Q Corresponding Intelligent
Communication Module

User's Manual

MITSUBISHI

Q series

Mitsubishi
Programmable Controller

MELSEC-Q

QD51
QD51-R24

• 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 CPU module user's manual.
In this manual, the safety precautions are ranked as "DANGER" and "CAUTION".

| ⟨!⟩ DANGER | Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury. |
| ⚠ CAUTION | 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.

[Disposal Precautions]

⟨!⟩ 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 programmable controller 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 before exercising other control (program change, operating status change (status control)) on the running programmable controller, read the manual carefully and fully confirm safety. Especially for the above control on the remote programmable controller from, an immediate action may not be taken for programmable controller trouble due to a data communication fault. In addition to configuring up the interlock circuit in the sequence program, corrective and other actions to be taken as a system for the occurrence of a data communication fault should be predetermined between and programmable controller CPU.
- Do not write data to the "System Area" in the intelligent function module's buffer memory. Also, do not turn ON the "Use Prohibited" output signals among the output signals from the programmable controller CPU to the intelligent function module.
  If data are written to the "System Area" or output in response to "Use Prohibited" signals, there is danger that the programmable controller system will malfunction.

[Disposal Precautions]

⚠ 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 away from each other.
  Not doing so could result in noise that would cause erroneous operation.
- If a BASIC program is registered in the flash ROM in the module, do not power OFF or reset the programmable controller CPU at the station where the module is installed during registration.
  If it is performed during registration, the data contents in the flash ROM will be erratic and it will be necessary to reset the setting values, etc. in the buffer memory and register them again in the flash ROM. It could also cause the module to break down or malfunction.

[Installation Precautions]

⚠ CAUTION

- Use the programmable controller in an environment that meets the general specifications contained in the user's manual of the CPU module to use.
  Using this programmable controller in an environment outside the range of the general specifications could result in electric shock, fire, erroneous operation, and damage to or deterioration of the product.
- While pressing the installation lever located at the bottom of module, insert the module fixing tab into the fixing hole in the base unit until it stops. Then, securely mount the module with the fixing hole as a supporting point.
  Incorrect loading of the module can cause a malfunction, failure or drop.
  When using the programmable controller in the environment of much vibration, tighten the module with a screw.
- Tighten the screw in the specified torque range.
  Undertightening can cause a drop, short circuit or malfunction.
  Overtightening can cause a drop, short circuit or malfunction due to damage to the screw or module.
- Completely turn off the externally supplied power used in the system before mounting or removing the module.
  Not doing so could result in damage to the product.
- 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

- When turning on the power supply or operating the module after installation or wiring work, be sure that the module's terminal covers are correctly attached. Not attaching the terminal cover could result in electric shock.
- External connections shall be crimped or pressure welded with the specified tools, or correctly soldered.
  Imperfect connections could result in short circuit, fires, or erroneous operation.
- 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 with the specified torque.
  If the terminal screws are loose, it could result in short circuits, fire, or erroneous operation.
  Tightening the terminal screws too far may cause damages to the screws and/or the module, resulting in fallout, 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.
- Be sure there are no foreign substances such as sawdust or wiring debris inside the module.
  Such debris could cause fires, damage, or erroneous operation.
- The module has an ingress prevention label on its top to prevent foreign matter, such as wire offcuts, from entering the module during wiring.
  Do not peel this label during wiring.
  Before starting system operation, be sure to peel this label because of heat dissipation.

[Startup and Maintenance precautions]

⚠ **CAUTION**

- Do not disassemble or modify the modules.
  Doing so could cause trouble, erroneous operation, injury, or fire.
- Completely turn off the externally supplied power used in the system before mounting or removing the module. Not doing so could result in damage to the product.
- Do not install/remove the module to/from the base unit, or the terminal block to/from the module more than 50 times after the first use of the product. (IEC 61131-2 compliant)
  Failure to do so may cause malfunction.
- Do not touch the terminals while power is on.
  Doing so could cause shock or erroneous operation.
- Switch off all phases of the externally supplied power used in the system when cleaning the module or retightening the terminal or module fixing screws.
  Not doing so could result in electric shock.
  Undertightening of terminal screws can cause a short circuit or malfunction.
  Overtightening of screws can cause damages to the screws and/or the module, resulting in fallout, short circuits, or malfunction.
- Before touching the module, always touch grounded metal, etc. to discharge static electricity from human body, etc.
  Not doing so can cause the module to fail or malfunction.

[Operating Precautions]

⚠ **CAUTION**

- When performing control (in particular, changing data, changing a program or changing the operation status (status control)) of the programmable controller 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

Print Date	∗ Manual Number	Revision
Apr., 2000	SH(NA)-080089-A	First printing
Sep., 2000	SH(NA)-080089-B	Addition Section 1.3, Section 2.3, 2.3.1, 2.3.2, 2.4
Dec., 2003	SH(NA)-080089-C	Correction Section 2.1, 6.2
Oct., 2004	SH(NA)-080089-D	Correction SAFETY PRECAUTIONS, About Manuals, Conformation to the EMC Directive and Low Voltage Instruction, About the Generic Terms and Abbreviations, Chapter 1, Section 2.1, 2.3, 2.4, 2.5, Section 3.1, 3.2.1, 3.3.2(2), 3.8.1, Section5.1, 5.5.1 Term change <table><tr><td>Before change</td><td>After change</td></tr><tr><td>GPPW</td><td>GX Developer</td></tr><tr><td>MELSECNET/10H</td><td>MELSECNET/H</td></tr></table>
Feb., 2006	SH(NA)-080089-E	Correction SAFETY PRECAUTIONS, About Manuals, Conformation to the EMC Directive and Low Voltage Instruction, How to read this manual, and Configuration, Section 1.1 to 1.3, Section 2.1, 2.5, Section 3.1, 3.2.1, 3.3.1, 3.3.3, 3.4, 3.5, 3.5.1 to 3.5.7, 3.6 to 3.8, 3.8.1, 3.8.2, 3.9, 3.10, 3.10.1 to 3.10.3, 3.11, 3.11.1 to 3.11.4, 3.12.1, 3.13.1, 3.13.2, Section 4.1, 4.2, 4.2.1, 4.2.2, 4.3, 4.3.2, 4.4.1, 4.4.5, 4.5, 4.5.3, 4.5.4, 4.8.2, Section 5.2 to 5.5, 5.5.2, 5.6.1, 5.6.2, Section 6.1, 6.2, Appendix-1, Appendix-2, Appendix-2.1, Appendix-3 Deletion Appendix-2.2 (incorporated into Appendix-2.1)
Oct.,2006	SH(NA)-080089-F	Correction SAFETY PRECAUTIONS, Section 2.1
Aug.,2008	SH(NA)-080089-G	Change of a term "PLC" was changed to "programmable controller". Correction SAFETY PRECAUTIONS, Compliance with the EMC and Low Voltage Directives, About The Generic Terms and Abbreviations, Section 1.2, 2.1, 2.3, 2.4, 2.6, 3.1 to 3.3, 5.1, 5.3 to 5.5, 5.6.2, 5.7, 6.2, Appendix-3

INTRODUCTION

Thank you for purchasing the MELSEC-Q series programmable controller.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the Q series programmable controller you have purchased, so as to ensure correct use.

CONTENTS

The following manuals are also related to this product.
In necessary, order them by quoting the details in the tables below.

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)

Compliance with the EMC and Low Voltage Directives

(1) For programmable controller system
To configure a system meeting the requirements of the EMC and Low Voltage Directives when incorporating the Mitsubishi programmable controller (EMC and Low Voltage Directives compliant) into other machinery or equipment, refer to Chapter 9 "EMC AND LOW VOLTAGE DIRECTIVES" of the QCPU User's Manual (Hardware Design, Maintenance and Inspection).
The CE mark, indicating compliance with the EMC and Low Voltage Directives, is printed on the rating plate of the programmable controller.

(2) For the product
No additional measures are necessary for the compliance of this product with the EMC and Low Voltage Directives.

● How to use this manual
   We will explain concerning the QD51 (-R24) for each purpose of use.
   Use this manual for reference for the following contents.

(1) When desiring to know the features, functions and component parts

(a) When desiring to know the features and functions
   • The features of the QD51 (-R24) are described in Chapter 1.
   • The common specifications and functions, etc. of the QD51 (-R24) are described in Chapter 3.

(b) When desiring to know the package contents and system component parts
   • The items included in the package when the QD51 (-R24) is purchased are described in "Product Configuration" before Chapter 1.
   • The user should procure parts and materials not included in the package contents separately.

(2) When desiring to know the processing necessary to get the QD51 (-R24) started

(a) When desiring to know the startup procedure
   • An abbreviated sequence of the procedures up to the point when the QD51 (-R24) is operated is described in section 5.2.

(b) When desiring to know about connections with the console debugger
   • The connection method is described in section 5.4.

(c) When desiring to know about connection with the opposite party's device
   • The connection method for each type of interface is described in section 5.5.

(d) When desiring to know the processing necessary before the QD51 (-R24) is started up
   • Setting of the parameters used in the QD51 (-R24) using GX Developer is explained in section 5.6.

(3) When desiring to know about program application

When desiring to know about application of the program from the A1SD51S or AD51H-S3
 • Program application is described in section 2.1 of the Appendix.

About the Generic Terms and Abbreviations

In this manual, unless specially clarified, explanation of the Intelligent Communications Module using the general names and abbreviated names shown below.

(1) Generic terms and Abbreviations of Affected Modules

In this manual, programmable controller CPUs, etc. related to the Intelligent Communications Module are displayed with the following generic terms and abbreviations. When it is necessary to clarify the relevant model name, the affected module's model name is included.

Generic Term/Abbreviation	Content of Generic terms and Abbreviations
ACPU	AnNCPU, AnACPU, AnUCPU
AnACPU	A2ACPU, A2ACPU-S1, A2ACPUP21/R21, A2ACPUP21/R21-S1, A3ACPU, A3ACPUP21/R21
AnNCPU	A1NCPU, A1NCPUP21/R21, A2NCPU, A2NCPU-S1, A2NCPUP21/R21, A2NCPUP21/R21-S1, A3NCPU, A3NCPUP21/R21
AnUCPU	A2UCPU, A2UCPU-S1, A2USCPU-S1, A2USHCPU-S1, A3UCPU, A4UCPU
AnA/AnU/QnACPU	AnACPU, AnUCPU, QnACPU
AnU/QnACPU	AnUCPU, QnACPU
QD51	Shows only the QD51.
QD51-R24	Shows only the QD51-R24.
QD51(-R24)	Shows both the AD51 and QD51-R24.
QCPU	Q00JCPU, Q00CPU, Q01CPU, Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU, Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU, Q12PRHCPU, Q25PRHCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q13UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q13UDEHCPU, Q26UDEHCPU
QCPU-A	Q02CPU-A, Q02HCPU-A, Q06HCPU-A
QnACPU	Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU, Q4ARCPU
Q/QnACPU	QCPU, QnACPU

(2) Other Generic terms and Abbreviations

In this manual, intelligent communications module data communications devices, etc. are explained using the following generic terms and abbreviations. When it is necessary to indicate clearly what the object of the explanation is, the name / model name of that item is included.

Generic Term/Abbreviation	Content of Generic terms and Abbreviations
Opposite device (external device)	Computer, display, measuring instrument, ID module, bar code reader, adjuster or other intelligent communications module, UC 24, etc. connected to this intelligent communications module for data communications.
Intelligent Function Module	Q series programmable controller module which runs according to commands from the programmable controller CPU. (Equivalent to a special function module of the A series programmable controller) (Examples) • CC-Link Interface Module • A/D, D/A Conversion Module • Ethernet Interface Module • Serial Communications Module
Intelligent Function Module Device	Buffer memory of the intelligent function module for storing data sent to and received from the programmable controller CPU (setting values, monitor values, etc.).
Special Function Module	A/QnA series programmable controller modules operated by commands from the programmable controller CPU. (Equivalent to Q series programmable controller intelligent function modules.) (Examples) • CC-Link Interface Module • A/D, D/A Conversion Module • High Speed Counter Module • Ethernet Interface Module • Computer Link Module, Serial Communications Module
Buffer Memory	Buffer memory of the intelligent function module for storing data sent to and received from the programmable controller CPU (setting values, monitor values, etc.).
I/F	Interface
CH	Channel
GX Developer	Generic product name for SWnD5C-GPPW-E, SWnD5C-GPPW-EA, SWnD5C-GPPW-EV, and SWnD5C-GPPW-EVA. ("n" means version 4 or later.) "-A" and "-V" mean "volume license product" and "version-upgrade product" respectively.
MELSECNET/10	MELSECNET/10 network system.
MELSECNET/H	MELSECNET/H network system.
RS-232 (Interface)	RS-232 compatible interface.
RS-422/485 (Interface)	RS-422 and RS-485 compatible interface.
YCHECK	Abbreviation for Z.YCHECK or ZP.YCHECK.
User's Manual (Hardware) or Hardware Manual	Q Corresponding Intelligent Communications Module User's Manual (Hardware)
Programming Manual (Commands) or Command Manual	AD51H-BASIC Programming Manual (Commands)
Programming Manual (Debug and Compile) or Debug and Compile Manual	AD51H-BASIC Programming Manual (Debug and Compile)

Meanings of Terms and Their Contents

The meanings of terms used in this manual and their contents are shown below.

Term	Contents
Console	This is a device which connects to the QD51 (-R24) and is used to perform programming and multitasking settings. A PC/AT personal computer with the software package installed becomes a console. Depending on the settings, the console becomes the debugger.
Debugger	This device connects to the QD51 (-R24) and performs debugging. A PC/AT personal computer with the software package installed becomes the debugger. The debugger can check variable values and can edit the program while it is being run.
Terminal	This connects to the QD51 (–R24) and is used to display the screen and input characters from the keyboard.
Interpreter	This is BASIC in a format which processes while executing commands one by one.
Compiler	This compiles programs created with the interpreter. It registers the compiled files and executes them. Compared to the interpreter, its execution speed is on the average 2 or 3 times faster.
Multitasking	This executes multiple programs by time slicing, executing them so that it seems that they are being executed simultaneously.
Run Mode	This is the mode used when the console is disconnected and the QD51 (-R24) is being operated independently. It is actually used when this device is being run as a system.
Programming Mode	This mode is used to connect to the console and carry out programming.
Multitask Debug Mode	This mode changes the contents of variables and traces them while executing a program, and carries out tracing.

Product Configuration

The QD51 (-R24) product configuration is as shown below.

Model Name	Product Name	Quantity
QD51	Model QD51 Intelligent Communications Module	1
QD51-R24	Model QD51-R24 Intelligent Communications Module	1
	Terminal Resistor 330 Ω, 1/4 W (For RS-422 communications)	2
	Terminal Resistor 110 Ω, 1/2 W (For RS-485 communications)	2

1 OVERVIEW

1

This manual explains the system configuration, performance specifications and functions of the QD51 / QD51-R24 Intelligent Communications Module.
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 QD51 (-R24) has the following functions.

(1) BASIC Program-Based Functions
- Sub-CPU Function
   Complex numerical calculations and functional calculations can be made with a BASIC program.
- Monitor Display Function
   It can display production conditions, the operating status, details of a breakdown, etc.
- Key Input Function
   The production schedule, production volume, operation, setting data, etc. can be input.
- Printer Function
   Production plans, performances, daily reports, breakdown details, planning data, inspection results and test performance, etc. can be printed out.
- Data Input Function
   Data can be input from a bar code reader or magnetic card reader, etc.
- External Device Connection Function
   A computer, etc. can be connected to the RS-232, RS-422/485 interfaces and data can be sent and received through a BASIC program.
- Clock Function
   Clock data of the programmable controller CPU can be read and written.

(2) Offline Programming Function
   BASIC programs can be created, corrected, stored to; user FD and printed out independently by peripheral devices.

(3) Online Programming Function
   BASIC programs can be created, run, and corrected from the console.

(4) Multitasking Debugging Function
   BASIC programs can be debugged while they are being run.

1

1.1 Operation Overview for QD51 (-R24)

The QD51 (-R24) operates by BASIC programs created by the user.
The BASIC version that can be used by the QD51 (-R24) is AD51H-BASIC, which is capable of multitasking processing of up to 2 tasks.

Besides the various calculations that can be carried out by a BASIC program created by the user,
• Data can be sent to and received from an external device.
• Data can be sent to and received from the programmable controller CPU.
The operation in the QD51 (-R24) when data are sent to or received from an external device or the programmable controller CPU is as shown below.
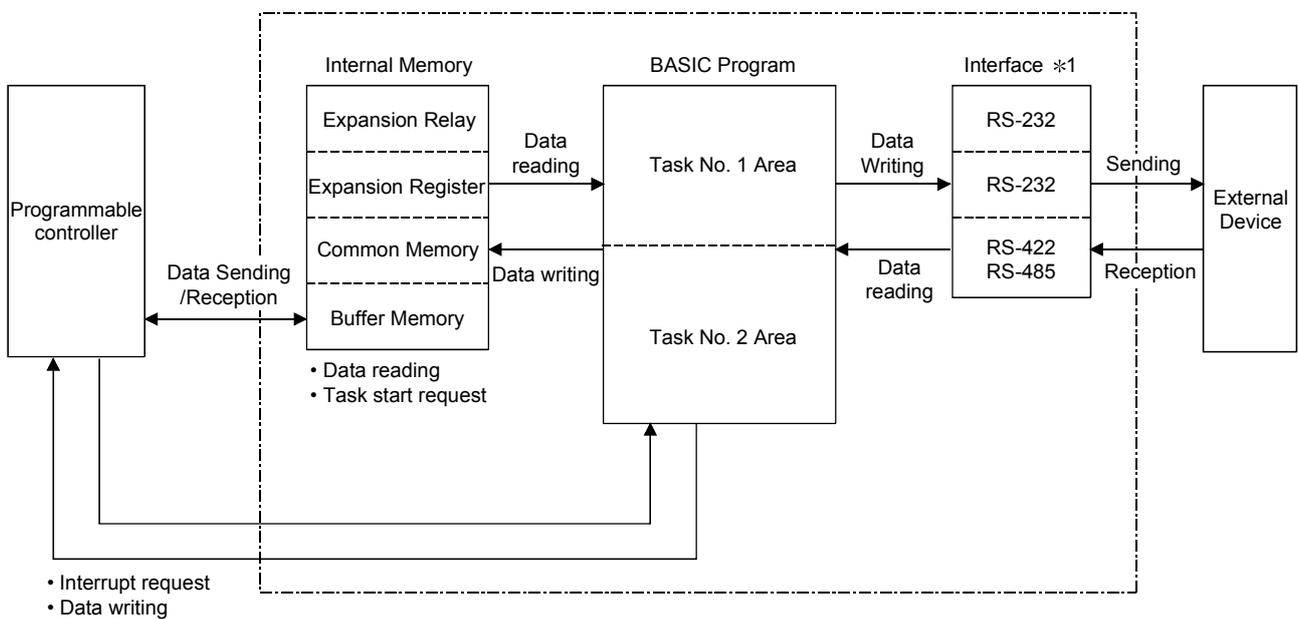
*1 In the case of the QD51
    RS-232   2 Channels
   In the case of the QD51-R24
    RS-232   1 Channel
    RS-422/485   1 Channel

(1) Sending Data to and Receiving Data from an External Device
The QD51 (-R24) is equipped with the following interfaces.
- QD51 : 2 RS-232 channels
- QD51-R24 : 1 RS-232 channel, 1 RS-422/485 channel

Production instructions, production results and other data can be sent to or received from a personal computer or other external device connected to the RS-232 or RS-422/485 interface or the console non-procedurally.
Data such as production results can also be printed out on a printer connected to the RS-232 interface.

(2) Sending Data to and Receiving Data from the Programmable Controller CPU
Through the execution of programmable controller access commands (PCRD/PCWT), the programmable controller CPU device data can be read and written.
With the programmable controller access commands, data can also be written to and read from the buffer memory of a special function module or an intelligent function module.

1.2 Features of QD51 (-R24)

Systems which use the QD51 (-R24) have the features shown below.

(1) Simultaneous execution of up to two BASIC programs (tasks)

Since multitask processing is possible, BASIC programs can be created by dividing the control tasks to be done by the QD51 (-R24).

The necessary task is started depending on the control level and the system can be controlled while exchanging data and synchronizing executions between tasks.
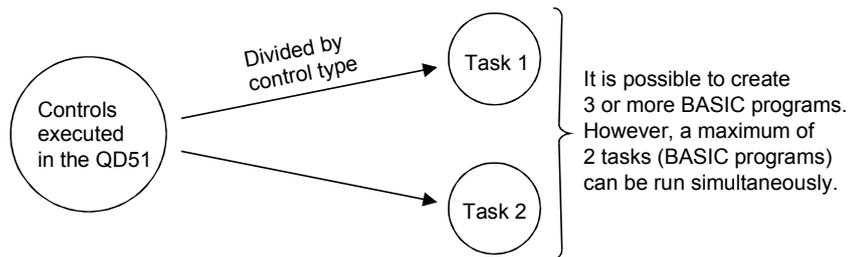
(2) Data communications with external devices

<Data communications with external devices>

Non-procedural data communications can be carried out with external devices connected to the RS-232 and RS-422/485 interfaces.

<Data communications with a console or terminal>

The following data communications can be performed with a console or terminal connected to the RS232 or RS-422/485 interface.

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

<Printout on a Printer>

Data can be printed out on a printer connected to the RS-232 interface.
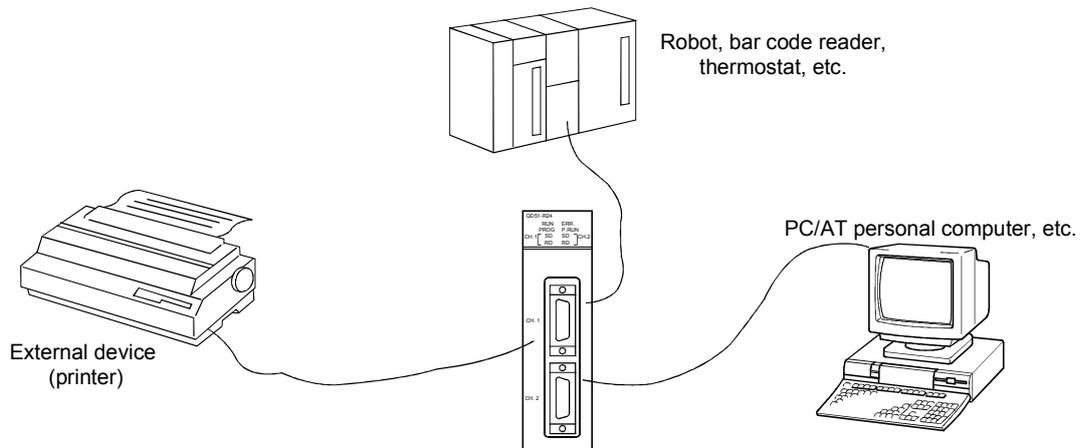
(3) Data communications with the programmable controller are possible

Data can be sent to or received from a programmable controller CPU installed with the QD51(-R24) or the one on the MELSECNET/H or MELSECNET/10 network.

The types of data which can be sent to and received from a programmable controller CPU are shown below.

1) Device writing and reading in the programmable controller CPU (Max. 960 words/1 time)

2) Writing to and reading from the buffer memory in the intelligent function module (Max. 960 words/ 1 time).

3) Remote RUN/STOP of the programmable controller CPU.

4) Interrupts to the programmable controller CPU.

(4) File control is possible

Using the console's FD or HD (hard disk), sequential files and random files can be handled.

However, it is necessary that the console be connected at all times.

<Sequential Files>

These are files in which data can be read and written sequentially and which enable efficient use of memory.

They are used mainly for operating instruction data files and results files, etc.

<Random Files>

These are files where reading starts from the necessary portion only and only the necessary portion is written.

This type of file is appropriate for large volume inventory files, instruction data for operator interfaces, master files, etc.

When reading or writing is done one time, the maximum data size is 256 bytes.

(5) Offline programming with a text editor is possible

A BASIC program can be created offline using a text editor, and the BASIC program can then be registered.

In offline programming, the edited program cannot be run.

(6) Online programming is possible

By connecting the QD51 (-R24) to a PC/AT personal computer and carrying out settings at a console, a BASIC program can be edited and debugged.

In the case of programming with a PC/AT personal computer.
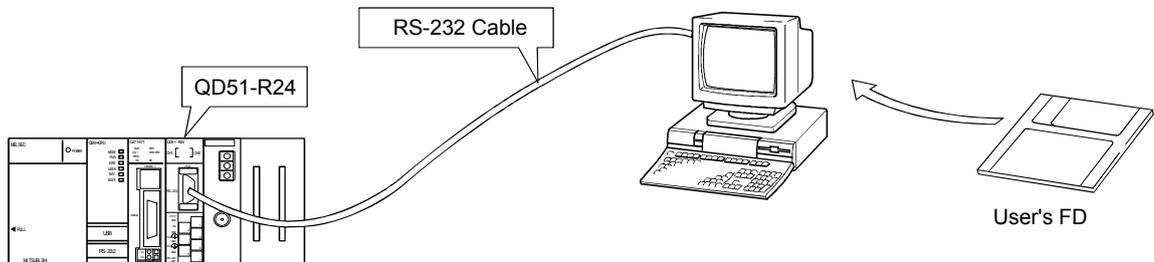
(7) Multitask debugging is possible

By connecting the QD51 (-R24) to a PC/AT personal computer and setting a debugger, debugging can be performed while performing multitask processing.

In the case of programming with a PC/AT personal computer.
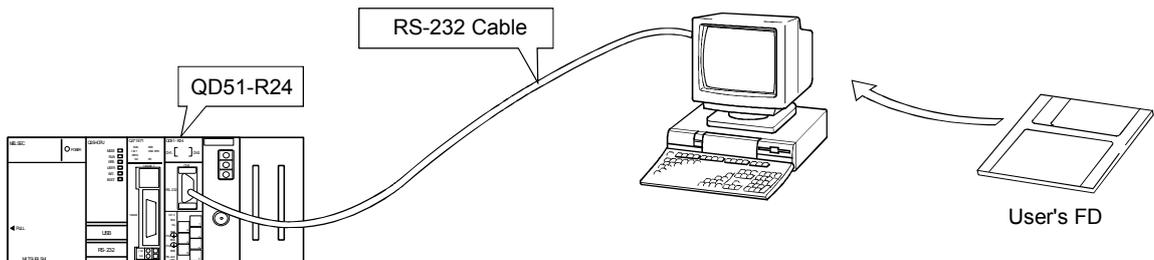
Multitask debugging is performed by inputting the Debug command from the debugger. By executing the Debug command, the following can be done.
• The BASIC program for the specified task No. area can be run and terminated.
• The variable values specified in the BASIC program for the specified task No. area can be read and written.
• Data can be read from and written to the QD51 (-R24)'s internal memory.
• Common events and message port use status displays, etc. can be shared between BASIC programs.

(8) Both interpreter BASIC and compiled BASIC can be used

By compiling a BASIC program created in interpreter BASIC with the BASIC compiler, the BASIC processing speed can be markedly increased.

Interpreter BASIC : BASIC in an execution format where the QD51 (-R24) converts the program to machine language during execution.

Compiled BASIC : BASIC in the format where the program is compiled (the commands are converted to machine language) at the stage when the program is completed and the QD51 (-R24) runs it in machine language directly.

(9) Multiple CPU system compatible functions

  (a) If access to a QCPU in a multiple CPU system is done by a BASIC
program, the access destination QCPU is specified, then reading/writing of
device data is executed.

    ∗ If the QD51 (-R24) is used in a multiple CPU system, set the QCPU that
controls the QD51 (-R24) (called the control CPU) with the GX
Developer.

    It is also possible to install the Function Version A QD51 (-R24) in a
multiple CPU system, and only the control CPU (No. 1) can be accessed.

1) : QCPU No. 1
2) : QCPU No. 2
3) : QCPU No. 3
4) : QCPU No. 4
 1 : Module controlled by QCPU No. 1.
 2 : Module controlled by QCPU No. 2.
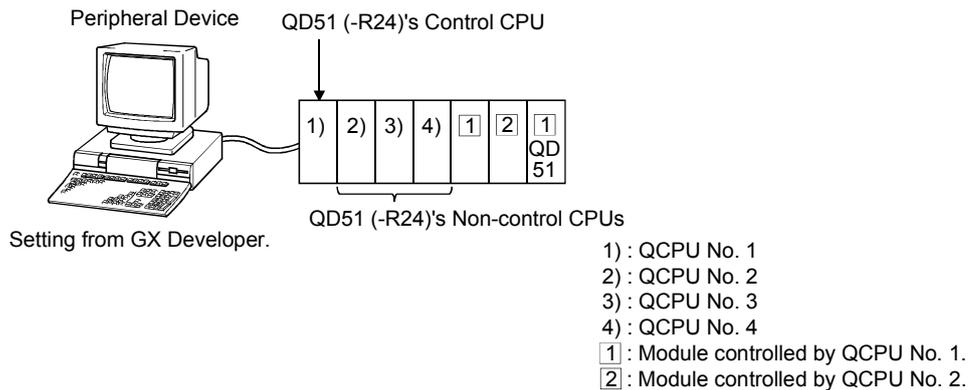
  (b) If the Function Version B QD51 (-R24) is used in a multiple CPU system,
sending the following types of data to and receiving them from the QD51
(-R24) can be done.

    1) It is possible for the control CPU to read data from and write them to
the buffer memory. It is possible to use the I/O signals as the output to
contact and output signal.

    2) It is possible for a non control CPU to read the data from the buffer
memory. Input/output signals can be used as contact points.
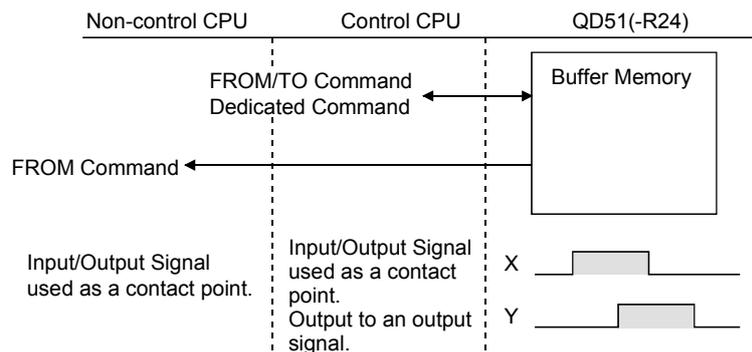
3) It is possible to access the Control CPU / Non Control CPU from the
QD51 (-R24) by a BASIC program.
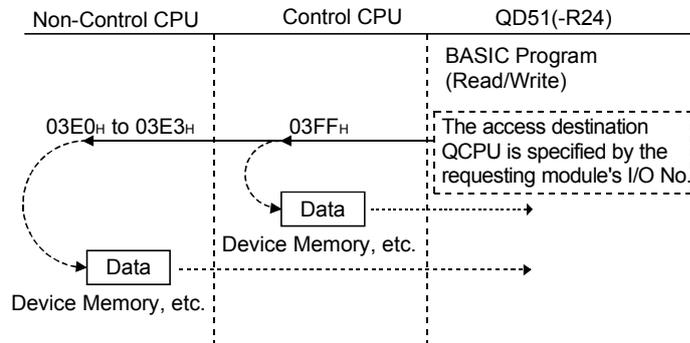
If accessing another station by a BASIC program, even if the relay
station or the access station is a multiple CPU system, the Control
CPU and Non Control CPU of the access station can be accessed.
(Example)
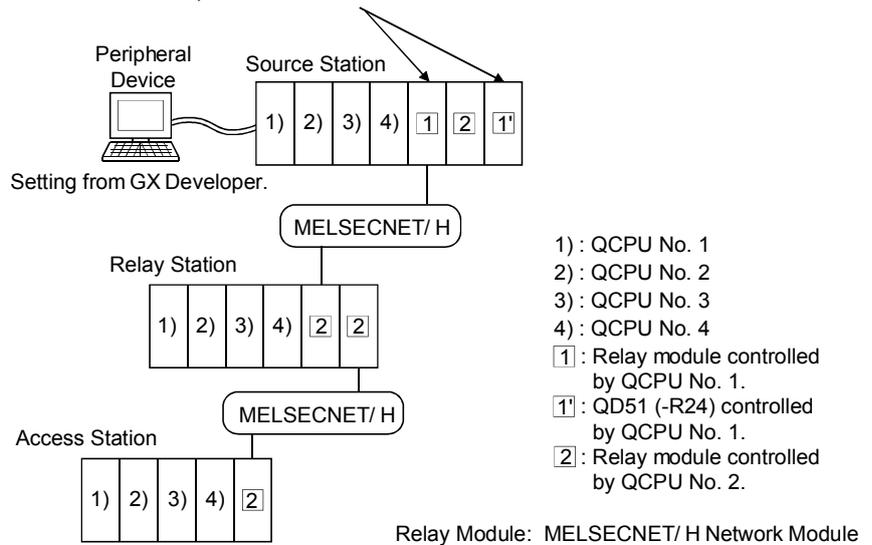
※ The relay module's object during other station access is as follows.
• MELSECNET/H, MELSECNET/10 Network Module
• Q Series C24
• Ethernet Interface Module
If a Function Version A module is included in the relay modules, only
the control CPU can be accessed.

1.3 About the Function Added/Changed to the Function Version B

This shows the functions that have been added to or changed in the Function Version B QD51 (-R24).

Function	Function overview	Explanatory section
Multiple CPU system Compatibility	When reading device data from or writing them to a multiple CPU system, it is possible to access the Control CPU / Non Control CPU specified by the user.	AD51H-BASIC Programming Manual, PCRD/PCWT Command

POINT

Refer to Section 2.6 concerning the method for checking the function version.

2 SYSTEM CONFIGURATION AND USABLE FUNCTIONS

This shows system configurations and the functions that can be used.

2.1 Application Systems

This section describes the applicable systems.

(1) Applicable modules and base units, and No. of modules

(a) When mounted with a CPU module

The table below shows the CPU modules and base units applicable to the QD51(-R24) and quantities for each CPU model.

Depending on the combination with other modules or the number of mounted modules, power supply capacity may be insufficient.

Pay attention to the power supply capacity before mounting modules, and if the power supply capacity is insufficient, change the combination of the modules.

Applicable CPU module			No. of modules *1	Base unit *2	
CPU type		CPU model		Main base unit	Extension base unit
Programmable controller CPU	Basic model QCPU	Q00JCPU	8	○	○
		Q00CPU	24		
		Q01CPU			
	High Performance model QCPU	Q02CPU	64	○	○
		Q02HCPU			
		Q06HCPU			
		Q12HCPU			
		Q25HCPU			
	Process CPU	Q02PHCPU	64	○	○
		Q06PHCPU			
		Q12PHCPU			
		Q25PHCPU			
	Redundant CPU	Q12PRHCPU	53	×	○
		Q25PRHCPU			
	Universal model QCPU	Q02UCPU	36	○	○
		Q03UDCPU	64		
		Q04UDHCPU			
		Q06UDHCPU			
		Q13UDHCPU			
		Q26UDHCPU			
		Q03UDECPU			
		Q04UDEHCPU			
		Q06UDEHCPU			
		Q13UDEHCPU			
		Q26UDEHCPU			
	Safety CPU	QS001CPU	N/A	×	× *3

2

Applicable CPU module		No. of modules [*1]	Base unit [*2]	
CPU type	CPU model		Main base unit	Extension base unit
C Controller	Q06CCPU-V	N/A	×	×
	Q06CCPU-V-B			

○: Applicable, ×: N/A

*1 Limited within the range of I/O points for the CPU module.
*2 Can be installed to any I/O slot of a base unit.
*3 Connection of extension base units is not available with any safety CPU.

(b) Mounting to a MELSECNET/H remote I/O station
The table below shows the network modules and base units applicable to the QD51(-R24) and quantities for each network module model.
Depending on the combination with other modules or the number of mounted modules, power supply capacity may be insufficient.
Pay attention to the power supply capacity before mounting modules, and if the power supply capacity is insufficient, change the combination of the modules.

Applicable network module	No. of modules [*1]	Base unit [*2]	
		Main base unit of remote I/O station	Extension base unit of remote I/O station
QJ72LP25-25	Up to 64	○	○
QJ72LP25G			
QJ72LP25GE			
QJ72BR15			

○: Applicable, ×: N/A

*1 Limited within the range of I/O points for the network module.
*2 Can be installed to any I/O slot of a base unit.

REMARKS

The Basic model QCPU module or C Controller module cannot create the MELSECNET/H remote I/O network.

(2) Support of the multiple CPU system
When using the QD51(-R24) in a multiple CPU system, refer to the QCPU User's Manual (Multiple CPU System) first.

(a) Compatible QD51(-R24)
If using the QD51(-R24) on a multiple CPU system, use function version B of the QD51(-R24) .

(3) Applicable Software Package

The following software package supports the QD51 (-R24)

(a) Console/debugger software package (essential)

Product Name	Model Name	Remarks
AD51H-BASIC Software Package	SW1IVD-AD51HP-E	For PC/AT Personal Computers

CAUTION

To use SW1IVD-AD51HP-E on Microsoft® Windows® 95 Operating System or Microsoft® Windows® 98 Operating System, operate it in the MS-DOS® mode.
Do not operate on the MS-DOS® prompt screen of Windows® 95/Windows® 98.
It will not run correctly.

(b) Software package for programmable controllers (Essential)

		Software version
		GX Developer
Q00J/Q00/Q001CPU	Single CPU system	Version 7 or later
	Multiple CPU system	Version 8 or later
Q02/Q02H/Q06H/Q12H/Q25HCPU	Single CPU system	Version 4 or later
	Multiple CPU system	Version 6 or later
Q02PH/Q06PHCPU	Single CPU system	Version 8.68W or later
	Multiple CPU system	
Q12PH/Q25PHCPU	Single CPU system	Version 7.10L or later
	Multiple CPU system	
Q12PRH/Q25PRHCPU	Redundant system	Version 8.45X or later
Q02U/Q03UD/Q04UDH/Q06UDHCPU	Single CPU system	Version 8.48A or later
	Multiple CPU system	
Q13UDH/Q26UDHCPU	Single CPU system	Version 8.62Q or later
	Multiple CPU system	
Q03UDE/Q04UDEH/Q06UDEH/ Q13UDEH/Q26UDEHCPU	Single CPU system	Version 8.68W or later
	Multiple CPU system	
When installing to a MELSECNET/H remote I/O station		Version 6 or later

(c) Commercially available compilation software package (necessary only for compilation)

When using a PC/AT personal computer

Purchase the product Turdo Assembler 5.0.

For Turbo Assembler, contact Borland Software Corporation.

Product Name	Model Name	Remarks
Turbo Assembler	Turbo Assembler Ver.5.0	IBM PC/AT English Version

(4) Connection Cable for Console/Debugger

Refer to Section 5.4 for connection to a console.

2.2 Combining Programmable Controller CPU(s) with Other Device(s)

This section shows the system configurations and functions which are available when the QD51 (-R24) is used.

(1) System Configurations

The following shows the system configurations (combinations) for data communications.

(a)   1:1 system configuration

RS-232/422

(b)   n:1 system configuration

RS-485

(c)   1:n system configuration
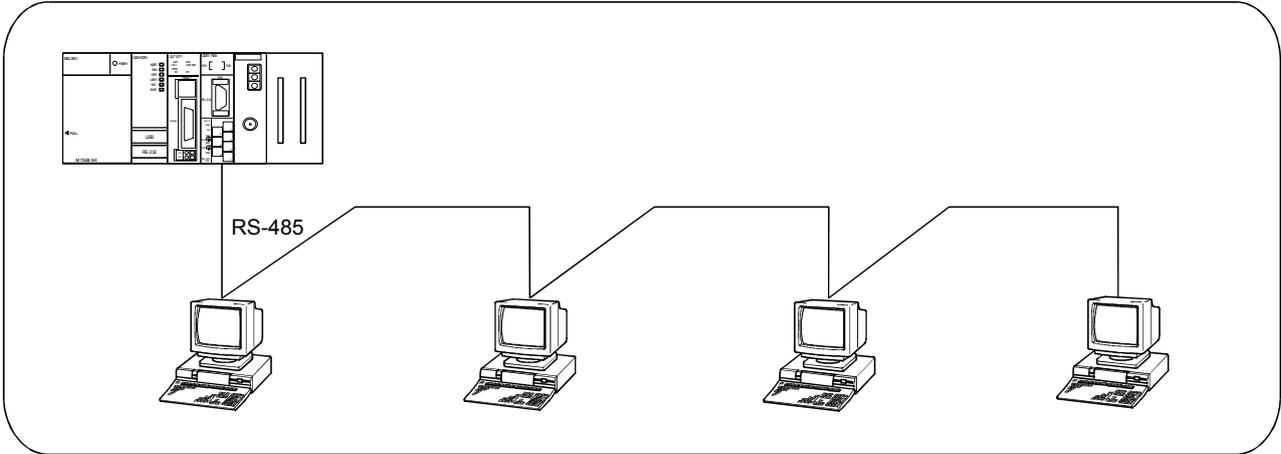
(d)   m:n system configuration
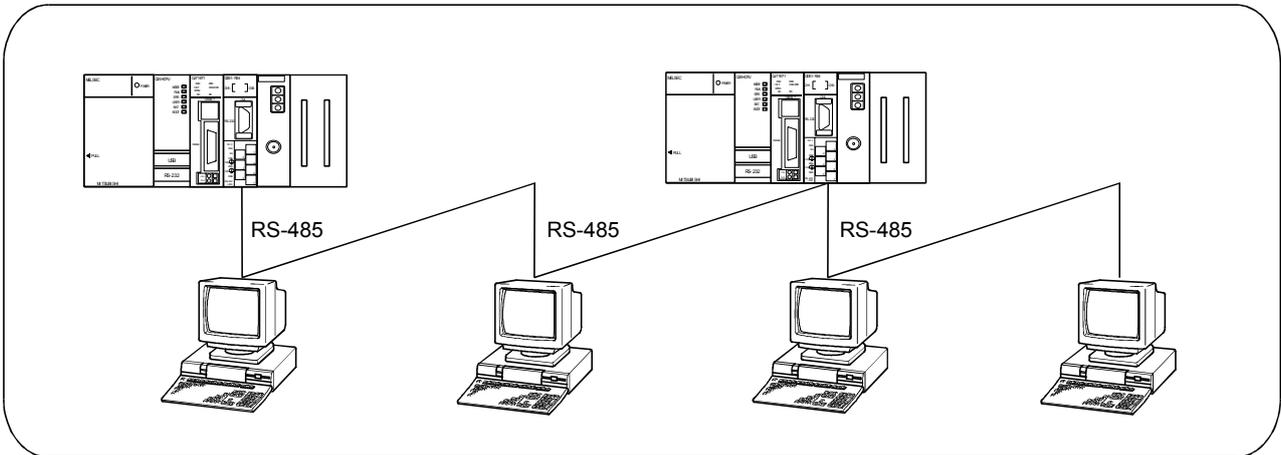
2.3 For Use in Multiple CPU System

This section describes the use of the QD51(-R24) in a multiple CPU system.

(1) Making access from a BASIC program to a non-control CPU
     When making access from a BASIC program to a non-control CPU, use the
     QD51 (-R24) of function version B.

Multiple CPU system
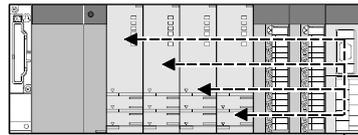
QD51(-R24) of function version B

(2) When other stations to be accessed belong to a multiple CPU system
     When accessing a CPU that is not controlling a replay module on the access
     target station, use function version B or later for the following modules:
     • QD51 (-R24) on local station
     • Relay modules *1 and QCPUs on local station, relay station, and accessed
       station
     (Example)

Whether the control CPU of the relay module is the same or different,
it is possible to access the other station.

1) : QCPU No.1
2) : QCPU No.2
3) : QCPU No.3
4) : QCPU No.4
[1] : Module used for routing, controlled
      by QCPU No.1
[1'] : QD51(-R24) controlled
      by QCPU No.1
[2] : Module used for routing, controlled
      by QCPU No.2

The modules used for the routing are MELSECNET/ H network modules
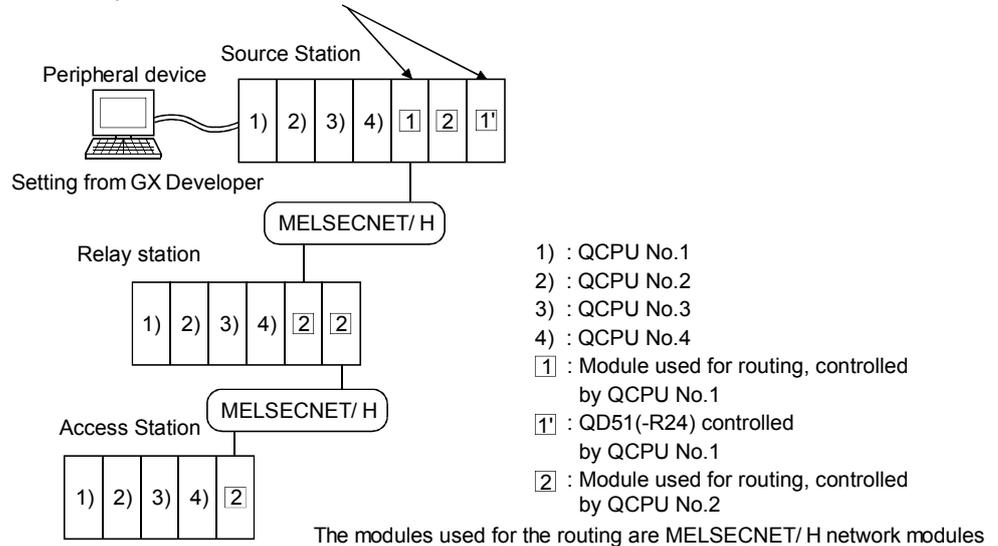
*1 When accessing other stations, available relay modules are as follows:
     • MELSECNET/H and MELSECNET/10 network modules
     • Q series C24
     • Ethernet interface modules

2.4 For Use with Redundant CPUs

This section describes the use of the QD51(-R24) with the Redundant CPUs.

(1) Dedicated instructions
The dedicated instructions are not applicable.

2.5 For Use at MELSECNET/H Remote I/O Station

This section describes the use of the QD51(-R24) on a MELSECNET/H remote I/O station.

When using the QD51(-R24) with a QCPU, it is not necessary to read this section.

(1) System Configuration

(Example)
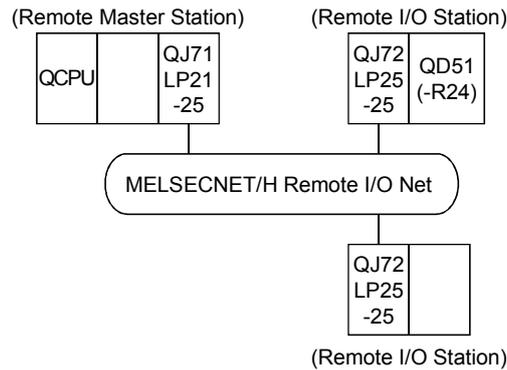
(2) Available functions

The functions which can be used when the QD51 (-R24) is installed in a MELSECNET/H remote I/O station are shown below.

Function	Availability
Communications by BASIC program (＊1)	○

○: Can be used.　　×: Cannot be used.

＊1 Available communications with MELSECNET/H remote I/O stations by BASIC program are as follows:

Available Function	Function
Reading from and writing to device memory.	Batch read, batch write
	Random read, test (random write)
Reading from and writing to intelligent function module buffer memory.	Reading from and writing to the specified intelligent function module's buffer memory.

The device range of a MELSECNET/H remote I/O station which can be accessed by reading from and writing to the device memory is as follows. (The same range as the default QCPU settings)

For MELSECNET/H remote I/O station devices and access device ranges, refer to Section 3.3.1 (3) of the AD51H-BASIC Programming Manual.

Device Name	Device Symbol	Device Name	Device Symbol
Special Relay	SM	Link Relay	B
Special Register	SD	Data Register	D
Input Relay	X	Link Register	W
Output Relay	Y	Link Special Relay	SB
Internal Relay	M	Link Special Register	SW

REMARK

For QnA/A Series MELSECNET/10 remote I/O stations, only reading from and writing to the intelligent function module's buffer memory can be done.

(3) Setting from GX Developer
In the case of mounting the QD51(-R24) on a MELSECNET/H remote I/O station, set the following parameters in GX Developer.
Each setting method is the same as the parameter setting for the QD51(-R24) installed to the QCPU station. Refer to Section 5.6 and subsequent sections.
For the method for displaying each setting screen, refer to the GX Developer Operating Manual.
(Parameter setting sections for the QD51 (-R24) when installed to a MELSECNET/H remote I/O station)

Parameter Setting Item	Setting	Remarks
I/O Assignment	Sets the module installation information.	Refer to Section 5.6.1.
Switch setting for I/O and intelligent function modules	Sets the transmission specifications and communications protocol for with other devices, etc.	Refer to Section 5.6.2.

POINT

(1) Connect GX Developer to the MELSECNET/H remote I/O station and set the parameters.

(2) After changing the settings, reset the MELSECNET/H remote I/O station.

2.6 How to Confirm the Function Version

This section describes how to confirm the function version of the QD51(-R24).

(1) How to confirm the function version of QD51(-R24)

(a) Looking at the "rating plate's SERIAL section" on the side face of the module
The serial No. and function version of the module is shown in the SERIAL field of the rating plate.
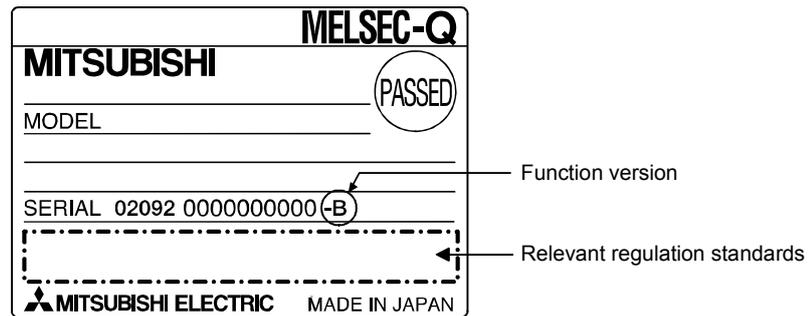
(b) Using GX Developer
The serial No. and function version of the module are displayed on the "Production Info. List" and "Module's Detailed Information" screens of GX Developer.
The following explains how to check them on the "Production Info. List" screen. (For the case of "Module's Detailed Information", refer to Section 7.1.1)

[Operating procedure]
[Diagnostics] → [System monitor] → [Product Inf. List]

[Serial No, Ver, and Production No.]
• Serial No. of the module is displayed in the Serial No. column.
• Function version of the module is displayed in the Ver. column.
• Serial No. printed on the rating plate is displayed in the Production No. column.[*1]
Note that, because the QD51(-R24) does not support the production No. display, "-" is displayed.

*1: The Production No. column display is active only when the CPU used is a Universal model QCPU.

POINT
The serial No. on the rating plate may not match the serial No. displayed on the product information screen of GX Developer. • The serial No. on the rating plate indicates the management information of the product. • The serial No. displayed on the product information screen of GX Developer indicates the functional information of the product, which is updated when a new function is added.

3 SPECIFICATIONS

The following shows the performance specifications of the QD51 (-R24).
For general specifications, refer to the QCPU User's Manual (Hardware Design, Maintenace and Inspection).

3.1 Performance Specifications

The performance specifications of the QD51 (-R24) are shown below.

Item		Specification	
		QD51	QD51-R24
Programming language		AD51H-BASIC (Interpreter, Compiler)	
Number of tasks		2	
Task start conditions		• Start by power is turned on. • Start by an interrupt from the programmable controller CPU. (Not possible when compiled BASIC is used.) • Start by a start request from another task.	
Internal memory	Program	Max. 64 kbytes (Task 1 capacity + Task 2 capacity $\leq$ 64 kbytes)	
	Common memory	8 kbytes	
	Buffer memory	6 kbytes	
	Expansion relay (EM)	1024 points	
	Expansion register (ED)	1024 points	
General Input/Output		Input: 26 points Output: 23 points	
Memory protection		Yes (Flash ROM write protectable) ($*1$)	
Interfaces	CH.1	RS-232 compatible (D-sub 9p)	RS-232 compatible (D-sub 9p)
	CH.2	RS-232 compatible (D-sub 9p)	—
	CH.3	—	RS-422/485 (Two-piece terminal block)
Communications method		Full duplex communications	
Synchronization method		Start stop synchronization method	
Transmission rate ($*2$)		300, 600, 1200, 2400, 4800, 14,400, 19,200, 28,800, 38,400 (bps) $*$ It is possible to use the 2 interfaces with a total transmission rate within 38,400 bps.	
Data format ($*2$)	Start bit	1	
	Data bits	7 or 8	
	Parity bit	Even, Odd, None	
	Stop bit	1 or 2	
Transmission control	DTR/DSR (ER/DR) control	RS-232 only can be used, RS-422/485 cannot be used.	
	RS/CS control		
	CD signal control	None	
	DC1/DC3 (Xon/Xoff control)	Both RS-232 and RS422/485 can be used.	
	DC2/DC4 control	None	
Clock function		None	
Power failure maintenance		None	
User program ROM Storage		The program area only can be stored in Flash ROM.	

<div align="right">(Continued to the next page)</div>

$*1$ For details, refer to Section 5.6.2.

$*2$ The transmission rate and data format are set using the BASIC program (ZOPEN command).
For details, refer to the Programming Manual (Commands).

3

Item		Specification	
		QD51	QD51-R24
Console		PC/AT personal computer.	
Multitask debugging		Possible (when the debugger is used)	
Line configuration ($*^3$)	RS-232	1 : 1	
	RS-422/485	—	1 : 1, 1 : n, n : 1, m : n
Transmission distance (Total cable length)	RS-232	Max. 15 m	
	RS-422/485	—	Max. 1200 m (Total cable length)
Number of writes to flash ROM		Maximum 100,000 times to the same area.	
Number of occupied I/O points		32 points per slot (I/O assignment: Intelli: 32 points) ($*^4$)	
Recommended cables	RS-232	7/0.127 □ P HRV-SV External diameter 8.5 mm or greater (Manufactured by Oki Electric Wire (Corp.) The number of pairs is shown in the □ part)	
	RS-422/485	SPEV (SB) –MPC-0.2 × 3P External diameter approx. 6.5 mm (Manufactured by Mitsubishi Electric Wire Co. Ltd.) SPEV (SB) -0.2 × 3P External diameter approx. 7.5 mm (Manufactured by Mitsubishi Electric Wire Co. Ltd.) ($*^5$)	
5 V DC internal current consumption		0.26A	0.31A
External dimensions		98 (3.86 in.) (H) × 27.4 (1.08 in.) (W) × 90 (3.54 in.) (D) [mm]	
Weight		0.20kg (0.44lb)	

$*3$ This shows combinations when the external device(s) and programmable controller CPU(s) are connected. (External device side : Programmable controller CPU side) The value, n or the total of m + n becomes a maximum of 32.

$*4$ To use the QD51(-R24), switch settings must be configured by GX Developer. Refer to Section 5.6 for the switch settings of GX Developer.

$*5$ The electrical characteristics of the recommended cables SPEV (SB) –MPC-0.2 × 3P and SPEV (SB) -0.2 × 3P are equivalent, however, their external dimensions and internal wire colors are different each other.

3.2 RS-232 Interface Specifications

This section describes the RS-232 interface specifications.

3.2.1 RS-232 connector specifications

The connector specifications for the RS-232 used to connect with another device are shown below.
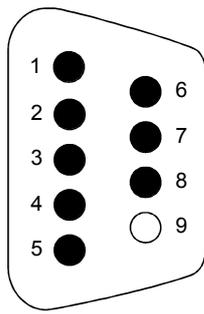
Pin No.	Signal abbreviation	Signal name	Signal direction QD51 (-R24) ⟷ External device
1	CD	Receive Carrier Detect	←
2	RD(RXD)	Receive Data	←
3	SD(TXD)	Transmit Data	→
4	DTR(ER)	Date Terminal Ready	→
5	SG	Signal Ground	← →
6	DSR(DR)	Data Set Ready	←
7	RS(RTS)	Request to Send	→
8	CS(CTS)	Clear to Send	←
9	—	—	

(1) Each control signal is explained below. (The connector pin No. is shown in parentheses.)

1) CD Signal (1)
   • Nothing is being controlled in the QD51 (-R24).
   • The signal status can be read by a BASIC program.

2) RD Signal (2)
   This is the signal for data reception.

3) SD Signal (3)
   This is the signal for data sending.

4) DTR Signal (4)
   • If DTR/DSR control is being exercised, the QD51 (-R24) turns ON or OFF depending on the size of the empty space in the OS area for reception data storage. (It is ON when it is ready to receive data.) When the DTR signal is OFF, reception data have been stored in the OS area, so read out the reception data from a BASIC program. If DTR/DSR control is not being exercised, it is normally ON.

5) DSR Signal (6)
   • If DTR/DSR control is being exercised, when this signal is OFF, data are not sent to the target device from the QD51 (-R24). When the target device is in the ready-to-receive state, set it so that it is normally ON.
   • If DTR/DSR control is not being exercised, the DSR signal's status is disregarded.

6)   RS Signal (7)
- If RS/CS control is being exercised, the QD51 (-R24) is turned ON or OFF according to the size of the empty space in the OS area for reception data storage. (It is ON when it is ready to receive data.) When the RS signal is OFF, reception data have been stored in the OS area, so read out the reception data from a BASIC program. If RS/CS control are not being exercised, it is normally ON.

7)   CS Signal (8)
- When OFF, data are not sent to the target device from the QD51 (-R24).
- When the target device is in the ready-to-receive state, set it so that it is normally ON.

REMARK

During data communications, the status of each control signal, DTR, DSR, RS and CD, can be checked with the ZCTRL command processing code 33.
Refer to the Programming Manual (Commands) for the ZCTRL command.

(2)   The ON and OFF states of each signal represent the following conditions.

	(Output Side)	(Input Side)
ON :	5 V DC to 15 V DC	3 V DC to 15 V DC
OFF :	-15 V DC to -5 V DC	-15 V DC to -3 V DC

(3)   Interface connector
The following RS-232 interface connector is used for the QD51(-R24).
- DDK Ltd.
  9- pin D-sub (female) screw type
  17L-10090-27 (D9AC) (-FA)
Use the following as a connector shell for the connection cable of the QD51(-R24) side.
- DDK Ltd.
  Plug, shell: 17JE-23090-02 (D8A) (-CG)

3.2.2 RS-232 cable specifications

- Use an RS-232 cable which complies with the RS-232 standard and which has a length of 15 m or less. (Recommended cable)
- Oki Electric Cable Co., Ltd.
  7/0.127 □P HRV-SV (□: Specify the number of pairs. For 13 pairs 7/0.127 13P HRV-SV)

3 - 5

3.3 RS-422/485 Interface Specifications

This section describes the RS-422/485 interface specifications.

3.3.1 RS-422/485 terminal block specifications

The following shows the RS-422/485 terminal block specifications for connection to the target device.

Signal abbreviation	Signal name	Signal direction QD51-R24 ↔ External device
SDA	Send Data (+)	→
SDB	Send Data (-)	→
RDA	Receive Data (+)	←
RDB	Receive Data (-)	←
SG	Signal Ground	↔
FG	Frame Ground	↔
FG	Frame Ground	↔
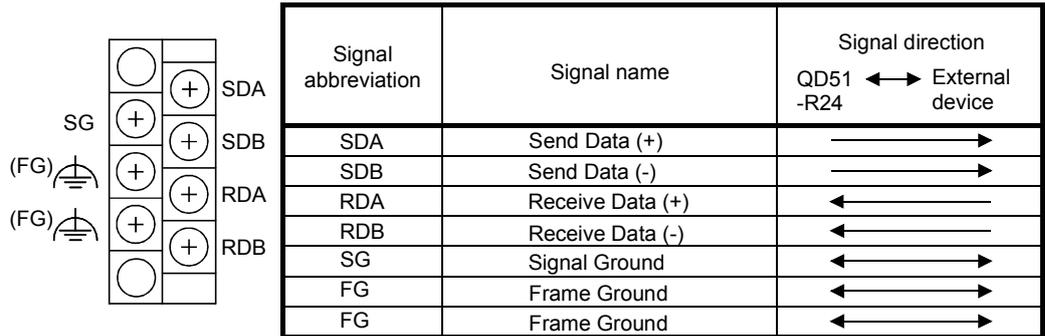
(1) Each signal is explained below.

1) SDA and SDB signals
   Signals for sending data to the target device from the QD51 (-R24).

2) RDA and RDB signals
   Signals for the QD51(-R24) to receive data from the target device.

(2) Terminal resistor
   Connect terminal resistors in accordance with Section 5.5.2.

3.3.2 RS-422/485 cable specifications

The following shows the RS-422/485 cable specifications.

(1)   Use an RS-422/485 cable (cable connected to the QD51(-R24) terminal block), which satisfies the following specifications, with a length of 1200 m or less.

(2)   Also, when connecting multiple devices at 1:n, n:1, or m:n, keep the total cable length within 1200 m.

(3)   The RS-422/485 cable specifications are shown below.

Item	Description
Cable Type	Shielded Cable
Number of Pairs	3
Conductor Resistance (20°C)	88.0 Ω / km or less
Insulation Resistance	10,000 MΩ - km or greater
Voltage Resistance	500 V DC 1 minute
Electrostatic Capacity (1 KHz)	Average 60 nF/km or less
Characteristic Impedance (100 KHz)	110 +/- 10Ω
Recommended conductor size	0.2 mm^2 to 0.75 mm^2

(Recommended Cables)
SPEV (SB) – MPC –0.2 × 3P : (Manufactured by Mitsubishi Electric Wire Co. Ltd.)
SPEV (SB) – 0.2 × 3P   : (Manufactured by Mitsubishi Electric Wire Co. Ltd.)

＊ The electrical characteristics of the recommended cables SPEV (SB) -MPC-0.2 × 3P and SPEV (SB) -0.2 × 3P are equivalent, but they differ somewhat in their external dimensions and internal wire colors.

3.3.3 Cautions during data communications with a RS-422/485 line

Caution is necessary in the following points when carrying out data communications with the opposite device via the QD51 (-R24)'s RS-422/485 interface.
As for the opposite device, please send and receive data with consideration of the following points.

(1) Countermeasure for data error reception on the external device side when connected to the RS-422/485
When data containing errors are received by the external device, install a pull-up or pull-down resistor in the external device.
By installing a pull-up or pull-down resistor (the criterion for the resistance value is approximately 4.7 kΩ, 1/4 W) it is possible to prevent reception of data containing errors.
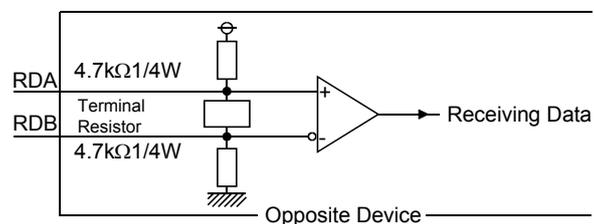
POINT
If there is a pull-up or pull-down resistor in the opposite device, there will be no errors in the data which are received.

REMARK

Here, we will explain concerning cases where a pull-up or pull-down resistor is not installed in the opposite device.
When no station is sending data, the send line is in the high impedance state.
In such a case, transmission may be unstable due to noise, etc. and thereby erroneous data may be received on the opposite device.
At such a time, it is possible that parity errors or framing errors, etc. could occur, so when an error occurs, skip over the data that are sent at such a time.

(2) RS-422/485 Interface Operation

1) RS-422/485 Interface Configuration

In the case of the RS-422-485 interface, the configuration of the QD51 (-R24) driver (Send) and receiver (Receive) are as shown in the following figure.
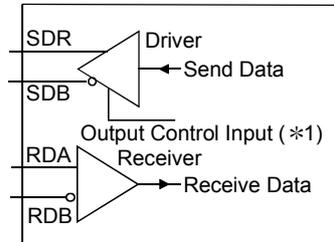
＊1 The "Output Control Input" (also called the send gate) of the driver (Send) portion in the figure at left decides whether data from SDA or SDB are output to an external device or not output.

2) RS-422/485 Interface Operation

In the above figure, when the "Output Control Input" is in the ON state, it enters the low impedance state (the state in which data can be sent). Also, when the "Output Control Input" is in the OFF state, it enters the high impedance state (the state in which data are not being sent).

3) QD51 (-R24) send start timing and send processing end timing

- Send start timing

  When data is sent, the high impedance status started by the operation described in above 1), 2) is terminated, then a mark of several bits is output, and the actual data is output.

- Send processing end timing

  After sending of data is completed, several μs are necessary for the Hardware gate OFF time until the send processing is ended (changed to the high impedance state).
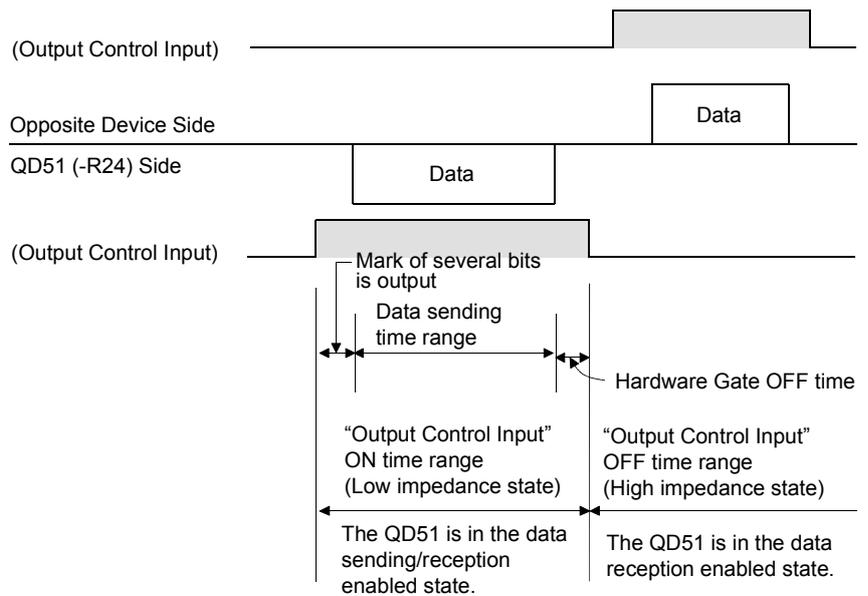
┌───┐
│ ┌──────────┐   │
│ │ POINT   │   │
│ └──────────┘   │
│ (1) When connecting each device's send signal line in system configuration of n:1 │
│   or m:n communication   │
│   Turning ON "Output Control Input" at 2 or more locations means that data are │
│   output (sent) from the devices simultaneously.   │
│   For normal data communications, the following settings are necessary on the │
│   opposite device side.   │
│   • Set the "Output Control Input" so that it is in the ON state only when │
│   sending data.   │
│   • Set the "Output Control Input" so that it is in the OFF state when data are │
│   not being sent.   │
│   The QD51-R 24 side automatically controls the output control input. │
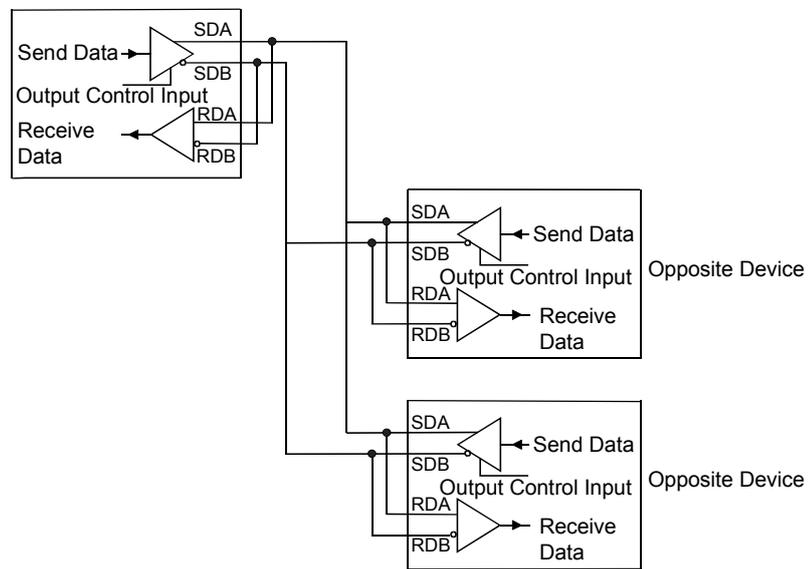
│   In the above wiring, sending and receiving are connected, the data sent by │
│   yourself is being received by yourself. Be sure to carry out programming so │
│   that such data are discarded by the BASIC program.   │
└───┘

3.4 List of Commands and Functions in AD51H-BASIC

The commands in AD51H-BASIC which can be used by the QD51 (-R24) are shown in the following table.
(For details concerning each command and function, refer to the Programming Manual (Commands).)

Classification		Command, Function	Operation
Program creation, correction commands	Program creation preparations	NEW	Deletes programs in the program area and initializes all the variables.
		AUTO	Automatically displays line numbers at the beginning of each line.
		CLEAR	Initializes all variables and sets up the memory area.
	Program editing	DELETE	Deletes the specified range of a program.
		RENUM	Reassigns line numbers of a program.
		LIST	Displays all or a part of programs in the program area.
		ZBAS	Returns the number of the BASIC task area in which the program currently being created or executed resides.
Memory capacity read function	Reading of the memory capacity	FRE	Returns the size of the unused program area in bytes.
Program execution, control commands	Program execution	RUN	Starts execution of the program in the program area.
			Reads a program from a memory card, or the FD or HD, and executes it.
		CONT	Resumes the program execution that was stopped by CTRL + C or BREAK key.
			Resumes the program execution that was stopped by the STOP or END instructions.
		RESUME	Resumes the program execution after error handling by the ON ERROR GOTO instruction.
	Program control	CHAIN	Deletes all or a part of the currently executing program, and reads and executes the specified program.
		FOR to NEXT	Repeatedly executes a series of commands the specified number of times.
		STOP	In the programming mode, pauses program execution to wait for command input.
			In the run mode, stops program execution and puts the module into an idle state.
		END	In the programming mode, terminates program execution to wait for command input.
			In the run mode, terminates program execution and puts the module into an idle state.
		WHILE to WEND	Repeatedly executes a group of commands between WHILE and WEND while the specified conditions are met.
	Program branching	GOTO	Changes the program flow to move to the specified line unconditionally.
		GOSUB to RETURN	Branches to a subroutine and returns to the line of the specified line number.
		ON GOTO	Branches to one of the lines specified with line numbers according to the value.
		ON GOSUB	Branches to one of the lines specified with the line numbers according to the value.
		IF GOTO ELSE	Selects the branch destination according to the result of the formula calculation.
		IF THEN ELSE	
Program debugging command	Program debugging	TRON	Starts program tracing.
		TROFF	Resets the program tracing set by TRON.
Format conversion command	Format conversion	CDBI	Converts a double precision real number to a 2-word (32-bit) integer for programmable controller CPUs.
		CSNI	Converts a single precision real number to a 2-word (32-bit) integer for programmable controller CPUs.

(Continued from the previous page)

Classification		Command, Function	Operation
Command to write to the designated bit	Writing to the designated bit	WTSET	Writes 0 or 1 to the specified bit of an integer array variable.
Command to read from the designated bit	Reading from the designated bit	RDSET	Reads 1-bit data from the specified bit of an integer array variable.
Character string functions	Character ↔ ASCII conversion	ASC	Returns a character code corresponding to the head of a string expression.
		CHR$	Returns a character whose code is an integer expression value.
	Character → Number conversion	STR$	Recognizes a value as a decimal number and converts it to a character string.
		VAL	Returns a numerical value that is represented by a character string.
	Number → Character string conversion	HEX$	Converts a decimal number to a hexadecimal string.
		OCT$	Converts a decimal number to an octal string.
		BIN$	Converts a decimal number to a binary string.
	Character sampling	LEFT$	Extracts the specified number of characters from the left of a character string
		MID$ (Second)	Extracts a partial string starting from the specified position in the original character string.
		RIGHT$	Extracts the specified number of characters from the right of a character string.
		STRING$	Returns the specified number of the specified characters.
		SPACE$	Returns a null string of the specified length.
	Character search	INSTR	Searches for the specified character string in the entire source string and returns the position of the first match.
	Character string length search	LEN	Returns the number of characters composing a character string.
Format control function	Format control	SPC	Returns the specified number of spaces.
		TAB	Moves the currently displayed character position to the specified position.
Screen control command	Screen control	CONSOLE	Specifies the number of items to be displayed on the console screen.
		CLS	Clears the screen display.
		LOCATE	Specifies the display position on the console screen.
Input/Output Control Commands	Port assignment	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	Makes a buzzer sound from the speaker built in the console.
	Printer output	LLIST	Outputs all or a part of programs in the program area to a printer.
		LPRINT	Outputs data to a printer.
		LPRINT USING	Outputs data in the specified format to a printer.
		LFILES	Outputs the names of the files on an FD or HD to a printer.
		ZLDV	Selects a printer port.
		WIDTH	Sets the output width in the printer.
	Output to screen	PRINT	Displays data on the screen.
		PRINT USING	Displays a character string or a numeric value in the specified format.
		KEY LIST	Displays a character string defined for a function key on the screen.
	Output to port	ZSEND	Sends data from the specified port.

Classification		Command, Function	Operation
Input/Output Control Commands	Input from keyboard	INPUT	Used to input data from the keyboard.
		LINE INPUT	Stores an entire key-input line (255 characters or less) into the character string variable area without any breaks.
	Input from port	ZRECEIVE	Receives data from a communication port.
	Changing of port settings	ZCNTL	Used to set communication data to the currently open communication port, or to read out the communication port status.
Input Functions	Inputs from the keyboard	INKEY$	Returns a character input from the keyboard, or a blank string (" ") if nothing is input.
		INPUT$	Reads a character string of the specified length from the console keyboard and returns it.
			Reads a character string of the specified length from a sequential file and returns it.
			Reads a character string of the specified length from a communication port of the QD51(-R24).
Port Control Functions	Port control	COM ON/OFF/STOP	Enables, disables or stops interrupts from communication lines.
		ON COM GOSUB	Defines the starting line of the processing routine, to which the operation branches upon interrupt from a communication line.
Error Control Functions	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	Generates an error of the specified error code.
Error Information Functions	Error Information	ERR	Returns a detected error code.
		ERL	Returns the number of the line where an error was detected.
Recursive Commands to the Programming Mode	Recursion to the programming mode	SYSTEM	Stops the control of the BASIC interpreter and returns to the QD51(-R24)'s system mode.
			Stops offline programming and returns to the main menu.
File Operation Commands	Input / Output of BASIC programs	LOAD	Reads programs on an FD or HD into the program area.
		MERGE	Merges a program in the program area with another on an FD or HD.
		SAVE	Saves programs to an FD or HD.
	File maintenance	FILES	Displays the names of the files stored on an FD or HD.
		KILL	Deletes files from an FD or HD.
			Deletes system names in an FD or HD.
		NAME	Renames a program or data file on an FD or HD.
Input / Output Commands to Files	File bugger assignment	OPEN	Opens a file and enables its I/O processing.
		CLOSE	Terminates I/O processing of a file.
	Sequential file input / output	PRINT #	Writes data to a sequential file.
		PRINT # USING	Writes data to a sequential file using the specified format.
		INPUT #	Reads data from a sequential file.
		LINE INPUT #	Reads out an entire line (255 characters) into the character string variable area without any breaks.
	Random file input / output	FIELD	Allocates the area for the specified variable to the random file buffer.
		LSET	Left-aligns and moves data to the random file buffer for execution of the PUT instruction.
		RSET	Right-aligns and moves data to the random file buffer for execution of the PUT instruction.
		PUT	Writes 1-record data from the buffer to a random file.
		GET	Reads 1-record data from a random file into the buffer.

Classification		Command, Function	Operation
File information functions	File information	EOF	Returns -1 if the end of a sequential file is detected.
		LOF	Returns the file size as the number of sectors.
		LOC	Returns the current logical location within a file.
File Data	Character string → Number	CVI	Converts a character string converted by the MKI$ function back to an integer.
		CVS	Converts a character string converted by the MKS$ function back to a single precision real number.
		CVD	Converts a character string converted by the MKD$ function back to a double precision real number.
		CVSMBF	Converts a character string, which was converted from an AD51H-BASIC single precision real number by the MKS$ function, into the internal representation of a floating point real number (IEEE single-precision representation) used in QCPU, AnA/AnU/QnACPU, AnUSCPUs.
		CVDMBF	Converts a character string, which was converted from an AD51H-BASIC double precision real number by the MKD$ function, into the internal representation of an IEEE double precision number.
	Number → Character string	MKI$	Converts integer-type numbers to character strings.
		MKS$	Converts single-precision type numbers to character strings.
		MKD$	Converts double-precision type numbers to character strings.
		MKSMBF$	Converts floating point real number data in internal representation (IEEE single-precision representation) used in QCPU, AnA/AnU/QnACPU, AnUSCPUs into a character string that can be changed into a numeric by the CVS function.
		MKDMBF$	Converts IEEE double-precision data in internal representation into a character string that can be changed into a numeric by the CVD function.
Definition, declaration commands	Definitions, declarations	DEF FN	Defines a user function and names it.
		DEFINT	Defines variables that start with the specified characters as integer types.
		DEFSNG	Defines variables that start with the specified characters as single precision real number types.
		DEFDBL	Defines variables that start with the specified characters as double precision real number types.
		DEFSTR	Defines variables that start with the specified characters as character types.
		KEY	Defines a character string to each of the console's function keys.
		DIM	Specifies the size of an array variable and allocates a necessary memory area.
		ERASE	Deletes the array defined by the DIM instruction from the memory.
		REM	Provides comments (remarks) in the program.
		COMMON	Sets variables, etc. used for transition to the program to be executed by the CHAIN instruction.
Number, character string operation commands	Data operations	DATA	Specifies numeric values or character strings to be read by the READ instruction.
		READ	Reads values defined by the DATA instruction and assigns them to variables.
		RESTORE	Specifies the first line number of the DATA instruction data that are to be read by READ.
	Substituting for variables	LET	Assigns the expression value to a variable.
		SWAP	Swaps the values of two variables.
		ZMOVE	Transfers data between variables.
		BSWAP	Swaps two values in byte units.
	Replacement of character strings	MID$ (Part1)	Replaces a part of a character string with another character string.

(Continued from the previous page)

Classification		Command, Function	Operation
Number functions	Arithmetic functions	ABS	Returns an absolute value for the expression.
		ATN	Returns the arctangent ($\tan^{-1}$) for the expression.
		COS	Returns the trigonometric cosine value.
		EXP	Returns the exponential function value of base e.
		FIX	Returns only the integer part after truncating the fractional part of a numeric value.
		INT	Returns an integer value for the expression
		LOG	Returns a natural logarithm value.
		RND	Returns a random number.
		ROT	Rotates the memory contents of the specified value and returns the bit-rotated value.
		SGN	Returns the sign of a value.
		SIN	Returns the trigonometric sine value.
		SQR	Returns the square root value.
		TAN	Returns the trigonometric tangent value.
		SHA	Arithmetically shifts the memory contents of the specified value and returns the shifted value.
		SHT	Logically shifts the memory contents of the specified value and returns the shifted value.
	Number search	SEARCH	Searches for the specified value from the elements of a given array variable and returns the location of the element.
	Format conversion functions	CDBL	Converts an integer or single precision real number into a double precision real number.
		CSNG	Converts an integer or double precision real number into a single precision real number.
		CINT	Converts a single or double precision real number into an integer.
		CIDB	Converts a 2-word (32-bit) integer used in programmable controller CPUs into a double precision real number.
		CISN	Converts a 2-word (32-bit) integer used in programmable controller CPUs into a single precision real number.
Programmable controller access functions	Programmable controller access	PCRD	Reads various data from a programmable controller CPU.
		PCWT	Writes various data to a programmable controller CPU.
			Operates a programmable controller 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 memory access functions	Buffer memory access	GETMEM	Reads data from the QD51(-R24)'s buffer memory, common memory or extension registers (ED).
		PUTMEM	Writes data to the QD51(-R24)'s buffer memory, common memory or extension registers (ED).
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 functions	TIME$	Sets up the time data (Hr, Min, Sec) of a programmable controller CPU and reads it out.
		DATE$	Sets up the date data (Yr, Mon, Day, Day of Week) of a programmable controller CPU and reads it out.

Classification		Command, Function	Operation
Real time control commands	Real time control	ZEVENT	Enables or disables event generation.
		DEF ZEVENT	Defines an event for synchronizing execution between programs.
			Defines an event to be triggered according to the QD51(-R24)'s extension relay (EM).
		ZRELEASE	Allows other programs to use a resource to which a resource number is assigned.
		ZRESERVE	Prohibits other programs from using a resource to which a resource number is assigned.
		ZSIGNAL	Generates the specified event from the program.
		ZSTART	Starts the specified program.
		ZURGENCY	Changes the priority of a program.
Real time control commands	Real time control	ZWAIT DELAY	Pauses program execution until the specified time has elapsed.
		ZWAIT EVENT	Pauses program execution until the specified event is generated.
Real time control functions	Real time control	ZMESSAGE	Defines a message port.
		ZMESSAGE CLOSE	Closes a message port.
		ZMESSAGE GET	Reads message from a message port.
		ZMESSAGE KILL	Deletes a defined message port.
		ZMESSAGE OPEN	Opens a message port.
		ZMESSAGE PUT	Writes the following messages to the message port.

3.5 QD51 (-R24) Internal Memory

The following memory can be used by the user in the QD51 (-R24).

(1) Memory where programs for execution can be stored
• Program Area ·Refer to Section 3.5.1
This is memory where BASIC programs which run the QD51 (-R24) are stored.
The program area can execute a maximum of two tasks and it has a maximum
capacity of 64 k bytes.

(2) Memory for communications

| Item | Content | Address | | | Reference Section |
| | | Programmable controller CPU | BASIC Program | | |
		FROM/TO command	GETMEM/PUTMEM command	Special Variable B@/W@	
Communications with the programmable controller CPU	Buffer memory	0000_H to $0BFF_H$	0000_H to $17FF_H$	—	Section 3.5.2
Communications between tasks	Common memory	$0C00_H$ to $1BFF_H$	1800_H to $37FF_H$	—	Section 3.5.3
Reading/writing ON/OFF data	Expansion register	$1C00_H$ to $1FFF_H$	—	ED0000 to ED1023	Section 3.5.4
	System Area	—	—	—	—
	Expansion relay	2380_H to $23BF_H$	—	EM0000 to EM1023	Section 3.5.6
	System Area	—	—	—	—
	Special register	2400_H to $247F_H$	—	ED9000 to ED9127	Section 3.5.5
	System Area	—	—	—	—
	Special relay	2500_H to 2507_H	—	EM9000 to EM9127	Section 3.5.7
	System Area	—	—	—	—

```
┌─────────────────────────────────────────────────────────────────┐
│ ┌──────────┐                                                       │
│ │ CAUTION  │                                                       │
│ └──────────┘                                                       │
│    Internal memory is not backed up by a battery.                 │
│    If the QD51 (-R24)'s power is turned OFF, the BASIC program written to it and │
│    the data in the buffer memory, etc. are erased. Therefore, observe the following │
│    (1) and (2).                                                    │
│ (1)  Save the created BASIC programs in a flash ROM before turning the power │
│       off.                                                         │
│       The following functions stop during a flash ROM update.     │
│       • Communications with external devices                      │
│       • Communications with the programmable controller CPU       │
│       • Execution of the BASIC program                            │
│       • Time measurement of the timeout processing                │
│       It is recommended to update the flash ROM before any of these operations │
│       and do not update during the operations.                    │
│       It may take one to two minutes to update the flash ROM.     │
│ (2)  Transmit  data of the buffer memories, etc. to a latched device in the │
│       programmable controller or to a file register.              │
└─────────────────────────────────────────────────────────────────┘
```

3.5.1 Program area

Following programs can be stored in the program area:
- BASIC programs created at the console
- Programs for execution read from the flash ROM

(1) Memory Capacity

The program area's maximum capacity is 64 k bytes.

This 64 k bytes is divided so it can be used for a maximum of 2 BASIC programs (Task 1 to Task 2).

```
┌─────────────────────┐ ╲
│  Task No. 1 Area    │  ╲
│  (Max. 64 k bytes)  │   ⎫ Maximum 64 k bytes (total for 2 tasks)
├─────────────────────┤   ⎬
│  Task No. 2 Area    │  ╱
│  (Max. 64 k bytes)  │ ╱
└─────────────────────┘
```
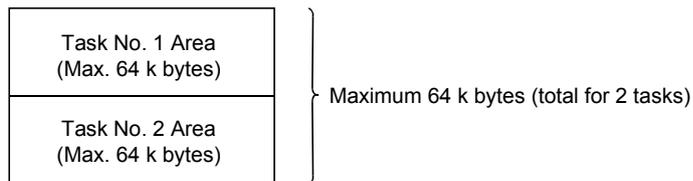
(2) Capacity for 1 BASIC program

The capacity for one BASIC program can be selected from 16 k bytes, 32 k bytes, 48 k bytes and 64 k bytes.

```
┌──────────────────────────────────────────────────────────────────────┐
│ CAUTION                                                                │
├──────────────────────────────────────────────────────────────────────┤
│ If compiled BASIC is used, approximately 2 times as much capacity is   │
│ required as with interpreter BASIC.                                    │
└──────────────────────────────────────────────────────────────────────┘
```

(3) Configuration for each Task No. Area

Besides the area for the BASIC program, an OS area, an area for the variables used in that task and a random file buffer area are included in each Task No. Area.

```
┌─────────────────────────┐ ╲
│      OS Area            │  ╲
│      (1 k byte)         │   
├─────────────────────────┤   
│                         │   
│  BASIC Program Area     │   ⎫ One Task No. Area
│                         │   ⎬ (16 k bytes, 32 k bytes, 48 k bytes and 64 k bytes)
├─────────────────────────┤   
│ Random File Buffer Area │   
├─────────────────────────┤  ╱
│    Variable Area        │ ╱
└─────────────────────────┘
```

POINT

A program for running the flash ROM is read to the QD51 (-R24)'s program area when the module is powered ON or is reset.

However, for an area where the multitask setting is set "ON", reading to the QD51 (-R24) is not done.

Program for running flash ROM			QD51 (-R24) Program Area
BASIC Task 1 Area	START	BASIC Program →	Task 1 Area
BASIC Task 2 Area	ON	→ ×	Task 2 Area

3.5.2 Buffer memory

Buffer memory is memory for carry out sending and receiving of data between the QD51 (-R24) and the programmable controller CPU.
(Data from the programmable controller CPU that are written to buffer memory can be read from multiple tasks. Data written to buffer memory from multiple tasks can be read from the programmable controller CPU.)

(1)   Reading/Writing of Data

(a)   Data from the programmable controller CPU are read from or written to the buffer memory by the FROM/TO command.

Program example
The following shows a program where values in D0 to D9 are written into the buffer memory area addressed 0_H to 9_H when the multitask execution start signal (X0B) is ON.
To write data from the programmable controller CPU to the buffer memory or to read data from the buffer memory, use the multitask execution start signal (X0B) as an interlock.

Write
Command   X0B
──┤├──────┤├──────────────────────────[T0 H0 H0 D0 K10]──

└──────── Interlock that enables or disables FROM/TO execution.

QD51(-R24)

Buffer Memory

Sequence Program (1-word units) ──TO (Writing of data)──>

0_H		0_H
		1_H
1_H		2_H
		3_H
to		to
8_H		10_H
		11_H
9_H		12_H
		13_H
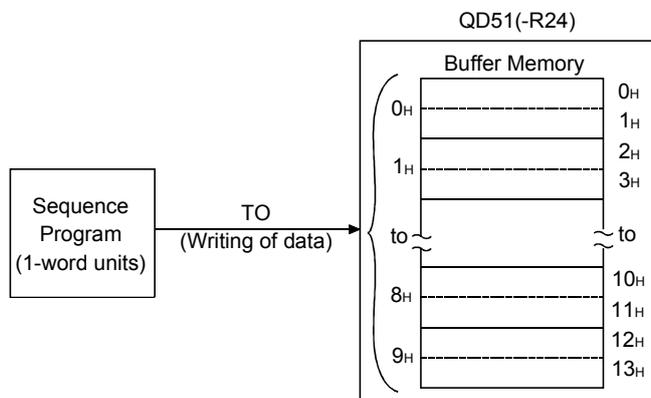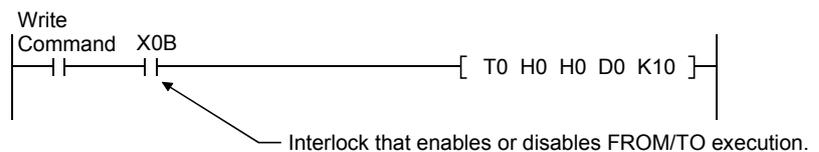
(b)   In the QD51 (-R24), reading of data from, or writing of data to buffer memory is done using the GETMEM/PUTMEM command in the BASIC program.

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

(2)   **Buffer Memory Capacity**
The capacity of the QD51 (-R24)'s buffer memory is 3 k words (6 k bytes).

(3)   **Buffer Memory Configuration**
One address of the buffer memory is composed by 16 bits.
The entire area can be used as desired by the user.

REMARK

Refer to the Programming Manual (Commands) concerning the GETMEM and PUTMEM commands.

(4)   Buffer Memory Addresses

Addresses in buffer memory differ depending on whether they are designated by the programmable controller CPU or by a BASIC program, and are as follows.

(a)   If designated from the programmable controller CPU

If designated by the programmable controller CPU, addresses are in 1-word units.

Also, buffer memory addresses are designated from 0_H to BFF_H in hexadecimal notation.

(b)   If designated by a BASIC program

If designated by a BASIC program, addresses are in 1-byte units.

Also, buffer memory addresses are designated from 0_H to $17FF_H$ in hexadecimal notation.

3.5.3 Common memory

Common memory is memory that can be used to send and receive data between the QD51 (-R24) internally and each task.
Data from the programmable controller CPU can also be read and written using this memory.

(1) Writing data to and reading data from common memory
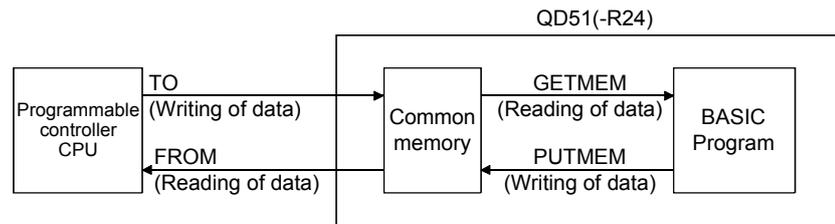
(a) In a BASIC program, data are read from or written to common memory using the GETMEM/PUTMEM command.

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
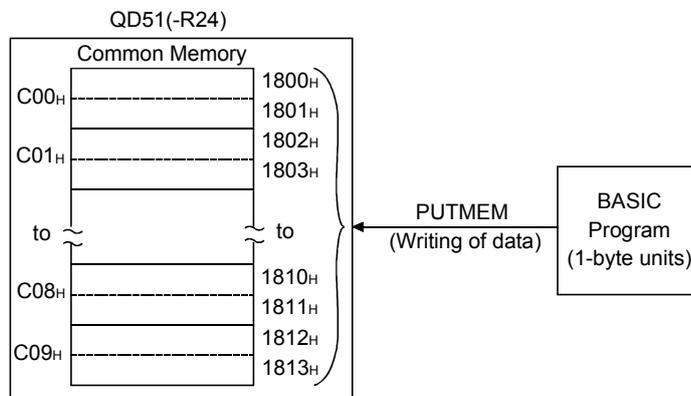
(b) In a sequence program, data are read from or written to common memory using the FROM/TO command.
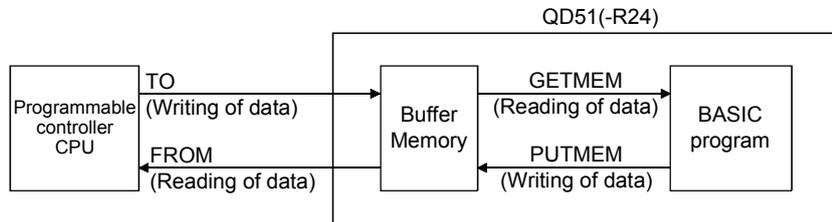
Program example

The following shows a program where values in D0 to D9 are written into the common memory area addressed $0C00_H$ to $0C09_H$ when the multitask execution start signal (X0B) is ON.

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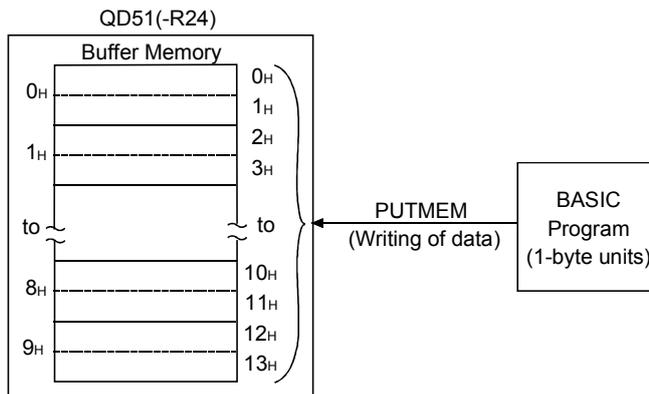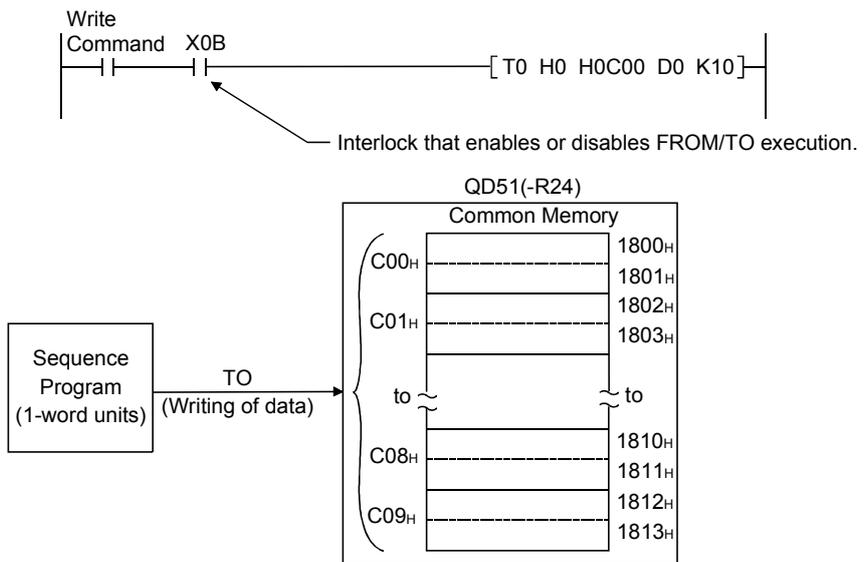

(2) Common Memory Capacity

In the QD51 (-R24), the size of common memory is 8 k bytes

(3) Common Memory Configuration

One address in common memory has a 1-byte (8 bits) configuration. The entire area can be used as desired by the user.

(4) Common Memory Addresses

Addresses in common memory differ depending on whether they are designated by the programmable controller CPU or by a BASIC program, and are as follows.

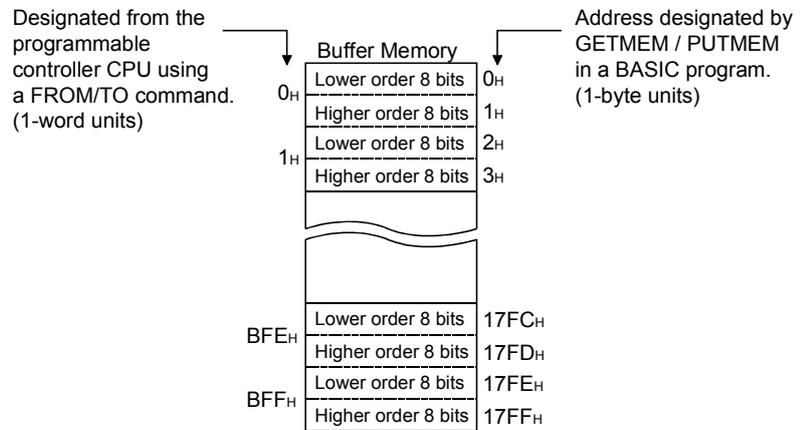(a) If designated from the programmable controller CPU
If designated by the programmable controller CPU, addresses are in 1-word units.
Also, common memory addresses are designated from C00$_H$ to 1BFF$_H$ in hexadecimal notation.

(b) If designated by a BASIC program
If designated by a BASIC program, addresses are in 1-byte units.
Also, common memory addresses are designated from 1800$_H$ to 37FE$_H$ in hexadecimal notation.
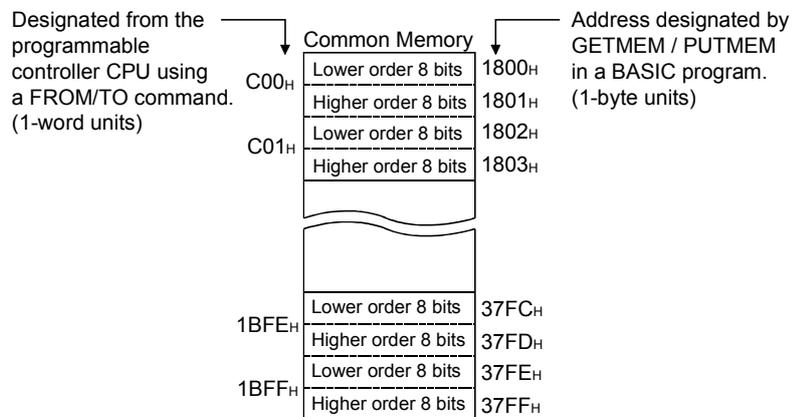
Designated from the programmable controller CPU using a FROM/TO command. (1-word units)	Common Memory		Address designated by GETMEM / PUTMEM in a BASIC program. (1-byte units)
C00$_H$	Lower order 8 bits	1800$_H$	
	Higher order 8 bits	1801$_H$	
C01$_H$	Lower order 8 bits	1802$_H$	
	Higher order 8 bits	1803$_H$	
1BFE$_H$	Lower order 8 bits	37FC$_H$	
	Higher order 8 bits	37FD$_H$	
1BFF$_H$	Lower order 8 bits	37FE$_H$	
	Higher order 8 bits	37FF$_H$	

REMARK

Refer to the Programming Manual (Commands) concerning the GETMEM and PUTMEM commands.

3.5.4 Expansion register (ED0 to ED1023)

The expansion register is internal memory where 1-word (16-bit) data can be stored. ＊1
If the expansion register is used, data can be stored internally in the QD51 (-R24) and
1-word data can be sent and received between tasks and between the QD51 (-R24)
and the programmable controller CPU. ＊2
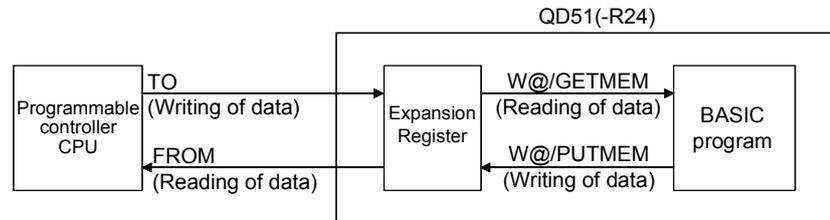
REMARK

＊1   The data in the expansion register are not backed up by a battery, so if the
programmable controller's power is turned OFF or it is reset, the data written to
the register are erased.
If the data need to be held even after the programmable controller is powered
OFF or reset, save the expansion register data in the following devices.
• File register of the programmable controller CPU
• Latched device of the programmable controller CPU

＊2   When data are sent and received between devices in 1-word units, the buffer
memory (Refer to Section 3.5.2) and common memory (Refer to Section 3.5.3)
can also be used.

(1) Writing data to and reading data from the expansion register
Data are written to or read from the expansion register by the following methods.
• Using the special variable W@ of the BASIC program
• Using the GETMEM/PUTMEM command in the BASIC program
• Using the FROM/TO command in the sequence program
The method of designating the expansion register differs in the special variable
W@, the GETMEM/PUTMEM command and the FROM/TO command, as shown
below.

(a) Writing/Reading of data by the special variable W@ in a BASIC program
With the special variable W@, the expansion register is designated by ED0 to ED1023. The designation is in 1-word units.
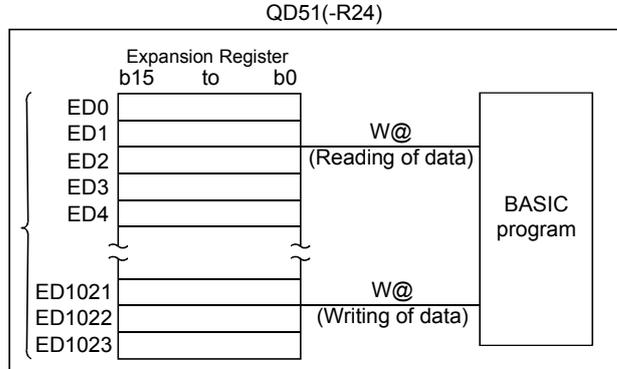
QD51(-R24)

Program example
The following shows a program where "123" is written into the QD51 (-R24)'s expansion register ED0.

150 W@ (ED, 0) =123

QD51(-R24)
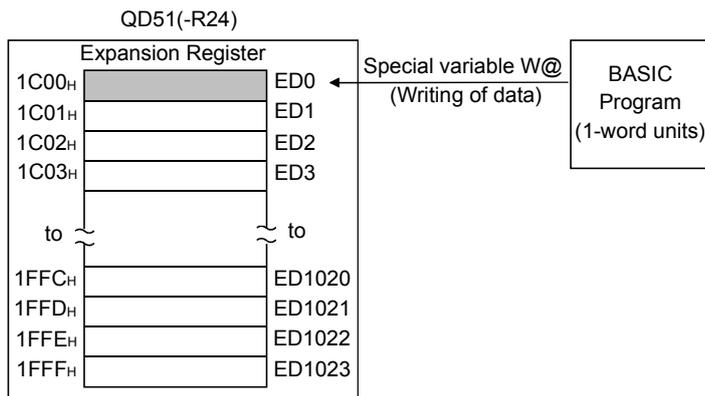
(b) Writing/Reading data by the PUTMEM/GETMEM command in a BASIC
program
With the PUTMEM/GETMEM command, data can be written to or read from
the extension register.
An address can be specified by ED0 to ED1023 in units of 1 byte.
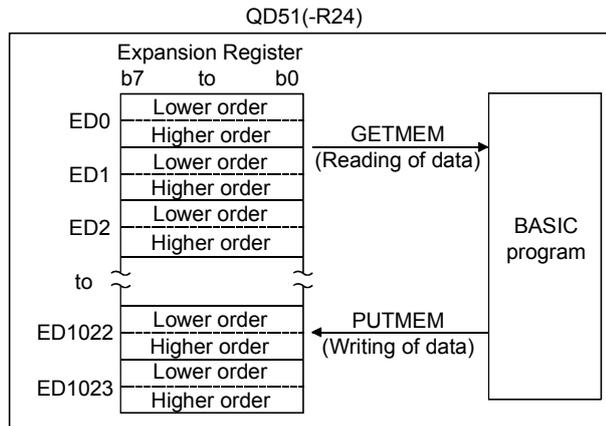
QD51(-R24)

Program example
In the following program, data (4 bytes) in A% (1) and A% (2) of the
QD51 (-R24) are written into ED4 and ED5.

20 PUTMEM TO W@ (ED,2), 4 FROM A%(), 2, 4
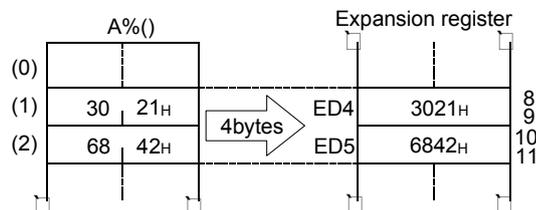
(c) Writing/Reading data by the FROM/TO command in a sequence program
With the FROM/TO command, designate the expansion register by the
address. The designation address is 1C00H to 1FFFH, in 1-word units.
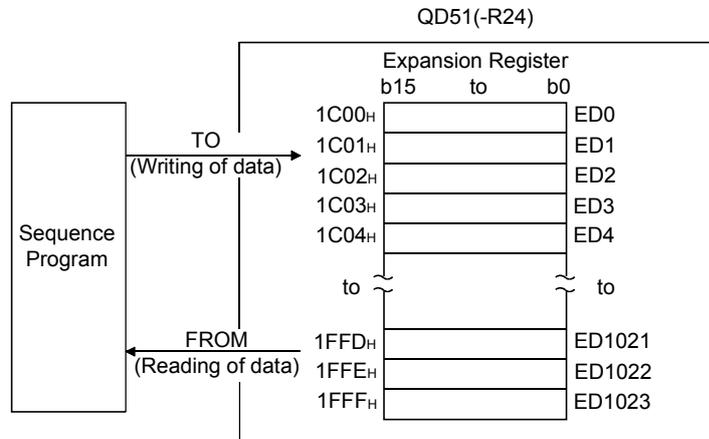
QD51(-R24)

	Expansion Register
	b15 to b0

Program example
The following shows a program where "123" is written into the
expansion register address 1C00$_H$ when the multitask execution start
signal (X0B) is ON.
To write data from the programmable controller CPU to the expansion
register or to read data from the expansion register, use the multitask
execution start signal (X0B) as an interlock.

Interlock that enables or disables FROM/TO execution.
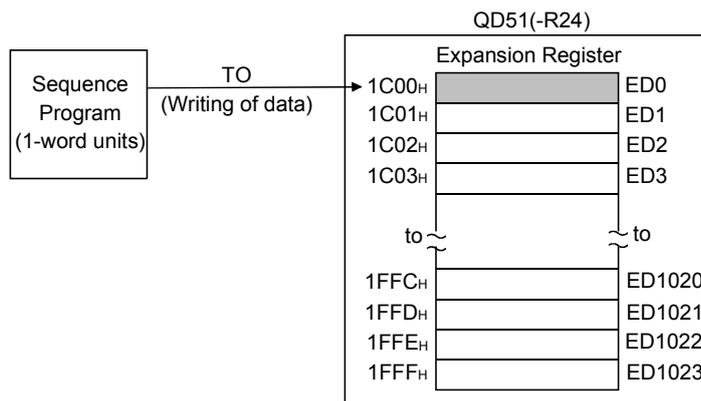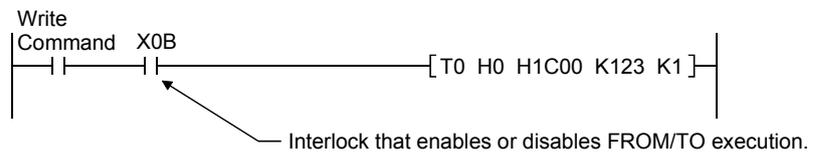
QD51(-R24)

(2) During execution of a BASIC program by multitasking, data can be
written or read from either task.
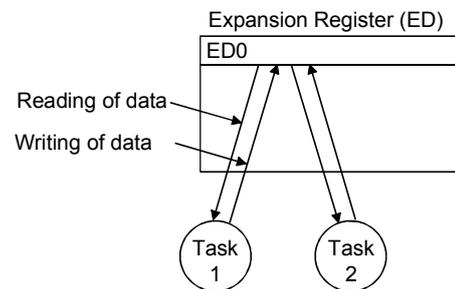
(3) Number of Expansion Register Points
The expansion register in the QD51 (-R24) is 1024 points, from ED0 to ED1023.
(The expansion register No. in ED0 to ED1023 is expressed in decimal notation.)

3.5.5 Special register (ED9000 to ED9127)

The special register is a register which has its use determined by the QD51 (-R24).

Data are stored by the QD51 (-R24)'s OS, so in a BASIC program, use the special variable W@ to read and use the stored data.

Buffer Memory Address as seen from the CPU	Special Register No.	Name	Content	Details of Contents	Related Special Relay	Related Special Register
&H2400	ED9000	Number of Errors	0 to 8	• The number of errors registered in the QD51 (-R24) is stored.	EM9000	ED9001 to ED9008
&H2401 to &H2408	ED9001 to ED9008	Error Code	0: Normal Other than 0: Error Code	• Error codes are stored in order in ED9001 to ED9008. • If more than 8 error occur, the error codes are not stored. • For error codes, refer to the Programming Manual (Commands).	EM9000	ED9000
&H2409 to &H2410	ED9009 to ED9016	Error Line No.	0: No Line No. 1 to FFFF: Line No.	• The line No. of errors detected by BASIC are stored. • Correspondence between the error code and the special register for storing the line numbers is as shown below. { For Error Code Storage / For Line Number Storage: ED9001 / ED9009 ED9002 / ED9010 ED9003 / ED9011 ED9004 / ED9012 ED9005 / ED9013 ED9006 / ED9014 ED9007 / ED9015 ED9008 / ED9016 }	EM9000	ED9001 to ED9008
&H2414	ED9020	Run Mode	0 to 4	• The setting number of the mode setting switch is stored. • 0, 1 : Run Mode • 2, 3 : Multitask Debug Mode • 4 : Programming Mode	—	—

Buffer Memory Address as seen from the CPU	Special Register No.	Name	Content	Details of Contents	Related Special Relay	Related Special Register
&H2416	ED9022	BASIC task status	0: BASIC task in the rest state 1: BASIC task in the run, stopped or wait state	• The status of BASIC tasks 1 and 2 is stored as shown below. b15 to b8 b7 to b0 0 / 0 / 0/1 0/1 BASIC Task 2 / BASIC Task 1 • 0: BASIC task in the rest state (The BASIC program is not being run.) • 1: BASIC task in the run state (The BASIC program is being run.) BASIC task in the stopped state (None of the BASIC programs are being run.) BASIC task in the wait state (Waiting for execution of the PRINT command or other command to be completed or a run interrupt command was executed.) • If a BASIC task is started by a ZSTART command, do so when the designated task is on bit "0".	—	—
&H2432	ED9050	PCRD, PCWT timeout time	0 to 32767	• This sets the timeout time of the PCRD and PCWT commands. • Setting units: 100 ms • 0: Unlimited wait 1 to 32767: × 100 ms is the wait time	—	—

(1) Reading data in the special register

Data are read from the special register by the following methods.

• Using the special variable W@ of the BASIC program

• Using the FROM command in the sequence program

The method of designating the special register differs in the special variable W@ and the FROM command, as shown below.

(a) Reading of data by the special variable W@ in a BASIC program

With the special variable W@, the special register is designated by ED9000 to ED9127. The designation is in 1-word units.
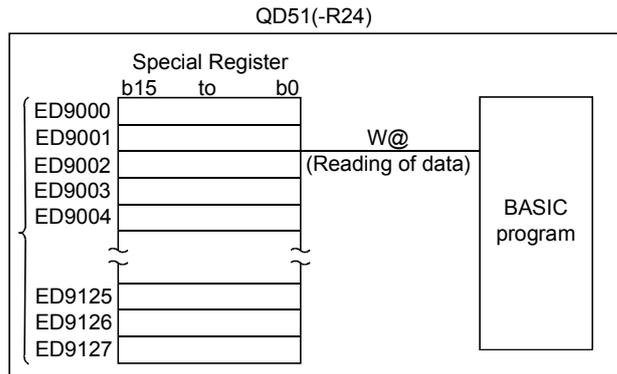
Program example

The following shows a program where data are read from the QD51 (-R24)'s special register ED9000.

100 A=W@ (ED, 9000)
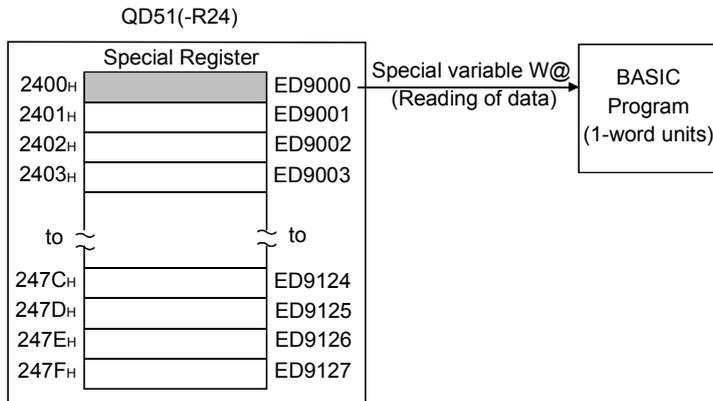
(b) Reading of data by the FROM command in a sequence program
The special register is designated using the FROM command by the address.
The designated address is 2400H to 247FH, in 1-word units.

QD51(-R24)

Special Register

	b15	to	b0	
2400H				ED9000
2401H				ED9001
2402H				ED9002
2403H				ED9003
2404H				ED9004
to				to
247DH				ED9125
247EH				ED9126
247FH				ED9127

FROM (Reading of data) → Sequence program
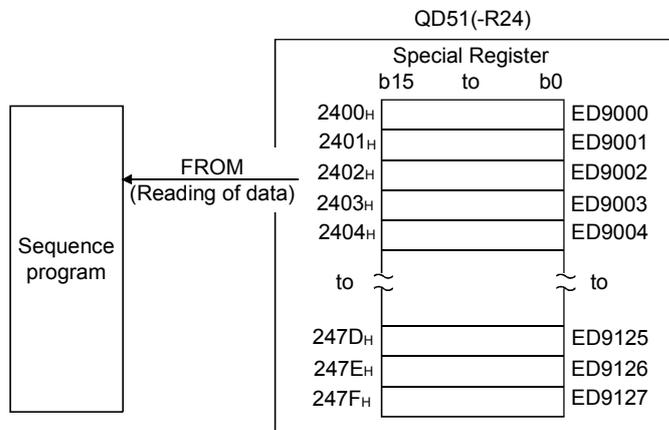
Program example
The following shows a program where a value in the special register address 2400H is read into D0 when the multitask execution start signal (X0B) is ON.
To read data from the special register to the programmable controller CPU, use the multitask execution start signal (X0B) as an interlock.

Read
Command   X0B
  ┤├────────┤├──────────────────────[FROM H0 H2400 D0 K1]─┤

Interlock that enables or disables FROM execution.

QD51(-R24)

Expansion Register

2400H		ED9000
2401H		ED9001
2402H		ED9002
2403H		ED9003
to		to
247CH		ED9124
247DH		ED9125
247EH		ED9126
247FH		ED9127

Sequence Program (1-word units) ← FROM (Reading of data)
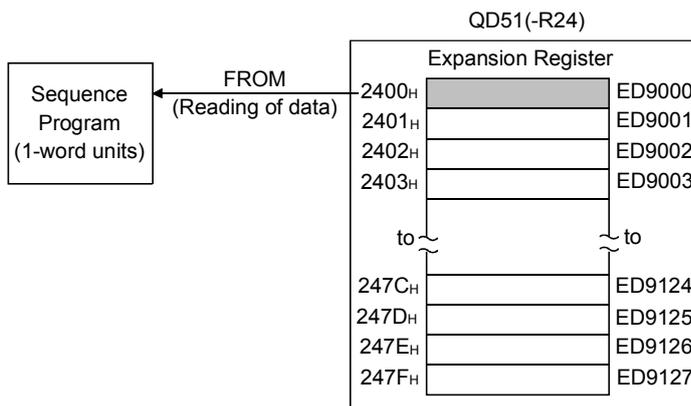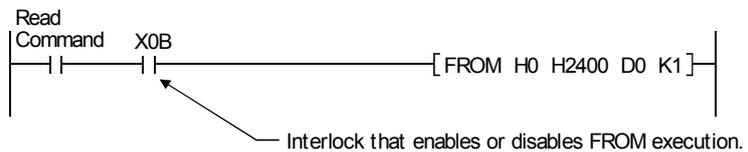
3.5.6 Expansion relay (EM0 to EM1023)

The expansion relay is a relay that can be used to send and receive ON/OFF data between tasks internally in the QD51 (-R24).

(1) Turning the expansion relay ON and OFF, and reading ON/OFF data

(a) The special variable B@ can be used in a BASIC program to turn the expansion relay ON and OFF and to read ON/OFF data.
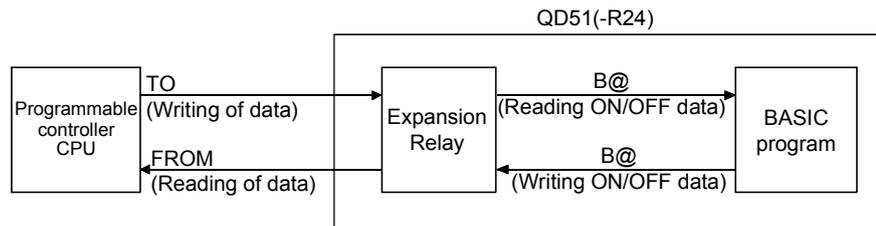
(b) During BASIC program execution by multitasking, either task can read the ON/OFF data, and this register can be used by either task through a BASIC program.
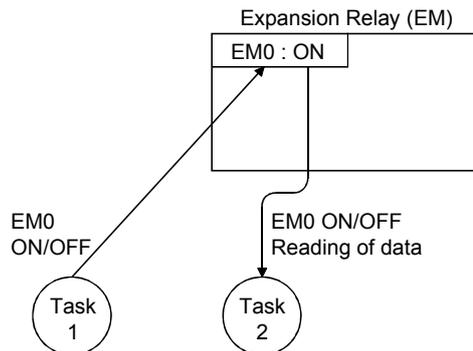
(2) Number of Expansion Relay Points

The QD51 (-R24) expansion relay has 1024 points.

REMARK

For details on special variable B@, refer to the Programming Manual (Commands).

(3) Writing/ Reading of Expansion Relay ON/OFF Data

ON/OFF data are written to or read from the expansion relay by the following methods.

• Using the special variable B@ of the BASIC program
• Using the FROM/TO command in the sequence program

The expansion relay designation method differs as follows in the special variable B@ and the FROM/TO command.

(a) Writing ON/OFF data and reading them using special variable B@ in a BASIC program

With special variable B@, the expansion relay is designated by EM0 to EM1023. Designation is in 1-bit units.
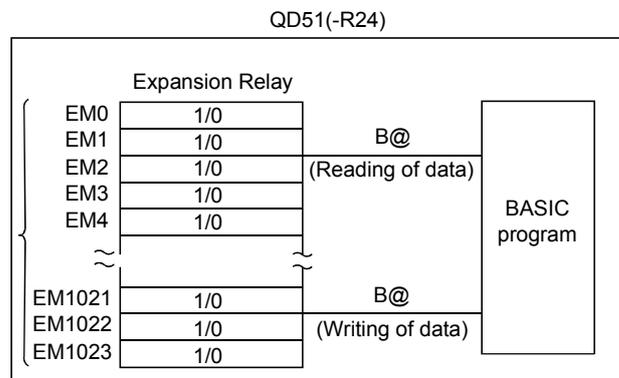
Program example

The following shows a program where "1" is written into the QD51 (-R24)'s expansion relay EM0.

150 B@ (EM, 0)=1
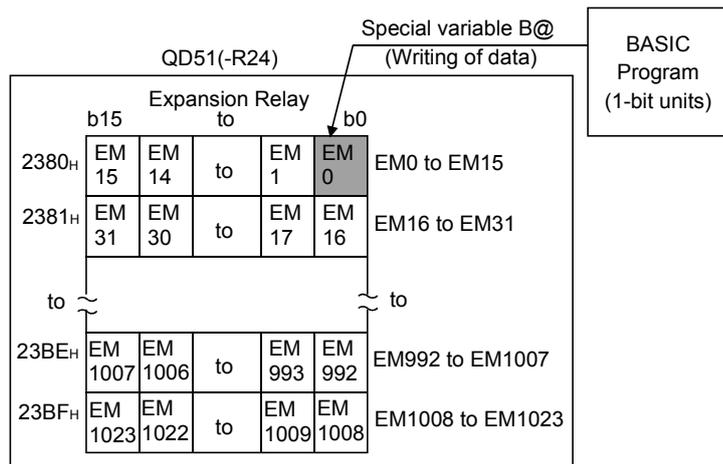
(b) In the FROM/TO command for writing or reading data in accordance with the FROM/TO command of a sequence program, an expansion relay is designated in the address. The designated address is in 1-word units, from 2380$_H$ to 23BF$_H$.
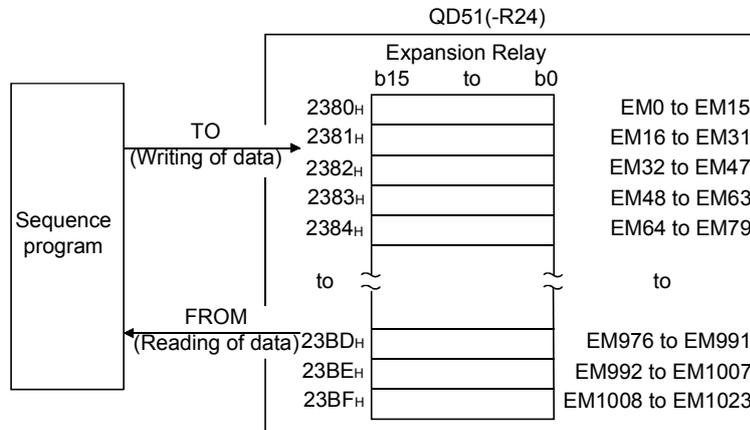
Program example

The following shows a program where "1" is written into the expansion relay address 2380$_H$ when the multitask execution start signal (X0B) is ON.

To write data from the programmable controller CPU to the expansion realy or to read data from the expansion relay, use the multitask execution start signal (X0B) as an interlock.
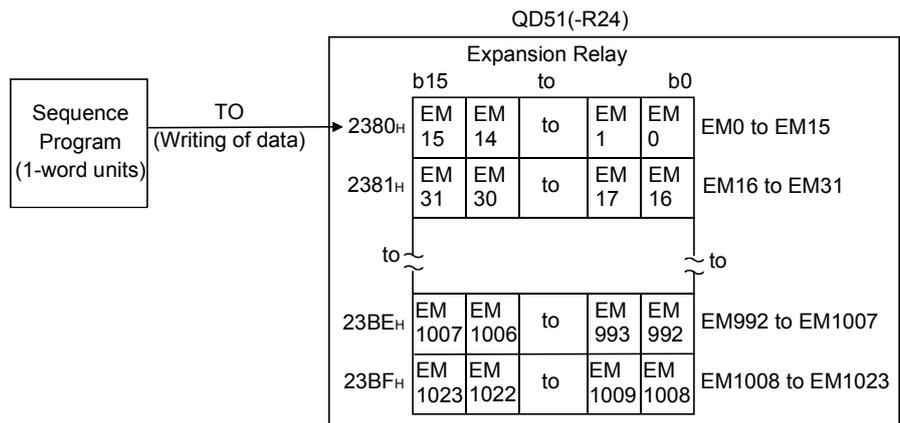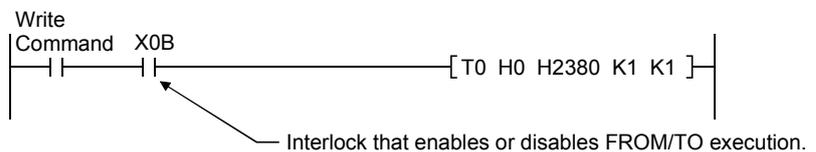
3.5.7 Special relay (EM9000 to EM9127)

The special relays are relays whose use is determined internally in the QD51 (-R24).

ON/OFF control is performed by the QD51 (-R24) OS, so in each BASIC program, read the ON/OFF status using the special variable B@ to use these relays.

| Buffer Memory Address as viewed from the CPU	Special Relay No.	Name	Content	Detailed Contents	Related Special Registers
&H2500, bit 0	EM9000	Error Registration Flag	ON : An error is registered. OFF: No error is registered.	Goes ON when one of the following occurs. • Warning Error • Error Message • System Down Error Message	ED9000 to ED9016
&H2501, bit 4	EM9020	Operation Flag	ON: Multitask execution in progress OFF: Task other than multitask	• Goes ON when the QD51 (-R24) is executing multitasks. • Goes OFF when the QD51 (-R24) is executing a task other than multitasks.	—

(1) Reading of Special Relay ON/OFF Data

ON/OFF data are read from the special relay by the following methods.
• Using the special variable B@ of the BASIC program
• Using the FROM command in the sequence program

The special relays designation method differs as shown below for the special variable B@ and the FROM command.

(a) Reading data by the special variable B@ in BASIC program
With the special variable B@, the special relays are designated by EM9000
to EM9127. Designation is in 1-bit units.
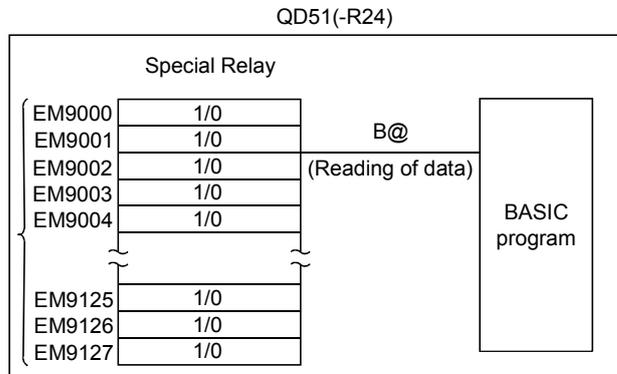
QD51(-R24)

Program example
The following shows a program where data are read from the QD51
(-R24)'s special relay EM9000.

150 A=B@ (EM, 9000)
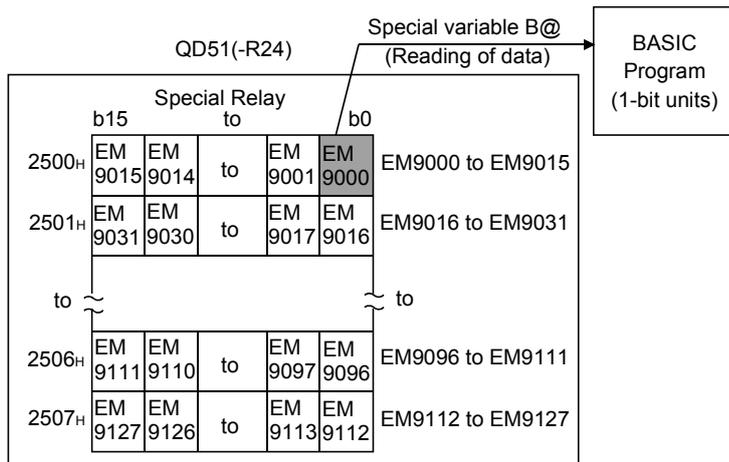
(b) Reading data by the FROM command in a sequence program

With the FROM command, special relays are designated by addresses.

Designation addresses are 2500$_H$ to 2507$_H$, in 1-word units.
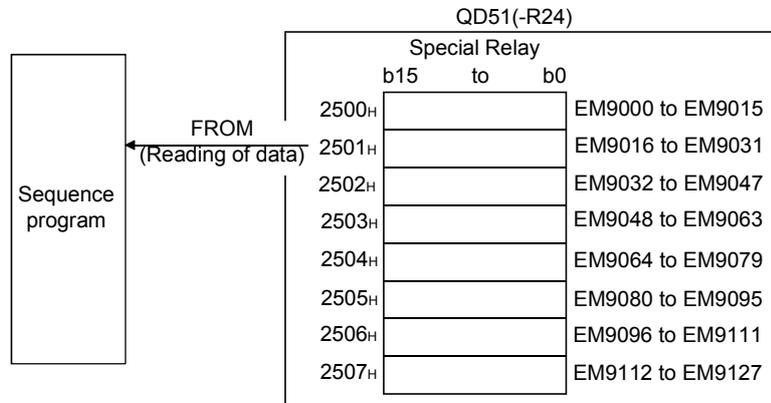
Program example

The following shows a program where a value in the special relay address 2500$_H$ is read into D0 when the multitask execution start signal (X0B) is ON.

To read data from the special relay to the programmable controller CPU, use the multitask execution start signal (X0B) as an interlock.
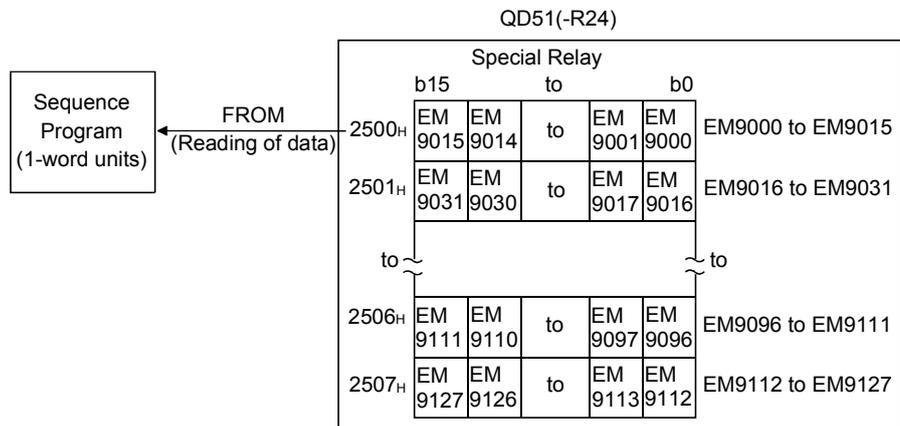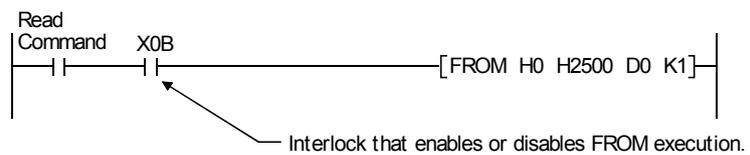
3.6 Receive Buffer

The receive buffer is a memory buffer used for sending and receiving data between the QD51 (-R24)'s RS-232 interface and RS-422/485 interface and an external device connected to them.
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	• 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$	• 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	• 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	• Stores the data received from the interface in the designated variable

Refer to the Programming Manual (Commands) for the data receive commands and the send/receive processing.
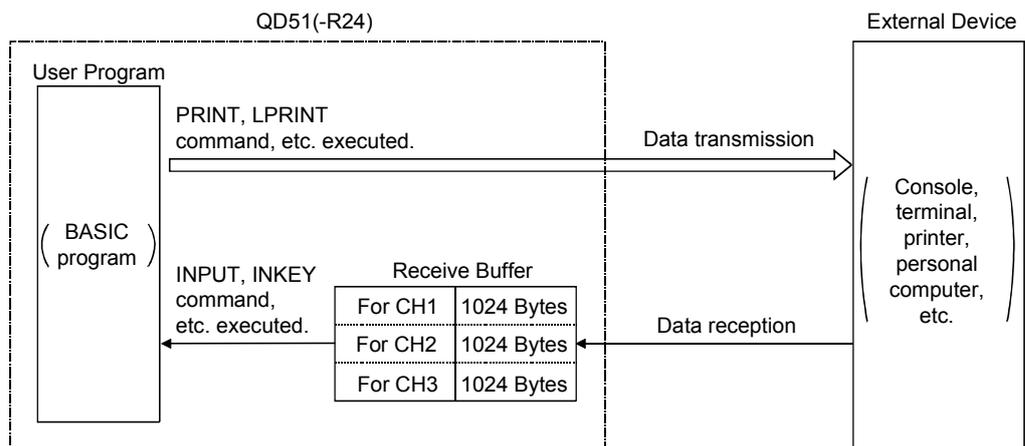
3.7 List of GX Developer Setting Items for the QD51 (-R24)

The following table shows a list of parameter setting items set using GX Developer.

Parameter Setting Item	Parameter Contents	Setting Contents		Explanation Section
I/O Allocation Setting	Allocates I/O in the QD51 (-R24), and enables the following switch settings.	Type		Section 5.6.1
		Model name		
		Number of points		
		Header X/Y		
		Detailed settings		
Switch Settings	Switch 1	Setting switch		Section 5.6.2
	Switch 2	Mode setting switch		
	Switch 3	Not used (Set 0000$_H$)		
	Switch 4			
	Switch 5			
Interrupt Pointer Settings	Carries out settings in order to execute an interrupt program.	CPU Side	Interrupt pointer header No.	Section 5.6.3
			Number of interrupt pointers	
		Intelligent module Side	Header I/O No.	
			Header SI No.	
Multiple CPU Settings	Set when using a Multiple CPU system.	Number of CPU's , etc.		Refer to QCPU User's Manual (Multiple CPU System)

3.8 List of Input/Output Signals From/To the Programmable controller CPU

The QD51 (-R24)'s input/output signals are explained.
The following I/O signal assignment is based on the case where the start I/O No. of the QD51 (-R24) is "0000" (installed to slot 0 of the main base unit).
Device X is an input signal from the QD51 (-R24) to a programmable controller CPU.
Device Y is an output signal from the programmable controller CPU to the QD51 (-R24).
The input/output signal list for the programmable controller CPU is shown in the following table.

Device No.	Signal Content	Reference Section	Device No.	Signal Content		Reference Section
X00	General input	Section 3.8.1 (1)	Y00	General Output		Section 3.8.2 (1)
X01			Y01			
X02			Y02			
X03			Y03			
X04			Y04			
X05			Y05			
X06			Y06			
X07			Y07			
X08			Y08			
X09			Y09			
X0A			Y0A			
X0B *1	Multitask execution start signal	Section 3.8.1 (2)	Y0B			
X0C	Multitask execution interrupt signal	Section 3.8.1 (3)	Y0C			
X0D *2	QD51 (-R24) system down signal	Section 3.8.1 (4)	Y0D			
X0E	QD51 (-R24) operating status signal	Section 3.8.1 (5)	Y0E			
X0F	Use prohibited	—	Y0F			
X10	General input	Section 3.8.1 (1)	Y10			
X11			Y11	General output/ Start Task No. Designation signal	BASIC Task 1	Section 3.8.2 (2)
X12			Y12		BASIC Task 2	
X13			Y13	General Output		Section 3.8.2 (1)
X14			Y14			
X15			Y15			
X16			Y16			
X17			Y17			
X18			Y18			
X19			Y19	Task Start Signal		Section 3.8.2 (3)
X1A			Y1A	Reset Request Signal		Section 3.8.2 (4)
X1B			Y1B	Use Prohibited		—
X1C			Y1C			
X1D			Y1D			
X1E			Y1E			
X1F	Use prohibited	—	Y1F			

*1 The multitask execution start signal (X0B) is a signal that announces whether access to the QD51 (-R24) by the programmable controller CPU is possible or not.   Use it as the system program interlock signal. (It goes ON when a BASIC program is run.)

*2 If the QD51 (-R24) system down signal (X0D) goes ON, start up the programmable controller CPU again. (Power supply reset, CPU module, QD51 reset)

Important

Do not output (turn ON) the "Use Prohibited" signal, which is one of the input/output signals related to the programmable controller CPU.

If an output is activated for the "Use Prohibited" signal, there is danger that the programmable controller system could malfunction.

3.8.1 Input (X) detailed explanation

Here, the input (X) for the QD51 (-R24) is explained.
The input No. (Xn) used in this section is the No. as viewed from the programmable controller CPU.

(1)   General Input (X00 to X0A, X10 to X1E: 26 points)
The general input can be used freely by the user and can be used to send ON/OFF data from the QD51 (-R24) to the programmable controller CPU.
Turning the general input ON and OFF is accomplished by the special variable B@.
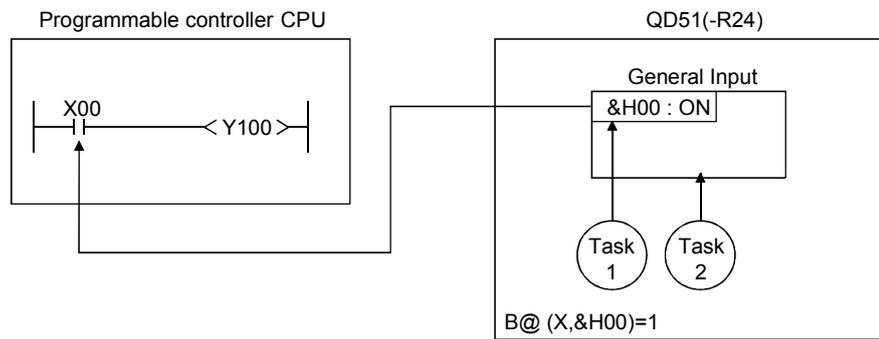
(2)   Multitask Execution Start Signal [X0B]

    (a)   When the QD51 (-R24)'s mode setting switch 1 ＊1 is set in the "Run Mode
        (0 or 1)", this signal is enabled (it carries out ON/OFF control).
        (If the QD51 (-R24) is in a mode other than the Run mode, it remains OFF
        and does not change.)

    (b)   If the QD51 (-R24) is in the Run mode, after the power is turned on or after
        a Reset, if multitask execution is enabled, this signal goes ON.
        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 programmable controller CPU will stop its operation.)
        ＊2

        If data D0 is written from the programmable controller CPU to the QD51 (-
        R24)'s buffer memory address 0.

FROM/TO command execution enable/disable interlock
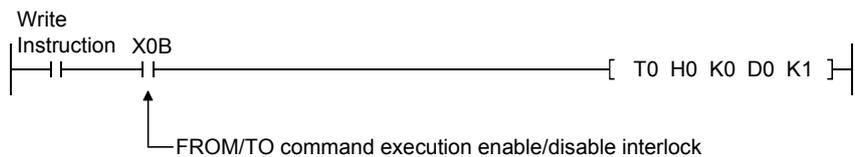
REMARK

＊1   Refer to Section 5.6.2 for details of the mode setting switch.

＊2   When the QD51(-R24) is reset by the reset request signal (Y1A), execution of
    the FROM/TO command within the access enabled time after the Y reset, will
    not cause an error.
    (Resetting the QD51(-R24) turns OFF the multitask execution start signal
    (X0B).)
    The access enabled time after Y reset can be set in the intelligent function
    module switch setting of the GX Developer. (Refer to Section 5.6.2.)

    (c) In the QD51 (-R24)'s Run mode, it is OFF in the cases shown below.

       1) When the reset request signal (Y1A) of the QD51 (-R24) is turned on and reset is performed.
(It goes ON again if the QD51 (-R24)'s reset processing is completed and the multitask execution enabled state is restored.)

       2) When a BASIC error is detected.
(Multitask execution interrupt signal (X0C) goes ON.)
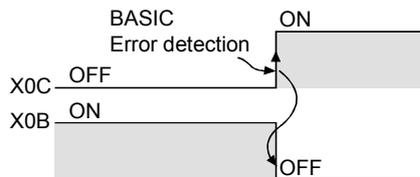
       3) When the QD51 (-R24) system is down.
(QD51 (-R24) system down signal (X0D) goes ON.)
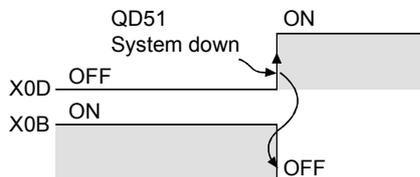
(3) Multitask Execution Interrupt Signal [X0C]

    (a) When the QD51 (-R24)'s mode setting switch is set in the "Run Mode (0 or 1)", this signal is enabled (it carries out ON/OFF control).
(If the QD51 (-R24) is in a mode other than the Run mode, it remains OFF and does not change.)

    (b) This signal goes ON when an error is detected in a BASIC program during multitask execution.
(At this time, the ERR. LED turns on, the special register (number: ED9000 or later, buffer memory address: 2400H or later) stores the error line number and error code*1, and the module is placed in the system mode. Correct the BASIC program where the error occurred.)

    (c) If the multitask execution interrupt signal (X0C) goes ON, the multitask execution start signal (X0B) goes OFF.
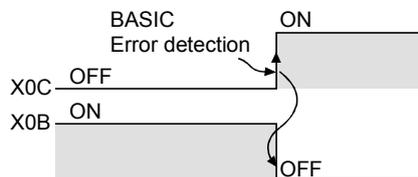
REMARK

*1 For the error code, refer to Appendix 4.4 of the Programming Manual (Commands).

(4) QD51 (-R24) System Down Signal [X0D]

    (a) When the QD51 (-R24)'s mode setting switch is set in the "Run Mode (0 or 1)", this signal is enabled (it carries out ON/OFF control).
(If the QD51 (-R24) is in a mode other than the Run mode, it remains OFF and does not change.)

    (b) This signal goes ON when the system goes down during multitask execution and an error is detected. At this time, the ERR. LED turns on and the special register (number: ED9000 or later, buffer memory address: 2400_H or later) stores the error code*1.

    (c) If the QD51 (-R24) system down signal (X0D) goes ON, the multitask execution start signal (X0B) will go OFF.
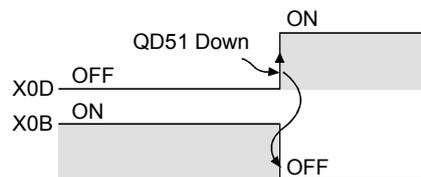
REMARK

*1 For the error code, refer to Appendix 4.4 of the Programming Manual (Commands).

(5) QD51 (-R24) Operating Status Signal [X0E]

    (a) This shows the QD51 (-R24)'s operating state.

    (b) It goes OFF during Power OFF or during Reset processing.

3.8.2 Output (Y) detailed explanation

Here, QD51 (-R24) outputs (Y) are explained.
The output No. (Yn) used in this section is the number as viewed from the programmable controller CPU.

(1) General Output (Y00 to Y10, Y13 to Y18: 23 points)

The general output signals can be used to send ON/OFF data from the programmable controller CPU to the QD51 (-R24). The programmable controller CPU can turn ON and OFF by a sequence programs or GX Developer's device test function, etc.

In the QD51 (-R24), reading of general output ON/OFF data is accomplished by the special variable B@.
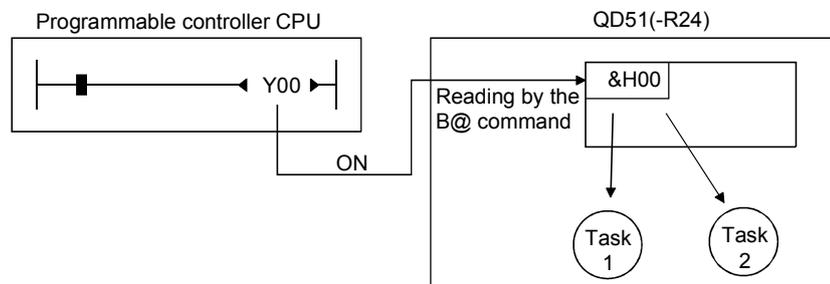
(2) Start Task No. Designation Signal (Y11, Y12: 2 points)

(a) These are signals which designate tasks to start by an interrupt from the programmable controller CPU and there is one point for each task.

(b) Tasks with "IT (start by an interrupt from the programmable controller CPU)" set in the multitask setting are subject to the start task No. designation signal (Y11/Y12).
The device No. for a task for which the multitask setting is not set on "IT" is treated the same as a general output.

Device No.	Multitask Setting		
	Start by an interrupt from the programmable controller CPU		Other than at left
Y11	Start task No. setting	Task No. 1	General Output
Y12		Task No. 2	(Impossible to start each task)

(c) Turn the start task No. designation signal (Y11/Y12) corresponding to the task you are starting ON. At the time the task start signal (Y19) rises, the start task No. designation signal (Y11/Y12) thus designated to be turned ON will be started.
However, if a task that is currently being executed is designated, it is disregarded.

(d) To turn OFF the signal designated by the start task No. that was turned ON, do so only after confirming that the designate task has been executed.
To check if the designated task has started, use the method of turning the general input (X) allocated to the Reset (OFF) instruction by the BASIC program as shown in the example on the next page.

┌─── Example ───

This example shows the procedure in the case where a task is started, and for turning the start task No. designated signal(Y11)/task start signal(Y19) OFF.

[Conditions]
1) Start Task No. ······ 1
2) Start task No. designation signal / Task start signal reset instruction ······ X11

Task start instruction

Task start completed

Input/Output No. as seen from the programmable controller CPU: X/Y00 to X/Y1F
Value designated by B@: X/Y&H00 to X/Y&H1F

[Procedure]
1) Set the task No. · · · · · Turn Y11 ON by a sequence program.
2) Start the task · · · · · · · Turn Y19 ON by a sequence program.
3) Turn the Start Task No. designation signal(Y11) / Task Start Signal(Y19) OFF
· · · · · · Turn X&H11 ON by a BASIC program.

└───

(3) Task Start Signal (Y19)

(a) This is a signal that starts the task designated by the start task No. designation signal (Y11/Y12). The task start signal (Y19) cannot be used as a general output even in cases where there is no task that is started by an interrupt from the sequencer CPU. Also, if the task is compiled BASIC, it cannot be started by an interrupt from the programmable controller CPU.

(b) The task start signal (Y19) is enabled only while the multitask execution start signal (X0B) is ON.

(c) The designated task ∗1 is started when the task start signal (Y19) rises (OFF → ON). (Even if the start task No. designation signal (Y11/Y12) goes ON while the task start signal (Y19) is ON, it is disregarded.)

```
Example
The example shows a program in the case where task 1 is started from the programmable controller
CPU.
[Conditions]
"IT (started by an interrupt from the programmable controller CPU)" is set with only task 1 set for
multitasking.

                                          ON
   Task start signal    Y19 OFF

                              ON              ON
   Task 1 start signal  Y11 OFF
                                              ∗2

                         ON ∗1
   General output       Y12 OFF          Task 1 starts
   treatment

                                     END

                              Task 1 executed

1) ∗1  Since it is treated as a general output, starting of the task is not carried out.
2) ∗2  Since the task start signal is not changing from OFF → ON, starting of task 1 is not carried out.
[Sequence Program]

                 Task 1 setting designation
                 ─┤ ├─                    ─<Y11>
                 Task start command
                 ─┤ ├─                    ─<Y19>
```

(d) In the case of the task designated by the start task No. designation signal (Y11/Y12) being in the process of starting, task start is disregarded. (Operation of the currently executing task is continued.)

(e) If the exclusive command YCHECK is used, the task's starting time can be shortened. Refer to Chapter 6 concerning the YCHECK command.

REMARK

*1   This is a task that corresponds to the device No. of the device designated by
     Y11, Y12 that is turned ON. However, a device No. that does not have "IT
     (started by an interrupt from the programmable controller CPU)" set for the
     multitask setting is disregarded.

(4) Reset Request Signal (Y1A)

   (a)   This is a signal that resets the QD51 (-R24) module only.
         Reset is executed after the access enabled time following the Y reset set by
         the GX Developer intelligent function module switch.

Reset after access enabled time (200/2000 ms)
following the Y reset set by GX Developer.
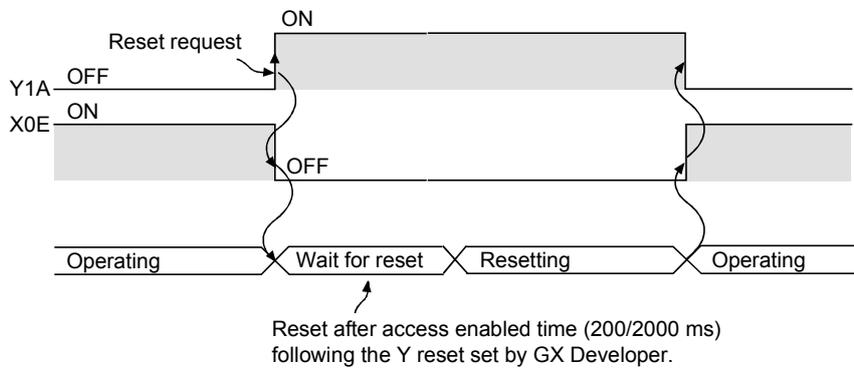
POINT

During reset, set up an interlock with the QD51 (-R24) operating status signal (X0E)
so that there will be no access from a sequence program.

3.9 Relationship between the QD51 (-R24) Start Condition and the Flash ROM Execution Program Area

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 BASIC program's storage state in the program area differs as follows depending on the set start conditions.

Start Condition		Area where the BASIC program to be run is stored	BASIC Program Read Timing	Reference Section
Multitask Setting	Symbol			
Starting by turning the power on or by reset	START	• Flash ROM run program area. (Designated BASIC task No. area)	• In the System mode, when the power is turned on or during a Rest. • In the Run / Multitask debug mode, during multitask execution.	Section 3.11.2
Starting by an interrupt from the programmable controller CPU	IT	• Flash ROM run program area. (Designated BASIC task No. area)		Section 3.11.3
Starting by a start request from another task	BOOT	• Flash ROM run program area. (Designated BASIC task No. area)		Section 3.11.4
	ON	• User FD or HD file area for the peripheral device. (Save the BASIC program as a file.)	• When run by the ZSTART command. [The BASIC program with the file name designated by the ZSTART command is read.]	Section 3.11.4
No multitask setting	OFF	—	• Since there is no multitask setting, reading of a BASIC program is not done.	—

POINT

The difference between the start condition "BOOT" and "ON" is as follows.
• BOOT   : The program to be run is stored in the designated BASIC task No. area in the main memory's program area.
• ON   : Only the memory capacity of the designated BASIC task No. area in flash ROM is set in the program area of the main memory. The program to be run is not stored in the BASIC task No. area.

REMARK

Refer to the AD51H-BASIC programming manual (Debug and Compile) concerning the system commands MSAVE and SET.

3.10 QD51 (-R24) Operation when the Power is Turned On

The QD51 (-R24) operates in the mode set by the intelligent function module switch
[Mode Setting] in GX Developer when the power is turned on.
Operation differs depending on the mode.

1) Programming Mode

2) Multitask Debug Mode

3) Run Mode

Here the operation of the QD51 (-R24) when the power is turned on is explained for
each mode.

3.10.1 In the case of the programming mode

This section explains the case in which the QD51 (-R24) is set to the Programming
mode by GX Developer
When it is turned ON or reset, and after the program to be run is read from the flash
ROM to the QD51 (-R24)'s program area (Refer to Section 3.5.1), the mode is
switched to the System mode.
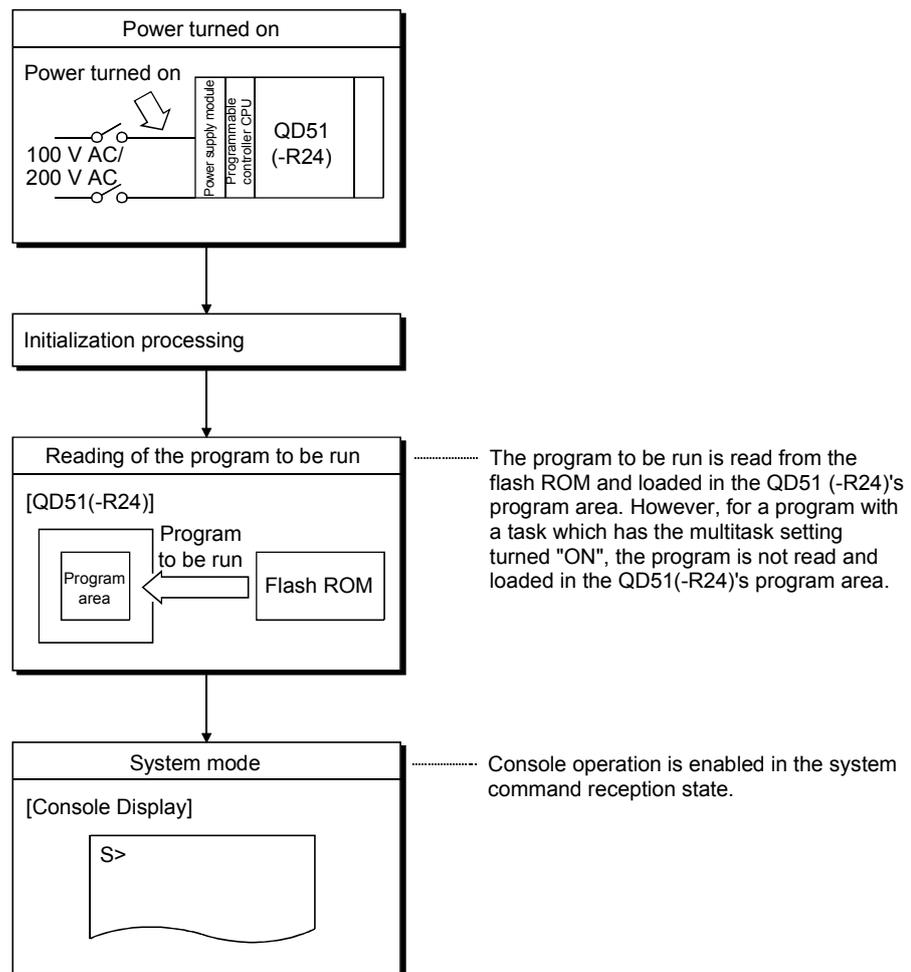
3.10.2 In the case of the multitask debug mode

This section explains the case in which the QD51 (-R24) is set to the Multitask debug mode by GX Developer

When it is turned ON or reset, and after the program to be run is read from the flash ROM to the QD51 (-R24)'s program area (Refer to Section 3.5.1), the mode is switched to the Multitask debug mode.

The program to be run is read from the flash ROM and loaded in the QD51 (-R24)'s 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 QD51(-R24)'s program area.

It is possible to carry out multitask debugging by the debugger while in the Debug command reception state. Run the program read from the Flash ROM in accordance with the multitask setting contents.
- Start by turning on the power or by a Reset.
- Start by an interrupt from the programmable controller CPU
- Start by a start request from another BASIC program.

3.10.3 In the case of the run mode

This section explains the case in which the QD51 (-R24) is set to the Run mode by GX Developer
When it is turned ON or reset, and after the program to be run is read from the flash ROM to the QD51 (-R24)'s program area (Refer to Section 3.5.1), the multitask processing is executed.
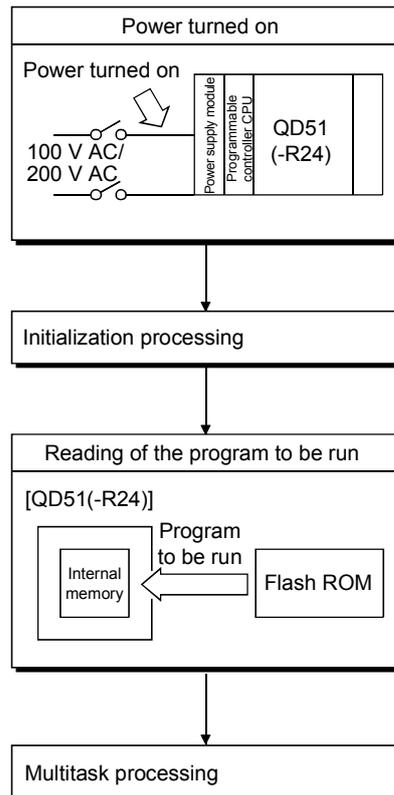
The program to be run is read from the flash ROM and loaded in the QD51 (-R24)'s program area. However, for a program with a task which has the multitask setting turned "ON", the program is not read and loaded in the QD51 (-R24)'s program area.

Run the program read from the Flash ROM in accordance with the multitask setting contents.
• Start by turning on the power or by a Reset.
• Start by an interrupt from the programmable controller CPU
• Start by a start request from another BASIC program.

3.11 Running a Task (a created program)

Here, running a task when the QD51 (-R24) is in the Run Mode and in the Multitask Debug Mode is explained.

3.11.1 Conditions for permitting running of a task and start conditions

In the QD51 (-R24), a task is permitted to be run when it satisfies all the conditions shown below (1). (Actual starting of a task depends on the start conditions designated in the multitask setting.)

(1)   **Task Run Permission Conditions**
Task start conditions are shown below (a), (b).

(a)   The start conditions (Refer to Section 3.9) are set in the multitask setting.

(b)   The GX Developer intelligent function module switch settings (Refer to Section 5.6) is set on "Run Mode (0, 1)" or "Multitask Debug Mode (2, 3)".
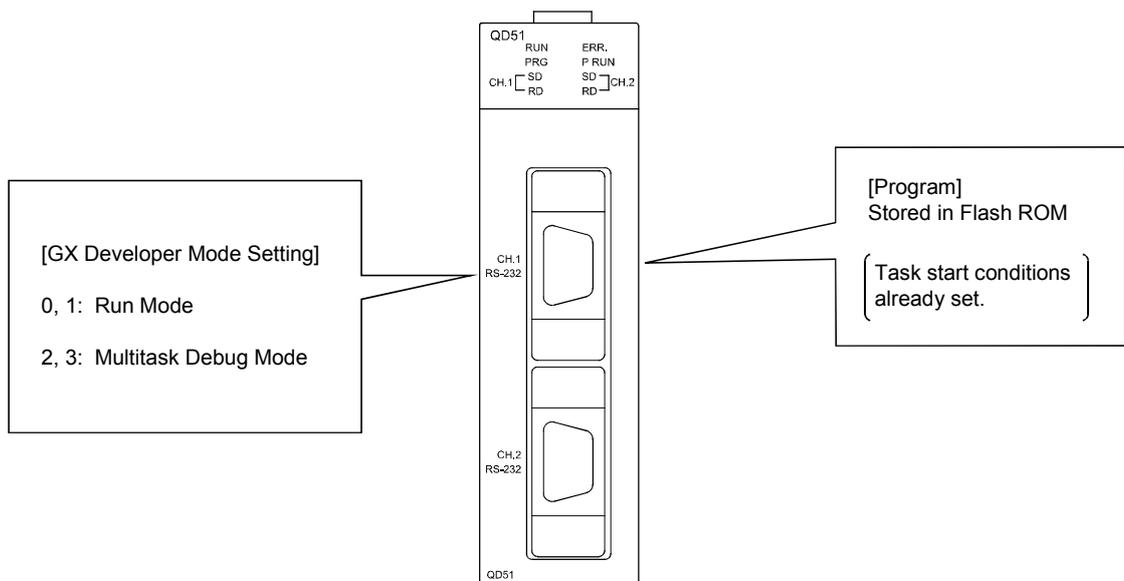
(2)   **Task Start Conditions**
Each task is run in accordance with the set start conditions.
The start conditions that can be set by the QD51 (-R24) are following three types.

Start Condition	Symbol	Reference Section
Starting by turning the power on or by reset	START	Section 3.11.2
Starting by an interrupt from the programmable controller CPU	IT	Section 3.11.3
Starting by a start request from another task	BOOT, ON	Section 3.11.4

3.11.2 Starting by turning the power on or by reset (start condition: START)

This is the case where the start condition in the multitask setting is set on "START". Starting of a task is done if the "Run Permission Conditions" listed in Section 3.11.1 are established.

(1) Processing when powered on or reset

When the power is turned on or there is a Reset, after the program to be run which is stored in the Flash ROM is read and loaded to the QD51 (-R24)'s program area, the designated task is run.

Flash ROM

BASIC Task No. Area	Start Condition		QD51 (-R24) Program Area
1	START	BASIC Program →	Task No. 1 Area
2	START	→	Task No. 2 Area
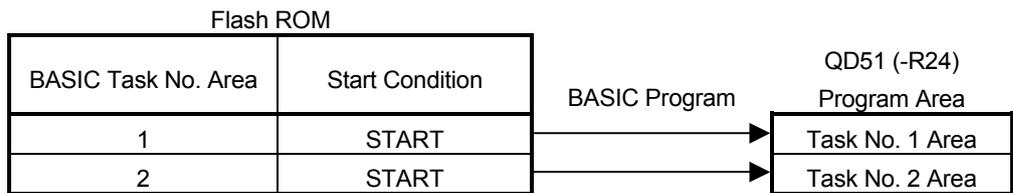
(2) Running a Task

A task that is started by the power being turned on or by a Reset is run as follows in accordance with the BASIC program that is created.

1) Run once only.
2) Run all the time.

(a) If it is run once only

A program is run once only if the "END" command is written at the end of the BASIC program.

The task starts when the power is turned on or when there is a Reset, and operation stops at the point when the END command is run.

```
   Power On              Reset

        └──────┬──────┘
               ↓
       ┌──────────────┐  ┐
       │   BASIC      │  │
       │   Program    │  │  Task with "START" set
       │              │  │  in the start condition.
       └──────────────┘  │
               ↓         │
          ┌───────┐      ┘
          │  END  │···········  BASIC program stops.
          └───────┘
```
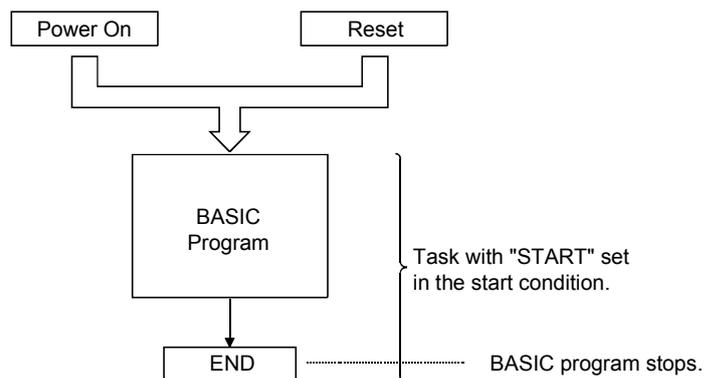
(b) If the program runs all the time

The program runs all the time if an END command is not used at the end of the BASIC program, but instead a "GOTO" command is used to cause it to jump to a specific line.

The task starts when the power is turned on or when there is a Reset, then runs until the GOTO command, then since it jumps to the specified line, it runs repeatedly between the specified line and the GOTO command.
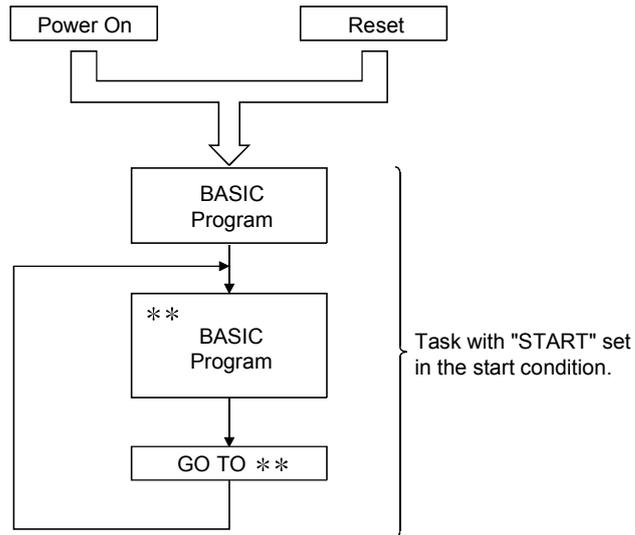
3.11.3 Starting by an interrupt from the programmable controller CPU (Start Condition: IT)

This is the case where the start condition in the multitask settings is set on "IT".
Starting of a task is done if the "Run Permission Conditions" listed in Section 3.11.1 are established and when there is a start request from the programmable controller CPU.

A start request from the programmable controller CPU is executed by:
1)   Start task No. designation signal (Y11/Y12).
2)   Task start signal (Y19).
(For details concerning the start task No. designation signal (Y11/Y12) and the task start signal (Y19), refer to Section 3.8.2.)

(1)   Processing when powered on or reset
When the power is turned on or there is a reset, the program to be run which is stored in Flash ROM is read and loaded into the QD51 (-R24)'s internal memory. The loaded BASIC program is then run when there is a start request from the programmable controller CPU.

(2)   Running a task

(a)   When the power is turned on or there is a reset, the Basic program that is read from the Flash ROM is started by a task start signal (Y19) from the programmable controller CPU to the QD51 (-R24).
When the task start signal (Y19) rises, the task with the start task No. designation signal (Y11/Y12) ON is started.

```
┌──────────────────┐
│ Start request from│
│ programmable     │
│ controller CPU   │
└──────────────────┘
         ⇩
┌──────────────────┐ ┐
│ BASIC Program    │ │
└──────────────────┘ ├─ Task with IT set in the start condition.
┌──────────────────┐ │
│      END         │ ┘···· BASIC program stops.
└──────────────────┘
```
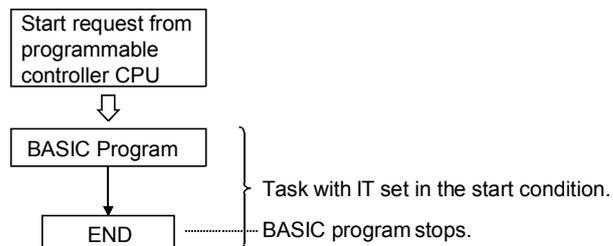
(b)   For a task that is started by an interrupt from the programmable controller CPU, write it so that an "END" command is run at the end of the BASIC program.
If an "END" command is run, the interrupt from the programmable controller CPU is reset and it is made possible for the next interrupt to be received.

POINT
When using compiled BASIC, "IT" cannot be set in the start condition.

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

This is the case where the start condition in the multitask settings is set to "BOOT" or "ON".
A task is started when the "Run Permission Conditions" are satisfied and when the start is requested by the ZSTART command from another BASIC program.
Refer to Section 3.11.1 for the Run Permission Conditions.
(For details of the ZSTART command, refer to the Programming Manual (Commands).)

(1)   Processing when powered on or reset

(a)   When the start condition is "BOOT"
When the module is turned ON or reset, the program to be run is loaded from the flash ROM to the internal memory of the QD51 (-R24).
The loaded BASIC program runs when a start is requested from another task.

(b)   When the start condition is "ON"
When the module is turned ON or reset, the program to be run is not loaded from the flash ROM to the internal memory of the QD51 (-R24).
When a start is requested from another task, the designated BASIC program is read from the file area of the user FD/HD to run.

(2) Running 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

[If the start condition is "BOOT"]

```
+------------------+
| Start request from|
| another BASIC program|
+------------------+
         ⇩
+------------------+
|  BASIC Program   |
+------------------+
         |
         ↓
     +-------+
     |  END  | ·······  • BASIC
     +-------+            Program
                         stops.
                       • Start request
                         is reset.
```

[If the start condition is "ON"]

```
+------------------+
| Start request from|
| another BASIC program|
+------------------+
         ⇩
+------------------+
| BASIC program is |
| read from the user|
| FD/HD.           |
+------------------+
         |
         ↓
+------------------+
|  BASIC Program   |
+------------------+
         |
         ↓
     +-------+
     |  END  | ·······  • BASIC Program
     +-------+            stops.
                       • Start request is
                         reset.
```
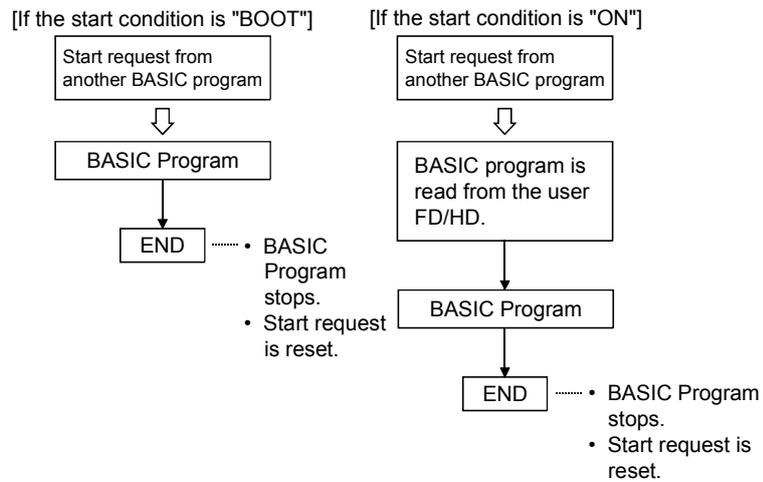
    (b) For a task that is started by a start request from another BASIC program, write it so that an "END" command is run at the end of the BASIC program. If an "END" command is run, the start request is reset and it is made possible for the next interrupt to be received.

3.12 Running a BASIC Program by Multitask Processing

Here, the method of running a QD51 (-R24) BASIC program, etc. is explained.

3.12.1 Multitask processing

The QD51 (-R24) us a unit which operates by BASIC programs created by the user.

The QD51 (-R24) has one main CPU (central processing unit).
Multitasking accomplishes parallel operations by successively switching the execution between multiple BASIC programs (up to 2 tasks).

```
┌── Example ──────────────────────────────────────────┐
│                                                      │
│              [Processing in the case of two tasks]   │
│                                                      │
│                              Task 1      Task 2      │
│          Shows the order of                          │
│          main CPU processing.                        │
│                                 1)          2)       │
│                                                      │
│                                 3)          4)       │
│                                                      │
│                                 5)          6)       │
│                                                      │
└──────────────────────────────────────────────────────┘
```
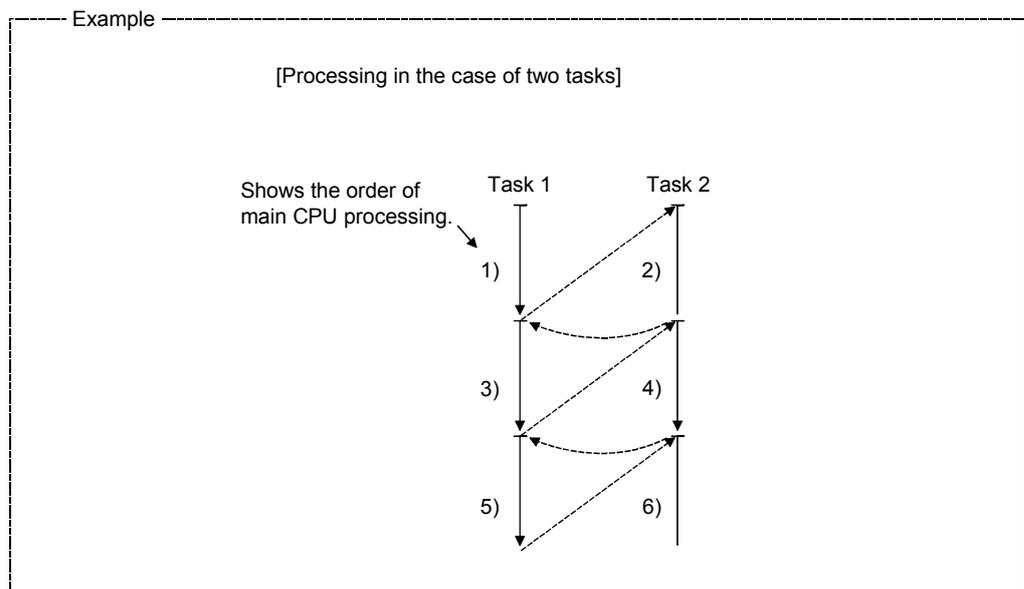
In QD51 (-R24) multitask processing, switching of BASIC program execution is done when the conditions shown below occur.

- When "Task schedule time (50 ms)" has passed.
- When a data input/output (data send/receive) command is run with respect to a screen, keyboard, disk, peripheral device or external device.
- When a command is run which carries out control of (interrupts, stops, ends) BASIC program execution.
- When a command is run which carries out multitask control (synchronizes execution, starts or changes the priority, etc.).
- When a program execution interrupt state is canceled (input/output is completed, the specified time has passed, etc.)

REMARK

Refer to the Programming Manual (Commands) concerning details of multitask processing and commands which switch the execution of programs.

3.13 BASIC Program Priority Order (Priority)

In the QD51 (-R24), a priority order (priority) can be assigned to BASIC programs that are created.
When BASIC programs are started, all BASIC programs are set at the same priority order.
Therefore, the BASIC programs created by the user are run equally.

The priority order can be changed by the ZURGENCY command.
If the priority order is changed, as long as a program execution switch command is not run, only the BASIC program with the highest priority is run without any scheduling being done by the OS.
(BASIC programs with a low priority are not run.)

3.13.1 Synchronizing execution between BASIC programs (Event control)

In multitask processing, each BASIC program is process independently.
for this reason, if data are shared between BASIC programs, it is necessary that the receiving side read the data after confirming that the sending side has sent the data. Event control is used to synchronize the execution of the data sending side BASIC program and the data receiving side BASIC program.
In event control, the BASIC program that is on the data receiving side is set in the signal wait (WAIT) state then after the sending side finishes sending the data, it gives a signal (event) and the receiving side BASIC program reads the data.
For this reason, the data sent by the sending side can be read precisely by the receiving side BASIC program.

In event control, the following commands are executed.
- DEF ZEVENT   : Definition of the event.
- ZEVENT ENABLE : Enabling generation of the event.
- ZWAIT EVENT   : Waiting for the event.
- ZSIGNAL   : Generating the event.
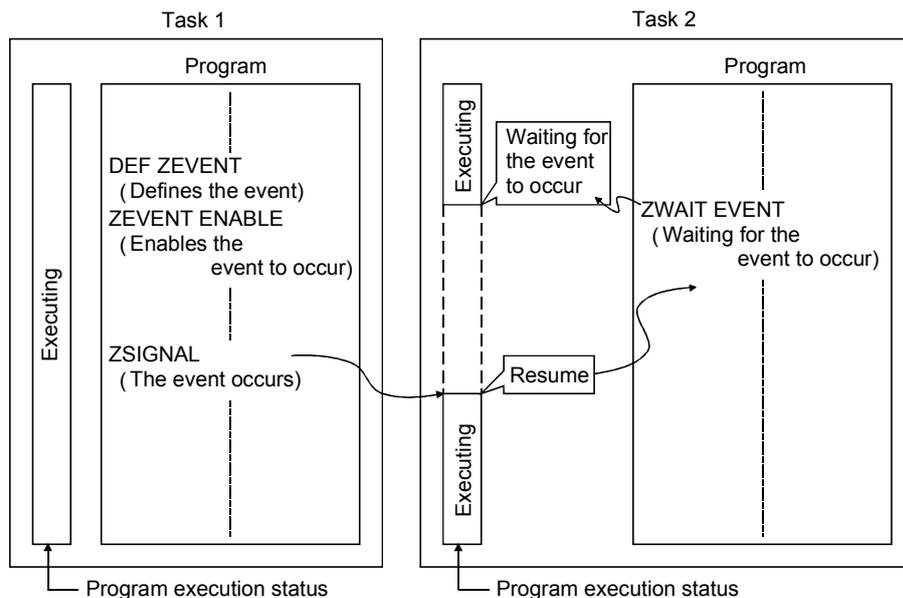
REMARK

Refer to the Programming Manual (Commands) concerning details on event control and the commands used.

3.13.2 Sharing equipment (Resources) by multitasking

If use of resources is shared by BASIC programs, the same resources cannot be used by multiple BASIC programs at the same time.

Exclusive control of resources is a way of making it impossible for another BASIC program to use resources while one BASIC program is using them.
This exclusive control of resources must in the end be decided and implemented by fellow programmers.
If this problem is not decided by the fellow program creators and therefore not provided for, exclusive control of resources cannot be implemented.

Exclusive control of resources is accomplished by the following commands.
• ZRESERVE : Requests exclusive control of a resource.
• ZRELEASE : Ends exclusive control of a resource.
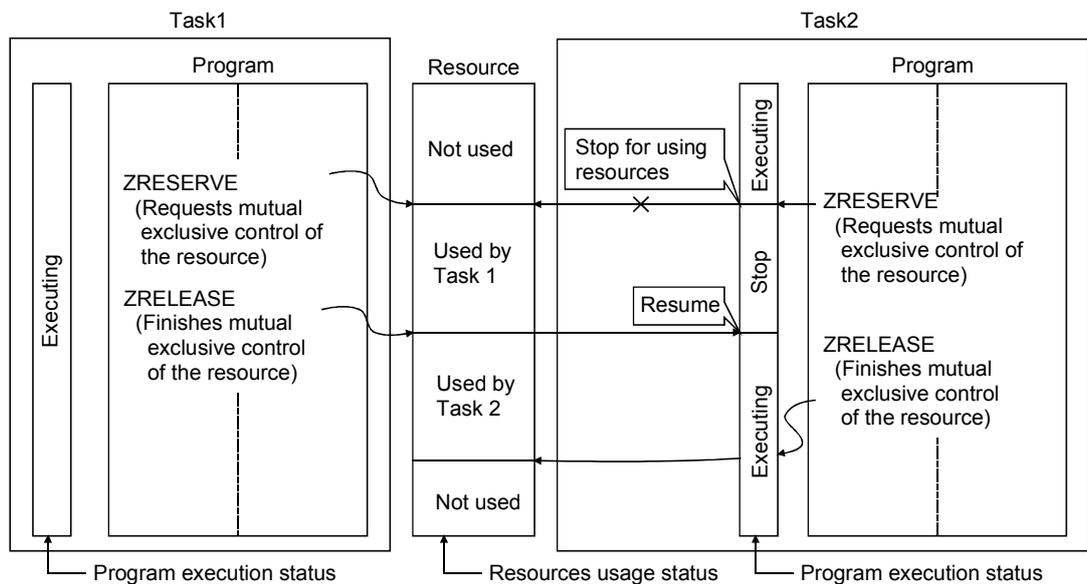
REMARK

Resources are said to be all the hardware and software that can be used by a BASIC program. For example, the screen, printer, files, disks or memory, etc.

4 ABOUT QD51 (-R24) FUNCTIONS

Here, the functions possessed by the QD51 (-R24) are explained.

4.1 List of Functions

A list of QD51 (-R24) functions is shown below.

Item		Content	Reference
Functions by BASIC programs	Sub-CPU Function	Complex numerical calculations and function calculations which would lengthen the programmable controller CPU' s scan time can be calculated by a BASIC program. • Collection, analysis and correction of measurement data • Function calculations such as Sine, Log and square roots, etc.	Refer to the Programming Manual (Commands).
	Monitor Display Function	This displays the operating status (production conditions, running status, breakdown contents, etc.) in a console or terminal connected to the QD51 (-R24).	Section 4.2
	Key Input Function	This carries out input of production schedules, production quantity, operations, setting data, etc. from the keyboard of a console or terminal connected to the QD51 (-R24).	Section 4.2
	Printer Function	This prints out production plans, results, daily reports, breakdown contents, planning data, inspection results, test results charts, etc with a printer connected to the QD51 (-R24).	Section 4.3
	Data Input Function	This enables input of data from a bar code reader or magnetic card reader, etc. that is connected to the QD51 (-R24). (Since sending and receiving can be done in a free format through a BASIC program, it is possible for the QD51 (-R24) to match the protocols of the opposite device.) • Input of production lot No., product name, quantity, etc. • Collection of measured values, test data.	—
	External Device Connection Function	This connects with a computer, etc. to the RS-232 or RS-422/485 interface of the QD51 (-R24) and sends and receives data through a BASIC program.	Section 4.4
	Clock Function	This writes and reads programmable controller CPU clock data (year, month, day, hour, minute, second, day of week) for programmable controller CPU's with the clock function.	Section 4.7
Online Programming Function		This creates executes and corrects BASIC programs through system commands by connecting a console to the QD51 (-R24). It also records programs run by system commands in Flash ROM and reads them.	Refer to the AD51H-BASIC Programming Manual (Debug and Compile).
Multitask Debugging Function		This carries out debugging while running BASIC programs by multitasking, by connecting a console and debugger to the QD51 (-R24).	

4.2 Communications with a Console / Terminal

Data can be displayed on the screen and can be input from the keyboard of a console/terminal connected to the QD51 (-R24).

A console is a device (PC/AT personal computer) connected to a channel that is set as a console by the GX Developer intelligent function module switch settings in the QD51 (-R24).

A terminal is a device connected to a channel which is not set as a console in the QD51 (-R24).

In a console and terminal, the procedure for screen display and input from the keyboard differ as shown below.

```
In the case of a console                          In the case of a terminal

    ( Start )                                           ( Start )
        │                                                   │
        │                                                   ▼
        │                                    ┌──────────────────────────────┐
        │                                    │ • Opening of the interface and│
        │                                    │   setting of communications   │
        │                                    │   parameters, etc.            │
        │                                    └──────────────────────────────┘
        │                                                   │
        └─────────────────────┬─────────────────────────────┘
                              ▼
                ┌──────────────────────────────┐
                │ • Setting of the channel No.  │
                │   for the console / terminal. │
                └──────────────────────────────┘
                              │
                              ▼
                ┌──────────────────────────────┐
                │ • Output to the screen, screen│
                │   control.                    │
                │ • Key input from the keyboard.│
                └──────────────────────────────┘
                              │
                              ▼
                         ( End )
```
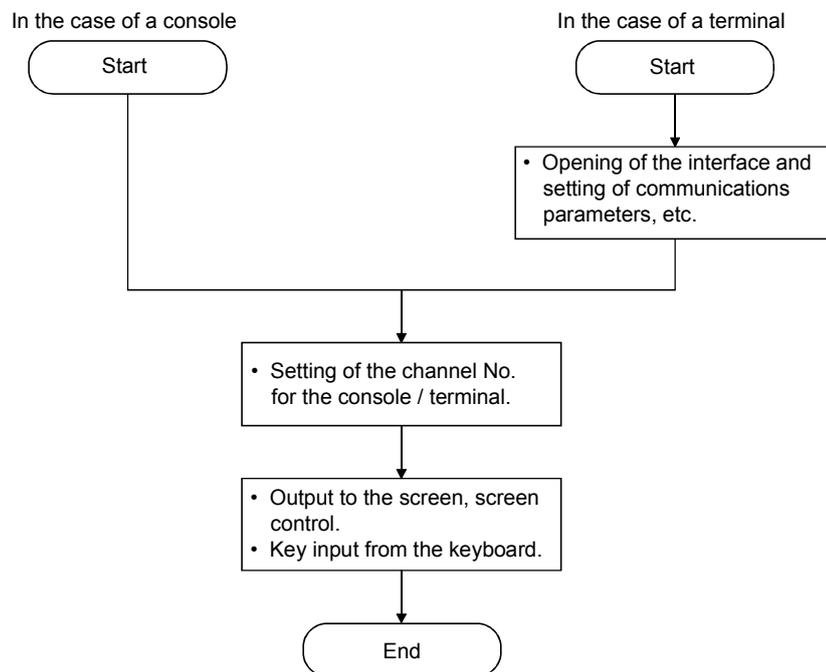
┌──────────┐
│ REMARK │
└──────────┘

Refer to Section 5.6.2 for setting of a console by GX Developer intelligent function module switch settings.

4.2.1 Communications with a console

(1)   Display of data on a console screen
Here, the way of thinking in the case of display of data on a console screen is explained.

Console
[In the case of a PC/AT personal computer]

QD51

CH.1

Data output
(Display of data to the screen)
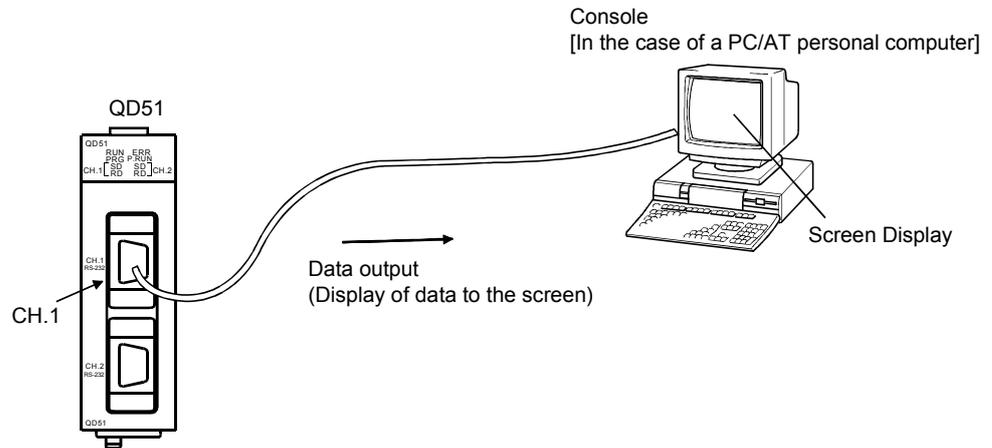
Screen Display

[Procedure for displaying data to a console screen]
Carry out display of data to a console screen by the following procedure.

Start

Set the data
output destination.   ······ Set the data output destination in the console channel
by the ZODV command.

Output data to the
console.   ······ Output data to the console by the PRINT command, etc.

Display data in the
console.

End

[Commands and functions used]
The commands and functions used to display data on a console screen are shown below.

Command, Function Name	Processing Contents
ZODV	Switches the data output destination.
PRINT	Displays data in the console screen.
PRINT USING	Displays data in the console screen. (The format can be specified.)
LOCATE	Specifies the data display position in a console screen.
CLS	Clears the console screen.

(2)   Inputting data from a console keyboard
Here, the way of thinking in the case of inputting of data from a console keyboard is explained.

Console
[In the case of a PC/AT personal computer]

QD51

Data input
(Input data from keyboard)

CH.1
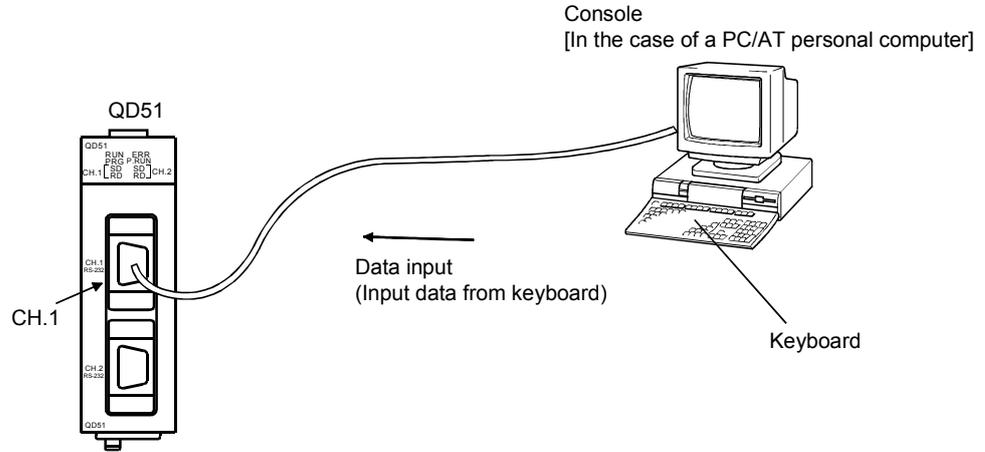
Keyboard

[Procedure for inputting data from a console keyboard]
Carry out inputting of data from a console keyboard by the following procedure.

```
        ┌─────────────┐
        │    Start     │
        └──────┬──────┘
               │
     ┌─────────┴─────────┐
     │ Set the data input │ ······ The data input destination is set in the console channel
     │ destination.       │          by the ZIDV command.
     └─────────┬─────────┘
               │
     ┌─────────┴─────────┐
     │ Wait for key input │ ······ Execution of an INPUT command, etc.
     │ from the keyboard. │
     └─────────┬─────────┘
               │
     ┌─────────┴─────────┐
     │ Key input from the │
     │ keyboard.          │
     └─────────┬─────────┘
               │
        ┌──────┴──────┐
        │     End      │
        └─────────────┘
```
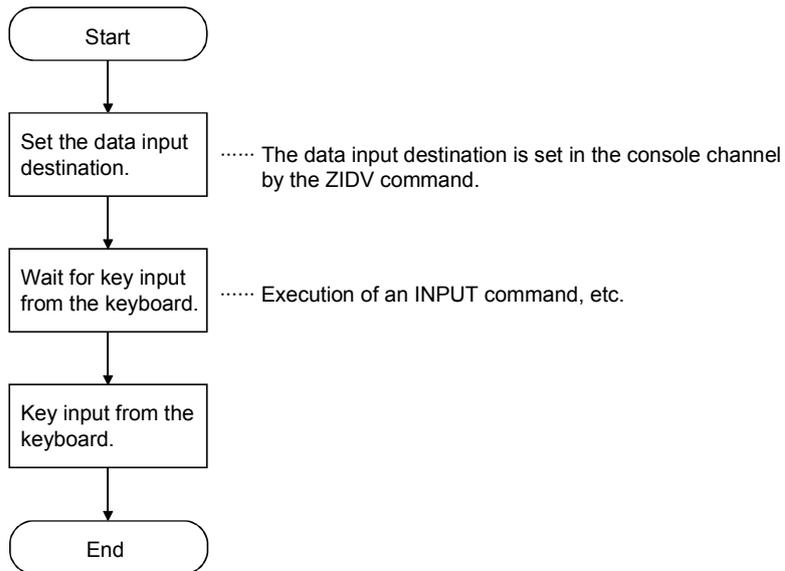
[Commands and functions used]

The commands and functions used to input data from the console keyboard are as shown below.

Command, Function Name	Processing Contents
ZIDV	Switches the data input destination.
INPUT	• Carries out key input of data. • Character code control codes (0_H to $1F_H$), commas ($2C_H$) and $\boxed{DEL}$ ($7F_H$) cannot be handled as data.
LINE INPUT	• Carries out key input of data. • Character code control codes (0_H to $1F_H$) and $\boxed{DEL}$ ($7F_H$) cannot be handled as data.
INKEY$ INPUT$	• Carries out key input of data. • Character code control codes (0_H, 03_H, 13_H, 80_H and FD_H to FF_H) cannot be handled as data.

4.2.2 Communications with a terminal
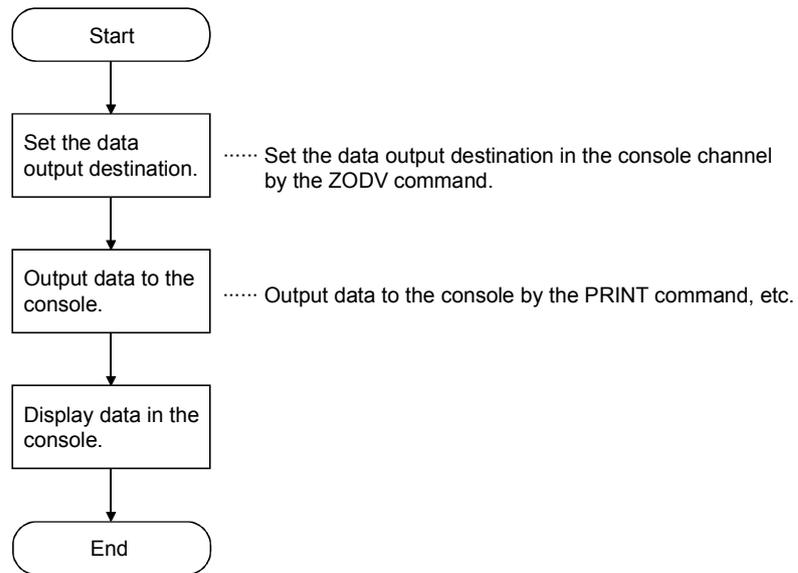
(1) Displaying data on a terminal screen

Here, the way of thinking in the case of displaying data on the terminal screen is explained.

[Procedure for displaying data on a terminal screen]

The procedure for displaying data on a terminal screen is as shown below.
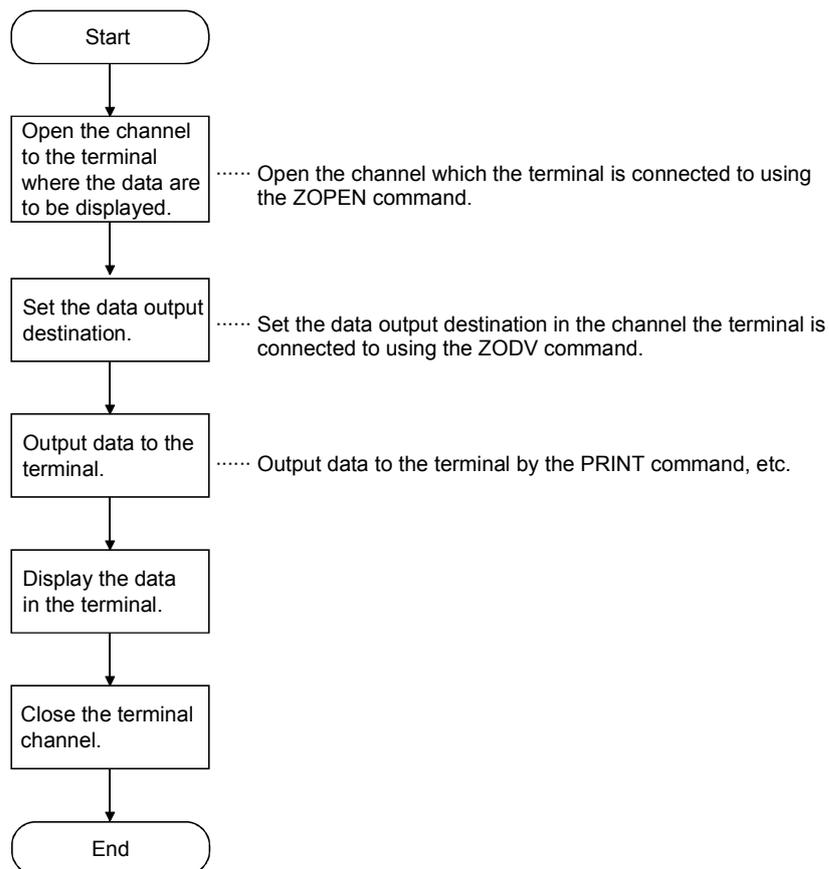
[Command and functions used]

The commands and functions used to display data on the terminal screen are as shown below.

Command, Function Name	Processing Contents
ZOPEN	Opens the interface and prepares for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets the communications parameters in the currently open interface and reads the interface status.
ZODV	Switches the data output destination.
PRINT	Displays the data on the terminal screen.
PRINT USING	Displays the data on the terminal screen. (The format can be designated).
LOCATE	Specifies the data display position on the terminal screen.
CLS	Clears the terminal screen.

(2)   Input from the terminal keyboard
Here the way of thinking when data are input from the terminal keyboard is explained.

Terminal

QD51

CH.1

Keyboard

Data input
(Input data from keyboard)
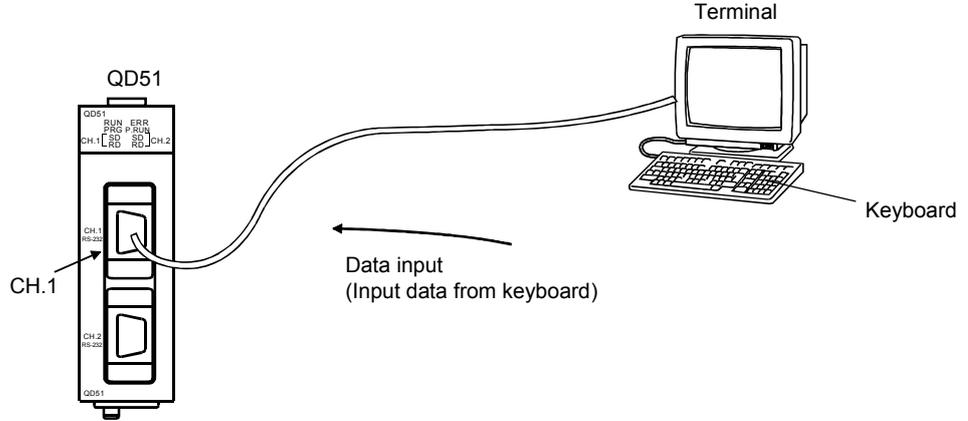
[Input procedure from the terminal keyboard]
The input procedure for input of data from the terminal keyboard is shown below.

```
    Start
```

| Open the channel of the terminal used for key input. | ...... Open the channel connecting to the terminal using the ZOPEN command. |

| Set the data input destination. | ...... Set the data input destination in the console channel using the ZIDV command. |

| The QD51 (-R24) waits for key input from the keyboard. | ...... Execute the INPUT command, etc. |

| Data are input using the keys of the keyboard. | |

| Close the terminal channel. | ...... Close the channel which the terminal is connected to using the ZOPEN command. |

```
    End
```

[Command, functions used]
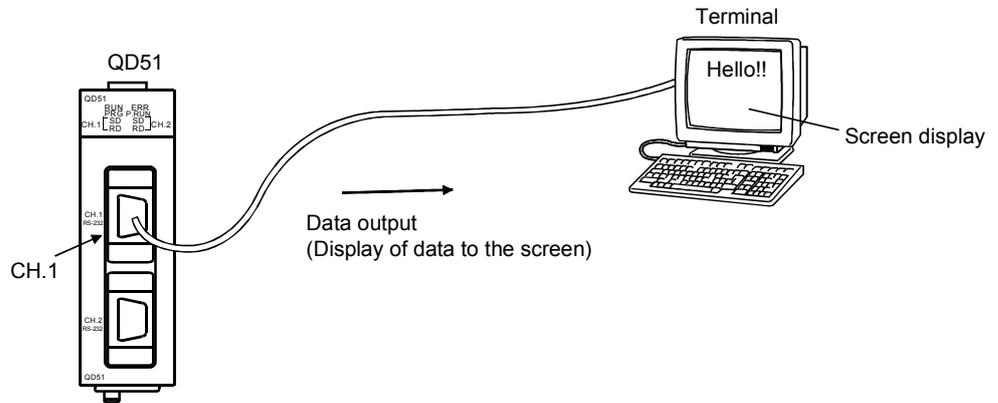
The commands and functions used to input data from a terminal keyboard are as shown below.

Command, Function Name	Processing Contents
ZOPEN	Opens the interface and prepares for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets the communications parameters in the currently open interface and reads the interface status.
ZIDV	Switches the data input destination.
INPUT	• Carries out key input of data. • Character code control codes (0_H to $1F_H$), commas ($2C_H$) and $\boxed{DEL}$ ($7F_H$) cannot be handled as data.
LINE INPUT	• Carries out key input of data. • Character code control codes (0_H to $1F_H$) and $\boxed{DEL}$ ($7F_H$) cannot be handled as data.
INKEY$ INPUT$	• Carries out key input of data. • Character code control codes (0_H, 03_H, 13_H, 80_H and FD_H to FF_H) cannot be handled as data.

4.3 Printing by a Printer

Data can be printed out from a printer connected to the QD51 (-R24)'s RS-232 interface.
A printer can also be connected to a PC/AT personal computer which is being used as a console, and can be used to print out data.

The procedure for printing out data differs in the following two cases.
1) Printer connected to an interface other than a console.
2) Printer connected to a console (PC/AT personal computer)

In the case of a printer connected to a console	In the case of a printer connected to an interface other than a console.
Start	**Start**
	Open the interface and set the communications parameters, etc.
Set the No. of the channel the printer is connected to.	Set the No. of the channel the printer is connected to.
Output data to the printer.	Output data to the printer.
	Close the interface.
End	**End**
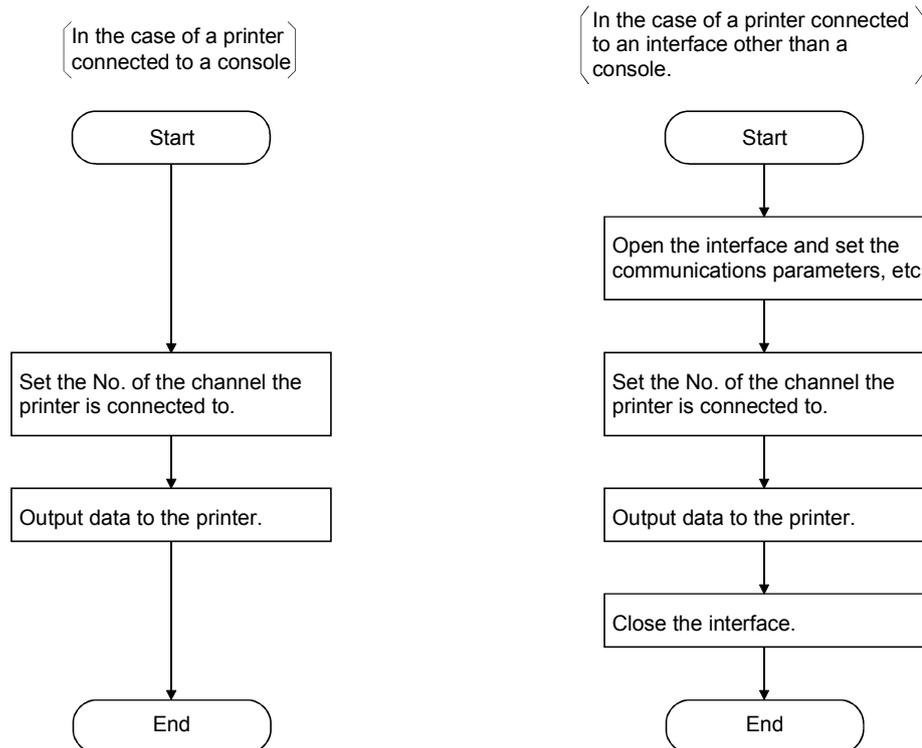
4.3.1 Printing from a printer connected to an interface that is not set in the console

Here, the way of thinking in the case of printing from a printer connected to an interface that is not set in the console is explained.
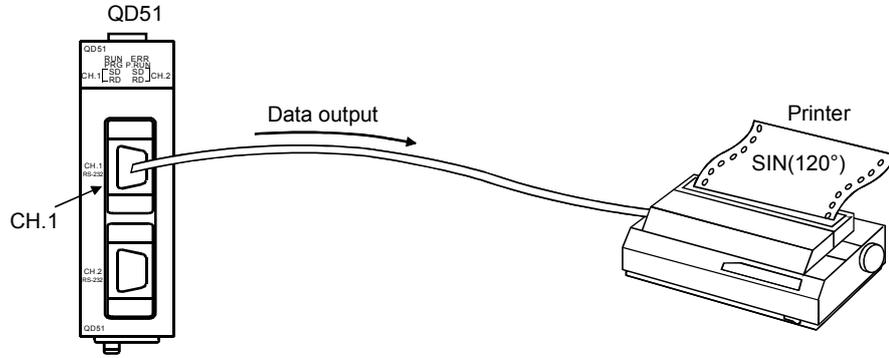
[Printout procedure]

The procedure for printing from a printer connected to an interface that is not set in the console is shown below.
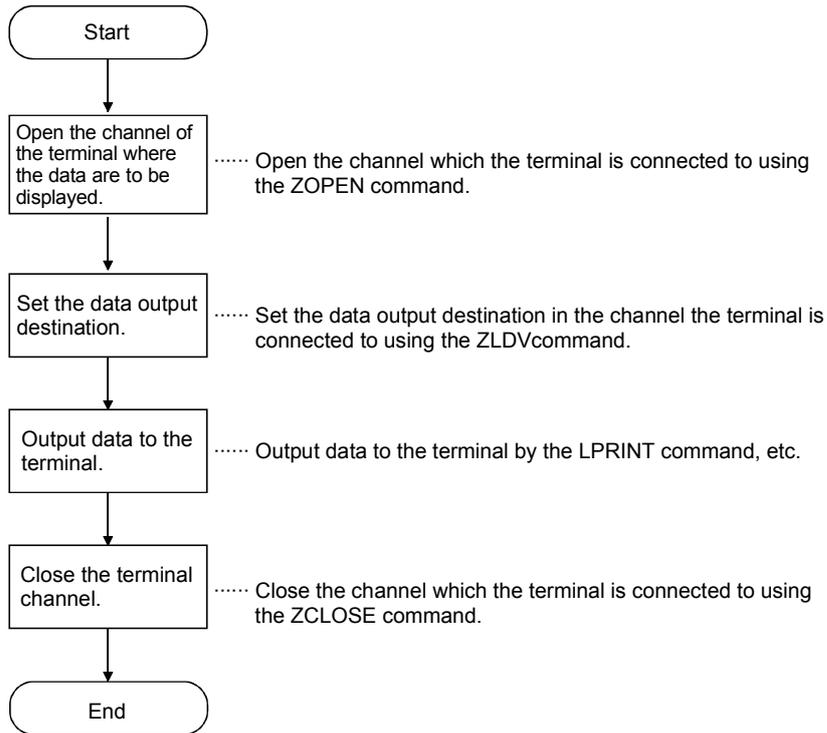
[Commands and functions used]

The commands used for printing from a printer connected to an interface that is not set in the console is as shown below.

Command, Function Name	Processing Contents
ZOPEN	Opens the interface and prepares for communications with an external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets the communications parameters in the currently open interface and reads the interface status
ZLDV	Switches the channel No. of the interface where the printer is connected.
LPRINT	Outputs data to the designated printer.
LPRINT USING	Outputs data to the designated printer. (The format can be specified.)
LLIST	Outputs the program list to the designated printer.

4.3.2 Printing from a printer connected to the console

Here, the way of thinking in the case of printing out by a printer that is connected to a PC/AT personal computer which is set as a console and connected to the interface is explained.

Console
[in the case of a PC/AT personal computer]

QD51

CH.1

Data output

Data output
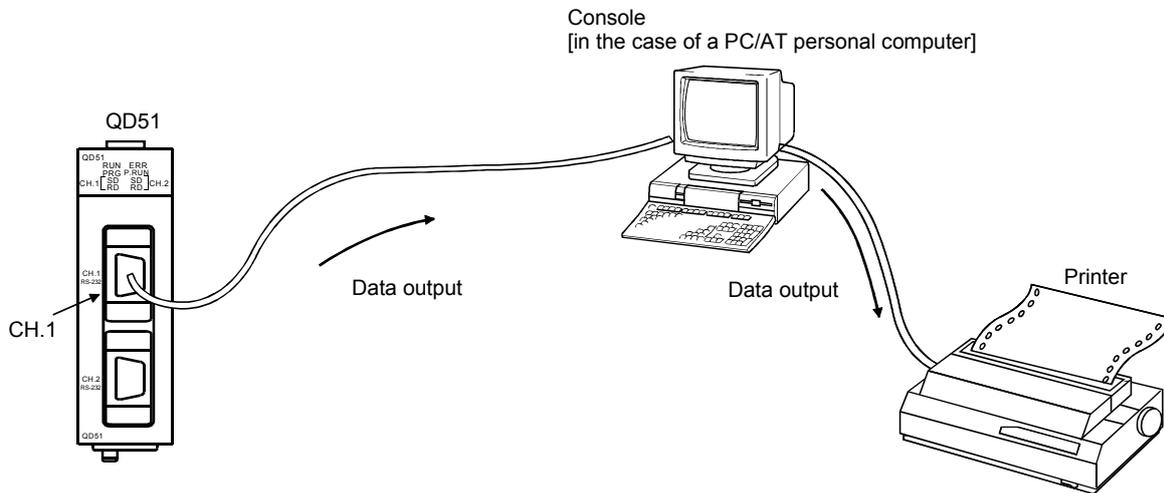
Printer

[Printout procedure]

The procedure for printing out from a printer connected to the console is shown below.

Start

Set the data output destination. ······ Sets the data output destination by the ZLDV command.

Output the data to the printer. ······ Outputs data to the printer by the LPRINT command, etc.

End
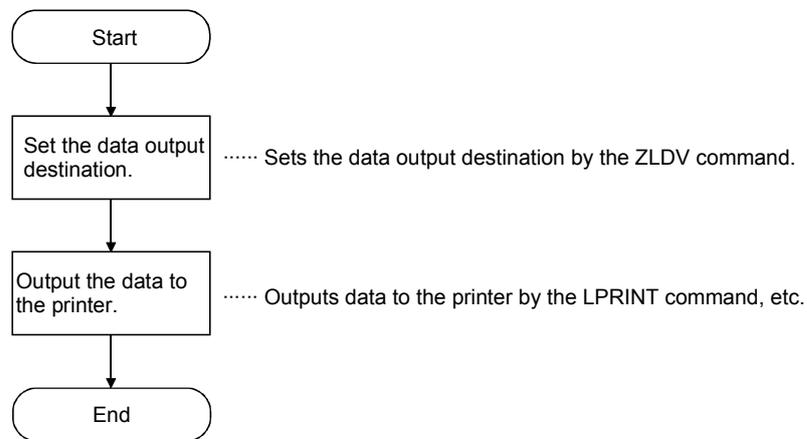
[Commands and functions used]

The commands and functions used to print out by a printer connected to the console are shown below.

Command, Function Name	Processing Contents
ZLDV	Switches the output destination to the printer connected to the console.
LPRINT	Outputs data to the designated printer.
LPRINT USING	Outputs data to the designated printer. (The format can be specified.)
LLIST	Outputs the program list to the designated printer.

4.4 Communications with the External Device

An external device ＊1 can be connected to the QD51 (-R24)'s RS-232 interface or RS-422/485 interface and communications carried out between them.

4.4.1 If data are being sent

Here, the way of thinking in the case of sending data from the QD51 (-R24) to a device other than a console, terminal or printer.

If data are being sent, the procedure is as follows.
1) Open the interface.
2) Prepare the data to be sent.
3) Specify the sent data byte count and time out time.
4) Send data by a ZSEND command.
5) Close the interface.
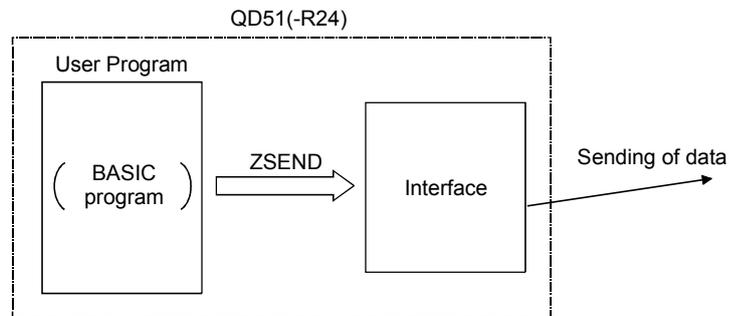
[Commands and functions used]

The commands and functions used to send data from the QD51 (-R24) to a device other than a console, terminal or printer are shown below. ＊2

Command, Function Name	Processing Contents
ZOPEN	Opens the interface and prepares for communications with the external device.
ZCNTL	Sets the communications parameters in the currently open interface and reads the interface status.
ZSEND	Sends the data from the designated interface.
ZCLOSE	Closes the interface.

REMARK

＊1 A console, terminal and printer are excluded.
    Communications with a console or terminal   : Refer to Section 4.2.
    Printout from a printer   : Refer to Section 4.3.
＊2 Refer to the Programming Manual (Commands) for details concerning commands and functions.

4.4.2 If data are received

Here, the way of thinking in the case of receiving data sent to the QD51 (-R24) from a device other than a console or terminal is explained.

If data are received, the procedure is as follows.
1)   Open the interface.
2)   Data are received.
3)   The received data are saved to variables by the ZRECEIVE Command/INPUT$ function.
4)   Close the interface.
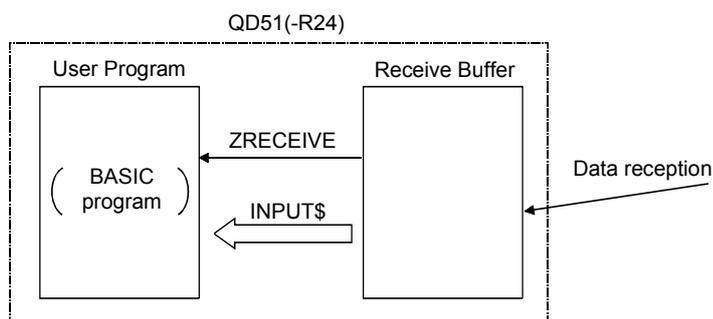
QD51(-R24)

[Commands and functions used]

The commands and functions used in receiving data sent to the QD51 (-R24) from a device other than a console or terminal are shown below.

Command, Function Name	Processing Contents
ZOPEN	Opens the interface and makes preparations for communications with the external device.
ZCLOSE	Closes the interface.
ZCNTL	Sets the communications parameters in the currently open interface and reads the interface status.
INPUT$	• 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.
ZRECEIVE	• Stores the data received from the interface in the designated variable.

REMARK

Refer to the Programming Manual (Commands) concerning details of commands and functions.

4.5 Communications with the Programmable controller CPU

Data can be written to the programmable controller CPU and read from the programmable controller CPU by the QD51 (-R24) using a BASIC program.
The data which can be written to and read from the programmable controller CPU are shown below.

No.	Data Name	Write	Read
1	Device Memory	○	○
2	Programmable controller CPU Model Name	×	○
3	Programmable controller CPU remote RUN/STOP	○	×
4	Interrupts to the programmable controller CPU	○	×
5	Intelligent function module/Special function module buffer memory	○	○

○: Can be run   ×: Cannot be run

Data can also be written to and read from the buffer memory of another intelligent function module/special function module using a BASIC program.

The way of thinking in the case where data are written to and read from the buffer memory in a device memory or an intelligent function module/a special function module is explained below.

4.5.1 ON/OFF data communications by general output (X/Y)

Using the QD51 (-R24)'s general purpose input / output, ON/OFF data can be communicated between the QD51 (-R24) and the programmable controller CPU.
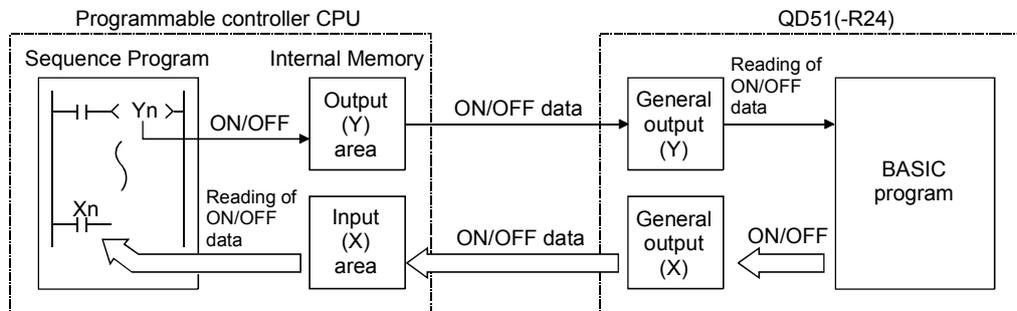
(1)   In the case of programmable controller CPU → QD51 (-R24)
Sending of ON/OFF data from the programmable controller CPU to the QD51 (-R24) is carried out by general output (Y).
The QD51 (-R24) uses the special variable B@ in a BASIC program to read the general output turned ON/OFF by the programmable controller CPU.

(2)   In the case of QD51 (-R24) → Programmable controller CPU
Sending of ON/OFF data from the QD51 (-R24) to the programmable controller CPU is carried out by general input (X).
The programmable controller CPU uses a sequence program to read the general input turned ON/OFF by a QD51 (-R24) BASIC program.

[Special variables used]
The special variables used to read ON/OFF data from general input (X) and general output (Y) are shown below.

Special Variable		Processing Contents
B@ (X, Formula)	:	Turns the QD51 (-R24)'s general input (X) ON and OFF.
B@ (Y, Formula)	:	Reads ON/OFF data from the QD51 (-R24)'s general output (Y).

4.5.2 Reading and writing data from/to buffer memory

16-bit data communications can be carried out between the QD51 (-R24) and the programmable controller CPU using the QD51 (-R24)'s buffer memory.
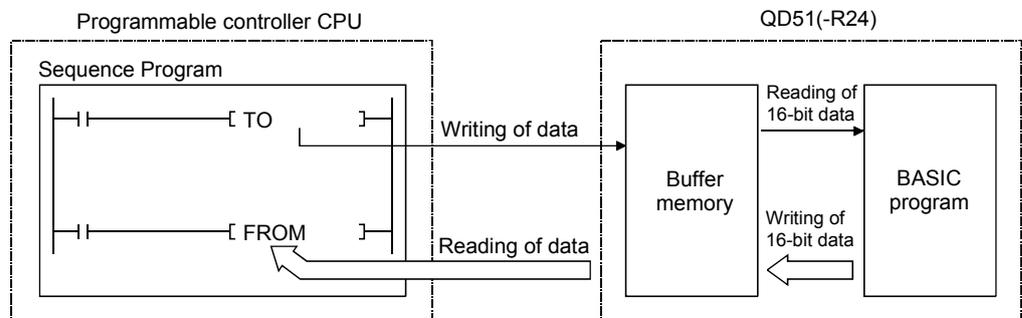
(1)   In the case of programmable controller CPU → QD51 (-R24)
16-bit data written to the QD51 (-R24)'s buffer memory by the programmable controller CPU's sequence program (TO command) are read using a BASIC program.

(2)   In the case of QD51 (-R24) → Programmable controller CPU
16-bit data written to the QD51 (-R24)'s buffer memory by a BASIC program are read by the programmable controller CPU's sequence program (FROM command).

[Commands and functions used]
Commands and functions used to read/write 16-bit data from/to the QD51 (-R24)'s buffer memory are shown below.

Command, Function Name		Processing Contents
GETMEM	:	Reads the contents of the QD51 (-R24)'s buffer memory.
PUTMEM	:	Writes 16-bit data to the QD51 (-R24)'s buffer memory.

4.5.3 Reading and writing data from/to the programmable controller CPU

Data can be written to and read from the programmable controller CPU's device memory directly by a QD51 (-R24) BASIC program.
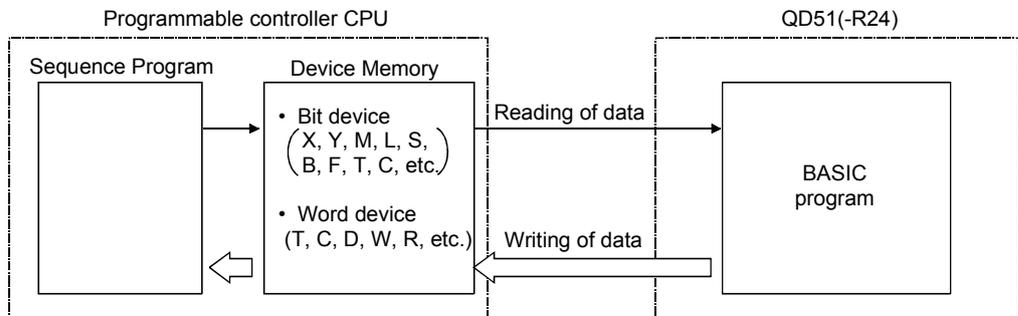
Refer to the Programming Manual (Commands) for the details of the device memories to and from which data can be written and read by the BASIC program.

[Commands and functions used]
The commands and functions used to write data to and read data from the programmable controller CPU's device memory are shown below.

Command, Function Name		Processing Contents
PCRD	:	Reads data from the programmable controller CPU.
PCWT	:	Writes data to the programmable controller CPU.

4.5.4 Reading and writing data from/to the programmable controller CPU's expansion file register

Data can be written to and read from the programmable controller CPU's expansion file register (file registers after block No. 1) directly by a QD51 (-R24) BASIC program.
Refer to Section 4.5.3 concerning writing of data to and reading data from a file register (block No. 0).

> **POINT**
>
> The QD51 (-R24) can read or write the expansion file register data only when accessing an A series programmable controller via network.
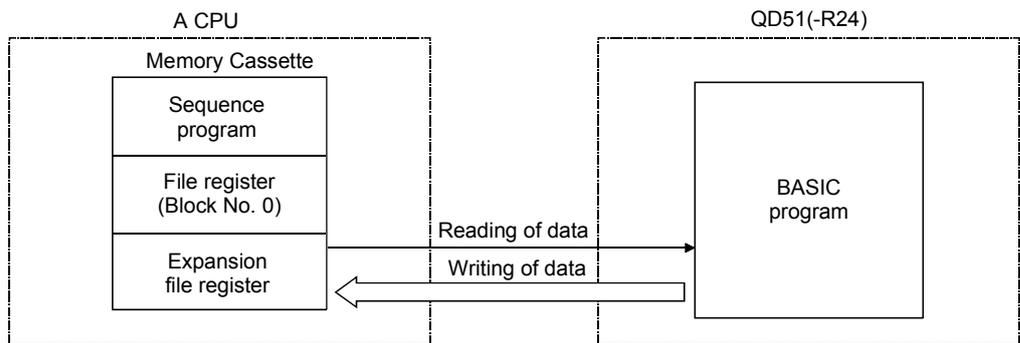
[Commands and functions used]
Commands and functions used to write data to and read data from the programmable controller CPU's expansion file register are shown below.

Command, Function Name		Processing Contents
PCRD	:	Reads data from the programmable controller CPU.
PCWT	:	Writes data to the programmable controller CPU.

4.6 Reading and Writing Data from/to the Buffer Memory of Another Intelligent Function Module/Special Function Module

Data can be written to and read from the buffer memory of another intelligent function module or special function module by a QD51 (-R24) BASIC program.
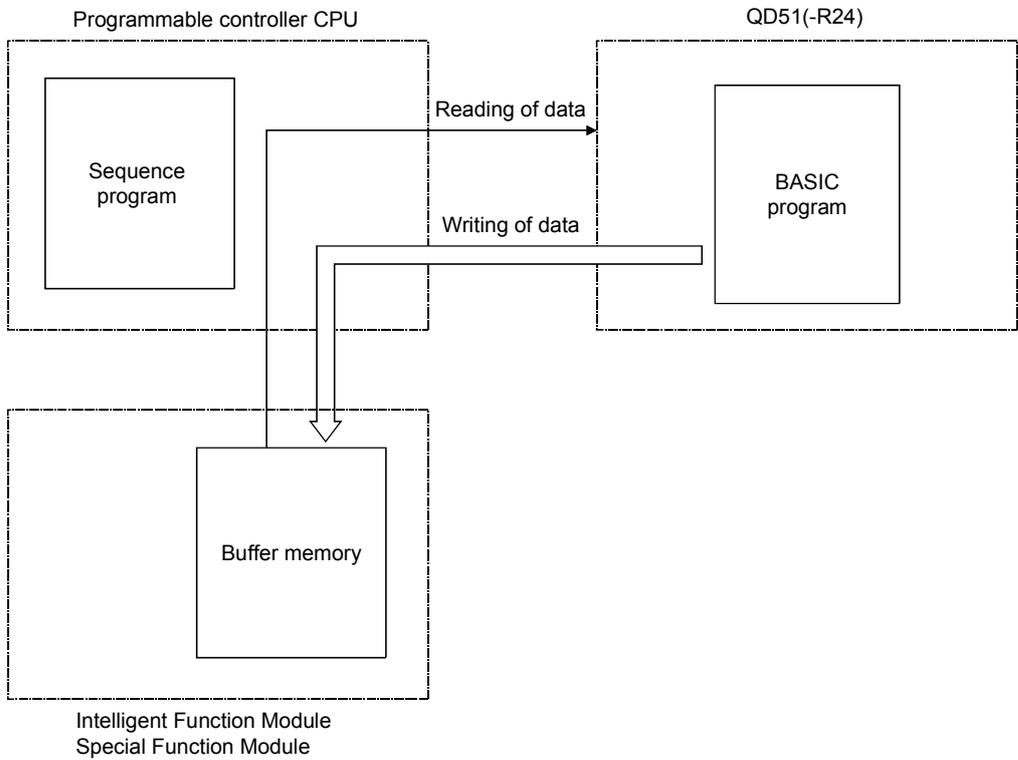
Programmable controller CPU   QD51(-R24)

```
Sequence
program
```
Reading of data

Writing of data

```
BASIC
program
```

```
Buffer memory
```

Intelligent Function Module
Special Function Module

[Commands and functions used]

Commands and functions used to write data to and read data from the buffer memory of another intelligent function module or special function module are shown below.

Command, Function Name		Processing Contents
PCRD	:	Reads data from the programmable controller CPU.
PCWT	:	Writes data to the programmable controller CPU.

REMARK

Refer to the Programming Manual (Commands) concerning details of the above commands and functions.

4.7 Reading and Writing Clock Data

The QD51 (-R24) does not include a clock function.
If a clock function is used with the QD51 (-R24), read the calendar (Year, Month, Day, Day of Week) or time (Hour, Minute, Second) from a programmable controller CPU with a clock function using a BASIC program.

Refer to the User's Manual for each programmable controller CPU to confirm which programmable controller CPU has a clock function.

If you are using a programmable controller CPU with a clock function, it is necessary to write the calendar date and time to the clock element.
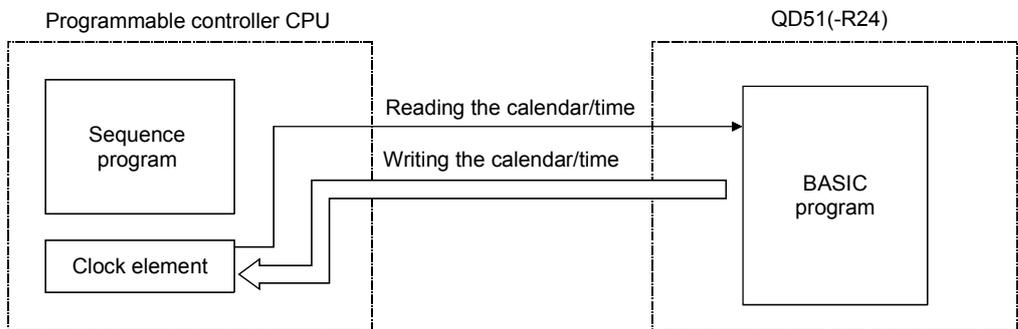
[Commands and functions used]

Commands and functions used to write the calendar and time to, and read them from the programmable controller CPU are shown below.

Command, Function Name		Processing Contents
DATE$	:	Writes the calendar to and reads the calendar from the programmable controller CPU's clock element.
TIME$	:	Writes the time to and reads the time from the programmable controller CPU's clock element.

POINT

The Q Series programmable controllers set the year in the clock data with 4 digits.
Effective years are 1990 to 2079.

4.8 Data Communications Between Tasks

During multitask processing, the content of variables or arrays in each task, etc. can only be referred to in each respective BASIC program.

For exchange of data between BASIC programs during multitask processing, use the following.
1)   Expansion Relay (EM)   : ON/OFF Data
2)   Expansion Register (ED)   : Data
3)   Common Memory   : Data

4.8.1 ON/OFF data communications by expansion relay (EM)

The QD51 (-R24)'s expansion relay (EM) is a device which can perform ON/OFF control and read the ON/OFF status from all tasks.
(For details of the expansion relay, refer to Section 3.5.6.)
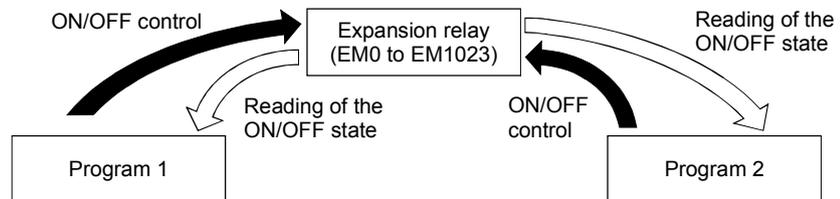
The expansion relays that can be used by the QD51 (-R24) are EM0 to EM1023, 1024 points.

[Commands and functions used]
Commands and functions that perform ON/OFF data communications by the expansion relay are shown below.

Command, Function Name		Processing Contents
B@	:	Performs expansion relay ON/OFF control and reading of the ON/OFF status.

4.8.2 Data communications by expansion register (ED)

The QD51 (-R24)'s expansion register (ED) is a device which can store data from all tasks and perform reading of data.
(For details of the expansion relay, refer to Section 3.5.4.)

Data storage → Expansion register (ED0 to ED1023) ← Reading of data

Reading of data   Storing of data

Program 1   Program 2
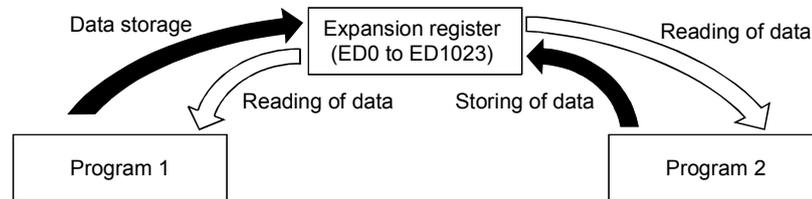
The expansion registers that can be used by the QD51 (-R24) are ED0 to ED1023, 1024 points.

[Commands and functions used]
The commands and functions used to carry out data communications by the expansion registers are shown below.

Command, Function Name		Processing Contents
W@	:	Performs storage of data in the expansion register and reading of data.
GETMEM	:	Performs reading of expansion register data.
PUTMEM	:	Performs storage of data in the expansion register.

4.8.3 Data communications by common memory

Buffer memory in the QD51 (-R24) is common memory which can be used to carry out storage and reading of data from all tasks.
(For details of the common memory, refer to Section 3.5.3.)

Data storage → Common memory ← Reading of data

Reading of data   Storing of data

Program 1   Program 2
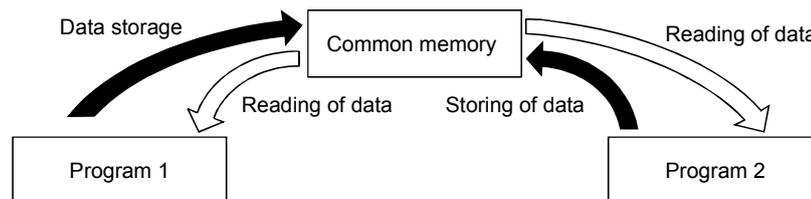
The common memory that can be used in the QD51 (-R24) consists of 4 k words (8 k bytes) of memory space.

[Commands and functions used]
The commands and functions used to carry out data communications by the common memory are shown below.

Command, Function Name		Processing Contents
GETMEM	:	Performs reading of data in common memory.
PUTMEM	:	Performs storage of data in common memory.

5 SETTINGS AND PROCEDURES UP UNTIL OPERATION

Here, the procedures and setting methods up to the point where operation can begin in a system where the QD51 (-R24) is used are explained.

POINT
(1) When the QD51 (-R24) is used, be sure to read the "Safety Precautions" provided in the front of this Manual.
(2) Packaging and the installation environment for the QD51 (-R24) are the same as for the programmable controller CPU.
(3) Refer to the User's Manual for the programmable controller CPU you are using concerning the module's packaging and installation.

5.1 Handling Precautions

Here the precautions to be observed when handling the QD51 (-R24) module are explained.

(1)   The QD51 (-R24)'s case is made of resin, so be sure not to drop it or subject it to strong impacts.

(2)   Tighten the screws such as module fixing screws within the following ranges.

Screw location	Tightening torque range	Remarks
RS-422/485 terminal block terminal screw (M3 screw)	0.42 to 0.58N•m	—
RS-232 cable connector screw (M2.6 screw)	0.20 to 0.39 N•m	Screw hole depth: L=3.2mm or less (Internal length from the surface)
Module fixing screw (normally not required) (M3 screw) (*[1])	0.36 to 0.48N•m	—

(*1) The module can be easily fixed onto the base unit using the hook at the top of the module.
However, it is recommended to secure the module with the module fixing screw if the module is subject to significant vibration.

5.2 Settings and Procedures Up Until Operation

A brief list of procedures up until operation of this module is shown below.

```
                    ( Start )
                        │
                        ▼
        ┌───────────────────────────────┐
        │ Check the functions used and the │
        │ specifications.                 │
        └───────────────────────────────┘
                        │
                        ▼
        ┌───────────────────────────────┐       • Connect the console/debugger.
        │ Connect the console and the module with │     • Refer to Section 5.4 of this manual.
        │ cables.                         │
        └───────────────────────────────┘
                        │
                        ▼
        ┌───────────────────────────────┐       • Connect the QD51 (-R24) and the external device.
        │ Connect the external device and the │     • Refer to Section 5.5 of this manual.
        │ module with a cable.            │
        └───────────────────────────────┘
                        │
                        ▼
        ┌───────────────────────────────┐
        │ Connect the GX Developer and QCPU │
        │ with a cable.                   │
        └───────────────────────────────┘
                        │
                        ▼
        ┌───────────────────────────────┐       • Set the QD51 (-R24)s Input / Output assignment.
        │ Perform all the settings with GX Developer. │   • Carry out QD51 (-R24) mode setting (programming mode) and set operation.
        └───────────────────────────────┘       • Refer to Section 5.6 of this manual.
                        │
                        ▼
        ┌───────────────────────────────┐       • Commands and functions
        │ Carry out programming.          │       • Refer to the programming manual (Commands).
        └───────────────────────────────┘
                        │
                        ▼
        ┌───────────────────────────────┐
        │ Carry out debugging.            │
        └───────────────────────────────┘
                        │
                        ▼
        ┌───────────────────────────────┐       • Carry out QD51 (-R24)s mode setting (Run mode/Multitask Debug mode)
        │ Set the run mode by the GX Developer. │     and set operation.
        └───────────────────────────────┘       • Refer to Section 5.6 of this manual.
                        │
                        ▼
                    ( End )
```

┌──────────┐
│ REMARK   │
└──────────┘

The transmission rate and data format are set using the BASIC program (ZOPEN commands).
For details, refer to the Programming Manual (Commands).

5.3 Names of Parts and Functions
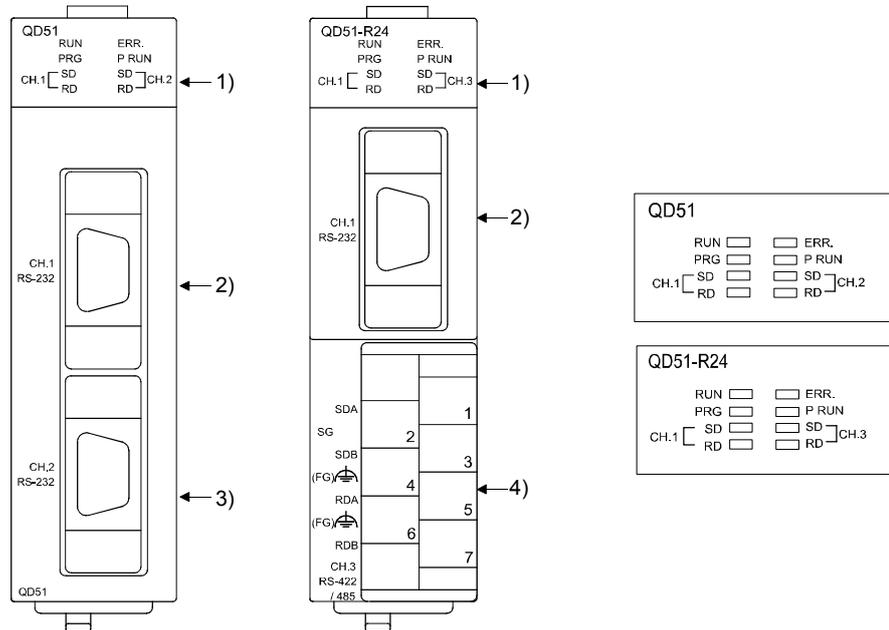
	Name	Contents
1)	Indicator LED	Indicator LED (Refer to (1) for the indication contents.)
2)	CH1-side RS-232 Interface	CH1 side RS-232 interface (D-Sub 9-pin) for serial communications with external devices and for connection of a console / debugger.
3)	CH2-side RS-232 Interface	CH2 side RS-232 interface (D-Sub 9-pin) for serial communications with external devices and for connection of a console / debugger.
4)	CH3-side RS-422/485 Interface	CH3 side RS-422/485 interface (two-piece terminal block) for making serial communication with the target device. (console/debugger cannot be connected.)

(1) List of indicator LED's

LED name	Indication content	●: Lights up/ Blinks	○: Off
RUN	Indicates normal operation.	Normal	Abnormal, reset
ERR.	Indicates whether or not there is an error ($*1$).	Occurrence of an error	Normal
PRG	Indicates the Mode.	Programming mode	Run / Multitask debug mode
P RUN	Indicates that a BASIC program is running.	BASIC program is running.	Basic program is stopped.
SD	Indicates the send state of each channel.	Data are being sent.	Not sending.
RD	Indicates the receive state of each channel.	Data are being received.	Not receiving.

($*1$) Lights up when a QD51 (-R24) hardware error occurs, or when a program's execution is abnormal.

5.4 Connection with a Console, Debugger

This section explains the connection methods of a personal computer with a console.

(1) Connection method

(a) Connection with a PC/AT personal computer
Cables need to be produced by the users to make a connection.

(2) Cable wiring

The wiring of the user-created cable is show below.

QD51 (-R24) side		Cable connection and signal direction (Connection example of full or half duplex communication)	Console side
Signal name	Pin number		Signal name
CD	1		CD
RD(RXD)	2		RD(RXD)
SD(TXD)	3		SD(TXD)
DTR(ER)	4		DTR(ER)
SG	5		SG
DSR(DR)	6		DSR(DR)
RS(RTS)	7		RS(RTS)
CS(CTS)	8		CS(CTS)
—	9		

CAUTION

The pin arrangement on the console side differs depending on the personal computer.
If the cable is made by the user, be sure to make the cables only after confirming the interface specifications for the personal computer you are using.

5.5 Connection with the External Device

Here the wiring between the QD51 (-R24) and the external device is explained.

As a precaution when connecting cables, and as a condition for achieving a highly reliable system in which the full potential of the QD51 (-R24)'s functions is realized, it is necessary to carry out wiring so that the influence of noise will be minimized.

(1)   Ground the shield at one point.

(2)   When connecting to the external device by a RS-232 line, use the connector shell shown in Section 3.2.1 for the connector on the QD51 (-R24) side.

(3)   M3 screws are used for the terminal screws on the RS-422/485 interface terminal block. Use compression terminals which are appropriate for these terminals.

(4)   For connections to the external device side, make connections after confirming the specifications for the external device.

(5)   Refer to Appendix 3 concerning the bending radius of connection cables.

5.5.1 Connecting to the RS-232 interface

Here connection precautions and an example of connection when carrying out full duplex communications using the QD51 (-R24)'s RS-232 interface are shown.

(1)   Precautions when connecting

(a)   Connect the FG signal and shield of the connection cable as shown below.

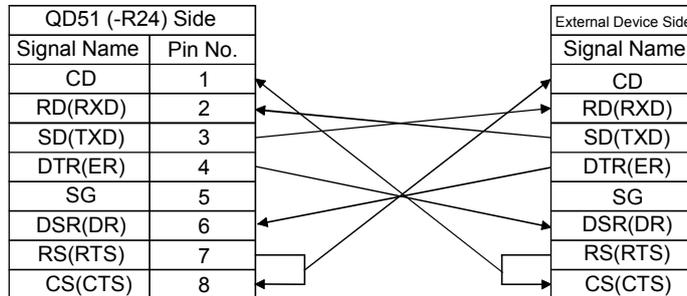|	Connection method	Remark
Connection cable FG signal	Connect to the connector body on the QD51 (-R24) side.	• Do not short circuit the connection cable FG signal and the SG signal.
Connection cable shield	Connect to the FG terminal of the external device side or the connector body of the QD51 (-R24).	• When the FG signal in the external device side is connected to the SG signal, do not connect the FG signal on the QD51 (-R24) side.

(b)   If data communications cannot be done normally due to external noise, make connections as shown below.

1)   Connect the external device side's FG terminal with the QD51 (-R24) side by the connection cable's shield.

2)   Connect each of the signals other than SG with the SG signal in twisted pairs.
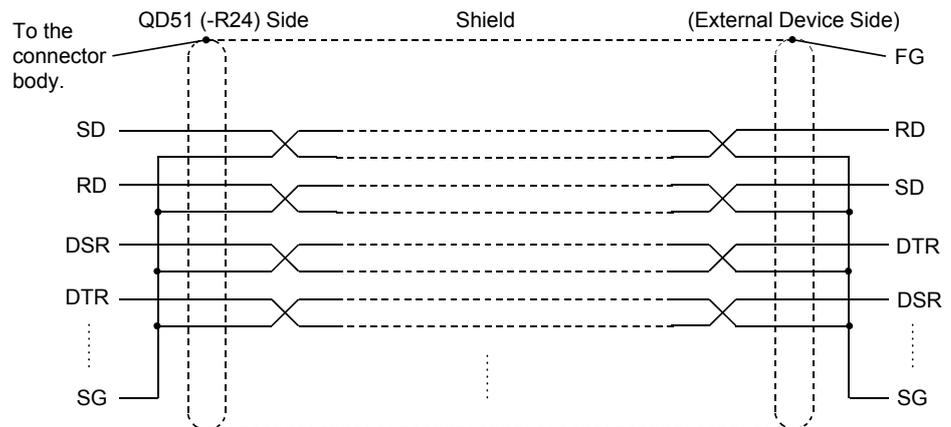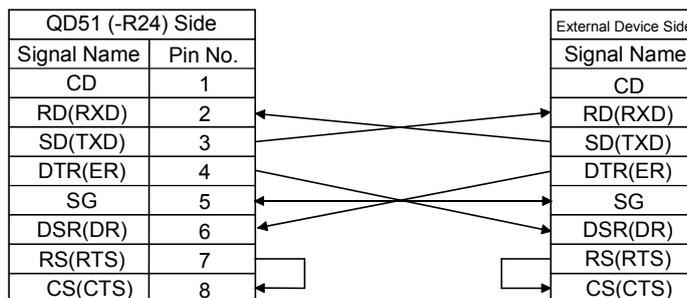
(2) Connection example

(a) Example of connection to an external device with a CD signal (pin 1) that can be turned ON and OFF.

QD51 (-R24) Side			External Device Side
Signal Name	Pin No.		Signal Name
CD	1		CD
RD(RXD)	2		RD(RXD)
SD(TXD)	3		SD(TXD)
DTR(ER)	4		DTR(ER)
SG	5		SG
DSR(DR)	6		DSR(DR)
RS(RTS)	7		RS(RTS)
CS(CTS)	8		CS(CTS)

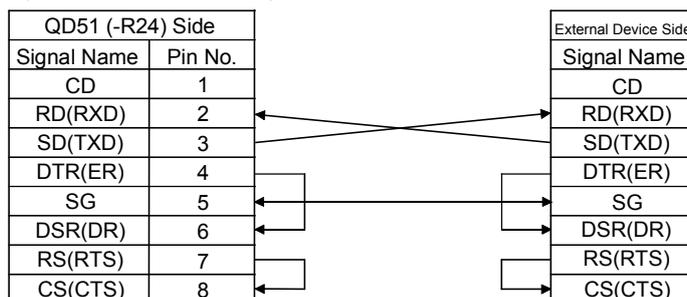✶ With the above wiring, it is possible to carry out DTR/DSR control and DC code control.

(b) Example of connection to an external device with a CD signal (pin 1) that cannot be turned ON and OFF.

1) Connection example 1

QD51 (-R24) Side			External Device Side
Signal Name	Pin No.		Signal Name
CD	1		CD
RD(RXD)	2		RD(RXD)
SD(TXD)	3		SD(TXD)
DTR(ER)	4		DTR(ER)
SG	5		SG
DSR(DR)	6		DSR(DR)
RS(RTS)	7		RS(RTS)
CS(CTS)	8		CS(CTS)

✶ With the above wiring, it is possible to carry out DTR/DSR control and DC code control.

2) Connection example 2

QD51 (-R24) Side			External Device Side
Signal Name	Pin No.		Signal Name
CD	1		CD
RD(RXD)	2		RD(RXD)
SD(TXD)	3		SD(TXD)
DTR(ER)	4		DTR(ER)
SG	5		SG
DSR(DR)	6		DSR(DR)
RS(RTS)	7		RS(RTS)
CS(CTS)	8		CS(CTS)

✶ With the above wiring it is possible to carry out DC code control.

POINT
When it is absolutely impossible to transmit data between the programmable controller CPU and an external device, together with a connection test, try data communications with the wiring connections shown in the above connection example 2. When data communications are possible with the wiring connections shown in connection example 2, correct the wiring connections after confirming the external device's interface specifications.

5.5.2 Connecting to the RS-422/485 interface

Here connection precautions and an example of connection when using the QD51 (-R24)'s RS-422/485 interface are shown.

(1) Precautions when connecting

(a) When the QD51 (-R24)'s SG signal and FG signal are connected to the external device, make connections in accordance with the external device's specifications.

(b) Connect the shield of the connection cable to the FG terminal of either one of the devices being connected.
If data communications cannot be done normally due to external noise, even when wiring connections are done as shown above, make wiring connections as shown below.

1) Connect the FG terminals of both sides with the connection cable's shield.
However, make the connection on the external device side in accordance with the external device's instruction manual.

2) Connect the QD51 (-R24)'s (FG) to the FG terminal of the power supply module in the station where the QD51 (-R24) is installed, or connect it to the FG terminal on the control panel where the QD51 (-R24) station's programmable controller is installed.

3) Connect the nnA and nnB of each signal in the connection cable in pairs.
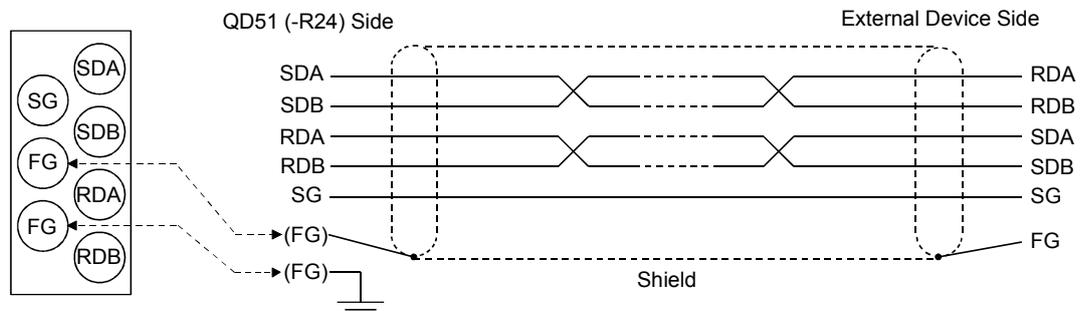
Correspondence between the RS-422/485 terminal block and signal positions

(c) It is necessary to set (or connect) a terminal resistor in both terminals when wiring.

On the QD51 (-R24) side, connect a terminal resistor (packed together with the QD51 (-R24))in accordance with this section to match the external device's specifications.

As for the external device side, connect or set a terminal resistor in accordance with the external device's user's manual.

(Terminal resistors connected to the QD51(-R24) side.)
- Connect a "330Ω, 1/4 W" terminal resistor when communicating using the RS-422 interface.
- Connect a "110Ω, 1/2 W" terminal resistor when communicating using the RS-485 interface.

＊ Distinguishing terminal resistors

330Ω   Orange Orange Brown   110Ω   Brown Brown Brown
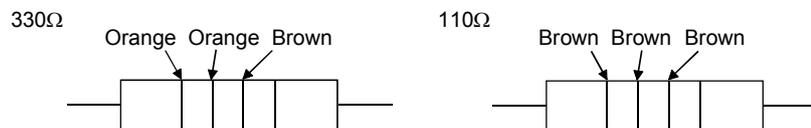
POINT
Including 1:n, n:1 and m:n connections, it is necessary that the device connecting to the QD51 (-R24)'s RS-422/485 interface have a matching RS-422 or RS-485 interface.

(2) Connection examples

(a) If connecting 1:1.

QD51 (-R24) Side	External Device Side	Terminal Resistor
Signal Name	Signal Name	
SDA	RDA	
SDB	RDB	R
RDA	SDA	
RDB	SDB	
SG	RSA	
FG	RSB	
FG	CSA	
	CSB	
	SG	
	FG	

Terminal Resistor R (on QD51 side)

R Terminal Resistor
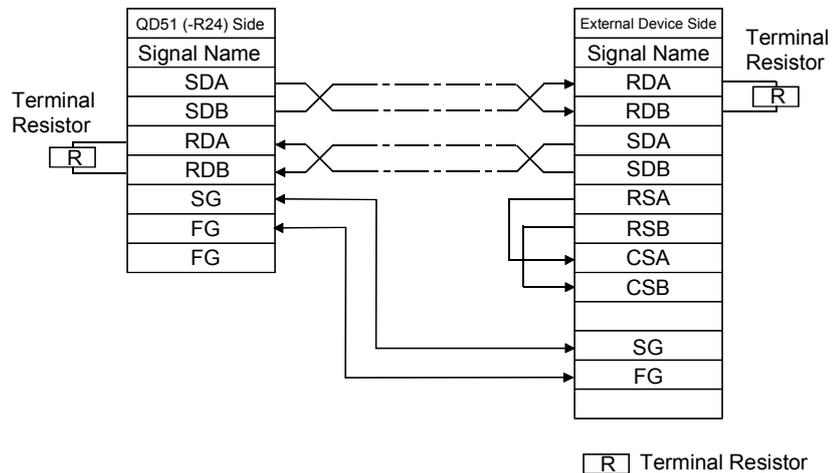
(b) If connecting 1:n (multiple drop)
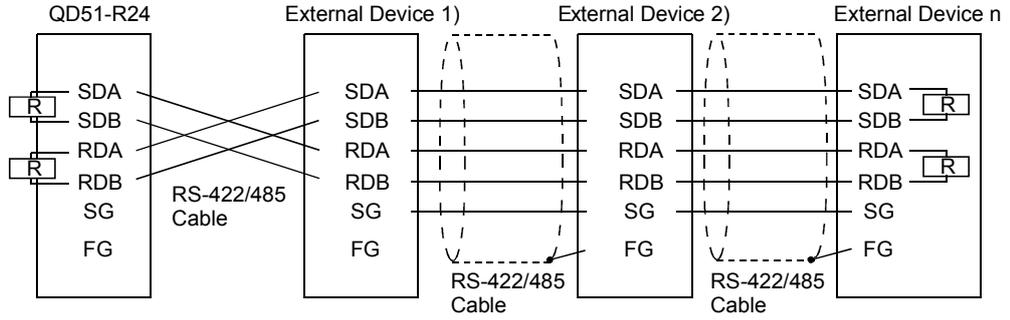
R Terminal Resistor

(c) If connecting n:1 (multiple drop)

R Terminal Resistor

(d) If connecting m:n (multiple drop) (External Device: QD51 (-R24))

R Terminal Resistor

5.6 Setting from the GX Developer

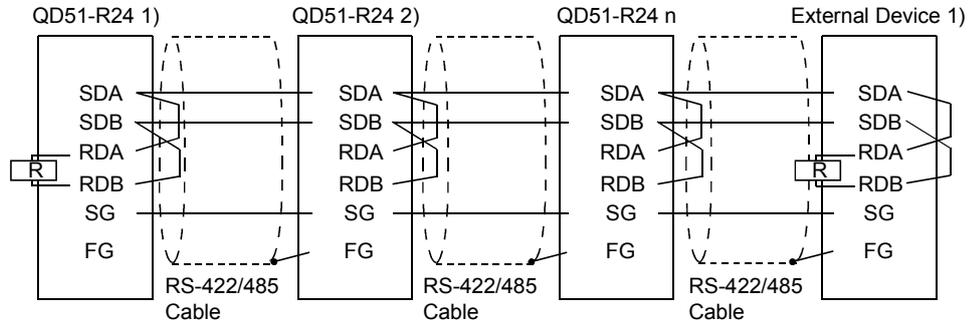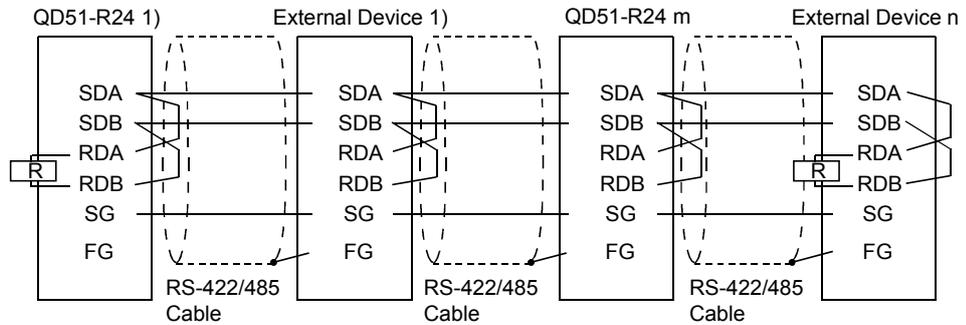Here, the various settings from the GX Developer which are necessary in order to carry out data communications with an external device from the QD51 (-R24) are explained.

5.6.1 I/O assignment settings
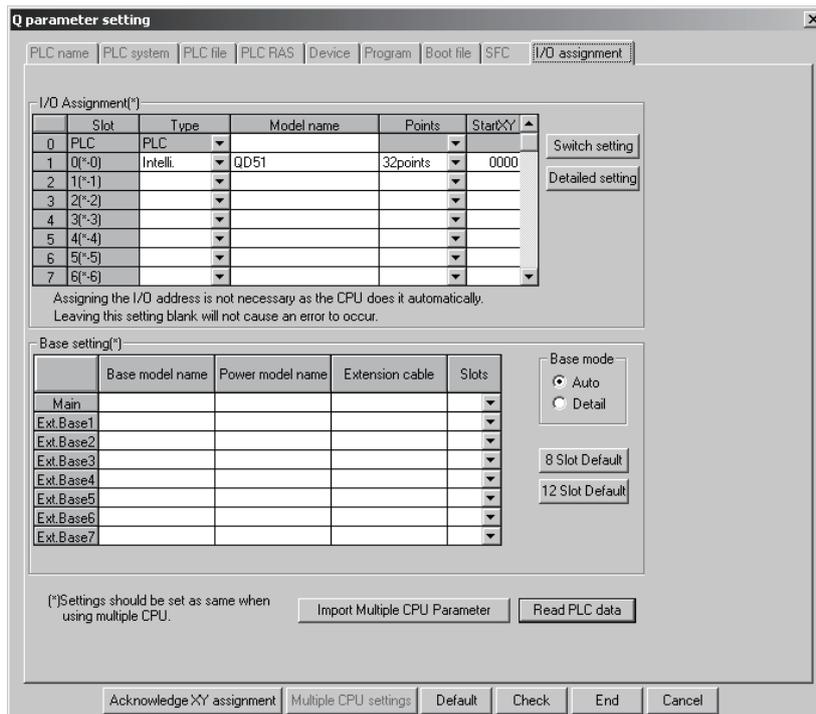
[Setting purpose]

I/O assignment settings are used to set the type of each module installed in the base module, to set the input/output signal ranges and to set the QD51 (-R24) switches.

[Start procedure]

[GX Developer] → [PC Parameters] → | I/O Assignment Setting |

Refer to the Operating Manual for the GX Developer concerning screen display methods.

[Setting screen]

[Display contents]

Item Name		Item Setting Contents	Remarks
I/O assignment	Type	Sets "Intelligent"	—
	Model name	Sets the model name of the installed module (QD51 or QD51-R24).	
	Number of points	Sets 32 points.	
	Header XY	Sets the header input / output (hexadecimal) of the relevant module.	
	Switch settings	Sets the console, debugger and mode setting switch, etc.	Refer to Section 5.6.2.
	Detailed settings	When a multiple CPU system is being used, this sets the control CPU for the QD51 (-R24).	Refer to QCPU User's Manual (Multiple CPU system).
Multiple CPU setting		Carries out settings when a multiple CPU system is used.	

5.6.2 Switch setting for I/O and intelligent function module

[Setting purpose]

The I/O module and intelligent function module switch settings set the specifications for data transfers with external devices, communications protocols, etc.
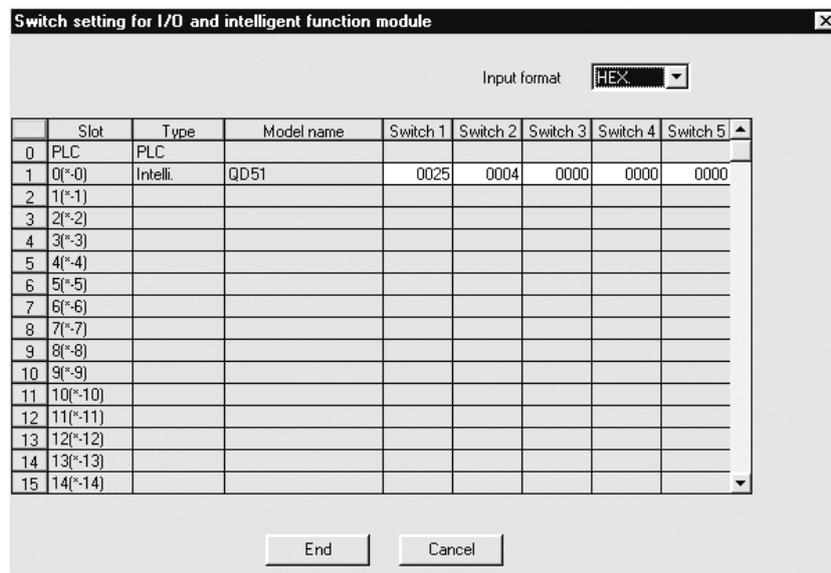
[Start procedure]

[GX Developer] → [PC Parameters] → [I/O Assignment Settings] →
| Switch Settings |

Refer to the Operating Manual for the GX Developer concerning screen display methods.

[Setting screen]

	Slot	Type	Model name	Switch 1	Switch 2	Switch 3	Switch 4	Switch 5
0	PLC	PLC						
1	0(*-0)	Intelli.	QD51	0025	0004	0000	0000	0000
2	1(*-1)							
3	2(*-2)							
4	3(*-3)							
5	4(*-4)							
6	5(*-5)							
7	6(*-6)							
8	7(*-7)							
9	8(*-8)							
10	9(*-9)							
11	10(*-10)							
12	11(*-11)							
13	12(*-12)							
14	13(*-13)							
15	14(*-14)							

Input format: HEX

End   Cancel

[Display contents]

Set each switch setting value in combination with 16-bit binary data, setting the mode, console, etc.

Switch No.	Content		Remarks
Switch 1	Setting switch	Console setting	Refer to (1).
		Console, debugger type setting	
		Ctrl + C setting	
		Access possible time setting after Y reset setting	
		Flash ROM write protection setting	
		Debugger setting	
Switch 2	Mode setting switch		Refer to (2).
Switch 3	Not used (Set 0000H.)		—
Switch 4			
Switch 5			

(1)   Setting switches

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |

Bit position	Contents		OFF(0)	ON(1)
b0	Console	Channel setting	Refer to (a)	
b1				
b2		Software	Not used	Used
b3	Not used		Fixed on OFF	
b4	Console, debugger type		VT	VG
b5	Ctrl + C setting		Disabled	Enabled
b6	Not used		Fixed on OFF	
b7	Access possible time setting after Y reset		200 ms	2000 ms

(a)   Console (Channel setting)

Bit position		Contents
b1	b0	
OFF	OFF	Without console
OFF	ON	With console CH1
ON	OFF	With console CH2
ON	ON	Setting impossible

| bF | bE | bD | bC | bB | bA | b9 | b8 |

Bit position	Contents		OFF(0)	ON(1)
b8	Not used		Fixed on OFF	
b9	Flash ROM write protection setting		Without	With
bA	Not used		Fixed on OFF	
bB				
bC	Debugger	Channel setting	Refer to (b)	
bD				
bE		Software	Not used	Used
bF	Not used		Fixed on OFF	

(b)   Debugger (Channel setting)

Bit position		Contents
bD	bC	
OFF	OFF	Without a debugger
OFF	ON	With debugger CH1
ON	OFF	With debugger CH2
ON	ON	Setting impossible

1) Console
   Sets whether a console will be used or not, which channel (CH1/CH2) will be used, and whether software is to be used or not.
   • Channel setting : Sets which channel the console will be connected to. (Cannot be connected to CH3.)
   • Software
     Used   : PC/AT personal computer that uses SW1IVD-AD51HP-E.
     Not used   : VT-382, VG-620 (However, these have already gone out of production.)

2) Console, debugger type
   This sets the type of console and debugger.
   • VT   : PC/AT personal computer that uses SW1IVD-AD51HP-E.
                    VT-382 (However, this model is already out of production.)
   • VG   : VG-620 (However, this model is already out or production.)

3) Ctrl + C setting
   This sets whether the Ctrl + C key input from the console during multitask execution in the run mode is enabled or disabled.
   • Enabled   : At the point when the Ctrl + C keys are input, multitask operation is stopped and the system switches to the System mode.
   • Disabled   : Multitask execution continues even if the Ctrl + C key input occurs from the console.

4) Access possible time setting after Y reset
   This sets the length of time from the point when the Reset request signal (Y1A) goes ON until the QD51 (-R24) enters the reset operation.
   During the set time interval (from the point when the Reset request signal (Y1A) goes ON until the QD51 (-R24) enters the Reset operation), the FROM/TO command can be executed.

5) Flash ROM write protection setting
   Sets whether write protection of the Flash ROM is enabled or disabled.
   • With   : Sets write protection.
   • Without   : Cancels write protection.

6) Debugger

This sets whether the debugger will be used or not used, which channel (CH1/CH2) will be used, and whether software will be used or not used.

- Channel setting : Sets which channel the debugger is connected to. (Cannot be connected to CH3.)
- Software Used : PC/AT personal computer that uses SW1IVD-AD51HP-E

Not used : VT-382, VG-620 (However, these have already gone out of production.)

(2) Mode setting switch

Setting No.	Content	Remarks
0000$_H$	Run mode	Mode for executing the BASIC program.
0001$_H$		
0002$_H$	Multitask debug mode	Mode for debugging by connecting a debugger.
0003$_H$		
0004$_H$	Programming mode	Mode for programmimg by connecting a console.
0005$_H$ to 000F$_H$	Setting prohibited	—

5.6.3 Intelligent function module interrupt point settings

Here, the interrupt pointer settings in GX Developer for sending and receiving data with a BASIC program by an interrupt program are explained.

[Purpose of settings]

Intelligent function module interrupt pointer settings set information for starting a programmable controller CPU interrupt program from a BASIC program.
By carrying out these settings, it becomes possible to start an interrupt program from a BASIC program and transfer data to the programmable controller CPU.

[Start procedure]

[PC Parameter] → [PC System Settings] → | Interrupt Pointer Settings |

For the screen display method, refer to the GX Developer Operating Manual.

[Setting screen]

[Display contents]

(1)   CPU Side Interrupt Pointer Header No.
This sets the header No. (50 to 255) of the interrupt pointer (I_{xx}) used by the sequence program.

(2)   CPU Side Interrupt Pointer Count
This sets the number (1) of interrupt pointers (I) used in the sequence program.

(3)   Intelligent Function Module Side Header I/O No.
Sets the QD51 (-R24)'s header I/O No..

(4)   Intelligent Function Module Side Header SI No.

1)   Creates a correspondence between the QD51 (-R24)'s control No. (SI) and the interrupt pointer (I_{xx}) used in the sequence program.

2)   Sets the header SI No. (0) of the interrupt control No..

5.7 Maintenance, Checks

Except for the following confirmation items, there are no check items in particular for the QD51 (-R24).

Other than the following points, in order to be able to use the system in its optimum state at all times, please carry out maintenance in accordance with the check items listed in the programmable controller CPU User's Manual.

(QD51 (-R24) Check Items)

1) Make sure there are no faulty contacts in the terminal resistors and connection cables.

2) Make sure that the module fixing screws and the terminal block installation screws are fully tightened and that none of the screws is loose.

POINT
When maintaining and checking the QD51 (-R24), refer to the "Safety Precautions" included in the front of this manual.

6 EXCLUSIVE COMMANDS

Exclusive commands are commands which make programming easier when using the functions of the intelligent function module.

In this chapter, the QD51 (-R24) commands which are exclusive QCPU (Q Mode) commands described in this manual are explained.

6.1 Exclusive Command List and Available Devices

(1) Exclusive command list

The exclusive commands explained in this chapter are listed below.

Use	Exclusive Command	Function Outline	Explanation Section
For checking the rise of the Task Start Signal (Y19)	YCHECK	Checks the rise of the task start signal (Y19) and shortens the start time of a BASIC program.	Section 6.2

(2) Available devices

The following devices are available for the exclusive commands.

Internal devices		File register	Constant
Bit	Word		
-	T, ST, C, D, W	R, ZR	-

6

6.2 Z(P).YCHECK

This checks the rise of the task start signal (Y19) and shortens the BASIC program start time.

Setting Data	Usable Device								
	Internal Device (System, User)		File Register	Link direct device J□\□		Intelligent function module device U□\G□	Index Register Zn	Constant		Other
	Bit	Word		Bit	Word			K, H	$	
(S1)	—	○		—						

[Command Symbol]   [Execution Condition]

Z. YCHECK

```
        ┌─┐_       Command
   ─┘    └──       ──┤ ├──    [ Z.YCHECK  "Un"  (S1) ]
                              *1
```

ZP. YCHECK

```
        ─┐_        Command
          └──      ──┤ ├──    [ ZP.YCHECK  "Un"  (S1) ]
                              *1
```
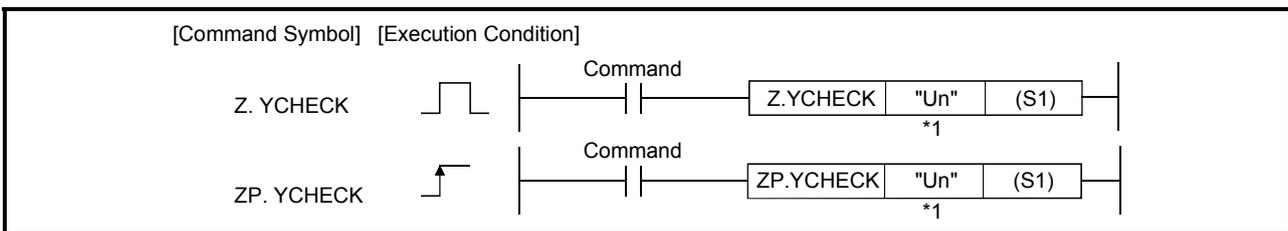
*1 If the originating station is a Basic model QCPU (function version B or later), or Universal model QCPU, " " (double quotation) of the first argument can be omitted.

Setting data

Setting Data	Content	Set Side	Data format
"Un"/Un	Module Header Input / Output Signal	User	String/BIN16 bit
(S1)	Dummy	—	Device name

The file register for each local device and program cannot be used as setting data.

Function:

(1) After turning ON QD51 (-R24) task start signal (Y19) designated at "Un," execute YCHECK instruction to start the BASIC program immediately.

(2) The YCHECK instruction is executed each scan if execution upon ON is selected and the instruction is ON. It is executed only once immediately after the command changes from OFF to ON if rising edge execution is selected.

Error:

(1) When a calculation error occurs, the error flag (SM0) goes ON and the error code is stored in SD0.
Refer to the QCPU User's Manual, check the error content and take the appropriate corrective measures.

Program Example:

Program to start the Task 1 BASIC program.

If the QD51 (-R24)'s input/output signal is X/Y00 to X/Y1F

```
    M1
0 ──┤ ├──────────────────────────────[SET    DY11  ]  Task 1 designated.

    ─────────────────────────────────[SET    DY19  ]  Program start signal turned ON.

    ──────────────────[ZP. YCHECK  "U0"  D0  ]  YCHECK executed.
```
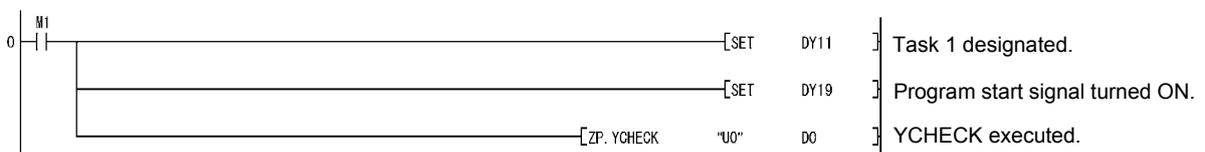
7 TROUBLESHOOTING

Here, the methods for checking the contents of errors that occur in the QD51 (-R24) are explained.
For error contents, refer to the Programming Manual (Commands).

7.1 About QD51 (-R24) Status Checks

7.1.1 Checking the LED lighting status, communications error status and QD51 (-R24) switch setting status

Here the method used to check the communications status between the QD51 (-R24) and an external device and the QD51 (-R24)'s switch setting status is explained.
When checking an error state due to the occurrence of trouble, etc., carry out the check by one of the following methods.

(1) If you are checking the module status and error code in the module detailed information screen of the GX Developer diagnostic function

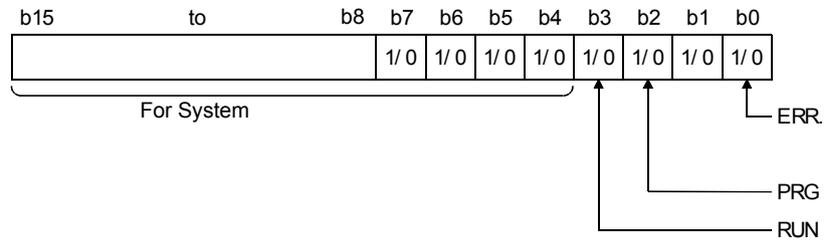[Start procedure]
GX Developer → [Diagnostics] → [System Monitor] →
| Module Detailed Information |

[Display contents]
- Module
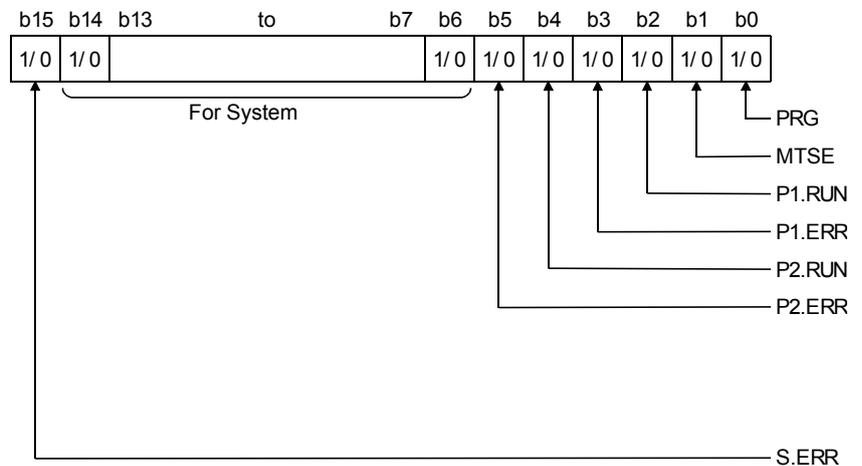 The following information is displayed.
Model name	: Installed module model name
Header I/O No.	: Relevant module's header Input / Output signal No.
Installation position	: The slot position where the module is installed.
Product information	: Product information

  ＊ The last character of the product information indicates the function version of the module.
  (Example) If the last character is "B," it means that the module is of function version B.

- Module access
  If the QD51 (-R24) System Down signal (X0D) is OFF and the QD51 (-R24) operating status signal (X0E) is ON, then Accessible is displayed.
- I/O address verification status
  This displays whether the module the user has set the parameters for matches the module that is installed or not.
- Latest error code
  Displays the error code for the most recent error that has occurred.
- Error display
  Displays the error codes for the error that have occurred with all the used functions as the object.

(2) If you are checking by the hardware information screen in the GX Developer diagnostic function

[Start Procedure]
GX Developer → [Diagnostics] → [System Monitor] →
| Module Detailed Information | → | Hardware Information |

H/W Information

Module
Module Name QD51   Product information 030920000000000 - B

Display format: ⊙ HEX ○ DEC

H/W LED Information

Item	Value
PRG	0001
MTSE	0000
P1.RUN	0000
P1.ERR	0000
P2.RUN	0000
P2.ERR	0000
S.ERR	0000

H/W SW Information

Item	Value
CONFIG	0025
MODE	0004

Start monitor   Stop monitor   Close
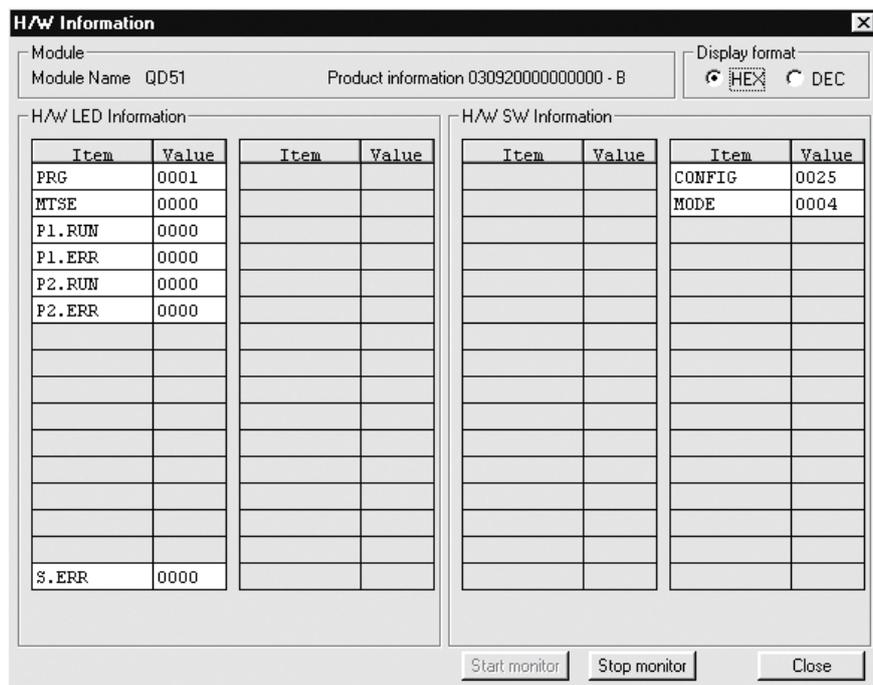
[Display contents]

1) H/W LED information package
Displays the display status of the QD51 (-R24)'s ERR LED, PRG LED and RUN LED.

b15	to	b8	b7	b6	b5	b4	b3	b2	b1	b0
			1/ 0	1/ 0	1/ 0	1/ 0	1/ 0	1/ 0	1/ 0	1/ 0

For System

b0 — ERR.
b2 — PRG
b3 — RUN

2) H/W LED information parameters
Displays the QD51 (-R24)'s A1SD51S compatible LED status.

b15	b14	b13	to	b7	b6	b5	b4	b3	b2	b1	b0
1/ 0	1/ 0				1/ 0	1/ 0	1/ 0	1/ 0	1/ 0	1/ 0	1/ 0

For System

b0 — PRG
b1 — MTSE
b2 — P1.RUN
b3 — P1.ERR
b4 — P2.RUN
b5 — P2.ERR
b15 — S.ERR

3) H/W SW information parameters
The QD51 (-R24)'s switch information is displayed.
No. 1 to 5 correspond to switches 1 to 5 shown in Section 5.6.2.

7.1.2 Errors that occur in BASIC

Refer to the Programming Manual (Commands), Appendix 4.4 concerning errors that occur during BASIC program execution.

APPENDIX

Appendix-1 Comparison of Functions with Previous Modules

A comparison of the QD51 (-R24)'s functions with the functions of previous modules is shown.

	Item	AD51H-S3	A1SD51S	QD51	QD51-R24
Cpu element		80C186 (15 MHz)		V53A (20 MHz)	
Arithmetic element		None			
Programming language		AD51H-BASIC (Interpreter, compiler)			
Number of tasks		8	2		
Task start conditions		• Start at power on • Start by interrupt from the programmable controller CPU (Impossible when compiled programs are run.) • Start by start request from another task.			
Internal memory	Program	384 k bytes (16/32/48/64 k bytes selectable for one task.)	Max. 64 k bytes (16/32/48/64 k bytes selectable for one task.)		
	Common memory	8 k bytes			
	Buffer memory	6 k bytes			
	Expansion relay (EM)	1024 points			
	Expansion register (ED)	1024 points			
General input/output		Input: 27 points Output: 17 points (8 points: For start from the programmable controller CPU)	Input: 27 points Output: 23 points (2 points: For task start from the programmable controller CPU)	Input: 26 points Output: 23 points (2 points: For task start from the programmable controller CPU)	
Memory protection		Yes (Memory card write protectable)	Yes (EEP-ROM write protectable)	Yes (Flash ROM write protectable)	
Interfaces	RS-232	2 channels		2 channels	1 channel
	RS-422/485	1 channel		—	1 channel
	Parallel	1 channel	—		
	Memory card	2 slots	—		
Clock function		None			
Power failure protection		Yes	Yes	None	
User program in ROM		Possible	Impossible However EEP-ROM is internal.	Impossible However, Flash ROM is internal.	
Console		• A7PHP • A7HGP • A7LMS • VG-620 • VT-382 • PC/AT personal computer		• PC/AT personal computer	
Multitask debugging		Possible (when a debugger is used.)			
I/o point share		48 points (The first 16 points are open and the last 32 points are special.)	32 points		
Slot share		2 slots	1 slot		

Appendix-2 About Use of Programs from Previous Modules

Here, use of programs created for A1SE51S or AD51H-S3 is explained.

POINT
In the QD51 (-R24), there are no switches for mode settings or console and other settings as in the previous modules (A1SD51S, etc.).
When the QD51 (-R24) is used, be sure to set the mode settings and console and other settings in the GX Developer parameter setting screen shown below. • "Intelligent Function Module Switch Settings" Screen

Appendix 2.1 About use of A1SD51S, AD51H-S3 programs

Here, use of programs created for the A1SD51S or the AD51H-S3 and incorporation of the QD51 (-R24) in existing systems is explained.
Programs created in A1SD51S or the AD51H-S3 can be read as is.
However, the points shown below differ slightly, so exercise caution.

(1) Usable channels

Usable channels are as shown below.

		AD51H-S3	A1SD51S	QD51	QD51-R24
Channel	CH. 1 (RS-232)	○	○	○	○
	CH. 2 (RS-232)	○	○	○	×
	CH. 3 (RS-422/RS-485)	○	○	×	○
	CH. 4 (Parallel)	○	×	×	×

○: Can be used ×: Cannot be used.

(a) Use in the QD51
If used in the QD51, there is no CH. 3 (RS-422/485) or CH. 4 interface in the QD51, so they cannot be used.
As for CH. 3 (RS-422/485), correct the BASIC program to another channel or use the QD51 -R24.
CH. 4 (Parallel) cannot be used.

(b) Use in the QD51 (-R24)
If used in the QD51, there is no CH. 2 (RS-232) or CH. 4 interface in the QD51, so they cannot be used.
As for CH. 2 (RS-232), correct the BASIC program to another channel or use the QD51.
CH. 4 (Parallel) cannot be used.

App

(2) Communications specifications

Data bit and stop bit settings differ.

Correct the BASIC program.

	AD51H-S3	A1SD51S	QD51(-R24)
Data bit	5, 6, 7, 8	5, 6, 7, 8	7, 8
Stop bit	1, 1.5, 2	1, 1.5, 2	1, 2

(3) Common memory backup

Common memory backup is as shown below.

	AD51H-S3	A1SD51S	QD51(-R24)
Common Memory	Backed up	Backed up	Not backed up

In the QD51 (-R24), common memory is not backed up by a battery. If backup is necessary, use the programmable controller CPU's file register (R).

Replace the part of the BASIC program, which is used for reading or writing the common memory by the GETMEN/PUTMEN command, with the one for reading or writing the file register (R) by the PCRD/PCWT command.

(4) Task start signal (Y19)

If the task start signal (Y19) has been used to start a task, add the exclusive command YCHECK.

In the A1SD51S or AD51H-S3, a task starts as an interrupt to the module when the task start signal (Y19) turns ON.

As the QD51 (-R24) monitors the task start signal (Y19) at 10ms intervals to start tasks, the start timing is different between the QD51(-R24) and the A1SD51S or AD51H-S3. Executing the dedicated YCHECK command as shown below makes the start timing the same as the A1SD51S or AD51H-S3.
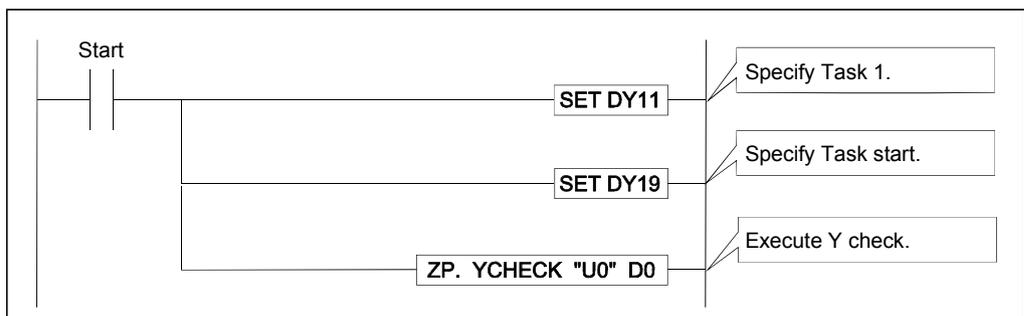
(5) Maximum number of tasks

The maximum number of tasks that can be run simultaneously is as shown below.

	AD51H-S3	A1SD51S	QD51(-R24)
Maximum number of tasks	8	2	2

In the QD51 (-R24), a maximum of 2 BASIC programs can be run at a time.

(6) PCRD/PCWT command

The processing codes shown below cannot be executed with respect to the Q/QnA CPU. Replace those codes which can be replaced.

(a) PCRD commands

Processing codes used in the A1SD51S/AD51H-S3		Treatment
Contents	Processing Code	
Monitoring of the device monitor register by the PCWT command	2(&H2)	Reading of Q/QnA series device memory Processing code 515 (&H203)
Reading of expansion file register data	4(&H4)	
Monitoring the expansion file register monitor registered by the PCWT command	5(&H5)	
Reading of continuous address designation data in the expansion file register	7(&H7)	
Reading of sequence programs	8(&H8)	Cannot be used in the Q/QnA Series Cannot be replaced
Reading of microcomputer programs	9(&H9)	
Reading of comment data	10(&HA)	
Reading of expansion comment data	11(&HB)	
Reading of the special function module buffer memory	12(&HC)	Reading of Q/QnA series intelligent function module buffer memory Processing code 533 (&H215)
Reading of programmable controller CPU model name	13(&HD)	Reading of Q/QnA series programmable controller CPU model name Processing code 513 (&H201)
Reading of parameter data (Reading of MELSECNET/10 parameters)	14(&HE)	Cannot be used in the Q/QnA Series Cannot be replaced
Reading of network information	21(&H15)	
Reading of routing parameters	22(&H16)	

(b) PCWT command

Processing codes used in the A1SD51S/AD51H-S3		Treatment
Contents	Processing Code	
Device memory monitor registration	2(&H2)	Writing to Q/QnA series device memory Processing Code 515 (&H203)
Random writing to device memory	3(&H3)	
Writing to expansion file register	4(&H4)	
Expansion file register monitor registration	5(&H5)	
Random writing to expansion file register	6(&H6)	
Continuous address designation data writing to expansion file register	7(&H7)	
Writing of sequence program	8(&H8)	Cannot be used in the Q/QnA Series Cannot be replaced
Writing of microcomputer program	9(&H9)	
Writing of comment data	10(&HA)	
Writing of expansion comment data	11(&HB)	
Writing to special function module buffer memory	12(&HC)	Writing to Q/QnA series intelligent function module buffer memory Processing code 533 (&H215)
Writing of parameter data (Writing of MELSECNET/10 parameters)	14(&HE)	Cannot be used in the Q/QnA Series Cannot be replaced
Analysis of parameter data	15(&HF)	

(7) Year of DATE$

The year setting of DATE$ varies.

	A1SD51S/AD51H-S3	QD51(-R24)	Countermeasure
Reading	Reads only the last two digits of the year. However, if the last two digits are 90 or more, the first two digits are 19. If the last two digits are less than 90, the first two digits are 20.	Reads the four-digit year as it is.	Modify the program to use four-digit year processing.
Writing	Writes only the last two digits of the year.	Writes the four-digit year as it is.	Designate the year as four digits correctly.

Appendix-3 External Dimensions

Outside cable diameter ×4 +10.

R2

90 (3.54)　4.3 (0.17)

QD51
RUN　ERR.
PRG　P RUN
CH.1 [SD / RD]　[SD / RD] CH.2

CH.1
RS-232

CH.2
RS-232

QD51

98 (3.86)

27.4 (1.08)

QD51-R24
RUN　ERR.
PRG　P RUN
CH.1 [SD / RD]　[SD / RD] CH.3

CH.1
RS-232

R1

SDA
SG
SDB
(FG)
RDA
(FG)
RDB
CH.3
RS-422
/485

r1

1
2
3
4
5
6
7

98 (3.86)

90 (3.54)　4.3 (0.17)

27.4 (1.08)
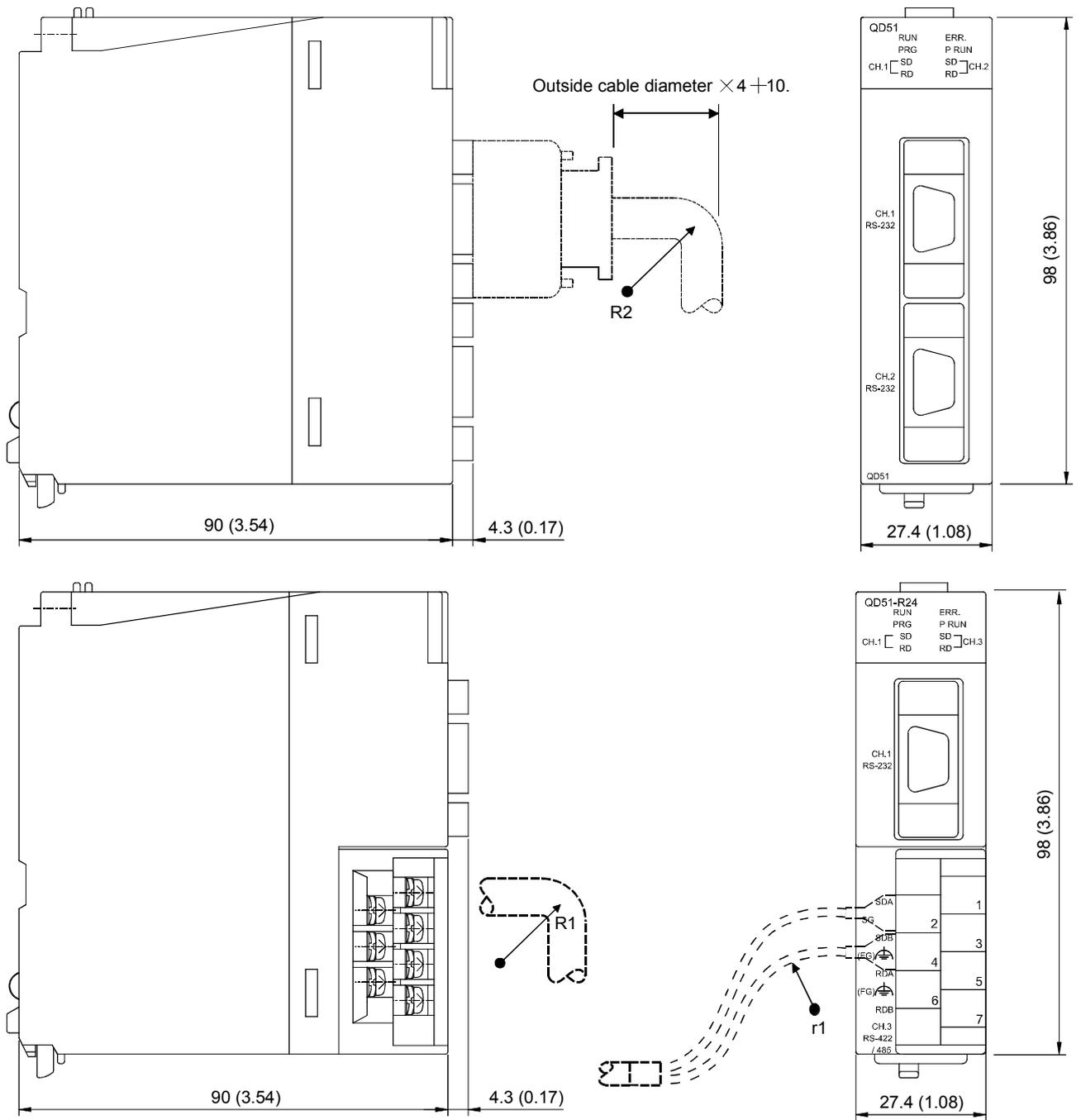
(Unit : mm (in.))

∗R1 (Bending radius near the terminal block)　: Outside cable diameter × 4
∗R2 (Bending radius near the connector)　: Outside cable diameter × 4
∗ r1 (Bending radius near crimp contact)　: Connectable as long as not bended
extremely

MEMO

INDEX

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.

Q Corresponding Intelligent Communication Module

User's Manual

MODEL	QD51-U-SHOSAI-E
MODEL CODE	13JR16
SH(NA)-080089-G(0808)MEE	

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.