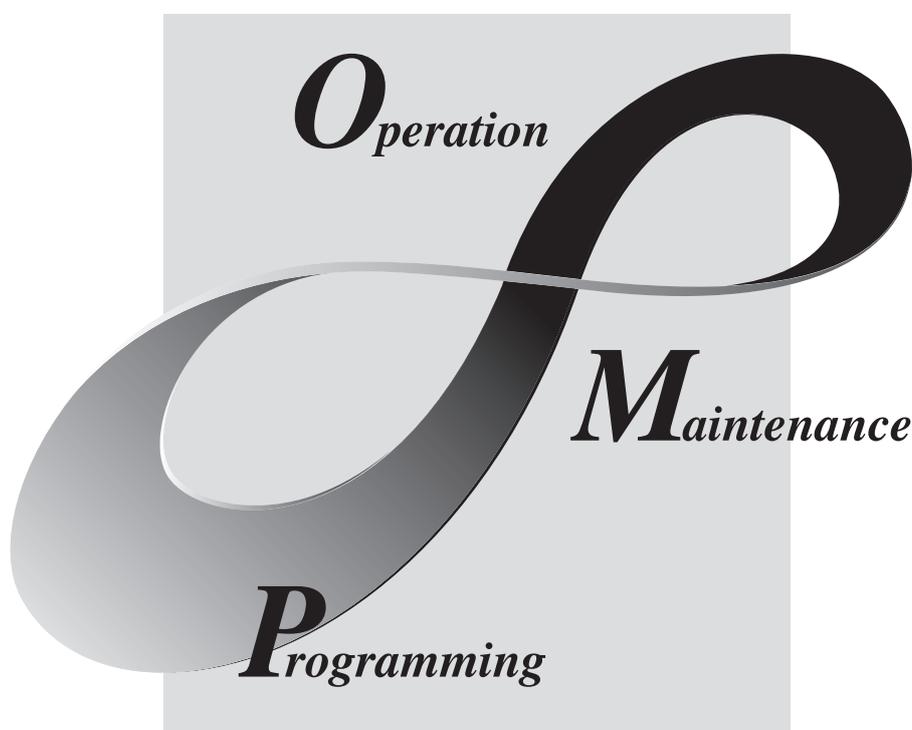


*GX Developer Version 8*

Operating Manual

(Structured Text)

**mitsubishi**



**MELSOFT**  
**Integrated FA Software**

**SW8D5C-GPPW-E**



# • SAFETY PRECAUTIONS •

(Always read these instructions before using this product.)

Before using this product, thoroughly read this manual and the relevant manuals introduced in this manual and pay careful attention to safety and handle the products properly.

The precautions given in this manual are concerned with this product. For the safety precautions of the programmable controller system, refer to the User's Manual for the CPU module.

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 minor or moderate injury or property damage.

Note that the ⚠ CAUTION level may lead to serious consequences according to the circumstances. Always follow the precautions of both levels because they are important for personal safety.

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

## [Designing Precautions]

### ⚠ DANGER

- When data change, program change, or status control is performed from a personal computer to a running programmable controller, create an interlock circuit outside the programmable controller to ensure that the whole system always operates safely.

Furthermore, for the online operations performed from a personal computer to a programmable controller CPU, the corrective actions against a communication error due to such as a cable connection fault should be predetermined as a system.

## [Startup/Maintenance Precautions]

### ⚠ CAUTION

- The online operations performed from a personal computer to a running programmable controller CPU (program change when a programmable controller CPU is RUN, forced I/O operation, operating status change such as RUN-STOP switching, and remote control operation) have to be executed after the manual has been carefully read and the safety has been ensured.

When changing a program while a programmable controller CPU is RUN (Online program change), it may cause a program corruption in some operating conditions. Fully understand the precautions described in GX Developer Operating Manual before use.

REVISIONS

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

| Print Date | * Manual Number   | Revision   |
|------------|-------------------|--|
| Feb., 2003 | SH (NA) 080367E-A | First printing   |
| Oct., 2003 | SH (NA) 080367E-B | Correction<br>Section 3.5.5, Section 4.6   |
| Jun., 2004 | SH (NA) 080367E-C | Correction<br>Abbreviations and Generic Terms in This Manual, Section 1.5.1                                  |
| May, 2008  | SH (NA) 080367E-D | Correction<br>Abbreviations and Generic Terms in This Manual, Section 1.5.1                                  |
| Dec., 2008 | SH (NA) 080367E-E | Correction<br>About Manuals, Abbreviations and Generic Terms in This Manual,<br>Section 1.5.1, Section 4.3.2 |
|            |                   |  |

Japanese Manual Version SH-080364-G

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2003 MITSUBISHI ELECTRIC CORPORATION

## INTRODUCTION

Thank you for choosing the Mitsubishi MELSOFT series Integrated FA software.  
Read this manual and make sure you understand the functions and performance of MELSEC series programmable controller thoroughly in advance to ensure correct use.  
Please make this manual available to the end user.

## CONTENTS

|   |      |
|---|------|
| SAFETY PRECAUTIONS.....                             | A- 1 |
| REVISIONS .....                                     | A- 2 |
| CONTENTS.....                                       | A- 3 |
| About Manuals .....                                 | A- 5 |
| How to Use This Manual.....                         | A- 6 |
| Abbreviations and Generic Terms in This Manual..... | A- 7 |

|                    |                     |
|--------------------|---------------------|
| <b>1. OVERVIEW</b> | <b>1- 1 to 1- 9</b> |
|--------------------|---------------------|

|  |      |
|--|------|
| 1.1 What Is the ST Language? .....                           | 1- 1 |
| 1.2 Features .....   | 1- 3 |
| 1.3 Installation .....                                       | 1- 4 |
| 1.3.1 Installation method .....                              | 1- 4 |
| 1.3.2 Operating environment.....                             | 1- 5 |
| 1.4 Screen Display and Names for Creating ST Programs.....   | 1- 6 |
| 1.4.1 ST edit screen .....                                   | 1- 6 |
| 1.5 Specifications .....                                     | 1- 9 |
| 1.5.1 Corresponding programmable controller CPUs.....        | 1- 9 |
| 1.5.2 Specifications and precautions for ST edit screen..... | 1- 9 |

|   |                     |
|---|---------------------|
| <b>2. ST PROGRAM CREATION PROCEDURE</b> | <b>2- 1 to 2- 2</b> |
|---|---------------------|

|                          |                     |
|--------------------------|---------------------|
| <b>3. ST PROGRAMMING</b> | <b>3- 1 to 3-34</b> |
|--------------------------|---------------------|

|   |      |
|---|------|
| 3.1 Creating a New Project.....                           | 3- 1 |
| 3.2 Entering an ST Program .....                          | 3- 3 |
| 3.2.1 Entering a function .....                           | 3- 4 |
| 3.2.2 Entering a label.....                               | 3- 6 |
| 3.2.3 Creating a comment.....                             | 3- 8 |
| 3.2.4 Control syntax upper case conversion function ..... | 3- 9 |
| 3.2.5 Auto indent function.....                           | 3-10 |
| 3.3 Useful Edit Functions .....                           | 3-11 |
| 3.3.1 Using the bookmark .....                            | 3-12 |
| 3.3.2 Displaying a function parameter .....               | 3-15 |
| 3.3.3 Window division.....                                | 3-16 |
| 3.3.4 Displaying the label information.....               | 3-17 |
| 3.3.5 Find/Replace .....                                  | 3-18 |

|  |      |
|--|------|
| 3.3.6 Line jump .....                          | 3-23 |
| 3.3.7 Open Function Block.....                 | 3-24 |
| 3.3.8 Copy/Cut/Paste .....                     | 3-26 |
| 3.3.9 Undo/Redo .....                          | 3-27 |
| 3.4 Performing Convert (Compile).....          | 3-28 |
| 3.5 Customizing the ST Edit Screen.....        | 3-31 |
| 3.5.1 Changing the auto indent/tab width ..... | 3-31 |
| 3.5.2 Changing the display colors.....         | 3-32 |
| 3.5.3 Changing the display font .....          | 3-34 |

|                  |                     |
|------------------|---------------------|
| <b>4. ONLINE</b> | <b>4- 1 to 4-20</b> |
|------------------|---------------------|

|  |      |
|--|------|
| 4.1 Read from PLC .....                                      | 4- 1 |
| 4.2 Write to PLC .....                                       | 4- 3 |
| 4.3 Monitoring the ST Program.....                           | 4- 5 |
| 4.3.1 Monitoring the ST program .....                        | 4- 5 |
| 4.3.2 Troubleshooting at error occurrence in ST program..... | 4- 7 |
| 4.4 Online Change .....                                      | 4-11 |
| 4.5 Device Test.....   | 4-13 |
| 4.6 Debug Function.....                                      | 4-15 |
| 4.6.1 Debug function flowchart .....                         | 4-15 |
| 4.6.2 Starting/Ending debug function.....                    | 4-16 |
| 4.6.3 Setting/Clearing break points.....                     | 4-17 |
| 4.6.4 Break execution/1 line execution .....                 | 4-19 |
| 4.6.5 Break point list .....                                 | 4-20 |
| 4.6.6 Clearing all break points.....                         | 4-20 |

|                 |                     |
|-----------------|---------------------|
| <b>5. PRINT</b> | <b>5- 1 to 5- 3</b> |
|-----------------|---------------------|

|              |                             |
|--------------|-----------------------------|
| <b>INDEX</b> | <b>Index- 1 to Index- 2</b> |
|--------------|-----------------------------|

## About Manuals

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

### Relevant Manuals

| Manual Name   | Manual Number<br>(Model Code) |
|---|-------------------------------|
| GX Developer Version 8 Operating Manual (Startup)<br>Explains the system configuration, installation method and startup method of GX Developer.<br>(Option)   | SH-080372E<br>(13JU40)        |
| GX Developer Version 8 Operating Manual<br>Explains the program creation method, printout method, monitoring method, debugging method, etc.<br>using GX Developer.<br>(Option)  | SH-080373E<br>(13JU41)        |
| GX Developer Version 8 Operating Manual (Function Block)<br>Explains the program creation method, printout method, etc. using GX Developer.<br>(Option)   | SH-080376E<br>(13JU44)        |
| Structured Text (ST) Programming Guide Book<br>Written for those who will create structured text (ST) programs for the first time. Explains the basic<br>operation methods and functions through sample programs.<br>(Option) | SH-080368E<br>(13JF69)        |
| QCPU (Q mode) Programming Manual (Structured Text)<br>Explains the programming methods in structured text language.<br>(Option)   | SH-080366E<br>(13JF68)        |
| QCPU Programming Manual(Common Instructions)<br>Explains the methods of using the sequence instructions, basic instructions and application instructions.<br>(Option)   | SH-080809ENG<br>(13JW10)      |
| GX Simulator Version 6 Operating Manual<br>Explains the setting and operation for using GX Simulator to monitor device memory and to simulate the<br>machine operation.<br>(Option)   | SH-080169<br>(13JU17)         |

### REMARK

Each Operating Manual and the Structured Text (ST) Programming Guidebook are contained in the CD-ROM together with the software package as a set.  
The Programming Manual is available separately in printed form as an option.  
Please place an order with the manual number (model code) in the above table.

## How to Use This Manual

### *This Manual ...*

This manual is a commentary that gives in-depth explanation of the operation methods to create structured text (ST) programs using GX Developer. Refer to this manual when information on operation details is necessary.

"Chapter 1 Overview" describes the outline of the structured text (ST) language, the installation method, the screen display and names for creating structured text (ST) programs, the corresponding programmable controller CPUs, and others.

"Chapter 2 ST Program Creation Procedure" describes a structured text (ST) program creation procedure in a flowchart.

"Chapter 3 ST Programming" describes how to create a new structured text (ST) program, how to perform operations of editing functions useful for input, and others.

"Chapter 4 Online" describes the procedure for writing the created structured text (ST) program to the programmable controller CPU, the device test operation method, and others.

"Chapters 5 Print" describes the printing operation procedure, etc.

The following explains the symbols and information used in this manual.

| Symbol  | Description   | Example   |
|---|---|---|
| Point   | Gives the section-related knowledge and useful information. |  |
| [ ]   | Menu name of menu bar                                       | [Project]   |
| ( )   | Icon of toolbar   |  |
| << >>   | Tab name of dialog box                                      | <<Select file>>   |
|  | Command button of dialog box                                |  |

### *Programming Manual ...*

Use the "QCPU (Q mode) Programming Manual (Structured Text)" to perform structured text (ST) programming with GX Developer. It is suitable for the users who have the knowledge and programming experience of programmable controller ladder programs and for the users who have the knowledge and programming experience of high-level languages such as the C language.

### *When using the structured text language for the first time ...*

Refer to the "Structured Text (ST) Programming Guidebook", which describes the outline of the structured text (ST) language, the procedures for creating a structured text (ST) program using GX Developer and writing it to the programmable controller CPU, the information necessary for that purpose, and others.

### *When information on other than structured text programming is necessary ...*

Refer to the "GX Developer Version 8 Operating Manual" or "GX Developer Version 8 Operating Manual (Startup)".

## Abbreviations and Generic Terms in This Manual

In this manual, the following generic terms and abbreviations are used to represent the GX Developer software package and programmable controller CPU. The package name is given when the target model name must be pointed out explicitly.

| Generic terms and abbreviations | Description and target module  |
|---------------------------------|--|
| ST                              | Abbreviation for structured text.  |
| GX Developer                    | Generic product name for model names SWnD5C-GPPW, SWnD5C-GPPW-A, SWnD5C-GPPW-V and SWnD5C-GPPW-VA.<br>n means Version 8 or later.  |
| FB                              | Abbreviation for function block.   |
| Basic model QCPU                | Generic term for Q00JCPU, Q00CPU and Q01CPU of function version B or later   |
| High Performance model QCPU     | Generic term for Q02 (H) CPU, Q06CPU, Q12HCPU and Q25HCPU  |
| Universal model QCPU            | Generic term for Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q04UDHCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU and Q26UDEHCPU.                                    |
| Process CPU                     | Generic term for Q02PHCPU, Q06PHCPU, Q12PHCPU and Q25PHCPU   |
| Redundant CPU                   | Generic term for Q12PRHCPU and Q25PRHCPU   |
| QCPU (Q mode)                   | Generic term for Q00J, Q00UJ, Q00, Q00U, Q01, Q01U, Q02(H), Q02PH, Q02U, Q03UD, Q03UDE, Q04UDH, Q04UDEH, Q06H, Q06PH, Q06UDH, Q06UDEH, Q10UDH, Q10UDEH, Q12H, Q12PH, Q12PRH, Q13UDH, Q13UDEH, Q20UDH, Q20UDEH, Q25H, Q25PH, Q25PRH, Q26UDH and Q26UDEHCPU. |

## 1 OVERVIEW

1

This manual explains the editing operation for the structured text (hereafter abbreviated to ST) of the GX Developer Version 8 software package (hereafter abbreviated to GX Developer).

For the explanation of the functions in other than ST, refer to the corresponding manuals given in "Relevant Manuals".

## 1.1 What Is the ST Language?

The ST language is defined in the International Standard IEC61131-3 that stipulates the logic description system in open controllers.

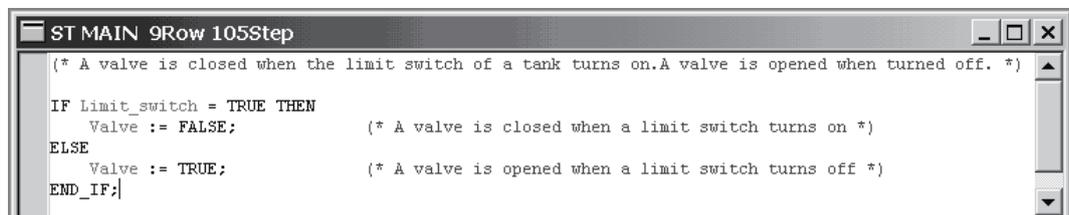
The ST language supports operators, control syntaxes and functions to permit the following descriptions.

- Control syntaxes such as conditional sentence-dependent selective branch and repetitive sentence-based repetition
- Expressions using operators (\*, /, +, -, <, >, =, etc.)
- Call of user-defined function blocks (FB)
- Call of functions (MELSEC functions, IEC functions)
- Description of comments

The main features of the ST language are as described below.

## (1) Free description in text format

The ST language allows the description of alphanumeric characters, comments and labels in text format.



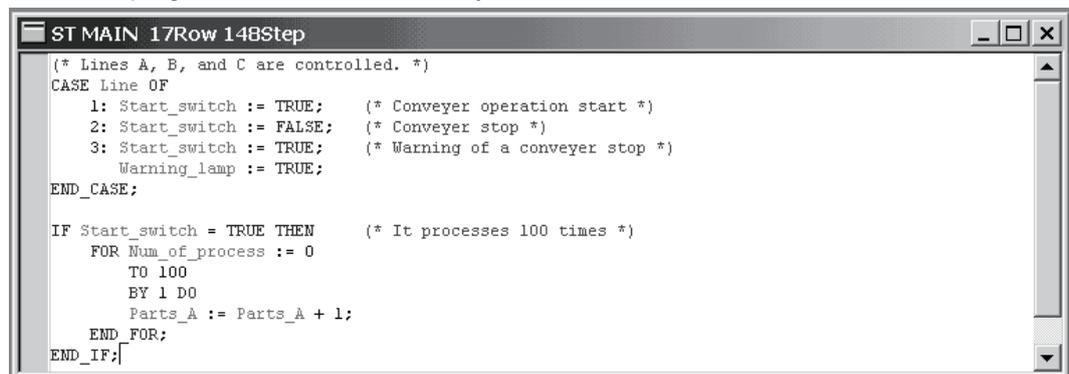
```

ST MAIN 9Row 105Step
(* A valve is closed when the limit switch of a tank turns on.A valve is opened when turned off. *)
IF Limit_switch = TRUE THEN
  Valve := FALSE;          (* A valve is closed when a limit switch turns on *)
ELSE
  Valve := TRUE;           (* A valve is opened when a limit switch turns off *)
END_IF;

```

## (2) Programming on the same level as those of the C and other high-level languages

Like the high-level languages such as C, the ST language can describe control with control syntaxes such as conditional sentence-dependent selective branches and repetitive sentence-based repetitions. Hence, easy-to-read programs can be written briefly.



```

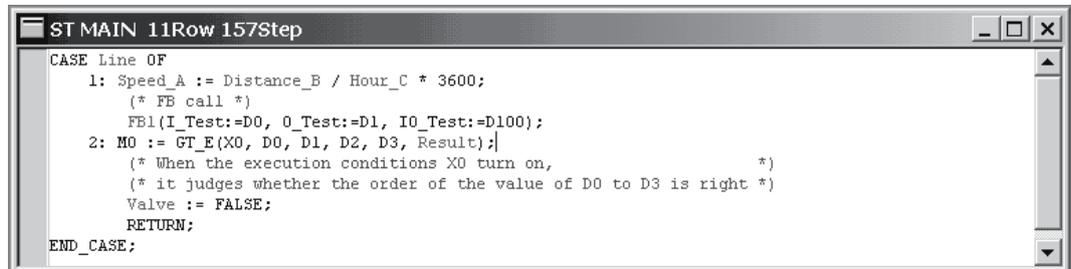
ST MAIN 17Row 148Step
(* Lines A, B, and C are controlled. *)
CASE Line OF
  1: Start_switch := TRUE;    (* Conveyor operation start *)
  2: Start_switch := FALSE;   (* Conveyor stop *)
  3: Start_switch := TRUE;    (* Warning of a conveyor stop *)
    Warning_lamp := TRUE;
END_CASE;

IF Start_switch = TRUE THEN  (* It processes 100 times *)
  FOR Num_of_process := 0
    TO 100
    BY 1 DO
    Parts_A := Parts_A + 1;
  END FOR;
END_IF;

```

### (3) Ease of describing operation processings

Capable of briefly describing easy-to-read operation processings that are difficult to describe in lists or ladders, the ST language has a high level of program readability and is suitable for the fields where complex arithmetic operations, comparison operations, etc. are performed.



```
ST MAIN 11Row 157Step
CASE Line OF
1: Speed_A := Distance_B / Hour_C * 3600;
   (* FB call *)
   FB1(I_Test:=D0, O_Test:=D1, IO_Test:=D100);
2: MO := GT_E(X0, D0, D1, D2, D3, Result);
   (* When the execution conditions X0 turn on, *)
   (* it judges whether the order of the value of D0 to D3 is right *)
   Valve := FALSE;
   RETURN;
END_CASE;
```



- ST programs assume that labels will be used.
- Please understand how to use labels in advance.

## 1.2 Features

ST programs are described in ST language.

Creating ST programs using GX Developer enables efficient programming to be performed in excellent operation environment.

The following provides the main features of ST programs in the MELSEC-Q series.

**(1) Design efficiency improved by defining processings as parts**

With often used processings defined as parts in the form of function blocks (FB) in ST language, they can be used in necessary areas of each program. This not only enhances the efficiency of program development but also reduces program mistakes, improving program quality.

For more information, refer to the "GX Developer Operating Manual (Function Block)" given in Relevant Manuals.

**(2) Program change during system operation (online change)**

Part of a running program can be changed without the programmable controller CPU being stopped.

**(3) Connection with other language programs**

Since the MELSEC-Q series also supports languages other than the ST, the language adequate for processing can be used to increase the efficiency of program development.

The High Performance model QCPU and Process QCPU allow execution conditions to be set on a file basis, and multiple program files to be written to a single programmable controller CPU.

Multiple languages support widespread application under optimum control.

**(4) A wealth of functions available**

The MELSEC functions compatible with various common instructions for the MELSEC-Q series and the IEC functions defined in IEC61131-3 are available for ST programs in the MELSEC-Q series.

For more information, refer to the "QCPU (Q mode) Programming Manual (Structured Text)" given in Relevant Manuals.

## 1.3 Installation

This section explains the installation method and operation environment necessary for creating ST programs.

### 1.3.1 Installation method

For programming in structured text (ST) language, install GX Developer in the following procedure.

- 1) Select [Start] - [Explorer] on Windows® to start.
- 2) Click the drive where the CD-ROM has been inserted.  
Double-click "Setup.exe".
- 3) Make setting and selection in the procedure of the installation wizard.
- 4) Check the "ST (Structured Text) language programming function" check box, and execute installation.



#### REMARK

For details, refer to the "GX Developer Operating Manual (Startup)" given in Relevant Manuals.

### 1.3.2 Operating environment

For operating environment when using the ST language programming function, refer to "GX Developer Operating Manual (Startup)" listed in Relevant Manuals while pay attention to the items shown in POINT below.



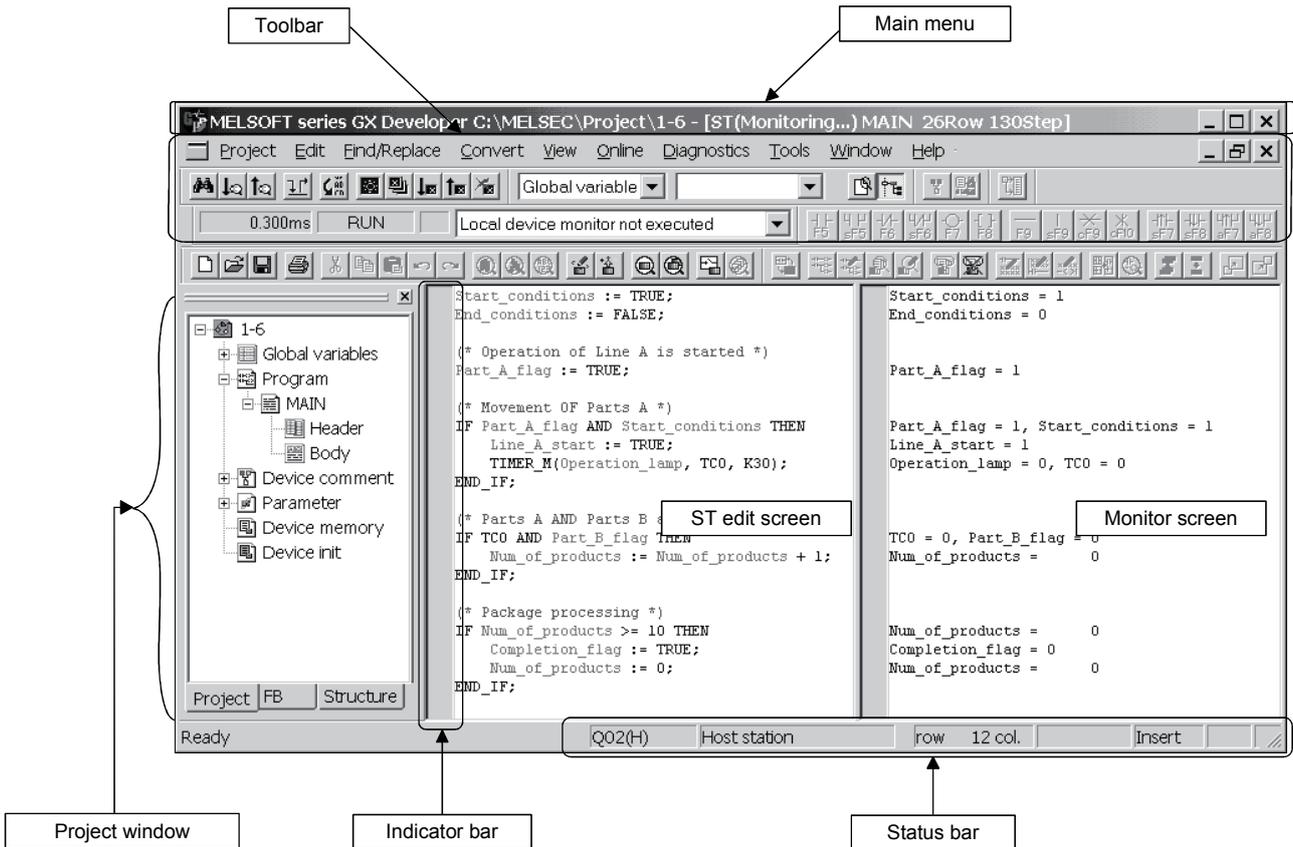
- The ST language function is not supported on Windows 95.
- A CPU of Pentium® II 450MHz or more is required.

### 1.4 Screen Display and Names for Creating ST Programs

This section explains the basic information such as the screen display, function list and key operations.

#### 1.4.1 ST edit screen

##### (1) Screen display and part names of main window



##### (2) Part names and functions

| Name           | Function  |
|----------------|---|
| Main menu      | Select the menu item.                             |
| Toolbar        | Clicking the selected icon executes the function. |
| Project window | Programs and various data are managed.            |
| ST edit screen | Screen for editing an ST program.                 |
| Monitor screen | Displays the condition of the executed program.   |
| Indicator bar  | Displays the condition during editing.            |
| Status bar     | Displays the cursor position on the edit screen.  |
|                | Displays the cursor mode on the program screen.   |

## (3) About the shortcut keys and toolbar

## 1) Shortcut keys

The shortcut keys are assigned to enable menu item selection and instruction input from the keyboard.

## 2) Toolbar

Displays the menu items with icons.

Whether the toolbar is displayed or hidden can be specified by choosing [View] - [Toolbar].

<List of shortcut keys and toolbar icons used mainly on ST edit screen>

| General                   | Shortcut Keys | Toolbar   | Mouse Right-click |
|---------------------------|---------------|---|-------------------|
| Move to first line        | Ctrl + Home   | —   | —                 |
| Move to last line         | Ctrl + End    | —   | —                 |
| All select                | Ctrl + A      | —   | —                 |
| Print                     | Ctrl + P      |    | —                 |
| Cut                       | Ctrl + X      |    | ○                 |
| Copy                      | Ctrl + C      |    | ○                 |
| Paste                     | Ctrl + V      |  | ○                 |
| Undo                      | Ctrl + Z      |  | ○                 |
| Redo                      | Ctrl + Y      |  | ○                 |
| Writing to PLC            | —             |  | —                 |
| Registered device monitor | —             |  | —                 |
| Device batch monitor      | —             |  | —                 |
| Check parameter           | —             |  | —                 |
| Select function           | Shift + F11   | —   | ○                 |
| Select label              | F11           | —   | ○                 |
| Project data list         | Alt + 0       | —   | —                 |
| Find                      | Ctrl + F      |  | ○                 |
| Find downward             | F5            |  | —                 |
| Find upward               | Shift + F5    |  | —                 |
| Replace                   | Ctrl + H      |  | ○                 |
| Line jump                 | Ctrl + J      |  | —                 |
| Bookmark setting          | Ctrl + F7     |  | ○                 |
| Bookmark list             | —             |  | —                 |
| Bookmark downward         | F7            |  | —                 |
| Bookmark upward           | Shift + F7    |  | —                 |
| Delete all bookmark       | —             |  | —                 |
| Convert/compile           | F4            |  | —                 |

| General                                     | Shortcut Keys   | Toolbar   | Mouse Right-click |
|---|-----------------|---|-------------------|
| Convert/Compile (all programs being edited) | Ctrl + Alt + F4 |  | —                 |
| Move to last line                           | Shift + F4      | —   | —                 |
| Start monitor (all windows)                 | Ctrl + F3       | —   | —                 |
| Stop monitor (all windows)                  | Ctrl + Alt + F3 | —   | —                 |
| Start monitor                               | F3              |  | —                 |
| Stop monitor                                | Alt + F3        |  | —                 |
| Device test                                 | Alt + 1         |  | ○                 |
| Remote operation                            | Alt + 6         | —   | —                 |

## 1.5 Specifications

This section explains the specifications for use of ST programs on GX Developer.

### 1.5.1 Corresponding programmable controller CPUs

The following models of programmable controller CPU are applicable to ST programs.

| Basic model QCPU            | High Performance model QCPU                        | Universal model QCPU  | Process CPU                                  | Redundant CPU          |
|-----------------------------|--|---|--|------------------------|
| Q00CPU<br>Q00JCPU<br>Q01CPU | Q02CPU<br>Q02HCPU<br>Q06HCPU<br>Q12HCPU<br>Q25HCPU | Q00UJCPU<br>Q00UCPU<br>Q01UCPU<br>Q02UCPU<br>Q03UDCPU<br>Q03UDECPU<br>Q04UDHCPU<br>Q04UDEHCPU<br>Q06UDHCPU<br>Q06UDEHCPU<br>Q10UDHCPU<br>Q10UDEHCPU<br>Q13UDHCPU<br>Q13UDEHCPU<br>Q20UDHCPU<br>Q20UDEHCPU<br>Q26UDHCPU<br>Q26UDEHCPU. | Q02PHCPU<br>Q06PHCPU<br>Q12PHCPU<br>Q25PHCPU | Q12PRHCPU<br>Q25PRHCPU |

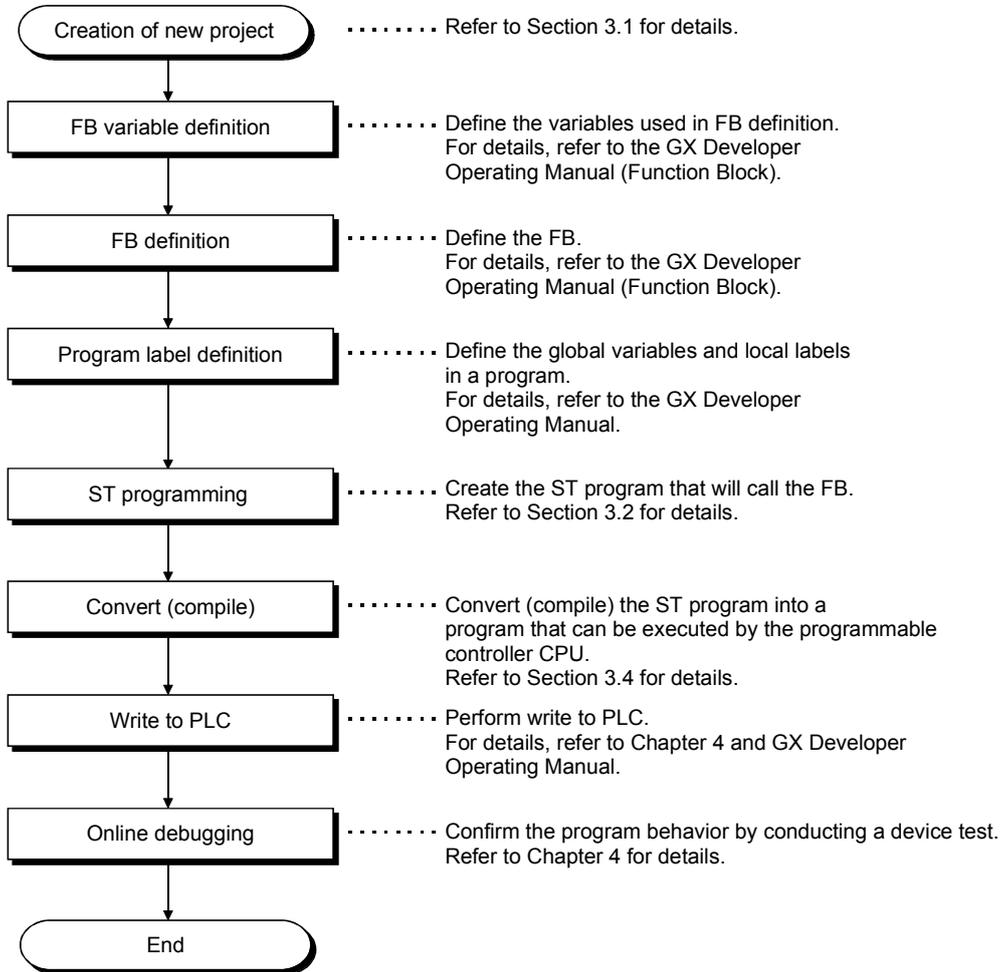
### 1.5.2 Specifications and precautions for ST edit screen

There are the following restrictions on the character input of the ST edit screen.

|  |  |
|--|--|
| Maximum number of characters               | 839680 characters (Two characters are used as the line feed code.) |
| Maximum number of columns (display region) | 999 characters   |
| Maximum number of lines                    | 65535 lines  |

## 2 ST PROGRAM CREATION PROCEDURE

The following flowchart indicates the general procedure of ST programming. In the following example, parts were created with the function block function and a main program was then created in ST language.



**Point**

- This general procedure is for reference. Each operation can be performed in any order.
- FB definition can be described in ST language or ladder form.



### 3 ST PROGRAMMING

This chapter explains the creation and editing methods to create a project using an ST program.

#### 3.1 Creating a New Project

This section explains the method of creating a new project.

**[Purpose]**

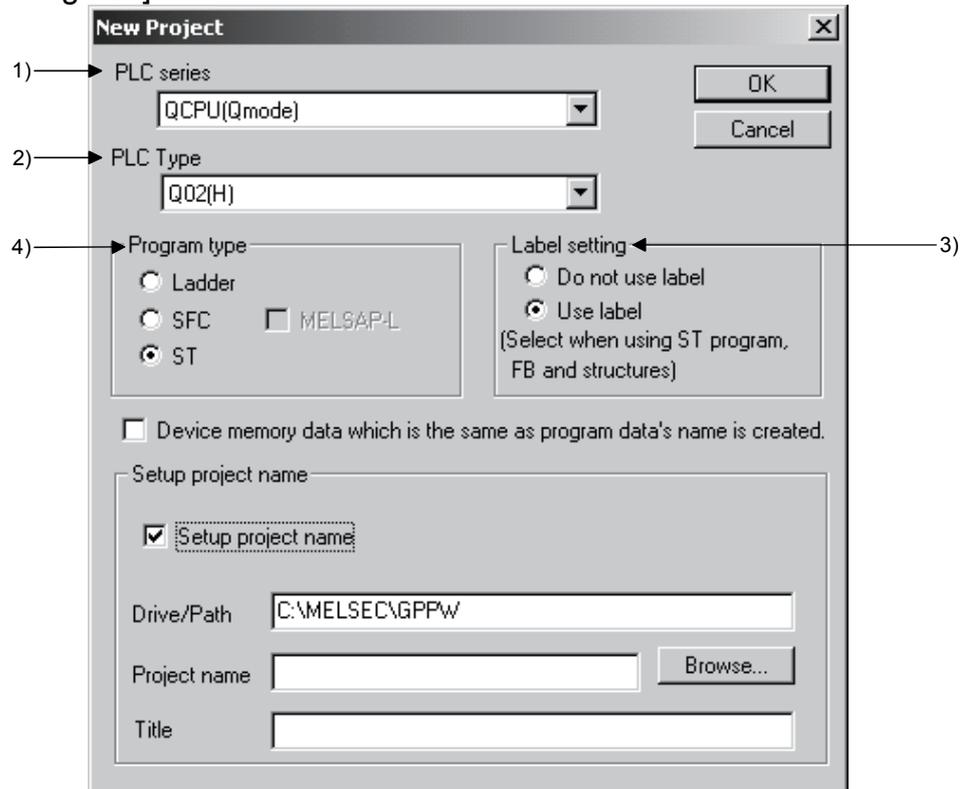
Set the PLC series, PLC type, label setting, program type and project name required to create a new project.



**[Operating Procedure]**

Choose [Project] → [New project], click (  ), or press [Ctrl] + [N].

**[Dialog Box]**



**[Description]**

1) PLC series

Select the PLC series.

There are the following PLC series.

- QCPU(Q mode)
- QnACPU series
- Motion(SCPU)
- QCPU(A mode)
- ACPU series
- FXCPU series

## 2) PLC Type

Select the programmable controller CPU type to be used.

## 3) Label setting

Make this setting when creating a label program.

## 4) Program type

Set the program type to be created.

To select "ST", "Use label" must be set in the label setting.

**[Setting procedure]**

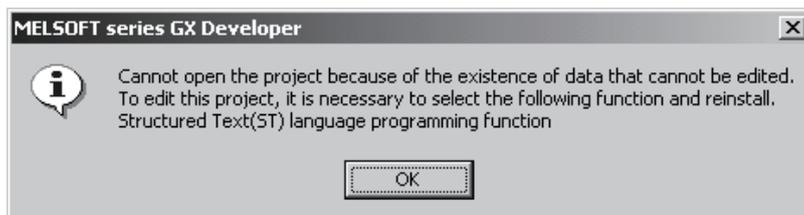
- i) As the PLC series, set the QCPU (Q mode).
- ii) As the PLC type, set the programmable controller CPU to be used.
- iii) As the label setting, set "Use label".
- iv) As the program type, set "ST".
- v) Set "Setup project name" and click the **OK** button to create a new project.  
"Setup project name" can either be set before or after program creation.

**REMARK**

The precautions for reading and copying the project will be explained.

## ● Reading the project

If the project that includes the ST program is read using GX Developer Version 7 or earlier where the ST language function is not installed, the following message is displayed and the project cannot be read.



## ● Copying the project

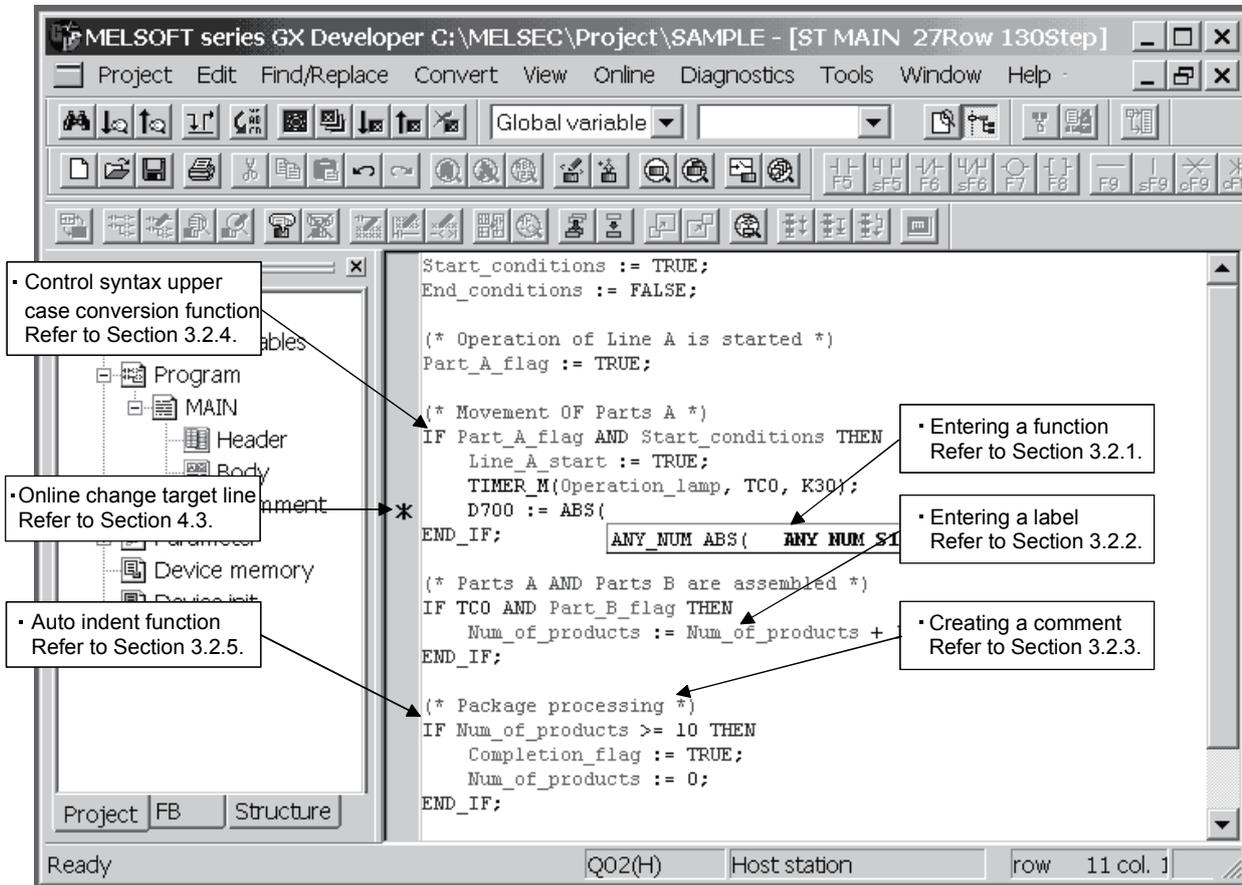
When the project is copied, the copied ST program and FB are in a not yet converted (not yet compiled) condition.

After copying, perform convert (compile) again.

For details, refer to the "GX Developer Operating Manual" given in Relevant Manuals.

### 3.2 Entering an ST Program

The ST edit screen allows free editing operation to be performed like a general text editor. This section introduces the functions useful for input.



3.2.1 Entering a function

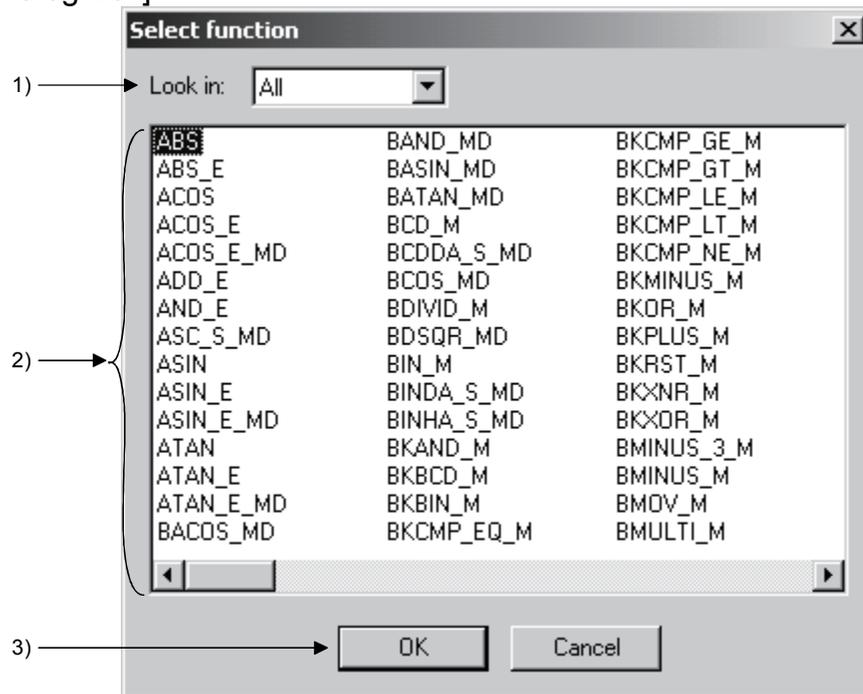
[Purpose]

A function can be entered directly. If a function name is unknown, the function selection function can be used for input.

[Operating Procedure]

Choose [Edit] → [Select function] or press **[Shift] + [F11]**.

[Dialog Box]



[Description]

1) Function classification list box

The following table indicates the classification of the functions that can be selected.

| Classification   | Description  |
|------------------|--|
| All              | All MELSEC functions and IEC functions are displayed in the function list box in order (ascending order) of names. |
| MELSEC functions | All MELSEC functions are displayed in the function list box in order (ascending order) of names.                   |
| IEC functions    | All IEC functions are displayed in the function list box in order (ascending order) of names.                      |

**REMARK**

For the MELSEC functions and IEC functions, refer to the "QCPU (Q mode) Programming Manual (Structured Text)".

2) Function list box

The function list selected in the function classification list box is displayed.

3) **OK** button

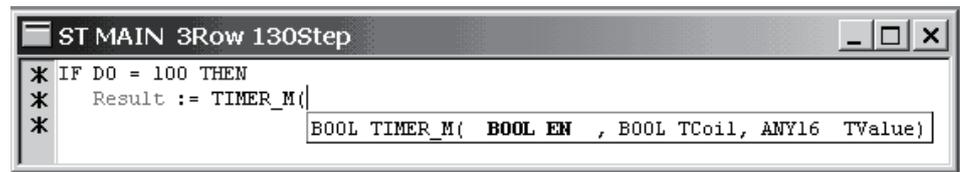
The function selected in the function list box is inserted into the ST edit screen.

[Setting procedure]

i) Select the function to be used from the function list box.

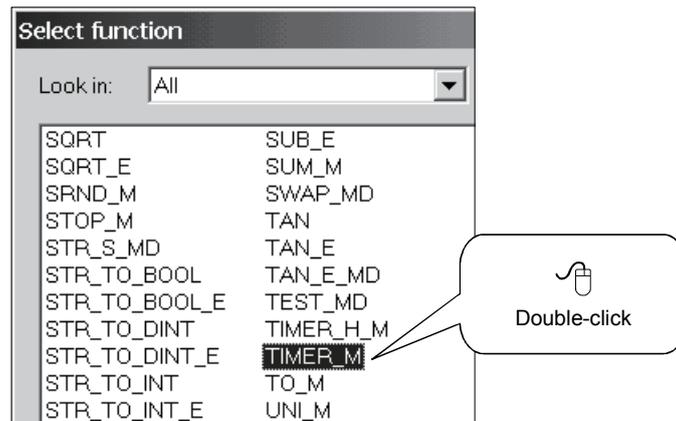
ii) Press the **OK** button or **Enter** key to insert the function into the cursor position on the ST edit screen.

After it has been inserted, enter its parameter to complete the function.



- A function can also be inserted by a double-click.

(Example: TIMER\_M)



- A function is also inserted when the ST edit screen is in the overwrite mode.
- When the initial of a function name is entered from the keyboard with the Select function screen open, the cursor moves to the first one of the function names that include that initial.

**REMARK**

The parameter can be displayed in the tool tip format.  
 For details, refer to "3.3.2 Displaying a function parameter".

### 3.2.2 Entering a label

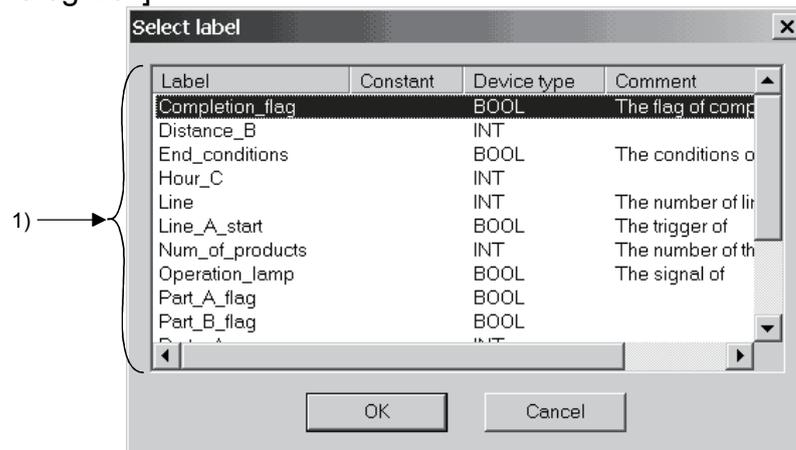
**[Purpose]**

If a label name is unknown during creation of the ST program, the label selection function can be used for input.

**[Operating Procedure]**

Choose [Edit] → [Select label] or press **F11** .

**[Dialog Box]**



**REMARK**

Set labels on the global variable (label) setting screen and local variable (local label) setting screen.

For details, refer to the "GX Developer Operating Manual" given in Relevant Manuals.

**[Description]**

1) Label list

The labels, constants, device types and comments set to the corresponding global variables and local variables are displayed on the ST edit screen.

The displayed labels are displayed in order of names.

## [Setting procedure]

- i) Select the label to be entered.
- ii) Press the **OK** button to insert the character string of the label name into the cursor position on the ST edit screen.

```

ST MAIN 10Row *****Step *
Start_conditions := TRUE;
End_conditions := FALSE;

(* Operation of Line A is started *)
Part_A_flag := TRUE;

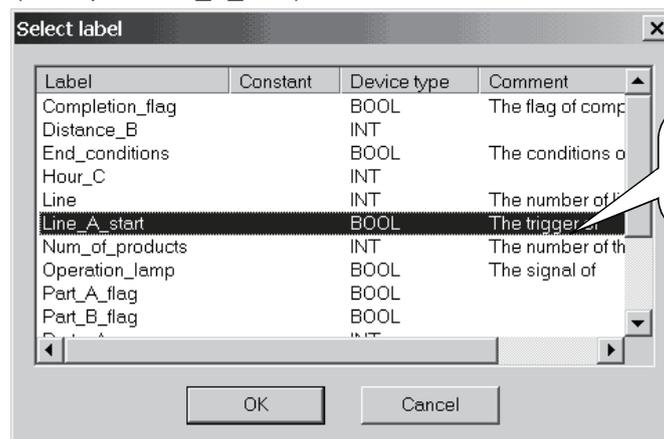
(* Movement OF Parts A *)
IF Part_A_flag AND Start_conditions THEN
  Line_A_start

```

**Point**

- A label name can also be inserted by a double-click.

(Example : Line\_A\_start)



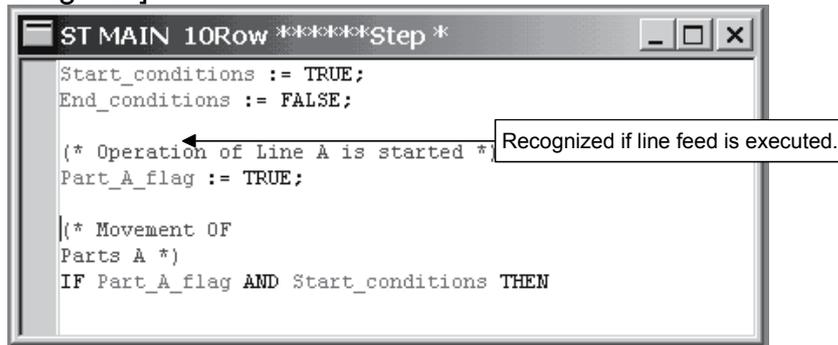
- A label name is also inserted when the ST edit screen is in the overwrite mode.
- When the initial of a label name is entered from the keyboard with the Select label screen open, the cursor moves to the first one of the label names that include that initial.
- The label display color can be changed.  
For the changing of the display colors, refer to "3.5.2 Changing the display colors".

### 3.2.3 Creating a comment

**[Purpose]**

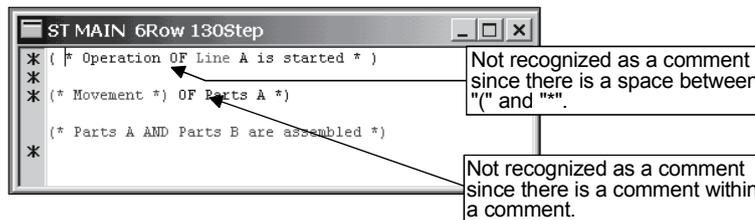
Program readability is improved by entering comments.  
 Enter a comment by enclosing it in "(" and ")".

**[Dialog Box]**



**Point**

- If characters are entered, they are not recognized as comments in the following cases.



- Comments differ from the statements, notes and device comments used in ladder programs.
- The comment display color can be changed.  
 For the changing of the display colors, refer to "3.5.2 Changing the display colors".

**REMARK**

For details, refer to the "QCPU (Q mode) Programming Manual (structured Text)" given in Relevant.

## 3.2.4 Control syntax upper case conversion function

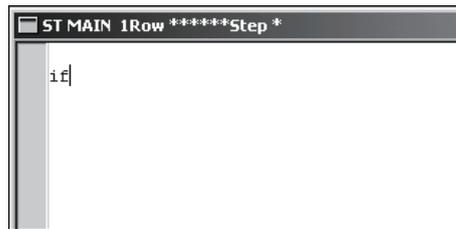
**[Purpose]**

If a control syntax is entered in lower case on the ST edit screen, it is converted into upper case automatically.

This function converts the target characters of the control syntax automatically to prevent input mistakes.

**[Dialog Box]**

When entered



After automatic conversion



- Target characters

The control syntaxes that will be converted into upper case are as shown below.

|   |
|---|
| IF, THEN, ELES, ELSIF, END_IF,<br>CASE, END_CASE,<br>FOR, TO, BY, DO, END_FOR,<br>WHILE, END_WHILE,<br>REPEAT, UNTILL, END_REPEAT,<br>EXIT, RETURN,<br>TRUE, FALSE, MOD, AND, XOR, OR |
|---|

- Non-conversion condition

When characters are entered within a comment sentence "( \* \* )", they are not converted.

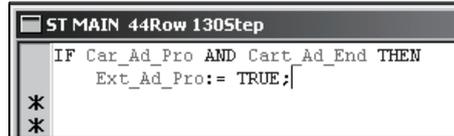


- Conversion is performed after the target characters have been entered or when any of the keys that separate characters (space, Enter, Tab) is pressed.
- The control syntax display color can be changed.  
For the changing of the display colors, refer to "3.5.2 Changing the display colors".

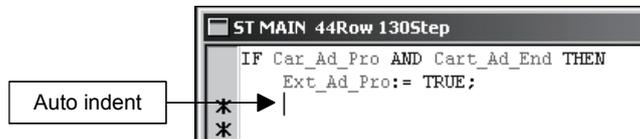
## 3.2.5 Auto indent function

## [Purpose]

Used to make setting to place the beginning of characters in the same position at the time of line feed on the ST edit screen.



```
ST MAIN 44Row 130Step
IF Car_Ad_Pro AND Cart_Ad_End THEN
  Ext_Ad_Pro:= TRUE;
*
*
```



```
ST MAIN 44Row 130Step
IF Car_Ad_Pro AND Cart_Ad_End THEN
  Ext_Ad_Pro:= TRUE;
*
*
```

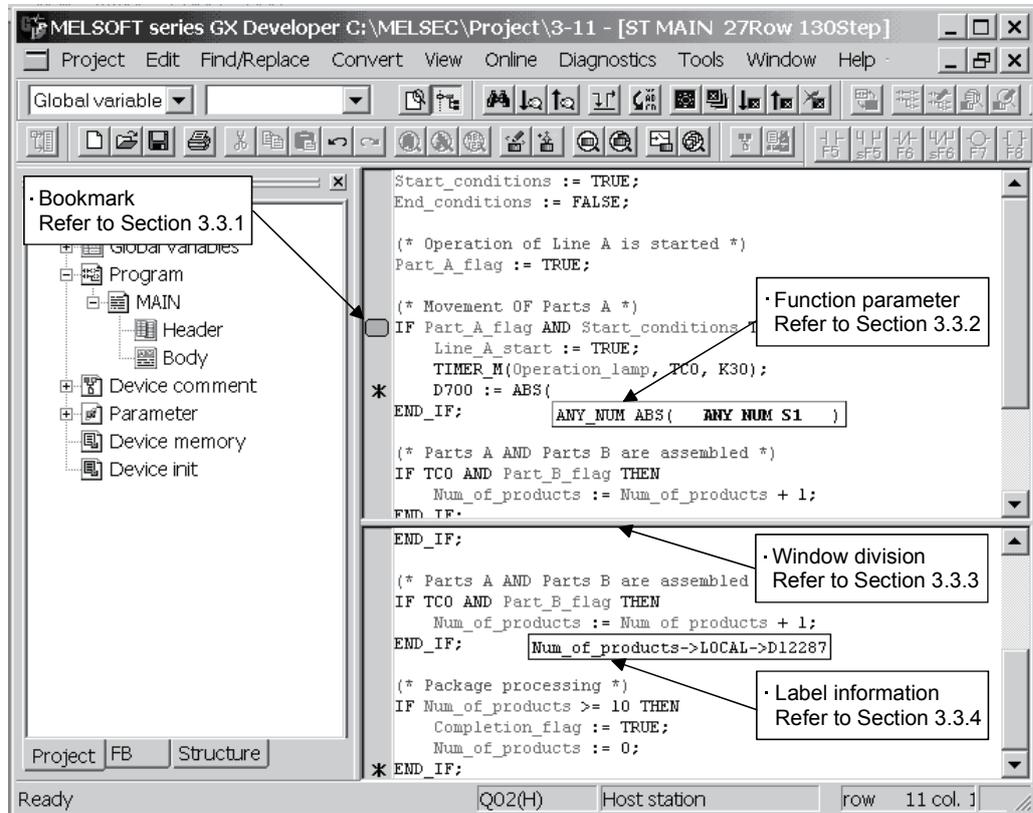
Auto indent

 **Point**

- The target keys of auto indent are the Tab and space keys.
- For the setting of auto indent, refer to "3.5.1 Changing the auto indent/tab width".

## 3.3 Useful Edit Functions

This section explains the useful functions related to the display of the ST edit screen.



## Other edit functions

- Find/Replace  
Used to find/replace the specified character string on the ST edit screen.  
For details, refer to "3.3.5 Find/Replace".
- Line jump  
Used to move to any line on the ST edit screen.  
For details, refer to "3.3.6 Line jump".
- Open Function Block  
Used to display the FB definitions used on the ST edit screen as a reference screen.  
For details, refer to "3.3.7 Open Function Block".
- Copy/Cut/Paste  
For details, refer to "3.3.8 Copy/Cut/Paste".
- Undo/Redo  
For details, refer to "3.3.9 Undo/Redo".

### 3.3.1 Using the bookmark

A bookmark is used to jump to a specific line.

This function is convenient when it is set as a mark to make a search in editing.

#### (1) Setting/deletion of bookmark

##### [Purpose]

Used to mark the line of the ST program or to delete the provided mark.

##### [Operating Procedure]

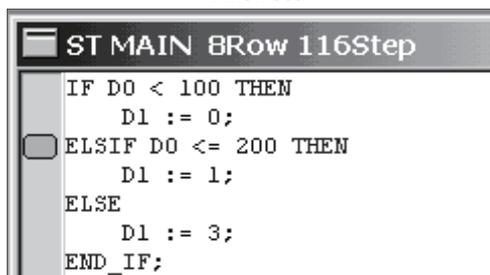
Move the cursor to the line where the bookmark is to be set/deleted.

Choose [Find/Replace] → [Bookmark setting/release], click (  ), or press

**Ctrl** + **F7**.

##### [Display screen]

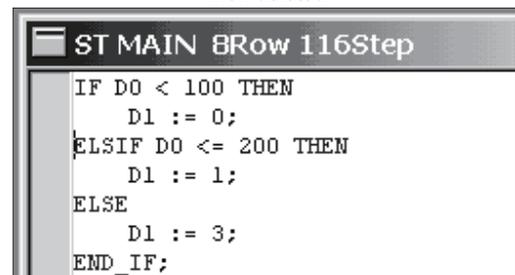
When set



```

ST MAIN 8Row 116Step
IF DO < 100 THEN
  D1 := 0;
 ELSIF DO <= 200 THEN
  D1 := 1;
ELSE
  D1 := 3;
END_IF;
  
```

When deleted



```

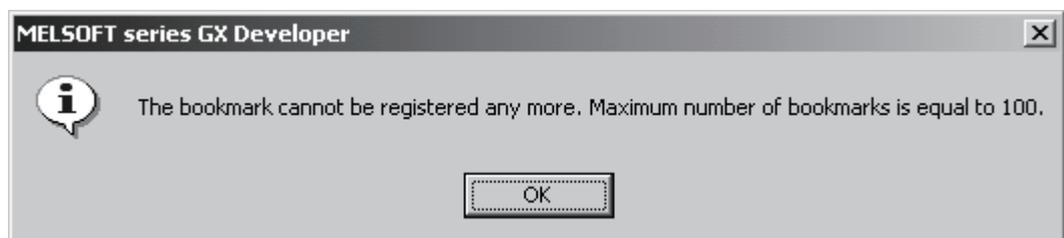
ST MAIN 8Row 116Step
IF DO < 100 THEN
  D1 := 0;
ELSIF DO <= 200 THEN
  D1 := 1;
ELSE
  D1 := 3;
END_IF;
  
```



By choosing [Find/Replace] → [Find] - "Set bookmark", bookmarks can be set at once on all the lines that have the found character string.  
For details, refer to "3.3.5 Find/Replace".

Up to 100 bookmarks can be set.

If more than 100 bookmarks are set, the following error message is displayed.



(2) Deletion of all bookmarks

[Purpose]

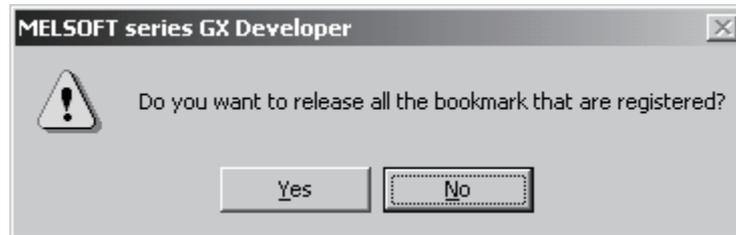
Used to delete all bookmarks set in the ST program at once.

[Operating Procedure]

Choose [Find/Replace] → [Release all bookmarks] or click (  ).

When "Release all bookmarks" is selected, the following confirmation message is displayed.

Execute after confirmation.



(3) Finding the bookmark line

[Purpose]

Used to find the specified bookmark line in the ST program.

[Operating Procedure]

The operation methods are as described below.

| Search Direction              | Operating Procedure   |
|-------------------------------|---|
| Downward from cursor position | Choose [Find/Replace] → [Find bookmark downward], click (  ), or press <b>F7</b> .       |
| Upward from cursor position   | Choose [Find/Replace] → [Find bookmark upward], click (  ), or press <b>Shift + F7</b> . |

A jump is made to the nearest bookmark line from the cursor position in the search direction.

(4) Bookmark list

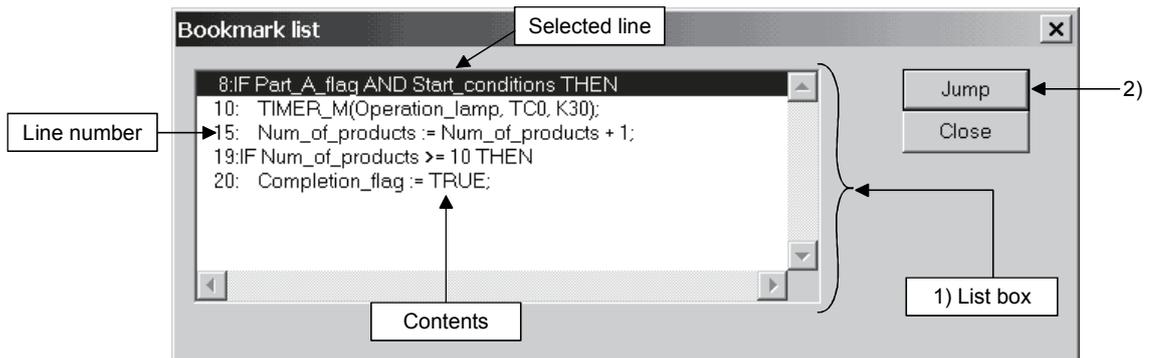
[Purpose]

Used to select the jump target line from among all the registered bookmarks.

[Operating Procedure]

Choose [Find/Replace] → [Bookmark list] or click (  ).

[Display screen]



[Description]

1) List box

Bookmark information is displayed in the form of "\*\*\*\*\* (line number).\*\*\*\*\* (registered contents)".

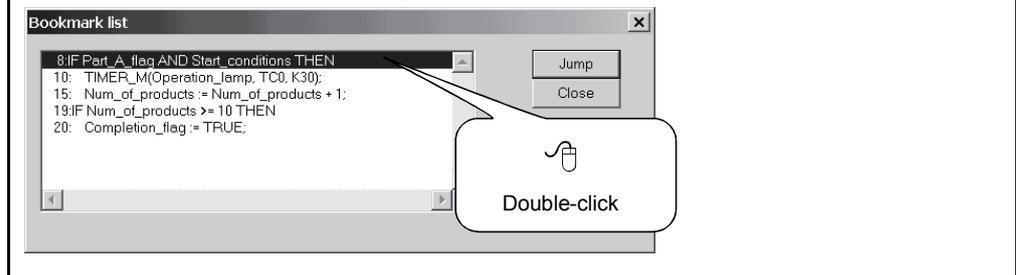
When the bookmark list screen is displayed, the first line is being selected.

2) **Jump** button

Select the jump target line in the list box and click the **Jump** button to move the cursor to the line that has the preset bookmark.



The cursor can also be moved by a double-click.



### 3.3.2 Displaying a function parameter

#### [Purpose]

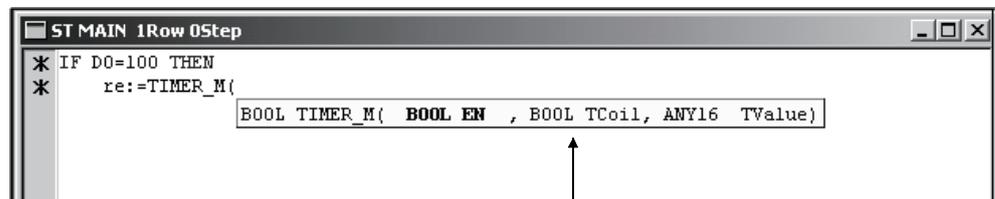
When a parameter is unknown during input of a function, the function parameter can be displayed for reference.

#### [Operating Procedure]

Choose [View] → [Function parameter] and check Function parameter.

Make selection from the Select function screen to input the function, or enter the function from the keyboard up to "(" . This displays the function parameter in the tool tip format.

#### [Dialog Box]



Function type, function name and parameter type are displayed.



For details of the function types, refer to the "QCPU (Q mode) Programming Manual (Structured Text)" given in Relevant Manuals.

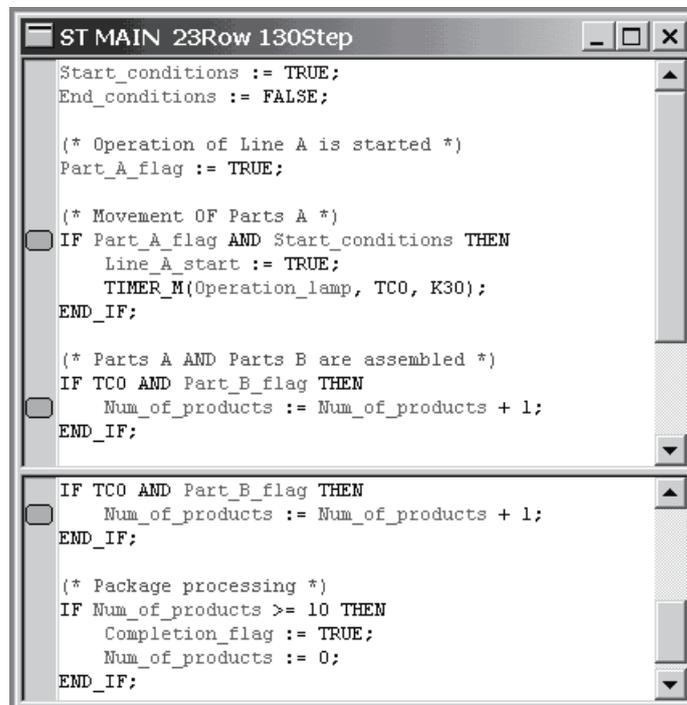
## 3.3.3 Window division

**[Purpose]**

When it is desired to confirm the program area that cannot be displayed on the screen during editing, divide the window into top and bottom areas.

**[Operating Procedure]**

- When dividing the window  
Choose [Window] → [Divide into two].
- When returning to one window  
Choose [Window] → [Divide into two].

**[Display screen]**


```

ST MAIN 23Row 130Step
Start_conditions := TRUE;
End_conditions := FALSE;

(* Operation of Line A is started *)
Part_A_flag := TRUE;

(* Movement OF Parts A *)
IF Part_A_flag AND Start_conditions THEN
  Line_A_start := TRUE;
  TIMER_M(Operation_lamp, TCO, K30);
END_IF;

(* Parts A AND Parts B are assembled *)
IF TCO AND Part_B_flag THEN
  Num_of_products := Num_of_products + 1;
END_IF;

IF TCO AND Part_B_flag THEN
  Num_of_products := Num_of_products + 1;
END_IF;

(* Package processing *)
IF Num_of_products >= 10 THEN
  Completion_flag := TRUE;
  Num_of_products := 0;
END_IF;

```



The program can be edited in either window.

## 3.3.4 Displaying the label information

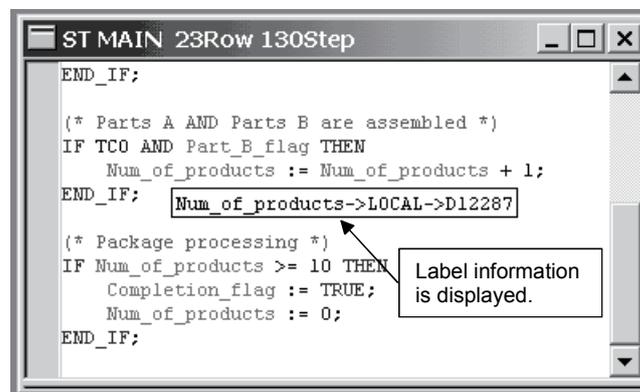
**[Purpose]**

Used to confirm the device assigned to the label after the ST program has been converted (compiled).

**[Operating Procedure]**

Choose [View] → [Label information] and check Label information.

When the mouse pointer is moved over the label on the ST edit screen, the label information is displayed in the tool tip format.

**[Display screen]**

The display format of the label information will be explained.

- When convert (compile) has not been performed  
Label name -> Label type -> Label comment
- When convert (compile) has been performed  
Label name -> Label type -> Label comment -> Device

The label type is displayed "GLOBAL" for a global variable, or "LOCAL" for a local label.



- A label comment that does not exist is not displayed.
- Before convert (compile), the device information is not displayed since the device has not yet been assigned.
- When the FB or structure is used, the FB definition name or structure definition name is displayed.

### 3.3.5 Find/Replace

#### (1) Find

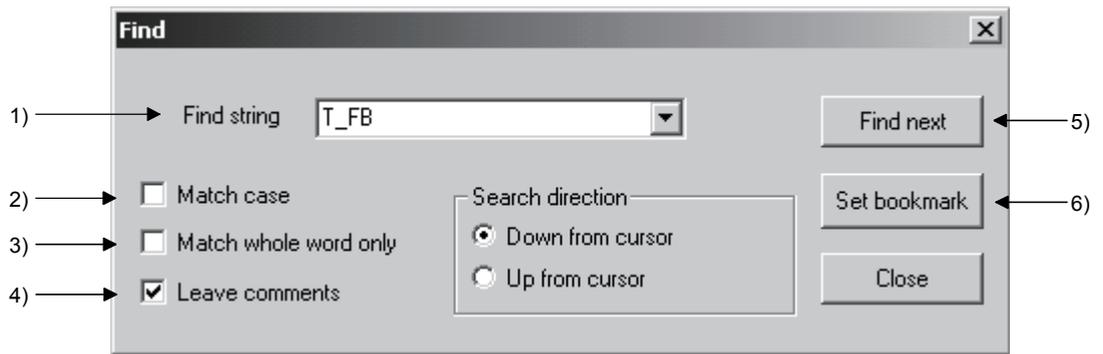
##### [Purpose]

Find the specified character string on the ST edit screen.

##### [Operating Procedure]

Choose [Find/Replace] → [Find] or press **Ctrl** + **F**.

##### [Dialog Box]



##### [Description]

#### 1) Find string

Input the character string to be found.

Alternatively, the character string can be displayed and selected from the list box.



- Specify the character string to be found within 256 characters.
- In the Find string list box, up to 10 character strings found in the past are displayed in the order from most to least recent.

#### 2) Match case

Select whether a distinction between upper case and lower case will be made or not.

#### 3) Match whole word only

Select whether a search will be made in a word unit or not.

The character string to be found is a collection of only alphabets or a collection of only numerals.

A tab, space, \_ (under bar), etc. are recognized as separating characters.

Example: When a search is made for "abc"

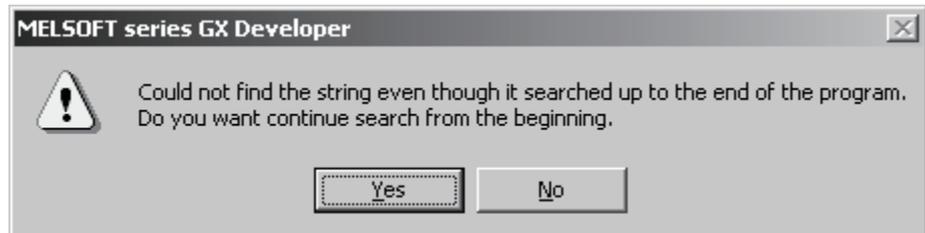
| Character string     | Search result | Character string | Search result |
|----------------------|---------------|------------------|---------------|
| abcdef               | ×             | abc!def          | ○             |
| abc <b>tab</b> def   | ○             | abc01234         | ○             |
| abc <b>space</b> def | ○             | 01234abc         | ○             |
| abc_                 | ○             |                  |               |

×...Not searched for, ○...Searched for

- 4) Leave comments  
Select whether a search will be made within comment sentences or not.
- 5) **Find next** button  
Starts a search.
- 6) **Set bookmark** button  
Searches the ST program on the ST edit screen for the character string to be found, and sets bookmarks on all the lines where the character string has matched.

#### [Setting procedure]

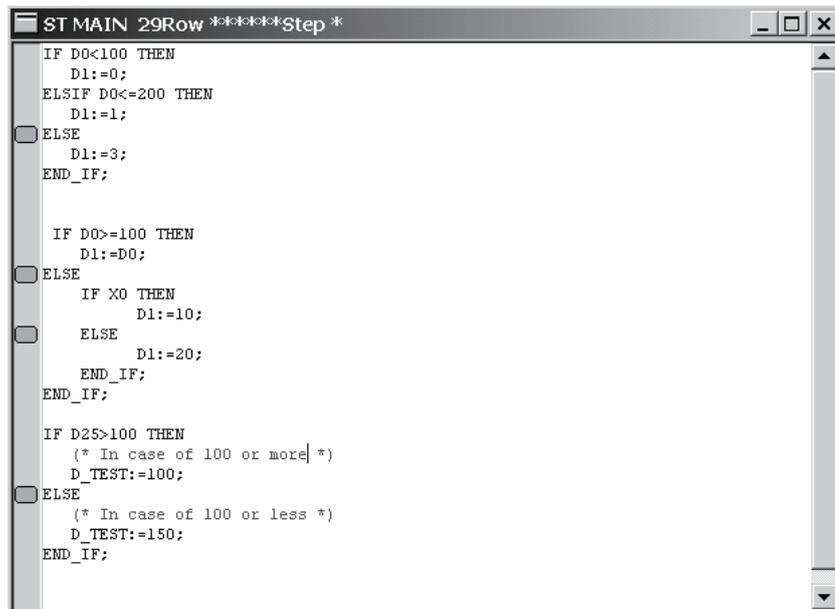
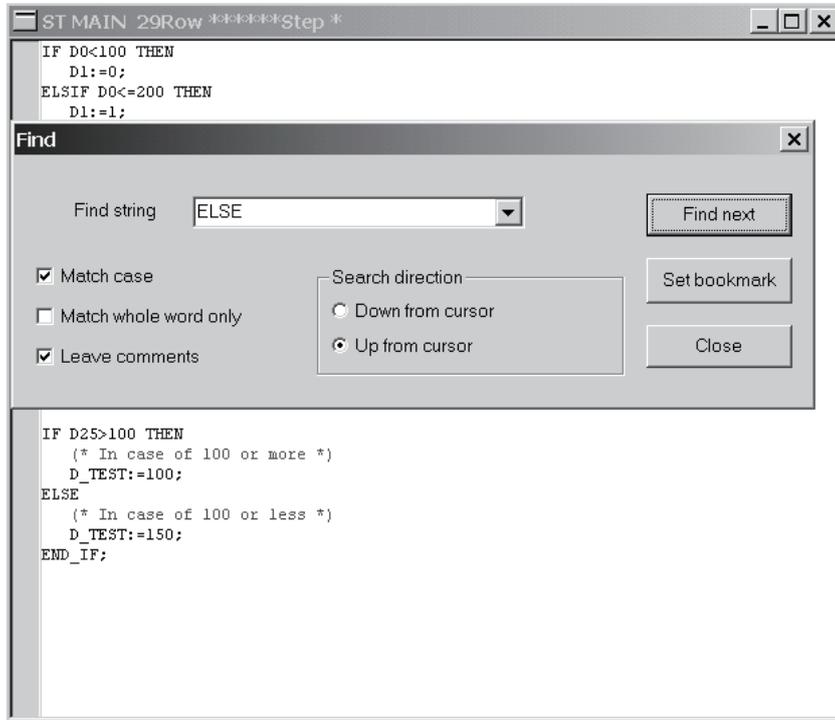
- i) When making a search for the next  
Input the character string to be found, and click the **Find next** button.
- When the character string to be found is found  
The found character string is displayed in a range-selected status. The cursor moves to the found position.  
To further continue a search, perform either of the following operations.
    - Find downward  
Choose [Find/Replace] → [Find downward] or press **F5**.  
A search is made downward, starting from the cursor position.
    - Find upward  
Choose [Find/Replace] → [Find upward] or press **Shift** + **F5**.  
A search is made upward, starting from the cursor position.
  - When the character string to be found is not found  
The following message is displayed.



ii) When making a search using bookmark setting

Input the character string to be set in Find string, and click the **Set bookmark** button.

Bookmarks are set on all the lines on the ST edit screen that have the character string to be found.



## (2) Replace

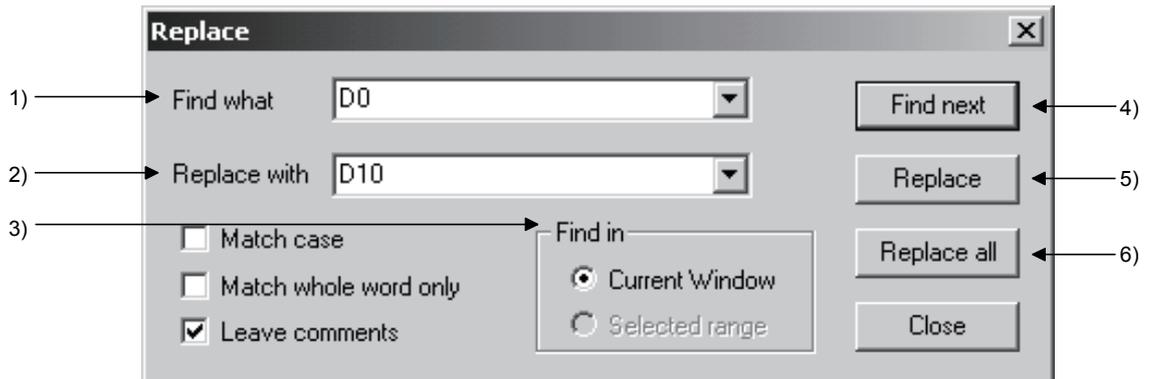
## [Purpose]

Search for a character string on the ST edit screen and replace it with the specified character string.

## [Operating Procedure]

Choose [Find/Replace] → [Replace] or press **Ctrl** + **H**.

## [Dialog Box]



## [Description]

## 1) Find what

Input the character string to be replaced.

Alternatively, the character string can be selected from the list box.

## 2) Replace with

Input the character string to replace the one to be replaced.

Alternatively, the character string can be selected from the list box.



- In each of Find what and Replace with, specify the character strings within 256 characters.
- In each of the Find what and Replace with list boxes, up to 10 character strings replaced in the past are displayed in the order from most to least recent.

## 3) Find in

When "Current window" is selected

Replacement is made in the program currently being edited.

When "Selected range" is selected

Replacement is made in the range selected by dragging the mouse.

When the replacement range is selected on the ST edit screen, "Selected range" on the Replace screen can be selected.

4) **Find next** button

Starts a search.

5) **Replace** button

Replaces only the character string found first.

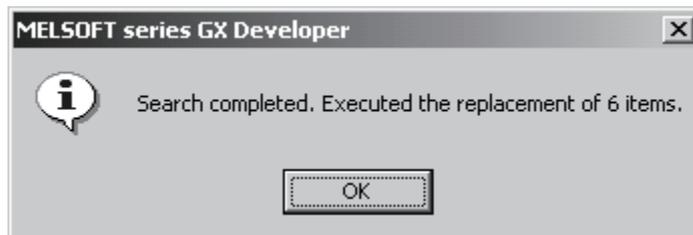
6) **Replace all** button

Replaces all the corresponding character strings on the target ST edit screen.

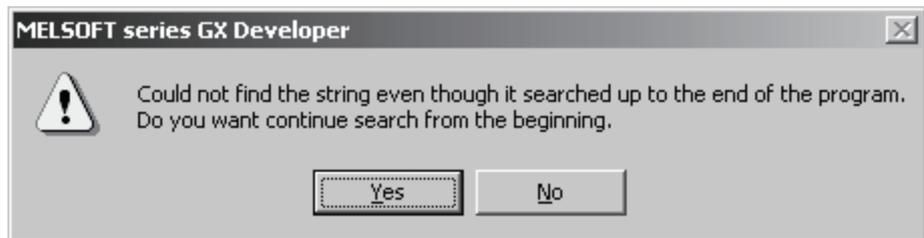
 **Point**

- As the replacement processing target range, select either "Current window" or "Selected range". If the range has not been selected, "Current window" is the replacement target.
- If the **Replace** button is pressed without a character string being set in Replace with, the character string in Find what on the ST edit screen is deleted.

- When the character string to be replaced is found for Replace all  
The following message is displayed.



- When the character string to be replaced is not found  
The following message is displayed. The cursor does not move.



## 3.3.6 Line jump

**[Purpose]**

Used to move the cursor to the specified line on the ST edit screen.

**[Operating Procedure]**

Choose [Find/Replace] → [Line jump], click (  ), or press **Ctrl** + **J**.

**[Dialog Box]****[Description]**

- 1) Line setting edit box  
Input the line to which the cursor will jump.
- 2) **J** button  
Causes the cursor to jump to the specified line.



If the specified line is beyond the program being edited, the cursor moves to the last line of the program.

If the following error message is displayed, a line jump cannot be made. Confirm the error definition and make setting again.

- When the **J** button is clicked with the specified line set to line 65536 or more or to other than an integer.



### 3.3.7 Open Function Block

**[Purpose]**

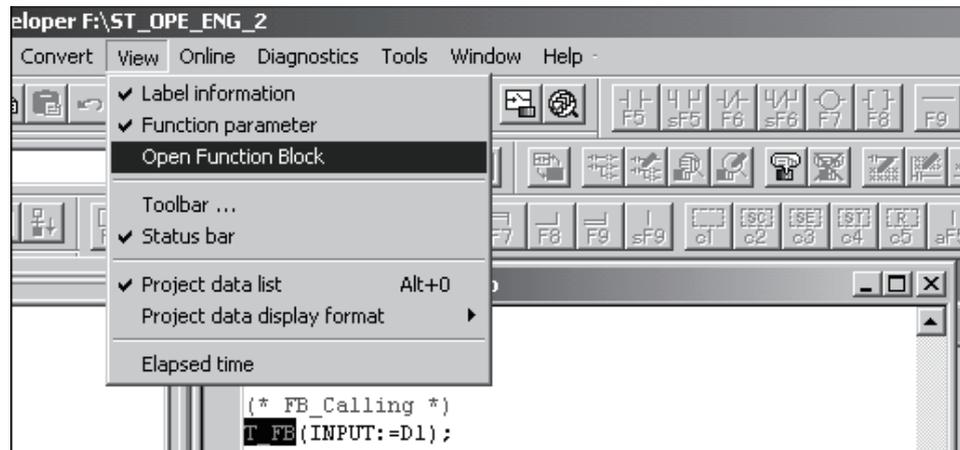
The FB window is used to confirm the contents of the FB definition program used in the ST program.

**[Operating Procedure]**

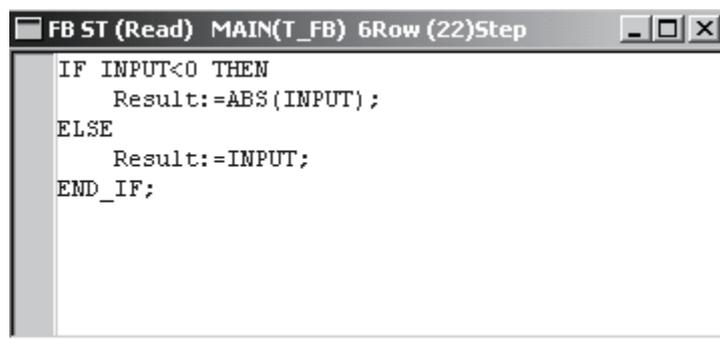
- Select the FB name.

Choose [View] → [Open Function Block] or right-click and choose [Open Function Block].

(FB name: T\_FB)



The FB window is displayed.  
(The contents of the T\_FB program are displayed.)



This program is read-only and cannot be edited.

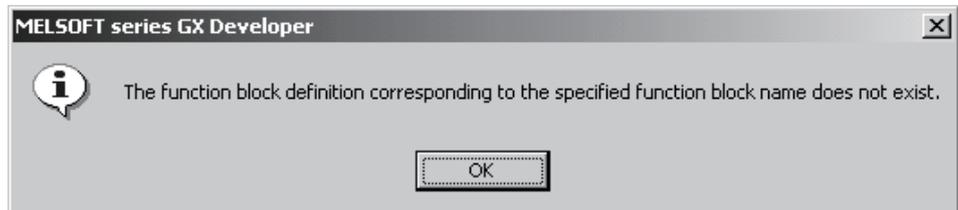
If either of the following error messages is displayed, the FB window cannot be opened.

Confirm the error definition and make setting again.

- When the FB definition is not yet converted (compiled)



- When the selected character string is not defined as the FB name



### 3.3.8 Copy/Cut/Paste

- In the case of copy

- [Operating Procedure]

- Choose [Edit] → [Copy], click (  ), or press **Ctrl** + **C**.

- i) Specify the range of the character strings to be copied.
    - ii) The character strings in the specified range are copied.

- In the case of cut

- [Operating Procedure]

- Choose [Edit] → [Cut], click (  ), or press **Ctrl** + **X**.

- i) Specify the range of the character strings to be cut.
    - ii) The character strings in the specified range are cut.

- In the case of paste

- [Operating Procedure]

- Choose [Edit] → [Paste], click (  ), or press **Ctrl** + **V**.

- i) Move the cursor to the position where the character strings will be pasted.
    - ii) The character strings copied or cut are pasted.

## 3.3.9 Undo/Redo

**[Purpose]**

The editing operation performed immediately before can be undone or the undoing operation can be redone.

**[Operating Procedure]**

## ● Undo

Choose [Edit] → [Undo], click (  ), or press **Ctrl** + **Z**.

## ● Redo

Choose [Edit] → [Redo], click (  ), or press **Ctrl** + **Y**.

 **Point**

- Number of operation times enabled for Undo/Redo ..... 40 times
- Operation disabled for Undo/Redo ..... Copy
  - Cursor movement
  - Bookmark setting/deletion
  - Convert (compile)
  - Project storage

### 3.4 Performing Convert (Compile)

#### [Purpose]

The created ST program is converted (compiled) into a program that can be executed by the programmable controller CPU.

#### [Operating Procedure]

- When the program currently being edited is converted (compiled)  
Choose [Conversion] → [Convert/Compile], press **F4**, or click (  ).
- When all the programs not yet converted (compiled) are batch-converted (compiled)  
Choose [Conversion] → [Convert/Compile (All programs being edited)], click (  ), or press **Alt** + **Ctrl** + **F4**.
- When all programs are batch-converted (compiled)  
Choose [Conversion] → [Convert/Compile (All programs)].



When [Convert/Compile (All programs)] is selected, the programs already converted (compiled) are also converted (compiled) again.

By performing convert (compile) again, the devices assigned to the programs whose devices have not been changed may be changed.

#### (1) When operation is completed normally

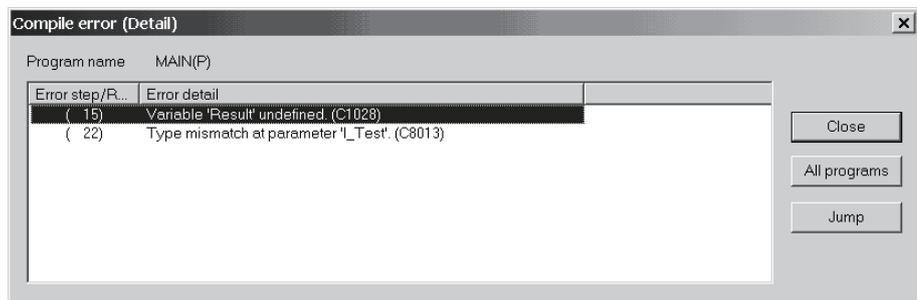
The "" on the title bar that indicates the programs are not yet converted (compiled) disappears and the number of steps is displayed.

#### (2) When error has occurred

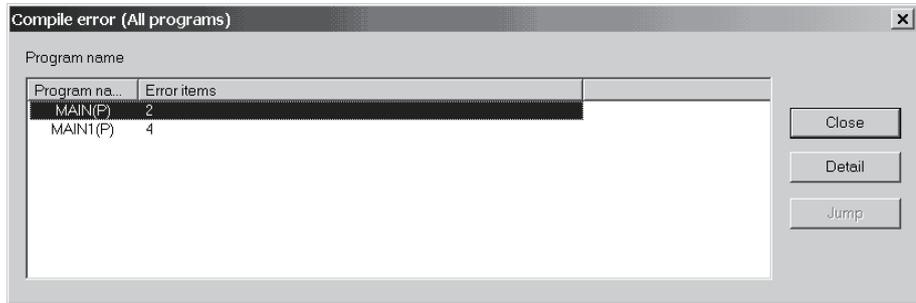
##### i) When an error has occurred in one program

When an error has occurred, ""\*\*\*\*\*Step\*" that indicates the program is not yet converted (compiled) is displayed in the window title.

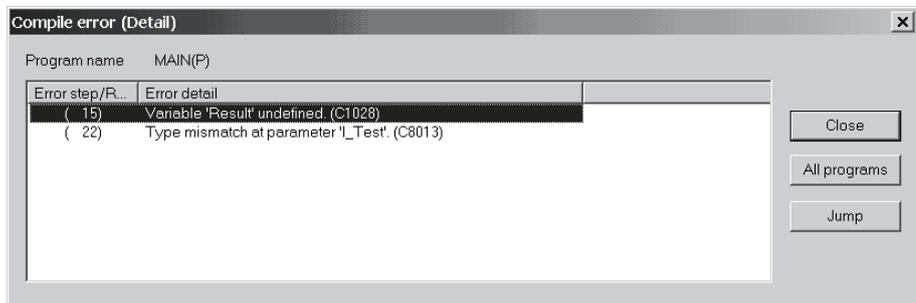
The following screen is displayed.



ii) When an error has occurred in more than one program  
The following screen is displayed.



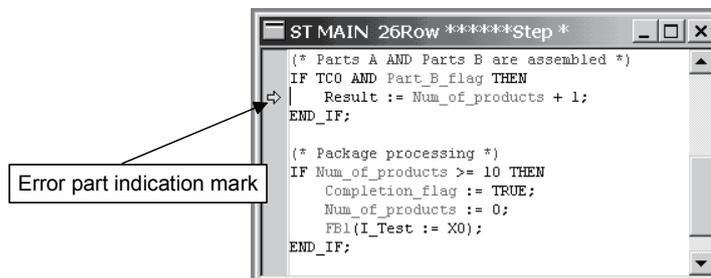
Select the program name and click the **Detail** button to display the error result of the corresponding program.



iii) How to jump to the error part

- Select the corresponding error in the error display list and click the **Jump** button.
- Select the corresponding error in the error display list and press the **Enter** key or double-click.

When the cursor jumps to the selected error line, the error part indication mark is displayed on the indicator bar as shown below for identification of the error part.





The position of the error part indication mark may differ from the actual error part. Locate the faulty part from the error definition displayed on the "Compile error (Detail)" screen and the program contents of the line where the error part indication mark is displayed.

Example of error part indication

Example of error part indication

```

ST MAIN 14Row *****Step *
IF S_LBL=TRUE THEN
  FB1(IO_TEST:=M0);
END_IF;
FB2(IO_TEST:=-M0);
IF S_LBL=TRUE THEN
  M0:=TRUE;
END_IF;
OUT_M(M0,M12);
BMINUS_M(M0,D123,LABEL);
    
```

Compile error (Detail)

Program name MAIN(P)

| Error step/Row | Error detail           |
|----------------|------------------------|
| ( 9)           | " , " missing. (C8006) |

**REMARK**

For details related to errors, refer to the "QCPU (Q mode) Programming Manual (Structured Text)" given in Relevant Manuals.

### 3.5 Customizing the ST Edit Screen

Operation-related data can be set on the ST edit screen.

#### 3.5.1 Changing the auto indent/tab width

##### [Purpose]

Set the auto indent/tab width.

##### (1) Auto indent

This function performs an auto indent when the **Enter** key is pressed during editing.

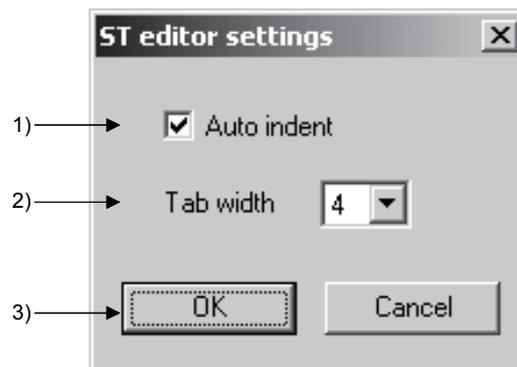
##### (2) Tab width

This function sets the tab width at the time when the **Tab** key is pressed.

##### [Operating Procedure]

Choose [Tools] → [ST editor settings].

##### [Dialog Box]



##### [Description]

##### 1) Auto indent check box

Checked: Auto indent valid

Not checked: Auto indent invalid

##### 2) Tab width combo box

Any of 4, 8 and 12 can be selected.

##### 3) **OK** button

Makes the setting valid.

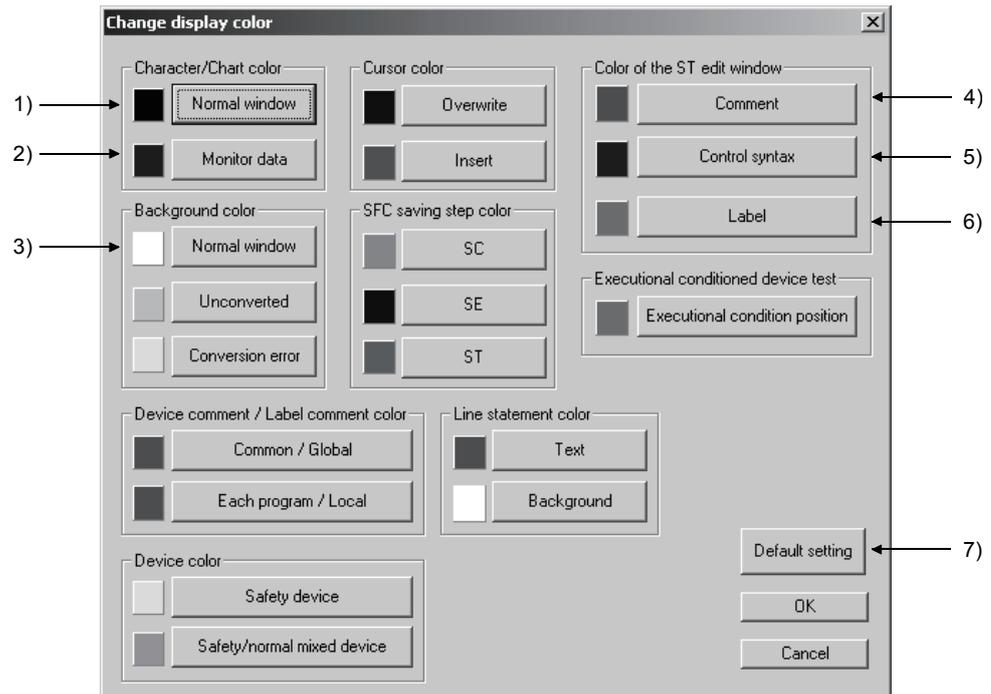
## 3.5.2 Changing the display colors

**[Purpose]**

The background, comments, control syntaxes, label character strings, etc. on the ST edit screen are displayed in the specified display colors.

**[Operating Procedure]**

Choose [Tools] → [Change display color].

**[Dialog Box]****REMARK**

Here, the parts related to the ST edit screen will be explained. For the other parts, refer to the "GX Developer Operating Manual" given in Relevant Manuals.

**[Description]**

- 1) Normal window (Character/Chart color)  
Specify the color of the display characters such as the device names and operators.
- 2) Monitor data (Character/Chart color)  
Specify the color of the display characters on the monitor screen.
- 3) Normal window (Background color)  
Specify the background color of the ST edit screen.
- 4) Comment  
Specify the color of the display characters in the comment parts.
- 5) Control syntax  
Specify the color of the display characters in the control syntax parts.
- 6) Label  
Specify the color of the display characters in the label parts.

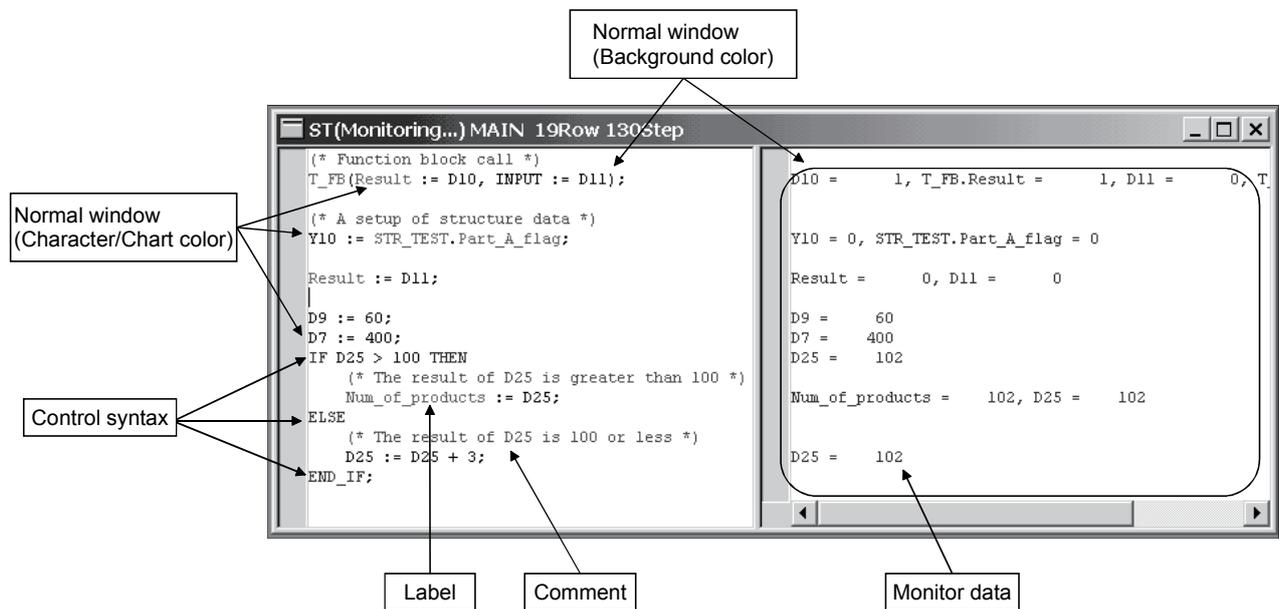
7) Default setting

Returns to the standard.

The standard colors are as follows.

|                                       |         |
|---------------------------------------|---------|
| Normal window (Character/Chart color) | : Black |
| Monitor data (Character/Chart color)  | : Blue  |
| Normal window (Background color)      | : White |
| Comment                               | : Green |
| Control syntax                        | : Blue  |
| Label                                 | : Pink  |

A display example on the ST edit screen is shown below.



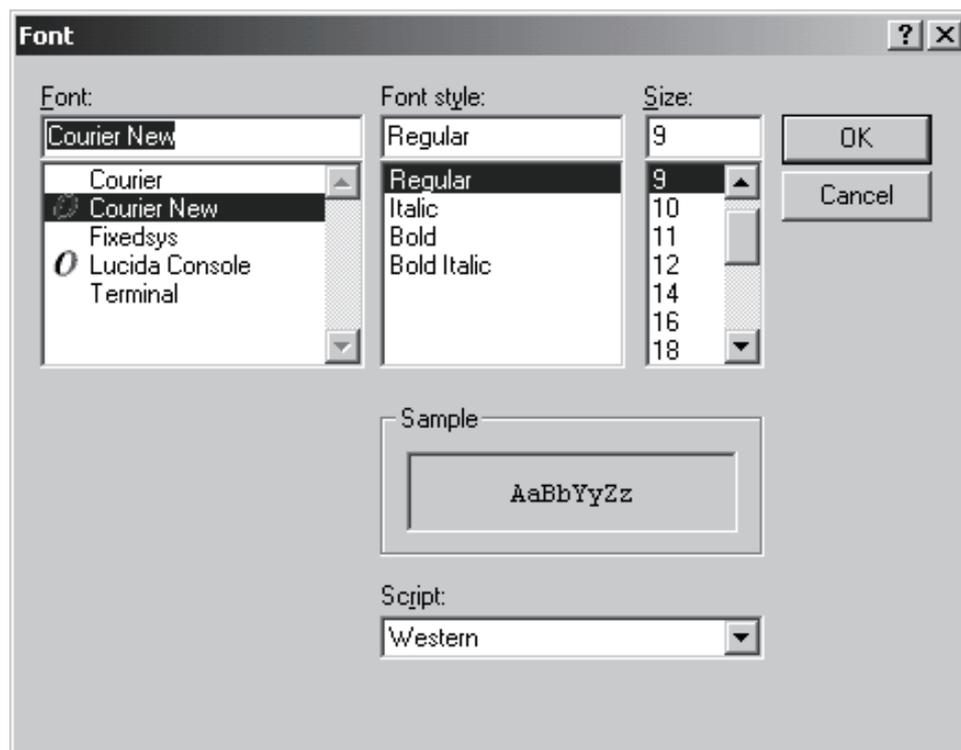
## 3.5.3 Changing the display font

**[Purpose]**

The font used on the ST edit screen or for monitoring can be changed.

**[Operating Procedure]**

Choose [Tools] → [Font].

**[Dialog Box]****[Description]**

- 1) Font  
Set the font name of the display characters.
- 2) Font style  
Set the style of the display characters.
- 3) Size  
Set the size of the display characters.
- 4) **OK** button  
Makes the setting valid.

**REMARK**

The default settings are as follows.

Font : Courier New  
 Font style : Regular  
 Size : 9

## 4 ONLINE

This chapter explains the read of the ST program from the programmable controller CPU, the write of the ST program to the programmable controller CPU, and the confirmation of monitor and program behavior.

## 4.1 Read from PLC

Read the ST program from the programmable controller CPU.

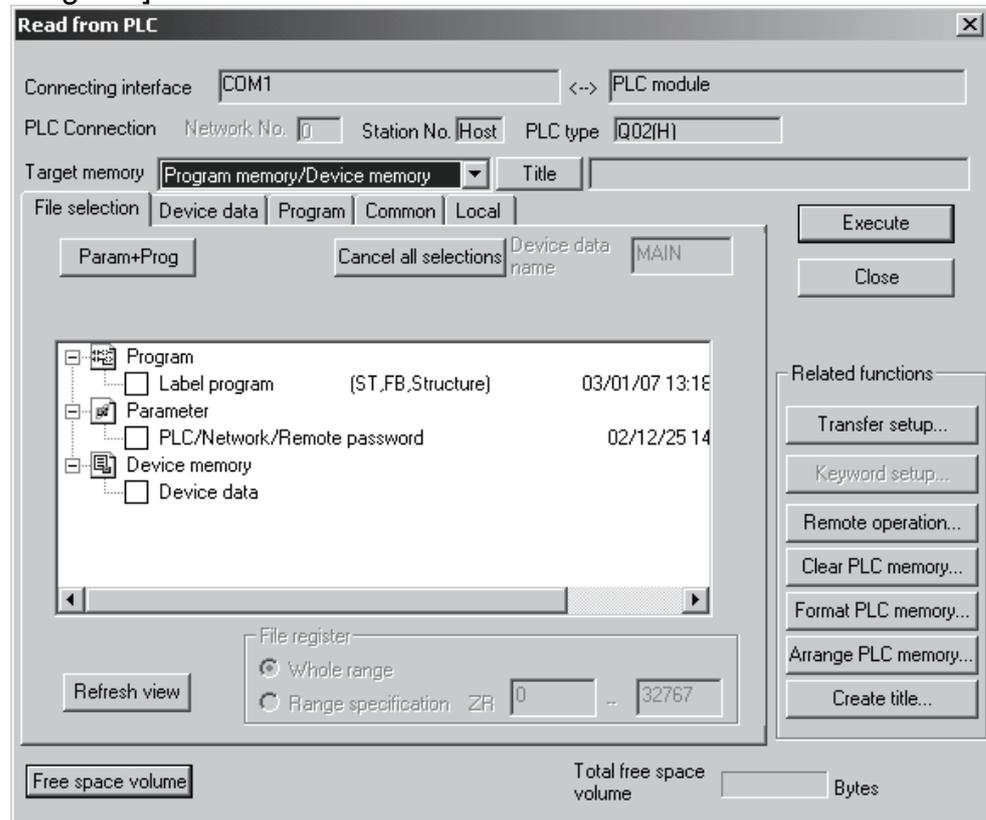
## [Purpose]

Used to read the ST program from the programmable controller CPU.

## [Operating Procedure]

Choose [Online] → [Read from PLC], or click (  ).

## [Dialog Box]



## [Setting procedure]

- i) Choose [Online] → [Transfer setup] and set the connection target.
- ii) Choose [Online] → [Read from PLC] to display the Read from PLC screen.
- iii) Select the corresponding item in the <<File selection>> tab.
  - When "Parameter + Prog" is selected  
The parameters and program are selected.
  - When the "Cancel all selections" button is selected  
All are deselected.
- iv) Click Execute.

 **Point**

- When the ST program is to be read from the programmable controller, the read range cannot be specified.
- When the label program is to be read from the programmable controller, read from PLC is started after the project where "Use label" was selected in the label setting is created or read.
- When there is no label program in the CPU, "Label program" is not displayed in the data list of the Read from PLC dialog.

**REMARK**

For details, refer to the "GX Developer Operating Manual" given in Relevant Manuals.

## 4.2 Write to PLC

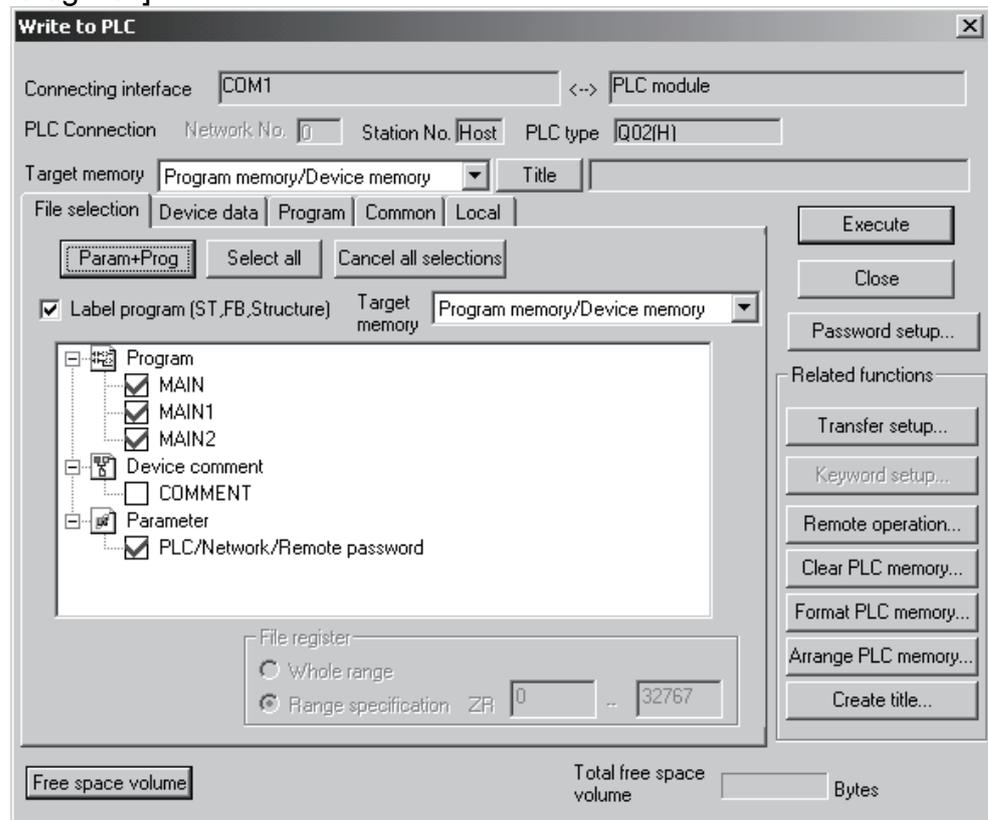
Write the already converted (compiled) ST program to the programmable controller CPU.

**[Purpose]**

Used to write the converted (compiled) program to the programmable controller CPU.

**[Operating Procedure]**

Choose [Online] → [Write to PLC], or click (  ).

**[Dialog Box]****[Setting procedure]**

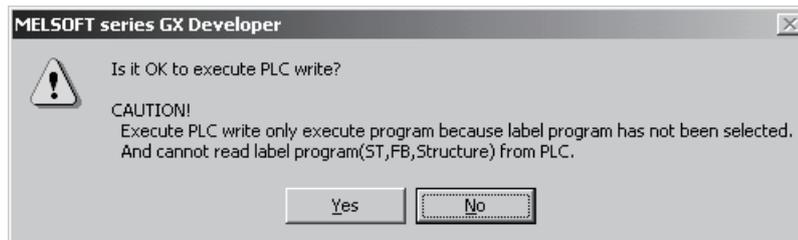
- i) Choose [Online] → [Transfer setup] and set the connection target.
- ii) Choose [Online] → [Write to PLC] to display the Write to PLC screen.
- iii) Select the corresponding item in the <<File selection>> tab.
  - When the "Label program (ST, FB, structure)" button is selected  
The label program can be written.
  - When **Param + Prog** is selected  
The parameters and program are selected.
  - When the **Select all** button is selected  
All are selected.
  - When the **Cancel all selections** button is selected  
All are deselected.
- iv) Click **Execute**.

**Point**

- When the ST program is to be written to the programmable controller, the write range cannot be specified.
- If the program specified for write to PLC is not yet converted (compiled), the following error message is displayed and writing that program to the programmable controller is suspended.



- If the "Label program (ST, FB, structure)" check button was not checked for write to PLC, the following error message is displayed. When Yes is selected, only the execution program is written to the programmable controller and the label program cannot be read.

**REMARK**

For details, refer to the "GX Developer Operating Manual" given in Relevant Manuals.

### 4.3 Monitoring the ST Program

Monitor the ST program written to the programmable controller CPU to confirm the operation status of the programmable controller CPU.

#### 4.3.1 Monitoring the ST program

##### [Purpose]

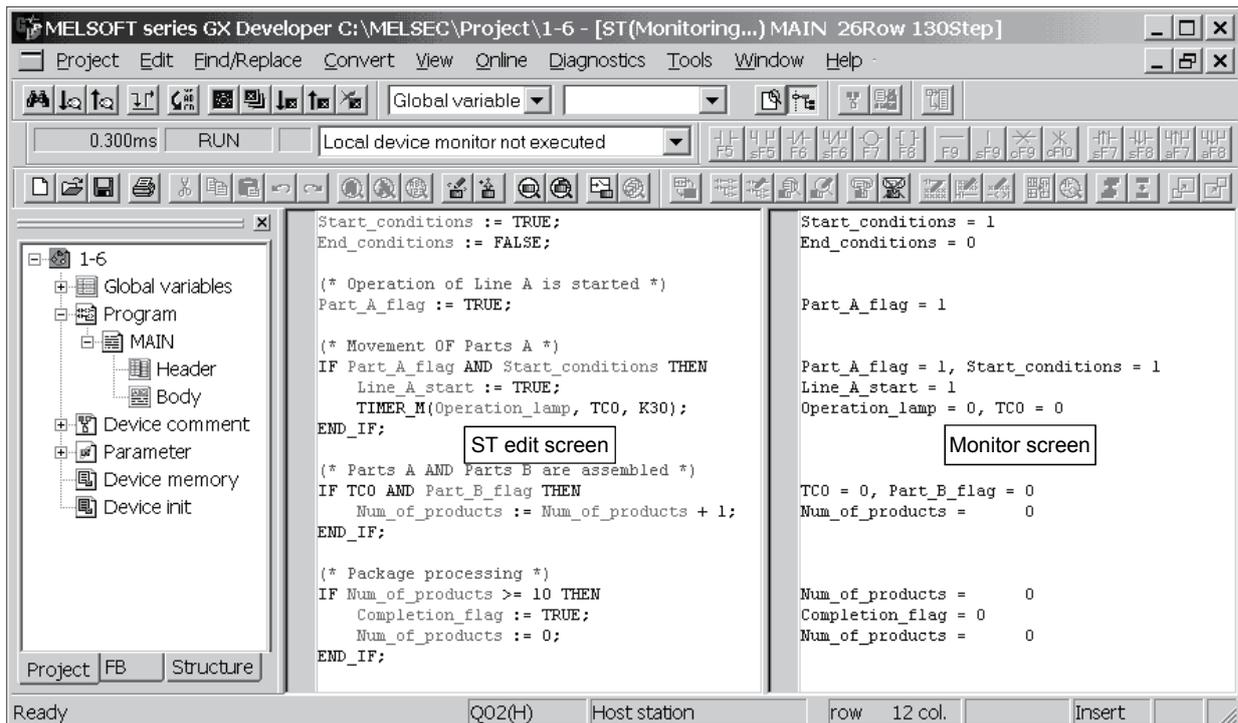
Used to monitor the ST program to confirm the operation status of the programmable controller CPU.

##### [Operating Procedure]

- When starting monitor  
Choose [Online] → [Monitor] → [Monitor], click (  ), or press [F3].
- When stopping monitor  
Choose [Online] → [Monitor] → [Stop monitor], click (  ), or press [Alt] + [F3].

##### [Display screen]

The following monitor screen is displayed.



The monitor screen displays the variable (label, structure, device), which is used on each line of the ST edit screen, on the same line of the monitor screen in a "label = monitor value" format.

When there are more than one identical variable on one line, the first one is displayed and the second and later are not displayed.

| Variable Type    | ST edit screen   | Monitor screen                                     | Remarks   |
|------------------|--|--|---|
| Bit              | Input := TRUE;<br>Input := FALSE;                          | Input = 1<br>Input = 0                             | TRUE 1<br>FALSE 0   |
| Word             | Word1 := -32767;   | Word1 = -32767                                     | Decimal: 6 characters   |
|                  |  | Word1 = H8001                                      | Hexadecimal: 5 characters   |
| Real             | Result := 340282.338;                                      | Result = 3.403e + 005                              |   |
| Character string | Str1 := "ABCDEFGH" ;                                       | Str1 = 16961                                       | Decimal: The first one word of a character string is displayed in decimal.<br>6 characters                                  |
|                  |  | Str1 = H4241                                       | Hexadecimal: The first one word of a character string is displayed in hexadecimal.<br>5 characters                          |
| Array            | Label [0] := 80;<br>Label [1] := 100;<br>Label [3] := 160; | Label [0] = 80<br>Label [0] = 80<br>Label [0] = 80 | Only the value that begins with [0] is displayed.<br>The display format changes depending on the type of the selected data. |
| Structure        | STR_A. name := "ABCDEFGHJIJ" ;<br>STR_A. point := 40 ;     | STR_A. name = 16961<br>STR_A. point = 40           | The display format changes depending on the type of the selected data.  |

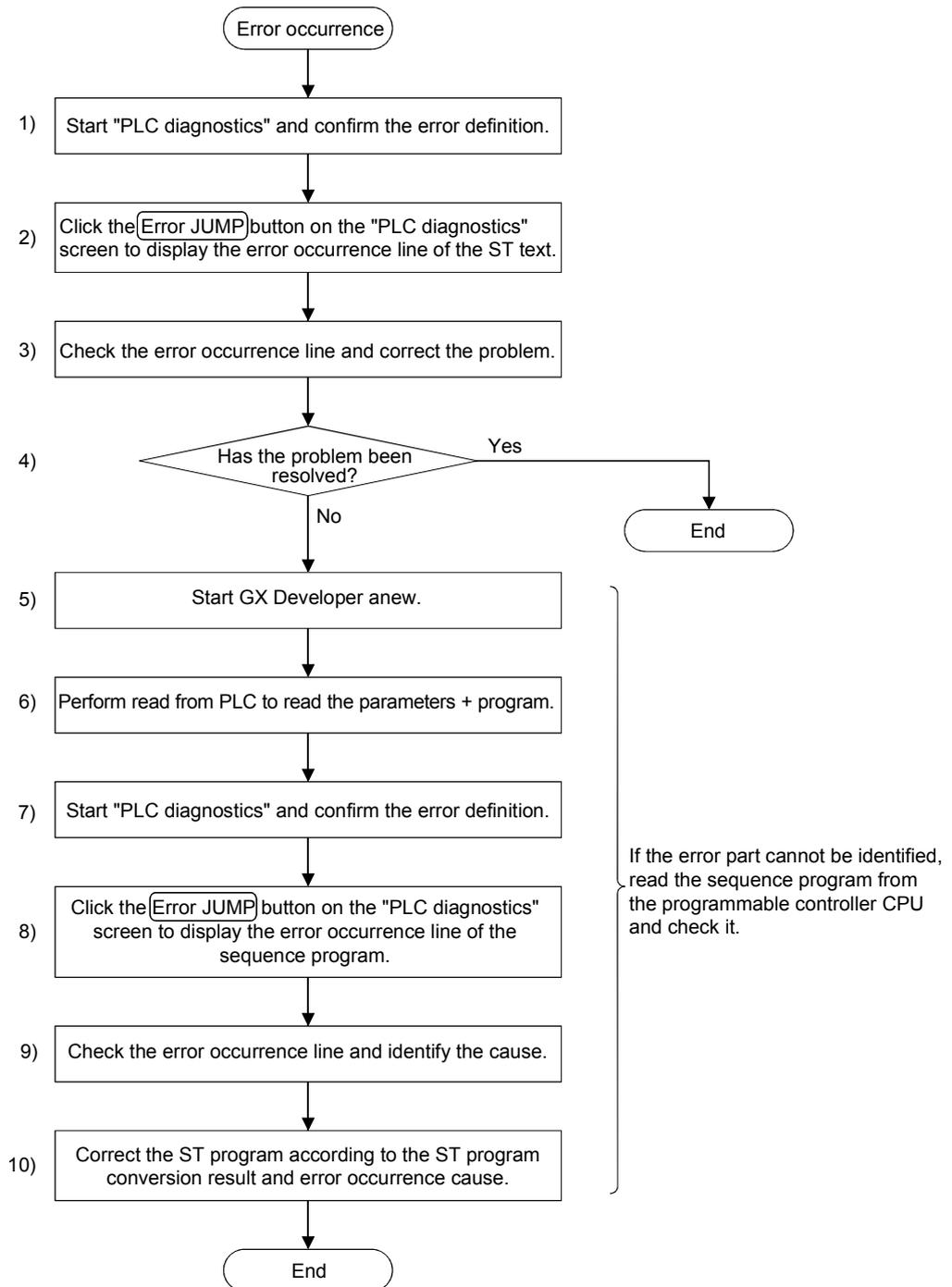


- Switching between decimal and hexadecimal can be performed by choosing [Online] → [Monitor] → [Change current value monitor (Decimal)] or [Change current value monitor (Hexadecimal)].
- The background color is the same as that of the ST edit screen.  
The display color selected by choosing [Tools] → [Change display color] → "Monitor data" is used.

### 4.3.2 Troubleshooting at error occurrence in ST program

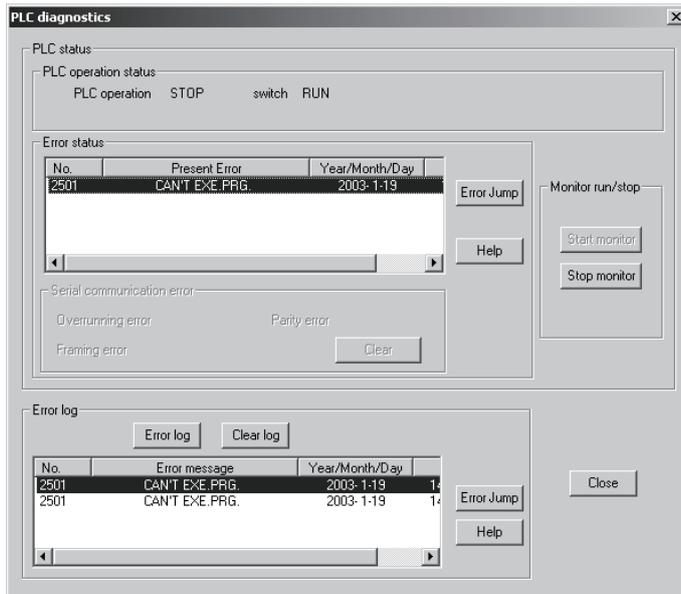
This section explains troubleshooting to be performed when the ST program written to the programmable controller CPU has resulted in an error.

Troubleshooting to be performed when an error has occurred in the programmable controller CPU for the ST program

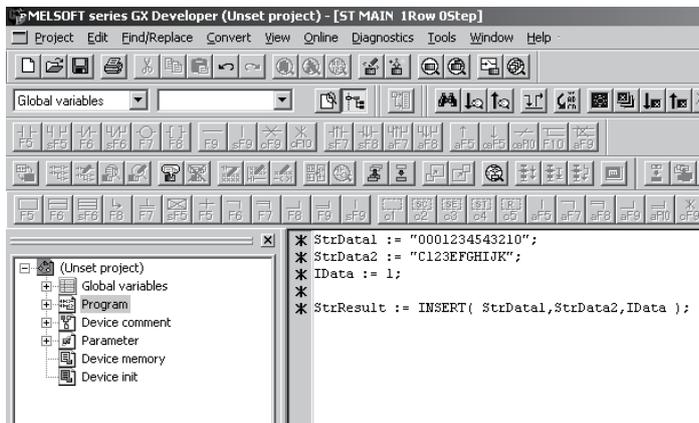


The operating procedure to be performed when an error occurred in the programmable controller CPU for the ST program will be explained using the actual screen as an example.

- (1) Display the ST program where the error has occurred. (Operations 1), 2) in the flowchart)  
 Perform "PLC diagnostics" to display the error occurrence line.



- 1) Confirm the error definition.
- 2) Click the **Error JUMP** button.



The error occurrence line is displayed.

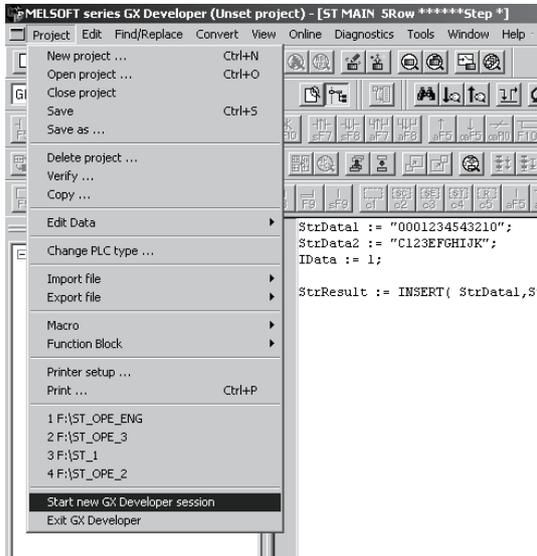
- (2) Diagnosis of the error part using the ST program (Operations 3), 4) in the flowchart)

Diagnose what should be corrected from the error occurrence line and error code, and make correction. For details, refer to the "QCPU (Q mode) Programming Manual (Structured Text)" given in Relevant Manuals. Since the cause of "error code: 4101" cannot be identified, perform the operations described in "(3) Display the sequence program where the error has occurred".

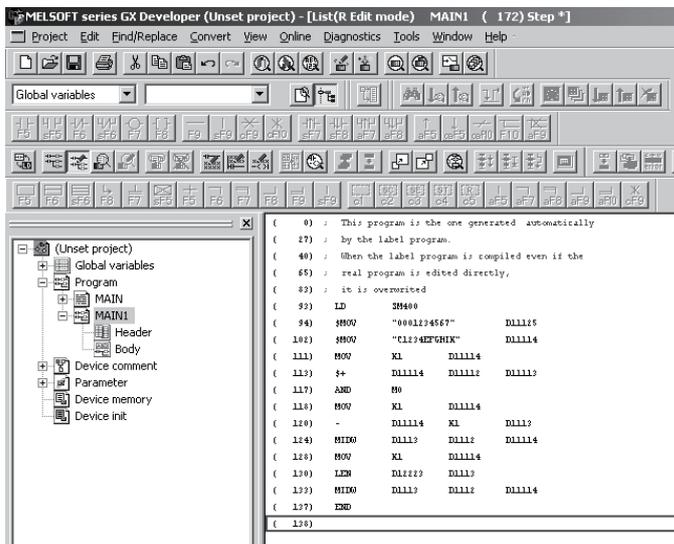
- (3) Display the sequence program where the error has occurred (Operations 5) to 8) in the flowchart)

To display the sequence program where the error has occurred, read the sequence program from the programmable controller.

- 5) Start GX Developer anew.



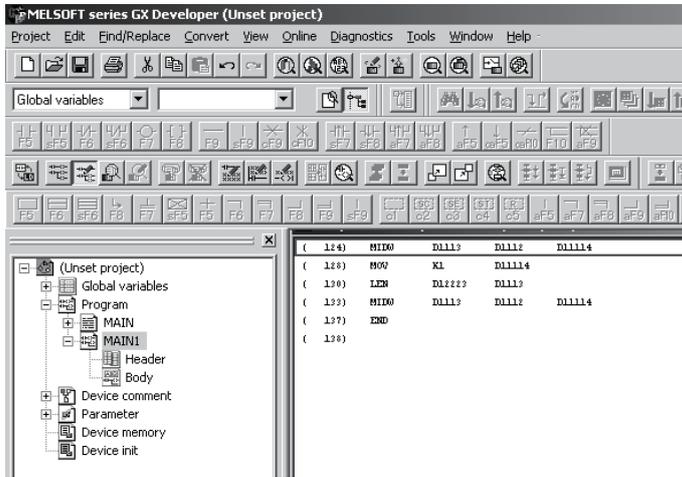
- 6) Perform read from PLC.



To next page



From previous page



7), 8) Click the **Error JUMP** button on the "PLC diagnostics" screen to display the corresponding step in the list.

(4) Diagnosis of the error part using the sequence program (Operations 9), 10) in the flowchart)

(a) Confirm the contents of the error code in the "QCPU Programming Manual (Common Instructions)" given in Relevant Manuals to identify the error cause.

There are the following causes for the occurrence of "error code: 4101" of the MIDW instruction.

MIDW S1 D S2

- i) The value of S2 is greater than the number of characters of D.
- ii) The value of S2 + 1 is greater than the number of characters of S1.

(b) Diagnose the part of the error cause in the ST program from the conversion result in the "QCPU (Q mode) Programming Manual (Structured Text)" given in Relevant Manuals, and correct the ST program.

Excerpt from the "QCPU (Q mode) Programming Manual (Structured Text)"

| ST Program                          | Conversion Result         |
|-------------------------------------|---------------------------|
| StrResult :=                        | LD SM400                  |
| INSERT (StrData1, StrData2, IData); | \$+ D11114 D11125 D11102  |
|                                     | AND<> D11113 K1           |
|                                     | MOV K1 D11100             |
|                                     | - D11113 K1 D11101        |
|                                     | MIDW D11125 D11102 D11100 |
|                                     | MOV D11113 D11100         |
|                                     | LEN D11114 D11101         |
|                                     | MIDW D11114 D11102 D11100 |

(c) Confirm the corresponding devices by performing device monitor, etc. to identify the error cause.

(5) Write the program to the programmable controller CPU and confirm that the error has been corrected. If the problem is not resolved, repeat the operations in 1) to 10) to correct the error.

## 4.4 Online Change

Change part of the sequence program and write it to the programmable controller CPU in a RUN status.

### [Purpose]

Used to perform write to PLC in a RUN status.

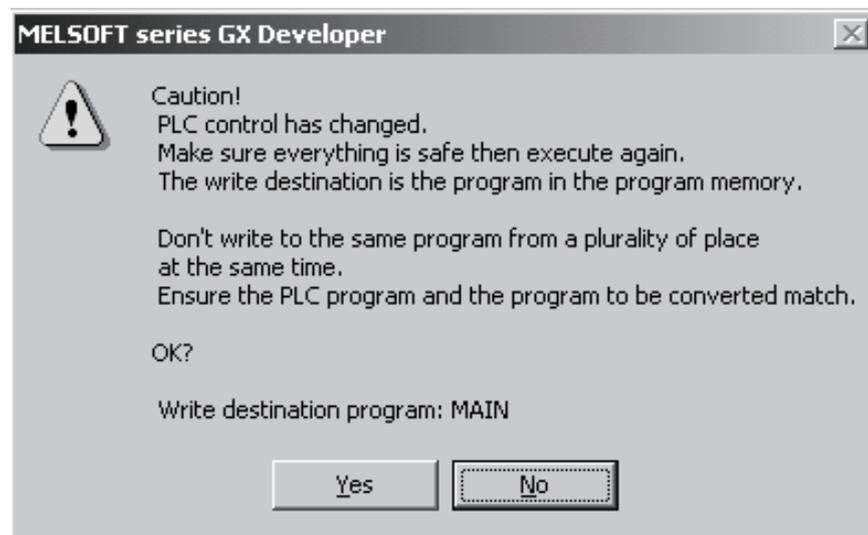
To indicate the online change target line, " \* " is displayed on the indicator bar.

### [Operating Procedure]

Choose [Convert] → [Convert/Compile (Online change)] or press **Shift** + **F4**.

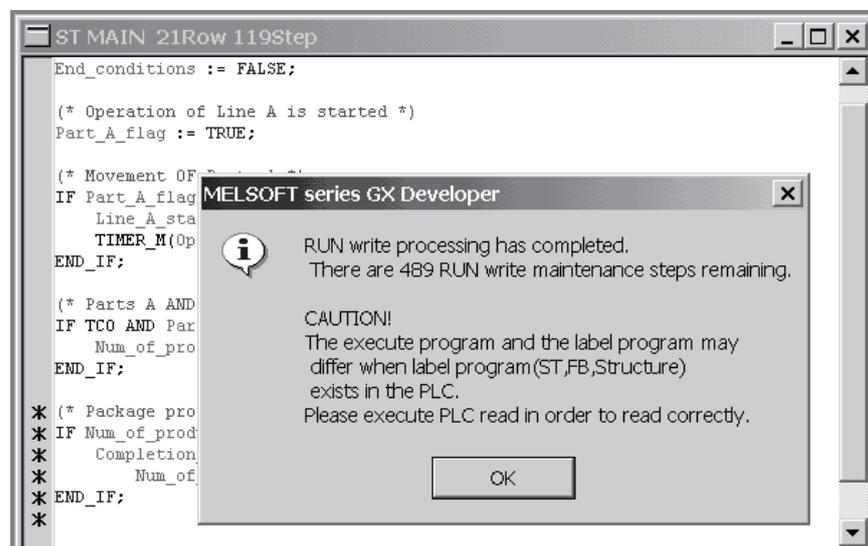
When online change is executed, the following online change confirmation message is displayed.

Execute online change after confirming the message.



### (1) When operation is completed normally

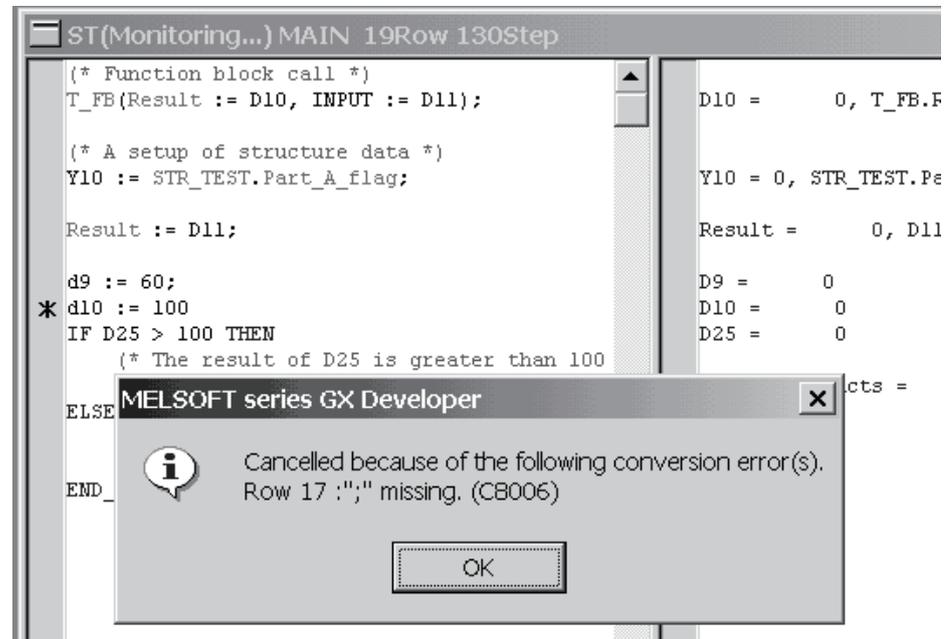
The " \* " displayed on the indicator bar on the target line disappears when the **OK** button is clicked.



## (2) When error has occurred

The error definition is displayed and the processing stops. After correcting the conversion error, execute online change again.

Example of error display (when "d10 := 100" is input)

**REMARK**

For details of the error display, refer to the "QCPU (Q mode) Programming Manual (Structured Text)" given in Relevant Manuals.

### 4.5 Device Test

Forcibly turn on/off the bit device of the programmable controller CPU or change the current value of the word device.

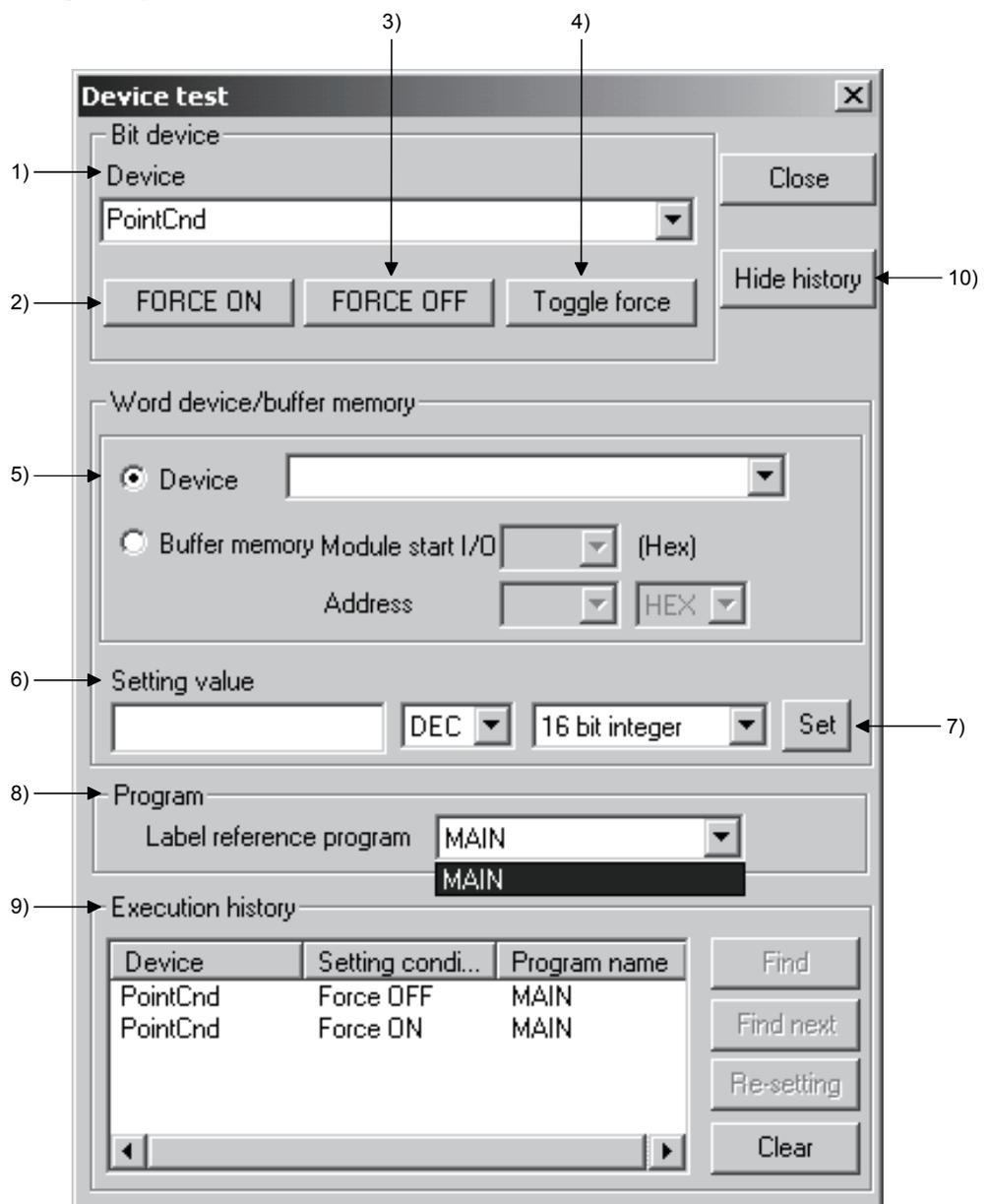
**[Purpose]**

Used to change the value of the specified device/variable (label, structure, device).

**[Operating Procedure]**

Choose [Online] → [Debug] → [Device test], click (  ), or press [Alt] + [1].

**[Dialog Box]**



**[Description]**

- 1) Device  
Specify the bit device to be forcibly turned on or off.
- 2) **FORCE ON** button  
Forcibly turns on the specified bit device.
- 3) **FORCE OFF** button  
Forcibly turns off the specified bit device.
- 4) **Toggle force** button  
Forcibly reverses the ON/OFF of the specified bit device.
- 5) Device  
Specify the word device whose current value will be changed.
- 6) Setting value  
Set a new value of the word device.
- 7) **Set** button  
Click after the setting is completed.  
The current value of the word device is changed.
- 8) Program  
Specify the program to be used in the device test.
- 9) Execution history  
The device test setting history is displayed.
- 10) Hide history (Execution result Display)  
The execution result of the device test is displayed/hidden.

**REMARK**

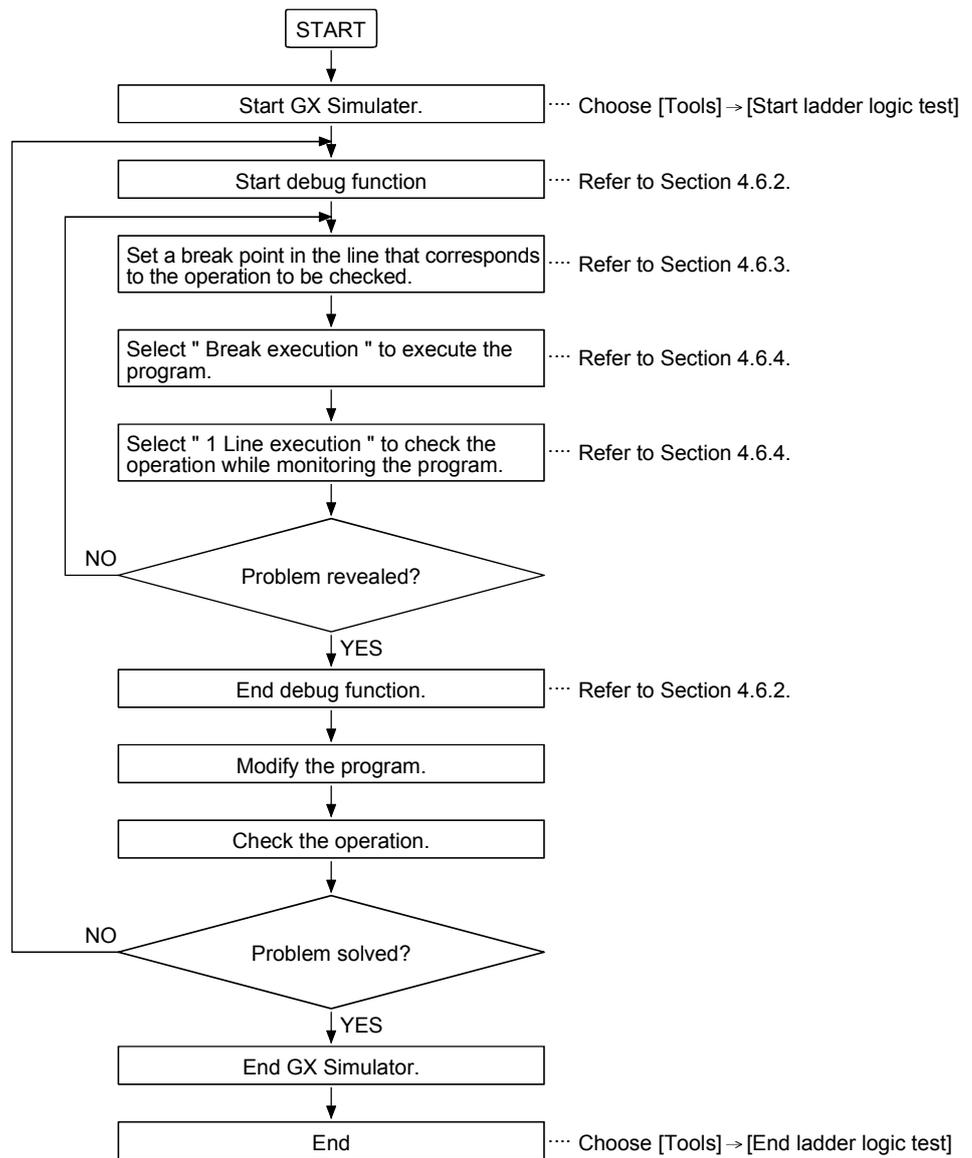
For details, refer to the "GX Developer Operating Manual" given in Relevant Manuals.

### 4.6 Debug Function

GX Simulator is required to execute debug function. Set break points within the program written in ST language, and execute it according to the set break points line-by-line. With this execution, the system operation can be monitored and checked.

#### 4.6.1 Debug function flowchart

The following flowchart shows an example for using debug function on GX Developer.





- GX Simulator Ver6.16S or later is required to use debug function.
- For details of GX Simulator, refer to GX Simulator Operating Manual.
- Debug is disabled when connected to programmable controller CPU

#### 4.6.2 Starting/Ending debug function

##### [Purpose]

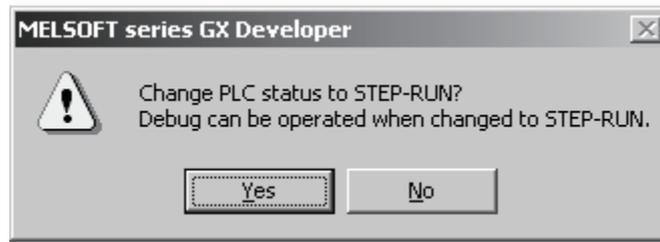
Start/End debug of the ST program.

##### [Operating Procedure]

Choose [Online] → [Debug] → [Debug].

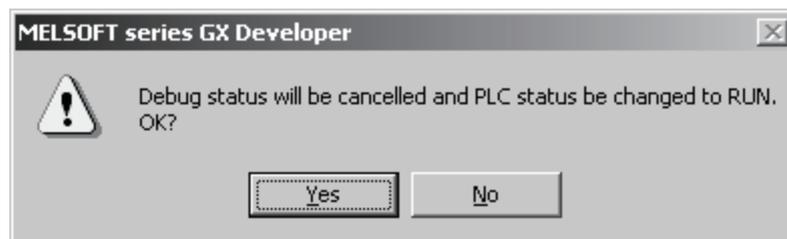
##### [Dialog Box]

- Start



Click **Yes** button to start debug function.

- End



Click **Yes** button. This displays a message telling that debug status is canceled.  
Click **OK** button to end debug function.

### 4.6.3 Setting/Clearing break points

Set/Clear break points.

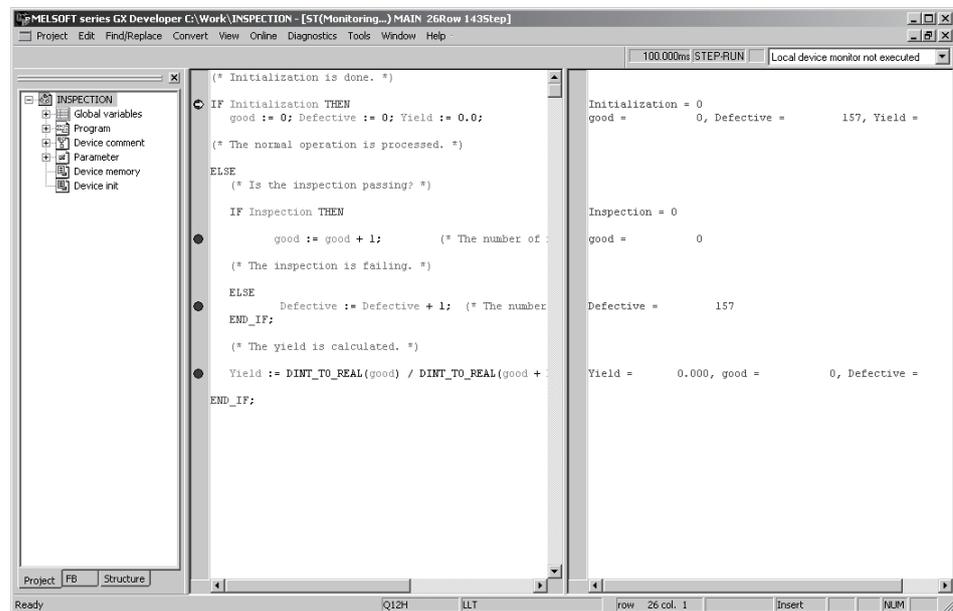
#### [Purpose]

Set break points in order that the execution will be halted at the specified location in a program during brake execution.

#### [Operating Procedure]

Choose [Online] → [Debug] → [Break point setting/cancellation], and click  or press **F9** button.

#### [Dialog Box]



#### [Setting procedure]

##### Break point setting

- 1) Place the cursor in the line where a break point to be set.
- 2) Choose [Online] → [Debug] → [Break point setting/cancellation], and click  or press **F9** button.
- 3)  appears on the indicator bar.

##### Break point clear

- 1) Place the cursor in the line where a break point is set.
- 2) Choose [Online] → [Debug] → [Break point setting/cancellation], and click  or press **F9** button.
- 3)  appears on the indicator bar.



- Up to 8 break points can be set within one program.

The following table shows the statements and the relevant restrictions on break point settings :

| No. | Control statement  | Break point setting (○: Available ×: N/A)   |
|-----|--------------------|---|
| 1   | IF                 | ○: The line that includes " THEN " *1<br>×: The line that includes " END_IF "   |
| 2   | CASE               | ○: The line that includes " OF " *2<br>○: The line that includes " : " (colon) right after an optional value *1<br>×: The line that includes " END_CASE " |
| 3   | FOR                | ○: The line that includes " DO " *1<br>×: The line that includes " FOR "<br>×: The line that includes " END_FOR "   |
| 4   | WHILE              | ○: The line that includes " DO " *1<br>×: The line that includes " END_WHILE "  |
| 5   | REPEAT             | ○: The line that includes " UNTIL " *1<br>×: The line that includes " REPEAT "<br>×: The line that includes " END_REPEAT "                                |
| 6   | EXIT               | ○: The line that includes " EXIT "<br>×: The line that includes " END_WHILE "   |
| 7   | RETURN             | ○: The line that includes " RETURN "<br>×: The line that includes " REPEAT "<br>×: The line that includes " END_REPEAT "                                  |
| 8   | Operation sentence | ○: The line that includes " ; " (semicolon) *3 at the end of a sentence   |
| 9   | FB utilization     | ○: The line that includes " ; " (semicolon) at the end of the control statement.  |
| 10  | FUNCTION           | ○: The line that includes " ; " (semicolon) at the end of the control statement.  |

\*1: Break point setting is available even when sentences within the control statement are all blank.

\*2: Break point setting is available. However, if the same variables (device, label) are used for the integer expression within " CASE <Integer expression> OF ", the break is executed in the line that includes " : " (colon) right after the first optional value instead of the line that includes " OF ".

\*3: Break point setting is not available when sentences within the control statement are all blank ( " ; " (semicolon) only).

## 4.6.4 Break execution/1 Line execution

Perform break execution/1 line execution of programs.

## [Purpose]

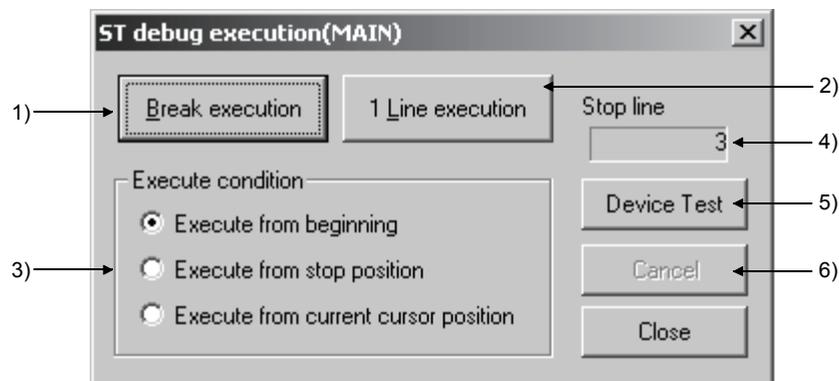
Break execution debugs programs by halting the program execution at the location specified by break point.

1 line execution debugs programs by halting the program execution line-by-line.

## [Operating Procedure]

Choose [Online] → [Debug] → [ST debug execution], and click  or press **F8** button.

## [Dialog Box]



## [Setting Items]

- 1) **Break execution** button  
Starts program execution from the location selected as "Execution condition" to the preset break point.
- 2) **1 Line execution** button  
Starts program execution line-by-line from the location selected as "Execution condition".
- 3) Execution condition  
Specify the line from which to start debugging.
- 4) Stop line  
Displays the line No. at which the program execution is being halted during break execution or 1 line execution.
- 5) **Device Test** button  
Displays "Device test" dialog box. For details, refer to Section 4.5.
- 6) **Cancel** button  
Interrupts break execution or 1 line execution.



· Break execution and 1 line execution can be performed by pressing **ALT + B** , **ALT + L** , respectively.  
This is available while "ST debug execution" dialog box is on the screen.

### 4.6.5 Break point list

A list that displays the set break points.

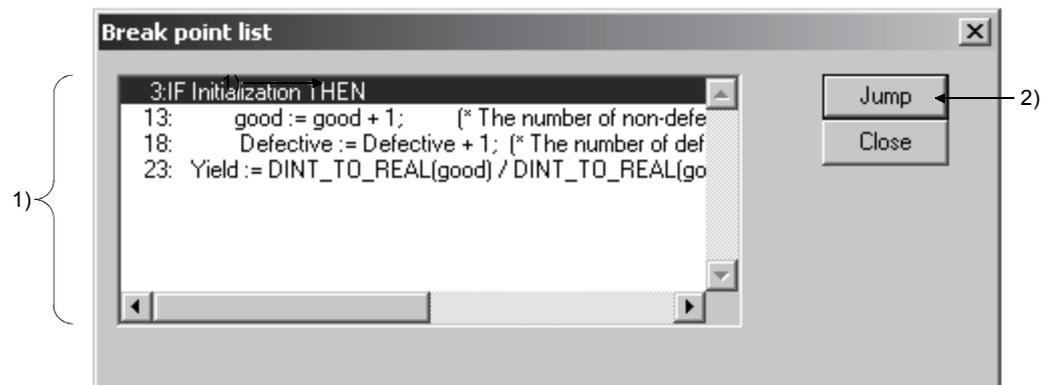
#### [Purpose]

Use the break point list to jump to the other line.

#### [Operating Procedure]

Choose [Online] → [Debug] → [Break point list].

#### [Dialog Box]



#### [Setting Items]

##### 1) Break point list box

This list box displays the line No. and statements at which break points are set.

Select a line and double click it (or then press **Enter**) to perform the same as

**Jump** button.

##### 2) **Jump** button

Click this button to move to the head of the line selected in the break point list box.

### 4.6.6 Clearing all break points

Clear all break points.

#### [Purpose]

Clear all break points.

#### [Operating Procedure]

Choose [Online] → [Debug] → [Cancel all break points].

#### [Dialog Box]



Click **Yes** button to clear all break points.

### 5 PRINT

This chapter explains the method of printing the ST program.  
For other printing methods, refer to the " GX Developer Operating Manual" given in Relevant Manuals.

#### (1) PRINT

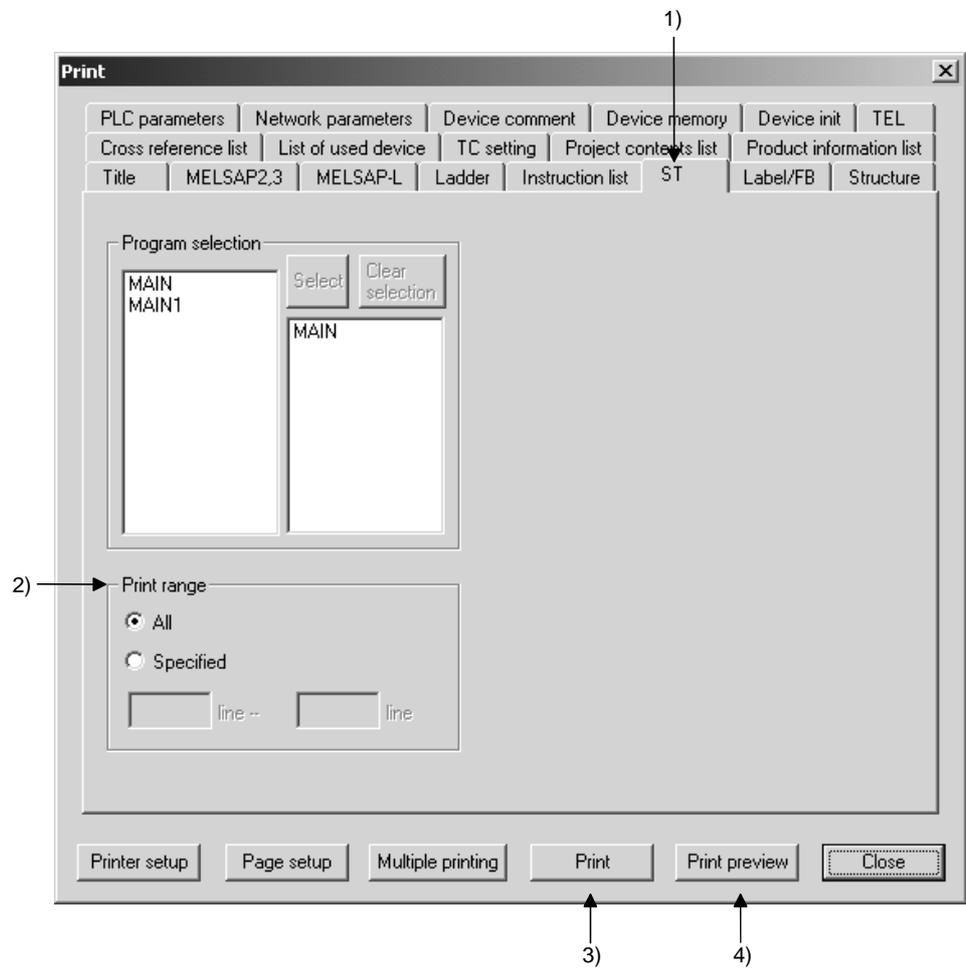
##### [Purpose]

Print the ST program.

##### [Operating Procedure]

Choose [Project] → [Print], click (  ), or press **Ctrl** + **P**.

##### [Dialog Box]



**[Description]**

- 1) <<ST>> tab  
Switches to the screen where print details will be set.
- 2) Print range  
Specify the print range of the ST program.
  - Entire range  
The ST program is printed from the first line to the last line.
  - Range specification  
The ST program is printed in the specified range.
- 3) **Print** button  
Prints the ST program.
- 4) **Print preview** button  
Displays the preview screen.

**[Setting procedure]**

After setting the information required for **Print**, click the Print button to start printing.

**(Print Example)**

```

1 Start_conditions := TRUE;
2 End_conditions := FALSE;
3
4 (* Operation of Line A is started *)
5 Part_A_flag := TRUE;
6
7 (* Movement OF Parts A *)
8 IF Part_A_flag AND Start_conditions THEN
9   Line_A_start := TRUE;
10  TIMER_M(Operation_lamp, TC0, K30);
11 END_IF;
12
13 (* Parts A AND Parts B are assembled *)
14 IF TC0 AND Part_B_flag THEN
15   Num_of_products := Num_of_products + 1;
16 END_IF;
17
18 (* Package processing *)
19 IF Num_of_products >= 10 THEN
20   Completion_flag := TRUE;
21   Num_of_products := 0;
22 END_IF;

```



- The line numbers are printed in serial numbers.
- If print cannot be performed midway through characters, a line feed is executed in that position.  
The line number is not provided for the part where the line feed was executed.
- When the print range is specified, print starts from the specified line number.
- The number of characters on one line changes depending on the printer setting and font.

**REMARK**

For the print of an FB program, refer to the "GX Developer Operating Manual (Function Block)" given in Relevant Manuals.

## (2) Print preview

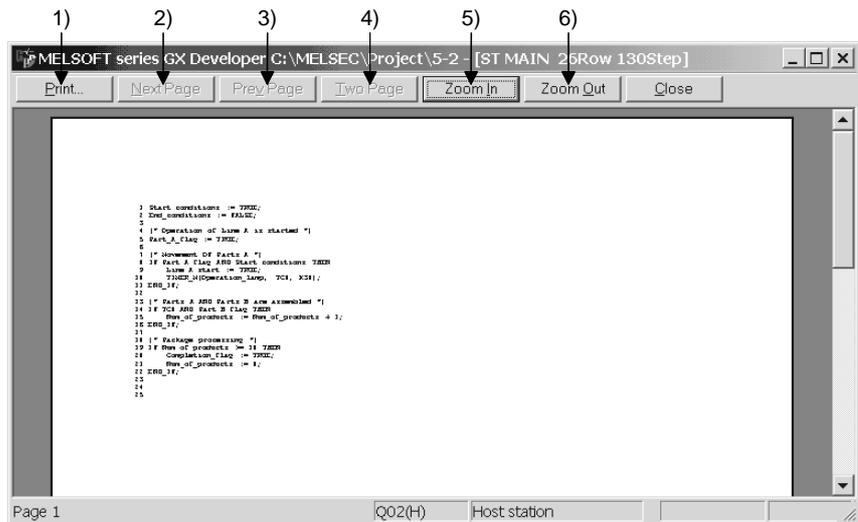
## [Purpose]

Display the print image of the ST program whose print range has been set.

## [Operating Procedure]

Choose [Project] → [Print] → **Print preview** button,  
click (  ) → **Print preview** button, or press **Ctrl** + **P** → **Print preview** button.

## [Dialog Box]



## [Description]

- 1) **Print** button  
Prints the data displayed on the print preview screen.
- 2) **Next Page** button  
Displays the print image of the next page.
- 3) **Prev Page** button  
Displays the print image of the previous page.
- 4) **Two Page** button  
Selects whether the print image will be displayed on a one page basis or two page basis.
- 5) **Zoom In** button  
Displays an enlarged print image on the screen.  
The display can be switched in three stages.
- 6) **Zoom Out** button  
Displays a reduced print image on the screen.

## INDEX

- [A]  
Auto indent function ..... 1- 3, 3-10
- [B]  
Bookmark  
Bookmark list ..... 3-14  
Deletion of all bookmarks ..... 3-13  
Finding the bookmark line ..... 3-13  
Setting/deletion of bookmark ..... 3-12  
Using the bookmark ..... 3-12
- [C]  
Comment  
Creating a comment ..... 3- 3, 3- 8  
Control syntax upper case conversion function  
Control syntax upper case conversion function ...  
..... 3- 3, 3- 9  
Target characters ..... 3- 9  
Convert (Compile)  
Conversion (Compile)..... 3-28  
Error part indication mark ..... 3-29  
Copy..... 3-11, 3-26  
Custom  
Auto indent..... 3-31  
Auto indent check box ..... 3-31  
Change display color ..... 3-32  
Changing the tab width..... 3-31  
Font ..... 3-34  
ST editor settings..... 3-31  
Tab width combo box ..... 3-31  
Cut ..... 3-11, 3-26
- [D]  
Device test..... 4-13  
Debug function  
Debug function flowchart..... 4-15  
Starting/Ending debug function ..... 4-16  
Setting/Clearing break points ..... 4-17  
Break execution/1 line execution ..... 4-19  
Break point list ..... 4-20  
Clearing all break points ..... 4-20
- [E]  
Enter ..... 3-11, 3-18
- [F]  
FB definition ..... 2- 1  
FB screen  
Open Function Block..... 3-11, 3-24  
FB variable definition ..... 2- 1  
Find  
Find ..... 3-11, 3-18  
Find string ..... 3-18  
Leave comments ..... 3-19  
Match case ..... 3-18  
Match whole word only ..... 3-18  
Set bookmark ..... 3-20  
Function  
Displaying a function parameter ..... 3-15  
Entering a function ..... 3- 3, 3- 4  
Function classification list box ..... 3- 4  
Function list box ..... 3- 5  
Select function ..... 3- 4
- [G]  
Go ..... 3-11, 3-18
- [H]  
Home ..... 3-11, 3-18
- [I]  
Insert ..... 3-11, 3-18
- [L]  
Label  
Displaying the label information ..... 3-17  
Entering a label ..... 3- 3  
label list ..... 3- 6  
Select label ..... 3- 6  
Label setting ..... 3- 2  
Line jump  
Line jump ..... 3-11  
Line setting edit box ..... 3-23
- [M]  
Monitor  
Monitor screen ..... 1- 6  
Monitoring ..... 4- 5
- [O]  
Online change ..... 4- 11
- [P]  
Paste ..... 3-11, 3-26  
programmable controller CPU ..... 1- 9  
Print  
Print ..... 5- 1  
Print preview ..... 5- 3

Project

- Copying the project..... 3- 2
- New project..... 3- 1
- PLC series ..... 3- 1
- PLC Type..... 3- 2
  
- Program type ..... 3- 2
- Reading the project ..... 3- 2
- Project Window ..... 1- 6

[R]

- Read from PLC..... 4- 1
- Redo ..... 3-11, 3-27
- Replace
  - Find in ..... 3-21
  - Find what ..... 3-21
  - Replace..... 3-11, 3-21
  - Replace all ..... 3-21
  - Replace with ..... 3-21

[S]

- ST edit screen
  - Indicator bar ..... 1- 6
  - Main menu ..... 1- 6
  - Shortcut Key ..... 1- 7
  - Status bar..... 1- 6
  - Toolbar ..... 1- 6, 1- 7
- ST language ..... 1- 1
- ST Programming ..... 3- 1

[T]

- Text format ..... 1- 1

[U]

- Undo ..... 3-11, 3-27

[W]

- Window division..... 3-11, 3-16
- Write to PLC ..... 4- 3

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

Pentium is a registered trademark of Intel Corporation in the United States and other countries.

Other company and product names herein are either trademarks or registered trademarks of their respective owners.

SPREAD

Copyright (C) 1998 Farpoin Technologies, Inc.



# *GX Developer Version 8*

## Operating Manual (Structured Text)

|                           |               |
|---------------------------|---------------|
| MODEL                     | GXDEV8-O-ST-E |
| MODEL CODE                | 13JU37        |
| SH(NA)-080367E-E(0812)MEE |               |



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

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

Specifications subject to change without notice.