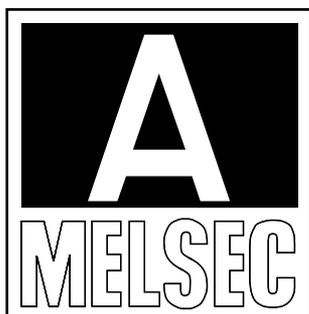
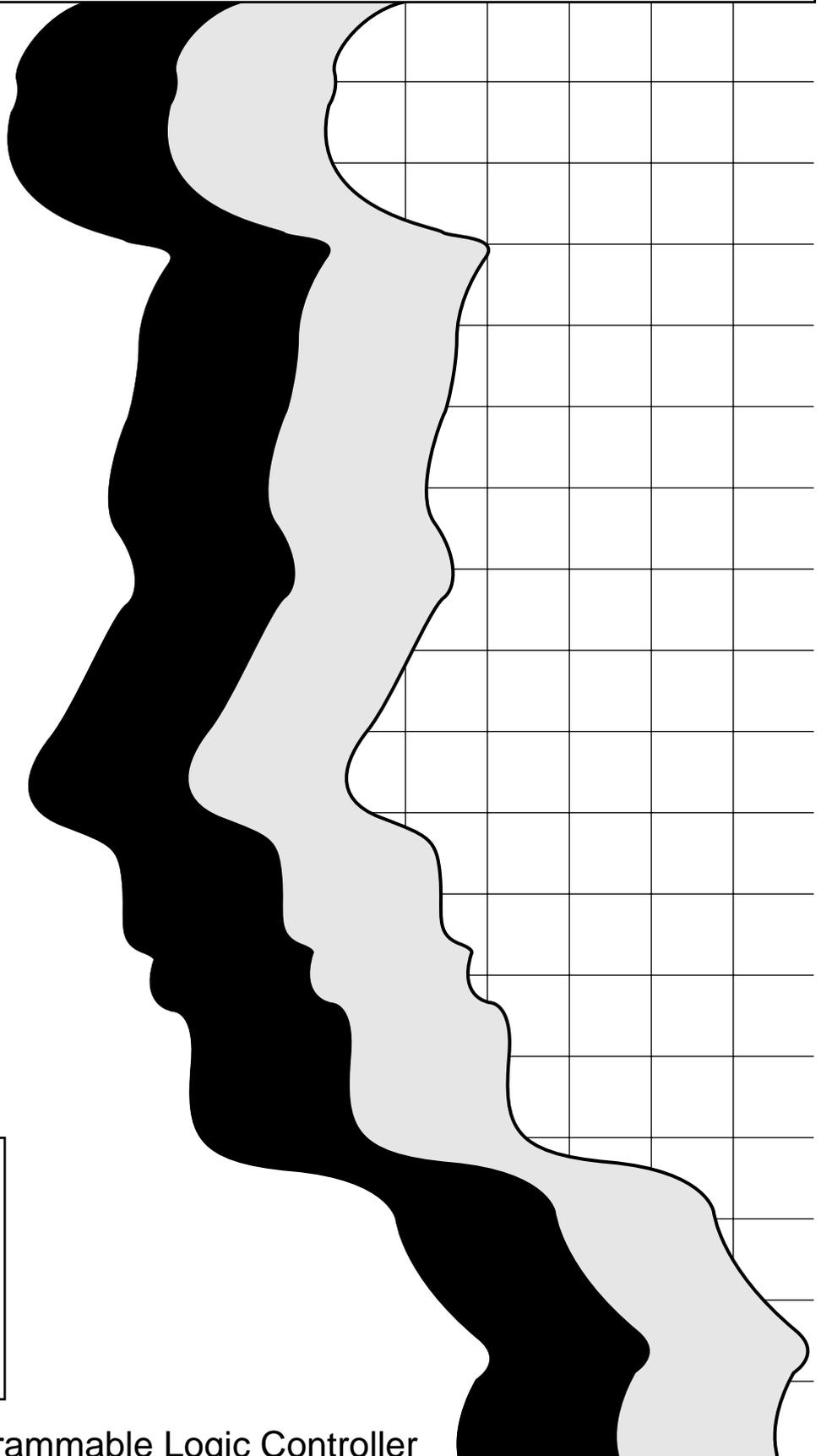


MITSUBISHI

Type A1SD51S Intelligent Communication Module

User's Manual



Mitsubishi Programmable Logic Controller

• SAFETY PRECAUTIONS •

(Always read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.

The instructions given in this manual are concerned with this product. For the safety instructions of the programmable controller system, please read the user's manual of the CPU module to use.

In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".



Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.



Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Note that the  CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

[DESIGN PRECAUTION]

DANGER

- Please refer to the manual for each station concerning the operating status of each station when communications errors occur in the station. There is danger of accidents due to wrong outputs or wrong operations.
- When connecting a peripheral device to the CPU module or performing control of a PLC which is being run through a BASIC program, configure an interlock circuit in the sequence program so that the system overall is operating on the safe side at all times. Also, when performing other control (program modifications, changing the operating status (status control)) of a PLC that is currently running, read the manual thoroughly and proceed only after taking adequate safety precautions. Particularly in cases where the above control is performed toward PLC in remote locations from the opposite party's device, there may be occasions where it is impossible to respond immediately to trouble on the PLC side resulting from data communications errors. At the same time as you are configuring an interlock circuit in the sequence program, decide methods for the system to use to treat data communications errors between the opposite party's device and the PLC CPU.

[DESIGN PRECAUTION]

DANGER

- Do not write data to the "System Area" in the special function module's buffer memory. Also, do not turn the output for signals for "Use Prohibited" signals among the output signals from the PLC CPU to the special function module (ON). If data are written to the "System Area" or output in response to "Use Prohibited" signals, there is danger that the PLC system will malfunction.

CAUTION

- Do not bunch the control wires or communication cables with the main circuit or power wires, or install them close to each other. They should be installed 100mm (3.94 in.) or more from each other. Not doing so could result in noise that may cause malfunction.
- If a BASIC program is registered in the EEPROM in the module, do not perform a power OFF or PLC CPU reset operation at the station where the module is installed during registration. If a power OFF or PLC CPU reset operation is performed at the station where the module is installed during registration, the data contents in the EEPROM will be indefinite and it will be necessary to reset the setting values, etc. in the buffer memory and register them again in the EEPROM. It could also cause the module to break down or malfunction.

[INSTALLATION PRECAUTIONS]

CAUTION

- Use the PLC in an environment that meets the general specifications contained in the user's manual of the CPU module to use. Using this PLC in an environment outside the range of the general specifications may cause electric shock, fire, malfunction, and damage to or deterioration of the product.
- Fully insert the fixing projection on the module bottom into the fixing hole in the base unit to mount the module and tighten the fixing screws with the specified torque. Failure to do so may cause a malfunction, failure or drop of the module.

[INSTALLATION PRECAUTIONS]

CAUTION

- Tighten the screws within the range of specified torque.
If the screws are loose, it may cause the module to fallout, short circuits, or malfunction.
If the screws are tightened too much, it may cause damage to the screw and/or the module, resulting in fall out, short circuits or malfunction.
- Before mounting/dismounting the module, be sure to shut off all phases of external power supply used by the system.
Failure to do so may cause product damage.
- Do not directly touch the conductive area or electronic components of the module.
Doing so may cause malfunction or failure in the module.

[WIRING PRECAUTIONS]

CAUTION

- If the module is turned on and operated after installation or wiring operations, etc., be sure install the terminal cover provided with this product. If the terminal cover is not installed, there is danger of malfunction.
- When connecting wires to the connector used for external connections, be sure to crimp, pressure weld or solder the wires correctly using the tool specified by the manufacturer.
If connections are not perfect, it could result in short circuits, fire or malfunction.
- Securely insatll the connector to the module.
- Be sure to fix communication cables leading from the module by placing them in the duct or clamping them.
Cables not placed in the duct or without clamping may hang or shift, allowing them to be accidentally pulled, which may cause a module malfunction and cable damage.
- When connecting cables, be sure to do so correctly after confirming the type of interface you are connecting to. If connection is made to a different interface or if wiring is faulty, it could cause the module or external device to break down.
- Tighten the terminal screws within the range of specified torque.
If the terminal screws are loose, it may result in short circuits or malfunction.
If the screws are tightened too much, it may cause damage to the screw and/or the module, resulting in the module falling out, short circuits or malfunction.
- When disconnecting the communications cable or power cable that is connected to the module, do not disconnect it by grasping the cable with your hand and pulling it.
Disconnect cables with connectors attached by taking hold of the connector at the connection with the module and pulling the connector. For cables connected to a terminal block, remove the cable after loosening the terminal block screws.
If the cable is pulled while it is connected to the module, it could cause malfunction or damage the module or the cable.

[WIRING PRECAUTIONS]

CAUTION

- Be careful not to let foreign objects such as swarf or wire chips get inside the module.
They may cause fires, failure or malfunction.

[STARTING AND MAINTENANCE PRECAUTIONS]

CAUTION

- Do not disassemble or modify the each module.
Doing so could cause failure, malfunction injury or fire.
- Before mounting/dismounting the module, be sure to shut off all phases of external power supply used by the system.
Failure to do so may cause module failure or malfunctions.
- Do not touch the connector while the power is on.
Doing so may cause malfunction.
- Before cleaning up and retightening terminal screws and module mounting screws, be sure to shut off all phases of external power supply used by the system.
Not doing so may cause failure or malfunction of the module.
If the screws are loose, it may cause the module to fallout, short circuits, or malfunction.
If the screws are tightened too much, it may cause damages to the screws and/or the module, resulting in fall out, short circuits or malfunction.
- Always make sure to touch the grounded metal to discharge the electricity charged in the electricity charged in the body, etc., before touching the module.
Failure to do say cause a failure or malfunctions of the module.

[OPERATING PRECAUTIONS]

CAUTION

- When performing control (in particular, changing data, changing a program or changing the operation status (status control)) of the PLC during operation using a BASIC program, do so only after reading the user's manual thoroughly and taking adequate safety precautions.
If there are errors when changing data, changing a program or in status control, it could result in system malfunction, or cause mechanical damage or accidents.

[DISPOSAL PRECAUTIONS]

CAUTION

- When disposing of this product, treat it as industrial waste.

REVISIONS

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

Print Date	*Manual Number	Revision
May, 1995	IB (NA) 66551-A	First edition
Dec., 2003	IB (NA) 66551-B	<p>Addition</p> <p>SAFETY PRECAUTIONS, Compliance with the EMC and Low Voltage Directives, WARRANTY</p> <p>Partial Correction</p> <p>CONTENTS, About Manuals, Section 1.2, 2.1, 2.2, 2.3, 3.1.1, 3.2.1, 3.2.3, 3.2.5, 3.3, 3.4, 3.5, 3.5.3, 3.7.1, 3.7.2, 3.7.3, 4.1, 4.2, 4.5, 4.5.2, 5.3.2, 5.3.3, 5.3.4, 5.4.1, 6.1, 6.4, Appendix 1</p>
Apr., 2005	IB (NA) 66551-C	<p>Partial Correction</p> <p>SAFETY PRECAUTIONS, About Manuals, Compliance with the EMC and Low Voltage Directives, Section 2.2, 3.1.1, 3.2.3, 3.2.4, 3.2.5, 5.1, Appendix 1</p>
Oct., 2006	IB (NA) 66551-D	<p>Partial Correction</p> <p>SAFETY PRECAUTIONS, About Manuals, Conformation to the EMC Directive and Low Voltage Instruction, Section 1.1, 1.2, 2.1, 2.3, 3.1.1, 3.2, 3.2.1 to 3.2.5, 3.3 to 3.5, 3.5.2 to 3.5.8, 3.6, 3.7.1 to 3.7.3, 3.8, 3.9.1 to 3.9.3, 3.10.1 to 3.10.4, 3.12.1, 3.12.2, 4.1, 4.7.3, 4.7.4, 5.1, 5.3.1, 5.3.5, 6.4, Appendix 1</p> <p>Delection</p> <p>Section 3.5.1</p>

CONTENTS

1. GENERAL DESCRIPTION	1-1-1-6
1.1 Outline of A1SD51S Operations	1-2
1.2 Features	1-3
2. SYSTEM CONFIGURATIONS	2-1-2-5
2.1 Overall Configurations	2-1
2.2 Applicable CPU Modules	2-3
2.3 List of System Devices	2-5
3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S	3-1-3-52
3.1 Performance Specifications	3-1
3.1.1 Performance specifications list	3-1
3.2 Specifications of Interfaces with External Devices	3-3
3.2.1 Interface uses	3-3
3.2.2 Available baud rates	3-4
3.2.3 RS-232 interface (CH1: Channel 1)	3-5
3.2.4 RS-232 interface (CH2: Channel 2)	3-6
3.2.5 RS-422/485 interface (CH3: Channel 3)	3-7
3.3 AD51H-BASIC Commands	3-9
3.4 A1SD51S Internal Memory and Memory Map	3-16
3.5 A1SD51S Internal Memory Areas	3-18
3.5.1 Program area: Not backed up by a battery	3-19
3.5.2 Buffer for communications with a PC CPU: Not backed up by a battery	3-20
3.5.3 Common memory shared between tasks	3-23
3.5.4 Extension register (ED)	3-25
3.5.5 Special registers (ED9000 to ED9127)	3-28
3.5.6 Extension relay (EM)	3-30
3.5.7 Special relays (EM9000 to EM9127)	3-32
3.6 Receive Buffer	3-33
3.7 I/O for a PC CPU	3-34
3.7.1 I/O (X, Y) list	3-34
3.7.2 Details on input devices (X)	3-35
3.7.3 Details of output devices (Y)	3-38
3.8 Relationship between A1SD51S Start Conditions and EEPROM Execution Program Areas	3-41
3.9 A1SD51S Operation when the Power is Turned ON	3-42
3.9.1 In the programming mode	3-42
3.9.2 In the multitask debug mode	3-43
3.9.3 In the run mode	3-44
3.10 Executing Tasks (Created BASIC Programs)	3-45
3.10.1 Task execution enable conditions and start conditions	3-45
3.10.2 Starting by turning the power on or by reset (Start condition: START)	3-46

3.10.3	Starting by an interrupt from the PC CPU (Start condition: IT)	3-48
3.10.4	Starting by a start request from another task (Start condition: BOOT, ON)	3-49
3.11	Execution of BASIC Programs by Multitasking	3-50
3.11.1	Multitasking	3-50
3.12	BASIC Program Priority	3-51
3.12.1	Synchronization of execution between BASIC programs (event control)	3-51
3.12.2	Sharing of devices (sources) by multitasking	3-52
4.	A1SD51S FUNCTIONS	4-1-4-24
4.1	Function List	4-1
4.2	Communications with a Console or Terminal	4-2
4.3	Communications with a Console	4-3
4.4	Communications with a Terminal	4-6
4.5	Printing out Data at a Printer	4-10
4.5.1	Printing out data at a printer connected to an interface not set as a console interface	4-11
4.5.2	Printing out data at a printer connected to a console (PC/AT)	4-12
4.6	Communications with a Device Other than the Console, Terminal or Printer	4-14
4.6.1	Transmitting data	4-14
4.6.2	Receiving data	4-15
4.7	Communications with the PC CPU or a Special-Function Module	4-16
4.7.1	ON/OFF data communications between the A1SD51S and the PC CPU using general-purpose I/O (X/Y) signals	4-17
4.7.2	Communicating 16-bit data using the buffer memory	4-18
4.7.3	Reading data from/writing data to the PC CPU device memory	4-19
4.7.4	Reading data from/writing data to the PC CPU extension file register	4-20
4.8	Reading Data from/Writing Data to the Buffer Memory of a Special-Function Module	4-21
4.9	Using Clock Data	4-22
4.10	Data Communications between Tasks	4-23
4.10.1	ON/OFF data communications by the extension relay (EM)	4-23
4.10.2	Data communications by the extension register (ED)	4-24
4.10.3	Data communications using the common memory	4-24
5.	NOMENCLATURE AND SETTINGS	5-1-5-13
5.1	Nomenclature	5-1
5.2	Indications of Indicator LEDs	5-2
5.3	Switch Settings	5-3
5.3.1	Setting the A1SD51S operation mode (BASIC program execution, multitask debugging or programming)	5-3
5.3.2	Settings when a console and a debugger are used	5-5
5.3.3	Settings for stopping and continuing BASIC programs with the [Break] or [Ctrl] + [C] keys	5-7
5.3.4	A1SD51S operation settings when the PC CPU is reset	5-8
5.3.5	Setting the time during which access from the PC CPU is possible after the A1SD51S is reset	5-10
5.3.6	Setting backup area clear	5-11
5.3.7	Setting EEPROM write protect	5-11

5.4	Loading and Installation	5-12
5.4.1	Precautions for handling	5-12
5.4.2	Installation conditions	5-12
5.5	Precautions for Wiring	5-13
6.	A1SD51S STARTUP PROCEDURE	6-1-6-4
6.1	Startup when a PC/AT is Used as the Console	6-1
6.2	Startup when a VT-382 is Used as the Console	6-2
6.3	Startup when a VG-620 is used as the Console	6-3
6.4	Startup when a PC/AT is Used as the Console with general-purpose editor	6-4
APPENDICES	A-PP1-APP-3
APPENDIX1	FUNCTIONAL COMPARISON BETWEEN A1SD51S, AD51H-S3, AD51H AND AD51 (S3)	APP-1
APPENDIX2	OUTLINE DRAWING	APP-3

About Manuals

The following manuals are also related to the A1SD51S intelligent communication module.

Related Manuals

Manual Name	Manual No. (Model Name Code)
AD51H-BASIC Programming Manual (Command) (Corresponds to the QD51, QD51-R24, A1SD51S, AD51H-S3.) Explains the commands, how to use the functions and the specifications of AD51H-BASIC. (Sold separately)	SH-080090 (13JF63)
AD51H-BASIC Programming Manual (Debug and Compile) (Corresponds to the QD51, QD51-R24, A1SD51S, AD51H-S3.) Explains concerning debugging of AD51H-BASIC, multitasking settings, and compilation methods. (Sold separately)	SH-080091 (13JF64)

Conformation to the EMC Directive and Low Voltage Instruction

For details on making Mitsubishi PLC conform to the EMC directive and low voltage instruction when installing it in your product, please see Chapter 3, "EMC Directive and Low Voltage Instruction" of the User's Manual (Hardware) for the CPU module used or the PLC CPU supplied with the base unit.

The CE logo is printed on the rating plate on the main body of the PLC that conforms to the EMC directive and low voltage instruction.

By making this product conform to the EMC directive and low voltage instruction, it is not necessary to make those steps individually.

1. GENERAL DESCRIPTION

This manual explains the system configurations, performance specifications, and functions of the A1SD51S Intelligent communication module (called the A1SD51S hereafter).

When applying the following program examples to the actual system, make sure to examine the applicability and confirm that it will not cause system control problems.

The A1SD51S has the following functions.

(1) Functions with BASIC programs

- Sub-CPU function: Performs complex calculations and functional calculus.
- Monitor display function: Displays production output status, operation status, and details of faults.
- Key input function: Allows the user to input data such as production schedules, production quantities, operations, and set values.
- Printer function: Allows the user to print data such as production plans, actual production, daily reports, details of faults, planning data, inspection results, and test records.
- Data input function: Allows the user to enter data by using devices such as bar-code readers and magnetic card readers.
- External device connection function: Allows data communications using a BASIC program, with the computer connected to the A1SD51S via the RS-232 or RS-422/485 interface.
- Clock function: Allows the user to access the clock data of a PC CPU.

(2) Offline programming function

Allows the user to create and modify programs, save created programs in the user's floppy disk, and print programs with a peripheral device alone.

(3) Online programming function

Allows the creation, execution, and modification of programs from a console.

(4) Multi-task debugging function

Allows debugging while a BASIC program is being executed.

1.1 Outline of A1SD51S Operations

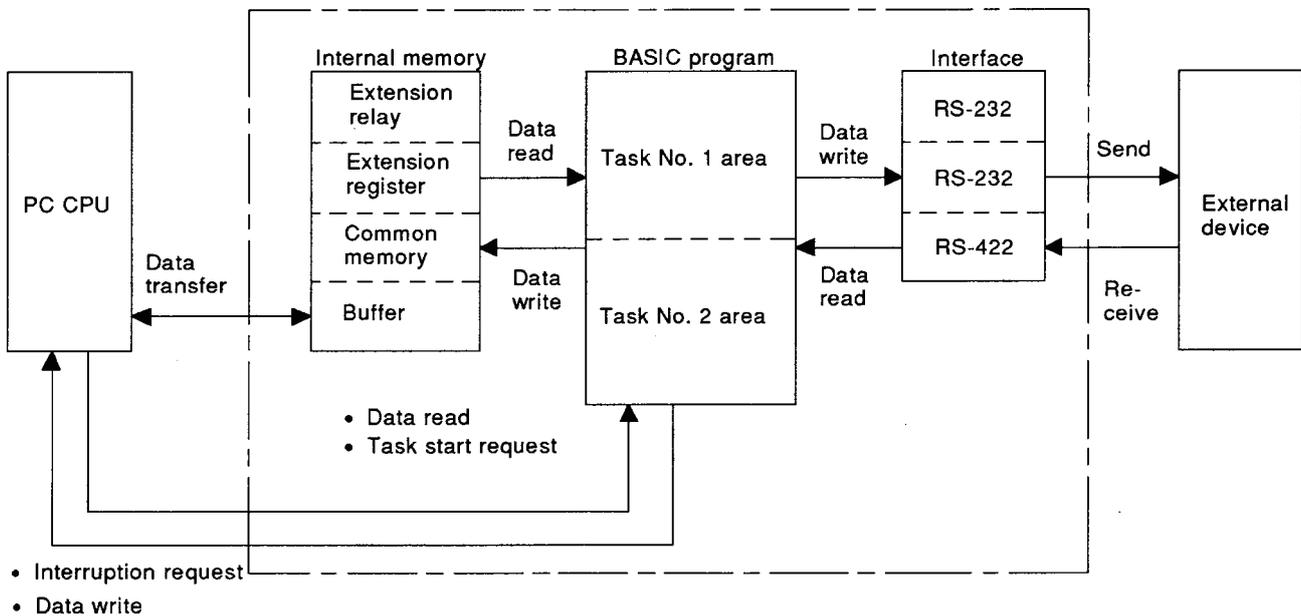
The A1SD51S operates in accordance with a BASIC program created by the user.

AD51H-BASIC can be used with the A1SD51S, and can execute multitask processing of up to two tasks.

In addition to various calculations, the following operations can be executed using created BASIC programs:

- Data communications with an external device
- Data transfer from/to a PC CPU

The following shows the operations when performing data communications with an external device and data transfer from/to a PC CPU.



(1) Data communications with an external device

The A1SD51S has two channels for the RS-232 interface and one channel for the RS-422/485 interface.

The data for production orders and actual production can be sent to or received from a console or an external device such as a personal computer connected to the RS-232 or RS-442/485 interface without using a protocol.

In addition, actual production data can be printed at a printer connected to the RS-232 interface.

(2) Data transfer from/to a PC CPU

Through the execution of PC access commands (PCRD/PCWT), the PC CPU device data can be read and written.

With the PC access commands, data can also be written to and read from the buffer memory of a special function module.

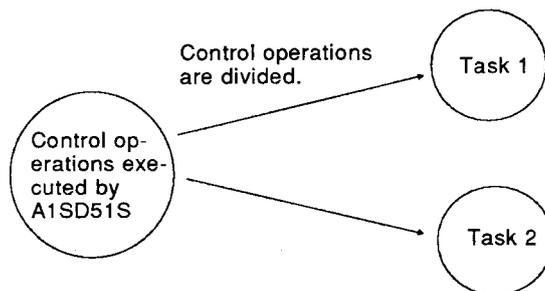
1.2 Features

Systems in which the A1SD51S is used have the following features.

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

Since multitask processing is possible, BASIC programs can be created by dividing the control operations to be executed by the A1SD51S into processing units.

The tasks required for each control operation can be started according to the contents of processing, and data communication between tasks and the timing of task execution can be controlled.



Three or more BASIC programs can be created. However, the maximum number of tasks (BASIC programs) that can be executed simultaneously is two.

(2) Data communications with a connected device are possible.

<Data communications with an external device>

Data communications with an external device connected to the RS-232 or the RS-442/485 interface can be controlled without using a protocol.

<Data communications with a console or terminal>

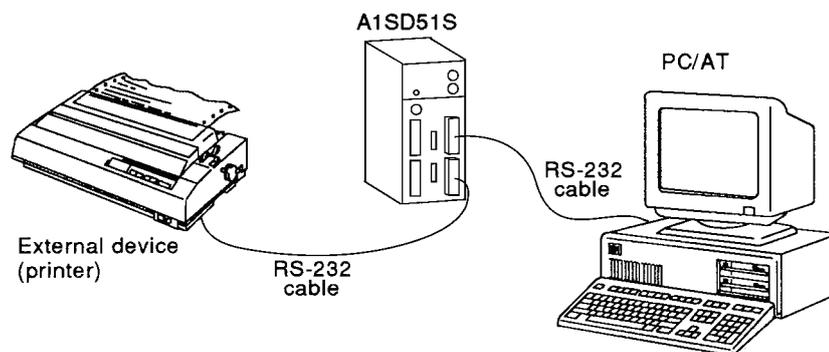
The following data communications can be performed with a console or terminal connected to the RS-232 or RS-422/485 interface.

- Data display to a console or terminal
- Data input to a console or terminal

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 at a printer connected to the RS-232 interface. Printers available on the market can be used.



(3) Data communications with a PC is enabled.

Data communications can be executed with a PC CPU to which the A1SD51S is connected, or a PC CPU in a MELSECNET(II), MELSEC-NET/B, or MELSECNET/10 network.

The following data communications can be executed with a PC CPU.

- (a) Read and write of device data in the PC CPU (max. 64 words/time)
- (b) Read and write of buffer data in the special function module (max. 128 bytes/time)
- (c) Remote RUN/STOP of a PC CPU
- (d) Read and write of sequence programs and parameters
- (e) Interruption to a PC CPU

(4) File management is enabled.

Sequential files and random files can be managed by using the floppy disk or the hard disk of the PC/AT.

<Sequential file>

It is efficient to use sequential files when reading or writing data sequentially.

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

<Random file>

Data can be read from or written to any part of a random file.

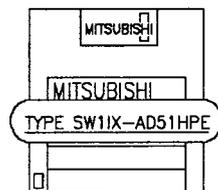
This type of file is suitable for storing large-size inventory data, display data for the operator interface, and master data.

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

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

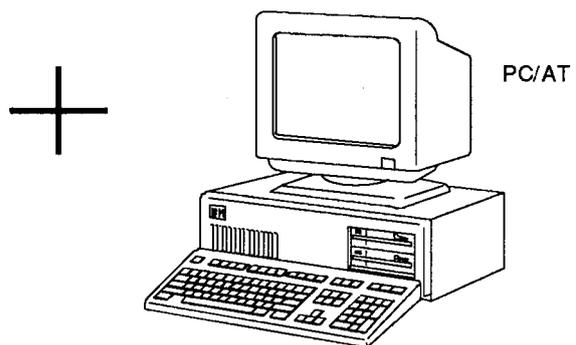
BASIC programs can be edited by using the PC/AT alone.

An edited program cannot be executed by using offline programming.



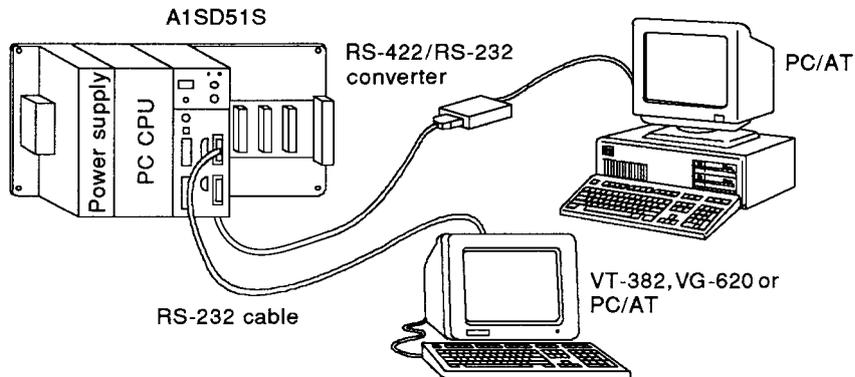
System floppy disk
(Installed in the PC/AT.)

or commercially available editor
(Windows® HyperTerm)



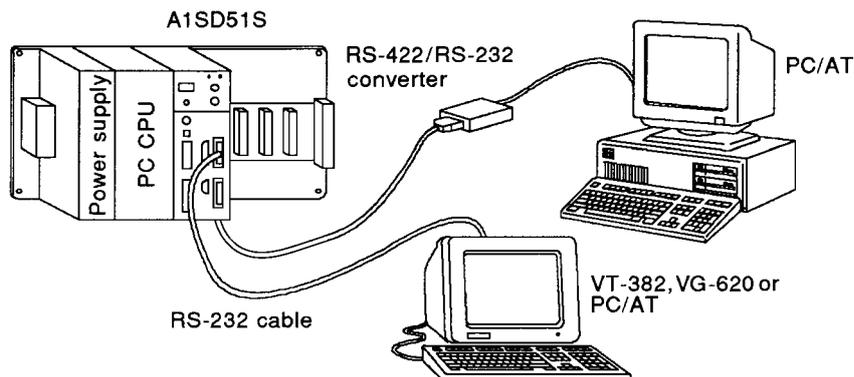
(6) Online programming is enabled.

BASIC programs can be edited and debugged by connecting an PC/AT or device available on the market to the A1SD51S and setting it as a console (see Section 2.1).



(7) Multitask debugging is enabled.

Debugging during multitask processing can be executed by connecting the PC/AT or a device available on the market to the A1SD51S and setting it as the debugger (see Section 2.1).



Multitask debugging is executed by inputting debug commands from the debugger.

The following operations can be executed by using debug commands:

- 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 A1SD51S
- Display of the states of events, and use of message ports, that are common to two or more BASIC programs

- (8) Interpreter BASIC and compiler BASIC can be both used.

Basic programs created with interpreter BASIC can be compiled with BASIC compiler to speed up the processing of the created BASIC programs.

Interpreter BASIC: With interpreter BASIC, the A1SD51S converts BASIC program commands one by one into a machine language while executing the program.

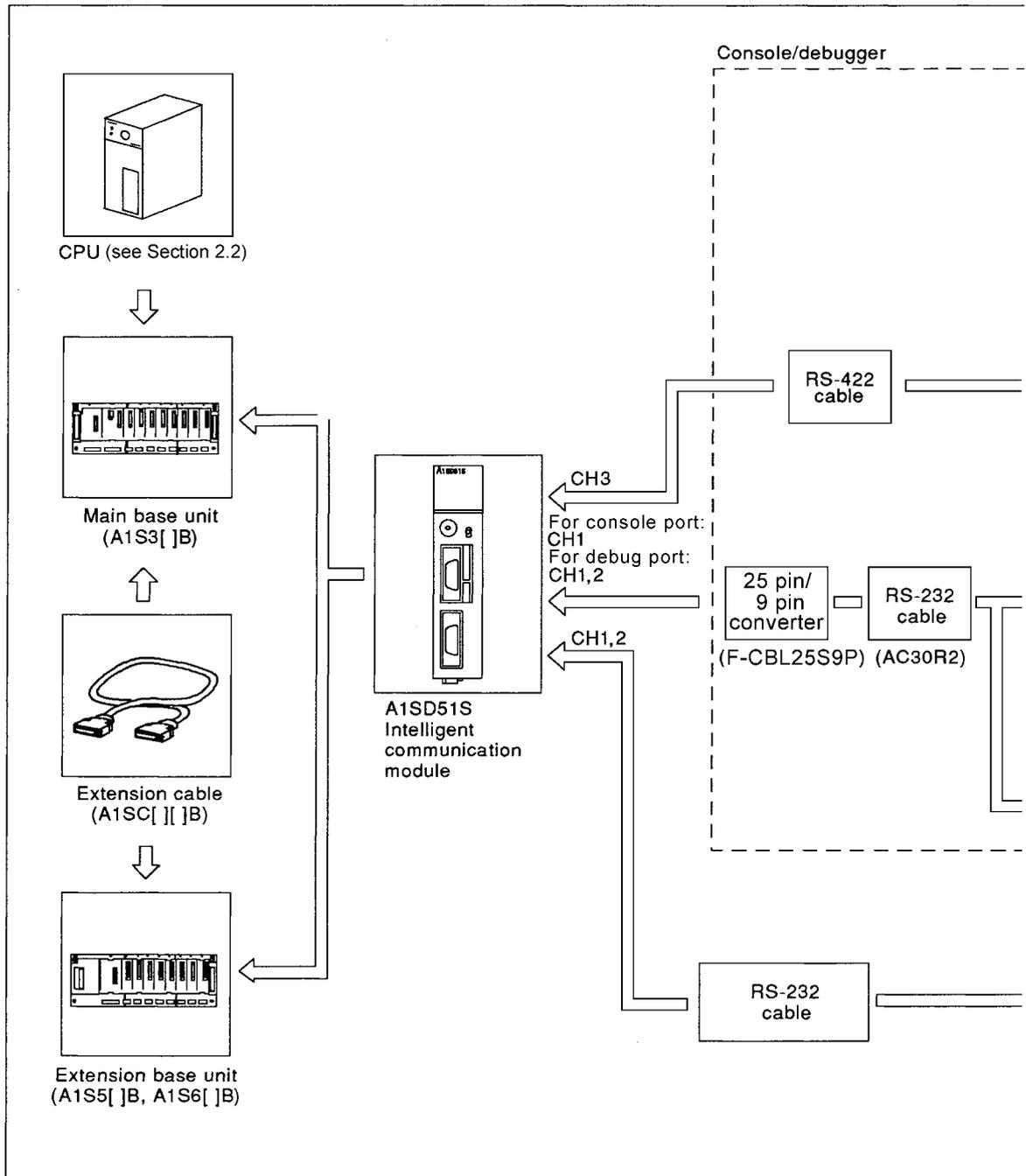
Compiler BASIC: With compiler BASIC, since the commands are compiled into a machine language when the program is completed, the A1SD51S executes the BASIC program already in a machine language without interpretation.

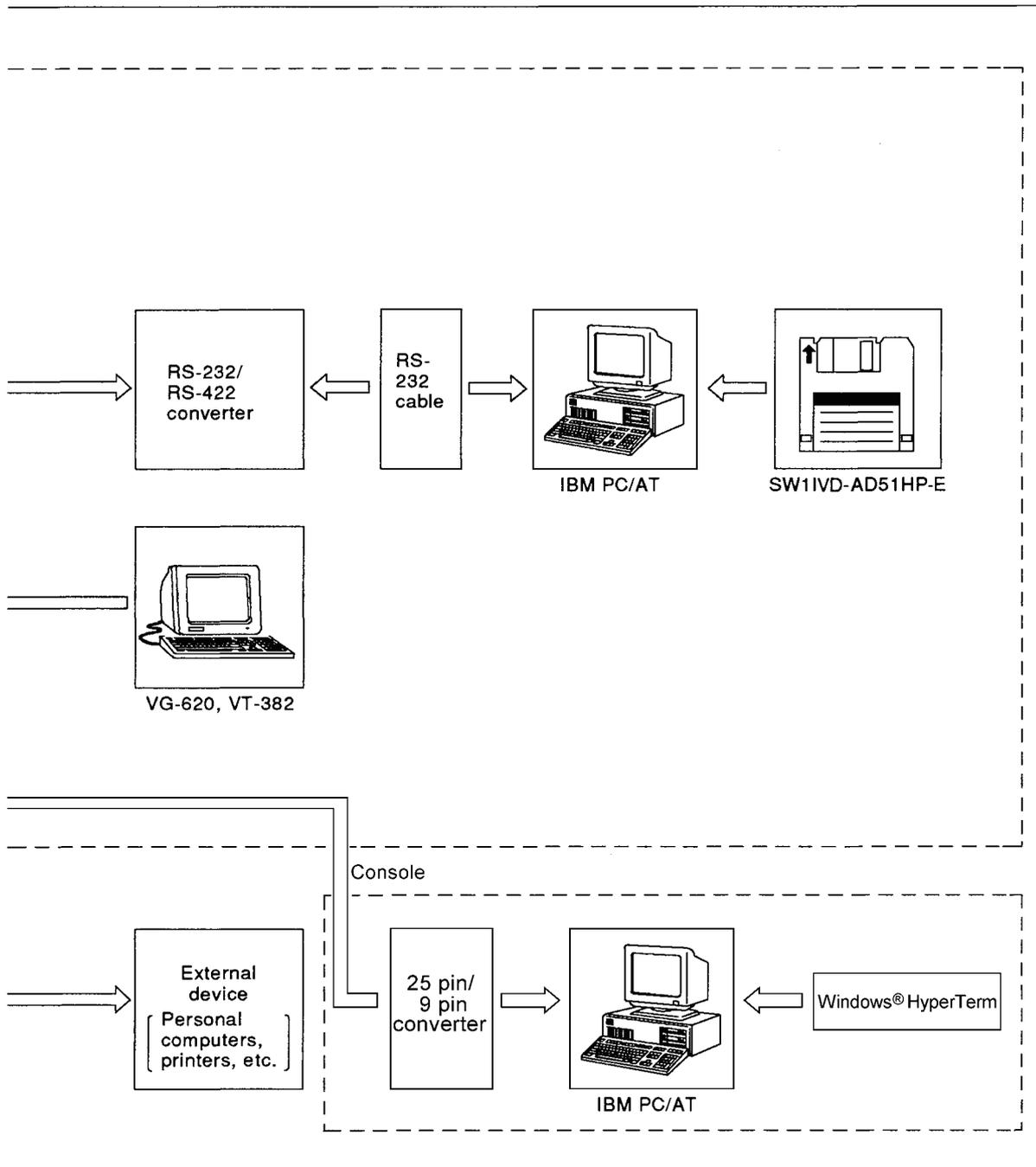
2. SYSTEM CONFIGURATIONS

This section explains the system configurations and system devices for use with the A1SD51S.

2.1 Overall Configurations

This diagram shows the overall system configuration including the A1SD51S.





2.2 Applicable CPU Modules

The A1SD51S can be used with the CPUs indicated below.

(1) Applicable CPU modules and the number of the A1SD51S modules

The table below shows the PC CPU modules that are compatible with the A1SD51S and the number of A1SD51S modules that can be connected to the PC CPU modules.

A1SD51S installation station	Application module	Number of modules that can be installed	Remarks
CPU module	A1S(-S1), A1SJ A1SH, A1SJH A2S(-S1) A2SH(-S1)	2	When using both the AnS series and A series special function modules GOT series shown below, the total number of modules that can be installed includes the number of these modules used and connected. <ul style="list-style-type: none"> · A1SJ71UC24-R2 · A1SJ71UC24-R4 · A1SJ71UC24-PRF · A1SD51S · A1SJ71E71N3-T · A1SJ71E71-B2/B5-S3 · A1SJ61BT11: Intelligent mode only · AD51(-S3) · AD51FD(-S3) · AD57G(-S3) · AJ71C21(-S1) For only the BASIC program · AJ71C23(-S3) · AJ71UC24 · AJ71P41 · AJ71E71N3-T · AJ71E71-S3 · AJ61BT11: Intelligent mode only · AOJ2C214-S1 · GOT series (Only when bus connection)
	A1SCPU-C24	1	<ul style="list-style-type: none"> · A1SJ71C24-R2 · A1SJ71C24-R4 · A1SJ71C24-PRF · A1SJ71E71N-T/B5/B2/B5T · A1SJ71E71-B2/B5
	A2AS(-S1)	6	<ul style="list-style-type: none"> · AD51H(-S3)
	Q2AS(-S1)*1 Q2ASH(-S1)*1		<ul style="list-style-type: none"> · AJ71C24(-S3/-S6/S8)
	Q02-A Q02H-A Q06H-A		<ul style="list-style-type: none"> · AJ71E71
MELSECNET/10 (Remote station)	A1SJ72QLP25 A1SJ72QBR15 A1SJ72QLR25	2	<ul style="list-style-type: none"> * When using a computer link module (AJ71UC24, etc.) as a multiple drop link module, it is not included in the above restrictions on the number of modules that can be installed. Multiple modules can be installed within the PC CPU's I/O points. * When using a CC-Link system master/local module (AJ61BT11, etc.) in the I/O mode, it is not included in the above restrictions on the number of modules that can be installed. Multiple modules can be installed within the PC CPU's I/O points.

*1 Accessible to only internal devices in the range of A2ASCPU(-S1).

(2) Applicable base units

The A1SD51S can be loaded to any slot of a main base unit or extension base unit with the following two exceptions:

- (a) The power supply capacity may be insufficient to load the A1SD51S to an extension base unit with no built-in power supply (A1S5[]B Extension Base Unit).

If at all possible, avoid loading an A1SD51S module on this type of extension base unit.

- (b) If an A1SD51S module has to be loaded on an extension base unit with no built-in power supply, consider (a) the current 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 the CPU module to be used.)

(3) Module installation position

Install A1SD51S on the final slot of the main base or the extension base. (Set the right side of A1SD51S to no module or blank cover (A1SG60).)

A1SD51S uses the super-capacitor (capacitor) for backup.

When the super-capacitor is used at a high temperature, the backup time becomes short.

To reduce the temperature of the super-capacitor, be sure to observe the installation position.

2. SYSTEM CONFIGURATIONS

MELSEC-A

2.3 List of System Devices

Table 2.1 shows the devices required to use the A1SD51S.
For details on each device, refer to its manual.

(1) Devices that can be connected to the A1SD51S

Table 2.1 shows the MELSEC-A series standard device, recommended devices, and other devices which can be connected to the A1SD51S.

Table 2.1 List of System Devices

Product Name	Model	Remarks		
Intelligent communication module	A1SD51S	• Main module		
Personal computer	IBM PC/AT	• Used as a console, a debugger, or a terminal		
General-purpose consoles	VG-620 VT-382			
Software package	SW11VD-AD51HP-E	• PC/AT compatible system software for starting up the A1SD51S		
Communication software	Windows® HyperTerm	• Terminal emulator software when PC/AT is used as the console.		
General-purpose editors	MIFES FINAL EDLIN	• Used for offline programming.		
User's floppy disk	_____	• A 2HD disk for storing data		
Printer	ESC/P is supported.	• Used to print out BASIC programs and data.		
Converters	EL-LINE-II	• An RS-232/RS-422 converter when using the PC/AT		
	KS-10P			
	FA-CBL25S9P	• 25 pin/9 pin converter for AC30R2	For A1SD51S side	
	FA-CBL25S9S		For PC/AT side	
Connection cable	_____	Classification	Cable length	Connection device
	AC30R2	• RS-232 cable	3 m	VG-620/VT-382
				PC/AT
				Printer

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

This section gives the performance specifications of the A1SD51S and explains each item.

3.1 Performance Specifications

This section describes the performance specifications and internal memory map of the A1SD51S.

3.1.1 Performance specifications list

Table 3.1 shows the performance specifications of the A1SD51S.

Table 3.1 Performance Specifications of the A1SD51S.

Item		Specifications
Program language		AD51H-BASIC
Number of tasks		Max. two tasks
Conditions for starting a task		<ul style="list-style-type: none"> • Started by power ON • Started by an interruption from the PC CPU • Started by the start command from another task
Internal memory	Program memory	Max. 64 K bytes/two tasks (Task size can be set to 16 K, 32 K, 48 K, 64 K bytes)
	Common memory	8 K bytes
	Extension register	1024 points (2 K bytes)
	Extension relay	1024 points
General-purpose I/O		Input: 27 points Output: 23 points
Buffer		3 K words (6 K bytes)
Built-in interface	RS-422/485 Interface	Conforms to RS-422 Channel 3 Used with a connector Transmission distance: within 500 m
	RS-232 Interface	Conforms to RS-232 Channels 1 and 2 Used with a connector Transmission distance: within 15 m
Memory backup		Backup capability provided (common memory, extension relay, extension register) *1
Writing a user program to the ROM		Disabled (However, a built-in EEP-ROM is installed in the body of the A1SD51S.) *2
Console		<ul style="list-style-type: none"> • PC/AT • VG-620 (manufactured by Victor Data Systems) • VT-382 (manufactured by Japan Digital Equipment Inc.)
Number of I/O points occupied		32 points (I/O assignment: special: 32 points)
5VDC Internal current consumption		0.4 A
External dimensions		130 (5.12) × 34 (1.39) × 93.6 (3.69) mm(in.)
Weight		0.3kg (0.66lb)

POINT

*1 Precautions for data backup

When data is backed up using the common memory, extended register (ED) and extended relay (EM), be careful for the following points. When data is cleared at the A1SD51S start-up (SW9 is ON), the installation position is not limited.

(1) Module installation position

Install A1SD51S at the final slot of the main base or the extension base.

(Make sure that the right side of A1SD51S has no module or blank cover (A1SG60).)

Note

A1SD51S uses the super-capacitor (capacitor) for backup.

When the super-capacitor is used at a high temperature, the backup time becomes short.

To reduce the temperature of the super-capacitor, be sure to observe the installation position.

The super-capacitor has its own service life.

If the lifetime is exceeded, replace the super-capacitor.

(2) Backup time

MIN. 48 hours (assurance value)

TYP. 120 hours (practical value)

(3) Other items

(1) If backup exceeding the backup time is required, backup the data with the PC CPU.

(A program that transfers data with the PC CPU is required.)

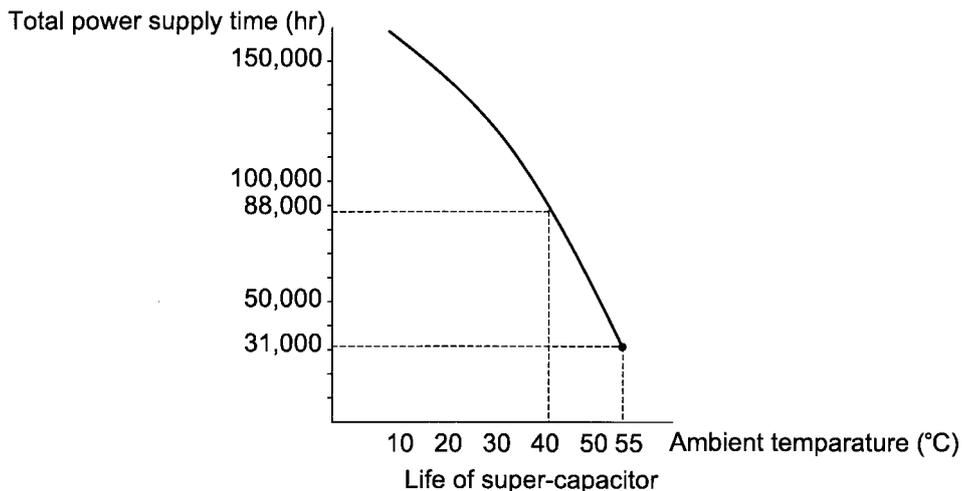
(2) The charging time of the super-capacitor is one hour.

(When the charging time is less than 1 hour, the backup time is short.)

(3) The life of the super-capacitor is shown in the diagram below. When the super-capacitor exceeds the service life (total power supply time) below, replace the super-capacitor.

(For replacement, please consult your local Mitsubishi service center or representative.

Before replacement, save the data of the common memory, extended register (ED) and extended relay (EM) in another medium.)



*2 The EEP-ROM can be rewritten 10000 times.

Rewriting is performed with the following system commands:

- MSAVE
- SET

3.2 Specifications of Interfaces with External Devices

The A1SD51S is equipped with the following interfaces:

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

For the specifications and connection of the interfaces, see Sections 3.2.3 to 3.2.5.

3.2.1 Interface uses

The use of each interface channel of the A1SD51S depends on the settings of SW1 to SW5 of mode setting switch 2 and the connected external device. Table 3.2 shows the uses of each interface channel.

Table 3.2 Uses of Each Interface Channel

O: Enabled X: Disabled

Interface Name	Uses		
	Console Port	Debugger Port	General-Purpose Port
RS-232 (CH1)	O *2	O *3	O
RS-232 (CH2)	X	O *3	O
RS-422/485 (CH3)	O *1	O *1	O

*1: The PC/AT can be used.

*2: The VT-382, VG-620 or PC/AT can be used.

*3: The VT-382 or VG-620 can be used.

(1) Console port

An interface can be designated as the console port by setting SW1 to SW5 of mode setting switch 2. (For details, see Section 5.3.2.)

The console connected to the console port stops multitask processing upon receiving run mode 03H ([Ctrl] + [C] key code).

(2) Debugger port

An interface can be designated as the debugger port by setting SW1 to SW5 of mode setting switch 2. (For details, see Section 5.3.2.)

The debugger connected to the debugger port permits multitask debugging.

(3) General-purpose port

The interface designated as neither console nor debugger port by the above setting serves as the general-purpose port. (For details, see Section 5.3.2.)

The peripheral device connected to the general-purpose port does not stop multitask processing upon receiving execution mode 03H ([Ctrl] + [C] key code).

3.2.2 Available baud rates

Baud rates up to 19200 bps can be selected for each interface channel of the A1SD51S.

However, when multiple interface channels are used simultaneously, the total baud rate for all the channels used is less than 28.8 Kbps (28800 bps).

$$(28800 \text{ bps}) > \left[\begin{array}{l} \text{Transmission speed} \\ \text{of Channel 1} \end{array} \right] + \left[\begin{array}{l} \text{Transmission speed} \\ \text{of Channel 2} \end{array} \right] + \left[\begin{array}{l} \text{Transmission speed} \\ \text{of Channel 3} \end{array} \right]$$

POINT

(1) When either Channel 1 or Channel 2 of the RS-232 interface is designated as a console or a debugger port, the data transmission for the interface is automatically set as follows:

- (a) Transmission rate : 9600 bps
- (b) Data length : 8 bits
- (c) Stop bits : 2 bits
- (d) Parity : None
- (e) Receiving buffer size : 1024 bytes

When using the VG-620, VT-382 as a console or a debugger, specify the same transmission settings as above.

(2) With a BASIC program, use the ZCNTL command to specify the transmission rate and other settings.

For further details on the ZCNTL command, refer to the AD51H-BASIC Programming Manual (Command).

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

3.2.3 RS-232 interface (CH1: Channel 1)

(1) Interface specifications

Table 3.3 shows the specifications of the RS-232 interface (CH1: Channel 1) of the A1SD51S.

Table 3.3 Specifications of the RS-232 Interface (CH1: Channel 1)

Item		Specifications																								
Applicable device	Debugger port	VG-620, VT-382, PC/AT																								
	Console port																									
	General-purpose port	External devices with the RS-232 interface																								
Transmission method		Conforms to RS-232																								
Synchronization method		Asynchronous system																								
Transmission specification settings	Transmission speed	Selectable from 300, 600, 1200, 2400, 4800, 9600, and 19200 bps			Selectable																					
	Parity bit	Yes or no (odd parity/even parity)																								
	Stop bit	1, 1.5, 2 bits																								
	Character data bit	5, 6, 7, 8 bits																								
	Communication control	DTR/DSR(ER/DR) control or DC code control																								
Connector of the A1SD51S	Model	17JE-13090-02(D8A) manufactured by DDK																								
	Pin arrangement and pin numbers	End view of connector	Pin No.	Signal abbreviation	Signal name	Signal direction A1SD51S ↔ External device																				
			2	RD (RXD)	Receive data	←																				
			3	SD (TXD)	Send data	→																				
			4	ER (DTR)	Data terminal ready	→																				
			5	SG	Signal ground	↔																				
			6	DR (DSR)	Dataset ready	←																				
			7	RS (RTS)	Request to send	→																				
			8	CS (CTS)	Clear to send	←																				
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>A1SD51S</p> <table border="1" style="border-collapse: collapse;"> <tr><td>(2)</td><td>RD (RXD)</td></tr> <tr><td>(3)</td><td>SD (TXD)</td></tr> <tr><td>(4)</td><td>ER (DTR)</td></tr> <tr><td>(5)</td><td>SG</td></tr> <tr><td>(6)</td><td>DR (DSR)</td></tr> <tr><td>(7)</td><td>RS (RTS)</td></tr> <tr><td>(8)</td><td>CS (CTS)</td></tr> </table> </div> <div style="text-align: center;"> <p>External device</p> <table border="1" style="border-collapse: collapse;"> <tr><td>RD (RXD)</td></tr> <tr><td>SD (TXD)</td></tr> <tr><td>ER (DTR)</td></tr> <tr><td>SG</td></tr> <tr><td>DR (DSR)</td></tr> <tr><td>RS (RTS)</td></tr> <tr><td>CS (CTS)</td></tr> </table> </div> </div>						(2)	RD (RXD)	(3)	SD (TXD)	(4)	ER (DTR)	(5)	SG	(6)	DR (DSR)	(7)	RS (RTS)	(8)	CS (CTS)	RD (RXD)	SD (TXD)	ER (DTR)	SG	DR (DSR)	RS (RTS)	CS (CTS)
(2)	RD (RXD)																									
(3)	SD (TXD)																									
(4)	ER (DTR)																									
(5)	SG																									
(6)	DR (DSR)																									
(7)	RS (RTS)																									
(8)	CS (CTS)																									
RD (RXD)																										
SD (TXD)																										
ER (DTR)																										
SG																										
DR (DSR)																										
RS (RTS)																										
CS (CTS)																										
<div style="display: flex; justify-content: center; align-items: center;"> <div style="text-align: center;"> <p>Connection example</p> </div> </div>																										

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

3.2.4 RS-232 interface (CH2: Channel 2)

(1) Interface specifications

Table 3.4 shows the specifications of the RS-232 interface (CH2: Channel 2) of the A1SD51S.

Table 3.4 Specifications of the RS-232 Interface (CH2: Channel 2)

Item		Specifications				
Applicable device	Debugger port	VG-620, VT-382, PC/AT				
	General-purpose port	External devices with the RS-232 interface				
Transmission method		Conforms to RS-232				
Synchronization method		Asynchronous system				
Transmission specification settings	Transmission speed	Selectable from 300, 600, 1200, 2400, 4800, 9600, and 19200 bps			Selectable	
	Parity bit	Yes or no (odd parity/even parity)				
	Stop bit	1, 1.5, 2 bits				
	Character data bit	5, 6, 7, 8 bits				
	Communication control	DTR/DSR(ER/DR) control or DC code control				
Connector of the A1SD51S	Model	17JE-13090-02(D8A) manufactured by DDK				
	Pin arrangement and pin numbers	End view of connector	Pin No.	Signal abbreviation	Signal name	Signal direction A1SD51S ↔ External device
			2	RD (RXD)	Receive data	←
			3	SD (TXD)	Send data	→
			4	ER (DTR)	Data terminal ready	→
			5	SG	Signal ground	↔
			6	DR (DSR)	Datase ^t ready	←
			7	RS (RTS)	Request to send	→
8			CS (CTS)	Clear to send	←	
Connection example						

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

MELSEC-A

3.2.5 RS-422/485 interface (CH3: Channel 3)

(1) Interface specifications

Table 3.5 shows the specifications of the RS-422/485 interface (CH3: Channel 3) of the A1SD51S.

Table 3.5 Specifications of the RS-422 Interface (CH3: Channel 3)

Item		Specifications					
Applicable device	Debugger port	PC/AT					
	Console port						
	General-purpose port	External devices with the RS-422/485 interface					
Transmission method		Conforms to RS-422					
Synchronization method		Asynchronous system					
Transmission specification settings	Transmission speed	Selectable from 300, 600, 1200, 2400, 4800, 9600, and 19200 bps			Selectable		
	Parity bit	Yes or no (odd parity/even parity)					
	Stop bit	1, 1.5, 2 bits					
	Character data bit	5, 6, 7, 8 bits					
Communication control		CS(CTS) control					
Connector of the A1SD51S	Model	17L-10250-27-D9A manufactured by DDK					
	Pin arrangement and pin numbers		End view of connector	Pin No.	Signal abbreviation	Signal name	Signal direction A1SD51S ↔ External device
			2	RDA	Receive data	←	
			15	RDB			
			3	SDA	Send data	→	
			16	SDB			
			4	RSA	Request to send	→	
			17	RSB			
			5	CSA	Clear to send	←	
			18	CSB			
20			SG	Signal ground	↔		
21	RR *1	Receive ready	←				
Connection example							
	Note	*1: To enable data reception by the A1SD51S, be sure to connect RR (receive ready) to both the SG (No. 20 pin) of the A1SD51S and the SG of the external device as shown above.					

(2) 1:n (multidrop) connection

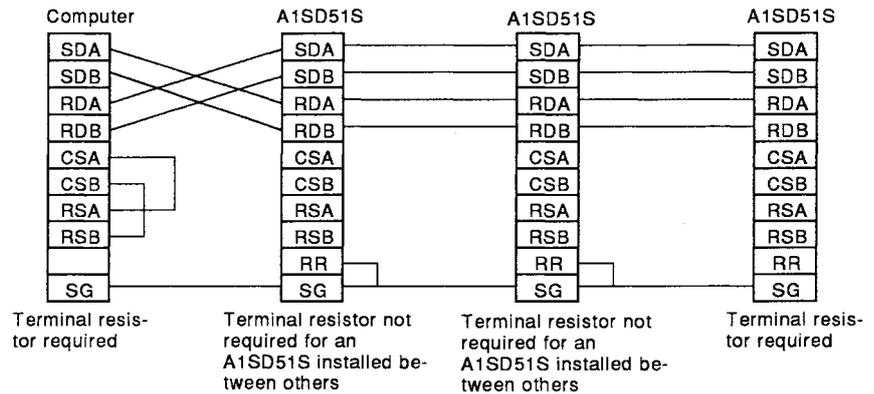
By using the RS-442/485 interface, a single computer can be connected with multiple A1SD51S modules (1:n connection).

(a) Specifications

Maximum connectable stations: 32 stations

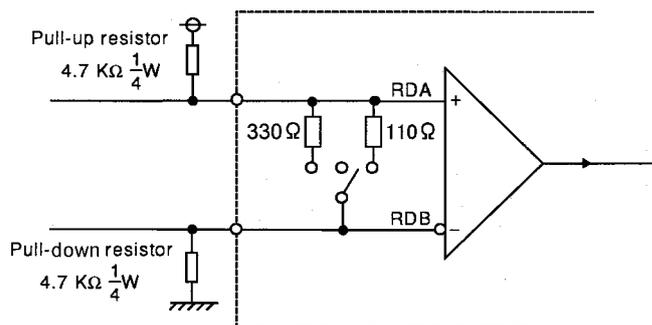
Maximum extension distance : 500 m

(b) Wiring



(c) Notes on wiring

- Do not make connections between the CSA/CSB and RSA/RSB terminals of different A1SD51S modules.
- Disable RS/CS control with Procedure Code 18 of the ZCNTL command.
- If both terminal stations are A1SD51S modules, connect a terminal resistor to the RS-422 or the RS-485 interface.
- If data receive errors are likely to occur due to noise along transmission lines, connect pull-up and pull-down resistors as shown below.



(d) Notes on programming

- When performing 1:n connection, enable high-impedance control with Procedure Code 48 of the ZCNTL command. For details of high-impedance control, refer to the AD51H-BASIC Programming Manual (Command).

3.3 AD51H-BASIC Commands

Table 3.6 lists the AD51H-BASIC commands that can be used with the A1SD51S.

(Refer to the AD51H-BASIC Programming Manual (Command) for details on the commands and functions.)

Table 3.6 AD51H-BASIC Commands and Functions

Classification		Command/ Function	Operation Description	
Commands 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 lines of a program.	
		LIST	Displays all (or part) of a program in the program area.	
ZBAS	Designates the number of the BASIC task area where the program is being created or executed.			
Functions to read memory capacity	Reading memory capacity	FRE	Sets the size of an unused area in the program area in terms of the number of bytes.	
Commands for program execution and control	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.	
		RESUME	Resumes the program execution after error handling by the ON ERROR GOTO instruction.	
	Controlling a program	CHAIN	This (a) clears all or part of the currently executed program, and (b) reads and executes a designated program.	
		FOR to NEXT	Repeatedly executes a series of commands a designated number of times.	
		STOP	When in the Programming mode, temporarily stops execution of the program and puts processing on standby for command input. ----- When in the Run mode, stops execution of the program and initiates a pause in processing.	
		END	When in the Programming mode, stops execution of the program and puts processing on standby for command input. ----- When in the Run mode, stops execution of the program and initiates a pause in processing.	
		WHILE to WEND	Repeatedly executes a series of commands between the WHILE and WEND commands while a designated condition is satisfied.	
	Branching a program	GOTO	Causes program execution to jump unconditionally to a designated line.	
		GOSUB to RETURN	Branches a program to the subroutine and returns it to a designated line number.	
		ON GOTO	Branches the program to one of the designated line numbers in accordance with the designated value.	
		ON GOSUB	Branches the program to one of the designated line numbers in accordance with the designated value.	
		IF GOTO ELSE ----- IF IHTEN ELSE	Selects the branching destination in accordance with the result of an operation formula.	
	Command for debugging a program	Debugging a program	TRON	Starts tracing a program
	TROFF	Resets the program trace set by the TRON command.		

(Continued from the previous page)

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

MELSEC-A

Table 3.6 AD51H-BASIC Commands and Functions (continued)

Classification		Command/ Function	Operation Description
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.
Command to write to a designated bit	Writing to a designated bit	WTSET	Writes 0 or 1 to a designated bit of an 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 an integer type array variable.
Character-string function	Character ↔ ASCII conversion	ASC	Converts the first character in a character-string formula to the corresponding character code.
		CHR\$	Converts the value of an integer expression into character codes.
	Character → numeric conversion	STR\$	Converts a numeric value (assumed to be decimal) 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 a substring of a specified length from a character-string, starting with the leftmost character of the string.
		MID\$ (version 2)	Designates a substring beginning at a designated position in a character-string
		RIGHT\$	Extracts a substring of a specified length from a character-string, starting with the rightmost character of the string.
		STRING\$	Designates a string containing a specified number of a specified character.
		SPACE\$	Designates a specified number of whitespace characters.
	Character search	INSTR	Searches for a substring within a character-string and designates the starting position.
	Character-string length search	LEN	Designates the number of characters in a character-string.
Format control function	Format control	SPC	Designates a specified number of whitespace characters.
		TAB	Advances the present character display position to a designated position.
Screen control commands	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.

(Continued from the previous page)

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

MELSEC-A

Table 3.6 AD51H-BASIC Commands and Functions (continued)

Classification		Command/ Function	Operation Description
I/O control commands	Port allocation	ZOPEN	Opens a communication channel (port) in preparation for performing communications with an external device.
		ZCLOSE	Closes the communication channel (port) used for communications with external devices.
	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 names of the files on an FD or HD to the printer.
		ZLDV	Switches the port of the printer.
	Output to a screen	WIDTH	Sets the output width for output to the printer.
		PRINT	Displays data on the screen.
	Output to ports	PRINT USING	Displays a character-string or a numeric value in a designated format.
		KEY LIST	Displays a character-string, defined by a function key, on the screen.
		ZSEND	Transmits data from a designated port.
	Input from the keyboard	NPUT	Enables data input from the keyboard.
		LINE INPUT	Stores an entire input line (up to 255 characters) in the character-string variables area without dividing it up.
Input from ports	ZRECEIVE	Receives data from a communication port.	
Port setting change	ZCNTL	Used to set communication data to the currently open communication port, or to read out the communication port status.	
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.
Port control function	Controls port state	COM ON/OFF/STOP	Control interruption-enabled, interruption-disabled, and stop from a communications line.
		ON COM GOSUB	Defines the starting line of a branch processing routine when an interruption is issued through a communications line.
Error control function	Error control	ON ERROR GOTO	Enables interrupt processing in the case of an error, and moves the operation to the first line of the error handling program.
		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 was detected.
Repetition command to programming mode	Repetition command to programming mode	SYSTEM	Stops control of the BASIC interpreter and switches the mode to the A1SD51S system mode. ----- Stops offline programming and returns the screen to the main menu.

(Continued from the previous page)

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

MELSEC-A

Table 3.6 AD51H-BASIC Commands and Functions (continued)

Classification		Command/ Function	Operation Description
File operation commands	BASIC program I/O	LOAD	Reads a program in an FD or HD 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 or HD.
	File maintenance	FILES	Displays the name of a file in an FD or HD.
		KILL	Clears a file in an FD or HD. Clears a system name in an FD or HD.
		NAME	Changes the file name of a program or data in an FD or HD.
File I/O commands	Allocation of file buffer	OPEN	Opens a file, and enables I/O processing.
		CLOSE	Ends file I/O processing.
	Sequential file I/O	PRINT #	Write 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.
	Random file I/O	LINE INPUT #	Reads a character line (up to 265 characters) from a sequential file, without dividing up the line, into the character-string variables area.
		FIELD	Allocates a designated variable area to the random file buffer.
		LSET	Transfers data to the random file buffer, storing it from the left side of the field, in preparation for the PUT command.
		RSET	Transfers data to the random file buffer, storing it from the right side of the field, in preparation for the PUT command.
		PUT	Writes one-record data from the buffer to a random file.
	GET	Reads one-record data from a random file to the buffer.	
File information function	File information	EOF	Designates -1 when the end of a sequential file is detected.
		LOF	Designates the size of a file as the number of sectors.
		LOC	Designates the current logical location within a file.
File data	Character-string → numeric value	CVI	Returns a character-string converted by the MKI\$ function to an integer.
		CVS	Returns a character-string converted by the MKS\$ function to a single-precision real number.
		CVD	Returns a character-string converted by the MKD\$ function to a double-precision real number.
		CVSMBF	Converts a character-string that has been converted from a single-precision real number of the AD51H-BASIC using an the MKS\$ function into the internal representation (single-precision internal representation of the IEEE format) of a floating decimal point real number used for AnA, AnU, QnA, Q2AS and AnAS CPU.
		CVDMBF	Converts a character-string that has been converted from a double-precision real number of the AD51H-BASIC using an the MKD\$ function into a 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 AnA, AnU, and AnUS CPU 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.

(Continued from the previous page)

Table 3.6 AD51H-BASIC Commands and Functions (continued)

Classification		Command/ Function	Operation Description
Definition and declaration commands	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.
		DEFSNG	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 a 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.
		RESTORE	Specifies the first line number of the DATA instruction data that are to be read by a READ.
	Assignment to variables	LET	Assigns the value of a formula to a variable.
		SWAP	Exchanges a value between two variables.
		ZMOVE	Transfers a value between two variables.
	BSWAP	Exchanges two values in a byte unit.	
Replacement of a character- string	MID\$ (version 1)	Replaces part of a character-string with another character-string.	

(Continued from the previous page)

Table 3.6 AD51H-BASIC Commands and Functions (continued)

Classification		Command/ Function	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	Rotates the memory contents of the specified value and returns the bit-rotated value.
		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 the tangent 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.
		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 and a double-precision real number 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-bit) 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 commands	Port switching	ZIDV	Specifies where the data is to be input by INPUT or any other instruction.
		ZODV	Specifies where the data is to be output by PRINT or any other instruction.
Buffer access function	Buffer access	GETMEM	Read data from the buffer and common memory of the A1SD51S.
		PUTMEM	Write data to the buffer and common memory of the A1SD51S.
Buffer memory access Variable	Buffer memory access	B@	Reads or writes bit information in the extension relays (EM), special relays (EM) and general-purpose inputs (X) and outputs (Y).
		W@	Reads or writes word information in the extension registers (ED) or special registers (ED).
Clock function commands	Clock function	TIME\$	Sets and reads a PC CPU time (hour, minute, and second)
		DATE\$	Sets and reads a PC CPU date (year, month, day, and day of the week)

(Continued from the previous page)

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

MELSEC-A

Table 3.6 AD51H-BASIC Commands and Functions (continued)

Classification		Command/ Function	Operation Description
Real-time control commands	Read-time control	ZEVENT	Sets whether an event occurrence is valid/invalid.
		DEF ZEVENT	Define an event to synchronize executions between programs. Define an event by the extension relay (EM) of the A1SD51S.
		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.
Real-time control command	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.

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

MELSEC-A

3.4 A1SD51S Internal Memory and Memory Map

Table 3.7 shows the memory that can be used with the A1SD51S and its memory map.

Table 3.7 A1SD51S Internal Memory and Memory Map

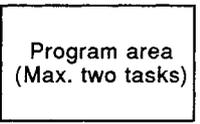
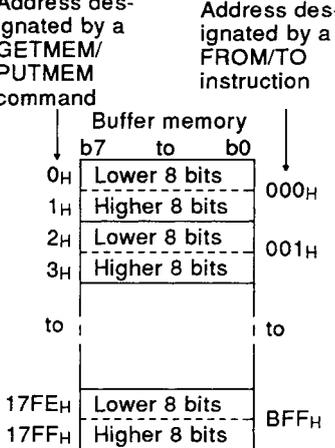
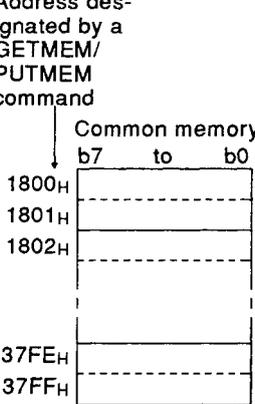
Internal Memory	Description	Memory Map	Reference Section
Execution program area	(1) Memory where a BASIC program executed by the A1SD51S is stored. (2) The BASIC programs are read from the EEPROM to the execution program area. (3) The task size can be set as 16 K, 32 K, 48 K, or 64 Kbytes.		3.5.1
Buffer memory	(1) Memory that can communicate 16-bit data with a PC CPU (2) Data can be read from and written to each task. (3) Data reading/writing is executed in 16-bit units in accordance with FROM/TO instructions from the PC CPU. (Use X1B as the interlock signal for the execution of the FROM/TO instruction.) (4) Data is read/written in 8-bit unit in accordance with a GETMEM/PUTMEM command from the BASIC program. (5) The buffer size is 6 Kbytes.		3.5.2
Common memory	(1) Memory used in the A1SD51S (Reading from/writing to a PC CPU is disabled.) (2) Data can be read from and written to each task. (3) Data is read/written in 8-bit unit in accordance with a GETMEM/PUTMEM command from the BASIC program. (4) The size of the common memory area is 8 Kbytes.		3.5.3

Table 3.7 A1SD51S Internal Memory and Memory Map (continued)

Internal Memory	Description	Memory Map	Reference Section																
<p>Extension register (ED)</p>	<p>(1) Memory used in the A1SD51S (Reading from/writing to a PC CPU is disabled.)</p> <p>(2) Data can be read from/written to each task.</p> <p>(3) Data reading/writing is executed by a GETMEM/PUTMEM command or special variable W@ from a BASIC program.</p> <ul style="list-style-type: none"> • GETMEM/PUTMEM command: 8-bit units • W@: 16-bit units <p>(4) The number of points in the extension register is 1024 (2048 bytes).</p>	<p>Device designated by W@</p> <p style="text-align: center;">↓ Extension register</p> <p style="text-align: center;">b15 to b0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">ED0</td><td style="width: 50%;"></td></tr> <tr><td>ED1</td><td></td></tr> <tr><td>ED2</td><td></td></tr> <tr><td>ED3</td><td></td></tr> <tr><td style="text-align: center;">to</td><td></td></tr> <tr><td>ED1022</td><td></td></tr> <tr><td>ED1023</td><td></td></tr> </table>	ED0		ED1		ED2		ED3		to		ED1022		ED1023		<p>3.5.4</p>		
ED0																			
ED1																			
ED2																			
ED3																			
to																			
ED1022																			
ED1023																			
<p>Special register (ED)</p>	<p>(1) An area with a designated use in the internal memory of the A1SD51S (Reading from/writing to a PC CPU is disabled.)</p> <p>(2) Data can be read from each task.</p> <p>(3) Data is read in 16-bit units using special variable W@ from a BASIC program.</p> <p>(4) The number of points in the special register is 128.</p>	<p>Device designated by W@</p> <p style="text-align: center;">↓ Special register</p> <p style="text-align: center;">b15 to b0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">ED9000</td><td style="width: 50%;"></td></tr> <tr><td>ED9001</td><td></td></tr> <tr><td>ED9002</td><td></td></tr> <tr><td>ED9003</td><td></td></tr> <tr><td>ED9004</td><td></td></tr> <tr><td style="text-align: center;">to</td><td></td></tr> <tr><td>ED9126</td><td></td></tr> <tr><td>ED9127</td><td></td></tr> </table>	ED9000		ED9001		ED9002		ED9003		ED9004		to		ED9126		ED9127		<p>3.5.5</p>
ED9000																			
ED9001																			
ED9002																			
ED9003																			
ED9004																			
to																			
ED9126																			
ED9127																			
<p>Extension relay (EM)</p>	<p>(1) Relay used in the A1SD51S (Reading from/writing to a PC CPU is disabled.)</p> <p>(2) ON/OFF data can be read from and written to each task.</p> <p>(3) ON/OFF data is read from/written to a BASIC program using special variable B@.</p> <p>(4) The number of points in the extension relay is 1024.</p>	<p>Device designated by B@</p> <p style="text-align: center;">↓ Extension register</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">EM0</td><td style="width: 50%;">ON/OFF</td></tr> <tr><td>EM1</td><td>ON/OFF</td></tr> <tr><td style="text-align: center;">to</td><td></td></tr> <tr><td>EM1023</td><td>ON/OFF</td></tr> </table>	EM0	ON/OFF	EM1	ON/OFF	to		EM1023	ON/OFF	<p>3.5.6</p>								
EM0	ON/OFF																		
EM1	ON/OFF																		
to																			
EM1023	ON/OFF																		
<p>Special relay (EM)</p>	<p>(1) An area with a designated use in the internal memory of the A1SD51S (Reading from/writing to a PC CPU is disabled.)</p> <p>(2) Data can be read from each task.</p> <p>(3) ON/OFF data is read from a BASIC program using special variable B@.</p> <p>(4) The number of points in the extension relay is 128.</p>	<p>Device designated by B@</p> <p style="text-align: center;">↓ Special register</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">EM9000</td><td style="width: 50%;">ON/OFF</td></tr> <tr><td>EM9001</td><td>ON/OFF</td></tr> <tr><td style="text-align: center;">to</td><td></td></tr> <tr><td>EM9127</td><td>ON/OFF</td></tr> </table>	EM9000	ON/OFF	EM9001	ON/OFF	to		EM9127	ON/OFF	<p>3.5.7</p>								
EM9000	ON/OFF																		
EM9001	ON/OFF																		
to																			
EM9127	ON/OFF																		

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

3.5 A1SD51S Internal Memory Areas

The following memory areas of the A1SD51S are available to the user.

(1) Memory area where the program for execution is stored

- Program area (see Section 3.5.1)
This memory stores BASIC programs executed by the A1SD51S.
This program area can be divided into two task areas and has a total capacity of 64 Kbytes.

(2) Memory for communications

Item	Content	Address			Reference Section
		PC CPU	BASIC Program		
		FROM/TO command	GETMEM/PUTMEM command	Special Variable B@/W@	
Communications with the PC CPU	Buffer memory	0000 _H to 0BFF _H	0000 _H to 17FF _H	—	Section 3.5.2
Communications between tasks	Common memory	—	1800 _H to 37FF _H	—	Section 3.5.3
Reading/writing ON/OFF data	Expansion register	—	—	ED0000 to ED1023	Section 3.5.4
	System Area	—	—	—	—
	Expansion relay	—	—	EM0000 to EM1023	Section 3.5.6
	System Area	—	—	—	—
	Special register	—	—	ED9000 to ED9127	Section 3.5.5
	System Area	—	—	—	—
	Special relay	—	—	EM9000 to EM9127	Section 3.5.7
	System Area	—	—	—	—

Buffer memory is not backed up by a battery.

If the A1SD51S's power is turned OFF, the data in the buffer memory, etc. are erased.

Transmit data of the buffer memories, etc. to a latched device in the PC or to a file register.

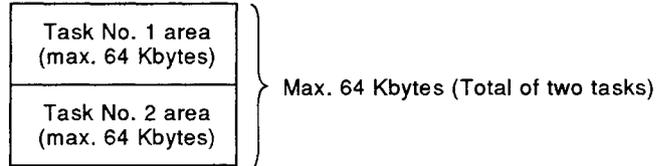
3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

3.5.1 Program area: Not backed up by a battery

BASIC programs created at a console and the program for execution read from the EEP-ROM are stored in the program area.

(1) Memory capacity

The program area is max. 64 Kbytes.
The 64 K bytes are divided into two program areas (tasks 1 and 2).

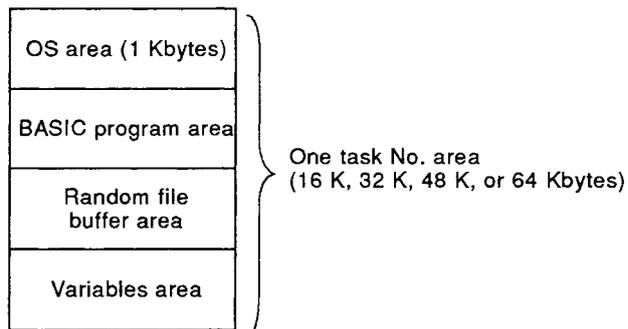


(2) BASIC program capacity

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

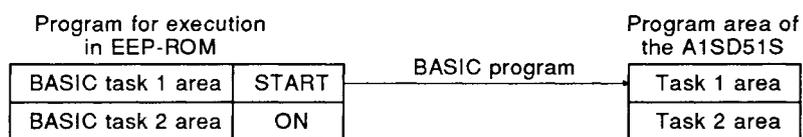


POINT

The program for execution stored in the EEP-ROM is read to the program area of the A1SD51S in the following cases:

- (a) Programming mode: When the power is turned ON or the A1SD51S is reset.
 - (b) Multitask debug mode
 - (c) Run mode
- } When the RUN keyswitch is set to RUN after the power is turned ON or the A1SD51S is reset.

However, data in an area where the multitask setting is ON will not be read to the program area of the A1SD51S.



3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

3.5.2 Buffer for communications with a PC CPU: Not backed up by a battery

The buffer is used for data communications between the A1SD51S and a PC CPU.
(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.)

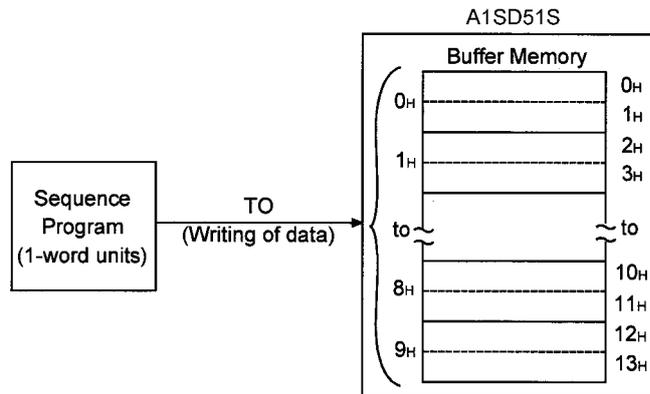
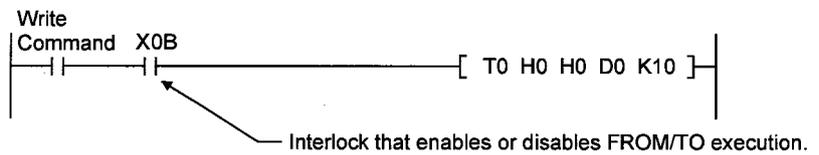
(1) Data read/write

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

Program example

The following shows a program where values in D0 to D9 are written into the buffer memory area addressed 0H to 9H when the multitask execution start signal (X0B) is ON.

To write data from the PC CPU to the buffer memory or to read data from the buffer memory, use the multitask execution start signal (X0B) as an interlock.



(b) Data is read from and written to the buffer of the A1SD51S by using a GETMEM/PUTMEM command in a BASIC program.

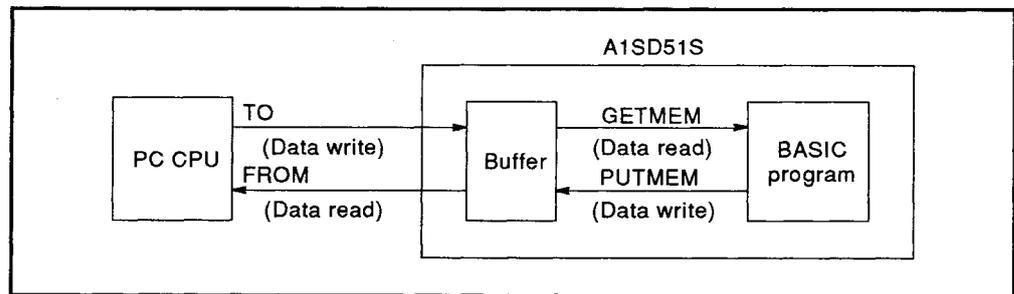
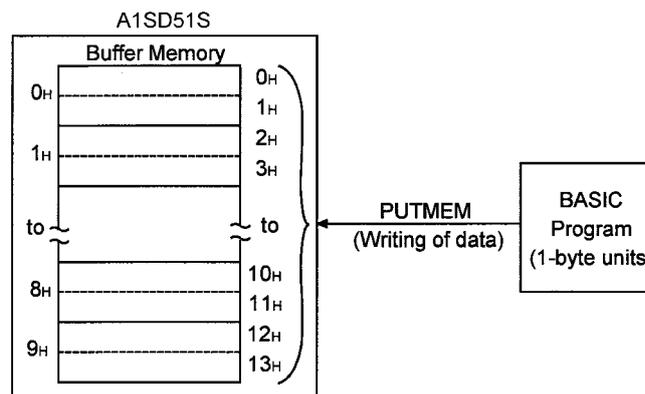


Fig. 3.2 Buffer Data Communications

Program example

The following shows a program where data from A%(0) to A%(9) (equivalent to 20 bytes) are written into the buffer memory area addressed 0_H to 13_H using the PUTMEM command.

```
100 PUTMEM TO 0, 0 FROM A%( ), 0, 20
```



REMARK

For details on GETMEM/PUTMEM commands, refer to the AD51H-BASIC Programming Manual (Command).

(2) Buffer capacity

The A1SD51S buffer consists of 3 K words (6 Kbytes).

(3) Buffer structure

A buffer address consists of 16 bits.
All areas are available to the user.

(4) Buffer address

The buffer addresses designated with a PC CPU differ from those designated with 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 with a BASIC program

Designate buffer addresses with a BASIC program in one-byte units.
Use hexadecimal numbers to designate buffer addresses 0H to 17FFH.

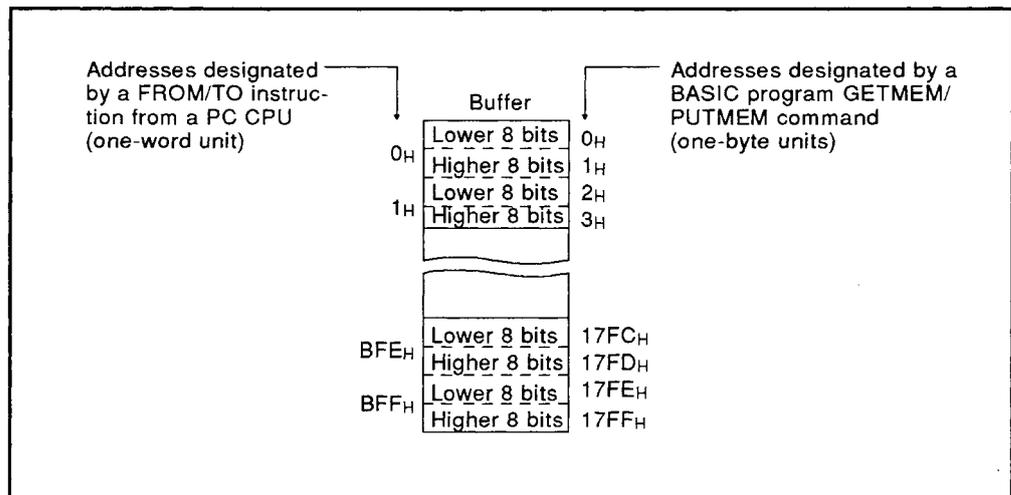


Fig. 3.3 Buffer Addresses

POINT

During the various processes of the special function module, the access from the PC CPU is processed as a priority. Thus, if the special function module's buffer memory is frequently accessed from the PC CPU, the PC CPU scan time will increase and a delay will occur in the special function module's processes. Access the buffer memory from the PC CPU with the FROM/TO command, etc., only when necessary.

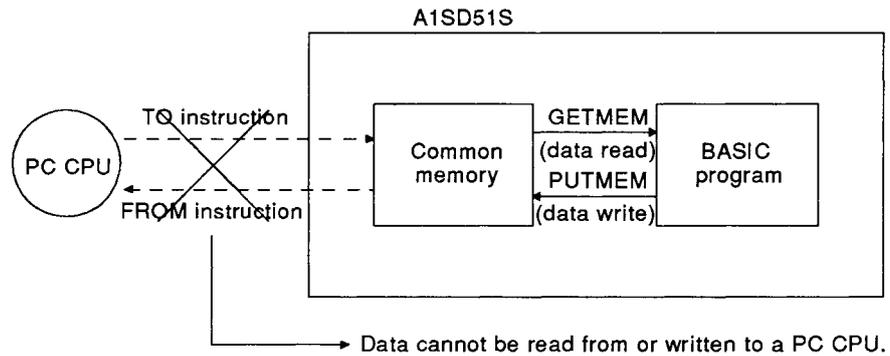
3.5.3 Common memory shared between tasks

The common memory can be used for data communications between tasks in the A1SD51S.

Data cannot be read from or written to a PC CPU.

(1) Reading data from/writing data to the common memory

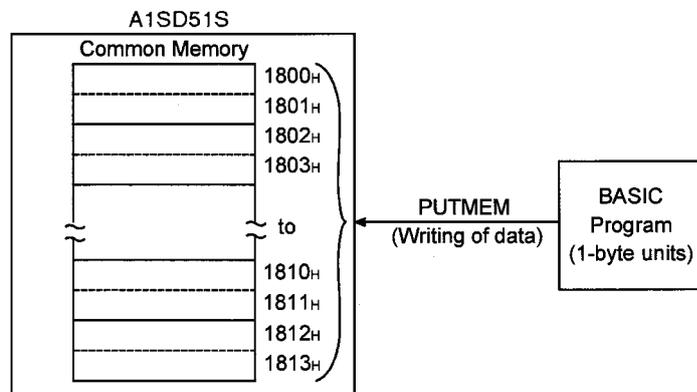
- (a) Use a BASIC program GETMEM/PUTMEM command to read from or write data to the common memory.



Program example

The following shows a program where data from A%(0) to A%(9) (equivalent to 20 bytes) are written into the common memory area addressed 1800_H to 1813_H using the PUTMEM command.

```
100 PUTMEM TO &H1800, 0 FROM A%(), 0, 20
```



(2) Memory capacity

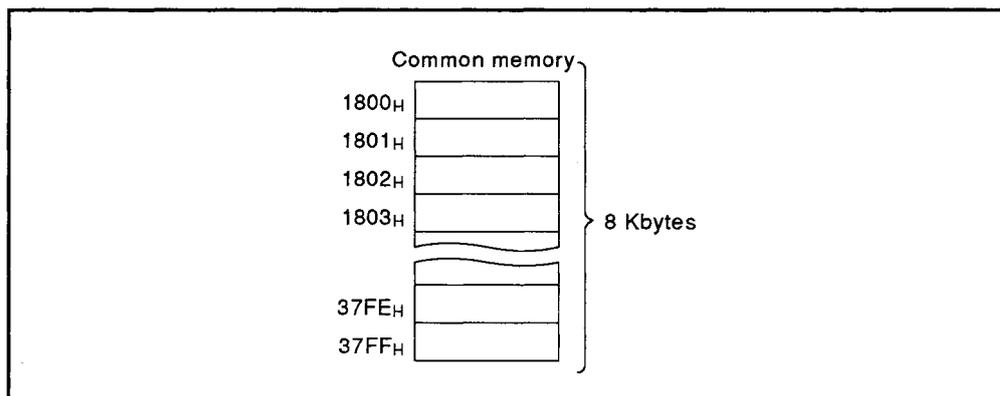
The A1SD51S common memory capacity is 8 Kbytes.

(3) Common memory configuration

A common memory address consists of 1 byte (8 bits).
All areas are available to the user.

(4) Common memory addresses

The common memory addresses start from address 1800H, next to the buffer.



REMARK

For details on GETMEM/PUTMEM commands, refer to the AD51H-BASIC Programming Manual (Command).

3.5.4 Extension register (ED)

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

Data can be stored in the A1SD51S and one-word data can be communicated between tasks by using an extension register.¹⁾

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

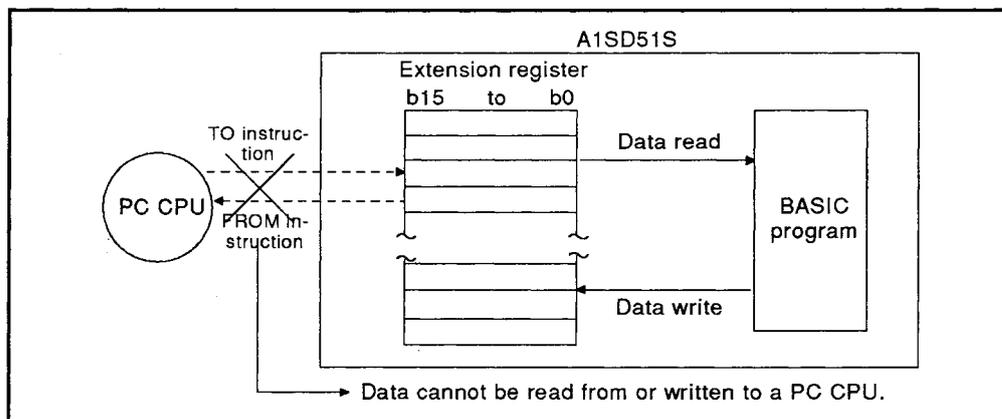


Fig. 3.4 Reading and Writing Data

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

Data can be read from and written to the extension register by using special variable W@ or a GETMEM/PUTMEM command.

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

(a) Reading and writing data with special variable W@

Designate the extension register from ED0 to ED1023 by using special variable W@.
Designate it in one-word units.

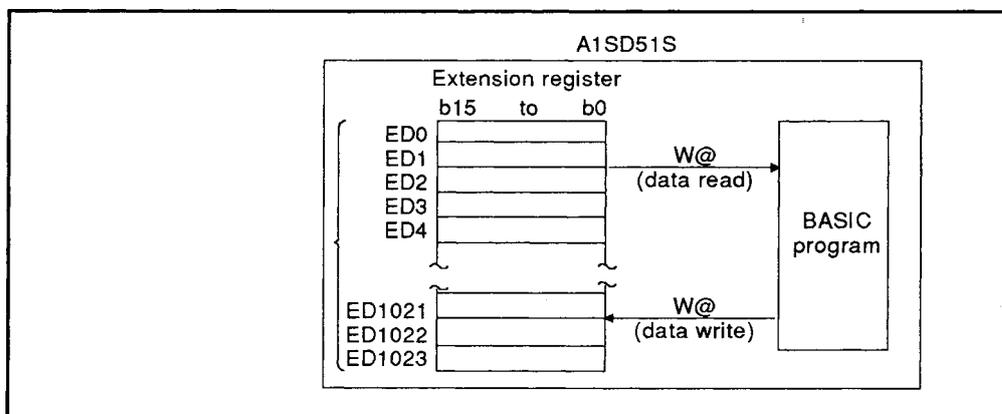
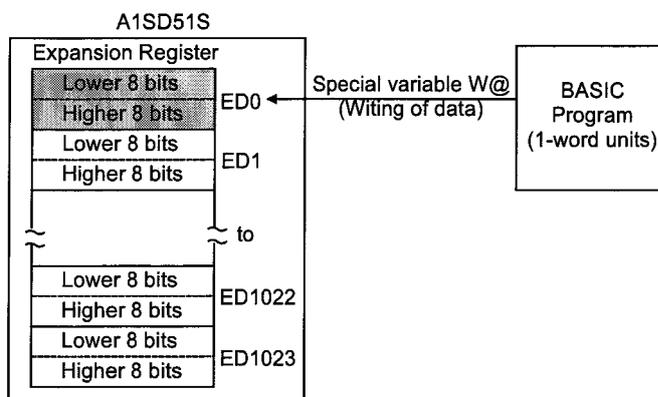


Fig. 3.5 Reading and Writing Data by Using Special Variable W@

Program example

The following shows a program where "123" is written into the A1SD51S's expansion register ED0.

```
150 W@ (ED, 0) =123
```



REMARK

*1: The buffer and common memory can be used for communicating one-word data between tasks.

(b) Reading and writing data using PUTMEM/GETMEM commands

With the PUTMEM/GETMEM command, data can be written to or read from the extension register.
Designate addresses in the range ED0 to ED1023.

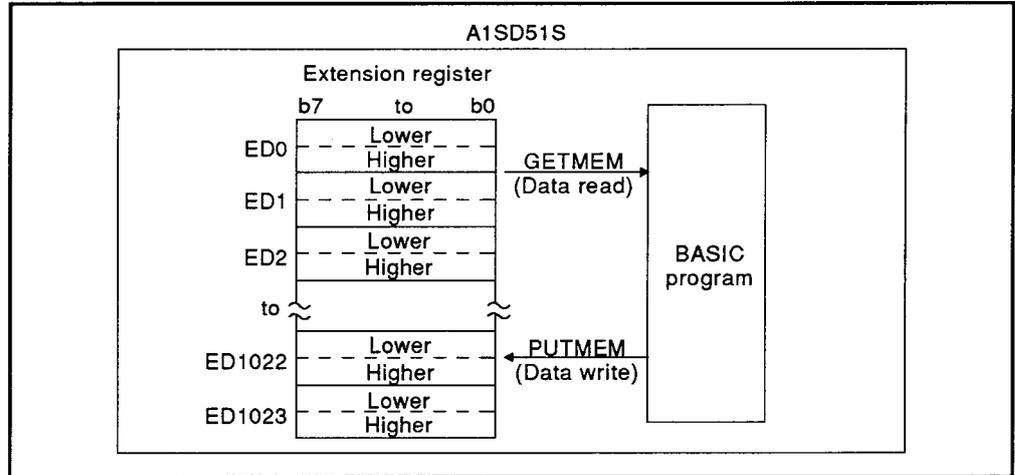
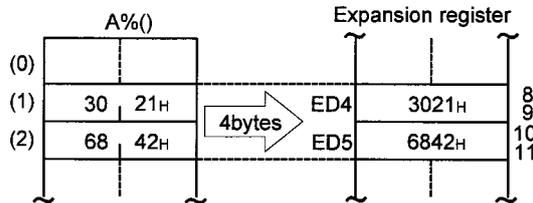


Fig. 3.6 Reading and Writing Data by Using a PUTMEM/GETMEM Command

Program example

In the following program, data (4 bytes) in A% (1) and A% (2) of the A1SD51S are written into ED4 and ED5.

```
20 PUTMEM TO W@ (ED,2), 4 FROM A%(), 2, 4
```



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

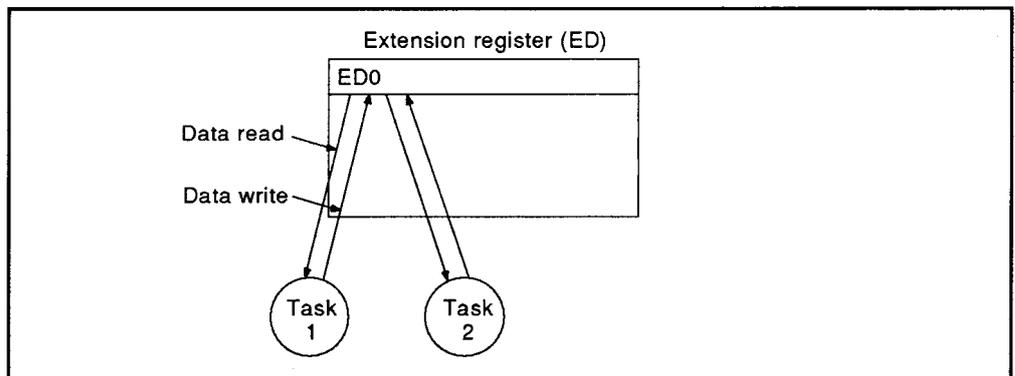


Fig. 3.7 Using an Extension Register with a BASIC Program

(3) The points of the extension register

The extension register of the A1SD51S has 1024 points (from ED0 to ED1023).
(Extension register numbers ED0 to ED1023 are displayed in decimal.)

3. PERFORMANCE SPECIFICATIONS OF THE A1SD51S

3.5.5 Special registers (ED9000 to ED9127)

Special registers are registers which can be set for specific purposes in the A1SD51S.

Since the OS of the A1SD51S stores data, this data can be read and utilized by using special variable W@ in the BASIC program.

Table 3.8 Special Registers

Special Register No.	Name	Description	Details	Related Special Relay	Related Special Register																		
ED9000	Error number	0 to 8	<ul style="list-style-type: none"> The error number registered by the A1SD51S is stored 	EM9000	ED9001 to ED9008																		
ED9001 to ED9008	Error code	0: Normal Except 0: Error code	<ul style="list-style-type: none"> Error codes displayed by the LED indicator 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 indicator can be cleared with the indicator reset switch. 	EM9000	ED9000																		
ED9009 to ED9016	Error line number	0: No line number 1 to FFFF: Line number	<ul style="list-style-type: none"> 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 border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>For Storing an Error Code</th> <th>For Storing a Line Number</th> </tr> </thead> <tbody> <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> </tbody> </table> <p style="margin-left: 20px;">[When an error is detected by the BASIC, the error code and the line number are alternately displayed by the LED indicator.]</p> <ul style="list-style-type: none"> A line number displayed by the LED indicator can be cleared by the indicator reset switch. 	For Storing an Error Code	For Storing a Line Number	ED9001	ED9009	ED9002	ED9010	ED9003	ED9011	ED9004	ED9012	ED9005	ED9013	ED9006	ED9014	ED9007	ED9015	ED9008	ED9016	EM9000	ED9001 to ED9008
For Storing an Error Code	For Storing a Line Number																						
ED9001	ED9009																						
ED9002	ED9010																						
ED9003	ED9011																						
ED9004	ED9012																						
ED9005	ED9013																						
ED9006	ED9014																						
ED9007	ED9015																						
ED9008	ED9016																						
ED9020	Run modes	0 to 4	<ul style="list-style-type: none"> The setting number of the mode setting switch is stored. <ul style="list-style-type: none"> 0, 1: Run mode 2, 3: Multitask debug mode 4: Programming mode 	—	—																		

Table 3.8 Special Registers (continued)

Special Register No.	Name	Description	Details	Related Special Relay	Related Special Register										
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.	<ul style="list-style-type: none"> The states of the BASIC tasks 1, 2 are stored as follows. <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: right;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: left;">b2</td> <td style="text-align: left;">b1</td> <td style="text-align: left;">b0</td> </tr> <tr> <td colspan="3" style="text-align: center;">0</td> <td style="text-align: center;">0/1</td> <td style="text-align: center;">0/1</td> </tr> </table> <div style="margin-left: 100px;"> </div> <div style="margin-left: 100px;"> </div> <ul style="list-style-type: none"> 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.) The BASIC task is in the STOP state. (All BASIC programs are not being executed.) The BASIC task is in the wait state. (The BASIC task is waiting for execution of a command (PRINT command, etc.) to be completed, or the execution interruption command has been executed.) When the bit of a designated task is "0", start the BASIC task using the ZSTART command. 	b15	to	b2	b1	b0	0			0/1	0/1	—	—
b15	to	b2	b1	b0											
0			0/1	0/1											

3.5.6 Extension relay (EM)

Since the extension relay is the relay used for communicating ON/OFF data between tasks in the A1SD51S.

(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 the extension relay ON/OFF using special variable B@.

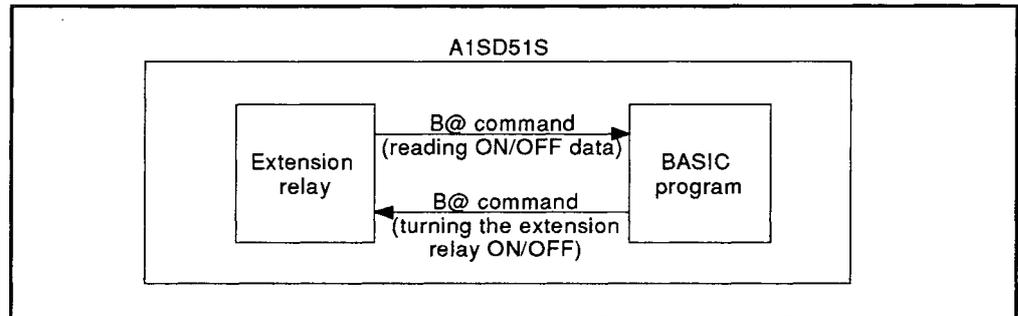
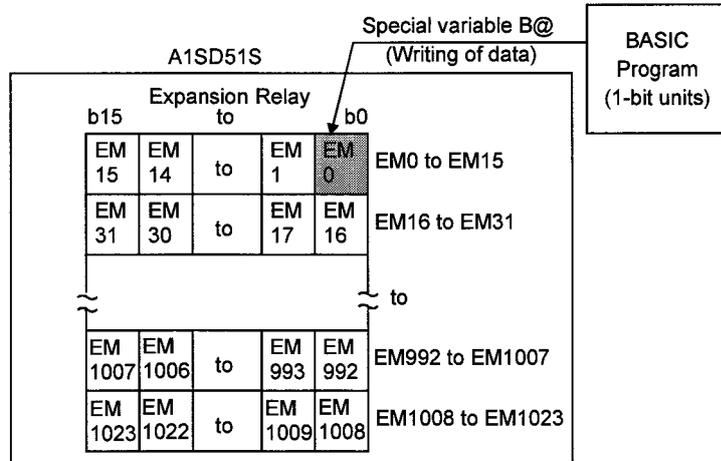


Fig. 3.8 Turning the Extension Relay ON/OFF and Reading ON/OFF Data

Program example

The following shows a program where "1" is written into the A1SD51S's expansion relay EM0.

```
150 B@ (EM, 0)=1
```



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

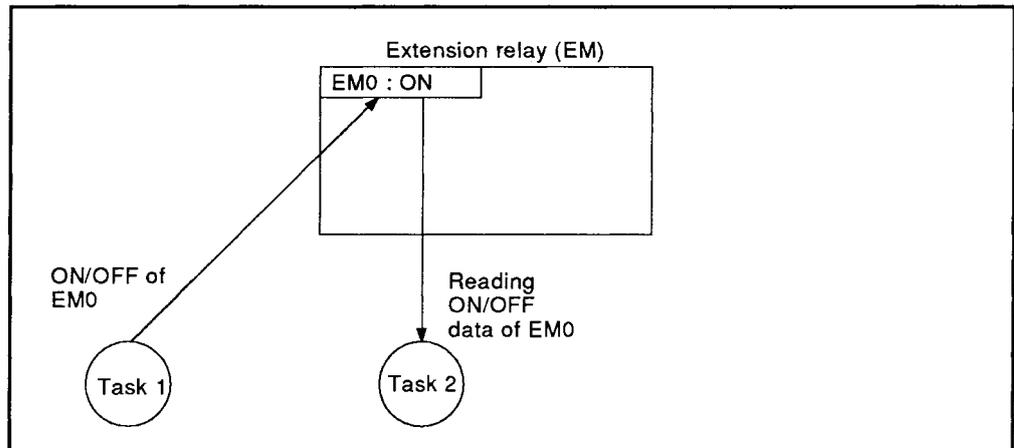


Fig. 3.9 Switching Extension Relay ON/OFF with the BASIC Program

- (2) Number of extension relay points

The A1SD51S extension relay has 1024 points.

REMARK

For details on special variable B@, refer to the AD51H-BASIC Programming Manual (Command).

3.5.7 Special relays (EM9000 to EM9127)

Special relays are relays which can be set for specific purposes in the A1SD51S.

Since the OS of the A1SD51S executes ON/OFF control, the ON/OFF state can be read and utilized by using special variable B@ in a BASIC program.

Table 3.9 Special Relays

Special Relay NO.	Name	Description	Details	Related Special Register
EM9000	Error registration flag	ON: Error registration OFF: No error registration	<ul style="list-style-type: none"> This comes ON when a warning error, error message, or system-down error message is displayed by the LED indicator. 	ED9000 to ED9016
EM9020	Operational flag	ON: Multitasking OFF: Not multitasking	<ul style="list-style-type: none"> Turned ON when the A1SD51S is multitasking. Turned OFF when the A1SD51S is not multitasking. 	—

3.6 Receive Buffer

The receive buffer is used to receive data from and transmit data to an external device connected to the RS-232 interface or the RS-422/485 interface of the A1SD51S.

The receive buffer for each respective interface is 1024 bytes. (The receive buffer size can be changed by the ZCNTL command.)

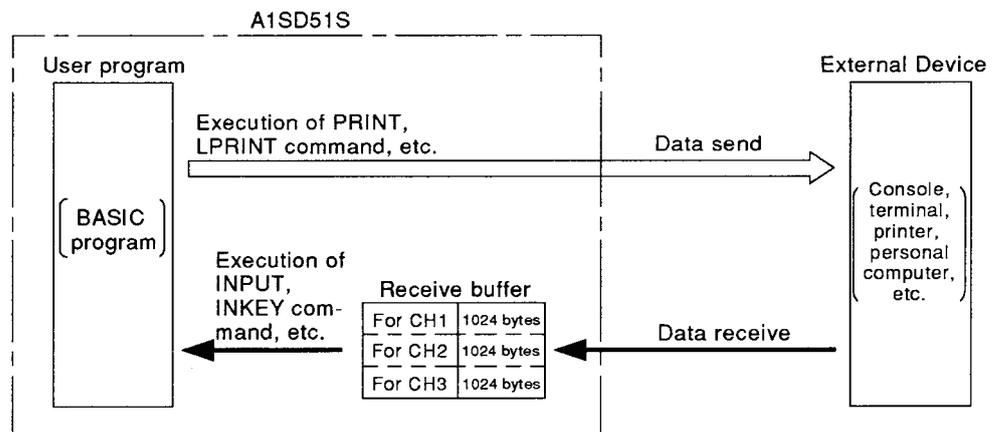
The receive buffer stores the data sent from an external device.

When the data receive command is executed, the receive buffer data are stored in the area shown by the designated variable.

The following lists the available data receive commands.

Command, Function Name	Processing Contents
INPUT	<ul style="list-style-type: none"> Carries out key input of data. Character code control codes (0_H to 1F_H), commas (2C_H) and DEL (7F_H) cannot be handled as data.
INKEY\$ INPUT\$	<ul style="list-style-type: none"> Stores the data received from the interface in the designated variable. Character code control codes (0_H, 03_H, 13_H, 80_H and FD_H to FF_H) cannot be handled as data.
LINE INPUT	<ul style="list-style-type: none"> Carries out key input of data. Character code control codes (0_H to 1F_H) and DEL (7F_H) cannot be handled as data.
ZRECEIVE	<ul style="list-style-type: none"> Stores the data received from the interface in the designated variable

See the Programming Manual (Command) for the data receive commands and the send/receive processing.



3.7 I/O for a PC CPU

The A1SD51S is used to communicate ON/OFF data between a PC CPU and the A1SD51S and to start a designated task from the PC CPU. Use X devices to send input signals from the A1SD51S to the PC CPU. For transmission of output signals from the PC CPU to the A1SD51S, use Y devices.

3.7.1 I/O (X, Y) list

Fig. 3.10 shows how to determine device numbers when an A1SD51S is loaded in slot No. 0 of the main base unit.

- The I/O numbers used for a sequence program depend on the slot number in which the A1SD51S is loaded.
- The values designated by special variables B@ of A1SD51S BASIC programs are fixed in the range &H00 to &H1F.

Table 3.10 shows device numbers and signal names. (For details on each signal, refer to Sections 3.7.2 and 3.7.3.)

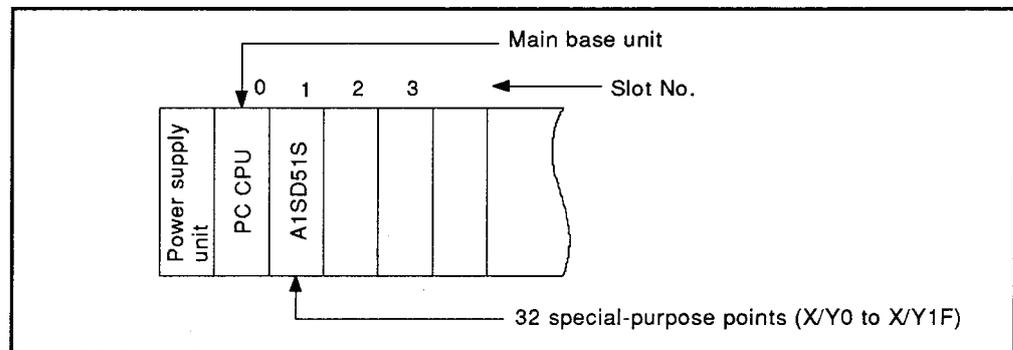


Fig. 3.10 Device Numbers Identified by the PC CPU when the A1SD51S is Loaded in Slot No. 0

Table 3.10 A1SD51S I/O Device Numbers and Signal Names

Signal Direction: A1SD51S → PC CPU			Signal Direction: PC CPU → A1SD51S		
Input No. Used by PC CPU	Value Designated by Special Variable B@	Signal Name	Input No. Used by PC CPU	Value Designated by Special Variable B@	Signal Name
X00 to X0A	&H00 to &H0A	General-purpose input	Y00 to Y10	&H00 to &H10	General-purpose output
X0B	&0B	Multitask execution start			
X0C	&0C	Multitask execution interrupt			
X0D	&0D	A1SD51S system down			
X0E to X0F	&0E to &0F	Unusable			
X10 to X1F	&H10 to &H1F	General-purpose input	Y11 to Y12	&H11 to &H12	General-purpose output/Start task No. designation*1
			Y13 to Y18	&H13 to &H18	General-purpose output
			Y19	&H19	Task start signal (cannot be used as a general-purpose output signal)
			Y1A to Y1F	&H1A to &H1F	Unusable

REMARK

*1: The task whose multitask setting is "start by interrupt from PC CPU" is taken as the start task number designation signal.
 The task whose multitask setting is not "start by interrupt from PC CPU" is taken as the general-purpose output signal.

3.7.2 Details on input devices (X)

This section describes the input devices (X) of the A1SD51S.
 The input device numbers (Xn) herein are those identified by the PC CPU.

- (1) General-purpose input devices (X00 to X0A, X10 to X1F: 27 points)

The user can use the general-purpose input devices to transmit ON/OFF data from the A1SD51S to the PC CPU.
 Use special variables B@ to turn the general-purpose input devices ON and OFF.

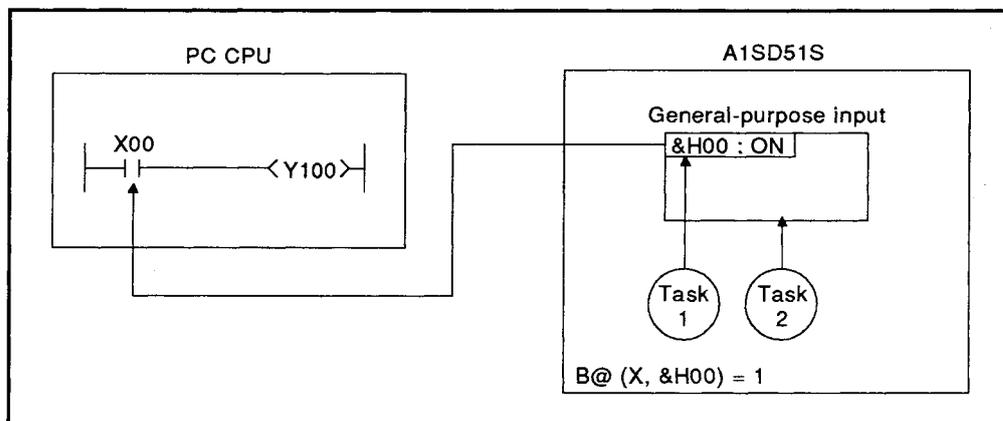


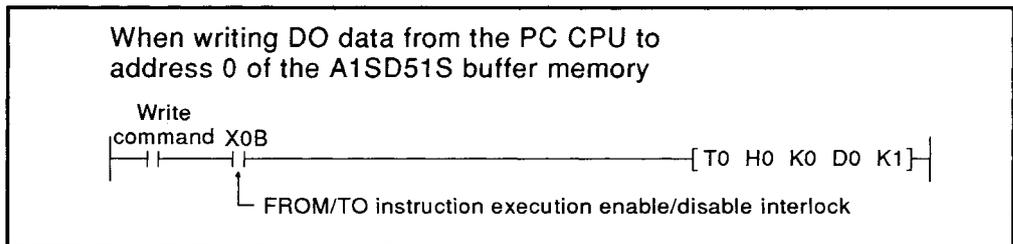
Fig. 3.11 Communications with the PC CPU by Turning a General-Purpose Input Device ON and OFF

(2) Multitask execution start signal [X0B]

- (a) This signal becomes valid (executes ON/OFF control) when mode setting switch 1*1 of the A1SD51S is set to "run mode (0 or 1)". (The flag remains invalid (OFF) when the A1SD51S is set to another mode.)
- (b) The flag turns ON when the A1SD51S is in the run mode and ready to perform multitasking after the power has been switched ON or the module has been reset.

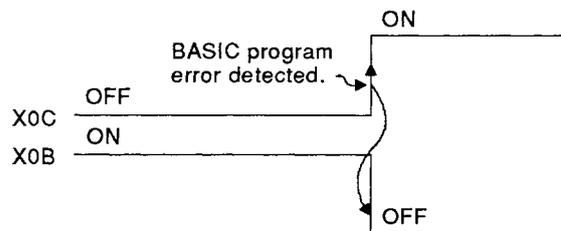
In a sequence program, use it as an interlock for enabling or disabling the FROM/TO command.

(When the QD51 (-R24) is not in the multitask execution enabled state, an attempt to execute the FROM/TO command will result in a "SP UNIT DOWN" error and the PC CPU will stop its operation.)*2

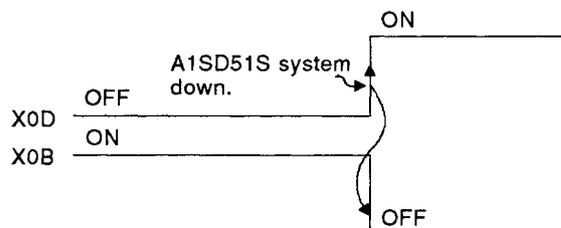


(c) The flag turns OFF under the following conditions when the A1SD51S is in the run mode.

- 1) The RUN switch of the A1SD51S is in the STOP position.
- 2) The A1SD51S is reset by the RUN switch.
(The flag will turn ON as soon as the A1SD51S completes resetting and reenters the multitask execution enabled state.)
- 3) A BASIC program error is detected.
(Multitask execution stop interrupt signal (X0C) will turn ON.)



4) The A1SD51S system is down.
(A1SD51S system down signal (X0D) turns ON.)



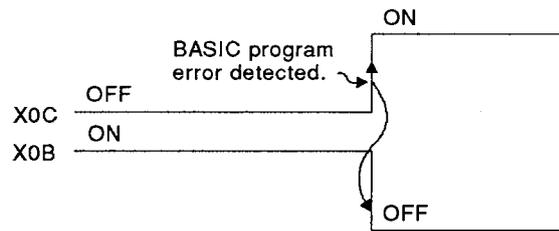
REMARK

*1: For details of mode setting switch 1, refer to Sections 5.3.1.

*2: If the A1SD51S is reset within the accessible time frame, no error will occur by execution of the FROM/TO instruction.
(The multitask execution start signal (X0B) turns OFF when the A1SD51S is reset.)
The accessible time is set with SW8 of mode setting switch 2.
For details, refer to Section 5.3.5.

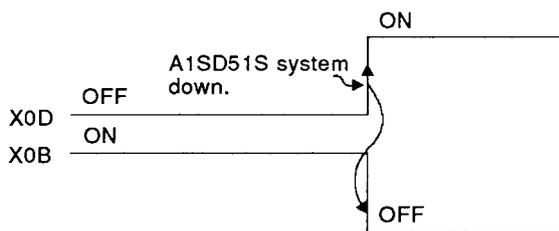
(3) Multitask execution interrupt signal [X0C]

- (a) This flag becomes valid (executes ON/OFF control) when mode setting switch 1 of the A1SD51S is set to "run mode (0 or 1)".
(The flag remains invalid (OFF) when the A1SD51S is set to another mode.)
- (b) The flag turns on when a BASIC program error is detected during multitasking.
(At the same time, the status is stored in ED9022^{*1}, and the A1SD51S goes into the system mode.
If an error occurs, correct the BASIC program.)
- (c) The multitask execution start signal (X0B) will turn OFF when the multitask execution interrupt signal (X0C) turns ON.



(4) A1SD51S system down signal [X0D]

- (a) This flag becomes valid (executes ON/OFF control) when mode setting switch 1 of the A1SD51S is set to "run mode (0 or 1)".
(The flag remains invalid (OFF) when the A1SD51S is set to another mode.)
- (b) The flag turns ON when an error which results in system down occurs during multitasking. The status is stored in ED9022^{*1}.
Resetting the A1SD51S by setting the RUN switch to the RESET position clears the A1SD51S down error.
- (c) The multitask execution start signal (X0B) will turn OFF when the A1SD51S system down signal (X0D) turns ON.



REMARK

*1: Refer to Section 3.5.5 for ED9022.

3.7.3 Details of output devices (Y)

This section describes the output devices (Y) of the A1SD51S. The output device numbers (Yn) herein are those identified by the PC CPU.

(1) General-purpose output devices (Y00 to Y10, Y13~Y18: 23 points)

The general-purpose output devices are used to transmit ON/OFF data from the PC CPU to the A1SD51S.

At the PC CPU, the general-purpose output devices can be turned ON or OFF by the sequence program or GX Developer's device test function, etc.

The A1SD51S reads the ON/OFF data of the general-purpose output devices using special variables B@.

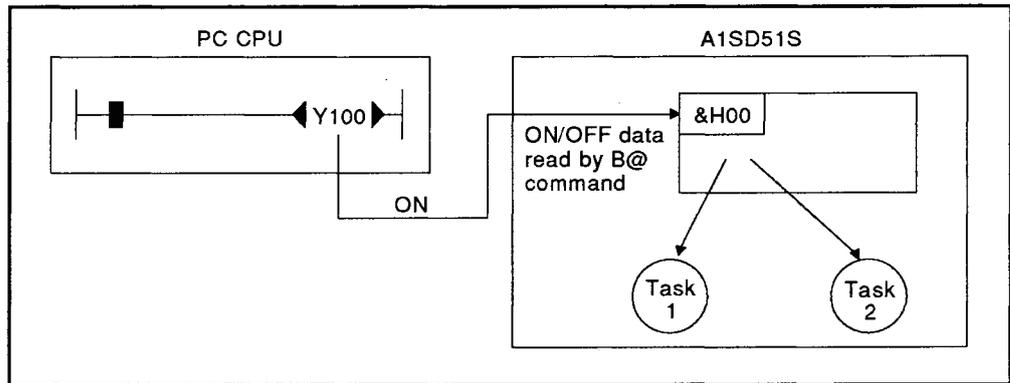


Fig. 3.12 Reading General-Purpose Output Devices Turned ON or OFF by PC CPU

(2) Start task No. designation signals (Y11, Y12: 2 points)

- (a) These signals designate tasks to be started by the PC CPU. One signal is assigned to each task.
- (b) Only a task whose multitask setting is "IT (start by interrupt from PC CPU)" can be a start task No. designation signal (Y11/Y12). The device numbers of tasks whose multitask setting is not "IT" are treated as general-purpose output devices.

Device No.	Multitask Setting		
	Start by Interrupt from PC CPU		Others
Y11	Task number setting to be started	Task No. 1	General-purpose output (Tasks cannot start.)
Y12		Task No. 2	

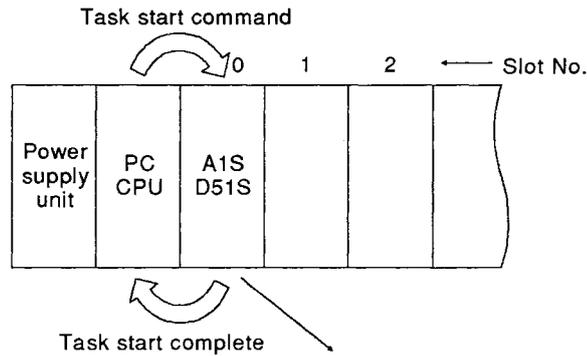
- (c) Turn ON the device number (Y11/Y12) corresponding to the task to be started. The task corresponding to the device (Y11/Y12) which is ON when the task start signal (Y19) is turned ON will start. However, the designation of the ongoing task will be ignored.
- (d) Turn OFF any turned-ON start task No. designation signal (Y11/Y12) after confirming that the designated task is completed. As shown on the following page, the start of a designated task can be confirmed by turning ON the general-purpose input device (X) allocated to the reset (OFF) command with a BASIC program.

Example-

This example shows the procedure for starting a task and for turning OFF the start task No. designation (Y11)/task start signals (Y19).

[Conditions]

- (1) Start task number ... 1
- (2) Start task number designation signal/task start signal reset command X11

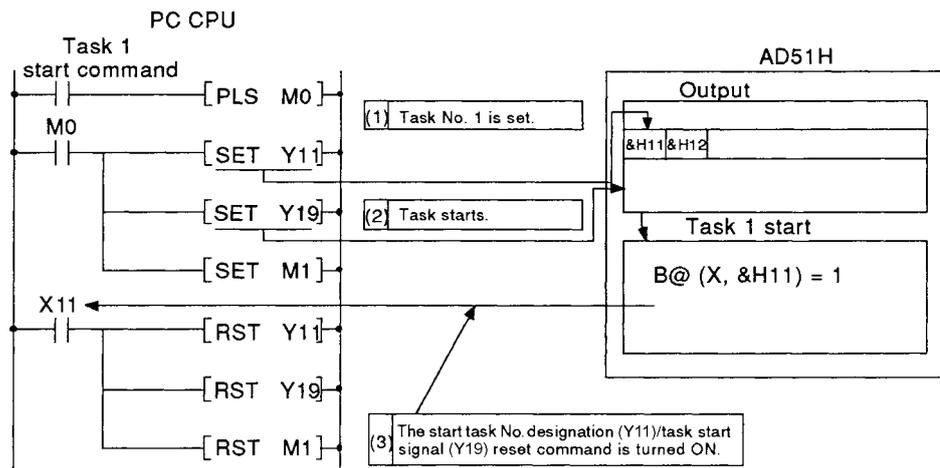


I/O device numbers identified by PC CPU:
X/Y00 to X/Y1F

Values designated by special variable B@:
X/Y&H00 to X/Y&H1F

[Procedure]

- (1) Set task 1. Turn ON Y11 using the sequence program.
- (2) Start the task. Turn ON Y19 using the sequence program.
- (3) Turn off the start task No. designation (Y11)/task start signals (Y19).
..... Turn ON X&H11 using a BASIC program.



(3) Task start signal (Y19)

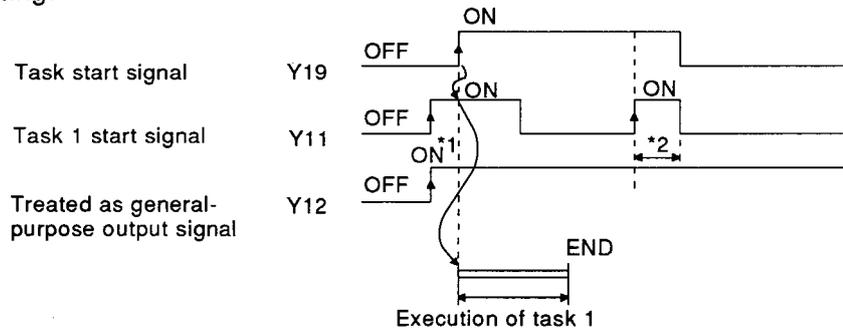
- (a) This signal starts the task designated by a start task No. designation signal (Y11/Y12).
The task start signal cannot be used as a general-purpose output signal even if there is no task to be started by an interrupt from the PC CPU.
- (b) The task start signal (Y19) is valid while the multitask execution start signal (X0B) is ON.
- (c) The designated task ^{*1} starts as soon as the task start signal is turned ON. (Any start task No. designation signal (Y11/Y12) turned ON while the task start signal (Y19) is ON will be ignored.)

Example

This example shows a program for starting task No. 1 from the PC CPU.

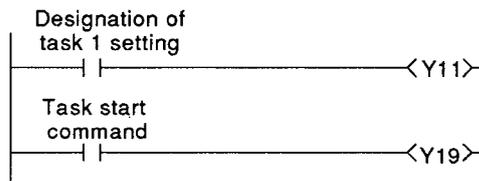
[Condition]

Only task 1 is set to "IT (start by interrupt from PC CPU)" in multitask setting.



- (1) *1: The task does not start because this signal is treated as a general-purpose output signal.
- (2) *2: Task 1 does not start because the task start signal is not turned ON.

[Sequence program]



- (d) While the task designated by a start task No. designation signal (Y11/Y12) is executed, the task start signal is ignored. (The ongoing task will proceed.)

REMARK

*1: The task corresponding to Y11 or Y12, whichever is turned ON, starts. However, any device number whose multitask setting is not "IT (start by interrupt by PC CPU)" will be ignored.

3.8 Relationship between A1SD51S Start Conditions and EEPROM Execution Program Areas

The start condition for each task is set "BOOT" as default.
 To set to any other than "BOOT", change it using the system command SET.
 The storage status of the BASIC program in the program area depends on the set start condition, as shown in Table 3.11.

Table 3.11 Relationship between A1SD51S Start Conditions and EEPROM Execution Program Areas

Start Condition		Area for Storing BASIC Program to be Executed	BASIC Program Read Timing	Reference Section
Multitask Setting	Symbol			
Starting by turning the power on or by reset	START	• Execution program area in EEPROM (designated BASIC task number area)	• When the power is turned ON, or the A1SD51S is reset, in the system mode. • During multitasking in the execution/multitask debug mode (RUN switch: RUN).	3.10.2
Starting by an interrupt from PC CPU	IT	• Execution program area in EEPROM (designated BASIC task number area)		3.10.3
Starting by a start request from another task	BOOT	• Execution program area in EEPROM (designated BASIC task number area)		3.10.4
	ON	• User's FD/HD file area in peripheral device (The BASIC program is stored as a file.)	• When the task is executed in response to a ZSTART command (The BASIC program whose file name is designated by the ZSTART command will be read.)	3.10.4
No multitask setting	OFF	————	• Since there is no multitask setting, the BASIC program is not read.	————

POINT

The difference between "BOOT" and "ON" is explained below:

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

REMARK

For details on the system commands MSAVE and SET, refer to the AD51H-BASIC Programming Manual (Debug and Compile).

3.9 A1SD51S Operation when the Power is Turned ON

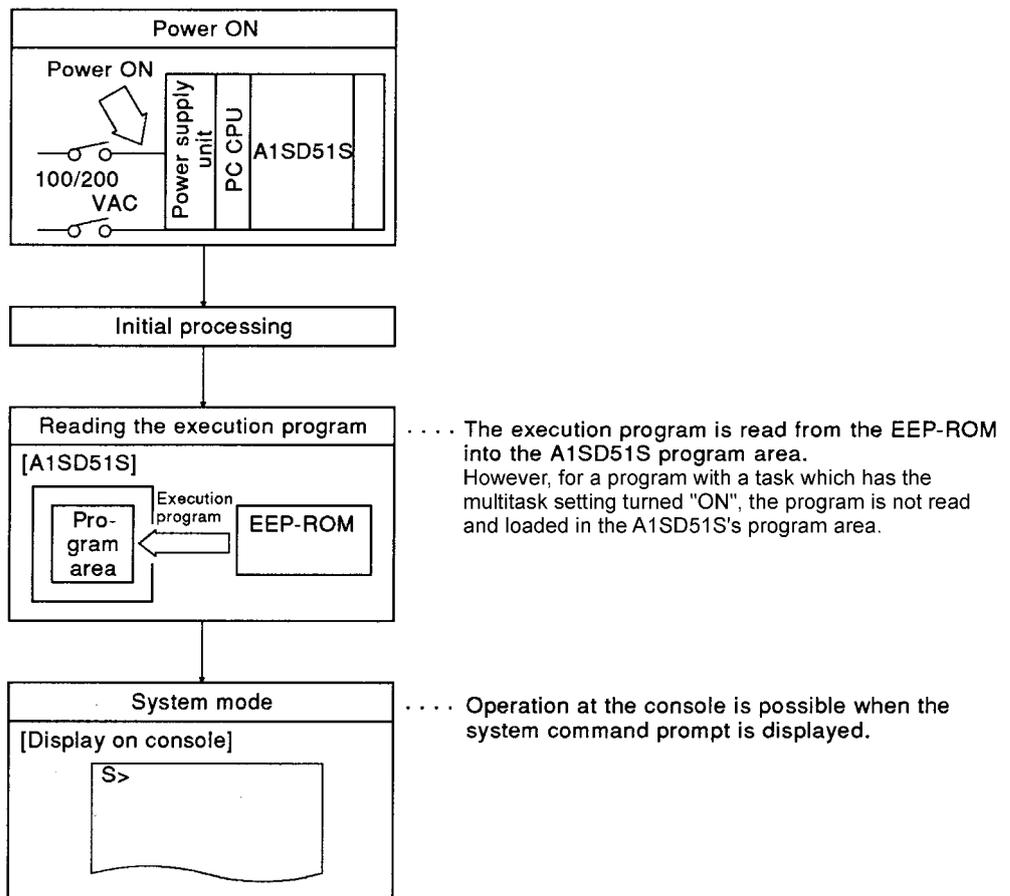
The operation of the A1SD51S when the power is turned ON depends on the mode selected with mode setting switch 1:

- (1) Programming mode
- (2) Multitask debug mode
- (3) Run mode

This section explains the operation of the A1SD51S in each mode set with mode setting switch 1 when the power is turned ON.

3.9.1 In the programming mode

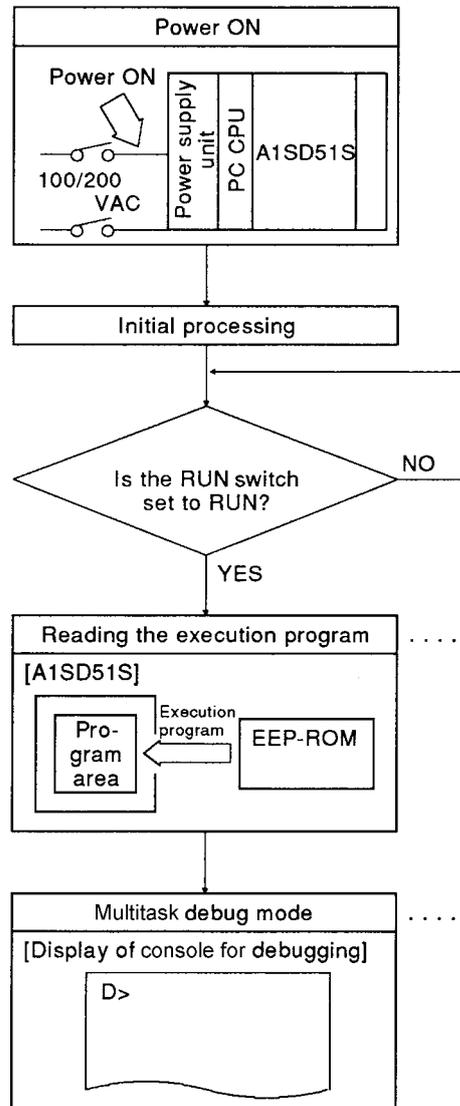
This section explains the case in which mode setting switch 1 on the A1SD51S is set to "Programming mode (4)".
 When powered up or reset, the A1SD51S loads the execution program from the EEP-ROM into its program area (Refer to Section 3.5.1.) and then goes into the system mode.



3.9.2 In the multitask debug mode

This section explains the case in which mode setting switch 1 on the A1SD51S is set to "Multitask debug mode (2 or 3)".

After power-up or reset, with the RUN switch set to RUN, the A1SD51S loads the execution program from the EEP-ROM into its program area (Refer to Section 3.5.1.) and then goes into the multitask debug mode.



... The execution program is read from the EEP-ROM into the A1SD51S program area. However, for a program with a task which has the multitask setting turned "ON", the program is not read and loaded in the A1SD51S's program area.

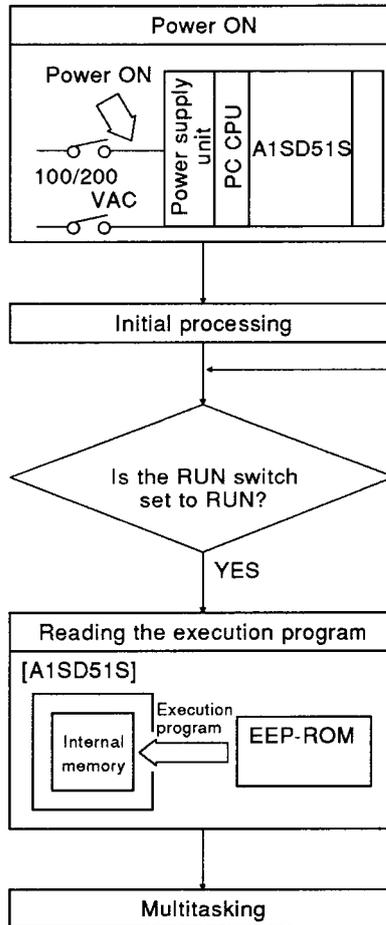
... Multitask debugging by a debugger is enabled when the debug command prompt is displayed. The read execution program is executed according to the multitask setting:

- Execution starts as soon as the power is turned ON or the A1SD51S is reset.
- Execution starts in response to an interrupt from the PC CPU.
- Execution starts in response to a start request from another BASIC program.

3.9.3 In the run mode

This section explains the case in which mode setting switch 1 on the A1SD51S is set to "Run mode (0 or 1)".

After power-up or reset, with the RUN switch set to RUN, the A1SD51S loads the execution program from the EEP-ROM into its program area (Refer to Section 3.5.1.) and then performs the multitask processing.



... The execution program is read from the EEP-ROM into the A1SD51S program area. However, for a program with a task which has the multitask setting turned "ON", the program is not read and loaded in the A1SD51S's program area.

... The read execution program is executed according to the multitask setting:

- Execution starts as soon as the power is turned ON or the A1SD51S is reset.
- Execution starts in response to an interrupt from the PC CPU.
- Execution starts in response to a start request from another BASIC program.

3.10 Executing Tasks (Created BASIC Programs)

This section describes the execution of tasks when the A1SD51S is in the run or multitask debug mode.

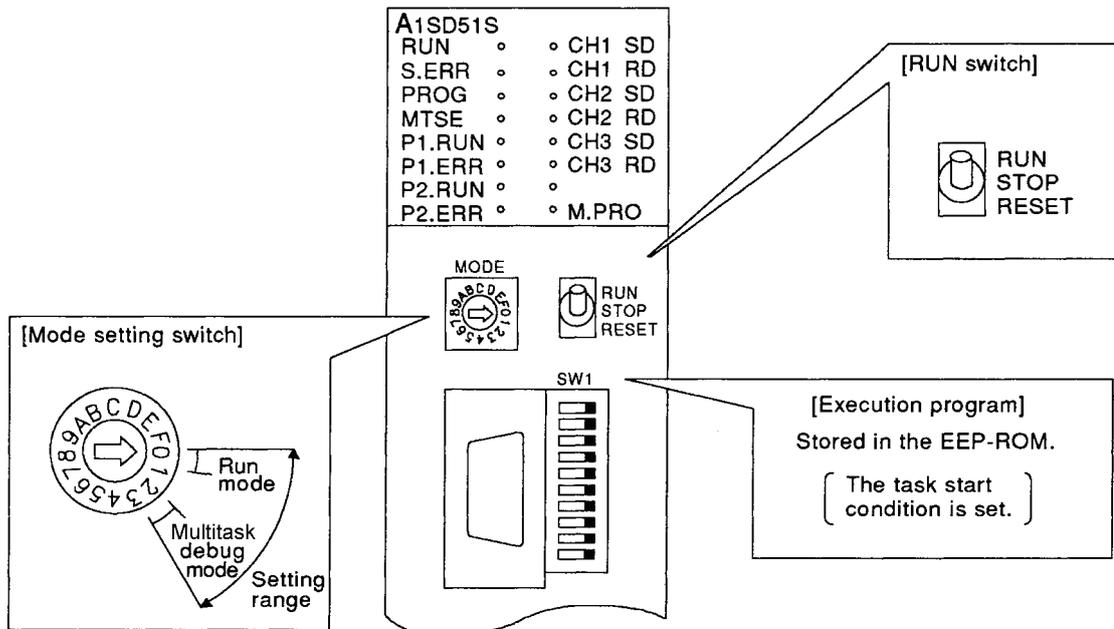
3.10.1 Task execution enable conditions and start conditions

The A1SD51S is ready to execute tasks when the following conditions are met. (An actual task starts according to the start conditions set in multitask setting.)

(1) Task execution enable conditions

The following are the task execution enable conditions:

- (a) A task start condition (Refer to Section 3.8) is set in multitask setting.
- (b) Mode setting switch 1 (Refer to Section 5.3.1) is set to "run mode (0, 1)" or "multitask debug mode (2, 3)".
- (c) The RUN switch is set to "RUN".



(2) Task start conditions

Each task is run in accordance with the set start conditions.

The start conditions that can be set by the A1SD51S are following three types.

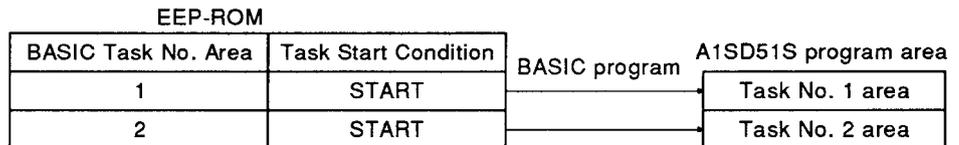
Start Condition	Symbol	Reference Section
Starting by turning the power on or by reset	START	Section 3.10.2
Starting by an interrupt from the PC CPU	IT	Section 3.10.3
Starting by a start request from another task	BOOT, ON	Section 3.10.4

3.10.2 Starting by turning the power on or by reset (Start condition: START)

In this case, "START" is set as the task start condition in multitask setting. Tasks will be executed only when the "execution enable conditions" described in Section 3.10.1 are satisfied.

(1) Processing when the power is turned ON or the A1SD51S is reset

As soon as the power is turned ON or the A1SD51S is reset, the execution program is read from the EEP-ROM into the module's program area and the designated task is executed.



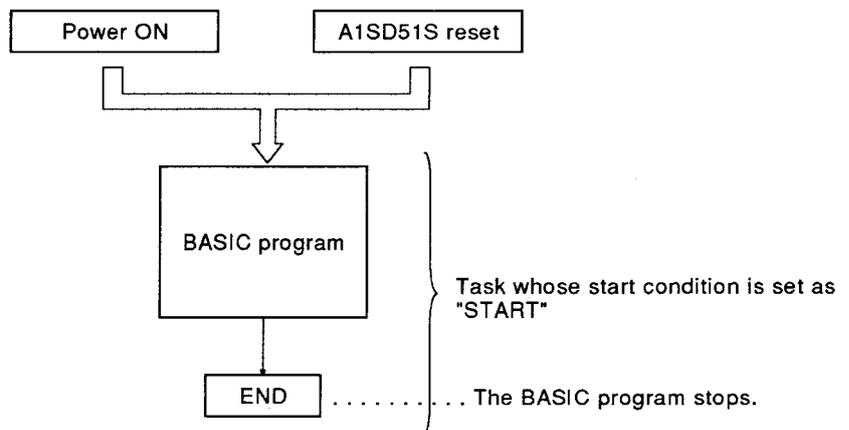
(2) Execution of tasks

Depending on the created BASIC program, a task started by turning ON the power or resetting the A1SD51S is:

- 1) executed only once; or
- 2) continually executed.

(a) Execution once only

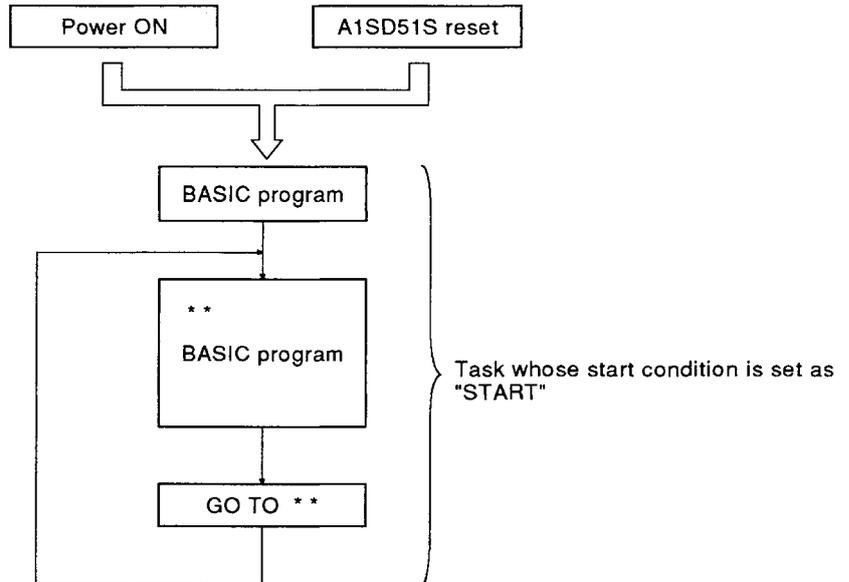
When an "END" command is written at the end of the BASIC program, the task will be executed only once. The task starts when the power is turned ON or the A1SD51S is reset, and stops on execution of the END command.



(b) Continual execution

In this case, a "GOTO" command, instead of the END command, is used at the end of the BASIC program to jump the task to any designated line.

The task starts as soon as the power is turned ON or the A1SD51S is reset, and jumps to the designated line when it reaches the GOTO command. The operations set between the designated line and the GOTO command will thus be repeated.



3.10.3 Starting by an interrupt from the PC CPU (Start condition: IT)

In this case, "IT" is set as the task start condition in multitask setting. Tasks will be executed only when the "execution enable conditions" described in Section 3.10.1 are satisfied and the PC CPU issues a start request.

For the start request from the PC CPU:

- 1) the start task No. designation signal (Y11/Y12) or
- 2) the task start signal (Y19)

is used.

(For details on the start task No. designation signal (Y11/Y12) and task start signal (Y19), refer to Section 3.7.3.)

(1) Processing when the power is turned ON or the A1SD51S is reset

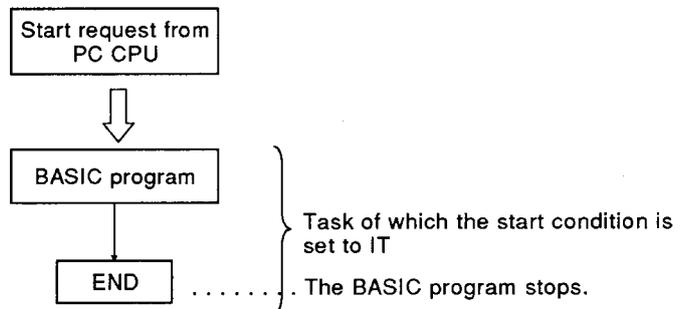
As soon as the power is turned ON or the A1SD51S is reset, the BASIC (execution) program is read from the EEP-ROM into the module's internal memory.

The read BASIC program will then be executed in response to a start request from the PC CPU.

(2) Execution of tasks

(a) When the power is turned ON or the A1SD51S is reset, the read BASIC program is started by turning ON of the A1SD51S task start signal (Y19) from the PC CPU.

The task whose start task No. designation signal (Y11/Y12) is ON when the task start signal (Y19) is turned ON will be executed.



(b) Any task to be started by an interrupt from the PC CPU requires an "END" command to be executed at the end of the program.

This operation resets the interrupt from the PC CPU, and enables the task to receive the following interrupt.

3.10.4 Starting by a start request from another task (Start condition: BOOT, ON)

In this case, "BOOT" or "ON" is set as the task start condition in multitask setting.

Tasks will be executed only when the "execution enable conditions" described in Section 3.10.1 are satisfied and another BASIC program issues a start request with the ZSTART command. (For details of the ZSTART command, refer to the AD51H-BASIC Programming Manual (Command).)

(1) Processing when the power is turned ON or the A1SD51S is reset

- (a) When "BOOT" is set as the task start condition, the BASIC (execution) program is read from the EEP-ROM into the A1SD51S's internal memory as soon as the power is turned ON or the module is reset. The read BASIC program will be executed when another task issues a start request.
- (b) When "ON" is set as the task start condition, the BASIC (execution) program is not read from the EEP-ROM into the A1SD51S's internal memory even after the power is turned ON or the module is reset. The BASIC program will be read out of the user's FD/HD file area and executed in response to a start request from another task.

(2) Execution of a task

(a) Starting a task

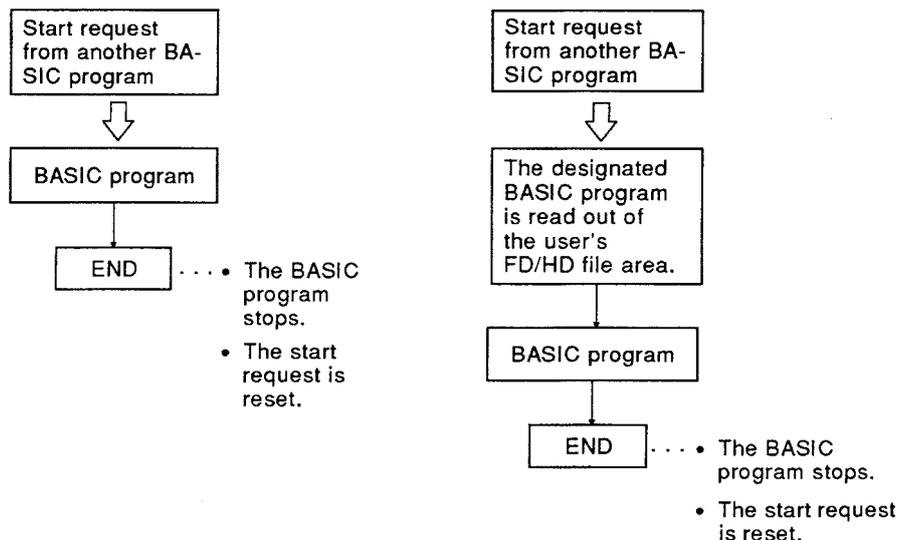
A task starts when it is designated by the "ZSTART" command from another BASIC program.
The designation method of the "ZSTART" command varies depending on the start condition.

BOOT : Designated by <number> of the task No. area

ON : Designated by <number> of the task No. area and by "<drive No.>:\<system name>\<file name>" of the BASIC program to be read

[When the task start condition is "BOOT"]

[When the task start condition is "ON"]



- (b) Any task to be started in response to a start request from another BASIC program requires an "END" command to be executed at the end of the program. This operation resets the start request, and enables the task to receive the following start request.

3.11 Execution of BASIC Programs by Multitasking

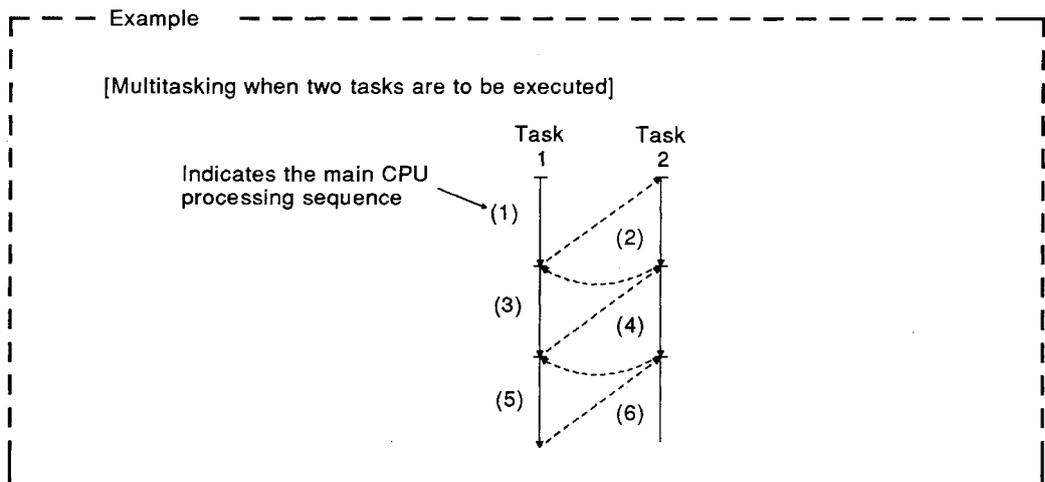
This section describes how to execute A1SD51S BASIC programs.

3.11.1 Multitasking

The A1SD51S is a module that operates in accordance with user-created BASIC programs.

The A1SD51S has one main CPU (central processing unit).

Multitasking accomplishes parallel operations by successively switching the execution between multiple BASIC programs (up to 2 tasks).



Under any of the following conditions, the A1SD51S switches the execution of BASIC programs during multitasking.

- when the preset "task schedule time (50 msec)" has elapsed;
- when data has been transmitted to the screen, keyboard, disk, peripheral or external device in response to an I/O (data communications) command;
- when a BASIC program execution control (suspension, stop, termination) command is executed;
- when a multitask control (synchronization of execution, start, priority change) command has been executed; or
- when the suspension of a program has been canceled (e.g. when an I/O operation is completed, or the designated time has elapsed).

REMARK

For details of multitasking and commands which switch program execution, refer to AD51H-BASIC Programming Manual (Command).

3.12 BASIC Program Priority

The A1SD51S assigns priority to created BASIC programs. When started, all BASIC programs are given the same priority. This means that all the created BASIC programs have the same execution priority.

However, the priority can be changed by the ZURGENCY command. Once the priority is changed, the OS will not perform scheduling unless a program execution switch is executed, and only the highest priority BASIC program will run. (The lower priority BASIC programs will not be executed.)

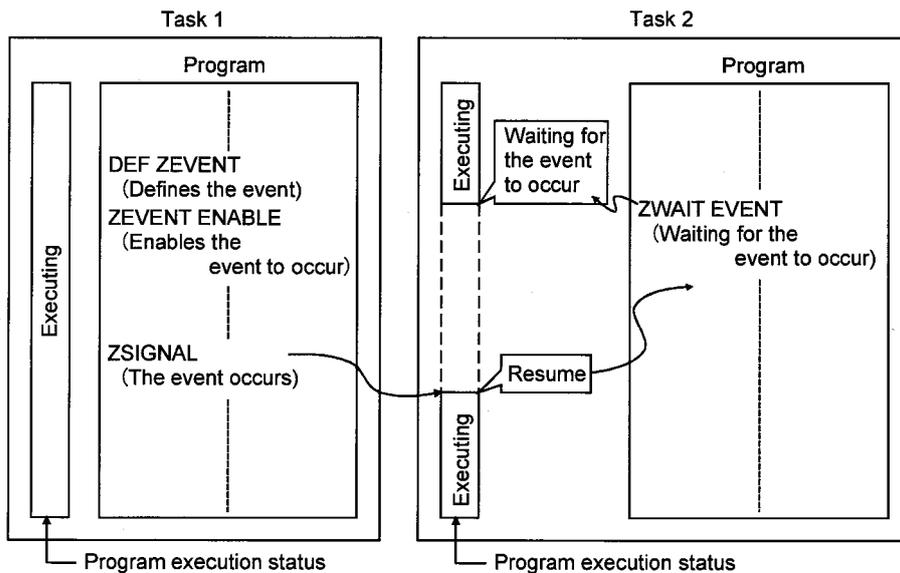
3.12.1 Synchronization of execution between BASIC programs (event control)

Multitasking runs BASIC programs independently. In data communications between two BASIC programs, the receiving program must confirm that the source program has transmitted data before reading it.

Event control synchronizes the execution of BASIC programs between which data will be communicated. In event control, the data-receiving BASIC program, which has been on standby for a signal (WAIT) state, starts running in response to a signal (event) after the source BASIC program completes data transmission. This enables the receiving BASIC program to correctly read the data transmitted from the source program.

Event control is executed by the following commands.

- DEF ZEVENT Defines an event.
- ZEVENT ENABLE Makes the occurrence of the event valid.
- ZWAIT EVENT Waits for the event to be executed.
- ZSIGNAL Executes the event.



REMARK

For details of event control and the available commands, refer to the AD51H-BASIC Programming Manual (Command).

3.12.2 Sharing of devices (sources) by multitasking

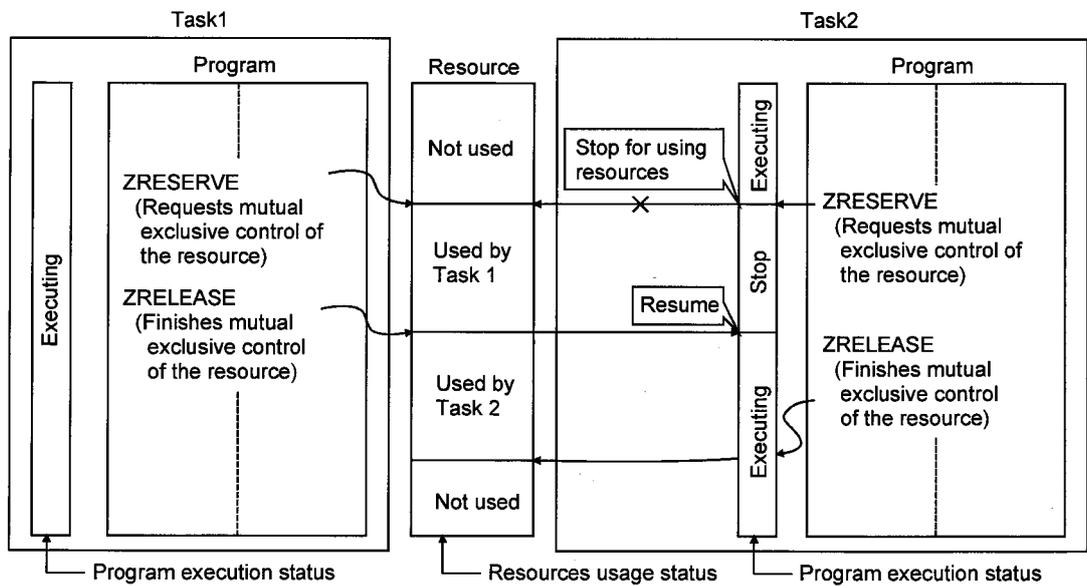
Two or more BASIC programs cannot share and use the same source simultaneously.

Exclusive control prevents other BASIC programs from using the source which a BASIC program uses.

This exclusive source control is executed according to the rules determined by programmers. If these rules are not observed, exclusive source control is impossible.

Exclusive source control is executed by using the following commands.

- ZRESERVE Requests exclusive source control.
- ZRELEASE Terminates exclusive source control.



REMARK

"Source" herein means all hardware and software available for BASIC programs, such as the screen, printer, file, disk and memory.

4. A1SD51S FUNCTIONS

This section describes the functions of the A1SD51S.

4.1 Function List

Table 4.1 shows the list of the A1SD51S functions.

Table 4.1 A1SD51S Functions

Function		Description	Reference Section
BASIC program functions	Sub CPU function	Uses a BASIC program to perform complex numeric and function calculations which may result in a prolonged PC CPU scan time. • Collection, analysis and correction of measurement data • Functional operations, such as sines, logarithms and square roots	AD51H-BASIC Programming Manual (Command).
	Monitor display function	Displays working states (progress of production, operation status, error, etc.) on the console or terminal connected to the A1SD51S.	4.3, 4.4
	Key input function	Allows entry of production schedules, production output, operation and set data at the keyboard of a console or terminal.	4.3, 4.4
	Printer function	Prints production plans, production records, daily reports, errors, plan data, inspection results, test records, etc. on a printer connected to the A1SD51S.	4.5
	Data input function	Allows data entry through a bar code reader or magnetic card reader connected to the A1SD51S. (Since a BASIC program enables the A1SD51S to transmit and receive data in a free format, it can perform data communications in the protocol the communicating device uses.) • Input of production lot numbers, production names, quantities, etc. • Collection of measurement values and test data	—
	External device connection function	Establishes data communications by using a BASIC program and connecting a computer to the RS-232 or RS-422/485 interface of the A1SD51S.	4.6
	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.	4.9
Offline programming function		Allows the PC/AT alone to create, correct and store BASIC programs on a user's FD and to print them.	Chapter 6
Online programming function		Creates, executes and corrects BASIC programs on a console connected to the A1SD51S in accordance with system commands. Registers and reads execution programs to/from a EEPROM in accordance with system commands.	AD51H-BASIC Programming Manual (Debug and Compile)
Multitask debugging function		Debugs BASIC programs on a console and a debugger connected to the A1SD51S while executing them by multitasking.	

4.2 Communications with a Console or Terminal

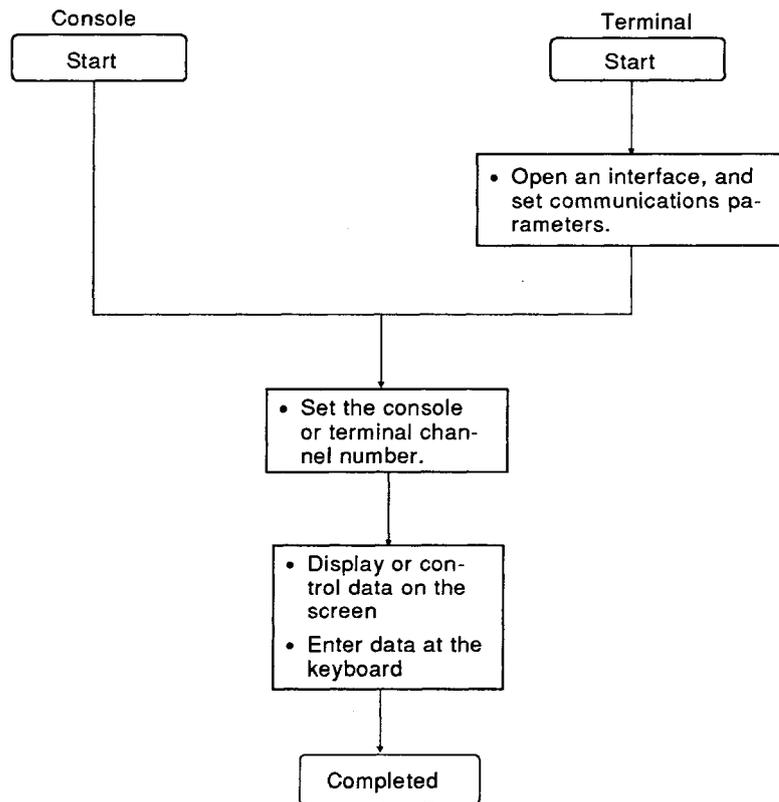
Data can be displayed on, or entered at a keyboard to, an PC/AT connected to CH.3 (RS-422/485 interface) of the A1SD51S, or a VG-620, VT-382 or PC/AT connected to CH.1 or CH.2 (RS-232 interface).

The PC/AT connected to CH.3 and the VG-620, VT-382 or PC/AT connected to CH.1 serve as a console and a terminal, respectively.

The device (PC/AT, VG-620 or VT-382) connected to the channel selected with mode setting switch 2 of the A1SD51S as the console channel can be used as the console.

A device (PC/AT, VG-620 or VT-382) connected to a channel not set at the A1SD51S as a console channel can be used as a terminal.

The procedure for screen display and entry at a keyboard is different for the console and the terminal, as shown below.



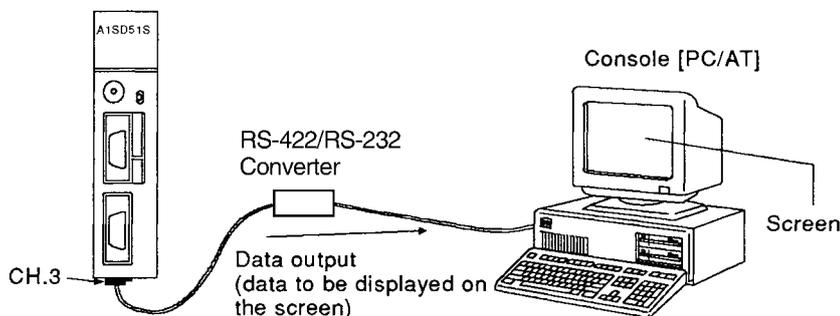
REMARKS

- (1) Refer to Chapter 2 for details on consoles.
- (2) Refer to Section 5.3.1 for console setting using mode setting switch 1.

4.3 Communications with a Console

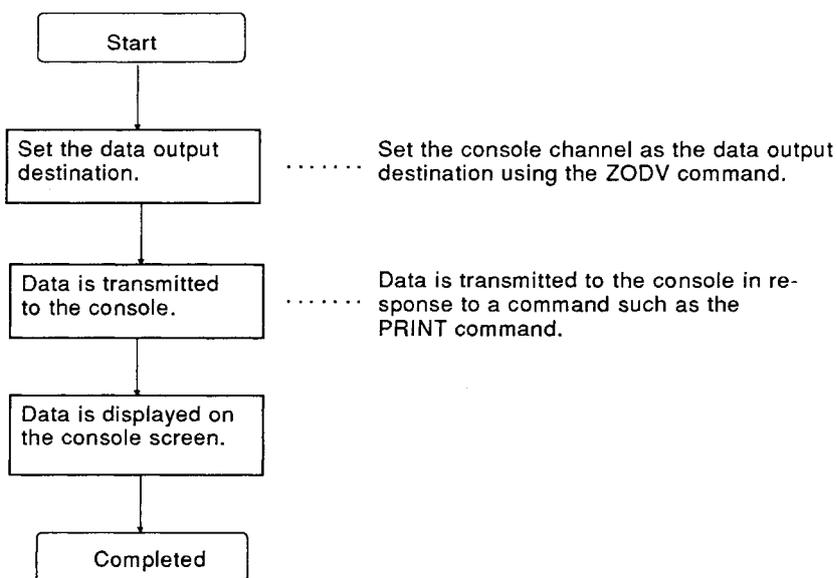
(1) Displaying data on the console screen

This section describes how to display data on the console screen.



[Procedure for displaying data on the console screen]

Data will be displayed on the console screen, as shown below.



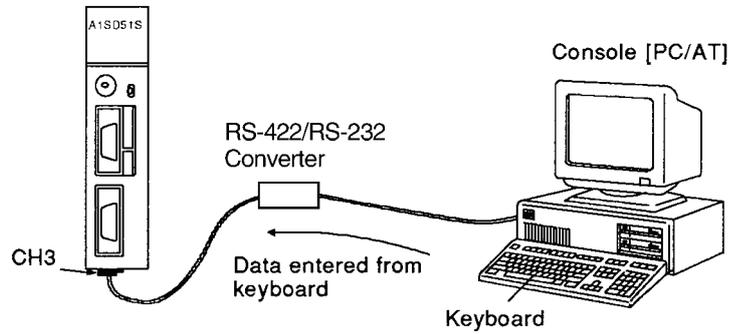
[Commands/Functions used]

The following list shows the commands/functions used to display data on the console screen.

Command/Function	Processing
ZODV	Switches the data output destination.
PRINT	Displays data on the console screen.
PRINT USING	Displays data on the console screen. (The format can be specified.)
LOCATE	Designates the data display position on the console screen.
CLS	Clears data on the console screen.

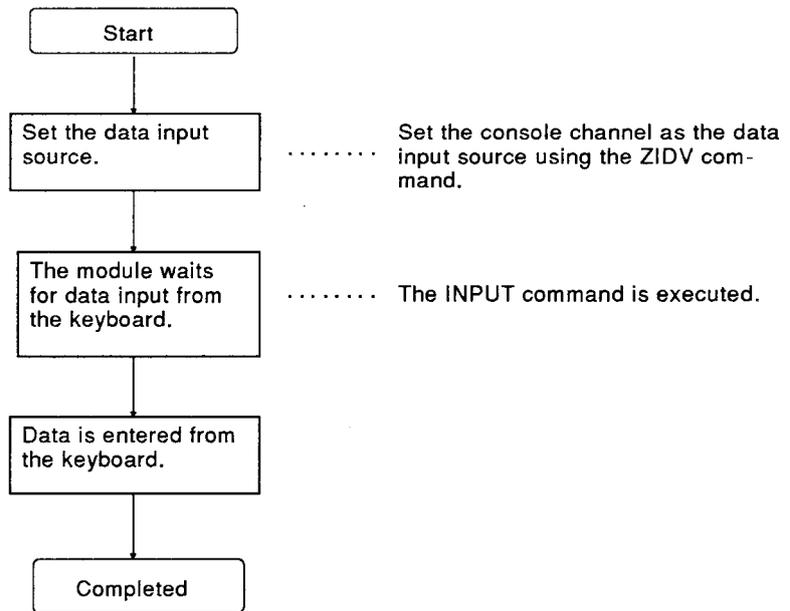
(2) Data input from the console keyboard

This section describes data input from the console keyboard.



[Procedure for entering data from the console keyboard]

Data is entered from the console keyboard in the manner shown below.



[Commands/Functions used]

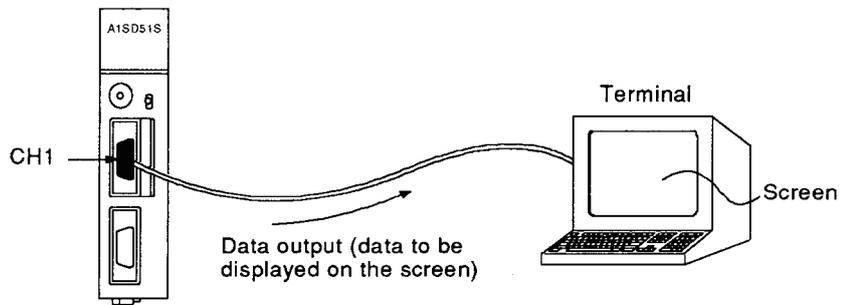
The following lists shows the commands/functions used for data input from the console keyboard.

Command/Function	Processing
ZIDV	Switches the data input source.
INPUT	<ul style="list-style-type: none">• Enables data input from the keyboard.• Character code control codes (0H to 1FH), comma (2CH) and [DEL] (7FH) are not treated as data.
LINE INPUT	<ul style="list-style-type: none">• Enables data input from the keyboard.• Character code control codes (0H to 1FH) and [DEL] (7FH) are not treated as data.
INKEY\$/INPUT\$	<ul style="list-style-type: none">• Enables data input from the keyboard.• Character code control codes (0H, 03H, 13H, 80H, FDH to FFH) are not treated as data.

4.4 Communications with a Terminal

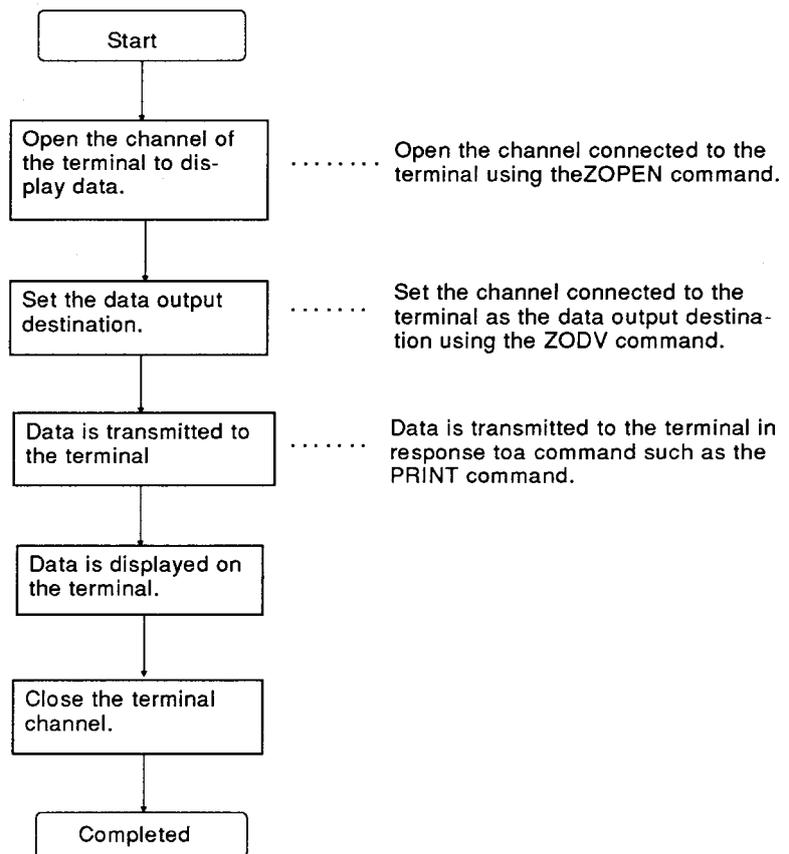
(1) Displaying data on the terminal screen

This section describes how to display data on the terminal screen.



[Procedure for displaying data on the terminal screen]

Data is displayed on the terminal screen in the manner shown below.



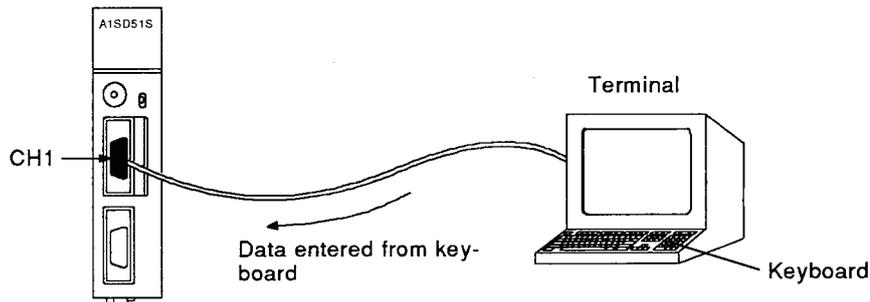
[Commands/Functions used]

The following list shows the commands/functions to be used to display data on the terminal screen.

Command/Function	Processing
ZOPEN	Opens the interface to make the module ready for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets communications parameters for the open interface, and reads the interface status.
ZODV	Switches the data output destination.
PRINT	Displays data on the terminal screen.
PRINT USING	Displays data on the terminal screen. (The format can be specified.)
LOCATE	Designates the data display position on the terminal screen.
CLS	Clears data on the terminal screen.

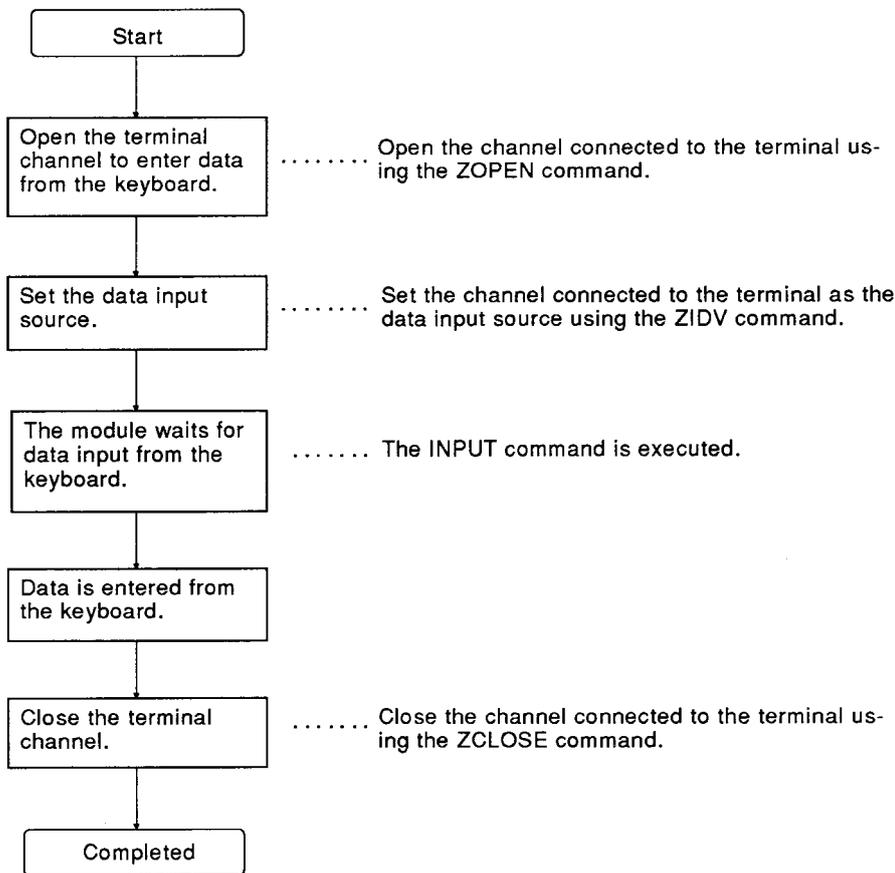
(2) Data input from the terminal keyboard

This section describes data input from the terminal keyboard.



[Procedure for entering data from the terminal keyboard]

Data is entered from the terminal keyboard in the manner shown below.



[Commands/Functions used]

The following list shows the commands/functions used for data input from the terminal keyboard.

Command/Function	Processing
ZOPEN	Opens the interface to make the module ready for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets communications parameters for the open interface, and reads the interface status.
ZIDV	Switches the data input source.
INPUT	<ul style="list-style-type: none">• Enables data input from the keyboard.• Character code control codes (0_H to 1F_H), comma (2C_H) and [DEL] (7F_H) are not treated as data.
LINE INPUT	<ul style="list-style-type: none">• Enables data input from the keyboard.• Character code control codes (0_H to 1F_H) and [DEL] (7F_H) are not treated as data.
INKEY&INPUT\$	<ul style="list-style-type: none">• Enables data input from the keyboard.• Character code control codes (0_H, 03_H, 13_H, 80_H, FD_H to FF_H) are not treated as data.

4.5 Printing out Data at a Printer

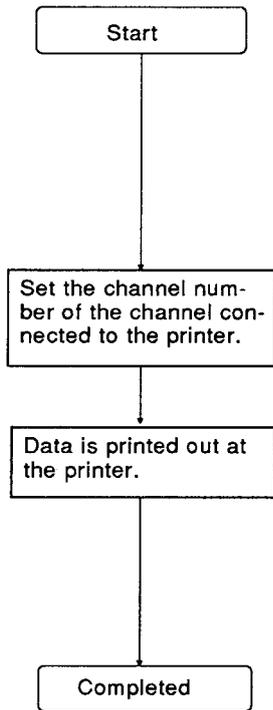
Data can be printed out at a printer connected to the RS-232 interface of the A1SD51S.

Data can also be printed out at a printer connected to a PC/AT used as the console.

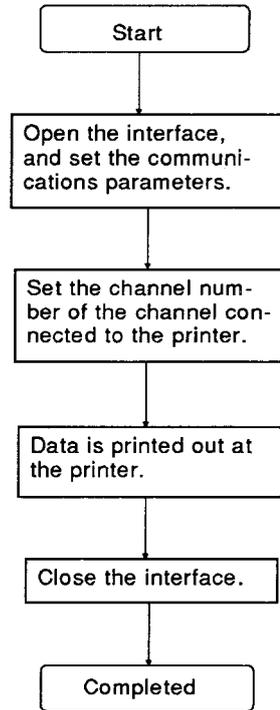
The printing procedure depends on:

- (1) the printer connected to an interface other than the console interface; or
- (2) the printer connected to the console (PC/AT).

(Printer connected to the console)

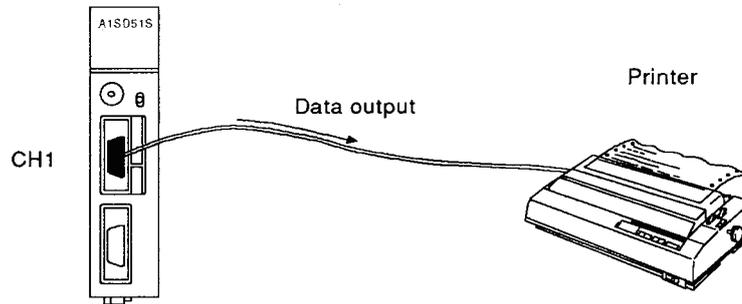


(Printer connected to an interface other than the console interface)



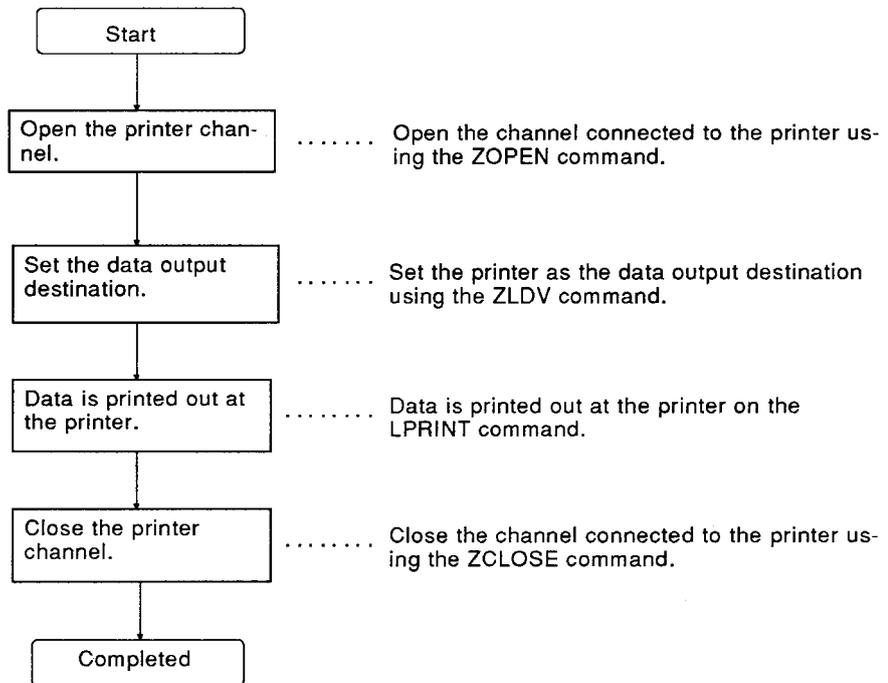
4.5.1 Printing out data at a printer connected to an interface not set as a console interface

This section describes how to print out data at a printer which is connected to an interface not set as a console interface.



[Printout procedure]

Data is printed out at a printer connected to an interface not set as a console interface in the manner shown below.



[Commands/Functions used]

The following list shows the commands/functions to be used to print out data on the printer connected to an interface unequipped with the console.

Command/Function	Processing
ZOPEN	Opens the interface to make the module ready for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets communications parameters for the open interface, and reads the interface status.
ZLDV	Switches to the channel number of the interface connected to the printer.
LPRINT	Enables the designated printer to print out data.
LPRINT USING	Enables the designated printer to print out data. (The format can be specified.)
LLIST	Enables the designated printer to print out a program list.

4.5.2 Printing out data at a printer connected to a console (PC/AT)

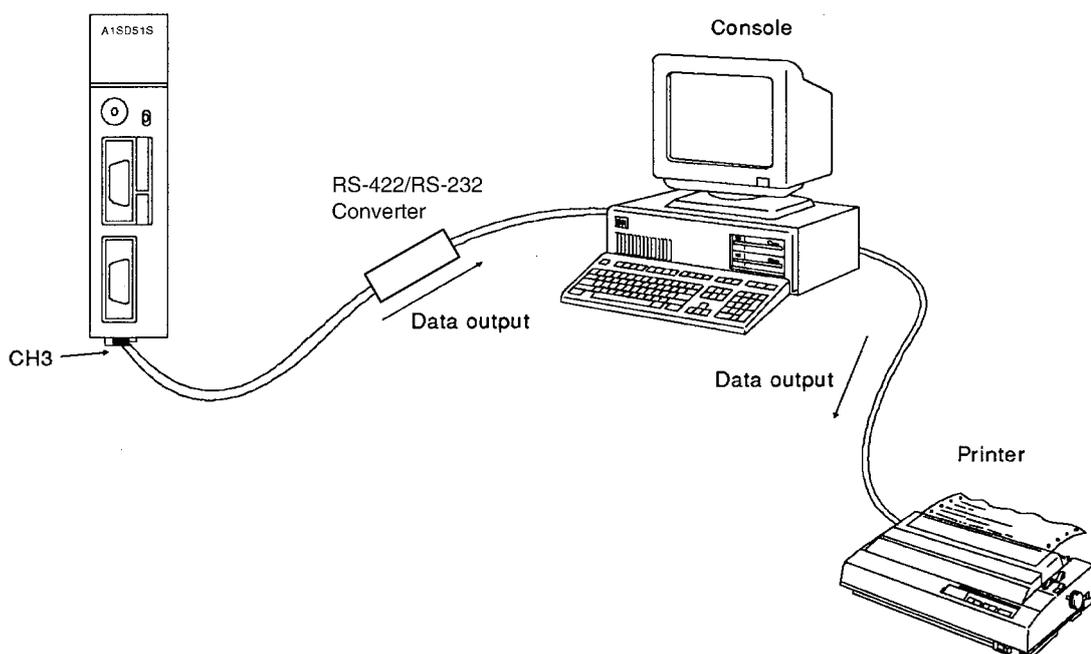
The method of printout by connecting the printer to PC/AT with the interface set on the console is described.

Printer settings are required for printout with PC/AT.

Printer settings for PC/AT are made on the OS environment settings.

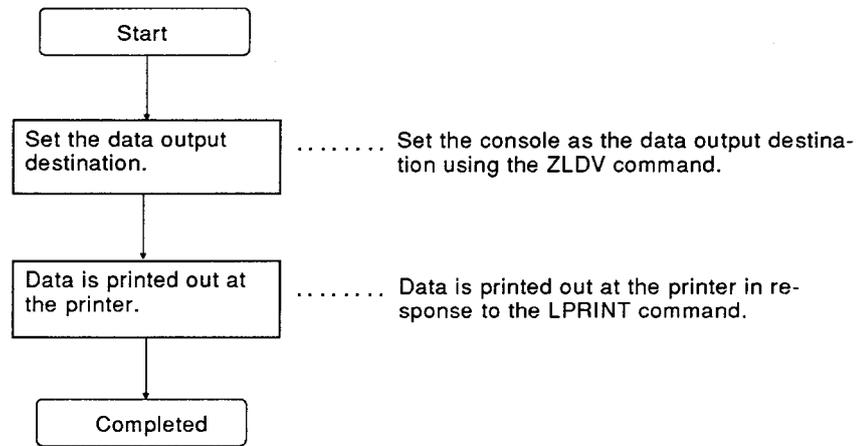
POINT

It is available only for the online programming using SW1IX-AD51HP.



[Printout procedure]

Data is printed out at the printer connected to the console in the manner shown below.



[Commands/Functions used]

The following list shows the commands/functions to be used to print out data on the printer connected to the console.

Command/Function	Processing
ZLDV	Switches the data output destination to the printer connected to the console.
LPRINT	Enables the designated printer to print out data.
LPRINT USING	Enables the designated printer to print out data. (The format can be specified.)
LLIST	Enables the designated printer to print out a program list.

4.6 Communications with a Device Other than the Console, Terminal or Printer

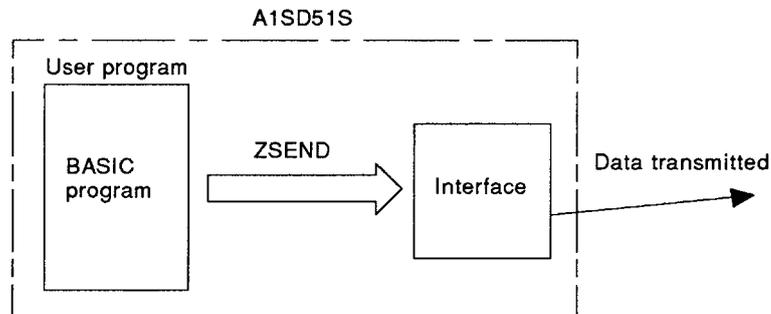
The A1SD51S communicates data with an external device *1 connected to its RS-232 or RS-422/485 interface.

4.6.1 Transmitting data

This section describes how the A1SD51S transmits data to a device other than the console, terminal or printer.

Data is transmitted by following the steps indicated below:

- (1) opening the interface
- (2) preparing the data to be transmitted
- (3) setting the number of bytes and time-out duration of the data to be transmitted
- (4) transmitting the data in response to the ZSEND command
- (5) closing the interface



[Commands/Functions used]

The following list shows the commands/functions used to transmit data from the A1SD51S to a device other than the console, terminal or printer. *2

Command/Function	Processing
ZOPEN	Opens the interface to make the module ready for communications with an external device.
ZCNTL	Sets communications parameters for the open interface, and reads the interface status.
ZSEND	Transmits data from the designated interface.
ZCLOSE	Closes the interface.

REMARK

*1: The console, the terminal and the printer are excluded.
 Communications with the console or terminal Refer to Section 4.3, 4.4.
 Printout at the printer Refer to Section 4.5.

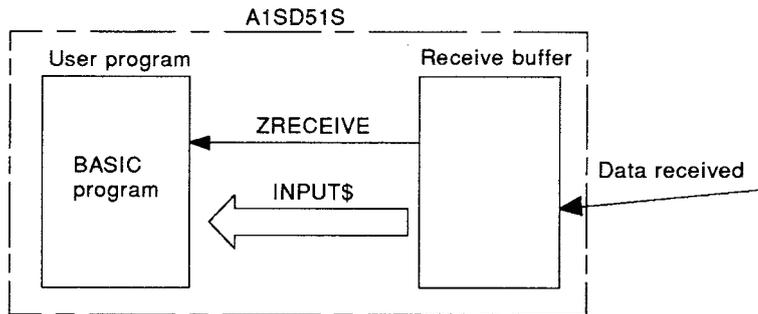
*2: For details of the commands and functions, refer to the AD51H-BASIC Programming Manual (Command).

4.6.2 Receiving data

This section describes how the A1SD51S receives data from a device other than the console or terminal.

Data is received by following the steps indicated below:

- (1) opening the interface
- (2) receiving data
- (3) storing the received data in variables using the ZRECEIVE command and the INPUT\$ function.
- (4) closing the interface.



[Commands/Functions used]

The following list shows the commands/functions to be used to enable the A1SD51S to receive data transmitted from a device other than the console or terminal.

Command/Function	Processing
ZOPEN	Opens the interface to make the module ready for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets communications parameters for the open interface, and reads the interface status.
INPUT\$	<ul style="list-style-type: none"> • Stores data received from the interface in designated variables. • Character code control codes (0H, 03H, 13H, 80H, FDH to FFH) are not treated as data.
ZRECEIVE	<ul style="list-style-type: none"> • Stores data received from the interface in designated variables.

REMARK

For details of the commands and functions, refer to the AD51H-BASIC Programming Manual (Command).

4.7 Communications with the PC CPU or a Special-Function Module

The A1SD51S is capable of writing data to/reading data from the PC CPU by means of a BASIC program.

The types of data that can be written to or read from the PC CPU are listed below.

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	PC CPU remote RUN/STOP	O	X

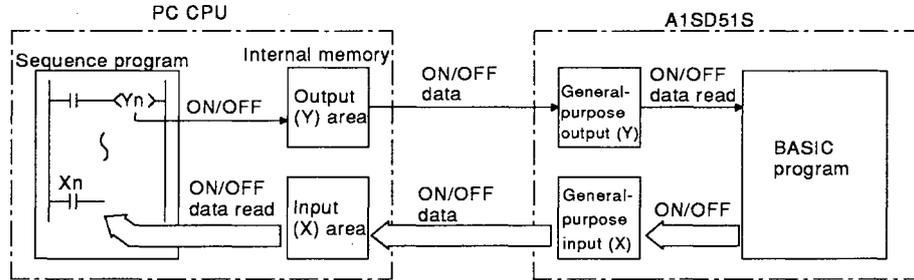
O: executable/X: not executable

The A1SD51S can also write data to/read data from the buffer memory of a special-function module by using a BASIC program.

Among the above PC CPU data, writing and reading of 1. device memory and 2. extension file register, as well as reading data from/writing data to the buffer memory of a special-function module, are described later in this manual.

4.7.1 ON/OFF data communications between the A1SD51S and the PC CPU using general-purpose I/O (X/Y) signals

Using its general-purpose I/O devices, the A1SD51S can communicate ON/OFF data with the PC CPU.



(1) Transmitting data from the PC CPU to the A1SD51S

ON/OFF data is transmitted from the PC CPU to the A1SD51S by general-purpose output (Y) devices.

Using the BASIC program and a special variable B@, the A1SD51S reads and utilizes the general-purpose devices turned ON or OFF by the PC CPU.

(2) Transmitting data from the A1SD51S to the PC CPU

ON/OFF data is transmitted from the A1SD51S to the PC CPU by general-purpose input (X) devices.

Using its sequence program, the PC CPU reads and utilizes the general-purpose devices turned ON or OFF by the A1SD51S BASIC program.

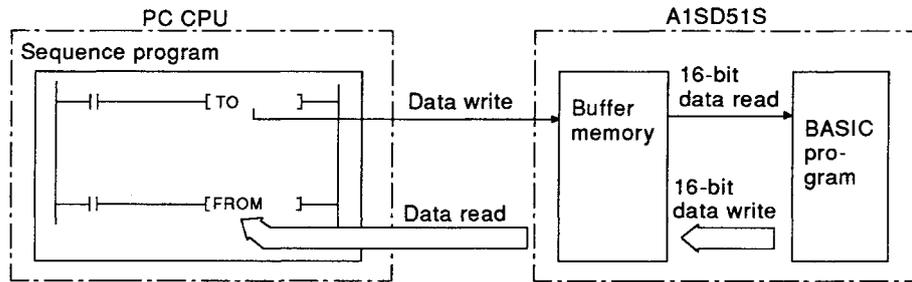
[Special variables used]

The following list shows the special variables used to read the ON/OFF data on general-purpose input (X) and output (Y) devices:

Special Variable	Processing
B@ (X, expression)	Turns A1SD51S general-purpose input (X) devices ON or OFF.
B@ (Y, expression)	Reads the ON/OFF data of A1SD51S general-purpose output (Y) devices.

4.7.2 Communicating 16-bit data using the buffer memory

Using its buffer memory, the A1SD51S can communicate 16-bit data with the PC CPU.



(1) Transmitting data from the PC CPU to the A1SD51S

Using its BASIC program, the A1SD51S reads and utilizes 16-bit data from its buffer memory, which is written by the PC CPU sequence program (TO instruction).

(2) Transmitting data from the A1SD51S to the PC CPU

Using its sequence program (FROM instruction), the PC CPU reads and utilizes 16-bit data from the A1SD51S buffer memory, which is written by the BASIC program.

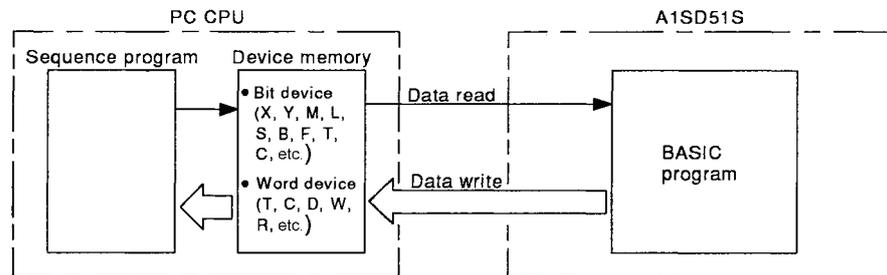
[Commands/Functions used]

The following list shows the commands/functions used to read 16-bit data from, and write 16-bit data to, the A1SD51S buffer memory.

Command/Function	Processing
GETMEM	Reads data from the A1SD51S buffer memory.
PUTMEM	Writes 16-bit data to the A1SD51S buffer memory.

4.7.3 Reading data from/writing data to the PC CPU device memory

The A1SD51S BASIC program enables data to be directly read from and written to the PC CPU device memory.



See the Programming Manual (Command) for the details of the device memories to and from which data can be written and read by the BASIC program.

[Commands/Functions used]

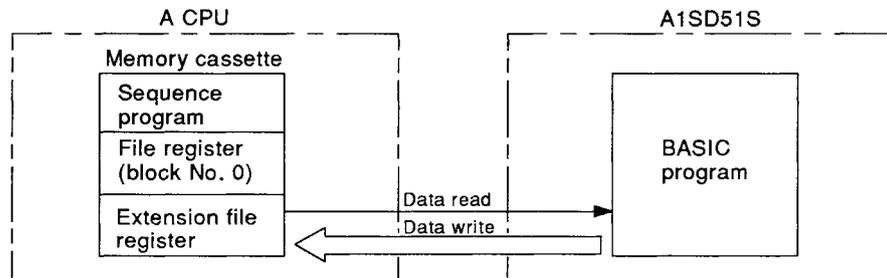
The following list shows the commands/functions used to read and write data from/to the PC CPU device memory.

Command/Function	Processing
PCRD	Reads data from the PC CPU.
PCWT	Writes data to the PC CPU.

4.7.4 Reading data from/writing data to the PC CPU extension file register

The A1SD51S BASIC program enables data to be directly read from and written to the PC CPU extension file register (file register from block No. 1 onward).

See Section 4.7.3 concerning writing of data to and reading data from a file register (block No. 0).



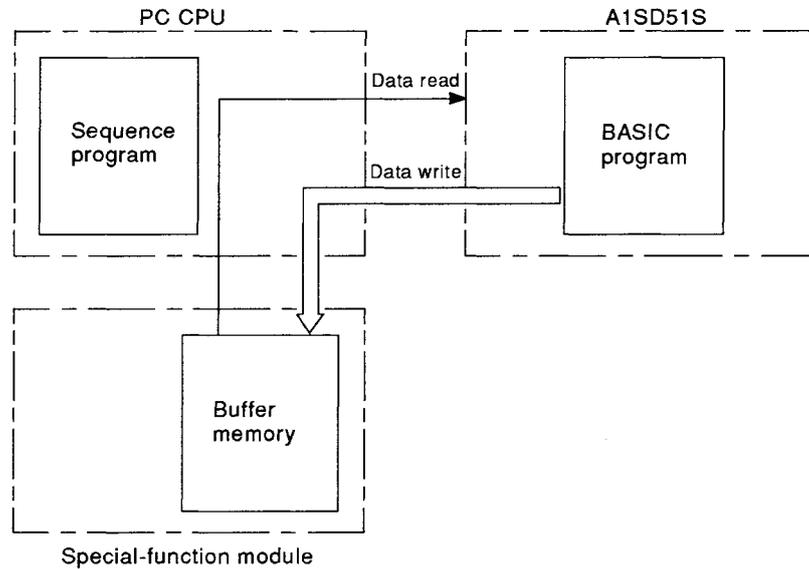
[Commands/Functions used]

The following list shows the commands/functions used to read data from/write data to the PC CPU extension file register.

Command/Function	Processing
PCRD	Reads data from the PC CPU.
PCWT	Writes data to the PC CPU.

4.8 Reading Data from/Writing Data to the Buffer Memory of a Special-Function Module

The A1SD51S BASIC program enables data to be read from and written to the buffer memory of a special-function module.



[Commands/Functions used]

The following list shows the commands/functions used to read data from and write data to the buffer memory of another special-function module:

Command/Function	Processing
PCRD	Reads data from the PC CPU.
PCWT	Writes data to the PC CPU.

REMARK

For details on the commands and functions above, refer to the AD51H-BASIC Programming Manual (Command).

4.9 Using Clock Data

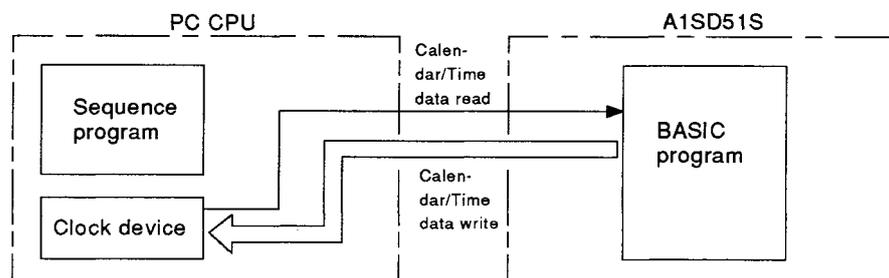
The A1SD51S does not have a clock function.

To use a clock function with the A1SD51S, read the calendar data (year, month, day, day of the week) or time data (hour, minute, second) from a PC CPU equipped with a clock function using the module's BASIC program.

The PC CPUs without clock function are shown below:

- A0J2CPU
- A0J2HCPU
- A1CPU
- A3HCPU
- A2CPU(-S1)
- A3MCPUCPU
- A3CPU

When using the clock function of a PC CPU, the calendar or time data must be written to the clock device beforehand.



[Commands/Functions used]

The following list shows the commands/functions used to read calendar and time data from, and write calendar and time data to, the PC CPU.

Command/Function	Processing
DATE\$	Reads calendar data from/writes calendar data to the PC CPU clock device.
TIME\$	Reads time data from/writes time data to the PC CPU clock device.

4.10 Data Communications between Tasks

During multitasking, the variables and arrays of each task can be referred to only by the corresponding BASIC program.

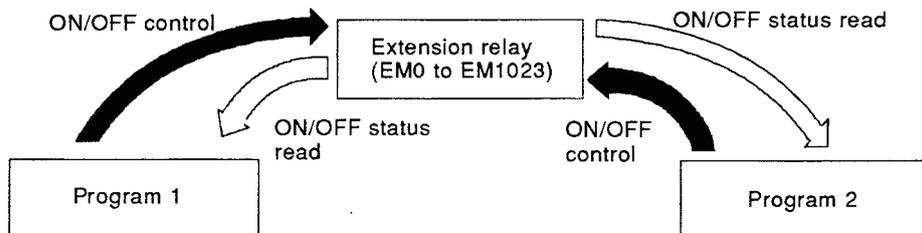
The following are used for data communications between BASIC programs during multitasking.

- (1) Extension relay (EM) ON/OFF data
- (2) Extension register (ED)] Data
- (3) Common memory]

4.10.1 ON/OFF data communications by the extension relay (EM)

The A1SD51S extension relay (EM) is a device capable of reading ON/OFF control data and ON/OFF status data from all tasks.

(For details of the extension relay, refer to Section 3.5.6.)



The A1SD51S can use 1024 extension relays, EM0 to EM1023.

[Command/Function used]

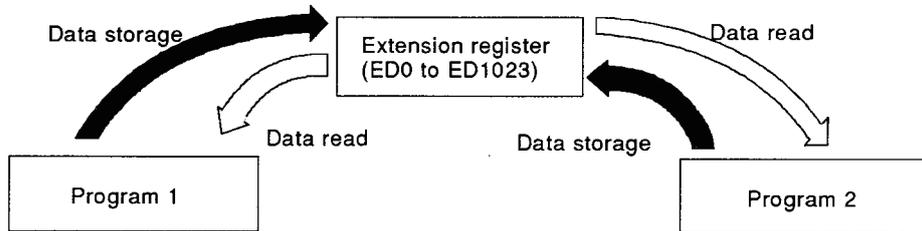
The following list shows the commands/functions used to communicate ON/OFF data using an extension relay.

Command/Function	Processing
B@	Reads extension relay ON/OFF control data and ON/OFF status data.

4.10.2 Data communications by the extension register (ED)

The A1SD51S extension register (ED) is a device capable of storing and reading data from all tasks.

(For details of the extension register, refer to Section 3.5.4.)



The A1SD51S can use 1024 extension registers, ED0 to ED1023.

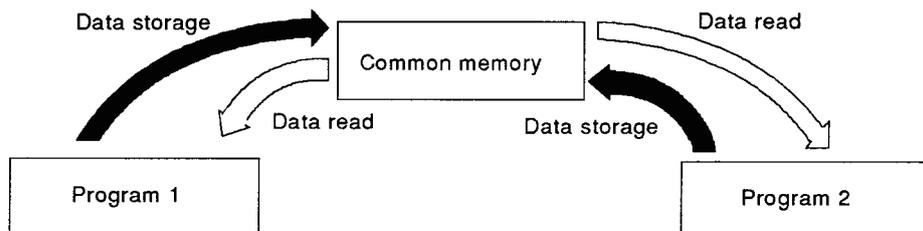
[Commands/Functions used]

The following list shows the commands/functions used for data communications using an extension register.

Command/Function	Processing
W@	Stores data in, and reads data from, the extension register.
GETMEM	Reads data from the extension register.
PUTMEM	Stores data in the extension register.

4.10.3 Data communications using the common memory

The A1SD51S buffer memory is a common memory capable of storing and reading data from all tasks. (For details on the common memory, refer to Section 3.5.3.)



The A1SD51S can use 4 Kwords (8 Kbytes) of common memory.

[Commands/Functions used]

The following list shows the commands/functions used for data communications using the common memory.

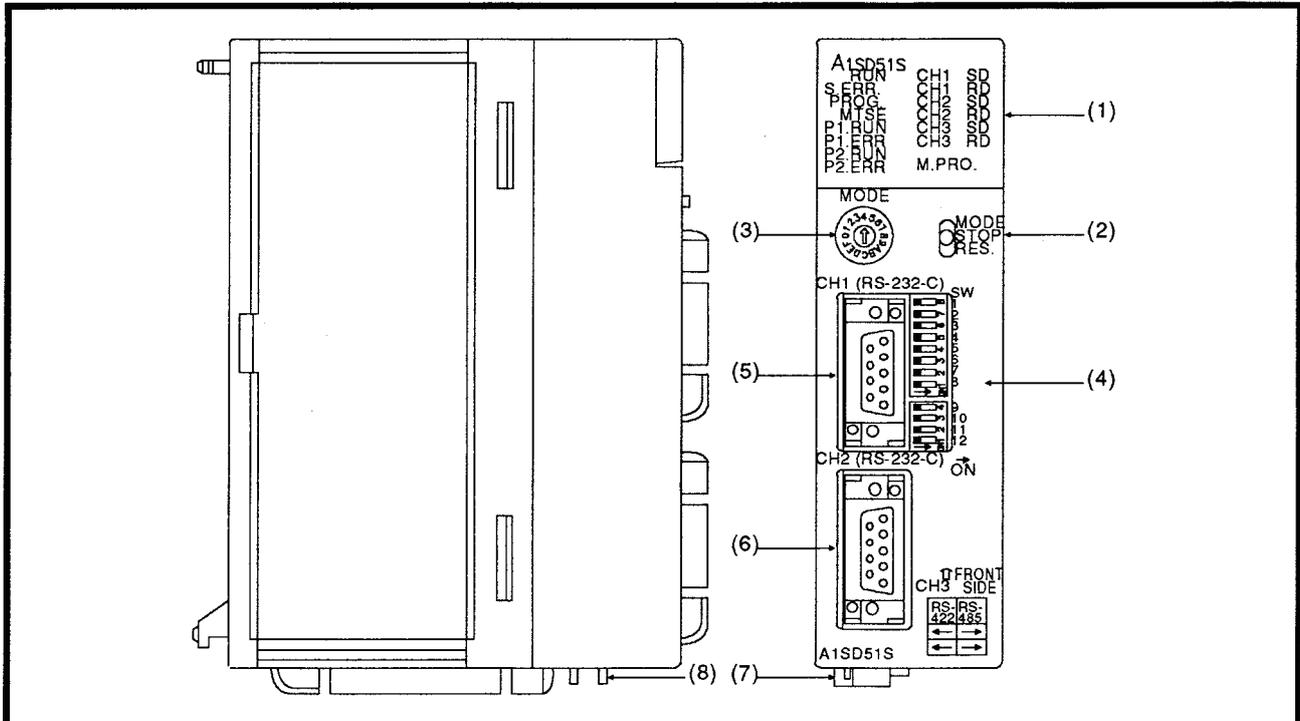
Command/Function	Processing
GETMEM	Reads data from the common memory.
PUTMEM	Stores data in the common memory.

5. NOMENCLATURE AND SETTINGS

This section covers the names and settings of the parts of the A1SD51S.

5.1 Nomenclature

The figure below indicates the names of the parts of the A1SD51S.



No.	Name	Function	Reference Section
(1)	Indicator LEDs	Display the operation state, ongoing communications, errors, etc. The meaning of the ON/OFF state depends on the LED.	5.2
(2)	RUN switch	Used to execute/stop tasks and to reset hardware. <ul style="list-style-type: none"> • RUNTo execute tasks (valid only the Run and Multitask debug modes). • STOPTo stop tasks (valid only the Run and Multitask debug modes). • RESETTo reset hardware. 	—
(3)	Mode setting switch 1	Used to set the operation mode of the A1SD51S.	5.3.1
(4)	Mode setting switch 2	Used to make settings for the console, debug port, memory protect, etc.	5.3.2 to 5.3.7
(5)	RS-232 interface (CH.1)	Used to connect a console, debugger or external device.	—
(6)	RS-232 interface (CH.2)	Used to connect a debugger or external device.	—
(7)	RS-422/485 interface (CH.3)	Used to connect a console, debugger or external device.	—
(8)	Terminal resistor selector switch	Used to select the RS-422/485 terminal resistor. <ul style="list-style-type: none"> • RS-422 330 Ω • RS-485 110 Ω 	—

5. NOMENCLATURE AND SETTINGS

MELSEC-A

5.2 Indications of Indicator LEDs

The indications of the indicator LEDs are explained below:

LED Layout	LED Name	Indication of LED	Initial State of LED
RUN ○ ○ CH1 SD S.ERR ○ ○ CH1 RD PROG ○ ○ CH2 SD MTSE ○ ○ CH2 RD P1.RUN ○ ○ CH3 SD P1.ERR ○ ○ CH3 RD P2.RUN ○ ○ P2.ERR ○ ○ M.PRO	RUN	Normal operation During normal operation: ON When a fault occurs: OFF	ON
	S.ERR	System error During normal operation: OFF When a fault occurs: ON	OFF
	PROG	Programming mode In programming mode: ON	*1
	MTSE	Multitask setting error When a setting error occurs: ON	OFF
	P1.RUN	Task 1 execution When task 1 is executed: ON	*2
	P1.ERR	Task 1 error When a task 1 error occurs: ON	OFF
	P2.RUN	Task 2 execution When task 2 is executed: ON	*2
	P2.ERR	Task 2 error When a task 2 error occurs: ON	OFF
	CH1 SD	CH.1 send status During data transmission: Flashes	OFF
	CH1 RD	CH.1 receive status During data reception: Flashes	OFF
	CH2 SD	CH.2 send status During data transmission: Flashes	OFF
	CH2 RD	CH.2 receive status During data reception: Flashes	OFF
	CH3 SD	CH.3 send status During data transmission: Flashes	OFF
	CH3 RD	CH.3 receive status During data reception: Flashes	OFF
	M.PRO.	Memory protect status When memory is protected: ON	*3

*1 The PROG LED comes ON when mode setting switch 1 is set to the programming mode. It is OFF in other modes.

*2 The P2.RUN LED comes ON if the task start condition is START and the RUN switch is set to RUN when the power is turned ON or the A1SD51S is reset. It is OFF under other conditions.

*3 The M.PRO. LED comes ON or goes OFF according to the setting of mode setting switch 2 SW10.

5.3 Switch Settings

This section describes the switch settings required to operate the A1SD51S.

POINT

After changing the settings of switches, turn the power to the PC CPU OFF and then back ON or reset the A1SD51S.

5.3.1 Setting the A1SD51S operation mode (BASIC program execution, multitask debugging or programming)

To use the A1SD51S, the operation mode must be set.

(1) Operation mode

There are three operation modes.

(a) Run mode

The execution mode is used to read execution programs from the EEP-ROM into the A1SD51S program area and control them. (Refer to Section 3.10.)

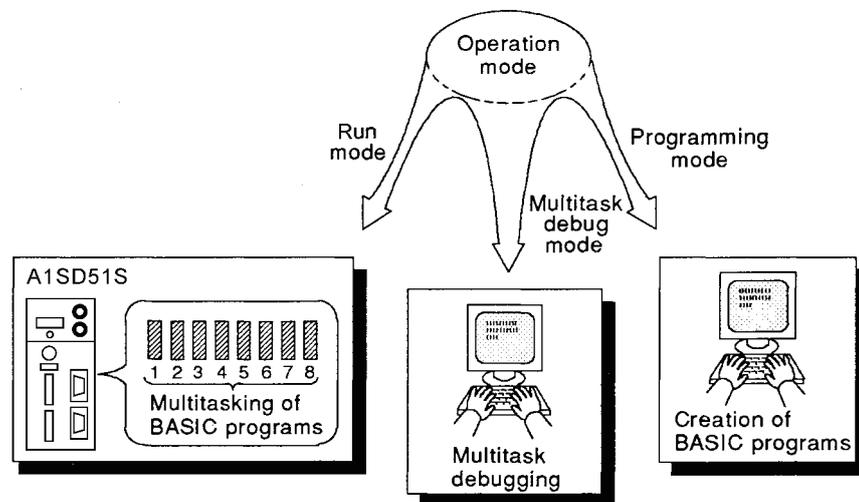
(b) Multitask debug mode

The multitask debug mode is used to debug two or more BASIC programs created in the programming mode, while they are being executed, by using a debugger. (Refer to AD51H-BASIC Programming Manual (Debug and Compile).)

(c) Programming mode

The programming mode is to create a BASIC program for each task using the console. (Refer to AD51H-BASIC Programming Manual (Debug and Compile).)

This mode also has a function for storing BASIC programs in a EEP-ROM and to make multitask settings.

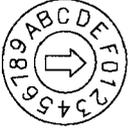


(2) Setting the operation mode

Set mode setting switch 1 to a number between No. 1 and No. 4 to set the operation mode.

Table 5.1 shows the relationship between the setting of operation mode setting switch 1 and the operation mode.

Table 5.1 Setting of Mode Setting Switch 1

Mode Setting Switch 1	Operation Mode	Number to be Set
	Run mode	0 or 1
	Multitask debug mode	2 or 3
	Programming mode	4
	Unusable	5 to F

5.3.2 Settings when a console and a debugger are used

This section describes how to set an interface (RS-232 or RS-422) to which a console for creating BASIC programs or a debugger for multitask debugging is to be connected.

(1) Console and debugger

The following peripheral devices and general-purpose consoles can be used as consoles and debuggers, respectively.

(a) Console

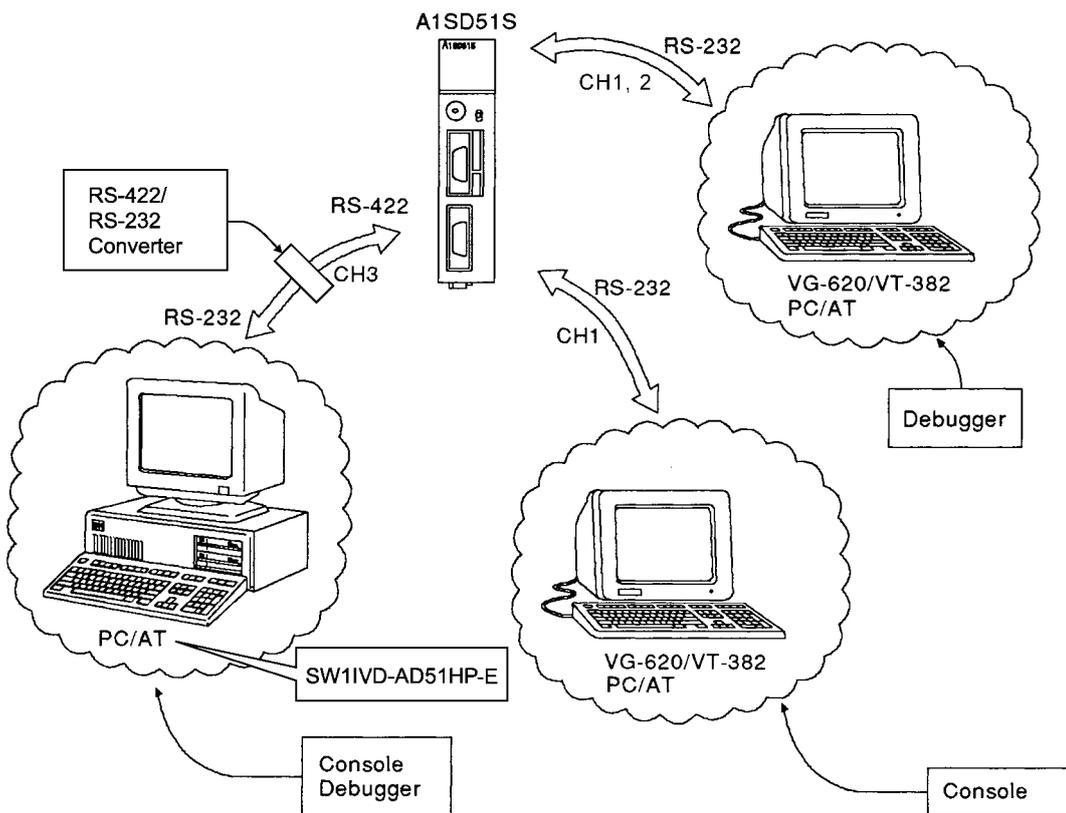
A console can be operated in the programming mode (position of mode setting switch 1: "4").

It is available whether the RUN switch is set to "RUN" or "STOP".

- 1) PC/AT Connect to the RS-422/485 interface.
- 2) VT-382/VG-620 ... Connect to the RS-232 (CH.1) interface.

(b) Debugger

- 1) PC/AT Connect to the RS-422/485 interface.
- 2) VT-382/VG-620 ... Connect to the RS-232 (CH.1/CH.2) interface.



(2) Console and debugger settings

Use ON/OFF setting combinations of SW1 to SW5 of mode setting switch 2 to set the devices (PC/AT, VG-620 or VT-382) to be used as the console and the debugger.

When the console and the debugger are to be connected to CH.1 and CH.2 of the RS-232 interface, these devices must be the same model (PC/AT, VT-620 or VT-382).

This means that a VG-620, a VT-382 and a PC/AT cannot be used simultaneously as the console and the debugger.

Table 5.2 shows the setting combinations of mode setting switch 2 for the console and the debugger.

Table 5.2 Setting Combinations of Mode Setting Switch 2 for Console and Debugger

Mode Setting Switch	Console	Debugger					
		None	RS-422/485	RS-232 (CH1)		RS-232 (CH2)	
			PC/AT	VT-382 PC/AT	VG-620	VT-382 PC/AT	VG-620
	PC/AT (connected to CH.3)		—				
	VT-382 PC/AT (connected to CH.1)			—	—		—
	VG-620 (connected to CH.1)			—	—	—	
	None						

REMARK

The combinations indicated by "—" in Table 5.2 cannot be set.

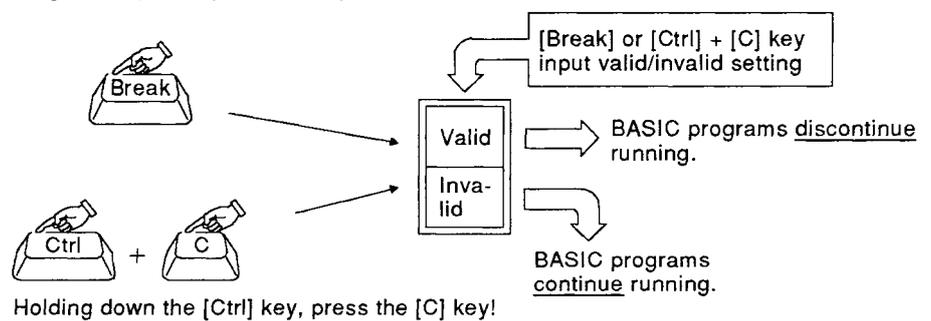
5.3.3 Settings for stopping and continuing BASIC programs with the [Break] or [Ctrl] + [C] keys

This section describes how to make the [Break]^{*2} or [Ctrl] + [C] keys on the console^{*1} valid or invalid during multitasking in the run mode.

(1) Program operation when the [Break] or [Ctrl] + [C] keys are pressed

(a) As soon as the [Break] or [Ctrl] + [C] keys on the console are pressed when they are valid, multitasking stops, and the operation mode is switched to the system mode.
To forcibly stop multitasking from the console, set these keys to "valid".

(b) When the [Break] or [Ctrl] + [C] keys on the console are invalid, multitasking continues even if they are pressed.
The AD51H receives [Break] or [Ctrl] + [C] key input as "key code (03H)".
Set these keys to the invalid state to prevent suspension of multitasking if they are pressed by mistake.



(2) [Break] or [Ctrl] + [C] key input valid/invalid setting

[Break] or [Ctrl] + [C] key input can be made valid or invalid by setting SW6 of mode setting switch 2 on the face of the A1SD51S to ON or OFF. Table 5.3 shows the settings of mode setting switch 2 and their functions.

Table 5.3 [Break] or [Ctrl] + [C] Key Input Valid/Invalid Setting

[Break] or [Ctrl] + [C] Key Input Setting	SW6 Setting
Invalid	
Valid	

REMARK

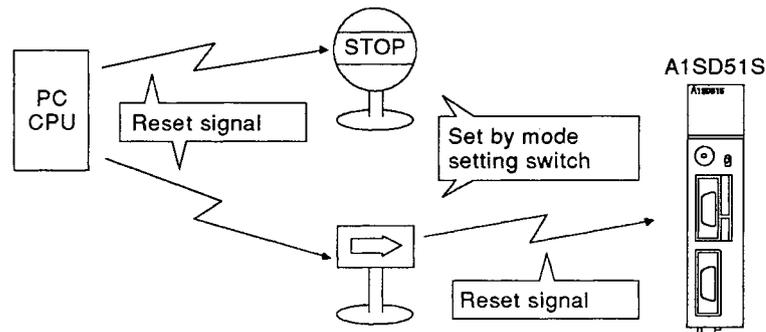
- (1) *1: The device set by the use of SW1 to SW5 of mode setting switch 2 functions as the console.
- (2) *2: Discontinuation of multitasking with the [Break] or [Ctrl]+[C] key is available only with the PC/AT.
- (3) Setting the [Break] or [Ctrl] + [C] keys invalid can prevent multitasking from discontinuing due to wrong operation of these keys.

5.3.4 A1SD51S operation settings when the PC CPU is reset

When the PC CPU is reset, the I/O and special-function modules loaded on the main and extension base units initialize their settings and restart in the initial state (same operation as that when the power is turned ON).

It is possible to make a setting so that, when the PC CPU is reset, the A1SD51S makes the reset signal from the PC CPU valid, thus causing operation to restart from the beginning, or makes the signal invalid, allowing the ongoing operation to continue.

(The mode setting switch is factory-set to "reset signal invalid".)



(1) Operation when the PC CPU is reset

(a) When the reset signal from the PC CPU is valid, the A1SD51S also initializes its settings and restarts in the initial state (same operation as that when the power is turned on).

(For details of A1SD51S operation, refer to Section 3.9.)

(b) When the reset signal from the PC CPU is invalid, the A1SD51S continues the ongoing operation. However, since the PC CPU is reset, the following points should be noted.

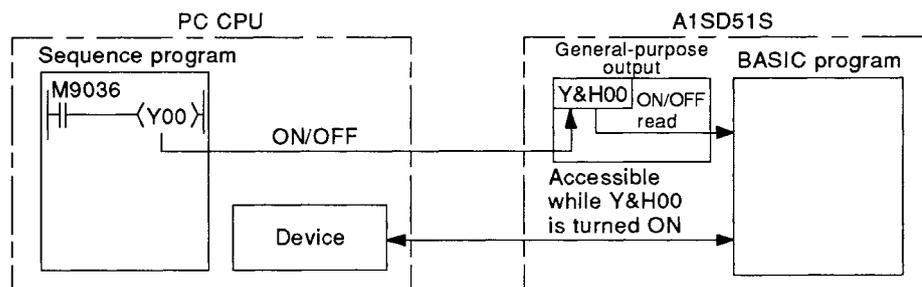
1) All output signals whose ON/OFF state is controlled by the PC CPU turn OFF.

Since the start signal for a designated task, which is set to start by an interrupt from the PC CPU, is turned ON again after the power to the PC CPU is turned ON, the task will be executed.

2) During the time the PC CPU is being reset (about five seconds when reset using the RUN switch), it cannot be accessed by the A1SD51S.

If the PC CPU is reset during access from the A1SD51S involving the PCRD or PCWT command in a BASIC program, the command causes a "time-over error".

The PC CPU is accessible from the A1SD51S only when an output (Y) signal from the PC CPU is used as the interlock signal and the output device assigned to the signal is turned ON.



POINT

If the A1SD51S performs resetting while a file stored on a FD or HD is open, the file may be destroyed.
 When the A1SD51S manages data files on a FD or HD, set the reset signal from the PC CPU to "invalid".

(2) Setting the reset signal from the PC CPU to valid or invalid

The reset signal from the PC CPU can be made valid or invalid by setting SW7 of mode setting switch 2 to ON or OFF.

Table 5.4 shows the relationship between the setting of mode setting switch 2 and the valid or invalid state of the reset signal from the PC CPU.

Table 5.4 Relationship between Setting of Mode Setting Switch 2 and Valid/Invalid State of Reset Signal from PC CPU

	Reset Signal Invalid	Reset Signal Valid
Mode setting switch 2	<p style="text-align: center;">→ ON</p>	<p style="text-align: center;">→ ON</p>

5.3.5 Setting the time during which access from the PC CPU is possible after the A1SD51S is reset

- (1) Operation when the A1SD51S is reset
 - (a) After resetting using the RUN switch, the A1SD51S cannot communicate with the PC CPU until it has initialized its settings and restarted in the initial state.
If an attempt is made, during A1SD51S resetting, to read data from or write data to the buffer memory with the FROM or TO instruction from the PC CPU, a "SP UNIT DOWN" error will occur in the PC CPU, and the sequence program will stop.
 - (b) While the A1SD51S is undergoing reset processing, the "multitask execution start signal (X0B)" remains OFF.
The time during which access from the PC CPU is possible after the A1SD51S is reset must be set so the PC CPU can reliably detect the "multitask execution start signal (X0B)" remaining OFF when the A1SD51S is reset.
Set a time longer than the PC CPU scan time.
- (2) Setting the access time
 - (a) The access time is factory-set to 200 msec.
If the PC CPU scan time is longer than 200 msec, set the access time to 2000 msec.
 - (b) The access time can be set by switching SW8 of mode setting switch 2 ON or OFF.
Table 5.5 shows the relationship between the setting of this switch and the access time.

Table 5.5 Access Time Setting

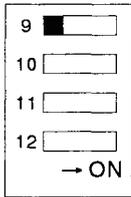
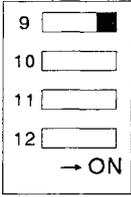
	Access Time	
	200 msec	2000 msec
Mode setting switch 2	<p style="text-align: center;">→ ON</p>	<p style="text-align: center;">→ ON</p>

5.3.6 Setting backup area clear

This section describes how to enable/disable clearance of the backup area (common memory, extension relay EM, extension register ED). If backup area clear is enabled, the common memory and the extension register will be all cleared and the extension relays will be turned OFF when the power to the A1SD51S is ON or the module is reset.

- (a) Use SW9 of mode setting switch 2 to enable or disable backup area clear.
The backup areas will be cleared if the switch is set to ON.
- (b) Table 5.6 shows the relationship between the setting of SW9 mode setting switch 2 and the backup area clear state.

Table 5.6 Backup Area Clear Setting

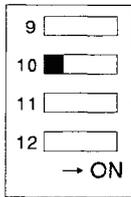
	Clear Disabled	Clear Enabled
Mode setting switch 2		

5.3.7 Setting EEPROM write protect

Write protect can be set for execution programs stored EEPROM.

- (a) Use SW10 of mode setting switch 2 to set the write protect status.
Write protection is ON when the switch is set to ON.
- (b) When write protection is ON, the entire area of the EEPROM is write-protected.
- (c) Table 5.7 shows the relationship between the setting of SW10 of mode setting switch 2 and the write protect status.

Table 5.7 Write Protect Setting

	Write Protection OFF	Write Protection ON
Mode setting switch 2		

5.4 Loading and Installation

This section describes the precautions for handling and installation conditions which ensure high system reliability and correct operation of the module functions.

5.4.1 Precautions for handling

The precautions for handling the A1SD51S are listed below.

- (1) Since the module case is made of resin, do not drop it or subject it to shock.
- (2) Do not remove the PC board from the module case. This may cause a fault.
- (3) Take precautions to prevent wire scraps or foreign material from falling into the top of the module.
- (4) Tighten the module mounting screws (usually unnecessary) with a torque in the following range:

Screw	Tightening Torque Range
Module mounting screw (usually unnecessary) (M4 x 0.7 screw)	78 to 118 N·cm

- (5) Load the module to the base with its hooks completely engaged with the base.
To remove the module, first disengage the hooks from the base, then pull the module toward you.
(For details, refer to the user's manual for a small building block type CPU.)

5.4.2 Installation conditions

Do not install the A1SD51S in the following locations.

- (1) Locations where the ambient temperature is outside the range 0°C to 55°C.
- (2) Locations where the ambient humidity is outside the range 10% RH to 90% RH.
- (3) Locations where dewing occurs due to sudden temperature changes.
- (4) Locations exposed to corrosive or combustible gases.
- (5) Locations exposed to large amounts of highly conductive dust, iron powder, oil mist, salt or organic solvents.
- (6) Locations where the module is exposed to direct sunlight.
- (7) Locations where a strong electric or magnetic field is generated.
- (8) Locations where the module will be subject to direct vibration or impact.

5.5 Precautions for Wiring

External wiring not susceptible to noise is essential to enable the A1SD51S to achieve its full performance with high reliability.

The precautions for the A1SD51S external wiring are as follows.

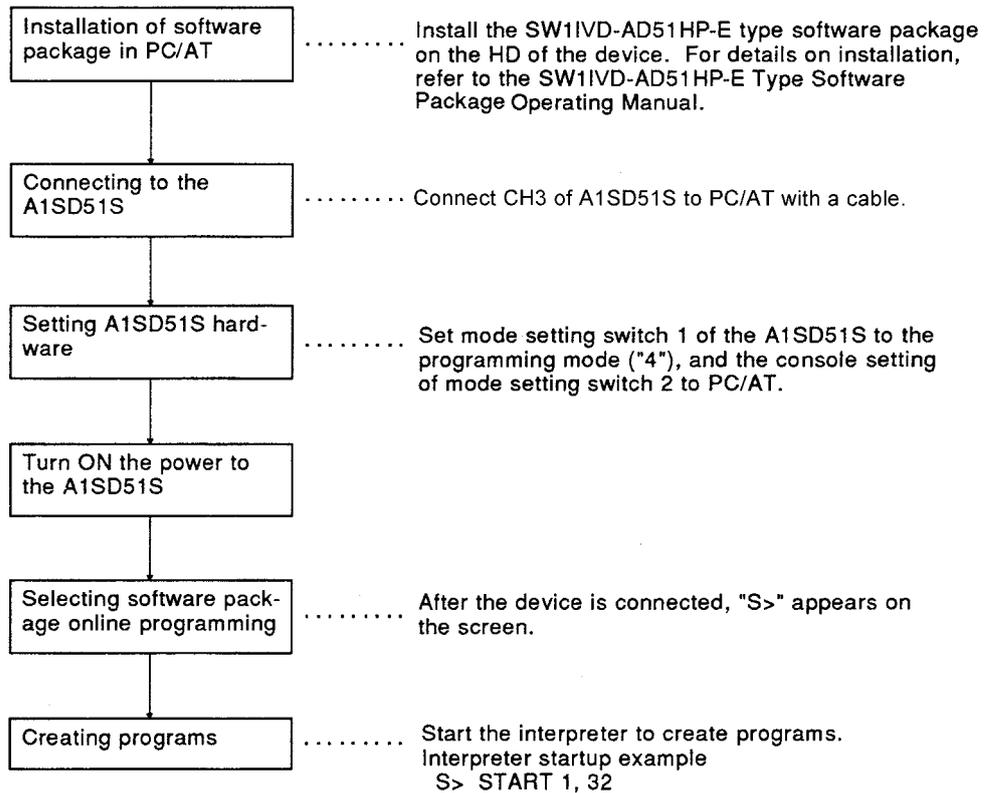
- (1) Do not lay the external wiring near, or bind it with, the main circuit cables, high-tension wires or load carrying wires connected to devices other than the module.
Otherwise, it may be affected by noise or surge induction.
Keep the external wiring at least 100 mm (3.93 inch) away from the above wires.
- (2) No wire is connected to the FG terminal of the power supply unit and that of the A1SD51S RS-232 and RS-422 interfaces.
Ground the shielded wire or cable to an external device.

6. A1SD51S STARTUP PROCEDURE

This chapter describes the steps from the connection of a console to the A1SD51S to starting up the A1SD51S.

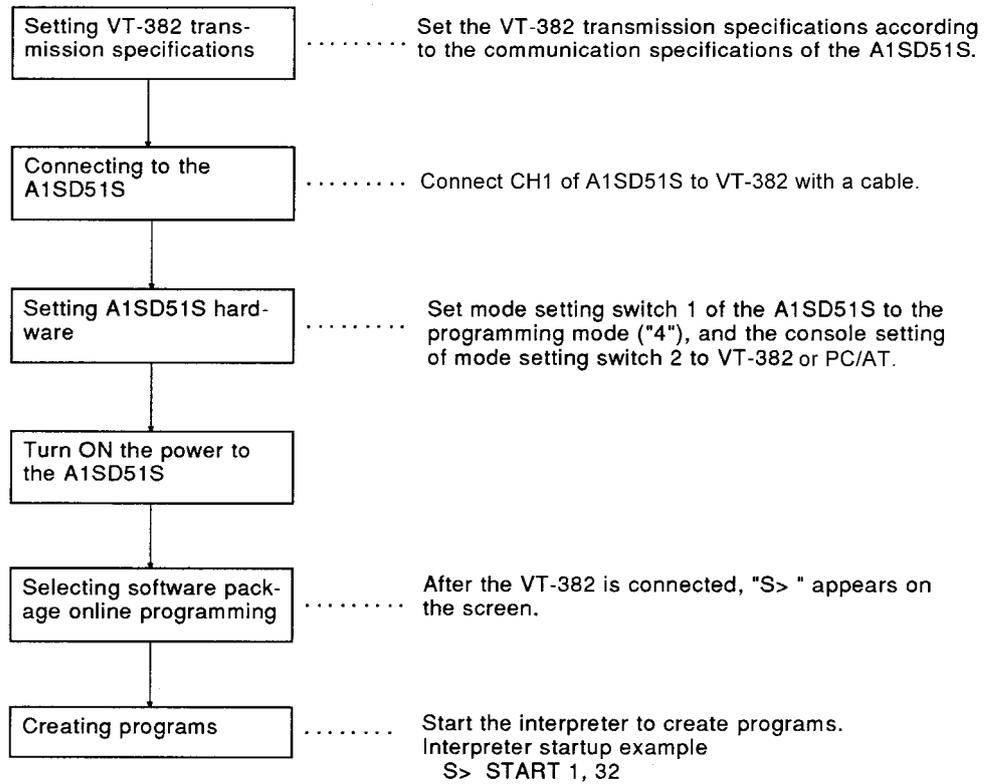
6.1 Startup when a PC/AT is Used as the Console

The following flowchart shows the steps for starting a PC/AT when it is used as the console.



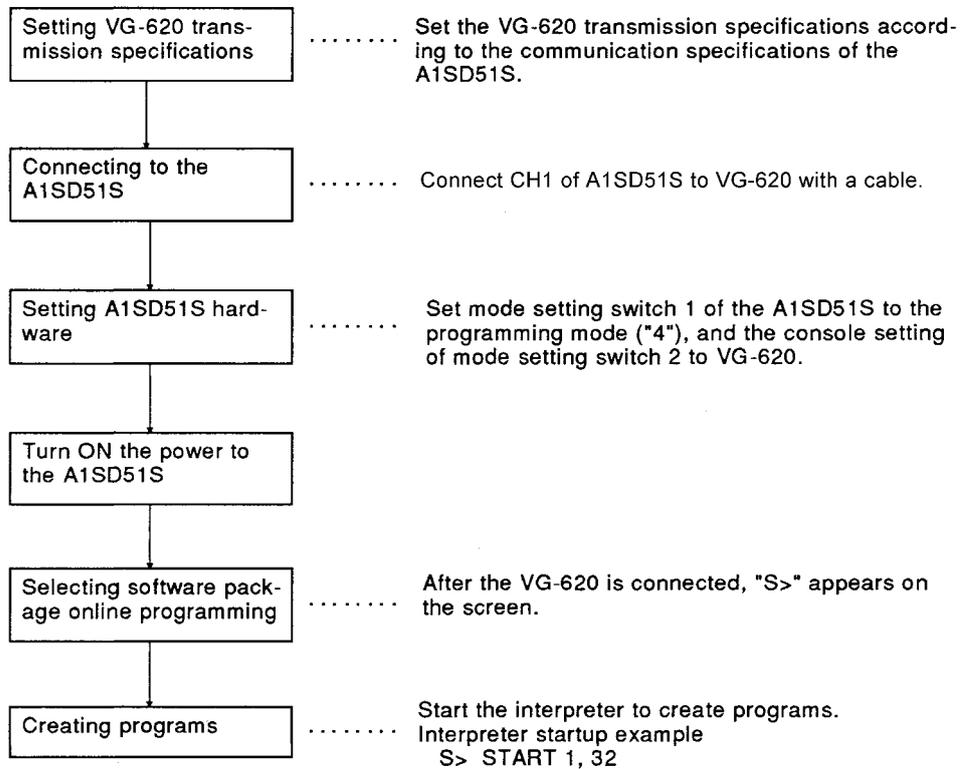
6.2 Startup when a VT-382 is Used as the Console

The following flowchart shows the steps for starting a VT-382 when it is used as the console.



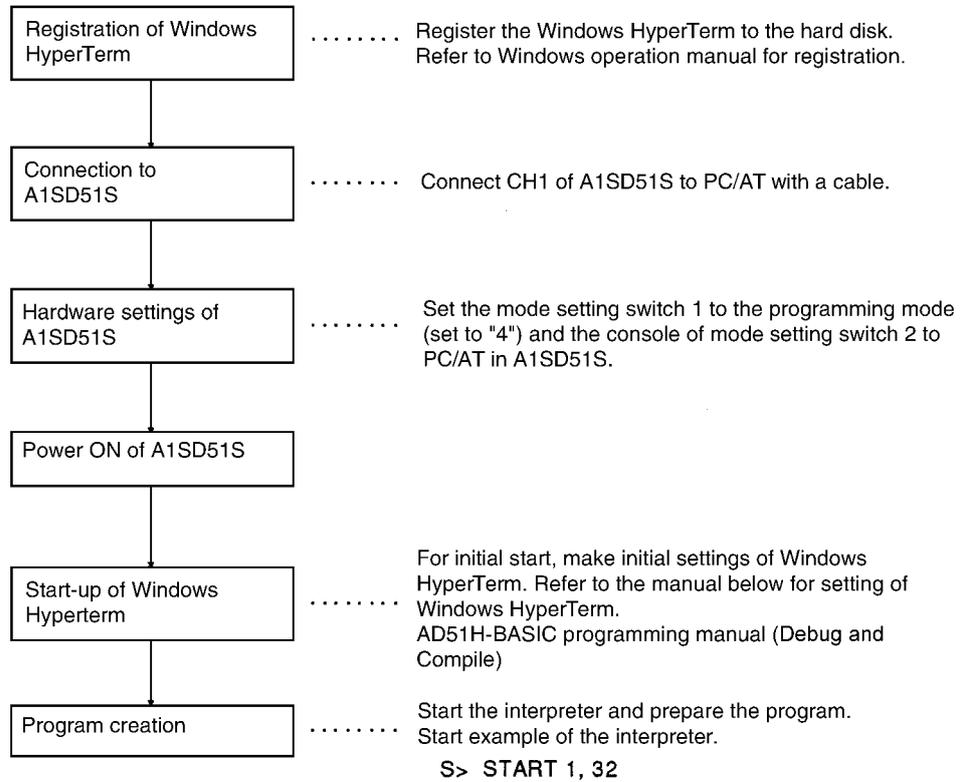
6.3 Startup when a VG-620 is used as the Console

The following flowchart shows the steps for starting a VG-620 when it is used as the console:



6.4 Startup when a PC/AT is Used as the Console (with general-purpose editor)

The procedures to start PC/AT using the general-purpose editor as console are shown below:



APPENDICES

APPENDIX 1 FUNCTIONAL COMPARISON BETWEEN A1SD51S, AD51H-S3, AD51H AND AD51 (S3)

Table 1.1 compares functions between the A1SD51S, the AD51H-S3, the AD51H and the AD51 (S3).

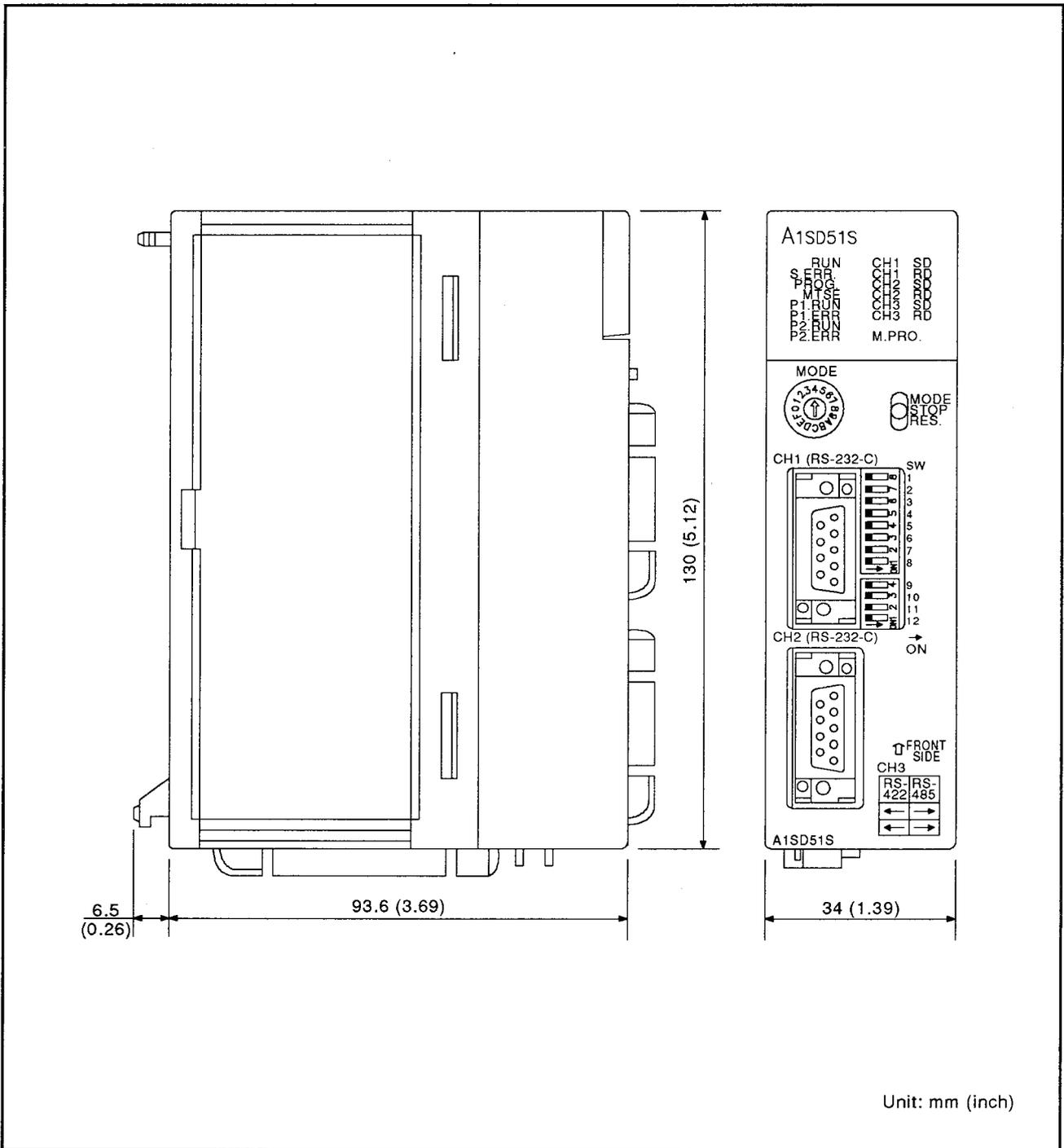
Table 1.1 Functional Comparison between A1SD51S, AD51H-3, AD51H and AD51 (S3)

Item		A1SD51S	AD51H-S3	AD51H	AD51(S3)
CPU element		80C186(15MGz)			HD64180(8MHz)
Operation element		Not equipped			Equipped
Programming language		AD51H-BASIC (interpreter, compiler)		AD51H-BASIC (interpreter)	GPC-BASIC (interpreter)
Number of tasks		2	8		8
Task start condition		<ul style="list-style-type: none"> • When the power is turned ON. • When an interrupt is issued from the PC CPU (but impossible during compilation). • When another tasks issues a start request. 			<ul style="list-style-type: none"> • When the power is turned ON. • When an interrupt is given from the PC CPU (for 1 task) • When a real time interrupt occurs.
Internal memory	Program	64 Kbytes (16/32/48/64 Kbytes selective for 1 task)	384kbytes (16/32/48/64 Kbytes selective for 1 task)		64 Kbytes + 48 Kbytes ↓ For storage to ROM
	Common memory	8 Kbytes			2 Kbytes
	Buffer memory	6 Kbytes			6 Kbytes
	EM	1024 points			————
	ED	1024 points			————
General-purpose I/O		Input: 27 points Output: 23 points (2 points: For task startup from PC CPU)	Input: 27 points Output: 17 points (9 points: For task startup from PC CPU)		Input: 26 points Output: 17 points
Memory protect		Available (EEP-ROM can be write-protect.)	Available (Memory card can be partly protected.)		Available
Interface	RS-232	2 channel			2 channels
	RS-422	1 channel			2 channels
	Parallel	————	1 channel		————
	Memory card	————	2 slots		————
Clock function		Not equipped			Equipped
Power interruption backup		Equipped	Not equipped (Power interruption backup available for memory card)		Equipped

Table 1.1 Functional Comparison between A1SD51S, A1H-3, AD51H and AD51 (S3) (continued)

Item	A1SD51S	AD51H-S3	AD51H	AD51(S3)
Storage to ROM	—	Possible (64KROM-W/128KROM-W/256KROM-W)		Possible (8KROM/16KROM)
Console	<ul style="list-style-type: none"> • PC/AT • VG-620 • VT-382 			<ul style="list-style-type: none"> • A6GPP • A6PHP • VG-620 • VT-220
Programming method	<ul style="list-style-type: none"> • Online programming • Offline programming (when PC/AT is used) 			<ul style="list-style-type: none"> • Online programming
Multitask debugging	Possible (when debugger is used)			Impossible
Number of occupied I/O points	32 points	48 points (16 empty points + 32 special-purpose points)		48 points (16 empty points + 32 special-purpose points)
Number of occupied slots	1 slot	2 slots		2 slots

APPENDIX 2 OUTLINE DRAWING



WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place.

Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. Product application

- (1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
- (2) The Mitsubishi programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable logic controller applications.

In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications.

However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

Microsoft, Windows, Windows NT are registered trademarks of Microsoft Corporation in the United States and other countries.

Adobe and Acrobat are registered trademarks of Adobe Systems Incorporation.

Pentium and Celeron are trademarks of Intel Corporation in the United States and other countries.

Ethernet is a trademark of Xerox. Co., Ltd in the United States.

Other company names and product names used in this document are trademarks or registered trademarks of respective owners.

Type A1SD51S Intelligent Communication Module

User's Manual

MODEL	A1SD51S-U-E
MODEL CODE	13JE90
IB(NA)-66551-D(0610)MEE	



HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.