GX Developer Version 8

Operating Manual

(Function Block)

MITSUBISHI

MELSOFT

*O*peration

*M*aintenance

*P*rogramming

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.

[Design 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

Print Date	* Manual Number	Revision
Feb., 2003	SH (NA)-080376E-A	First edition
Oct., 2003	SH (NA)-080376E-B	Correction Section 3.13, Section 5
Jun., 2004	SH (NA)-080376E-C	Correction Abbreviations and Generic Terms in This Manual
Sep., 2004	SH (NA)-080376E-D	Addition Section 3.6.3
Dec., 2004	SH (NA)-080376E-E	Correction Section 3.6.3
Aug., 2005	SH (NA)-080376E-F	Correction Section 3.13
Dec., 2005	SH (NA)-080376E-G	Correction Section 1.5.1, Section 1.5.2, Section 3.3
Mar., 2006	SH (NA)-080376E-H	Correction Section 1.1, Section 3.9, Section 3.10
Oct., 2006	SH (NA)-080376E-I	Correction Section 8.1, Section 8.2
Jul., 2007	SH (NA)-080376E-J	Correction Abbreviations and Generic Terms in This Manual, Section 3.7.3
Dec., 2008	SH (NA)-080376E-K	Correction Abbreviations and Generic Terms in This Manual, Section 1.5.1, Section 1.5.2

Japanese Manual Version SH-080359-L

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 MELSOFT series
programmable controller thoroughly in advance to ensure correct use.

CONTENTS

1 OVERVIEW   1- 1 to 1- 9

2 GENERAL PROCEDURES FOR EDITING THE FB   2- 1 to 2- 2

3 CREATING NEW SEQUENCE PROGRAM USING FB   3- 1 to 3-44

4 UTILIZING THE EXISTING FB TO CREATE SEQUENCE PROGRAM   4- 1 to 4- 9

5 DELETING UNNECESSARY FB DEFINITION FROM THE PROJECT   5- 1 to 5- 2

6 COMBINING MULTIPLE PROGRAMS INTO ONE   6- 1 to 6- 2

7 VERIFYING THE PROJECTS   7- 1 to 7- 2

8 PRINTING   8- 1 to 8- 5

APPENDICES   Appendix- 1 to Appendix - 2

INDEX   Index - 1 to Index - 2

About Manuals

The following manuals are also related to this product.

In necessary, order them by quoting the details in the tables below.

Related Manuals

Manual Name	Manual Number (Model Code)
GX Developer Version 8 Operating Manual (Startup) Describes the system configuration, installation method and starting method of GX Developer. (Option)	SH-080372E (13JU40)
GX Developer Version 8 Operating Manual Explains the functions of the programming, printout, monitoring and debugging methods and so on GX Developer. (Option)	SH-080373E (13JU41)
GX Developer Version 8 Operating Manual (Structured Text) Describes the structured text (ST) program creation methods, print-out methods, etc. using GX Developer.	SH-080367E (13JU37)
QCPU(QMode) Programming Manual (Structured Text) Describes the structured text language programming methods.	SH-080366E (13JF68)
Structured Text (ST) Programming Guide Book Written for those who will create structured text (ST) programs for the first time. Describes the basic operation methods and functions through sample programs.	SH-080368E (13JF69)

REMARK

The above manuals are contained in the CD-ROM together with the software package as a set.
If you want the manuals separately, they are available optionally as printed matter.
When placing an order, please indicate the manual numbers (type codes) given in the above table.

Abbreviations and Generic Terms in This Manual

This manual uses the following generic names and abbreviations to represent the GX Developer software package and programmable controller CPU modules. The module type is given when the corresponding type must be indicated clearly.

Generic Term/Abbreviation	Description
GX Developer	Generic product name of product types SW8D5C-GPPW, SW8D5C-GPPW-A, SW8D5C-GPPW-V and SW8D5C-GPPW-VA.
FB	Abbreviation of the function block.
ST	Abbreviation of a structured text.
QnACPU	Generic name of the MELSEC-QnA series programmable controller CPUs.
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.
ACPU	Generic name of the MELSEC-A series programmable controller CPUs. It includes the QCPUs (A mode) and motion controllers (SCPU).
QCPU (A mode)	Generic name of the Q02(H)-A and Q06H-A.
FXCPU	Generic name of the MELSEC-FX series programmable controller CPUs. (The target programmable controllers are the FX0, FX0S, FX0N, FX1, FX2, FX2C, FX1S, FX1N, FX2N and FX2NC.)
QnA series	When GX Developer is started on the QnACPU.
Q series	When GX Developer is started on the QCPU (Q mode).

1 OVERVIEW

This manual describes the operations of editing the function block (hereafter abbreviated to the FB) of the GX Developer Version 7 software package (hereafter abbreviated to GX Developer).
For the explanation of the functions other than the FB, refer to the corresponding manuals given in the "Related Manuals".

1.1 What Is Function Block (FB)?

The FB is a function designed to convert a ladder block, which is used in a sequence program repeatedly, into a component (FB) to utilize it in the sequence program. This not only increases the efficiency of program development but also reduces programming mistakes to improve program quality.

Definition of FB
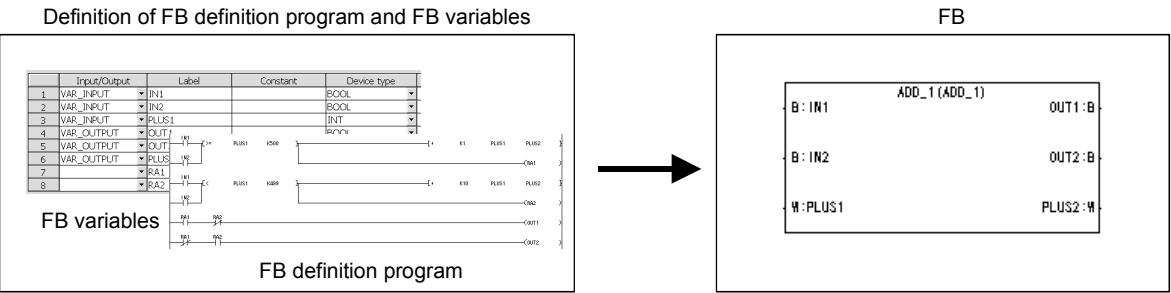
Utilization of FB
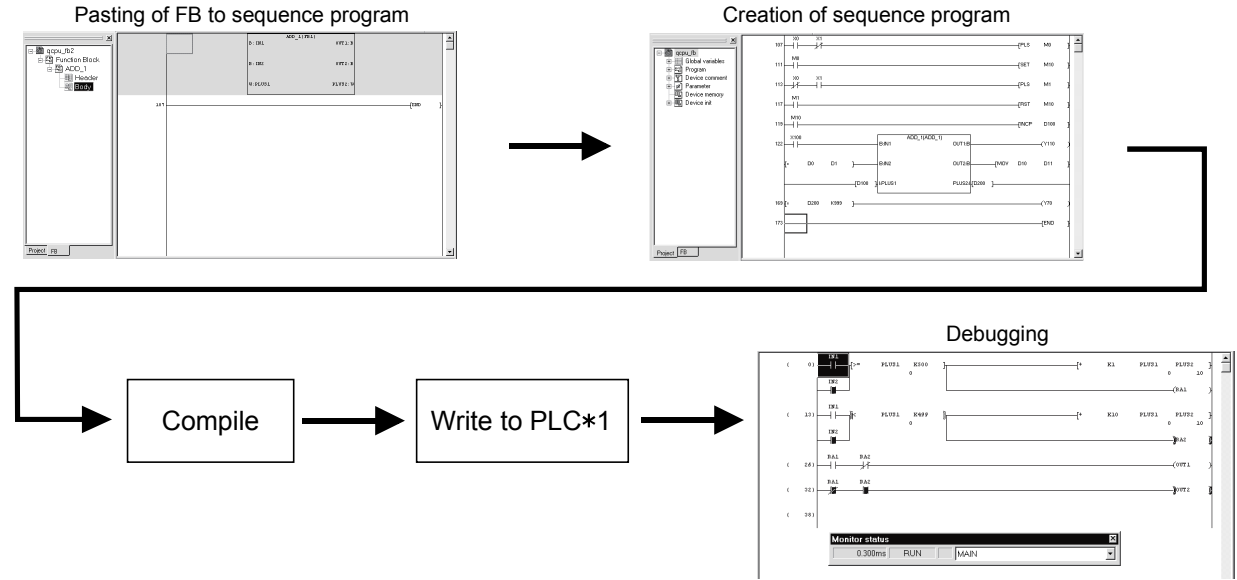
∗1: GX Developer performs write to PLC to write the compiled sequence program to the programmable controller CPU.
When writing the program to a programmable controller with "Label program (ST, FB, Structure)" selected, the FB can be restored using "Read from PLC".

1

1.2 Features

The FB has the following features.

(1) Conversion of program into component
Displaying a standard program as a single block improves the reuse and readability of the program, facilitating editing and debugging. This ensures ease of configuring a large-scale program.

(2) Uniform program quality
Reusing the components of a standard program provides uniform program quality, which is independent of the technological level of a program developer. It also prevents programming mistakes during utilization.

(3) Editing function of high operability
The FB can be utilized easily in a sequence program by drag and drop operation.

1.3 Layouts and Names of Windows for Creating FB

This section describes the layouts and names of the windows used to create the FB.

(1) Window used to create FB definition program (FB definition window)
This window is used to edit an FB definition program, which is created in a ladder or list format.
Labels define variable types, device types and others on the FB variable setting window in (2).

<Ladder program>

FB definition

FB variables

Program body

FB definition program

Function Block tab (FB tab)

FB definition window

<ST Program>
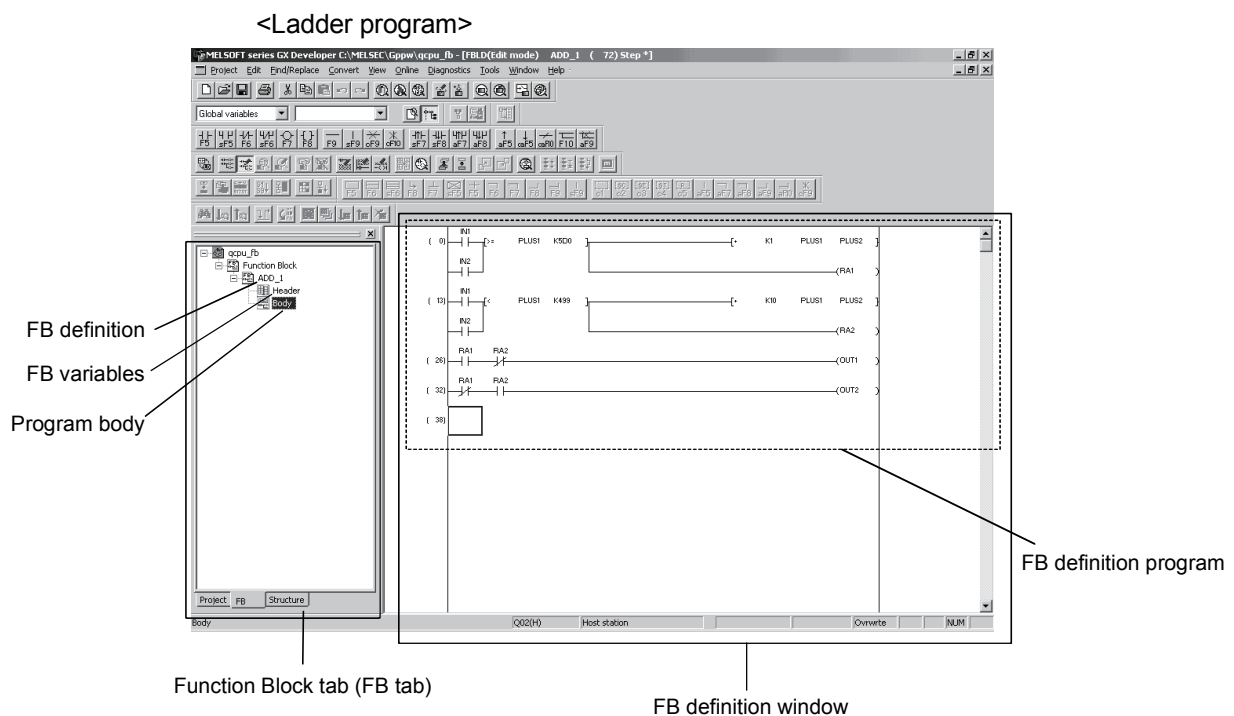
FB definition program
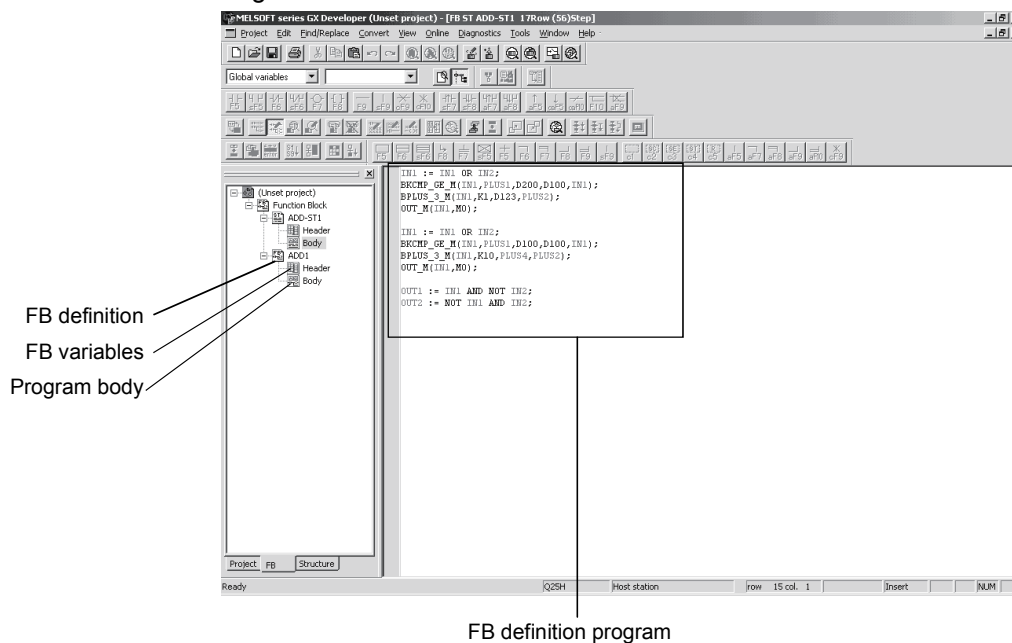
(2) Window used to create FB definition labels (variables) (FB variable setting window)

This window is used to define the labels used for the FB definition.

FB variable setting window
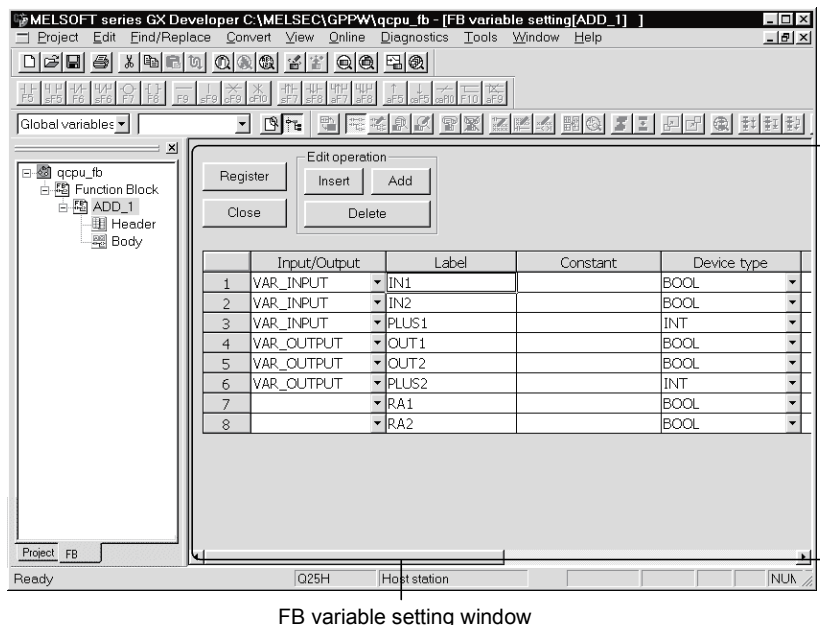
(3) Window used to create a sequence program
     (Sequence program window)
     This window is used to edit a sequence program where the FB is pasted.
     Create the sequence program in a ladder format. A list format cannot be used.

Project data list   ＊1   Sequence program window   Input ladder section
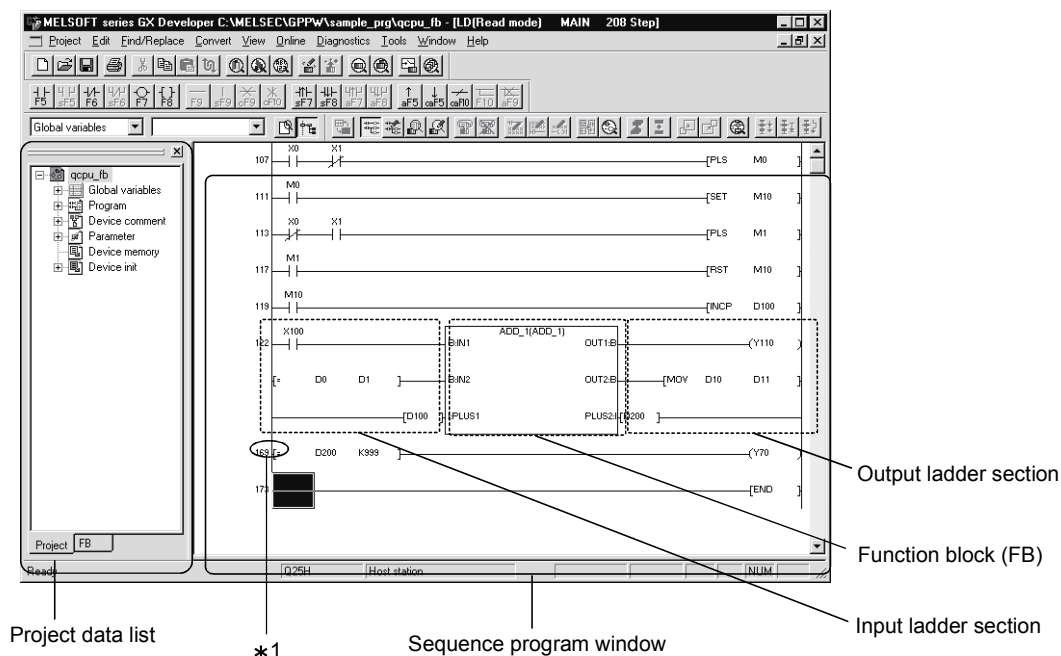
＊1: The step numbers of the sequence program, which is not yet compiled, are
     displayed with parentheses, i.e. (Step number).
     After completion of compile, the step numbers are displayed without parentheses.
     When compile is complete, the step numbers may be changed.

Name	Description
FB tab	Tab that displays the FB definition of the project data list.
FB variables (Header)	Label used for the FB definition.
FB definition	Data having a Body and Header as a pair.
FB definition window	Window used to edit an FB definition program.
FB definition program (body)	Program whose labels have been created on the FB definition window.
FB variable setting window	Window used to set FB variables.
Project tab	Table that displays project information in the project data list.
FB	Function block pasted to a sequence program.
FB section	Square symbol that represents the FB pasted to the sequence program.
Input ladder section	Input ladder on the left side of the FB section.
Output ladder section	Output ladder on the right side of the FB section.
Sequence program window	Window used to edit the sequence program where the FB is pasted.
FB definition name	Name of the FB definition displayed on the FB tab
FB name	Name of the FB pasted to the sequence program. Different names are attached to the FBs that are pasted from the same FB definition.
FB window	Window used to read or monitor the program in the FB pasted to the sequence program. The FB window cannot be used to write a program. Use the FB definition window to write a program.

1.4 What You Should Understand Before Using the FB

As in a normal sequence program, the FB is used to create a program using labels in a ladder or list format of a ladder program or in a ST program.
Before creating the FB, therefore, fully understand the way to create a sequence program using labels on GX Developer.
When you want to know any operation to create a sequence program on GX Developer, refer to the "GX Developer Version 8 Operating Manual" given in the related manuals.

1.5 Specifications

1.5.1 Compatible CPUs

The following CPU models allow you to use the FB on GX Developer.

PLC Series	Type
QCPU (Q mode)	Q00JCPU, Q00UJCPU, Q00CPU, Q00UCPU, Q01CPU, Q01UCPU, Q02CPU, Q02HCPU, Q02PHCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q04UDHCPU, Q04UDEHCPU, Q06HCPU, Q06PHCPU, Q06UDHCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q12HCPU, Q12PHCPU, Q12PRHCPU, Q13UDHCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q25HCPU, Q25PHCPU, Q25PRHCPU, Q26UDHCPU, Q26UDEHCPU
QnACPU	Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q2ACPU, Q2ACPU-S1, Q3ACPU, Q4ACPU, Q4ARCPU

The FB is not available for the ACPU, QCPU (A mode), FXCPU and motion controller of the PLC series, either.

1.5.2 Specifications of and precautions for the FB

The following are the specifications of the FB and precautions for use of the FB.

(1) For creation of FB definition program
(a) Specifications

Item	Specifications
Program capacity	2048 steps (For ST Program: FB Definition Convert, maximum 2048 steps)
Number of labels usable as devices	500 pcs. • Input variables + I/O variables = 24 pcs. • Output variables + I/O variables = 24 pcs. (At least one input variable and one output variable are needed.) • Labels other than the input, output and I/O variables are used as internal variables.
Programming language	Ladder, list,ST

(b) Precautions
1) The FB cannot be used within the FB.
2) Index qualification is not available for labels.
   To make index qualification, use an actual device as the target device.
3) If an internal variable is used as the result of a division instruction, its remainder cannot be used for control since a label cannot be assigned to the remainder of division.
   To use the remainder of a division instruction for control, use an actual device as the device for storing the division result.
4) Since the numerical device that may be handled by an input variable is up to a double word, the instruction exceeding three words in input count cannot be input from the outside.
   To input three or more words from the outside, use an actual device as the device for storing an input value or create the input instructions by the number of necessary times.
5) Since the numerical device that may be handled by an output variable is up to a double word, the instruction whose result will be three or more words cannot be output to the outside.
   To output three or more words to the outside, use an actual device as the device for storing the result.

(2) For creation of sequence program including FB
(a) Specifications

Item	Specifications
Language of sequence program that can be created	Ladder,ST
Pastable FB	Already compiled FB If you attempt to paste the FB not yet compiled, GX Developer displays an error message.
Number of FBs that may be pasted to one ladder block	1 pc. The output from the FB cannot be connected directly to the input of the other FB. If you want to connect the FBs, receive the output from the FB with a coil once, and then connect the contact of that coil to the input of the other FB.

(b) Precautions
1) A return connection cannot be used with an input ladder section and an output ladder section.

・Return connection cannot be programmed.
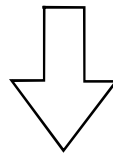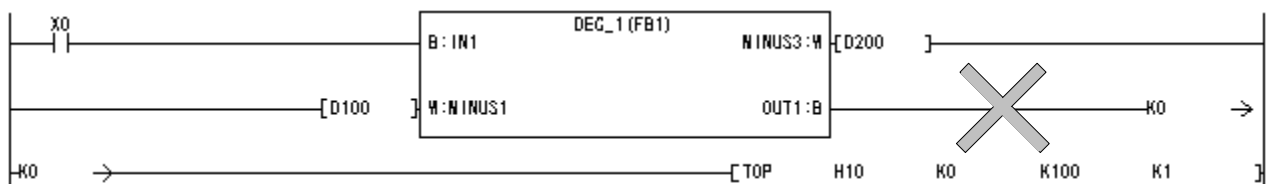
・Receive it with a coil once and then use the contact of that coil as the condition of the target instruction.
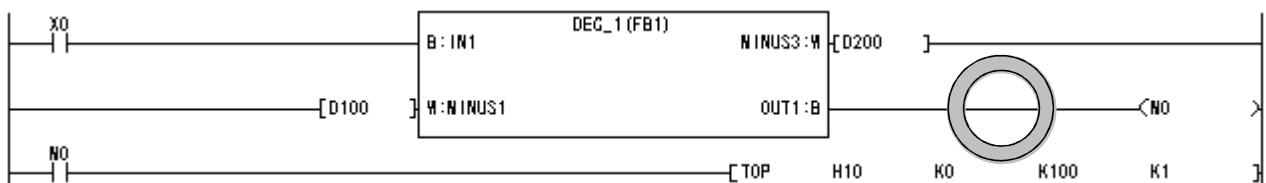
2) A parallel connection cannot be used with an input ladder section and an output ladder section as it causes an error at the ladder conversion.
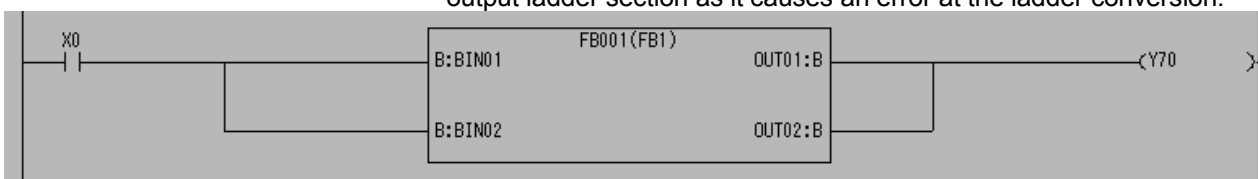
3) The FB cannot be used in SFC and MELSAP-L programs.

4) If the device type of the FB differs from that of the input or output ladder
   section, GX Developer displays an error message.
   There are the following five FB device types.
   • BOOL   : Data represented by ON/OFF.
   • INT   : Data represented by 16 bits.
   • DINT   : Data represented by 32 bits.
   • REAL   : Floating-point data represented by 32 bits.
   • STRING   : Character string data represented by JIS8 code.
     For example, connection of a contact to the input of the FB whose
     device type is a word will result in an error.

5) The FB definition automatically assigns devices to the labels used in a
   program. When creating a sequence program, avoid automatically
   assigned devices.
   Like local devices, the automatically assigned devices are set in
   accordance with the automatically assigned device setting, which
   defaults to the following device ranges.
   • Word device : D6144 to D12287
   • Bit device   : M4096 to M8191
   • Timer   : T64 to T2047
   • Counter   : C512 to C1023

MEMO

2 GENERAL PROCEDURES FOR EDITING THE FB

The following flowcharts give the general procedures from project creation to debugging for use of the FB.

(1) Creating new sequence program using FB

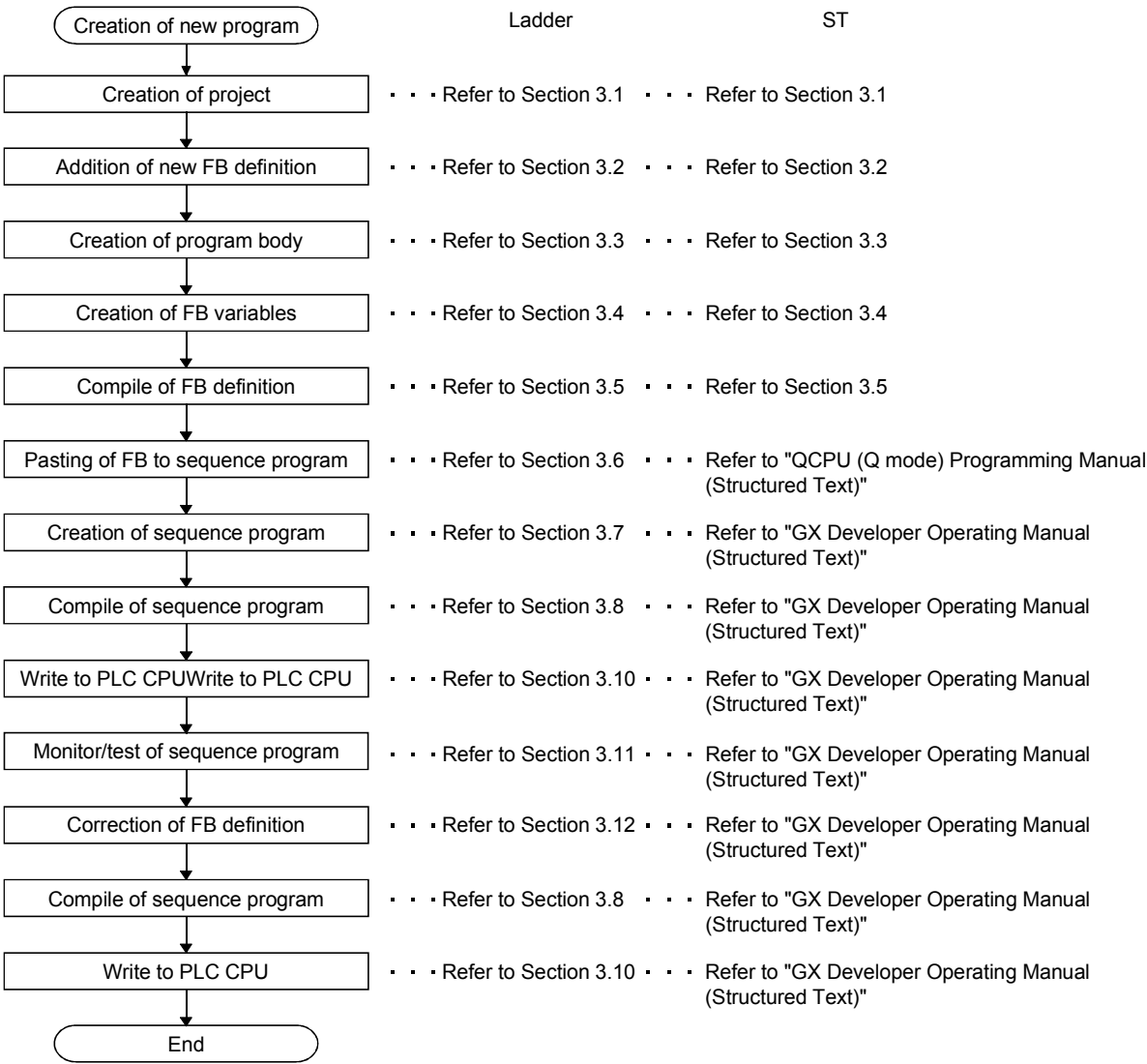|	Ladder	ST
Creation of new program		
Creation of project	· · · Refer to Section 3.1	· · · Refer to Section 3.1
Addition of new FB definition	· · · Refer to Section 3.2	· · · Refer to Section 3.2
Creation of program body	· · · Refer to Section 3.3	· · · Refer to Section 3.3
Creation of FB variables	· · · Refer to Section 3.4	· · · Refer to Section 3.4
Compile of FB definition	· · · Refer to Section 3.5	· · · Refer to Section 3.5
Pasting of FB to sequence program	· · · Refer to Section 3.6	· · · Refer to "QCPU (Q mode) Programming Manual (Structured Text)"
Creation of sequence program	· · · Refer to Section 3.7	· · · Refer to "GX Developer Operating Manual (Structured Text)"
Compile of sequence program	· · · Refer to Section 3.8	· · · Refer to "GX Developer Operating Manual (Structured Text)"
Write to PLC CPUWrite to PLC CPU	· · · Refer to Section 3.10	· · · Refer to "GX Developer Operating Manual (Structured Text)"
Monitor/test of sequence program	· · · Refer to Section 3.11	· · · Refer to "GX Developer Operating Manual (Structured Text)"
Correction of FB definition	· · · Refer to Section 3.12	· · · Refer to "GX Developer Operating Manual (Structured Text)"
Compile of sequence program	· · · Refer to Section 3.8	· · · Refer to "GX Developer Operating Manual (Structured Text)"
Write to PLC CPU	· · · Refer to Section 3.10	· · · Refer to "GX Developer Operating Manual (Structured Text)"
End		

POINT

This general procedure is for reference.
Each operation can be performed in any order, independently of the above procedure.

2

MEMO

3 CREATING NEW SEQUENCE PROGRAM USING FB

3

The following explains the operations to be performed to create a new sequence program uing the FB.
This chapter is designed to create the following FB definition (program body, FB variables) and sequence program.

(1) FB definition (program)
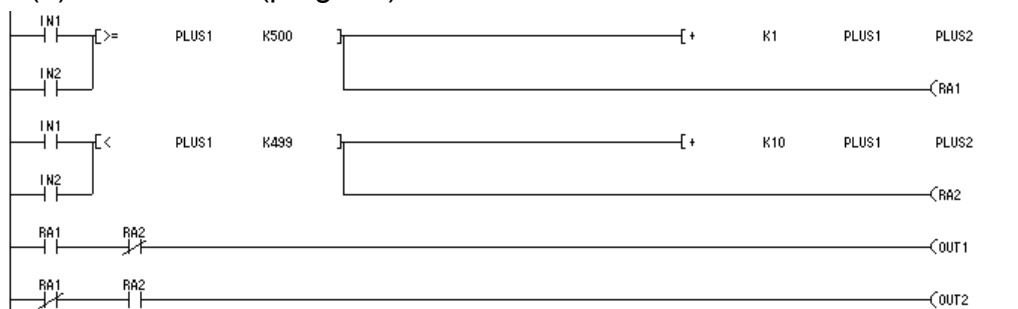
(2) FB definition (FB variables)

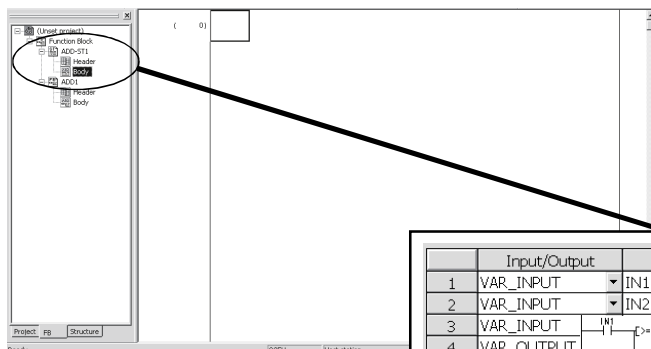|	Input/Output		Label	Constant	Device type	
1	VAR_INPUT	▼	IN1		BOOL	▼
2	VAR_INPUT	▼	IN2		BOOL	▼
3	VAR_INPUT	▼	PLUS1		INT	▼
4	VAR_OUTPUT	▼	OUT1		BOOL	▼
5	VAR_OUTPUT	▼	OUT2		BOOL	▼
6	VAR_OUTPUT	▼	PLUS2		INT	▼
7		▼	RA1		BOOL	▼
8		▼	RA2		BOOL	▼

(3) Sequence program
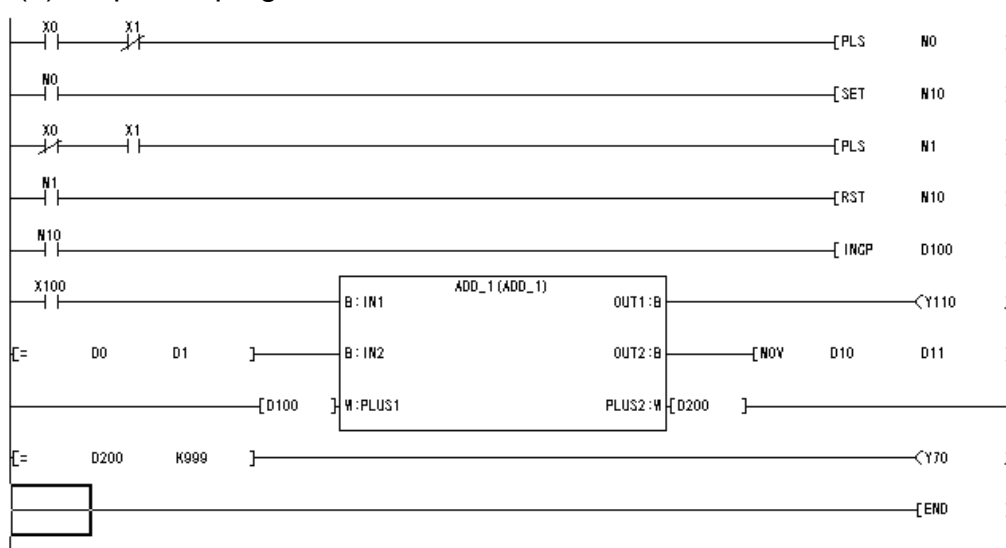
The sample project (qcpu_fb) of the sequence program described in this chapter is contained in the CD-ROM of this product.
After installing GX Developer, choose [Project]-[Open project] to open the sample project.

Handle the FB as described below.
1) Create a FB definition in the FB tab of the project.
2) Create the program body and FB variables of the FB definition.
3) Paste the FB definition registered to the FB tab to the sequence program.
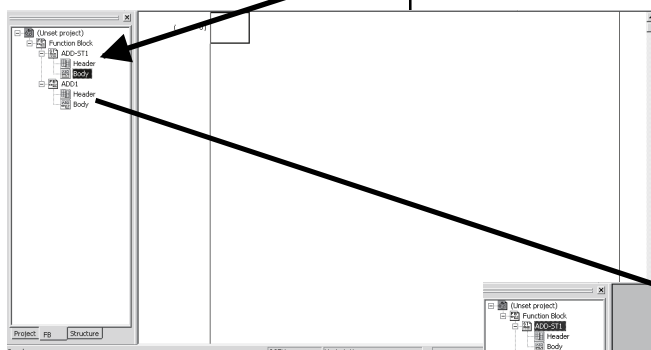4) Create the ladder block including the FB after pasting the FB.
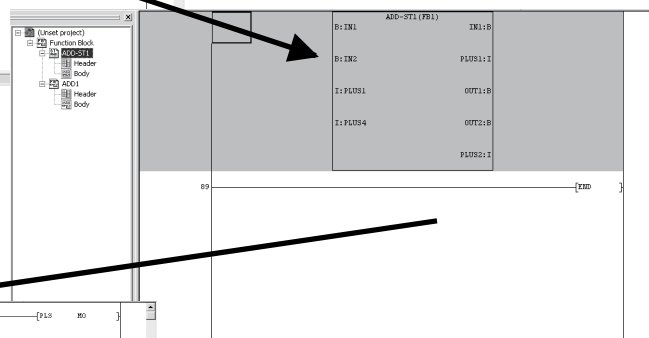
Creation of new FB definition
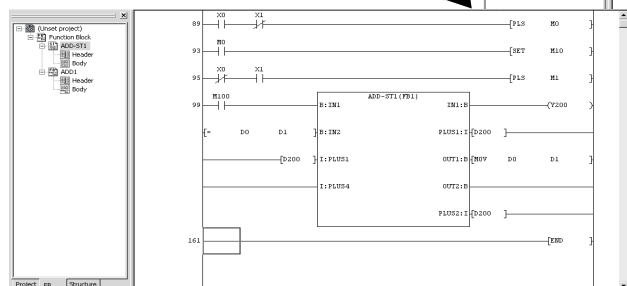

Creation of program body and FB variables

Registration of program body and FB variables

Pasting of FB to sequence program

Creation of sequence program

3.1 Crating New Project for FB

[Setting Purpose]

Set the PLC series, PLC type and project name necessary to create a new project for the FB.

[Operation Procedure]

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

[Dialog Box]
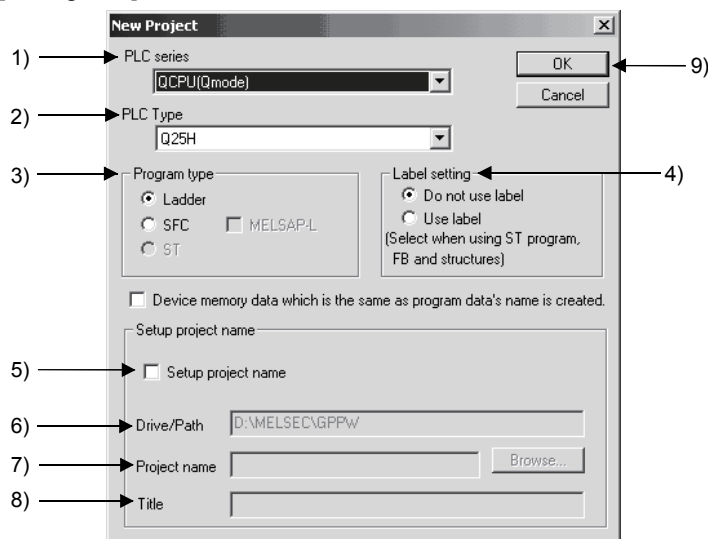
[Description]

1) PLC Series

As the PLC series, choose the QCPU (Q mode) or QnACPU.

2) PLC Type

As the PLC type, choose the CPU type to be used.

3) Program Type

When creating the FB, select Ladder program or ST program.

4) Label setting

When setting the FB, select "Use label".

5) Setup project name

Designate a project name to save the created data.

When designating a project name before creating a program, check the checkbox.

The project name can be designated before and after program creation.

When designating the project name after data creation, use the [Save As] menu.

6) Drive/Path

Set the path where the project will be saved (is saved).

Specifying only the project name without specifying the drive/path (blank) automatically creates the default drive/path and saves the project there.

7) Project name
   Designates a project name.

   The following shows the characters and the number of characters that can be used to designate a drive path, project name, or data name.

   • Number of characters
     The total number of characters used for designating both the project path and the project name (8 or more characters may be set) is 150.
     <Example>
     C:\SW3D5GPPW\ABCDEFGHIJKLMNOPQRSTUVWXYZ

   • Characters not available in A, QnA and FX series
     /, \, >, <, ＊, ?, " '', |, :, ; (: and \ can be set for drive designation only)
     Do not use a period (.) or space ( ) at the end of the project name. (If there is a space right after the project name, the space is deleted automatically.)

8) Title
   Sets the title for the project in up to 32 characters.

9) OK button
   Click this button to create a new project.

## 3.2 Adding New FB Definition to the Created Project

[Setting Purpose]

Add a new FB definition for the new FB to the created project.
The FB definition consists of an FB Body and Header used to set the labels
(variables) used in the program.

[Operation Procedure]

• Choose [Project] → [Edit Data] → [New].
• Click the right button of the mouse on the project data list and choose [New].

[Dialog Box]
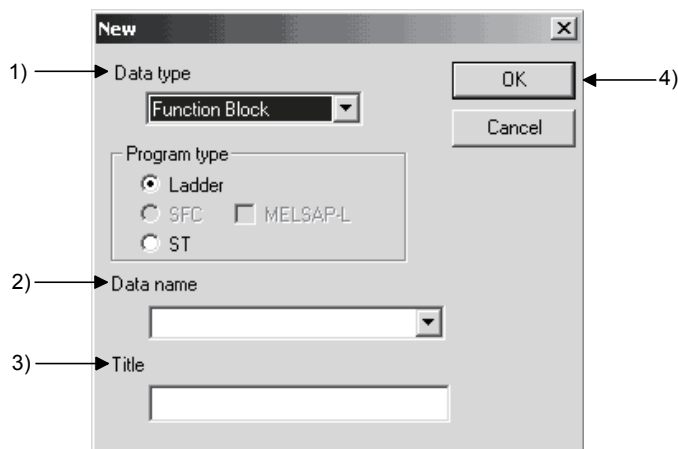
[Description]

1) Data type

Choose Function block.

2) Program type

When creating the FB, select Ladder program or ST program.

3) Data name

Set the data name (FB definition name) to be added newly.
Set the data name within eight characters.

• Usable characters

Alphanumeric characters, _, ^, $, , ~ (tilde), !, #, %, &, ( ), -, { }, @,
* (apostrophe), ' (single quotation)

• Unusable characters

Any of " = | : ; , \ [ ] + * ? <> . / will result in an error.

If the other FB definition has already been created in the project, displaying the
New dialog box sets that FB definition name in the Data name field.

4) Title

Sets the title for the project in up to 32 characters.

5) OK button

Click this button to create the new FB definition.

POINT

(1) There are no restrictions on the number of FB definitions that can be created in a project. (It depends on the capacity of the project saving destination.)

(2) If multiple FB definitions are created in one project, we recommend you to create a title for ease of searching for the object FB definition.
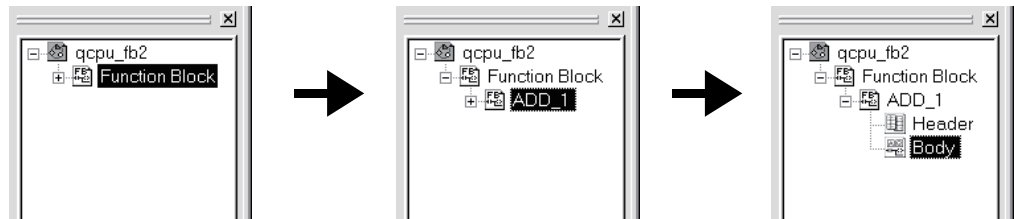
3.3 Creating FB Definition Program
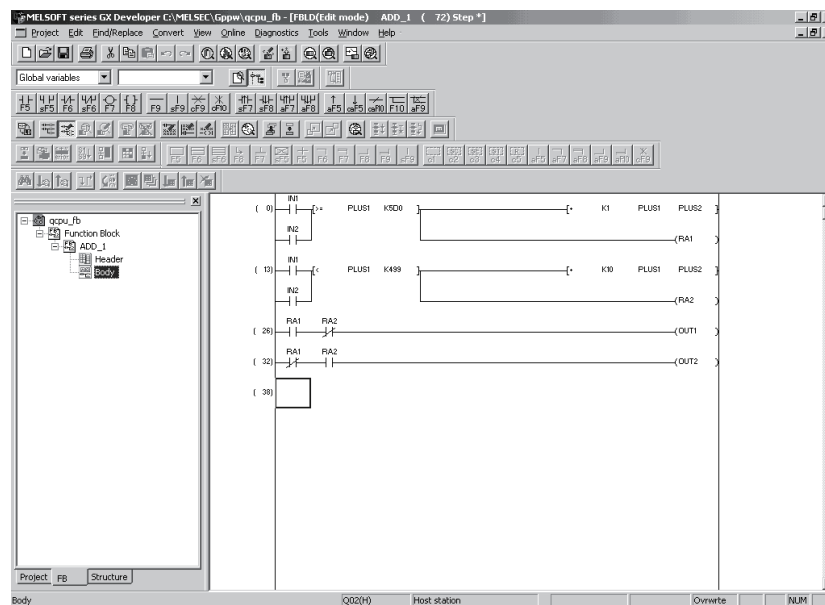
[Setting Purpose]

Create an FB definition program using labels (variables).
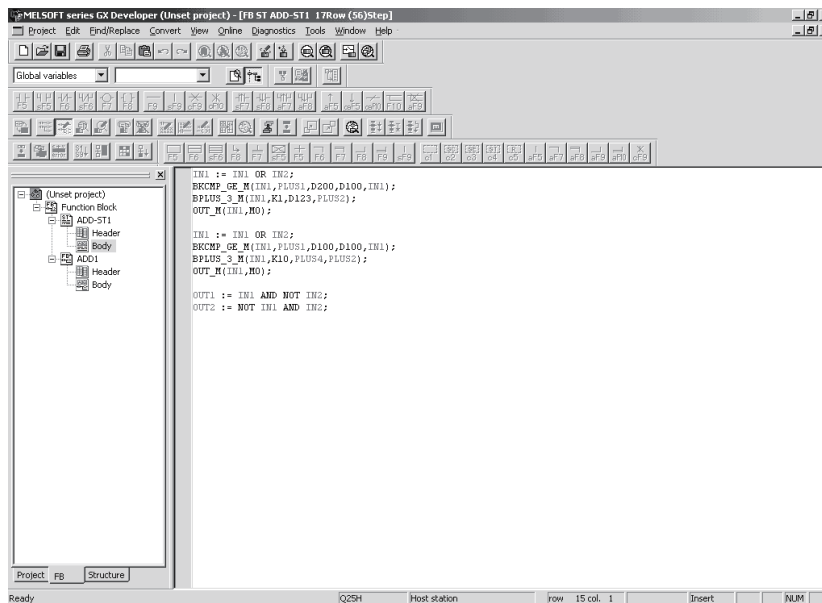
[Operation Procedure]

Double-click Function block on the <<FB>> tab and double-click FB definition or double-click Body individually.

[Window] Ladder program

[Window] ST Program

[Program Creation]

(1) To create an FB definition program, perform the same operation as in creating a
normal sequence program on the FB definition window.
The program can be created in the following specifications.

Item	Description
Program capacity	Max. 2048 steps (For ST Program: FB Definition Convert, maximum 2048 steps)
Language	Ladder, list, ST
Supported mode	Write, read
Number of labels	Max. 500 pcs.

(2) Use labels (variables) to create the program.

The used labels define the following four different functions. Use FB variables to define
the labels. (Refer to Section 3.4.)

Type	Description
Input variable	Variable input from the FB outside. Up to 24 pcs. including I/O variables, at least 1 pc.
Output variable	Variable output to the FB outside. Up to 24 pcs. including I/O variables, at least 1 pc.
I/O variable	Variable having the functions of both the input and output. Up to 24 pcs.
Internal variable	Variable used only in the FB inside. {500 - (input variables + output variables + I/O variables)} pcs.

Create a label within 16 characters.

Characters that cannot be used as a label are those of the reserved words and actual device names.

If the characters of the reserved word or actual device name are used, re-set the other label name as the "The reserved word is used." or "Device name is used." error message appears. Refer to Appendix 1 for unusable character strings.

REMARK

Though the FB definition allows a program to be created using actual devices (e.g. X100, Y110), it will be difficult to use multiple pieces of the same FB definition in a sequence program. (We cannot recommend this as it may result in double-coil.)

The following program example is used to explain how to use labels.

(a) FB definition program using labels

\<Ladder program\>
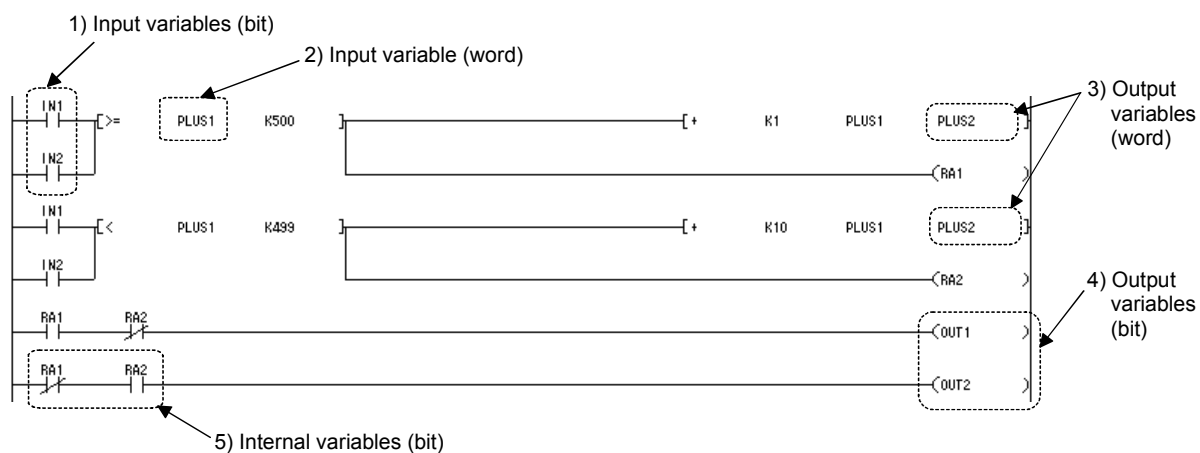
1) IN1 and IN2 are variables that turn ON/OFF using external ON/OFF information as conditions.

2) PLUS1 is a variable that inputs external numerical information to the FB inside.

3) PLUS2 is a variable that outputs the result calculated in the FB inside (numerical information) to the outside.

4) OUT1 and OUT2 are variables that control external outputs with the results calculated in the FB inside (ON/OFF information).

5) RA1 and RA2 are variables used in the FB inside only.

<ST Program>

2) Input variable (bit)   4) Output variable (word)

1) I/O variable (bit)   3) Input variable (word)

```
IN1 := IN1 OR IN2;
BKCMP_GE_M(IN1,PLUS1,D200,D100,IN1);
BPLUS_3_M(IN1,K1,D123,PLUS2);
OUT_M(IN1,M0);

IN1 := IN1 OR IN2;
BKCMP_GE_M(IN1,PLUS1,D100,D100,IN1);
BPLUS_3_M(IN1,K10,PLUS4,PLUS2);
OUT_M(IN1,M0);

OUT1 := RA1 AND NOT IN2;
OUT2 := NOT RA1 AND RA2;
```

5) Output variables (bit)

6) Internal variables (bit)
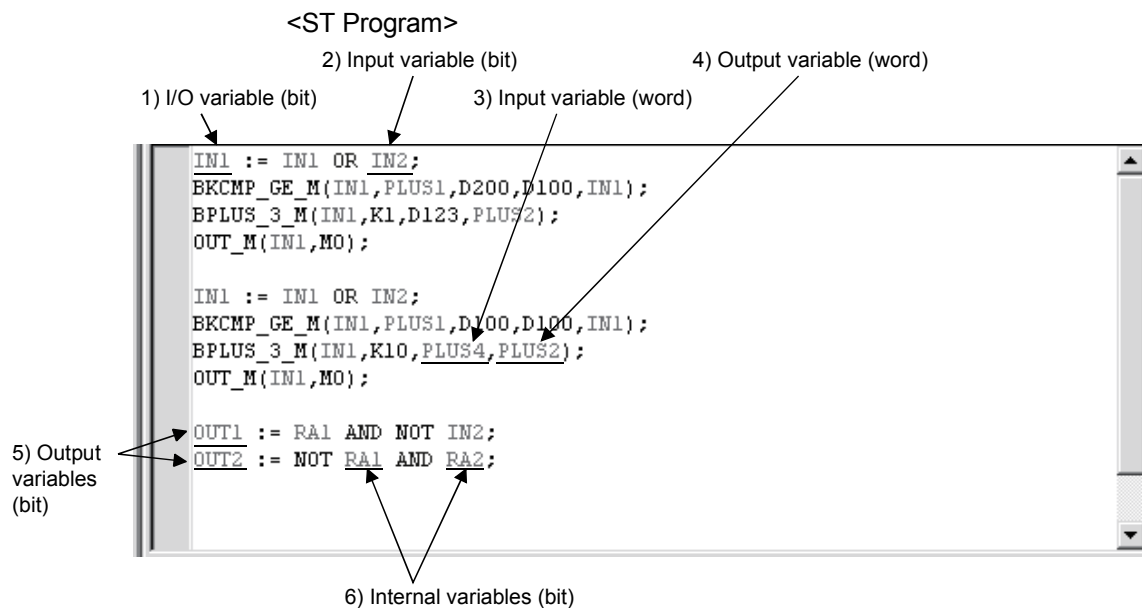
1) IN1 is a variable that turns ON/OFF using external ON/OFF information as a condition and outputs the result calculated in the FB inside (ON/OFF information) to the outside.
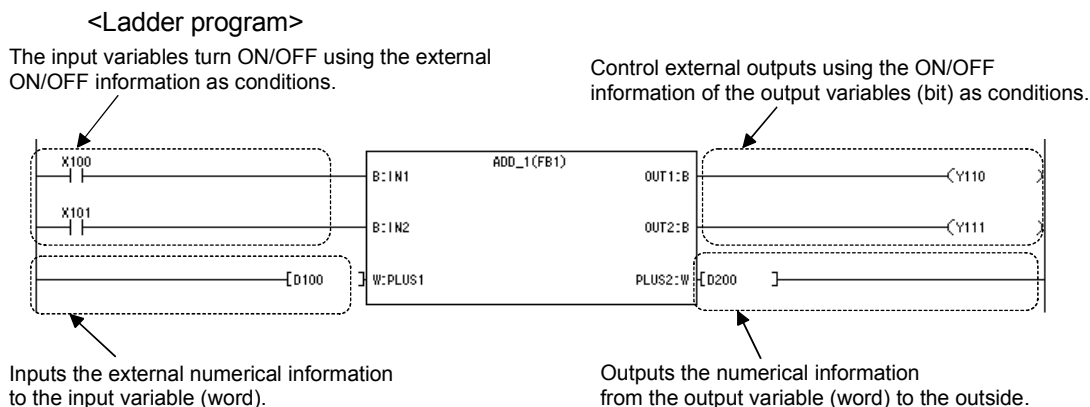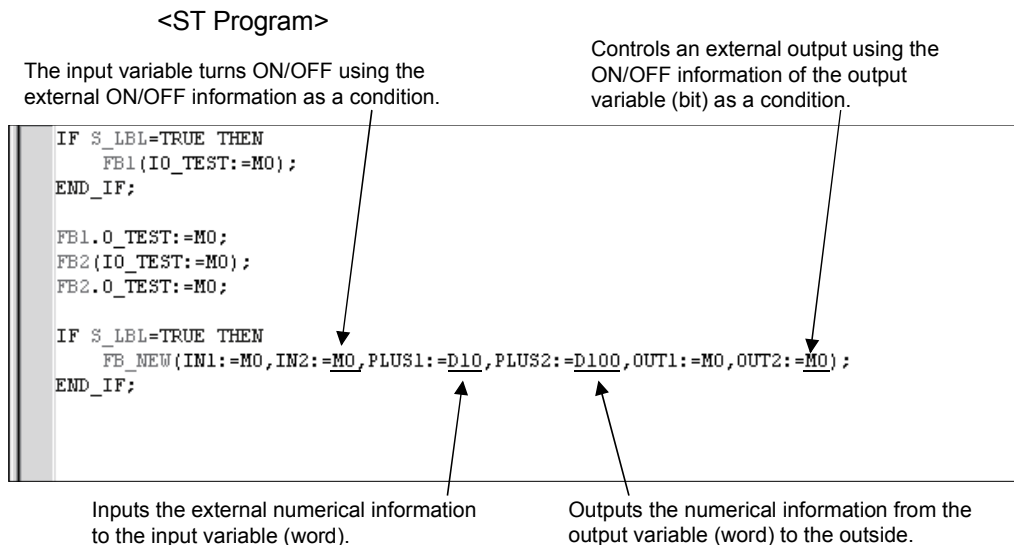2) IN2 is a variable that turns ON/OFF using external ON/OFF information as a condition.
3) PLUS1 is a variable that inputs external numerical information to the FB inside.
4) PLUS2 is a variable that outputs the result calculated in the FB inside (numerical information) to the outside.
5) OUT1 and OUT2 are variables that control external outputs with the results calculated in the FB inside (ON/OFF information).
6) RA1 and RA2 are variables used in the FB inside only.

(b) Operation example of the sequence program using the created FB definition

<Ladder program>

The input variables turn ON/OFF using the external ON/OFF information as conditions.

Control external outputs using the ON/OFF information of the output variables (bit) as conditions.

Inputs the external numerical information to the input variable (word).

Outputs the numerical information from the output variable (word) to the outside.

<ST Program>

The input variable turns ON/OFF using the
external ON/OFF information as a condition.

Controls an external output using the
ON/OFF information of the output
variable (bit) as a condition.

```
IF S_LBL=TRUE THEN
    FB1(IO_TEST:=MO);
END_IF;

FB1.O_TEST:=MO;
FB2(IO_TEST:=MO);
FB2.O_TEST:=MO;

IF S_LBL=TRUE THEN
    FB_NEW(IN1:=MO,IN2:=MO,PLUS1:=D10,PLUS2:=D100,OUT1:=MO,OUT2:=MO);
END_IF;
```

Inputs the external numerical information
to the input variable (word).

Outputs the numerical information from the
output variable (word) to the outside.

(3) The other FB cannot be used in the FB definition program.

(4) For the FB definition, do not create the program that jumps to outside the FB
definition. (Example: CALL instruction that jumps to a subroutine program)

(5) Create a program on the FB definition window, starting with step 0.
The FB definition program does not have an END instruction.
Entry of an END instruction will display an error.

(6) After program creation is complete, convert the program.
Make program conversion in any of the following three operations.

(a) F4 key

(b) [Convert] → [Convert/Compile]

(c) Right-click on the FB definition window and then choose [Convert/Compile]
(for ladder only) .

(7) On the FB definition window, the window title is provided with "*" to indicate that the
program is not compiled.

MELSOFT series GX Developer C:\MELSEC\GPPW\qcpu_fb2 - [FBLD(Edit mode)   ADD_1   (34) Step *]

∗   : Not yet compiled
No marking: Already compiled

(8) Note the following when creating the FB definition program with labels.
  (a) Index qualification cannot be used with labels.
     An index-qualified label is recognized as a new label.
     (Example)
     When "PLUS1Z1" is input to qualify the label "PLUS1" with the index register
     "Z1", the FB definition program is recognized as the label "PLUS1Z1".
     (Action)
     Use an actual device as the device to be index-qualified.

  (b) Since the numerical device that may be handled by an input variable is up to a
     double word, the instruction exceeding three words in input count cannot be
     input from the outside.
     (Example)
     For the data write instruction "TO" to the intelligent function module, up to two
     words can be input by an input variable.
     (Action)
     To input three or more words from the outside, use an actual device as the
     device for storing an input value or create the input instructions by the number of
     necessary times.

  (c) Since the numerical device that may be handled by an output variable is up to a
     double word, the instruction whose result will be three or more words cannot be
     output to the outside.
     (Example)
     For the BIN 32-bit multiplication "D*", the device for storing the multiplication
     result needs four words.
     (Action)
     To output three or more words to the outside, use an actual device as the device
     for storing the result.

REMARK

This manual does not detail the operations for creating programs.
For details of the operations for creating programs, refer to the GX Developer
Version 8 Operating Manual given in the related manuals.
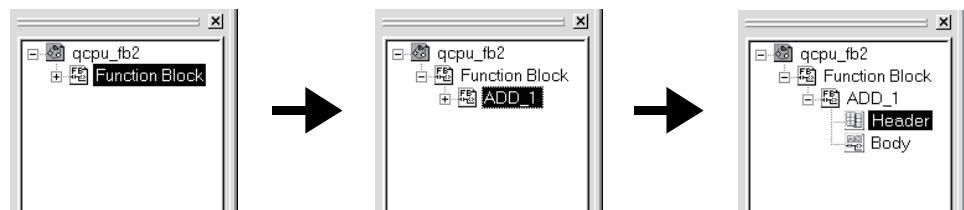
3.4 Creating FB Variables

[Setting Purpose]

Set the variable types, constant values, device types and comments to the labels (variables) used in the FB definition program.

The FB variables must be paired with the FB definition program body.

[Operation Procedure]

Double-click Function block on the <<FB>> tab and double-click "Header" or double-click Header individually.

[Window]

[Description]

1) Input/Output

Choose the label type. There are the following four different types.

• VAR_INPUT: Variable input from the FB outside

• VAR_OUTPUT: Variable output to the FB outside

• VAR_IN_OUT: Variable having the functions of both the input and output

• "Blank": Variable used in the FB inside

2) Label

Set the label (variable) used in the FB definition program.

3) Constant

Set a constant value when you want to set a constant to the label where "Blank" has been selected for the variable type.

Enter a constant value with K, H, E or " " for ease of identifying that it is a decimal, hexadecimal, real number or character string constant.

4) Device type
   • For the input, output and I/O variables, choose any of Bit, Word, Double
     word, Real number and Character string.
   The contents of the devices are as follows.
        1) BOOL   : Data represented by ON/OFF.
        2) INT   : Data represented by 16 bits.
        3) DINT   : Data represented by 32 bits.
        4) REAL   : Floating-point data represented by 32 bits.
        5) STRING : Character string data represented by JIS8 code.
   • For the internal variable, choose any of ARRAY, TIMER, COUNTER,
     STORED TIMER and POINTER.
        6) Array: Specify the number of data to be handled when multiple data,
              such as the BMOV or FMOV instruction, are handled.
              Selecting STRING or ARRAY displays the corresponding one of the
              following dialog boxes.
           Number of letters
              : The input range is 1 to 50
                characters.
           Number of elements
              : The input range is 1 to 255.
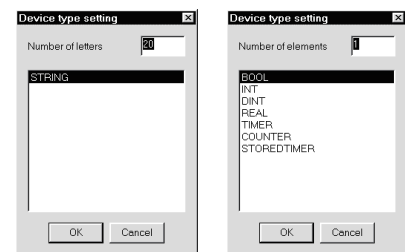
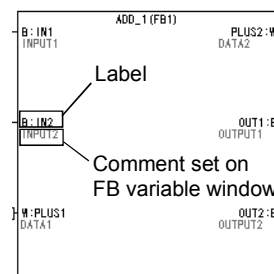5) Comment
           Set a comment within 64
           characters.
           The comment set in this field
           appears under the label within the
           FB when the sequence program
           is displayed with comments.
           For the ST Program, choose
           [View] → [Label information] and
           then put the cursor on ST
           Program to display comments in
           a tool tip format.

When FB pasted to
sequence program is
displayed with comments

6) Edit operation
   Insert button

   Used to insert a blank row in the current cell position.
   When you want to batch-insert multiple rows, choose the cells of the rows you
   want to insert. Blank rows are inserted in the cell-selected range.
   Add button

   Used to add a blank row under the row of the current cell position.
   When you want to batch-add multiple rows, choose the cells of the rows you
   want to add. Rows are added, starting with the row under the first one of the
   range-selected cells.
   Delete button

   Used to delete the cell-selected range.

7) [Resister] button

Clicking the [Resister] button registers the FB variables.

If any of the labels, constant values
and device types is set incorrectly,
the dialog box on the right appears
to indicate that the variables cannot
be registered.
Choosing Contents of error and
clicking the Jump button displays the
error location.
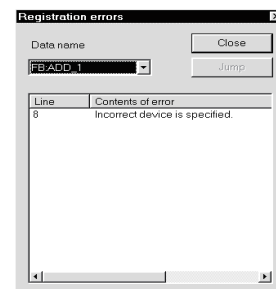
| POINT |

On the FB variable setting window, the set data are sorted in label order, constant
order and device type order. The sorting order is as follows.
1. Symbols
2. Alphabets (uppercase characters)
3. Alphabets (lowercase characters)

The FB definition automatically assigns
devices to the labels used in the program.
When creating a sequence program, avoid
automatically assigned devices.
Like local devices, the automatically assigned
devices are set in accordance with the
automatically assigned device setting, which
defaults to the following device ranges.
  1) Word device : D6144 to D12287
  2) Bit device   : M4096 to M8191
  3) Timer   : T64 to T2047
  4) Counter   : C512 to C1023
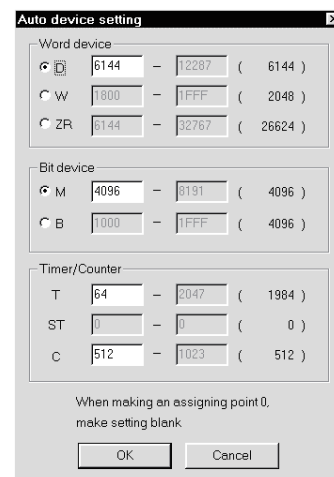
When you want to change the device range, choose [Edit] → [Auto device
setting] to display the Auto device setting dialog box, and change the device
range.

| REMARK |

For details of the operations performed on the Auto device setting dialog box, refer
to "Label Program" in the GX Developer Version 8 Operating Manual.

3.5 Converting the FB Definition (Convert (Compile))

[Setting Purpose]

   Convert the created FB definition so that it may be used in a sequence program.

[Operation Procedure]
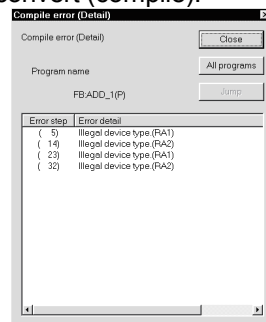
   Choose [Convert] → [Convert /Compile].

[Operation Items]

   (1) The following dialog box appears on normal completion of convert (compile).

   Click YES to write the label comments over the device comments.

   (2) The following dialog box appears on detection of errors during execution of
       convert (compile).

   Choosing Contents of error and clicking the Jump
   button displays the error locations.

   (3) When errors are detected, correct the error locations and then execute convert
       (compile) again.
       In the sequence program, you can use only the FB definition that has completed
       convert (compile).

(4) When you have created multiple FB definitions, you can compile all FB
definitions by performing compile operation once.
The dialog box shown in (2) appears on detection of errors during execution of
compile.

Clicking All programs shows the dialog box which allows you to confirm the

number of errors in all programs.
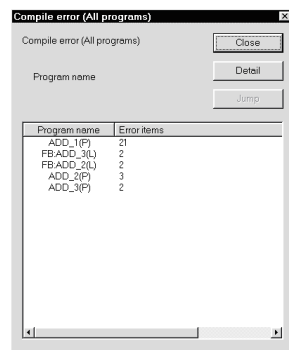
(Display example)

[ADD_1(P)] .... Number of errors in FB definition ADD_1

[FB:ADD_3(L)]... Number of errors in FB definition ADD_3

[FB:ADD_2(L)]... Number of errors in FB definition ADD_2

[ADD_2(P)] .... Number of errors in FB definition ADD_2

[ADD_3(P)] .... Number of errors in FB definition ADD_3

Choosing the program name and clicking Detail displays

the detailed dialog box of the selected item.

3.6 Pasting the Converted FB to Sequence Program (FB Utilization)

3.6.1 Pasting the FB definition to sequence program

[Setting Purpose]

Paste the FB definition to the sequence program being created so that the FB may be used.

[Setting Procedure]

Any of the following three different operations can be performed to utilize the FB.

(1) Method using drag and drop from the FB tab with the mouse

(2) Method using the function block utilization dialog box
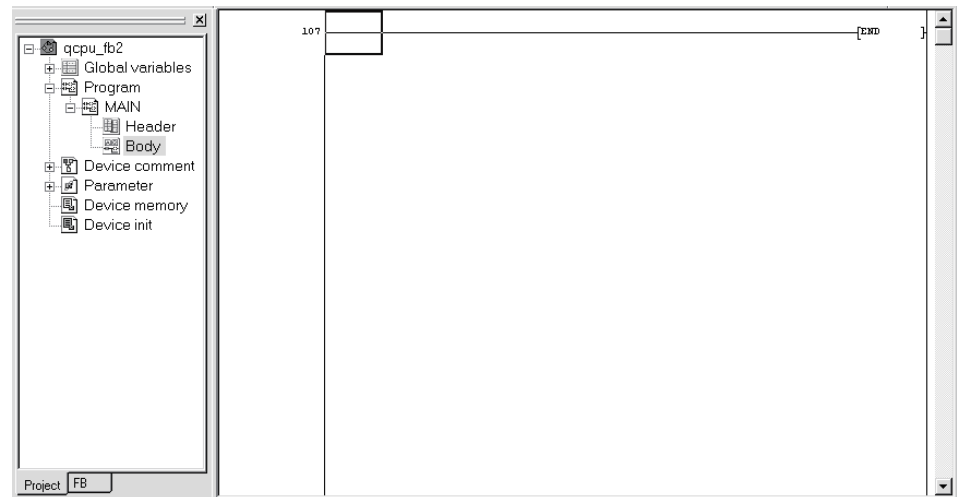
(3) Method using the instruction dialog box

The following explains the procedures of the above operations.

(1) Method using drag and drop from the FB tab with the mouse

(a) Display the sequence program where the FB definition will be pasted.

Select the target sequence program name from the Project tab to display the sequence program.
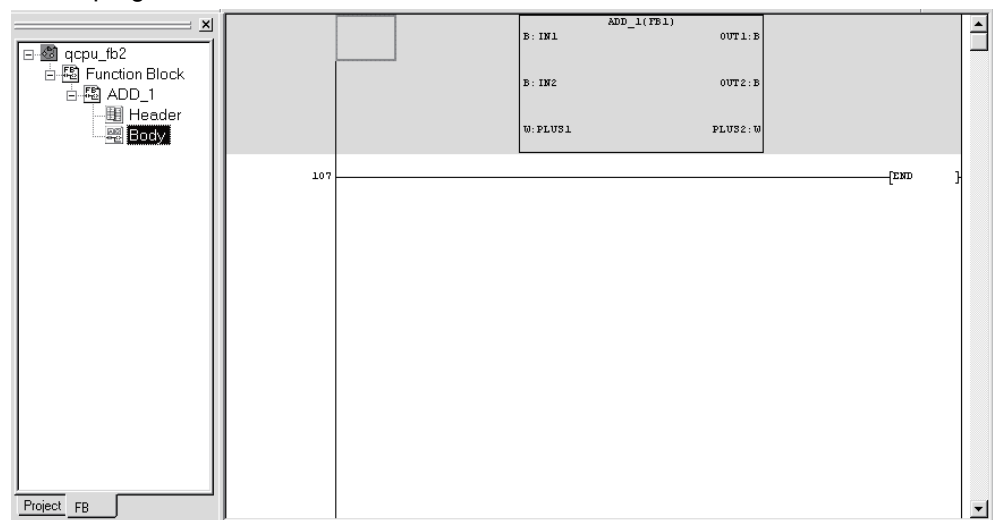
Set the mode to Ladder (Write).

(b) Switch from the Project tab to the FB tab, and drag and drop the FB definition
from the FB tab onto the sequence program.
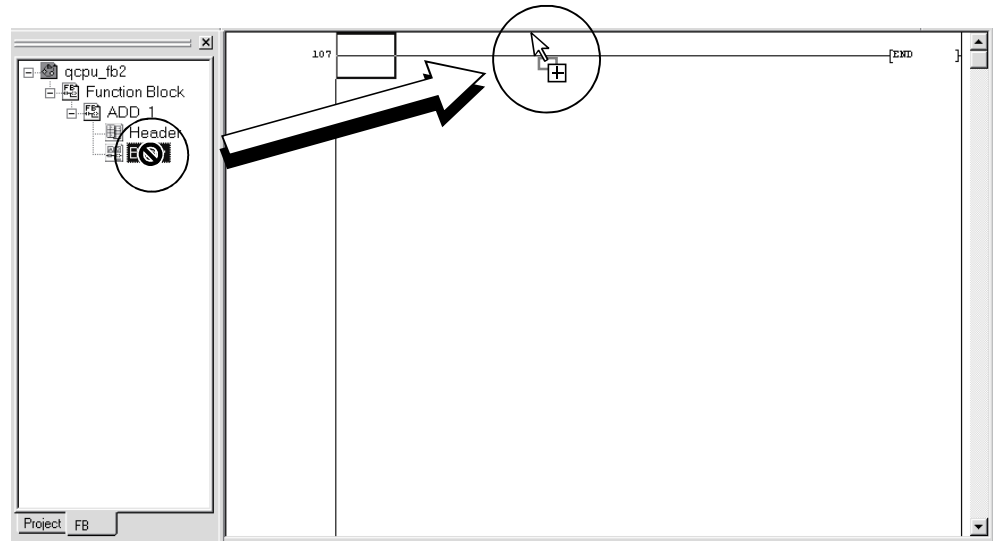
(c) The FB appears in the cursor-specified place (drop position) on the sequence
program.


```
POINT
```
(1) The position where the FB definition will be pasted is determined by the place
specified by the cursor.
  (a) When the paste position is a ladder block other than the FB
     (In the insert/overwrite mode) It is inserted in the row above the ladder block of
                                     the paste position.
  (b) When the paste position is a ladder block including the FB
     (In the insert mode) It is inserted in the row above the ladder block of the paste
                           position.
     (In the overwrite mode) It is written over the FB of the paste position.

(2) There are no restrictions on the number of FBs pasted to the sequence program.
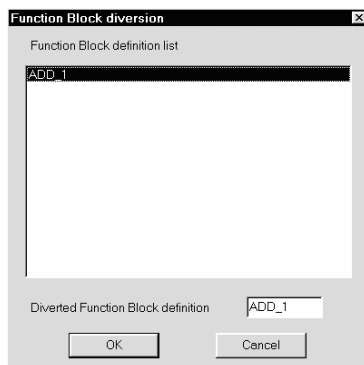
(2) Method using the function block utilization dialog box
    (a) Display the sequence program where the FB definition will be pasted.
       Select the target sequence program name from the Project tab to display the
       sequence program.
       Set the mode to Ladder (Write). (Perform the same operation as in (1)-(a).)

    (b) Choose [Project] → [Function Block] → [Diversion] to display the Function
       Block diversion dialog box.

Double-click the FB definition name to be
utilized from the Function Block diversion
dialog box.
Alternatively, click and choose the FB
definition name to be utilized, and click
the | OK | button.

    (c) The FB appears in the cursor-specified place (drop position) on the sequence
       program. (This is the same as in (1)-(c).)

(3) Instruction list input dialog box
    (a) Display the sequence program where the FB definition will be pasted.
       Select the target sequence program name from the Project tab to display the
       sequence program.
       Set the mode to Ladder (Write). (Perform the same operation as in (1)-(a).)

    (b) From the keyboard, enter "FB. 'FB definition name'" in the sequence program.
       This displays the Enter symbol window and writes it in the device instruction
       input field. Example: FB.ADD_1

    (c) Clicking the | OK | button displays the FB in the cursor-specified place on the
       sequence program.
       (This is the same as in (1)-(c).)

| REFERENCE |

For FB utilization in an ST Program, refer to the "QCPU (Q mode) Programming
manual (Structured Text)".

3.6.2 Changing the name (FB name) of the pasted FB (FB name changing)

This operation is optional. Perform this operation as necessary.

[Setting Purpose]
  Change the FB name of the pasted FB.

```
┌─────────────────────────────────────────────────────────────┐
│  POINT                                                        │
│  The name "(FB1)" of the FB pasted to the sequence program is │
│  displayed within                                             │
│  parentheses "()" right after the FB definition name.         │
│                                                               │
│  FB definition name ──┌──────ADD_1 (FB1)──────┐── FB name     │
│                       │ B : IN1        OUT1 :B │              │
│                       │                        │              │
│                       │ B : IN2        OUT2 :B │              │
│                       │                        │              │
│                       │ W :PLUS1      PLUS2 :W │              │
│                       └────────────────────────┘              │
│                                                               │
│  If multiple pieces of the same FB or the other FBs are pasted│
│  to a sequence                                                │
│  program, the FBs are automatically assigned the specific FB  │
│  names that increase                                          │
│  in number consecutively, e.g. "FB1", "FB2", "FB3".           │
└─────────────────────────────────────────────────────────────┘
```

Perform this operation when you want to make multiple FBs pasted to a sequence program more easily identifiable.
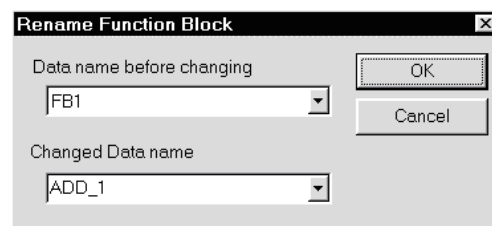
[Operation Procedure]
  Choose [Project] → [Function Block] → [Rename FB].

[Setting Procedure]
  (1) Type the old data name and new data name in the Rename Function Block dialog box.
    You can select the old data name from the list box.
    If you reselect the FB whose name will be changed, the FB name has already been entered in the Data name before changing field.

```
┌─────────────────────────────────────────────────────┐
│ Rename Function Block                            [X] │
│                                                      │
│  Data name before changing      ┌──────────────┐     │
│  ┌──────────────────────────┐   │     OK       │     │
│  │FB1                     ▼ │   └──────────────┘     │
│  └──────────────────────────┘   ┌──────────────┐     │
│  Changed Data name              │   Cancel     │     │
│  ┌──────────────────────────┐   └──────────────┘     │
│  │ADD_1                   ▼ │                        │
│  └──────────────────────────┘                        │
└─────────────────────────────────────────────────────┘
```

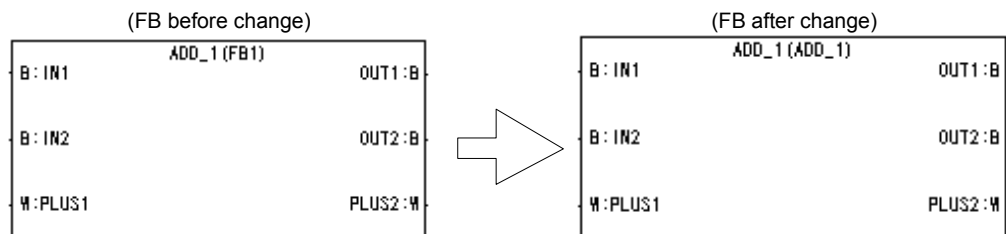   The FB name that may be entered is within 16 characters.
    The characters that cannot be used as the FB name are those of the reserved words and actual device names.
    If you used the characters of the reserved word or actual device name, the following error message will appear. Therefore, set the other label name. Refer to Appendix 1 for the unusable character strings.

```
┌─────────────────────────────────────────────────────┐
│ MELSOFT series GX Developer                      [X] │
│                                                      │
│  (i)   Illegal characters have been used for a data  │
│        name.                                         │
│        Use legal characters.                         │
│                                                      │
│              ┌──────────────┐                        │
│              │     OK       │                        │
│              └──────────────┘                        │
└─────────────────────────────────────────────────────┘
```

(2) Click the OK button to display the new FB name in the FB.

When you have changed the FB name, the display changes as shown below.

(FB before change)

```
         ADD_1 (FB1)
B : IN1                OUT1 :B

B : IN2                OUT2 :B

W :PLUS1              PLUS2 :W
```

(FB after change)

```
         ADD_1 (ADD_1)
B : IN1                OUT1 :B

B : IN2                OUT2 :B

W :PLUS1              PLUS2 :W
```

POINT
FB name changing can be executed in the write mode. It cannot be executed in the read mode.

3.6.3 Changing the module start I/O addresses of the FB definition

[Setting Purpose]
Change the batch of module start I/O addresses (Un, Un/Gm), I/O devices (X, Y, DX, DY) that are used in the FB definition.

[Operation Procedure]
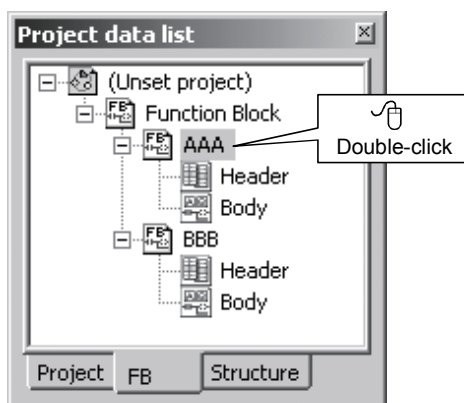
1. Select the FB tab within the Project data list.

2. Double-click and select the program body of the FB definition that uses the module start I/O address to be changed.
   (The FB definition (body) will appear.)

3. Select [Project] → [Function Block] → [FB change module address].

4. Specify a new module start I/O address for the lowest module start I/O address set in the FB definition.

The batch of module start I/O addresses will be changed as shown below.

Before FB definition (body) change
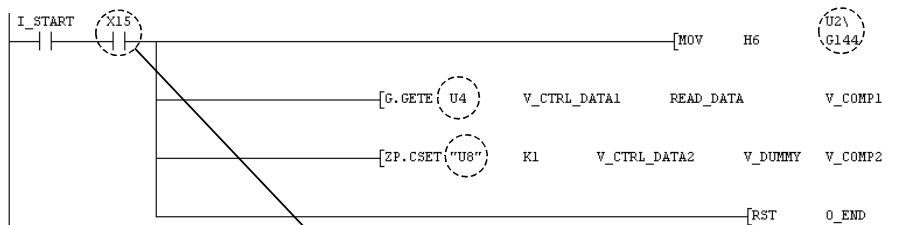
The lowest module start I/O address in the FB definition is displayed.
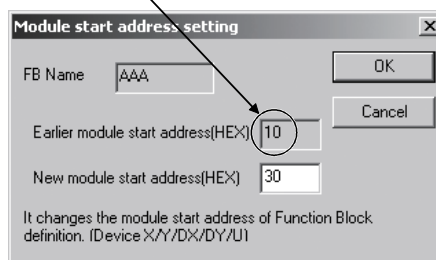
After FB definition (body) change
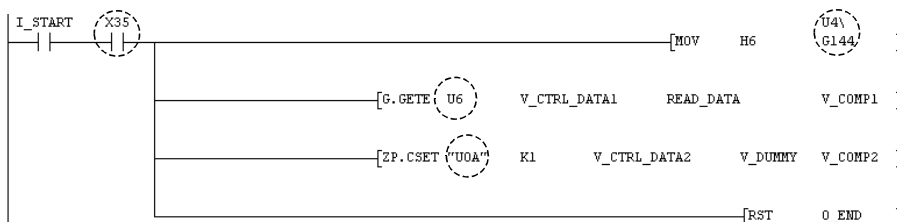
POINT

• The module start I/O addresses and I/O devices on the active FB definition
  screen will be changed.
  The active FB definition screen can be confirmed by checking the title bar or
  module start I/O address setting screen of GX Developer.

3.7 Creating the Sequence Program

3.7.1 Editing the sequence program

[Setting Purpose]

Create an input ladder section and an output ladder section for the FB pasted to the sequence program.
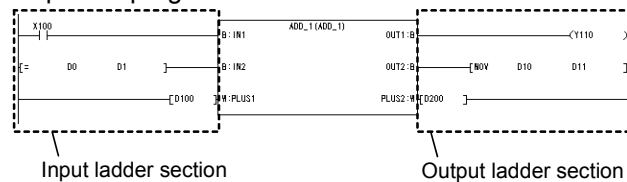
Add other ladders to the FB where the input and output ladder sections have been created to complete the sequence program.
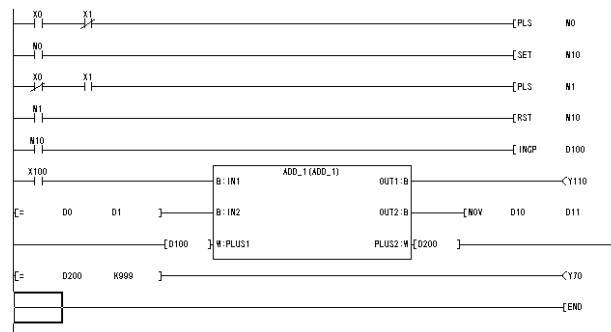
[Program Creation]

(1) After the FB has been pasted, the sequence program can be created by performing the same operation as in creating a normal ladder format sequence program.

REMARK

This manual does not detail the operations for creating programs.
For details of the operations for creating programs, refer to the GX Developer Version 8 Operating Manual given in the related manuals.

(2) In the pasted FB, the labels of the input variables are displayed on the left side and those of the output variables on the right side. The input side labels are shown as "'Device type' : 'Label name'" and the output side labels as "'Label name' : 'Device type'".

The device types are indicated by the following alphabets.

(a) BOOL   : B
(b) INT   : W
(c) DINT   : D
(d) REAL   : E
(e) STRING   : S
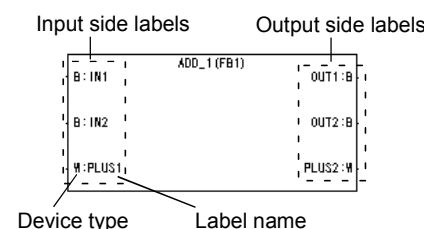
(3) Each label is connected with the input or output ladder section.

The ladder that can be connected to the input or output ladder section depends on the device type of the connected label.

(a) When the device type is BOOL

1) In the input ladder section, create instructions that will be conditions for input variables, e.g. contact, association and comparison operation instructions.

2) In the output ladder section, create instructions that will provide outputs, e.g. output and arithmetic operation instructions. The output variable will be a condition for the output instruction.

(b) When the device type is INT/DINT/REAL/STRING

1) Connect a word device (including a digit-specified bit device or constant) to the input ladder section. The information of the word device is stored into the input variable.

2) Connect a word device (including a digit-specified bit device) to the output ladder section. The information of the output variable is stored into the word device.

To create the ladder where the word device will be connected, click the [{}] button or press the [F8] key to display the Enter symbol window, and enter the device in the device instruction input field. (When the device type is a DINT/REAL/STRING, consecutive devices starting with the entered device are occupied.)

The following is the example where the input and output ladder sections have been connected to the FB.

Connect a contact for the condition that turns ON/OFF the input variable (BOOL).
In this example, IN1 turns ON when X100 turns ON.
When D0 = D1, IN2 turns ON.

The output variables (BOOL) will be conditions that control the external outputs.
In this example, Y110 turns ON when OUT1 turns ON.
When OUT2 turns ON, [MOV D10 D11] is executed.
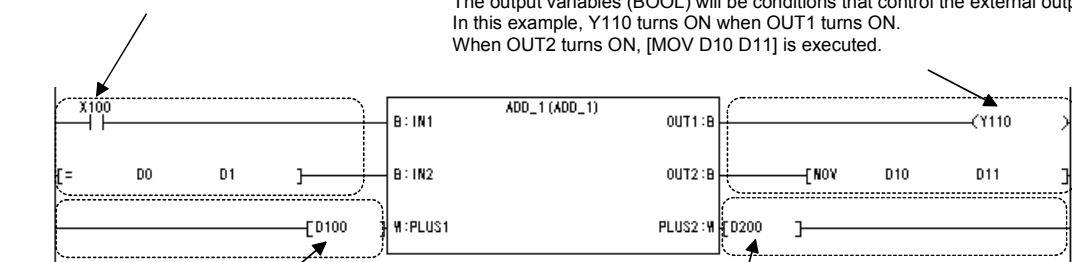
Connect a data register for inputting numerical information from the outside to the input variable (INT).
In this example, the numerical information of D100 is stored into PLUS1.

Connect a data register for outputting numerical information from the output variable (INT) to the outside.
In this example, PLUS2 stores the numerical information into D200.

The display specifications of the input and output ladder sections are as follows.

(a) Display width of the input ladder section: Equal to the widths of four contacts

(b) Display width of the output ladder section: Equal to the widths of three contacts + one coil

Return connection cannot be made in the input and output ladder sections. Therefore, when a ladder cannot be created in the above display width, use an internal relay or like to continue the ladder in the other ladder block.

(4) The input variable/output variable of the FB does not result in an error if it is not connected from the input ladder section/output ladder section.
After conversion of the sequence program, the unconnected input variables/output variables are displayed in the form that they are connected to the FB by horizontal lines from the left bus in the input ladder section or from the right bus in the output ladder section.
In this case, the input variables are indefinite and the output variables are not output to the outside.
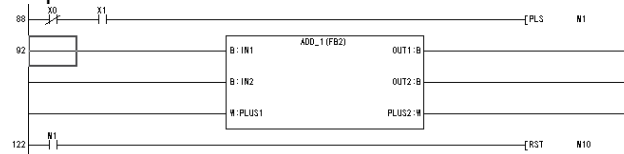
(5) Edit the sequence program that includes the FB as described below.
  (a) Cut/copy can be made in one ladder block including the FB.
      Cut/copy cannot be made only in any ladder section of the input ladder section, FB and output ladder section.
      Since the copied FB is judged as the FB different from the copy source FB, the FB name changes automatically from "FB1" to "FB2" to "FB3".
  (b) A row can be inserted in one ladder block including the FB.
  (c) A row can be deleted in one ladder block including the FB.
  (d) A column cannot be inserted where a ladder return occurs.
  (e) A column can be deleted in the input ladder section/output ladder section.
  (f) Macro registration/utilization cannot be performed in the ladder block that includes the FB.

3.7.2 Confirming the data of the pasted FB

[Setting Purpose]

In the pasted status, the data of the FB pasted to the sequence program are hidden to simplify the display of the sequence program.

Perform this operation when you want to confirm the data of the FB during creation of the sequence program.

[Operation Procedure]

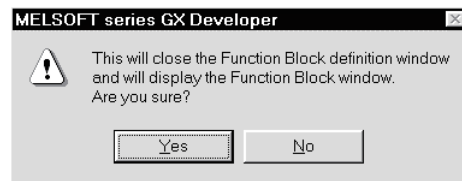(1) Double-click the corresponding FB.
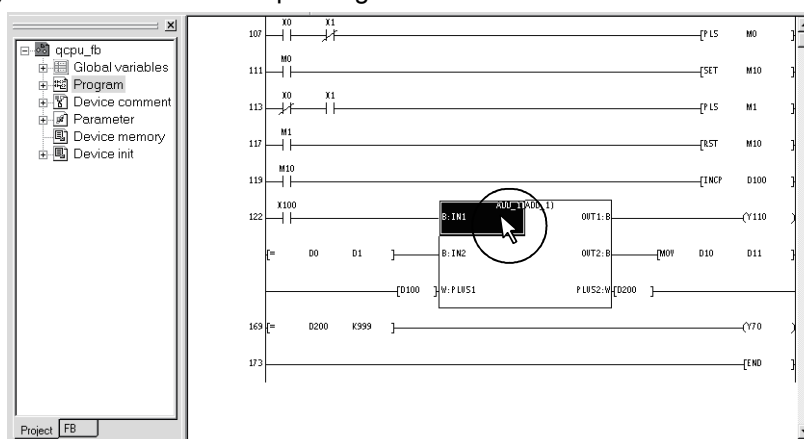
REMARK

For operation for an ST Program, refer to the "GX Developer Version 8 Operating Manual (Structured Text)".

(2) The FB window appears.
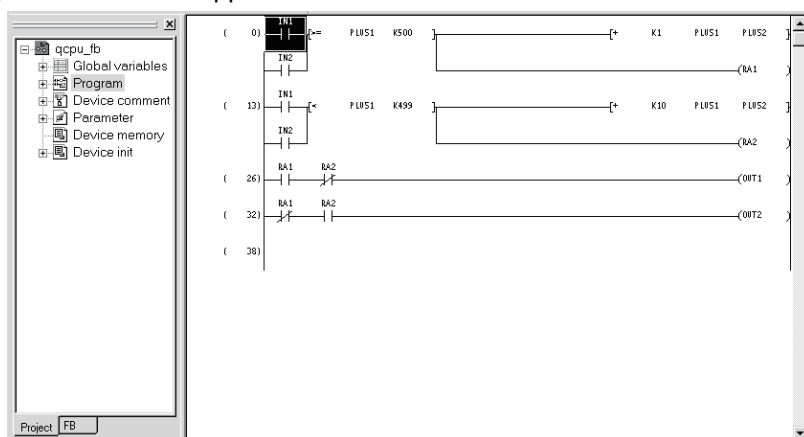
The following dialog box appears if the FB definition window or FB variable setting window has already been displayed.

Clicking the ⌐Yes¬ button closes the FB definition window or FB variable setting window and displays the FB window.

POINT

The FB window is used to read, monitor and test an FB program. Program correction cannot be made on this window.

When correcting the FB program, open the FB definition window.

The FB window closes when the FB definition window opens.

3.7.3 Searching/replacing the sequence program (Search/replace)

[Setting Purpose]
Perform this operation when searching for or replacing a device, instruction or like in the created sequence program.

[Operation Procedure]
Choose [Find/Replace] → [Find device], [Find instruction], [Find step no.],
[Find character string], [Find contact or coil],
[Find comment], [Replace device], [Replace instruction],
[Replace device], [Replace instruction],
[Change open/close contact], [Replace character string],
[Change module start address],
[Replace statement/note type],
[Cross reference list], [List of used device].

REMARK

This manual does not detail the search/replace operations.
For details of the search/replace operations, refer to the GX Developer Version 8 Operating Manual given in the related manuals.

[Item Explanation]
(1) Search/replace differs in execution range as described below depending on the target window.

Target Window	Execution Range
Sequence program window	The sequence program being displayed is the target of search/replace. However, the pasted FB is not included in the target. ＊1
FB definition window	Only the FB definition program being displayed is the target of search/replace.
FB window	Only the FB definition program being displayed is the target of search.

(2) Search/replace is usable on the windows as indicated below.

Search/Replace Item	Sequence Program Window	FB Definition Window	FB Window
Find device Find instruction Find step no. Find character string Find contact or coil Find comment	Can be executed.	Can be executed.	Can be executed.
Replace device Replace instruction Change open/close contact Replace character string Change module start address Replace statement/note type	Can be executed.	Can be executed.	Cannot be executed.
Cross reference list List of used device	Can be executed.	Can be executed.	Can be executed.

＊1: Only character string search can search the FB pasted to the sequence program for the FB definition name or FB name.

POINT

When searching for the FB pasted to the sequence program, you can find the object FB easily by specifying the FB definition name or FB name of the target FB in character string search.

(1) After typing the FB definition name or FB name, click the [Find Next] button.

Type FB definition name or FB name.
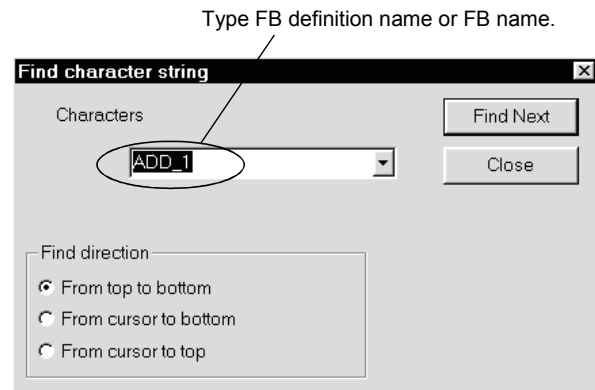
(2) The cursor moves to the top left of the FB section.

Cursor moves to top left of object FB.
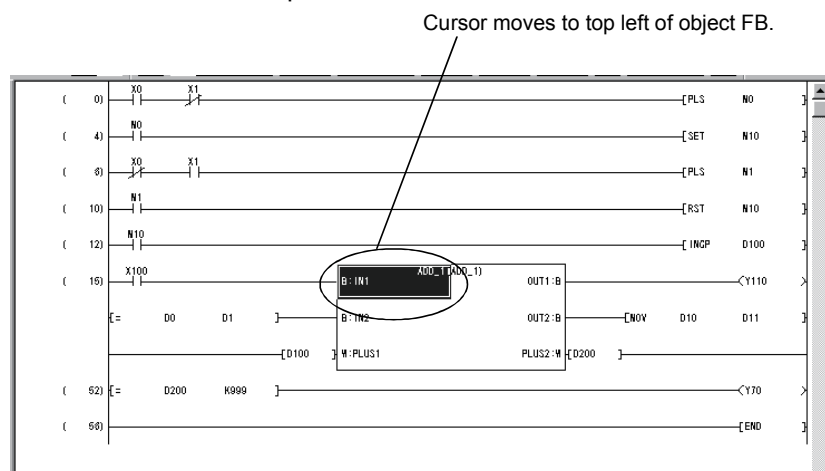
3.7.4 Creating/displaying the comments, statements or notes (Sentence creation)

[Setting Purpose]

Create comments, statements or notes in the sequence program to improve the readability of the program.

The created comments, statements or notes can be displayed by setting the display format.

[Operation Procedure]

(1) Creation procedure

Choose [Edit] → [Documentation] → [Comment], [Statement] or [Note]

(2) Display procedure

Choose [Edit] → [Comment], [Statement] or [Note].

REMARK

This manual does not detail the operations for creating comments, statements and notes.

For details of the creating operations, refer to the GX Developer Version 8 Operating Manual given in the related manuals.

[Description]

(1) Comment

Comments can be created for the devices in the input and output ladder sections of the ladder block that includes the FB.

(2) Statement

Statements can be created in only the first ladder of the input ladder section in the ladder block that includes the FB.

They cannot be created in the output ladder section.

(3) Note

Notes can be created in only the output ladder section of the ladder block that includes the FB.

They cannot be created in the input ladder section.

3.8 Converting the Sequence Program into Program to Be Executed on the Programmable Controller CPU (Convert (Compile))

[Setting Purpose]

The sequence program where the FB has been pasted cannot be executed as-is on the programmable controller CPU. Perform Convert (Compile) operation to convert it into an executable program.

[Operation Procedure]

Choose [Convert] → [Convert/Compile], [Convert/Compile (All programs being edited)] or [Convert/Compile (All programs)] or [Convert/Compile (Online change)].

[Description]

(1) The following dialog box appears on normal completion of compile.
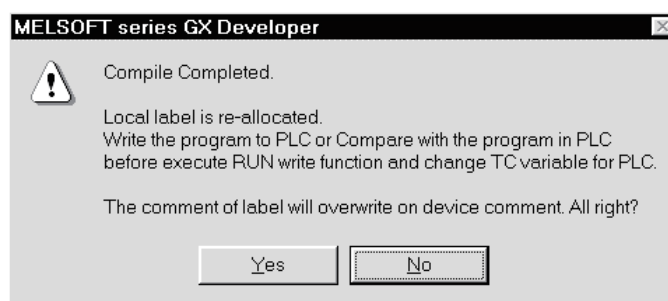
The following dialog box appears on detection of errors.
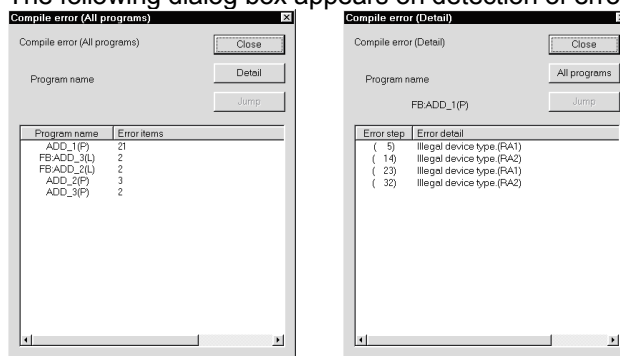
"Compile error (All programs)" appears if errors occurred over multiple programs.
"Compile error (Detail)" appears if errors occurred in one program only.

(2) If not converted (compiled), the sequence program is provided with "✻" in the window title.

✻   : Not yet compiled
No marking: Already compiled

MELSOFT series GX Developer C:\MELSEC\GPPW\qcpu_fb2 - [LD(Edit mode)   MAIN   (72) Step *]

(3) The program executed after Convert (Compile) is not displayed on GX Developer.

REMARK

In the converted (compiled) sequence program, devices are assigned to labels in accordance with the Auto device setting.
The Auto device setting defaults to the following device ranges.
1) Word device   : D6144 to D12287
2) Bit device   : M4096 to M8191
3) Timer   : T64 to T2047
4) Counter   : C512 to C1023
The devices assigned to the labels are used in order, starting with the last numbers (D12287, M8191, T2047, C1023).

The following is the example of a program executed after convert (compile)
(a) Sequence program
    This ladder example stores 100 into D400 when X0 turns on as the value of D400 exceeds 4000.

(b) Program executed after convert (compile)

Label	Assigned Device
in1	M8191
in2	D12287
out10	D12286
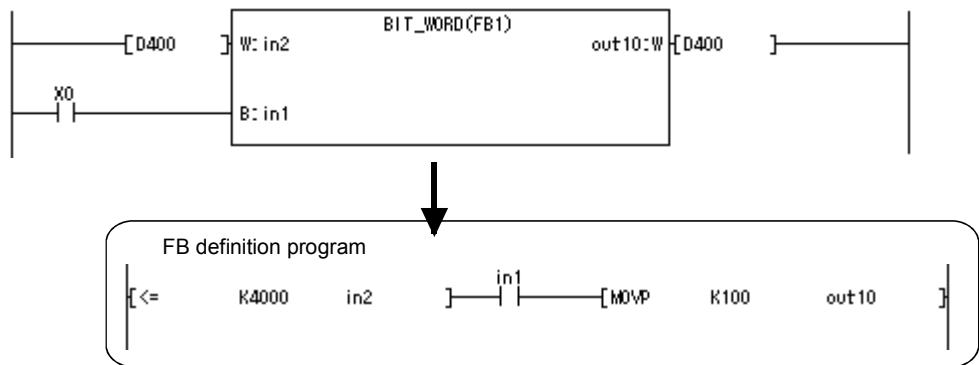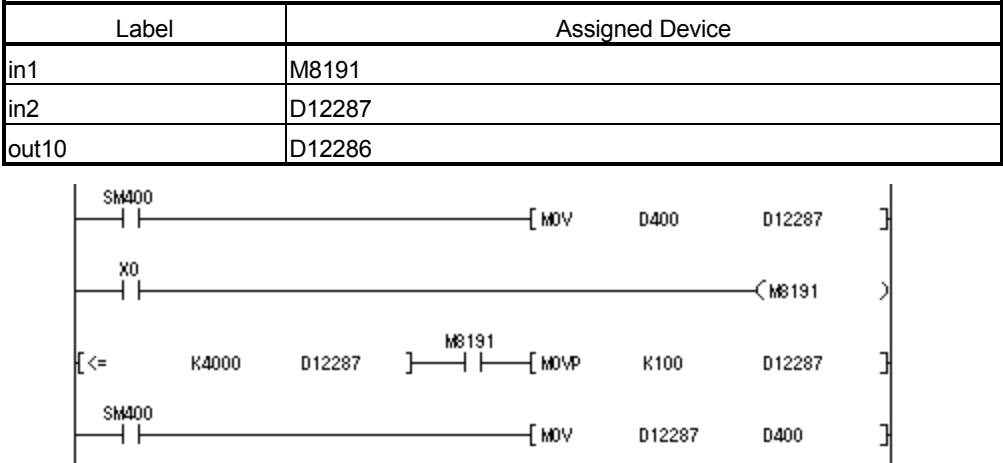
The contact X1 in the input ladder section and the devices D400 in the input and output ladder sections cannot be replaced directly by the labels in1, in2 and out10 in the FB definition program.
• The contact X1 is connected to the coil of the device M8191 assigned to in1.
• The device D400 in the input ladder section always transfers data to the device D12287 assigned to in2.
• The device D12286 assigned to out11 always transfers data to the device D400 in the output ladder section.

Since D12286/D12287 in this executed program are undefined at the startup of the programmable controller CPU, D400 changes to an indefinite value in the output ladder section if D400 in the input ladder section is less than 4000.
To avoid such a state, connect the same labels and set the variable type as I/O variables. Setting the variable type as the I/O variables connects the same devices to the input and output ladder sections.
The following is the program example where the above remedy has been taken.

(a) Sequence program
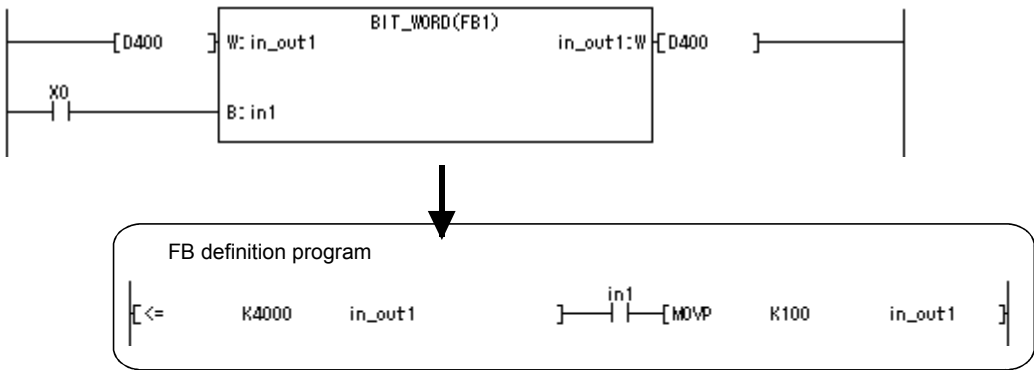
(b) Program executed after compile

Label	Assigned Device
in1	M8191
in _ out 10	D12287
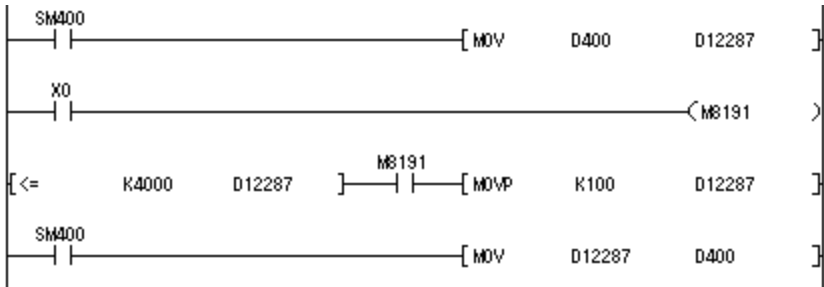
3.9 Reading the Sequence Program from the Programmable Controller CPU (Read from PLC)

[Setting Purpose]

Read the sequence program from the programmable controller CPU.

[Operation Procedure]

Choose [Online] → [Read from PLC] or click 🔓.

REMARK

This manual does not detail the operations for reading from programmable controller.

For details of the operations for reading from programmable controller, refer to the GX Developer Version 8 Operating Manual given in the related manuals.

[Setting Procedure]

(1) Set the access target in Transfer setup.

(2) Display the Read from PLC dialog box.

(3) Turn on and choose the check box of the data name which will be read from the programmable controller.

To select the parameters and all programs of the project, click the
Param+Prog button.

(4) Click the Execute button.

POINT

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.

3.10 Writing the Sequence Program to the Programmable Controller CPU (Write to PLC)

[Setting Purpose]

Write the converted sequence program to the programmable controller CPU.

[Operation Procedure]

Choose [Online] → [Write to PLC] or click 🖳 .

REMARK

This manual does not detail the operations for writing to programmable controller. For details of the operations for writing to programmable controller, refer to the GX Developer Version 8 Operating Manual given in the related manuals.

[Setting Procedure]

(1) Set the access target in Transfer setup.

(2) Display the Write to PLC dialog box.

(3) Turn on and choose the check box of the data name which will be written to the programmable controller.

To select the parameters and all programs of the project, click the Param+Prog button.

(4) Click the Execute button.

POINT

(1) For the Q or QnA, integrated statements are written to the programmable controller CPU together with the programs.

Peripheral statements are not written to the programmable controller CPU. In this case, the statement/note merge option has the function to merge statements/notes with those in the personal computer when read from PLC is performed.

When the program has been read from the programmable controller CPU without use of this function, be careful not to overwrite the old program accidentally to save a program having no statements/notes.

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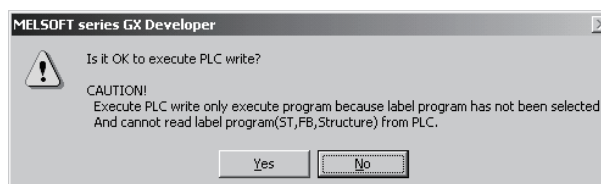
3.11 Monitoring and Testing the Sequence Program

3.11.1 Monitoring the sequence program

[Setting Purpose]

Monitor the sequence program that has been written to the programmable controller CPU to check the operating status of the programmable controller CPU.

The FB pasted to the sequence program can be monitored like contacts and coils.

The program in the FB can also be monitored like the sequence program.

[Operation Procedure]

• When monitoring

Choose [Online] → [Monitor] → [Monitor mode] or click 🖳 ([F3]).

• When stopping monitoring

Choose [Online] → [Monitor] → [Stop monitor] or click 🖳 ([Alt]+[F3]).

• When resuming monitoring

Choose [Online] → [Monitor] → [Start monitor] or click 🖳 ([F3]).

[Window]

The following explains the example where the FB pasted to the sequence program and the program in the FB are monitored.

(1) The sequence program is displayed on the following monitor window.
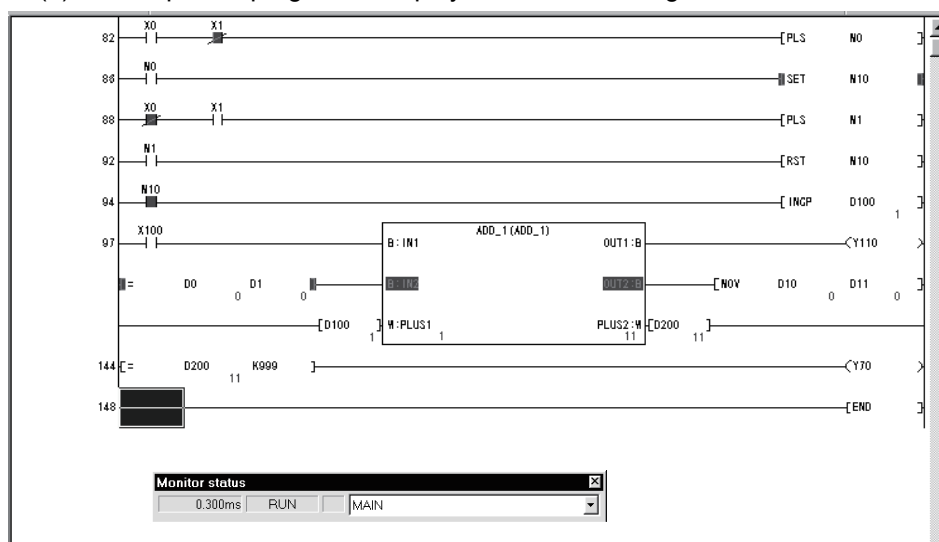
The ladder block that includes the FB in the sequence program is displayed on the monitor screen as described below.

(a) For input variable (BOOL)

1) When the condition is ON, the label of the input variable is highlighted.

Condition is ON.   Label is highlighted.

```
 =      D0        D1
         0         0           B : IN2
```
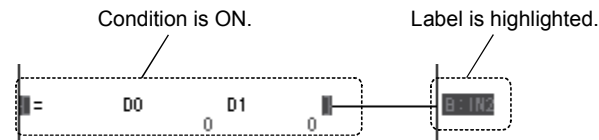
2) When the condition is OFF, the label of the input variable is not highlighted.

Condition is OFF.   Label is not highlighted.

```
  X100
  --| |--                           B : IN1
```

(b) For input variable (INT/DINT/REAL)

The numeric value stored under the label is displayed.

Numeric value of storage source device   Numeric value stored in input variable

```
                          [D100   ] W:PLUS1
                             1         1
```
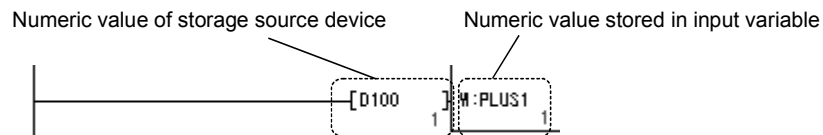
(c) For output variable (BOOL)

1) When the output variable is ON, the label is highlighted.

ON indication of output variable

```
 OUT2:B        [MOV      D10       D11      ]
                          0         0
```
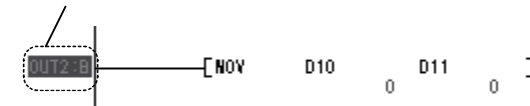
2) When the output variable is OFF, the label is not highlighted.

OFF indication of output variable

```
 OUT1:B                            (Y110    )
```
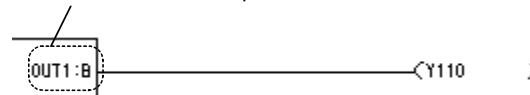
(b) For output variable (INT/DINT/REAL)

The numeric value stored under the label is displayed.

Numeric value stored in output variable

```
 PLUS2:W [D200   ]
    11      11
```

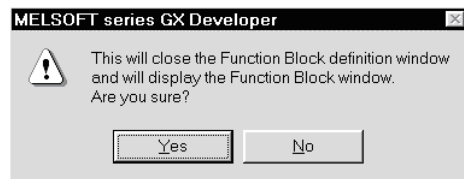Stored numeric value
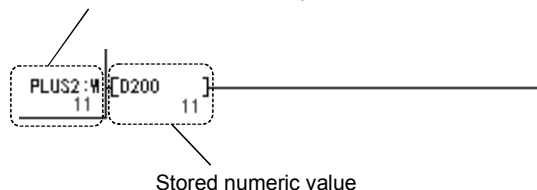
(2) When monitoring the FB program, double-click the FB on the sequence program
    to display the FB window. Then, execute monitoring on the displayed FB
    window.
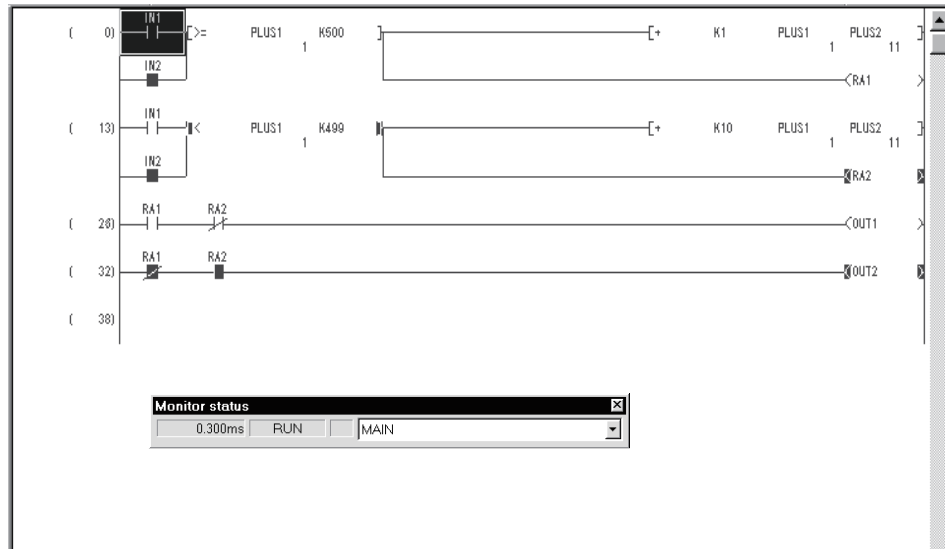
The following dialog box appears if the FB definition window or FB variable
setting window has already been displayed.

Clicking the $\boxed{\text{Yes}}$ button closes the FB definition window or FB variable setting
window and displays the FB window.

3.11.2 Testing the sequence program (Device test)

[Setting Purpose]

Forcibly turn ON/OFF the bit device of the programmable controller CPU or change the current value of the word device to check the created sequence program.

The FB pasted to the sequence program can be device-tested like the contacts and coils.

The program in the FB can also be device-tested like the sequence program.

[Operation Procedure]

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

[Dialog Box]
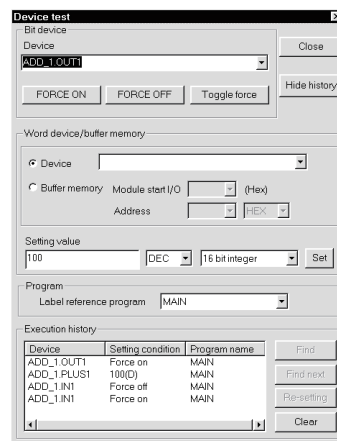
[Setting Procedure]

The following example explains the device test conducted on the FB pasted to the sequence program.

(1) Choose the target label of the FB to be device-tested.
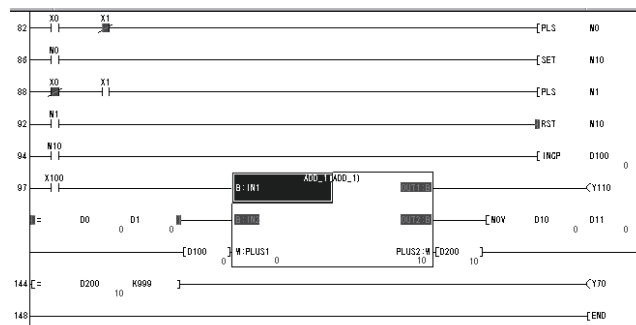
(2) If you choose [Device test], the device name (ADD_1.IN1) selected in the Device field has already been entered.

Selected label is entered.
It indicates "'FB name' . 'Label name'".
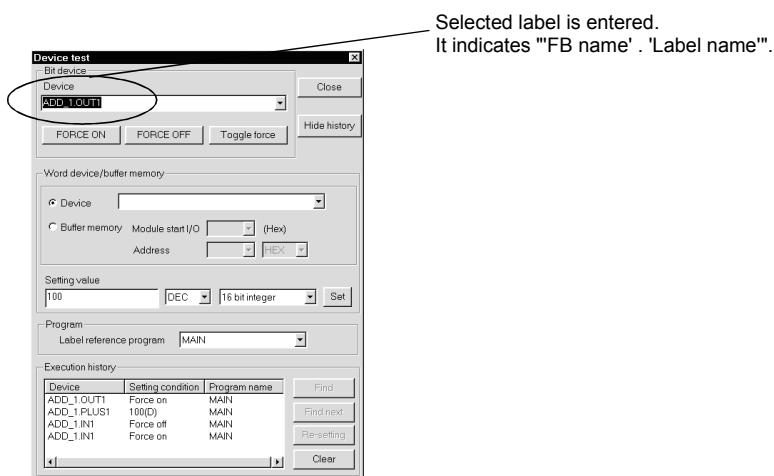
(a) When the device type of the selected label is the bit, the label is entered into the Device field of Bit device.
(b) When the device type is the INT/DINT/REAL, the label is entered into the Device field of Word device/buffer memory.

(3) Conduct the device test on the selected label.

Perform device test on selected label.
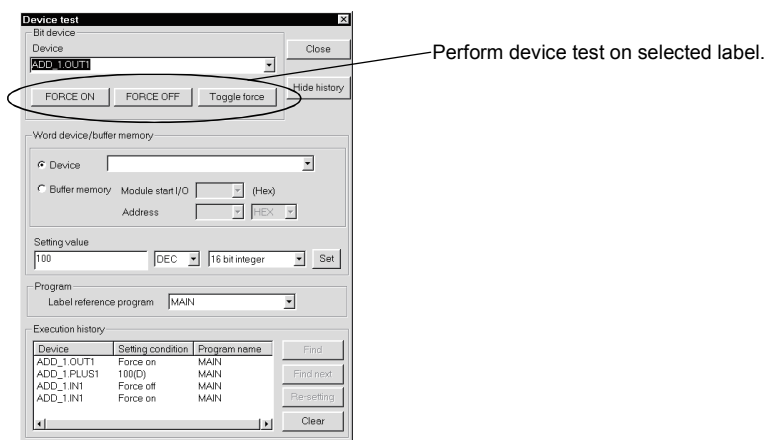
(4) The device test of the label is completed.

(5) When device-testing another label, enter the label directly into the Device field.
Type "'FB name' . 'Label name'" in the Device field.

POINT

Holding down the ⌷Shift⌷ key and double-clicking the label of the FB in the sequence program enables forced ON/OFF if the device type of the target label is the bit.
The device test dialog box appears if the device type of the target label is the INT/DINT/REAL.
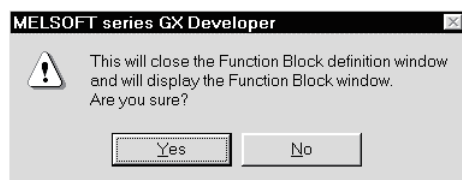

3.12 Correcting the FB

[Setting Purpose]

If the FB of the monitored/tested sequence program does not operate as expected, correct the program of the FB.

POINT

(1) The FB of the sequence program including the FB cannot be displayed if read from PLC is performed to read the program from the programmable controller CPU.
GX Developer displays the sequence program that has been compiled.
Hence, carefully store the program that has not yet been compiled since it is needed to correct the FB.

(2) Correct the FB program on the FB definition window.
The program cannot be corrected on the FB window.

(3) Note that if multiple pieces of the same FB definition have been pasted to the sequence program, correction to the FB definition program is reflected on all FBs.

[Setting Procedure]

(1) On the FB tab, double-click and choose the FB definition to be corrected.

(2) As the FB definition window appears, correct the program.
If the FB window has already been displayed, the following dialog box appears.

Clicking the $\boxed{\text{Yes}}$ button closes the FB window and displays the FB definition window or FB variable setting window.

(3) After completion of program correction, compile the FB definition.
If you do not execute compile, the correction is not reflected on the FB pasted to the sequence program.

(4) Compile the sequence program where the corrected FB definition has been pasted.
If you do not execute compile, the correction made to the FB is not reflected on the sequence program.

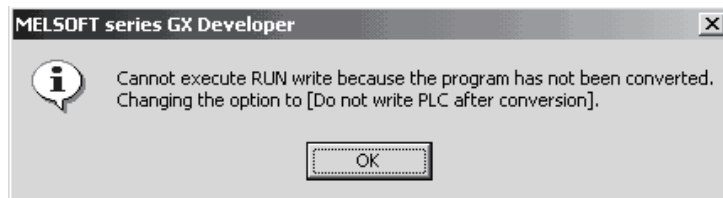## 3.13 Making Online Program Correction to the Sequence Program

[Setting Purpose]

Write the corrected sequence program to the programmable controller CPU that is in the RUN status.
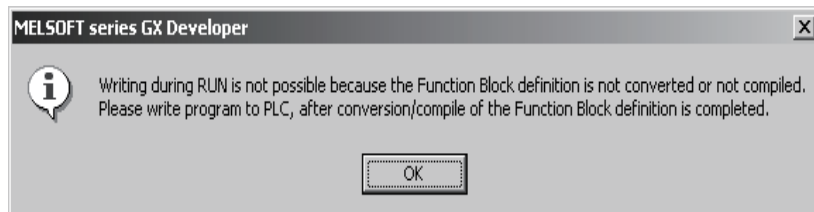
[Operation Procedure]

Choose [Convert] → [Convert/Compile (Online change)] or press $\boxed{\text{Shift}}$ + $\boxed{\text{F4}}$.

POINT

(1) For other precautions concerning online change, refer to GX Developer Operating Manual.

(2) If the FB definition to be online changed is utilized in multiple programs, online change cannot be made. The following message appears.

> MELSOFT series GX Developer
>
> (i) Cannot execute RUN write because the program has not been converted.
> Changing the option to [Do not write PLC after conversion].
>
> OK

(3) If the FB variable/FB definition program of the program to be online changed is changed, online change cannot be made. The following message appears.

> MELSOFT series GX Developer
>
> (i) Writing during RUN is not possible because the Function Block definition is not converted or not compiled.
> Please write program to PLC, after conversion/compile of the Function Block definition is completed.
>
> OK

---

4 UTILIZING THE EXISTING FB TO CREATE SEQUENCE PROGRAM

This chapter describes the operations for utilizing the created FB to create a sequence program.

The operation procedures in Section 4.1.1 and Sections 4.5 to 4.9 are the same as those in Section 3.1 and Sections 3.5 to 3.9. For explanations, refer to the corresponding sections of Chapter 3.

4.1 Creating Project for FB

4.1.1 Creating a new project

[Setting Purpose]

Set the PLC series, PLC type and Project name necessary to create a new project for the FB.

Refer to Section 3.1 for the explanation of the operation procedure.

4.1.2 Reading the existing project

[Setting Purpose]

Read the saved project to add the FB to the existing project.

[Operation Procedure]

Choose [Project] → [Open project] or click 📂 (Ctrl + O).

[Dialog Box]
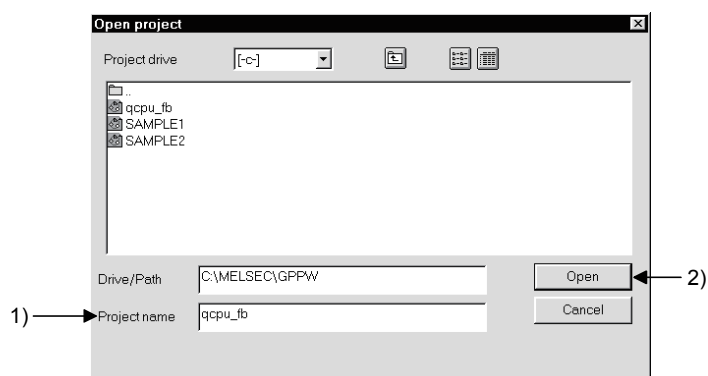
[Description]

1) Project name

Click the object project and set the project name.

2) Open button

After you finished the setting, click this button to read the project.

4.2 Utilizing the Existing FB Definition (Copy)

There are the following two ways to utilize the existing FB definition.
(1) Utilizing the existing FB definition from another project (refer to Section 4.2.1)
(2) Utilizing the existing FB definition within the same project (refer to Section 4.2.2)

These utilization methods will be described in the following sections.

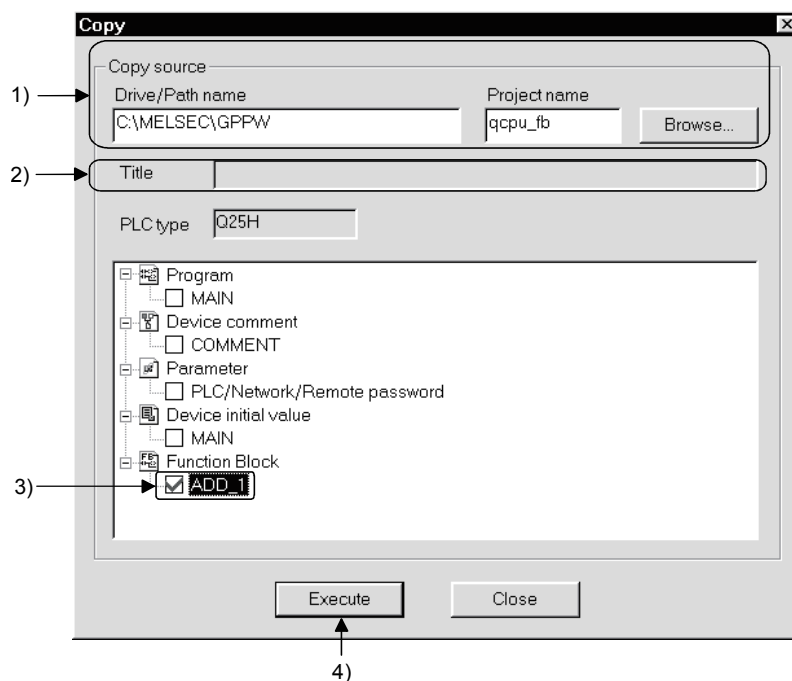## 4.2.1 Utilizing the existing FB definition from another project

[Setting Purpose]
  Copy the FB definition created in another project to utilize it.

[Operation Procedure]
  Choose [Project] → [Copy].

4

[Dialog Box]

[Description]
  1) Drive/Path name, Project name
     From the Browse button, set the project having the FB definition you want to

     copy.
  2) PLC type
     The PLC type of the copy source project is displayed.
  3) Copy source FB definition
     Choose the FB definition you want to copy.

4) ☐Execute☐ button
   Click this button to execute copy.

   When the program is to be copied from another project, the following dialog box is
   displayed.

a) ——————
b) ——————
c) ——————
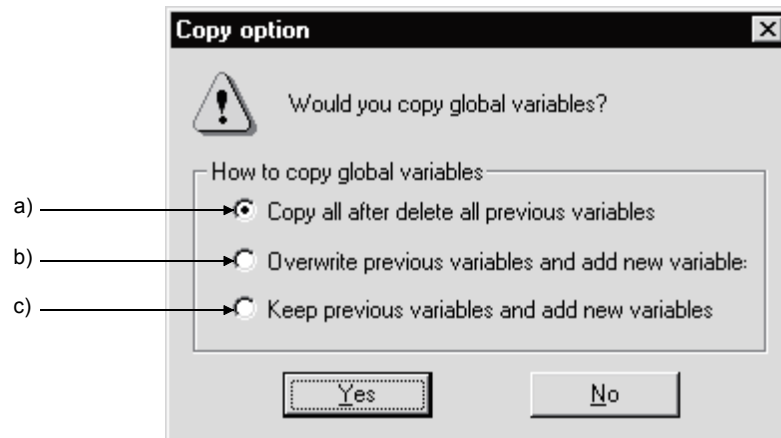
   a) "Copy all after delete all previous variables"
      The global label variable setting selected at the copy source is written over the
      copy destination.
   b) "Overwrite previous variables and add new variable"
      If the same global label exists at the copy source and copy destination, the
      global label at the copy source is written over the copy destination.
      The non-overlapping global label is added to the global variable setting at the
      copy destination.
   c) "Keep previous variables and add new variables"
      If the same global label exists at the copy destination and copy source, the
      global label at the copy source is not written over the copy destination.
      The non-overlapping global label is added to the global variable setting at the
      copy destination.

[Setting Procedure]
   (1) By clicking the ☐ Execute ☐ button on the copy dialog box, the FB definition having
       the same name as that of the copy source is created in the project.

   (2) If the FB definition having the same name as that of the copy source already
       exists in the copy destination, the following dialog box appears to indicate that
       copy has been stopped.

       To execute copy, change the FB definition name of the copy destination to
       another name and execute copy again.
       Changing the FB definition name also changes the FB definition name of the FB
       pasted to the sequence program.
       Refer to Section 4.3 for changing the FB definition name.

4.2.2 Utilizing the existing FB definition within the same project
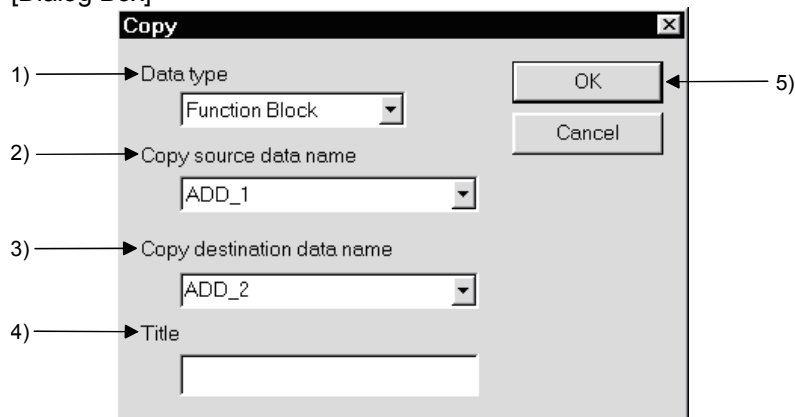
[Setting Purpose]

Copy the FB definition already created in the same project to utilize it.

[Operation Procedure]

- Click the right button of the mouse on the <<Function Block>> tab and choose →
  [Copy].
- Choose [Project] → [Edit Data] → [Copy].
- Click the right button of the mouse on the <<Project>> tab and choose → [Copy].

[Dialog Box]

[Description]

1) Data type

Choose Function block.

When you choose the FB definition of the copy source and open the Copy dialog
box, Function block has already been selected from the beginning.

2) Copy source data name

Set the data name (FB definition name) you want to copy.

When you choose the FB definition of the copy source and open Copy dialog box,
the data name has already been set from the beginning.

Alternatively, as the list box displays the FB definition names already created,
choose the object FB definition name from that list.

3) Copy destination data name
   Set the data name (FB definition name) of the copy destination.
   The list box displays the FB definition names already created. You can set the
   data name by choosing the FB definition name from that list and correct part of
   that name. Set the data name within eight characters.

   • Usable characters
     Alphanumeric characters, _, ^, $, , ~ (tilde), !, #, %, &, ( ), -, { }, @,
     * (apostrophe), ' (single quotation)
   • Unusable characters
     Any of " = | : ; , \ [ ] + * ? <> . / will result in an error.

4) Title
   Set the title to be set to the project within 32 characters.

5) OK button
   Click this button when the setting is finished.

[Setting Procedure]
   (1) Clicking the OK button on the copy dialog box displays the following dialog
       box.
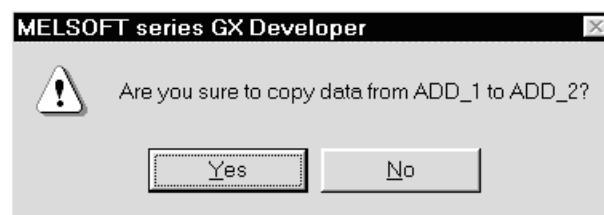
       Click the Yes button to start copy.

   (2) If the FB definition name of the copy destination already exists, the following
       dialog box appears to indicate that copy has been stopped.

4.3 Changing the FB Definition Name

[Setting Purpose]

Set a new name when you want to change the name of the FB definition copied from another project.

[Operation Procedure]

- Click the right button of the mouse on the <<Function Block>> tab and choose → [Rename].
- Choose [Project] → [Edit Data] → [Rename].
- Click the right button of the mouse on the <Project> tab and choose → [Rename].

[Dialog Box]
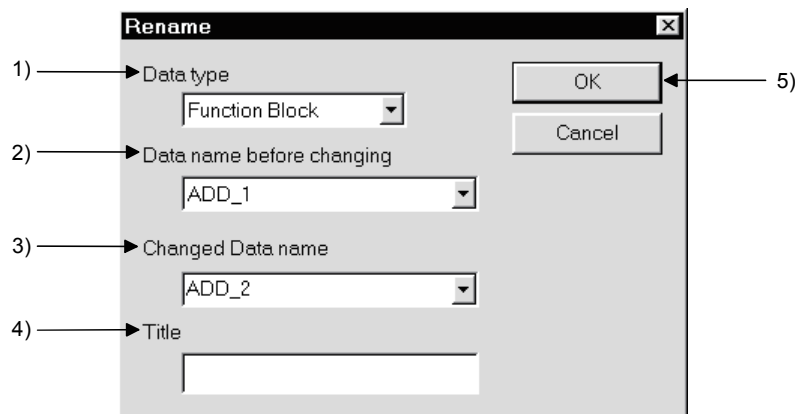
[Description]

1) Data type

Choose Function block.

When you choose the object FB definition and open the Rename dialog box, Function block has already been selected from the beginning.

2) Data name before changing

Set the data name (FB definition name) you want to change.

When you choose the object FB definition and open the Rename dialog box, the data name has already been set from the beginning.

Alternatively, as the list box displays the FB definition names already created, choose the object FB definition name from that list.

3) Change Data name

Set a new data name (FB definition name).
The list box displays the FB definition names already created. You can set the data name by choosing the FB definition name from that list and correct part of that name. Set the data name within eight characters.

• Usable characters
  Alphanumeric characters, _, ^, $, , ~ (tilde), !, #, %, &, ( ), -, { }, @,
  * (apostrophe), ' (single quotation)
• Unusable characters
  Any of " = | : ; , \ [ ] + * ? <> . / will result in an error.

4) Title

Set the title to be set to the project within 32 characters.

5) OK button

Click this button when the setting is finished.

4.4 Correcting the Program Body or FB Variables of the Utilized FB Definition

[Setting Purpose]

Correct the utilized FB definition to be the program that meets its purpose.

[Setting Procedure]

(1) Double-click the utilized FB to display the FB definition window or FB variable setting window.
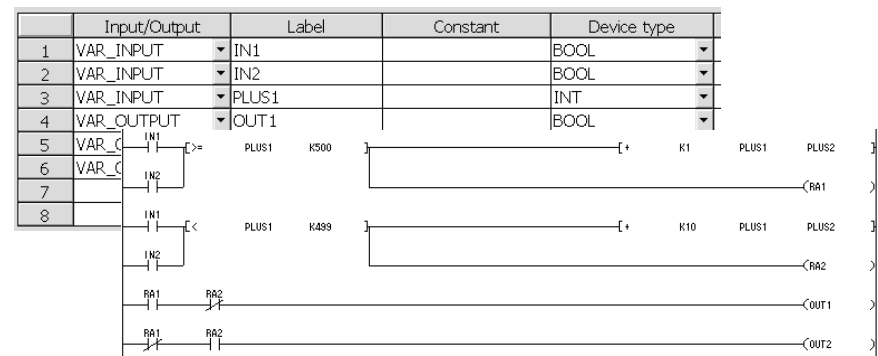
(2) Correct the program on the FB definition window.
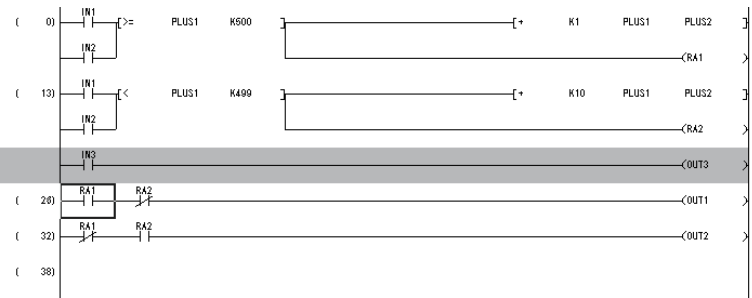
(3) When you have corrected, added and/or deleted the labels, edit the labels on the FB variable setting window.
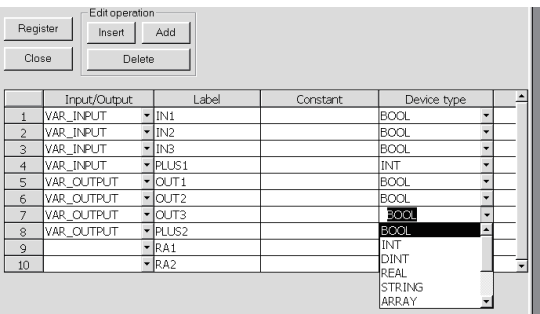
4.5 Converting the FB Definition (Convert (Compile))

[Setting Purpose]
Convert the corrected FB definition for use in the sequence program.

Refer to Section 3.5 for the explanation of the operation procedure.

4.6 Pasting the Converted FB to the Sequence Program (FB Utilization)

[Setting Purpose]
Paste the FB to the sequence program being created to enable the function block.

Refer to Section 3.6 for the explanation of the operation procedure.

4.7 Creating the Sequence Program

[Setting Purpose]
Create an input ladder section and an output ladder section for the FB pasted to the sequence program to complete the sequence program.

Refer to Section 3.7 for the explanation of the operation procedure.

4.8 Converting the Sequence Program into the Program to Be Executed on the Programmable Controller CPU (Convert (Compile))

[Setting Purpose]
The sequence program where the FB has been pasted cannot be executed as-is on the programmable controller CPU. Perform compile operation to convert it into the program to be executed.

Refer to Section 3.8 for the explanation of the operation procedure.

4.9 Writing the Converted Sequence Program to the Programmable Controller CPU (Write to PLC)

[Setting Purpose]
Write the converted sequence program to the programmable controller CPU.

Refer to Section 3.10 for the explanation of the operation procedure.

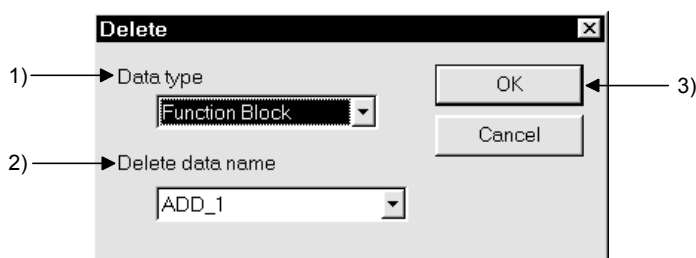# 5 DELETING UNNECESSARY FB DEFINITION FROM THE PROJECT

[Setting Purpose]

　Delete the FB definition that has become unnecessary due to project modifications.

[Operation Procedure]

　• Click the right button of the mouse on the <<FB>> tab and choose → [Delete].

　• Choose [Project] → [Edit Data] → [Delete].

　• Click the right button of the mouse on the <<Project>> tab and choose → [Delete].

[Dialog Box]

[Description]

　1) Data type

　　Choose Function block.

　　When you choose the object FB and open the Delete dialog box, Function block has already been selected from the beginning.

　2) Delete data name

　　Set the data name (FB definition name) to be deleted.

　　When you choose the object FB and open the Delete dialog box, the data name has already been set from the beginning.

　　Alternatively, as the list box displays the FB definition names already created, choose the object FB definition name from that list.

　3) ⌈ OK ⌋ button

　　Click this button when the setting is finished.

[Setting Procedure]

　(1) Setting the object FB definition name and clicking the ⌈ OK ⌋ button displays the following message.
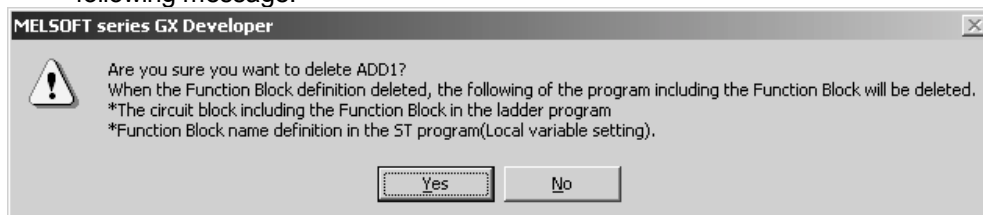
　(2) Click the ⌈ Yes ⌋ button to delete the FB definition.

5

POINT

(1) Deleting the FB definition also deletes the ladder block that includes the deleted FB definition (FB) from the sequence program.

Before executing delete, it is recommended to confirm by making character string search that the FB definition to be deleted is not being used in the sequence program. (Refer to Section 3.7.3.)

(2) When deletion of the FB definition is to be executed, the following message appears to indicate that conversion has been stopped if the sequence program that includes the FB definition (FB) to be deleted is not yet converted.

MELSOFT series GX Developer	✕
ⓘ Execution is not possible because program containing Function Block is not changed.	
OK	
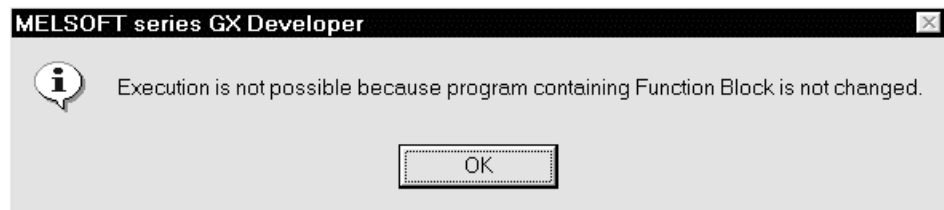
Before executing delete, convert the sequence program or delete the ladder block that includes the FB first from the sequence program.

6 COMBINING MULTIPLE PROGRAMS INTO ONE

6.1 Combining Sequence Programs into One

[Setting Purpose]

When multiple sequence programs including FBs exist, combine the sequence programs into one.

[Setting Procedure]

The sequence programs including FBs cannot be combined into one by the operation performed using [Merge data] in the [Tools] menu.
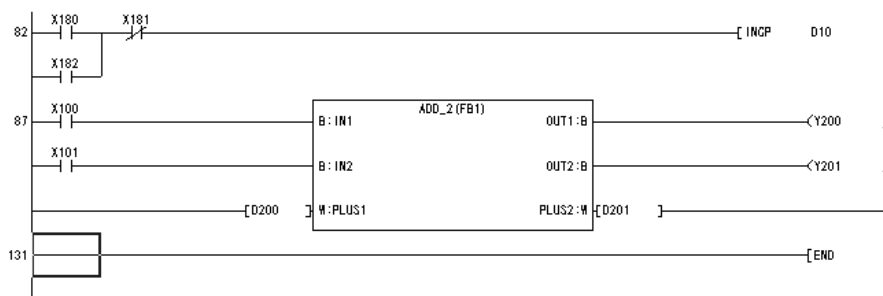
Hence, perform copy and paste operation between the sequence programs in the same project to combine them into one.

The procedure will be described using the following two programs as an example.

(a) Copy destination program: MAIN
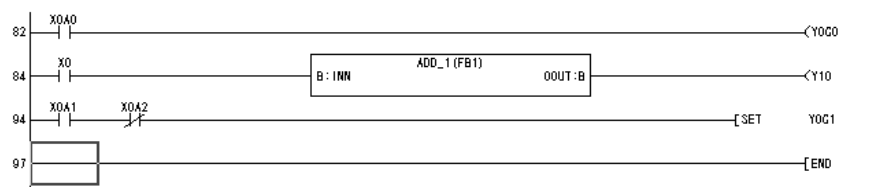
(b) Copy source program: MAIN1

(1) Select and copy the area from step 0 to the necessary step of the copy source sequence program.

6
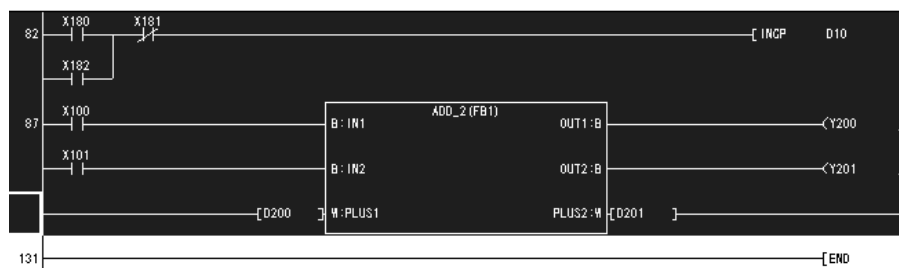
(2) Paste the copied sequence program to the step of the copy destination sequence program where you want to make insertion.
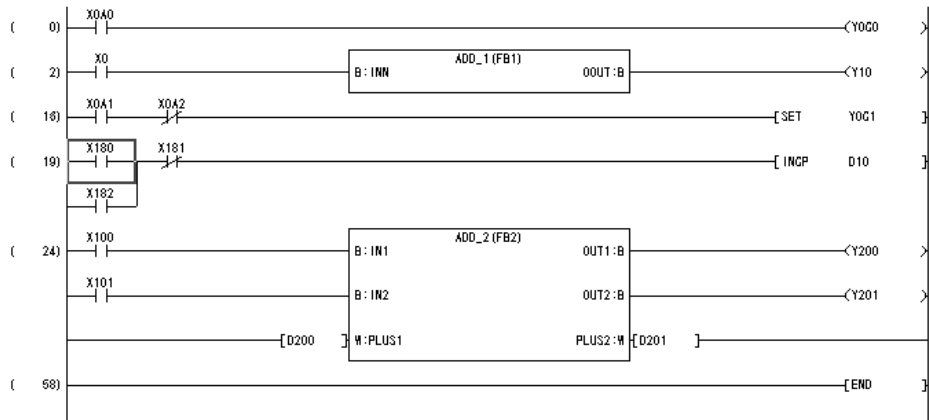
(3) Convert and save the sequence program.

```
┌─────────────────────────────────────────────────────────────────┐
│  POINT                                                            │
├───────────────────────────────────────────────────────────────── │
│ Copy and paste operation cannot be performed between the sequence │
│ program of one project and that of the other project.             │
│ In this case, perform the following operation.                    │
│ (1) Choose [Project] → [Copy] to copy the sequence program and FB │
│     definition of the other project.                              │
│     At this time, copy cannot be executed if the PLC type of the  │
│     copy source project differs from that of the copy destination │
│     project.                                                      │
│     In this case, change the PLC type of the copy source project  │
│     to that of the copy destination project and save it as        │
│     another project.                                              │
│ (2) Perform copy and paste operation between the sequence programs│
│     of the same project.                                          │
└───────────────────────────────────────────────────────────────────┘
```

6.2 Combining Comments into One (Data Joint)

[Setting Purpose]
Combine the comments of multiple sequence programs into one.

[Operation Procedure]
Choose [Tools] → [Merge data].

```
┌──────────────────┐
│  REMARK          │
└──────────────────┘
For details of the operation, refer to "Merge data" of the GX Developer Version 8
Operating Manual.
```

```
┌───────────────────────────────────────────────────────────────────┐
│  POINT                                                              │
├─────────────────────────────────────────────────────────────────── │
│ • Comments not saved in the personal computer cannot be combined.   │
│ • Merge data cannot combine the sequence programs that include FBs.  │
└───────────────────────────────────────────────────────────────────┘
```

7 VERIFYING THE PROJECTS

[Setting Purpose]

Verify data between the projects of the same PLC type.

[Operation Procedure]

Choose [Project] → [Verify].

REMARK

For details of the operation, refer to "Verifying Data in Projects" of the GX Developer Version 7 Operating Manual.

Executing project verify displays the verify results of the following items.

(1) Parameters

(2) Label programs

(3) Actual programs

The following are the display examples of verify results.

(a) Verify results: When the results are OK

```
[Project verify: Label program]
 Verify source
   Project name -F:\FB_ENG2
   Data name -MAIN
 Verify destination
   Project name -F:\FB_ENG2
   Data name -MAIN

   <Memory>                                      <File>
Row number: Text                              Row number: Text


No unmatched areas.
```

(b) Verify results: When the results are not good

```
[Project verify: Label program]
 Verify source
   Project name -F:\FB_ENG3
   Data name -MAIN
 Verify destination
   Project name -F:\FB_ENG2
   Data name -MAIN

   <Memory>                                      <File>
Row number: Text                              Row number: Text

    9                                  There is no corresponding row.

   11   IF S_LBL=TRUE THEN             There is no corresponding row.

   12      FB3(IO_TEST:=M20);          There is no corresponding row.

   13   END_IF;                        There is no corresponding row.

   15   FB4(IO_TEST:=M20);             There is no corresponding row.

   17   IF S_LBL=FALSE THEN            There is no corresponding row.

   18      M123:=TRUE;                 There is no corresponding row.

   19                                  There is no corresponding row.

   20   END_IF;                        There is no corresponding row.

   24                                  There is no corresponding row.

   25                                  There is no corresponding row.

11 items unmatched.
```

7

POINT

(1) Verified data are as follows.
   • Parameters
   • Sequence programs where FBs have been pasted
   • Actual programs available by compiling the above sequence programs
    (programs written to the programmable controller CPU)
   FB definitions (program bodies, FB variables) are not verified.

(2) If sequence programs not yet compiled are verified, the following message
   appears to indicate that verify has been stopped.
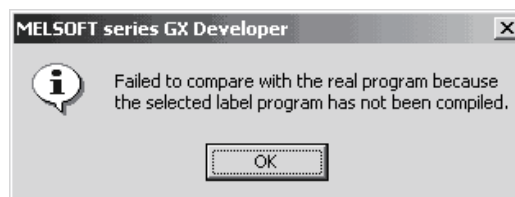
(3) Multiple data can be selected as shown below for the verify source and verify
   destination. Verify is performed between the data of the same name.
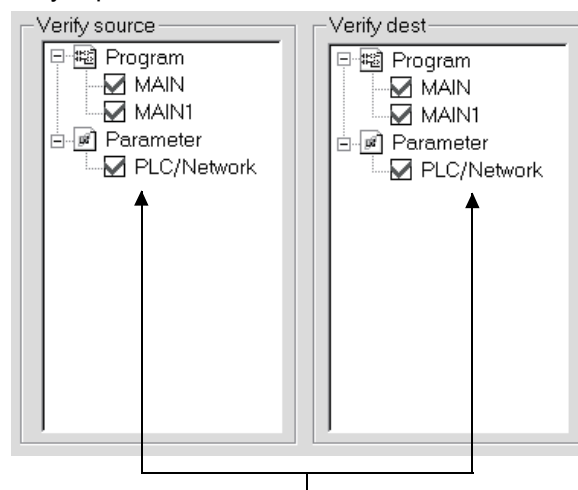
Choose the data names by turn on their check boxes.

8 PRINTING

This chapter describes the operation methods for printing the sequence program including FB and printing the FB definition.
For other printing methods, refer to the GX Developer Version 7 Operating Manual given in the related manuals.

8.1 Printing the Sequence Program Including FB

[Setting Purpose]

Print the sequence program that includes the FB.

[Operation Procedure]

Choose [Project] → [Print] or click 🖨 (Ctrl + P).

[Dialog Box]

(1) Ladder program
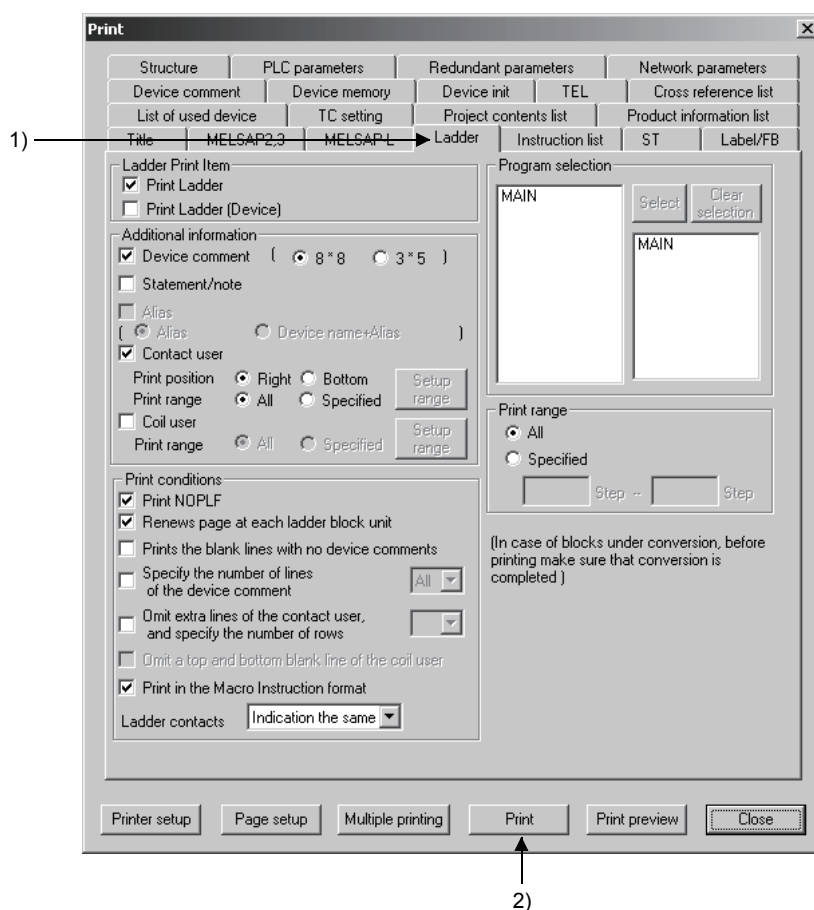
[Description]

1) <<Ladder>> tab

Click this tab to switch to the print detail setting screen.

2) Print button

Click this button to print the sequence program.

8

[Setting Procedure]
  (1) After setting the necessary information for printing, click the [Print] button to start printing.

(Print example) Ladder program
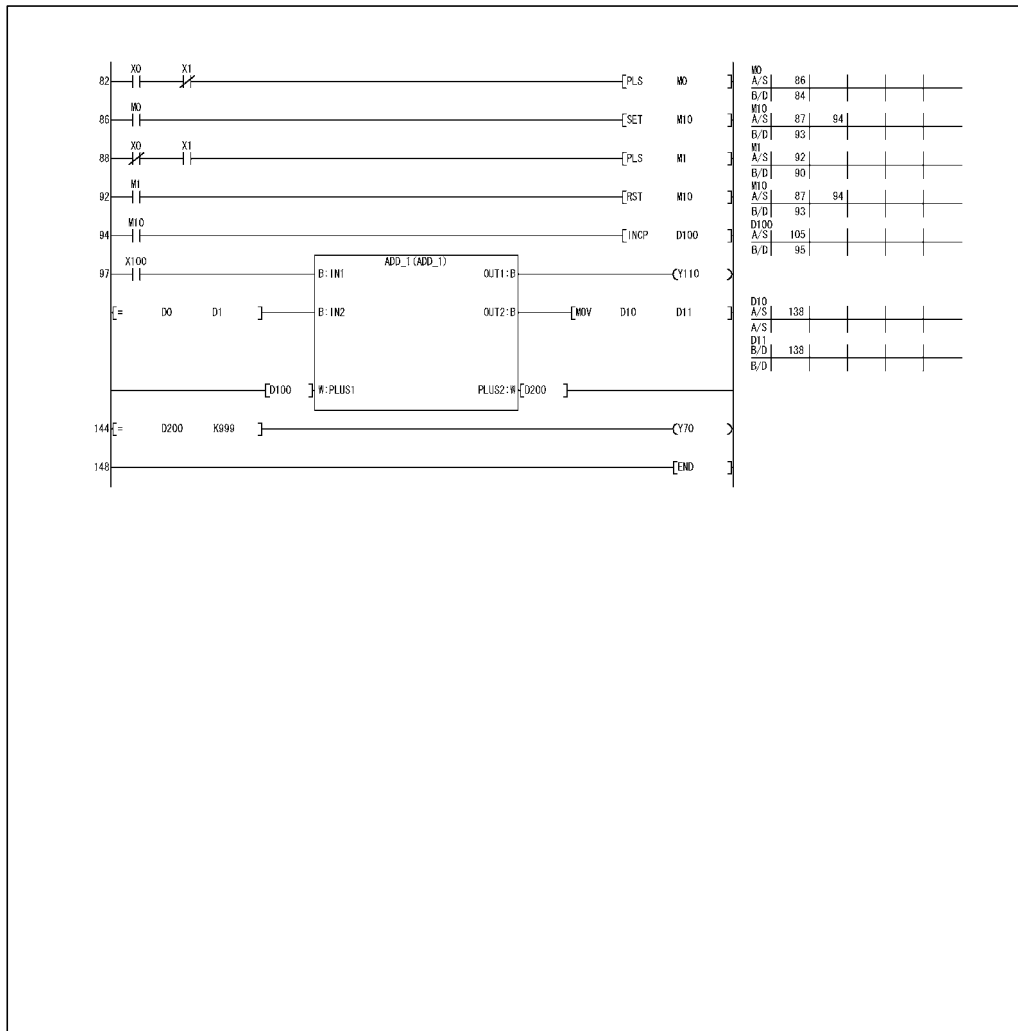
REMARK

The following items cannot be printed when the FB was used to create the program.
  • MELSAP2, 3
  • MELSAP-L
  • List printing

(2) ST program
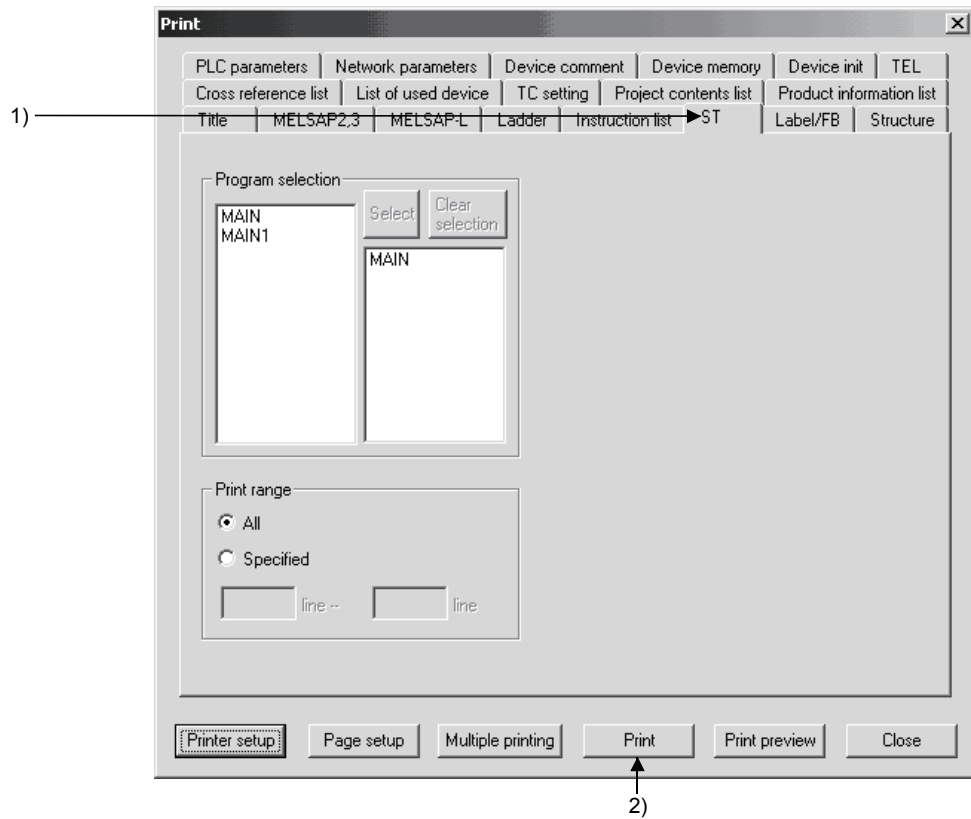
[Description]
  1) <<ST>> tab
     Click this tab to switch to the window for setting the print details.
  2) [Print] button
     Click this button to print the ST program.

[Setting Procedure]
  (1) After setting the data necessary for printing, click the [Print] button to execute
     printing.

  (Printing example) ST Program

```
 1 IF S LBL=TRUE THEN
 2   FB1(IO_TEST:=M0);
 3 END_IF;
 4
 5 FB2(IO_TEST:=M0);
 6
 7 IF S LBL=TRUE THEN
 8   M0:=TRUE;
 9
10 END IF;
11 IF S LBL=TRUE THEN
12   FB3(IO_TEST:=M10);
13 END_IF;
14
15 FB4(IO_TEST:=M20);
16
17 IF S LBL=FALSE THEN
18   M123:=TRUE;
19
20 END_IF;
21
22
23
24
```

8.2 Printing the FB Definition

[Setting Purpose]

Print the FB definition program and/or FB variables of the FB definition.

[Operation Procedure]

Choose [Project] → [Print] or click 🖨 (Ctrl + P).

[Dialog Box]
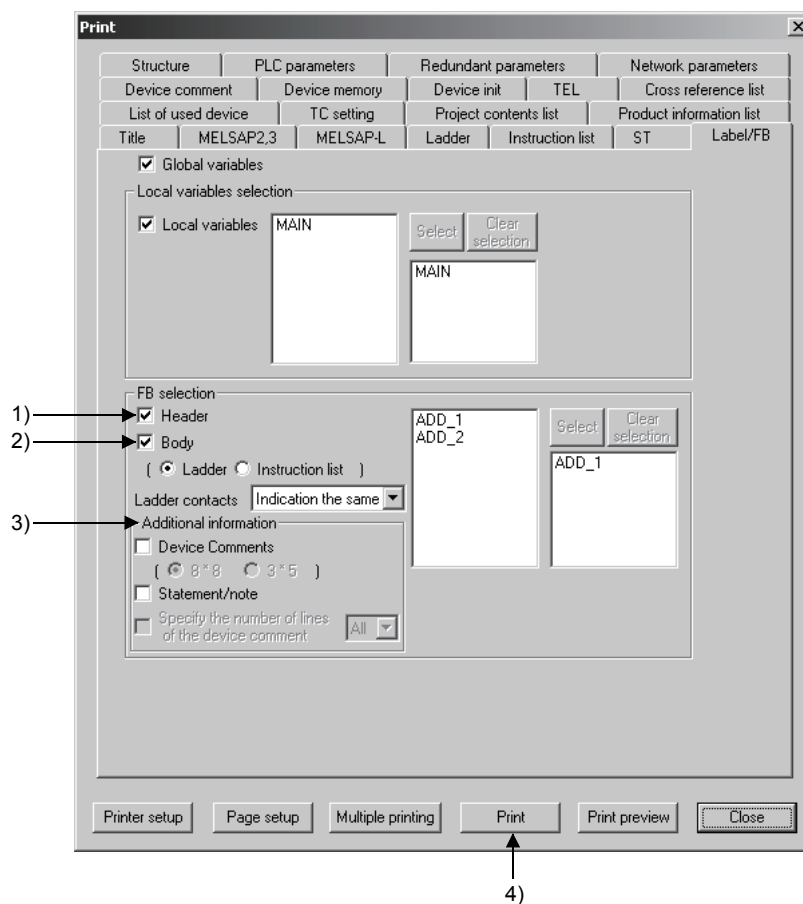
[Description]

1) Header

Turning on the check box prints the FB variables.

2) Body

Turning on the check box prints the FB definition program.
Choose the program format from Ladder or List. It defaults to Ladder.

3) Additional information

Select whether the FB definition program to be printed is annotated with device comments and/or statements/notes.
Select the corresponding check box to annotate the program.
The number of device comment lines to be printed can be specified.

4) Print button

Click this button to print the sequence program.

[Setting Procedure]
(1) Choose the FB definition to be printed.
Click and highlight the FB definition to be printed from the list of the FB definitions registered to the project, and then click the [Select] button.

FB definitions registered to the project are listed.

Click the Select button to choose the FB definition.
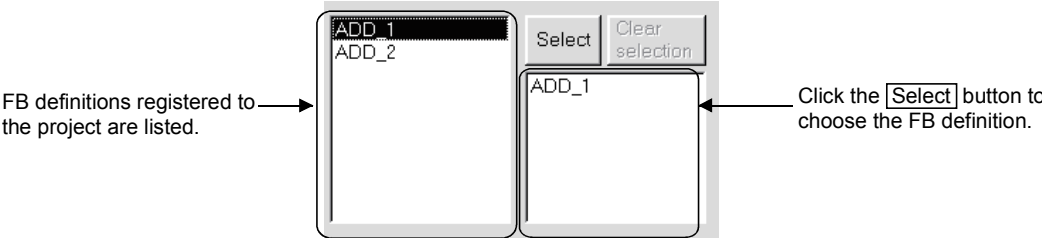
(2) Select the item to be printed (Header, Body) and click the [Print] button to start printing.

(Print example: FB variables)

(Print example: Program body)
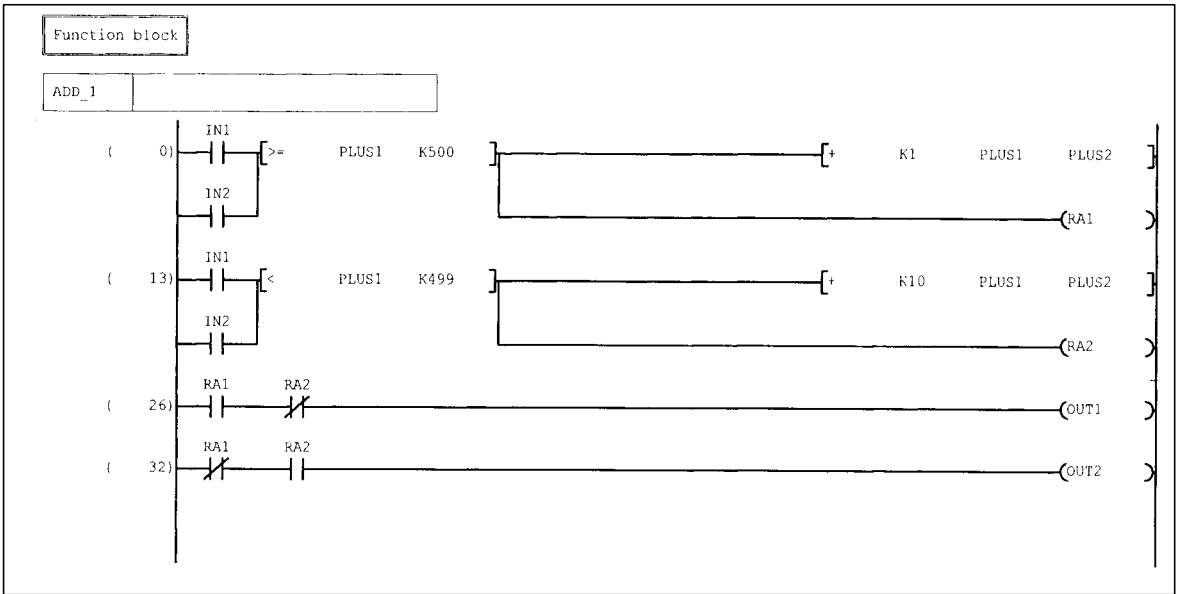
APPENDICES

Appendix 1 Character Strings Unusable as Labels and FB Names

The following character strings are unusable as labels and FB names for function block programming.
The string of device name, sequence instruction, SFC instruction, application instruction can not be used as label.
An error may occurred during registering or at compile time when any of the following strings are used

	The strings which can not be used in label programming
A	A,ACJ,ADD,ANB,AND,ANDF,ANDN,ANDP,ANI,ANY,ANY_BIT,ANY_DATE,ANY_DERIVED, ANY_ELEMENTARY,ANY_INT,ANY_MAGNITUDE,ANY_NUM,ANY_REAL,ANY_SIMPLE,ANY_STRING, ARRAY
B	B,BCD(P),BEND,BIN(P),BKBCD(P),BKBIN(P),BL,BLOCK,BMOV(P),BOOL, BOOL_TO_BYTE(DINT,DWORD,INT,REAL,SINT,UDINT,UINT,USINT,WORD), BYTE(DINT,DWORD,INT,REAL,SINT,TIME,UDINT,UINT,USINT,WORD)_TO_STRING, BYTE_TO_BOOL(DINT,DWORD,INT,REAL,SINT,UDINT,UINT,USINT,WORD), B_BCD_TO_DINT(INT,SINT),BXCH(P),BYTE
C	C,CAL,CALC,CALCN,CJ,CML(P)
D	D,DBCD(P),DBIN(P),DBL(P),DCML(P),DDEC(P),DEC(P),DELTA(P),DFLT(P),DGBIN(P),DGRY(P),DI, DINC(P),DINT,DINT(P),DINT_TO_BCD(BOOL,BYTE,DWORD,INT,REAL,SINT,TIME,UDINT,UINT,USINT, WORD),DIV,DMOD,DMOV(P),DNEG(P),DWORD,DWORD_TO_BOOL(BYTE,DINT,INT,REAL,SINT, UDINT,UINT,USINT,WORD),DX,DXCH(P),DY,D_BCD_TO_DINT(INT,SINT)
E	E,EGF,EGP,EI,EMOV(P),END,ENEG(P),EQ,EQ(GE,GT,LE,LIMIT,LT,MAX,MIN,NE,SEL)_STRING
F	F,FD,FEND,FF,FLT(P),FMOV(P),FX,FY
G	G,GBIN(P),GE,GOEND,GRY(P),GT
H	H
I	I,IMASK,INC(P),INT,INT(P),INT_TO_BOOL(BYTE,DINT,DWORD,REAL,SINT,UDINT,UINT,USINT,WORD) ,INV,IRET
J	J,JMP,JMPC,JMPCN
K	K
L	L,LD,LDF,LDI,LDN,LDP,LE,LED,LEDA,LEDB,LEDC,LEDR,LINT,LREAL,LT,LWORD
M	M,MC,MCR,MEF,MEP,MOD,MOV(P),MPP,MPS,MRD,MTR,MUL
N	N,NE,NEG(P),NOP,NOPLF,NOT
O	OR,ORB,ORF,ORI,ORN,ORP,OUT(H)
P	P,PAGE,PCHK,PLF,PLS,PLSY,PWM
Q	Q
R	R,RAMP,RCJ,READ,REAL,REAL_TO_BOOL(BYTE,DINT,DWORD,INT,SINT,UDINT,UINT,USINT,WORD) ,RECV,REQ,RET,RETC,RETCN,RFRP,RFS,ROTC,RST,RTOP

	The strings which can not be used in label programming
S	S,SB,SCJ,SD,SEND,SET,SFCP,SFCPEND,SFT(P),SG,SINT,SINT_TO_BOOL(BYTE,DINT,DWORD,INT, REAL,UDINT,UINT,USINT,WORD),SM,SPD,SREAD,ST,STEPC,STEPD,STEPG,STEPI,STEPID,STEPIR, STEPISC,STEPISE,STEPIST,STEPN,STEPR,STEPSC,STEPSE,STEPST,STMR,STN,STOP,STRING, STRING_TO_BYTE(DINT,DWORD,INT,REAL,SINT,TIME,UDINT,UINT,USINT,WORD),SUB,SW, SWAP(P),SWRITE,SZ
T	T,TIME,TR,TRANA,TRANC,TRANCA,TRANCO,TRANCOC,TRANJ,TRANL,TRANO,TRANOA,TRANOC, TRANOCA,TRANOCJ,TRANOJ,TRUNC_DINT(INT,SINT),TTMR
U	U,UDCNT1(P),UDCNT2(P),UDINT,UDINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UINT,USINT, WORD),UINT,UINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UDINT,USINT,WORD),ULINT, USINT,USINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UDINT,UINT,WORD)
V	V,VAR,VAR_CONSTANT,VAR_EXT,VAR_EXTERNAL,VAR_EXTERNAL_CONSTANT, VAR_EXTERNAL_FB,VAR_EXTERNAL_PG,VAR_GLOBAL,VAR_GLOBAL_CONSTANT, VAR_GLOBAL_FB,VAR_GLOBAL_PG,VAR_IN_OUT,VAR_INPUT,VAR_OUTPUT,VAR_TEMP,VD,VOID
W	W,WORD,WORD(P),WORD_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UDINT,UINT,USINT), WRITE,WSTRING,W_BCD_TO_DINT(INT,SINT)
X	X,XCH(P),XOR,XORN
Y	Y
Z	Z,ZCOM,ZNRD,ZNRF,ZNTO,ZNWR,ZR
Function group	MELSEC function group, IEC function group

Precaution on label name

1. Can not use the space character.

2. Can not use the numerals as the starting character.

3. Can not use the following:
(,),*,/,+,-,<,>,=,&,!,",#,$,%,', ^,|,@,[,],{,},;,:,.,,,?,\,_
But, in case of underscore character, an error may occurred when it is present at the end or two or more underscore characters are used consecutively

4. A device name followed by any of 0 to F also results in an error.
Examples: XFFF, M100

5. Do not use "EnDm" as a label name (Example: E001D9).
(n and m are any values.)
If such a label name is used, it may be recognized as a real number value and unavailable as a label name.

App.

INDEX

GX Developer Version 8

Operating Manual (Function Block)

MODEL	GXDEV8-O-FB-E
MODEL CODE	13JU44
SH(NA)-080376E-K(0812)MEE	

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

Specifications subject to change without notice.