MITSUBISHI

PROGRAMMABLE CONTROLLER

MELSEC-A

User's Manual

Intelligent communication module type AD51H

MITSUBISHI ELECTRIC

REVISIONS

*The manual number is given on the bottom left of the back cover.

Print Date	*Manual Number	Revision
Dec., 1991	IB (NA) 66345-A	First edition

INTRODUCTION

Thank you for choosing the Mitsubishi MELSEC-A Series of General Purpose Programmable Controllers. Please read this manual carefully so that the equipment is used to its optimum. A copy of this manual should be forwarded to the end User.

CONTENTS

[ONLINE PROGRAMMING & DEBUGGING]

1. GENERAL INFORMATION

This manual explains the system configurations, performance specifications, and functions of the AD51H intelligent communications module (hereafter called the AD51H).

This manual also explains (a) online programming when the AD51H is connected to a console, and (b) multitask debugging operations when the AD51H is connected to a debugger.

The following manuals should also be referred to:

(1) The AD51H-BASIC Programming Manual

This manual explains the common commands and functions of AD51H-BASIC that can be used with both the AD51H and AD51H-BASIC.

Read this manual to create the BASIC program which operates the AD51H.

(2) The SW0IX-AD51HPE Operating Manual

This manual tells how to install the contents of the SW0IX-AD51HPE system floppy disk (hereafter FD) to the hard disk (hereafter HD) of an IBM PC/AT (hereafter called a PC/AT).

Read this manual carefully to perform offline programming, file maintenance, and storage of execution programs to the ROM using a PC/AT.

REMARK

The online programming (debugging) part of this manual gives the commands for executing online programming and multitask debugging.

1.1 General Operation Information

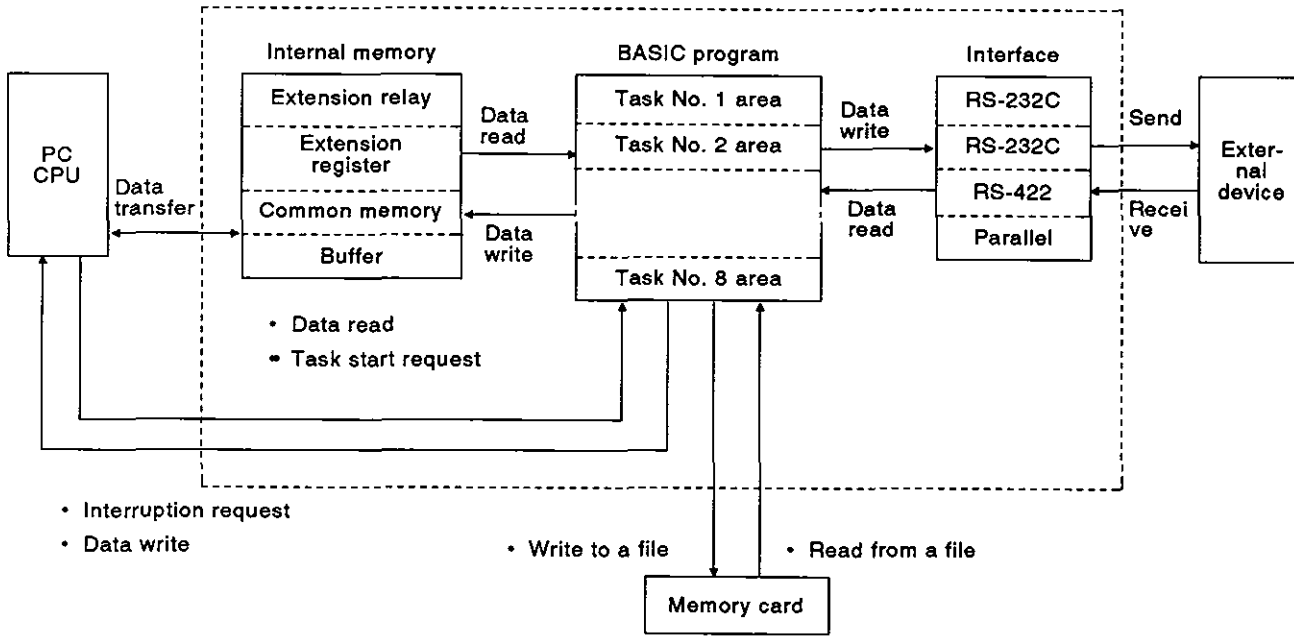
The AD51H operates with a BASIC program created by the user.

AD51H-BASIC can be used with the AD51H, and multitask processing of up to 8 tasks can be executed.

The following operations can be executed using created BASIC programs:

- Data communications with an external device
- Data transfer from/to a PC CPU
- File management using a memory card

The following shows operations when performing data communications with an external device, data transfer from/to a PC CPU, and file management using the AD51H:



(1) Data communications with an external device

The AD51H has two channels for the RS-232C interface, one channel for the RS-422 interface, one channel for the parallel interface, and two channels for the memory card interface.

Data of production orders and actual production output can, without using protocols, be sent/received to/from a console or an external device such as a personal computer (which is connected to an RS-232C or RS-422 interface).

In addition, actual production output data can be printed by a printer connected to the RS-232C, RS-422, or parallel interface.

Section 2.1 (Hardware) explains how to connect external devices and printers.

(2) Data transfer from/to a PC CPU

Device data can be read/written from/to a PC CPU by using the PC access functions (PCRD/PCWT) which transfer data from/to a PC CPU.

In addition, data can be read/written from/to a special-function module by using the PC access functions.

(The AD51H-BASIC Programming Manual gives details about the PC access functions.)

(3) File management using a memory card

Programs and data can be read/written from/to memory cards installed in the memory card interface (MEMORY CARD 1 and 2) of the AD51H by using (a) file operation commands (LOAD, SAVE), and (b) file input/output commands (PRINT#, INPUT#).

(The AD51H-BASIC Programming Manual gives details about these commands.)

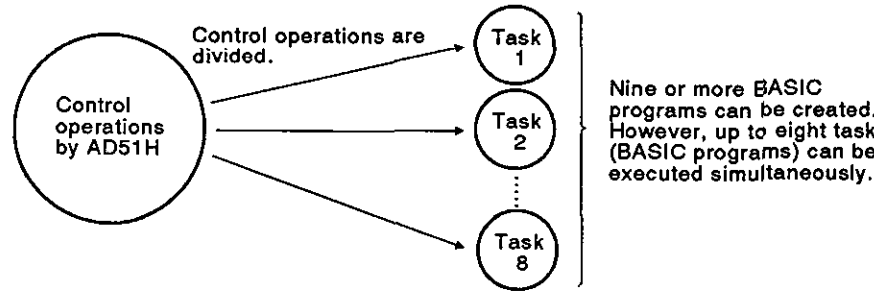
1.2 Features

The AD51H system has the following features:

(1) Up to eight BASIC programs (tasks) can be executed simultaneously.

Since multitask processing is enabled, BASIC programs can be created by dividing the control operations to be executed by the AD51H into the unit of processing.

Tasks required for each control operation can be started according to the contents of processing, and communications timing and execution between tasks can be controlled.



(2) Data communications with a connected device is enabled.

<Data communications with an external device>

Data communications with an external device connected to an RS-422 or RS-232C interface can be controlled without using protocols.

<Data communications with a console and a terminal>

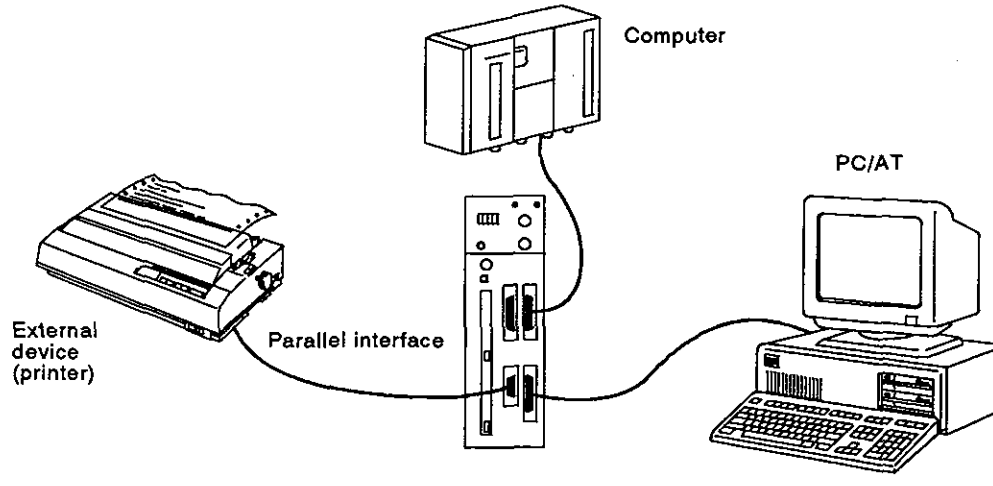
Data can be displayed on, and input through, a console and a terminal connected to an RS-422 or RS-232C interface.

A PC/AT or other devices available on the market can be used as a console or a terminal (see Section 2.1).

<Printouts using a printer>

Data can be printed out by using a printer connected to the RS-232C or parallel interface.

Printers available on the market can be used (see Section 2.6).



(3) Data communications with a PC is enabled.

Data communications can be executed from/to a PC CPU to which the AD51H is connected, or a PC CPU on MELSECNET.

(a) The following data communications can be executed with a PC CPU:

1) Read and write of device data in the PC CPU (max. 64 words/time)

2) Read and write of buffer data in a special-function module (max. 128 words/time)

3) Remote RUN/STOP of a PC CPU

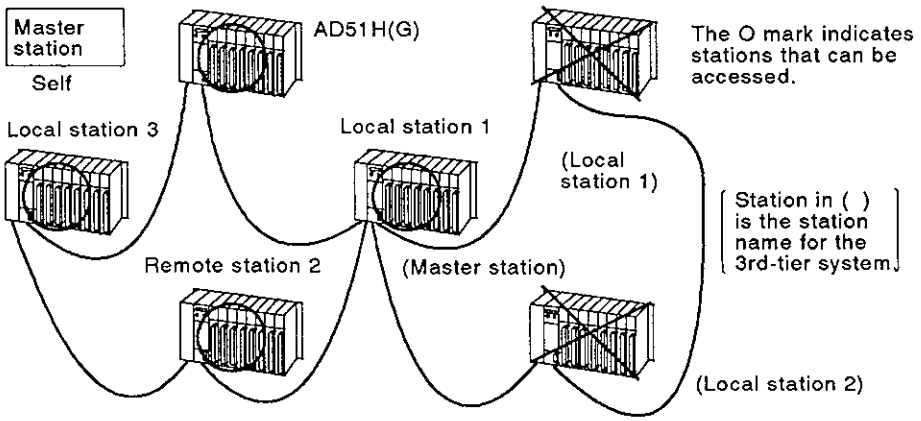
4) Read and write of a sequence program and a parameter

5) Interruption to a PC CPU

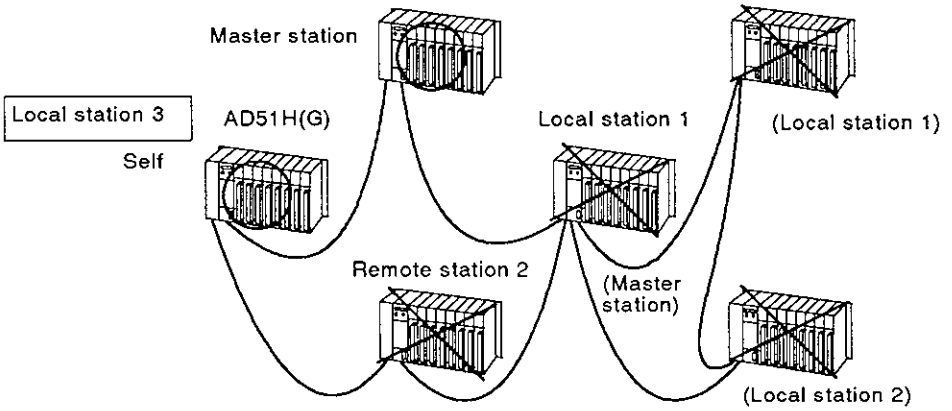
(b) Data communications with a PC on MELSECNET can be executed between the following stations:

Master station: Data communications with a local station and a remote I/O station is enabled.

Local stations: Data communications with a master station is enabled.



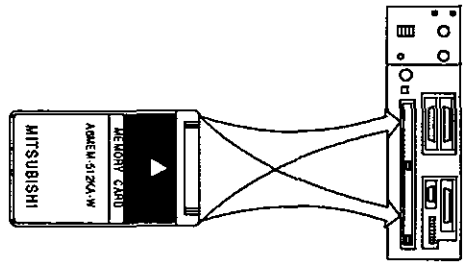
(a) When the AD51H is installed in the master station for the 2nd tier



(b) When the AD51H is installed in a local station for the 2nd tier

(4) File management is enabled.

Sequential files and random files can be managed by using an FD or HD of a PC/AT, or memory card(s) installed in the memory card interface of the AD51H.



When managing a file in a memory card, use the file area in the memory card.

[Sequential file]

A sequential file is efficient when reading or writing data sequentially.

This file is used mainly for work instruction data files and production output achievement files.

[Random file]

Data can be read/written from/to any part in this file.

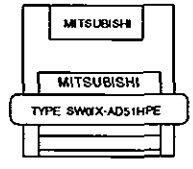
This file is suitable for storing large-capacity inventories, display data for the operator I/F, and master data.

The data size of a one-time read/write is (max.) 256 bytes.

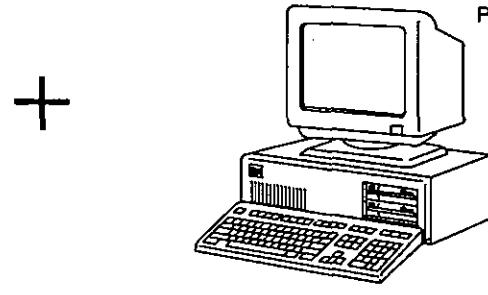
(5) Offline programming using a PC/AT is enabled.

BASIC programs can be edited offline by using a PC/AT.

An edited program cannot be executed by using offline programming.

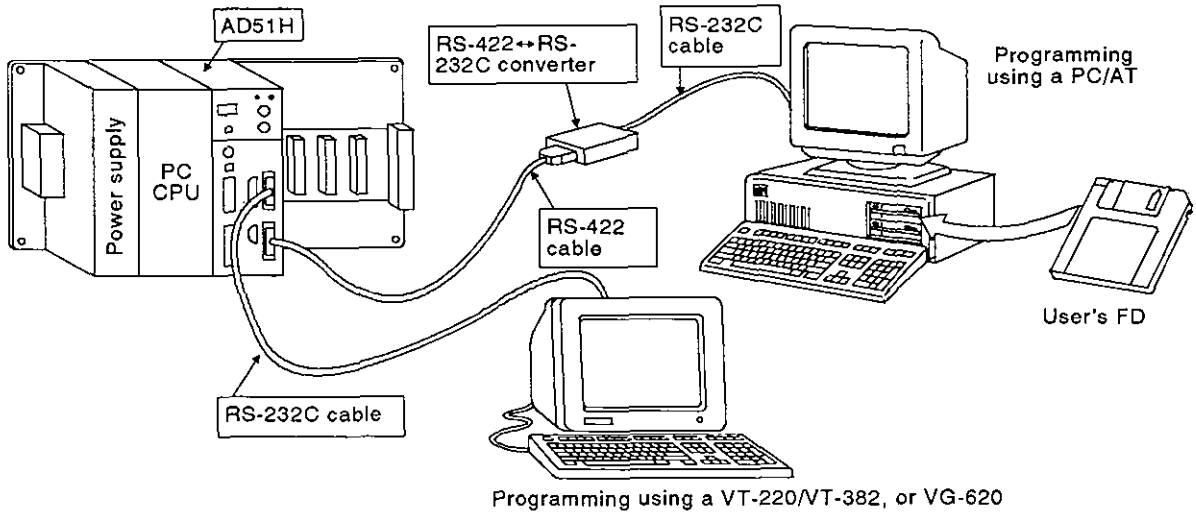


System FD
(Installed in a PC/AT.)



(6) Online programming is enabled.

BASIC programs can be edited and debugged by connecting a PC/AT or device available on the market (see Section 2.1) to the AD51H and setting it as the console.

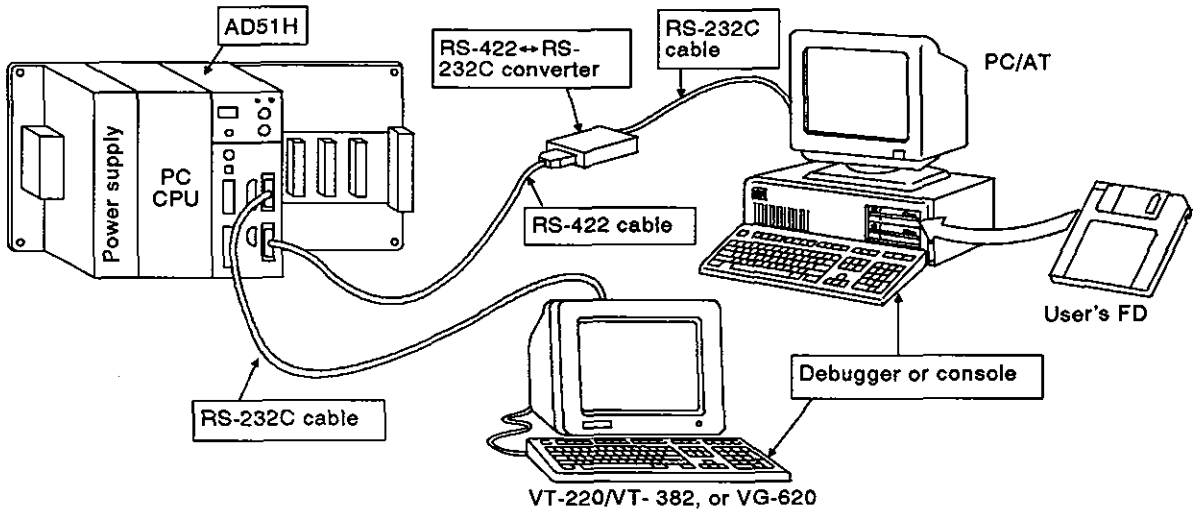


REMARK

Section 4.2.2 tells how to set the console.

(7) Multitask debugging is enabled.

Debugging during multitask processing can be executed by connecting a PC/AT or general-purpose device to the AD51H and setting it as the debugger.



Multitask debugging is executed by inputting a debug command from the debugger. The following operations can be executed by using a debug command:

- Executing and stopping the BASIC program in a specified task number area
- Read/write of the value of a specified variable from/to the BASIC program in a specified task number area
- Read/write of data from/to the internal memory of the AD51H
- Display of the state of use of events and message ports being shared between BASIC programs

The online programming (debug) part of this manual gives details about debug commands.

REMARK

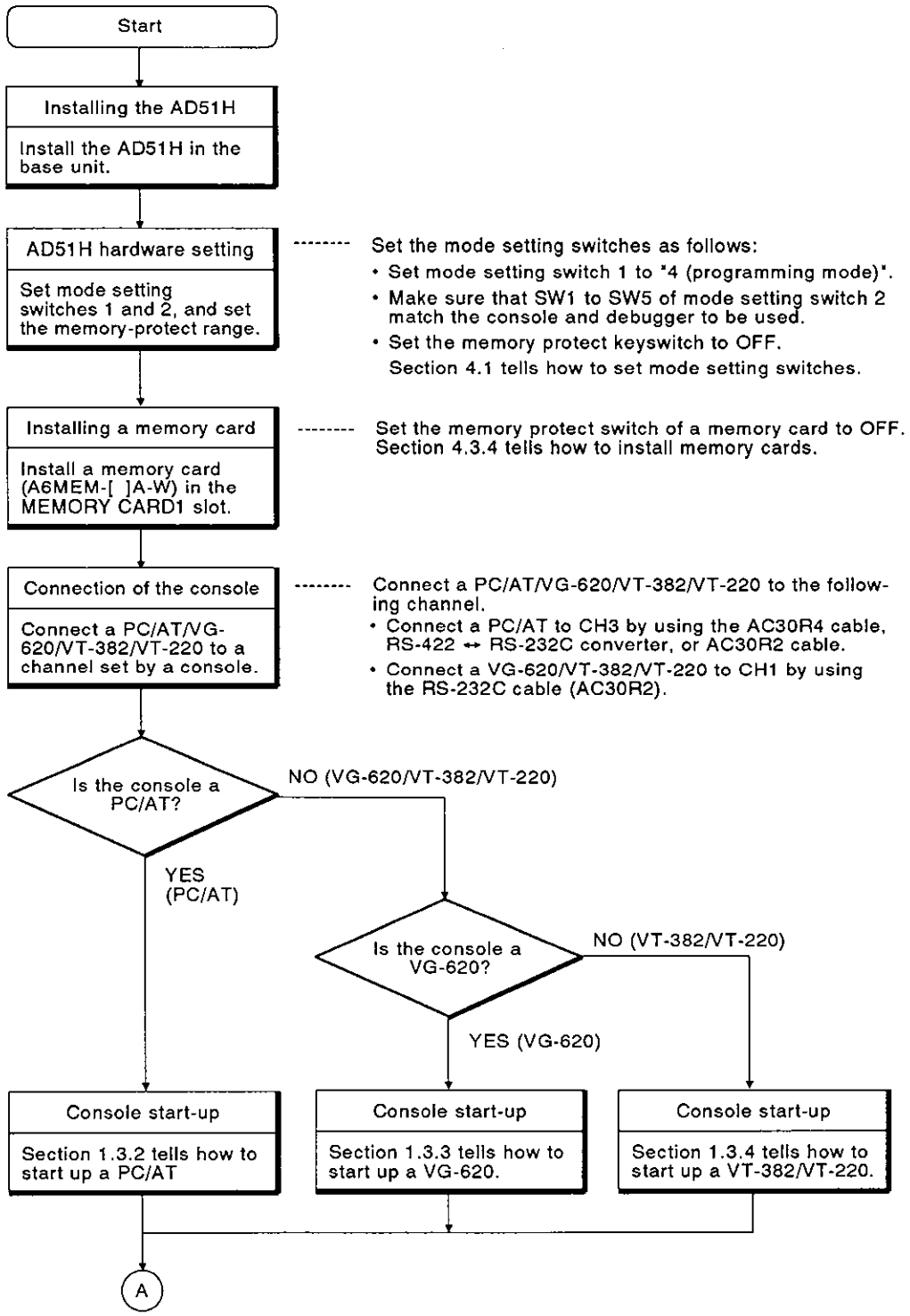
(1) Section 4.2.2 tells how to set the debugger.

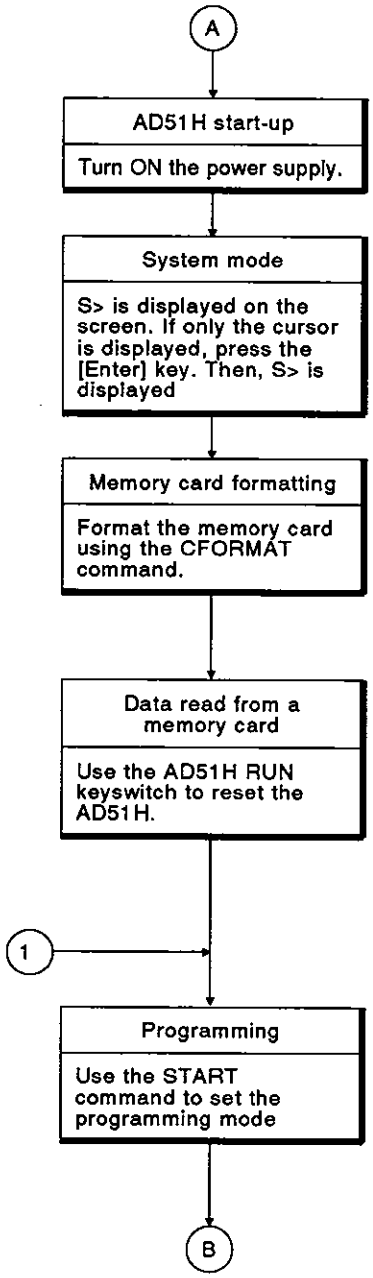
1.3 Procedure for Executing BASIC Programs in the Multitasking Mode

This section tells how to execute BASIC programs in the multitasking mode.

The manual or section of each item gives details about the setting operation of each item.

1.3.1 Procedure for executing BASIC programs in the multitasking mode





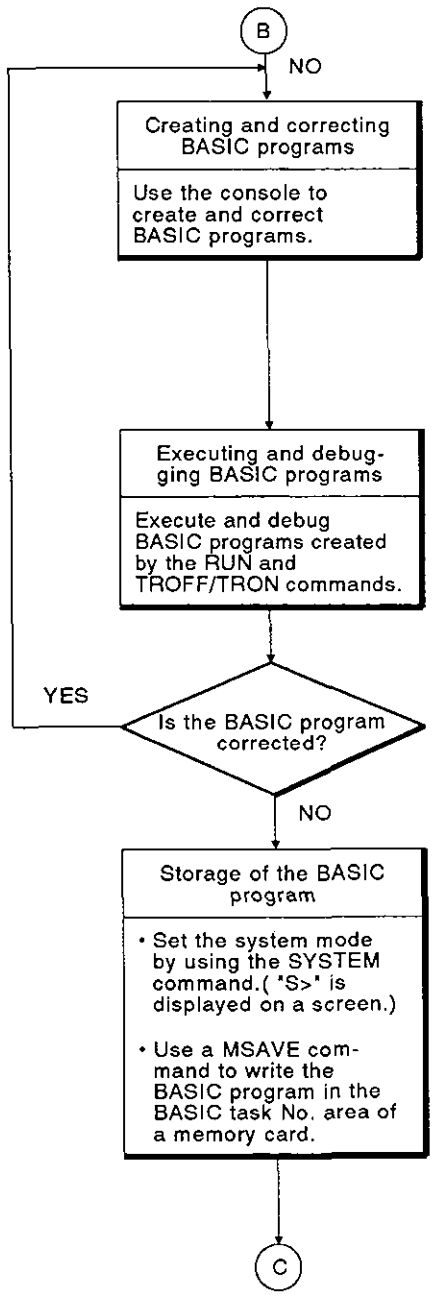
--------- Operation by a system command is enabled on the console.

--------- Use the CFORMAT command to set the following:
- Capacity of a memory card
- Capacity of the execution program area
- Capacity of the file area

Section 3.2.2 of the online programming and debugging part gives details about the CFORMAT command.

--------- Make the capacity allocation of the AD51H's program area the same as that of the memory card. (If the AD51H is reset using the RUN keyswitch of the AD51H, data formatted by the CFORMAT command is read to the AD51H.)

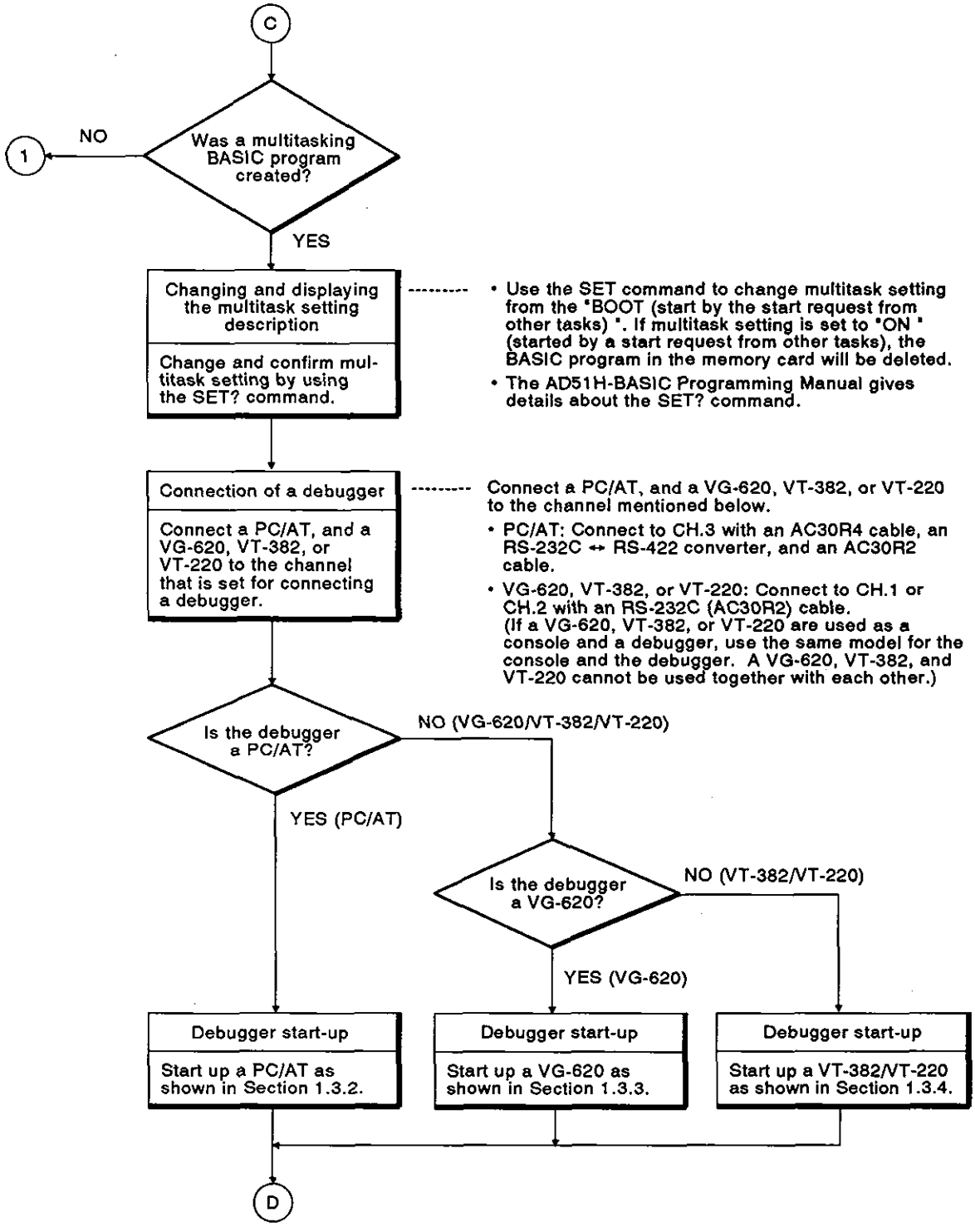
--------- Use the START command to set the BASIC task No. to be created and the program capacity of the BASIC task No.
Section 3.5 (Debugging) gives details about the START command.

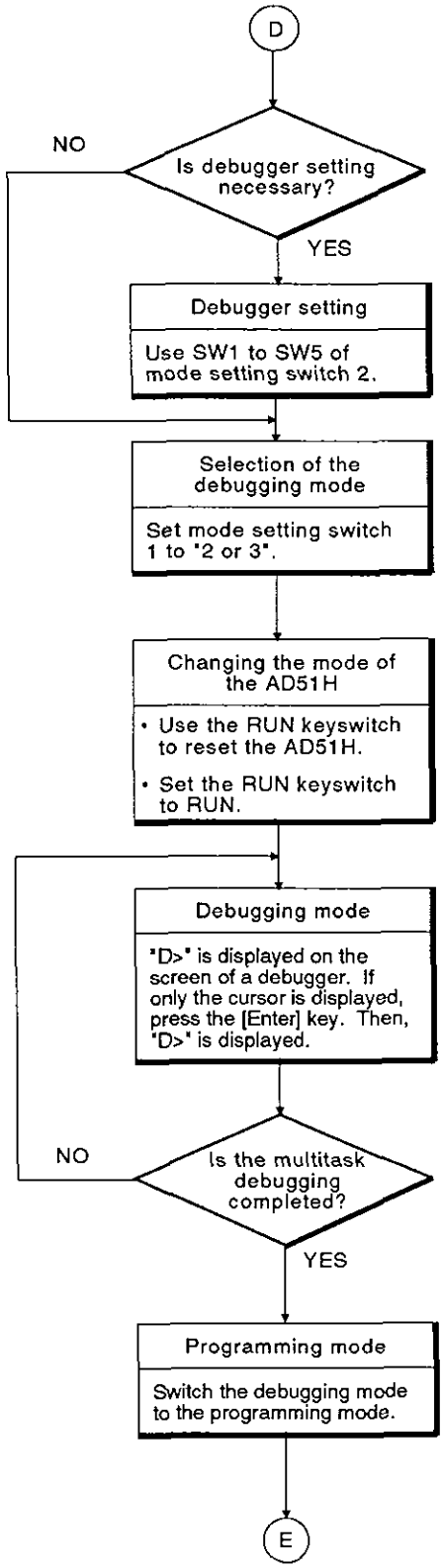


---------- The AD51H-BASIC Programming Manual tells how to create BASIC programs.
- The created program is stored in the program area of the AD51H.

---------- The AD51H-BASIC Programming Manual gives details about the RUN and TROFF/TRON commands.

- The online programming (debug) part of this manual gives details about the SYSTEM and MSAVE commands.
- When writing to the BASIC program is being executed by using the MSAVE command, multitask setting is set to the "BOOT (start by the start request from other tasks) ".


C
Was a multitasking BASIC program created?
NO
1
YES
Changing and displaying the multitask setting description
Change and confirm multitask setting by using the SET? command.
• Use the SET command to change multitask setting from the "BOOT (start by the start request from other tasks) ". If multitask setting is set to "ON " (started by a start request from other tasks), the BASIC program in the memory card will be deleted.
• The AD51H-BASIC Programming Manual gives details about the SET? command.
Connection of a debugger
Connect a PC/AT, and a VG-620, VT-382, or VT-220 to the channel that is set for connecting a debugger.
Connect a PC/AT, and a VG-620, VT-382, or VT-220 to the channel mentioned below.
• PC/AT: Connect to CH.3 with an AC30R4 cable, an RS-232C ↔ RS-422 converter, and an AC30R2 cable.
• VG-620, VT-382, or VT-220: Connect to CH.1 or CH.2 with an RS-232C (AC30R2) cable.
(If a VG-620, VT-382, or VT-220 are used as a console and a debugger, use the same model for the console and the debugger. A VG-620, VT-382, and VT-220 cannot be used together with each other.)
Is the debugger a PC/AT?
NO (VG-620/VT-382/VT-220)
YES (PC/AT)
Is the debugger a VG-620?
NO (VT-382/VT-220)
YES (VG-620)
Debugger start-up
Start up a PC/AT as shown in Section 1.3.2.
Debugger start-up
Start up a VG-620 as shown in Section 1.3.3.
Debugger start-up
Start up a VT-382/VT-220 as shown in Section 1.3.4.
D



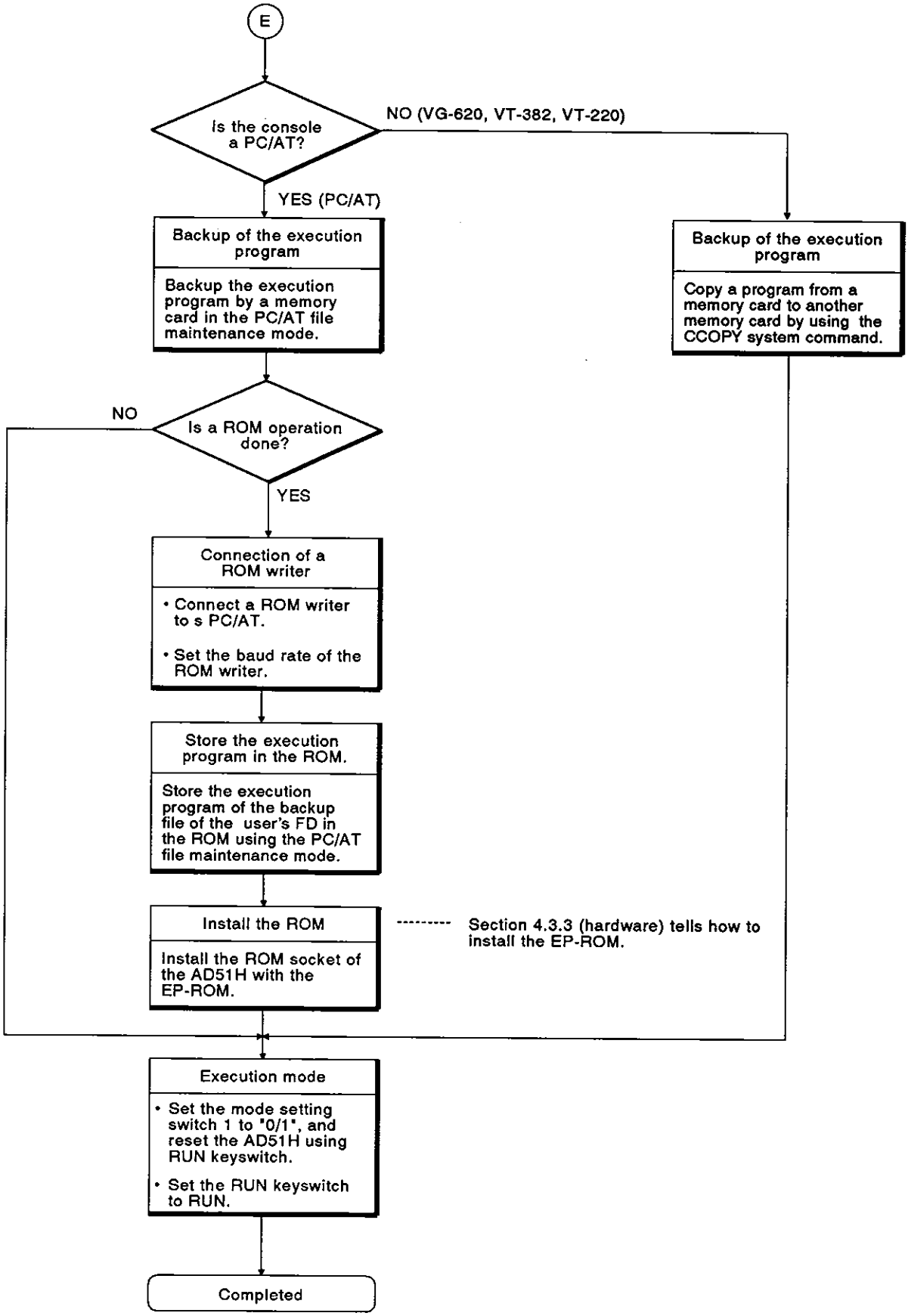
--------- Section 4.2.2 gives details about debugger setting.

--------- Sections 4.2 and 4.2.1 give details about the mode setting switch.

--------- • The AD51H operates in the mode (the mode set by mode setting switch 1) set when the power is turned ON or reset.
• When changing the mode, reset the AD51H, or turn ON the power supply to the AD51H again. (ON → OFF → ON)

--------- Operation by a debug command is enabled on a debugger.
The online programming (debug) part of this manual gives details about the debug command.

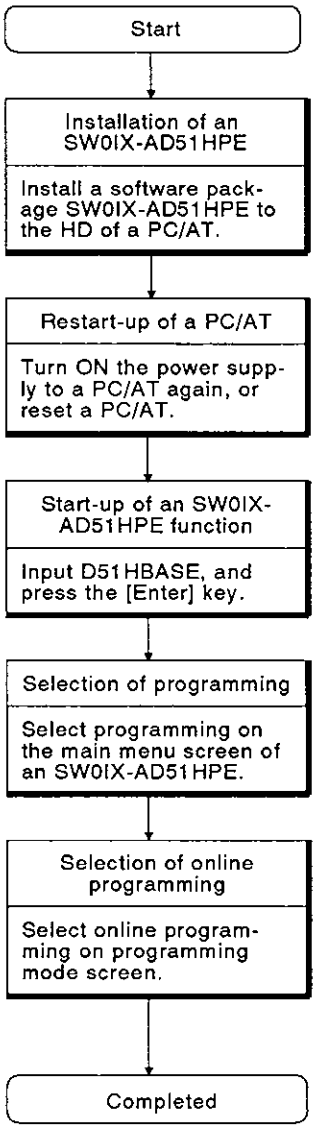
--------- • Input the GO command to a debugger, and switch the mode to the programming mode.
• Set mode setting switch 1 to "4", and reset the AD51H.


E
Is the console a PC/AT?
NO (VG-620, VT-382, VT-220)
YES (PC/AT)
Backup of the execution program
Backup the execution program by a memory card in the PC/AT file maintenance mode.
Backup of the execution program
Copy a program from a memory card to another memory card by using the CCOPY system command.
NO
Is a ROM operation done?
YES
Connection of a ROM writer
• Connect a ROM writer to s PC/AT.
• Set the baud rate of the ROM writer.
Store the execution program in the ROM.
Store the execution program of the backup file of the user's FD in the ROM using the PC/AT file maintenance mode.
Install the ROM
Install the ROM socket of the AD51H with the EP-ROM.
Section 4.3.3 (hardware) tells how to install the EP-ROM.
Execution mode
• Set the mode setting switch 1 to "0/1", and reset the AD51H using RUN keyswitch.
• Set the RUN keyswitch to RUN.
Completed

1.3.2 Procedure for using a PC/AT as the console or debugger

This section tells how to use a PC/AT as a console or a debugger by connecting the PC/AT to the RS-422 interface (CH.3) of the AD51H.

The SW0IX-AD51HPE AD51H-BASIC Operating Manual gives details about the operating procedure.

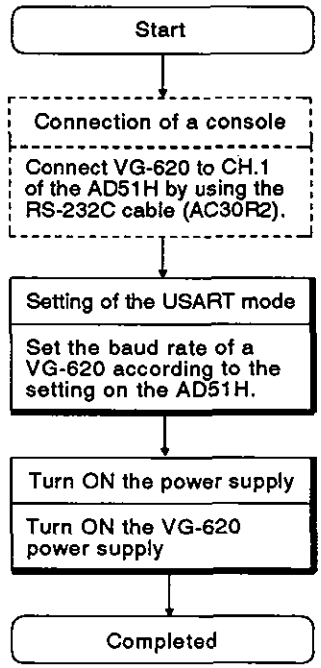


1.3.3 Procedure to use a VG-620 as a console or a debugger

This section tells how to use a VG-620 as a console or a debugger by connecting the VG-620 to the RS-232C interface (CH.1, CH.2) of the AD51H.

When a VG-620 is used as a console, use CH.1.

When a VG-620 is used as a debugger, use CH.1 or CH.2.



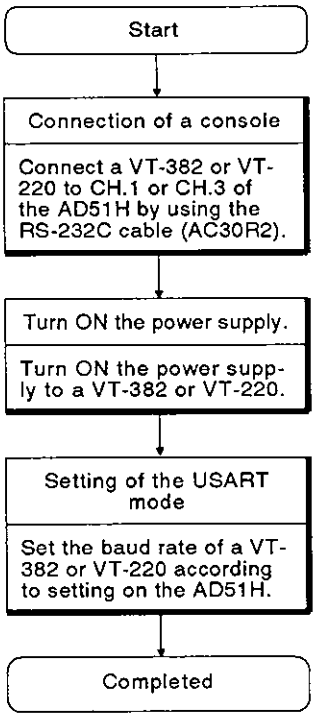
- Section 3.2.2 tells how to set the USART mode of the AD51H.
- The VG-620 Manual gives details about USART mode settings (i.e., for the baud rate of the VG-620).

1.3.4 Procedure to use a VT-382 or VT-220 as a console or a debugger

This section tells how to use a VT-382 or VT-220 as a console or a debugger by connecting a VT-382 or VT-220 to the RS-232C interface (CH.1,CH.2) of the AD51H.

When a VT-382 or VT-220 is used as a console, use CH.1.

When a VT-382 or VT-220 is used as a debugger, use CH.1 or CH.2.



..... • Section 3.2.2 tells how to set the USART mode of the AD51H.
• The VT-382 and VT-220 Manuals give details about USART mode settings (i.e., for the baud rate of a VT-382 or VT-220).

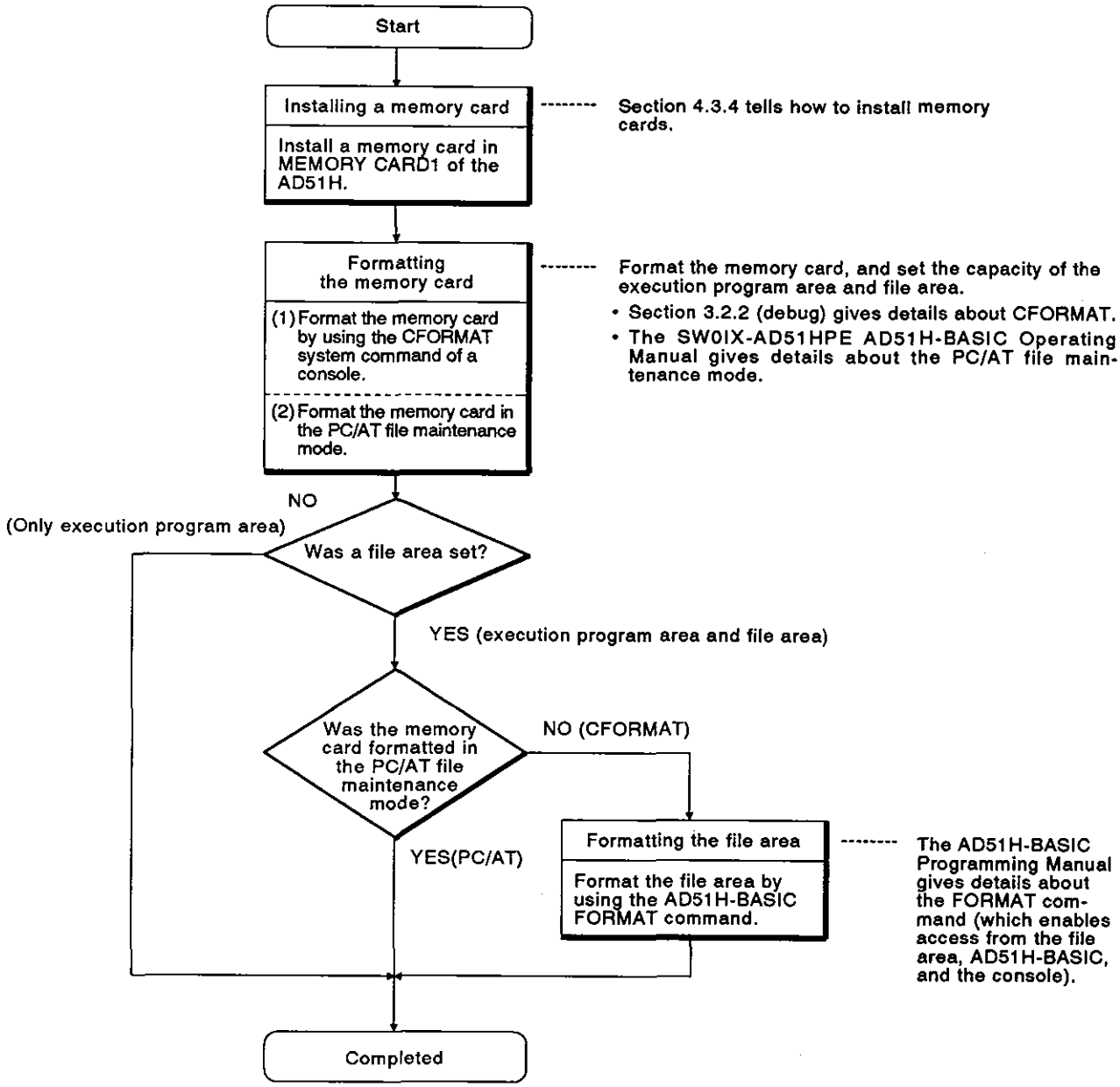
1.3.5 Procedure to use a memory card

This section tells how to install a memory card in the AD51H memory card interfaces (MEMORY CARD1/2).

(1) When formatting a memory card to be installed in MEMORY CARD1

The memory card to be installed in MEMORY CARD1 is used as a file to store the execution program and data.

Format the memory card according to the following procedure:



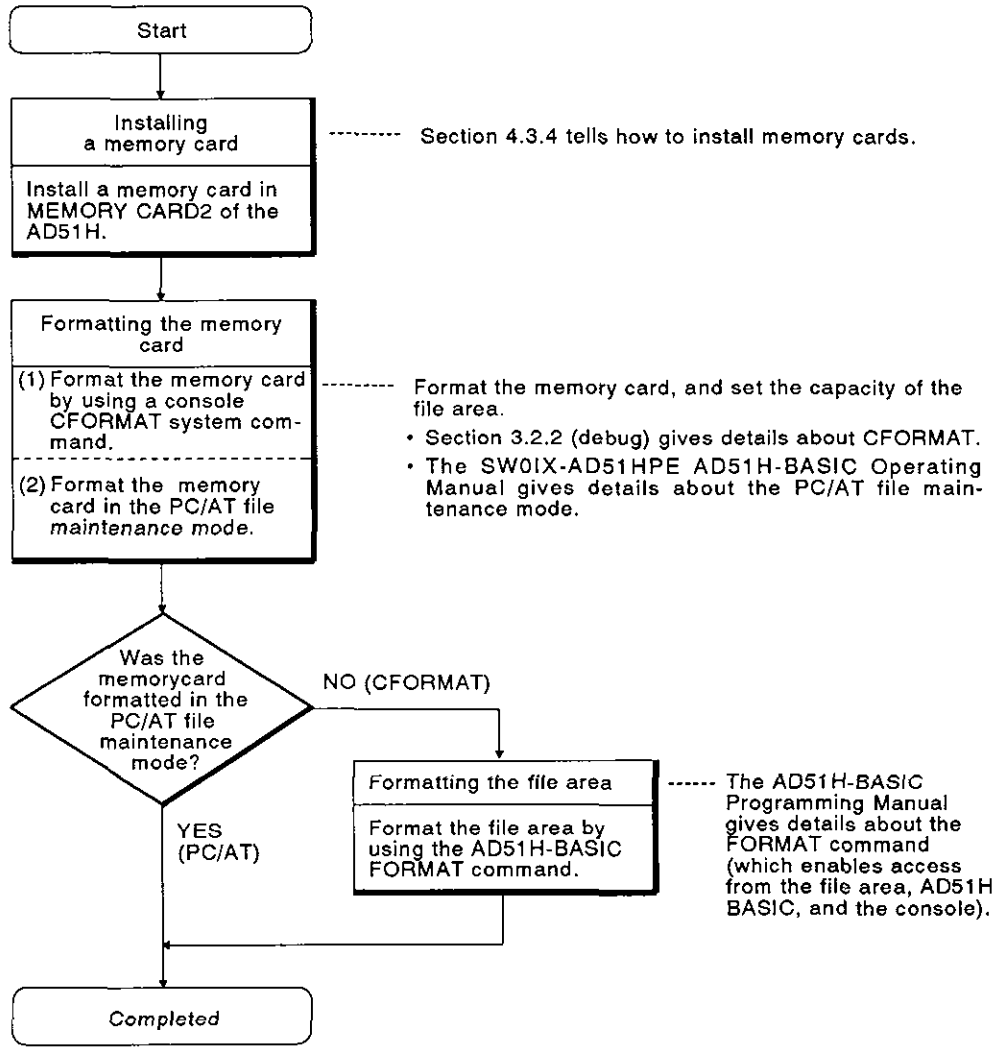
POINT

When a memory card is formatted in the PC/AT file maintenance mode, a file area is formatted simultaneously.

(2) When formatting a memory card to be installed in MEMORY CARD2

The memory card to be installed in MEMORY CARD2 is used as a file to store data.

Format the memory card according to the following procedure:



POINT

When a memory card is formatted in the PC/AT file maintenance mode, *a file area is formatted simultaneously.*

1.4 Basic Information

This section gives basic information for using the AD51H.

1.4.1 When creating BASIC programs

(1) The BASIC system used with the AD51H is AD51H-BASIC.

Since AD51H-BASIC is not compatible with GPC-BASIC used with the AD51(S3), the AD51H cannot use programs created for the AD51(S3).

(The AD51H-BASIC programming Manual gives details about the operations, commands, and functions of the AD51H-BASIC.)

(2) Create BASIC program and debug them using a console (PC/AT, VG-620, VT-382, or VT-220).

Connect a PC/AT to the CH3 RS-422 interface via the RS-422 ↔ RS-232C converter. Connect a VG-620, VT-382, or VT-220 to the CH1 RS-232C interface.

When a console is used, console setting must be executed using mode setting switch 2.

(Section 4.2.2 tells how to perform console setting using mode setting switch 2.)

(3) The internal memory of the AD51H is not battery-backed.

To turn OFF the power supply:

Turn OFF the power supply after storing the BASIC program created by the console to the user's FD, memory card or the HD of a PC/AT.

Use the following commands to store BASIC programs:

- To store to a memory card:
  - MSAVE system command (stores in the execution program area storage)
  - AD51H-BASIC SAVE command (stores in a file)
- To store it to the user's FD or a HD:
  - AD51H-BASIC SACE command

Section 3.3.2 (debug) gives details about the MSAVE command.

The AD51H-BASIC Programming Manual gives details about the SAVE command.

(4) When writing BASIC programs to the execution program area of a memory card by using a MSAVE command, the condition for starting multitask setting is set to "BOOT".

Use the SET system command to change the multitask setting.

Section 3.4.1 (debug) gives details about SET commands.

(5) The following four commands can be set to start multitask processing:

- START command : • Started by multitask execution
- IT : • Started by a start request from the PC CPU
- BOOT : • Started by a start request from other tasks (A BASIC progaram is stored.)
- ON : • Started after reading the BASIC program from a file area by a start request given by an other task to the AD51H. (Only area setting is executed, and the BASIC program is not stored.)

If a task (in which the BASIC program was stored in a memory card by using the MSAVE command) is set to ON, the BASIC program of the turned-on task will be deleted from the memory card.

(6) To access a memory card with the BASIC program:

Access the memory card after confirming that the memory card access switch is turned ON.

The ON/OFF state of the memory card access switch can be confirmed by the ON/OFF state of the EM9021 special relay.

(Section 6.7 (control functions) gives details about special relays.)

1.4.2 Using memory cards

(1) To install or remove a memory card when the power supply is turned ON:

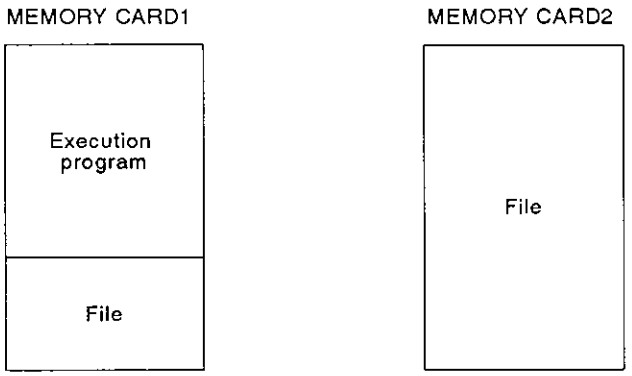
Install or remove the memory card after setting the memory card access switch to OFF (which displays OK on the LED indicator).

Section 4.3.4 (hardware) tells how to install and remove memory cards.

(2) The memory card is used (a) for the execution program area in which the BASIC program to be executed by the multitask is stored, and (b) the file area in which data is stored.

The memory card interfaces of the AD51H are MEMORY CARD1 and MEMORY CARD2.

- MEMORY CARD1 is used to store the execution program and files.
- MEMORY CARD2 is only used to store files.



(3) Any memory card which is used must be formatted.

- When only the execution program area is formatted:

  Use the CFORMAT system command or the PC/AT file maintenance mode.

- When both the execution program area and file area are formatted:

  Use the CFORMAT system command and the AD51H-BASIC FORMAT command, or the PC/AT file maintenance mode.

- When only the file area is formatted:

  Use the CFORMAT system command and the AD51H-BASIC FORMAT command, or the PC/AT file maintenance mode.

Section 3.2.2 (debug) gives details about the about the CFORMAT command.

The SW0IX-AD51HPE AD51H-BASIC Operating Manual gives details about the PC/AT file maintenance mode.

(4) If any of the following happens while data is being written to a memory card, the file where the data is stored may be lost:

(a) The AD51H is reset.

(b) A PC CPU is reset.

(c) Momentary power failure (20 msec or longer) occurs.

(d) The power supply to the PC CPU goes OFF.

The following corrective actions must be done to prevent any of these situations from occurring when writing data to the memory card with the BASIC program.

(a) Resetting the AD51H

To reset the AD51H using the RUN keyswitch:

Reset the AD51H after turning OFF the memory card access switch (which displays OK on the LED indicator).

(b) Resetting a PC CPU

To reset a PC CPU using the RUN keyswitch

- Reset a PC CPU after turning OFF a memory access switch of the AD51H (which displays OK on the LED indicator).
- Set mode setting switch 2 of the AD51H so that the AD51H can continue operating even if a PC CPU is reset.

(Section 4.2.2 (hardware) tells how to set mode setting switch 2.)

(c) Occurrence of a momentary power failure (20 msec or longer)

If a momentary power failure (20 msec or longer) occurs, the AD51H interrupts multitask processing, and the state of the AD51H is switched to the initial start state (the same as when the power supply was turned ON).

Take corrective actions to make sure that momentary power failures (20 msec or longer) do not occur.

(d) Turning OFF the power supply to a PC CPU

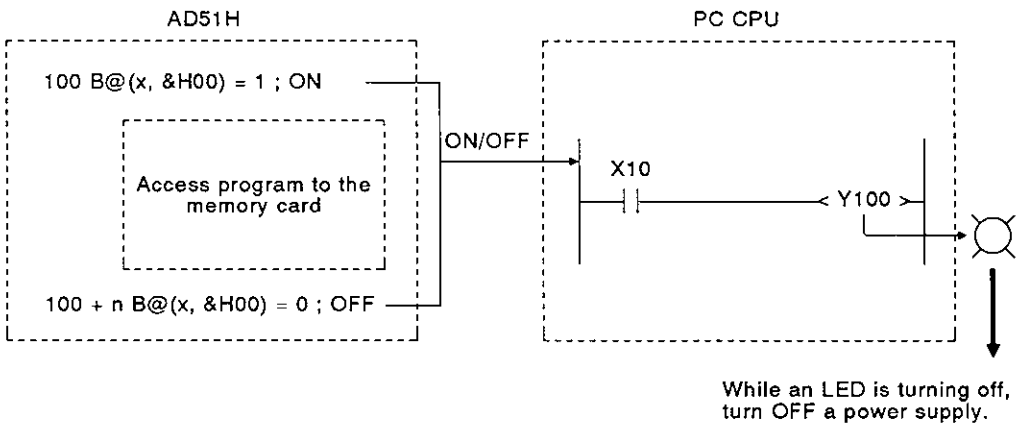
To turn OFF the power supply to a PC CPU:

Turn OFF the power supply after confirming that the AD51H does not accessing the memory card.

For example, use the flag (general-purpose input) that shows access to the memory card by the AD51H.

Turn ON the flag before accessing a memory card. Then, turn OFF the flag by completing the access to the memory card.

The PC CPU uses the flag that shows access to the memory card as a contact and outputs the signal. When a memory card is accessed by the AD51H, an LED goes ON. When the LED goes off, turn OFF the power to the PC CPU.



1.4.3 When performing the operation using the ROM

(1) Use a ROM writer connected to PC/AT to write an execution program in the ROM.

Section 8.6.1 tells how to write an execution program in the ROM.

(The SW0IX-AD51HPE AD51H-BASIC Operating Manual gives details about operating a PC/AT and setting the ROM writer.)

(2) Use the EP-ROM to be used for read/write of a word unit for the AD51H.

(Section 3.4 (hardware) gives details about EP-ROMs that can be used with the AD51H.)

(3) When inserting the EP-ROM into a ROM socket, install the EP-ROM securely.

And, when removing the EP-ROM from the ROM socket, use an IC extraction tool (for 40 pins).

(Section 4.3.3 (hardware) tells how to install and remove the EP-ROM.)

(4) When the operation by ROM is executed, set the EP-ROM type and the read destination of the execution program by mode setting switch 2.

(Section 4.2.2 (hardware) tells how to set mode setting switch 2.)

1.4.4 When performing multitask debugging

(1) PC/AT, VG-620, VT-382, or VT-220 can be used as a debugger to execute multitask debugging.

Connect PC/AT to the RS-422 interface of CH.3 via the RS↔422-RS-232C converter, and connect VG-620, VT-382, or VT-220 to the RS-232 interface of CH.1.

When a console or a debugger is used, console setting or debugger setting by mode setting switch 2 is necessary.

(Section 4.2.2 tells how to set mode setting switch 2.)

(2) A console and a debugger can be combined in the following six ways:

[Console]	[Debugger]
• PC/AT	VG-620
• PC/AT	VT-382 (VT-220)
• VG-620	PC/AT
• VG-620	VG-620
• VT-382 (VT-220)	PC/AT
• VT-382 (VT-220)	VT-382 (VT-220)

(3) When the RUN keyswitch is set to RUN, multitask debugging is valid.

1.4.5 When using a printer connected to PC/AT

When a printout is executed by using a printer connected to PC/AT, perform printer setting by the following commands to the AUTOEXEC.BAT file in MS-DOS.

• When the RS-232C printer is used

MODE COM1:(baud rate), (parity), (data length), (stop bit)

• When a parallel printer is used

MODE LPT1:

1.4.6 When executing BASIC programs in the multitasking mode

(1) When executing BASIC programs in the multitasking mode, if (a) execution of the BASIC program is stopped by inputting the [Ctrl] + [C] keys, and (b) the mode is switched to the system mode because an error occurs, the console will sometimes not display a normal message.

Use the LED indicator on the AD51H to confirm the mode or error contents.

(2) When execution of a BASIC program uses sources (data in external devices, memory cards, etc.), more than one BASIC program cannot use (share) the same source simultaneously.

When the BASIC program uses a source, perform exclusive control of a source so that other BASIC programs cannot use the same source.

(Section 6.5.4 (control function) and the AD51H-BASIC Programming Manual give details about the exclusive control of a source.)

(3) The following processings cannot be executed simultaneously from several tasks during communications between an AD51H and a PC/AT.

- Input/output processing as a console
- File management (read/write data from/to the user's FD or the HD of a PC/AT)
- Data printout

(4) When data is received using a ZRESEAV or INPUT command, it can be influenced by noise, etc., sometimes resulting in an error.

Be sure to execute an error processing using the "ON ERROR GO TO" command, and also, a retry processing to receive data.

1.4.7 When using a VG-620, VT-382, or VT-220 as a console or debugger

When a VG-620, VT-382, or VT-220 is used as the console or debugger, the USART mode must be set (so that the console and AD51H baud rates match).

Section 3.2.2 (hardware) gives details about USART mode settings.

In addition, the VG-620, VT-382, and VT-220 manuals give details about USART mode settings for the VG-620, VT-382, and VT-220.

1.5 Terms

1.5.1 Formal names and abbreviations of CPU model names

The CPU model names used in this manual are given in Table 1.1.

Table 1.1 Formal Names and Abbreviations of CPU Model Names

Formal Name		Model Names	Abbreviations
PC CPU	Compact-type CPUs	A0J2CPU(P23/R23)	A0J2
		A0J2HCPU(P21/R21)	A0J2H
	Building block-type CPUs	A1CPU(P21/R21)	A1
		A1NCPU(P21/R21)	A1N
		A2CPU(P21/R21)	A2(-S1)
		A2CPU(P21/R21)-S1	
		A2NCPU(P21/R21)	A2N(S1)
		A2NCPU(P21/R21)-S1	
		A2ACPU(P21/R21)	A2A(S1)
		A2ACPU(P21/R21)-S1	
		A3CPU(P21/R21)	A3
		A3NCPU(P21/R21)	A3N
		A3ACPU(P21/R21)	A3A
		A3HCPU(P21/R21)	A3H
		A3MCPU(P21/R21)	A3M
		A73CPU(P21/R21)	A73
		A1SCPU	A1S

1.5.2 Terms

[Console]

Peripheral devices (PC/AT, VG-620, VT-382, or VT-220) connected to an interface set as the console using SW1 to SW5 of mode setting switch 1

When using a PC/AT, execution of the BASIC program can be stopped by inputting (a) the [BREAK] key, or (b) the [Ctrl] + [C] keys.

When using a VG-620, VT-382, or VT-220, execution of the BASIC program can be stopped by inputting the [Ctrl] + [C] keys.

By setting SW6 of mode setting switch 1, the BASIC program cannot be stopped, even if the [BREAK] key or the [Ctrl] + [C] keys are input.

[Terminal]

A peripheral device (PC/AT, VG-620, VT-382, or VT-220) connected to the interface that is not set for a console by SW1 to SW5 of mode setting switch 1

Even if the [BREAK] key or the [Ctrl] + [C] keys are input, execution of the BASIC program cannot be stopped.

Pressing the [BREAK] key or the [Ctrl] + [C] keys is processed as a 03H code.

[Debugger]

A peripheral device (PC/AT, VG-620, VT-382, or VT-220) connected to an interface can be set as a debugger by SW1 to SW5 of mode setting switch 1

[HARDWARE]

CONTENTS

2. SYSTEM CONFIGURATIONS

This section explains the system configurations and system devices for use with the AD51H.

2.1 PC CPU Systems Used With the AD51H

This section explains PC CPU systems, applicable systems, and precautions which must be taken when using the AD51H.

Either a building block-type CPU or a compact-type CPU can be used as a PC CPU.

REMARK

Section 1.5.1 gives the model names of building block-type and compact-type CPUs.

2.1.1 Overall configurations

(1) When using a building block-type CPU

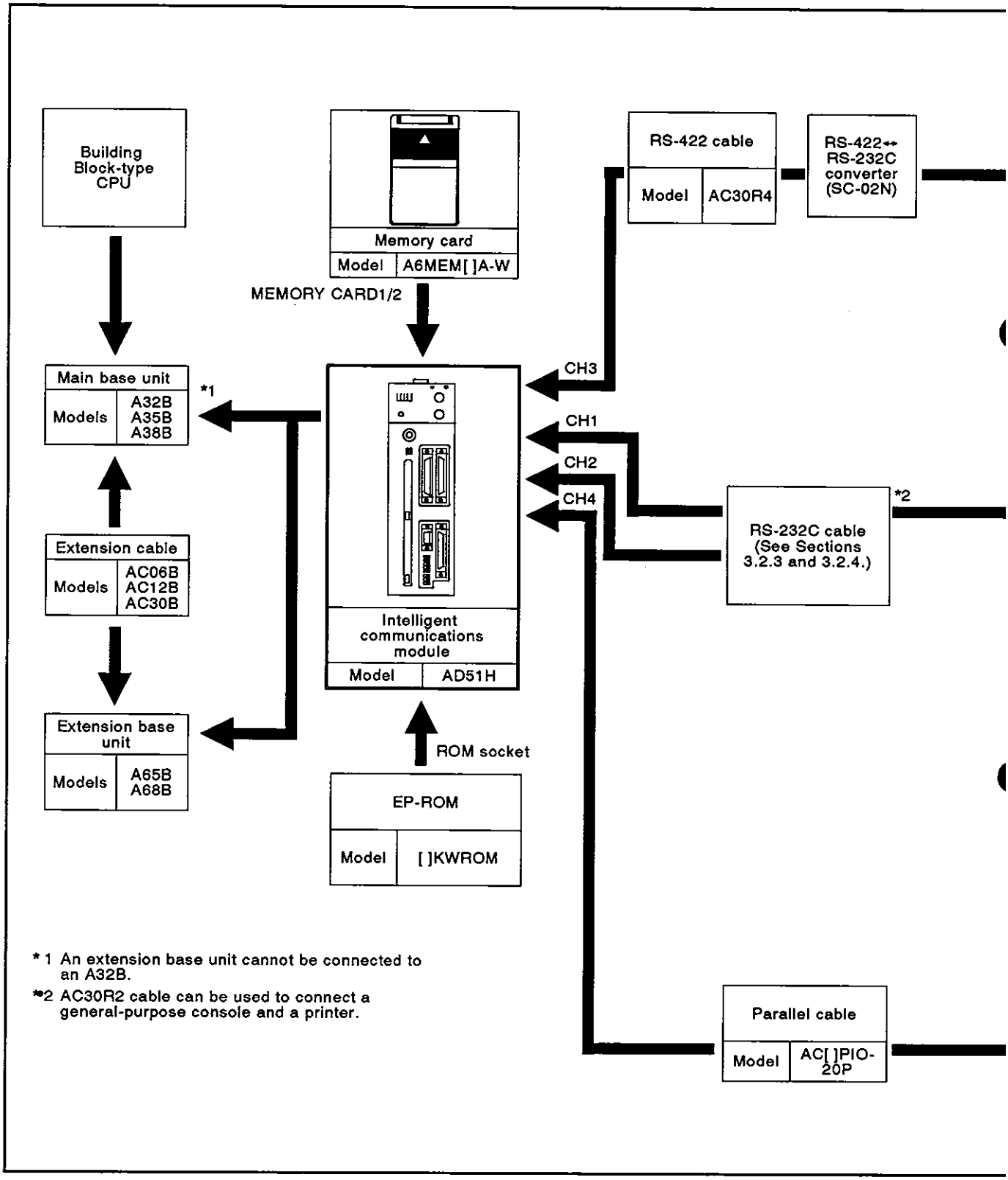
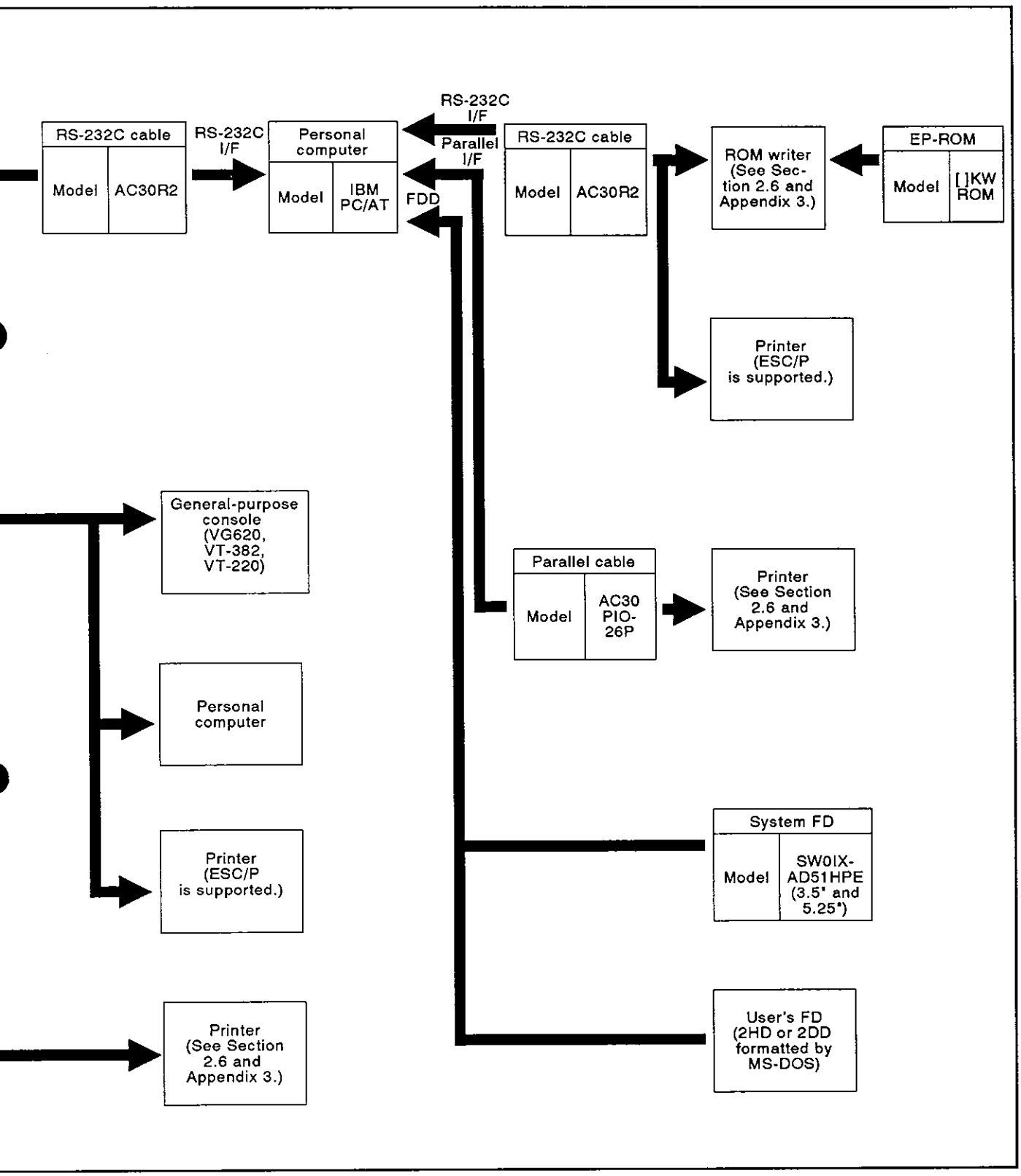
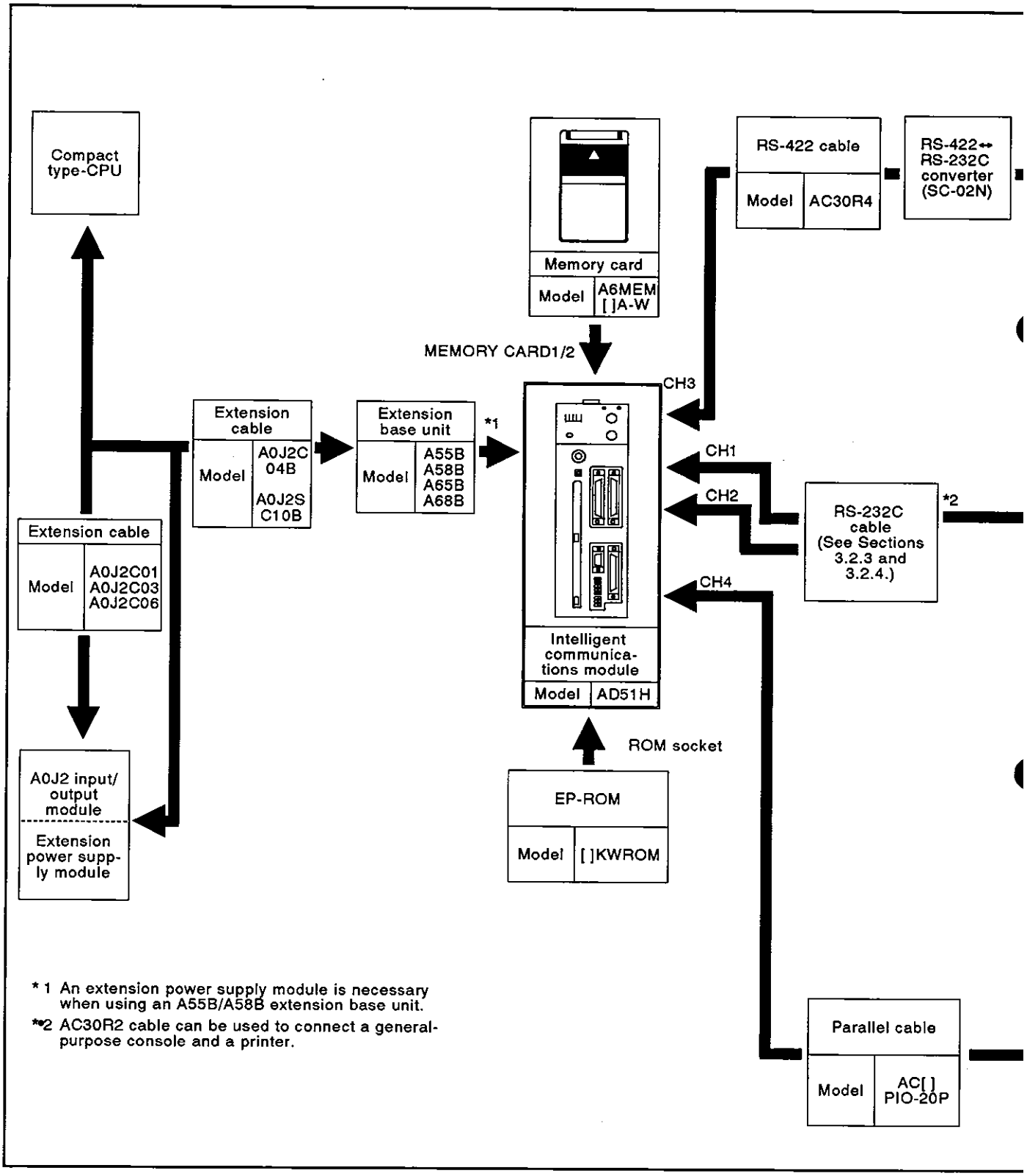


Fig. 2.1 Building Block-Type CPU Overall Configuration


RS-232C cable
Model
AC30R2
RS-232C I/F
Personal computer
Model
IBM PC/AT
RS-232C I/F
Parallel I/F
FDD
RS-232C cable
Model
AC30R2
ROM writer (See Section 2.6 and Appendix 3.)
EP-ROM
Model
[ ]KW ROM
Printer (ESC/P is supported.)
General-purpose console (VG620, VT-382, VT-220)
Parallel cable
Model
AC30 PIO-26P
Printer (See Section 2.6 and Appendix 3.)
Personal computer
System FD
Model
SW0IX-AD51HPE (3.5" and 5.25")
Printer (ESC/P is supported.)
Printer (See Section 2.6 and Appendix 3.)
User's FD (2HD or 2DD formatted by MS-DOS)

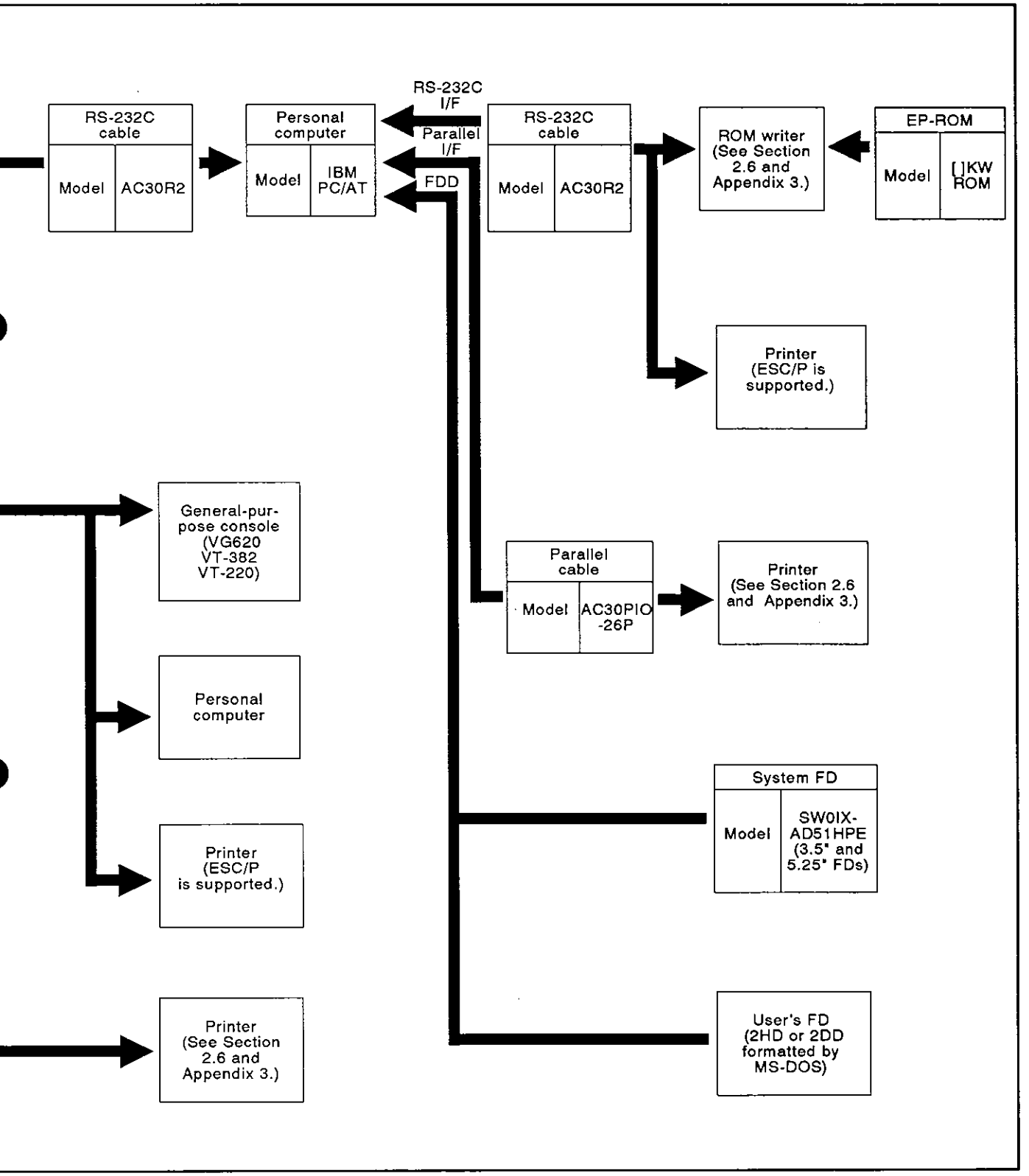
(2) When using a compact type-CPU



*1 An extension power supply module is necessary when using an A55B/A58B extension base unit.

*2 AC30R2 cable can be used to connect a general-purpose console and a printer.

Fig. 2.2 Compact-Type CPU Overall Configuration


RS-232C cable
Model
AC30R2
Personal computer
Model
IBM PC/AT
RS-232C I/F
Parallel I/F
FDD
RS-232C cable
Model
AC30R2
ROM writer (See Section 2.6 and Appendix 3.)
EP-ROM
Model
[ ]KW ROM
Printer (ESC/P is supported.)
General-purpose console (VG620 VT-382 VT-220)
Parallel cable
Model
AC30PIO-26P
Printer (See Section 2.6 and Appendix 3.)
Personal computer
System FD
Model
SW0IX-AD51HPE (3.5" and 5.25" FDs)
Printer (ESC/P is supported.)
Printer (See Section 2.6 and Appendix 3.)
User's FD (2HD or 2DD formatted by MS-DOS)

2.1.2 Applicable CPU modules

The AD51H can be used with the CPUs given below.

(1) Applicable CPU modules and the number of the AD57G modules

The table below shows the PC CPU modules to which the AD51H is applicable and the number of the AD51H modules connectable to the PC CPU modules.

The CPU modules listed below include those which have the MELSECNET link function.

• Example : A1CPU shown below indicates that A1CPUP21 and A1CPUR21 can also be loaded with AD51Hs.

CPU Module	Number of Connectable AD51Hs	Note
A0J2H A1, A1N, A1S A2(-S1), A2N(-S1) A3, A3N A3H, A3M A73	2	The following types of modules should be taken into the number when they are used together with AD51H. • A0J2-C214S1 Computer Link/Multidrop Link Module (only when the computer link function is used) • AD51(S3) Intelligent Communication Module • AJ71C21(S1) Terminal Interface Module (only when the BASIC program mode is used) • AJ71C22(S1) Multidrop Link System Module • AJ71C23 Higher Controller High-Speed Link Module • AJ71C24(-S3/S6) Computer Link Module • AJ71E71 Ethernet Interface Module
A2A(-S1) A3A	6	

(A2CCPU is not applicable.)

(2) Applicable base unit

The AD51H can be loaded to any slot of the main base unit or extension base unit with these two exceptions:

(a) The power supply capacity may be insufficient to load the AD51H to an extension base unit with no built-in power supply (A55B or A58B).

Wherever possible, avoid loading an AD51H module into this type of extension base unit.

If it is necessary to use an AD51H module on an extension base unit with no built-in power supply, it is important to consider (a) the power supply capacity of the main base unit and (b) the voltage drop along the extension cables when selecting the power supply module and the extension cables.

(For details, see the User's Manual of respective CPU module to be used.)

(b) The AD51H must not be loaded to the last slot of the A3CPU's 7th extension base unit.

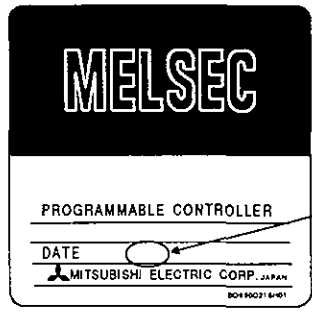
(c) When an A0J2 and an A0J2H are used with an A55B or A58B, an extension power supply module (A0J2PW or A0J2PW-DC24) is required.

POINT

Only PC CPU manufactured after February, 1986 can be used with the AD51H.

Applicability can be confirmed by checking the date area of the PC CPU name plate.

The following shows how to check the PC CPU name plate:



PC CPU that can be used with the AD51H

603

This shows it was manufactured in March, 1986.

2.2 List of System Devices

Table 2.1 gives the devices needed to use the AD51H.

The particular device manual gives details about each device.

(1) Device that can connected to the AD51H

Table 2.1 shows the MELSEC-A series standard devices, recommended devices, and other devices which can be connected to the AD51H.

Table 2.1 shows the applicability of each device in regard to the following.

- Online : Online creation of BASIC programs
- Offline : Offline creation of BASIC programs
- Debug : Multitask debug of BASIC programs
- Execution : Multitask execution of BASIC programs

Table 2.1 List of Devices That Can be Connected to the AD51H

●: Necessary devices, O: Optional devices

Product Name		Model	Remarks			System Configuration			
						Off line	On-line	De bug	Exe cution
Intelligent communications module		AD51H	Main module			—	●	●	●
Memory card		—	Memory capacity	Use		—	●	●	●
		A6MEM-128KA-W	256K bytes	Execution program and file storage					
		A6MEM-256KA-W	512K bytes						
EP-ROM		64KROM-W	128K bytes	Operation by ROM Write to the ROM using the ROM writer connected to a PC/AT.		—	—	—	O
		128KROM-W	256K bytes						
		256KROM-W	512K bytes						
Personal computer		IBM PC/AT	Used as a console, a debugger or a terminal.			●	●	●	O
General-purpose consoles		VG-620				—			
		VT-382							
		VT-220							
Printer	RS-232C	ESC/P is supported.	This is used to print out BASIC programs and data. (Appendix 3 gives details about specifications and connection cables.)			—	O	O	O
	Parallel	M6265-1							
		VP-1500							
		VP-3000							
Connection cable		—	Classification	Cable length	Connection device	—	—	—	—
		AC30R4	RS-422	3 meters	PC/AT *1	—	●	●	—
		AC30R2	RS-232C	3 meters	Printer	—	O	O	O
					VG-620/VT-382/ VT-220	—	●	●	—
		AC30PIO-20P	Parallel cable	3 meters	Printer	—	O	O	O

REMARK

*1 Connect the AD51H and PC/AT via the RS-422 ↔ RS-232C converter.

(2) Devices that can connected to a PC/AT

Table 2.2 shows the MELSEC-A standard and other devices which can be connected to a PC/AT.

Table 2.2 classifies connection devices as follows:

- AD51H
- Printer R: Printer with an RS-232C interface
- Printer P: Printer with a parallel interface
- ROM writer

Table 2.2 List of Devices That Can Connected to a PC/AT

● : Necessary devices, O: Optional devices

Product Name		Model	Remarks	Connection Device			
				AD51H	Printer R	Printer P	ROM Writer
Personal computer		PC/AT	Main module	●	●	●	●
Operating system		MS-DOS	Version 3.21 or later	●	●	●	●
Applicable versions		SW0IX-GPPAE--1 to 4	GPP function				
		SW0IX-AD51HPE	Used with the AD51H	●	●	●	●
User's FD		—	2DD or 2HD formatted by MS-DOS				●
Printer	RS-232C	ESC/P is supported.	This is used to print BASIC programs and data. (Appendix 3 gives details about specifications and connection cables.)		●		
	Parallel	M6265-1				●	
		VP-1500					
		VP-3000					
ROM writer		PECKER11	Writing to the execution program ROM (Memory card→EP-ROM)				●
		PECKER30					
EP-ROM		64KROM-W	128K bytes				●
		128KROM-W	256K bytes				
		256KROM-W	512K bytes				
Cable		AC30R4	RS-422 cable, 3 meters	●			
		AC300R4	RS-422 cable, 30 meters	O			
		AC30R2	RS-232C cable, 3 meters		●		●
		AC30PIO-26P	Parallel cable, 3 meters			●	
		Power supply cable	Power supply cable of PC/AT	●	●	●	●

3. SPECIFICATIONS

3.1 General Specifications

Table 3.1 shows the general specifications.

Table 3.1 General Specifications

<table>
<tr><th>Item</th><th colspan="5">Specifications</th></tr>
<tr><td>Operating ambient temperature</td><td colspan="5">0 to 55°C</td></tr>
<tr><td>Storage ambient temperature</td><td colspan="5">–20 to 75°C</td></tr>
<tr><td>Operating ambient humidity</td><td colspan="5">10 to 90% RH, no condensation allowed</td></tr>
<tr><td>Storage ambient humidity</td><td colspan="5">10 to 90% RH, no condensation allowed</td></tr>
<tr><td rowspan="3">Vibration resistance</td><td rowspan="3">Conforms to *JIS C 0911</td><td>Frequency</td><td>Acceleration</td><td>Amplitude</td><td>Sweep Count</td></tr>
<tr><td>10 to 55 Hz</td><td>—</td><td>0.075 mm (0.003 inch)19 55 to 150 Hz</td><td rowspan="2">10 times *(1 oc-tave/minute)</td></tr>
<tr><td>55 to 150 Hz</td><td>1 g</td><td>—</td></tr>
<tr><td>Shock resistance</td><td colspan="5">Conforms to JIS C 0912 (10 g x 3 times in 3 directions)</td></tr>
<tr><td>Noise resistance</td><td colspan="5">By noise simulator 1500 Vpp noise voltage, 1 μs noise width and 25 to 60 Hz noise frequency</td></tr>
<tr><td>Dielectric withstand voltage</td><td colspan="5">1500 VAC for 1 minute across AC external terminals and ground
500 VAC for 1 minute across DC external terminals and ground</td></tr>
<tr><td>Insulation resistance</td><td colspan="5">5 MΩ or larger by 500 VDC insulation resistance tester across AC external terminals and ground</td></tr>
<tr><td>Grounding</td><td colspan="5">Class 3 grounding; grounding is not required when it is impossible.</td></tr>
<tr><td>Operating ambience</td><td colspan="5">No corrosive gases or dust.</td></tr>
<tr><td>Cooling method</td><td colspan="5">Self-cooling</td></tr>
</table>

REMARK

One octave marked * indicates a change from the initial frequency to double or half frequency. For example, any of the changes from 10 Hz to 20 Hz, from 20 Hz to 40 Hz, from 40 Hz to 20 Hz, and 20 Hz to 10 Hz are referred to as one octave.

Note: *JIS: Japanese Industrial Standard

3.2 Specifications and Connections of Interfaces for External Devices

The AD51H has the following interfaces for connecting with external devices:

- RS-232C interface : 2 channels
- RS-422 interface : 1 channel
- Parallel interface : 1 channel

Sections 3.2.3 to 3.2.6 give details about the specifications and connections of each interface.

3.2.1 Interface uses

Interfaces connected to external device differ in use according to the device setting and the settings of SW1 to SW5 of mode setting switch 2.

Table 3.2 gives interface uses.

Table 3.2 Interface Uses

o: Available x: Not available

Interface Names	Uses			
	Console Port	Debug Port	General-purpose Port	Printer
RS-232C (CH1)	o *2	o *2	o	o
RS-232C (CH2)	x	o *2	o	o
RS-422 (CH3)	o *1	o *1	o	o
Parallel (CH4)	x	x	x	o

*1 PC/AT is applicable.

*2 VT-382, VG-620, or VT-220 are applicable.

(1) Console port

An interface set to a console by SW1 to SW5 of mode setting switch 2 is set to a console port. (See Section 4.2.2)

If execution mode 03H ([Ctrl] + [C] key code) is received by a PC/AT, VT-382, VG-620, or VT-220 connected to a console port, execution of multitasking is stopped.

(2) Debug port

An interface set to a debugger by SW1 to SW5 of mode setting switch 2 is set to a debug port. (See Section 4.2.2)

Multitask debugging can be executed by a PC/AT, VT-382, VG-620, or VT-220 connected to the debug port.

(3) General-purpose port

An interface where console setting by SW1 to SW5 of mode setting switch 2 is not done is set to a general-purpose port.

(See Section 4.2.2.)

Execution of multitasking cannot be stopped even if 03H ([Ctrl] + [C] key code) is received by a PC/AT, VT-382, VG-620, or VT-220 connected to a console port.

(4) Printer

BASIC programs and data can be printed out by connecting a printer.

3.2.2 Available baud rates

The AD51H has the following four channels for connecting with external devices:

- RS-232C interface : 2 channels
- RS-422 interface : 1 channel
- Parallel interface : 1 channel

AD51H can set up to 9600 bps for each channel.

However, if several channels are used, the total (maximum) of the set values of the baud rates is 28.8 KBPS (28800 BPS).

$$(28800\text{BPS}) > \left(\text{Baud rate of CH.1}\right) + \left(\text{Baud rate of CH.2}\right) + \left(\text{Baud rate of CH.3}\right) + \left(\text{Baud rate of CH.4}\right)$$

POINT

(1) When the RS-232C interface is set to a console or debugger, the RS-232C interface of CH.1 or CH.2 is set to the following specifications.

(a) Baud rate : 9600 bps
(b) Data length : 8 bits
(c) Stop bit : 2 bits
(d) Parity : None
(e) Receive buffer size : 1024 bytes

If a VG-620, VT-382, or VT-220 is used as a console or debugger, set the VG-620, VT-382, or VT-220 as given above.

(2) Use the ZCNTL command to set the baud rate and others with BASIC programs.

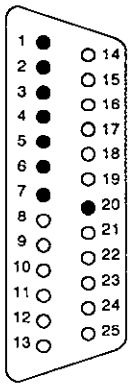
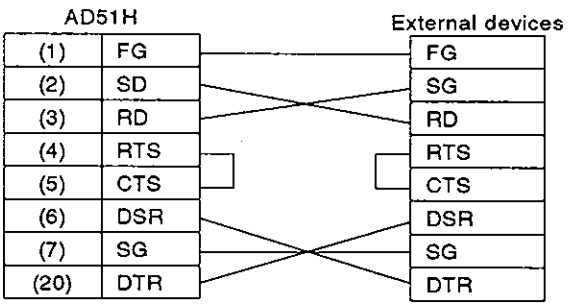
The AD51H-BASIC Programming Manual tells how to use the ZCNTL command.

3.2.3 RS-232C interface (CH1: channel 1)

(1) Interface specifications

Table 3.3 gives the specifications of the RS-232C interface (CH1: channel 1) of the AD51H.

Table 3.3 RS-232C Interface Specifications

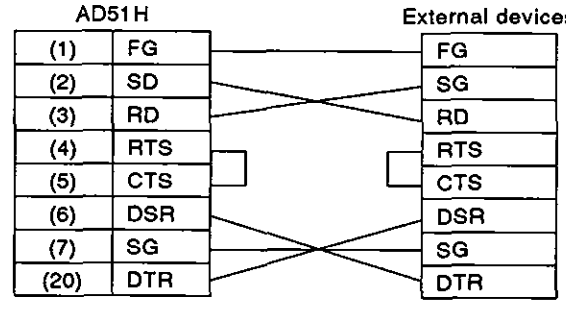
<table>
<tr><th colspan="2">Item</th><th colspan="6">Specifications</th></tr>
<tr><td rowspan="3">Device to be Connected</td><td>Debug port</td><td colspan="6" rowspan="2">VG-620, VT-382, VT-220</td></tr>
<tr><td>Console port</td></tr>
<tr><td>General-purpose port</td><td colspan="6">Computer with the RS-232C interface, personal computer, printer, modem, VG-620, VT-382, or VT-220</td></tr>
<tr><td colspan="2">Transmission method</td><td colspan="6">Conforms to EIA and RS-232C.</td></tr>
<tr><td colspan="2">Transmission speed (bps)</td><td colspan="6">300, 600, 1200, 2400, 4800, or 9600 can be selected.</td></tr>
<tr><td colspan="2">Synchronous method</td><td colspan="6">Asynchronous method</td></tr>
<tr><td rowspan="8">Setting of the USART mode</td><td>Baud rate (bps)</td><td colspan="6">300, 600, 1200, 2400, 4800, or 9600 can be selected.</td></tr>
<tr><td rowspan="2">Parity bit</td><td colspan="6">Without parity</td></tr>
<tr><td colspan="6">With parity : even/odd parity</td></tr>
<tr><td rowspan="2">Stop bit</td><td colspan="6">1 bit</td></tr>
<tr><td colspan="6">2 bit</td></tr>
<tr><td rowspan="2">Character data bit</td><td colspan="6">Data : 7 bits</td></tr>
<tr><td colspan="6">Data : 8 bits</td></tr>
<tr><td>Communications control</td><td colspan="6">Controlled by the DTR terminal</td></tr>
<tr><td rowspan="10">Connector (AD51H side) used</td><td>Model name</td><td colspan="6">17L-10250-27-D9A (DDK Product)</td></tr>
<tr><td rowspan="9">Pin arrangement</td><td>Appearance of connector</td><td>Pin number</td><td>Signal abbreviation</td><td>Signal directions Inside ↔ Outside</td><td colspan="2">General information</td></tr>
<tr><td rowspan="8">  </td><td>1</td><td>FG</td><td></td><td colspan="2">Frame grounding maintenance</td></tr>
<tr><td>2</td><td>SD</td><td>→</td><td colspan="2">Sent data</td></tr>
<tr><td>3</td><td>RD</td><td>←</td><td colspan="2">Received data</td></tr>
<tr><td>4</td><td>RTS</td><td>→</td><td colspan="2">Request to send</td></tr>
<tr><td>5</td><td>CTS</td><td>←</td><td colspan="2">Sending is enabled.</td></tr>
<tr><td>6</td><td>DSR</td><td>←</td><td colspan="2">Data equipment ready</td></tr>
<tr><td>7</td><td>SG</td><td></td><td colspan="2">Signal ground</td></tr>
<tr><td>20</td><td>DTR</td><td>→</td><td colspan="2">Terminal READY notification</td></tr>
<tr><td colspan="2">Connection</td><td colspan="6">  </td></tr>
</table>

3.2.4 RS-232C interface (CH2: channel 2)

(1) Interface specifications

Table 3.4 gives the specifications of the RS-232C interface (CH2: channel 2) of the AD51H.

Table 3.4 RS-232C Interface Specifications

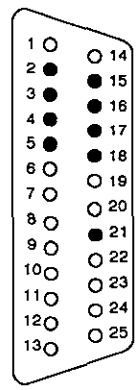
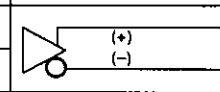
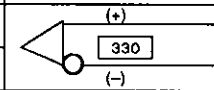
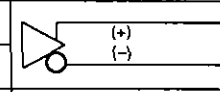
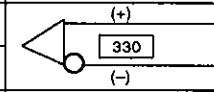
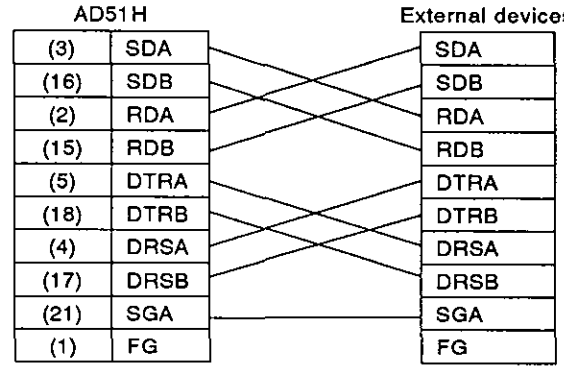
<table>
<tr><th colspan="2">Item</th><th colspan="6">Specifications</th></tr>
<tr><td rowspan="2">Device to be Connected</td><td>Debug port</td><td colspan="6">VG-620, VT-382, VT-220</td></tr>
<tr><td>General-purpose port</td><td colspan="6">Computer with the RS-232C interface, personal computer, printer, modem, VG-620, VT-382, and VT-220</td></tr>
<tr><td colspan="2">Transmission method</td><td colspan="6">Conforms to EIA and RS-232C.</td></tr>
<tr><td colspan="2">Transmission speed (bps)</td><td colspan="6">300, 600, 1200, 2400, 4800, or 9600 can be selected.</td></tr>
<tr><td colspan="2">Synchronous method</td><td colspan="6">Asynchronous method</td></tr>
<tr><td rowspan="7">Setting of the USART mode</td><td>Baud rate (bps)</td><td colspan="6">300, 600, 1200, 2400, 4800, or 9600 can be selected.</td></tr>
<tr><td rowspan="2">Parity bit</td><td colspan="6">Without parity</td></tr>
<tr><td colspan="6">With parity : even/odd parity</td></tr>
<tr><td rowspan="2">Stop bit</td><td colspan="6">1 bit</td></tr>
<tr><td colspan="6">2 bit</td></tr>
<tr><td rowspan="2">Character data bit</td><td colspan="6">Data : 7 bits</td></tr>
<tr><td colspan="6">Data : 8 bits</td></tr>
<tr><td>Communications control</td><td colspan="6">Controlled by the DTR terminal</td></tr>
<tr><td rowspan="11">Connector (AD51H side) used</td><td>Model name</td><td colspan="6">17L-10250-27-D9A(DDK Product)</td></tr>
<tr><td rowspan="10">Pin arrangement</td><td>Appearance of connector</td><td>Pin number</td><td>Signal abbreviation</td><td>Signal directions Inside ↔ Outside</td><td colspan="2">General information</td></tr>
<tr><td rowspan="9">25 O O 13
24 O O 12
23 O O 11
22 O O 10
21 O O 9
20 ● O 8
19 O ● 7
18 O ● 6
17 O ● 5
16 O ● 4
15 O ● 3
14 O ● 2
● 1</td><td>1</td><td>FG</td><td></td><td colspan="2">Frame grounding maintenance</td></tr>
<tr><td>2</td><td>SD</td><td>→</td><td colspan="2">Sent data</td></tr>
<tr><td>3</td><td>RD</td><td>←</td><td colspan="2">Received data</td></tr>
<tr><td>4</td><td>RTS</td><td>→</td><td colspan="2">Request to send</td></tr>
<tr><td>5</td><td>CTS</td><td>←</td><td colspan="2">Sending is enabled.</td></tr>
<tr><td>6</td><td>DSR</td><td>←</td><td colspan="2">Data equipment ready</td></tr>
<tr><td>7</td><td>SG</td><td></td><td colspan="2">Signal ground</td></tr>
<tr><td>20</td><td>DTR</td><td>→</td><td colspan="2">Terminal READY notification</td></tr>
<tr><td></td><td colspan="6"></td></tr>
<tr><td>Connection</td><td colspan="6"> 
</td></tr>
</table>

3.2.5 RS-422 interface (CH3: channel 3)

(1) Interface specifications

Table 3.5 gives the specifications of the RS-422 interface (CH3: channel 3) of the AD51H.

Table 3.5 RS-422 Interface Specifications

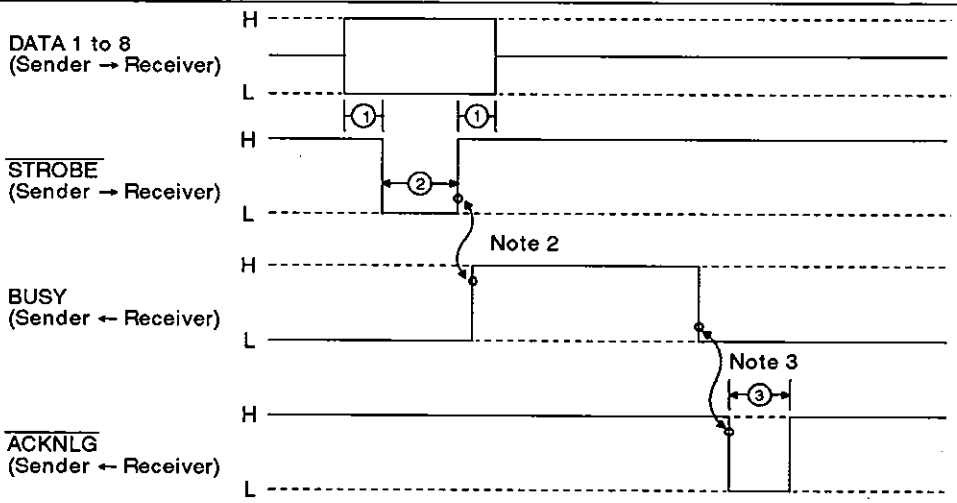
<table>
<tr><th colspan="2">Item</th><th colspan="6">Specifications</th></tr>
<tr><td rowspan="3">Device to be Connected</td><td>Debug port</td><td colspan="6" rowspan="2">PC/AT</td></tr>
<tr><td>Console port</td></tr>
<tr><td>General-purpose port</td><td colspan="6">Computer with the RS-422 interface, personal computer, and printer</td></tr>
<tr><td colspan="2">Transmission method</td><td colspan="6">Conforms to EIA and RS-422.</td></tr>
<tr><td colspan="2">Synchronous method</td><td colspan="6">Asynchronous method</td></tr>
<tr><td rowspan="7">Setting of the USART mode</td><td>Baud rate (bps)</td><td colspan="6">300, 600, 1200, 2400, 4800, or 9600 can be selected.</td></tr>
<tr><td rowspan="2">Parity bit</td><td colspan="6">Without parity</td></tr>
<tr><td colspan="6">With parity : even/odd parity</td></tr>
<tr><td rowspan="2">Stop bit</td><td colspan="6">1 bit</td></tr>
<tr><td colspan="6">2 bit</td></tr>
<tr><td rowspan="2">Character data bit</td><td colspan="6">Data : 7 bits</td></tr>
<tr><td colspan="6">Data : 8 bits</td></tr>
<tr><td>Communications control</td><td colspan="6">Controlled by the DTR terminal</td></tr>
<tr><td rowspan="2">Connector (AD51H side) used</td><td>Model name</td><td colspan="6">17L-10250-27-D9A (DDK Product)</td></tr>
<tr><td>Pin arrangement</td><td>Appearance of connector</td><td colspan="2">Signal abbreviation</td><td>Block figure</td><td>Pin number</td><td>Signal direction</td></tr>
<tr><td colspan="2" rowspan="6"> 
</td><td rowspan="2">Send data</td><td>SDA</td><td rowspan="2"> 
</td><td>(3)</td><td rowspan="2">→Outside</td></tr>
<tr><td>SDB</td><td>(16)</td></tr>
<tr><td rowspan="2">Received data</td><td>RDA</td><td rowspan="2"> 
</td><td>(2)</td><td rowspan="2">←Outside</td></tr>
<tr><td>RDB</td><td>(15)</td></tr>
<tr><td rowspan="2">READY stop notification</td><td>CSA</td><td rowspan="2"> 
</td><td>(5)</td><td rowspan="2">→Outside</td></tr>
<tr><td>CSB</td><td>(18)</td></tr>
<tr><td rowspan="2">Data equipment READY</td><td>RSA</td><td rowspan="2"> 
</td><td>(4)</td><td rowspan="2">←Outside</td></tr>
<tr><td>RSB</td><td>(17)</td></tr>
<tr><td>Signal ground</td><td>SGA</td><td></td><td>(21)</td><td></td></tr>
<tr><td colspan="2">Connection</td><td colspan="6"> 
</td></tr>
</table>

3.2.6 Parallel interface (CH4: channel 4)

(1) Interface specifications

Table 3.6 gives the specifications of the parallel interface (CH4: channel 4) of the AD51H.

Table 3.6 Parallel Interface Specifications

<table>
<tr><th colspan="2">Item</th><th colspan="10">Specifications</th></tr>
<tr><td colspan="2">Conformity standard</td><td colspan="10">Conforms to Centronics.</td></tr>
<tr><td colspan="2">Insulation method</td><td colspan="10">Photocoupler insulation</td></tr>
<tr><td colspan="2" rowspan="2">Signal level</td><td>Input</td><td colspan="9">VIH = 2V, VIL = 0.8V</td></tr>
<tr><td>Output</td><td colspan="9">VOH = 2.4V, VOL = 0.5V</td></tr>
<tr><td colspan="2">Timing chart</td><td colspan="10"> 

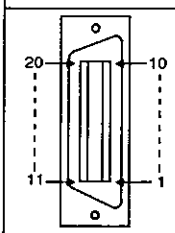
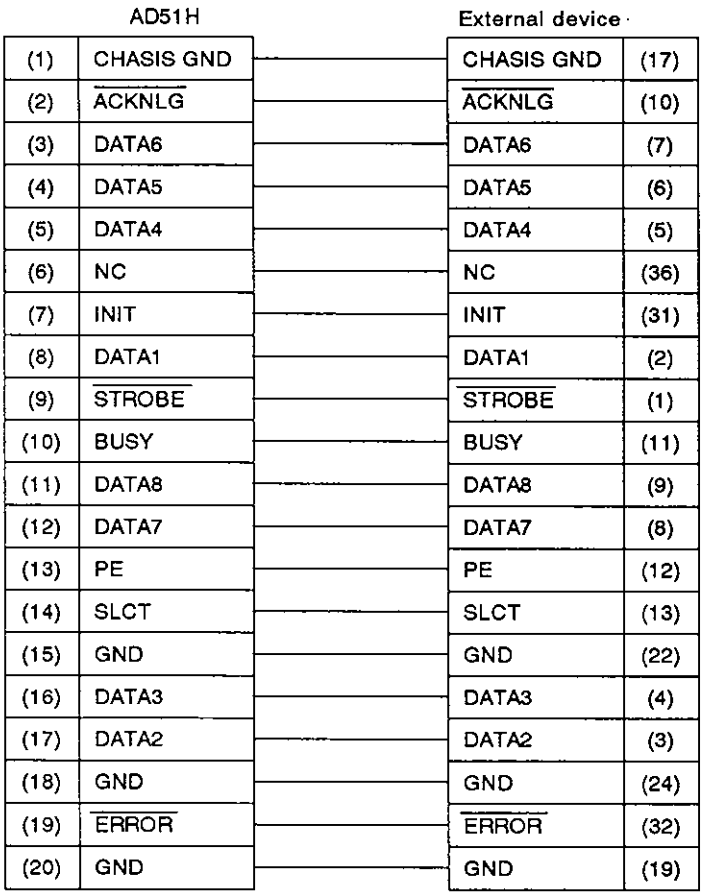
(Note 1) 1)Min. 1.0μS
2)Min. 1.0μS, max. 500μS 3) Min. 2μS
(Note 2) The BUSY signal goes ON at the leading or falling edge of $\overline{\text{STROBE}}$.
(Note 3) At the falling edge of the BUSY signal, $\overline{\text{ACKNLG}}$ is transmitted within 100μS.
The BUSY signal is canceled after $\overline{\text{ACKNLG}}$ is received.</td></tr>
<tr><td rowspan="22">Connector (AD51H side) used</td><td>Model name</td><td colspan="10">10220-52A2JL</td></tr>
<tr><td rowspan="21">Pin arrange-ment</td><td>Appearance of connector</td><td>No.</td><td>Signal name</td><td>No.</td><td>Signal name</td><td>No.</td><td>Signal name</td><td>No.</td><td>Signal name</td></tr>
<tr><td rowspan="5"> 
</td><td>1</td><td>CHASIS GND</td><td>6</td><td>NC</td><td>11</td><td>DATA8</td><td>16</td><td>DATA3</td></tr>
<tr><td>2</td><td>$\overline{\text{ACKNLG}}$</td><td>7</td><td>INIT</td><td>12</td><td>DATA7</td><td>17</td><td>DATA2</td></tr>
<tr><td>3</td><td>DATA6</td><td>8</td><td>DATA1</td><td>13</td><td>PE</td><td>18</td><td>GND</td></tr>
<tr><td>4</td><td>DATA5</td><td>9</td><td>$\overline{\text{STROBE}}$</td><td>14</td><td>SLCT</td><td>19</td><td>$\overline{\text{ERROR}}$</td></tr>
<tr><td>5</td><td>DATA4</td><td>10</td><td>BUSY</td><td>15</td><td>GND</td><td>20</td><td>GND</td></tr>
</table>

Table 3.6 Parallel Interface Specifications

Item	Specifications
Connection	

- For the external device, set the pin numbers for other than the above pins to NC (NONE CONNECTION)

3.3 Memory Card Interface Specifications

This section explains memory card interface specifications.

(1) Memory card interface specifications

Table 3.7 gives the specifications of the memory card interface of the AD51H.

Table 3.7 Memory Card Interface Specifications

Item	MEMORY CARD	
	1	2
Data that can be stored	• File data • Execution program	• File data
Applicable memory card (Section 6.2.1 gives memory card specifications.)	Model name	Memory capacity
	A6MEM-256KA-W	256K bytes
	A6MEM-512KA-W	512K bytes

3.4 ROM Socket Specifications

The following explains the specifications of the ROM socket and applicable EP-ROMs used with the AD51H.

(1) Specifications of ROM sockets

Table 3.8 ROM Socket Specifications

Item	Specifications
Number of pins	40 pins
EP-ROM that can be installed	64KWROM (128K bytes)
	128KWROM (256K bytes)
	256KWROM (512K bytes)

(2) Specifications of applicable EP-ROMs

Table 3.9 EP-ROM Specifications

Item \ Model Name		64KWROM	128KWROM	256KWROM
Memory specifications		EP-ROM (Only read is enabled.)		
Memory capacity		128K bytes	256K bytes	512K bytes
Structure		40 pin IC package		
Storage data	Name	Execution program		
	Capacity	128K bytes	256K bytes	384K bytes

REMARK

Storing execution programs to the EP-ROM:

Use a ROM writer connected to a PC/AT to store execution programs to the EP-ROM.

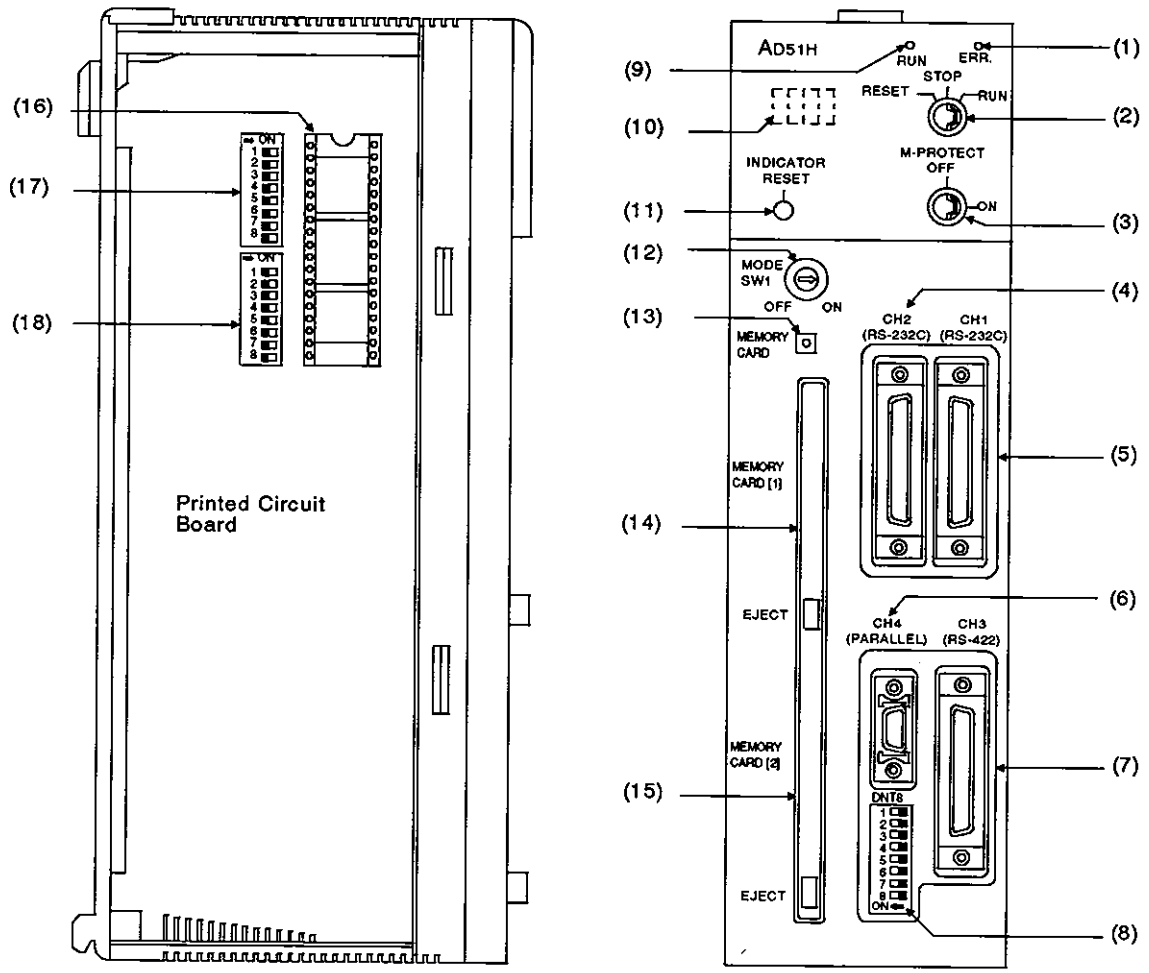
Section 2.6 and Appendix 3 give details about ROM writers that can be connected to a PC/AT.

4. NAMES, FUNCTIONS, AND SETTINGS OF PARTS

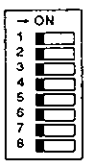
This section explains the names, functions, and settings of parts of the AD51H.

4.1 Names and Functions of Parts

The illustrations below show the names of parts of the AD51H.



No.	Name	Function	Reference Section
(1)	ERROR LED	• This displays the normal/abnormal (error) state of the AD51H. ON : When an error occurs OFF : Normal	—
(2)	RUN keyswitch	• Used for executing and stopping multitasking and for hardware resetting. RUN : Multitasking is executed (only valid in the execution or debugging mode). STOP : Execution of multitasking is stopped (only valid in the execution or debugging mode). RESET : Hardware reset	—
(3)	Memory-protect keyswitch	• Used for the valid/invalid setting of the memory-protect range set by the memory-protect range setting switch. OFF : Invalid ON : Valid	4.2.9

No.	Name	Function	Reference Section
(4)	RS-232C interface	• Used to connect a debugger, general-purpose console, or printer.	3.2.4
(5)	RS-232C interface	• Used to connect a console, debugger, general-purpose console, or printer.	3.2.3
(6)	Parallel interface	• Used to connect a printer.	3.2.6
(7)	RS-422 interface	• Used to connect a console, debugger, general-purpose console, or printer.	3.2.5
(8)	Mode setting switch 2 (MODE SW2)	 • Used for the following settings of the AD51H: Console or debug port The program operation is suspended by inputting (a) the [BREAK] key, or (b) the [Ctrl] + [C] keys. Area to which an execution program is booted Scheduled time of task	4.2.2 to 4.2.5
(9)	RUN LED	• This displays the execution/stop of a task. ON : During multitasking OFF : Multitasking is not being executed. An error occurred during multitasking, and multitasking stopped.	—
(10)	LED indicator	• Displays the operation state and the error code when an error occurs.	
(11)	Indicator reset switch	• Used for resetting when an error is displayed on the LED indicator. (If the cause of the displayed error code has not been eliminated, the error code will be redisplayed.) • When multiple errors occur, error codes are displayed one after another every time an error code indicator is reset.	—
(12)	Mode setting switch 1	• Used for setting the operating mode of the AD51H.	4.2.1
(13)	Memory card access switch	• Used for installing and removing memory cards during the power supply is ON.	4.3.4
(14)	MEMORY CARD1	• A memory card is inserted here, and execution programs and data files are managed. • Read/write of execution programs and access to files are enabled.	3.3 and 4.3.4
(15)	MEMORY CARD2	• A memory card is inserted here, and data files are managed. • Access to files is enabled.	3.3 and 4.3.4
(16)	Memory socket	• Used to install the EP-ROM. • Used to read an execution program from the EP-ROM inserted in the socket.	3.4 and 4.3.5
(17)	Mode setting switch 2	 • Used for the following settings of the AD51H: Operation of the AD51H when resetting a PC CPU Possible access time of FROM/TO instructions from a PC CPU when resetting the AD51H EP-ROM type to be used • Set 4, 5, 7, and 8 to OFF.	4.2.5 to 4.2.7
(18)	Memory-protect range switch	 • Used for setting the memory-protect range of a memory card (used for BASIC programs) installed in MEMORY CARD1. ON : Protected OFF : Not protected	4.2.9

4.2 Settings

This section explains the required settings used with the AD51H.

4.2.1 When setting the operating modes (BASIC program execution, multitask debugging, and programming)

The execution mode must be set to use the AD51H.

(1) Operating modes

There are three kinds of operating modes.

(a) Execution mode

This mode is used to (a) read execution programs from a memory card or the EP-ROM to the program area of the AD51H, and (b) control operations. (See Section 6.4.)

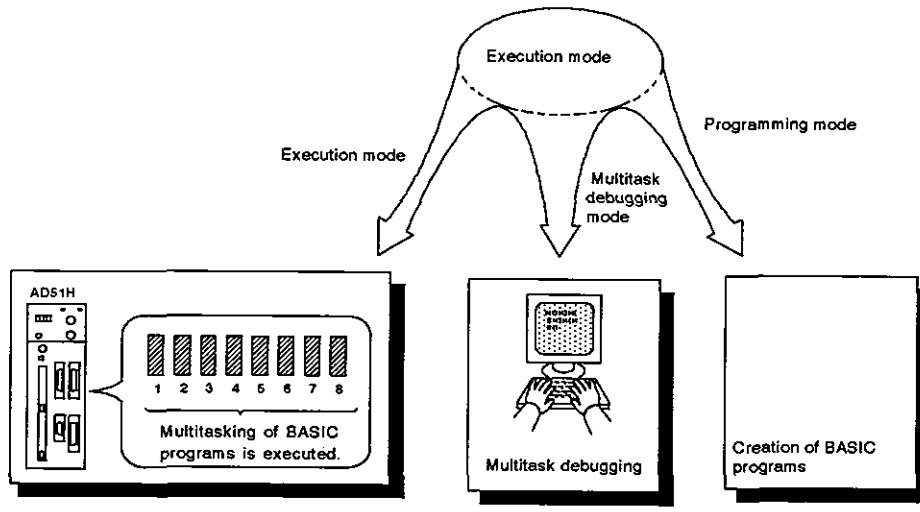
(b) Multitask debugging mode

This mode is used for debugging multiple BASIC programs created in the programming mode by using a debugger.

(See the online programming (debug) part.)

(c) Programming mode

This mode is used to (a)program one-task BASIC programs by using a console, (b) store BASIC programs to a memory card, and (c) do all multitask settings (except programming). (See online programming (debug) part.)

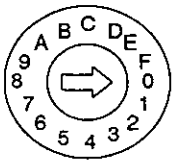


(2) Setting the operating mode

Use mode setting switches 0 to 4 to set the operating mode.

Table 4.1 shows the settings of mode setting switch 1.

Table 4.1 Settings of Mode Setting Switch 1

Operating Mode	Operating Mode Setting No.	Mode Setting Switch
Execution mode	0 or 1	
Multitask debugging mode	2 or 3	
Programming	4	
Unusable	5 to F	

4.2.2 Console and debugger settings

This section tells how to set an interface (RS-232C or RS-422) to which a console (used to create BASIC programs) or a debugger (used to do multitask debugging) is connected.

(1) Consoles and debuggers

Peripheral devices and general-purpose consoles which can be used as consoles and debuggers are explained below.

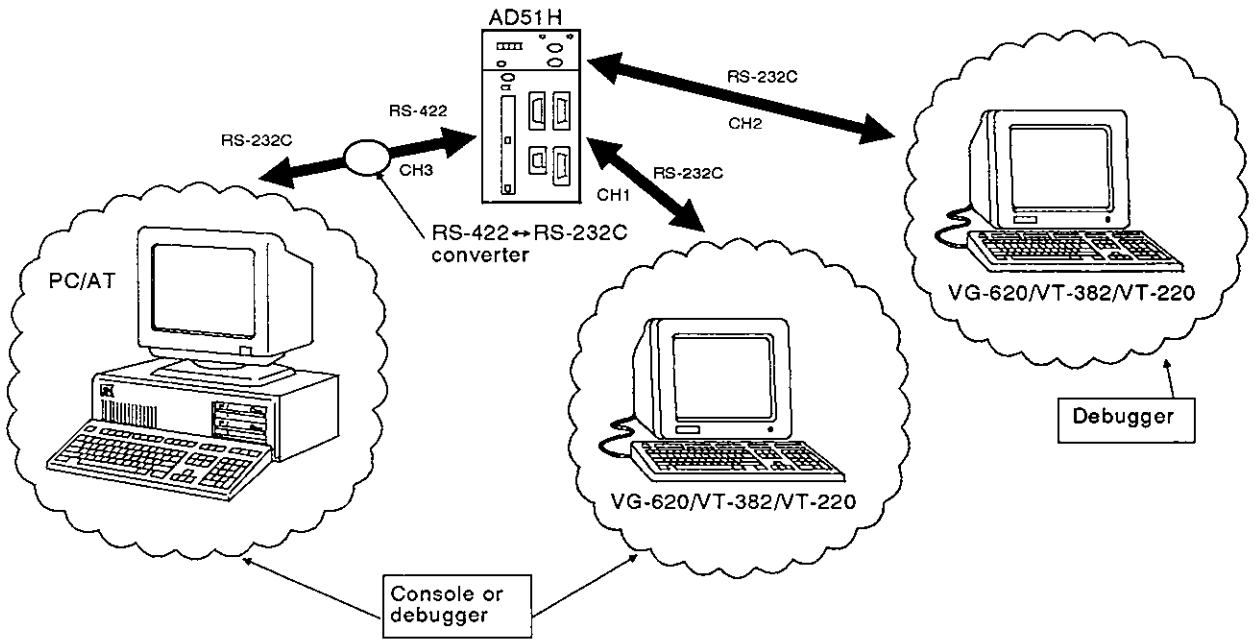
(a) Console

A console is operable in the programming mode (when mode setting switch 1 is set to "4").
Set the RUN keyswitch to either RUN or STOP.

1) Connect a PC/AT to the RS-422 interface via the RS-422 ↔ RS-232C converter.
2) Connect a VT-382, VG-620, or VT-220 to the RS-232C (CH.1) interface.

(b) Debugger

1) Connect a PC/AT to the RS-422 interface via the RS-422 ↔ RS-232C converter.
2) Connect a VT-382, VG-620, or VT-220 to the RS-232C (CH.1 or CH.2) interface.



POINT

A debugger can be operated when the RUN keyswitch is set to RUN in the debugging mode (when the execution mode setting switch is set to "2" or "3").

If the RUN keyswitch is set to STOP, a PC/AT, VT-382, or VG-620 set as the debugger cannot be operated.

(2) Console and debugger settings

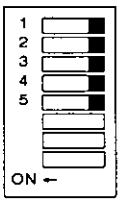
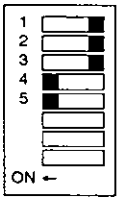
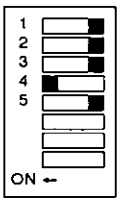
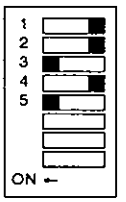
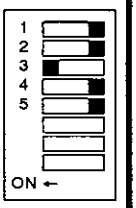
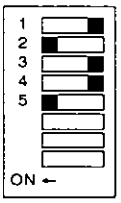
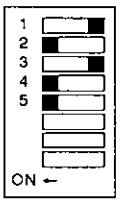
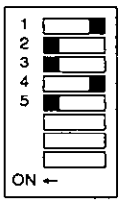
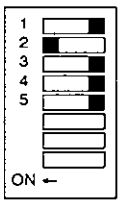
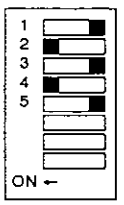
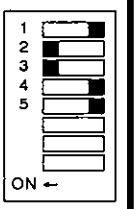
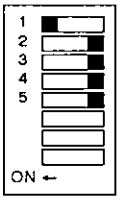
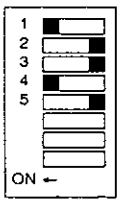
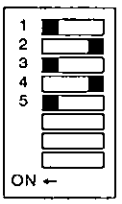
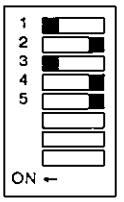
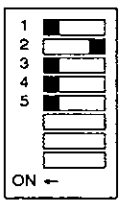
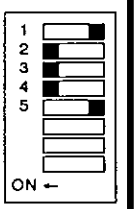
Use a combination of ON/OFF settings of SW1 to SW5 of mode setting switch 2 to set a device (PC/AT, VG-620, VT-382, or VT-220) to be used as a console or a debugger.

If a console or debugger is connected to CH1 and CH2 of the RS-232C interface, only devices (VG-620, VT-382, or VT-220) with the same model name can be used.

Therefore, a VG-620, VT-382, or VT-220 cannot be used simultaneously as a console and debugger.

Table 4.2 shows the settings of mode setting switch 2 for consoles and debuggers.

Table 4.2 Settings of Mode Setting Switch 2 for Consoles and Debuggers

Mode Setting Switch	Console	Debugger					
		None	RS-422	RS-232C (CH1)		RS-232C (CH2)	
			PC/AT	VT-382 or VT-220	VG-620	VT-382 or VT-220	VG-620
Mode setting switch 2 (SW1 to SW5) (Set each switch to the ■ side.)	PC/AT		—				
	VT-382 or VT-220 (connected to CH1)			—	—		—
	VG-620 (connected to CH1)			—	—	—	
	None						

REMARK

Combinations indicated as — in Table 4.2 are not available.

4.2.3 Setting the BASIC program stop/continue operation instructed by the [Break], or [Ctrl] + [C] keys

This sets the inputting of the [Break] *2, or [Ctrl] + [C] keys on the console *1 to valid or invalid during multitasking in the execution mode.

(1) Operations when pressing the [Break], or [Ctrl] + [C] keys

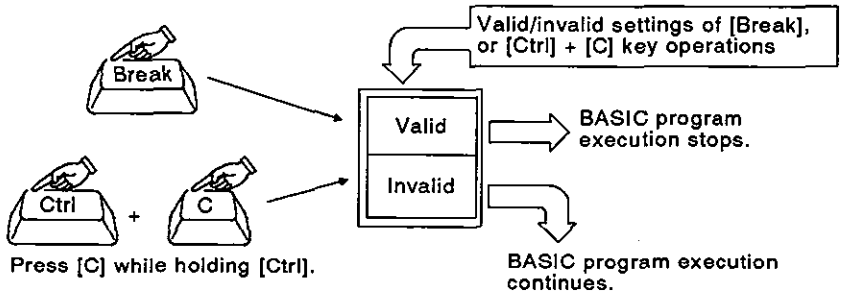
(a) When (a) the key operation is set to valid, and (b) if the [Break], or [Ctrl] + [C] keys (on the console) are pressed, multitasking stops and the operation mode switches to the system mode.

To forcibly stop multitasking from the console, set this to valid.

(b) When (a) this is set to invalid, and (b) if the [Break], or [Ctrl] + [C] keys (on the console) are pressed, multitasking continues.

Operation of the [Break], or [Ctrl] + [C] keys is received by the AD51H as "key code (03H)".

Set this to invalid to prevent accidental suspension of multitasking caused when the [Break], or [Ctrl] + [C] keys are pressed by mistake.

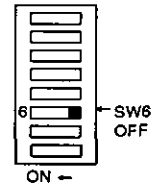
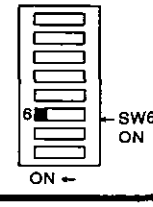


(2) Valid/invalid settings of [Break], or [Ctrl] + [C] key operations

Set the [Break], or [Ctrl] + [C] key operations to valid or invalid by the ON/OFF settings of SW6 of mode setting switch 1 (on the front of the AD51H).

Table 4.6 shows the setting descriptions and settings of mode setting switch 1.

Table 4.6 Valid/Invalid Settings of the [Break], or [Ctrl] + [C] Key Operations

Settings of the [Break], or [Ctrl] + [C] Key Operations	SW6 Setting
Invalid	
Valid	

REMARK

1)*1 : A device set by SW1 to SW5 of mode setting switch 1 is set to a console.

2)*2 : Multitask can be stopped from a PC/AT only by pressing the [Break] key.

However, multitasking cannot be stopped from a VG-620, VT-382, or VT-220 by pressing the [Break] key.

3) Setting the [Break], or [Ctrl] + [C] key operations to invalid prevents accidental suspension of multitasking caused when the [Break], or [Ctrl] + [C] keys are pressed by mistake.

4.2.4 Designation of the read target of execution programs

This section explains settings for reading execution programs stored in a memory card or the EP-ROM to the program area of the AD51H when (a) the power supply is turned ON, or (b) the AD51H is reset.

Designate the read target of execution programs to the memory card or the EP-ROM.

(1) Operations when designating the memory card or the EP-ROM

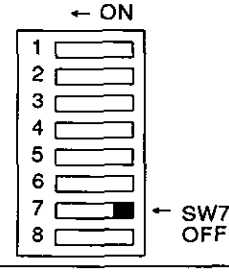
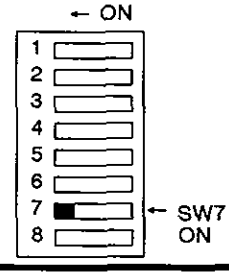
(a) If a memory card is designated, execution programs stored in the memory card installed in the MEMORY CARD1 interface are read to the internal memory of the AD51H.
If there is no memory card in MEMORY CARD1, execution programs will be read from the EP-ROM.

(b) If the EP-ROM is designated, execution programs stored in the EP-ROM (installed in the memory socket on the left side of the AD51H) are read to the program area of the AD51H.
If there is no EP-ROM in the memory socket, execution programs are read from a memory card.

(2) Designating the read target of the execution program

Designate the read target of the execution program by the ON/OFF settings of SW7 of mode setting switch 2 (on the front of the AD51H). Table 4.7 shows settings of mode setting switch 2 and the read target priorities of the execution program.

Table 4.7 Read Target Priorities of the Execution Program

Read Target Priorities of the Execution Program	SW7 Setting
The memory card in MEMORY CARD1 has priority. (If there is no memory card in MEMORY CARD1, execution programs are read from the EP-ROM.)	
The EP-ROM has priority. (If there is no EP-ROM in the memory socket, the execution program is read from the memory card in MEMORY CARD1.)	

POINT

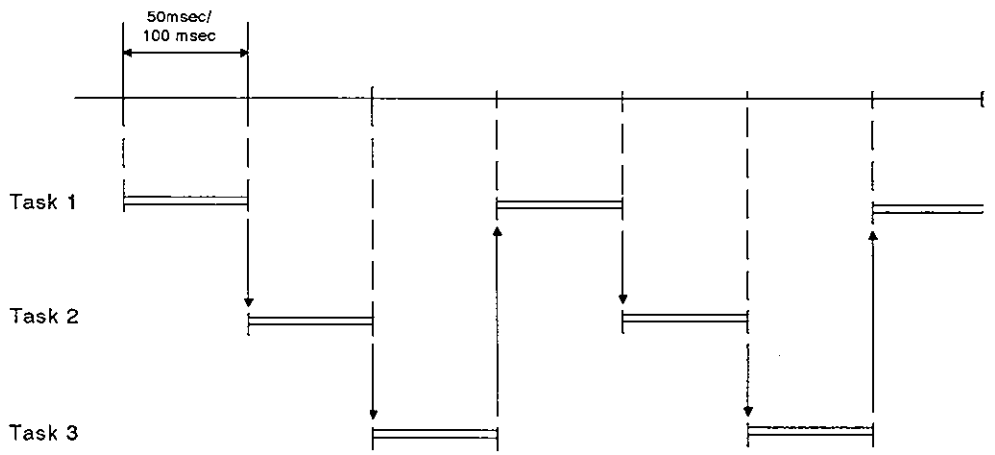
If no execution program is stored in a designated target memory card or EP-ROM, an INIE error occurs, and the execution mode is disabled.

4.2.5 Setting the task switching time

The task switching time is the time it takes to switch BASIC programs during multitask processing.

(1) BASIC program switching

BASIC programs are switched according to the scheduled times of set tasks.



(2) Setting the task switching time

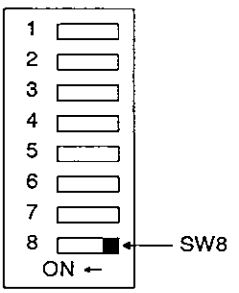
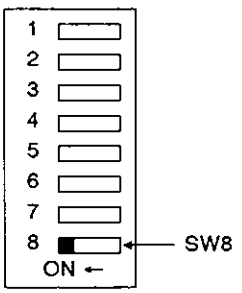
(a) The time is factory-set to 50 msec.

50 msec or 100 msec can be set.

(b) Set the task switching time by the ON/OFF setting of SW8 of mode setting switch 2.

Table 4. shows the setting conditions of mode setting switch 2 for the task switching time.

Table 4. Setting the Task Switching Time Using Mode Setting Switch 2

	Scheduled Time of a Task	
	50 msec	100 msec
Setting of mode setting switch 2		

POINT

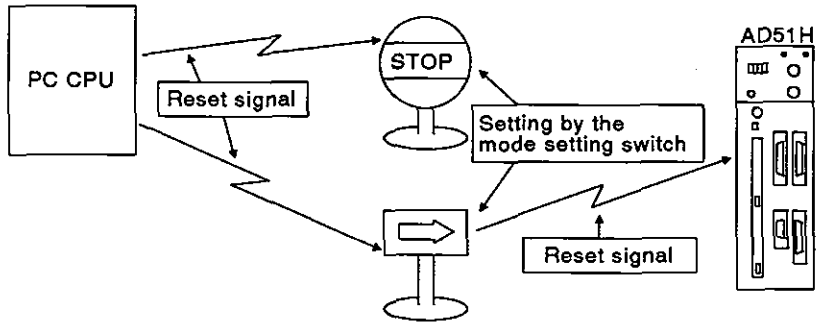
Section 6.5.1 and the AD51H-BASIC Programming Manual give details about switching of the BASIC program execution.

4.2.6 Setting AD51H's operations when a PC CPU is reset

When a PC CPU is reset, the I/O modules and special-function modules installed to the main base unit and extension base units are initialized and restarted up from the initial state. (This operation is the same as when the power supply is turned ON.)

AD51H operations continue or are restarted according to the valid/invalid setting of the PC CPU reset signal.

(The reset signal is factory-set to invalid.)



(1) Operation when resetting a PC CPU

(a) When the PC CPU reset signal is set to valid, the AD51H is initialized and restarts from the initial state.

(Section 6.3 gives the AD51H operations.)

(b) If the PC CPU reset signal is set to invalid, the AD51H continues operations.

The followings occur since the PC CPU has been reset.

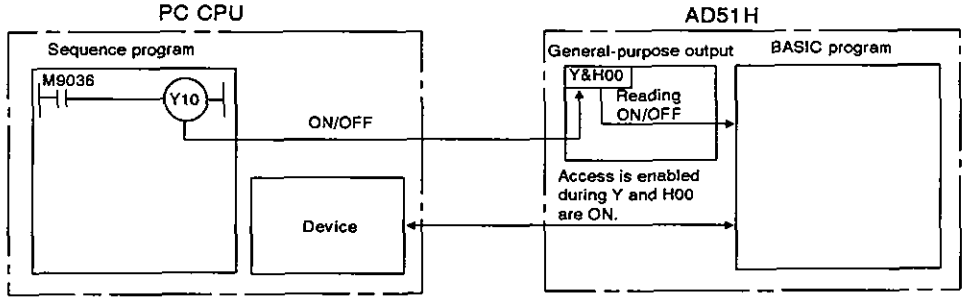
1) ON/OFF outputs controlled by the PC CPU turn OFF.

When the power supply to the PC CPU is turned ON, the task start signal goes ON, and any task to be started by an interruption signal from the PC CPU will be executed.

2) When PC CPU reset processing is executed (five seconds after a reset by the RUN keyswitch), the AD51H cannot access that PC CPU.

If the AD51H uses a BASIC program PCRD/PCWT function to access the PC CPU, and if that PC CPU is reset, the PCRD/PCWT function will cause a time excess error.

Use the output (Y) from the PC CPU as an interlocking signal so that the AD51H can access the PC CPU only when the interlock (output) signal is ON.



POINT

When a file in a memory card or FD is being open, and if the AD51H executes reset processing, data in the opened file is destroyed.

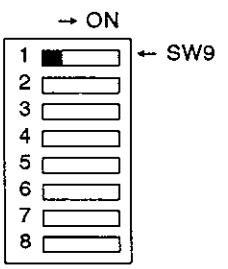
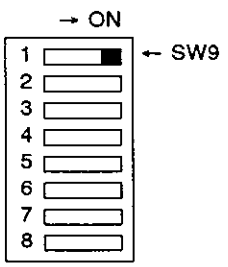
The PC CPU reset signal must be set to invalid when the AD51H is handling data in a file of a memory card or FD.

(2) Valid/invalid setting of the PC CPU reset signal

Valid/invalid setting of the PC CPU reset signal is determined by the ON/OFF setting of SW9 of mode setting switch 2 on the left side of the AD51H.

Table 4.8 shows valid/invalid setting of the PC CPU reset signal using mode setting switch 2.

Table 4.8 Valid/Invalid Setting of the PC CPU Reset Signal

	Setting the Reset Signal to Invalid	Setting the Reset Signal to Valid
Setting of mode setting switch 2		

4.2.7 Setting the time to access a PC CPU after the AD51H is reset

(1) Operations when the AD51H is reset

(a) Communications cannot be executed with a PC CPU until the AD51H is initialized and started up from the initial state after using the RUN keyswitch to reset the AD51H.

If read/write from/to the AD51H's buffer is executed by the FROM/TO instruction from a PC CPU while the AD51H is executing reset processing, this causes a SP UNIT DOWN error and sequence program operations will stop.

(b) The multitask execution start flag (X1B) turns OFF while the AD51H executes reset processing.

The PC CPU access time needs to be set so that a PC CPU surely detects that the multitask execution start flag (X1B) is OFF after the AD51H is reset.

Set this time longer than the PC CPU scan time.

(2) Access time setting

(a) The access time is factory-set to 200 msec.

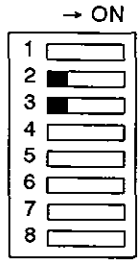
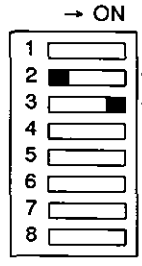
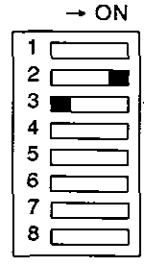
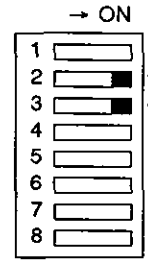
When the PC CPU scan time is 200 msec or more, set a value longer than the scan time as the PC CPU access time.

200 msec, 500 msec, 1000 msec, or 2000 msec can be set.

(b) Set the access time by the ON/OFF setting of SW10 to SW11 of mode setting switch 2 on the left side of the AD51H.

Table 4.9 shows the setting of the access time by the mode setting switch.

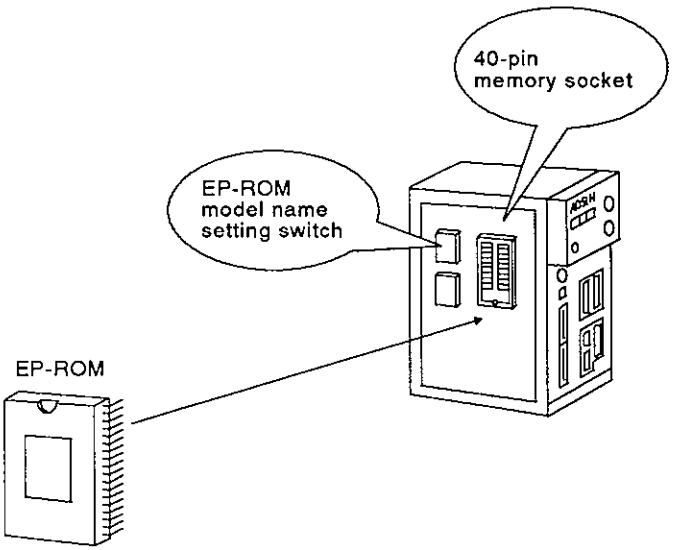
Table 4.9 Setting of the Access Time by the Mode Setting Switch

	Access Time			
	200 msec	500 msec	1000 msec	2000 msec
Setting of mode setting switch 2				

4.2.8 Setting the EP-ROM model name

Either a 64K ROM, 128K ROM, or 256K ROM can be used with the AD51H.

The EP-ROM model name setting designates whether a 64K ROM, 128K ROM, or 256K ROM is installed in the ROM socket of the AD51H.

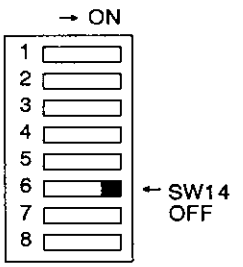
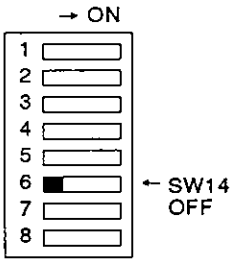


(1) How to set the EP-ROM model name

Set the EP-ROM model name by the ON/OFF setting of SW14 of the mode setting switch on the left side of the AD51H.

Table 4.10 shows the setting of the EP-ROM model name by mode setting switch 2.

Table 4.10 Setting of the EP-ROM Model Name

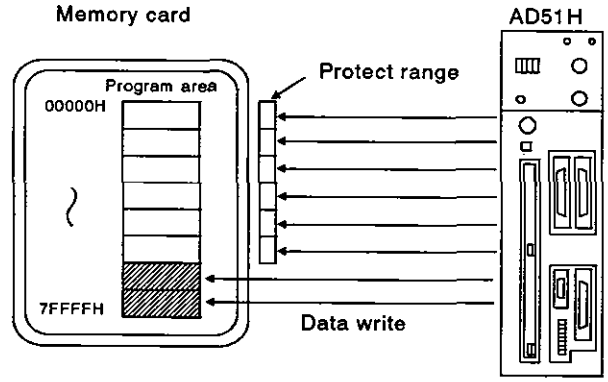
EP-ROM Model Name	Setting of SW14
64K ROM or 128K ROM	
256K ROM	

4.2.9 Setting the memory-protect range of a memory card

This section explains the setting of the protective range of the execution program area in a memory card installed in MEMORY CARD1.

Only execution programs stored in a memory card can be protected, and the rest of the area (except for execution programs) can be used for storing data files.

(If memory-protect is set by the memory card's protect switch, all areas of the memory card are set to memory-protect, and the area except the execution program area cannot be used for data storage.



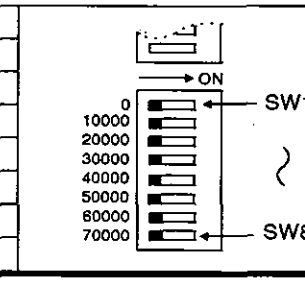
(1) Memory-protect is set by SW1 to SW8 of the memory-protect range setting switch.
If the memory-protect range setting switch is ON, memory-protect is set.

(2) Memory-protect can be set for the 512K-byte area beginning with the head of the memory area.
The memory-protect area can be divided into eight areas each of which has 64K bytes, and memory-protect can be set in units of 64K bytes. Set the memory-protect range from 0H continuously to the end address of the range.

(3) Table 4.11 shows the memory-protect range that can be set by the memory-protect range setting switch.

Table 4.11 Memory-Protect Range

Memory-Protect Range	Setting of the Memory-Protect Setting Switch
0H to 0FFFFH	SW1
10000H to 1FFFFH	SW2
20000H to 2FFFFH	SW3
30000H to 3FFFFH	SW4
40000H to 4FFFFH	SW5
50000H to 5FFFFH	SW6
60000H to 6FFFFH	SW7
70000H to 7FFFFH	SW8



ON : Memory-protect is set.
OFF : Memory-protect is canceled.

POINT

The range set by the memory-protect setting switch is valid only when the memory-protect keyswitch is turned ON.

4.3 Handling and Installation

This section explains the handling instructions for, and the installation environment of the AD51H.

4.3.1 Handling instructions

This section gives the proper procedures for handling the AD51H.

(1) Since the case is made of plastic, do not drop the AD51H or subject it to mechanical shock.

(2) Do not remove the printed circuit board of any module from its case.

(3) When wiring, make sure wire offcuts do not get into the module.

(4) Tighten the module mounting screws (if necessary) as shown in the following table:

Screw	Tightening Torque kg·cm (lb·inch)
Module screws (optional) (M4 x 0.7)	8 (6.93) to 12 (10.39)

(5) To attach the AD51H to the base, press the AD51H against the base so that the latch locks securely. To detach the AD51H, push in the latch. Then, after the latch is disengaged from the base, pull the AD51H toward you.

(The User's Manual of each building block-type CPU gives details.)

4.3.2 Installation environment

Never install the system in the following environments:

(1) Locations where the ambient temperature is outside the range of 0 to 55°C.

(2) Locations where the ambient humidity is outside the range of 10 to 90%RH.

(3) Locations where dew condensation takes place due to sudden temperature changes.

(4) Locations with corrosive or combustible gas.

(5) Locations with a high level of conductive powder such as dust and iron filings, oil mist, salt, and organic solvent.

(6) Locations exposed to the direct sunlight.

(7) Locations where strong power and magnetic fields are generated.

(8) Locations where vibration and shock can be directly transmitted to the system modules.

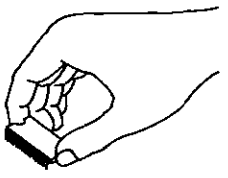
4.3.3 Inserting and removing the EP-ROM

The section gives the procedures and precautions when inserting or removing an EP-ROM in or out of the ROM socket.

(1) Inserting the EP-ROM

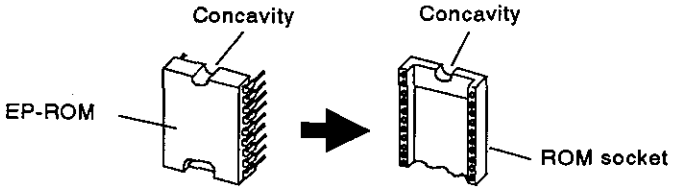
(a) Do not touch memory lead pins with the hand or fingers.

Static electricity may clear the memory, or a pin may be bent.



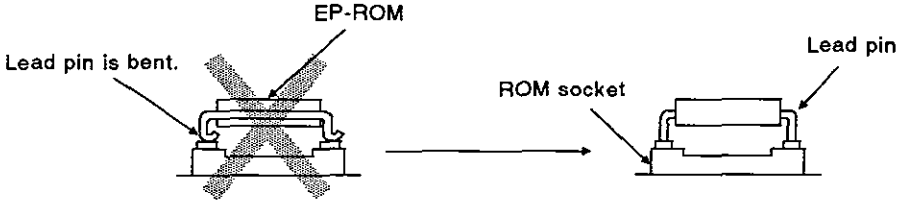
(b) When inserting the EP-ROM, make sure the concavity of the EP-ROM matches the concavity of the ROM socket.

If the EP-ROM is inserted incorrectly, the memory will be cleared when the power is turned ON.

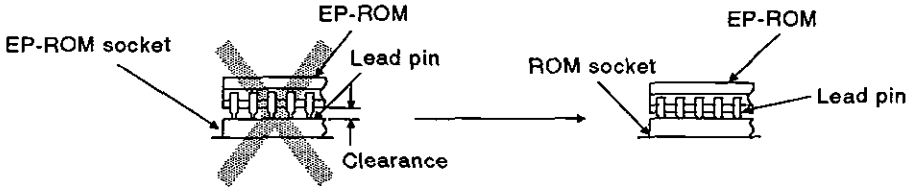


(c) Be very careful not to bend EP-ROM memory lead pins when inserting the EP-ROM.

If an EP-ROM pin is bent, when the power supply is turned ON, normal reading cannot be executed and the memory may even be cleared.



(d) After the EP-ROM is inserted, make sure it is set firmly in the socket without a clearance.



(2) Removing the EP-ROM

Use a special pulling tool (for 40-pin ICs) to remove the EP-ROM from the ROM socket.

4.3.4 Inserting and removing memory cards

(1) How to insert and remove memory cards when the power is OFF

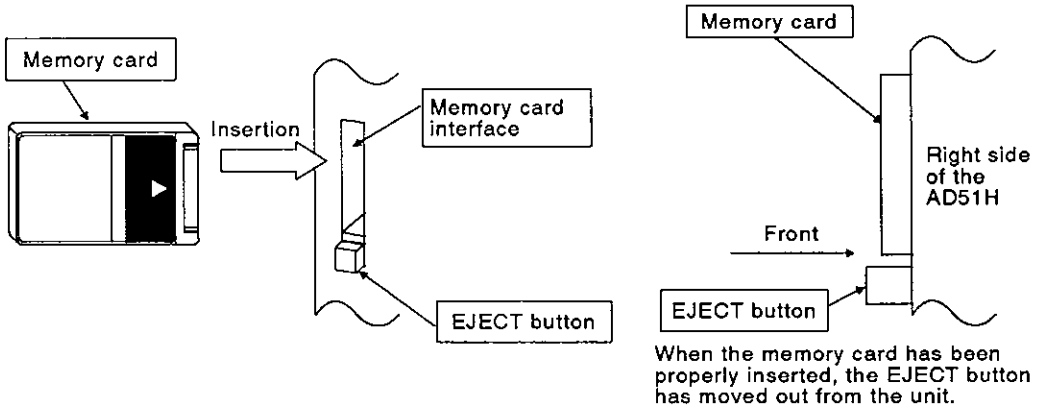
When inserting or removing a memory card while the power is ON, follow the procedure given in 4.3.4 (2).

(a) Inserting a memory card

When inserting a memory card, make sure the card is inserted with the arrow side down. Push the memory card in firmly until it is securely installed in the connector.

If the memory card is properly installed in the connector, the EJECT button will move out.

After inserting the memory card, make sure that the EJECT button has moved out, as shown below.



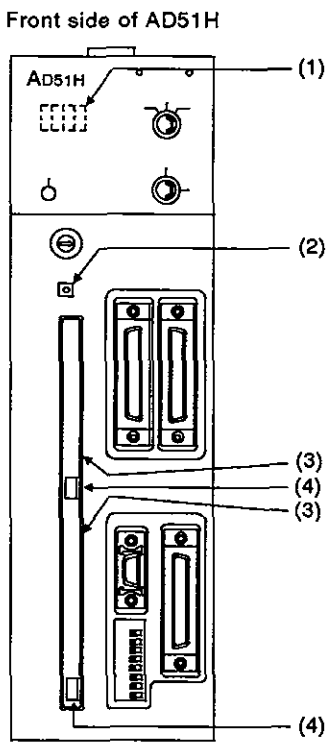
(b) Removing a memory card

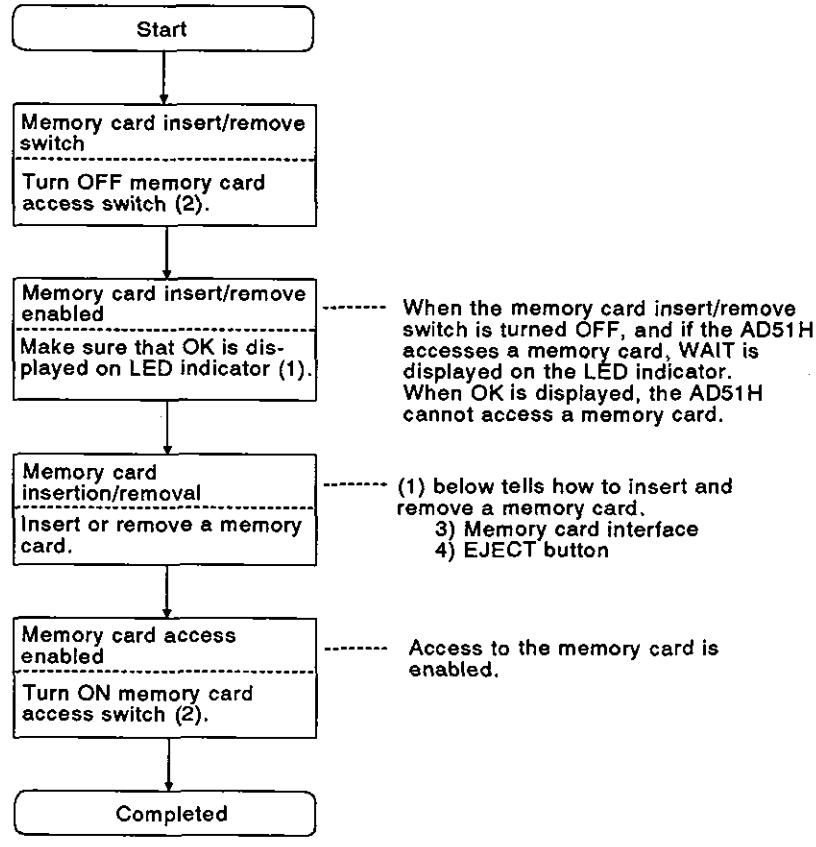
Press the EJECT button firmly to remove the memory card.

Pressing the EJECT button dislodges the memory card from the connector and slides it forward. The card can then be pulled out by hand.

(2) How to insert and remove memory cards when the power is ON

When the power is ON, the following procedure must be followed to insert or remove memory cards (after making sure the AD51H is not accessing a memory card).





POINT

(1) When inserting or removing a memory card when the power is ON, follow the procedure shown in (2) above.

If a memory card is inserted or removed while the AD51H is accessing the memory card, the data in that memory card will be cleared.

(2) The memory card access switch is used for both MEMORY CARD1 and MEMORY CARD2.

If the memory card access switch is turned ON, the AD51H cannot access a memory card in MEMORY CARD1 or MEMORY CARD2.

(3) Memory cards can be accessed after accessibility to the memory card is confirmed by a special relay (EM9201) and a special register (ED9021).

(4) If the AD51H accesses a memory card in the file maintenance mode of an SW0IX-AD51HPE, do not turn OFF the memory card access switch.

If the memory card access switch is turned OFF, processing will be interrupted.

4.4 Precautions on Wiring

External wiring should be noise-resistant so that the functions of the AD51H can be fully performed with a high system reliability.

The following precautions must be taken when doing external wiring to the AD51H.

(1) Do not put external wiring near (closer than 100 mm) a main circuit cable, a high-tension line, or a load-carrying wire (other than the load-carrying wire of a PC).

Do not bundle the wires together.

If these precautions are not taken, the AD51H may be subject to noise, surges, and/or inductions.

(2) The FG terminal of a power supply module is not connected to the FG terminal of the RS-232C or RS-422 interface of the AD51H.

The shield of a shielded cable or wire should be connected to the ground of the external device by one-point grounding.

5. MEMORY CARD BATTERY REPLACEMENT

This section gives the period and method of replacement of the battery in a memory card.

5.1 When Should the Battery be Replaced?

(1) Indicates low battery power

The AD51H verifies whether battery power to a memory card inserted in MEMORY CARD1 or MEMORY CARD2 is low.

When the AD51H detects low battery power, an error message is displayed on the LED indicator.

Error Message	Meaning
BAT1	Indicates low battery power to the memory card in MEMORY CARD1
BAT2	Indicates low battery power to the memory card in MEMORY CARD2

(2) When should the battery be replaced?

The contents of a memory card can be retained about 168 hours (one week) after the low battery error message is displayed on the LED indicator.

However, when the low battery error message is displayed, the battery should be replaced as soon as possible.

(3) Battery life

The memory card is battery-backed. Mitsubishi recommends that the battery should be replaced at certain intervals as shown below.

Operating Condition \ Model		A6MEM-256KAW	A6MEM-512KAW
When the memory card is battery-backed	Minimum guaranteed lifetime	1900 Hr	900 Hr
	Real lifetime (typical)	11300 Hr	5900 Hr
Working time after the low battery error message is given (ambient temperature: 25°C)		79 Hr	39 Hr

Note: Mitsubishi cannot guarantee the backup performance if a new battery is used after five years of storage.

POINT

When the memory card is installed in MEMORY CARD1 or MEMORY CARD2 of the AD51H, the data in that memory card is backed up by the power of the AD51H.

5.2 Replacing the Battery

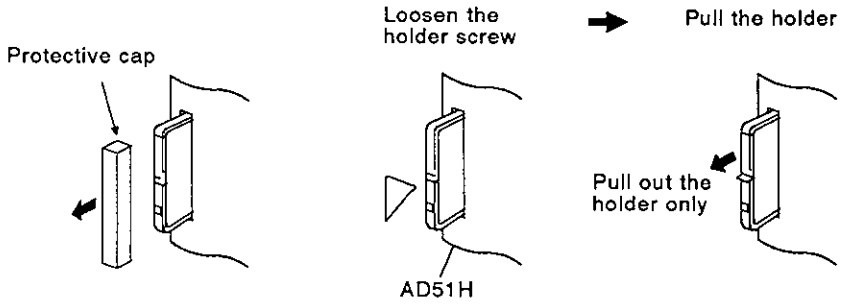
To replace the battery, install a memory card in MEMORY CARD1 or MEMORY CARD2, and make sure that power to the AD51H module is ON.

If this is not followed, all data in the memory card will be cleared because the memory card is not backed up by a capacitor.

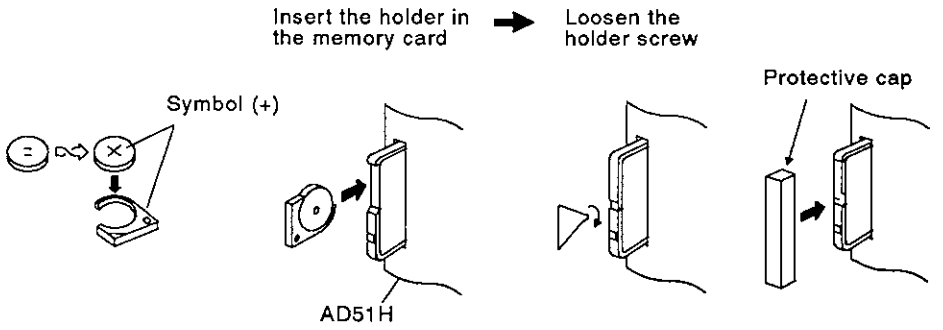
Be sure to turn OFF the power to the output modules of the PC system to which the AD51H is installed to prevent static electricity from damaging data in the memory card.

[Replacement procedure]

1) Prepare a lithium battery (model BR2325 or equivalent).
2) Insert a memory card into the memory card interface on the module.
   (Section 4.3.4 gives how to install a memory card in the AD51H.)
3) Set the RUN keyswitches of the AD51H and the PC CPU to the STOP position.
4) Make sure that power to the output modules is OFF. Then, turn ON the PC CPU power.
5) Remove the protective cap from the battery holder of the memory card.
6) Loosen the holder screw 4 to 5 mm, and pull the battery holder from the memory card.



7) Put the new battery in the battery holder, and install the battery holder to the memory card as described in 4. BATTERY INSTALLATION.



8) Attach the protective cap to the battery side of the memory card.

POINT

The instruction manual included in the memory card package gives details about memory card handling.

[CONTROL FUNCTIONS]

CONTENTS

6. PERFORMANCE SPECIFICATIONS OF THE AD51H

This section explains and lists the performance specifications of the AD51H.

6.1 Performance Specifications

This section gives the performance specifications and internal memory maps of the AD51H.

6.1.1 Performance specifications list

Table 6.1 gives the performance specifications of the AD51H.

Table 6.1 Performance Specifications of the AD51H

Item		Specifications	Reference Section
CPU device		80C186 (15MHz)	—
Program language		AD51H-BASIC	—
Number of tasks		(Max.) eight tasks	—
Conditions for starting a task		• Started by power ON • Started by an interrupt from the PC CPU • Started by the start command from other task	6.4
Internal memory		Program memory: (Max.) 384K bytes/eight tasks (Task size can be set to 16K, 32K, 48K, or 64K bytes)	6.6.2
		Common memory: 8K bytes	6.6.6
		Extension register: 1024 points (2K bytes)	6.6.4
		Extension relay: 1024 points	6.6.3
General-purpose I/O		• Input: 27 points • Output: 17 points	6.10
Buffer		3K words (6K bytes)	6.6.5
Memory-protected		0H to 7FFFFH (Execution program in the memory card	4.2.9
Built-in interface	RS-422 I/F	• Conforms to EIA. RS-422 • Used with a connector • Channel 3 • Transmission distance: within 500 m	3.2.5
	RS-232C I/F	• Conforms to EIA. RS-232C • Used with a connector • Channels 1 and 2 • Transmission distance: within 15 m	3.2.3 to 3.2.4
	Centronics I/F	• Conforms to Centronics • Used with a connector • Channel 4 • Transmission distance: within 3 m	3.2.6
	Memory loading I/F	Applicable memory card: A6MEM-[ ]A-W	3.3
Memory backup		No backup (However, the data in the memory card is battery-backed.)	5
Writing a user program to the ROM		Enabled (execution programs only) (Write to the ROM using the ROM writer connected to a PC/AT.)	—
Console		• PC/AT • VG-620 • VT-382/VT-220	—
Number of occupying I/O points		48 points (vacant 16 points + X/Y 32 points)	6.10
Internal current consumption (5V)		1.0 A	—
Outside dimensions		250 (9.84)(H) x 76 (2.99)(W) x 120 (4.72)(D) mm	—
Weight		0.9 Kg (1.98 lb)	—

6.1.2 AD51H-BASIC commands

Table 6.2 gives the AD51H-BASIC commands that can be used with the AD51H. (The AD51H-BASIC Programming Manual gives details about commands and functions.)

Table 6.2 AD51H-BASIC Commands and Functions

Classification		Commands and Functions	Operation Description
Command to create and modify a program	Preparation for creating a program	NEW	Deletes the program in the program area and initializes all variables.
		AUTO	Automatically displays the line number at the beginning of a line.
		CLEAR	Initializes a variable and sets a memory area.
	Editing a program	DELETE	Clears the designated program range.
		RENUM	Renumbers the line numbers of a program.
		LIST	Displays all (or part) of a program in the program area.
Function to read memory capacity	Reading memory capacity	FRE	Sets the size of an unused area in the program area by the number of bytes.
Command to execute and control a program	Executing a program	RUN	Executes a program in the program area.
			Reads and executes a program from an FD, HD, or memory card.
		CONT	Restarts execution of a program stopped by pressing the [BREAK], or [CTRL] + [C] keys.
			Restarts the program execution stopped by a STOP or END command.
	Controlling a program	CHAIN	This (a) clears all or part of the currently executed program, and (b) reads and executes a designated program.
		FOR ~ NEXT	Repeatedly executes a series of commands a designated number of times.
		STOP	When in the edit mode, temporarily stops execution of a program and makes the processing state wait for a command input.
			When in the execution mode, stops execution of a program and makes the processing state pause.
		END	When in the edit mode, stops execution of a program and makes the processing state wait for a command input.
			When in the execution mode, stops execution of a program and makes the processing state pause.
		WHILE ~ WEND	Repeatedly executes a series of commands between the WHILE and WEND commands while a designated condition is satisfied.
	Branching a program	GOTO	Unconditionally jumps the program execution to a designated line.
		GOSUB ~ RETURN	Branches a program to the subroutine and returns it to a designated line number.
		ON GOTO	Branches a program to one of designated line numbers according to a designated value.
		ON GOSUB	Branches the program to one of designated line numbers according to a designated value.
		IF GOTO ELSE	Selects a target branch according to the result of an operation formula.
		IF THEN ELSE	
Command to debug a program	Debugging a program	TRONF	Starts tracing a program.
		TROFF	Resets the program trace set by the TRON command.
Command to convert type	Type conversion	CDBI	Converts a double-precision real number into a two-word (32-bit) integer used by a PC CPU.
		CSNI	Converts a single-precision real number into a two-word (32-bit) integer used by a PC CPU.

Table 6.2 AD51H-BASIC Commands and Functions (cont.)

Classification		Commands and Functions	Operation Description
Command to write to a designated bit	Writing to a designated bit	WTSET	Writes 0 or 1 to a designated bit of integer type array variable.
Command to read from a designated bit	Reading from a designated bit	RDSET	Reads one-bit data from a designated bit of integer type array variable.
Character-string function	Character↔ASCII conversion	ASC	Converts the first character in a character-string formula to a corresponding character code.
		CHR$	Converts the value of an integer expression into character codes.
	Character→ numeric conversion	STR$	Regards a designated numeric value as a decimal and converts it into a character string.
		VAL	Converts a character string into a numeric value.
	Numeric value→ character string conversion	HEX$	Converts a decimal number into a character string of hexadecimal numbers.
		OCT$	Converts a decimal number into a character string of octal numbers.
		BIN$	Converts a decimal number into a character string of binary numbers.
	Character extraction	LEFT$	Extracts and designates a character string (with a designated number of characters) beginning with the left end of the character string.
		MID$ (part 2)	Designates a character string beginning with a designated position in a character string.
		RIGHT$	Extracts and designates a character string (with a designated number of characters) beginning with the right end of the character string.
		STRING$	Designates a character for designated number of characters.
		SPACE$	Designates a blank character string for designated number of strings.
	Character search	INSTR	Searches a designated part of a character string and designates the position found first.
	Character-string length search	LEN	Designates the number of characters that constitute a character string.
Format control function	Format control	SPC	Designates a blank space in the designated number of spaces.
		TAB	Advances a present character display position to a designated position.
Screen control command	Screen control	CONSOLE	Designates the number of display columns on the console screen.
		CLS	Clears the display from the screen.
		LOCATE	Designates the display position on the console screen.
I/O control command	Board allocation	ZOPEN	Opens a communications port for an external device.
		ZCLOSE	Closes the channel of the port used for communications with an external device.
	Buzzer output	BEEP	Sounds a buzzer from the built-in speaker in the console.
	Printer output	LLIST	Outputs all programs, part of a program, or any program in a designated range of the program area to the printer.
		LPRINT	Outputs data to the printer.
		LPRINT USING	Outputs data to the printer in a designated format.
		LFILES	Outputs the name and size of a file to the printer.
		ZLDV	Switches the port of the printer.
		WIDTH	Sets the output width to the printer.
	Output to the screen	PRINT	Displays data on the screen.
		PRINT USING	Displays a character string or numeric value in a designated format.
		KEY LIST	Displays a character string defined by a function key on the screen.
	Output to the port	ZSEND	Transmits data from a designated port.
	Input from the key-board	INPUT	Enables data input from the keyboard.
		LINE INPUT	Stores an input line to the character-string variables area without dividing the input line (within 255 characters).
	Input from the port	ZRECEIVE	Receives data from a communications port.
	Port setting change	ZCNTL	Changes setting of a port to an open communications port or reads the state of a communications port.

Table 6.2 AD51H-BASIC Commands and Functions (cont.)

Classification		Commands and Functions	Operation Description
Input function	Input from the keyboard	INKEY$	When a character is input from the keyboard, this designates the character. If no character is input, this designates an empty character string.
		INPUT$	Reads and designates a designated-length character string from the keyboard of a console.
			Reads and designates a designated-length character string from a sequential file.
			Reads and designates a designated-length character string from a communications port of the AD51H.
Port control function	Controls port state	COM ON/OFF/STOP	Controls interruption-enabled, interruption-disabled, and stop from a communications line.
		ON COM GOSUB	This defines the starting line of a branch processing routine when an interruption is given through a communications line.
Error control function	Error control	ON ERROR GOTO	When an error occurs, this executes a program from a designated line number.
		ERROR	Makes an error with a designated error code occur.
Error information function	Error information	ERR	Designates a detected error code.
		ERL	Designates the line number where an error is detected.
Repetition command to programming mode	Repetition to the programming mode	SYSTEM	Stops control of the BASIC interpreter and switches the mode to the AD51H system mode.
			Stops offline programming and returns the screen to the main menu.
File operation command	BASIC program I/O	LOAD	Reads a program in an FD, HD, or memory card to the program area.
		MERGE	Combines a program in the program area with a program in an FD or HD.
		SAVE	Stores a program in an FD, HD, or memory card.
	File maintenance	FILES	Displays the name of a file in an FD, HD, or memory card
		FORMAT	Initializes (formats) the file area in a memory card.
		KILL	Clears a file in an FD, HD, or memory card.
			Clears a system name in an FD, HD, or memory card.
		NAME	Changes the file name of a program or data in an FD, HD, or memory card.
I/O command to the file	Allocation of file buffer	OPEN	Opens a file, and enables I/O processings.
		CLOSE	Ends file I/O processing.
	I/O of a sequential file	PRINT#	Writes data to a sequential file.
		PRINT# USING	Designates the data format and writes data to a sequential file.
		INPUT#	Reads data from a sequential file.
		LINE INPUT#	Reads a character line (up to 265 characters) from a sequential file without dividing the line to the character string variables area.
	I/O of a random file	FIELD	Allocates a designated variable area to buffer of the random file.
		LSET	To prepare for the PUT command, this transfers data to the left end of buffer of a random file.
		RSET	To prepare for the PUT command, this transfers data to the right end of buffer of a random file.
		PUT	Writes one-record data from the buffer to a random file.
		GET	Reads one-record data from a random file to the buffer.

Table 6.2 AD51H-BASIC Commands and Functions (cont.)

Classification		Commands and Functions	Operation Description
File information function	File information	EOF	Designates -1 when the end of a sequential file is detected.
		LOF	Designates the size of a file in the number of sectors.
		LOC	Designates the present end position in a file.
File data	Character string → numeric value	CVI	Returns a character string converted by the MKS$ function to an integer.
		CVS	Returns a character string converted by the MK$ function to a single-precision real number.
		CVD	A character converted by the MKD$ function is returned to a double-precision real number.
		CVSMBF	Converts a character string, that is converted from a single-precision real number of the AD51H-BASIC using an MKS$ function, into the internal representation (single-precision internal representation of the IEEE format) of a floating decimal point real number used for A2A and A3A.
		CVDMBF	Converts a character string, that is converted from a double-precision real number of the AD51H-BASIC using an MKD$ function, into the double-precision internal representation of the IEEE format.
	Numeric value → character string	MKI$	Converts an integer type numeric value into a character string.
		MKS$	Converts a single-precision type numeric value into a character string.
		MKD$	Converts a double-precision type numeric value into a character string.
		MKSMBF$	Converts internal representation data (single-precision internal representation of the IEEE format) of a floating decimal point real number used for A2A and A3A into a character string that can be expressed as a numeric value by using the CVS function.
		MKDMBF$	Converts data of the double-precision internal representation of the IEEE format into a character string that can be expressed as a numeric value by using the CVD function.
Definition and declaration command	Definition and declaration	DEF FN	Defines a user function and gives it a name.
		DEFINT	Defines a variable (that begins with a character in a designated range) as an integer type variable.
		DEFSEG	Defines a variable (that begins with a character in a designated range) as a single-precision real number type variable.
		DEFDBL	Defines a variable (that begins with a character in a designated range) as a double-precision real number type variable.
		DEFSTR	Defines a variable (that begins with a character in the designated range) as a character type variable.
		KEY	Defines a character string to each function key of a console.
		DIM	Designates the size of an array variable and allocates the memory capacity needed for the array.
		ERASE	Clears an array defined by the DIM command from the memory.
		REM	Designates a comment in a program.
		COMMON	Sets the variables to be transferred to a program which is executed by the CHAIN command.
Figure and character-string operation command	Data operation	DATA	Designates numeric values and character strings to be read by the READ command.
		READ	Reads a value defined by the DATA command and assigns it to a variable.
		RESTOR	Reads the value of a DATA command which is to be read by a READ command from the beginning of the DATA command of a designated line number.
	Assignment to variables	LET	Assigns the value of a formula to a variable.
		SWAP	Exchanges the values of two variables.
		ZMOVE	Transfers a value between two variables.
		BSWAP	Exchanges two values in a byte unit.
	Replacement of a character string	MID$ (part 1)	Replaces part of a character string with another character string.

Table 6.2 AD51H-BASIC Commands and Functions (cont.)

Classification		Commands and Functions	Operation Description
Numeric value function	Arithmetical function	ABS	Designates the absolute value of a formula.
		ATN	Designates the inverse tangent (tan-1) of a formula.
		COS	Designates the value of the cosine in a trigonometric function.
		EXP	Designates the value of an exponential function that has e as its base.
		FIX	Truncates the decimal part of a numeric value and designates it as an integer.
		INT	Designates the integer value of a formula.
		LOG	Designates the value of a natural logarithm.
		RND	Designates a random number.
		ROT	Designates a value processed by bit rotation.
		SGN	Designates a numeric value code.
		SIN	Designates the value of the sine in a trigonometric function.
		SQR	Designates the value of a square root.
		TAN	Designates the value of a trigonometric function.
		SHA	Designates a value when the memory capacity of a designated value was processed by an arithmetical shift.
		SHT	Designates a value when the memory capacity of a designated value was processed by a logical shift.
	Numeric value search	SEARCH	Searches for a value designated from the element of any array variable, and designates the position of the element.
	Type conversion function	CDBL	Converts an integer and a single-precision real number into a double-precision real number.
		CSNG	Converts an integer and a double-precision real number into a single-precision real number.
		CINT	Converts a single-precision real number value and a double-precision real number value into an integer.
		CIDB	Converts a two-word (32-bit) integer used by a PC CPU into a double-precision real number.
		CISN	Converts a two-word (32 bits) integer used by a PC CPU into a single-precision real number.
PC access function	PC access	PCRD	Reads various data from a PC CPU.
		PCWT	Writes various data to a PC CPU.
			Operates a PC CPU.
Port switching command	Port switching	ZIDV	Sends the INPUT command from a designated console.
		ZODV	Outputs the PRINT command to a designated console.
Buffer access function	Buffer access	GETMEM	Reads data from the buffer, common memory, and extension register (ED) of the AD51H.
		PUTMEM	Writes data to the buffer, common memory, and extension register (ED) of the AD51H.
Clock function command	Clock function	TIME$	Sets and reads a PC CPU time (hour, minute, and second).
		DATE$	Sets and reads the PC CPU date (year, month, day, and day of the week).
Real-time control command	Real-time control	ZEVENT	Sets whether an event occurrence is valid/invalid.
		DEF ZEVENT	Defines an event to synchronize executions between programs.
			Defines an event by the extension relay (EM) of the AD51.
		ZRELEASE	Permits other programs to use a source that was allocated a source number.
		ZRESERVE	Prohibits other programs from using a source that was allocated a source number.
		ZSIGNAL	Makes an event designated by a program occur.
		ZSTART	Starts a designated program.
		ZURGENCY	Changes the priority of a program.

Table 6.2 AD51H-BASIC Commands and Functions (cont.)

Classification		Commands and Functions	Operation Description
Command for real-time control	Real-time control	ZWAIT DELAY	Suspends execution of a program until a designated time passes.
		ZWAIT EVENT	Suspends execution of a program until a designated event occurs.
Real-time control function	Real-time control	ZMESSAGE	Defines a message port.
		ZMESSAGE CLOSE	Closes a message port.
		ZMESSAGE GET	Reads a message from a message port.
		ZMESSAGE KILL	Clears a defined message port.
		ZMESSAGE OPEN	Opens a message port.
		ZMESSAGE PUT	Writes a message to a message port.

6.1.3 AD51H internal memory and memory map

Table 6.3 shows the memory and memory map that can be used with the AD51H.

Table 6.3 AD51H Internal Memory and Memory Map

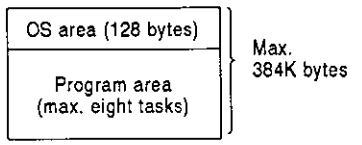
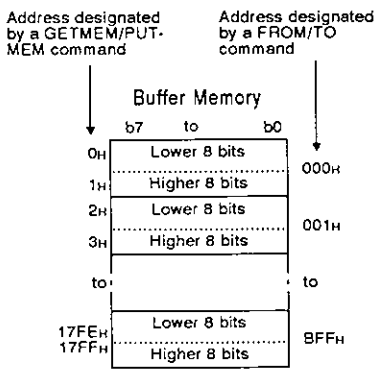
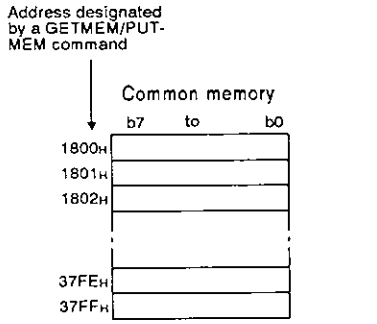
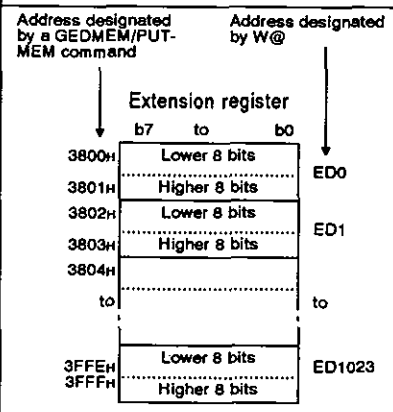
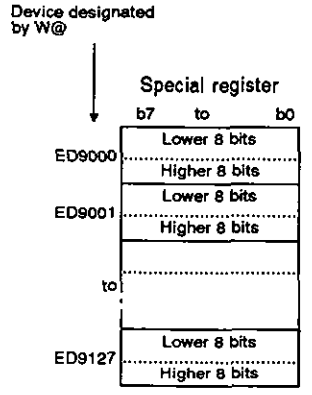
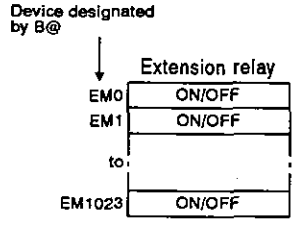
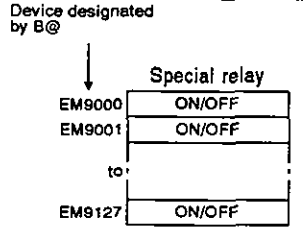
Internal Memory	Description	Memory Map	Reference Section
Execution program area	(1) Memory where a BASIC program executed by the AD51H is stored. (2) The BASIC programs are read from the EP-ROM or a memory card to the execution program area. (The copy target is set by mode setting switch 2.) (3) 64K bytes to 384K bytes can be selected. (Set in units of 64K bytes) (4) 16K, 32K, 48K, or 64K bytes can be set to the task size.		6.6.2
Buffer memory	(1) Memory that can communicate 16 bits of data from/to a PC CPU (2) Read/write of data from/to each task is enabled. (3) Read/write of data is executed in 16-bit units by a FROM/TO command from the PC CPU. (Use X1B as the interlock signal for the execution of the FROM/TO command.) (4) Read/write of data is executed in 8-bit units by a GETMEM/PUTMEM command from a BASIC program. (5) The buffer size is 6K bytes.		6.6.5
Common memory	(1) Memory used in the AD51H (Read/write from/to a PC CPU is disabled.) (2) Read/write of data from/to each task is enabled. (3) Read/write of data is executed in 8-bit units by a GETMEM/PUTMEM command from/to a BASIC program. (4) The size of the common memory area is 8K bytes.		6.6.6

Table 6.3 AD51H Internal Memory and Memory Map (cont.)

Internal Memory	Description	Memory Map	Reference Section
Extension register (ED)	(1) Memory used in the AD51H (Read/write from/to a PC CPU is disabled.) (2) Read/write of data is enabled from/to each task. (3) Read/write of data is executed by a GETMEM/PUTMEM command or special variable W@ from a BASIC program. • GETMEM/PUTMEM : 8-bit unit • W@ : 16-bit unit (4) The number of points in an extension register is 1024 (2048 bytes).		6.6.4
Special register (ED)	(1) An area with a designated use in the internal memory of the AD51H (Read/write from/to a PC CPU is disabled.) (2) Data can be read from each task. (3) Data read is executed in 16-bit units by special variable W@ from a BASIC program. (4) The number of points in the special register is 128.		6.8
Extension relay (EM)	(1) Relay used in the AD51H (Read/write from/to a PC CPU is disabled.) (2) Read/write of ON/OFF data from/to each task is enabled. (3) Read/write of ON/OFF data from/to a BASIC program is executed by special variable B@. (4) The number of points in the extension relay is 1024.		6.6.3
Special relay (EM)	(1) An area with a designated use in the internal memory of the AD51H (Read/write from/to a PC CPU is disabled.) (2) Data can be read from each task. (3) Read of ON/OFF data from/to a BASIC program is executed by special variable B@. (4) The number of points in the extension relay is 128.		6.7

6.1.4 I/O (X, Y) list

Fig. 6.1 shows how to determine a device number when the AD51H is installed in slot No. 0 or slot No. 1 of the main base unit.

- The I/O number used with a sequence program varies according to the slot in which the AD51H is installed.
- The value to be designated by special variable B@ of a BASIC program of the AD51H is fixed from &H00 to &H1Fs.

Table 6.4 gives device numbers and signal names. (Sections 6.10.2 and 6.10.3 give details about signal names.)

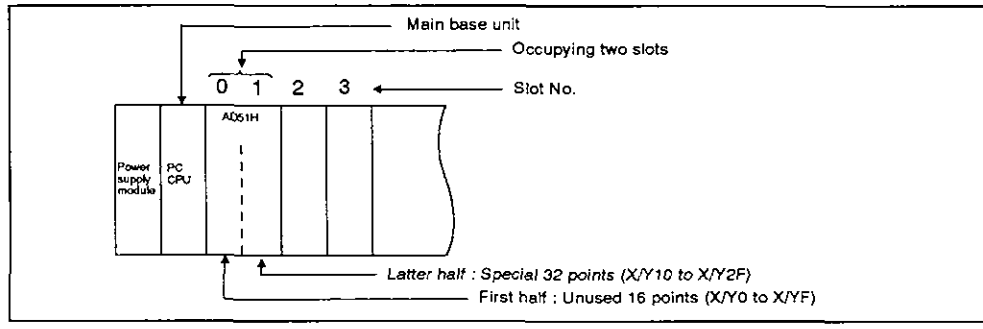


Fig. 6.1 Device Numbers Set to a PC CPU Time When the AD51H is Installed in Slot No. 0 or Slot No. 1

Table 6.4 AD51H I/O Device Numbers and Signal Names

Signal Direction AD51H → PC CPU			Signal Direction PC CPU → AD51H			
Input Number Used by a PC CPU	Value Designated by Special Variable B@	Signal Name	Output Number Used by a PC CPU	Value Designated by Special Variable B@	Signal Name	
X00 to X0F	—	Unusable	Y00 to Y0F	—	Unusable (Usable as a substitute for the internal relay (M).)	
X10, X1A	&H00, &H0A	General-purpose input	Y10 to Y20	&H00 to &H10	General-purpose output	
X1B	&H0B	Multitask execution starts				
X1C	&H0C	Multitask execution stops				
X1D	&H0D	AD51H system down				
X1E, X1F	&H0E, &H0F	Unusable				
X20 to X2F	&H10 to &H1F	General-purpose input	Y21	&H11	General-purpose output/start BASIC task No. designation *1	BASIC task 1
			Y22	&H12		BASIC task 2
			Y23	&H13		BASIC task 3
			Y24	&H14		BASIC task 4
			Y25	&H15		BASIC task 5
			Y26	&H16		BASIC task 6
			Y27	&H17		BASIC task 7
			Y28	&H18		BASIC task 8
			Y29	&H19	Program start signal (Unusable for general-purpose output)	
			Y2A Y2F	&H1A &H1F	Unusable	

REMARK

*1 Designate the starting task number when a task is set to "start by an interrupt from PC CPU" by the multitask setting.
Designate a general-purpose output for a task which is not set to "start by an interrupt from PC CPU".

6.1.5 Uses and settings of switches

Table 6.5 gives the uses and settings of switches that can be set on the AD51H.

Table 6.5 Uses and Settings of Switches

Printed Mark	Name		Use	Reference Section
MODE SW1 (on the front of the AD51H)	Mode setting switch 1		Setting of the operating modes • 0/1 : Execution mode • 2/3 : Multitask debugging mode • 4 : Programming mode	4.2.1
MODE SW2 (on the front of the AD51H)	Mode setting switch 2	SW No.	—	—
		1	Setting of a console/debugger • The use of a PC/AT, VG-620, VT-382, or VT-220 is set by the ON/OFF combinations of SW1 to SW5.	4.2.2
		2		
		3		
		4		
		5		
		6	Valid/invalid setting of [Ctrl] + [Pause] / [Ctrl] + [C] keys • OFF : Invalid • ON : Valid	4.2.3
		7	Setting of the read target of an execution program • OFF : Memory card • ON : EP-ROM	4.2.4
		8	Setting of the task switching time • OFF : 50ms • ON : 100ms	4.2.5
SW2 (the left side of the AD51H)		1	Setting of the AD51H operation when a PC CPU is reset • OFF : AD51H continues operation. • ON : AD51H is also reset.	4.2.6
		2	Setting of the access enable time from a PC CPU after resetting the AD51H 2 3 : SW NO. OFF OFF : 200 msec ON OFF : 500 msec OFF ON : 1000 msec ON ON : 2000 msec	4.2.7
		3		
		4	Turn this OFF.	—
		5		
		6	Setting of the type of EP-ROM to be used • OFF : 64K bytes/128K bytes • ON : 256K bytes	4.2.8
		7	Turn this OFF.	—
		8		

Table 6.5 Uses and Settings of Switches (cont.)

Printed Mark	Name		Use			Reference Section
M.PRO (on the left side of the AD51H)	Memory-protect range setting switch	SW No.	ON	OFF	Protected range	4.2.9
		1	Not protected	Protected	00000H to 0FFFFH	
		2			10000H to 1FFFFH	
		3			20000H to 2FFFFH	
		4			30000H to 3FFFFH	
		5			40000H to 4FFFFH	
		6			50000H to 5FFFFH	
		7			60000H to 6FFFFH	
		8			70000H to 7FFFFH	
M-PROTECT (on the front of the AD51H)	Memory-protect keyswitch		Valid/invalid setting of the memory-protect range set by the memory-protect range setting switch • OFF : Invalid (not protected) • ON : Valid (protected)			4.1
RUN-STOP-RESET (on the front of the AD51H)	RUN switch		Execution/stop of multitasking and hardware reset • RUN : Execution of multitasking • STOP : Stop of multitasking • RESET : Hardware reset			4.1
MEMORY CARD (on the front of the AD51H)	Memory card access switch		Inserting/removing a memory card when the power is ON • OFF : When OK is displayed on the LED display, a memory card can be inserted or removed. • ON : Access to the memory card is enabled. (A memory card cannot be inserted or removed.)			

6.1.6 Relationship between the conditions for starting the AD51H and the execution program area of a memory card

When storing a BASIC program to a memory card by an MSAVE system command, BOOT is set as the condition for starting each task during multitasking. To set the starting condition to other than BOOT, use the SET system command.

Conditions for storing BASIC programs to a memory card vary according to the starting condition, as shown in Table 6.6.

Table 6.6 Relationship between the AD51H Starting Conditions and the Execution Program Areas of Memory Cards

Starting Condition		BASIC Program Storage Area	BASIC Program Read Timing	Reference Section
Multitask Setting	Symbol			
Started by multitask execution	START	• Execution program area of a memory card (the designated BASIC task number area) • Execution program area of the EP-ROM (the designated BASIC task number area)	• When power goes ON or is reset in the system mode • During multitasking in the execution or multitask debugging mode (RUN keyswitch : RUN)	6.4.2
Started by an interrupt from a PC CPU	IT	• Execution program area of a memory card (the designated BASIC task number area) • Execution program area of the EP-ROM (the designated BASIC task number area)		6.4.3
Started by other start requests from the task	BOOT	• Execution program area of a memory card (the designated BASIC task number area) • Execution program area of the EP-ROM (the designated BASIC task number area)		6.4.4
	ON	• File area in the memory card (The BASIC programs are stored as files.) • File area of the user's FD and the HD of a PC/AT (The BASIC programs are stored as files.)	• When executing by the ZSTART command (The program in the file name designated by the ZSTART command is read.)	6.4.4
Without multitask setting	OFF	—	• Reading of a BASIC program is not executed because there is no multitask setting	—

POINT

The BOOT and ON starting conditions are defined as follows.

- BOOT : The execution program is stored in the designated BASIC task number area of the memory card.
- ON : Only the memory capacity of the designated BASIC task number area of the memory card is set. The execution program is not stored in the BASIC task number area.

REMARK

This manual (online programming debug) gives details about MSAVE and SET system commands.

6.2 Memory Card Specifications, Memory Map, and Memory-Protect Range

This section explains the specifications of the memory cards that can be used with the AD51H, the memory map, and the memory-protect range.

6.2.1 Memory card specifications

Table 6.7 gives memory card specifications.

Table 6.7 Memory Card Specifications

Item	Memory Card Model Name	
	A6MEM-256KA-W	A6MEM-512KA-W
Memory capacity (bytes)	256K	512K
Access time (ns)	200	
Memory-protect	Enabled (by the WRITE PROTECT switch of the memory card)	
Backup function	By a built-in battery Battery: BR2016 *1	
Battery life (years)	5	
Outside dimensions	86.5 (3.40)(H) x 54 (2.12)(W) x 3 (0.12)(D)	

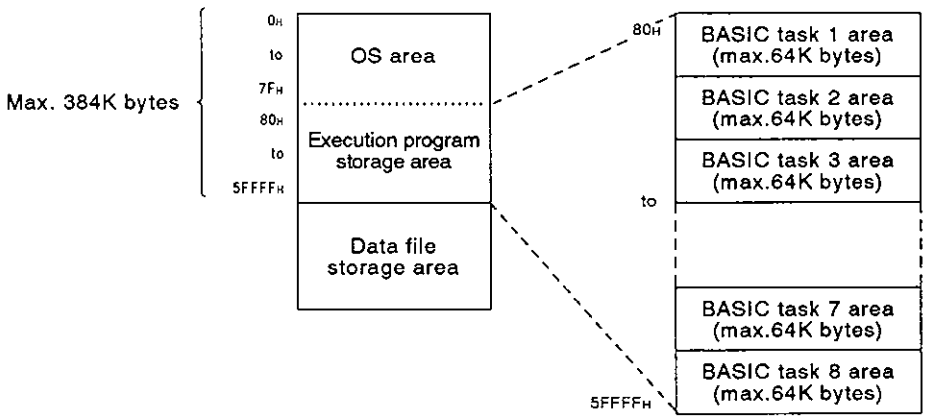
6.2.2 Memory map

(1) Memory map of a memory card installed in MEMORY CARD1

The execution program and file where a BASIC program is registered can be stored in the memory card installed in MEMORY CARD1.

(a) An execution program with (max.) eight tasks (384K bytes) can be stored in a memory card.

(b) Any area (except the execution program storage area) can be used for a BASIC program data file.



REMARK

*1 When purchasing a new battery, consult your nearest Mitsubishi representative for details

6.2.3 Memory-protect range

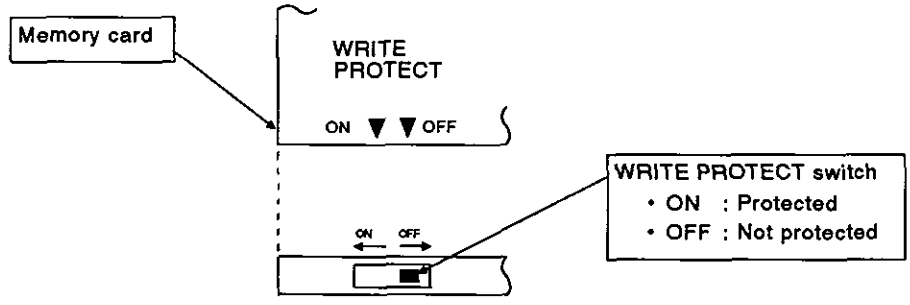
Set the memory-protect range by using either of the following methods:

- Batch protect by the memory card WRITE PROTECT switch
- Split protect by the AD51H memory-protect range setting switch

(1) Batch protect by the WRITE PROTECT switch of a memory card

The memory of a memory card is protected by the WRITE PROTECT switch of the memory card.

- If the WRITE PROTECT switch is set to ON, the memory in all areas is protected.
- If the WRITE PROTECT switch is set to OFF, the memory is not protected.

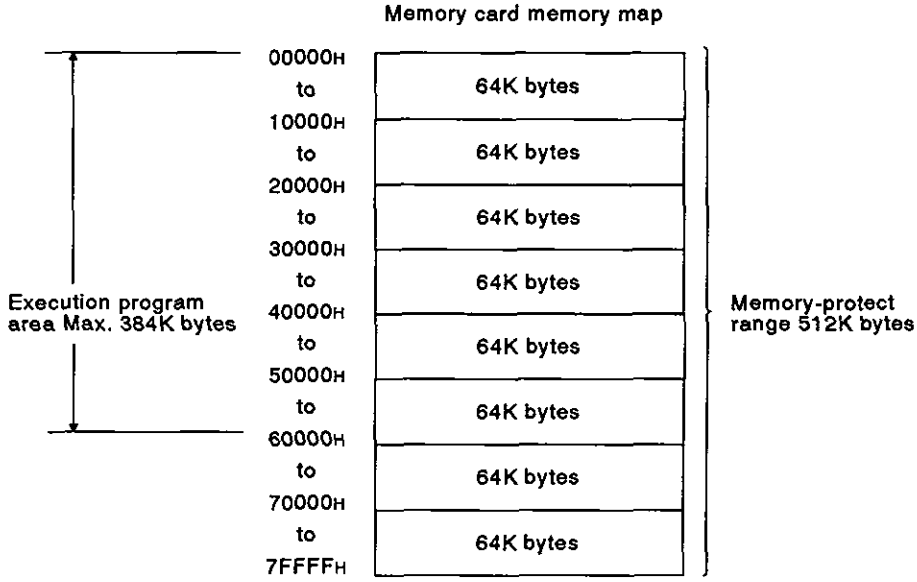


(2) Split protect by a memory protect range setting switch of the AD51H

There is a memory-protect range setting switch on a memory card installed in MEMORY CARD1 on the left side of the AD51H.

The memory-protect range has 512K bytes from address 0H to 7FFFH, and the range can be divided into 64K-byte areas.

The memory-protect range is as shown below.



6.3 AD51H Operations When the Power is Turned ON

When power to the AD51H goes ON, the operations change according to the following modes (set by mode setting switch 1).

(1) Programming mode

(2) Multitask debugging mode

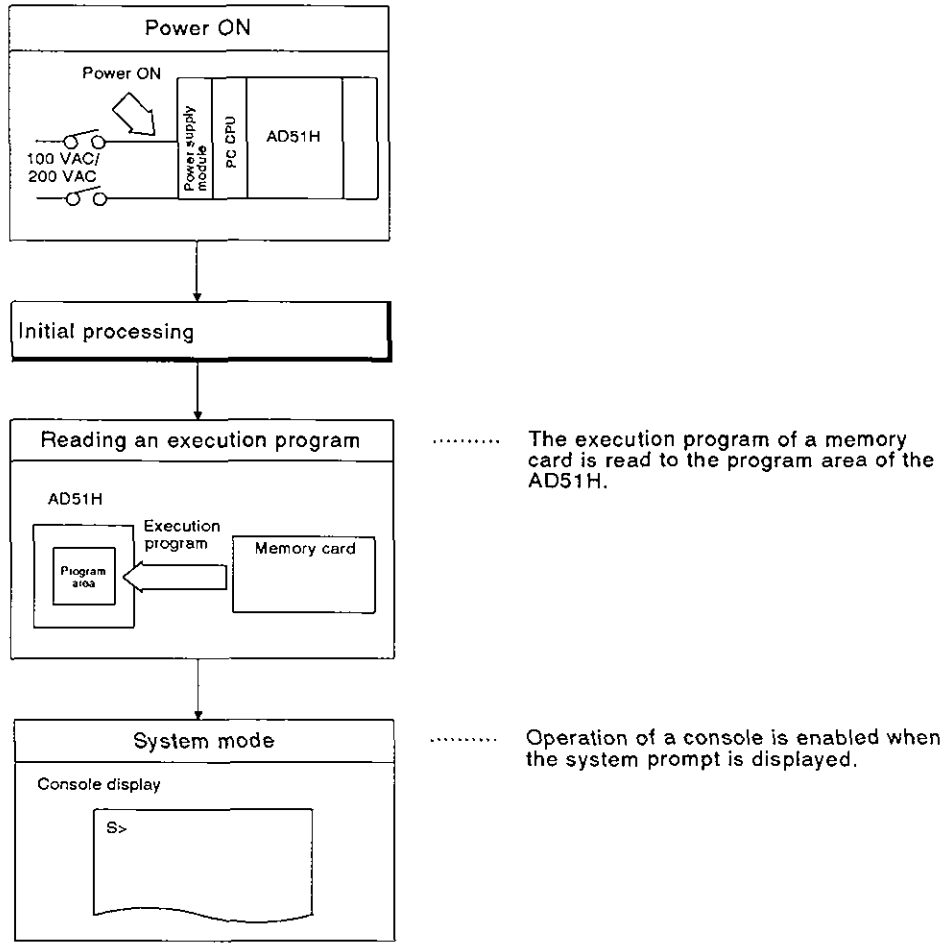
(3) Execution mode

This section explains the operations of the AD51H in the modes set by mode setting switch 1 when the power goes ON.

6.3.1 Programming mode

If the power supply is turned ON or the AD51H is reset in the programming mode, the mode is switched to the system mode after reading an execution program from the memory card to the program area of the AD51H.

(see Section 6.6.2.)



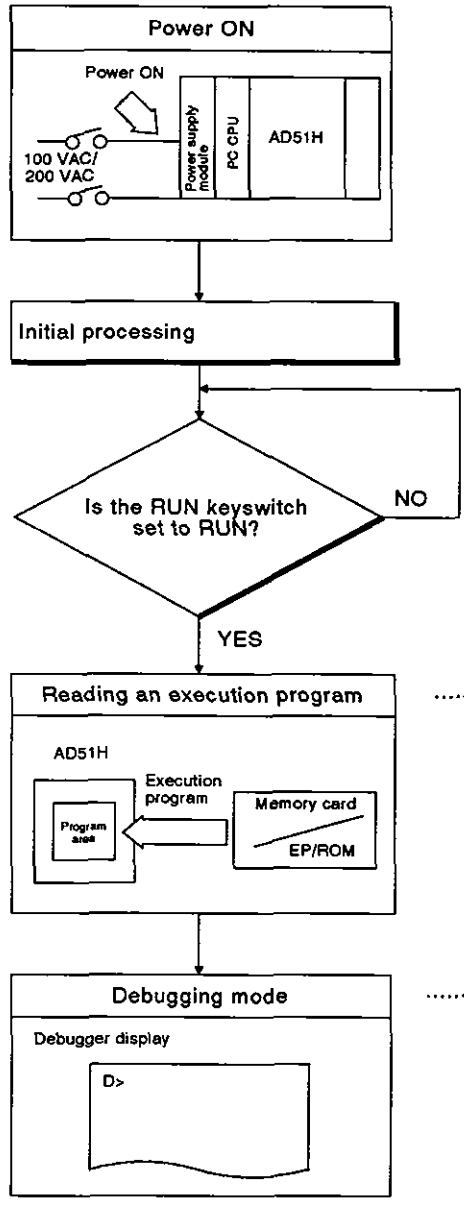
..... The execution program of a memory card is read to the program area of the AD51H.

..... Operation of a console is enabled when the system prompt is displayed.

6.3.2 Multitask debugging mode

After (a) turning ON the power supply or resetting the AD51H in the multitask debugging mode, and (b) when the RUN keyswitch is set to RUN, an execution program is read from a memory card or the EP-ROM to the program area of the AD51H, and the mode is switched to the debugging mode.

(see Section 6.6.3.)



..... The execution program is read to the execution program area of a memory card or the EP-ROM.

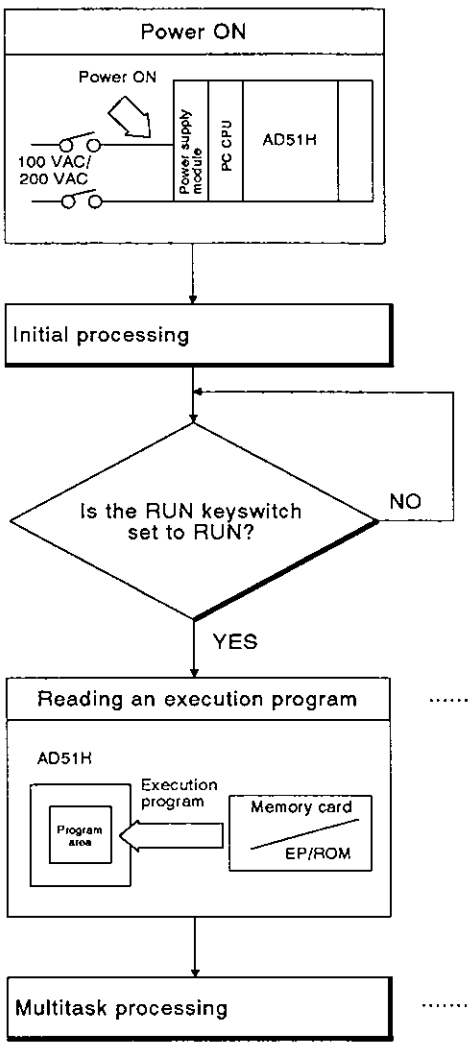
..... The multitask operation by a debugger is enabled when a debug command is received.
A read execution program is executed according to the contents of multitask setting.

- Started by turning ON the power or resetting the AD51H
- Started by an interrupt from a PC CPU
- Started by other start requests from a BASIC program

6.3.3 Execution mode

After(a) turning ON the power supply or resetting the AD51H in the execution mode, and (b) when the RUN keyswitch is set to RUN, the execution program is read from the memory card or the EP-ROM to the program area of the AD51H, and the multitask processing is executed.

(see Section 6.6.3.)



..... Reads the execution program to the execution program area of a memory card or the EP-ROM.
However, the execution program of a task of which multitask setting is set to ON is not read to the AD51H.

..... A read execution program is executed according to the multitask setting contents.

- Started by turning ON the power or resetting the AD51H
- Started by an interrupt from a PC CPU
- Started by other start requests from a BASIC program

6.4 Executing Tasks (Created BASIC Programs)

This section explains the execution of tasks when the AD51H is in the execution mode or multitask debugging mode.

6.4.1 Conditions for starting and executing tasks

When all the following conditions are satisfied, the AD51H goes into the task execution enabled state.

(The actual start of a task depends on the starting condition when multitasking is set.)

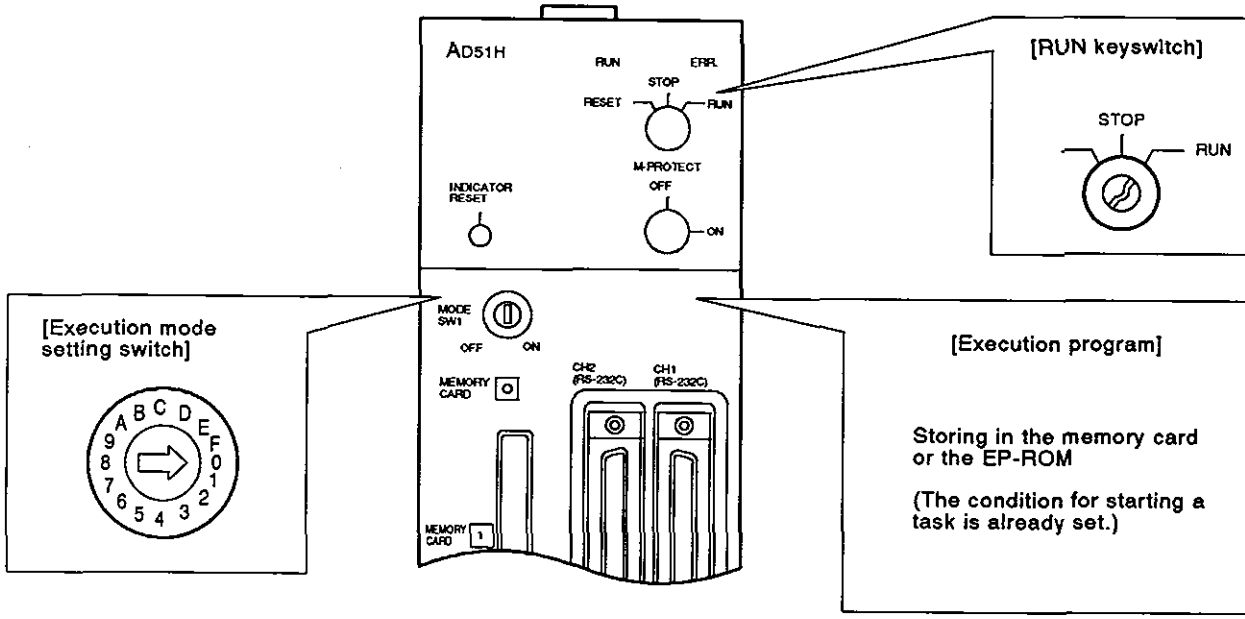
(1) Task execution conditions

The following gives the conditions for starting a task:

(a) A task starting condition is set with multitask setting.

(b) The mode setting switch 1 is set to "online mode (0, 1)" or "debugging mode (2, 3)".

(c) The RUN keyswitch is set to RUN.



(2) Condition for starting a task

Each task is executed according to the set starting condition.

The following starting conditions can be set with the AD51H.

(a) Starting by turning ON the power or resetting the AD51H (See Section 6.4.2)

(b) Starting by an interrupt from a PC CPU (See Section 6.4.3)

(c) Starting by other start requests from a BASIC program (See Section 6.4.4)

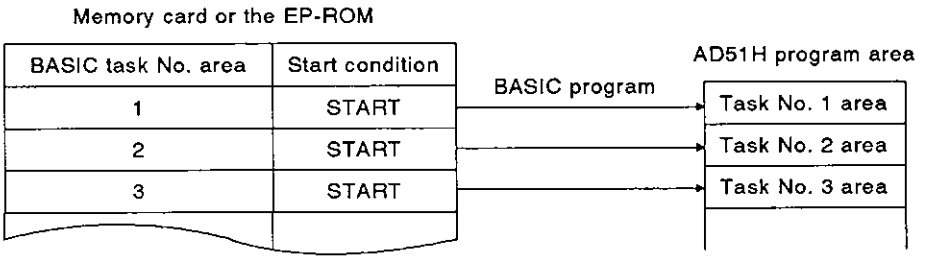
6.4.2 Starting by turning ON the power or resetting the AD51H

In this case the starting condition is set to START by the multitask setting.

When the execution conditions explained in Section 6.4.1 have been satisfied, the task is started.

(1) Processing when the power is turned ON or reset

When the power is turned ON or reset, the designated task is executed after reading an execution program stored in a memory card or the EP-ROM to the program area of the AD51H.



(2) Execution of a task

A task started when the power is turned ON or reset can be executed by the created program as shown below.

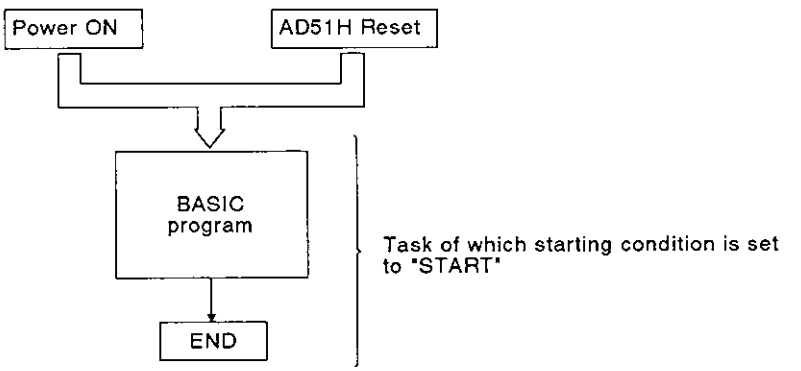
1) Executed only once

2) Always executed

(a) When the task is executed only once

If an END command is written at the end of a BASIC program, the task will only be executed once.

The task is started by turning ON the power or resetting the AD51H. When the END command has been executed, the operation stops.

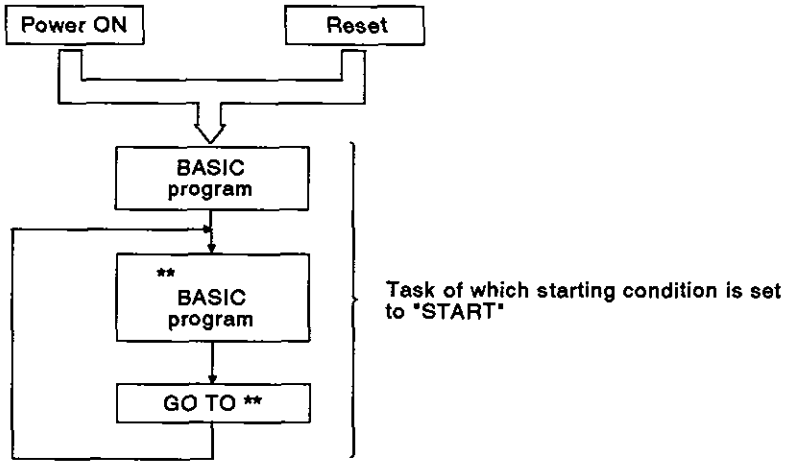


(b) When the task is always executed

In this case, the program operation is jumped to a designated line to the end of a BASIC program by using the GOTO command (instead of the END command).

The task is started by turning ON the power or resetting the AD51H. Then, when the GOTO command has been executed, the operation jumps to the designated line.

Therefore, the operation can be executed over and over between the designated line and the GOTO command.



6.4.3 Starting by an interrupt from a PC CPU

In this case, the starting condition is set to IT by the multitask setting.

When (a) the execution conditions explained in Section 6.4.1 have been satisfied, and (b) a start request is sent from a PC CPU, the task is started.

The start request from a PC CPU can be done by using either of the following:

1) Start task number designation signal
2) Task start signal

(Section 6.10.3 gives details about these signals.)

(1) Processing when the power is turned ON or the AD51H is reset

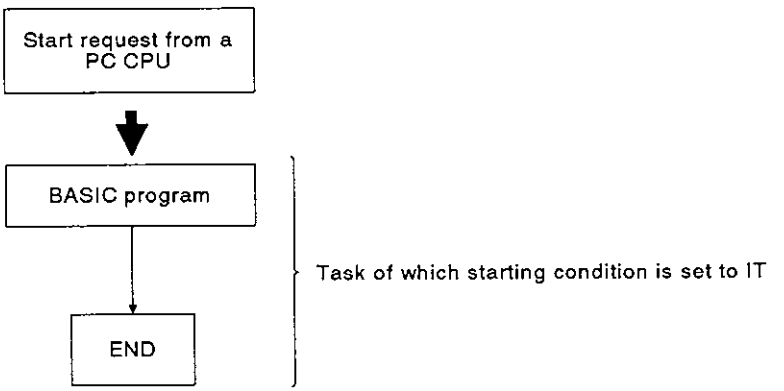
When the power is turned ON or the AD51H is reset, the execution program stored in a memory card or the EP-ROM is read to the AD51H internal memory.

The read BASIC program is executed by the start request from a PC CPU.

(2) Execution of a task

(a) When the power is turned ON or the AD51H is reset, the read BASIC program is started by sending the task start signal from a PC CPU and starting up the AD51H.

When the task start signal is sent, the task of which start task number target signal number is turned ON starts.



(b) Write the END command at the end of a BASIC program and execute it with the task started by an interrupt from a PC CPU.

When the END command has been executed, the interrupt from a PC CPU can be reset, and the following interrupt can be received.

6.4.4 Starting by other start requests from a BASIC program

In this case, the starting condition is set to BOOT or ON by the multitask setting.

When (a) the execution conditions explained in Section 6.4.1 have been satisfied, and (b) the start request is sent from another BASIC program by the ZSTART command, the task is started.

(The AD51H-BASIC Programming Manual gives details about the ZSTART command.)

(1) Processing when the power is turned ON or the AD51H is reset

(a) When the power is turned ON or the AD51H is reset, if the starting condition is set to BOOT, the execution program stored in a memory card or the EP-ROM is read to the AD51H internal memory.

The read BASIC program is executed by a start requests from another task.

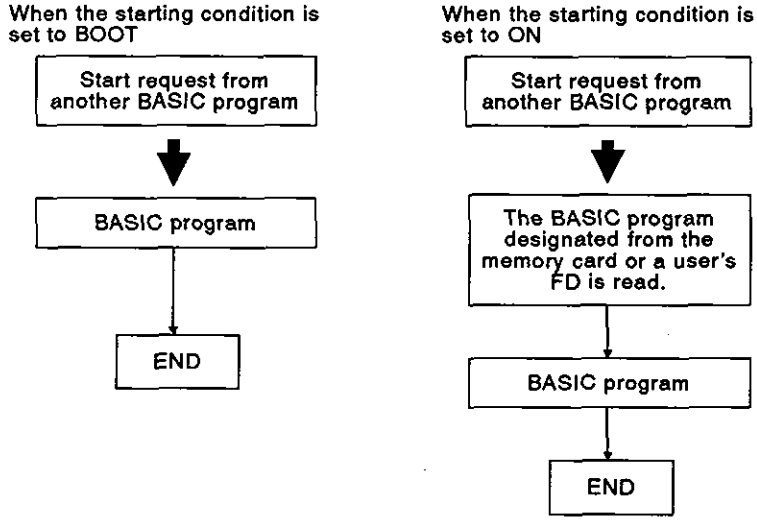
(b) When the power is turned ON or the AD51H is reset, if the starting condition is set to ON, the execution program stored in a memory card or the EP-ROM is not read to the AD51H internal memory.

If the start request is sent from another task, the designated BASIC program is read from the memory card or a user's FD and is executed.

(2) Execution of a task

(a) When (a) the starting condition is set to BOOT, and (b) if the task has been designated by the ZSTART command from another BASIC program, the task starts.

If the starting condition is set to ON, the task is started after reading the BASIC program designated from the memory card or a user's FD (whenever it is designated by the ZSTART command from another BASIC program).



(b) Execute the END command at the end of a BASIC program for the task to be started by the start request from another BASIC program.

If the END command is executed, the start request can be reset, and the following interrupt can be received.

6.5 Execution of a BASIC Program by Multitask Processing

This section tells how to execute BASIC programs of the AD51H.

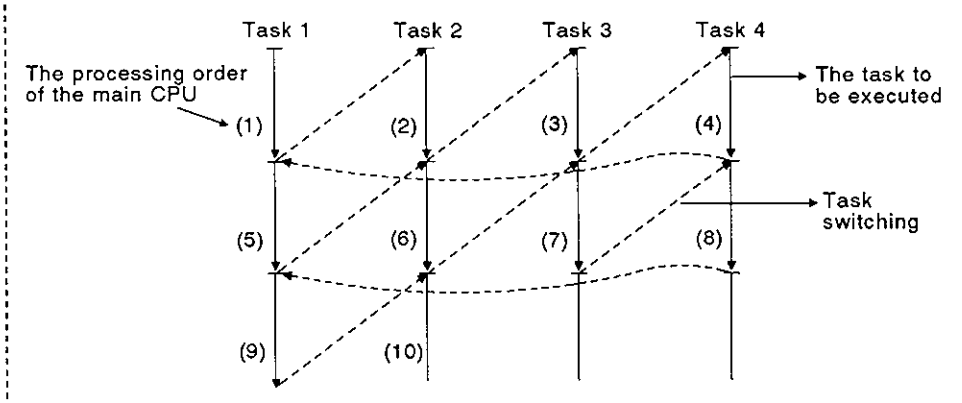
6.5.1 Multitask processing

The AD51H operates BASIC programs created by the user.

The AD51H can do parallel operations of BASIC programs by multitask processing. Multitask processing consists of switching the execution of several BASIC programs (max. eight tasks) one after another. This is done by the main CPU (central processing unit).

Example:

Processing (in this case, four tasks)



The execution of BASIC programs is switched by the multitask processing of the AD51H when the following state occurs:

- Set task schedule time (50 msec or 100 msec) has passed (see Section 4.2.5).
- An I/O (data communications) data command to a screen, keyboard, disk, peripheral or external device has been executed.
- The command that controls (interrupts, stops or completes) execution of a BASIC program has been executed.
- The command that controlled multitasking (changes the synchronism, start, and priority of execution) has been executed.
- The execution interruption state of a program was canceled. (Either the I/O was completed or the designated time was passed.)

REMARK

The AD51H-BASIC Programming Manual gives details about multitask processing and program execution switching commands.

6.5.2 BASIC program priorities

A created BASIC program can be given priority by the AD51H.

When BASIC program execution is started, all BASIC programs are set to the same priority.

Therefore, created BASIC programs are executed equally.

However, the priority can be changed by the ZURGENCY command. If the priority is changed, only high-priority BASIC programs will be executed.

(Low-priority BASIC programs will not be executed.)

6.5.3 Synchronizing the executions of BASIC programs (event control)

Multitasking executes each BASIC program independently.

Therefore, when data transfer is executed between the BASIC programs, the receiver needs to read data after confirming that the sender transmitted that data.

Event control synchronizes execution of the BASIC programs of the sender and receiver.

Event control puts a BASIC program of the receiver signal in the wait state (WAIT), sends a signal (event) after the sender finishes transmitting data, and continues the BASIC program of the receiver.

Therefore, data transmitted by the sender can be read by a BASIC program of the receiver.

Event control is executed by the following commands:

- DEF-ZEVENT : Definition of event
- ZEVENT ENABLE : Event occurrence is valid.
- ZWAIT EVENT : Event wait
- ZSIGNAL : Event occurrence

REMARK

The AD51H-BASIC Programming Manual gives details about these event control commands.

6.5.4 Sharing of sources during multitasking

A source cannot be used simultaneously by several BASIC programs.

Exclusive control of a source prevents another BASIC program from using the same source.

Since exclusive control of this source is executed according to rules made by programmers, if these rules are not followed, exclusive control of a source cannot be achieved.

Exclusive control of a source is executed by the following commands:

- ZRESERVE : Requests exclusive control of a source.
- ZRELEASE : Completes exclusive control of a source.

REMARK

In this context, "source" means all the hardware and software that can be used by a BASIC program (the screen, printer, file, disk, memory, etc).

6.6 AD51H Internal Memory Areas

The memory areas of the AD51H, given below, are available to the user.

(1) Memory area where the execution program is stored

- Program area ..... (see Section 6.6.2)

  This memory stores BASIC programs executed by the AD51H.

  This program area can be divided into eight task areas and has a total of 384K bytes.

(2) Memory area used for read/write of ON/OFF data

- Extension relay (EM) : 1024 points (EM0 to EM1023)
  ..... (see Section 6.6.3)

(3) Memory area used to read/write 16-bit data

- Extension register (ED) : 1024 points (ED0 to ED023)
  ..... (see Section 6.6.4)
- Buffer : 6K bytes ..... (see Section 6.6.5)
- Common memory : 8K bytes ..... (see Section 6.6.6)

Since the memory areas of (1) to (3) above are not battery-backed, if the AD51H power goes OFF, written BASIC programs and all data will be cleared.

Make sure to store all created BASIC programs in the memory card in MEMORY CARD1 before turning OFF the power.

Store all necessary data to the memory card(s) in MEMORY CARD1 or MEMORY CARD2 before turning OFF the power.

6.6.1 Memory map

When executing read/write of data by using extension register (ED), a buffer, or a common memory, designate addresses by using a BASIC program GETMEM/PUTMEM command.

Figure 6.2 shows the memory maps of a memory designated by a GETMEM/PUTMEM command.

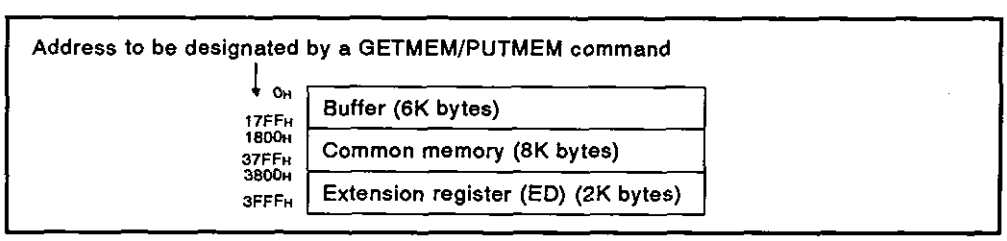


Fig. 6.2 Memory Map

REMARK

The AD51H-BASIC Programming Manual gives details about GETMEM/PUTMEM commands.

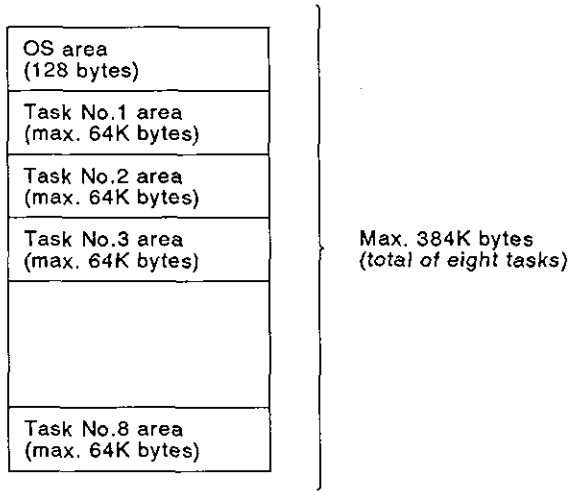
6.6.2 Program area : This is not backed up by a battery

BASIC programs created by a console and an execution program read from the memory card or the EP-ROM are stored in the program area.

(1) Memory capacity

The program area is (max.) 384K bytes.

384K bytes are divided into eight BASIC program areas (task 1 to task 8).

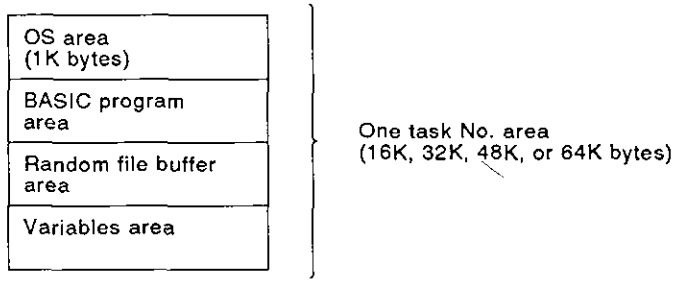


(2) BASIC program capacity

The BASIC program capacity can be set to 16K, 32K, 48K, or 64K bytes for each program.

(3) Composition of each task number area

Each task number area includes a BASIC program area, an OS area, a variables area, and a random file buffer area.



POINT

The execution program of a memory card installed in MEMORY CARD1 is read to the program area of the AD51H in the following cases:

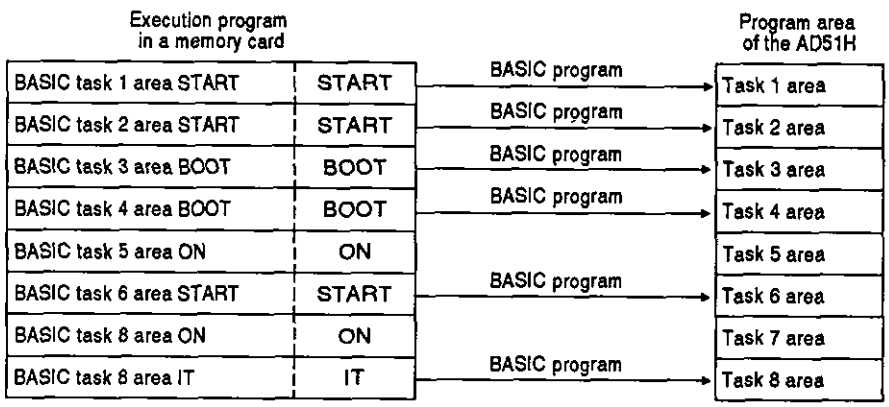
(a) Program mode:
When the power is turned ON or the AD51H is reset

(b) Multitask debugging mode

(c) Execution mode

} When the RUN keyswitch is set to RUN after power is turned ON or the AD51H is reset

However, data in an area where multitask setting is ON will not be read to the program area of the AD51H.



6.6.3 Extension relay (EM) : This is not backed up by a battery

Since the extension relay is the relay used for communicating ON/OFF data between tasks in the AD51H, it cannot output ON/OFF data to an external device.

(1) ON/OFF state of the extension relay and reading of ON/OFF data

(a) The BASIC program reads the ON/OFF state of the extension relay and turns ON/OFF the extension relay using special variable B@.

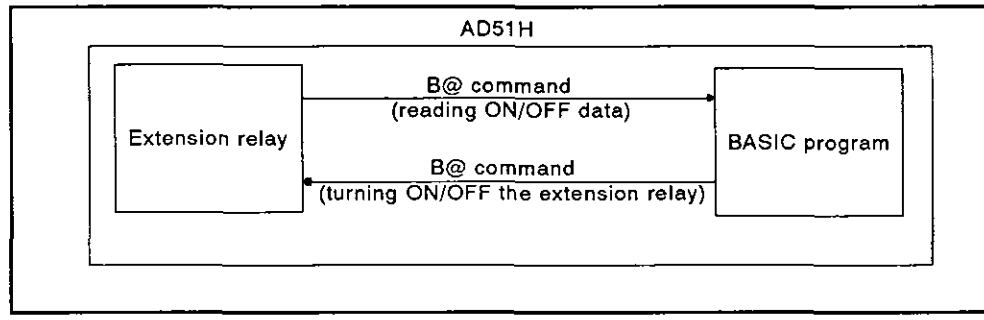


Fig. 6.3 Turning ON/OFF the Extension Relay and Reading ON/OFF Data

(b) ON/OFF data can be read from any task and the data can be used with a BASIC program while doing multitasking of the BASIC program when the BASIC programs are being executed by multitasking.

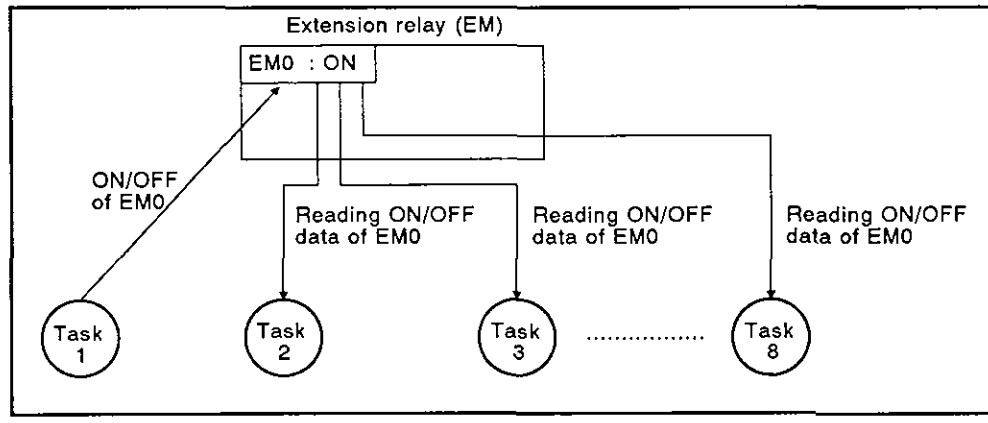


Fig. 6.4 ON/OFF State of the Extension Relays Controlled by the BASIC Programs

(2) Number of extension relay points

The AD51H extension relay has 1024 points.

REMARK

The AD51H-BASIC Programming Manual gives details about special variable B@.

6.6.4 Extension register (ED) : This is not backed up by a battery

The extension register is the internal memory in which one-word (16-bit) data can be stored. *1

Data can be stored in the AD51H and one-word data can be communicated between tasks by using an extension register. *2

Communications between a PC CPU and the extension register cannot be executed by using a FROM/TO command.

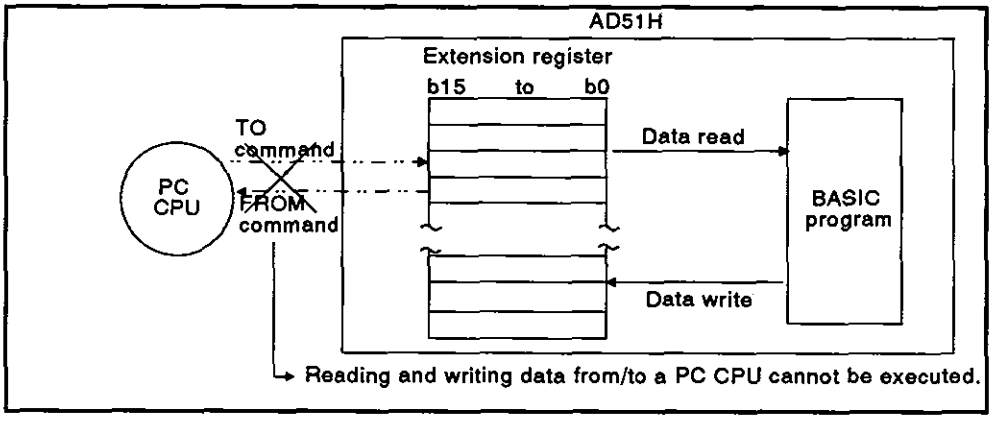


Fig. 6.5 Reading and Writing Data

(1) Reading and writing data from/to the extension register

Reading and writing of data from/to the extension register can be executed by using special variable W@ or a GETMEM/PUTMEM command.

This section gives how to designate the extension register by using special variable W@ or a GETMEM/PUTMEM command.

(a) Reading and writing of data by special variable W@

Designate the extension register from ED0 to ED1023 by using special variable W@.

Designated in one-word units.

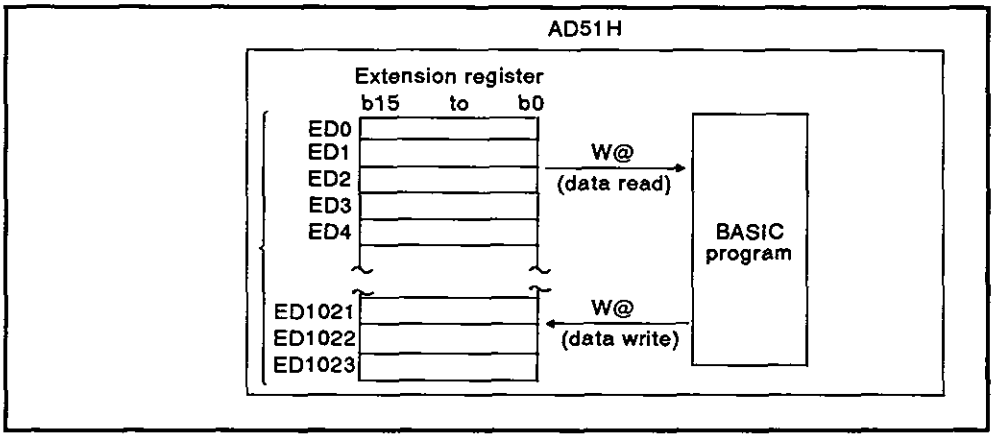


Fig. 6.6 Reading and Writing Data by Using Special Variable W@

REMARK

(1) *1 : Since extension register data is not battery-backed, if the PC power supply is turned OFF or the PC is reset, any written data will be cleared.

Store data in a memory card before turning OFF the power or resetting the PC.

(2) *2 : The buffer (see Section 6.6.5) and common memory (see Section 6.6.6) can be used for communicating one-word data between tasks.

(b) Reading and writing data by a PUTMEM/GETMEM command

Designate an extension register address by using a PUTMEM/GETMEM command.

Designate addresses from 3800H to 3FFFH in one-byte units.

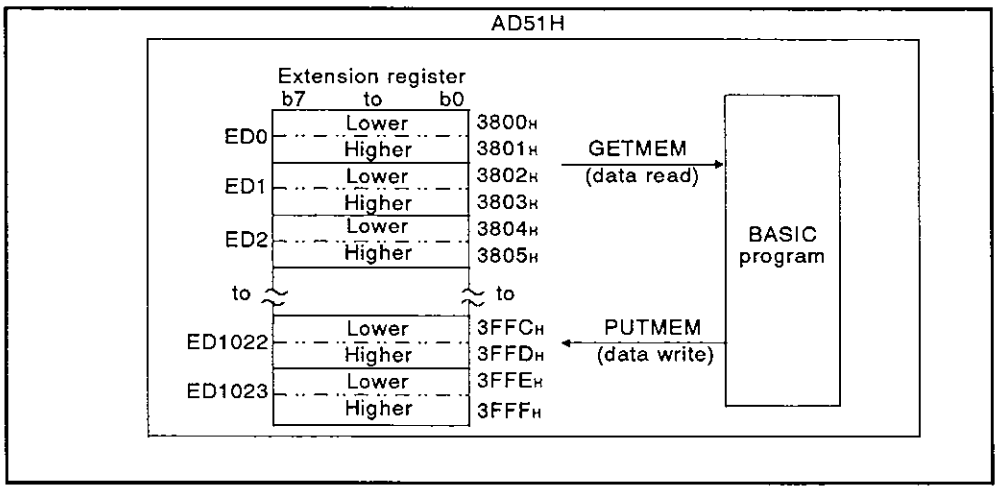


Fig. 6.7 Reading and Writing Data by Using a PUTMEM/GETMEM Command

(2) While executing BASIC programs by multitasking, data can be read or written from/to any task.

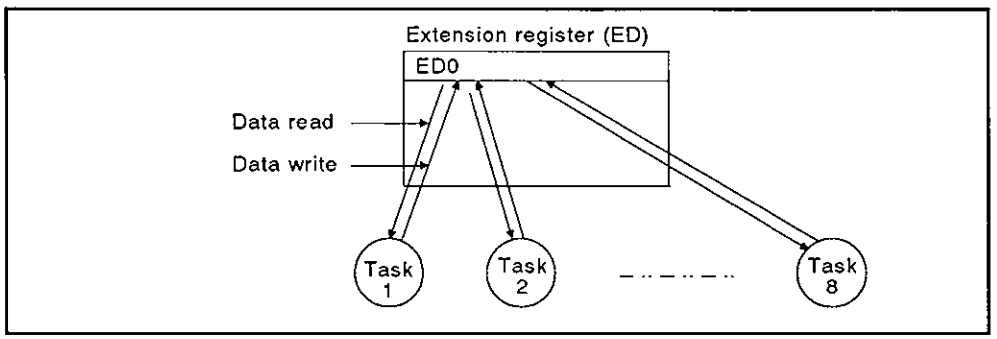


Fig. 6.8 Using an Extension Register with a BASIC Program

(3) Number of extension register points

The extension register of the AD51H has 1024 points (from ED0 to ED1023).

(Extension register numbers ED0 to ED1023 are displayed in decimal.)

6.6.5 Buffer for communications with a PC CPU : This is not backed up by a battery

The buffer is used for data communications between the AD51H and a PC.

Data written to the buffer from a PC CPU can be read during multitasking.

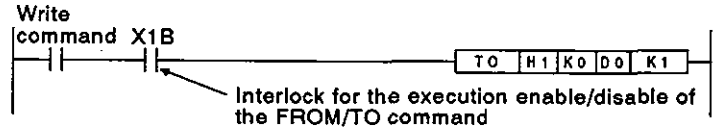
Data written to the buffer during multitasking can be read from a PC CPU.

(1) Read/write of data

(a) Read/write of data from/to the buffer is executed by using a FROM/TO command from a PC CPU.

POINT

To read/write data from/to the buffer from a PC CPU, use the multitask execution start signal (X1B) as an interlock.



(b) Read/write of data from/to the buffer of the AD51H is executed by using a GETMEM/PUTMEM command with a BASIC program.

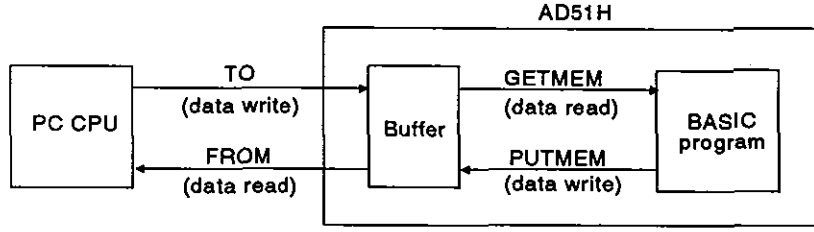


Fig. 6.9 Buffer Data Communications

(2) Buffer capacity

The AD51H buffer consists of 3K words (6K bytes).

(3) Buffer structure

A buffer address consists of 16 bits.

All areas can be used.

REMARK

The AD51H-BASIC Programming Manual gives details about GETMEM/PUTMEM commands.

(4) Buffer addresses

The buffer addresses designated by a PC CPU are different from those designated by a BASIC program, as explained below.

(a) When designating addresses from a PC CPU

Designate addresses from a PC CPU in one-word units.

Use hexadecimal numbers to designate buffer addresses 0H to BFFH.

(b) When designating addresses by a BASIC program

Designate buffer addresses by a BASIC program in one-byte units.

Use hexadecimal numbers to designate buffer addresses 0H to 17FFH.

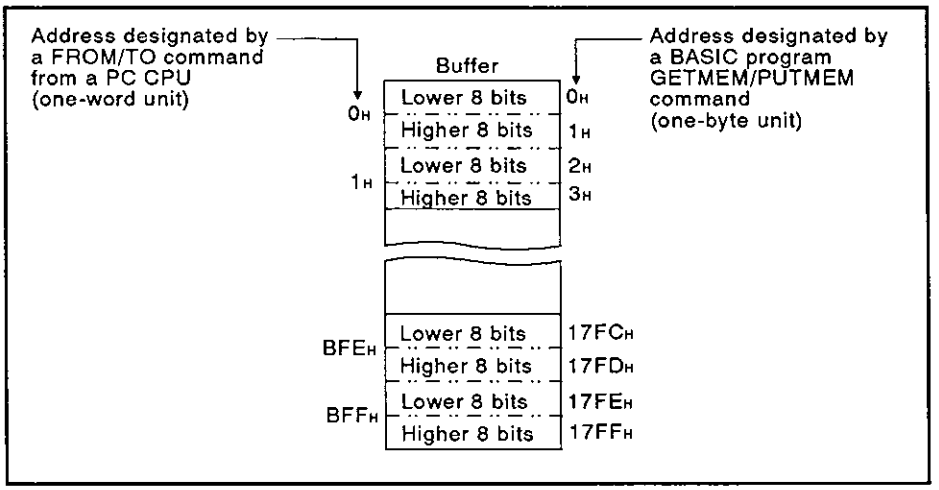


Fig. 6.10 Buffer Addresses

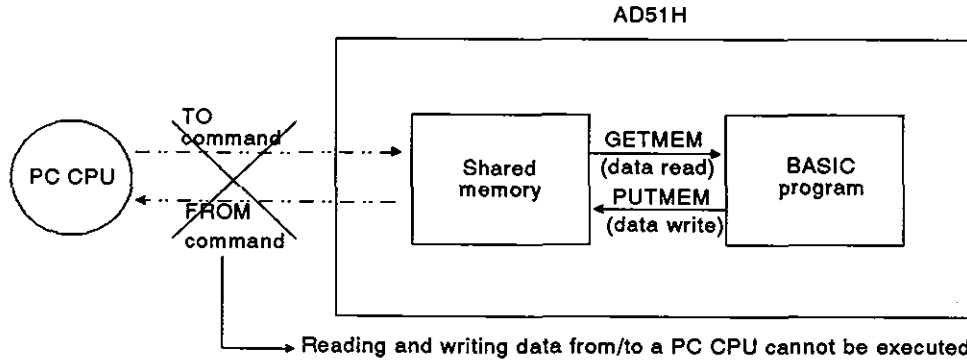
6.6.6 Common memory shared between tasks : This is not backed up by a battery

The common memory can be used for data communications between tasks in the AD51H.

Read/write of data cannot be executed from/to a PC CPU.

(1) Read/write of data from/to the common memory

(a) Use a BASIC program GETMEM/PUTMEM command to read/write data from/to the common memory.



(2) Common memory capacity

The AD51H common memory consists of 8K bytes.

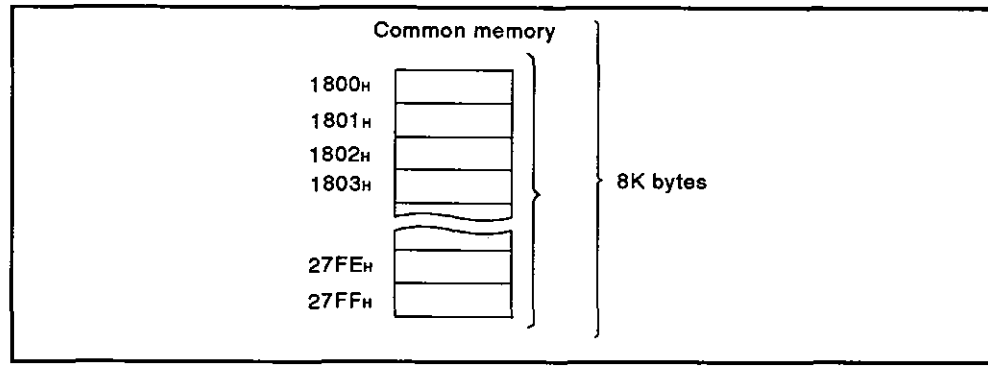
(3) Common memory structure

A common memory address consists of 1 byte (8 bits).

All areas can be used.

(4) Common memory addresses

A common memory address begins with the address (1800H) next to the buffer.



REMARK

The AD51H-BASIC Programming Manual gives details about GETMEM/PUTMEM commands.

6.7 Special Relays (EM9000 to EM9128)

Special relays are relays which can be set for specified purposes in the AD51H.

Since the OS of the AD51H executes ON/OFF control, the ON/OFF state can be read and utilized by using special variable B@ with a BASIC program.

Table 6.8 Special Relays

Special Relay Number	Name	Description	Details	Related Special Register
EM9000	Error registration signal	ON : Error registration OFF: No error registration	This goes ON when a warning error, error message, or system-down error message is displayed on the LED.	D9001 to D9008
EM9020	Operation signal	ON : Multitasking OFF: Except multitasking	• Turned ON when the AD51H is multitasking. • Turned OFF when the AD51H is not multitasking.	—
EM9021	Memory card access permission signal	ON : Access is enabled. OFF: Access is disabled.	• This goes ON when the memory card access switch is turned ON. Accesses the memory card when EM9021 is ON. • This goes OFF when the memory card access switch is turned OFF.	—

6.8 Special Registers (ED9000 to ED9128)

Special registers are registers which can be set for specific purposes in the AD51H.

Since the OS of the AD51H stores data, such data can be read and utilized by using special variable W@ with the BASIC program.

Table 6.9 Special Registers

<table>
<tr><th>Special Register Number</th><th>Name</th><th>Description</th><th>Details</th><th>Related Special Relay</th><th>Related Special Register</th></tr>
<tr><td>ED9000</td><td>Error number</td><td>0 to 8</td><td>The error number registered by the AD51H is stored.</td><td>EM9000</td><td>ED9001 to ED9008</td></tr>
<tr><td>ED9001 to ED9008</td><td>Error code</td><td>0 : Normal
Except 0 :
Error code</td><td>• Error codes displayed by the LED are stored sequentially from ED9001 to ED9008.
• If an error with more than seven factors occurs, the error code is not stored.
• An error code displayed by the LED can be cleared by the indicator reset switch.</td><td>EM9000</td><td>ED9000</td></tr>
<tr><td>ED9009 to ED9016</td><td>Error line number</td><td>0 : No line number
1 to FFFF:
Line number</td><td>• The line number of an error detected by the BASIC is stored.
• Error codes and the line number storage special registers correspond as shown below.
<table><tr><th>For Storing an Error Code</th><th>For Storing a Line Number</th></tr><tr><td>ED9001</td><td>ED9009</td></tr><tr><td>ED9002</td><td>ED9010</td></tr><tr><td>ED9003</td><td>ED9011</td></tr><tr><td>ED9004</td><td>ED9012</td></tr><tr><td>ED9005</td><td>ED9013</td></tr><tr><td>ED9006</td><td>ED9014</td></tr><tr><td>ED9007</td><td>ED9015</td></tr><tr><td>ED9008</td><td>ED9016</td></tr></table>{ When an error is detected by the BASIC, the error code and the line number are alternately displayed by the LED. }
• A line number displayed by the LED can be cleared by the indicator reset switch.</td><td>EM9000</td><td>ED9001 to ED9008</td></tr>
<tr><td>ED9020</td><td>Execution modes</td><td>0 to 4</td><td>• The setting number of the mode setting switch is stored.
0, 1 : Execution mode
2, 3 : Multitask debug mode
4 : Programming mode</td><td>—</td><td>—</td></tr>
</table>

Special Register Number	Name	Description	Details	Related Special Relay	Related Special Register
ED9021	Memory card state	0 : Memory card access switch ON 1 : Memory cards cannot be inserted or removed. 2 : Memory cards can be inserted or removed.	• ON/OFF states of the memory card access switch and memory card insertion/removal are stored. • 0 : When the memory card access switch is ON and a memory card can be accessed • 1 : The memory card access switch is turned OFF, or the memory card cannot be inserted/removed because the memory card is being accessed or opened. • 2 : The memory card can be inserted/removed after "1" is stored.	EM9021	—
ED9022	BASIC task states	0 : The BASIC task is in the STOP state. 1 : The BASIC task is in the execution, STOP, or wait state.	• The states of the BASIC tasks 1 to 8 are stored as follows: b15 to b8 b7 to b0 0 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 BASIC task 8 | BASIC task 7 | BASIC task 6 | BASIC task 5 | BASIC task 4 | BASIC task 3 | BASIC task 2 | BASIC task 1 • 0 : The BASIC task is in the STOP state. (BASIC program is not being executed.) • 1 : The BASIC task is in the execution state. (BASIC program is being executed.) • BASIC task is in the STOP state. (All BASIC tasks are not being executed.) • BASIC task is in the wait state. (The BASIC task is waiting for a command (PRINT command, etc.) to be completed, or the execution interruption command was executed.) • When the bit of a designated task is "0", start the BASIC task using the ZSTART command.	—	—

6.9 Communications Buffer

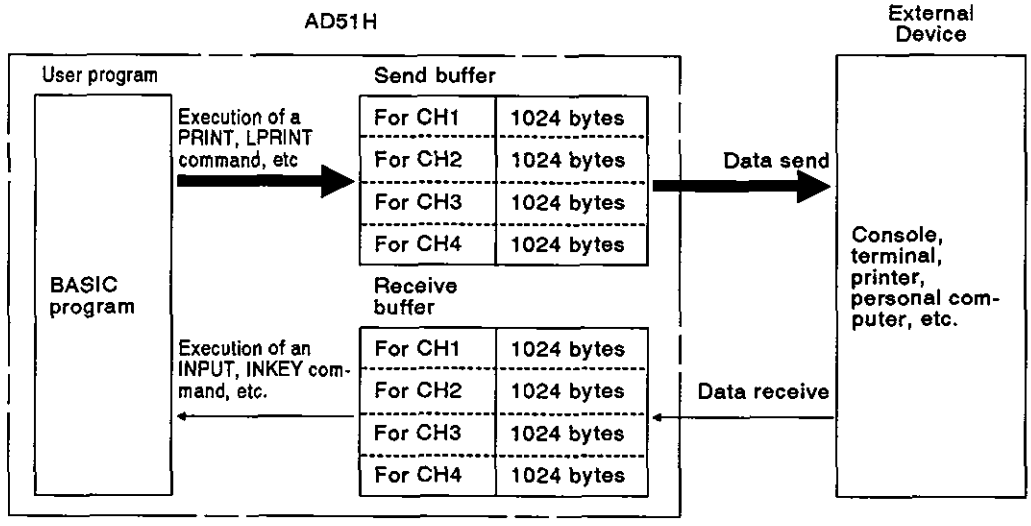
The communications buffer is used to receive and transmit data from/to an external device connected to the RS-232C interface, the RS-422 interface, or the parallel interface of the AD51H.

The OS of the AD51H controls the communications buffer when these interfaces are used.

Each interface is assigned 1024 bytes of send buffer and 1024 bytes of receive buffer.

When a command that transmits data to an external device (such as a PRINT or LPRINT command) is executed using the BASIC program of the AD51H, the send buffer stores that data.

The receive buffer stores data transmitted from the external device to the AD51H. Then, buffer data is stored in a designated variables area by an INPUT or INKEY command.



POINT

The receive buffer area can be changed by the ZCNTL command of the AD51H-BASIC.

6.10 I/O for a PC CPU

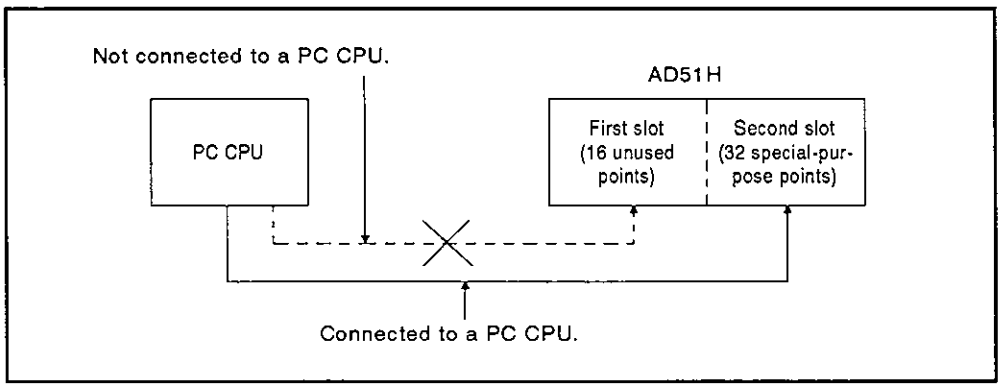
The AD51H I/O is used to (a) communicate ON/OFF data between a PC CPU and the AD51H, and (b) start a designated task from the PC.

Use device X to send an input signal from the AD51H to a PC CPU.

Use device Y to send an output signal from a PC CPU to the AD51H.

The AD51H uses two slots - the first slot has 16 unused points and the second slot has 32 special-purpose points.

Communications between the AD51H and a PC CPU can be executed by using these 32 points.



6.10.1 I/O (X, Y) list

Fig. 6.11 shows how to determine a device when an AD51H is installed in slot number 1 or 0 of a main base unit.

- The I/O number used for a sequence program differs depending on the slot in which the AD51H is installed.
- The value designated by special variable B@ of the BASIC program of the AD51H is specified to &H00 to &H1F.

Table 6.10 shows device numbers and signal names.

Sections 6.10.2 to 6.10.3 give details about these signal names.

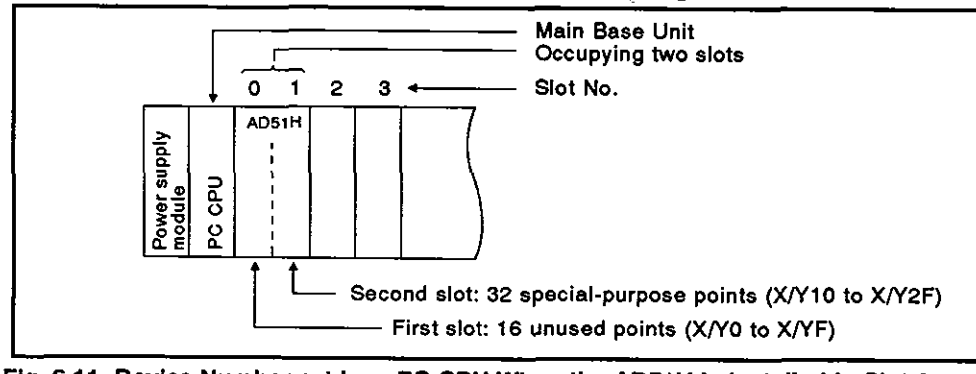


Fig. 6.11 Device Number set by a PC CPU When the AD51H is Installed in Slot 0 or 1

Table 6.10 I/O List of the AD51H

Signal Direction : AD51H → PC CPU			Signal Direction : PC CPU → AD51H			
Input Number Used by a PC CPU	Value Designated by Special Variable B@	Signal Name	Input Number Used by a PC CPU	Value Designated by Special Variable B@	Signal Name	
X00 to X0F	—	Unusable	Y00 to Y0F	—	Unused (Available as the substitution for internal relay (M).)	
X10 X1A	&H00 &H0A	General-purpose input	Y10 to Y20	&H00 to &H10	General-purpose output	
X1B	&H0B	Start of multitasking				
X1C	&H0C	Stop of multitasking				
X1D	&H0D	AD51H system down				
X1E X1F	&H0E &H0F	Unusable				
X20 to X2F	&H10 to &H1F	General-purpose input	Y21	&H11	Designation of a general-purpose output/start BASIC task No. *	BASIC task 1
			Y22	&H12		BASIC task 2
			Y23	&H13		BASIC task 3
			Y24	&H14		BASIC task 4
			Y25	&H15		BASIC task 5
			Y26	&H16		BASIC task 6
			Y27	&H17		BASIC task 7
			Y28	&H18		BASIC task 8
			Y29	&H19	Program start signal (unusable for a general-purpose output)	
			Y2A to Y2F	&H1A to &H1F	Unusable	

REMARK

* Designate the start task number if a task is set to "start by the interrupt from PC CPU" by multitask setting.
Designate a general-purpose output if the task is not set to "start by the interrupt from PC CPU".

6.10.2 Detailed explanation of input (X)

This section explains input (X) of the AD51H.

The input numbers (Xn) used in this section are the numbers set by a PC CPU.

(1) General-purpose inputs (X10 to X1A, X20 to X2F: 27 points)

The user can use general-purpose inputs for transmitting ON/OFF data from the AD51H to a PC CPU.

Use special variable B@ to turn general-purpose inputs ON or OFF.

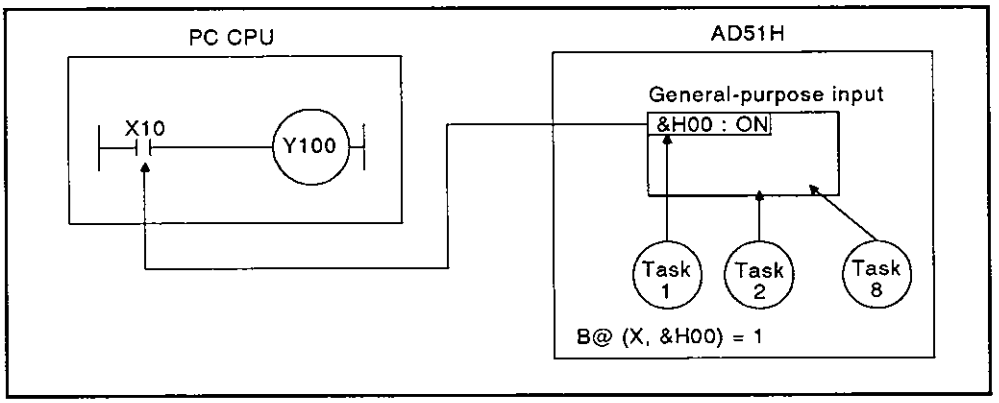


Fig. 6.12 Communications with a PC CPU Using a General-Purpose Input

(2) Multitasking start flag (X1B)

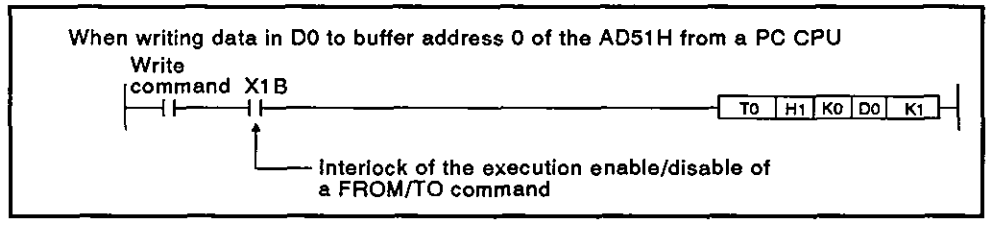
(a) When mode setting switch 1 *1 of the AD51H is set to "execution mode (0 or 1)", this flag becomes valid (i.e., ON/OFF control is executed.).

(When the AD51H is set to other modes, this is in the OFF state.)

(b) It goes ON if (a) the AD51H is in the execution mode, or (b) the AD51H goes into the multitask execution enable state after the power is turned ON or the AD51H is reset.

Use this flag as the execution enable/disable interlock signal of the FROM/TO instruction for data communications with the AD51H buffer (see Section 6.6.5) using a sequence program.

When a FROM/TO instruction is executed from a PC CPU, if the AD51H is not in the multitask execution enable state, an "SP UNIT DOWN" error occurs, and the PC CPU stops operating. *2



(c) This flag goes OFF in either of the following cases when the AD51H is in the execution mode.

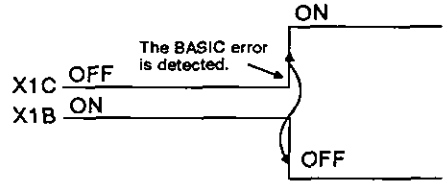
1) RUN keyswitch of the AD51H is in the STOP position.

2) RUN keyswitch of the AD51H was reset.

(When the AD51H completes reset processing and returns to the multitask execution enable state, this signal goes ON.)

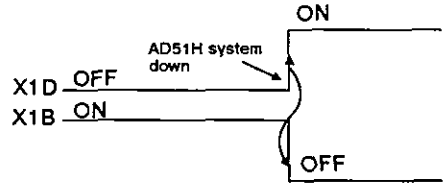
3) A BASIC error is detected.

(Multitask execution stop flag : X1C turns ON.)



4) AD51H system down

(AD51H down flag : X1D goes ON.)



REMARK

(1) *1 : Sections 4.1 and 4.2.1 give details about mode setting switch 1.

(2) *2 : When the multitask execution start flag (X1B) is turned OFF by resetting the AD51H, if a FROM/TO command is executed within the access enable time of a set FROM/TO command, an error does not occur.

(3) Multitask execution stop flag (X1C)

(a) When mode setting switch 1 of the AD51H is set to "execution mode (0 or 1)", this flag becomes valid. (ON/OFF control is executed.)

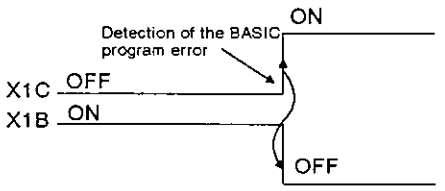
(If the AD51H is set to another mode, this flag goes OFF.)

(b) If a BASIC program error is detected during multitasking, this signal goes ON.

(At this time, error step and error code • numbers are displayed on an LED, and the mode is switched to the system mode.)

(Correct the BASIC program error.)

(c) If the multitask execution stop flag (X1C) goes ON, the multitask execution start flag (X1B) goes OFF.



(4) AD51H down flag (X1D)

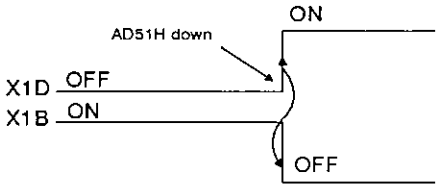
(a) When the execution mode setting switch 1 of the AD51H is set to "execution mode (0 or 1)", this flag becomes valid. (ON/OFF control is executed.)

(b) If the error that caused the system-down message during multitasking is detected, this goes ON.

At this time, an error code signal is displayed on the LED. *

If the AD51H is reset by setting the RUN keyswitch to position of RESET, the AD51H system-down error signal is cleared.

(c) When the AD51H system-down flag goes ON, the multitask execution start flag (X1B) goes OFF.



REMARK

* Section 9 gives details about displayed error codes.

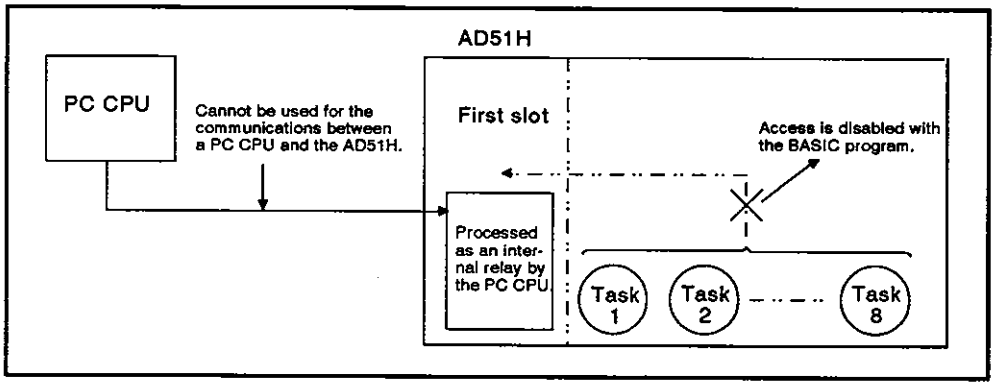
6.10.3 Detailed explanation of output (Y)

This section explains output (Y) of the AD51H.

Output numbers (Yn) used in this Section are the numbers set by a PC CPU.

(1) Unused area (16 points)

These 16 points are in the first slot of the AD51H. These outputs cannot be accessed by a BASIC program.



When output (Y) is turned ON or OFF by a PC CPU, this area can be used as a substitute for internal relay (M).

(2) General-purpose outputs (Y10 to Y20 : 17 points)

A general-purpose output can be used for transmitting ON/OFF data from a PC CPU to the AD51H.

A PC CPU can turn ON/OFF general-purpose outputs using a sequence program or a test function of a peripheral device.

The AD51H uses special variable B@ to read general-purpose output data turned ON/OFF by a PC CPU.

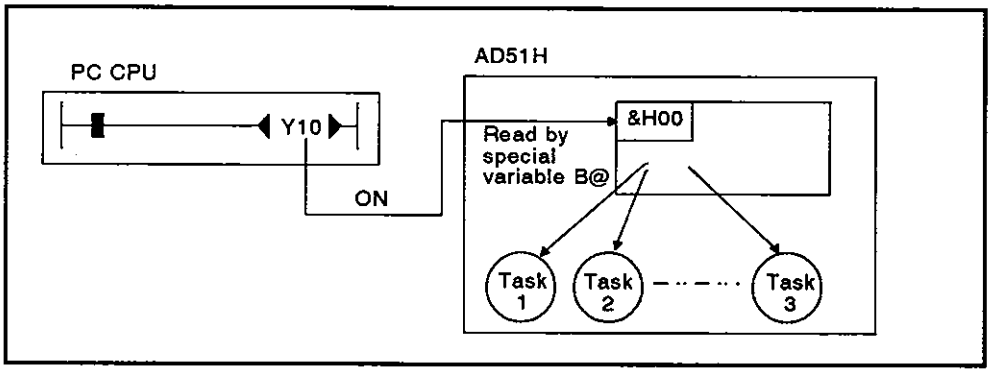


Fig. 6.13 Reading of General-Purpose Output Data Turned ON/OFF by a PC CPU

(3) Start task number designation signals (Y21 to 28: 8 points)

(a) These signals designate tasks to be started by an interrupt from a PC CPU.

Each signal is allocated to each task from task numbers 1 to 8.

(b) Only a task set to "IT (start by the interrupt from a PC CPU)" by multitask setting can be a start task number designation signal.

Device number of the task that is not set by multitask setting becomes the same with the allocation of a general-purpose output.

Device No.	Multitask Setting		
	Start by an Interrupt from a PC CPU		Others
Y21	Setting of the task No. to be started	Task No. 1	General-purpose output (tasks cannot be started.)
Y22		Task No. 2	
Y23		Task No. 3	
Y24		Task No. 4	
Y25		Task No. 5	
Y26		Task No. 6	
Y27		Task No. 7	
Y28		Task No. 8	

(c) Turn ON the device number (Y21 to Y28) which corresponds to the task which is started.

When task start signal (Y29) goes ON, start the designated task (task that was turned ON by Y21 to Y28).

However, if the task that is being executed is designated, (Y29) is ignored.

(d) Turn OFF the start task number designation signal that was turned ON after confirming that the designated task is executed.

Task starting can be confirmed by turning ON a general-purpose input (X) allocated to a reset (OFF) command by a BASIC program (as shown in the example on the following page).

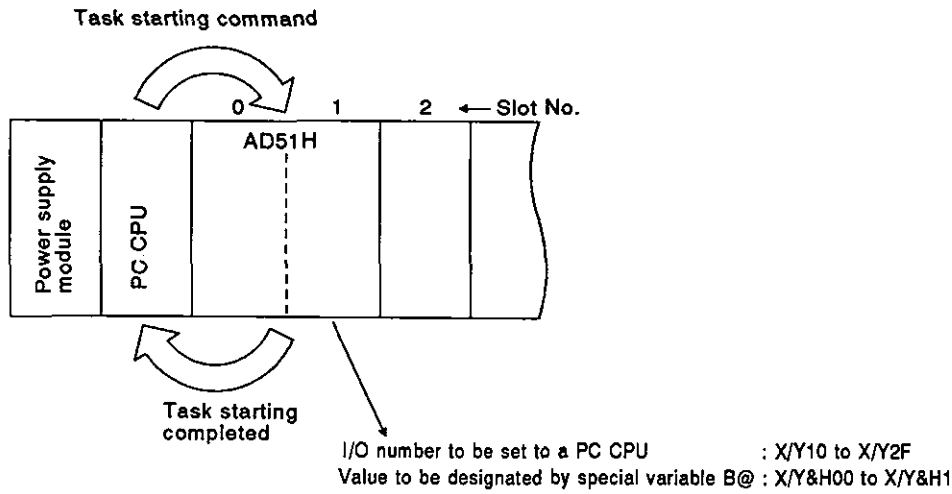
Example:

This example shows the procedure for (a) starting a task, and (b) turning OFF the start task number designation signal/task start signal.

[Conditions]

1) Start task number: 1

2) Start task No. designation signal/task start signal reset command: X21



[Procedure]

1) Task setting of number 1:

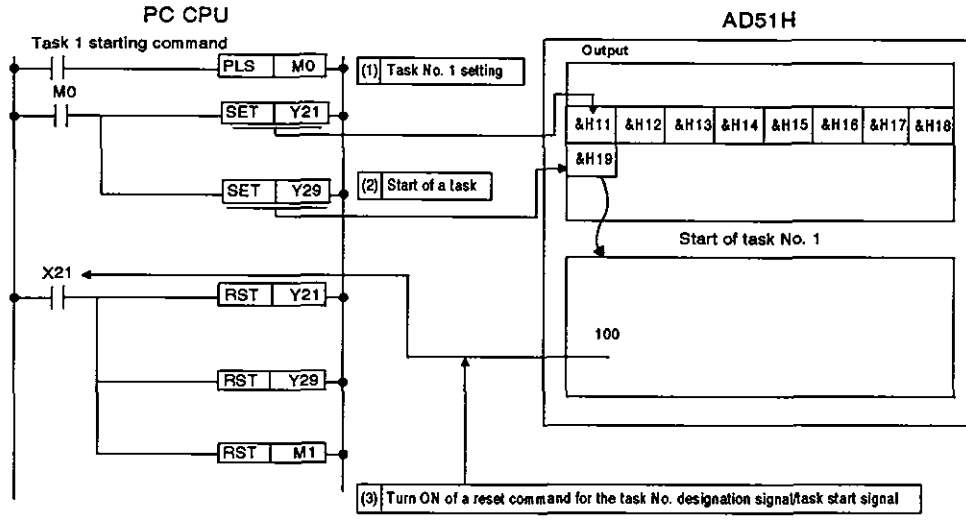
   (Y21) is turned ON by a sequence program.

2) Starting the task:

   (Y29) is turned ON by a sequence program.

3) The start task number designation signal/task start signal go OFF:

   X&H11 are turned ON by the BASIC program.



(4) Task start signal (Y29)

(a) This signal starts the task designated by a start task number designation signal (Y21 to Y28).

Even if no task has been started by an interrupt from a PC CPU, this task start signal cannot be used as a general-purpose output.

(b) The task start signal is valid only when the multitask execution enable signal (X1B) is turned ON.

(c) When a task start signal goes ON, the (X1B) signal starts a designated task. *

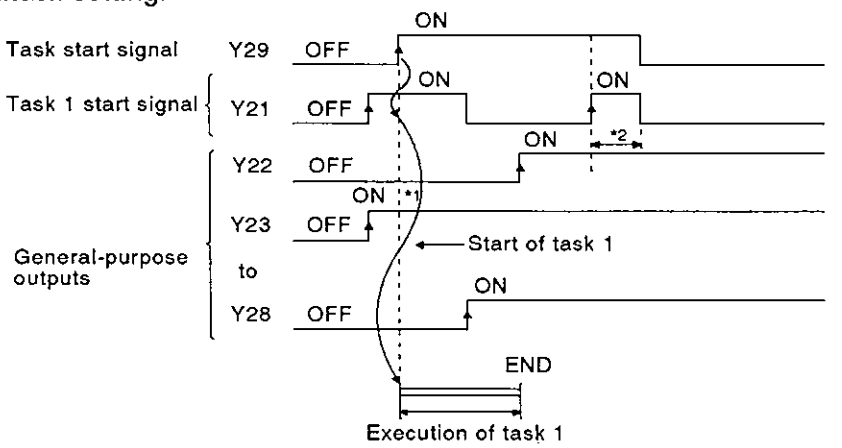
(When the task start signal is ON, if the task number designation signal is turned ON, the (X1B) signal is ignored.)

Example:

This example indicates a program to start task 1 from a PC CPU.

[Condition]

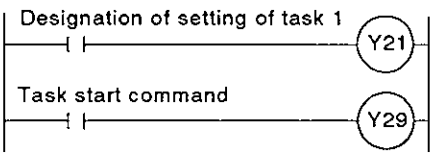
Set only task 1 to "IT (start by the interrupt from a PC CPU)" by multitask setting.



*1 : The task is not started because this is processed as a general-purpose output.

*2 : Task 1 is not started because a task start signal does not go ON.

[Sequence program]



(d) When a task designated by a start task number designation signal (Y21 to Y28) is being started, the start for any other task is ignored.

(The task currently being executed is continued.)

REMARK

* This is the task that corresponds to device number being turned ON (Y21 to Y28).

However, any device number that is not set to "IT (start by the interrupt from a PC CPU)" by multitask setting is ignored.

7. AD51H FUNCTIONS

This section explains the functions of the AD51H.

7.1 AD51H Functions

Table 7.1 gives the functions of the AD51H.

Table 7.1 AD51H Functions

Item		Description	Reference Section
BASIC Program Functions	Sub CPU function	1) A BASIC program can be used to calculate complex numerical values and functions that lengthen the scan time of a PC CPU. • Collection, analysis and compensation of measurement data • Functional operations such as sin, log, and square root.	The AD51H-BASIC Programming Manual gives details about commands and functions used by the AD51H.
	Monitoring display function	Displays the operating states such as production status, operating status, and trouble information on a console or terminal connected to the AD51H.	7.2
	Key operation function	Inputs the production schedule, the production quantity, operations, and set data from the keyboard of a console or terminal connected to the AD51H.	7.2
	Printer function	Prints the production plans, achievements, daily reports, trouble information, plan data, inspection results, and test results by using a printer connected to the AD51H.	7.3
	Data input function	Inputs data from a bar code reader or magnetic card reader connected to the AD51H. (By using a BASIC program, the AD51H can transmit and receive data in a free data format independently of the protocol of the communicating device.) • Inputs a production lot number, product name, quantity, etc. • Collects measurement values and test data	—
	External device connection function	Transmits and receives data from/to a BASIC program by connecting a computer to the RS-232C or RS-422 interface of the AD51H.	7.4
	File management function	Reads/writes data from/to a memory card installed in a memory card interface of the AD51H. • Management of set data and positioning data • Collection and storage of production data • Collection and storage of inspection data	—
	Clock function	Reads and writes clock data (year, month, day, hour, minute, second, and day of the week) from/to a PC CPU which has a clock function.	7.6
Offline programming function		Creates, corrects, and stores a BASIC program to a user's FD and prints it out independently by a PC/AT.	
Online programming function		Creates, executes, and corrects a BASIC program by using system commands when a console (PC/AT, VT-386, VG-620, or VT-220) is connected to the AD51H. Stores/reads an execution program to/from a memory card using a system command.	Online programming/ debugging
Multitask debugging function		Executes and debugs a BASIC program during multitasking by connecting a console and a debugger to the AD51H.	

7.2 Communications with a Console or Terminal

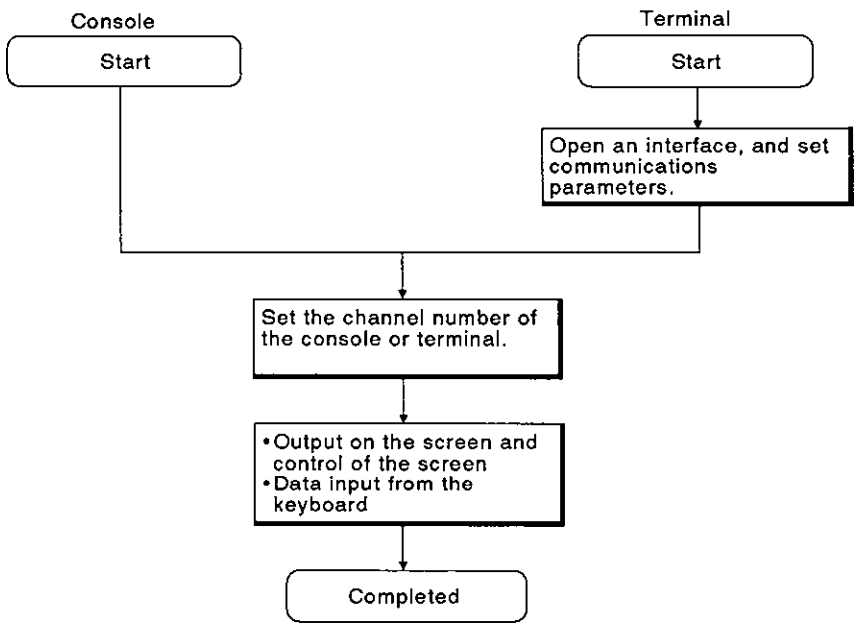
Data can be displayed on the screen of a PC/AT connected to the RS-422 interface (CH.3) of the AD51H or a VG-620, VT-382, or VT-220 connected to the RS-232C interface (CH.1 or 2) of the AD51H. Data can also be input from the keyboard.

A VG-620, VT-382, or VT-220 connected to CH.1 and a PC/AT connected to CH.3 can be used as a console and a terminal.

The device (PC/AT, VG-620, VT-382, or VT-220) connected to a channel which is set for the console by mode setting switch 2 of the AD51H can be used as a console.

A device (PC/AT, VG-620, VT-382, or VT-220) connected to a channel which is set for other than the console by the AD51H can be used as a terminal.

The procedures for (a) displaying a screen, and (b) inputting from the keyboard of a console or terminal are as follows:



REMARK

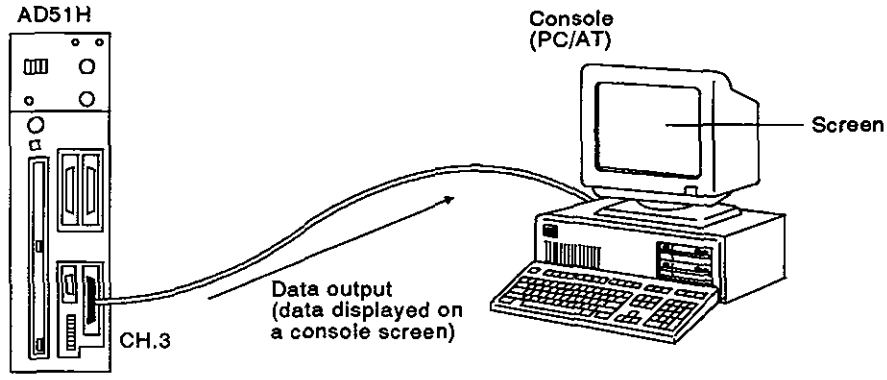
1) Section 1.5.2 gives details about consoles.

2) Section 4.3.2 explains how to set a console using mode setting switch 1.

7.2.1 Communications with a console

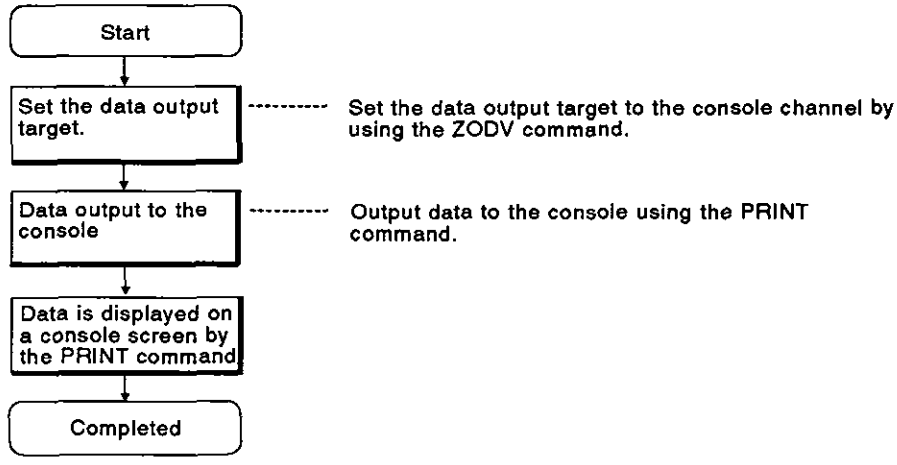
(1) Displaying data on a console screen

This section explains how data is displayed on a console screen.



[Procedure for displaying data on a console screen]

Use the following procedure to display data on a console screen:



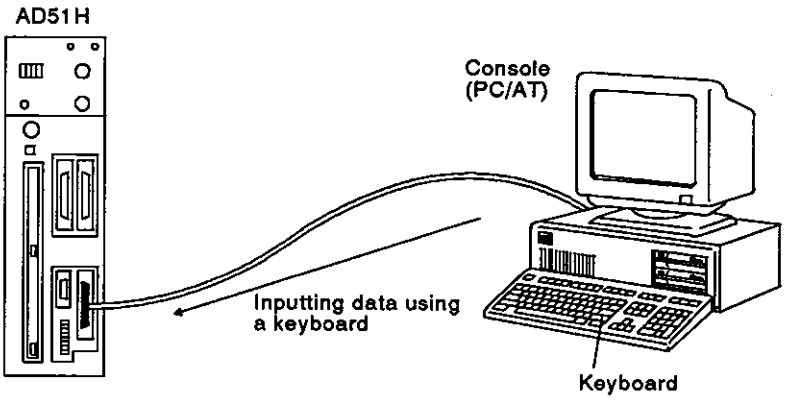
[Commands and functions]

The following list gives the commands and functions used for displaying data on a console screen.

Commands and Functions	Processing
ZODV	Switches a data output target.
PRINT	Displays data on a console screen.
PRINT USING	Displays data on a console screen. (Format can be designated.)
LOCATE	Designates the position where data is displayed on a console screen.
CLS	Clears a console screen.

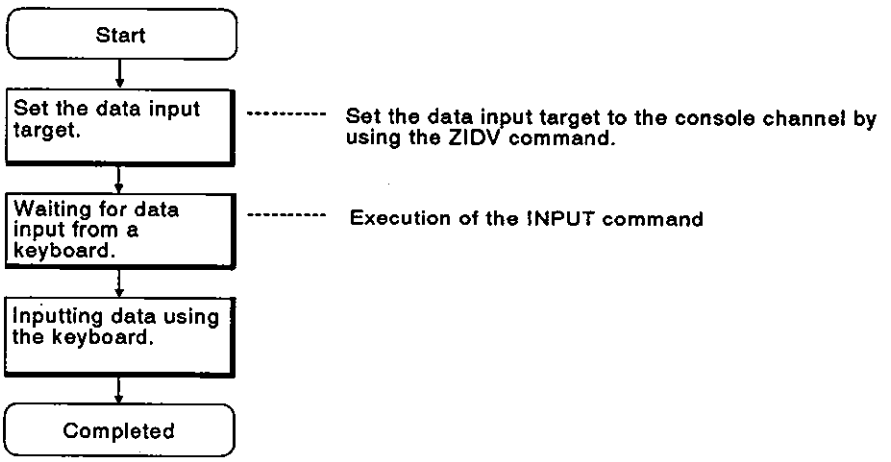
(2) Inputting data using a console keyboard

This section explains how to input data using a console keyboard.



[Procedure for inputting data using a console keyboard]

Use the following procedure to input data using a console keyboard:



[Commands and functions]

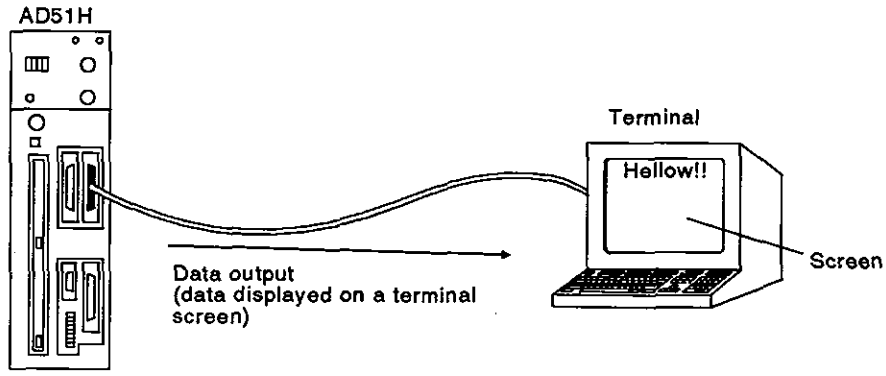
The following list gives the commands and functions used for inputting data using a console keyboard.

Commands and Functions	Processing
ZIDV	Switches a data input target.
INPUT	• Enables data input with the keyboard. • Cannot use the control codes 0H to 1FH, comma (2CH), and [DEL] (7FH) for character codes as data.
LINE INPUT	• Enables data input with the keyboard. • Cannot use the control codes 0H to 1FH and [DEL] (7FH) for character codes as data.
INKEY$ INPUT$	• Enables data inputting with a key. • Cannot use the control codes 0H, 03H, 13H, 80H, and FDH to FFH for character codes as data.

7.2.2 Communications with a terminal

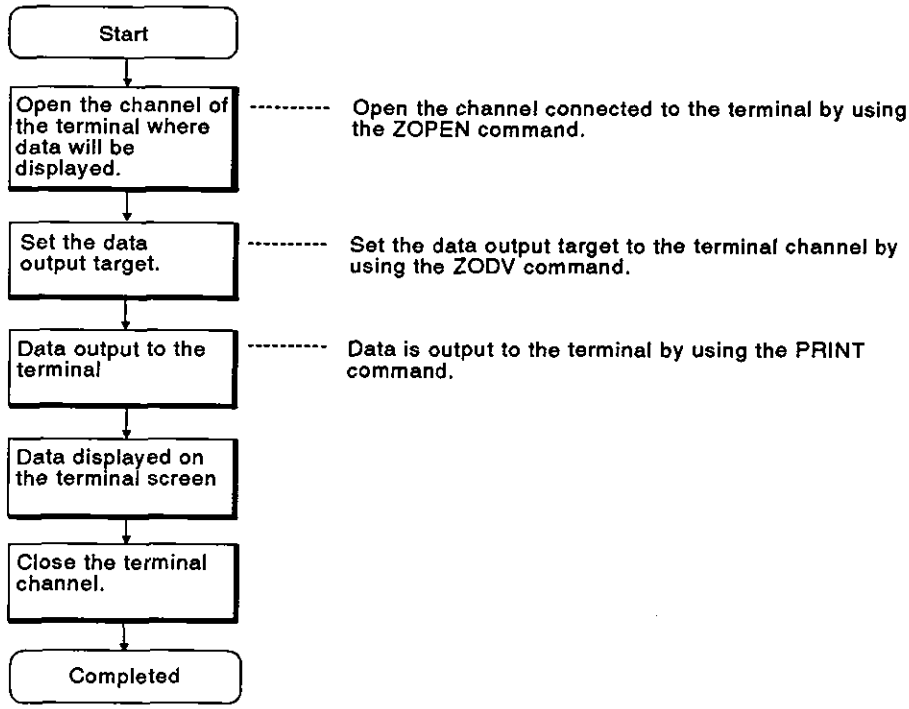
(1) Displaying data on a terminal screen

This section explains how data is displayed on a terminal screen.



[Procedure for displaying data on a terminal screen]

Use the following procedure to display data on a terminal screen:



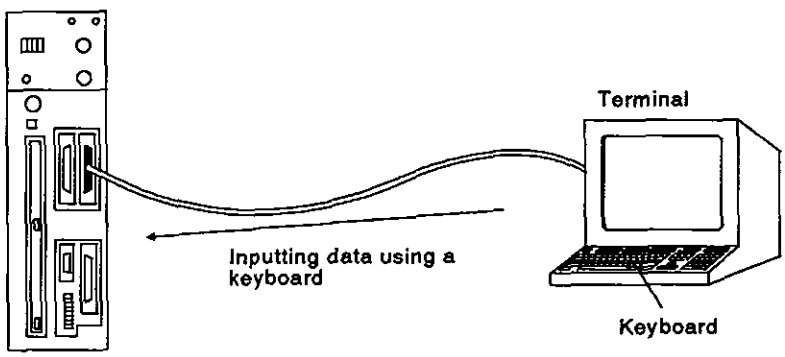
[Commands and functions]

The following list gives the commands and functions used for displaying data on a terminal screen.

Commands and Functions	Processing
ZOPEN	Opens an interface and prepares to communicate with an external device.
ZCNTL	Sets communications parameters for an opened interface and reads the interface status.
ZODV	Switches a data output target.
PRINT	Displays data on a terminal screen.
PRINT USING	Displays data on a terminal screen. (Format can be designated.)
LOCATE	Designates the position where data is displayed on a terminal screen.
CLS	Clears a terminal screen.

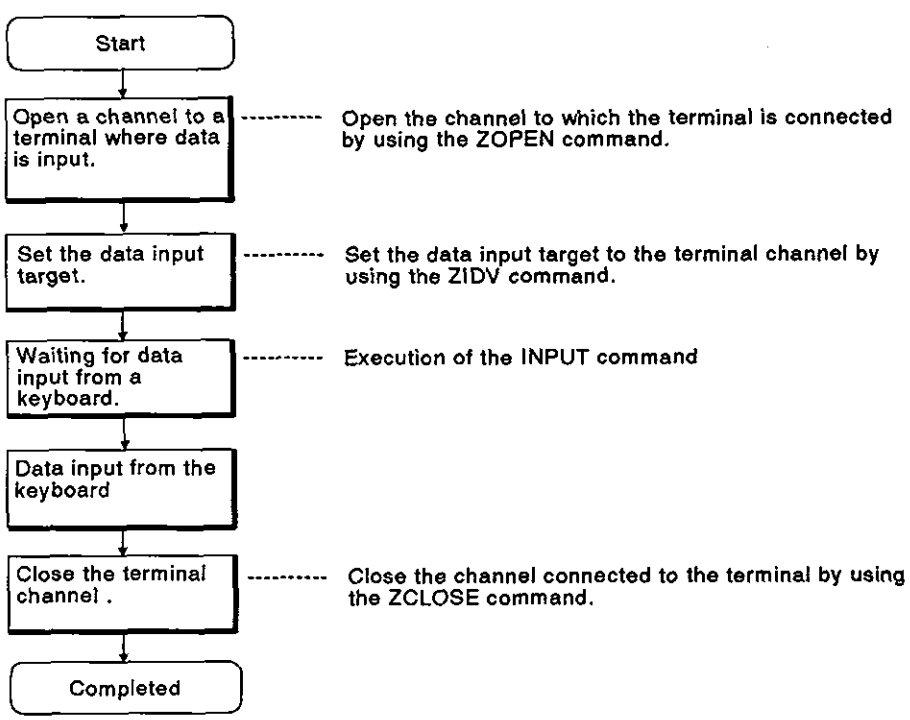
(2) Inputting data using a terminal keyboard

This section explains how to input data using a terminal keyboard.



[Procedure for inputting data using a terminal keyboard]

Use the following procedure to input data using a terminal keyboard:



[Commands and functions]

The following list gives the commands and functions used for inputting data using a terminal keyboard.

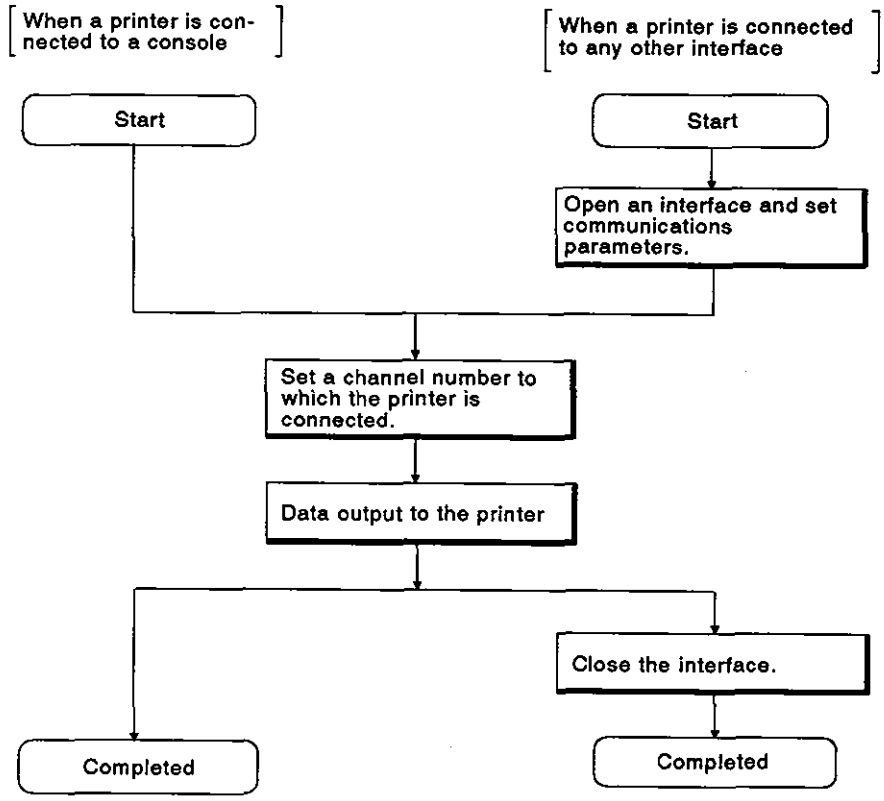
Commands and Functions	Processing
ZOPEN	Opens an interface and prepares to communicate with an external device.
ZCNTL	Sets communications parameters for an opened interface and reads the interface state.
ZIDV	Switches a data output target.
INPUT	• Enables data inputting with a key • Cannot use the control codes 0H to 1FH, comma (2CH), and [DEL] (7FH) for character codes as data.
LINE INPUT	• Enables data inputting with a key. • Cannot use the control codes 0H to 1FH and [DEL] (7FH) for character codes as data.
INKEY$ INPUT$	• Enables data inputting with a key. • Cannot use the control codes 0H, 03H, 13H, 80H, and FDH to FFH) for character codes as data.

7.3 Printing

Data can be printed by using a printer connected to the RS-232C interface, RS-422 interface, or parallel interface of the AD51H.

Data can also be printed by connecting a printer to a PC/AT used as a console.

The procedure using a printer connected to a console (PC/AT) is different from the procedure using a printer connected to any other interface.

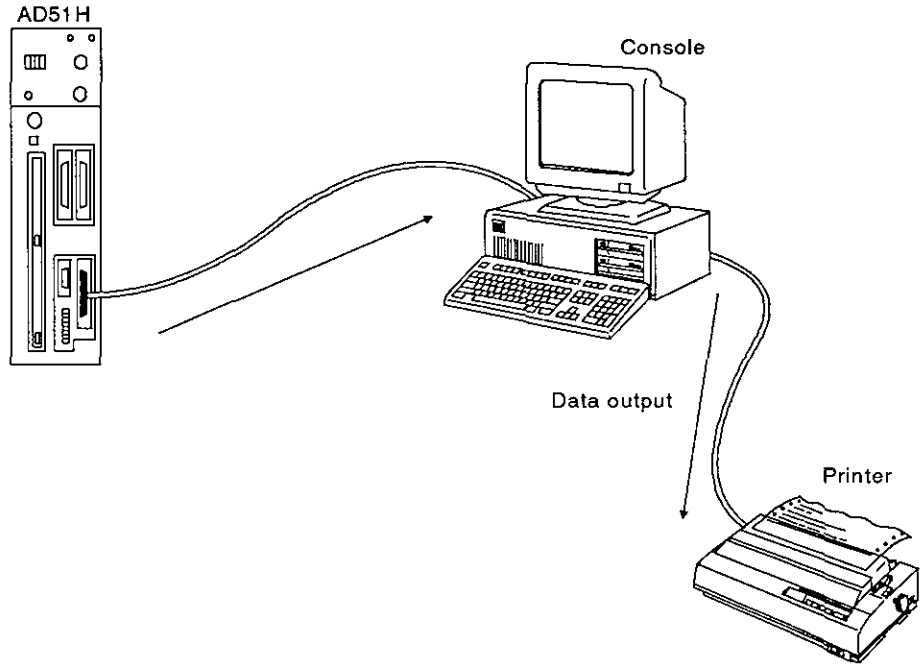


7.3.1 Printout using a printer connected to a console (PC/AT)

This section explains how to print by connecting a printer to a PC/AT connected to an interface set for the console.

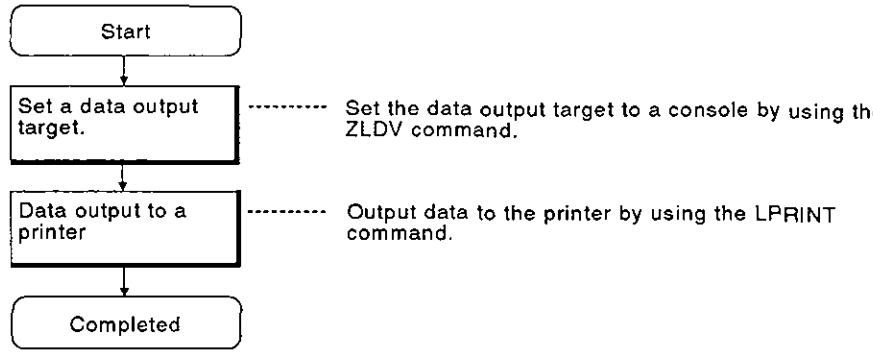
When using a PC/AT for printing data, the printer settings are necessary.

Section 1.4.5 gives details about setting printers.



[Printout procedure]

Use the following procedure to print data using a printer connected to a console:



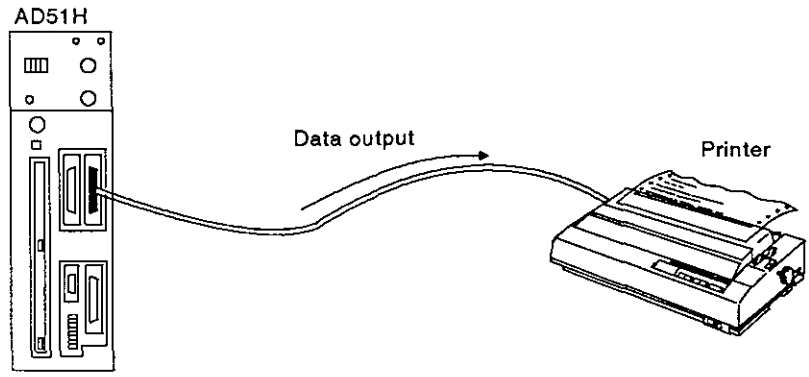
[Commands and functions]

The following list gives the commands and functions used for printing data by using a printer connected to a console.

Commands and Functions	Processing
ZLDV	Switches the output target to a printer connected to a console.
LPRINT	Outputs data to a designated printer.
LPRINT USING	Outputs data to a designated printer. (Format can be designated.)
LLIST	Outputs a program list to a designated printer.

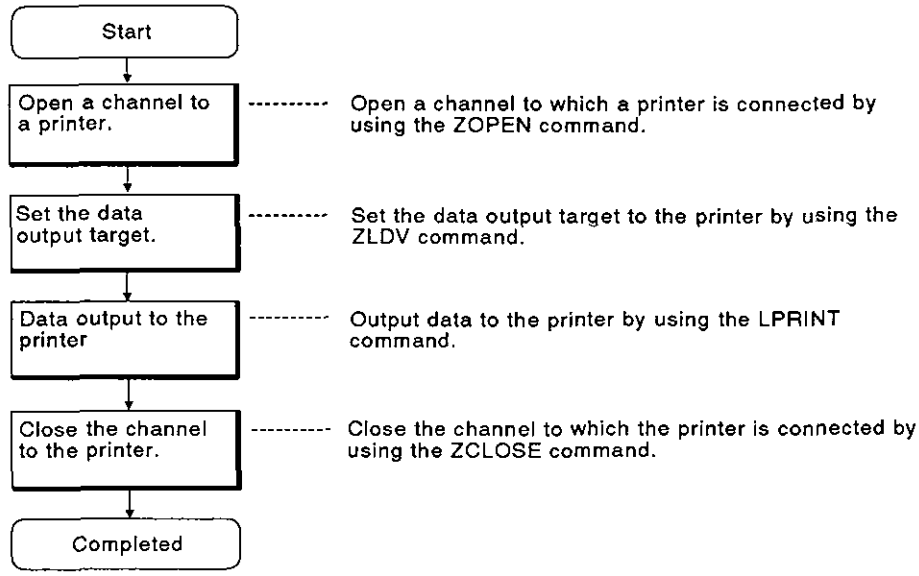
7.3.2 Printout using a printer connected to an interface which is not set for a console

This section explains how to print by connecting a printer to an interface which is not set for a console.



[Printout procedure]

Use the following procedure to print data by using a printer connected to an interface which is not set for a console:



[Commands and functions]

The following list gives the commands and functions used for printing data by using a printer connected to an interface which is not set for a console.

Commands and Functions	Processing
ZOPEN	Opens an interface and prepares for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets communications parameters for an opened interface and reads the interface state.
ZLDV	Switches a channel number to the channel number of the interface to which a printer is connected.
LPRINT	Outputs data to a designated printer.
LPRINT USING	Outputs data to a designated printer. (Format can be designated.)
LLIST	Outputs a program list to a designated printer.

7.4 Communications With a Device Other Than a Console, Terminal, or Printer

Communications is enabled by connecting an external device to the RS-232C interface or the RS-422 interface of the AD51H.

7.4.1 Sending data

This section explains how data is sent from the AD51H to a device other than a console, terminal, or printer.

Data send processes:

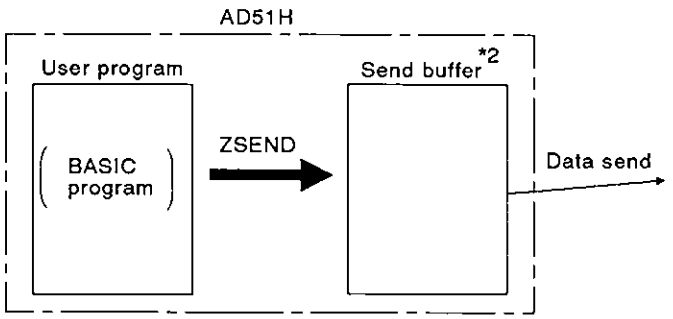
1) Opens an interface.

2) Prepares data to be sent.

3) Sets the number of bytes and the time-out duration of the sent data.

4) Transmits data by using the ZSEND command.

5) Closes the interface.



[Commands and functions]

The following list gives the commands and functions used for sending data from the AD51H to a device other than a console, terminal, or printer. *3

Commands and Functions	Processing
ZOPEN	Opens an interface and prepares for communications with an external device.
ZCNTL	Sets communications parameters for an open interface and reads the interface status.
ZSEND	Sends data from a designated interface.
ZCLOSE	Closes an interface.

REMARK

(1)*1 : Other than a console, terminal, or printer
Section 7.2 gives details about communications using consoles and terminals.
Section 7.3 gives details about print operation.

(2)*2 : Section 6.9 gives details about the send buffer.

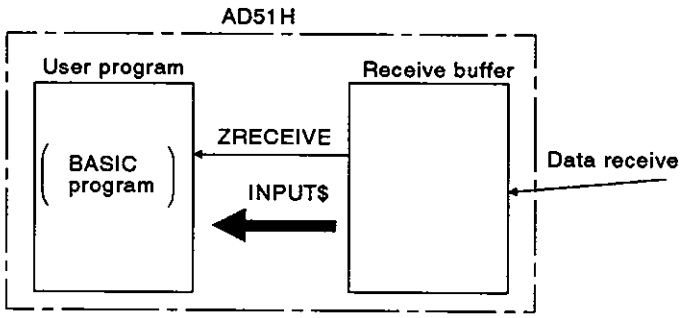
(3)*3 : The AD51H-BASIC Programming Manual gives details about commands and functions.

7.4.2 Receiving data

This section explains how the AD51H receives data sent from a device other than a console or terminal.

Data receive processes:

1) Opens an interface.
2) Receives data
3) Stores the received data in the variables area by using either the ZRECEIVE command or INPUT$ function.
4) Closes the interface.



[Commands and functions]

The following list gives the commands and functions used by the AD51H for receiving data sent from a device other than a console or terminal.

Commands and Functions	Processing
ZOPEN	Opens an interface and prepares for communications with an external device.
ZCNTL	Sets communications parameters for an opened interface and reads the interface state.
INPUT$	• Stores data received from an interface in a designated variables area. • Cannot use the control codes 0H, 03H, 13H, 80H, and FDH to FFH of character codes as data.
ZRECEIVE	• Stores data received from an interface in a designated variables area.

REMARK

The AD51-BASIC Programming Manual gives details about commands and functions.

7.5 Communications With a PC CPU and a Special-Function Module

The AD51H can read/write data from/to a PC CPU by using a BASIC program.

The following shows data that can be read/written from/to a PC CPU:

No.	Data Name	Write	Read
1	Device memory	O	O
2	Extension file register	O	O
3	PC CPU model name	X	O
4	Parameter data	O	O
5	Sequence program	O	O
6	Microcomputer program	O	O
7	Comment data	O	O
8	Extension comment data	O	O
9	Remote RUN/STOP of a PC CPU	O	X

O: Enabled X: Disabled

Data can also be read/written from/to the buffer of a special-function module by using a BASIC program.

This section explains the read/write of (a) data (1 and 2 above) from/to a PC CPU, and (b) data from/to the buffer of a special-function module.

POINT

When MELSECNET is used, data communications with a PC CPU in other stations on MELSECNET is enabled.

When used in a master station:

Data communications with a local station and a remote I/O station is enabled.

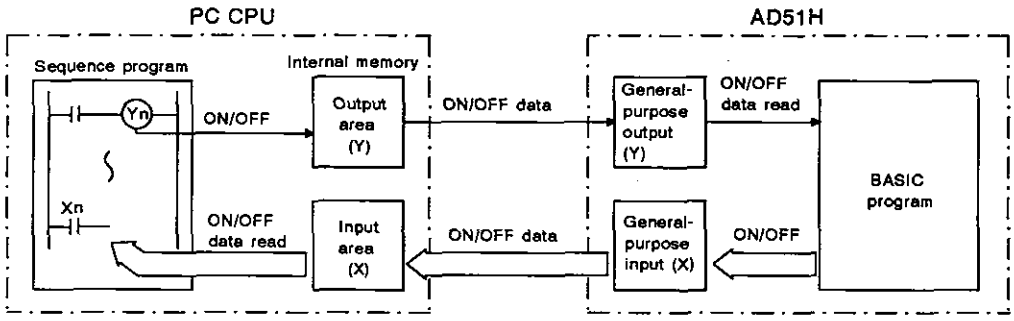
When used in a local station:

Data communications with a master station is enabled.

The AD51H-BASIC Programming Manual gives details.

7.5.1 Communicating ON/OFF data using the general-purpose inputs/outputs (X/Y) between the AD51H and a PC CPU

ON/OFF data is sent and received between the AD51H and a PC CPU by using the general-purpose inputs/outputs (X/Y) of the AD51H.



(1) Send from a PC CPU to the AD51H

ON/OFF data is sent from a PC CPU to the AD51H by using general-purpose output (Y).

A general-purpose output turned ON or OFF by a PC CPU is read by using special variable B@ with a BASIC program of the AD51H.

(2) Send from the AD51H to a PC CPU

Data is sent from the AD51H to a PC CPU by using general-purpose input (X).

A general-purpose input turned ON or OFF by a BASIC program of the AD51H is read by a sequence program in the PC CPU.

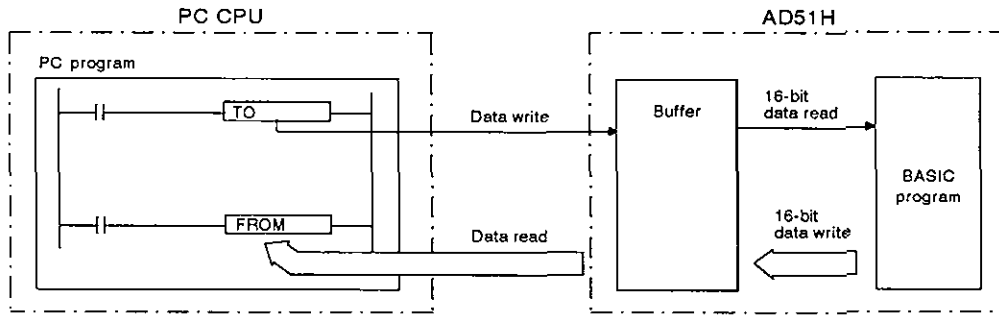
[Special variables]

The following list gives the special variables used for reading ON/OFF data of general-purpose input (X) and ON/OFF data of general-purpose output (Y).

Special Variables	Processing
B@ (X, expression)	Turns ON/OFF a general-purpose input (X) of the AD51H.
B@ (Y, expression)	Reads output data (Y) and general-purpose ON/OFF data of the AD51H.

7.5.2 16-bit data communications using the buffer

16-bit data is sent and received between the AD51H and a PC CPU by using the AD51H's buffer.



(1) Send from a PC CPU to the AD51H

16-bit data written to the AD51H buffer with a sequence program (TO command) of a PC CPU is read by using a BASIC program.

(2) Send from the AD51H to the PC CPU

16-bit data written to the AD51H buffer by using a BASIC program is read by using a sequence program (FROM command) of a PC CPU.

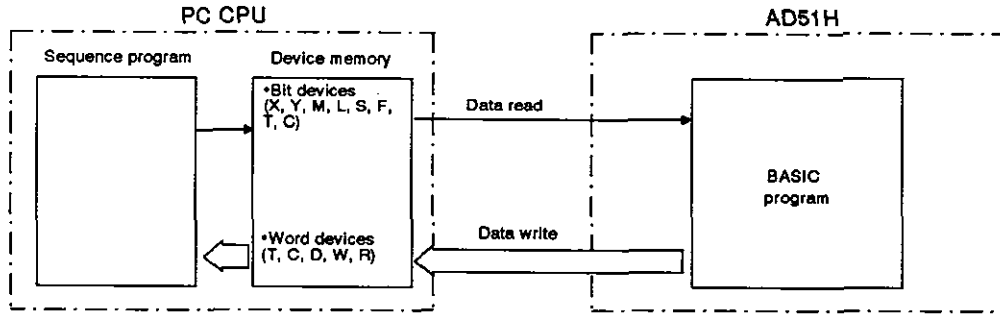
[Commands and functions]

The following list gives the commands and functions used for reading/writing 16-bit data from/to the AD51H buffer.

Commands and Functions	Processing
GETMEM	Reads the contents of the AD51H buffer.
PUTMEM	Writes 16-bit data to the AD51H buffer.

7.5.3 Reading/writing data from/to the device memory of a PC CPU

Data can be read/written from/to the device memory of a PC CPU by using a BASIC program of the AD51H.



The following devices can be read from and written to by using a BASIC program:

(a) Bit devices : X, Y, M, L, S, F, T, C

(b) Word devices : T(present value), C(present value), D, W, R

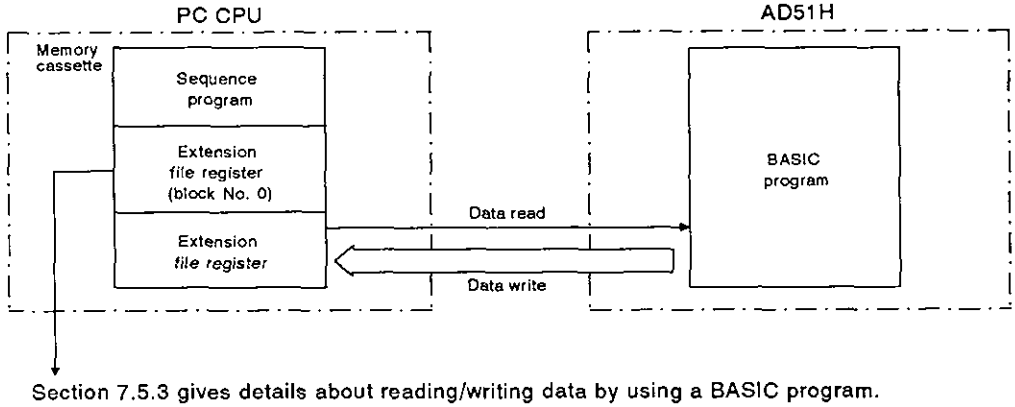
[Commands and functions]

The following list gives the commands and functions used for reading/writing data from/to the device memory of a PC CPU.

Commands and Functions	Processing
PCRD	Reads data from a PC CPU.
PCWT	Writes data to a PC CPU.

7.5.4 Reading/writing data from/to a PC CPU extension file register

Data can be read/written from/to the extension file register (extension file register of block No. 1 and later) of a PC CPU by using a BASIC program of the AD51H.



Section 7.5.3 gives details about reading/writing data by using a BASIC program.

[Commands and functions]

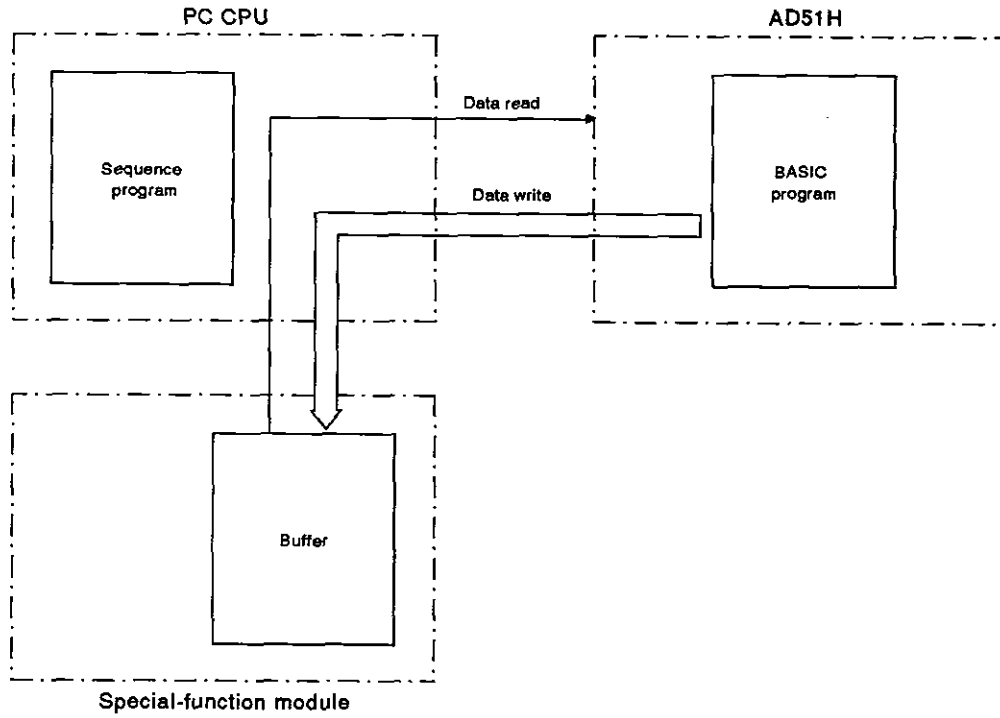
The following list gives the commands and functions used for reading/writing data from/to the extension file register of a PC CPU.

Commands and Functions	Processing
PCRD	Reads data from a PC CPU.
PCWT	Writes data to a PC CPU.

7.5.5 Reading/writing data from/to the buffer of a special-function module

Data can be read/written from/to the buffer of a special-function module by using a BASIC program of the AD51H.

Appendix 2 gives details about the special-function module's buffer addresses.



[Commands and functions]

The following list gives the commands and functions used for reading/writing data from/to the buffer of another special-function module.

Commands and Functions	Processing
PCRD	Reads data from a PC CPU.
PCWT	Writes data to a PC CPU.

REMARK

The AD51H-BASIC Programming Manual gives details about commands and functions.

7.6 Using Clock Data

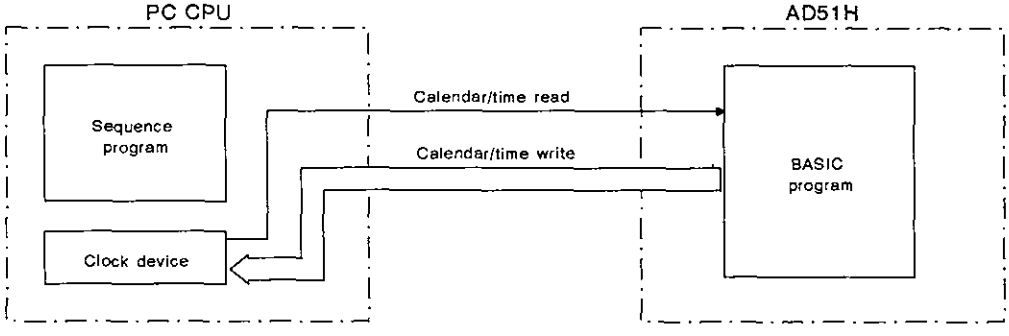
The AD51H does not have a clock function.

Therefore, when using the clock function on the AD51H, the calendar data (year, month, day, and day of the week) or time data (hour, minute, and second) must be read using a BASIC program from a PC CPU which has the clock function.

The following PC CPUs have a clock function:

- A1N
- A2N(S1)
- A3N
- A2A(S1)
- A3A
- A73
- A1S

When using the clock function of a PC CPU, the calendar and time data must have been written to the clock device.



[Commands and functions]

The following list gives the commands and functions used for reading/writing the calendar and time from/to a PC CPU.

Commands and Functions	Processing
DATE$	Reads and writes calendar data from/to the clock device of a PC CPU.
TIME$	Writes data to the clock device of a PC CPU and reads the time from it.

7.7 Data Communications Between Tasks

Details about the variables and array of each task can be referred to the BASIC program for that task during multitasking.

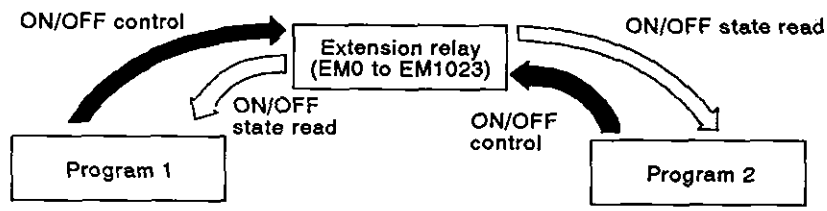
The following is used for data communications between BASIC programs during multitasking:

1) Extension relay (EM) : ON/OFF data
2) Extension register (ED) } : Data
3) Common memory }

7.7.1 Communicating ON/OFF data using the extension relay (EM)

ON/OFF control of, and read of ON/OFF state data stored in, the extension relay (EM) of the AD51H can be executed from all tasks.

(Section 6.5.3 gives details about extension relays.)



The AD51H can use 1024 extension relay points from EM0 to EM1023.

[Commands and functions]

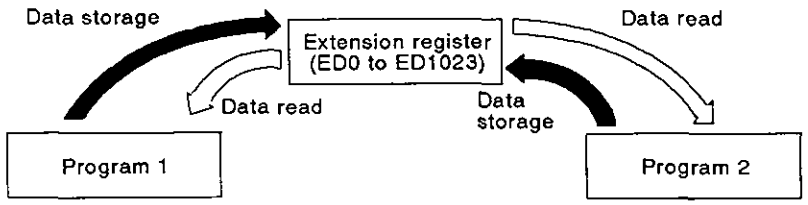
The following list gives the commands and functions used for receiving and sending ON/OFF data when using an extension relay.

Commands and Functions	Processing
B@	Reads ON/OFF control data and ON/OFF state data.

7.7.2 Data communications using the extension register (ED)

Data can be read/written from/to the extension register (ED) of the AD51H using any task.

(Section 6.5.4 gives details about extension registers.)



The AD51H can use 1024 extension register points from ED0 to ED1023.

[Commands and functions]

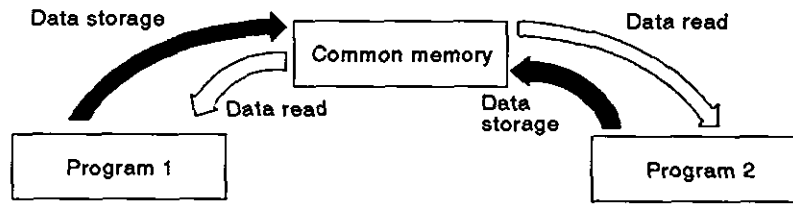
The following list gives the commands and functions used for receiving and sending data from/to an extension register.

Commands and Functions	Processing
W@	*Reads and stores data from/in an extension register.*
GETMEM	Reads data from an extension register.
PUTMEM	Stores data in the extension register.

7.7.3 Data communications using the common memory

The AD51H buffer is the common memory from/to which data can be read/written using any task.

(Section 6.5.6 gives details about the common memory.)



The AD51H can use 1K-word (16K bytes) common memory.

[Commands and functions]

The following list gives the commands and functions used for receiving and sending data when using the common memory.

Commands and Functions	Processing
GETMEM	Reads data from the common memory.
PUTMEM	Stores data in the common memory.

8. PROGRAMMING AND DEBUGGING

This section discusses the system configurations, AD51H switch settings, and reference manuals needed for programming and debugging a BASIC program.

8.1 Creating a BASIC Program

A BASIC program for operating the AD51H can be created as given below:

(1) In the online mode

A BASIC program can be created by connecting a console to the AD51H.

A PC/AT, VG-620, VT-382, or VT-220 can be used as the console.

(2) In the offline mode

A BASIC program can only be created using a PC/AT.

A VG-620, VT-382, or VT-220 cannot be used.

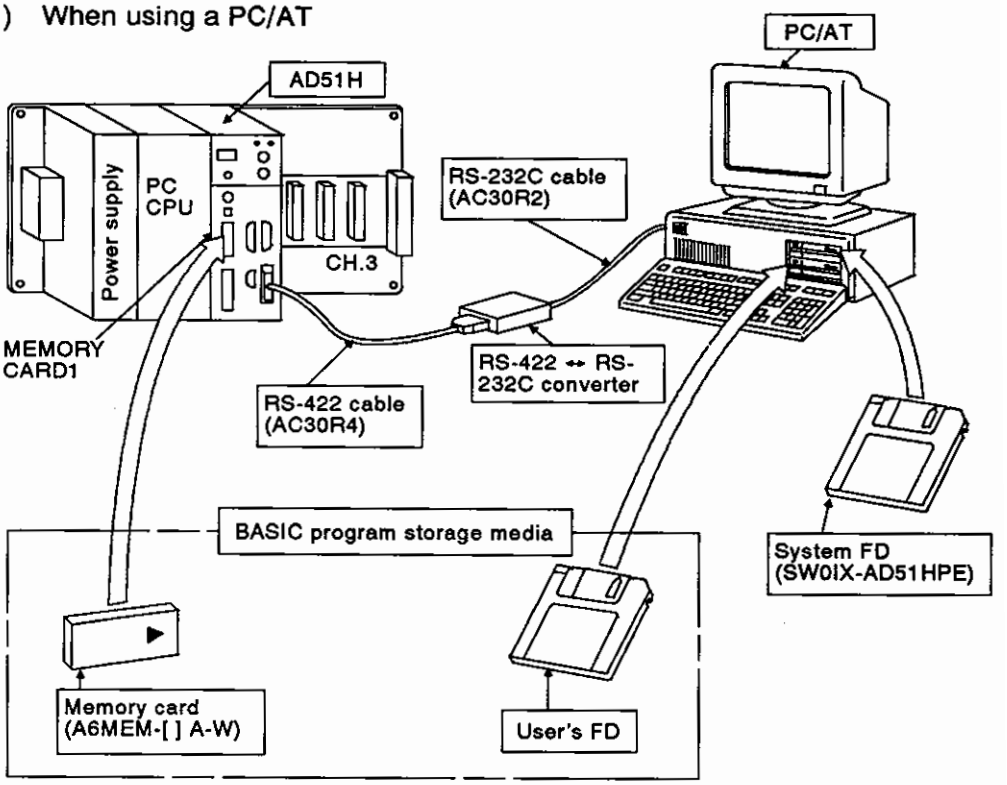
8.1.1 Creating and storing a BASIC program in the online mode

This section discusses the system configurations, AD51H switch settings, and reference manuals needed for creating a BASIC program in the online mode by connecting a console to the AD51H.

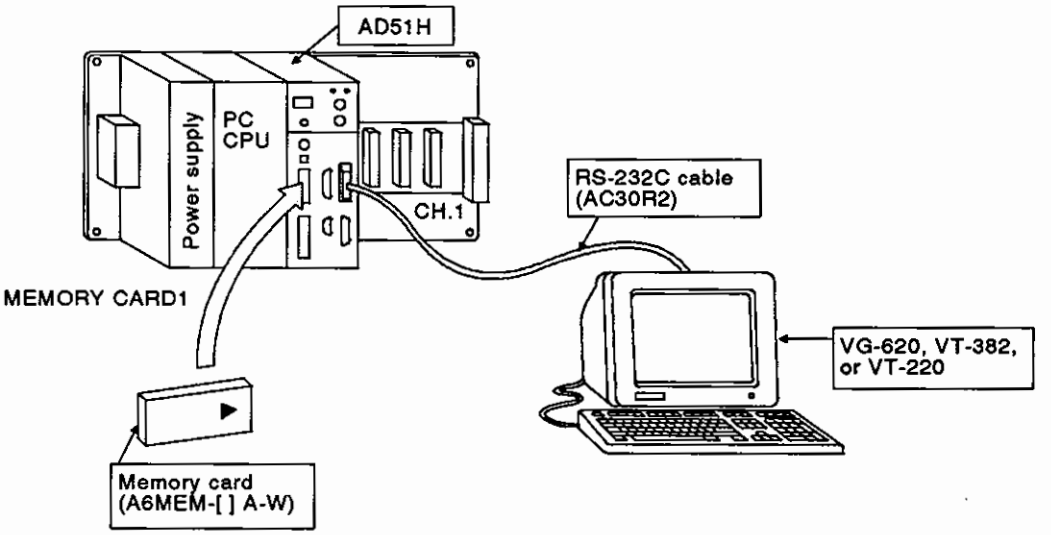
[System configuration]

The illustration below shows the system configuration needed for creating a BASIC program by connecting a console to the AD51H.

(1) When using a PC/AT



(2) When using a VG-620, VT-382, or VT-220



[Necessary devices]

The following list gives the equipment needed for creating a BASIC program by connecting a console to the AD51H.

Name	Model Name	Console		
		PC/AT	VG-620	VT-382 or VT-220
Cable	AC30R4	o	—	—
	AC30R2	o	o	o
System FD	MS-DOS (3.21 or later)	o	—	—
	SW0IX-AD51HPE			
User's FD *1	2HD, 2DD	o	—	—
Memory card *2	A6MEM-[ ] A-W		o	o
RS-422 ↔ RS-232C converter		o	—	—

REMARK

1) o : Necessary device

2) *1 : Used to store a BASIC program to an FD.

3) *2 : Used to store a BASIC program to a memory card.

Section 6.2 gives details about a memory card.

[AD51H switch settings]

This section discusses the settings needed for creating a BASIC program by using a console connected to the AD51H.

Section 4.1 gives details about setting switch positions.

(1) Setting a console

Set a console by using SW1 to SW5 of mode setting switch 2.

Section 4.2.2 gives details about settings when a PC/AT, VG-620, VT-382, or VT-220 is used as a console.

(2) Setting the memory-protect function to disabled (when using a memory card)

Set the memory-protect function to disabled by using the memory-protect keyswitch.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch (when using a memory card)

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON.

(Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

Create a BASIC program by operating a console.

The online programming/debugging part of this manual gives details about console operations.

POINT

When using a new memory card, format the memory card.

- Format a memory card by using the CFORMAT command or in the file maintenance mode of the PC/AT started up with the SW0IX-AD51HPE.

The online programming/debugging part of this manual gives details about the CFORMAT command.

The SW0IX-AD51HPE Operating Manual gives details about the PC/AT file maintenance mode.

REMARK

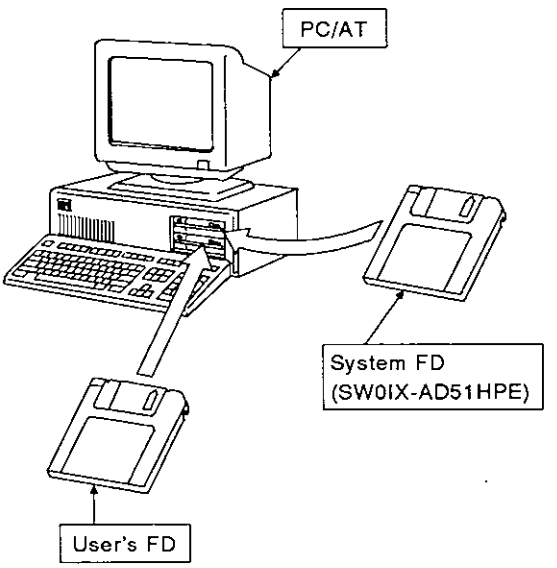
The AD51H-BASIC Programming Manual gives details about the use of the AD51H-BASIC to create a BASIC program.

8.1.2 Creating and storing a BASIC program in the offline mode

A BASIC program can be created by using a PC/AT.

This section discusses the system configuration and reference manuals needed for creating a BASIC program in the offline mode by using only a PC/AT.

[System configuration]



[Necessary devices]

The following list gives the devices that are needed for creating a BASIC program by using only a PC/AT.

Name	Model Name	Console
		PC/AT
System FD	MS-DOS (3.21 or later)	o
	SW0IX-AD51HPE	
User's FD	2HD, 2DD	o

REMARK

o: Necessary device

[Operating method]

The SW0IX-AD51HPE Operating Manual gives details about PC/AT operations.

8.2 Reading/Storing a BASIC Program From/To a Memory Card

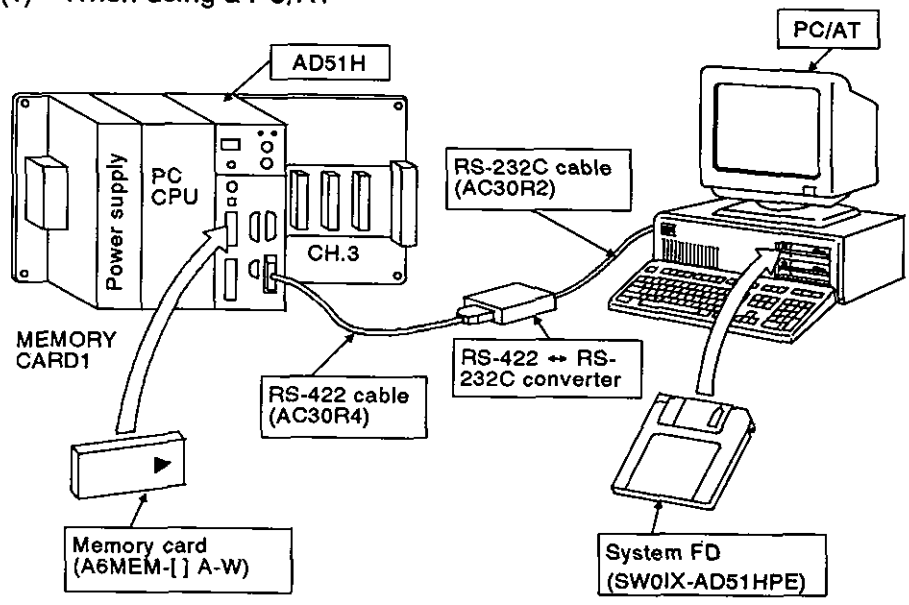
This section discusses the system configuration, AD51H switch settings, and reference manuals needed for reading a BASIC program from a memory card installed in MEMORY CARD1 or 2.

The system configurations and AD51H switch settings are the same as those given in Section 8.1.1.

[System configuration]

The illustration below shows the system configuration needed for reading/storing a BASIC program from/to a memory card.

(1) When using a PC/AT



[Necessary devices]

The following list gives the devices that are needed for reading/storing a BASIC program from/to a memory card.

Name	Model Name	Console		
		PC/AT	VG-620	VT-382 or VT-220
Cable	AC30R4	o	—	—
	AC30R2	o	o	o
System FD	MS-DOS (3.21 or later)	o	—	—
	SW0IX-AD51HPE			
Memory card	A6MEM-[ ] A-W	o	o	o
RS-422 ↔ RS-232C converter		o	—	—

REMARK

o: Necessary device

[AD51H switch settings]

This section discusses the settings needed for reading/storing a BASIC program from/to a memory card.

Section 4.1 gives details about setting switch positions.

(1) Setting a console

Set a console by SW1 to SW5 of mode setting switch 2.

Section 4.2.2 gives details about settings when a PC/AT, VG-620, VT-382, or VT-220 is used as a console.

(2) Setting the memory-protect function to disabled

Set the memory-protect function to disabled by using the memory-protect keyswitch on the front of the AD51H.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch (when using a memory card)

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON.

(Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

Reading/storing a BASIC program from/to a memory card is executed by using a console.

(1) Read/store a BASIC program from/to the execution program area of a memory card by using a MSAVE/MLOAD system command.

(2) Read/store a BASIC program from/to the file area of a memory card by using the a SAVE/LOAD command of the AD51H-BASIC.

The online programming/debugging part of this manual gives details about MSAVE/MLOAD commands and console system commands.

The AD51H-BASIC Programming Manual gives details about AD51H-BASIC SAVE/LOAD commands.

8.3 Reading/Storing a BASIC Program From/To a User's FD

This section discusses the system configuration and reference manuals needed for reading/storing a BASIC program from/to a user's FD (2HD or 2DD) inserted in the FDD of a PC/AT.

The system configuration and AD51H switch settings are the same as those given in Section 8.1.1.

The area when creating a BASIC program in the online mode is different from the area when creating a BASIC program in the offline mode.

When creating a BASIC program in the online mode

(a) Memory card in MEMORY CARD1 → User's FD

(b) User's FD → Memory card in MEMORY CARD1

When creating a BASIC program in the offline mode

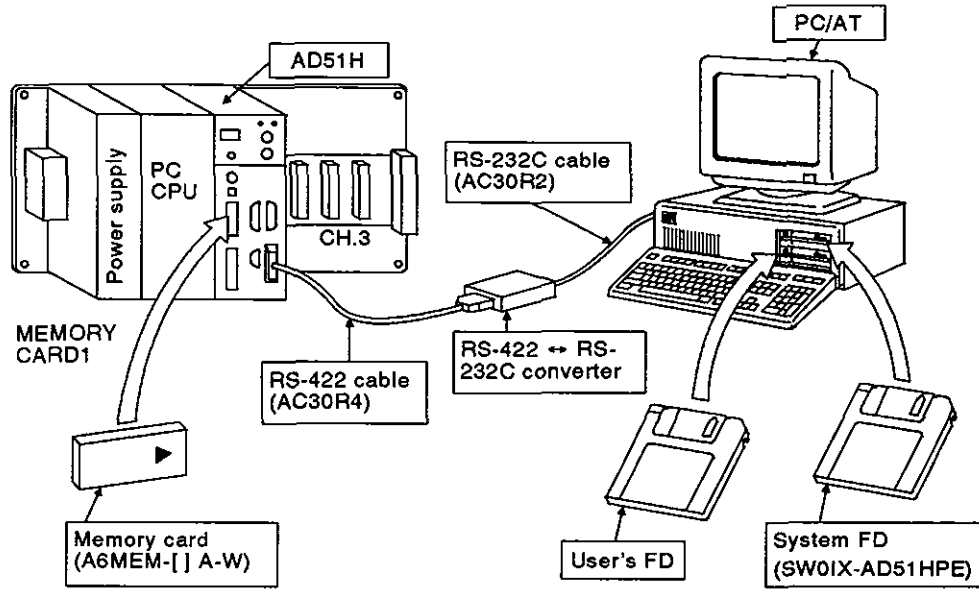
(a) PC/AT → User's FD

(b) User's FD → PC/AT

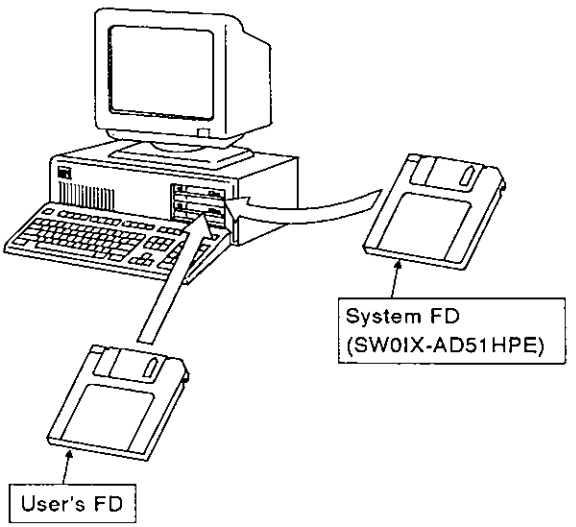
[System configuration]

The illustration below shows the system configuration needed for reading/storing a BASIC program from/to a user's FD.

(1) When creating a BASIC program in the online mode



(2) When creating a BASIC program in the offline mode



[Necessary devices]

The following list gives the devices that are needed for reading/storing a BASIC program from/to a user's FD.

Name	Model Name	Online	Offline
Console	PC/AT	o	o
Cable	AC30R4, AC30R2	o	—
System FD	MS-DOS (3.21 or later) SW0IX-AD51HPE	o	o
User's FD	2HD, 2DD	o	o
Memory card	A6MEM-[ ] A-W	o	—
RS-422 ↔ RS-232C converter		o	—

[AD51H switch settings]

This section discusses the settings needed for reading/storing a BASIC program from/to a user's FD in the AD51H in the online mode.

Section 4.1 gives details about setting switch positions.

(1) Setting a console

Set a console by SW1 to SW5 of mode setting switch 2.

Section 4.2.2 gives details about settings required when a PC/AT is used as a console.

(2) Setting the memory-protect function to disabled

Set the memory-protect function to disabled by using the memory-protect keyswitch on the front of the AD51H.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON. (Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

Read/store a BASIC program from/to a user's FD in the AD51H by using a PC/AT (console).

(1) Online

Read a BASIC program from a user's FD to the program area of the AD51H by using the LOAD command of the AD51H-BASIC.

Write a BASIC program to a user's FD from the program area of the AD51H by using the SAVE command of the AD51H-BASIC.

(2) Offline

Read a BASIC program from a user's FD to the internal memory of PC/AT by using the LOAD command of the AD51H-BASIC.

Write a BASIC program to a user's FD from the internal memory of PC/AT by using the SAVE command of the AD51H-BASIC.

The AD51H-BASIC Programming Manual gives details about SAVE and LOAD commands.

8.4 Checking and Modifying Multitask Settings on a Display

Multitask settings of an execution program stored in a memory card can be displayed and checked on a console.

Multitask settings can also be modified.

8.4.1 Checking multitask settings on a display

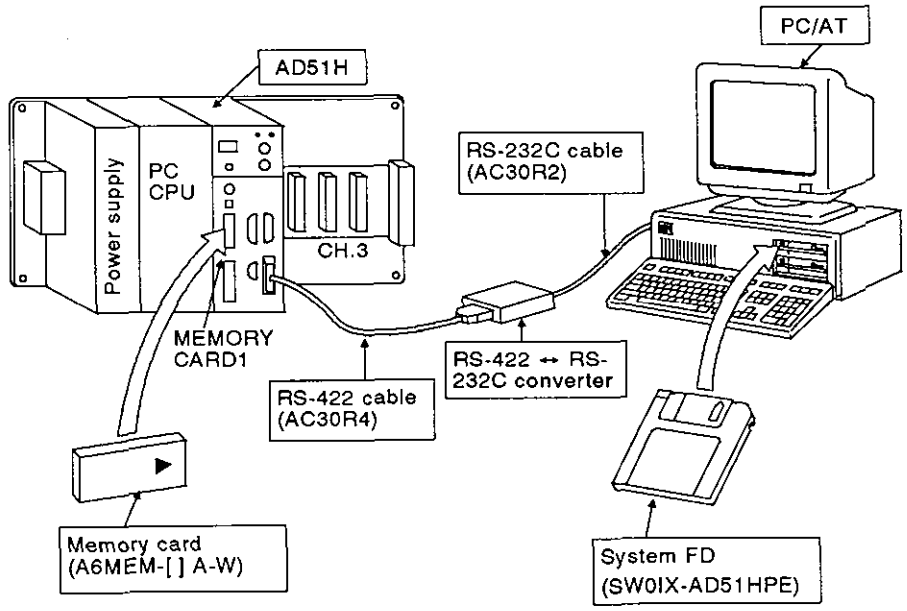
This section discusses the system configuration, AD51H switch settings, and reference manuals needed for displaying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

The system configuration and AD51H switch settings are the same as those given in Section 8.1.1.

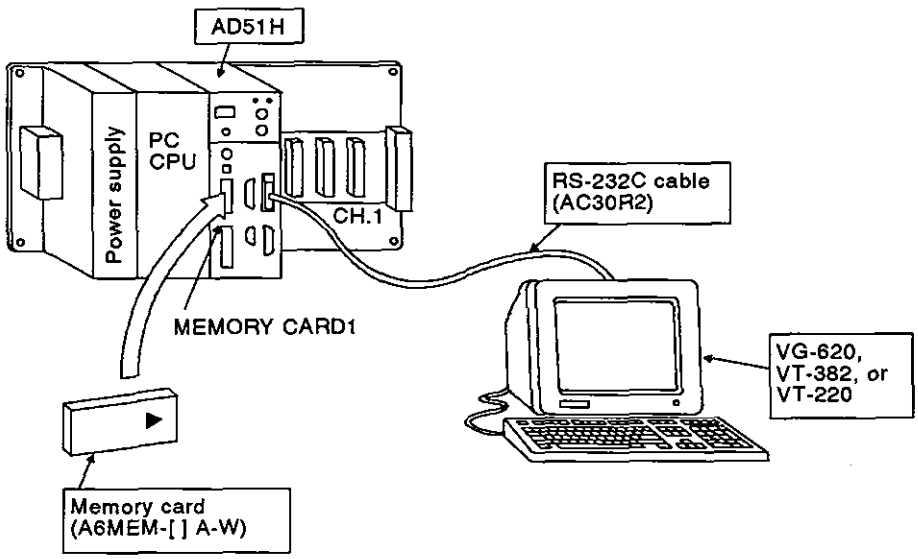
[System configuration]

The illustration below shows the system configuration needed for displaying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

(1) When using a PC/AT



(2) When using a VG-620, VT-382, or VT-220



[Necessary devices]

The following list gives the devices needed for displaying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

Name	Model Name	Console		
		PC/AT	VG-620	VT-382 or VT-220
Cable	AC30R4	o	—	—
	AC30R2	o	o	o
System FD	MS-DOS (3.21 or later)	o	—	—
	SW0IX-AD51HPE			
Memory card	A6MEM-[ ] A-W	o	o	o
RS-422 ↔ RS-232C converter		o	—	—

o: Necessary device

[AD51H switch settings]

This section discusses the settings needed for displaying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

Section 4.1 gives details about setting switch positions.

(1) Setting a console

Set a console by SW1 to SW5 of mode setting switch 2.

Section 4.2.2 gives details about settings when a PC/AT, VG-620, VT-382, or VT-220 is used as a console.

(2) Setting the memory-protect function to disabled

Set the memory-protect function to disabled by using the memory-protect keyswitch on the front of the AD51H.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON.

(Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

To display multitask settings, use the SET? system command.

The online programming/debugging part of this manual, Section 3.2, gives details about the SET? system command.

8.4.2 Modifying multitask settings

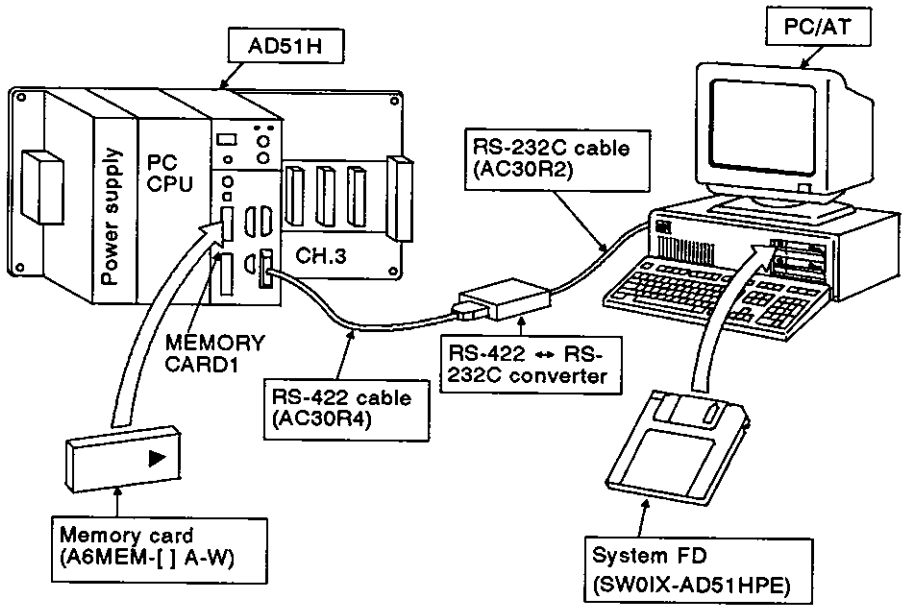
This section discusses the system configuration, AD51H switch settings, and reference manuals needed for modifying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

The system configuration and AD51H switch settings are the same as those given in Section 8.1.1.

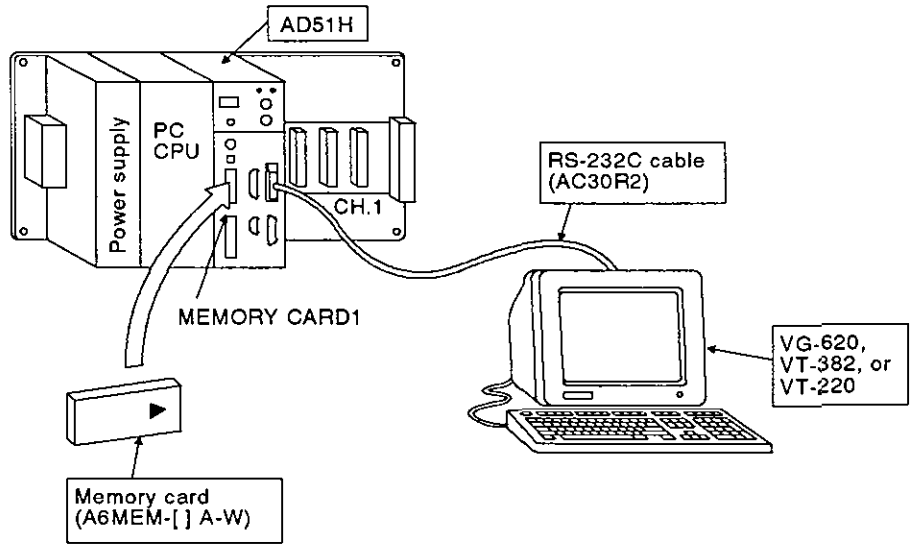
[System configuration]

The illustration below shows the system configuration that is required for modifying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

(1) When using a PC/AT



(2) When using a VG-620, VT-382, or VT-220



[Necessary devices]

The following list gives the devices that are needed for modifying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

Name	Model Name	Console		
		PC/AT	VG-620	VT-382 or VT-220
Cable	AC30R4	o	—	—
	AC30R2	o	o	o
System FD	MS-DOS (3.21 or later)	o	—	—
	SW0IX-AD51HPE			
Memory card	A6MEM-[ ] A-W	o	o	o
RS-422 ↔ RS-232C converter		o	—	—

o: Necessary device

[AD51H switch settings]

This section discusses the settings needed for displaying multitask settings of an execution program stored in the memory card in MEMORY CARD1.

Section 4.1 gives details about setting switch positions.

(1) Setting a console

Set a console by SW1 to SW5 of mode setting switch 2.

Section 4.2.2 gives details about settings when a PC/AT, VG-620, VT-382, or VT-220 is used as a console.

(2) Setting the memory-protect function to disabled

Set the memory-protect function to disabled by using the memory-protect keyswitch on the front of the AD51H.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON.

(Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

To modify multitask settings of an execution program of a memory card, use the SET system command.

The control function part of this manual, Section 6.4, gives details about the multitask setting commands (START, IT, BOOT, ON).

The online programming/debugging part of this manual, Section 3.2, gives details about the SET system command.

8.5 Debugging a BASIC Program

The AD51H can perform either single-task or multitask debugging of a BASIC program created by offline or online programming.

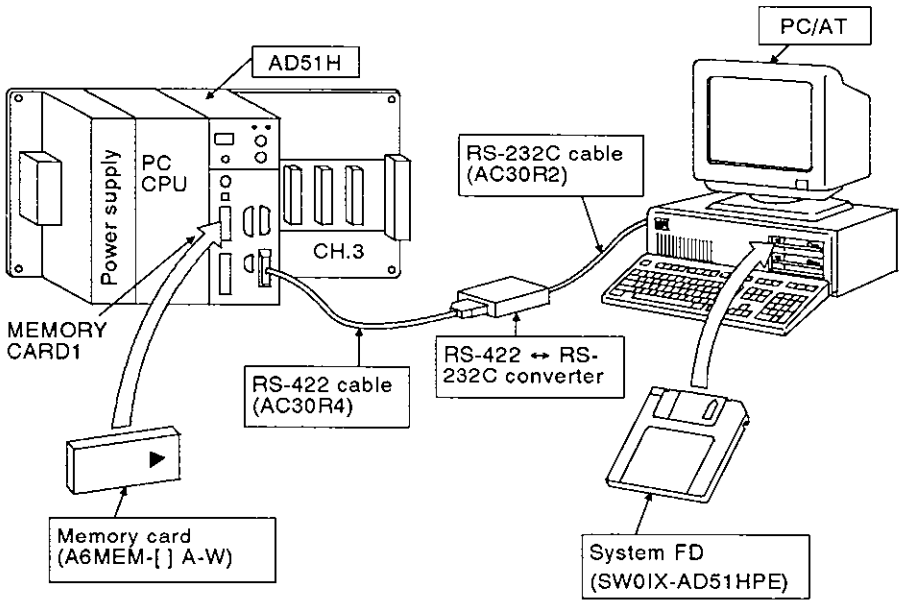
8.5.1 Single-task debugging

This section discusses the system configuration, AD51H switch settings, and reference manuals needed for single-task debugging of a BASIC program created on a console.

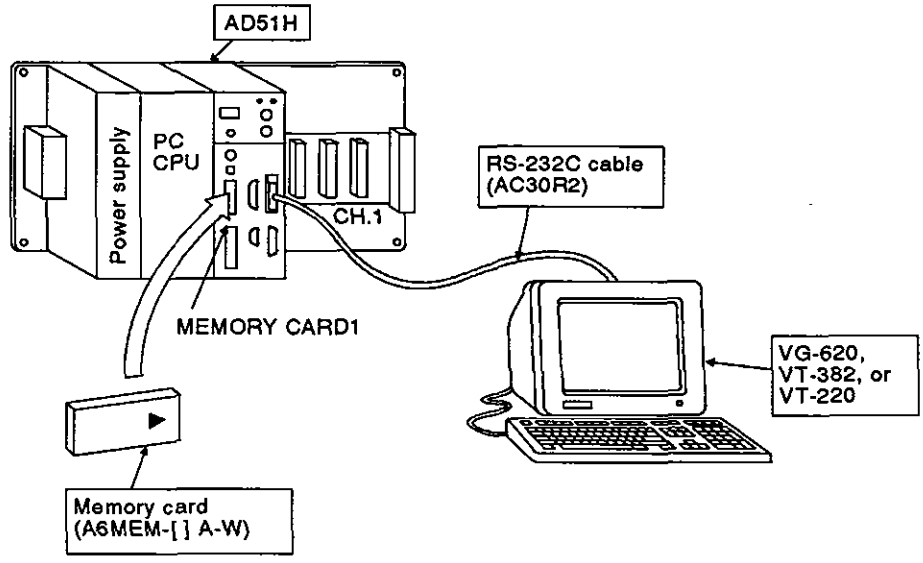
[System configuration]

The illustration below shows the system configuration needed for single-task debugging of a BASIC program created on a console.

(1) When using a PC/AT



(2) When using a VG-620, VT-382, or VT-220



[Necessary devices]

The following list gives the devices that are needed for single-task debugging of a BASIC program created on a console.

Name	Model Name	Console		
		PC/AT	VG-620	VT-382 or VT-220
Cable	AC30R4	o	—	—
	AC30R2	o	o	o
System FD	MS-DOS (3.21 or later)	o	—	—
	SW0IX-AD51HPE			
Memory card	A6MEM-[ ] A-W	o	o	o
RS-422 ↔ RS-232C converter		o	—	—

o: Necessary device

[AD51H switch settings]

This section discusses the settings needed for single-task debugging of a BASIC program created on a console.

Section 4.1 gives details about setting switch positions.

(1) Setting a console

Set a console by using mode setting switches.

Section 4.2.2 gives details about settings required when a PC/AT, VG-620, VT-382, or VT-220 is used as a console.

(2) Setting the memory-protect function to disabled (when using a memory card)

Set the memory-protect function to disabled by using the memory-protect keyswitch on the front of the AD51H.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch (when using a memory card)

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON.

(Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

Debug a BASIC program task by task created on a console by using the RUN, TROFF, and TRON commands of the AD51H-BASIC.

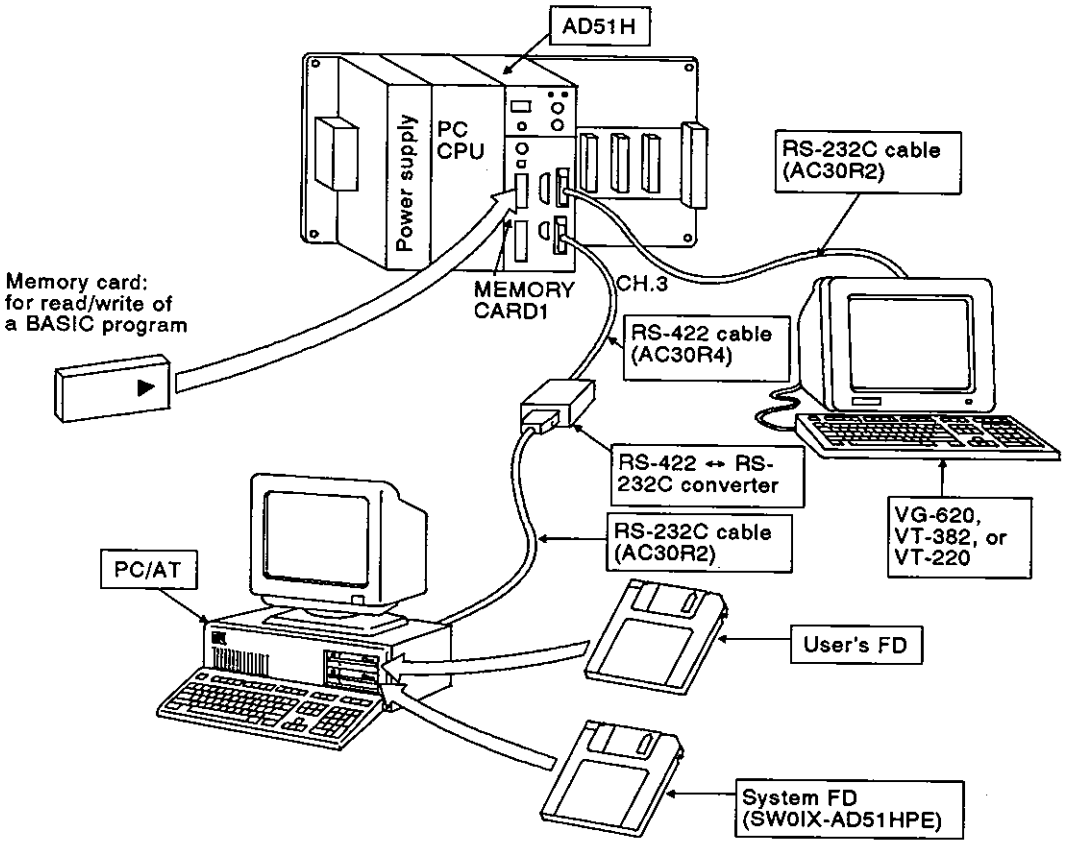
The AD51H-BASIC Programming Manual gives details about RUN, TROFF, and TRON commands.

8.5.2 Multitask debugging

This section discusses the system configuration, AD51H switch settings, and reference manuals needed for multitask debugging after single-task debugging is completed.

[System configuration]

The illustration below shows the system configuration needed for multitask debugging.



[Necessary devices]

The following list gives the devices that are needed for multitask debugging.

Name	Model Name	Console/Debugger		
		PC/AT	VG-620	VT-382
Cable	AC30R4	o	—	—
	AC30R2	o	o	o
System FD	MS-DOS (3.21 or later)	o	—	—
	SW0IX-AD51HPE			
Memory card	A6MEM-[ ] A-W	o	o	o
RS-422 ↔ RS-232C converter		o	—	—

o: Necessary device

[AD51H switch settings]

This section discusses the settings needed for multitask debugging.

Section 4.1 gives details about setting switch positions.

(1) Setting a console and a debugger

Set a console and a debugger by using mode setting switches.

Section 4.2.2 gives details about settings required when a PC/AT, VG-620, VT-382, or VT-220 is used as a console and a debugger.

(2) Setting the memory-protect function to disabled (when using a memory card)

Set the memory-protect function to disabled by using the memory-protect keyswitch on the front of the AD51H.

When the memory-protect keyswitch is set to OFF, the memory-protect function becomes disabled, and a BASIC program can be written to a memory card.

(3) Setting the memory card access switch (when using a memory card)

Section 4.3.4 tells how to insert/remove a memory card.

After inserting or removing a memory card, set the memory card access switch on the front of the AD51H to ON.

(Access to the memory card is enabled.)

(4) Setting the execution mode to the programming mode

Set the execution mode by using the mode setting switch on the front of the AD51H.

When switching the mode to the programming mode, set the mode setting switch to "4".

(5) Setting the RUN keyswitch

Set the RUN keyswitch to RUN or STOP.

[Operating method]

Execute multitask debugging by using a debugger connected to the AD51H.

The online programming/debugging part of this manual, Section 4, gives details about debugging operations.

8.6 Writing an Execution Program to a ROM and Executing a Program Stored in the ROM

An execution program stored in a memory card is booted to the AD51H for multitasking when the power is turned ON.

Also, an execution program can be written to a ROM and booted to the AD51H for multitasking when the power is turned ON. This is called the ROM operation.

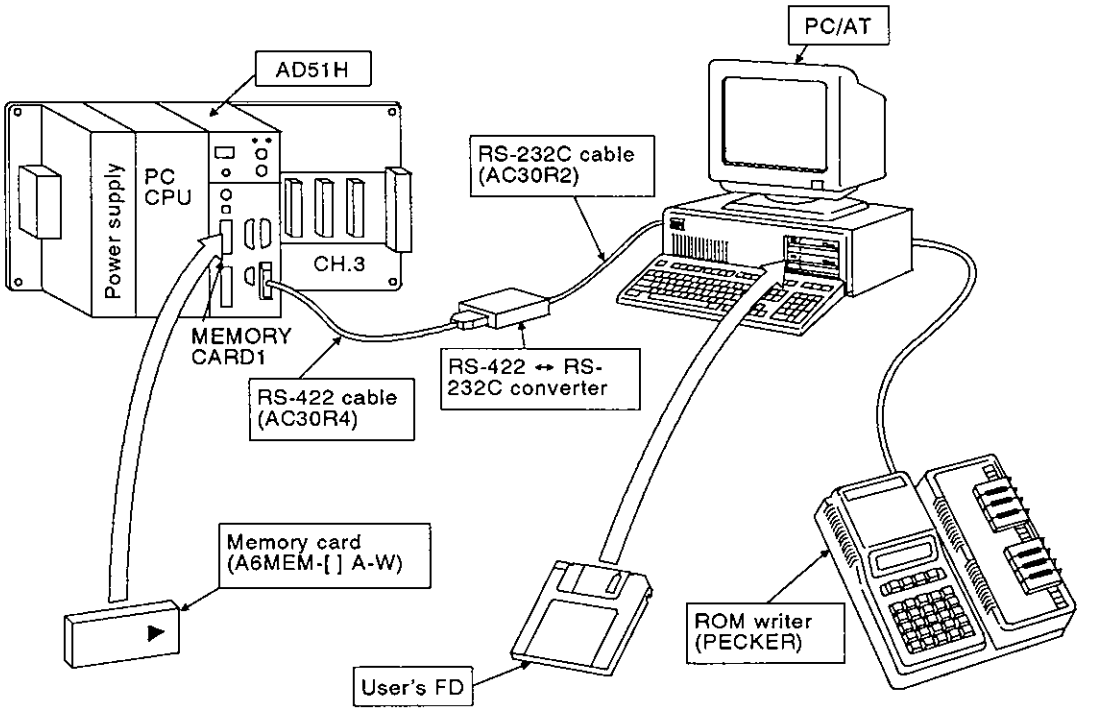
This section discusses the system configuration, AD51H switch settings, and reference manuals needed for writing an execution program to a ROM and executing the ROM operation.

8.6.1 Writing an execution program stored in a memory card to a ROM

This section discusses the system configuration, AD51H switch settings, and reference manuals needed for writing an execution program stored in a memory card installed in MEMORY CARD1 of the AD51H to a ROM.

[System configuration]

The illustration below shows the system configuration needed for writing an execution program stored in a memory card to a ROM.

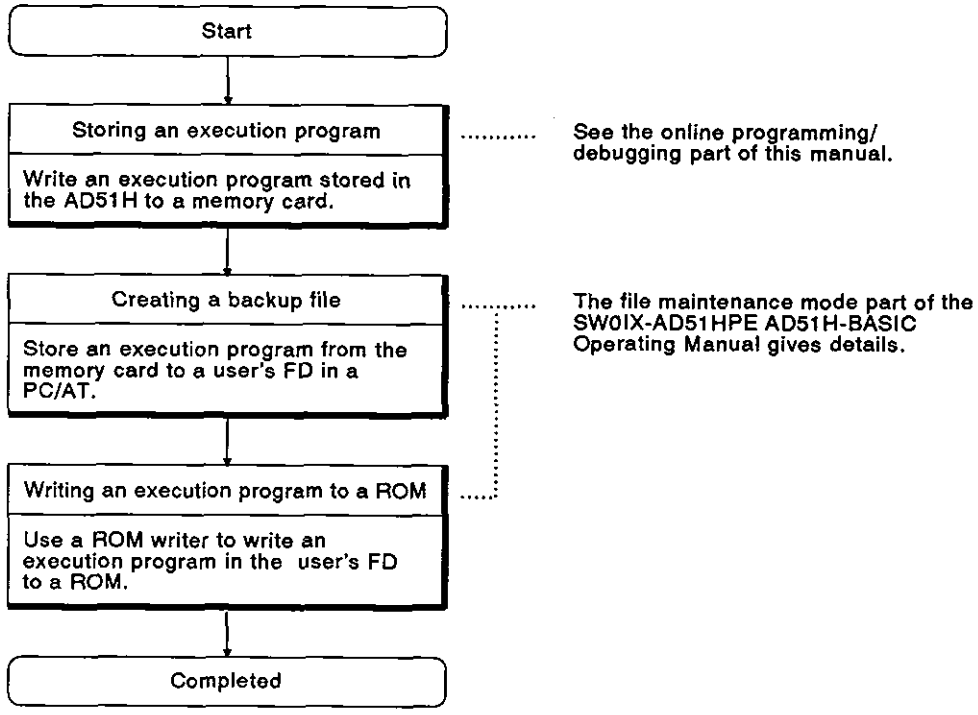


[Necessary devices]

The following list gives the devices needed for writing an execution program stored in a memory card to a ROM.

Name	Model Name	Remarks
Console	PC/AT	—
Cable	AC30R4, AC30R2	—
System FD	MS-DOS (3.21 or later)	Used for installation to a PC/AT.
	SW0IX-AD51HPE	
User's FD	2HD, 2DD	Used to create a backup file of an execution program and the file of an execution program to be written to a ROM.
Memory card	A6MEM-[ ] A-W	Section 6.7 gives details.
ROM writer	PECKER [ ]	—
RS-422 ↔ RS-232C converter		—

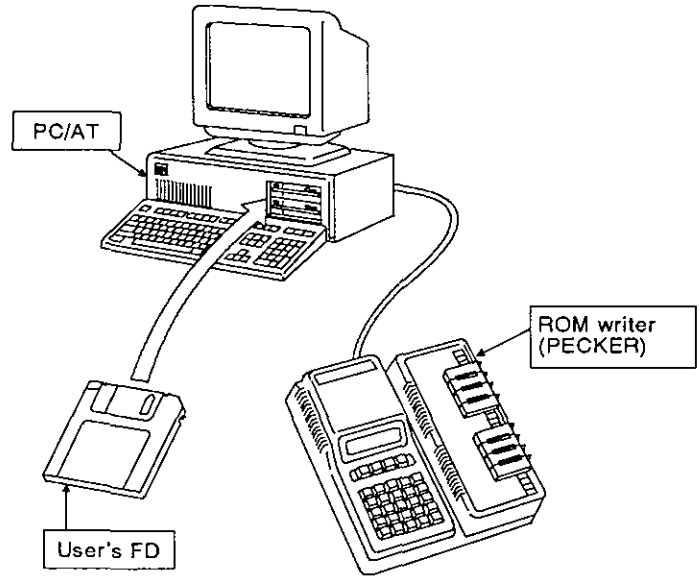
[Operating procedure and reference manuals]



8.6.2 Writing an execution program stored in a user's FD in a PC/AT to a ROM

This section gives the system configuration and reference manuals needed for writing an execution program stored in a user's FD in a PC/AT to a ROM.

[System configuration]

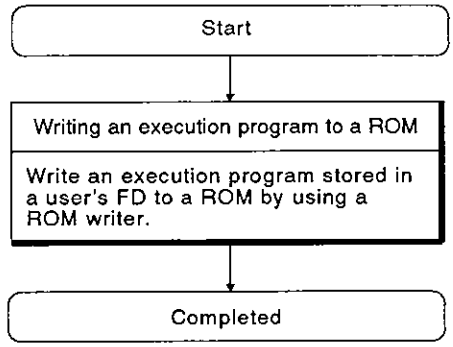


[Necessary devices]

The following list gives the devices needed for writing an execution program stored in a user's FD to a ROM.

Name	Model Name	Remarks
Console	PC/AT	—
System FD	MS-DOS (3.21 or later)	Used for installation to a PC/AT.
	SW0IX-AD51HPE	
User's FD	*2HD, 2DD*	Used to create a backup file of an execution program and the file of an execution program to be written to a ROM.
ROM writer	PECKER [ ]	—

[Operating procedure and reference manuals]



..... The file maintenance mode part of the SW0IX-AD51HPE AD51H-BASIC Operating Manual gives details.

9. MESSAGES DISPLAYED ON THE LED INDICATOR

Four types of messages are displayed on the LED indicator on the front of the AD51H:

(a) Operating state messages

(b) Warning messages

(c) BASIC program error messages

(d) System-down messages

(1) Priorities and indicator resetting

The LED indicator of the AD51H displays messages in order of priority. Messages of higher priority are displayed first.

Some of them can be cleared by the indicator reset switch on the front of the AD51H.

The display priority and clear enable/disable setting of the indicator reset switch are shown below:

		Display priority	Indicator reset
(a)	Operating state messages	Low	Disabled
(b)	Warning messages	↑	Enabled
(c)	BASIC program error messages	↓	Enabled
(d)	System-down messages	High	Disabled

9.1 Operating State Messages

There are seven kinds of operating state messages as shown in Table 9.1.

Table 9.1 AD51H Operating State Messages

Message	Display Mode			Description
	Program-ming	Debugging	Execution	
BOOT	o	o	o	• Displayed when the power supply to the AD51H is turned ON or the AD51H is reset, and the AD51H system is booted.
CARD	o	o	o	• Displayed when the memory card install request switch is turned ON while a memory card is accessed. (OK is displayed when a memory card can be inserted or removed.)
INIT			o	• Displayed when an execution program is being loaded from a memory card or an EP-ROM to the AD51H.
OK			o	• Indicates that a memory card can be inserted or removed. • When the memory card install request switch is ON and if a file in the memory card is being accessed, this is displayed when the memory card is closed.
PROG	o			• Displayed when the mode selection rotary switch is set to programming mode (4).
RUN			o	• Multitasking is being executed.
STOP			o	• Displayed when multitasking is stopped because the RUN keyswitch is set to STOP.

9.2 Warning Messages

Warning messages are displayed when an error which does not lead to a system down of the AD51H occurs.

There are two kinds of warning messages as shown in Table 9.2.

Table 9.2 AD51H Warning Messages

Message	Error Code	Display Mode			Description	Corrective Action
		Programming	Debugging	Execution		
BAT1	0F 80	o	o	o	• Displayed when the battery of the memory card installed in MEMORY CARD1 is low.	Replace the battery (see Section 5.2).
BAT2	0F 81	o	o	o	• Displayed when the battery of the memory card installed in MEMORY CARD 2 is low.	Replace the battery (see Section 5.2).

* Error codes are stored in addresses ED9001 to ED9008.

9.3 System-Down Messages

System-down messages are displayed when an error which prevents the system from returning to normal occurs.

There are five kinds of system-down messages as shown in Table 9.3.

Table 9.3 AD51H System-down Messages

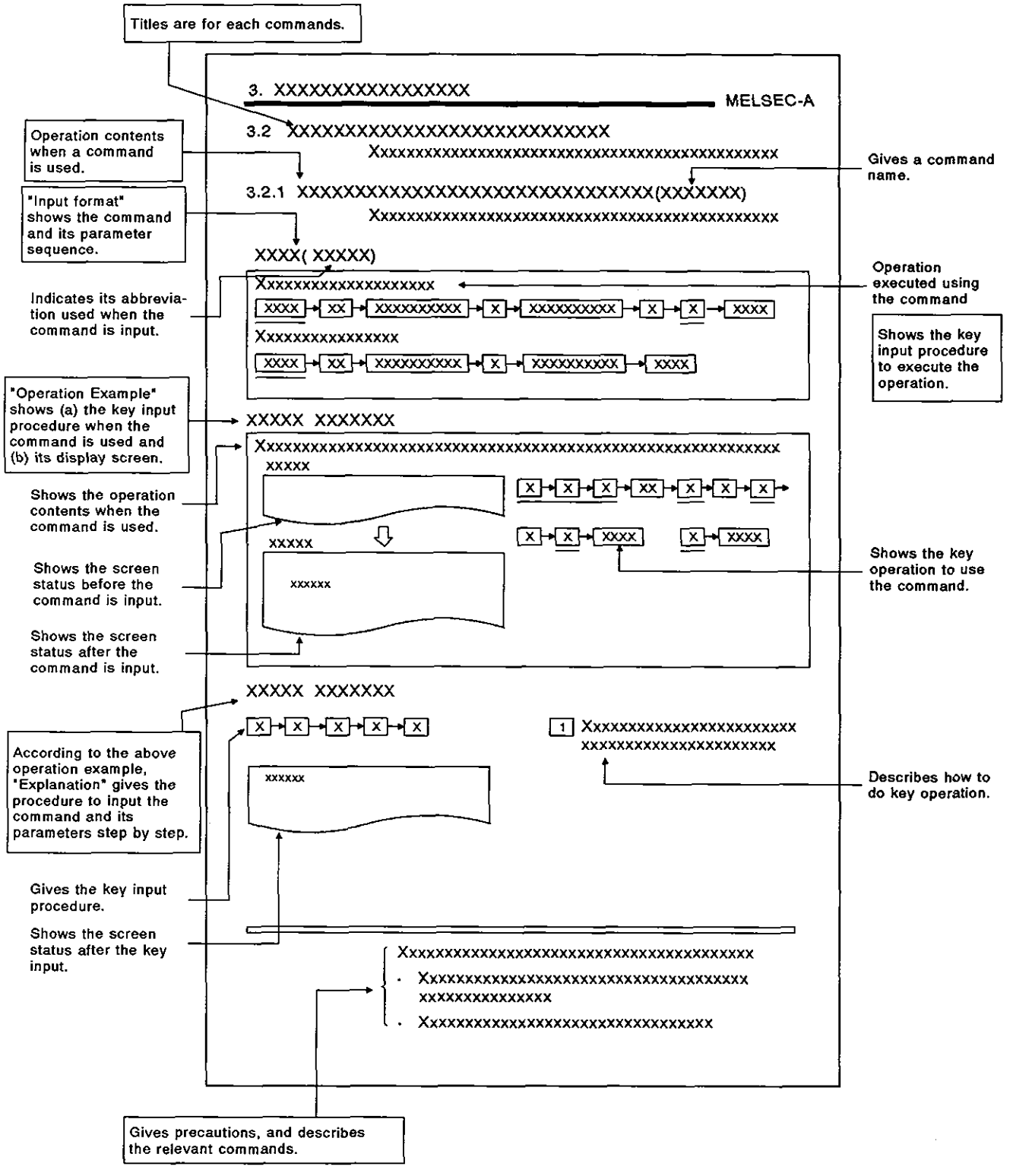
Message	Error Code	Display Mode			Description	Corrective Action
		Programming	Debugging	Execution		
INIE	0F 20			o	• Displayed when a memory card or EP-ROM in which an execution program is stored has not been installed.	• Install a memory card in which an execution program is stored. • Install an EP-ROM in which an execution program is stored.
KEYE	0F 10	o	o	o	• Displayed when the power supply to the AD51H is turned ON with the RUN keyswitch set to RESET.	• Set the RUN keyswitch to RUN or STOP, and turn ON the power supply again or reset the AD51H.
MTSE	0F 21			o	• Displayed when multitasking is incorrectly set.	• Reset multitasking correctly.
WDTE	0F 00	o	o	o	• Displayed when a WDT error occurs.	• Hardware fault Consult your nearest Mitsubishi representative.

[ONLINE PROGRAMMING & DEBUGGING]

CONTENS

1. COMMAND EXPLANATION FORMAT

This manual gives explanations about the commands as shown below:



2. STARTING UP THE AD51H AND MODE CHANGE

This section gives how the AD51H mode changes (when the AD51H starts up and after the AD51H starts up).

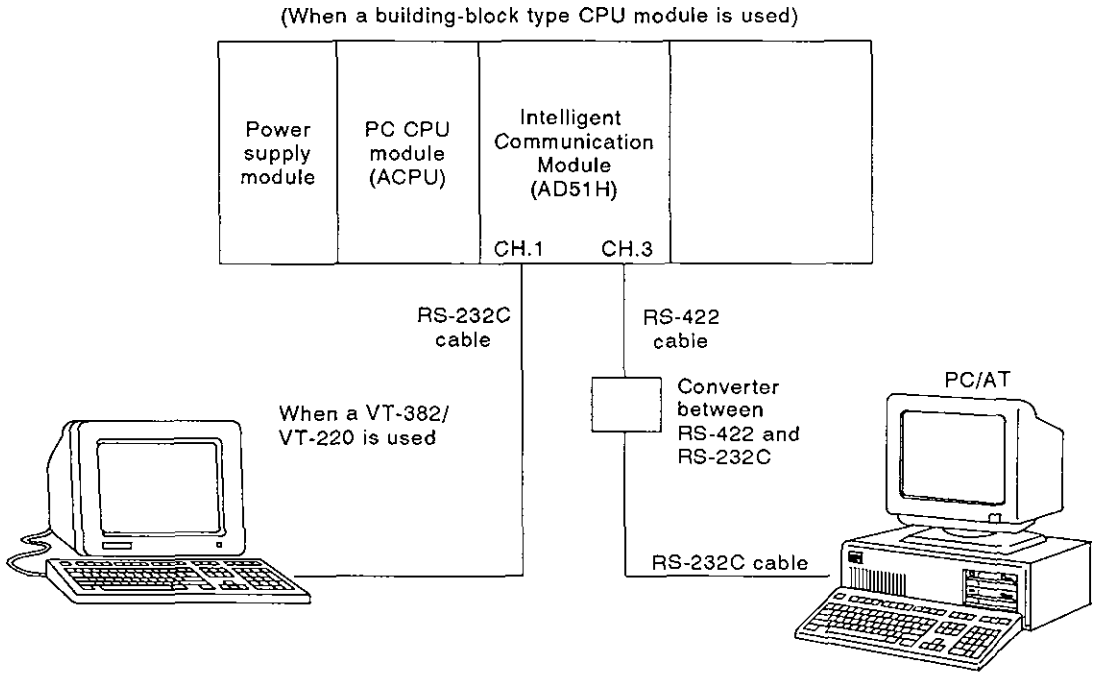
This development is related to the online programming (see Section 3) and the multitask debugging (see Section 4).

2.1 Using a PC/AT and a General-Purpose Terminal as the Console (Display Terminal) and the Debugger

This section describes how to start up the AD51H when a PC/AT and a General-Purpose Terminal as the Console and the debugger.

Use either the PC/AT (connected to the AD51H) or the general-purpose terminal (connected to the AD51H) as the console, and the other as the debugger.

The switch settings (AD51H mode setting switches SW1 to SW5) determine which machines are used as the console and the debugger. (See Section 2.1.1.)



2.1.1 Preparations required to start up the AD51H

This shows the preparations required before the system starts up.

For details, see the operating manuals of the used devices.

(1) AD51H

(a) Setting the AD51H

Set the switches used for operating the AD51H.

The Hardware part in this manual gives how to set the switches and how to use the switches.

The following switches must be set according to the operation mode when the AD51H starts up and to how to use the PC/AT and general-purpose terminal.

1) Mode setting switch 1

Set this switch to 0 to 4 according to the AD51H operation mode.

[0]/[1] : Used when executing a BASIC program that has already been debugged after the AD51H starts up.

[2]/[3] : Used when the multitask debugging (see Section 4) is done after the AD51H starts up.

[4] : Used when editing a BASIC program, debugging a single program, or operating in the system mode (see Section 3) after the AD51H starts up.

2) Mode setting switch 2

Set the switches SW1 to SW5 so that they match the machine used as the console, the machine used as the debugger, and the used interfaces.

(Example)

Set the switches as show below when using a PC as the console, a VT-382/VT-220 (connected to CH.1 of the AD51H) as the debugger:

No.	1	2	3	4	5
Setting	OFF	OFF	OFF	ON	ON

Section 4.2.1 in the manual (Hardware) gives information about how to set those switches.

(b) Installing memory cards

To store execution programs and data, install necessary memory cards in the AD51H. (Up to two cards can be installed.)

To install the cards, connect a battery beforehand when necessary.

When a memory card with a write-protect tab is used, make sure that the protect is released when a BASIC program is written after the AD51H starts up or the memory card is newly used.

Sections 2 and 4 in this manual (Hardware) give how to install a memory card and how to connect a battery.

(c) Loading the AD51H into a base unit

After setting the AD51H and installing memory cards, load the AD51H into a slot of the base unit.

(2) PC/AT

(a) Installing the software

Install the following the software packages in the PC/AT.

- Operating system : MS-DOS (Ver 3.21 or after)
- AD51H-BASIC software package : (SW0IX-AD51HPE)

(b) Connecting the AD51H to the PC/AT

Use a AC30R4 cable to connect the AD51H RS-422 interface (CH.3) to the PC/AT serial interface.

Connect (COM 1) using an converter between the RS-422 and RS-232C interfaces.

(3) General-purpose terminal

(a) When a VG-620 is used (*1)

1) Set the VG-620 USART mode as shown below:

- Baud rate : 9600 BPS
- Data length : 8-bit
- Stop bit : 2-bit
- Parity : None

2) Connecting the AD51H to the VG-620

Use an AC30R2 cable connect the AD51H RS-232C interface (CH.1/CH.2) to the VG-620 RS-232C interface.

- When the general-purpose terminal is used as the console: CH.1
- When the general-purpose terminal is used as the debugger: CH.1 or 2

(Set the utilized interfaces so that they are consistent with the switches SW1 to SW5 of the AD51H mode setting switch 2.)

REMARK

*1: The manual of the VG-620 gives information about how to set and connect the VG-620.

(b) When a VT-382/VT-220 is used (*1)

1) Set the VT-382 USART mode as shown below:

- Baud rate : 9600 BPS
- Data length : 8-bit
- Stop bit : 2-bit
- Parity : None

2) Connecting the AD51H to the VT-382/VT-220

Use an AC30R2 cable to connect the AD51H RS-232C interface (CH.1/CH.2) to the VT-382/VT-220 RS-232C interface.

- When the VT382/VT-220 is used as the console: CH.1
- When the VT382/VT-220 is used as the debugger: CH.1 or 2

(Set the utilized interfaces so that they are consistent with the switches SW1 to SW5 of the AD51H mode setting switch 2.)

REMARK

*1: The manual of the VT-382/VT-220 gives information about how to set and connect the VT-382/VT-220.

2.1.2 Starting up the AD51H

This section describes how to start up the AD51H used with a PC/AT and a general-purpose terminal.

(1) Starting up the PC/AT and the general-purpose terminal

The following gives how to start up the PC/AT and the general-purpose terminal.

(a) When starting up the PC/AT

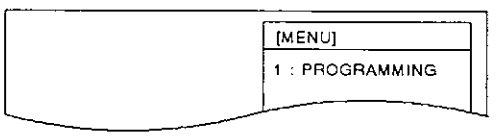
1) Power ON

Turn ON the power to the PC/AT.

2) Starting the SW0IX-AD51HP installed in the PC/AT

Input "D51HBASE", and press the [↵] key. Then, the SW0IX-AD51HPE starts.

- The SW0IX-AD51HPE main menu appears on the screen of the PC/AT.



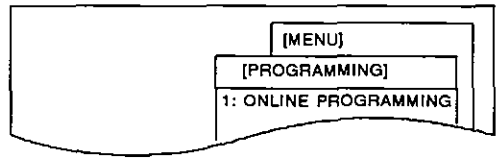
POINT

The SW0RX-A51HP AD51H-BASIC Operating Manual gives details about how to execute the following operations:

- Starting up the SW0IX-AD51HPE
- Setting the operating conditions for the PC/AT
- Setting the PC/AT to the online programming mode

3) Changing the PC/AT mode (to the online programming mode)

i) When the PC/AT screen displays the SW0IX-AD51HPE main menu, select the PROGRAMMING from the menu.



ii) Select ONLINE PROGRAMMING from the menu.

(b) When starting up the general-purpose terminal

Turn ON the power to the general-purpose terminal.

(2) Starting up the AD51H

Turn ON the power to the AD51H

According to the switches SW1 to SW5 of the AD51H mode setting switches 1 and 2, a prompt screen appears on the PC/AT and the general-purpose terminal.

AD51H Mode Setting Switch (1)	AD51H Mode	Display on the Console	Display on the Terminal (used for debugging)
		(Depends on the Mode Switch 2 Setting (SW1 to SW5))	
0 or 1	Execution mode (2)	Displays the data output from a BASIC program.	Displays the data output from a BASIC program.
2 or 3	Debug mode	Displays the data output from a BASIC program.	D>
4	System mode	S>	Displays the data output from a BASIC program.

(3) Starting the online programming operation/multitask debugging operation

(a) When the AD51H is in the debug mode

1) Execute the debugging operation in the multitasking system according to Section 4.

2) When changing the AD51H mode and continuing the operation, see Section 2.4.

(b) When the AD51H is in the system mode

1) Execute operations in the system mode according to Section 3.

2) To edit and debug a BASIC program, use the START command to return the AD51H to the editing mode (1), and execute the operation.

The AD51H Programming Manual gives how to edit and debug a BASIC program.

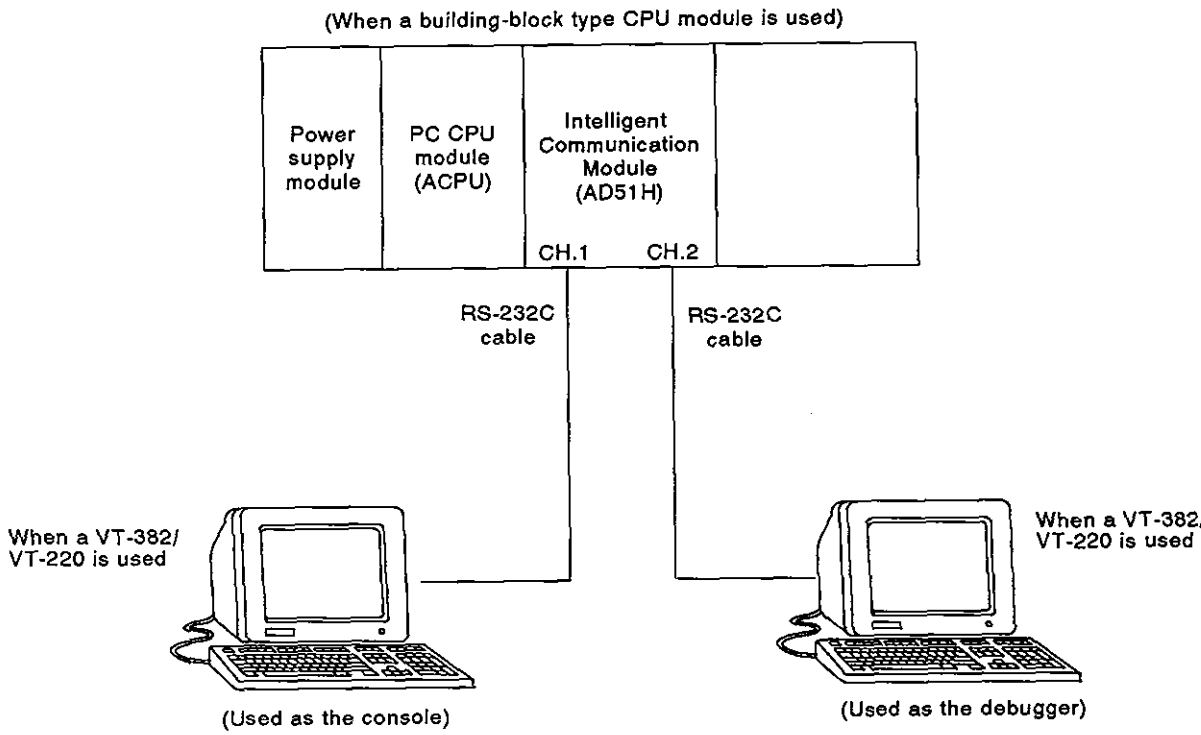
3) When changing the AD51H mode and continuing the operation, see Section 2.4.

2.2 Startup When Using Two General-Purpose Terminals as the Console and the Debugger

This section gives the AD51H startup procedure when using two general-purpose terminals as the console and the debugger.

When two general-purpose terminals are used:

- CH.1 corresponds to the console;
- CH.2 corresponds to the debugger.



2.2.1 Preparations required to start up the AD51H

This shows the preparations required before the system starts up.

For details, see the operating manuals of the used devices.

(1) AD51H

(a) Setting the AD51H

Set the switches used for operating the AD51H.

The Hardware part in this manual gives how to set the switches and how to use the switches.

The following switches must be set according to the operation mode when the AD51H starts up and to how to use the general-purpose terminals.

(Section 2.1.1 (1)-(a) gives general information.)

1) Mode setting switch 1

2) Mode setting switch 2

(Example)

Set the switches as show below when using two general-purpose terminals VT-382/VT-220 (connected to CH.1 and 2 of the AD51H respectively) as the console and the debugger:

No.	1	2	3	4	5
Setting	OFF	ON	ON	OFF	ON

Section 4 in the manual (Hardware) gives information about how to set those switches.

(b) Installing memory cards

To store execution programs and data, install necessary memory cards in the AD51H. (Up to two cards can be installed.)

To install the cards, connect a battery beforehand when necessary.

When a memory card with a write-protect tab is used, make sure that the protect is released when a BASIC program is written after the AD51H starts up or the memory card is newly used.

Sections 2 and 4 in this manual (Hardware) give how to install a memory card and how to connect a battery.

(c) Loading the AD51H into a base unit

After setting the AD51H and installing memory cards, load the AD51H into a slot of the base unit.

(2) General-purpose terminal

(a) When a VG-620 is used

When a VG-620 is used as the console or debugger for the AD51H, set and connect the VG-620 according to Section 2.1.1 (3)-(a).

(b) When a VT-382/VT-220 is used

When a VT-382/VT-220 is used as the console or debugger for the AD51H, set and connect the VT-382/VT-220 according to Section 2.1.1 (3)-(b).

2.2.2 Starting up the AD51H

This section describes how to start up the AD51H used with two general-purpose terminals.

(1) Power ON

(a) Turn ON the power to the general-purpose terminals.

(b) Then, turn ON the power to the AD51H.

(2) Starting the online programming operation/multitask debugging operation

After the AD51H starts up, according to the switches SW1 to SW5 of the AD51H mode setting switches 1 and 2, a prompt screen appears on the PC/AT and the general-purpose terminal.

(a) When the AD51H is in the debug mode

1) Execute the debugging operation in the multitasking system according to Section 4.

2) When changing the AD51H mode and continuing the operation, see Section 2.4.

(b) When the AD51H is in the system mode

1) Execute operations in the system mode according to Section 3.

2) To edit and debug a BASIC program, use the START command to return the AD51H to the editing mode (1), and execute the operation.

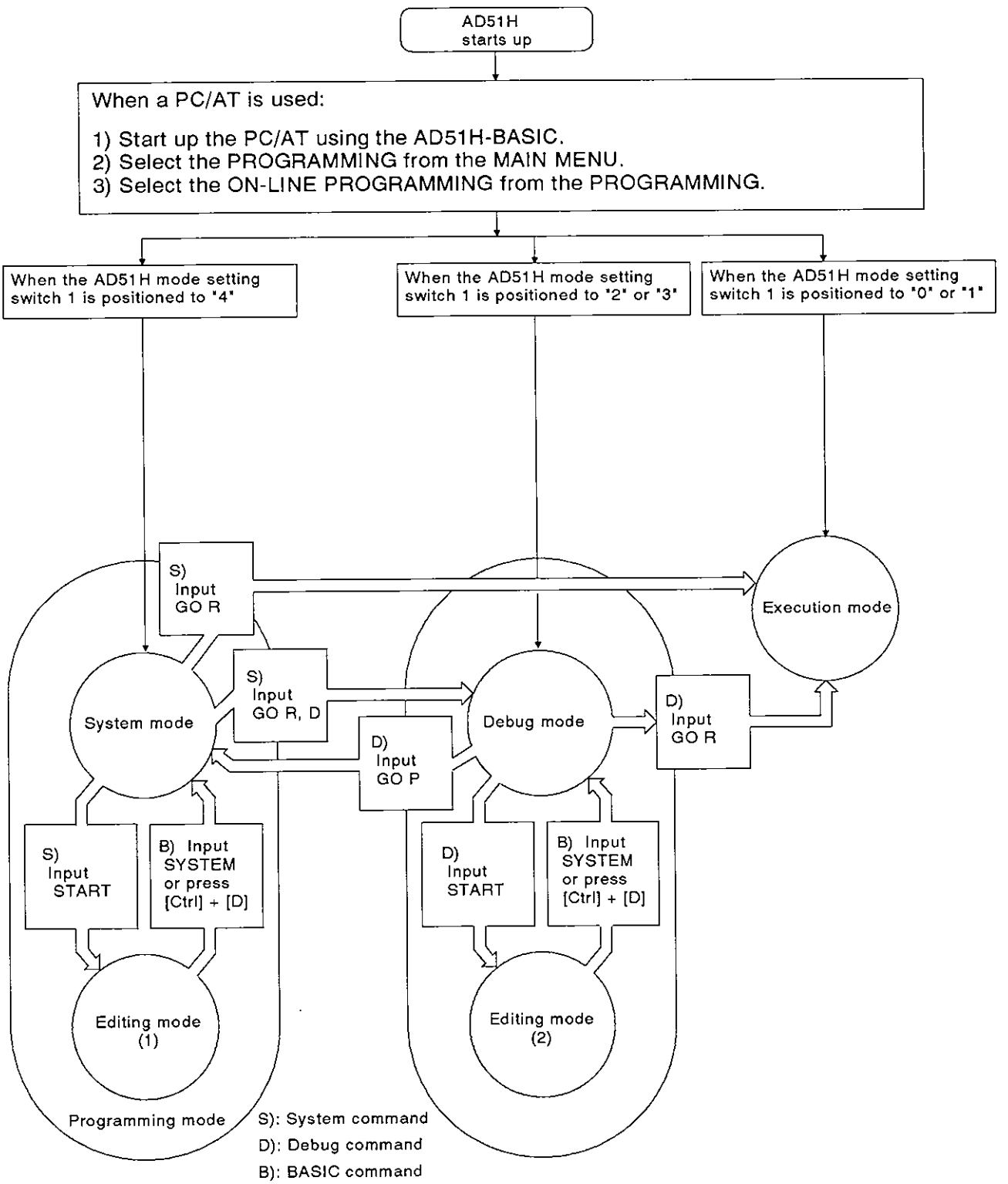
The AD51H Programming Manual gives how to edit and debug a BASIC program.

3) When changing the AD51H mode and continuing the operation, see Section 2.4.

2.3 AD51H Mode and Mode Change

After the AD51H starts up, the mode can be changed giving the system command from the console (see Section 3) or giving the debug command from the debugger (see Section 4).

This section gives the outline of how to change the AD51H mode with the system command and the debug command.



(1) Programming modes

(a) In the modes, it is possible to edit/debug a BASIC program, write/read data to/from a memory card, and to set a multitask.

(b) The programming modes are divided into the system mode and the editing mode (1).

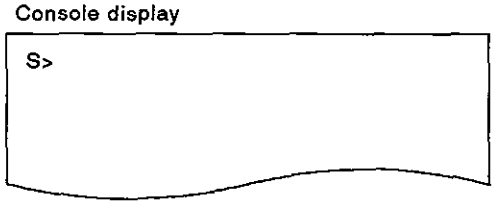
(2) System mode

(a) The AD51H enters this mode when the AD51H mode setting switch is positioned to "4" or when the GO command (GO P) is input from the debugger in the debug mode.

(b) The console is controlled by the operating system in the AD51H.

(c) By giving the system command, the following operations for each BASIC program can be executed:

Console display



- Writes/reads a BASIC program to/from the execution program area of the memory card (installed to MEMORY CARD [1]).
- Sets the multitask.

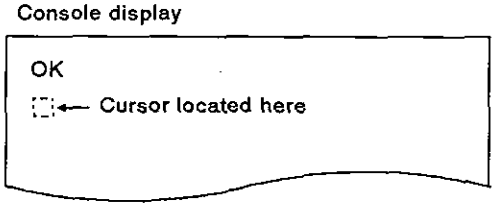
(3) Editing mode (1)

(a) The AD51H enters this mode when the START command is input using the console (in the system mode).

(b) The interpreter (an operating system to analyze and execute BASIC commands) uses the console.

(c) By giving a AD51H-BASIC command or function, the following operations for each BASIC program.

Console display



- Editing and debugging
- Writes/reads a BASIC program to/from the file storage area of a memory card.

(4) Execution mode

(a) The AD51H enters this mode when the AD51H mode setting switch 1 is positioned to "0" or "1" or when the GO command is input from the console or debugger.

(The AD51H enters this mode when the RUN keyswitch is positioned to RUN.)

(b) Set the multitask to execute several BASIC programs in the normal mode.

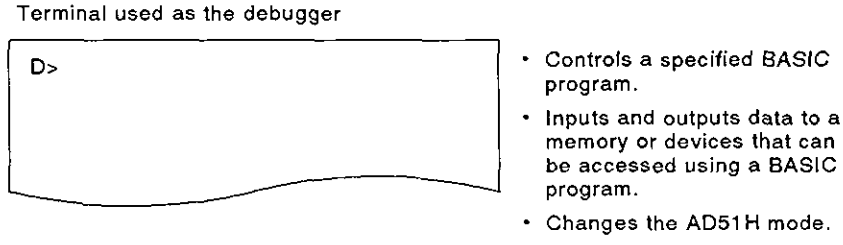
(5) Debug mode

(a) The AD51H enters this mode when the AD51H mode setting switch 1 is positioned to "2" or "3" or when the GO command is input using the console.

(The AD51H enters this mode when the RUN keyswitch is positioned to RUN.)

(b) The AD51H debug (an operating system to analyze and execute debug commands) uses the debugger.

(c) By giving a debug command input from the debugger, it is possible to debug each BASIC program in the multitask system.



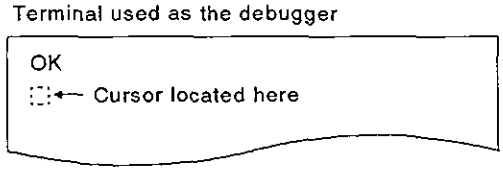
(6) Editing mode (2)

(a) The AD51H enters this mode when the START command is input from the debugger

(Another task than specified using the START command continues in the multitask system.)

(b) The debugger is controlled by the interpreter.

(c) By giving an AD51H-BASIC command or function, it is possible to correct a BASIC program when another BASIC program is being edited.



3. ONLINE PROGRAMMING

Online programming refers to such operations as editing/debugging the BASIC program, and writing/reading the BASIC program to a memory card, FD, or the PC/AT HD.

(Only the BASIC programs in a task can be debugged online.)

This section tells how to use system commands when (a) editing/debugging the BASIC program, and (b) writing/reading the BASIC program to/from a memory card, FD, or the PC/AT HD using the console in the system mode.

(1) Since most of this section concerns key inputting and displays on the console, the explanations assume that all key inputting and displays refer to the console.
When key inputting and displays refer to the debugger rather than the console, the word "debugger" is always used to avoid misunderstanding.

(2) Executing the operations discussed in this section requires the following preparations (see Section 2 for details):

- Setting the AD51H switch to program online ..... See Section 2
- Connecting the console ..... See Section 2

3.1 System Commands

Table 3.1 lists the system commands (input from the keyboard to the console) used for online programming.

Table 3.1 List of System Commands

Classification	System Command	Function	Reference Section
Memory Card Control	CCOPY	Makes a copy of data from a memory card, and writes the copy to another memory card. (Sets a backup memory card.)	3.2.1
	CFORMAT	Formats a memory card (physical format).	3.2.2
	CFORMAT?	Displays information about memory card formatting.	3.2.2
Execution Program Information Control	MLOAD	Reads data from a designated BASIC task area of the memory card using the corresponding BASIC task area in the AD51H.	3.3.1
	MSAVE	Writes data in a designated BASIC task area in the AD51H to the corresponding BASIC task area (multitask setting is automatically done).	3.3.2
Multitask Setting Control	SET	Changes the multitask setting description.	3.4.1
	SET?	Displays the multitask setting description.	3.4.2
Mode Control	START	Switches the AD51H from the system mode to the editing mode (1). (For editing/debugging each program)	3.5.1
	GO	Switches the AD51H from the system mode to the execution mode (2) or debug mode.	3.5.2
Interpreter Operation Control	TKILL	Stops the interpreter operation in a designated BASIC task area in the AD51H.	3.6
Others	EXIT	Displays the MAIN MENU screen on the console.	3.7
	HELP	Displays a list of system commands, descriptions of functions, and command input formats.	3.8

3.2 Copying/Deleting Data From a Memory Card

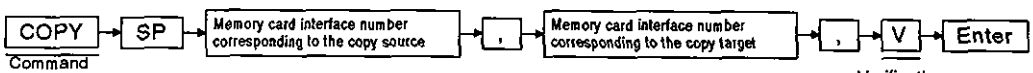
This section tells how to use system commands to copy/delete data from a memory card.

3.2.1 Copying data from a memory card and writing that data to another memory card (CCOPY command)

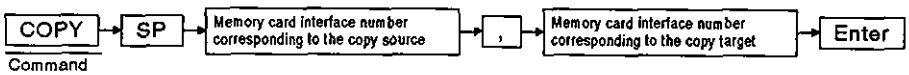
This operation backs up data by writing it to another memory card.

INPUT PROCEDURE (This command is also referred to as "CC")

To use the copy source to verify that correct copying has been done:



When only making a copy:

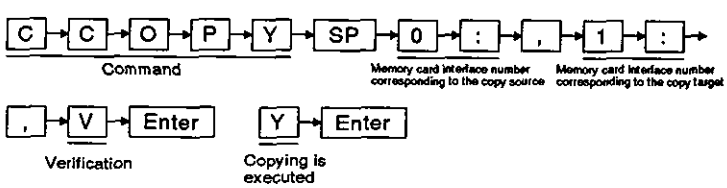


OPERATION EXAMPLE

Writes a copy of data from the memory card installed in MEMORY CARD1 to the memory card installed in MEMORY CARD2, and verifies that copying has been done correctly.

Before the command is input

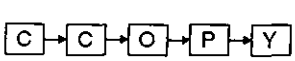
```
S>
```



After the command is input

```
S>CCOPY 0:, 1:, V
COPY(Y/N)?Y
COPY OK
S>
```

OPERATING PROCEDURE




```
S>CCOPY
```

1 Input the CCOPY command to write a copy of data in one memory card to another memory card.

(1) Precautions when using the CCOPY command
- Before copying, use the CFORMAT command to format the memory card in the copy target drive.
- Make sure that the copy source memory card capacity ≤ copy target memory card capacity.

SP → 0 → : → , → 1 → :

```
S>CCOPY 0:, 1:
```

2 Input the copy source memory card interface number (accompanied by a colon), followed by the copy target memory card interface number (accompanied by a colon and a comma). The numbers that can be input are "0" and "1".

0: Corresponds to the AD51H MEMORY CARD1
1: Corresponds to the AD51H MEMORY CARD2

(This example assumes that a copy of data in the memory card installed in MEMORY CARD1 is written to the memory card installed in MEMORY CARD2.)

, → V → Enter

```
S>CCOPY 0:, 1:, V
```

3 Designate "V" to verify that data has been correctly copied from the copy source to the copy target.

If verification is not executed, press the [Enter] key.

(This example assumes that verification is executed.)

Y → Enter

```
S>CCOPY 0:, 1:, V
COPY(Y/N)?Y
```

4 The "COPY (Y/N)?" dialog box appears.

Press the [Y] key to execute copying.

Press the [N] key to cancel the copy operation.

(The console remains in a wait state until either key is pressed.)

5 The next line shows the result of the execution.

```
S>CCOPY 0:, 1:, V
COPY(Y/)?Y
COPY OK
S>
```

When copying is executed normally, the screen shows "COPY OK".

If copying is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example assumes that copying is executed normally.)

6 "S>" appears after the execution result is displayed.

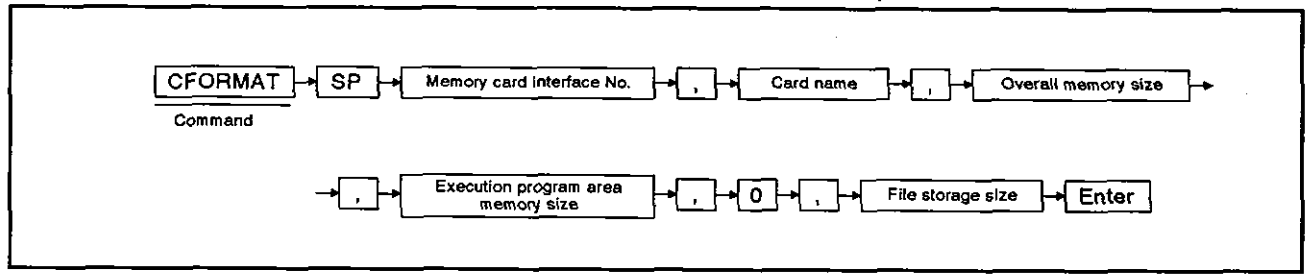
Input the necessary command.

(2) References
- Formatting a memory card: CFORMAT command ..... (see Section 3.2.2)
- Displaying memory card format information: CFORMAT? command .... (see Section 3.2.3)

3.2.2 Formatting a memory card (CFORMAT command)

This operation formats a memory card installed in MEMORY CARD1 or MEMORY CARD2.

INPUT PROCEDURE (This command is also referred to as "CF")



OPERATION EXAMPLE

Format a 512K byte memory card installed in MEMORY CARD1 as shown below:

Before the command is input

```
S>
```

After the command is input

```
S>CFORMAT 0:, "TASK-DTM", 32, 6, 0, 26
FORMAT(Y/N)?Y
FORMAT OK
S>
```

C F O R M A T SP 0 : ,
Command
Memory card interface No.

" T A S K - D T M " ,
Card name

..... Memory card is named

8 ,
Overall memory size
..... Sets the memory card capacity (for 8 x 64K bytes)

6 ,
Execution area memory
..... Sets the capacity of the execution program area (for 6 x 64K bytes) in the memory card

0 ,

2 Enter
File storage size
..... Sets the capacity of the file storage area (for 2 x 64K bytes) in the memory card

Y Enter
Formatting is executed

OPERATING PROCEDURE

C F O R M A T

1 Input the CFORMAT command to format a memory card.

```
S>CFORMAT
```

(1) Precautions when using the CFORMAT command
- Formatting a memory card deletes all data in that memory card.
- Release the write-protect tab when formatting a memory card which has a write-protect tab.
- Turn OFF the memory protect keyswitch on the AD51H when the memory card is installed in MEMORY CARD1.

SP → 0 → : → ,

```
S>CFORMAT 0:,
```

2 Input the interface number (which corresponds to the memory card to be formatted) accompanied by a colon.

The numbers that can be input are "0" and "1".

0: Corresponds to the AD51H MEMORY CARD1
1: Corresponds to the AD51H MEMORY CARD2

This example assumes that the memory card installed in MEMORY CARD1 will be formatted.

" → T → A → S → K → - →
D → T → M → " → ,

```
S>CFORMAT 0:, "TASK-DTM",
```

3 Input a name for the memory card using up to 16 alphanumeric characters and symbols.

This name must start with an alphabetic character. The name must be in quotation marks.

This example assumes that the memory card is named "TASK-DTM".

8 → ,

```
S>CFORMAT 0:, "TASK-DTM", 8
```

4 Designate the overall capacity of the memory card using a number equal to or greater than one (in units of 64K bytes).

Set the capacity which is consistent with the following formula:

Overall memory size (Overall capacity) = (Execution program area memory size + File storage size)

This example assumes that a 512K byte memory card is formatted.

(8 x 64K bytes → 512K bytes)

6 → ,

```
S>CFORMAT 0:, "TASK-DTM", 8, 6,
```

5 Designate the capacity of the execution program area in the memory card by using a number from 0 to 6 (in units of 64K bytes).

The maximum capacity of the execution area is 384K bytes. This area is allocated to the operating system area (128 bytes) and the BASIC task number area (where the execution programs is stored).

(This example assumes that the execution program area capacity is set to 384K bytes.)

0 → ,

```
S>CFORMAT 0:, "TASK-DTM", 8, 6, 0,
```

6 Input "0" as dummy data.

2 → Enter

```
S>CFORMAT 0:, "TASK-DTM", 8, 6, 0, 2
```

7 Designate the capacity of the file storage area in the memory card using a number equal to or greater than one (in units of 64K bytes).

This area stores BASIC programs (that are not stored in the BASIC task area) and data files.

(This example assumes that the file storage capacity is set to 128K bytes.)

(2 x 64 bytes → 128K bytes)

Y → Enter

```
S>CFORMAT 0:, "TASK-DTM", 8, 6, 0, 2
FORMAT(Y/N)?Y
```

8 The "FORMAT (Y/N)?" dialog box appears.

Press the [Y] key to execute formatting.

Press the [N] key to cancel the format operation.

(The console remains in a wait state until either key is pressed.)

(This example assumes that formatting is executed.)

9 The next line shows the result of the execution.

When formatting is executed normally, the screen shows "FORMAT OK".

If formatting is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example assumes that formatting is executed normally.)

```
S>CFORMAT 0:, "TASK-DTM", 8, 6, 0, 2
FORMAT(Y/N)?Y
FORMAT OK
S>
```

10 "S>" appears after the execution result is displayed.

Input the necessary command.

---

(2) Precautions when using the CFORMAT command to Designate capacities
- The overall memory size (overall capacity) must be consistent with the memory card to be formatted.
  The overall memory size must be equal to the execution program area memory size plus the file storage size.
- In the execution program area of a memory card, if the entire BASIC task number area is divided into eight parts, the maximum capacity of one part is approximately 48K bytes.
- The size can be designated in decimal, hexadecimal ("&H[ ][ ][ ][ ]"), or binary ("&B[ ][ ] to [ ]").

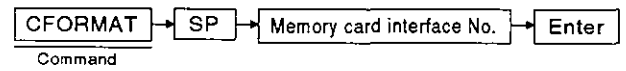
(3) Logically formatting a memory card
- The execution area is logically formatted when the SET or MSAVE command is initially used.
- Logically format the file storage area using the BASIC FORMAT command.

(4) Reference
- Displaying memory card format information...CFORMAT? command (see Section 3.2.3)

3.2.3 Displaying memory card format information (CFORMAT? command)

This operation displays memory card formatting information installed in MEMORY CARD1 or MEMORY CARD2 of the AD51H.

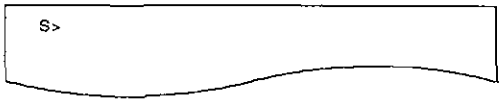
INPUT PROCEDURE (This command is also referred to as "CF?")

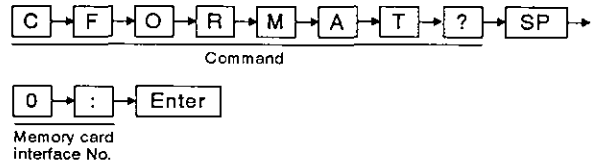


OPERATION EXAMPLE

Displays format information from the memory card installed in MEMORY CARD1.

Before the command is input

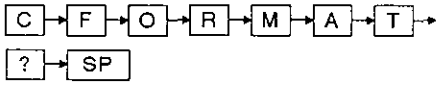




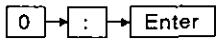
After the command is input

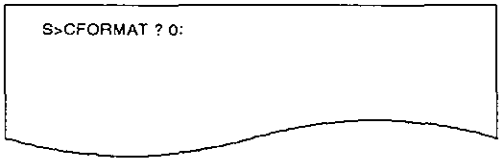
```
S>CFORMAT? 0:
     Card      Name  :  "TASK-DTM"
     Card      Size  :  512K bytes (8)
     Program   Size  :  384K bytes (6)
     Canvus    Size  :  0K byte (0)
     File      Size  :  128K bytes (2)
```

OPERATING PROCEDURE



1 Input the CFORMAT? command to display memory card format information.





2 Input the interface number (corresponding to the memory card whose format information is displayed) accompanied by a colon.

The numbers that can be input are "0" and "1".

0: Corresponds to the AD51H MEMORY CARD1
1: Corresponds to the AD51H MEMORY CARD2

If the [Enter] key is pressed without being designated, the result is the same as if "0" were designated.

(This example assumes that format information from the memory card installed in MEMORY CARD1 is displayed.)

```
S>CFORMAT? 0:
     Card      Name  :  "TASK-DTM"
     Card      Size  :  512K bytes (8)
     Program   Size  :  384K bytes (6)
     Canvus    Size  :  0K byte (0)
     File      Size  :  128K bytes (2)
```

3 Displays the command execution result.

When formatting is executed normally, the next lines show format information from the designated memory card.

If formatting is not executed normally, an error message appears (see Section 3.9).

When formatting is executed normally, the display on the left is shown.

(1)	Card Name	.....	Name of the memory card that is formatted.
(2)	Card Size	.....	Corresponds to the capacity of the whole memory card designated when the memory card was formatted. ( ) indicates the entire memory size designation when the memory card is formatted using the CFORMAT command.
(3)	Program Size	.....	Corresponds to the capacity of the execution program area when the memory card is formatted. ( ) indicates the capacity designation when the memory card is formatted using the CFORMAT command.
(4)	Canvas Size	.....	Should be ignored.
(5)	File Size	.....	Corresponds to the capacity of the file storage area designated when the memory card was formatted. ( ) indicates the capacity designation when the memory card was formatted using the CFORMAT command.

4 "S>" appears after the execution result is displayed.

Input the necessary command.

(1) Reference
- Formatting a memory card .....CFORMAT command (see Section 3.2.2)

3.3 Writing/Reading an Execution Program

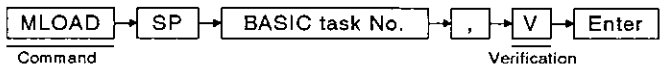
This section tells how to (a) write an execution program (stored in an AD51H BASIC task number area) to a memory card, and (b) read an execution program from the memory card using a BASIC task number area. Both are done by using system commands (to control execution program information).

3.3.1 Reading an execution program stored in a memory card using the AD51H (MLOAD command)

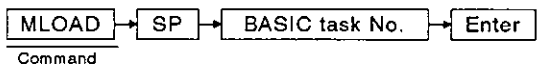
This operation reads an execution program stored in a memory card using the corresponding BASIC task number area in the AD51H.

INPUT PROCEDURE (This command is also referred to as "ML")

To verify the read data with the source data after reading the execution program:



When only reading:



OPERATION EXAMPLE

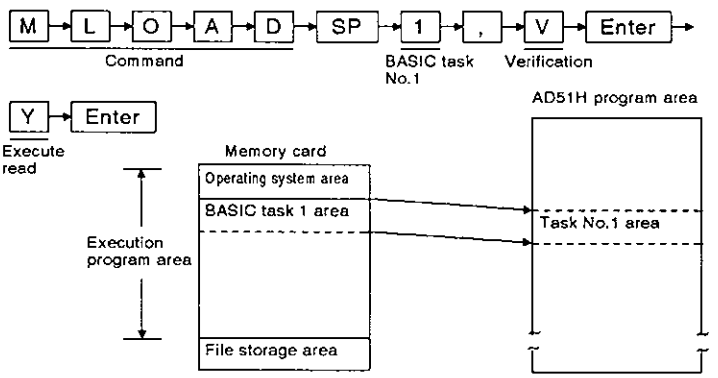
Reads an execution program (stored in the BASIC task No.1 area of a memory card) using the BASIC task No.1 area, and verifies the read program with the source program.

Before the command is input

```
S>
```

After the command is input

```
S>MLOAD 1, V
LOAD(Y/N)?Y
LOAD OK
S>
```



(1) Memory card location
- To use the MLOAD command for a memory card, the memory card must be installed in AD51H MEMORY CARD1.

OPERATING PROCEDURE

M → L → O → A → D → SP

```
S>MLOAD
```

1 Input the MLOAD command to read the execution program (stored in the memory card) using the AD51H execution program area.

1

```
S>MLOAD 1
```

2 Designate the BASIC task area (1 to 8) of the program stored in the execution program area.

(This example assumes that the BASIC task 1 program area is read.)

, → V → Enter

```
S>MLOAD 1,V
```

3 Designate "1" to verify the read program with the source program after reading the program.

Press the [Enter] key to read without doing verification.

(This example assumes that the read program is verified with the source program.)

(2) Precautions when using the MLOAD command
- The BASIC task area (designated by using the MSAVE or SET commands) must be as large as the AD51H BASIC task area (designated using the START command).
- Make sure that the interpreter is not working in the AD51H BASIC task area (where the execution program stored in a memory card is written).
  When the interpreter is working, stop the interpreter operation by using the TKILL command.

Y → Enter

```
S>MLOAD 1,V
LOAD(Y/N)?Y
```

4 The "LOAD (Y/N)?" dialog box appears.

Press the [Y] key to execute reading.

Press the [N] key to cancel the read operation.

(The console remains in a wait state until either key is pressed.)

(This example assumes that reading is executed.)

5 The next line shows the result of the execution.

```
S>MLOAD 1,V
LOAD(Y/N)?Y
LOAD OK
S>
```

When reading is executed normally, the screen shows "LOAD OK".

If reading is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example assumes that reading is executed normally.)

6 "S>" appears after the execution result is displayed.

Input the necessary command.

(2) References

- Writing an execution program (stored in the AD51H) to a memory card .....MSAVE command (see Section 3.3.2)
- Changing the multitask setting.....SET command (see Section 3.4.1)
- Displaying the multitask setting description.....SET? command (see Section 3.4.2)
- Setting the AD51H to the editing mode (1) .....START command (see Section 3.5.1)
- Stops the interpreter operation in a specific BASIC task area .....TKILL command (see Section 3.6)

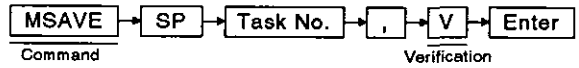
3.3.2 Writing an execution program (stored in the AD51H) to a memory card (MSAVE command)

This operation writes an execution program (stored in a designated BASIC task area of the AD51H) to the corresponding BASIC task area of a memory card.

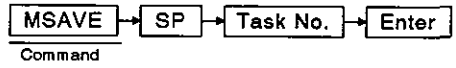
Executing this operation automatically sets multitasking in the designated task area.

INPUT PROCEDURE (This command is also referred to as "MS")

To verify the written program with the source program:



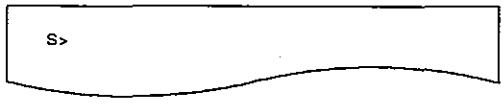
Only to write:



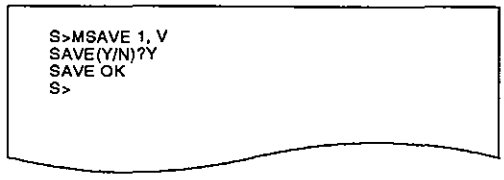
OPERATION EXAMPLE

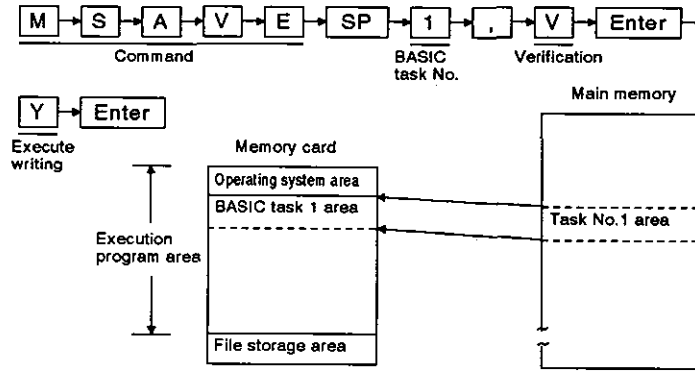
Writes an execution program (stored in the BASIC task 1 area) to the BASIC task 1 area in a memory card, and verifies the written program with the source one.

Before the command is input



After the command is input





(1) Memory card location
- To use the MSAVE command with a memory card, the memory card must be installed in MEMORY CARD1 of the AD51H.

(2) Precautions when using the MSAVE command
- Use the START command to start the interpreter. Then (a) execute the SYSTEM command to the interpreter, or (b) press the [Ctrl] + [D] keys, and execute the MSAVE command.
- Since writing another execution program to the BASIC task area (to which an execution program has already been written) can cause an overwrite in the BASIC task area, take the following steps:
  (a) Write all execution programs to the execution area of the memory card.
  (b) Reset multitasking when appropriate.

OPERATING PROCEDURE

M → S → A → V → E → SP

```
S>MSAVE
```

1 Input the MSAVE command to write an execution program stored in the AD51H to the memory card.

1

```
S>MSAVE 1
```

2 Designate the AD51H BASIC task area (1 to 8).

(This example assumes that the execution program is stored in the BASIC task 1 area.)

, → V → Enter

```
S>MSAVE 1,V
```

3 Designate "V" to verify the written program with the source program after writing the program.

Press the [Enter] key to read without verification.

(This example assumes that the written program has been verified with the original.)

(3) Precautions when using the MSAVE command

- Data for the AD51H BASIC task area is written as an execution program to the corresponding BASIC task area of a memory card.
- After writing the execution program, multitasking is automatically set in the designated BASIC task area. The setting description is shown below:
  The SET command explanation in this manual gives details.

  Starting condition ..... "BOOT" attribute is set.
  Size ..... Designated task size when the START command is executed is set.
  Execution sequence ..... Nothing is set.

Y → Enter

```
S>MSAVE 1, V
SAVE(Y/N)?Y
```

4 The "SAVE (Y/N)?" dialog box appears.

Press the [Y] key to execute writing.

Press the [N] key to cancel the write operation.

(The console remains in a wait state until either key is pressed.)

(This example assumes that writing is executed.)

5 The next line shows the result of the execution.

```
S>MSAVE 1, V
SAVE(Y/N)?Y
SAVE OK
S>
```

When writing is executed normally, the screen shows "SAVE OK".

If writing is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example assumes that the MSAVE command is executed normally.)

6 "S>" appears after the execution result is displayed.

Input the necessary command.

(4) References

- Reading an execution program (stored in a memory card) using the main memory .....MLOAD command (see Section 3.3.1)
- Changing the multitask setting.....SET command (see Section 3.4.1)
- Displaying the multitask setting description.....SET? command (see Section 3.4.2)
- Setting the AD51H to the editing mode (1).....START command (see Section 3.5.1)

3.4 Setting/Changing/Displaying Multitasking Descriptions

This section tells how to set/change/display multitasking descriptions using system commands to control the multitask setting.

The multitask setting is used to set the starting conditions when starting up the AD51H and executing several programs during multitasking.

The multitask setting contains the following items designated by using the MSAVE or SET commands:

(1) Starting condition

Sets the condition when the BASIC program is stored in the BASIC task area.

(a) START

- After power to the AD51H is turned ON or the AD51H is reset, an execution program (stored in the memory card execution program area) is read using the AD51H execution program area, and program execution is started.

(b) BOOT

- When the AD51H is started up, an execution program stored in the memory card BASIC task area is read by using the AD51H execution program area.
- If the program being executed gives a ZSTART command, then the designated BASIC program will start.

(c) IT

- When the AD51H is started up, an execution program stored in the memory card BASIC task area is read by using the AD51H execution program area.
- If a PC CPU turns ON an output signal (such as the start task number designation flag or task start signal) for the AD51H, then the designated BASIC program will start.

(d) ON

- After starting up the AD51H, if the BASIC program being executed gives a ZSTART command, then the designated program will be read from the memory card file storage area and the execution will start.

(e) OFF

- Invalidates the multitask setting for a task area.

  If this is executed for a task area, then the BASIC program cannot be executed in that area.

(2) Task size

Designates the size of the BASIC task area (16K, 32K, 48K, or 64K bytes).

(3) Execution sequence

After the AD51H starts up, if several programs are installed in the corresponding task areas and those programs are executed, designate which program will be executed first.

If an execution program is written to the memory card execution program area used for multitasking, then the multitasking is automatically set in that area.

This section tells how to set/change/confirm the multitask setting.

(1) Changing the task size in the multitask setting
- To change the task size in the multitask setting in order to enlarge the corresponding task area, take the following steps:
  (a) Write all execution programs to the execution area of the memory card.
  (b) Reset each task size so that all execution programs (max.8) can be written using the appropriate area designated when the memory card was formatted.

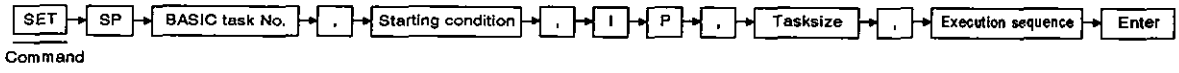
(2) Section 3.3.2 gives details about the MSAVE command.

3.4.1 Setting/Changing the multitask (SET command)

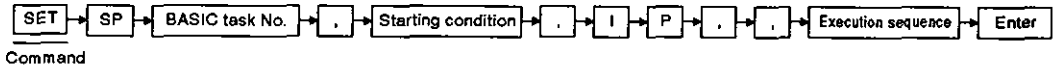
This operation sets multitasking for a task area and changes the multitask setting.

INPUT PROCEDURE (This command is also referred to as "S")

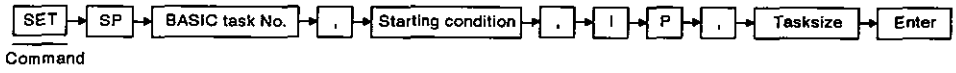
To set/change the starting condition, size and execution sequence:



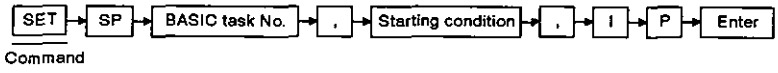
To change the starting condition and execution sequence:



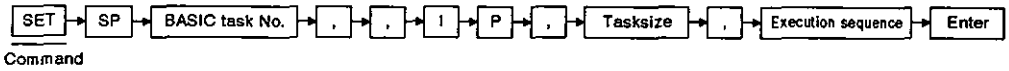
To change the starting condition and size:



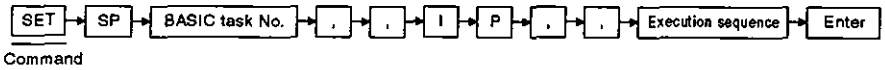
To change the starting condition:



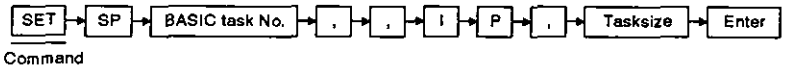
To change the size and execution sequence:



To change the execution sequence:



To change the size:



OPERATION EXAMPLE

Sets multitasking for the BASIC task 1 area.

Before the command is input

```
S>
```

S → E → T → SP → 1 → , → S → T → A → R → T

Command: S E T SP | Task No.: 1 | Starting condition: , S T A R T

After the command is input

```
S>SET 1, START, IP, 48, 2
SET OK
S>
```

, → I → P

, → 4 → 8 (Task size)

, → 2 → Enter (Execution sequence)

OPERATING PROCEDURE

S → E → T → SP

```
S>SET
```

1 Input the SET command to set/change multitasking.

●

```
S>SET 1
```

2 Designate the BASIC task area (1 to 8) for which multitasking is set.

(This example assumes that multitasking is set/changed for the AD51H BASIC task 1 area.)

, → S → T → A → R → T

```
S>SET 1, START
```

3 To set/change the starting condition for the AD51H BASIC task area, select one of the following items:

1) START
2) BOOT
3) IT
4) ON
5) OFF

If the starting condition is not designated, input a comma (,). In this case, the previously designated starting condition will be used.

(This example assumes that START is selected.)

---

(1) Section 3.4 (Programming and Debugging) and Section 6.4 (Control Function) give details about starting conditions.

, → I → P

```
S>SET 1, START, IP
```

4 Designate "IP" as the type of execution program.

, → 4 → 8

```
S>SET 1, START, IP, 48
```

5 Input "16", "32", "48", or "64" to set/change the corresponding BASIC task area.

If the task size is not designated, input a comma (,).

In this case, the previously designated task size will be used.

(This example assumes that the task size is set to 48K bytes.)

, → 2 → Enter

```
S>SET 1, START, IP, 48, 2
```

6 When the AD51H is started, input a number from 1 to 8 to set/change the execution sequence of the multitask programs for which "START" is designated. (When "1" is designated, the corresponding program is given the highest priority.)

If several task areas have the same number, the tasks stored in those areas are processing in ascending order of the task numbers.

When the execution sequence is not designated, press the [Enter] key.

In this case, the previously designated execution sequence is used.

(This example assumes that "2" is designated for the execution sequence.)

7 The next line shows the result of the execution.

```
S>SET 1, START, IP, 48, 2
SET OK
S>
```

When the SET command is executed normally, the screen shows "SET OK". If the SET command is not executed normally, an error message appears.

See Section 3.9, and correct the error. (This example assumes that the SET command is executed normally.)

8 "S>" appears after the execution result is displayed.

Input the necessary command.

---

(2) Size designation
- The size can be designated in decimal, hexadecimal ("&H[ ][ ][ ][ ]"), or binary ("&B[ ][ ] to [ ]").

(3) References
- Writing BASIC task area information stored in the AD51H to a memory card ..... MSAVE command (see Section 3.3.2)
- Displaying the multitask setting description ..... SET? command (see Section 3.4.2)
- Setting the AD51H to editing mode (1) ..... START command (see Section 3.5.1)

3.4.2 Displaying the multitask setting description (SET? command)

This operation displays the multitask setting description for each BASIC task area in the AD51H.

INPUT PROCEDURE (This command is also referred to as "S?")

To designate one of the BASIC task areas:

SET? → SP → BASIC task No. → Enter
(SET?: Command)

To designate all BASIC task areas:

SET? → Enter
(SET?: Command)

OPERATION EXAMPLE

Writes an execution program (stored in the BASIC task 1 area) to the BASIC task 1 area in a memory card, and verifies the written program with the source one.

Before the command is input

```
S>
```

S → E → T → ? → SP → 1 → Enter
(S→E→T→?: Command; 1: BASIC task No.)

After the command is input

```
S>SET? 1
          Task No.  Type  Size  Start  Condition  Start No.
              1       IP    48          START         2
S>
```

OPERATING PROCEDURE

S → E → T → ?

```
S>SET?
```

1 Input the SET? command to display the multitask setting description.

SP → 1 → Enter

```
S>SET? 1
```

2 Designate the BASIC task area (1 to 8) whose setting description will be displayed.

When all areas are designated, press the [Enter] key.

(This example assumes that "1" is designated.)

3 When the SET? command is executed normally, the screen shows information about the designated task area.

```
S>SET? 1
          Task No.  Type  Size  Start  Condition  Start No.
                1    IP    48          START          2
S>
```

If the SET? command is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example shows information about the BASIC task 1 area.)

To read the displayed description, see the SET? command explanation

(1) Task No. ..... Indicates the corresponding task number.

(2) Type ..... Corresponds to the IP that was input right after the start condition by using the SET? command.

(3) Size ..... Indicates the corresponding task area size.
Corresponds to the task size set by the SET? command.

(4) Start Condition ..... Used to execute the BASIC program in a designated area.
Corresponds to the start condition set by the SET? command.

(5) Start No. ..... Indicates the execution sequence when the start condition is set to the START command.
Corresponds to the execution sequence set by the SET? command.
When the start condition is not "START", "–" will be displayed at the position corresponding to this item since it is ignored.

4 "S>" appears after the execution result is displayed.

Input the necessary command.

(2) References
- Writing the BASIC task area information (stored in the AD51H) to a memory card ..... MSAVE command (see Section 3.3.2)
- Setting/changing the multitask description..... SET command (see Section 3.4.1)
- Setting the AD51H to the editing mode (1) ..... START command (see Section 3.5.1)

3.5 Changing the AD51H Mode

This section tells how to use the system commands (used to control the mode) to change the AD51H mode.

3.5.1 Setting the AD51H to the editing mode (1) (START command)

This operation edits/debugs a program.

INPUT PROCEDURE (This command is also referred to as "ST")

To set/change the task area size:

START → SP → BASIC task No. → , → Task size → Enter
(START: Command)

To not change the task area size:

START → SP → BASIC task No. → Enter
(START: Command)

OPERATION EXAMPLE 1

Starts editing a new BASIC program in the BASIC task 1 area.
Sets the task 1 area to 48K bytes.

Before the command is input

```
S>
```

S → T → A → R → T → SP → 1 → , → 4 → 8 → Enter
(START: Command; 1: Task No.; 48: Task size)

After the command is input

```
S>START 1, 48
```

```
OK
[ ]
```
Cursor located here

If the interpreter was not operating during the execution, the following message appears before "OK":
"AD51H-BASIC ON-LINE PROGRAMMING Ver [ ]"

OPERATION EXAMPLE 2

Starts editing an existing BASIC program in the BASIC task 1 area.
Changes the task 1 area size to 64K bytes.

Before the command is input

```
S>TKILL 1:
KILL OK
S>
```

S → T → A → R → T → SP → 1 → , → 6 → 4 → Enter
(START: Command; 1: Task No.; 64: Task size)

After the command is input

```
S>TKILL 1:
KILL OK
S>START 1, 64
```

```
OK
```

If the interpreter was not operating during the execution, the following message appears before "OK":
"AD51H-BASIC ON-LINE PROGRAMMING Ver [ ]"

OPERATING PROCEDURE

S → T → A → R → T → SP

```
S>START
```

1 Input the START command to set the AD51H to the editing mode (1).

●

```
S>START 1
```

2 Designate the task area in which the BASIC program to be edited/debugged will be stored, selecting from 1 to 8.

The task number can be omitted.

Omitting the task number is regarded as doing the following designation:

1) When the START command is initially input, the designation is "1".
2) If the START command was already used, the previous task number is used.

(This example assumes that the BASIC program is edited/debugged in the task 1 area.)

, → 4 → 8 → Enter

```
S>START 1, 48
```

3 Input "16", "32", "48", or "64" to set/change the task area size (in units of 1K byte).

After completing editing/debugging, giving the MSAVE command (a) writes the data in the BASIC task area to the memory card, and (b) automatically sets the task size.

When designating a BASIC task area for which multitasking is not set, be sure to input the task size.

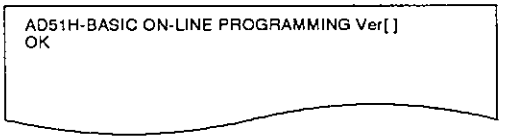
Even when designating a BASIC task area for which multitasking is already set, be sure to input the task size if the task size needs to be changed.

If the set task is not changed, just press the [Enter] key.

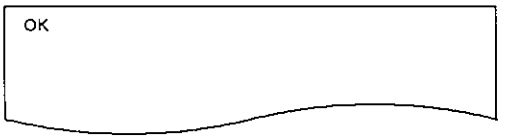
(This example assumes that the BASIC task 1 area size is set/changed to 48K bytes.)

---

(1) Size designation
- The size can be designated in decimal, hexadecimal ("&H[ ][ ][ ][ ]"), or binary ("&B[ ][ ] to [ ]").



or



4 The next line shows the result of the execution.

When the START command is executed normally, the screen enters the state shown on the left.

Start editing/debugging the program.

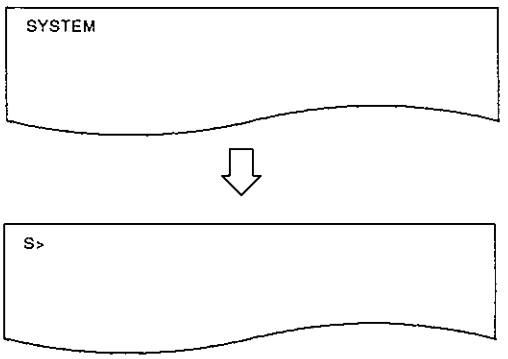
The AD51H-BASIC Programming Manual tells how to edit/debug the BASIC program.

If the START command is not executed normally, an error message appears.

See Section 3.9, and correct the error.

The example on the upper left shows the description displayed if the interpreter was not started before the execution.

The example on the lower left shows the description displayed when the interpreter was already started before the execution.



5 To return the AD51H from the edit mode (1) to the system mode after editing/debugging is completed, execute one of the following:

1) Execute the BASIC SYSTEM command.
   - Stops executing the BASIC program.
   - Closes the open file and the communications line.

2) Press the [Ctrl] + [D] keys.
   - Stops executing the BASIC program.
   - Leaves the open file and the communications line as they are.
   - Can resume (or continue) executing the program using the CONT command if the BASIC program was not changed when the AD51H was reset to the editing mode using the START command.

(2) Precautions when using the START command
- When the task size must be changed, use the TKILL command to stop operation of the interpreter in the utilized task area in order to edit/debug the BASIC program in the following task areas:
  1) A task area for which the multitask is set
  2) A task area that already contains the BASIC program

  In addition, when the task size is enlarged, take the following steps to write the BASIC program to the memory card execution area (using the MSAVE command) after editing/debugging a program:
  1) Write all execution programs to the execution area of the memory card.
  2) Reset multitasking so that all appropriate execution programs (max.8) can be written using the area designated when the memory card was formatted.

(3) Debugging the BASIC program after executing the START command
- In the editing mode (1), debug the program as shown in the programming manual.
- The debug commands shown in Section 4 cannot be used.

(4) References
- Writing the BASIC task area information stored in the main memory to a memory card ..... MSAVE command (see Section 3.3.2)
- Changing the AD51H mode ..... GO command (see Section 3.5.2)
- Stopping the interpreter operation in a designated task area ..... TKILL command (see Section 3.6)

3.5.2 Setting the AD51H to the execution/system mode (GO command)

This operation switches the AD51H between the system mode and the execution/debug/execution (2).

To give the command to debug the BASIC multitask program, set the AD51H to the debug mode. (See Section 4)

To start executing the BASIC program by setting multitasking, set the AD51H to the execution mode (2) to start

When the AD51H is returned to the system mode, giving the system command enables the BASIC program to be edited/debugged.

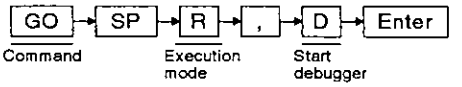
The following table gives the mode and debug start designation when the GO command is input and the state of the console and debugger after the GO command is executed.

<table>
<tr><th>Mode Designation</th><th>Debug Start Designation (YES/NO)</th><th>Console State</th><th>Debugger Terminal State</th><th>Remarks</th></tr>
<tr><td rowspan="2">R
(Execution mode (1))</td><td>YES
(To the debug mode)</td><td rowspan="2">Clears the displayed description.
Sets the terminal to be used for running the BASIC program.</td><td>Starts the debugger, clears the screen, and displays "D>" on the screen.
Enables the debug commands to be input.</td><td rowspan="2">Setting multitasking installs the BASIC program in the corresponding task area, starting the execution.</td></tr>
<tr><td>NO
(To the execution mode (2))</td><td rowspan="2">Displayed descriptions remain as they are.
Becomes a general-purpose port used for running the BASIC program.</td></tr>
<tr><td>P
(System mode during programming)</td><td>Cannot be designated.</td><td>Clears the displayed description, and displays "S>".
Enables the system commands to be input.</td><td>Stops executing BASIC programs in each task area.</td></tr>
</table>

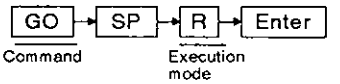
(1) BASIC program state when the GO command is executed
- Designating the execution mode (1) starts executing the BASIC program in the same way when the AD51H is started up by setting the AD51H mode switch (1) to "0", "1", or "3".
- Designating the programming mode stops the execution of all programs in task areas. In this case, since the task area memory state is not changed, all BASIC programs remain as they are.

INPUT PROCEDURE (No command abbreviation)

To set the AD51H to the debug mode:



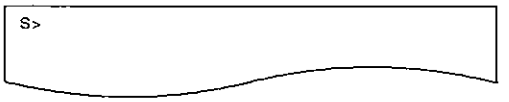
To set the AD51H to the execution mode:

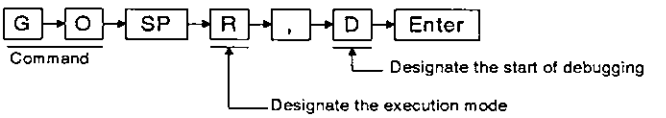


OPERATION EXAMPLE

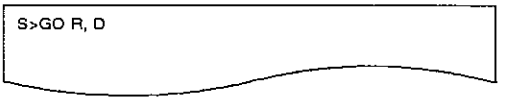
Sets the AD51H to the debug mode:

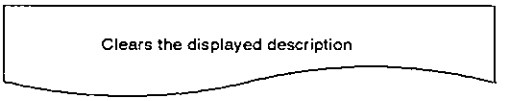
Before the command is input



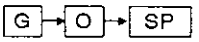


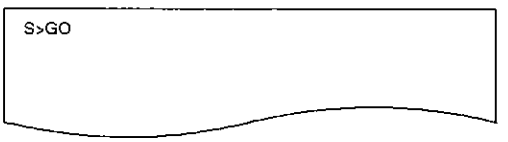
After the command is input



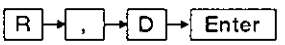


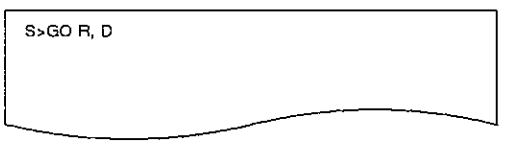
OPERATING PROCEDURE





1 Input the GO command to change the AD51H mode.





2 Designate the mode.

Input "R" to set the AD51H to the execution mode.

Input "R, D" to set the AD51H to the debug mode.

(This example assumes that the AD51H is set to the debug mode.)

3 The next line shows the result of the execution.

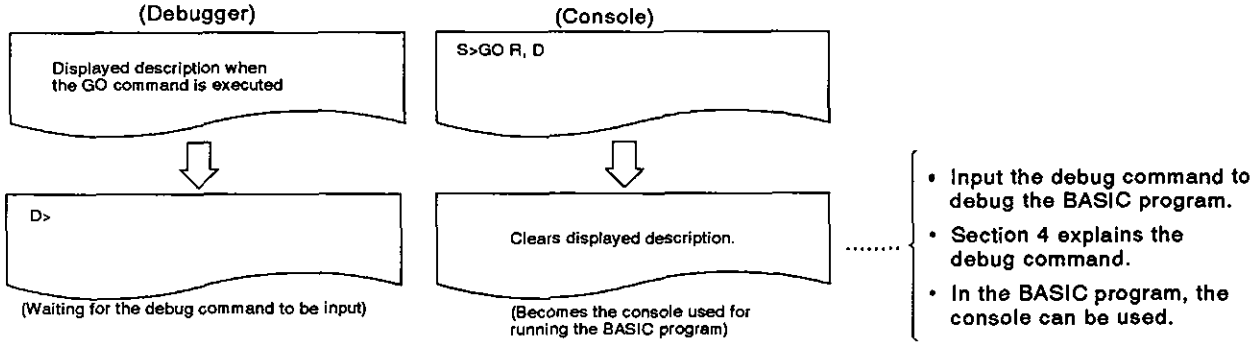
When the GO command is executed normally, the screen enters the state shown below:

If the GO command is not executed normally, an error message appears.

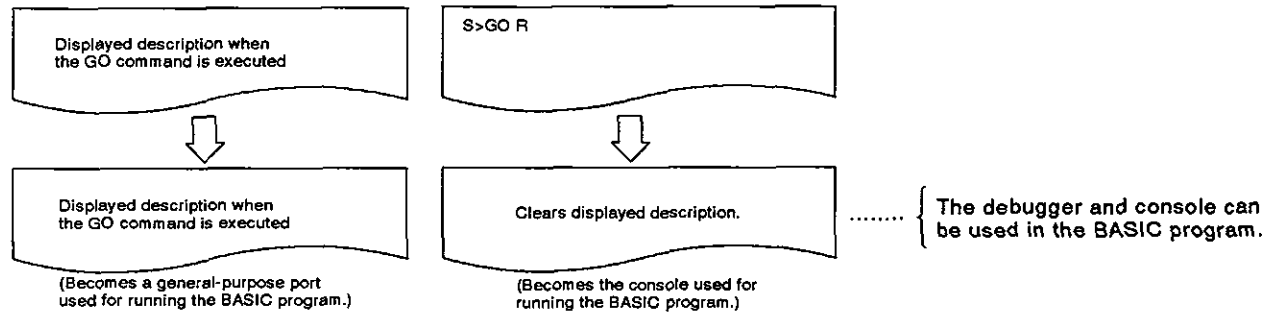
See Section 3.9, and correct the error.

The following example shows the displayed description when the GO command is executed normally:

1) When the debug mode is designated:



2) When the execution mode is designated:



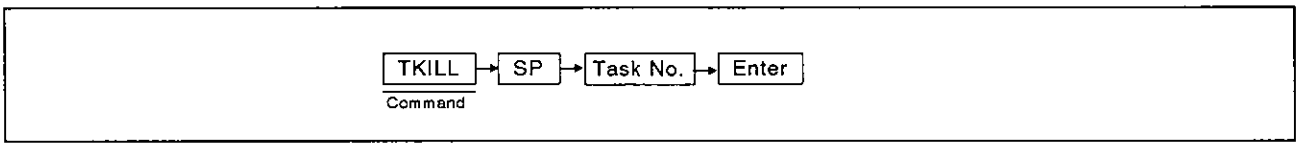
(2) AD51H mode change
See the AD51H mode change chart in Section 2.4.

(3) Reference
• Displaying the MAIN MENU on the console screen.....EXIT command (see Section 3.7)

3.6 Stopping the Interpreter Operation in a Designated Task Area

This section tells how to stop the interpreter operation in a designated task area using the system command (TKILL) to control the interpreter operation.

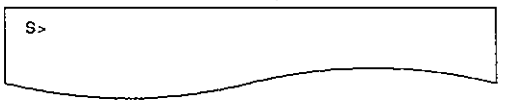
INPUT PROCEDURE (This command is also referred to as "TK")

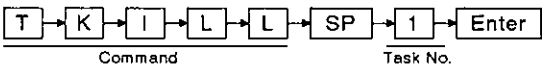


OPERATION EXAMPLE

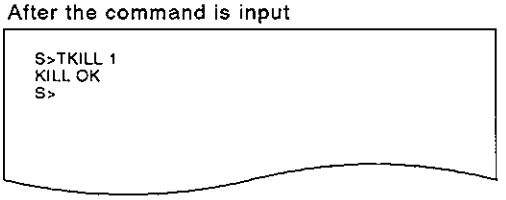
Stops the interpreter operation in the BASIC task 1 area.

Before the command is input

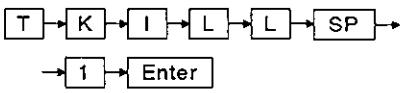


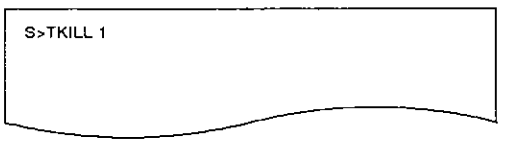


After the command is input



OPERATING PROCEDURE





1 To stop operating the interpreter, input the TKILL command accompanied by the corresponding task number (1 to 8).

(This example assumes that the interpreter operating in the task 1 area has been stopped.)

```
S>TKILL 1
KILL OK
S>
```

2 The next line shows the result of the execution.

When the TKILL command is executed normally, the screen shows "TKILL OK".

If the TKILL command is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example assumes that the TKILL command is executed normally.)

3 "S" appears after the execution result is displayed.

Input the necessary command.

(1) When the TKILL command should be used

To execute the following operations and stop the interpreter operation.

1) Setting the AD51H to the system mode and changing a task size using the START/SET command.

2) Setting the AD51H to the system mode and reading an execution program in a designated memory card BASIC task area using the corresponding task area in the AD51H.

(2) References

- Setting the AD51H to the editing mode (1) .....START command (see Section 3.5)
- Changing the AD51H mode .....GO command (see Section 3.5.2)

3.7 Displaying the MAIN MENU on the Console Screen

This section tells how use the EXIT command to display the MAIN MENU on the console screen when a PC/AT is used as the console.

When a VG-620 or a VG-382/VT-220 is used as the console, pressing any key redisplays "S>" after the EXIT command is input.

INPUT PROCEDURE (This command is also referred to as "E")

EXIT → Enter
Command

OPERATION EXAMPLE

Displays the MAIN MENU on the console screen.

Before the command is input

S>

E → X → I → T → Enter
Command

After the command is input

OPERATING PROCEDURE

E → X → I → T → Enter

1 Input the EXIT command to display the MAIN MENU.

S>EXIT

(1) BASIC program state when the EXIT command is executed
Even when the EXIT command is executed, BASIC programs in the BASIC task areas will be executed.

(2) Precautions when the command is input
To display the MAIN MENU to edit the BASIC program in a task area, stop the BASIC program before the EXIT command is input
(Execution of the BASIC program can influence the system control.)

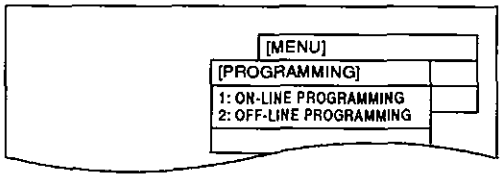
2 Do the corresponding operation after displaying the result of the execution.

When the display is executed normally, the console screen enters the following state:

If the display is not executed normally, an error message appears.

See Section 3.9, and correct the error.

(This example assumes that the display is executed normally.)



1) When a PC/AT is used as the console:

   The console screen displays the MAIN MENU.

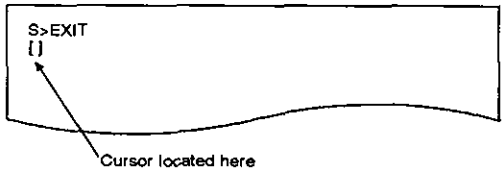
   Select an operation from the MAIN MENU.

   The SW0IX-AD51HPE AD51H-BASIC Operating Manual tells how to use the MAIN MENU.

2) When a VG-620 or a VT-382/VT-220 is used as the console:

   The console enters the state of waiting for key input.

   Press any key to display "S>", and input a system command.



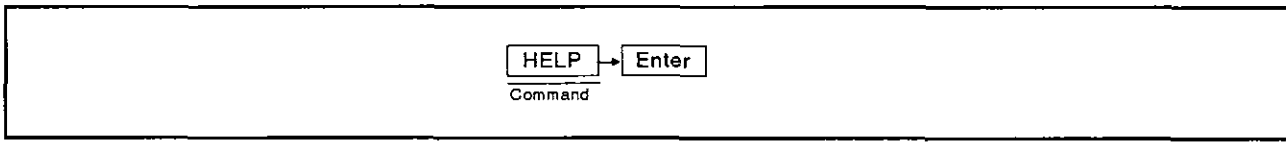
(2) AD51H mode changes
See the AD51H mode change chart in Section 2.4.

(3) Reference
Changing the AD51H mode ..... GO command (see Section 3.5.2)

3.8 Confirming the System Command Input Procedure

This section tells how to use the HELP command when displaying a command input procedure to confirm the input procedure.

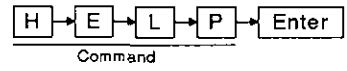
INPUT PROCEDURE (This command is also referred to as "H")



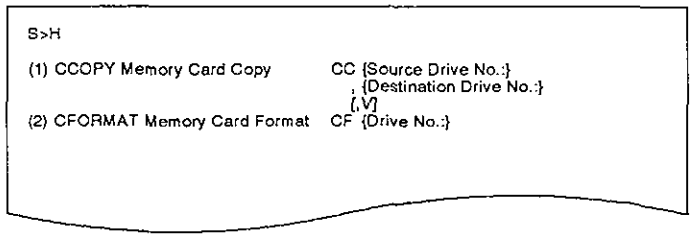
OPERATION EXAMPLE

Displays the system command input procedure.

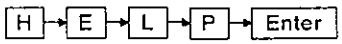
Before the command is input



After the command is input



OPERATING PROCEDURE



1 Input the HELP command to display the system command input procedure.

2 The result of the execution is displayed.

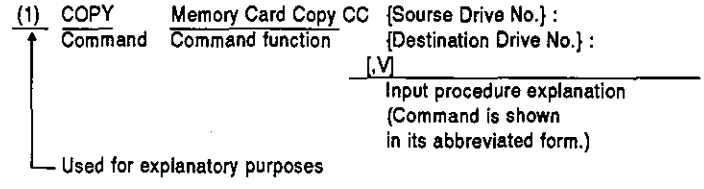
When the HELP command is executed normally, the subsequent lines show nine kinds of system command functions and the input procedure.

```
S>H
(1) CCOPY Memory Card Copy          CC {Source Drive No.:}
                                       , {Destination Drive No.:}
                                       [,V]
(2) CFORMAT Memory Card Format      CF {Drive No.:}
```

When three kinds of system command functions and the input procedure (as shown on the next page) are displayed, press any key other than the [ESC] key.

Pressing the [ESC] key stops the HELP function.

(Example)



If the HELP command is not executed normally, an error message appears.

See Section 3.9, and correct the error.

3 "S>" appears after the execution result is displayed.

Input the necessary command.

(1) Display description of a command input procedure
A space after the command requires pressing the [SP] key one time.
Since braces ("{" and "}") are only used to indicate the beginning and end of a command argument, they don't need to be actually input.
Since brackets ("[" and "]") are used only to indicate an 'omissible part", they don't need to be actually input.

4. MULTITASK DEBUGGING

Multitask debugging is used to (a) find an error in a program, and (b) correct that error during multitasking.

This section tells how to input a debug command from the debugger to debug all programs when BASIC programs are executed by multitasking.

(1) Since most of this section concerns key inputting and displays on the debugger, the explanations assume that all key inputting and displays refer to the debugger.
When key inputting and displays refer to the console rather than the debugger, the word "console" is always used to avoid misunderstanding.

(2) Executing the operations discussed in this section requires the following preparations:
- Setting the AD51H switch to program online ..... See Section 2
- Connecting the debugger ..... See Section 2
- Creating BASIC programs and debugging them .....
- Storing programs in memory .....See Section 3.3.2
- Setting multitasking .....See Section 3.4.1

(3) Precautions when inputting debug commands
If the debugging system (operating system to analyze a debug command and execute it) is not in a state to execute an input command, the execution of that command will be suspended until the debugging enters an executable state.
After displaying "D>" again, input the necessary command.

4.1 Debug Commands

Table 4.1 lists the debug commands used for multitask debugging.

Table 4.1 List of Debug Commands

Classification	Debug Command	Function		Reference Section
Task control	TSTATUS	Displays the state of BASIC programs in the designated task area.		4.2.1
	TRUN	Starts executing a BASIC program in the designated task area.		4.2.2
	TSTOP	Stops executing a BASIC program in the designated task area.		4.2.3
	TCONTINUE	Restarts execution of a stopped BASIC program in the designated task area.		4.2.4
	T?	Displays the value of a designated variable in the BASIC program (existing in the designated task area).		4.2.5
	TLET	Assigns a value to a designated variable in the BASIC program (existing in the designated task area).		4.2.6
Memory access control	MREAD	Displays the range of addresses that can be shared by BASIC programs.	• Buffer • Common memory • Internal devices (ED, EM)	4.3.1
	MWRITE	Writes a value to a designated address that can be shared by BASIC programs.		4.3.2
	B@	Displays internal device bit data that can be shared by BASIC programs.	• General-purpose input (X) • General-purpose output (Y) • Internal device (EM)	4.3.3
	B@	Writes bit data that can be shared by BASIC programs to an internal device.		4.3.4
	W@	Displays internal device word data that can be shared by BASIC programs.	• Internal device (ED)	4.3.5
	W@	Writes word data in that can be shared by BASIC programs to an internal device.		4.3.6
OS Information confirmation	ZSTATUS	Displays the operating states of the event, message port, and source (that can be shared by BASIC programs).		4.4
Mode control	START	Switch the AD51H from the debug mode to editing mode (2). (To edit a program during multitasking)		4.5.1
	GO	Switch the AD51H from the debug mode to the system mode or execution mode (2); or vice versa.		4.5.2
Others	EXIT	Displays the MAIN MENU on the debugger.		4.6
	HELP	Displays the debug command list, functions, and command input procedures.		4.7

4.2 Controlling BASIC Program Operations

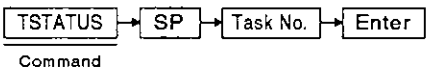
This section tells how to use the debug commands (to control a task) when controlling BASIC programs.

4.2.1 Displaying the state of a designated program (TSTATUS command)

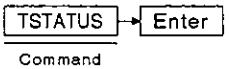
This operation displays the state of a BASIC program in the designated task area.

INPUT PROCEDURE (This command is also referred as "TS")

To designate one task area:



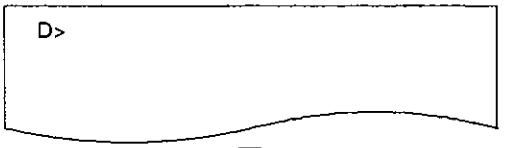
To designate all task areas:

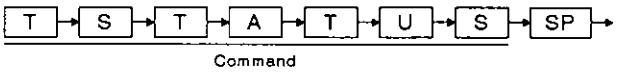


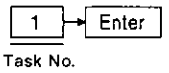
OPERATING EXAMPLE

Displays the status of a BASIC program in the task 1 area.

Before the command is input



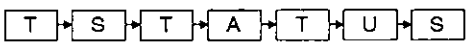




After the command is input

```
D>TSTATUS 1
TASK NO STATUS PRIORITY STEP NO
      1    WAIT      1        150
```

OPERATING PROCEDURE




```
D>TSTATUS
```

1 Input the TSTATUS command to display the state of a BASIC program.

SP → 1 → Enter

```
D>TSTATUS 1
```

2 Input the task number (1 to 8) to display the state.

When all task areas are designated, press the [Enter] key.

(This example assumes that task 1 is designated.)

```
D>TSTATUS 1
TASK NO STATUS PRIORITY STEP NO
     1    WAIT      1        150
```

3 The execution result is displayed.

When the display is not executed normally, the next line shows "TSTATUS: Error" accompanied by the error code.

See Section 4.8, and correct the error.

When the display is executed normally, the contents are shown below:

(This example assumes that the task 1 area status is displayed.)

1) TASK NO : Designated task number

2) STATUS : BASIC program state

DORMANT : Indicates that the printer has not yet been started up in the designated area.

RUN : Indicates that a BASIC program is running.

WAIT : Indicates that a BASIC program is in a wait state.

STOP : Indicates that a BASIC program is not being executed. (*1)

3) PRIORITY : Current priority of a BASIC program.

If the STATUS is DORMANT, "0" is displayed.

4) STEP NO : Step number currently being executed.

If the STATUS is STOP, "0" is displayed.

4 "D>" appears after the execution result is displayed.

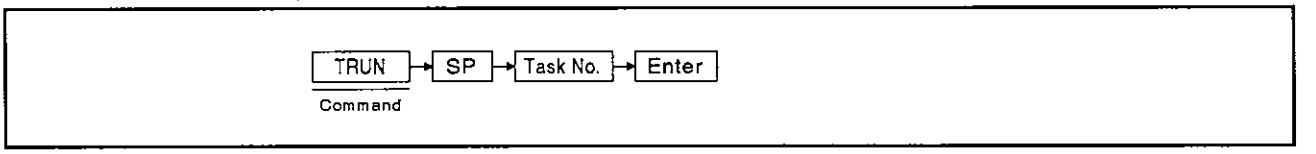
Input the necessary command.

*1 If a designated program is stopped by using the TSTOP debug command, the debugger will be in the STOP state.

4.2.2 Starting the execution of a designated BASIC program (TRUN command)

This operation starts executing a BASIC program in the designated task area.

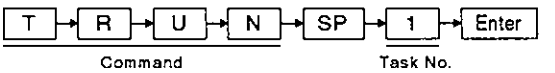
INPUT PROCEDURE (This command is also referred to as "TR")



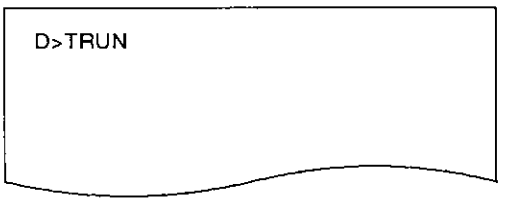
OPERATION EXAMPLE

Starts executing a BASIC program in the task 1 area.

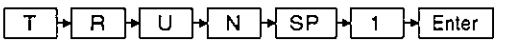
Before the command is input

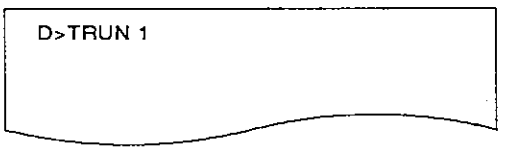


After the command is input



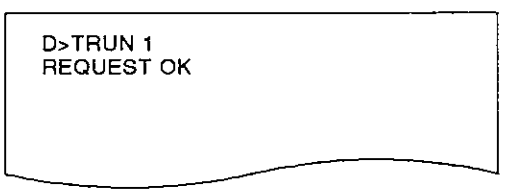
OPERATING PROCEDURE





1 Input the TRUN command and task number (1 to 8) to start executing a BASIC program.

(This example assumes that task 1 is designated.)



2 The next line shows the execution result.

When the TRUN command is executed normally, the screen shows "REQUEST OK".

If the TRUN command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the TRUN command is executed normally.)

3 "D>" appears after the execution result is displayed.
Input the necessary command.

(1) Precautions when inputting a command

- If a task number which corresponds to an area containing a program being executed is designated, then an error will occur.
  If a task number which corresponds to an area which contains no BASIC program is designated, then an error will occur.

(2) Operations when the interpreter has not been started up in the designated task area

- When a task number (which corresponds to an area where the interpreter has not been started up) is designated, the interpreter is started up in the designated area and the BASIC program in that area is started.

(3) Reference

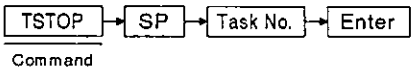
- Stopping the execution of a designated BASIC program..... TSTOP command (see Section 4.2.3)

4.2.3 Stopping the execution of a designated BASIC program (TSTOP command)

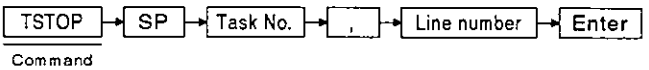
This operation stops the execution of a designated BASIC program in the designated task area.

INPUT PROCEDURE (This command is also referred to as "TP")

To stop the execution immediately:



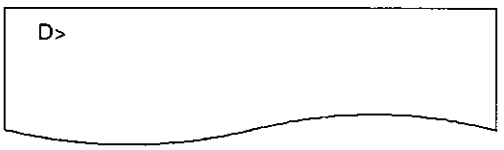
To stop the execution on a designated line:

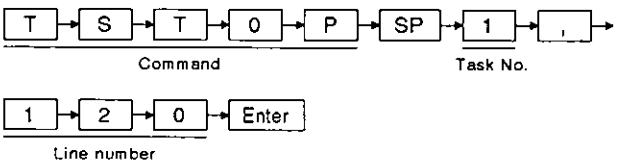


OPERATION EXAMPLE

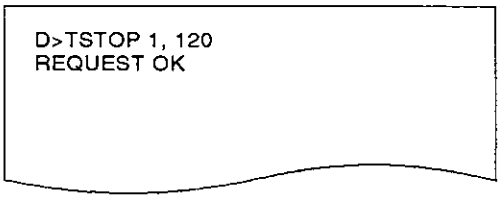
Stops the execution of a designated BASIC program (being executed in the task 1 area) on line 120.

Before the command is input

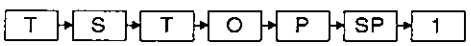


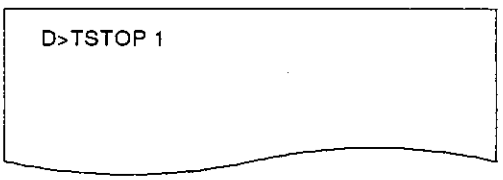


After the command is input



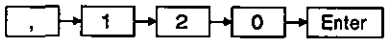
OPERATING PROCEDURE





1 Input the TSTOP command and task number (1 to 8) to stop the execution of a BASIC program.

(This example assumes that task 1 is designated.)




```
D>TSTOP 1, 120
```

2 Input the line number (where the execution will be stopped) in decimal.

When the program is stopped immediately, press the [Enter] key.

When designating "65535" or "–1" as the line number before this command is input, the next execution stop command is canceled.

(This example assumes that the execution is stopped on line 120.)

```
D>TSTOP 1, 120
REQUEST OK
```

3 The next line shows the result of the stop command execution.

When the command is executed normally, the screen shows "REQUEST OK" along with the following stop message:

BREAK : Task No. [ ] Line No. [ ]

Line number where the execution is stopped

Task number of the stopped program

When designating "65535" or "-1" as the line number, giving the TSTOP command normally displays "BREAK Cancel: Task No.[ ]".

If the TSTOP command contains an error, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the TSTOP command is executed normally.)

When the TSTOP command is executed normally, the corresponding program enters the STOP state.

Execution of the program can be resumed by giving the TCONTINUE command.

4 "D>" appears after the execution result is displayed. Input the necessary command.

(1) Precautions when inputting a TSTOP command

- When designating a line number, this number must exist in the program. Input this number in decimal. If a line number that does not exist in the program is input, the program cannot be stopped. In this case, designate "65535" or "–1", and re-input this command.
- A BASIC program can be stopped only at one position.
- When a BASIC program is stopped, make sure that operation does not also stop the system control.

(2) References

- Confirming the current BASIC program status ..... TSTATUS command (see Section 4.2.1)
- Retrying execution of a program from the first line ..... TRUN command (see Section 4.2.2)
- Resuming the execution from the interrupted line ..... TCONTINUE command (see Section 4.2.4)
- Confirming a variable ..... T? command (see Section 4.2.5)
- Assigning a value to a variable ..... TLET command (see Section 4.2.6)

(3) TSTOP command operations

- Stops the execution of a program at the step preceding the designated line when the TSTOP command along with a line number is input.

  If a TSTOP command without a line number is input, the program will not be executed. In this case, after the interpreter completes the command being executed when the [Enter] key was input, the execution stops.

  Therefore, if a line containing several commands (a multistatement) is input, any commands following the command being executed when the [Enter] key was input will not be executed.

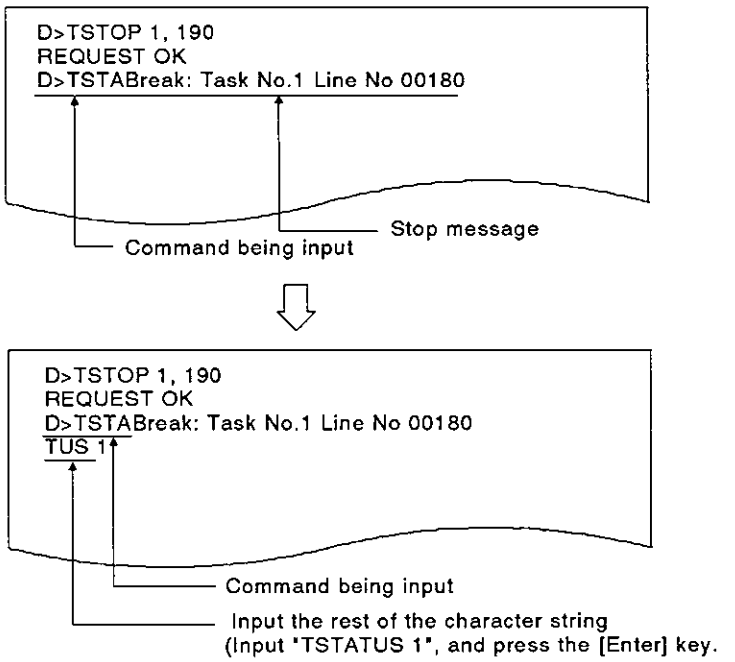
(4) Precautions when inputting the TSTOP command to stop a BASIC program

- When stopping the execution of a BASIC program, the debug (OS) displays the stop message on the line where the cursor is located.

  If a command is input when the debug displays the stop message, the message and the input command overlap on the screen.

  If this happens, continue inputting, since the input command is valid.

(Example) When the stop message appears during the TSTATUS command input



(5) How to stop the interpreter operation in the designated task area

- When stopping the interpreter operation in the designated task area, create a program containing an END command.

4.2.4 Resuming a stopped BASIC program (TCONTINUE command)

This operation resumes the execution of the BASIC program (in the designated task area) stopped by the TSTOP command.

INPUT PROCEDURE (This command is also referred to as "TC")

To resume the execution from the step following the last executed line:

TCONTINUE → SP → Task No. → Enter
(TCONTINUE: Command)

To resume the execution after a command is stopped:

TCONTINUE → SP → Task No. → , → Line No. → Enter
(TCONTINUE: Command)

OPERATION EXAMPLE

Resumes execution of a BASIC program (in the task 1 area) from line 120.

Before the command is input

```
D>TCONTINUE 1, 120
```

T → C → O → N → T → I → N → U → E
(Command)

SP → 1 → , → 1 → 2 → 0 → Enter
(1: Task No.; 120: Line No.)

After the command is input

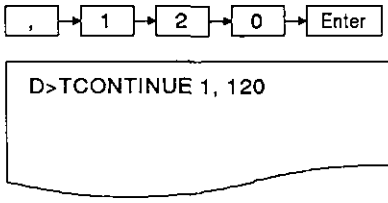
```
D>TCONTINUE 1, 120
REQUEST OK
D>
```

OPERATING PROCEDURE

T → C → O → N → T → I → N →
U → E → SP → 1

```
D>TCONTINUE 1
```

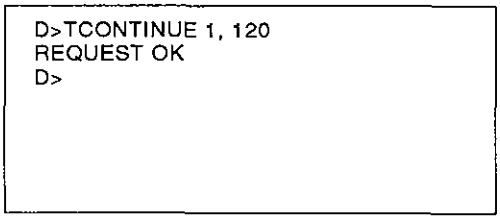
[1] Input the TCONTINUE command and a task number (1 to 8) to resume a BASIC program stopped by the TSTOP command.
(This example assumes that task 1 is designated.)



2 Input the line number (where the execution will be re-started) *in decimal.*

If the program is resumed from the step following the last number already executed, press the [Enter] key.

(This example assumes that line 120 is designated.)



3 The next line shows the result of the stop command execution.

When the command is executed normally, the screen shows "TCONTINUE: REQUEST OK".

If the command is not executed normally, TCONTINUE: Error accompanied by the error code appears.

(This example assumes that the command is executed normally.)

When the command is executed normally, the corresponding program enters the RUN state.

4 "D>" appears after the execution result is displayed.

Input the necessary command.

(1) Precautions when inputting the TCONTINUE command

- The TCONTINUE command can only be executed to a BASIC program stopped by the TSTOP command.

  If the command is executed to a program other than the program stopped by the TSTOP command, an error will occur.

  When the AD51H is set to editing mode (2) by designating a task area that contains a stopped BASIC program, this program cannot be resumed using the TCONTINUE command even if the AD51H is returned to the debug mode using the SYSTEM command.

  The state of a BASIC program can be confirmed by using the TSTATUS command.

(2) Precautions when designating a line number

- Any designated line number must exist in the program.

  Input this number in decimal.

  If a line number that does not exist in the program is input, the program will be resumed from the step following the last executed line.

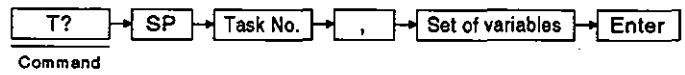
(3) References

- Confirming the current BASIC program status ..... TSTATUS command (see Section 4.2.1)
- Stopping the execution of a designated BASIC program ..... TSTOP command (see Section 4.2.3)
- Resuming the execution from the first line ..... TRUN command (see Section 4.2.2)

4.2.5 Displaying the value of a designated variable in a designated BASIC program (T? command)

This procedure displays designated variable values used in the BASIC program stored in designated task No. area.

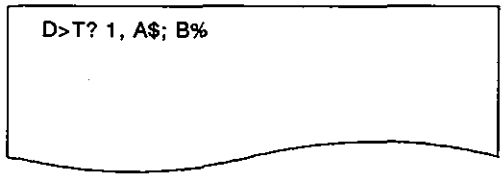
INPUT PROCEDURE (No command abbreviation)

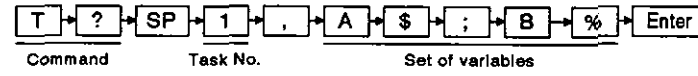


OPERATION EXAMPLE

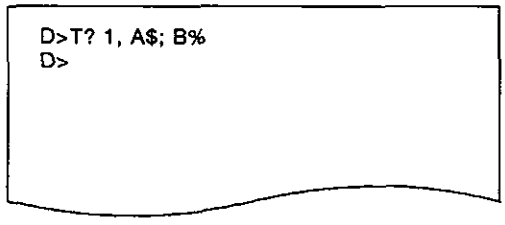
To display values of variables A$ and B% used in the BASIC program stored in task No.1 area with which execution is currently suspended:

Before the command is input

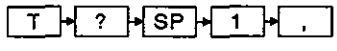


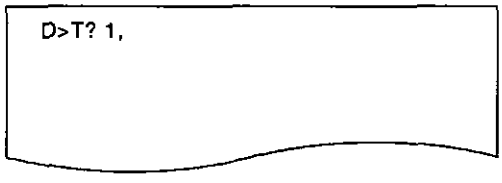


After the command is input



OPERATING PROCEDURE

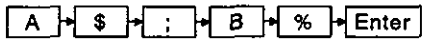


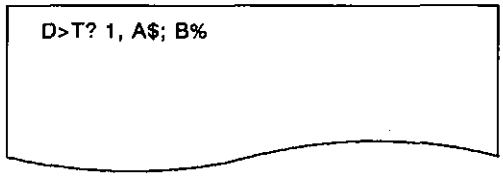


1 Input the T? command and a task number (1 to 8).

The task number must correspond to the area where a designated program exists.

(This example assumes that task 1 is designated.)





2 Input the names of variables whose values are to be displayed.

Like the PRINT command, the T? command can be used with a numerical and character string expression.

By using commas (,) and semicolons (;), several variables can be displayed.

(This example assumes that the values of A$ and B% are displayed.)

```
D>T? 1, A$; B%
D>
```

3 The next line shows the result of the stop command execution.

When the command is executed normally, the screen shows "T?:" accompanied by the values of the variables.

If the command is not executed normally, an T?: Error message accompanied by the error code appears.

See Section 4.8, and correct the error.

(This example assumes that the command is executed normally, which means that A$ and B% store "51H=" and "123" respectively.)

4 "D>" appears after the execution result is displayed.

Input the necessary command.

(1) Precautions when inputting commands

- If a BASIC program (that is set to DORMANT) is designated, then an error will occur.
- Designate the T? command along with its parameters in a line.
  The display caused by the T? command can consist of up to 1024 characters.
- Mitsubishi recommends that the BASIC program to be designated should be in the STOP state when the T? command is input.

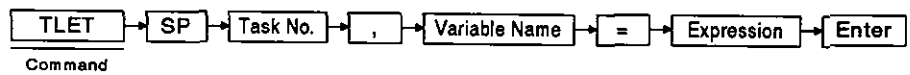
(2) References

- Confirming the current BASIC program status ..... TSTATUS command (see Section 4.2.1)
- Stopping the execution of a designated BASIC program ..... TSTOP command (see Section 4.2.3)
- Resuming the execution of a stopped BASIC program ..... TCONTINUE command (see Section 4.2.4)
- Assigning a value to a variable ..... TLET command (see Section 4.2.6)

4.2.6 Assigning a value to the designated value in the BASIC program (TLET command)

This operation assigns a value to the designated value in the BASIC program.

INPUT PROCEDURE (This command is also referred to as "TL")

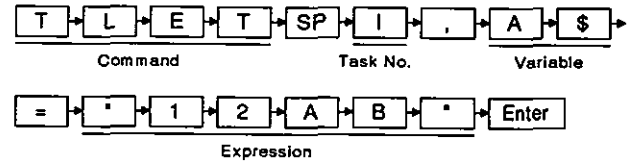


OPERATION EXAMPLE

To assign values to variables A$ and B% used in the BASIC program stored in a task No. area with which execution is currently suspended:

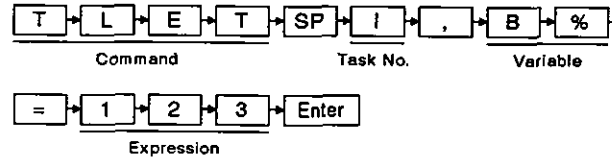
Before the command is input

```
D>TLET 1, A$="12AB"
```

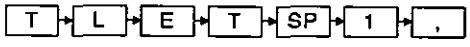


After the command is input

```
D>TLET 1, A$="12AB"
OK
D>
```



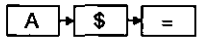
OPERATING PROCEDURE




```
D>TLET 1,
```

1 Input the TLET command and a task number from 1 to 8 (corresponds to the area where the program exists).

(This example assumes that task 1 is designated.)

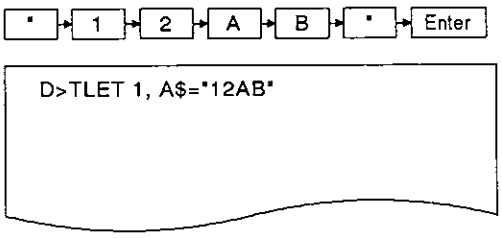



```
D>TLET 1, A$=
```

2 Designate the name of the variable to which a value is assigned.

Like the LET command, the TLET command can be used with an array variable name (ex. C(0), D$(1%), etc.).

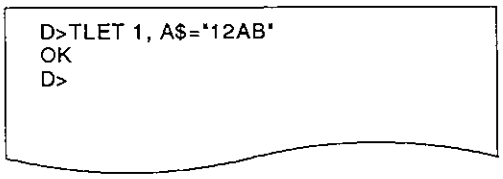
(This example assumes that the character variable A$ is designated.)



3 Input the value to be assigned to the variable.

Like the LET command, the TLET command can be used with a mathematical expression or character expression.

(This example assumes that the character constant of "12AB" is assigned to the A$ character variable.)



4 The next line shows the execution result.

When the TLET command is executed normally, the screen shows "OK".

If the TLET command is not executed normally, an error message accompanied by the error code appears.

See Section 4.8, and correct the error.

(This example assumes that the command is executed normally.)

5 "D>" appears after the execution result is displayed.

Input the necessary command.

(1) Precautions when inputting the TLET command

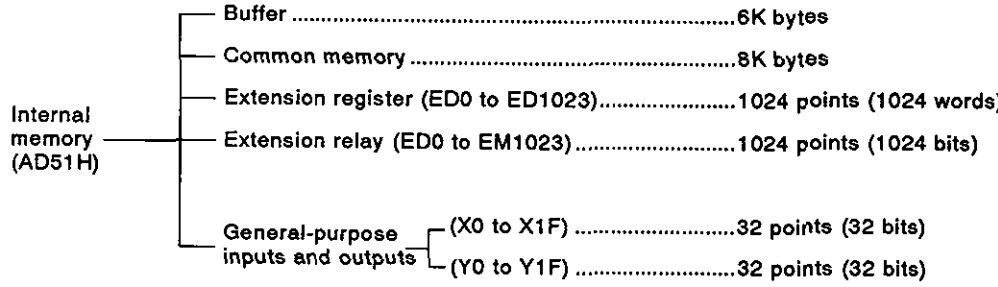
- If a BASIC program that is set to DORMANT is designated, an error will occur.
- Mitsubishi recommends that the BASIC program to be designated should be in the STOP state when the TLET command is input.

(2) References

- Confirming the current BASIC program status ..... TSTATUS command (see Section 4.2.1)
- Stopping the execution of a designated BASIC program ..... TSTOP command (see Section 4.2.3)
- Resuming the execution of a stopped BASIC program ..... TCONTINUE command (see Section 4.2.4)
- Confirming a variable value ..... T? command (see Section 4.2.5)

4.3 Reading/Writing From/To the Internal Memory

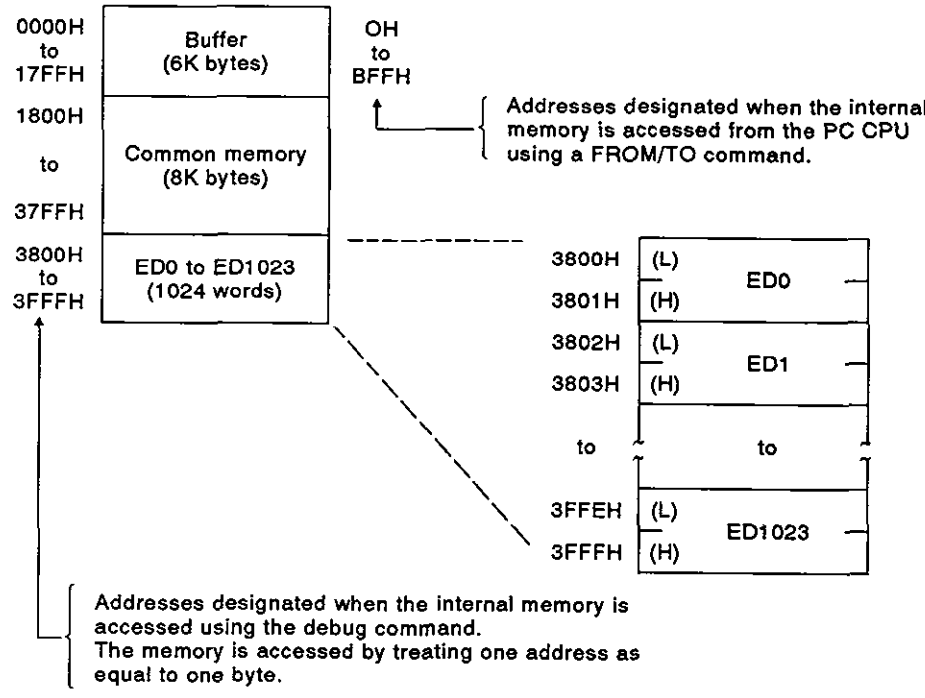
This section tells how to use debug commands when reading/writing from/to the internal memory.



The MREAD and MWRITE commands discussed in this section need designated addresses when the internal memory is accessed.

The correspondence of the addresses used with these commands to the internal memory is shown below:

Access the internal memory within the address ranges.

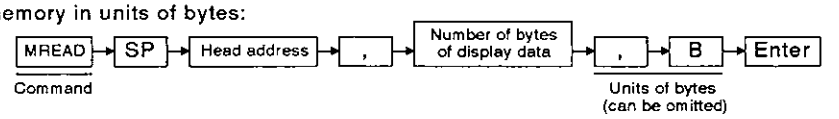


4.3.1 Displaying values in the buffer, common memory, and internal device ED (MREAD command)

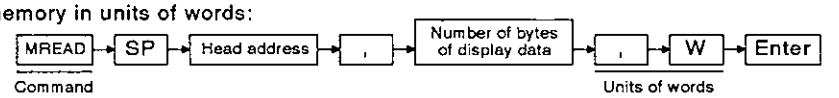
This procedure displays data stored in a designated memory (buffer, common memory, or ED).

INPUT PROCEDURE (This command is also referred to as "MR")

To display values in memory in units of bytes:



To display values in memory in units of words:

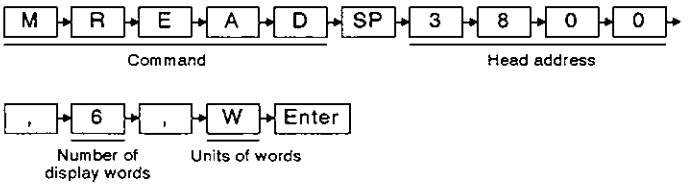


OPERATION EXAMPLE

Displays values in extension registers ED0 to ED5 in units of words.

Before the command is input

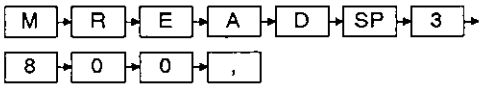
```
D>
```



After the command is input

```
D>MREAD 3800, 6, W
3800 : 0000 0000 0000 0000 0000 0000
D>
```

OPERATING PROCEDURE




```
D>MREAD 3800,
```

1 Input the MREAD command and a head address (with which the display begins) using up to four digits in hexadecimal (0 to 3FFF).

Section 4.3 gives details about the correspondence of addresses to the device memory.

(This example assumes that address 3800H in ED0 is designated.)

When displaying data in units of words, set the first of the lower digits to a even number.

If the first of the lower digits is an odd number, the data in the designated address will not be displayed.

(1) Precautions when designating the number of bytes (words) of display data
When designating the number of bytes (words) of display data, the following conditions must be satisfied:
Address number + Number of display words/Number of display words −1 <3FFFH
If a part of the memory above address 3FFFH is designated, the data in all addresses up to 3FFFH will be displayed.

6 → , → W → Enter

```
D>MREAD 3800, 6, W
```

2 Designate the number of bytes (words) of display memory and the display type.

When designating "B" (can be omitted) as the display type, input the number of words of data in the designated memory range to be displayed.

When designating "W" as the display type, input the number of bytes of data in the designated memory range to be displayed. Input the number of bytes (words) in hexadecimal.

When designating the number in units of bytes:
1H < (Number of bytes) < 4000H

When designating the number in units of words:
1H < (Number of words) < 2000H

(This example assumes that six words are designated.)

```
D>MREAD 3800, 6, W
3800 : 0000 0000 0000 0000 0000 0000
D>
```

3 The next line shows the execution result.

When the MREAD command is executed normally, the data in the designated memory range is set in designated units.

When units of bytes is displayed, each line shows data in addresses [ ][ ][ ]0 to [ ][ ][ ]F (for 16 addresses) as shown below:

However, if the designated address ends with a number other than "0 (n)", spaces are placed for that address on the screen.

[ ][ ][ ][ ] : [ ][ ] [ ][ ] [ ][ ] [ ][ ] to [ ][ ] [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

- [ ][ ][ ][ ]: Indicates the address number.
- [ ][ ] [ ][ ] [ ][ ] [ ][ ] to [ ][ ]: Displays data for 16 addresses (address by address) in hexadecimal, dividing the data into sets of two columns.
- [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]: Display characters corresponding to each address. In the case of OH to 1FH, "." is displayed.

When displaying data in units of words, each line displays the data for 16 addresses as shown below:

[ ][ ][ ][ ] : [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] to [ ][ ][ ][ ]

- [ ][ ][ ][ ]: Indicates the address number.
- [ ][ ][ ][ ] to [ ][ ][ ][ ]: Displays data for 16 addresses (address by address) in hexadecimal, dividing the data into sets of two addresses.

If the MREAD command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the MREAD command is executed normally.)

4 "D>" appears after the execution result is displayed.

Input the necessary command.

(2) Operation when more than 16 lines of data is displayed
The screen can display 16 lines of data (max.).
If more than 16 lines of data must be displayed, pressing any key but the [Esc] key displays the additional data.
Pressing the [Esc] key clears the display.

(3) References
Writing a value to the designated memory ..... MWRITE command (see Section 4.3.2)
Confirming word information in internal device ED ..... W@ command (see Section 4.3.5)
Writing word information in internal device ED ..... W@ command (see Section 4.3.6)

4.3.2 Writing values to the buffer, common device, or designated ED memory (MWRITE command)

This operation writes values to the buffer, common device, or designated ED memory.

INPUT PROCEDURE (This command is also referred to as "MW")

To write a value to the memory in units of bytes (one address):

MWRITE → SP → Head address → , → B → Enter

Command: MWRITE
Units of bytes (can be omitted): B

To write a value to the memory in units of words (two addresses):

MWRITE → SP → Head address → , → W → Enter

Command: MWRITE
Units of words: W

OPERATION EXAMPLE

To write values (0AH, 14H, and 1EH) in word units to ED0 to ED2:

Before the command is input

```
D>
```

After the command is input

```
D>MWRITE 3800, W
3800 : 0000 000A
3801 : 0000 0014
3802 : 0000 001E
3803 : 0000
D>
```

M → W → R → I → T → E → SP → 3 → 8 → 0 → 0
Command: MWRITE
Head address: 3800

, → W → Enter
Write type: W

0 → 0 → 0 → A → Enter
Writes a value to addresses 3800H to 3801H (ED0).

0 → 0 → 1 → 4 → Enter
Writes a value to addresses 3802H to 3803H (ED1).

0 → 0 → 1 → E → Enter
Writes a value to addresses 3804H to

. → Enter
Write completed

OPERATING PROCEDURE

M → W → R → I → T → E → SP →
3 → 8 → 0 → 0 → ,

```
D>MWRITE 3800,
```

1 Input the MWRITE command and a memory address (with which the written value begins) using up to four digits in hexadecimal (0 to 3FFF).

Section 4.3 gives details about the correspondence of the device memory to the addresses.

(This example assumes that address 3800H in ED0 is designated.)

W → Enter

```
D>MWRITE 3800, W
```

2 Designate the write value type.

Designate "B" (can be omitted) as the write value type, when writing the value in units of bytes.

Designate "W" as the write value type, when writing the value in units of words.

(This example assumes that the value are written in units of words.)

```
D>MWRITE 3800, W
3800 : 0000
```

3 The next line shows the execution result.

When the MWRITE command is executed normally, the screen shows the designated addresses and their values as hexadecimal numbers in designated units.

When byte units are designated, the display is as shown below.

Input the value to be written using a one- or two-digit hexadecimal number.

0 → 0 → 0 → A → Enter

```
D>MWRITE 3800, W
3800 : 0000 000A
```

[ ][ ][ ][ ] : [ ][ ] [ ][ ]

- [ ][ ] — Indicates the address number
- [ ][ ] — Displays the memory value corresponding to the address using a hexadecimal number.
- [ ][ ][ ][ ] — Input the value (for one address)

When word units are designated, the display is as shown below:

Input the value to be written using up to four digits in hexadecimal.

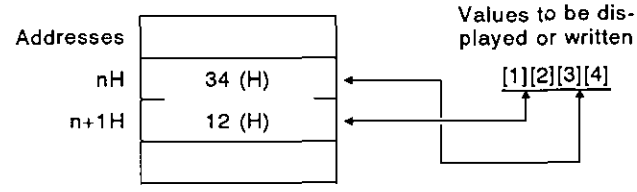
(Inputting only significant digits is valid.)

0 → 0 → 1 → 4 → Enter

```
D>MWRITE 3800, W
3800 : 0000 000A
3801 : 0000 0014
```

[ ][ ][ ][ ] : [ ][ ][ ][ ] : [ ][ ][ ][ ]

- [ ][ ][ ][ ] — Input the values (for two address) in hexadecimal.
- [ ][ ][ ][ ] — Displays memory values for two addresses using up to four digits in hexadecimal.
- [ ][ ][ ][ ] — Indicates the address number



0 → 0 → 1 → E → Enter

```
D>MWRITE 3800, W
3800 : 0000 000A
3801 : 0000 0014
3802 : 0000 001E
```

Use the following keys to write values:

[0] to [9], [A] to [F]: Used to input a value

[.]: Used to end the write operation

[\]: Used to move the address backwards.

[Enter]: Used for inputting (designated using the above keys) or when not changing the current memory value

. → Enter

```
D>MWRITE 3800, W
3800 : 0000 000A
3801 : 0000 0014
3802 : 0000 001E
3803 : 0000
D>
```

If the MWRITE command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the value is written to memory addresses 3800H to 3805H (ED0 to ED2).)

4 "D>" appears after the execution result is displayed.
Input the necessary command.

(1) Processing when the device memory range is exceeded

•• If a value is written to a device memory above 3FFFH, the MWRITE command is automatically stopped.

(2) References

•• Confirming a value to the designated memory..... MREAD command (see Section 4.3.1)

• Confirming word information in extension register ED
.....W@ command (see Section 4.3.5)

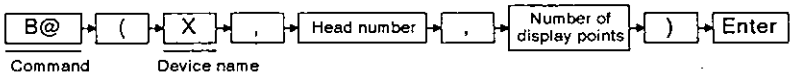
•• Writing word information to extension register ED
.....W@ command (see Section 4.3.6)

4.3.3 Displaying general-purpose input (X)/output (Y), or extension relay EM bit data (B@ command)

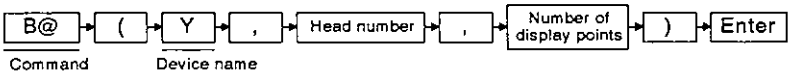
This operation displays bit data for general-purpose input (X)/output (Y) devices (used for communicating with a PC CPU), or an extension relay EM (used for data communications in a BASIC program).

INPUT PROCEDURE (No command abbreviation)

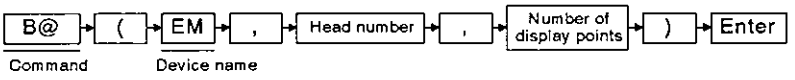
To display bit data for general-purpose input (X) devices:



To display bit data for general-purpose output (Y) devices:



To display bit data for an extension relay EM:

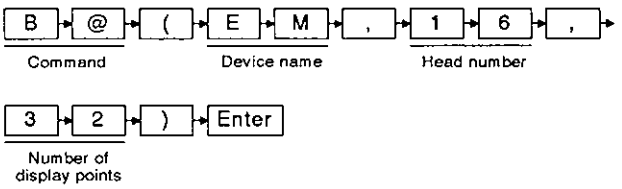


OPERATION EXAMPLE

To display bit data of EM16 to EM47:

Before the command is input

```
D>
```



After the command is input

```
D>B@ (EM, 16, 32)
EM0016 : 00000000-00000000
EM0032 : 00000000-00000000
D>
```

(1) General-purpose I/O devices used for communications between a PC CPU and the AD51H
Those devices handle (a) bit data between a sequence program in a PC CPU and a BASIC program in the AD51H, and (b) bit data controlled by the operating systems.
Section 6.10 gives details about general-purpose I/O devices.

B → @ → (→ E → M → ,

```
D>B@ (EM,
```

1 Input the B@ command and the device name.

Input the device name as shown below:

X : When a general-purpose input is designated (PC CPU ← AD51H)

Y : When a general-purpose output is designated (PC CPU → AD51H)

EM : When an extension relay is designated

(This example assumes that an extension relay is designated.)

1 → 6 → ,

```
D>B@ (EM, 16,
```

2 Input the head number (with which the bit data display will begin).

When "X" or "Y" is designated, input the head number using a one- or two-digit hexadecimal number.

When "EM" is designated, input the head number using up to four digits in decimal.

X/Y: 0 to 1F, EM: 0 to 1023

(This example assumes that EM16 is designated as the head number.)

3 → 2 →) → Enter

```
D>B@ (EM, 16, 32)
```

3 Input the number of display points (bits) corresponding to the device range to be displayed in decimal or hexadecimal.

X/Y: 1(1H) ≤ "Number of display points" 32 ≤ (20H)

EM: 1(1H) ≤ "Number of display points" 1024 ≤ (400H)

(This example assumes that 32 points are designated.)

4 The next line shows the execution result.

```
D>B@ (EM, 16, 32)
EM0016 : 00000000-00000000
EM0032 : 00000000-00000000
```

When the B@ command is executed normally, the screen shows bit data in the designated device ranged.

When "X" or "Y" is designated:

Each line shows bit data of devices [ ][ ][ ]0 to [ ][ ][ ]F for 16 points as shown below:

When "EM" is designated:

Each line shows bit data of devices [multiple of 16] to [next multiple of 16] for 16 points as shown below:

"0" and "1" indicate OFF and ON respectively.

However, if the number designated as "X" or "Y" is not "0"(n), or the number designated as "EM" is not a multiple of 16, spaces are placed in the display columns corresponding to devices 0 or a multiple of 16 to (designated number – 1).

[ ][___]: [ ][ ] to [ ][ ] – [ ][ ] to [ ][ ]

- [ ][___]: Number / Device name
- [ ][ ] to [ ][ ]: Displays bit data for eight bits from the number shown on the left using "0" or "1".
- [ ][ ] to [ ][ ]: Displays bit data for the next eight bits using "0" or "1".

If the B@ command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the B@ command is executed normally.)

5 "D>" appears after the execution result is displayed.

Input the necessary command.

(1) Precautions when designating the number of display points

- When designating the number of display points, the following conditions must be satisfied:
  X/Y ..... Number + Number of display points −1 ≤ 1F (H)
  EM ..... Number + Number of display points −1 ≤ 1023
- If the designated number is outside the device range, device data will be displayed until the last device number is reached.

(2) Operation for displaying more than 16 lines of data

- The screen can display 16 lines of data (max.).
  If more than 16 lines of data must be displayed, pressing any key but the [Esc] key displays the additional data.
- Pressing the [Esc] key clears the display.

(3) Reference

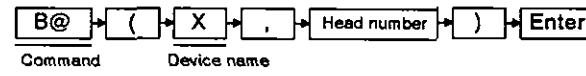
- Writing bit data to internal device EM ..... B@ command (see Section 4.3.4)

4.3.4 Writing bit data to general-purpose input signal (X) devices and internal device EM (B@ command)

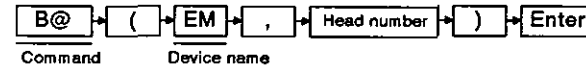
This operation writes bit data to general-purpose input signal (X) (output to a PC CPU) or to internal devices EM (used for data communications between BASIC programs).

INPUT PROCEDURE (No command abbreviation)

To write bit data to general-purpose signal (X) devices:



To write bit data to internal devices EM:



OPERATION EXAMPLE

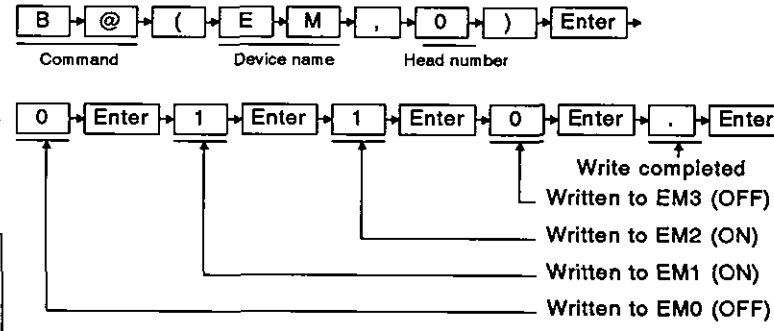
Writes bit data (0, 1, 1, 0) to devices EM0 to EM3.

Before the command is input

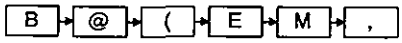
```
D>
```

After the command is input

```
D>B@ (EM, 0) 0110.
EM0000 : 0 0
EM0001 : 0 1
EM0002 : 0 1
EM0003 : 0 0
EM0004 : 0.
D>
```



OPERATING PROCEDURE




```
D>B@ (EM,
```

1 Input the B@ command and device name.

Input the device name as shown below:

X : Name when a general-purpose input signal is designated (PC CPU ← AD51H)

EM : Name when an internal device EM is designated

(This example assumes that the internal device EM is designated.)

---

(1) General-purpose I/O devices used for communications between an PC CPU and the AD51H

- Those devices handle (a) bit data between a sequence program in a PC CPU and a BASIC program in the AD51H, and (b) bit data controlled by the operating systems. Section 6.10 gives details about general-purpose I/O devices.

(2) Precautions when using the B@ command

- To operate the AD51H normally, do not write bit data to general-purpose input signal devices X0B to X0F.

0 →) → Enter

```
D>B@ (EM, 0)
EM0000 : 0 0
```

2 Input the device number with which the write operation will begin.

When "X" is designated, input the head number using a one- or two-digit hexadecimal number.

When "EM" is designated, input the head number using up to four digits in decimal.

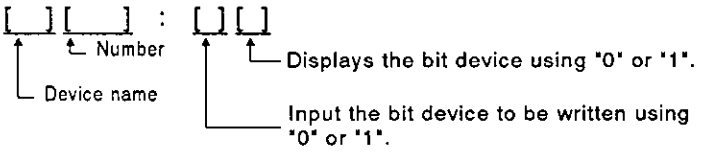
X: 0 to 1F, EM: 0 to 1023

(This example assumes that EM0 is designated.)

3 The next line shows the execution result.

When the B@ command is executed normally, the screen shows bit data corresponding to the specification using "0" and "1".

Input the bit data to be written using "0" and "1".



Use the following keys to write bit data:

[0] : Used to turn OFF the corresponding bit

[1] : Used to turn ON the corresponding bit

[\] : Used to move the corresponding bit backwards

[.] : Used to end the write operation

If the B@ command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the B@ command is executed normally.)

4 "D>" appears after the execution result is displayed.

Input the necessary command.

(1) Processing when the device memory range is exceeded

- When the device memory range in which bit data is written is exceeded, the B@ command is automatically stopped.

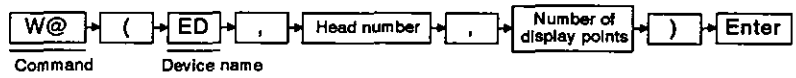
(2) Reference

- Confirming word data in internal device EM ..... B@ command (see Section 4.3.3)

4.3.5 Displaying word data in internal devices ED (W@ command)

This operation displays word devices in internal devices ED that are used for data communications between BASIC programs.

INPUT PROCEDURE (No command abbreviation)

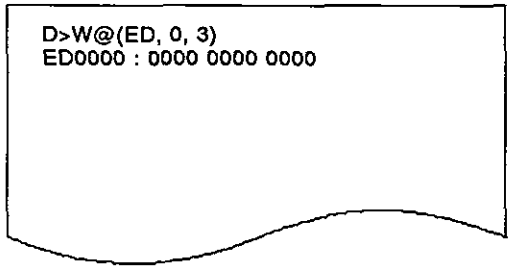


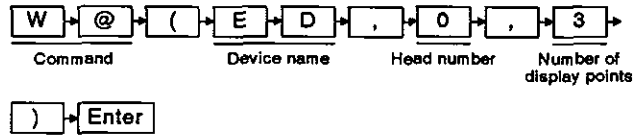
OPERATION EXAMPLE

Displays word data (values) in devices ED0 to ED2.

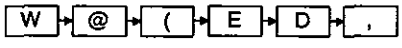
Before the command is input

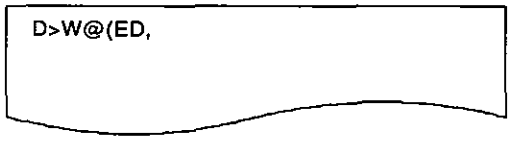
After the command is input



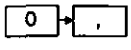


OPERATING PROCEDURE





1 Input the W@ command and the internal device name.



2 Input the head number (with which the word display will begin) using up to four digits (0 to 1023) in decimal.

(This example assumes that ED0 is designated.)

3 →) → Enter

3 Input the number of display points (words) in decimal.

ED: 1 ≤ Number of display points ≤ 1024

(This example assumes that three points are designated.)

```
D>W@(ED, 0, 3)
ED0000 : 0000 0000 0000
```

4 The next line shows the execution result.

When the W@ command is executed normally, the screen shows word data corresponding to the designated range.

Each line shows bit data of devices (a multiple of eight) to (the next multiple of eight) for 8 points in hexadecimal as shown below:

However, when the designated number is not a multiple of eight (n), spaces are placed in the display columns corresponding to devices (a multiple of eight) to (designated number – 1).

ED [ ][ ][ ][ ] : [ ][ ][ ][ ] [ ][ ][ ][ ] to [ ][ ][ ][ ]

- [ ][ ][ ][ ]: Displays the number
- [ ][ ][ ][ ] to [ ][ ][ ][ ]: Sequentially displays word data for eight points using four hexadecimal digits.

If the W@ command is not executed normally, W@: Error accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the W@ command is executed normally.)

5 "D>" appears after the execution result is displayed.

Input the necessary command.

---

(1) Precautions when designating the number of display points

- When designating the number of display points, the following condition must be satisfied:
  ED: Head number + Number of display points −1 ≤ 1023
- If the designated number exceeds ED1023, device data will be displayed up to ED1023.

(2) Operation when more than 16 lines of data is displayed

- The screen can display 16 lines of data (max.).
  If more than 16 lines of data must be displayed, pressing any key but the [Esc] key displays the additional data.
- Pressing the [Esc] key clears the display.

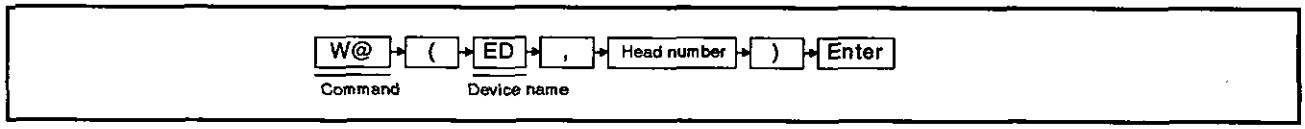
(3) References

- Confirming values in the designated memory (requires the address specification) ..... MREAD command (see Section 4.3.1)
- Writing values to the designated memory (requires the address specification) ..... MWRITE command (see Section 4.3.2)
- Writing word data in internal device ED ..... W@ command (see Section 4.3.6)

4.3.6 Writing word data to internal devices ED (W@ command)

This operation writes word data to internal devices ED that are used for data communications between BASIC programs.

INPUT PROCEDURE (No command abbreviation)

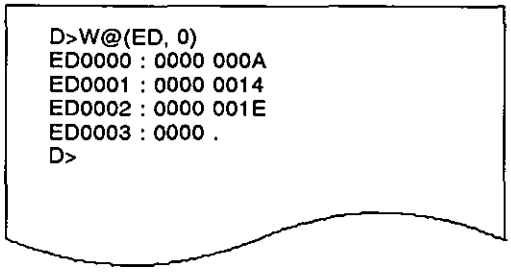


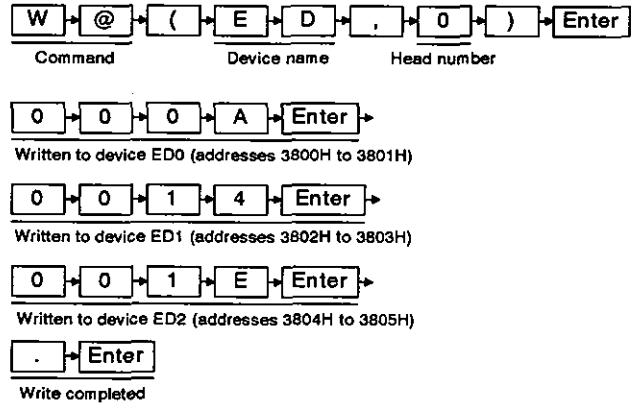
OPERATION EXAMPLE

Writes word data (0AH, 14H, 1EH) to devices ED0 to ED2.

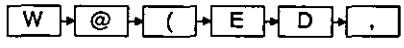
Before the command is input

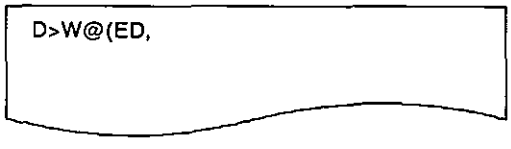
After the command is input





OPERATING PROCEDURE





1 Input the W@ command and the internal device ED name.

(1) Writing to internal devices ED
Word data can be written to internal devices ED using the MWRITE command as well.

0 →) → Enter

```
D>W@(ED,0)
```

2 Input the ED number (with which word data writing begins) using up to four digits (0 to 1023) in decimal.

(This example assumes that ED0 is designated.)

```
D>W@(ED,0)
ED0000 : 0000
```

3 The next line shows the execution result.

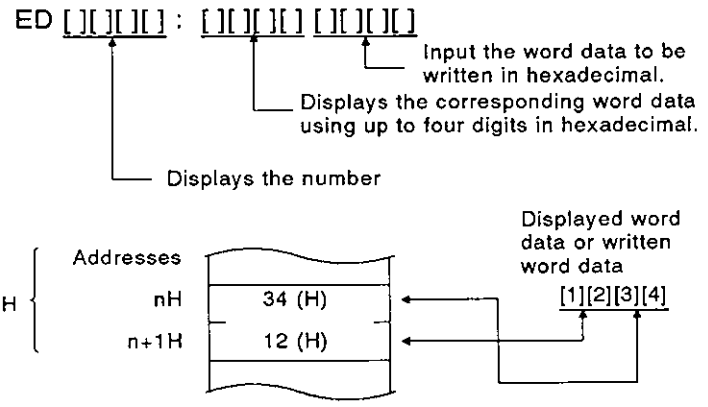
When the W@ command is executed normally, the screen shows the designated device number and word data as shown below:

Input word data (to be written) using up to four digits in hexadecimal.

(It is possible to input only significant digits.)

0 → 0 → 0 → A → Enter

```
D>W@(ED,0)
ED0000 : 0000 000A
```



0 → 0 → 1 → 4 → Enter

```
D>W@(ED,0)
ED0000 : 0000 000A
ED0001 : 0000 0014
```

Use the following keys to write bit data:

[0] to [9], [A] to [F] : Used to input word data

[\] : Used to move the corresponding number backwards

[.] : Used to end the write operation

[Enter] : Used to input when the above-mentioned keys are used. Used to leave the current data as it is.

0 → 0 → 1 → E → Enter

```
D>W@(ED,0)
ED0000 : 0000 000A
ED0001 : 0000 0014
ED0002 : 0000 001E
```

If the W@ command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that word data is written to devices ED0 to ED2 (Addresses 3800H to 3805H).)

. → Enter

```
D>W@(ED,0)
ED0000 : 0000 000A
ED0001 : 0000 0014
ED0002 : 0000 001E
ED0003 : 0000
D>
```

4 "D>" appears after the execution result is displayed.

Input the necessary command.

(1) Operation when the designated number exceeds ED1023

• If the designated number exceeds ED1023, device data will be displayed up to ED1023

(2) References

Confirming values in the designated memory (requires the address specification) ..... MREAD command (see Section 4.3.1)

Writing values to the designated memory (requires the address specification) ..... MWRITE command (see Section 4.3.2)

Writing word data to internal device ED ..... W@ command (see Section 4.3.5)

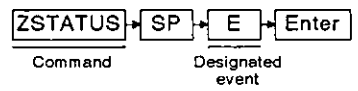
4.4 Confirming the State of Events, Message Ports, and Source Numbers

This section tells how to use the commands to confirm the current states of events, message ports, and source numbers shared by BASIC programs.

4.4.1 Displaying event declaration states (valid/invalid) (ZSTATUS command)

This operation displays the current declaration states (valid/invalid) of events (shared by BASIC programs) for each event.

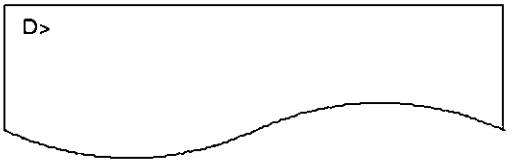
INPUT PROCEDURE (This command is also referred to as "ZS")

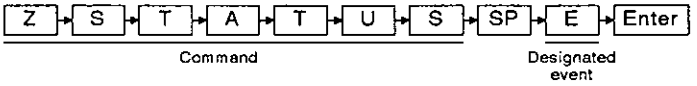


OPERATION EXAMPLE

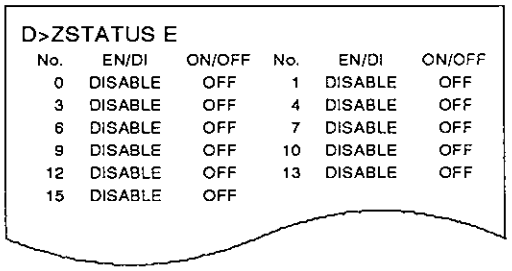
Displays the current declaration states (valid/invalid) for each event.

Before the command is input





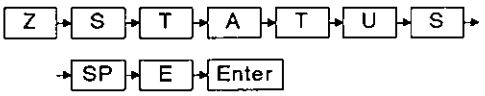
After the command is input



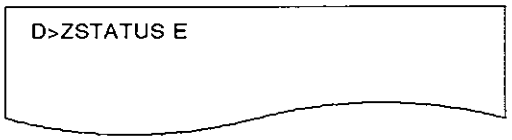
D>ZSTATUS E

No.	EN/DI	ON/OFF	No.	EN/DI	ON/OFF
0	DISABLE	OFF	1	DISABLE	OFF
3	DISABLE	OFF	4	DISABLE	OFF
6	DISABLE	OFF	7	DISABLE	OFF
9	DISABLE	OFF	10	DISABLE	OFF
12	DISABLE	OFF	13	DISABLE	OFF
15	DISABLE	OFF			

OPERATING PROCEDURE



1 Input the ZSTATUS command and "E" (to designate the display data type as an event)



(1) How to use events
Events are used with the following commands:
(The AD51H Programming Manual gives details.)

- Defining an event ..... DEF ZEVENT command
- Declaring an event valid or invalid ..... ZEVENT command
- Generating an event ..... ZSIGNAL command
- Waiting to generate an event ..... ZWAIT EVENT command

```
D>ZSTATUS E
No.   EN/DI     ON/OFF   No.   EN/DI     ON/OFF
 0    DISABLE   OFF       1    DISABLE   OFF
 3    DISABLE   OFF       4    DISABLE   OFF
 6    DISABLE   OFF       7    DISABLE   OFF
 9    DISABLE   OFF      10    DISABLE   OFF
12    DISABLE   OFF      13    DISABLE   OFF
15    DISABLE   OFF
```

2 The execution results are displayed.

When the B@ command is executed normally, the next lines show whether each event (0 to 63) is valid or invalid.

The display contents are given in the following table:

EN/DI	ON/OFF	Meaning
ENABLE	ON	Event (corresponding to the number) is already defined and declared as valid.
ENABLE	OFF	Event (corresponding to the number) is already defined, but not declared.
DISABLE	ON	Event (corresponding to the number) is already defined and declared as invalid.
DISABLE	OFF	Event (corresponding to the number) is not defined.

If the B@ command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the B@ command is executed normally.)

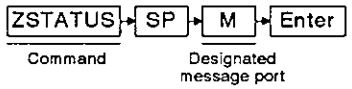
3 "D>" appears after the execution result is displayed.

Input the necessary command.

4.4.2 Displaying the state of a message transmitted to a message port shared by BASIC programs (ZSTATUS command)

This operation displays the state of transmission of a message at each message port which is shared by BASIC programs.

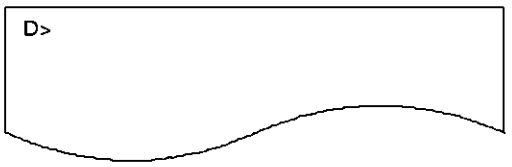
INPUT PROCEDURE (This command is also referred to "ZS")

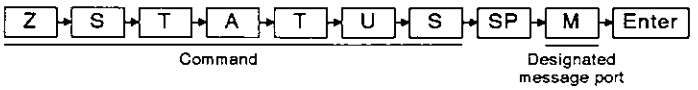


OPERATION EXAMPLE

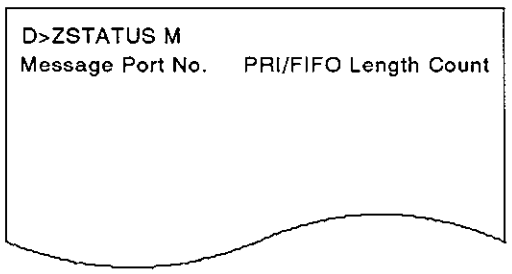
To display the current state of message transmission:

Before the command is input

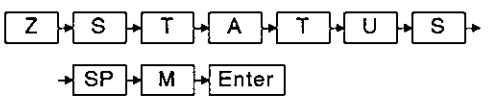




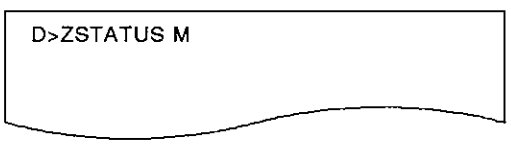
After the command is input



OPERATING PROCEDURE



1 Input the ZSTATUS command and "M" to display the transmission state of the message port.



(1) Message transmission via a message port

Messages can be transmitted/received between BASIC programs by defining the message port in the BASIC programs.

The AD51H-BASIC Programming Manual gives details.

All related commands begin with "ZMESSAGE".

```
D>ZSTATUS M
Message Port No.    PRI/FIFO Length Count
```

2 The execution results are displayed.

When the ZSTATUS command is executed, the next lines show the transmission states (information about unreceived messages) for each message port.

The display contents are shown below:

Message Port N : Message port number defined by the user.

PRI/FLSO : Shows the type of corresponding message port as follows:

PRI : "FIFO" designated when defining the port.

FIFO : "FIFO" not designated when defining the port.

Length : Byte length when defining the corresponding message port.

Count : Number of unreceived messages that were transmitted to the corresponding port.

If the ZSTATUS command is not executed normally, the next line shows an error message along with the error code.

(This example assumes that the ZSTATUS command is executed normally.)

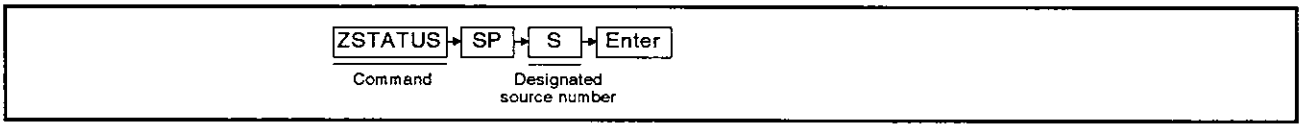
3 "D>" appears after the execution result is displayed.

Input the necessary command.

4.4.3 Displaying the reserve/release states of source numbers used for exclusive control (ZSTATUS command)

This operation displays the reserve/release states of source numbers used for exclusive control of memory and external devices.

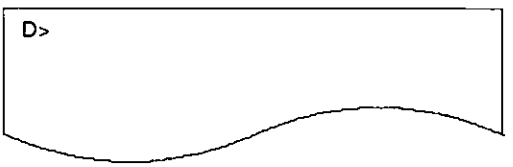
INPUT PROCEDURE (This command is also referred to as "ZS")

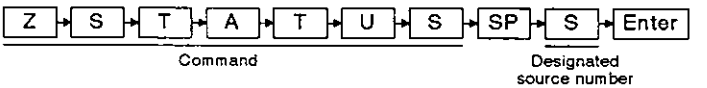


OPERATION EXAMPLE

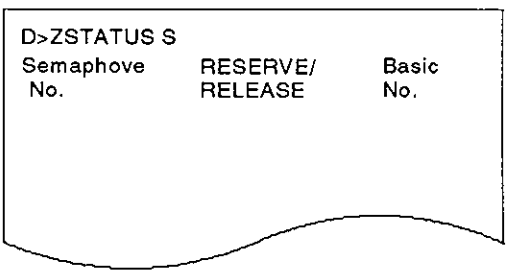
Displays the reserve/release state of the current source number.

Before the command is input

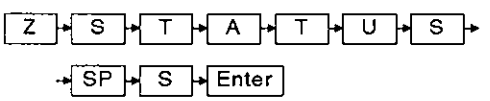




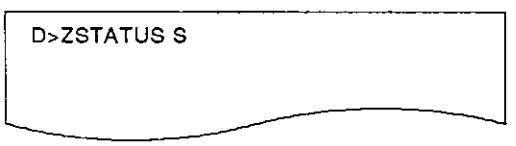
After the command is input



OPERATING PROCEDURE



1 Input the ZSTATUS command and "S" to display the state of the source number.



(1) Exclusively controlling a source by reserving/releasing its source number
When executing several BASIC programs at the same time, the BASIC programs uses the following commands exclusively to control the sources.
The AD51H-BASIC Programming Manual gives details.
Reserving a source number ..... ZRESERVE command
Releasing a source number..... ZRELEASE command

```
D>ZSTATUS S
Semaphove      RESERVE/      Basic
 No.           RELEASE       No.
```

2 The execution results are displayed.

When the ZSTATUS command is executed normally, the next lines show the reserve/release states for source numbers 0 to 31.

The display contents are shown below:

Semaphove No. : Source number

RESERVE/RELEASE: Indicates the reserve/release state corresponding to the source.

RESERVE : Indicates the source is in the reserve state.

RELEASE : Indicates the source is in the release state.

Basic No. area : Corresponding source No.

If the ZSTATUS command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

(This example assumes that the ZSTATUS command is executed normally.)

3 "D>" appears after the execution result is displayed.

Input the necessary command.

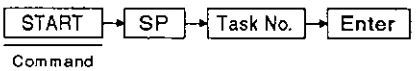
4.5 Changing the AD51H Mode

This section tells how to use debug commands to change the mode of the AD51H.

4.5.1 Setting the AD51H to editing mode (2) (START command)

This operation sets the AD51H to the editing mode to edit (create, change) another program using the designated task area when several BASIC programs are executed.

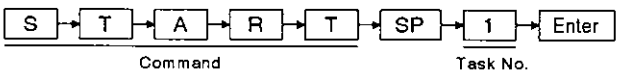
INPUT PROCEDURE (This command is also referred to as "ST")



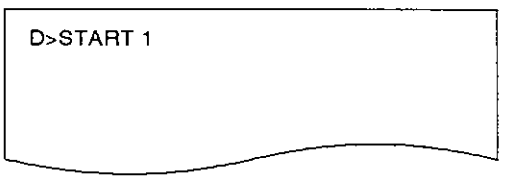
OPERATION EXAMPLE

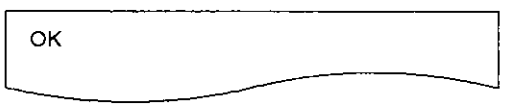
Sets the AD51H to editing mode (2) to modify a BASIC program (whose execution is stopped) in the task 1 area.

Before the command is input



After the command is input

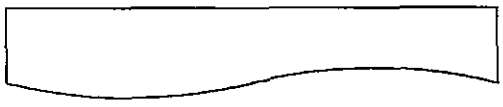




When the interpreter was not started, the following message appears before "OK".
"AD51H-BASIC ON-LINE PROGRAMMING Ver [ ]"

OPERATING PROCEDURE

S → T → A → R → T → SP →
1 → Enter



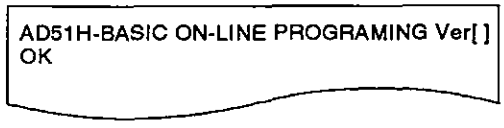
1 Input the START command and a task number (corresponding to the program to be edited) (1 to 8) .

The task number can be omitted.

When the task number is omitted, the START command is designated as shown below:

1) When the START command is initially input, the task number is automatically set to 1.
2) When the START command is not initially input, the previously-designated task number is used.

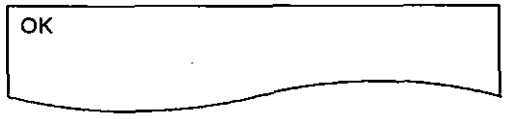
(This example assumes that task 1 is designated.)



2 The execution results are displayed.

When the START command is executed normally, the screen shows the display contents indicated on the left.

Thereafter, start editing the BASIC program.



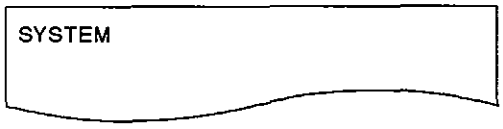
The programming manual tells how to edit a BASIC program.

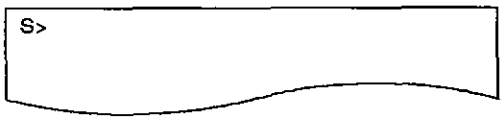
If the START command is not executed normally, an error message accompanied by the error code appears.

See Section 3.9, and correct the error.

The display contents when the interpreter was not started are shown on the upper left.

The display contents when the interpreter was started are shown on the lower left.





3 After completing the BASIC program, do either 1) or 2) below when returning the AD51H from editing mode (2) to the debug mode:

1) Execute the SYSTEM command.
   - Stops the BASIC program execution.
   - Closes the open files and communications line.
2) Press the [Ctrl] + [D] keys.
   - Stops the BASIC program execution.
   - Leaves the open files and communications line as they are.
   - If the BASIC program was not modified, the execution of the program can be resumed using the TCONTINUE debug command.
     When resetting the AD51H to the editing mode, the execution can be resumed using the CONTINUE command.

(1) Precautions when using the START command

- When a BASIC program is being executed in the task area used for editing another program, stop the program using the TSTOP command.

(2) Other BASIC program operations when the START command is executed

- When several BASIC programs are being executed, even if an edit operation in a task area is started, programs in other task areas will continue to be executed.

(3) Necessary measures for changing designated task sizes

- Set the AD51H to editing mode (1) and do the following:

  1) Return the AD51H to the debug mode using the SYSTEM command.

  2) Stop the BASIC programs in the task areas using the TSTOP command so that the operation does not also stop the system control.

  3) Use the GO command to set the AD51H to the system mode.

  4) Use the TKILL system command to end the interpreter operation in the task areas.

  5) Use the START command to set the AD51H to editing mode (1).
  After giving the START command, change the task size, and edit the program.

Section 2.4 gives the mode change chart.

(4) Reference

- Changing the AD51H mode .....GO command (see Section 4.5.2)

4.5.2 Setting the AD51H to the system mode, execution mode (2), or debug mode (GO command)

This operation switches the debug mode to the system mode or execution mode (2), and vice versa.

By setting the AD51H to the system mode, it is possible to give system commands to the console to edit/debug a BASIC program (exists in a task area).

(1) When the AD51H switches to execution mode (2), setting multitasking starts the execution of a BASIC program.

Returning the AD51H to the debug mode restarts the debug operation.

Setting multitasking starts the execution of the BASIC program.

The following table gives the relationship between the mode/debug start specifications (when the GO command is input) and the console/debugger states (after the GO command is input):

<table>
<tr><th>Mode Setting</th><th>Debug Start Specification</th><th>Console State</th><th>Debug Terminal State</th><th>Remarks</th></tr>
<tr><td rowspan="2">R
(Execution Mode (1))</td><td>YES
(to the debug mode)</td><td rowspan="2">Clears the display contents.
Set to the console used for running the BASIC program.</td><td>Restarts the debug, clears the display contents, and displays "D>". In this state, the debug command can be input.</td><td rowspan="2">Setting multitasking reloads the BASIC program to the corresponding task area and starts the execution.</td></tr>
<tr><td>NO
(to execution mode (2))</td><td rowspan="2">The display contents remains unchanged.</td></tr>
<tr><td>P
[Programming in the system mode]</td><td>Cannot be designated.</td><td>Clears the display contents, and displays "S>". The system commands can be input.</td><td>Stops the execution of BASIC programs in the task areas.</td></tr>
</table>

(1) BASIC program states when the GO command is executed

- When execution mode (1) is designated, if mode switch (1) is set to 0 to 3, the BASIC program starts in the same way when the AD51H starts up.
- When the system mode is designated, the execution of all BASIC programs in the task areas is stopped.

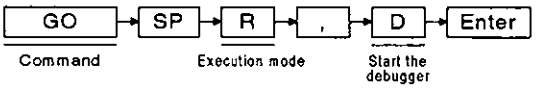
  (All BASIC programs but the program in the DORMANT state enter the STOP state)

  Since all task areas (in the main memory) conditions are unchanged, the BASIC programs in the task areas remain as they are.

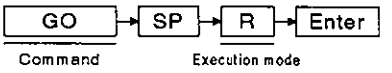
  By changing the AD51H from the system mode to editing mode (1), it is possible to edit/debug a BASIC program in a task area.

INPUT PROCEDURE (No command abbreviation)

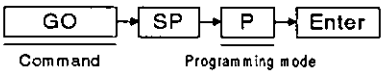
To reset the AD51H to the debug mode:



To set the AD51H to execution mode (2):

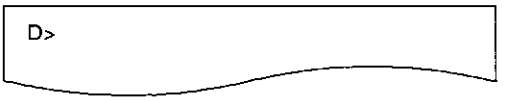


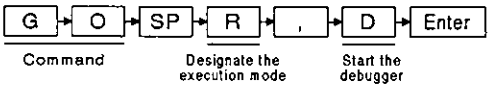
To set the AD51H to the system mode (one of the programming modes):



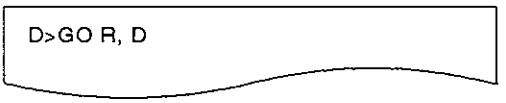
OPERATION EXAMPLE

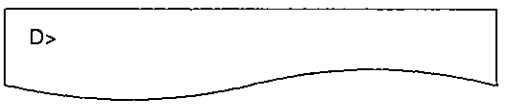
Returns the AD51H to the debug mode.
Before the command is input



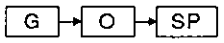


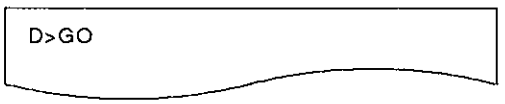
After the command is input



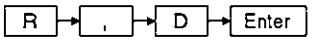


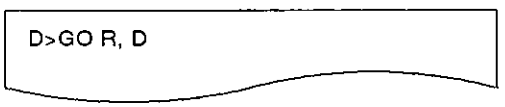
OPERATING PROCEDURE





1 Input the GO command to change the AD51H command.





2 Designate the mode.
Input "R" to set the AD51H to execution mode (2) or the debug mode.
Input "P" to set the AD51H to the system mode.
When "R" is designated, to return the AD51H from execution mode (2) to the debug mode, input "D" following the "R".
(This example assumes that the AD51H is returned to the debug mode.)

(2) Precautions when changing the mode
When switching the AD51H from the debug mode to another mode, Mitsubishi recommends that the execution of BASIC programs should be stopped (see the TSTOP command) beforehand to prevent the system control from being stopped.

3 The result of the GO command execution is displayed.

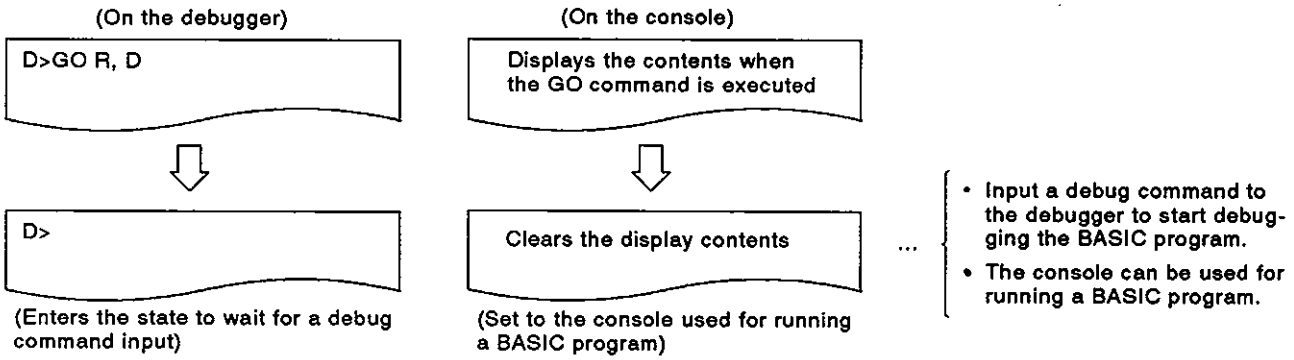
When the command is executed normally, the screen shows the following:

If the GO command is not executed normally, an error message accompanied by the error code appears.

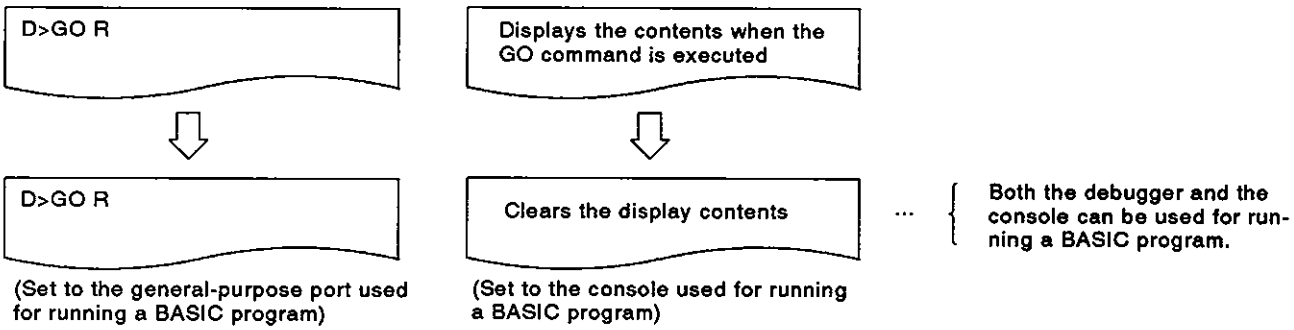
See Section 3.9, and correct the error.

The following example gives the display contents when the GO command is executed normally:

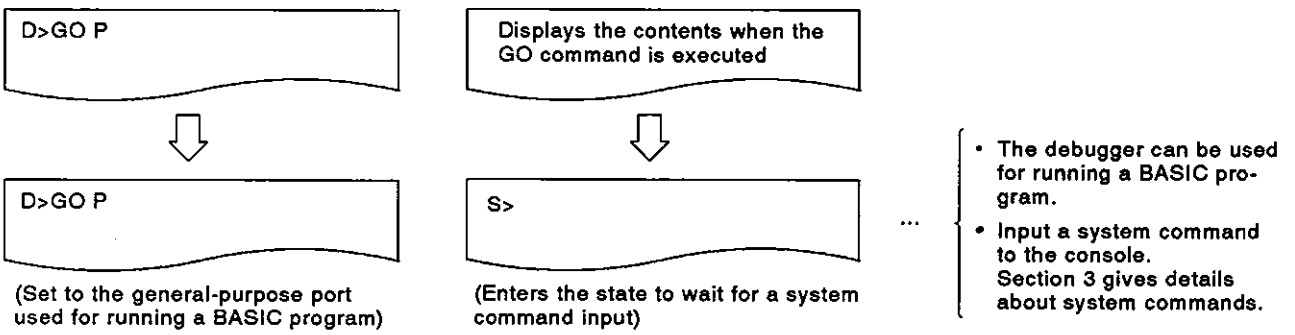
(1) When returning the AD51H to the debug mode:



(2) When setting the AD51H to execution mode (2):



(3) When setting the AD51H to the system mode:



(3) AD51H mode changes
- Section 2.4 gives the AD51H mode change chart.

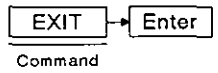
(4) Reference
- Displaying the MAIN MENU on the debugger .....EXIT command (see Section 4.6)

4.6 Displaying the MAIN MENU on the Debugger

This section tells how to use the EXIT command to display the MAIN MENU (contained in the AD51H-BASIC software package) when a PC/AT is used as the debugger,

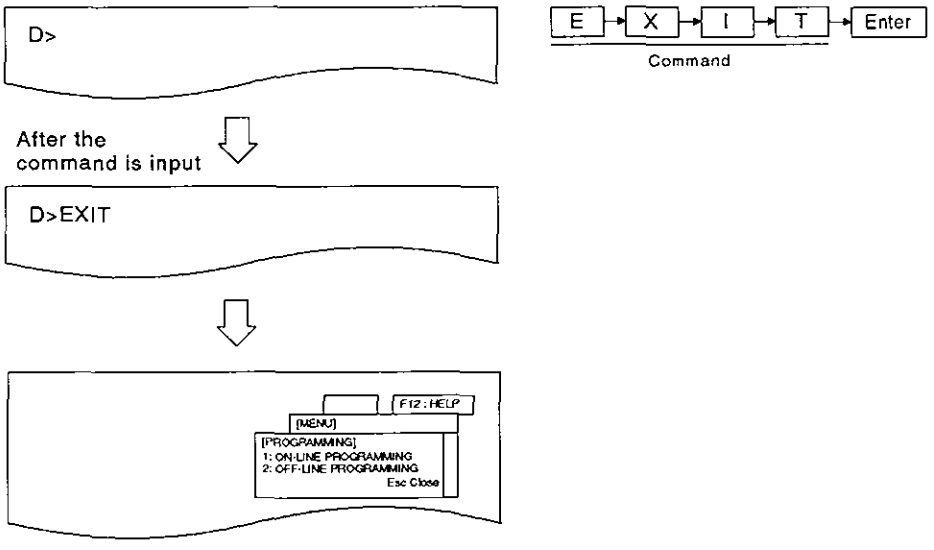
When a VG-620 or a VT-382/VT-220 is used as the debugger, pressing any key redisplays "D>" after the EXIT command is input.

INPUT PROCEDURE (This command is also referred to as "E")

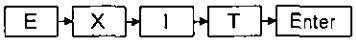


OPERATION EXAMPLE

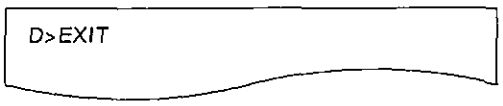
Displays the MAIN MENU on the PC/AT used as the debugger.
Before the command is input



OPERATING PROCEDURE



1 Input the EXIT command to display the MAIN MENU.



(1) BASIC program states when the EXIT command is executed
Even if the EXIT command is executed, execution of BASIC programs in the task area continues.

(2) Precautions when inputting the TSTOP command
When a BASIC program in a task area is edited using the menu screens, give the TSTOP command to stop the execution of the BASIC programs to prevent the system control from being stopped.

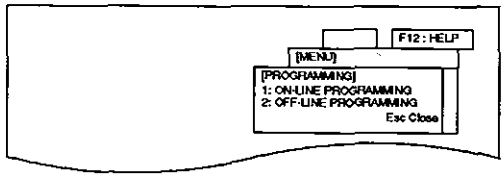
2 After the execution result is displayed, do the corresponding operation.

When the EXIT command is executed normally, the screen shows the following contents:

If the EXIT command is not executed normally, an error message accompanied by the error code appears.

See Section 5, and correct the error.

[When a PC/AT is used]



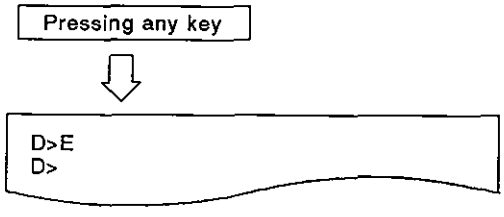
The following display contents appear when the command is executed normally:

1) When a PC/AT is used as the debugger:

The MAIN MENU appears on the debugger. Select one of the items from the MAIN MENU.

The SW0IX-AD51HPE AD51H-BASIC Operating Manual gives details about menu-driven operations.

[When a VG-620 or a VT-382/VT-220 is used]



2) When a VG-620 or a VT-382/VT-220 is used as the debugger:

The debugger enters the state to wait for a key input.

Pressing any key displays "D>". Input a debug command.

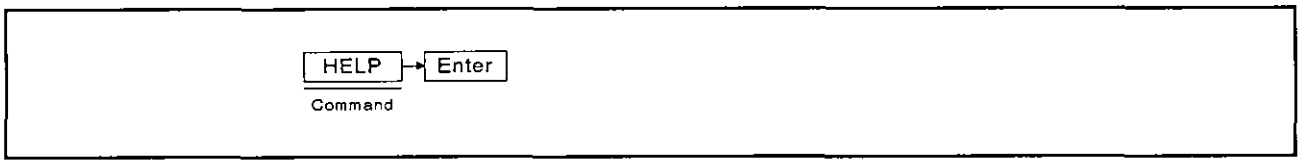
(3) AD51H mode changes
- Section 2.4 gives the AD51H mode change chart.

(4) References
- Stopping the execution of a designated BASIC program ..... TSTOP command (see Section 4.2.3)
- Changing the AD51H mode ..... GO command (see Section 4.5.2)

4.7 Confirming the Input Procedure for Debug Commands

This section tells how to use the HELP command to display on the debugger the input procedure for debug commands.

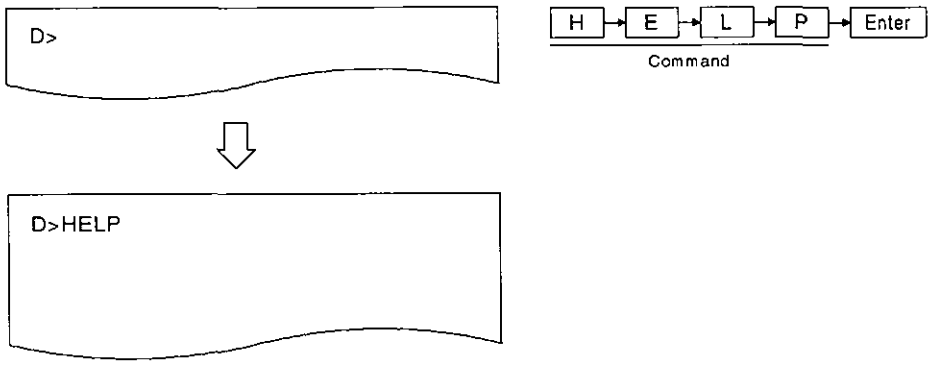
INPUT PROCEDURE (This command is also referred to as "H")



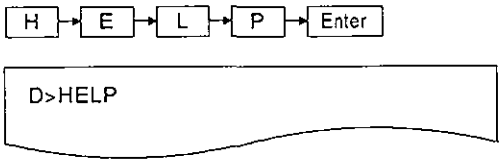
OPERATION EXAMPLE

Displays the input procedure for debug commands.

Before the command is input



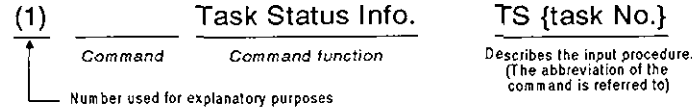
OPERATING PROCEDURE



1 Input the HELP command to display the input procedure for debug commands.

2 The result of the command execution is displayed.
When the HELP command is executed normally, the next lines show the types, functions, and input procedure for debug commands.

(Example)



If the HELP command is not executed normally, the next line shows an error message.
See Section 5, and correct the error.

3 After displaying the result of the command execution, "D>" appears.
Input the necessary command.

(1) Description of the command input procedures

A space (located following a command) for one column indicates pressing the [SP] (space) key.

A pair of parentheses "(" and ")" indicates inputting parentheses.

A pair of braces "{" and "}" indicates an argument. However, they do not actually have to be input.

A pair of brackets ("[" and "]") indicates "can be omitted". However, they do not actually have to be input.

[APPENDICES]

APPENDIX 1 Function Comparison between the AD51H and the AD51 (S3)

Table 1.1 compares the functions of the AD51H and AD51 (S3).

Table 1.1 AD51H and the AD51(S3) Function Comparison

Item		AD51H	AD51(S3)
CPU device		80C186 (15MHz)	HD64180 (8MHz)
Computing device		None	Available
Programming language		AD51H-BASIC (interpreter)	GPP-BASIC (interpreter)
Number of tasks		8	8
Task-starting conditions		• Started when the power is turned ON. • Started by an interruption from a PC CPU. • Started when a request to start from another task is received.	• Started when the power is turned ON. • Started by an interruption from a PC CPU. • Started by a real-time interruption.
Internal memory	Program	384K bytes (each task can be set to 16K/32K/48K/64K bytes.)	64K bytes + 48K bytes (when the ROM is also used to store programs)
	Common memory	8K bytes	2K bytes
	Buffer	6K bytes	6K bytes
	EM	1024 points	—
	ED	1024 points	—
General-purpose inputs and outputs		Inputs : 27 points Outputs : 26 points (Nine points are used for starting a task from a PC CPU)	Inputs : 26 points Outputs : 17 points
Memory-protect		Available (memory card can be protected)	Available
Interface	RS-232C	2 channels	2 channels
	RS-422	1 channel (multi-drop cannot be used for the connector)	2 channels (multi-drop can be used: CH1)
	Parallel	1 channel	None
	Memory card	2 channels	None
Clock function		None (It is possible to read/write the clock data from/to a PC CPU.)	Provided
Latch		None (memory card can be latched)	Provided
Writing a BASIC Program to ROM		Possible [(64K ROM-W/128K ROM-W/256K ROM-W) *1]	Possible [(8K ROM/16K ROM) *2]
Console		• PC/AT • VG-620 • VT-220/VT-382	• A6GPP/A6PHP • VG-620 • VT-220
Programming methods		• Online programming • Offline programming (when a PC/AT is used)	• Online programming
Multitask debugging		Possible (when the debugger is used)	Impossible
Number of occupied I/O points		48 points (vacant: 16 points, special-purpose: 32 points)	48 points (vacant: 16 points, special-purpose: 32 points)

APPENDIX 2 Special-Function Module Buffer Addresses

This section gives the special-function module buffer addresses designated when data is read/written (PCRD/PCWT) from/to the buffer using the AD51H:

Manuals of the used modules give details about the buffer contents.

2.1 Analog-Digital Converter Modules

(1) A68AD analog-digital converter module

Buffer Contents	Addresses Designated from the AD51H		Addresses when a FROM/TO Command is Used	Model	
	Lower 8 bits	Higher 8 bits		A68AD	A68AD-S2
Number of channels	80H	81H	0	O	O
Average processing designation	82H	83H	1		
Channel 1 averaging time and averaging count	84H	85H	2		
Channel 2 averaging time and averaging count	86H	87H	3		
Channel 3 averaging time and averaging count	88H	89H	4		
Channel 4 averaging time and averaging count	8AH	8BH	5		
Channel 5 averaging time and averaging count	8CH	8DH	6		
Channel 6 averaging time and averaging count	8EH	8FH	7		
Channel 7 averaging time and averaging count	90H	91H	8		
Channel 8 averaging time and averaging count	92H	93H	9		
Channel 1 digital output value	94H	95H	10		
Channel 2 digital output value	96H	97H	11		
Channel 3 digital output value	98H	99H	12		
Channel 4 digital output value	9AH	9BH	13		
Channel 5 digital output value	9CH	9DH	14		
Channel 6 digital output value	9EH	9FH	15		
Channel 7 digital output value	A0H	A1H	16		
Channel 8 digital output value	A2H	A3H	17		
Data write error code	C4H	C5H	34		
A1D conversion-enabled flag	C6H	C7H	35	X	O

O: Can be designated,
X: Cannot be designated

(2) A616AD analog-digital converter module

Buffer Contents		Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Direct access	INPUT designation	10H	0H
	MX.CH designation	12H	1H
	Digital output value	14H	2H
Sampling time designation		16H	3H
Data format selection		18H	4H
Error code storage		1AH	5H
Error occurrence multiplex module CNT.number storage		1CH	6H
Unused area (unusable)		—	to
Conversion-enabled/-disabled designation	A616AD	2EH	FH
	INPUT 0 A60MX, A60MXR	30H	10H
	INPUT 1 A60MX, A60MXR	32H	11H
	INPUT 2 A60MX, A60MXR	34H	12H
	INPUT 3 A60MX, A60MXR	36H	13H
	INPUT 4 A60MX, A60MXR	38H	14H
	INPUT 5 A60MX, A60MXR	3AH	15H
	INPUT 6 A60MX, A60MXR	3CH	16H
	INPUT 7 A60MX, A60MXR	3EH	17H
Request to set data		40H	18H
Unused area (unusable)		—	—
INPUT channel digital output value		70H to 8EH	30H to 3FH
Unused area (unusable)		—	—
MX.CH. channel digital output value		210H to 30EH	100H to 17EH

(3) A84AD analog-digital converter module

Buffer Contents	Addresses Designated from the AD51H		Addresses when a FROM/TO Command is Used
	Lower 8 Bits	Higher 8 Bits	
Unused area	10	11	0
Average processing designation	12	13	1
Channel 1 averaging time and averaging count	14	15	2
Channel 2 averaging time and averaging count	16	17	3
Channel 3 averaging time and averaging count	18	19	4
Channel 4 averaging time and averaging count	1A	1B	5
Unused area (unusable)	—	—	—
Channel 1 digital output value	24	25	10
Channel 2 digital output value	26	27	11
Channel 3 digital output value	28	29	12
Channel 4 digital output value	2A	2B	13
Channel 1 internal set mode flag	2C	2D	14
Channel 2 internal set mode flag	2E	2F	15
Channel 3 internal set mode flag	30	31	16
Channel 4 internal set mode flag	32	33	17
Channel 1 temperature detected value	34	35	18
Channel 2 temperature detected value	36	37	19
Channel 3 temperature detected value	38	39	20
Channel 4 temperature detected value	3A	3B	21
Channel 1 Set Value Check Code	3C	3D	22
Channel 2 Set Value Check Code	3E	3F	23
Channel 3 Set Value Check Code	40	41	24
Channel 4 Set Value Check Code	42	43	25
Data Write Error Code	44	45	26
Analog Output-Enabled Signal Flag	46	47	27
Channel 1 Installation Module Code	48	49	28
Channel 2 Installation Module Code	4A	4B	29
Channel 3 Installation Module Code	4C	4D	30
Channel 4 Installation Module Code	4E	4F	31
Channel 1 Temperature Set Range (Offset Value)	50	51	32
Channel 1 Temperature Set Range (Gain Value)	52	53	33
Channel 2 Temperature Set Range (Offset Value)	54	55	34
Channel 2 Temperature Set Range (Gain Value)	56	57	35
Channel 3 Temperature Set Range (Offset Value)	58	59	36
Channel 3 Temperature Set Range (Gain Value)	5A	5B	37
Channel 4 Temperature Set Range (Offset Value)	5C	5D	38
Channel 4 Temperature Set Range (Gain Value)	5E	5F	39

(4) A616TD temperature-digital converter module

Buffer Contents		Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Data format selection		10H	0H
Error code storage		12H	1H
Error occurrence A60MX[ ]CONNECT number storage		14H	2H
Thermal couple type setting error channel number storage		16H	3H
Sampling time present value storage		18H	4H
Unused area (unusable)		—	to
Conversion-enabled/-disabled designation	A616TD	2EH	FH
	Multiplex module	30H to 3EH	10H to 17H
Request to set data		40H	18H
Unused area (unusable)		—	to
Disconnection detected enabled designation		50H to 5EH	20H to 27H
Unused area (unusable)		—	to
Digital output value temperature setting		70H to 8EH	30H to 3FH
Disconnection detected channel number storage		90H to 9EH	40H to 47H
Unused area (unusable)		—	to
Digital output value outside the range channel number storage		B0H to BEH	50H to 57H
Unused area (unusable)		—	to
Temperature detected value outside the range channel number storage		D0H to DEH	60H to 67H
Unused area (unusable)		—	to
INPUT channel digital output value storage		F0H to FEH	70H to 7FH
Error Compensation Value Setting		110H to 20EH	80H to FFH
Thermal couple type setting		210H to 30EH	100H to 17FH
MX.CH. channel digital output value storage		310H to 40EH	180H to 1FFH
MX.CH. channel temperature detected value storage		410H to 50EH	220H to 27FH

2.2 Digital-Analog Converter Modules

(1) A62AD digital-analog converter module

Buffer Contents	Addresses Designated from the AD51H		Addresses when a FROM/TO Command is Used
	Lower 8 Bits	Higher 8 Bits	
Channel 1 digital value	10H	11H	0
Channel 2 digital value	12H	13H	1
Channel 1 set voltage value check code	14H	15H	2
Channel 2 set voltage value check code	16H	17H	3
Channel 1 set voltage value check code	18H	19H	4
Channel 2 set voltage value check code	1AH	1BH	5

(2) A616DAI digital-analog converter module

A616DAV digital-analog converter module

Buffer Contents	Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
D-A conversion-enabled/-disabled channel	10H	0H
Analog output-enabled/-disabled channel	12H	1H
Unused (unusable)	—	2H to FH
CH. 0 digital value	30H	10H
CH. 1 digital value	32H	11H
CH. 2 digital value	34H	12H
CH. 3 digital value	36H	13H
CH. 4 digital value	38H	14H
CH. 5 digital value	3AH	15H
CH. 6 digital value	3CH	16H
CH. 7 digital value	3EH	17H
CH. 8 digital value	40H	18H
CH. 9 digital value	42H	19H
CH. A digital value	44H	1AH
CH. B digital value	46H	1BH
CH. C digital value	48H	1CH
CH. D digital value	4AH	1DH
CH. E digital value	4CH	1EH
CH. F digital value	4EH	1FH
Unused (unusable)	—	20H to 2FH
CH. 0 set value check code	70H	30H
CH. 1 set value check code	72H	31H
CH. 2 set value check code	74H	32H
CH. 3 set value check code	76H	33H
CH. 4 set value check code	78H	34H
CH. 5 set value check code	7AH	35H
CH. 6 set value check code	7CH	36H
CH. 7 set value check code	7EH	37H
CH. 8 set value check code	80H	38H
CH. 9 set value check code	82H	39H
CH. A set value check code	84H	3AH
CH. B set value check code	86H	3BH
CH. C set value check code	88H	3CH
CH. D set value check code	8AH	3DH
CH. E set value check code	8CH	3EH
CH. F set value check code	8EH	3FH

2.3 High-Speed Counter Module

(1) AD61 high-speed counter module

Buffer Contents	Addresses Designated from the AD51H		Addresses when a FROM/TO Command is Used	
	Channel 1	Channel 2	CH1	CH2
Unused area (unusable)	80H	C0H	0	32
	81H	C1H		
Preset value write (lower)	82H	C2H	1	33
Preset value write (middle)	83H	C3H		
Preset value write (higher)	84H	C4H	2	34
	85H	C5H		
Mode register	86H	C6H	3	35
	87H	C7H		
Present value read (lower)	88H	C8H	4	36
Present value read (middle)	89H	C9H		
Present value read (higher)	8AH	CAH	5	37
	8BH	CBH		
Set value read/write (lower)	8CH	CCH	6	38
Set value read/write (middle)	8DH	CDH		
Set value read/write (higher)	8EH	CEH	7	39
	8FH	CFH		

2.4 Positioning Modules

(1) AD71(S1)/AD71-S2 positioning modules

Buffer Contents		Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Data used for starting positioning on the X axis		200H to 391H	0 to 200
Error reset		392H, 393H	201
Data used for starting positioning on the Y axis		458H to 5E9H	300 to 500
Positioning data	Data for positioning on the X axis	2040H to 235FH	3872 to 4271
Positioning velocities		2360H to 267FH	4272 to 4671
Dwell times		2680H to 299FH	4672 to 5071
Positioning addresses		29A0H to 2FDFH	5072 to 5871
Positioning data	Data for positioning on the Y axis	2FE0H to 32FFH	5872 to 6271
Positioning velocities		3300H to 361FH	6272 to 6671
Dwell times		3620H to 393FH	6672 to 7071
Positioning addresses		3640H to 3F7FH	7072 to 7871
X axis parameters		3F80H to 3F9FH	7872 to 7887
Y axis parameters		3FA8H to 3FC7H	7892 to 7907
X axis zero-return data		3FD0H to 3FDDH	7912 to 7917
Y axis zero-return data		3FE4H to 3FF1H	7922 to 7928

(2) AD72 positioning module

Buffer Contents	Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Data used for starting positioning on the X axis	200H to 391H	0 to 200
Error reset	392H 393H	201
Data used for starting positioning on the Y axis	458H to 5E9H	300 to 500
Area used for monitoring	6B0H to 6BFH	600 to 607
Data for positioning on the X axis	2040H to 2FDFH	3872 to 5871
Data for positioning on the Y axis	2FE0H to 3F7FH	5872 to 7871
X axis parameters	3F80H to 3F9FH	7872 to 7891
Y axis parameters	3FA8H to 3FC7H	7892 to 7911
X axis zero-return data	3FD0H to 3FDDH	7912 to 7917
Y axis zero-return data	3FE4H to 3FF1H	7922 to 7928

(3) AD70 positioning module

Buffer Contents			Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Fixed parameters	Stroke upper limit		80H to 8BH	0 to 5
	Stroke lower limit			
	Electronic gear	Command pulse magnification numerator		
		Command pulse magnification denominator		
Variable parameters	Velocity limit		A8H to B3H	20 to 25
	Acceleration time			
	Deceleration time			
	In-position range			
	Positioning mode			
Zero return data	Zero point address		D0H to DFH	40 to 47
	Zero return velocity			
	Creep velocity			
	Travel distance after turning ON the near-zero point dog			
Positioning data	Positioning pattern		F8H to 109H	60 to 68
	Positioning address P1			
	Positioning velocity V1			
	Positioning address P2			
	Positioning velocity V2			
Control change areas	Present value change areas		120H to 133H	80 to 89
	Velocity change area			
	JOG velocity area			
	Error counter clear command			
	Analog output adjustment area			
	Velocity/position change area			
Monitoring areas	Feed position data		148H to 15FH	100 to 111
	Real present value			
	Error code (ERR.1)			
	Error code (ERR.2)			
	Error counter value			
	Travel distance after turning ON the near-zero point dog			
	Velocity/position change command			
	Velocity operation			

2.5 Position Detection Modules

(1) A61LS position detection module

Buffer Contents		Addresses Designated from the AD51H		Addresses when a FROM/TO Command is Used
		Lower 8 Bits	Higher 8 Bits	
Post-compensation present value storage area		80H	81H	0
Overflow detected flag		82H	83H	1
Underflow detected flag		84H	85H	2
Resolver rotation speed storage area		86H	87H	3
Output status storage area		88H	89H	4
Measured length storage area	(L)	8AH	8BH	5
	(H)	8CH	8DH	6
Compensation amount storage area		8EH	8FH	7
Error code storage area		90H	91H	8
Battery error detected flag		92H	93H	9
Limit switch function channel output enable setting area		94H	95H	10
Limit switch function program No.setting area		96H	97H	11
Positioning target address setting area		98H	99H	12
Positioning data setting area		9AH to 138H	9BH to 139H	13 to 44

(2) A62LS position detection module

Buffer Contents			Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Present value (scaling binary)		(L)	80H	0
		(H)	82H	1
Present value (sensor binary)		(L)	84H	2
		(H)	86H	3
All channel output status			88H	4
Program No. answer back			8AH	5
Operation mode			8CH	6
Error code			8EH	7
Limit switch output disable setting			90H	8
Program No. setting			92H	9
Positioning target stop location setting data (scaling binary)		(L)	94H	10
		(H)	96H	11
CH. 0	Number of multi-dogs		98H	12
	Dog 0 ON position setting data	(L)	9AH	13
		(H)	9CH	14
	Dog 0 OFF position setting data	(L)	9EH	15
		(H)	A0H	16
	to		to	to
	Dog 9 OFF position setting data	(L)	E6H	51
		(H)	E8H	52
CH. 1	Number of multi-dogs		EAH	53
	Dog 0 ON position setting data	(L)	ECH	54
		(H)	EEH	55
	Dog 0 OFF position setting data	(L)	F0H	56
		(H)	F2H	57
	to		to	to
	Dog 9 OFF position setting data	(L)	138H	92
		(H)	13AH	93
CH. 2 to CH. 15	Number of multi-dogs		13CH	94
	Dog 0 ON position setting data	(L)	13EH	95
		(H)	140H	96
	Dog 0 OFF position setting data	(L)	142H	97
		(H)	144H	98
	to		to	to
	Dog 9 ON position setting data	(L)	560H	664
		(H)	562H	665
	Dog 9 OFF position setting data	(L)	564H	666
		(H)	566H	667

2.6 Computer Link Module

(1) AJ71C24(-S3, S6) computer link module

Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
1000H to 11FFH	0 to FFH
1200H to 123FH	100H to 11FH (special-purpose area)
1240H to 1FFFH	120H to 7FFH

(2) AJ71C22 multi-drop link module

Buffer Contents	Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Number of access slave stations setting area	1000H	0H
Transmission sequence setting areas	1002H to 1010H	1H to 8H
Number of points of data (input from a slave station) setting area	1012H to 1020H	9H to 10H
Number of points of data (output to a slave station) setting area	1022H to 1030H	11H to 18H
(Unusable)	—	to
Data input from a slave station storage area	1040H to 107EH	20H to 3FH
Data output to a slave station storage area	1080H to 10BEH	40H to 5FH
Error number storage area	10C0H	60H
work area (Can be accessed by the user)	10C2H to 1FFEH	61H to 7FFH

(3) AJ71C21(S1) terminal interface module

Buffer Contents	Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
User's area	400H to 7FEH	0H to 1FFH
Special-purpose area	800H to 810H	200H to 208H
System area (unusable)	—	209H to 211H
Special-purpose area	824H to 830H	212H to 218H
System area (unusable)	—	219H to 21FH
User's area	840H to FDEH	220H to 5EFH

2.7 MELSECNET/MINI Master Module

(1) AJ71PT32 MELSECNET/MINI master module

Buffer Contents	Addresses Designated from the AD51H	Addresses when a FROM/TO Command is Used
Total number of I/O stations	20H	0
Retry count	22H	1
Unused area (unusable)	—	to
Batch refresh transmission data	34H to 72H	10 to 41
Unused area (unusable)	—	to
Remote I/O station card data	ACH to BAH	70 to 77
Unused area (unusable)	—	to
Accumulated faulty station detection	D4H to DAH	90 to 93
Unused area (unusable)	—	to
Faulty station detection	E8H to EEH	100 to 103
Unused area (unusable)	—	to
Communication error code	F6H	107
Error detected code	F8H	108
Unused area (unusable)	—	to
Batch refresh receive data	FCH to 13AH	110 to 141
Unused area (unusable)	—	to
Communication line error retry counter	160H	160
Retry counters	162H to 182H	161 to 192
Unused area (unusable)	—	to
Split refresh stations	214H to 236H	250 to 282
Unused area (unusable)	—	to
Split refresh transmission data	278H to 476H	300 to 555
Unused area (unusable)	—	to
Accumulated input error detection	4CCH	598
Input error station detection	4CEH	599
Split refresh receive data	4D0H to 6CEH	600 to 855

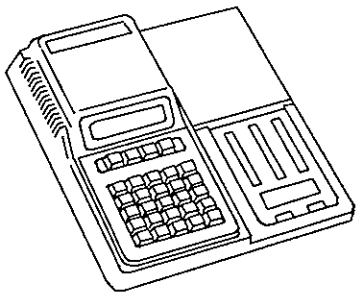
APPENDIX 3 Applicable Devices

(1) Printers

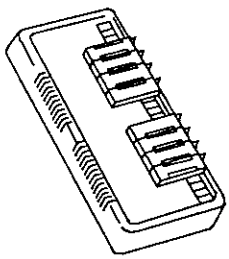
Model	M6265-1	VP-1500	HG-3000
Manufactured by	Mitsubishi Electric	SEIKO EPSON	SEIKO EPSON
Interfaces	Parallel	Parallel	Parallel
Applicable paper sizes	14 in.	15 in.	15 in.
External dimensions mm (in.) (H) x (W) x (D)	116 x 552 x 345 (4.57 x 21.75 x 13.58)	142 x 630 x 407 (5.59 x 24.80 x 16.02)	177 x 664 x 450 (6.97 x 26.14 x 17.72)
Cables used	AC30PIO-26P *1 (3m)(9.84 ft.)	AC30PIO-26P *1 (3m)(9.84 ft.)	AC30PIO-26P *1 (3m) (9.84 ft.)
	AC30PIO-20P *2 (3m) (9.84 ft.)	AC30PIO-20P *2 (3m)(9.84 ft.)	AC30PIO-20P *2 (3m) (9.84 ft.)

*1 Used when the printer is connected to a PC/AT.
*2 Used when the printer is connected to the AD51H.

(2) ROM writers



(PECKER 30 ROM writer)



(PECKER 30D adapter)

Models	PECKER 1 (PKW1100)	PECKER 30 *1 (PKW3100)
Manufactured by	AVAL	
Adapters	RX-1	B adapter (B1) or D adapter (D1)
Applicable ROMs	64KROM-W, 128KROM-W, 256KROM-W (manufactured by Mitsubishi Electric)	

*1 Use a PECKER 30 that has a system ROM version 2.4 or later.

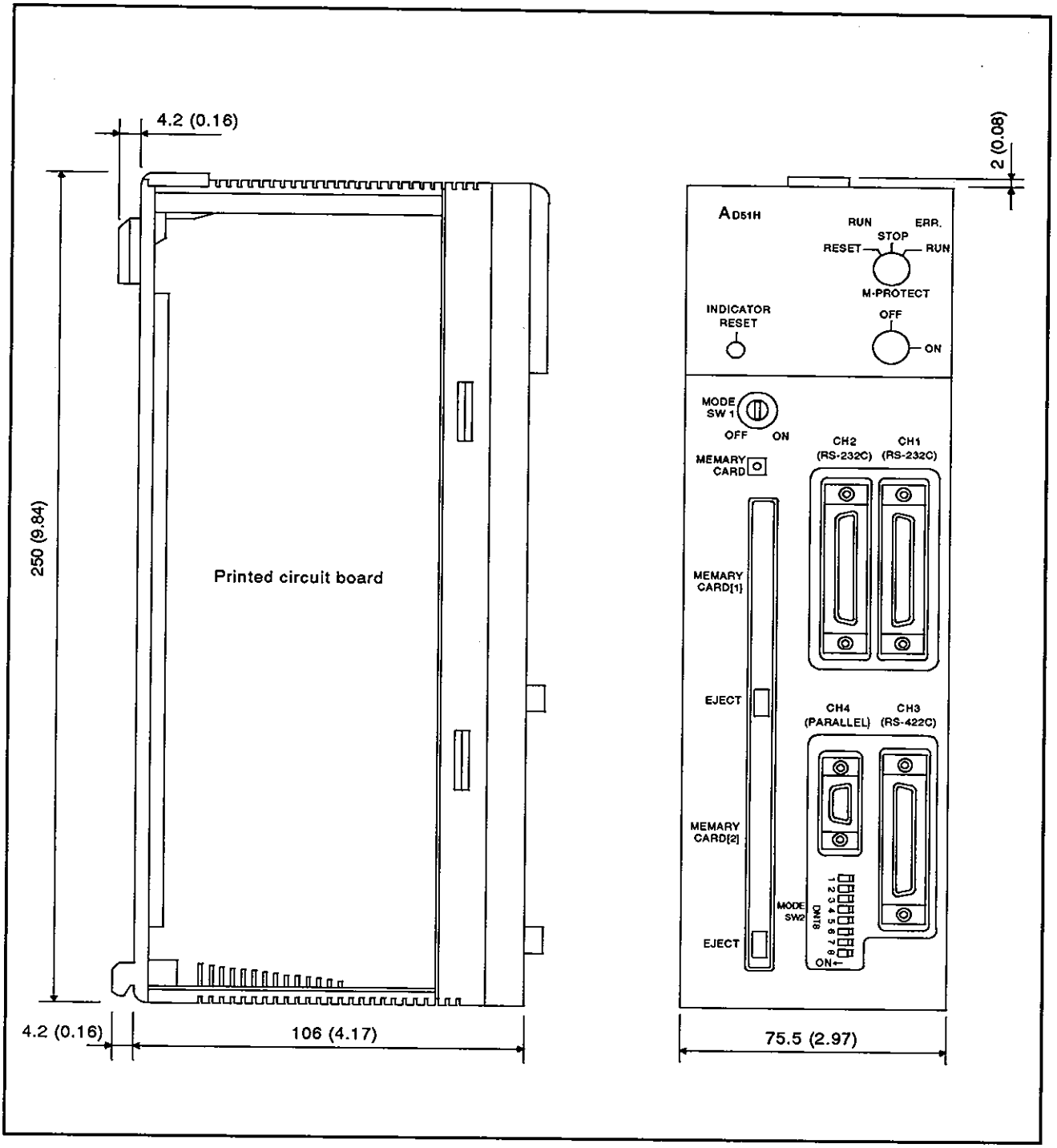
NOTES:

1 *When there is a lot of noise (extraneous signals), attach a noise filter to the ROM.*

This should be done because the ROM writer has a very little noise resistance.

2 *RS-232C cable (AC30R2) can also be used as the connection cable.*

APPENDIX 4 Dimensions



Unit: mm (in.)

IMPORTANT

The components on the printed circuit boards will be damaged by static electricity, so avoid handling them directly. If it is necessary to handle them take the following precautions.

(1) Ground human body and work bench.

(2) Do not touch the conductive areas of the printed circuit board and its electrical parts with any non-grounded tools etc.

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) 66345-A (9112) MEE Printed in Japan