MITSUBISHI

PROGRAMMABLE CONTROLLER

MELSEC-K

Instruction Manual

Intelligent Communication Module type KD51E

MITSUBISHI ELECTRIC

A) 66016-A

CONTENTS

1. GENERAL DESCRIPTION

1

1. GENERAL DESCRIPTION

1.1 General Description

The intelligent communication unit KD51E (hereinafter referred to as "KD51E") is a multi-purpose communication unit equipped with general-purpose functions such as monitor, data collection, logging and computer link by free format.

FEATURES

1. Monitor of programmable controller operating conditions
   The operating conditions of programmable controller are monitored by the CRT.

2. Collection and analysis of data
   The internal data of programmable controller are utilized for the collection and analysis (four operations, functional operation) of data.

3. Logging of data
   Work results and failure information, which have been collected from the programmable controller, can be printed out. The built-in clock function allows the logging of data at any desired time.

4. Up/down load of sequence program
   Read/write of sequence program can be performed on list diagram basis and ladder circuit diagram basis.

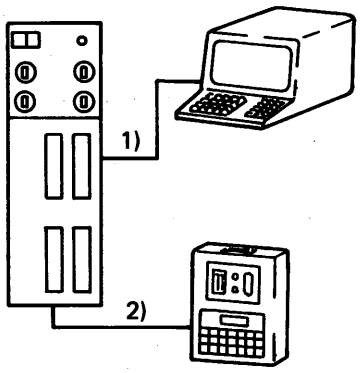
5. Link with computer
   Data can be sent and received in free format. Transmission formats are RS-232-C and RS-422.

6. Programming by GPC-BASIC
   A program, which is used to effect the aforementioned functions 1 ~ 5, is prepared by GPC-BASIC.

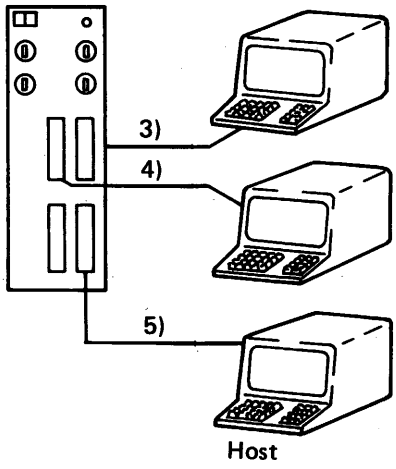
1.2 Functions and Applications of System Examples

Function and Application	Description	Application Example	System Number
Link with computer	Data can be sent and received in free format. Transmission systems are RS-232-C or RS-422.	Collection of programmable controller data by personal computer or host computer, and up/down load of sequence program	1)–2)
Up/down load of sequence program	Up/down load of sequence program is possible in list format or at ladder circuit diagram level.	Line control by computer equipped with CAD system.	1)–2) 3)–4)–5)
Monitor of programmable controller operating conditions	Operating conditions of programmable controller are monitored by display.	Display of production conditions and operating instructions.	3)–4)–5)
Collection and analysis of data	Data are collected from programmable controller and external equipment (such as personal computer and computer) and analyzed (such as four operations and functional operation).	Analysis, display and print-out of failure cause, etc.	3)–4)–5) 6)–7)
Logging of data	Data, such as work results and failure information, collected from programmable controller are printed out. Built-in clock function allows data logging at any desired time.	Preparation of daily work report	3)–4)–5) 6)–7)

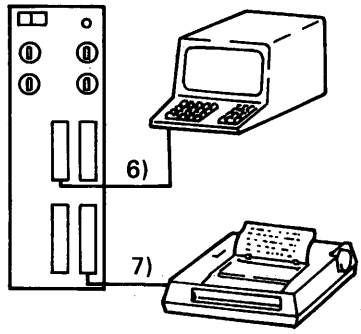
Table 1.1 Functions and Applications of System Examples



System Example 1



System Example 2



System Example 3

MEMO

2. LIST OF EQUIPMENT

2. LIST OF EQUIPMENT

Description		Type Name	Remarks
Intelligent communication unit		KD51E	Main unit (consisting of three substrates) Standard-equipped battery (K6BAT)
Memory	EP-ROM	4KROM	8K bytes, for channels 2 and 3
	IC-RAM	4KRAM	
	Extension memory	K3MB1	40K-byte RAM, for channels 4 and 5
Battery		K6BAT	For IC-RAM and internal clock element
Digital printer		KD51PR	Digital printer with RS-232-C, can be loaded into programmable controller base.
Connector for interface		232-CON	RS-232-C (CH0, CH1, CH2)
		422-CON	Connector for RS-422 (CH3)

Table 2.1 List of Equipment

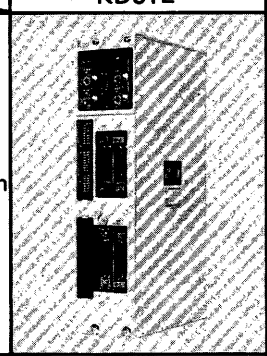
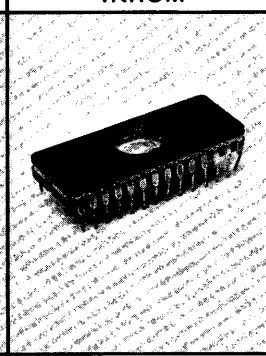
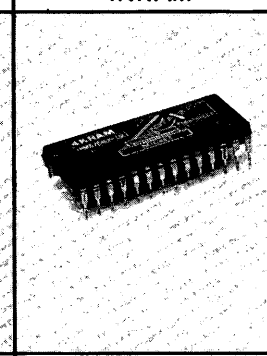
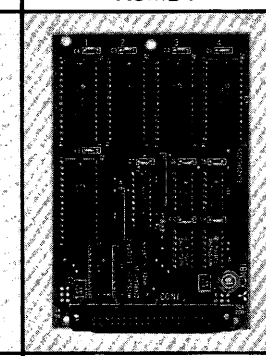
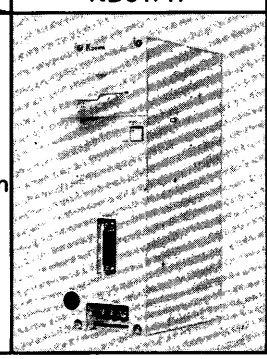
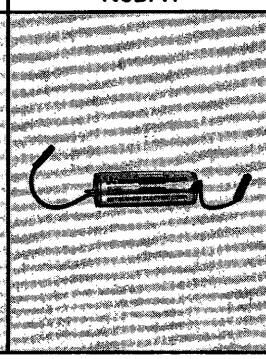
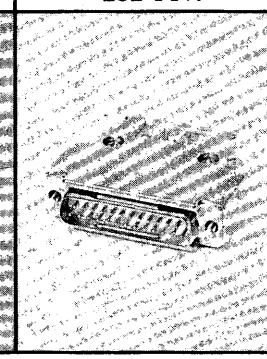
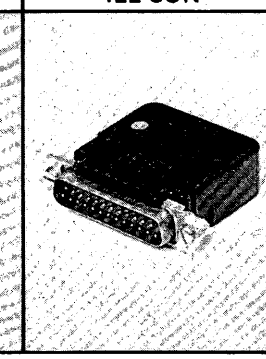
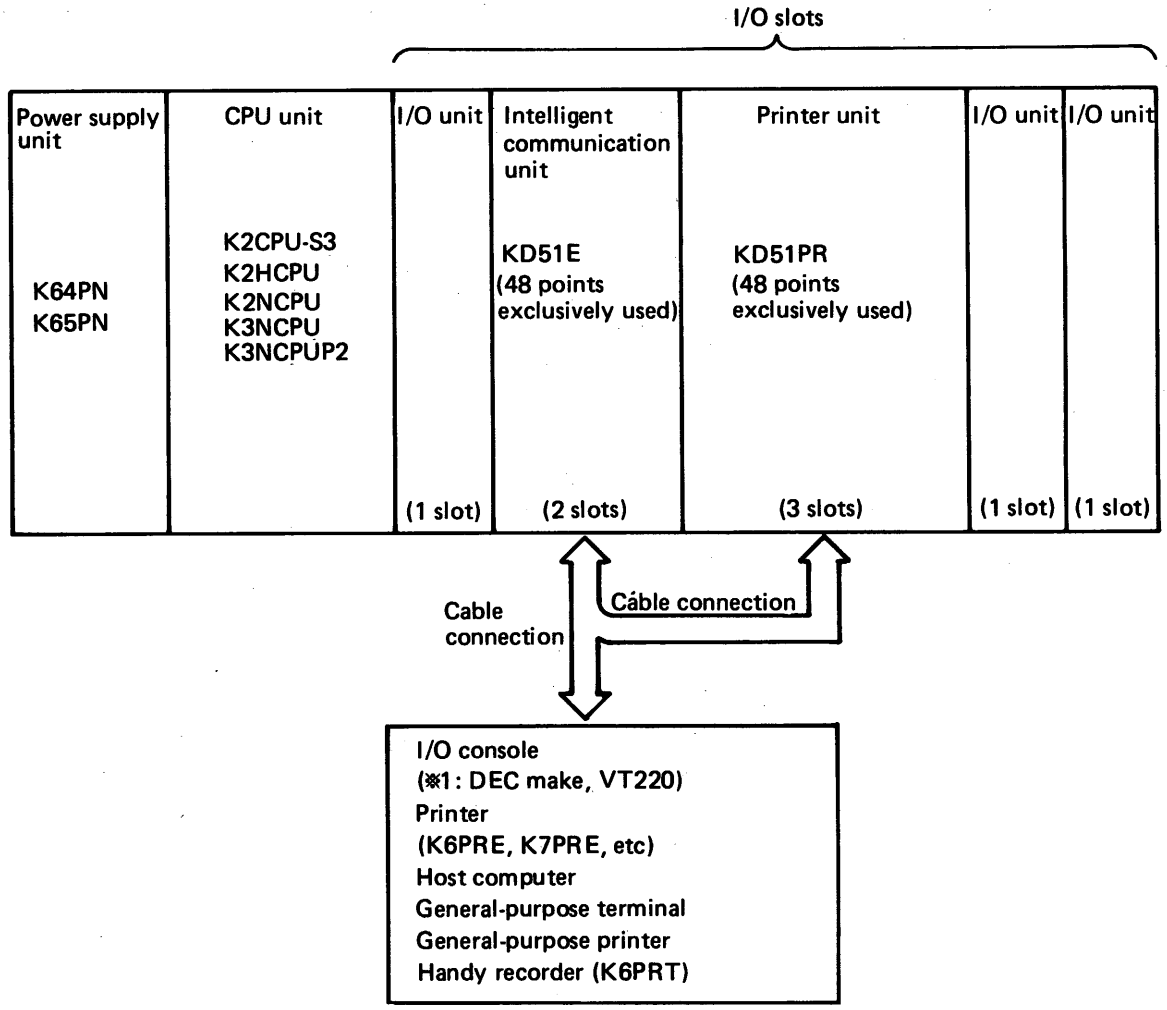
	KD51E	4KROM	4KRAM	K3MB1
Configuration				
	KD51PR	**K6BAT**	**232-CON**	**422-CON**
Configuration				

Table 2.2 List of Equipment Condigurations

3. SYSTEM CONFIGURATION

3. SYSTEM CONFIGURATION

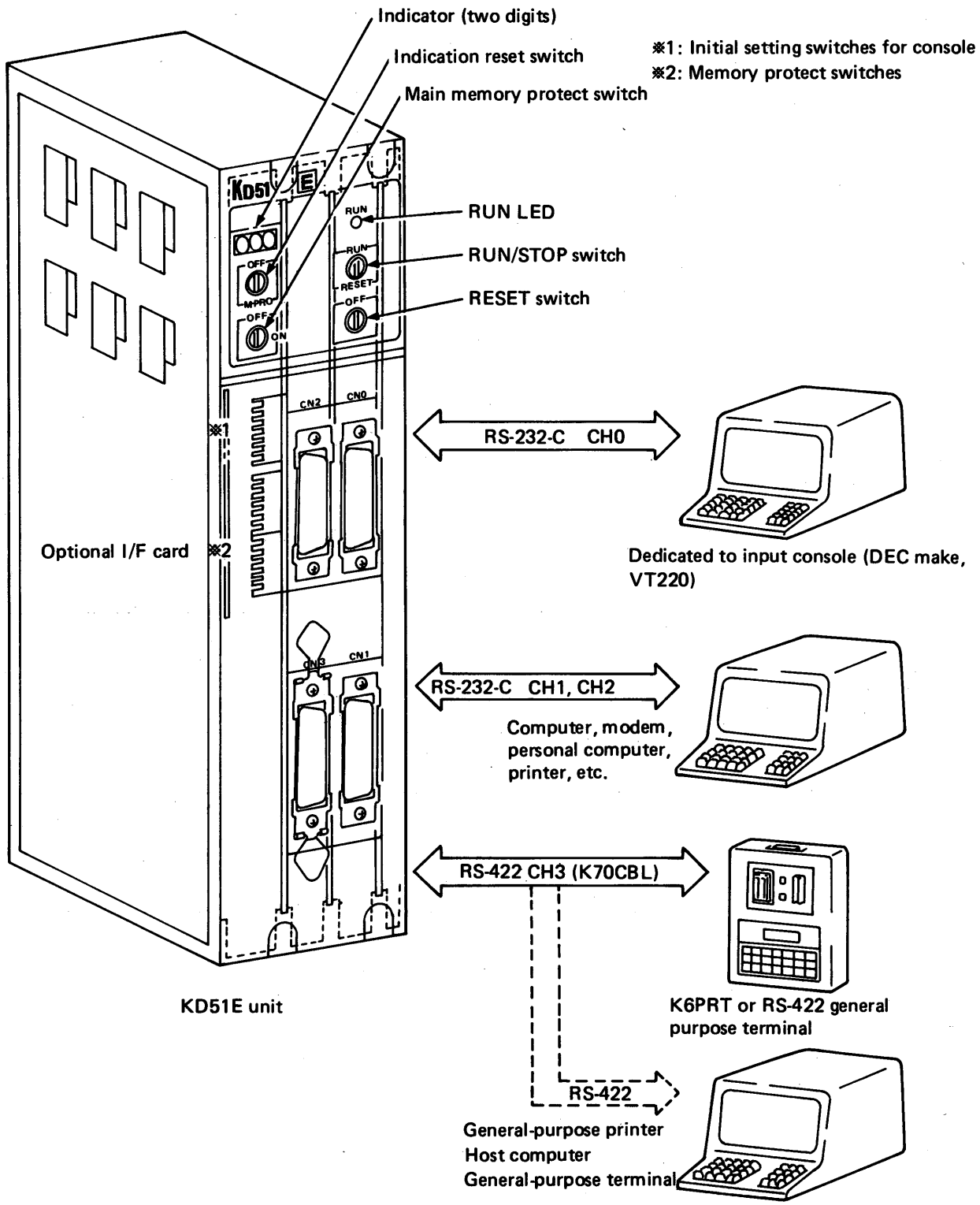
3.1 System Configuration



CAUTION

- The KD51E can be loaded into any desired I/O slots of the basic base or extension base.
- One unit of KD51E can be used for one programmable controller. However, KD51E cannot be used for a remote channel.
- When the failure number F is used, the KD51E cannot be loaded into the slot 0 (located next to the CPU) as in the case of other output units.
- The KD51E cannot be used together with the computer link unit (KJ71L4 or KJ71L7) for one programmable controller.

3.2 Connection to KD51E



MEMO

4. SPECIFICATIONS

4. SPECIFICATIONS

4.1 General Specifications

Item		Specifications
Power supply	Power supply system	Power is supplied from power supply unit of programmable controller (via base)
	Power consumption	5VDC, 3A
Operating ambient temperature		0 ~ 55°C
Storage ambient temperature		–10 ~ 75°C
Operating ambient humidity		10 ~ 90%RH (no dew condensation)
Storage ambient humidity		10 ~ 90%RH (no dew condensation)
Vibration resistance		Conforms to class 3, IIB, JIS C 0911 (16.7 Hz, 3-mm double amplitude, 2 hrs.)
Shock resistance		Conforms to JIS C 0912 (10 g x 3 times in X, Y, and Z directions)
Noise resistance※		1000 Vpp noise voltage, 1 μs noise width, 25 ~ 60 Hz noise frequency by noise simulator
Operating ambience		There should be no corrosive gases and dust should be minimal.
Cooling method		Self-cooling
Mounting screw tightening torque		17 kg·cm (M4 x 0.7 mm screw)

※: In regards to noise resistance, values have been obtained with no unit connected to the RS-232-C interface

4.2 Performance Specifications

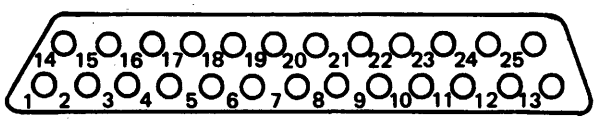
Item	Specifications
CPU element	Z80B (5 MHz)
Programming language	GPC-BASIC
Number of tasks	A maximum of eight tasks
Start condition of task	Start by power-on
	Start by interruption caused by KCPU
	Start by real time interruption (setting is possible in the range of 0.01 ~ 99.99 seconds in units of 0.01 second)
Memory for common work area	2K bytes (IC-RAM) (6000H ~ 67FFH)
Memory for user program (For details, see Section 7.2.1)	Maximum: 104K bytes = 64K bytes + 40K bytes (K3MB1) Content of 64K-byte memory 32K bytes x 2 channels = 64K bytes →In regards to 24K bytes, RAM or ROM can be selected in units of 8K bytes
Memory protect setting range	4F00H ~ 4FFFH (common channel) 8000H ~ DFFFH (channels 2 ~ 4) 8000H ~ 9FFFH (channel 5)
Connectable programmable controller CPU	K2CPU-S3, K2HCPU, K2NCPU, K3NCPU, K3NCPUP2
Number of exclusively used I/O points	48 points (The first 16 points for communication with programmable controller CPU. The latter 32 points for OS.) OS: operation system for internal
Arithmetic logical unit (ALU)	Performs high-speed processing of intrinsic functions (trigonometric function, inverse trigonometric function, logarithm, exponential function, $\sqrt{\ }$, absolute value) of BASIC.
Clock function	Year, month, day, hour, minute →read/write 24-hour mode, leap year automatic compensation
Power failure latch function	Lithium battery for backup of user program memory (RAM), internal RAM memory and clock element program, total back up period of 300 days, battery service life of five years
External dimensions (mm/inch)	300/11.81 (H) x 68/2.68 (W) x 166/6.34 (D)

4.3 Interface Performance Specifications

RS-232-C

I/F Channel	Interface Name	Item	Specifications
CH0 ~ CH2	RS-232-C	Connected unit	Console (only CH0), computer, personal computer, printer, modem, etc. with RS-232C interface (CH1, CH2)
		Transmission system	Conforms to EIA. RS-232-C
		Transmission speed (BPS)	• 300, 600, 1200, 2400, 4800, 9600 selectable • CH0 is set by front switches SW1 ~ SW3. • CH1 and 2 are set via console.
		Synchronous system	Asynchronous mode
		USART mode setting	Baud rate setting (300, 600, 1200, 2400, 4800, 9600, BPS selectable) Parity bit setting: No parity check / Parity check: Even parity / Odd parity Stop bit setting: Stop bit: 1 / Stop bit: 2 Character data bit setting: Data: 7 bits / Data: 8 bits Communication control setting: XON/XOFF control / Control by DR terminal ※: CH0 is set by front DIP switches (SW1 ~ 8)

Connector Specifications

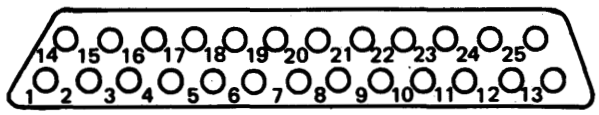


Pin Number	Signal Abbreviation	Signal Direction KD51E ↔ exterior	General Description
1	FG		Frame ground
2	SD	→	Sent data
3	RD	←	Received data
4	RTS	→	Request to send
5	CTS	←	Clear to send
6	DSR	←	Data set ready
7	SG		Signal ground
20	DTR	→	Data terminal ready

RS-422

I/F Channel	Interface Name	Item	Specifications
CH3	RS-422	Connected unit	K6PRT, printer, personal computer, with RS-422, etc.
		Transmission system	Conforms to EIA. RS-422
		Synchronous system	Asynchronous mode
		USART mode	Baud rate setting (300, 600, 1200, 2400, 4800, 9600 BPS selectable) Parity bit setting: No parity check / Parity check: Even parity / Odd parity Stop bit setting: Stop bit: 1 / Stop bit: 2 Character data bit setting: Data: 7 bits / Data: 8 bits Communication control setting: XON/XOFF control / Control by DR terminal ※: When K6PRT is connected, setting is automatic in K6PRT OPERATION mode.

Connector specifications



Signal Name	Block Diagram	Pin	Signal Direction KD51E ↔ exterior
Sent data	(+) (−)	③ ⑯	→
Received data	(+) (−)	② ⑮	←
Data terminal ready	(+) (−)	⑤ ⑱	→
Data set ready		④ ⑰	←
DC power		⑫ ⑬ ㉔ ㉕	Not use
Signal ground		⑦ ⑧ ㉙ ㉑	SG
Frame ground		①	

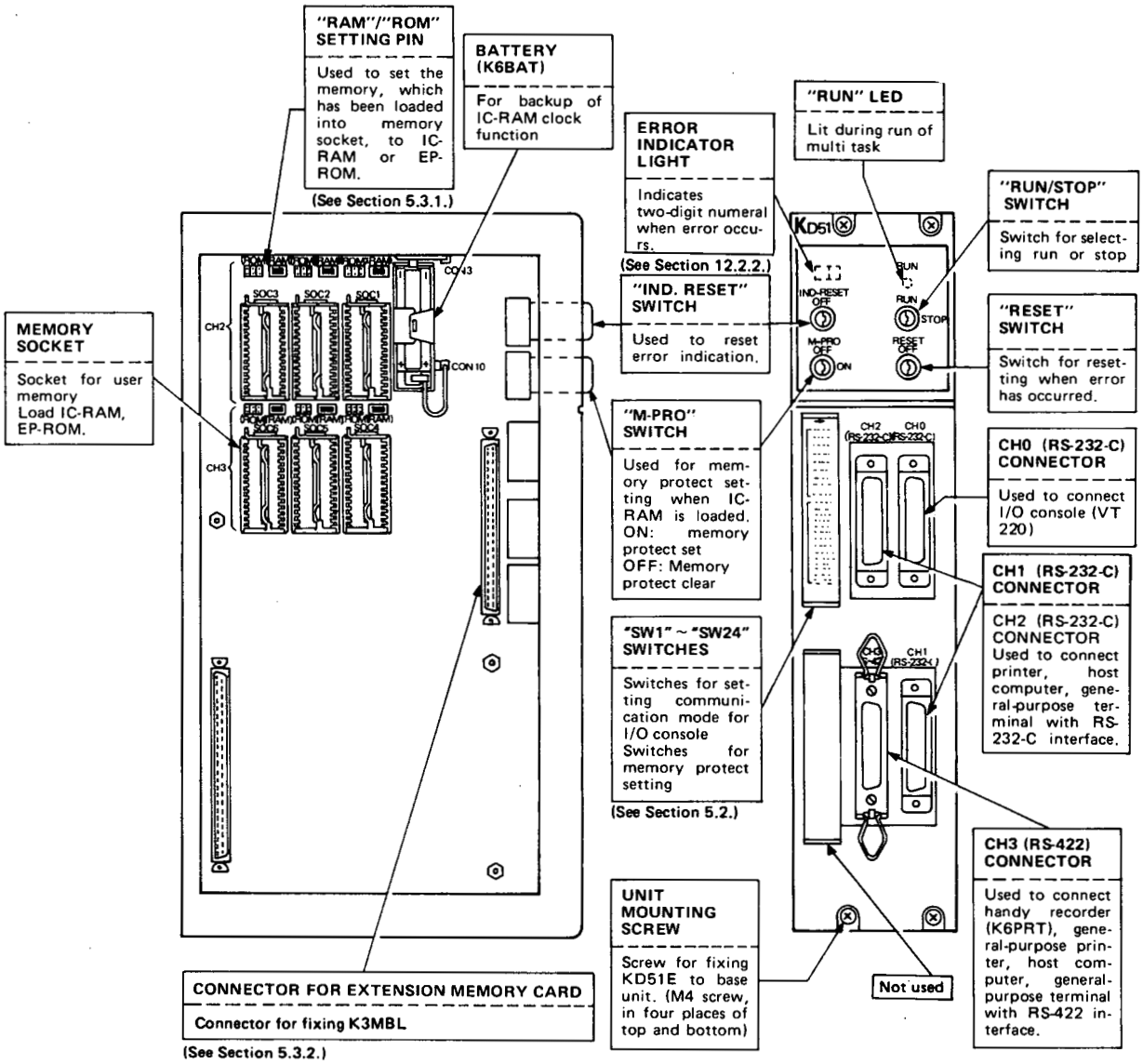
※: Be sure to connect pin 21 to signal ground of connected unit.

MEMO

5. HANDLING

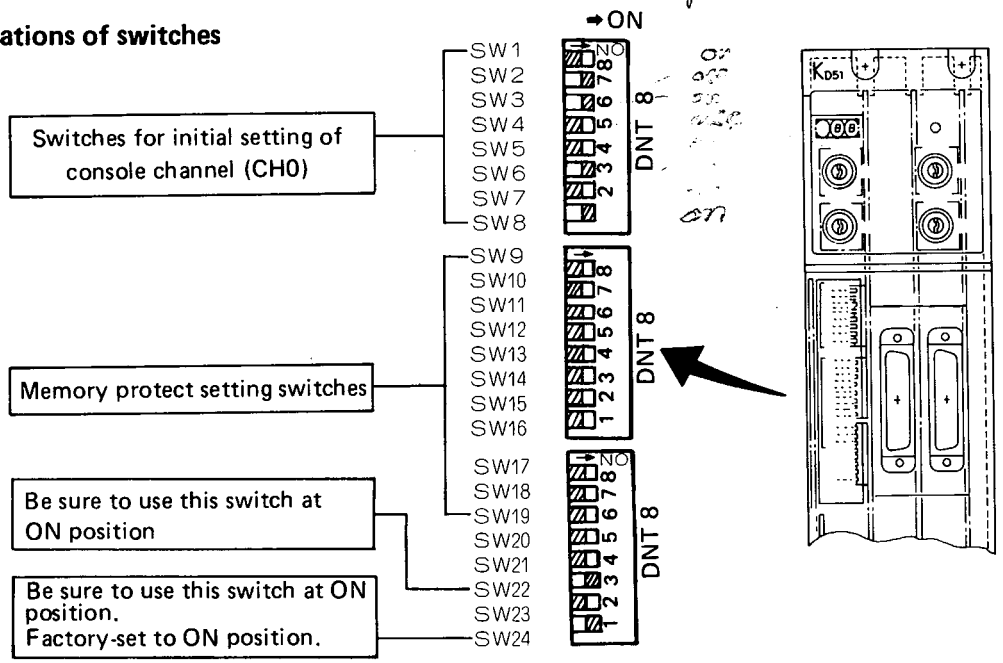
5. HANDLING

5.1 Nomenclature and Explanation



5.2 Front DIP Switch Setting

5.2.1 Applications of switches



5.2.2 Initial setting of console channel (CH0)

Perform programming by use of the console which is connected to the channel 0 (CH0). When programming, initial setting is always required at first. Therefore, perform initial setting according to the following tables.

Baud Rate / Switch Number	Transmission Speed (BPS)					
	300	600	1200	2400	4800	9600
SW1	ON	OFF	ON	OFF	ON	OFF
SW2	OFF	ON	ON	OFF	OFF	ON
SW3	OFF	OFF	OFF	ON	ON	ON

(Recommended communication mode:
9600 BPS
No parity
Stop bit: 1
Data length: 8
8th bit "SPACE"
Conforms to the mode of VT220 at power-on.)

Switch Number	Application	OFF	ON
SW4	Setting of parity check	No	Yes
SW5	Setting of even parity/odd parity	Even	Odd
SW6	Setting of data length	7 bits	8 bits
SW7	Setting of stop bit	1 bit	2 bits
SW8	Setting of communication control	DR terminal control	※XON/XOFF control

Note: When power is turned on, the VT220 is set to XON/OFF control state.

CAUTION

When the power is turned on after the setting of communication mode, communicable state is set.

5.2.3 Setting of memory protect switch

The maximum capacity of memory for user program is 104K bytes (64K bytes + 40K bytes of extension memory). When the RAM is used, the memory protect can be set for 80K bytes in units of 8K bytes. The memory protect inhibits write to the IC-RAM.

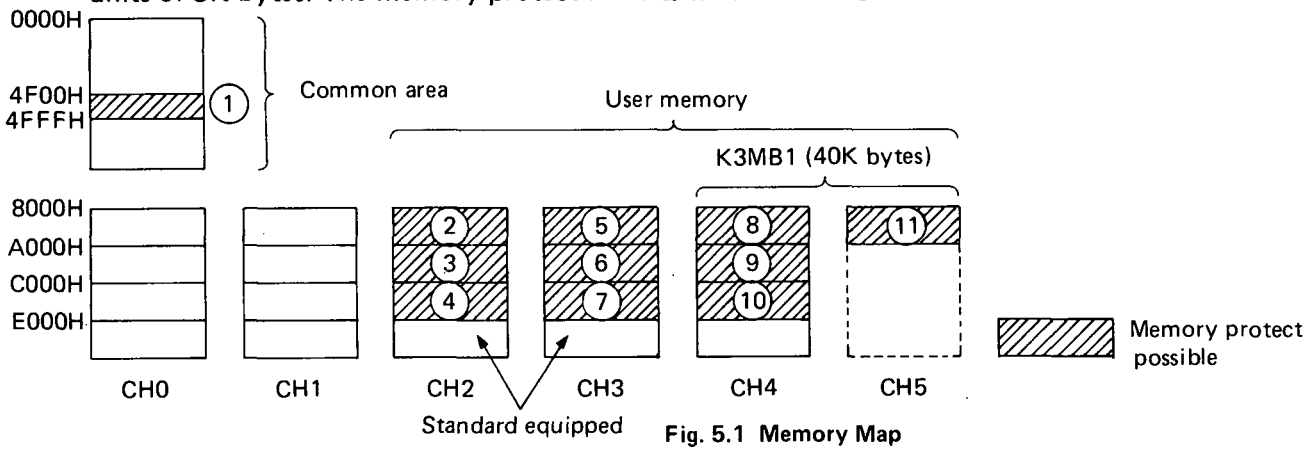


Fig. 5.1 Memory Map

Division	Memory Protect Setting Number	Memory Protect Area with Switch at ON Position	Channel
(1)	SW9	4F00H ~ 4FFFH	Common area
(2)	SW10	8000H ~ 9FFFH	2
(3)	SW11	A000H ~ BFFFH	2
(4)	SW12	C000H ~ DFFFH	2
(5)	SW13	8000H ~ 9FFFH	3
(6)	SW14	A000H ~ BFFFH	3
(7)	SW15	C000H ~ DFFFH	3
(8)	SW16	8000H ~ 9FFFH	4
(9)	SW17	A000H ~ BFFFH	4
(10)	SW18	C000H ~ DFFFH	4
(11)	SW19	8000H ~ 9FFFH	5

Table 5.1 Memory Protect Area

CAUTION

- 4F00H ~ 4FFFH (256 bytes) of SW9 is an area which stores data such as all set data of BASIC program and set data of multi task. During multi task run, SW9 should be in ON position. (See CAUTIONS FOR INITIAL SETTING in page 42.)
- After setting SW9 ~ 19, the memory protect is effected in the areas set by SW9 ~ 19 by moving the memory protect key switch to ON position.
- During preparation and correction of BASIC program, keep the memory protect off.

5.2.4 Applications of front key switchs

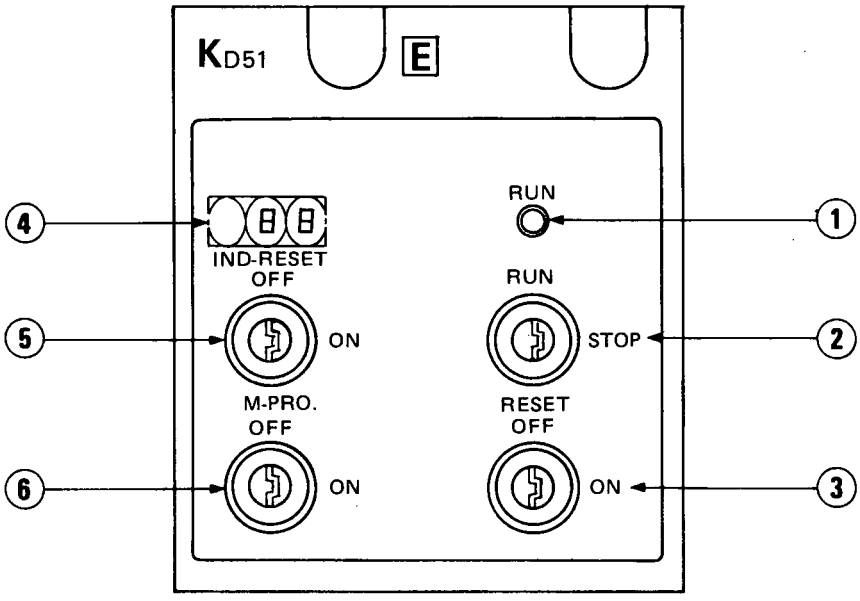


Fig. 5.2

(1) RUN LED

- This LED is lit during run of multi task.

(2) RUN/STOP key switch

- To start multi task, move this key switch to RUN position and operate the input console. (See Section 6.5.)
- During BASIC programming, the RUN command is effective when the RUN/STOP switch is in RUN position.

(3) RESET key switch

- This key switch is used to stop multi task when error has occurred or during run of multi task and re-execute multi task from the initial state.

(4) Error code indicator

- This indicator displays an error code in two digits when error has occurred. For the contents of errors, see Section 12.2.2.

(5) IND. RESET (error code indication reset) key switch

- This key switch is used to reset the indication of error code. When the cause of error still remains, the error code is displayed again after this switch is moved to ON position.
- When a plurality of errors have occurred, the next error code is displayed each time this switch is moved to ON position so that all error codes can be checked.

(6) M-PRO. (memory protect) key switch

- After setting the memory protect of each memory area according to Section 5.2.3, move the memory protect switch to ON position.

5.3 Loading

5.3.1 Loading of memory

Hold the memory with care not to touch the reed area. (For how to hold the memory, see Fig. 5.3.)

↓

Raise the clamping lever on the top side of socket.

↓

Insert the memory so that ▨ portion of memory is positioned in the direction of ▨ portion indicated on the socket.

↓

While holding the central portion of memory, lower the lever of socket.

↓

Check that the memory is not lifted from the socket.

↓

Set the RAM/ROM setting pin depending on the type of loaded memory (IC-RAM or EP-ROM).

↓

Completion

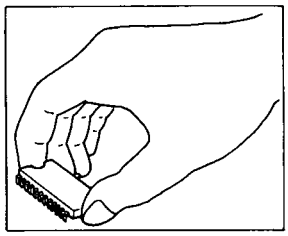


Fig. 5.3 How to Hold Memory

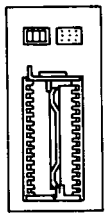


Fig. 5.4 Before Loading Memory

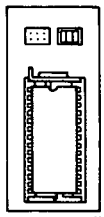


Fig. 5.5 Loading State of IC-RAM

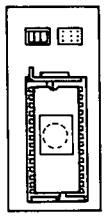


Fig. 5.6 Loading State of EP-ROM

Memory address	C000H~ DFFFH	A000H~ BFFFH	8000H~ 9FFFH
Memory capacity	8K bytes	8K bytes	8K bytes
Channel 2	(ROM) (RAM) SOC3	(ROM) (RAM) SOC2	(ROM) (RAM) SOC1
Channel 3	(ROM) (RAM) SOC6	(ROM) (RAM) SOC5	(ROM) (RAM) SOC4

CAUTION

1. Be sure to load the memory according to the indication on the socket. Snugly fit the memory into the socket. Be careful not to loosely fit the memory.
2. When handling the memory, do not touch its reed area. Also, do not bend the reed area.
3. The memory can be loaded into any desired socket of SOC1 ~ SOC6. However, caution should be exercised because memory addresses change depending on the loaded socket.
4. When the EP-ROM is used for the memory, apply the attached masking tape to the surface of EP-ROM after writing program.
5. When the memory has been unloaded or will be stored, be sure to put it in the case which contained the memory at the time of delivery.
6. Never place the memory on a metal, which leaks or may possibly leak, or on an object which is charged with static electricity, such as woods, plastic, vinyl, fiber, cable and paper.
7. The RAM/ROM select pin is factory-set to RAM position.

5.3.2 Loading of auxiliary memory

When channels 4 and 5 are added, load Type K3MB1 auxiliary memory card (hereinafter referred to as "K3MB1") into the KD51E.

Remove three screws from the K3MB1 loading position of KD51E (two M2.6 x 10 mm screws and one M3 x 6 mm screw).

↓

Insert the connector of K3MB1 into the connector of KD51E.

↓

Fit the removed three screws to KD51E and tighten the screws to fix K3MB1.

↓

Completion

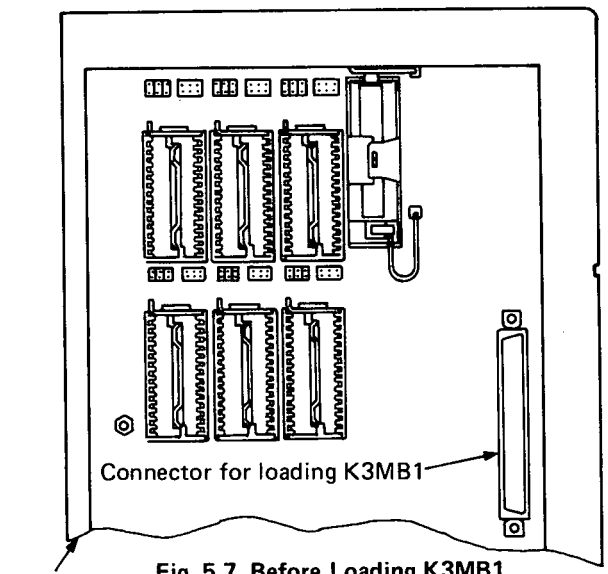


Fig. 5.7 Before Loading K3MB1

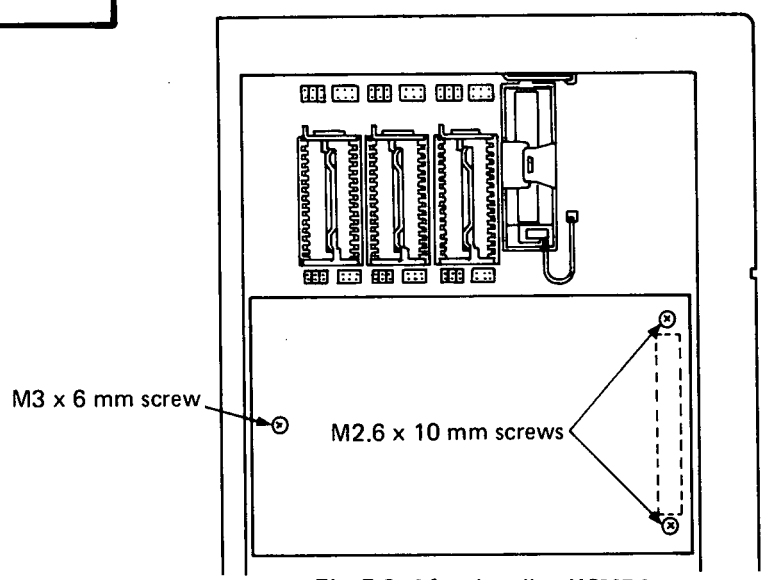


Fig. 5.8 After Loading K3MB1

WARNING

1. Since the K3MB1 is not backed up by a capacitor, the removal of K3MB1 from the KD51E unit will erase the memory contents of K3MB1.
2. When loading or unloading the memory of CH3 after loading the K3MB1, it is required to unload the K3MB1.

5.3.3 Loading of battery

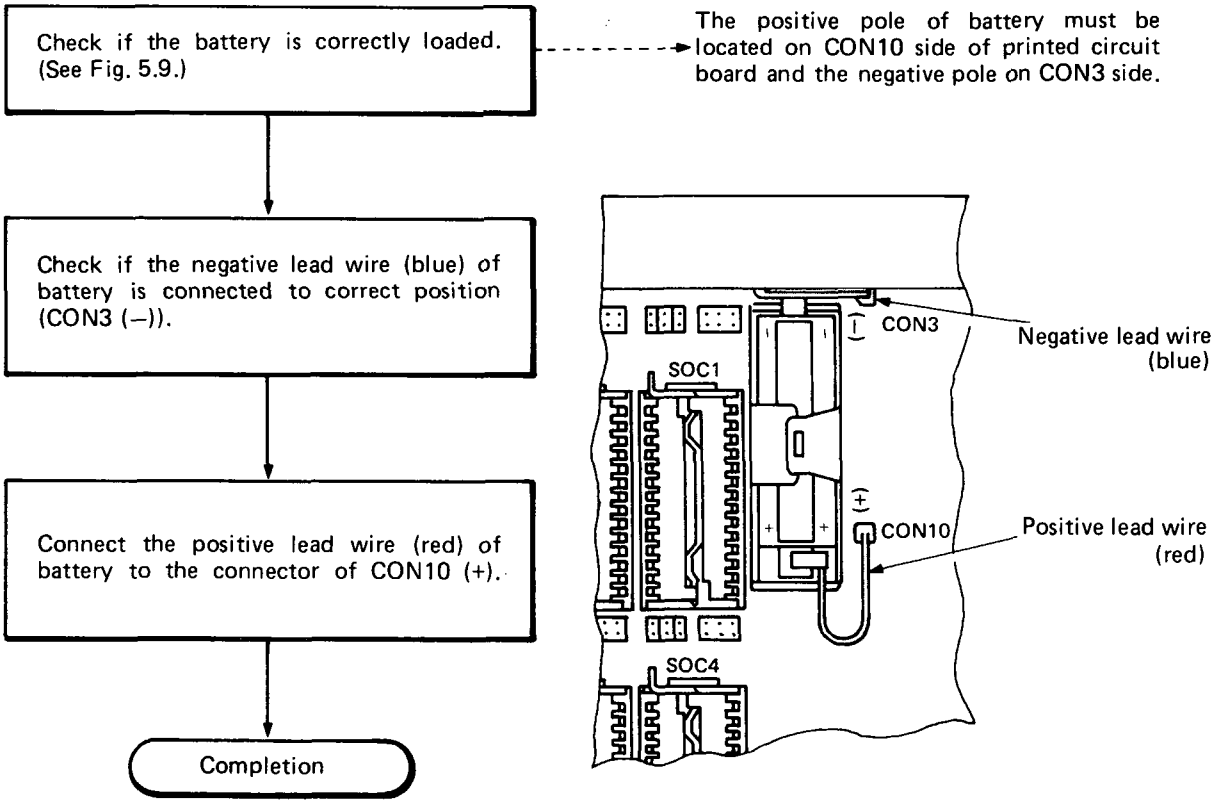


Fig. 5.9 Loading of KD51E Battery

CAUTION

To prevent the battery from being consumed, the lead wires of battery have been disconnected at the factory before shipment. In the following cases, be sure to connect the lead wires of battery to the connectors (CON3, CON10) of printed circuit board:

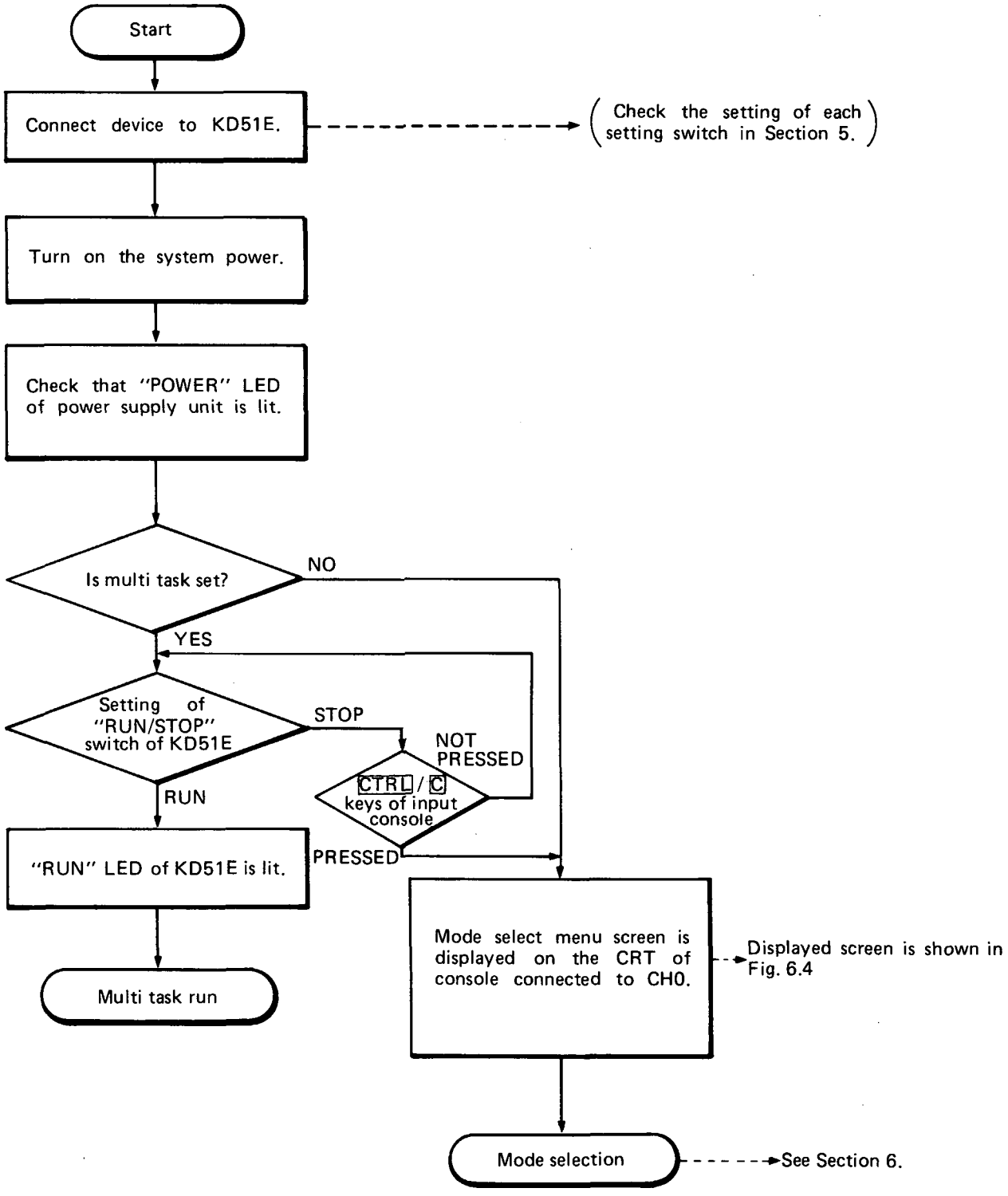
- The battery is required for the backup of RAM. When the EP-ROM is used, however, the battery is also required for the backup of real time clock. Therefore, be sure to wire the battery before use.

6. OPERATING PROCEDURES

6. OPERATING PROCEDURES

6.1 Power-On

This section shows a flow chart from power-on, mode selection to run of multi task.



6.2 Operating Procedure

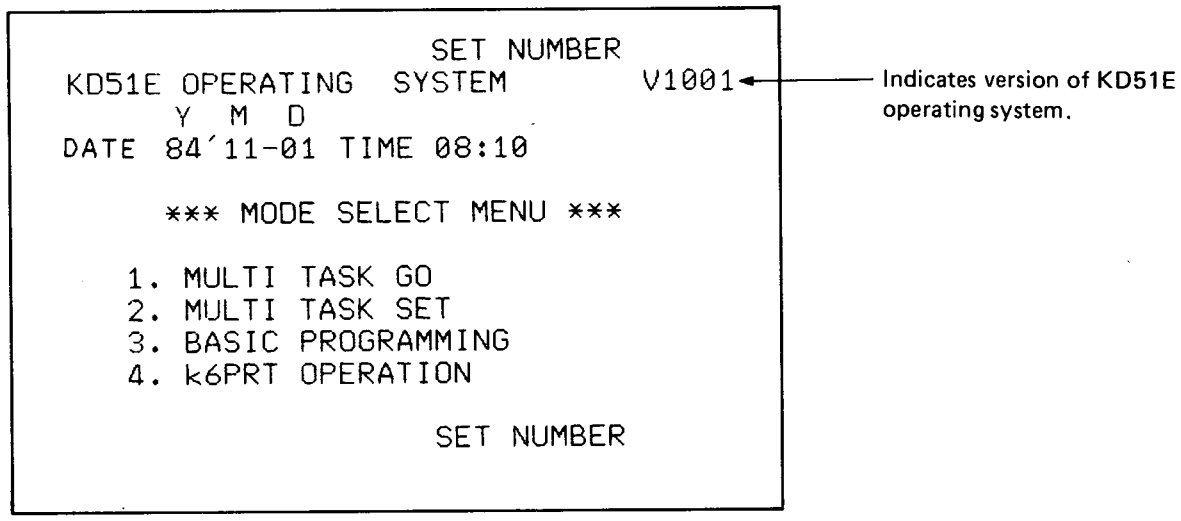


Fig. 6.1 Mode Select Menu Screen

Any one of the four modes shown in Fig. 6.1 can be used as required. Fig. 6.2 shows a typical operating procedure.

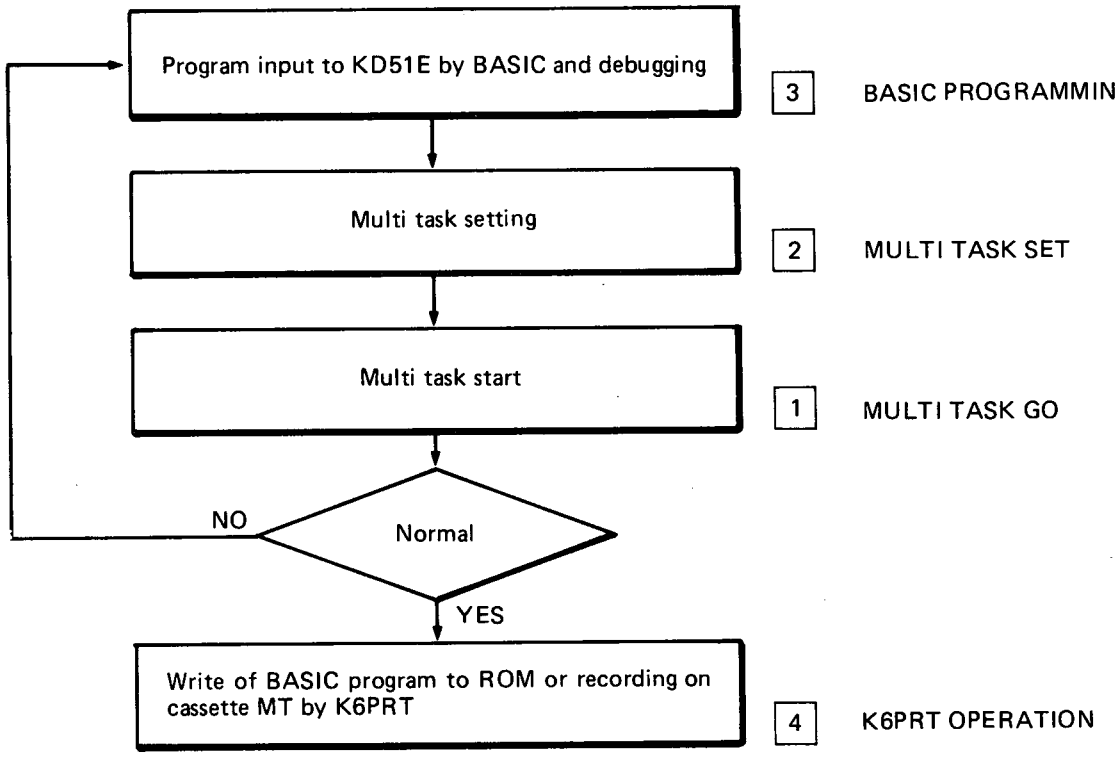


Fig. 6.2

The following pages explain the operating procedure in each mode according to the typical operating procedure shown in Fig. 6.2.

6.3 Setting Procedure of BASIC Programming

```
   KD51E OPERATING  SYSTEM   V1001
     Y  M  D
DATE 84'11-01 TIME 08:10

        *** MODE SELECT MENU ***

      1. MULTI TASK GO
      2. MULTI TASK SET
      3. BASIC PROGRAMMING
      4. k6PRT OPERATION

                        SET NUMBER3
```

Fig. 6.3

3 RETURN ·····BASIC PROGRAMMING

8 4 1 1 0 1 0 8 1 0 RETURN
Year Month Day Hour Minute

- After RETURN key is pressed, the built-in clock function is activated at the preset time.
- When the clock setting is not required, press the RETURN key to proceed to the next screen.
- If non-existing date and/or time is set at the time of clock setting, the "CANNOT SET" error message is displayed. In this case, set the date and time once again.

```
          B A S I C    P R O G R A M M I N G

            NEW:0  CORRECT:1  CONTINUE:2
            COMPLETE:3  ALL DATA DISPLAY:4      1

                          PROGRAMMING(1-8)      2

     1  PROGRAM HEAD ADDRESS                  8000

     2  PROGRAM LAST ADDRESS                  9FFF

     3  ADDITIONAL PROGRAM HEAD ADDRESS       81DE

     4  WORK AREA HEAD ADDRESS                FF00

     5  CHANNEL                                  3

     * REMARK *  PLEASE NOTE THIS DATA
```

BASIC programming data

Fig. 6.4 "BASIC programming" Initial Setting Screen

0 RETURN	1 RETURN	2 RETURN	3 RETURN	4 RETURN
• To prepare a new program	• To correct a program	• To continue program correction or debugging with programming data which are displayed presently.	• To return to the mode select menu screen after completion of BASIC programming or debugging.	• To display all programming data of BASIC program for program numbers 1 ~ 8.

6.4 Setting of BASIC Programming

6.4.1 New programming

NEW

```
        B A S I C    P R O G R A M M I N G
          NEW:0  CORRECT:1  CONTINUE:2
          COMPLETE:3  ALL DATA DISPLAY:4      0

                        PROGRAMMING(1-8)      1

    1  PROGRAM HEAD ADDRESS                 8000

    2  PROGRAM LAST ADDRESS                 9FFF

    3  ADDITIONAL PROGRAM HEAD ADDRESS      8000

    4  WORK AREA HEAD ADDRESS               AF00

    5  CHANNEL                                 3
```

(1) Programming mode setting

0 RETURN

Select	Application
NEW: 0	New programming
CORRECT: 1	Correction
CONTINUE: 2	Continuation
COMPLETE: 3	Completion
ALL DATA DISPLAY: 4	All programming data display

```
        B A S I C    P R O G R A M M I N G
          NEW:0  CORRECT:1  CONTINUE:2
          COMPLETE:3  ALL DATA DISPLAY:4      0

                        PROGRAMMING(1-8)      1

    1  PROGRAM HEAD ADDRESS                 8000

    2  PROGRAM LAST ADDRESS                 9FFF

    3  ADDITIONAL PROGRAM HEAD ADDRESS      8000

    4  WORK AREA HEAD ADDRESS               AF00

    5  CHANNEL                                 3
```

(2) Programming number setting

1 RETURN

- A maximum of eight user programs can be prepared. Each program requires a number (1 ~ 8). In this example, the program number is 1.
- When plural programs are prepared, do not provide them the same number.

```
        B A S I C    P R O G R A M M I N G
          NEW:0  CORRECT:1  CONTINUE:2
          COMPLETE:3  ALL DATA DISPLAY:4      0

                        PROGRAMMING(1-8)      1

    1  PROGRAM HEAD ADDRESS                 8000

    2  PROGRAM LAST ADDRESS                 9FFF

    3  ADDITIONAL PROGRAM HEAD ADDRESS      8000

    4  WORK AREA HEAD ADDRESS               AF00

    5  CHANNEL                                 3
```

(3) Program area setting

8 0 0 0 RETURN

9 F F F RETURN

- Set user program area. First, set the PROGRAM HEAD ADDRESS and then set the PROGRAM LAST ADDRESS. In this example, the user program area is 8000H ~ 9FFFH (8K bytes).
- The ADDITIONAL PROGRAM HEAD ADDRESS is automatically set to the same address as the PROGRAM HEAD ADDRESS. In this example, the ADDITIONAL PROGRAM HEAD ADDRESS is 8000H.

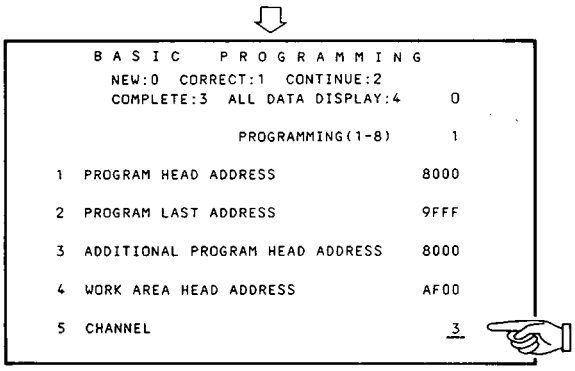
```
        B A S I C    P R O G R A M M I N G
          NEW:0  CORRECT:1  CONTINUE:2
          COMPLETE:3  ALL DATA DISPLAY:4      0

                        PROGRAMMING(1-8)      1

    1  PROGRAM HEAD ADDRESS                 8000

    2  PROGRAM LAST ADDRESS                 9FFF

    3  ADDITIONAL PROGRAM HEAD ADDRESS      8000

    4  WORK AREA HEAD ADDRESS               AF00

    5  CHANNEL                                 3
```

(4) Work area (256 bytes) setting

A F 0 0 RETURN

- One user program always requires a work area of 256 bytes (for interpreter). Set the work area of 256 bytes to the address located behind the PROGRAM LAST ADDRESS which has been set as explained above. At this time, the lower two digits should be "00".

Next page



(5) Channel setting

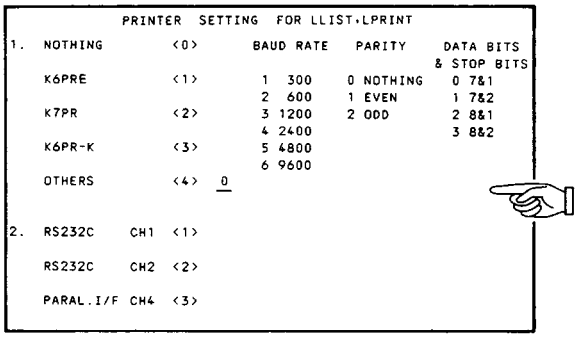
NEW

3 RETURN

- The user memory area has maximum 104K bytes (when the extension memory is loaded) and is divided into channels 2, 3, 4, and 5. (See Section 5.2.3.) Set the channel to be used. In this example, the channel 3 is set.

CAUTION

When it is desired to correct the setting due to setting mistake, press the ESC key before starting the operation of channel setting (5). The operation of programming mode setting can then be performed. Press the RETURN key and move the underline cursor to a position where the setting is desired to be corrected.



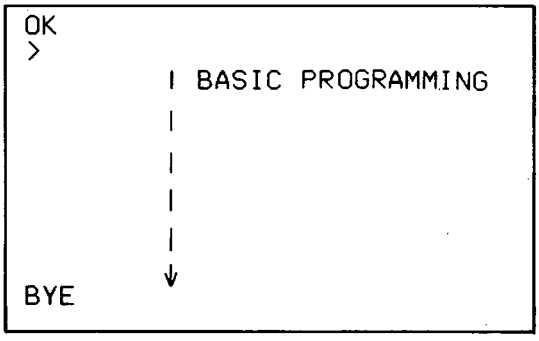
Printer setting

The CRT displays a printer setting screen in Section 6.8.

- In this screen, set a connected printer. The printer may be set after the completion of program. For the setting procedure, see Section 6.8.

RETURN

→Setting displayed on the screen (in this case, "0")

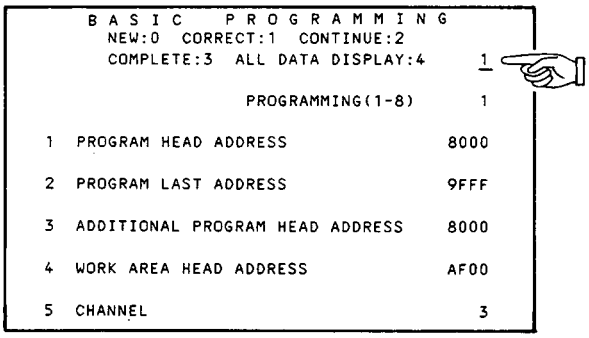


(6) BASIC programming

- Prepare a user program.

B Y E RETURN

- After completion of program preparation, press the keys shown at left.



(7) New programming/programming completion setting

0 RETURN

OR

3 RETURN

- When it is required to prepare the next program, follow Section 6.4.1.
- When the preparation of program has been completed, the CRT returns to the mode select menu screen shown in Fig. 6.1.

6.4.2 Correction of program

CORRECT

```
        B A S I C    P R O G R A M M I N G
          NEW:0  CORRECT:1  CONTINUE:2
          COMPLETE:3  ALL DATA DISPLAY:4      1

                        PROGRAMMING(1-8)      1

    1  PROGRAM HEAD ADDRESS                 8000

    2  PROGRAM LAST ADDRESS                 9FFF

    3  ADDITIONAL PROGRAM HEAD ADDRESS      8000

    4  WORK AREA HEAD ADDRESS               AF00

    5  CHANNEL                                 3
```

(1) Programming mode setting

1 RETURN

RETURN

RETURN

RETURN

RETURN

RETURN

RETURN

- The CORRECT mode is used to correct a prepared program or to set a connected printer.

```
              PRINTER  SETTING  FOR LLIST,LPRINT
1.  NOTHING          <0>      BAUD RATE    PARITY      DATA BITS
                                                      & STOP BITS
    K6PRE            <1>       1  300     0 NOTHING    0 7&1
                               2  600     1 EVEN       1 7&2
    K7PR             <2>       3 1200     2 ODD        2 8&1
                               4 2400                  3 8&2
    K6PR-K           <3>       5 4800
                               6 9600
    OTHERS           <4>  0

2.  RS232C     CH1   <1>

    RS232C     CH2   <2>

    PARAL.I/F  CH4   <3>
```

(2) Printer setting

RETURN

- In this screen, set a connected printer. For the setting procedure, see Section 6.8.

```
OK
> LIST
```

(3) Program correction

L I S T RETURN

- When the L I S T RETURN keys are pressed, the prepared program is displayed beginning with the top 20 lines. When keys other than the CONT/C keys are pressed, the next 20 lines are displayed.

- Read a line to be corrected and correct the program.

6.4.3 Continue mode

CONTINUE

```
      B A S I C    P R O G R A M M I N G
        NEW:0  CORRECT:1  CONTINUE:2
        COMPLETE:3  ALL DATA DISPLAY:4      2

                    PROGRAMMING(1-8)        1

  1  PROGRAM HEAD ADDRESS                  8000

  2  PROGRAM LAST ADDRESS                  9FFF

  3  ADDITIONAL PROGRAM HEAD ADDRESS       8000

  4  WORK AREA HEAD ADDRESS                AF00

  5  CHANNEL                                  3
```

(1) Programming mode setting

2 RETURN

- This mode is used when it is desired to change the programming mode setting screen to the BASIC program screen immediately after the preparation of program. This mode can also be used to correct a program.

```
OK
>
```

(2) Correction of program

6.4.4 All programming data display mode

ALL DATA DISPLAY

```
      B A S I C    P R O G R A M M I N G
        NEW:0  CORRECT:1  CONTINUE:2
        COMPLETE:3  ALL DATA DISPLAY:4      4

                    PROGRAMMING(1-8)        1

  1  PROGRAM HEAD ADDRESS                  8000

  2  PROGRAM LAST ADDRESS                  9FFF

  3  ADDITIONAL PROGRAM HEAD ADDRESS       8000

  4  WORK AREA HEAD ADDRESS                AF00

  5  CHANNEL                                  3
```

(1) Programming mode setting

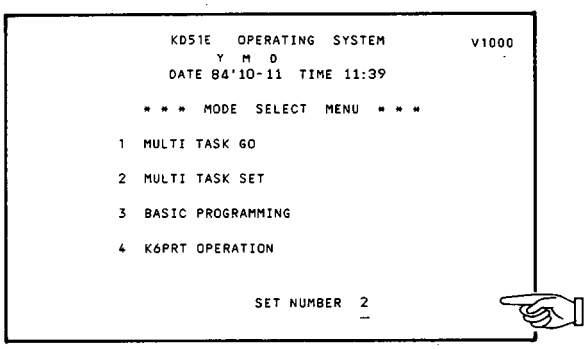
4 RETURN

- This mode is used to check the memory area of prepared program after the preparation of program.

```
                    B A S I C    P R O G R A M M I N G

PROGRAM   PROGRAM       PROGRAM       ADDITIONAL     WORK AREA
NUMBER    HEAD ADDRESS  LAST ADDRESS  HEAD ADDRESS   HEAD ADDRESS   CHANNEL
  1         8000          9FFF          8000           AF00           3
  2         FFFF          FFFF          00FF           0000           0
  3         0000          FF00          FFFF           FFFF           F
  4         00FF          0000          0000           0000           0
  5         8000          9000          0000           00FF           3
  6         0000          0000          FF00           FFFF           F
  7         FFFF          00FF          0000           0000           0
  8         8F00          8FFF          8F00           FFFF           F
```

- Check the memory area of prepared program in the screen shown at left.
- The program areas, which are not set, have no relation to the displayed addresses.
- To return the CRT to the program mode screen, press the ESC key.

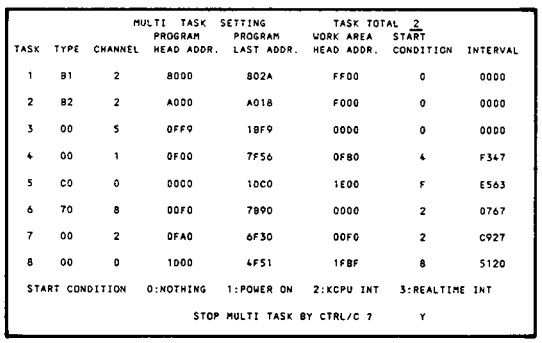
6.5 Setting of Multi Task



(1) Mode selection

2 RETURN

- Select MULTI TASK SET.

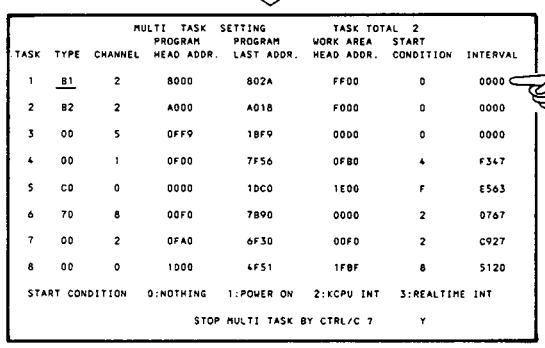


(2) Setting of the number of tasks

2 RETURN

→ Number of programs to be run (1 ~ 8)

- First set the number of programs to be run (1 ~ 8). In this example, the number of programs is two.



(3) Setting of BASIC program number

- Register the tasks of BASIC program.
- The same BASIC program number cannot be set for plural tasks.
- The task number is related to the priority of program run during run of multi task. (Generally, a lower task number has priority to a higher number.)

B 1 RETURN

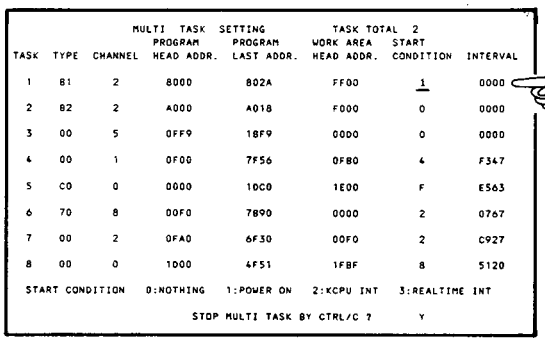
→ Program number

→ Should always be "B".

- The maximum number of tasks is eight. The programs are run per 16 lines according to priority (TASK 1 ~ 8). In this screen, set the priority of prepared programs.

WARNING

The same TYPE number, which can be set, is only one within TASK 1 ~ 8.



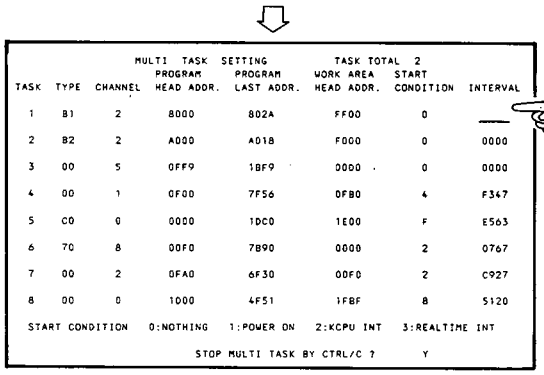
(4) Setting of task start condition

1 RETURN

- Set the start condition of multi task.

Setting Number	Application
0	Non execution
1	Start by power-on
2	Start by interruption caused by KCPU
3	Start by real time interruption

Next page



MULTI TASK SETTING TASK TOTAL 2

TASK	TYPE	CHANNEL	PROGRAM HEAD ADDR.	PROGRAM LAST ADDR.	WORK AREA HEAD ADDR.	START CONDITION	INTERVAL
1	B1	2	8000	802A	FF00	0	
2	B2	2	A000	A018	F000	0	0000
3	00	5	0FF9	1BF9	0000	0	0000
4	00	1	0F00	7F56	0FB0	4	F347
5	C0	0	0000	1DC0	1E00	F	E563
6	70	8	00F0	7B90	0000	2	0767
7	00	2	0FA0	6F30	00F0	2	C927
8	00	0	1000	4F51	1FBF	8	5120

START CONDITION 0:NOTHING 1:POWER ON 2:KCPU INT 3:REALTIME INT

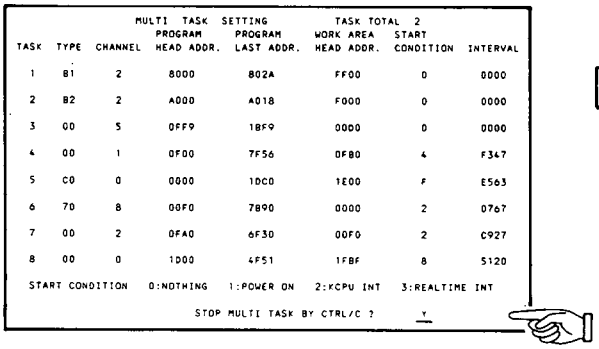
STOP MULTI TASK BY CTRL/C ? Y

(5) Setting of multi task start interval

0 1 0 0 RETURN

Setting range: 1 ~ 9999
(in 10 ms increments)

- When "3" (REAL TIME INT) has been set in the START CONDITION, set a start interval in the INTERVAL. Then, the task starts at the set intervals of time.



MULTI TASK SETTING TASK TOTAL 2

TASK	TYPE	CHANNEL	PROGRAM HEAD ADDR.	PROGRAM LAST ADDR.	WORK AREA HEAD ADDR.	START CONDITION	INTERVAL
1	B1	2	8000	802A	FF00	0	0000
2	B2	2	A000	A018	F000	0	0000
3	00	5	0FF9	1BF9	0000	0	0000
4	00	1	0F00	7F56	0FB0	4	F347
5	C0	0	0000	1DC0	1E00	F	E563
6	70	8	00F0	7B90	0000	2	0767
7	00	2	0FA0	6F30	00F0	2	C927
8	00	0	1000	4F51	1FBF	8	5120

START CONDITION 0:NOTHING 1:POWER ON 2:KCPU INT 3:REALTIME INT

STOP MULTI TASK BY CTRL/C ? Y

(6) Setting of multi task stop condition

RETURN

- Set the stop condition of multi task run. Normally, set Y(YES). When the CTRL/C keys of the console are pressed during run of multi task, multi task is stopped and the CRT returns to the mode select menu screen. When N (NO) has been set, move the RUN/STOP key on the front panel of KD51E unit to STOP position and then press the CTRL/C keys of the console. Then, the CRT returns to the mode select menu screen.

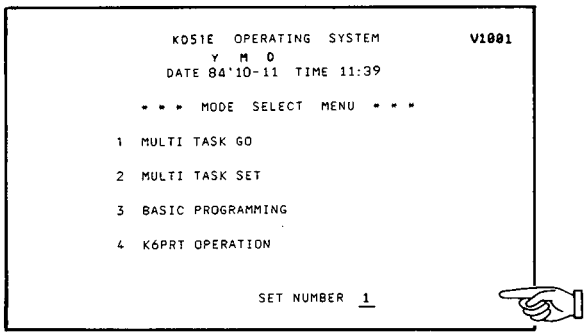
Setting of Connected Printer
(See Section 6.8.)

CAUTION

The relation between the BASIC program run and the RUN/STOP key position is as follows:

Mode \ Switch Key Position		Stop Position	RUN Position
During run of multi task	Multi task stop condition has been set so that multi task is stopped when CTRL/C keys of console.	Program is not run.	Program is run. However, run is stopped by pressing CTRL/C keys of console.
	Multi task stop condition has been set so that bulti task is not stopped when CTRL/C keys are pressed.	Program is not run.	Program is run.
BASIC programming mode After execution of RUN command		Program is not run.	Program is run. However, run is stopped by pressing CTRL/C keys of console.

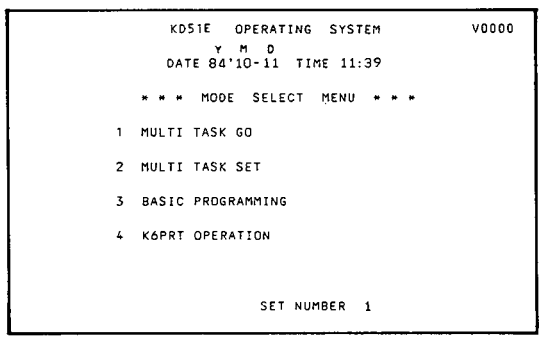
6.6 Setting of Multi Task Start



(1) Mode selection

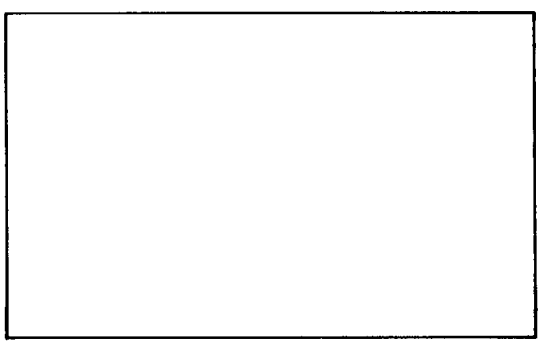
1 RETURN

- Select this mode to start multi task.



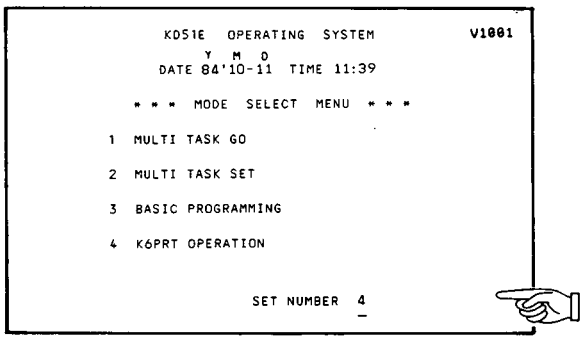
(2) Mode selection

- Set date and time. Proceed to the program run screen by pressing the RETURN key.



(3) Display of screen by BASIC program

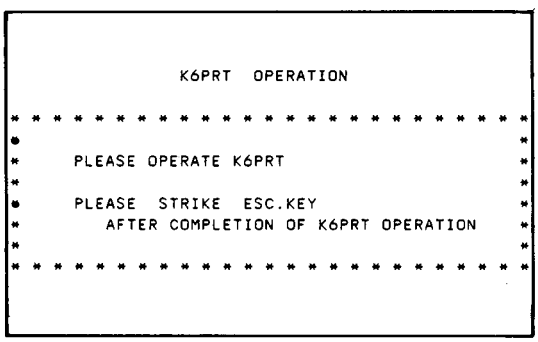
6.7 Setting of K6PRT (Handy recorder)



(1) Mode selection

4 RETURN

- Select this mode when the K6PRT is connected to CH3 of KD51E and operated.

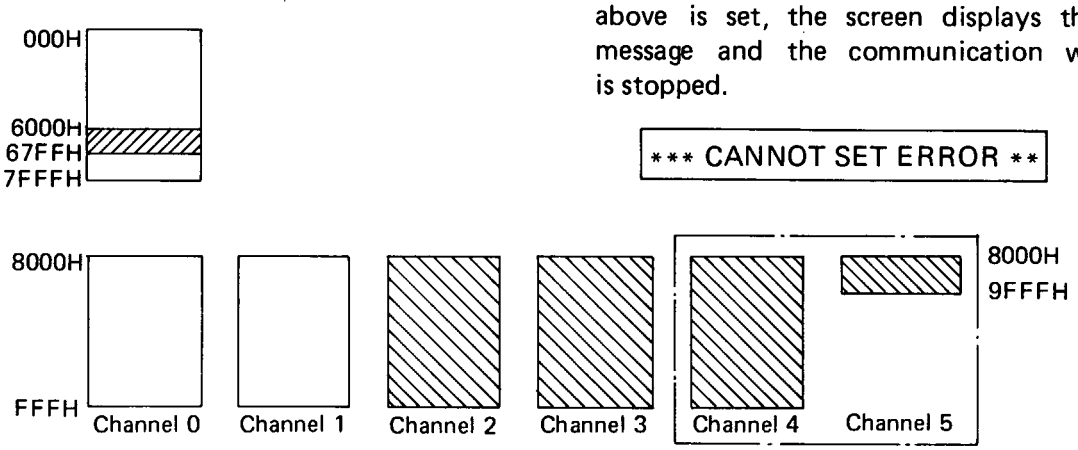


(2) K6PRT operation authorize

- When the screen shown at left is displayed, the K6PRT can be operated. After completion of K6PRT operation, press the ESC key. Then, the CRT returns to the mode select menu screen.

In the memory map shown in Fig. 6.5, the hatched ▨ memory areas can be accessed by the K6PRT. When specifying a channel by K6PRT, set channel 2 or 3 (channel 2 ~ 5 when the extension memory is loaded). If a channel other than the above is set, the screen displays the following message and the communication with K6PRT is stopped.

*** CANNOT SET ERROR **



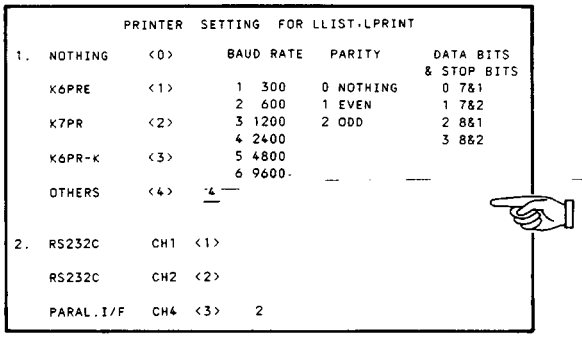
When extension memory (K3MB1) is loaded

Fig. 6.5 KD51E Memory Map

6.8 Setting of Connected Printer

1 Set the connected printer in the following cases:
- In the BASIC programming mode explained in Section 6.4, when setting has been completed in the BASIC programming screen.
- In the multi task setting mode explained in Section 6.5, when setting has been completed in the multi task setting screen.

2 Set the connected printer in the following connected printer setting screen.

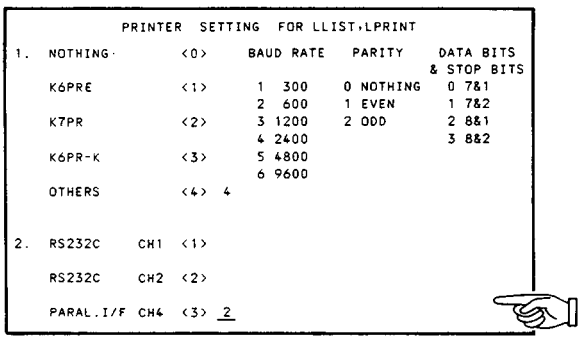


(1) Printer setting

4 RETURN

- Set the printer to be connected.

Setting Number	Printer to Be Connected
0	No connected printer
1	K6PRE
2	K7PR(2400BPS)
3	K6PR-K
4	General-purpose printer with RS-232-C (KD51PR)



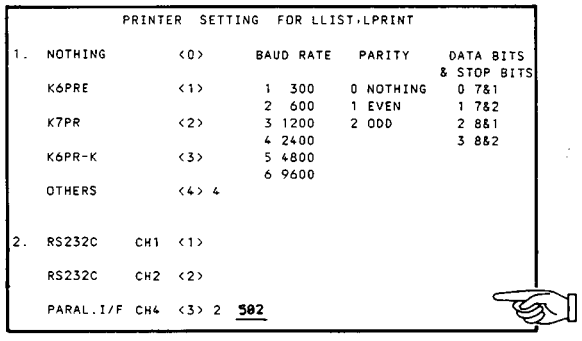
(2) Setting of channel to which printer is connected.

2 RETURN

- Set the channel to which the printer is connected. The channel to be set is CH1 or CH2.

WARNING

The channel 4 is not used. Do not set the channel 4.



(3) Setting of communication mode

5 0 2 RETURN

- Setting of data length, stop bit (1 ~ 3)
- Setting of parity (0 ~ 2)
- Setting of baud rate (1 ~ 6)

- When 4 (OTHERS) has been set in the printer setting (2), set the communication mode in three digits depending on the connected printer.

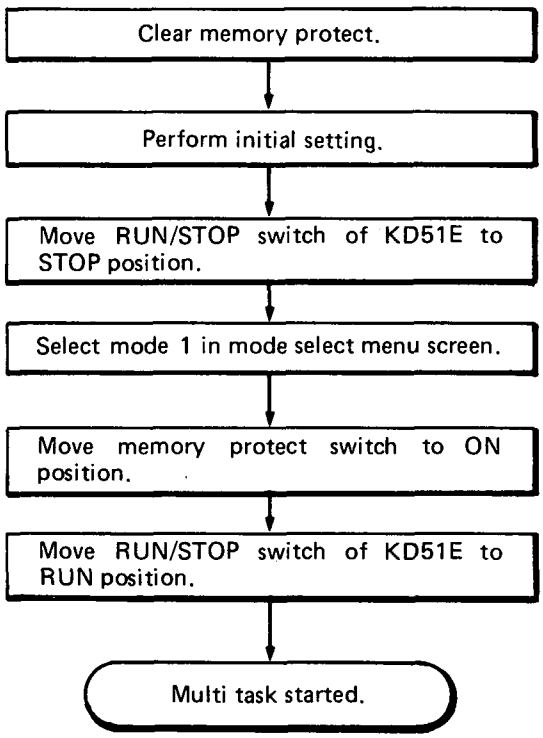
CAUTIONS FOR INITIAL SETTING

(1) Input codes, which are effective at the time of initialization (mode select menu screen, BASIC programming initial screen, multi task screen, printer setting screen), are only the following codes.

Character	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Y	N	SP	DEL	ESC	RETURN
Code (hexadecimal)	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46	59	4E	20	7F	1B	0D

(Note that "Y" and "N" are effective only when the CTRL/C keys are pressed during multi task setting.)

(2) At the time of initial setting, do not set memory protect to the system RAM area of addresses 4F00 ~ 4FFF. To set memory protect to the aforementioned area during run of multi task, follow the procedure shown below:



7. SOFTWARE CONFIGURATION

7. SOFTWARE CONFIGURATION

7.1 Software Configuration

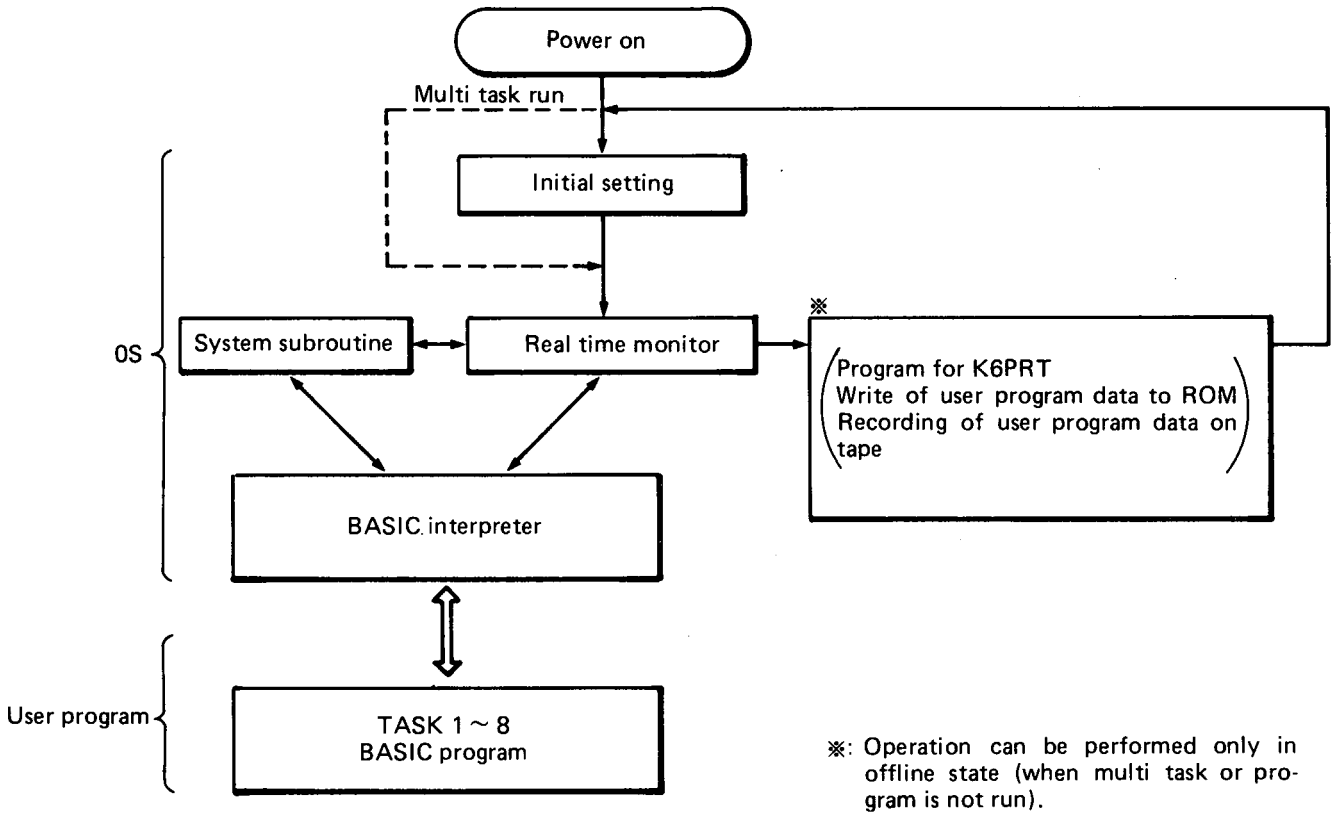


Fig. 7.1 Software Configuration

(1) As shown in Fig. 7.1, a maximum of eight user programs can be processed in parallel under control of real time monitor.

(2) The start conditions of user program are available in three types; "power on", "KCPU interruption caused by KCPU" and "real time interruption".

(3) Each task of user program is only the basic program.

7.2 Memory Map

7.2.1 Memory map

The memory areas have been expanded by using the latter half (32K bytes) of 64-byte memory space of KD51E, with the channels switched, as shown in Fig. 7.2.

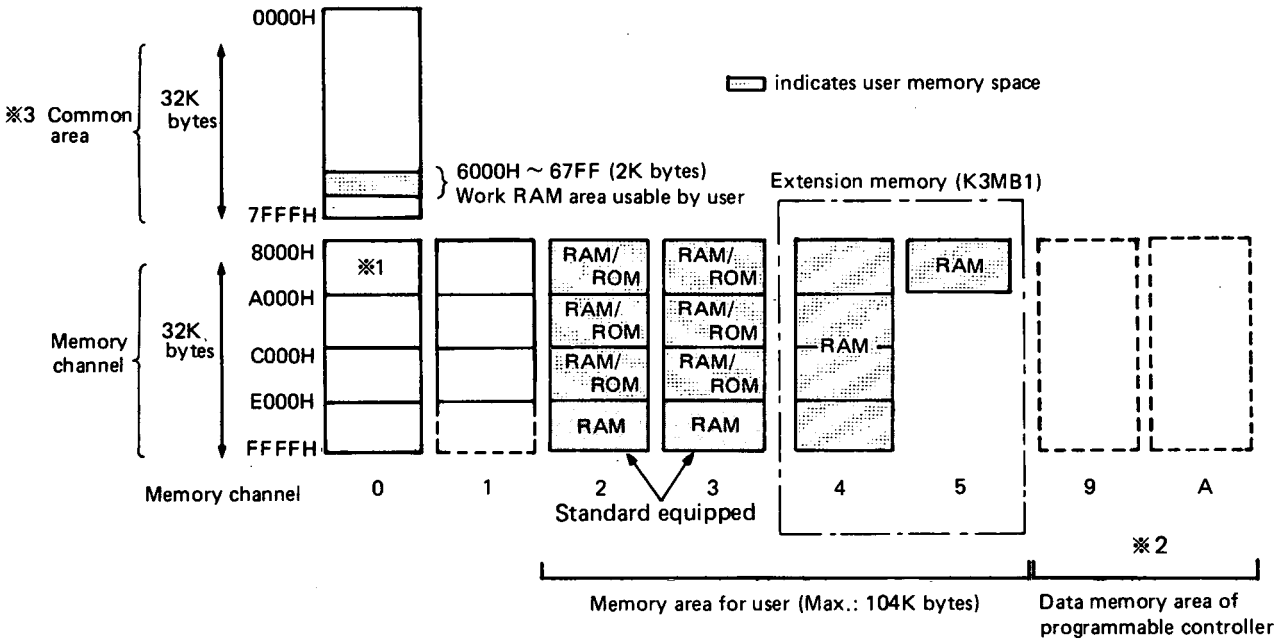


Fig. 7.2 Memory Map

※1: OS for system subroutine. See Table 8.2 on page.

※2: For details of channel 9 and channel A, see Section 7.2.2 and 7.2.3.

※3: The common area is an area which is commonly used for channels 0 ~ 5 and can be directly accessed by each channel.

7.2.2 Data memory maps of K2CPU-S3, K2HCPU and K2NCPU

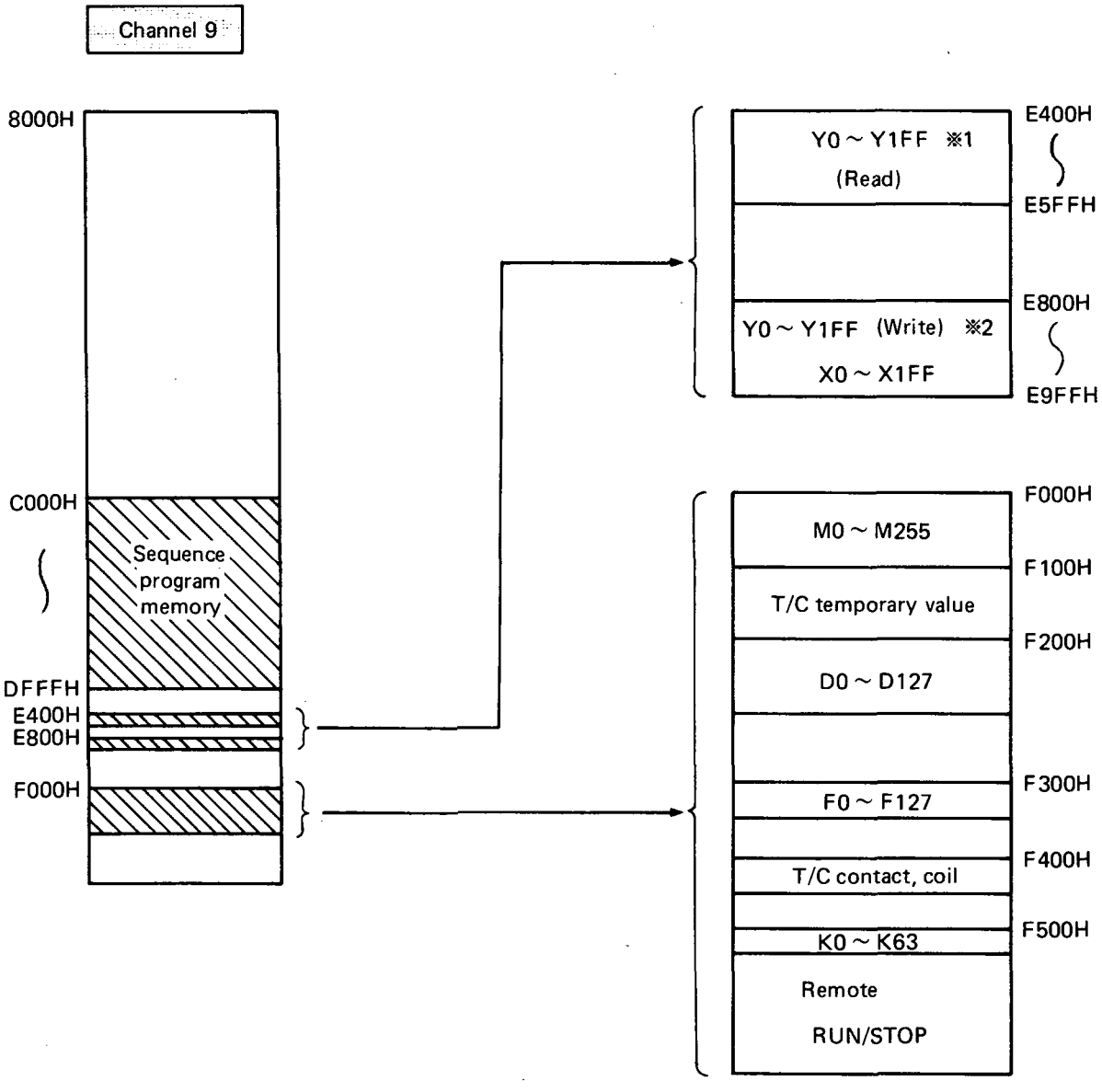


Fig. 7.3 Memory Map

CAUTION

※1 is an area for read of output Y, and cannot be used for write.

※2 is an area which allows read/write, and is used for write of output Y and read of input X.

7.2.3 Data memory map of K3NCPU, K3NCPUP2

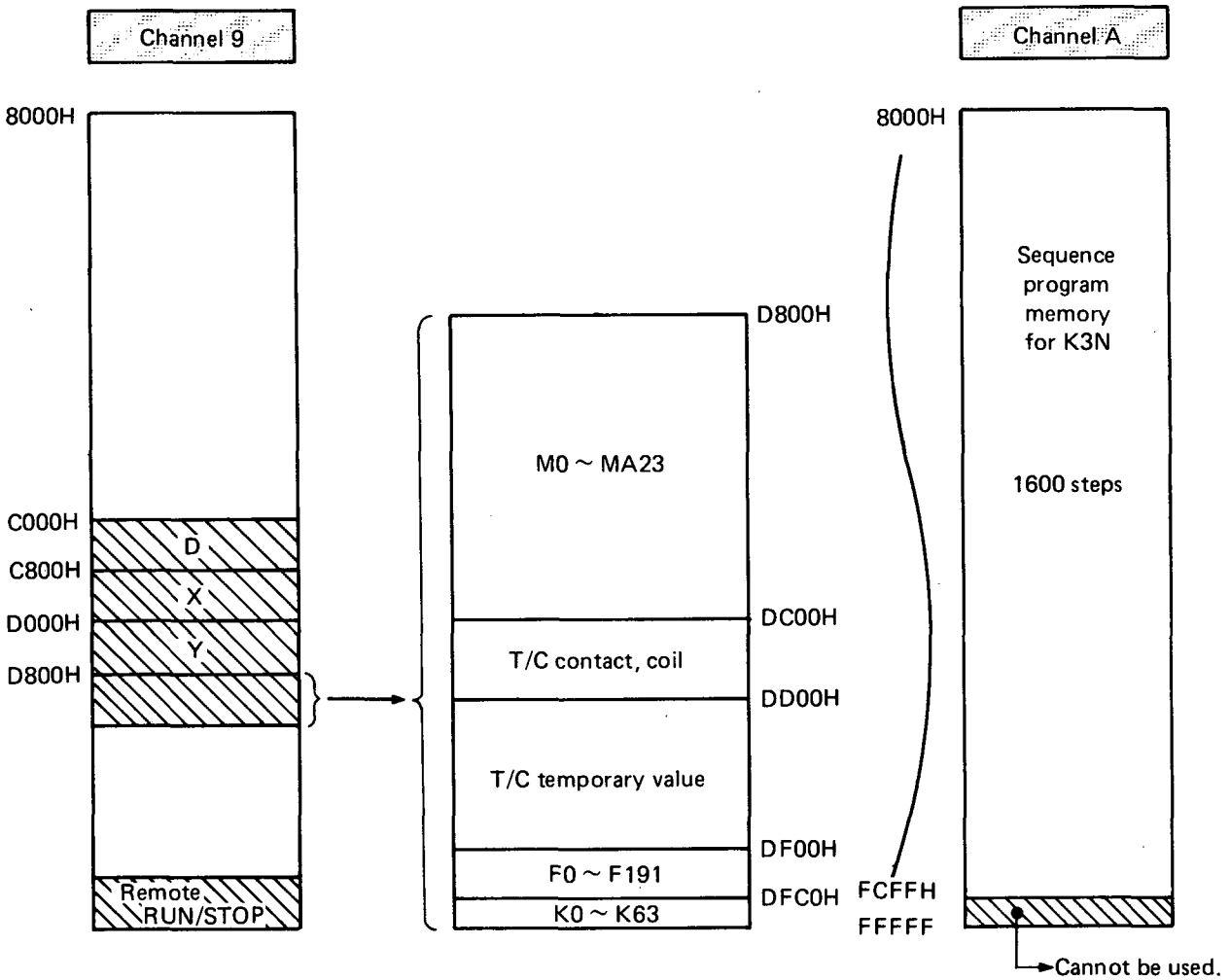


Fig. 7.4 Memory Map of Channel 9

Fig. 7.5 Memory Map of Channel A

CAUTION

The K2HCPU, K2NCPU and K2CPU-S3 cannot specify the channel A.

7.2.4 User memory space

1 User memory space is the hatched area of Fig. 7.2.

2 The maximum memory area is 2K bytes of channel 0 (addresses 6000H ~ 67FFH) + 32K bytes of channel 2 and 3 (addresses 8000H ~ FFFFH) x 2 + 40K bytes of extension memory = 106K bytes.

7.2.5 Access to another channel by user program

1 The 2K bytes of addresses 6000H ~ 67FFH in the common area can be directly accessed by any of the channels 0 ~ 5 without requiring the switching of channel.

2 The memory within the same channel can be accessed directly.
(In BASIC, this is equivalent to the use of PEEK command, POKE command, indirect variable, or @ array variable.)

3 To access another channel by user program, it is required to use the BASIC commands in Table 7.1. (For details, see GPC-BASIC.)

	Instruction Word
BASIC command	ZRD1, ZRD2, ZWR1, ZWR2, ZMOV

Table 7.1 BASIC Commands Which Allows Access to Another Channel

IMPORTANT

Access to memory area other than the user memory area shown in Fig. 7.2 may result in wild run of the system. Therefore, extreme care should be exercised.

8. INSTRUCTION WORDS

8. INSTRUCTION WORDS

The instruction words, which can be used for the KD51E, are GPC-BASIC commands. The KD51E has system subroutines, which can be used in the BASIC program.

8.1 GPC-BASIC

- The BASIC commands, which can be used for the KD51E, are shown in Table 8.1.
- For details of GPC-BASIC commands, see "GPC-BASIC" which is available separately. The graphic mode cannot be used.

8.2 System Subroutine

- The system subroutines, which can be used for the KD51E, are shown in Table 8.2.

[Designation of system subroutine]

- In the GPC-BASIC program, system subroutine is called by the CALL command.
- The format of CALL statement is as follows:

```
A = CALL (VARIABLE 1, VARIABLE 2, [VARIABLE 3, VARIABLE 4])
          └→ Channel of all system subroutione
```

VARIABLE 1 - The channel of all system subroutines is channel 0.
VARIABLE 2 - Address of system subroutine. (See Table 8.2.)
VARIABLE 3 - Variable transferred to system subroutine and set to registers (D), (E).
VARIABLE 4 - Variable transferred to system subroutine and set to registers (B), (C).

- For variable 3 and variable 4, see "GPC-BASIC".
- Variable, which is set to the work area, should be stored in the memory by the POKE command, etc. before executing the CALL command.

Command Name			Function
Key Command KEY	AUTO		Automatic generation of line number
	BYE		Return to BASIC programming data display screen
	CONT		Resumption of program run after BREAK
	COMPILE		Compilation to multi task executable program
	DELETE		Deletion of program from specified line number to specified line number
	EDIT		Correction of statement in one line
	EXECUTE		Run of program after "RUN" or "COMPILE"
	LIST	A B C	Display of program on screen
	LLIST		Print-out of program
	NEW		Erasure of program
	RENUM	A B	Renumbering of line number
	RUN		Run of program
	ZDV		Display of I/O console
	———		Erasure of line
Program Command PRG	BREAK		Resumption of program run after run, temporary stop or "CONT"
	CALL		Calling of machine language program
	CLS	A B	Deletion of CRT screen
	CLOSE	A B C	Closing of specified channel of RS-232-C/RS-433 (Setting of only channel 1 ~ 3 is possible.)
	END		End of program run
	FOR-----NEXT		Repeated run of program from "FOR" to "NEXT"
	GOTO		Move to specified line number
	GO SUB----- ----------RETURN		Move to specified subroutine Return from subroutine
	IF		Judgement of expression result
	INPUT	A B	Input through keyboard

When types of command are indicated [A] , [B] , [C] type ▩ can be used.

Table 8.1 List of BASIC Commands

Command Name			Function
PRG	INKEY		Substitution of input through keyboard for variable
	LET		Substitution of expression value for variable
	LOCATE	A B C	Moving of cursor position
	LPRINT	A B C	Print-out of data
	ONGOSUB . . . . . RETURN		Move to subroutine with line number specified by value of expression
	ONGOTO		Move to line number specified by value of expression
	OPEN	A B C	Opening of specified channel of RS-232-C/RS-422 (Setting of only channel 1 ~ 3 is possible.)
	PEEK		Read of one-byte data from specified address of memory
	POKE		Write of one-byte data to specified address of memory
	PRINT	A B	Display of data on screen
	REM		Used to write note. Has no influence with run.
	SIZE	A B	Display of text program capacity
	STOP		Stop of program run
	ZCRV		Inversion of character color on CRT screen
	ZDATE		Display of year, month, day, hour and minute
	ZIDV	A B C	Change-over of input console device
	ZMOV		Data transfer from one memory to another
	ZNOR		Restoration of inversed character color after "ZCRV"

In the Europe version, these commands are not processed (invalid).

Table 8.1 List of BASIC Commands (Continued)

Command Name			Function
PRG	ZODV	A / B	Change-over of output console device
	ZRD1		Read of one-byte data from specified channel
	ZRD2		Read of two-byte data from specified channel
	ZTIME		Suspension of run for specified interval of time
	ZWR1		Write of one-byte data to specified channel
	ZWR2		Write of two-byte data to specified channel
Instrinsic function INT	ABF		Absolute value of mathematical expression value (real number)
	ABS		Absolute value of mathematical expression value (integer)
	ACOS		Inverse cosine ($\cos^{-1}$) of mathematical expression
	ASIN		Inverse sine ($\sin^{-1}$) of mathematical expression
	ATAN		Inverse tangent ($\tan^{-1}$) of mathematical expression
	COS		Cosine (cos) of mathematical expression
	EXP		Value of exponential function of which base is "e" (e = 2.718281)
	LN		Value of natural logarithm (loge)
	LOG		Value of common logarithm (log10)
	NOT		Generation of "1" when value of mathematical expression is "0", generation of "0" when the value is other than "0".
	RND		Substitution of random number for variable
	SIN		Sine (sin) of mathematical expression
	SORT		Square root value of mathematical expression
	TAN		Tangent (tan) of mathematical expression
Arithmetic operator ALU	+		Addition
	−		Subtraction
	*		Multiplication
	/		Division
	∧		Exponent
	−		Sign inversion
	%		Remainder calculation

Table 8.1 List of BASIC Commands (Continued)

Command Name		Function
Comparing operator COM	=	Is equal to
	#	Is not equal to
	<	Is less than
	>	Is greater than
	<=	Is not greater than
	>=	Is not less than
Logic operation LOG	#	Negation (NOT)
	&	Logical multiplication (AND)
	!	Logical addition (OR)
	\	Exclusive logical sum (EXOR)

Table 8.1 List of BASIC Commands (Continued)

	System Subroutine	Function	Channel	Address
1	SPC ※1	KCPU discrimination	0	8078H
2	SCA	Clock write	0	803CH
3	SCB	Clock read	0	803FH
4	SBD4	BIN → BCD (four digits)	0	8042H
5	SDB4	BCD → BIN (four digits)	0	8045H
6	SBD6	BIN → BCD (six digits)	0	8048H
7	SDB6	BCD → BIN (six digits)	0	804BH
8	SBA	BIN addition (24 bits)	0	804EH
9	SBS	BIN subtraction (24 bits)	0	8051H
10	SBM	BIN multiplication (24 bits)	0	8054H
11	SBW	BIN division (24 bits)	0	8057H
12	SAI	ASCII (hexadecimal) → BIN	0	8060H
13	SIA	BIN → ASCII (hexadecimal)	0	8063H
14	SAF	ASCII → real number	0	8066H
15	SFA	Real number → ASCII	0	8069H
16	SBF	Integer → real number	0	806CH
17	SFB	Real number → integer	0	806FH
18	SAN	ASCII (decimal) → BIN	0	8072H
19	SNA	BIN → ASCII (decimal)	0	8075H
20	SRB	8251 block data read	0	8009H
21	SWB	8251 block data write	0	800CH
22	SRK	Sequence circuit read	0	8024H
23	SWK	Sequence circuit write	0	8027H
24	SRI	Sequence instruction read	0	802AH
25	SWI	Sequence instruction write	0	802DH
26	SRC	Number of received data bytes	0	800FH
27	SRF	Number of vacant bytes of receive buffer	0	8012H
28	SKC	KCPU run/stop check	0	8030H
29	SKR ※2	KCPU remote run	0	8033H
30	SKP ※2	KCPU remote stop	0	8036H
31	SKI	Interval setting of access time to KCPU	0	8039H
32	SHX	System Subroutine	0	8015H
33	SHD	DR control (intial state)	0	8018H
34	SEA	No conversion into code (initial state)	0	8021H
35	SAE	Conversion into EBCDIC code	0	801EH
36	SEN	END instruction search	0	801BH

Table 8.2 List of System Subroutines

CAUTION

※1. In regards to the discrimination of KCPU by the systm subroutine (SPC), SPC discriminates K3(N)CPU or K2 (K2NCPU, K2CPU-S3, K2HCPU).
Discrimination between K2CPU-S3, K2HCPU and K2NCPU cannot be made.

※2. When the RUN/STOP switch of KCPU is in RUN position, the remote run/stop of KCPU can be effected by the KD51E. However, the remote run/stop of KCPU cannot be used for the K2CPU-S3. When remote run (SKR) or remote stop (SKP) is called by the K2CPU-S3, output is as shown below:

	SKR call		SKP call	
	Output		Output	
During KCPU run	0	Normally completed	7	Error
During KCPU stop	7	Error	0	Normally completed

9. COMMUNICATION FUNCTION WITH KCPU

9. COMMUNICATION FUNCTION WITH KCPU

The KD51E is a unit which exclusively uses 48 points and is provided with 16-point general-purpose I/O image. The user can utilize these 16 points (for each of X and Y) to communicate with the KCPU.
This section describes the usage and cautions of the access method to KCPU by the BASIC program.

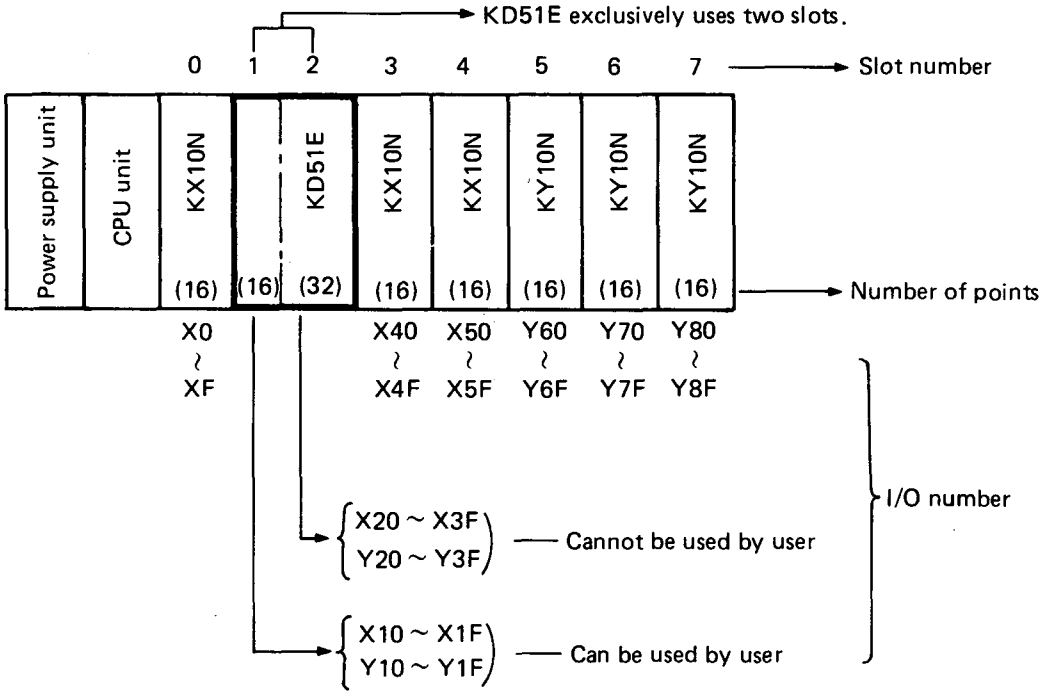


Fig. 9.1 Allocation of I/O Numbers to KD51E

CAUTION

1. The 16 points of general-purpose I/O image can be used for the sequence program of KCPU and the BASIC program of KD51E.

2. The 16 points of general-purpose I/O image cannot be output to the exterior of KD51E.

3. The KD51E exclusively uses two slots.

4. Read/write of one-byte data is performed in an access to the KCPU.

5. In each access by KD51E during run of KCPU, scan time is elongated approximately 1 ms.

6. When the KCPU is accessed during run, set the interval of acces time so that access is not detected by the WDT (watchdog timer).

   (1) Set the interval of access time is set at one location of any task, the set value of access time interval is applied thereafter.

   (2) When the interval of access time is set at one location of any task, the set value of access time interval is applied thereafter.

   (3) The set value of access time interval can be commonly used for each task.

   (4) When the access time interval has not been set in the user program, the KCPU is accessed once 10 ms.

   (5) Set the access time so that it satisfies the following expression:

   Scan timer (ms) + (1 ms x the number of accesses) < KCPU watchdog timer

7. During stop of KCPU, the setting of access time interval to "0" allows access at any time and read/write operation at high speed. (During run of KCPU, the setting of access time interval to "0" greatly elongates scan time if vast data of KCPU are read/written. Therefore, caution must be exercised.

9.1 Read/Write by General-Purpose I/O

- Read/write by general-purpose I/O is used to utilize the I/O data of programmable controller for the BASIC program of KD51E.
- Since the data of general-purpose I/O are stored in the common area memory of KD51E, read/write of these data can be performed from the BASIC program without switching the channel.
- When the setting of task start condition is "KCPU INT" (see Section 6.5), the 16th point of general-purpose output is used as input for interruption by KCPU.

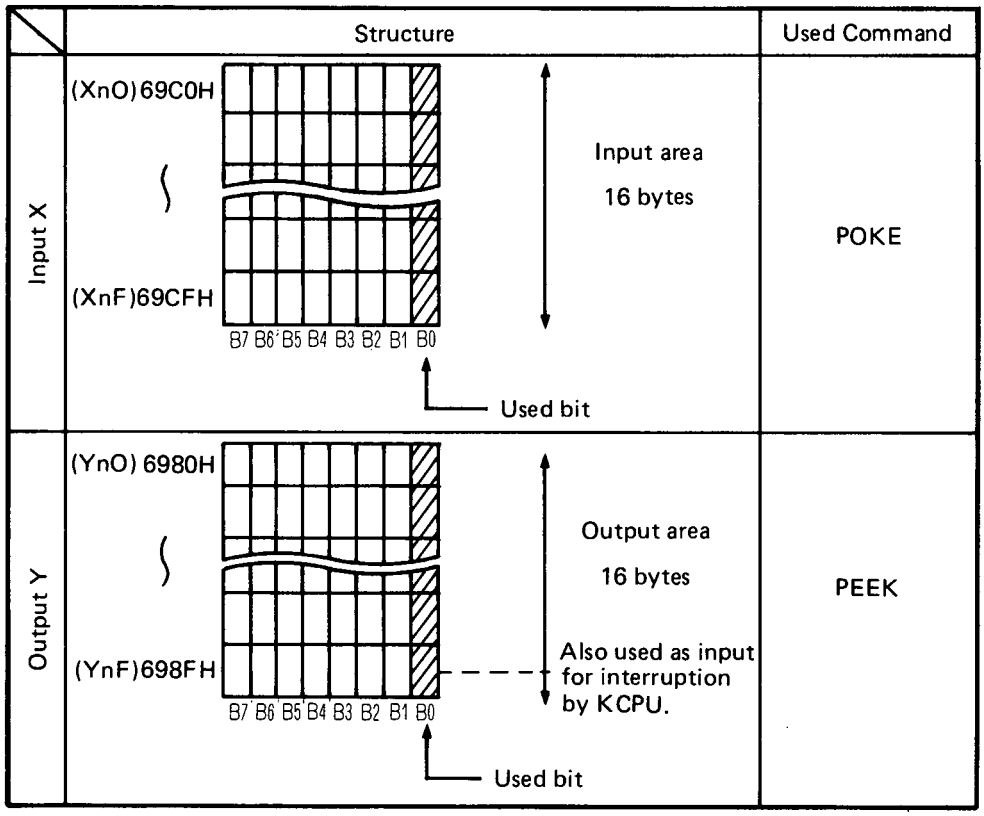


Table 9.1 Memory Addresses of KD51E image

[Usage]

- Used to discriminate failure types of 1 ~ 15 points of output image by interrupting the KD51E by the KCPU (by use of the 16th point of output) at the time of failure.
- Used to control sequence program by turning on/off 1 ~ 16 points of output image Y by the BASIC program. ("KCPU INT" cannot be set as the task start condition.)

EXAMPLE

Program which reads the on/off data of Y53 and turns off X58 when Y53 is on and turns on X58 when Y53 is off in the unit configuration shown in Fig. 9.2.

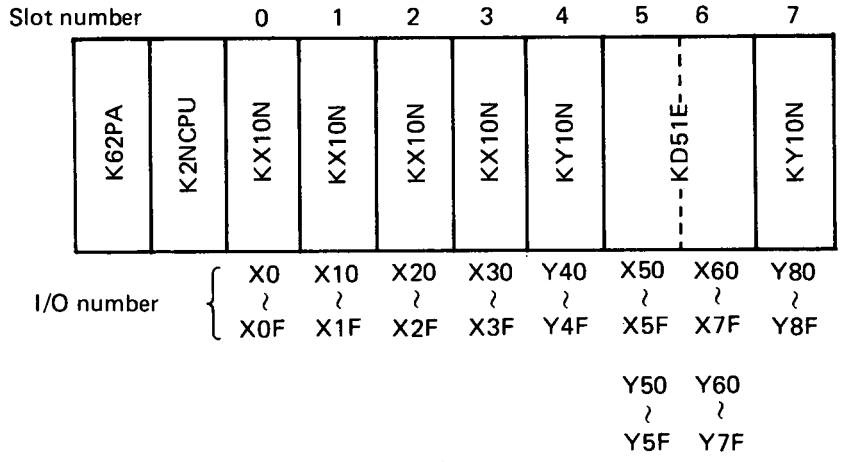


Fig. 9.2 Unit Configuration

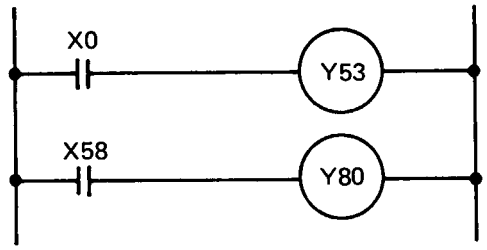


Fig. 9.3 Program of Programmable Controller

```
90   A = $6983 ..... Address of Y53 is set to variable A.
100  B = $69C8 ..... Address of X58 is set to variable B.
110  C = PEEK (A) ..... Data of Y58 is read.
120  C = C & 1 ..... Only the first bit of read data is effective.
130  IF C = GOTO 170 ..... When the first bit is 1 (ON), run is started from line number 170.
140  D = 1
150  POKE B, D ..... When the first bit is 0 (OFF), 1 (ON) is written to Y58.
160  END
170  D = 0
180  POKE B, D ..... When the first bit is 1 (ON), 0 (OFF) is written to Y58.
190  END
```

Fig. 9.4 BASIC Program Example of KD51E

9.2 Interruption Caused by KCPU

- Used for the interruption of the KD51E by the KCPU is the 16th point of output image Y among general-purpose I/O images.
- At the rise of the 16th point of output Y, the KD51E is interrupted by the KCPU.

[Usage]

When "Interruption by KCPU" has been set as the task start condition during multi task setting (see Section 6.5), the corresponding task starts each time the 16th point of output image Y is turned on. The interruption is reset by the END instruction in the corresponding task.

EXAMPLE

Fig. 9.5 shows a program, which interrupts the KD51E when X1 turns on in the unit configuration shown in Fig. 9.4.

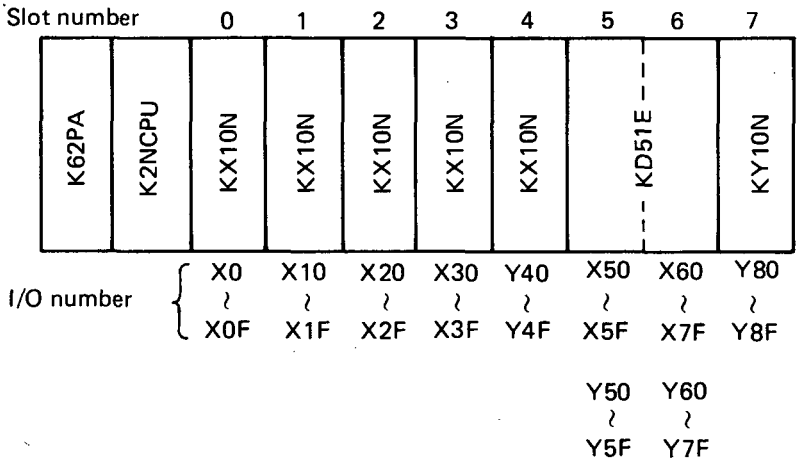


Fig. 9.5 Unit Configuration

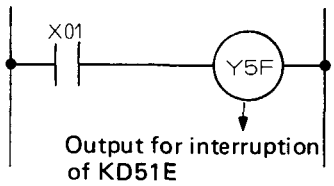


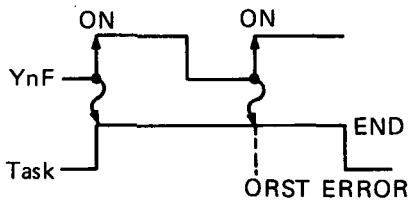
Fig. 9.6 Program Example

CAUTION

Only one task can be set for "interruption caused by KCPU".

IMPORTANT

During interruption caused by KCPU, when YnF turns on again prior to the execution of END command, ORST error results, but the run of task does not stop.



9.3 Read/Write of Sequence Program

- The following system subroutines are used for read/write of sequence program.

System Subroutine	Application
SRK	Read of sequence circuit
SWK	Write of sequence circuit
SRI	Read of sequence instruction
SWI	Write of sequence instruction

IMPORTANT

Only SWK cannot be used during run.

- When the above system subroutines are used, use the following system subroutines as required.

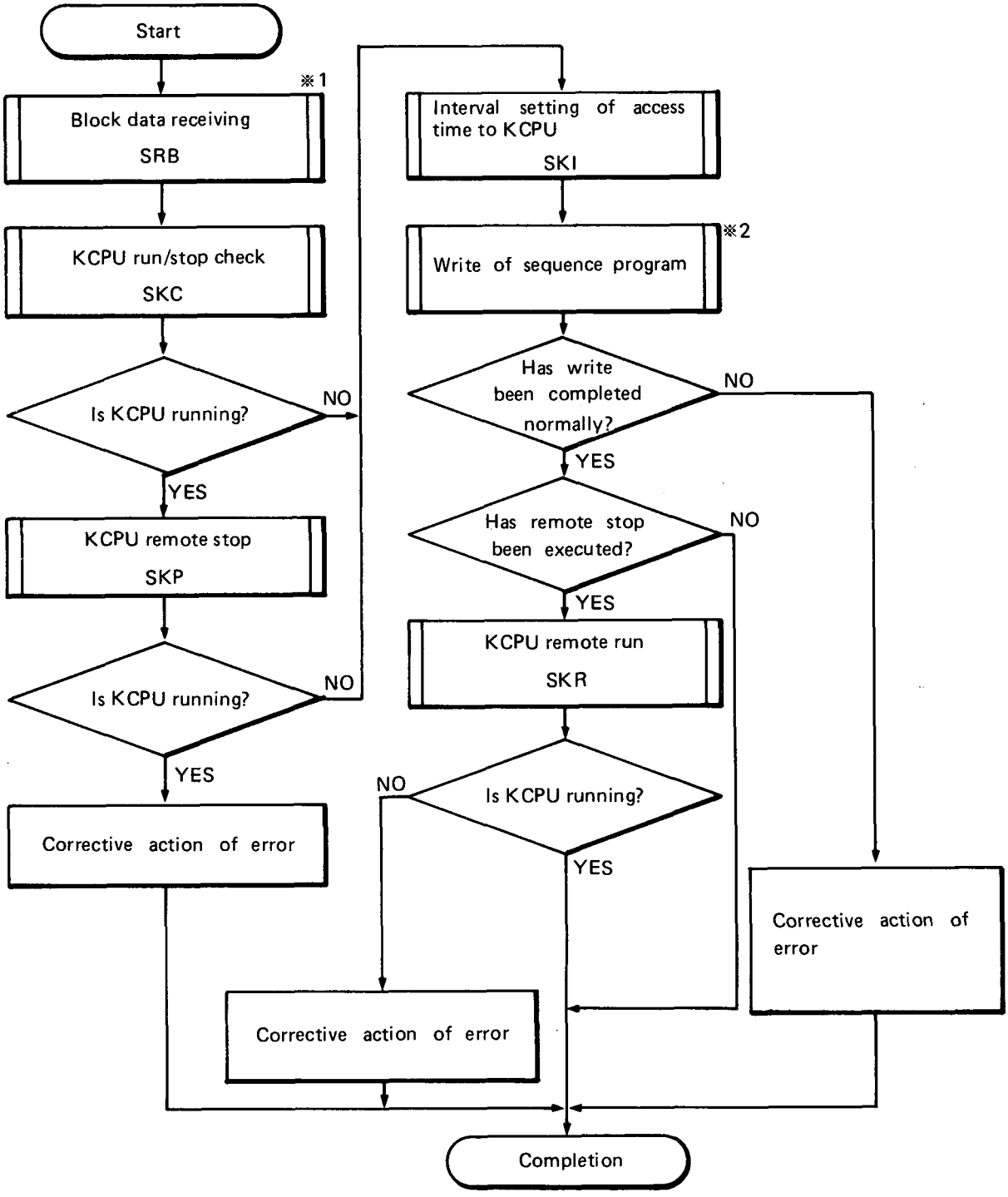
System Subroutine	Application
SKC	RUN/STOP check of KCPU
SKP	Remote stop of KCPU
SKR	Remote RUN of KCPU
SKI	Interval setting of access time to KCPU

CAUTION

1. For data format which is used to store sequence circuit and sequence instruction in the KD51E, see Section 10.
2. The KCPU is accessed in units of one byte.
3. Scan time is elongated 1 ms by an access to the KCPU.
4. The interval of access time to KCPU, which has been set by SKI, influences the scan time of sequence program. Therefore, caution should be exercised. (When the interval of access time to KCPU during stop of KCPU is set to "0", processing can be performed faster.)
5. The data of sequence circuits and sequence instructions, which have been read from the programmable controller, can be stored only in 6000H ~ 67FFH of common area.
6. Store the data of sequence circuits and sequence instructions, which have been written to the programmable controller, in 6000H ~ 67FFH of common area.

9.4 Schematic Flow Chart of BASIC Program by Use of System Subroutines

(1) Example of write of sequence program to KCPU from external computer

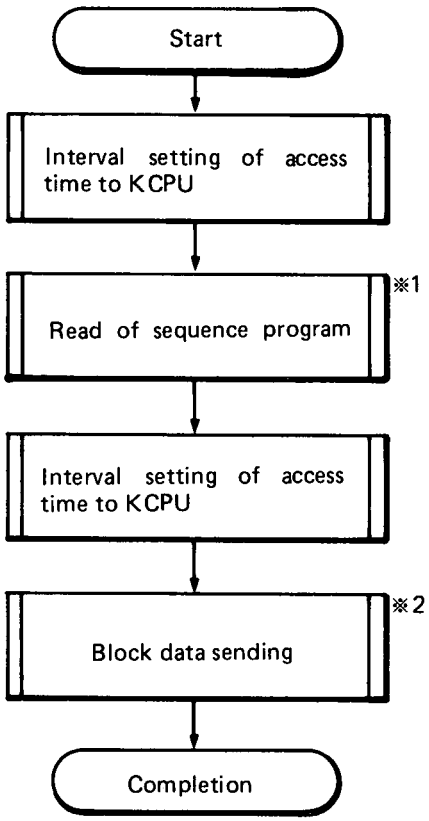


CAUTION

※1. The program data, which are desired to be written, are received from external equipment. In the ladder mode, data of one circuit are received. In the list mode, data of specified step are received.

※2. In the ladder mode, SWK is used. In the list mode, SWI is used.

(2) Example of read of sequence program from KCPU



CAUTION

1. (※1) In the ladder mode, SRK is used. In the list mode, SRI is used.
2. (※2) The program data, which have been read from the programmable controller, are sent to the external equipment.
3. Read can be also performed during run of KCPU.

9.5 Read/Write by BASIC Command

- For read/write of sequence data, the following commands are used.

Command	Application
ZRD1	1-byte read
ZRD2	2-byte read
ZWR1	1-byte write
ZWR2	2-byte write
ZMOV	Block transfer of memory

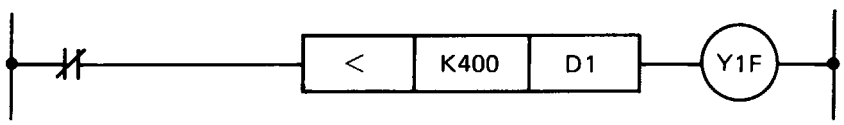
[Usage]

- Used for read/write of sequence data by use of the aforementioned commands.
- Channels used for read/write are CH9 and A of KD51E.

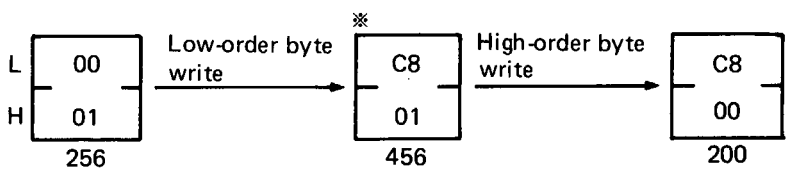
CAUTION

1. The ZWD2 command is used for the write of two-byte data and writes data from KD51E to KCPU per one byte. After the write of low-order byte, therefore, error may occur until the write of high-order byte is completed.

Example 1: To change the content of D1



When the content of D1 is changed by ZWR2 to "200" when the temporary value of D1 is "256" in the above program, the following will result.



When the sequence program is executed at the timing shown by ※ mark, Y1F turns on.

Example 2:

To avoid error as shown in Example 1, it is recommended to use the following sequence program and BASIC program.

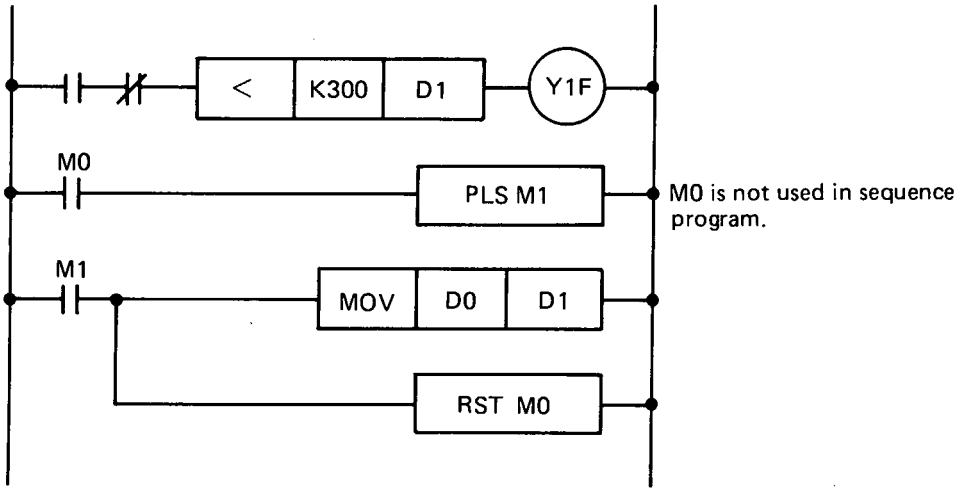


Fig. 9.7 Sequence program

```
LIST.
 OK.
 >100    A= ZWR2 ($9, $F200, $200). . . . . ."200" is written to D0.
  110    B = ZWR1 ($9, $F000, $1) . . . . . .M0 is turned on.
  120    END
```

Fig. 9.8 BASIC Program

2. Like the ZWR2 command, the SRD2 command reads two-byte data per byte. Therefore, the same measure as in CAUTION 1. must also be taken when the ZRD2 command is used.

3. Since the read/write of two-byte data by the ZWR2/ZRD2 command is performed consecutively when the K2N is connected, the trouble as in CAUTION 1. will not occur.

MEMO

10. PREPARATION OF SEQUENCE PROGRAM

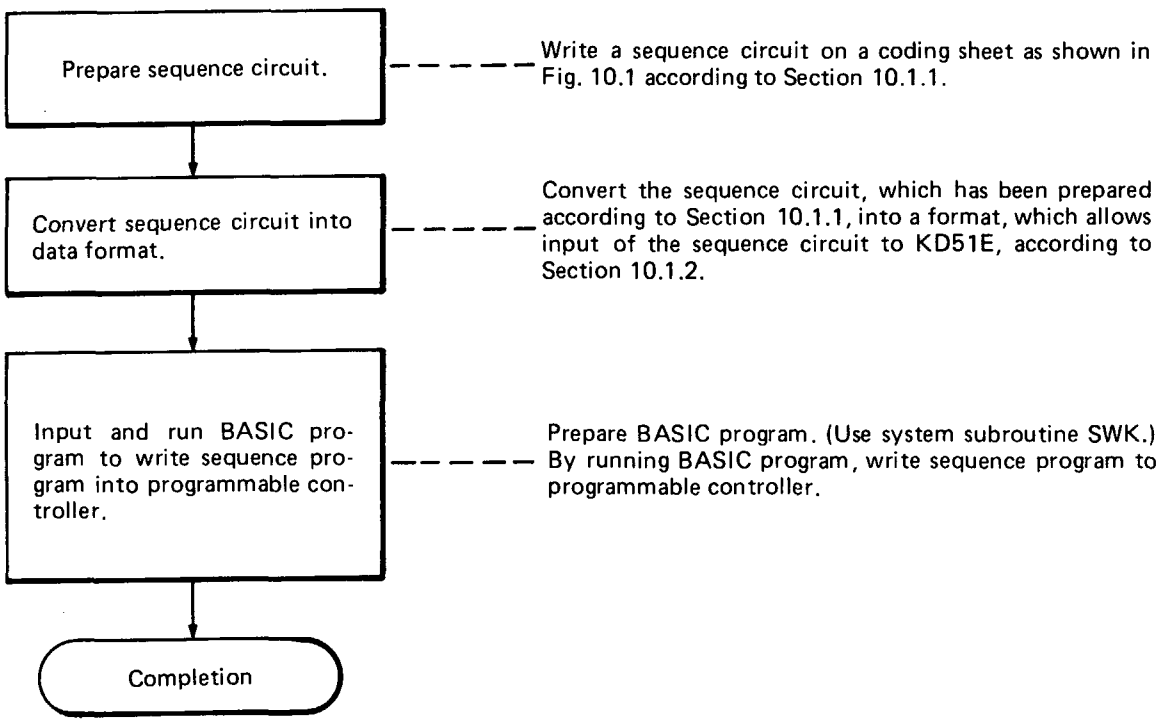
10. PREPARATION OF SEQUENCE PROGRAM

- Sequence program can be written and read from the I/O console, which is connected to the KD51E, to the programmable controller.
- Write and read sequence program after converting the sequence circuit into data format. Two types of write/read methods are available; ladder mode and list mode.
- Read/write of sequence program, which has been converted into specified data format, is performed by the following system subroutines:

Read in ladder mode:	SRK	For usage, see Section 9.3 and "GPC-BASIC".
Write in ladder mode:	SWK	
Read in list mode:	SRI	
Write in list mode:	SWI	

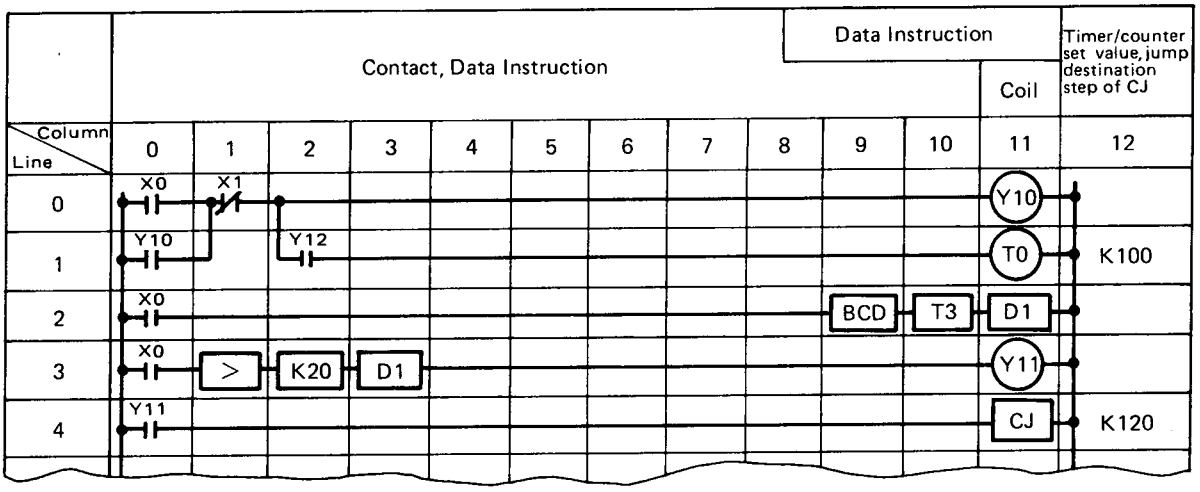
- This section explains the conversion of sequence program into data format at the time of write and read in ladder mode and list mode.
- Since the same data format is used for read and write, explanation is given only for write.

10.1 Write in Ladder Mode



10.1.1 Preparation of sequence circuit

Prepare a coding sheet shown in Fig. 10.1 and write a sequence program in the sheet.



- Columns 0 ~ 10: Enter the contact of sequence instruction and the data instruction of magnitude comparison ([>], [<], [=]). When data instruction other than magnitude comparison ([>], [<], [=]) is used, however, use columns 9 ~ 11. (See line 2.)
- Column 11: Enter the coil of sequence instruction.
- Columns 9 ~ 11: Enter data instruction other than magnitude comparison ([>], [<], [=]).
- Column 12: Enter constant when timer/counter or instruction such as conditional jump is used.

CAUTION

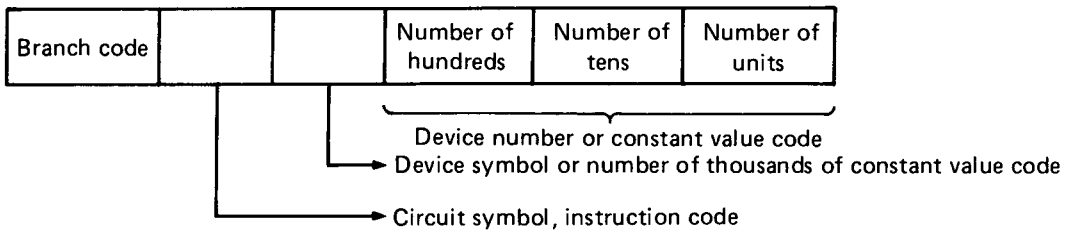
Since one step is indicated in each space of coding sheet, write a three-step instruction, such as MOV and BCD, in three spaces.

10.1.2 Conversion into data format

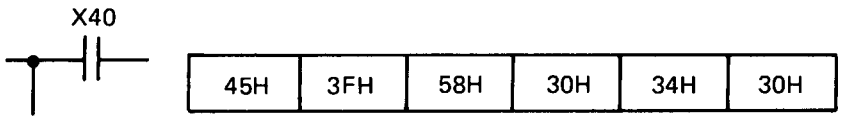
(1) Structure of data format

- To write the sequence circuit, which has been prepared in Section 10.1.1, to the programmable controller, by the KD51E, it is required to convert the circuit into the format explained below.
- Conversion into code is made per space of coding sheet.
- One space comprises ASCII codes of six-bytes or two-bytes structure as shown in the following page.

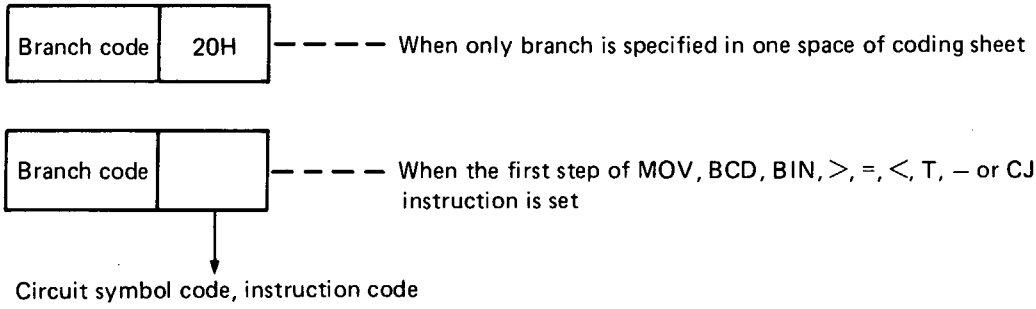
1 Structure of six bytes



Example



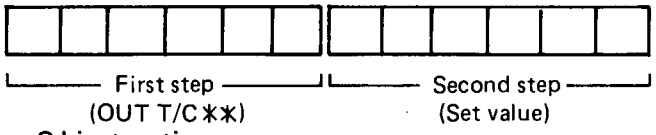
2 Structure of two bytes



CAUTION

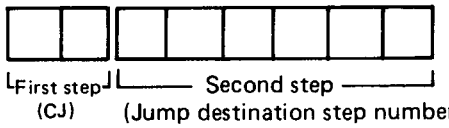
The structures of timer/counter coil designation, CJ instruction, data instructins (MOV, BCD, BIN, >, =, <, +, –) are as shown below:

- For timer, counter



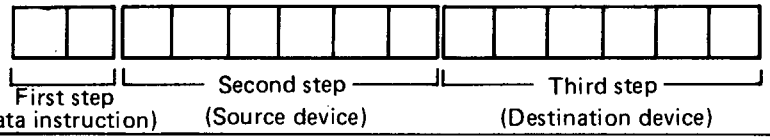
The second step is set value.

- For CJ instruction



Set jump destination step number at the second step.

- For data instruction (MOV, BCD, BIN, >, =, <, + and –)

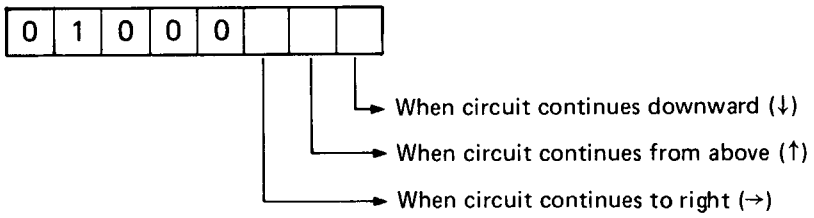


10

(2) Branch code

Branch is referred to as the state of line which connects, for example, a contact and a contact or a contact and a coil in a sequence circuit.
The branch code is determined depending on the state of branch which is located in front of contact or coil.

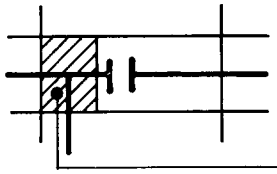
The structure of a branch code is as follows. When there is branch, set "1" to the lower three bits.



Branch Symbol	Branch Code	Character	Branch State
	40H	ⓐ	No branch
	42H	B	Branch to top
	43H	C	Branch to top and bottom
	44H	D	Branch to right
	45H	E	Branch to right and bottom
	46H	F	Branch to right and top
	47H	G	Branch to right, top and bottom

EXAMPLE

The branch data in one space is determined by the hatched area in the following figure.



Since the branch symbol is ⊤, the branch code is 45H.

CAUTION

When one instruction consists of two or more steps, set the branch codes of the second and third steps to 44H.

(3) Circuit symbols, instructions, digit codes

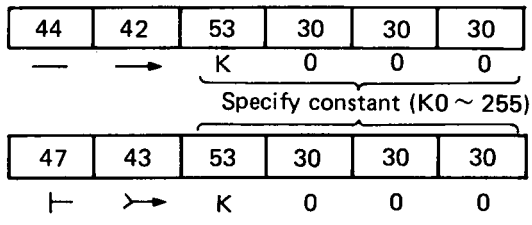
The circuit symbols, instructions and digit codes are used to specify the number of digits like symbols, instructions or MOV instruction in the sequence circuit.

Code	Character	Circuit Symbol	Instruction, Digit	Remarks
20H	Space		When there is only branch symbol	
3FH	?		"a" contact	
40H	@		"b" contact	
41H	A		OUT	
42H	B		Return	(Note 1)
43H	C		Return	
44H	D		RST	
45H	E		SFT	
46H	F		CJ	
47H	G		SET	
48H	H		PLS	
49H	I		MCR	
4AH	J		MC	
4BH	K		MOV	
4CH	L		>	
4DH	M		=	
4EH	N		<	
4FH	O		+	
50H	P		−	
51H	Q		BCD	
52H	R		BIN	
53H	S		Constant (K)	Used only when specifying the second and third steps of data instruction and CJ instruction.
54H	T		The number of significant digits of X, Y, M is one	
55H	U		The number of significant digits of X, Y, M is two	
56H	V		The number of significant digits of X, Y, M is three	
57H	W		The number of significant digits of X, Y, M is four	
58H	X		T, C, D	

10

CAUTION

When 11 or more contacts are provided consecutively, specify the return symbol 42H, in six-byte structure in the column 11 and specify the return symbol 43H, in six-byte structure in the column 0 of the line which follows.



(4) I/O symbols

Set symbols which indicate I/O devices, such as input, output, timer and counter, according to the following table. When constant is specified or when circuit component is of two-byte configuration, do not set these symbols.

Code	Character	I/O Symbol
4BH	K	K (only for MC, MCR and return)
58H	X	X (input)
59H	Y	Y (output)
4DH	M	M (internal relay)
54H	T	T (timer)
43H	C	C (counter)
46H	F	F (external failure memory, application instruction)
44H	D	D (data register)

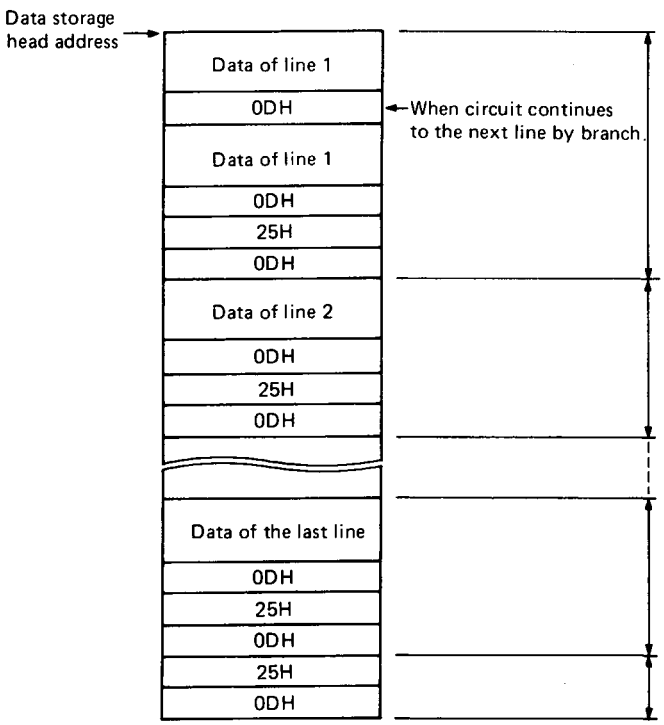
(5) I/O numbers, constant values

Set I/O number (the numbers of hundreds, tens and units) or constant value (the numbers of thousands, hundreds, tens and units) according to the following table:

Code	Character	Number	Code	Character	Number
30H	0	0	38H	8	8
31H	1	1	39H	9	9
32H	2	2	41H	A	A
33H	3	3	42H	B	B
34H	4	4	43H	C	C
35H	5	5	44H	D	D
36H	6	6	45H	E	E
37H	7	7	46H	F	F

10.1.3 Write to programmable controller

- Write the data of one circuit block in the sequence program to the programmable controller by calling the system subroutine SWK by the BASIC program.
- To mark off a line, write the ODH (CR key) code at the end of the line.
- To mark off a circuit block, write the 25H (%) code and ODH (CR key) code at the end of the circuit block.
- Write the END instruction (25H code and ODH code) at the end of the sequence circuit (below the marking-off codes of circuit block).
- The data of one circuit block are stored in the memory as shown below.

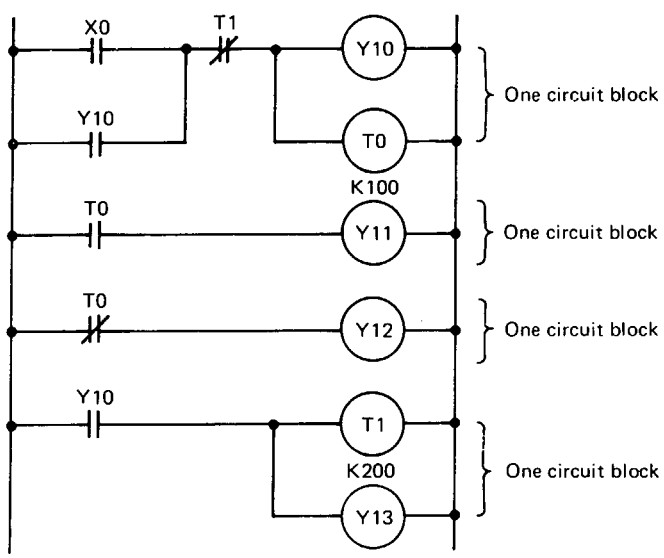


CAUTION

1. As the data storage area, only 6000H ~ 67FFH of common area can be used.
2. In the case of read from the programmable controller (SRK), also store the read data in 6000H ~ 67FFH of common area.

END instruction —
The END instruction comprises one circuit block. When writing the END instruction to the programmable controller, it is also required to call the system subroutine (SWK).
(The data byte length is "2".)

- One circuit block in the sequence circuit is as shown below.



Example

The following shows a write example of program to the programmable controller in the ladder mode.

1. In this example, the sequence program in Table 10.1 is written to the programmable controller.

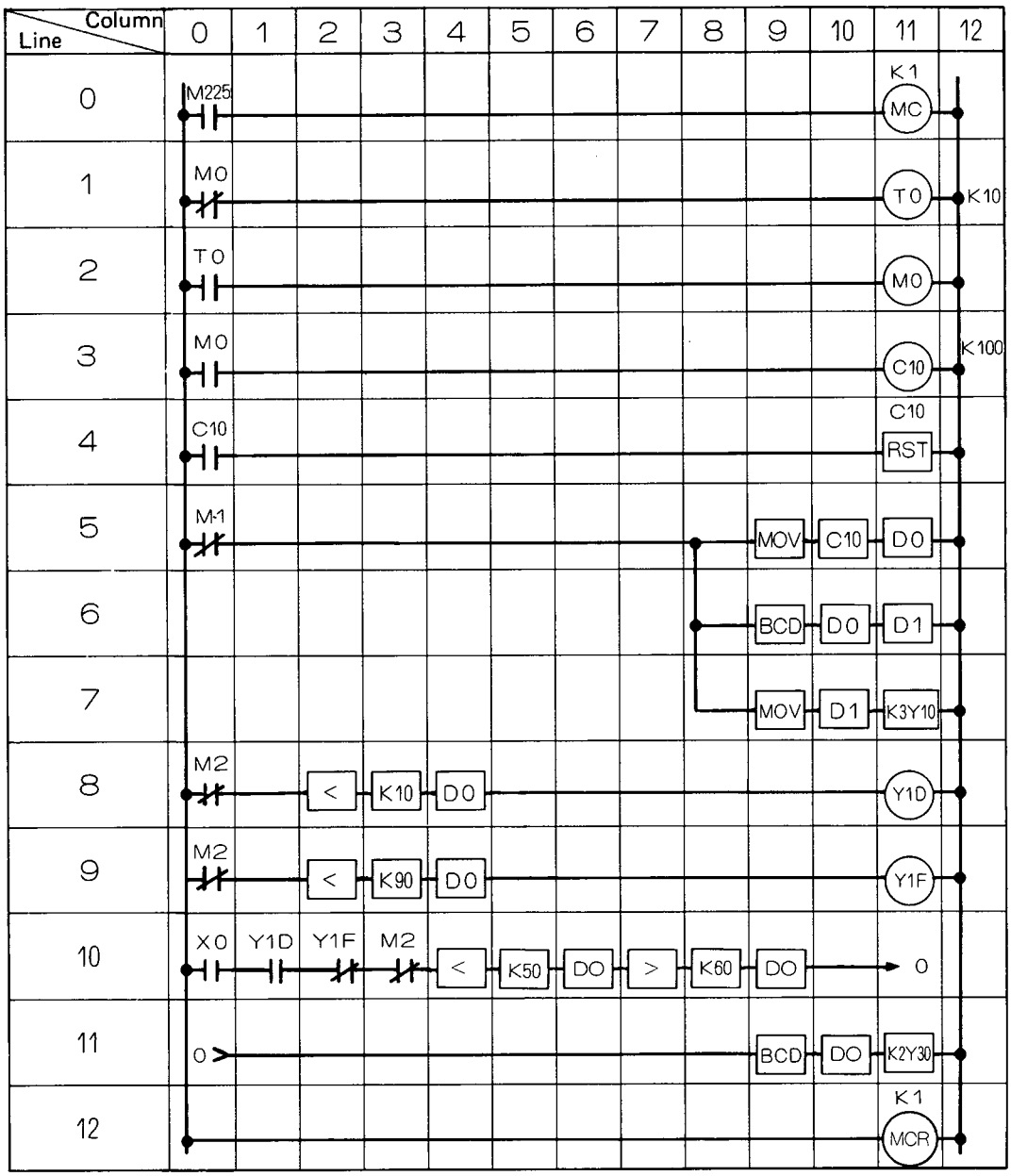


Table 10.1 Sequence Program Example

2. The following shows the data format which is used to key-in the sequence program shown in Table 10.1.

Line 0

Code	47	3F	4D	32	32	35	44	4A	4B	30	30	31	0D	25	0D
Character	G	?	M	2	2	5	D	J	K	O	O	1	CR	%	CR
	↕→	⊣⊢	M	2	2	5	→	MC	K	O	O	1			
	Column 0						Column 11								

←------ Indicates ASCII code.
←------ Indicates key name.
←------ Indicates circuit in preceding page.
←------ Indicates circuit position in preceding page.

Line 1

Code	47	40	4D	30	30	30	44	41	54	30	30	30	44	53	30	30	31	30	0D	25	0D
Character	G	@	M	O	O	O	D	A	T	O	O	O	D	S	O	O	1	O	CR	%	CR
	↕→	⊣⊢	M	O	O	O	→	-o-|	T	O	O	O	→	K	O	O	1	O			
	Column 0						Column 11						Column 12								

Line 2

Code	47	3F	54	O	O	O	44	41	4D	30	30	30	0D	25	0D
Character	G	?	T	O	O	O	D	A	M	O	O	O	CR	%	CR
	↕→	⊣⊢	T	O	O	O	→	-o-|	M	O	O	O			
	Column 0						Column 11								

Line 3

Code	47	3F	4D	30	30	30	44	41	43	30	31	30	44	53	30	31	30	30	0D	25	0D
Character	G	?	M	O	O	O	D	A	C	O	1	O	D	S	O	1	O	O	CR	%	CR
	↕→	⊣⊢	M	O	O	O	→	-o-|	C	O	1	O	→	K	O	1	O	O			
	Column 0						Column 11						Column 12								

Line 4

Code	47	3F	43	30	31	30	44	44	43	30	31	30	0D	25	0D
Character	G	?	C	O	1	O	D	D	C	O	1	O	CR	%	CR
	↕→	⊣⊢	C	O	1	O	→	RST	C	O	1	O			
	Column 0						Column 11								

Line 5

Code	47	40	4D	30	30	31	45	20	44	4B	44	58	43	30	31	30	44	58	44	30	30	30	0D
Character	G	@	M	O	O	1	E		D	K	D	X	C	O	1	O	D	X	D	O	O	O	CR
	↕→	⊣⊢	M	O	O	1	T		→	MOV	→		C	O	1	O	→		D	O	O	O	
	Column 0						Column 8		Column 9		Column 10						Column 11						

Line 6

Code	43	20	47	20	44	51	44	58	44	30	30	30	44	58	44	30	30	31	0D
Character	C		G		D	Q	D	X	D	O	O	O	D	X	D	O	O	1	CR
			↕→		→	BCD	→		D	O	O	O	→		D	O	O	1	
	Column 0		Column 8		Column 9		Column 10						Column 11						

Line 7

Code	43	20	46	20	44	4B	44	58	44	30	30	31	44	56	59	30	31	30	0D	25	0D
Character	C		F		D	K	D	X	D	O	O	1	D	V	Y	O	1	O	CR	%	CR
					→	MOV	→		D	O	O	1	→	K3	Y	O	1	O			
	Column 0		Column 8		Column 9		Column 10						Column 11								

Line 8

Code	47	40	4D	30	30	32	44	4E	44	53	30	30	31	30	44	58	44	30	30	30	44	41	59	30	31	44	0D	25	0D
Character	G	@	M	O	O	2	D	N	D	S	O	O	1	O	D	X	D	O	O	O	D	A	Y	O	1	D	CR	%	CR
	↕→	⊣⊢	M	O	O	2	→	<	→	K	O	O	1	O	→		D	O	O	O	→	-o-|	Y	O	1	D			
	Column 0						Column 2		Column 3						Column 4						Column 11								

Line 9

Code	47	40	4D	30	30	32	44	4E	44	53	30	30	39	30	44	58	44	30	30	30	44	41	59	30	31	46	0D	25	0D
Character	G	@	M	0	0	2	D	N	D	S	0	0	9	0	D	X	D	0	0	0	D	A	Y	0	1	F	CR	%	CR
	↕→	⊣⊢	M	0	0	2	→	<	→	K	0	0	9	0	→		D	0	0	0	→	⊶	Y	0	1	F			
	Column 0						Column 2		Column 3						Column 4						Column 11								

Line 10

Code	47	3F	58	30	30	30	44	3F	59	30	31	44	44	40	59	30	31	46	44	40	4D	30	30	32	44	4E			
Character	G	?	X	0	0	0	D	?	Y	0	1	D	D	@	Y	0	1	F	D	@	M	0	0	2	D	N			
	↕→	⊣⊢	X	0	0	0	→	⊣⊢	Y	0	0	D	→	⊣⊢	Y	0	1	F	→	⊣⊢	M	0	0	2	→	<			
	Column 0						Column 1						Column 2						Column 3						Column 4				

Line 10

Code	44	53	30	30	35	30	44	58	44	30	30	30	44	4C	44	53	30	30	36	30	44	58	44	30	30	30			
Character	D	S	0	0	5	0	D	X	D	0	0	0	D	L	D	S	0	0	6	0	D	X	D	0	0	0			
	→	K	0	0	5	0	→		D	0	0	0	→	>	→	K	0	0	6	0	→		D	0	0	0			
	Column 5						Column 6						Column 7		Column 8						Column 9								

Line 10

Code	44	42	4B	30	30	30	0D																						
Character	D	B	K	0	0	0	CR																						
	→	→	K	0	0	0																							
	Column 11																												

Line 11

Code	47	43	4B	30	30	30	44	51	44	58	44	30	30	30	44	55	59	30	33	30	0D	25	0D						
Character	G	C	K	0	0	0	D	Q	D	X	D	0	0	0	D	U	Y	0	3	0	CR	%	CR						
	↕→	>>	K	0	0	0	→	BC0	→		D	0	0	0	→	K2	Y	0	3	0									
	Column 0						Column 9		Column 10						Column 11														

Line 12

Code	47	20	44	49	4B	30	30	31	0D	25	0D	25	0D																
Character	G		D	I	K	0	0	1	CR	%	CR	%	CR																
	↕→		→	MCR	K	0	0	1																					
	Column 0		Column 11																										

10.2 Write in List Mode

The sequence program, which has been written in the list mode, can be written from the KD51E to the programmable controller by the system subroutine SWI, after converting it into ASCII codes.

10.2.1 Structure of data format

One step consists of 10 bytes as shown below.

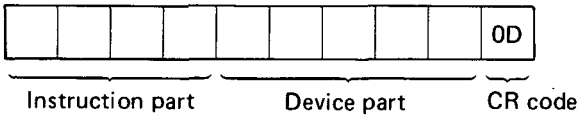


Fig. 10.2 Structure of Data Format

10.2.2 Conversion into ASCII codes

Instruction Part				Device Part					CR
L	D			X				0	CR
A	N	I		M			1	2	CR
O	U	T		T				9	CR
K								5	CR
M	O	V							CR
		K	4	X				0	CR
				D			1	0	CR
E	N	D							CR

Two-step instruction

Three-step instruction

Fig. 10.3 Program in List Mode

Instruction Part				Device Part					CR
4C	44	20	20	58	20	20	20	30	0D
41	4E	49	20	4D	20	20	31	32	0D
4F	55	54	20	54	20	20	20	39	0D
4B	20	20	20	20	20	20	20	35	0D
4D	3F	56	20	20	20	20	20	20	0D
20	20	4B	34	58	20	20	20	30	0D
20	20	20	20	44	20	20	31	30	0D
45	4E	44	20	20	20	20	20	20	0D

Fig. 10.4 Data Format

10

Fig. 10.2 shows the data format structure of one step which consists of 10 bytes. Convert the program in the list mode shown in Fig. 10.3, per character, into ASCII codes as shown in Fig. 10.4. The vacant spaces of the program in the list mode in Fig. 10.3 are converted into space codes (20H) in the data format.

WARNING

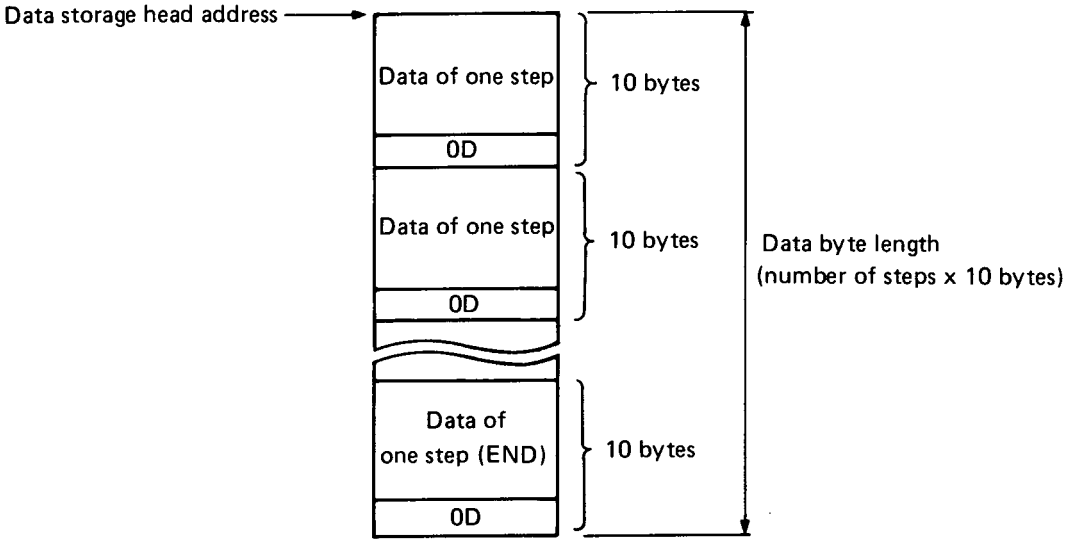
Error occurs when "X 001" is specified for a device number, etc. Be sure to specify "X ␣␣1" (␣ indicates a blank.)

The following table shows the ASCII codes which correspond to characters used in the list mode.

Character	ASCII Code	Character	ASCII Code	Character	ASCII Code	Character	ASCII Code	Character	ASCII Code
0	30H	A	41H	K	4BH	U	55H	–	2DH
1	31H	B	42H	L	4CH	V	56H	space	20H
2	32H	C	43H	M	4DH	W	57H	CR	0DH
3	33H	D	44H	N	4EH	X	58H		
4	34H	E	45H	O	4FH	Y	59H		
5	35H	F	46H	P	50H	Z	5AH		
6	36H	G	47H	Q	51H	<	3CH		
7	37H	H	48H	R	52H	=	3DH		
8	38H	I	49H	S	53H	>	3EH		
9	39H	J	4AH	T	54H	+	2BH		

10.2.3 Data storage

Store the sequence program, which has been converted into the data format, in the memory as shown below:



CAUTION

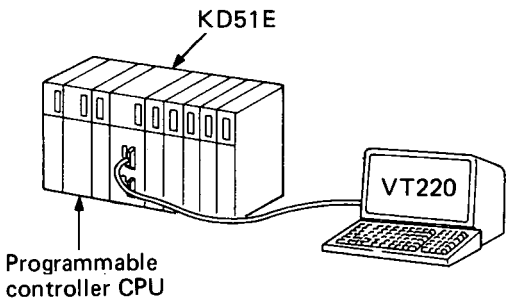
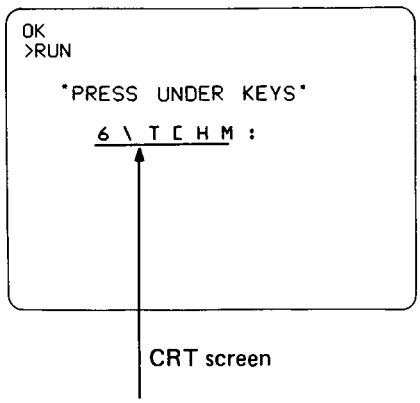
1. As the data storage area, only 6000H ~ 67FFH of common area can be used.
2. In the case of read from the programmable controller (SRI), also store the read data in 6000H ~ 67FFH of common area.

MEMO

11. PROGRAM EXAMPLES

11. PROGRAM EXAMPLES

11.1 Program for Erasing Display on CRT of VT220 by Pressing Keys

System Configuration	Program Run Screen
KD51E + VT220  	R U N RETURN  By pressing keys, which correspond to the above characters, in order of display, keyed-in characters are erased.

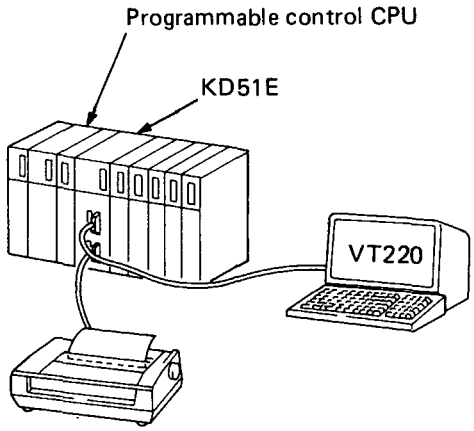
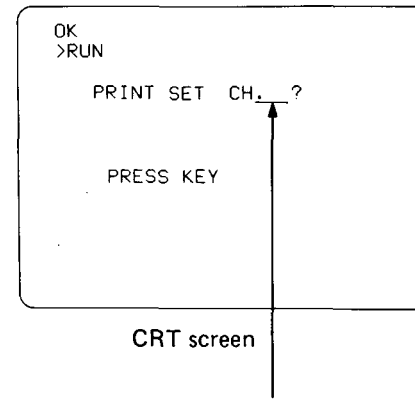
BASIC programming

```
OK
>LIST
   100 CLS
   110 ZIDV 0 ····The keyboard, which is connected to channel 0 of RS-232-C, can be used as an input console.(Setting is not required only for the channel 0.)
   120 LOCATE 10,20;"PRESS UNDER KEYS"
   130 LOCATE 12,21;PRINT"6 \ T [ H M :"  ····· Characters enclosed in " " are displayed on the screen.
   140 I=1
   150 A=INKEY
   160 IF (I=1)&(A=$36) GOTO 240
   170 IF (I=2)&(A=$5C) GOTO 240
   180 IF (I=3)&(A=$54) GOTO 240
   190 IF (I=4)&(A=$5B) GOTO 240
   200 IF (I=5)&(A=$48) GOTO 240
   210 IF (I=6)&(A=$4D) GOTO 240
   220 IF (I=7)&(A=$3A) GOTO 240
   230 GOTO 150
   240 LOCATE 14,15+I*2;PRINT "     "
   250 LOCATE 14,17+I*2;PRINT $A
   260 LOCATE 12,19+I*2;PRINT "  "
   270 I=I+1
   280 IF I#8 GOTO 150
   290 END
```

····· Keyed-in characters are judged.

Keyed-in Character	ASCII Code
6	$36
\	$5C
T	$54
[	$5B
H	$48
M	$4D
:	$3A

11.2 Program for Printing-out Characters Corresponding to Pressed Keys of VT220

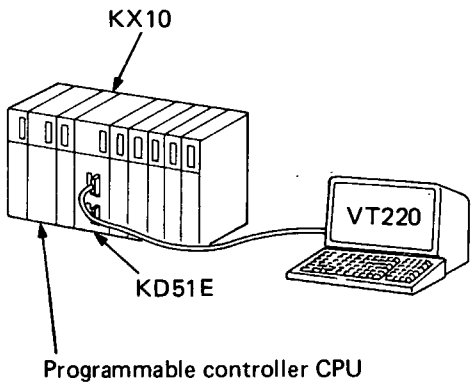
System Configuration	Program Run Screen
KD51E + VT220 + KS6PR (or General-purpose printer)  	[R] [U] [N] [RETURN]  Press the key corresponding to the number of printer connection channel. Then, comment shown below is displayed. Thereafter, keyed-in characters are printed out by the printer.

BASIC programming

```
OK
>LIST
   100 CLS
   110 LOCATE 8,23
   120 PRINT "PRINT SET CH. ?"        ..... "PRINT SET CH?" is displayed on the screen.
   130 A=INKEY                        ..... Key in the print set CH through the keyboard.
   140 IF A=$30 GOTO 130
   150 IF A=$31 GOTO 180              ..... The print set CH is discriminated.
   160 IF A=$32 GOTO 180                    (Only CH1 or 2 is effective.)
   170 IF A>$32 GOTO 130
   180 LOCATE 8,23:PRINT "PRESS KEY"  ..... "PRESS KEY" is displayed on the screen.
   190 A=INKEY                        ..... Press the keys which correspond to characters printed by the printer.
   200 LPRINT *A                      ..... Characters, which have been keyed-in in line number 190, are printed
   210 GOTO 190                             by the printer.
```

11.3 Program for Displaying ON/OFF Data of I/O Card

System Configuration	Program Run Screen
KCPU + I/O Card + KD51E + VT220	[R] [U] [N] [RETURN]




```
OK
>RUN

   Y 0  ON
   Y 1  OFF
   Y 2  OFF
   Y 3  ON
   Y 4  ON
   Y 5  OFF
```

CRT screen

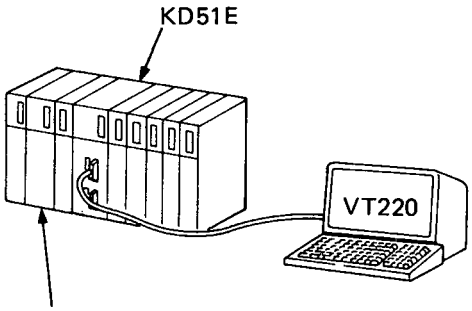
ON/OFF states of Y00 ~ Y05 during program run are displayed.

BASIC programming

```
OK
>LIST
   100 CLS
   110 A=0
   120 B=8
   130 LOCATE B,20;PRINT 'Y',#2,A
   140 A=A+1
   150 B=B+1
   160 IF A#6 GOTO 130
   170 C=$E400
   180 D=8
   190 F=0
   200 E=ZRD1($9,C,X)
   210 X=X&1
   220 IF X=0 LOCATE D,25;PRINT'OFF';GOTO 240
   230 LOCATE D,25;PRINT'ON'
   240 D=D+1
   250 C=C+1
   260 F=F+1
   270 IF F#6 GOTO 200
   280 END
```

- 130: Y0 ~ Y5 are displayed on the screen.
- 170: The address of Y0 is set.
- 180: Indicates the Y-axis coordinate of ON/OFF display.
- 200: The state of Y is read.
- 210: The first bit (ON/OFF data) of Y is made effective.
- 220: When the first bit of Y is "0", OFF is displayed. When it is "1", ON is displayed.
- 250: By adding "1" to the address of Y, the read data of line number 200 are changed in due order, beginning with Y0.

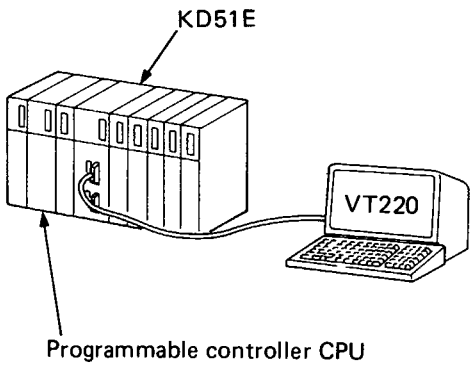
11.4 Program for Displaying RUN/STOP State of KCPU

System Configuration	Program Run Screen
KCPU + KD51E + VT220  Programmable controller CPU	R U N RETURN OK >RUN KCPU RUN/STOP DISPLAY KCPU STOP CRT screen Run/stop state of KCPU during program run is displayed. In this example, the state is stop. (Other display examples) KCPU RUN/STOP DISPLAY KCPU RUN — KCPU RUN state KCPU RUN/STOP DISPLAY KCPU COMMUNICATION ERROR — error of communication between KD51E and KCPU

BASIC programming

```
OK
>LIST
   100 CLS
   110 LOCATE 8,20;PRINT"KCPU RUN/STOP DISPLAY"      "KCPU RUN/STOP DISPLAY" is displayed on the screen.
   120 A=CALL(&H0,&H8030)                            System subroutine (SKC) is called.
   130 IF A=0 GOTO 170                               Run or step of KCPU is discriminated. (A=0: OFF, A=1: ON)
   140 IF A=1 GOTO 190
   150 LOCATE 10,22;PRINT"KCPU COMMUNICATION ERROR"  When A is not 0 or 1, "KCPU COMMUNICATION ERROR" is
   160 END                                           displayed on the screen.
   170 LOCATE 10,22;PRINT"KCPU STOP"                 "KCPU STOP" is displayed on the screen.
   180 END
   190 LOCATE 10,22;PRINT"KCPU RUN"                  "KCPU RUN" is displayed on the screen.
   200 END
```

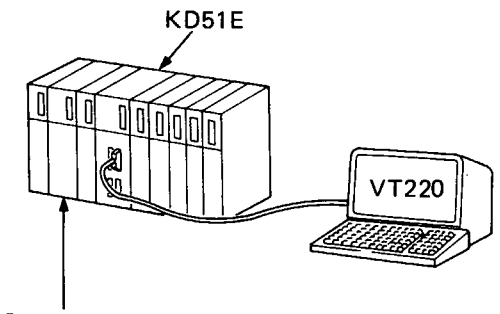
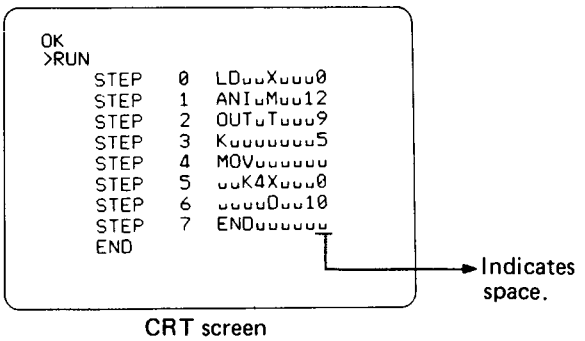
11.5 Program for Writing Sequence Program in Ladder Mode (Not allowed during KCPU run)

System Configuration	Program Run Screen
KCPU + KD51E + VT220  	[R] [U] [N] [RETURN] OK >RUN 0LINE G?M225DJK001 % 1LINE G◉M000DAT000DS0010 % 12LINE % END CRT screen Key-in characters of data format in due order beginning with line 0. (Key-in characters according to the data format in page 70.) When "END" is displayed, program has been written to KCPU. The contents of KCPU can be checked by PU or GPP.

BASIC programming

```
OK
>LIST
   100 A=$6000 ..... Indicates the head address which stores data format written to KCPU in ladder mode by system subroutine (SRK) of line number 320.
   110 B=$F000
   120 B(0)=$6000 ..... Indicates initial setting of system subroutine (SRK) of line number 320.
   130 B(2)=0
   140 Z=$20
   150 PRINT *Z, ..... The heads of characters in "LINE 0" and "LINE 1" are aligned.
   160 C=0 ..... Indicates line numbers beginning with 0.
   170 PRINT C,'LINE␣␣␣' ..... Line numbers, beginning with 0, and "LINE" are displayed on the screen.
   180 D=0
   190 E=0
   200 G=INKEY ..... Key-in data through input console.
   210 A:D)=G ..... Data, which have been keyed in through input console, are stored in address 6000H and thereafter.
   220 E=E+1
   230 IF(A:D-1)=$0D)&(A:D)#$25) C=C+1;PRINT C,'LINE␣␣␣',
   240 PRINT *G,'␣',                    } Data format in ladder mode is marked off line by line and displayed on the screen. (␣ indicates a blank.)
   250 IF $25=A:D-1) GOTO 280 ..... Completion of one circuit block is judged.
   260 D=D+1
   270 GOTO 200
   280 C=C+1
   290 D=D+1
   300 B(1)=E
   310 H=CALL(0,$8039,0) ..... Access time is set to "0" by system subroutine (SKI).
   320 I=CALL(0,$8027,Z,$F000) ..... One circuit block is written to KCPU by system subroutine (SRK).
   330 B(2)=B(3) ..... The head step number of next circuit block, which has been stored in B (3) as the SRK execution result, is transferred to B (2).
   340 IF A:0)#$25 GOTO 170
   350 PRINT'END'
   360 END
```

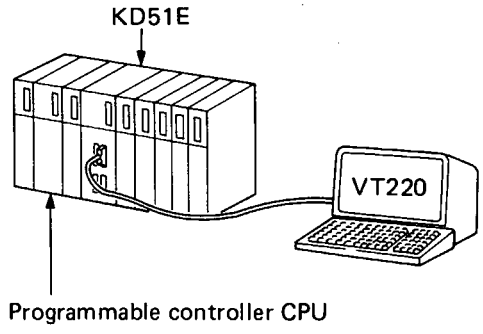
11.6 Program for Writing Sequence Program in List Mode

System Configuration	Program Run Screen
KD51E + VT220 + KCPU  	[R] [U] [N] [RETURN]  CRT screen Key in characters in list mode in due order beginning with step 0. The ␣ mark indicates space. After "END" is keyed in, "END" is displayed. This means the completion of write to KCPU.

BASIC programming

```
OK
>LIST
   100 A=$6000 ..... Indicates the head address which stores data format written to KCPU in list mode by system subroutine (SWI) of line number 270.
   110 B=$F000
   120 B(0)=$6000 ..... Indicates initial setting of system subroutine (SWI) of line number 270.
   130 B(1)=0
   140 C=0
   150 D=0
   160 PRINT"STEP",D," ..... "STEP" and "STEP NUMBER" are displayed on the screen.
   170 FOR E=0 TO 9
   180 F=INKEY
   190 A:C)=F ..... Data format in list mode, which have been keyed-in in line number 180, are displayed on the screen and stored in address 6000H and thereafter.
   200 PRINT *F,
   210 C=C+1
   220 NEXT E
   230 D=D+1
   240 B(2)=D
   250 IF $45#A:C-10) GOTO 160 ..... Check is made if END instruction of sequence program has been keyed in.
   260 G=CALL(0,$8039,0) ..... Access time is set to "0" by system subroutine (SKI).
   270 H=CALL(0,$802D,2,$F000) ..... When END instruction of sequence program is keyed-in, sequence program is written to programmable controller by system subroutine SWI.
   275 IF O#H PRINT"ERROR";GOTO 100
   280 PRINT "END" ..... When the write of sequence program to programmable controller has been completed, "END" is displayed on the screen.
   290 END
```

11.7 Program for Reading Sequence Program in Ladder Mode

System Configuration	Program Run Screen
KD51E + VT220 + KCPU  	R U N RETURN OK >RUN 0LINE G?M250DJK001 % 1LINE G@MOOODATOOODSOO10 % 12LINE % END CRT screen

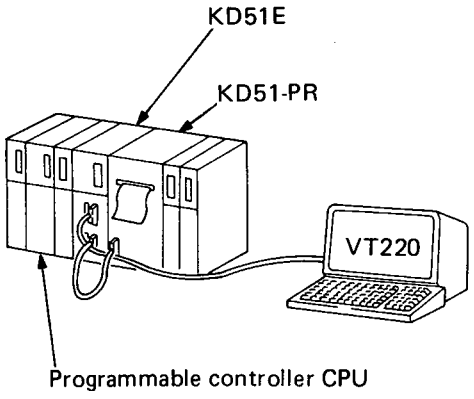
BASIC programming

```
OK
>LIST
   100 A=$F000 ..... Specifies the head address which stores keyed-in data of line numbers 110 ~ 130.
   110 INPUT "DATE STORGE HEAD ADDRESS=",A(0)
   120 INPUT "DATE STORGE BYTE LENGTH =",A(1)  ..... Keyed-in data are displayed on the screen.
   130 INPUT "READING LADDER HEAD STEP=",A(2)
   140 B=0
   150 G=$A000 ..... Specifies the transfer destination address of line number 200 to variable G.
   160 C=A(0)
   170 D=CALL(0,$8039,0) ..... Access time is set to "0" by system subroutine SKI.
                              (Access time is set to "0" only when programmable controller CPU is at stop.)
   180 E=CALL(0,$8024,2,$F000) ..... Sequence program is read in ladder mode and stored in addresses of data of line numbers
   190 F=A(4)+C-1                    110 ~ 130.
   200 H=ZMOV($03,C,F,G) ..... Data in address specified by C of channel 0 and thereafter are transferred to address
   210 G=G+A(4)                    specified by G of channel 3 and thereafter.
   220 I=0
   230 PRINT B,"LINE     ", ..... Line numbers and "LINE" are displayed on the screen.
   240 J=A(0)
   250 K=J:I)
   260 PRINT *K," ",                When % ($25) is entered, sequence program is displayed on the screen and line number
   270 IF $25=J:I) PRINT;GOTO 310 ..... 310 is executed.
   280 IF ($0D=J:I))&($25#J:I+1) PRINT;B=B+1;PRINT B,"LINE ",
   290 I=I+1
   300 GOTO 250
   310 A(2)=A(3)
   320 B=B+1
   330 C=F+1
   340 IF $25#J:0) GOTO 170
   350 PRINT "END"
   360 END
```

CAUTION

Only 6000H ~ 67FFH can be used for data storage areas in line numbers 110 and 120.

11.8 Program for Reading Sequence Program in List Mode

System Configuration	Print-out during Program Run
KD51E + KD51PR + VT220 + KCPU  	STEP 0 LD X 0 STEP 1 ANI M 12 STEP 2 OUT T 9 STEP 3 K 5 STEP 4 MOV STEP 5 K4X 0 STEP 6 D 10 STEP 7 END END

BASIC programming

```
OK
>LIST
   100 A=$F000 ..... Specifies the head address which stores keyed-in data of line numbers 110 ~ 130.
   110 INPUT "DATE STORGE HEAD ADDRESS=",A(0)
   120 INPUT "READ LADDER HEAD STEP   =",A(1) ..... Keyed-in data are displayed on the screen.
   130 INPUT "READ LADDER STEP NUMBER =",A(2)
   140 B=0
   150 C=0
   160 D=0 ..... Indicates step numbers.
   170 E=A(0)
   180 F=CALL(0,$8039,0) ..... Access time is set to "0" by system subroutine SKI.
                              (Access time is set to "0" only when programmable controller CPU is at stop.)
   190 G=CALL(0,$802A,2,$F000) ..... Sequence program is read in list mode and stored in addresses of data of line numbers
   200 H=A(0)                       110 ~ 130.
   210 I=A(2)*10+A(0)-1
   220 H=ZMOV($03,H,I,$A000) ..... Data in address specified by H of channel 0 and thereafter are transferred to address
   230 B=B+1                       specified by A000 of channel 3 and thereafter.
   240 LPRINT "STEP" ,#4,C,"    ", ..... "STEP" and step numbers are printed by KD51E.
   250 FOR K=0 TO 9                  ↳Carriage return inhibit
   260 LPRINT *E:D), ..... Read sequence program in list mode is printed by KD51PR.
   270 D=D+1 ..... "1" is added to line to be printed.
   280 NEXT K
   285 LPRINT *($03) ..... Printing command code 03H is sent.
   290 C=C+1
   300 IF E:D-10)=$45 GOTO 320 ..... When "E" of END instruction is detected, line number 320 is executed.
   310 IF A(2)#B GOTO 200 ..... When printing of one line is completed, printing is executed beginning with line number 200.
   320 LPRINT "END" ..... After END instruction is printed, "END" is printed on KD51E.
   325 LPRINT *($03) ..... Printing command code 03H is sent.
   330 END
```

CAUTION

Only 6000H ~ 67FFH can be used for data storage areas in line numbers 110 and 120.

MEMO

12. ERROR MESSAGES

12. ERROR MESSAGES

12.1 Error Messages Displayed during Mode Selecting Operation

Errors displayed on the mode select menu screen are as shown in the following table:

Error Meassage	Display Screen	Display Condition	Corrective Action
CANNOT SET	Mode select menu screen	Keyed-in set value is not 1 ~ 4	Set once again.
MEMORY PROTECT ERROR		Memory protect has been set in system data area.	Move memory protect switch SW9 to OFF position.
ERROR	Multi task setting screen	Multi task setting data has error.	Set data once again.
CANNOT SET	BASIC program data setting screen	(1) Input data of programming mode is not 0 ~ 4 or F. (2) Input data of programming number is not 0 ~ 8.	Set data once again.
DATA ■ SET ERROR!		Setting of BASIC program data is wrong.	
CHANNEL SET ERROR	K6PRT operation screen	Channel number set by K6PRT is wrong.	Set the channel once again on K6PRT.

Table 12.1 Error Messages Displayed during Mode Select Operation

12.2 Error Messages Displayed during Multi Task Run

12.2.1 Error messages displayed on the screen

The following error messages are displayed during BASIC programming or multi task run. When the following error has occurred during multi task run, error message is displayed on the output console screen of 0CH.

Error Measage	Display Condition	Corrective Action
STACK ERROR! KD51 STOP!	Stack has been used exceeding the stack area set on the system side.	In BASIC, reduce GOSUB or FOR ~ NEXT statement to tenfold or less per task.
BTWF ERROR! KD51 STOP!	Contents of RAM for task schedule on system side have been rewritten.	Check if memory on system side has been accessed by user program by mistake.
WAIT ERROR! KD51 STOP!		
KD51 STOP! TASK NO.	There is statement of BASIC which cannot be interpreted by interpreter during run of multi task.	Correct BASIC program.
STOP COMMAND KD51 STOP! TASK NO.	STOP command of BASIC has been executed during run of multi task.	Remove STOP command or change it to END, GOTO, GOSUB, RETURN, ONGOTO, or ONGOSUB command.
BREAK COMMAND KD51 STOP! TASK NO.	BREAK command of BASIC has been executed during run of multi task.	Remove BREAK command.
TEXT END KD51 STOP! TASK NO.	END, GOTO, GOSUB, ONGOTO, ONGOSUB, or RETURN command is not provided at the end of BASIC program.	Correct program.
WHAT	Grammatical error has been detected in BASIC program (CAUTION 1)	Correct program.
HOW		
SORRY	Program area is insufficient.	Expand program area.
ROM OR MEMORY PROTECT AREA! PLEASE DO NOT CORRECT PROGRAM	Program area is ROM area or area where memory protect is set.	Message for alarm (CAUTION 2)

Table 12.2 Error Messages Displayed during Run of Multi Task

CAUTION

1. "WHAT" or "HOW" is displayed in the following cases:

   (1) An undefined command has been used.
   (2) The description format of command has error.
   (3) A line number is not specified for the GOTO, GOSUB, ONGOTO or ONGOSUB command.
   (4) The RETURN command has been detected although the GOSUB or ONGOSUB command is not used.

2. Since the program setting area is the ROM area or memory protect setting area, never correct the program. If the program is corrected, the contents of memory area, which stores the BASIC program data, will be damaged. When this message is displayed, therefore, usable BASIC commands are only LIST, LLIST and BYE commands.
   In the case of memory protect setting area, the program can be corrected by clearing memory protect.

12.2.2 Error display by indicator

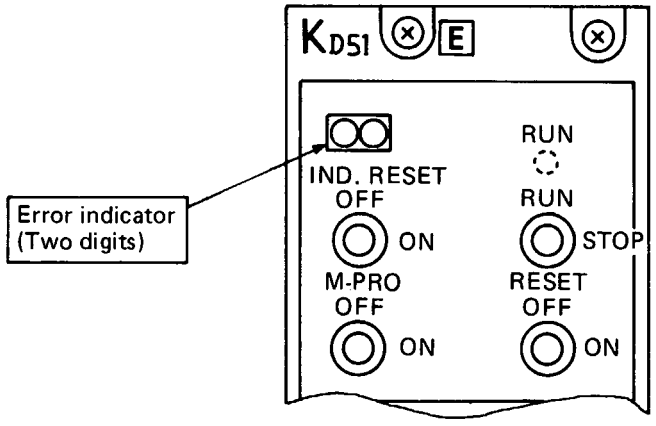


Fig. 12.1 Error Indicator

When error has occurred in the operation or action of KD51E, the "indicator light" shown in Fig. 12.1 displays a two-digit error number. For the contents of errors, see the table 12.3.

Error Number	Error Name	Display Condition	Location of Error	Corrective Action
00	Battery error (CAUTION 1)	Battery is not loaded. Voltage of battery is not proper.	——	Load battery. Change battery with new one.
10	Multi task setting error	Multi task has been run although setting of multi task is wrong.	——	Set multi task once again.
11	BASIC program error	There is grammatical error in BASIC program.	Task 1	Correct program.
12			Task 2	
13			Task 3	
14			Task 4	
15			Task 5	
16			Task 6	
17			Task 7	
18			Task 8	
21	STOP error	STOP command of BASIC has been executed during run of multi task.	Task 1	Remove STOP command or change it to END, GOTO, GOSUB, ONGOTO, ONGOSUB, or RETURN command.
22			Task 2	
23			Task 3	
24			Task 4	
25			Task 5	
26			Task 6	
27			Task 7	
28			Task 8	
31	BREAK error	BREAK command of BASIC has been executed during run of multi task.	Task 1	Remove BREAK command.
32			Task 2	
33			Task 3	
34			Task 4	
35			Task 5	
36			Task 6	
37			Task 7	
38			Task 8	

Table 12.3 Errors Displayed by Error Indicator

Error Number	Error Name	Display Condition	Location of Error	Corrective Action
41	Text end error	END, GOTO, GOSUB, ONGOTO, ONGOSUB, or RETURN command is not provided at the end of BASIC program.	Task 1	Correct program.
42			Task 2	
43			Task 3	
44			Task 4	
45			Task 5	
46			Task 6	
47			Task 7	
48			Task 8	
51	ORST error (CAUTION 1)	Task of which run has not completed, has been started again.	Task 1	Reconsider start condition of task.
52			Task 2	
53			Task 3	
54			Task 4	
55			Task 5	
56			Task 6	
57			Task 7	
58			Task 8	
60	Stack excess error	Stack has been used exceeding stack area set on system side.	——	In BASIC, reduce GOSUB or FOR~NEXT statement to tenfold or less per task. In microcomputer, reduce stack to 128 bytes or less per task.
70	Plural WAIT error BTWF error	Contents of RAM for task schedule on system side have been rewritten.	——	Check if memory on system side has been accessed by user program by mistake.
80	Receive buffer full error (CAUTION 1)	Received data in receive buffer has reached 511 bytes.	RS-232-C CH0	Do not send data of 512 or more bytes at one time.
81			RS-232-C CH1	
82			RS-232-C CH2	
83			RS-422 CH3	
90	Send buffer full error (CAUTION 1)	Sent data in send buffer has reached 127 bytes.	RS-232-C CH0	Check if connection of cable and connector is correct. Vacate corresponding receive buffer.
91			RS-232-C CH1	
92			RS-232-C CH2	
93			RS-422 CH3	
99	Programmable controller CPU error (CAUTION 1)	Programmable controller CPU has been reset during communication with programmable controller CPU or communication has stopped due to WDT error. *Note: "99" may be displayed when instantaneous power failure has occurred. Run of KD51E is not affected by this error.*	——	Run of KD51E is not affected by this error.

Table 12.3 Errors Displayed by Error Indicator (Continued)

CAUTION

1. In the case of "battery error", "ORST error", "receive buffer full error", "send buffer error" or "programmable controller CPU error", the processing of KD51E is continued.

2. The capacity of receive buffer is 511 bytes per channel. Therefore, data of 512 or more bytes are ignored.

3. After the "send buffer full error" message is displayed, data sending is held for at least one minute. When the send buffer is still full after one minute, data is not sent and the next processing is initiated. However, when the system subroutine SWB (block data sending) is called, the time preset on the timer becomes effective.

4. Possible causes of full send buffer are as follows:
   - During DR control, the DTR signal (number 6 pin of RS-232-C) from external equipment is LOW.
   - During XON/XOFF control, the XON code is not sent after the XOFF code is received from external equipment.

MEMO

APPENDIX

13

APPENDIX

1. CAUTIONS DURING PREPARATION OF BASIC PROGRAM

1.1 Initial Setting during BASIC Programming

The following table shows the contents of the initial screen items which are set during BASIC programming.

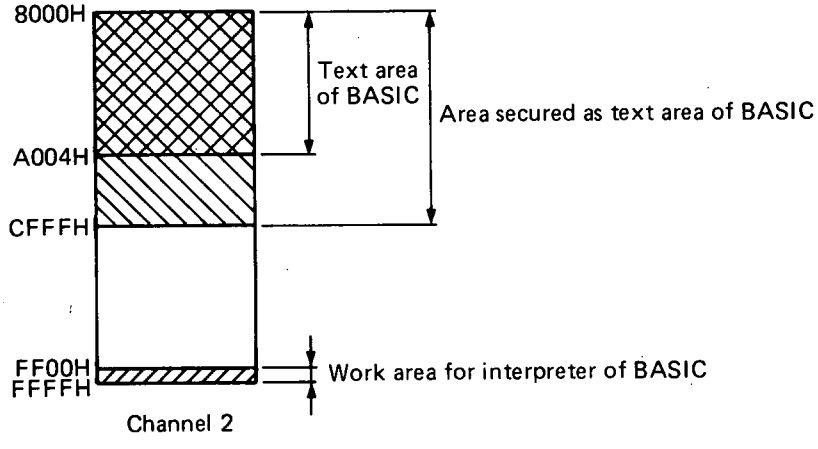
Item	Description
1. PROGRAM HEAD ADDRESS	Head address of BASIC text (Specify address 8000H or address located below 8000H.)
2. PROGRAM LAST ADDRESS	The last address used to secured area as text area of BASIC
3. ADDITIONAL PROGRAM HEAD ADDRESS	Indicates head address when another text of BASIC is inserted. (Automatically set on OS side)
4. WORK AREA HEAD ADDRESS	Work area used by interpreter of BASIC and fixed to 256 bytes. (This is not a work area used by user.)
5. CHANNEL	Channel where text of BASIC is inserted.

Table 1.1

EXAMPLE

When the channel 2 (8000H ~ FFFFH) of KD51E internal memory has been set as shown below, the memory map is as shown below.

1. PROGRAM HEAD ADDRESS	8000H
2. PROGRAM LAST ADDRESS	CFFFH
3. ADDITIONAL PROGRAM HEAD ADDRESS	A005H
4. WORK AREA HEAD ADDRESS	FF00H
5. CHANNEL	2



CAUTION

1. The basic variable is allotted inside the work area for interpreter of BASIC.
2. The vacant area (D000H ~ FEFFH) in the figure of EXAMPLE can be freely used for the @ array variable and indirect variable. Never use the hatched areas.
3. Since the work area for interpreter of BASIC is allotted below the text area and 256 bytes are automatically assigned, be sure to set "00H" to the lower two digits.
4. When preparing two or more tasks in the same channel, the program area and the work area for interpreter should not overlap.

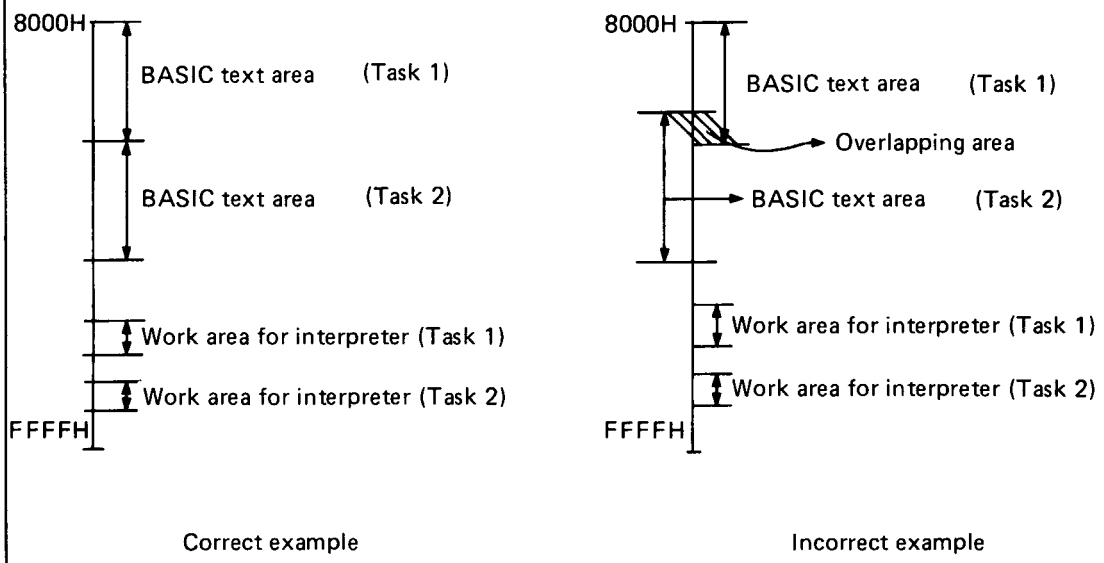


Fig. 1.1 Overlap of Tasks

1.2 Start Condition and Programming

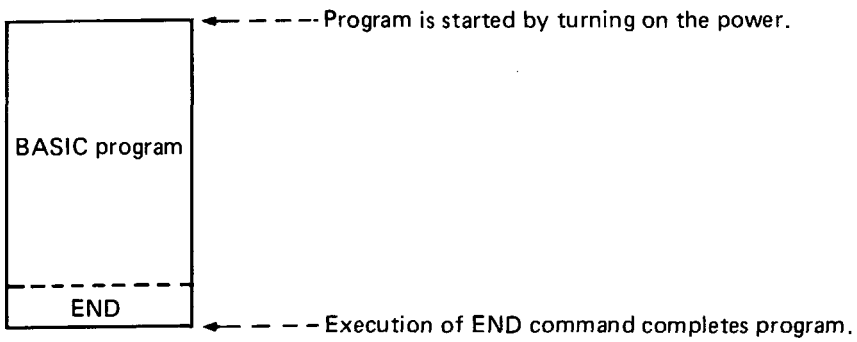
There are the following four types of BASIC program run formats:

1) Program is run only once after power-on.
2) Program is always run after power-on.
3) Program is run by interruption caused by KCPU.
4) Program is run at set intervals of real time.

13

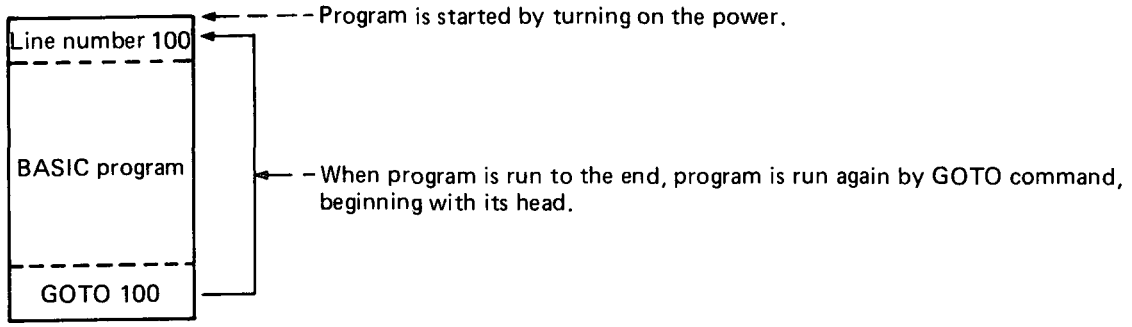
1.2.1 Program is run only once after power-on

Prepare the program so that the END command is executed at the end of program. Select "POWER ON" as the start condition of task.



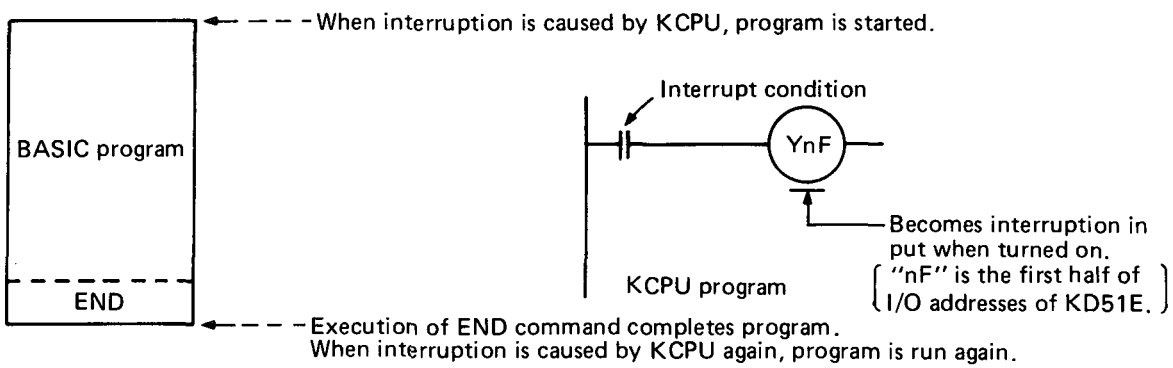
1.2.2 Program is always run after power-on

Prepare the program so that run is returned to the head of program by the GOTO command, without using the END command at the end of program. Select "POWER ON" as the start condition of task.



1.2.3 Program is run by interruption caused by KCPU (See Section 6.5)

Prepare the program so that the END command is executed at the end of program. Select "KCPU INT" as the start condition of task.



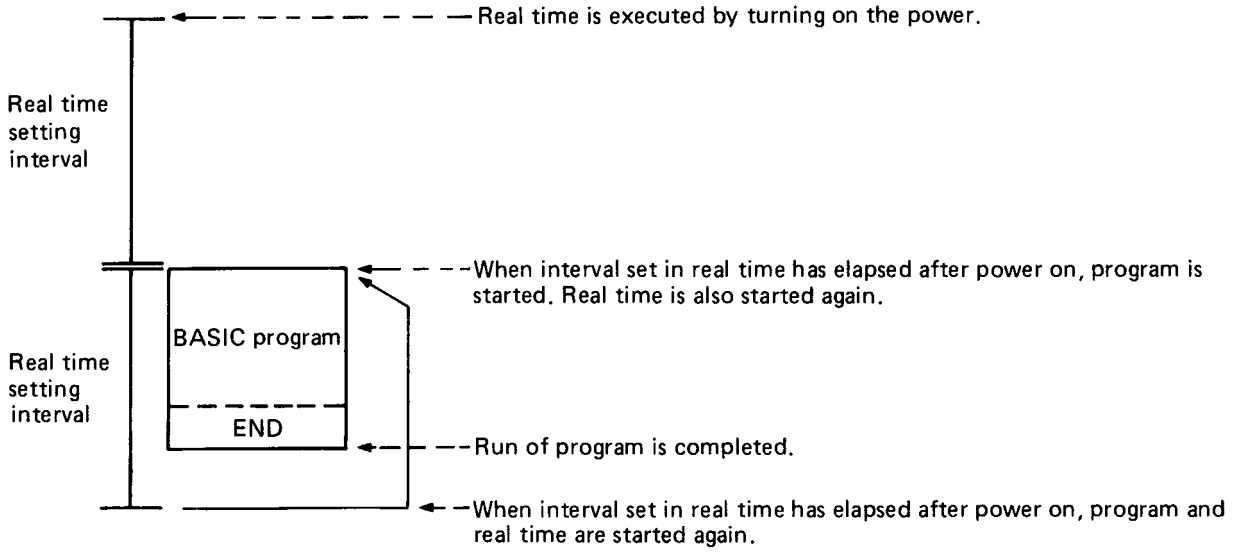
CAUTION

1. The task, which runs the program by interruption caused by the KCPU, should be one. If two or more tasks are provided, "plural ORST occurrence" error may occur.

2. By executing the END command, the task which runs the program by interruption caused by the KCPU resets the interruption caused by the KCPU.

1.2.4 Program is run at set intervals of real time

Prepare the program so that the END command is executed at the end of program. elect "REAL TIME INT" as the start condition of task and set the interval.



CAUTION

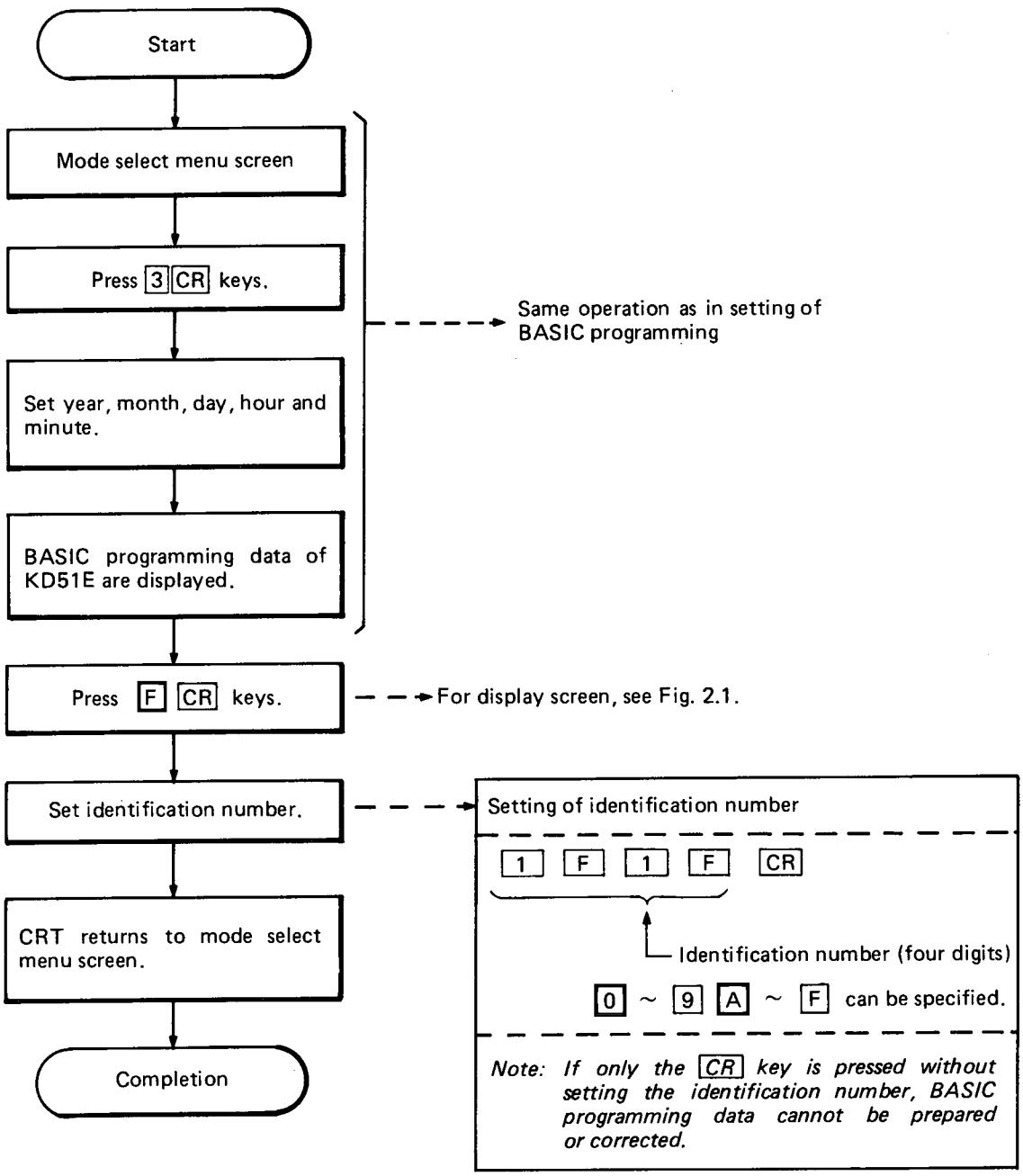
It is required to set the interval of real time greater than the value obtained by adding the run time of all programs which are started by multi task. If the task, for which the start condition "REAL TIME INT" has been selected, does not execute the END command within the set interval, "plural ORST occurrence" error will result.

13

2. MASK OF BASIC PROGRAM

2.1 Mask Method of BASIC Program

The BASIC program can be masked by the following operation.



CAUTION

1. The mask of BASIC program cannot be cleared.
2. The identification number is required to add or correct the BASIC program.
3. Since the identification number cannot be changed, record the number when masking the BASIC program.

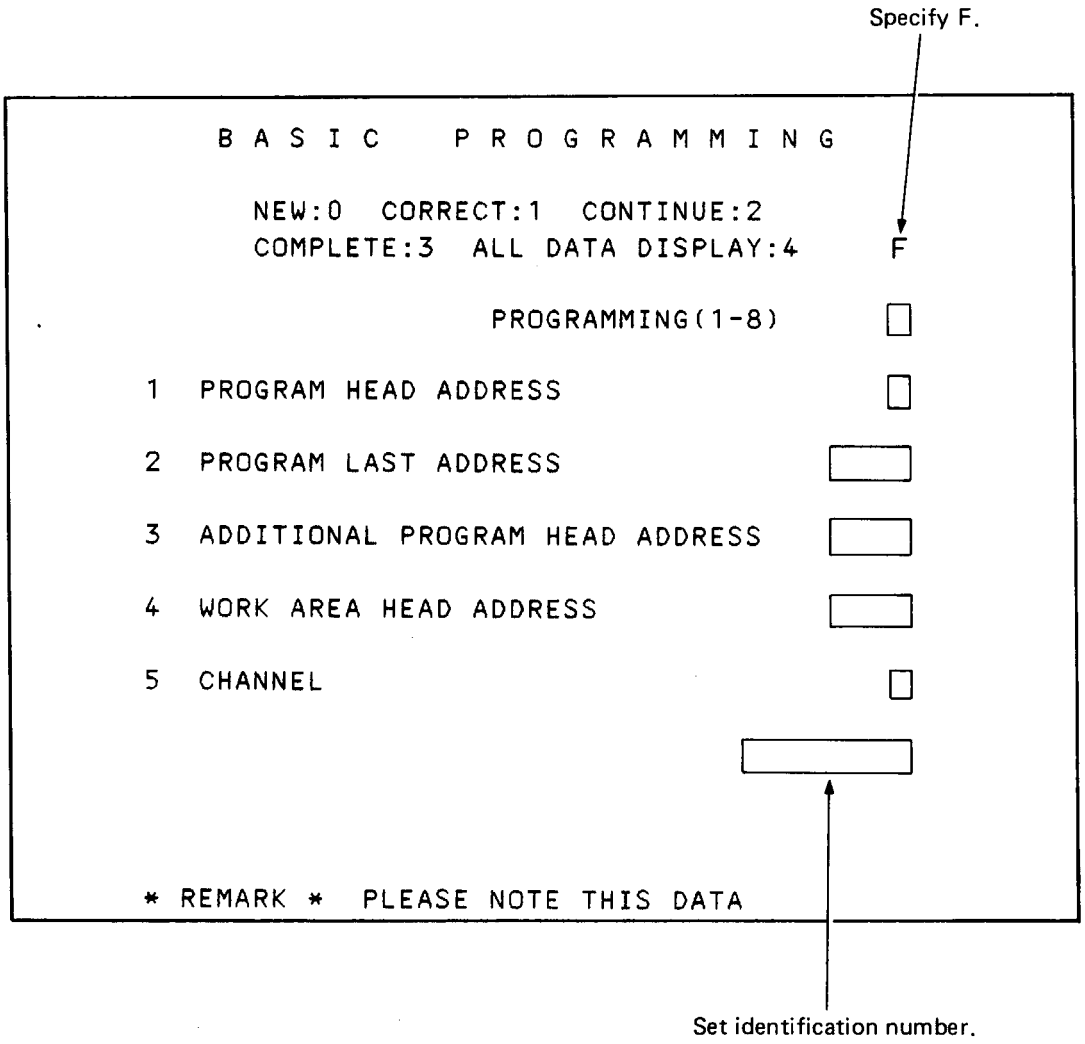
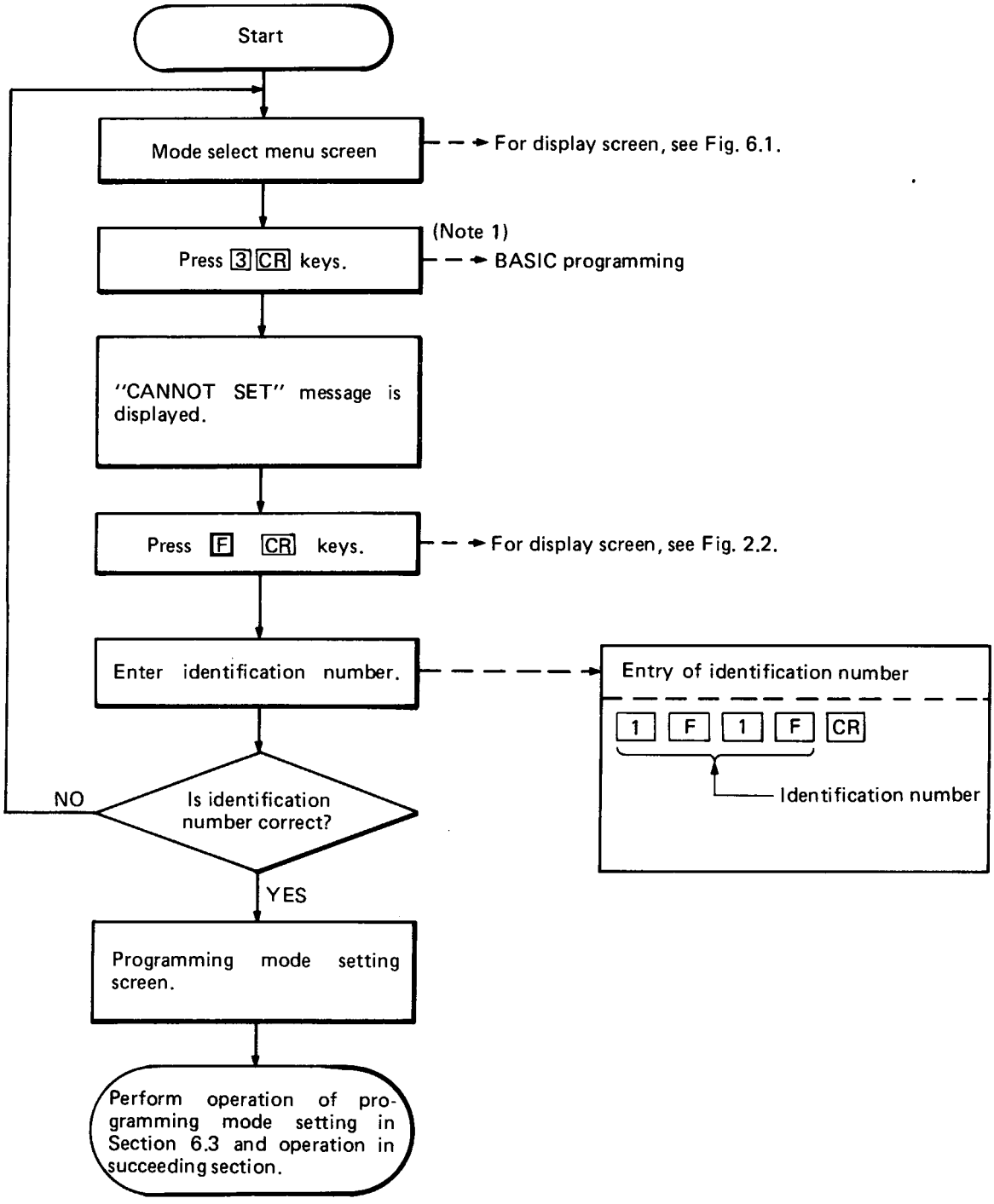


Fig. 2.1 Masking Screen of BASIC Program

13

2.2 Correction of BASIC Program

After masking, the BASIC program can be corrected by the following operation. However, since this is not the clearing operation of mask, the following operation is required to correct the BASIC program.



CAUTION

If the F CR keys are pressed with the mode select menu screen displayed on the CRT, Fig. 2.2 is not displayed.

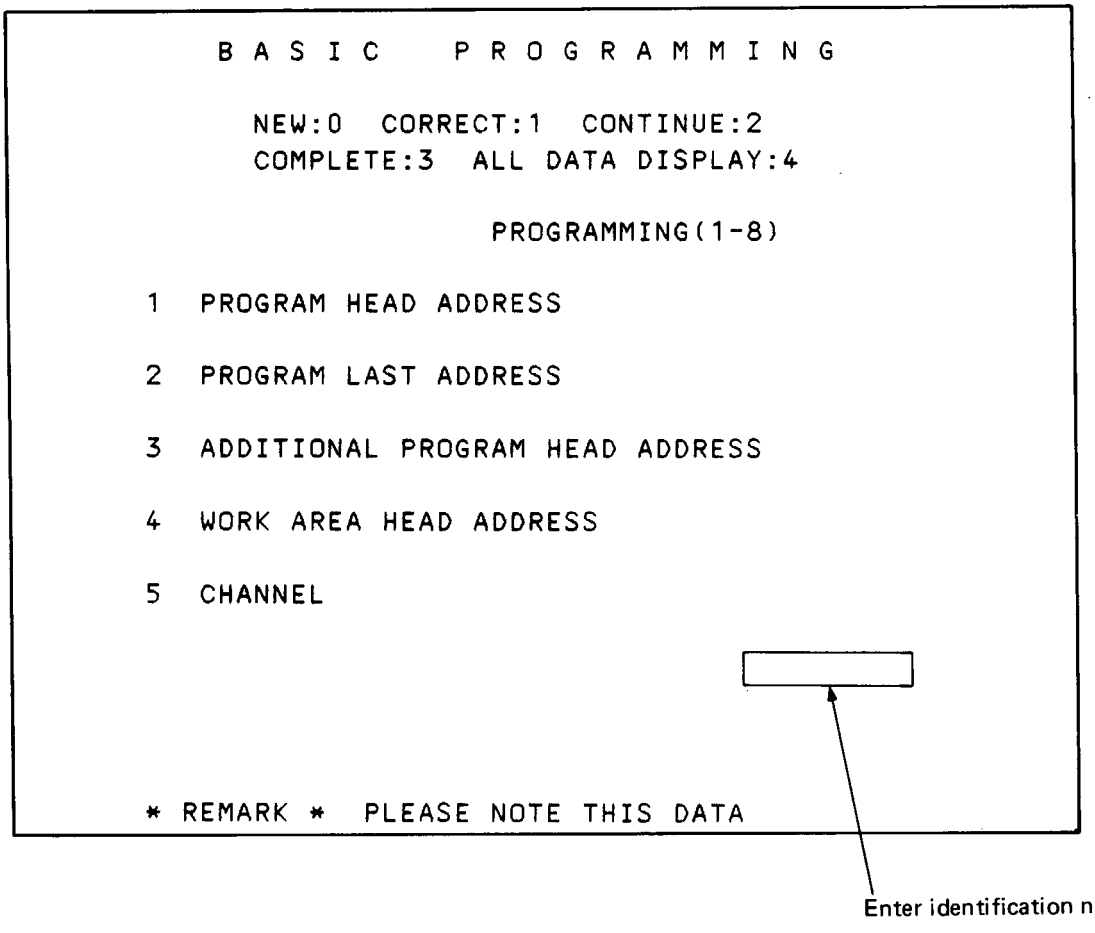


Fig. 2.2 Screen at the Time of BASIC Program Correction

13

3. CAUTIONS FOR REMOTE RUN/STOP OF KCPU

When remote run or remote stop of K3NCPU(P2), K2NCPU or K2HCPU is performed by the KD51E, there are the following differences in the clear function of data (D, M, T, C) inside the KCPU, depending on the type of KCPU.

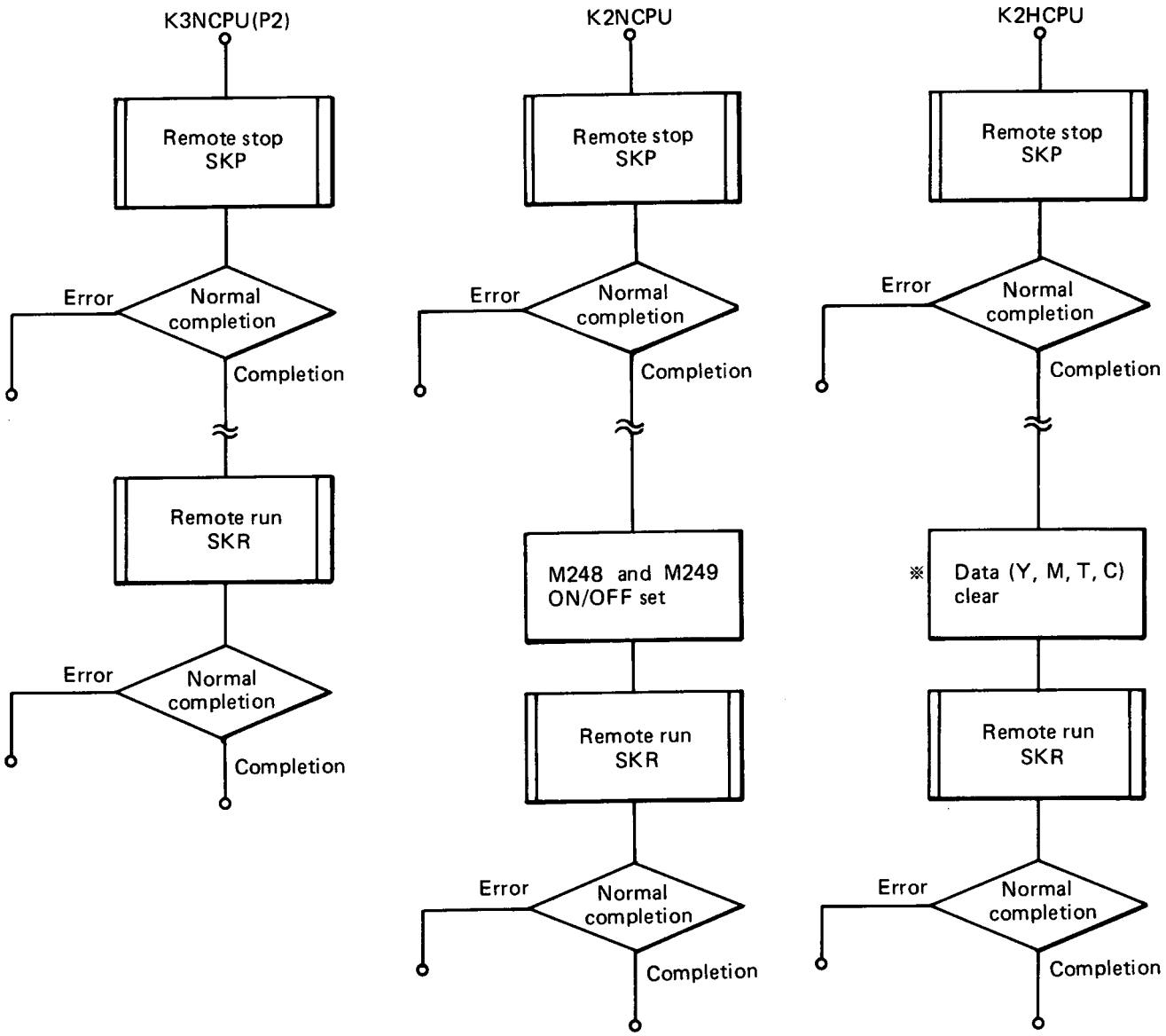
<table>
<tr><th>K3NCPU(P2)</th><th>K2NCPU</th><th>K2HCPU</th></tr>
<tr>
<td>When SKP (KCPU remote stop)※ is called, data are cleared according to the input condition of SKP.

The user can specify one of the following modes as input condition of SKP
{ All data clear
Only unlatched data clear
No clear</td>
<td>When SKR (KCPU remote run)※ is called, data are cleared according to the states of M248 and M249 of K2NCPU.
<table>
<tr><th>M248</th><th>M249</th><th>Data Clear Area</th><th>Output Image Clear</th></tr>
<tr><td>OFF</td><td>OFF</td><td>Not Cleared</td><td>Not cleared</td></tr>
<tr><td>OFF</td><td>ON</td><td>Only unlatched data are cleared</td><td>Cleared</td></tr>
<tr><td>ON</td><td>OFF</td><td>All data cleared</td><td>Cleared</td></tr>
<tr><td>ON</td><td>ON</td><td>All data cleared</td><td>Not cleared</td></tr>
</table>
</td>
<td>When SKP is called and when SKR is called, data are not cleared.</td>
</tr>
</table>

Table 3.1

※: For details of system subroutines, SKP and SKR, see the instruction manual of "GPC-BASIC".

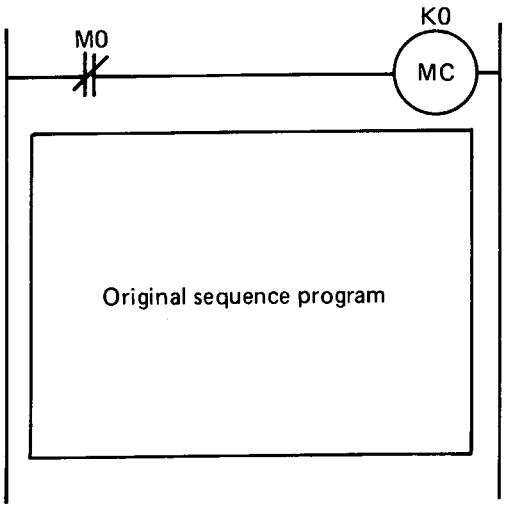
(1) Typical remote run/stop procedure of each KCPU



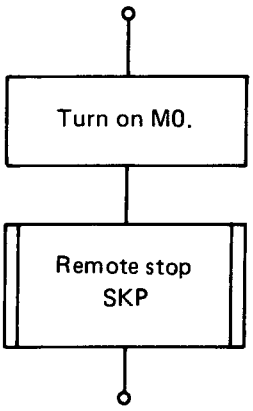
※: If not required, this operation may be omitted.

(2) Since the K3NCPU(P2) or K2HCPU is not provided with a mode which is used to clear the output image during remote run/stop, the final output prior to remote stop is provided again immediately after the execution of remote run.
When it is not desired to provide the final output again, prepare a sequence program as shown in Example 1 and Example 2 so that remote stop is executed after all outputs are turned off.

Example 1

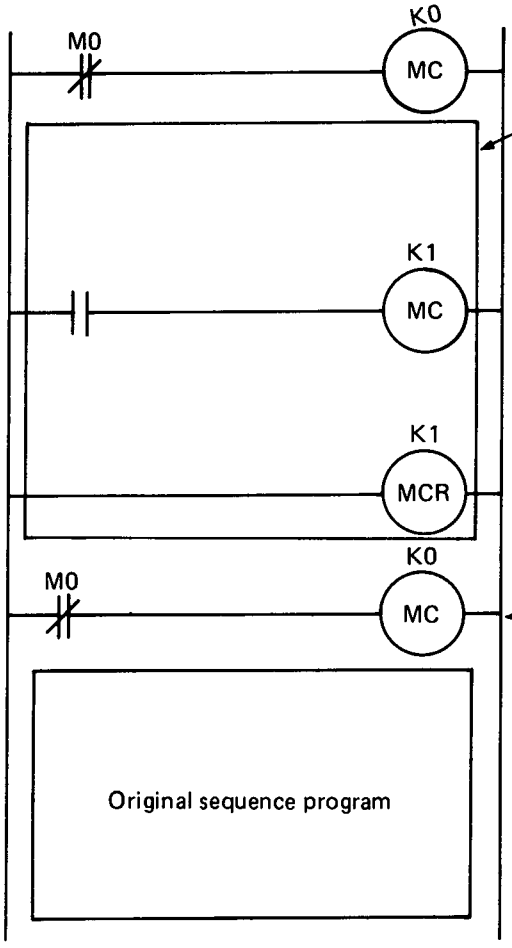


- M0 should be normally off.
- Perform programming in the BASIC program as shown in the following flow chart.

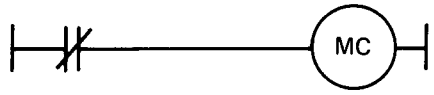


Example 2

Master control is used in the sequence program



- Insert a circuit of



immediately below the master control reset in the sequence program.

- Prepare the BASIC program so that remote stop is executed after M0 is turned on, as in Example 1.

4. CAUTIONS FOR USE OF BASIC COMMANDS

4.1 INKEY Command

When plural tasks are simultaneously set to keying-in operation waiting state, only one task of tasks waiting for keying-in operation returns a key code. In this case, to which task the key code is returned depends on the timing of keying-in operation. Therefore, prevent plural tasks from being simultaneously set to keying-in operation waiting state.

4.2 PRINT and LPRINT Commands for Printer

When the PRINT or LPRINT command is used for the printer, data cannot be printed out if a comma (,), which inhibits carriage return, is provided at the end of a statement.
The printer (K6PRE, K6PR-K, K7PR, etc.) starts printing of data after receiving the CR code (0DH). If a comma (,), which inhibits carriage return, is provided, the KD51E does not sent 0DH. Therefore, data are not printed out.
When KD51PR is used, see Section 10 in APPENDIX..

4.3 CRT Display Commands

If the CRT display command of CLS, ZCON, ZCOFF, ZNOR, ZCRV, PRINT or LOCATE is used in plural tasks for one CRT, a desired display screen may not be obtained. It is recommended to prepare the program for display, which is used for one CRT, by using the CRT display command only in one task.

4.4 OPEN and CLOSE Commands

- RS-232-C-CH1, CH2 and RS-422-CH3 are set to communication disable (send and receive disenable) in initial state, and cannot be used if they are not opened by user program except in the K6PRT mode.
  [However, data can be sent to the channel set in the printer setting screen by use of the LPRINT or LLIST command if the channel is not opened by the user program. ※Only sending]

- The OPEN command is a command for the initialization and initial setting of communication control of each channel of RS-232-C and RS-422 and also for the initialization of send and receive buffers of KD51E. Therefore, when this command is executed, the communication mode is newly set for the specified channel and also the send and receive buffers of specified channel are forcedly vacated.

- The CLOSE command is similar to the OPEN command. The CLOSE command initializes the specified channel and then sets the channel to send/receive disenable state, and at the same time, initializes the send and receive buffers of KD51E. It is not always required to use this command together with the OPEN statement, and since the functions of CLOSE command can be fully covered by the OPEN command, it is not necessarily required to use the CLOSE command.

- The OPEN command and the CLOSE command are commonly used for each task. Therefore, the channel opened by a task, which has been started first, can be used for a task which will be executed later, without opening the channel.

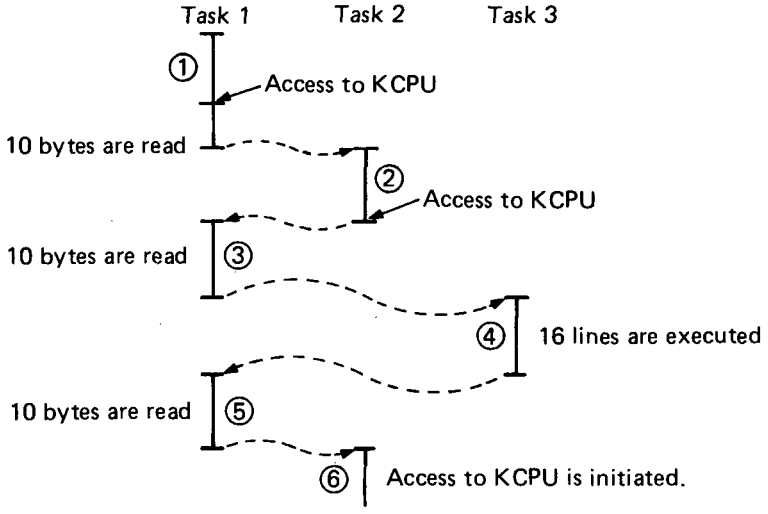
5. CAUTIONS WHEN PLURAL TASKS ACCESS KCPU

When plural tasks attempt to consecutively read or write data of 10 or more bytes from the KCPU by use of command, such as ZMOV, ZRD 1, ZRD 2, ZWR 1 or ZWR2, priority is given to the processing of a task which has a lower number. Until the task with lower number completes access to the KCPU, the task with higher number cannot access to the KCPU. Therefore, it appears that the processing of task with higher number has been suspended.

Example:

There are three tasks and task 1 and task 2 use a KCPU access program . For instance, if data of 30 bytes are read in the access to the KCPU by task 1, program run proceeds as follows:

① Task 1 is executed. The KCPU is accessed, data of 10 bytes are first read, and when the waiting state is set, program run proceeds to task 2.
② Task 2 is executed. However, when the KCPU is accessed, the waiting state is set.
③ Another 10-byte data are read. When the waiting state is set, program run proceeds to task 3.
④ 16 lines of task 3 are executed.
⑤ Another 10-byte data are read. (Read of 30-byte data is completed.)
⑥ Access to KCPU by task 2 is initiated.



6. I/O CONSOLE

For the I/O console of KD51E, the VT220 is recommended.

6.1 VT220 Used for I/O Console

The character codes and display control codes of KD51E are matched to those of VT220.

Key Arrangement of Keyboard

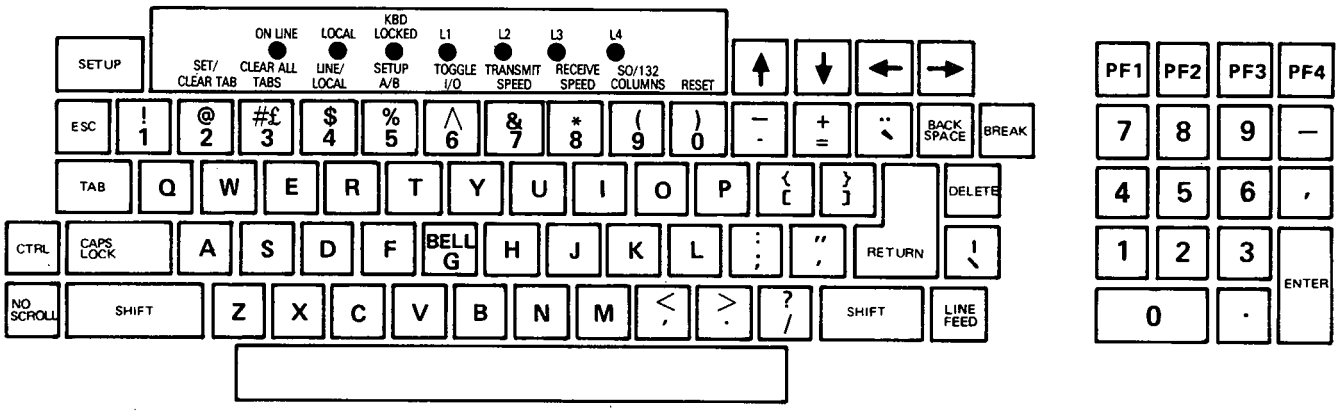


Fig. 3 Key Arrangement of VT220

13

Key Character Code List

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P		p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	£	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	¦
D	CR	GS	–	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	–	o	DEL

← High Rank (HEX)

↑ Low Rank (HEX)

Fig. 4 Sending Character Code List

6.2 Other General-Purpose Terminal Used for I/O Console

When a general-purpose terminal other than VT220 is used, match the display control codes to those of VT220.

Function	Contents	Used Code (ASCII)
Line feed operation	Carriage return operation	CR, LF codes (0DH, 0AH))H, 0AH)
Screen clear	Whole screen clear	ESC + [(5BH) + 2(32H) + J(4AH)
XON operation	Specifies transfer enable from external unit.	DC1 code (11H)
XOFF operation	Specifies transfer disable from external unit.	DC3 code (13H)
Escape operation	Escape sequence introducer	ESC code (1BH)
Cursor addressing	Specifies cursor position absolutely.	ESC + [(5BH) (Note) +line designation code(31H~32H+35H)+;(3BH) +column designation code(31H~38H+31H)+H(48H)
Character qualification	Character reverse display stop	ESC + [(5BH) + (30H) + m(6DH)
	Character reverse display start	ESC + [(5BH) + 7(37H) + m(6DH)
Cursor home	Sets cursor to home position	ESC + [(5BH) + H (48H)
Back space operation	Moves cursor to left side by one column	BS code (08H)

Table 3 Display Control Code List

Note: Code setting example of cursor addressing

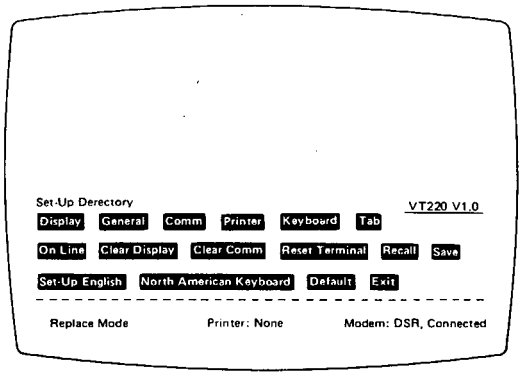
Example:

```
LOCATE  0  ,  23
        ↑     ↑
      Line  Column
        |
        ↓ Code conversion
ESC  +  [  +  1  +  ;  +  2  +  4  +  H
(1BH)   (5BH)  (31H)  (3BH)  (32H)  (34H)  (48H)
```

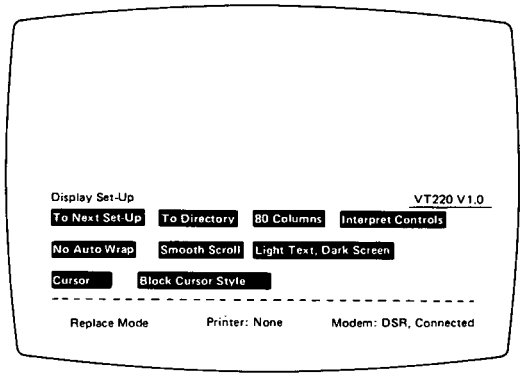
6.3 VT220 Set-Up Derectory

Proceed with the set-up according to the following CRT.

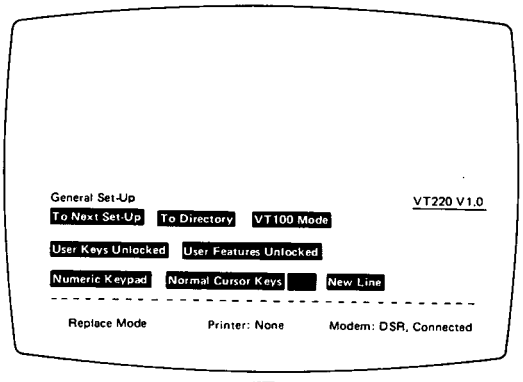
Set-Up Derectory



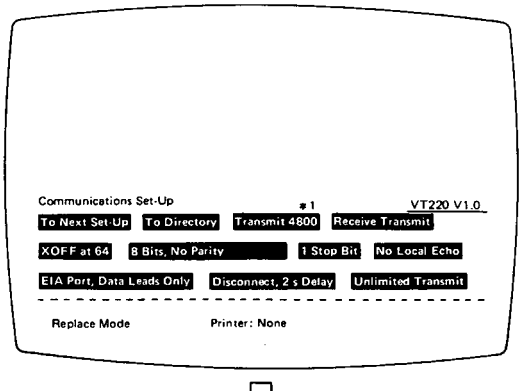
Display Set-Up



General Set-Up

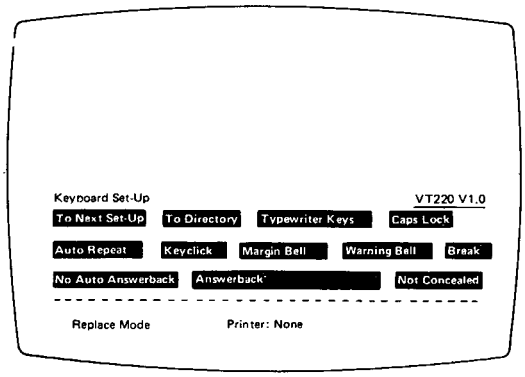


Communications Set-Up

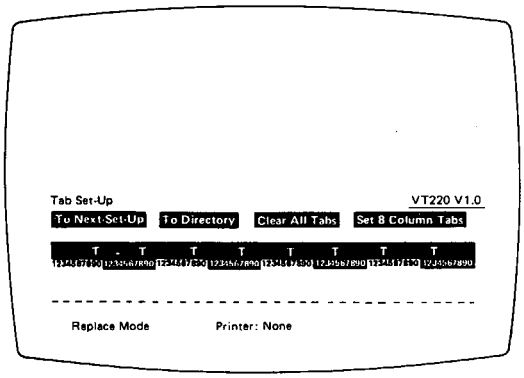


**1: Either 4800 or 9600 may be set.*

Keyboard Set-Up



Tab Set-Up



Change the mode of VT220 from the set-up mode to the online mode.

Reset by use of the RESET switch located at the front of KD51E.

```
                    SET NUMBER
KD51E OPERATING  SYSTEM        V1001
      Y  M  D
DATE  84'11-01  TIME 08:10

      *** MODE SELECT MENU ***

   1. MULTI TASK GO
   2. MULTI TASK SET
   3. BASIC PROGRAMMING
   4. k6PRT OPERATION

                    SET NUMBER
```

7. LISTS OF CHARACTER CODES USED FOR KD51PR, K6PR AND K7PR

Table 4 KD51PR

Line	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	SP	0	@	P	、	p			SP	ー	タ	ミ	年	ℓ
1	!	1	A	Q	a	q			。	ア	チ	ム	月	²
2	"	2	B	R	b	r			「	イ	ツ	メ	日	³
3	#	3	C	S	c	s			」	ウ	テ	モ	円	○
4	$	4	D	T	d	t			、	エ	ト	ヤ	入	●
5	%	5	E	U	e	u			・	オ	ナ	ユ	出	□
6	&	6	F	V	f	v			ヲ	カ	ニ	ヨ	℃	■
7	'	7	G	W	g	w			ァ	キ	ヌ	ラ	Ω	◇
8	(	8	H	X	h	x			ィ	ク	ネ	リ	μ	◆
9	)	9	I	Y	i	y			ゥ	ケ	ノ	ル	Σ	
A	*	:	J	Z	j	z			ェ	コ	ハ	レ	φ	
B	+	;	K	[	k	{			ォ	サ	ヒ	ロ	∞	
C	,	<	L	¥	l	|			ャ	シ	フ	ワ	+	
D	−	=	M	]	m	}			ュ	ス	ヘ	ン	±	
E	.	>	N	^	n	~			ョ	セ	ホ	゛	≠	
F	/	?	O	−	o				ッ	ソ	マ	゜	〒	

Table 5 K6PR

Line	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	SP	0	@	P	、	p	⊣	⌐	SP	ー	タ	ミ	=	×
1	!	1	A	Q	a	q	⊢	⌐	。	ア	チ	ム	⊧	円
2	"	2	B	R	b	r	⊣	▮	「	イ	ツ	メ	⊧	年
3	#	3	C	S	c	s	⊢	▮	」	ウ	テ	モ	⊧	月
4	$	4	D	T	d	t	⊂	▮	、	エ	ト	ヤ	◢	日
5	%	5	E	U	e	u	⊃	▬	・	オ	ナ	ユ	◣	時
6	&	6	F	V	f	v	⊂	▮▬	ヲ	カ	ニ	ヨ	◥	分
7	'	7	G	W	g	w	⊐	⟨	ァ	キ	ヌ	ラ	◤	秒
8	(	8	H	X	h	x	⊣	⟩	ィ	ク	ネ	リ	♠	
9	)	9	I	Y	i	y	⊢	ー	ゥ	ケ	ノ	ル	♥	
A	*	:	J	Z	j	z	⊣	▮	ェ	コ	ハ	レ	◆	
B	+	;	K	[	k	{	⊩	⊢	ォ	サ	ヒ	ロ	♣	
C	,	<	L	＼	l	|	◀	⌐	ャ	シ	フ	ワ	●	
D	−	=	M	]	m	}	▶	⌙	ュ	ス	ヘ	ン	○	
E	.	>	N	^	n	~	▬	→	ョ	セ	ホ	゛	／	
F	/	?	O	−	o		M	ノ	ッ	ソ	マ	゜	＼	

Table 6 K7PR

Line	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	SP	0	@	P	、	p	ー	⊥	SP	ー	タ	ミ	=	×
1	!	1	A	Q	a	q	ー	T	。	ア	チ	ム	⊧	円
2	"	2	B	R	b	r	ー	⊣	「	イ	ツ	メ	⊧	年
3	#	3	C	S	c	s	ー	⊢	」	ウ	テ	モ	⊧	月
4	$	4	D	T	d	t	▬	ー	、	エ	ト	ヤ	◢	日
5	%	5	E	U	e	u	▬	ー	・	オ	ナ	ユ	◣	時
6	&	6	F	V	f	v	■	|	ヲ	カ	ニ	ヨ	◥	分
7	'	7	G	W	g	w	■	|	ァ	キ	ヌ	ラ	◤	秒
8	(	8	H	X	h	x	|	r	ィ	ク	ネ	リ	♠	〒
9	)	9	I	Y	i	y	|	┐	ゥ	ケ	ノ	ル	♥	市
A	*	:	J	Z	j	z	|	⌙	ェ	コ	ハ	レ	◆	区
B	+	;	K	[	k	{	|	⌟	ォ	サ	ヒ	ロ	♣	町
C	,	<	L	¥	l	|	|	⌜	ャ	シ	フ	ワ	●	村
D	−	=	M	]	m	}	|	⌝	ュ	ス	ヘ	ン	○	人
E	.	>	N	^	n	~	■	⌞	ョ	セ	ホ	゛	／	■
F	/	?	O	−	o		+	ノ	ッ	ソ	マ	゜	＼	

8. WIRING INSTRUCTION

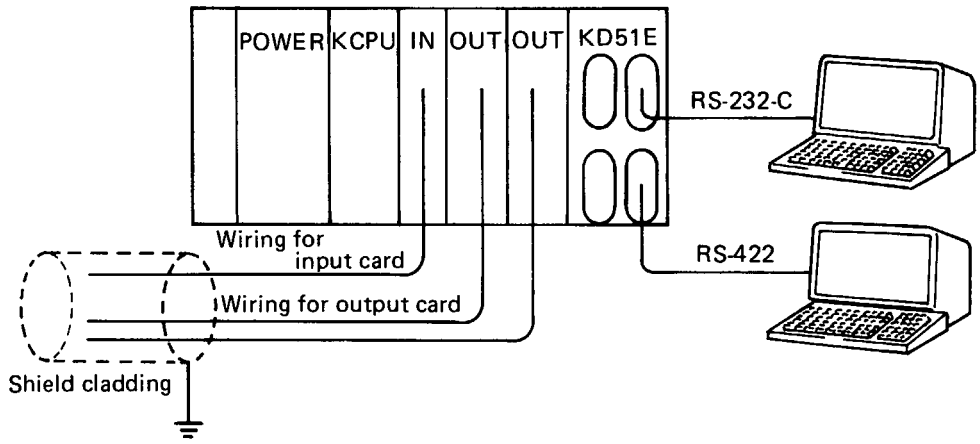


Fig. 8.1

Caution for cable wiring from KD51E interface to connected equipment

Seprate the RS-232-C and RS-422 cables from I/O wiring routes as shown in Fig. 8.1 Noise resistance is considerably influenced depending on connected equipment. It is recommended to use shielded cables.

9. DEVICE ADDRESS TABLE

- For K2NCPU/K2CPU-S3/2KHCPU (page 120 ~ 126)

Process input X000 ~ 1FF

X000 ~ X07F								BYTE	1
	X00	X01	X02	X03	X04	X05	X06	X07	
0	E800	E810	E820	E830	E840	E850	E860	E870	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E808	E818	E828	E838	E848	E858	E868	E878	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X080 ~ X0FF								BYTE	1
	X08	X09	X0A	X0B	X0C	X0D	X0E	X0F	
0	E880	E890	E8A0	E8B0	E8C0	E8D0	E8E0	E8F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E888	E898	E8A8	E8B8	E8C8	E8D8	E8E8	E8F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X100 ~ X17F								BYTE	1
	X10	X11	X12	X13	X14	X15	X16	X17	
0	E900	E910	E920	E930	E940	E950	E960	E970	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E908	E918	E928	E938	E948	E958	E968	E978	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X180 ~ X1FF								BYTE	1
	X18□	X19□	X1A□	X1B□	X1C□	X1D□	X1E□	X1F□	
0	C980	C990	C9A0	C9B0	C9C0	C9D0	C9E0	C9F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	C988	C998	C9A8	C9B8	C9C8	C9D8	C9E8	C9F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Process output Y000 ~ 1FF

Y000 ~ Y07F								BYTE	1
	Y00□	Y01□	Y02□	Y03□	Y04□	Y05□	Y06□	Y07□	
0	E800	E810	E820	E830	E840	E850	E860	E870	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E808	E818	E828	E838	E848	E858	E868	E878	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y080 ~ Y0FF								BYTE	1
	Y08□	Y09□	Y0A□	Y0B□	Y0C□	Y0D□	Y0E□	Y0F□	
0	E880	E890	E8A0	E8B0	E8C0	E8D0	E8E0	E8F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E888	E898	E8A8	E8B8	E8C8	E8D8	E8E8	E8F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y100 ~ Y17F

BYTE 1

	Y10□	Y11□	Y12□	Y13□	Y14□	Y15□	Y16□	Y17□	
0	E500	E510	E520	E530	E540	E550	E560	E570	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E508	E518	E528	E538	E548	E558	E568	E578	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y180 ~ Y1FF

BYTE 1

	Y18□	Y19□	Y1A□	Y1B□	Y1C□	Y1D□	Y1E□	Y1F□	
0	E980	E990	E9A0	E9B0	E9C0	E9D0	E9E0	E9F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E988	E998	E9A8	E9B8	E9C8	E9D8	E9E8	E9F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y000 ~ Y07F

BYTE 1

	Y00□	Y01□	Y02□	Y03□	Y04□	Y05□	Y06□	Y07□	
0	E400	E410	E420	E430	E440	E450	E460	E470	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E408	E418	E428	E438	E448	E458	E468	E478	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y080 ~ Y0FF

BYTE 1

	Y08□	Y09□	Y0A□	Y0B□	Y0C□	Y0D□	Y0E□	Y0F□	
0	D080	D090	D0A0	D0B0	D0C0	D0D0	D0E0	D0F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D088	D098	D0A8	D0B8	D0C8	D0D8	D0E8	D0F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y100 ~ Y17F

BYTE 1

	Y10□	Y11□	Y12□	Y13□	Y14□	Y15□	Y16□	Y17□	
0	D100	D110	D120	D130	D140	D150	D160	D170	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D108	D118	D128	D138	D148	D158	D168	D178	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y180 ~Y1FF

BYTE 1

	Y18□	Y19□	Y1A□	Y1B□	Y1C□	Y1D□	Y1E□	Y1F□	
0	D180	D190	D1A0	D1B0	D1C0	D1D0	D1E0	D1F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D188	D198	D1A8	D1B8	D1C8	D1D8	D1E8	D1F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Temporary memory M000 ~ 255

M000 ~ M089									BYTE	1
	M00	M01	M02	M03	M04	M05	M06	M07	M08	
0	F000	F00A	F014	F01E	F028	F032	F03C	F046	F050	0
1	1	B	5	F	9	3	D	7	1	1
2	2	C	6	20	A	4	E	8	2	2
3	3	D	7	1	B	5	F	9	3	3
4	4	E	8	2	C	6	40	A	4	4
5	F005	F00F	F019	F023	F02D	F037	F041	F04B	F055	5
6	6	10	A	4	E	8	2	C	6	6
7	7	1	B	5	F	9	3	D	7	7
8	8	2	C	6	30	A	4	E	8	8
9	9	3	D	7	1	B	5	F	9	9

M090 ~ M179									BYTE	1
	M09	M10	M11	M12	M13	M14	M15	M16	M17	
0	F05A	F064	F06E	F078	F082	F08C	F096	FOA0	FOAA	0
1	B	5	F	9	3	D	7	1	B	1
2	C	6	70	A	4	E	8	2	C	2
3	D	7	1	B	5	F	9	3	D	3
4	E	68	2	C	6	90	A	4	E	4
5	F05F	F069	F073	F07D	F087	F091	F09B	FOA5	FOAF	5
6	60	A	4	E	8	2	C	6	B0	6
7	1	B	5	F	9	3	D	7	1	7
8	2	C	6	80	A	4	E	8	2	8
9	3	D	7	1	B	5	F	9	3	9

M180 ~ M255									BYTE	1
	M18	M19	M20	M21	M22	M23	M24	M25		
0	FOB4	FOBE	FOC8	FOD2	FODC	FOE6	FOFO	FOFA		0
1	5	F	9	3	D	7	1	B		1
2	6	C0	A	4	E	8	2	C		2
3	7	1	B	5	F	9	3	D		3
4	8	2	C	6	E0	A	4	E		4
5	FOB9	FOC3	FOCD	FOD7	FOE1	FOEB	FOF5	FOFF		5
6	A	4	E	8	2	C	6			6
7	B	5	F	9	3	D	7			7
8	C	6	D0	A	4	E	8			8
9	D	7	1	B	5	F	9			9

Temporary value area of timer/counter T/C000 ~ 127

T/C000 ~ T/C127													BYTE	2
	T/C00	T/C01	T/C02	T/C03	T/C04	T/C05	T/C06	T/C07	T/C08	T/C09	T/C10	T/C11	T/C12	
0	F100	F114	F128	F13C	F150	F164	F178	F18C	F1A0	F1B4	F1C8	F1DC	F1F0	0
	1	5	9	D	1	5	9	D	1	5	9	D	1	
1	2	6	A	E	2	6	A	E	2	6	A	E	2	1
	3	7	B	F	3	7	B	F	3	7	B	F	3	
2	4	8	C	F140	4	8	C	F190	4	8	C	F1E0	4	2
	5	9	D	1	5	9	D	1	5	9	D	1	5	
3	6	A	E	2	6	A	E	2	6	A	E	2	6	3
	7	B	F	3	7	B	F	3	7	B	F	3	7	
4	8	C	F130	4	8	C	F180	4	8	C	F1D0	4	8	4
	9	D	1	5	9	D	1	5	9	D	1	5	9	
5	A	E	2	6	A	E	2	6	A	E	2	6	A	5
	B	F	3	7	B	F	3	7	B	F	3	7	B	
6	C	F120	4	8	C	F170	4	8	C	F1C0	4	8	C	6
	D	1	5	9	D	1	5	9	D	1	5	9	D	
7	E	2	6	A	E	2	6	A	E	2	6	A	E	7
	F	3	7	B	F	3	7	B	F	3	7	B	F	
8	F110	4	8	C	F160	4	8	C	F1B0	4	8	C		8
	1	5	9	D	1	5	9	D	1	5	9	D		
9	2	6	A	E	2	6	A	E	2	6	A	E		9
	3	7	B	F	3	7	B	F	3	7	B	F		

Coil/contact area of timer/counter T/C000 ~ 127

T/C000 ~ T/C089									BYTE	2	
	T/C00	T/C01	T/C02	T/C03	T/C04	T/C05	T/C06	T/C07	T/C08		
0	F400	F40A	F414	F41E	F428	F432	F43C	F446	F450		0
1	1	B	5	F	9	3	D	7	1		1
2	2	C	6	20	A	4	E	8	2		2
3	3	D	7	1	B	5	F	9	3		3
4	4	E	8	2	C	6	40	A	4		4
5	F405	F40F	F419	F423	F42D	F437	F441	F44B	F455		5
6	6	10	A	4	E	8	2	C	6		6
7	7	1	B	5	F	9	3	D	7		7
8	8	2	C	6	30	A	4	E	8		8
9	9	3	D	7	1	B	5	F	9		9

Data register D000 ~ 127

T/C090 ~ T/C127									BYTE	2	
	T/C09	T/C10	T/C11	T/C12							
0	F45A	F464	F46E	F478							0
1	B	5	F	9							1
2	C	6	70	A							2
3	D	7	1	B							3
4	E	8	2	C							4
5	F45F	F469	F473	F47D							5
6	60	A	4	E							6
7	1	B	5	F							7
8	2	C	6								8
9	3	D	7								9

D000 ~ D127													BYTE	2	
	D00	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10	D11	D12		
0	F200	F214	F228	F23C	F250	F264	F278	F28C	F2A0	F2B4	F2C8	F2DC	F2F0		0
	1	5	9	D	1	5	9	D	1	5	9	D	1		
1	2	6	A	E	2	6	A	E	2	6	A	E	2		1
	3	7	B	F	3	7	B	F	3	7	B	F	3		
2	4	8	C	F240	4	8	C	F290	4	8	C	F2E0	4		2
	5	9	D	1	5	9	D	1	5	9	D	1	5		
3	6	A	E	2	6	A	E	2	6	A	E	2	6		3
	7	B	F	3	7	B	F	3	7	B	F	3	7		
4	8	C	F230	4	8	C	F280	4	8	C	F2D0	4	8		4
	9	D	1	5	9	D	1	5	9	D	1	5	9		
5	A	E	2	6	A	E	2	6	A	E	2	6	A		5
	B	F	3	7	B	F	3	7	B	F	3	7	B		
6	C	F220	4	8	C	F270	4	8	C	F2C0	4	8	C		6
	D	1	5	9	D	1	5	9	D	1	5	9	D		
7	E	2	6	A	E	2	6	A	E	2	6	A	E		7
	F	3	7	B	F	3	7	B	F	3	7	B	F		
8	F210	4	8	C	F260	4	8	C	F2B0	4	8	C			8
	1	5	9	D	1	5	9	D	1	5	9	D			
9	2	6	A	E	2	6	A	E	2	6	A	E			9
	3	7	B	F	3	7	B	F	3	7	B	F			

External failure memory F000 ~ 99

F000 ~ F089									BYTE	1	
	F00	F01	F02	F03	F04	F05	F06	F07	F08		
0	DF00	DF0A	DF14	DF1E	DF28	DF32	DF3C	DF46	DF50		0
1	1	B	5	F	9	3	D	7	1		1
2	2	C	6	20	A	4	E	8	2		2
3	3	D	7	1	B	5	F	9	3		3
4	4	E	8	2	C	6	40	A	4		4
5	DF05	DF0F	DF19	DF23	DF2D	DF37	DF41	DF4B	DF55		5
6	6	10	A	4	E	8	2	C	6		6
7	7	1	B	5	F	9	3	D	7		7
8	8	2	C	6	30	A	4	E	8		8
9	9	3	D	7	1	B	5	F	9		9

F090 ~ F127									BYTE	1
	F09□	F10□	F11□	F12□						
0	F35A	F364	F36E	F378						0
1	B	5	F	9						1
2	C	6	70	A						2
3	D	7	1	B						3
4	E	8	2	C						4
5	F35F	F369	F373	F37D						5
6	60	A	4	E						6
7	1	B	5	F						7
8	2	C	6							8
9	3	D	7							9

Master control K000 ~ 063

K000 ~ K063									BYTE	1
	K00□	K01□	K02□	K03□	K04□	K05□	K06□			
0	DFC0	DFCA	DFD4	DFDE	DFE8	DFF2	DFFC			0
1	1	B	5	EF	9	3	D			1
2	2	C	6	0	A	4	E			2
3	3	D	7	1	B	5	F			3
4	4	E	8	2	EC	6				4
5	DFC5	DFCF	DFD9	DFE3	DFED	DFF7				5
6	6	D0	A	4	E	8				6
7	7	1	B	5	F	9				7
8	8	2	C	6	F0	A				8
9	9	3	D	7	1	B				9

- For K3 (Page 127 ~ 145)

Process input X000 ~ 7FF

X000 ~ Y07F								BYTE	1
	X00□	X01□	X02□	X03□	X04□	X05□	X06□	X07□	
0	C800	C810	C820	C830	C840	C850	C860	C870	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	C808	C818	C828	C838	C848	C858	C868	C878	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X080 ~ X0FF								BYTE	1
	X08□	X09□	X0A□	X0B□	X0C□	X0D□	X0E□	X0F□	
0	C880	C890	C8A0	C8B0	C8C0	C8D0	C8E0	C8F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	C888	C898	C8A8	C8B8	C8C8	C8D8	C8E8	C8F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X100 ~ X17F								BYTE	1
	X10□	X11□	X12□	X13□	X14□	X15□	X16□	X17□	
0	C900	C910	C920	C930	C940	C950	C960	C970	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	C908	C918	C928	C938	C948	C958	C968	C978	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X180 ~ X1FF

BYTE 1

	X18□	X19□	X1A□	X1B□	X1C□	X1D□	X1E□	X1F□	
0	E980	E990	E9A0	E9B0	E9C0	E9D0	E9E0	E9F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E988	E998	E9A8	E9B8	E9C8	E9D8	E9E8	E9F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X200 ~ X27F

BYTE 1

	X20□	X21□	X22□	X23□	X24□	X25□	X26□	X27□	
0	CA00	CA10	CA20	CA30	CA40	CA50	CA60	CA70	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CA08	CA18	CA28	CA38	CA48	CA58	CA68	CA78	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X280 ~ X2FF

BYTE 1

	X28□	X29□	X2A□	X2B□	X2C□	X2D□	X2E□	X2F□	
0	CA80	CA90	CAA0	CAB0	CAC0	CAD0	CAE0	CAF0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CA88	CA98	CAA8	CAB8	CAC8	CAD8	CAE8	CAF8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X300 ~ X37F								BYTE	1
	X30□	X31□	X32□	X33□	X34□	X35□	X36□	X37□	
0	CB00	CB10	CB20	CB30	CB40	CB50	CB60	CB70	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CB08	CB18	CB28	CB38	CB48	CB58	CB68	CB78	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X380 ~ X3FF								BYTE	1
	X38□	X39□	X3A□	X3B□	X3C□	X3D□	X3E□	X3F□	
0	CB80	CB90	CBA0	CBB0	CBC0	CBD0	CBE0	CBF0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CB88	CB98	CBA8	CBB8	CBC8	CBD8	CBE8	CBF8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X400 ~ X47F								BYTE	1
	X40□	X41□	X42□	X43□	X44□	X45□	X46□	X47□	
0	CC00	CC10	CC20	CC30	CC40	CC50	CC60	CC70	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CC08	CC18	CC28	CC38	CC48	CC58	CC68	CC78	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X480 ~ X4FF

BYTE 1

	X48□	X49□	X4A□	X4B□	X4C□	X4D□	X4E□	X4F□	
0	CC80	CC90	CCA0	CCB0	CCC0	CCD0	CCE0	CCF0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CC88	CC98	CCA8	CCB8	CCC8	CCD8	CCE8	CCF8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X500 ~ X57F

BYTE 1

	X50□	X51□	X52□	X53□	X54□	X55□	X56□	X57□	
0	CD00	CD10	CD20	CD30	CD40	CD50	CD60	CD70	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CD08	CD18	CD28	CD38	CD48	CD58	CD68	CD78	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X580 ~ X5FF

BYTE 1

	X58□	X59□	X5A□	X5B□	X5C□	X5D□	X5E□	X5F□	
0	CD80	CD90	CDA0	CDB0	CDC0	CDD0	CDE0	CDF0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CD88	CD98	CDA8	CDB8	CDC8	CDD8	CDE8	CDF8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X600 ~ X67F								BYTE	1
	X60□	X61□	X62□	X63□	X64□	X65□	X66□	X67□	
0	CE00	CE10	CE20	CE30	CE40	CE50	CE60	CE70	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CE08	CE18	CE28	CE38	CE48	CE58	CE68	CE78	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X680 ~ X6FF								BYTE	1
	X68□	X69□	X6A□	X6B□	X6C□	X6D□	X6E□	X6F□	
0	CE80	CE90	CEA0	CEB0	CEC0	CED0	CEE0	CEF0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CE88	CE98	CEA8	CEB8	CEC8	CED8	CEE8	CEF8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X700~ X77F								BYTE	1
	X70□	X71□	X72□	X73□	X74□	X75□	X76□	X77□	
0	CF00	CF10	CF20	CF30	CF40	CF50	CF60	CF70	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CF08	CF18	CF28	CF38	CF48	CF58	CF68	CF78	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

X780 ~ X7FF								BYTE	1
	X78□	X79□	X7A□	X7B□	X7C□	X7D□	X7E□	X7F□	
0	CF80	CF90	CFA0	CFB0	CFC0	CFD0	CFE0	CFF0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	CF88	CF98	CFA8	CFB8	CFC8	CFD8	CFE8	CFF8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Process output Y000 ~ 7FF

Y000 ~ Y07F								BYTE	1
	Y00□	Y01□	Y02□	Y03□	Y04□	Y05□	Y06□	Y07□	
0	D000	D010	D020	D030	D040	D050	D060	D070	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D008	D018	D028	D038	D048	D058	D068	D078	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y080 ~ Y0FF								BYTE	1
	Y08□	Y09□	Y0A□	Y0B□	Y0C□	Y0D□	Y0E□	Y0F□	
0	E480	E490	E4A0	E4B0	E4C0	E4D0	E4E0	E4F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E488	E498	E4A8	E4B8	E4C8	E4D8	E4E8	E4F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y100 ~ Y17F

BYTE 1

	Y10□	Y11□	Y12□	Y13□	Y14□	Y15□	Y16□	Y17□	
0	E900	E910	E920	E930	E940	E950	E960	E970	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E908	E918	E928	E938	E948	E958	E968	E978	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y180 ~ Y1FF

BYTE 1

	Y18□	Y19□	Y1A□	Y1B□	Y1C□	Y1D□	Y1E□	Y1F□	
0	E580	E590	E5A0	E5B0	E5C0	E5D0	E5E0	E5F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	E588	E598	E5A8	E5B8	E5C8	E5D8	E5E8	E5F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y200 ~ Y27F

BYTE 1

	Y20□	Y21□	Y22□	Y23□	Y24□	Y25□	Y26□	Y27□	
0	D200	D210	D220	D230	D240	D250	D260	D270	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D208	D218	D228	D238	D248	D258	D268	D278	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y280 ~ Y2FF

BYTE 1

	Y28□	Y29□	Y2A□	Y2B□	Y2C□	Y2D□	Y2E□	Y2F□	
0	D280	D290	D2A0	D2B0	D2C0	D2D0	D2E0	D2F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D288	D298	D2A8	D2B8	D2C8	D2D8	D2E8	D2F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y300 ~ Y37F

BYTE 1

	Y30□	Y31□	Y32□	Y33□	Y34□	Y35□	Y36□	Y37□	
0	D300	D310	D320	D330	D340	D350	D360	D370	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D308	D318	D328	D338	D348	D358	D368	D378	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y380 ~ Y3FF

BYTE 1

	Y38□	Y39□	Y3A□	Y3B□	Y3C□	Y3D□	Y3E□	Y3F□	
0	D380	D390	D3A0	D3B0	D3C0	D3D0	D3E0	D3F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D388	D398	D3A8	D3B8	D3C8	D3D8	D3E8	D3F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y400 ~ Y47F

BYTE 1

	Y40□	Y41□	Y42□	Y43□	Y44□	Y45□	Y46□	Y47□	
0	D400	D410	D420	D430	D440	D450	D460	D470	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D408	D418	D428	D438	D448	D458	D468	D478	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y480 ~ Y4FF

BYTE 1

	Y48□	Y49□	Y4A□	Y4B□	Y4C□	Y4D□	Y4E□	Y4F□	
0	D480	D490	D4A0	D4B0	D4C0	D4D0	D4E0	D4F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D488	D498	D4A8	D4B8	D4C8	D4D8	D4E8	D4F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y500 ~ Y57F

BYTE 1

	Y50□	Y51□	Y52□	Y53□	Y54□	Y55□	Y56□	Y57□	
0	D500	D510	D520	D530	D540	D550	D560	D570	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D508	D518	D528	D538	D548	D558	D568	D578	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y580 ~ Y5FF

BYTE 1

	Y58□	Y59□	Y5A□	Y5B□	Y5C□	Y5D□	Y5E□	Y5F□	
0	D580	D590	D5A0	D5B0	D5C0	D5D0	D5E0	D5F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D588	D598	D5A8	D5B8	D5C8	D5D8	D5E8	D5F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y600 ~ Y67F

BYTE 1

	Y60□	Y61□	Y62□	Y63□	Y64□	Y65□	Y66□	Y67□	
0	D600	D610	D620	D630	D640	D650	D660	D670	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D608	D618	D628	D638	D648	D658	D668	D678	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y680 ~ Y6FF

BYTE 1

	Y68□	Y69□	Y6A□	Y6B□	Y6C□	Y6D□	Y6E□	Y6F□	
0	D680	D690	D6A0	D6B0	D6C0	D6D0	D6E0	D6F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D688	D698	D6A8	D6B8	D6C8	D6D8	D6E8	D6F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y700 ~ Y77F								BYTE	1
	Y70□	Y71□	Y72□	Y73□	Y74□	Y75□	Y76□	Y77□	
0	D700	D710	D720	D730	D740	D750	D760	D770	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D708	D718	D728	D738	D748	D758	D768	D778	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Y780 ~ Y7FF								BYTE	1
	Y78□	Y79□	Y7A□	Y7A□	Y7B□	Y7C□	Y7D□	Y7E□	
0	D780	D790	D7A0	D7B0	D7C0	D7D0	D7E0	D7F0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	D788	D798	D7A8	D7B8	D7C8	D7D8	D7E8	D7F8	8
9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F

Temporary memory M000 ~ MA23

M000 ~ M089									BYTE	1
	M00□	M01□	M02□	M03□	M04□	M05□	M06□	M07□	M08□	
0	D800	D80A	D814	D81E	D828	D832	D83C	D846	D850	0
1	1	B	5	F	9	3	D	7	1	1
2	2	C	6	20	A	4	E	8	2	2
3	3	D	7	1	B	5	F	9	3	3
4	4	E	8	2	C	6	40	A	4	4
5	D805	D80F	D819	D823	D82D	D837	D841	D84B	D855	5
6	6	10	A	4	E	8	2	C	6	6
7	7	1	B	5	F	9	3	D	7	7
8	8	2	C	6	30	A	4	E	8	8
9	9	3	D	7	1	B	5	F	9	9

M090 ~ M179									BYTE	1
	M09□	M10□	M11□	M12□	M13□	M14□	M15□	M16□	M17□	
0	D85A	D864	D86E	D878	D882	D88C	D896	D8A0	D8AA	0
1	B	5	F	9	3	D	7	1	B	1
2	C	6	70	A	4	E	8	2	C	2
3	D	7	1	B	5	F	9	3	D	3
4	E	8	2	C	6	90	A	4	E	4
5	D85F	D869	D873	D87D	D887	D891	D89B	D8A5	D8AF	5
6	60	A	4	E	8	2	C	6	B0	6
7	1	B	5	F	9	3	D	7	1	7
8	2	C	6	80	A	4	E	8	2	8
9	3	D	7	1	B	5	F	9	3	9

M180 ~ M269									BYTE	1
	M18□	M19□	M20□	M21□	M22□	M23□	M24□	M25□	M26□	
0	D8B4	D8BE	D8C8	D8D2	D8DC	D8E6	D8F0	D8FA	D904	0
1	5	F	9	3	D	7	1	B	5	1
2	6	C0	A	4	E	8	2	C	6	2
3	7	1	B	5	F	9	3	D	7	3
4	8	2	C	6	E0	A	4	E	8	4
5	D8B9	D8C3	D8C0	D8D7	D8E1	D8EB	D8F5	D8FF	D909	5
6	A	4	E	8	2	C	6	D900	A	6
7	B	5	F	9	3	D	7	1	B	7
8	C	6	D0	A	4	E	8	2	C	8
9	D	7	1	B	5	F	9	3	D	9

M270 ~ M359									BYTE	1
	M27□	M28□	M29□	M30□	M31□	M32□	M33□	M34□	M35□	
0	D90E	D918	D922	D92C	D936	D940	D94A	D954	D95E	0
1	F	9	3	D	7	1	B	5	F	1
2	10	A	4	E	8	2	C	6	60	2
3	1	B	5	F	9	3	D	7	1	3
4	2	C	6	30	A	4	E	8	2	4
5	D913	D91D	D927	D931	D93B	D945	D94F	D959	D963	5
6	4	E	8	2	C	6	50	A	4	6
7	5	F	9	3	D	7	1	B	5	7
8	6	20	A	4	E	8	2	C	6	8
9	7	1	B	5	F	9	3	D	7	9

M360 ~ M449									BYTE	1
	M36□	M37□	M38□	M39□	M40□	M41□	M42□	M43□	M44□	
0	D968	D972	D97C	D986	D990	D99A	D9A4	D9AE	D9B8	0
1	9	3	D	7	1	B	5	F	9	1
2	A	4	E	8	2	C	6	B0	A	2
3	B	5	F	9	3	D	7	1	B	3
4	C	6	80	A	4	E	8	2	C	4
5	D96D	D977	D981	D98B	D995	D99F	D9A9	D9B3	D9BD	5
6	E	8	2	C	6	A0	A	4	E	6
7	F	9	3	D	7	1	B	5	F	7
8	70	A	4	E	8	2	C	6	C0	8
9	1	B	5	F	9	3	D	7	1	9

M450 ~ M539

BYTE 1

	M45□	M46□	M47□	M48□	M49□	M50□	M51□	M52□	M53□	
0	D9C2	D9CC	D9D6	D9E0	D9EA	D9F4	D9FE	DA08	DA12	0
1	3	D	7	1	B	5	F	9	3	1
2	4	E	8	2	C	6	DA00	A	4	2
3	5	F	9	3	D	7	1	B	5	3
4	6	D0	A	4	E	8	2	C	6	4
5	D9C7	D9D1	D9DB	D9E5	D9EF	D9F9	DA03	DA0D	DA17	5
6	8	2	C	6	F0	A	4	E	8	6
7	9	3	D	7	1	B	5	F	9	7
8	A	4	E	8	2	C	6	10	A	8
9	B	5	F	9	3	D	7	1	B	9

M540 ~ M629

BYTE 1

	M54□	M55□	M56□	M57□	M58□	M59□	M60□	M61□	M62□	
0	DA1C	DA26	DA30	DA3A	DA44	DA4E	DA58	DA62	DA6C	0
1	D	7	1	B	5	F	9	3	D	1
2	E	8	2	C	6	50	A	4	E	2
3	F	9	3	D	7	1	B	5	F	3
4	20	A	4	E	8	2	C	6	70	4
5	DA21	DA2B	DA35	DA3F	DA49	DA53	DA5D	DA67	DA71	5
6	2	C	6	40	A	4	E	8	2	6
7	3	D	7	1	B	5	F	9	3	7
8	4	E	8	2	C	6	60	A	4	8
9	5	F	9	3	D	7	1	B	5	9

M630 ~ M719

BYTE 1

	M63□	M64□	M65□	M66□	M67□	M68□	M69□	M70□	M71□	
0	DA76	DA80	DA8A	DA94	DA9E	DAA8	DAB2	DABC	DAC6	0
1	7	1	8	5	F	9	3	D	7	1
2	8	2	C	6	A0	A	4	E	8	2
3	9	3	D	7	1	B	5	F	9	3
4	A	4	E	8	2	C	6	C0	A	4
5	DA7B	DA85	DA8F	DA99	DAA3	DAAD	DAB7	DAC1	DACB	5
6	C	6	90	A	4	E	8	2	C	6
7	D	7	1	B	5	F	9	3	D	7
8	E	8	2	C	6	B0	A	4	E	8
9	F	9	3	D	7	1	B	5	F	9

M720 ~ M809

BYTE 1

	M72□	M73□	M74□	M75□	M76□	M77□	M78□	M79□	M80□	
0	DAD0	DADA	DAE4	DAEE	DAF8	DB02	DB0C	DB16	DB20	0
1	1	B	5	F	9	3	D	7	1	1
2	2	C	6	F0	A	4	E	8	2	2
3	3	D	7	1	B	5	F	9	3	3
4	4	E	8	2	C	6	10	A	4	4
5	DAD5	DADF	DAE9	DAF3	DAFD	DB07	DB11	DB1B	DB25	5
6	6	E0	A	4	E	8	2	C	6	6
7	7	1	B	5	F	9	3	D	7	7
8	8	2	C	6	DB00	A	4	E	8	8
9	9	3	D	7	1	B	5	F	9	9

M810 ~ M899

BYTE 1

	M81□	M82□	M83□	M84□	M85□	M86□	M87□	M88□	M89□	
0	DB2A	DB34	DB3E	DB48	DB52	DB5C	DB66	DB70	DB7A	0
1	B	5	F	9	3	D	7	1	B	1
2	C	6	40	A	4	E	8	2	C	2
3	D	7	1	B	5	F	9	3	D	3
4	E	8	2	C	6	60	A	4	E	4
5	DB2F	DB39	DB43	DB4D	DB57	DB61	DB6B	DB75	DB7F	5
6	30	A	4	E	8	2	C	6	80	6
7	1	B	5	F	9	3	D	7	1	7
8	2	C	6	50	A	4	E	8	2	8
9	3	D	7	1	B	5	F	9	3	9

M900 ~ M989

BYTE 1

	M90□	M91□	M92□	M93□	M94□	M95□	M96□	M97□	M98□	
0	DB84	DB8E	DB98	DBA2	DBAC	DBB6	DBC0	DBCA	DBD4	0
1	5	F	9	3	D	7	1	B	5	1
2	6	90	A	4	E	8	2	C	6	2
3	7	1	B	5	F	9	3	D	7	3
4	8	2	C	6	B0	A	4	E	8	4
5	DB89	DB93	DB9D	DBA7	DBB1	DBBB	DBC5	DBCF	DBD9	5
6	A	4	E	8	2	C	6	D0	A	6
7	B	5	F	9	3	D	7	1	B	7
8	C	6	A0	A	4	E	8	2	C	8
9	D	7	1	B	5	F	9	3	D	9

M990 ~ MA23

BYTE 1

	M99□	MA0□	MA1□	MA2□						
0	DBDE	DBE8	DBF2	DBFC						
1	F	9	3	D						
2	E0	A	4	E						
3	1	B	5	F						
4	2	C	6							
5	DBE3	DBED	DBF7							
6	4	E	8							
7	5	F	9							
8	6	F0	A							
9	7	1	B							

T/C000 ~ T/C089

BYTE 1

	T/C00□	T/C01□	T/C02□	T/C03□	T/C04□	T/C05□	T/C06□	T/C07□	T/C08□	
0	DC00	DC0A	DC14	DC1E	DC28	DC32	DC3C	DC46	DC50	0
1	1	B	5	F	9	3	D	7	1	1
2	2	C	6	20	A	4	E	8	2	2
3	3	D	7	1	B	5	F	9	3	3
4	4	E	8	2	C	6	40	A	4	4
5	DC05	DC0F	DC19	DC23	DC2D	DC37	DC41	DC4B	DC55	5
6	6	10	A	4	E	8	2	C	6	6
7	7	1	B	5	F	9	3	D	7	7
8	8	2	C	6	30	A	4	E	8	8
9	9	3	D	7	1	B	5	F	9	9

Coil/contact area of timer/counter T/C000 ~ 255

T/C090 ~ T/C179									BYTE	1
	T/C09	T/C10	T/C11	T/C12	T/C13	T/C14	T/C15	T/C16	T/C17	
0	DC5A	DC64	DC6E	DC78	DC82	DC8C	DC96	DCAO	DCAA	0
1	B	5	F	9	3	D	7	1	B	1
2	C	6	70	A	4	E	8	2	C	2
3	D	7	1	B	5	F	9	3	D	3
4	E	8	2	C	6	90	A	4	E	4
5	DC5F	DC69	DC73	DC7D	DC87	DC91	DC9B	DCA5	DCAF	5
6	60	A	4	E	8	2	C	6	BO	6
7	1	B	5	F	9	3	D	7	1	7
8	2	C	6	80	A	4	E	8	2	8
9	3	D	7	1	B	5	F	9	3	9

T/C180 ~ T/C255									BYTE	1
	T/C18	T/C19	T/C20	T/C21	T/C22	T/C23	T/C24	T/C25		
0	DCB4	DCBE	DCC8	DCD2	DCDC	DCE6	DCFO	DCFA		0
1	5	F	9	3	D	7	1	B		1
2	6	CO	A	4	E	8	2	C		2
3	7	1	B	5	F	9	3	D		3
4	8	2	C	6	EO	A	4	E		4
5	DCB9	DCC3	DCCD	DCD7	DCE1	DCEB	DCF5	DCFF		5
6	A	4	E	8	2	C	6			6
7	B	5	F	9	3	D	7			7
8	C	6	DO	A	4	E	8			8
9	D	7	1	B	5	F	9			9

Temporary valve area of timer/counter T/C000 ~ 255

T/C000 ~T/C129													BYTE	2
	T/c00	T/c01	T/c02	T/c03	T/c04	T/c05	T/c06	T/c07	T/c08	T/c09	T/c10	T/c11	T/c12	
0	DD00	DD14	DD28	DD3C	DD50	DD64	DD78	DD8C	DDAO	DDB4	DDC8	DDDC	DDFO	0
	1	5	9	D	1	5	9	D	1	5	O	D	1	
1	2	6	A	E	2	6	A	E	2	6	A	E	2	1
	3	7	B	F	3	7	B	F	3	7	B	F	3	
2	4	8	C	DD40	4	8	C	DD90	4	8	C	DDEO	4	2
	5	9	D	1	5	9	D	1	5	9	D	1	5	
3	6	A	E	2	6	A	E	2	6	A	E	2	6	3
	7	B	F	3	7	B	F	3	7	B	F	3	7	
4	8	C	DD30	4	8	C	DD80	4	8	C	DDDO	4	8	4
	9	D	1	5	9	D	1	5	9	D	1	5	9	
5	A	E	2	6	A	E	2	6	A	E	2	6	A	5
	B	F	3	7	B	F	3	7	B	F	3	7	B	
6	C	DD20	4	8	C	DD70	4	8	C	DDCO	4	8	C	6
	D	1	5	9	D	1	5	9	D	1	5	9	D	
7	E	2	6	A	E	2	6	A	E	2	6	A	E	7
	F	3	7	B	F	3	7	B	F	3	7	B	F	
8	DD10	4	8	C	DD60	4	8	C	DDBO	4	8	C	DE00	8
	1	5	9	D	1	5	9	D	1	5	9	D	1	
9	2	6	A	E	2	6	A	E	2	6	A	E	2	9
	3	7	B	F	3	7	B	F	3	7	B	F	3	

T/C130 ~ T/C255 BYTE 2

	T/c13	T/c14	T/c15	T/c16	T/c17	T/c18	T/c19	T/c20	T/c21	T/c22	T/c23	T/c24	T/c25	
0	DE04	DE18	DE2C	DE40	DE54	DE68	DE7C	DE90	DEA4	DEB8	DECC	DEE0	DEF4	0
	5	9	D	1	5	9	D	1	5	9	D	1	5	
1	6	A	E	2	6	A	E	2	6	A	E	2	6	1
	7	B	F	3	7	B	F	3	7	B	F	3	7	
2	8	C	DE30	4	8	C	DE80	4	8	C	DED0	4	8	2
	9	D	1	5	9	D	1	5	9	D	1	5	9	
3	A	E	2	6	A	E	2	6	A	E	2	6	A	3
	B	F	3	7	B	F	3	7	B	F	3	7	B	
4	C	DE20	4	8	C	DE70	4	8	C	DEC0	4	8	C	4
	D	1	5	9	D	1	5	9	D	1	5	9	D	
5	E	2	6	A	E	2	6	A	E	2	6	A	E	5
	F	3	7	B	F	3	7	B	F	3	7	B	F	
6	DE10	4	8	C	DE60	4	8	C	DEB0	4	8	C		6
	1	5	9	D	1	5	9	D	1	5	9	D		
7	2	6	A	E	2	6	A	E	2	6	A	E		7
	3	7	B	F	3	7	B	F	3	7	B	F		
8	4	8	C	DE50	4	8	C	DEA0	4	8	C	DEF0		8
	5	9	D	1	5	9	D	1	5	9	D	1		
9	6	A	E	2	6	A	E	2	6	A	E	2		9
	7	B	F	3	7	B	F	3	7	B	F	3		

Data register D000 ~ DA23

D000 ~ D129 BYTE 2

	D00	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10	D11	D12	
0	C000	C014	C028	C03C	C050	C064	C078	C08C	C0A0	C0B4	C0C8	C0DC	C0F0	0
	1	5	9	D	1	5	9	D	1	5	9	D	1	
1	2	6	A	E	2	6	A	E	2	6	A	E	2	1
	3	7	B	F	3	7	B	F	3	7	B	F	3	
2	4	8	C	C040	4	8	C	C090	4	8	C	C0E0	4	2
	5	9	D	1	5	9	D	1	5	9	D	1	5	
3	6	A	E	2	6	A	E	2	6	A	E	2	6	3
	7	B	F	3	7	B	F	3	7	B	F	3	7	
4	8	C	C030	4	8	C	C080	4	8	C	C0D0	4	8	4
	9	D	1	5	9	D	1	5	9	D	1	5	9	
5	A	E	2	6	A	E	2	6	A	E	2	6	A	5
	B	F	3	7	B	F	3	7	B	F	3	7	B	
6	C	C020	4	8	C	C070	4	8	C	C0C0	4	8	C	6
	D	1	5	9	D	1	5	9	D	1	5	9	D	
7	E	2	6	A	E	2	6	A	E	2	6	A	E	7
	F	3	7	B	F	3	7	B	F	3	7	B	F	
8	C010	4	8	C	C060	4	8	C	C0B0	4	8	C	C100	8
	1	5	9	D	1	5	9	D	1	5	9	D	1	
9	2	6	A	E	2	6	A	E	2	6	A	E	2	9
	3	7	B	F	3	7	B	F	3	7	B	F	3	

D130 ~ D259 BYTE 2

	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	
0	C104	C118	C12C	C140	C154	C168	C17C	C190	C1A4	C1B8	C1CC	C1E0	C1F4	0
	5	9	D	1	5	9	D	1	5	9	D	1	5	
1	6	A	E	2	6	A	E	2	6	A	E	2	6	1
	7	B	F	3	7	B	F	3	7	B	F	3	7	
2	8	C	C130	4	8	C	C180	4	8	C	C1D0	4	8	2
	9	D	1	5	9	D	1	5	9	D	1	5	9	
3	A	E	2	6	A	E	2	6	A	E	2	6	A	3
	B	F	3	7	B	F	3	7	B	F	3	7	B	
4	C	C120	4	8	C	C170	4	8	C	C1C0	4	8	C	4
	D	1	5	9	D	1	5	9	D	1	5	9	D	
5	E	2	6	A	E	2	6	A	E	2	6	A	E	5
	F	3	7	B	F	3	7	B	F	3	7	B	F	
6	C110	4	8	C	C160	4	8	C	C1B0	4	8	C	C200	6
	1	5	9	D	1	5	9	D	1	5	9	D	1	
7	2	6	A	E	2	6	A	E	2	6	A	E	2	7
	3	7	B	F	3	7	B	F	3	7	B	F	3	
8	4	8	C	C150	4	8	C	C1A0	4	8	C	C1F0	4	8
	5	9	D	1	5	9	D	1	5	9	D	1	5	
9	6	A	E	2	6	A	E	2	6	A	E	2	6	9
	7	B	F	3	7	B	F	3	7	B	F	3	7	

D260 ~ D389

BYTE 2

	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36	D37	D38	
0	C208	C21C	C230	C244	C258	C26C	C280	C294	C2A8	C2BC	C2D0	C2E4	C2F8	0
	9	D	1	5	9	D	1	5	9	D	1	5	9	
1	A	E	2	6	A	E	2	6	A	E	2	6	A	1
	B	F	3	7	B	F	3	7	B	F	3	7	B	
2	C	C220	4	8	C	C270	4	8	C	C2C0	4	8	C	2
	D	1	5	9	D	1	5	9	D	1	5	9	D	
3	E	2	6	A	E	2	6	A	E	2	6	A	E	3
	F	3	7	B	F	3	7	B	F	3	7	B	F	
4	C210	4	8	C	C260	4	8	C	C2B0	4	8	C	C300	4
	1	5	9	D	1	5	9	D	1	5	9	D	1	
5	2	6	A	E	2	6	A	E	2	6	A	E	2	5
	3	7	B	F	3	7	B	F	3	7	B	F	3	
6	4	8	C	C250	4	8	C	C2A0	4	8	C	C2F0	4	6
	5	9	D	1	5	9	D	1	5	9	D	1	5	
7	6	A	E	2	6	A	E	2	6	A	E	2	6	7
	7	B	F	3	7	B	F	3	7	B	F	3	7	
8	8	C	C240	4	8	C	C290	4	8	C	C2E0	4	8	8
	9	D	1	5	9	D	1	5	9	D	1	5	9	
9	A	E	2	6	A	E	2	6	A	E	2	6	A	9
	B	F	3	7	B	F	3	7	B	F	3	7	B	

D390 ~ D519

BYTE 2

	D39	D40	D41	D42	D43	D44	D45	D46	D47	D48	D49	D50	D51	
0	C30C	C320	C334	C348	C35C	C370	C384	C398	C3AC	C3C0	C3D4	C3E8	C3FC	0
	D	1	5	9	D	1	5	9	D	1	5	9	D	
1	E	2	6	A	E	2	6	A	E	2	6	A	E	1
	F	3	7	B	F	3	7	B	F	3	7	B	F	
2	C310	4	8	C	C360	4	8	C	C3B0	4	8	C	C400	2
	1	5	9	D	1	5	9	D	1	5	9	D	1	
3	2	6	A	E	2	6	A	E	2	6	A	E	2	3
	3	7	B	F	3	7	B	F	3	7	B	F	3	
4	4	8	C	C350	4	8	C	C3A0	4	8	C	C3F0	4	4
	5	9	D	1	5	9	D	1	5	9	D	1	5	
5	6	A	E	2	6	A	E	2	6	A	E	2	6	5
	7	B	F	3	7	B	F	3	7	B	F	3	7	
6	8	C	C340	4	8	C	C390	4	8	C	C3E0	4	8	6
	9	D	1	5	9	D	1	5	9	D	1	5	9	
7	A	E	2	6	A	E	2	6	A	E	2	6	A	7
	B	F	3	7	B	F	3	7	B	F	3	7	B	
8	C	C330	4	8	C	C380	4	8	C	C3D0	4	8	C	8
	D	1	5	9	D	1	5	9	D	1	5	9	D	
9	E	2	6	A	E	2	6	A	E	2	6	A	E	9
	F	3	7	B	F	3	7	B	F	3	7	B	F	

D520 ~ D649

BYTE 2

	D52	D53	D54	D55	D56	D57	D58	D59	D60	D61	D62	D63	D64	
0	C410	C424	C438	C44C	C460	C474	C488	C49C	C4B0	C4C4	C4D8	C4EC	C400	0
	1	5	9	D	1	5	9	D	1	5	9	D	1	
1	2	6	A	E	2	6	A	E	2	6	A	E	2	1
	3	7	B	F	3	7	B	F	3	7	B	F	3	
2	4	8	C	C450	4	8	C	C4A0	4	8	C	C4F0	4	2
	5	9	D	1	5	9	D	1	5	9	D	1	5	
3	6	A	E	2	6	A	E	2	6	A	E	2	6	3
	7	B	F	3	7	B	F	3	7	B	F	3	7	
4	8	C	C440	4	8	C	C490	4	8	C	C4E0	4	8	4
	9	D	1	5	9	D	1	5	9	D	1	5	9	
5	A	E	2	6	A	E	2	6	A	E	2	6	A	5
	B	F	3	7	B	F	3	7	B	F	3	7	B	
6	C	C430	4	8	C	C480	4	8	C	C4D0	4	8	C	6
	D	1	5	9	D	1	5	9	D	1	5	9	D	
7	E	2	6	A	E	2	6	A	E	2	6	A	E	7
	F	3	7	B	F	3	7	B	F	3	7	B	F	
8	C420	4	8	C	C470	4	8	C	C4C0	4	8	C	C410	8
	1	5	9	D	1	5	9	D	1	5	9	D	1	
9	2	6	A	E	2	6	A	E	2	6	A	E	2	9
	3	7	B	F	3	7	B	F	3	7	B	F	3	

D650 ~ D779 BYTE 2

	D65	D66	D67	D68	D69	D70	D71	D72	D73	D74	D75	D76	D77	
0	C514	C528	C53C	C550	C564	C578	C58C	C5A0	C5B4	C5C8	C5DC	C5F0	C604	0
	5	9	D	1	5	9	D	1	5	9	D	1	5	
1	6	A	E	2	6	A	E	2	6	A	E	2	6	1
	7	B	F	3	7	B	F	3	7	B	F	3	7	
2	8	C	C540	4	8	C	C590	4	8	C	C5E0	4	8	2
	9	D	1	5	9	D	1	5	9	D	1	5	9	
3	A	E	2	6	A	E	2	6	A	E	2	6	A	3
	B	F	3	7	B	F	3	7	B	F	3	7	B	
4	C	C530	4	8	C	C580	4	8	C	C5D0	4	8	C	4
	D	1	5	9	D	1	5	9	D	1	5	9	D	
5	E	2	6	A	E	2	6	A	E	2	6	A	E	5
	F	3	7	B	F	3	7	B	F	3	7	B	F	
6	C520	4	8	C	C570	4	8	C	C5C0	4	8	C	C610	6
	1	5	9	D	1	5	9	D	1	5	9	D	1	
7	2	6	A	E	2	6	A	E	2	6	A	E	2	7
	3	7	B	F	3	7	B	F	3	7	B	F	3	
8	4	8	C	C560	4	8	C	C5B0	4	8	C	C600	4	8
	5	9	D	1	5	9	D	1	5	9	D	1	5	
9	6	A	E	2	6	A	E	2	6	A	E	2	6	9
	7	B	F	3	7	B	F	3	7	B	F	3	7	

D780 ~ D909 BYTE 2

	D78	D79	D80	D81	D82	D83	D84	D85	D86	D87	D88	D89	D90	
0	C618	C62C	C640	C654	C668	C67C	C690	C6A4	C6B8	C6CC	C6E0	C6F4	C708	0
	9	D	1	5	9	D	1	5	9	D	1	5	9	
1	A	E	2	6	A	E	2	6	A	E	2	6	A	1
	B	F	3	7	B	F	3	7	B	F	3	7	B	
2	C	C630	4	8	C	C680	4	8	C	C6D0	4	8	C	2
	D	1	5	9	D	1	5	9	D	1	5	9	D	
3	E	2	6	A	E	2	6	A	E	2	6	A	E	3
	F	3	7	B	F	3	7	B	F	3	7	B	F	
4	C620	4	8	C	C670	4	8	C	C6C0	4	8	C	C710	4
	1	5	9	D	1	5	9	D	1	5	9	D	1	
5	2	6	A	E	2	6	A	E	2	6	A	E	2	5
	3	7	B	F	3	7	B	F	3	7	B	F	3	
6	4	8	C	C660	4	8	C	C6B0	4	8	C	C700	4	6
	5	9	D	1	5	9	D	1	5	9	D	1	5	
7	6	A	E	2	6	A	E	2	6	A	E	2	6	7
	7	B	F	3	7	B	F	3	7	B	F	3	7	
8	8	C	C650	4	8	C	C6A0	4	8	C	C6F0	4	8	8
	9	D	1	5	9	D	1	5	9	D	1	5	9	
9	A	E	2	6	A	E	2	6	A	E	2	6	A	9
	B	F	3	7	B	F	3	7	B	F	3	7	B	

D910 ~ DA23 BYTE 2

	D91	D92	D93	D94	D95	D96	D97	D98	D99	DA0	DA1	DA2		
0	C71C	C730	C744	C758	C76C	C780	C794	C7A8	C7BC	C7D0	C7E4	C7F8		0
	D	1	5	9	D	1	5	9	D	1	5	9		
1	E	2	6	A	E	2	6	A	E	2	6	A		1
	F	3	7	B	F	3	7	B	F	3	7	B		
2	C720	4	8	C	C770	4	8	C	C7C0	4	8	C		2
	1	5	9	D	1	5	9	D	1	5	9	D		
3	2	6	A	E	2	6	A	E	2	6	A	E		3
	3	7	B	F	3	7	B	F	3	7	B	F		
4	4	8	C	C760	4	8	C	C7B0	4	8	C			4
	5	9	D	1	5	9	D	1	5	9	D			
5	6	A	E	2	6	A	E	2	6	A	E			5
	7	B	F	3	7	B	F	3	7	B	F			
6	8	C	C750	4	8	C	C7A0	4	8	C	C7F0			6
	9	D	1	5	9	D	1	5	9	D	1			
7	A	E	2	6	A	E	2	6	A	E	2			7
	B	F	3	7	B	F	3	7	B	F	3			
8	C	C740	4	8	C	C790	4	8	C	C7E0	4			8
	D	1	5	9	D	1	5	9	D	1	5			
9	E	2	6	A	E	2	6	A	E	2	6			9
	F	3	7	B	F	3	7	B	F	3	7			

External failure memory F000 ~ 191

F000 ~ F089									BYTE	1
	F00	F01	F02	F03	F04	F05	F06	F07	F08	
0	F300	F30A	F314	F31E	F328	F332	F33C	F346	F350	0
1	1	B	5	F	9	3	D	7	1	1
2	2	C	6	20	A	4	E	8	2	2
3	3	D	7	1	B	5	F	9	3	3
4	4	E	8	2	C	6	40	A	4	4
5	F305	F30F	F319	F323	F32D	F337	F341	F34B	F355	5
6	6	10	A	4	E	8	2	C	6	6
7	7	1	B	5	F	9	3	D	7	7
8	8	2	C	6	30	A	4	E	8	8
9	9	3	D	7	1	B	5	F	9	9

F090 ~ F179									BYTE	1
	F09	F10	F11	F12	F13	F14	F15	F16	F17	
0	DF5A	DF64	DF6F	DF78	DF82	DF8C	DF96	DFAD	DFAA	0
1	B	5	F	9	3	D	7	1	B	1
2	C	6	70	A	4	E	8	2	C	2
3	D	7	1	B	5	F	9	3	D	3
4	E	8	2	C	6	90	A	4	E	4
5	DF5F	DF69	DF73	DF7D	DF87	DF91	DF9B	DFA5	DFAF	5
6	60	A	4	E	8	2	C	6	B0	6
7	1	B	5	F	9	3	D	7	1	7
8	2	C	6	80	A	4	E	8	2	8
9	3	D	7	1	B	5	F	9	3	9

F180 ~F191									BYTE	1
	F18	F19								
0	DFB4	DFBE								0
1	5	F								1
2	6									2
3	7									3
4	8									4
5	DFB9									5
6	A									6
7	B									7
8	C									8
9	D									9

Master control K000 ~ 063

K000 ~ K063									BYTE	1
	K00	K01	K02	K03	K04	K05	K06			
0	F500	F50A	F514	F51E	F528	F532	F53C			0
1	1	B	5	F	9	3	D			1
2	2	C	6	20	A	4	E			2
3	3	D	7	1	B	5	F			3
4	4	E	8	2	C	6				4
5	F505	F50F	F519	F523	F52D	F537				5
6	6	10	A	4	E	8				6
7	7	1	B	5	F	9				7
8	8	2	C	6	30	A				8
9	9	3	D	7	1	B				9

13

10. CAUTIONS FOR USE OF KD51PR

The KD51PR can be connected to either port of RS-232-C CH1 or CH2 of KD51E. However, prepare the BASIC program using care for the following point:

If data is sent from the KD51E to the KD51PR during printing or line feeding operation of KD51PR, the KD51PR cannot receive the data as correct data. Therefore, "?" marks are printed. See Example 1.
To avoid such trouble, prepare a user program as shown in Example 2.
The following examples show programs which cause the KD51PR to repeatedly print "ABCDE".

Example 1:

KD51PR is used in 2K buffer OFF and buffer full SET mode.

BASIC program

```
100  L PRINT "ABCDE"
110  GOTO 100
```

Since the CR code (0DH) and LF code (0AH) are sent at the end of printed data during execution of line 100, the KD51PR starts printing.
However, since the processing of line 100 is resumend immediately after execution of line 110, the data "ABCDE", CR code and LF code are sent from the KD51E although the KD51PR is still performing the printing operation.

Print result

```
ABCDE
?A?B?D?
?
?E?
?E??D?
?D?E??A?C?E?
```

Example 2:

Setting of KD51PR -------- 2K buffer ON, buffer full SET

BASIC program

```
100  L PRINT "ABCDE"
110  L PRINT "12345"
120  L PRINT *$03,
130  Z TIME 1000
140  COTO 100
```

Print data and function codes (CR code and LF code) are sent from KD51E.

Print command code 03H is sent from KD51E.

Since 03H code has been received, KD51PR is performing printing operation. ZTIME command is provided to wait for completion of printing.

Print result

```
ABCDE
12345
ABCDE
12345
ABCDE
12345
```

In this example, it takes eight seconds from when the print command is given to the completion of printing of one "ABCDE" and "12345".

- "ABCDE" printing – 2.5 seconds
- CR, LF (line feed) – 2.5 seconds
- "12345" printing – 2.5 seconds
- CR, LF (line feed) – 2.5 seconds

- Like Example 1, the receive buffer of KD51PR has a capacity of only one line (32 characters) in 2K buffer OFF mode. Therefore, if the line 100 is programmed like

  100 LPRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF",
  (32 characters)

  overrun error occurs upon sending of LF code (0AH).
  When it is desired to print 32 characters at one time, provide a comma (,) at the end of statement, e.g. 100 LPRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF",.

- Caution for use of block data receiving (SRB) system subroutine

  When SRB is used, the baud rate of receiving port of KD51E should always be the same as the baud rate of sending port of connected equipment. Even when the baud rates are different and confused data are received, the output of SRB will be normally completed if overrun error, framing error or parity error has not occurred.

- Do not turn off the power of connected equipment when the KD51E is in the online mode (during run of user program). If the power is turned off, confused data will be sent to the KD51E and this data may enter the KD51E as input data.
  For example, when the KD51E is running a program which is waiting for keying-in operation, the KD51E may regard the data, which has been generated by power-off, as keyed-in data.

MEMO

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100 TELEX. J24532 CABLE MELCO TOKYO
NAGOYA WORKS . 1-14 , YADA-MINAMI 5 . HIGASHI-KU . NAGOYA , JAPAN

IB (NA) 66016-A (8505) REC Printed in Japan