Structured Text(ST)

Programming Guide Book

MITSUBISHI



Mitsubishi Programmable Logic Controller



• SAFETY PRECAUTIONS •

(Always read these instructions before using this equipment.)

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

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

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



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

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

[Designing Precautions]

For data change, program change and status control to be performed for the running
programmable logic controller from a personal computer, configure interlock circuits in the
outside of the PLC system so that the whole system will always operate safely.
 Also, for online operations to be performed from the personal computer to the PLC CPU,
predetermine as a system the corrective actions for communication errors that will occur due to
poor cable connection, etc.

[Startup/Maintenance Precautions]

 Before performing online operations (program change during PLC CPU RUN, forced I/O operation, RUN-STOP or similar operating condition change, remote operation) with the personal computer connected to the running PLC CPU, read the manual carefully and ensure safety fully.

Note that program change during PLC CPU RUN (online change) may cause such problems as program corruption depending on the operation condition. Use the equipment after fully understanding the precautions given in the GX Developer Operating Manual.

REVISIONS

I		* The manual number is given on the bottom left of the back cover.
Print Date	* Manual Number	Revision
Feb., 2003	IB (NA)-0800248E-A	First printing
Jul., 2003	IB (NA)-0800248E-B	Correction
		Section 5.3
Oct., 2003	IB (NA)-0800248E-C	Correction
		Section 5.2
Jun., 2004	IB (NA)-0800248E-D	Correction
		Abbreviations and Generic Terms in This Manual, Chapter 3
		· · · ·
		Japanese Manual Version SH-080365-E

Japanese Manual Version SH-080365-E

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

© 2003 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

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

CONTENTS

SAFETY PRECAUTIONS	A- 1
REVISIONS	A- 2
CONTENTS	A- 4
About Manuals	A- 5
How to Use This Manual	A- 6
Abbreviations and Generic Terms in This Manual	A- 7

1 OVERVIEW

3 ST PROGRAMMING	3- 1 to 3- 16
Creating a new ST project	
Defining the labels	
Entering a program	
Converting (compiling) the ST program	
4 READ/WRITE FROM/TO PLC CPU	4- 1 to 4- 2

5 DEBUGGING THE PROGRAM	5- 1 to 5- 7
5.1 Monitoring the Sequence Program 5.2 Device Test	
5.3 Online Change	
6 SAVING THE PROGRAM INTO THE PERSONAL COMPUTER	6- 1 to 6- 2

7 INTRODUCTION TO USEFUL FUNCTIONS FOR ST PROGRAM EDITING 7-1 to 7-2

1-1 to 1-2

8 ST PROGRAMMING APPLICATION (PASTING FB TO LADDER PROGRAM)

8-1 to 8-20

8.1 Creating an FB	
Creating a new project	
Adding a new FB	
Defining FB variables	
Creating an FB in ST language	8- 9
8.2 Pasting the FB to a Main Program	
Defining the local variables	
Creating a main program	
8.3 Online	
Writing to PLC CPU	
Monitoring the sequence program	
Confirming the program behavior	

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 Version8 Operating Manual (Startup) Explains the system configuration, installation method and startup method of GX Developer. (Option)	IB-0800242E
GX Developer Version8 Operating Manual Explains the program creation method, printout method, monitoring method, debugging method, etc. using GX Developer. (Option)	IB-0800243E
GX Developer Version8 Operating Manual (Structured Text) Explains the operation methods for creating structured text programs. (Option)	IB-0800247E
GX Developer Version8 Operating Manual (Function Block) Explains the program creation method, printout method, etc. using GX Developer. (Option)	IB-0800246E
QCPU (Q mode) Programming Manual (Structured Text) Explains the programming methods in structured text language. (Option)	SH-080366E (13JF68)
QCPU (Q mode)/QnACPU Programming Manual (Common Instructions) Explains the methods of using the sequence instructions, basic instructions and application instructions. (Option)	SH-080039 (13JF58)

REMARK

Each Operating Manual is contained in the CD-ROM together with the software package as a set.

The Programming Manual is available separately in printed form as an option. Please place an order with the manual number (model code) in the above table.

How to Use This Manual

This Guidebook ...

This guidebook is a commentary written for those who will use the GX Developer Version 8 software package (hereafter abbreviated to GX Developer) to create structured text (hereafter abbreviated to ST) programs for the first time. "Chapter 1" introduces the overview of the ST language and the features of the ST

language in the MELSEC-Q series. "Chapter 2 to Chapter 6" introduce a series of basic operation methods, such as the methods of creating, debugging and saving programs in ST language through sample programs.

"Chapter 7" introduces useful functions available from GX Developer. "Chapter 8" introduces the method of creating a program, which uses an ST-written

function block (FB) in a ladder program from the main program created in ladder form, as an application program.

"Chapters 4, 5 and 8" use the PLC CPU for explanation.

Programming Manual ...

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

Operating Manual ...

The "GX Developer Version 8 Operating Manual (Structured Text)" is a commentary that gives in-depth explanation of the operation methods for creating structured text programs using GX Developer. Refer to the manual when information on operations details is necessary.

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

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

Abbreviations and Generic Terms in This Manual,

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

Generic Terms and Abbreviations	Description/ target unit
ST	Stands for structured text.
FB	Stands for function block.
GX Developer	Generic product name for model names SWnD5C-GPPW, SWnD5C-GPPW-A, SWnD5C-GPPW-V and SWnD5C-GPPW-VA. n means Version 8 or later.
Basic model QCPU	Generic term for Q00JCPU, Q00CPU and Q01CPU of function version B or later
High Performance model QCPU	Generic term for Q02 (H) CPU, Q06CPU, Q12HCPU and Q25HCPU
Process CPU	Generic term for Q12PHCPU and Q25PHCPU
Redundant CPU	Generic term for Q12PRHCPU and Q25PRHCPU
QCPU (Q mode)	Generic term for Q00(J)CPU, Q01CPU, Q02(H)CPU, Q06HCPU, Q12HCPU, Q25HCPU, Q12PHCPU, Q12PRHCPU, Q25PHCPU and Q25PRHCPU.

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

Symbol	Description	Example
Point	Gives the section-related knowledge and useful information.	Point
[]	Menu name of menu bar	[Project]
()	Icon of toolbar	2
<< >>	Tab name of dialog box	< <select file="">></select>
	Command button of dialog box	Jump Button

What is the ST language ?	The ST language is defined in the International Standard IEC61131-3 that stipulates the logic description system in open controllers. The ST language supports operators, control syntaxes and functions to permit the following descriptions.
	Control syntaxes such as conditional sentence-dependent selective branch and repetitive sentence-based repetition
	 Expressions using operators (*, /, +, -, <, >, =, etc.)
	Call of user-defined function blocks (FB)
	Call of functions (MELSEC functions, IEC functions)
	Description of comments
What are the features ?	The following introduces the main features of ST programs in the MELSEC-Q series.
W in Tł	esign efficiency improved by defining processings as parts. /ith often used processings defined as parts in the form of function blocks (FB) ST language, they can be used in necessary areas of each program. his not only enhances the efficiency of program development but also reduces ogram mistakes, improving program quality.
	rogram change during system operation (online change). art of a running program can be changed without the PLC CPU being stopped.
Si fo Fo pr	onnection with other language programs. nce other languages than the ST are also supported, the language adequate r processing can be used to increase the efficiency of program development. or example, write sequence control in a ladder program, and operation ocessing in ST language. ultiple languages support widespread application under optimum control.
	wealth of functions available.

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

1

MEMO

MELSOFT

The following flowchart indicates the basic procedure from ST program creation to online debugging.

In the following example, programming was performed using only an ST program.



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

MEMO

-	

MELSOFT

Chapter 3 explains general basic operations from the input to convert (compile) of an ST program. The following items will be explained in this chapter.

L. P.	Creating	a new ST	project.
-------	----------	----------	----------

- Defining the labels to be used in an ST program.
- Creating an ST program.
- Converting (compiling) the created ST program into an executable sequence program.
- Correcting the program if a convert (compile) error occurs.

Creating a new ST project.

Creating a new ST project

The operation method to create a new project will be explained.



 Click [Project] → [New project] in the menu.

3 ST PROGRAMMING

3

From previous page		
Set the New project dialog.	 2) Enter as follows. PLC series PLC Type Label setting Program type 3) Click the OK butt 	: ST
Image: Sold Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Project Edit End[Replace Convert Yew Online Diagnostics Tools Window Help Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (Unset project) - [ST MAIN 1Row 05tep] Image: Sold Developer (St Main 1Row 05tep]	 4) A new ST project i * The ST edit screen program can be en 	n opens and an ST

REMARK

Here, "Q02(H)" is set as the PLC type.

There are the following PLC CPU types that are applicable to ST programs.

Basic model QCPU	High performance model QCPU	Process CPU	Redundant CPU
Q00CPU Q00JCPU Q01CPU	Q02CPU Q02HCPU Q06HCPU Q12HCPU	Q12PHCPU Q25PHCPU	Q12PRHCPU Q25PRHCPU
	Q25HCPU		

Defining the labels

What does "defining the labels" mean?	To use labels, variables used as labels must be clarified. This is called "defining the labels". If a program that uses undefined labels is converted (compiled), an error occurs and a sequence program cannot be created.
	There are two different labels; global variables and local variables. The global variables can be used in the whole project. The local variables can be used in only the program where the labels have been defined.
	Here, the local variables used in the program example that will be entered later will be actually defined.
Displaying the Local variables setting The operation method to define local variables, refer to the "G	ariables will be explained.
(Unset project) Global variables Program Parameter Device comment 1) Parameter Device memory Device init Project FB Structure	1) Double-click "Program" in the < <project>> tab.</project>
	2) Double-click "MAIN".
Image: Construction of the second	2) Double-click "MAIN".
	ō next page



Setting the local variables (Header)

1) Enter a label name.

Enter a label name within 16 characters. The character strings that cannot be used as a label name are reserved words and actual devices. Enter other labels.

* For the reserved words, refer to the "GX Developer Operating Manual".

📕 Local variabl	e setting[MAIN] *				_ 🗆 ×
Register Close	Edit operation	Display global ve			<u> </u>
A	Label	Constant	Device type	Comment	
	itialization		▼		
					▼ \\
			$\overline{\mathbf{n}}$		

2) Enter a device type.

Enter it directly or make selection from the list box.

Local variable setting[MAIN] *		
	play global variables	
	stant Device type	Comment
Imitialization	BOOL INT DINT REAL STRING ARRAY	
	\bigcup	

3) Enter a comment into the label.

Enter it within 64 characters.

The comment can be displayed in the tool tip format of the label information.

* For the label information, refer to "CHAPTER 7 INTRODUCTION TO USEFUL FUNCTIONS FOR ST PROGRAM EDITING" or "GX Developer Operating Manual (Structured Text)".

🔚 Local variable setting[MAIN] >	ĸ			
Register Edit operation Close Insert Add	Display global v			<u> </u>
Au Label	Constant	Device type		Comment
1 Initialization		BOOL	 The process is initialized 	
4				 ▶ //
		$\overline{\mathbf{n}}$	To next page	

From previous page

4) When entering labels continuously, click the Insert or Add button under Edit operation to add a line. The buttons under Edit operation have the following functions.

Insert button ... Inserts a blank line into the current cell position.

Add button Inserts a blank line into the place one line below the current cell position.

Delete button ... Deletes one line in the current cell position.

E	Local v	aria	ble setting[MAIN] >	k				_ 🗆 🗙
	Regist Close		Edit operation	Show assigned				<u>*</u>
	1	Au	Label	Constant	De	vice type		Comment
	1		Initialization		BOOL	•	The process is initialized	
	2					•)
								_1
	(
┛								
					-			

5) After input is complete, click the Register button.

🔚 Local variable setting	[MAIN] *		
Register	eration t Add Display global v Delete Show assigned		<u>*</u>
Au Lat	el Constant	Device type	Comment
1 Initialization	1	BOOL	The process is initialized
2 Defective			Total of number of defective goods
3 good			Total of number of quality items
4 Yield			Yield
5 Inspection		BOOL	Inspection flag
•			
		7	
	SOFT Series GX Deve		Click the OK button.
Γ.	Check the redefine end	ror of the label.	
	The time period for che with increase in variable		
	ОК		
	\square	To next page	



Registration is completed.

Click the OK button.

When registration is made, "*" displayed on the title bar disappears.

R	cal v Regis	ster	able setting[MAIN] Edit operation Insert Add Delete	B Display global s			× ▲
		Au	Label	Constant	Device type		Comment
1	1		Initialization		BOOL	-	The process is initialized
2	2		Defective		DINT	•	Total of number of defective goods
3	3		good		DINT	-	Total of number of quality items
4	1		Yield		REAL	-	Yield
5	5		Inspection		BOOL	-	Inspection flag
•							▼ ▶ //

REMARK

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

Entering a program

What should be noted during input?	A program can be input freely in text format using the ST edit screen. Note the following points during input.
	 Use a space, Tab key or Enter key to enter a blank. When the defined label, control syntax or comment is input, the character color changes. If it does not change, the possible cause is an input mistake or undefined label.

Now, actually input a program in List-1.

List-1.

```
IF Initialization THEN
good := 0; Defective := 0; Yield := 0.0;
ELSE
IF Inspection THEN
good := good +1;
ELSE
ELSE
Defective := Defective + 1;
END_IF;
Yield := DINT_TO_REAL(good)/DINT_TO_REAL(good + Defective);
END_IF;
```

Displaying the ST edit screen



Entering a label

To enter a label, a label name may be entered directly or the label selection function be used. To use the label selection function, labels must have been entered in advance.

Here, the input method using the label selection function will be explained.



Entering a function

Enter a function in upper case. To enter, a function may be entered directly or the function selection function be used.

Here, the input method using the function selection function will be explained.



From previous page	
ST MAIN 25Row *****Step *	4) The function name is inserted.
<pre>IF Initialization THEN good := 0; Defective := 0; Yield := 0.0; ELSE</pre>	* The function argument type is displayed in the tool tip format.
IF Inspection THEN good := good + 1;	
ELSE Defective := Defective + 1; Function argument type is displayed.	
4) Inserted.	
$\overline{\Box}$	
ST MAIN 25Row *****Step *	5) Refer to the function argument type displayed in the tool tip format, and
ELSE IF Inspection THEN	enter the argument to complete the entry.
good := good + 1;	
ELSE Defective := Defective + 1; END_IF;	
<pre>Yield:=DINT_T0_REAL(good)/DINT_T0_REAL(good + Defective); END_IF;</pre>	
5) Complete	

Entering a comment

Comments do not affect the program behavior at all. When program processings are described as comments, they give the at-a-glance pictures of the processings.

First, enter a comment on Line 1 of the program.

ST MAIN 26Row ******Step *	٦×۱
(* Initialization is done. *)	
<pre>IF Initialization THEN good := 0; Defective := 0; Yield := 0.0;</pre>	
ELSE	
IF Inspection THEN	
good := good + 1;	
ELSE	
Defective := Defective + 1; END_IF;	
	•
To next page	
 For home page 	

* Enter a comment by enclosing it with "(*" that represents the beginning of the comment and "*)" that represents an end.



Refer to the example given on the left and enter comments. (List-2)

This completes program input.

Display of label information The label information can be displayed in the tool tip format. Operation : Place the mouse pointer in the label position. Display : Label name -> Label type -> Label comment -> Device *			
*: The device is displayed after convert (compile) is performed.			
ST MAIN 25Row 143Step			
(* Initialization is done. *)			
<pre>IF Initialization THEN good := 0; Defective := 0; Yield := 0.0;</pre>			
(* The normal operation is processed. *)			
ELSE (* Is the inspection passing? *) IF Inspection THEN Label information is displayed.			
<pre>good := good + 1; (* The number of non-defective articles is</pre>			
ELSE			
 Defective := Defective + 1; (* The number of defective goods is add Change of display color 			
Control syntax, comment and label character string colors, ST edit screen			
background color, etc. can be changed.			
Operation: Choose [Tools] \rightarrow [Change display color] in the menu.			
 Setting of auto indent Indentation at the time when the Enter key is pressed and the Tab width at the time when the Tab key is pressed can be set. Operation: Choose [Tools] → [ST editor settings] in the menu. 			
For details, refer to the "GX Developer Operating Manual (Structured Text)".			

Converting (compiling) the ST program

What is convert (compile)?

Changing the program created on the ST edit screen into a sequence program that can be executed by the PLC CPU is called convert (compile).

Performing convert (compile)

The convert (compile) operation method will be explained using the created program.

Convert	View	Online	Diagnostics	Tools	Window	Help ·
Conve	rt/Com	pile			F4	
Conve	rt/Com	pile (All p	rograms being	; edited)	∖ ∕سانم ا	1 E4
Conve	rt/Com	pile (All p	rograms)			Click.
Conve	rt/Com	pile (Onlii	ne change)		5	

 Click [Convert] → [Convert/Compile] in the menu.

(1) At normal completion

The following message is displayed.

MELSOFT series GX Developer	Convert (compile) is completed.
Compile Completed. Local label is re-allocated. Write the program to PLC or Compare with the program in PLC before execute RUN write function and change TC variable for PLC. The comment of label will overwrite on device comment. All right? Yes No	Since the confirmation screen shown on the left is displayed, click the No button. If the Yes button is clicked, the "Comment data to be referred to (comment by program) does not exist" message may be displayed.
<pre>If MAIN 25Row 143Step (* Initialization THEN good := 0; Defective := 0; Yield := 0.0; (* The normal operation is processed. *) ELSE (* Is the inspection passing? *) IF Inspection THEN good := good + 1; (* The number of non-defective articles is added. *) (* The inspection is failing. *) ELSE Defective := Defective + 1; (* The number of defective goods is added. *) KID_IF; (* The yield is calculated. *) Yield:=DINT_TO_FEAL(good)/DINT_TO_FEAL(good + Defective); END_IF;</pre>	* At normal completion of convert (compile), the number of steps is displayed on the title bar.

(2) When error occurs

The Compile error (Detail) dialog is displayed.

Now, actually see the debugging operation at occurrence of a compile error.

1) Change the program so that a compile error will occur.



2) Perform convert (compile).



Change Line 3 in List-2.

RYOUHIN := 0; \rightarrow RYOUHIN : = 0. 0;

Click [Convert] \rightarrow [Convert/Compile] in the menu.

3) A compile error occurs and the dialog is displayed.



Confirm the error step/line and error definition.

4) Confirm the line on which the error has occurred.



- 1) Select the error definition with the mouse.
- 2) Click the Jump button.



5) Track down the cause and correct the faulty part.The error indication mark is displayed on the ST edit screen.Confirm the error definition and program contents, and correct the program.

STMAIN 25Row ******Step *
<pre>tF Initialization THEN good := 0.0; good := 0.0; () The normal operation is processed. *)</pre>
Error indication mark
<pre>good := good + 1; (* The number of non-defective articles is added. *)</pre>
(* The inspection is failing. *)
ELSE Defective := Defective + 1; (* The number of defective goods is added. *) END_IF;
(* The yield is calculated. *)
Yield:=DINT_T0_REAL(good)/DINT_T0_REAL(good + Defective);
END_IF;

Correct Line 3 in List-2.

good : = 0. 0; \rightarrow good := 0;

Click [Convert] \rightarrow [Convert/Compile] in the menu.

Point

The error location and actually corrected part may be different. Identify the faulty part from the error definition displayed in the "Compile error (Detail)" dialog and the program contents of the line where the error indication mark is displayed. Chapter 4 explains the procedure to write the converted (compiled) sequence program to the PLC CPU and the procedure to read the sequence program from the PLC CPU.

Performing write to PLC

The operation method for write to PLC will be explained.

Display the Write to PLC dialog and write the program and parameters to the PLC CPU.

Online Diagnostics Tools Window Help Transfer setup 1) Click. Read from PLC 1) Click. Write to PLC 1) Verify with PLC Write to PLC(Flash ROM) Image: Comparison of the plane in the pla	 * When performing write to PLC, put the PLC CPU in a STOP status. 1) Click [Online] → [Write to PLC] in the menu.
Write to PLC X Connecting interface CDM1 PLC Construction ation No. Host PLC type 4) Click. PLC Conserved ation No. Host PLC type 4) Click. Target m 3) Click. emory Title File selection cerustar morgiam connon Local Execute Label program (ST FB.Structure) Target memory Program memory/Device memory Password setup Related functions Transfer setup Keyword setup MAIN1 MAIN2 Keyword setup Parameter COMMENT Parameter Plc/Network/Remote password Clear PLC memory Clear PLC memory Vhole range Select. egister Grane specification Whole range Total free space Bytes	 2) Select the "Label program (ST, FB, structure)" check button in the <<file selection="">> tab.</file> * When the check button is not selected, only the actual program is written. 3) Click "Param + Prog". 4) Click the Execute button.
MELSOFT series GX Developer X Completed.	 * Reset the PLC CPU and put it in a RUN status. If an error occurs, choose [Diagnostics] → [PLC diagnostics] in the menu of GX Developer, and confirm the error definition.

4

Performing read from PLC

The operation method for read from PLC will be explained.

Display the Read from PLC dialog and read the program and parameters from the PLC CPU.

Online Diagnostics T Transfer setup 1) Click. Read from PLC 1) Write to PLC 1) Write to PLC 1) Write to PLC 1) Delete PLC data 1)	1) Click [Online] - [Read from PLC] in the menu.
Read from PLC XI Connecting interface COM1 PLC Comparing interface COM1 PLC Comparing interface COM1 Variable in the interface Common No. Host PLC Comparing interface Close	 2) Click "Param + Prog" in the <<file selection="">> tab.</file> 3) Click the [Execute] button.
Program (ST,FB,Structure) 03/01/07 13.16 Parameter 02/12/25 14 Device memory 02/12/25 14 Device data 02/12/25 14 File register Remote operation Clear PLC memory File register Refresh view Range specification 2R Total free space Bytes	
MELSOFT series GX Developer	If an error occurs, choose [Diagnostics] - [PLC diagnostics] in the menu of GX Developer, and confirm the error definition.

Chapter 5 explains the online debugging operation of the sequence program written to the PLC CPU using the monitor function and device test function.

- Monitoring the sequence program.
- Changing the bit device value and conducting a device test.
- Changing part of the sequence program and writing it to the PLC CPU in RUN status.

5.1 Monitoring the Sequence Program

This section explains the operation method to monitor the sequence program.



The labels displayed on the ST edit screen are displayed on the same lines of the monitor screen.

5

5.2 Device Test

The value of the label (bit device/word device) in the PLC CPU can be changed directly. Here, the bit device value is changed to confirm the program behavior.

Confirming the program behavior

The operation to change the bit device value will be explained.



 Click [Online] → [Debug] → [Device test] in the menu.

Forcibly turn ON the label "Inspection" that represents the bit device.

Device test Bit device 2) Enter "Inspection".	×			
Device	Close			
FORCE ON FORCE OFF Toggle force	Hide history			
Wc 3) Click.				
© Device	•			
Buffer memory Module start I/0 (Hex)				
Address HEX	-			
Setting value DEC 16 bit integer Set				
Program Label reference program MAIN				
Execution history				
Device Setting condition Program name	Find			
Inspection Force ON MAIN	Find next			
	Re-setting			
	Clear			

ablaTo next page

- 2) Input " Inspection" into the bit device.
- 3) Click the FORCE ON button.

From previous page

Make confirmation on the monitor screen.

🔳 ST	(Monitoring) MAIN 25Row 143Step	
(*	Initialization is done. *)	
IF	Initialization THEN good := 0; Defective := 0; Yield := 0.0;	Initialization = 1 good = 503493, Defective = 1, Yield =
(*	The normal operation is processed. *)	
ELS	E (* Is the inspection passing? *)	ON
	IF Inspection THEN	Inspection = 1
	good := good + 1; (* The number of non-defective a	good = 503493 "good" value increases.
	(* The inspection is failing. *)	
	ELSE Defective := Defective + 1; (* The number of defective go END_IF;	Defective = 1
	(* The yield is calculated. *)	
	<pre>Yield:=DINT_T0_REAL(good)/DINT_T0_REAL(good + Defective);</pre>	Yield = 1.000, good = 503493, Defective =
ENI		

Also change the other label values and confirm the program behavior.

Point
GX Developer supports the following debug functions for the programs created in
ST language.
 Break execution : debugs programs by halting the program execution at the location specified by break point.
• 1 line execution : debugs programs by halting the program execution line-by-line.
For details, refer to the relevant section in GX Developer Operating Manual (Structured Text).

5.3 Online Change

When the PLC CPU is in a RUN status, part of the sequence program can be changed. This is called online change.

Actually change part of the sequence program and perform online change.

Changing part of the program and performing online change Change the calculation expression of "Yield" and perform online change.

1) Change part of the program.

Yield := DINT_TO_REAL (good)/DINT_TO_REAL (good + Defective); ↓ Yield := (DINT_TO_REAL (good)/DINT_TO_REAL (good + Defective))*100.0;

🗖 ST	(Monitoring) MAIN 25Row 148Step	
(*	Initialization is done. *)	
IF	Initialization THEN good := 0; Defective := 0; Yield := 0.0;	Initialization = 0 good = 2919230, Defective = 1834337, Yield :
(*	The normal operation is processed. *)	
ELS		
	(* Is the inspection passing? *)	
	IF Inspection THEN	Inspection = 1
	good := good + 1; (* The number of non-defective articl	good = 2919230
	(* The inspection is failing. *)	
	ELSE	
	Defective := Defective + 1; (* The number of defective goods i END_IF;	Defective = 1834337
	(* The yield is calculated. *)	
(*)	Yield:=(DINT_T0_REAL(good)/DINT_T0_REAL(good + Defective))*100.0;	Yield = 0.614, good = 2919230, Defective =
END	_IF; Changed.	

"*" indicating the line to be online changed is displayed on the indicator bar of the target line.

2) Execute online change.



Click [Convert] \rightarrow [Convert/Compile (Online change)] in the menu.

From previous page

3) The confirmation message is displayed.

MELSOFT series GX Developer 🔀	Click the Yes button.
Caution! PLC control has changed. Make sure everything is safe then execute again. The write destination is the program in the program memory.	
Don't write to the same program from a plurality of place at the same time. Ensure the PLC program and the program to be converted match.	
OK?	
Write destination program: MAIN	
Yes No	
To next page	


"*" that indicates the online change target line disappears.

On the monitor screen, confirm that the present value of "Yield" has changed.

MEMO

In Chapter 6, the completed project is saved with a name.

Saving the project

Save the created project with a name.



The program created this time was saved as described below. Drive/Path : C:\MELSEC Project name : SAMPLE_ST Title :work-check

This completes a series of operations from the creation of the new ST project to the input and online debugging of the program to the storage of the project. Fully understand the operations performed until now, and proceed to Chapter 8.

MEMO

This chapter introduces useful functions for editing ST programs. For more information, refer to the "GX Developer Operating Manual (Structured Text)".

(1) Window division

Every time it is desired to confirm the contents midway through editing of a large program, it is troublesome to scroll the screen to see the program ...

At such a time, use "Window division".

Choosing [Window] \rightarrow [Divide into two] in the menu displays the screen in vertically divided windows. The divided windows can be scrolled/edited individually.

(2) Bookmark

When it is desired to jump to a specific line, it is troublesome to search the program from the beginning ... At such a time, use "Bookmark".

Preset the bookmark by choosing [Find/Replace] \rightarrow [Bookmark setting/release] or [Find/Replace] \rightarrow [Find] \rightarrow "Set bookmark" in the menu.

By choosing [Find/Replace] \rightarrow [Bookmark list] in the menu, any line can be selected from the Bookmark list dialog to make a jump to that line.

(3) Display of label information

It is desired to know the device assigned to the label ...

At such a time, use "Label information".

When the mouse pointer is placed on the label,

Label name -> Label type -> Label comment -> Device

is displayed in the tool tip format for at-a-glance confirmation of the contents.

Note: Applicable to converted (compiled) programs.

Confirmation can also be made by activating "Show assigned device" on the Local variables setting screen.

(4) Select function

It is desired to input the function whose name has been forgotten ...

At such a time, use "Select function".

Choosing [Edit] \rightarrow [Select function] displays the Select function dialog to allow the function name to be selected. Also, since the function argument type is displayed in the tool tip format when the function is inserted, the argument can be input with reference to that argument type.

(5) Change of display color and font

It is difficult to differentiate between the character strings on the edit screen since they have the same color, or it is desired to change the character size ...

At such times, use "Change display color" or "Font".

Choosing [Tools] \rightarrow [Change display color] in the menu displays the Change display color dialog to allow the comment, control syntax, character, label and background colors to be selected. Changing the display color improves readability.

Choosing [Tools] \rightarrow [Font] in the menu displays the Font dialog to allow the font type, style and size to be selected. Making easy-to-use setting improves operability.

MEMO

Chapters 1 to 7 explained the basic operation methods and functions for creating ST programs. In Chapter 8, create a function block (FB) in ST language and paste it to a ladder program to create a program to be used.

The main items to be explained in this chapter are as follows.

- Adding a new FB.
- Defining FB variables.
- Creating an ST program.
- Creating an FB in ST language.
- Using the ST-written FB in a main program.

8.1 Creating an FB

What is a function block (FB)? An FB, where often used processings are predefined as parts, can be used in the necessary area of each program.

Creating a new project

Creating a new project

The creation method of a new project to create a main program in ladder format will be explained.

Click [Project] \rightarrow [New project] in the menu to display the New project dialog.

New Project			×	1) Enter as follows.	
PLC series				 PLC series 	:QCPU (Q mode)
QCPU(Qmod	le)	T		 PLC type 	:Q02(H)
				 Label setting 	:Use label
PLC Type			2) Click.	Program type	:Ladder
Q02(H)		Ľ		0 71	
Program type		Label setting	r I I	2) Click the OK bu	itton.
Ladder	_	C Donot © Use lab		,	
	MELSAP-L		using ST program,		
C ST	٨	FB and strug			
1) Set		ime as program (data's name is created.		
Se 1) Set	tilese.				
🗖 Setup proje	ect name				
Drive/Path	C:\MELSEC				
Project name			Browse		
Title [
I					
	~	То Г	next page		

8

8 ST PROGRAMMING APPLICATION (PASTING FB TO LADDER PROGRAM)

MELSOFT

<i>From previous page</i>	
#MELSOFT series GX Developer (Unset project) - [LD(Edit mode) MAIN 124 Step] X Project Edit End/Replace Convert Yew Online Diagnostics Look Window Help X D P Point Edit End/Replace Convert Yew Online Diagnostics Look Window Help X D P Point Edit End/Replace Convert Yew Online Diagnostics Look Window Help X D P Point Edit End/Replace Convert Yew Online Diagnostics Look Window Help X D P Point Edit End/Replace Convert Yew Online Diagnostics Look Window Help X D P Point Edit End/Replace Convert Yew Online Diagnostics Look Window Help X D P Point Edit Series X D P Point Edit Vision Diagnostics X D P Point Edit Series X D P Point Edit Vision Diagnostics X D P Point Edit Vision Dia	3) A new project is created.
Image: Second state and s	
Adding a new FB	

Adding an FB

The operation method to add a new ST-written FB will be explained.



1) Right-click "Function Block" in the <<FB>> tab to display the menu.

8



8 ST PROGRAMMING APPLICATION (PASTING FB TO LADDER PROGRAM)

MELSOFT



6) A new data name: CALCULAT is added.

Defining FB variables

The labels used in the FB are called FB variables (FB labels).

■ Displaying the FB variable (FB labels) setting screen

Here, the operation method to define FB variables (FB labels) will be explained.



1) Double-click "Header" in the <<FB>> tab.

2) The FB variable (FB labels) setting screen is displayed.

Setting the FB variables (FB labels)

1) Select the Input/Output type.

Select the label type. There are the following four types.

- · VAR_INPUT Variable input from FB outside
- · VAR_OUTPUT Variable output to FB outside
- · VAR_IN_OUT...... Variable having the input and output functions
- "Blank" Variable used in FB inside

FB variable setting[CALCUL	_AT] The yield is ca	lculated*		
Register Edit operation	Add			
Input/Output	Label	Constant	Device type	Com 🔺
VAR INPUT		1	¥.	

- 2) Enter the label name.
 - Enter the label name within 16 characters.

FB variable setting[CALCULAT] The yield is calcula	ted∗		
Register Edit operation Insert Add Close Delete			
Input/Output Label	Constant	Device type	Com 📥
		▼	•
•			•
		To next page	



3) Enter the device type.

Enter it directly or make selection from the list box.

FB variable setting[CALCULAT] The yield is ca	lculated*		
Register Edit operation Insert Add Close Delete			
Input/Output Label	Constant	Device type	Com_
1 VAR INPUT GOOD IN			•
			L L
]	
4) Enter a comment into the label.			
Set it within 64 characters.			
FB variable setting[CALCULAT] The yield is calcul	ated*		
Register Edit operation Add			<u> </u>

	Input/Output	Label	Constant	Device type	Cor	mment
1 VA	R INPUT 🔻	GOOD IN		DINT	Number of quality items	
						•
•						► //.

5) After input is complete, click the Register button.

FB variable setting[CALCULAT] The view	eld is calculated*			<u>_ 🗆 ×</u>
Edit operation Register Lett Add Close Delete				^
Input/Output Labe	I Constant	Device type		Comment
1 VAR INPUT - GOOD IN		DINT	Number of guality items	
2 VAR_INPUT ▼ DEFECTIVE	IN	DINT 🔻	Number of defective goods	
3 VAR_OUTPUT VIELD_OUT		REAL 🔻	The yield is calculated	
				▼ ▶ //
	~	」 └ ──── To next pag	e	

From previous pag	e
MELSOFT Series GX Developer	Click the OK button.
Check the redefine error of the label.	
The time period for checking gets longer with increase in variables.	
OK	
MELSOFT series GX Developer 🛛 🗙	Registration is completed. Click the OK button.
Registration completed.	
OK	
	ļ

When registration is made, "*" displayed on the title bar disappears.

FB variable setting[CALCU Edit operation Insert Close	Add	culated			<u> </u>
Input/Output	Label	Constant	Device type		Comment
1 VAR INPUT	GOOD IN		DINT	Number of quality items	
2 VAR_INPUT -	DEFECTIVE_IN		DINT	Number of defective goods	
3 VAR_OUTPUT -	YIELD_OUT		REAL 🗸	The yield is calculated	
					 ▼ ∦

REMARK

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

Creating an FB in ST language

The operation to input the program in List-3 will be explained.

List-3

YIELD_OUT := DINT_TO_REAL(GOOD_IN)/DINT_TO_REAL(GOON_IN + DEFECTIVE_IN);

Displaying the FB definition screen



Inputting the program

Input the FB program body as in the method of inputting the main program described in Chapter 3.

Input the program in List-3.



Converting (compiling) the FB

Click [Convert] \rightarrow [Convert/Compile] in the menu to perform convert (compile).



Compile processing is completed. Click the OK button.

At normal completion of convert (compile), the number of steps is displayed on the title bar.

FB ST CALCULAT The yield is calculated 1	Row (15)Step
YIELD_OUT := DINT_TO_REAL(GOOD_IN)/DINT_T	O_REAL(GOOD_IN + DEFECTIVE_IN);

8.2 Pasting the FB to a Main Program

Create a main program (ladder) using the FB created in Section 8.1.

Defining the local variables

Define the labels used in the main program.

Displaying the Local variables setting screen



1) Double-click "Header".

2) Local variable setting screen is displayed.

Setting the local variables (headers)

Refer to Chapter 3 and make setting as follows.

	ocal Regi Clo	ster	ble setting[MAIN]		l device		
		Au	Label	Constant	Device type	Comment	▲
	1		Initialization			The process is initialized	
	2		Defective		DINT	 Total of number of defective goods 	
	3		good		DINT	 Total of number of quality items 	
	4		Yield		REAL	▼ Yield	
	5		Inspection		BOOL	 Inspection flag 	-
•							•
				\bigcup	7		



After input is complete, click the Register button. The registration of the local variables is completed. Click the OK button.

When registration is made, "*" displayed on the title bar disappears.

Local	varia	ble setting[MAIN]			
Clos		Edit operation	Display global v		<u> </u>
	Au	Label	Constant	Device type	Comment 🔺
1		Initialization		BOOL	The process is initialized
2		Defective		DINT 🗸	Total of number of defective goods
3		good		DINT 🗸	Total of number of quality items
4		Yield		REAL 🗸	Yield
5		Inspection		BOOL	Inspection flag
					_
•					

Creating a main program



Display the main program edit screen and input the following program (List-4).

Displaying the edit screen

Double-click "Body" in the <<Project>> tab to display the edit screen.





Inputting the program in ladder format

Refer to the following diagram and input the program.

; <u>X</u>]	LD(Edit mode) MAIN (35) Step *			_ 🗆 ×
⊡- (Unset project) ⊕-⊞ Global variables ⊕- Program	Initialization	-[DMOV	K0 good	}
🖻 📾 MAIN		K0	Defective)
		-EMOV	EO Yield)
- B Device memory - B Device init		[RST	Inspection)
	Initialization Inspection	[D+	Kl good)
	[Inspection	Kl	Defective)
	(0)		[END	}
Project FB Structure				•

Pasting the FB

Switch to the <<FB>> tab, and drag and drop the FB program to the target place.



The FB is inserted into the main program.



8 ST PROGRAMMING APPLICATION (PASTING FB TO LADDER PROGRAM)

MELSOFT



Inputting the input ladder section and output ladder section

Refer to the following diagram, and input the input ladder section and output ladder section.



Performing convert (compile)

Click [Convert] \rightarrow [Convert/Compile] in the menu to perform convert (compile).

MELSOF	T series GX Developer
A	Compile Completed.
	Local label is re-allocated. Write the program to PLC or Compare with the program in PLC before execute RUN write function and change TC variable for PLC.
	The comment of label will overwrite on device comment. All right?
	Yes No

Convert (compile) is completed. Click the No button.

When convert (compile) is completed, the number of steps is displayed on the title bar.



<u>8.3 Online</u>

Write the sequence program to the PLC CPU, and confirm the program behavior using the monitor function and device test function.

Writing to PLC CPU

Performing write to PLC

Refer to Chapter 4 and perform write to PLC.

Choose [Online] \rightarrow [Write to PLC] in the menu to display the Write to PLC dialog.

Write to PLC Connecting interface COM1 PLC Connection Network No. PLC Connection Network No. Target memory Program memory/Device memory Target memory Title File selection Device data Param+Prog Select all Cancel all selections Target memory Param+Prog Select all Cancel all selections Target memory Program Target memory Program Target memory Program Target memory Program 2) Click. Perameter PLC/Network/Remote password 1) Select. Out- 1000 erange © Range specification 2R	X 3) Click. Execute Close Password setup Related functions Transfer setup Keyword setup Keyword setup Remote operation Clear PLC memory Format PLC memory Arrange PLC memory Create title	 * When performing write to PLC, put the PLC CPU in a STOP status. 1) Choose the "Label program (ST, FB, structure)" check button in the <<file selection="">> tab.</file> * If the check button is not chosen, only the actual program is written. 2) Click "Param + Prog". 3) Click the Execute button.
Free space volume	Bytes	
MELSOFT series GX Developer		3) Write to PLC is completed.
Completed.		* Reset the PLC CPU and put it in a RUN status.
OK		If an error occurs, choose [Diagnostics] \rightarrow [PLC diagnostics] in the menu of GX Developer, and confirm the error

definition.

Monitoring the sequence program

Monitor and confirm the sequence program.

The monitor start/stop operation is as follows.

- When starting monitor
 [Online] → [Monitor] → [Monitor mode]
 When stopping monitor
- $[Online] \rightarrow [Monitor] \rightarrow [Stop monitor]$
- When resuming monitor
 [Online] → [Monitor] → [Start monitor]

Monitor display



The comments set on the Local variables setting screen can be displayed by choosing [View] \rightarrow [Comment] in the menu.

Confirming the program behavior

Change the value of the bit device in the PLC CPU and confirm the program behavior.

Conducting a device test

Refer to Section 5.2 and change the value of the bit device.

Choose [Online] \rightarrow [Debug] \rightarrow [Device test] in the menu to display the Device test dialog.

Device test Bit device - 1) Input "Inspection".	×
Device	Close
Inspection -	
FORCE ON FORCE OFF Toggle force	Hide history
Word 2) Click.	
© Device	•
C Buffer memory Module start I/O (Hex)	
Address HEX	-
Setting value	
DEC 💌 16 bit integer	▼ Set
Program-	
Label reference program MAIN	-
Execution history	
Device Setting condition Program name	Find
Inspection Force ON MAIN	Find next
	Re-setting
	Clear

- 1) Input "Inspection" into the bit device.
- 2) Click the FORCE ON button.

- Check that the program is running correctly.
- * Also change the other label values and confirm the program behavior.



Saving the project

Refer to Section 6 and save the created project with a name.

Click [Project] \rightarrow [Save as] in the menu to display the Save the project with a new name dialog.

Save the project with a new name	1) Enter as follows. • Drive/Path : C:\MELSEC • Project name : SAMPLE_FBST • Title : FB is used and calculated.
Gppw 2) Click. Drive/Path C:\MELSEC Project name SAMPLE_FBST Title FB is used and calculated.	2) Click the Save button.
1) Set these.	
The program created this time was saved as described below.Drive/Path: C:\MELSECProject name: SAMPLE_FBSTTitle: FB is used and calculated	

This ends the explanation of a series of operation methods for program creation.

To further proceed to the next step, it is recommended to refer to the manuals given in the section "Relevant Manuals".

MEMO

INDEX

[A]

Adding an FB	. 8- 2
Alarm history display function	6-16
Alarm list display function	
System alarm	6-19
User alarm	6-18

[C] C

C	onvert (compile)	
	Converting (compiling) the FB	8-10
	Error indication mark	3-16
	Performing convert (compile)	3-14
	The compile error (Detail)	3-15
,	What is convert (compile)	3-14

[D]

T - - 4

Device l'est
Confirming the program behavior 5- 2, 8-18
Displaying the FB definition screen 8-9
Inputting the program8-10
Displaying the FB variable screen 8-5
Displaying the ST edit screen 3-9

[E]

E	diting
	Bookmark7-1
	Change of font 7-1
	Change of display font 7-1
	Display of label information 7-1
	Select function
	Window division

[L] Label

Label	
What does defining the labels mean	3- 3
Local variable	
Displaying the setting screen3-	3, 8-11
Enter a comment into the label	3- 5
Enter a device type	3- 5
Enter a label name	3- 5
Setting the local variables3-	5, 8-12

[M]
Monitor
Monitoring the Sequence Program
5- 1, 8-17
[0]
Online change5- 4
[P]
Pasting the FB8-14
Performing read from PLC4- 2
Performing write to PLC 4- 1, 8-16
Program
Entering a comment3-12
Entering a function3-11
Entering a label3-10
Entering the characters
Function argument3-12
Function selection function
Inputting the input ladder section and output
ladder section 8-15
Inputting the program in ladder format8-14
What should be noted during input3-8
Project
Creating a new project8-1
Creating a new ST project3- 1
Saving the project 6- 1, 8-19
(C)
[S]
Enter a comment into the label
Enter the device type
Enter the label name8-6

[W]

What is a function block (FB)		
What is a function block (FB)	8-	1
What is the ST language	1-	1

Ind

MEMO			

Structured Text (ST)

Programming Guide Book

MODEL ST-GUIDE-EL

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212, JAPAN NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

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