Writing to an FX0 series PC when 2K step is set:
Step 800 and after are ignored.
If a program with more than 800 steps is written to an FX0 series PC, step 800 and after will not be written to the PC.

Reading from an FX0 series PC when 2K step is set:
NOP instructions will be written after the end of the program or step 800, which ever is lower.

(b) Comment capacity: Do not set the comment capacity.

Comment capacity set less than 2 blocks:
If writing is executed to an FX0 series PC, the "WRITE ERROR" message is displayed.
(No comment is written.)

Comment capacity set more than 3 blocks:
If writing is executed to an FX0 series PC, the "WRITE ERROR" message is displayed and program writing cannot be executed.

(c) File register: Do not set a file register.

Writing to an FX0 series PC when a file register is set:
If writing is executed to an FX0 series PC, the "WRITE ERROR" message is displayed and program writing cannot be executed.
8. APPENDICES

(2) Latch range setting: Latch ranges cannot be changed when an FX0 series PC is used. When writing is executed after setting (changing) a latch range, the "COMPLETED" message is displayed and writing is executed. However, the FX0 series PC ignores the new latch range.

(3) Program title registration: Program titles can be registered when an FX0 series PC is used.

(4) Entry code registration: Entry codes can be registered when an FX0 series PC is used.

2. Changing constants during RUN

When an FX0 PC is used, constants (timer and counter set values, present value of data register) can be changed even though an EEPROM is used as the program memory. However, the PC scan time becomes longer (20 to 30 msec) and a response delay (20 to 30 msec) is caused by input interrupt I00[ ] to I30[ ].

3. Program cable

The FX-20P-CAB type program cable (1.5 m, supplied separately) should be used to connect the FX-10P-E to an FX PC. However, when connecting to an FX0, the FX-20P-CAB0 type program cable (1.5 m) should be used (supplied separately).
MEMO
OPERATION MANUAL
FX-10P-E

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-8310
HIMEJI WORKS: 840, CHIYODA CHO, HIMEJI, JAPAN

<table>
<thead>
<tr>
<th>MODEL</th>
<th>FX-10P-O-E</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL CODE</td>
<td>09R913</td>
</tr>
</tbody>
</table>

Effective Mar. 2003
Specifications are subject to change without notice.