Chapter 1 EasyBuilder Pro Installation and Startup Guide	10
1.1 EasyBuilder Pro Installation	10
1.2 Steps to Install EasyBuilder Pro	11
Chapter 2 Project Manager Operations	17
2.1 HMI IP, Password	18
2.2 Utility	20
2.2.1 Steps to Download Project via USB or CF Card	21
2.3 Transfer	22
2.3.1 Download	22
2.3.2 Upload	23
2.4 Simulation	25
2.4.1 On-line Simulation/Off-line Simulation	25
2.5 Pass-Through	27
2.5.1 Ethernet	27
2.5.2 COM port	28
Chapter 3 Create an EasyBuilder8000 Project	30
3.1 Create a New Project	30
3.2 Save and Compile the Project	34
3.3 Off-line and On-line Simulation	35
3.3.1 Off-line Simulation	35
3.3.2 On-line Simulation	36
3.4 Download the Project to HMI	37
Chapter 4 Hardware Settings	42
4.1 I/O Ports of HMI	42
4.1.1 USB Port	42
4.1.2 Ethernet Port	42
4.1.3 CF Card or SD Card	42
4.1.4 Serial I/O Port	42
4.2 HMI System Settings	43
4.2.1 System Reset	43
4.2.2 System Toolbar	45
4.2.2.1 Large Keyboard	46
4.2.2.2 Small Keyboard	47
4.2.2.3 System Information	47
4.2.2.4 System Setting	48
4.2.2.5 Touch Screen Calibration Mode	57

4.3 HMI Download Settings	
Chapter 5 System Parameter Settings	61
5.1 Device	
5.1.1 How to Control a Local PLC	63
5.1.2 How to Control a Remote PLC	
11B5.1.3 How to Control a Remote HMI	70
5.2 Model	
5.3 General	
5.4 System Setting	
5.5 Security	
5.6 Font	
5.7 Extended Memory	
5.8 Printer/Backup Server	
Chapter 6 Window Operations	
6.1 Window Types	
6.1.1 Base Window	
6.1.2 Common Window	
6.1.3 Fast Selection Window	
6.1.4 System Message Window	
6.2 Create, Set, and Delete a Window	
6.2.1 Create a Window	
6.2.2 Window Settings	
6.2.3 Open, Close and Delete a Window	
Chapter 7 Event Log	
7.1 Event Log Management	
7.1.1 Excel Editing	
7.2 Create a New Event Log	
Chapter 8 Data Sampling	
8 1 Data Sampling Management	110
8 2 Create a New Data Sampling	
Chapter 9 Object General Properties	
9.1 Selecting PLC	
9.1.1 Setting the Reading and Writing Address	
9.2 Using Shape Library and Picture Library	
9.2.1 Settings of Shape Library	

9.2.2 Settings of Picture Library	132
9.3 Setting Text Content	135
9.4 Adjusting Profile Size	140
9.5 Variables of Station Number	141
9.6 Broadcast Station Number	143
Chapter 10 Security	144
10.1 Settings of Password and Classes	144
10.2 Security of Objects	146
10.3 Examples of Security	150
Chapter 11 Index Register	155
11.1 Introduction	155
11.2 Examples of Index Register	156
Chapter 12 Keyboard Design and Usage	161
12.1 Steps to Design a Pop-up Keypad	161
12.2 Steps to Design a Keyboard with Direct Window	165
12.3 Steps to Design a Fixed Keyboard	168
12.4 Creating UNICODE Keyboard	169
Chapter 13 Objects	171
13.1 Bit Lamp	171
13.2 Word Lamp	174
13.3 Set Bit	179
13.4 Set Word	183
13.5 Function Key	191
13.6 Toggle Switch	198
13.7 Multi-State Switch	201
13.8 Slider	205
13.9 Numeric Input and Numeric Display	209
13.10 ASCII Input and ASCII Display	221
13.11 Indirect Window	226
13.12 Direct Window	231
	201
13.13 Moving Shape	235
13.13 Moving Shape 13.14 Animation	235 241
13.13 Moving Snape 13.14 Animation 13.15 Bar Graph	235 241 246
13.13 Moving Snape 13.14 Animation 13.15 Bar Graph 13.16 Meter Display	235 241 246 254
13.13 Moving Snape 13.14 Animation 13.15 Bar Graph 13.16 Meter Display 13.17 Trend Display	235 241 246 254 262

13.19 Data Block Display	
13.20 XY Plot	
13.21 Alarm Bar and Alarm Display	
13.22 Event Display	
13.23 Data Transfer (Trigger-based)	
13.24 Backup	
13.25 Media Player	
13.26 Data Transfer (Time-based)	
13.27 PLC Control	
13.28 Schedule	
13.29 Option List	
13.30 Timer	
13.31 Video In	
13.32 System Message	
Chapter 14 Shape Library and Picture Library	391
14.1 Creating Shape Library	391
14.2 Creating Picture Library	
Chapter 15 Label Library and Multi-Language Usage	405
15.1 Introduction	405
15.2 Settings of Font of Label Library	
15.3 How to Create a Label Library	
15.4 Using Label Library	410
15.5 Settings of Multi-Language	411
Chapter 16 Address Tag Library	414
16.1 Creating Address Tag Library	
16.2 Using Address Tag Library	
Chapter 17 Transferring Recipe Data	418
17.1 Updating Recipe Data with Ethernet or USB cable	
17.2 Updating Recipe Data with SD Card or USB Flash Drive	
17.3 Transferring Recipe Data	
17.4 Saving Recipe Data Automatically	
Chapter 18 Macro Reference	
18.1 Instructions to the Macro Editor	422
18.2 Macro Construction	
18.3 Svntax	

18.3.1 Constants and Variables	432
18.3.1.1Constants	432
18.3.1.2 Variables	432
18.3.2 Operators	435
18.4 Statement	438
18.4.1 Definition Statement	438
18.4.2 Assignment Statement	438
18.4.3 Logical Statements	439
18.4.4 Selective Statements	440
18.4.5 Reiterative Statements	442
18.4.5.1 for-next Statements	442
18.4.5.2 while-wend Statements	444
18.4.5.3 Other Control Commands	444
18.5 Function Blocks	446
18.6 Build-In Function Block	449
18.6.1 Mathematical Functions	449
18.6.2 Data Transformation	455
18.6.3 Data Manipulation	460
18.6.4 Bit Transformation	463
18.6.5 Communication	465
18.6.6 String Operation Functions	481
18.6.7 Miscellaneous	508
18.7 How to Create and Execute a Macro	516
18.7.1 How to Create a Macro	516
18.7.2 Execute a Macro	521
18.8 Some Notes about Using the Macro	522
18.9 Use the Free Protocol to Control a Device	523
18.10 Compiler Error Message	529
18.11 Sample Macro Code	535
18.12 Macro TRACE Function	540
18.13 The Usage of String Operation Functions	549
18.14 Macro Password Protection	560
Chapter 19 Set HMI as a MODBUS Server	562
19.1 Setting HMI as MODBUS Device	562
19.1.1 Creating a MODBUS Server	562
19.1.2 Read from / Write to MODBUS Server	565
19.2 Changing the Station Number of a MODBUS Server in Runtime	568
19.3 About MODBUS Address Type	569

Chapter 20 How to Connect a Barcode Device	570
20.1 How to Connect a Barcode Device	570
Chapter 21 Ethernet Communication and Multi-HMI Connection	574
21.1 HMI to HMI Communication	575
21.2 PC to HMI Communication	577
21.3 Operate the PLC Connected with other HMI.	579
Chapter 22 System Reserved Words / Bits	581
22.1 The Address Ranges of Local HMI Memory	582
22.1.1 Bits	582
22.1.2 Words	583
22.2 HMI Time	584
22.3 User Name and Password	585
22.4 Data Sampling	586
22.5 Event Log	587
22.6 HMI Hardware Operation	589
22.7 Local HMI Network Information	590
22.8 Recipe and Extended Memory	591
22.9 Storage Space Management	593
22.10 Touch Position	594
22.11 Station Number Variables	595
22.12 Index Register	597
22.13 MTP File Information	598
22.14 MODBUS Server Communication	599
22.15 Communication Parameters Settings	601
22.16 Communication Status with PLC (COM)	604
22.17 Communication Status with PLC (Ethernet)	606
22.18 Communication Status with PLC (USB)	609
22.19 Communication Status with Remote HMI	610
22.20 Communication Status with Remote PLC	613
22.21 Communication Error Messages & No. of Pending Cmd	616
22.22 Miscellaneous Functions	617
22.23 Remote Print/Backup Server	619
22.24 EasyAccess	620
22.25 Pass-Through Settings	621
22.26 Disable PLC No Response Dialog Box	622
22.27 HMI and Project Key	623
22.28 Fast Selection Window Control	624

22.29 Input Object Function 22.30 Local/Remote Operation Restrictions	625 626
Chapter 23 HMI Supported Printers	627
Chapter 24 Recipe Editor	630
24.1 Introduction 24.2 Settings of Recipe Editor	630 632
Chapter 25 EasyConverter	634
<ul> <li>25.1 Introduction</li></ul>	
Chapter 26 EasyPrinter	648
<ul> <li>26.1 Using EasyPrinter as a Printer Server</li></ul>	
Chapter 27 EasySimulator	
27.1 Prepare Files 27.2 Modify the Content of xob_pos.def	668 669
Chapter 28 Multi-HMI Intercommunication (Master-Slave Mode)	671
28.1 How to Create a Project of Master HMI	672

28.2 How to Create a Project of Slave HMI	. 673
28.3 How to Connect M1500 Project of Slave HMI	.676
Chapter 29 Pass-Through Function	. 680
29.1 Ethernet Mode	. 681
29.1.1 How to Change the Virtual Serial Port	. 682
29.1.2 How to Use Ethernet Mode	. 685
29.2 COM Port Mode	. 687
29.2.1 Settings of COM Port Mode	. 687
29.2.2 HMI Work Mode	. 689
29.3 Using System Reserved Addresses to Enable Pass-Through Function	. 693
Chapter 30 Project Protection	. 694
30.1 XOB password	. 694
30.2 Decompilation is prohibited	. 695
30.3 Disable HMI upload function [LB9033]	. 696
30.4 Project protection [Project Key]	. 697
30.5 Project password [MTP file]	. 698
Chapter 31 Memory Map Communication	. 700
Chapter 32 ASCII Protocol	. 710
32.1 Command List	. 710
32.2 Optional Parameters	.711
32.3 Network Support	. 712
32.3.1 Wiring	. 712
32.3.2 Addressing	. 712
32.3.3 Broadcast Messages	. 712
32.4 Command Usage	. 713
32.4.1 RD (Batch Read)	. 713
32.4.1.1 Request	. 713
32.4.1.2 Reply	. 714
32.4.2 WD (Batch Write)	. 715
32.4.2.1 Request	. 715
32.4.2.2 Reply	.716
32.4.3 RR (Random Read)	.717
32.4.3.1 Request	. 717
32.4.3.2 Reply	. 718
32.4.4 RW (Random Write)	. 719
32.4.4.1 Request	. 719

32.4.4.2 Reply	
32.4.5 RC (Read Coils)	
32.4.5.1 Request	720
32.4.5.2 Reply	721
32.4.6 WC (Write Coils)	
32.4.6.1 Request	
32.4.6.2 Reply	724
32.4.7 Error Codes	725
Chapter 33 EasyDiagnoser	726
33.1 Overview and Configuration	
33.2 EasyDiagnoser Settings	729
33.3 Error Code	736
33.4 Save As	737
33.5 Window Adjustment	738
Chapter 34 AB EtherNet/IP Free Tag Names	739
34.1 Import User-Defined AB Tag to EB8000	740
34.2 Adding New Data Type	742
34.3 Paste	745
34.4 Miscellaneous	748
34.5 Module-Defined	749
Chapter 35 FTP Server Application	753
35.1 Login FTP Server	753
35.2 Backup History Data and Update Recipe Data	753



# Chapter 1 EasyBuilder Pro Installation and Startup Guide

### 1.1 EasyBuilder Pro Installation

#### Software:

Download EasyBuilder Pro configuration software from EasyBuilder Pro CD or visiting Weintek Labs, Inc.'s website at <u>http://www.weintek.com</u> to obtain all software versions available (including Simplified Chinese, Traditional Chinese, English, Italian, Korean, Spanish, and French version) and latest upgraded files.

#### Hardware Requirements (Recommended):

CPU: INTEL Pentium II or higher Memory: 64MB or higher Hard Disk: 2.5GB or higher (Disc space available at least 10MB) CD-ROM: 4X or higher Display: 256 color SVGA with 800 x 600 resolution or greater Keyboard and Mouse Ethernet: for project downloading/uploading RS-232 COM: At least one available RS-232 serial port required for on-line simulation Printer

#### **Operating System:**

Windows 2000 / Windows NT / Windows XP / Windows Vista / Windows 7.



### 1.2 Steps to Install EasyBuilder Pro

#### 1. Installing EasyBuilder Pro:

Put the EasyBuilder Pro Installation CD into the CD drive. The computer will run the program automatically and bring up a screen showing an area to click to begin the EasyBuilder Pro installation. If the auto-run sequence does not start, browse the CD, and find the root directory of **[Autorun.exe]** manually. The installation screen is shown below.



2. Click **[Install]**, users will see the window below, select the language and click **[Next]** following the installation instructions.

Select	Setup Language	X
1P	Select the language to use during the installation:	
	English	/*
	English Español Français Italiano	
	English, Spanish, French, Italian, Simplified Chinese, Traditional Chinese, Korean	





3. Users will be asked if they would like to remove the old versions of EasyBuilder Pro. Please tick those should be removed and click **[Next]** to continue.

🔂 Setup - EasyBuilder8000 ¥4.34
EB8000 Remove Information Find other version of EB8000 that has been installed.
Would you want to remove EB8000? Select the version that you want to remove.  EasyBuilder 8000 V3.4.5 Chinese (Traditional)
EB8000V410 EB8000V420 EB8000V421
EB8000V430 EB8000V433 EB8000V433
< <u>B</u> ack <u>N</u> ext > Cancel



4. Designate a new folder for EasyBuilder Pro installation or choose the folder recommended and then click **[Next]**.

🚰 Setup - EasyBuilder8000 ¥4.34
Select Destination Location Where should EasyBuilder8000 V4.34 be installed?
Setup will install EasyBuilder8000 V4.34 into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\EB8000 Browse
At least 417.6 MB of free disk space is required.
< <u>B</u> ack <u>N</u> ext > Cancel

5. Users will be enquired to select a start menu folder to save the program's shortcuts. Click **[Browse]** to designate a folder or use the folder recommended then click **[Next]**.

🔂 Setup - EasyBuilder8000 ¥4.34	
Select Start Menu Folder Where should Setup place the program's shortcuts?	J.
Setup will create the program's shortcuts in the following Start Menu folder. To continue, click Next. If you would like to select a different folder, click Browse.	
EB8000 Browse	
< <u>Back</u> <u>N</u> ext > Cance	el



6. Users will be enquired if there are any additional tasks to be done. For example: [Create a desktop icon]. Tick it if needed then click [Next] to continue.

j <mark>p</mark> Setup - EasyBuilder8000 ¥4.34	
Select Additional Tasks Which additional tasks should be performed?	
Select the additional tasks you would like Setup to perform while installing EasyBuilder8000 V4.34, then click Next. Additional icons:	
< <u>B</u> ack <u>N</u> ext >	Cancel

7. At this moment all the settings are done. Please check if they are all correct. If any changes need to be made, click **[Back]** or click **[Install]** to start installing.

🔂 Setup - EasyBuilder8000 ¥4.34	
Ready to Install Setup is now ready to begin installing EasyBuilder8000 V4.34 on your computer.	
Click Install to continue with the installation, or click Back if you want to review or change any settings.	
Destination location: C:\EB8000	
Start Menu folder: EB8000	
Additional tasks: Additional icons: Create a desktop icon	
۲ ۲	
< <u>B</u> ack Install Cance	



#### 8. Installation processing.

🔂 Setup - EasyBuilder8000 ¥4.34	
Installing Please wait while Setup installs EasyBuilder8000 V4.34 on your computer.	
Extracting files C:\EB8000\library\meter_01.flb	
	Cancel

9. Click [Finish] to complete the installation.





10. Start EasyBuilder Pro project from menu [Start] / [All Programs] / [EB8000].



The description of each item in EasyBuilder Pro menu:

Item	Description
👔 AB Data Type Editor	When using AB Tags, this tool can be applied to edit Tags
	structure
😼 EasyBuilder8000	EasyBuilder Pro editing software
👺 EasyConverter	Conversion tool for Data Sampling and Event Log
💖 EasyDiagnoser	Communication monitoring tool via online simulation
🛬 EasyPrinter	Remote printer server
😼 EasySimulator	Tool for executing simulation without installing EasyBuilder
	Pro
🦪 Project Manager	EasyBuilder Pro project management
🖄 RecipeEditor	Tool for setting format of Recipe data. Users can open Recipe
	data or data in External Memory here.
🔁 ReleaseNote	Notes for EasyBuilder Pro version and latest information
🛃 Uninstall EasyBuilder8000	To uninstall EasyBuilder Pro



HMI i Series support downloading/uploading project via USB cable.

After installing EasyBuilder Pro, Please go to [Computer Management] / [Device Manager] to check if USB driver is also installed, if not, please refer to <u>installation</u> <u>steps</u> to manually install.



# **Chapter 2 Project Manager Operations**

After installing EB8000 software, users will see a **[Project Manager]** shortcut, double click it, users will see a window as shown below.

The Project Manager is a software shell for launching several utilities. Some functions are duplicated in the EasyBuilder8000 screen-editing program. Project Manager can operate as a stand-alone program.

In this chapter, each function will be introduced respectively.

🏈 Project Manager		
HMI IP, Password		
Туре : МТ6000/8000 і	i Series 🛛 🔽	
Settings	Reboot HMI	
Connection		
OEthernet OUSB (	cable (i series only)	
Data/Event Log File Information		
Utility		
EasyBuild	er8000	
EasyConverter	EasyAddressViewer	
EasyPrinter	EasyDiagnoser	
Recipe/Extended Memory Editor		
Build Download Data for CF/ USB Disk		
Download	Upload	
On-line Simulation	Off-line Simulation	
Pass-thro	pugh	
Help	Exit	



# 2.1 HMI IP, Password

Project Manager	
Password Reset/Download :	
Upload : 111111	
OK	Cancel

#### Settings

When operating MT8000/MT6000 HMI by Ethernet or USB cable, users need to designate the correct IP address and password in HMI. Press [Settings], [Reset and Download] functions share a set of password while [Upload] function uses another set.

The password provides protection against unauthorized access to the HMI. Be sure to record any password change, otherwise, while resetting password to default, the project and data in HMI will be completely erased.

#### Reboot HMI

There are certain situations that the HMI should reboot, for example, when updating the files in it. Users don't need to cut power while rebooting. After rebooting, everything returns to the conditions of startup.



🏈 Project Manager		
HMI IP, Password		
Type : MT6000/8000 i Series		
Settings	Reboot HMI	
Connection		
○Ethernet ⊙USB	cable (i series only)	
Data/Event Log File Information		
EasyBuild	der8000	
EasyConverter	EasyAddressViewer	
Casily Control Col	- Edit Picerseer	
EasyPrinter EasyDiagnoser		
Recipe/Extended Memory Editor		
Build Download Data for CF/ USB Disk		
Download	Upload	
On-line Simulation	Off-line Simulation	
Pass-thr	ough	
Cohere L IMAT		
stop HMI scan font : finished		
reset HMI : finished	<u> </u>	
Help	Exit	

Set the correct IP address when operating HMI via Ethernet.

HMI IP: 192.168.0.103 👻



# 2.2 Utility

Item	Description
EasyBuilder8000	To launch the EasyBuilder8000 screen editor.
Easy Converter	Conversion tool for Data Sampling and Event
	Log.
Easy Printer	Remote printer server.
EasyAddressViewer	Review the register range of device types for
	each PLC supported.
EasyDiagnoser	Communication monitoring tool via online
	simulation.
Recipe / Extend	Provide file format conversion and data editing
Memory Editor	function for Recipe/Extend Memory.
Build Download Data	The project and data can also be downloaded to
for CF Card/USB Disk	the HMI by CF card or USB memory stick. This
	function is to build this kind of download data as
	shown below.

#### \* Build Download Data for CF Card/USB Disk

Project Manager
Select the folder to save download data :
KA Browse
Sources
✓ Project
PLEASE INPUT XOB FILE NAME ! Browse
Recipe (RW)
PLEASE INPUT RECIPE FILE NAME ! Browse
Recipe A (RW_A)
PLEASE INPUT RECIPE_A FILE NAME ! Browse
✓ Data log
PLEASE INPUT DATA LOG FILE NAME ! Browse
Build





Setting	Description
	Insert CF card or USB stick to PC and press
Select the folder to save download data	[Browse] to assign the file path (or
	directory name) and then press [Build]. The
	whole contents of the source files will be
	downloaded to USB stick or CF card
Project	Press [Browse] to assign the desired
Recipe (RW)	specific files for download data.
Recipe A (RW_A)	
Data log	

Note: The path of download data should avoid designating root directory of PC. For example, "**c**:\", also, directory name such as "**f**:\\" is illegal and should be written as "**f**:\".

## 2.2.1 Steps to Download Project via USB or CF Card

Take downloading data to the folder named "123" (K:\123) in USB stick for example.

When USB stick (project or recipe included) is inserted to the HMI, a pop-up [Download / Upload] dialog will appear after few seconds. Please select [Download] and input Download Password. Check [Download project files] and [Download history files] in [Download Settings] dialog, and then press [OK]. After that, [Pick a Directory] dialog will appear. Please select directory: *usbdisk/device-0/123* and then press [OK]. Project will be automatically updated.

Note: Even if users only download historical files, it is still necessary to reboot HMI manually.



### 2.3 Transfer

#### 2.3.1 Download

Download source files to HMI through Ethernet or USB cable. Press [Download] and the dialog displays as below:

ownload		
Firmware		
Project	C:\Documents and Settings\Nicolas\456.xob	Browse
RW BW	PLEASE INPUT RECIPE FILE NAME !	Browse
RW_A	PLEASE INPUT RECIPE 1 FILE NAME !	Browse
🗹 Data log	PLEASE INPUT DATA LOG FILE NAME !	Browse
☐ Install X-series med ✓ Startup screen	lia-player drivers C:Documents and Settings\Nicolas\cheer_8.bmp	Browse
	* i series only (OS image must be 20090415 or later)	
Connection • Ethernet	O USB cable (i series only)	Þ
IP: 19	2.168.1.212	
✓ Reboot HMI after of ■ Reset data log	Jownload Reset recipe Reset event log	

Description
Check this to update all of the kernel
programs of HMI. It is necessary when
the latest EB8000 version is
downloaded the first time.
To assign the desired specific path for
file downloading.

EasyBuilder8000 User's Manual



Install X-series media-player	It is necessary when EB8000 is				
drivers	downloaded to X series the first time.				
Startup Screen	If this box is ticked, the assigned BMP				
	picture will be downloaded to HMI. After				
	downloading, HMI will reboot, this				
	picture will be shown after rebooting,				
	and then load in the project. Users are				
	allowed to use their logos as the start up				
	screen through this method.				
Reboot HMI after download	Automatically reboot HMI after				
	downloading.				
Reset recipe	Check the box to erase the selected				
Reset event log	specific files in HMI before downloading				
Reset data log	process.				

### 2.3.2 Upload

Upload files from HMI to PC by Ethernet or USB cable and the dialog shows as below: Users have to assign the desired path for file storage before uploading.

<b>TEK</b>	Ea	asyBuilder8000 Us
Upload		
	MT6000/8000 i Series	
Project	C:\Documents and Settings\Nicolas\456.xob	Browse
<b>I</b> R₩	PLEASE INPUT RECIPE FILE NAME !	Browse
RW_A	PLEASE INPUT RECIPE 1 FILE NAME !	Browse
🗹 Data log	C:\Documents and Settings\	Browse
Event log	C:\Documents and Settings\	Browse
	Use CSV (Comma Seprated Values) format to save data/eve	ent log files.
Extend Memory (EM)	PLEASE INPUT EM FOLDER NAME !	Browse
Connection		
Ethernet	OSB cable (i series only)	
HMI name : Defa * OS 20091002 or late	ult HMI   Search  Search all  r supports	

Setting	Description
Project	To assign the desired specific path for
RW	file uploading.
RW A	
Data log	
Event log	
Extend Memory	



### 2.4 Simulation

# 2.4.1 On-line Simulation/Off-line Simulation

There are two types of simulations: On -line simulation & Off-line simulation.

By virtual device, PC simulates the operations of HMI without connecting with PLC and HMI. This shortens the time needed greatly even without the HMI in your hand.

While using Off-line simulation, users are allowed not to download the written project file to HMI, but still see how it is shown and operated on PC. Users don't need to connect PLC with PC under this mode. On the contrary, On-line simulation is executed by connecting PC with PLC and accurately set the communication parameters. When simulating on PC, if the control target is a local PLC (i.e. the PLC directly connected to PC), there is **10 minutes simulation limit.** 

Before executing On-line/Off-line Simulation features, please select the source of XOB file.



When executing on-line/off-line simulation, right click to use two functions:



Exit simulation

Run EasyDiagnoser

Screenshot

a. "Run EasyDiagnoser"

Execute EasyDiagnoser to monitor current communication status.

b. "Screenshot"

Capture and save current screen image as picture file in the screenshot folder under installation directory.



# 2.5 Pass-Through

The pass-through function allows the PC application to connect PLC via HMI. In this function, the HMI acts as a converter.

Pass-through provides two types of modes: Ethernet and COM port. Click **[Pass-through]** button on Project Manager to start the settings.

#### For more information, please refer to related chapter.

### 2.5.1 Ethernet

ass-through		
⊙ Ethernet	O COM port	
Virtual COM F	Port (PC <-> PLC)	
	СОМЗ	
PLC Connecti	on Port (HMI <-> PLC)	
HMUP	: 192.168.1.200	Stop Pass-through
		32
Install	Uninstall	Аррі
		E:

# 2.5.2 COM port

Ethernet	COM port			
HMLIF	P:		~	
	Get HMI Comm	unication Parameters		
	HMI work mode : Un	iknown		
⊂ Source COM	Port (PC -> HMI)			
	COM 1 🗸		R9232	~
Reveluet.		Data kita i	2.00	
Baud rate	. 9600	Data bits :	7 Bits	×
Pariț	V: None 🚩	Stop bits :	1 Bit	*
Destination C	OM Port (HMI -> PLC)			
	СОМ 2 🗸		RS232	*
Baud rate		Data bits :	7 Dite	
Dudita	. 9000			
Panţ	/: None 🎽	Stop bits :	1 Bit	*
Citart Press	turrunt Ottan D			
Start Pass-	hrough Stop Pa	ass-through		



# Chapter 3 Create an EasyBuilder8000 Project

In this Chapter, we will take Mitsubishi PLC as an example to illustrate how to create and compile a new EB8000 project, to simulate it on PC and to download the project to HMI.

#### 3.1 Create a New Project

First of all, click **[New]** icon on the toolbar to create a new project.

💊 Easy	Build	er 80	00 -	- [	Te
EB <u>F</u> ile	<u>E</u> dit	<u> </u> viev	v <u>O</u>	ption	ì
		¥	Ē	ß	Ľ
New	ncho	LJIS	3	~	1

Select HMI Model, check [Use template] and click [OK].

EasyBuilder(Copyright c 2006 Weintek Lab., Inc.)
Welcome to EasyBuilder8000. Please select your model.
Model : MT6070iH/MT8070iH/MT6100i/MT8100i (800 x 480) V Display mode : Landscape V Use template
OK Cancel

Under **[Device]** Tab, click **[New...]** button to correctly set up the **[Device Properties]** for communicating with the PLC.



Font	I	Extended Memory	Printer/Bac	kup Server
Device	Model	General	System Setting	Security
Device list :				
No Na	me Location	Device type	nterface I/E Protocol	Station no
ice Properties				
Name :	MITSUBISHI FXOr	n/FX2		
1		PLC		
4				
Location :	Local	Settings		
PLC type :	MITSUBISHI FXO	n/FX2		~
5	/.1.10, MITSUBIS	SHI FXON.so		
	-	-		
PLC I/F :	RS-485 4W	~	PLC default station no	o. : [0
сом : б	OM1 (9600.E.7.	1)		Settings
1		-,		Coccargo
[	Use broadcast	command		
Interv	al of block pack (	words) : 5	~	
Max, read	d-command size (	words): 32	×	
Max. write	e-command size (	words): 32	*	

Click [OK], device "MISUBISHI FX0n/FX2" is added to the [Device List].

System Parameter Settings								
Font Extended Memory Printer/Backup Server								
Device	Model	Gen	ieral	System	Setting S	Security		
Device list :								
No.	Name	Location	Device ty	pe	Interface	I/F Proto		
Local HMI	Local HMI	Local	cal MT8104iH (800 x		Disable	N/A		
Local PLC 1	MITSUBISHI FXOn	Local	MITSUBIS	SHI FXOn	COM1(9600,E,7,1)	RS4854V		



Now, if users would like to add a new object, such as **[Toggle Switch]**, click the icon on the tool bar.



A **[New Toggle Switch Object]** dialog will be shown as below. Correctly set the parameters of the object, click **[OK]** and place the object wherever users like in the window.

General Security Shape Label
Description .
Peed address
PLC name : MITSUBISHI FX0n/FX2
Address : Y
Invert signal
Write address :
PLC name : MITSUBISHI FX0n/FX2
Address : Y V 1
Write when button is released
Attribute
Switch style : Set OFF
Масто
Execute macro



A project with an object is completed as shown below.

EasyBuilder 8000 : MTP2 ·	[10 - WINDOW	<b>7_010 ]</b>								
EB File Edit View Option	Draw Objects	Libaary <u>T</u> o	ols <u>W</u> indow	Help					-	. 8×
D 🖻 🖬   X 🖻 🖻 의	2 🗠 / 🚭 🤋 🖡	? 🗣	<u>1</u> == 1	: 🗄 🦛 🐗	🛃 🖻 🖡	3 🕽 🔹	i 🛥 🛃			
-		- /	 ∧* ≡	= =   .	7 A -	II Language	1		-	
							- 14 - 14 - 14			100.1
		킄  미t•아	· <u> </u>	Left Alignment			ie ha   H		Go	100.
🕨 🛛 🖉 🔨 К С	비오 다 나는 비 나는 비	f 🕰 🖭	二 🛠 🛛	2 😨 🕒	🛄 🖉 🗄	📆 🎹 🚺	12	3 <	) ) S	tate O
Windows 👻 🗙	1 10 - WINDO	₩_010 ×						⊳		
- 3: Fast Selection								^		
- 5: PLC Response									<b>9</b>	1 💷
6: HMI Connection									1 H	8 🗟
- 7: Password Restriction									HPO 4	) 🔂
9	TS 0								123 1	<b>9</b>
= *10: WINDOW_010	13_0									r -0.
IS_U (Y-1, Y-1) (Bc										
- 12									de Ca	י 🔳
- 13									<u></u> -0- ½	ž
14									888 🗄	
10										8
										-
- 18										×
20									- 2	•
- 21									<u>•</u> 1 🖸	7
- 22										
23									÷	*
24										8
26										3
29								¥		
	<							>		
Left alignment			MT8104X (	640 x 480)						11.3



# **3.2 Save and Compile the Project**

In the menu, select [File] then select [Save], file will be saved as .mtp file. After file is saved, select [Tools] then select [Compile] to compile the project and check if the project can run correctly. A .xob file will be obtained after correctly compiling. A .xob file is needed while downloading to HMI.



A successfully compiled file will get the dialog as below:

Compiling				
Project name : C:\Documents and Settings\Nicolas\桌面\MTP1.mtp XOB file name : C:\Documents and Settings\Nicolas\桌面\MTP1.xob				
XOB password : Set	) (used in decompiler)		Decompilation is prohibited	
Select the languages used on the H	MI			
☑ Language 1	Language 2	Language 3	🗌 Language 4	
Language 5	Language 6	🔲 Language 7	🗌 Language 8	
C:\EB8000\V3.40\eng\font\MTP1\$0. C:\EB8000\V3.40\eng\font\MTP1\$1. 0 error(s), 0 warning(s) Object size : 38544 bytes Font size : 314444 bytes Picture size : 792988 bytes Shape size : 878 bytes Sound size : 36474 bytes Macro size : 14 bytes Total size : 1184944 bytes (1.1 succeeded	ttf (Arial) ttf (Times New Roman) 3M)			
Double click error messages to modi	ify the attributes of relativ	e objects !	<u> </u>	
Compile Build f	ont files			lose

Users are allowed to select the languages needed for the project by clicking **[Language 1 to 8]**.



### 3.3 Off-line and On-line Simulation

There are two types of simulations: On -line simulation & Off-line simulation.

While using Off-line simulation, users don't need to connect PLC with PC but still see how PLC is operated via a virtual device. On the contrary, On-line simulation is executed by connecting PC with PLC and accurately set the communication parameters.

Note: When doing On-line simulation on PC, if the target is a local PLC (i.e. the PLC directly connected to PC), there is a **10-minutes simulation limit.** 

#### 3.3.1 Off-line Simulation

To execute, click [Off-line Simulation].

* 👳	🕎 😫 🛄 🖉 📉	
<u>A</u> <u>U</u>	LOff-line Simulation	
C+2		

After clicking, users will see their project shown as below.





#### 3.3.2 On-line Simulation

To execute, click [On-line Simulation] after correctly connecting the device.




## 3.4 Download the Project to HMI

In the menu, select **[Tool]** then select **[download]** to download the project file to HMI. Before downloading, be sure to check if all the settings are correct.

*	<u>7</u>	<b>7</b>	막	CF	Z	esu.
<u>A</u>	U	L	an	own	load	•

Download	X
reset event log reset data log initializing downloading project downloading C:\Documents and Settings\Administrator\点面\程式檔\EB8000V410_100614_eng\firmware\cc downloading C:\Documents and Settings\Administrator\点面\程式檔\EB8000V410_100614_eng\firmware\gu downloading C:\Documents and Settings\Administrator\点面\程式檔\EB8000V410_100614_eng\firmware\gu downloading C:\Documents and Settings\Administrator\点面\程式檔\EB8000V410_100614_eng\firmware\gu downloading C:\Documents and Settings\Administrator\点面\程式檔\EB8000V410_100614_eng\firmware\gu downloading C:\Documents and Settings\Administrator\点面\程式檔\EB8000V410_100614_eng\font\test\$1 scan font : finished reboot HMI : finished finished	)) , , ,
● Ethernet:       ○ USB cable (i series only)       Password : Set         ↓       IP       Name         ↓       IP       Name         ↓       IP :       ✓	
Firmware     Font files     * Necessary if update firmware or execute download first time.	
Reset recipe     Reset event log	_
Reboot HMI after download	
Download     Stop     Exit	

Setting	Description				
HMI IP	Assign the IP address of HMI				
Password	Input the password				
Firmware	Check [Firmware] to update all of the kernel				
	programs of HMI.				



	Note: It is necessary when downloading file to	
	HMI the first time.	
Font Files	Download the font used in project to HMI.	
Reset recipe	Checking these, the selected files will be	
Reset event log	erased before downloading.	
Reset data log		
Reboot HMI after download	Checking this, HMI will reboot after finishing	
	downloading.	
Automatically using current	If this is checked, system will download project	
settings to download after	to HMI according to last settings.	
compiling		

#### \* Automatically Using Current Settings to Download after Compiling

1. Firstly, please go to **[Option] / [Function Properties]** then tick **[Automatic save and compile when download and simulate]**.

🥆 EasyBuilder8000 : MTP12.mtp - [10 - WINDOW_010 ]							
100	2 🖬   X 🖻 🖻	요요 🚳 🔋 😢 🙀 🔟 🏢 本					
* 5	🖉 🕎 🛄 🖉 🗄						
	-	★ A* = = I A + U Language 1					
ies <u>F</u> il	e <u>E</u> dit <u>V</u> iew <u>O</u> ptic	n <u>D</u> uaw Objects Libuary <u>T</u> ools <u>W</u> indow <u>H</u> elp					
	Windows	Function Properties					
タ 	Object list 3: Fast Selection 4: Common Wi 5: PLC Respons	Display : 🗹 Object ID					
	6: HMI Cornec 7: Password Res 8: Storage Space	Using function key to make shape library object					
	■ *10: WINDOW	Automatic save and compile when download and simulate					
<ul> <li>Image: A state</li> <li>Image: A state<td></td><td>OK Cancel</td></li></ul>		OK Cancel					

2. Secondly, in **[Download]** dialogue box, tick **[Automatically using current settings to download after compiling]** to enable this function.





Click [Download] to start downloading the project.

Another way to download project to HMI is to set a HMI name. Before using this function, please input the HMI Name in the **[System settings]** window in HMI as shown below.

ystem settings	
Network <sup>V</sup> Time/Dat	e <sup>\</sup> Securit / HMI name setting bg \\
HMI name tes	

After setting the HMI Name, please click **[Name]** in the **[Download]** window on PC as below,



Download			
© Ethernet	OUSB cable (i series only)		Password : Set
+ OS 20091002 or later	Search Search all	192.168.1.103 (Test)	
Firmware * Necessary if update	Font files	d first time.	
	Reset event log	Reset data log	
Reboot HMI after d	lownload		
Automatically using	current settings to download	after compiling	
Download	Stop		Exit

Setting	Description				
HMI Name	Input the HMI name for downloading project				
Search	Input the HMI name to search the designated HMI				
	Image: 100 Name     Image: 100 Name       HMI name : 1001     Search all       * OS 20091002 or later supports     Image: 100 Name				
Search all	Click to search the HMI shares the same network				
	Search al     * OS 20091002 or later supports				

Password	Input the password				
Firmware	Check [Firmware] to update all of the kernel				
	programs of HMI.				
	Note: It is necessary when downloading file to HMI				
	the first time.				
Font Files	Download the font used in project to HMI.				
Reset recipe	Checking these, the selected files will be erased				
Reset event log	before downloading.				
Reset data log					
Reboot HMI after download	Checking this, HMI will reboot after finishing				
	downloading.				
Automatically using current	If this is checked, system will download project to				
settings to download after	HMI according to last settings.				
compiling					

Click [Download] to start downloading the project.



# **Chapter 4 Hardware Settings**

## 4.1 I/O Ports of HMI



## 4.1.1 USB Port

Support devices with USB interface, such as mouse, keyboard, USB stick, printer...etc.

## 4.1.2 Ethernet Port

Connect devices with Ethernet communication interface, such as PLC, laptop...etc; support exchanging data via Network.

## 4.1.3 CF Card or SD Card

Download/ Upload project via CF Card or SD Card, including Recipe transfer, Event Log, Data Log...etc.

## 4.1.4 Serial I/O Port

COM ports, RS-232, RS485-2W/4W, can be connected to PLC or other peripheral devices. Here we view RS-422 the same as RS-485 (4 wire). Please refer to the *"PLC connection"* 



*guide*" to make sure that PLC and HMI are correctly connected. Meanwhile, please make sure all DIP switches at the back of HMI are pulled down (means off, the default value).

In addition, Weintek provides [MT8-COM1 Multi-Connector cable] and [MT8-COM3 Multi-Connector cable] to expand one COM port to multiple independent COM ports so that the convenience and efficiency of the operation can be improved.

## 4.2 HMI System Settings

Before operating HMI, users have to complete the HMI system settings. After this, users can develop their own operation interface through EB8000 editing software. The following illustrates each system setting respectively.

## 4.2.1 System Reset

Each HMI is equipped with a set of reset button and DIP switch. When users use DIP switch to change modes, corresponding functions will be triggered.

If system password is lost or forgotten, users can set DIP Switch 1 to "ON" and the rest remain "OFF", then reboot HMI.





HMI will switch to touch screen calibration mode. After calibration, the pop-up window appears as shown below. Users will be inquired if they would like to restore the system password to the default.

Warning!	X
Restore to default password?	
Yes No [04]	

When **[YES]** is chosen, another pop-up dialog appears as below. The system will ask users to type **[yes]** to confirm to restore system password to default. Then click **[OK]**.

(The default password is 111111. However, other passwords, including download and upload password, have to be reset.)



The illustration above shows the steps to restore factory settings of T and i Series HMI. For X Series, users will need a connected USB keyboard, and press any key (or space key) right when the first image displayed as HMI power ON to enter the menu. Select "Factory Mode", the window mentioned will pop up when system displays project. In case users may miss the very first image shown, to press space key continuously since HMI power ON will ensure entering the system setting window.

Note: The project and data in the HMI will all be removed once it is reset.



Dip Switch	SW1	SW2	SW3	SW4	Mode
	ON	OFF	OFF	OFF	Touch screen calibration mode (T, i series)
	OFF	ON	OFF	OFF	Hide System Toolbar (i , X V2 series)
	OFF	OFF	ON	OFF	Boot loader mode
	OFF	OFF	OFF	ON	Enable front panel power switch (X series)
1234	OFF	OFF	OFF	OFF	Normal

## 4.2.2 System Toolbar

After rebooting HMI, users can set the system with System Toolbar at the bottom of the screen. Normally, this bar is hidden automatically. Only by touching the target at the right-bottom corner of screen will the System Toolbar pops up.







# 4.2.2.1 Large Keyboard

Use large keyboard to input text information.

		Virtual Keyb	oard				
F1 F2 F	-3 F4	F5 F6 F7 F8	F9 F10	F11 F12	Backspace		
Esc 1	2 3	4 5 6 7	89	0 –	= \ `		
Tab q	Tab q w e r t y u i o p [ ] Del						
Ctrl	Ctrl a s d f g h j k l ; ' Ret						
Shift	z ×	c v b r	n m ,	. /	Shift		
Caps	Alt		+	→	↑ ↓		



## 4.2.2.2 Small Keyboard

Use small keyboard to input numerical information.

Virtual Keyboard 📃				
Num Lock	1	*		
7 Home	8 Up	9 PgUp	+	
4 Left	5	6 Right	-	
1 End	2 Down	3 PgDn	<b>Futur</b>	
0 Ins		Del	Enter	

## 4.2.2.3 System Information

Network: Display Network information, including HMI IP address and related information.

System information (Default HMI) 🛛 🛛 🖂
Network\Version\
IP Address: 192.168.1.40
Net Mask: 255.255.255.0
Route Address: 192.168.1.254
Ok

Version: Display information of the HMI system version.





## 4.2.2.4 System Setting

Set or modify system parameters. Password has to be confirmed for security.

System settings
Enter your password:
Password:
Ok Cancel

#### a. Network

A project can be downloaded to HMI via Ethernet. The IP address of target (HMI) must be correctly set. If **[Auto Get IP Address]** is selected, IP address will be automatically assigned from local DHCP network. If **[IP address get from below]** is selected, IP address and other network information have to be inputted by the user.



System settings I>				
Network \Time/Date	e \Security Misc ory me \r setting \			
Obtain an IP A     O IP address ge	t from below			
IP address	192, 168, 1, 40			
Subnet Mask	255, 255, 255, 0			
GateWay	192 . 168 . 1 . 254			
	Cancel Apply OK <-			

### b. Time/Date

This page is for setting HMI local time and date.

System settings				
Network) Time/Date \Security Misc ory me \r setting				
Year: 2010 Mon: 7 Day: 21				
Week: <b>3</b>				
Cancel Apply OK <				



#### c. Security

The default of the password is 111111. EB8000 provides strict security for the HMI.



#### [Local Password]

Password for entering the system

### [Upload Password]

Password for uploading the project

#### [Download Password]

Password for downloading the project

### [Upload (History) Password]

Password for uploading the historical data.

Password confirmation:



	IX
Please enter you r	new password
password: *****	•
confirm: *****	k
Password match!	
ОК	Cancel

### d. History

For clearing the history data in HMI: [Recipe], [Eventlog] and [Datalog].

System settings			
Network <sup>V</sup> Time/Date <sup>V</sup> Security/ History me r setting			
Clear Recipe			
Clear Eventlog			
Clear Datalog			
Clear			
Cancel Apply OK <-			

#### e. Miscellaneous

Use the rolling bottom on the screen to adjust the brightness of LCD.



System settings
Network Time/Date Security Misc pry me r setting Backlight
Brightness
Download setting
Popup download window
Restart after download/upload
Cancel Apply OK <

#### f. Upgrade firmware

For users to upgrade firmware or to enable portrait mode. (Supported only by I series)

System settings	×
/ Network <sup>Y</sup> Time/Date <sup>Y</sup> Securit // Firmware setting \g \ Upgrade	
Upgrade firmware	
Portrait Mode	
● 0 ● 90 ● 270 (It will take effect at next reboot)	
Cancel Apply OK <	



#### g. CF card Status

When new external device is detected, this function will be enabled.

Netword	Time/Date Secunty History Backli CF card
	CF card Status
	Download/Upload
	Download Project
	Upload Project
	Restart Project and exit
	Cancel
	Restart after download/Upload
	Time remaining I

#### h. VNC server

Allows users to monitor and control the remote HMI through Ethernet.



System settings
Network <sup>V</sup> Time/Date <sup>V</sup> Securit/// VNC server setting
Start V/VC
Stop V/NC
VNC login password

Step 1. Enable VNC server and set the password in HMI.

Step 2. Install Java IE or VNC viewer in PC.

After installing Java IE, enter HMI IP: (The following takes <u>http://192.168.1.28</u> as an example)

🗿 VNC viewer for Java - Microsoft In	iternet Explorer	F		[		
檔案(F) 編輯(E) 檢視(V) 我的最愛	?(A) 工具(T)	説明(H)				
🔇 l-ā 🔹 🕥 - 💌 💈 🄇	🏠 🔎 搜尋	🔶 我的最愛	<b>⊗</b> ⊗-	28	W	*
網址(D) 🕘 http://192.168.1.28/				<b>~</b>	移至	Links ×
						1
REAL	2					
	🏄 VNC Viewe	r : Connection D	etails 🛛		$\mathbf{X}$	
	Serve	er: 1				
	Encryptio	n: Not supported	~			
	About	Options	ок	Cancel		
VNC Viewer Free Edition 4.1						
Copyright (C) 2002-2004 RealVNC Ltd	l,					
See http://www.realvnc.com for informa	ation on VNC.					

For VNC viewer, enter HMI IP address and password.

🕌 VNC Viewer : Connection Details			▦ - □ ⊠
Server:	192.168.1.28		
Encryption:	Not supported	~	
About	Options	ОК	Cancel

🕌 VNC Authentication [No Encryption] 🌐 🔳 🔀		
Username:		
Password:	xxxxxxx	





Note:

- (1) One HMI allows only one user to log in VNC server at one time.
- (2) If users leave VNC server unused for one hour, HMI system will log out automatically.

#### i. HMI name

Set the HMI name to download/upload a project.

System settings
/Network <sup>)</sup> Time/Date <sup>)</sup> Securit/ HMI name setting \g\
HMI name: Default HMI
Cancel Apply OK <드



## 4.2.2.5 Touch Screen Calibration Mode



In this mode when users power on MT8000 series, the screen will display a "+" sign upper-left of the screen. Use a stylus or finger to touch the center of the "+" until it moves. The "+" moves to upper-left, upper-right, lower -left, lower-right and center of screen. When all five "+" are touched, the "+" will disappear. The Touch Screen parameter will be stored at Flash Rom.

**Note:** Only X series HMI are with this shortcut of touch screen calibration mode in system toolbar. For other series, please use DIP switch 1 to adjust.



### 4.3 HMI Download Settings

A project or data can be downloaded to HMI via SD card or USB disk. Insert SD card or USB disk and designate the directory path. All contents under this directory will be downloaded to HMI. When HMI detects new external devices, the following screen appears:

	Download
	Upload
Restart	Project and
	Cancel
Restart :	after download/

Several functions can be selected at this time and some of them need password confirmation as illustrated below:

Download Settings	$\times$
Password: Download project files Download history files Clear history files Ok Cancel	

After the password is confirmed, directory names of the SD card...etc will be displayed in **[Pick a Directory]** window as below (pccard -> CF card (SD card); usbdisk -> USB device)



	Pick a Directory		
Directory: /pccard		F	£
<b>∲≷ pccard</b> ⊕Susbdisk			
- usbulsk			
1			
		ок <= С	ancel

Select the download path for project and click **[OK]** for downloading.

Note: Users have to create download data from [Build Download Data for CF/USB Disk] in Project Manager.

Generally, Project Manager divides downloaded files into two directories:

MT8000

Project storage

History

When users download the history data, this directory will be created.

An example which shows the directory of target file is shown below.

USB Disk/CF Card Data	
Select the folder to save download data :	
F:\download	Browse
Use user-defined start screen	
Start-screen file destination :	
cil	Browse
Build	Exit



The structure of saved data is as the diagram below:



Users have to select **the top layer of the directory of the target file** when downloading. In other words, take the structure above as an example; **download** must be selected instead of choosing **mt8000** or **history**.

Take the illustration below as another example: If USB disk only stores **mt8000** directory but don't include history data. In this case, users must choose **disk\_a\_1** (the top layer of target file that contains file of mt8000) to correctly download the file.

Pick a Directory	$   \times  $
Directory: /usbdisk/disk_a_1	
₱ <sup>®</sup> pccard ₱ <sup>®</sup> usbdisk	
<mark> </mark>	
Image: state sta	
OK < Can	cel



# **Chapter 5 System Parameter Settings**

Enter EB8000, select menu [Edit] / [System Parameters...] and the [System Parameter Settings] dialog appears:

Font			Extended Memory		Printer/Ba	ckup Server
Device		Model	General	Syste	m Setting	Security
evice list :						
No.	Name	Location	Device type	Interface	I/F Protocol	Station no.
Local HMI	Local HMI	Local	MT6070iH/MT8070	Disable	N/A	N/A

System Parameter Settings are divided into eight parts: [Device], [Model], [General], [System Setting], [Security], [Font], [Extended Memory], and [Printer/Backup Server].

These will be introduced respectively in this chapter.



## 5.1 Device

Parameters in **[Device]** tab determine all of the attributes of each device controlled by the HMI they are connected with. The device can be a PLC, a remote HMI, or a PC.

After opening a new \*.mtp file in EB8000, a default device: "Local HMI" is shown in the **[Device List]**. This "Local HMI" is used to identify current HMI, which means, every \*.mtp file must at least contains one "Local HMI" in **[Device List]**.

Select **[Settings]** under the device list, A dialogue **[Device Properties]** will be shown as below. From this we know that the attribute of "Local HMI" is a "HMI" and the location is "Local".

Device Properties	
Name : Local HMI	
⊙ HMI	OPLC
Location : Local	Settings
Interval of block pa	ick (words) : 5
	OK Cancel

Steps to add a new device:



## 5.1.1 How to Control a Local PLC



The so-called "local PLC" means a PLC which is connected to the local HMI directly. To control a local PLC, users need to add this type of device first. Click **[New...]** under the Device list and the **[Device Properties]** dialog appears. Please correctly fill in all of the properties required.

Take a local PLC MITSUBISHI FX0n/FX2 as an example:

Device Properties
Name : MITSUBISHI FX0n/FX2
Location : Local Settings
PLC type : MITSUBISHI FX0n/FX2
V.1.10, MITSUBISHI_FX0N.so
PLC I/F : RS-485 4W
PLC default station no. : 0
Default station no. use station no. variable
Use broadcast command
COM : COM1 (9600,E,7,1)
Settings
Interval of block pack (words) : 5
Max. read-command size (words): 32 🗸
Max. write-command size (words) : 32
OK Cancel

Setting	Description
Name	The name of the device set by user.
HMI or PLC	To confirm whether this connected device is a HMI or PLC. It's [PLC]
	in this example.



Location	<b>[Local]</b> or <b>[Remote]</b> . Showing whether this device is connected to Local HMI or being remote controlled. Select <b>[Local]</b> in this case.				
PLC type	Type of PLC. Select MITSUBISHI FX0n/FX2 in this case.				
PLC I/F	Five PLC interfaces are available: [RS-232], [RS-485 2W], [RS-485				
	4W], [Ethernet], and [USB].				
	If the interface is [RS-232], [RS-485 2W], or [RS-485 4W], click				
	[Settings] and then [Com Port Settings] dialog appears. Users				
	need to correctly set the COM port communication parameters.				
	COM Port Settings				
	COM : COM 1 V Timeout (sec) : 1.0 V Band rate : 9600 V Turn around delay (ma) : 0				
	Data bits : 7 Bits 👻 Send ACK delay (mt) : 0				
	Parity : Even M Parameter 1 : 0				
	Stop bills : 1 But Parameter 2 : 0 Parameter 3 : 0				
	OK Cancel				
	[Timeout]				
	If the communication between PLC and HMI is disconnected over the				
	set time limit in [Timeout] parameter, a pop out window No. 5 will be				
	shown in HMI as an alert saying "PLC No Response".				
	PLC no response				
	[Turn around delay]				
	While sending the next command to PLC, HMI will delay it obeying				
	the set time interval in [Turn around delay] parameter. This may				
	influence the efficiency of the communication between HMI and PLC.				
	If no specific request to be made, "0" is to be set.				
	If the PLC used is in SIEMENS S7-200 Series, this parameter needs				
	to be set to "5" and [Parameter 1] "30".				
	If the interface is [Ethernet] click [Settings] ] and then [ID Address				



	Settings] dialogue appears. Users need to correctly set IP address			
	and Port no. of the PLC.			
	IP Address Settings			
	IP address : 192 . 168 . 1 . 34			
	Port no. : [500]			
	Timeout (sec): 1.0 V Turn around delay (ms): 0			
	Send ACK delay (ms): 0 Parameter 1: 0			
	Parameter 2 : 0 Parameter 3 : 0			
	OK Cancel			
	If the interface is <b>[UCD]</b> as further acttings need to be done. Discus			
	in the interface is <b>[USB]</b> , no further settings need to be done. Please			
	check if all the settings in [Device Properties] are correct.			
PLC default	PLC should be set with a read address alone with a station no. for			
station no.	HIVI to locate and communicate with it. If this address does not			
	as the station no. of PLC			
	In addition, station no, can be set in the read address of PLC directly			
	Take address 1#20 as an example			
	rake audress 1#20 as an example.			
	Read address			
	PLC name : MITSUBISHI FX0n/FX2 Setting			
	Address : T V 1#20			
	Address			
	PLC name : MITSUBISHI FX0n/FX2			
	Address : 1#20			
	Address format : DDD [range : 0 ~ 255]			
	summer construction of the second			
	"1" means PLC station no, and has to be named from 0 to 255.			
	"20" means PLC address, the "#" sign is used to separate station no.			
	and address.			
Default station	When setting PLC properties, station no. variables can be selected			
no. use station	and used as [PLC default station no.]. LW10000~LW10015 can be			
no. variable	used to set station no. variables.			
	When using this function, if the station no. is not specified for PLC			



	address, it will be decided by the station no, variable of default station			
	no. In this example var3 is set for default station no. The following			
	demonstrates how the PLC address station no is set			
	a The station number of PLC is "5"			
	PLC name : MODBUS RTU			
	Address : 4x • 5#111			
	b. The PLC station no. is decided by var7 (LW-10007)			
	PLC name : MODBUS RTU			
	Address : 4x var7#111			
	c. PLC address is set to "111", since PLC station no. is not specified,			
	and the default station no. is using var3, the PLC station no. is			
	decided by var3 (LW-10003).			
	Address (MODBUS RTU			
	Address : 4x			
Use broadcast	This is for setting the station no. of broadcast command. Command			
command	for the users of this set station no. will be seen as broadcast			
	command. For example, if the broadcast station number is set as			
	255, HMI with an address such as 255#200, will send this command			
	to all the PLC connected to it, but will ignore the replies of PLC after			
	receiving this command. (This only works on Modbus).			
	Broaucast station no. : 255			
Interval of	If the interval between read addresses of different commands is less			
block pack	than this value, these commands can be combined to one. But			
(words)	combining function is disabled if this value is "0".			
	For example, the interval value is set as "5" and users would like to			
	read out 1 word from LW3 and 2 words from LW6 respectively.			
	(Means to read from LW6 to LW7). Since the interval of addresses			
	between LW3 and LW6 is less than 5, these two commands can be			
	combined to one. The contents of combination therefore become 5			
	consecutive words from LW3 (read from LW3~LW7).			
	Note: Maximum command combination data size must be less than			
	[Max. read-command size].			
Max.	The Max. data size to be read out from device at one time. Unit: word			



read-command size (words)	
Max.	The Max. data size to be written to device at one time. Unit: word.
write-command	
size (words)	

After all settings are completed, a new device named "Local PLC 1" is added to the [Device list].

Device list :					
No.	Name	Location	Device type	Interface	I/F Protoc
Local HMI	Local HMI	Local	MT8121T (800 x 600)	Disable	N/A
Local PLC 1	MITSUBISHI FXO	Local	MITSUBISHI FX0n/	COM1(9600,E,7,1)	RS485 4 W



## 5.1.2 How to Control a Remote PLC



The so -called "remote PLC" means a PLC connected to a remote HMI. To control a remote PLC, users need to add this type of device. Click **[New...]** under **[Device list**] and the **[Device Properties]** dialog appears. Users need to set all the required properties correctly.

Here take a remote PLC, SIEMENS S7/200, as an example:

Device Properties
Name : SIEMENS S7/200
Location : Remote Settings IP : 192.168.1.35 (Port = 8000)
PLC type : SIEMENS S7/200
V.2.30, SIEMENS_S7_200.so
PLC I/F : RS-485 2W
PLC default station no. : 2
Default station no. use station no. variable
Use broadcast command
COM : COM1
Settings
Interval of block pack (words) : 5
Max. read-command size (words) : 32
Max. write-command size (words) : 32
OK Cancel

Setting

Description





HMI or PLC	This is to confirm whether this device is a HMI or PLC.			
	t is <b>[PLC]</b> in this case.			
Location	Users can select [Local] or [Remote]. Select [Remote] in this case			
	and set the IP address of the remote HMI which is connected to			
	SIEMENS S7/200 PLC. Click [Settings] of [Location] to set this IP			
	address.			
	IP Address Settings			
	⊙ Ethernet			
	IP address : 192 . 168 . 1 . 35			
	Port po : 8000			
	OK Cancel			
	Type of DLC. Select SIEMENS S7/200 in this case			
PLC Type	Type of PLC. Select SIEMENS S7/200 In this case.			
PLC I/F	This setting defines which interface the remote PLC uses. If the			
	remote PLC uses a COM port, interface used should be selected from			
	[RS-232], [RS-485 2W], and [RS485 4W].			
PLC default	This setting defines which default station no. is used by remote PLC.			
station no.				
COM	This setting defines which COM port the remote PLC uses to connect			
	with remote HMI. The settings should be correct.			

After all settings are completed, a new device named "Remote PLC" is added to the **[Device list]**.

Device list :						
	No.	Name Location		Device type	Interface	
	Local	Local HMI	Local	MT8121T (800 x 600) Dis	Disable	
	Local	MITSUBISHI FX0n/FX2 <sub>cal</sub>		MITSUBISHI FX0n/FX2	COM1(9600,E,7,1)	
	Remo	SIEMENS S7/200	Remote(IP:192.168.1	SIEMENS S7/200	COM1(9600,E,8,1)	



## 5.1.3 How to Control a Remote HMI



The so-called "remote HMI" means through network, this HMI is controlled by a local HMI or a PC running on-line simulation. To control a remote HMI, users need to add this type of device. Click **[New...]** under **[Device list]** and the **[Device Properties]** dialog appears. Users need to set all the required properties correctly.

Device Properties
Name : Remote HMI
⊙ HMI O PLC
Location : Remote <u>Settings</u> IP : 192.168.1.11 (Port = 8000)
Interval of block pack (words) : 5
OK Cancel

Setting	Description
HMI or PLC	This is to confirm whether this device is a HMI or PLC.
	It is <b>[HMI]</b> in this case.
Location Users can select [Local] or [Remote]. Select [Remote] in this c	
	and set the [IP address] and [Port no.] of the remote HMI. Click
	[Settings] of [Location] to set these, the dialogue is shown below.



The [Port no.] of remote HMI can be seen in [Model] in [System parameters] once the\* .mtp file of remote HMI is opened. The port no. of remote HMI and local HMI must be the same.

IP Address Settings
⊙ Ethernet
IP address : 192 . 168 . 1 . 11
Port no. : 8000
OK Cancel

After all settings are completed, a new device named "Remote HMI" is added to the **[Device list]**.

No.	Name	Location	Device type	Interface	I/F	St
Local	Local HMI	Local	МТ8хох	N/A	N/A	N/A
Local	MITSUBISHI F	Local	MITSUBISHI F	COM1 (96	RS4	0
Rem	SIEMENS S7/200	Remote(IP:192.168.1.10, P	SIEMENS S7/2	COM1 (96	RS4	2
Rem	Remote HMI	Remote(IP:192.168.1.11, P	MT8xxx	Ethernet	TC	N/A



# 5.2 Model

Parameters in [Model] tab determine the HMI model, [Timer] and [Printer] settings.

System Parameter Settings						
Font	Extended Memory		Printer/Backup Server			
Device	Model G	eneral System	n Setting	Security		
HMI model :	MT8121T (800 × 600)			~		
HMI station no :	0 💌					
Port no. :	8000 (used as MODBUS server's port no.)					
Timer						
Clock source :	External device					
PLC name :	Local HMI Setting					
Address :	LW	✓ 0	16-bit Un	signed		
Printer						
Туре :	SP-M, D, E, F	*				
COM :	СОМ 3 🗸 🗸					
Baud rate :	19200 💌	Data bi	ts : 8 Bits	<b>~</b>		
Parity :	None 💌	Stop bi	ts : 1 Bit	~		
Pixels of width :	100 pixel(s)	Screen hard copy scal	e: 100%	~		
* 100 pixels (for 1610 type) or 220 pixels (for 2407, 4004 type)						
Storage space management						
History data space 4.0M	Max. XOB file size					
*Hint : If change storage space, please reset HMI's data logs and event logs.						

Setting	Description				
HMI model	Select current HMI model as shown below.				
	MT6056T/MT8056T (320 x 234)				
	MT6056T/MT8056T (320 × 234) MT6070T/MT8070T (480 × 234) MT6104T/MT8080T/MT8104T (640 × 480) MT8121T (800 × 600) MT8104X (640 × 480) MT8104XH/MT8121X (800 × 600) MT8150X (1024 × 768) MT6070i/8070i (480 × 234)				
	W18070iH/MT6100i/MT8100i (800 x 480) When changing HMI model and press [OK], users will be inquired if				


	they would like to [Resize pop-up windows or objects].
	Resize pop-up windows/objects
	Resize popula windows
	Resize objects
	Keyboard windows
	Resize function key objects
	OK Cancel
HMI station	Set the [HMI station no.] used by current HMI. If no specific request is
no.	to be made, just use the default number.
Port no.	Set the <b>[Port no.]</b> used by current HMI. It is used as port no. of
	MODBUS server. If no specific request is to be made, just use the
	default number.
Timer	[Clock source]
	To set up the signal for timer object. The time information of timer is
	used by [Data Sampling], [Event Log]etc. which are objects that
	need the time records.
	a. [HMI RIC] means the time signal comes from internal clock of the HMI.
	b. [External device] means the time signal comes from external
	device. To correctly set source address of time signal is necessary.
	Take the illustration below as an example: It indicates the source of
	time signal is from "TV" of the "Local PLC". The source address "TV"
	starts from address 0 contains 6 consecutive words and each of them
	contains the following information:
	TV 0 $\rightarrow$ Second (the limited range: 0~59)
	TV 1 $\rightarrow$ Minute (the limited range: 0~59)
	TV 2 $\rightarrow$ Hour (the limited range: 0~23)
	$IV 3 \rightarrow Day$ (the limited range: 1~31)
	$IV 4 \rightarrow$ Month (the limited range: 1~12)
	$IV 5 \rightarrow$ Year (the limit range: 1970~2037)



	~ Timer
	Clock source : External device
	PLC name : Setting
	Address : TV V 0 16-bit Unsigned
	Address
	PLC name : MITSUBISHI FX0n/FX2
	Device type : TV
	Address : 0
	Address format : DDD [range : $0 \sim 255$ ]
	Index register
	16-bit Unsigned 🔽
Printer	[Type]
	Display printers supported. For HP PCL Series, it has to be connected
	through USB interface while other printers through COM port. For more
	information, please refer to "Chapter 23 Printer Types supported by
	МТ8000".
	Type : HP PCL Series (USB) V SB only
	Paper size : SP.M.D.F.F.
	EPSON ESC/P2 Series
	Using [COM] port to connect printer, users should set accurate
	parameters. When the type of printer is [SP-M, D, E, F], the [pixels of
	width has to be set accurately, i.e. the set pixel(s) can not exceed
	printer's default setting. Otherwise this printing won't succeed.
	Type : SP-M, D, E, F
	СОМ : СОМ 3 🛛 🖌
	Baud rate : 19200 💙 Data bits : 8 Bits 💙
	Parity : None 💙 Stop bits : 1 Bit 💙
	Pixels of width : 100 pixel(s) Screen hard copy scale : 100%
	* 100 pixels (for 1610 type) or 220 pixels (for 2407, 4004 type)
Storage	1. Storage space available for the project and history data is 12MB. By
space	adjusting the space of these two parts, users can reach their





management	memory requirements, for example, using smaller sized project to						
( For T series	get bigger memory space for historical data. It works contrariwise.						
only)	<ol> <li>Minimum Project size is 6MB; Maximum Project size is 10 MB (default is 8MB). Minimum Historical data size is 2MB; Maximum Historical data size is 6 MB (default is 4MB).</li> <li>For adjusting standard and a size is 6 mB (default is 4MB).</li> </ol>						
	in HMI before downloading project file.						
	Storage space management						
	History data space 4.0M  *Hint : If change storage space, please reset HMI's data logs and event logs.						



# 5.3 General

Parameters in [General] tab determine all properties related to screen display.

Font		Exter	nded Memory		Printer/Ba	ackup Server	
Device	Mod	el	General	S.	ystem Setting	Security	ŗ
Fast selection by	atton .						
	Attribute :	Enable	× (	Settin	gs		
	Position :	Left	<b>~</b>	Hide butt	on when HMI star	ts	
Screen saver —							
Back	a light saver :	1	💌 mir	rute (s)			
	i	Enable ba	ck light when als	um occurs			
s	creen saver :	2	💌 mir	rute (s)			
Saver	window no. :	10. WINDO	W_010				~
Options							
Startup v	window no. :	10. WINDO	W_010				¥
Comm	on window :	Above base	window 🔽	Object	t layout : Nature		~
Keyboard	l caret color :				💌 R W_	A enabled	
Event		_					
		🔽 Use L W9	9450~9455 as tin	ne tags of e	vent logs		
		BCD	*				
Extra, n	o. of events :	0					
Keyboard —							
		50. Keypad 51. Keypad 52. Keypad 53. Keypad 54. Keypad 55. Keypad 56. Keypad	1 - Integer 2 - Integer 3 - Integer 4 - Integer 5 - Integer 6 - Integer 7 - HEX			Add Delete	
Project protectio	on (i series onl	y) —					
<b>V</b>	Enable	Proje	ct key : 11111		(range : 0 ~ 42	94901750)	

Setting	Description
Fast	Setting all the attributes for fast selection button that is designated as
selection	window number 3.
button	
	a. [Attribute]



	Enable Disable Enable						
	Enable or disable fast selection window. Select [Enable] and click [Settings] to set the attributes, including color and text. b. [Position]						
	Left Left Right						
	Select the position on the screen of HMI where this button appears. If <b>[Left]</b> is chosen, the button will show up on screen bottom-left; if <b>[Right]</b> is chosen, the button will show up on screen bottom-right.						
Screen	a. [Back light saver]						
saver	If the screen is left untouched and reaches the time limit set here, back light will be off. The setting unit is minute. Back light will be on again once the screen is touched. If <b>[none]</b> is set, the back light will always be on while using.						
	<ul> <li>b. [Screen saver]</li> <li>If the screen is left untouched and reaches the time limit set here. The current screen will automatically switch to a window assigned in [Saver window no.]. The setting unit is minute. If [none] is set, this function is disabled.</li> </ul>						
	c [Saver window no ]						
	To assign a window for screen saver						
Option	a. [Startup window no.]						
	Designate the window shown when start up HMI.						
	b. [Common window]						
	Above base window Below base window Above base window						
	The objects in the common window (window 4) will be shown in each						
	base window. This selection determines the layers these objects are						
	placed above or below the objects in the base window.						
	c. [Keyboard caret color]						



Set the color of caret that appears when inputting in [Numeric Input] and [Word Input] objects.

### d. [Object layout]

Nature	*
Control	
Nature	

If **[Control]** mode is selected, when operating HMI, [Animation] and [Moving Shape] objects will be displayed above other kinds of objects neglecting the sequence that the objects are created. If **[Nature]** mode is selected, the display will follow the sequence that the objects are created, first created be displayed first.

# e. [RW\_A enabled]

Enable or disable recipe data RW\_A. Enable this, the objects can then control the content of RW\_A .The size of RW\_A is 64K.

Event	[Extra no. of events]
	The default number of the event in the system is 1000. If users would like
	to add more records, the setting value can be modified up to 10000.
Keyboard	Users can select to use different types of keyboards for [Numeric Input]
	and [Word Input]. Up to 32 keyboards can be added. If users want to
	design their own keyboard, a window should be designated for creating
	it. Press [add] after creating, and add the window to the list. For more
	information, please see "Chapter 12 Key Pad Design and Usage" where
	also shows how to fix this keyboard in screen instead of adding it to the
	list.
Project	User's project can be restrained and executed on specific HMI (only for i
protection (i	series HMI). Please refer to "Chapter 30 Project protection" for more
series only)	information.



# 5.4 System Setting

Parameters in **[System Setting]** tab are for setting up some miscellaneous functions of EasyBuilder.

Font       Extended Memory       Printer/Backup Server         Device       Model       General       System Setting       Security         Startup language after redownloading the project :       Language 1       Image: Constraint of the project is an investigation of the project is a disconnection icon on relative objects when PLC communication fails         VNC server       Prohibit password remote-write operation (or set LE9053 on)         Prohibit password remote-read operation (or set LE9054 on)       Intra intra intervent is a disconnection icon on relative objects when PLC communication fails         VNC server       Password is intra intra intervent intra intervent is an intervent intervent intervent intervent intervent is a intervent interve	ystem Parameter Se	ttings				
Device       Model       General       System Setting       Security         Startup language after redownloading the project :       Language 1       Image: Security         Startup language after redownloading the project :       Language 1       Image: Security         Execute init. MACRO when power on       Auto logout       Image: Security         Years       Enable       1       Image: Security         Years       Enable       1       Image: Security         Auto logout       Image: Security       Image: Security         Years       Enable       1       Image: Security         When a user does not operate the HMI for longer than the setting time, the system will automatically logout.       Image: Security         Hide system setting bar       Hide mouse cursor       *         * Use LB-9062 to open hardware setting dialog.       Image: Security       Image: Security         Disable buzzer       Prohibit remote HMI connecting to this machine       Image: Security         Disable upload function (effective after rebooting HMI)(or set LB9033 on)       Prohibit password remote-write operation (or set LB9054 on)         Use a disconnection icon on relative objects when PLC communication fails       Image: Security       YmC server         YMC server       Password : Itititit       Ititit       Ititit       Ititit </th <th>Font</th> <th></th> <th>Extended Memory</th> <th></th> <th>Printer/Bac</th> <th>kup Server</th>	Font		Extended Memory		Printer/Bac	kup Server
Startup language after redownloading the project : Language 1	Device	Model	General	Sys	tem Setting	Security
Execute init. MACRO when power on         Auto logout         ✓ Enable       1	S	tartup langu:	age after redownloading	the project	: Language 1	~
Auto logout         Y Enable       1       minute(s)         When a user does not operate the HMI for longer than the setting time, the system will automatically logout.         Hide system setting bar       Hide mouse cursor         * Use LB-9062 to open hardware setting dialog.         Disable buzzer       Prohibit remote HMI connecting to this machine         Disable buzzer       Prohibit remote HMI connecting to this machine         Disable upload function (effective after rebooting HMI)(or set LB9033 on)         Prohibit password remote-read operation (or set LB9054 on)         Use a disconnection icon on relative objects when PLC communication fails         VNC server         Y Password from project         Password from project         Password i from project         Password i communication         L W range : [0]       Y Page         W protection         W protection         R W range : [0]       Y Page         Poisable R W remote-write (via COM port or ethernet)         R W range : [0]       Y Page         EasyAccess server       Y Y Page	Execute init. M.	ACRO when	poweron			
✓ Enable       1       ✓ minute (s)         When a user does not operate the HMI for longer than the setting time, the system will automatically logout.         □ Hide system setting bar       □ Hide mouse cursor         * Use LB-9062 to open hardware setting dialog.         □ Disable buzzer       □ Prohibit remote HMI connecting to this machine         □ Disable buzzer       □ Prohibit remote HMI connecting to this machine         □ Disable upload function (effective after rebooting HMI)(or set LB9033 on)       □ Prohibit password remote-read operation (or set LB9054 on)         □ Use a disconnection icon on relative objects when PLC communication fails         ✓ VNC server       ✓ Password from project         ✓ Password from project       Password : 111111         LW range : 0       ~ 99         ■ Disable RW remote-write (via COM port or ethernet)       RW range : 0         W range : 0       ~ 999	Auto logout ——					
When a user does not operate the HMI for longer than the setting time, the system will automatically logout.     Hide system setting bar   Hide mouse cursor   * Use LB-9062 to open hardware setting dialog.   Disable buzzer   Prohibit remote HMI connecting to this machine   Disable upload function (effective after rebooting HMI)(or set LB9033 on)   Prohibit password remote-read operation (or set LB9053 on)   Prohibit password remote-write operation (or set LB9054 on)   Use a disconnection icon on relative objects when PLC communication fails   VNC server   Password from project   Password from project   Password :   111111   LW range :   0   99      EasyAccess server   Ver login EasyAccess server	🗹 Enable	1	💌 minute(s)			
<ul> <li>Hide system setting bar ☐ Hide mouse cursor</li> <li>* Use LB-9062 to open hardware setting dialog.</li> <li>Disable buzzer ☐ Prohibit remote HMI connecting to this machine</li> <li>Disable upload function (effective after rebooting HMI)(or set LB9033 on)</li> <li>Prohibit password remote-read operation (or set LB9053 on)</li> <li>Prohibit password remote-write operation (or set LB9054 on)</li> <li>Use a disconnection icon on relative objects when PLC communication fails</li> <li>VNC server</li> <li>✓ Password from project Password : 111111</li> <li>LW protection</li> <li>✓ Disable LW remote-write (via COM port or ethernet) L W range : 0 ~ 99</li> <li>RW protection</li> <li>✓ Disable RW remote-write (via COM port or ethernet) R W range : 0 ~ 999</li> <li>EasyAccess server</li> <li>✓ Login EasyAccess server</li> <li>✓ Wow ihmi.net</li> </ul>	When a user does : logout.	not operate ti	he HMI for longer than t	he setting t	time, the system w	ill automatically/
<ul> <li>Disable buzzer □ Prohibit remote HMI connecting to this machine</li> <li>Disable upload function (effective after rebooting HMI)(or set LB9033 on)</li> <li>Prohibit password remote-read operation (or set LB9054 on)</li> <li>Prohibit password remote-write operation (or set LB9054 on)</li> <li>Use a disconnection icon on relative objects when PLC communication fails</li> <li>VNC server</li> <li>✓ Password from project Password : 111111</li> <li>LW protection</li> <li>✓ Disable LW remote-write (via COM port or ethernet)</li> <li>LW range : 0 99</li> <li>RW protection</li> <li>✓ Disable RW remote-write (via COM port or ethernet)</li> <li>RW range : 0 999</li> </ul>	Hide system set * Use LB-9062 to	ting bar open hardwa	Hide mo are setting dialog.	use cursor		
<ul> <li>□ Disable upload function (effective after rebooting HMI)(or set LB9033 on)</li> <li>□ Prohibit password remote-read operation (or set LB9053 on)</li> <li>□ Prohibit password remote-write operation (or set LB9054 on)</li> <li>□ Use a disconnection icon on relative objects when PLC communication fails</li> <li>- VNC server</li> <li>✓ Password from project</li> <li>Password : 111111</li> <li>- LW protection</li> <li>✓ Disable LW remote-write (via COM port or ethernet)</li> <li>LW range : 0</li> <li>~ 99</li> <li>- RW protection</li> <li>✓ Disable RW remote-write (via COM port or ethernet)</li> <li>RW range : 0</li> <li>~ 999</li> <li>- EasyAccess server</li> <li>✓ Login EasyAccess server</li> <li>✓ Wrww.ihmi.net</li> </ul>	Disable buzzer		📃 Prohibit	remote Hł	MI connecting to	this machine
<ul> <li>Prohibit password remote-read operation (or set LB9053 on)</li> <li>Prohibit password remote-write operation (or set LB9054 on)</li> <li>Use a disconnection icon on relative objects when PLC communication fails</li> <li>VNC server</li> <li>✓ Password from project</li> <li>Password : 111111</li> <li>LW protection</li> <li>W range : 0 ~ 99</li> <li>RW protection</li> <li>RW range : 0 ~ 999</li> <li>EasyAccess server</li> <li>✓ Login EasyAccess server</li> <li>✓ Www.thmi.net</li> </ul>	📃 Disable upload	function (eff	ective after rebooting Hi	4I)(or set l	LB9033 on)	
<ul> <li>Prohibit password remote-write operation (or set LB9054 on)</li> <li>Use a disconnection icon on relative objects when PLC communication fails</li> <li>VNC server</li> <li>✓ Password from project</li> <li>Password : 111111</li> <li>LW protection</li> <li>✓ Disable LW remote-write (via COM port or ethernet) LW range : 0 ~ 99</li> <li>RW protection</li> <li>✓ Disable RW remote-write (via COM port or ethernet) RW range : 0 ~ 999</li> <li>EasyAccess server</li> <li>✓ Login EasyAccess server</li> </ul>	📃 Prohibit passwo	rd remote-re	ad operation (or set LB9	053 on)		
<ul> <li>□ Use a disconnection icon on relative objects when PLC communication fails</li> <li>VNC server</li> <li>☑ Password from project</li> <li>Password : 111111</li> <li>LW protection</li> <li>☑ Disable LW remote-write (via COM port or ethernet) LW range : 0 ~ 99</li> <li>RW protection</li> <li>☑ Disable RW remote-write (via COM port or ethernet) RW range : 0 ~ 999</li> <li>EasyAccess server</li> <li>☑ Login EasyAccess server</li> <li>W www.ihmi.net</li> </ul>	📃 Prohibit passwo	rd remote-w	rite operation (or set LB9	9054 on)		
VNC server            ✓ Password from project             LW protection             ✓ Disable LW remote-write (via COM port or ethernet)             LW range : 0             Password remote-write (via COM port or ethernet)             RW protection             Password remote-write (via COM port or ethernet)             RW protection             Possable RW remote-write (via COM port or ethernet)             RW range : 0             PossyAccess server             Login EasyAccess server	📃 Use a disconnec	tion icon on	relative objects when PL	C commu	nication fails	
<ul> <li>✓ Password from project</li> <li>Password : 111111</li> <li>LW protection</li> <li>✓ Disable L W remote-write (via COM port or ethernet) L W range : 0 ~ 99</li> <li>RW protection</li> <li>✓ Disable R W remote-write (via COM port or ethernet) R W range : 0 ~ 999</li> <li>EasyAccess server</li> <li>✓ Login EasyAccess server</li> </ul>	-VNC server					
L W protection ✓ Disable L W remote-write (via COM port or ethernet) L W range : 0 ~ 99 - R W protection ✓ Disable R W remote-write (via COM port or ethernet) R W range : 0 ~ 999 - EasyAccess server ✓ Login EasyAccess server www.ihmi.net	Password from	project	Passw	ord : 111	111	
<ul> <li>✓ Disable L W remote-write (via COM port or ethernet) L W range : 0 ~ 99</li> <li>R W protection</li> <li>✓ Disable R W remote-write (via COM port or ethernet) R W range : 0 ~ 999</li> <li>EasyAccess server</li> <li>✓ Login EasyAccess server</li> <li>www.ihmi.net</li> </ul>	LW protection					
L W range : 0 ~ 99 - R W protection Disable R W remote-write (via COM port or ethernet) R W range : 0 ~ 999 - EasyAccess server Login EasyAccess server www.ihmi.net	🔽 Disable L W ren	note-write (v	ia COM port or ethernet)			
R W protection Disable R W remote-write (via COM port or ethernet) R W range : 0 ~ 999 - EasyAccess server V Login EasyAccess server www.ihmi.net			LW range : O	~ 99		
Disable R W remote-write (via COM port or ethernet) R W range : 0 ~ 999 EasyAccess server Login EasyAccess server	- RW protection -					
R W range : 0 ~ 999 - EasyAccess server V Login EasyAccess server www.ihmi.net	Disable R W ren	note-write (v	ia COM port or ethernet			
EasyAccess server			RW range : 0	~ 999	)	
✓ Login EasyAccess server www.ihmi.net	– EasyAccess server –					
	🗹 Login EasyAcc	ess server	www.ihmi.net			

Some functions are duplicated from system tag, such as [Disable buzzer (LB-9019)], [Hide system setting bar (LB-9020)], [Hide mouse cursor (LB-9018)], [Disable upload function (LB-9033)], and [Prohibit remote HMI connecting this machine (LB-9044)]. It means that

user can also operate these functions via system tag. To select a system tag, users can tick **[system tag]** of the **[address]** while adding new object. To check all the system tags, users can visit **[Library]** in EB8000, select **[Tag]** then **[System]**.

#### [Startup language after redownloading the project]

Set the language to use when start up HMI after redownloading the project.

#### [Execute init. Macro when power on]

Designate the macro to be executed when HMI power on.

#### [Auto logout]

If HMI is left unused for longer than the time set here, HMI will logout automatically.

#### [Use a disconnection icon on relative objects when PLC communication fails]

When using this function and fail to communicate with PLC, this icon will be shown in the lower right corner of the object as shown:

The disconnection icon :



When using this function and fail to communicate with PLC, this icon will be shown in the lower right corner of the object as shown:



#### [VNC Server]

Set the login password for VNC server.

#### [LW protection], [RW protection]



If users check **[Disable LW/RW remote-write]** and set the protect range in **[LW/RW range]**, values of this protected range can't be adjusted via remote HMI.

### [Easy Access server]

Through this technology, users can easily access to any MT8000i/X connected to the internet and operate them on PC just like holding touch screen in hand. Unlike most server used in HMI, Easy Access don't need to transmit updated graphic image but real time data only. This makes transmission really quick and efficient. For further information, please refer to "*EasyAccess*".



## 5.5 Security

Parameters in **[Security]** tab determine the classes accessible for each user to adjust the objects, and users' password. The security classes of objects are classified from **[A~F]**, and **[none]** for not ticking any class. Up to twelve passwords can be set. Only numeral setting is acceptable for password and the range is 0~999999999.

System Parameter Se	ettings			
Font	Exten	ded Memory	Printer/Bac	kup Server
Device	Model	General	System Setting	Security
-Select operatable cla	isses for each user			
Enable	Password : 0	🗹 A	□B □C □D	E F
User 2 Enable	Password : 222	A	✓B □C □D	E F
User 3 Enable	Password : 333	A		E F
User 4				
-User 5				
User 6 Enable				

According to the security setting, EB8000 will control the classes accessible for each user to adjust the objects once they input their passwords.

In EB8000, while constructing a project, the security classes of objects are classified from **[A~F]**, and **[None]** and can be set as shown below.

If **[None]** is set, every user can access to adjust this object.

User restriction —		
Object class :	None 🗸	
	None	
	Class A Class B	
	Class C	
	Class D	
a 1	Class E Class F	



For example, when the security class of User1 is set as below, only objects with class A, C, E and "none" can the user adjust. For more information, please see "Chapter 10 Security of Objects".

- User 1									-
🗹 Enable	Password :	1111	🗹 A	B	✓C	D	🖌 E	F	

#### [Project password (MTP file)]

Users can set password to protect the MTP file in [System parameter] / [Security tab].

Users have to input the password set here when they want to edit the MTP file.

(MTP password range: 1~4294967295)

Tick [Enable] then click [Setting], and the window is as shown below.

ser 11	
Enable Password : 198545	561 (range: 1 ~ 4294967295)
Jser 12 OK	Cancel

Before editing project, a pop-up window will ask password for access the project.

<u>File T</u> ool <u>H</u> elp		
	EasyBuilder8000	3
<b>5, 5, 15, 16</b> , <b>16</b> ,	Password : *** (Limited to 3	times)
Kindows	OK Cance	- <b>-</b>



MTP files are protected by additional encryption, please follow the steps below:

- a. EB8000 V440 or later can open old version EB8000 projects using the password originally set in old version EB8000.
- b. The old version EB8000 can't open the projects that are built in EB8000 V440 or later which are protected by password, if necessary, please disable (don't tick [Enable]) the password first.



# 5.6 Font

Parameters in [Font] tab determine the font of non-ASCII which is used in EB8000.

stem Parameter Settings				
Device	Model	General	System Setting	Security
Font		Extended Memory	Printer/Bacl	kup Server
Fonts for non-asc Albany Arial Arial Black Arial Black Arial Bold Arial Narrow Arial Narrow Arial Narrow Bookman Old S Calibri Century Gothic Comic Sans MS Consolas Courier New Franklin Gothic Garamond Georgia Gulim Impact Latha Lucida Console Malgun Gothic Mangal	ii strings old tyle Medium		Add Delete Delete All	

#### [Fonts for non- strings]

Fonts for non-ASCII strings are listed above. When users use non-ASCII character set or double byte character set (including simplified or traditional Chinese character, Japanese, or Korean) which is not listed in **[Fonts for non-ASCII strings]** table, EB8000 will select a font from the list to substitute for it automatically.

Users can also test which non-ASCII strings of Windows can be used in EB8000 and add them to **[Fonts for non-ASCII strings]** table.

#### [Line spacing]

Decide the interval between lines in the text.





# 5.7 Extended Memory

Device	Model	General	System Setting	Security
Font		Extended Memory	Printer	Backup Server
EM0				
File name :	em0.emi	🔘 SD can	d 💿 USB 1	O USB 2
EM1				
File name :	em1.emi	🔘 SD can	d 💿 USB 1	O USB 2
EM2				
File name :	em2.emi	🔘 SD can	d 💿 USB 1	O USB 2
ЕМ3				
File name :	em3.emi	SD can	d 💽 USB 1	O USB 2
EM4				
File name :	em4.emi	SD can	d 💽 USB 1	O USB 2
EM5				0.7777.0
File name :	em5.em1	SD can	d 💽 USB 1	O USB 2
EM6				
File name :	emb.emi	SD can	d 💽 USB 1	O USB 2
EM7				
File name :	em/.emi	SD can	a 💽 USB 1	O 02B 2
EM8				
File name :	em8.em1	SD can	a 💽 USB 1	O 02B 2
EM9				
File name :	em9.emi	🔾 SD can	d 💽 USB 1	O USB 2

Parameters in **[Extended Memory]** tab determine the path of the extended memory.

Extended Memory is numbered from EM0 to EM9. Method to use extended memory is similar to that of other device type (i.e. LW or RW address). Users can simply select from **[Device type]** list while adding a new object. Size of each extended memory is up to 2G word.





Data in extended memory is stored in **[SD card]**, **[USB1]**, or **[USB2]** in a form of a file. The files in extended memory **[EM0]** ~ **[EM9]** are entitled as em0.emi~em9.emi. Users can use **RecipeEditor.exe** to open the file and edit the data in the extended memory.

Data in extended memory will not be erased when power is cut, which means next time when user start HMI again, data in extended memory remains just the same before power off. This is similar to Recipe data (EW, RW\_A). What is different is that users can select where they want to save the data (SD card, USB1 or USB2)

To read data in extended memory from a removed device, the content of data will be viewed as "0"; if users would like to write data to a removed device, the "PLC no response" message will appear in HMI.

EB8000 supports "hot swapping" function for SD card and USB devices. Users can insert or remove the device for extended memory without cutting the power. With this function, users can update or take data in extended memory.



# 5.8 Printer/Backup Server

Parameters in [Printer/Backup Server] tab are for setting up MT remote printer.

ystem Parameter So	ttings			
Device	Model	General	System Setting	Security
Font		Extended Memory	Printer/Baa	skup Server
✓ Use MT Remo Note: Use EasyPir Output settings —	te Printer/Backup ater to configure l	Server PC for printing screen har	dcopy and storing backup	data.
Orientatio:	n : 💿 Horizont	al 🔺 🔿 Vertical	$\checkmark$	
Printer siz	e : 💿 Original :	size 🚫 Fit to pri	nter margins	
Marg	in :	0 🤤 mm		
	0	mm       Image: mm	0 🛟 mm	
Communication s	ettings			
IP addres	s: 127 .	0.0.1		
Po	rt: 8005			
User nan	ne : 111111			
Passwoi	d : 111111			

Setting	Description
Output settings	[Orientation]
	Set how will words or pictures be printed out, [horizontal] or
	[vertical].
	[Printer size]
	Set to print out in original size or to fit the set printer margins.
	[Margin]
	Set the top, bottom, right and left margin width.
Communication	[IP address]
settings	Assign the IP address of a remote printer via network.



[Port], [User name], [Password]
Assign the access information.
Port can be set from 1 to 65535.
Maximum length of user name or password is 12 characters.

\* Please refer "Chapter 26 Easy Printer" for more information.



# **Chapter 6 Window Operations**

The basic component of a HMI screen is a Window, This shows its importance. With a window, all kinds of information like objects, pictures, and words can be shown in HMI screen. Generally, there's more than one window in a project, many windows will be constructed in one project. Users are able to configure 1997 windows or screens numbered from 3~1999 in EB8000. For how many windows can be used in one project, it depends on the storage size for windows of HMI. For example, the storage size of MT8000 i series for windows is 16MB, then the size of windows or screens constructed cannot exceed 16MB. Under this limit users can make most use of it to create as many windows as possible.

## 6.1 Window Types

There are 4 types of windows in EB8000 each with different functions and usages.

- a. Base Window
- b. Common Window
- c. Fast Selection Window
- d. System Message Window

### 6.1.1 Base Window

Base window is the most frequently-used type of window. Apart from being used as main screen, it is also used as:

- a. Foundation base: used as the background for other windows
- b. Keyboard window
- c. Pop-up window for [function key] object
- d. Pop-up window for [direct window] and [indirect window] object
- e. Screen saver



Base window should be in the same size as the HMI screen. That is to say, the resolution of base window and that of HMI should be identical.

The start up screen is a base window and is shown below:

COM CPU PWR		
<complex-block><complex-block>  Image: Security   Image: Security<!--</th--></complex-block></complex-block>		

### 6.1.2 Common Window

Window no. 4 is the default common window. Objects in this window will be displayed in other base windows, but it does not include popup window. Therefore, objects in different windows, whether shared or same, will be placed in common window, for example, the logo of the product, or a common button. When system is in operation, Clicking **[Function Key]** and selecting **[Change common window]** allow users to change the source of common window. For example, users can change the common window from window 4 to window 20.



New Function Key Object	×
General Security Shape Label	
Description :	
Activate after button is released	
Change full-screen window 💿 Change common window	
🔿 Display popup window	
Window no.: 20. Window_020	~

# 6.1.3 Fast Selection Window

Window no. 3 is defined as the Fast Selection Window. This window can coexist with base window. Generally speaking, it is used to place the frequently-used operation buttons as the picture below:



When using Fast Selection Window, window no. 3 should be created first, and then users need to set each function of Fast Selection button. The **[Fast Sel]** button in the picture above is the Fast Selection button, which is used to Enable/Disable Fast Selection window



control. Every setting of the **[Fast Selection button]** is in **[System Parameter Settings]**. Please refer to the dialog below.

System Parameter Settings				
Font		Extended Memory	Printer/B	ackup Server
Device	Model	General	System Setting	Security
Fast selection button       Attribute :       Position :       Left			starts	

Apart from Enable/Disable Fast Selection window by Fast Selection button, system register also provides the following addresses for users to Enable/Disable certain functions in order to control fast selection window/button. The related registers are listed below. Please refer to "Chapter 22 system reserved words and bits" for more details.

[LB-9013] FS window control [Enable (open) / Disable (close)]

[LB-9014] FS button control [Enable (open) / Disable (close)]

[LB-9015] FS window / button control [Enable (open) / Disable (close)]

## 6.1.4 System Message Window

Window no. 5~8 are the defaults of system message windows.

Window	Description
Window no. 5 is the "PLC	When the communication between PLC and HMI is
Response" message	disconnected, this message window will pop up
window	automatically right on the window opened previously.
Window 6 is the "HMI	When failing to connect with remote HMI, this message
connection" message	window will pop up automatically.
window	
Window 7 is the "Password	If user wants to control an object without authorization,
Restriction" message	this window may pop up as an alert or not depending on
window	how this object is set originally.



Window 8 is the "Storage	When HMI built-in memory, USB disk or SD card run out
Space Insufficient"	of storage space, this message window will pop up
message window	automatically.
	Users can use system address tag to view the free
	memory space in HMI, USB disk, or SD card device.
	[LW-9072] HMI current free space (K bytes)
	[LW-9074] SD current free space (K bytes)
	[LW-9076] USB 1 current free space (K bytes)
	[LW-9078] USB 2 current free space (K bytes)
	For checking which device is insufficient in space while
	this insufficiency occurs, the following system address
	tags can be used.
	[LB-9035] HMI free space insufficiency alarm (when ON)
	[LB-9036] SD free space insufficiency alarm (when ON)
	[LB-9037] USB 1 free space insufficiency alarm (when
	ON)
	[LB-9038] USB 2 free space insufficiency alarm (when
	ON)

The text shown in window no. 5~8 can be adjusted by users to fit what is needed. For example, text in window no. 5 is "PLC No Response", users can change it to "HMI and PLC disconnected!" This works for other windows as well, which makes it easier to read.

#### Note:

- (1) A screen can display 16 pop-up windows simultaneously in maximum including System Message Window, Direct window and Indirect window.
- (2) A window can only be displayed once simultaneously. That is to say, users cannot use2 Direct (Indirect) windows to open the same window in one base window at the same time.
- (3) Windows 0~9 are for system use only while windows 10~1999 are for users to define.

## 6.2 Create, Set, and Delete a Window

The picture below shows the windows information (window tree) in EB8000. This window is always shown on left side of the editing zone. There are 2 ways to check all types of windows in EB 8000. If users change **[Object List]** to **[Window Preview]**, every window will be shown in pictures. The following section introduces how to create and set these windows.





## 6.2.1 Create a Window

There are two ways to create a window:

One is to select a window number in window tree and right click, then select **[New]**. Complete all the settings in the pop-up dialogue and click **[OK]** as shown below:

∓ *10: Μ 11: Sł	lain Menu 10w Toolbar
13:0	New
- 14: F	Open
- 15: S	Close
- 16: C	Delete
17:5	Settings
19	
20: Te	ext Object

indow Settings
Name : Window_011 Window no. : 11
Size Width : 480 Height : 272
Frame Width: 4 Color:
Color : Pattern : Filled Pattern color :
Underlay window
Bottom : None
Middle : None
Top : None
Popup window       Start pos.     X : 0       Y : 0     Monopoly
OK Cancel

Setting	Description
Name	The name shown after window is numbered.
	The principle is to make it easy to read and be remembered. For
	example: "Operate Manually" etc.
	*10: WINDOW_010     11: Operate Manually
Window no.	Number of window. Numbered from 3~1999.
Size	[Width] and [Height] of the window. Generally, the resolution of base
	window and that of HMI is identical. For example, if the HMI used is
	MT6100i, the resolution is 800 * 480. Then the newly built window width
	will be 800 and height 480.
Frame	The <b>[Width]</b> of the frame of the window. Range from 0~16, the default is
	"4".



	The <b>[Color]</b> of the frame of the window. Users can select a color they like from the list, or simply click <b>[Custom]</b> to adjust a self-defined color. If the Width of the frame is set "0", then this setting will be ignored.
Background	[Color]
	I he color of the background of the window.
	The pattern of the background of the window If needed users can
	choose a pattern they like from <b>[pattern style]</b> that pops up after clicking
	button of the pattern.
	Pattern Style
	[Pattern color]
	The color of the pattern.
	[Filled]
	Tick to determine if a window is filled with the color and pattern set for the
Underlay	Bottom] [Middle] [Top]
window	Up to three base windows can be specified as underlav windows for
	each base window, from [Bottom] to [Top]. The objects (but not the
	backgrounds) in underlay windows are displayed in this order in base
	window.
Popup	[X], [Y]
window	Base window can also be used as pop-up window. Use <b>[X]</b> and <b>[Y]</b> to set
	the coordinates indicate where in the screen will this base window pop



up. The origin of the coordinates is the left-top corner of the screen.
[Monopoly]
If the option is checked, when a base window used as a pop-up window appears, users are not allowed to operate other windows before this base window is closed. If a base window is used as a keyboard window, "Monopoly" is automatically enabled.

Another way to create a window is to select **[Window]** from menu in EB8000 and then select **[Open Window]** to open the dialogue. Please refer to the illustration below.



Open Win	dow		
No.	Window name	Size	
No. 4 5 6 7 8 *10 50 51 52 53 54 60 62 63	Window name Fast Selection Common Window PLC Response HMI Connection Password Restriction Storage Space Insufficient WINDOW_010 Keypad 1 - Integer Keypad 2 - Integer Keypad 3 - Integer Keypad 3 - Integer Keypad 5 - HEX ASCII Small ASCII Upper S ASCII Lower S	Size 80,200 320,234 185,73 300,100 256,80 320,234 164,213 198,234 200,170 304,213 306,220 312,130 312,130 312,130	New Settings Delete Open

Window No., Window Name and Size are listed in the [Open Window] dialogue.

Click **[New...]** and choose window type from **[Select Window Style]** dialog. Complete all the settings and click **[OK]**, a new window is created.



Select Window Style
Base Window
Fast Selection
Common Window
Exit

Once the base window is built, its window number sticks with it and can't be changed. But the size, color, and name of the window can still be modified.

### 6.2.2 Window Settings

EB8000 provides three methods to modify window attributes:

a. Right click on the designated window from window tree and select **[Settings]** to open the **[Window Settings]** dialogue to change the window properties.

📩 *10: Main Mere	
11: Show Too	New
- 12	Open
13: Object	Close
14: Function	Delete
15: Security	202020
- 16: Communi	Settings

b. Right click directly in the window without selecting any object and then select [Attribute].
 Or, click in EB8000 menu without selecting any object can also open the [Window Settings] dialogue.



Cut	
Сору	
Paste	
Delete	
	-
Attribute	
Goto desti, window	
	-

c. Select **[Window]** from menu in EB8000 and select **[Open Window]**, a dialogue appears. Designate a window to modify then choose **[Settings]** to open the **[Window Settings]** dialogue.

# 6.2.3 Open, Close and Delete a Window

To open an existing window, not only double click the window No. from the window tree, users can also right click the assigned window from the window tree and choose **[Open]** to open it.

Similarly, to close or to delete an existing window is same as the procedure above .Please note that the window to be deleted has to be closed. That is to say, only a closed window can be deleted.



# Chapter 7 Event Log

"Event log" is used to define the content of an event and the conditions triggering it. In EB8000, this triggered event, also called "alarm", and its processing procedure can be saved to designated places such as HMI memory storage or external memory device. The saved file is with a name in a format as EL yyyymmdd.evt. In this name, yyyymmdd records the time that this file is built, and will be set automatically by the system. Take file name EL\_20100524.evt as an example, this shows that this created file records the event occurred on 24<sup>th</sup> of May, 2010.

EB8000 also provides the following system address tags to manage the event log:

Address	Description
LB-9021	reset current event log (set ON)
LB-9022	delete the earliest event log file on HMI memory (set ON)
LB-9023	delete all event log files on HMI memory (set ON)
LB-9024	refresh event log information on HMI memory (set ON)
LB-9034	save event/data sampling to HMI, USB disk, SD card (set ON)
LB-9042	acknowledge all alarm events (set ON)
LB-9043	unacknowledged events exist (when ON)
LB-11940	delete the earliest event log file on SD card (set ON)
LB-11941	delete all event log files on SD card (set ON)
LB-11942	refresh event log information on SD card (set ON)
LB-11943	delete the earliest event log file on USB 1 (set ON)
LB-11944	delete all event log files on USB 1 (set ON)
LB-11945	refresh event log information on USB 1 (set ON)
LB-11946	delete the earliest event log file on USB 2 (set ON)
LB-11947	delete all event log files on USB 2 (set ON)ON)
LB-11948	refresh event log information on USB 2 (set ON)
LW-9060	(16bit) : no. of event log files on HMI memory
LW-9061	(32bit) : size of event log files on HMI memory
LW-9450	(16bit) : time tag of event log - second
LW-9451	(16bit) : time tag of event log - minute
LW-9452	(16bit) : time tag of event log - hour
LW-9453	(16bit) : time tag of event log - day



LW-9454	(16bit) : time tag of event log - month
LW-9455	(16bit) : time tag of event log - year
LW-10480	(16bit) : no. of event log files on SD card
LW-10481	(32bit) : size of event log files on SD card
LW-10483	(16bit) : no. of event log files on USB 1
LW-10484	(32bit) : size of event log files on USB 1
LW-10486	(16bit) : no. of event log files on USB 2
LW-10487	(32bit) : size of event log files on USB 2

# 7.1 Event Log Management

With objects like [Alarm Bar], [Alarm Display] and [Event Display], users are able to clearly understand the life cycle of the whole event from happening, waiting for processing, until the alarm stops. Before using these objects, the content of an event has to be defined first.

Click the [Alarm (Event Log)] icon, and the dialog appears as below:



Alarm (Event) Log		
Category : All [D]	×	
No. Category Text Mode Condition Read address Notification address Buzzer		
Enable back light when alarm occurs		
History files		
Save to HMI memory Save to CF card Save to USB 1 Save to USB 2		
Print		
Event trigger time     OHH:MM:SS     OHH:MM     ODD:HH:MM		
✓ Event trigger date ○ MM/DD/YY ③ DD/MM/YY ○ DD.MM.YY ○ YY,	/MM/DD	
New Insert Delete Settings		
Copy Paste Export Import	Exit	



Setting	Description
Category	EB8000 classifies events. All events are divided into categories 0~255.
	[Alarm Bar], [Alarm Display], and [Event Display] can be used to restrain
	which category to display.
	[Category] is for selecting which category of the events to be displayed.
	Category : All [2]
	0[2]
	4 [0]
	12 [0]
	The $[2]$ of $0[2]$ in this illustration domonstrates there are two defined events in
	category 0
	Alarm (Event) Log
	Category : 0 [2]
	No.         Category         Text         Mode         Condition         Read address         Notification address         Buzzer           1         0         Language 1         BIT         ON         LB-0         Disable         Disable
	2 0 Event 1 (When LB=0) BIT ON LB-0 Disable Disable
History	Determine the storage device of an event log. However, when users simulate
files	the project in PC, the files will be saved under the same event log
	subdirectory as EasyBuilder8000.exe.
	[Save to HMI memory]
	Save the event log data in MT8000 memory.
	[Save to SD card]
	Save the event log data in SD card.
	[Save to USB 1]
	Save the event log data in USB disk 1. Numbering rule of USB disk is: the
	disk inserted to the USB interface in the first place is numbered 1, next is
	numbered 2 and the last is numbered 3. It is not related to the interface
	position.
	[Save to USB 2]
	Save the event log data in USB disk 2.
	[Preservation limit]
	After choosing the device to save the Event log, users can see the



	[Preservation limit] selection. This setting determines how many days the data to be preserved. For example, the preservation time is set two days, which means HMI memory will keep the data of yesterday and the day before yesterday. Data that is not built in this period will be deleted automatically to prevent the storage space from running out.							
	[	Preservation limit	Days of pre	eservation : 2	day(s)			
Print	To e para	enable this settin ameter settings	g, users hav ].	e to finish th	ie settings of	f printer in <b>[syst</b>	em	
	Prin	t V Sequence no.						
	_	✓ Event trigger time	O HH:MM:SS	⊙HH:MM	OD:HH:MM			
		Event trigger date	OMM/DD/YY	⊙DD/MM/YY	ODD.MM.YY	OYY/MM/DD		

# 7.1.1 Excel Editing

Alarm (Event) Log	
Category : 🗚 [2]	×

There is an Excel icon in the top-right corner of the **[Alarm (Event Log) dialog]** for users to edit an Event log through Excel. An editing procedure includes: Edit in Excel, Import from Excel to Event Log and Export to Excel.

## A. Edit in Excel

EB8000 provides a standardized sample of Excel in C:\EB8000\EventLogExample.xls for users to edit alarm (event) log. The sample includes some dropdown lists for an easier usage



	А	В	С	D	E	F	G	Н	I	J	K
1	Category	Priority level	Address type	PLC name	Device type	System tag	User-defined tag	Address	Index	Data Format	Enab
2	0	Middle	Word	Local HMI	EMO	False	False	22	null	32-bit Signed	True
3	1	Low	Bit	Local HMI	LB-9009 : initialized as ON	True	True	122	IDX 1	16-bit BCD	False
4	2	High	Word	Local HMI	RWI	False	False	2222	IDX 4	32-bit BCD	<b>√</b> ue
5										16-bit BCD 32-bit BCD	
6										16-bit Unsigned	
7										32-bit Unsigned	
0										52-bit Signed	

#### Caution:

- 1. **[System tag]** and **[User-defined tag]** can not be set true simultaneously. If both of them are set true, the system will view System tag to be true and User-defined tag to be false. If Device type is set as User-defined tag, please set System tag to be false.
- 2. The format of Color is R: G: B. the values of R, G, and B should be integer from 0 to 255.
- 3. Click Excel icon to open EventLogExample.xls



# B. Import from Excel to Event log

Click **[Import excel button]** to import Excel file to Event log.

A	larm	(Event) L	og						
	C	ategory : 4	4II (O)	~				×	
	No.	Category	Text Mod	de Condition	Read address	Notification address	Buzzer		
	~Hist	:ory files —	Dack light	, when didinin	JULUIIS				
		Save to	o HMI mer	mory 🔲 9	Save to CF card	Save to USB	1 🔲 Sa	ave to USB 2	
		Preservation limit			s of preservatior	n : 7 day(	(s)		
	Print								
	Sequence no.								
		✓ Event trigger time		ne OHH	ł:MM:SS 🧿	)нн:мм 🔿 с	D:HH:MM		
		Event trigger date		te 🔾 MN	1/DD/YY 🧿	) DD/MM/YY OC	D.MM.YY		
		New	Ir	nsert	Delete	Settings			

#### Caution:

WE!NTEK

- 1. When user-defined tag is set true in Excel, the system will compare this device type with the user-defined tag in system. If no suitable tag can be found, the system will set the user defined tag in event log to be false.
- 2. Before importing library (label library and sound library), please make sure library names exist in the system, otherwise the system will simply use the file name of the imported excel file.

## C. Export to Excel

Click **[Export excel button]** to export data in Event log to excel.
1			
	WE!	NTE	K

A	Alarm (Event) Log								
Category : All [2]					×				
	No.	Category	Text	Mode	Condition	Read address	Notification address	Buzzer	
	1	0	Event 0	WORD	< 0.00	L₩-0	Disable	Disable	
	2	0	Event 1	BIT	ON	LB-0	Disable	Disable	
1		_							
		🗹 Enable	back lig	ht wher	n alarm occi	urs			
	-Hist	ory files —							
		Save to	o HMI m	emory	Save Save	e to CF card	Save to USB 1	Save to USB 2	
		Preserv	ation lim	nit	Davs of	f preservation :	7 dav(s)		
	Print								
	_	Sequer	nce no.						_
		🗹 Event i	trigger t	ime	OHH:M	M:SS 💿 H	IH:MM ODD	:HH:MM	
		Event :	trigger d	late	OMM/D	D/YY 💿 🛛	D/MM/YY ODD	.MM.YY OYY/MM/DD	
	-	Now		Incort		Delete	Settings		
		C		Deed		Conce	Januart		
		Сору		Paste		Export	Import	Exit	



## 7.2 Create a New Event Log

Click [New...]; [Event Log] dialog appears with two tabs.

## [General] tab:

Alarm (Event) Log
General Message
Category : 0 Priority level : Low
Address type : Word 💌
Read address
PLC name : Local HMI 🗸 Setting
Address : LW V 0 16-bit Unsigned
_ Notification
Enable 🚫 Set ON 💿 Set OFF
PLC name : Local HMI 🗸 Setting
Address : LB 🗸 0
Condition
Trigger if value is : 🗾 🔽 🗸 🗸 🗸 🗸
In tolerance : 0.1 Out tolerance : 0.2

Setting	Description
Category	The category of an event.
Priority level	The priority of an event: Users can define [Low], [Middle], [High], or
	[Emergency] according to the importance of the event. When the
	number of event log equals to the max number available in the system,
	the less important events (lower priority) will be deleted and new events
	will be added in. (the default is 1000, please refer to "General" in
	"Chapter 5 System Parameters" to set this number)
Address	The type of address—[Bit] or [Word] mode.
type	
Read	By reading the address set here, system obtains a value and will use it to
address	check if an event reaches the condition to be triggered. Please refer to



	"Chapter 9 Object General Properties" for more information.		
Notification	When an event is triggered, the specific message is sent out from		
	Notification address. Select [Set ON] to send ON message to this		
	address or select [Set OFF] to send OFF message to this address.		
	Please refer to "Chapter 9 Object General Properties" for detail.		
Condition	The trigger condition of an event. When [Address type] of an event is		
	[Bit], then [ON] or [OFF] in [Trigger] can be selected. The illustration		
	below shows if Trigger <b>[ON]</b> is selected, and the status of [Read address]		
	changes from OFF to ON, an event will be triggered and generate an		
	event log record (or an alarm).		
	Condition		
	OFF-SON ON-SOFF		
	When the [Address type] of an event is [Word], several selections are		
	available as follows:		
	<>>> <>>> <>>> <>>>= <=		
	Under the condition, system will read values from [Read address] and		
	compare them with the trigger conditions to decide if an event is to be		
	triggered. If the trigger condition is set as [==] or [<>], [In tolerance] and		
	[Out tolerance] need be set while [In tolerance] is used as trigger		
	condition and [Out tolerance] is used as system's normal condition.		
	Example 1:		
	Condition		
	In tolerance : 1 Out tolerance : 2		
	The illustration above indicates that if the value of [Read address] is		
	greater or equal to 29(=30-1), or less or equal to 31(=30+1), the event		



will be triggered.
29 <= [Read address] value <=31
After the event is triggered, only when the value of [Read address] is greater than 32(=30+2) or less than 28(=30-2) will the system return to normal condition.
[Read address] value < 28 or [Read address] value >32
Example 2:
Condition         Trigger if value is :       30         In tolerance :       1         Out tolerance :       2
Take another example above, it indicates that the event is triggered when the value of [Read address] is less than $29(=30-1)$ or greater than $31(=30+1)$ .
[Read address] value <29 or [Read address] value >31
When the event is triggered, system returns to normal condition only when the value of [Read address] is greater or equal to 28(=30-2), or less or equal to 32(=30+2).
28 <= [Read address] value <= 32

[Message] tab: Please see the illustration below

N N	E!N	TEK

Alarm (Event) Log	
General Message	
Text	
Content :	test
	Use label library Label Library
Font :	Arial
Color :	
- Write value for Eve	ent/Alarm Display object
Write value :	11
Sound	
Sound	Sound Library Been
	Soum Enlary Deep
Enable con	muous beep until acknowled ged or recovered
	lime delay of continuous beep : 10 second(s)
	Play
Print	
	On trigger Return to normal
Font size :	16
Addresses of WAT	CH1, WATCH2, WATCH3, WATCH4
Multi-watch :	4 Syntax
PLC name :	Local HMI 🗸 Setting
Address :	LW 0 16-bit Unsigned
PLC name :	Local HMI Setting
Address :	LW 0 16-bit Unsigned
PLC name :	Local HMI Setting
Address :	LW 0 16-bit Unsigned
PLC name :	Local HMI 🗸 Setting
Address :	LW 0 16-bit Unsigned

Setting	Description
Text	[Content]
	The text content of event log shown in [Alarm Bar], [Alarm Display]
	and [Event Display]. Please refer to "Chapter 9 Object General
	Properties" for more information.
	The data of LW address of the triggered event can be included in
	the content.
	Format: %#d



	%: initial sign	
	# : LW's address	
	d : end sign	
	For example, if the content is set as "High Temperature = %20d", when an event is triggered, the value of LW20 will be displayed. If the value of LW20 is 13 when an event is triggered, the content displayed in [Event Display] object will be "High Temperature = 13".	
	Except for LW, when an event is triggered, data in certain device type can also be shown in the content. This device type should be the same as that of the [read address] of event log.	
	Format: \$#d	
	\$: initial sign	
	# : PLC's address	
	d : end sign	
	For example, if Device type in Read address is MW, when content is set as "High Temperature = \$15d" and the value in MW15 is 42 while the event is triggered, the displayed content in [Event Display] will be "High Temperature = 42".	
	[Font] [Color]	
	Users can set Font and Color for each event. The font and color of an [alarm display] or [event display] object comes from this setting. As illustration below, these two events use different colors and font styles.	
	1         14/09/07         15:02         Event 1 (when LB1 == 1)           0         14/09/07         15:02         Event 3 (when LW1 = 20)	
Write value for	When an event item in an [event display] object is touched, the	
Event Display	value is written to the assigned address. Please refer to "Chapter 13	
object	Objects" for information about [event display] object.	
Sound	The warning alarm used when an event is triggered can be	
	selected.	
	Click [Sound Library] to choose warning sound, and click [Play] to	



check the sound.

#### [Enable continuous beep until acknowledged or recovered]

Continuous beep can be set which will only stop when the event is acknowledged or recovered. (System register [LB-9042] can be set to acknowledge all alarm events at one time.) The system default [Beep] sound is used as this continuous beep.

When using continuous beep for Event Log, a delay period can be set between triggering the alarm and the start of beeping.





Use the below sys	ntax to embed PLC data in the content of an event log.
%(WATCH#)d.*	Display signed decimal integer
%(WATCH#)f.*	Display floating point
%(WATCH#)s	Display string
%(WATCH#)X	Display unsigned hexadecimal integer, using "ABCDEF."
%(WATCH#)x	Display unsigned hexadecimal integer, using "abcdef."
	where # : watch no., range : 1~4 * : the number of digits after the decimal point If * is 0, ".*" can be ignored.
Examples	
1.Pressure = %	(WATCH1)d.1
2.Temperature	1 is %(WATCH1)f.2, Temperature2 is %(WATCH2)f.2
3.Alarm : IP =	%(WATCH1)X : %(WATCH2)X : %(WATCH3)X : %(WATCH4))
4.Counter is %	6(WATCH3)d
5.Message = 🕅	%(WATCH1)s, Index = %(WATCH3)d



## **Chapter 8 Data Sampling**

"Data Sampling" defines how the data is sampled, including sampling time and sampling address. EB8000 saves the sampled data to the user assigned location.

The directory of saved data: [Storage location]\[filename]\yyyymmdd.dtl

[Storage location] can be HMI, SD (CF) card, USB1 or USB2 which is designated by users.

[Filename] is usually a name defined by user. This name can't be used repeatedly by other sampled data files.

yyyymmdd shows when the file is built and is set by the system automatically.

EB8000 provides the following system registers for data sampling management:

Address	Description
LB-9025	delete the earliest data sampling file on HMI memory (set ON)
LB-9026	delete all data sampling files on HMI memory (set ON)
LB-9027	refresh data sampling information on HMI memory (set ON)
LB-9034	save event/data sampling to HMI, USB disk, SD card (set ON)
LB-11949	delete the earliest data sampling file on SD card (set ON)
LB-11950	delete all data sampling files on SD card (set ON)
LB-11951	refresh data sampling information on SD card (set ON)
LB-11952	delete the earliest data sampling file on USB 1 (set ON)
LB-11953	delete all data sampling files on USB 1 (set ON)
LB-11954	refresh data sampling information on USB 1 (set ON)
LB-11955	delete the earliest data sampling file on USB 2 (set ON)
LB-11956	delete all data sampling files on USB 2 (set ON)
LB-11957	refresh data sampling information on USB 2 (set ON)
LW-9063	(16bit) : no. of data sampling files on HMI memory
LW-9064	(32bit) : size of data sampling files on HMI memory
LW-10489	(16bit) : no. of data sampling files on SD card
LW-10490	(32bit) : size of data sampling files on SD card
LW-10492	(16bit) : no. of data sampling files on USB 1
LW-10493	(32bit) : size of data sampling files on USB 1
LW-10495	(16bit) : no. of data sampling files on USB 2
LW-10496	(32bit) : size of data sampling files on USB 2







## 8.1 Data Sampling Management



Before using [Trend display] or [History data display] objects to review the content of data sampling, the method of how the data is sampled has to be defined. Click **[Data Sampling]** on the toolbar and then **[Data Sampling Object]** dialog appears as below.

D	ata S	ampling Obje	ect					
	No.	Description	Read address	Sample mode	Trigger address	Clear address	Hold address	Auto. stop
1		New	Delete	Setting	gs			Exit

Setting	Description
New	Add a new "data sampling" definition.
Delete	Delete a specific "data sampling" definition.
Settings	Modify a "data sampling" definition.



## 8.2 Create a New Data Sampling

Click [New...] and the [Data Sampling Object] setting dialog appears as below:

Data Sampling Object	
Description : Sampling mode Time-based O Trigger-based Sampling time interval : 1 second(s)	PLC name : Local HMI
Read address         PLC name : Local HMI         Address : LW         Data Record         Max. data records : 1000         Data Format         Data length : 0 word(s)	Hold address
	OK Cancel

#### [Sampling mode]

EB8000 provides two methods of data sampling: [Time-based] and [Trigger-based].

If **[Time-based]** mode is selected, EB8000 samples the data in a fixed frequency. Users have to set the **[sampling time interval]**.



If **[Trigger-based]** mode is selected, users can use the status of specific address to trigger the data sampling.

Sampling mode	d 💽 Trigger-based	Mode : OFF->ON
PLC name :	Local HMI	Setting
Address :	LB 🔽 160	



### [Mode]

Conditions to trigger the data sampling:

- [OFF  $\rightarrow$  ON] This will trigger data sampling when the status of assigned address changes from OFF to ON.
- [ON  $\rightarrow$  OFF] This will trigger data sampling when the status of assigned address changes from ON to OFF.
- $[ON \leftarrow \rightarrow OFF]$  Trigger data sampling when the status of assigned address is changed.

### [Read address]

Select a device type to be the source of data sampling.

### [Data Record]

#### [Max. data records]

Max. number of data records that can be saved by one data sampling definition in one day. If **[sampling time interval]** is set as 0.1 second then the max number of data records is 86400.

- 1. If the data source of **[trend display]** is in **[real-time]** mode, the earlier record will be deleted and new record will be added and displayed in the [trend display] object.
- 2. If the data source of **[trend display]** is in **[historical]** mode, the data will still be sampled.

#### [Auto stop]

When the number of data sampling equals to **[Max. data records]**, and the [Auto stop] option is selected, HMI will stop sampling data automatically.

Condition	Set [Max. data records] as	Set [Max. data records] as 10		
	10 without checking [auto.	and check [auto. Stop]		
	Stop]			
Trend display – real	The data will keep the latest	Stop displaying after reaching		
time	10 records in the screen	10 data records.		
Data sampling	Keep recording and delete	Stop recording after reaching		
	the earlier data	10 data records.		

Example:

#### [Data Format]

The formats of different data in one data sampling: A data sampling may include more than one type of records. Data sampling in EB8000 is able to retrieve different types of



records at the same time. Users can click **[Data Format]** to open the dialog to define the content in one data sampling.

Take the following as an example, user defines three types of data: "Index" (16-bit Unsigned), "Pressure 1" (16-bit Signed) and "Temperature" (32-bit Float) respectively, which makes data length to be 4 words in total. In this way, EB8000 retrieves a 4-words-lengthed data each time from the assigned address to be the content in one data sampling.

Data Format		
0. "index" 1. "Pressure 1" 2. "Temperature"	16-bit Unsigned 16-bit Signed 32-bit Float	New Delete Settings
-		Exit

#### Caution:

After executing off-line simulation, if users need to change data format, please delete data log file in **C:\EB8000\\*\*\*\datalog** and then run off-line simulation again. The symbol "\*\*\*" means the storage location of data files.

#### [Clear address]

If the status of the assigned address is set ON, the obtained data will be cleared and the number of data sampling will go back to zero. This won't affect sampled data that is already saved in file.

Caution: this function is used for [trend display] in [real-time] mode only.

#### [Hold address]

If the status of the assigned address is set ON, sampling will be paused until the status of assigned address returns to OFF.

#### [History files]

Assign the storage location for data sampling. However, when users execute simulation on PC, all data will be saved to the same subdirectory of datalog as EasyBuilder 8000.exe.



#### [Save to HMI memory]

Save the data sampling in MT8000 HMI.

#### Caution:

The data can only be saved when its size reaches 4kb; otherwise, users need to use [LB-9034] to force storing this data.

#### [Save to SD card]

Save the data sampling in SD card.

#### [Save to USB 1]

Save the data sampling in USB disk no.1. Numbering rule of USB disk is: the disk inserted to the USB interface in the first place is numbered 1, next is numbered 2 and the last is numbered 3. It is not related to the interface position.

#### [Save to USB 2]

Save the data sampling to USB disk no.2.

#### [Folder name]

Set the file name of the data sampling. A folder name must be composed entirely of ASCII characters!!

#### [Preservation limit]

This setting determines how many days the data to be preserved.

For example, the preservation time is set two days, which means USB 1 will keep the data of yesterday and the day before yesterday. Data that is not built in this period will be deleted automatically to prevent the storage space from running out.

If today were July 1<sup>st</sup>, the USB 1 will keep the data of June 30<sup>th</sup> and June 29<sup>th</sup> in the memory but the data of June 28<sup>th</sup> will be deleted.

-History files	
Save to HMI memory	Save to CF caud
Save to USB 1	Save to USB 2
Folder name : Pressure	
Preservation limit	Days of preservation : 2 day(s)



# **Chapter 9 Object General Properties**

The contents of [general] properties settings of an object include:

- 1. Selecting the connected PLC.
- 2. Setting reading and writing address
- 3. Using shape library and picture library
- 4. Setting text content
- 5. Adjusting profile size

## 9.1 Selecting PLC

It is required to designate which PLC to operate while using some objects as shown below. **[PLC name]** represents the controlled PLC. In this example there are 2 PLC: "Local HMI" and "Allen-Brandley DF1." These listed available PLC devices are sourced from **[Device List]** in **[System Parameters Settings]**.

PLC name :	Allen-Brandley DF1	*
	Local HMI Allen-Brandley DF1	

## 9.1.1 Setting the Reading and Writing Address



Read address	
PLC name :	Local HMI 🛛 Setting
Address :	LB 🗸 0 IDX 0
Address	×
PLC name :	Local HMI
Device type :	LB
Address :	0 System tag
Address format :	DDDDD [range : 0 ~ 11999]
Index :	INDEX 0 (16-bit) V Index register

The picture above shows a reading address or writing address contains:

## [PLC name]

This is for selecting device type. Different PLC are with different selections of [device type].

Address	X
PLC name :	Local HMI 🗸 🗸
Device type :	LB
Address :	LB RW_Bit
Address format :	RW_A_Bit LW_Bit
Index :	
	OK Cancel

#### [Address]

Set the reading and writing address.

#### [System tag]

Address tag includes "system tag" and "user-defined tag." Click **[Setting...]** beside **[PLC name]** and tick **[system tag]**. This allows users to use the preserved addresses by system for particular purpose.

These address tags are divided into bit or word (LB or LW).

After selecting **[System tag]** not only will the **[Device type]** displays the content of the chosen tag, **[Address]** will also display the register chosen as shown below.





Address	X
PLC name :	Local HMI 🛛 💌
Device type :	LB-9000 : initialized as ON
Address :	LB9000
Address format : (	DDDDD [range : 0 ~ 11999]
	Index register
	OK Cancel

The illustration below shows a part of system tags. For further information, please refer "Chapter 16 Address Tag Library" and "Chapter 22 System Reserved Words and Bits".

Address		×
PLC name :	Local HMI	*
Device type :	LB-9000 : initialized as ON	~
Address :	LB-9000 : initialized as ON LB-9001 : initialized as ON	^
Address format :	LB-9002 : initialized as ON LB-9003 : initialized as ON	=

#### [Index register]

Deciding to use Index register or not, please refer to "Chapter 11 Index Register" for more information.

#### Selecting Data Type

EB 8000 supports data types that are listed below. Selecting correct data type is necessary especially while using address tag.

16-bit Unsigned 🛛 🗸 🗸
16-bit BCD
32-bit BCD
16-bit Hex
32-bit Hex
16-bit Binary
32-bit Binary
16-bit Unsigned
16-bit Signed
32-bit Unigned
32-bit Signed
32-bit Float





## 9.2 Using Shape Library and Picture Library

[Shape Library] and [Picture Library] are used for enhancing the visual effect of an object. For setting these, please go to **[Shape]** tab in the dialog for adding new object to set up [Shape Library] and [Picture Library].

New Bit Lamp Object	×
General Security Shape Label	
Shape Shape Library	hape
<ul> <li>✓ Inner</li> <li>✓ Frame</li> <li>✓ Interior pattern :</li> <li>✓ Patte</li> <li>Duplicate these attributes to every state</li> </ul>	m Style
Picture Picture Library Vse p	icture
0 1 State : 0	<b>~</b>
Picture : Red	
BMP	
Background :	

## 9.2.1 Settings of Shape Library

[Shape Library...]



Users can tick **[Use shape]** to enable this setting and select the shape from the library.

### [Inner]

Tick [Inner] to enable this setting and select a color for inner part of the shape. Click drop down button to open the **[Color]** dialogue to choose a color from the list or **[customize]** their own color and click **[Add to Custom Colors]** for system to remember this color.

Color	? 🛛
Basic colors:	
Custom colors:	
	Hue: 160 <u>R</u> ed: 0 Sat: 240 <u>G</u> reen: 0
Define Custom Colors >>	Color/Solid Lum: 120 Blue: 255
OK Cancel	Add to Custom Colors

## [Frame]

Tick [Frame] to enable this setting and select a **[color]** for the frame of the shape. The way of setting is same as above.

#### [Interior Pattern]

Click to select the style of the interior pattern of the shape. The color of this pattern can also be set.

## [Pattern Style]

Click [Pattern Style] button to open the dialogue.



Pattern Style
OK Cancel

#### [Duplicate these attributes to every state]

Duplicate all attributes of the current state to other states.

#### How to set [Shape Library...]

Click [Shape Library...] button, the following dialog appears. The currently selected shape is marked by a red frame.







The illustration above provides information of one of the Shapes in the Shape Library as follows:

2: Green	The number and the name of the shape in the library.
States: 3	The number of the states of the shape. In this case, it shows the
	Shape possesses three states.
Frame	Indicates that the Shape is set with "frame" only.

The illustration below shows that the Shape is set with "inner" and "frame."



Note: About all the settings in **[Shape Library]**, please refer to the illustrations in "Chapter 14 Shape Library and Picture Library" for details.

Click **[OK]** and preview the design of the shape after the setting is completed.



Preview	
0 1	State : 🚺 💙
	Picture :
	l
Background :	·

## 9.2.2 Settings of Picture Library

#### [Picture Library]

Users can click **[Use picture]** to enable selecting a picture from the library.

### How to set [Picture Library...]

Click **[Picture Library...]** button and **[Picture library]** dialog appears. The currently selected picture is marked by a red frame.



Picture Library				
Library :		State : 0	<b>•</b> 0 (	12345
button1 icon1			$\bigcirc$	
	0:PB Red	1:PB Green	2:PB Yellow	3:PB Blue
	States : 2	States : 2	States : 2	States : 2
	30054 bytes	30054 bytes	30054 bytes	30054 bytes
	BMP (100x100)	BMP (100x100)	BMP (100x100)	BMP (100x100)
State 0 ~ State 11 :				
	4:Lamp Red	5:Lamp Green	6:Lamp Yellow	7:Lamp Blue
	States : 2	States : 2	States : 2	States : 2
	30054 bytes	30054 bytes	30354 bytes	30054 bytes
	BMP (100x100)	BMP (100x100)	BMP (100x101)	BMP (100x100)
Select Lib New Lib	Unattach Lib.	Rename	Export	ОК
Import Picture   Modify Picture		Delete All States	Delete Cur. State	e Cancel
		Insert Before	Insert After	

WE!NTEK



The illustration above provides information of one of the Pictures in the Picture Library as follows:

Picture	0 : PB Red	The number and name of the Picture
name		
Total states	2	The number of the states of the picture
Image size	30054	The size of the Picture
	bytes	
Image	BMP	The format and resolution of the Picture; BMP means
format	(100x100)	bitmap Picture and its format can also be JPG, PNG, DPD,



or GIF. Picture Length: 100 pixels and height: 100 pixels in this case.

Note: About all the settings in **[Picture Library]**, please refer to the illustrations in "Chapter 14 Shape Library and Picture Library" for details.

Click **[OK]** and preview the design of the picture after the setting is completed.





## 9.3 Setting Text Content

Go to **[Label]** tab while adding new object to set the text content as shown below.

New Bit Lamp Object	
General Security Shape Label	
Use label	
Use bitmap font	Label Library
State : 0 V I	
Font : Arial	~
Color : Color :	Size : 16 💌
Align : Left 💙	Blink : None 💉
Underline Underline Underline Underline Every state	
Movement	
Speed :	Continuous
Content :	
¢st	
<u>&lt;</u>	
Tracking Du	plicate this label to every state

#### [Use label]

Check [Use label] and click **[Label Library]** button to add and edit the text. EB8000 supports Windows true-font.



#### [Use label library]

Check [Use label library] to choose a label tag that exists in Label Library as shown below.

🔲 Use label				
🔽 Use label library	Label tag :	NONAME	*	
			Label Library	)

#### [Label Library...]

Note: About all the settings in **[Label Library]**, please refer to the illustrations in "Chapter 15 Label Library and use Multi-Language" for details.

### [Font]

Select font style from font list. EB8000 supports Windows true-font as shown below.

Font :	Arial	~
Colori	Arial	~
COI01 .	Arial Black	-
6 1: ·	Arial Bold	_
Augn :	Arial Narrow	
	Arial Narrow Bold	
	Batang	_
	Book Antiqua	=
	Book Antiqua Bold	
	Bookman Old Style	
	Bookman Old Style Bold	-
	Bookshelf Symbol 7	

## [Color]

Select the text color.

## [Size]

Select the text size. The text sizes supported by EB8000 are listed below.





## [Align]

Select how users would like to align the text in multiple lines

Align :	Left	*
	Left Center Right	

The text aligned [Left].

11	1			
22	22	22	2	
33	33	33	33	3

The text aligned [Center].

111
222222
333333333

The text aligned [Right].

				1	1	1
	2	2	2	2	2	2
333	3	3	3	3	3	3



#### [Blink]

To decide how will the text blink:

Choose [None] to disable this feature or set blinking interval as [1 second] or [0.5 seconds].



[Italic] Use Italic font.

Italic Label

## [Underline]

Use Underline font.

## Underline Label

### [Movement] setting

**[Direction]** Set the direction of the marquee effect.

No movement	~
No movement	
Left	
Right	
Up	
Down	

## [Continuous]

Whether this selection is tick or not influences how the marquee effect is displayed:



If **not** checking [Continuous], the next text appears only when the previous text disappears completely. See the picture below.





If checking [Continuous], the text will be displayed continuously.



### [Speed]

Adjust the speed of the text movement.

### [Content]

Set the content of the text. If using **[Label Library]**, the content will be sourced from Label Library.

### [Tracking]

When [Tracking] is selected, moving the text of one state will also move the text of other states.

### [Duplicate this label to other states]

This function is used to duplicate the current text content to the other states.



## 9.4 Adjusting Profile Size

When an object is created, double click it and go to the [Profile] tab to adjust the position and size of the object.

Bit Lamp Object's	Properties		
General Shape	Label Profile		
Position Pinned	X : 14 😂	Y : 40 🗢	
_Size	Width : 50 ᅌ	Height : 50 🗢	

#### a. Position

Set if the position and size of the object is **[Pinned]**. When it is checked, the position and size of the object cannot be changed. X and Y mean the **[X]** and **[Y]** coordinate of the left-top corner of the object.

#### b. Size

Adjust the [width] and [height] of the object.





## 9.5 Variables of Station Number

EB8000 version 1.31 or higher allows users to set variables of station number in PLC address. As shown below, "var2" is one of 16 station number variables.

Address	×
PLC name : MITSUBISHI FX0n/FX2	*
Device type : X	*
Address : var2#123	

The syntax of variable of station number:

#### varN#address

The range of N is integer from 0~15; address means PLC address.

16 variables are available : var0 ~ var15. These variables of station number read values from address LW-10000~LW-10015. The list below shows variables and its corresponding system reserved address LW :

var0	I W-10000
varo	
var1	LW-10001
var2	LW-10002
var3	LW-10003
var4	LW-10004
var5	LW-10005
var6	LW-10006
var7	LW-10007
var8	LW-10008
var9	LW-10009
var10	LW-10010
var11	LW-10011
var12	LW-10012
var13	LW-10013
var14	LW-10014
var15	LW-10015



For example, "var0" reads value from LW-10000, when value in LW-10000 is "32", var0#234 = 32#234 (the station number is 32); similarly, "var13" reads value from LW-10013, when value in LW10013 is" 5", var13#234 = 5#234.





## 9.6 Broadcast Station Number

MT6000/8000 provides two ways for users to enable using broadcast command. First is to set it directly in **[system parameter settings] [Device]** tab:

PLC type : MITSUBISHI FX0n/FX2	•
V.1.00, MITSUBISHI_FX0N.so	
PLC I/F : RS-485 4W	PLC default station no. : 0
COM : COM1 (9600,E,7,1)	Settings
Use broadcast command	Broadcast station no. : 0

Second way is to use system tag to enable or disable broadcast station number or to change it.

Corresponding system tags are listed as below:

- LB-9065 disable/enable COM 1 broadcast station no.
- LB-9066 disable/enable COM 2 broadcast station no.
- LB-9067 disable/enable COM 3 broadcast station no.
- LW-9565 COM 1 broadcast station no.
- LW-9566 COM 2 broadcast station no.
- LW-9567 COM 3 broadcast station no.



# Chapter 10 Security

Security of objects in EB8000 includes two parts:

- 1. User password and corresponding operable classes
- 2. Security settings of every single object

## **10.1 Settings of Password and Classes**

Go to **[Edit]**/ **[System Parameter Settings]**/ **[Security]** to set user password and operable classes of objects.

There are seven security levels, classified from "A to F" and includes "none".

Password should be digits from **0 to 9** and up to **12** sets of user password are available.

ystem Parameter S	Settings			
Font	Extend	Extended Memory		ckup Server
Device	Model	General	System Setting	Security
- Select operatable o	lasses for each user —			
Enable	Password : 111	🗹 A	□в 🔽 С 🔲 І	E F
User 2 Enable	Password : 0	<b>V</b> A	B C I	E F
User 3	Password : 0	A		E F
User 4				
User 5				
-User 6				

Once password is entered, the objects that the user can adjust are set here. For example, when the security class of "User 1" is set as above, only objects with class "A, C, E" and "none" can the user adjust.


## The correct process of inputting password:

- 1. Input the passwords to the system reserved register [LW-9220: password] (2 Words, 32 bits).
- 2. Use [LW-9219: User no. (1~12)] (1 Word, 16bit) to designate current user.

**Note**: value in [LW-9219] must be 1~12, which represents "User 1"~"User 12" respectively.

If the input password is wrong, state of [LB-9060: password error] will be set ON; If the input password is correct, state of [LB-9060] returns to OFF automatically.

The passwords of user 1 to user 12 can be obtained from system reserved registers [LW-9500: user 1's password] to [LW-9522: user 12's password], 24 words in total.

#### Users can change passwords even when the HMI is in operation:

When state of system reserved register [LB-9061: update password (set ON)] switches from OFF to ON, EB8000 will use the data saved in [LW-9500] to [LW-9522] to update the password and use the new password in future.

**Note**: The user operable classes of objects won't be changed due to the change of password.

When the state of [LB-9050: user logout] switches from OFF to ON, **current user will be forced to logout the system**. At this time, only the object defined as "None" can be operated.

# [LW-9222: classes can be operated for current user] records the operable classes for current user:

bit0 = 1 means the operable object for current user is class "A";

bit1=1 means the operable object for current user is class "B "and so on.



# 10.2 Security of Objects

General       Security       Shape       Label         Safety control       Min. press time (sec) : 0         Display confirmation request       Max. waiting time (sec) : 10         Interlock       Interlock         Use interlock function         Hide when disabled         Grayed label when disabled         Enable when Bit is ON
Safety control Min. press time (sec) : 0 Display confirmation request Max. waiting time (sec) : 10 Interlock Use interlock function Hide when disabled Grayed label when disabled Enable when Bit is ON Enable when Bit is OFF
Min. press time (sec) : 0 Display confirmation request Max. waiting time (sec) : 10 Interlock Use interlock function Hide when disabled Grayed label when disabled Enable when Bit is ON Enable when Bit is OFF
<ul> <li>Display confirmation request</li> <li>Max. waiting time (sec) : 10</li> <li>Interlock</li> <li>Use interlock function</li> <li>Hide when disabled</li> <li>Grayed label when disabled</li> <li>Enable when Bit is ON</li> <li>Enable when Bit is OFF</li> </ul>
Interlock          Image: Use interlock function         Image: Hide when disabled         Grayed label when disabled         Enable when Bit is ON
<ul> <li>✓ Use interlock function</li> <li>✓ Hide when disabled</li> <li>☐ Grayed label when disabled</li> <li>⊙ Enable when Bit is ON</li> <li>○ Enable when Bit is OFF</li> </ul>
<ul> <li>Hide when disabled</li> <li>Grayed label when disabled</li> <li>Enable when Bit is ON</li> <li>Enable when Bit is OFF</li> </ul>
<ul> <li>Grayed label when disabled</li> <li>Enable when Bit is ON</li> <li>Enable when Bit is OFF</li> </ul>
Senable when Bit is ON Enable when Bit is OFF
PLC name : Local HMI
Address : LB 🗸 0
User restriction
Object class : Class A 🖌
Disable protection permanently after initial activation
Display warning message if access denied
Make invisible while protected
Sound
Enable Sound Library Sound Index : Default Play

Setting	Description			
Safety control	[Safety control] is mainly used to prevent operator from			
	miss-operating an object accidentally. There are two methods for			
	protection:			
	[Min. press time (sec)]			
	Only when pressing the object continuously longer than the time set			
	here can an object to be activated successfully.			
	[Display confirmation request]			
	After pressing the object, a confirm dialog appears. Users need to click			









WE!NTEK

Once the permitted security class of the user meets that of the object, the system will stop checking the security class when operating this



	object permanently. Which means even if the user is changed this
	object can still be operated freely.
	[Display warning message if access denied]
	When the user's current security level does not meet that of the object,
	a warning dialog appears.
	Password Protected! Access Denied !! Close
	Window 7 is set as an alert message for authority security. Users can design the content of the message
	[Make invisible while protected]
	When a user's security level does not meet that of the object, the
	object will be hidden.
Sound	Each object can be set whether to use the beeper to make a specific
	sound or not. A system register [LB9019] is used as the switch of the
	beeper. When state of [LB9019] is OFF, beeper is enabled. If restart
	HMI, the settings of beeper stay the same.



# **10.3 Examples of Security**

The following illustrates the steps of security feature:

Step1: First of all, create a new project. Go to [System parameter]/ [Security], add three users and set different passwords and classes.

em Parameter	Settings							
Font		Extended Memory			Prin	nter/Back	kup Serv	er
Device	Model	General		Syster	n Settin;	g	Se	curity
User 1 Enable	Password : 1	11	💌 A	В	□c	D	E	F
User 2 Enable	Password : 2	222	💌 A	₽B	□c	D	E	F
User 3	Password :	333		₽B	ГС	D	E	F

"User 1" can operate objects with class A, "user 2" can operate objects with class A and B, and "user 3" can operate objects with class A, B, and C.

Step2: Set objects in Window\_10 as below:

User name :	##### <i>LW9219</i>	
Password :	##### <i>LW9220</i>	
1	bit 15	bit 0
Current status :	111111111111	11111 <i>LW9222</i>
	SB_3 Logout	LB9050
	Class A Button	
	<sup>SB_1</sup> Class B Button	
	<sup>SB_2</sup> Class C Button	]



[NE\_0] and [NE\_1] are [numeric input] objects with addresses [LW-9219] and [LW-9220] that are for inputting user ID and password. [LW-9219] is for entering user ID (1~12), with the length of 1 word, in a data format of 16-bit Unsigned as below.

Numeric Display Object's Properties	×
General Numeric Format Security Shape Font Profile	
Display —	-
Data format : 16-bit Unsigned 💽 🦳 Mask	
Number of digits	
Left of decimal Pt. : 4 📚 Right of decimal Pt. : 0	\$

[LW-9220] is for entering user password with a length of 2 words, in a data format of 32-bit Unsigned as below.

neral Numeric Format	Security	Shape	Font	Profile
lay				
Data format :	32-bit Unsi	igned	¥ [	🗌 Mask
unber of digits				
mber of digits				

[ND\_0] is [numeric display] object with address [LW-9222] to indicate user's state. The data is in the format of 16-bit Binary.

Display—				
	Data format :	16-bit Binary	*	Mask

[SB\_0]~ [SB\_2] are [Set Bit] objects which are set with different classes but all selected **[Make invisible while protected]**. i.e. [SB\_0] is class A, [SB\_1] is class B, and [SB\_2] is class C. The settings of [SB\_0] object:

Object class Class A	Object class : Class A  Disable protection permanently after initial acti	ermanently after initial activatio	Class A	bject class :
	Disable protection permanently after initial acti	ermanently after initial activatio		
			protection per	Disable p
Disable protection permanently after initial activa	<b>=</b>			- · · ·
Disable protection permanently after initial active	Display warning message if access denied	ssage if access denied	/aming messa	Display v

The [Set Bit] object (SB\_3, LB-9050) is for user logout and is set as below:

New Set Bit Object 🛛 🔀
General Security Shape Label
Description :
Write address
PLC name : Local HMI Setting
Address : LB-9050 : user logout

**Step 3:** After completing the design and settings of the objects, please save, compile project and do the off-line simulation. The illustration below is initial screen of off-line simulation.

Since no password is entered this time, object [ND\_0] [LW9222] shows "00000000000000000" which means current user can only use objects with "none" class.

Moreover, [SB\_0]~[SB\_2] are objects with security levels of class A~ class C and at the same time [Make invisible while protected] is selected, therefore, [SB\_0]~[SB\_2] objects are hidden by the system.

User name : 1 LW9219
Password : 0 LW9220
bit 15 bit 0
Current status : 00000000000000000000000000000000000
Logout LB9050



Step 4: When user enters the password of user 1, "111", the display will become:

User name : Password :	1 <i>LW9219</i> 111 <i>LW9220</i>
E Current status :	bit 15 bit 0 000000000000001 LW9222 Logout LB9050
	Class A Button

Since "user 1" is permitted to use objects with class "A", object [SB\_0] appears and allows user to operate. Now, bit 0 in [LW-9222] becomes "1".

Step 5: Next, when user enters "user 3's" password (333), the display will become:

User name : 3 LW9219
Password : 333 LW9220
bit 15 bit ()
Di 15 Di D
Current status : 000000000000111 LW9222
1 10050
Logout
Class A Button
Class B Button
Class C Button

Since "user 3" is permitted to use objects with class "A, B, and C". Now, bit 0 ~ bit 3 in [LW-9222] becomes "111" and allows current user to use objects with class A, B, and C.



**Step 6:** At this time, if [SB\_3] [LB-9050] is pressed to force current user to logout, the system will return to initial state. In other words, current user can only use objects with "none" class.

User name :	3 LW9219
Password :	333 LW9220
bit 1:	5 bit 0
Current status :	D0000000000000000000000000000000000000
	Logout LB9050



# **Chapter 11 Index Register**

# **11.1 Introduction**

EB8000 provides 32 index registers for users to use addresses flexibly. Via index register, users can update object's read / write address without changing its content while HMI is running the project.

The addresses of the 32 index registers are as follows:

INDEX 0 [LW-9200] (16-bit) ..... INDEX 15 [LW-9215] (16-bit) INDEX 16 [LW-9230] (32-bit) ..... INDEX 31 [LW-9260] (32-bit)

INDEX0~INDEX31 are descriptions of Index Register. [LW-9200]~ [LW9260] are the addresses of these index registers.

INDEX 0 ~ INDEX 15 are 16-bit registers with the range up to 65536 words; INDEX 16 ~ INDEX 31 are 32-bit registers with the range up to 4G words.

While using **[Index Register]**, the address of the **[device type]** will be decided by the value of "constant in set address+ value in chosen index register". Index register works in all **[device lists]** built in **[system parameter settings]**, no matter addresses in bit or word format.



# 11.2 Examples of Index Register

The following examples show how to use index registers.

The "Read address" is set to [LW100] and [Index register] is not checked.

In this way the read address won't change while running the project.

-Read address						
PLC name : Local HMI		Setting				
Address : LW	✓ 100	16-bit Unsigned				
l						
Press Setting						
Address						
PLC name : Local HMI		×				
Device type : LW		~				
Address : 100	System tag					
Address format : DDDDD [rang	ge : 0 ~ 10500]					
Index register						

But in another case as below, **[Index register]** is checked and the chosen index register is **[INDEX3]**. In this way the read address will change to [LW (100+INDEX3)]. INDEX 3 represents value in address [LW9203]. In other words, if value in [LW9023] is "5", the read address in this case will be LW (100+5) = [LW105].

- Read address PLC name : Address :	Local HMI Setting LW 100 IDX 3 16-bit Unsigned
Address	×
PLC name :	Local HMI
Device type :	LW
Address :	100 System tag
Address format :	DDDDD [range : 0 ~ 10500]
Index :	INDEX 3 (16-bit) V Index register
	16-bit Unsigned 💙
	OK Cancel

WE!NTEK

For example as below, set Index 3 as "0", which means the value in address [LW9203] is "0". Under this situation, the contents of [LW100 + Index 3] and [LW101 + Index 3] are the same as those of the [LW100] and [LW101].

Object A       12       Object C       12       0         LW 100 + Index 3       LW 100       LW 120       12       0         34       34       0       12       0         LW 101 + Index 3       LW 101       LW 121       12         Object B       0       10       10       12         Index 3       1       10       10       12	
weînniem	



At this time, the settings of read address of Object A:

PLC name :	Local HMI 😽 😽		
Device type :	LW 💌		
Address :	100 System tag		
Address format : DDDDD [range : $0 \sim 10500$ ]			
Index :	INDEX 3 (16-bit) 🔽 🔽 Index register		

The settings of read address of Object B:

PLC name : Local HMI	*	
Device type : LW-9203 (16bit) : address index 3	~	
Address : LW9203 System tag		
Address format : DDDDD [range : $0 \sim 10500$ ]		

The settings of read address of Object C:

[			
PLC name : Local H	-MI		
Device type : LW	~		
Address : 100	System tag		
Address format : DDDDD	Address format : DDDDD [range : $0 \sim 10500$ ]		
	Index register		

Now, if users change value of Index 3 to "20", the contents of [LW100 + Index 3] and [LW101 + Index 3] will become those of [LW120] and [LW121], i.e. the values in [LW100+20=LW120] and [LW101+20=LW121].

WE!NTEK	EasyBuilder8000 User's Manual
	Object A       6       Object C       12       56         LW 100 + Index 3       LW 100       56       LW 120         78       34       78       LW 121         Object B       20       LW 101       LW 121         Index 3       Index 3       LW 101       LW 121

Similarly, the index register can also work with bit address.

1 word = 16 bits, in other words, 1 value change of index register means the change of 16 bits.

See the example below. When INDEX 5 is set as "0", the state of [Bit Lamp] [LB0] and [LB6] are the same as those of [Toggle Switch] ~ [LB0+Index 5] and [LB6+Index 5] and are displayed ON.





If users change value of index 5 to "1", the state of [Bit Lamp] [LB16] and [LB22] are the same as those of [Toggle Switch] ~ [LB0+Index 5] and [LB6+Index 5] and are displayed ON.



In conclusion: From illustration above, we realize that Index register is used to change addresses. Through changing the data in index register, we can make an object to read and write different data from different addresses without changing its own address of the device. Therefore, we can transmit or exchange data among different addresses.



# Chapter 12 Keyboard Design and Usage

Both **[Numeric Input]** and **[ASCII Input]** objects need to use keyboard as input tool. Except for calling up a popup keyboard, users can design a keyboard without title bar or a fixed keyboard in the window. Even UNICODE keyboard can be created. Both numeric keyboard and ASCII keyboard are created with **[Function Key]** object. The process and usage are illustrated below.

# 12.1 Steps to Design a Pop-up Keypad

## Step 1

Create and open a window for a keyboard to be added. For example, set [WINDOW 200] as the window for a keyboard.

 198
 199
 200: Keyboard
 201
 202
 203

## Step 2

Adjust the height and width of WINDOW 200 and create a variety of **[Function Key]** objects in it. Input signals will be triggered by pressing [Function Keys].

FK_0	FK_1	FK_2	FK_3
FK_4	FK_5	FK_6	FK_7
FK_8	FK_9	FK_10	FK_11
FK_12	FK_13	FK_14	

The [Function Key] objects in [WINDOW 200] are arranged as above. These objects should be set in **[ASCII/UNICODE mode]**.



Here FK\_11 is used as the [Escape (Esc)] key and its settings:

ASCII/UNICODE mode					
	🔘 [Enter]	🔘 [Backspace]	🔘 [Clear]	⊙ [E∞]	
	O [ASCII] / [UNICODE]				

FK\_14 is used as the **[ENTER]** key and its settings:



[Function Key] s other than FK\_11 and FK\_14 are mostly used to input number or text. For example, FK\_0 is used for inputting number [1] and its settings:

1	ASC	CII/UNICODE n	node			
		🔘 [Enter]	🔘 [Backspace]	🔘 [Clear]	🔘 [Ex:]	
	_	⊙ [ASCII] / [	UNICODE] 1			

Select a suitable Picture for each [Function Key] object. GP\_0 is a **[picture]** object which is placed in the bottom layer as the background.



#### Step 3

Go to [General] tab in [System Parameter Settings] and click [Add...] in [Keyboard]. [Add a keyboard] dialog appears. Select [WINDOW 200] and press [OK].



Window no.	200. Keyboard	•

As illustrated below, a new item: "200.Keyboard" will be added to [Keyboard] in [General] tab in [System Parameter Settings.]

System Parameter Settir	gs		
Font	Extended Memory	Printer/Back	up Server
Device I	Model General	System Setting	Security
-Fast selection button			
Attribu	te : Enable 💌 🚺	Settings	
Positio	n: Left 🔽 🗌	Hide button when HMI star	rts
-Screen saver			
Back light save	er : None 🛛 🛛 minu	ute (s)	
Screen save	er : None 🛛 🔽 minu	te (s)	
_ Options			
Startup window n	D.: 10. WINDOW_010		~
Common windo	w : Above base window 🛛 💌	Object layout : Nature	~
Keyboard caret cold	or:	🔽 R W_A enabled	
Event			
	🔽 Use L W9450~9455 as time	tags of event logs	
	BCD 🗸		
Extra. no. of even	ts: 0		
-Keyboard			1
	55. Keypad 6 - Integer 56. Keypad 7 - HEX 57. Keypad 8 - Floating 60. ASCII Middle 61. ASCII Small 200. keyboard.		Add Delete

After a Keyboard object is created, when open the **[Numeric Input]** or **[ASCII Input]** object, "200.Keyboard" can be found in **[Keyboard]** Data Entry tab, as shown below. **[Popup Position]** is used to decide the display position of the Keyboard in screen.



EB8000 divides the screen into 9 areas.

Window no. : 200. keyboard				*
Popup position : {relative to HMI screen}	000	0000	000	

Select **[200.Keyboard]**. When users press **[Numeric Input]** or **[ASCII Input]** object, WINDOW 200 will pop up in HMI screen. Users can press keys in keyboard to input data.





# **12.2 Steps to Design a Keyboard with Direct Window**

If users don't need the title bar of the keyboard, a direct window can be used as follows.

# Step 1

Create a [Direct window] and set a read address to activate it.

In [General] / [Attribute] tab of adding new object dialogue, select [No title bar] and [Window no].

## Step 2

Set the [Profile] of [direct window] object to same size as [WINDOW 200].





Direct Window Obje	ct's Properties		
General Profile			
-Position			
Pinned	X : 80 🗘	¥ : 57 😜	
Size	W:14 . 160	II.:	
	Wiath : 100 🤤	Height: 230	

## Step 3

Create a [Numeric Input] object, and don't select [Use a popup keypad].

New Numeric Inpu	nt Object	×
General Data Entry	Numeric Format Security Shape Font	_
Mode	: Touch	
Input order Enable		
	Use a popup keypad	
Hint : If the k or on the sam	keyboard is an USB keyboard, on indirect/direct window, ne window, please don't check "Use a popup keypad".	

#### Step 4

Add a **[Set Bit]** object, set **[LB 0]** as **ON** and overlay it on the **[Numeric Input]** object. Add **[Set Bit]** objects on the **[Enter]** and **[ESC]** function keys respectively. Set **[LB0]** as **OFF**. In this way when user presses either [Enter] or [ESC] will close the keyboard.



w Set Bit Object 🛛 🔀
eneral Security Shape Label
Description :
- Write address
PLC name : Local HMI 🛛 Setting
Address : LB 🗸
Write after button is released
Attribute
Set style : Set ON 🗸

₩. ##	нн	ŧ	Ī		
WC_0			-		



# 12.3 Steps to Design a Fixed Keyboard

Users can also place a fixed keyboard in the window instead of popup keyboard or direct window. The keyboard can't be moved or canceled this way.

## Step 1

Create a [Numeric Input] object, and don't select [Use a popup keypad].

## Step 2

Design a keyboard with [function keys] and place them in screen.

-AS	CII/UNICODE n	node			
	🔘 [Enter]	🔘 [Backspace]	🔘 [Clear]	○[E∞]	
	⊙ [ASCII] / [	UNICODE] 1			

## Step 3

When pressing [numeric input] object, users can input value with function keys directly.

1				
1	2	3		
4	5	6		
7	8	9		
Basks	Baskspace			



# 12.4 Creating UNICODE Keyboard

To create UNICODE keyboard is in the same way as numeric keyboard, all with function keys. The settings are as below:

ASCHIOMICODE	mode			
🔘 [Enter]	🔘 [Backspace]	🔘 [Clear]	🔘 [Ex:]	
⊙ [ASCII] / [U	INICODE] a			
ASCII/UNICODE	mode			
🔘 [Enter]	🔘 [Backspace]	🔘 [Clear]	🔘 [Ex]	
💽 [ASCII] / [U	NICODE] ß			
~ASCII/UNICODE	mode			
🔘 [Enter]	🔘 [Backspace]	🔘 [Clear]	○[Ex:]	
○ [Enter]	O [Backspace] NICODE] Υ	🔘 [Clear]	○[Ex]	_
C [Enter]	O [Backspace] [NICODE] Υ mode	○ [Clear]	○ [Ex]	_
C [Enter] (ASCII] / [U ASCII/UNICODE (Enter]	<ul> <li>[Backspace]</li> <li>NICODE] γ</li> <li>mode</li> <li>[Backspace]</li> </ul>	○ [Clear] ○ [Clear]	○[Ex] ○[Ex]	_

After the settings are completed, function keys  $[\alpha]$   $[\beta]$   $[\gamma]$   $[\delta]$  are created.

Create an **[Enter]** key. A simple UNICODE keyboard is built. Place a **[ASCII Input]** object in window, set **[No. of Words]** as **[8]** (1 word =2 bytes) and tick **[Use UNICODE]** as below.



neral	Data Entry	Security S	Shape Fo	ont Prof	ile		
1	Description :						
Read :	Max 📃	k 💽	🛛 Use UNIC	ODE [	Reverse h	igh/low byte	•
	PLC name :	Local HMI				V Setti	ng
	Address :	LW	-	✓ 0			
ldress	PI C name :	l ocal HMI					
ldress	PLC name :	Local HMI					
ldres: D	PLC name : evice type :	Local HMI LW					
ldress D	PLC name : evice type : Address :	Local HMI LW 0		Systen	ntag		
ldres: D Addri	PLC name : evice type : Address : ess format :	Local HMI LW 0 DDDDD [ran	ge : 0 ~ 105	00]	) tag		
ldress D Addr	PLC name : evice type : Address : ess format :	Local HMI LW 0 DDDDD [ran	ge : 0 ~ 105	00]	n tag register		

UN	CODE_
FK_0	α
FK_1	β
FK_2	γ
FK_3	δ
FK_4 I	Enter

In conclusion: Numeric keyboard and ASCII keyboard are all made by combining function keys. Users can group up the self made keyboard and add to library for future use. If not using the default keyboard, self defined keyboard can also be used. Add newly made keyboard to **[System parameter settings]/ [General]/ [Keyboard]**.



# Chapter 13 Objects

This chapter is to illustrate the ways of using and setting all kinds of objects. For those settings general for all the objects, such as index register, label, shape, and so on, please refer to "Chapter 9 Object's General Properties".

# 13.1 Bit Lamp

#### Overview

Bit Lamp object displays the ON and OFF state of a bit address. If the bit state is OFF, the State 0 shape will be displayed. If the bit state is ON, the State 1 shape will be displayed.



## Configuration

Click the **[Bit Lamp]** icon in the toolbar and the **[Bit Lamp Object's Properties]** dialogue box will appear, fill in the content of and press **[OK]**, a new bit lamp object will be created. See the pictures below.





New Bit Lamp Object
General Security Shape Label
Description :
Read address PLC name : Local HMI
Address : LB V 0
Invert signal
Blinking time : 0.5 second(s)
Mode : Alternating image on state 0
OK Cancel Help

Setting	Description
Description	A reference name that's assigned by user for the object. The system does
	not make use of this reference name since it is for user's document only.





# 13.2 Word Lamp

# Overview

A Word Lamp object displays the corresponding shape according to the value in the designated word address. (up to maximum of 256 states)



# Configuration

Click the **[Word Lamp]** icon in the toolbar and the**[Word Lamp Object's Properties]** dialogue box will appear, fill in each items and press **[OK]** button, a new word lamp object will be created. See the pictures below.





Setting	Description
[Mode] /	Word lamp object offers the following three modes for selection:
[Offset]	
	a. Value
	Calculate result of word value to subtract [Offset] and display its
	corresponding shape.
L	

General Security Share Label         Description:         Mode:       Value         Mode:       Value         PLC name:       Local HMI         Address:       LW         PLC name:       Local HMI         Address:       LW         No. of states:       2         In the above setting, if the value of [LW200] is "5", the shape of state "2" is displayed. See the picture below.         5       State 2         LW200       LW200, Offset = 3	New Word Lamp	Object				
Description:       Mode:       Value       Offset:       3         Read address       PLC name:       Local HMI       Visit Setting       16-bit Unsigned         Address:       LW       200       16-bit Unsigned       16-bit Unsigned         Attribute       No. of states:       2       2         In the above setting, if the value of [LW200] is "5", the shape of state "2" is displayed. See the picture below.       5       State 2         LW200       LW200, Offset = 3       b. LSB         Transfer the read address value to binary, the lowest 8 bits other than value	General Security	Shape Label				
Mode:       Value       Offset:       3         PLC name:       Local HMI       Image: Setting       Image: Setting       16-bit Unsigned         Address:       LW       200       If 6-bit Unsigned       Image: Setting       Image: Setting         Attribute       No. of states:       2       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: Setting       Image: Setting         Image: Setting.       Image: Setting       Image: Setting       Image: S	Description	1:				
Read address         PLC name : Local HMI         Address : LW         Address : LW         You of states : 2         Attribute         No. of states : 2         It is above setting, if the value of [LW200] is "5", the shape of state "2" is splayed. See the picture below.         5       State 2         LW200       LW200, Offset = 3         LSB	Mode	: Value	C	)ffset: 3		-
PLC name :       Local HMI         Address :       LW200         Attribute       No. of states :         No. of states :       2         It is above setting, if the value of [LW200] is "5", the shape of state "2" is splayed. See the picture below.         5       State 2         LW200       LW200, Offset = 3	Read address —					_
Address : LW 200 16-bit Unsigned Attribute No. of states : 2 2 3 the above setting, if the value of [LW200] is "5", the shape of state "2" is splayed. See the picture below. 5 52422 $LW200 LW200, Offset = 3$ LSB	PLC name	: Local HMI		*	Setting	
Attribute No. of states : 2 The above setting, if the value of [LW200] is "5", the shape of state "2" is isplayed. See the picture below. 5 $5$ $Cate 2$ $LW200$ $Cot fiset = 3$ . LSB	Address	: LW	200		16-bit Unsigne	ed
Attribute No. of states : 2 · · · · · · · · · · · · · · · · · ·						
Attribute No. of states : 2 $\checkmark$ In the above setting, if the value of [LW200] is "5", the shape of state "2" is displayed. See the picture below. 5 State 2 $LW200 LW200, Offset = 3$ D. LSB						
Attribute No. of states : 2 In the above setting, if the value of [LW200] is "5", the shape of state "2" is displayed. See the picture below. 5 State 2 $LW200 LW200, Offset = 3$ b. LSB						
Attribute No. of states : 2 $\checkmark$ n the above setting, if the value of [LW200] is "5", the shape of state "2" is tisplayed. See the picture below. 5 State 2 $LW200 LW200, Offset = 3$ $SLSB$						
Attribute No. of states : 2 $\checkmark$ of states : 2						
Attribute No. of states : 2 The above setting, if the value of [LW200] is "5", the shape of state "2" is isplayed. See the picture below. 5 State 2 <i>LW200</i> <i>LW200, Offset = 3</i> <b>. LSB</b>						
Attribute       No. of states : 2         In the above setting, if the value of [LW200] is "5", the shape of state "2" is displayed. See the picture below.         5       State 2         LW200       LW200, Offset = 3         D. LSB						
No. of states : 2 The above setting, if the value of [LW200] is "5", the shape of state "2" is isplayed. See the picture below.	Attribute					
the above setting, if the value of [LW200] is "5", the shape of state "2" is splayed. See the picture below. 5 State 2 <i>LW200 LW200, Offset = 3</i> LSB			No. of a	states : 2		≤
In the above setting, if the value of [LW200] is "5", the shape of state "2" is is isplayed. See the picture below.						
the above setting, if the value of [LW200] is "5", the shape of state "2" is isplayed. See the picture below. 5 State 2 <i>LW200 LW200, Offset = 3</i> <b>LSB</b>						
isplayed. See the picture below. 5 State 2 <i>LW200 LW200, Offset = 3</i> <b>LSB</b>	n the above se	tting, if the value	of [LW200	)] is "5", th	e shape o	of state "2" is
5 State 2 <i>LW200 LW200, Offset = 3</i> <b>LSB</b>	isplayed. See	the picture below	Ι.			
5       State 2         LW200       LW200, Offset = 3         LSB       Event of the read address value to bipary, the lowest 8 bits other than value to bipary.					~	
<i>LW200 LW200, Offset = 3</i> b. LSB		5		State	2	
<i>LW200 LW200, Offset = 3</i> b. LSB						
<b>b. LSB</b>		LW200	LW20	00, Off.	set = 3	
<b>5. LSB</b>						
Fransfer the read address value to binary, the lowest 8 bits other than value	o. LSB					
Fransfer the read address value to binany, the lowest 8 bits other than value						
Tansier the read address value to binary, the lowest o bits other than value	ransfer the rea	ad address value	to binary,	the lowes	t 8 bits oth	her than value
decides the state. Please refer to the following table.	decides the sta	te. Please refer to	o the follow	ving table		



Γ

		Deed address	Dimensionalise	Distributed state
		Read address	Binary value	Displayed state
	-	value	0000	All hits are 0 display the share of
		0	0000	All bits are 0, display the shape of
	_	1	0001	The lowest bit other than 0 is bit
		1	0001	I he lowest bit other than 0 is bit
		0	0010	U, display the shape of state 1
		2	0010	I ne lowest bit other than U is bit
	_	2	0044	1, display the shape of state 2
		3	0011	I he lowest bit other than 0 is bit
	_			0, display the shape of state 1
		4	0100	The lowest bit other than 0 is bit
	_			2, display the shape of state 3
		7	0111	The lowest bit other than 0 is bit
	_			0, display the shape of state 1
		8	1000	The lowest bit other than 0 is bit
				3, display the shape of state 4
	The state	es of the object has splays different s	ave nothing to c hape of states a	lo with the word value. The system according to time frequency.
Read address	Click [Sett [System t object. Us	ting] to Select ag], [Index registers can also set a	the <b>[PLC name</b> ster] of the word address in <b>[Gen</b>	], [Device type], [Address], d device that controls the word lamp eral] tab while adding a new object.
	Address			X
	PLC	name : Local HMI		✓
	Device	etype: LW		✓
	Ad	ldress : 0	System ta	g
	Address f	ormat : DDDDD [range : I	0~10500]	
			📃 Index regi	ster
		16-bit Unsigned	~	
				K Cancel
Attribute	[No. of sta The numb	ates] per states one o	bject possesse	s. State 0 is also counted as one



state.. Suppose the number of the states is 8, the valid states will be 0, 1~7. In this case if the word value is 8 or higher, the system will display the shape of last state.

#### Restrictions

In label dialog, Language 1 is able to change attribute settings, and for Language 2~8, only font size can be changed and other settings follows language 1.

neral Security Shape Label	Oenepal Security Shape Lebel
Use label Use label khrsry Use label khrsry Use bitnap foot Label Label_0 Label Libery	□ Uze label □ Uze label library Label tag : Label_0 □ Uze bitmap font Label Library
Lengrage: 1	Lánguage : 2 V State : 0 V V D I
Attribute	Attribute
Color: Size : 16	Color: Size: 16
Aliga: Left 💙 Blink: None 🗸	Align : Left 🛩 Blink : Note 🛩
Every state Every language All Movement Disection : No movement	Every state Every language All Movement Direction : Royanovement
Context :	Content :
0	10 12
Tracking Duplicate this label to every inste	Tracking Duplicate this label to every state



# 13.3 Set Bit

## Overview

The **[Set Bit]** object provides two operation modes: the "manual operation" mode defines a touch area, users can activate the touch area to set the state of the bit device to be ON or OFF. When users select the "automatic operation" mode, the operation will be automatically activated in pre-configured conditions, the touch area has no action in any circumstance.

# Configuration

Click the **[Set Bit]** icon in the toolbar and the **[New Set Bit Object]** dialogue box will appear, fill in each items and press **[OK]** button, a new Set Bit object will be created. See the pictures below.



lew Set Bit Object	X
General Security Shape Label	
Boodary Brack	
Description :	
Write address	ן ה
PLC name : Local HMI	
Address : LB 0	
Write after button is released	
Attribute	-
Set style : Toggle	
_ Macro	5
Execute macro Macro : ID1 (ID : 1)	
Trigger mode : OFF->ON	
OFF->ON ON->OFF	μ
OFF<->ON	
OK Cancel Help	

Setting	Description	
Write address	Click [Setting] to select the [PLC name], [Device type], [Address],	
	[System tag], [Index register] of the bit device that system set value to.	
	Users can also set address in [General] tab while adding a new object.	
	Address	
-------------	-------------------------------	--
	PLC pame : Local HMI	
	Device type : LB	
	Address : 0	System tag
	Address format : DDDDD [range	: 0 ~ 11999]
		Index register
	l	
		OK Cancel
-	[Write after button is rel	eased]
	If this function is selected	, the operation is activated after button is touched
	and released, otherwise,	if not selected, operation will be activated once
	the button is touched. If the	ne "Momentary" switch is selected as the
	operation mode, the [Writ	e after button is released] function will be
	ignored.	
Attribute	Please refer to the followi	ng description for different types of operation
[Set Style]	Set style	Description
	Set ON	When the operation is activated, the bit device
		will be set to ON.
	Set OFF	When the operation is activated, the bit device
		will be set to OFF.
	Toggle	When the operation is activated, the bit device
		will be set from ON to OFF or from OFF to ON.
	Momentary	When touch and hold the area, the bit device
		will be set to ON, and the bit device will be set
	Periodical toggle	The state of the hit device will be switched
		between ON and OFF periodically
		Operation's time interval can be selected in
		the combo box showed in the picture below:
		Time interval : 1.0 second (s)
	Set ON when window	When the window containing the Set Bit object
	opens	is opened, the bit device will be automatically
		set to ON.



	Set OFF when window opens	When the window containing the Set Bit object is opened, the bit device will be automatically
	Set ON when window	When the window containing the Set Bit object
	closes	is closed, the bit device will be automatically
		set to ON.
	Set OFF when window	When the window containing the Set Bit object
	closes	is closed, the bit device will be automatically
		set to OFF.
	Set ON when	When the backlight is turned on, the bit device
	backlight on	is automatically set ON.
	Set OFF when	When the backlight is turned on, the bit device
	backlight on	is automatically set OFF.
	Set ON when	When the backlight is turned off, the bit device
	backlight off	is automatically set ON.
	Set OFF when	When the backlight is turned off, the bit device
	backlight off	is automatically set OFF.
Macro	Users can use [set bit]	object to activate macro commands. Macro
	commands have to be bu	ilt before configure this function. Please refer to
	related chapter on how to	edit Macros.
Set style	~ Attribute	
	Set style : Toggle	✓
	Execute macro	Macro : ID1 (ID : 1)
		Trigger mode : OFF->ON
		OFF->ON
		OFF<->ON
	When [Set style] is select	ted as <b>[Toggle]</b> , there are three different modes
	to trigger macro command	1. i.e. OFF->ON. ON->OFF. or ON<->OFF.
		.,



# 13.4 Set Word

#### Overview

The **[Set Word]** object provides two operation modes: the "manual operation" mode and the "automatic operation" mode. The "manual operation" mode defines a touch area, and users can activate the area to set the value of the word device. When users select the "automatic operation" mode, the operation will be automatically activated in pre-configured conditions, the touch area has no action in any circumstance.

#### Configuration

Click the **[Set Word]** icon in the toolbar and the **[New Set Word Object]** dialogue box will appear, fill in each items and press **[OK]** button, a new Set Word object will be created. See the pictures below.



New Set Word Object
General Security Shape Label
Description :
- Write address
PLC name : Local HMI V Setting
Address : LW V 0 16-bit Unsigned
Write after button is released.
Notification
Enable Set ON Set OFF
Before writing After writing
PLC name : Local HMI
Address : LB 🗸
Attribute
Set Style : Write constant value
Set value : 12
OK Cancel Help

Setting	Description
Write	Click [Setting] to Select the [PLC name], [Device type], [Address],
address	[System tag], [Index register] of the word device that system set value to.
	Users can also set address in [General] tab while adding a new object.

	Address
	PLC name : Local HMI
	Device type : LW
	Address : 0 System tag
	Address format : DDDDD [range : 0 ~ 10500]
	Index register
	16-bit Unsigned 💌
	OK Cancel
	Write after button is released]
	If this function is selected, the operation is activated after button is touched
	and released, otherwise, if not selected, operation will be activated once the
	button is touched.
Notification	When this function is selected, in the "manual operation" mode, the state of
	the designated bit device will be set to [ON] or [OFF] after/before the
	operation is completed.
	[Before writing] / [After writing]
	Set the state of the designated bit device before or after writing to word
	device.
	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of the Notification bit that system set value to.
	Users can also set the address in the Notification area.
	Address
	PLC name : Local HMI
	Device type : LB
	Address : 0 System tag
	Address format : DDDDD [range : 0 ~ 11999]
	Index register
	OK Cancel







	Set Style :	Press and hold incremen	t (JOG++)	*
	Inc. value :	1	Upper limit : 10	
	JOG delay :	1.0 second (s) 🛛 👻	JOG speed : 0.5 second(s)	~
e. Press	and hold in	crement (JOG	·)	
Press an	d hold decr	ement function.	When the touch and	hold ge
than the	time set in	I [JOG delay],	the value of the wor	d devic
subtracte	d by the val	ue set in <b>[Dec. v</b>	alue] at the speed set	in <b>[JO</b> (
and the f				ŀ
	A their sets			
	Set Style :	Press and hold decremen	ıt (JOG)	~
	Dec. value :	1	Bottom limit : 0	
	IOG deleve	1.0 mond (a)	IOG meed : 0.5 moond (a)	
	JOU delay .		soo speed . U.S second(s)	
Periodica	ally incremer	nt function. A set	word object can use f	the inte
Periodica [ <b>Time int</b> value of limit].	ally incremer t <b>erval]</b> and t the word de	nt function. A set he value set in <b>[l</b> evice, and the re	word object can use t nc. value] to automations and won't exceed the	the inte cally inc value
Periodica [ <b>Time int</b> value of limit].	ally incremer t <b>erval]</b> and t the word de	nt function. A set he value set in <b>[l</b> evice, and the re	word object can use f n <b>c. value]</b> to automations won't exceed the	the inte cally inc value
Periodica [Time inf value of limit].	ally increment terval] and t the word de Attribute	nt function. A set he value set in <b>[l</b> evice, and the re	word object can use f nc. value] to automatic sult won't exceed the	the inte cally inc value
Periodica [Time inf value of limit].	ally increment terval] and t the word de Attribute Set Style : [ Inc. value :	nt function. A set he value set in <b>[l</b> evice, and the re Periodic JOG++ (up->0->	word object can use f nc. value] to automatic sult won't exceed the up->)	the inte cally inc value
Periodica [Time int value of limit].	ally increment terval] and to the word de Attribute Set Style : [ Inc. value : [	nt function. A set he value set in <b>[l</b> evice, and the re Periodic JOG++ (up->0-> 1	word object can use f nc. value] to automatic sult won't exceed the up->) Upper limit : 0	the inte cally inc value
Periodica [Time int value of limit].	ally increment terval] and to the word de Attribute Set Style : [ Inc. value : [ Time interval : [	nt function. A set he value set in <b>[l</b> evice, and the re Periodic JOG++ (up->0-> 1 1.0 second (s)	word object can use f nc. value] to automatic sult won't exceed the up->) Upper limit : 0	the inte cally inc value
Periodica [Time inf value of limit]. g. Auton	ally increment terval] and to the word de Attribute Set Style : [ Inc. value : [ Time interval : [	nt function. A set he value set in [li evice, and the re Periodic JOG++ (up->0-> 1 1.0 second (s)	word object can use f nc. value] to automatic sult won't exceed the up->) Upper limit : 0	the inte cally ind value
Periodica [Time inf value of limit]. g. Auton	ally increment terval] and to the word de Attribute Set Style : [ Inc. value : [ Time interval : [ Thatic JOG++	nt function. A set he value set in [li evice, and the re Periodic JOG++ (up->0-> 1 1.0 second(s)	word object can use f nc. value] to automatic sult won't exceed the up->) Upper limit : 0	the inte cally inc value
Periodica [Time inf value of limit]. g. Auton Periodica	Ally increment terval] and t the word de Attribute Set Style : [ Inc. value : [ Time interval : [ hatic JOG++ ally decreme	nt function. A set he value set in <b>[</b> evice, and the re Periodic JOG++ (up->0-> 1 1.0 second (s) • •	word object can use f nc. value] to automation sult won't exceed the up->) Upper limit : 0	the inte cally inc value
Periodica [Time inf value of limit]. g. Auton Periodica [Time inf	Ally increment terval] and t the word de Attribute Set Style : [ Inc. value : [ Time interval : [ hatic JOG++ ally decrement terval] and t	nt function. A set he value set in [li evice, and the re Periodic JOG++ (up->0-> 1 1.0 second (s) + nt function. A set he value set in [li	word object can use f nc. value] to automation sult won't exceed the up->) Upper limit : 0 t word object can use f nc. value] to automation	the inte cally inc value
Periodica [Time inf value of limit]. g. Auton Periodica [Time inf value of	Attribute Set Style : [ Inc. value : [ Time interval : [ Attribute Set Style : [ Inc. value : [ Attribute Inc. value : [ Attribute Set Style : [ Inc. value : [ Attribute Inc. value : [ Attribu	nt function. A set he value set in [li evice, and the re Periodic JOG++ (up->0-> 1 1.0 second (s) • • • • • • • • • • • • • • • • • • •	word object can use f nc. value] to automatic sult won't exceed the up->) Upper limit : 0 t word object can use f nc. value] to automatic sult won't exceed the	the inte cally inc value the inte cally inc value



Set Style : Automatic JOG++ (up to high limit)         Inc. value : 0         Upper limit : 10	
Time interval : 0.5 second (s)	
n. Automatic JOG	
Derived a program on trunction. A part word a biggt can use the interval	oot in
Fendulcally decrement function. A set word object can use the interval	set in
the value of the word device, and the result won't go less than the value	lue in
[Bottom limit].	
F=	
A theibute	
Set Style : Automatic JOG (down to low limit)	
Dec. value : 1 Bottom limit : 0	
Time interval : 1.0 second (s)	
I. Periodical bounce	
Periodically bouncing function. A Set word object will add the value s	set in
[Inc. value] to the value of the word device with the regulated interval	set in Lond
then subtrast the value set in <b>line</b> , value) from the value of the word d	I, ano Iovico
with the regulated interval set until the result value reaches the value i	in the
[Bottom limit] For example, the value in the word device will ch	nande
periodically from $0 \sim 10$ then from $10 \sim 0$	lange
Attribute	
Set Style : Periodic step up (low to high)	
Low limit : 0 High limit : 10	
Inc. value : 1	
Inc. value : 1 Time interval : 0.5 second (s)	
Inc. value : 1 Time interval : 0.5 second (s)	
Inc. value : 1 Time interval : 0.5 second (s)  j. Periodical step up	
Inc. value : 1 Time interval : 0.5 second(s) j. Periodical step up Stepping up function. A Set word object will add the value set in [Inc. v	alue
Inc. value : 1 Time interval : 0.5 second (s) v j. Periodical step up Stepping up function. A Set word object will add the value set in [Inc. v to the value of the word device with the regulated interval set in [	alue]



value of the word device will return to the value of the **[Low limit]** and then repeat the action to keep the value in an active state. In the example shown below, the value of the word device will change periodically in order of 0, 1, 2,..., 9, 10, 0, 1, 2, .....

Set Style : Periodic step up	(low to high)	~
Low limit : 0	High limit : 10	
Inc. value : 1		

#### k. Periodical step down

Stepping down function. A Set word object will subtract the value set in [Dec. value] from the value of the word device with the regulated interval set in **[Time interval]** until the result value reaches the value of the **[Low limit]**, and the value of the word device will return to the value of the **[High limit]** and then repeat the action to keep the value in an active state. In the example shown below, the value of the word device will change periodically in order of 10, 9, 8,..., 1, 0, 10, 9, 8, .....

	Attribute		
	Set Style : Periodic step down (high	to low) 💌	
	Low limit : 0	High limit : 10	
	Dec. value : 1		
	Time interval : 0.5 second (s)		
		j	
I. Set v	vhen window opens		
When	the window containing the object	t is opened, the value of [S	et value]
will be	5,	, , <b>b</b>	-
	automotically surittan into the sur	rd dovice	
will be	automatically written into the wo	rd device.	
will be	automatically written into the wo	rd device.	
	Attribute	rd device.	
will be	Attribute Set Style : Set when window opens	rd device.	
wiii be	Attribute Set Style : Set when window opens	rd device.	
will be	Attribute Set Style : Set when window opens Set value : 5	rd device.	
m. Set	Attribute Set Style : Set when window opens Set value : 5	rd device.	
m. Set	Attribute Set Style : Set when window opens Set value : 5 When window closes	rd device.	
m. Set	Attribute Set Style : Set when window opens Set value : 5 when window closes	rd device.	
m. Set	automatically written into the wo Attribute Set Style : Set when window opens Set value : 5 when window closes the window containing the object	rd device.	<b>alue]</b> will



Attribute Set Style : Set when window closes Set value : 5 n. Set when backlight on When the backlight is turned from off to on, the value of [Set value] will	
Set Style : Set when window closes Set value : 5 n. Set when backlight on When the backlight is turned from off to on, the value of [Set value] will	
Set value : 5         n. Set when backlight on         When the backlight is turned from off to on, the value of [Set value] will	
n. Set when backlight on	
When the backlight is turned from off to on, the value of <b>[Set value]</b> will	
When the backlight is turned from off to on, the value of <b>[Set value]</b> will	
when the backinght is turned norm on to on, the value of [Set value] will	l be
automatically written into the word device.	
Attribute	
Set Style : Set when backlight on 💉	
Set value : 5	
o. Set when backlight off	
When the backlight is turned from on to off, the value of <b>[Set value]</b> will	l be
automatically written into the word device.	
Attribute	
Set Style : Set when backlight off	
Set value : 5	



# 13.5 Function Key

## Overview

Function key object is used to change base window, pop-up window and close window. It can also be used to design the keypad buttons.

## Configuration

Click the **[Function Key]** icon in the toolbar and the **[Function Key Object's Properties]** dialogue box will appear, fill in each items and press the **[OK]** button, a new function key object will be created. See the pictures below.





New Function Key Object	
General Security Shape Label	
Description :	
Activate after button is released	
<ul> <li>Change Full-screen whitdow</li> <li>Change common whitdow</li> <li>Display popup window</li> </ul>	
Window no. : 50. Keypad 1 - Integer	<b></b>
Return to previous window	
ASCII/UNICODE mode	
○ [Enter] ○ [Backspace] ○ [Clear] ○ [E∞]	
○ [ASCII] / [UNICODE]	
O Execute macro	
O Window title bar	
Hard copy screen to USB disk or printer	
Screen hard copy	
~ Notification	
Enable OSet ON OSet OFF	
PLC name : Local HMI 🗸 Se	tting
Address : LB 🗸 0	
OK Cancel	Help

Function Key object provides the following operation modes:

Setting	Description
[Active after	If this function is selected, the operation is activated when touched and
button is	released. If the function is not selected, the operation is activated once being
released]	touched.
[Change	Change base window.
full-screen	<b>NOTE:</b> Do not use this function to pop up the window which has been opened
window]	by direct / indirect window object.



[Change	Change common window; refer to the "windows" chapter for related						
common	information.						
window]							
[Display popup	Pop up window. The pop up window must be on the top of the base window.						
window]	There is a [Close this popup window when parent window is closed]						
	option with this function, see the picture below; when the function is selected,						
	the pop up window will be closed when executing change base window.						
	Otherwise, users have to set a "Close" button on the pop-up window to close						
	the window.						
	💿 Display popup window						
	Close this popup window when change full-screen window						
[Window no.]	This is used to select the window no. when performing [change base window],						
	[change common window], and [pop up the window]						
[Return to	This is used to return to the previous base window. Fox example, when						
previous	changing window 10 to window 20, users can use this function to return to						
window]	window 10. This function is only available for base window change.						
[Close window]	Close the pop-up windows on the top of the base window.						
Items in	[ASCII/UNICODE mode] is used as elements to configure a keypad, the						
ASCII/UNICODE	keypad is used where numbers or texts are needed to be input to the						
mode	[numeric input] object or [ ASCII input] object. Refer to the "Designing and						
	Using Keypad" chapter for detailed information.						
	ASCII/UNICODE mode						
	[Enter]						
	Same as the keyboard's "enter" function.						
	[Backspace]						
	Same as the keyboard's "backspace" function.						
	[Clear]						
	To clear the temperate input alphanumeric strings stored in the buffer.						
	[Esc]						
	Same as the [Close window] function, it is used to close the keyboard						
	window.						
	[ASCII/UNICODE]						
	To set the characters that are input in the numeric input object and the ASCII						
	input object. Digital characters such as 0, 1, 2… or ASCII characters like a, b,						



	c,etc. are available selection.
[Execute	Macro commands are executed with this selection. Macro commands have to
Macro]	be built before users choose this function. Please refer to related chapter on
	how to edit Macros.
	Everythe maxim Maxim : maxim 1 (ID : 1)
[Window title	A [function Key] which is defined as Window Title Bar can move the popup
bar]	window position on the screen. Firstly users can select the popup window that
	has the title bar, and then click another position to move the window.
	Note: this function is only available on indirect/direct window when [no
	title bar] is selected.
	Colorithe a find with her firstly.
	Select the window title bar firstly. Touching the screen for the new position the popup window will be moved.
Screen hard	Hardcopy current display screen to the printer connected with MT8000. Before
copy]	using this function, please choose printer model in [System Parameter] /
	[Model] / [printer]. If printer does not support color print, user can select
	grayscale to have a better printout effect. Black and white is for improving text
	printing quality.
	Screen hard copy     Printer: HP PCL Series (USB)
	black and white
	Notification grayscale
	Enable
Notification	When the function is selected, MT8000 will set the state of the designated bit
	device to [ON] or [OFF] after the action is completed.
	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of the Notification bit that system set value to.



Users can also set the address in the Notification area.

## Non-ASCII character input

Below we illustrate the method to input non-ascii character such as Traditional Chinese, Simplified Chinese, Japanese, Greece and so on.

#### Step1: Setting non-ascii fonts

Go to System parameter/Font and add non-ascii fonts in the "Fonts for non-ascii strings" list. For example, use "AR MinchoL JIS" for Japanese, " AR MingtiM GB" for Simplified Chinese, " AR MingtiM KSC" for Korean, " Arial" for Greek, please refer illustration below.



#### Step2: Design non-ascii input keypad

Create "window11" for non-ascii input keypad, keypad design is shown below.



Those objects on the window are function keys with input code in accord with the label. For example, to input "简" function key, create a function key object/General/[ASCII]/[UNICODE] mode, type in "简" in the column as below illustration.



2 🖨 🤋 📢 (	Function Key Object's Properties
9 ⊡ ☆ ‱ &	General Security Shape Label Profile
<mark>∨</mark> A* A* ≣ ( Ð E & ∃ m	Description :
■ FK-2 BAC FK_9 FK_9	<ul> <li>Change full-screen window</li> <li>Change common window</li> <li>Display popup window</li> </ul>
	Return to previous window     Close window     ASCU/UNICODE mode
	◯ [Enter] ◯ [Backspace] ◯ [Clear] ◯ [Esc]
	⊙ [ASCII] / [UNICODE] 简

Go to Function key/Label and then select "Use label", type "简" in the content and in the Attribute/Font select " AR MingtiM GB", it must be the same as setp1's setting, as illustrated below.

The label of non-ascii function key must use the same Font. For example, in Simplified Chinese keypad, the fonts all use " AR MingtiM GB".

Jeneral Security Shape Label Profile	
Use label	
Use label library	
State : 0 V I I O O	ו
Attribute	·
Font : AR MingtiM GB	
Color : Color :	Size : 16
Font : AR MingtiM GB Color : Align : Center	Size : 16 Blink : None



After complete the keypad configuration, add window11 into System Parameters / General / keyboard as illustration below.

ystem Parameter Settings						
Device Model General S	Security Font	Extend	ed Memory	Printer	Server	
-Fast selection button Attribute :	Disable	~				
Screen saver Back light sour :	Nene		vioute (c)			
Screen saver :	None	u 🗸	ninute(s)			
Option			.,			
Startup window no. :	10. WINDOW_(	010				~
Extra. no. of events :	0		Common wind	.ow : [	Above base window	~
Keyboard caret color :	<u> </u>		Object lay	out : ]	Nature	~
Keyhoard	🔽 R W_A enable	ed				
	50. Keypad 1 - 1 51. Keypad 2 - 1 52. Keypad 3 - 1 53. Keypad 4 - 1 54. Keypad 5 - 1 55. Keypad 6 - 1 56. Keypad 7 - 1 57. Keypad 8 - 1 60. ASCII Midd 61. ASCII Small 11. SimpleChine	nteger nteger nteger nteger nteger tex HEX Floating le ese_Keybo	bard		Add Delete	



# 13.6 Toggle Switch

## Overview

Toggle Switch object is a combination of bit lamp object and set bit object. The object can be used not only to display the state of a bit device but also to define a touch area, when activated, the state of the bit device will be set to "ON" or "OFF".

## Configuration

Click the "Toggle Switch" icon on the toolbar and the "New Toggle Switch Object" dialogue box will appear, fill in each item and press OK button, a new toggle switch object will be created. See the pictures below.





New Toggle Switch Object
General Security Shape Label
Description :
Read address
PLC name : Local HMI Setting
Address : LB 🗸 0
Invert signal
Write address :
PLC name : Local HMI
Address : LB 🛛 🗸 0
Write when button is released
Attribute Switch style : Toggle
Macro Execute macro
OK Cancel Help

Setting	Description
Read	Click [Setting] to Select the [PLC name], [Device type], [Address],
address	[System tag], [Index register] of the bit device that control the display
	of toggle switch state. Users can also set address in General tab while
	adding a new object.
	[Invert signal]
	Display shape with inverse state; for example, the present state is "OFF", but it displays the shape of "ON" state.
Write	Click [Setting] to Select the [PLC name], [Device type], [Address],



address	[System tag], [Index register] of the bit device that system set value to.				
	The write address can be the same as or different from the read address.				
	Users can also set address in General tab while adding a new object.				
	[Write when button is released]				
	If this function is selected, the operation is activated at touch up. If the				
	function is not selected, the operation is activated at touch down.				
Attribute	This is used to select the operation mode. The available operation				
	modes for selection include "Set ON", "Set OFF", "Toggle",				
	and "Momentary". Refer to the illustrations in the "Set Bit Object" section				
	of this chapter for related information.				
Macro	Users can execute macro command by trigging toggle switch This				
	function is the same as that of set bit object. Please refer to "the chapter				
	of set bit object".				



# 13.7 Multi-State Switch

#### Overview

Multi-State Switch object is a combination of word lamp object and set word object. The object can be used not only to display the state of a word device but also to define a touch area, when activated, the value of the word device can be set.

#### Configuration

Click the "Multi-State Switch" icon on the toolbar and the "New Multi-State Switch Object" dialogue box will appear, fill in each items, and click OK button, a new Multi-State Switch object will be created. See the pictures below.



	WE	INT	EK

New Multi-State Switch Object
General Security Shape Label
Description :
Read address
PLC name : Local HMI Setting
Address : LW 🗸 0 16-bit Unsigned
Write address :
PLC name : Local HMI Setting
Address : LW 🗸 0 16-bit Unsigned
Write when button is released
Attribute
Switch style : JOG+ No. of states : 1
Cyclical : Disable
User-defined mapping
OK Cancel Help

Setting	Description			
[Mode]/	There are "Value" and "LSB" display mode. Refer to the "Word Lamp			
[Offset]	Object" section of this chapter for related information.			
Read	Click [Setting] to Select the [PLC name], [Device type], [Address],			
address	[System tag], [Index register] of the word device that controls the			
	display of multi-state switch.			
	Users can also set address in General tab while adding a new object.			
Write	Click [Setting] to Select the [PLC name], [Device type], [Address],			
address	[System tag], [Index register] of the word device that system set value			







# b. "JOG-"

When the Multi-State Switch object is activated, the value of the write address will be subtracted by 1. In the "Value" display mode, if the resulting value is smaller than the value of [Offset] and "Enable" in
[Cyclic] is selected, the value of the register will change to ([No. of
states] – 1) + [Offset] and shows the state ([No. of states] – 1); otherwise
the value of the word device will remain in [Offset] and shows the state 0.
[User-defined mapping]
Users can modify the value of state, illegal input and error notification.
Remain current state: if input an illegal value, multi-state switch will remain current state.
Jump to error state: if input an illegal value, multi-state switch will jump to error state.



# 13.8 Slider

## Overview

The slide object can be used to create a slot area that changes the word's value by dragging the pointer.

## Configuration

Click the "Slide object" icon on the toolbar and the dialogue box will appear, fill in each items and click OK button, a new slide object will be created. See the pictures below.



eneral Outline S	ecurity Shape	1		
Description :				
Write address	(3)		(A)	
PLC name :	Local HMI		~	Setting
Address :	LW	<b>v</b> 0		16-bit Unsigned
PLC name :	Local HMI	0	~	Setting
Address :	LB	BOOGRAF AND		
Address : Watch address	LB			
Address : - Watch address	Enable	60000 E		
- Watch address - PLC name :	Enable		V	Setting

Setting	Description
Write	Click [Setting] to Select the [PLC name], [Device type], [Address],
address	[System tag], [Index register] of the word device that system set value to.



	Users can also set address in General tab while adding a new object.
Notification	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of the Notification bit that system set value to.
	Users can also set the address in the Notification area.
	When this function is selected, the state of the designated bit device can be
	set before/after the operation is completed. There are [ON] and
	[OFF] selection to set the state.
	[Before writing] / [After writing]
	Set the state of the designated register before or after write to the word
	device.
Watch	When sliding, the current value can be displayed in real-time fashion.
address	

New Slider Object 🛛 🔀
General Outline Security Shane
A theilente
Direction : Right Resolution : 1
Low/High limit : O Constant 💿 Address
PLC name : Local HMI Setting
Address : LW V 16-bit Unsigned
Coarse increment Increment : 10
Slider button type Width : 20
Frame : Transparent
Background :
Slot : Transparent
OK Cancel Help



Setting	Description		
Attribute	[Direction]		
	The bar on the slide of	direction, i.e. left, rig	ght, up and down.
	right left	IP down	
	[Resolution]		
	To specify the scale v	alue of the slider, i	f N is the specified minimum
	N=10, the numerical of	display shows only	multiples of 10.
	N=5, the numerical di	splay shows only n	nultiples of 5.
	IN=1, the numerical di	isplay shows only n	
[Low limit &	a. Constant		
High limit]	The low limit and hig	h limit of the word	device is set as constant value.
	i.e. [Input low] and [In	iput high].	
	<ul> <li>b. Address</li> <li>The low / high limit of the word device is controlled by a designate address.</li> <li>Click [Setting] to Select the [PLC name], [Device type], [Address</li> <li>[System tag], [Index register] of designated address or users c also set address in Attribute.</li> </ul>		
	Low/High limit : 🔘	) Constant 💿 Addres	8
	PLC name : Lo	ocal HMI	Setting
	Address : L	w 🔽 O	16-bit Unsigned
	V	Coarse increment	Increment : 10
	Control address	Low Limit	High Limit
	16-bit format	Address+0	Address+1
	32-bit format	Address+0	Address+2
	[Coarse increment:]		
	If this option is sele	ected, the word va	alue will increase/decrease one
	[increment] value for	everv touch activat	ion. If not, the word value will be



	set the value in accord with the touch activated point.		
Slider button	There are four slider button types for selection. You also can adjust the		
type	width of moving piece.		
Color	This is used to select slide object frame, background and slot's color.		
	Slider button type Slot SLO Frame Background		



# 13.9 Numeric Input and Numeric Display

#### Overview

Both of the Numeric Input object and the Numeric Display object can be used to display the value of the word devices. The difference is the numeric input object can be used to input data from the keypad, the input value is written to the designated word devices.

#### Configuration

Click the "Numeric Input" or "Numeric Display" icon on the toolbar and the "New Numeric Input Object" or "New Numeric Display Object" dialogue box will appear, fill in each item, click OK button and a new "Numeric Input Object" or "Numeric Display Object" will be created. See the pictures below.



The difference between the "New Numeric Display Object" and "New Numeric Input Object" dialogue boxes is that the latter has the settings for "Notification" and keypad input while the former doesn't have. The picture below shows the [General] tab in "New Numeric Input Object".



New Numeric Input Object	×
General Data Entry Numeric Format Security Shape Font	
Description :	
Peed (Write use different addresses	
- Read address	
PLC name : Local HMI Setting	
Address : LW 🗸 0	
With address	
PLC name : Local HMT	
Address : LW V 10	
Notification	
Before writing	
PLC name : Local HMI	
Address : LB V	
- Notification on invalid input	
🗹 Enable 💿 Set ON 🚫 Set OFF	
Address : LB 20	

Setting	Description
Read/Write	Numeric Input object is provided with [Read/Write use different addresses]
use	selection, users can set different addresses for Read and for Write data.
different	
address	
Read	Select the [PLC name], [Device type], [Address] of the word device that
address	system display its value and write new data to it.
Write	Select the [PLC name], [Device type], [Address] of the word device that



address	system writes to.
Notification	When this function is selected, the state of the designated bit device will be
	set to [ON] or [OFF] after/before the value of the register is changed successfully.
	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of the Notification bit that system set value to.
	Users can also set the address in the Notification area.
	[Before writing] / [After writing]
	Set the state of the designated bit device before or after update the word device.
Notification	When inputting invalid values, it can now automatically set the status of
on valid	designated address.
input	



New Numeric Input Object	X
General Data Entry Numeric Format Security Shape Font	
Mode : Touch 💙	
~ Input order	
Enable	
Input order : 1 😴 Group 2 📚	
Stop sequential input function after input	
Kaubaard	
Use a popup keypad	
Hide title bar	
Window no. : 50. Keypad 1 - Integer	
Popup position :	
{relative to HMI screen}	
Hint : If the keyboard is an USB keyboard, on indirect/direct window, or on the same window, please don't check "Use a popup keypad".	
Restart the keypad if input value is out of range	

Setting	Description	
[Mode]	• [Touch]	
	The object enters input state when a user touches it.	
	• [Bit control]	
	The object enters input state when turning ON the	
	designated bit register, and ends input state when turning	
	OFF. Notice that if there is another input object already in	
	input state, turning ON the designated bit register won't	
	make this input object enters input state until the previous	
	one ends inputting data.	
Allow input bit	Click [Setting] to Select the [PLC name], [Device type],	
address	[Address], [System tag], [Index register] of the bit register	
	that controls the object enters and ends input state.	
	Users can also set address in Data Entry tab.	

Input order	By setting Input Order and Input Order Group, users can continuously input data between multiple input objects. The system will automatically transfer input state to the next input
	<ul> <li>• Enable Select [Enable] and set Input Order to enable this feature. Furthermore, users can also select [Group] to set Input Order Group.</li> <li>a. The range of Input Order: 1 ~ 511.</li> <li>b. The range of Input Order Group: 1 ~ 15.</li> <li>c. The Input Order Group of an input object with [Group] unselected is 0.</li> </ul>
	<ul> <li>Criterion of searching the next input object <ul> <li>a. The system only searches it among the input objects with the same Input Order Group.</li> <li>b. The system picks the input object with smaller Input Order to enter input state before another one with bigger Input Order.</li> <li>c. If two input objects have the same Input Order Group and Input Order, the system picks the one at bottom layer to enter input state first.</li> </ul></li></ul>
	• When selecting [Touch] as Mode Refer to the following illustration, when users complete inputting data on "AE_2", the system transfers input state to "AE_0". The reason why not transferring to "NE_0" is because the Input Order Group of "NE_0" is different from that of "AE2".



WE!NTEK

	EasyBuilder8000 User's Manual
	Input order
	<ul> <li>When selecting [Bit control] as Mode <ul> <li>Users have to specify an Input Order for the object.</li> <li>No need to set Input Order Group because all the input objects with [Bit control] as Mode have the same Input Order Group that is different from any input object with [Touch] as Mode.</li> </ul> </li> </ul>
Keyboard	• Select [Use a popup keypad] Specify the pop-up position for the keyboard window. The system displays the keyboard window on inputting data and closes it on end.
	<ul> <li>Unselect [Use a popup keypad] The system does not automatically display keyboard window. Users have to complete the input process via following methods:</li> <li>a. Design a custom keypad and place it in the same window with the input object.</li> <li>b. Use an external keyboard.</li> </ul>
	Hide title bar
	Reypads without title bar can be selected for Numeric Input / ASCII Input object.
	Restart the keypad if input value is out of range
	For Input Value object, re-input can be automatically requested when input error occurs.



• When selecting [Bit control] as Mode, the system will automatically unselect [Use a popup keypad] in [Keyboard].

The picture below shows the [Numeric Format] tab, included in both of the numeric input object and the numeric display object, which is to set the data display format.

New Numeric Input Object
General Data Entry Numeric Format Security Shape Font
Display Data format : 16-bit Unsigned  Mask
Left of decimal Pt. : 4 Right of decimal Pt. : 0
Scaling option
Engineering low : 0 Engineering high : 9999
Limits
O Direct O Dynamic limits
PLC name : Local HMI Setting
Address : LW 0
Use alarm color
Low limit :
OK Cancel Help

Setting	Description
Display	[Data format]
	To select the data format of the word device designated by the "Read


	address". The selec	ction list is shown as follows:
	Format	
	16-bit BCD	
	32-bit BCD	
	16-bit Hex	
	32-bit Hex	
	16-bit Binary	
	32-bit Binary	
	16-bit Unsigned	
	16-bit Signed	
	32-bit Unsigned	
	32-bit Signed	
	32-bit Float	
	[Mask]	
	When the data is di	splayed, "*" will be used to replace all digitals and the color
	warning function wi	ll be cancelled.
Number of	[Left of decimal Pt	]
digits	The number of digit	s before the decimal point.
	The number of digit	s after the decimal point.
Scaling	[Do conversion]	
option	The data displayed	on the screen is the result of processing the raw data from
	the word address of	designated by the "Read address." When the function is
	selected, it is requir	red to set [Engineering low], [Engineering high], and [Input
	low] and [Input high	I] In the "Limitation". Supposed that "A" represents the raw
	В =	[Engineering low] + (A - [Input low]) × ratio
	where, the ratio =	([Engineering high] - [Engineering low]) / ([Input high] -
	See the example	in the picture below, the raw data is 15, after being
	converted by the al	bove formula as $10 + (15 - 0) \times (50 - 10) / (20 - 0) = 40$ ,
	and the result "40" v	will be displayed on the numeric input object.



	Scaling option		
		Do conversion	
	Engineering lo	w: 10 Eng	gineering high : 50
	Limits		
	💽 Direct	🔵 Dynamic limits	
	Input low : 0	I	nput high : 20
l imits	To set the source of the	range for the input	data and to set the warning color
Linits	effect	range for the input	
	[Direct]		
	The low limit and high lin	nit of the input data	can be set in [Input low] and [Input
	high] respectively. If the	input data is out of	the defined range, the input value
	will be ignored.		
	[Dynamic limits]		
	Limits		
	Direct	O Demonia limite	
	PLC name : Lo	cal HMI	Setting
	Address : LV	v 🔽 O	
	Set the low limit and h	high limit of the in	uput data to be derived from the
	designated register. The	data length of the	designated register is the same as
	the input object itself. In	the above examp	le the low limit and high limit are
	derived from [LW100] an	nd the following exp	lains the usage of the low limit and
	high limit from designate	d address.	
	Click [Setting] to Selec	ct the [PLC name],	[Device type], [Address],
	[System tag], [Index re	gister] for designa	ted register.
	Users can also set addre	ess in Numeric For	mat tab.
	Designated address	Input Low Limit	Input High Limit
	16-bit format	LW100	LW101 (Address+1)
	32-bit format	LW100	LW102 (Address+2)
	[Low limit]		
	When the value of the F	PLC's register is sn	naller than [Low limit], the value is



displayed with pre-defined color.

#### [High limit]

When the value of the PLC's register is larger than [High limit], the value is displayed with pre-defined color..

#### [Blink]

When the value of the PLC's register is smaller than [Low limit] or larger than [High limit], the object will display data with Blinking. The picture below shows the [Font] tab, available in both of the numeric input object and the numeric display object to set font, font size, color, and aligning mode.

Numeric Input Object's Properties	×
General Numeric Format Security Shape Font Profile	
Attribute Font : Comic Sans MS	
Color : Size : 12	
Align : Left	

Setting	Description
Attribute	[Color]
	When the data is within high and low limit, it will be displayed with this color.
	[Align]
	There are three aligning modes: "Left", "Leading zero", and "Right". The
	picture below shows the style of each mode.



Left 12
Leading zero 0012
Right 12
[Size]
Set font size.



# 13.10 ASCII Input and ASCII Display

### Overview

Both of the ASCII Input object and the ASCII Display object can display the value of the designated word devices in ASCII format. The ASCII input object can also accept the data input from the keypad and change the value of the word devices.

### Configuration

Click the "ASCII Input" or "ASCII Display" icon on the toolbar and the "New ASCII Input Object" or "New ASCII Display Object" dialogue box will appear, fill in each item, press OK button, a new "ASCII Input Object" or "ASCII Display Object" will be created. See the pictures below.



The difference between the "New ASCII Display Object" and "New ASCII Input Object" dialogue boxes is that the latter has the settings for "Notification" and keypad input while the former doesn't have. The picture below shows the [General] tab of the "New ASCII Input Object".

WE!NTEK

General Data Entry Security Shape Font
Description :
Mask UNICODE Reverse high/low byte
PLC name : Local HMI
Address : LW V
_ Notification
Enable Oset ON OSet OFF
Before writing After writing
PLC name : Local HMI
Address : LB V 0
OK Cancel Help

Setting	Description
[Mask]	When the data is displayed, "*" will be used to replace all texts.
[Use	Click "Use UNICODE" to display data in UNICODE format. Otherwise the
UNICODE]	system displays the character in ASCII format. This feature can be used with
	function key [UNICODE]. Not every Unicode has corresponding font stored
	in the system. The font of UNICODE is only available for those Unicode
	character that registered function key.
[Reverse	In normal condition, the ASCII code is displayed in "low byte", "high byte"
high/low byte]	order. The reverse selection makes the system display ASCII characters in
	"high byte", "low byte" order.



Read address	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of the word device that system display its
	value and write new data to it.
	Users can also set address in General tab while adding a new object.
	[No. of words]
	To set the length of ASCII data in the unit of words. Each ASCII character
	take one byte, each word contains two ASCII characters.
	In the example shown below, the object will display 3 * 2 = 6 characters.
	No of words: 3
	abbdef
Notification	When this function is selected, the state of the designated bit device will be
	set to [ON] or [OFF] after/before the value of the register is changed
	successfully.
	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of the Notification bit that system set value
	to.
	Users can also set the address in the Notification area.
	[Before writing] / [After writing]
	Set the state of the designated bit device before or after update the word
	device.

About the Data Entry tab, please refer to "Numeric Input and Numeric Display" section.





New ASCII Input Object
General Security Shape Font
0 their to
Font · Drial
Color : Size : 16
Align : Left 💌
Content :
OK Cancel Help

Setting	Description
Attribute	The picture shows the [Font] tab of the ASCII Input object and the
	ASCII display object. Users can set the font, font size, font color, and
	aligning mode.
	Attribute
	Font : Comic Sans MS
	Color : 🗾 Size : 12 🖌
	Align : Left



[Align]
There are two aligning modes: "Left" and "Right". The picture below
shows how each mode performs.
Left alignment ab
bde
<i>Right alignment</i> ab
bde
Due
[Size]
Set font size.



# 13.11 Indirect Window

#### Overview

"Indirect Window" object is to define a popup window location (position / size) and a word device. When the content of the word device is written a valid window number, the window will be popup in the predefined location. The popup window will be closed when the value of the word device is reset (0). The system will only take action when the content of word device is changed. (0  $\rightarrow$  valid window number, nonzero  $\rightarrow$  0, A  $\rightarrow$  B valid window number).

### Configuration

Click the "Indirect Window" icon on the toolbar and the "New Indirect Window Object" dialogue box will appear, fill in each items, click OK button, a new "Indirect Window Object" will be created. See the pictures below.





New Indirect Window Object
General
Description :
Read address
PLC name : Local HMI Setting
Address : LW 🗸 0 16-bit Unsigned
Attribute Stude : No title has
OK Cancel Help

Setting	Description		
Read	Click [Setting] to Select the [PLC name], [Device type], [Address],		
address	[System tag], [Index register] of the word device that control the		
	window popup.		
	Users can also set address in General tab while adding a new object.		
Attribute	[Style]		
	To set the display style of the popup window. There are two styles, "No		
	title bar" and "With title bar".		
	a. "No title bar"		
	The popup window does not have title bar, and its position is fix as		
	predefined in configuration.		



WINDOW 35
b. "With title bar"
The popup window contains title bar, and its position can be dragged at
online operation.
WINDOW 35

#### Example to use indirect window

Here is a simple example to illustrate indirect window object. The pictures show how to configure an indirect window and use the word device [LW100] to change the popup window.

	CRead address	
WP_0	PLC name : Local HMI	Setting
	Address : LW 💙 0	16-bit Unsigned
SW_0	Set constant 35 to LW100	
SW_1	Set constant 36 to LW100	
SW_2	Set constant 0 to LW100	
	→ 34 → *35: WINDOW_035 → *36: WINDOW_036 → 37 → 38	



Use the set word object SW\_0 to set the value of [LW100] as 35, and the location of indirect window will display window 35.



Use the set word object SW\_1 to set the value of [LW100] as 36, and the location of indirect window will display window 36.



No matter window 35 or 36 is displayed on the indirect window location, press SW\_2 to set the value of [LW100] to 0 will close the popup window. The other way to close the popup window from indirect window object is to configure a function key with [close window]. Once you press the function key, the popup window will be closed.



**NOTE:** Only 16 windows maximum can be displayed simultaneously at run time, and do not use this function to open the window when the same window has been opened by function key or direct window.



# 13.12 Direct Window

### Overview

"Direct window" object is to define a popup window location (position / size), a bit device and a predefined valid window number. When the content of the bit device is set ON/OFF, the window will be popup in the predefined location. The popup window will be closed when the content of the bit device is reset. The system will only take action when the content of bit device is changed (OFF  $\rightarrow$  ON, ON  $\rightarrow$  OFF).

The difference between the "Direct window" and the "Indirect window" is that the direct window object sets the popup window in configuration. When system is in operation, users can use the state of the designated register to control popup or close the window.

### Configuration

Click the "Direct Window" icon on the toolbar and the "New Direct Window Object" dialogue box will appear, fill in each items, press OK button, and a new "Direct Window Object" will be created. See the pictures below.





ew Direct Window Object	×
General	
Description :	
Trigger: ON	
Read address	
PLC name : Local HMI Setting	
Address : LB 💙 0	
C Attribute	
Style : No title bar 💙	
Window No. : 3. Fast Selection	
OK Cancel Hel	р

Setting	Description
Read	Click [Setting] to Select the [PLC name], [Device type], [Address],
address	[System tag], [Index register] of the bit device that control the window
	popup.
	Users can also set address in General tab while adding a new object.
Attribute	[Style]
	Refer to the "Indirect Window Object" for related information.
	[Window no.]
	Set the popup window number.

# Example to use direct window



Here is an example to explain how to use the direct window object. The picture below shows the settings of the direct window object. In the example, use [LB10] to call up the window 35.

WC 0	Read address ——		
NC_0	PLC name :	Local HMI	Setting
	Address :	LB 🔽 0	
TS_0	[		
Toggle Switch			
Read address : LB10,			
Write address : LB10			
Mode : "Toggle"	Attribute		,
	Style :	No title bar 💌	
	Window No. :	35. WINDOW_035	~

When the state of LB10 is set to ON, the window 35 will be popup; when the state of LB10 is OFF, the window 35 will be closed. See the picture below.







**NOTE:** Only 16 windows maximum can be displayed simultaneously at run time, and do not use this function to open the window when the same window has been opened by function key or direct window.



# 13.13 Moving Shape

### Overview

Moving Shape object is used to define the object's state and moving distance. The Moving Shape object is used to place an object in a window at a location specified by the PLC. The state and the absolute location of the shape in the window depend on the current values of three continuous PLC registers. Typically, the first register controls the state of the object, the second register controls the horizontal position (X), and the third register controls the vertical position (Y).

### Configuration

Click the "Moving Shape" icon on the toolbar and "New Moving Shape Object" dialogue box will appear, fill in each items, press OK button, and a new "Moving Shape Object" will be created. See the pictures below.





New Moving Shape	• Object
General Shape I	abal
Source Support	AUG1
Description :	
PLC name :	Local HMI 💌
Read address	
PLC name :	Local HMI Setting
Address :	LW V 100 16-bit Unsigned
Attribute	
Mode :	X axis only
No. of states :	1
Min. X :	0 Max. X : 400
Display ratio	
State :	0 Ratio : 1
Limit address	
	Limit from register
PLC name :	Local HMI Setting
Address :	LW 0 16-bit Unsigned
	OK Cancel Help

Setting	Description
Read	Click [Setting] to Select the [PLC name], [Device type], [Address],
address	[System tag], [Index register] of the word devices that control the display
	of object's state and moving distance.
	Users can also set address in General tab while adding a new object.
	The table below shows the address to control object's state and moving
	distance in each different data format.
	•



Data format	Address to control object state	Address to control Moving Distance on the X-axis	Address to control Moving distance on the Y-axis
16-bit format	Address	Address + 1	Address + 2
32-bit format	Address	Address + 2	Address + 4

For example, if the object's read address is [LW100] and the data format is "16-bit Unsigned", [LW100] is to control the object's state, [LW101] is to control the object's moving distance on the X-axis, and [LW102] is to control the object's moving distance on the Y-axis.

The picture below shows that the object's read address is [LW100] and initial position is (100, 50). Supposed you want the object moved to the position (160, 180) and be displayed in the shape of State 2, the value of [LW100] must be set to 2, [LW101] = 160-100 = 60, [LW102] = 180-50 = 130.





	Attribute Mode : X No. of states : 8 Min. X : 0	axis only	Max. X : 600	
Da	ta format	Address to	Address to	
		control object	control Movin	ng
		state	Distance on t	he
16	bit format	Address		_
32		Address	Address + 2	
		AUUICOO	Tuui 633 T 2	
	Attribute Mode : Y No. of states : 8 Min. Y : 0	axis only	Max. Y : 600	
Da	ta format	Address to	Address to	
		control object	control Movin	ng
		state	Distance on t Y-axis	he
16	-hit format			
10	bit ionnat	Address	Address + 1	
32	-bit format	Address Address	Address + 1 Address + 2	
c. X & Y The obj range in respecti	-bit format axis ect is allowed XY direction vely.	Address Address d to move along the state of the second se	Address + 1 Address + 2 he X-axis and Y-a X], [Max. X] and [	axis. The moving [Min. Y], [Max. Y]
<b>c. X &amp; Y</b> The objurange in respection	-bit format axis ect is allowed XY direction vely.	Address Address d to move along the is defined by [Min.	Address + 1 Address + 2 he X-axis and Y-a X], [Max. X] and [	axis. The moving [Min. Y], [Max. Y]
<b>c. X &amp; Y</b> The obj range in respecti	-bit format axis ect is allowed XY direction vely.	Address Address d to move along the is defined by [Min.	Address + 1 Address + 2 he X-axis and Y-a X], [Max. X] and [	axis. The moving [Min. Y], [Max. Y]
c. X & Y The obj range in respecti	-bit format axis ect is allowed XY direction vely. Attribute Mode : X No. of states : 8	Address Address d to move along the is defined by [Min. & Y extine	Address + 1 Address + 2 he X-axis and Y-a X], [Max. X] and [	axis. The moving [Min. Y], [Max. Y]
32 c. X & Y The obj range in respecti	-bit format -bit format axis ect is allowed XY direction vely. Attribute Mode : X No. of states : 8 Min. X : 0	Address Address d to move along the is defined by [Min.	Address + 1 Address + 2 he X-axis and Y-a X], [Max. X] and [ Max. X : 600	axis. The moving [Min. Y], [Max. Y]
c. X & Y The obj range in respecti	-bit format -bit format axis ect is allowed XY direction vely. Attribute Mode : X No. of states : 8 Min. X : 0 Min. Y : 0	Address Address d to move along the is defined by [Min.	Address + 1 Address + 2 Address + 2 Address + 2 Address + 2 Address + 2 Address + 1 Address + 1 Address + 1 Address + 1 Address + 2 Address + 2 Addres	axis. The moving [Min. Y], [Max. Y]



	control object state	control Moving Distance on the X-axis	control Moving distance on the Y-axis
16-bit format	Address	Address + 1	Address + 2
32-bit format	Address	Address + 2	Address + 4

#### d. X axis w/ scaling

The object is for X axis movement with scale. Supposed that the value of the designated register is DATA, the system uses the following formula to calculate the moving distance on the X-axis.

#### X axis move distance =

(DATA – [Input low]) \* ([Scaling high – Scaling low]) / ([Input high] – [input low])

Attribute			_
Mode :	X axis w/ scaling	1	/
No. of states :	8 💌		
Input low :	0	Input high : 600	
Scaling low :	300	Scaling high : 1000	

For example, the object is only allowed to move within 0~600, but the range of the register's value is 300~1000, set [Input low] to 300 and [Input high] to 1000, and set [Scaling low] to 0 and [Scaling high] to 600, and the object will move within the range.

Data format	Address to control object state	Address to control Moving Distance on the X-axis
16-bit format	Address	Address + 1

#### e. Y axis w/ scaling

The object is for Y axis movement with scale, and the formula to calculate the moving distance on the Y-axis is the same as the one in "X axis w/ scaling."

Data format	Address to	Address to
	control object	control Moving



				state		Dis	stance on the		
						Y-a	axis		
		16-bit form	nat	Addres	SS	Ad	dress + 1		
		32-bit format Address Address + 2							
	f. X a	f. X axis w/ reverse scaling							
	This	This function is the same as "X axis w/ scaling", but the moving direction is							
	in rev	in reverse.							
	g. Y a	g. Y axis w/ reverse scaling							
	This	function is t	he sa	me as "	Y axis w/ sca	ling	", but the mov	ving direction is	
	in rev	verse.							
Display	The s	size of shap	e in di	fferent	states can be	set	individually a	s shown in the	
ratio	pictu	re below.							
	Ratio : 1 Ratio : 1.2 Ratio : 1.4 Ratio : 1.6								
	5	State 0	St	ate 1	Stat	e	2 St	ate 3	
Limit	The	object's mo	ving r	ange ca	an be set not	t onl	ly by [Min. X]	, [Max. X] and	
address	[Min.	Y] [Max. Y	], but	also by	the designat	ed r	registers. Sup	posed that the	
	objec	t's moving	rang	e is se	et by the va	lue	of the desig	nated register	
	"Address", then the address of [Min. X], [Max. X] and [Min. Y] [Max. Y] are								
	listed in the following table.								
	Da	ta format	[Min.	X]	[Max. X]	[	Min. Y]	[Max. Y]	
			addr	ess	address	a	address	address	
	16-	bit format	Addre	ess	Address + 1	A	Address + 2	Address + 3	
	32-	bit format	Addre	ess	Address + 2	A	Address + 4	Address + 6	



## 13.14 Animation

#### Overview

The Animation object is used to place an object on the screen at a specified location determined by a predefined path and data in the PLC. The state and the absolute location of the shape on the screen depend on current reading value of two continuous PLC registers. Typically, the first register controls the state of the object and the second register controls the position along the predefined path. As the PLC position register changes value, the shape or picture jumps to the next position along the path.

#### Configuration

Click the "Animation" icon on the toolbar, move the mouse to each moving position and click the left button to define all moving positions one by one. When settings of all moving positions are completed, click the right button of the mouse, a new animation object will be created. See the pictures below.



To change the object's attributes, you can double click the left button of the mouse on the object, and the "Animation Object's Properties" dialogue box, as shown in the picture below, will appear.

General       Shape       Label       Profile         Description :
Description : Attribute no. of states : 8 Position : Ocontrolled by register Based upon time interval Read address PLC name : Local HMI Address : LW O I 6-bit Unsigned
Attribute no. of states : 8 Position : O Controlled by register Based upon time interval Read address PLC name : Local HMI Address : LW O 16-bit Unsigned
no. of states : 8 Position : O Controlled by register Based upon time interval Read address PLC name : Local HMI Address : LW O 16-bit Unsigned
Position : Controlled by register Based upon time interval Read address PLC name : Local HMI Setting Address : LW 0 16-bit Unsigned
Read address   PLC name : Local HMI   Address : LW   0     16-bit Unsigned
PLC name : Local HMI Setting Address : LW O O 16-bit Unsigned
Address : LW V 0 16-bit Unsigned
OK Cancel Help

Setting	Description
Attribute	[Total no. of states]
	To set the number of the states for this object.
a. Controlled	When select "Controlled by register", the designated register controls the
by register	object's state and position.
	Read address
	If select "Controlled by register" option, it is necessary to set the read
	address.
	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] for the read address.



Users can also set address in General tab while adding a new object. In the table below, it describes the address that control shape's state and position in different data format. Data Format Address to control Address to control object's state object's position 16-bit format Address + 1 Address Address + 2 32-bit format Address For example, if the designated register is [LW100] and the data format is "16-bit Unsigned", then [LW100] represents object's state, [LW101] represents position. In the picture below, [LW100] = 2, [LW101] = 3, so the object's state is 2 and position is 3. pasitian 5 pasitian O pasitian 1 pasitian 4 pasitian 2 pasitian 3 State 2 3 LW100 LW101 If "Based upon time interval" is chosen, the object automatically changes b. Based status and display location. "Time interval attributes" is to set the time upon time interval interval for states and positions. Time interval attributes Position speed : 10 \*0.1 second(s) Image state change : Time-based ¥ Backward cycle Image update time : 5 \*0.1 second(s) [Position speed] Position changes speed, the unit is 0.1 second. Supposed that [Speed] is set to 10, the object will change its position every 1 second. [Backward cycle] If the object has four positions: position 0, position 1, position 2, and position 3, and [Backward cycle] is not selected. In this case when the



object moves to the last position (position 3), next position will be back to the initial position 0, and repeat the action over again. The moving path is shown as follows:
position 0 $\rightarrow$ position 1 $\rightarrow$ position 2 $\rightarrow$ position 3 $\rightarrow$ position 0 $\rightarrow$ position 1 $\rightarrow$ position 2
If [Backward cycle] is selected, when the object moves to the last position (position 3), it will move backwards to the initial position 0, and repeat the moving mode over again. The moving path is shown as follows.
position 0 $\rightarrow$ position 1 $\rightarrow$ position 2 $\rightarrow$ position 3 $\rightarrow$ position 2 $\rightarrow$ position 1 $\rightarrow$ position 0
[Image state change] State change mode. There are "Position dependant" and "Time-based" options. When "Position dependant" is selected, it means that following the change of position, the state will change too. When "Time-based" is selected, it means that the position will change based on "Position speed" and shape state will change based on "Image update time" Time interval attributes Position speed : 4 *0.1 second (s) Image state change : Position-dependent
Position-dependant Time-based

The following dialog shows size setup of animation object. Call up the animation object dialogue box by double clicking.



nimation Object's	Properties		
General Shape I	abel Profile		
- Position	aber		
Pinned	X : 191 😂	¥: 122	
Size	Midth · A1A	Haight 144	1
	ΥΥΊΜΟΙ. 414 💌	Height . 144 💌	
Shape rectangle s	ize		
	Width : 84 🗢	Height : 33 🛟	
Trajectory			1
	Position 0	*	
	X : 191 🗢	¥: 147 🗢	
		Cancel	Help

Setting	Description
Shape	To set the size of the shape.
rectangle size	
Trajectory	To set the position of each point on the moving path.



# 13.15 Bar Graph

#### Overview

Bar graph object displays PLC register data as a bar graph in proportion to its value.

#### Configuration

Click the "Bar Graph" icon on the toolbar, the "Bar Graph" dialogue box will be shown up, fill in each items of settings, click OK button, a new "Bar Graph Object" will be created. See the picture below.



The following picture shows the "General" tab of the bar graph object.



New Bar Graph Object 🛛 🔀
General Outline Shape
Description :
Read address
Address : Local HMI
Address : LW 0 10-Dic Unsigned
OK Cancel Help

#### **Read address**

Click [Setting...] to Select the **[PLC name]**, **[Device type]**, **[Address]**, **[System tag]**, **[Index register]** of the word devices that controls the bar graph display. Users can also set address in General tab while adding a new object.

The following picture shows the "Outline" tab of the bar graph object.



New Bar Graph Object	X
General Outline Shape	
Attribute	
Type : Normal 🛛 Direction : Up	~
Zero : 0 Span : 10	
Bar width ratio (%) : 100	\$
Bar color/style	$\equiv$
Transparent	
Frame : Background :	-
Bar : Bar style :	
Target indicator	
Enable Color :	-
Target value : 0 + 2 Tolerance : 0	
Alarm indicators	$\leq$
Low limit : 0 High limit : 0 + 1	
Low color : High color : High color :	-
─ Target/alarm/zero(span) dynamic address ───────────────────────────────────	$\leq$
Enable	
PLC name : Local HMI	
Address : LW 🗸 0 16-bit Unsign	ied
	_
Dynamic zero/span	
OK Cancel He	elp

Setting

Description



Attribute	[lype]
	There are "Normal" and "Offset" for selection. When select "Offset",
	there must be a original value for reference. Please refer the illustration
	below.
	Attribute
	Zero: U Span: IU
	Origin : 5 Bar width ratio (%) : 100
	[Direction]
	To select the bar graph direction, and there are "Up", "Down", "Right".
	and "Left" for selection.
	[Zero] \ [Span]
	The filled bar percentage can be calculated with the following formula:
	The filled bar percentage can be calculated with the following formula.
	The filled har percentage = (Register value – Zere) / [Span] – [Zere]) *
	The filled bar percentage – (Register value – Zero) / [Spari] – [Zero])
	100%
	$M/h = r = r = 1 + (Off = r)^{2}$ if (Desister value Zere) > 0, the her will fill us
	when select Offset, if (Register value – $2 \text{ero}$ ) > 0, the bar will fill up
	from origin setting; if (Register value – Zero) < 0, the bar will fill up but
	down side from origin setting.
	For example,
	Origin =5, Span=10, Zero=0 and use different value in read address, it
	will display as illustration below.
	When read address value is 4,
	10
	$high \ limit = 10 \qquad \qquad \_10$
	oriain = 5
	low limit = 5
	4







	[Target Value] - [Tolerance] < = Register value < = [Target Value] + [Tolerance]					
	See the picture below, in here [Target Value] = 5, [Tolerance] = 1, if the register value is equal to or larger than 5-1=4 and equal to or less than $5+1=6$ , the filled area's color of the bar will change to the "Target color"					
	Target value = 5 Tolerance = 1					
Alarm	When register's value is larger than [High limit], the color of filled area					
Indicator	will change to [High color], when register's value is smaller than [Low					
	limit], the color of filled area will change to [Low color].					
Target/Alarm Dynamic Address	When select [Enable], the [Low limit] and [High limit] of "Alarm indicator" and the [Target Value] of "Target indicator" all come from designated register. See the picture below.					



The The	Alarm Low lim	nit is LW 20 / Th or is LW22 / Th	e Alarm High lir e Zero is LW23	nit is LW21 / The Span	is LW
The targe	following table et. The "Addre	e shows the rea	d address of lov device address,	v limit, high for exampl	limit,
l		Dynamic zero/spa			
	PLC name Address	Enable Local HMI UW	20	Sett	ing Insigned
	Low color	:	High color :		
	Low limit	:: 20	High limit :	20 + 1	
	- 4 Jarm indicators	· 20 + 2		U	
	Terretvelue	Enable	Color :		
	- Target indicator -				
	Bar		<ul> <li>Bar style :</li> </ul>		
	-Bar color/style	Transparent	Paakamuud		
			Bar width ratio (%) :	100	*
	Zero	: 20 + 3	Span :	20 + 4	
	Туре	: Normal	Direction :	Up	~
	Attribute				


16-bit	Address	Address +	Address	Address	Address
format		1	+ 2	+ 3	+ 4
32-bit	Address	Address +	Address	Address	Address
format		2	+ 4	+ 6	+ 8



# 13.16 Meter Display

## Overview

The meter display object can display the value of word device with meter.

### Configuration

Click the "Meter Display" icon on the toolbar and the "Meter Display Object's Properties" dialogue box will appear, fill in each items, press OK button, and a new "Meter Display Object" will be created. See the picture below.



The picture below shows the "General" tab in the "Meter Display Object's Properties" dialogue box.



New Meter Display Object
General Outline Limits Shape
Description :
Read address
PLC name : Local HMI
Address : LW 0 16-bit Unsigned
OK Cancel Help

#### **Read address**

Click [Setting...] to Select the **[PLC name]**, **[Device type]**, **[Address]**, **[System tag]**, **[Index register** of the word devices that controls the display of meter. Users can also set address in General tab while adding a new object.



New Meter Display Object
General Outline Limits Shape
Degree
Start degree : 0 🗢 End degree : 360 🗢
Background
Background : Profile :
Full circle
Tick marks
✓ Enable
Color : Coordinate
Main scale : 4 😒 Sub. scale : 2 😂
Length : 15
Pointer
Arm style Frame :
Inner :
Width : 4 💌 Length : 40 🗢
Pin point
Radius : 7
Inner : Frame : Frame :
⊙ Circle ○ Rectangle
OK Cancel Help

In the above dialogue box, users can set the meter display object's outline. Refer to the picture below for the names of each part of the meter.





Setting	Description
Degree	Set the object's "start degree" and "end degree", the angle range is
	0-360 degrees. The following pictures show several results of different
	settings.
	TTT
	[Start degree] = 200 [End degree] = 70
	[Start degree] – 290, [End degree] – 70
	•
	[Start degree] = 120, [End degree] = 240
	[Start degree] = 40, [End degree] = 140
	$\wedge$
	E.
	[Start degree] = 225. [End degree] = 315
Background	Set the object's background color and profile color.
<b>C</b>	[Full circle]
	When the "Full circle" is selected, the object will display the whole circle,



	otherwise the object will display the defined degree range. See the
	picture below.
	Full circle
	non-full circle
	[Transparent]
	When the "Transparent" is selected, the object will not display the
	background and profile color. See the picture below.
Tick marks	To set the tick mark's number and color.
Pointer	To set Pointer's style, length, width, and color.
Pin point	To set pin point's style, radius, and color

The following pictures show the "Limit" tab and the sign of low and high limit set in the "Limit" tab.



New Meter Display Object
General Outline Limits Shape
Value
Zero : 0 🗢 Span : 100 🗢
Range limits
Enable
Low : Mid : High :
Width: 3
Use user-defined radius
V Dynamic limits
PLC name : Local HMI Setting
Address : LW 🗸 0 16-bit Unsigned
Scale label
✓ Use scale label
Font : Arial
Color : Size : 16
Right of decimal point : 0
OK Cancel Help

Setting	Description
Value	To set object's display range. Meter display object will use the value of
	[Zero] and [Span] and the value of register to calculate the pointer's
	indication position. For example, supposed that [Zero] = 0, [Span] = 100,
	when the value of register is 30 and [Start degree] = 0, [End degree] =
	360, then the degree indicated by pointer is:
	{(30 – [Zero])/([Span] – [Zero])} * ([[End degree] - [Start degree]] =
	${(30-0) / (100-0)} * (360-0) = 108$
	Pointer will indicate the position of 108 degrees. See the picture below.



Range limit	To set the value of low and high limit, the display color, width of the sign of
	Below illustration use above setting to display the range mark.
	30 60
	0 100
	[user-defined radius]
	Range limits       0.00         Low :       Mid :       10         Width :       10       25.00         VUse user-defined radius       80       50.00
	Renge limits Width: 10 Width: 10 Use user-defined radius 30 50.00
[Dynamic	When "Dynamic limits" is not selected, the low limit and high limit are fixed
Limits] /	value, which directly comes from the settings. See the example below, the
uncheck	low limit is 30 and high limit is 60.
	Low limit 30 A High limit 60





[Dynamic	When Dynamic limi	ts is selected, the low lir	nit and high limit are de	ecided
Limits] /	by the register.			
check	Click [Setting] to	Select the [PLC name],	[Device type], [Addre	ess],
	[System tag], [Inde	ex register] for Dynami	ic limits.	
	Users can also set a	address in Limits tab wh	ile adding a new objec	:t.
	Please refer to the f	following dialog.		
		Dynamic limits		
	PLC name : Lo	ocal HMI	Setting	
	Address : LV	W 🔽 0	16-bit Unsigned	
	There following tab	le shows the read addr	ess of low limit and hi	ah limit
	The "Address" mea	ans the register's addre	ss If the register is []	W1001
	the "Address" is 100	).		,
	Data format	Low limit's read	High limit's read	
		address	address	
	16-bit format	Address	Address + 1	
	32-bit format	Address	Address + 2	
Scale label	To select the attribut	e of scale label on mete	er display.	
	MD_0	-Scale label Vse scale label		
	-75 0 25-	Font : Arial	Size : 10	-
		No. of decimal : 0		· ·



## 13.17 Trend Display

## Overview

Trend display object can use the curve to represent the data recorded by data sampling object. The sampling operation is conducted by data sampling objects. The trend display object display the result of sampling. The following picture shows an example of trend display object.



#### Configuration

Click the "Trend Display" icon on the toolbar and the "Trend Display Object's Properties" dialogue box will appear, fill in each items, press the OK button and a new "Trend Display Object" will be created. See the picture below.



The following picture shows the "General" tab in the "Trend Display Object's Properties" dialogue box.



lew Trend Display Object	×
General Trend Channel Shape	_
Description :	
Data Sampling Object index : 0. 💌	
Trend type : Real-time 💌	
Note : if no. of channels is changed, you must reset HMI's data logs !!	
Distance between data samples : 💿 Pixel 💿 Time	
Distance : 100 pixel(s)	
Hold control	
Enable	
PLC name : Local HMI Setting	
Address : LB 🛛 🗸 0	
Watch line	
Enable	
PLC name : Local HMI	
OK Cancel Help	

Setting	Description
[Data	To select data sampling object as the source of data. Refer to the "data
Sampling	sampling" section for related information.
Object index]	
[Trend mode]	To select the mode of data source. There are "Real-time" and "History"
	for selection.
	a. Real-time
	In this mode, it can display the sampling data from the beginning of the
	MT8000 operation to the present time. If previous data are required, you
	must select the "History" mode to read the data from historical record.
	You can use the "Hold control" object to pause the update of trend



display, but it is only pause the update of the trend display, and it will not stop the operation of data sampling object. The picture below shows the "Hold control" setting page. Set the state of the designated register to ON, it will pause the updating of the trend display.

	🗹 Enable			
PLC name :	Local HMI		× (	Setting
Address :	IB	<b>v</b> 0		

#### b. History

In this mode, the data come from the historical record of the designated data sampling object in [Data sampling object index]. Data sampling object will use the sampling data which was sorted in according to dates. The system use "History control" to select the historical records that are created by the same data sampling object. The picture below shows the "History control" setting page.

-History control				
PLC name :	Local HMI		×	Setting
Address :	LW	✓ 0	16-	bit Unsigned

The system sorts the historical records of sampling data by date; the latest file is record 0 (In normal condition it is sampling data today), the second latest file is record 1, and so on.

If the value of designated register in "History control" is n, the trend display object will display data record n.

Here is an example to explain usage of "History control." In the above picture, the designated register is [LW200], if the sampling data available in the files are pressure\_20061120.dtl, pressure\_20061123.dtl, pressure\_20061127.dtl, and pressure\_20061203.dtl and it is 2006/12/3 today. Based on the value of [LW200], the sampling data files selected by the trend display object is shown as follows:



	_			
		Value of [LW200]	The files of the sampling data from	
			the historical record	
		0	pressure_20061203.dtl	
		1	pressure_20061127.dtl	
		2	pressure_20061123.dtl	
		3	pressure_20061120.dtl	
[Distance	[Pixel]	]	· · · · · ·	
between data				
samples] /		Distance between det		
Pixel		Distance between dat	a sourpres. • Prxel • Inne	
			Distance : 20 pixel(s)	
	Select	[Pixel], the [Distan	ce] can be used to set the distance betwee	en
	two sa	impling points. See f	the picture below.	
	40.00			
	12/03	/06		
			$\frown$	
			point 1	
		V point O		
	L			
				1
<b>[X</b> axis time	[Time	1		
rangel / Time	•	•		
0.1				
		X axis tir	me range : O Pixel O Time	
			Distance : 20 second (s)	
	Select	[Time], the [Distanc	e] is used to set the X-axis in unit of time	
	elapse	ed. See the picture b	elow.	





	20 seconds
	Γ 1
	12/03/06
	Otherwise, select Time for X axis time range and go to Trend/Grid for
	enable "Time scale" function. Please refer "Time scale" on the following.
Watch line	Watch line
	✓ Enable
	PLC name : Local HMI 🖌 Setting
	Address : LW 💙 300
	Using the "Watch line" function when user touches the trend display
	object it will display a "watch line" and export the sampling data at the
	position of watch line to the designated word device. You may register a
	position of watch line to the designated word device. You may register a
	numeric display object to display the result. Please refer to the following
	picture



			и	Vatch lin	e	
12/03/06						
				_	A/	
	_/		Å		v v	~
4	LW300					
"Watch line" f The address sampling data address" The correspondin last in sequer	function also registered i a will be exp data forma g address o nce.	o can expor n "watch lin ported to the at of each ch of each char	t samp e" is th e word nannel nnel is	oling data ne start a l devices l may be arrange	a of multi address a starting different d from th	ple channels, and those from "start , the he first to the
For example:						
[LW300]	Ch. 0 : 16	-bit Unsigne	ed	(1 word)	1	
[LW301]	Ch. 1 : 32	-bit Unsigne	ed	(2 words	6)	
[LW303]	Ch. 2 : 32	-bit Unsigne	ed	(2 words	3)	
[LW305]	Ch. 3 : 16	-bit Signed		(1 word)		

The picture below shows the attribute of "trend display".



Trend Display (	Object's Properties
General Trend	Channel Shape Profile
Fr	ame : Background :
0.43	Show scroll controls
- ona	🗹 Enable Color : 🗖 🗖
Ho	oriz. : 👍 🔿 division(s) 🛛 Verti. interval : 👍 📚 second(s)
Time scale -	✓ Enable
For	mat : HH:MM 🔽 Font : Albertus Bold 💙
Ci	olor : Size : 8
- Time/Date	
✓ Time	⊙ HH:MM:SS ○ HH:MM Color :
🗹 Date	⊙ MM/DD/YY ○ DD/MM/YY ○ DD.MM.YY ○ YY/MM/DD
	J
	OK Cancel Help

Setting	Description
[Frame]	The color of frame.
[Background]	The color of background.
[Show scroll	To enable / disable scroll control on the bottom of trend display object.
controls]	
Grid	Set the distance and the color of grid.
	[Horiz.]
	Set the number of horizontal line.
	[Verti. interval]
	a. Pixel
	Point distances :      Pixel     Time
	When select [pixel] to set the display interval (see note on the above







Time / DateThe time of latest sampling data will be marked on the top left corner of<br/>the object. It is used to set the time display format and color.

The picture below shows the attribute of "channel tab".

Trend Display	Object's P	roperties						X
General Trend	Chann	el Shape	Profile	]				
– Data sampling	g object —							
Channel	Display	Descriptio	'n			Data 1	tvide	
0		16-bit Uns	igned			16-bit	Unsigned	
1	<b>~</b>	16-bit BC	D			16-bit	BCD	
2		32-bit Uns	signed			32-bit	Unsigned	
▶ 3	<b>~</b>	32-bit Floa	at			32-bit	Float	
PLC na Addr	me : Loca ess : LW	ynamic limi   HMI		0	Width :	2	Settir 32-bit Flo	ng
		ОК		Canc	el			Help

Setting	Description
[Channel]	Set each sampling line's format and color, and the display data's low
	limit and high limit.
	The max. channel can up to 20 channels.
Limit / uncheck	[Zero]
"Dynamic	
limits"	[Zero] and [Span] are used to set the low limit and high limit of
	sampling data, So if the low limit is 50 and high limit is 100 for one



	sampling line, the so all the samplir	en [Zero] and [S ng data can be	Span] must be set a displayed in the tre	as [50] and [100], end display object.
Limit / check	When Dynamic L	imits is selecte.	d, the low limit and	high limit are
"Dynamic	derived from the	designated wo	rd device. The data	a length of the
limits"	example below,			of object. In the
	Data Format	Low limit	High limit	
	16-bit format	Address	Address + 1	
	32-bit format	Address	Address + 2	
	An extended fund	ction is zoom in	and zoom out fund	ction.

### Example of zoom in/out function

For zoom in / out the trend graph, user has to check the Limit/Dynamic limits as picture below.

	V Dynamic limits			
PLC name :	Local HMI		*	Setting
Address :	LW	✓ 0		32-bit Float

For example, the LW0 and LW1 are to control low limit and high limit, you may change the value of LW1 to zoom in / out.

This following picture is in original size. The range of trend is between  $0\sim30$ . The arrow on the right side are set word (LW1, increment (JOG+) and LW1, decrement (JOG-)) for control the zoom in and zoom out function.



01/03/03	
<b>K K</b>	
LVV1 value	

Decrease LW1's value to exhibit zoom in function as shown below: The value of LW1 decreased to 11.



Increase LW1's value to exhibit zoom out function as shown below: The value of LW1 increased to 41.

	-	/	
4 4		• •	



# 13.18 History Data Display

### Overview

"History Data Display" object displays data stored by data sampling object. It displays history data in numeric format. Please note that the history data display will not refresh automatically, it only retrieve the data from the designated record and display at the time window popup. If the content of the designated record is updated, the history data display will not change accordingly.

No.	Time	Date	Ch.0	Ch.1	Ch.2
3577	21:52	16/09/07	0	0	0
3576	21:52	16/09/07	0	0	0
3575	21:52	16/09/07	0	0	0
3574	21:52	16/09/07	0	0	0
3573	21:52	16/09/07	0	0	0
3572	21:52	16/09/07	0	0	0
3571	21:52	16/09/07	0	0	0
3570	21:52	16/09/07	0	0	0
3569	21:52	16/09/07	0	0	0
3568	21.22	16/00/07	0	0	
•					

#### Configuration

Click the "History Data Display" icon on the toolbar, the "History Data Display" dialogue box show up on the screen. Fill in each items and click OK button, a new object will be created. See the pictures below.





New History Data Display Object 🛛 🔀
General Data Format Title Shape
Data Sampling Object index : 0.
Color : Column interval : 0
Profile color Transparent
Frame : Background :
Text     Font : Arial   Size : 12
Time Time HH:MM Color:
Date DD/MM/YY Color :
Color :
◯ Time ascending
History control
PLC name : Local HMI Setting
Address : LW 200
OK Cancel Help

Setting	Description
[Data	Select the corresponding "Data sampling object" where the history data comes
Sampling	from.
object	
index]	
Grid	Set grid enable or disable.



	No.         Time         Date         Ch.0         Ch.1         Ch.2           3982         22:02         16/09/07         0         0         0
	<u>3981 22:02 16/09/07 0 0 0</u>
	3979 22.02 16/09/07 0 0 0
	3977 22:02 16/09/07 0 0 0
	3976 22:02 16/09/07 0 0 0
	3975 22:02 16/09/07 0 0 0
	3974 22:02 16/09/07 0 0 0
	Set color of grid.
	[Column interval]
	Set space of column.
	No. Time Date Ch.0 Ch.1 Ch.2 ▲ No. Time Date ▲
	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$
	<u>3665</u> 21:57 16/09/07 1 0 0 <u>3665</u> 21:57 16/09/07
	<u>3664</u> 21:57 16/09/07 1 0 0 3663 21:57 16/09/07 1 0 0 3663 21:57 16/09/07
	<u>3662</u> 21:57 16/09/07 1 0 0 <u>3662</u> 21:57 16/09/07 1 0 0
	<u>3661 21:57 16/09/07 1 0 0 3661 21:57 16/09/07</u>
	<u>3659</u> 21:56 16/09/07 0 0 0 0 3659 21:56 16/09/07
Profile	Set color of frame and background. If it is set as transparent, the frame and
	background will be ignored
Time and	Enable or disable the time and date of data sampling and format.
Date	[Time ascending]
	"Time ascending" means to put the earlier data in the top and the latest data in
	the bottom.
	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$
	3 00:24:29 16/09/07 7 6
	4 00:24:30 16/09/07 9 8
	5 00:24:31 16/09/07 6 4
	/         00:24:33         10/09/07         1         4           8         00:24:34         16/00/07         3         6
	9 00:24:35 16/09/07 6 6



	[Time descendi	ng]							
	"Time descendir	ıg" mear	ns to put th	e latest da	ata in t	he top	and	the earlier data	
	in the bottom.								
	Γ	NI-	<b>—</b> :	Dete					
		1707	11me	Date	<u>Ch.0</u>	Cn.1	<u> </u>		
		4/8/	22.24.10	10/09/07	2	2			
		4700	22.24.00	10/09/07	3	2	-		
		4700	22.23.09	10/09/07	3	2	-		
		4702	22.23.00	10/09/07	3	2	-		
		4/03	22.23.07	16/09/07	<u>ゝ</u>	2	-		
		4702	22.23.30	16/09/07	<u> </u>	2	-		
		4701	22.23.33	16/09/07	20	2	-		
		4700	22.23.34	16/09/07	3	2	-		
		4778	22.23.33	16/09/07	3	2			
		•							
History	The history files	are na	med with	date code	e. The	e histo	ry co	ntrol is used to	С
Control	select the design	nated his	story data	files for di	splay.	In cas	e the	value of histor	y
	control is 0, the I	atest file	e is selecte	d. If it is 1	, the s	econd	latest	t file is selected	Ι,
	and so on.								
	Click [Setting]	to Selec	t the [PLC	; name], [	Devic	e type	], [Ad	ldress],	
	[System tag] []	ndex re	- nisterl of	History c	ontrol			•	
						•			
	Users can also s	set addre	ess in Gen	eral tab w	nile ad	iaing a	new	object.	



New History Data Display Object
General Data Format Title Shape
General Data Format Title Shape Channel: Channel 0 ~ Channel 7 Channel 0 [16-bit Unsigned] Left of decimal Pt. : 5 Right of decimal Pt. : 0 Display Center Channel 1 [16-bit BCD] Display Channel 2 [32-bit Unsigned] Display Channel 3 [32-bit Float] character no. : 4 Right of decimal Pt. : 0 Display Channel 3 [32-bit Float] character no. : 4 Right of decimal Pt. : 0 Tisplay Channel 3 [32-bit Float] character no. : 4 Right of decimal Pt. : 0 Tisplay Channel 3 [32-bit Float] character no. : 4 Channel 2 [32-bit Float] Channel 3 [32-bit Float] Character no. : 4 Channel 2 [32-bit Float] Character no. : 4 Channel 2 [32-bit Float] Character no. : 4 Channel 2 [32-bit Float] Character no. : 4 Character
OK Cancel Help

Each history data display object can display up to 20 channels. You can select the channels which you want to watch on the screen.

In the example below, there are four channels in the data sampling object, Ch.0 and Ch.3 are selected for display only. The data format of each channel is decided by the related data sampling objects.



No	Timo	Date	ChO	Ch 3	
140.	TITLE	Date	OII.0	On.0	
5272	22:43:09	16/09/07	4	1	
5271	22:43:08	16/09/07	2	0	
5270	22:33:42	16/09/07	0	0	
5269	22:33:41	16/09/07	0	0	
5268	22:33:40	16/09/07	0	0	
5267	22:33:39	16/09/07	0	0	
5266	22:33:38	16/09/07	0	0	
5265	22:33:37	16/09/07	0	0	
5264	22:33:36	16/09/07	0	0	
5263	22.33.35	16/00/07	0	0	<b>•</b>

When display [String] format in history data display object, users may choose:

# a. Display in [UNICODE] mode

b. Reverse high byte and low byte data then display.

-Channel 1 [String - 5 v	vord (s)] —			
character	no. : 4	-	UNICODE	
📝 Display	Center	•	🔲 Reverse high/low byte	



neral Data Format	Title S	hape		
🔽 Use title				
Title background —				
📃 Transparent		Color :		-
Title name	Title	Label library	Label tag	-
Sequence no.	No.	None		
Time	Time	None		
Date	Date	None		
Channel 0	ch.O	None		
Channel 1	ch.1	None		
Channel 2	ch.2	None		
Channel 3	ch.3	None		=
Channel 4	ch.4	None		
Channel 5	ch.5	None		
Channel 6	ch.6	None		
Channel 7	ch.7	None		
Channel 8	ch.8	None		
Channel 9	ch.9	None		
Channel 10	ch.10	None		
Channel 11	ch.11	None		
Channel 12	ch.12	None		
Channel 13	ch.13	None		
Channel 14	ch.14	None		~
			Setting	

Setting	Description
[Use title]	To enable or disable title.
	No.         Time         Date         Ch.0           5272         22:43:09         16/09/07         4           5271         22:43:08         16/09/07         2
Title	[Transparent]
background	To enable or disable transparent.
	[Background color]



is dialogue window defines the title.
u can use label tag library for title with multi-language. Go to [setting] d select one from label library.
Title : No.   Label tag : no. label   Use label library     OK   Cancel

#### Note:

If you have run the off-line simulation and the sampling data is saved in the record, then you want to change the format of sampling data, be sure to delete previous data record in C:\EB8000\HMI\_memory\datalog to avoid the system misinterpret the old data record.



# 13.19 Data Block Display

## Overview

Data Block is a combination of several word devices with continuous address, for example LW12, LW13, LW14, LW15 and so on. Use Data Block Display object to display multiple data blocks in trend curve, for example, it can display two data blocks LW12~LW15 and RW12~RW15 in trend curve simultaneously. It is very useful to observe and compare the difference of trend curves.



Snapshot of Data Block Display



## Configuration

# [New object]

Click the "Data Block Display" icon *Click*, "Data Block Display's properties" dialogue box appears as follows:

New Data Block Display Object
General Display Area Shape
Description :
No. of channel : 1
Cursor line
Color:
PLC name : Local HMI
Address : LW V
Channel: 0
Control address
PLC name : Local HMI 🗸 Setting
Address : LW V 10
No. of data address : 10 + 1
Data storage start address :
PLC name : Local HMI Setting
Address : LW V 12 16-bit Unsigned
Limit
Min. : 0 Max. : 32767
OK Cancel Help

Setting	Description
[No. of	Set the no of channel for this object. Each channel represents one data
channel]	block. The max. no. of channel is 12.
Cursor Line	Using the "Cursor line" function, when user touches the Data Block
	display object, it will display a cursor line on the data block display object,
	and transfer the position of cursor and the data at the cursor position to



	the designated registers.
	Please refer 19.3 On line operation for further information.
[Channel]	Select each channel and set the attributes.
Control	[PLC name]
address	Select the PLC where the target data block located.
	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of Control address.
	Users can also set address in General tab while adding a new object.
	[Device type]
	Select the device type where the target data block located.
	[Control word address]
	"Control word" is used to control and clear trend curve display.
	0 = No action (default)
	1 = Plot trend curve
	2 = Clear trend curve
	3 = Redraw trend curve
	After executing the operation above, the system will reset the control word to zero.
	[No. of data address]
	"No. of data address" is default as "Control word address +1".
	"No. of data" is to store the number of word device in each data block, i.e.
	the number of data to plot in trend curve. The maximum value is 1024.
	[Data storage start address]
	Click [Setting] to Select the <b>[PLC name]</b> , <b>[Device type]</b> , <b>[Address]</b> , <b>[System tag]</b> , <b>[Index register] of Data storage start address</b> . Users can also set address in General tab while adding a new object.



Г

	.[Offset value storage address]
	If "offset to start address" is enabled, the "Offset value storage address" is default as "Control word address" + 2.
	[Format]
	If you select 16-bit data format, the address of each data will be start address, start address + 1, start address + 2 and so on.
	If you select 32-bit data format, the address of each data will be start address, start address + 2, start address + 4 and so on.
Limit	Set the minimum and maximum limit of trend curve, the trend curve is
	limited by the minimum and maximum limit.



Data Block Display Object's Properties	×
General Display Area Shape Profile	
Data samples : 50 🗢 Samples to scroll : 10 🗢	
Profile color	
Transparent Frame : Background : Background :	
Color : Color :	
Horiz. : 5 🔹 division(s) Verti. : 5 📚 division(s)	
Channel Channel: Pen property Color: Width: 2	
OK Cancel Hel	р

Setting	Description
[Data	Set the data samples, samples to scroll, frame and color of background.
samples]	Data samples : 50 😪 Samples to scroll : 10 📚
	Profile color
	Transparent Frame : Background :



Enable	background		
Disable	e background		





#### On line operation

#### How to show a trend curve

- a. Write the number of data to [No. of data address], i.e. "control word address+1"
- b. Have the content of data block ready for display.
   **NOTE**: data block start from "control word address + 2".
- c. Write "1" to [Control word address], the previous trend curve remains and the new content in data block will be plot on the screen.
- d. The system will write "0" to [Control word address] after the trend curve displayed.



**NOTE** : During the period between c and d, do not change the content of [Control], [No. of Data] and [Data], it might cause error for trend curve plot.

#### How to clear a trend curve

- a. Write "2" to [Control word address], all the trend curves will be cleared.
- b. The system will write "0" to [Control word address] after the trend curve is cleared.




#### How to clear the previous trend curve and display new one

- a. Write the number of data to [No. of data address], i.e. "control word address+1"
- b. Have the content of data block ready for display.
   Note: data block start from "control word address + 2".
- c. Write "3" to [Control word address], the previous trend curves will be cleared and the new content in data block will be plot on the screen.
- d. The system will write "0" to [Control word address] after the trend curve displayed.





#### How to use offset mode

If "offset to start address" is selected, the "Data storage start address" will be calculated from "control word address + [Offset value storage address]". "Offset value storage address" is "control word address +2".

In the following example, the content of "Offset value storage address" is "m", therefore the data block is started from the address "control word address + m".



NOTE

If the control register is 32 bits device, only bit 0-15 will be used as control purpose, bit 16-31 will be ignored. (as illustration below)

	32 1	oit device
3	1 1	6 15 0
+0	0	Control
+1	0	No. of Data
+2	0	Offset

If you do not use "offset to start address", the system will continuously read [Control] and [No. of Data]. At the time [Control] is changed to non-zero, the system will then read the data block. If you use "offset to start address", the system will continuously read [Control], [No. of Data] and [Offset].

It is recommended to use "offset to start address" for data block display

with multiple channels and the same device type. You can register [Control], [No. of Data] and [Offset] in continuous address for each channel. The system will read the control words of all the channels in one read command and it shall speed up the response time.

Please refer to the following picture. The control words of channel 1 is located from address 0, the control words of channel 2 is located from address 3, there are continuous address and the system will read all the control words in one read command.



### How to use watch (Cursor Line) feature

	🛃 Enable	Color :	<b>•</b>
PLC name :	Local HMI		Y
Device type :	LW		~
Address :	1		

You may use the "Watch" function to check the value of any point in trend curve. When operator touches the data block object, it will display a "Cursor line", the system will write the index and value of that data in cursor line to the designated address. The user



shall register NI objects with the designated address. The operator shall be able to observe the numeric value in across with the cursor line.

In the following example, the data block display contains two data blocks. The data format of channel 1 is 16 bit BCD and that of channel 2 is 32 bit unsigned. The cursor is positioned in data index 3 which is corresponding to the fourth data in data block. The system writes "data index" and the content of watched data to the watch address as shown in the following picture.



- **NOTE** 1. [Data Index] is a 16 bit unsigned integer; when the designated register of cursor line is 32 bit device, it will be stored in the bit 0-15.
  - 2. The watch function can only inspect current value in the data block. If there are multiple trend curves of the same channel on the screen, the data of previous trend curves is not exist, only the latest value is available for watch.
  - 3. If the trend curve is cleared, when position the cursor line, the "0" will be displayed as shown below.





4. If there are only three data in Channel 1, when position the cursor in Data 4, the "0" will be displayed as shown below.



#### Limitation:

- 1. The maximum number of channels is 12.
- 2. The system can draw up to 32 trend curves.
- 3. The system can draw up to 1024 points for each channel.



## 13.20 XY Plot

### Overview

XY Plot object displays two dimension data. Each data contains X and Y values and each curve is composed of a stream of XY data. The maximum number of trend curves in a XY plot is 16 channels.

#### Configuration

## [New object]

Click the "XY plot" icon key and "XY Plot Object" dialog box appears.

New XY Plot Object 🛛 🔀
General Display Area Shape
Description :
Direction : Right Vo. of channels : 2
No. of data address : 10 + 1
Channel: 0
Read address
PLC name : Local HMI
Separated address for X and Y data
X data
Address : LW VIII 100 To-bit onsigned
Y data
PLC name : Local HMI
Address : LW 200 16-bit Unsigned
Dynamic limits
X axis
Y axis Low : 0 High : 32767
OK Cancel Help

Setting	Description	
General	a. Direction: There are four selection	ns, right, left, up or down.
	Right:	Left:
	r Right direction r t origin → x	Left direction Y x↑ origin



	Up: Down:
	Up direction origin
	origin → * *Down direction
	h No of channel
	Set the no. of channels of the XX plot. Each channel may conduct the
	draw operation alone
Control	
address	Select the PLC where the control address coming from
2001033	Click [Setting 1 to Select the <b>IPI C name] [Device type] [Address]</b>
	[System tan] [Index register] of Control address
	Users can also set address in General tab while adding a new object
	[Device type]
	Select the device type where the control address coming from.
	[Control address]
	"Control address" is used to control the display of XY curve for each
	channel.
	1= Plot XY curve
	Write "1" to control address, the system will plot the XY curve, the
	previous XY curve if exists would not be clear. The system will reset the
	control address after operation complete.
	2= Clear XY trend curve
	Write "2" to control address, the system will clear all the previous XY
	curves and reset the control address after operation complete.
	3= Refresh XY trend curve
	Write "3" to control address the system will clear the previous XY curve
	and plot the new XY curve and reset the control address after operation
	complete



	[No. of data address]		
	This address store the number of XY data. Each channel can have up to		
	1023 XY data.		
Channel	Setting the channels detail for graph display.		
Read	[PLC name]		
Address	Select the PLC where the control address coming from.		
	Click [Setting] to Select the [PLC name], [Device type], [Address],		
	[System tag], [Index register] of Read address.		
	Users can also set address in General tab while adding a new object.		
	[PLC address]		
	Read address		
	PLC name : Local HMI		
	Separated address for X and Y data		
	PLC name : Local HMI Setting		
	Address : LW V 100 16-bit Unsigned		
	Click [Setting] to Select the [PLC name], [Device type], [Address], ,		
	[Index register], for read address.		
	<ul> <li>The usage of each address as follows, (Dynamic limits is not enabled.)</li> </ul>		
	Enabled.)		
	For example:		
	The Read address is LVV100.		
	X data U reads value from reading address LW100.		
	X data 2 reads value from reading address LW101.		
	X data 2 reads value from reading address LW102.		
	X data 4 reads value from reading address LW104		
	X data 5 reads value from reading address LW105 and so on		
	A data 5 reads value from reading address LVV105 and so on		
	• The usage of each address as follows, (Dynamic limits is enabled.)		
	For example:		
	The Read address is LW100.		
	X low limit reads value from reading address LW100.		
	X high limit reads value from reading address LW101.		
	Y low limit reads value from reading address LW102.		
	Y high limit reads value from reading address LW103.		



X data	0 reads valu	e from read	ing address L	W104.	
Y data	0 reads valu	e from read	ing address I	W105	
X data	1 reads valu	e from read	ing address L	W106	
Y data	1 reads valu	e from read	ing address L	W107	
i data					
lf vou	check "Sepa	rated addre	ess for X and `	Y data". it	allows vou to
differe	nt address fo	r X and Y a	xis respectivel	V.	,
ſ	Read address			,	
	PLC name	Local HMI			*
	¥ 1.1	💌 Separated ad	dress for X and Y dat	a	
	PLC name :	Local HMI		~	Setting
	Address	LW	✓ 100		16-bit Unsigned
	- Y data				
	PLC name	Local HMI		~	Setting
	Address	: LW	✓ 200		16-bit Unsigned
The R	ead address i	is LW100 a	nd LW200.		
X data	l				
X low	limit reads va	lue from rea	ading address	LW100.	
X high	limit reads va	alue from re	ading address	s LW101.	
X data	0 reads valu	e from read	ing address L	W102.	
X data	1 reads valu	e from read	ing address L	W103.	
X data	2 reads valu	e from read	ing address LV	W104.	
X data	3 reads valu	e from read	ing address L	W105 and	so on
Ydata					
Y low	limit reads va	lue from rea	ading address	LW200.	
Y high	limit reads va	alue from re	ading address	s LW201.	
Y data	0 reads valu	e from read	ing address L	W202.	
Y data	1 reads valu	e from read	ing address L	W203.	
Y data	2 reads valu	e from read	ing address L	W204.	
Y data	3 reads valu	e from read	ing address L	W205 and	so on



Limits	The above settings are based on dynamic limits, you can also have
	dynamic limits disable and set the fix high and low limits.
	Limits
	X axis
	Low: 0 High: 32767
	Low: U High: 32767
	The high and low limits is used as scale to calculate the percentage of X
	and Y axis. I.e. X or Y % = ( X or Y reading value – low limit ) /
	( nign limit – Iow limit )
	Pased on your settings, the memory allocation for limit and XX data will
	be as follows
	The following setting is for 16-bit signed data format and dynamic limits
	Read address
	PLC name : Local HMI
	Separated address for X and Y data
	Address : I w 0 16-bit Signed
	X low limit reads value from reading address LW0.(n+0)
	X high limit reads value from reading address LW1. (n+1)
	Y low limit reads value from reading address LW2. (n+2)
	Y high limit reads value from reading address LW3. (n+3)
	X data 0 reads value from reading address LW4. (n+4)
	Y data 0 reads value from reading address LW5. (n+5)
	The following setting is for 32-bit float data format and dynamic limits.



Read address         PLC name :       Local HMI         Separated address for X and Y data         PLC name :       Local HMI         Address :       LW         100       32-bit Float
X low limit reads value from reading address LW100.(n+0) X high limit reads value from reading address LW102. (n+2) Y low limit reads value from reading address LW104. (n+4) Y high limit reads value from reading address LW106. (n+6) X data 0 reads value from reading address LW108. (n+8) Y data 0 reads value from reading address LW110. (n+10)
<b>NOTE</b> There are four different type of selection to designate memory location for high/low limits and XY data. Please refer to the following settings.



Separated a	address for X and Y da	taj		
	imits	Uynamic limits		
X	Y	X	Y	
Data 0	Data 0	Min	Min	
Data 1	Data 1	Niax Data 0	Nax Dota0	
Data 3	Data 2	Data 1	Data 1	
Data S	Datab	Data 1	Data 1	
		·	·	
		:	•	
Separated a	address for X and Y da	ta		
Dynamic H	imits	V Dynamic limits		
X	+ Y	Х +	Y	
X	Data 0	×Mir	י. ר	
Y	Data 0	X Ma	X	
Х	Data 1	Y Mir	1	
Y	Data 1	Y Mai V Dat	X	
X	Data 2	XDai VDat	.au :a0	
Ϋ́́	Data 2	XDat	ao ta1	
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Data 3	YDat	a1	
I		XDat	ta2	
	:	Y Dat	ta2	
		:		
If dynamic limit is ch	ecked vou ma	v change the	high and	low limits to
realize zoom in and	zoom out fu	nction (Pleas	o rofor t	rend display
				liena aispiay
object's dynamic limit.	.)			
				• • • • • •
In the following exam	ple, the dynam	nic limit is sele	ected, whe	ere XL=X low
limit, XH=X high limit,	YL=Y low limit	, YH=Y high li	mit, and <b>&gt;</b>	(Y, XY1, XY2
are three XY data. No	w we change t	the high limit o	of X and Y	respectively
and you may observe	the effect of zo	oom in and zo	om out.	





# [Display Area tab]

New XY Plot Object 🔀
General Display Area Shape Profile color Transparent
Curve Channel: 0 Pen property Color: Width: 1
Maker Point width : 5 Line Point X-axis projection Y-axis projection Reference line
Limit from PLC Low limit : 0 High limit : 100
Reference line 2    Reference line 3    Reference line 4
OK Cancel Help

Setting	Description
Profile	Enable Transparent: It will not display the background color.
color	Disable Transparent: It will display the background color
Curve	Set the attribute of XY curve (color and width) for each channel.



	Channel : O Pen property Color : Width : 1 
Maker	There are four different type of XY plot, i.e. Line, Point, X-axis projection and Y-axis projection, check one of them.
	For Line and Point selection, set appropriate point width (unit in pixels).
	Maker
	Point width : 5
	Chile Cronn C x-axis projection C r-axis projection
	Line & Point:
	Line Dot
	X-axis projection is shown as the following:
	X-Axis Projection Y-Axis Projection
	Remarks: Please refer to the figure below, there is a curve containing 7 points from PO











R	eference line  Limit from PLC	
	PLC name : Local HMI Address : LW	<ul> <li>✓ Setting</li> <li>✓ 0</li> <li>16-bit Unsigned</li> </ul>
	<ul> <li>Reference line 1</li> <li>Reference line 2</li> <li>Reference line 3</li> <li>Reference line 4</li> </ul>	20        40        60        80

Note:

XY Plot can be drawn repeatedly up to 32 times:

1 channel→32 times

2 channels → 16 times

The way to calculate: 32 divided by the number of channels



## 13.21 Alarm Bar and Alarm Display

### Overview

Alarm bar and Alarm display objects are used to display alarm messages. Alarm messages are those events registered in the "Event log" and meet trigger conditions. Alarm bar and Alarm display objects display these alarms in order of priority and triggering time.

Alarm bar object scroll all alarm messages in one line, alarm display object displays alarm messages in multi-line and each line represents one alarm message. The following pictures show that the alarm message are displayed in alarm display and alarm bar objects. Refer to the "Event Log" chapter for related information.

1 (When LW 1 >= 10) 13:21:06 Event 0 (when LW0

Alarm bar object

13/12/06	13:21:38	Event 2 (when LB10 = ON)
13/12/06	13:21:38	Event 3 (when LB11 = ON)
13/12/06	13:21:38	Event 0 (when LW0 == 100)
13/12/06	13:21:38	Event 1 (When LW 1 >= 10)

#### Alarm display object

### Configuration

Click the "Alarm bar" icon on the toolbar, the "Alarm bar" dialogue box appears; similarly, click the "Alarm display" icon on the toolbar, the "Alarm display" dialogue box appears, fill in the setting in the "General tab" and press the OK button, a new object will be created. See the pictures below.





New Alarm Bar Object
Alarm Shape Font
Include categories : 0 thru 0 {see Alarm (Event) Log object}
Scroll speed : Speed 6 🛛 🗸 Acknowledge style : Click 🗸
Calar
Transparent
Frame : Background :
Format
Sort Time ascending
Order & Characters
Display items Display order
Event trigger date     Event trigger time     Event trigger time
Event message
If "Display chars" is 0, it means that the system will display all of characters.
Date : MM/DD/YY V Time : HH:MM:SS V
OK Cancel Help

Setting	Description
Include	Select category of events that belongs to the alarm display or alarm
categories	bar object. (category of an event is set in event log)



	For example, if the category of an alarm bar is set to 2~4, it will	
	display all the alarm messages with "category" equal to 2, 3, or 4.	
	Please refer to "Category" statement in "Event Log" chapter.	
Scroll Speed	Set the scroll speed of alarm bar.	
Color	Set frame and background color of alarm bar.	
Format	a. Sort	
	Set the order to display alarm message.	
	[Time ascending]	
	Put the latest trigger alarm message in the bottom.	
	[Time descending]	
	Put the latest trigger alarm message in the top.	
	b. Order & Characters	
	Users can decide the display item, and how the item display order.	
	c. Date (Event trigger date)	
	Display the date tag with alarm message. There are four formats of	
	date tag	
	uale lay.	
	1. MM/DD/YY / 2. DD/MM/YY / 3. DD.MM.YY / 4. YY/MM/DD	
	d. Time (Event trigger time)	
	Display the time tag with alarm message. There are three formats of	
	time tag.	
	1. HH:MM:SS / 2. HH:MM / 3. DD:HH:MM / 4. HH	

Set font and color of alarm message in the "Font" tab. See the picture below.



New Alarm Bar Object	×
Alarm Shape Font	
Attribute	
Font : Comic Sans MS	~
Color : Size : 12	~
✓ Italic	



## 13.22 Event Display

### Overview

Event display object displays active and finished events. The events are registered in "Event log" object. The active events are the events which are in trigger condition, or have been triggered and unacknowledged.

The event display object displays those active events in the order of trigger time. See the picture below. Event display object can also display the time of the events been triggered, acknowledged and recovered.



### Configuration

Click the "Event Display" icon on the toolbar, the "Event Display" dialogue box appears, set each items in the "General" tab, press OK button and a new "Event Display Object" will be created. See the pictures below.



New Event Display Object 🛛 🔀
General Event Display Shape Font
Description :
Mode : Real-time 🗸
Write address :
Address : 1 W V 0 16-bit Unsigned
Control address
Enable event management
OK Cancel Help

Setting	Description
[Mode]	Select the event source format, there are "Real-time" and "History" for
	selection.
	a. Real-time
	Write address
	This displays the events in the log triggered from HMI starts up till
	present. When the events are acknowledged, the value in [Alarm (Event)
	Log]/ [Message]/ [Write value for Event Display object] will be exported



to th	e [write address] of	[event display] object.		
(W)	rite value for event display —			
	Write value : 200			
b. History Control				
• [E	nable reading multip	le histories] <b>not</b> selected.		
In the the in so to so	his mode it displays event history in daily eparated files with da elect one history reco	event log from history record. The system basis. The event history of each date is s ate tags attached. The "History control" is ord file.	save aved used	
The a wo	picture below shows ord device for "Histor	s the "History control" setting, which desigr y control".	nates	
His	tory control			
De	vice type : LW	×		
	Address : 100	System tag		
		Index register		
	16-bit Unsigned	×		
I he lates the	system selects hist st history record (nor history record one da	tory record by an index. Index 0 refers to mally it is history record today). Index 1 refe ay before the latest, and so on.	o the ers to	
The sele	current value in "H ct corresponding his	istory control" register is used as the indet tory record.	ex to	
Here	e is an example to e rol" register is [LW1	xplain how to use "History control". The "hi 00], supposed that the history records sav	story ed in	
syst	EL 20061120.evt.			
	EL_20061123.evt,			
	EL_20061127.evt EL_20061203.evt			
following table shows the corresponding historical record displayed be event display object according to the value of [LW100].				
	Value of [LW100]	Corresponding Historical Record		
	0	EL_20061203.evt		
	1	EL_20061127.evt		
	2	EL_20061123.evt		





<ul> <li>• [Enable reading multiple histories] selected. Definition: Displays a list of events triggered in multiple days. Illustration: Take LW0 to be the [History Control] [Address] as an example, the range of data to be displayed will be formed by LW0 and LW1 while value in LW0 represents the first history data to start with.</li> <li>Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 \ No.1 \ No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.</li> <li>If the value in LW0 is "3", the first data to be displayed will be data No. 3.</li> <li>If the value in LW0 is "3", the first data to be displayed will be data No. 3.</li> <li>If the value in LW0 is "3", the first data to be displayed will be data No. 4.</li> <li>As for LW1, 2 modes can be selected.</li> <li>a. Number of days</li> <li>If here is local HMI is of LW0 is "1", LW1 is "3", then the start to days before.</li> <li>Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 lisef is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only</li> </ul>	3		EL_20061120.evt		
<ul> <li>[Enable reading multiple histories] selected.</li> <li>Definition: Displays a list of events triggered in multiple days.</li> <li>Illustration: Take LW0 to be the [History Control] [Address] as an example, the range of data to be displayed will be formed by LW0 and LW1 while value in LW0 represents the first history data to start with.</li> <li>Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 \ No.1 \ No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.</li> <li>If the value in LW0 is "3", the first data to be displayed will be data No. 3.</li> <li>If L_2010000 No.2 If KB EVT ###</li> <li>As for LW1, 2 modes can be selected.</li> <li>a. Number of days</li> <li>If istory control Volume is the displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before.</li> <li>Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 tisef is counted). We can see that in this example, since data of 20100609 to set start, the data displayed will only</li> </ul>					
Definition: Displays a list of events triggered in multiple days.         Illustration: Take LW0 to be the [History Control] [Address] as an example, the range of data to be displayed will be formed by LW0 and LW1 while value in LW0 represents the first history data to start with.         Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 \ No.1 \ No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Definition of the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Definition of the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Definition of the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Definition of the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Definition of the value in LW0 is "1" KB EVT ###         As for LW1, 2 modes can be selected.         a. Number of days         Image: Definition of the value of the value in LW1, 2 modes can be displayed will start from number in LW0.         The range of History Data to be displayed will start from number in LW0.         The value in LW1 represents how many days to be included from the start to days before.         Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only </th <th><ul> <li>[Enabl</li> </ul></th> <th>e reading multip</th> <th>ole histories] selecte</th> <th>d.</th> <th></th>	<ul> <li>[Enabl</li> </ul>	e reading multip	ole histories] selecte	d.	
Illustration: Take LW0 to be the [History Control] [Address] as an example, the range of data to be displayed will be formed by LW0 and LW1 while value in LW0 represents the first history data to start with.         Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 · No.1 · No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Displayed will be date they are established, No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Displayed will be date No.3.         Image: Displayed No.1.         Image: Displayed will be date No.3.         Image: Displaye	Definitio	n: Displays a lis	t of events triggered	in multiple days.	
example, the failinge of data to be displayed will be formed by LWO and LW1 while value in LW0 represents the first history data to start with.         Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 · No.1 · No.2). If the value in LWO is "3", the first data to be displayed will be data No. 3.         Image: Displayed will be date they are established, No.4 if the event is the first data to be displayed will be data No. 3.         Image: Displayed will be date they are established, (No.0 · No.1 · No.2). If the value in LWO is "3", the first data to be displayed will be data No. 3.         Image: Displayed will be date they are established, No.4 if the event is the first data to be displayed will be data No.3.         Image: Displayed will be date to be displayed will be event is the first data to be displayed will be data No.3.         Image: Displayed will be date to be displayed will start from number in LWO.         Image: Displayed will start from the start to days before.         Image: As illustrated below, if value of LWO is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	Illustratio	on: Take LW0 to	be the [History Co	ontrol] [Address] as an will be formed by LWO	n ond
Even while value in Evro represents the first fistory data to start with:         Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 · No.1 · No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Control in the contrecontrecontecontrol in the control in the control in t		ile value in LW0	ala lo be displayed v	bistory data to start with	anu th
Example: As illustrated below, for showing it clearer, the history data is numbered according to the date they are established, (No.0 · No.1 · No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Constraint of the date they are established, (No.0 · No.1 · No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Constraint of the date they are established, (No.0 · No.1 · No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Constraint of the date they are established, (No.0 · No.1 · No.2). If the value in LW1, 2 modes can be selected.         a. Number of days         Image: Constraint of the date reading multiple histories         Mode: Image: Constraint of the date reading multiple histories         Mode: Image: Constraint of the date of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only					
numbered according to the date they are established, (No.0 · No.1 ·         No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Description of the start data to be displayed will be data No. 3.         Image: Description of the start data to be displayed will be data No. 3.         Image: Description of the start data to be displayed will be data No. 3.         Image: Description of the start data to be displayed will start from number in LW0.         A for LW1, 2 modes can be selected.         a. Number of days         Image: Description of the start data to be displayed will start from number in LW0.         The range of History Data to be displayed will start from number in LW0.         The value in LW1 represents how many days to be included from the start to days before.         Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	Example	e: As illustrated l	below, for showing i	t clearer, the history da	ita is
No.2). If the value in LW0 is "3", the first data to be displayed will be data No. 3.         Image: Description of the start form 20100609, itself is counted). We can see that in this example, since data of 20100609 itself is counted). We can see that in this example, since data of 20100609 itself is counted). We can see that in this example, since data of 20100609 itself is counted).	numbere	ed according to	the date they are es	tablished, (No.0 丶 No.1	
data No. 3.       IKB EVI ###         Image: Sector of the sec	No.2).	If the value in L	W0 is "3", the first d	lata to be displayed wil	l be
Image: Section 2010/0604       No.3       6 KB EVT ###         Image: Section 2010/0608       No.2       17 KB EVT ###         Image: Section 2010/0610       No.1       4 KB EVT ###         As for LW1, 2 modes can be selected.       12 KB EVT ###         As for LW1, 2 modes can be selected.       .         a. Number of days       Image: Secting	data No.	. 3.			
Image: Setting and the set of the s		SEL_2010060	04 No.4 15 No.3	1 KB EVT 檔案 6 KB EVT 檔案	
As for LW1, 2 modes can be selected.         a. Number of days         Image: History control         Image: Plc name: Local HMI         Image: Plc name: Plc name: Local HMI         Image: Plc name: Plc name: Plc name         Image: Plc name: Plc name         Image: Plc name		EL_2010060	08 No.2	17 KB EVT 檔案	
As for LW1, 2 modes can be selected. a. Number of days History control PLC name : Local HMI Address : LW Delta Local HMI Setting Infe-bit Unsigned Mode : Wumber of days Mode : Wumber of days The range of History Data to be displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only		🕎 EL_2010060 🕎 EL_2010061	10 <b>No.0</b>	4KB EVT 檔案 12KB EVT 檔案	
As for LW1, 2 modes can be selected. a. Number of days History control PLC name : Local HMI Address : LW Description Mode : Number of days The range of History Data to be displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only					
As for LW1, 2 modes can be selected. a. Number of days History control PLC name : Local HMI Address : LW C name : Local HMI Address : LW C maber of days Mode : Number of days The range of History Data to be displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only					
<ul> <li>a. Number of days</li> <li> History control PLC name : Local HMI v Setting Address : LW v o if 6-bit Unsigned Mode : Number of days </li> <li> The range of History Data to be displayed will start from number in LWO. The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LWO is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only</li></ul>	As for L	W1, 2 modes ca	in be selected.		
<ul> <li>a. Number of days</li> <li>History control</li> <li>PLC name : Local HMI</li> <li>Address : LW</li> <li>Enable reading multiple histories</li> <li>Mode : Number of days</li> </ul> The range of History Data to be displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only					
History control         PLC name :       Local HMI         Address :       LW         W       If 6-bit Unsigned         Wode :       Number of days         The range of History Data to be displayed will start from number in LWO.         The value in LW1 represents how many days to be included from the start to days before.         Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	a. Num	ber of days			
Address :       IW       Image: Iteration in the start from number in LWO.         The range of History Data to be displayed will start from number in LWO.         The value in LW1 represents how many days to be included from the start to days before.         Example: As illustrated below, if value of LWO is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	-His	story control	TMI	Setting	
Finally reading multiple histories Mode : Mode : Mode of days The range of History Data to be displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only		Address : I W		16-bit Unsigned	
Mode:       Number of days         The range of History Data to be displayed will start from number in LW0.         The value in LW1 represents how many days to be included from the start to days before.         Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only		Ena	ble reading multiple histories		
<ul> <li>The range of History Data to be displayed will start from number in LW0. The value in LW1 represents how many days to be included from the start to days before.</li> <li>Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only</li> </ul>		Mode : Numbe	er of days	<b>~</b>	
<ul> <li>The range of History Data to be displayed will start from number in LW0.</li> <li>The value in LW1 represents how many days to be included from the start to days before.</li> <li>Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only</li> </ul>					
The value in LW1 represents how many days to be included from the start to days before. Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	The rang	ge of History Da	ta to be displayed w	ill start from number in	LW0.
Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	The valu	ie in LW1 repres	sents how many day	is to be included from t	he
Example: As illustrated below, if value of LW0 is "1", LW1 is "3", then the range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	Start to C	lays before.			
range of data will start form 20100609, and include data of 2 days before (while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	Example	e: As illustrated I	below, if value of LW	√0 is "1", LW1 is "3", the	en the
(while 20100609 itself is counted). We can see that in this example, since data of 20100607 does not exist, the data displayed will only	range of	data will start fo	orm 20100609, and i	nclude data of 2 days t	oefore
since data of 20100607 does not exist, the data displayed will only	(while 20	0100609 itself is	counted). We can s	see that in this example	3,
	since da	ta of 20100607	does not exist, the o	data displayed will only	
include 20100609 and 20100608.	include	20100609 and 2	0100608.		



野EL_20100604       No.4       1 KB       EVT 檔案         野EL_20100605       No.3       6 KB       EVT 檔案         野EL_20100608       No.2       17 KB       EVT 檔案         野EL_20100609       No.1       4 KB       EVT 檔案         野EL_20100609       No.1       4 KB       EVT 檔案         野EL_20100610       No.0       12 KB       EVT 檔案
b. Index of the last history
History control
PLC name : Local HMI 🗸 Setting
Address : LW 🗸 0 16-bit Unsigned
Enable reading multiple histories
Mode : Index of the last history
Range of data to be displayed will take value in LW0 as a start point and
value in LW1 as an end.
Example: if value in LW0 is "1", and LW1 "3", the displayed data will start
from No. 1, and include 3 history data (No.1, No.2, No.3).
「型EL_20100604 No.4 1 KB EVT 檔案 「図EL_20100605」 No.3 G KB EVT 描字
[1] EL_20100608 No.2 17 KB EVT 檔案
[P] EL_20100609 No.1 4 KB EVT 檔案
PEL_20100610 NO.U 12 KB EVT 檔案
The maximum size of data that can be displayed by evotom is 4MD; the
exceeding part will be ignored
The following shows how data will be stored while the data size is too
hig
Example:
a 5 history data each with a size of 0.5MB $\rightarrow$ The size of data to be
displayed will be 5 x 0 5MB
b 5 history data each with a size of $1MB \rightarrow$ The size of data to be
displayed will be 4 x 1MB
c. 5 history data, each with a size of 1.5MB $\rightarrow$ The size of data to be
displayed will be 2 x 1.5MB+1 x 1MB (partial)
Definition:
1. To select confirmed or recovered events to be displayed or
hidden.
2. In [Real- time] mode, select events to be deleted.



Illustration:			
- Control address			
PLC name :	Local HMI	~	Setting
Address :	LW 💌	100	16-bit Unsigned
	🗸 Enable event managemen	ut	
If the address of Hi	story control is set	LW100:	
1. When the value	e in [LW100+0] is "0	)" $\rightarrow$ All events w	vill be displayed.
<ol> <li>When the value hidden.</li> </ol>	e in [LW100+0] is "1	I" →The confirm	ed events will be
<ol> <li>When the value hidden.</li> </ol>	e in [LW100+0] is "2	2" →The recover	ed events will be
4. When the value events will be h	e in [LW100+0] is "3 idden.	3" →The confirm	ed and recovered
5. When the value events under [re	e in [LW100+1] is "1 eal-time] mode.	I" → Users can c	lelete the selected



New Event Display Object 🛛 🔀
General Event Display Shape Font
Include categories : 0 thru 0 {see Alarm (Event) Log object}
Acknowledge style : Click
Max. event no. : 200
Color
Transparent
Frame : Background :
Acknowledge : Return to normal :
Select box :
Format
Sort
Inne ascending 💽 Time descending
Order & Characters
Display items Display chars Display order
Sequence no.     O     Event trigger time     Event processe
Event trigger date 0 Return to normal time
Acknowledge time 0
Return to normal time 0
Event message 0
Té "Dimbre cham" is 0. A means that the matern will firmher all the chamatern
II Display chars is 0, it means that the system will display all of characters.
Date : MM/DD/YY V Time : HH:MM:SS
OK Cancel Help

Setting	Description
Include	Select category of events that belongs to the event display object.
categories	(category of an event is set in event log)
	For example, if the category of an event log display is set to 2~4, it will display all the active event messages with "category" equal to 2, 3, or 4.
	Please refer to "Category" statement in "Event Log" chapter.



	Alarm (Event) Log	Event Display Object's Properties
	Category : AI [7]	General Event Display Shape Font
	Al [7]	Include categories : 0 thru 2
	1 0 1 [2]	
	2 0 3[0]	
	$\begin{array}{c} 3 \\ 4 \\ 1 \\ 6 \\ \hline \end{array}$	Frame : Frame
	5 2 7 [0]	Acknowledge :
	6 2 8 [0] 9 [0] 7 2 10 [0]	Format Select box :
Acknowledge	You may select "Click" or "Double	click" to acknowledge a new event.
style	When a new event comes up, the o	perator can "Click" or "Double click" to
	acknowledge the new event, the sy	stem will change the text color of that
	event and export the "write valu	e" registered with the event to the
	designated register.	
	Take use of this feature, the user ca	an register a popup window and put
	the warning message in the window	ν, then configure an indirect window
	object, when the event is acknowled	dged, the "write value" is written into
	the read address of the indirect win	dow and call up the popup window.
Max. event	The maximum number of events to	be displayed in the event display
no.	object. When the number of events is larger than the maximum, the oldest	
_	event will be removed from the event	nt display object.
Color	Set the color of events in different s	tates.
	a. Acknowledge b. Return to normal	
	c. Select box – The system draw a	highlight box around the latest
	acknowledged event.	
	Acknowledge	
	6 13:12:19 Ex	/ent 1 (When LW 1 >= 10)
	5 13:12:18 EN	vent 2 (when LB10 = ON) vent 3 (when LEV/) = (000)
	3 13:12:15 Ev	vent 2 (when LB10 = ON)
	2 13:12:14 Ex 1 13:12:14 Ex	vent 1 (When LW 1 >= 10) vent 0 (when LW0 == 100)
	Sequence no. R	eturn to normal Select box



Format	trigger date	trigger time	notification time	return to nor	mal time
	0 12/14/06	15:26:21	15:26:31	15:26:36	Event 0 (when LV
	2 12/14/06	15:26:48	9226280		Event 1 (When Li Event 2 (when Li
	a. Sort				
	Set the order to	display aları	n message.		
	[Time ascendir	ng]			
	Put the latest tri	gger alarm n	nessage in the	e bottom.	
	[Time descending]				
	Put the latest trigger alarm message in the top.				
	b. Order & Cha	racters			
	Users can decid	le the displag	y item, and ho	w the item dis	splay order.
	c. Date [Event	trigger date	]		
	Display the date tag.	tag with ala	rm message.	There are fou	ur formats of date
	1. MM/DD/YY /	2. DD/MM/Y	Y / 3. DD.MM.	YY / 4. YY/M	IM/DD
	d. Time [Event	trigger time	2]		
	Display the time tag.	tag with ala	rm message.	There are thr	ee formats of time
	1. HH:MM:SS /	2. HH:MM / 3	3. DD:HH:MM	/ 4. HH	

The font tab sets the font size and italic attribute. The font of event message is set with the event log object.



# 13.23 Data Transfer (Trigger-based)

### Overview

Data Transfer (Trigger-based) object can transfer values from the source registers to the destination registers. The data transfer operation can be activated by pressing the object or setting a trigger bit.

#### Configuration

Click "Data Transfer (Trigger-based) object" icon on the toolbar, "Data Transfer (Trigger-based) object" dialogue box will show up, set each item in the "General" tab, press OK button, a new Trigger Data Transfer object will be created. See the picturea below.



ĸ		
few Dat	a Transfe	ar (1
General	Security	Sha

New Data Transfer (Trigger-based) Object	×
General Security Shape Label	
Derwinting :	
- Source address	
PLC name : Local HMI	
Address : LW V 0	
- Destination address	
PLC name : Local HMI	
Address : LW V 100	
d theiburts	
No. of word : 1	
Mode : External trigger 🖌 Trigger mode : ON->OFF	
Trigger address External trigger	
PLC name : Local HMI	
Address : LB 🗸 0	
OK Cancel Help	

Setting	Description
Source	Set source address of data transfer.
address	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of Source address.
	Users can also set address in General tab while adding a new object
Destination	Set the destination address of data transfer.
address	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of Destination address.
	Users can also set address in General tab while adding a new object



Attribute	[No. of words]		
	The number of words to be transferred from source to destination.		
	Set the trigger mode of data transfer.		
	[Mode]		
	a. Touch trigger		
	Press the object to activate data transfer operation.		
	b. External trigger		
	Register a bit device to trigger the data transfer operation.		
	$\begin{bmatrix} ON \rightarrow OFF \end{bmatrix}$		
Bit device change from O <b>[OFF</b> $\rightarrow$ <b>ON]</b> Bit device change from O	Bit device change from ON to OFF to activate data transfer operation.		
	$OFF \rightarrow ON1$		
	Bit device change from OFF to ON to activate data transfer operation		
	$[ON \leftarrow \rightarrow OFF]$		
	Bit device change state to activate data transfer operation.		
	< Attribute		
	No. of word : 1		
	Mode : External trigger 💙 Trigger mode : ON->OFF 💙		
	- Trigger address		
	PLC name : Local HMI		
	Address : LB V 0		



# 13.24 Backup

#### Overview

The backup function can store the recipe data (RW, RW\_A), event log and sampling data to USB device or Remote backup server. The [LB-9039] represents the backup status, when backup operation is in progress, the status of [LB-9039] is ON.

#### Configuration

Click "Backup Object" icon on the toolbar, the "Backup Object" dialogue box will show up. See the pictures below.




lew Backup Object	2
General Security Shape Label	
Description :	
Source	5
○RW ○RW_A ⊙ Historical event log ○ Historical data sampling	
-Backup position	5
⊙ USB 1 ○ USB 2 ○ SD card	
○ Remote printer/backup server	
Note : Use LW-9032~9039 to change the backup folder name.	
Note : Use [Remote printer/backup server] to store data to a remote PC. Enable the server in [System Parameter][Printer/Backup Server] settings.	
Save format	ĥ
Format : MT8000 Event Log File (*.evt)	
Range	ĥ
Start : 💿 Today 🛛 🔿 Yesterday	
Within : All	
Trigger	ĥ
Mode : External trigger (bit)	
Condition : OFF->ON	
Trigger address	ĥ
PLC name : Local HMI	
Address : LB 🗸	
	J

Setting	Description
Source	[RW], [RW_A], [Historical event log], [Historical data sampling]
	Select one from the above for the source. There may be several data
	sampling objects registered in the project. If you select [Historical
	data log], use "Data Sampling object index:" to select the right one
	as shown below.



	New Backup Object		
	General Security Shape Label		
	Description : Source RW ORW_A OHistorical event log OHistorical data log Data Sampling object index : 1. Backup position 1.		
Backup Position	Select the destination where the source files will be copied to.		
FOSICION	a. USB1 or USB2 or SD card		
	The external device connected to HMI.		
	b. Remote printer/backup server		
	To select this, users have to enable <b>MI remote printer/backup</b>		
	Menu ⇔ Edit ⇔ Svstem Parameters ⇔ Printer/Backup Server		
0 former at			
Save format	User can select the desired format to back up the file.		
	a. MT8000 Event Log File (*.evt) / MT8000 Data Log File (*.dtl)		
	h Comma Sanaratad Valuae (* cev)		
	<ul> <li>Event Log saved as csv file</li> </ul>		
	Save format		
	Format : MT8000 Event Log File (*.evt)		
	MT8000 Event Log File (*.evt) Comma Separated Values (*.csv)		
	Data Log saved as csv file           Save format           Format :           MT8000 Data Log File (*.dtl)           MT8000 Data Log File (*.dtl)           Comma Separated Values (*.csv)		
	When back up event log in csv format, users can find data fields in EXCEL as below.		



	<b>X</b> )	licrosoft Ex	cel - EL_2	0101029.xl	s		
	:2)	檔案①	編輯(王) 核	観(♡) 插.	入(1) 格式(0	() 工具	
	In		AIA		😤 + 🎯 🗆	7 - 1 🔍	
	1	<b>9</b> 3 <b>9</b> 3 (2)		SMI	8 6 6	やり回鹿嶺	
		D20		<ul> <li>€</li> </ul>			
		A	В	С	D	E	
	1	[Creation ti	me]				
	2	Fri Oct 29	10:59:28 20	010			
	3	Data fields	cotegory	timo	moccodo		
	4	Detel	category	ume	message		
	6		0	11.19.42	"Emergenc	w"	
	7	Ő	5	11:19:43	"5"	9	
	8	2	0	11:19:46	"LOW"		
	9	2	5	11:19:49	"5"		
	10	1	0	11:19:52	"Word"		
	11	2	0	11:19:52	"Word"		
	12						
	13						
•	14						
C	) ->	event is t	riggered				
			00				
1	->	event is a	acknowle	dged			
2	<u>2</u> ->	event ret	urns to n	ormal			
Range [	Sta	rt] from [	Today] o	r <b>[Yeste</b> i	rday]		
[	Wit	hin]					
S	Sele	ct the rai	nae of tir	ne perio	d. for exa	mple. S	Select [Yesterdav] in
	Stor	tl and co	loot "2 d		moone to	covo th	o filos vostordav and
	Stai	ij, anu se		ay(s) . It		Save II	e liles yesteruay and
t	he c	lay before	e yesterd	ay. Sele	ct "All" to	save all	the files available in
t	he s	system.					
			-De				
				arra -		~	
				ារពា: 🧕	) Today	🔘 Yesterd	ay
				Within :	.11	~	
				4	(]] 		
			-At	tribute	day(s) day(s)		
				Mod 7	day(s) day(s)	~	
				i	4 day(s)		, 
			- T	damer address	1 day(s)		



Attribute	There are two ways to activate Backup function.				
	a. Touch trigger				
	Touch the object to activate backup operation.				
	b. External trigger (bit)				
	Register a bit device to trigger the backup operation.				
	$[ON \rightarrow OFF]$				
	Bit device change from ON to OFF to activate backup operation.				
	$[OFF \rightarrow ON]$				
	Bit device change from OFF to ON to activate backup operation.				
	$[ON \leftarrow \rightarrow OFF]$				
	Bit device change state to activate backup operation.				
	Irigger address When use "External trigger" assign an appropriate bit device as				
	shown below.				
	Trigger				
	Mode : External trigger (bit)				
	Condition : OFF->ON				
	Trigger address				
	PLC name : Local HMI				
	Address : LB 0				
	c. External trigger (word)				
	When selecting [External trigger (word)], users can specify the				
	number of days to backup data using [Trigger address].				



Trigger
Mode : External trigger (word)
Address : LW V
[Trigger address] users (suppose the surrent Trigger Address is set
[Ingger address] usage (suppose the current ingger Address is set
to LW-0) :
I.W.O: When the value of this address changes from 0 to 1, trigger
backup.
LW-1: The data in this address is for specifying the start date of
backup.
LW-2: The data in this address is for specifying the number of days
for bookup
The Syntax is shown below:





# 13.25 Media Player

For the first time using Media Player object, it's necessary to download the project to the HMI *via Ethernet*. EasyBuilder8000 will install Media Player drivers during the download.

# Overview

The Media Player function is not only used to play video files but also to provide uses of additional controls such as seeking, zooming, volume adjusting and so on. With the Media Player, users can provide operation and maintenance instructions or standard procedures on video, which can help to create an environment that enables any on-site operators to perform tasks efficiently from clear, comprehensible instructions. (Note: The Media Player function is only available on the MT8000X Series models.)

## Configuration

Click "Media Player object" icon on the toolbar, "Media Player object" dialogue box show up, set each item in the "General" tab, press OK button, a new Media Player object will be created. See the pictures below. (Note: The instruction of this section is an example to play a video file located in the "/example" directory.)



New Media Player Object	
General Preview	
Description :	
Control address	
Enable	
PLC name : Local HMI	Setting
Address : LW	• 0
Command : 0	Status : 0 + 3
Parameter 1 : 0 + 1	File index : 0 + 4
Parameter 2 : 0 + 2	Start time : 0 + 5
	End time : 0 + 6
Update video playing	; time
Ext. device	Folder name : example
Attribute	Background :
ОК	Cancel Help

#### General tab :

Click [Setting...] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of Control address.

Users can also set address in General tab while adding a new object.

a. In [Control address], select [Enable] and register a word device to control the operation of media player object (example : LW0)



- Control address	
PLC name : Local HMI	Setting
Address : LW	• 0
Command : 0	Status : 0 + 3
Parameter 1 : 0 + 1	File index : 0 + 4
Parameter 2 : 0 + 2	Start time : 0 + 5
	End time : 0 + 6

b. In [Control address], unselect the [Update video playing time]

Update video	playing time

c. In [Ext. device], select [USB1] and input "example" as [Folder name].

CI.	Ext. device —			
	🔿 SD	💿 USB1	🔿 USB2	Folder name : example

d. In [Attribute], unselect [Auto. repeat] and choose black as the background color.

Attribute	
🔲 Auto. repeat	Background :

#### Preview tab :

Users can examine whether the MT8000 supports the video format via preview function.





- a. Click [Load...] and select the file to be examined. (Users should put the file in the /example directory of an USB disk)
- b. If the media player starts playing the video, it means the MT8000 supports this video format. Use [<<] and [>>] to navigate video by 1 minute each time.
- c. To play another video, click [Stop] to close the video file and repeat from step a.

#### Prepare the video file:

- a. Remove all external devices (SD/USB disk) connected to the MT8000.
- b. Plug the USB disk, which has the video file in it, into the MT8000.

#### Note

The first step is there for ensuring the USB disk (in step b) will be recognized as USB1.

#### Start/Stop playing video

#### 1. Start playing video

- a. Set [Parameter 1] to 0.
- b. Set [Command] to 1, the system will open the video file and start playing.
- c. After the system start operation, it will reset the [Command] to "0".



# Note

During the period between step b and c, don't change the content of [Command], [Parameter 1], and [Parameter 2], it may cause unpredictable result.

# 2. Stop playing video

- a. Set [Command] to 5, the system will stop playing and close the video file.
- b. After the system complete step a, it will reset the [Command] to "0".

Note

During the period between step a and b, don't change the content of [Command], [Parameter 1], and [Parameter 2], it may cause unpredictable result.



# Media player setting guide

# General tab :

w Media Player Obj	ect
Jeneral Preview	
Description :	
Control address	Early
· · · ·	Enable
PLC name : Lo	cəl HMI 👻
Device type : LV	φ 💌
Address : O	System tag
Address format : DD	DDD [range : 0 ~ 10500]
	Index register
Command : 0	Status : 0 + 3
Parameter 1 : 0 +	1 File index : 0 + 4
Parameter 2 : 0 +	2 Start time : 0 + 5
	End time : 0 + 6
	Update video playing time
Ext. device	
SD OS	B1 OUSB2 Folder name : example
Attribute	
🔲 Auto. repeat	Background :

Setting		Description
	Enable control	Enable
Control address	address	<ul><li>a. You can use "Control address" to control the operation of media player</li><li>b. Register a device address for "Control address".</li></ul>
		<ul> <li>Disable</li> <li>There is no manual control of video play operation. The system will start to play the first</li> </ul>



			video at designated folder when the window is			
			popup.			
	Comman	nd	Users set this address to control the operation of			
			media player.			
			<ul><li>Command (control address + 0)</li></ul>			
	Paramet	er 1	Parameter 1 for control operation.			
			<ul><li>Parameter 1 (control address + 1)</li></ul>			
	Parameter 2 Status		Parameter 2 for control operation			
			<ul><li>Parameter 2 (control address + 2)</li></ul>			
			The system will turn bits ON when state changes or			
			malfunctions.			
			Status (control address + 3)			
	File index		The system will write file index when starting to play			
			a video.			
			<ul><li>File index (control address + 4)</li></ul>			
	Start time		The system will write video start time when starting			
			to play a video. (unit = sec) (Always 0)			
	End time Update		Start time (control address + 5)			
			The system will write video end time when starting to			
			play a video. (unit = sec)			
			End time (control address + 6)			
			• Enable			
		video	The system will write video elapsed time into			
		playing	[playing time] register in every [update period]			
	Video	time	seconds.			
	playing	Update	Update period of [playing time], range between 1 to			
	time	period	60 sec.			
		Playing	Update the video elapsed time periodically. (unit =			
		time	sec)			
			Playing time (control address + 7)			
	SD		Play video files in SD card.			
	USB1		Play video files in USB1.			
Video	Video USB2		Play video files in USB2.			
file store	Folder n	ame	The name of the folder storing video files. Users			
location			must put video files in a folder (e.g. "/example")			
			instead of root directory.			
			Note			



		1. [Folder name] couldn't be empty.
		<ol> <li>[Folder name] couldn't include Λ:*?"&lt;&gt; .</li> </ol>
		3. A folder name must be composed entirely of
		ASCII characters.
	Auto. repeat	When finish playing a video file, the system will
Attributo		automatically play next video.
Allibule		e.g. [video 1] $\Rightarrow$ [video 2] $\Rightarrow$ $\Rightarrow$ [video n] $\Rightarrow$ [video 1]
	Background	Select the background color of the object.

Normally the format of the above registers is 16-unsigned integer. If a 32-bit word device is chosen as the control address, only 0-15 bits are effective. Users should zero the 16-31 bits.

# Control command :

a. Play index file

[Command] = 1 [Parameter 1] = file index [Parameter 2] = ignore (set 0)

- Note 1. The files are sorted with file name in ascending order, the "file index=0" is for to the first file, and son on.
  - 2. If it is unable to scan file, it will set [status] bit 8 to ON.
  - 3. If check [Auto. repeat], it will automatically play the next file after finish.

# b. Play previous file

[Command] = 2 [Parameter 1] = ignore (set 0) [Parameter 2] = ignore (set 0)

# Note1. If the [file index] is previously 0, it will re-play the same video from the start.2. If it is unable to search the right file, it will set [status] bit 8 to ON.

3. If check [Auto. repeat], it will automatically play the next file after finish.

# c. Play next file

[Command] = 3 [Parameter 1] = ignore (set 0)



[parameter 2] = ignore (set 0)

Note 1. If there is no next video file, it will play the first (index 0) file.

- 2. If it is unable to search the right file, it will set [status] bit 8 to ON.
  - 3. If check [Auto. repeat], it will automatically play the next file after finish.

#### d. Pause / Play Switch

[Command] = 4 [Parameter 1] = ignore (set 0) [Parameter 2] = ignore (set 0)

#### e. Stop playing and close file

[Command] = 5 [Parameter 1] = ignore (set 0) [Parameter 2] = ignore (set 0)

#### f. Start playing at designated target location

[Command] = 6 [Parameter 1] = target location (sec) [Parameter 2] = ignore (set 0)

#### g. Forward

[Command] = 7 [Parameter 1] = target location (sec) [Parameter 2] = ignore (set 0)

- Note 1. Increase playing time by [Parameter 1] seconds. If the system is previously playing video, it continues to play after the operation. If previously paused, it keeps paused.
  - 2. If the playing time is over end time, the system play video from last second.

#### h. Backward

[Command] = 8

Note Parameter 1 (target location) should less than end time. If it is over end time, the system play video from last second.



[Parameter 1] = target location (sec) [Parameter 2] = ignore (set 0)

Note 1. Decrease playing time by [Parameter 1] seconds. If the system is previously playing video, it continues to play after the operation. If previously paused, it keeps paused.

2. If the playing time is less than start time, the system play video from the beginning.

# i. Adjust volume

[Command] = 9[Parameter 1] = volume ( $0 \sim 128$ ) [Parameter 2] = ignore (set 0)

Note Default volume is 128.

## j. Set video display size

[Parameter 1] = display size  $(0 \sim 16)$ [Parameter 2] = ignore (set 0)

#### [Command] = 10

#### Note 1. [0] : Fit video image to object size. 2. [1 ~ 16] : Magnification from 25% ~ 400%. Set 1 for 25%, 2 for 50%, 3 for 75% and so on.

#### k. Status (control address + 3)



	15	09	08	(	02 01	00	bit
	Reserved (all 0)	0	0		0	0	
Bit 00: open fi	le bit (0: file closed;	1: 1	file ope	ened)			
Bit 01: play file	01: play file bit (0: not playing video;1: playing video)						
Bit 08: comma	and error bit (0: comr	mar	d acce	epted;			
	1: inco	rrec	t comr	nand or parame	ters)		
Bit 09: file erro	or bit (0: file format a	cce	pted;				
	1: unknown fil	le fo	ormat c	or reading file en	ror)		

When playing a video, the system will turn ON [open file bit] and [play file bit]. If the file is unable to be scanned or the command is incorrect, the [command error bit] will be set ON  $(0\rightarrow 1)$ .

Note1. If file format is unsupported or disk I/O error happens during playing (e.g.<br/>user unplugs the USB disk), the [file error bit] will be set ON  $(0\rightarrow 1)$ .

2. Refer to the following figure, the value of [status] at each state would be: "Stop" [status] = 0
"Deves" [status] 4 ([seen file bit])

"Pause" [status] = 1 ([open file bit])

"Playing" [status] = 3 ([open file bit] + [play file bit])



★ Users should only set values to [Command], [Parameter 1] and [Parameter 2], and regard the other registers as read-only.

# Restrictions



- The system can only play one video file each time.
- If [Auto. repeat] is unselected, the system will stop playing video and close the file after complete a video play operation.
- If [control address] is unselected, the system will find the first file in the designated directory and start playing it.



# 13.26 Data Transfer (Time-based)

# Overview

Data transfer (Time-based) object is the same as Data transfer (Trigger-based) object, it also transfers the data from source to destination register. The difference is the way to activate data transfer operation. The Data transfer (time-based) object conducts data transfer operation based on time schedule, it can also transfer data in the unit of bits.

# Configuration

Click "Data Transfer (Time-based) Object" icon on the toolbar, the summary of data transfer objects is shown as follows:

Data Transfer (Time-based)
Data Transfer (Time-based) Object
<ol> <li>[Local HMI:LB200] -&gt; [Local HMI:LB210], Mode : Bit, Time interval=2.0 second(s), transfer length=10 bit(s)</li> <li>[Local HMI:LW250] -&gt; [Local HMI:LW260], Mode : Word, Time interval=2.0 second(s), transfer length=1 word(s)</li> </ol>
New Delete Settings Exit

Press the "New..." button in the above dialogue box, the Data Transfer (Time-based) Object dialogue box appear as shown in the picture below, set item and press OK button, the object will be created.



Data Transfer (Time-based) Object	
Description :	
Attribute	
Address type : Bit Interval : 3.0. second(s)	~
Active only when designated window opened	
- Source address	
PLC name : Local HMT Setti	na
Address : IB	
Destination address	
PLC name : Local HMI 🛛 🗸 Setti	ng
Address : LB 🗸 0	
OK Cano	:el

Setting	Description		
Attribute	[Address type]		
	Select the bit or word device.		
	[No. of words] or [No. of bits]		
	When select "Word type", the unit of data transfer is word, set the number		
	of data to transfer. See the picture below.		
	Attribute		
	Address type : Word V Interval : 3.0 second(s)		
	No. of words : 4		
	When select "Bit type" the unit of data transfer is hit set the number of		
	data to transfer. See the nicture below.		
	Attribute		
	Address type : Bit Interval : 3.0 second(s)		
	No. of bits : 15		



	[Interval]
	Select the wait interval for each data transfer, for example, select 3
	seconds, the system will conduct data transfer operation every 3 seconds.
	Note 1. Specifying a small interval or a big number of data to transfer may cause an overall performance decrease due to the time consuming in transferring data. Therefore, users should always try to choose a longer interval and a smaller amount of data to transfer.
	2. When a short interval is inevitable, be aware of the interval must be longer than the data transfer operation. For example, if the data transfer operation take 2 seconds, you must set the interval longer than 2 seconds.
Source	Set source address.
address	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of source address.
	Users can also set address in General tab while adding a new object.
Destination	Set destination address.
address	Click [Setting] to Select the [PLC name], [Device type], [Address],
	[System tag], [Index register] of destination address.
	Users can also set address in General tab while adding a new object.

After completing all settings and pressing the "OK" button, a new Data Transfer (Time-based) Object is created. The summary displays all the registered data transfer objects with brief information as shown below.

#### Data Transfer (Time-based) Object

[Local HMI:LB200] -> [Local HMI:LB210], Mode : Bit, Time interval=2.0 second(s), transfer length=10 bit(s
 [Local HMI:LW250] -> [Local HMI:LW260], Mode : Word, Time interval=2.0 second(s), transfer length=1 \
 [Local HMI:LB30] -> [Local HMI:LB60], Mode : Bit, Time interval=3.0 second(s), transfer length=15 bit(s)



# 13.27 PLC Control

# Overview

The PLC control object activates a specific operation when the corresponding control device is triggered.

# Configuration

Click the "PLC Control" icon and the "PLC Control Object" summary appears as shown below.



LW150 LW151	Change window Write data to PLC(current base window) Conserved DLC constraint	
LW4UU LB300	General PLC control Back light control(write back)	

Press the "New..." button and the "PLC Control" dialogue box appears. Set all the attributes of PLC control and press OK button, a new PLC control object will be created.



PLC Control	
Description :	
PLC name :	Local HMI
Attribute	
Type of control :	Change window
Active only w	Change window Write data to PLC(current base window)
Turn on back	General PLC control Back light control (write back)
Trigger address	Sound control
PLC name :	Local HMI Setting
Address :	LW 0 16-bit Unsigned
	OK Cancel

Setting	Description			
Attribute &	[Type of control]			
Trigger	To set the type of control. Click the select button and you can drag down a			
address	list of all available PLC control functions			
	Attribute         Type of control :       Change window         Change window         Write data to PLC(current base window)         General PLC control         Back light control(write back)         Back light control         Sound control         Device type :			
	a. "Change window"			
	This is used to change base window. When the value of [Trigger address]			
	is written in a valid window number, the system will close the current			
	window and open the window designated by the [Trigger address]. The			



new window number will be written to the [Trigger address + 1]. PLC Control Description : PLC name : Local HMI ¥ Attribute Type of control : Change window Active only when designated window opened 10. WINDOW\_010 V Turn on back light Clear data after window changed Trigger address PLC name : Local HMI Setting... ✓ 0 16-bit Unsigned Address : LW As an example of the above configuration. When writing a valid window number – 11 into LW0, the system will close the current window and open window 11, then write 11 into LW1 (LW0+1) If you use 32-bit device as trigger address, and the device type of the trigger address is in word basis, then the system will write the window number into [Trigger address +2]. Below is the list of write address for each different type of data format. Write address **Data Format Trigger address** Address + 1 16-bit BCD Address Address + 2 32-bit BCD Address Address Address + 1 16-bit Unsigned 16-bit Signed Address Address + 1 32-bit Unsigned Address Address + 2 Address + 2 Address 32-bit Signed Note: If [LB-9017] = ON, the write back operation will not be executed. If "Clear data after window changed" is selected, the [Trigger address] will be reset to 0 after new window is open. b. "Write data to PLC (current base window)" When the system changes the base window, the new window number will



be written into the [Trigger address].

#### c. "General PLC Control"

This function performs data transfer between PLC and HMI when users set appropriate value in [Trigger address].

Control code	Operation for data transfer	
[Trigger address]		
1	PLC register → HMI RW	
2	PLC register → HMI LW	
3	HMI RW → PLC register	
4	HMI LW $\rightarrow$ PLC register	

With this function the system uses four continuous word devices, please refer to the following explanation.

Address	Purpose	Description
[Trigger	Control code	The valid control code is listed
address]		in the above table. When a new
		control code is written into the
		register, the system will conduct
		the data transfer function.
[Trigger	Number of words to	
address+1]	transfer	
[Trigger	Offset to the start	If the value is "n", the start
address+2]	address of PLC	address of PLC register is
	register	"Trigger address + 4 + n".
[Trigger	The start address of	
address+3]	LW or RW	
address+3]	LW or RW	

As an example, to transfer PLC registers [DM100, 101 ... 105] to HMI [RW10, 11 ... 15], follow the steps below:

- 1. Set Trigger address to DM10.
- 2. Set [DM11] = 6 (no. of words to transfer)
- 3. Set [DM12] = 86 (DM10+4+86= DM100)
- 4. Set [DM13] = 10 (RW10)
- 5. Set [DM10] = 1, The system will execute the data transfer operation.
- d. "Back light control (write back)"



I his c	
eveta	operation is the same as "Back light control (write back)" except
e "S	Sound control"
_Se	pund
	Sound Library Beep
	Play Trigger mode : OFF->ON
Activ	ate the [Trigger address], the system will play the sound
/ 1011/1	
Selec	t a sound from sound library for the PLC Control.
You r	nay configure three different ways to activate the [Trigger addre
(1) 61	at a change from $OEE$ to $ON$ ( $OEE > ON$ )
(1) 31	
(2) St	ate change from ON to OFF (ON->OFF)
(3) St	ate change (either from ON->OFF or OFF->ON)
f. "E	Execute macro program"
	Trigger address
	PLC name : Local HMI
	Address : LB
	Trigger mode : OFF->ON
Activa	ate the [Trigger address], the system will execute the Macro
Activa	ate the [Trigger address], the system will execute the Macro.







Note
1. The system performs a <i>background printing process</i> when the
printed window is not the current base window.
2. For a window designed to be printed at background, users should
put neither direct window nor indirect window in it.



# 13.28 Schedule

Schedule object is used to turn on/off a bit or write a value to a word device at designated time. The time schedule setting is very flexible, it can be on daily basis or weekly basis. For more advance application you can use a table (a block of word devices) to set start and terminate time, then update the table at any scheduled time.

# Configuration

Click the "Schedule" icon on the toolbar and the "Scheduler list" dialogue box will appear, press the "New", the schedule object dialogue box will appear as shown below:

6	1
R.	E
	1

Scheduler	
New	Exat



Scheduler	
General Time Set	Prohibit
Description :	Scheduler 1
A stimums Is	Power-ON start/end action
-Action mode	⊙ Bit ON ○ Bit OFF ○ Word write
Action address	
PLC name :	Local HMI Setting
Address :	LB 🔽 100
	OK Cancel Help

# Example 1:

The motor is scheduled to be power ON at 8:00 and power off at 17:00, Monday to Friday. Here we use LB100 to control the motor. Follow the steps to set up the schedule object.





Click [New...], to add a new object,

# [General tab]

[Power-ON start/end action]

Detail message please refer to below Scheduler settings guide.

Power-ON start/end action

1. Check [Bit ON] in [Action mode],



2. Set LB100 in [Action address]

Action address —				
PLC name :	Local HMI		~	Setting
Address :	LB	✓ 100		



# [Time Set tab]

3. Select [Time Set] tab, check [Constant]

General	Time Set	Prohibit
⊙ Con	istant	🔿 Address

4. Unselect [Setting on individual day]. In [Start], adjust time as 8:00:00 and select Monday to Friday.

📃 Setting	y on individu	al day					
_Start							
8	\$	\$	0	\$			
🗌 Sun	🗹 Mon	🗹 Tue	🔽 Wen	🗹 Thu	🗹 Fri	🗌 Sat	

5. In [End], select [Enable termination action] and adjust time as 17:00:00.

End Enable termin	nation action		
17 🗘	0	0	

6. Click [OK], a new schedule object is created and display on the schedule list.



Scheduler	
1: Scheduler 1 : [Bit ON]->[Local HMI:LB100], Time:[MON TUE WED THU FRI]	
New Delete Settings	Exit

## Example 2:

Set temperature at 90F at 8:00 and set it back to 30F (standby mode) at 17:00, Monday to Friday.





Click [New...], to add a new schedule object. Follow the steps to set up the schedule object. The [LW100] is used to store set value of temperature.

Scheduler				×
General Time Set	Prohibit			
Description	Scheduler 1			
	Power-ON start	/end action		
Action mode	O Bit ON	O Bit OFF	💽 Word write	
Action address —				_
PLC name :	Local HMI		Setting	
Address	LW	✓ 100	16-bit Unsigned	t
- Word write value :	ættings	<b>•</b> • • • •		
Start value	• Constant	Address		
	ОК	Cancel	П	lp

# [General tab]

1. [Power-ON start/end action]

Power-ON start/end action

2. Check [Word write] in [Action mode],

-Action mode -				
	🔘 Bit ON	🔘 Bit OFF	💿 Word write	

3. Set LW100 in [Action address]



Action address —			
PLC name :	Local HMI		Setting
Address :	LW	✓ 100	16-bit Unsigned

4. Check [Constant] and set [Write start value] to 90 in [Word write value settings],

Mond	unite unite o	attin an		
_ W010	write value s	ermißs		
۲	) Constant	🔘 Address		
	Write	start value : 90		

# [Time Set tab]

5. Select [Time Set] tab, check [Constant]

General	Time Set	Prohibit
⊙ Con	istant	🔿 Address

6. Unselect [Setting on individual day]. In [Start], adjust time as 8:00:00 and select Monday to Friday.

📃 Setting	on individus	lday					
Start —							
8	<b>\$</b>	\$	0	\$			
🗌 Sun	🗹 Mon	🗹 Tue	🔽 Wen	🗹 Thu	🗸 Fri	📃 Sat	

7. In [End], select [Enable termination action] and adjust time as 17:00:00.

- End Enable termin	ation action		
17 🜲	0 🛟	0	



8. Select [General] tab, set [Write start value] to 90 and [Write end value] to 30.

9. Click [OK], the settings appear in the Scheduler list.

# Schedule settings guide

General tab

scheduler				δ
General Time Set	Prohibit			
Description	Scheduler 1			
Action mode	Power-ON start/	end action	💽 Word write	
Action address —				
PLC name :	Local HMI		Setting	
Address :	LW			neu
Word write value s	ettings • Constant 90	◯ Address		
	ОК	Cancel		Help


Setting	Description
Action Mode	Select the type of operation performed at designated time.
[Bit ON]	At start time, turn ON the specific bit. At end time, turn OFF the bit. Example: Start time = 09:00:00 End time = 17:00:00
	Start time         End time           ON
[Bit OFF]	At start time, turn OFF the specific bit. At end time, turn ON the bit. Example: Start time = 09:00:00 End time = 17:00:00 Start time End time ON OFF 09:00:00 12:00:00 17:00:00
[Word write]	At start time, the specific [Write start value] is written to the action address. At end time, [Write end value] is written to the action address. Example: Device address = LW100 Start time = 09:00:00 End time = 12:00:00 Write start value = 10 Write end value = 0



	LW100	10	LW100	0	
		09:00:00		12:00:00	<b>→</b>
Action address	Specify the a	ddress wher	e the scheduler	performs a	ctions on.



Setting	Description								
Power-ON	Select the action to perform when power is turned on.								
start/end	Enable								
action	If the MT8000 power is turned ON within the scheduler range, the start action is performed. If the MT8000 power is turned ON outside of the scheduled range, the termination action is performed. Inside the scheduled range:								
	Start time Power ON End time								
	Start action Termination action								
	Outside the scheduled range:								
	Power ON Start time End time								
	Termination action Start action Termination action								
	<ul> <li>Disable         If power is turned ON but the time is later than the Start Time, the action is not automatically performed. However, the termination action is automatically performed.         Also, if the termination action is not set, the schedule range is unable to recognize and the action is not performed.     </li> </ul>								
Word write	These settings are active only when Action Mode is set to [Word								
value Settings	Write].								

When performing start action, the system will write this value into
action address.
[Write start value]
For [Constant]
Designates the value to be written at start time.
For [Address]
Designates the address used to store the start time value.
[Write end value]
When performing end action, the system will write this value into
action address.
For [Constant]
Designates the value to be written at end time.
For [Address]
Designates the address used to store the end time value.
Note
• You can use this option if the [Enable termination action] in [Time
Set] tab is selected.

Time Set tab (when [Constant] is selected)



Setting	Description				
Constant/Addres	Select the method to set the start time and end time.				
S	Constant				
	Specifies a fixed time and day.				
	Address				
	<ul> <li>Description</li> <li>Select the method to set the start time and end time.</li> <li>Constant <ul> <li>Specifies a fixed time and day.</li> </ul> </li> <li>Address <ul> <li>Address</li> <li>The start/end time is retrieved from the device address at or line operation.</li> </ul> </li> <li>Enable <ul> <li>Start time and end time can be set in different day of we There is only one start time and one end time during the we You have to set both start time and end time with this mode.</li> </ul> </li> </ul>				
	line operation.				
Setting on	Enable				
individual day	Start time and end time can be set in different day of week.				
	There is only one start time and one end time during the week.				
	You have to set both start time and end time with this mode.				







	(For example)	Monday	Tuesday
	Start day: Monday Start: 22:00:00 End: 01:00:00	22:00:00	01:00:00
Start	Set the start time and day.		
	When [Setting on individual day	y] is disabled, u	iser can designate
	more than one day.		
End	Set the end time and day.		
	When [Enable termination acti specified.	on] is selected,	, the end time can be
	The day settings can only be s enabled.	et when [Settin	g on individual day] is



#### Time Set tab (when [Address] is selected)

If "address" mode is selected, the system retrieves the start/end time and day from word devices. Therefore, users can set and change scheduled time in operation.

cheduler
General Time Set Prohibit
O Constant 💿 Address
Time setting address
PLC name : Local HMI
Address : LW 🗸 0
Control :
Status : 0 + 1
Action mode : $0 + 2$
Start time (day) : $\Box + 3$
Start time (hour) : $\Box + 4$
Start time (minute): $\Pi + 5$
Start time (second) : 0 + 6
End time (day) : 0 + 7
End time (hour) : 0 + 8
End time (minute) : 0 + 9
End time (second) : 0 + 10
OK Cancel Help

User designates the [Time setting address] as the top address used to store time settings data. The 11 word devices are automatically allotted.

Normally the format of the above word devices is 16-unsigned integer. If a 32-bit word device is chosen, only 0-15 bits are effective and users should zero the 16-31 bits.

a. Control (Time setting address + 0)



The layout of the Control word is shown below. Users set the [time acquisition request bit] ON  $(0 \rightarrow 1)$  to make the system reads the [Action mode], [Start time], and [End time] and uses them as the new scheduled time.

15	0	Bit
Reserved (0 fixed)	0	

Bit 00: time acquisition request bit (0: no action, 1: perform time read)

**NOTE** The system would not read start and end time data unless the [time acquisition request bit] is set ON.

**b. Status** (Time setting address + 1)

The layout of the Status word is shown below.

When the system competes the read operation, it will turn the [time acquisition complete bit] ON  $(0\rightarrow 1)$ . Also, if the read time data is incorrect, the [error notification bit] will be turned ON  $(0\rightarrow 1)$ .

15	)2	01	00	Bit
Reserved (0 fixed)		0	0	

Bit 00: time acquisition complete bit (0: null, 1: read operation complete) Bit 01: error notification bit (0: no error, 1: start or end time format is incorrect)

**NOTE** After system reads the time data and turns the [time acquisition complete bit] ON, be sure to turn [Control] [time acquisition request bit] OFF. Once this bit is turned OFF, the system will set both the [Status] [time acquisition complete bit] and [error notification bit] to OFF.





#### c. Action mode (Time setting address + 2)

Enable and disable the [Termination time action] and [Setting on individual day].

15 0	2 01	00	Bit
Reserved (0 fixed)	0	0	

Bit 00: Termination time setting (0: disable, 1: enable) Bit 01: Setting on individual day (0: disable, 1: enable)

- **NOTE** 1. If [setting on individual day] is OFF, the system still reads all 11 word devices but ignores the end time data.
  - 2. If [setting on individual day] is ON, be sure to enter all start and end time information. If 2 or more of the start/end day bits are turned ON simultaneously,

an error occurs.

d. Start/End Day (Start Day: Time setting address + 3, End Day: Time setting address + 7)

Designates the day used as a trigger for the start/termination action.

15	07	06	05	04	03	02	01	00	Bit
Reserved (0 fixed)		Sat	Fri	Thu	Wed	Tue	Mon	Sun	

Bit 00: Sunday (0: none, 1: select) Bit 01: Monday (0: none, 1: select) Bit 02: Tuesday (0: none, 1: select) Bit 03: Wednesday (0: none, 1: select) Bit 04: Thursday (0: none, 1: select) Bit 05: Friday (0: none, 1: select) Bit 06: Saturday (0: none, 1: select)

e. Start/End Time (Start Time: Time setting address + 4 to + 6, End Time: Time setting address + 8 to + 10)
Set the time values used for the start/termination actions in the following ranges. Hour: 0 - 23
Minute: 0 - 59
Second: 0 - 59
If you specify a value outside the range, an error will occur.

**NOTE** The time data format shall be *16-bit unsigned,* system doesn't accept BCD format.



#### Prohibit tab

Scheduler	×
General Time Set Prohibit	
Prohibit Use prohibit function	
PLC name : Local HMI	
Address : LB V 0	
Sound Enable Sound Library Play	
OK Cancel Help	

Setting	Description		
Prohibit	Enable		
	MT8000 reads the bit status before performing start action. If		
	the bit is ON, the schedule action is not performed.		
Sound	Enable		
	When performing start and termination action, the system will		
	simultaneously play the specified sound.		



### **Restrictions:**

- User can register the maximum of 32 entries in Scheduler list.
- The time scheduler features are one time actions. When the start time or end time is reached, the system writes the value to device just one time. (not repeated)



- Once the system execute start action, it will read [Write start address] and [Write end address] altogether, after then, even you change the value of [Write end address], the system would not use the new value.
- When the operator changes RTC data, for those schedule object with both start time and end time setting, the system will check if the time update changes the status from out of schedule range to within schedule range, if it is, the start action will be performed.
- If there are several schedule objects registered the same start time or end time, when time up the system will perform the operation from the first to the last in ascending order.
- When [Time Set] are specified as [Address] mode, the system will read [control] word periodically.
- When [Time Set] are specified as [Address] and start time and end time is over valid range, the system may not execute operation properly.
- When [Time Set] are specified as [Address], the action will not start up until time data update is success.



## 13.29 Option List

### Overview

An Option List displays a list of items that the user can view and select. Once the user selects an item, the value corresponding to the item will be written to a word register. There are two forms for this object – Listbox and Drop-down list. The listbox lists all items and highlights the selected one. However, the drop-down list normally displays only the selected item. Once the user touches it, the system will display a listbox (which is similar to the one with Listbox style) beneath the object.



### Configuration

Click the "Option List" icon , "Option List object properties" dialogue box appears as follows:



Option list Mapping   Description :   Attribute   Mode :   Drop-down List   Background :   Selection :   Direction :   Down   Source of item data :   Item address   PLC name :   Local HMI   Address :   LW   Item address   PLC name :   Local HMI   Address :   LW   Item address   PLC name :   Local HMI   Address :   LW   Item address   Item address   Address :   LW   Item address   Monitor Device   Item address   Monitor I address   Item address   address  <	New Option List Object
Option ist Mapping   Description :   Attribute   Mode :   Drop-down List   Selection :   Direction :   Down   Source of item data :   Item address   PLC name :   Local HMI   Address :   LW   Item address   PLC name :   Local HMI   Address :   LW   Item address   PLC name :   Local HMI   Address :   LW   Item address   PLC name :   Local HMI   Address :   LW   Item address   O   Address :   LW   Item address   O   Address :   LOCAL HMI   Setting   Address :   Local HMI   V   Item address   O   PLC name :   Local HMI   V   Item address   V   Item address   V   PLC name :   Local HMI   V   Setting	
Attribute Mode : Drop-down List V Background : Direction : Down Source of item data : Item address PLC name : Local HMI Address : LW V O 16-bit Unsigned Control address PLC name : Local HMI Address : LW V O 16-bit Unsigned Control address PLC name : Local HMI Address : LW V O 16-bit Unsigned Item address Address : LW V O 16-bit Unsigned Item address Address : LW V O 16-bit Unsigned Monitor address PLC name : Local HMI Address : LW V O 16-bit Unsigned Direction : Direction : Monitor address PLC name : Local HMI Item address ASCII O UNICODE The length of each item : PLC name : Local HMI Address : LW V 10	Option list Mapping Security Shape Label
Attribute Mode : Drop-down List  Background : Selection : Direction : Down  Source of item data : Item address PLC name : Local HMI  Address : LW  0 16-bit Unsigned Control address PLC name : Local HMI  Setting Address : LW  0 16-bit Unsigned Identity of the setting is the set is the setting is the set is the setting is t	Description :
Prode : Drop-down List   Direction : Down   Source of item data : Item address   PLC name : Local HMI   Address : LW   UW 0   16-bit Unsigned     Control address   PLC name : Local HMI   Address : LW   LW 0   Item address   PLC name : Local HMI   Address : LW   LW 0   Address : LW   Local HMI Setting   Address : LW   Local HMI Setting   Address : LW   Local HMI Setting	Attribute Destruction Destruction
Selection : Direction : Down Source of item data : Item address PLC name : Local HMI Address : LW O 16-bit Unsigned Control address PLC name : Local HMI Address : LW O Eddress : Setting Address : Setting Address : LW O Eddress : Setting Address : LW O Eddress : Setting NORD(s) PLC name : Local HMI Address : LW O Eddress : Setting Address : LOCAL HMI Address : LOCAL HMI Addres	Mode : Drop-down List Y Background :
Direction : Down Source of item data : Item address Monitor address PLC name : Local HMI Address : LW O Control address PLC name : Local HMI Address : LW O [Address] : set 1 to update item data [Address] : set 1 to update item data [Address] : set 1 to update item data [Address] + 1 : item count Item address O ASCII O UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW O 10 10 10 10 10 10 10 10 10 10	Selection :
Source of item data : Item address Monitor address PLC name : Local HMI Address : LW O Control address PLC name : Local HMI Address : LW O [Address] : set 1 to update item data [Address] : set 1 to updat	Direction : Down
Monitor address PLC name : Local HMI Address : LW O G G Control address PLC name : Local HMI Address : LW O G G Address : LW O G G Address] : set 1 to update item data [Address] : set 1 to update item data [Address] : 1 : item count Item address ASCII UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Setting Address : LW O G G G G G G G G G G G G G G G G G G	Source of item data : Item address
PLC name : Local HMI Address : LW O 16-bit Unsigned Control address PLC name : Local HMI Address : LW O [Address] : set 1 to update item data [Address] : set 1 to update item data [Address] + 1 : item count Item address O ASCII O UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW O 10 O O O O O O O O O O	Monitor address
Address : LW V O I6-bit Unsigned Control address PLC name : Local HMI V Setting Address : LW V O [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII O UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI V Setting	PLC name : Local HMI Setting
Control address PLC name : Local HMI Address : LW O [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW I0	Address : LW 🗸 0 16-bit Unsigned
Control address PLC name : Local HMI  Address : LW  O [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII OUNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW  I0	
Control address         PLC name :       Local HMI         Address :       LW         Iddress] :       set 1 to update item data         [Address] :       set 1 to update item data         [Address] + 1 :       item count         Item address       Item address         ASCII       UNICODE       The length of each item :         PLC name :       Local HMI       Setting         Address :       LW       10	
Control address PLC name : Local HMI   Address : LW  (Address] : set 1 to update item data [Address] + 1 : item count  Item address ASCII UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW  10	
Control address PLC name : Local HMI  Address : LW  O [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII OUNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW  I0	
Control address PLC name : Local HMI  Address : LW  O [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII OUNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW  10	
PLC name : Local HMI Address : LW [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW 10	- Control address
Address: LW Address: LW [Address]: set 1 to update item data [Address] + 1 : item count Item address ASCII UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW 10 Setting	
Address : LW [Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII OUNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Address : LW 10	PLC name : Local HMI
[Address] : set 1 to update item data [Address] + 1 : item count Item address ASCII OUNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Setting Address : LW 10	Address : LW V 0
[Address] + 1 : item count Item address ASCII O UNICODE The length of each item : 5 WORD(s) PLC name : Local HMI Setting Address : LW 10	[Address] : set 1 to update item data
Item address         ASCII       UNICODE       The length of each item : 5       WORD(s)         PLC name :       Local HMI       Setting         Address :       LW       10	[Address] + 1 : item count
ASCII     UNICODE     The length of each item : 5     WORD(s)       PLC name :     Local HMI     Setting       Address :     LW     10	Item address
PLC name : Local HMI Setting	○ ASCII
Address : LW V 10	PLC name : Local HMI Setting
	Address : LW V 10

## Option list tab

Setting	Description
Attribute	[Mode]
	Select the object style; one of Listbox and Drop-down list.
	[Item no.]
	Set the number of items for the object. Each item represents a state displayed in the list and a value to be written to the [Monitor address].
	[Background]



	Select background color for the object.		
	[Selection]		
	Select background color for the selected/bigblighted item		
	[Source of item data]		
	• • • • • • • • • • • • • • • • • • • •		
	There are Predefine, Dates of historical data, and Item address for		
	selection.		
Predefine mode	Monitor address		
	Select the [PLC name], [Device type], [Address] of the word		
	register device that controls the display of the object and the system		
	writes the value of the item to the word register.		
	[Write when button is released]		
	If this function is selected, the operation is activated at touch up. If		
	the function is not selected, the operation is activated at touch		
	down.		
	NOTE		
	This option is only available in listbox style.		
Dates of	Item data from dates of historical data (History index mode)		
historical data			
mode	Option List object can be used with Historical Event-Display,		
	Trend-Display and Data-Display for displaying the History File on		
	the Historical Display objects as below illustration.		
	06/06/09		
	[Туре]		
	Alarm (Event) log is used to display Historical Event-Display		
	Data sampling is used to display Historical Trend-Display or		
	Data-Display.		
	-r - 7		



	[Date]
	Set the date format.
	[Data Sampling object]
	Users have to select which Data sampling object is triggered when selecting "Data sampling" as [Type].
	Users should select the same data sampling object with the one selected in Historical Trend-Display or Data-Display.
	Note:
	1. The system will automatically disable Mapping table when History Index mode is selected.
	2. When users select "Drop-Down List" in [Attribute] and enable History Index mode, the Option List displays "?" in Error State.
Item address mode	When selecting [Item address], users have to correctly set the content of [Control address] and [Item address].
	Control address
	[Address]
	Set "1" to the data of the designated register of this address for
	updating items displayed in Option List using the content of
	designated register of [Item address]. After updating, the data in this
	register will restore to "0".
	[Address] + 1
	The next address of the designated [Control address], data in this
	address is for setting the number of items.



Item address
This address is for storing the contents of the items.
[ASCII]
Use ASCII as item contents.
[UNICODE]
Use UNICODE as item contents, such as Chinese characters.
The UNICODE to be used must also be used in other objects. EasyBuilder8000 will then compile these font files in advance, and save to HMI when downloading, only in this way the UNICODE can be displayed correctly.
[The length of each item]
As for item length, it's now restricted to less than 1024 when [number of items] times [The length of each item].
<b>Note</b> : The system will automatically disable Mapping table when Item address mode is selected.

Mapping tab



Nev	New Option List Object 🛛 🔀				
Ot	ptio	on list Mapp	ing Security Shape	Label	
		Item	Value	Item data	
		0	0	test1	•
		1	1	test2	
		2	2	test3	
		3	3	test4	
		4	4	test5	
	▶ 5 (error) ?				
					Set default
1 5					
-	- Error notification				
			Frahle	Set OM	O Set OFF
				Set ON	U Set OFF
		PLC name	e: Local HMI		V Setting
		Addres	S: IB		
		Addres			

Setting	Description	
Mapping table	This table displays all available states/items, their item data	
	and values. To change the number of available items, please	
	refer to [Option list tab] $\rightarrow$ [Attribute] $\rightarrow$ [Item no.].	
	[Item]	
The system lists all available items. Each item represent state that will be displayed in the list. This field is read-		
	[Value]	
	Here user can assign value for each item, basing on the following two criteria:	
	a. [For reading]: If any change of the content from [Monitor	



	<ul> <li>address] is detected, the object compares the content with these values and selects the first matched item. If no item is matched, the status goes to error state and signals the notification bit register (if requested).</li> <li>b. [For writing]: The system writes this value to [Monitor</li> </ul>		
	address] when user selects an item.		
	[Item data]		
	Users can assign data for each item. The option list object displays the data of all items in the list for users to review and select.		
	[Error state]		
	<ul> <li>For example, item 8 is the error state when specifying 8 in [Item no.]. Similarly, if you set [Item no.] to 11 then state 11 would be the error state, and so on.</li> </ul>		
	b. On error state, the listbox-style option list removes the highlight to represent no item is selected and the drop-down list displays the data of error state.		
	c. The item of error state is only applied to the drop-down list style. The listbox-style list has nothing to do with this item.		
[Set default]	Set default values for all states, i.e. set 0 for item 0, 1 for item		
	1, and so on.		
Error	The system will set ON/OFF to the specified bit register when		
Notification	error is detected. The signal of the bit register could be used to		
	trigger a procedure for correcting the error.		



## 13.30 Timer

### Overview

Use timer variables to enable timer instructions. Timer variables consist of the following six special variables.

Timer Variable	Variables Type	Description
Input bit (IN)	Bit type	The master switch of timer.
Measurement bit	Bit type	Turn ON when the timer begin
(TI)		counting.
Output bit (Q)	Bit type	Activate when the timer finish
		counting.
Preset time (PT)	Word type	Set the timer value.
Elapsed time (ET)	Word type	Display current elapsed value of
		timer.
Reset bit (R)	Bit type	Reset the elapsed time (ET) to 0.

#### Configuration

Click the "Timer" icon , "Timer object properties" dialogue box appears as follows:

New Timer Object
Timer
Description :
Mode : Accumulated OFF delay V Time base : 0.1 second (s)
Input bit (IN)
PLC name : Local HMI Setting
Address : LB 🔽 0
Measurement bit (TI)
PLC name : Local HMI Setting
Address : LB 1
Output bit (Q)
PLC name : Local HMI Setting
Address : LB 2
Preset time (PT)
PLC name : Local HMI Setting
Address : LW 0 16-bit Unsigned
Elapsed time (ET)
PLC name : Local HMI
Address : LW 1 16-bit Unsigned
Reset bit (R)
PLC name : Local HMI
Address : LB 3
OK Cancel Help

Mode	Description
On delay	Point 1: When the IN turns ON, the TI be turned ON and
	the elapsed time ET increases. The Q remains OFF.
	Point 2: When the ET equals the PT, the Q be turned
	ON and the TI be turned OFF.







	<b>Point 2</b> : When the IN turns OFF, and if the ET is less than the PT, the TI be turned OFF. The ET is in the retentive state.
Q 1 2 3 4 5 6 7	<b>Point 3</b> : When the IN turns ON, the TI be turned ON. The timer measurement starts again and the ET is added to the kept value. The Q remains OFF.
	<b>Point 4</b> : When the ET reaches the PT, the TI be turned OFF and the Q be turned ON.
	<b>Point 5</b> : When the IN turns OFF, the Q be turned OFF. (Reset the ET to 0 by using Reset bit (R).)
Accumulated Off delay	<b>Point 1</b> : When the IN turns ON, the Q be turned ON and TI remains OFF.
	<b>Point 2</b> : When the IN turns OFF, the TI be turned ON and the elapsed time ET increases. (the Q remains ON)
1 2 3 4 5 6 7 8 9 10	<b>Point 3</b> : When the IN turns ON, the timer measurement pauses.
	<b>Point 4</b> : When the IN turns OFF, the paused timer measurement continues.
	<b>Point 5</b> : When the ET equals the PT, the TI and Q are turned OFF. (Reset the ET to 0 by using Reset bit (R).)



## 13.31 Video In

MT8000X series provide Video Input function. Users can install surveillance camera, then monitor the factory any time they want. The video images can also be stored in devices and play them with Media Player, or analyze them on PC.

This function can be utilized in different aspects. Apart from monitoring factory, it can also be used in driving device or Building Automation monitoring.

For hardware, MT8000X series provide 2 channels for Video Input. Users can freely switch channels to monitor, and capture images without being influenced when pause playing. The captured images will still be real-time external image input. The supported formats are NTSC and PAL.

Video In Object's Properties
General Profile
Description :
Encode format : NTSC 🗸
Capture address
☑ Use capture function
PLC name : Local HMI 🗸 Setting
Address : LB 🗸 0
Storage medium SD © USB 1 O USB 2
Record time
Before : 5 📚 seconds After : 5 📚 seconds
Control address
Use control function
PLC name : Local HMI
Address : LW 🗸 0 16-bit Unsigned

Setting	Description
Use	Definition: For inputting external video image into HMI and play
Control	it with HMI.



Function	Illustration:		
	Control address		
	Use control function		
	PLC name : Local HMI Setting		
	Address : LW V 100 16-bit Unsigned		
	Suppose [Control Address] is designated as "LW100":		
	A. Users can set [Control Address+ 0] to enable/stop Video		
	Input function.		
	$[LW100] = 0 \rightarrow Stop Playing.$		
	[LW100] = 1 $\rightarrow$ Input video image in VIP 1 and display it in		
	screen.		
	[LW100] = 2 $\rightarrow$ Input video image in VIP 2 and display it in		
	screen.		
	[LW100] = 3 $\rightarrow$ Input video image in VIP 1 but don't display it		
	in screen. In this way users can still execute Capture image.		
	[LW100] = 4 $\rightarrow$ Input video image in VIP 2 but don't display it		
	in screen. In this way users can still execute Capture image.		
	B. Users can set [Control Address +1] to control the displaying		
	of video image:		
	[LW101] = 1 $\rightarrow$ Pause/Continue playing.		
	C. If users change value in [Control Address + 0], the system		
	will keep the new value.		
	D. If users change value in [Control Address + 1], system will		
	execute the corresponding command first then erase the		
	new value and set it back to "0".		
	E. If not using [Control Function], system will play the channel		
	set in [Input channel] automatically.		



WE!NTEK

Capture address		
☑ Use capture function		
PLC name : Local HMI Setting		
Address : LB 🗸 0		
Storage medium O SD O USB 1 O USB 2		
Record time Before : 5 seconds After : 5 seconds		
Take the illustration above as sample, set <b>[Record time]</b>		
shanges from OEE to ON, system will be triggered to conture		
changes from OFF to ON, system will be triggered to capture ,		
one image each second, from 5 seconds before the triggering		
time to 5 seconds after the triggering time.		

Note:

WE!NTEK

- 1. Video In Object can only be used in MT8000X which supports VIP function.
- 2. Only video image in one channel can be input at any moment while running system.
- 3. Capture function won't be influenced by "pause" playing. The video image that should be played while not paused will still be captured.
- 4. Recommended Format and Resolution:

	1:1	50%
NTSC	720 x 480	360 x 240
PAL	720 x 576	360 x 288

This function only supports NTSC and PAL format.



# 13.32 System Message

Confirmation requir	ed		
Message :	Please confirm the operation		
	Use label library		
ОК :	ОК		
	Use label library		
Cancel :	Cancel		
	Use label library		
Font :	Arial		
Deny write-comma	nd		
Message :	The system is being prohibited from writing device registers!		
Font :	Arial		
	Use label library		
Allow write-comma	nd		
Message :	The system is now allowed to write device registers.		
Font :	Arial		
	Use label library		

Use this utility to edit messages that displays in popup message boxes.

Setting	Description
Confirmation	Display whenever security requires the user to confirm operation.
required	
-	The [Message] shown on confirmation dialogue, and the text label of the 2
	buttons [OK] and [Cancel], can all be set. Please use the same font for the
	labels of [Message], [OK] and [Cancel]. Additionally, only when selecting
	[Label Library] for [Message], the use of Label Library for [OK] and [Cancel]



	buttons can be enabled.
	Confirmation required
	Message : Please confirm the operation
	OK: OK
	Cancel: Cancel Use label library
	Font : Times New Roman
Deny	Display when system tag LB-9196 (local HMI supports monitor function
write-command	only) is turned ON.
Allow write-command	Display when system tag LB-9196 (local HMI supports monitor function only) is turned OFF.



## **Chapter 14 Shape Library and Picture Library**

EB8000 provides Shape Library and Picture Library features to add visual effects on objects. Each Shape and Picture includes up to 256 states. This chapter expatiates on how to create Shape Library and Picture Library.

For usage of shape and picture library, please refer to "Chapter 9 Object General Properties".

## 14.1 Creating Shape Library

A shape is a graph composed of lines, rectangles, and circles. A complete Shape can possess more than one state, and each state can include two parts: frame and inner. See the illustration below:





The frame and inner of a shape can be used separately or together by an object. Click **[Call up Shape Library]**, and the **[Shape Library]** dialogue appears as below:









Setting	Description		
Library	Shape Libraries which have been added into the current project.		
	Select the library source of a Shape from the list.		
State	Select the state to be displayed by current Shape. If the selected		
	Shape isn't displayed, it means that the Shape does not exist or the		
	state of the Shape isn't defined.		
Select Lib.	Click [Select Lib.], and the following dialog appears for users to		
	select the file path of the Shape Library to be added.		
	By previewing the content of the library right side of the window, users		
	can select suitable library.		







#### EasyBuilder8000 User's Manual

4	10 - WINDOW_010 X	●	Old name : [Unstitut New name : obc
23A	apt Library duary 2 vortical System Evane System Evane System Lamp 0: ShareO Sibile: 2 Laner	he D D D D D D D D D D D D D D D D D D D	Zelect Lib New Lib Unstlich Lib Unstlich Lib Delete All States Delete Cur State Resaure

The following shows how to create a new Shape Library and add a Shape with two states to it.

#### Step 1

Click **[New Lib.]** and input the name of the new Shape Library.

New Library
Library name : TEST
OK Cancel

A new Shape Library "TEST" will be added to the **[Shape Library]** dialogue. At this moment, no Shape is in the library.

Library :	State	: 0 💌	012	34567
button1 button2 TEST System Frame System Button System Lamp				
	0: States : 0 None	1: States : 0 None	2: States : 0 None	3: States : 0 None



r

9

 $\bowtie$ 

#### Step 2

Add a state to the selected Shape. First, use the drawing tools to draw a graph in the window and select the graph to be added to the Shape Library.



Chick the [Save to Shape Library] button in toolbar and the following dialogue appears.



Setting	Description		
Shape library	Select the Shape Library for the graph to be added to. In this		
	example, "TEST" library is selected.		
Description	The name of the Shape.		
Shape no.	The number in Shape Library current graph will be added in.		
State	Select the state of the Shape which this graph represents. In this		
	case the state is set "0". EB8000 provides 256 states for each		



	Shape.		
Frame	If [Frame] is selected, the graph will become a frame of the		
	Shape.		
Inner	If <b>[Inner]</b> is selected, the graph will become an inner part of the		
	Shape.		

This part shows the current status of the shape, at this moment shape [no. 0] in **[state 0]** in library "Test" is with undefined frame and inner.

Width: 77	Height: 73
Frame : undefined	
Inner : undefined	

After clicking **[OK]**, the graph will be added to Shape Library. Illustration below shows that Shape **[No.0]** in library "Test" has only one state, **[state0]**, and is defined as a frame.



#### Step 3

Likewise, create another Shape state by the same process as in Step 2, but this new graph has to be defined as **[state 1]**:


Save to Shape Library	2
Shape library	: TEST 💌
Description	: Untitled
Shape no.	: 0 🗸 State : 1 🔽
Frame     Ir	mer Width : 77 Height : 73 Frame : undefined Inner : undefined
<del></del>	OK Cancel

A complete Shape with two states is created. See the following picture.

bottonl	
System Frame System Button System Lamp	0: Untitled States : 2 Frame
tate 0 - State 11 :	4: States : 0 None



# 14.2 Creating Picture Library

Click the **[Call up Picture Library]** button in toolbar, and the **[Picture Library]** dialogue appears.

		) 🛃 🖻 🖪	9 🔊 🕻 9	1	
		Call up	Picture Library		
icture Library					
Library :			State : 🚺		12345
icon1 Button2 Demo graph0 graph1 map1 Computer				$\bigcirc$	
Button1 background		0:PB Red	1:PB Green	2:PB Yellow	3:PB Blue
button1		States : 2	States : 2	States : 2	States : 2
		30054 bytes	30054 bytes	30054 bytes	30054 bytes
		BMP	BMP	BMP	BMP
State 0 ~ State 11 :					
		4:Lamp Red	5:Lamp Green	6:Lamp Yellow	7:Lamp Blue
		States : 2	States : 2	States : 2	States : 2
		30054 bytes	30054 bytes	30354 bytes	30054 bytes
		BMP	BMP	BMP	BMP
Select Lib	New Lib	Unattach Lib.	Rename	Export	
Import Picture	Modify Picture		Delete All States	Delete Cur. State	e Close
			Insert Before	Insert After	

Setting	Description
Library	Picture Libraries which have been added into the current project.
	Select the library source of a Picture from the list.
State	Select the state that current graph represents. If the selected
	Picture isn't displayed, it means that the Picture does not exist or
	the state of the Picture isn't defined.
Select Lib.	Click [Select Lib] and the following dialog appears for users to
	select the file path of the Picture Library to be added.
	By previewing the content of the library right side of the window,
	users can select suitable library.



	Onen 🛛
	New Folder
	Boodground //b     Boodground //b     Boodground //b     Boodground //b     Boodground //b     Boodground //b     Boodground //b
	Documents Ebutton1_32./fb Eflag_EUROPE.
	Button2.fb Bicon1.fb
	Desktop Button2_32./b Employer.fb
	button3.fb @M18000_Demo
	My Documents
	Buttor4_32.fb BMT8000_Demo
	Mu Consulter .fb III Tank.fb
	File game: background ifb gen
	My Network Files of type: picture lib (".fib, ".bib) Cancel
New Lib.	Click the button to add a new Picture Library.
	New Library
	Library name : Test
	OK Cancel
Unattach Lib.	Click the button to delete the Picture Library in [Library] from the
	current project.
	EasyBuilder 8000
	Do you really want to unattach this library ?
	Yes No
Delete All States	Delete all states of the selected Picture.
Delete Cur. State	Delete current state of the selected Picture.
Rename	Rename the selected Picture.
	Rename
	Old name : Red
	New years - Plus
	Me. Manne - Droc
	OK Cancel
Insert Before	Add a new state before the current state.
Insert After	Add a new state after the current state.
Import Picture	Add a new picture to the Picture Library.



	Import Picture       Picture       Picture no. : 0       Picture name : 123       Next       Cancel
Modify Picture	Modify the selected picture.
Export	Export the selected picture to the appointed place. As shown below, users can get the original picture.
	Save jn: EB8000 Adviver_win32 Adviver_win32 Advivers_x86 Cocuments My Recent Documents File name: Save File name: Save
	File name:     Save       My Network     Save as type:       Bitmap file.(*.bmp)     Cancel

**Note:** The compatible picture format are \*.bmp, \*.jpg, \*.gif, \*.dpd, and \*.png. When adding a GIF picture in Picture Library, if this picture file is animated, the number of times to play this animation can be set by users as below.





The example below shows how to create a new Picture Library and add a Picture with two states into it.

### Step 1

Click [New Lib.] and input the name of the new Picture Library.

New Library	×
Library name : TEST	
OK Cancel	

A new Picture Library "TEST" will be added to the **[Picture Library]** dialogue. At this moment, there is no Picture in the library.



### Step 2

Prepare the pictures to be added; suppose the two graphs below are used to represent state 0 and state 1 respectively.



Click [Import Picture] and a dialogue appears as below. Set [Picture no.] and [Picture name] for it, and then click [Next].

Import Picture
Picture
Picture no. : 0 💌
Picture name : F YELLOW
Next Cancel

#### Step 3

When the dialogue below is shown, select the source of picture for state 0, and select the correct transparent color. In the example below, the blue color RGB (0, 0, 255) is a transparent color. After the settings of the state 0 are completed, click [Next] button to continue the settings of the other state.



Before choosing transparent color, check **[Transparent]** box first and then left click on location-to-be of the graph. At this time, EB8000 will automatically display RGB value of the transparent color. Take above as an example, the actual shape shown as below:



#### Step 4

WEINTEK

Likewise, select the source of a picture for state 1 and select the correct transparent color for it. After the settings are completed, click the **[Finish]** button.





WE!NTEK

Below shows the complete picture created. A new picture "F Yellow" can be found in the [Picture Library] dialogue. From the information we know the picture is in the format of bitmap and with two states.

Library :	TEST	~
	$\bigcirc$	
Graph name :	0:F Yellow	1:
Total states :	2	
lmage size :	30054	0
	ВМР	



# Chapter 15 Label Library and Multi-Language Usage

Label Library is used in the Multi-Language environment. Users can design the content of Label Library to meet their demands. Select the suitable label from Label Library when text is needed.

## **15.1 Introduction**

The system in operation will display the corresponding text to the language in use according to the settings. EB8000 supports 8 different languages simultaneously. Click **[Label Library Manager]** and the dialogue appears as below:



1	Font												
La	anguage no. State no.	: 8 : <b>]</b>		× ×		0 1 :	23	45	6 7				
D.	Label tag na	me N	o. of states	Lang	uage 1	Languag	je 2 L	anguage 3.	Language 4	Language 5	Language 6	Language 7	Language 8

Setting	Description
State no.	Indicates the current state; each Label has maximum of 256
	states (state no. 0~255). The State no. is determined by
	[Language no.] selected. If user use 8 languages, 256/8=32



	(states), if user use 4 languages, 256/4=64 (states).
Now	Add a new lobal tag
New	Adu a new label lay.
Settings	Modify the content of Label.
Delete	Delete the selected Label.
Delete All	Delete all current label tags
Сору	Copy the content of the label.
Paste	Paste the copied label.
Save Label File	Save all current label tags as .lbl file
Load Label File	Load existing .lbl file to label library
Export EXCEL File	Export the current label tag library in csv or xls file format. It is
	allowed to select one language or all to be exported. This
	function does not support UNICODE.
Import EXCEL File	Import a label tag library (csv or xls file format) to the current
	project (MTP). It is allowed to select one language or all to be
	imported. This function does not support UNICODE.



# 15.2 Settings of Font of Label Library

In **[Label Tag Library]** users can see the existing tag and the languages this tag contains. Different fonts can be selected for different languages.

Tag Library			
I Font			
ont		Comment	
Language 1 : [	Antique Olive Compact	English	
Language 2 :	Book Antiqua Bold	Chinese	
Language 3 :	Century Gothic Bold	Japanese	
Language 4 :	CourierPS Oblique	Korean	
Language 5 :	Euro Sign	French	
Language 6 :	Cataneo BT	Spanish	
Language 7 :	Calligraphic 421 BT	Italian	
Language 8 :	Helvetica Narrow Bold Oblique	Arabic	

## [Font]

Under the Multi-Language configuration, users can select font type for each language.

## [Comment]

Input the comment of each language.



# 15.3 How to Create a Label Library

The following illustrations show how to create a Label Library.

First of all, open the **[Label Tab Library]** dialogue and click **[New...]**. Correctly input the settings as shown below and then click **[OK]**.



## [Label name]

The name of label. In this case it is set "Pump Alarm".

#### [No. of states]

The number of states possessed by the Label.

When the process is complete, a new Label "Pump Alarm" with 2 states will be added to the Label Library. See the picture below.

bel Ta	ig Library									
adel L	Font anguage no. : 8 State no. : 0		<ul><li>✓</li><li>✓</li><li>✓</li><li>✓</li></ul>	0 1 2 (	3 4 5 (	6 7				
No.	Label tag name	No. of states	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7	Language 8
1	Pump Alarm	2								
			la de la companya de				¢\$			

Select "Pump Alarm", click **[Settings ...]** and the **[Label Tag Content]** dialog appears for users to set up the corresponding language content.

EasyBuilder8000 User's Manual

ibel Tag Content				
Label Content Settings				
Label name : Pump	Alarm	State no. : 🛛 💌	• • 0 1 2 3	\$ 5 6 7
Language 1	Language 2	Language 3	Language 4	
how are you	▲【你好	<u></u>	<u>~</u>	~
Language 5	Language 6	Language 7	Language 8	
				2
Change No. of States				





# 15.4 Using Label Library

When there are already some defined labels in Label Library, users can find those Labels in **[Label tag]** by selecting **[Use label library]** in the object's **[Label]** tab.

New Bit Lamp Object
General Security Shape Label
Use label Use label library Use label library Use bitmap font Use bitmap font Label 0 pump Alarm
Language : 1 State : 0 Attribute
Font : Arial Black
Italic     Underline       Duplicate these attributes to       Every state     Every language
Movement Direction : No movement
Content :
How are You
Tracking Duplicate this label to every state

When **[Use label library]** is selected, **[Content]** dialog shows the content of selected label tag and the settings of Font type are also included in the Label Library.



# 15.5 Settings of Multi-Language

When users would like to have the object's text to show multi-language, except for using Label Library, it needs to use the system reserved register [LW-9134: language mode]. The value of [LW-9134] can be set from 0 to 7. Different data of [LW-9134] corresponds to different Languages. The way of using LW-9134 will differ if the languages are not all chosen when compiling the downloaded file.

For example: If 5 languages are defined by user in Label Library as Language1 (Traditional Chinese), Language2 (Simplified Chinese), Language3 (English), Language4 (French), and Language5 (Japanese). If only Language 1, 3, 5 are downloaded by user, the corresponding language of the value in LW-9134 will be 0-> Language1 (Traditional Chinese), 1-> Language3 (English), 2-> Language5 (Japanese).

Another example below demonstrates how to use multi-language feature.

First of all, create a [Text Object]:

New Text Object	×
Text	
Use label library Label tag : Test	~
Use bitmap font Label Library	
Language : 1	
Attribute	
Font : Arial	~
Color : Size : 16	~
Align : Left 💙 Blink : None	×
Italic Underline Duplicate these attributes to	
Every language	
Movement Direction : No movement	
Content :	
English	
<	~

Next, create a **[Numeric Input]** Object. Set its Read address as below: The Read address in use is the system reserved register [LW-9134].

Jeneral	Data Entry	Numeric Format	Security	Shape	Font	]	
1	Description :						
Read	address						
	PLC name :	Local HMI				× [	Setting
						and the second se	
	Address :	LW-9134 (16bit)	: language	mode	*		
	Address :	LW-9134 (16bit)	: language	mode	*		
Address	Address : s	LW-9134 (16bit)	: language	mode	~		
Address	Address : s PLC name :	LW-9134 (16bit) Local HMI	: language	mode	~		
Address	Address : s PLC name : vevice type :	LW-9134 (16bit) Local HMI LW-9134 (16bit)	: language	mode	~		×
Address	Address : s PLC name : pevice type : Address :	LW-9134 (16bit) Local HMI LW-9134 (16bit) LW9134	: language : language	mode mode ]System t	ag		× •

The following illustrations are the results of simulation.

WE!NTEK

When the value of [LW-9134] is changed, the content of the Text Object will also be changed automatically.





# Chapter 16 Address Tag Library

# 16.1 Creating Address Tag Library

Users are generally recommended to define commonly-used addresses in the Address Tag Library when start to build a project. It not only avoids inputting addresses repeatedly but also expresses the function of an address more clearly. Click **[Address Tag Library Manager]** in toolbar to call up the **[Address Tag Library]** dialogue as below.



ress I	ag Library						
) Custo	mized OS	ystem					
lo.	Address tag n	ame		PLO	C name	Address type	Address
	Tag_0	Restrict.		Loi	cal HMI	Bit	LB-0
	Tag 1			Lo	cal HMI	Word	LW-0
			JUL				
Jsers c	an import MT50	0 tag to represent the 4	address.				
Jsers c	an import MT50 New	0 tag to represent the Delete	address.	Settings			
Jsers c Sav	an import MT50 New e Tag File	0 tag to represent the Delete Load Tag File	address.	Settings			

Setting	Description
Customized	Display the Address Tags defined by users.
System	Display the Address Tags reserved by system.
New	Add a new Address Tag.
Delete	Delete a selected Address Tag.
Delete All	Delete all current Address Tags.
Settings	Modify the selected Tag.
Save Tag File	Save all current Address Tags as .tgl file.
Load Tag file	Load existing .tgl file to Address Tag Library.
Export CSV	Export current Address Tag Library to the appointed space in CSV
	format.



Import CSV	Import the saved CSV file of Address Tag Library to current project.
Export EXCEL	Export current address tag library to the appointed space in XLS
	format.
Import EXCEL	Import the saved XLS file of address tag library to current project.

The picture below shows system reserved registers.

	omized 🕑 by	Jstem					
lo.	Address tag n	ame		PLC name	Address type	Address	1
	LB-9000 : init	ialized as ON		Local HMI	Bit	LB-9000	
1	LB-9001 : init	ialized as ON		Local HMI	Bit	LB-9001	
l	LB-9002 : init	ialized as ON		Local HMI	Bit	LB-9002	
ř.	LB-9003 : init	ialized as ON		Local HMI	Bit	LB-9003	
j 👘	LB-9004 : init	ialized as ON		Local HMI	Bit	LB-9004	
i	LB-9005 : init	ialized as ON		Local HMI	Bit	LB-9005	
'	LB-9006 : init	ialized as ON		Local HMI	Bit	LB-9006	
}	LB-9007 : init	ialized as ON		Local HMI	Bit	LB-9007	
}	LB-9008 : init	ialized as ON		Local HMI	Bit	LB-9008	
ĵ.	1.0.0000			1		1.5.0000	>
Users	can import MT50	0 tag to represent the -	address.				
	New	Delete	Delete All	Settings			
Sa	ive Tag File	Load Tag File					
Ē	xport CSV	Import CSV	Export EXCEL	Import EXCEL	ſ	Exit	_

Before using the Address Tag Library, users need to add the content of the library first. Click **[New...]**, and the **[Address Tag]** dialogue appears as below:

	Address Tag
	Tag name : Test tag
	PLC name : MITSUBISHI FX0n/FX2
	Address type : O Bit 💿 Word
	Device type : TV
	Address : 200
	Address format : DDD [range : $0 \sim 255$ ]
	OK Cancel
L	
ing	Description
name	The name of the Address Tag.



PLC name	The name of the PLC which is selected from the [device list].
Address type	The type of Address; there are [bit] and [word] types available.
Device type	The type of the device; the types available are related to [PLC
	name] and [Address type].
Address	The content of the address.

Click **[OK]** when the settings are done, and a new tag will be found in the **[Customized]** library as below.

Address Tag Library						
⊙ Cus	tomized OSystem	1				
No.	Address tag name	PLC name	Address type	Address	Read/Write	
1	Alarm	MITSUBISHI FX0n/FX2	Bit	X-0	Read/Write	
2	temperature	MITSUBISHI FXOn/FX2	Word	TV-100	Read/Write	
3	Test tag	MITSUBISHI FX0n/FX2	Word	TV-200	Read/Write	



# 16.2 Using Address Tag Library

After creating the Address Tag Library, select the related PLC in **[General]** tab while adding a new object and click **[Setting...]**. Check **[User-defined tag]**, the tags can now be used as shown below.

eneral Security S	Shape Label			
Description :				
Mode :	Value 💌	Offset :	0	
Read address				
PLC name :	MITSUBISHI FX0n/FX2		~	Setting
Address :	temperature	TV100		16-bit Unsigned
ldress				
Idress PLC name :	MITSUBISHI FXOn/FX2			(
Idress PLC name : Device type :	MITSUBISHI FXOn/FX2 temperature			· · ·
Idiress PLC name : Device type : Address :	MITSUBISHI FX0n/FX2 temperature TV100	[	User	-defined tag
dd ress PLC name : Device type : Address : Address format :	MITSUBISHI FX0n/FX2 temperature TV100 DDD [range : 0 ~ 255]		User	✓ ✓ -defined tag
dd ress PLC name : Device type : Address : Address format :	MITSUBISHI FX0n/FX2 temperature TV100 DDD [range : 0 ~ 255]	Index register	User User	✓ ✓ -defined tag

There are some items in **[Device type]** for selecting.

Device type :	Temperature	~
Adamses :	Temperature	
AUUIESS .	Test Tag	

When the settings are completed, the window tree will show the name of the Address Tag used for the object as below.





# **Chapter 17 Transferring Recipe Data**

Recipe Data are stored in flash memory. When system start-up, both RW and RW\_A memory will be restored from the recipe data in flash memory. The way of reading and writing Recipe Data is the same as operating the normal Word Register.

The size of Recipe Data in RW is 512K words, and RW\_A is 64K words. User can update Recipe Data with SD Card, USB flash drive, USB cable or Ethernet and use this data to update data in PLC. It is possible to upload Recipe Data to the designated directory of PC; furthermore, it can save the PLC's data in recipe memory. The following explains all of the ways of operating recipe data.

# 17.1 Updating Recipe Data with Ethernet or USB cable

Click **[Download]** in Project Manager. Select **[RW]** and **[RW\_A]** and designate the directory of the source files. After downloading is completed, start up HMI again, and the contents of RW and RW\_A will be updated.

When **[Reset recipe]** is selected, before start downloading, EB8000 will set all the data of [RW] and [RW\_A] to "0" first.

EasyBuilder8000 User's Manual

WE!NTEK

D	ownload						
	✓ Firmware						
	✓ Project	C:\EB8000\project\test.xob	Browse				
	🗹 RW	PLEASE INPUT RECIPE FILE NAME !	Browse				
	☑ RW_A	PLEASE INPUT RECIPE 1 FILE NAME !	Browse				
	🗹 Data log	PLEASE INPUT DATA LOG FILE NAME !	Browse				
	☐ Install X-series media-p ☐ Startup screen	layer drivers					
	Connection © Ethernet	USB cable (i series only)	Þ				
	Reboot HMI after dow	nload 🗹 Reset recipe 🗌 Reset event I	og				
	Download	Stop Settings	Exit				



# 17.2 Updating Recipe Data with SD Card or USB Flash Drive

Click [Build Download Data for CF/USB Disk] in Project Manager.

This function is for building the download data and the settings shows as below.

Project Manager 🔀
Select the folder to save download data :
KA Browse
Sources
Browse
Recipe (RW)
PLEASE INPUT RECIPE FILE NAME ! Browse
Recipe A (RW_A)
PLEASE INPUT RECIPE_A FILE NAME ! Browse
🔽 Data log
Browse
Build

Insert SD card or USB flash drive to PC and click **[Browse...]** to assign the file path and then click **[Build]** to set all contexts of the download data. EB8000 will then build the sources into SD card or USB flash drive.

Note: The path of download data should avoid designating root directory of PC. For example, "**c**:\", also, directory name such as "**f**:\\" is illegal and should be written as "**f**:\".



# 17.3 Transferring Recipe Data

Use the **[Data Transfer (Trigger-based) object]** to transfer Recipe Data to the appointed address, or save the data of the designated address in [RW] and [RW\_A] as well. Please refer to the [Data Transfer (Trigger-based) object] section for more information.

# **17.4 Saving Recipe Data Automatically**

In order to prolong the life of flash memory of HMI, EB8000 will save Recipe Data automatically **every minute** to avoid losing data when HMI shuts down. EB8000 provides user with [LB-9029: save all recipe data to machine (set ON)] system register bit function to save Recipe Data manually. EB8000 will save Recipe Data when user sets ON to [LB9029]. But when user sets ON to [LB-9028: reset all recipe data (set ON)], EB8000 will clear all Recipe Date and return to "0".



# **Chapter 18 Macro Reference**

Macros provide the additional functionality your application may need. Macros are automated sequences of commands that are executed at run-time. Macros allow you to perform tasks such as complex scaling operations, string handling, and user interactions with your projects. This chapter describes syntax, usage, and programming methods of macro commands.

## 18.1 Instructions to the Macro Editor

- 2. Macro editor provides the following new functions:
  - a. displaying line number
  - b. Undo / Redo
  - c. Cut / Copy / Paste
  - d. Select All
  - e. Toggle Bookmark / Previous Bookmark / Next Bookmark / Clear All Bookmarks
  - f. Toggle All Outlining

The instructions below show you how to use these new functions.

 Open the macro editor; you'll see the line numbers displayed on the left-hand side of the edit area.

WEINTEK	EasyBuilder8000 User's Manual
Масто Масто list : [[D : 2] 10000 [ID : 3] 5000	<ul> <li>✓ </li> <li>✓ </li></ul>
WorkSpace Macro ID: 4 Macro_command main()	Macro name : macro_4
	edit area
Click the right mouse button to display edit menu.      GET/SET FN      Compile	Exit Help

4. Right click on the edit area to open the pop-up menu as shown below:

<u>U</u> ndo	Ctrl+Z
<u>R</u> edo	Ctrl+Y
Cu <u>t</u>	Ctrl+X
<u>С</u> ору	Ctrl+C
<u>P</u> aste	Ctrl+V
Select <u>A</u> ll	Ctrl+A
Toggle <u>B</u> ookmark	Ctrl+F2
Toggle <u>B</u> ookmark <u>N</u> ext Bookmark	Ctrl+F2 F2
Toggle <u>B</u> ookmark <u>N</u> ext Bookmark Pre <u>v</u> ious Bookmark	Ctrl+F2 F2 Shft+F2
Toggle <u>B</u> ookmark <u>N</u> ext Bookmark Pre <u>v</u> ious Bookmark Clear All Bookmarks	Ctrl+F2 F2 Shft+F2
Toggle <u>B</u> ookmark <u>N</u> ext Bookmark Pre⊻ious Bookmark Clear All Bookmarks Toggle All Outlining	Ctrl+F2 F2 Shft+F2

The disabled items are colored grey, which indicates that it is not possible to use that function in the current status of the editor. For example, you should mark a selected area to enable the copy function, otherwise it will be disabled.

Accelerators are supported as described in the menu.

 Above the edit area locates the toolbar. It provides "Undo", "Redo", "Cut", "Copy", "Paste", "Toggle Bookmark", "Next Bookmark", "Previous Bookmark" and "Clear All Bookmarks" buttons for instant use.



6. Modifications made to the editor will enable the undo function. Redo function will be enabled after the undo action is taken. To perform the undo/redo action, right click to select the item or use the accelerator (Undo: Ctrl+Z, Redo: Ctrl+Y).





Ctrl+A

Ctrl+F2

Shft+F2

F2

Select <u>A</u>ll

Toggle <u>B</u>ookmark

Previous Bookmark

Clear All Bookmarks

Toggle All Outlining

Update All Outlining

<u>N</u>ext Bookmark

 Select a word in the editor to enable the cut and copy function. After cut or copy is performed, the paste function is enabled.

Macro ID: 4

5

ጋር ነ 🖻 🖻 🦽 🌤 🌤

end macro\_com

abc abc

macro\_command main()

and

WorkSpace

edo<sup>Macro</sup> ID: 4

3

4

5

አ 🖻 🖻 🦽 🌤 🌤 🌤

macro\_command main()

end macro\_command

abc abc abc

WorkSpace				kSpace			
Macro ID : 4				Macro IE 으 으	): 4 % 🖻 🕻	≥ ^ % % %	,
1 2 macro_command main() 3 4 abc				1 2 macro_command main() 3			
5 end	<u>U</u> ndo <u>R</u> edo	Ctrl+Z Ctrl+Y		5	end ma	<u>Undo</u> <u>R</u> edo	C <b>trl+Z</b> Ctrl+¥
	Cu <u>t</u> Copy Paste	Ctrl+X Ctrl+C Ctrl+V			-	Cu <u>t</u> <u>C</u> opy <u>P</u> aste	Ctrl+X Ctrl+C Ctrl+V
	Select <u>A</u> ll	Ctrl+A			-	Select <u>A</u> ll	Ctrl+A
	Toggle <u>B</u> ookmark <u>N</u> ext Bookmark Pre <u>v</u> ious Bookmark Clear All Bookmarks	Ctrl+F2 F2 Shft+F2			-	Toggle <u>B</u> ookmark <u>N</u> ext Bookmark Pre <u>v</u> ious Bookmark Clear All Bookmarks	Ctrl+F2 F2 Shft+F2
	Toggle All Outlining Update All Outlining					Toggle All Outlining Update All Outlining	

8. Use "Select All" to include all the content in the edit area.

WorkSpace							
Macro ID: 4							
_ <u>_</u> _							
1							
2 macro_command m	2 macro_command main()						
4  int  a = 0							
5 float f = 1.12							
$6  char \ c = 0x40$	<u>U</u> ndo	Ctrl+Z					
7	<u>R</u> edo	Ctrl+Y					
8 end macro_comma	Cut	Ctrl+X					
	<u>С</u> ору	Ctrl+C					
	<u>P</u> aste	Ctrl+V					
	Select <u>A</u> ll	Ctrl+A					
	Toggle Bookmark	Ctrl+F2					



- 9. If the macro code goes very long, for easier reading, bookmarks are provided. The illustration below shows how it works.
  - a. Move your cursor to the position in the edit area where to insert a bookmark. Right

click, select "Toggle Bookmark". There will be a blue little square that represents a

bookmark on the left side of edit area.

WorkSpace						×
Macro 3	ID : 4			Macro name :	sub	
20	, X 🖻	B 13%	*			
18 19 20 21 22 23	macro short int gr bool k int te	command main( value=1,adres macroID=0, gn DError=0 empCount=0,tem	) s=0,i=0 Count=0 pVal=0	, gntestVal=0, gncorrectVal=0		~
24 25 26 27	int gr //>>: gnCur	nCurrentMID=0, Undo Redo	gnCurr Ctrl+Z Ctrl+Y	entMCount=0		
28 29 30	SetDa //>>>	Cu <u>t</u> <u>C</u> opy <u>P</u> aste	Ctrl+X Ctrl+C Ctrl+V	HMI", LW, 1000, 1) MID>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>		
32 33 34	E for : SetDa	Select <u>A</u> ll Toggle <u>B</u> ookmark	Ctrl+A Ctrl+F2	LW, adress, 1)		
35 36 37	temp\ valua adres	<u>N</u> ext Bookmark Pre <u>v</u> ious Bookmark Clear All Bookmark	F2 Shft+F2 s			
38 39 40	//>>: gnCur SetDa	Toggle All Outlining Update All Outlining	: :	MCount>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>		

- b. If there's already a bookmark where the cursor is placed, select "Toggle Bookmark" to close it, otherwise to open it.
- Right click and select "Next Bookmark", the cursor will move to where the next bookmark locates. Selecting" Previous Bookmark" will move the cursor to the previous bookmark.

#### WE!NTEK EasyBuilder8000 User's Manual WorkSpace WorkSpace WorkSpace WorkSpace Macro ID: 4 Macro ID: 4 Macro ID: 4 Macro ID: 4 X 电 光 耳 22 22 X 🖻 🖥 22 \* /s % % % <u>\_</u> macro macr macro command main() 19 macro 19 19 19 short value=1,adress=0,i=0 short 20 shor 20 shor 20 20 int gnmacroID=0, gnCount=0, gntestVal=0, gn int int 21 21 int g 21 21 bool 22 bool 22 bool 22 bool bError=0 22 int intint tempCount=0,tempVal=0 23 int t 23 23 23 24 int 24 int24 int gnCurrentMID=0, gnCurrentMCount=0 int g 25 D 25 25 25 <>>> 2.6 //>>> 26 11>> 2.6 Next<sup>-</sup>Bookmark 27 gnCu 27 gnCurrentMID=1 27 gnour 28 SetDa 28 SetDa 28 SetD 28 SetData(gnCurrentMID, "Local HMI", LW, 1000 17>> //>> 29 11>>> 29 29 D 30 30 30 30 xt Bookmark GetData(adress, "Local HMI", LW, 1, 1) GetDa 31 31 32 32 32 🖯 for i 33 🖯 for 33 🖯 🗗 🖬 33 □ for i=0 to 9 step 1 33 ĩ 34 SetDa 34 SetDa SetD 34 SetData(value, "Local HMI", LW, adress, 1) temp tempV 35 tempVal = value 35 35 templ 3.5 Next Bookmark unc (value) valu 36 value 36 value 36 37 adres 37 adre adress=func(adress) 37 adres 37 //>>> //>> 38 //>>> 38 38 38 gnCu gnCur gnCurrentMCount= 2 gnCur 39 39 39 SetDa 40 SetDa 40 SetD 40 SetData(gnCurrentMCount, "Local HMI", LW, 1 40 41 11>>> 41 11>> 41 11>: 42 //>> 42 //>> 42 42 //>>>

- d. Selecting "Clear All Bookmarks" will close all bookmarks.
- 10. Macro editor provides macro code outlining function, for easier viewing. This function is to hide macro codes that belong to same block, and display them with an  $\square$  icon. There will be a tree diagram on the left side of edit area. Users can click  $\square$  to hide the block or  $\square$  to open as shown below:

WEINTEK	EasyE	Builder8000 User's Manual
WorkSpace	WorkSpace	WorkSpace
Macro ID : 13	Macro ID: 13	Macro ID: 13
02 / BB / % % %	DC / BB / % % %	D G % h B 🔺 🛪 🛪
<pre>26 27 GetData(OK[0], "Local HMI", LB 28 29 日 for J =0 to 3 30 //&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt; set gnCurrentMC 31 gnCurrentMCount= 1 32 SetData(gnCurrentMCount, "Local 33 /'</pre>	26 .ocal H 27 26 27 27 28 29 30 29 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 30 20 20 30 20 20 30 20 20 30 20 20 30 20 20 30 20 20 30 20 20 30 20 20 30 20 20 30 20 20 20 30 20 20 20 30 20 20 20 30 20 20 20 20 30 20 20 20 20 20 20 20 20 20 2	26 27 GetData(OK[0], "Loo 28 29 40 47 //>>>>>>> asa 48 //GetData(gnmacroI) 49 gnmacroID = gnCurra 50 //SetData(gnmacroI) 51 //GetData(gnCount, 52 gnCount = gnCurran 53 //SetData(gnCount, 54 gntestVal = gnCurran 55 SetData(gnCount = 2 57 SetData(gnCorrectVal = 2 57 SetData(gnCorrectVal = 2) 57 SetData(gnCorrectVal = 2)
42 gnCurrentMCount= 2 43 SetData(gnCurrentMCount, "Loca	47 //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	58 //ASYNC_TRIG_MACRO 59 bError = assert(gm
44 //>>>>>>> set gnCurrentMC	49 gnmacroID = gnCurrentMI	1 60 //>>>>>>>>> ass
45 L next J	50 //SetData(gnmacroID, "L	61
40 47 //>>>>>>> assert >>>>> 48 //GetData(gnmacroID, "Local HM 49 gnmacroID = gnCurrentMID	52 gnCount = gnCurrentMCou 53 //SetData(gnCount, "Loc 54 gntestVal = gnCurrentMC	63 end macro_command

11. Right click to select "Toggle All Outlining" to open all macro code blocks.

WorkSpace	WorkSpace
WorkSpace         Macro ID:       13         Image: Second Secon	WorkSpace         Macro ID:       13         Image: Second Secon
56     gncor     Previous Bookmark     Shft+F2       57     SetDa     Clear All Bookmarks     1 HMI",       58     //ASY     Toggle All Outlining     nCount,       59     bErro     Update All Outlining     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	40 - end if 41 //>>>>>>> set gnCurrentMCount>>>>> 42 gnCurrentMCount= 2 43 SetData(gnCurrentMCount, "Local HMI", LW 44 //>>>>>>>> set gnCurrentMCount>>>>> 45 next J



12. Sometimes the outlining might be incorrect since that the keywords are misjudged. For

example:



To solve this problem <sup>,</sup> right click to select "Update All Outlining" to retrieve correct outlining.

WorkSpace		
Macro ID: 18		
L 🗅 🗠 🐇 🖻 🛍 🦽 🕉 🥉		
1 2 macro_command main() 3 4 9 1 1 1 1 1 1 1 1 1 1 1 1 1		

13. The statements enclosed in the following keywords are called a "block" of the macro code:

- a. Function block: sub end sub
- b. Reiterative statements:
  - i. for next
  - ii. while wend
- c. Logical statements:
  - i. if end if
- d. Selective statements: select case end select



# **18.2 Macro Construction**

A Macro is made up of statements. The statements contain constants, variables and operations. The statements are put in a specific order to create the desired output.

A Macro is constructed in the following fashion:

Global Variable Declaration	Optional
Sub Function Block Declarations Local Variable Declarations	Optional
End Sub	
macro_command main() Local Variable Declarations	Required
[Statements]	
end macro_command	Required

Macro must have one and only one main function which is the execution start point of macro. The format is:

macro\_command Function\_Name()

#### end macro\_command

Local variables are used within the main macro function or in a defined function block. Its value remains valid only within the specific block.

Global variables are declared before any function blocks and are valid for all functions in the macro. When local variables and global variables have the same declaration of name, only the local variables are valid.

The example below is a simple Macro which includes a variable declaration and a function call.

```
macro_command main()
short pressure = 10  // local variable declaration
SetData(pressure, "Allen-Bradley DF1", N7, 0, 1)  // function calling
end macro_command
```

# 18.3 Syntax

# **18.3.1 Constants and Variables**

## 18.3.1.1Constants

Constants are fixed values and can be written directly into statements. The format is as below:

Constant Type	Note	Example
Decimal integer		345, -234, 0, 23456
Hexadecimal	Must begin with 0x	0x3b, 0xffff, 0x237
ASCII	String must be enclosed in single	'a', 'data', 'name'
	quotes	
Boolean		true, false

Example of some statements using constants:

macro\_command main() short A, B // A and B are variables A = 1234 B = 0x12 // 1234 and 0x12 are constants end macro\_command

## 18.3.1.2 Variables

Variables are names that represent information. The information can be changed as the variable is modified by statements.


## Naming Rules for Variables

- 1. A variable name must start with an alphabet.
- 2. Variable names longer than 32 characters are not allowed.
- 3. Reserved words cannot be used as Variable names.

There are 8 different Variable types, 5 for signed data types and 3 for unsigned data types:

Variable Type	Description	Range
bool	1 bit (discrete)	0, 1
Char	8 bits (byte)	±127
short	16 bits (word)	±32767
Int	32 bits (double word)	±2147418112
float	32 bits (double word)	
unsigned char	8 bits (byte)	0 to 255
unsigned short	16 bits (word)	0 to 65535
unsigned int	32 bits (double word)	0 to 4,294,967,295

### **Declaring Variables**

Variables must be declared before being used. To declare a variable, specify the type before the variable name.

Example:

int a short b, switch float pressure unsigned short c

## **Declaring Arrays**

Macros support one-dimensional arrays (zero-based index). To declare an array of variables, specify the type and the variable name followed by the number of variables in the array enclosed in brackets "[]". Arrays are 1 to 4096 variables in length. (Macros only support up to 4096 variables per macro).

Example:

int	a[10]
short	b[20], switch[30]
float	pressure[15]



Minimum of array index is 0 and maximum of array index is (array size - 1).

Example:

char data 100] // array size is 100 where: minimum of array index is 0 and maximum of array index is 99 (100 - 1)



### Variable and Array Initialization

There are two ways variables can be initialized:

1. By statement using the assignment operator (=)

```
Example:
```

```
int a
float b[3]
a = 10
b[0] = 1
```

2. During declaration

char a = '5', b = 9

The declaration of arrays is a special case. The entire array can be initialized during declaration by enclosing comma separated values inside curly brackets "{}".

```
Example:
float data[4] = {11, 22, 33, 44} // now data[0] is 11, data[1] is 22....
```

### 18.3.2 Operators

Operations are used to designate how data is to be manipulated. In each statement, the operator on the left is set to the conditions on the right.

Operator	Description	Example
=	Assignment operator	pressure = 10

Arithmetic Operators	Description	Example
+	Addition	A = B + C
-	Subtraction	A = B – C
*	Multiplication	A = B * C
1	Division	A = B / C
%	Modulo division (return	A = B % 5
	remainder)	



Comparison Operators	Description	Example
<	Less than	if A < 10 then B = 5
<=	Less than or equal to	if A <= 10 then B = 5
>	Greater than	if A > 10 then B = 5
>=	Greater than or equal to	if A >= 10 then B = 5
==	Equal to	if A == 10 then B = 5
<b>\$</b>	Not equal to	if A <> 10 then B = 5

Logic Operators	Description	Example
And	Logical AND	if A < 10 and B > 5 then C = 10
Or	Logical OR	if A >= 10 or B > 5 then C = 10
Xor	Logical Exclusive OR	if A xor 256 then B = 5
Not	Logical NOT	if not A then B = 5

Shift and bitwise operators are used to manipulate bits within char, short, and int variable types with both signed and unsigned. The priority of these operators is from left to right within the statement.

Shift Operators	Description	Example
<<	Shifts the bits in a bitset to the	A = B << 8
	left a specified number of	
	positions	
>>	Shifts the bits in a bitset to the	A = B >> 8
	right a specified number of	
	positions	

Bitwise Operators	Description	Example
&	Bitwise AND	A = B & 0xf
	Bitwise OR	A = B   C
^	Bitwise XOR	A = B ^ C
~	One's complement	A = ~B



## Priority of All Operators

The overall priority of all operations from highest to lowest is as follows:

Operations within parenthesis are carried out first Arithmetic operations Shift and Bitwise operations Comparison operations Logic operations Assignment

### **Reserved Keywords**

The following keywords are reserved for Macro use. They cannot be used for variable, array, or function names.

+, -, \*, /, %, >=, >, <=, <, <>, ==, and, or, xor, not, <<, >>,=, &, |, ^, ~ exit, macro command, for, to, down, step, next, return, bool, short, int, char, float, void, if, then, else, break, continue, set, sub, end, while, wend, true, false SQRT, CUBERT, LOG, LOG10, SIN, COS, TAN, COT, SEC, CSC, ASIN, ACOS, ATAN, BIN2BCD, BCD2BIN, DEC2ASCII, FLOAT2ASCII, HEX2ASCII, ASCII2DEC, ASCII2FLOAT, ASCII2HEX, FILL, RAND, DELAY, SWAPB, SWAPW, LOBYTE, HIBYTE, LOWORD, HIWORD, GETBIT, SETBITON, SETBITOFF, INVBIT, ADDSUM, XORSUM, CRC, INPORT, OUTPORT, POW, GetError, GetData, GetDataEx, SetData, SetDataEx, SetRTS, GetCTS, Beep, SYNC TRIG MACRO, ASYNC TRIG MACRO, TRACE, FindDataSamplingDate, FindDataSamplingIndex, FindEventLogDate, FindEventLogIndex StringGet, StringGetEx, StringSet, StringSetEx, StringCopy, StringMid, StringDecAsc2Bin, StringBin2DecAsc, StringDecAsc2Float, StringFloat2DecAsc, StringHexAsc2Bin, StringBin2HexAsc, StringLength, StringCat, StringCompare, StringCompareNoCase, StringFind, StringReverseFind, StringFindOneOf, StringIncluding, StringExcluding, StringToUpper, StringToLower, StringToReverse, StringTrimLeft, StringTrimRight, StringInsert



### 18.4 Statement

### **18.4.1 Definition Statement**

This covers the declaration of variables and arrays. The formal construction is as follows: type name where define the type of name

Example:

int A //define a variable A as an integer

type name[constant] where define the type of array name

Example: int B[10]

size 10

where define a variable B as a one-dimensional array of

## 18.4.2 Assignment Statement

Assignment statements use the assignment operator to move data from the expression on the right side of the operator to the variable on the left side. An expression is the combination of variables, constants and operators to yield a value.

Variable = Expression

Example

A = 2 where a variable A is assigned to 2



## 18.4.3 Logical Statements

Logical statements perform actions depending on the condition of a Boolean expression. The syntax is as follows:

### Single-Line Format

if <Condition> then [Statements] else [Statements] end if

Example:

if a == 2 then b = 1 else b = 2 end if

### **Block Format**

If <Condition> then [Statements] else if <Condition – n> then [Statements] else [Statements] end if

Example: if a == 2 then b = 1 else if a == 3 then b = 2 else b = 3 end if

Syntax description:

if	Must be used to begin the statement
<condition></condition>	Required. This is the controlling statement. It is FALSE when the
	<condition> evaluates to 0 and TRUE when it evaluates to non- zero.</condition>
then	Must precede the statements to execute if the <condition> evaluates to</condition>
	TRUE.
[Statements]	It is optional in block format but necessary in single-line format without
	else. The statement will be executed when the <condition> is TRUE.</condition>
else if	Optional. The else if statement will be executed when the relative
	<condition-n> is TRUE.</condition-n>
<condition-n></condition-n>	Optional. see <condition></condition>
else	Optional. The else statement will be executed when <condition> and</condition>
	<condition-n> are both FALSE.</condition-n>
end if	Must be used to end an if-then statement.

## **18.4.4 Selective Statements**

The select-case construction can be used to perform selective group of actions depending on the value of the given variable. The actions under the matched case are performed until a break command is read. The syntax is as follows.

### Default case free Format

Select Case [variable]		
Case [value]		
[Statements]		
break		
end Select		

Example:

Select Case A



Case 1 b=1 break end Select

### Default case Format

Select Case [variable] Case [value] [Statements] break Case else [Statements] break

#### end Select

#### Example:

Select Case A Case 1 b=1 break Case else b=0 break end Select

### Multiple cases in the same block

Select Case [variable] Case [value1] [Statements] Case [value2] [Statements] break

#### end Select

Example:

Select Case A



Case 1 Case 2 b=2 Case 3 b=3 break end Select

Syntax description:

Select Case	Must be used to begin the statement
[variable]	Required. The value of this variable will be compared to the value of
	each case.
Case else	Optional. It represents the default case. If none of the cases above are
	matched, the statements under default case will be executed. When a
	default case is absent, it will skip directly to the end of the select-case
	statements if there is no matched case.
break	Optional. The statements under the matched case will be executed until
	the break command is reached. If a break command is absent, it simply
	keeps on executing next statement until the end command is reached.
end Select	Indicates the end of the select-case statements

# 18.4.5 Reiterative Statements

Reiterative statements control loops and repetitive tasks depending on condition. There are two types of reiterative statements.

## 18.4.5.1 for-next Statements

The for-next construction is for stepping through a fixed number of iterations. A variable is used as a counter to track progress and test for ending conditions. Use this for fixed execution counts. The syntax is as follows:



for [Counter] = <StartValue> to <EndValue> [step <StepValue>]
 [Statements]
next [Counter]

#### or

for [Counter] = <StartValue> down <EndValue> [step <StepValue>]
 [Statements]
next [Counter]

### Example:

for a = 0 to 10 step 2 b = a

next a

Syntax description:

for	Must be used to begin the statement
[Counter]	Required. This is the controlling statement. The result of evaluating the
	variable is used as a test of comparison.
<startvalue></startvalue>	Required. The initial value of [Counter]
to/down	Required. This determines if the <step> increments or decrements the</step>
	<counter>.</counter>
	"to" increments <counter> by <stepvalue>.</stepvalue></counter>
	"down" decrements <counter> by <stepvalue>.</stepvalue></counter>
<endvalue></endvalue>	Required. The test point. If the <counter> is greater than this value, the</counter>
	macro exits the loop.
step	Optional. Specifies that a <stepvalue> other than one is to be used.</stepvalue>
[StepValue]	Optional. The increment/decrement step of <counter>. It can be omitted</counter>
	when the value is 1 If [step <stepvalue>] are omitted the step value</stepvalue>
	defaults to 1.
[Statements]	Optional. Statements to execute when the evaluation is TRUE. "for-next"
	loops may be nested.
next	Required.
[Counter]	Optional. This is used when nesting for-next loops.



## 18.4.5.2 while-wend Statements

The while-wend construction is for stepping through an unknown number of iterations. A variable is used to test for ending conditions. When the condition is TRUE, the statements are executed repetitively until the condition becomes FALSE. The syntax is as follows.

while <condition></condition>	
[Statements]	
wend	

Example:

```
while a < 10
a = a + 10
wend
```

Syntax description:

while	Must be used to begin the statement
continue	Required. This is the controlling statement. When it is TRUE, the loop
	begins execution. When it is FALSE, the loop terminates.
return [value]	Statements to execute when the evaluation is TRUE.
wend	Indicates the end of the while-end statements

### 18.4.5.3 Other Control Commands

break	Used in for-next and while-wend. It skips immediately to the end of the
	reiterative statement.
continue	Used in for-next and while-wend. It ends the current iteration of a loop
	and starts the next one.
return	The return command inside the main block can force the macro to stop
	anywhere. It skips immediately to the end of the main block.





## **18.5 Function Blocks**

Function blocks are useful for reducing repetitive codes. It must be defined before use and supports any variable and statement type. A function block is called by putting its name followed by parameters, in parenthesis, in the Main Macro Function. After the function block is executed, it returns the value to the Main Function where it is used as an assignment or condition. A return type is not necessary in definition of function, which means that a function block is not always necessary to return a value. The parameters can also be absent in definition of function while the function has no need to take any parameters from the Main Function. The syntax is as follows:

### Definition of function with return type:

```
sub type <name> [(parameters)]
Local variable declarations
[Statements]
[return [value]]
end sub
```

```
Example:

sub int Add(int x, int y)

int result

result = x +y

return result

end sub

macro_command main()

int a = 10, b = 20, sum
```

```
sum = Add(a, b)
```

```
end macro_command
```

### or:

```
sub int Add()
int result, x=10, y=20
result = x +y
return result
end sub
```



macro\_command main()

int sum

sum = Add()

end macro\_command

### Definition of function without return type:

sub <name> [(parameters)]
Local variable declarations
[Statements]
end sub

### Example:

```
sub Add(int x, int y)
```

int result

result = x +y

end sub

```
macro_command main()
```

int a = 10, b = 20

Add(a, b)

end macro\_command

### or:

```
sub Add()
int result, x=10, y=20
result = x +y
end sub
```



```
macro_command main()
```

Add()

end macro\_command

Syntax description:

sub	Must be used to begin the function block
type	Optional. This is the data type of value that the function returns. A
	function block is not always necessary to return a value.
(parameters)	Optional. The parameters hold values that are passed to the function
	by the Main Macro. The passed parameters must have their type
	declared in the parameter field and assigned a variable name.
	For example: sub int MyFunction(int x, int y). x and y would be
	integers passed to the function by the Main Macro. This function is
	called by a statement that looks similar to this: ret = MyFunction(456,
	pressure) where "pressure" must be integer according to the definition
	of function.
	Notice that the calling statement can pass hard coded values or
	variables to the function. After this function is executed, an integer
	values is return to 'ret'.
Local variable	Variables that are used in the function block must be declared first.
declaration	This is in addition to passed parameters. In the above example x and
	y are variables that the function can used. Global variables are also
	available for use in function block.
[Statements]	Statements to execute
[return [value]]	Optional. Used to return a value to the calling statement. The value
	can be a constant or a variable. Return also ends function block
	execution. A function block is not always necessary to return a value,
	but, when the return type is defined in the beginning of the definition of
	function, the return command is needed.
end sub	Must be used to end a function block.



## **18.6 Build-In Function Block**

EasyBuilder8000 has some build-in functions for retrieving and transferring data to the PLC, data management and mathematical functions.

## **18.6.1 Mathematical Functions**

Name	SQRT
Syntax	SQRT(source, result)
Description	Calculate the square root of source into result.
	Source can be a constant or a variable, but result must be a variable.
	Source must be a nonnegative value.
Example	macro_command main()
	float source, result
	SQRT(15, result)
	source = 9.0
	SQRT(source, result)// result is 3.0
	end macro_command

Name	CUBERT
Syntax	CUBERT (source, result)
Description	Calculate the cube root of source into result.
	Source can be a constant or a variable, but result must be a variable.
	Source must be a nonnegative value.
Example	macro_command main()
	float source, result
	CUBERT (27, result) // result is 3.0 source = 27.0 CUBERT(source, result)// result is 3.0



Name	POW
Syntax	POW (source1, source2, result)
Description	Calculate source1 raised to the power of source2.
	Source1 and source2 can be a constant or a variable, but result must be a
	variable.
	Source1 and source2 must be a nonnegative value.
Example	macro_command main()
	float y, result
	y = 0.5
	POW (25, y, result) // result = 5
	end macro_command

Name	SIN
Syntax	SIN(source, result)
Description	Calculate the sine of source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	float source, result
	SIN(90, result)// result is 1
	source = 30
	SIN(source, result)// result is 0.5
	end macro_command

Name	COS
Syntax	COS(source, result)
Description	Calculate the cosine of source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()



float source, result
COS(90, result)// result is 0
source = 60 GetData(source, "Local HMI", LW, 0, 1) COS(source, result)// result is 0.5
end macro_command

Name	TAN
Syntax	TAN(source, result)
Description	Calculate the tangent of source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	float source, result
	TAN(45, result)// result is 1
	source = 60
	TAN(source, result)// result is 1.732
	end macro_command

Name	СОТ
Syntax	COT(source, result)
Description	Calculate the cotangent of source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	float source, result
	COT(45, result)// result is 1
	source = 60
	COT(source, result)// result is 0.5774



Name	SEC	
Syntax	SEC(source, result)	
Description	Calculate the secant of source into result.	
	Source can be a constant or a variable, but result must be a variable.	
Example	macro_command main()	
	float source, result	
	SEC(45, result)// result is 1.414	
	source = 60	
	SEC(source, result)// if source is 60, result is 2	
	end macro_command	

Name	CSC	
Syntax	CSC(source, result)	
Description	Calculate the cosecant of source into result.	
	Source can be a constant or a variable, but result must be a variable.	
Example	macro_command main()	
	float source, result	
	CSC(45, result)// result is 1.414	
	source = 30	
	CSC(source, result)// result is 2	
	end macro_command	

Name	ASIN
Syntax	ASIN(source, result)
Description	Calculate the hyperbolic sine of source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()



float source, result	
ASIN(0.8660, result)//	result is 60
source = 0.5 ASIN(source, result)//	result is 30
end macro_command	

Name	ACOS	
Syntax	ACOS(source, result)	
Description	Calculate the hyperbolic cosine of source into result.	
	Source can be a constant or a variable, but result must be a variable.	
Example	macro_command main()	
	float source, result	
	ACOS(0.8660, result)// result is 30	
	source = 0.5	
	ACOS(source, result)// result is 60	
	end macro_command	

Name	ATAN	
Syntax	ATAN(source, result)	
Description	Calculate the hyperbolic tangent of source into result.	
	Source can be a constant or a variable, but result must be a variable.	
Example	macro_command main()	
	float source, result	
	ATAN(1, result)// result is 45	
	source = 1.732	
	ATAN(source, result)// result is 60	
	end macro_command	



Name	LOG	
Syntax	LOG (source, result)	
Description	Calculates the natural logarithm of a number.	
	Source can be either a variable or a constant.	
	Result must be a variable.	
Example	macro_command main()	
	float source = 100, result	
	LOG (source, result)// result is approximately 4.6052	
	end macro_command	

Name	LOG10	
Syntax	LOG10 (source, result)	
Description	Calculates the base-10 logarithm of a number.	
	Source can be either a variable or a constant.	
	Result must be a variable.	
Example	macro_command main()	
	float source = 100, result	
	LOG10 (source, result)// result is 2	
	end macro_command	

Calculates a random integer saved into result.	
ery	



## 18.6.2 Data Transformation

Name	BIN2BCD	
Syntax	BIN2BCD(source, result)	
Description	Transforms a binary-type value (source) into a BCD-type value (result).	
	Source can be a constant or a variable, but result must be a variable.	
Example	macro_command main()	
	short source, result	
	BIN2BCD(1234, result)// result is 0x1234	
	source = 5678	
	BIN2BCD(source, result)// result is 0x5678	
	end macro_command	

Name	BCD2BIN	
Syntax	BCD2BIN (source, result)	
Description	Transforms a BCD-type value (source) into a binary-type value (result).	
	Source can be a constant or a variable, but result must be a variable.	
Example	macro_command main()	
	short source, result	
	BCD2BIN(0x1234, result)// result is 1234	
	source = 0x5678	
	BCD2BIN(source, result)// result is 5678	
	end macro_command	



EasyBuilder8000 User's Manual

Name	DEC2ASCII
Syntax	DEC2ASCII(source, result[start], len)
Description	Transforms a decimal value (source) into ASCII string saved to an array
	(result).
	len represents the length of the string and the unit of length depends on
	result's type., i.e. if result's type is "char" (the size is byte), the length of the
	string is (byte * len). If result's type is "short" (the size is word), the length
	of the string is (word * len), and so on.
	The first character is put into result[start], the second character is put into
	result[start + 1], and the last character is put into result[start + (len -1)].
	Source and len can be a constant or a variable, but result must be a
	variable. Start must be a constant.
Example	macro_command main()
	short source
	char result1[4]
	short result2[4]
	source = 5678
	DEC2ASCII(source, result1[0], 4)
	<pre>// result1[0] is '5', result1[1] is '6', result1[2] is '7', result1[3] is '8'</pre>
	// the length of the string (result1) is 4 bytes( = 1 * 4)
	DEC2ASCII(source_result2[0]_4)
	// result2[0] is '5' result2[1] is '6' result2[2] is '7' result2[3] is '8'
	// the length of the string (result2) is 8 bytes( = $2 \times 4$ )
	end macro_command

Name	HEX2ASCII
Syntax	HEX2ASCII(source, result[start], len)
Description	Transforms a hexadecimal value (source) into ASCII string saved to an
	array (result).
	len represents the length of the string and the unit of length depends on
	result's type., i.e. if result's type is "char" (the size is byte), the length of the
	string is (byte * len). If result's type is "short" (the size is word), the length
	of the string is (word * len), and so on.
	source and len can be a constant or a variable, but result must be a
	variable. start must be a constant.



Example	macro_command main()
	short source
	char result[4]
	source = 0x5678
	HEX2ASCII (source, result[0], 4)
	<pre>// result[0] is '5', result[1] is '6', result[2] is '7', result[3] is '8'</pre>
	end macro_command

Name	FLOAT2ASCII
Syntax	FLOAT2ASCII (source, result[start], len)
Description	Transforms a floating value (source) into ASCII string saved to an array
	(result).
	len represents the length of the string and the unit of length depends on
	result's type., i.e. if result's type is "char" (the size is byte), the length of the
	string is (byte * len). If result's type is "short" (the size is word), the length
	of the string is (word * len), and so on.
	Source and len can be a constant or a variable, but result must be a
	variable. Start must be a constant.
Example	macro_command main()
	float source
	char result[4]
	source = 56.8
	FLOAT2ASCII (source, result[0], 4)
	<pre>// result[0] is '5', result[1] is '6', result[2] is '.', result[3] is '8'</pre>
	end macro_command

Name	ASCII2DEC
Syntax	ASCII2DEC(source[start], result, len)
Description	Transforms a string (source) into a decimal value saved to a variable
	(result).
	The length of the string is len. The first character of the string is
	source[start].



	Source and len can be a constant or a variable, but result must be a
	variable. Start must be a constant.
Example	macro_command main()
	char source[4]
	short result
	source[0] = '5'
	source[1] = '6'
	source[2] = '7'
	source[3] = '8'
	ASCII2DEC(source[0], result, 4) // result is 5678
	end macro command

Name	ASCII2HEX
Syntax	ASCII2HEX (source[start], result, len)
Description	Transforms a string (source) into a hexadecimal value saved to a variable
	(result).
	The length of the string is len. The first character of the string is
	source[start].
	Source and len can be a constant or a variable, but result must be a
	variable. Start must be a constant.
Example	macro_command main()
	char source[4]
	short result
	source[0] = '5'
	source[1] = '6'
	source[2] = '7'
	source[3] = '8'
	ASCII2HEX (source[0], result, 4) // result is 0x5678
	end macro_command



Name	ASCII2FLOAT
Syntax	ASCII2FLOAT (source[start], result, len)
Description	Transforms a string (source) into a float value saved to a variable (result).
	The length of the string is len. The first character of the string is
	source[start].
	Source and len can be a constant or a variable, but result must be a
	variable. Start must be a constant.
Example	macro_command main()
	char source[4]
	float result
	source[0] = '5'
	source[1] = '6'
	source[2] = '.'
	source[3] = '8'
	ASCII2FLOAT (source[0], result, 4) // result is 56.8
	end macro_command



# 18.6.3 Data Manipulation

Name	FILL
Syntax	FILL(source[start], preset, count)
Description	Sets the first count elements of an array (source) to a specified value
	(preset).
	source and start must be a variable, and preset can be a constant or
	variable.
Example	macro_command main()
	char result[4]
	char preset
	FILL(result[0], 0x30, 4)
	// result[0] is 0x30, result[1] is 0x30, , result[2] is 0x30, , result[3] is 0x30
	preset = 0x31
	FILL(result[0], preset, 2) // result[0] is 0x31, result[1] is 0x31
	end macro_command

Name	SWAPB
Syntax	SWAPB(source, result)
Description	Exchanges the high-byte and low-byte data of a 16-bit source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	short source, result
	SWAPB(0x5678, result)// result is 0x7856
	source = 0x123
	SWAPB(source, result)// result is 0x2301
	end macro_command



Name	SWAPW
Syntax	SWAPW(source, result)
Description	Exchanges the high-word and low-word data of a 32-bit source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	int source, result
	SWAPW (0x12345678, result)// result is 0x56781234
	source = 0x12345
	SWAPW (source, result)// result is 0x23450001
	end macro_command

Name	LOBYTE
Syntax	LOBYTE(source, result)
Description	Retrieves the low byte of a 16-bit source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	short source, result
	LOBYTE(0x1234, result)// result is 0x34
	source = 0x123
	LOBYTE(source, result)// result is 0x23
	end macro_command

Name	HIBYTE
Syntax	HIBYTE(source, result)
Description	Retrieves the high byte of a 16-bit source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	short source, result
	HIBYTE(0x1234, result)// result is 0x12





source = 0x123
HIBYTE(source, result)// result is 0x01
end macro\_command

Name	LOWORD
Syntax	LOWORD(source, result)
Description	Retrieves the low word of a 32-bit source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	int source, result
	LOWORD(0x12345678, result)// result is 0x5678
	source = 0x12345
	LOWORD(source, result)// result is 0x2345
	end macro_command

Name	HIWORD
Syntax	HIWORD(source, result)
Description	Retrieves the high word of a 32-bit source into result.
	Source can be a constant or a variable, but result must be a variable.
Example	macro_command main()
	int source, result
	HIWORD(0x12345678, result)// result is 0x1234
	source = 0x12345
	HIWORD(source, result)// result is 0x0001
	end macro_command



# 18.6.4 Bit Transformation

Name	GETBIT
Syntax	GETBIT(source, result, bit_pos)
Description	Gets the state of designated bit position of a data (source) into result.
	Result's value will be 0 or 1.
	Source and bit_pos can be a constant or a variable, but result must be a
	variable.
Example	macro_command main()
	int source, result
	short bit_pos
	CETRIT/0 regult 2)// regult in 1
	GETBIT(9, result, 5)// Tesult is T
	source = 4
	bit_pos = 2
	GETBIT(source, result, bit_pos)// result is 1
	end macro_command

Name	SETBITON
Syntax	SETBITON(source, result, bit_pos)
Description	Changes the state of designated bit position of a data (source) to 1, and
	put changed data into result.
	Source and bit_pos can be a constant or a variable, but result must be a
	variable.
Example	macro_command main()
	int source, result
	short bit_pos
	SETBITON(1, result, 3)// result is 9
	source = 0
	bit_pos = 2
	SETBITON (source, result, bit_pos)// result is 4



Name	SETBITOFF
Syntax	SETBITOFF(source, result, bit_pos)
Description	Changes the state of designated bit position of a data (source) to 0, and
	put in changed data into result.
	Source and bit_pos can be a constant or a variable, but result must be a
	variable.
Example	macro_command main()
	int source, result
	short bit_pos
	SETBITOFF(9, result, 3)// result is 1
	source = 4
	bit_pos = 2
	SETBITOFF(source, result, bit_pos)// result is 0
	end macro_command

Name	INVBIT
Syntax	INVBIT(source, result, bit_pos)
Description	Inverts the state of designated bit position of a data (source), and put
	changed data into result.
	Source and bit_pos can be a constant or a variable, but result must be a
	variable.
Example	macro_command main()
	int source, result
	short bit_pos
	INVBIT(4, result, 1)// result = 6
	source = 6
	bit_pos = 1
	INVBIT(source, result, bit_pos)// result = 4



# 18.6.5 Communication

Name	DELAY
Syntax	DELAY(time)
Description	Suspends the execution of the current macro for at least the specified
	interval (time). The unit of time is millisecond.
	Time can be a constant or a variable.
Example	macro_command main()
	int time == 500
	DELAY(100)// delay 100 ms
	DELAY(time)// delay 500 ms
	end macro_command

Name	ADDSUM
Syntax	ADDSUM(source[start], result, data_count)
Description	Adds up the elements of an array (source) from source[start] to
	source[start + data_count - 1] to generate a checksum.
	Puts in the checksum into result. Result must be a variable.
	Data_count is the amount of the accumulated elements and can be a
	constant or a variable.
Example	macro_command main()
	char data[5]
	short checksum
	data[0] = 0x1
	data[1] = 0x2
	data[2] = 0x3
	data[3] = 0x4
	data[4] = 0x5



ADDSUM(data[0], checksum, 5)//	checksum is 0xf
end macro_command	

Name	XORSUM
Syntax	XORSUM(source[start], result, data_count)
Description	Uses an exclusion method to calculate the checksum from source[start] to
	source[start + data_count - 1].
	Puts the checksum into result. Result must be a variable.
	Data_count is the amount of the calculated elements of the array and can
	be a constant or a variable.
Example	macro_command main()
	char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5}
	short checksum
	XORSUM(data[0], checksum, 5)// checksum is 0x1
	end macro_command

Name	CRC
Syntax	CRC(source[start], result, data_count)
Description	Calculates 16-bit CRC of the variables from source[start] to source[start +
	count - 1].
	Puts in the 16-bit CRC into result. Result must be a variable.
	Data_count is the amount of the calculated elements of the array and can
	be a constant or a variable.
Example	macro_command main()
	char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5}
	short 16bit_CRC
	CRC(data[0], 16bit_CRC, 5)// 16bit_CRC is 0xbb2a
	end macro_command

Name	OUTPORT



Syntax	OUTPORT(source[start], device_name, data_count)
Description	Sends out the specified data from source[start] to source[start + count -1] to
	PLC via a COM port or the ethernet.
	Device_name is the name of a device defined in the device table and the
	device must be a "Free Protocol"-type device.
	Data_count is the amount of sent data and can be a constant or a variable.
Example	To use an OUTPORT function, a "Free Protocol" device must be created
	first as follows:
	System Parameter Settings
	Font Extended Memory Printer/Backup Server
	Device list :
	No. Name Location Device type Interface I/F Proto
	Local HMI Local HMI Local MT6056i (320 x Disable N/A
	Local Server MODBUS R TU Device Local Free Protocol COM1 (19200, E, 8, 1) RS232
	Device Properties
	Name : MODBUS RTU Device
	Location : Local Settings
	Richman Free Protocol
	V.I.OU, FREE_PROTOCOL.SO
	PLC I/F : RS-232
	COM : COM1 (19200,E,8,1) Settings
	The device is named "MODBUS RTU Device". The port attribute depends
	on the setting of this device. (the current setting is "19200,E, 8, 1")
	Polow is an example of executing an extian of writing single soil (SET ON)
	to a MODBUS device
	macro_command main()
	char command[32]
	short address, checksum



 FILL(command[0], 0, 32)// command initialization

 command[0] = 0x1// station no

 command[1] = 0x5// function code : Write Single Coil

 address = 0

 HIBYTE(address, command[2])

 LOBYTE(address, command[3])

 command[4] = 0xff// force bit on

 command[5] = 0

 CRC(command[0], checksum, 6)

 LOBYTE(checksum, command[6])

 HIBYTE(checksum, command[7])

 // send out a "Write Single Coil" command

 OUTPORT(command[0], "MODBUS RTU Device", 8)

 end macro\_command

Name	INPORT
Syntax	INPORT(read_data[start], device_name, read_count, return_value)
Description	Reads data from a COM port or the ethernet. These data is stored to
	read_data[start]~ read_data[start + read_count - 1].
	device_name is the name of a device defined in the device table and the
	device must be a "Free Protocol"-type device.
	read_count is the required amount of reading and can be a constant or a
	variable.
	If the function is used successfully to get sufficient data, return_value is 1,
	otherwise is 0.
Example	Below is an example of executing an action of reading holding registers of
	a MODBUS device.
	// Read Holding Registers
	macro command main()


```
char command[32], response[32]
short address, checksum
short read_no, return_value, read_data[2]
FILL(command[0], 0, 32)// command initialization
FILL(response[0], 0, 32)
command[0] = 0x1// station no
command[1] = 0x3// function code : Read Holding Registers
address = 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])
read_no = 2// read 2 words (4x_1 and 4x_2)
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])
CRC(command[0], checksum, 6)
LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])
 send out a 'Read Holding Registers" command
\parallel
OUTPORT(command[0], "MODBUS RTU Device", 8)
// read responses for a 'Read Holding Registers' command
INPORT(response[0], "MODBUS RTU Device", 9, return_value)
if return_value > 0 then
 read data[0] = response[4] + (response[3] << 8)// data in 4x_1
 read data[1] = response[6] + (response[5] << 8)// data in 4x 2
  SetData(read data[0], "Local HMI", LW, 100, 2)
end if
end macro_command
```



Name	GetData		
Syntax	GetData(read_data[start], device_name, device_type, address_offset,		
	data_count)		
	or		
Description	GetData(read_data, device_name, device_type, address_offset, 1)		
Description	Receives data from the PLC. Data is stored into read_data[start]~		
	Data count is the amount of received data. In general, read, data is an array		
	but if data count is 1, read data can be an array or an ordinary variable.		
	Below are two methods to read one word data from PLC.		
	macro_command main()		
	short read_data_1[2], read_data_2		
	GetData(read_data_1[0], "FATEK KB Series", RT, 5, 1)		
	GetData(read_data_2, "FATEK KB Series", RT, 5, 1)		
	end macro_command		
	Device_name is the PLC name enclosed in the double quotation marks (") and		
	this name has been defined in the device list of system parameters as follows		
	(see FATEK KB Series):		
	System Parameter Settings		
	Font Extended Memory Printer/Backup Server		
	Device Model General System Setting Security		
	Device list :		
	Local HMI Local HMI Local MT6056i (320 x., Disable N/A N/A		
	Local Ser MODBUS RTU Local Free Protocol COM1 RS 0		
	Remote P FATEK FB Series Remote (IP:210.68.117.2   FATEK FB Series   COM1   RS   1		
	Device_type is the device type and encoding method (binary or BCD) of the		
	PLC data. For example, if device_type is LW_BIN, it means the register is LW		
	and the encounty method is binary. If use bin encounty method, _BIN_Can be ignored		
	If device type is LW BCD, it means the register is LW and the encoding		

method is BCD.

Address\_offset is the address offset in the PLC. For example, GetData(read\_data\_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5.

If address\_offset uses the format – "N#AAAAA", N indicates that PLC's station number is N. AAAAA represents the address offset. This format is used while multiple PLCs or controllers are connected to a single serial port. For example, GetData(read\_data\_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the PLC's station number is 2. If GetData() uses the default station number defined in the device list as follows, it is not necessary to define station number in address\_offset.

PLC type : FATEK FB Series	•
V.1.10, FATEK_FB.so	
PLC I/F : RS-232	PLC default station no. : 2
COM : COM1 (9600,E,7,1)	Settings
Use broadcast command	

The number of registers actually read from depends on both the type of the read\_data variable and the value of the number of data\_count.

type of read_data	data_count	actual number of 16-bit register read
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bool (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2



	int (32-bit)	2	4	
	float (32-bit)	1	2	
	float (32-bit)	2	4	
	When a GetData() is	s executed usin	ng a 32-bit data type (int or float), the	
	function will automation	cally convert the	e data. For example,	
				l
	macro_command ma	in()		
	GetData(f "MODBUS	" 6x 2 1) //1	will contain a floating point value	
	end macro command	1		
Example	macro_command ma	in()		
	bool a			
	bool b[30]			
	short c			
	short d[50]			
	int e			
	int f[10]			
	double g[10]			
	// get the state of LE	32 to the variabl	e a	
	GetData(a, "Local HM	1I", LB, 2, 1)		
	// get 30 states of LI	30 ~ LB29 to th ⊣MI" I B 0 30	e variables b[0] ~ b[29]	
		IIVII, LD, 0, 30		
	// get one word from	LW2 to the vai	iable c	
	GetData(c, "Local HM	1I", LW, 2, 1)		
	//			
	// get 50 words from GetData(d[0] "Local I	I LVVU ~ LVV49 t HMI" I W 0 50	o the variables d[U] ~ d[49]	
		inni, 200, 0, 00	)	
	// get 2 words from I	_W6 ~ LW7 to t	he variable e	
	// note that the type	of e is int		
	GetData(e, "Local HM	1I", LW, 6, 1)		
	// get 20 words (10 i	nteger values)	from LW0 ~ LW19 to variables f[0] ~ f[9]	



// since each integer value occupies 2 words
GetData(f[0], "Local HMI", LW, 0, 10)
// get 2 words from LW2 ~ LW3 to the variable f
GetData(f, "Local HMI", LW, 2, 1)
end macro_command

Name	GetDataEx
Syntax	GetDataEx (read_data[start], device_name, device_type, address_offset,
	data_count)
	or
	GetDataEx (read_data, device_name, device_type, address_offset, 1)
Description	Receives data from the PLC and continue executing next command even if
	no response from this device.
	Descriptions of read_data, device_name, device_type, address_offset and
	data_count are the same as GetData.
Example	macro_command main()
	bool a
	bool b[30]
	short c
	short d[50]
	int e
	int f[10]
	double g[10]
	// get the state of LB2 to the variable a
	GetDataEx (a, "Local HMI", LB, 2, 1)
	// get 30 states of LB0 ~ LB29 to the variables b[0] ~ b[29]
	GetDataEx (b[0], "Local HMI", LB, 0, 30)
	//
	$\beta$ get one word norm $EVVZ$ to the variable c GetDataEv (c. "Local HMI" LW 2, 1)
	// get 50 words from LW0 ~ LW49 to the variables d[0] ~ d[49]



GetDataEx (d[0], "Local HMI", LW, 0, 50)
<ul> <li>// get 2 words from LW6 ~ LW7 to the variable e</li> <li>// note that he type of e is int</li> <li>GetDataEx (e, "Local HMI", LW, 6, 1)</li> </ul>
<ul> <li>// get 20 words (10 integer values) from LW0 ~ LW19 to f[0] ~ f[9]</li> <li>// since each integer value occupies 2 words</li> <li>GetDataEx (f[0], "Local HMI", LW, 0, 10)</li> </ul>
<pre>// get 2 words from LW2 ~ LW3 to the variable f GetDataEx (f, "Local HMI", LW, 2, 1)</pre>
end macro_command

Name	SetData
Syntax	SetData(send_data[start], device_name, device_type, address_offset,
	data_count)
	or
	SetData(send_data, device_name, device_type, address_offset, 1)
Description	Send data to the PLC. Data is defined in send_data[start]~ send_data[start
	+ data_count - 1].
	data_count is the amount of sent data. In general, send_data is an array,
	but if data_count is 1, send_data can be an array or an ordinary variable.
	Below are two methods to send one word data.
	macro_command main()
	short send_data_1[2] = { 5, 6}, send_data_2 = 5
	SetData(send_data_1[0], "FATEK KB Series", RT, 5, 1)
	SetData(send_data_2, "FATEK KB Series", RT, 5, 1)
	end macro_command
	device name is the DLC name enclosed in the device quotation marks (")
	and this name has been defined in the device list of system parameters
	dovice type is the dovice type and encoding method (bingry or PCD) of the
	DLC data For example, if device type and encouring method (binary of BCD) of the
	PLC data. For example, if device_type is LW_BIN, it means the register is



LW and the encoding method is binary. If use BIN encoding method, "\_BIN" can be ignored.

If device\_type is LW\_BCD, it means the register is LW and the encoding method is BCD.

address\_offset is the address offset in the PLC. For example, SetData(read\_data\_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5.

If address\_offset uses the format – "N#AAAAA", N indicates that PLC's station number is N. AAAAA represents the address offset. This format is used while multiple PLCs or controllers are connected to a single serial port. For example, SetData(read\_data\_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the PLC's station number is 2. If SetData () uses the default station number defined in the device list, it is not necessary to define station number in address\_offset.

The number of registers actually sends to depends on both the type of the send\_data variable and the value of the number of data\_count.

type of read_data	data_count	actual number of 16-bit register send
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bool (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4
float (32-bit)	1	2
float (32-bit)	2	4



	function will automatically send int-format or float-format data to the
	device. For example,
	macro command main()
	float $f = 2.6$
	SetData(f "MODBUS" 6x 2 1) // will send a floating point value to the
	device
	and macro, command
Example	macro_command main()
	int i
	hat f
	bool b[30]
	Int $e = 5$
	int f[10]
	for $i = 0$ to 29
	b[i] = true
	next
	for i = 0 to 49
	d[i] = i * 2
	next i
	for i = 0 to 9
	f [i] = i * 3
	next i
	// set the state of LB2
	SetData(a, "Local HMI", LB, 2, 1)
	// set the states of LB0 ~ LB29
	SetData(b[0], "Local HMI", LB, 0, 30)
	// set the value of LW2
	SetData(c, "Local HMI", LW, 2, 1)



// set the values of LW0 ~ LW49
SetData(d[0], "Local HMI", LW, 0, 50)
// set the values of LW6 $\sim$ LW7, note that the type of e is int
SetData(e, "Local HMI", LW, 6, 1)
// set the values of LW0 ~ LW19
// 10 integers equal to 20 words, since each integer value occupies 2
words
SetData(f[0] "Local HMI" $\downarrow$ W 0 10)
and maara command
eno macro_commano

Name	SetDataEx
Syntax	SetDataEx (send_data[start], device_name, device_type, address_offset,
	data_count)
	or
	SetDataEx (send_data, device_name, device_type, address_offset, 1)
Description	Send data to the PLC and continue executing next command even if no
	response from this device.
	Descriptions of send_data, device_name, device_type, address_offset and
	data_count are the same as SetData.
Example	macro_command main()
	int i
	bool a = true
	bool b[30]
	short c = false
	short d[50]
	int e = 5
	int f[10]
	for i = 0 to 29
	b[i] = true
	next i
	for i = 0 to 49



d[i] = i * 2
next i
for i = 0 to 9
f [i] = i * 3
next i
// set the state of LB2
SetDataEx (a, "Local HMI", LB, 2, 1)
// act the states of LDO - LDO
$\gamma$ set the states of LB0 ~ LB29 Set DataEx (b[0] "Lacal HMI" LB 0.30)
// set the value of LW2
SetDataEx (c, "Local HMI", LW, 2, 1)
// set the values of LW0 ~ LW49
SetDataEx (d[0], "Local HMI", LW, 0, 50)
// set the values of LW6 ~ LW7, note that the type of e is int
SetDataEx (e, "Local HMI", LW, 6, 1)
// set the values of LW0 ~ LW19
// 10 integers equal to 20 words, since each integer value occupies 2
SetDataEx (t[U], "Local HIMI", LVV, U, 10)
and maara, command

Name	GetError	
Syntax	GetError (err)	
Description	Get an error code.	
Example	macro_command main()	
	short err	
	char byData[10]	



GetDataEx(byData[0], "MODBUS RTU", 4x, 1, 10)// read 10 bytes // if err is equal to 0, it is successful to execute GetDataEx() GetErr(err)// save an error code to err end macro\_command

Name	PURGE	
Syntax	PURGE (com_port)	
Description	com_port refers to the COM port number which ranges from 1 to 3. It can	
	be either a variable or a constant.	
	This function is used to clear the input and output buffers associated with	
	the COM port.	
Example	macro_command main()	
	int com_port=3	
	PURGE (com_port)	
	PURGE (1)	
	end macro_command	

Name	SetRTS		
Syntax	SetRTS(com_port, source)		
Description	Set RTS state for RS232.		
	com_port refers to the COM port number 1 . It can be either a variable or a		
	constant. Source also can be either a variable or a constant.		
	This command raise RTS signal while the value of source is greater than 0		
	and lower RTS signal while the value of source equals to 0.		
Example	macro_command main()		
	char com_port=1		
	char value=1		
	SetRTS(com_port, value) // raise RTS signal of COM1 while value>0		



SetRTS(1, 0) // lower RTS signal of COM1

## end macro\_command

Name	GetCTS	
Syntax	GetCTS(com_port, result)	
Description	Get CTS state for RS232.	
	com_port refers to the COM port number 1. It can be either a variable or a	
	constant. Result is used for receiving the CTS signal. It must be a variable.	
	This command receives CTS signal and stores the received data in the	
	result variable. When the CTS signal is pulled high, it writes 1 to result,	
	otherwise, it writes 0.	
Example	macro_command main()	
	char com_port=1	
	char result	
	GetCTS(com_port, result) // get CTS signal of COM1	
	GetCTS (1, result) // get CTS signal of COM1	
	end macro_command	

Name	Веер			
Syntax	Beep ()			
Description	Plays beep sound.			
	This command plays a beep sound with frequency of 800 hertz and			
	duration of 30 milliseconds.			
Example	macro_command main()			
	Beep()			
	end macro_command			



## 18.6.6 String Operation Functions

Name	StringGet		
Syntax	StringGet(read_data[start], device_name, device_type, address_offset,		
-	data_count)		
Description	Receives data from the PLC. The String data is stored into read data[start]~		
	read data[start + data count - 1]. read data must be a one-dimensional char		
	array.		
	Data count is the number of received characters, it can be either a constant or		
	a variable.		
	Device_name is the PLC name enclosed in the double quotation marks (") and		
	this name has been defined in the device list of system parameters as follows		
	(see FATEK KB Series):		
	Suntan Barranta Satting		
	System ratameter settings		
	Font Extended Memory Printer/Backup Server		
	Device Model General System Setting Security		
	No Nome Location Device type Interf I/E St		
	Local HMI Local HMI Local MT6056i (320 x Disable N/A N/A		
	Local Ser MODBUS RTU Local Free Protocol COM1 RS 0		
	Remote P FATEK FB Series Remote (IP:210.68.117.2 FATEK FB Series COM1 RS 1		
	Device_type is the device type and encoding method (binary or BCD) of the		
	PLC data. For example, if device_type is LW_BIN, it means the register is LW		
	and the encoding method is binary. If use BIN encoding method, "BIN" can be		
	ignored.		
	If device_type is LW_BCD, it means the register is LW and the encoding		
	method is BCD.		
	Address_offset is the address offset in the PLC.		

For example, StringGet(read_data_1[0], "FATEK KB S represents that the address offset is 5.			3 Series", RT, 5, 1)	
	If address_offset use number is N. AAAAA multiple PLCs or cont StringGet(read_data_ PLC's station numbe defined in the device in address_offset.	s the format – " represents the trollers are conr _1[0], "FATEK k er is 2. If Strir list as follows, it	N#AAAAA", N indic address offset. Th nected to a single se (B Series", RT, 2#5 ngGet() uses the d is not necessary to	ates that PLC's station is format is used while erial port. For example, , 1) represents that the lefault station number o define station number
	PLC type : FATE	K FB Series		•
	V.1.10	), FATEK_FB.so		
	PLC I/F : RS-23	32 🔻	PLC default stat	ion no. : 2
	COM : COM1	(9600,E,7,1)		Settings
	🔲 Use	e broadcast command		
	The number of regis number of data_coun	sters actually re t since that the	ead from depends read_data is restric	on the value of the steed to char array.
	type of read_data	data_count	actual number o read	of 16-bit register
	char (8-bit)	1		1
	char (8-bit)	2		1
	1 WORD register(16-bit) equals to the size of 2 ASCII characters. At to the above table, reading 2 ASCII characters is actually reading the of one 16-bit register.			characters. According y reading the content
Example	macro_command main() char str1[20]			
	<pre>// read 10 words from LW0~LW9 to the variables str1[0] to str1[19] // since that 1 word can store 2 ASCII characters. reading 20 ASCII</pre>			0] to str1[19] ling 20 ASCII



// characters is actually reading 10 words of register
StringGet(str1[0], "Local HMI", LW, 0, 20)
end macro_command

Name	StringGetEx	
Syntax	StringGetEx (read_data[start], device_name, device_type, address_offset,	
	data_count)	
Description	Receives data from the PLC and continue executing next command even if	
	no response from this device.	
	Descriptions of read_data, device_name, device_type, address_offset and	
	data_count are the same as GetData.	
Example	macro_command main()	
	char str1[20]	
	short test=0	
	// macro will continue executing test = 1 even if the MODBUS device is	
	// not responding	
	StringGetEx(str1[0], "MODBUS RTU", 4x, 0, 20)	
	test = 1	
	// macro will not continue executing test = 2 until MODBUS device responds	
	StringGet(str1[0], "MODBUS RTU", 4x, 0, 20)	
	test = 2	
	end macro_command	

Name	StringSet	
Syntax	StringSet(send_data[start], device_name, device_type, address_offset,	
	data_count)	
Description	Send data to the PLC. Data is defined in send_data[start]~ send_data[start	
	+ data_count - 1]. send_data must be a one-dimensional char array.	
	data_count is the number of sent characters, it can be either a constant or a	
	variable.	
	device_name is the PLC name enclosed in the double quotation marks (")	
	and this name has been defined in the device list of system parameters.	



device\_type is the device type and encoding method (binary or BCD) of the PLC data. For example, if device\_type is LW\_BIN, it means the register is LW and the encoding method is binary. If use BIN encoding method, "\_BIN" can be ignored.

If device\_type is LW\_BCD, it means the register is LW and the encoding method is BCD.

address\_offset is the address offset in the PLC.

For example, StringSet(read\_data\_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5.

If address\_offset uses the format – "N#AAAAA", N indicates that PLC's station number is N. AAAAA represents the address offset. This format is used while multiple PLCs or controllers are connected to a single serial port. For example, StringSet(read\_data\_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the PLC's station number is 2. If SetData () uses the default station number defined in the device list, it is not necessary to define station number in address\_offset.

The number of registers actually sends to depends on the value of the number of data\_count, since that send\_data is restricted to char array.

type of read_data	data_count	actual number of 16-bit register send
char (8-bit)	1	1
char (8-bit)	2	1

1 WORD register(16-bit) equals to the size of 2 ASCII characters. According to the above table, sending 2 ASCII characters is actually writing to one 16-bit register. The ASCII characters are stored into the WORD register from low byte to high byte. While using the ASCII display object to display the string data stored in the registers, data\_count must be a multiple of 2 in order to display full string content. For example: macro\_command main() char src1[10]="abcde"

StringSet(src1[0], "Local HMI", LW, 0, 5)



	end macro_command
	The ASCII display object shows:
	abcd
	If data_count is an even number that is greater than or equal to the length of the string, the content of string can be completely shown:
	macro_command main() char src1[10]="abcde"
	StringSet(src1[0], "Local HMI", LW, 0, 6) end macro command
	abcde
Example	macro_command main()
	char str1[10]="abcde"
	// Send 3 words to LW0~LW2
	// Data are being sent until the end of string is reached.
	// Even though the value of data_count is larger than the length of string
	// , the function will automatically stop.
	end macro_command

Name	StringSetEx
Syntax	StringSetEx (send_data[start], device_name, device_type, address_offset,
	data_count)
Description	Send data to the PLC and continue executing next command even if no
	response from this device.
	Descriptions of send_data, device_name, device_type, address_offset and
	data_count are the same as StringSet.
Example	macro_command main()
	char str1[20]="abcde"
	short test=0
	<pre>// macro will continue executing test = 1 even if the MODBUS device is</pre>
	// not responding



StringSetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
test = 1
<pre>// macro will not continue executing test = 2 until MODBUS device responds StringSet(str1[0], "MODBUS RTU", 4x, 0, 20) test = 2</pre>
end macro_command

Name	StringCopy
Syntax	success = StringCopy ("source", destination[start])
	or
	success = StringCopy (source[start], destination[start])
Description	Copy one string to another. This function copies a static string (which is
	enclosed in quotes) or a string that is stored in an array to the destination
	buffer.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	destination[start] must be an one-dimensional char array.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of source string exceeds the max. size of destination
	buffer, it returns false and the content of destination remains the same.
	The success field is optional.
Example	macro_command main()
	char src1[5]="abcde"
	char dest1[5]
	bool success1
	success1 = StringCopy(src1[0], dest1[0])
	// success1=true, dest1="abcde"
	char dest2[5]
	bool success2
	success2 = StringCopy("12345", dest2[0])
	// success2=true, dest2="12345"
	char src3[10]="abcdetghij"
	char dest3[5]



bool success3
success3 = StringCopy(src3[0], dest3[0])
// success3=false, dest3 remains the same.
char src4[10]="abcdefghij"
char dest4[5]
bool success4
success4 = StringCopy(src4[5], dest4[0])
// success4=true, dest4="fghij"
end macro_command

Name	StringDecAsc2Bin
Syntax	success = StringDecAsc2Bin(source[start], destination)
	or
	success = StringDecAsc2Bin("source", destination)
Description	This function converts a decimal string to an integer. It converts the decimal
	string in source parameter into an integer, and stores it in the destination
	variable.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	Destination must be a variable, to store the result of conversion.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the source string contains characters other than '0' to '9', it returns
	false.
	The success field is optional.
Example	macro_command main()
	char src1[5]="12345"
	int result1
	bool success1
	success1 = StringDecAsc2Bin(src1[0], result1)
	// success1=true, result1 is 12345
	char result2
	bool success2
	success2 = StringDecAsc2Bin("32768", result2)
	// success2=true, but the result exceeds the data range of result2



char src3[2]="4b"
char result3
bool success3
success3 = StringDecAsc2Bin (src3[0], result3)
// success3=false, because src3 contains characters other than '0' to '9'
end macro_command

Name	StringBin2DecAsc
Syntax	success = StringBin2DecAsc (source, destination[start])
Description	This function converts an integer to a decimal string. It converts the integer
	in source parameter into a decimal string, and stores it in the destination
	buffer.
	Source can be either a constant or a variable.
	Destination must be an one-dimensional char array, to store the result of
	conversion.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of decimal string after conversion exceeds the size of
	destination buffer, it returns false.
	The success field is optional.
Example	macro_command main()
	int src1 = 2147483647
	char dest1[20]
	bool success1
	success1 = StringBin2DecAsc(src1, dest1[0])
	// success1=true, dest1="2147483647"
	short src2 = 0x3c
	char dest2[20]
	bool success2
	success2 = StringBin2DecAsc(src2, dest2[0])
	// success2=true, dest2="60"
	int src3 = 2147483647
	char dest3[5]
	bool success3



success3 = StringBin2DecAsc(src3, dest3[0])
// success3=false, dest3 remains the same.
end macro_command

Name	StringDecAsc2Float
Syntax	success = StringDecAsc2Float (source[start], destination)
	or
	success = StringDecAsc2Float ("source", destination)
Description	This function converts a decimal string to floats. It converts the decimal
	string in source parameter into float, and stores it in the destination variable.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	Destination must be a variable, to store the result of conversion.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the source string contains characters other than '0' to '9' or '.', it
	returns false.
	The success field is optional.
Example	macro_command main()
	char src1[10]="12.345"
	float result1
	bool success1
	success1 = StringDecAsc2Float(src1[0], result1)
	// success1=true, result1 is 12.345
	float result2
	bool success2
	success2 = StringDecAsc2Float("1.234567890", result2)
	// success2=true, but the result exceeds the data range of result2, which
	// might result in loss of precision
	char src3[2]="4b"
	float result3
	bool success3
	success3 = StringDecAsc2Float(src3[0], result3)
	// success3=false, because src3 contains characters other than '0' to '9' or
	// `.'



end macro\_command

Name	StringFloat2DecAsc
Syntax	success = StringFloat2DecAsc(source, destination[start])
Description	This function converts a float to a decimal string. It converts the float in
	source parameter into a decimal string, and stores it in the destination
	buffer.
	Source can be either a constant or a variable.
	Destination must be an one-dimensional char array, to store the result of
	conversion.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of decimal string after conversion exceeds the size of
	destination buffer, it returns false.
	The success field is optional.
Example	macro_command main()
	float src1 = 1.2345
	char dest1[20]
	bool success1
	success1 = StringFloat2DecAsc(src1, dest1[0])
	// success1=true, dest1=" 1.2345"
	float src2 = 1.23456789
	char dest2 [20]
	bool success2
	success2 = StringFloat2DecAsc(src2, dest2 [0])
	// success2=true, but it might lose precision
	float src3 = 1.2345
	char dest3[5]
	bool success3
	success3 = StringFloat2DecAsc(src3, dest3 [0])
	// success3=false, dest3 remains the same.
	end macro_command

Name	StringHexAsc2Bin



Syntax	success = StringHexAsc2Bin (source[start], destination)
	or
	success = StringHexAsc2Bin ("source", destination)
Description	This function converts a hexadecimal string to binary data. It converts the
	hexadecimal string in source parameter into binary data, and stores it in the
	destination variable.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	Destination must be a variable, to store the result of conversion.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the source string contains characters other than '0' to '9', 'a' to 'f' or
	'A' to 'F', it returns false.
	The success field is optional.
Example	macro_command main()
	char src1[5]="0x3c"
	int result1
	bool success1
	success1 = StringHexAsc2Bin(src1[0], result1)
	// success1=true, result1 is 3c
	short result2
	bool success2
	success2 = StringDecAsc2Bin("1a2b3c4d", result2)
	<pre>// success2=true, result2=3c4d.The result exceeds the data range of</pre>
	// result2
	char src3[2]="4g"
	char result3
	bool success3
	success3 = StringDecAsc2Bin (src3[0], result3)
	// success3=false, because src3 contains characters other than '0' to '9'
	// , 'a' to 'f' or 'A' to 'F'
	end macro_command

Name	StringBin2HexAsc
Syntax	success = StringBin2HexAsc (source, destination[start])



Description	This function converts binary data to a hexadecimal string. It converts the
	binary data in source parameter into a hexadecimal string, and stores it in
	the destination buffer.
	Source can be either a constant or a variable.
	Destination must be an one-dimensional char array, to store the result of
	conversion.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of hexadecimal string after conversion exceeds the size
	of destination buffer, it returns false.
	The success field is optional.
Example	macro_command main()
	int src1 = 20
	char dest1[20]
	bool success1
	success1 = StringBin2HexAsc(src1, dest1[0])
	// success1=true, dest1="14"
	short src2 = 0x3c
	char dest2[20]
	bool success2
	success2 = StringBin2HexAsc(src2, dest2[0])
	// success2=true, dest2="3c"
	int src3 = 0x1a2b3c4d
	char dest3[6]
	bool success3
	success3 = StringBin2HexAsc(src3, dest3[0])
	// success3=false, dest3 remains the same.
	end macro_command

Name	StringMid
Syntax	success = StringMid (source[start], count, destination[start])
	or
	success = StringMid ("string", start, count, destination[start])
Description	Retrieve a character sequence from the specified offset of the source string
	and store it in the destination buffer.



	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]). For source[start], the
	start offset of the substring is specified by the index value. For static source
	string("source"), the second parameter(start) specifies the start offset of the
	substring.
	The count parameter specifies the length of substring being retrieved.
	Destination must be an one-dimensional char array, to store the retrieved
	substring.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of retrieved substring exceeds the size of destination
	buffer, it returns false.
	The success field is optional.
Example	macro_command main()
	char src1[20]="abcdefghijklmnopqrst"
	char dest1[20]
	bool success1
	success1 = StringMid(src1[5], 6, dest1[0])
	// success1=true, dest1="fghijk"
	char src2[20]="abcdefghijklmnopqrst"
	char dest2[5]
	bool success2
	success2 = StringMid(src2[5], 6, dest2[0])
	// success2=false, dest2 remains the same.
	char dest3[20]="12345678901234567890"
	bool success3
	success3 = StringMid("abcdefghijklmnopqrst", 5, 5, dest3[15])
	// success3= true, dest3=" 123456789012345fghij"
	end macro_command

Name	StringLength
Syntax	length = StringLength (source[start])
	or
	length = StringLength ("source")
Description	Obtain the length of a string. It returns the length of source string and stores



it in the length field on the left-hand side of '=' operator.
The source string parameter accepts both static string (in the form:
"source") and char array (in the form: source[start]).
The return value of this function indicates the length of the source string.
macro_command main()
char src1[20]="abcde"
int length1
length1= StringLength(src1[0])
// length1=5
char src2[20]={'a', 'b', 'c', 'd', 'e'}
int length2
length2= StringLength(src2[0])
// length2=20
char src3[20]="abcdefghij"
int length3
length3= StringLength(src3 [2])
// length3=8
end macro_command

Name	StringCat
Syntax	success = StringCat (source[start], destination[start])
	or
	success = StringCat ("source", destination[start])
Description	This function appends source string to destination string. It adds the
	contents of source string to the last of the contents of destination string.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	Destination must be an one-dimensional char array.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of result string after concatenation exceeds the max. size
	of destination buffer, it returns false.
	The success field is optional.
Example	macro_command main()
	char src1[20]="abcdefghij"



char dest1[20]="1234567890"
bool success1
success1= StringCat(src1[0], dest1[0])
<pre>// success1=true, dest1="123456790abcdefghij"</pre>
char dest2 [10]="1234567890"
bool success2
success2= StringCat("abcde", dest2 [0])
// success2=false, dest2 remains the same.
char src3[20]="abcdefghij"
char dest3[20]
bool success3
success3= StringCat(src3[0], dest3[15])
// success3=false, dest3 remains the same.
end macro command

Name	StringCompare
Syntax	ret = StringCompare (str1[start], str2[start])
	ret = StringCompare ("string1", str2[start])
	ret = StringCompare (str1[start], "string2")
	ret = StringCompare ("string1", "string2")
Description	Do a case-sensitive comparison of two strings.
	The two string parameters accept both static string (in the form: "string1")
	and char array (in the form: str1[start]).
	This function returns a Boolean indicating the result of comparison. If two
	strings are identical, it returns true. Otherwise it returns false.
	The ret field is optional.
Example	macro_command main()
	char a1[20]="abcde"
	char b1[20]="ABCDE"
	bool ret1
	ret1= StringCompare(a1[0], b1[0])
	// ret1=false
	char a2[20]="abcde"
	char b2[20]="abcde"



bool ret2
ret2= StringCompare(a2[0], b2[0])
// ret2=true
char a3 [20]="abcde"
char b3[20]="abcdefg"
bool ret3
ret3= StringCompare(a3[0], b3[0])
// ret3=false
end macro_command

Name	StringCompareNoCase
Syntax	ret = StringCompareNoCase(str1[start], str2[start])
	ret = StringCompareNoCase("string1", str2[start])
	ret = StringCompareNoCase(str1[start], "string2")
	ret = StringCompareNoCase("string1", "string2")
Description	Do a case-insensitive comparison of two strings.
	The two string parameters accept both static string (in the form: "string1")
	and char array (in the form: str1[start]).
	This function returns a Boolean indicating the result of comparison. If two
	strings are identical, it returns true. Otherwise it returns false.
	The ret field is optional.
Example	macro_command main()
	char a1[20]="abcde"
	char b1[20]="ABCDE"
	bool ret1
	ret1= StringCompareNoCase(a1[0], b1[0])
	// ret1=true
	char a2[20]="abcde"
	char b2[20]="abcde"
	bool ret2
	ret2= StringCompareNoCase(a2[0], b2[0])
	// ret2=true
	char a3 [20]="abcde"
	char b3[20]="abcdefg"

bool ret3
ret3= StringCompareNoCase(a3[0], b3[0])
// ret3=false
end macro_command

Name	StringFind
Syntax	position = StringFind (source[start], target[start])
	position = StringFind ("source", target[start])
	position = StringFind (source[start], "target")
	position = StringFind ("source", "target")
Description	Return the position of the first occurrence of target string in the source
	string.
	The two string parameters accept both static string (in the form: "source")
	and char array (in the form: source[start]).
	This function returns the zero-based index of the first character of substring
	in the source string that matches the target string. Notice that the entire
	sequence of characters to find must be matched. If there is no matched
	substring, it returns -1.
Example	macro_command main()
	char src1[20]="abcde"
	char target1[20]="cd"
	bool pos1
	pos1= StringFind(src1[0], target1[0])
	// pos1=2
	char target2[20]="ce"
	bool pos2
	pos2= StringFind("abcde", target2[0])
	// pos2=-1
	char src3[20]="abcde"
	bool pos3
	pos3= StringFind(src3[3], "cd")
	// pos3=-1
	end macro_command



Name	StringReverseFind
Syntax	position = StringReverseFind (source[start], target[start])
	position = StringReverseFind ("source", target[start])
	position = StringReverseFind (source[start], "target")
	position = StringReverseFind ("source", "target")
Description	Return the position of the last occurrence of target string in the source
	string.
	The two string parameters accept both static string (in the form: "source")
	and char array (in the form: source[start]).
	This function returns the zero-based index of the first character of substring
	in the source string that matches the target string. Notice that the entire
	sequence of characters to find must be matched. If there exists multiple
	substrings that matches the target string, function will return the position of
	the last matched substring. If there is no matched substring, it returns -1.
Example	macro_command main()
	char src1[20]="abcdeabcde"
	char target1[20]="cd"
	bool pos1
	pos1= StringReverseFind(src1[0], target1[0])
	// pos1=7
	char target2[20]="ce"
	DOOI pos2
	posz= StringReverseFind( abcdeabcde , targetz[0])
	// posz=-1
	char src3[20]="abcdeabcde"
	bool pos3
	pos3= StringReverseFind(src3[6], "ab")
	// pos3=-1
	end macro_command

Name	StringFindOneOf
Syntax	<pre>position = StringFindOneOf (source[start], target[start])</pre>
	<pre>position = StringFindOneOf ("source", target[start])</pre>



	<pre>position = StringFindOneOf (source[start], "target")</pre>
	<pre>position = StringFindOneOf ("source", "target")</pre>
Description	Return the position of the first character in the source string that matches
	any character contained in the target string.
	The two string parameters accept both static string (in the form: "source")
	and char array (in the form: source[start]).
	This function returns the zero-based index of the first character in the
	source string that is also in the target string. If there is no match, it returns
	-1.
Example	macro_command main()
	char src1[20]="abcdeabcde"
	char target1[20]="sdf"
	bool pos1
	pos1= StringFindOneOf(src1[0], target1[0])
	// pos1=3
	char src2[20]="abcdeabcde"
	bool pos2
	pos2= StringFindOneOf(src2[1], "agi")
	// pos2=4
	char target3 [20]="bus"
	bool pos3
	pos3= StringFindOneOf("abcdeabcde", target3[1])
	// pos3=-1
	end macro_command

Name	StringIncluding
Syntax	success = StringIncluding (source[start], set[start], destination[start])
	success = StringIncluding ("source", set[start], destination[start])
	success = StringIncluding (source[start], "set", destination[start])
	success = StringIncluding ("source", "set", destination[start])
Description	Retrieve a substring of the source string that contains characters in the set
	string, beginning with the first character in the source string and ending
	when a character is found in the source string that is not in the target string.
	The source string and set string parameters accept both static string (in the
	form: "source") and char array (in the form: source[start]).



This function returns a Boolean indicating whether the process is
successfully done or not. If successful, it returns true, otherwise it returns
false. If the length of retrieved substring exceeds the size of destination
buffer, it returns false.
macro_command main()
char src1[20]="cabbageabc"
char set1[20]="abc"
char dest1[20]
bool success1
success1 = StringIncluding(src1[0], set1[0], dest1[0])
// success1=true, dest1="cabba"
char src2[20]="gecabba"
char dest2[20]
bool success2
success2 = StringIncluding(src2[0], "abc", dest2[0])
// success2=true, dest2=""
char set3[20]="abc"
char dest3[4]
bool success3
success3 = StringIncluding("cabbage", set3[0], dest3[0])
// success3=false, dest3 remains the same.
end macro_command

Name	StringExcluding
Syntax	success = StringExcluding (source[start], set[start], destination[start])
	success = StringExcluding ("source", set[start], destination[start])
	success = StringExcluding (source[start], "set", destination[start])
	success = StringExcluding ("source", "set", destination[start])
Description	Retrieve a substring of the source string that contains characters that are
	not in the set string, beginning with the first character in the source string
	and ending when a character is found in the source string that is also in the
	target string.
	The source string and set string parameters accept both static string (in the
	form: "source") and char array (in the form: source[start]).
	This function returns a Boolean indicating whether the process is



	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of retrieved substring exceeds the size of destination
	buffer, it returns false.
Example	macro_command main()
	char src1[20]="cabbageabc"
	char set1[20]="ge"
	char dest1[20]
	bool success1
	success1 = StringExcluding(src1[0], set1[0], dest1[0])
	// success1=true, dest1="cabba"
	char src2[20]="cabbage"
	char dest2[20]
	bool success2
	success2 = StringExcluding(src2[0], "abc", dest2[0])
	// success2=true, dest2=""
	char set3[20]="ge"
	char dest3[4]
	bool success3
	success3 = StringExcluding("cabbage", set3[0], dest3[0])
	// success3=false, dest3 remains the same.
	end macro_command

Name	StringToUpper
Syntax	success = StringToUpper (source[start], destination[start])
	success = StringToUpper ("source", destination[start])
Description	Convert all the characters in the source string to uppercase characters and
	store the result in the destination buffer.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of result string after conversion exceeds the size of
	destination buffer, it returns false.
Example	macro_command main()
	char src1[20]="aBcDe"



char dest1[20]
bool success1
success1 = StringToUpper(src1[0], dest1[0])
// success1=true, dest1="ABCDE"
char dest2[4]
bool success2
success2 = StringToUpper("aBcDe", dest2[0])
// success2=false, dest2 remains the same.
end macro command

Name	StringToLower
Syntax	success = StringToLower (source[start], destination[start])
	success = StringToLower ("source", destination[start])
Description	Convert all the characters in the source string to lowercase characters and
	store the result in the destination buffer.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of result string after conversion exceeds the size of
	destination buffer, it returns false.
Example	macro_command main()
	char src1[20]="aBcDe"
	char dest1[20]
	bool success1
	success1 = StringToUpper(src1[0], dest1[0])
	// success1=true, dest1="abcde"
	char dest2[4]
	bool success2
	success2 = StringToUpper("aBcDe", dest2[0])
	// success2=false, dest2 remains the same.
	end macro_command



Name	StringToReverse
Syntax	success = StringToReverse (source[start], destination[start])
	success = StringToReverse ("source", destination[start])
Description	Reverse the characters in the source string and store it in the destination
	buffer.
	The source string parameter accepts both static string (in the form:
	"source") and char array (in the form: source[start]).
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of reversed string exceeds the size of destination buffer,
	it returns false.
Example	macro_command main()
	char src1[20]="abcde"
	char dest1[20]
	bool success1
	success1 = StringToUpper(src1[0], dest1[0])
	// success1=true, dest1="edcba"
	char dest2[4]
	bool success2
	success2 = StringToUpper("abcde", dest2[0])
	// success2=false, dest2 remains the same.
	end macro_command

Name	StringTrimLeft
Syntax	success = StringTrimLeft (source[start], set[start], destination[start])
	success = StringTrimLeft ("source", set[start], destination[start])
	success = StringTrimLeft (source[start], "set", destination[start])
	success = StringTrimLeft ("source", "set", destination[start])
Description	Trim the leading specified characters in the set buffer from the source
	string.
	The source string and set string parameters accept both static string (in the
	form: "source") and char array (in the form: source[start]).
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of trimmed string exceeds the size of destination buffer, it
	returns false.



Example	macro_command main()
	char src1[20]= "# *a*#bc"
	char set1[20]="# *"
	char dest1[20]
	bool success1
	success1 = StringTrimLeft (src1[0], set1[0], dest1[0])
	// success1=true, dest1="a*#bc"
	char set2[20]={'#', ' ', '*'}
	char dest2[4]
	success2 = StringTrimLeft ("# *a*#bc", set2[0], dest2[0])
	// success2=false, dest2 remains the same.
	char src3[20]="abc *#"
	char dest3[20]
	bool success3
	success3 = StringTrimLeft (src3[0], "# *", dest3[0])
	// success3=true, dest3="abc *#"
	end macro_command

Name	StringTrimRight
Syntax	success = StringTrimRight (source[start], set[start], destination[start])
	success = StringTrimRight ("source", set[start], destination[start])
	success = StringTrimRight (source[start], "set", destination[start])
	success = StringTrimRight ("source", "set", destination[start])
Description	Trim the trailing specified characters in the set buffer from the source string.
	The source string and set string parameters accept both static string (in the
	form: "source") and char array (in the form: source[start]).
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of trimmed string exceeds the size of destination buffer, it
	returns false.
Example	macro_command main()
	char src1[20]= "# *a*#bc# * "
	char set1[20]="# *"
	char dest1[20]
	bool success1


success1 = StringTrimRight(src1[0], set1[0], dest1[0])
// success1=true, dest1="# \*a\*#bc"
char set2[20]={'#', ' ', '\*'}
char dest2[20]
success2 = StringTrimRight("# \*a\*#bc", set2[0], dest2[0])
// success2=true, dest2="# \*a\*#bc"
char src3[20]="ab\*\*c \*#"
char dest3[4]
bool success3
success3 = StringTrimRight(src3[0], "# \*", dest3[0])
// success3=false, dest3 remains the same.
end macro\_command

Name	StringInsert
Syntax	success = StringInsert (pos, insert[start], destination[start])
	success = StringInsert (pos, "insert", destination[start])
	success = StringInsert (pos, insert[start], length, destination[start])
	success = StringInsert (pos, "insert", length, destination[start])
Description	Insert a string in a specific location within the destination string content. The
	insert location is specified by the pos parameter.
	The insert string parameter accepts both static string (in the form: "source")
	and char array (in the form: source[start]).
	The number of characters to insert can be specified by the length
	parameter.
	This function returns a Boolean indicating whether the process is
	successfully done or not. If successful, it returns true, otherwise it returns
	false. If the length of string after insertion exceeds the size of destination
	buffer, it returns false.
Example	macro_command main()
	char str1[20]="but the question is"
	char str2[10]=", that is"
	char dest[40]="to be or not to be"
	bool success



success = StringInsert(18, str1[3], 13, dest[0])
<pre>// success=true, dest="to be or not to be the question"</pre>
success = StringInsert(18, str2[0], dest[0])
<pre>// success=true, dest="to be or not to be, that is the question"</pre>
success = StringInsert(0, "Hamlet:", dest[0])
// success=false, dest remains the same.
end macro_command





# 18.6.7 Miscellaneous

Name	SYNC_TRIG_MACRO
Syntax	SYNC_TRIG_MACRO(macro_id)
Description	Trigger the execution of a macro synchronously (use macro_id to
	designate this macro) in a running macro.
	The current macro will pause until the end of execution of this called
	macro.
	macro_id can be a constant or a variable.
Example	macro_command main()
	char ON = 1, OFF = 0
	SetData(ON, "Local HMI", LB, 0, 1)
	SYNC_TRIG_MACRO(5)// call a macro (its ID is 5)
	SetData(OFF, "Local HMI", LB, 0, 1)
	end macro_command

ASYNC_TRIG_MACRO
ASYNC_TRIG_MACRO (macro_id)
Trigger the execution of a macro asynchronously (use macro_id to
designate this macro) in a running macro.
The current macro will continue executing the following instructions after
triggering the designated macro; in other words, the two macros will be
active simultaneously.
macro_id can be a constant or a variable.
macro_command main()
char ON = 1, OFF = 0
SetData(ON, "Local HMI", LB, 0, 1)
ASYNC_TRIG_MACRO(5)// call a macro (its ID is 5)
SetData(OFF, "Local HMI", LB, 0, 1)



end macro\_command

Name	TRACE	
Syntax	TRACE(format, argumer	nt)
Description	Use this function to send	specified string to the EasyDiagnoser. Users can
	print out the current valu	e of variables during run-time of macro for
	debugging.	
	When TRACE encounters the first format specification (if any), it converts	
	the value of the first argument after format and outputs it accordingly.	
	format refers to the format control of output string. A format specification,	
	which consists of optional (in []) and required fields (in bold), has the	
	following form:	
	%[f	ags] [width] [.precision] <b>type</b>
	Each field of the format	specification is described as below:
	flags (optional):	
	-	
	+	
	width (optional):	
	A nonnegative	decimal integer controlling the minimum
	number of char	acters printed.
	precision (optional):	
	A nonnegative	decimal integer which specifies the precision and
	the number of c	haracters to be printed.
	type:	
	C or c : s	specifies a single-byte character.
	d : :	signed decimal integer.
	i : :	signed decimal integer.
	0 : (	unsigned octal integer.
	U : I	unsigned decimal integer.
	X or x : u	unsigned hexadecimal integer.
	Eore :	Signed value having the form.
	[-	- ] <i>d.dddd</i> <b>e</b> [ <i>sign</i> ] <i>ddd</i> where <i>d</i> is a single decimal
	di	git, <i>dddd</i> is one or more decimal digits, <i>ddd</i> is
	ex	cactly three decimal digits, and <i>sign</i> is + or –.
	f :5	Signed value having the form [ – ] <i>dddd.dddd</i> ,
	W	here <i>dddd</i> is one or more decimal digits.



	The length of output string is limited to 256 characters. The extra
	characters will be ignored.
	The argument part is optional. One format specification converts exactly
	one argument.
Example	macro_command main()
	char c1 = 'a'
	short s1 = 32767
	float f1 = 1.234567
	TRACE("The results are") // output: The results are
	TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1)
	// output: c1 = a, s1 = 32767, f1 = 1.234567
	end macro_command

Name	FindDataSamplingDate
Syntax	return_value = FindDataSamplingDate (data_log_number, index, year,
	month, day)
	or
	FindDataSamplingDate (data_log_number, index, year, month, day)
Descripti	A query function for finding the date of specified data sampling file according
on	to the data sampling no. and the file index. The date is stored into "year",
	"month" and "day" respectively in the format of YYYY, MM and DD.
	Data Sampling Object
	No. Description Read address Sample mode Trigger address Clear address Hold addres
	1 Local HMI : L WO Periodical Disable Disable Disable
	Z Local HMI : LWO Periodical Disable Disable Disable
	data sampling no.
	The directory of saved data: [Storage location]\[filename]\yyyymmdd.dtl. The
	data sampling files under the same directory are sorted according to the file
	name and are indexed starting from 0. The most recently saved file has the
	smallest file index number. For example, if there are four data sampling files
	as follows:
	20101210.dtl
	20101230.dtl
	20110110.dtl
	20110111.dtl



	The file index are:
	20101210.dtl -> index is 3
	20101230.dtl -> index is 2
	20110110.dtl -> index is 1
	20110111.dtl -> index is 0
	"return_value" equals to 1 if referred data sampling file is successfully found,
	otherwise it equals to 0.
	"data_log_number" and "index" can be constant or variable. "year", "month",
	"day" and "return_value" must be variable.
	The "return_value" field is optional.
Example	macro_command main()
	short data_log_number = 1, index = 2, year, month, day
	short success
	// if there exists a data sampling file named 20101230.dtl, with data sampling
	// number 1 and file index 2.
	// the result after execution: success == 1, year == 2010, month == 12 and
	//day == 30
	success = FindDataSamplingDate(data_log_number, index, year, month,
	day)
	end macro_command

Name	FindDataSamplingIndex
Syntax	return_value = FindDataSamplingIndex (data_log_number, year, month,
	day, index)
	or
	FindDataSamplingIndex (data_log_number, year, month, day, index)
Descripti	A query function for finding the file index of specified data sampling file
on	according to the data sampling no. and the date. The file index is stored into
	"index". "year", "month" and "day" are in the format of YYYY, MM and DD
	respectively.
	Data Sampling Object
	No. Description Read address Sample mode Trigger address Clear address Hold addres
	1 Local HMI : L WO Periodical Disable Disable Disable
	Z Local HMI : L WO Periodical Disable Disable Disable
	data sampling no.



	The directory of saved data: [Storage location]\[filename]\yyyymmdd.dtl. The
	data sampling files under the same directory are sorted according to the file
	name and are indexed starting from 0. The most recently saved file has the
	smallest file index number. For example, if there are four data sampling files
	as follows:
	20101210.dtl
	20101230.dtl
	20110110.dtl
	20110111.dtl
	The file index are:
	20101210.dtl -> index is 3
	20101230.dtl -> index is 2
	20110110.dtl -> index is 1
	20110111.dtl -> index is 0
	"return_value" equals to 1 if referred data sampling file is successfully found,
	otherwise it equals to 0.
	"data_log_number", "year", "month" and "day" can be constant or variable.
	"index" and "return_value" must be variable.
	The "return_value" field is optional.
Example	macro_command main()
	short data_log_number = 1, year = 2010, month = 12, day = 10, index
	short success
	// if there exists a data sampling file named 20101210.dtl, with data sampling
	// number 1 and file index 2.
	<pre>// the result after execution: success == 1 and index == 2</pre>
	success = FindDataSamplingIndex (data_log_number, year, month, day,
	index)
	end macro_command

Name	FindEventLogDate	
Syntax	return_value = FindEventLogDate (index, year, month, day)	
	or	
	FindEventLogDate (index, year, month, day)	
Description	A query function for finding the date of specified event log file according to	
	file index. The date is stored into "year", "month" and "day" respectively in	
	the format of YYYY, MM and DD.	



	The event log files stored in the designated position (such as HMI memory
	storage or external memory device) are sorted according to the file name
	and are indexed starting from 0. The most recently saved file has the
	smallest file index number. For example, if there are four event log files as
	follows:
	EL_20101210.evt
	EL_20101230.evt
	EL_20110110.evt
	EL_20110111.evt
	The file index are:
	EL_20101210.evt -> index is 3
	EL_20101230.evt -> index is 2
	EL_20110110.evt -> index is 1
	EL_20110111.evt -> index is 0
	"return_value" equals to 1 if referred data sampling file is successfully
	found, otherwise it equals to 0.
	"index" can be constant or variable. "year", "month", "day" and
	"return_value" must be variable.
	The "return_value" field is optional.
Example	macro_command main()
	short index = 1, year, month, day
	short success
	// if there exists an event log file named EL_20101230.evt $^{,}$ with index 1
	// the result after execution: success == 1, year == 2010, month == 12, day
	//== 30
	success = FindEventLogDate (index, year, month, day)
	end macro_command

Name	FindEventLogIndex		
Syntax	return_value = FindEventLogIndex (year, month, day, index)		
	or		
	FindEventLogIndex (year, month, day, index)		
Description	A query function for finding the file index of specified event log file		
	according to date. The file index is stored into "index". "year", "month" and		
	"day" are in the format of YYYY, MM and DD respectively.		
	The event log files stored in the designated position (such as HMI memory		



	storage or external memory device) are sorted according to the file name
	and are indexed starting from 0. The most recently saved file has the
	smallest file index number. For example, if there are four event log files as
	follows:
	EL_20101210.evt
	EL_20101230.evt
	EL_20110110.evt
	EL_20110111.evt
	The file index are:
	EL_20101210.evt -> index is 3
	EL_20101230.evt -> index is 2
	EL_20110110.evt -> index is 1
	EL_20110111.evt -> index is 0
	"return_value" equals to 1 if referred data sampling file is successfully
	found, otherwise it equals to 0.
	"year", "month" and "day" can be constant or variable. "index" and
	"return_value" must be variable.
	The "return_value" field is optional.
Example	macro_command main()
	short year = 2010, month = 12, day = 10, index
	short success
	// if there exists an event log file named EL_20101210.evt, with index 2
	<pre>// the result after execution: success == 1, index == 2</pre>
	success = FindEventLogIndex (year, month, day, index)
	end macro_command





# **18.7 How to Create and Execute a Macro**

# 18.7.1 How to Create a Macro

Macro programming can be divided into some steps as follows,

#### Step 1:

Click on "Macro Manager" icon on the tool bar of EasyBuilder 8000 to open Macro Manager dialogue box as follows.



Macro	×
Macro list :	
Animation	New
change remote 1 win change remote 2 win	Delete
change remote 3 win COS	Edit
INVBIT OnAuto	Сору
OnAutoEnd OnAutoStart	Paste
OnChangeDubai OnChangeWTI	ОК
OnInit OnShowBrent	Cancel
OnShowDubai OnShowWTI	
On Time_500ms On Timer0	
Scaling Scaling	
Scrippin saver	*
Macro under development :	_
	Hab
	пер



On Macro Manager, all macros compiled successfully are displayed in "Macro list", and all macros in developing are displayed in 'Macro under development". The following is a description of the various buttons.

# [New]

Opens a blank "WorkSpace" editor for creating a new macro.

# [Delete]

Deletes the selected macro.

# [Edit]

Opens the "WorkSpace" editor, and loads the selected macro.

# [Copy]

Copies the selected macro into the clipboard.

# [Paste]

Pastes the macro in the clipboard into the list, and creates a new name for the macro.

# Step 2:

Press the "New" button to open a blank "WorkSpace" editor. Every macro has a unique number defined in "Macro ID" edit box, and macro name must exist, otherwise an error will appear while compiling.

WorkSpace			×
Macro ID: 29	Macro name : macro	test	
macro_command main()			
end macro_command			
GET/SET FN Compile		Exit	Help

# Step 3:

WE!NTEK

Design your macro. If it is necessary to use build-in functions (like SetData() or Getdata()), press 'Get/Set FN..." button to open API dialog and select the function and set essential parameters.



API	×
Function name : Ge	tData
Variable 1 Variable type : sh	nort (16-hit)
Variable : da	ata   Array index : 0
Read address PLC name : Lo Device type : LV Address : 0 Address format : ddo BI	ical HMI   V  V  A  A  A  A  A  A  A  A  A  A  A
[Description] Read data from a dev [Usage]	ice.
[Example] char byData[10]	
<	•
	OK Cancel

# Step 4:

WE!NTEK

After the completion of a new macro, press 'Compile" button to compile the macro.

WorkSpace	×
Macro ID: 29 Macro name: macro_test	
macro_command main() short data[4]	
GetData(data[0], "Local HMI", LW, 0, 4)	
end macro_command	
0 error(s)	
GET/SET FN Compile Exit He	lp

If there is no error, press "Exit" button and find that a new macro "macro\_test" exists in "Macro list".

Macro		×
Macro list :		
Animation Auto	<u>^</u>	New
change remote 1 win change remote 2 win		Delete
change remote 3 win COS		Edit
INVBIT macro_test		Сору
OnAutoEnd OnAutoEnd		Paste
OnChangeBrent OnChangeDrinhai		ОК
OnChangeWTI OnInit		Cancel
OnShowBrent OnShowDubai		
OnShowWTI OnTime_500ms		
On Timer0 On Update Chart		
Scaling Scan Alarm	~	
Macro under development :		
		Help



# 18.7.2 Execute a Macro

There are several ways to execute a macro.

- a. With a PLC Control object
  - 1. Open the PLC Control object and set the attribute to "Execute macro program".
  - Select the macro by name. Choose a bit and select a trigger condition to trigger the macro. The macro will continue to be re-triggered as long as the condition is met. In order to guarantee that the macro will run only once, consider latching the trigger bit, and then resetting the trigger condition within the macro.
  - 3. Use a <u>Set Bit</u> or <u>Toggle Switch</u> object to activate the bit.
- b. With a Set Bit or Toggle Switch object
  - 1. On the General tab of the Set Bit or Toggle Switch dialog, select the "Execute Macro" option.
  - 2. Select the macro to execute. The macro will execute one time when the button is activated.
- c. With a Function Key object
  - 1. On the General tab of the Set Bit or Toggle Switch dialog, select the Execute Macro option.
  - 2. Select the macro to execute. The macro will execute one time when the button is activated.



# 18.8 Some Notes about Using the Macro

1. The maximum storage space of local variables in a macro is 4K bytes. So the maximum array size of different variable types are as follows:

chara[4096] bool b[4096] short c[2048] int d[1024] float e[1024]

- 2. A maximum of 256 macros are allowed in an EasyBuilder 8000 project.
- 3. A macro may cause the HMI to lock up. Possible causes are:
  - A macro contains an infinite loop with no PLC communication.
  - The size of an array exceeds the storage space in a macro.
- 4. PLC communication time may cause the macro to execute slower than expected. Also, too many macro instructions may slow down the PLC communication.



# **18.9 Use the Free Protocol to Control a Device**

When EasyBuilder 8000 does not provide an essential driver for communication with a device, Users also can make use of OUTPORT and INPORT to control the device. The data sent with OUTPORT and INPORT must follow the device's communication protocol. The following example explains how to use these two functions to control a MODBUS RTU device.

First, create a new device in the device table. The device type of the new device is set to "Free Protocol" and named with "MODBUS RTU device" as follows:

	Device list :					
	No.	Name	Location	Device type	Interface	
	Local HMI	Local HMI	Local	MT8121T (800 x 600)	Disable	
	Local PLC 1	MODBUS RTU Device	Local	Free Protocol	COM1(19200,E,8,1)	
Device	Properties					
Lo	Name : MOD HM cation : Local	BUS RTU Device	ettings			
Plo	V.1.0 V.1.0 LC I/F : RS-2 COM : COM1	Protocol 0, FREE_PROTOCOL.so 32 1 (19200,E,8,1)			Settings	

The interface of the device (PLC I/F) uses "RS-232" now. If connecting a MODBUS TCP/IP device, the interface must select 'Ethernet". In addition, it is necessary to set correct IP and port number as follows:

PLC type :	Free Protocol	*
	V.1.00, FREE_PROTOCOL.so	
PLC I/F :	Ethernet 💌	
	Use UDP (User Datagram Protocol )	
IP :	192.168.1.103, Port=502	Settings



Suppose that HMI will read the data of  $4x_1$  and  $4x_2$  on the device. First, utilize OUTPORT to send out a read request to the device. The prototype of OUTPORT is:

OUTPORT(command[start], device\_name, cmd\_count)

Since "MODBUS RTU device" is a MODBUS RTU device, the read request must follow MODBUS RTU protocol. The request uses"Reading Holding Registers (0x03)" command to read data. The following picture displays the content of the command. (The items of the station number (byte 0) and the last two bytes (CRC) are ignored).

Request					
-	Function code	1 Byte	0x03		
	Starting Address	2 Bytes	0x0000 to 0xFFFF		
	Quantity of Registers	2 Bytes	1 to 125 (0x7D)		

#### Response

Depending on the protocol, the content of a read command as follows (The total is 8 bytes):

command[0] : station number	(BYTE 0)
command[1] : function code	(BYTE 1)
command[2] : high byte of starting address	(BYTE 2)
command[3] : low byte of starting address	(BYTE 3)
command[4] : high byte of quantity of registers	(BYTE 4)
command[5] : low byte of quantity of registers	(BYTE 5)
command[6] : low byte of 16-bit CRC	(BYTE 6)
command[7] : high byte of 16-bit CRC	(BYTE 7)
So a read request is designed as follows :	

char command[32] short address, checksum

FILL(command[0], 0, 32) // initialize command[0]~command[31] to 0



```
command[0] = 0x1 // station number
command[1] = 0x3 // read holding registers (function code is 0x3)
address = 0// starting address (4x_1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[2])
read_no = 2// the total words of reading is 2 words
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[4])
LOBYTE(read_no, command[5])
CRC(command[0], checksum, 6)// calculate 16-bit CRC
LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])
```

Lastly, use OUPORT to send out this read request to PLC

OUTPORT(command[0], "MODBUS RTU Device", 8)// send request

After sending out the request, use INPORT to get the response from PLC. Depending on the protocol, the content of the response is as follows (the total byte is 9):

command[0] : station number	(BYTE 0)
command[1] : function code	(BYTE 1)
command[2] : byte count	(BYTE 2)
command[3] : high byte of 4x_1	(BYTE 3)
command[4] : low byte of 4x_1	(BYTE 4)
command[5] : high byte of 4x_2	(BYTE 5)
command[6] : high byte of 4x_2	(BYTE 6)
command[7] : low byte of 16-bit CRC	(BYTE 7)
command[8] : high byte of 16-bit CRC	(BYTE 8)

The usage of INPORT is described below:

INPORT(response[0], "MODBUS RTU Device", 9, return\_value)// read response

Where the real read count is restored to the variable return\_value (unit is byte). If return\_value is 0, it means reading fails in executing INPORT.



Depending on the protocol, response[1] must be equal to 0x3, if the response is correct. After getting correct response, calculate the data of  $4x_1$  and  $4x_2$  and put in the data into LW100 and LW101 of HMI.

```
if (return value >0 and response[1] == 0x3) then
  read data[0] = response[4] + (response[3] << 8)// 4x 1
  read data[1] = response[6] + (response[5] << 8)// 4x 2
  SetData(read data[0], "Local HMI", LW, 100, 2)
end if
The complete macro is as follows:
// Read Holding Registers
macro command main()
char command[32], response[32]
short address, checksum
short read no, return value, read data[2], i
FILL(command[0], 0, 32)//
                          initialize command[0]~command[31] to 0
FILL(response[0], 0, 32)
command[0] = 0x1// station number
command[1] = 0x3// read holding registers (function code is 0x3)
address = 0
address = 0// starting address (4x 1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])
read no = 2/ the total words of reading is 2 words
HIBYTE(read no, command[4])
LOBYTE(read no, command[5])
CRC(command[0], checksum, 6)// calculate 16-bit CRC
LOBYTE(checksum, command[6])
```



HIBYTE(checksum, command[7])

```
OUTPORT(command[0], "MODBUS RTU Device", 8 )// send request
INPORT(response[0], "MODBUS RTU Device", 9, return_value)// read response
if (return_value > 0 and response[1] == 0x3) then
read_data[0] = response[4] + (response[3] << 8)// 4x_1
read_data[1] = response[6] + (response[5] << 8)// 4x_2
```

```
SetData(read_data[0], "Local HMI", LW, 100, 2) end if
```

end macro\_command

The following example explains how to design a request to set the status of 0x\_1. The request uses "Write Single Coil(0x5)" command.

#### Request

Function code	1 Byte	0x05
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

#### Response

Function code	1 Byte	0x05
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

#### Error

Error code	1 Byte	0x85
Exception code	1 Byte	01 or 02 or 03 or 04

#### The complete macro is as follows:

// Write Single Coil (ON)

macro\_command main()

char command[32], response[32] short address, checksum short i, return\_value

```
FILL(command[0], 0, 32)// initialize command[0]~ command[31] to 0
FILL(response[0], 0, 32)
```





```
command[0] = 0x1// station number
command[1] = 0x5// function code : write single coil
```

address = 0 HIBYTE(address, command[2]) LOBYTE(address, command[3])

```
command[4] = 0xff// force 0x_1 on
command[5] = 0
```

```
CRC(command[0], checksum, 6)
```

```
LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])
```

```
OUTPORT(command[0], "MODBUS RTU Device", 8)// send request
INPORT(response[0], "MODBUS RTU Device", 8, return_value)// read response
```

end macro\_command



# 18.10 Compiler Error Message

# 1. Error Message Format:

# error c# : error description

(# is the error message number) Example: error C37 : undeclared identifier : i

When there are compile errors, the error description can be referenced by the compiler error message number.

# 2. Error Description

# (C1)syntax error : 'identifier'

There are many possibilities to cause compiler error.

For example:

macro\_command main()

char i, 123xyz // this is an unsupported variable name end macro command

# (C2) 'identifier' used without having been initialized

Macro must define the size of an array during declaration.

For example: macro\_command main() char i int g[i] // i must be a numeric constant end macro\_command

# (C3) redefinition error : 'identifier'

The name of variable and function within its scope must be unique.

For example: macro\_command main() int g[10] , g // error end macro\_command



#### (C4) function name error : 'identifier'

Reserved keywords and constant can not be the name of a function

For example : sub int if() // error

#### (C5) parentheses have not come in pairs

Statement missing "(" or ")"

For example :

macro\_command main ) // missing "("

# (C6) illegal expression without matching 'if'

Missing expression in "if" statement

# (C7) illegal expression (no 'then') without matching 'if'

Missing "then" in "if" statement

(C8) illegal expression (no 'end if') Missing "end if"

# (C9) illegal 'end if' without matching 'if'

Unfinished "If' statement before "End If"

# (C10) illegal 'else'

The format of "if" statement is : if [logic expression] then [ else [if [logic expression] then ] ]

end if

Any format other than this format will cause a compile error.

# (C17) illegal expression (no 'for') without matching 'next'

"for" statement error : missing "for" before "next"

# (C18) illegal variable type (not integer or char)

Should be integer or char variable



#### (C19) variable type error

Missing assign statement

#### (C20) must be keyword 'to' or 'down'

Missing keyword "to" or "down"

(C21) illegal expression (no 'next')

The format of "for" statement is:

for [variable] = [initial value] to [end value] [step]

next [variable]

Any format other than this format will cause a compile error.

#### (C22) 'wend' statement contains no 'while'

"While" statement error : missing "while" before "Wend"

#### (C23) illegal expression without matching 'wend'

The format of "While" statement is :

while [logic expression]

wend

Any format other than this format will cause a compile error.

#### (C24) syntax error : 'break'

"break" statement can only be used in "for", "while" statement.

#### (C25) syntax error : 'continue'

"continue" statement can only be used in "for" statement, or "while" statement.

#### (C26) syntax error

Error in expression.

#### (C27) syntax error

The mismatch of an operation object in expression can cause a compile error.



```
For example :
macro_command main()
```

int a, b for a = 0 to 2 b = 4 + xyz // illegal : xyz is undefined next a end macro command

## (C28) must be 'macro\_command'

There must be 'macro\_command'

#### (C29) must be key word 'sub'

The format of function declaration is:

```
sub [data type] function_name(...)
```

end sub

```
For example::
sub int pow(int exp)
......
end sub
```

Any format other than this format will cause a compile error.

#### (C30) number of parameters is incorrect

Mismatch of the number of parameters

# (C31) parameter type is incorrect

Mismatch of data type of parameter. When a function is called, the data type and the number of parameters should match the declaration of function, otherwise it will cause a compile error.

# (C32) variable is incorrect

The parameters of a function must be equivalent to the arguments passing to a function to avoid compile error.



# (C33) function name : undeclared function

# (C34) expected constant expression

Illegal array index format.

- (C35) invalid array declaration
- (C36) array index error

# (C37) undeclared identifier : i 'identifier'

Any variable or function should be declared before use.

## (C38) un-supported PLC data address

The parameter of GetData(  $\dots$  ), SetData(  $\dots$  ) should be legal PLC address. If the address is illegal, this error message will be shown.

# (C39) 'idenifier' must be integer, char or constant

The format of array is: Declaration: array\_name[constant] (constant is the size of the array) Usage: array\_name[integer, character or constant]

Any format other than this format will cause a compile error.

# (C40) execution syntax should not exist before variable declaration or constant definition

# (C41) float variables cannot be contained in shift calculation

# (C42) function must return a value



## (C43) function should not return a value

- (C44) float variables cannot be contained in calculation
- (C45) PLC address error
- (C46) array size overflow (max. 4k)
- (C47) macro command entry function is not only one

# (C48) macro command entry function must be only one

The only one main entrance of macro is :

macro\_command function\_name( )
end macro\_command

#### (C49) an extended addressee's station number must be between 0 and 255

#### For example :

SetData(bits[0], "PLC 1", LB, 300#123, 100)

// illegal : 300#123 means the station number is 300, but the maximum is 255

#### (C50) an invalid PLC name

PLC name is not defined in the device list of system parameters.

#### (C51) macro command do not control a remote device

A macro can only control a local machine.

For example :

SetData(bits[0], "PLC 1", LB, 300#123, 100)

"PLC 1" is connected with the remote HMI ,so it can not work.



# 18.11 Sample Macro Code

1. "for" statement and other expressions (arithmetic, bitwise shift, logic and comparison)

```
macro_command main()
int a[10], b[10], i
b[0]= (400 + 400 << 2) / 401
b[1]= 22 *2 - 30 % 7
b[2]= 111 >> 2
b[3]= 403 > 9 + 3 >= 9 + 3 < 4 + 3 <= 8 + 8 == 8
b[4]= not 8 + 1 and 2 + 1 or 0 + 1 xor 2
b[5]= 405 and 3 and not 0
b[6]= 8 & 4 + 4 & 4 + 8 | 4 + 8 ^ 4
b[7] = 6 - (\sim 4)
b[8]= 0x11
b[9]= 409
for i = 0 to 4 step 1
    if (a[0] == 400) then
         GetData(a[0],"Device 1", 4x, 0,9)
         GetData(b[0],"Device 1", 4x, 11,10)
end If
next i
end macro command
```

2. "while", "if" and "break" statements

```
macro_command main()

int b[10], i

i = 5

while i == 5 - 20 % 3

GetData(b[1], "Device 1", 4x, 11, 1)

if b[1] == 100 then

break

end if
```



wend end macro\_command

# 3. Global variables and function call

```
char g
sub int fun(int j, int k)
int y
SetData(j, "Local HMI", LB, 14, 1)
GetData(y, "Local HMI", LB, 15, 1)
g = y
return y
end Sub
macro_command main()
int a, b, i
a = 2
b = 3
i = fun(a, b)
SetData(i, "Local HMI", LB, 16, 1)
end macro_command
```

# 4. "if" statement

```
if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 0, 1)
else
    SetData(k[2], "Device 1", 4x, 0, 1)
end if
if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 1, 1)
else if k[2] == 1 then
    SetData(k[3], "Device 1", 4x, 2, 1)
end If
if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 3, 1)
else if k[2] == 2 then
    SetData(k[3], "Device 1", 4x, 4, 1)
else
    SetData(k[4], "Device 1", 4x, 5, 1)
end If
```

end macro\_command

5. "while" and wend" statements

```
macro_command main()

char i = 0

int a[13], b[14], c = 4848

b[0] = 13

while b[0]

a[i] = 20 + i * 10

if a[i] == 120 then

c = 200

break

end if

i = i + 1
```



wend

SetData(c, "Device 1", 4x, 2, 1) end macro\_command

# 6. "break" and "continue" statements

```
macro_command main()
char i = 0
int a[13], b[14], c = 4848
b[0] = 13
while b[0]
    a[i] = 20 + i * 10
    if a[i] == 120 then
    c =200
         i = i + 1
         continue
    end if
    i = i + 1
    if c == 200 then
    SetData(c, "Device 1", 4x, 2, 1)
    break
    end if
wend
end macro_command
```

# 7. Array

```
macro_command main() int a[25], b[25], i
```

b[0] = 13



for i = 0 to b[0] step 1 a[i] = 20 + i \* 10 next i

SetData(a[0], "Device 1", 4x, 0, 13) end macro\_command



# 18.12 Macro TRACE Function

1. TRACE function is added to MACRO, and can be used with EasyDiagnoser, for viewing current content of the variable used.

The following illustrates how to use TRACE function in MACRO.

First of all, add macro\_1 in the project, and in macro\_1 add TRACE ("LW = %d", a). "%d" indicates to display current value of LW in decimal. The content of macro\_1 is as the following:

macro\_command main()

short a

GetData(a, "Local HMI", LW, 0, 1)

a= a + <mark>1</mark>

SetData(a, "Local HMI", LW, 0, 1)

TRACE ("LW0 = %d", a)

end macro\_command

For the detailed usage of TRACE function, please refer to the illustration in the following paragraph.
Macro	
Macro list :	
[ID:1] macro_1	New
WorkSpace	inter
Macro ID : 1	
macro_command main()	
short a	
GetData(a, "Local HMI", LW, 0, 1)	
a= a + 1	
SetData(a, "Local HMI", LW, 0, 1)	
<pre>TRACE("LWO = %d", a)</pre>	
end macro_command	
Macro unde	

WEINTEK

Secondly, add Numeric Display and Function Key objects in window 10 of the project. The settings of these objects are shown below. Function Key object is used to execute macro\_1.





Lastly, compile the completed project and execute Off-line or On-line simulation.

	<u>O</u> bjects	<u>L</u> ibrary <u>T</u> o	ols <u>W</u> indow	Help	
1010	💡 🎀	🎙 🎙 🔟 🗄	8 후 8 🛠 🖉	👳 🙀 🎒 🛄 🖉 📉 📰 🗄 🐗 🕫	Ø
		• A -	4 E E E	Off-line Simulation (Ctrl+T)	
1	후 릐	<u>400</u> -0- to	भ 🗄 🚍	:::::::::::::::::::::::::::::::::::::	<u>i</u> I

When processing simulation on PC, right click and select "Run EasyDiagnoser" in the pop-up menu.



Afterwards, EasyDiagnoser will be started. [Logger] window displays whether EasyDiagnoser is able to connect with the HMI to be watched or not. [Output] window displays the output of the TRACE function. The illustration below shows that EasyDiagnoser succeeds in connecting with HMI.



When EasyDiagnoser is not able to connect with HMI, [Logger] window displays content

as shown below:

WE!NTEK

Logger [11:26:32] Connection established with the targe [11:26:32] Looking for the target HMI	₽ 💌 et HM ▲



The possible reason of not being able to get connection with HMI can be failure in executing simulation on PC. Another reason is that the Port No. used in project for simulation on PC is incorrect (or occupied by system). Please change Port No. as shown, compile project then do simulation again.

Font Extended Memory Printer/Backup Serv					
Device	Model	General	System Setting	Security	
HMI we	del : Miteozo:IIA	MT0070/UAAT6100/AAT	[8100; /800 v 480)	•	
	MIOU/UH/I	MIOO/OHIMIOIOOMI	101001 (000 x 400)	•	
HMI int HMI station	. no : 0		101001 (000 x 400)		

When opening EasyDiagnoser, the Port No. should be set the same as that in project. Only in this way can the communication succeed.

Select HMI	
1 P Name	4
HMI Name:       Default HMI       ▼         Search       192.168.1.103 (Tina-MT         192.168.1.117 (nicolas_         192.168.1.118 (nicolas_         192.168.1.208 (kevin_MI         Search All         * OS 20091002 or later supports	8100i) mt8104×h) 8100i) T8070iH) mi) oey) 7_1200\
Project Port: 8005	OK Exit

The three successive ports of the project port no. are preserved for HMI communication. Take the setting above as example, Port No. is set as 8005, therefore port 8005, 8006 and



8007 will be preserved. In this case when executing simulation on PC, please make sure

that these ports are not occupied by other programs.

### 2. TRACE Syntax List :

Name	TRACE					
Syntax	TRACE(format, argument)					
Description	Use this function to send specified string to the EasyDiagnoser. Users can					
	print out the current value of variables during run-time of macro for					
	debugging.					
	When TRACE encounters the first format specification (if any), it converts					
	the value of the first argument after format and outputs it accordingly.					
	format refers to the format control of output string. A format specification,					
	which consists of optional (in []) and required fields (in bold), has the					
	following form:					
	%[flags] [width] [.precision] <b>type</b>					
	Each field of the format specification is described as below:					
	<i>flags</i> (optional):					
	-					
	+					
	width (optional):					
	A nonnegative decimal integer controlling the minimum					
	number of characters printed.					
	<i>precision</i> (optional):					
	A nonnegative decimal integer which specifies the precision and					
	the number of characters to be printed.					
	type:					
	C or c : specifies a single-byte character.					
	d : signed decimal integer.					
	i : signed decimal integer.					
	o : unsigned octal integer.					
	u : unsigned decimal integer.					
	X or x : unsigned hexadecimal integer.					
	E or e : Signed value having the form.					
	[ – ] <i>d.dddd</i> <b>e</b> [ <i>sign</i> ] <i>ddd</i> where <i>d</i> is a single decimal					
	digit, <i>dddd</i> is one or more decimal digits, <i>ddd</i> is					
	exactly three decimal digits, and <i>sign</i> is + or –.					



	f : Signed value having the form [ – ] <i>dddd.dddd</i> ,
	where <i>dddd</i> is one or more decimal digits.
	The length of output string is limited to 256 characters.
	The <i>argument</i> part is optional.
Example	macro_command main()
	char c1 = 'a'
	short s1 = 32767
	float f1 = 1.234567
	TRACE("The results are") // output: The results are
	TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1)
	// output: c1 = a, s1 = 32767, f1 = 1.234567
	end macro_command

- Newly Added LB9059 disable MACRO TRACE function (when ON)
   When set ON, the output message of TRACE won't be sent to EasyDiagnoser.
- Users can directly execute EasyDiagnoser.exe from Project Manager. In Project Manager, current HMI on line will be listed; users can simply select the HMI to be watched.

Please note that Project Port should be the same as Port No. used in project file.

WEINTEK	EasyBuilder8000 User	's Manual
🏈 Project Manager		
HMI IP, Password	Select HMI	X
Type : MT8000 X Series	1 P Name	Þ
	HMI Name: Default HMI 🗸 192.168.1.103 (Tina-MT8100i)	~
⊙ Ethernet	Search         192.168.1.117 (nicolas_mt8104xh)           192.168.1.118 (nicolas_8100i)	≣
HMI IP :	Search All 192.168.1.208 (kevin_MT8070iH) 192.168.1.211 (MT8070iH - Katte)	
	* OS 20091002 or later supports 192.168.1.226 (susan-hmi) 192.168.1.226 (susan-hmi)	~
Data/Event Log File I		
Utility		Exit
EasyBuilder80		
EasyConverter	EasyAddressViewer	
EasyPrinter	EasyDiagnoser	
Recipe/Extended Mer	mory Editor	
Build Download Data for	CF/ USB Disk	

- 5. Download project to HMI to start operating. When EasyDiagnoser is unable to get connection with the HMI to be watched, it is possible that HMI power is not ON, or Port No. is incorrect. This may cause EasyDiagnoser to connect then disconnect with HMI continuously. Please check if the Port No. in EasyDiagnoser settings is same as that of the project. The way to change it is described before.
- When EasyDiagnoser succeeds in connecting with HMI, simply execute macro\_1,
   [Output] window will then display the output of the TRACE function.









# 18.13 The Usage of String Operation Functions

String operation functions are added to macro which provides users a more convenient way to operate strings. The term "string" means a sequence of ASCII characters, each of which occupies 1 byte. The sequence of characters can be stored into 16-bit registers with least significant byte first. For example, create an ASCII input object and setup as follows:

Jeneral	Data Entry	Security	Shape	Font	1						
D	escription :										
	🔲 Mask		🗌 Use U	INICOD	E	Reve	erse higl	h/low b	yte		
- Read ad	ldress										
F	PLC name :	Local HMI					~	Se	etting		
	Address :	LW		~	)					_	
P	LC name : [	Local HMI									~
Dev	/ice type :	LW									~
	Address :	D			Sys	tem tag		E	User-o	defined	tag
	and the second se										
Addres	s format : D	)DDDD [ran	ge : 0 ~	10799]							

Run simulation and input "abcdef":



The string "abcdef" is stored in LW0~LW2 as follows (LB represents low byte and HB represents high byte):

	HB	LB
LW0	'B'	'A'
LW1	'D'	'C'
LW2	'F'	'E'
LW3		
LW4		
LW5		

The ASCII input object reads 1 word (2 bytes) at a time as described in the previous chapter. Suppose an ASCII input object is set to read 3 words as shown in the above



example, it can actually read at most 6 ASCII characters since that one ASCII character occupies 1 byte.

The functionality of each string operation function is described in the following table:

Function name	Description
StringGet	Read string data from a device.
StringGetEx	Read string data from a device and continue
	executing next command even if no response from
	that device.
StringSet	Write string data to a device.
StringSetEx	Write string data to a device and continue executing
	next command even if no response from that device.
StringCopy	Copy one string to another.
StringMid	Retrieve a substring.
StringDecAsc2Bin	Convert a decimal string to an integer.
StringBin2DecAsc	Convert an integer to a decimal string.
StringDecAsc2Float	Convert a decimal string to floats.
StringFloat2DecAsc	Convert a float to a decimal string.
StringHexAsc2Bin	Convert a hexadecimal string to binary data.
StringBin2HexAsc	Convert binary data into a hexadecimal string.
StringLength	Obtain the length of a string.
StringCat	Append source string to destination string.
StringCompare	Do a case-sensitive comparison of two strings.
StringCompareNoCase	Do a case-insensitive comparison of two strings.
StringFind	Find a substring inside a larger string.
StringReverseFind	Find a substring inside a larger string; starts from the end.
StringFindOneOf	Find the first matching character from a set.
StringIncluding	Extracts a substring that contains only the characters
	in a set.
StringExcluding	Extracts a substring that contains only the characters
	not in a set.
StringToUpper	Convert the characters of a string to uppercase.
StringToLower	Convert the characters of a string to lowercase.
StringToReverse	Reverse the characters of a string.
StringTrimLeft	Trim the leading specified characters in a set from
	the source string.
StringTrimRight	Trim the trailing specified characters in a set from the



	source string.
StringInsert	Insert a string in a specific location within another
	string.

For more detailed information of the above string operation functions, please check out the "Build-In Function Block" section. In order to demonstrate the powerful usage of string operation functions, the following examples will show you step by step how to create executable project files using the new functions; starts from creating a macro, ends in executing simulation.

1. How to read (or write) a string from a device.

Create a new macro:

Масто	
Macro list :	New
	Delete

Edit the content:

ş	ForkSpace	
		Macro ID: 1
	] <b>2</b> C	X 🖻 🛍 🔺 🕻 🕻 🌾
	1 2 3	macro_command main()
	4	char str[20]
	6	<pre>StringGet(str[0], "Local HMI", LW, 0, 20)</pre>
	7	<pre>StringSet(str[0], "Local HMI", LW, 50, 20)</pre>
	8	end macro_command

The first function "StringGet" is used to read a string from LW0~LW19, and store it into the str array. The second function "StringSet" is used to output the content of str array.

Add ASCII Input and Function Key objects in window 10 of the project. The settings of these objects are shown as below. Function Key object is used to execute macro\_1.



	AAAAA <sup>FK_9</sup> GO
	AAAA
Execute macro     Macro	: [ID:001] macro_1
Read address PLC name : Local HMI Address : LW	✓ Setting
Address	
PLC name : Local HMI Device type : LW Address : 0 Address format : DDDDD [range : 0 ~ 1	System tag User-defined tag
	Index register No. of word : 10
Tag Library	
Read address PLC name : Local HMI Address : LW	✓ Setting
Address	
PLC name : Local HMI Device type : LW Address : 50 Address format : DDDDD france : 0 ~ 1	System tag User-defined tag
	Index register





simulation. Follow the steps below to operate the executing project:

Test 1: AB	CDE GO	Step 1: input string
Test 1: AB	CDE GO	Step 2: press "GO" button
Test 1: AB	CDE <b>GO</b> CDE	Step 3: output string

2. Initialization of a string.

Create a new macro and edit the content:

ş	VorkSpace	
		Macro ID : 1
	] <u>၁</u> င	X 🖻 🛍 🔺 🛪 🛪 🌾
	1 2 3	macro_command main()
	4	char str1[20]="abcde"
	5	char str2[20]={'a','b','c','d','e'}
	7	<pre>StringSet(str1[0], "Local HMI", LW, 0, 20)</pre>
	8	<pre>StringSet(str2[0], "Local HMI", LW, 50, 20)</pre>
	10	end macro_command



The data enclosed in double quotation mark ("") is viewed as a string. str1 is initialized as a string while str2 is initialized as a char array. The following snapshot of simulation shows the difference between str1 and str2 using two ASCII input objects.



Macro compiler will add a terminating null character ('\0') at the end of a string. The function "StringSet" will send each character of str1 to registers until a null character is reached. The extra characters following the null character will be ignored even if the data count is set to a larger value than the length of string.

On the contrary, macro compiler will not add a terminating null character ('\0') at the end of a char array. The actual number of characters of str2 being sent to registers depends on the value of data count that is passed to the "StringSet" function.

3. A simple login page.

Create a new macro and edit the content:

W	orkSpace	
		Macro ID : 1 Mac
	] <b>2</b> 2	ž 🖻 🛍 👍 🌤 🌤 🌤
	1	macro_command main()
	2	char name[20]="admin"
	3	char password[20]="123456"
	4	char name_input[20]
	5	char password_input[20]
	6	<pre>char message_success[40]="Success! Access Accepted."</pre>
	7	<pre>char message_fail[40]="Fail! Access Denied."</pre>
	8	char message_clear[40]
	9	bool name_match=false
	10	bool password_match=false
	11	
	12	<pre>StringGet(name_input[0], "Local HMI", LW, 0, 20)</pre>
	13	<pre>StringGet(password_input[0], "Local HMI", LW, 50, 20)</pre>
	14	name_match = StringCompare(name_input[0], name[0])
	15	password_match = StringCompare(password_input[0], password[0])
	16	
	17	<pre>FILL(message_clear[0], 0x20, 40)// FILL with white space</pre>
	18	<pre>StringSet(message_clear[0], "Local HMI", LW, 100, 40)</pre>
	19 [	if(name_match==true and password_match==true) then
	20	StringSet(message_success[0], "Local HMI", LW, 100, 40)
	21	else
	22	StringSet(message_fail[0], "Local HMI", LW, 100, 40)
	23	- end if
	24	end macro_command





The first two "StringGet" functions will read the strings input by users and store them into arrays named name\_input and password\_input separately. Use the function "StringCompare" to check if the input account name and password are matched. If the account name is matched, name\_match is set true; if the password is matched, password\_match is set true. If both name\_match and password\_match are true, output the string "Success! Access Accepted.". Otherwise, output the string "Fail! Access Denied.".

Add ASCII Input and Function Key bojects in window 10 of the project. The settings of these objects are shown as below. Function Key object is used to execute macro\_1.

Object 2 Account Name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA				
AD_0 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	Account Name: Password:	Object 2 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	Object 1	
			AAAAAAA	Object 4

### Object settings:

Object 1: Function k	(ey ==	
⊙Execute macro	Macro : [ID:001] macro_1	~
Object 2: ASCII Inpu	Jt III	

L	WE!NTE	C

- Read address PLC name : Lo Address : Ly	ocal HMI V Setting	
Address		×
PLC name : Device type : Address : Address format :	Local HMI   LW  0  DDDDD [range : 0 ~ 10799]	
	No. of word : 10	
Tag Library	OK Cancel	

# Object 3: ASCII Input

Jeneral Data Entr	y Security Shape Font Profile
Description	
🗹 Mask	Use UNICODE Reverse high/low byte
Read address —	
PLC name	: Local HMI 🛛 Setting
Address	: LW 🔽 50
PLC name : Device type :	Local HMI LW
PLC name : Device type : Address :	Local HMI LW 50 System tag User-defined tag
PLC name : Device type : Address : Address format :	Local HMI LW 50 System tag User-defined tag DDDDD [range : 0 ~ 10799]
PLC name : Device type : Address : Address format :	Local HMI LW 50 System tag User-defined tag DDDDD [range : 0 ~ 10799] Index register
PLC name : Device type : Address : Address format :	Local HMI LW 50 System tag User-defined tag DDDDD [range : 0 ~ 10799] Index register No. of word : 10

Object 4: ASCII Display

WEINTEK	EasyBuilder8000 User's Manual
Read address       PLC name :     Local HMI       Address :     LW	
Address	
PLC name : Local HMI	✓
Device type : LW	✓
Address : 100 System tag User-defin	ed tag
Address format : DDDDD [range : $0 \sim 10799$ ]	
Index register	
No. of word : 20	
Tag Library OK	Cancel

Lastly, compile the completed project and execute Off-line or On-line isimulation. Follow the steps below to operate the executing project:

<u> </u>	Å
	Account Name admin Step 1: input
	Login
! @ #	\$ % ^ & * ( ) BS
Esc a s	d f g h j K l Enter
Caps Z X	c v b n m < > ?
Clear	SPACE + =

WEINTEK	EasyBuilder8000 User's Manual
O * ♣ Account Name: admin Password: *****	Step 2: input
Login 1 2 3 4 5 6 7 8 9 0 BS 1 Q W E R T Y U I O P [ ] Esc A S D F G H J K L Enter Caps Z X C V B N M , . / \ Clear SPACE + - *	
; *	
Account Name: admin Password: ****** Login	Step 3: press "Login" button
Fast Sel	
EasyView	

		EasyBuilder8000 User's Manual
© *	ł	
	Account Name:admin Password: Login	
Fast Sel	Success! Access Accepted.	Login success
	EasyView	
• *	*	
	Account Name: ADMIN Password: ****** Login	
Fast Sel	Fail! Access Denied.	Login fail
	EasyView	



### 18.14 Macro Password Protection

Macro under develop	ment :
ſ	Password
	Password : 111111 (max : 10 characters)
	OK Cancel
Password protect	Set password
*Decompilation cann	ot recover MACROs when checks [Password protect].

On MACRO editing window there's the [Password protect] selection, tick it and click [Set password...] to set a password less than or equals to 10 characters (support ASCII character only, ex. "a\$#\*hFds").

After setting MACRO password, users will have to input correct password when opening MACRO editing window.

Password	X
Password :	(max : 10 characters)
ОК	Cancel

EasyBuilder8000 should be rebooted for typing the password again after 3 incorrect attempts.





[Caution] When MACRO is password protected, decompilation of XOB file will not be able to restore MACRO contents.



# Chapter 19 Set HMI as a MODBUS Server

## **19.1 Setting HMI as MODBUS Device**

Once HMI is set as MODBUS Server, the data of HMI can be read or written via MODBUS protocol.



Refer to the illustration above, it shows HMI is set as MODBUS Server. The HMI, PC or other devices can use MODBUS protocol to read or write the data from HMI via Ethernet or RS232/485 interface. Please follow the steps as below.

### 19.1.1 Creating a MODBUS Server

First of all, add a new device "MODBUS Server" in the **[Device]** tab of **[System Parameter Settings]**. The **[PLC I/F]** can be set to RS232, RS485 2W, RS485 4W, Ethernet.



Font		Extended Memory		Printer/Bac	kup Server
Device	Model	General	Syster	m Setting	Security
Device list :					
No Na	ume Locati	on Device type	Interface	I/E Protoco	Station no
operties					
Name :		or			
	OHMI	<ul><li>● PLC</li></ul>			
Location :	Local	Settings	444		
PLC type :	MODBUS Serv	/er			
	V 1 00 MODD				
	V.1.00, MODB	US_SERVER.SO			
PLC I/F :	RS-485 2W	~		Station no	D. : 1
	RS-232				
	RS-485 2W RS-485 4W				
COM :	Ethernet				Settir
	USB Use broadc.	ast command			
Interv	val of block pa	ck (words) : 5	*		
Max, rea	d-command siz	ze (words) : 120			
B. Santa and a statistic	e-command siz	ze (words) : 120	× .		

If [PLC I/F] is set as [RS232] or [RS485], please fill in [COM Port Settings] also.

PLC type : MODBUS Server	*
V.1.00, MODBUS_SERVER.so	
PLC I/F : RS-232	Station no. : 1
COM : COM1 (9600,E,8,1)	Settings

If [PLC I/F] is set as [Ethernet], the [IP address] is the same as HMI.

For communication, MODBUS Server [Port no.] should be set the same as HMI Port no.



PLC type : MODBUS Server	~
V.1.00, MODBUS_SERVER.so	
PLC I/F : Ethernet	Station no. : 1
Use UDP (User Datagram Protocol )	
IP:Local,Port=8000(=HMI Port)	Settings
Use broadcast command	

Please refer to HMI Port no. to set MODBUS Server Port no. Go to [Model] tab of [System Parameter Settings], the HMI [Port no]. is shown there.

Font	E	xtended Memory	Printer/Bac	kup Server
Device	Model	General	System Setting	Security
HMI mo	del : MT6056T/M	T8056T (320 x 234)		×

After finishing the setting, MODBUS Server will be listed in **[Device]** tab.

You can send MODBUS command to read or write the data from MODBUS Server after downloading the XOB file to HMI.

S	ostem Parama	eter Settings						×
ſ	Font		Extended	l Memory		Printer/Back	kup Server	
	Device	Model		General	Sys	tem Setting	Security	
	Device list :							
	No.	Name	Location	Device type		Interface		I/F
	Local HMI	Local HMI	Local	MT6056T/M1	8056T	Disable		N/.
	Local Server	MODBUS Server	Local	MODBUS Ser	ver	Ethernet(IP=Loc:	al, Port=8000)	TC



# 19.1.2 Read from / Write to MODBUS Server

HMI (the client) can read from / write to another HMI (the server) via MODBUS protocol. Add a new device in the project of client. If client's **[PLC I/F]** is set as **[Ethernet]**, please select"MODBUS TCP/IP" as **[PLC type]** and fill in the correct **[IP]** (the IP of server HMI) and **[Port no.]**.

Device Properties
Name : MODBUS TCP/IP (Ethernet)
Location : Local Settings
PLC type : MODBUS TCP/IP (Ethernet)
V.1.50, MODBUS_TCPIP.so
PLC I/F : Ethernet  PLC default station no. : 1
Use UDP (User Datagram Protocol )
IP: 192.168.1.111, Port=8000 Settings
IP Address Settings
IP address : 192 . 168 . 1 . 111
Port no. : 8000
Timeout (sec) : 1.0 🗸 Turn around delay (ms) : 0
Send ACK delay (ms) : 0 Parameter 1 : 0
Parameter 2 : 0 Parameter 3 : 0
OK Cancel

If the client use **[RS232/485]** interface, the **[PLC type]** must be set as "MODBUS RTU". Please make sure the communication parameter setting is correct.



Name :	MODBUS RTU	J				
		● PLC				
Location :	Local	~	Settings			
PLC type :	MODBUS RT	U				
	V.1.90, MOD	BUS_RTU.	so			
PLC J/F :	RS-485.2W	~	PLC	default station n	0.:1	
COM : DM Port Setting:	COM1 (9600,	E,8,1)			Set	tings.
COM : DM Port Settings COM	COM1 (9600,	E,8,1)		Timeout (sec) :		tings.
COM : DM Port Settings COM Baud rate	COM1 (9600,	E,8,1)	Turn arc	Timeout (sec) : bund delay (ms) :	1.0 0	tings.
COM : DM Port Settings COM Baud rate Data bits	COM1 (9600, COM 1 9600 8 Bits	E,8,1)	Turn ard Send	Timeout (sec) : bund delay (ms) : ACK delay (ms) :	1.0 0	tings.
COM : OM Port Setting COM Baud rate Data bits Parity	COM1 (9600, COM1 (9600, Second 1 Second	E,8,1)	Turn ard Send	Timeout (sec) : ound delay (ms) : ACK delay (ms) : Parameter 1 :	1.0 0 0	tings.
COM : OM Port Setting: COM Baud rate Data bits Parity Stop bits	COM1 (9600, COM 1 9600 8 Bits Even 1 Bit	E,8,1)	Turn arc Send	Timeout (sec) : bund delay (ms) : ACK delay (ms) : Parameter 1 : Parameter 2 :	1.0 0 0 0	tings.

Set and click **[OK]**, a new device"MODBUS RTU" will be listed in the **[Device]** tab.

System Parameter Settings							
ſ	Font		Extend	ed Memory	Printer/Ba	ckup Server	
	Device Model			General System Setting		Security	
Device list :							
	No.	Name	Location	Device type	Interface	I/F Protocol	Sta
	Local HMI	Local HMI	Local	MT6056i (320 x	Disable	N/A	N/A
	Local PLC 1	MODBUS RTU	Local	MODBUS RTU	COM1(9600,E,8,1)	R\$485 2 W	1

In the setting page of each object, there is a "MODBUS RTU" in the **[PLC name]** selection list; you can then select appropriate device type and address.



Read address ——			
PLC name :	MODBUS RTU	Settin	g
Address :	0x 💌	0	
	0x		
	1x		
	3x_Bit		
l	4×_Bit		
-	6×_Bit		
	0×_multi_coils		

Since the server is HMI, the corresponding read and write address are as follows :

reading / writing	0x/1x(1~9999)	to reading / writing LB(0~9998)
reading / writing	3x/4x/5x(1~9999)	to reading / writing LW(0~9998)
reading / writing	3x/4x/5x(10000~75533)	to reading / writing RW(0~65533)



### **19.2 Changing the Station Number of a MODBUS Server in Runtime**

Change the related reserved registers to modify the station number of a MODBUS/ASCII server (HMI).

- [LW-9541] The station number of a MODBUS/ASCII server (COM 1)
- [LW-9542] The station number of a MODBUS/ASCII server (COM 2)
- [LW-9543] The station number of a MODBUS/ASCII server (COM 3)
- [LW-9544] The station number of a MODBUS/ASCII server (Ethernet)



## **19.3 About MODBUS Address Type**

Address types under MODBUS protocol in EB8000 are 0x, 1x, 3x, 4x, 5x, 6x,  $3x_bit$  and  $4x_bit$ .

Modbus RTU function code:

0x	0x01 Read coil	0x05 write single coil
0x_multi_coils	0x01 Read coil	0x0f write multiple coil
1x	0x02 Read discrete input	N/A for write operation
3x	0x04 Read input register	N/A for write operation
4x	0x03 Read holding register	0x10 write multiple register
5x	0x03 Read holding register	0x10
6x	0x03 Read holding register	0x06 write single register
3x_bit	0x04 Read input register	N/A for write operation
4X_bit	0x03 Read holding register	0x10 write multiple register

#### Note:

① Address type "5x" is mapping to Hold Reg. The communication protocol of 5x is almost same as "4x" except "5x"makes double word swap.

If 4x contains following information

Address 1 2 3 4 5 6 ... Data in word 0x1 0x2 0x3 0x4 0x5 0x6 Data 0x20001 0x40003 0x60005

For 5x, it becomes Address 1 2 3 4 5 6 ... Data in word 0x2 0x1 0x4 0x3 0x6 0x5 Data 0x10002 0x30004 0x50006

- ② Address type 6x is limited to data of one word only.
- <sup>(3)</sup>The communication protocol of 3x\_bit and 4x\_bit are the same as 3x and 4x. The difference is that 3x\_bit and 4x\_bit read single bit of the whole data.



# **Chapter 20 How to Connect a Barcode Device**



### 20.1 How to Connect a Barcode Device

Weintek HMI support connecting barcode (USB/COM) device. Please add a new barcode device in [Edit]/ [System Parameter Settings]/ [Device list] first as shown below.

Barcode Device Settings         COM :       COM 1         Baud rate :       9600         Data bits :       8 Bits         Parity :       None         Stop bits :       1 Bit         V       Use a start code         Start code :       255         Terminator         OCR/LF       STX/ETX         None	Click <b>[Settings]</b> and finish <b>[Barcode Device</b> <b>Settings]</b> : The settings are detailed respectively below.	System Parameter Settings	Printer/Backup Server System Setting Security
	Barcode Device Settings COM : COM 1 V Baud rate : 9600 V Data bits : 8 Bits V Parity : None V Stop bits : 1 Bit V O O N	Read byte limit  Read byte limit  Use a start code  Start code : 255  inator  CR/LF O STX/ETX O Other  Ione	Settings

EK	EasyBuilder8000 U
Barcode Device Settings	
COM : COM 1 Baud rate : 9600 Data bits : 8 Bits Parity : None Stop bits : 1 Bit	<ul> <li>Read byte limit</li> <li>10 </li> <li>Use a start code</li> <li>Start code : 255</li> <li>Terminator</li> <li>CR/LF OSTX/ETX Other </li> <li>None</li> </ul>
	OK Cancel

### [COM] \ [Baud rate] \ [Data bits] \ [Parity] \ [Stop bits]

When use COM interface, please set the communication parameters of barcode device accordingly. When USB interface is used, the parameters needn't to be set.

### [Read byte limit]

This function will restrict the number of byte to read in order to prevent barcode device from reading too much data. The range is  $10 \sim 512$ .

For example:

When [Read byte limit] is set to "10", if the data the barcode device should read: "0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37 0x30 0x38 0x33 0x38". (12 bytes)

Only the first 10 bytes will be read in this case. "0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37 0x30 0x38"

### [Use a start code]

With this function, HMI will only view the first data read by barcode device that identifies with start code to be legal input. Otherwise the data read will be ignored. All the data other than start code will be saved in designated address.

For example: if the start code is 255(0xff), and original data read: "0xff 0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37",



The data saved in designated barcode device address will be: "0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37"

### [Terminator]

Terminator means the end of data, when terminator is detected; it stands for the end of data stream.

[CR/LF]	0x0a or 0x0d stands for the end of data.
[STX/ETX]	0x02 or 0x03 stands for the end of data.
[Other]	User can set the terminator manually.
[None]	HMI will save all read data to designated address of barcode device.

After completing all settings described above, a new "Barcode" device will be listed in the **[Device list]**.

Now the barcode device can be selected in **[PLC type]** on the object parameters setting dialogue box. There are 2 types of address:

Address	Address	Description		
type	name	Description		
	FLAG	FLAG 0 indicates the status of data reading. When reading		
Dit		data is complete, the status of FLAG 0 will be changed		
ы		from OFF to ON. It will not return to OFF automatically,		
		users are free to s	et base on actual usage.	
Mord		BARCODE 0	Number of bytes currently read.	
vvord	DARCODE	BARCODE 1 ~ n	Store the data read by barcode device.	

The following is a barcode device setting example, the barcode read is 9421007480830. BARCODE 0 is the address of "Numeric Display" object (bytes) and BARCODE 1~n is the





device corresponding address are listed below:

Barcode	Data			
corresponding address				
	13 bytes (decimal)			
	The data saved in this address is 14 bytes = 7 words. If			
BARCODE U	the number of byte is odd, system will add a byte (0x0			
	to make it even.			
BARCODE 1	3439HEX			
BARCODE 2	3132HEX			
BARCODE 3	3030HEX			
BARCODE 4	3437HEX			
BARCODE 5	3038HEX			
BARCODE 6	3338HEX			
BARCODE 7	0030HEX			
BARCODE 8	empty			

USB barcode interface does not support on-line simulation.
 HMI now only supports barcode device to connect with one USB interface. When Device Table of project includes this kind of device, keyboard will be detected as barcode device, and LB-9064 will be set to ON automatically when power on. For restoring keyboard to normal function and to pause using barcode device, set LB-9064 to OFF. For restoring barcode device, simply set LB-9064 to ON.



Please confirm your Internet connection before downloading the demo project.



# **Chapter 21 Ethernet Communication and Multi-HMI Connection**

Through Ethernet network, EB8000 provides following methods for data transmission:

- 1. HMI to HMI communication.
- 2. PC to HMI communication.
- 3. Operating the PLC connected to other HMI.

There are two ways of Ethernet communication: one is to use RJ45 straight through cable with hubs. Another is to use RJ45 crossover cable with no hubs, but this is limited to the condition of point to point connection (HMI to HMI, or PC to HMI). The following illustrates ways of how to set up and perform the Ethernet connection.





### 21.1 HMI to HMI Communication



Different HMI can monitor and control each other's data through Ethernet network. With system reserved register (LB and LW); one HMI can master performance of other HMI(s). One HMI can handle requests from a maximum of other 32 HMI simultaneously.

Here is an example of communicating two HMI (HMI A and HMI B). When HMI A wants to use a **[set bit]** object to control the address [LB123] of HMI B, the procedure for setting the Project files (MTP) of HMI A is as follows:

### Step 1

Set the IP address of the two HMI (Refer to the related chapter for details). Suppose the IP address of HMI A and HMI B are set as "192.168.1.1" and "192.168.1.2" respectively.

#### Step 2

Run the project of HMI A in EB8000, under [Device] tab in [System Parameter Settings] menu, add the [IP address] and [Port number] of HMI B as below.

System Parameter Settings								
Font Extended Memory Printer/Backup Server								
	Device Mo		lel	General System Setting		g 🛛	Security	
Device list :								
	No.	No. Name Location Device t Interf I/F Pr St		Staf				
	Local HMI	Local HMI	fI Local MT6056i Disable N/A		N/A	N/A		
	Remote HMI 1	HMI B	Remote(IP:192.168.1.2, Port=8000) MT8xxx Ethemet TCP/		TCP/IP	N/A		

L	WE!NTEK

Device Properties	
Name :	HMI B
Location :	Remote Settings IP : 192.168.1.2 (Port = 8000)
<u> </u>	P Address Settings
	IP address : 192 . 168 . 1 . 2
	Port no. : 8000
	OK Cancel

# Step 3

Select "HMI B" for **[PLC name]** of **[New Set Bit Object]** dialogue, and now HMI A can operate the content of the LB of HMI B.

ew Set Bit Object	×
General Security Shape Label	
Description :	
	5
PLC name : HMI B Setting	וכ
Address : LB 💙 123	
Write after button is released	
_ Attribute	51
Set style : Set ON	~


### 21.2 PC to HMI Communication



With Simulation Function of EB8000, PC can catch data of HMI through Ethernet network and save the data files in computer.

PC can master HMI by operating the system reserved register (LB and LW) of HMI. On the contrary, HMI can also directly control operation of PC, for example, asking PC to save data from HMI or PLC.

The number of HMI mastered by PC is unlimited.

Suppose PC is going to communicate with two HMI (HMI A and HMI B), the procedure for setting PC's MTP projects is as follows:

#### Step 1

Set the IP address of the two HMI (Refer to the related chapter for details). Suppose that the **[IP address]** of HMI A and HMI B are set as "192.168.1.1" and "192.168.1.2" respectively.

#### Step 2

Run the project of PC in EB8000, under **[Device]** tab in **[System Parameter Settings]** menu, add the **[IP address]** and **[Port number]** of HMI A and HMI B as below.

rom		Ext	Extended Memory Printer/Backup Server		ate 0			
Device	Model General System Setting		Security					
Device list :								AL HA
No.	Name	Location			Device type	Interface	1/1	ge 1
Local HMI I	Local HMI	Local			MT6070iH/MT8070	Disable	N.	2 🛠
Remote HMI 1	HMI B	Remote (II	2:192.168.1.2, 1	Port=8000)	MT8xxx	Ethernet	T	
Remote HMI 2 I	HMI A	Remote (II	2:192.168.1.1, 1	Port=8000)	MT8xxx	Ethernet	T	
vice Properties	s Jame : 🔳	MI A						
	0	HMI	OPLC					





#### Step 3

Select correct PLC for **[PLC name]** In **[General]** tab of **[Set Bit Object's Attributes]**. If LB of HMI A is to be controlled, "HMI A" must be selected as below.

Set Bit Object's Properties	×
General Security Shape Label Profile	
Description :	
Write address	
PLC name : HMI A Setting	
Address : LB 💙 123	
Write after button is released	
Attribute	
Set style : Set ON	

#### Step 4

Use HMI MTP projects on PC and perform simulation (either online mode or offline mode), all data of HMIA and B can be controlled by PC.

It is also available for HMI to control data of PC, simply considering PC another HMI. Add PC as a new Remote HMI device to the MTP projects of HMI A or HMI B and set the IP address pointing to the PC.



### 21.3 Operate the PLC Connected with other HMI.



Through Ethernet network, PC or HMI can also operate PLC that is connected to other HMI; for example, suppose there is a Mitsubishi PLC connected to COM 1of HMI B. When PC or HMI A wants to read data from this PLC, the procedure for setting PC or HMI A MTP projects is as follows:

#### Step 1

Set the [IP address] of HMI B; suppose the IP address of HMI B is set as"192.168.1.2".

#### Step 2

Run project of PC or HMI A and add a Remote PLC device (defined as Mitsubishi FX0n\_FX2 in the example below) in **[Device]** tab in **[System Parameter Settings]** menu, then set the correct communication parameters.

Device Properties	
Name : p	PLC on HMI B
C	⊖HMI ⊙PLC
Location :	Remote Settings IP : 192.168.1.2 (Port = 8000)
PLC type :	MITSUBISHI FX0n/FX2
Ņ	/.1.10, MITSUBISHI_FXON.so
PLC I/F :	RS-485 4W V PLC default station no. : 0
сом : б	IOM1 Settings



Since this device is a remote PLC connected with Remote HMI B, the **[IP address]** should be the same as HMI B (192.168.1.2)

#### Step 3

In using the **[set bit]** object to operate the Mitsubishi PLC connected to HMI B, just need to select "PLC on HMI B" for **[PLC name]**, then it is able to operate the PLC connected to the remote HMI B on PC through the simulation function.

Set Bit Object's Proj	perties	×
General Security S	Shape Label Profile	_
Description :		
- Write address		
PLC name :	PLC on HMI B Setting	
Address :	X 💙 123	
	Write after button is released	
Attribute		51
Set style :	Set ON 🛩	



# **Chapter 22 System Reserved Words / Bits**

Some Local Words and Local Bits are reserved for system usage. These registers are all with different functions described below:

	Address lag name	PLC name	Address type	Address	Read/Write
1	LB-9000 : initialized as ON	Local HMI	Bit	LB-9000	Read/Write
2	LB-9001 : initialized as ON	Local HMI	Bit	LB-9001	Read/Write
3	LB-9002 : initialized as ON	Local HMI	Bit	LB-9002	Read/Write
1	LB-9003 : initialized as ON	Local HMI	Bit	LB-9003	Read/Write
5	LB-9004 : initialized as ON	Local HMI	Bit	LB-9004	Read/Write
5	LB-9005 : initialized as ON	Local HMI	Bit	LB-9005	Read/Write
7	LB-9006 : initialized as ON	Local HMI	Bit	LB-9006	Read/Write
3	LB-9007 : initialized as ON	Local HMI	Bit	LB-9007	Read/Write
)	LB-9008 : initialized as ON	Local HMI	Bit	LB-9008	Read/Write
10	LB-9009 : initialized as ON	Local HMI	Bit	LB-9009	Read/Write
1	LB-9010 : data download indicator	Local HMI	Bit	LB-9010	Read only
2	LB-9011 : data upload indicator	Local HMI	Bit	LB-9011	Read only
13	LB-9012 : data download/upload indicator	Local HMI	Bit	LB-9012	Read only
.4	LB-9013 : FS window control[hide(ON)/show(OFF)]	Local HMI	Bit	LB-9013	Read/Write
15	LB-9014 : FS button control[hide(ON)/show(OFF)]	Local HMI	Bit	LB-9014	Read/Write
16	LB-9015 : FS window/button control[hide(ON)/show(OFF)]	Local HMI	Bit	LB-9015	Read/Write
17	LB-9016 : status is on when a client connects to this HMI	Local HMI	Bit	LB-9016	Read/Write
.8	LB-9017 : disable write-back in PLC control's [change window]	Local HMI	Bit	LB-9017	Read/Write
٩	I R-0018 · disable mouse curear (set ON)	Local HMT	Rit	I R-0018	Road/M/rite
Users	can import MT500 tag to represent the address.	1			P



# 22.1 The Address Ranges of Local HMI Memory

## 22.1.1 Bits

Memory	Device Type	Range	Format
Local Memory	LB	0 ~ 12079	DDDDD
Bits			
Local Word Bits	LW_BIT	0 ~ 1079915	DDDDdd
			DDDDD: address
			dd: bit no. (00 ~ 15)
Retentive	RBI	0 ~ 65535f	DDDDDh
Memory Bit			DDDDD: address
Index			h: bit no. (0 ~ f)
			Example:
			567 <u>a</u>
			RW_Bit address = 567 +
			[LW-9000]
			bit offset = a
Retentive	RW_Bit	0 ~ 524287f	DDDDDh
Memory Word			DDDDD: address
Bits			h: bit no. (0 ~ f)
Retentive	RW_A_Bit	0 ~ 65535f	DDDDh
Memory A Word			DDDDD: address
Bits			h: bit no. (0 ~ f)



### 22.1.2 Words

Memory	Device Type	Range	Format
Local Memory	LW	0 ~ 10799	DDDDD
Words			
Retentive	RW	0 ~ 524287	DDDDD
Memory Words			
Retentive	RWI	0 ~ 65535	DDDDD
Memory Word			
Index			Example:
			567
			RW address = 567 +
			[LW-9000]
Retentive	RW_A	0 ~ 65535	DDDDD
Memory A Word			
Extended	EM0 ~ EM9	0~	DDDDDDDDD
Memory Words		4294901760	Limited by device, max.
			2G



### 22.2 HMI Time

		Read/Write Control			
Address	Description	Local	MACRO	Remote	
		нмі	WACKU	нмі	
LW-9010	(16bit-BCD) : local second	R/W	R/W	R/W	
LW-9011	(16bit-BCD) : local minute	R/W	R/W	R/W	
LW-9012	(16bit-BCD) : local hour	R/W	R/W	R/W	
LW-9013	(16bit-BCD) : local day	R/W	R/W	R/W	
LW-9014	(16bit-BCD) : local month	R/W	R/W	R/W	
LW-9015	(16bit-BCD) : local year	R/W	R/W	R/W	
LW-9016	(16bit-BCD) : local week	R	R	R	
LW-9017	(16bit) : local second	R/W	R/W	R/W	
LW-9018	(16bit) : local minute	R/W	R/W	R/W	
LW-9019	(16bit) : local hour	R/W	R/W	R/W	
LW-9020	(16bit) : local day	R/W	R/W	R/W	
LW-9021	(16bit) : local month	R/W	R/W	R/W	
LW-9022	(16bit) : local year	R/W	R/W	R/W	
LW-9023	(16bit) : local week	R	R	R	
LW-9030	(32bit) : system time (unit : 0.1 second)	R	R	R	



### 22.3 User Name and Password

		Read/Write Control			
Address	Description	Local HMI	MACRO	Remote HMI	
LB-9050	user logout	W	W	W	
LB-9060	password error	R	R	R	
LB-9061	update password (set ON)	W	W	W	
LW-9219	(16bit) : user no. (1~12)	R/W	R/W	R/W	
LW-9220	(32bit) : password	R/W	R/W	R/W	
LW-9222	(16bit) : classes can be operated for current user (bit 0:A, bit 1:B,bit 2:C,)	R	R	R	
LW-9500	(32bit) : user 1's password	R/W	R/W	R/W	
LW-9502	(32bit) : user 2's password	R/W	R/W	R/W	
LW-9504	(32bit) : user 3's password	R/W	R/W	R/W	
LW-9506	(32bit) : user 4's password	R/W	R/W	R/W	
LW-9508	(32bit) : user 5's password	R/W	R/W	R/W	
LW-9510	(32bit) : user 6's password	R/W	R/W	R/W	
LW-9512	(32bit) : user 7's password	R/W	R/W	R/W	
LW-9514	(32bit) : user 8's password	R/W	R/W	R/W	
LW-9516	(32bit) : user 9's password	R/W	R/W	R/W	
LW-9518	(32bit) : user 10's password	R/W	R/W	R/W	
LW-9520	(32bit) : user 11's password	R/W	R/W	R/W	
LW-9522	(32bit) : user 12's password	R/W	R/W	R/W	



Please confirm your Internet connection before downloading the demo project.



## 22.4 Data Sampling

		Read/Write Control			
Address	Description	Local	MACRO	Remote	
		НМІ	MACINO	НМІ	
LB-9025	delete the earliest data sampling file on HMI	\٨/	\٨/	\٨/	
	memory (set ON)	vv	vv	vv	
LB-9026	delete all data sampling files on HMI memory (set	\٨/	\٨/	W/	
	ON)				
LB-9027	refresh data sampling information on HMI memory	W	w	W	
	(set ON)				
LB-9034	save event/data sampling to HMI, USB disk, SD	W	w	W	
	card (set ON)				
LB-11949	delete the earliest data sampling file on SD card	W	w	W	
	(set ON)				
LB-11950	delete all data sampling files on SD card (set ON)	W	W	W	
LB-11951	refresh data sampling information on SD card (set	W	w	W	
	ON)				
LB-11952	delete the earliest data sampling file on USB 1	W	W	W	
	(set ON)				
LB-11953	delete all data sampling files on USB 1 (set ON)	W	W	W	
LB-11954	refresh data sampling information on USB 1 (set	W	w	W	
	ON)				
LB-11955	delete the earliest data sampling file on USB 2	W	W	W	
	(set ON)				
LB-11956	delete all data sampling files on USB 2 (set ON)	W	W	W	
LB-11957	refresh data sampling information on USB 2 (set	W	w	W	
	ON)				
LW-9063	(16bit) : no. of data sampling files on HMI memory	R	R	R	
LW-9064	(32bit) : size of data sampling files on HMI	R	R	R	
	memory				
LW-10489	(16bit) : no. of data sampling files on SD card	R	R	R	
LW-10490	(32bit) : size of data sampling files on SD card	R	R	R	
LW-10492	(16bit) : no. of data sampling files on USB 1	R	R	R	
LW-10493	(32bit) : size of data sampling files on USB 1	R	R	R	
LW-10495	(16bit) : no. of data sampling files on USB 2	R	R	R	
LW-10496	(32bit) : size of data sampling files on USB 2	R	R	R	



## 22.5 Event Log

		Rea	ntrol	
Address	Description	Local	MACRO	Remote
		НМІ		НМІ
LB-9021	reset current event log (set ON)	W	W	W
LB-9022	delete the earliest event log file on HMI memory	\ <b>\</b> /	۱۸/	\٨/
	(set ON)	vv	vv	vv
LB-9023	delete all event log files on HMI memory (set	\A/	\٨/	\\/
	ON)	••	••	vv
LB-9024	refresh event log information on HMI memory	W	W	W
	(set ON)			
LB-9034	save event/data sampling to HMI, USB disk, SD	W	W	W
	card (set ON)			
LB-9042	acknowledge all alarm events (set ON)	W	W	W
LB-9043	unacknowledged events exist (when ON)	R	R	R
LB-11940	delete the earliest event log file on SD card (set	\A/	\٨/	W.
	ON)	••	••	vv
LB-11941	delete all event log files on SD card (set ON)	W	W	W
LB-11942	refresh event log information on SD card (set	\A/	\٨/	١٨/
	ON)	••	••	vv
LB-11943	delete the earliest event log file on USB 1 (set	W	W	W
	ON)			
LB-11944	delete all event log files on USB 1 (set ON)	W	W	W
LB-11945	refresh event log information on USB 1 (set ON)	W	W	W
LB-11946	delete the earliest event log file on USB 2 (set	W	W	W
	ON)			
LB-11947	delete all event log files on USB 2 (set ON)ON)	W	W	W
LB-11948	refresh event log information on USB 2 (set ON)	W	W	W
LW-9060	(16bit) : no. of event log files on HMI memory	R	R	R
LW-9061	(32bit) : size of event log files on HMI memory	R	R	R
LW-9450	(16bit) : time tag of event log – second *1	R/W	R/W	R/W
LW-9451	(16bit) : time tag of event log – minute *1	R/W	R/W	R/W
LW-9452	(16bit) : time tag of event log – hour *1	R/W	R/W	R/W
LW-9453	(16bit) : time tag of event log – day *1	R/W	R/W	R/W
LW-9454	(16bit) : time tag of event log – month *1	R/W	R/W	R/W
LW-9455	(16bit) : time tag of event log – year *1	R/W	R/W	R/W
LW-10480	(16bit) : no. of event log files on SD card	R	R	R



LW-10481	(32bit) : size of event log files on SD card	R	R	R
LW-10483	(16bit) : no. of event log files on USB 1	R	R	R
LW-10484	(32bit) : size of event log files on USB 1	R	R	R
LW-10486	(16bit) : no. of event log files on USB 2	R	R	R
LW-10487	(32bit) : size of event log files on USB 2	R	R	R



1. If LW-9450~LW-9455 are used as tags of Event Log time source, please set [system parameters] / [General] correctly.





### 22.6 HMI Hardware Operation

		Read/Write Co		ontrol	
Address	Description	Local HMI	Macro	Remote HMI	
LB-9018	disable mouse cursor (set ON)	R/W	R/W	R/W	
LB-9019	disable/enable buzzer	R/W	R/W	R/W	
LB-9020	show (set ON)/ hide (set OFF) system setting bar	R/W	R/W	R/W	
LB-9033	disable(when on)/enable (when off) HMI upload function(i series only)	R/W	R/W	R	
LB-9040	backlight up (set ON) *1	W	W	W	
LB-9041	backlight down (set ON) *1	W	W	W	
LB-9047	reboot HMI (set ON when LB9048 is on)	W	W	W	
LB-9048	reboot-HMI protection	R/W	R/W	R/W	
LB-9062	open hardware setting dialog (set ON)	W	W	W	
LB-9063	disable(set ON)/enable(set OFF) popuping information dialog while finding an USB disk (i series support only)	R/W	R/W	R/W	
LW-9008	(32bit-float) : battery voltage (i series supports only) *2	R	R	R	
LW-9025	(16bit) : CPU loading (x 100%)	R	R	R	
LW-9026	(16bit) : OS version (year)	R	R	R	
LW-9027	(16bit) : OS version (month)	R	R	R	
LW-9028	(16bit) : OS version (day)	R	R	R	
LW-9040	(16bit) : backlight index *1	R	R	R	
LW-9080	(16bit) : backlight saver time (unit : minute)	R/W	R/W	R/W	
LW-9081	(16bit) : screen saver time (unit : minute)	R/W	R/W	R/W	



1. LW-9040 used together with LB-9040~LB-9041 can adjust the backlight brightness with level 0~31.

2. For LW-9008, when the battery voltage level goes below 2.89V, it is recommended to replace the battery.



## 22.7 Local HMI Network Information

		Read/Write Co		ontrol	
Address	Description	Local HMI	Macro	Remote HMI	
LW-9125	(16bit) : HMI ethernet gateway 0 (machine used only)	R/W	R/W	R/W	
LW-9126	(16bit) : HMI ethernet gateway 1 (machine used only)	R/W	R/W	R/W	
LW-9127	(16bit) : HMI ethernet gateway 2 (machine used only)	R/W	R/W	R/W	
LW-9128	(16bit) : HMI ethernet gateway 3 (machine used only)	R/W	R/W	R/W	
LW-9129	(16bit) : HMI ethernet IP 0 (machine used only)	R/W	R/W	R/W	
LW-9130	(16bit) : HMI ethernet IP 1 (machine used only)	R/W	R/W	R/W	
LW-9131	(16bit) : HMI ethernet IP 2 (machine used only)	R/W	R/W	R/W	
LW-9132	(16bit) : HMI ethernet IP 3 (machine used only)	R/W	R/W	R/W	
LW-9133	(16bit) : ethernet port no.	R	R	R	
LW-9135	(16bit) : media access control (MAC) address 0	R	R	R	
LW-9136	(16bit) : media access control (MAC) address 1	R	R	R	
LW-9137	(16bit) : media access control (MAC) address 2	R	R	R	
LW-9138	(16bit) : media access control (MAC) address 3	R	R	R	
LW-9139	(16bit) : media access control (MAC) address 4	R	R	R	
LW-9140	(16bit) : media access control (MAC) address 5	R	R	R	



# 22.8 Recipe and Extended Memory

		Rea	d/Write Co	ntrol
Address	Description	Local HMI	MACRO	Remote HMI
LB-9028	reset all recipe data (set ON)	W	W	W
LB-9029	save all recipe data to machine (set ON)	W	W	W
LB-9460	EM0's storage device (SD card) does not exist (when ON)	R	R	R
LB-9461	EM1's storage device (SD card) does not exist (when ON)	R	R	R
LB-9462	EM2's storage device (SD card) does not exist (when ON)	R	R	R
LB-9463	EM3's storage device (SD card) does not exist (when ON)	R	R	R
LB-9464	EM4's storage device (SD card) does not exist (when ON)	R	R	R
LB-9465	EM5's storage device (SD card) does not exist (when ON)	R	R	R
LB-9466	EM6's storage device (SD card) does not exist (when ON)	R	R	R
LB-9467	EM7's storage device (SD card) does not exist (when ON)	R	R	R
LB-9468	EM8's storage device (SD card) does not exist (when ON)	R	R	R
LB-9469	EM9's storage device (SD card) does not exist (when ON)	R	R	R
LB-9470	EM0's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9471	EM1's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9472	EM2's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9473	EM3's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9474	EM4's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9475	EM5's storage device (USB1 disk) does not exist (when ON)	R	R	R



LB-9476	EM6's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9477	EM7's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9478	EM8's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9479	EM9's storage device (USB1 disk) does not exist (when ON)	R	R	R
LB-9480	EM0's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9481	EM1's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9482	EM2's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9483	EM3's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9484	EM4's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9485	EM5's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9486	EM6's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9487	EM7's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9488	EM8's storage device (USB2 disk) does not exist (when ON)	R	R	R
LB-9489	EM9's storage device (USB2 disk) does not exist (when ON)	R	R	R



		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LB-9035	HMI free space insufficiency alarm (when ON)	R	R	R
LB-9036	SD card free space insufficiency alarm (when ON)	R	R	R
LB-9037	USB 1 free space insufficiency alarm (when ON)	R	R	R
LB-9038	USB 2 free space insufficiency alarm (when ON)	R	R	R
LW-9070	(16bit) : free space insufficiency warning (Mega bytes)	R	R	R
LW-9071	(16bit) : reserved free space size (K bytes)	R	R	R
LW-9072	(32bit) : HMI current free space (K bytes)	R	R	R
LW-9074	(32bit) : SD current free space (K bytes)	R	R	R
LW-9076	(32bit) : USB 1 current free space (K bytes)	R	R	R
LW-9078	(32bit) : USB 2 current free space (K bytes)	R	R	R

# 22.9 Storage Space Management

Want to know how to use LW-9072~LW-9078 together with Backup object?



Please confirm your Internet connection before downloading the demo project.



### 22.10 Touch Position

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LW-9041	(16bit) : touch status word(bit 0 on = user is touching the screen)	R	R	R
LW-9042	(16bit) : touch x position	R	R	R
LW-9043	(16bit) : touch y position	R	R	R
LW-9044	(16bit) : leave x position	R	R	R
LW-9045	(16bit) : leave y position	R	R	R

Want to know how to trigger relevant registers to change window with finger slide?



Please confirm your Internet connection before downloading the demo project.



## 22.11 Station Number Variables

			Read/Write Control		
Address	Description	Local	MACRO	Remote	
		НМІ		HMI	
LW-10000	(16bit) : var0 - station no variable (usage : var0#address)	R/W	R/W	R/W	
LW-10001	(16bit) : var1 - station no variable (usage : var1#address)	R/W	R/W	R/W	
LW-10002	(16bit) : var2 - station no variable (usage : var2#address)	R/W	R/W	R/W	
LW-10003	(16bit) : var3 - station no variable (usage : var3#address)	R/W	R/W	R/W	
LW-10004	(16bit) : var4 - station no variable (usage : var4#address)	R/W	R/W	R/W	
LW-10005	(16bit) : var5 - station no variable (usage : var5#address)	R/W	R/W	R/W	
LW-10006	(16bit) : var6 - station no variable (usage : var6#address)	R/W	R/W	R/W	
LW-10007	(16bit) : var7 - station no variable (usage : var7#address)	R/W	R/W	R/W	
LW-10008	(16bit) : var8 - station no variable (usage : var8#address)	R/W	R/W	R/W	
LW-10009	(16bit) : var9 - station no variable (usage : var9#address)	R/W	R/W	R/W	
LW-10010	(16bit) : var10 - station no variable (usage : var10#address)	R/W	R/W	R/W	
LW-10011	(16bit) : var11 - station no variable (usage : var11#address)	R/W	R/W	R/W	
LW-10012	(16bit) : var12 - station no variable (usage : var12#address)	R/W	R/W	R/W	
LW-10013	(16bit) : var13 - station no variable (usage : var13#address)	R/W	R/W	R/W	
LW-10014	(16bit) : var14 - station no variable (usage : var14#address)	R/W	R/W	R/W	
LW-10015	(16bit) : var15 - station no variable (usage : var15#address)	R/W	R/W	R/W	





Please confirm your Internet connection before downloading the demo project.



# 22.12 Index Register

		Read/Write Control		itrol
Address	Description	Local HMI	MACRO	Remote HMI
LW-9200	(16bit) : address index 0	R/W	R/W	R/W
LW-9201	(16bit) : address index 1	R/W	R/W	R/W
LW-9202	(16bit) : address index 2	R/W	R/W	R/W
LW-9203	(16bit) : address index 3	R/W	R/W	R/W
LW-9204	(16bit) : address index 4	R/W	R/W	R/W
LW-9205	(16bit) : address index 5	R/W	R/W	R/W
LW-9206	(16bit) : address index 6	R/W	R/W	R/W
LW-9207	(16bit) : address index 7	R/W	R/W	R/W
LW-9208	(16bit) : address index 8	R/W	R/W	R/W
LW-9209	(16bit) : address index 9	R/W	R/W	R/W
LW-9210	(16bit) : address index 10	R/W	R/W	R/W
LW-9211	(16bit) : address index 11	R/W	R/W	R/W
LW-9212	(16bit) : address index 12	R/W	R/W	R/W
LW-9213	(16bit) : address index 13	R/W	R/W	R/W
LW-9214	(16bit) : address index 14	R/W	R/W	R/W
LW-9215	(16bit) : address index 15	R/W	R/W	R/W
LW-9230	(32bit) : address index 16	R/W	R/W	R/W
LW-9232	(32bit) : address index 17	R/W	R/W	R/W
LW-9234	(32bit) : address index 18	R/W	R/W	R/W
LW-9236	(32bit) : address index 19	R/W	R/W	R/W
LW-9238	(32bit) : address index 20	R/W	R/W	R/W
LW-9240	(32bit) : address index 21	R/W	R/W	R/W
LW-9242	(32bit) : address index 22	R/W	R/W	R/W
LW-9244	(32bit) : address index 23	R/W	R/W	R/W
LW-9246	(32bit) : address index 24	R/W	R/W	R/W
LW-9248	(32bit) : address index 25	R/W	R/W	R/W
LW-9250	(32bit) : address index 26	R/W	R/W	R/W
LW-9252	(32bit) : address index 27	R/W	R/W	R/W
LW-9254	(32bit) : address index 28	R/W	R/W	R/W
LW-9256	(32bit) : address index 29	R/W	R/W	R/W
LW-9258	(32bit) : address index 30	R/W	R/W	R/W
LW-9260	(32bit) : address index 31	R/W	R/W	R/W



# 22.13 MTP File Information

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LW-9100	(16bit) : project name (16 words)	R	R	R
LW-9116	(32bit) : project size in bytes	R	R	R
LW-9118	(32bit) : project size in K bytes	R	R	R
LW-9120	(32bit) : compiler version	R	R	R
LW-9122	(16bit) : project compiled date [year]	R	R	R
LW-9123	(16bit) : project compiled date [month]	R	R	R
LW-9124	(16bit) : project compiled date [day]	R	R	R



## 22.14 MODBUS Server Communication

\_\_\_\_\_

			Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI	
LB-9055	MODBUS server (COM 1) receives a request (when ON)	R	R	R	
LB-9056	MODBUS server (COM 2) receives a request (when ON)	R	R	R	
LB-9057	MODBUS server (COM 3) receives a request (when ON)	R	R	R	
LB-9058	MODBUS server (ethernet) receives a request (when ON)	R	R	R	
LW-9270	(16bit) : request's function code - MODBUS server (COM 1)	R	R	R	
LW-9271	(16bit) : request's starting address - MODBUS server (COM 1)	R	R	R	
LW-9272	(16bit) : request's quantity of registers - MODBUS server (COM 1)	R	R	R	
LW-9275	(16bit) : request's function code - MODBUS server (COM 2)	R	R	R	
LW-9276	(16bit) : request's starting address - MODBUS server (COM 2)	R	R	R	
LW-9277	(16bit) : request's quantity of registers - MODBUS server (COM 2)	R	R	R	
LW-9280	(16bit) : request's function code - MODBUS server (COM 3)	R	R	R	
LW-9281	(16bit) : request's starting address - MODBUS server (COM 3)	R	R	R	
LW-9282	(16bit) : request's quantity of registers - MODBUS server (COM 3)	R	R	R	
LW-9285	(16bit) : request's function code - MODBUS server (ethernet)	R	R	R	
LW-9286	(16bit) : request's starting address - MODBUS server (ethernet)	R	R	R	
LW-9287	(16bit) : request's quantity of registers - MODBUS server (ethernet)	R	R	R	
LW-9541	(16bit) : MODBUS/ASCII server station no.	R/W	R/W	R/W	



	-			
	(COM 1)			
LW-9542	(16bit) : MODBUS/ASCII server station no.			R/W
	(COM 2)			
LW-9543	(16bit) : MODBUS/ASCII server station no.			
	(COM 3)	R/W	K/ VV	r./ v v
LW-9544	(16bit) : MODBUS/ASCII server station no.			
	(ethernet)	K/ VV	K/ VV	r./ v v
LW-9570	(32bit): received data count (bytes) (COM 1	Б	D	Б
	MODBUS server)	ĸ	ĸ	ĸ
LW-9572	(32bit): received data count (bytes) (COM 2	Р	Р	Р
	MODBUS server)	ĸ	ĸ	ĸ
LW-9574	(32bit): received data count (bytes) (COM 3	Р	Р	Р
	MODBUS server)	ĸ	ĸ	ĸ
LW-9576	(32bit): received data count (bytes) (Ethernet	Р	Р	
	MODBUS server)	ĸ	ĸ	ĸ



# 22.15 Communication Parameters Settings

		Read/Write Co		ontrol	
Address	Description	Local HMI	MACRO	Remote HMI	
LB-9030	update COM 1 communication parameters (set ON)	R/W	R/W	R/W	
LB-9031	update COM 2 communication parameters (set ON)	R/W	R/W	R/W	
LB-9032	update COM 3 communication parameters (set ON)	R/W	R/W	R/W	
LB-9065	disable/enable COM1 broadcast station no.	R/W	R/W	R/W	
LB-9066	disable/enable COM2 broadcast station no.	R/W	R/W	R/W	
LB-9067	disable/enable COM3 broadcast station no.	R/W	R/W	R/W	
LW-9550	(16bit) : COM 1 mode(0:RS232,1:RS485 2W,2:RS485 4W)	R/W	R/W	R/W	
LW-9551	(16bit) : COM 1 baud rate(7:1200,8:2400,0:4800,1:9600,2:19200,3:3840 0,4:57600,)	R/W	R/W	R/W	
LW-9552	(16bit) : COM 1 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/W	R/W	
LW-9553	(16bit) : COM 1 parity (0 : none, 1: even, 2 : odd)	R/W	R/W	R/W	
LW-9554	(16bit) : COM 1 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/W	R/W	
LW-9555	(16bit) : COM 2 mode(0:RS232,1:RS485 2W,2:RS485 4W)	R/W	R/W	R/W	
LW-9556	(16bit) : COM 2 baud rate(7:1200,8:2400,0:4800,1:9600,2:19200,3:3840 0,4:57600,)	R/W	R/W	R/W	
LW-9557	(16bit) : COM 2 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/W	R/W	
LW-9558	(16bit) : COM 2 parity (0 : none, 1: even, 2 : odd)	R/W	R/W	R/W	
LW-9559	(16bit) : COM 2 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/W	R/W	
LW-9560	(16bit) : COM 3 mode(0:RS232,1:RS485 2W)	R/W	R/W	R/W	
LW-9561	(16bit) : COM 3 baud rate(7:1200,8:2400,0:4800,1:9600,2:19200,3:3840 0,4:57600,)	R/W	R/W	R/W	
LW-9562	(16bit) : COM 3 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/W	R/W	
LW-9563	(16bit) : COM 3 parity (0 : none, 1: even, 2 : odd)	R/W	R/W	R/W	
LW-9564	(16bit) : COM 3 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/W	R/W	
LW-9565	(16bit) : COM 1 broadcast station no.	R/W	R/W	R/W	
LW-9566	(16bit) : COM 2 broadcast station no.	R/W	R/W	R/W	
LW-9567	(16bit) : COM 3 broadcast station no.	R/W	R/W	R/W	
LW-10500	(16bit) : PLC 1 timeout (unit : 100ms)	R/W	R/W	R/W	



LW-10501	(16bit) : PLC 1 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10502	(16bit) : PLC 1 send ACK delay (unit : ms)	R/W	R/W	R/W
LW-10503	(16bit) : PLC 1 parameter 1	R/W	R/W	R/W
LW-10504	(16bit) : PLC 1 parameter 2	R/W	R/W	R/W
LW-10505	(16bit) : PLC 2 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10506	(16bit) : PLC 2 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10507	(16bit) : PLC 2 send ACK delay (unit : ms)	R/W	R/W	R/W
LW-10508	(16bit) : PLC 2 parameter 1	R/W	R/W	R/W
LW-10509	(16bit) : PLC 2 parameter 2	R/W	R/W	R/W
LW-10510	(16bit) : PLC 3 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10511	(16bit) : PLC 3 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10512	(16bit) : PLC 3 send ACK delay (unit : ms)	R/W	R/W	R/W
LW-10513	(16bit) : PLC 3 parameter 1	R/W	R/W	R/W
LW-10514	(16bit) : PLC 3 parameter 2	R/W	R/W	R/W
LW-10515	(16bit) : PLC 4 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10516	(16bit) : PLC 4 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10517	(16bit) : PLC 4 send ACK delay (unit : ms)			
	(SIEMENS S7/400 Link type)			
LW-10518	(16bit) : PLC 4 parameter 1 (SIEMENS S7/400	R/W	R/W	R/W
	rack)	17.44		
LW-10519	(16bit) : PLC 4 parameter 2 (SIEMENS S7/400	R/W	R/W	R/W
	CPU slot)	1011		
LW-10520	(16bit) : PLC 5 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10521	(16bit) : PLC 5 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10522	(16bit) : PLC 5 send ACK delay (unit : ms)	R/W	R/W	R/W
	(SIEMENS S7/400 Link type)	1000		
LW-10523	(16bit) : PLC 5 parameter 1 (SIEMENS S7/400	R/W	R/W	R/W
	rack)			
LW-10524	(16bit) : PLC 5 parameter 2 (SIEMENS S7/400	R/W	R/W	R/W
	CPU slot)			
LW-10525	(16bit) : PLC 6 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10526	(16bit) : PLC 6 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10527	(16bit) : PLC 6 send ACK delay (unit : ms)	R/W	R/W	R/W
	(SIEMENS S7/400 Link type)			
LW-10528	(16bit) : PLC 6 parameter 1 (SIEMENS S7/400	R/W	R/W	R/W
	rack)			
LW-10529	(16bit) : PLC 6 parameter 2 (SIEMENS S7/400	R/W	R/W	R/W



	CPU slot)			
LW-10530	(16bit) : PLC 7 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10531	(16bit) : PLC 7 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10532	(16bit) : PLC 7 send ACK delay (unit : ms)		R/W	
	(SIEMENS S7/400 Link type)	R/W		K/W
LW-10533	(16bit) : PLC 7 parameter 1 (SIEMENS S7/400			
	rack)			
LW-10534	(16bit) : PLC 7 parameter 2 (SIEMENS S7/400		R/W	R/W
	CPU slot)	R/W		
LW-10535	(16bit) : PLC 8 timeout (unit : 100ms)	R/W	R/W	R/W
LW-10536	(16bit) : PLC 8 turn around delay (unit : ms)	R/W	R/W	R/W
LW-10537	(16bit) : PLC 8 send ACK delay (unit : ms)			R/W
	(SIEMENS S7/400 Link type)	FK/ V V	R/W	
LW-10538	(16bit) : PLC 8 parameter 1 (SIEMENS S7/400			
	rack)	FK/ V V	r///	K/W
LW-10539	(16bit) : PLC 8 parameter 2 (SIEMENS S7/400			DAA
	CPU slot)		Γ\/ V V	F\$/ V V



# 22.16 Communication Status with PLC (COM)

\_\_\_\_\_

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LB-9150	auto. connection for PLC 1 (COM1) (when ON)	R/W	R/W	R/W
LB-9151	auto. connection for PLC 2 (COM2) (when ON)	R/W	R/W	R/W
LB-9152	auto. connection for PLC 3 (COM3) (when ON)	R/W	R/W	R/W
LB-9200	PLC 1 status (SN0, COM1), set on to retry connection	R/W	R/W	R/W
LB-9201	PLC 1 status (SN1, COM1), set on to retry connection	R/W	R/W	R/W
LB-9202	PLC 1 status (SN2, COM1), set on to retry connection	R/W	R/W	R/W
LB-9203	PLC 1 status (SN3, COM1), set on to retry connection	R/W	R/W	R/W
LB-9204	PLC 1 status (SN4, COM1), set on to retry connection	R/W	R/W	R/W
LB-9205	PLC 1 status (SN5, COM1), set on to retry connection	R/W	R/W	R/W
LB-9206	PLC 1 status (SN6, COM1), set on to retry connection	R/W	R/W	R/W
LB-9207	PLC 1 status (SN7, COM1), set on to retry connection	R/W	R/W	R/W
LB-9500	PLC 2 status (SN0, COM2), set on to retry connection	R/W	R/W	R/W
LB-9501	PLC 2 status (SN1, COM2), set on to retry connection	R/W	R/W	R/W
LB-9502	PLC 2 status (SN2, COM2), set on to retry connection	R/W	R/W	R/W
LB-9503	PLC 2 status (SN3, COM2), set on to retry connection	R/W	R/W	R/W
LB-9504	PLC 2 status (SN4, COM2), set on to retry connection	R/W	R/W	R/W
LB-9505	PLC 2 status (SN5, COM2), set on to retry connection	R/W	R/W	R/W
LB-9506	PLC 2 status (SN6, COM2), set on to retry connection	R/W	R/W	R/W



LB-9507	PLC 2 status (SN7, COM2), set on to retry connection	R/W	R/W	R/W
LB-9800	PLC 3 status (SN0, COM3), set on to retry connection	R/W	R/W	R/W
LB-9801	PLC 3 status (SN1, COM3), set on to retry connection	R/W	R/W	R/W
LB-9802	PLC 3 status (SN2, COM3), set on to retry connection	R/W	R/W	R/W
LB-9803	PLC 3 status (SN3, COM3), set on to retry connection	R/W	R/W	R/W
LB-9804	PLC 3 status (SN4, COM3), set on to retry connection	R/W	R/W	R/W
LB-9805	PLC 3 status (SN5, COM3), set on to retry connection	R/W	R/W	R/W
LB-9806	PLC 3 status (SN6, COM3), set on to retry connection	R/W	R/W	R/W
LB-9807	PLC 3 status (SN7, COM3), set on to retry connection	R/W	R/W	R/W





# 22.17 Communication Status with PLC (Ethernet)

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LB-9153	auto. connection for PLC 4 (ethernet) (when ON)	R/W	R/W	R/W
LB-9154	auto. connection for PLC 5 (ethernet) (when ON)	R/W	R/W	R/W
LB-9155	auto. connection for PLC 6 (ethernet) (when ON)	R/W	R/W	R/W
LB-9156	auto. connection for PLC 7 (ethernet) (when ON)	R/W	R/W	R/W
LB-9157	auto. connection for PLC 8 (ethernet) (when ON)	R/W	R/W	R/W
LB-9158	auto. connection for PLC 9 (ethernet) (when ON)	R/W	R/W	R/W
LB-10070	forced to reconnect PLC 4 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/W	R/W
LB-10071	forced to reconnect PLC 5 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/W	R/W
LB-10072	forced to reconnect PLC 6 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/W	R/W
LB-10073	forced to reconnect PLC 7 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/W	R/W
LB-10074	forced to reconnect PLC 8 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/W	R/W
LB-10075	forced to reconnect PLC 9 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/W	R/W
LB-10100	PLC 4 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-10400	PLC 5 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-10700	PLC 6 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11000	PLC 7 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11300	PLC 8 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11600	PLC 9 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11900	PLC 10 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11901	PLC 11 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11902	PLC 12 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11903	PLC 13 status (ethernet), set on to retry connection	R/W	R/W	R/W
LB-11904	PLC 14 status (ethernet), set on to retry	R/W	R/W	R/W



	connection			
LB-11905	PLC 15 status (ethernet), set on to retry			DAA
	connection	FC/ V V	r./ v v	R/W
LB-11906	PLC 16 status (ethernet), set on to retry			
	connection	F(/ V V	r/ v v	r/w
LW-9600	(16bit) : PLC 4's IP0 (IP address =			
	IP0:IP1:IP2:IP3)			
LW-9601	(16bit) : PLC 4's IP1 (IP address =	R/M	R/W	R/M
	IP0:IP1:IP2:IP3)	10.00		
LW-9602	(16bit) : PLC 4's IP2 (IP address =	R/M	R/W	R/M
	IP0:IP1:IP2:IP3)	10.00		
LW-9603	(16bit) : PLC 4's IP3 (IP address =	R/W	RW	R/W
	IP0:IP1:IP2:IP3)	1000		
LW-9604	(16bit) : PLC 4's port no.	R/W	R/W	R/W
LW-9605	(16bit) : PLC 5's IP0 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1000		
LW-9606	(16bit) : PLC 5's IP1 (IP address =	R/W	RW	R/W
	IP0:IP1:IP2:IP3)	1000		
LW-9607	(16bit) : PLC 5's IP2 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1011		1011
LW-9608	(16bit) : PLC 5's IP3 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1011		1011
LW-9609	(16bit) : PLC 5's port no.	R/W	R/W	R/W
LW-9610	(16bit) : PLC 6's IP0 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1000		1011
LW-9611	(16bit) : PLC 6's IP1 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1000		
LW-9612	(16bit) : PLC 6's IP2 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1011		1011
LW-9613	(16bit) : PLC 6's IP3 (IP address =	R/W	R/W	R/W
	IP0:IP1:IP2:IP3)	1000		
LW-9614	(16bit) : PLC 6's port no.	R/W	R/W	R/W
LW-9615	(16bit) : PLC 7's IP0 (IP address =	R/W	RW	R/W
	IP0:IP1:IP2:IP3)	1000		1011
LW-9616	(16bit) : PLC 7's IP1 (IP address =	R/\//	R/W	R/\\/
	IP0:IP1:IP2:IP3)			
LW-9617	(16bit) : PLC 7's IP2 (IP address =	R/\//	R/W	R/W
	IP0:IP1:IP2:IP3)	1.7.4.4	1.7.7.7	1.7.4.4



LW-9618	(16bit) : PLC 7's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9619	(16bit) : PLC 7's port no.	R/W	R/W	R/W
LW-9620	(16bit) : PLC 8's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9621	(16bit) : PLC 8's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9622	(16bit) : PLC 8's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9623	(16bit) : PLC 8's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9624	(16bit) : PLC 8's port no.	R/W	R/W	R/W
LW-9625	(16bit) : PLC 9's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9626	(16bit) : PLC 9's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9627	(16bit) : PLC 9's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9628	(16bit) : PLC 9's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9629	(16bit) : PLC 9's port no.	R/W	R/W	R/W



# 22.18 Communication Status with PLC (USB)

		Read/Write Control			
Address	Description	Local	MACDO	Remote	
		нмі	WACRU	НМІ	
LB-9190	auto. connection for PLC (USB 1) (when ON)	R/W	R/W	R/W	
LB-9191	PLC status (USB 1), set on to retry connection	R/W	R/W	R/W	
LB-9193	auto. connection for PLC (USB 2) (when ON)	R/W	R/W	R/W	
LB-9194	PLC status (USB 2), set on to retry connection	R/W	R/W	R/W	



### 22.19 Communication Status with Remote HMI

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LB-9068	auto. connection for remote HMI 1 (when ON)	R/W	R/W	R/W
LB-9069	auto. connection for remote HMI 2 (when ON)	R/W	R/W	R/W
LB-9070	auto. connection for remote HMI 3 (when ON)	R/W	R/W	R/W
LB-9071	auto. connection for remote HMI 4 (when ON)	R/W	R/W	R/W
LB-9072	auto. connection for remote HMI 5 (when ON)	R/W	R/W	R/W
LB-9073	auto. connection for remote HMI 6 (when ON)	R/W	R/W	R/W
LB-9074	auto. connection for remote HMI 7 (when ON)	R/W	R/W	R/W
LB-9075	auto. connection for remote HMI 8 (when ON)	R/W	R/W	R/W
LB-9100	remote HMI 1 status (set on to retry connection)	R/W	R/W	R/W
LB-9101	remote HMI 2 status (set on to retry connection)	R/W	R/W	R/W
LB-9102	remote HMI 3 status (set on to retry connection)	R/W	R/W	R/W
LB-9103	remote HMI 4 status (set on to retry connection)	R/W	R/W	R/W
LB-9104	remote HMI 5 status (set on to retry connection)	R/W	R/W	R/W
LB-9105	remote HMI 6 status (set on to retry connection)	R/W	R/W	R/W
LB-9106	remote HMI 7 status (set on to retry connection)	R/W	R/W	R/W
LB-9107	remote HMI 8 status (set on to retry connection)	R/W	R/W	R/W
LB-9149	forced to reconnect remote HMI when IP changed on-line (set ON)	R/W	R/W	R/W
LW-9800	(16bit) : remote HMI 1's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9801	(16bit) : remote HMI 1's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9802	(16bit) : remote HMI 1's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9803	(16bit) : remote HMI 1's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9804	(16bit) : remote HMI 1's port no.	R/W	R/W	R/W
LW-9805	(16bit) : remote HMI 2's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9806	(16bit) : remote HMI 2's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9807	(16bit) : remote HMI 2's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W



LW-9808	(16bit) : remote HMI 2's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9809	(16bit) : remote HMI 2's port no.	R/W	R/W	R/W
LW-9810	(16bit) : remote HMI 3's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9811	(16bit) : remote HMI 3's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9812	(16bit) : remote HMI 3's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9813	(16bit) : remote HMI 3's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9814	(16bit) : remote HMI 3's port no.	R/W	R/W	R/W
LW-9815	(16bit) : remote HMI 4's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9816	(16bit) : remote HMI 4's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9817	(16bit) : remote HMI 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9818	(16bit) : remote HMI 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9819	(16bit) : remote HMI 4's port no.	R/W	R/W	R/W
LW-9820	(16bit) : remote HMI 5's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9821	(16bit) : remote HMI 5's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9822	(16bit) : remote HMI 5's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9823	(16bit) : remote HMI 5's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9824	(16bit) : remote HMI 5's port no.	R/W	R/W	R/W
LW-9825	(16bit) : remote HMI 6's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9826	(16bit) : remote HMI 6's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9827	(16bit) : remote HMI 6's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9828	(16bit) : remote HMI 6's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W



LW-9829	(16bit) : remote HMI 6's port no.	R/W	R/W	R/W
LW-9830	(16bit) : remote HMI 7's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9831	(16bit) : remote HMI 7's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9832	(16bit) : remote HMI 7's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9833	(16bit) : remote HMI 7's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9834	(16bit) : remote HMI 7's port no.	R/W	R/W	R/W
LW-9835	(16bit) : remote HMI 8's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9836	(16bit) : remote HMI 8's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9837	(16bit) : remote HMI 8's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9838	(16bit) : remote HMI 8's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9839	(16bit) : remote HMI 8's port no.	R/W	R/W	R/W


## 22.20 Communication Status with Remote PLC

\_\_\_\_

		Rea	Read/Write Control		
Address	Description	Local	MACRO	Remote	
		нмі	WACKU	НМІ	
LW-10050	(16bit) : IP0 of the HMI connecting to remote PLC				
	1 (IP address = IP0:IP1:IP2:IP3)		11/11		
LW-10051	(16bit) : IP1 of the HMI connecting to remote PLC	R/W	R/M	R/W	
	1 (IP address = IP0:IP1:IP2:IP3)	10.00	1.7.00		
LW-10052	(16bit) : IP2 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	1 (IP address = IP0:IP1:IP2:IP3)	1011			
LW-10053	(16bit) : IP3 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	1 (IP address = IP0:IP1:IP2:IP3)				
LW-10054	(16bit) : port no. of the HMI connecting to remote	R/W	R/W	R/W	
	PLC 1				
LW-10055	(16bit) : IP0 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	2 (IP address = IP0:IP1:IP2:IP3)				
LW-10056	(16bit) : IP1 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	2 (IP address = IP0:IP1:IP2:IP3)				
LW-10057	(16bit) : IP2 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	2 (IP address = IP0:IP1:IP2:IP3)				
LW-10058	(16bit) : IP3 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	2 (IP address = IP0:IP1:IP2:IP3)				
LW-10059	(16bit) : port no. of the HMI connecting to remote	R/W	R/W	R/W	
	PLC 2				
LW-10060	(16bit) : IP0 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	3 (IP address = IP0:IP1:IP2:IP3)				
LW-10061	(16bit) : IP1 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	3 (IP address = IP0:IP1:IP2:IP3)				
LW-10062	(16bit) : IP2 of the HMI connecting to remote PLC	R/W	R/W	R/W	
1.14/ 40000	3 (IP address = IP0:IP1:IP2:IP3)				
LVV-10063	(16bit) : IP3 of the HMI connecting to remote PLC	R/W	R/W	R/W	
	3 (IP address = IP0:IP1:IP2:IP3)				
LVV-10064	(16bit) : port no. of the HMI connecting to remote	R/W	R/W	R/W	
1.11/ 10065	PLC 3				
LVV-10000	(1000, 100, 100, 100, 100, 100, 100, 100	R/W	R/W	R/W	
1.W-10066	+ (ii) address = ii 0.iF 1.iF 2.iF 3) (16bit) : IP1 of the HMI connecting to remote PI C				
	(10010). IF 1 of the finit connecting to remote PLC 4 (ID address = ID0·ID1·ID2·ID3)	R/W	R/W	R/W	
	4 (IF addless = IFU.IF I.IF2.IF3)				



LW-10067	(16bit) : IP2 of the HMI connecting to remote PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10068	(16bit) : IP3 of the HMI connecting to remote PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10069	(16bit) : port no. of the HMI connecting to remote PLC 4	R/W	R/W	R/W
LW-10300	(16bit) : remote PLC 1's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10301	(16bit) : remote PLC 1's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10302	(16bit) : remote PLC 1's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10303	(16bit) : remote PLC 1's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10304	(16bit) : remote PLC 1's port no.	R/W	R/W	R/W
LW-10305	(16bit) : remote PLC 2's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10306	(16bit) : remote PLC 2's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10307	(16bit) : remote PLC 2's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10308	(16bit) : remote PLC 2's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10309	(16bit) : remote PLC 2's port no.	R/W	R/W	R/W
LW-10310	(16bit) : remote PLC 3's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10311	(16bit) : remote PLC 3's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10312	(16bit) : remote PLC 3's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10313	(16bit) : remote PLC 3's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10314	(16bit) : remote PLC 3's port no.	R/W	R/W	R/W
LW-10315	(16bit) : remote PLC 4's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10316	(16bit) : remote PLC 4's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W



LW-10317	(16bit) : remote PLC 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10318	(16bit) : remote PLC 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-10319	(16bit) : remote PLC 4's port no.	R/W	R/W	R/W



# 22.21 Communication Error Messages & No. of Pending Cmd.

		Read/Write Contr		ntrol
Address	Description	Local HMI	MACRO	Remote HMI
LW-9350	(16bit) : pending command no. in local HMI	R	R	R
LW-9351	(16bit) : pending command no. in PLC 1 (COM 1)	R	R	R
LW-9352	(16bit) : pending command no. in PLC 2 (COM 2)	R	R	R
LW-9353	(16bit) : pending command no. in PLC 3 (COM 3)	R	R	R
LW-9354	(16bit) : pending command no. in PLC 4 (ethernet)	R	R	R
LW-9355	(16bit) : pending command no. in PLC 5 (ethernet)	R	R	R
LW-9356	(16bit) : pending command no. in PLC 6 (ethernet)	R	R	R
LW-9357	(16bit) : pending command no. in PLC 7 (ethernet)	R	R	R
LW-9390	(16bit) : pending command no. in PLC (USB)	R	R	R
LW-9400	(16bit) : error code for PLC 1	R	R	R
LW-9401	(16bit) : error code for PLC 2	R	R	R
LW-9402	(16bit) : error code for PLC 3	R	R	R
LW-9403	(16bit) : error code for PLC 4	R	R	R
LW-9404	(16bit) : error code for PLC 5	R	R	R
LW-9405	(16bit) : error code for PLC 6	R	R	R
LW-9406	(16bit) : error code for PLC 7	R	R	R
LW-9407	(16bit) : error code for PLC 8	R	R	R
LW-9490	(16bit) : error code for USB PLC	R	R	R



## 22.22 Miscellaneous Functions

		Read/Write C		ontrol	
Address	Description	Local HMI	Macro	Remote HMI	
LB-9000 ~	initialized as ON	R/W	R/W	R/W	
LB-9010	data download indicator	R	R	R	
LB-9011	data upload indicator	R	R	R	
LB-9012	data download/upload indicator	R	R	R	
LB-9016	status is on when a client connects to this HMI	R	R	R	
LB-9017	disable write-back in PLC control's [change window]	R/W	R/W	R/W	
LB-9039	status of file backup activity (backup in process if ON)	R	R	R	
LB-9045	memory-map communication fails (when ON)	R	R	R	
LB-9049	enable (set ON)/disable (set OFF) watch dog (i series support only) *1	R/W	R/W	R/W	
LB-9059	disable MACRO TRACE function (when ON) *2	R/W	R/W	R/W	
LB-9064	enable USB barcode device (disable keyboard) (when ON) *3	R/W	R/W	R	
LW-9006	(16bit) : connected client no.	R	R	R	
LW-9024	(16bit) : memory link system register	R/W	R/W	R/W	
LW-9032	(8 words) : folder name of backup history files to SD, USB memory	R/W	R/W	R/W	
LW-9050	(16bit) : current base window ID	R	R	R	
LW-9134	(16bit) : language mode *4	R/W	R/W	R/W	
LW-9300	(16bit) : driver ID of local PLC 1	R	R	R	
LW-9301	(16bit) : driver ID of local PLC 2	R	R	R	
LW-9302	(16bit) : driver ID of local PLC 3	R	R	R	
LW-9303	(16bit) : driver ID of local PLC 4	R	R	R	
LW-9530	(8 words) : VNC server password	R/W	R/W	R/W	





1. When LB-9049 watch dog function is enabled, if there's a failure in communication for i Series HMI, system will reboot 10 seconds later.

2. LB-9059 Demonstration Project



3. LB-9064 Demonstration Project

4. When users would like to have the object's text to show multi-language, except for using Label Library, it needs to use the system reserved register [LW-9134: language mode]. The value of LW-9134 can be set from 0 to 7. Different data of LW-9134 corresponds to different Languages. The way of using LW-9134 will differ if the languages are not all chosen when compiling the downloaded file.

For example: If 5 languages are defined by user in Label Library as Language 1 (Traditional Chinese), Language 2 (Simplified Chinese), Language 3 (English), Language 4 (French), and Language 5 (Japanese). If only Language 1, 3, 5 are downloaded by user, the corresponding language of the value in LW-9134 will be 0 -> Language 1 (Traditional Chinese), 1 -> Language 3 (English), 2 -> Language 5 (Japanese).

Want to know how to swith languages using Option List object toghther with LW-9134?



Please confirm your Internet connection before downloading the demo project.



# 22.23 Remote Print/Backup Server

		Read/Write Cor		ntrol
Address	Description	Local HMI	MACRO	Remote HMI
LB-10069	forced to reconnect remote printer/backup server when IP changed on-line (set ON)	R/W	R/W	R/W
LW-9770	(16bit) : remote printer/backup server IP0 (IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9771	(16bit) : remote printer/backup server IP1 (IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9772	(16bit) : remote printer/backup server IP2 (IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9773	(16bit) : remote printer/backup server IP3 (IP0:IP1:IP2:IP3)	R/W	R/W	R/W
LW-9774	(6 words) : remote printer/backup server user name	R/W	R/W	R/W
LW-9780	(6 words) : remote printer/backup server password	R/W	R/W	R/W



## 22.24 EasyAccess

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LB-9051	disconnect (set OFF)/connect (set ON) EasyAccess server	R/W	R/W	R/W
LB-9052	status of connecting to EasyAccess server	R	R	R

For further information on EasyAccess, please visit http://www.ihmi.net/.



# 22.25 Pass-Through Settings

		Read/Write Control		
Address	Description	Local HMI	MACRO	Remote HMI
LW-9900	(16bit) : HMI run mode (0 : normal mode, 1~3 : test mode (COM 1~COM 3)	R/W	R/W	R/W
LW-9901	(16bit) : pass-through source COM port (1~3 : COM 1~COM 3)	R/W	R/W	R/W
LW-9902	(16bit) : pass-through destination COM port (1~3 : COM 1~COM 3)	R/W	R/W	R/W



# 22.26 Disable PLC No Response Dialog Box

		Rea	d/Write Co	/Write Control	
Address	Description	Local HMI	MACRO	Remote HMI	
LB-9192	disable USB 1 PLC's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-9195	disable USB 2 PLC's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11960	disable PLC 1's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11961	disable PLC 2's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11962	disable PLC 3's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11963	disable PLC 4's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11964	disable PLC 5's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11965	disable PLC 6's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11966	disable PLC 7's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	
LB-11967	disable PLC 8's "PLC No Response" dialog (when ON)	R/W	R/W	R/W	



## 22.27 HMI and Project Key

Address		Rea	ntrol	
	Description	Local HMI	MACRO	Remote HMI
LB-9046	project key is different from HMI key (when ON)	R	R	R
LW-9046	(32bit) : HMI key (i series only)	R/W	R/W	R



Please confirm your Internet connection before downloading the demo project.





## 22.28 Fast Selection Window Control

Address		Read/Write Control		
	Description	Local	Local MACRO	Remote
		HMI	НМІ	
LB-9013	FS window control[hide(ON)/show(OFF)]	R/W	R/W	R/W
LB-9014	FS button control[hide(ON)/show(OFF)]	R/W	R/W	R/W
LB-9015	FS window/button control[hide(ON)/show(OFF)]	R/W	R/W	R/W



# 22.29 Input Object Function

		Read/Write Cont		ntrol
Address	Description	Local HMI	MACRO	Remote HMI
LW-9002	(32bit-float) : input high limit	R	R	R
LW-9004	(32bit-float) : input low limit	R	R	R
LW-9052	(32bit-float) : the previous input value of the numeric input object	R	R	R
LW-9150	(32 words) : keyboard's input data (ASCII)	R	R	R
LW-9540	(16bit) : reserved for caps lock	R	R	R



## 22.30 Local/Remote Operation Restrictions

		Rea	d/Write Control	
Address	Description	Local HMI	MACRO	Remote HMI
LB-9044	disable remote control (when ON)	R/W	R/W	R/W
LB-9053	prohibit password remote-read operation (when ON)	R/W	R/W	R/W
LB-9054	prohibit password remote-write operation (when ON)	R/W	R/W	R/W
LB-9196	local HMI supports monitor function only (when ON)	R/W	R/W	R/W
LB-9197	support monitor function only for remote HMIs (when ON)	R/W	R/W	R/W
LB-9198	disable local HMI to trigger a MACRO (when ON)	R/W	R/W	R/W
LB-9199	disable remote HMI to trigger a MACRO (when ON)	R/W	R/W	R/W



# **Chapter 23 HMI Supported Printers**

#### 1. EPSON ESC/P2



Impact Printer: LQ-300, LQ-300+, LQ-300K+ (RS232) LQ-300+II (RS232)

Inkjet Printer: Stylus Photo 750

#### Laser Printer: EPL-5800

### 2. HP PCL Series



USB port, conform to HP PCL level 3 protocol.

#### Laser Printer:

HP LaserJet P1505n: HP PCL 5e

- PCL 5 was released on HP LaserJet III in March 1990, added Intellifont font scaling (developed by Compugraphic, now part of Agfa), outline fonts and HP-GL/2 (vector) graphics.
- PCL 5e (PCL 5 enhanced) was released on HP LaserJet 4 in October 1992 and added bi-directional communication between printer and PC, and Windows fonts.



#### Caution: For HP printer, we do not support

- 1. HP LaserJet P1005, which is not PCL 5.
- 2. HP LaserJet P1006
- 3. HP LaserJet 1000, which supports HostBase Printing language
- 4. HP LaserJet 1010, which supports HostBase Printing language
- 5. HP Color LaserJet 1500, which supports HostBase Printing language

6. HP Color LaserJet 3500, which supports HostBase Printing language MT8000 does not support HostBase Printing language.

# Please ensure that the HP printer supports PCL5 before connecting with MT8000 series. Otherwise MT8000 will stop responding with a black screen.

Inkjet Printer:

HP DeskJet 920C, 930C, D2360, D2560, D2568

#### 3. SP-M, D, E, F

EPSON ESC protocol 9-pin printer.

RS232 port SIUPO http://www.siupo.com SP-M, D, E, F series SP-E1610SK (paper width: 45mm) SP-E400-4S (paper width: 57.5mm)



SP-MDEF



Recommended SP printer type for customers outside China

Please refer to manual before using printer.



#### 4. Axiohm A630



Micro printer Axiohm A630 from France connected via serial port.

## 5. SPRT (SP-DIII, DIV, D5, D6, A, DN, T)



SP-DN40SH dot matrix printer SP-RMDIII40SH thermal printer

#### 6. EPSON TM-L90





## **Chapter 24 Recipe Editor**

Recipe Editor is a Win32 application and can only run on MS Windows 2000 / XP / Vista / 7. It allows users to create, view and modify Recipe (\*.rcp) and EMI (\*.emi) files. Additionally, it can convert Recipe and EMI files to CSV format and vice versa.

#### 24.1 Introduction

Under Recipe Editor [file] -> [new], the following dialogue appears:

Set	Data Format			
	Address range (ur From 0 Data format	nit : word) To 100	Select your data format Format 0 Save Format D	▼ elete Format
	Size	Туре	Description	Add
	1 WORD	16-bit Unsinged	Data 0	Delete Clear All Modify
	<			OK Cancel

Setting	Description
Address range	Fill in the address range users want to examine. The unit is "word".
Add	Add a column to the current data format template.
Delete	Delete the selected column.
Clear All	Delete all columns.
Modify	Modify the description and data info for the selected column.



Save Format	Save the settings of the current data format template so that users							
	can load it every time when needed without recreating it repeatedly.							
	The template data will be stored as "data.fmt" file in the							
	EasyBuilder8000 installation directory.							
Delete Format	Delete an existed data format template.							
Select your	Select an existing data format template for examining the Recipe or							
data format	EMI data.							

After clicking [Add...], [Data Type] dialogue appears as follow:

Data Type	×
Description : Data 0	
C 16-bit BCD	C 32-bit BCD
C 16-bit HEX	O 32-bit HEX
16-bit Unsigned	C 16-bit Signed
O 32-bit Unsigned	C 32-bit Signed
C Float	
C String	WORD(s)
ОК	Cancel

First, users can assign a name as **[Description]** for the column and then select the correct data type. If **[String]** is selected, users must specify the length of the string.



## 24.2 Settings of Recipe Editor

#### How to Add a Recipe / EMI File

(1) Under Recipe Editor [file] -> [new], the following dialogue appears:

Set Data Format	
Address range (unit : word) From 0 To 100 Save Format Del	▼ lete Format
Size Type Description	Add Delete Clear All Modify
	OK Cancel

(2) Click [Add...] and select [16-bit Unsigned] as data format type.

Description : Data 1	
C 16-bit BCD	C 32-bit BCD
C 16-bit HEX	C 32-bit HEX
🕫 16-bit Unsigned	○ 16-bit Signed
© 32-bit Unsigned	C 32-bit Signed
⊂ Float	
C String	WORD(s)



(3) After all the settings finished, a new document appears as follow

🖆 Rec	ipe/Ex	tended Me	nory Ed	litor - New document*	
<u>F</u> ile <u>E</u>	<u>dit ⊻</u> i	ew <u>W</u> indow	/ <u>H</u> elp		
0	ê 🔒	X 🖻	16	<b>?</b>	
	_				
	🖆 Ne	w documen	đ*		
	ID	ADDRESS	Data 0		
	0	0	0		
	1	1	0		
	2	2	0		
	3	3	0		
	4	4	0		
	5	5	0		
	7	7	0		
	8	8	U		
	9	10	0		
	10	10	0		
	12	12	0		
	13	13	0		
	14	14	0		
	15	15	0		
	16	16	0		
	17	17	0		
	18	18	0		
	19	19	0		
	20	20	0		
Ready				NUM	

(4) Users can view and modify the data listed.

(5) In [Save As], select the correct format and file name to create a recipe or emi file.

#### **Export to CSV File**

After opening a recipe or emi file, select [Save As] and choose file format as CSV.

#### Import CSV File

Under Recipe Editor [file] -> [Import CSV File] choose a CSV file to open. After editing, users can save it as a recipe or emi file so that it can be downloaded to HMI.



## Chapter 25 EasyConverter

This application program is utilized when converting the history record of data sampling (dtl) or event log (evt) stored in HMI to Excel (csv) that is readable on PC installed with Microsoft Excel. The completed conversion can be exported to Excel.

#### **25.1 Introduction**

File Edit View Options Help

In Project Manager, clicking [EasyConverter] will pop up the application program.

There are four functions as follows:

- 1. Export to Excel
- 2. Scaling function
- 3. Multi-File Conversion
- 4. Command line

EasyBuilder8000 User's Manual



## 25.2 Settings of EasyConverter

#### 25.2.1 How to Export to Excel

When open the file, a setting dialog will pop up as follow:

1 16-bit Unsignet 16-bit Unsigne 1 0 No Scaling & Offset	No	Name	Туре	Word Size	Digits	Scaling
Scaling & Offset N/A	1	16-bit Unsigned	16-bit Unsigne	1	0	No 💌
Scaling & Offset N/A						
Scaling & Offset						
Scaling & Offset N/A						
Scaling & Offset						
Scaling & Offset						
N/A			uu			>
	< Sc	aling & Offset	Ruit			>
	Sc N	aling & Offset	17 MU 			>
	Sc N	aling & Offset			_	) >
	Sc N/	aling & Offset /A				>

Click [OK],



File	Edit	<u>V</u> iew	Options	Help	
°	8			?	
[Creation of the second	ation t Nov 2 a] e", "16 i1:06, i1:07, i1:08, i1:23, i1:24, i1:25, i1:26, i1:27, i1:28	ime] 7 10: bit 0 1 0 1 1 2 3	:51:06 Unsign	200	3

Then click [Export to Microsoft Excel].

	A	B	С	
1	[Creation t	ime]		
2	Thu Nov 27 10:51:06 2008			
3	[Data]			
4	"Time"	"16-bit Unsi	gned"	
5	10:51:06	0		
6	10:51:07	0		
7	10:51:08	1		
8	10:51:23	0		
9	10:51:24	0		
10	10:51:25	1		
11	10:51:26	1		
12	10:51:27	2		
13	10:51:28	3		
14	10:51:29	3		
1.5	10.51.20	4		

When converting event log in csv format, users can find data fields in EXCEL as below.

	dicrosoft Ex	wel - EL_2	0101029.x	ls	
:e	檔案①	編輯(王) 核	歳(Ⅵ) 插	入① 格式(	0) 工具
10	📬 🖬 🕻	616		2 • 🛷 🗎	🤈 🖬 😡
	2 2 2	1 🔁 🖄 I	301	8 5 6	***回覆後
-	D20		*	fx	
	A	В	C	D	E
1	[Creation t	ime]	300		
2	Fri Oct 29	10:59:28 20	010		
3	[Data fields	5]			
4	event	category	time	message	
5	[Data]				
6	0	0	11:19:42	"Emergen	cy"
7	0	5	11:19:43	"5"	-
8	2	0	11:19:46	"LOW"	
9	2	5	11:19:49	"5"	
10	1	0	11:19:52	"Word"	
11	2	0	11:19:52	"Word"	
12					
13					
14					

- 0 -> event is triggered
- 1 -> event is acknowledged
- 2 -> event returns to normal



### 25.2.2 How to Use Scaling Function

The **Scaling** is utilized to offset data.

No	Name	Туре	Nord Size	Digit	Scal	ing
1	16-bit Unsigned	16-bit Unsigned	1	0	No	~

new value = {{value+A}xB}+C, users can set value on A, B, and C.

#### Why do we need the Scaling function?

For example, here is a data of voltage and data format is 16-bit unsigned (range: 0~4096).

If users want to map those data to volt range form -5 to +5, the calculation:

new value =  $\{\{value+0\} \times 0.0024\} + (-5), as follow:$ 



s	атр	ling Data Info	rmation				×
	Selea	ct number of digi	ts after decim	nal point	t:		
	No	Name	Туре	Word S	Size Digits	Scaling	
	1	16-bit Unsigned	16-bit Unsi;	1	3	Yes 💌	
1	<u> </u>	- K					
	-50	aling & Urrset					]
	_	A	В		C	-	
		0.0000	0.002	4	-5.000	JU	
	ne	w value = (( valu	⊫e+A)xB)	+			
		= ( value	x 0.0024 ) +	-5.000	0		
		d Cotting					-
		u setting	_				
							5 I

Settings of data above can be saved as a sample and loaded next time.



After the scaling,

## Original file

## File after utilizing scaling function



#### 25.2.3 How to Use Multi-File Conversion

Step1: Click [File] / [Multi-File] a setting dialog will pop up.

Step2: Click [Add File...] to add files into "List".

Multi-File
Convert file list: C:\EB8000\eng1.34\datalog\123\20081127.dtl C:\EB8000\eng1.34\datalog\trend\20081128.dtl
Add File Delete File
Combine to a file
C:\Documents and Settings\user\test.xls OK Cancel

**Step3:** After adding files, check **[Combine to a file]**, files will be separated into sheets of one EXCEL file labeled with the dated it is added.



	A	В	С	D	E	F	G	
7	#######################################	11:02:32	620	0				
8	######################################	11:32:33	680	0				
9	######################################	11:32:34	680	0				
10	######################################	11:32:35	680	0				
11	######################################	11:32:36	680	0				
12	######################################	11:32:37	680	0				
13	######################################	11:32:38	680	0				
14	######################################	11:32:39	680	0				
15	######################################	11:32:40	680	0				
16	######################################	11:32:41	680	0				
17	######################################	11:32:42	700	0				
18	######################################	11:32:43	680	0				
19	######################################	11:32:44	680	0				
20	######################################	11:32:45		0				~
<b>I</b> 4 4	- ) 1 200	<u>281127</u> 20	081128/		<		>	
就緒					1	IUM		

Note: If users don't check this box, the files will be exported to Excel individually.





#### 25.3 Enable Setting File

User can load an existing Setting file to apply to a data log file(s).

Sampling Data Information Select number of digits after decimal point: No Name Туре Word Size Digits Scaling 16-bit Unsigned 1 Yes 💌 2 2 16-bit Unsigned 1 0 No -3 3 1 0 16-bit Unsigned No -Scaling & Offset A В С 0.0000 1.0000 2.0000 new value = (( value + A )  $\times$  B ) + = value + 2.0000 Load Setting.. Save Setting. ΟK Cancel

Step1: Save the setting to test.lgs after filling out [scaling & offset].

Step2: In a new data sampling, click [Load Setting] to load test.lgs.



No	Name	Туре	Word Size	Digits	Scaling
1	1	16-bit Unsigned	1	0	No 💌
2	2	16-bit Unsigned	1	0	No 💌
3	3	16-bit Unsigned	1	0	No 💌
Sc	aling & C	)ffset			
Sc N	aling & C 'A	Dffset ————			

Step3: Press [Export to Microsoft Excel] button to examine the data.

277 · · · · ·			
<u>File</u>	<u>E</u> dit <u>V</u> iew	Options	<u>H</u> elp
6		B 9	?
[Creati	on time]		
Thu No	v 27 10	:51:06	2008
[Data]			
Time,	"16-bit	Unsigne	d"
10:51:	06,0		
10:51:	07,0		
10:51:	08,1		
10:51:	23,0		
10:51:	24,0		
10:51:	25,1		
10:51:	26,1		
10:51:	27,2		
10:51:	28.3		



## 25.3.1 For "Combination" and "Enable Setting File"



#### Step1: Click [Multi-File]

Step2: Select [Add File...]

Auto-File		le l
Convert file list:		
	Add File	Delete File
Enable Setting file	Add File.	. Delete File
Enable Setting file	Add File.	Delete File
Enable Setting file Combine to a file	Add File.	Delete File
Combine to a file	Add File.	Delete File



**Step3:** Select the files that you would like to combine and check both **[Enable Setting file]** and **[Combine to a file]** boxes. With [Combine to a file] edit, please indicate a file name for the new outcome.

Multi-File	
Convert file list:	
C:\Documents and Settings\user\20081127.dtl C:\Documents and Settings\user\20081128.dtl	
	Add File Delete File
Enable Setting file	
C:\EB8000\datalog\pressure\1.lgs	
Combine to a file	
C:\EB8000\datalog\pressure\2.xls	OK Cancel

Step4: After pressing [OK], the data will be displayed.



Step5: Open the newly combined file to examine the data in Microsoft Excel.



### 25.4 Command Line

For EasyConverter, users can run in a command mode.

EasyConverter [/c] [/s] [/t [num]] setting source destination

Setting	Description
setting	Indicate the setting file.(*.lgs)
source	Indicate the source file.(*.dtl or *.evt)
destination	Indicate the destination file.(*.csv or *.xls)
	Type of file output. If this is set, a CSV file will be output,
70	otherwise an EXCEL file.
	Whether involving a setting file or not. If this is set, it indicates
/5	that users utilize a setting file.

For example: EasyConverter.exe /c /s "E:\Work\20080625.lgs" "E:\Work\ 20080625.dtl" "E:\Work\"



# Chapter 26 EasyPrinter

EasyPrinter is a Win32 application and can only run on MS Windows 2000 / XP / Vista / 7. It enables MT8000 Series to output screen hardcopies to a remote PC via Ethernet. Please see the following illustration:



Here are some advantages of using EasyPrinter:

- EasyPrinter provides two modes of hardcopy output: Print-Out and Save-to-File. Users can use either way or both ways.
- Since EasyPrinter is running on MS Windows system, it supports most of the printers available in the market.
- Multiple MT8000 HMI can share one printer via EasyPrinter. Users don't have to prepare printers for each MT8000 HMI.

Additionally, EasyPrinter can also be a backup server. Users can use backup objects in MT8000 HMI to copy history files such as Data-Sampling and Event-Log histories onto a remote PC via Ethernet. Please see the following illustration:




# 26.1 Using EasyPrinter as a Printer Server



Users can make screen hardcopies with a **[Function Key]** object. The hardcopies will be transferred to the MT Remote Printer Server via Ethernet and then printed out.

# 26.1.1 Setup Procedure in EasyPrinter

In [Menu] → [Options], select [Settings...] and the following dialogue appears:



MT8xxx Remote F	Printer Server Settings
General	Server
Hardcopy Backup	Port number of the server socket:       8005         User name:       [Max. length = 12 characters]       admin         Password:       [Max. length = 12 characters]       111111         Naming Convention for HMI Folder (when writing files)
	Use IP address     O Use HMI name (assign HMI name by L W9032~L W9039)  Prefix:
	Properties       IP

- 1. In [Server], assign [Port number of the server socket] to "8005", [User name] to "admin" and [Password] to "111111". (Note: These are default values.)
- 2. In [Naming Convention for HMI Folder], select [Use IP address] and assign "IP\_" as the [Prefix].
- 3. In [Properties], select [Minimize to system tray].

Click [Hardcopy] tab on the left side in the dialogue box as follows:

MT8xxx Remote Printer Server Settings		
General	Output	
Hardcopy	Print out to:	
Backup	In plaseriet 3360 FCL 5	

- 4. In **[Output]**, select **[Print out to]** and choose a printer as the output device for screen hardcopies. (Note: Users can only choose from the printers available in their system, so it is possible that "hp LaserJet 3380 PCL 5" can't be found in the list as the example.)
- 5. Click **[OK]** to apply the settings.
- 6. In [Menu] **>** [File], select [Enable Output] to allow EasyPrinter to output any



incoming print request, i.e. screen hardcopy.

## 26.1.2 Setup Procedure in EasyBuilder8000

In [Menu] → [Edit] → [System Parameters], click [Printer Server] tab and select [Use MT Remote Printer Server], the following dialogue appears:

System Parameter Setti	ngs			X
Device	Model	General	System Setting	Security
Font	Exte	ended Memory	Printer/Bac	kup Server
🔽 Use MT Remote Pr	inter/Backup Se	erver		
Note: Use EasyPinter t	to configure PC	for printing screen h	ardcopy and storing back	up data.
Orientation :	💿 Horizontal	🔺 🔿 Vertica	al <	
Printer size :	💿 Original size	e 🔿 Fit to p	printer margins	
Margin :		15 🜲 mm	1	
	15 🗘	] mm	15 📦 mm	
		15 📚 mm	1	
Communication setting	(S			
IP address :	192 . 168	. 1 . 26		
Port :	8005			
User name :	admin			
Password :	111111			

- 7. In **[Output settings]**, assign appropriate values for left/top/right/bottom margins. (Note: The margins are all assigned to 15mm in the example.)
- In [Communication settings], fill in the [IP address] of the printer server same as step 1, assign the [port number] to "8005", [User name] to "admin" and [Password] to "111111".



In [Menu] → [Objects] → [Buttons], select [Function Key] and assign [Screen hardcopy] to [MT Remote Printer Server].



- 9. Place the **[Function Key]** object in the common window (window no. 4), and users will be able to make screen hardcopies anytime when needed.
- 10. [Compile] and [download] project to MT8000 HMI. Press the [Function Key] object set in step 9 to make a screen hardcopy.

NOTE

- Users can also use a **[PLC Control]** object to make screen hardcopies.
- Users cannot print alarm information via EasyPrinter.
- EasyPrinter can only communicate with HMI via Ethernet, so this feature is unavailable in MT6000 Series.





## 26.2 Using EasyPrinter as a Backup Server



Users can upload historical data such as Data-Sampling and Event-Log history files onto MT remote backup server with **[Backup]** objects.

# 26.2.1 Setup Procedure in EasyPrinter

In [Menu] → [Options], select [Settings...] and the following dialogue appears:

MT8xxx Remote F	Printer Server Settings
General	Server
Hardcopy Backup	Port number of the server socket:     8005       User name: [Max. length = 12 characters]     admin       Password: [Max. length = 12 characters]     111111
	Naming Convention for HMI Folder (when writing files) © Use IP address © Use HMI name (assign HMI name by L W9032~L W9039)
	Prefix: IP_ (Ex: IP_192.168.1.25)
	Properties Minimize to system tray Detailed message

 In [Server], assign [Port number of the server socket] to "8005", [User name] to "admin" and [Password] to "111111". (Note: These are default values.)



- In [Naming Convention for HMI Folder], select [Use IP address] and assign "IP\_" as the [Prefix].
- 3. In [Properties], select [Minimize to system tray].

Click **[Backup]** tab on the left side in the dialogue box as follows:

MT8xxx Remote F	Printer Server Settings
General Hardcopy Backup	Output <u>Backup files in:</u> D:\MT8000 When target file has existed: Overwrite it. (The content will be destroyed) <u>Append BAK to the file name</u>
	Convert Batch File           Enable           D:\MT8000\convert2csv.def

- 4. In **[Output]**, click the **(Q)** button to browse and select a path for storage of the incoming history files.
- 5. Click **[OK]** to apply the settings.
- 6. In [Menu] → [File], select [Enable Output] to allow EasyPrinter to store any incoming backup request in the location specified in step 4.

## 26.2.2 Setup Procedure in EasyBuilder8000

In [Menu] → [Edit] → [System Parameters], click [Printer Server] tab and select [Use MT Remote Printer Server], the following dialogue appears:



System Parameter Setti	ngs			×
Device	Model	General	System Setting	Security
Font	Ex	tended Memory	Printer/Bac	kup Server
🔽 Use MT Remote Pr	rinter/Backup ន	erver		
Note: Use EasyPinter	to configure P(	C for printing screen 1	hardcopy and storing back	up data.
Orientation :	💿 Horizontal	🔺 🔿 Vertio	al <	
Printer size :	💿 Original siz	e 🔿 Fit to	printer margins	
Margin :		15 📚 m	n	
	15 🔹	; mm	15 🛟 mm	
		15 📚 m	n	
Communication setting	iz			
IP address :	192 . 16	8.1.26		
Port :	8005			
User name :	admin			
Password :	111111			

- In [Communication settings], fill in the [IP address] of printer server same as step 1, assign [port number] to "8005", [User name] to "admin" and [Password] to "111111".
- In [Menu] → [Objects], select [Backup] and the following dialogue appears:

WE!NTEK

EasyBuilder8000 User's Manual

New Backup Object 🛛 🔀
General Security Shape Label
Description :
Source
○ R W ○ R W_A ④ Historical event log ○ Historical data log
- Backup position
◯ USB 1 ◯ USB 2 ⊙ Remote printer/backup server
Note : Use L W9032~9039 to change the backup folder name.
Note : Use [Remote printer/backup server] to store data to a remote PC. Enable the server
in [System Farameter][Frinter/Backup Server] settings.
Range
Start : 💿 Today 🔿 Yesterday
Within : All
Mode : Touch trigger

- 8. In [Source], select [Historical event log].
- 9. In [Backup position], select [Remote printer server].
- 10. In [Range], select [Today] and [All].

WEINTEK

- 11. In [Attribute], select [Touch trigger].
- 12. Place the **[Backup]** object in the common window (window no. 4), and users will be able to make backups anytime when needed.
- 13. **[Compile]** and **[download]** project to MT8000 HMI. Press the **[Backup]** object set in step 12 to make a backup of the Event-Log history data.

NOTE	•	The <b>[Backup]</b> object can be triggered via a bit signal.
	•	Users can arrange a [Scheduler] object, which turns a bit ON at the end of
		week, to trigger a [Backup] object to automatically back up all history data.



# 26.3 EasyPrinter Operation Guide

# 26.3.1 Appearance



Area	Name	Description
1	lob List	This window lists all incoming tasks, i.e. screen
	JOD LISI	hardcopy and backup requests.
0	Droporty Window	This window shows the information about the
2		task selected from "Job List."
2	Draviaw Window	This window shows the preview image of the
3		screen hardcopy task selected from "Job List."
Λ	Download Progress	This window shows the download progress of
4	Window	incoming requests.
		This window shows the time and message of
5	Message Window	events such as incoming request, incorrect
		password, etc.



# 26.3.2 Operation Guide

The following tables describe the meaning and explain how to use all EasyPrinter menu items.

Menu → File	Description
Enable Output	Selected
	EasyPrinter processes the tasks one by one.
	<ul> <li>Unselected</li> </ul>
	EasyPrinter arranges the incoming tasks in memory.

## NOTE

• EasyPrinter can only reserve up to 128 MB of task data in memory. If the memory is full, any request coming in afterwards will be rejected and users must either operate [Enable Output] or delete some tasks to make room for new tasks.

Menu → Edit	Description		
Edit	To edit a screen hardcopy task.		
	Edit Print Job		
	Orientation       Scaling            • Portrait           • Original             • Landscape           • Fit To Margin          Margins (mm)          Eft: 15          Left:       15          Top:       15		
	Users can freely change the properties of <b>[Orientation]</b> , <b>[Scaling]</b> and <b>[Margins]</b> here		
Delete	To delete the selected tasks permanently.		
Select All	To select all tasks from "Job List."		

#### NOTE

- The backup task is not editable.
- [Edit] is available only when a task is selected.
- [Delete] is available when at least one task is selected.



Menu $\rightarrow$ View	Description
Properties Bar	To show or hide the Property Window.
Preview Bar	To show or hide the Preview Window.
Download Bar	To show or hide the Download Progress Window.
Logger Bar	To show or hide the Message Window.

• In [Download Progress] Window, users can select the mode to show download progress by clicking the header of the [progress] column. Please see the following illustration:

Download Progress		E	
From	Progr	✓ Perce Data	ntage Display Length Display

• EasyPrinter can reserve up to 10,000 messages in Message Window. If a new message comes in, the oldest message will be deleted.

Menu→Options	Description
Toolbars	To show or hide toolbars.
Status Bar	To show or hide the status bar.
Settings	Configuration for EasyPrinter. Please refer to the following illustrations:
	[General]



General       Server         Hardcopy       Port number of the server socket:       2005         Backup       Port number of the server socket:       2005         Backup       Port number of the server socket:       2005         Password: [Max. length = 12 characters]       111111         Naming Convention for HMI Folder (when writing files)       0 Use IP address         Use IP address       Use HMI name (assign HMI name by L W9032-L W9039)         Prefix:       IP		Cara yan	
Hardcopy       Port number of the server socket:       2000         Backup       User name: [Max. length = 12 characters]       admin         Password: [Max. length = 12 characters]       admin         Password: [Max. length = 12 characters]       111111         Naming Convention for HMI Folder (when writing files)       0 Use IP address         Use HMI name (assign HMI name by L W9032-L W9039)       Prefix:         IP	General	Server	0000
Backup	Hardcopy	Port number of the server socket:	8005 admin
Image: Product proof in the server socket         Naming Convention for HMI Folder (when writing files)         Image: Use IP address         Image: Use IP address         Image: Use HMI name (essign HMI name by LW9032-LW9039)         Prefix:         IP	Backup	<u>U</u> ser name: [Max. length = 12 characters]	111111
Naming Convention for HMI Folder (when writing files)         ● Use IP address         ● Use HMI name (assign HMI name by L W9032-L W9039)         Prefix:         IP		Towners: [rear might - 15 contained]	
<ul> <li>● Use IP address</li> <li>● Use HMI name (assign HMI name by LW9032LW9039)</li> <li>Prefix:</li> <li>IP (Ex: IP_192.168.1.25)</li> <li>Properties</li> <li>♥ Minimize to system tray</li> <li>● Detailed message</li> </ul> [Server] → [Port number of the server socket] Set the Ethernet socket number for HMI to connect range goes from 1 to 65535 and 8005 is the default [Server] → [User name] & [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:		Naming Convention for HMI Folder (when w	riting files)
<pre>[Server] → [Port number of the server socket] Properties Minimize to system tray Detailed message</pre> [Server] → [Port number of the server socket] Set the Ethernet socket number for HMI to connect range goes from 1 to 65535 and 8005 is the default [Server] → [User name] & [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:		O Use IP address	0022 1 10020)
<pre>Prefix: IP (Ex: IP_192.168.1.25) Properties IV Minimize to system tray IDetailed message [Server] → [Port number of the server socket] Set the Ethernet socket number for HMI to connect range goes from 1 to 65535 and 8005 is the default [Server] → [User name] &amp; [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:</pre>		O ose rivit name (assign rivit name by L w	9032~17 (19039)
IF		Prefix:	25)
Properties Minimize to system tray Detailed message [Server] → [Port number of the server socket] Set the Ethernet socket number for HMI to connect range goes from 1 to 65535 and 8005 is the default [Server] → [User name] & [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:			
[Server] → [Port number of the server socket] Set the Ethernet socket number for HMI to connect range goes from 1 to 65535 and 8005 is the default [Server] → [User name] & [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:		Properties Minimize to system trav	
[Server] → [Port number of the server socket] Set the Ethernet socket number for HMI to connect range goes from 1 to 65535 and 8005 is the default [Server] → [User name] & [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:		Detailed message	
[Server] → [User name] & [Password] Set the user name and password to restrict that only authorized HMI can send requests to EasyPrinter. [Naming Convention for HMI Folder] EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:		Le out number of the serve	-
<b>[Naming Convention for HMI Folder]</b> EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:	Set the Ethe range goes	ernet socket number for HM from 1 to 65535 and 8005 is	l to connect t s the default
EasyPrinter creates different folders to store files (e hardcopy bitmap files, backup files) from different H There are two ways to name the folders:	Set the Ether range goes • [Server] → Set the use authorized I	ernet socket number for HM from 1 to 65535 and 8005 is [User name] & [Password r name and password to res HMI can send requests to Ea	I to connect t s the default ] trict that only asyPrinter.
hardcopy bitmap files, backup files) from different H There are two ways to name the folders:	Set the Ether range goes [Server] → Set the use authorized I [Naming Co	ernet socket number for HM from 1 to 65535 and 8005 is [User name] & [Password r name and password to res HMI can send requests to Ea Donvention for HMI Folder]	I to connect t the default trict that only asyPrinter.
There are two ways to name the folders:	Set the Ether range goes <b>[Server] →</b> Set the use authorized I <b>[Naming C</b> EasyPrinter	ernet socket number for HM from 1 to 65535 and 8005 is [User name] & [Password r name and password to res HMI can send requests to Ea povention for HMI Folder] creates different folders to s	I to connect to the default I trict that only asyPrinter.
	Set the Ether range goes [Server] → Set the use authorized I [Naming Coll EasyPrinter hardcopy bi	ernet socket number for HM from 1 to 65535 and 8005 is [User name] & [Password r name and password to res -IMI can send requests to Ea <b>Drivention for HMI Folder</b> ] creates different folders to s tmap files, backup files) from	I to connect to the default trict that only asyPrinter.

EasyPrinter names the folder after the IP address of the HMI sending the request. (i.e. [Prefix] + [IP address]) Please see the following illustration:



#### b. Use HMI name

WEINTEK

EasyPrinter names the folder after the name of the HMI sending the request. (i.e. [Prefix] + [HMI name])

#### [Properties] → [Minimize to system tray]

Select this option to minimize EasyPrinter to system tray instead of task bar. Users can double-click the icon in system tray to restore the EasyPrinter window.

## • [Properties] → [Detailed message]

Select this option to display more detailed messages about events in the message window.



#### a. Print-Out

Select this option to inform EasyPrinter to print out the hardcopy result with specified printers.

b. Save-to-File



Select this option to inform EasyPrinter to convert the hardcopy result into a bitmap file and save it in the specified directory. Users can find the bitmap files at:

```
[Specified Path] →
[HMI Folder] →
yymmdd_hhmm.bmp
```

For example, when a hardcopy request is given at 17:35:00 12/Jan/2009, the bitmap file will be named "090112\_1735.bmp". And if there is another bitmap file generated in the same minute, it will be named "090112\_1735\_01.bmp" and so on.

VT8xxx Remote P	Printer Server Settings
General	Output
Hardcopy	Backup files in:
Backup	D:\MT8000
buckup	When target file has existed:
	Overwrite it. (The content will be destroyed)
	Append .BAK to the file name.
	Convert Batch File
	Enable
	D:\MT8000\convert2csv.def 🕞
<b>[Output]</b> EasyPrinte	r stores the backup files to the specified p
[Output] EasyPrinte	r stores the backup files to the specified p
[Output] EasyPrinte For Event-I	r stores the backup files to the specified p Log historical data files:
[Output] EasyPrinte For Event-I [Specified F	r stores the backup files to the specified pa Log historical data files: Path] →
[Output] EasyPrinte For Event-I [Specified F [HMI	r stores the backup files to the specified p Log historical data files: Path] → Folder] →
[Output] EasyPrinte For Event-I [Specified F [HMI	r stores the backup files to the specified p Log historical data files: Path] → Folder] → [eventlog] →
[Output] EasyPrinte For Event-I [Specified F [HMI	r stores the backup files to the specified pa Log historical data files: Path] → Folder] → [eventlog] → EL_yyyymmdd.evt
[Output] EasyPrinte For Event-I [Specified F [HMI	r stores the backup files to the specified pa Log historical data files: Path] → Folder] → [eventlog] → EL_yyyymmdd.evt ampling historical data file:
[Output] EasyPrinte For Event-I [Specified F [HMI] For Data-S [Specified F	r stores the backup files to the specified pa Log historical data files: Path] → Folder] → [eventlog] → EL_yyyymmdd.evt ampling historical data file: Path] →



[Folder name of the Data-Sampling
object]→
yyyymmdd.dtl
For Recipe:
[Specified Path] →
[HMI Folder] →
[recipe] →
recipe.rcp or recipe_a.rcp
[Convert Batch File]
Select [Enable] and assign a Convert Batch File for
automatically converting uploaded history files to CSV or
MS Excel format. Please refer to the next section for the
details of Convert Batch File.

## NOTE

•

Users can assign HMI names from LW9032 to LW9039.

• EasyPrinter names the folder after IP address if HMI name is not set.



## 26.4 Convert Batch File

EasyPrinter provides a mechanism for converting the uploaded Data-Sampling and Event-Log history files stored in binary mode to CSV files automatically. Users requesting this function have to prepare a Convert Batch File to provide EasyPrinter with the information of how to convert the history files.



As shown in the illustration above, the conversion is actually carried out by EasyConverter. EasyPrinter simply follows the criteria in Convert Batch File and activates EasyConverter with proper arguments to achieve the conversion.

#### NOTE

- EasyConverter is another Win32 application converting history data into CSV or MS Excel (\*.xls) files. Users can find it in the EasyBuilder 8000 installation directory.
- Users requesting this function must ensure EasyPrinter and EasyConverter are placed in the same directory.

## 26.4.1 The Default Convert Batch File

The following is the default Convert Batch File included in the EasyBuilder 8000 software package:

#### The default Convert Batch File (convert2csv.def)

- 1: "dtl", "EasyConverter /c \$(PathName)"
- 2: "evt", "EasyConverter /c \$(PathName)"

There are two lines of text in the file. Each line has two arguments separated by a comma and forms a criterion of how to deal with a specific type of files, e.g. Data-Sampling and Event-Log history files. The first argument specifies the extension name for the type of the files to be processed and the second one specifies the exact command to execute in console mode. Please note "\$(PathName)" is a key word to tell EasyPrinter to replace it with the real name of the backup file in conversion. For example, if a Data-Sampling history file named 20090112.dtl is uploaded and stored, EasyPrinter will send out the following command to a console window:

EasyConverter /c 20090112.dtl

And then the CSV file named 20090112.csv is created.

Therefore, the criteria of the default Convert Batch File are:

- 1. Convert all Data-Sampling history files (\*.dtl) into CSV files.
- 2. Convert all Event-Log history files (\*.evt) into CSV files.

NOTE	•	Actually, the "\$(PathName)" in the second argument stands for the full path
		name of the file. In the previous case, EasyPrinter replaces it with:
		[Specified Path] \ [HMI Folder] \ [datalog] \
		[Folder name of the Data-Sampling object] \ 20090112.dtl
	•	EasyPrinter interprets the Convert Batch File on a line basis, i.e. each line
		forms a criterion.
	•	Any two arguments should be separated by a comma.
	•	Every argument should be put in double quotes.
	•	Do not put any comma inside an argument.
	•	For further information about how to use EasyConverter, please refer to the
		"chapter25 Easy Converter".

# 26.4.2 Specialized Criteria

Sometimes users may need a special handling for the files uploaded from a specific HMI. Here is an example:

Specialized Criterion for the HMI with IP = 192.168.1.26 3: "dtl", "EasyConverter /c \$(PathName)", "192.168.1.26"



Or users can also specify the HMI with its name.

- Specialized Criterion for the HMI with name = Weintek\_01
- 4: "dtl", "EasyConverter /c \$(PathName)", "Weintek\_01"

Or in the case of needing special handling for different Data-Sampling history files.

Specialized Criterion for the Data-Sampling object's folder name = Voltage

5: "dtl", "EasyConverter /s Voltage.lgs \$(PathName)", "\*", "Voltage"

The 5<sup>th</sup> criterion can only be performed on the history files uploaded from the **[Data Sampling]** objects with the folder name "Voltage". The 3<sup>rd</sup> argument ("\*") indicates this criterion accepts the qualified Data-Sampling files from any HMI. Users can also change the 3<sup>rd</sup> argument to "192.168.1.26", "192.168.1.\*", HMI name, etc. for narrowing the target HMI.

# 26.4.3 The Format of a Convert Batch File

No	Argument	Description	
1	File Type	This argument specifies the extension name of the	
		uploaded files this criterion targets. (e.g. "dtl" for	
		Data-Sampling history files, "evt" for Event-Log	
		history files)	
2	Command Line	The exact command EasyPrinter sends to a	
		console window if the uploaded file is qualified.	
3	a. HMI IP address	This argument specifies the HMI this criterion	
	b. HMI name	targets.	
4	Condition 1	<ul> <li>If the file type is "dtl"</li> </ul>	
		This argument specifies the folder name of the	
		[Data Sampling] objects this criterion targets.	
		• Others	
		No use.	
5	Condition 2	No use. (reserved for further use)	

The following table explains all arguments in a criterion.



# 26.4.4 The Order of Examining Criteria

EasyPrinter examines criteria in ascending order every time a file is uploaded. Once the file is qualified for a criterion, it stops the examination and starts over for next file. Therefore, users should place the criteria with more specification upward in the Convert Batch File and place the less-specific criteria downward. Take the 5 criteria mentioned in the previous sections for example, the correct order is:

Correct order for the previous criteria
'dtl", "EasyConverter /s Voltage.lgs \$(PathName)", "*", "Voltage"
'dtl", "EasyConverter /c \$(PathName)", "EasyView"
'dtl", "EasyConverter /c \$(PathName)", "192.168.1.26"
'dtl", "EasyConverter /c \$(PathName)"
'evt", "EasyConverter /c \$(PathName)"



# Chapter 27 EasySimulator

EasySimulator enables users to perform Online/Offline simulation without installing EayBuilder software. To achieve that, users have to prepare the following files in one folder.

# 27.1 Prepare Files



- 11. gui.exe
- 12. xob\_pos.def

Users can find all the above files in EasyBuilder installation directory, which means users have to install EasyBuilder software package on a PC and copy the files to the target PC.





# 27.2 Modify the Content of xob\_pos.def

**Step 1.** Open xob\_pos.def using a text editing tool (e.g. Notepad) and set the contents correctly.

Eile Edit	s - Notepad Format View Help	
"0" "c:\Eas "c:\Eas	//operation mode, 0 : off-line, 1 : on-line Simulator" // define the directory of com.exe and gui.exe Simulator\MT8000_Demo_800x600.xob" //define the directory of xc	b file
2		5

Line No.	Description
1	["0"]
	Perform Offline simulation
	["1"]
	Perform Online simulation
2	Specify the full path where the files (e.g. com.exe, gui.exe,
	EasySimulator.exe, etc.) locate.
3	Specify the full path of the project file (*.xob)

Step 2. Double click EasySimulator.exe to start the simulation.



**Step 3.** ON-Line/OFF-Line simulation is displayed on the screen.







# Chapter 28 Multi-HMI Intercommunication (Master-Slave Mode)

Multi-HMI intercommunication means that HMI uses COM port to connect with a remote HMI, and read/write data from/to PLC connected to remote HMI as below:



Above shows the PLC is connected with HMI 1, and HMI 1 is connected with HMI 2 via COM port, so that HMI 2 can control the PLC through HMI 1.

An example describes how to use EB8000 to create projects used on HMI 1 and HMI 2.



# 28.1 How to Create a Project of Master HMI

The following is the project content of HMI 1 in [System Parameter Settings] / [Device].

System Parameter Settings								
	Font Extended Memory Printer/Backup Server						er	
	Device	Model	G	eneral	System	Setting Sec	urity	
]	Device list :							
	No.	Name	Location	Device typ	e	Interface	I/F Pro	
	Local HMI	Local HMI	Local	MT6070iH/	MT8070	Disable	N/A	
	Local PLC 1	FATEK FB Series	Local	FATEK FB	Series	COM1(9600,E,7,1)	RS232	
	Local Server	Master-Slave Se	Local	Master-Slave	e Se	COM2(115200,E,8,1)	R\$232	

- 1. Due to COM 1 of HMI 1 connects PLC, the device list must include **[Local PLC 1]** in this case is "FATEK FB Series". The communication parameter must be set correctly.
- 2. Due to COM 2 of HMI 1 is used to receive commands from HMI 2; users must add a new device [Master-Slave Server] for setting communication properties of COM 2. Picture above shows the parameters of COM 2 are "115200, E, 8, 1", and uses RS232. These parameters are not required to be the same as PLC, but the "data bits" must set to 8. In general, a higher baud rate for COM 2 is recommended for communication more efficient.



# 28.2 How to Create a Project of Slave HMI

The following is the project content of HMI 2 in [System Parameter Settings] / [Device].

System Parameter Settings								
	Font Extended Memory Printer/Backup Server							
De	Device Model		General	System Setting	Security			
Device	e list :							
No.		Name	Location	Device type	Interface			
Local	Local HMI Local HMI		Local	MT6070iH/MT8070	Disable			
*Rem	note PLC 1	FATEK FB Series	COM 1 (master-slave	FATEK FB Series	COM1(115200,E,8			

Due to the PLC that HMI 2 reads from is connected with HMI 1, thus HMI 2 views PLC as a remote device. Therefore, it is necessary to add a [\*Remote PLC 1] into the device list and in this case is "FATEK FB Series". The way to create [\*Remote PLC 1] is described below:

## Step 1

Create a new device"FATEK FB Series" for [PLC type]. [PLC default station no.] must be the same as the connected PLC.

PLC type: FATEK FB Series	<b>v</b>
V.1.10, FATEK_FB.so	
PLC I/F : RS-232	PLC default station no 1

## Step 2

Correctly set the parameters. COM 1 of HMI 2 connects with COM 2 of HMI 1, so they both must have the same communication parameters and interfaces, ignoring the PLC parameters. As below, COM 2 of HMI 1 and COM 1 of HMI 2 use RS232 and the parameters are [115200, E, 8, 1].

20



PLC type : FATEK FB Series	
V.1.10, FATEK_FB.∞	
PLC I/F	COM Port Settings
COM: COM1	
	COM : COM 1 🛛 🗸
	Baud rate : 115200 💌
	Data bits : 8 Bits 💌
Interval of block pack (word	Parity : Even 💙
Max. read-command size (word	Stop bits : 1 Bit 💌
Mana and a second star Annual	

## Step 3

Since HMI 2 views PLC a remote device, here we change [Location] to [Remote], and select [COM port] to connect remote HMI (HMI 1).

	Location	⊖ HMI Remot		PLC	ettings P	COM 1	(master-slave mode)	
	PLC type :	FATE	IP Addre	ss Setti	ngs			
		V.1.10	◯ Ether	net	OCOM por	t (use r	naster-slave protocol)	
	PLC I/F :	RS-23						
	COM :	СОМІ						
		🗌 U se				Oł	Cancel	
Devic	e list :							
No.		Name	;	Locati	on		Device type	Interface
Loca	1 HMI	Local H	IMI	Local			MT6056T/MT8056T	Disable
*Ren	note PLC 1	FATER	(FB Series	COM 1	(master-slave r	node)	FATEK FB Series	COM1(115

After completing all settings described above, users can find a new device named [\*Remote PLC 1] in the [device table]. This device has the "\*" symbol means that HMI uses a COM port (not Ethernet) to control a remote PLC via other HMI.

Users can check local registers of HMI to view the communication status. ([\*Remote PLC1] uses same registers as [Local PLC1])



Тад	Description	
	When ON, auto. connection with PLC (COM 1) when disconnected.	
When OFF, ignore disconnection with PLC.		
	When ON, auto. connection with PLC (COM 2) when disconnected.	
LD-9151	When OFF, ignore disconnection with PLC.	
	When ON, auto. connection with PLC (COM 3) when disconnected.	
LD-9152	When OFF, ignore disconnection with PLC.	

Тад	Description
	These local registers indicate the connection states with PLC (through COM1)
	(Inrough COMT).
	LB9200 Indicates the connection state with PLC (station no. 0), and
LB-9200~	LB9201 indicates the connection state with PLC (station no. 1) and
LB-9455	so on.
	When ON, indicates connection state is normal.
	When OFF, indicates disconnection with PLC.
	Set ON again, the system will then try to connect with PLC.
	These local registers indicate the connection states with PLC
	(through COM2).
	LB9500 indicates the connection state with PLC (station no. 0), and
LB-9500~	LB9501 indicates the connection state with PLC (station no. 1) and
LB-9755	so on.
	When ON, indicates connection state is normal.
	When OFF, indicates disconnection with PLC.
	Set ON again, the system will then try to connect with PLC.
	These local registers indicate the connection states with PLC
	(through COM3).
	LB9800 indicates the connection state with PLC (station no. 0), and
LB-9800~	LB9801 indicates the connection state with PLC (station no. 1) and
LB-10055	so on.
	When ON, indicates connection state is normal.
	When OFF, indicates disconnection with PLC.
	Set ON again, the system will then try to connect with PLC.



# 28.3 How to Connect MT500 Project of Slave HMI

Allowing MT500 using Master-Slave protocol to read MT6000/8000 Local data and data of PLC connected with MT6000/8000.

MT8000 Settings

## Step 1

Select "Master-Slave Server" driver and click [Settings...]. If a PLC is connected, follow the original settings.

Font         Extended Memory         Printer/Backup Server           Device         Model         General         System Setting         Security           Device list :         No.         Name         Location         Device type         Interface         I/F Protoc           Local HMI         Local         MT6070iH/MT8070         Disable         N/A           Local Server         Master-Slave Se         Local         Master-Slave Se         COM1(115200,E,8,1)         RS232		FIGT NOTIL	ч <b>в</b> »						
Device     Model     General     System Setting     Security       Device list :     No.     Name     Location     Device type     Interface     I/F Protoc       Local HMI     Local     MT6070iH/MT8070     Disable     N/A       Local Server     Master-Slave Se     Local     Master-Slave Se     COM1(115200,E,8,1)     RS232	Font			Extended	( Memory		Printer/Backup :	server	
Device list :         No.       Name       Location       Device type       Interface       I/F Protoc         Local HMI       Local       MT6070iH/MT8070       Disable       N/A         Local Server       Master-Slave Se       Local       Master-Slave Se       COM1(115200,E,8,1)       RS232	Device	]	Model		General	Sys	tem Setting	Security	
No.         Name         Location         Device type         Interface         I/F Protoc           Local HMI         Local         MT6070iH/MT8070         Disable         N/A           Local Server         Master-Slave Se         Local         Master-Slave Se         COM1(115200,E,8,1)         RS232	Device list :								
Local HMI         Local         MT6070iH/MT8070         Disable         N/A           Local Server         Master-Slave Se         Local         Master-Slave Se         COM1(115200,E,8,1)         RS232	No.	Name		Location	Device type		Interface	I/F Protoc	
Local Server Master-Slave Se Local Master-Slave Se COM1(115200,E,8,1) RS232	Local HMI	Local HM	I	Local	MT6070iH/M	T8070	Disable	N/A	
	Local Server	Master-Sla	ve Se	Local	Master-Slave S	Se	COM1(115200,E,8,1	) RS232	

## Step 2

Select RS232 and click [Settings...].

Device Properties	
Name :	Master-Slave Server
Location :	Local Settings
PLC type :	Master-Slave Server
	V.1.00, MASTER_SLAVE.so
PLC I/F :	RS-232
COM :	COM1 (115200,E,8,1) Settings
	OK Cancel
	OK Cancel

## Step 3

Fill in MT500 PLC ID No. in Parameter 1 (Refer to MT500 settings).

COM Port Settings			
COM :	СОМ 1 🛛 🔽	Timeout (sec) :	1.0 💌
Baud rate :	115200 💌	Turn around delay (ms) :	0
Data bits :	8 Bits 💌	Send ACK delay (ms) :	0
Parity :	Even 💌	Parameter 1 :	10
Stop bits :	1 Bit 💌	Parameter 2 :	0
		Parameter 3 :	0
		ОК	Cancel



#### ➢ MT500 Settings

## Step 1

In EB500/System Parameter Setting, set Multiple HMI: Slave, HMI-HMI link speed: 115200

*Note:* Set the same Baud Rate in MT500 and MT8000.

System Parameter Set	tting		X
PLC General In	dicator Security Edit	tor Hardware Aux.	
PLC type :	MITSUBISHI FX0n/F	X2 💌	
HMI model :	MT510T/MT508T (64	10 x 480) 💌	
PLC I/F port :	RS-485 4 W 💉	Baud rate : 9600	~
Data bits :	7 Bits 💌	Parity : Even	~
Stop bits :	1 Bit 👻		
Parameter 1 :	0	Tum around delay : 0	
Parameter 3 :	0	Parameter 4 : 0	
Parameter 5 :	0	Parameter 6 : 0	
HMI station no. :		PLC station no. : 0	~
Multiple HMI :	Slave 💌	HMI-HMI link speed : 115200	~
Connect I/F :	Serial 💌		
Local I	Paddress : 0 .	0 • 0 • 0	
Server I	Paddress : 0 ·	0 • 0 • 0	
Subnetw	ork mask : 📋 🛛 .	0 0 0	
Default route I	Paddress : 0 .		
PLC time out cons	stant (sec) : 3.0	PLC block pack : 3	•
	0	K Cancel	Help

## Step 2

Double click on PLC Address View.exe to check PLC ID No. and fill in Parameter 1 of MT8000.

PLCAddressView						X
MITSUBISHI FX0n/FX2						-
PLC/Address Type ID	Bit/Word	Address Type	Addressing Format	Max	Min	~
MITSUBISHI FX0n/FX2	PLC ID=10					
0	Bit(HMI)	LB	ddd	9999	0	
1	Bit(PLC)	X	000	377	0	
2	Bit(PLC)	Y	000	377	0	
3	Bit(PLC)	М	ddd	9999	0	-
4	Bit(PLC)	Ī	ddd	255	0	
5	Bit(PLC)	C	ddd	255	0	
8	Word(HMI)	LW	ddd	9999	0	
9	Word(PLC)	TV	ddd	255	0	
10	Word(PLC)	CA	ddd	199	0	
11	Word(PLC)	D	ddd	9999	0	
12	DWord(PLC)	CV2	ddd	255	200	
13	Word(PLC)	SD	ddd	9999	8000	
121	Word(HMI)	RWI	ddd	32767	0	
120	Bit(HMI)	RBI	ddd(h)	2047f	0	
140	Bit(HMI)	RB	ddd(h)	2047f	0	
141	Word(HMI)	R₩	ddd	65535	0	
160	Bit(HMI)	Ms_RB	ddd(h)	4095f	0	
161	Bit(HMI)	Ms_LB	ddd	9999	0	
180	Word (HMI)	Ms_RW	ddd	65535	0	~
					Exit	

Note: There will always be a PLC selected in EB500 system parameters setting, in this case, even to read/write MT8000 Local Data only, the ID of selected PLC in EB500 system parameters must also be filled in EB8000 parameter 1.

Also, when using S7-200, S7-300 drivers, since in EB500 the high and low bytes are sent in reverse order, this will cause MT500 to misread MT8000 Local data.

## Step 3

COM Port used: RS232, connect it with RS232 of MT8000, the communication is then enabled.

Bit/Word	MT500	MT8000	Range	Memo
В	Ms_RB	RW_Bit	dddd:0~4095 (h): 0~f	
В	Ms_LB	LB	dddd:0~9999	
W	Ms_RW	RW	ddddd:0~65535	
W	Ms_LW	LW	dddd:0~9999	

Device address:



# **Chapter 29 Pass-Through Function**

The pass-through function allows the PC application to control PLC via HMI. In this case the HMI acts as a converter.

The pass-through function provides two modes: **[Ethernet]** and **[COM port]**. Click **[Pass-through]** in **[Project Manager]** will open a setting dialogue.



Pass-through	
	port
Virtual COM Port (PC <-> F	LO)
Please in:	stall weintek virtual serial port driver



## 29.1 Ethernet Mode

## [How to install virtual serial port driver]

Before using [Ethernet] mode, please check whether Weintek virtual serial port driver is installed as described below:

# If [Virtual COM port (PC<->PLC)] displays [Please install weintek virtual serial port driver], please click [Install].

Pass-through		×
⊙ Ethernet	○ COM port	
Virtual COM F	Port (PC <-> PLC)	
	Please install weintek virtual serial port driver	
PLC Connecti	on Port (HMI <-> PLC)	
HMLIP	· · · · ·	
Install	Uninstall Apply	

If the dialogue below pops up during installation, please click [Continue Anyway].

Hardwa	re Installation
1	The software you are installing for this hardware: Weintek Virtual Serial Port has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.) Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.
	Continue Anyway STOP Installation



After process is completed, the virtual COM port is displayed as follow.

Virtual COM Port (PC <-> PLC)		
	СОМЗ	

## **29.1.1 How to Change the Virtual Serial Port**

Open [System Properties] -> [Device Manager] to check if the virtual serial port is installed successfully.



If users want to change the number of virtual serial port, please click [Weintek Virtual Serial Port] to open [Port Settings] / [Advanced...], as follows:



For example, user changes virtual serial port from COM 3 to COM 9.

WE!NTEK





Weintek Virtual Serial Port (COM3) Properties 🛛 🔹 💽
General Port Settings Driver Details
Bits per second: 9600
Advanced Settings for COM3
Use FIFO buffers (requires 16550 compatible UART) Select lower settings to correct connection problems. Select higher settings for faster performance.
Receive Buffer: Low (1)
Transmit Buffer: Low (1)
COM Port Number: COM9

Select COM 9 and click **[OK]**, the virtual serial port will be changed to COM 9.



It can be found that the virtual COM port be changed to COM 9 in [Project Manager].

Virtual COM Port (PC <-> PLC)			
	СОМ9		


# **29.1.2 How to Use Ethernet Mode**

After installing virtual serial port driver, users should follow four steps to use Ethernet mode of pass-through.

#### Step 1

Set IP of the HMI connected with PLC. For example, HMI IP is 192.168.1.206

#### Step 2

Assign serial port properties of the port connects HMI with PLC. For example, COM2 (use RS232) is used to connect PLC.

#### Step 3

Click [Apply], and these settings will be updated.

PLC	COM 2 Ethernet PC Application Weintek Virtual Serial Port (COM7) A
	Virtual COM Port (PC <-> PLC)
в	COM7
-	PLC Connection Port (HMI <-> PLC)
	HMI IP: 192.168.1.206
	COM 2 🔽 RS232 🔽
	Install Uninstall Apply C

#### Step 4



In the PC application, the number of the serial port must be the same as the virtual one. For example, using a Mitsubishi application, if the virtual serial port is COM 7, please open [PC side I/F Serial setting] / [COM port] to select COM 7, as follows:



After completing all settings, when users execute PLC application on PC, the HMI will be switched automatically to pass-through mode (the communication between HMI and PLC will be suspended this moment and it will be resumed if the application closes), as follows:



At this moment the application is controlling PLC directly via virtual serial port.



# 29.2 COM Port Mode



#### Source COM Port

The port is used to connect HMI with PC.

#### **Destination COM Port**

The port is used connect HMI with PLC.

When using **[COM port]** mode of pass-through, users should correctly set the properties of source COM port and Destination COM port.

# 29.2.1 Settings of COM Port Mode

There are two ways to enable **[COM port]** mode of pass-through function.

- (1) Use Project Manager
- (2) Use system registers LW-9901 and LW-9902

LW-9901: pass-through source COM port (1~3: COM1~COM3)

LW-9902: pass-through destination COM port (1~3: COM1~COM3)

**Note**: When finish using Pass Through function, users should click [Stop Pass-through] to disable it so that HMI can start to communicate with PLC

#### Start pass-through in project manager.

Click [Pass-through] button in Project Manager to set the communication parameters.

Pass-through		×
◯ Ethernet	⊙ COM port	
HMUP:	192.168.1.37	
	Get HMI Communication Parameters	
F	HMI work mode : Unknown	
Source COM Po	rt (PC -> HMI)	
	COM 1 RS232	~
Baud rate :	9600 V Data bits : 8 Bits	~
Parity :	Even Stop bits : 1 Bit	~
Destination CON	/ Port (HMI -> PLC)	
	COM 2 RS232	~
Baud rate :	9600 💌 Data bits : 8 Bits	~
Parity :	Even Stop bits : 1 Bit	~
Start Pass-thro	ough Stop Pass-through	
	Exit	

#### [HMI IP]

Assign HMI IP address.

#### [Get HMI Communication Parameters]

For getting the settings of source and destination COM port. The parameters come from reserved addresses detailed as follows.

|--|

LW-9901 (Source COM port)	1 : COM 1	2 : COM 2	3 : COM 3
LW-9902(Destination COM	1 : COM 1	2 : COM 2	3 : COM 3
port)			

## COM 1 mode settings

LW-9550 (PLC I/F)	0 : RS232	1 : RS485/2	W	2 : RS48	35/4W
LW-9551 (baud rate)	0 : 4800	1:9600	2 : 1920	0	3 : 38400
	4 : 57600	5 : 1152	200		
LW-9552 (data bits)	7 : 7 bits	8 : 8 bits			
LW-9553 (parity)	0 : none	1 : even	2 : odd		



EasyBuilder8000 User's Manual

LW-9554 (stop bits)	1 : 1 bit	2 : 2 bits

## COM 2 mode settings

LW-9555 (PLC I/F)	0 : RS232	1 : RS485/2	W	2 : RS48	35/4W
LW-9556 (baud rate)	0 : 4800	1:9600	2 : 1920	0	3 : 38400
	4 : 57600	5 : 1152	200		
LW-9557 (data bits)	7 : 7 bits	8 : 8 bits			
LW-9558 (parity)	0 : none	1 : even	2 : odd		
LW-9559 (stop bits)	1 : 1 bit	2 : 2 bits			

#### COM 3 mode setting

LW-9560 (PLC I/F)	0 : RS232	1 : RS485/2	W	
LW-9561 (baud rate)	0:4800	1:9600	2 : 19200	3 : 38400
	4 : 57600	5 : 1152	200	
LW-9562 (data bits)	7 : 7 bits	8 : 8 bits		
LW-9563 (parity)	0 : none	1 : even	2 : odd	
LW-9564 (stop bits)	1 : 1 bit	2 : 2 bits		

Click **[Get HMI Communication Parameters]** to update HMI current states and communication parameters.

## 29.2.2 HMI Work Mode

There are three work modes in the pass-through function,

Mode	Description
Unknown	Before getting the settings of HMI, the work mode is displayed
	"Unknown".
Normal	After getting the settings of HMI, if work mode displays "Normal"
	PC can't control PLC via HMI.
Pass-through	HMI is working on pass-through state; at this time, the PC
	application can control PLC via source com port.

#### [Source COM Port] \ [Destination COM Port]



The communication parameters of source and destination COM port are displayed in these two areas. The settings will be used when **[Start pass-through]** is clicked.

# The "Baud rate", "Data bits", "Parity", and "Stop bits" of [Source COM Port] and [Destination COM Port] have to be the same.

[Source COM Port] connects PC, so select RS232 mode; [Destination COM Port] connects PLC, so settings depend on the PLC requirements.

The illustration below shows the setting when HMI connects SIEMENS S7/200.

The HMI COM 1 (RS232) connects PC, COM 3 (RS485 2W) connects PLC. The communication parameter of PLC is "9600, E, 8, 1". Before starting pass-through, users must set the parameters in MTP project and download the project to HMI.

Device Properties
Name : SIEMENS S7/200
◯ HMI
Location : Local Settings
PLC type : SIEMENS S7/200
V.1.90, SIEMENS, S7_200.so
PLC I/F : RS-485 2W PLC default station no. : 2
COM : COM3 (9600,E,8,1)
Use broadcast command
Interval of block pack (words) : 5
Max. read-command size (words) : 32
Max. write-command size (words) : 32
OK Cancel

After the project is downloaded to HMI, open the same project and change the PLC I/F and COM port to COM 1 RS232 (PC uses COM 1 to connect HMI) as follows:



Device Properties
Name : SIEMENS \$7/200
○ HMI ● PLC Location : Local ▼ Settings
PLC type : SIEMENS \$7/200
V.1.90, SIEMENS_S7_200.so PLC MF: RS-232  PLC default station no. : 2 COM : COM1 (9600,E.8.1) Settings
Use broadcast command
Interval of block pack (words) : 5
Max. read-command size (words) : 32 🔍
Max. write-command size (words) : 32
OK Cancel

After that, press **[Pass-through]** to assign HMI IP address; for example, 192.168.1.37. Finally, press **[Get HMI Communication Parameters]**, as follows:

HMLIP:	192.168.1.3	37		~	
	Get HMI	Communic	ation Parameters	•	
	HMI work m	ode : Norm	al		
Bource COM Po	ort (PC -> HM	11)			
	COM 1	~		RS232	~
Baud rate :	9600	*	Data bits :	8 Bits	~
Parity :	Even	~	Stop bits :	1 Bit	~
Destination COM	M Port (HMI -	-> PLC)			
	COM 3	~		RS485 2W	~
Baud rate :	9600	*	Data bits :	8 Bits	~
Parity :	Even	~	Stop bits :	1 Bit	~
		16			



Press **[Start Pass-through]** and HMI work mode is switched into "Pass-through". Users can execute on-line simulation. Now PC application can control PLC via HMI, and HMI is acting as a converter at this moment.

Note: The communication between HMI and PLC will be paused when pass-through is active. If users want to resume communication between HMI and PLC, please press **[Stop Pass-through]** to disable this function.



# 29.3 Using System Reserved Addresses to Enable Pass-Through Function

Other way to enable pass-through is to use LW-9901/LW-9902 to set source COM port and destination COM port directly. When the values of LW-9901 and LW-9902 match conditions as below, HMI will start pass-through automatically:

- a. The values of LW-9901 and LW-9902 have to be 1 or 2 or 3 (1: COM 1, 2: COM 2, 3: COM 3).
- b. The values of LW-9901 and LW-9902 should not be the same.

**Note**: If users want to stop pass-through, just change the values of LW-9901 and LW-9902 to 0.

If users need to change the communication parameters, just change the value in related reserved addresses and set ON to LB-9030, LB-9031 and LB-9032. HMI will be forced to accept new settings.

Tag	Description
LB-9030	Update COM1 communication parameters (set ON)
LB-9031	Update COM2 communication parameters (set ON)
LB-9032	Update COM3 communication parameters (set ON)



# **Chapter 30 Project Protection**

The copyright of program design must be protected. EB8000 supports protection function of project file to ensure users' design achievement.

## 30.1 XOB password

After project (MTP) is completed, users can compile the file to XOB format that can be downloaded to HMI. Users can set password to protect the XOB file in Compiling window.

A password must be input if users want to decompile the XOB file to MTP. (XOB password range: 0~4294901760)

Compiling	×
Project name : C:\Documents and Settings\Nicolas\桌面\MTP1.mtp	
XOB password :     Set     (used in decompiler)     Decompilation is prohibited	
Select the languages used on the HMI	
C:\Documents and Settings\Nicolas   C:\Documents and Settings\Nicolas   C:\Documents and Settings\Nicolas   C:\Documents and Settings\Nicolas   O error(s), 0 warning(s)   Object size : 56204 bytes   Font size : 314444 bytes   Picture size : 792988 bytes   Shape size : 842 bytes   Sound size : 36474 bytes   Macro size : 14 bytes   Total size : 1200966 bytes (1.15M)	
succeeded	~
Double click error messages to modify the attributes of relative objects !         Compile       Image: Build font files         Close	



# **30.2 Decompilation is prohibited**

If this box is ticked, the system will automatically deny **[XOB password]**. Furthermore, the XOB file can't be decompiled to MTP file.

Compiling	X
Project name: C:\Documents and Settings\Nicolas\桌面\MTP1.mtp	
XOB file name: C:\Documents and Settings\Nicolas\桌面\MTP1.xob	
Select the languages used on the HMI	
∠ Language 1	
C:\Documents and Settings\Nicolas\桌面\EB8000V410_100614_eng\font\MTP1\$0.ttf (Arial) C:\Documents and Settings\Nicolas\桌面\EB8000V410_100614_eng\font\MTP1\$1.ttf (Times New Roman)	^
0 error(s), 0 warning(s)	
Object size: 56204 bytesFont size: 314444 bytesPicture size: 792988 bytesShape size: 842 bytesSound size: 36474 bytesMacro size: 14 bytes	<b>2</b>
Total size : 1200966 bytes (1.15M)	
succeeded	<b>~</b>
Double click error messages to modify the attributes of relative objects !	
Compile Build font files	Close



# 30.3 Disable HMI upload function [LB9033]

EB8000 provides system reserved address LB9033. When this address is set ON, the HMI will disable upload function of XOB file and vice versa. HMI needs to be rebooted to active LB9033.

30       LB-9029 : save all recipe data to machine (set ON)       Local HMI         31       LB-9030 : update COM 1 communication parameters (set ON)       Local HMI         32       LB-9031 : update COM 2 communication parameters (set ON)       Local HMI         33       LB-9032 : update COM 3 communication parameters (set ON)       Local HMI         34       LB-9033 : disable HMI upload function (when ON)(i series only)       Local HMI         35       LB-9033 : disable HMI upload function (when ON)(i series only)       Local HMI         36       LB-9035 : HMI free space insufficiency alarm (when ON)       Local HMI         37       LB-9036 : CF free space insufficiency alarm (when ON)       Local HMI         38       LB-9037 : USB 1 free space insufficiency alarm (when ON)       Local HMI         39       LB-9038 : USB 2 free space insufficiency alarm (when ON)       Local HMI	Bit Bit Bit Bit Bit Bit	LB-9029 LB-9030 LB-9031 IB-9032 LB-9033 IB-9134	Read/Write Read/Write Read/Write Read/Write Read/Write
81       LB-9030 : update COM 1 communication parameters (set ON)       Local HMI         82       LB-9031 : update COM 2 communication parameters (set ON)       Local HMI         83       LB-9032 : update COM 3 communication parameters (set ON)       Local HMI         84       LB-9033 : disable HMI upload function (when ON)(i series only)       Local HMI         85       LB-9033 : disable HMI upload function (when ON)(i series only)       Local HMI         86       LB-9035 : HMI free space insufficiency alarm (when ON)       Local HMI         87       LB-9036 : CF free space insufficiency alarm (when ON)       Local HMI         88       LB-9037 : USB 1 free space insufficiency alarm (when ON)       Local HMI         89       LB-9038 : USB 2 free space insufficiency alarm (when ON)       Local HMI	Bit Bit Bit Bit Bit	LB-9030 LB-9031 IB-9032 LB-9033 IB-9034	Read/Write Read/Write Read/Write Read/Write
B2         LB-9031 : update COM 2 communication parameters (set ON)         Local HMI           IB-9032 : update COM 3 communication parameters (set ON)         Local HMI           IB-9033 : disable HMI upload function (when ON)(i series only)         Local HMI           IB-9033 : disable HMI upload function (when ON)(i series only)         Local HMI           IB-9033 : disable HMI upload function (when ON)(i series only)         Local HMI           IB-9035 : HMI free space insufficiency alarm (when ON)         Local HMI           IB-9035 : FMI free space insufficiency alarm (when ON)         Local HMI           IB-9036 : CF free space insufficiency alarm (when ON)         Local HMI           IB-9037 : USB 1 free space insufficiency alarm (when ON)         Local HMI           IB-9038 : USB 2 free space insufficiency alarm (when ON)         Local HMI	Bit Bit Bit Bit	LB-9031 1B-9032 LB-9033	Read/Write Read/Write Read/Write
IB-9032 : update COM 3 communication parameters (set ON)         Local HMI           IB-9033 : disable HMI upload function (when ON)(i series only)         Local HMI           IB-9033 : disable HMI upload function (when ON)(i series only)         Local HMI           IB-9034 : save event/data log to HMI, USB disk, CF card (set         Local HMI           IB-9035 : HMI free space insufficiency alarm (when ON)         Local HMI           IB-9036 : CF free space insufficiency alarm (when ON)         Local HMI           IB-9037 : USB 1 free space insufficiency alarm (when ON)         Local HMI           IB-9038 : USB 2 free space insufficiency alarm (when ON)         Local HMI	Bit Bit Bit	LB-9032 LB-9033	Read/Write Read/Write
4         LB-9033 : disable HMI upload function (when ON)(i series only)         Local HMI           5         LB-9034 : save event/data log to HMI, USB disk, CF card (set         Local HMI           6         LB-9035 : HMI free space insufficiency alarm (when ON)         Local HMI           7         LB-9036 : CF free space insufficiency alarm (when ON)         Local HMI           8         LB-9037 : USB 1 free space insufficiency alarm (when ON)         Local HMI           9         LB-9038 : USB 2 free space insufficiency alarm (when ON)         Local HMI	Bit Bit	LB-9033	Read/Write
5         LB-9034 : save event/data log to HMI, USB disk, CF card (set         Local HMI           6         LB-9035 : HMI free space insufficiency alarm (when ON)         Local HMI           7         LB-9036 : CF free space insufficiency alarm (when ON)         Local HMI           8         LB-9037 : USB 1 free space insufficiency alarm (when ON)         Local HMI           9         LB-9038 : USB 2 free space insufficiency alarm (when ON)         Local HMI	Bit	18-9034	
6         LB-9035 : HMI free space insufficiency alarm (when ON)         Local HMI           7         LB-9036 : CF free space insufficiency alarm (when ON)         Local HMI           8         LB-9037 : USB 1 free space insufficiency alarm (when ON)         Local HMI           9         LB-9038 : USB 2 free space insufficiency alarm (when ON)         Local HMI	Dit		Read/Write
7         LB-9036 : CF free space insufficiency alarm (when ON)         Local HMI           8         LB-9037 : USB 1 free space insufficiency alarm (when ON)         Local HMI           9         LB-9038 : USB 2 free space insufficiency alarm (when ON)         Local HMI	DIL	LB-9035	Read/Write
B LB-9037 ; USB 1 free space insufficiency alarm (when ON) Local HMI B LB-9038 ; USB 2 free space insufficiency alarm (when ON) Local HMI	Bit	LB-9036	Read/Write
9 LB-9038 : USB 2 free space insufficiency alarm (when ON) Local HMI	Bit	LB-9037	Read/Write
	Bit	LB-9038	Read/Write
D LB-9039 : status of file backup activity (backup in process if Local HMI	Bit	LB-9039	Read/Write
1 LB-9040 : backlight up (set ON) Local HMI	Bit	LB-9040	Read/Write
2 LB-9041 : (16bit) : reserved free space size (Mega bytes) Local HMI	Bit	LB-9041	Read/Write
3 LB-9042 ; acknowledge all alarm events (set ON) Local HMI	Bit	LB-9042	Read/Write
4 LB-9043 ; unacknowledged events exist (when ON) Local HMI	Bit	LB-9043	Read/Write
5 LB-9044 : disable remote control (when ON) Local HMI	Bit	LB-9044	Read/Write
5 LB-9045 : memory-man communication fails (when ON) Local HMI	Rit	I R-9045	Read/Write
			2



# 30.4 Project protection [Project Key]

User's project can be restrained to be executed only on specific HMI (for i series HMI only).

Please go to [System Parameters Settings]/ [General]/ [Project protection].



Users can use LW9046~LW9047 (32-bit) to set the **[HMI key]**. The value can't be read or written into these two registers even by remote HMI. While using this function, user can set the password (**[project key]** password range: 0~4294901750), and the XOB file can only be executed on specific HMI whose [HMI Key] is the same as [Project key]. If [Project Key] is different from [HMI key], the system will turn LB9046 ON. HMI needs to be rebooted while setting [HMI key] every time.





# 30.5 Project password [MTP file]

Users can set password to protect the MTP file in [System parameter] / [Security] tab.

Enabling this, password must be input if user wants to edit MTP file. (MTP password range: 1~4294967295)

🔲 Enable 🔒	ITP file
User 11	Password : 19854561 (range: 1 ~ 4294967295)
User 12 Enable	OK Cancel
oject password (M	TP file)
🗹 Enable	Setting

After setting, each time when try to open the project, a window pops up for password input.



Note: When using "Window Copy" function, if the source file is protected by MTP password, users have to input correct password for EasyBuilder8000 to execute window copy.



Source pro	oject :
C:\Docum	ents and Settings\Nicolas\桌面\666.mtp 🛛 🔗
Cor Ea	syBuilder8000
	Password : ***** (Limited to 3 times)
PLC *	Source project is protected with password.
Alar	
Sch	



# **Chapter 31 Memory Map Communication**

MemoryMap communication protocol is similar to IBM 3764R, it is used when memory data is with low variation. (High variation may cause MemoryMap overloading.) MemoryMap is used for communication between two devices. When setting the MemoryMap with two devices, one has to be set as Master, and another is Slave. In normal condition, Master and Slave do not communicate except when the assigned memory data in one of them has changed. Once data is identical the communication will stop.. So this is used for keeping the consistency of assigned part of data between two devices (Master and Slave) via corresponding registers.

The corresponding memory has the same property as MT8000's register MW(MB) from Master and Slave (The 1000 words MW(MB) are reserved for MemoryMap in MT8000 for communication.) The feature of memory: MB is correspondence with MW, according to the following list, MB0~MBf and MW0, MB10~MB1f and MW1..., they all indicate the same register.

Device name	Format	Range
MB	dddd(h)	dddd:0~9999 h:0~f(hex)
MW	dddd	dddd:0~9999

When using MemoryMap communication protocol, the master and slave have to use the same communication setting. The wiring diagram as follow:

RS232	
Master	Slave
TX(#)	RX(#)
RX(#)	TX(#)
GND(#)	GND(#)

RS485 (4W)	
Master	Slaver
TX+(#)	RX+(#)
TX-(#)	RX-(#)
RX+(#)	TX+(#)
RX-(#)	TX-(#)



GND(#) GND(#)

Note: # means being decided by PLC or controller.

The flowchart of communication as following:







#### Caution:

Flowchart 2 is available for slave but not master, STX is asking signal for communication, ACK is feedback signal, and NAK is busy signal.

There are two data formats, one is for MB and another is for MW:

For MB command			
Offset (byte)	Format	Description	
0	0x02	The operating sign to MB	
1	0x##	Address (Low byte)	
2	0x##	Bit Address (High byte)	
		For example:MB12=>1*16+2=18, is 0x12 and 0x00	
3	0x00( or 0x01)	The data of MB address.	
		(This is Bit, so has to be 0 or 1)	
4,5	0x10,0x03	Stop sign	
6	0x##	checksum, xor from 0 byte to fifth byte.	

For MW command			
Offset(byte)	Format	Description	
0	0x01	The operating sign to MW	
1	0x##	Address (Low byte)	
2	0x##	Bit Address (High byte)	
		If there is a 0x10 included in address, and insert a	



	0x10 after it, the byte will move to next position. For
	example: 0x10, 0x04 will become 0x10,0x10,0x04
0x##	Sending byte (The byte has to be even, due to
	operating for word). If byte is 0x10 then insert a
	0x10 after it, the byte will move to next position
0x##(L)	The data of initial address for corresponding
0x##(H)	address for 1,2 byte, n is byte of data, if data
0x##(L)…	includes 0x10 and then insert a 0x10, the sending
	byte number remains same, then n=n+1, and so
	on
0x10,0x03	End sign
0x##	checksum <sup>,</sup> Xor check-up and bytes in the front
	0x## 0x##(L) 0x##(H) 0x##(L) 0x10 <sup>,</sup> 0x03 0x##

Below is an example for observation process of communication. If Master has a 0x0a in MW3, according to this protocol, master will communicate with slave immediately, and slave will put the 0x0a in corresponding MW3, the procedure is as following:

Master sending STX(0x02h).

Slave receives STX(0x02h) from master, and sending ACK(0x06h) to master. Master receives ACK(0x06h) from slave.

Master sending 0x01,0x03,0x00,0x02,0x0a,0x00,0x10,0x03,0x19, as shown below:

Offset(byte)	Format	Description	
0	0x01	The operating sign for MW	
1	0x03	Address(Low byte)	
2	0x00	Bit Address (High byte)	
3	0x02	Sending byte (The byte has to be even, due to	
		MW3 is two byte).	
4,5	0x0a,	MW3 content is 0x0a <sup>,</sup> 0x00	
	0x00		
6,7	0x10,	End sign	
	0x03		
8	0x19	checksum <sup>,</sup>	
		0x01^0x03^0x00^0x02^0x0a^0x00^0x10^0x03=0x	
		19	

Slave received data from master and then sending ACK(0x06h). Master receives ACK(0x06h) from slave.



When finishing communication, master sending revised data of MW to slave, and slave changes the MW which corresponds to that of master. At this time, master and slave keep the same data in the same address.

Another example below, the address and data include 0x10; please notice the change in data format. Now, if we have 0x10 in MW16 in slave, according to this protocol, slave will communicate with master immediately, and master will put 0x10 in data of corresponding MW16, the procedure is as following:

Slave sending STX(0x02h)

Master receives STX(0x02h) from slave, and sending ACK(0x06h) to Slave.

Slave receives ACK(0x06h) from master

Slave sending data 0x01,0x10,0x10,0x00,0x02,0x10,0x10,0x00,0x10,0x03,0x10 as shown below:

Offset (byte)	Format	Description		
0	0x01	The operating sign to MW		
1	0x10	ddress(Low byte)		
2	0x10	Insert 0x10		
3	0x00	Bit Address (High byte)		
4	0x02	Sending byte (MW10 is two bytes)		
5	0x10	0x10 is low byte in MW10		
6	0x10	Insert 0x10		
7	0x00	0x00 in high byte		
8,9	0x10,	End sign		
	0x03			
10	0x10	checksum <sup>,</sup>		
		0x01^0x10^0x10^0x00^0x02^0x10^0x10^0x00^0x		
		10^0x03=0x10		

Master receives data from slave and sending ACK(0x06h) to slave.

Slave receives ACK(0x06h) from master.

When finishing communication, slave sending the address and content of MW to master, at this time, master changes data of MW corresponding to that of Slave, then master and slave keep the same data in the same address.





Below is an example for communication between two HMI via MemoryMap. First of all, create a new project in EasyBuilder

Edit/System Parameter Setting/PLC

	Memory Ma	p		
	O HMI	● PLC		
Location :	Local	<b>∼</b> [ s	ettings	
PLC type :	Memory Mag	p		
	V.1.00, MEM	IORY_MAP.so	li -	
PLC I/F	RS-232	~	PLC default station	no.: 0
COM		(00,E,0,1)	4	Setting
Port Settings	Lise broad	icast comman	đ	(c)
COM ·	COM 1	~	Timeout (sec) -	0.5
COM :	COM 1		Timeout (sec) :	0.5
COM : Baud rate :	COM 1 115200	~	Timeout (sec) : Turn around delay (ms) :	0.5
COM : Baud rate : Data bits :	COM 1 115200 8 Bits	~	Timeout (sec) : Turn around delay (ms) : Send ACK delay (ms) :	0.5
COM : Baud rate : Data bits : Parity :	COM 1 115200 8 Bits Even	× ×	Timeout (sec) : Turn around delay (ms) : Send ACK delay (ms) : Parameter 1 :	0.5
COM : Baud rate : Data bits : Parity : Stop bits :	COM 1 115200 8 Bits Even 1 Bit	× × ×	Timeout (sec) : Turn around delay (ms) : Send ACK delay (ms) : Parameter 1 : Parameter 2 :	0.5 0 0 0
COM : Baud rate : Data bits : Parity : Stop bits :	COM 1 115200 8 Bits Even 1 Bit	• • •	Timeout (sec) : Tum around delay (ms) : Send ACK delay (ms) : Parameter 1 : Parameter 2 : Parameter 3 :	0.5 0 0 0 0



	e: Memory M	ар		
	⊖ HMI	⊙ PLC		
Locatio	n : Local	*	Settings	
PLC typ	e : Memory Ma	ap		
	V.1.00, ME	MORY_MAP.s	2	
PLC I/	F : RS-232	~	PLC default statio	n no. : 0
		-	Timeout (sec) :	0.8
	Contraction of the	~	Timeout (sec) :	0.8 💙
сом :	COM 1		-	
COM : Baud rate :	COM 1 115200	~	Turn around delay (ms) :	0
COM : Baud rate : Data bits :	COM 1 115200 8 Bits	~	Turn around delay (ms) : Send ACK delay (ms) :	0
COM : Baud rate : Data bits : Parity :	COM 1 115200 8 Bits Even		Turn around delay (ms) : Send ACK delay (ms) : Parameter 1 :	0
COM : Baud rate : Data bits : Parity : Stop bits :	COM 1 115200 8 Bits Even 1 Bit	<ul> <li></li> <li><td>Turn around delay (ms) : Send ACK delay (ms) : Parameter 1 : Parameter 2 :</td><td>0 0 0 0</td></li></ul>	Turn around delay (ms) : Send ACK delay (ms) : Parameter 1 : Parameter 2 :	0 0 0 0
COM : Baud rate : Data bits : Parity : Stop bits :	COM 1 115200 8 Bits Even 1 Bit	~ ~	Turn around delay (ms) : Send ACK delay (ms) : Parameter 1 : Parameter 2 : Parameter 3 :	0 0 0 0 0

Note:

- 1. Between two HMI, Time out has to set to 0.5 sec. and another has to set to 0.8 sec.
- 2. [Data bit] has to be 8 bits.
- 3. The rest of the settings should be identical between two HMI.

Adding two objects on window10, a toggle switch setting is as illustration below:

New Toggle Switch Object
General Security Shape Label
Description :
Read address
PLC name : Memory Map Setting
Address : MB 🗸 0
Invert signal
Write address :
PLC name : Memory Map
Address : MB 0
Write when button is released
Attribute
Switch style : Toggle 🛛 🖌
Macro
Execute macro

A multistate switch object setting is as following:



New Multi-State Sw	ritch Object	×
General Security S	Shape Label	_
Description :		
Mode :	Value Offset : 0	
Read address		h
PLC name :	Memory Map Setting	
Address :	MW V 0 16-bit Unsigned	
– Write address : –––		1
PLC name :	Memory Map Setting	
Address :	MW 0 16-bit Unsigned	
	Write when button is released	
Attribute		5
Switch style :	JOG+ No. of states : 3	
Cyclical :	Enable	
	User-defined mapping	

[Save],[Compile],[Download]

Change parameter in [System Parameter Setting]/[PLC] and download to another HMI.

The HMI display is as following:







Users may try to touch the screen; the other HMI will act the same as current HMI. The communicating way is the same as above-mentioned. The point is to keep the same data in the same register.



# **Chapter 32 ASCII Protocol**

# 32.1 Command List

The following commands are used for communication between the ASCII host and the HMI.

Mnemonic	Command	Description
	Name	
RD	Batch Read	Reads specified data in a continuous block
WD	Batch Write	Writes specified data in a continuous block
RR	Random	Reads data from multiple, non-consecutive
	Read	devices
RW	Random	Writes data to multiple, non-consecutive devices
	Write	
RC	Read Coil	Reads the specified coils in a continuous block
WC	Write Coil	Writes the specified coils in a continuous block



# **32.2 Optional Parameters**

Parameters settings are used as follows:

ASCII Server Set	tings	
COM :	СОМ 1 🛛 👻	Timeout (sec) : 1.0
Baud rate :	9600 💌	Turn around delay (ms) : 0
Data bits :	8 Bits 💌	Protocol
Parity :	Even 🔽	⊙ Robûst ○ Simple
Stop bits :	1 Bit 💌	Response to write commands
	3	⊙ ON <sup>④</sup> OFF
		OK Cancel

#### Protocol

Robust

The protocol uses the non-printable characters STX (02H) and ETX (03H), ACK (06H), and NAK (15H); and includes a 2-byte checksum.

② Simple

Some Host devices (such as Motion Controllers) are not capable of generating the non-printable characters, or calculating the checksum. In this mode, the data packets are formed as defined below, but do not include the STX, ACK, ETX, NAK, or checksum. The 0x0D is at the end of the packet, the packet sent by MT8000 also has a 0x0D at the end.

**Response to write commands:** sets whether or not MT8000 responds to write commands...

- ③ Responses On
- ④ Responses Off

Note: If set to 1, the Turn Around Delay setting (Parameter 2) has no affect.



# 32.3 Network Support

# 32.3.1 Wiring

The MT8000 ASCII protocol shall support network wiring using RS485 2-wire, 4-wire, or RS232.

# 32.3.2 Addressing

The protocol shall support each MT8000 having a unique Station ID. Valid Station ID shall be from 1 to 255.

# 32.3.3 Broadcast Messages

MT8000 doesn't support Broadcast Message.



#### 32.4 Command Usage

# 32.4.1 RD (Batch Read)

#### 32.4.1.1 Request

This command reads up to 99 consecutive 16-bit items from the 'LW' memory area of HMI. The command is always 14 bytes long.

Byte 1	Bytes	Bytes 4,	Bytes 6-9	Bytes 10, 11	Byte 12	Bytes 13, 14
	2,3	5				
1 Byte	2 Bytes	2 Bytes	4 Bytes	2 Bytes	1 Byte	2 Bytes
STX	Station	RD	Addr.	No. of Items	ETX	Checksum

Byte 1: Always STX (0x02)

Bytes 2, 3: The Station Number of the HMI to read (2 Hex digits)

Bytes 4, 5: The command to execute

Bytes 6-9: This is the starting address to read from. Must be 4 bytes long,

Bytes 10, 11: This is the number of addresses to read, up to 99. Must be 2 bytes long. Byte 12: Always ETX (0x03)

Bytes 13, 14: The checksum is the lowest 8 bits of the sum of bytes 2 through 12.

Example: Read 3 words starting from address LW100, from the HMI at station 10 (0AH). This will read addresses LW100 – LW102.

Byte 1	Bytes 2,	Bytes 4,	Bytes 6-9	Bytes 10, 11	Byte 12	Bytes 13, 14
	3	5				
STX	0A	RD	0100	03	ETX	2E
02	30,41	52,44	30,31,30,30	30,33	03	32,45

The checksum (bytes 13 and 14) is calculated as the lowest 8 bits of the sum of the Hex codes for bytes 2 - 12.

30 + 41+ 52 + 44 + 30 + 31 + 30 + 30 + 30 + 33 + 03 = 22E.

The lowest 8 bits of the result returns 2E.

## 32.4.1.2 Reply

The reply length is

#### L = (N \* 4) + 8

Where N = the number of requested devices

If the command is successful, the reply length will be at least 12 bytes, but could be as long as 404 bytes. It consists of the STX, followed by four bytes for each requested device, then the ETX and Checksum.

Byte	Bytes	Bytes	Bytes	Bytes	Bytes	Bytes	Bytes
1	2, 3	4,5	6-9	10-13	14-17	18 - (L-7)	(L-6) -
							(L-3)
STX	Station	CMD	Data 1	Data 2	Data 3	Data 4 –	Data
						Data (N-1)	Ν

Byte	Byte
L-2	L-1, L
ETX	Checksum

The example above returns the following, assuming the HMI contains the following data:

Address	Data
100	75 (4BH)
101	8047 (1F6FH)
102	16,321 (3FC1H)

The following is the packet sent from the HMI

STX	'0'	'A'	'R'	'D'	'0'	'0'	'4'	'B'	'1'	'F'	'6'	'F'	'3'	'F'	'C'	'1'
02H	30H	41H	52H	44H	30H	30H	34H	42H	31H	46H	36H	46H	33H	46H	43H	31H



03H 43H 32H

The values in each requested device are returned in Hex. The checksum is calculated on bytes 2 - (L-2).

In the event of an error, the reply is

Byte 1	Byte 2,3	Byte 4,5	Byte 6
NAK	Station	'R', 'D'	Err Code

# 32.4.2 WD (Batch Write)

## 32.4.2.1 Request

This command writes up to 99 consecutive 16-bit items to the LW memory area of HMI. The length of the command is

#### L = (N \* 4) + 14

Where N = the number of requested devices

The command will be at least 18 bytes long, but can be up to 410 bytes long.

Byte	Bytes	Byte	Byte	Bytes	Bytes	Bytes	Bytes	Bytes	Byt	Byte
1	2, 3	S	S	10,	12-15	16-19	20 - (L-7)	(L-6) -	е	L-1, L
		4, 5	6-9	11				(L-3)	L-2	
STX	Statio	WD	Addr.	No. of	Data	Data	Data 3 –	Data	ETX	Chec
	n			Items	1	2	Data (N-1)	Ν		k-su
										m



Byte 1: Always STX (0x02)
Bytes 2, 3: The Station Number of the HMI to write (2 Hex digits)
Bytes 4, 5: The command to execute
Bytes 6-9: This is the starting address to write to. Must be 4 bytes long,
Bytes 10, 11: This is the number of addresses to write. Must be 2 bytes long.
Bytes 12 – (L-3): The data to write. Up to 99 items, each with four Hex digits.
Byte (L-2): Always ETX (0x03).
Bytes L-1, L: Checksum

Example: Write 3 words starting from address LW201, to the HMI at station 17 (11H). This will write to addresses LW201, LW202, and LW203.

LW201 = 101 (0x65) LW202 = 575 (0x23F) LW203 = 1049 (0x419)

Byte	Bytes	Bytes	Bytes 6-9	Bytes	Bytes	Bytes	Bytes	Byte	Bytes
1	2, 3	4, 5		10,11	12-15	16-19	20-23	24	25,26
STX	11	WD	0201	03	0065	023F	0419	ETX	9A
02	31,31	57,44	30,32,30,31	30,33	30,30,36,35	30,32,33,46	30,34,31,39	03	39,41

The checksum (bytes 25 and 26) is calculated as the lowest 8 bits of the sum of the Hex codes for bytes 2 - 24.

31+ 31 + 57 + 44 + 30 + 32 + 30 + 31 + 30 + 33 + 30 + 30 + 36 + 35 + 30 + 32 + 33 + 46 + 30 + 34+ 31+ 39 + 03 = 49A.

The lowest 8 bits of the result returns 9A.

#### 32.4.2.2 Reply

If the command is successful, the reply is

Byte 1	Byte 2,3	Byte 4,5
ACK	Station	'W', 'D'

In the event of an error, the reply is



Byte 1	Byte 2,3	Byte 4,5	Byte 6
NAK	Station	'W', 'D'	Err Code

# 32.4.3 RR (Random Read)

#### 32.4.3.1 Request

This command reads up to 99 independently-addressed 16-bit items from the LW memory area of HMI. The length of the command is

#### L = (N \* 4) + 8

Where N = the number of requested devices

The command will be at least 12 bytes long, but can be up to 402 bytes long.

Byte	Bytes	Bytes	Bytes	Byte	Bytes	Bytes	Byt	Byte
1	2, 3	4, 5	6-9	S	14 - (L-7)	(L-6) -	е	L-1, L
				10-1		(L-3)	L-2	
				3				
STX	Statio	RR	Addr	Addr	Addr 3 –	Addr N	ETX	Check-s
	n		1	2	Addr (N-1)			um

Byte 1: Always STX (0x02)

Bytes 2, 3: The Station Number of the HMI to read (2 Hex digits)

Bytes 4, 5: The command to execute

Bytes 6-9: This is the first address from which to retrieve data. Must be 4 bytes long, Bytes 10-13: This is the second address from which to retrieve data. Must be 4 bytes long,

Bytes 14 – (L-7): The remaining addresses from which to retrieve data. Each address must be 4 bytes long.



Byte (L-2): Always ETX (0x03).

Bytes L-1, L: Checksum, calculated as the lowest 8 bits of the sum of bytes 2 – (L-2).

32.4.3.2 Reply

If successful, the reply length is

L = (N \* 4) + 8

Where N = the number of requested devices

If successful, the reply length will be at least 12 bytes, but can be up to 406 bytes. It consists of the STX, followed by four bytes for each requested device, then the ETX and Checksum.

Byte	Bytes	Bytes	Bytes	Bytes	Bytes	Bytes	Bytes
1	2,3	4,5	6-9	10-13	14-17	15 - (L-7)	(L-6) -
							(L-3)
STX	Station	Cmd	Data 1	Data 2	Data 3	Data 4 –	Data
						Data (N-1)	Ν

Byte	Byte
L-2	L-1, L
ETX	Checksu
	m

The values in each requested device are returned in Hex. The checksum is calculated as the lowest 8 bits of the sum of bytes 2 - (L-2)...

In the event of an error, the reply is

Byte 1	Byte 2,3	Byte 4,5	Byte 6
NAK	Station	'R', 'R'	Err Code



# 32.4.4 RW (Random Write)

#### 32.4.4.1 Request

This command writes up to 99 independently-addressed 16-bit items to LW memory area of HMI. The length of the command is

#### L = (N \* 8) + 8

Where N = the number of requested devices

The command will be at least 16 bytes long, but can be up to 800 bytes long.

Byte	Bytes	Bytes	Bytes	Bytes	Bytes	Bytes	
1	2,3	4, 5	6-9	10-13	14-17	18-21	
STX	Statio	RW	Addr	Data	Addr	Data	
	n		1	1	2	2	

Bytes	Bytes	Byte	Byte	
(L-10) - (L-7)	(L-6) - (L-3)	L-2	L-1, L	
Addr N Data N		ETX	Check-sum	

Byte 1: Always STX (0x02)

Bytes 2, 3: The Station Number of the HMI to read (2 Hex digits)

Bytes 4, 5: The command to execute

Bytes 6-9: This is the first address to write data to. Must be 4 bytes long,

Bytes 10-13: This is the data to write to the address specified by the previous 4 bytes. Must be 4 bytes long,

Bytes 14 – (L-3): The remaining addresses and data to write to the HMI. Each address and data item must be 4 bytes long.



Byte (L-2): Always ETX (0x03).

Bytes L-1, L: Checksum, calculated as the lowest 8 bits of the sum of bytes 2 – (L-2).

# 32.4.4.2 Reply

#### If the command is successful, the reply is

Byte 1	Byte 2,3	Byte 4,5
ACK	Station	'R', 'W'

In the event of an error, the reply is

Byte 1	Byte 2,3	Byte 4,5	Byte 6	
NAK	Station	'R', 'W'	Err Code	

# 32.4.5 RC (Read Coils)

## 32.4.5.1 Request

This command reads up to 99 consecutive coils from the 'LB' memory area of HMI. The command is always 14 bytes long.

Byte 1	Bytes 2,	Bytes 4,	Bytes 6-9	Bytes 10, 11	Byte 12	Bytes 13, 14
	3	5				
1 Byte	2 Bytes	2 Bytes	4 Bytes	2 Bytes	1 Byte	2 Bytes
STX	Station	RC	Addr.	No. of Items	ETX	Checksum

Byte 1: Always STX (0x02)

Bytes 2, 3: The Station Number of the HMI to read (2 Hex digits)

Bytes 4, 5: The command to execute


Bytes 6-9: This is the starting address to read from. Must be 4 bytes long, Bytes 10, 11: This is the number of coils to read, up to 99. Must be 2 bytes long. Byte 12: Always ETX (0x03)

Bytes 13, 14: The checksum is the lowest 8 bits of the sum of bytes 2 through 12.

Example: Read 12 coils starting from address LB100, from the HMI at Station 7. This will read coils LB100 – LB111.

Byte 1	Bytes	Bytes 4,	Bytes 6-9	Bytes 10, 11	Byte 12	Bytes 13, 14
	2,3	5				
STX	07	RC	0100	02	ETX	22
02	30,37	52,43	30,31,30,30	30,32	03	32,32

The checksum (bytes 13 and 14) is calculated as the lowest 8 bits of the sum of the Hex codes for bytes 2 - 12.

30 + 37 + 52 + 43 + 30 + 31 + 30 + 30 + 30 + 32 + 03 = 222.

The lowest 8 bits of the result returns 22.

#### 32.4.5.2 Reply

The reply length is

#### L = N + 8

Where N = the number of requested devices

If the command is successful, the reply length will be at least 9 bytes, but could be as long as 107 bytes. It consists of the STX, followed by one byte for each requested device, then the ETX and Checksum.

Byte	Bytes	Byte	Byte	Byte	Byte	Bytes
1	2,3	S	2 3		4	5 - (L-4)
		4,5				
STX	Statio	RC	Data	Data	Data	Data 4 –
	n		1	2	3	Data (N-1)



Byte	Byte	Byte
(L-3)	L-2	L-1, L
Data N	ETX	Checksum

If the HMI contains the following data:

100	101	102	103	104	105	106	107	108	109	110	111
0	0	1	0	1	0	1	1	0	0	0	1

The following data is returned

STX	'0'	'7'	'R'	'C'	'0'	'0'	'1'	'0'	'1'	'0'	'1'	'1'	'0'	'0'	'0'
02H	30H	37H	52H	43H	31H	30H	31H	31H	31H	30H	31H	31H	30H	30H	30H

'1'	'1' ETX		'6'	
31H	03H	34H	36H	

The values in each requested device are returned in Hex. The checksum is calculated on bytes 2 - (L-2).

In the event of an error, the reply is

Byte 1	Byte 1 Byte 2,3		Byte 6
NAK	Station	'R', 'C'	Err Code

## 32.4.6 WC (Write Coils)

#### 32.4.6.1 Request

This command writes up to 99 consecutive coils to the 'LB' memory area of HMI. The length of the command is

L = N + 14



#### Where N = the number of requested devices

Byt	Bytes	Bytes	Bytes	Bytes	Byte	Byte	Bytes
е	2,3	4, 5	6-9	10-11	12	13	14 - (L-4)
1							
ST	Station	WC	Addr.	No. of	Data	Data	Data 3 –
Х				Items	1	2	Data (N-1)

The command will be at least 15 bytes long, but can be up to 113 bytes long.

Byte	Byte	Byte
(L-3)	L-2	L-1, L
Data N	ETX	Check-sum

Byte 1: Always STX (0x02)

Bytes 2, 3: The Station Number of the HMI to read (2 Hex digits)

Bytes 4, 5: The command to execute

Bytes 6-9: This is the starting address to write to. Must be 4 bytes long,

Bytes 10, 11: This is the number of addresses to write. Must be 2 bytes long.

Bytes 12 – (L-3): The data to write. Up to 99 items, each with one Hex digit.

Byte (L-2): Always ETX (0x03).

Bytes L-1, L: Checksum

Example: Write 5 bits starting from address LB214 to the HMI at station 12. This will write to addresses LB214 – LB218.

Write the following data:

		0			
214 215		216	217	218	
1	1	0	0	1	

Byt	Bytes	Bytes	Bytes	Bytes	Byte	Byte	Byte	Byte	Byte	Byte	Bytes
e 1	2,3	4, 5	6-9	10,11	12	13	14	15	16	17	18, 19
STX	0C	WC	0214	05	1	1	0	0	1	ETX	2F
02	30,43	57,43	30,32,31,3	30,35	31	31	30	30	31	03	32,46
			4								



The checksum (bytes 18 and 19) is calculated as the lowest 8 bits of the sum of the Hex codes for bytes 2 - 17.

30 + 43 + 57 + 43 + 30 + 32 + 31 + 34 + 30 + 35 + 31 + 31 + 30 + 30 + 31 + 03 = 32F.

The lowest 8 bits of the result returns 2F.

## 32.4.6.2 Reply

#### If the command is successful, the reply is

Byte 1	Byte 2, 3	Byte 4,5		
ACK	Station	'W', 'C'		

In the event of an error, the reply is

Byte 1	Byte 2, 3	Byte 4, 5	Byte 6
NAK	Station	'W', 'C'	Err Code



## 32.4.7 Error Codes

The following table lists the error conditions, and the Error Codes returned for those errors.

Code	Description
06H	Invalid Checksum
10H	Unknown Command
11H	Data Length Error – data overflowed receive
	buffer
12H	Communication Data Error – ETX not found
7AH	Illegal Address
7BH	More than 99 data items were requested



## Chapter 33 EasyDiagnoser

## 33.1 Overview and Configuration

#### Overview

EasyDiagnoser is a tool for detecting the error occurs while HMI is communicating with PLC.

#### Configuration

Step 1.

Open Project Manager and click EasyDiagnoser.

🍑 Project Manager		$\mathbf{X}$		
HMI IP, Password				
Туре: МТ6000/8000	) i Series 🛛 👻			
Settings	Reboot HMI	]		
Connection				
⊙Ethernet ○USE	3 cable (i series only)			
HMI IP :	~			
Data/Event Log	File Information	]		
- Utility				
EasyBuil	lder8000			
EasyConverter	EasyAddressViewer	EasyDiagnoser		
EasyPrinter	EasyDiagnoser -			
Recipe/Extended Memory Editor				
Build Download Dat	a for CF/ USB Disk	1		
Download	Upload			
On-line Simulation	Off-line Simulation	]		
Pass-th	rough			
Help	Exit			

#### Step 2.



Set the IP address of the HMI to communicate with.

Users can input IP address manually or simply click [Search All]. Please input Project Port as well.



It is also available to right click and select "Run EasyDiagnoser" for entering the setting window when executing On-Line Simulation in EB8000.





1

WE!NTEK

After setting completed, click OK, EasyDiagnoser operation window appears as below:

🤯 Weintek MT I	Diagn	ostic 1	fool - Easyl	Diagnoser								
<u>F</u> ile <u>V</u> iew <u>O</u> p	tions	<u>H</u> elp										
Command: Rea	ad + V	Vrite	~	Device:	All			tation:	0		Output	<u>g</u> ×
				_						=		~
Address Type: P	AU .		×	<u>R</u> ange	e: Lu	~ 95	1999		<u>C</u> apture			
No Cm	nd. 🛛	PID	Device		St.	Index	Address / Lengt	n Tir	me Er	ror		
											<	>
											Revices Output	
Polling Packages								Ø×	Logger			<u>@</u> ×
Package ID			Device		Station	Index	Address / Len	gth	[15:15:	22] Loc	oking for the target HMI	
± 4 (1)			Local HMI	I			[LB] 00562/1		[16:16:	22] Cor	nnection established with the target Hi	VII.
8 (0)			Local HMI	l			[LB] 00574/1					
<ul> <li>£ 67 (32)</li> </ul>			SIEMENS	S7/300	1		[M] 00000/1					
£ 68 (14)			SIEMENS	S7/300	1		[DB10] 00000.	14				
Ready											CA	P NUM SCRL



## 33.2 EasyDiagnoser Settings

Item	Description
File	Save As The captured information of Easy Diagnoser can be saved as *.xls which can be read in Excel. Weintek MT Diagnostic Ele View Options Help Save Asd + Write Egit Augress Type.
	Exit Exit current file.
View Device Bax Ctul+Alt+D Sector Package Bax Ctul+Alt+P Logger Bax Ctul+Alt+L Qutput Bax Ctul+Alt+O	Click [Device Bar] to display Device window. Click [Package Bar] to display Package window. Click [Logger Bar] to display Logger window. Click [Output Bar] to display Output window.
Options Options Help <u>Toolbars</u> ✓ Status Bar Update Package List F5 Show Object ID (HMD) Clear Activity List	Toolbars         Display toolbar icons of [Device Bar] [Package Bar] [Logger         Bar] [Output Bar].         Weintek MT Diagnostic Tool - EasyDiagnoser         File       View Options         Help         Image: State Sta
	Show Status Bar         At the bottom of EasyDiagnoser window, display information of CAP, NUM, and SCRL.         Ready       CAP NUM SCRL ,;;
	Update Package List When users change window in HMI, update the Polling Package information of current window with this list. Show Object ID (HMI) Show the ID of objects in HMI as shown below.



#### • Activity area

WE!NTEK

In the activity area, users can observe the communication between HMI and PLC.



Coj	ommand: Read + Write 🔽 Device: All 🔽 Station: 0									
<u>A</u> do	Address Type: All 💽 🗌 Range: 0 ~ 99999						apture			
	No	Cmd.	PID	Device	St.	Index	Address / Length	Time	Error	^
►	139	R	68	SIEMENS \$7/300	1	255	[DB10] 00000 / 14	50	0	
	138	R	4	Local HMI			[LB] 00562/1	20	0	
	137	R	8	Local HMI			[LB] 00574/1	10	0	
	136	R	67	SIEMENS \$7/300	1	255	[M] 00000 / 1	40	0	
	135	R	4	Local HMI			[LB] 00562/1	20	0	
	134	R	8	Local HMI			[LB] 00574/1	20	0	
	133	R	68	SIEMENS \$7/300	1	255	[DB10] 00000 / 14	30	0	
	132	R	4	Local HMI			[LB] 00562/1	20	0	
	131	R	8	Local HMI			[LB] 00574/1	20	0	
	130	R	67	SIEMENS \$7/300	1	255	[M] 00000 / 1	40	0	
	129	R	4	Local HMI			[LB] 00562/1	20	0	~

Item	Description				
Command	a. Read + Write				
	Display Read and Write commands in activity area.				
	b. Read				
	Display only Read commands in activity area.				
	c. Write				
	Display only Write commands in activity area.				
Device	a. All				
	Display information of Local HMI and PLC. It depends on the setting of				
	command as following.				
	<ul> <li>If command is set Read + Write, the Read and Write information of</li> </ul>				
	Local HMI and PLC will be displayed in activity area.				
	• If command is set <b>Read</b> , the Read information of Local HMI and PLC				
	will be displayed in activity area.				
	• If command is set <b>Write</b> , the Write information of Local HMI and PLC				
	will be displayed in activity area.				
	b. Local HMI				
	Display information of Local HMI, it depends on the setting of command				
	as following.				
	<ul> <li>If command is set Read + Write, the Read and Write information of</li> </ul>				
	Local HMI will be displayed in activity area.				
	<ul> <li>If command is set Read, the Read information of Local HMI will be</li> </ul>				



	displayed in activity area.
	<ul> <li>If command is set Write, the Write information of Local HMI will be</li> </ul>
	displayed in activity area.
	Display information of PLC, it depends on the setting of command as
	following
	lonowing.
	• If command is set <b>Read + Write</b> , the Read and Write information of PLC
	will be displayed in activity area.
	<ul> <li>If command is set <b>Read</b>, the Read information of PLC will be displayed</li> </ul>
	in activity area.
	<ul> <li>If command is set Write, the Write information of PLC will be displayed</li> </ul>
	in activity area.
Station	Select specific Station for display on the screen. (This function will be
	disabled when selecting [All] in Device).
Address	Users can select all or a part of address types to be displayed on the
Туре	screen. (This function will be disabled when selecting [All] in Device).
Range	Set the range of address types to be displayed. (This function will be
	disabled when selecting [All] in Address Type).
Capture	Click to start/stop capturing communication message.
Error	Please refer to the section coming later.

## Polling Packages

Polling Packages					ø×
	Package ID	Device	Station	Index	Address / Length
±	4 (1)	Local HMI			[LB] 00562/1
	8 (0)	Local HMI			[LB] 00574/1
Đ	67 (32)	SIEMENS S7/300 Ethernet	1		[M] 00000/1
Đ	68 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000 / 3
Đ	69 (3)	SIEMENS S7/300 Ethernet	1	11	[DB10] 00003/3
Đ	70 (3)	SIEMENS S7/300 Ethernet	1	12	[DB10] 00006/3
Đ	71 (5)	SIEMENS S7/300 Ethernet	1		[DB10] 00009/5

Item	Description
Package ID	Use the information of package ID to check the PID in activity area for
	finding the problem.
Device	Displays HMI and PLC type.
Station	Displays PLC station number.
Index	Display objects-used index register numbers.



Address/Length Displays device type address. Length-how many words of the Package.

Po	lling	Packages				9
	0	oject		Screen	ID	Address
÷	4 (	1)	Local HMI			[LB] 00562/1
	8 (	0)	Local HMI			[LB] 00574/1
Ξ	67	(32)	SIEMENS S7/300 Ethernet	1		[M] 00000/1
	►	Toggle S		10	30	[M] 00000
		Toggle S		10	30	[M] 00000
		Toggle S		10	29	[M] 00000
		Toggle S		10	29	[M] 00000
		Toggle S		10	28	[M] 00000
		Toggle S		10	28	[M] 00000
		Toggle S		10	27	[M] 00000

After opening Package, the information such as Object, Screen, ID, Address inside it will be displayed.

Object Package ID where this object is placed.		
Screen	Window in the project where this object is placed.	
ID	ID of the object.	
Address	Address of the object.	

#### Note:

**a.** Click **[Package ID]**, the device station number will be displayed in 3<sup>rd</sup> column.

iress / Length
00562/1
00574/1
00000/1
10] 00000 / 3

**b.** Double click **[Package ID]** then select **[object]**, the 1<sup>st</sup> column directs the object's position.

For example, select [Numeric Input] and the screen no. displays 10.

This shows that this object is in window no. 10 in the project and will be marked with pink frame in HMI as shown below.



1	Polling	Packages				G	×
	0	bject		Screen	ID	Address	
	± 4	(1)	Local HMI			[LB] 00562/1	
	8	(0)	Local HMI			[LB] 00574/1	
	<b>⊞</b> 67	(32)	SIEMENS S7/300 Ethernet	1		[M] 00000/1	
	<b>a</b> 68	3 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000 / 3	
	•	Numeric I		10	2	[DB10] 00000	
		Numeric I		10	3	[DB10] 00001	
		Numeric I		10	4	[DB10] 00002	

# SIEMENS S7-300 Ethernet



#### • Devices

Devices window displays information of HMI and PLC.

Dev	rices	9	×
Ξ	Local HMI		^
	Index	0	
	Type Name	MT8000 Series HMI	
	Location	Local	
	Block Interval	5 words	
	Max. Read Length	256 words	
	Max. Write Length	256 words	_
	SIEMENS S7/300 Ethernet		=
	Index	1	
	Type Name	SIEMENS S7/300 Ethernet	
	Location	Local	
	PLC I/F	Ethernet (192.168.1.97:1	
	Block Interval	5 words	
	Max. Read Length	20 words	
	Max. Write Length	20 words	_
			~

#### • Output (Macro debug)

With Trace function offered by Macro, the executing status of Macro can be seen. Please refer to EB8000 user's manual *"Chapter 18 MACRO"* for more information.

In illustration below, for [ID 2, Ln 7] and [ID 2, Ln 8]

ID 2 represents Macro name.

Ln 7 and Ln 8 represent that they are in 7<sup>th</sup> and 8<sup>th</sup> lines of Macro.

💖 Weintek MT Diagnostic Tool - EasyDiagnoser	
<u>F</u> ile <u>V</u> iew <u>O</u> ptions <u>H</u> elp	
i 📰 🐗 🚊 🖄	
Output	$\mathcal{O}$ ×
[ID 2, Ln 7] The results are [ID 2, Ln 8] c1 = a, s1 = 32767, f1 = 1.234567 [ID 2, Ln 7] The results are [ID 2, Ln 8] c1 = a, s1 = 32767, f1 = 1.234567 [ID 2, Ln 7] The results are [ID 2, Ln 8] c1 = a, s1 = 32767, f1 = 1.234567 [ID 2, Ln 7] The results are	
[ID 2, Ln 8] c1 = a, s1 = 32767, f1 = 1.234567 [ID 2, Ln 7] The results are [ID 2, Ln 8] c1 = a, s1 = 32767, f1 = 1.234567 [ID 2, Ln 7] The results are	>



#### 33.3 Error Code

In activity area, users can find the reason of error through error codes listed below.

- 0: Normal
- 1: Time out
- 2: Fail Error
- 12: Ignore

When error occurs, error message will be shaded red as shown below.

The error code is 1 since PLC is disconnected with HMI.

The error code is 12 since "PLC No Response" message window is shown.

V	Weintek b	(T Diagn	ostic Tool	l - EasyDiagnoser							
<u> </u>	le <u>V</u> iew	Options	<u>H</u> elp								
	🤞 🗎	Ż									
Co <u>r</u>	Command: Read + Write <ul> <li>Device: SIEMENS S7/300 Ethernet</li> <li>Station:</li> </ul>										
<u>A</u> dd	lress Type	e: All		✓ Range	: 0	~ 99	9999		apture		
	No	Cmd.	PID	Device	St.	Index	Address / Length	Time	Error	^	
	591	R	71	SIEMENS \$7/300 Et	1	255	[DB10] 00009 / 5	310	12	-	
	590	R	67	SIEMENS \$7/300 Et	1	255	[M] 00000 / 1	310	12		
	589	R	68	SIEMENS \$7/300 Et	1	10	[DB10] 00000 / 3	300	12		
	588	R	69	SIEMENS \$7/300 Et	1	11	[DB10] 00003/3	310	12		
	587	R	70	SIEMENS \$7/300 Et	1	12	[DB10] 00006 / 3	310	12		
	586	R	71	SIEMENS \$7/300 Et	1	255	[DB10] 0000975	1210	12		
	585	R	67	SIEMENS \$7/300 Et	1	255	[M] 00000 / 1	1120	12		
	584	R	68	SIEMENS \$7/300 Et	1	10	[DB10] 00000 / 3	1020	1		
	583	R	69	SIEMENS S7/300 Et	1	11	[DB10] 00003/3	40	0		
	582	R	70	SIEMENS S7/300 Et	1	12	[DB10] 00006 / 3	30	0		
	581	R	71	SIEMENS S7/300 Et	1	255	[DB10] 00009/5	40	0	~	



## 33.4 Save As

The captured information of Easy Diagnoser can be saved as \*.xls which can be read in Excel.

1	Weintek b	(T Diagno	ostic Tool	- EasyDiag	noser						
<u>[ F</u> i	le <u>V</u> iew	Options	<u>H</u> elp								
	Save <u>A</u> s										
<u> </u>	Exit	d + W	/rite	~	Device:	SIEMENS	87/300 E	thernet 🔽 [			
<u>A</u> dd	ress Type	e: All		~	<u>R</u> ange	: 0	~ 99	999			
	No	Cmd.	PID	Device		St.	Index	Address / Length			
	176	R	68	SIEMENS 8	37/300 Et	1	10	[DB10] 00000/3			
	175	R	69	SIEMENS 8	37/300 Et	1	11	[DB10] 00003/3			
	174	R	70	SIEMENS 8	37/300 Et	1	12	[DB10] 00006/3			
	173	R	71	SIEMENS 8	37/300 Et	1	255	[DB10] 00009/5			



### 33.5 Window Adjustment

Users can drag or use smart docking icons in editing window to place the windows to the desired position.



Note:

EasyDiagnoser doesn't support Siemens S7/1200 (Ethernet) and Allen-Bradley Ethernet/IP (CompactLogix/ControlLogix) – Free Tag Names since both of the PLC use tag.



# Chapter 34 AB EtherNet/IP Free Tag Names

When using the driver of Allen-Bradley EtherNet/IP-Tag (CompactLogix/ ControlLogix) in EB8000, users can import User-Defined Tag from CSV file of RSLogix5000. However, data type of User-Defined, Predefined and Module-Defined Structure won't be imported.

	A	В	С	D	Е	F	
7	TYPE	SCOPE	NAME	DESCRIPT	DATATYPE	SPECIFIER	ATTRIBUTES
8	TAG		Local:1:C		AB:Embedded_IQ16F:C:0		
9	TAG		Local:1:I		AB:Embedded_IQ16F:I:0		
10	TAG		Local:2:C		AB:Embedded_OB16:C:0		
11	TAG		Local:2:I		AB:Embedded_OB16:I:0		
12	TAG		Local:2:0		AB:Embedded_OB16:O:0		
13	TAG		Array2D		DINT[25,5]		(RADIX := Decimal, Cons
14	TAG		ArrayBool		BOOL[256]		(RADIX := Decimal, Cons
15	TAG		Array DIN 1	•	DINT[130]		(RADIX := Decimal, Cons
16	TAG		ArrayReal		REAL[125]		(RADIX := Float, Constant
17	TAG		B001		INT[15]		(RADIX := Decimal, PLC)
18	TAG		b003		INT[255]		(RADIX := Decimal, PLC)
10	TAG		h1		POOT		(PADIX - Desimal Cons

Therefore, AB Data Type Editor in EB8000 is for users to import and edit User-Defined, Predefined and Module-Defined Structure.



## 34.1 Import User-Defined AB Tag to EB8000

**Step 1.** Create Tags from RSLogix5000.

RSLogix 5000 - AB [1769-L23E-QB1 18.11]* - [Controller Tags - AB(controller)]									
🎽 File Edit View Search Logic Communications	🖻 File Edit View Search Logic Communications Tools Window Help								
🖺 🚅 🚽 🎒 🖏 🕫 🕫 🕫 🕫 🔍									
Rem Run  Rem Run Mode No Forces Rem Run Mode	Path: AB_ETH-1\192.168.1.13	0\Backplane\0*	▼ 品						
	Favorites Add-On A S	-( )(U)(L)- Safety 🔏 Alarms 🔏 Bit	Timer/Ci						
Controller Organizer - 4 ×	Scope: 🛐 AB 🛛 👻 Sh	ow: All Tags		*					
Controller Tags	Name == A	Value 🗲 For	rce Mask 🔹 Style	Data Type					
Controller Fault Handler	+ ABC	56	Decimal	DINT					
🗀 Power-Up Handler	🕂 🔄 🛨 Array2D	{}	{} Decimal	DINT[25,5]					
😑 📛 Tasks	±-ArrayBool	{}	{} Decimal	BOOL[256]					
🖃 😋 Main Task		{}	{} Decimal	DINT[130]					
Hunscheduled Programs / Phases		{}	{} Float	REAL[125]					
Motion Groups	ь1	0	Decimal	BOOL					
Ungrouped Axes		{}	{} Decimal	INT[360]					
- 🗀 Add-On Instructions	+-Local:1:C	{}	{}	AB:Embedded_IQ					
📄 📇 Data Types	+-Local:1:I	{}	{}	AB:Embedded_IQ					
	+-Local:2:C	{}	{}	AB:Embedded_0					
H umgs Add-On-Defined	+-Local:2:1	{}	{}	AB:Embedded_0					
The on Pointer	+-Local:2:0	{}	{}	AB:Embedded 0					
🗄 🙀 Module-Defined	VarBool	0	Decimal	BOOL					
🗀 Trends	+ VarDint	21862	Decimal	DINT					
🖻 🔄 I/O Configuration	T + Varint	0	Decimal	INT					
CompactLogix5323E-QB1 System	VarBeal	0.0	Float	BEAL					
□ 1769-L23E-QBLAB	+ VarSint	-128	Decimal	SINT					

Step 2. Export Tags data to CSV file.

🔏 RSLogix 5000 - AB [1769-L23E-QB1 18.11	]*-	[Co	ontroller Tags - AB(contro	o]]]	er)]
🃝 File Edit View Search Logic Communicatio	ms	Tool	s Window Help		
	-		Options		<b>.</b> 🖪 🕞 🛛 🖉 🔍 Q Q 🗌
	्रम्ब		Security	۲.	
Kem Hun (), Run Mode	<b>A</b> - (	9	Documentation Languages		8.1.130\Backplane\0* 🛛 👻
No Forces			<u>T</u> ranslate PLC5/SLC		+/+ -( )(U)(L)-
	L		Import	F	🔏 Safety 🔏 Alarms 🔏 Bit 🥻 Tim
Controller Organizer 🚽 📮	×		Export	Ы	Tags and <u>L</u> ogic Comments
Controller AB	^		<u>M</u> otion		Component
Controller AB	^		<u>M</u> otion Monitor Equipment Phases	Þ	Component
Controller AB Controller Tags Controller Fault Handler Power-Up Handler	^		<u>M</u> otion Monitor Equipment <u>P</u> hases	•	Component 35 56 {}
Controller AB			Motion Monitor Equipment Phases Custom Tools	•	Component 35 56 () ()
Controller AB     Controller Tags     Controller Fault Handler     Power-Up Handler     Tasks     Ga Main Task     Ga Main Program		07.2	Motion Monitor Equipment Phases Custom Tools	•	Component 35 56 () ()
Controller AB Controller Tags Controller Fault Handler Controller Faul		<b>E</b>	Motion Monitor Equipment Phases Custom Tools ControlFLASH	•	Component 35 56 () () ()

**Step 3.** In EB8000, create Allen-Bradley EtherNet/IP-Tag (CompactLogix/ControlLogix) driver.

Input PLC IP address. In System Parameter Settings dialog click [Import Tag...] button.



\$	ystem Param	eter Settings				X						
						_	Open	J				? 🔀
Ι,	Font		Extended	Memory	Printer/Backup Server			Look jn:	C EB8000	<u> </u>	🦻 📂 🛄	•
	Device	Model		General Sys	stem Setting Security				driver			
	Device list :						Mj	Recent	firmware font			
	No.	Name	Location	Device type	Interface				Dibrary			
	Local HMI	Local HMI	Local	MT6070iH/MT8070	. Disable		C	esktop	project			
	Local PLC 4	Allen-Bradley E	Local	Allen-Bradley E	Ethemet(IP=192.168.1.130, Po	rt=4			ausb1			
							My D	Ocuments	🚞 usbdriver 🚞 virtual_com			
							1		🗐 AB-Tags			
							My	Computer				
									File name:	AB-Tags	~	<u>O</u> pen
							Му	Network	Files of type:	RSLogix 5000 Import/Export File (*.CSV)	~	Cancel
							Eas	s <b>y</b> Build	er8000	$\mathbf{X}$		
									<b>.</b>			
	<					>	4	<u>.</u>	import tag	information successfully.		
	New	. Dele	te	Settings	Import Tag				1	定		
	Project descrij	ption :										

Step 4. In object dialog, select PLC, click Tag and select a controller tag.

eneral	Security	Shape Label		
	Description	:		
Read	address —			
	PLC name :	Allen-Bradley EtherNet,	/IP-Tag (CompactLog	gix/ 🔽 🚺 Setting
	Tag :	0		· ?
		Name	Data Type	Description
		😑 Controller Tags		
		ArrayBool     h1	BOOL[256] BOOL	
		VarBool	BOOL	
Dist				
ышк	urg			
	Mode			
	ITTUILIE.	•C		



## 34.2 Adding New Data Type

**Step 1.** Right click on the assigned data type (usually labeled as [User-Defined]), then click [New Data Type] to start editing.

🛔 AB Data Type Editor	Σ
<ul> <li>Data Types</li> <li>User-Drefered</li> <li>Strings New Data Type</li> <li>Predefined</li> <li>Module-Defined</li> </ul>	Name Description
	Name Data Type Descriptione
	Add Paste Edit Delete Omember OK
	Reload Save Exit

**Step 2.** Input the [Name] of the data type. [Description] can be skipped. For adding data member, click [Add].



😑 Data Types		
■ User-Defined	Name TestStruct	
	Description	~
■ Module-Defined		
		~
	Name Data Type Description	e
		>
	Add Paste Edit Delete Omember	ж

Step 3. Input in [Name] and [Data Type] then click [OK] to leave.

) WEINTEK

Add data men	ber	X
Name 🤇	Data1	-
Description		×
Data Type		Binary Access
	<ul> <li>User-Defined</li> <li>Strings</li> <li>Predefined</li> <li>Module-Defined</li> </ul>	
Array Dir	nensions	
	Dim <u>2</u> Dim <u>1</u>	Dim <u>0</u>
0	÷ 0 ÷ 0	•
🗹 Show Dat	a Types by Groups OK	Cancel



**Step 4.** After adding all data members, click [OK]. The built data type will be listed on the left side.

🛔 AB Data Type Editor		×
<ul> <li>Data Types</li> <li>User-Defined</li> <li>TestTypeA</li> <li>TestTypeB</li> <li>TestTypeC</li> <li>TestTypeE</li> <li>TestStruct</li> <li>Strings</li> <li>Predefined</li> <li>Module-Defined</li> </ul>	Name TestStruct   Description     Name   Data Type   Descriptione   Data1   INT     Add   Paste   Edit   Delete   1	
	Reload Save Exit	

**Note:** After changing [Name] or [Description] of a data type, [OK] must be clicked to activate revision.



#### 34.3 Paste

**Step 1.** When adding new data members, this function allows users to add multiple data at one time. The way is to click [Paste] in the [AB Data Type Editor] window.



**Step 2.** The way to edit is to input data name in each line first, then use space or tab key to leave a space in each line. And then input data type or click [Sample] to see some reference. It is recommended to directly copy and paste from RSLogix5000 to avoid errors.

Name: T	estTypeA			
)escription:				
Members:	Data Type	ata Type Size: 60 byte	(s)	External Access
VarBool	BOOL	Decimal		Read/Write
De alé man				
BOOKArray	BUUL[32]	Decimal		Read/Write
VarReal	REAL	Float		Read/Write Read/Write
VarReal RealArray	REAL REAL[5]	Float Float		Read/Write Read/Write Read/Write
VarReal RealArray VarInt	REAL REAL[5]	Float Float Float Decimal		Read/Write Read/Write Read/Write Read/Write
VarReal RealArray VarInt IntArray	REAL REAL[5] INT INT[3]	Float Float Decimal Decimal		Read/Write Read/Write Read/Write Read/Write Read/Write
VarReal RealArray VarInt IntArray VarDint	BODE[32]           REAL           REAL[5]           INT           INT[3]           DINT	Float Float Decimal Decimal Decimal		Read/Write Read/Write Read/Write Read/Write Read/Write Read/Write
VarReal RealArray VarInt IntArray VarDint DintArray	BODE[32]           REAL           REAL[5]           INT           INT[3]           DINT           DINT[3]	Float Float Decimal Decimal Decimal Decimal		Read/Write Read/Write Read/Write Read/Write Read/Write Read/Write Read/Write



**Step 3.** The table above shows the defined data types in RSLogix. Select [Name] and [Data Type] with mouse. This can be done by pressing and holding on the first option, then slide down to the bottom until the scroll rolls to the end then stop holding. All the items will then be selected. Press ctrl+v to copy then paste to the editing window.

Paste		×
VarBool BOOL BoolArrayBOOL[32] VarReal REAL RealArray REAL[5] VarInt INT IntArray INT[3] VarDint DINT DintArray DINT[3] 		
10 lines	Sample OK Cancel	)

**Step 4.** At this moment press [OK] to finish operating then return to the main window to view the successfully added multiple data.





#### 34.4 Miscellaneous

• Revising member data:

Directly double click on the data member to be revised in the main window, or click on the data member then press [Edit].

• Deleting data member:

Select the data to be deleted then click [Delete]. For deleting all data members, press and hold [Delete] button on the keyboard then click the [Delete] button in the main editing window.

• Adjusting the order of data members:

After selecting a single data member, use the move up and move down buttons in main window to adjust the order. This makes selecting items in EB8000 easier.

• Deleting data type:

In the list on the left side of the main window, select the data type to be deleted then press [Delete] on the keyboard. A confirming window pops up; click [Yes] to start deleting.

• Saving the result of revision:

After revising, [Save] button in main window must be clicked. Restart EB8000, the result of revision can be viewed.

• To Re-edit:

For giving up all revision done and to re-edit, click [Reload] button in main window.



## 34.5 Module-Defined

Here is an example showing how to define a default structure for a module.

In I/O Configuration of RSLogix contains setting of I/O module.



The Tags of these modules won't list the structure when exported to CSV file. Therefore, users should build it first.

	A	В	С	D	Е	F	G	Н
-7	TYPE	SCOPE	NAME	DESCRIPT	DATATYPE	SPECIFIER	ATTRIBUT	FES
8	TAG		Local:1:I		AB:1769_DI16:I:0			
9	TAG		Local:2:I		AB:1769_DI32:I:0			
10	TAG		Local:3:C		AB:1769_D016:C:0			
11	TAG		Local:3:I		AB:1769_D016:I:0			
12	TAG		Local:3:0		AB:1769_D016:0:0			
13	TAG		Local:4:C		AB:1769_D032:C:0			
14	TAG		Local:4:I		AB:1769_D032:I:0			
15	TAG		Local:4:0		AB:1769_D032:0:0			
16	TAG		Local:5:C		AB:1769_D016:C:0			
17	TAG		Local:5:I		AB:1769_D016:I:0			
18	TAG		Local:5:0		AB:1769_D016:0:0			
19	TAG		Local:6:C		AB:1769_IF16:C:0			
20	TAG		Local:6:I		AB:1769_IF16:I:0			
21	TAG		Local:6:0		AB:1769_IF16:0:0			
22								

# (1)

In [Controller Organizer/Data Types/Module-Defined] of RSLogix5000, double click Data Type of the module. Data members of that type of the module will be listed in a window pops up. Copy the [Name] and [Data Type] of the Members.





(2)

WE!NTEK

In [AB Data Type Editor.exe] in EB8000, right click on [Module-Defined], and then click [New Data Type...].

AB Data Type Editor	X
<ul> <li>Data Types</li> <li>User-Defined</li> <li>Strings</li> <li>Predefined</li> <li>Module-Defined</li> <li>Module-Defined</li> <li>Mew Data Type</li> <li>AB:Embedded_IQ16F:I:0</li> <li>AB:Embedded_OB16:C:0</li> </ul>	Name Description
AB:Embedded_OB16:I:0 AB:Embedded_OB16:O:0	Name Data Type Descriptione
	Reload Save Exit

In [Name] of [New Data Type], input Module-Defined Name.

3

Click [Paste], in dialogue box press Ctrl+V to paste Name and Data Type.



(4)



Select data then click [Edit], since the data of the modules can be operated by bit, here [Binary Access] should be selected, then click [OK] to return to [Data Type Editor].

Edit data men	nber	
Name	Data	-
Description		4
Data Type	INT	Binary Access
- Array Di	AB:1769_DI16:I:0 AB:Embedded_IQ16F:C:0 AB:Embedded_IQ16F:I:0 AB:Embedded_OB16:C:0 AB:Embedded_OB16:I:0 AB:Embedded_OB16:0:0 ALARM ALARM_ANALOG ALARM_DIGITAL AUX_VALVE_CONTROL	
	Dim <u>2</u> Dim <u>1</u> D	Dim <u>O</u>
0	0	*
Show Da	ta Types by Groups OK	Cancel

Click [OK] to finish setting.



# **Chapter 35 FTP Server Application**

In addition to backup history data from HMI to USB memory stick or EasyPrinter, FTP Server can also be applied to do this. After downloading project to HMI, FTP Server can be used to backup history data and recipe data, and also to update recipe data. The files in FTP Server can't be deleted.

### 35.1 Login FTP Server

**Step 1.** Before login FTP Server, please check the OS Image version: MT6000/8000 i Series: OS Image 20100818 or later MT8000 X Series: OS Image 20100906 or later

**Step 2.** Enter HMI IP: <u>ftp://192.168.1.103</u>, login user name: uploadhis, and the HMI history upload password (if not changed, the default is 111111). Or, to directly enter <u>ftp://uploadhis:11111@192.168.1.103/</u>



**Step 3.** After entering IP, <u>ftp://192.168.1.103/</u> is shown, and the "datalog", "eventlog", and "recipe" folders can be seen.



### 35.2 Backup History Data and Update Recipe Data

Step 1. To backup "Data Sampling" records.



Select "datalog" folder, the file names set by EB8000 can be seen, click them to check the "datalog" files.



Users can save "Data Sampling" records on PC by copy & paste.

Step 2. To backup "Event (Alarm) Log" records.

Select "eventlog" folder to check the files. Save them on PC by copy & paste.



Step 3. "Event (Alarm) Log" records.

Select "recipe" folder to check the file records of events. Save them on PC by copy & paste.

To update the recipe data in HMI, overwrite "recipe.rcp" with new data and restart HMI in one minute.



**Note:** After updating "recipe.rcp", HMI must be restarted in one minute or use [LB-9048] and [LB-9047] to restart HMI. Set [LB-9048] to ON and then set [LB-9047] to ON to successfully restart HMI.

[LB-9047] reboot HMI (set ON when LB9048 is ON)

[LB-9048] reboot-HMI protection