MITSUBISHI ELECTRIC

CNC

MELDAS 60 Series
MELDASMAGIC64

PLC PROGRAM DEVELOPMENT MANUAL
(FOR PERSONAL COMPUTER)

Introduction

This manual describes the handling of the PLC development software, a support tool used to develop the user PLC (ladders, messages) for the MELDAS 64 Series or MELDASMAGIC 64 Series.
Please read this manual before starting use.
Please read the "Precautions for Safety" listed on the following page to ensure safe use of the MELDAS 64 Series and MELDASMAGIC 64 Series.

Details described in this manual

⚠ CAUTION

⚠ For items described in "Restrictions" or "Usable State", the instruction manual issued by the machine maker takes precedence over this manual.

⚠ Items not described in this manual must be interpreted as "not possible".

⚠ This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine maker before starting use.

⚠ Some screens and functions may differ or may not be usable depending on the NC version.

General precautions

Refer to the following documents for details on the handling.

(1) MELDAS64 Series

MELDAS 64 PLC Onboard Instruction Manual.........................	BNP-B2213
MELDAS 64 PLC Programming Manual (Ladder section)	BNP-B2212
MELDAS 64 PLC Interface Manual	BNP-B2211

(2) MELDASMAGIC64 Series

MELDASMAGIC 64 PLC Onboard Instruction Manual.............	BNP-B2213
MELDASMAGIC 64 PLC Programming Manual (Ladder section)	BNP-B2212
MELDASMAGIC 64 PLC Interface Manual..............................	BNP-B2211
MELDASMAGIC 64 MELDASMAGIC Monitor Operation Manual	BNP-B2192
MELDASMAGIC 64 Utility Manual	BNP-B2196

(Note) In this manual, the MELDASMAGIC man-machine interface software is abbreviated as the MMI software or MMI.

Precautions for Safety

Always read the specifications issued by the machine maker, this manual, related manuals and enclosed documents before starting installation, operation, programming, maintenance or inspections to ensure correct use. Thoroughly understand the basics, safety information and precautions of this numerical controller before using the unit.
The safety precautions are ranked as "DANGER", "WARNING" and "CAUTION" in this manual.

⚠ **DANGER**	When there is a great risk that the user could be subject to fatalities or serious injuries if handling is mistaken.
⚠ **WARNING**	When the user could be subject to fatalities or serious injuries if handling is mistaken.
⚠ **CAUTION**	When the user could be subject to injuries or when physical damage could occur if handling is mistaken.

Note that even if the item is ranked as " ⚠ CAUTION", incorrect handling could lead to serious results. Important information is described in all cases, so please observe the items.

⚠ **DANGER**
Not applicable in this manual

⚠ **WARNING**
Not applicable in this manual.

⚠ **CAUTION**
1. Items related to product and manual ⚠ For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine maker takes precedence over this manual. ⚠ Items not described in this manual must be interpreted as "not possible". ⚠ This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine maker before starting use. ⚠ Some screens and functions may differ or may not be usable depending on the NC version.

Contents

Contents

Contents

1. Outline

This manual describes the handling of the PLC development software, a support tool used to develop the user PLC (ladders, messages).
Ladder circuits and mnemonic expressions are used as the programming language with this tool, so a sequence program can be created easily.
A function (message creation function) to create messages such as the alarm messages and operator messages, and the ladder contact comments and coil comments is also provided.

(Note 1) PLC is the abbreviation for Programmable Logic Controller.

(Note 2) The terms "control unit" and "controller" used in this manual refer to the "NC Card" in the MELDASMAGIC 64 Series.

(Note 3) The " ⏎ " key used in this manual refers to the Enter key (Return key). Note that the key names may differ according to the type of personal computer being used.

2. System Configuration

2.1 System Configuration for PLC Development

The system configuration used for PLC development is shown below.

(1) MELDAS 64 Series

Communication terminal

The ladder is developed using the communication terminal. (Onboard development)

Control unit

Program development, ladder monitor and PLC RUN/STOP, etc.

To connector AUX

Base I/O unit

RS-232-C

Up/downloading is carried out with the control unit's maintenance function.

RS-232-C

Personal computer
Used for development and saving data.
(Hard disk or floppy disk)

Commercial printer
(Ex.: PC-PR201G2)

(Note) When developing (onboard development) with the communication terminal, refer to the "MELDAS 64 Series PLC Onboard Development Manual".

(2) MELDASMAGIC 64 Series

By using the NC's onboard function from the M64 monitor, the ladders can be developed and the signal operation can be monitored, etc.
By using the optional PLC development software, the ladders can be developed even if an NC Card is not provided.

Personal computer
Used for development and saving data
(Hard disk or floppy disk)

Display unit

MELDASMAGIC monitor
(onboard)
PLC development
software
Various utility disk tools

Utility disk

MAGIC
Monitor

PLC
development
software

Keyboard

RS-232-C

NC Card

Commercial printer
(Ex.: PC-PR201GS)

PLC ladder area
NC Card built-in RAM
16K steps
128K bytes

Base I/O unit

Dedicated printer
(PRT-02B)

(Note 1) When developing (onboard development) with the MELDASMAGIC monitor, refer to the "MELDASMAGIC Series PLC Onboard Development Manual".

(Note 2) Refer to the MELDASMAGIC Monitor Operation Manual for details on the MELDASMAGIC monitor.

2.2 Devices Required for Ladder Development

	Device, tool	Details	Supplier
Hardware	(1) Personal computer	**PC9801 Series (NEC)** CPU : 80386 or higher Required memory: Main memory 640K bytes 　　　　　 + Protected RAM 1M byte or more Required disk unit: 1.25M byte compatible floppy 　　　　　　 disk drive x 1 unit 　　　　　 Hard disk open capacity 1M 　　　　　 byte or more **PC/AT compatible unit** CPU : 80386 or higher Required memory: Main memory 640K bytes 　　　　　 + Protected RAM 1M byte or more Required disk unit: 1.44M byte compatible floppy 　　　　　　 disk drive x 1 unit 　　　　　 Hard disk open capacity 1M 　　　　　 byte or more	Commercial
	(2) Printer	This is used to output the ladder. (Ex.) PC-PR201 GS	Commercial
	(3) Cable	This cable is used to connect the personal computer and printer.	Commercial
	(4) Cable	This cable is used to connect the control unit and personal computer. (R020 or equivalent)	Mitsubishi Electric
Software	PLC development software package PC9801 Series 　　　 DEV-A10Z01 PC/AT compatible unit 　　　 DEV-A10Z02	PLC4B: Ladder editing tool 　The ladders and messages can be edited, linked, divided and printed. LIST4B: List editing tool 　The list can be edited and converted into a text file. COMU: Ladder transfer tool 　The PLC program file is transferred between the personal computer and NC using a serial I/F.	Mitsubishi Electric
	MELDASMAGIC 50 　Utility disk	CNV4BMM: Ladder file format conversion tool 　(PC/AT version only) 　The file is converted between the PLC4B format and MELDASMAGIC format. MAGICTRS: Ladder file transfer tool 　This is Windows software used to transfer the file between the personal computer disk and NC.	Mitsubishi Electric
	Japanese conversion F.E.P. (ATOK)	Word processing type is used to type in Kanji codes when editing the messages.	Commercial
	OS	PC9801 Series: MS-DOS Version 3.3 and above PC/AT compatible unit: MS-DOS Version 5.0 and 　　　　　　　　　 above	Commercial

(Note 1) For the various tools of the PLC development software to run properly, a protected RAM (1M byte or more) in the personal computer and the operating system MS-DOS Version 3.3 and above (MS-DOS Version 5.0 and above for PC/AT compatible unit) must be provided.

(Note 2) Either the PC9800 Series or PC/AT series personal computer can be used.
Note that even if the personal computer is compatible, the PLC development software may not run in some cases.

3. Development Procedure

The procedure used to create a new user PLC on the personal computer is shown below.

1 | Tool installation | The PLC development software tools are copied into the personal computer.
Refer to the section "4.1 Installing the Tools".

2 | Starting | The PLC development software is started with the PLC4B.
Refer to the section "4.2 Starting Up and Ending PLC4B".

3 | File registration | The names of the ladder and message files are registered.
Refer to the section "5. Registering the Edit File".

4 | Ladder creation | A sequence program is created using a ladder circuit.
Refer to the section "6. Creating a Ladder Circuit".

5 | Message creation | Messages such as the alarm messages are created.
Refer to the section "7. Messages".

6 | Printing | The created ladder and message list is output.
Refer to the section "8. Printing".

7 | Link file creation | The ladder file and message file are linked, and a link file is created.
Refer to the section "9. Linking".

8 | End | The PLC development software is quit.

9 | Transfer to actual machine | The created link file is transferred to the NC.
Refer to the section "11. Transferring Ladder Files (MELDAS 64)" or "12. Transferring Ladder Files (MELDASMAGIC 64)".

This procedure will be explained in the following sections.

3.1 General Flow of Ladder Development Procedure

The following procedure is used to develop a built-in ladder with the PLC development software.

(1) MELDAS 64 Series

```
                    ┌──────────────┐
                    │    Start     │
                    └──────────────┘
                           │
┌──────────────────────────────────────┐      Refer to the section "2. System
│ Connect the personal computer and     │ ───► Configuration" for details on the
│ control unit used for the PLC          │      connection.
│ development tool (PLC4B).              │
└──────────────────────────────────────┘
                           │
┌──────────────────────────────────────┐      Refer to the "PLC Onboard Instruction
│ Stop the PLC using RUN/STOP under the  │ ───► Manual" for details.
│ menu file of the onboard function or   │
│ by setting the rotary switch NCSYS on  │
│ the control unit to "1".               │
└──────────────────────────────────────┘
```

Control unit


```
┌──────────────────────────────────────┐      Refer to the section "4. Start Up" for how to start
│ Start the PLC development tool (PLC4B) │ ───► the PLC development tool (PLC4B). To edit a
│ and create or edit the ladders and     │      ladder that is already running in the control unit,
│ messages.                              │      start the ladder file's communication program
└──────────────────────────────────────┘      (COMU.EXE), and read the file into the
                           │                    personal computer.
┌──────────────────────────────────────┐
│ Quit the PLC development tool, and     │
│ convert the ladder file using the      │
│ MELDAS-compatible ladder conversion    │
│ tool (chscx.exe).                      │
└──────────────────────────────────────┘
                           │
┌──────────────────────────────────────┐      Refer to section "11. Transferring Ladder
│ Start the ladder file's communication  │ ───► Files (MELDAS 64)" for details.
│ program (COMU.EXE), and transfer the   │
│ link file to the control unit.         │
└──────────────────────────────────────┘
                           │
┌──────────────────────────────────────┐
│ Run the PLC using RUN/STOP under the   │
│ menu file of the onboard function or   │
│ by setting the rotary switch NCSYS on  │
│ the control unit to "0".               │
└──────────────────────────────────────┘
                           │
┌──────────────────────────────────────┐      Refer to the "PLC Onboard Instruction
│ Debug using the onboard monitor        │ ───► Manual" for details.
│ function, etc.                         │
└──────────────────────────────────────┘
                           │
                    ╱──────────────╲
              ┌─── ╱ Completion of    ╲
              No   ╲ creating (debugging)╱
                    ╲──────────────╱
                           │ Yes
                    ┌──────────────┐
                    │     End      │
                    └──────────────┘
```

(2) MELDASMAGIC 64 Series

```
            ┌─────────────┐
            │    Start    │
            └─────────────┘
```

Flowchart step	Reference
Connect the personal computer and NC Card used for the PLC development tool (PLC4B).	Refer to the section "2. System Configuration" for details on the connection.
Stop the PLC using RUN/STOP under the menu file of the onboard function.	Refer to the "PLC Onboard Instruction Manual" for details.
Start the PLC development tool (PLC4B) and create or edit the ladders and messages.	Refer to the section "4. Start Up" for how to start the PLC development tool (PLC4B). To edit a ladder that is already running in the NC Card, start the file transfer tool (MAGICTRS), and read the file into the personal computer.
Quit the PLC development tool, start the file format conversion tool (Chgcx.EXE), and convert the file into a format usable by the MELDASMAGIC 64.	Refer to section "12.3 Converting the PLC Program Data" for details.
Quit the PLC development tool, start the file transfer tool (MAGICTRS.EXE), and transmit the link file to the NC Card.	Refer to the section "12. Transferring Ladder Files (MELDASMAGIC 64)" for details.
Run the PLC using RUN/STOP under the menu file of the onboard function.	
Debug using the onboard monitor function, etc.	Refer to the "PLC Onboard Instruction Manual" for details.

```
         ◇ Completion of creating (debugging) ◇
No ◄──────┘                              │
                                        Yes
                                         │
                                ┌─────────────┐
                                │     End     │
                                └─────────────┘
```

3.2 Before Starting Operations

The basic items for using the PLC development software are explained in this section.

3.2.1 Menu operation

The following title screen will display when the PLC development software is started up.

```
PLC  DEVELOPMENT  SOFTWARE

        ┌──────────┐                    ┌────────────┐
        │ IBM PC-AT │                    │ SYSTEM No. │
        └──────────┘                    └────────────┘
     COPYRIGHT  (C)  1992  MITSUBISHI  ELECTRIC  CORPORATION
     ALL  RIGHTS  RESERVED




     ➤ 1MSSG              4LADDER  5LINK  6EXIT IO  7FILE  8EXIT
```

└── Menu display section

When the title screen is displayed, the menu will appear on the bottom of the screen.

Use the function keys PF1 to PF10 to select a menu. The menu number and function key numbers correspond to each other.

(These keys may be F1 to F10 depending on the personal computer model.)

The menu configuration is shown below.
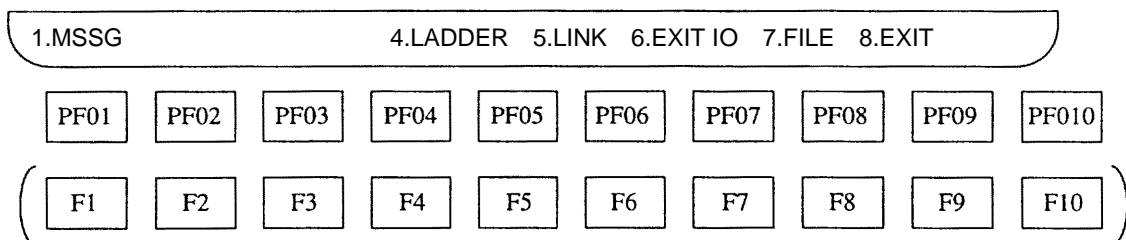
```
 1.MSSG                    4.LADDER  5.LINK  6.EXIT IO  7.FILE  8.EXIT

 PF01  PF02  PF03  PF04  PF05  PF06  PF07  PF08  PF09  PF010

  F1    F2    F3    F4    F5    F6    F7    F8    F9    F10
```

When 1.MSSG is selected:

```
                        PF 1

 1.ALARM  2.OPERAT  3.PLC SW  4.DEVICE  5.COIL  6.COMMENT  7.INIT  SUBMENU

            PF 1 ~ 6            PF 7                          PF8

 1.READ  2.WRITE  3.INSERT  4.DELETE  5.CONVT      7.MONIT  SUBMENU  MENUE

        2.WRITE              5.CONVT               SUBMENU  MENUE
```

When 4.LADDER is selected:

PF 4

2.CIRCUT

PF 2

1.READ   2.WRITE   3.INSERT   4.DELETE   5.CONVT   MENUE

PF 1 ～ 5

Note 1)

INS

1. ┤├ 2. ┤／├ 3. └┐┌┘ 4. └┐／┌┘ 5. ＜ ＞ 6. ┤ ├ 7. ─   8. │   9．SET
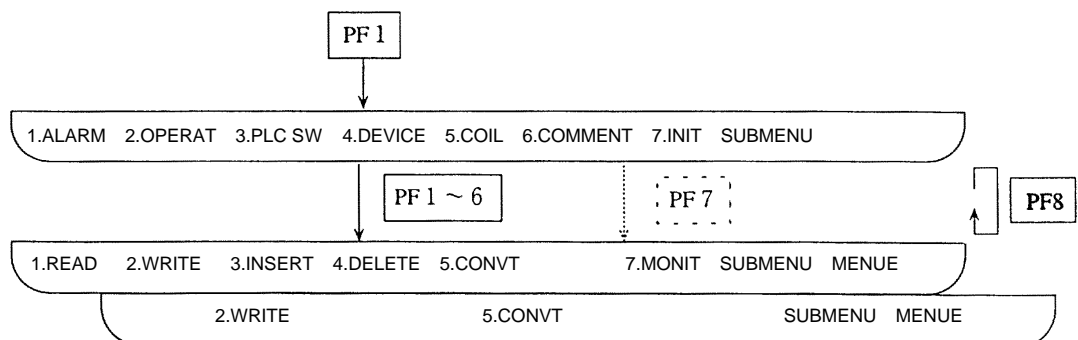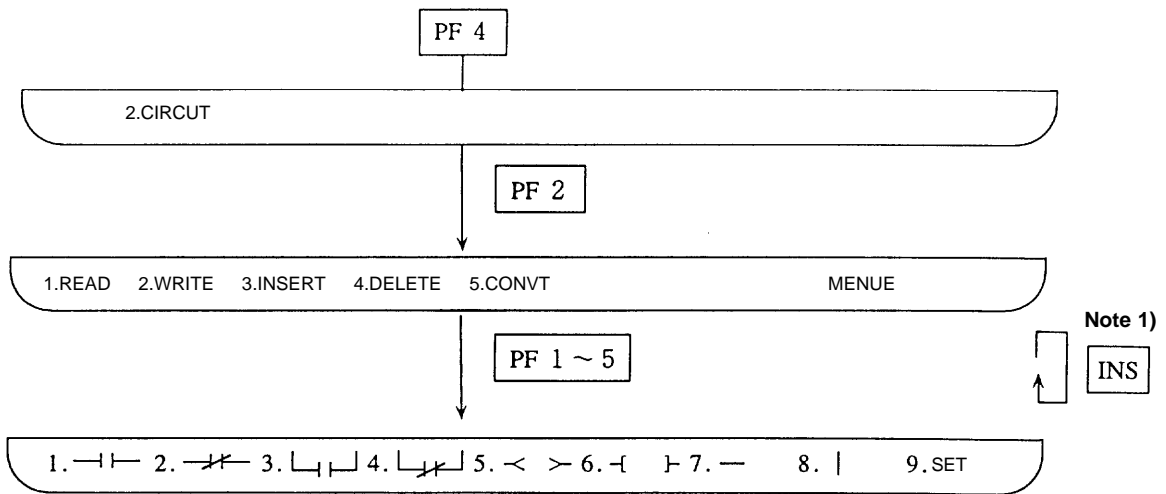
Note 1) ESC = can be used for INS.

When 5.LINK is selected:

PF 5

1.READ   5.LINK   7.DIVIDE
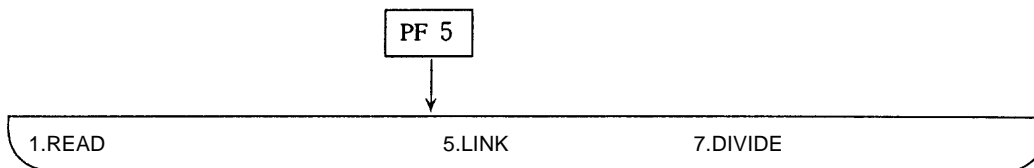
When 6.EXIT IO is selected:

PF 6

3.PRINT   5.PRINT2
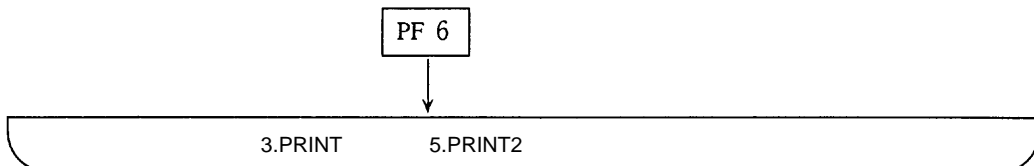
When 7.FILE is selected:

PF 7

2.WRITE
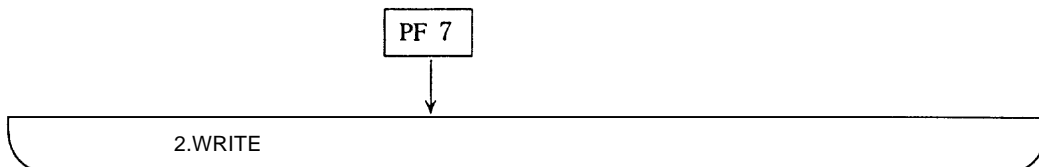
The PLC development software title screen will appear when PF10 is pressed.

3.2.2 Movement of the cursor

○ During ladder creation: · 1 step movement   ↑ ↓ ← →

○ During message creation: · To move between setting sections   TAB

                            · To move between columns   ← →

– 9 –

3.2.3 Explanation of terms

(1) Device and device number

The device is the address signal used to classify the signals handled by the PLC. The device number is the serial number allocated to that device.

The device numbers for devices X, Y, U, W and H are expressed with a hexadecimal, and the others are decimals.

List of devices

Device	Device number		Unit	Description
X∗	X0~X4BF	(1216 points)	1 bit	Signal input to PLC for machine input, etc.
Y∗	Y0~Y53F	(1344 points)	1 bit	Signal output from PLC for machine output, etc.
U∗	U0~U178	(384 points)	1 bit	Signal input to PLC for No.2 system.
W∗	W0~W1FF	(512 points)	1 bit	Signal output from PLC for No. 2 system.
M	M0~M5119	(5120 points)	1 bit	Temporary memory
G	G0~G3071	(3072 points)	1 bit	Temporary memory
F	F0~F127	(128 points)	1 bit	Temporary memory. Alarm message interface.
L	L0~L255	(256 points)	1 bit	Latch relay (back up memory)
E∗	E0~E127	(128 points)	1 bit	Special relay
T	T0~T15	(16 points)	1 bit/16 bit	10ms unit timer
	T16~T95	(80 points)	1 bit/16 bit	100ms unit timer
	T96~T103	(8 points)	1 bit/16 bit	100ms unit integral timer
Q	Q0~Q39	(40 points)	1 bit/16 bit	10ms unit timer (fixed timer)
	Q40~Q135	(96 points)	1 bit/16 bit	100ms unit timer (fixed timer)
	Q136~Q151	(16 points)	1 bit/16 bit	100ms unit integral timer (fixed timer)
C	C0~C23	(24 points)	1 bit/16 bit	Counter
B	B0~B103	(104 points)	1 bit/16 bit	Counter (fixed counter)
D	D0~D1023	(1024 points)	16 bit/32 bit	Data register. Register for calculation.
R∗	R0~R8191	(8192 points)	16 bit/32 bit	File register. PLC to CNC interface. The user released registers are R500 to R549 and R1900 to R2799. R1900 to R2799 are backed up by the battery.
A	A0, A1	(2 points)	16 bit/32 bit	Accumulator
Z	–	(1 point)	16 bit	D or R address index (for ±n)
V	–	(1 point)	16 bit	D or R address index (for ±n)
N	N0~N7	(8 points)	–	Nesting level of master control.
P∗	P0~P255	(256 points)	–	Label for conditional jump and subroutine call. (P300 to P511 are used to call C language modules.)
K	K-32768~K32767		–	Decimal constant for 16-bit command.
	K-2147483648~K2147483647		–	Decimal constant for 32-bit command.
H	H0~HFFFF		–	Hexadecimal constant for 16-bit command.
	H0~HFFFFFFFF		–	Hexadecimal constant for 32-bit command.

(Note 1) The devices with a ∗ mark in the device column have designated application.
Do not use the non-defined device numbers even if they are blank.

(Note 2) In addition to the above devices, there are the I, J and S devices, but these must not be used.

(Note 3) There are limits to the usage range, so refer to the "PLC Programming Manual (ladder section)" for explanations on each device.
(For example, the user release range for point P is P0 to P159.)
(Note 4) The X100 to X13F, Y100 to Y13F, R80 to R83 and R180 to R183 devices cannot be used with the MELDASMAGIC64 Series.

(2) Circuit signals

The following eight types of circuit signals are used.

Circuit signal	Description
—||—	Used for A contact circuit.
—|/|—	Used for B contact circuit.
└||┘	Used for A contact OR circuit.
└|/|┘	Used for B contact OR circuit.
—< >—	Used for coils (Y, M, G, F, L, E, T, C etc.)
—[ ]—	Used for programming machine commands. (MOV, =, +, −, etc.)
|	Used for connecting between circuit symbols.
—	Used for connecting between circuit symbols.

(3) Circuit block

The circuit block is the circuit closed with the —< >— —[ ]— circuit symbols.

(Example)

Circuit block

– 11 –
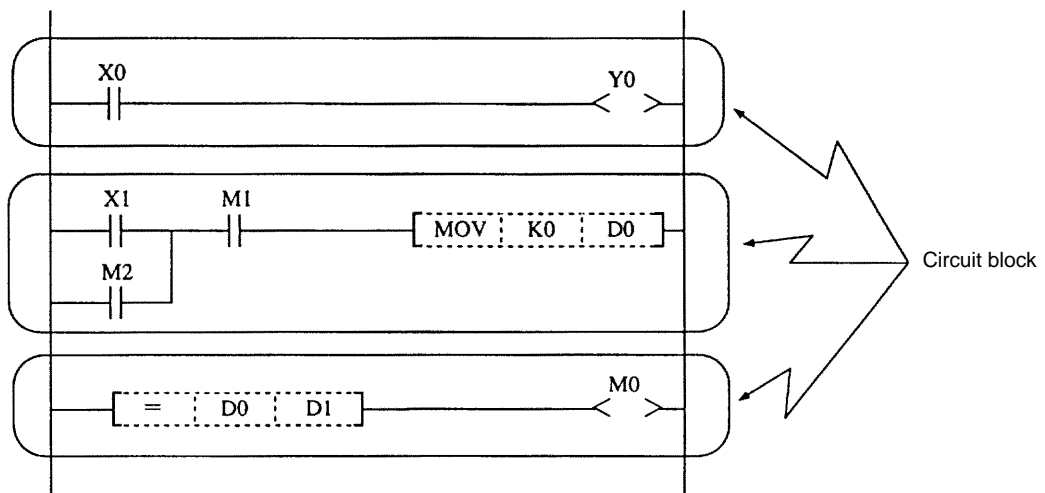
(4) Step numbers and pointers (P)

The step numbers are allocated in order to the created program, and is automatically changed when the circuit is edited.

The pointers (P) are used as labels for the jumping destination of the conditional jump command and for the jumping destination for the subroutine command. The points (P) shown below are for special applications.

P128~P159 : These can be used in the same manner as P0~P127 but these will also function as the page return when printing out the ladder circuit diagram.

P250~P225 : These are used to separate the PLC program processing levels.

P251 is assigned to the head of the high-speed processing program.
P252 is assigned to the head of the main processing program.

END is not displayed.
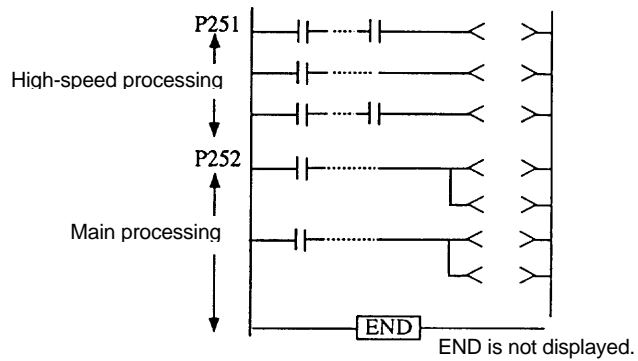
(Example)
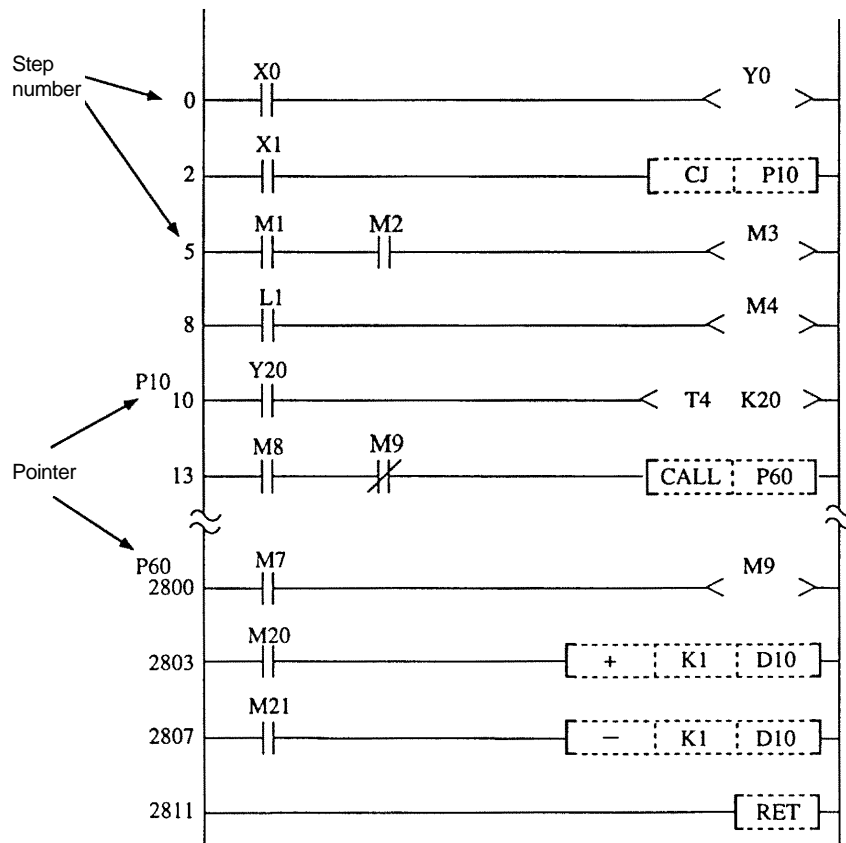
4. Start Up

4.1 Installing the Tools

Confirm that there is no PLC directory below the hard disk drive root directory. If there is a PLC directory, change the name of the existing PLC directory, or more to a place other than the root directory.

(Operation procedure)

(Note 1) This operation example uses the A drive of the personal computer for the hard disk drive and the B drive for the floppy disk drive.
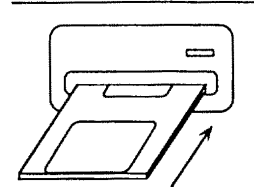
(1) Turn on the personal computer and start up MS-DOS.

(2) Insert the floppy disk labeled as "PLC Development Tool Version PC-98" (or Version PC-AT) into the B drive.

(3) Input the install command.
(Example)
Input "B:INSTALL_A:_B: ⏎ ".
(_ indicates a space.)

(4) Comments will appear as shown on the right, and the installation will be completed.

(5) The following files will be copied to the designated drive during the installation process.

```
A:\> B:INSTALL_A:_B:

Input the underlined section, and press the return key.
```

```
        1_File(s)_copied
              .
              .
              .
        1_File(s)_copied
LHA's SFX 2.13L (c) Yoshi, 1991 ...............................
COMU.MAE...............................................................
MELDAS_PLC_DEVELOPMENT_TOOL_INSTALLATION_IS
FINISHED.
 A:\>
```

<For Version PC-98>			<For Version PC/AT>	
A: \CHG4PB.BAT	A:\PLC\BIN	\ATOKCH.EXE	C:\CHG4PB.BAT	C:\PLC\BIN\ATOKCH.EXE
\HELP4B.BAT		\CHG4PB.EXE	\HELP4B.BAT	\CHG4PB.EXE
\LIST4B.BAT		\CODECCNV.INF	\LIST4B.BAT	\CODECCNV.INF
\M3MFD.BAT		\COME.BAT	\M3MFD.BAT	\COME.BAT
\RUN386.EXE		\COME.EXE	\RUN386.EXE	\COME.EXE
\HELP4B.BAT		\COME.MAE	\HELP4B.BAT	\COME.MAE
		\COME.MAJ		\COME.MAJ
		\M3MFD.EXE		\M3MFD.EXE
		\NORMCNV.EXE		\NORMCNV.EXE
		\HELP4B.EXE		\HELP4B.EXE
		\HELP4B.MAE		\HELP4B.MAE
		\HELP4B.MAJ		\HELP4B.MAJ
		\PLC2B.EXP		\PLC2B.EXP
		\PLCCM4B.H		\PLCCM4B.H
		\PLCLIST.H		\PLCLIST.H
		\PLC_LB.EXP		\PLC_LB.EXP
		\SET9801.EXE		\SETIBMPC.EXE
		\SETMFAS.EXE		

(6) When the installation is completed, use a commercial editor, and delete the following device drivers related to the memory from "CONFIG.SYS".

   EMM386.EXE
   EMM.SYS
   HIMEN.SYS
   SMARTDRV.SYS

Change the "CONFIG.SYS" details as shown below.

   [Example of CONFIG.SYS details]
   BUFFERS=30
   FILES=30
   DEVICE=A:\DOS\RSDRV.SYS

Refer to a commercial book on MS-DOS for details on CONFIG.SYS.

4.2 Starting Up and Ending PLC4B

PLC4B is the tool used to newly create or edit the ladders and messages.

[Start up method]

(1) The PLC development software will start up by inputting command "plc4b". Refer to section 5 for explanations on each screen.

```
A:\>plc4b ⏎
```

(2) When the PLC development software starts, the title screen will appear.
     The initial screen of the PLC development software will appear when the PF10 key is pressed on any screen while PLC4B is running.

[Ending method]

(1) Press PF10 on the keyboard, and display the initial screen of the PLC development software.

(2) Press PF8 (8.END). The prompt (A:\>) will appear on the screen, and the program will return to the normal MS-DOS mode.

5. Registering the Edit File

Register the file name and estimated file size when creating a new ladder circuit or message.

[Basic operation]

$\boxed{\text{7.FILE}}$ → $\boxed{\text{2.WRITE}}$ → $\left[\begin{array}{l}\text{Set each necessary setting section} \\ \text{number file name and size.}\end{array}\right]$ → ⏎
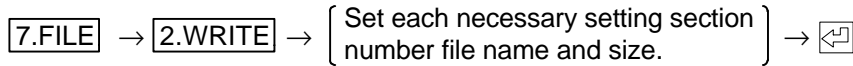
[Operation procedure]

(1)　The PLC FILE screen will display when menu $\boxed{\text{7.FILE}}$ is pressed.

(2)　The setting section will appear when $\boxed{\text{2.WRITE}}$ is pressed.

(3)　Set the ladder file name and estimated file size, and message 1 file name and estimated file size. The file name must be a maximum of 10 characters.

　　(Note) If the file name exceeds 10 characters the last 10 characters will be valid.
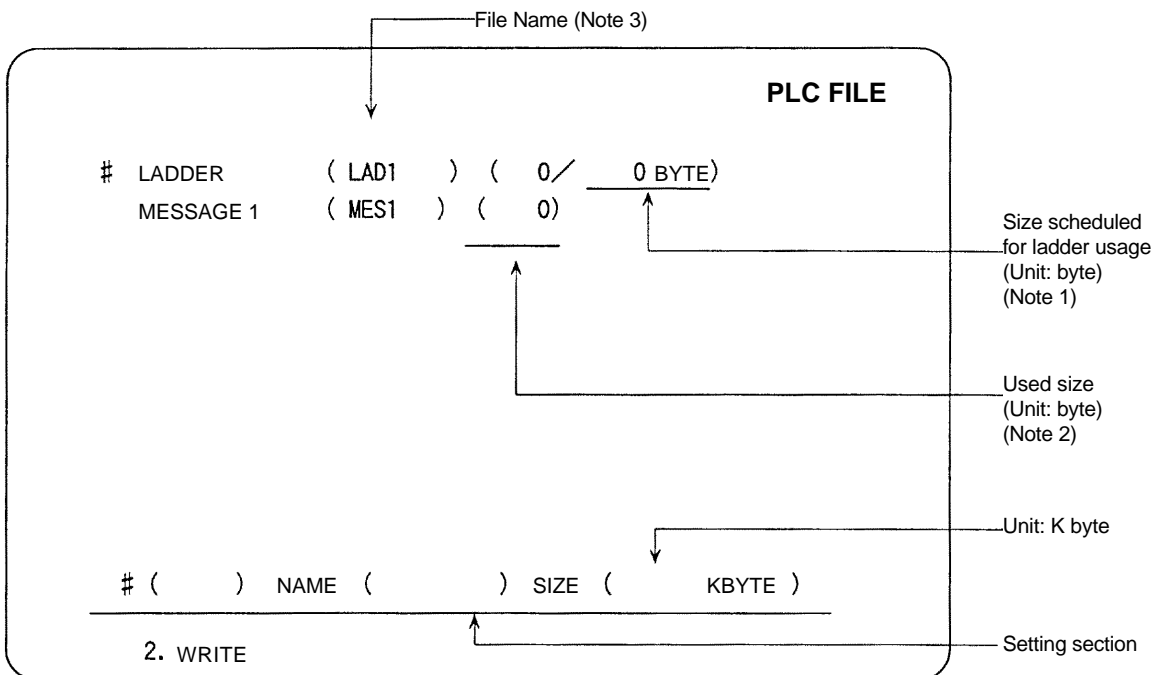
(Note 1) The file name and size scheduled for ladder usage is set in "# LADDER".
　　　　The size scheduled for ladder usage is a maximum of 16 Kbytes (16384 bytes).
　　　　　　One ladder step: 4 bytes
(Note 2) Only the file name is set in "# MESSAGE 1".
　　　　When various messages are registered with the initial setting on the message creation size, the size will be displayed automatically.
(Note 3) Set the file that exists in the same directory as PLC4B.EXE or the name of the file created in the same directory here.

6. Creating a Ladder Circuit

The method for editing the sequence program using the ladder circuit is explained here.
A list of ladder circuit editing functions is shown below.

Function item	
Circuit write	Creation of new circuit
	Correction of existing circuit
	Addition of circuit block
Circuit read	Circuit read with step number
	Circuit read with device number
	Circuit read with contact or coil number
	Circuit read with command
	Final circuit read with END command
Circuit insertion	Insertion in circuit symbol units
Circuit delete	Delete of circuit block
	Delete in circuit symbol units

6.1 Creating a New Circuit

This operation is used to create a new ladder circuit or to erase an existing ladder circuit.

[Basic operation]

4.LADDER → 2.CIRCUT → 2.WRITE → 9.SET → Y → ⏎ →

→ (Writing operation) → INS → 5.CONVT → ⏎

[Operation procedure]

(1) Register the edit file before creating a new circuit.

(2) Press function menu 4.LADDER and the sub menu will display.

(3) Press 2.CIRCUT , 2.WRITE , 9.SET and the following interactive screen will display.

(4) Press Y , ⏎ to create a new circuit or erase an existing circuit. Press 1.READ for other applications.

(5) Only the two vertical lines on both sides and step number 0 will display if Y , ⏎ are pressed.

(6) Start the writing operation
   (1) Write the program from the cursor position using the circuit symbols (menu keys) and alphanumeric keys.
   (2) When approximately one screen worth of program is written, press INS 5.COUNT , ⏎ .

   Note) A maximum of two screens worth of program can be created at once, but INS

   5.CONVT , ⏎ should be pressed after one screen if possible.
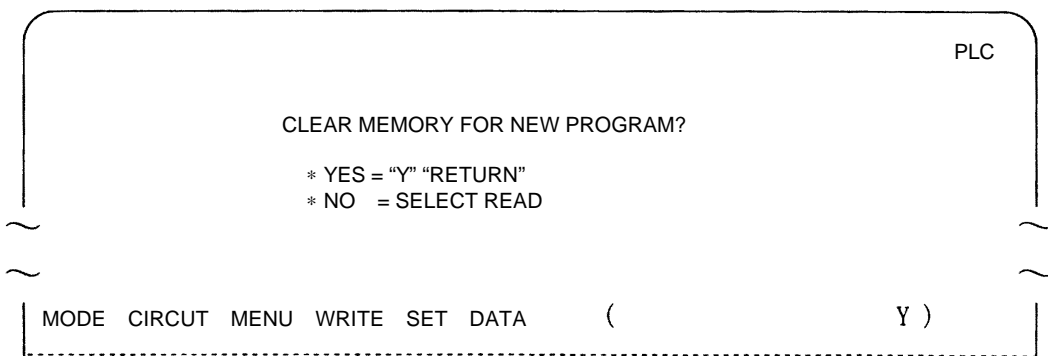
   The final object will not be created on the screen if INS 5.CONVT , ⏎ are not pressed after the circuit is completed.

   (3) The message "COMPLETED" will appear on the screen when INS 5.CONVT , ⏎ are pressed and the conversion is completed.

   (Example)
   The following screen will display when the following are pressed:

   4.LADDER → 2.CIRCUT → 2.WRITE → 9.SET

```
                                                              PLC

                 CLEAR MEMORY FOR NEW PROGRAM?

                   * YES = "Y" "RETURN"
                   * NO  = SELECT READ

   MODE  CIRCUT  MENU  WRITE  SET  DATA      (              Y )
```

   Note) The existing circuits will all be deleted when Y ⏎ are pressed.

The following screen will display when $\boxed{Y}$ → $\boxed{\hookleftarrow}$ are pressed. Perform the writing operation on this screen.

```
                                                              PLC

         ■
   0  ┤

      ┤

      ┤

 ～    ┤                                                            ～
 ～                                                                 ～
      Memory (   0 /   4096 steps)


   MODE  CIRCUT  MENU  WRITE  SET  DATA    (            ■ )
```
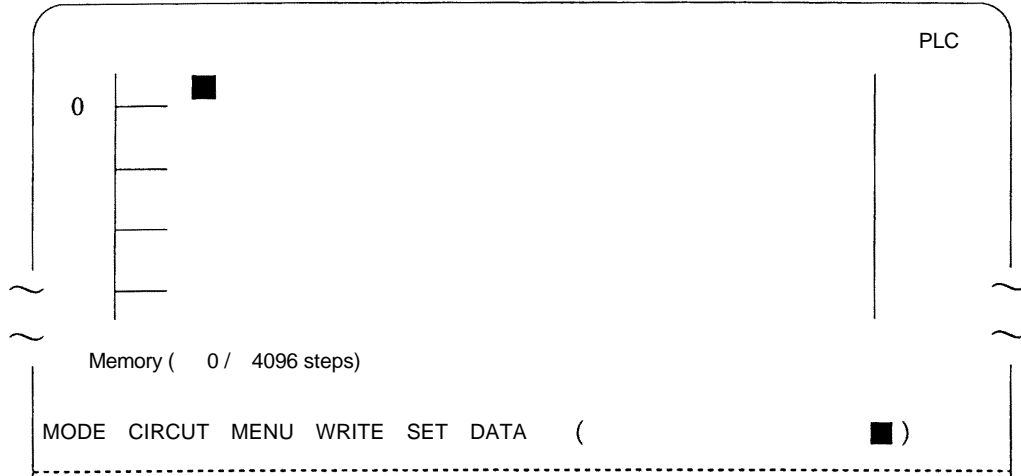
6.2 Modification of Existing Circuit

An existing sequence circuit can be modified.

[Basic operation]

(Read out circuit) → INS 2.WRITE → | ↑ | | ↓ | | → | | ← | → (Input commands) → INS 5.CONVT → ⏎
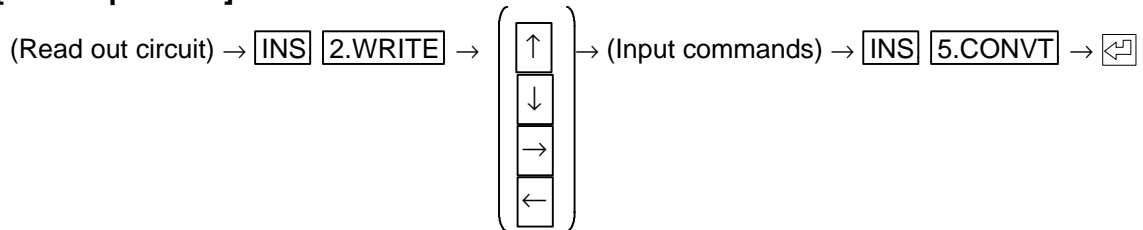
[Operation procedure]

(1) Read out the circuit to be modified with the read procedure.

(2) When 2.WRITE is pressed, the circuit block at the cursor position will display and the following blocks will disappear.
  ○ To modify a circuit block other than the one displayed on the top of the screen, move the cursor to the top of the target block and press 2.WRITE. That circuit block will display on the top line and the other blocks will disappear.

(3) Move the cursor to the circuit position to be modified and input the commands.
   (Use ↑ ↓ ← → keys)

(4) Always press INS 5.CONVT ⏎ after modifying the circuit.
  ○ The message "COMPLETED" will appear when the ⏎ key is pressed, and the circuit will display after modification.

> **POINT**
>
> (1) When the existing circuit where the step number is changed is modified, the step numbers of the existing program and the CJ command jump destination will also be automatically changed.

(Example) To change coil Y10 to Y35.

2.CIRCUT → 1.READ → 9.SET → 2 → 0 → ⏎ → INS

Read step No.20

↳ 2.WRITE → → → 5.─< >─ → Y → 3 → 5 → ⏎ INS 5.CONVT → ⏎

Change Y10 to Y35

Change to Y35

```
20   X10                              Y10
    ──┤├──┬──                       ──< >──┤
     M15  │
    ──┤/├──┘
```
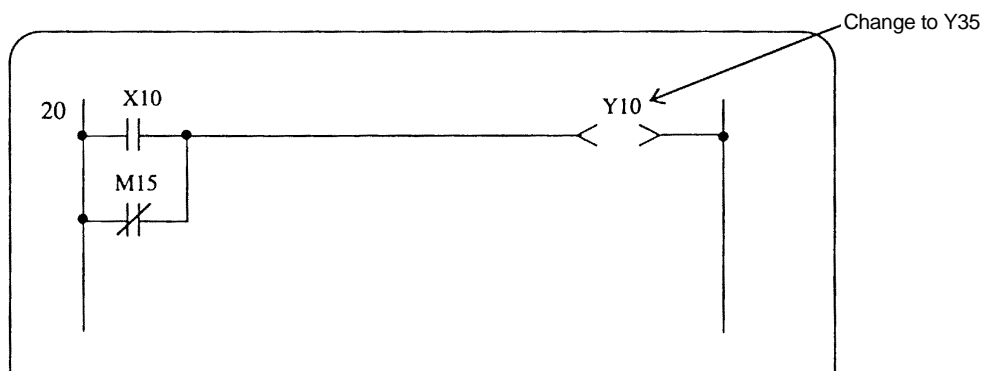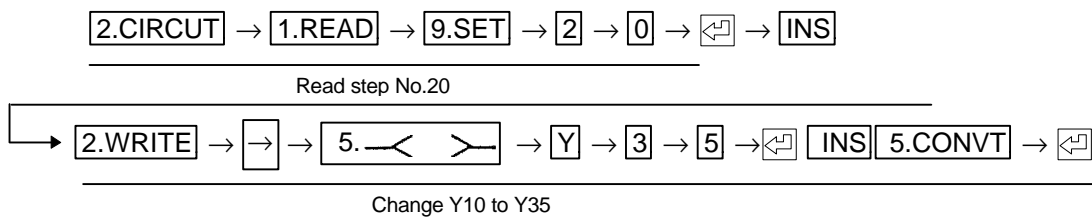
6.3 Adding a Circuit Block

This operation is used to add a circuit to the existing sequence circuits.

[Basic operation]

(Circuit read operation) → INS 2.WRITE → + → INPUT → (Command input) → INS 5.CONVT → ⏎

↓
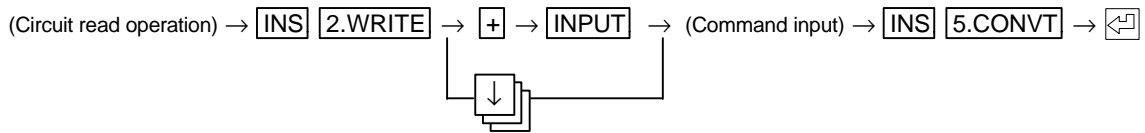
[Operation procedure]

(1) Read the circuit to where a circuit block is to be added with the read operation.
(2) When 2.WRITE is pressed, the circuit block at the cursor position will display and the following blocks will disappear.
  ○ To add a circuit block to the block other than the one displayed on the top of the screen, move the cursor to the top of the target block and press 2.WRITE . That circuit block will display on the top line and the other blocks will disappear.
(3) Press + ⏎ and the top circuit block will disappear, and the next step number and vertical lines of the circuit block will display.
(4) Write the additional circuit block by inputting the commands.
(5) Press INS 5.CONVT ⏎ after adding the circuit.

  (Example) Adding circuit blocks X5 and Y15 to step No. 50

2.CIRCUT → 1.READ → 9.SET → 5 → 0 → ⏎ → (continued on the next page)

Read step No. 50
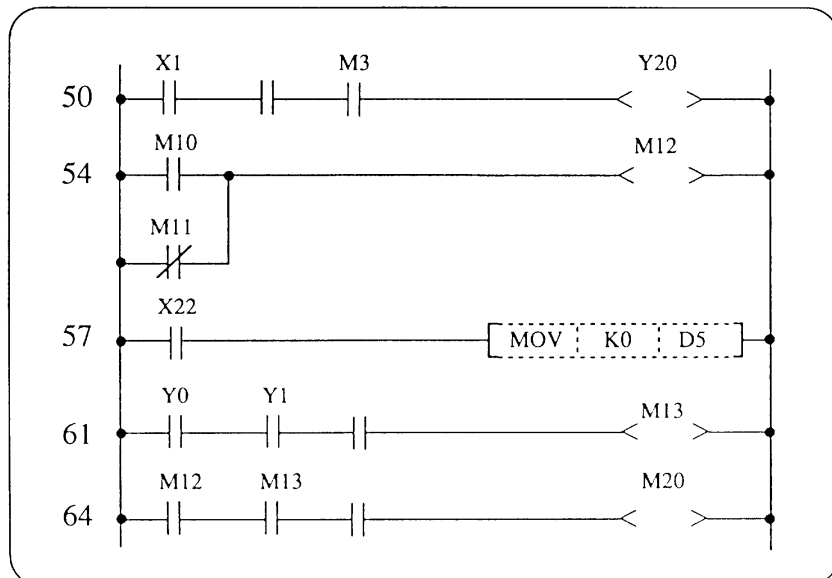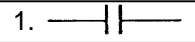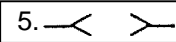
INS → 2.WRITE → + → ⏎ →

Set the write area after the read circuit block.

```
        54  ├──
            │
            ├──
            │
            ├──
            │
```
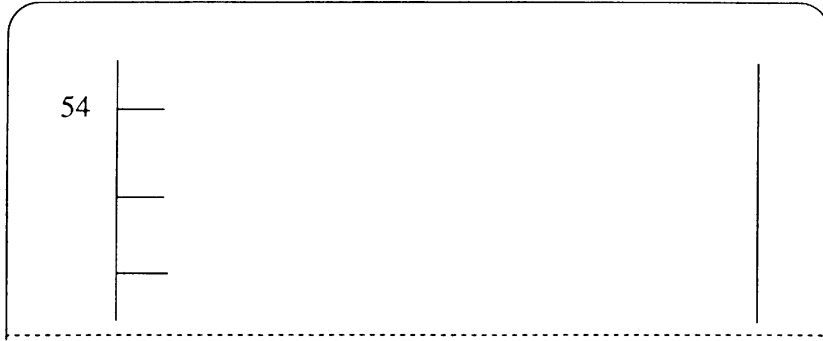
→ 1. ─┤├─ → X → 5 → ⏎ → 5. ─< >─ → Y → 1 → 5 → INPUT →

Write X5 and Y15

INS → 5.CONVT → ⏎

```
            X5                                    Y15
    54  ●──┤├──────────────────────────────< >──●
```
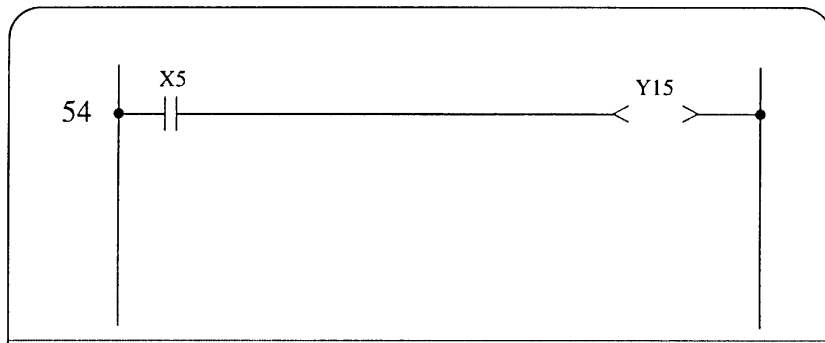
POINT

(1) When a circuit block is inserted or added, the following program step numbers and CJ command, etc., jump destinations will be automatically changed.

6.4 Reading the Circuit

6.4.1 Reading the circuit with step number

This operation is used to read the circuit by designating a step number.

[Basic operation]

2.CIRCUT → 1.READ → 9.SET → STEP NO. → ⏎

[Operation procedure]

(1) Press 2.CIRCUT, 1.READ, 9.SET, STEP NO., ⏎ and one screen worth of data
following the designated number will display.
○ Even if in the middle of the designated step number command, the circuit will be displayed
from the start of that command.

(2) Press - ⏎ and the circuit on the previous screen will display. The circuit on the next screen will
display by pressing + ⏎ . The succeeding or preceding screens can be displayed by pressing
⏎ continuously. "LADDER END" will display in the message display section when the last circuit
block is displayed.
Pressing ⏎ after - ⏎ will function in the same manner.

Note) Displaying a screen by pressing - ⏎ will take three times longer than pressing
+ ⏎ .

(3) If a circuit block is divided over two screens, use the ↓ key to move the cursor to the bottom of
the screen. Press the ↓ key continuously to scroll the screen one line. Hold down the cursor key
until the whole block appears on the screen.
The screen can be scrolled by pressing ↓ or ↑ even if the circuit block is not split between
screens. Note that pressing the ↓ key will scroll the screen but will not display the next circuit.

(Example 1) To read the step No. 100 program

2.CIRCUT → 1.READ → 9.SET → 1 → 0 → 0 → ⏎
                                Step No.
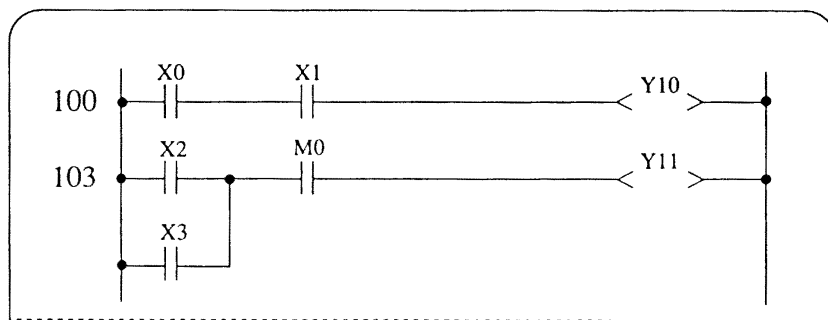
(Example 2) When step No. 100 program is read, the cursor is moved to the bottom of the screen with $\boxed{\downarrow}$, and $\boxed{\downarrow}$ is pressed four times and $\boxed{\uparrow}$ four times.

The next circuit is not displayed.
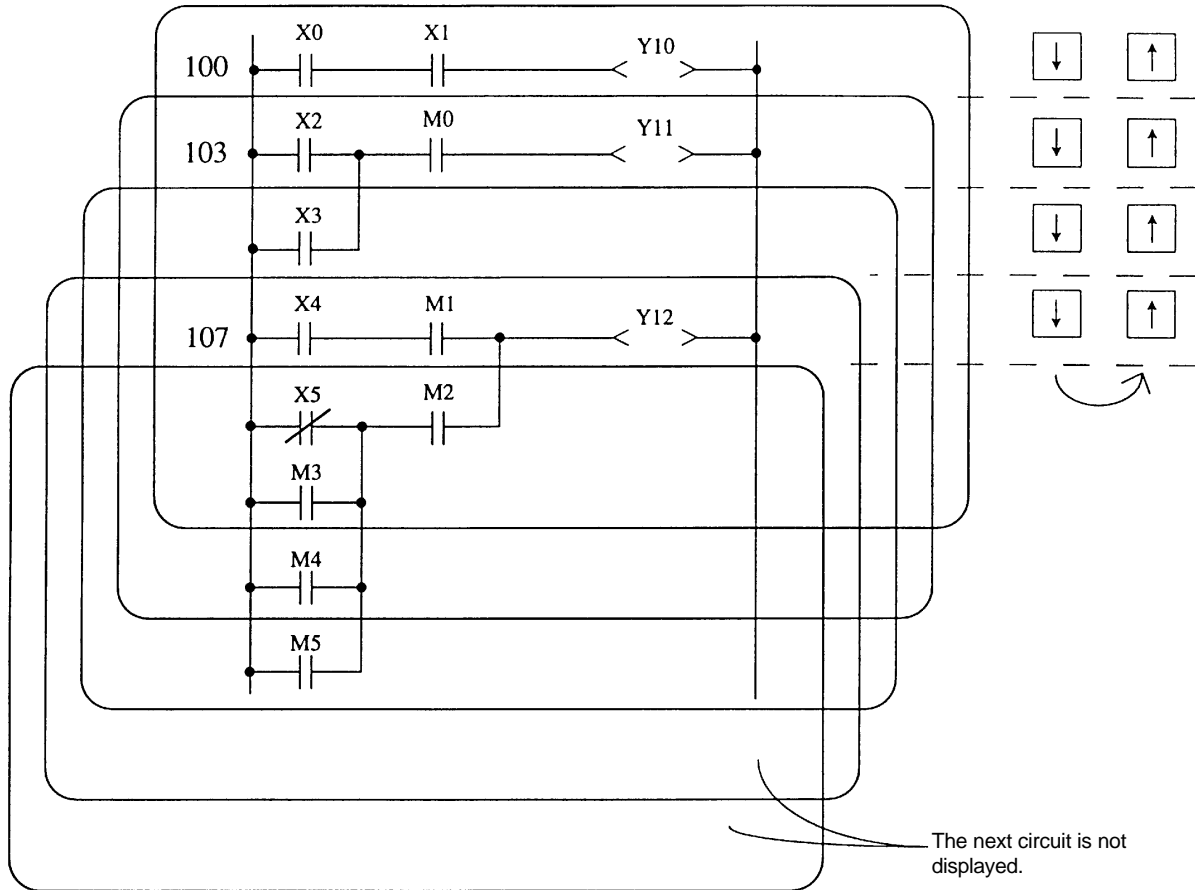
Note) If the read circuit is split, the remaining screen can be displayed with the $\boxed{\downarrow}$ key, but the next circuit cannot be displayed. The circuit before the read circuit, step No. 100 above, also cannot be displayed with $\boxed{\uparrow}$.

(Example 3) When step No. 100 program is read, and $\boxed{+} \rightarrow \boxed{\leftarrow} \rightarrow \boxed{\leftarrow} \rightarrow \boxed{-} \rightarrow \boxed{\leftarrow} \rightarrow \boxed{\leftarrow}$ are pressed.
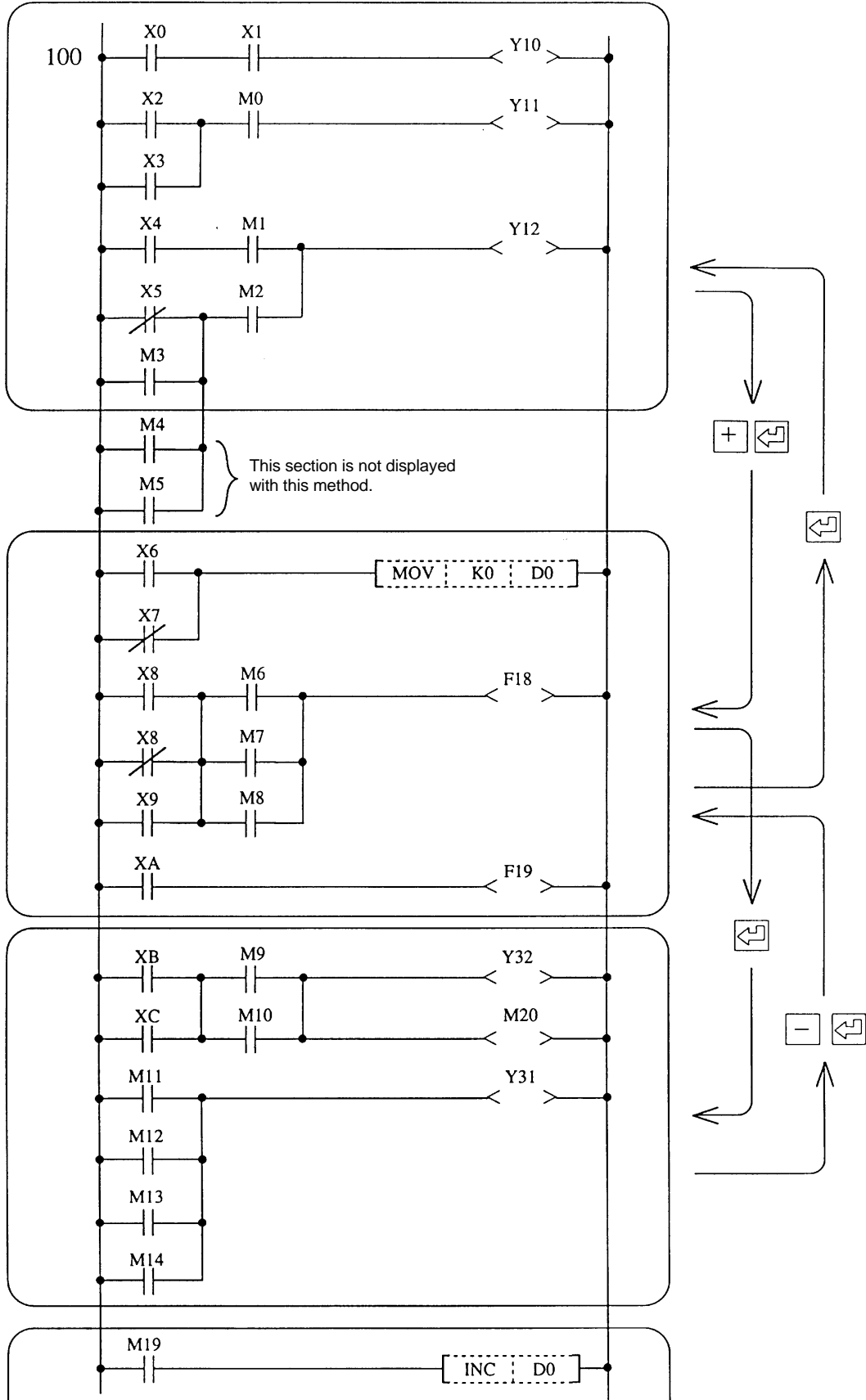
6.4.2 Reading the circuit with device number

This operation is used to read the circuit containing a designated device number.

[Basic operation]

$\boxed{\text{2.CIRCUT}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{DEVICE NO.}} \rightarrow \boxed{\leftarrow}$

[Operation procedure]

(1) Press $\boxed{\text{2.CIRCUT}}$, $\boxed{\text{1.READ}}$, $\boxed{\text{DEVICE NO.}}$, $\boxed{\leftarrow}$ and the circuit block containing the
designated device number will appear.
If there are several circuit blocks with the designated device, the screen with the smallest step number will display first. If $\boxed{\leftarrow}$ is pressed again, the circuit with the next smallest step number will be displayed under the currently displayed block.
If the screen is full and $\boxed{\leftarrow}$ is pressed, the top circuit block will disappear and another circuit block will be added to the bottom.

(2) If the device number is changed during the display, that circuit block will be added to the bottom of the currently displayed block.

(3) If reading is attempted after all blocks with the corresponding device number have been displayed, the message "PROG NOT FOUND" will appear.

  (Example) Read the circuit block containing output device Y30.

$\boxed{\text{2.CIRCUT}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{Y}} \rightarrow \boxed{3} \rightarrow \boxed{0} \rightarrow \boxed{\leftarrow}$
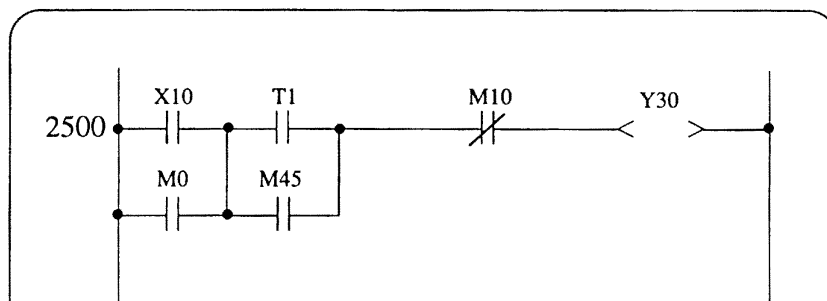
6.4.3 Reading the circuit with contact or coil number

This operation is used to read a circuit containing a designated contact or coil number.

[Basic operation]

2.CIRCUT → 1.READ → CIRCUIT SYMBOL → DEVICE NO. → ↵

```
  / 1. ─┤├─  \
  | 2. ─╱├─  |
  | 3. └┤├┘  |
  | 4. └╱├┘  |
  \ 5. ─< >─ /
```
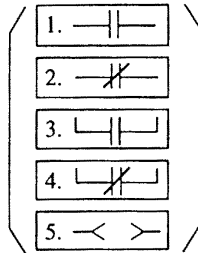
[Operation procedure]

(1) Press 2.CIRCUT , 1.READ , CIRCUIT SYMBOL , (1. ─┤├─ , 2. ─╱├─ ,
3. └┤├┘ , 4. └╱├┘ , 5. ─< >─) → DEVICE NO. and ↵ .
The circuit block containing the designated contact or coil number will display.
If several circuit blocks contain the designated contact or coil number, the circuit block with the smallest step number will display. Press ↵ again to display the circuit with the next smallest step number under the currently displayed circuit block.
If the screen is full and ↵ is pressed, the top circuit block will disappear and another circuit block will be added to the bottom.

(2) If the contact or coil number is changed during the display, that circuit block will be added to the bottom of the currently displayed block.

(3) If reading is attempted after all blocks with the corresponding contact or current number have been displayed, the message "PROG NOT FOUND" will appear.

(4) Contact A will be read if 1. ─┤├─ DEVICE NO. is pressed and a device number is set when searching with a contact, and contact B if 2. ─╱├─ DEVICE NO. is pressed and a device number is set.

(5) The OUT command will be read out by pressing 5. ─< >─ and setting a device number.

(Example) Read the circuit block containing contact A with device No. X10.

2.CIRCUT → 1.READ → 1. ─┤├─ → X → 1 → 0 → ↵
Specified contact A with device No.X10.

```
           ■ X10                          M15
  20   ●───┤├───●──────────────────────< >──────┤
           Y35
         ──╱├──
  ┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄
```
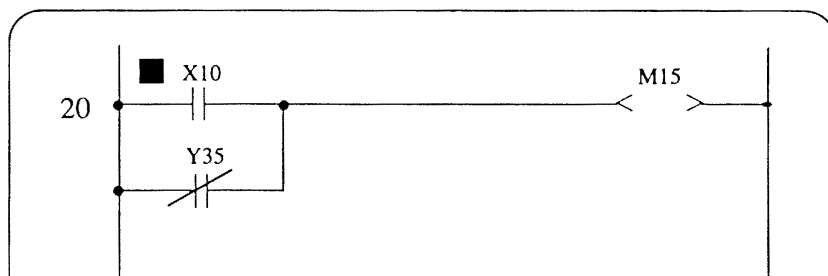
6.4.4 Reading the circuit with command

This operation is used to read a circuit containing a designated command.

[Basic operation]

$$\boxed{\text{1.CIRCUT}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{6. } \dashv[\quad]\vdash} \rightarrow \boxed{\text{COMMAND}} \rightarrow \boxed{\hookleftarrow}$$

[Operation procedure]

(1) Press $\boxed{\text{2.CIRCUT}}$, $\boxed{\text{1.READ}}$, $\boxed{\text{6. } \dashv[\quad]\vdash}$, $\boxed{\text{COMMAND}}$, $\boxed{\hookleftarrow}$ and the program with the designated command will display from the circuit block with the smallest step number. The device number designation is insignificant.
If $\boxed{\hookleftarrow}$ is pressed again, the circuit with the next smallest number will display under the currently displayed block.

(2) If the command is changed during the display, that block will display under the currently displayed block.

(3) If the screen is full and $\boxed{\hookleftarrow}$ is pressed, the top circuit block will disappear and another circuit block will be added to the bottom. (Refer to Example 2)

(4) If reading is attempted after all corresponding blocks have been displayed, the message "PROG NOT FOUND" will appear.

(Example 1) Read the circuit block with the PLS M80 command. (M80 is insignificant here.)

$$\boxed{\text{2.CIRCUT}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{6. } \dashv[\quad]\vdash} \rightarrow \boxed{\text{P}} \rightarrow \boxed{\text{L}} \rightarrow \boxed{\text{S}} \rightarrow \boxed{\text{M}} \rightarrow \boxed{8} \rightarrow \boxed{0} \rightarrow \boxed{\hookleftarrow}$$
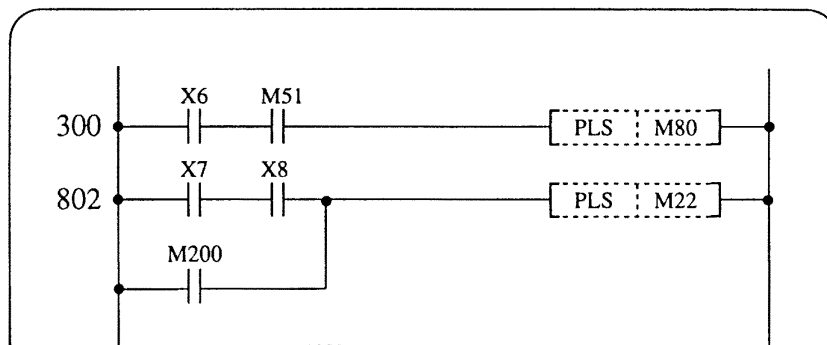
(Example 2) Read the circuit block with the MOV command.

$\boxed{\text{2.CIRCUT}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{6.} \dashv[\quad]\vdash} \rightarrow \boxed{\text{M}} \rightarrow \boxed{\text{O}} \rightarrow \boxed{\text{V}} \rightarrow \boxed{\hookleftarrow}$

Press $\boxed{\hookleftarrow}$ after the screen is full.
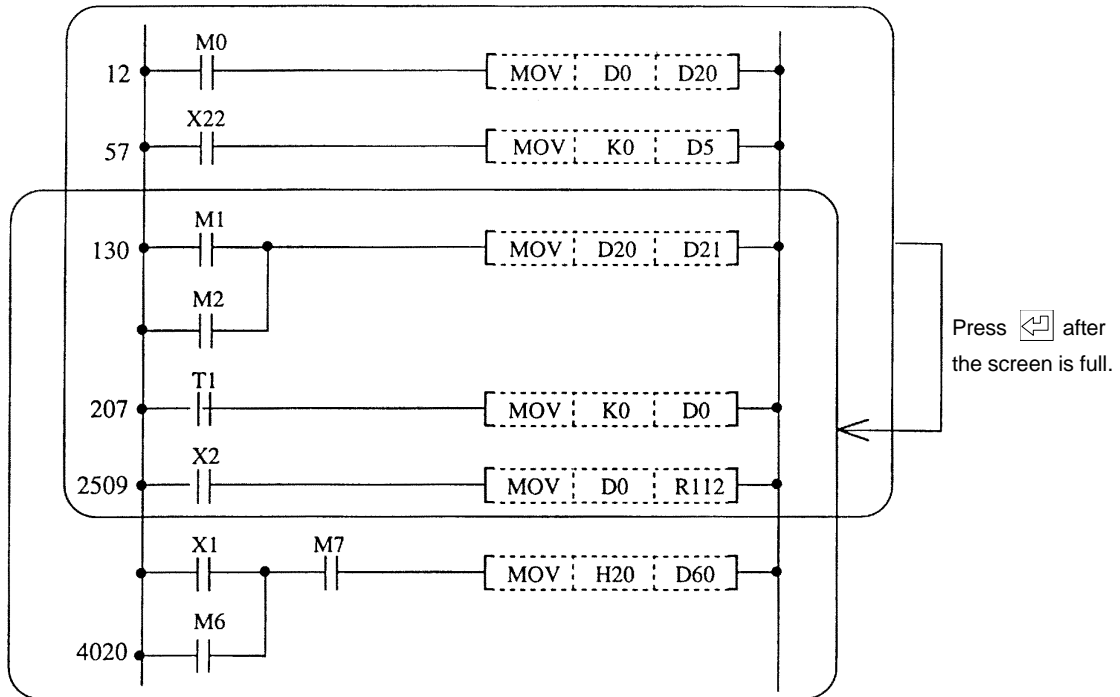
6.4.5 Reading the circuit with the END command

This operation is used to read the circuit block immediately before the block with the END command.

[Basic operation]

2.CIRCUT → 1.READ → 6. ─[   ]─ → E → N → D → ↵

[Operation procedure]

(1) Press 2.CIRCUT , 1.READ , 6. ─[   ]─ , E , N , D , and then ↵ .
The circuit block immediately before the one with the END command will display.

(Example) Read the last circuit block.

2.CIRCUT → 1.READ → 6. ─[   ]─ → E → N → D → ↵

```
                M10                              Y100
 6038  ●────┤├─────────────────────────< >────●


 ~                                                      ~
         Memory (6040/8192 steps)              LADDER END
```
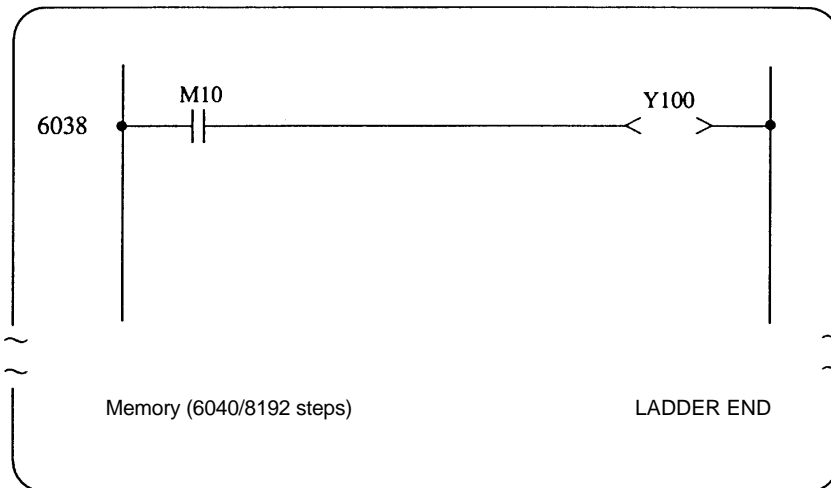
6.4.6 Circuit read functions

Read method	Operation example	Example of displayed circuit	Remarks
Step number	`9.SET` `2` `8` `↵`	 28	Reads the circuit block with the designated step and succeeding steps.
Device number	`Y` `3` `↵` (`M` `1` `0` `P` `1` `0` `↵`) ⋮	 Y3 Y3 Y3 MOV K4Y3 D2	Reads the circuit block containing the designated device number regardless of the circuit symbols.
	`T` `1` `5` `↵` [`C` `3` `↵`]	 T15 T15 RST T15	When reading a circuit block by designated T or C, the circuits with "–< >–" cannot be read. "–<T15 K15>–", etc., cannot be read.
Circuit symbol and device number	1. —│├— `Y` `3` `↵`	 Y3	Reads each circuit symbol separately. The combination of circuit symbol "–[ ]–" and device number cannot be used for reading.
	2. —│╫— `Y` `3` `↵`	 Y3	
	5. —< >— `Y` `3` `↵`	 Y3	
	5. —< >— `T` `1` `5` `↵`	 < T15 K10 >	Reads the circuit blocks of the designated timer or counter coil.
Circuit symbol and command code	6. —[ ]— `M` `O` `V` `↵`	 MOV D1 D2 MOV K10 R13 MOV K10 Z	Reads the blocks with the MOV command regardless of the device numbers.
	6. —[ ]— `E` `N` `D` `↵`	 5721	Reads the last circuit block in the sequence.

6.5 Inserting a Circuit

6.5.1 Inserting with circuit symbol units

This operation is used to insert the circuit in circuit symbol units.

[Basic operation]

(Circuit read operation) → $\boxed{\text{3.INSERT}}$ → $\left(\begin{array}{c}\boxed{\uparrow}\\\boxed{\downarrow}\\\boxed{\rightarrow}\\\boxed{\leftarrow}\end{array}\right)$ → (Command Input) → $\boxed{\text{INS}}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{\hookleftarrow}$
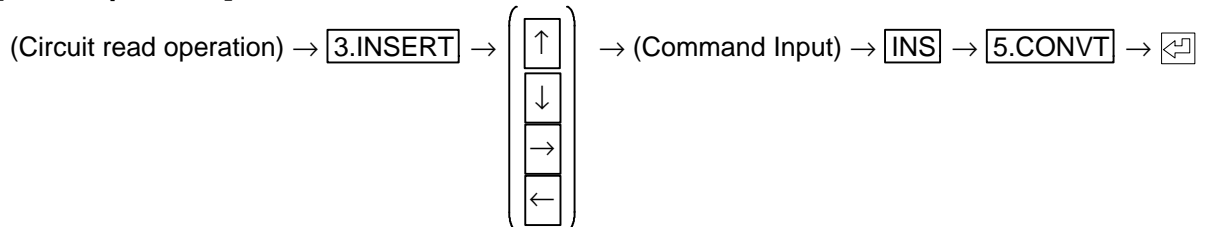
[Operation procedure]

(1) Using the circuit read operation display the block to where the circuit symbol is to be inserted.

(2) Press $\boxed{\text{3.INSERT}}$ and all circuit blocks other than the one at the cursor position will disappear.

- To insert a circuit symbol into a circuit block other than the one displayed at the screen top, move the cursor to the beginning of the target block and then press $\boxed{\text{3.INSERT}}$. The designated circuit block will move to the top of the screen, and all other blocks will disappear.

(3) Move the cursor to the position where the symbol is to be inserted using the cursor keys, and then input the command.

   (Note) Inputting the data at the top of the screen may not be possible depending on the system. An operation error will occur if this is attempted in such a system, and "OPERATING ERROR" will display in the message display section.

(4) Always press $\boxed{\text{5.CONVT}}$, $\boxed{\hookleftarrow}$ after inserting the circuit.

- "COMPLETED" will display in the message display section when $\boxed{\hookleftarrow}$ is pressed and the updated circuit will display.

> **POINT**
>
> (1) When insertion or addition is performed, the step numbers of the succeeding program and the CJ command jump destinations will also be changed automatically.

(Example) Read step No. 18 and insert contact B with device No. X8 as an AND circuit after the contact with device No. Y15.

$\boxed{\text{2.CIRCUT}}$ → $\boxed{\text{1.READ}}$ → $\boxed{\text{9.SET}}$ → $\boxed{1}$ → $\boxed{8}$ → $\boxed{\hookleftarrow}$ → $\boxed{\text{INS}}$

Read step No.18

$\boxed{\text{3.INSERT}}$ → $\boxed{\rightarrow}$ → $\boxed{2. \dashv\!\!/\!\!\vdash}$ → $\boxed{X}$ → $\boxed{8}$ → $\boxed{\hookleftarrow}$ → $\boxed{\text{INS}}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{\hookleftarrow}$

Insert contact B with device No. X8 after Y15
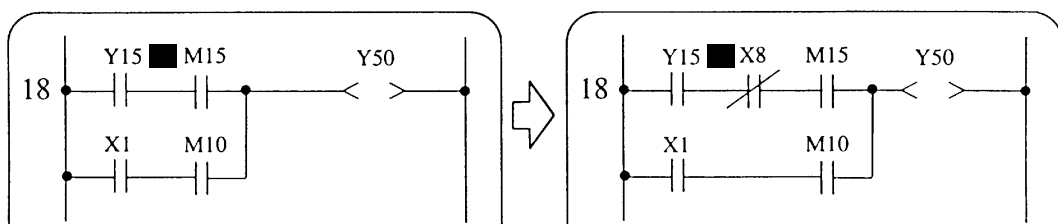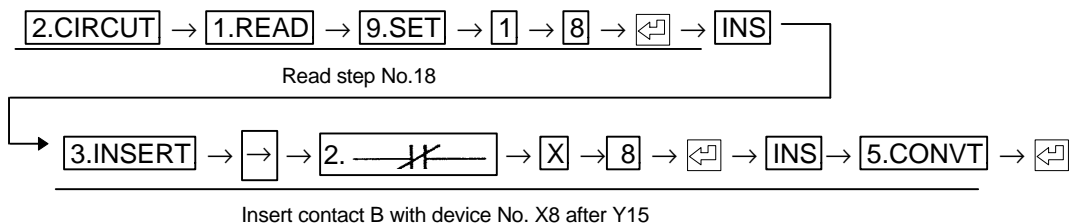
Examples of insertion and addition of other circuit symbols are shown below.

   (1) Inserting a contact

      (a) $\boxed{\text{3.INSERT}} \rightarrow \boxed{\rightarrow} \rightarrow \boxed{\text{1.}\,\dashv\vdash} \rightarrow \boxed{\text{X}} \rightarrow \boxed{\text{1}} \rightarrow \boxed{\text{0}} \rightarrow \boxed{\hookleftarrow}$

      (b) $\boxed{\text{3.INSERT}} \rightarrow \boxed{\text{3.}\,\llcorner\!\dashv\vdash\!\lrcorner} \rightarrow \boxed{\text{X}} \rightarrow \boxed{\text{1}} \rightarrow \boxed{\text{0}} \rightarrow \boxed{\hookleftarrow}$
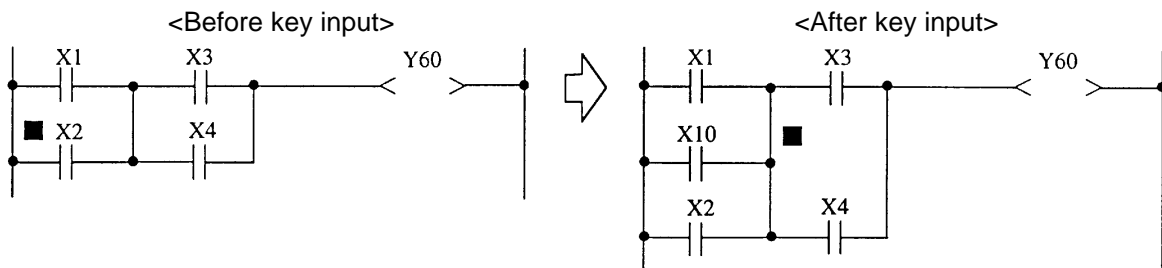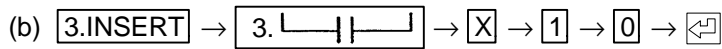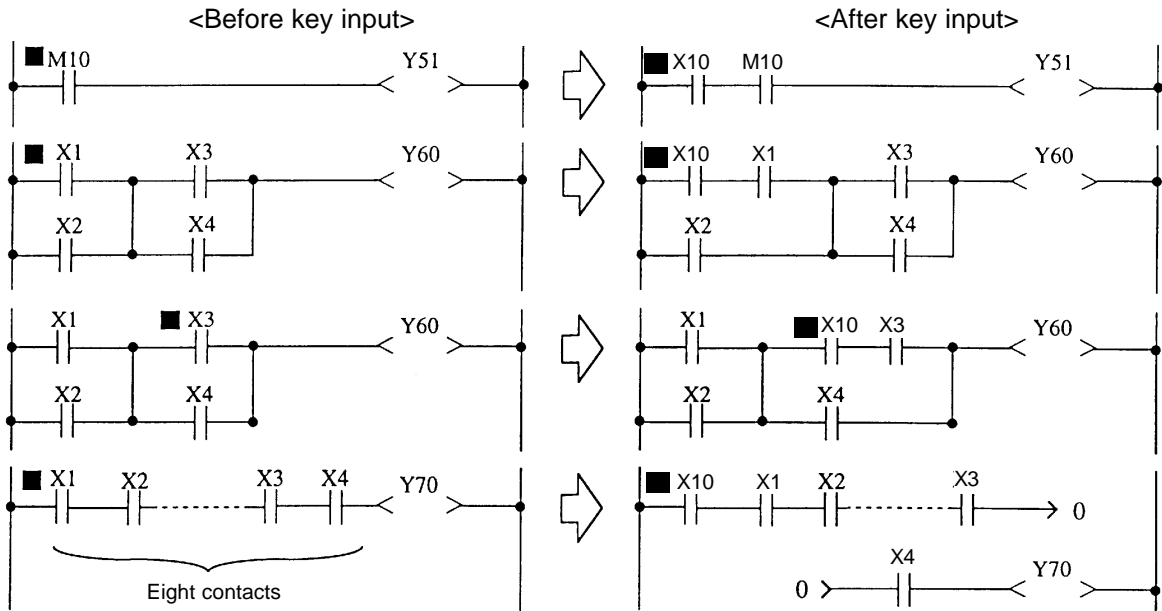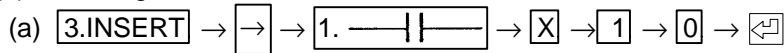
   (2) Inserting a horizontal bar

        $\boxed{\text{3.INSERT}} \rightarrow \boxed{\text{8.}\,\text{——}} \rightarrow \boxed{\hookleftarrow}$

– 32 –
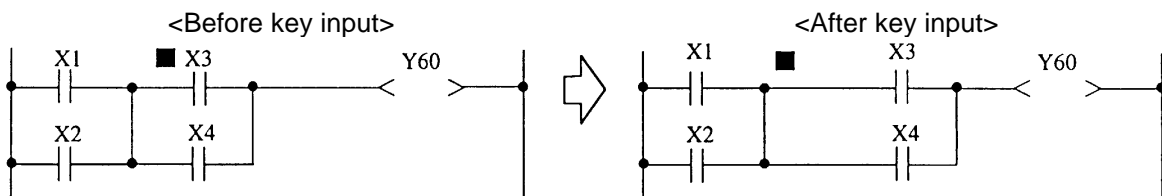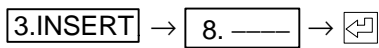
(3)   Inserting a vertical bar

$\boxed{\text{3.INSERT}} \rightarrow \boxed{\text{7. } |} \rightarrow \boxed{\text{↵}}$

<After key input>   <Before key input>
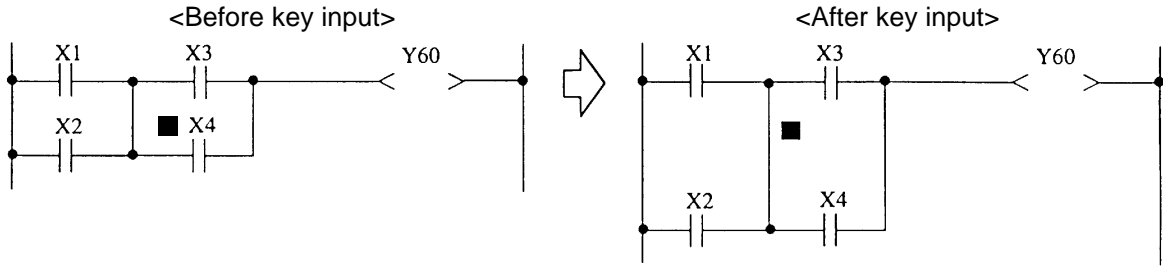
6.6 Deleting a Circuit

6.6.1 Deleting a circuit block

This operation is used to delete the sequence circuits one at a time.

[Basic operation]

(Circuit read operation) → 4.DELETE → | 5. —< >— | → ⏎
| 6. —[ ]— |

[Operation procedure]

(1) Display the circuit to be deleted with the read operation.

(2) Press 4.DELETE and all circuit blocks other than the one at the cursor position will disappear.

  • To delete a circuit block other than the one displayed at the screen top, move the cursor to the beginning of the target block and then press 4.DELETE . The designated circuit block will move to the top of the screen.

(3) Press | 5. —< >— | or | 6. —[ ]— | . "DELETE 1-CIRCUIT" will display in the message display section. Press the key again to delete the designated block.

  • Even if the target circuit block contains the commands for the output device, the block can be deleted by pressing 4.DELETE , | 5. —< >— | or | 6. —[ ]— | .

Note: 5.CONVT , ⏎ do not need to be pressed after deleting the circuit block.

POINT

(1) When deletion is performed, the step numbers of the succeeding program and the CJ command jump destinations will also be changed automatically.

(Example) Read step No. 25 and delete the circuit block containing it.

2.CIRCUT → 1.READ → 9.SET → 2 → 5 → ⏎ → INS → 4.DELETE → | 5. —< >— | → ⏎
Read step No.25   Delete circuit block

Read step No.25

Display after pressing

4.DELETE | 5. —< >— | ⏎

– 34 –
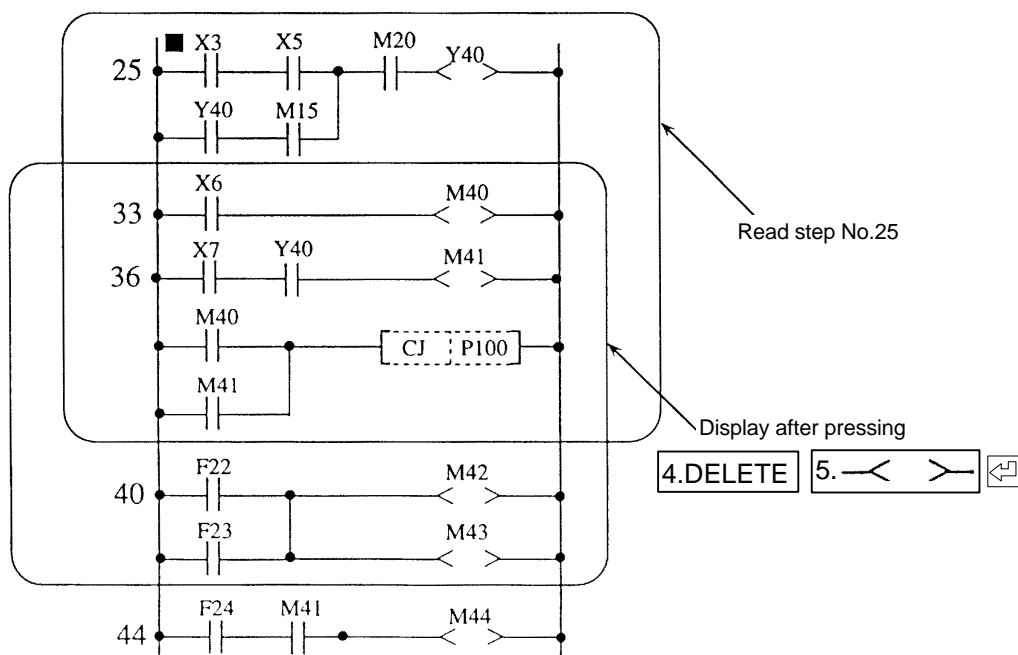
(Example) To delete the faulty circuit if a "LADDER ERROR", etc., occurs when `5.CONVT` , ⏎ are pressed after creating the circuit, or to delete a circuit block before pressing `5.CONVT` , ⏎ .

Circuit where "LADDER ERROR" has occurred. (Step numbers are not assigned.)

(Circuit edit operation) → `5.CONVT` → "LADDER ERROR"

LADDER ERROR occurs in the Y25 circuit block

→ `↓` → `INS` `4.DELETE` → `5. —< >—` → "DELETE 1-CIRCUIT (DISP)" → ⏎ → `INS` → `5.CONVT` → ⏎

Move the cursor down

Delete Y25 circuit block

Convert the Y26 circuit block
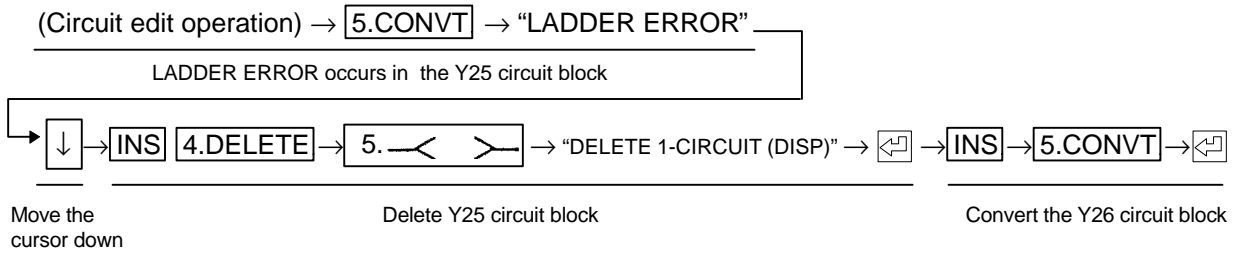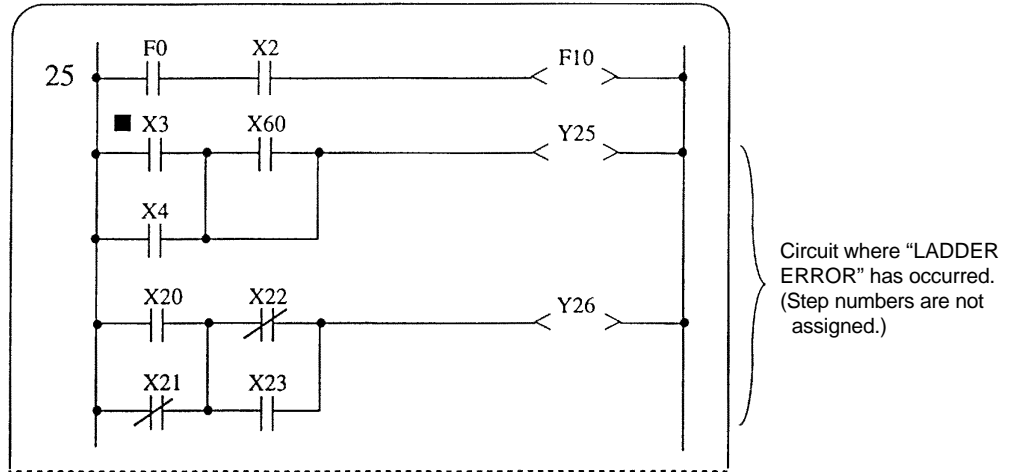
(Note) "DELETE 1-CIRCUIT" will display in the message display section when a normal circuit block is deleted. "DELETE 1-CIRCUIT (DISP)" will display when a circuit block with no step number assigned is deleted.

6.6.2 Deleting a circuit symbol

This operation is used to delete a sequence circuit in circuit symbol units.

[Basic operation]

(Circuit read operation) → | 4.DELETE | → ┃ ↑ ┃ → ┃ 1. ─┤├─ ┃ → ↵ → 5.CONVT → ↵
                                          ┃ ↓ ┃   ┃ 2. ─┤/├─ ┃
                                          ┃ ← ┃   ┃ 3. └┤├┘ ┃
                                          ┃ → ┃   ┃ 4. └┤/├┘ ┃
                                                   ┃ 7. ─── ┃
                                                   ┃ 8. │ ┃
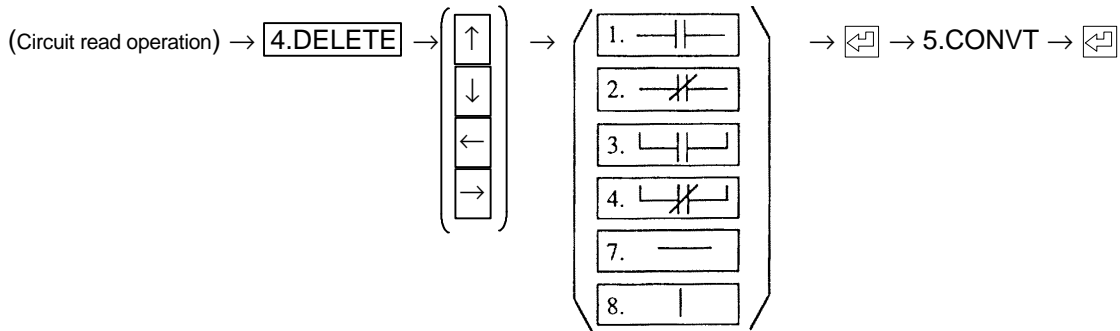
[Operation procedure]

(1) Display the circuit block with the circuit symbol to be deleted with the read operation.

(2) Press | 4.DELETE | and all circuit blocks other than the one at the cursor position will disappear.
   - To delete a circuit block other than the one displayed at the screen top, move the cursor to the beginning of the target block and then press | 4.DELETE |. The designated circuit block will move to the top of the screen.

(3) Move the cursor to the circuit symbol to be deleted with the cursor keys, and press | 1. ─┤├─ | , | 2. ─┤/├─ | , | 3. └┤├┘ | , | 4. └┤/├┘ | , | 7. ─ | or | 8. │ | and then press ↵ to delete it.

(4) Always press | 5.CONVT | , ↵ after deleting the circuit.
   - "COMPLETED" will display in the message display section when ↵ is pressed and the updated circuit will display.

> **POINT**
>
> (1) When deletion in circuit symbol units is performed, the step numbers of the succeeding program and the CJ command jump destinations will also be changed automatically.

(Example) Read step No. 12 and delete X3.

| 2.CIRCUT | → | 1.READ | → | 9.SET | → | 1 | → | 2 | → ↵ → | INS | ┐
                Read step No. 12

└→ | 4.DELETE | → | → | → | ↓ | → | 3. └┤├┘ | → ↵ → | INS | | 5.CONVT | → ↵
              Delete X3
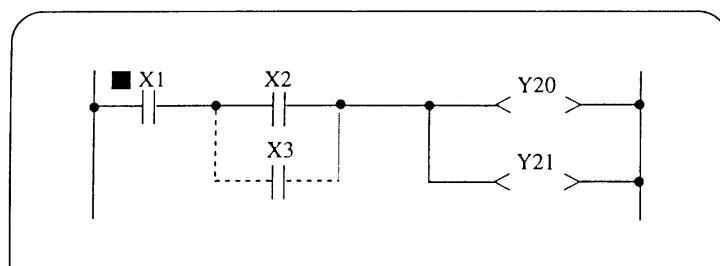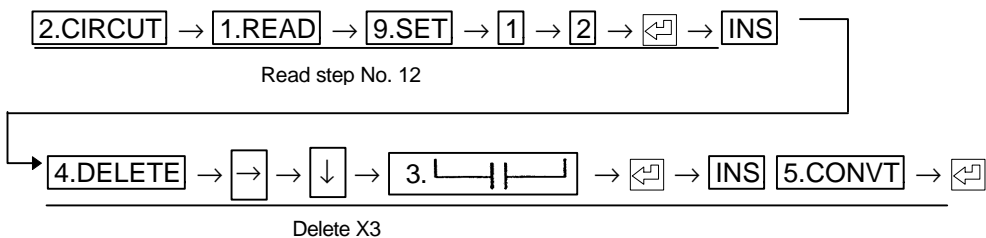
Examples of deletion of other circuit symbols are shown below.

(1) Deleting a contact

(a) 4.DELETE → → → → → ↵ or 4.DELETE → → → →
```
7. ──
8. │
```
→ ↵

<Before key input>

■X1   X2   X3   Y80
  ─┤├──●──┤├──┤├──●──< >
            X4
          ──┤├──

<After key input>

X1   X2   ■ Y80
─┤├──●──┤├─────< >
       X4
     ──┤├──

(Note) 5.CONVT is invalid in this circuit.

(b) 4.DELETE → → → ↓ → 3.─┤├─ → ↵

<Before key input>

■X1   X2   Y80
  ─┤├──●──┤├──●──< >
        X3
      ──┤├──

<After key input>

X1   X2   Y80
─┤├──┤├────────< >
      ■

(c) 4.DELETE → → → ↓ → 3.─┤├─ → ↵

<Before key input>

■X1   X2   X3   Y80
  ─┤├──●──┤├──┤├──●──< >
        X4
      ──┤├──

<After key input>

X1   X2   X3   Y80
─┤├──┤├──┤├──────< >
            ■
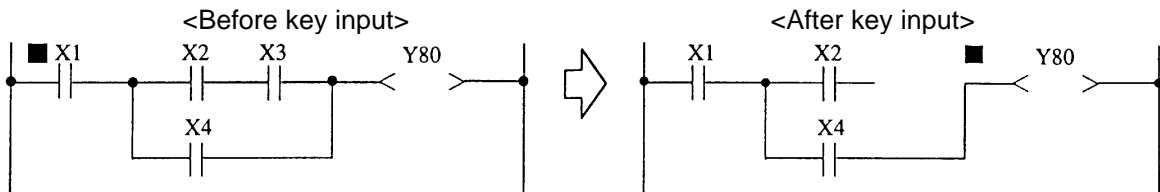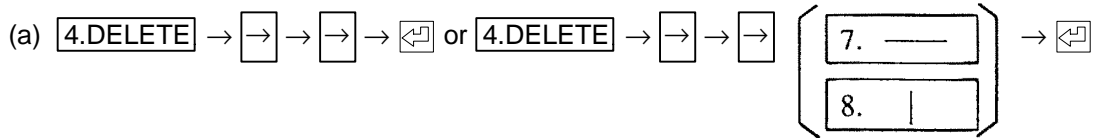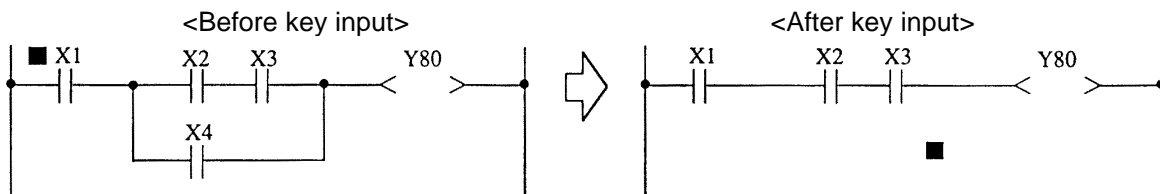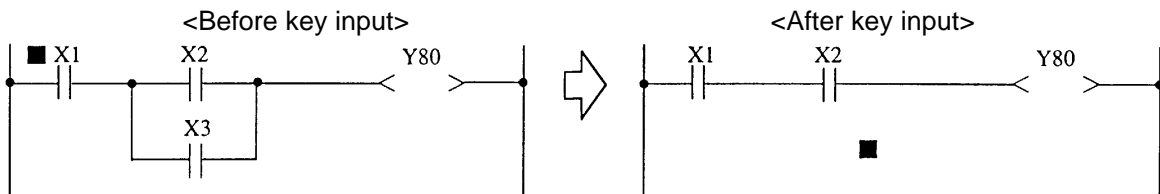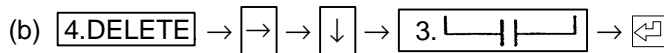
(2) Deleting a vertical line

(a) 4.DELETE → → → 7. │ → ↵

<Before key input>

■X1   X2   X3   Y80
  ─┤├──●──┤├──┤├──●──< >
        X4   X5
      ──┤├──┤├──

<After key input>

X1 ■ X2   X3   Y80
─┤├────┤├──┤├──●──< >
        X4   X5
      ──┤├──┤├──
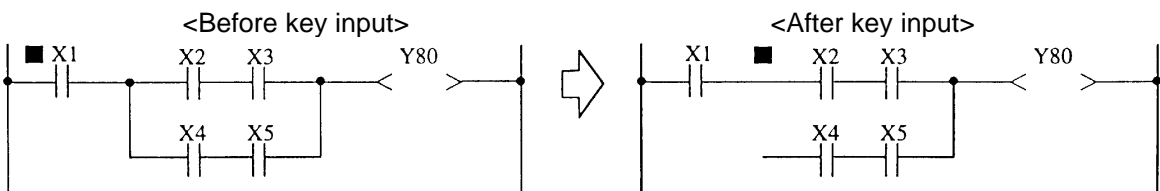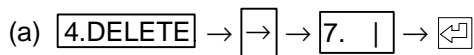
(Note) 5.CONVT is invalid in this circuit.

6.7 Circuit Extension Function

The circuit extension function is used to create a ladder circuit that is wider than eight contacts and one coil (equivalent to nine contacts.) The circuit extension function operation method and specifications are explained below.

CAUTION

(1) When 2.WRITE is pressed, a maximum of eight returns can be created if there is

one extended line. When 3.INSERT is pressed a maximum of one return can be created.

The ladder circuit display buffers are 18 stages long and nine contacts (including a coil) wide. The section not displayed on the screen is displayed with the cursor keys ↑ ↓ .

9 stages × 2 buffers
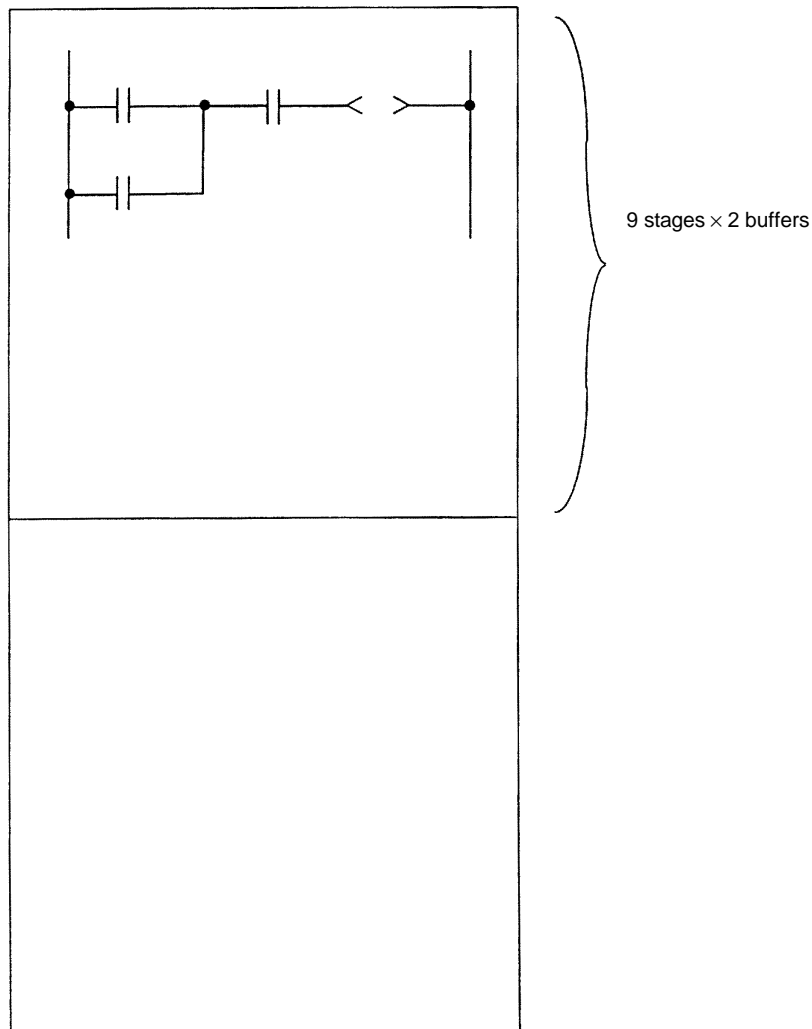
6.7.1 Examples of extension circuit operation

(1) Example of writing single return
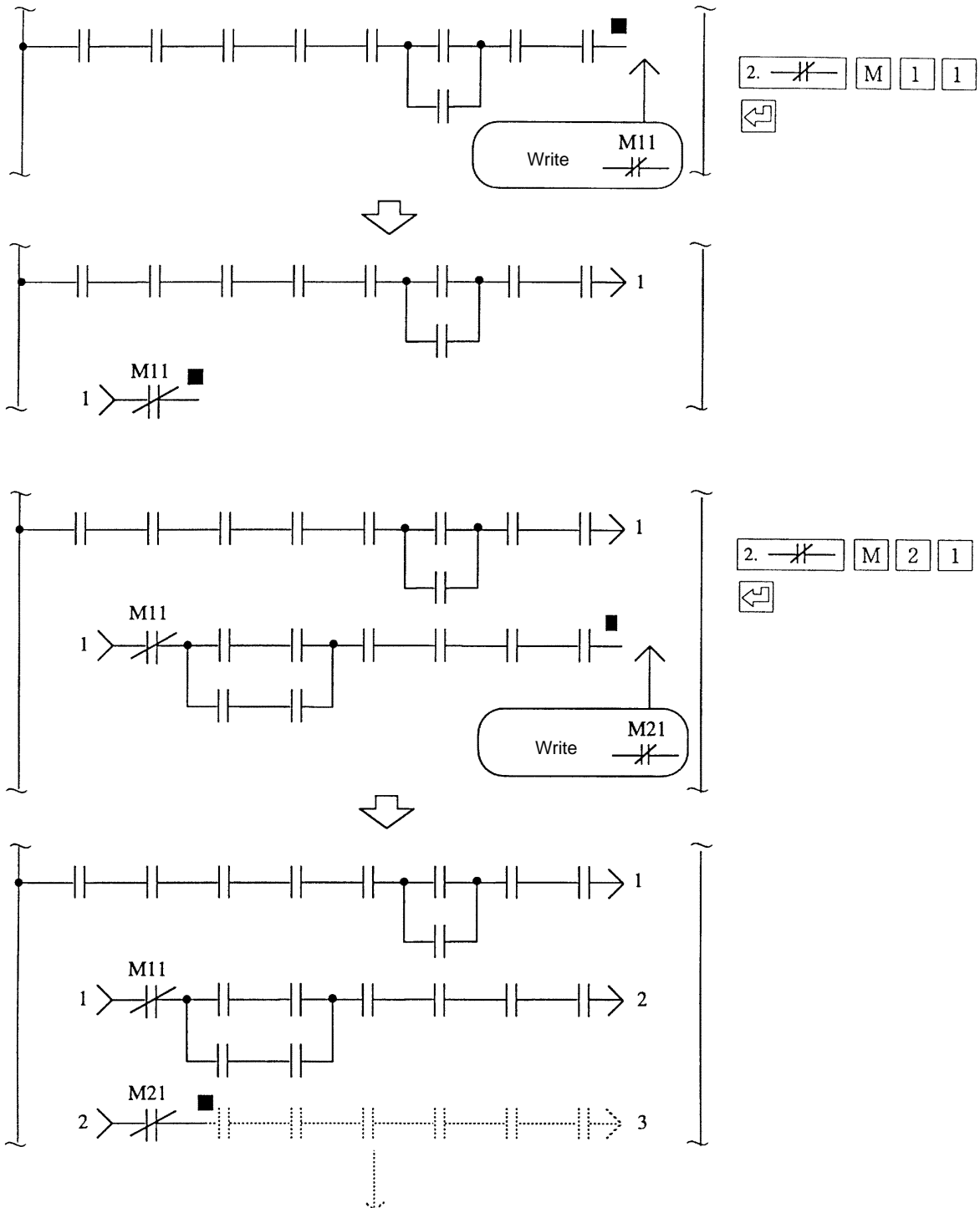
A maximum of six returns can be created if there is one extended line in the same manner.

• Circuit symbols that can be extended during writing :
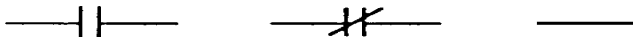
• Circuit symbols that cannot be extended during writing :

If any of these symbols are used, an "OPERATING ERROR" will occur when ⏎ is pressed.

– 39 –

(2) Example of writing two or more extended lines

Write   X1
    —||—

1. —||— X 1
⏎

Write   X2
    —||—

X 2 ⏎

Write │ (vertical line)

↑ 7. │ ⏎
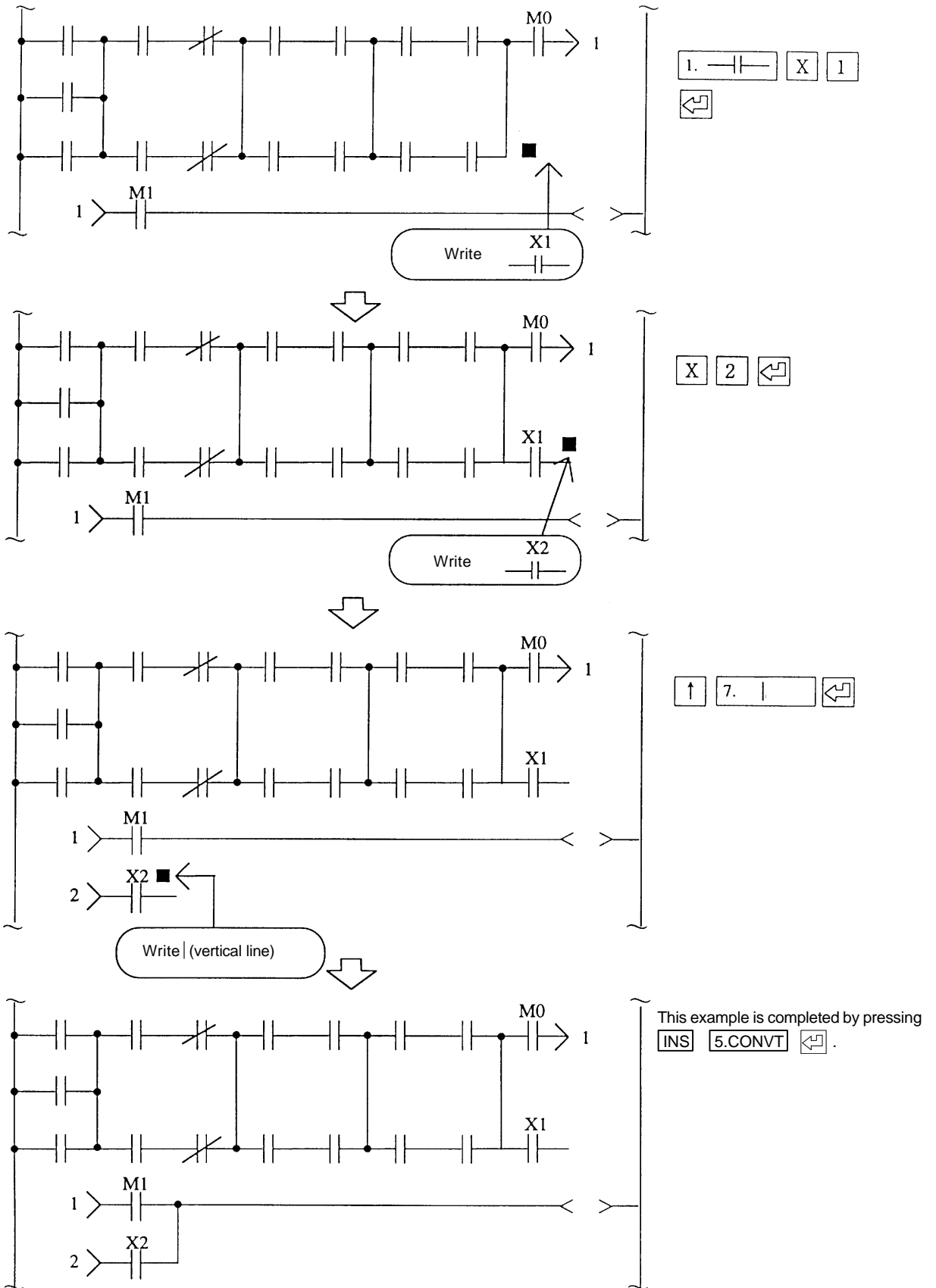
This example is completed by pressing
INS 5.CONVT ⏎ .

(3) Example of writing function command into return section

Write – (horizontal line)

8. ⟶ [↵]

6. –[ ]– [M][O][V]

[SP][K][1][0][SP]

[D][1][2][3]

Write –[MOV ⋮ K10 ⋮ D123]–

This example is completed by pressing [INS] [5.CONVT] [↵] .
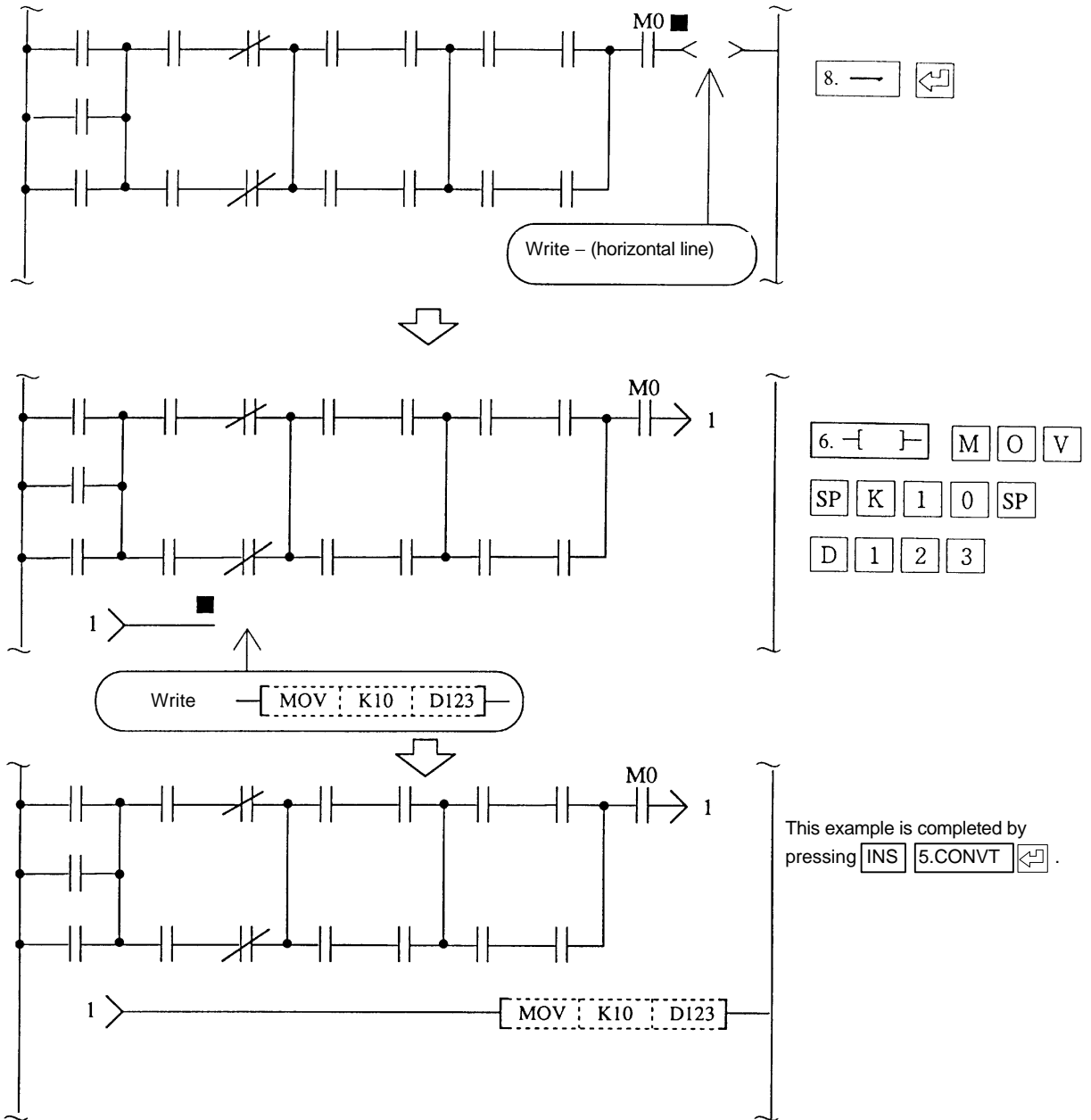
When writing a function command into the return section, extend with a – (horizontal line) first, and then write in the function command. An "OPERATING ERROR" will occur if the function command is written in directly.

(4) Example 1 of inserting a single extended line
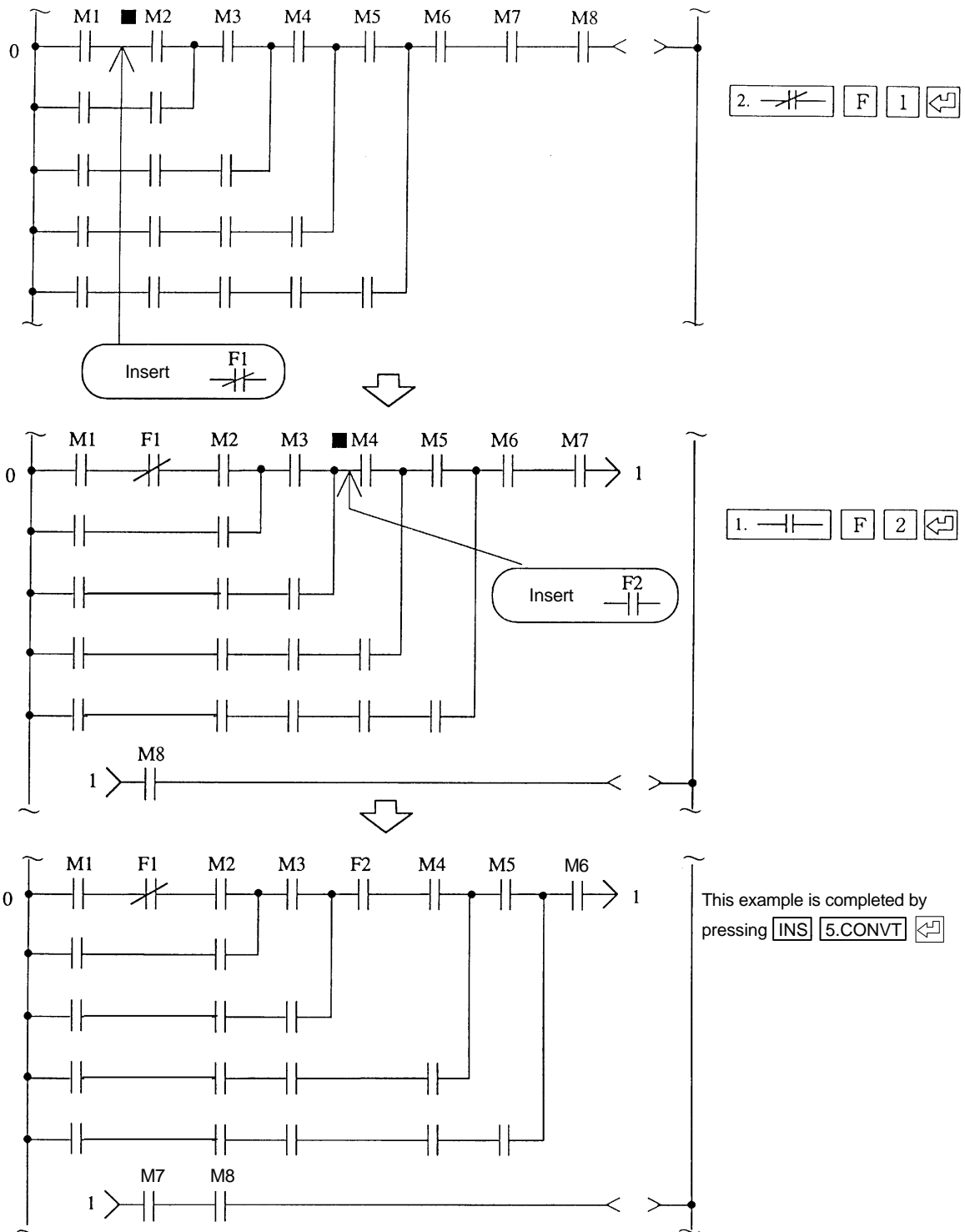
- The insertion function inserts a contact —||—   —|/|—   —— etc., at the cursor position.
  If nine contacts (including a coil) are exceeded, the line will be extended as shown above.
  However, only one return can be created with the insertion operation.
- Circuit symbols that can be extended during insertion :
  —||—   —|/|—   ——

- Circuit symbols that cannot be extended during insertion :
  └||┘   └|/|┘ | —< >—   —{ }—

  If any of these symbols are used, an "OPERATING ERROR" will occur when ⏎ is pressed.

(5) Example 2 of inserting a single extended line (when more than one coil exists)

Insert   F3

2. ─╫─   F 3 ↵
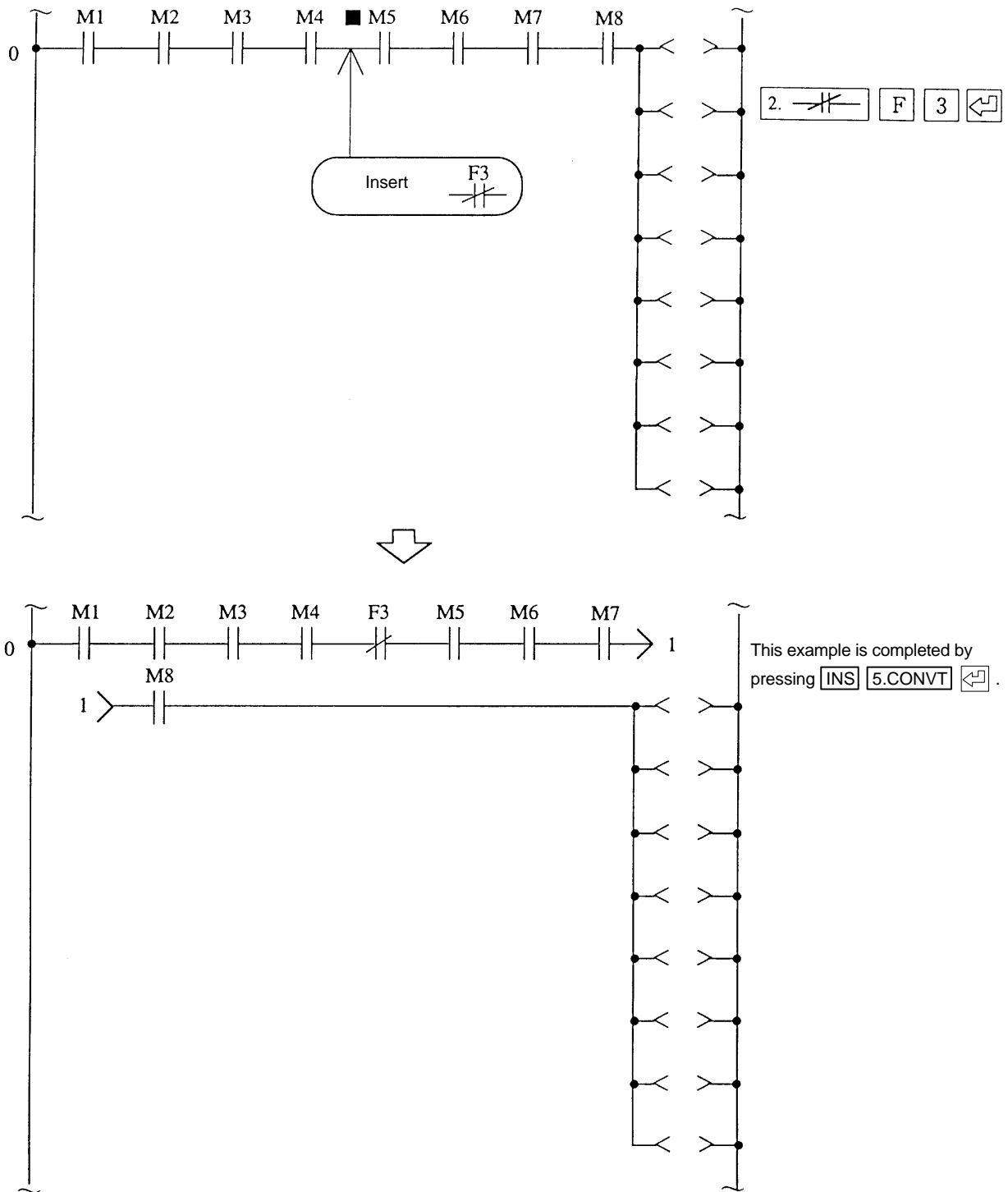
This example is completed by pressing INS 5.CONVT ↵ .

(6) Example of inserting when a function command exists in the coil section
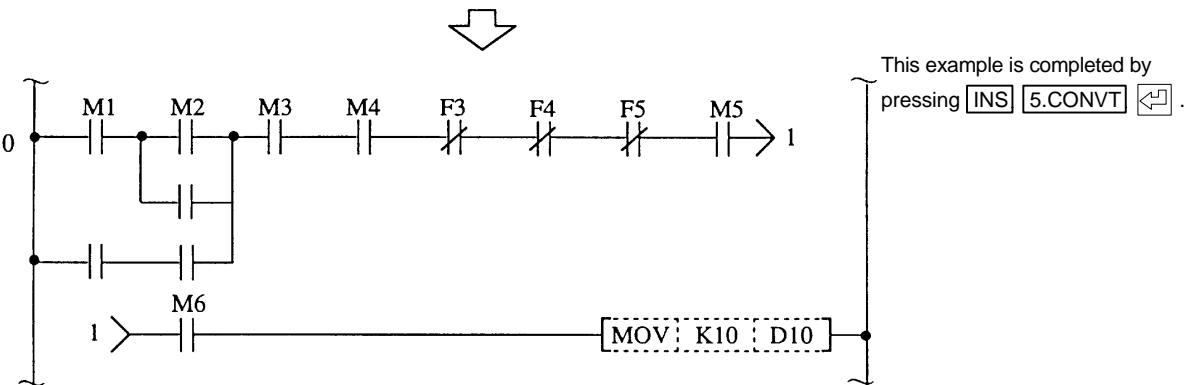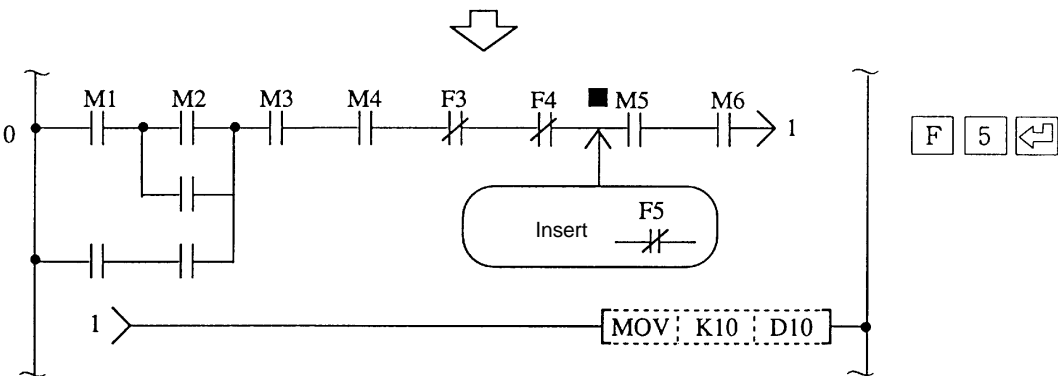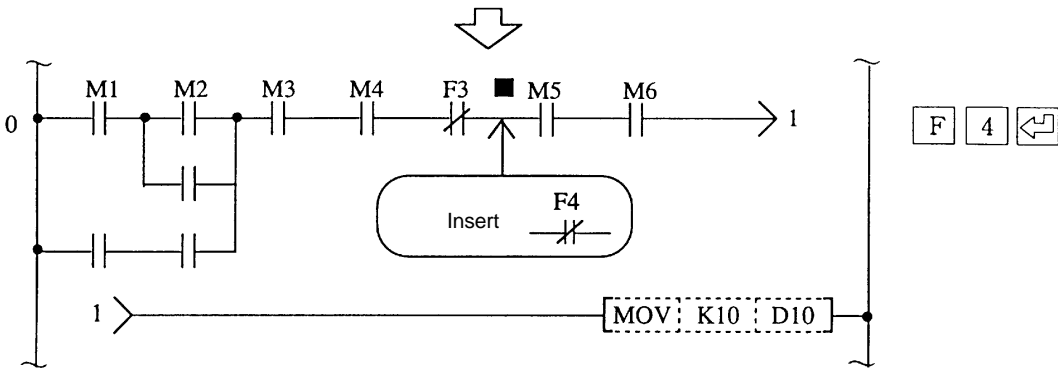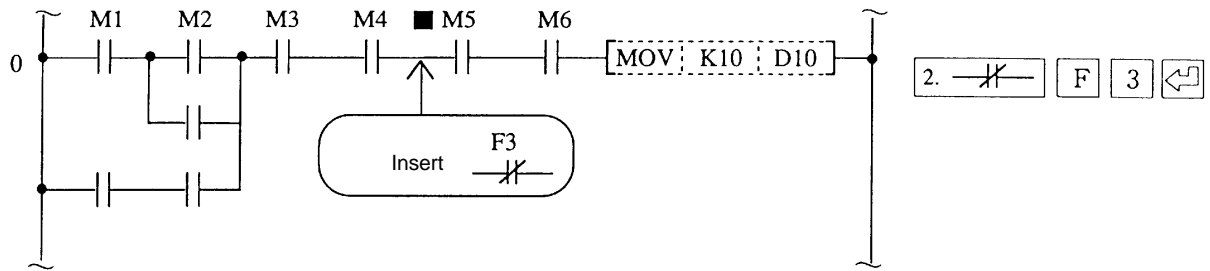
Note) Insertion is not possible if a function command handled as a contact such as =, >, or < exists in the return section. An "OPERATING ERROR" will occur if insertion is attempted.
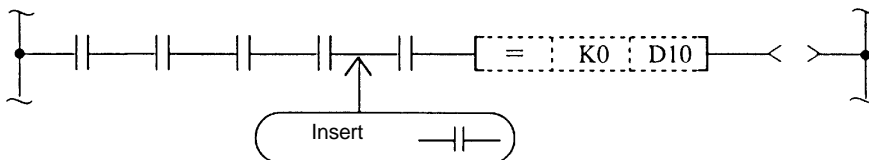
(7) Example 1 of inserting when there are two or more extended lines
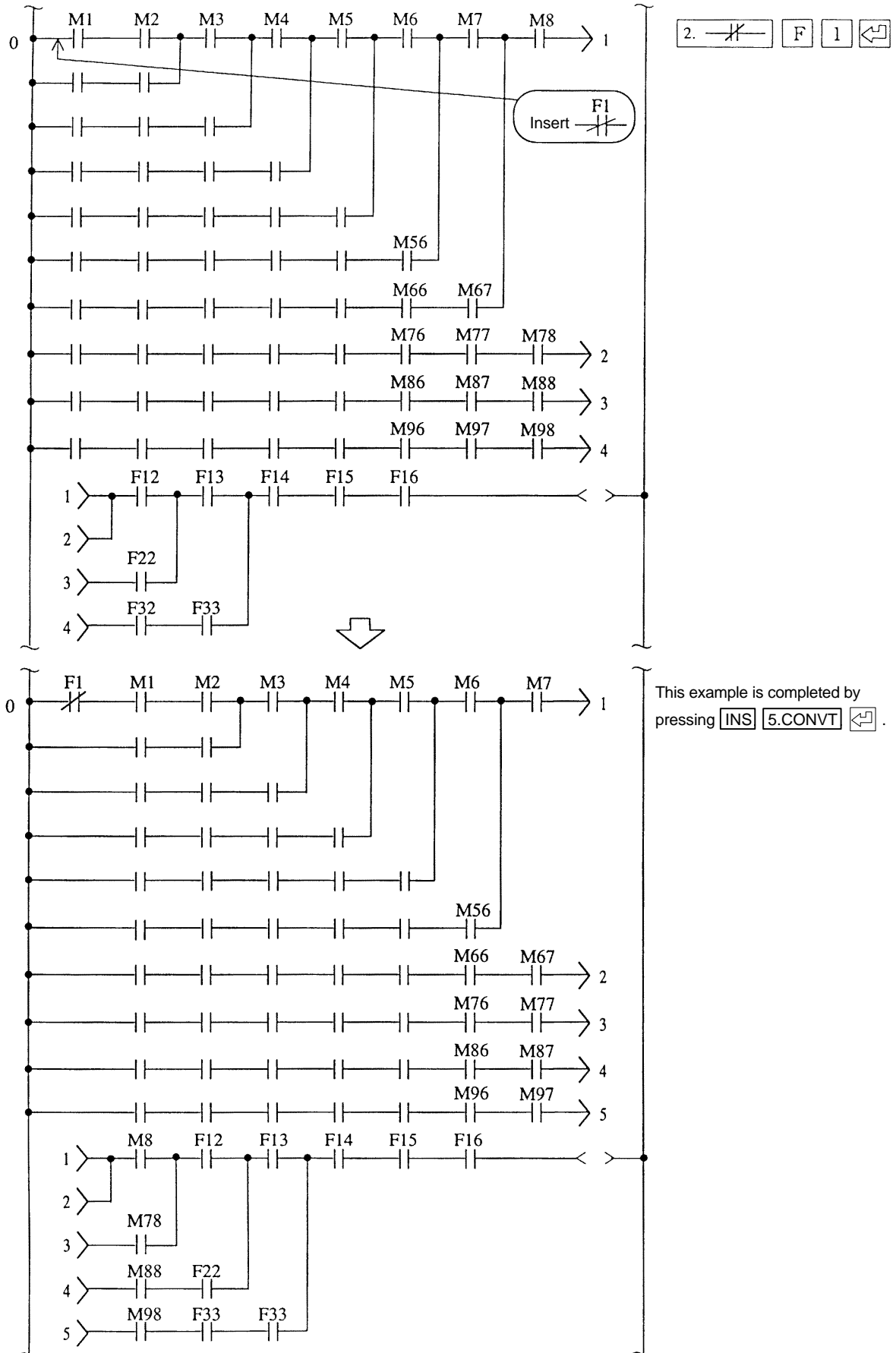
(8) Example 2 of inserting when there are two or more extended lines
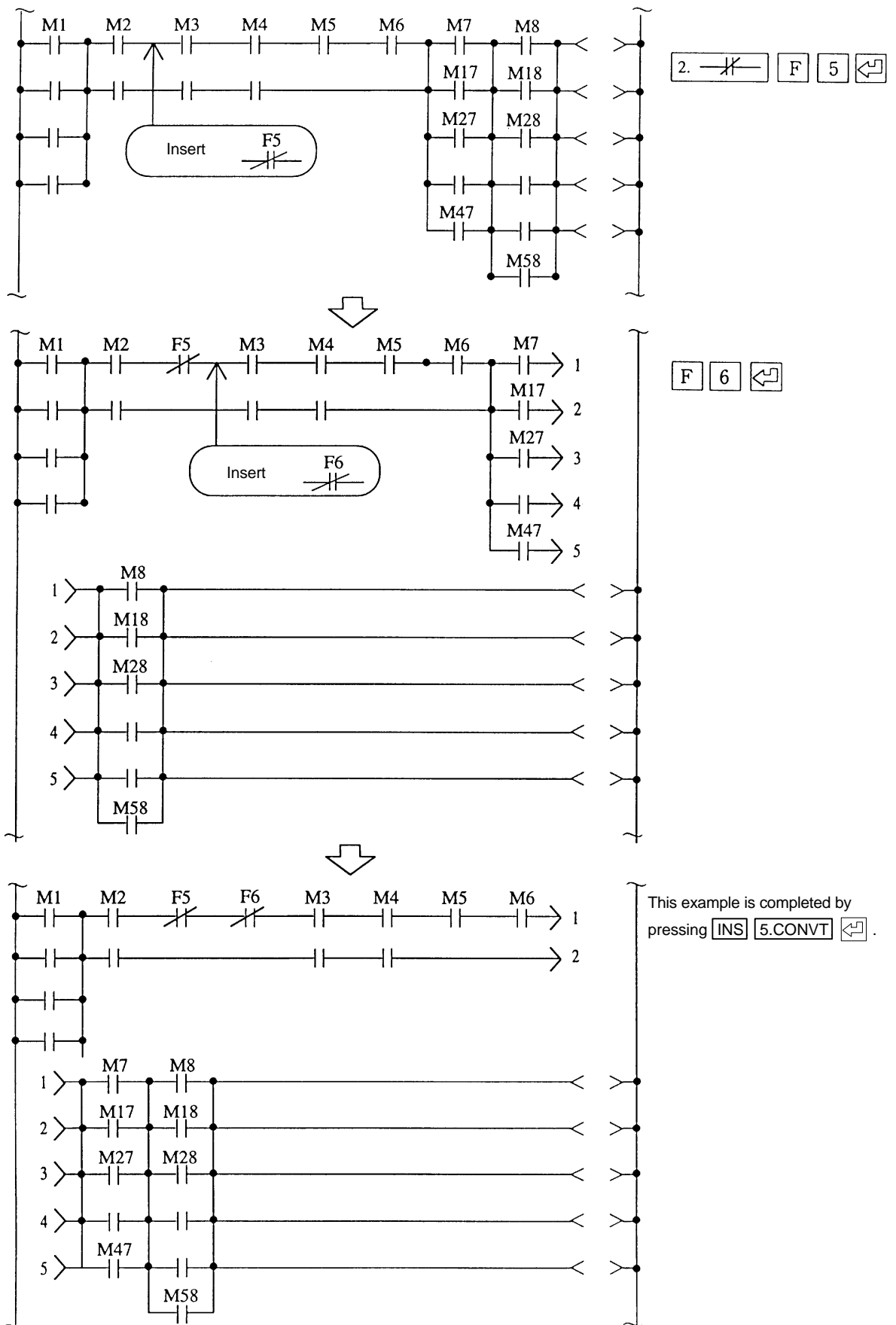
6.7.2 Error messages
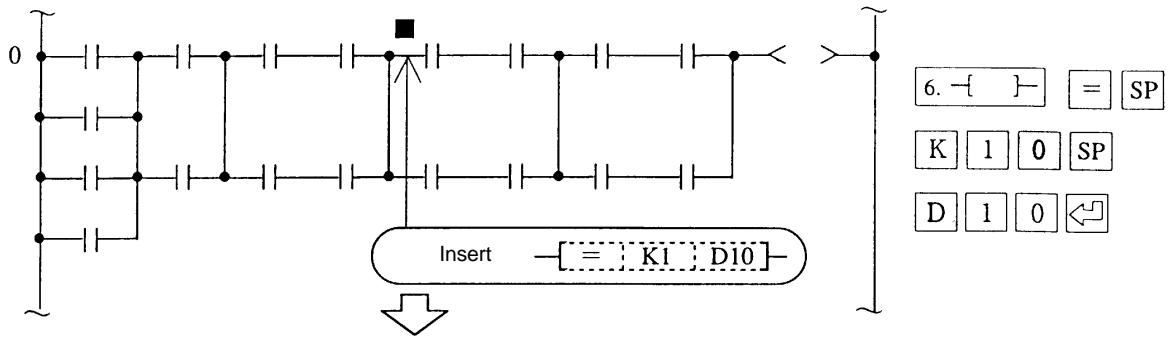
An "OPERATING ERROR" occurs.
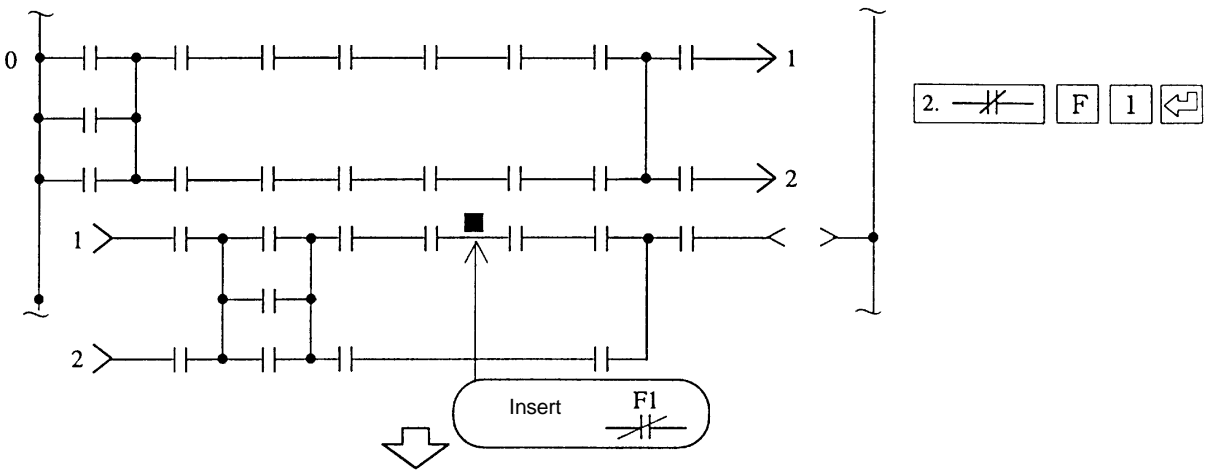
A "CIRCUIT CONTINUATION ERROR" or "OPERATING ERROR" occurs.
Only one return can be created when two or more lines are extended.
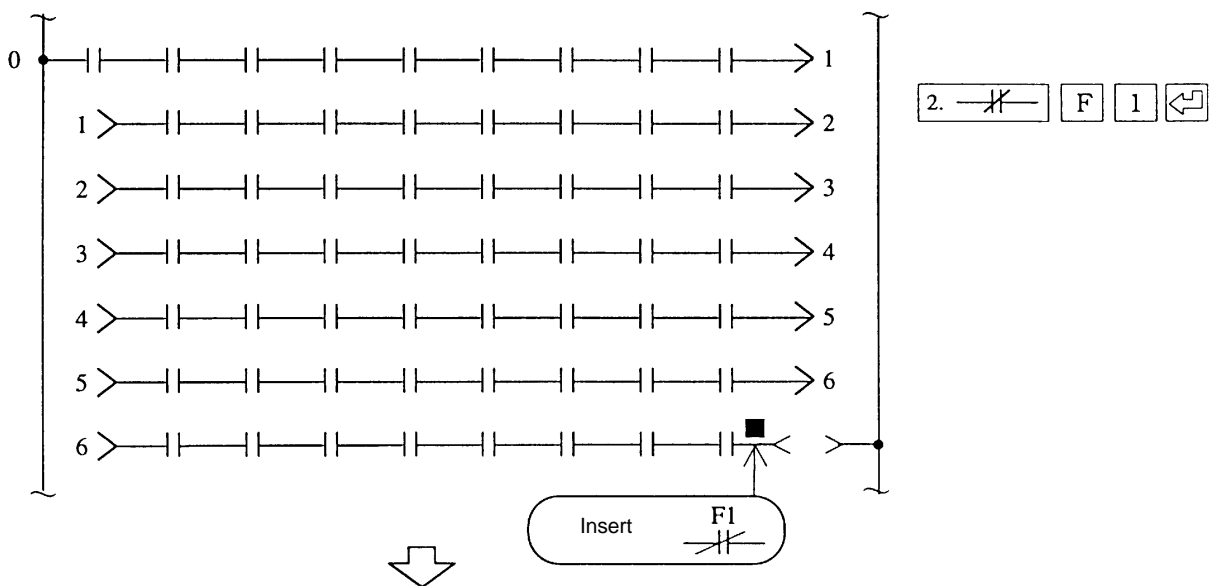
"CIRCUIT CONTINUATIVE SIZE OVER" occurs. The maximum number of returns is six.
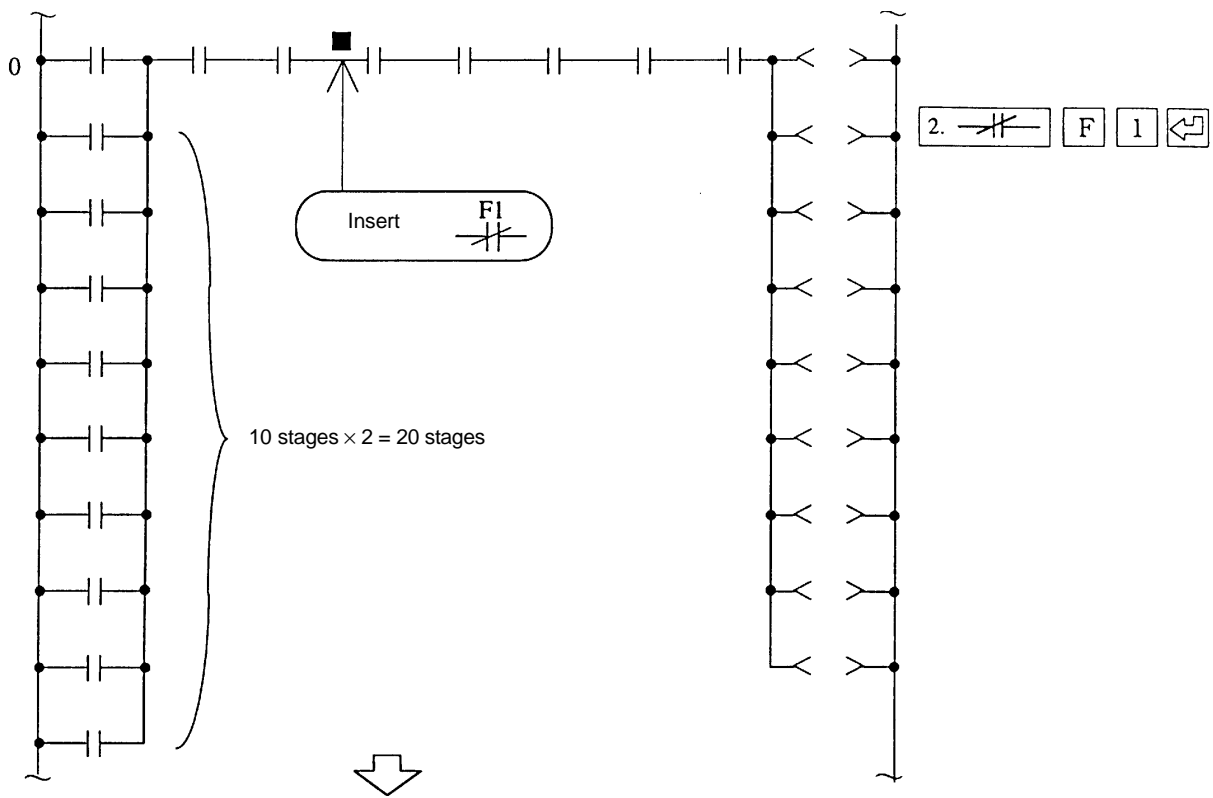
"LADDER OVERFLOW" occurs. The maximum number of stages after extension is 18.

A "CIRCUIT CONTINUATION ERROR" will occur if during conversion if a vertical branch exists at the return start position.
Program as shown below:

A vertical branch is possible at this position.
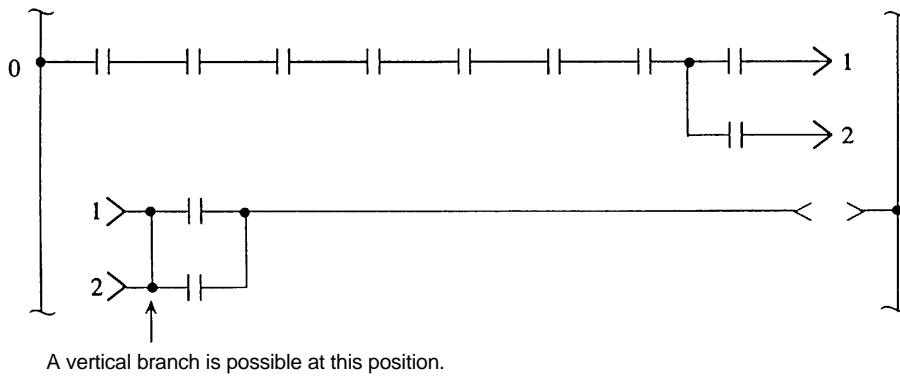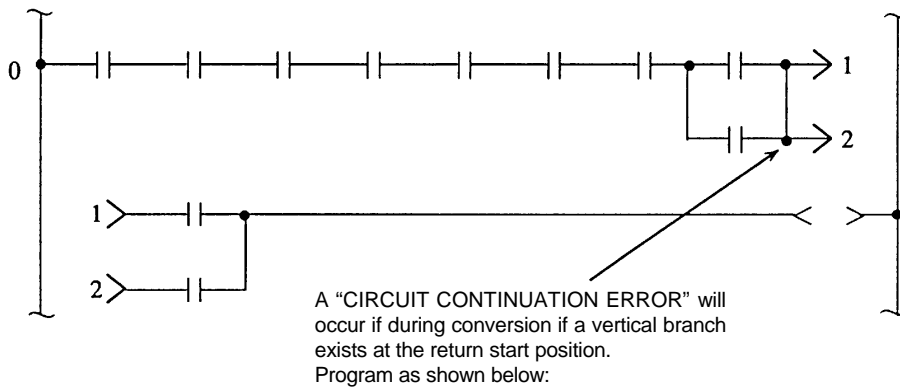
6.7.3 Relation of number of returns and circuit length

The relation between the number of returns and the maximum length of the circuit that can be created is as shown below.

Number of returns	0	1	2	3	4	5	6	7	8
Circuit length (stages)	18	11	7	5	4	3	3	2	2

A "LADDER ERROR" will occur if the limits are exceeded.

Example)

The circuit length is up to seven lines when there are two returns.

CAUTION
Max. 18 stages
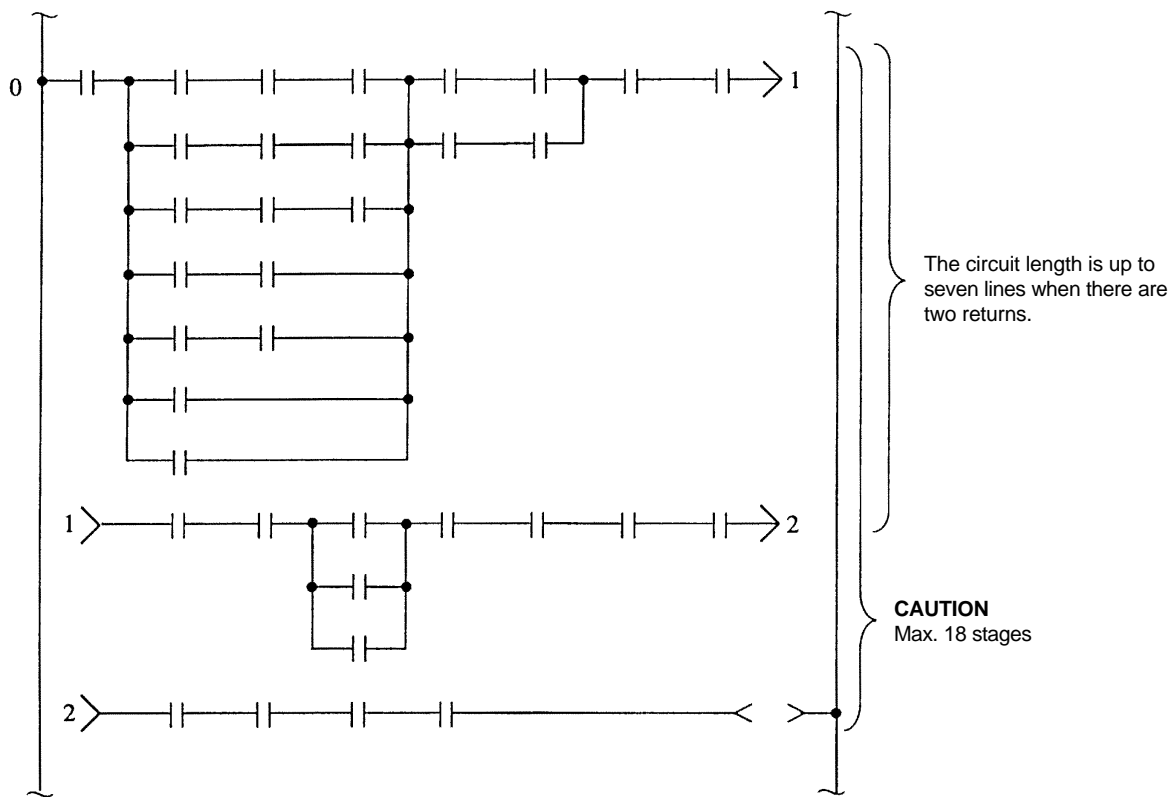
7. Messages

Sequence programs as well as message data used for the CNC message function and the ladder circuit printing function can be created with the PLC development software. The messages that can be created are as follow:

(1) Alarm message (ALARM MESS)
(2) Operation message (OPERATE MESS)
(3) PLC switch name(PLC SW MESS)
(4) Device message (DEVICE MESS)
(5) Coil message (COIL MESS)
(6) Comment message (COMMENT MESS)

The functions of each message are explained below.

(1) ALARM MESS

The alarm messages displayed on the CNC DIAGNOSIS screen are created as character strings. Alphanumeric, kana and kanji characters can be used. Two half-byte characters are used to display one kanji character.

The maximum length of an ALARM MESS is 32 characters, and a maximum of 256 messages can be registered. However, the limits will depend on the CNC memory size.

(2) OPERATE MESS

The operator messages displayed on the CNC DIAGNOSIS screen are created as character strings.

Alphanumeric, kana and kanji characters can be used. Two half-byte characters are used to display one kanji character.

The maximum length of an OPERATE MESS is 60 characters, and a maximum of 256 messages can be registered. However, the limits will depend on the CNC memory size.

(3) PLC SW MESS

The switch names displayed on the CNC DIAGNOSIS screen are created as character strings. Alphanumeric, kana and kanji characters can be used. Two half-byte characters are used to display one kanji character.

The maximum length of an PLC SW MESS is 14 characters, and a maximum of 32 switches can be registered. However, the limits will depend on the CNC memory size.

(4) DEVICE MESS

Create a contact name to be printed with the device numbers during printing of the ladder circuit. The names are created as character strings.
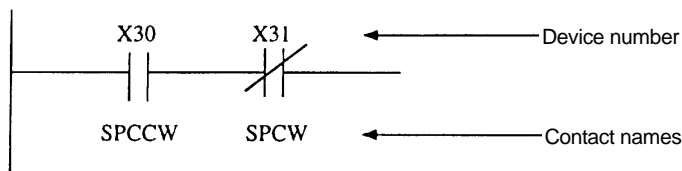
Alphanumeric, kana and kanji characters can be used. Two half-byte characters are used to display one kanji character.

The maximum length of a DEVICE MESS is 6 characters, and a maximum of 1000 device names can be registered. However, the limits will depend on the CNC memory size.

(5) COIL MESS

Create a coil name to be printed with the device numbers during printing of the ladder circuit. The names are created as character strings.
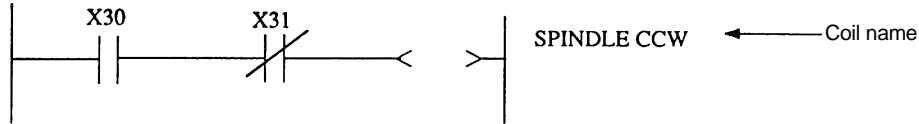
Alphanumeric, kana and kanji characters can be used. Two half-byte characters are used to display one kanji character.

The maximum length of a COIL MESS is 18 characters, and a maximum of 2700 coil names can be registered. However, the limits will depend on the CNC memory size.

(6) COMMENT MESS

Create a tool name (8 alphanumeric characters × 5) to display on the CNC TOOL REGISTRATION screen and a character string to display the load type and scale of the load meter display.

Alphanumeric, kana and kanji characters can be used. Two half-byte characters are used to display one kanji character.

The maximum length of a COMMENT MESS is 60 characters, and a maximum of 100 comments can be registered. However, the limits will depend on the CNC memory size.

Note) The number of comment characters will differ according to the application such as the tool name being 8 characters and the comment for the load meter display 60 characters. The number of characters will be determined by the longer comment. Thus the number of characters is normally 60.

CAUTION
Semi-graphics cannot be used for messages created on the personal computer. If semi-graphics are required for the load meter, etc., leave that line blank, and create a message including the semi-graphics with the CNC onboard. An odd number of characters must not be used for one message as the CNC may not function correctly.

7.1   Initial

Select the initial screen and set the message size before creating a new message. Messages cannot be created if this is not performed. The data length must always be an even number.

[Basic operation]

$\boxed{\text{1.MSSG}} \rightarrow \boxed{\text{7.INIT}} \rightarrow \boxed{\text{2.WRITE}} \rightarrow (\text{Set data size}) \rightarrow \boxed{\text{5.CONVT}} \rightarrow \boxed{\text{Y}} \rightarrow \boxed{\hookleftarrow}$

[Operation procedure]

(1)   Select $\boxed{\text{1.MSSG}}$, and then $\boxed{\text{7.INIT}}$.

(2)   Press $\boxed{\text{2.WRITE}}$ and the setting section will display.

(3)   Set the used size and data length for each message.

(4)   Press $\boxed{\text{5.CONVT}}$. The message "CHANGE SIZE (Y/N)" will appear, so press $\boxed{\text{Y}}$.
   "COMPLETED" will display when conversion is completed.

   Note)   Conversion is performed only when each message size is changed.

```
                          Index

                                                          PLC MESSAGE TABLE


                    USED SIZE        LENGTH
   1.ALARM MESS    (    0/    0)    (  32 BYTE )
   2.OPERATE MESS  (    0/    0)    (  60 BYTE )
   3.PLC SW MESS   (    0/    0)    (  14 BYTE )
   4.DEVICE MESS   (    0/    0)    (   6 BYTE )
   5.COIL MESS     (    0/    0)    (  18 BYTE )
   6.COMMENT MESS  (    0/    0)    (  60 BYTE )
   7.CONTROL CODE (0)
                                                          Index number

                                                          Number of messages

                                                          Data length

      # (    ) DATA (        ) (          BYTE)
                                                          Setting section

   MODE INIT MENU CONVT


      2.WRITE            5.CONVT         SUBMENU         MENUE
```
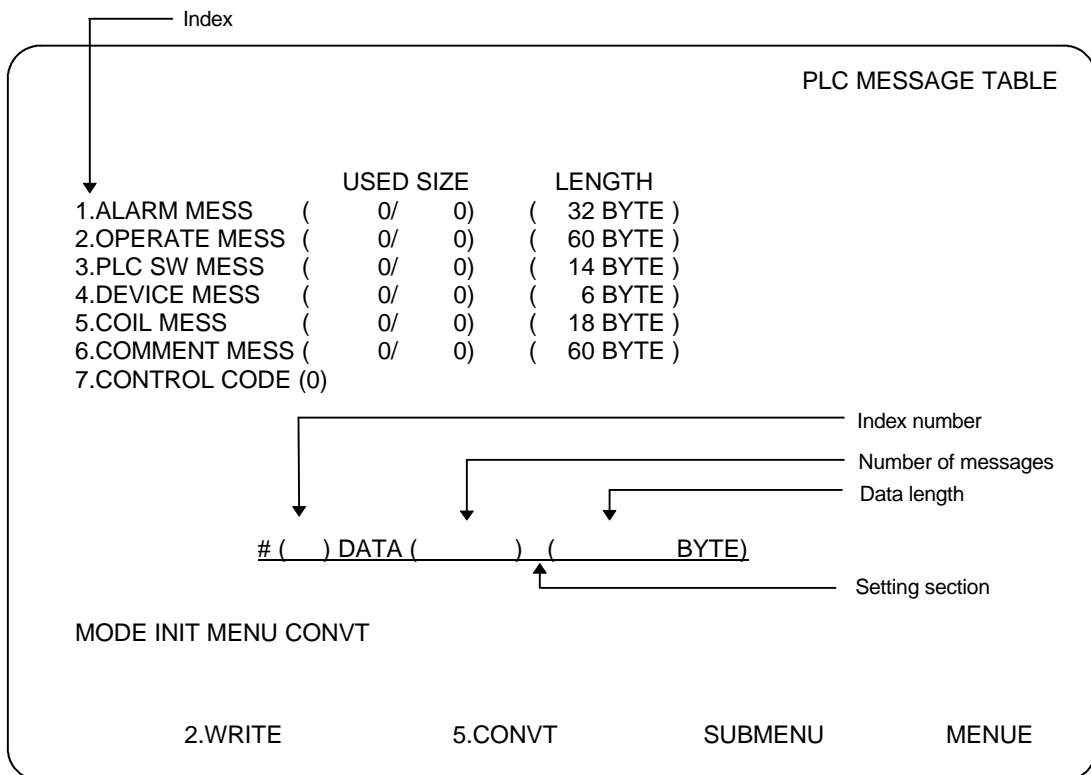
The maximum number of messages and maximum length of each message are shown below.

	Item	Max. number of messages	Max. data length	Remarks
1	ALARM MESS	256	32 bytes/message	The alarm message data memory size is set.
2	OPERATE MESS	256	60 bytes/message	The operator message data memory size is set.
3	PLC SW MESS	32	14 bytes/switch	The PLC switch data memory size is set. Normally set this to 14.
4	DEVICE MESS	1000	6 bytes/contact	The memory size of the ladder diagram contact comments such as SP CCW ┤├ is set.
5	COIL MESS	2700	18 bytes/coil	The memory size of ladder diagram coil comments such as Y100 SPINDLE ─[ ]─┤ CCW is set.
6	COMMENT MESS	100	60 bytes/comment	The memory size of comment data for tool registration, load meter, etc. is set. Up to 60 characters can be used for the load meter comment, so normally set this to 60.
7	CONTROL CODE	—	—	Not used. Set to 0.

7.1.1　Example of settings

```
                                              PLC MESSAGE TABLE

                   USED SIZE        LENGTH
     1.ALARM MESS    (     0/    0)  (   32 BYTE )
     2.OPERATE MESS  (     0/    0)  (   60 BYTE )
     3.PLC SW MESS   (     0/    0)  (   14 BYTE )
     4.DEVICE MESS   (     0/    0)  (    6 BYTE )
     5.COIL MESS     (     0/    0)  (   18 BYTE )
     6.COMMENT MESS  (     0/    0)  (   60 BYTE )
     7.CONTROL CODE  ( 0 )




              # (    ) DATA (         ) (           BYTE)


     MODE INIT MENU WRITE


              2.WRITE          5.CONVT        SUBMENU         MENUE
```

The above screen will appear when the 7.INIT menu
is selected, and the message file set with the edit file
is searched for but not found.

```
                                        PLC MESSAGE TABLE

                         USED SIZE      LENGTH
     1.ALARM MESS     (     0/    0)  (   32 BYTE )
     2.OPERATE MESS   (     0/    0)  (   60 BYTE )
     3.PLC SW MESS    (     0/    0)  (   14 BYTE )
     4.DEVICE MESS    (     0/    0)  (    6 BYTE )
     5.COIL MESS      (     0/    0)  (   18 BYTE )
     6.COMMENT MESS   (     0/    0)  (   60 BYTE )
     7.CONTROL CODE   ( 0 )
```

For example, perform the following to set USED
SIZE in ALARM MESS :

2.WRITE 1 → → 2 5 6 → 3 2 ⏎

Note) This can be omitted as the default values
are set in the data length.

Set the numbers in USED SIZE in OPERATE MESS
to COMMENT MESS are required.

Next, register this data into a file.
The display shown on the right will appear when
5.CONVT is pressed. The size can be changed or

registered by pressing Y ⏎.

```
          # (    ) DATA (         ) (           BYTE)

     MODE INIT MENU CONVT
          2.WRITE      5.CONVT      SUBMENU  MENUE
```

The numbers in the USED SIZE sections are as follows:

(Number of messages / Number of registered messages)

Number of registered messages : The number of messages registered in the message table.
Number of messages   : The number of messages that can be registered.

The number of messages and number of registered messages are displayed in the "MEMORY (/)" of each message create screen after this.

CAUTION

If the data length per block of each message is changed, the message data that had been created previously will be invalid.

The message "SELECT INIT!" will appear if an ALARM or OPERATOR MESSAGE screen, etc., is selected without setting the number of messages or data length in the initial setting.

7.2 Writing

The method for writing each message is explained in this section.

[Basic operation]

1.MSSG → ┌ 1.ALARM ┐ → (Read) → 2.WRITE → (Write data) → MENUE
         │ 2.OPERAT │
         │ 3.PLC SW │   → 5.CONVT → ⏎
         │ 4.DEVICE │
         │ 5.COIL   │
         └ 6.COMMENT ┘
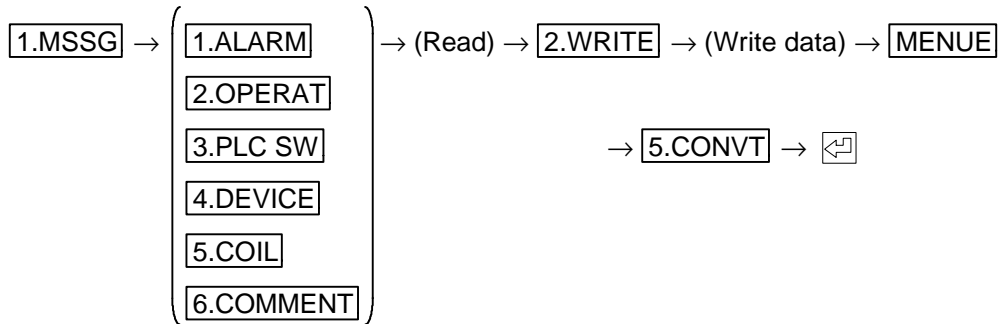
[Operation procedure]

(1) Press 1.MSSG , select each message menu, and then read.

(2) Press 2.WRITE and the setting section will display.

(3) Press MENUE after writing the message data and then press 5.CONVT , ⏎.
     "COMPLETED" will display on the top of the screen when the conversion is completed.

Examples of operation for each message are given on the following pages.

7.2.1 ALARM MESS and OPERATE MESS

[Operation example] Writing ALARM MESS "EMERGENCY".

(1) The following screen will display when 1.MESS 1.ALARM 1.READ 1 2.WRITE ⏎ are pressed.

```
                                                          ALARM MESS
   1  (          )    (                                    )
   2  (          )    (                                    )
   3  (          )    (                                    )
   4  (          )    (                                    )
   5  (          )    (                                    )
   6  (          )    (                                    )
   7  (          )    (                                    )
   8  (          )    (                                    )
   9  (          )    (                                    )
  10  (          )    (                                    )
  11  (          )    (                                    )
  12  (          )    (                                    )
  13  (          )    (                                    )
  14  (          )    (                                    )

                                                   ── Index number
                                                   ── Data register number (D)
                                                   ── Message
   #  (      )  (      )  (                              )
   MEM (          /   100 STEP)

   MODE ALARM     MENU     WRITE
```

(2) The following data will display when 1 → 0 → E M E R G E N C Y ⏎ are pressed.

```
                                                          ALARM MESS
   1  (        0  )    ( EMERGENCY                        )
   2  (          )     (                                  )
   3  (          )     (                                  )
   4  (          )     (                                  )
   5  (          )     (                                  )
   6  (          )     (                                  )
   7  (          )     (                                  )
   8  (          )     (                                  )
   9  (          )     (                                  )
  10  (          )     (                                  )
  11  (          )     (                                  )
  12  (          )     (                                  )
  13  (          )     (                                  )
  14  (          )     (                                  )




   #  (      )  (      )  (                              )
   MEM (          /   100 STEP)

   MODE ALARM     MENU     WRITE
```

(3) The message data has not been transmitted to the file yet.
     Press MENU 5.CONVT ⏎ .
     "COMPLETED" will display when the operation is completed correctly.

~   ~

~   ~

     # (   ) (   ) (   )
     MEM (/   100 STEP)   COMPLETED

     MODE ALARM   MENU   WRITE

     The message data has been transmitted to the file.

(4) The data register number is 0 here. If 1 to 1023 is input, an identification number can be
     displayed after the message when the ALARM MESS is displayed on the CNC.
     For example, if 1 is input, the data register (D) 1 contents are displayed after the message. If the
     D1 contents are 3, 3 will display. The   F or R method can be used for the ALARM MESS display.
     To use F, start the index from 0. To use R, start the index at 1. Note that even if a message with
     index 0 is created, the message cannot be displayed in the R method.

(5) The OPERATE MESS setting method is the same as the ALARM MESS setting method but only
     the R method can be used. A message created with index 0 will be insignificant.
     Refer to the PLC Programming Manual (Ladder Section) for the specifications of ALARM MESS
     and OPERATE MESS.
     The machine name and title used during printing can also be creating using the operator
     messages.

9000	Machine name start number	Used during printing
9001	Machine name end number	Used during printing
9002	Title display start number	Used during printing
9003	Title display end number	Used during printing

Example of title display and machine name printing

```
                                                              FILE MESSAGE1

    ............................................................................................................
    ....
                            STANDARD SEQUENCE DIAGRAM                          ·
    ·                                                                          ·
    ·                           ;98-08-18      BY.MITSUBISHI                   ·
    ·                                                                          ·
    ·                                                                          ·
    ............................................................................................................
    ....
    ·
    ·
    .........          MODIFY MEMO          ...........
    ·
    ·
    ·
    ·
    ·
    ..
```

Machine name printout ——— ⌈ MELDAS 64 SERIES LADDER
(Printed on every page)   BND-400W000-A0   '98-08-18
                            ⌊ MACHINE TYPE   ABC100

Example of OPERATE MESS used for title display and machine name printing

```
 0   (9000)    (MELDAS 64 SERIES LADDER              )  ⎫  Machine name message
 1   (9000)    (BND-400W000-A0          '98-08-18    )  ⎬  (Max. of three lines are valid)
 2   (9001)    (MACHINE TYPE      ABC100             )  ⎭
 3   (9002)    (.................................................)
 4   (9002)    (.                                  .)
 5   (9002)    (.       STANDARD SEQUENCE DIAGRAM   .)
 6   (9002)    (.                                  .)
 7   (9002)    (.             ;98-08-18     BY.MITSUBISHI   .)
 8   (9002)    (.                                  .)
 9   (9002)    (.................................................)
10   (9002)    (.                                   )  ⎫  Title display message
11   (9002)    (.                                   )  ⎬  (The number of lines is not limited)
12   (9002)    (.....   MODIFY MEMO   .             )
13   (9002)    (.                                   )
14   (9002)    (.                                   )
15   (9002)    (.                                   )
16   (9002)    (.                                   )
17   (9002)    (.                                   )
18   (9003)    (.                                   )  ⎭
```

Set 9000 to 9003   Maximum of 60 characters

7.2.2 PLC switches

[Operation example]   Assign "MST_LOCK" to PLC switch 1.

(1)   Press $\boxed{\text{1.MSSG}}$ → $\boxed{\text{3.PLC SW}}$ → $\boxed{\text{1.READ}}$ $\boxed{0}$ $\boxed{\twoheadleftarrow}$ $\boxed{\text{2.WRITE}}$, and the following setting section will display.

```
# (          )   (                          )
```
Message
Index number

(2)   Input $\boxed{0}$ $\boxed{\rightarrow}$ $\boxed{\text{M}}$ $\boxed{\text{S}}$ $\boxed{\text{T}}$ $\boxed{\text{SP}}$ $\boxed{\text{L}}$ $\boxed{\text{O}}$ $\boxed{\text{C}}$ $\boxed{\text{K}}$

```
# (     0 )   (MST_LOCK      )
```

(3)   Press $\boxed{\twoheadleftarrow}$ and the following display will appear.

```
                                                        PLC SWITCH
    #   1  ( MST LOCK            )
        2  (                     )
        3  (                     )
        4  (                     )
        5  (                     )
        6  (                     )
        7  (                     )
        8  (                     )
        9  (                     )
       10  (                     )
       11  (                     )
       12  (                     )
       13  (                     )
       14  (                     )


    # (     )   (                            )
    MEM (          /   32 STEP)

    MODE   PLC SW      MENU      WRITE
```

(4)   Execute $\boxed{\text{MENUE}}$ and press $\boxed{\text{5.CONVT}}$ $\boxed{\twoheadleftarrow}$. "COMPLETED" will display and the message writing will be completed. The PLC switch number displayed on the CNC will be the index number incremented by one.

CAUTION

To use the PLC switch name file created here with the MELDASMAGIC MMI software (option), a conversion must be made with the "PLC Switch Conversion Tool (CNVPLCSW)" found on the "Utility disk". Refer to the "Utility Disk Instruction Manual" for details on how to use this conversion tool.

7.2.3 Device (Contact) comment and coil comment

[Operation example] Write device "M100" and comment "SP.CW" in contact 10.

(1) Press 1.MSSG → 4.DEVICE → 1.READ → 1 0 ⏎ 2.WRITE , and the following setting section will display.

```
# (    ) (    ) (        ) (            )
```
- Contact comment or coil comment
- Device number
- Device
- Index number
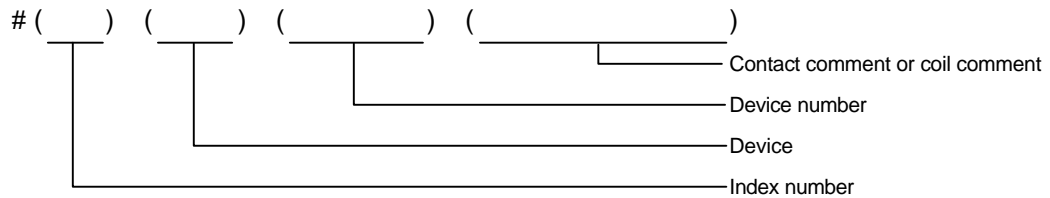
(2) Input 1 0 → M → 1 0 0 → S P . C W ⏎

```
# (  10) (M) (  100) (          SP. CW)
```

(3) Press ⏎ and the following will display.

```
                                            PLC CONTACT COMMENT
    # 10 (  M  ) (    100  ) (SP. CW )
      11 (     ) (         ) (        )
      12 (     ) (         ) (        )
      13 (     ) (         ) (        )
      14 (     ) (         ) (        )
      15 (     ) (         ) (        )
      16 (     ) (         ) (        )
      17 (     ) (         ) (        )
      18 (     ) (         ) (        )
      19 (     ) (         ) (        )
      20 (     ) (         ) (        )
      21 (     ) (         ) (        )
      22 (     ) (         ) (        )
      23 (     ) (         ) (        )
      24 (     ) (         ) (        )


    # (     ) (                          )
    MEM (   100 /   1000 STEP)

    MODE   DEVICE      MENU      WRITE
```

(4) The coil comment can be set in the same manner as the contact comment.

Note)

```
        (A)
        SP. CW                    Y100      (B)
    ───────┤├──────────────────<   >───┤ SPINDLE  CW
```

The (A) comment is the contact comment.
The (B) comment is the coil comment.

7.2.4 Comment

[Operation example]

(1) Press 1.MSSG 6.COMMENT 1.READ 0 ⏎ 2.WRITE and the following setting section will display.

```
# (  _  )  (  _  )  (  ___  )  (  _____  )
                                          └─ Message
                            └─ Device number
              └─ Device
      └─ Index number
```

(2) Set the device as 1, the device number as 0 and create a message for the TOOL REGISTRATION screen. A maximum of five steps with a maximum of eight characters can be created.
The CNC will display only the first five steps even if more than five steps are created.

(3) Set the device as 2, the device number as 0 or 1 and create a message for the load meter. The load meter comment is divided into the title, scale and unit, etc., and created.

No.	Device	Device No.	Comment display length	Details
1	1	Not used (default value 0)	8 alphanumeric characters	Device 1 comment is read from top to bottom and displayed from left to right on the TOOL REGISTRATION screen.
2	2	0: First series screen comment 1: Second series screen comment	30 characters or 10 characters	The comments with a device 2 is read from the top and displayed as the load meter comments.

(4) Execute MENUE and press 5.CONVT → ⏎ after writing the comment messages.

An example of comment creation is shown below.

```
 0 ( 1 )  (  0 )  ( [Spindle]                                              )
 1 ( 1 )  (  0 )  ( [Standby 1]                                            )
 2 ( 1 )  (  0 )  ( [Standby 2]                                            )
 3 ( 1 )  (  0 )  ( [Standby 3]                                            )
 4 ( 1 )  (  0 )  ( [Index]                                                )
 5 ( 2 )  (  0 )  ( Spindle load 1                                         )
 6 ( 2 )  (  0 )  ( 0            50           100                          )
 7 ( 2 )  (  0 )  ( *       %                                              )
 8 ( 2 )  (  0 )  (|_|_|_|_|_|_|_|_|_|_|_|_|_|                             )
 9 ( 2 )  (  0 )  ( Z axis load 1                                          )
10 ( 2 )  (  0 )  ( 0            50           100                          )
11 ( 2 )  (  0 )  ( *       %                                              )
12 ( 2 )  (  0 )  (|_|_|_|_|_|_|_|_|_|_|_|_|_|                             )
13 ( 2 )  (  1 )  ( X axis load 2                                          )
14 ( 2 )  (  1 )  ( 0            50           100                          )
15 ( 2 )  (  1 )  ( *       %                                              )
16 ( 2 )  (  1 )  (|_|_|_|_|_|_|_|_|_|_|_|_|_|                             )
17 ( 2 )  (  1 )  ( Z axis load 2                                          )
18 ( 2 )  (  1 )  ( 0            50           100                          )
19 ( 2 )  (  1 )  ( *       %                                              )
20 ( 2 )  (  1 )  (|_|_|_|_|_|_|_|_|_|_|_|_|_|                             )
21 ( _ )  (  1 )  (                                                        )
      └─────┬──────── Device No.
            └─ Device
```

The comments number 13 to 20 (device number:1) are the comment data for the second series when using a 2-series system.

Load meter display

    The bar graph start position is fixed to the 11th character from the left.

    The display length (number of characters) is set in R155/R355.

    The R154/R354 value is displayed (binary 0~32767). — Load meter 2

    The display length (number of characters) is set in R153/R353

    The R152/R352 value is displayed (binary 0~32767). — Load meter 1

(Note) R152 to 155 control the first series screen display when using a 2-series system, and R352 to 355 control the second series screen display when using a 2-series system.

No.				On the actual 9-inch CRT:
5	(2)	(0)	(SP load)	Display 10 characters from 1st line left
6	(2)	(0)	(0 50 100)	Display 30 characters from 1st line right
7	(2)	(0)	(* %)	Display 10 characters from 2nd line left
8	(2)	(0)	(|_|_|_|_|_|_|_|_|_|_|_|_|_|)	Display 30 characters from 2nd line right
9	(2)	(0)	(Z ax. load)	Display 10 characters from 3rd line left
10	(2)	(0)	(0 50 100)	Display 30 characters from 3rd line right
11	(2)	(0)	(* %)	Display 10 characters from 4th line left
12	(2)	(0)	(|_|_|_|_|_|_|_|_|_|_|_|_|_|)	Display 30 characters from 4th line right

Device No.

Device

Precautions for creating load meter

A pair is created with two lines using the first line for the first step and the second line as the second step. In the above example, No. 5, 7, 9 and 11 are the first step, and No. 6, 8, 10 and 12 are the second step.

1. The comment message length must be 60 bytes. (The maximum display data on the actual machine is 30 characters, so this is set to 60 characters for compatibility with other models.)
2. The message display will display a maximum of 10 bytes on the first step and 30 bytes on the second step.
3. Write the device number in the second step even if the message that is 10 bytes or shorter is being displayed.
4. Do not use a Kanji character for the 10th byte. (Kanji characters that extend over the 9 to 10th bytes are possible, but those that extend over the 10 to 11th bytes are not possible.)
5. The third step and seventh step of the comment display will not display the first six characters of the ten characters. In this example, the first six characters are *   %.

7.2.5 Precautions for creating messages

(1) Semi-graphics (↑, ↓, etc.) cannot be used for the message data created with the personal computer. If semi-graphics must be used with the load meter, etc., leave that line blank, and create the message containing the semi-graphics with the CNC's onboard function.

(2) An even number of characters must always be set for each message. The CNC may not function correctly if an odd number is set.

(3) A maximum of 60 characters (30 characters $\times$ 2 lines) can be displayed for the operator message. Thus, if a Kanji character (2 character data) is used for the 30th or 31st character, it will not be displayed correctly. Use a blank space for the 30th character here.

Example of operator message

```
The external operator message_
display uses two lines.
```

Use a space for the 30th character.

#1 (0)   (The external operator message_display uses two lines.)
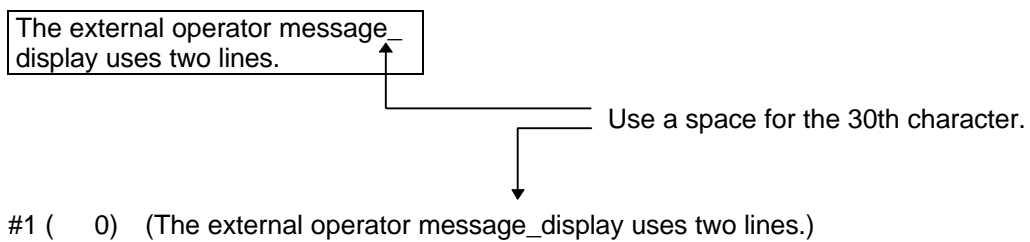
7.3 Read

The method for reading the messages is explained in this section.

[Basic operation]

1.MSSG → ⎛ 1.ALARM ⎞ → 1.READ → (Set the index number of the message to be read into the → ⏎ setting area.)

    2.OPERAT

    3.PLC SW

    4.DEVICE

    5.COIL
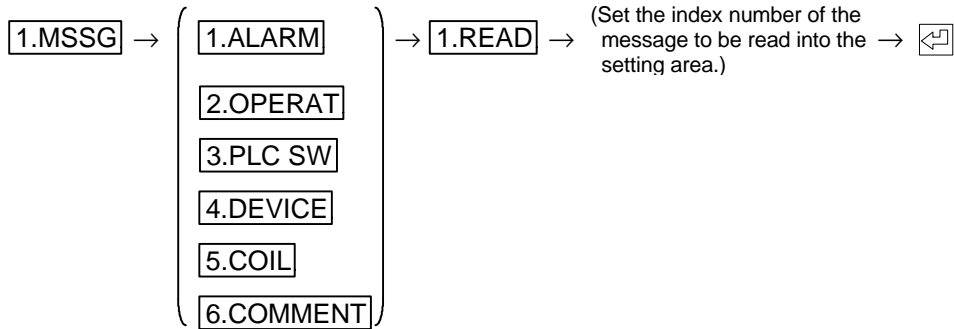
    ⎝ 6.COMMENT ⎠

[Operation example]

(1) Press 1.MSSG and then select one from 1.ALARM to 6.COMMENT .

(2) Press 1.READ and the setting section will display. Input the index number of the message to be read into the setting area, and press ⏎ .
15-lines worth of messages will display from the index number.

(3) Input "+" or "−" into the index number setting area to feed the page and then press ⏎ .
"+" indicates forward page feed and "−" indicates reverse page feed.

7.4 Insertion

The method for inserting each message is explained in this section.
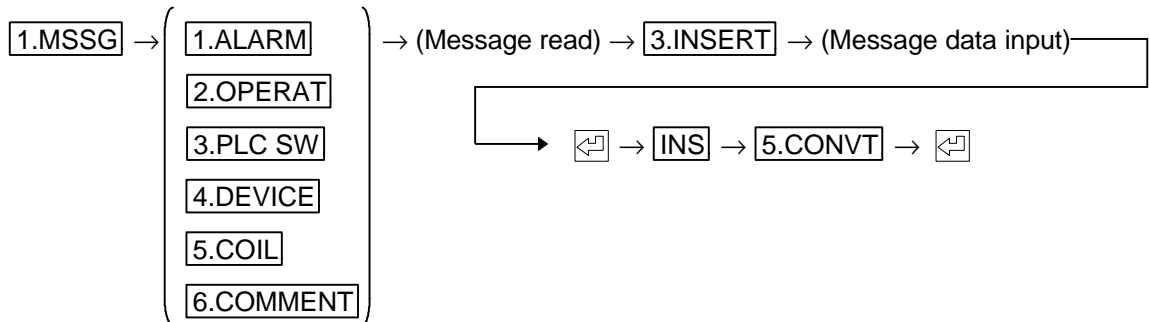
[Basic operation]

1.MSSG → ⎛ 1.ALARM ⎞ → (Message read) → 3.INSERT → (Message data input)
⎜ 2.OPERAT ⎜
⎜ 3.PLC SW ⎜
⎜ 4.DEVICE ⎜   ⤷ ⏎ → INS → 5.CONVT → ⏎
⎜ 5.COIL ⎜
⎝ 6.COMMENT ⎠

[Operation example]

(1) Press 1.MSSG and then select one from 1.ALARM to 6.COMMENT .

(2) Read the message section to be inserted.

(3) Press 3.INSERT , set the index number and data to be inserted into the setting area, and then press ⏎ .
An example is shown below.

For example to insert data between 101 and 102, designate the 102 position, and create the data to be inserted.

```
    100   (1001)   (EMERGENCY STOP!              )
    101   (1001)   (EMERGENCY!                   )
    102   (1002)   (SPINDLE ERROR                )
    103   (1003)   (OIL PRESSURE PUMP ERROR )
                                    :
                                    :
```

(102) (1002) (SPINDLE ERROR)
When ⏎ is pressed, the following will occur.

```
    100   (1001)   (EMERGENCY STOP!              )
    101   (1001)   (EMERGENCY!                   )
    102   (1002)   (SPINDLE ERROR                )
    103   (1002)   (SPINDLE ERROR                )
    104   (1003)   (OIL PRESSURE PUMP ERROR )
                                    :
                                    :
```

(4) Press INS if insertion is completed, press 5.CONVT and then ⏎ .
"COMPLETED" will display when conversion is completed.

(Note) The inserted data will be invalid if write or read is selected without conversion.

7.5 Deletion

The method for deleting each message is explained in this section.
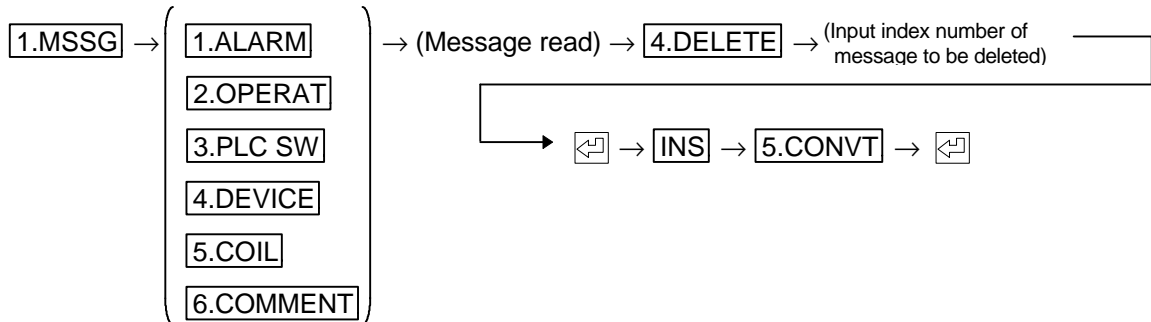
[Basic operation]

$\boxed{\text{1.MSSG}}$ → $\left(\begin{array}{c}\boxed{\text{1.ALARM}}\\ \boxed{\text{2.OPERAT}}\\ \boxed{\text{3.PLC SW}}\\ \boxed{\text{4.DEVICE}}\\ \boxed{\text{5.COIL}}\\ \boxed{\text{6.COMMENT}}\end{array}\right)$ → (Message read) → $\boxed{\text{4.DELETE}}$ → (Input index number of message to be deleted)

$\boxed{\leftarrow}$ → $\boxed{\text{INS}}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{\leftarrow}$

[Operation example]

(1) Press $\boxed{\text{1.MSSG}}$ and then select one from $\boxed{\text{1.ALARM}}$ to $\boxed{\text{6.COMMENT}}$.

(2) Read the message section to be deleted.

(3) Press $\boxed{\text{4.DELETE}}$, set the index number to be deleted into the setting area, and then press $\boxed{\leftarrow}$.
An example is shown below.

```
# 100   (1001)   (EMERGENCY STOP!          )
   101   (1001)   (EMERGENCY!               )
   102   (1002)   (SPINDLE ERROR            )
   103   (1003)   (OIL PRESSURE PUMP ERROR )
                                 :
                                 :
```

Set the index and press $\boxed{\leftarrow}$.
\# (102) (1002) (SPINDLE ERROR)
The following will occur.

```
   100   (1001)   (EMERGENCY STOP!          )
   101   (1001)   (EMERGENCY!               )
   102   (1003)   (OIL PRESSURE PUMP ERROR )
                                 :
                                 :
```

(4) Press $\boxed{\text{INS}}$ if deletion is completed, press $\boxed{\text{5.CONVT}}$ and then $\boxed{\leftarrow}$.

7.6   Conversion

The edited data is written into the message file.

[Basic operation]

$\boxed{\text{1.MSSG}}$ → (Edit message) → $\boxed{\text{INS}}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{\leftarrow}$

[Operation example]

(1)   Press $\boxed{\text{1.MSSG}}$ and edit the message (write, insert, delete).

(2)   Press $\boxed{\text{INS}}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{\leftarrow}$ after editing.
    The contents of the message file will be updated when "COMPLETED" is displayed.

8. Printing

The PLC development software can edit as well as print the ladder circuits and messages.
There are two methods of printing: printer and printer 2. Printer 2 allows the size of the ladder circuit to
be selected.

```
                                                                    PRINT  MODE
        1.LADDER              ( 1) PRINT OUT
        2.COMMENT    (   0)   SET   =   1
        3.ALARM MSSG          ( 0)          31.USED PAPER SIZE     11INCH
        4.OPERATE MSSG        ( 0)
        5.COMMENT MSSG        ( 0)          32.PRINT START POINT  X=0
        6.PLC SW MSSG         ( 0)
                                            33.DATA LENGTH        8+1

                                            34.START PAGE NO(    1) 0 <= X < 10000
        20.MACHINE NAME       (1)
        21.LADDER FILE        (  BASE.LD            )
        22.POINT START STEP   (      0)
        23.    END   STEP     (   100)
        24.MODE SELECT        ( 0)      0:LAD,1:LAD.+COM,2:LAD.+COM.+CRS
        25.MESSAGE FILE       (BASE.MS        )
        26.I/O DEVICE         ( )
        27.POINT START NO.    (  0)
        28.    END  NO.       (  0)
        29.MODE SELECT        ( 0)      0.ALL  DEVICE  1.DEVICE  2.DEV.+DEV  NO.  Screen for
                                                                                  3. PRINT

        #(  21  )DATA(          LAD1      )
                              3PRINT    5PRINT2
```

```
                                                          PRINT 2

   1.LADDER            ( 1) PRINT OUT        30.USED PAPER ( 1 )
   2.DEVICE, COIL      ( 0) SET = 1          0:15∗11INCH       1:297∗210mm    2:210∗297mm
   3.ALARM MSSG        ( 0)                  3:364∗257mm       4:257∗364mm    5:9∗11INCH
   4.OPERATE MSSG      ( 0)                  6:FREE FORM
   5.COMMENT MSSG      ( 0)
   6.PLC SW MSSG       ( 0)




   20.MACHINE NAME        ( 1)
   21.LADDER FILE         ( BASED.LD  )
   22.POINT START STEP    (   0)
   23.    END STEP        ( 100)
   24.MODE SELECT         ( 0)        0:LAD, 1:LAD. +COM, 2:LAD. +COM. +CRS
   25.MESSAGE FILE        ( BASED.MS  )
   26.I/O DEVICE          ( )
   27.POINT START NO.     (  0)
   28.    END NO.         (  0)
   29.MODE SELECT         ( 0)        COIL--0.ALL DEVICE 1.DEVICE 2.DEV.+DEV NO.   Screen for
                                      DEVI--3.ALL DEVICE 4.DEVICE 5.DEV.+DEV NO.   5.PRINT2

   # (   ) DATA (        )
              3PRINT        5PRINT2
```

[Basic operation]

$\boxed{\text{6.EXT IO}}$ → $\begin{bmatrix} \boxed{\text{3.PRINT}} \\ \boxed{\text{5.PRINT2}} \end{bmatrix}$ → (Set printing conditions) → $\boxed{\text{G}}$ $\boxed{\text{O}}$ $\boxed{\hookleftarrow}$

[Example of operation] ... For both printer and printer 2

(1) Select $\boxed{\text{6.EXT IO}}$ and then select printer or printer 2.

(2) Next input the various conditions into the setting section.

    # (⌐) (⌐) _____ Data
            └_____┘ Index
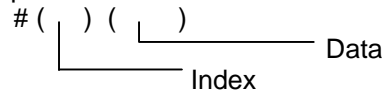
(3) Input $\boxed{\text{G}}$ $\boxed{\text{O}}$ in the index setting section, and press $\boxed{\hookleftarrow}$. Printing will start.
    When completed, the GO display will disappear and the cursor will appear.

8.1 Printing the Ladder Program (Printer)

The procedure for selecting the printer and printing the ladder circuit is shown below.

```
Select LADDER.

To print with the machine name
added, set 20. MACHINE NAME to 1.

Set the ladder file name.

Set the START STEP and END
STEP.
All steps will be printed when both the
START STEP and END STEP are set
to 0.

Set MODE SELECT.

Set the message file when "20.
MACHINE NAME" and "24. MODE
SELECT (with comment)" are
designated.

Set the start signal.
# (GO) DATA (  )

Execute printing.

The start signal will disappear and #
(  ) DATA (  ) will appear when printing
is completed.
```

```
1.LADDER          ( 1)  PRINT OUT
2.COMMENT         ( 0)  SET = 1
3.ALARM MSSG      ( 0)
4.OPERATE MSSG    ( 0)
5.COMMENT MSSG    ( 0)
6.PLC SW MSSG     ( 0)




20.MACHINE NAME       ( 1)
21.LADDER FILE        ( BASED.LS  )
22.POINT START STEP   (   0)
23.    END STEP       ( 100)
24.MODE SELECT        ( 0)      0:LAD,
25.MESSAGE FILE       ( BASED.MS  )
26.I/O DEVICE         ( )
27.POINT START NO.    (  0)
28.    END NO.        (  0)
29.MODE SELECT        ( 0)      0:ALL


# (  ) DATA (        )
           3PRINT      5PRI
```

Set the page to start printing from in 34. START PAGE NO. Normally, printing will start from page 1.

CAUTION
Do not use the same name for the ladder file and message file.

8.2 Printing the Comment List (Printer)

The procedure for selecting the printer and printing the comment list (coil comments, contact comments) is shown below.

Select COMMENT MSSG.

↓

To print with the machine name added, set 20. MACHINE NAME to 1.

↓

Set the message file name. Set into 25. MESSAGE FILE ( ).

↓

Set the comment list mode.
29. MODE SELECT ( )
Select one of the following:
  0. ALL DEVICE
  1. DEVICE
  2. DEV. + DEV. NO.

↓

If 1 or 2 is selected above, set the designated device. Designate device A to Z.

↓

Next, set the range of the designated device. Refer to "Section 3. Development Procedure" for the setting range.

⋮

Set the start signal. # (GO) DATA ( )

⋮

Execute printing. The GO message will disappear and # ( ) DATA ( ) will appear when printing is completed.

```
1.LADDER          ( 0)  PRINT OUT
2.COMMENT         ( 1)  SET = 1
3.ALARM MSSG      ( 0)
4.OPERATE MSSG    ( 0)
5.COMMENT MSSG    ( 0)
6.PLC SW MSSG     ( 0)



20.MACHINE NAME       ( 1)
21.LADDER FILE        ( BASED.LD  )
22.POINT START STEP   (    0)
23.    END STEP       ( 100)
24.MODE SELECT        ( 0)      0:LAD,
25.MESSAGE FILE       ( BASED.MS  )
26.I/O DEVICE         ( )
27.POINT START NO.    (   0)
28.    END NO.        (   0)
29.MODE SELECT        ( 0)      0:ALL


# (   ) DATA (       )
              3PRINT      5PRI
```

– 73 –

8.3 Printing the Ladder Program (Printer 2)

The procedure for selecting the printer 2 and printing the ladder circuit is shown below.

Select LADDER.

↓

To print with the machine name added, set 20. MACHINE NAME to 1.

```
1.LADDER          ( 1)  PRINT OUT
2.DEVICE, COIL    ( 0)  SET = 1
3.ALARM MSSG      ( 0)
4.OPERATE MSSG    ( 0)
5.COMMENT MSSG    ( 0)
6.PLC SW MSSG     ( 0)
```

↓

Set the ladder file name.

↓

Set the START STEP and END STEP. All steps will be printed when both the START STEP and END STEP are set to 0.

```
20.MACHINE NAME      ( 1)
21.LADDER FILE       ( BASE.LD )
22.POINT START STEP  (   0)
23.    END STEP      ( 100)
24.MODE SELECT       ( 0)        0:LAD,
25.MESSAGE FILE      ( BASE.MS )
26.I/O DEVICE        ( )
27.POINT START NO.   (   0)
28.    END NO.       (   0)
29.MODE SELECT       ( 0)        COIL–
                                 DEVI–

# (   ) DATA (        )
              3PRINT      5PRI
```

↓

Set the MODE SELECT.

↓

Set the message file when "20. MACHINE NAME" and "24. MODE SELECT (with comment)" are designated.

Note) Only the ladder will be printed if the message file is not designated.

↓

Designate the printer paper. 30. USED PAPER (0) 0: 15*1INCH   1: 297*210MM 2: 210*297MM 3: 364*257MM 4: 257*364MM 5: 9*11INCH

```
                      PRINT 2

30.USED PAPER ( )
  0:15∗11INCH    1:297∗210mm   2:210∗297mm
  3:364∗257mm    4:257∗364mm   5:9∗11INCH
  6:FREE FORM

31. WIDE          ( 1) 0:1/6INCH  1:1/8INCH
32.LINE           ( 80)
33.COLUMNS        ( 75)
34.REDUCTION      ( 0)
35.LEFT MARGIN    ( 0)
36.START PAGE NO  ( 1)  0 <= X < 10000
LAD, 1:LAD. +COM, 2:LAD. +COM. +CRS
```

Note) "0" is set for the screen selection.

↓

Set the line spacing. 31. WIDE (1) 0: 1/6 INCH 1: 1/8 INCH

Note) "0" is set for the screen selection.

```
COIL--0.ALL DEVICE 1.DEVICE 2.DEV.+DEV NO.
DEVI--3.ALL DEVICE 4.DEVICE 5.DEV.+DEV NO.


5PRINT2
```

↓

To next page.

```
                    |
                    ↓
┌──────────────────────────────┐
│ Set the start signal.         │
│ # (GO) DATA (   )             │
└──────────────────────────────┘
                    ┊
                    ↓
┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
  Execute printing.            
└ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘
                    ┊
                    ↓
┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
  The start signal will disappear and #
  ( ) DATA ( ) will appear when printing
  is completed.
└ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘
```

Set the page to start printing from in 34. START PAGE NO. Normally, printing will start from page 1.

CAUTION

1. Do not set the same name for the ladder file and message file.

2. Only the ladder will be printed if the message file name is not designated even if "1" or "2" is set in 24. MODE SELECT ( ).

3. The contact comments and coil comments must be created with the message function. A maximum of 6 characters can be set for the contact comment and 18 characters for the coil contact.

4. The OR circuit is not separated when the page is changed, but if the coil's cross section is more than four lines when the OR circuit is printed on the last line of the page, the OR circuit will be continued on the next page.

8.4 Printing the Contact Comments and Coil Comments (Printer 2)

The procedure for selecting printer 2 and printing the contact comments and coil comments is shown below.

Set "1" in 2. DEVICE/COIL ( ).

```
1.LADDER            ( 0)  PRINT OUT
2.DEVICE, COIL      ( 1)  SET = 1
3.ALARM MSSG        ( 0)
4.OPERATE MSSG      ( 0)
5.COMMENT MSSG      ( 0)
6.PLC SW MSSG       ( 0)
```

To print with the machine name added, set 20. MACHINE NAME to 1.

Set the message file name in 25. MESSAGE FILE ( ).

```
20.MACHINE NAME       ( 1)
21.LADDER FILE        ( BASE.LD  )
22.POINT START STEP   (   0)
23.    END STEP       ( 100)
24.MODE SELECT        ( 0)       0:LAD,
25.MESSAGE FILE       ( BASE.MS  )
26.I/O DEVICE         ( )
27.POINT START NO.    (  0)
28.    END NO.        (  0)
29.MODE SELECT        ( 0)        COIL−
                                  DEVI−

# (   ) DATA (       )
           3PRINT      5PRI
```

Set 29. DEVICE/COIL MODE ( ).
Select one of the following:
 0: ALL DEVICE
 1: DEVICE
 2: DEV. + DEV. NO.
 3: ALL DEVICE
 4: DEVICE
 5: DEV. + DEV. NO.

If 1, 2, 4 or 5 is set in 29. MODE SELECT ( ) above, set device A to Z in the I/O DEVICE ( ).

```
1.LADDER            ( 0)  PRINT OUT
2.DEVICE, COIL      ( 1)  SET = 1
3.ALARM MSSG        ( 0)
4.OPERATE MSSG      ( 0)
5.COMMENT MSSG      ( 0)
6.PLC SW MSSG       ( 0)
```

If 2 or 5 is set in 29. MODE SELECT ( ) above, set the range of the designated device.
Set the device number in :
 27. POINT START NO. ( )
 28. END NO. ( )

Note) If END NO. is "0", all device Nos. will be printed.

```
20.MACHINE NAME       ( 1)
21.LADDER FILE        ( BASE.LD  )
22.POINT START STEP   (   0)
23.    END STEP       ( 100)
24.MODE SELECT        ( 0)       0:LAD,
25.MESSAGE FILE       ( BASE.MS  )
26.I/O DEVICE         ( )
27.POINT START NO.    (  0)
28.    END NO.        (  0)
29.MODE SELECT        ( 0)        COIL−
                                  DEVI−

# (   ) DATA (       )
           3PRINT      5PRI
```

Set the start signal.
(GO) DATA ( )

The GO message will disappear and # ( ) DATA ( ) will appear when printing is completed.

8.5 Example of Printing

An example of printing with printer 2 using the PLC development software is shown below.

8.5.1 Printing with ladder + comment + cross

Printer   Example LP-7000

FILE LAD 400M8

P252	0	M1000 P-ON 3 / M1000 P-ON 3
	4	M1000 P-ON 3 — [MOV R70 M1084] K4
	8	M1000 P-ON 3 — [MOV R2934 M1024] K4 385
	12	X0 ST Y218 Automatic operation start 13
	14	X1 *SP Y219 *Automatic operation stop 15
	16	X2 SBK Y21A Single block 17
	18	X3 DRN Y21D Dry run 19
	20	X4 MLK Y1C8 Automatic machine lock 1 21
		Y1C9 Automatic machine lock 2 22
		Y1CB Automatic machine lock 4 23
		Y1CC Automatic machine lock 5 24
		Y1CD Automatic machine lock 6 25
		Y1D0 Manual machine lock 1 26
		Y1D1 Manual machine lock 2 27
		Y1D3 Manual machine lock 4 28
		Y1D4 Manual machine lock 5 29
		Y1D5 Manual machine lock 6 30
	31	X4 MLK / X6 ZCAN Y1CA Automatic machine lock 3 33
	34	X5 BDT 1 Y23F Optional block skip 35
	36	X6 ZCAN Y1D2 Manual machine lock 3 37
	38	X7 AFL Y29A Miscellaneous function lock 39
	40	X8 MEM Y210 Memory 41

M1000

1	2	3	4
8	358	362	386
396			

8.5.2 Example of contact comment printing

COMMENT   FILE MSG500

X180	RDY1	X1CB		X216	SNGE	
X181	RDY2	X1CC		X217	ASLE	
X182	RDY3	X1CD		X218	F11	
X183	RDY4	X1CE		X219	F12	
X184	RDY5	X1CF		X21A	F14	
X185	RDY6	X1D0		X21B	F18	
X186	RDY7	X1D1		X21C		
X187	RDY8	X1D2		X21D		
X188	AX1	X1D3		X21E		
X189	AX2	X1D4		X21F		
X18A	AX3	X1D5		X220	DM00	
X18B	AX4	X1D6		X221	DM01	
X18C	AX5	X1D7		X222	DM02	
X18D	AX6	X1D8		X223	DM30	
X18E	AX7	X1D9		X224		
X18F	AX8	X1DA		X225	GR1	
X190	MVP1	X1DB		X226	GR2	
X191	MVP2	X1DC		X227		
X192	MVP3	X1DD		X228	EF	
X193	MVP4	X1DE		X229	MMS	
X194	MVP5	X1DF		X22A		
X195	MVP6	X1E0	JO	X22B		
X196	MVP7	X1E1	HO	X22C		
X197	MVP8	X1E2	SO	X22D		
X198	MVM1	X1E3	PTPO	X22E		
X199	MVM2	X1E4	ZRNO	X22F		
X19A	MVM3	X1E5		X230	MF1	
X19B	MVM4	X1E6		X231	MF2	
X19C	MVM5	X1E7		X232	MF3	
X19D	MVM6	X1E8	MEMO	X233	MF4	
X19E	MVM7	X1E9	TO	X234	SF1	
X19F	MVM8	X1EA		X235		
X1A0	ZP11	X1EB	DO	X236		
X1A1	ZP12	X1EC		X237		
X1A2	ZP13	X1ED		X238	TF1	
X1A3	ZP14	X1EE		X239	TF2	
X1A4	ZP15	X1EF		X23A		
X1A5	ZP16	X1F0	MA	X23B		
X1A6	ZP17	X1F1	SA	X23C	BF1	
X1A7	ZP18	X1F2	OP	X23D	BF2	
X1A8	ZP21	X1F3	STL	X23E	BF3	
X1A9	ZP22	X1F4	SPL	X23F	BF4	
X1AA	ZP23	X1F5	RST	X240		
X1AB	ZP24	X1F6	CXN	X241	CDO	
X1AC	ZP25	X1F7	RWD	X242	VRO	
X1AD	ZP26	X1F8	DEN	X243	FLO	
X1AE	ZP27	X1F9	TIMP	X244	ZSO	
X1AF	ZP28	X1FA	TSMZ	X245	USO	
X1B0	ZP31	X1FB		X246	ORAO	
X1B1	ZP32	X1FC	CXFIN	X247		
X1B2	ZP33	X1FD		X248	SMA	
X1B3	ZP34	X1FE		X249	SSA	
X1B4	ZP35	X1FF	HINT	X24A	SEMG	
X1B5	ZP36	X200	RPN	X24B	SSRN	
X1B6	ZP37	X201	CUT	X24C	SSRI	
X1B7	ZP38	X202	TAP	X24D	SZPH	
X1B8	ZP41	X203	THRD	X24E	SIMP	
X1B9	ZP42	X204	SYN	X24F	STLQ	
X1BA	ZP43	X205	CSS	X250		
X1BB	ZP44	X206	SKIP	X251		
X1BC	ZP45	X207	ZRNN	X252		
X1BD	ZP46	X208	INCH	X253		
X1BE	ZP47	X209	DLKN	X254		
X1BF	ZP48	X20A	FIDN	X255		
X1C0		X20B	TLFO	X256		
X1C1		X20C	SUPP	X257		
X1C2		X20D	SLOW	X258		

8.5.3 Example of coil comment printing

COMMENT LIST   FILE MSG500

DEVICE	COMMENT LIST	DEVICE	COMMENT LIST	DEVICE	COMMENT LIST
Y180	AXIS OFF 1	Y1CB	Automatic machine lock 4	Y216	
Y181	AXIS OFF 2	Y1CC	Automatic machine lock 5	Y217	
Y182	AXIS OFF 3	Y1CD	Automatic machine lock 6	Y218	Automatic operation start
Y183	AXIS OFF 4	Y1CE	Automatic machine lock 7	Y219	∗Automatic operation stop
Y184	AXIS OFF 5	Y1CF	Automatic machine lock 8	Y21A	Single block
Y185	AXIS OFF 6	Y1D0	Manual machine lock 1	Y21B	∗Block start interlock
Y186	AXIS OFF 7	Y1D1	Manual machine lock 2	Y21C	∗Cutting start interlock
Y187	AXIS OFF 8	Y1D2	Manual machine lock 3	Y21D	Dry run
Y188	SERVO OFF 1	Y1D3	Manual machine lock 4	Y21E	
Y189	SERVO OFF 2	Y1D4	Manual machine lock 5	Y21F	Error detect
Y18A	SERVO OFF 3	Y1D5	Manual machine lock 6	Y220	NC reset 1
Y18B	SERVO OFF 4	Y1D6	Manual machine lock 7	Y221	NC reset 2
Y18C	SERVO OFF 5	Y1D7	Manual machine lock 8	Y222	Reset & rewind
Y18D	SERVO OFF 6	Y1D8	Feed axis selection 1	Y223	Chamfering
Y18E	SERVO OFF 7	Y1D9	Feed axis selection 2	Y224	
Y18F	SERVO OFF 8	Y1DA	Feed axis selection 3	Y225	Gear shift complete
Y190	MIRROR IMAGE 1	Y1DB	Feed axis selection 4	Y226	M function complete 1
Y191	MIRROR IMAGE 2	Y1DC	Feed axis selection 5	Y227	M function complete 2
Y192	MIRROR IMAGE 3	Y1DD	Feed axis selection 6	Y228	Tool length measurement
Y193	MIRROR IMAGE 4	Y1DE	Feed axis selection 7	Y229	Tool length measurement 2
Y194	MIRROR IMAGE 5	Y1DF	Feed axis selection 8	Y22A	
Y195	MIRROR IMAGE 6	Y1E0	Feed axis selection −1	Y22B	Program resume
Y196	MIRROR IMAGE 7	Y1E1	Feed axis selection −2	Y22C	Playback
Y197	MIRROR IMAGE 8	Y1E2	Feed axis selection −3	Y22D	Macro interrupt
Y198		Y1E3	Feed axis selection −4	Y22E	Rapid traverse
Y199		Y1E4	Feed axis selection −5	Y22F	
Y19A		Y1E5	Feed axis selection −6	Y230	Manual absolute
Y19B		Y1E6	Feed axis selection −7	Y231	Display lock
Y19C		Y1E7	Feed axis selection −8	Y232	F 1 digit speed change valid
Y19D		Y1E8		Y233	Recalculation request
Y19E		Y1E9		Y234	Integrated time input 1
Y19F		Y1EA		Y235	Integrated time input 2
Y1A0		Y1EB		Y236	
Y1A1		Y1EC		Y237	
Y1A2		Y1ED		Y238	∗Data protect key 1
Y1A3		Y1EE		Y239	∗Data protect key 2
Y1A4		Y1EF		Y23A	∗Data protect key 3
Y1A5		Y1F0		Y23B	∗Data protect key 4
Y1A6		Y1F1		Y23C	
Y1A7		Y1F2		Y23D	
Y1A8	Automatic interlock 1	Y1F3		Y23E	
Y1A9	Automatic interlock 2	Y1F4		Y23F	Optional block skip
Y1AA	Automatic interlock 3	Y1F5		Y240	
Y1AB	Automatic interlock 4	Y1F6		Y241	
Y1AC	Automatic interlock 5	Y1F7		Y242	
Y1AD	Automatic interlock 6	Y1F8		Y243	
Y1AE	Automatic interlock 7	Y1F9		Y244	
Y1AF	Automatic interlock 8	Y1FA		Y245	
Y1B0	Automatic interlock −1	Y1FB		Y246	
Y1B1	Automatic interlock −2	Y1FC		Y247	
Y1B2	Automatic interlock −3	Y1FD		Y248	No.1 handle axis No.11
Y1B3	Automatic interlock −4	Y1FE		Y249	No.1 handle axis No.12
Y1B4	Automatic interlock −5	Y1FF		Y24A	No.1 handle axis No.14
Y1B5	Automatic interlock −6	Y200		Y24B	No.1 handle axis No.18
Y1B6	Automatic interlock −7	Y201		Y24C	No.1 handle axis No.116
Y1B7	Automatic interlock −8	Y202		Y24D	
Y1B8	Manual interlock 1	Y203		Y24E	
Y1B9	Manual interlock 2	Y204		Y24F	No.1 handle axis valid
Y1BA	Manual interlock 3	Y205		Y250	
Y1BB	Manual interlock 4	Y206		Y251	
Y1BC	Manual interlock 5	Y207		Y252	
Y1BD	Manual interlock 6	Y208	JOG	Y253	
Y1BE	Manual interlock 7	Y209	HANDLE	Y254	
Y1BF	Manual interlock 8	Y20A	Step	Y255	
Y1C0	Manual interlock −1	Y20B	Manual random feed	Y256	
Y1C1	Manual interlock −2	Y20C	Zero point return	Y257	
Y1C2	Manual interlock −3	Y20D		Y258	

8.6 Free Form Setting of the Printing Paper

8.6.1 Outline

When "6. FREE FORM" is selected for "30. USED PAPER" on the "PRINT2" screen, the following printing is possible.

- Free No. of lines (Designate with "32. LINE")
- Free No. of columns (Designate with "33. COLUMNS")
- Reduction (Designate with "34. REDUCTION")
- Free left margin (Designate with "35. LEFT MARGIN")

Note that this function is dedicated for the "PRINT2" screen.

8.6.2 Screen configuration

```
                                                        PRINT 2

    1.LADDER           ( 0)  PRINT OUT     30.USED PAPER ( 0 )
    2.DEVICE, COIL     ( 0)  SET = 1          0:15∗11INCH      1:297∗210mm      2:210∗297mm
    3.ALARM MSSG       ( 0)                   3:364∗257mm      4:257∗364mm      5:9∗11INCH
    4.OPERATE MSSG     ( 0)                    6:FREE FORM
    5.COMMENT MSSG     ( 0)
    6.PLC SW MSSG      ( 0)
                                            31.WIDE             ( ) 0:1/6INCH  1:1/8INCH
                                            32.LINE             (   )
                                            33.COLUMNS          (   )
                                            34.REDUCTION        (   )
                                            35.LEFT MARGIN      (    )
                                            36.START PAGE NO    (    ) 0 <= X < 10000
    20.MACHINE NAME        ( 0)
    21.LADDER FILE         (        )
    22.POINT START STEP    (    0)
    23.    END STEP        (    0)
    24.MODE SELECT         ( 0)         0:LAD, 1:LAD. +COM, 2:LAD. +COM. +CRS
    25.MESSAGE FILE        (        )
    26.I/O DEVICE          (  )
    27.POINT START NO.     (    0)
    28.    END NO.         (    0)
    29.MODE SELECT         ( 0)         COIL--0.ALL DEVICE 1.DEVICE 2.DEV.+DEV NO.
                                        DEVI--3.ALL DEVICE 4.DEVICE 5.DEV.+DEV NO.

    # (    ) DATA  (            )
                     3 PRINT         5 PRINT2
```

8.6.3 Details of function

When "6. FREE FORM" is selected for "30. USED PAPER" on the "PRINT 2" screen, the data set in items 32. to 35. will be validated.

If a setting other than 6 is made, the 32. to 35. settings will be ignored. Thus, if reduction printing or a left margin is to be designated, set 6 for the "30. USED PAPER" setting.

Display item	Details	Setting range (Default value)
30. USER PAPER	0: 15∗11-inch, 1: A4 portrait, 2: A4 landscape, 3: B4 portrait, 4: B4 landscape, 5: 9∗11-inch, 6: Free	
32. LINE	Designate the number of lines to be printed on one page. Designate this value so that the lines will fit on the paper being used. When the designated number of lines have been printed, the change page code will be output. The number of lines here refers to the lines from the file name to the page number. Set the paper size with the printer.	40 ~ 100 (80)
33. COLUMNS	Designate the number of columns on one line. Designate this so that the columns will fit on the paper being used. If there are more than 75 columns, a comment or cross reference can also be printed. The number of cross references printed on one line will depend on the number of columns. Number of cross references = $\dfrac{\text{Number of columns - 75}}{6} + 1$ (round down decimal values) When the comment or cross reference is printed, they may not fit on one line when the normal character size is used. In this case, designate reduction printing.	70 ~ 136 (75)
34. REDUCTION	Designate whether to carry out reduction printing. (Note that some printers may not be capable of this function.)	0/1 (0)
35. LEFT MARGIN	Designate the left margin width with a number of columns. PC-98: Number of normal characters regardless of the reduction designation. PC/AT: Normally, the normal characters. When reduction is designated, the number of reduction characters. **(Note)** Some printers may not be capable of this function.	0 ~ 40 (0)
36. START PAGE NO	Designate the page to start printing. The page number will be printed at the bottom center of the page from the set value. If a value other than 0 is designated for "22. POINT START STEP", the printing will start from the designated step position. At this time, the printing page number will start from the designated value. If an illegal value is set in "36. START PAGE NO", the "START PAGE ILLEGAL" error will display. The "START PAGE ILLEGAL" error will occur in the following case: 1) When a setting not within the setting range is made.	0 ~ 9999 (1)

8.6.4 Precautions

(1) If the ladder mode is 0 or if the number of columns does not satisfy 75, the comment or cross reference cannot be printed.
(2) The reduction and left margin settings are valid only when the "30. USED PAPER" setting is "6. FREE FORM".
(3) The default value of each data can be changed to an arbitrary value by using the external file "PR_DATA".
Refer to the section "8.7 External File (PR_DATA)" for details.

8.7 External File (PR_DATA)

The initial values of the parameters on the PRINT 2 screen can be set.
The parameters shown below can be set.

8.7.1 Outline

If the printer output setting data and control codes are registered in the external file (PR_DATA), the following settings and changes will be possible.
- Setting of initial value of parameters for printing
- Changing of printer control codes.

8.7.2 Details of functions

(1) Parameters for printing

The following parameters are read when the tool is started, and are used as the initial setting values.

```
24.  MODE SELECT   (LDM)
30.  USED PAPER    (PPP)
31.  WIDE          (CRW)
32.  LINE          (LNS)
33.  COLUMN        (CLS)
34.  REDUCTION     (RED)
35.  LEFT MARGIN   (LFT)
```
   ∗ The names given in parentheses are used when defining in the external file.

Cautions

(1) If there is no external file, the default values will be set as the initial values for all parameters.
(2) The default value will be set as the initial values for any parameter that is not designated.
   If a value that exceeds the setting range is set, or if a character other than a number is set,
   the default value will be set as the initial value.

   Setting range :

	LDM	0 ~ 2
	PPP	0 ~ 6
	CRW	0 / 1
	LNS	40 ~ 100
	CLS	70 ~ 136
	RED	0 / 1
	LFT	0 ~ 40

If LDM is set to 1 or 2, and PPP is set to 1 or 5, both LDM and PPP will be invalid.

(This is because the printing details and paper size do not match.)

---

(2) Printer control code

This tool outputs the printer control codes that correspond to PC/AT (ESC/P) or PC98 (PR201).
To use a printer other than these, register the control code that matches the printer into the external file. The following control codes can be set.

Reduction designation (_RED)

Reduction cancel   (_NRD)

Left margin   (_LFT)

   ∗ The names given in parentheses are used when defining in the external file.

These codes are output to the printer according to the state when the printer output starts.
- The reduction designation control code is output when "34. REDUCTION" is set to 1.
- The reduction cancel control code is output when "34: REDUCTION" is set to 0.
- The left margin control code is added to the margin width set in "35. LEFT MARGIN" and output.

Cautions

(1) If a control code is not designated, the default control code will be used.

Default control code: For PC/AT .... Follows ESC/P

For PC98 ...... Follows PR201

(2) The control codes are sequentially output one byte at a time. The output will stop at the point that an illegal code is found (more than three digits, or other than 0-9, A-F or special codes).

Even if an incorrect code is designated, the output will continue to the end if the format is correct. In this case, the following printer operation cannot be guaranteed, so take special care when designating the control code.

(3) If the length of one line exceeds 256 characters, the line following the 257th character will be invalid.

Thus, if all control codes are described with two digits, up to 84 control codes can be output.

---

(3) External file directory

The directory in which the external file is placed is as follows.

Storage directory: \PLC\BIN (Directory that stores the main tool)

The external file is automatically referred to by the tool, so if the name and directory differ, the reference will not be possible.

External file name: "PR_DATA"

(4) File format

The external file is a text file, and the data is set as follows.

Parameter name　　= Data (decimal)

Control code name = Data, data, data... (hexadecimal)

If there are multiple control codes, always separate each byte with a comma (,). If there is a line return midway, the output will end at the return.

Example of external file	Parameter:	
LDM=2	→ Ladder mode	= 2
PPP=6	→ Printer paper	= 6
CRW=1	→ Line width	= 1
LNS=80	→ Lines	= 80
CLS=136	→ Columns	= 136
RED=0	→ Reduction	= 0
LFT=5	→ Left margin	= 5
¦_RED=0F	→ Control code: Reduction designation	= 0F [HEX]
¦_NRD=12	→ Reduction cancel	= 12 [HEX]
¦_LFT=1B,6C,X0	→ Left margin	= 1B [HEX] 6C [HEX]
		special code
		The printer control code will be written.

– 83 –

Special code: The data that starts with an "X" such as "X0" in the control code is called a special code. The special code indicates a code that cannot be designated because the code changes due to the conditions, etc.

Currently, the left margin width is a special code. The method of expressing the margin width differs according to the printer, so the width is changed by the special code type.

ESC/P   : Binary, 1 byte (0A for 10 digits)
PR201   : Character string, 3 bytes (30,31,30 for 10 digits)
For ESC/P type : Designate the special code "X0".
For PR201 type : Designate the special code "X1".

(Example)   ESC/P_LFT=1B,6C,X0
PR201_LFT=1B,4C,X1

If set as shown above, the tool will generate the margin width data from the left margin setting value, and will output the data to the printer.

The left margin value can be applied with a fixed value, but in this case, the left margin setting value set on the screen will be ignored.

8.7.3 Appendix ESC/P and PR201 printer control codes

∗ The format used to designate with the external file "pr_data" is shown in the parentheses.

Reduction designation	ESC/P : 0x0F (_RED=0F) PR201 : 0x1C, 0x6D, "1/1, 1/2, H." (_RED=1C,6D,31,2F,31,2C,31,2F,32,2C,48,2E)
Reduction cancel	ESC/P : 0x12 (_NRD=12) PR201 : 0x1C, 0x6D, "1/1, 1/1, H." (_NRD=1C,6D,31,2F,31,2C,31,2F,31,2C,48,2E)

Left margin	ESC/P : 0x1B, 0x6C, margin width (_LFT=1B,6C,X0)
	        For margin width 10... 0x1B,0x6C,0x0A
	PR201 : 0x1B, 0x4C, margin width" (_LFT=1B,4C,X1)
	        For margin width 10... 0x1B,0x4C,"010"

8.8 Canceling the printing

The printing will be canceled if the ESC key is pressed during printing.

(1) Printing will stop when the ESC key is pressed.

(2) The message "PRINTING CANCELED" will appear when the printing is canceled.

9. Linking

Linking is a function used to merge the ladder file and message file described earlier into one file. The file created by merging the ladder file and message file is called a link file. Normally, this link file is transferred to the control unit. The ladder file and message file will remain intact even after they are linked.

There is also a function used to divide the link file into a ladder file and message file.

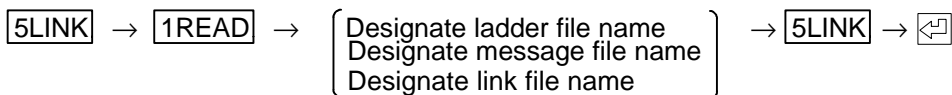Up to eight message files can be linked.

```
                          PLC FILE LINK

   #1   LINK      (LINK         )  (     21568/29952)
   2    LADDER    (LADDER       )  (      8000/16384)
   3    MESSG.1   (MSSG1        )  (      4096/ 4096)
   4    MESSG.2   (MSSG2        )  (      4096/ 4096)
   5    MESSG.3   (MSSG3        )  (      4096/ 4096)
   6    MESSG.4   (             )  (               )
   7    MESSG.5   (             )  (               )
   8    MESSG.6   (             )  (               )
   9    MESSG.7   (             )  (               )
   10   MESSG.8   (             )  (               )
   11   ADDRESS   (41000100     )
   12   C ADDR.   (0001FF00     )
   13   C ADDR.2  (0002FF00     )
   14   LANG. I/F (0003FF00     )
   15   C-MACRO   (0004FF00     )



   # (   ) NAME (         )

   1READ                5LINK      7DIVIDE
```

Link screen

9.1 File Linking

The procedures for creating a link file are described in this section.

[Basic operations]

$\boxed{\text{5LINK}}$ → $\boxed{\text{1READ}}$ → $\left\lbrace \begin{array}{l}\text{Designate ladder file name}\\\text{Designate message file name}\\\text{Designate link file name}\end{array}\right\rbrace$ → $\boxed{\text{5LINK}}$ → $\boxed{\hookleftarrow}$

[Operation example]

(1) The following setting area will appear when $\boxed{\text{5LINK}}$ → $\boxed{\text{1READ}}$ are pressed.

        # (   ) NAME (   )

                                     └─── File name **(Note 1)**

                   └─── Index

(2) Register the name of the ladder file created previously into "#2 LADDER". The used size will also appear at this time.

(3) Register the name of the message file created previously into "#3 MESSG.1". The used size will also appear at this time.

To create a multi-language compatible program, register the various language message files in "#4 to 10".

(4) Register the name of the link file in "#1 LINK".**(Note 2)**

(5) When $\boxed{\text{5LINK}}$ and $\boxed{\hookleftarrow}$ are pressed, the message "EXECUTION" will appear. When the linking is completed "COMPLETED" will appear.

The link file size will also appear at this time.

  (Note 1) Only files created in the same directory as PLC4B.EXE or files in the same directory can be designated here.

  (Note 2) If a link file with the same name as the ladder file is designated and the linking is completed normally, the link file cannot be actually sent. (The transmission capacity will be 0.)
Designate a name that differs from the ladder file.

9.1.1 Operation example

(1) Register the required file with `1READ`, and then press `5LINK` and `⏎`.

When `5LINK` is pressed, the setting area parentheses ( ) will appear, but nothing needs to be set here in particular.

```
                              PLC FILE LINK
#1      LINK        (LINK       )   (    21568/29952)
2       LADDER      (LAD0       )   (        0/   0)
3       MESSG.1     (MSG1       )   (     4096/ 4096)
4       MESSG.2     (MSG2       )   (     4096/ 4096)
5       MESSG.3     (MSG3       )   (     4096/ 4096)
6       MESSG.4     (           )   (               )
7       MESSG.5     (           )   (               )
8       MESSG.6     (           )   (               )
9       MESSG.7     (           )   (               )
10      MESSG.8     (           )   (               )
11      ADDRESS     (           )   (               )
12      C ADDR.     (           )
13      C ADDR.2    (           )
14      LANG. I/F   (           )
15      C-MACRO     (           )

        (    )

1READ                          5LINK              7DIVIDE
```

(2) The file indicated with an asterisk ∗ does not exist. Designate the file name again.
`1READ`
`2` `→` `L` `A` `D`
`5LINK`

```
                              PLC FILE LINK
#1      LINK        (LINK       )   (    21568/29952)
2       LADDER      (LAD0       )   (        0/   0)∗
3       MESSG.1     (MSG1       )   (     4096/ 4096)
4       MESSG.2     (MSG2       )   (     4096/ 4096)
5       MESSG.3     (MSG3       )   (     4096/ 4096)
6       MESSG.4     (           )   (               )
7       MESSG.5     (           )   (               )
8       MESSG.6     (           )   (               )
9       MESSG.7     (           )   (               )
10      MESSG.8     (           )   (               )
11      ADDRESS     (           )   (               )
12      C ADDR.     (           )
13      C ADDR.2    (           )
14      LANG. I/F   (           )
15      C-MACRO     (           )

        (    )                  PROG. NOT FOUND

1READ                          5LINK              7DIVIDE
```

(3) The file indicated with an asterisk ∗ already exists. Designate the file name again if required.

If `Y` and `⏎` are pressed, the file linking will start with the designated files.

```
                              PLC FILE LINK
#1      LINK        (LINK       )   (    21568/29952)∗
2       LADDER      (LAD        )   (     8000/ 16384)
3       MESSG.1     (MSG1       )   (     4096/ 4096)
4       MESSG.2     (MSG2       )   (     4096/ 4096)
5       MESSG.3     (MSG3       )   (     4096/ 4096)
6       MESSG.4     (           )   (               )
7       MESSG.5     (           )   (               )
8       MESSG.6     (           )   (               )
9       MESSG.7     (           )   (               )
10      MESSG.8     (           )   (               )
11      ADDRESS     (           )   (               )
12      C ADDR.     (           )
13      C ADDR.2    (           )
14      LANG. I/F   (           )
15      C-MACRO     (           )

        ( Y )                  VI-FILE IS EXIST.  OK (Y/N)

1READ                          5LINK              7DIVIDE
```

(4) The file linking will start when the following keys are pressed.
`1READ`
`1` `→` `L` `I` `N` `K` `1`
`5LINK` `⏎`

```
                              PLC FILE LINK
#1      LINK        (LINK1      )   (    21568/29952)
2       LADDER      (LAD        )   (     8000/ 16384)
3       MESSG.1     (MSG1       )   (     4096/ 4096)
4       MESSG.2     (MSG2       )   (     4096/ 4096)
5       MESSG.3     (MSG3       )   (     4096/ 4096)
6       MESSG.4     (           )   (               )
7       MESSG.5     (           )   (               )
8       MESSG.6     (           )   (               )
9       MESSG.7     (           )   (               )
10      MESSG.8     (           )   (               )
11      ADDRESS     (           )   (               )
12      C ADDR.     (           )
13      C ADDR.2    (           )
14      LANG. I/F   (           )
15      C-MACRO     (           )

        (    )                  EXECUTION

1READ                          5LINK              7DIVIDE
```

(5) The message "COMPLETED" will appear when the linking is completed.

```
                                              PLC FILE LINK
#1     LINK        (LINK1      )    (    21568/29952)
 2     LADDER      (LAD        )    (     8000/ 16384)
 3     MESSG.1     (MSG1       )    (     4096/ 4096)
 4     MESSG.2     (MSG2       )    (     4096/ 4096)
 5     MESSG.3     (MSG3       )    (     4096/ 4096)
 6     MESSG.4     (           )    (              )
 7     MESSG.5     (           )    (              )
 8     MESSG.6     (           )    (              )
 9     MESSG.7     (           )    (              )
10     MESSG.8     (           )    (              )
11     ADDRESS     (           )    (              )
12     C ADDR.     (           )
13     C ADDR.2    (           )
14     LANG. I/F   (           )
15     C-MACRO     (           )

        (   )                        COMPLETED

1READ                      5LINK             7DIVIDE
```

9.2 File Division

The procedures for dividing the link file will be described in this section.

[Basic operation]

$\boxed{\text{5LINK}} \rightarrow \boxed{\text{1READ}} \rightarrow$ (Designate link file name) $\rightarrow \boxed{\text{7DIVIDE}}$

[Operation example)

(1) The following setting area will appear when $\boxed{\text{5LINK}} \rightarrow \boxed{\text{1READ}}$ are pressed.

(   ) NAME (   )

                          └——————————— File name

            └——————————————————————————— Index

(2) Register the name of the link file created previously in "#1 LINK". The names and sizes of the ladder file and message files in the link file will appear.

(3) When $\boxed{\text{7DIVIDE}}$ and $\boxed{\hookleftarrow}$ are pressed " ( ) " will appear.

If $\boxed{\text{Y}}$ and $\boxed{\hookleftarrow}$ are pressed, the message "EXECUTION" will appear.

When the division is completed, the message "COMPLETED" will appear.

9.2.1 Operation example

(1) Read in the link file, and press the 7DIVIDE and ⏎ keys.

```
1READ

1  →  L  N  K  ⏎

7DIVIDE  ⏎
```

```
                                        PLC FILE LINK
#1    LINK      (LINK        )    (        /    )*
 2    LADDER    (            )    (        /    )
 3    MESSG.1   (            )    (        /    )
 4    MESSG.2   (            )    (        /    )
 5    MESSG.3   (            )    (        /    )
 6    MESSG.4   (            )    (             )
 7    MESSG.5   (            )    (             )
 8    MESSG.6   (            )    (             )
 9    MESSG.7   (            )    (             )
10    MESSG.8   (            )    (             )
11    ADDRESS   (            )    (             )
12    C ADDR.   (            )
13    C ADDR.2  (            )
14    LANG. I/F (            )
15    C-MACRO   (            )

      (    )                      PROG. NOT FOUND

1READ                  5LINK              7DIVIDE
```

(2) The file indicated with an asterisk ∗ does not exist. Designate the file name again.

```
1READ

1  →  L  I  N  K  1  ⏎

7DIVIDE  ⏎
```

```
                                        PLC FILE LINK
#1    LINK      (LINK1       )    (   21568/29952)
 2    LADDER    (LAD         )    (    8000/ 16384)
 3    MESSG.1   (MSG1        )    (    4096/ 4096)
 4    MESSG.2   (MSG2        )    (    4096/ 4096)
 5    MESSG.3   (MSG3        )    (    4096/ 4096)
 6    MESSG.4   (            )    (             )
 7    MESSG.5   (            )    (             )
 8    MESSG.6   (            )    (             )
 9    MESSG.7   (            )    (             )
10    MESSG.8   (            )    (             )
11    ADDRESS   (            )    (             )
12    C ADDR.   (            )
13    C ADDR.2  (            )
14    LANG. I/F (            )
15    C-MACRO   (            )

      (    )

1READ                  5LINK              7DIVIDE
```

(3) The file indicated with an asterisk ∗ already exists. Designate the file name again if required.
If Y and ⏎ are pressed, the file division will start with the designated files.

```
                                        PLC FILE LINK
#1    LINK      (LINK1       )    (   21568/29952)
 2    LADDER    (LAD         )    (    8000/ 16384)*
 3    MESSG.1   (MSG1        )    (    4096/ 4096)*
 4    MESSG.2   (MSG2        )    (    4096/ 4096)*
 5    MESSG.3   (MSG3        )    (    4096/ 4096)*
 6    MESSG.4   (            )    (             )
 7    MESSG.5   (            )    (             )
 8    MESSG.6   (            )    (             )
 9    MESSG.7   (            )    (             )
10    MESSG.8   (            )    (             )
11    ADDRESS   (            )    (             )
12    C ADDR.   (            )
13    C ADDR.2  (            )
14    LANG. I/F (            )
15    C-MACRO   (            )

      (    )        VI-FILE IS EXIST.  OK (Y/N)

1READ                  5LINK              7DIVIDE
```

(4) The file division will start when the following keys are pressed.
```
Y  ⏎
```

```
                                        PLC FILE LINK
#1    LINK      (LINK1       )    (   21568/29952)
 2    LADDER    (LAD         )    (    8000/ 16384)
 3    MESSG.1   (MSG1        )    (    4096/ 4096)
 4    MESSG.2   (MSG2        )    (    4096/ 4096)
 5    MESSG.3   (MSG3        )    (    4096/ 4096)
 6    MESSG.4   (            )    (             )
 7    MESSG.5   (            )    (             )
 8    MESSG.6   (            )    (             )
 9    MESSG.7   (            )    (             )
10    MESSG.8   (            )    (             )
11    ADDRESS   (            )    (             )
12    C ADDR.   (            )
13    C ADDR.2  (            )
14    LANG. I/F (            )
15    C-MACRO   (            )

      (    )                      EXECUTION

1READ                  5LINK              7DIVIDE
```

(5) The message "COMPLETED" will
appear when the division is completed.

```
                            PLC FILE LINK
#1    LINK        (LINK1      )    (   21568/29952)
 2    LADDER      (LAD        )    (    8000/ 16384)
 3    MESSG.1     (MSG1       )    (    4096/ 4096)
 4    MESSG.2     (MSG2       )    (    4096/ 4096)
 5    MESSG.3     (MSG3       )    (    4096/ 4096)
 6    MESSG.4     (           )    (              )
 7    MESSG.5     (           )    (              )
 8    MESSG.6     (           )    (              )
 9    MESSG.7     (           )    (              )
10    MESSG.8     (           )    (              )
11    ADDRESS     (           )    (              )
12    C ADDR.     (           )
13    C ADDR.2    (           )
14    LANG. I/F   (           )
15    C-MACRO     (           )

      (    )                         COMPLETED

1READ            5LINK      7DIVIDE
```

9.3 Others

9.3.1 Address

This is not used with the MELDAS 64 Series and MELDASMAGIC 64 Series.

9.3.2 C Address

This is not used with the MELDAS 64 Series and MELDASMAGIC 64 Series.

9.3.3 C Address 2

This is not used with the MELDAS 64 Series and MELDASMAGIC 64 Series.

9.3.4 Language I/F

This is not used with the MELDAS 64 Series and MELDASMAGIC 64 Series.

9.3.5 C-Macro

This is not used with the MELDAS 64 Series and MELDASMAGIC 64 Series.

10. Editing an Existing File

The methods for editing a link file, ladder file or message file created previously will be described in this section. To edit these files, they must be placed in the root directory of the disk in which the PLC development software is installed.

10.1 Link File

The existing link file is edited after dividing it into the ladder file and message file.

[Basic operation]
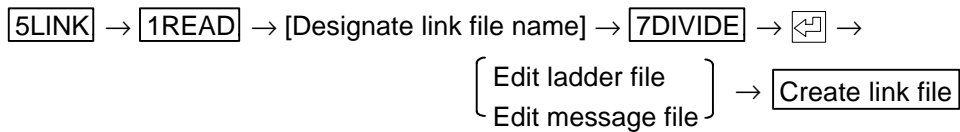
$\boxed{\text{5LINK}} \rightarrow \boxed{\text{1READ}} \rightarrow$ [Designate link file name] $\rightarrow \boxed{\text{7DIVIDE}} \rightarrow \boxed{\leftarrow} \rightarrow$

$\left[\begin{array}{l} \text{Edit ladder file} \\ \text{Edit message file} \end{array} \right] \rightarrow \boxed{\text{Create link file}}$

[Operation procedures]

(1) The following setting area will appear when $\boxed{\text{5LINK}} \rightarrow \boxed{\text{1READ}}$ are pressed.

   # ( ) NAME (   )

(2) When the following is input, the ladder file name, message file name, capacity of each file and used size will appear.

   # (1) NAME (*Name of link file to be edited*) $\boxed{\leftarrow}$

(3) The parentheses ( ) will appear when the keys are pressed in the order of $\boxed{\text{7DIVIDE}} \rightarrow \boxed{\leftarrow}$.

(4) The message "EXECUTION" will appear when $\boxed{\text{Y}} \rightarrow \boxed{\leftarrow}$ are pressed, and the division will start.

   The message "COMPLETED" will appear when the division is completed.

(5) When $\boxed{\text{4LADDER}} \rightarrow \boxed{\text{2CIRCUIT}}$ are pressed on the Title screen, the ladder file can be edited.

(6) When $\boxed{\text{1MESSAGE}} \rightarrow \boxed{\text{(*Each message file*)}}$ are pressed on the Title screen, the message file can be edited.

(7) When completed with the editing, return to the Title screen, and press $\boxed{\text{5LINK}} \rightarrow \boxed{\text{5LINK}} \rightarrow \boxed{\leftarrow}$.

   The message "EXECUTION" will appear, and the edited files will be linked.
   The message "COMPLETED" will appear when the linking is completed.

10.2 Ladder File

An existing ladder file is edited with the following procedure.

[Basic operation]

$\boxed{\text{5LINK}}$ → $\boxed{\text{1READ}}$ → [Designate ladder file name] → [Edit ladder file]

[Operation procedures]

(1) The following setting area will appear when $\boxed{\text{5LINK}}$ → $\boxed{\text{1READ}}$ are pressed.

( ) NAME (   )

(2) When the following is input, the ladder file capacity and used size will appear.

(2) NAME (*Name of ladder file to be edited*) $\boxed{\hookleftarrow}$

(3) When $\boxed{\text{4LADDER}}$ → $\boxed{\text{2CIRCUIT}}$ are pressed on the Title screen, the ladder file can be edited.

10.3 Message File

An existing message file is edited with the following procedure.

[Basic operation]

$\boxed{\text{5LINK}}$ → $\boxed{\text{1READ}}$ → [Designate message file name] → [Edit message file]

[Operation procedures]

(1) The following setting area will appear when $\boxed{\text{5LINK}}$ → $\boxed{\text{1READ}}$ are pressed.

( ) NAME (   )

(2) When the following is input, the message file capacity and used size will appear.

(3) NAME (*Name of message file to be edited*) $\boxed{\hookleftarrow}$

(3) When $\boxed{\text{1MESSAGE}}$ → $\boxed{(\textit{Each file name})}$ are pressed on the Title screen, the message file can be edited.

11. Converting Ladder Files

11.1 Outline

The conversion tool required to convert the PLC program files (link files) developed on a personal computer to a format that can be handled by the MELDAS 64 Series is described in this section.
(File name: chgcx. exe. Hereafter, conversion tool.)

Using this tool, PLC programs developed on a personal computer and MELDAS500/50 Series ladder assets can be used and smoothly transferred to the M64 Series.

(Precautions)

(1) PLC program files converted and created with this conversion tool cannot be corrected with the PLC development software.

(2) PLC program files converted and created with this conversion tool cannot be operated by any control unit (MELDAS500/50 Series) besides the MELDAS64 Series, even if transferred to such a unit.

11.2 Conversion operation

The basic command format is shown below. Items in < > brackets can be omitted.
(Refer to section "11.3.1 Normal Operation Example" for an operation example.)

CHGCX < - option > in_file<.ext> <out_file.ext >

< − option >   : Designate the conversion option. (Can be omitted.)
　　　　　　　　　　　　−m : Message deletion mode. Delete the link file message.
　　　　　　　　　　　　−h : Output mode in which the header is not deleted. Refer to section "11.4.5 Header".
　　　　　　　　　　　　　　(Always add this option in the MELDAS64 Series.)
　　　　　　　　　　　　−n : Ladder/message name conversion mode. Refer to section "11.3.2 Operation Example for Link File Conversion".
　　　　　　　　　　　　−p : Mode in which the NOP commands after the END command are not deleted.
　　　　　　　　　　　　　　(Always add this option in the MELDAS64 Series.)
　　　　　　　　　　　　−c : Mode in which ANDSW commands are used for software command operation codes.
in_file< . ext > : Designate the file name to be converted. (Extensions can be omitted.)
　　　　　　　　　　Refer to section "11.4.1 File Name Restrictions Before Conversion" for details on file names.
< out_file. ext > : Designate the file name after conversion. (Can be omitted.)
　　　　　　　　　　If omitted, the file name after conversion will be the same as that in the in_file, with extensions added.
　　　　　　　　　　The new file after conversion will be created in the same directory as the file before conversion.
　　　　　　　　　　Refer to section "11.4.2 File Name Restrictions After Conversion" for details on file names.

Always add the above options "−h" and "−p" in the MELDAS64 Series.

11.3 Operation Examples

11.3.1 Normal Operation Example

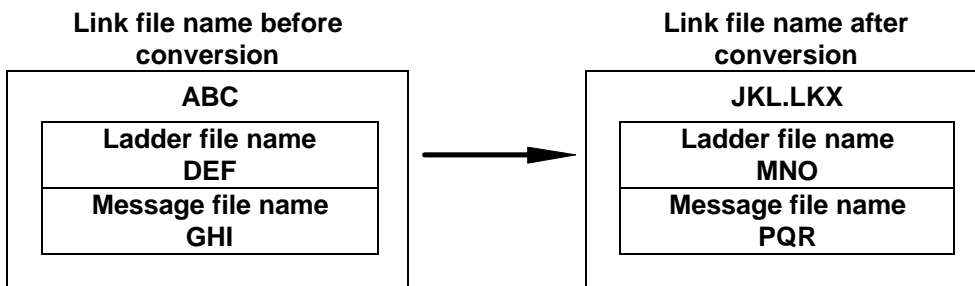The conversion operation is carried out as follows when the file type to be converted is a ladder file.

Ladder file name DEF	⟶	Ladder file name MNO.LDX
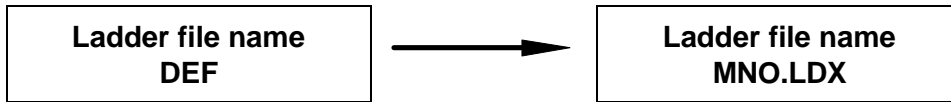
The conversion operation for the above conversion example is as follows. (In this case, options "h" and "p" are added.)

> **CHGCX** $-$**HP DEF MNO**

11.3.2 Operation Example for Link File Conversion

When the file type to be converted is a link file, the file names of the ladder file and message file included in the link file can be converted using the option "n". If option "n" is not used, the file name before conversion will be used as is.

Link file name before conversion		Link file name after conversion
ABC	⟶	**JKL.LKX**
Ladder file name DEF		Ladder file name MNO
Message file name GHI		Message file name PQR

The conversion operation for the above conversion example is as follows.

1) Convert the file using the command "CHGCX".

> **A**> **CHGCX** $-$**N ABC JLK**

2) Designate the ladder file name of the converted file.
   At this time, the same name can be designated by inputting a " . " (period).

> **Old ladder file name is DEF**
> **Please type new ladder file name <u>MNO</u>**

3) Designate the message file name of the converted file.
   At this time, the same name can be designated by inputting a " . " (period).

> **Old message file name is GHI**
> **Please type new ladder file name <u>PQR</u>**

11.4 Restrictions

The following restrictions apply to designated file names before and after conversion.

11.4.1 File Name Restrictions Before Conversion

1) The file name must be expressed as [file name. extension]. Extensions are not required.
2) The file name must be within 128 characters (including drive name, backslashes, extensions and periods.)
3) All characters in the file name must be characters that can be used normally.

11.4.2 File Name Restrictions After Conversion

1) The file name must be designated as [file name. extension].
   When the extension is omitted, an extension is automatically depending on the contents of the file before conversion.
   The following shows details of the automatically added extensions.

Details of the file before conversion	File name after conversion
Link file	[file name. LKX]
Ladder file	[file name. LDX]

2) The file name must be within 128 characters (including drive name, backslashes, extensions and periods.)
3) All characters in the file name must be characters that can be used normally.

11.4.3 File Attributes of Output Files

Attributes dedicated for reading are added.

11.4.4 Device No.

An error will occur if an attempt is made to convert ladder files using the following device Nos. Change to another device No. before conversion.

X4C0 ~ X4FF (machine input spare)

Y540 ~ Y57F (machine output spare)

S140 ~ S1FF (rotation tool input/output spare)

I400 ~ I4BF (input signal spare)

11.4.5 Header

Header address information is required in the M64 Series. An output in which the header is not deleted can be obtained by adding the "-h" option. Always add the "-h" option in the M64 Series.

11.4.6 Check Sum

The check sum of the data from the head to end (excluding the header) is calculated and stored in the entry table.

11.4.7 Message data

Individual ladder files and ladder message link files can be converted.
Single files of message data and help messages cannot be converted.

12. Transferring Ladder Files (MELDAS 64)

12.1 Outline

The communication program required to transmit and receive the PLC program files (ladder and message link file) between the [NC] and [personal computer] will be described in this section.
(File name: COMU.EXE. Hereafter, communication program.)

A PLC program file (ladder + message) developed on the personal computer can be transferred to the control unit.

(Note 1)

The following two types of files can be communicated.
  (1) Files generated by linking with PLC development software and converted into MELDAS64 Series format.
  (2) PLC program created with NC onboard PLC development function and input/output with the maintenance data input function (#99 250).

When linking with the PLC development software, if a link file with the same name as the ladder file is designated, the file cannot be sent.
(The linking will be completed normally, but the transmission capacity will be 0.)
Do not designate the same name as the ladder file for the link file name.

12.2 Starting Method

Type in the start command as shown below.

```
A:\>comu ⏎
```

The communication program screen will appear when the communication program starts.

12.3 Communication Execution Sequence

12.3.1 From [personal computer] to [NC]

The PLC program (ladder, message) is created with the following procedure using the PLC development software.
The file to be transmitted must be a file that has been linked with the PLC development software and converted into MELDAS64 Series format.
(Read the details of **(Note 1)** carefully.)

(1) Start the communication program. (Personal computer side)

(2) Check the NC RS-232-C parameters (baud rate, parity, etc.), and set them according to the communication program.

(3) Set the Maintenance Data Input Wait state on the NC DATA INPUT screen.

```
#  (99)  (250)   ←————— (2) Designate the reception file (maintenance).

INPUT    OUTPUT ←————— (1) Designate INPUT.

                        (3) The reception wait state will be entered with  INPUT .
```
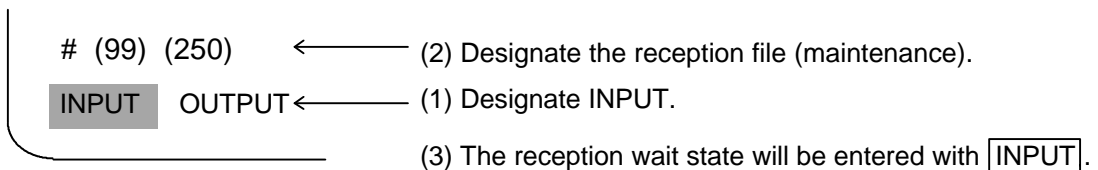
Transmission will start from the communication program.
When the communication program displays "Transmitting Completed", the transmission has been completed normally.

12.3.2 From [NC] to [personal computer]

(1) Start the communication program. (Personal computer side)

(2) Check the NC RS-232-C parameters (baud rate, parity, etc.), and set them according to the communication program.

(3) Set the communication program to the reception wait state.

(4) Output the maintenance data on the NC DATA OUTPUT screen.

```
#  (99)  (250)   ←————— (2) Designate the transmission file (maintenance).

INPUT    OUTPUT ←————— (1) Designate OUTPUT.

                        (3) The transmission will state with  INPUT .
```
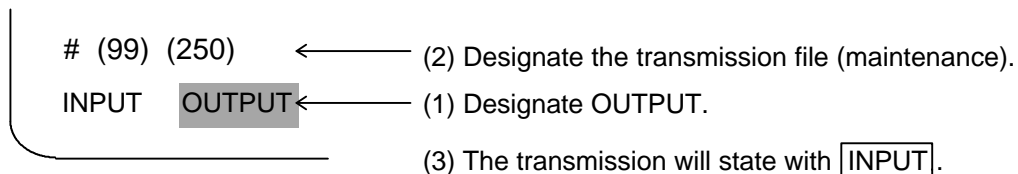
When the communication program displays "Receiving Completed", the reception has been completed normally.
The received file cannot be read into the PLC development software.

12.4 Communication Program Screen Configuration

The following screen will appear when this program (COMU.EXE) is started.

```
              1         2         3         4         5         6         7         8
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1  *** < PLC Program Transfer Utility [BND 635W035- ▢ > *      (MELDAS64 )
 2
 3
 4    Copyright (C) MITSUBISHI ELECTRIC CORPORATION 1994. all rights reserved
 5
 6    Baud Rate     :  9600     4800      1200      600       300
 7    Data Length   :  8
 8    Stop Bit      :  1        2
 9    Parity        :  0 (ODD)  E (EVEN)  N (NONE)
10    Monitor       :  OFF      ON
11
12    Com. Mode     :  TRANSMIT  RECEIVE
13
14    File Name     :  —
15
16    Execute ?     :  START     MS-DOS
17
18
19    [Monitor]
20    Transmit Data :
21    Receive  Data :
22
23    Message       :
24
25          Arrow Key [UP][DOWN][LEFT][RIGHT]      :      Item Select
```

<Fig. 12.1>

(Note 2)
> The system number shown on the first line of the screen is BND-655W036-**for the PC-AT compatible unit communication program.

(Note 3)
- The cursor used to select the items by moving vertically (using the $\uparrow$ and $\downarrow$ keys) will be called the "vertical cursor" hereafter.
- The highlighted items are the default settings.
- The basic operation can be carried out by selecting the items with the arrow keys.

(Note 4) Use the following keys when the arrow keys cannot be used.

$\boxed{\downarrow}$ = $\boxed{\text{CTRL}}$ + $\boxed{\text{J}}$

$\boxed{\uparrow}$ = $\boxed{\text{CTRL}}$ + $\boxed{\text{K}}$

$\boxed{\leftarrow}$ = $\boxed{\text{CTRL}}$ + $\boxed{\text{H}}$

$\boxed{\rightarrow}$ = $\boxed{\text{CTRL}}$ + $\boxed{\text{L}}$

12.5 Communication Program Operation Methods

12.5.1 SIO setting

Refer to <Fig. 12.1> for the default settings.

The setting can be changed with the arrow keys ($\boxed{\uparrow}$ • $\boxed{\downarrow}$ • $\boxed{\leftarrow}$ • $\boxed{\rightarrow}$).

The setting details are as follow.

Baud Rate	Baud rate setting
Data Length	Number of data bits
Stop Bit	Number of stop bits
Parity	Parity bit setting
Monitor	Setting of transmission/reception data display

12.5.2 Com.Mode (Communication mode)

The communication mode (TRANSMIT/RECEIVE) is set.

The setting can be changed with the arrow keys ($\boxed{\leftarrow}$ • $\boxed{\rightarrow}$).

TRANSMIT Transmission mode (personal computer → controller)

RECEIVE   Reception mode (personal computer ← controller)

12.5.3 File Name (File name input)

When the vertical cursor is set to "File Name", the cursor will appear at the input area, and "Enter File Name" will appear in the message area.

Type in the file name. (The $\boxed{\text{BS}}$ key is valid.)

The $\boxed{\hookleftarrow}$ key is identical to the $\boxed{\downarrow}$ key operation. (This key will not function as the fixing key.)

To trans | B:\USER\DATA1.DAT | e file to be transmitted.

 (Example) To transmit the file "DATA1.DAT" under the directory "USER" in drive B.

To rece | B:\USER\READ1.DAT | file to be received.

 (Example) To save the reception data in the file "READ1.DAT" under the directory "USER" in drive B.

12.5.4 Execute?

If the ⏎ key is pressed when "START" is highlighted, the transmission or reception will start according to the "Com.Mode" setting.

[Transmitting] The designated file will be transmitted to the control unit.

> **(Note 5)**
>
> If the designated transmission file is not found, error message 7 (refer to Table 12.1) will appear.

[Receiving] The reception data will be saved in the designated file.

> **(Note 6)**
> - If a file with the same name exists, it will be overwritten.
> - If the file is not found, a file will be created.
> - A directory will not be created. If a directory that does not exist is designated, error message 8 (refer to Table 12.1) will appear.
> - If the FLD is write-protected, a message will be output from MS-DOS. (The screen will be distorted in this case, but work can be continued by following the given instructions.)

If the ⏎ key is pressed when "MS-DOS" is highlighted, this program will be quit, and the system will return to MS-DOS.

12.5.5 Monitor

TRANSMIT DATA   : The transmission data will be displayed when the Monitor is ON.
RECEIVE DATA   : The reception data will be displayed when the Monitor is ON.

12.5.6 Message

Instructions or warnings will be displayed in this area as required.
Refer to section "12.6 Message List" for details on the messages.

> **(Note 7)**
>
> If the communication is not completed normally (including when an error is judged during [COMP]), the personal computer may not accept the STOP key (including Ctrl + C).
>
> Reset the personal computer if communication is not completed normally.

12.6 Message List

The following messages will appear as required in the Message area of <Fig. 12.1>.

No.	Message	Meaning
01	Transmitting	File is being transmitted.
02	Transmitting Completed	The transmission has been completed.
03	Receiving	The file is being received.
04	Receiving Completed	The reception has been completed.
05	No Answer	There is no answer. (The line is not connected.)
06	Communication Error	Communication error (overrun, framing error, etc.)
07	No File	The designated file or directory was not found. (During transmission)
08	File Create Failure	A directory that does not exist was designated, and a file was not created. (During reception)
09	Enter File Name	Request for entering of file name. (When the cursor is at "File Name")
10	Press Enter Key (Communication Start)	Communication will start when ⏎ key is pressed. (When cursor is at "Execute?", and "START" is selected.)
11	Press Enter Key (Return to MS-DOS)	MS-DOS will be returned to when ⏎ key is pressed. (When cursor is at "Execute?", and "MS-DOS" is selected.)
12	H-Parity Error	A horizontal parity error occurred in the reception data. (Data Bit 7 is the horizontal parity.)
13	File Create Failure (Capacity)	This message appears when the storage medium (FLD, etc.) capacity is low during reception.
14	File Modify OK? (Y/N) (Note 10)	This message will appear to confirm whether to clear the area at 00H when the received file's 164H (140H+24H) to 223H (1FFH+24H) is not 00H. (Press Y or N)
15	Receiving Completed (Copying File Completed)	This message appears when the corrections to message 14 have been completed. The "04 Receiving Completed" message appears when the check is passed.

(Note 8)

The message area will be cleared when the vertical cursor (cursor that moves with $\downarrow$ and $\uparrow$) is moved.

(Note 9)

If reception is executed while the FLD is write-protected, a warning message will be output from the MS-DOS, so the screen will be partially distorted.

If the FLD write-protection tab is released and the instructions are followed, work can be continued.

(Note 10)

When message 14 appears:

This message means that when the 164H to 223H bytes from the head of the received file were checked, data that was not the binary [00H] was found.
If this area is not [00H], the file may not run correctly on the "PLC development software". Thus, this message will appear and request the user to take measures.

When [Y] is pressed, 164H to 223H in the reception file will be unconditionally changed to [00H].
(The uncorrected file will not be saved.)

An error will always occur if [COMP] is carried out on the corrected file.

When [N] is pressed, nothing will be changed, so there will be no guarantee that the file will run correctly on the "PLC development software".

13. Transferring Ladder Files (MELDASMAGIC 64)

13.1 Outline

The PLC program data (ladder and message) created with "PLC4B" on the personal computer can be transferred to the NC (MELDASMAGIC 64).
The PLC program data created with the PLC development function (onboard) on the NC side can also be transferred to the personal computer.

In this section, the methods for transmitting and receiving the PLC program data (ladder and message) between the [NC] and [personal computer] using the transfer tool (magictrs.exe) will be described.
The data is transferred from the personal computer to the NC with the following procedure when using the MELDASMAGIC 64.

1 | Create PLC program file *1 | The PLC program file is created using "PLC4B". (*1)
Refer to sections "4. Start Up" to "9. Linking".

2 | Transfer to NC *3 | Using the MELDASMAGIC transfer tool (magictrs.exe), the converted PLC data is transferred to the NC. (*3)
Refer to the section "12.4 Transferring PLC program data".
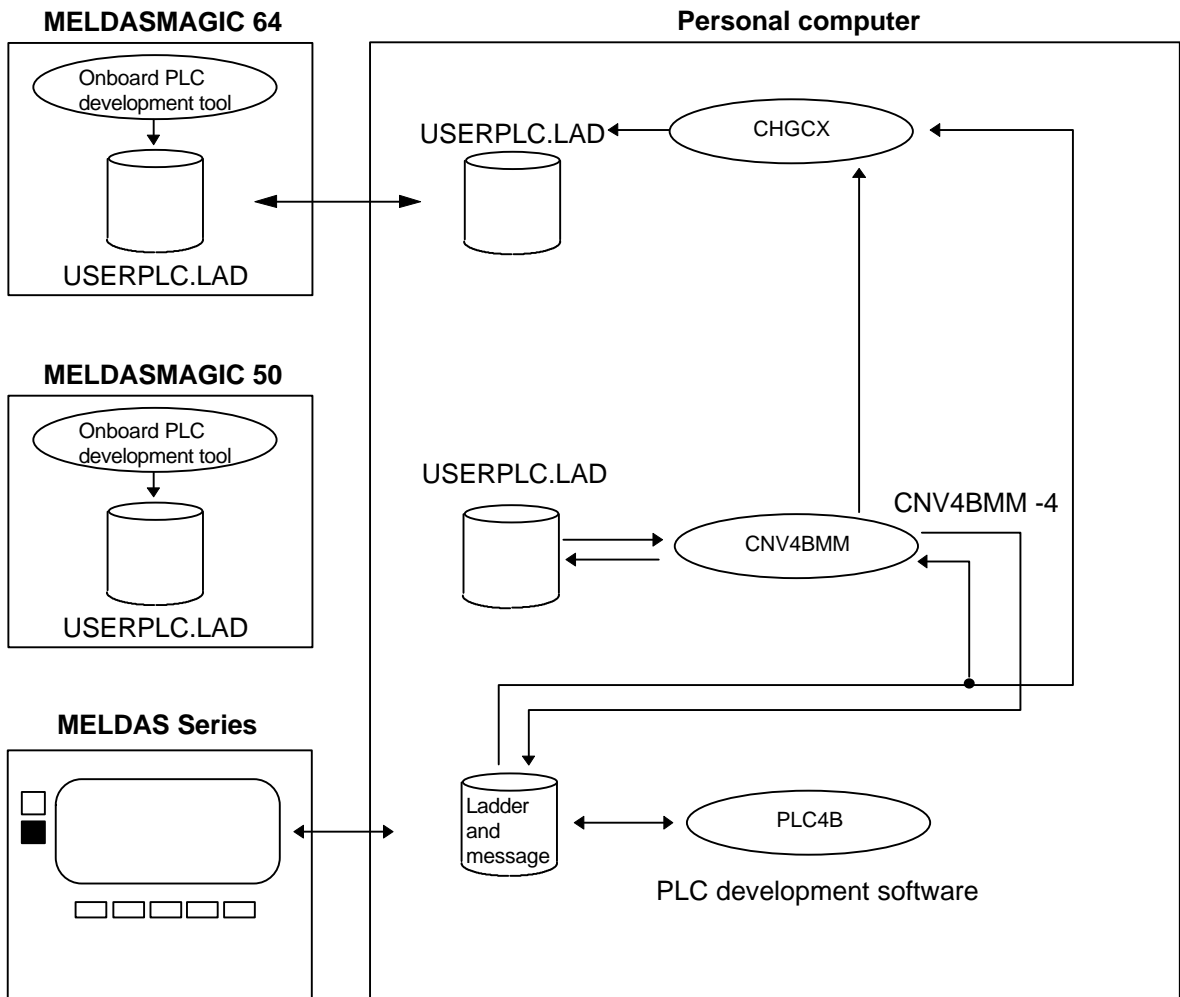
(Note 1) The following two types of files can be communicated.
1. A file generated by linking with the PLC development software and converted into MELDASMAGIC format.
2. PLC program generated with MELDASMAGIC 64 onboard (M64 monitor) that has been input/output with the maintenance data input function (#99 250).

(Note 2) When linking a ladder file and message file with the PLC development software (PLC4B), do not designate the same name as the ladder file for the link file name.
If a link file with the same name as the ladder file is designated, the file cannot be transferred.
(The linking will be completed normally, but the transmission capacity will be 0.)

13.2 Operating Environment

13.2.1 Personal computer operating conditions

(1) PC-9800 Series

Use a PC-9800 Series that is running "MS-DOS Version 3.3 and above".

Caution
The PLC development software (PLC4B) can be used on the PC-9800 Series, but the conversion tool (CNV4BMM) is dedicated for the PC/AT compatible unit, and thus will not run on the PC-9800 Series. Before converting a PLC program developed with the PC-9800 Series, load it into a PC/AT compatible unit using a FLD, etc.

(2) PC/AT compatible unit

Use an IBM-PC/AT or compatible unit that is running "MS-DOS Version 5 and above".

13.2.2 Required software

(1) Transfer tool (MAGICTRS.EXE)

This tool is located in the "UTILITY disk".
Use the file transfer function in the MELDASMAGIC 64 MMI software (option).
Refer to the "Utility Operation Manual" for details on installing the "UTILITY disk" and the operation of the tools, etc.

13.3 Converting PLC Program Data

Convert the PLC program data (ladder and message link file) created with the PLC4B into a format that can be handled by the MELDASMAGIC using the following procedure.

This procedure can also be used to convert a file used with the MELDASMAGIC into a format that can be handled by the PLC4B (format that can be used with conventional NC).

13.3.1 Conversion procedures

Convert the data with the following procedures.

(1) Start MS-DOS.

(2) Start the conversion tool (cnv4bmm.exe), and type in the following command.

(The following example shows when the tool is installed in the C:\directory.)

```
C:\>C:\meltools\dos\cnv4bmm.exe ⏎
```

The following type of explanation will appear.

```
[CNV4BMM]  PLC4B format File Converter  Ver. A0
    Convert  PLC4B format File into MELDASMAGIC format File.
  Usage: CNV4BMM [-option] input_file [output_file]
          option : [−m | −4 | −a]
          −m : convert PLC4B format into MELDASMAGIC format
          −4 : convert MELDASMAGIC format into PLC4B format
          −a : automatically convert into another format (default)
    input_file : input file name
   output_file : output file name
          If no output file name ordered, it use "USERPLC.LAD".
  Copyright (C) 1995  Mitsubishi Electric Corporation  ALL Rights Reserved.
```

The following three methods can be used for conversion.
    −m : Convert from PLC4B format into MELDASMAGIC format
    −4 : Convert from MELDASMAGIC format into PLC4B format
    −a : Automatically judge input_file format and convert

(3) Following the explanation above, designate the file name and option, and type in the command.

(Example) To convert a file (c:\plc1) created with the PLC development software, etc., and located in the drive C root directory into a file that can be input into MELDASMAGIC.

```
C:\>C:\meltools\dos\cnv4bmm.exe -m plc1 ⏎
```

A file named USERPLC.LAD will be created in the execution directory (C:\).
(In this example, output_file has been omitted, so the default file name will be assigned.)

Caution

1) File name designation
   - If the same name is designated for the input file name and output file name, an error will occur.
   - If the input file name is omitted, an error will occur.
   - If the output file name is omitted, the output file name will be set to "USERPLC.LAD".
   - All lowercase characters in the input or output file name will be interpreted as uppercase characters.
2) Command inputs (option designation)
   - The option characters are case sensitive. For example, -A or -M will cause an error.
   - Only one of the options (-m, -4 or -a) can be designated. If two or more options are designated, an error will occur.
   - If no option (-m, -4 or -a) is designated, the same process as when -a is designated will take place.
3) File size
   - The MELDASMAGIC can handle up to 4K steps of ladder data. When converting ladder data developed with the conventional model (MELDAS 50 Series, MELDAS 500 Series), or ladder data developed with the PLC4B, take care so that this size is not exceeded.

   If 4K steps is exceeded, the excessive data will not be converted.

13.4 Transferring PLC Program Data

Data is transferred between the personal computer and NC with the following procedure using the MELDASMAGIC transfer tool (magictrs) or the MELDASMAGIC MMI software.

13.4.1 From [personal computer] to [NC Card]

(1) Stop the PLC and enter the emergency stop state. The following method can be used.

    Method) Stop the PLC from the onboard PLC screen.

        When the function keys are pressed in the order of F12 → F3 (File) → F4 (4RUN/SP), the following message will appear. Set 1.

            0: RUN   1: STOP

    (Note) When the PLC program data has been created using the PLC4B and is to be converted and transferred, the PLC is stopped. Check the state before proceeding with the work.

(2) Start the transfer tool with the following procedure.

  [When using MMI software file transfer window]

    (Note) Start with this method when using the MELDASMAGIC 64.

      The MELDASMAGIC MMI software is an option specification.

    1) Start Windows.

    2) Start the MELDASMAGIC MMI software.

    3) Click on the [Monitor] - [Search] and [File Transfer] commands, and open the File Transfer window.

  [When using transfer tool (magictrs.exe)]

    (Note) This method can be used to start when using MELDASMAGIC 64.

    1) Start Windows.

    2) Select the [File] → [Run...] menus under the Program Manager.

    3) Type the following in the command line, and click on the [OK] button.
   *MELDASMAGIC MMI Install Destination Directory*\MAGIC.EXE/M=*NC Card No.*

 **(Example)** When MELDASMAGIC MMI software is installed in C:\.

```
C:\>MAGICTRS.EXE  ⏎
```

(4) Select the NC side directory "M01:\LAD" with the following procedure.
    1) Double-click on "M01:\".
    2) Double-click on "LAD".

(5) Designate the PLC file to be transferred (file that has been converted with CNV4BMM) with the following procedure.
    1) Double-click on the directory containing the PLC file to be transferred.
    2) Select and click on the file in the file list.
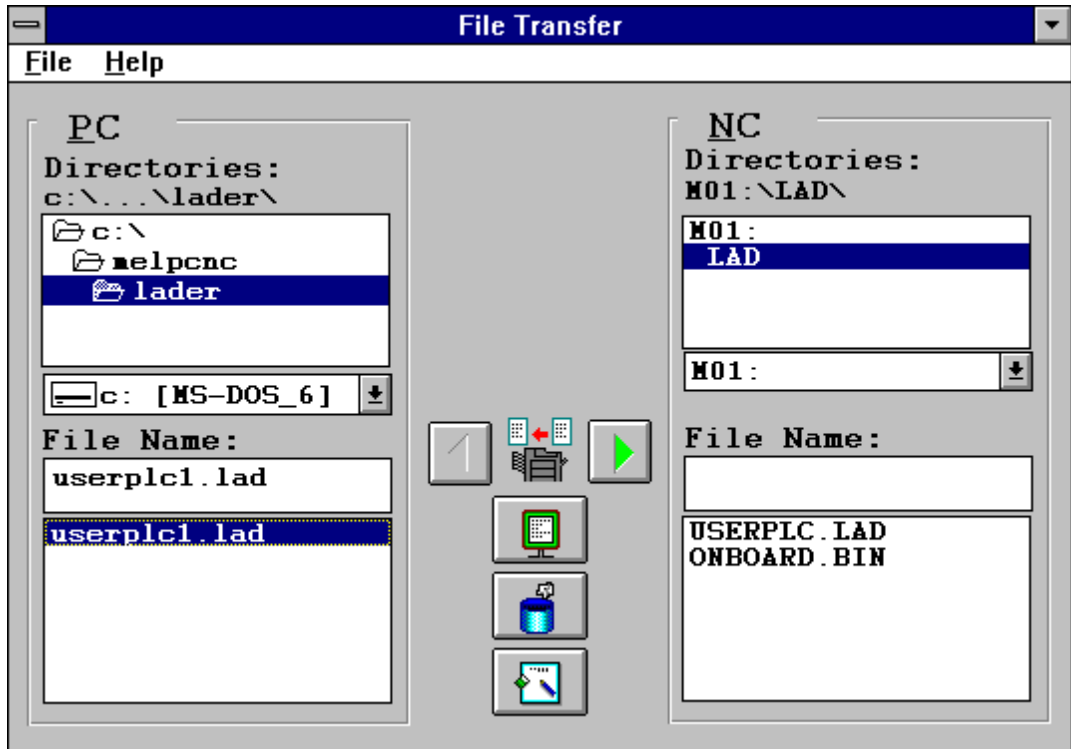
(6) Click on the copy button PC → NC (▷).
(7) The Copy window will open. The file name designated in step (5) will appear in the text box. Change this to "USERPLC.LAD".
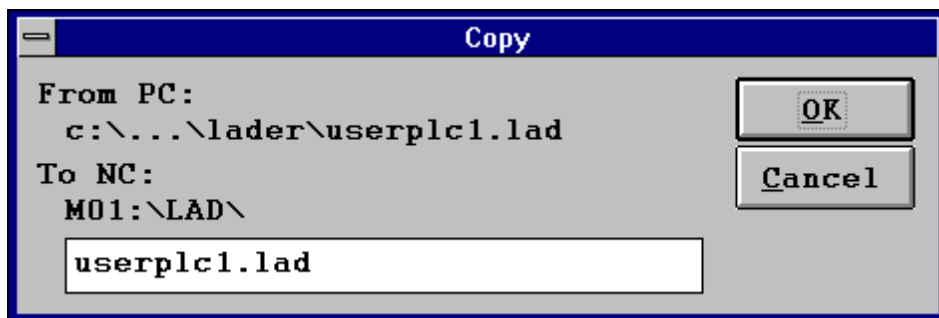
Change the file name.
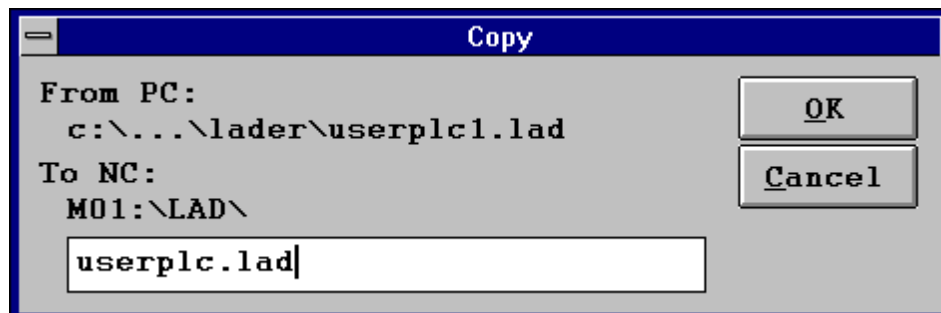
(8) Click on the [OK] button.

   If an error message does not appear, the transmission is completed.

(9) Start the PLC and confirm that there is no error. The PLC can be started with the following two methods.

   (Note 1) If "Method 1" was selected for stopping the PLC in step (1), select Method 1. If "Method 2" was selected, select Method 2.

   Method 1) Return the RS611 card rotary switch CS2 to 0.

   Method 2) Stop the PLC from the onboard PLC screen.

   When the function keys are pressed in the order of F12 → F3 (file) → F4 (4RUN/SP), the following message will appear. Type in 0.

   0: RUN   1: STOP

CAUTION

(1) Always set the file name to "USERPLC.LAD" when transferring from the PC to the NC.

(2) Refer to section "13.5 Message List" for details on the error messages.

13.4.2 From [NC Card] to [personal computer]

(1) Stop the PLC and enter the emergency stop state.
(Refer to the section "13.4.1 From [personal computer] to [NC Card]" for details.)

(2) Start the transfer tool.
(Refer to the section "13.4.1 From [personal computer] to [NC Card]" for details.)

(3) Select the PLC file "M01:\LAD\USERPLC.LAD" (NC side) to be transferred with the following
procedure.
(1) Double-click on "M01:\".
(2) Double-click on "LAD".
(3) Select and click on "USERPLC.LAD" in the file list.

(4) Designate the file transfer destination (personal computer side directory).
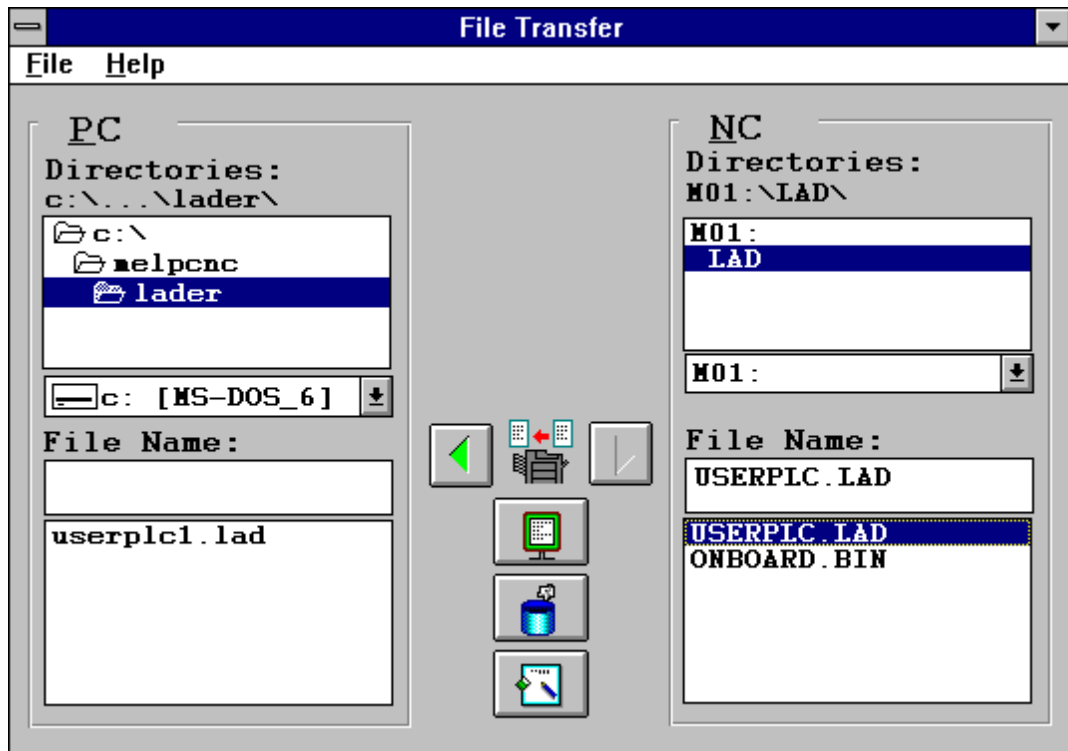
(5) Click on the copy button NC → PC (◁).

(6) The Copy window will open. The "USERPLC.LAD" will appear in the text box. Change the file name
here.

(7) Click on the [OK] button. If an error message does not appear, the transmission is completed.

(8) Start the PLC and confirm that there is no error.
(Refer to the section "13.4.1 For [personal computer] to [NC Card] for details.)

CAUTION
(1) The following steps must be taken to edit the file transferred to the personal computer with the PLC4B.   1. Using the conversion tool CNV4BMM, convert the transferred file into a format that can be read by the PLC4B. (Refer to the section "13.3 Converting PLC program data".)   2. Divide the link file (converted) into a ladder file and message file.   (Refer to the section "9.2 File division".)   (2) Refer to section "13.5 Message List" for details on the error messages.

13.5 Message List

The following messages will appear as required when the data is transferred with the File Transfer window.

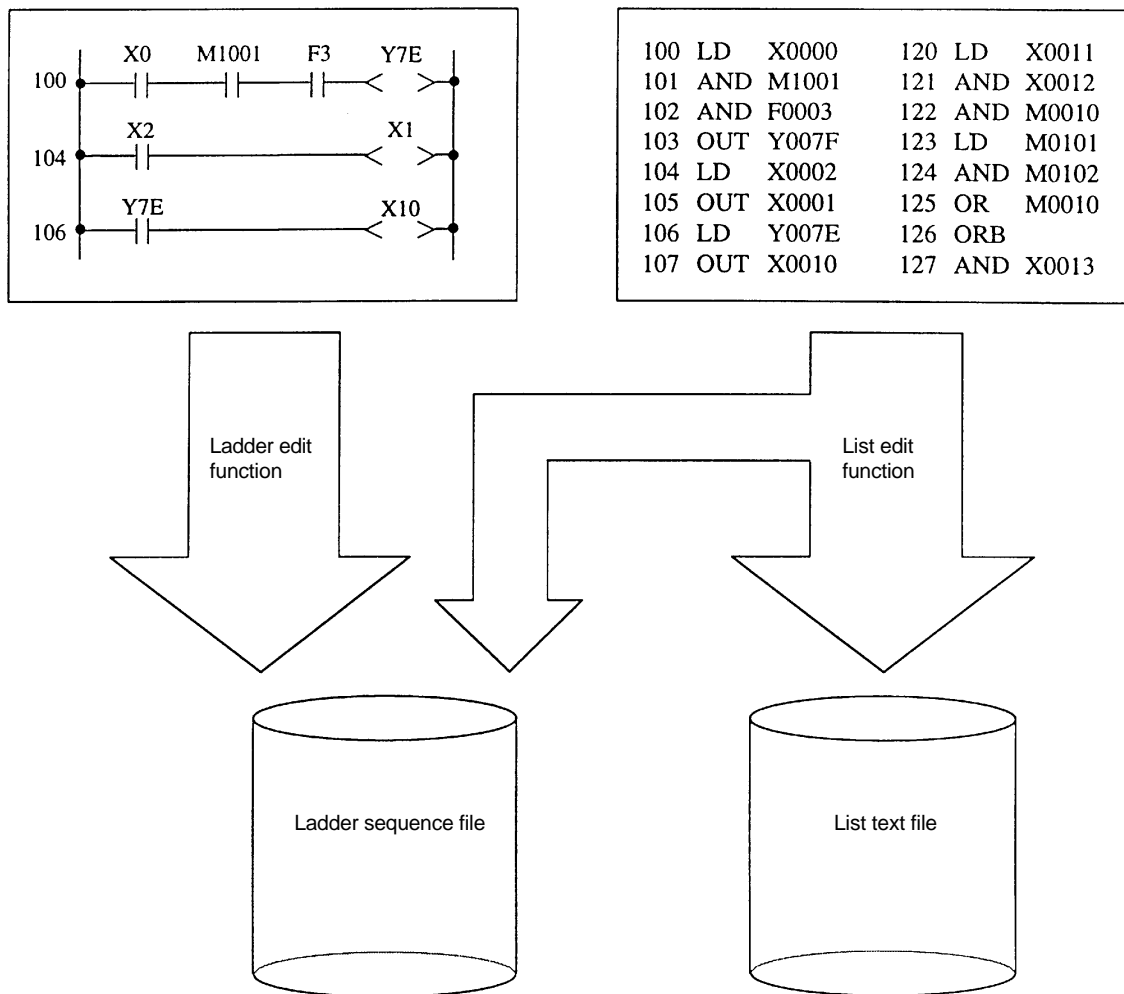| No.	Message	Measures
1	The designated file does not exist.	Select the file from the file list on the window.
2	It failed in copying file.	Confirm the file name, copy destination directory or file name.
3	Designated file name already exists. Designate the other file name.	Confirm the file name, and make sure that it is not duplicated.
4	Copying, deleting or renaming of files cannot be executed while a program running.	Re-execute after the program has ended.
5	Files cannot be copied while PLC operation.	Re-execute after the PLC has ended.

14. List Mode Function

14.1 Function

The codes can be directly input with characters instead of using the ladder circuit symbols when using the PLC development software. This is called the list mode function.
Programming can be performed more efficiently than using circuit symbols by using the list mode function.
The list mode function also has a function to convert the ladder sequence file into a list text file.

```
        X0    M1001   F3      Y7E
100  ----| |------| |------| |----<   >----

        X2                    X1
104  ----| |------------------<   >----

        Y7E                   X10
106  ----| |------------------<   >----
```

```
100 LD   X0000      120 LD   X0011
101 AND  M1001      121 AND  X0012
102 AND  F0003      122 AND  M0010
103 OUT  Y007F      123 LD   M0101
104 LD   X0002      124 AND  M0102
105 OUT  X0001      125 OR   M0010
106 LD   Y007E      126 ORB
107 OUT  X0010      127 AND  X0013
```

Ladder edit function

List edit function
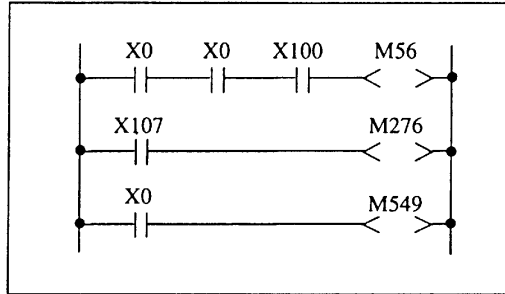
Ladder sequence file

List text file

The list text file can be edited using an editor such as mifes. Changing of the devices, etc., on the NC side can be performed easily with the function to convert the ladder sequence file into a list text file.
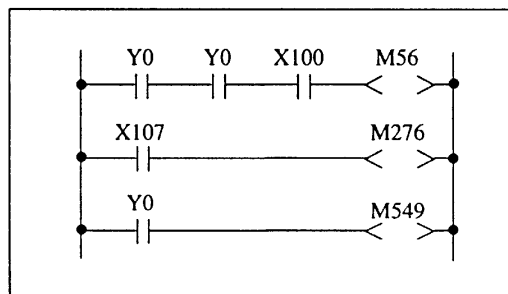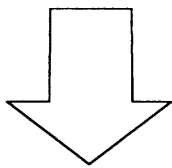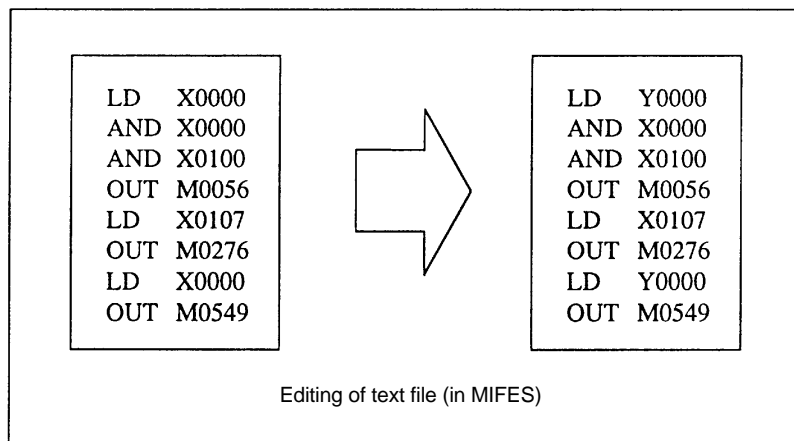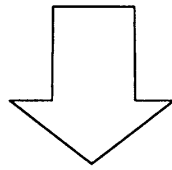
The function to convert the ladder sequence file into a list text file is used as shown below.
To convert LD X0 in the ladder sequence file to LD Y0 (device batch conversion)

Convert the ladder sequence file into a list file as shown on the left.

```
LD    X0000
AND   X0000
AND   X0100
OUT   M0056
LD    X0107
OUT   M0276
LD    X0000
OUT   M0549
```

```
LD    Y0000
AND   X0000
AND   X0100
OUT   M0056
LD    X0107
OUT   M0276
LD    Y0000
OUT   M0549
```

Editing of text file (in MIFES)

Convert the list text file into a ladder sequence file.

14.2 Menu Operation

The following title screen will display when the PLC development software is started up.
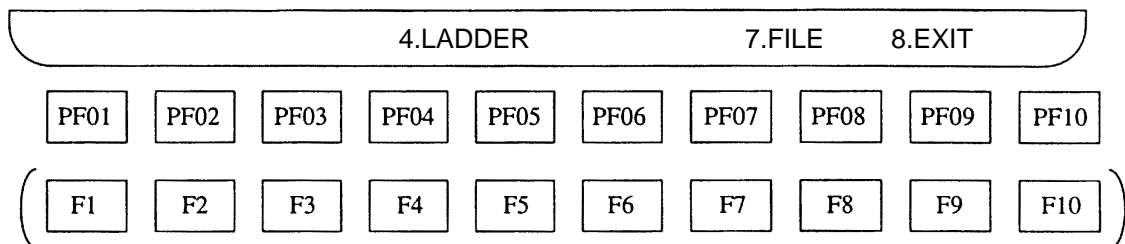
```
┌─────────────────────────────────────────────────────┐
│                                                       │
│                                                       │
│                                                       │
│      PLC  DEVELOPMENT  SOFTWARE                        │
│                                                       │
│                                                       │
│     ┌───────────┐                      ┌─────────┐    │
│     │Model name │         SYSTEM NO.   │ Version │    │
│     └───────────┘                      └─────────┘    │
│     COPYRIGHT  (C)  1992  MITSUBISHI  ELECTRIC  CORPORATION │
│                                                       │
│     ALL  RIGHTS  RESERVED                             │
│                                                       │
│                                                       │
│                                                       │
│                                                       │
│                 4.LADDER        7.FILE    8.EXIT      │
└─────────────────────────────────────────────────────┘
                 └──Menu display section
```
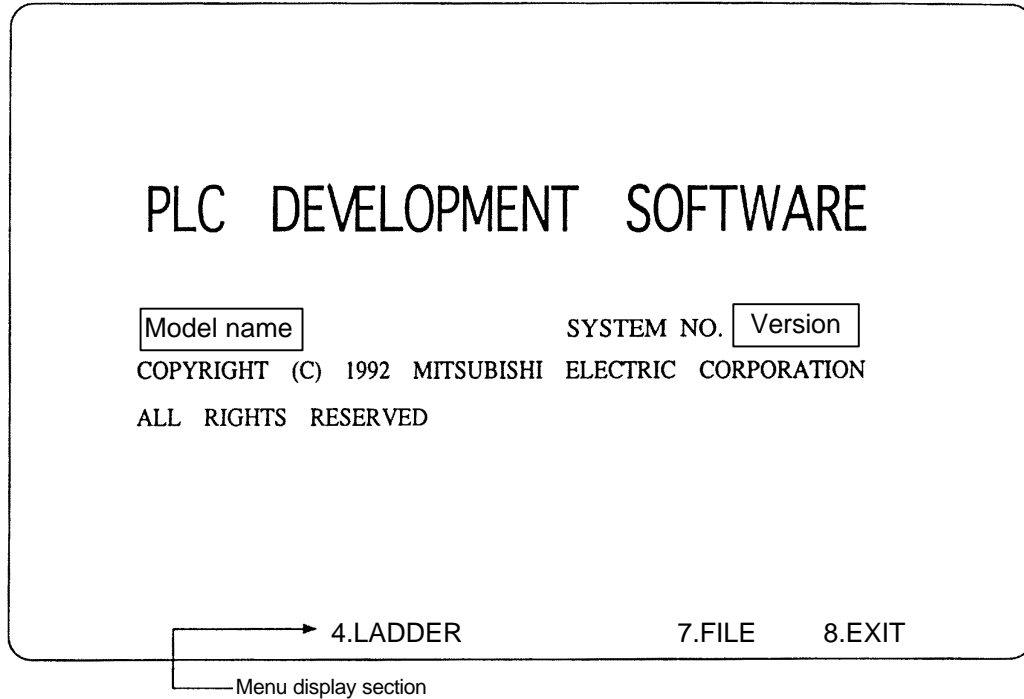
When the title screen is displayed, the menu will appear on the bottom of the screen.

Use the function keys PF01 to PF10 to select a menu. (These keys may be F1 to F10 depending on the personal computer model.)

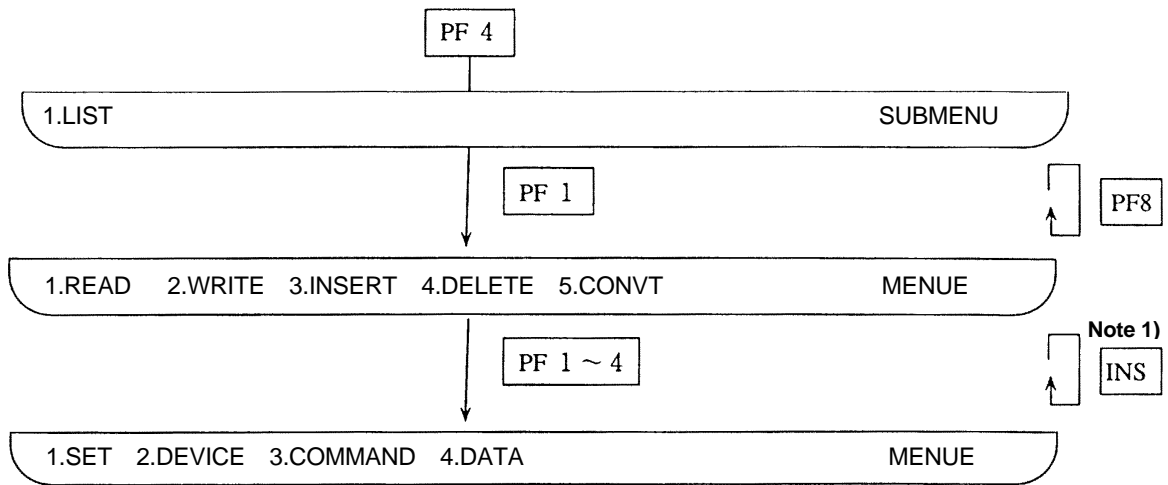The menu number and function key numbers correspond to each other.
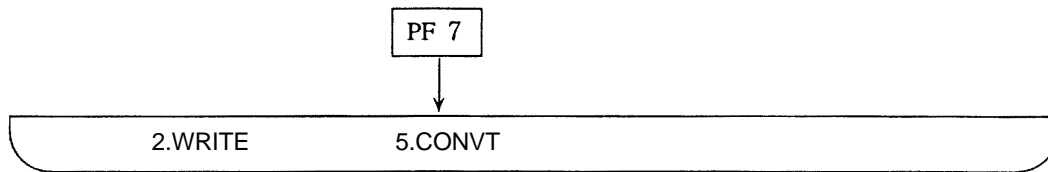
The menu configuration is shown below.

```
            4.LADDER          7.FILE     8.EXIT

┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐
│ PF01 │ │ PF02 │ │ PF03 │ │ PF04 │ │ PF05 │ │ PF06 │ │ PF07 │ │ PF08 │ │ PF09 │ │ PF10 │
└──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘

┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐ ┌──────┐
│  F1  │ │  F2  │ │  F3  │ │  F4  │ │  F5  │ │  F6  │ │  F7  │ │  F8  │ │  F9  │ │ F10  │
└──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘ └──────┘
```

When 4.LADDER is selected:

PF 4

1.LIST   SUBMENU

PF 1   PF8

1.READ   2.WRITE   3.INSERT   4.DELETE   5.CONVT   MENUE

PF 1 ~ 4   **Note 1)** INS

1.SET   2.DEVICE   3.COMMAND   4.DATA   MENUE

(Note 1) ESC = can be used instead of INS.

PF 7

2.WRITE   5.CONVT

The PLC development software title screen will appear when PF10 is pressed.

14.3 Starting Up and Ending the List Mode (LIST4B)

[Start up method]

(1) Input the following startup command to start the list mode function.

A:\> list4b ⏎

(2) The list mode function will start, and the title screen will display.

[Ending method]

(1) Press PF10 on the keyboard, and display the title screen of the PLC development software.

(2) Press PF8 (8 EXIT) to end LIST4B. The prompt (A:\>) will appear on the screen, and the program will return to the normal MS-DOS mode.

14.4 List of List Mode Edit Functions

A list of edit functions used for editing the sequence program using the list mode is shown below.

Function item	
List write	Creation of new list
	Correction of existing list
List read	List read with step number
	List read with device number
	List read with command
List insertion	Insertion in list command units
List delete	Deletion of list in command units
	Deletion of list with range designation

14.5 Registering Edited Files

The following operation is used to register the ladder file name and text file name.

[Basic operation]

$\boxed{\text{7.FILE}}$ → $\boxed{\text{2.WRITE}}$ → (Set the necessary setting number and file name.) → $\boxed{\leftarrow}$

[Operation procedure]

(1) Press function menu $\boxed{\text{7.FILE}}$ and the PLC FILE screen will display.

(2) Press $\boxed{\text{2.WRITE}}$ and the setting section will appear.

(3) Set the ladder file name, message file name 1 and text file name.
   • Set the estimated file size for the ladder file.
   • A maximum of 8 characters can be used for the file name.

   (Example) To register the text file as "MELLIST".

   $\boxed{\text{7.FILE}}$ → $\boxed{\text{2.WRITE}}$ → $\boxed{3}$ → $\boxed{\text{M}}$ $\boxed{\text{E}}$ $\boxed{\text{L}}$ $\boxed{\text{L}}$ $\boxed{\text{I}}$ $\boxed{\text{S}}$ $\boxed{\text{T}}$ → $\boxed{\leftarrow}$

```
                                                    PLC   FILE

   # 1 LADDER        ( MELLADDER  )  (  4096/   0 BYTE)
   # 2 MESSAGE       ( MELMSSG1   )  (  4096)
   # 3 TEXT          ( MELLIST    )  ( 12345)








        #(  ) NAME (      ) SIZE (       KBYTE)

         2WRITE          5CONVT
```

14.6 Converting Text Files and Ladder Files

The following operation is used to convert a text file into a ladder file and vice versa.

[Basic operation]

$\boxed{\text{7.FILE}} \rightarrow \boxed{\text{5.CONVT}} \rightarrow \begin{pmatrix} \boxed{1} \\ \boxed{2} \end{pmatrix} \rightarrow \boxed{\hookleftarrow} \rightarrow \boxed{\text{Y}} \rightarrow \boxed{\hookleftarrow}$
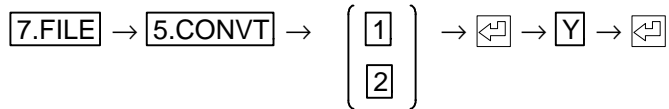
[Operation procedure]

(1) Press function menu $\boxed{\text{7.FILE}}$.

(2) Press $\boxed{\text{5.CONVT}}$ and the text file and ladder file conversion direction will appear.

(3) Designate the conversion direction.

- $\boxed{1}$ $\boxed{\hookleftarrow}$: The ladder file will be converted to a text file.

- $\boxed{2}$ $\boxed{\hookleftarrow}$: The text file will be converted to a ladder file.

The message "COMPLETED" will appear when the conversion is completed.

(Note) After conversion, the converted details will be written into each designated file. If the designated file name (for example, a text file name when 1 is selected above) already exists, the message "V1-FILE IS EXIST. OK (Y/N)" will display.

If $\boxed{\text{Y}}$ $\boxed{\hookleftarrow}$ is pressed here, the conversion process will start.

If $\boxed{\text{N}}$ $\boxed{\hookleftarrow}$ is pressed or if another operation is carried out, the conversion process will be canceled.

(Note) There must be only one "END" command when converting from a text file to a ladder file. An error will occur if there are several "END" commands.

(Example) To convert a ladder file into a text file

$\boxed{\text{7.FILE}} \rightarrow \boxed{\text{5.CONVT}} \rightarrow \boxed{1} \rightarrow \boxed{\hookleftarrow}$

Preset the ladder file name and text file name before starting this operation.
If the text file exists, the message "VI-FILE IS EXIST. OK (Y/N)" will display. Press $\boxed{\text{Y}}$ $\boxed{\hookleftarrow}$ if the file can be replaced.

The message "COMPLETED" will appear when the conversion is completed.

```
                                    VI-FILE IS EXIST. OK (Y/N)

1.LADDER > TEST   2.TEXT>LADDER   (1)

   2.WRITE          5.CONVT
```

⬇

```
                                            COMPLETED

1.LADDER > TEST   2.TEXT>LADDER   (1)

   2.WRITE          5.CONVT
```

14.7 Writing PLC Lists

14.7.1 Creating a new PLC list

This operation is used to create a new PLC list or to erase and existing PLC list.

[Basic operation]

4.LADDER → 1.LIST → 2.WRITE → 1.SET → Y → ⏎→ Writing operation) →
→ 5.CONVT → ⏎

[Operation procedure]

(1) Register the edit file before creating a new circuit.

(2) Press function menu 4.LADDER and the sub menu will display.

(3) Press 1.LIST , 2.WRITE , 1.SET and the following interactive screen will display.

(4) Press Y , ⏎ to create a new circuit or erase an existing circuit. Press 1.READ for other applications.

(5) Only the bus bar on both sides and the step number will display if Y , ⏎ are pressed.

(6) Start the writing operation.

    (a) Create a program from the cursor position using the alphanumeric keys.

    (b) When approximately one screen worth of program is written, press 5.CONVT , ⏎ .

        Note) A maximum of two screens worth of program can be created at once, but 5.CONVT , ⏎ should be pressed after one screen if possible.

        The final object will not be created on the screen if 5.CONVT , ⏎ are not pressed after the list is completed.

    (c) The message "COMPLETED" will appear on the screen when 5.CONVT , ⏎ are pressed and the conversion is completed.

  (Example) The following screen will display when the following are pressed:

4.LADDER → 1.LIST → 2.WRITE → 1.SET

Note) The existing lists will all be deleted when Y , ⏎ are pressed.

```
                                                      PLC LIST


               CLEAR MEMORY FOR NEW PROGRAM?

                  *YES = "Y" "RETURN"
                  *NO  = SELECT READ


         FILE (    TEST )
         MEM (    0/  512 STEP )

         MODE LIST MENU WRITE SET  #(  )(          )

       1SET 2DEVICE 3COMMAND  4DATA            MENUE
```

Perform the writing operation on this screen.

```
                                                      PLC LIST
           0                  15
           1                  16
           2                  17
           3                  18
           4                  19
           5                  20
           6                  21
           7                  22
           8                  23
           9                  24
          10                  25
```
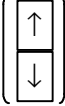
14.7.2 Modification of existing PLC list

An existing PLC list can be modified.

[Basic operation]

(Read out program list) → $\boxed{\begin{array}{c}\boxed{↑}\\\boxed{↓}\end{array}}$ → $\boxed{2.\text{WRITE}}$ → (Input commands) → $\boxed{5.\text{CONVT}}$ → $\boxed{\leftarrow┘}$

**[Operation procedure]**

(1) Read out the PLC list to be modified with the read procedure.

(2) Move the cursor to the PLC list position to be modified. (Use $\boxed{↓}$ $\boxed{↑}$ keys)

(3) Press $\boxed{2.\text{WRITE}}$ and the step number at the cursor will display at the setting section.

Press $\boxed{4.\text{DATA}}$ and the data at the cursor position will display in the setting section.

• Move the cursor at the setting section with the $\boxed{→}$ $\boxed{←}$ keys and modify the command.

(4) The program list can also be changed by inputting the step number and command instead of step (3).

(5) Always press $\boxed{5.\text{CONVT}}$, $\boxed{\leftarrow┘}$ after modifying the PLC list.

• The message "COMPLETED" will appear in the message section after the PLC list is written into the file. The step number will also be updated at this time.

(6) When the data is rewritten, the step numbers will be updated according to the number of steps in the command. The step number on the left will also be updated.

POINT

(1) When the existing circuit where the step number is changed is modified, the step numbers of the existing program and the CJ command jump destination will also be automatically changed.

(Example 1) To change "R0070" to "R0170" at step 5 using the $\boxed{4.\text{DATA}}$ key.

$\boxed{1.\text{LIST}}$ → $\boxed{1.\text{READ}}$ → $\boxed{5}$ → $\boxed{\leftarrow┘}$ → $\boxed{4.\text{DATA}}$ → $\boxed{→}$ ••• $\boxed{←}$ → $\boxed{R}$ $\boxed{0}$ $\boxed{1}$ $\boxed{7}$ $\boxed{0}$ $\boxed{\leftarrow┘}$

→ $\boxed{5.\text{CONVT}}$ → $\boxed{\leftarrow┘}$

```
                                                    PLC LIST
     1   LD    M1000          20  LD    X0004
     2   ORI   M1000          21  OUT   Y01C8         →  R0170
     3   OUT   M1000          22  OUT   Y01C9
     4   LD    M1000          23  OUT   Y01CB
     5   MOV   R0070 K4M1008  24  OUT   Y01D0
     8   LD    M1000          25  OUT   Y01D1
     9   MOV   R2934 K4M1024  26  OUT   Y01D2
    12   LD    X0000          27  OUT   Y01D3
    13   OUT   Y0218          28  LD    X0004
    14   LD    X0001          29  OR    X0006
    15   OUT   Y0219          30  OUT   Y01CA
    16   LD    X0002          31  LD    X0005
    17   OUT   Y021A          32  OUT   Y023F
    18   LD    X0003          33  LD    X0007
    19   OUT   Y021D          34  OUT   Y029A

 FILE  (       TEST)
 MEM (        0/    512 STEP )
 MODE   LIST  MENU  WRITE   SET  # (    ) (                    )

 1.SET   2.DEVICE   3.COMMAND   4.DATA              MENUE
```

(Example 2) To rewrite to after step 20 using the ⎡2. WRITE⎤ key.

⎡1.LIST⎤ → ⎡1.READ⎤ → ⎡2⎤ → ⎡0⎤ → ⎡⏎⎤→ ⎡2.WRITE⎤ → ⎡L⎤ ⎡D⎤ ⎡SP⎤ ⎡M⎤ ⎡1⎤ ⎡0⎤ → ⎡⏎⎤

⎡5.CONVT⎤ → ⎡⏎⎤

```
                                                    PLC LIST
        1   LD      M1000           20   LD    X0004
        2   ORI     M1000           21   OUT   Y01C8
        3   OUT     M1000           22   OUT   Y01C9
        4   LD      M1000           23   OUT   Y01CB              LD   M10
        5   MOV     R0070 K4M1008   24   OUT   Y01D0
        8   LD      M1000           25   OUT   Y01D1
        9   MOV     R2934 K4M1024   26   OUT   Y01D2
       12   LD      X0000           27   OUT   Y01D3
       13   OUT     Y0218           28   LD    X0004
       14   LD      X0001           29   OR    X0006
       15   OUT     Y0219           30   OUT   Y01CA
       16   LD      X0002           31   LD    X0005
       17   OUT     Y021A           32   OUT   Y023F
       18   LD      X0003           33   LD    X0007
       19   OUT     Y021D           34   OUT   Y029A


   FILE  (      TEST)
   MEM (        0/    512 STEP )
   MODE   LIST   MENU   WRITE   SET   # (   ) (                    )

   1.SET    2.DEVICE    3.COMMAND    4.DATA              MENUE
```

14.8 Reading the PLC List

14.8.1 Reading the PLC list with step number

This operation is used to read the PLC list by designating a step number.

[Basic operation]

$\boxed{\text{1.LIST}} \to \boxed{\text{1.READ}} \to \boxed{\text{1.SET}} \to \boxed{\text{STEP NO.}} \to \boxed{\hookleftarrow}$

[Operation procedure]

(1) Press $\boxed{\text{1.LIST}}$, $\boxed{\text{1.READ}}$, $\boxed{\text{1.SET}}$, $\boxed{\text{STEP NO.}}$, $\boxed{\hookleftarrow}$ and one screen worth of data following the designated number will display.
   - The PLC list will be displayed from the start of that command even if in the middle of the designated step number command.

(2) Press $\boxed{+}$ $\boxed{\hookleftarrow}$ and the list on the screen following the current PLC list screen will display.

   Press $\boxed{-}$ $\boxed{\hookleftarrow}$ and the list on the screen before the current PLC list screen will display.

   The succeeding or preceding screens can be displayed by pressing $\boxed{\hookleftarrow}$ continuously.

(3) The cursor can be moved down one line by pressing $\boxed{\downarrow}$. If pressed continuously, the cursor will continue to move down. When pressed on the last line, the PLC list will scroll over to the right. The cursor can be moved up one line by pressing $\boxed{\uparrow}$. If pressed continuously, the cursor will continue to move up, and will stop at the top line of the left column.

(Example 1) To read out the step number 300 list

$\boxed{\text{1.LIST}} \to \boxed{\text{1.READ}} \to \boxed{3}\boxed{0}\boxed{0} \to \boxed{\hookleftarrow}$

The screen on the right will appear when the following are pressed:

```
                                                    PLC
296  OUT  Y002A        316  LD   T0002
297  LD   X0223        317  OUT  Y002E
298  OUT  Y002B        318  LD   X01F8
299  LD   M1000        319  OUT  Y002F
300 ■ MOV  K5 D0500    320  LD   X01F7
303  LD   X0230        321  OUT  Y0040
304  OUT  T0000 D0500  322  LD   X0200
306  LD   T0000 D0500  323  OUT  Y0041
307  OUT  Y002C        324  LD   X0201
308  LD   X0234        325  OUT  Y0042
309  OUT  T0001 D0500  326  LD   X0202
311  LD   T0001        327  OUT  T0043
312  OUT  Y002D        328  LD   X0203
313  LD   X0238        329  OUT  Y0044
314  OUT  T0002 D0500  330  LD   X0208
FILE  (       TEST)
MEM (    400/    512 STEP )
MODE   LIST  MENU  READ  SET  # (   ) (            )
1.SET    2.DEVICE    3.COMMAND    4.DATA        MENUE
```

Press $\boxed{+}$ $\boxed{\hookleftarrow}$ and the list on the last screen will appear.

```
                                                    PLC
326 ■ LD   X0202       341  OUT  Y004C
327  OUT  T0043        342  LD   X0181
328  LD   X0203        343  OUT  Y004D
329  OUT  Y0044        344  LD   X0182
330  LD   X0208        345  OUT  Y004E
331  OUT  Y0045        346  LD   X0183
332  LD   X0188        347  OUT  Y004F
333  OUT  Y0048        348  LD   X0205
334  LD   X0189        349  OUT  Y0050
335  OUT  Y0049        350  LD   X0206
336  LD   X018A        351  OUT  Y0051
337  OUT  Y004A        352  LD   X020A
338  LD   X018B        353  OUT  Y0052
339  OUT  Y004B        354  LD   X0209
340  LD   X0180        355  OUT  Y0053
FILE  (       TEST)
MEM (    400/    512 STEP )
MODE   LIST  MENU  READ  SET  # (   ) (            )
1.SET    2.DEVICE    3.COMMAND    4.DATA        MENUE
```

14.8.2 Reading the PLC list with device number

This operation is used to read the PLC list by designating a device number.

[Basic operation]

$\boxed{1.\text{LIST}}$ → $\boxed{1.\text{READ}}$ → $\boxed{2.\text{DEVICE}}$ → $\boxed{\text{DEVICE NO.}}$ → $\boxed{\leftarrow}$

[Operation procedure]

(1) Press $\boxed{1.\text{LIST}}$, $\boxed{1.\text{READ}}$, $\boxed{2.\text{DEVICE}}$, $\boxed{\text{DEVICE NO.}}$, $\boxed{\leftarrow}$ and one screen worth of data from the command with the designated device will appear.
   If there are several commands with the designated device, the screen with the smallest step number will display first.
(2) If reading is attempted after all program lists with the corresponding device number have been displayed, the message "PROG NOT FOUND" will appear.

(Example) To read out "D500"

$\boxed{1.\text{LIST}}$ → $\boxed{1.\text{READ}}$ → $\boxed{2.\text{DEVICE}}$ → $\boxed{D}$ $\boxed{5}$ $\boxed{0}$ $\boxed{0}$ → $\boxed{\leftarrow}$

```
                                                        PLC LIST
    296   OUT   Y002A          316   LD    T0002
    297   LD    X0223          317   OUT   Y002E
    298   OUT   Y002B          318   LD    X01F8
    299   LD    M1000          319   OUT   Y002F
    300 ■ MOV   K5 D0500       320   LD    X01F7
    303   LD    X0230          321   OUT   Y0040
    304   OUT   T0000 D0500    322   LD    X0200
    306   LD    T0000 D0500    323   OUT   Y0041
    307   OUT   Y002C          324   LD    X0201
    308   LD    X0234          325   OUT   Y0042
    309   OUT   T0001 D0500    326   LD    X0202
    311   LD    T0001          327   OUT   T0043
    312   OUT   Y002D          328   LD    X0203
    313   LD    X0238          329   OUT   Y0044
    314   OUT   T0002 D0500    330   LD    X0208

    FILE  (       TEST)
    MEM (     400/    512 STEP )
    MODE   LIST   MENU   READ   SET   # (   ) (              )

    1.SET    2.DEVICE    3.COMMAND    4.DATA         MENUE
```

When the $\boxed{\leftarrow}$ key is pressed again, the next "D500" will be searched for and displayed.

14.8.3 Reading the PLC list with command

This operation is used to read a PLC list containing a designated command.

[Basic operation]

$\boxed{\text{1.LIST}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{3.COMMAND}} \rightarrow \boxed{\text{COMMAND CODE}} \rightarrow \boxed{\hookleftarrow}$

[Operation procedure]

(1) Press $\boxed{\text{1.LIST}}$, $\boxed{\text{1.READ}}$, $\boxed{\text{3.COMMAND}}$, $\boxed{\text{COMMAND CODE}}$, $\boxed{\hookleftarrow}$, the PLC list with the designated command will display from the list with the smallest number.
If $\boxed{\hookleftarrow}$ is pressed again, the PLC list with the next smallest number will display.

(2) If reading is attempted after all commands have been displayed, the message "PROG NOT FOUND" will appear.

(Example) To read out "MOV"

$\boxed{\text{1.LIST}} \rightarrow \boxed{\text{1.READ}} \rightarrow \boxed{\text{3.COMMAND}} \rightarrow \boxed{\text{M}}\boxed{\text{O}}\boxed{\text{V}} \rightarrow \boxed{\hookleftarrow}$

```
296   OUT   Y002A          316   LD    T0002        PLC LIST
297   LD    X0223          317   OUT   Y002E
298   OUT   Y002B          318   LD    X01F8
299   LD    M1000          319   OUT   Y002F
300 ■ MOV   K5 D0500       320   LD    X01F7
303   LD    X0230          321   OUT   Y0040
304   OUT   T0000 D0500    322   LD    X0200
306   LD    T0000 D0500    323   OUT   Y0041
307   OUT   Y002C          324   LD    X0201
308   LD    X0234          325   OUT   Y0042
309   OUT   T0001 D0500    326   LD    X0202
311   LD    T0001          327   OUT   T0043
312   OUT   Y002D          328   LD    X0203
313   LD    X0238          329   OUT   Y0044
314   OUT   T0002 D0500    330   LD    X0208

FILE  (       TEST)
MEM (      400/     512 STEP )
MODE   LIST  MENU   READ   SET   # (    ) (                 )

1.SET    2.DEVICE   3.COMMAND    4.DATA              MENUE
```
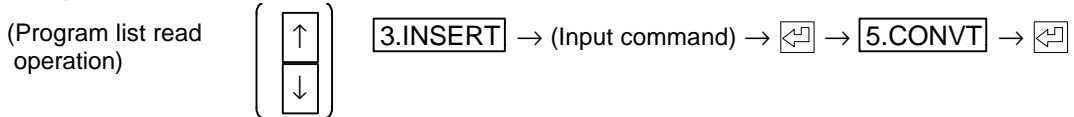
When the $\boxed{\hookleftarrow}$ key is pressed again, the next "MOV" will be searched for and displayed.

14.9 Inserting a PLC List

This operation is used to insert the PLC list in command units.

[Basic operation]

(Program list read operation) 〔↑ / ↓〕 [3.INSERT] → (Input command) → [↵] → [5.CONVT] → [↵]

[Operation procedure]

(1) Display the position to where the PLC list is to be inserted using the circuit read operation.

(2) Move the cursor to the position where the PLC list is to be inserted. (Use [↑] [↓])

(3) Press [3.INSERT] and the step number at the cursor position will display in the setting section.

Press [4.DATA] and the data at the cursor position will display in the setting section.

(4) Write the command to be inserted in the setting section, and press [↵] .

(5) Always press [5.CONVT] [↵] after inputting the command.

- The message "COMPLETED" will appear in the message display section when the PLC list has been written into the file. The PLC list read will display again after being written into the file. The step number will be updated simultaneously.

POINT

(1) When the existing circuit where the step number is changed is modified, the step numbers of the existing program and the CJ command jump destination will also be automatically changed.

(Example) To insert "AND M123" at step 5.

[1.LIST] → [1.READ] → [5] →[↵]→ [3.INSERT] → [A] [N] [D] [SP] [M] [1] [2] [3] →

Reading of step number 5

→ [↵] → [5.CONVT] → [↵]

```
   1  LD    M1000          20  LD    X0004        PLC LIST
   2  ORI   M1000          21  OUT   Y01C8
   3  OUT   M1000          22  OUT   Y01C9
   4  LD    M1000          23  OUT   Y01CB
   5 ■ MOV   R0070 K4M1008  24  OUT   Y01D0
   8  LD    M1000          25  OUT   Y01D1
   9  MOV   R2934 K4M1024  26  OUT   Y01D2
  12  LD    X0000          27  OUT   Y01D3
  13  OUT   Y0218          28  LD    X0004
  14  LD    X0001          29  OR    X0006
  15  OUT   Y0219          30  OUT   Y01CA
  16  LD    X0002          31  LD    X0005
  17  OUT   Y021A          32  OUT   Y023F
  18  LD    X0003          33  LD    X0007
  19  OUT   Y021D          34  OUT   Y029A

FILE  (       TEST)
MEM (     400/    512 STEP )
MODE   LIST  MENU  INSERT       # (  5) (      AND   M123   )

1.SET    2.DEVICE   3.COMMAND   4.DATA              MENUE
```
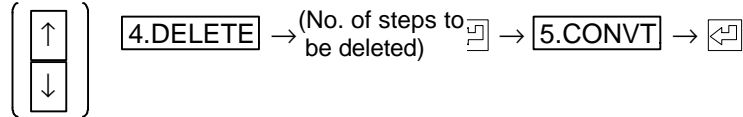
14.10 Deleting a PLC List

This operation is used to delete the PLC list in command units or in a batch.

[Basic operation]

(Program list read operation)
$$\begin{pmatrix} \boxed{↑} \\ \boxed{↓} \end{pmatrix}$$
$\boxed{\text{4.DELETE}}$ → (No. of steps to be deleted) $\boxed{⏎}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{⏎}$

[Operation procedure]

(1) Display the PLC list to be deleted with the read operation.

(2) Move the cursor to the PLC list position to be deleted. (Use $\boxed{↑}$ $\boxed{↓}$)

(3) Press $\boxed{\text{4.DELETE}}$ and the step number at the cursor position will display in the setting section.

To delete several commands in a batch, set the last step to be deleted in the brackets on the right of the setting section.

(4) Press $\boxed{⏎}$ and the step number displayed in the setting section will be deleted.

(5) Always press $\boxed{\text{5.CONVT}}$ $\boxed{⏎}$ after inputting the command.

- The message "COMPLETED" will appear in the message display section when the PLC list has been written into the file. The read PLC list will display again after being written into the file. The step number will be updated simultaneously.

POINT
(1) When part of the PLC list is deleted, the step numbers of the following program and the CJ command jump destination will also be automatically changed.

(Example 1) To delete 1 command.

$\boxed{\text{1.LIST}}$ → $\boxed{\text{1.READ}}$ → $\boxed{\text{1.SET}}$ → $\boxed{5}$ → $\boxed{⏎}$ → $\boxed{\text{4.DELETE}}$ → $\boxed{⏎}$ → $\boxed{\text{5.CONVT}}$ → $\boxed{⏎}$

Reading of step number 5

```
  1  LD   M1000         20  LD   X0004       PLC LIST
  2  ORI  M1000         21  OUT  Y01C8
  3  OUT  M1000         22  OUT  Y01C9
  4  LD   M1000         23  OUT  Y01CB
  5 ■MOV  R0070 K4M1008  24  OUT  Y01D0
  8  LD   M1000         25  OUT  Y01D1
  9  MOV  R2934 K4M1024  26  OUT  Y01D2
 12  LD   X0000         27  OUT  Y01D3
 13  OUT  Y0218         28  LD   X0004
 14  LD   X0001         29  OR   X0006
 15  OUT  Y0219         30  OUT  Y01CA
 16  LD   X0002         31  LD   X0005
 17  OUT  Y021A         32  OUT  Y023F
 18  LD   X0003         33  LD   X0007
 19  OUT  Y021D         34  OUT  Y029A

FILE  (      TEST)
MEM (     400/     512 STEP )
MODE   LIST   MENU   INSERT        # (   5) (              )

1.SET    2.DEVICE    3.COMMAND    4.DATA              MENUE
```

(Example 2) To delete from step 5 to 19 in one batch.

$\boxed{1.\text{LIST}} \rightarrow \boxed{1.\text{READ}} \rightarrow \boxed{1.\text{SET}} \rightarrow \boxed{5}\ \boxed{\leftarrow} \rightarrow \boxed{4.\text{DELETE}} \rightarrow \boxed{1}\ \boxed{9}\ \boxed{\leftarrow}$

$\rightarrow \boxed{5.\text{CONVT}} \rightarrow \boxed{\leftarrow}$

```
                                                            PLC LIST
    1   LD    M1000              20   LD    X0004
    2   ORI   M1000              21   OUT   Y01C8
    3   OUT   M1000              22   OUT   Y01C9
    4   LD    M1000              23   OUT   Y01CB
    5 ■ MOV   R0070 K4M1008      24   OUT   Y01D0
    8   LD    M1000              25   OUT   Y01D1
    9   MOV   R2934 K4M1024      26   OUT   Y01D2
   12   LD    X0000              27   OUT   Y01D3
   13   OUT   Y0218              28   LD    X0004
   14   LD    X0001              29   OR    X0006
   15   OUT   Y0219              30   OUT   Y01CA
   16   LD    X0002              31   LD    X0005
   17   OUT   Y021A              32   OUT   Y023F
   18   LD    X0003              33   LD    X0007
   19   OUT   Y021D              34   OUT   Y029A

FILE  (       TEST)
MEM (     400/     512 STEP )
MODE   LIST   MENU   INSERT          # (   5) (                 19 )

1.SET    2.DEVICE    3.COMMAND    4.DATA                  MENUE
```

15. Messages

15.1 Alarm Messages

No.	Message	Display conditions	Remedies
0	Blank		
1	NO END	The END command is not found in the sequence program.	Display the end of the circuit on the screen, move the cursor to the position displaying the step and input P255.
2	OPERATING ERROR	The operation method was mistaken.	Press the correct key.
3	DEVICE NO. ERROR	An invalid device number was input in the write mode.	Set a valid device number.
4	LADDER ERROR	A circuit that cannot be converted is in the circuit mode.	Revise to the correct circuit.
5	PROGRAM SIZE OVER		
6	NOT WRITE		
7	SETTING ERROR	The key input operation was mistaken while creating a message.	Press the correct key.
8	STEP NO. ERROR	A number of steps exceeding the memory capacity was designated when reading the sequence program.	Change the number of designated steps.
9	PROGRAM SIZE OVER	The parameter and set memory capacity was exceeded during circuit conversion.	Increase the memory capacity.
10	COMMAND CODE ERROR	A code that cannot be converted into a sequence command was found.	Rewrite to a correct command.
11	SETTING ERROR		
12	PROG NOT FOUND	The designated program was not found during the search.	Input the correct program name (file name) into the file designation section.
13	COIL ALREADY USED	The same coil was designated in duplicate when creating the sequence program.	The same coil can be programmed (used), but check that the same coils have not been programmed by mistake.
14	CIRCUIT CONTINUATIVE SIZE OVER	The number of returns in the circuit exceeded eight returns.	Change the circuit so that there are less than eight or less returns.
15	SETTING ERROR	The data input in the message screen is incorrect.	Input the data again.
16	CIRCUIT CONTINUATION ERROR	Two returns return two or more times in the circuit.	Change the ladder circuit so that the two returns return only once.
17	DEVICE ERROR	An invalid value was input for the device.	Input a value within the range again.
18	NO COMMENT DATA	Display of the comment was attempted with no existing contact or coil comment.	Create a contact or coil comment.
19	FILE TYPE INCORRECT	The ladder file and message file were mistaken and registered.	Register as a correct file.

15.2 Messages

No.	Message	Display conditions	Remarks
0			
1	DELETE 1-CIRCUIT	Displayed when one ladder circuit is deleted.	
2	LADDER END	Displayed when the last program was read during the circuit display.	
3	DISPLAY OVER FLOW	Displayed when the display of a circuit covering two screens was attempted.	
4	DELETE 1-CIRCUIT	Displayed when the circuit block not having a step number before circuit conversion is deleted.	
5	COMPLETED	Displayed when the conversion is completed.	
6	DISPLAYING COMMENTS	Displayed while a coil comment or contact comment is displayed.	
7	PRESS <CNV>	Displayed when a ladder circuit is edited and reading is attempted without converting the file.	
8	SELECT INIT!	Displayed when the message has not been initialized.	
9	EXECUTION	Displayed when conversion or initialization is being performed.	
10	SELECT MODE	Displayed if a key is pressed when a mode is not selected.	
11	SELECT FUNCTION	Displayed when a function must be selected in the next operation.	
12	SELECT FILE!	Displayed when a ladder or message file name is not registered.	

Revision History

Sub-No.	Date of revision	Revision details
∗	April 1998	First edition created.