Chapter 1 EasyBuilder Pro Installation and Startup Guide	9
1.1 EasyBuilder Pro Installation	9
1.2 Steps to Install EasyBuilder Pro	10
Chapter 2 Project Manager Operations	16
2.1 HMI IP, Password	17
2.2 Utility	18
2.2.1 Build Download Data for Saving in CF/SD Card or USB Disk	18
2.2.2 Steps to Download Project to HMI via USB or CF/SD Card	18
2.3 Transfer	19
2.3.1 Download	19
2.3.2 Upload	20
2.4 Simulation	21
2.4.1 Off-line Simulation/On-line Simulation	21
2.5 Pass-Throug81Bh	23
Chapter 3 Create an EasyBuilder Pro Project	24
3.1 Create a New Project	24
3.2 Save and Compile the Project	26
3.3 Off-line and On-line Simulation	27
3.4 Download the Project to HMI	28
Chapter 4 Hardware Settings	33
4.1 I/O Ports of HMI	33
4.2 HMI System Settings	34
4.2.1 System Reset	34
4.2.2 System Toolbar	35
4.2.3 System Information	36
4.2.4 System Setting	36
Chapter 5 System Parameter Settings	40
5.1 Device	41
5.1.1 How to Control a Local PLC	42
5.1.2 How to Control a Remote PLC	47
5.1.3 How to Control a Remote HMI	49
5.2 Model	51
5.3 General	55
5.4 System Setting	58
5.5 Security	61
5.6 Font	64
5.7 Extended Memory	66
5.8 Printer/Backup Server	68

Chapter 6 Window Operations	70
6.1 Window Types	70
6.1.1 Base Window	70
6.1.2 Fast Selection Window	71
6.1.3 Common Window	72
6.1.4 System Message Window	73
6.2 Create, Set, and Delete a Window	75
6.2.1 Creating and Setting a Window	75
6.2.2 Open, Close and Delete a Window	78
Chapter 7 Event Log	79
7.1 Event Log Management	79
7.1.1 Excel Editing	80
7.2 Create a New Event Log	81
7.2.1 Alarm (Event) Log General Settings	81
7.2.2 Alarm (Event) Log Message Settings	83
7.3 Event Log Relevant Registers	85
Chapter 8 Data Sampling	
8.1 Data Sampling Management	86
8.2 Create a New Data Sampling	87
8.3 System Registers Relevant to Data Sampling	91
Chapter 9 Object General Properties	92
9.1 Selecting PLC	92
9.1.1 Setting the Reading and Writing Address	92
9.2 Using Shape Library and Picture Library	95
9.2.1 Settings of Shape Library	96
9.2.2 Settings of Picture Library	99
9.3 Setting Text Content	101
9.4 Adjusting Profile Size	106
9.5 Variables of Station Number	107
9.6 Broadcast Station Number	109
Chapter 10 Security	110
10.1 User Password and Operable Object Classes	110
10.2 Object Security Settings	111
10.3 Setting Example	112
Chapter 11 Index Register	115
11.1 Introduction	115
11.2 Examples of Index Register	116
Chapter 12 Keyboard Design and Usage	119

12.1 Steps to Design a Pop-up Keyboard	
12.2 Steps to Design a Keyboard with Direct Window	
12.3 Steps to Design a Fixed Keyboard on Screen	
12.4 Steps to Design a UNICODE Keyboard	124
Chapter 13 Objects	125
13.1 Bit Lamp	
13.2 Word Lamp	128
13.3 Set Bit	
13.4 Set Word	137
13.5 Function Key	145
13.6 Toggle Switch	152
13.7 Multi-State Switch	
13.8 Slider	
13.9 Numeric Input and Numeric Display	
13.10 ASCII Input and ASCII Display	175
13.11 Indirect Window	
13.12 Direct Window	
13.13 Moving Shape	
13.14 Animation	
13.15 Bar Graph	200
13.16 Meter Display	208
13.17 Trend Display	216
13.18 History Data Display	227
13.19 Data Block Display	235
13.20 XY Plot	248
13.21 Alarm Bar and Alarm Display	
13.22 Event Display	
13.23 Data Transfer (Trigger-based)	275
13.24 Backup	278
13.25 Media Player	
13.26 Data Transfer (Time-based)	297
13.27 PLC Control	301
13.28 Schedule	
13.29 Option List	329
13.30 Timer	
13.31 Video In	341
13.32 System Message	345
Chapter 14 Shape Library and Picture Library	

14.1 Creating Shape Library	347
14.2 Creating Picture Library	354
Chapter 15 Label Library and Multi-Language Usage	
15.1 Introduction	
15.2 Building Label Library	
15.3 Setting Label Font	
15.4 Using Label Library	364
15.5 Settings of Multi-Language (System Register LW-9134)	
Chapter 16 Address Tag Library	
16.1 Creating Address Tag Library	367
16.2 Using Address Tag Library	369
Chapter 17 Transferring Recipe Data	
17.1 Updating Recipe Data with Ethernet or USB cable	371
17.2 Updating Recipe Data with CF/SD Card or USB Disk	372
17.3 Transferring Recipe Data	373
17.4 Saving Recipe Data Automatically	373
Chapter 18 Macro Reference	
18.1 Instructions to the Macro Editor	374
18.2 Macro Construction	
18.3 Syntax	
18.3.1 Constants and Variables	
18.3.2 Operators	
18.4 Statement	
18.4.1 Definition Statement	
18.4.2 Assignment Statement	
18.4.3 Logical Statements	
18.4.4 Selective Statements	
18.4.5 Reiterative Statements	
18.5 Function Blocks	
18.6 Build-In Function Block	402
18.6.1 Mathematical Functions	402
18.6.2 Data Transformation	408
18.6.3 Data Manipulation	413
18.6.4 Bit Transformation	416
18.6.5 Communication	418
18.6.6 String Operation Functions	434
18.6.7 Miscellaneous	460
18.7 How to Create and Execute a Macro	467

18.7.1 How to Create a Macro	467
18.7.2 Execute a Macro	472
18.8 User Defined Macro Function	473
18.8.1 Import Function Library File	474
18.8.2 How to Use Macro Function Library	476
18.8.3 Function Library Management Interface	478
18.9 Some Notes about Using the Macro	485
18.10 Use the Free Protocol to Control a Device	486
18.11 Compiler Error Message	492
18.12 Sample Macro Code	499
18.13 Macro TRACE Function	504
18.14 The Usage of String Operation Functions	513
18.15 Macro Password Protection	525
Chapter 19 Set HMI as a MODBUS Server	527
19.1 Setting HMI as MODBUS Device	527
19.1.1 Creating a MODBUS Server	527
19.1.2 Read from / Write to MODBUS Server	530
19.2 Changing the Station Number of a MODBUS Server in Runtime	533
19.3 About MODBUS Address Type	534
Chapter 20 How to Connect a Barcode Device	535
20.1 How to Connect a Barcode Device	535
Chapter 21 Ethernet Communication and Multi-HMI Connection	539
21.1 HMI to HMI Communication	540
21.2 PC to HMI Communication	541
21.3 Operate the PLC Connected with Other HMI	542
Chapter 22 System Reserved Words / Bits	543
22.1 The Address Ranges of Local HMI Memory	544
22.1.1 Bits	544
22.1.2 Words	545
22.2 HMI Time	546
22.3 User Name and Password	547
22.4 Data Sampling	548
22.5 Event Log	550
22.6 HMI Hardware Operation	552
22.7 Local HMI Network Information	553
22.8 Recipe and Extended Memory	554
22.9 Storage Space Management	556
22.10 Touch Position	557

I

22.11 Station Number Variables	558
22.12 Index Register	559
22.13 MTP File Information	560
22.14 MODBUS Server Communication	561
22.15 Communication Parameters Settings	
22.16 Communication Status with PLC (COM)	
22.17 Communication Status with PLC (Ethernet)	568
22.18 Communication Status with PLC (USB)	572
22.19 Communication Status with Remote HMI	573
22.20 Communication Status with Remote PLC	579
22.21 Communication Error Messages & No. of Pending Cmd	
22.22 Miscellaneous Functions	583
22.23 Remote Print/Backup Server	585
22.24 EasyAccess	
22.25 Pass-Through Settings	587
22.26 Disable PLC No Response Dialog Box	588
22.27 HMI and Project Key	589
22.28 Fast Selection Window Control	590
22.29 Input Object Function	591
22.30 Local/Remote Operation Restrictions	592
Chapter 23 HMI Supported Printers	
23.1 The Supported Printer Types	593
23.2 How to Add a New Printer and Start Printing	596
23.2.1 Add Printer Type	596
23.2.2 Start Printing	597
Chapter 24 Recipe Editor	
24.1 Introduction	598
24.2 Recipe Editor Setting	598
Chapter 25 EasyConverter	601
25.1 How to Export to Excel	601
25.2 How to Use Scaling Function	603
25.3 How to Use Multi-File Conversion	605
Chapter 26 EasyPrinter	606
26.1 Using EasyPrinter as a Printer Server	607
26.1.1 Setup Procedure in EasyPrinter	607
26.1.2 Setup Procedure in EasyBuilder8000	608
26.2 Using EasyPrinter as a Backup Server	611
26.2.1 Setup Procedure in EasyPrinter	611

26.2.2 Setup Procedure in EasyBuilder8000	612
26.3 EasyPrinter Operation Guide	615
26.3.1 Appearance	615
26.3.2 Operation Guide	616
26.4 Convert Batch File	622
26.4.1 The Default Convert Batch File	622
26.4.2 Specialized Criteria	623
26.4.3 The Format of a Convert Batch File	624
26.4.4 The Order of Examining Criteria	625
Chapter 27 EasySimulator	626
27.1 Prepare Needed Files	626
27.2 Modify the Content of "xob_pos.def"	628
Chapter 28 Multi-HMI Intercommunication (Master-Slave Mode)	629
28.1 How to Create a Project of Master HMI	630
28.2 How to Create a Project of Slave HMI	631
28.3 How to Connect with MT500 Project of Slave HMI	634
Chapter 29 Pass-Through Function	637
29.1 Ethernet Mode	638
29.1.1 How to Change the Virtual Serial Port	639
29.1.2 How to Use Ethernet Mode	642
29.2 COM Port Mode	644
29.2.1 Settings of COM Port Mode	644
29.2.2 HMI Work Mode	646
29.3 Using System Reserved Addresses to Enable Pass-Through Function	650
Chapter 30 Project Protection	651
30.1 XOB Password	652
30.2 Decompilation is Prohibited	653
30.3 Disable HMI Upload Function [LB-9033]	654
30.4 Project Key	655
30.5 Project Password (MTP file)	656
Chapter 31 Memory Map Communication	657
Chapter 32 FTP Server Application	667
32.1 Login FTP Server	667
32.2 Backup History Data and Update Recipe Data	669
Chapter 33 EasyDiagnoser	671
33.1 Overview and Configuration	671
33.2 EasyDiagnoser Settings	674
33.3 Error Code	681

33.4 Save As	
33.5 Window Adjustment	
Chapter 34 AB EtherNet/IP Free Tag Names	684
34.1 Import User-Defined AB Tag to EB8000	685
34.2 Adding New Data Type	687
34.3 Paste	690
34.4 Miscellaneous	693
34.5 Module-Defined	694



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, Russian, and French version) and latest upgraded files.

Hardware Requirements (Recommended):

CPU: INTEL Pentium II or higher Memory: 256MB or higher Hard Disk: 2.5GB or higher (Disc space available at least 500MB) CD-ROM: 4X or higher Display: 256 color SVGA with 1024 x 768 resolution or greater Keyboard and Mouse Ethernet: for project downloading/uploading USB Port 2.0: for project downloading/uploading RS-232 COM: At least one available RS-232 serial port required for on-line simulation Printer

Operating System:

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 🛛 🛛 🔀
P	Select the language to use during the installation:
	English
	OK Kel
E	nglish, Spanish, French, Italian,
S	implified Chinese, Traditional







3. Users will be asked if they would like to remove the old versions of EasyBuilder 8000. Please tick those should be removed and click **[Next]** to continue.

B000 Remove Information Find other version of EB8000 that has been installed.	
Would you want to remove EB8000? Select the version that you w	ant to remove.
EasyBuilder 8000 V3.4.5 Chinese (Traditional)	
EB8000V410 EB8000V420	
EB8000V421	
EB8000V430	
EB8000V433	
V	
< Back Ne	xt > Cano



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

Setup - EasyBuilder Pro ¥5.00
Select Destination Location Where should EasyBuilder Pro V5.00 be installed?
Setup will install EasyBuilder Pro V5.00 into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse. E:\EBpro Browse
At least 449.6 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 - EasyBuilder Pro 🛛 5.00	
Select Start Menu Folder Where should Setup place the program's shortcuts?	
Setup will create the program's shortcuts in the following Start	
To continue, click Next. If you would like to select a different folder, click	Browse
< Back Next >	Cancel



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.

🔂 Setup - EasyBuilder Pro ¥5.00	
Select Additional Tasks Which additional tasks should be performed?	
Select the additional tasks you would like Setup to perform while insta Pro V5.00, then click Next. Additional icons:	lling EasyBuilder
Create a desktop icon	
< <u>B</u> ack Next	> 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 - EasyBuilder Pro ¥5.00		
Ready to Install Setup is now ready to begin installing Eas	syBuilder Pro V5.00 on your computer.	
Click Install to continue with the installatic change any settings.	on, or click Back if you want to review or	
Destination location: C:\EBpro Start Menu folder: EBpro Additional tasks: Additional icons:		
Create a desktop icon	2	2
	< Back Install	Cancel



8. Installation processing.

🕼 Setup - EasyBuilder Pro ¥5.00	
Installing Please wait while Setup installs EasyBuilder Pro V5.00 on your computer.	
Extracting files C:\EBpro\library\button1.flb	
	Cancel
	Cancer

9. Click [Finish] to complete the installation.





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



The description of each item in EasyBuilder Pro menu:

Installed file	Description
👔 AB Data Type Editor	Support AB TAG mechanism and improve the flexibility of an object
E NB Dala Type Eurior	in read/write.
😼 EasyBuilder8000	EasyBuilder Pro editing software.
🕎 EasyConverter	Conversion tool for Data Sampling and Event Log.
🤯 EasyDiagnoser	Tool for analyzing and detecting connection between HMI and PLC.
🕒 Kasa Printar	Tool for saving hardcopy or backup data is individually
🛬 EasyPrinter	downloadable even without full application.
	Upon completion of project programming, you can execute Online
😼 EasySimulator	Simulation on PC by directly connect with PLC or Offline Simulation
	on PC without connecting PLC.
🦪 Project Manager	EasyBuilder Pro project management.
Decise Editors	Tool for setting format of Recipe data. Users can open Recipe data
🖄 RecipeEditor	or data in External Memory here.
🔁 ReleaseNote	Notes for EasyBuilder Pro version and latest information.
🚔 EasyAddressViewer	Review the register range of device types for each PLC supported.

Refer to 🥏

■ 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 EasyBuilder Pro software, double click on **[Project Manager]** shortcut. The Project Manager is a software shell for launching several utilities. Some functions are duplicated in the EasyBuilder Pro project editing program. Project Manager can operate as a stand-alone program.

🏈 Project Manager When operating HMI, After rebooting, designate Password HMI IP, Password everything returns to first. Type : MT6000/8000 i Series the startup condition. × Settings... Reboot HMI Connection Connect via USB ○ Ethernet ⊙ USB cable (i series only) cable or Ethernet to Launch project check the HMI history editor. files information. Data/Event Log File Information Review the register Utility Conversion tool for range of device types EasyBuilder8000 Data Sampling/ of supported PLC. Event Log. EasyConverter EasyAddressViewer Tool for analyzing EasyPrinter EasyDiagnoser Remote printer connection between /backup server. Recipe/Extended Memory Editor HMI and PLC. Build Download Data for CF/SD/USB Disk... Memory format conversion and data Download... editing. Upload... Build data for On-line Simulation... Off-line Simulation... downloading to HMI via CF/SD/USB. Pass-through... Allow PC applications to connect PLC via HMI. Help Exit

Click the buttons on the dialog box for detail.



2.1 HMI IP, Password

[Settings]

When operating HMI via Ethernet or USB cable, users need to designate the password for HMI to protect against unauthorized access.

[Reset / Download] functions share a set of password while [Upload] function uses another set.



Be sure to record any password change, otherwise, while resetting password to default, the project and data on HMI will be completely erased.

🦪 Project Manager					
HMI IP, Password					
Туре: МТ6000/8000	i Series 🛛 👻				
Settings	Reboot HMI				
Connection Ethernet	cable (i series only)				
HMI IP : 192.168.1.226	5 💌				
Data/Event Log	File Information				
Utility					
EasyBuild	der8000				
EasyConverter	EasyAddressViewer				
EasyPrinter EasyDiagnoser					
Recipe/Extended	d Memory Editor				
Build Download Data 1	for CF/SD/USB Disk				
Download	Upload				
On-line Simulation	Off-line Simulation				
Pass-thr	ough				
stop HMI scan font : finished reset HMI : finished					
Help	Exit				

[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. Set the correct IP address when operating HMI via Ethernet.

[Data/Event Log File Information]

After setting, connect with HMI to check the number of history files in HMI

📮 data log	
😑 test	
- 20110306.dtl	
- 20110303.dtl	
- 20110304,dtl	
- 20110305.dtl	
🖻 event log	
EL_20110303.evt	
EL_20110304.evt	



2.2 Utility

2.2.1 Build Download Data for Saving in CF/SD Card or USB Disk

- 1. Insert CF/SD/USB to PC.
- 2. Assign data storing path.
- 3. Assign files to download.
- 4. Build data.

The source files will be saved in the inserted device for users to download to HMI. This function is to build the required data.

Project Manager	
Select the folder to save download data :	2
KA	Browse
Sources	3
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	
4 PLEASE INPUT DATA LOG FILE NAME !	Browse
Build	Exit

2.2.2 Steps to Download Project to HMI via USB or CF/SD Card

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

- 1. Insert USB (project included.) to HMI.
- 2. On [Download / Upload] dialog box select [Download].
- 3. Input Download Password.
- 4. On [Download Settings] dialog box, check [Download project files] and [Download history files].
- 5. Press [OK].
- 6. On [Pick a Directory] dialog box, select directory: usbdisk/device-0/123.
- 7. Press [OK].

Project will be automatically updated.



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



2.3 Transfer

2.3.1 Download

Download source files to HMI through Ethernet or USB cable.

Firmware Check to update	Download		K
HMI kernel programs. Must			
do when first time download	Firmware		
data to HMI.	Project	C:\Documents and Settings\Administrator\桌面\MTP1.xob	Browse
	RW RW	PLEASE INPUT RECIPE FILE NAME !	Browse
Browse Click to assign	RW_A	PLEASE INPUT RECIPE 1 FILE NAME !	Browse
° √	🗹 Data log	PLEASE INPUT DATA LOG FILE NAME !	Browse
desired downloading path. $~~$ \checkmark	Destination :	PLEASE INPUT DESTINATION FOLDER	Browse
	Install X-series media-	-player drivers	
🗹 Install X-series media-player drivers	Startup screen	F:\testpic.bmp	Browse
	1		
Necessary when first time			
download data to X series			
HMI using EasyBuilder Pro.			
5 ,	Connection ③ Ethernet	OUSB cable (i series only)	
Startup screen (i series only)	1 IP Name	· · · · · · · · · · · · · · · · · · ·	<u>م</u>
Download assigned BMP to	IP:	~	
HMI. On HMI, it will be			
shown after rebooting then			
load in project. Users may			
use company logos.	Reboot HMI after dov	wnload Reset recipe Reset event	log
-	Reset data log		
Reboot HMI after download	Download	Stop Settings	Exit
Automatically reboot after			
download.			

[Reset recipe] / [Reset event log] / [Reset data log] Erase specified files on HMI before download.



2.3.2 Upload

Upload files from HMI to PC via Ethernet or USB cable.

Users have to assign the desired path for file storage before uploading.

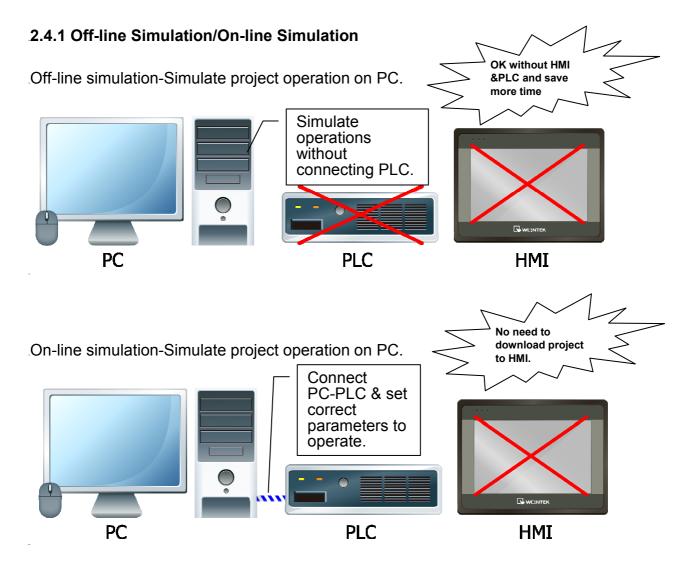
Upload			
	MT6000/8000 i Series		
Project	C:\Documents and Settings\Nicolas\456.xob	Browse	
RW	PLEASE INPUT RECIPE FILE NAME !	Browse	Click Browse
RW_A	PLEASE INPUT RECIPE 1 FILE NAME !	Browse	
🗹 Data log	C:\Documents and Settings\	Browse	 To assign desired
Event log	C:\Documents and Settings\	Browse	uploading path.
	☑Use CSV (Comma Seprated Values) format to save data/eve	nt log files.	
Extend Memory (EM)	PLEASE INPUT EM FOLDER NAME !	Browse	
	EM file location : OCF/SD card OUSB 1 OUSB 2	J	
Connection © Ethernet) USB cable (i series only)	4	
HMI name : Defa	Search all		
Upload	Stop Settings	Exit	



The file will be uploaded to PC in *.XOB file format. For editing this file using EasyBuilder Pro, please decompile it into *. MTP file first.



2.4 Simulation



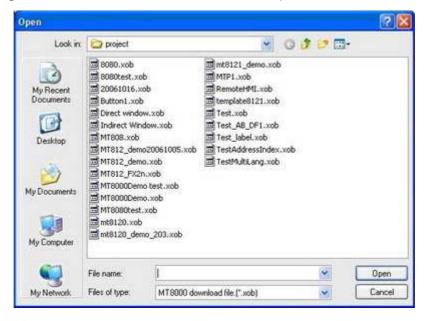


When On-line simulating on PC, if the control target is a local PLC (i.e. the PLC directly connected to PC), there is **10 minutes simulation limit.**





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



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



[Run EasyDiagnoser]

To monitor current communication status.

[Screenshot]

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



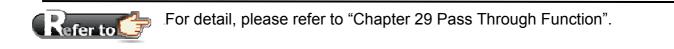
2.5 Pass-Through

This function allows the PC application to connect PLC via HMI. In this case, the HMI acts as a converter.



Pass-through provides two modes: [Ethernet] and [COM port].

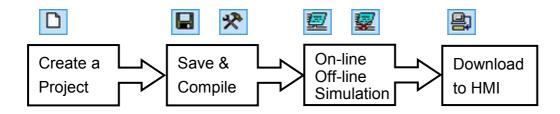
When using [Ethernet], please install the virtual serial port driver first.



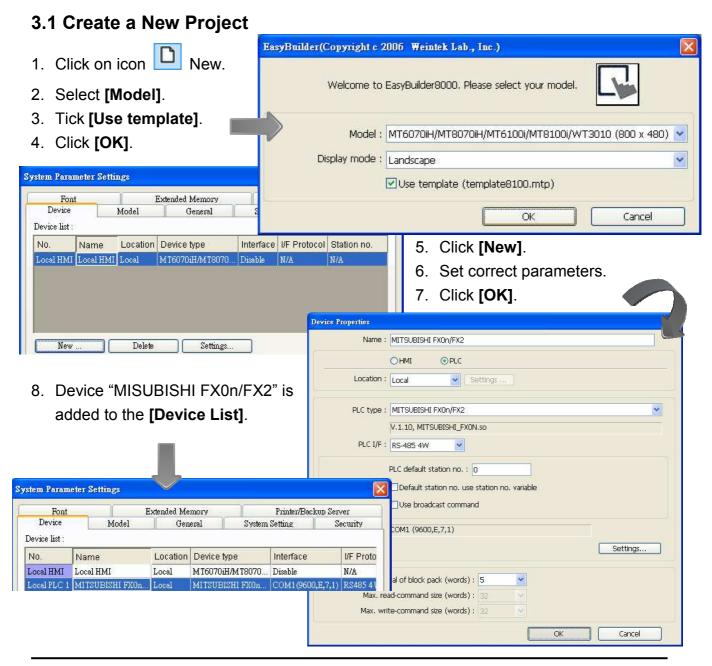


Chapter 3 Create an EasyBuilder Pro Project

Click on the icons to see illustration.



In this Chapter, we will take Mitsubishi PLC as an example.





Now let's add a new object.

- 1. Click on the object icon 🕥 Toggle Switch Object.
- 2. Set correct parameters.

New Toggle Switch Object									
General Security Shape Label									
Description :									
Read address									
PLC name : MITSUBISHI FX0n/FX2	Setti	ng							
Address : Y									
Invert signal									
- Write address :									
PLC name : MITSUBISHI FX0n/FX2	Setti	ng							
Address : Y 1	EasyBuilder 8000 : MTP2	- [1] - WINDO	₩ 010 1					ſ	
Attribute	EB File Edit View Option	Draw Objects	Library <u>T</u> ools	<u>W</u> indow <u>H</u> elp					
Switch style : Set OFF	D 🖨 🖬 ½ Pa Ra ≤	2015 ?	₩? %4 _	Ⅲ <u>本</u> 言4// ∧* 三 三 三	A STREET	and the second second			
		日日串	A second	1215 111	The second second			i咕le	10 II
	🔰 🛃 📝 🖌 🕅 🖌		🕌 🗛 🖪 🔟	* 🛛 💆		🕾 🎹 🚺	12		State
Place the object wherever	Windows \checkmark X 3: Fast Selection 4: Common Window		O₩_010 ×					► ►	
you like on window.	5: PLC Response 6: HMI Connection							₽ 1	
A project with one object is	- 7: Password Restriction - 8 - 9							0-14 	
now created.	➡ *10: WINDOW_010 TS_0 (Y-1, Y-1) (Bc	TS_0							
								ل ې م	
									• 📰
	17 18								∎ <u>⊡</u>
		-							
	- 25 - 26								B B
	- 27 28 29							~	



3.2 Save and Compile the Project

On EasyBuilder Pro Tool Bar:

<u>F</u> ile	Edit	<u>V</u> iew	Option	$\underline{\mathbf{D}}\mathbf{raw}$	⊇ł		
D	<u>N</u> ew			Ctd+	N		
2	<u>O</u> pen			Ctd+O			
	<u>C</u> lose						
	<u>S</u> ave			Ctd+	s		



1. Click to [Save] *.MTP file.

Tool	s	<u>W</u> indow	<u>H</u> elp		
*	C	ompile			
<u>2</u>	0	n-line Simu	lation		Ctd+I
₩	0	ff-line <u>S</u> imu	lation		Ctd+T
鲁	Download F7				
CF	в	uild Data fo	α USB Disk	or CF Card I	Download
⊿	R	eboot HMI.			
2 . C	lic	ck to [C	ompile	e] to *.X	OB file
for	d	ownloa	ding to	HMI, th	is also
che	ec	ks if th	e proje	ct can ru	un

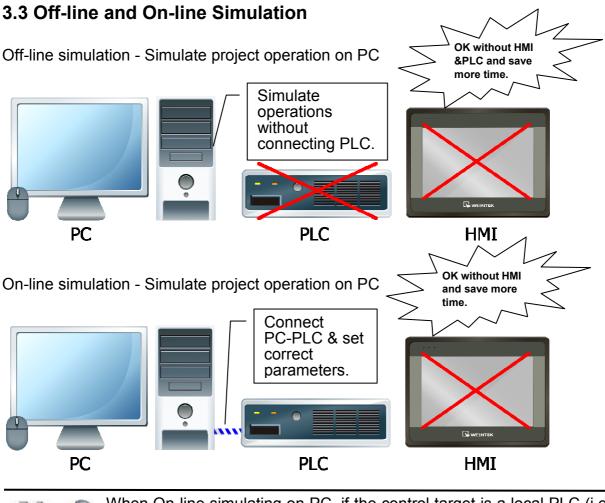
correctly.

	Compiling				X
Users are allowed	Project name : C:\Docum	ents and Settings\Administ	rator\桌面\MTP1.mtp		
to select the					
languages needed	XOB file name : C:\Docum	ents and Settings\Administ	rator\桌面\MTP1.xob		
for the project by					
clicking	XOB password : Set	(used in decompiler)	Decompilation	is prohibited	
[Language 1 to 8].	Select the languages used on th				
	Startup la	nguage after redownloadin	g the project : Language :	1 💌	
	Language 1	Language 2	☑ Language 3	☑Language 4	
	✓Language 5	✓ Language 6	☑ Language 7	Language 8	
	Oerror(s), Owarning(s)				<u>~</u>
	Object size : 44878 bytes Font size : 314444 bytes				
	Picture size : 792988 bytes Shape size : 842 bytes				
	Sound size : 36474 bytes Macro size : 14 bytes				
	Tag size : 20 bytes				
	Total size : 1189660 bytes ((1.13M)			
	succeeded				~
	Double click error messages to mo		ve objects !		Class
	Compile Bu	ild font files			Close

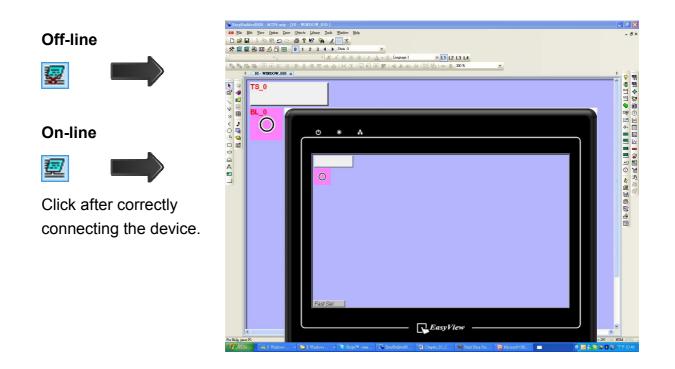
A successfully compiled file will get this dialog box.



ote



When On-line 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.**





3.4 Download the Project to HMI

■ Way 1 [Ethernet] / HMI IP

Before **[Download]**. be sure to check if all the S

before [Download], be sure to check if all the			<u>C</u> ompile	
settings are correct.		瓕	On-line Simulation	Ctd+I
		₩	Off-line Simulation	Ctul+T
		말	Download	F7
		CF	Build Data for USB Disk or CF	Card Download
		2	Reboot HMI	
Input [Password] & _ 🛛 🔤	nload			
	lete startup-screen file wnloading D:\Jane文件\EB file		iic.bmp	<u>^</u>
do	wnloading C:\EB8000434\firm: wnloading C:\EB8000434\firm: wnloading C:\EB8090434\firm:	ware\qui.i		
	whioading C:\E88090434\firm whioading C:\Documents and whioading C:\E88000434\font	Settings\Adr	ninistrator\桌面\MTP1.xob	
Update HMI kernel		3		
reb	boot HMI : finished shed			
when first time				
download files to HMI.	Ethernet OUSB cable (i series only) Password : Set			
	IP :			
✓ Font files				
Download the font				
	\			
used in project.	Firmware	Font files		
	'Necessary if update firmware	or execute	download first time.	
	☑Use user-defined startup scr			
The selected files will	File location : D:\Ja	ne文件\EB fil	e 和圖片\testpic.bmp	Browse
be erased before	Reset recipe	Reset ever	t log 🛛 🔽 Reset data sampling	2
downloading.	Reboot HMI after download	ŀ.		
		settings to d	ownload after compiling	
Reboot HMI after download	Download	Stop		Exit

Tools Window

 $\underline{H}elp$

HMI will reboot after downloading.

Automatically using current settings to download after compiling

If this is checked, system will download project to HMI according to last settings.

Please see illustration below.



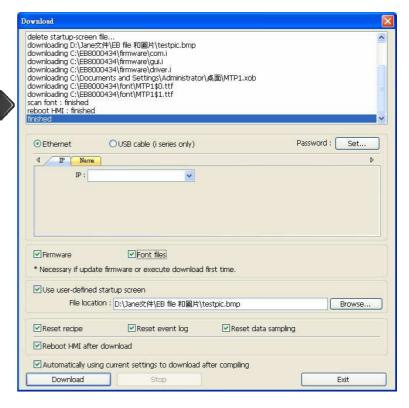
Automatically using current settings to download after compiling

Function Properties The way to enable this Display : 🗹 Object ID function: Display Common Window objects on Base Windows Option Draw Objects Using function key to make shape library object Grid/Snap... Function Properties... Automatic save and compile when download and simulate OK Cancel

- 1. Click [Function Properties].
- 2. Tick [Automatic save and compile when download and simulate].



- 3. [Save] project.
- 4. Click [Download].
- 5. On dialog box, tick [Automatically using current settings to download after compiling].
- 6. Click [Download].
- 7. After finish setting, next time when click [Download], EasyBuilder Pro will automatically compile and



download project to the latest target HMI.



■ Way 2 [Ethernet] / HMI Name

Automatically using current settings to download after compiling

Stop

Download

1. On HMI set HMI name first.

System settings		
Network ¹ Time/Date ¹ Securit / HMI name ¹ settin	apall	
HMI name test		Image: set of the set of
2. On PC, select the set HMI name and st	art downloading	
		Input the HMI name to search the designated HMI.
	4 IP Name	
	HMI name : test	
Ethernet OUSB cable (i series only)	*	Search all
HMI name : test	* OS 20091002 or later support	
Search Search all * OS 20091002 or later supports		Click to search the HMIs share the same network.
✓ Firmware ✓ Font files	4 IP Name HMI name :	-192.168.1.11 (Default HMI)
* Necessary if update firmware or execute download first time. Vuse user-defined startup screen File location : D:\Jane交件\EB file 和圖片\testpic.bmp		-192.168.1.118 (nicolas_8100i) -192.168.1.226 (test) Search Search all
Reset recipe Reset event log Reset data sampl	* OS 20091002 or later supports -	
Reboot HMI after download		

Exit



■ Way 3 [USB Cable]

Download O Ethernet O USB cable (i series only) Password : Set	Select USB cable to download project to HMI. The way of setting is same as Way 1 mentioned above. USB cable only works for i Series HMI.
Firmware Font files * Necessary if update firmware or execute download first time.	
Use user-defined startup screen	
Reset recipe Reset event log Reset data sampling	
Reboot HMI after download	
Automatically using current settings to download after compiling	
Download Stop Exit	

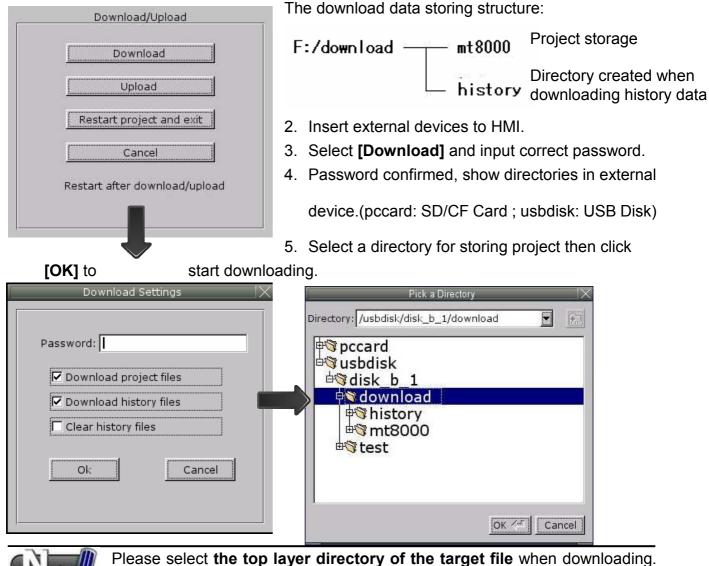
Before downloading via USB cable, please make sure the USB driver is correctly installed. Go to [Computer Management] / [Device Manager] to check if USB driver is installed, if not, please refer to installation steps to manually install.



■ Way 4 [USB Disk / SD Card]

F:\download	Browse
ources	
✓ 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

 In Project Manager click [Build Download Data for CF / SD / USB Disk] to build the data to be downloaded first. Generally divided into 2 directories, if set as the way shown:



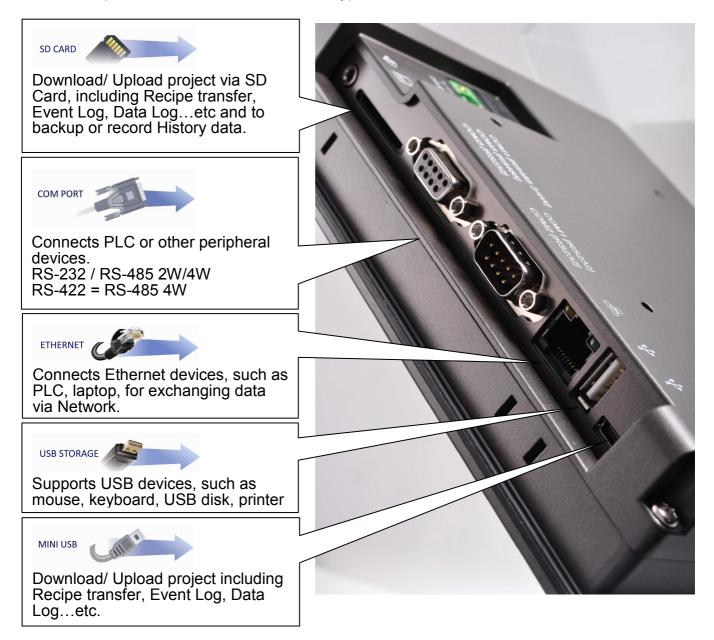
For the structure above, select *download*, not *mt8000* or *history*.



Chapter 4 Hardware Settings

4.1 I/O Ports of HMI

The I/O ports are different form one HMI type to another.



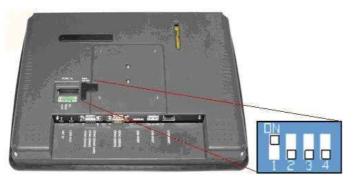
In addition, Weintek provides [MT8-COM1 Multi-Connector cable] and [MT8-COM3 Multi-Connector cable] to expand one COM port to multiple independent COM ports so that the convenience and efficiency of operation can be improved.



4.2 HMI System Settings

For the first time operating HMI, users have to complete the HMI system settings. After this, users can develop their own operation interface through EasyBuilder Pro editing software.

4.2.1 System Reset



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

If system password is lost or forgotten, please set DIP Switch 1 to "ON" and the rest remain "OFF", then reboot HMI. HMI will switch to touch screen

calibration mode.



A "+" sign appears on the screen, touch the center of the sign, after all 5 signs are touched, "+" disappears and the touch screen parameter will be stored in HMI system.
 After calibration, confirm to restore the system password to the default, select [YES].
 Confirm to restore to default password again by typing [yes] and clicking [OK]. The project files and history records stored in HMI will all be removed. (The default password is 111111. However, other passwords, including download/upload passwords have to be reset.)

SW1	SW2	SW3	SW4	Mode
ON	OFF	OFF	OFF	Touch screen calibration mode (X Series excluded)
OFF	ON	OFF	OFF	Hide system toolbar (T Series excluded)
OFF	OFF	ON	OFF	Boot loader mode (X Series excluded)
OFF	OFF	OFF	ON	Enable front panel power switch (X Series only)



The above shows the steps to

OFF OFF OFF OFF Normal

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.

Dip Switch



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 bottom-right corner of the screen will the System Toolbar pops up.



How to hide HMI System Setting Toolbar

EasyBuilder Pro supports the function of using system tag [LB-9020] to enable/disable system setting bar, or set the [DIP Switch 2] to ON/OFF for activating this function. When [LB-9020] is set ON, the bar is displayed, and set OFF to hide the system setting bar. When [DIP Switch 2] is set ON, the system setting bar is disabled, and when set OFF; the system setting bar is able to control. Users have to restart HMI to enable/disable this function.

Note: [LB-9020] is available for all HMI series. [DIP Switch 2] is available for i and X Series.





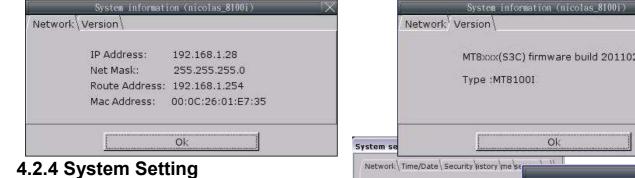


Screen Calibration shortcut, X Series only, other series, turn SW1 to ON. When X Series touch screen drifting problem occurs, please

connect an USB mouse to select this mode.

4.2.3 System Information

Network: Display network information & HMI IP. Version: Display HMI system version.



Set or modify system parameters. Confirm password for security.

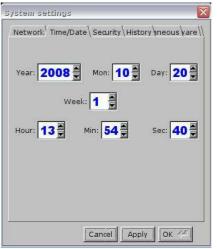
System settings 🛛 🔀	
Enter your password:	
Lincol your pubbillorun	
Deserver de L	
Password:	
Ok Cancel	

Network

Download project to HMI via Ethernet. Confirm IP address of target HMI. [Assign IP by local DHCP] or [Manually input IP information].

	System inform	ation (nicolas_8100i) 🛛 📉	
Netw	ork ¹ Version \		
	MT8xxx(S3C) firmware build 20110210 Type :MT8100I		
	Type Miloro	11	
		Ok	
em se			
twork\Time/Dat	e \ Security \istory \me \se		
Obtain an IP	Address Automatically		
O IP address g	et from below	Please enter you new password	
IP address	192 . 168 . 1 .		
Subnet Mask	255 . 255 . 255 .	password: *****	
GabeV/av	192 168 1	confirm: *****	
		Password match!	
		I OK I I Cancel I	

Time/Date Setting HMI local time/date.



Security

Password protection, default 111111.



Hardware Settings

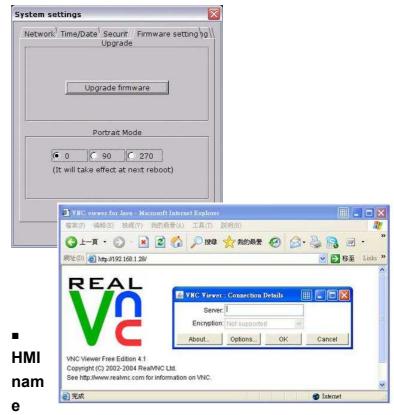


[Password for entering system] [Password for uploading project] [Password for downloading project] [Password for uploading history data] Password confirmation window:

ystem settings	System settings
/Network ¹ Time/Date ¹ Security ¹ Miscellaneou Backlight	/ Network ¹ Time/Date ¹ Security ¹ History ¹ / VNC server
Brightness	Start VNC
Download setting	Stop VNC
Popup download window	
Restart after download/upload	VNC login password
Cancel	Cancel Apply OK /*

■ Firmware setting

Upgrade firmware / enable portrait mode. (*i* series only)



Set HMI name to download/upload project.

em settings stwork ¹ Time/I	Date ¹ Securit	HMI name	setting \g	<u>×</u>
HMI name [Default HMI]	
	Cancel	Apply	ок 🖉	

• History Clear history data on HMI.

[Recipe] / [Eventlog] / [Datalog]

Clear Recipe
Clear Eventlog
Clear Datalog
Clear

Miscellaneous

Rolling button for adjusting LCD brightness.

37



VNC server

Remote HMI monitoring and controlling.

1. Enable HMI VNC server, set password.

2. Install Java IE or VNC Viewer on PC.

3-1 Input remote HMI IP in IE, example:

http://192.168.1.28。

3-2 In VNC Viewer input remote HMI IP and

password.

Serve	er: 192.168.1.28	
Encryptio	n: Not supported	4
About	Options Of	K Cancel
	entication (No Encry	ption (III) 💶 🗖
YNC Auth emame:	entication (No Encry	ption] (III) (III) (III)









☑One HMI allows only one user to log in VNC server at one time. When leaving VNC server unused for one hour, HMI system will log out

automatically.



Chapter 5 System Parameter Settings

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

Font		1.	Extended Memory	Printer/Backup Server		
Device		Model	General	Syste	m Setting	Security
evice list :						
No.	Name	Location	Device type	Interface	I/F Protocol	Station no.
.ocal HMI	Local HMI	Local	мт6070ін/мт8070	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
Location : Local Settings
Interval of block pack (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.

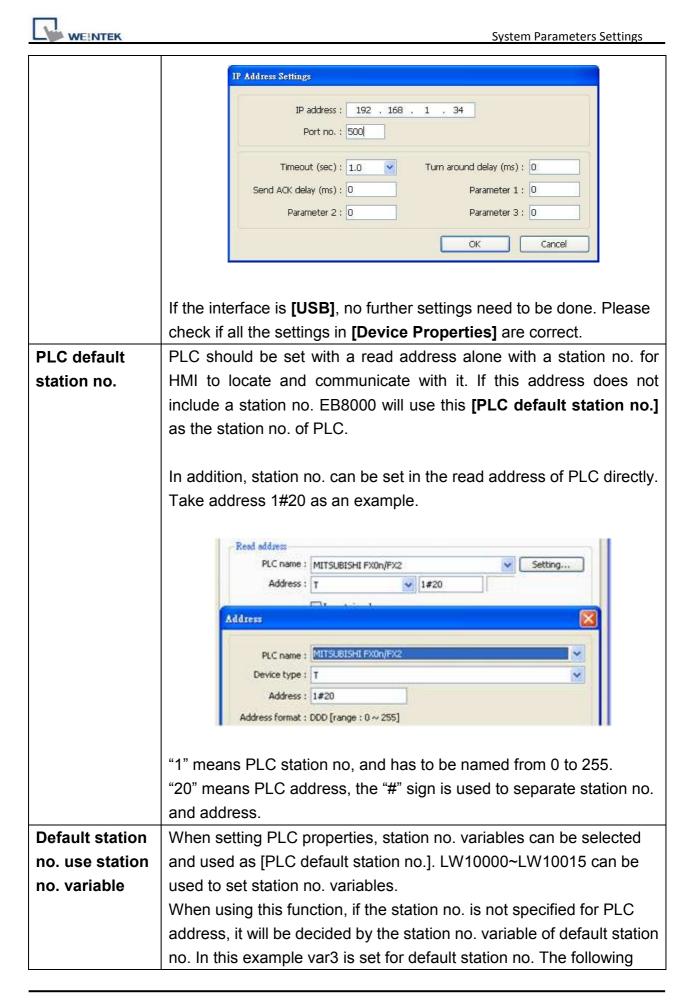
Take a local PLC MITSUBISHI FX0n/FX2 as an example:

Device Properties
Name : MITSUBISHI FX0n/FX2
Location : Local Settings
PLC type : MITSUBISHI FX0n/FX2
V.1.10, MITSUBISHI_FXON.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)
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	[Local] or [Remote]. Showing whether this device is connected to	
	Local HMI or being remote controlled. Select [Local] 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 Timeout (sec): 1.0				
	Baud rate : 9600 V Turn around delay (mr): 0				
	Data bits: 7 Bits Send ACK delay (mt): 0				
	Parity Even Parameter I 0 Stop bits : 1 Bit 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				
	SB_0 SB_1 FK_1 Close				
	TTown and date d				
	[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), slick (Cettings, Jacobiker, ID, Address)				
	If the interface is [Ethernet], click [Settings] and then [IP Address				
	Settings] dialogue appears. Users need to correctly set IP address				
	and Port no. of the PLC.				



	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).
	PLC name : 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).
	Use broadcast command Broadcast 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

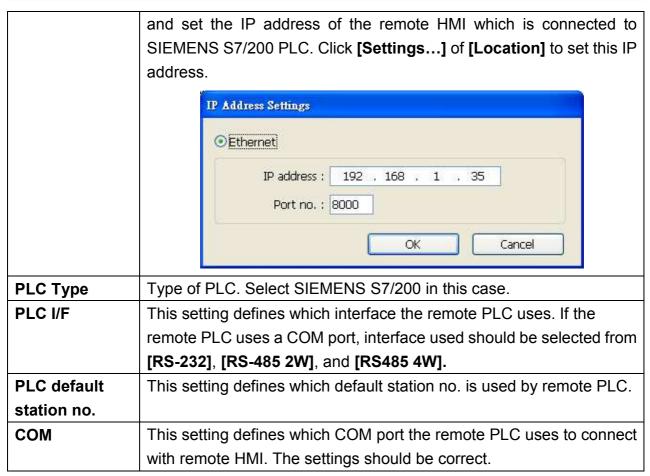


The so -called "remote PLC" means a PLC connected to a remote HMI. To control a remote PLC, users need to add this type of device. Click **[New...]** under **[Device list]** and the **[Device Properties]** dialog appears. Users need to set all the required properties correctly.

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

levice Properties	
Name :	SIEMENS S7/200
	OHMI ⊙PLC
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
Inte	rval of block pack (words) : 5
Max. re	ad-command size (words) : 32
Max. wr	ite-command size (words): 32
	OK Cancel

Setting	Description
HMI or PLC	This is to confirm whether this device is a HMI or PLC.
	It is [PLC] in this case.
Location	Users can select [Local] or [Remote]. Select [Remote] in this case



WEINTEK

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

No.	Name	Location	Device type	Interface
	Local HMI	Local	MT8121T (800 x 600)	Disable
Local	MITSUBISHI FX0n/F	X2 _{Ical}	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 Settings	IP : 192.168.1.11 (Port = 8000)
Interval of block pack (words) : 5	•
<u> </u>	
	OK Cancel

Setting	Description
HMI or PLC	This is to confirm whether this device is a HMI or PLC.
	It is [HMI] in this case.
Location	Users can select [Local] or [Remote]. Select [Remote] in this case
	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	MT8xxx	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

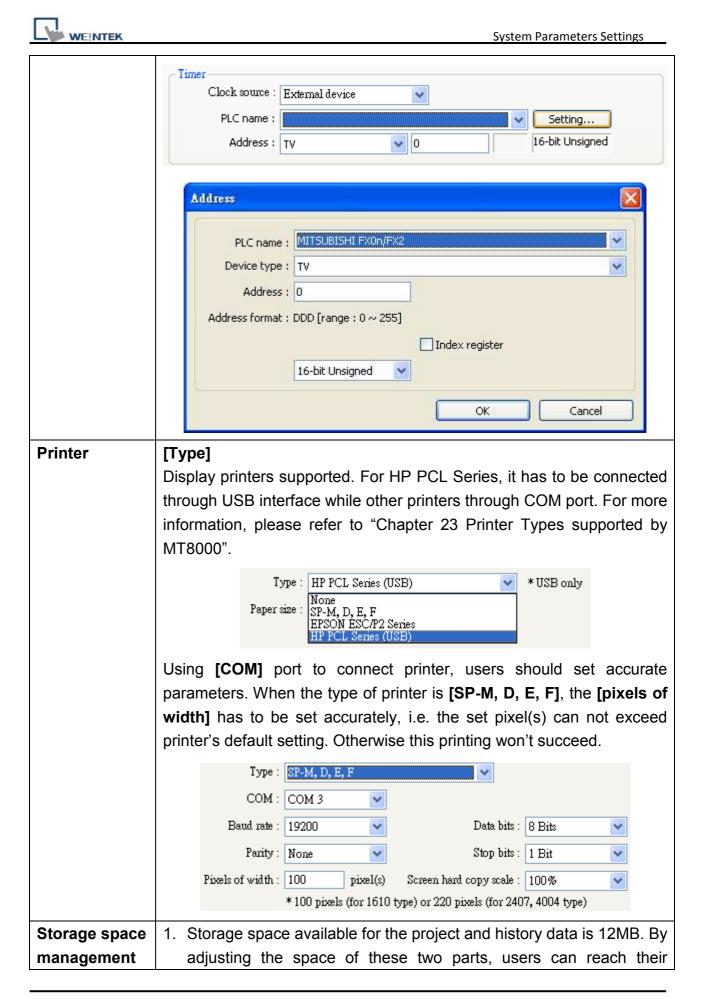
Parameters in [Model] tab determine the HMI model, [Timer] and [Printer] settings.

Font	1 3	Extended Memory		Printer/Back	up Server
Device	Model	General	System S	etting	Security
HMI model :	MT8121T (300 x 600)			*
HMI station no :	0	*			
Port no. :	8000	(used as	: MODBUS server'	s port no.)	
imer					
Clock source :	External dev	ice 🔽			
PLC name :	Local HMI			Settin	g
Address :	LW	✓ 0		16-bit Un:	signed
rinter					
Туре :	SP-M, D, E,	F	~		
COM :	COM 3	~			
Baud rate :	19200	~	Data bits :	8 Bits	~
Parity :	None	~	Stop bits :	1 Bit	~
Pixels of width :	100	pixel(s) Screen	n hard copy scale :	100%	~
	*100 pixels	(for 1610 type) or 2	220 pixels (for 240	7, 4004 type)
torage space manage	ment				
History data space	1.1.1.1	ana encencier	anga kacama	, Max.	XOB file size
4.0M			-0		8.0M

Setting	Description
HMI model	Select current HMI model as shown below.
	MT6056T/MT8056T (320 x 234) MT6056T/MT8056T (320 x 234) MT6070T/MT8070T (480 x 234) MT6104T/MT8080T/MT8104T (640 x 480) MT8121T (800 x 600) MT8104X (640 x 480) MT8104XH/MT8121X (800 x 600) MT8150X (1024 x 768) MT6070i/8070i (480 x 234) MT8070iH/MT6100i/MT8100i (800 x 480)
	When changing HMI model and press [OK], users will be inquired if



	System Parameters Settings			
	they would like to [Resize pop-up windows or objects].			
	Resize pop-up windows/objects			
	General windows Image: Resize pop-up windows Image: Resize objects			
	Keyboard windows Resize 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 [Port no.] 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 RTC] 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: 			
	$\begin{array}{rcl} TV & 0 & \rightarrow & Second \mbox{ (the limited range: 0~59)} \\ TV & 1 & \rightarrow & Minute \mbox{ (the limited range: 0~29)} \\ TV & 2 & \rightarrow & Hour \mbox{ (the limited range: 0~23)} \\ TV & 3 & \rightarrow & Day \mbox{ (the limited range: 1~31)} \\ TV & 4 & \rightarrow & Month \mbox{ (the limited range: 1~12)} \end{array}$			
	TV 5 \rightarrow Year (the limit range: 1970~2037)			





(For T series	memory requirements, for example, using smaller sized project to
only)	get bigger memory space for historical data. It works contrariwise.
	2. 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).
	3. For adjusting storage space, users should erase history data saved
	in HMI before downloading project file.
	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.
	TITUL : II CHARGE SIDIAGE SPACE, PREASE IESEL HIMI S DAIA 1025 AND EVENI 1025.





5.3 General

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

Font		Extend	ed Memory		Printer	Backup Server	
Device	Model		General	S	ystem Setting	Secu	ity
ast selection butto	m					intra conversion	
	Attribute : En	nable	*	Settin	1gs		
	Position : Le	ft	~	Hide butt	on when HMI s	tarts	
						7777C)	
Screen saver			and const				
Back lij	ght saver : 1		🚩 mir	ute (s)			
		Enable back	k light when als	rm occurs	\$		
Scre	en saver : 2)	💌 mir	ute (s)			
Savar win			1.010	13.82			
		. WINDOW	_010				
Options							
Startup win	idow no. : 10). WINDOW	r_010				*
Common	window : A	bove base w	indow 🔽	Objec	t layout : Natu	e	~
2000 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 10			Harris Harris	1000		220	Reacosal
Keyboard ca	ret color : 📕				🔽 R V	V_A enabled	
Zvent							
	V]Use L W94	50~9455 as tin	e tags of (event logs		
	E	CD	~				
P +	L. م		LICE				
Extra. no. o	of events : 0	č					
Keyboard ———	-						
	5	0. Keypad 1 1. Keypad 2	- Integer - Integer		^	Add	
	5	Keypad 3	- Integer		= 6	Delete	
	5	3. Keypad 4 4. Keypad 5	- Integer i - Integer			Dente	
	5	Keypad 6	i - Integer				
	5	6. Keypad 7	- HEX				
	(i series only)						
Project protection		1. 1. <u>1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1</u>	and the second		(ran ma : 0	4294901750)	
Project protection	nable	Project	key: 111111		(range . 0 ~	4294901700)	

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 V Left Right
	Select the position on the screen of HMI where this button appears. If
	[Left] is chosen, the button will show up on screen bottom-left; if [Right]
	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 [none] is set, the back light will always be on
	while using.
	b. [Screen saver]
	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.
	c. [Saver window no.]
	To assign a window for screen saver.
Option	a. [Startup window no.]
option	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	1	Printer/Bac	kup Server
Device	Model	General	System :	Setting	Security
	Startup languag	æ after redownloading ti	ue project : L	anguage 1	~
Execute init.	MACRO when p	ower on			
Auto logout					
🗹 Enable	1	💌 minute(s)			
When a user do logout.	es not operate the	HMI for longer than th	e setting time,	the system w	ill automatically
Hide system * Use LB-9062	setting bar to open hardware	Hide mou e setting dialog.	se cursor		
Disable buzz	er	Prohibit r	emote HMI co	onnecting to	this machine
— Disable unlo	ad function (effer	tive after rebooting HM	l)(or set LB90	- 133 on)	
<u></u>		l operation (or set LB90	영상 기가 방송 사가 가지?		
_		te operation (or set LB90			
		elative objects when PLC		ion fails	
VNC server					
Password fro	m project	Passwo	rd : 111111		
LW protection	remote-write (via	COM port or ethernet)			
222			~ 99		
222	L	W range : 0			
222	L	W range : 0			
☑ Disable L W R W protection		W range : 0 COM port or ethernet)			
☑ Disable L W R W protection	remote-write (via]~ [999		
☑ Disable L W R W protection	remote-write (via R	.COM port or ethernet)]~ [999		

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.

Font		Extende	ed Memory			Prir	nter/Back	up Serv	er
Device	Model	1	General	1	Syster	n Settin	٤	Se	curity
lect operatable c User 1	lasses for each u	ser	144						
🗹 Enable	Password :	0		🗹 A	В	□с	D	E	F
User 2 🗹 Enable	Password :	222		💌 A	₽B	□c	D	E	F
User 3 🔽 Enable	Password :	333		🗹 A	₽B	₽C	D	E	F
User 4 Enable									
User 5 Enable									
User 6 DEnable									

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	
	Class E Class F	
Sound	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	С	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.

Jær 10 DEnable	ITP file 🕅
Jser 11 Enable	Password : 19854561 (range: 1 ~ 4294967295)
User 12 DEnable	OK Cancel
oject password (M	TP file)
🗹 Enable	Setting

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

💊 EasyBuilder80(JU	
<u>File T</u> ool <u>H</u> elp		
0 🗃 🖬 🛛 🗸	EasyBuilder8000	3
4444	Password : *** (Limited to 3 tir	nes)
Windows	OK Cancel	 >
Object list	~	



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.

Device	Model	General	System Setting	Security
Font	I	Extended Memory		kup Server
onts for non-a	_			
3 of 9 Barcod Albany			Add	
Arial Arial Black Arial Bold			Delete	
Arial Narrow Arial Narrow I	Bold		Delete All	
Batang Bookman Old			L	_
Calibri Century Gothi	577.505405			
Comic Sans M Consolas				
Courier New				
Franklin Goth Garamond	c Medrum			
Georgia				
Julim				
Gulim Impact Latha				

[Fonts for non-strings]

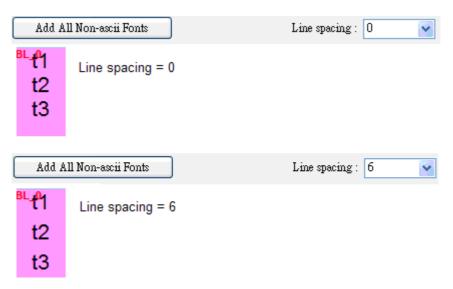
Fonts for non-ASCII strings are listed above. When users use non-ASCII character set or double byte character set (including simplified or traditional Chinese character, Japanese, or Korean) which is not listed in **[Fonts for non-ASCII strings]** table, EB8000 will select a font from the list to substitute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

File name : em4 emi

File name : em5:emi

File name : emf.emi

File name : em7.emi

File name : em8 emi

File name : em9 emi

EM5

EM6

EM7

EM8

EM9

Device	Model	General	S	ystem Setting	Securi
Font	E	xtended Memory		Printer	/Backup Server
EMO					
File name :	m0 emi	🔘 SD c	ard	💿 USB 1	OUSB 2
EM1					
File name :	uml.emi	◯ SD c	ard	💽 USB 1	O USB 2
EM2		11			
File name :	m2.emi		ard	💿 USB 1	O USB 2
EM3					
-76 782	m3.emi	◯ SD c	ard	💿 USB 1	OUSB 2

○ SD card

○ SD card

◯ SD card

○ SD card

○ SD card

O SD card

OUSB 1

⊙ USB 1

💿 USB 1

💿 USB 1

OUSB 1

OUSB 1

OUSB 2

O USB 2

OUSB 2

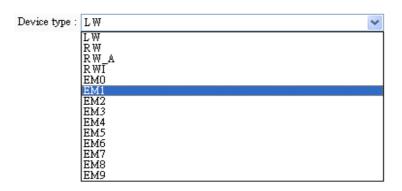
USB 2

OUSB 2

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.

Device	Model	General	System Setting	Security	
Font Extended Memory Printer/Backup Set					
Output settings Orientatio	nter to configur n : ⓒ Horizon re : ⓒ Origins	e PC for printing screen ha ntal 🔺 🔿 Vertical	cinter margins	data.	
User nar					

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

A window is a basic element in a project. With a window, all kinds of information like objects, pictures, and text can be shown on HMI screen. Generally, there's more than one window in a project, many windows will be constructed in one project. 1997 windows numbered from 3~1999 in EasyBuilder Pro can be built and edited. 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 i series HMI 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, each with different functions and usages:

- (1) Base Window (2) Fast Selection Window (3) Common Window
- (4) System Message Window

6.1.1 Base Window

The most frequently used window, used as:

- a. main screen
- b. background for other windows
- c. keyboard window
- d. pop-up window for [function key] object.
- e. pop-up window for [direct window] and [indirect window] objects.
- f. screen saver

The start up screen shown on the right is a Base Window.





Base window should be in the same size as the HMI screen. Therefore, the resolution of base window and HMI should be identical.

6.1.2 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 shown below:

The [Shortcut] button is a fast selection button used to show / hide fast selection window.

७ * ₺		
WEINTEK MT8000 series	1000	10000
Switch Lamp	Popup Windows	Scheduler
Trend Graphs Realog Meters	Data Block	Multiple Languages
Bar Graphs Numeric Entry	Audio	Alarms
Popup Windows Bais Block 17 True Type Fonts	Transferring Data	Backlight
Rudio Popup Menu IS	True Type Fonts	Popup Menu
Switch Lamp Animation Riarms	Multiple PLCs	Clock
Biose Shortcut Shortcut Switch Setting Bar	Hardware Expansion	Cursor
EasyView –		

Fast Selection Button setting dialog: [System Parameter Settings / General]

Or use system registers to control:

- [LB-9013] FS window control [hide(ON)/show(OFF)]
- [LB-9014] FS button control [hide(ON)/show(OFF)]
- [LB-9015] FS window/button control [hide(ON)/show(OFF)]



6.1.3 Common Window

Window no. 4 is the default Common Window. Objects on this window will be displayed on other base windows, but it does not include popup windows. Therefore, objects on different windows, whether shared or same, will be placed on common window, for example, the product logo, or a common button.

When system is in operation, select [Function Key] / [Change common window] to change the source of common window. For example, change the common window from window 4 to window 20.

New Far	iction Ke	y Object)		X
General	Security	Shape	Label		
	Description	1:			
		📃 Act	ivate after bu	tton is released	
0	Change ful	l-screen w	/indow	💿 Change common window	
01	Display pop	oup windo	w		
Ţ	Vindow no	.: 20. W	indow_020		~
<u></u>		10			

In [Option]/[Function Properties] select whether or not to [Display Common Window objects on Base Windows] when editing project. This can avoid overlapping objects on base window with objects on common window.

Function Properties
Display : 🗹 Object ID
Display Common Window objects on Base Windows
Using function key to make shape library object
Automatic save and compile when download and simulate
OK Cancel

6.1.4 System Message Window

Windows No. 5,6,7,8 are the default System Message Windows:

[Window No. 5: PLC Response]

When the communication between PLC and HMI is disconnected, this message window will pop up automatically right on the base window opened previously.

PLC no response	
Close	



■ "PLC no response" window can be set not to pop-up using system reserved registers. Please refer to "Chapter 22 System Reserved Words & Bits".

[Window No. 6: HMI Connection]

When failing to connect HMI with remote HMI, this message window will pop up automatically.



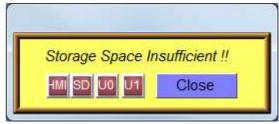
[Window No. 7: Password Restriction]

If user attempts to control an object without authorization, this window may pop up as a warning or not depending on how this object is set originally.



[Window No.8: Storage Space Insufficient]

When HMI built-in memory, USB disk or SD card run out of storage space, this message window will pop up automatically. (When system detects that memory space left is under 4MB)





Users can use system address tags 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/CF current free space (K bytes)
[LW-9076]	USB 1 current free space (K bytes)
[LW-9078]	USB 2 current free space (K bytes)
-	which device is insufficient in space while the insufficiency occurs, the em address tags can be used.
[LB-9035]	HMI free space insufficiency alarm (when ON)
[LB-9036]	SD/CF card 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 on windows no. 5~8 can be adjusted by users to fit what is needed, making the message easier to be understood by the operators.

■A screen can display 16 pop-up windows simultaneously in maximum including System Message Window, Direct Window and Indirect Window.

■A window can only be displayed once simultaneously. That is, users cannot use 2 Direct

(Indirect) windows to open the same window on one base window at the same time.

■ Windows 3~9 are for system use only while windows 10~1999 are for users to define.



6.2 Create, Set, and Delete a Window

<u>S</u> tandard Toolbar	dow_020		
<u>F</u> roject Toolbar	Windows	Windows	
Libray Toolbar	Object list	Window preview	
<u>T</u> ext Toolbar		0003	
<u>E</u> dit Toolber	- 4: Common Window		
State T <u>o</u> olbar	5: PLC Response 6: HMI Connection	Fast Selection	
Draw Toolbar	7: Password Restriction	0004	
PL <u>C</u> Toolbar			
Object 1 Toolbar	⊕ *10: WINDOW_010	Common Window	
Object 2 Toolbar	$-\frac{11}{12}$	0005	
Window Tree	13		

Go to EasyBuilder Pro / [View] / [Window Tree] to check the built windows.

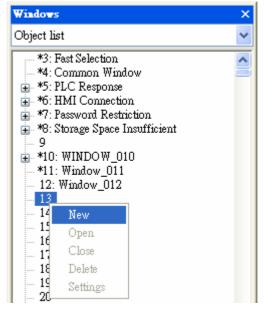
[Object List] displays window number and defined window names. The currently opened and edited window will contain a (*) mark, press the (+) beside the window number to see the objects, object ID, addresses and descriptions this window contains.

[Window Preview] displays windows in small pictures.

6.2.1 Creating and Setting a Window

■ Way 1

On window tree right click on a window number then select **[New]**.





	Window Settings
[Name] The name will appear on the + title bar and also in	Name : Window_013
window tree.	Window no. : 13
[Window no.] 4 3 ~ 1999	Size Width : 800 Height : 480
Set base window size identically to	Frame Width: 4 Color:
HMI resolution.	Background Color : Pattern :
	Filled Pattern color :
[Underlay window]	Underlay window
Refer to the	Bottom: 10. WINDOW_010
description below.	Middle : 11. Window_011
	Top: 12. Window_012
[Popup window]	Popup window
Base window can	Start pos. X : 0 Y : 0 Monopoly
also be used as pop-up window.	
Use [X] and [Y] to set the coordinates	OK
indicate where on the screen will this base window pop up. The origin of the coordinates is the upper-left 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.

[Underlay window]

One object can be placed on different windows (but not all windows). Underlay Window can be seen as an extra Common Window. The objects are placed on the Base Window where they are built. Up to three Underlay Windows can be defined by users.

Underlay Window is a base window which can be displayed simultaneously with the base window which calls it up. Up to three base windows can be specified as underlay windows for each base window, from **[Bottom]** to **[Top]**. The objects (but not the backgrounds) on underlay windows are displayed in this order on base window.



■Objects in the background can't be edited on the window they are shown, to edit objects on underlay windows, open the window they are built on using EasyBuilder Pro editing software.



∎ Way 2

EasyBuilder Pro / [Window] / [Open Window]

Click [New] to select the window style to be built and click [OK].

No.	Window name	Size		
3	Fast Selection	100,450	New	
4	Common Window	800,480	C	Base Window
5	PLC Response	300,100	Settings	
6 7	HMI Connection	300,100		Fast Selection
8	Password Restriction	410,100	Delete	1.445.4466446.1
8 10	Storage Space Insufficient WINDOW 010	256,80 800,480		
11	Window 011	800,480	Open	Common Window
12	Window 012	800,480	(open	
13	Window 013	800,480		
50	Keypad 1 - Integer	209,280		Exit
51	Keypad 2 - Integer	198,236		
52	Keypad 3 - Integer	200,170		
53	Keypad 4 - Integer	304,213		
54	Keypad 5 - Integer	160,230		
55	Keypad 6 - Integer	241,331		
56	Keypad 7 - HEX	306,223		
57	Keypad 8 - Floating	209,254		
60	ASCII Middle	576,240		
61 62	ASCII Small ASCII Upper M	480,200		
63	ASCII Lower M	576,240 576,240		
64	ASCII Upper S	480,200		
65	ASCII Lower S	480,200		
00	HOLIT LOTTOR	100,200		
			Exit	

There are three ways to call up [Window Settings] dialog:

∎ Way 1

Right click on the window number in the window select **[Settings]**.



∎ Way 2

EasyBuilder Pro / [Window] / [Open Window] click on the window to be set and then click [Settings].

∎ Way 3

On the window, right click when no object is and click **[Attribute]**.



6.2.2 Open, Close and Delete a Window

Open an existing window:

- 1. Double click on the window number in window tree.
- 2. In window tree select the window to be opened -> right click -> click [Open].

Close or delete an existing window:

Nearly the same procedure as the above, please note that to delete a window, it has to be closed first.

Chapter 7 Event Log



7.1 Event Log Management



Alarm Bar / Alarm Display / Event Display

Category

EasyBuilder Pro classifies events by dividing them into 0 ~ 255 categories. Select one category to add or view event log. In [] it shows how many events are in this category.

History files

Specify the storage device of an event log. However, when simulating the project on PC, the files will be saved under the installation directory. For USB numbering, the first inserted will be USB1, the second will be USB2, regardless of the interface position. Using these objects to view the process of the whole event from triggering→waiting for processing→until alarm stops. Define event content first.

No.
 -lis
-lis [



[Preservation limit]

This setting determines how many days the data to be preserved. For example, the **[Days of preservation]** is set to 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.

Print

To enable this setting, please finish the settings of printer in [System Parameter Settings]/ [Model].

7.1.1 Excel Editing

Use Excel to edit [Event Log].

×

Click on the Excel icon on Event Log setting dialog to open the Excel template for editing. This template is under installation directory - EventLogExample.xls and includes ready made dropdown lists and validation mechanism.

	Α	В	С	D	E	F	G	Н	Ι	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	−ue
5										16-bit BCD 32-bit BCD	
6										16-bit Unsigned 16-bit Singed	
7										32-bit Unsigned 32-bit Signed	
0										52-bit atglied	



- [System tag] and [User-defined tag] can not be set to true simultaneously, otherwise, the system will view [System tag] to be true, and [User-defined tag] to be false. If setting [User-defined tag] as [Device type], please set [System Tag] to be false.
 - 2. [Color] format is R:G:B, each should be an integer form 0 ~ 255.
 - 3. When setting [User-defined tag] to be true, if the system compares the [Device type] with the user-defined tag in system, and no suitable tag is found, the system will set the user-defined tag in event log to be false.
 - 4. Before importing Library (Label Library / 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.



7.2 Create a New Event Log

7.2.1 Alarm (Event) Log General Settings

Click [New], appears the [Alarm (Event) Log] dialog which includes two tabs, go to [General] tab.

[Category]

Select event category, 0 ~ 255.

[Priority level]

When the number of Event Log equals to the max number available in the system (default 1000), the lower priority events will be deleted and new events will be added in.

[Read address]

System reads data from this address to check if the event matches the trigger condition.

[Notification]

When enabled, system will set the specified register to ON or OFF when the event is triggered.

[Condition]

When [Bit] is selected, Event Log will detect the ON or OFF state of a Bit address.

General	Message					
	Category :	0		Priority level	Low	~
Ad	dress type :	Word	*			
Read	address					
	PLC name :	Local HMI			2	Setting
	Address :	LW	~)	16-	bit Unsigned
Notific	ation	🗹 Enable		◯ Set Of	N Se	et OFF
	PLC name :	Local HMI			~	Setting
	Address :	LB	~)		
Condit		alue is : ==	v 0	<u> </u>		
Ir	tolerance :	0.1		Out tolerance	0,2	

When [Word] is selected, Event Log will detect the value of a Word address to check if it equals to, greater than, or less than a specified value.

Example 1	
	Condition
	Trigger if value is : 30
	In tolerance : 1 Out tolerance : 2

The setting above indicates:

When **[Read address]** value is greater than or equals to 29 (= 30 - 1) and less than or equals to 31(= 30 + 1), the event will be triggered. The trigger condition:

29 <= [Read address] value <= 31

After the event is triggered, when **[Read address]** value is greater than 32(= 30 + 2) or less then 28(= 30 - 2) the system will 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	

The setting above indicates:

When [Read address] value is less than 29 (= 30 - 1)

or greater than 31(= 30 + 1), the event will be triggered. The trigger condition:

[Read address] value < 29 or [Read address] value > 31

After the event is triggered, when **[Read address]** value is greater than or equals to 28(= 30 - 2) and less than or equals to 32(= 30 + 2) the system will return to normal condition:

28 <= [Read address] value <= 32



7.2.2 Alarm (Event) Log Message Settings

Alarm (Event) Log [Message] tab:

[Content]

The text content of Event Log shown in [Alarm Bar], [Alarm Display] and [Event Display] Please see the examples next page.

[Font] / [Color]

The font and color can be set differently for each event. The font and color shown in [Alarm Bar], [Alarm Display] or [Event Display] come from this setting.

[Write value for Event/Alarm Display object]

When an event in [Event Display] or [Alarm Display] is acknowledged, the value is written to the assigned address.

[Sound]

When enabled, a designated sound will be played when an event is triggered. Continuous beep can be set which will only stop when the event is acknowledged or recovered.

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

An illustration of how the beep is related to the event:

arm (Event) Log	
eneral Message	
Text	
Content :	Event 3
	Use label library Label Library
Font :	Arial
Color :	
Write value for Eve	ent/Alarm Display object
Write value :	11
Sound	
🗹 Enable	Sound Library 15winSpaceRestoreUp (28 k)
🗹 Enable con	tinuous beep until acknowledged or recovered
<u>N=</u>	Time delay of continuous beep : 10 second(s)
	Play
-	
Print	On trigger Return to normal
Font size :	
	CH1, WATCH2, WATCH3, WATCH4
Multi-watch :	1 Syntax
PLC name :	Local HMI Setting
Address :	LW 0 16-bit Unsigned
	OK Cancel Haln
	Acknowledge
[
Event	
[User defined
Sound	
10	seconds The time interval for beep sound
	is once in each second





Example 1

The data of LW address of the triggered event can be included in the content: Format: **%#d** (% = initial sign # = address d = end sign) When an event is triggered, if LW-20 = 13: Setting: "High Temperature = %20d"→ Display: "High Temperature = 13"

Example 2

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, take MW address as example:

Format: **\$#d** (**\$** = initial sign **#** = address d = end sign)

When an event is triggered, if MW-15 = 42:

Setting:"High Temperature = \$15d" → Display:"High Temperature = 42"

[Address of Watch] Click [Syntax] to edit and display the value in watch address when the event is triggered. Up to four watch addresses can be set.

	ntax to embed PLC data in the content of an event log.		
Isage			
%(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.		
xamples 1.Pressure = %	s(WATCH1)d.1		
	1 is %(WATCH1)f.2, Temperature2 is %(WATCH2)f.2		
3.Alarm : IP = 9	%(WATCH1)X : %(WATCH2)X : %(WATCH3)X : %(WATCH4)X		
4.Counter is %	(WATCH3)d		
5.Message = %	(WATCH1)s, Index = %(WATCH3)d		
	EXIT		

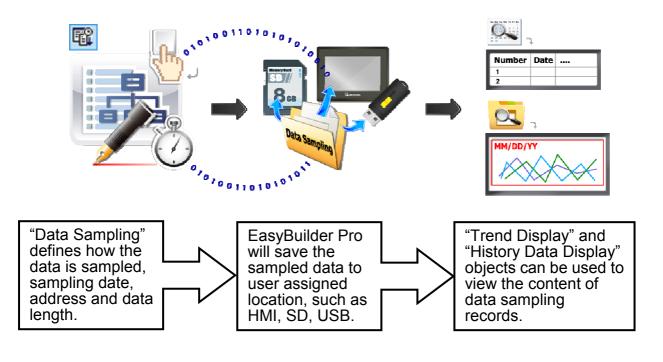
7.3 Event Log Relevant Registers

EasyBuilder Pro	provides the following	system 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

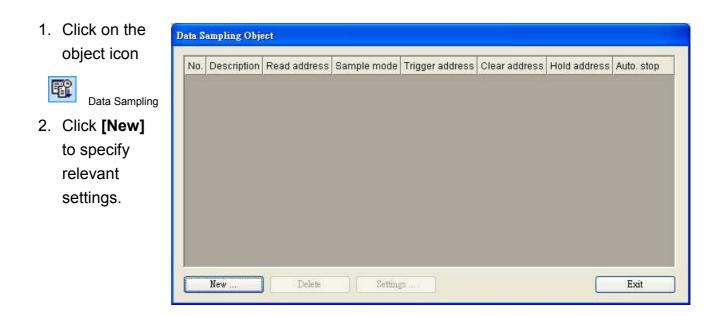


Chapter 8 Data Sampling



8.1 Data Sampling Management

Please define how the data is sampled before using Missing Trend Display or History Data Display to review the content of Data Sampling.





8.2 Create a New Data Sampling

Click on any of the fields on the dialog box for detail.

Data Sampling Object	
Description : Sampling mode Time-based 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 Data length :	Hold address
	OK Cancel

[Sampling mode]

[Time-based] mode samples data in a fixed frequency. The **[Sampling time interval]** can be defined from every "0.1 second(s)" to every "120 mins".

Sampling mode	I	-
 Time-based Trigger-based 	0.1 second(s)	
Time-based Contraction		
	0.3 second(s)	
	0.4 second(s)	
	0.5 second(s)	4
	0.6 second(s)	
Sampling time interval : 1 second(s)	0.7 second(s)	
samping the interval . I second(s)	0.8 second(s)	
	0.9 second(s)	
	1 second(s)	
	2 second(s)	
	3 second(s)	

[Trigger-based] mode uses the status of specific address to trigger Data Sampling.

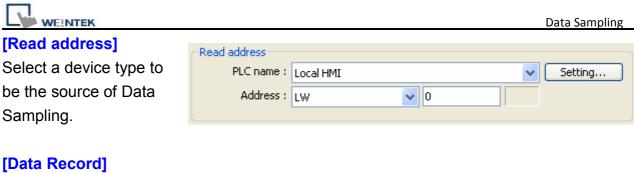
🔘 Time-based		Mode : OFF-		OFF->ON ON->OFF
PLC name :	Local HMI	¥	Setting	OFF<->ON
Address :	LB	160		

Conditions to trigger Data Sampling:

[OFF -> ON] Trigger when the status of assigned address changes from OFF to ON.

[ON -> OFF] Trigger when the status of assigned address changes from ON to OFF.

[OFF <-> ON] Trigger when the status of assigned address is changed.



	⊂Data Record	
Max. data records : 1000	Max. data records : 1000	🔄 Auto. stop
Max. number of data	Data Format	Data length : 0 word(s)

records can be saved by

one Data Sampling in one day is 86400. (1 record per second for 24hours) If **[sampling time interval]** is set to "0.1 second" then the max number of data records is 86400 only.

Auto. stop		
Condition	[Max. data records]: "10"	[Max. data records]: "10"
	& don't tick [Auto. stop]	& tick [Auto. stop]
	Delete earlier sampled data	Stops after reaching 10 data
Trend Display Real Time	and display the latest 10	records.
	records on "Trend Display".	
	Keep on sampling data and	Stops after reaching 10 data
Trend Display Historical	display all history data on	records.
	"Trend Display".	
	Keep on sampling data and	Stops after reaching 10 data
History Data Display	display all history data on	records.
	"History Data Display.	
R ?	Keep on sampling new	Stops sampling after
Data Sampling	data.	reaching 10 data records.

Data Format			
Data Format			
<mark>0. "index"</mark> 1. "Pressure 1" 2. "Temperature"	16-bit Unsigned 16-bit Signed 32-bit Float		New Delete Settings
ata Format Description :	index		
Read Address			
Data type :	16-bit BCD 16-bit BCD 32-bit BCD 16-bit Unsigned 16-bit Signed		
Save to default	32-bit Unsigned 32-bit Signed 32-bit Float	OK	Cancel

A Data Sampling may include more than one type of records. Data Sampling in EasyBuilder Pro is able to retrieve different types of records at the same time. Users can define the content of Data Sampling. As shown, user defines three types of data with data length 4 words in total. In this way, EasyBuilder Pro retrieves a 4-words-lengthed data each time from the assigned address to be the content in one Data Sampling.





If you have run the simulation and the sampling data is saved in the record, then you want to change the format of sampling date, be sure to delete previous data record in EasyBuilder Pro installation directory to avoid the system misinterpret the old data record.

[Clear address]

If the status of the assigned address is set ON, the data obtained by "Trend Display" [real-time] mode will be

cleared and the number

Clear address			
	🗹 Enable		
PLC name :			Setting
Address :	LB	✓ 0	

of data sampling returns zero. This won't affect the sampled data that is already saved in file.

[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.

Hold address Enable PLC name : Local HMI Setting... ✓ 0 Address : LB

[History files]

[Save to HMI]

Save Data Sampling to HMI only when its size reaches "4kb", or, use [LB-9034] to force storing data.

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

[Save to SD card / USB1 / USB2]

Save Data Sampling to the specified external device.

Note that the USB ports are not designated as no.1 or no.2. The USB device plugged in first is USB1; the USB device plugged in next is USB2.

[Folder name]

Specify Data Sampling file name which must be in ASCII characters.



The folder name will be written as: [Storage Location] \ [Folder Name] \ yyyymmdd.dtl

Preservation limit

Days of preservation : 2 day(s)

This determines how many days the data to be preserved. "2" days means the data of yesterday and the day before yesterday will be kept. Data not built in this period will be deleted to prevent the storage space from running out. EX: if today were July 1st, data of June 30th and June 29th will be preserved and data of June 28th be deleted.



When running simulation on PC, all data sampling will be saved to the **datalog** folder which is under the directory of **[Storage location]**.



8.3 System Registers Relevant to Data Sampling

EasyBuilder Pro 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



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

В	✓ 0	IDX 0	
Jr			
ocal HMI			
.В			
Ð	System	itag	
DDDD [range : 0 ~	11999]		
	100 C 2 C 2 C		
	В	B System	B System tag



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].

PLC name	: Local HMI	*
Device type	LB	
Address	LB RW_Bit	
Address format	RW_A_Bit	
	RBI	
- Index		

[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.

PLC r	name : Local HMI	[
Device	type : LB-9000 :	initialized as ON	
Ade	dress : LB9000	System tag	
Address fo	ormat : DDDDD [ra	ange : 0 ~ 11999]	
		- Index register	

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".





ltess		
PLC name :	Local HMI	~
Device type :	LB-9000 : initialized as ON	~
Address :	LB-9000 : initialized as ON	N
Address format :	LB-9001 : initialized as ON 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].

neral Security Shape Label	
Shape Shape Lit	orary
🗹 Inner 📃	l
Frame	·
Interior pattern :	Pattern Style
Duplicate these attrib	utes to every state
Picture	
Picture Li	brary
Preview	
	Version (Press
0 1	State : 0
	State : 0 💌
	Picture : Red
	Picture : Red BMP
	Picture : Red BMP
	Picture : Red BMP
0 1	Picture : Red BMP

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:	
Define Custom Colors >>	Hug: 160 Red: 0 Sat: 240 Green: 0 Color/Solid Lum: 120 Blue: 255
OK Cancel	Lum: 120 Blue: 255 Add to Custom Colors

[Frame]

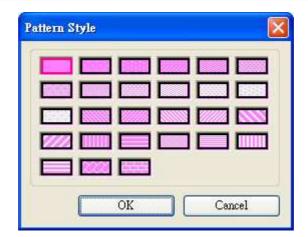
Tick [Frame] to enable this setting and select a **[color]** for the frame of the shape. The way of setting is same as above.

[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.



[Duplicate these attributes to every state]

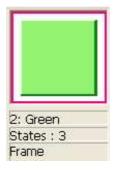
Duplicate all attributes of the current state to other states.

How to set [Shape Library...]

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



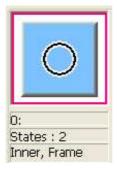




The illustration above provides information of one of the Shapes in the Shape Library as follows:

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.

State :
Picture :
P



9.2.2 Settings of Picture Library

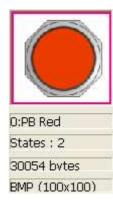
[Picture Library]

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.

brary :			State : 0		12345
utton1 con1				\bigcirc	
		D: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)
tate 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 (100×100)	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	Cancel
			Insert Before	Insert After	7



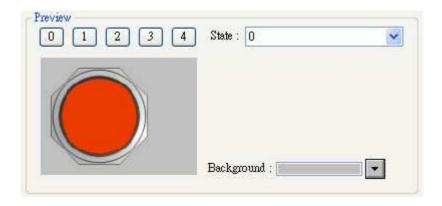


The illustration above provides information of one of the Pictures in the Picture Library as follows:

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.

eneral Security Shape Label	
🔽 Use label	
🔲 Use label library	
Use bitmap font	Label Library
State : 0] []
Font : Arial	~
Color:	Size : 16 💌
Align : Left 🔍	Blink : None 🗸 🗸
Every state	
Movement Direction : Left	Continuous
Speed :	
1 🖵	
Content :	
Content : ¢st	2

[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 :	Arrial	~
	Arial Arial Black Arial Bold	^
Align :	Arial Narrow Arial Narrow Bold Batang	
	Book Antiqua Book Antiqua Bold	Ξ
	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].

111
222222
333333333

The text aligned [Center].

111				
222222				
3333333333				

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

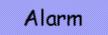
[Movement] setting

[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]

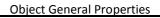
Set the content of the text. If using **[Label Library]**, the content will be sourced from Label Library.

[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.

Position Pinned X : 14 Y : 40	eneral	Shape	Label	Profile		
Prinned X: 14 🙄 Y: 40						100
] Pinned		X : 14	Y	: 40
ize			W	idth : 50	Height	: 5

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.

lress		E
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.

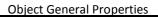
16 variables are available : var0 ~ var15. These variables of station number read values from address LW-10000~LW-10015. The list below shows variables and its corresponding system reserved address LW :

var0	LW-10000
vor1	LW-10001
var1	LVV-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



Object General Properties

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:

V	.1.00, MITSUBISHI_FXON.	50	
PLC I/F : R	(S-485 4W 🔹	PLC default station no. :	0
COM : C	OM1 (9600,E,7,1)		Setting

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

Object Security in EasyBuilder Pro includes:

- User password and corresponding operable object classes.
- 2. Security settings of each object.



10.1 User Password and Operable Object Classes

[System Parameter Settings] / [Security]:

Password should be digits from **0 to 9** and up to **12** sets of user password are available. There are seven security levels, classified from **A to F** and includes **none**.

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

Font	Ext	ended Memory		Printer/	Backup Server
Device	Model	General	Syste	m Setting	Security
M Enable	Password . 11		A DD		
🗹 Enable	Password : 11		A DB]D 🗹 E 🔲 F
User 2	St. 52. (1998)		20 19 -	-]D 🗌 E 🔲 F
Enable	Password : 22	9. I I I I I I I I I I I I I I I I I I I	0 L 41 T		



10.2 Object Security Settings

[Safety control]

To prevent miss-operation. [Min. press time (sec)]

Continuously press the object longer than the time set here to activate the object.

	a		(and a second s	
neral	Security	Shape	Label	
atety	control —			Min. press time (sec) : 0
_				

[Display confirmation request] After pressing the object, a dialog appears for operation confirmation. If response to this dialog comes later than the set [Max.

waiting time (sec)], this dialog disappears

automatically and the operation will be canceled.

[Interlock]

When ticked, whether this object can be operated depends on the state of the specified Bit address.

As shown, if LB-0 is ON, the object can be operated.

[Hide when disabled] When the specified Bit is OFF, hide the object. [Grayed label when disabled] When the specified Bit is OFF, the label of the object turns gray.

Interlock

Ve interlock function

Graved label when disabled

PLC name : Local HMI

• Enable when Bit is ON

Address : LB

🗹 Hide when disabled

[User restriction]

Only when user's permitted class matches the object's can it be operated.

[Object class] "none"

	toggie
User restriction Object class : Class A	
Disable protection permanently after initial activation	
Display warning message if access denied	
Make invisible while protected	

✓ 0

means any user can operate this object.

[Disable protection permanently after initial activation] Once the permitted class of the user matches that of the object, the system will stop checking the security class

permanently; even a different user can operate freely.

[Display warning message if access denied] When

the classes of user and object do not match, a warning dialog (Window 7) appears. The content of the message can be modified.

Password I	Protected! Acc	ess Denied !!
	Close	

[Make invisible while protected] When the classes of user and object do not match, hide the object.



Setting...

C Enable when Bit is OFF



10.3 Setting Example

Example 1

1. Create a project, go to [System Parameter Settings] / [Security] to enable 3 users: User 1 = Operable class: A System Parameter Settings User 2 = Font Extended Memory Printer/Backup Server Security Model General System Setting Device Operable class: A, B Select operatable classes for each user User 3 =User 1 🗹 A 🔲 B 🔂 C 🛄 D 💭 E 🕞 F 🗹 Enable Password : 111 Operable class: A, B, C User 2 Password : 222 A B C D E F M Enable User 3 A B C D E F Password : 333 Enable 2. Design Window 10: LW9219 #### User name : [Numeric Input] object [LW-9219] user no. (1~12) LW9220 Password : #### Length = 1word (16-bit Unsigned) bit 15 bit 0 [LW-9220] password 11111111111111111111 LW9222 Current status : Length = 2 words LB9050 (32-bit Unsigned) Logout [Numeric Display] object [LW-9222] current user Class A Button operable classes (16-bit Binary) Class B Button [Set Bit] object [LB-9050] user logout Class C Button

Three [Set Bit] objects, each

set to different classes but all select [Made invisible while protected].

After designing and setting the objects, please save, compile the project and do off-line simulation. Below shows how it works when simulating.



3. When no password is entered yet, it displays "0000000000000", meaning user operable object class "none". [SB_0] ~ [SB_2] objects are classified "A" ~ "C" and selected [Made invisible while protected], therefore they are hidden at this moment.

User name : 1 LW9219 Password : 0 LW9220 bit 15 bit 0 Current status : 00000000000000 LW9222 Logout LB9050	4. Input User 1 password "111". Since User 1 is only allowed to operate class A objects, [SB_0] object will appear for operating. [LW-9222] bit 0 turns "1", meaning user operable class: A.
5. Input User 3 password "333". Since User 3 is allowed to operate class A, B, C objects, [LW-9222] bit 0 ~bit 2 turns "1", meaning user operable class: A ~ C.	User name : 1 LW9219 Password : 111 LW9220 bit 15 bit 0 Current status : 00000000000001 LW9222 Logout LB9050 Class A Button
User name : 3 LW9219 Password : 333 LW9220 bit 15 bit 0 Current status : 00000000000111 LW9222 Logout LB9050 Class A Button Class B Button Class C Button	6. Click [LB9050] user logout, the system will return to initial state, current user car only operate class "none" objects.





■ **Password input** If the password is incorrect, [LB-9060] will be set to ON; if the password is correct, [LB-9060] will return OFF.

User 1~12 password can be read from system registers [LW-9500] ~ [LW- 9522], 24 words in total.

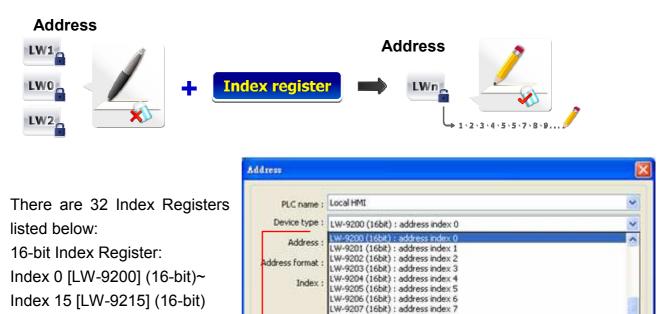
■ Change password when HMI is in operation When [LB-9061] turns from OFF to ON, data in [LW-9500] ~ [LW-9522] can be used to update user password, and use the new password in the future. The user operable object classes will not change due to the change of password.

Chapter 11 Index Register

11.1 Introduction

Max. range: 65536 words

EasyBuilder Pro provides 32 Index Registers for users to change addresses flexibly. With Index Register, users can update object's read/write address without changing its content while HMI is running the project.



	Tag Library LW-9209 (16bit) : address index 19 LW-9210 (16bit) : address index 11 LW-9212 (16bit) : address index 12 LW-9213 (16bit) : address index 13	
	LW-9215 (16bit) : address index 14 LW-9215 (16bit) : address index 15	
32-bit Index Register:	LW-9219 (16bit) : user no. (1~12)	
Index 16 [LW-9230] (32-bit)~	LW-9220 (32bit) : password LW-9222 (16bit) : classes can be operated for current user (bit 0:A, bit 1:B,bit : LW-9220 (32bit) : address index 16	
Index 31 [LW-9260] (32-bit)	LW-9232 (32bit) ; address index 17 LW-9234 (32bit) ; address index 18	
Max. range: 4294967296	LW-9236 (32bit) : address index 19 LW-9238 (32bit) : address index 20	
words	LW-9240 (32bit) : address index 21 LW-9242 (32bit) : address index 22 LW-9244 (32bit) : address index 23 LW-9246 (32bit) : address index 24 LW-9248 (32bit) : address index 25	
	LW-9250 (32bit) : address index 26	а.

LW-9208 (16bit) : address index 8

LW-9209 (16bit) : address index 9

Index 0 ~ Index 31: Descriptions of Index Registers.

[LW-9200] ~ [LW-9260]: Index Registers word addresses.

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.

[Index register] not checked:	Address			×
Read address is set to [LW-10]	PLC name :	Local HMI	~	
and won't change while	Device type :		V	Ì
running project.	Address :	10	System tag	-
	Address format :	DDDDD [range : 0 ~ 10799]]	
			Index register	
		16-bit Unsigned 🛛 💟		
	Tag Library		OK Cancel	

[Index register] is checked	Address			
and index register [INDEX 0] is	1			
selected: read address is set to	PLC name :			~
[LW-0 + INDEX 0]	Device type : Address :	1	System tag	×
INDEX 0: Index Register 0	Address format :	DDDDD [range : 0 ~ 10799]	
or data of address	Index :	INDEX 0 (16-bit)	🛛 🗹 Index register	
[LW-9200].		16-bit Unsigned 🛛 👻		
If data of address [LW-9200] is	Tag Library		ОК	Cancel
"5", read address is set to				
[LW(0+5)] = [LW-5].				

Here's a demo project shown as an example:

Demo Project - Index Register			
Index Function of Word			
0000	0000	0003	0006
Index 0 (LW-9200)	LW-0 + Index 0 (LW-9200)	LW-10	LW-20
Index Function of Bit	()	<u> </u>	
0000		\bigcirc	\bigcirc
Index 6 (LW-9206)	LB-0 + Index 6 (LW-9206)	LB-16	LB-32

		Index Register
Example 1 Index Fun	ction of Word	
Index Function of Word 0000 Index 0 (LW-9200)	0000 LW-0 + Index 0 (LW-9200)	0003 0006 LW-10 LW-20
Index 0 is "0" = data in address [LW-9200] = "0"	Read [LW-0 + Index 0] = Read [LW-0] content	[LW-10] is set to "3" [LW-20] is set to "6"
Index Function of Word 0010 Index O (LW-9200)	0003 LW-0 + Index 0 (LW-9200)	0003 0006 LW-10 LW-20
Index 0 [LW-9200] is set to "10"	Read [LW-0 + Index 0] = Read [LW-10] = "3"	
Example 2 In the same	ction of Bit ie way, Index Register can b ling 1 Word in value of inde>	
Index Function of Bit 0001 Index 6 (LW-9206)	LB-0 + Index 6 (LW-9206)	LB-16
Index 6 [LW-9206] is set to "1"	The switch [LB-0 + Index of the switch [LB-0 + Index of the switch]	6] reads LB-16 address = ON
Index Function of Bit 0002 Index 6 (LW-9206)	LB-0 + Index 6 (LW-9206)	LB-16
Index 6 is set to "2"	The switch [LB-0 + Index of	6] reads LB-32 address = OFF



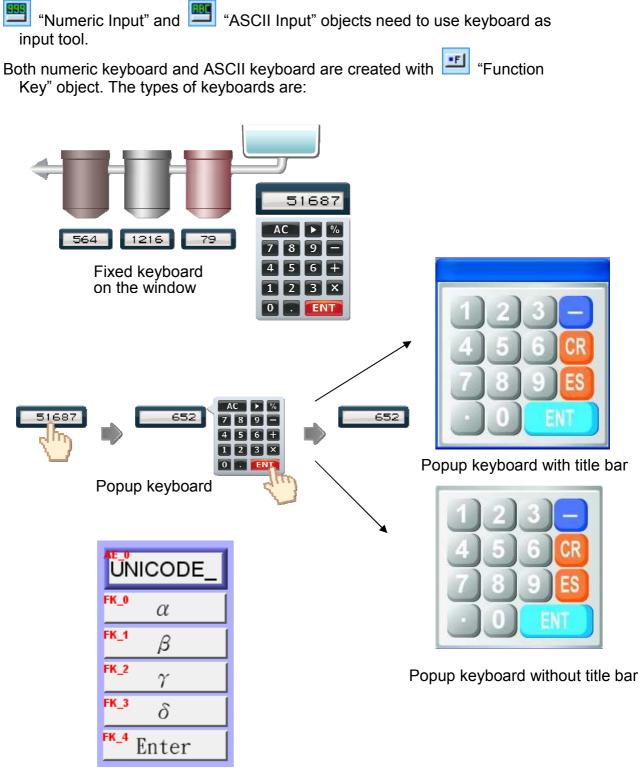
Conclusion: Index Register is used to change addresses. Through changing the data in Index Register, we can make an object to read and write different addresses without changing its own address of the device. Therefore we can transmit or exchange data among different addresses.



Please confirm your internet connection before downloading the demo project.



Chapter 12 Keyboard Design and Usage



UNICODE keyboard

12.1 Steps to Design a Pop-up Keyboard

Step 1 Create and open a window for a keyboard to be added. For example, set to "WINDOW 200".

Step 2 Adjust the height and width of "WINDOW 200" and create a variety of "Function Key" objects in

[ASCII/UNICODE mode].

For example:

() [Enter]

ASCII/UNICODE mode

🔘 [Enter]

📀 [ASCII] / [UNICODE] 1

○ [ASCII] / [UNICODE]

[FK_11] is used as the **[Esc]** key.

[Backspace]

[FK_14] is used as the [Enter] key.

🔘 [Backspace]

O [Backspace]

C [ASCII] / [UNICODE] The rest are mostly used to input number or text. For example, [FK_0] is used for inputting number

🔘 [Clear]

🔘 [Clear]

🔘 [Clear]

📀 [Ex:]

○[E∞]

🔘 [Exc]

For example, [FK_0] is used for inputting numbe "1". ASCIL/UNICODE mode

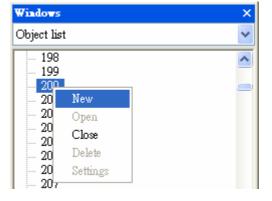
Step 3 Select a suitable picture for each "Function Key" object. [GP_0] is a picture object which is placed at the bottom layer as the background.

Step 4 Select [System Parameter Settings] / [General] / [Keyboard] / [Add] [Window 200]. Up to 32 keyboard windows cab be added.



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	









Step 5 After the keyboard window is added, when creating "Numerical Input" and "ASCII Input" objects, "200 Keyboard" can be found in

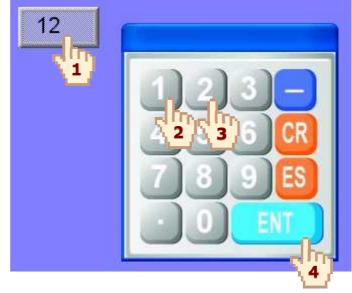
[Data Entry] / [Keyboard] / [Window no.].

The [Popup position] is

used to decide the display position of the keyboard on screen. The system divides the screen into 9 areas.

Keyboard V Use a popup keypa Hide title bar	əd			
Window no. : 200. Keyboard				
Popup position : {relative to HMI screen}	000	000	000	
Hint : If the keyboard is an USB key or on the same window, please don't	vboard check	, on i ("Use	ndirect a pop	/direct window, up keypad".

Step 6 Select "200.Keyboard". When users press "Numerical Input" or "ASCII Input" objects, WINDOW 200 will pop up on HMI screen. Users can press keys on keyboard to input data.





12.2 Steps to Design a Keyboard with Direct Window

Step 1 Create a Direct Window" and set a read address to activate it. In [General]/[Attribute] select [No title bar] and correct [Window No.].

ew Direct Window	v Object	Direct Window Ob	ject's Properties	
Jeneral Description : Trigger: Read address PLC name : Address :	ON 💌	Windo	۲ : 40 Width : 160 e [Profile] of "Direct w" to the same size eated keyboard win	e as
	No title bar			

Step 3 Create a "Numeric Input" object, and don't tick [Use a popup keypad].

New Numeric Input Object	New Set Bit Object
General Data Entry Numeric Format Security Shape Font Mode : Touch	General Security Shape Label Description : Write address PLC name : Local HMI Address : LB Write after button is released
Keyboard	Set style : Set ON Step 4 Create a "Set Bit" object, [LB-0], [Set ON] and overlay it on the "Numeric Input" object.

Step 5 Add "Set Bit" objects on **[Enter]** and **[ESC]** function keys respectively. **[LB-0]**, **[Set OFF]**, in this way when users press either [Enter] or [ESC] will close the keyboard.



12.3 Steps to Design a Fixed Keyboard on Screen

Users can also place a fixed keyboard on 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, in [Data Entry] / [Keyboard] don't tick [Use a popup keypad].

-Keyboard -

🗌 Üse a popup keypad

Step 2

Use "Function Keys" to design the keyboard and place them on screen.

Step 3

Press "Numeric Input" object, users can input value with function keys directly.

1	h-1	
1	2	3
	5	6
7	8	9
Basks	space	Enter



12.4 Steps to Design a UNICODE Keyboard

SCII Input Obje	ct's Properties		
General Data Entr	ry Security Shape Font Profile	UNICODE_	ASCII/UNICODE mode
Description		<mark>FK_0</mark> α	
🔲 Mask	. 🔽 Use UNICODE 🛛 🗌 Reverse high/low byte	^{FK_1} β	ASCII/UNICODE mode (Enter) (Backspace]
Read address —		FK_2	[ASCII] / [UNICODE] β
	2: Local HMI Setting	γ	
Address ddress	: LW 0	<mark>FК_3</mark> ∂	ASCII/UNICODE mode C [Enter] C [Backspace]
PLC name :	Local HMI	^{FK_4} Enter	⊙ [ASCII] / [UNICODE] Y
Device type :			ASCII/UNICODE mode
Address :			C[Enter] C[Backspace]
	DDDDD [range : 0 ~ 10799]		⊙ [ASCII] / [UNICODE] 8
	Index register		
	No. of word : 8		
F	Place an 🛅 "ASCII Input" object	Create 🗾 "F	unction Keys" [α] [β]
O	on screen, tick [Use UNICODE] .		n, and an [Enter] INICODE keyboard
	Users can "Group" the sel Group Library" for future use	-	I and "Save to



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

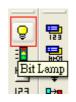
Overview

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.



L	WE!NTEK

Feneral Secu	rity Shape	Label					
Descri	ption :						
Read addres	\$						
	ame : Local	HMI	-520		V [Setting	
Add	ress : LB		✓ 0				
	Inv	vert signal					
Blinking							
Blinking			Bł	nking time	: 0.5 second	.(s)	~
	10de : Alter	nating image		nking time	: 0.5 second	(s)	~
	Mode : Alter	nating image		nking time	: 0.5 second	(8)	~
	Mode : Alter	nating image		nking time	: 0.5 second	(s)	~
	víode : Alter	nating image		nking time	: 0.5 second	.(s)	>
	Mode : Alter	nating image		nking time	: 0.5 second	(6)	>
	4ode : Alter	nating image		nking time	: 0.5 second	.(s)	× ×
	Mode : Alter	nating image		nking time	: 0.5 second	(\$)	× ×
	Mode : Alter	nating image		nking time	: 0.5 second	(6)	× ×

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.



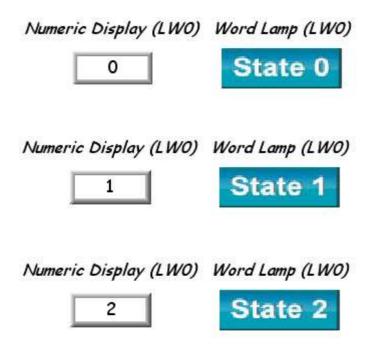
Read	Click [Setting] to select the [PLC name], [Address], [Device
address	type], [System tag], [Index register] of the bit device that controls
	the bit lamp object. Users can also set address in [General] tab while
	adding a new object.
	Address
	PLC name : Local HMI
	Address : 0 System tag
	Address format : DDDDD [range : 0 ~ 11999]
	Index register
	OK Cancel
	[Invert signal]
	Display shape with inverse state; for example, the present state is
	"OFF", but it displays the shape of "ON" state.
Blinking	Set blinking attribute of bit lamp.
	[Blinking mode]
	a. None
	No blinking.
	b. Alternating image on state 0
	Alternatively display the shape of state 0 and state 1 when the bit
	value is OFF (state 0).
	c. Alternating image on state 1
	Alternatively display the shape of state 0 and state 1 when the bit
	value is ON (state 1).
	d. Blinking on state 0
	Display the shape of state 0 in blinking when the bit value is OFF
	(state 0).
	e. Blinking on state 1
	Display the shape of state 1 in blinking when the bit value is ON (state 1).



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.



L	WE!NTEK

eneral Security S	hape Labe	el			
Description :	<u> </u>				
Mode :	Value	*	Offset :	0	
Read address		193	2016214304		
PLC name :	Local HMI			*	Setting
Address :	LW	🗸 🗸			16-bit Unsigned
Attribute			No. of states :	2	×

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

NTEK	n Olfred			Objects	1
Descripti Mo Read address PLC nan	y Shape Label	200		Setting bit Unsigned	
Attribute —		No. of sta	tes: 2		
	etting, if the valu e the picture belo		is "5", the s	shape of sta	te "2" is
	5	S	tate 2		
	LW200	LW200	D, Offse	<i>t = 3</i>	
b. LSB					
	ead address valu ate. Please refer			bits other th	nan valı

	Read addre value	ess Binary value	Displayed state
	0	0000	All bits are 0, display the shape of state 0
	1	0001	The lowest bit other than 0 is bit 0, display the shape of state 1
	2	0010	The lowest bit other than 0 is bit 1, display the shape of state 2
	3	0011	The 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
	-	ect have nothing to c	
ead address	The states of the objective displays difference click [Setting] to Se	ect have nothing to c ent shape of states a elect the [PLC name	do with the word value. The system according to time frequency. according to time [Address] ,
ead address	The states of the object. Users can also	ect have nothing to c ent shape of states a elect the [PLC name register] of the word	do with the word value. The system according to time frequency. e], [Device type], [Address], d device that controls the word lam
ead address	The states of the objective displays different displays di	ect have nothing to d ent shape of states a elect the [PLC name register] of the word set address in [Ger set address in [Ger signed v 10500]	do with the word value. The system according to time frequency. b], [Device type], [Address], d device that controls the word lam heral] tab while adding a new object Image: Second S



The number states one object possesses. State 0 is also counted as one state.. Suppose the number of the states is 8, the valid states will be 0, $1\sim7$. 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.

lew Word Lamp Object 🛛 🔀	New Word Lamp Object
General Security Shepe Label	General Security Shape Lobel
Uzs label Uzs label khrasy Uzs label khrasy Uzs label habery.	□ One label ⊘ One label library Label bag Label_0 ♥ □ One latmap fout □ Label Library
Langroups : 1 💌 State: 0 🛩 🗲 🕨 0 1	Langrage : 2 💌 State : 0 🛩 4 🕨 0 1
Attribute Foat: [Arm] Color: See 16	Athilute Font Anni Color :
Align: Left Blink: None Dali: Underline Deplicate these statilenes to Every state Every langrage All	Align : Lett v Plink : From ve Noix Underline Duplicate these ethilosise to Every state Every longuage All
Movement Disaction : No movement	Movement Direction : No service of Con-
Cooleat	Contrast :
a	
Tracking Dipplante this label to sweep inde	Tracking Doplarts that label to every state
OK Cancel Help	OK Cancel Help



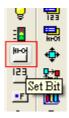
13.3 Set Bit

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.



Li	WE!NTEK

eneral Security Shape	Label		
Description :			
Write address			
PLC name : Local H	IMI	Ý	Setting
Address : LB	¥	0	
		Write after button is rele	eased
Attribute			
Set style : Toggle			~
Macro			
💽 Execute macro	Macro :	ID1 (ID : 1)	~
	Trigger mode :	OFF->ON	~
		OFF->ON ON->OFF	<u>.</u>
		OFF<->ON	10

Setting	Description
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. Users can also set address in [General] tab while
	adding a new object.

1	
	WE NTEK

	Address	
	PLC name : Local H Device type : LB Address : 0 Address format : DDDDD	System tag
	touched and released activated once the bu	cted, the operation is activated after button is d, otherwise, if not selected, operation will be utton is touched. If the "Momentary" switch is ation mode, the [Write after button is released]
Attribute		llowing description for different types of operation
[Set Style]	mode.	
	Set style	Description
	Set ON	When the operation is activated, the bit
		device will be set to ON.
	Set OFF	device will be set to ON. When the operation is activated, the bit
	Set OFF	device will be set to ON.When the operation is activated, the bit device will be set to OFF.
		device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit
	Set OFF	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from
	Set OFF Toggle	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.
	Set OFF	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit
	Set OFF Toggle	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit
	Set OFF Toggle	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit
	Set OFF Toggle Momentary	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit device will be set to ON, and the bit device will be set to OFF once the finger removes from area.
	Set OFF Toggle	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit
	Set OFF Toggle Momentary	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit device will be set to ON, and the bit device will be set to OFF once the finger removes from area.The state of the bit device will be switched
	Set OFF Toggle Momentary	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit device will be set to ON, and the bit device will be set to OFF once the finger removes from area.The state of the bit device will be switched
	Set OFF Toggle Momentary	device will be set to ON.When the operation is activated, the bit device will be set to OFF.When the operation is activated, the bit device will be set from ON to OFF or from OFF to ON.When touch and hold the area, the bit device will be set to ON, and the bit device will be set to OFF once the finger removes from area.The state of the bit device will be switched



Г		
	window opens	object is opened, the bit device will be
		automatically set to ON.
	Set OFF when	When the window containing the Set Bit
	window opens	object is opened, the bit device will be
		automatically set to OFF.
	Set ON when	When the window containing the Set Bit
	window closes	object is closed, the bit device will be
		automatically set to ON.
	Set OFF when	When the window containing the Set Bit
	window closes	object is closed, the bit device will be
		automatically set to OFF.
	Set ON when	When the backlight is turned on, the bit
	backlight on	device is automatically set ON.
	Set OFF when	When the backlight is turned on, the bit
	backlight on	device is automatically set OFF.
	Set ON when	When the backlight is turned off, the bit
	backlight off	device is automatically set ON.
	Set OFF when	When the backlight is turned off, the bit
	backlight off	device is automatically set OFF.
Macro	Users can use [set bit] object to activate macro commands. Macro
	commands have to be b	ouilt before configure this function. Please refer
	to related chapter on ho	w to edit Macros.
Set style	Attribute	
	Set style : Toggle	✓
	- Macro	
	🗹 Execute macro	Macro: ID1 (ID:1)
		Trigger mode : OFF->ON
	·	OFF->ON ON->OFF
		OFF<->ON
		ected as [Toggle], there are three different
		command, 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.



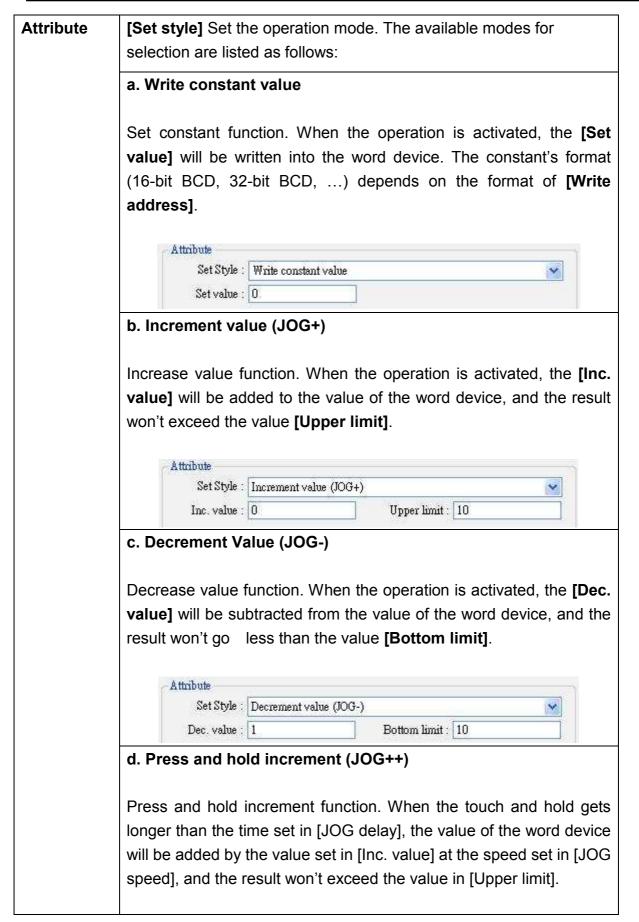
WEINTEK

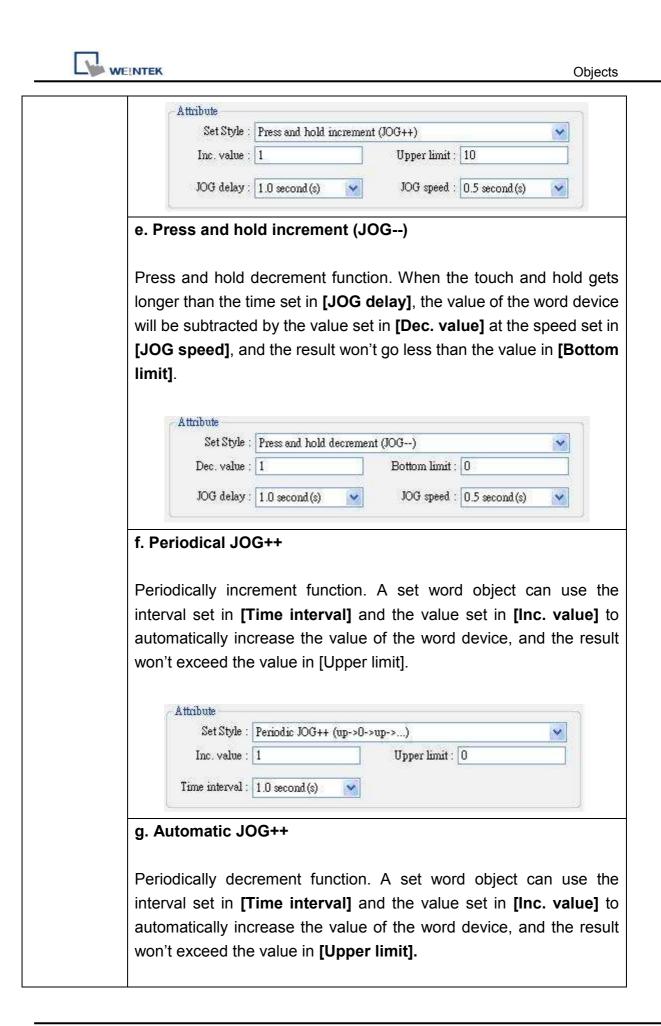
eneral 3	Security S	hape Label			
D	escription :	[
Write ad	ldress				
P	LC name :	Local HMI		*	Setting
	Address :	LW	•		16-bit Unsigned
		Write after buttor	1 is released		
Notifica			<u>A</u> ayaw	0.7	
		💌 Enable	💽 Set ON	⊖ Set (JFF
		🗹 Before writing	🔲 After writing	(
P	LC name :	Local HMI		~	Setting
	Address :	LB	0		
Attribut	e				
	Set Style :	Write constant value	1		*
	Set value :	12			

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]				
	OK Cancel				







Set Style : Automs	tic JOG++ (up to hi;	gh limit)	*
Inc. value : 0		Upper limit : 10	
Time interval : 0.5 sec	ond (s) 🔽		
h. Automatic JOG			
Pariodically docromon	t function A	sot word obj	oct con use
Periodically decremen nterval set in [Time i i		-	
automatically decrease	-		-
won't go less than the			
		ioni ming.	
Attribute	6 IOC (1		
Set Style : Automa	Steam And a start	Size i usene	
Dec. value : 1		Bottom limit : 0	
Time interval : 1.0 seco	ud (s) 💉		
B			
. Periodical bounce			
Deriodically bouncing f	unation A S	at word abject w	vill add than
Periodically bouncing f		-	
set in [Inc. value] to the standard set in [Time in			-
nterval set in [Time in			
n [Upper limit] , and the			
he value of the word		-	
esult value reaches the		-	-
he value in the word o	evice will cha	ange periodically	y from $0~10$
rom 10~0.			
Attribute			
Set Style : Periodic	step up (low to hig	h)	~
Low limit : 0		High limit : 10	
Inc. value : 1		30 A.	
Time interval : 0.5 seco	nd (s) 🔽		
[



device will char	nge periodically in order of 0, 1, 2,, 9, 10,
2,	
Attribute	
Set Style	Periodic step up (low to high)
Low limit	: 0 High limit : 10
Inc. value	
Time interval	: 0.5 second (s)
k. Periodical st	ep down
Stepping down f	function. A Set word object will subtract the valu
	,
in [Dec. value] f	rom the value of the word device with the regu
interval set in IT	ime interval] until the result value reaches the
-	-
of the [Low limi	t], and the value of the word device will return t
value of the [Hig	the limit] and then repeat the action to keep the
in an active state	a In the example shown below the value of the
	•
	•
device will chan	
device will chan	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan Attribute Set Style	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan Attribute Set Style Low limit	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan Attribute Set Style Low limit Dec. value	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan Attribute Set Style Low limit Dec. value	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan Attribute Set Style Low limit Dec. value	ge periodically in order of 10, 9, 8,, 1, 0, 10,
device will chan Attribute Set Style Low limit Dec. value Time interval	ge periodically in order of 10, 9, 8,, 1, 0, 10 Periodic step down (high to low) U High limit : 10 High limit : 10
device will chan Attribute Set Style Low limit Dec. value	ge periodically in order of 10, 9, 8,, 1, 0, 10 Periodic step down (high to low) U High limit : 10 High limit : 10
device will chan Attribute Set Style Low limit Dec. value Time interval	ge periodically in order of 10, 9, 8,, 1, 0, 10, Periodic step down (high to low) U High limit : 10 High limit : 10 D 5 second(s)
device will chan Attribute Set Style Low limit Dec. value Time interval	ge periodically in order of 10, 9, 8,, 1, 0, 10 Periodic step down (high to low)
device will chan Attribute Set Style Low limit Dec. value Time interval	ge periodically in order of 10, 9, 8,, 1, 0, 10
device will chan Attribute Set Style Low limit Dec. value Time interval I. Set when windo	ge periodically in order of 10, 9, 8,, 1, 0, 10 Periodic step down (high to low) Periodic step down (high to low) High limit : 10 High limit : 10 How opens we containing the object is opened, the value o
device will chan Attribute Set Style Low limit Dec. value Time interval I. Set when windo	ge periodically in order of 10, 9, 8,, 1, 0, 10 Periodic step down (high to low)
device will chan Attribute Set Style Low limit Dec. value Time interval I. Set when windo	ge periodically in order of 10, 9, 8,, 1, 0, 10, Periodic step down (high to low) Description: High limit : 10 High limit : 10 How opens bow containing the object is opened, the value o
device will chan Attribute Set Style Low limit Dec. value Time interval I. Set when windo	ge periodically in order of 10, 9, 8,, 1, 0, 10, Periodic step down (high to low) Description: High limit : 10 High limit : 10 How opens bow containing the object is opened, the value o
device will chan Attribute Set Style Low limit Dec. value Time interval I. Set when windo value] will be au Attribute	High limit : 10 High l
device will chan Attribute Set Style Low limit Dec. value Time interval I. Set when windo value] will be au Attribute	ge periodically in order of 10, 9, 8,, 1, 0, 10, Periodic step down (high to low) High limit : 10 dow opens dow opens we containing the object is opened, the value or utomatically written into the word device. Set when window opens



	Attribute Set Style : Set value :	Set when window closes	
n. S	et when bac	klight on	
	Set Style :	Set when backlight on	
	Set value :	5	
	DEL VOLUE .		
o. S	et when bac	klight off	



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.



WE!NTEK

neral Security Shape Label		
Description :		
📃 Activate after butt	on is released	
💿 Change full-screen window	O Change con	nmon window
🔘 Display popup window		
Window no. : 50. Keypad 1 - Integ	er	Y
OReturn to previous window	O Close windo	9W
ASCII/UNICODE mode		
○ [Enter] ○ [Backspace]	🔘 [Clear]	○ [E∞]
O [ASCII] / [UNICODE]		
O Execute macro		
🔿 Window title bar		
Hard copy screen to USB disk or printer		
O Screen hard copy		
Notification		
Enable	🔘 Set ON	💽 Set OFF
PLC name : Local HMI		Setting
Address : LB	0	

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	NOTE: 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] [Backspace] [Clear] [Ex]					
	[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]	Macro commands are executed with this selection. Macro commands have to be built before users choose this function. Please refer to related chapter on how to edit Macros.					
	Execute macro Macro: macro 1 (ID : 1)					
[Window title bar]	A [function Key] which is defined as Window Title Bar can move the popup 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.					
	Alarn Status					
	Select the window title bar firstly. Touching the screen for the new position the popup window will be moved.					
[Screen hard copy]	Hardcopy current display screen to the printer connected with MT8000. Before 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) Mode : color black and white grayscale color 					
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.

ystem	Paramet	er Settin	igs				Ľ
Device	Model	General	Security	Font	Extended Memory	Printer Server	
Font	s for no-as	cii strings -					
EV.	MinchoL				1	411	
AR	MingtiM	GB				Add	
	MingtiM]	KSC				Delete	
AR	8						
AR Ari	al nHei						

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.

🗆 🌣 💒 🗛	
J 🚧 ñuñ Es	General Security Shape Label Profile
A* A* ≣ [Description :
日 串 引 可	Activate after button is released
FK-A- FK	Change full-screen window
■ 简 B ^{FK} -BAC	O Display popup window
PACT	
Th BACI	
EV. 0	
	Return to previous window
	Return to previous window Close window ASCII/UNICODE mode
2	

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".

General	Security	Shape	Label	Profile			
V	Use label						
同	Use label l	ibrary					
	State :	0		• 0 1			
Attri	bute		10.000				
	(Font:	AR Min	gtiM GB				*
	Color :	1		.	Size :	16	*
	0 1i	Center		~	Blink :	None	*
	Align :	1000000000					
	Angn .	Italic		🔲 Underline			

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

Device Model General	Security Font Extended Memory Printer Server	
Fast selection button		- i
Attribute	E Disable	-
Screen saver		
Back light saver	: None minute (s)	
Screen saver	: None 🛛 minute(s)	
Option		
Startup window no.	: 10. WINDOW_010	*
Extra. no. of events	: 0 Common window : Above base window	~
Keyboard caret color	: Dbject layout : Nature	~
	RW_A enabled	
Keyboard		
	50. Keypad 1 - Integer 51. Keypad 2 - Integer	
	52. Keypad 3 - Integer 53. Keypad 4 - Integer 54. Keypad 5 - Integer 55. Keypad 6 - Integer 56. Keypad 7 - HEX 57. Keypad 8 - Floating 60. ASCII Middle 61. ASCII Small	



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.



WEINTEK

ocal HMI B] Invert signal ocal HMI B	~				Settir	
B] Invert signal ocal HMI B						
B] Invert signal ocal HMI B						
] Invert signal ocal HMI B				~	Settir	ıg
ocal HMI B				~	Settir	ıg
В				~	Settir	ng
В	~			~	Settir	ng
no Managemente des ante	~	9.0		Conceller 1		
400000 30 30	66	0				
] Write when b	utton is re	leased				
loggle	~					
10						
			10	10	10	10

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.

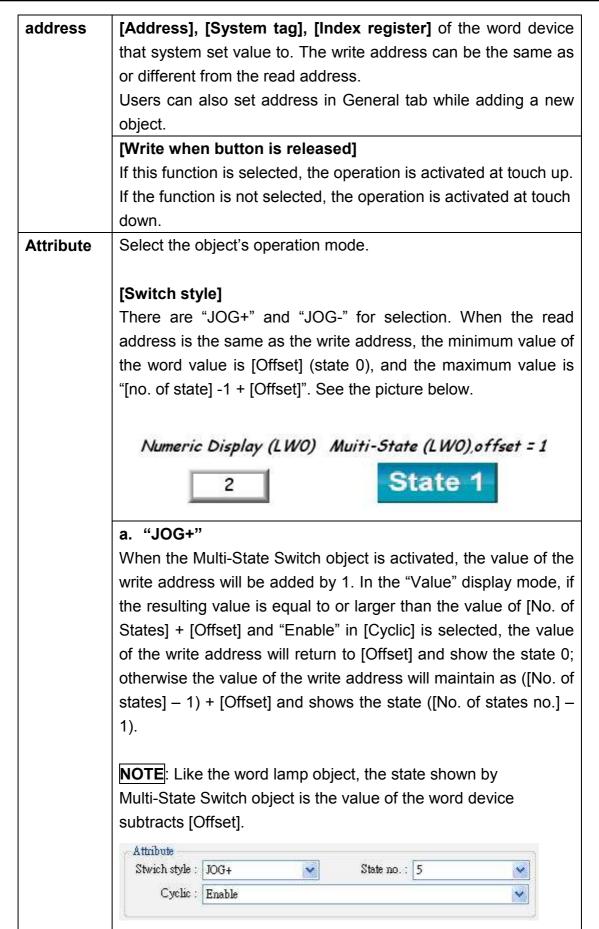


L	WEINTEK

eneral Security S	Shape Label	
Description :		
Mode :	Value 🗸 Offset : 0	
Read address		
PLC name :	Local HMI	Setting
Address :	LW 🔽 0	16-bit Unsigned
Attribute	Write when button is released	
Attribute Switch style :	JOG+ No. of states : 1	
Cyclical :		
Cyclical .	User-defined mapping	×

Setting	Description		
[Mode]/	There are "Value" and "LSB" display mode. Refer to the "Word		
[Offset]	Lamp 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],		







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			
Description :				
Write address	(a).			
PLC name :	Local HMI		~	Setting
Address :	LW	v 0		16-bit Unsigned
PLC name : Address :	Eocorr II II	After writi		Setting
Watch address	D P			
Walch address	🗹 Enable			
WAICH AUUTESS				()
PLC name :	Local HMI		1	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				

	e Security Shape	<u>]</u>			
Attribute	and a second		the strate of		
Direc	tion : Right	~	Resolution :	1	
Low/High l	imit : 🚫 Constant	💿 Addre	s		
PLC na	me : Local HMI			*	Setting
Addr	ess : LW	~]0)		16-bit Unsigned
	Coarse incre	ment	Increment :	10	3
Slider button					
Fr	ume :		Transparent		_
Backgro	und : .	· · ·	 Transparent		
	Slot :] Transparent		
	P2.50 2.481		Tronoberour		

Setting	Description				
Attribute	[Direction]				
	The bar on the slide of	direction, i.e. left, rig	ght, up and down.		
	right left	JP down			
	[Resolution]				
	To specify the scale value, when	value of the slider, i	f N is the specified minimum		
	N=10, the numerical	display shows only	multiples of 10.		
	N=5, the numerical di		-		
	N=1, the numerical d	isplay shows only r	nulliples of 1.		
[Low limit &	a. Constant				
High limit]	The low limit and high limit of the word device is set as constant value. i.e. [Input low] and [Input high].				
	address. Click [Setting…] to S	Select the [PLC na x register] of des	e is controlled by a designated ame], [Device type], [Address], ignated address or users can		
	Low/High limit : 🔘)Constant 💿 Addres	8		
	PLC name : L		Setting,		
	Address :	w 🔽 0	16-bit Unsigned		
		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		alue will increase/decrease one		



	[increment] value for every touch activation. If not, the word value will be set the value in accord with the touch activated point.		
Slider button type	There are four slider button types for selection. You also can adjust the width of moving piece.		
Color	This is used to select slide object frame, background and slot's color.		

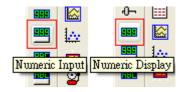
13.9 Numeric Input and Numeric Display

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".

eneral	Data Entry	Numeric Format	Security	Shape	Font	
1	Description :					
		1				
	Read/W	Vrite use different ad	dresses			
Read a	address					
	PLC name :	Local HMI			~	Setting
	Address :	LW	~	0		
Myita	address					
WILLE	PLC name :	Local HMI			~	Setting
	Address :		~	10		
		1	transfer t		1	
Notific	cation	🔽 Enable	C	SetON	💿 Set	370
		and a second second		and the second	20	
		Before writing	L	After writi	ng	
	PLC name :	Contraction and the			~	Setting
	Address :	LB	*	0		
Notific	ation on inv	alid input				
		Enable	(Set ON	🔿 Set	OFF
	DI C	December 201				
	PLC name :	Carter Constant State Con-			~	Setting
	Address :	LB	~	20	di second	

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



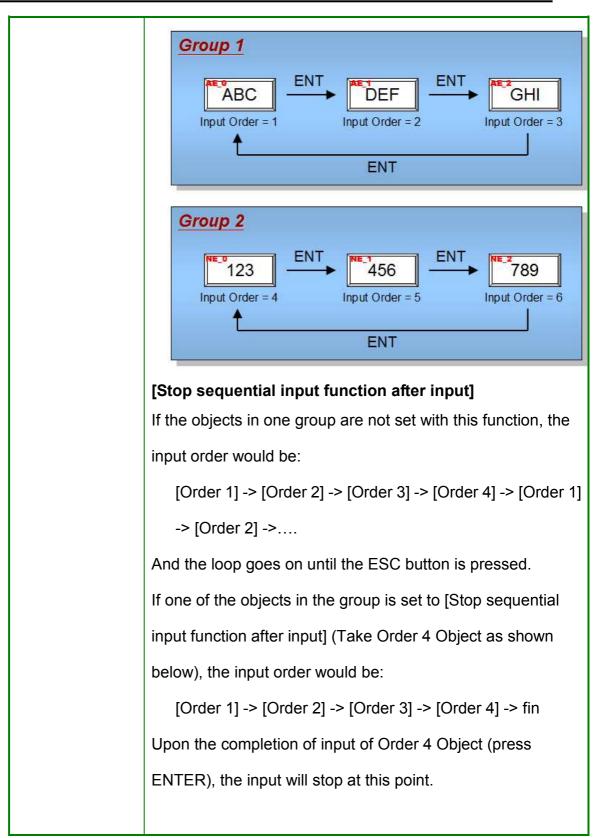
address	system writes to.
Notification	When this function is selected, the state of the designated bit device will be set to [ON] or [OFF] after/before the value of the register is changed successfully. Click [Setting] to Select the [PLC name] , [Device type] , [Address] , [System tag] , [Index register] 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	

WE!NTEK

meral Data Entry Numeric Format S	ecurity 🛛 Shape 🚽	Font
Mode : Touch	~	
Input order		
🗹 Enable		
Input order : 1	🔽 Group	2
Carl Stars and an in the stars of the		
Stop sequential input function afte	ւուիտ	
Keyboard		
🔽 Use a popup keyp	ad	
🔲 Hide title bar		
Window no. : 50. Keypad 1 - Integ	er	~
Popup position : {relative to HMI screen}	0000	
Hint : If the keyboard is an USB key or on the same window, please don	vboard, on indirect/	direct window, un keymad"

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 object after users complete inputting data, i.e. press ENT.
	 Enable Select [Enable] and set Input Order to enable this feature. Furthermore, users can also select [Group] to set Input Order Group. a. The range of Input Order: 1 ~ 511. b. The range of Input Order Group: 1 ~ 15. c. The Input Order Group of an input object with [Group] unselected is 0.
	 Criterion of searching the next input object a. The system only searches it among the input objects with the same Input Order Group. b. The system picks the input object with smaller Input Order to enter input state before another one with bigger Input Order. 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.
	• 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".



	· · · · · · · · · · · · · · · · · · ·
	Input order Enable
	Input order : 4 🔄 🕼 Group 1 🚖
	Stop sequential input function after input
	Order 1 Order 2 Order 3 Order 4
	Input order
	I Enable
	Input order : 1 Group 1 🚖
	sub sei eenen uha renenen aisi uha
	When selecting [Bit control] as Mode
	a. Users have to specify an Input Order for the object.
	b. 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.
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.
	 Unselect [Use a popup keypad]
	The system does not automatically display keyboard
	window. Users have to complete the input process via
	following methods:
	a. Design a custom keypad and place it in the same
	window with the input object.
	b. Use an external keyboard.
	Hide title bar
	Keypads 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.

Display Data format : 16-bit Unsigned 🛛 😽	
Data format : 16-bit Unsigned 🛛 😽	
	Mask 📃
Number of digits	· · · · · · · · · · · · · · · · · · ·
Left of decimal Pt. : 4	Right of decimal Pt. : 0
Scaling option	
Do conversion	
Engineering low : 0	Engineering high : 9999
Limits	
O Direct 💿 Dynamic limits	
PLC name : Local HMI	Setting
Address : LW	0
Use alarm color	
Low limit :	🗹 Blink
High limit :	Blink

Setting

Description



Display	[Data format]
	To select the data format of the word device designated by the "Read
	address". The selection 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 displayed, "*" will be used to replace all digitals and the color
	warning function will be cancelled.
Number of	[Left of decimal Pt.]
digits	The number of digits before the decimal point. [Right of decimal Pt.]
	The number of digits 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 designated by the "Read address." When the function is
	selected, it is required to set [Engineering low], [Engineering high], and [Input
	low] and [Input high] in the "Limitation". Supposed that "A" represents the raw
	data and "B" represents the result data, the converting formula is as follows:
	B = [Engineering low] + (A - [Input low]) × ratio
	where, the ratio = ([Engineering high] - [Engineering low]) / ([Input high] - [Input low])
	See the example in the picture below, the raw data is 15, after being



	converted by the above and the result "40" will b Scaling option Engineering to	e displayed on the	15 – 0) × (50 – 10) / (20 · numeric input object. gineering high : 50	- 0) = 40,	
	Limits —				
		O Decembra lineite			
	Direct	O Dynamic limits	aput bish : 20		
	Input low : 0		nput high : 20		
Limits	To set the source of the effect.	range for the input	data and to set the warning	ng color	
	•	•	can be set in [Input low] a f the defined range, the in		
	[Dynamic limits]				
	Limits				
	O Direct	💿 Dynamic limits			
	PLC name : Lo	ocal HMI	Setting		
	Address : []	N 🗸 0			
	1				
	Set the low limit and high limit of the input 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 example, the low limit and high limit are derived from [LW100] and the following explains the usage of the low limit and high limit from designated address. Click [Setting] to Select 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 PLC's register is smaller 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.

eneral	Numeric Format	Security	Shape	Font	Profile	
Attrib	ute					
	Font : Comic Sa	ns MS				*
	angered as			Size	: 12	*
	Color : 📕					
	Color : 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.

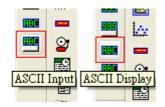
13.10 ASCII Input and ASCII Display

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".

	Data Entry	Security	Shape	Font	
	Description :				
	🗌 Ma	sk	📃 Use UN.		Reverse high/low byte
Read	address	C			
PLC name :	1		Trail Land	Setting	
	Address :	LW		• 0	
Notifi	cation		_		
		Enable		O Set ON	Set OFF
		🛃 Before v	writing	🔲 After writi	ng
	PLC name :	Local HMI	-		Setting
	Address :	LB		✓ 0	

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 Obje	et		
General Security Shap	e Font		
- Attribute Font : Arial			
Color : Align : Left		Size : 16	×
<u></u>			
Content :			~
5			~
lie -	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.
	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
Right alignment ab
[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

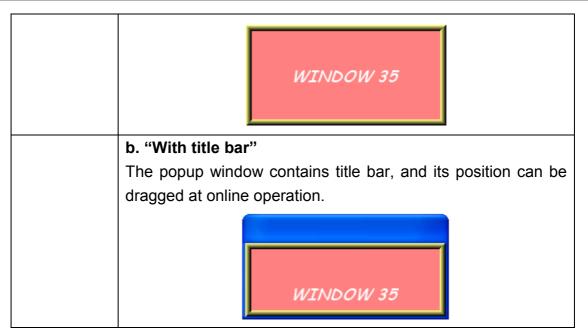
Click the "Indirect Window" icon on the toolbar and the "New Indirect Window Object" dialogue box will appear, fill in each items, click OK button, a new "Indirect Window Object" will be created. See the pictures below.



eneral					
Description :					
÷ 1 11					
Read address PLC name :	Local HMI			100	Cathing
	a The service of the			~	Setting 16-bit Unsigned
Address :	LW	*	0		16-bit Unsigned
Attribute Style :	No title bar	~			

Setting	Description					
Read	Click [Setting] to Select the [PLC name], [Device type],					
address [Address], [System tag], [Index register] of the word d						
	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.					





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.

	Read address					
WP_0	PLC name : Address :		Local HMI 😪			Setting
			LW 🕑 0			16-bit Unsigned
w_0	constant 35 to L	w100				
Jen	CONSTANT 30 TO L	W100				
Set Set	t constant 36 to L	W100				
SW_2 Se	t constant O to LV	V100				
		DO 111 025				
	∎ *35: WIN ∎ *36: WIN 					

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.

WINDOW	35	
	Set cons	tant 35 to LW100
	Set cons	tant 36 to LW100
	Set cons	tant 0 to LW100

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.

WINDON	7 36	
	Set cons	tant 35 to LW100
	Set cons	tant 36 to LW100
	Set cons	stant 0 to LW100

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.



L	WEINTEK

General		
Description :		
Trigger:	ON 👻	
Read address		
PLC name :	Local HMI S	ietting
Address :	LB 💙 0	
Attribute		
	No title bar	
Style :	No title bar	
Style :		•
Style :		
Style :		
Style :		

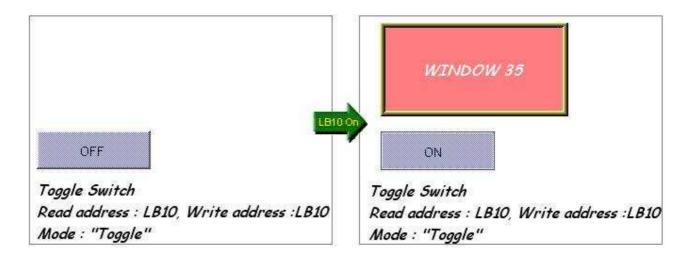
Setting	Description					
Read	Click [Setting] to Select the [PLC name], [Device type],					
address	[Address], [System tag], [Index register] of the bit device that					
	control the window popup.					
	Users can also set address in General tab while adding a new					
	object.					
Attribute	[Style]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

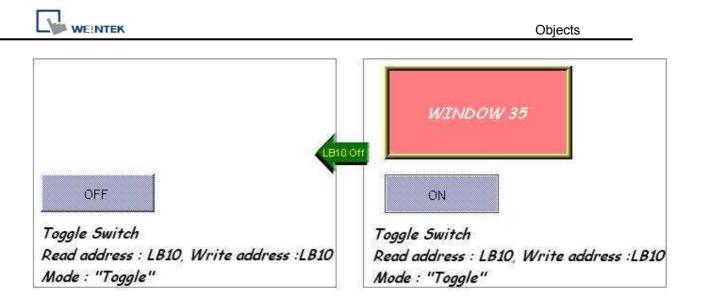
Example to use direct window

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		
	PLC name :	Local HMI	Setting
	Address :	LB 🔽 O	
TS_0			
Toggle Switch			
Read address : LB10,			
Write address : LB10			
Mode : "Toggle"	Attribute		
	Style :	No title bar 🛛 😽	
	Window No. :	35. WINDOW_035	×
	1		

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.



Description :	(
	L			
PLC name :	Local HMI			*
Read address PLC name :	Discontration		11/201	Course
		and Leve a		Setting
Address :	LW	100		16-bit Unsigned
Attribute				
Mode :	X axis only			~
No. of states :	1	*		
Min. X :	0	Max	. X : 400	
Limit address	Limit from reg		atio : 1	
PLC name :	The subsection of the subsecti			Setting
	1.00	v 0		16-bit Unsigned
Address :				1 - CALIFORD FRAME COPYER (200

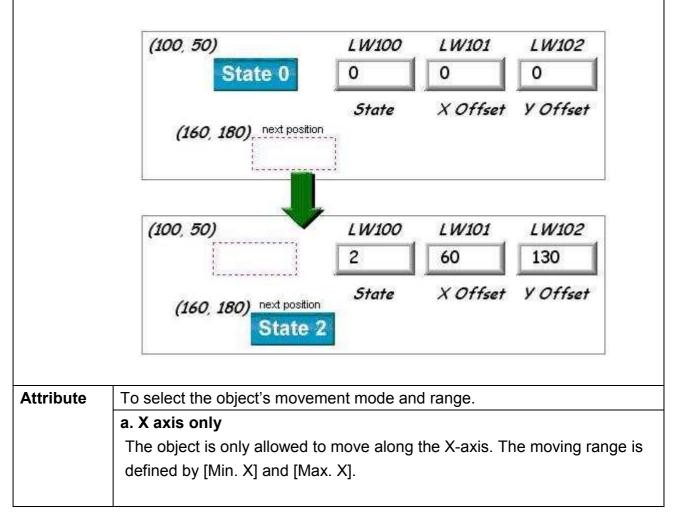
Setting	Description				
Read	Click [Setting] to Select the [PLC name], [Device type], [Address],				
address [System tag], [Index register] of the word devices that control the o					
	of object's state and moving distance.				
	Users can also set address in General tab while adding a new object.				
	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
ſ	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		Max. X : 600
	Data format	Address to	Address to
_		control object	control Moving
_		state	Distance on the
			X-axis
	16-bit format	Address	Address + 1
;	32-bit format	Address	Address + 2
b. Y a	xis only		
The o	bject is only allo	wed to move along	the Y-axis. The movin
define	ed by [Min. Y] an	nd [Max. Y].	
	Mode : Y	axis only	
	No. of states : 8	*	
	No. of states : 8 Min. Y : 0		Мах. Ү : 600
			Мах. У : 600
			Мах. У : 600 Address to
	Min. Y : 0		
	Min. Y : 0	Address to	Address to
	Min. Ÿ : 0	Address to control object state	Address to control Moving Distance on the Y-axis
	Min. Y : 0	Address to control object	Address to control Moving Distance on the Y-axis Address + 1
	Min. Ÿ : 0	Address to control object state	Address to control Moving Distance on the Y-axis
	Min. Y : 0 Data format 16-bit format	Address to control object state Address	Address to control Moving Distance on the Y-axis Address + 1
c. X &	Min. Y : 0 Data format 16-bit format 32-bit format Y axis	Address to control object state Address Address	Address to control Moving Distance on the Y-axis Address + 1
c. X & The c	Min. Y : 0 Data format 16-bit format 32-bit format Y axis bbject is allowed	Address to control object state Address Address Address	Address to control Moving Distance on the Y-axisY-axisAddress + 1Address + 2
c. X & The c range	Min. Y : 0 Data format 16-bit format 32-bit format Y axis bbject is allowed	Address to control object state Address Address Address	Address to control Moving Distance on the Y-axis Address + 1 Address + 2 e X-axis and Y-axis.
c. X & The c range	Min. Y : 0 Data format 16-bit format 32-bit format Y axis bject is allowed in XY direction	Address to control object state Address Address Address	Address to control Moving Distance on the Y-axis Address + 1 Address + 2 e X-axis and Y-axis.
c. X & The c range	Min. Y : 0 Data format 16-bit format 32-bit format Y axis bject is allowed in XY direction	Address to control object state Address Address Address	Address to control Moving Distance on the Y-axis Address + 1 Address + 2 e X-axis and Y-axis.
c. X & The c range	Min. Y : 0 Data format 16-bit format 32-bit format Y axis bbject is allowed in XY direction ctively.	Address to control object state Address Address d to move along th is defined by [Min.	Address to control Moving Distance on the Y-axis Address + 1 Address + 2 e X-axis and Y-axis.
c. X & The c range	Min. Y : 0 Data format 16-bit format 32-bit format Y axis object is allowed in XY direction ctively. Attribute Mode : X No. of states : 8	Address to control object state Address Address d to move along th is defined by [Min.	Address to control Moving Distance on the Y-axis Address + 1 Address + 2 e X-axis and Y-axis. X], [Max. X] and [Min.
c. X & The c range	Min. Y : 0 Data format 16-bit format 32-bit format Y axis bbject is allowed in XY direction ctively.	Address to control object state Address Address d to move along th is defined by [Min.	Address to control Moving Distance on the Y-axis Address + 1 Address + 2 e X-axis and Y-axis.

Objects



k;				
	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
 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 				
	Attribute Mode : No. of states : Input low : Scaling low :	D	Input high : 600 Scaling high : 1000	
	1 / 3	ect is only allowed is 300~1000, set		, U
	0	g low] to 0 and [Sc		
	e within the range			
	Data format	Address to control object state	Address to control Movi Distance on X-axis	
	16-bit format	Address	Address + 1	
	32-bit format	Address	Address + 2	
The	-	xis movement with on the Y-axis is		



		Data form	at	Addre	ss to	۸d	Idress to	
			ιαι		ol object	_	ontrol Moving	
				state	n object		stance on the	
				Slate			axis	
		16-bit form	aat	Addres	20		Idress + 1	-
						-		
		32-bit form		Addres	SS	Ad	ldress + 2	
		xis w/ reve		•				
			the sai	me as "	'X axis w/ sca	ling	, but the mov	ving direction is
		verse.						
	-	axis w/ rev		•			.	
			the sa	me as "	Y axis w/ sca	ling	, but the mov	ving direction is
		verse.						
Display		•	e in di	fferent	states can be	set	individually as	s shown in the
ratio	pictu	re below.						
	R	atio : 1	Ratio	: 1.2	Ratio : 1	4	Rati	io : 1.6
	5	State 0	St	ato 1	Stat		2 04	-4- 2
	State 0 State 1 State 2 State 3							
			_					
Limit								, [Max. X] and
address	-		-		•		•	posed that the
	-	-	-		-		-	nated register
					of [Min. X], [M	ax.	X] and [Min.	Y] [Max. Y] are
	listed	in the follo	wing ta	able.				
						_		
	Da	ta format	[Min.	-	[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

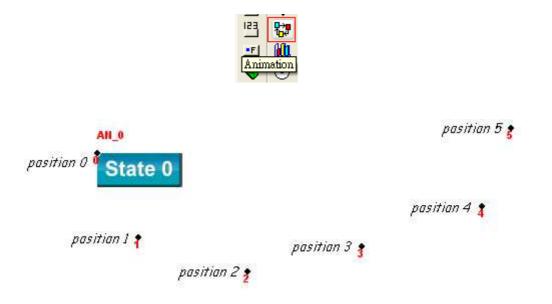
13.14 Animation

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.

Feneral	Shape I	abel H	rofile			
	Description :	[17
Attrib	ute					
	no. of states :	8		*		
	Position :	💿 Contro	olled by reg	ister 🔘 I	Based upon time	interval
Read	address					
	PLC name :	Local HM	I		Y	Setting
	Address :	LW		✓ 0		16-bit Unsigned

Setting	Description
Attribute	[Total no. of states]
	To set the number of the states for this object.
а.	When select "Controlled by register", the designated register
Controlled	controls the object's state and position.
by register	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 positior	n in different data forma	t.		
	Data Format	Address to control	Address to control		
		object's state	object's position		
	16-bit format	Address	Address + 1		
	32-bit format	Address	Address + 2		
		L			
	For example, if th	ne designated register i	s [LW100] and the data		
	format is "16-bit	Unsigned", then [LW1	00] represents object's		
	state, [LW101]	represents position.	In the picture below,		
	[LW100] = 2, [LW	/101] = 3, so the object'	s state is 2 and position		
	is 3.				
			pasitian 5		
	pasitian 0		£		
	pasitian 1		pasitian 4		
		pasitian 2	12		
		pasitian pasitian	³ State 2		
			Ottatio L		
	LW100 2	LW101	3		
b. Based	If "Based upon tir	me interval" is chosen,	the object automatically		
upon time	changes status a	nd display location. "Tir	me interval attributes" is		
interval	to set the time int	erval for states and pos	sitions.		
	Time interval attribute	- 			
	Position spee		l second(s)		
	Image state chang		Backward cycle		
	Image update tin	ne: 5 *0.1	l second(s)		
	[Position speed]]			
	Position changes	s speed, the unit is 0.1	second. Supposed that		
	[Speed] is set to	10, the object will char	nge its position every 1		



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

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

neral Shape La	bel Profile	
Position		
🔄 Pinned	X: 191 📚	¥: 122 📚
Size		
	Width : 414 😒	Height : 144 📚
Shape rectangle siz	e	
	Width : 84 😂	Height : 33 😭
Trajectory		
	Position 0	~
	X : 191 😒	¥ : 147 📚

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.

Local HMI		Setting 16-bit Unsigned
	0	
LW	D	16-bit Unsigned

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.

Objects

WE!NTEK

eral Outline Shape	
ttribute Type : Normal	Direction : Up
Zero : 0	Span : 10
	Bar width ratio (%) : 100 📑
3ar color/style	
Transparent	
Frame :	Background :
Bar :	🛛 🗸 🛛 Bar style :
arget indicator	
Enable	Color : 📕
Target value : 0 + 2	Tolerance : 0
larm indicators	
Low limit : 🚺	High limit : 0 + 1
Low color :	High color : 📕
arget/alarm/zero(span) dynamic addre	SS
🔽 Enable	
PLC name : Local HMI	Setting
Address : LW	0 16-bit Unsigner
	annig ?
Dynamic zero/sp	<u>pan</u> i

Setting	De

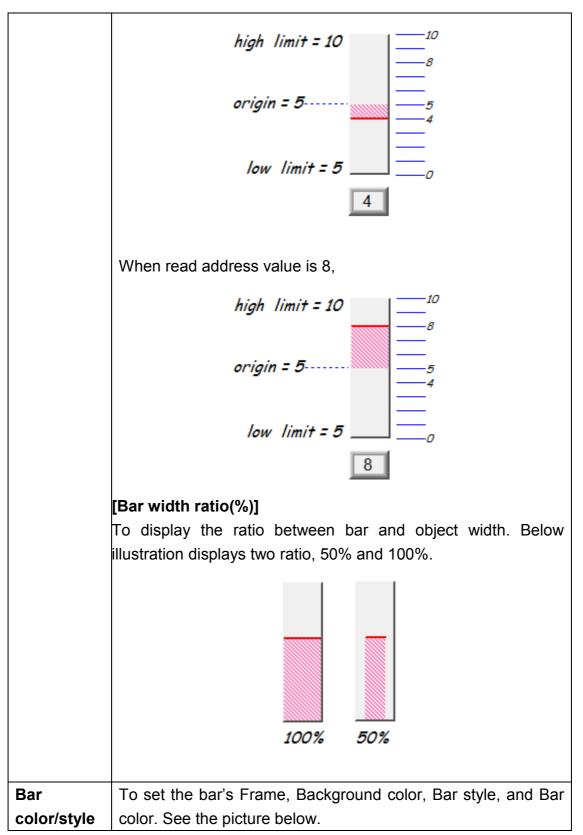
escription



Attribute	[Type]							
	There are "Normal" and "Offset" for selection. When select							
	"Offset", there must be a original value for reference. Please							
	refer the illustration below.							
	Attribute Type : Dffset V Direction : Up							
	Zero : 0 Span : 10 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 = (Register value – Zero) / [Span] – [Zero]) * 100%							
	When select "Offset", if (Register value – Zero) > 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,							









	Frame Background Bar Bar style
Target Indicator	When the register value meets the following condition, the color of filled area will change to the "Target color" Click [Setting] to Select the [PLC name] , [Device type] , [Address] , [System tag] , [Index register] of dynamic address . Users can also set address in Outline tab while adding a dynamic address. [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 Indicator	When register's value is larger than [High limit], the color of filled area 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/Alar	When select [Enable], the [Low limit] and [High limit] of "Alarm
m Dynamic	indicator" and the [Target Value] of "Target indicator" all come
Address	from designated register. See the picture below.
	New Bar Graph Object
	General Outline Shape
	Attribute
	Type : Normal 💟 Direction : Up 💟
	Zero : 20 + 3 Span : 20 + 4
	Bar width ratio (%) : 100 😂
	Bar color/style
	Transparent
	Frame : Background :
	Bar : Bar style : Bar style :
	Target indicator
	Enable Color :
	Target value : 20 + 2 Tolerance :
	Alarm indicators
	Low limit : 20 High limit : 20 + 1
	Low color : High color : High color :
	Target/alarm/zero (span) dynamic address
	Enable
	PLC name : Local HMI Setting
	Address : LW 20 16-bit Unsigned
	Dynamic zero/span
	OK Cancel Help
	The following table shows the read address of low limit, high
	limit, and target. The "Address" means the device address, for
	example, if the device address is [LW20] and data format is
	16-bit,
	The Alerm Low limit is 114/20 / The Alerry Link limit is 114/24
	The Alarm Low limit is LW 20 / The Alarm High limit is LW21



The Targe [.] LW24	•
Data	Data
orma	orma
6-bit	6-bit
ormat	ormat
32-bit	32-bit
format	format



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.

ew Mei	ter Display	Object			
General	Outline L	imits Shape	<u>п</u> .		
	Description :	[
Read	address			Local Distance	
	PLC name :			~	Setting
	Address :	LW	0		16-bit Unsigned

Read address

WEINTEK

Click [Setting...] to Select the [PLC name], [Device type], [Address],

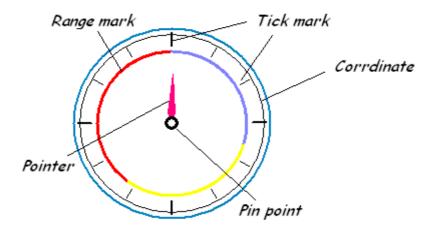
[System tag], **[Index register** of the word devices that controls the display of meter.

Users can also set address in General tab while adding a new object.

WE!NTEK

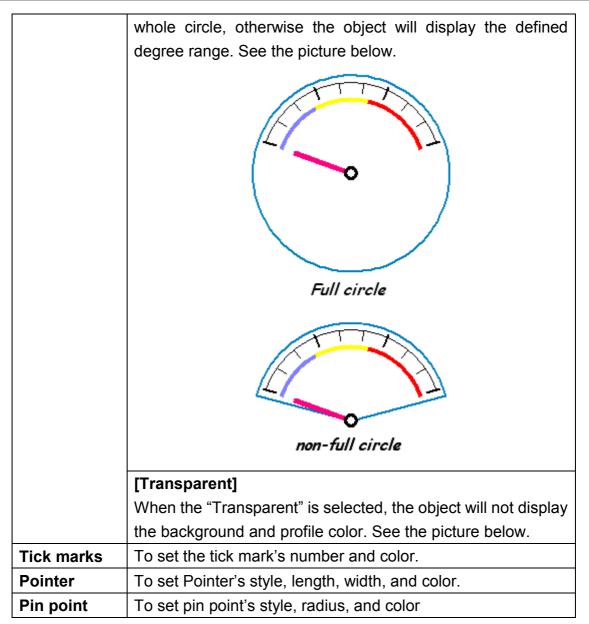
neral Outline Limits Shape		
Degree Start degree : 0	End degree : 360	
Start degree : U	Ena degree : 300	\$
Background		
Background :	Profile :	
Full circle	Transparent	
Tick marks		
🗹 Enable		
Color : _	Coordin	nate
Main scale : 4	🗢 Sub. scale : 2	٢
Length : 15	•	
Pointer		
Arm style	Frame :	- 12
	Inner :	-
Width : 4	Length : 40	\$
Pin point		
Radius : 7	\$	
Inner :	Frame :	-
💿 Circle 💦 Rectan	1816	

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.
	[Start degree] = 290, [End degree] = 70
	[Start degree] = 120, [End degree] = 240
	[Start degree] = 40, [End degree] = 140
	[Start degree] = 225, [End degree] = 315
Background	Set the object's background color and profile color.
	[Full circle]
	When the "Full circle" is selected, the object will display the





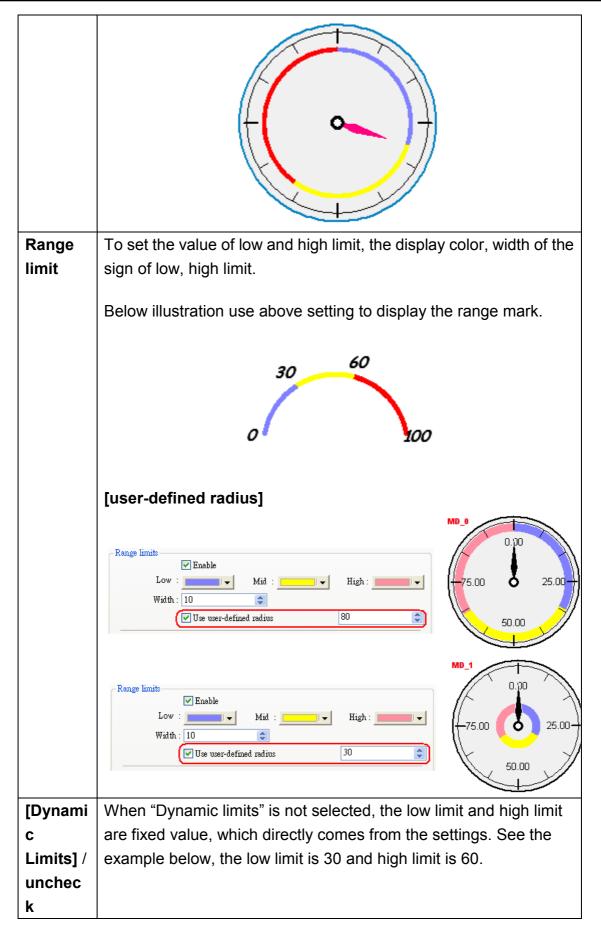
The following pictures show the "Limit" tab and the sign of low and high limit set in the "Limit" tab.

WEINTEK

New Meter Display Object			
General Outline Limits Shap			
Value			
Zero : 0	\$	Span : 100	D 🗘
Range limits			
Vidth : 3	Mid :	H	ligh :
and a second	efined radius		
Dynamic 1	imits		
PLC name : Local HMI			Setting
Address : LW	0		16-bit Unsigned
Scale label			
Use scale 1	abel		
Font : Arial			~
Color :		Size : 16	~
	Right of d	ecimal point : 0	
ļ			
	OK Ca	ncel	Help

Description	
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 deg	
= 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.	







Objects

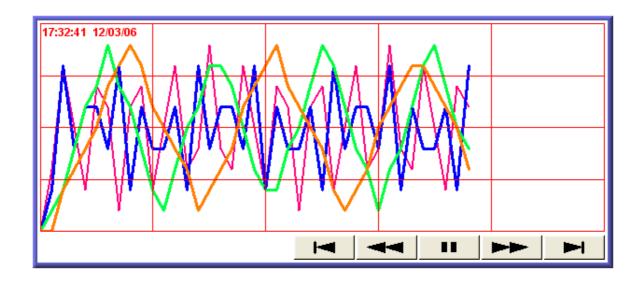
		Dynamic limits		
	Low limit 30	-	it 60 🗢	
[Dynami	When Dynamic limits is selected, the low limit and high limit are			
с	decided by the register.			
Limits] /	Click [Setting] to Select the [PLC name], [Device type],			
check	[Address], [System tag], [Index register] for Dynamic limits. Users can also set address in Limits tab while adding a new			
	object.			
		.		
	Please refer to th	e following dialog.		
	V I	Dynamic limits		
	PLC name : Local HMI Setting			
			16-bit Unsigned	
	Address : LW	• 0		
	There following ta	able shows the read addr	ress of low limit and high	
	There following ta	able shows the read addr ss" means the register's a	ress of low limit and high	
	There following ta limit. The "Addres	able shows the read addr ss" means the register's a	ress of low limit and high	
	There following ta limit. The "Addres	able shows the read addr ss" means the register's a	ress of low limit and high	
	There following ta limit. The "Addres [LW100], the "Ad	able shows the read addr ss" means the register's a dress" is 100.	ress of low limit and high address. If the register is	
	There following ta limit. The "Addres [LW100], the "Ad	able shows the read addr ss" means the register's a dress" is 100. Low limit's read	ress of low limit and high address. If the register is High limit's read	
	There following ta limit. The "Addres [LW100], the "Ad	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address	ress of low limit and high address. If the register is High limit's read address	
Scale	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2	
Scale label	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address Address bute of scale label on met	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2	
	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format To select the attrib	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address Address bute of scale label on met Scale label	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2 ter display.	
	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format To select the attrib	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address Address bute of scale label on met Scale label Font : Arial	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2 ter display.	
	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format To select the attrik	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address Address bute of scale label on met Scale label Font : Arial Color :	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2 ter display.	
	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format To select the attrib	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address Address bute of scale label on met Scale label Font : Arial	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2 ter display.	
	There following ta limit. The "Addres [LW100], the "Ad Data format 16-bit format 32-bit format To select the attrik	able shows the read addr ss" means the register's a dress" is 100. Low limit's read address Address Address bute of scale label on met Scale label Font : Arial Color :	ress of low limit and high address. If the register is High limit's read address Address + 1 Address + 2 ter display.	



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.

WE!NTEK

	-			- ii
Description :	1			_
	Data Sar	npling Object index :	0.	~
		Trend type :	Real-time	~
Note : if no. o	f channels is changed, y	ou must reset HMI's d	ata logs !!	
Distan	ice between data samples	: 💿 Pixel 🔘	Time	
	Distance	: 100 pi	xel(s)	
Hold control				
	🗹 Enable			
PLC name :	Local HMI		Setting.	
Address :		v 0		_
	L] L]		
Watch line	Enable			
PLC name :				_
			Setting.	
Address :	LW	• 0		

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	v 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

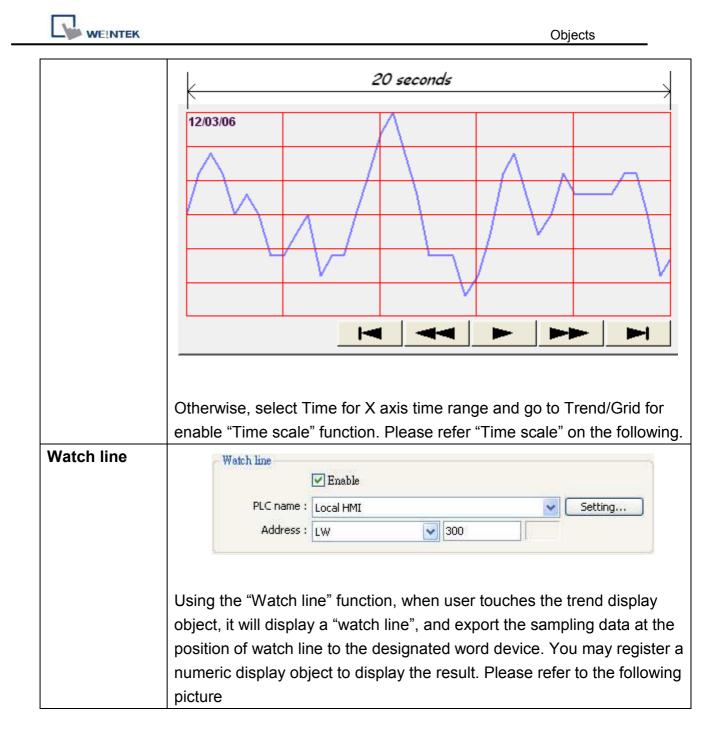
The system sorts the historical records of sampling data by date; the latest file is record 0 (In normal condition it is sampling data today), the second latest file is record 1, and so on.

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 da	ta samples : 💿 Pixel 🔿 Time
Pixel		Distance : 20 pixel(s)
	two sampling points. See	ce] can be used to set the distance between the picture below.
[X axis time	12/03/06	20 pixels
range] / Time	V soit 4	
	A axis b	me range : O Pixel O Time Distance : 20 second (s)
	Select [Time], the [Distand	ce] is used to set the X-axis in unit of time
	elapsed. See the picture b	pelow.





			Watc	h line	
12/03/06			\checkmark		
	(~
					$ \land \land \land$
			<u>×</u>	\sim \vee	\sim
	_/				
4	LW300				
The address sampling data address" The	registered i a will be exp data forma g address o	n "watch line ported to the it of each cha	is the si word dev annel ma	tart address vices starting y be differer	g from "start
For example:					
[LW300]	Ch. 0 : 16	-bit Unsigned	d (1w	vord)	
[LW301]	Ch. 1 : 32	-bit Unsigned	d (2 w	vords)	
[LW303]	Ch. 2 : 32	-bit Unsigned	d (2 w	vords)	
[LW305]	Ch. 3 : 16	-bit Signed	(1 w	vord)	

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

Ob	jects

meral Trend	Channel Shape Profile
	ame : Background :
Grid	Enable Color :
	oriz. : 4 😴 division(s) Verti. interval : 4 📚 second(s)
- Time scale -	Enable
Fo	rmat : HH:MM 💉 Font : Albertus Bold 😪
C	olor: Size: 8
Time/Date	⊙ HH:MM:SS ○ HH:MM Color :
	⊙MM/DD/YY ○DD/MM/YY ○DD.MM.YY ○YY/MM/DD
🔽 Date	

Setting	Description						
[Frame]	The color of frame.						
[Background]	The color of background.						
[Show scroll	Fo 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						



graph and "G sampling poi picture below	nt will be	· -		-			-
Verti. interval : 4	🔹 point	t(s)					
b. Time	·· • • •			c			
When select interval] is us				-		-	
See the pictu			0-			5	
Verti. interval : 4	seco	nd (s)					
According to	these of	ttingg th		m will oo	laulata th		or of
According to vertical grid li			le syster	n will ca	iculate tr	ie numb	er or
Ū		,					
Time Scale	. <u>time</u>	le ere the e	h	امم مما			
To enable the [Format]	e time sca	ie on the	dottom o	t trena als	spiay		
To select time	e scale as	HH:MM	or HH:MN	M:SS			
[Font]							
To select font	style						
[Size] To select font	t size. Red	commend	use font	size: 8.			
TD_MM/DD/YY			Next I				1
	\wedge						
			~				
			<u> </u>		<u> </u>		
		\backslash /					
/		V					
HH:MM HH	I:MM HH:	MM HH:	MM HH	:MM HH	:MM HH:	мм нн	l:MM



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

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

ener	al Trend	Chann	el Shape Profile	
Dat	ta sampling	object		
	Channel	Display	Description	Data type
	0	~	16-bit Unsigned	16-bit Unsigned
	1	4	16-bit BCD	16-bit BCD
	2		32-bit Unsigned	32-bit Unsigned
+	3	¥	32-bit Float	32-bit Float
	en property Co	lor :		Width : 2
	Co		ynamic limits	Width : 2
	Co PLC nar		ynamic limits	

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 "Dynamic	[Zero]、[Span]



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, then [Zero] and [Span] must be set as [50] and [100], so all the sampling data can be displayed in the							
	trend display object.							
Limit / check	When Dynamic Limits is selected, the low limit and high limit							
"Dynamic	are derived from the designated word device. The data length							
limits"	of the word device for limits is related to the data format of							
	object. In the exa	mple below,						
	Data Format	Low limit	High limit					
	16-bit format	Address	Address + 1					
	32-bit format Address Address + 2							
	An extended function is zoom in and zoom out function.							

Example of zoom in/out function

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

	 Dynamic limits 			
PLC name :	Local HMI		~	Setting
Address :	LW	▼ 0		32-bit Float

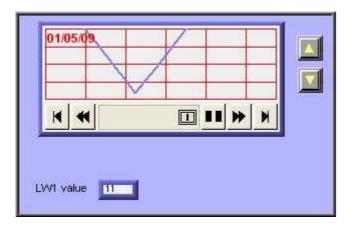
For example, the LW0 and LW1 are to control low limit and high limit, you may change the value of LW1 to zoom in / out.

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.



H		н 4	
• •		₩ ₩	

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.

K	₩ ₩	

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		16/09/07	0	0	0
3568	21.52	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.



L	WE!NTEK

neral Data Fo	rmat Title Shape		
	ampling Object index : 0.		~
Grid	🗹 Enable		
Col	or :	Column interval : 0	\$
Profile color	Aller Con		
	Transparent		
Fran	ne :	Background :	
Text Eo	nt : Arial 🗸 🗸	Size : 12	
	nt : Anal 💌	5126 . <u>12</u>	~
Time Time	НН:ММ	Color :	I¥
Date			
🔽 Date	DD/MM/YY	Color :	•
🔽 Sequence	no.	Color :	•
O Time asce	nding 💿 Time des	xending	
History control			
PLC nam	ie : Local HMI	Setting.	
Addre:	55 : LW	200	

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

Objects



		-	12	0.5				1.00	= 1		
		No.	Time	Date		Ch.0			2		
		3982	22:02	16/09		0	0	0	<u></u>		
		3981	22:02	16/09	Contraction of the	0	0	0	50		
		3980 3979	22:02	16/09	107	0	0	0	-8		
		3978	22:02	16/09	107	0	0	0	- F		
		3977	22:02	16/09		0	0	0	70		
		3976	22:02	16/09		0	0	0	<u>.</u>		
		3975	22:02	16/09		0	0	0	-8		
		3974	22:02	16/09	1.5 1.6 1.6 1.6	0	0	0	-		
			22.02	16/00	107	0	n,	10	<u> </u>		
		4						1			
	[Color]										
	Set color of grid										
	[Column interv	al]			_	_		_	_		
	Set space of co	lumn.			_						
		ate Ch				N		Tim	e,	Date	
		09/07 1 09/07 1	0		11	36		21:5 21:5		16/09/07 16/09/07	<u>+' </u>
		09/07 1		0		366		21:5		16/09/07	
		09/07 1	-	0		360		21:5	_	16/09/07	
		09/07 1 09/07 1		0		36		<u>21:5</u> 21:5		16/09/07 16/09/07	
		09/07 1	0	0		366		21:5		16/09/07	- 1
	3660 21:56 16/	09/07 0	0	0		366	60	21:5	6	16/09/07	
	3659 21:56 16/	09/07 0	0		11	36		21:5		16/09/07	
				<u></u>	1	•			· ·	mandarri	· •
Profile	Set color of fram	me and	backgi	round.	lf it	is set	t as t	ransp	arent	t, the fran	ne and
color	background will	be igno	red.					·			
Time and	Enable or disab	le the ti	me and	l date d	of da	ata sa	mplin	g and	form	at.	
Date	[Time ascendir	ng]					-				
	"Time ascending		s to pu	t the e	arlie	er data	in th	e top	and f	the latest	data in
	the bottom.		1		_			I.	-	•	
										1	
		No.	Tim		Dat			Ch.1	<u>C</u> ▲		
		1	00:24		6/09		2	2			
		2	00:24		6/09	107	4	4	_		
		3	00:24		6/09	107	7	6	_		
		4	00:24	1.30 1	6/09 8/00	07	9	8	_		
		5	00:24	1.32 1	0/09	07	6 4	4	_		
		7	00.24	1.32 1	6/09	07	4	4	—		
		8	00.24	34 1	6/09	/07	3	6	_		
		9	00.24	35 1	6/09	/07	6	6	_		
		10	00.2/	1.36 1	6/00	/07	Ř I	Λ	_		
		▲							•		
										-	



[Time descending]

"Time descending" means to put the latest data in the top and the earlier data in the bottom.

No. Time Date Ch.0 Ch.1 C 4787 22:24:15 16/09/07 2 2 4786 22:23:59 16/09/07 3 2 4784 22:23:59 16/09/07 3 2 4783 22:23:57 16/09/07 3 2 4781 22:23:57 16/09/07 3 2 4780 22:23:54 16/09/07 3 2 4781 22:23:54 16/09/07 3 2 4780 22:23:54 16/09/07 3 2 4780 22:23:54 16/09/07 3 2 4779 22:23:53 16/09/07 3 2 4778 22:23:54 16/09/07 3 2 4778 22:23:53 16/09/07 3 2 4778 22:23:54 16/09/07 3 2 4780 22:23:53 16/09/07 3 2 42:00 16/09/07 3 2 1 A778 20:23:52 16/100/07 3<									1	
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			No.	Time	Date	Ch.0	Ch.1	C 🔺		
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4787	22:24:15	16/09/07	2	2			
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4786	22:24:00	16/09/07	3	2			
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4785	22:23:59	16/09/07	3	2			
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4784	22:23:58	16/09/07	3	2			
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4783	22:23:57	16/09/07	_				
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4782	22:23:56	16/09/07	_				
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4781	22:23:55	16/09/07	-				
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4780	22:23:54	16/09/07	-				
History The history files are named with date code. The history control is used to select the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			4779	22:23:53	16/09/07		2			
Controlselect the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.			1778	00.03-20	16/00/07	1 3	2			
Controlselect the designated history data files for display. In case the value of history control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.										
control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.	History	The history files	are na	amed with	date cod	e. The	e histo	ory co	ntrol is us	sed to
control is 0, the latest file is selected. If it is 1, the second latest file is selected, and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.	Control	select the design	nated hi	story data	files for di	isplay.	In cas	se the	value of h	nistory
and so on. Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.		J. J		•						•
Click [Setting] to Select the [PLC name], [Device type], [Address], [System tag], [Index register] of History control.						, 110 3		nates.		colou,
[System tag], [Index register] of History control.		and so on.								
		Click [Setting]	to Sele	ct the [PLC	C name], [[Devic	e type	e], [Ad	ldress],	
		[System tag], []	ndex re	aister1 of	History c	ontrol	L			
Users can also set address in General tab while adding a new object.					•				a hia at	
		Users can also s	et addre	ess in Gen	ierai tab w	mie ac	ading a	anew	object.	

WE!NTEK

neral Data Format	Title Shape	
Chan	nel : Channel 0 ~ Channe	el 7 💌
Channel 0 [16-bit U		
Left of deci		Right of decimal Pt. : 0 📚
🗹 Display	Center 💽 [Leading zero
Channel 1 [16-bit E	CD]	
🔲 Display		
Channel 2 [32-bit U	nsigned 1	
	angene J	
🔲 Display		
Channel 3 [32-bit F		
	ter no. : 4 😒	Right of decimal Pt. : 0
🗹 Display	Center 💽 [Leading zero

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.	Time	Date	Ch.0	Ch.3	
5272	22:43:09	16/09/07	4	1	
5271	22:43:08	16/09/07	2	0	
5270	22:33:42		0	0	
5269	22:33:41	16/09/07	0	0	
5268	22:33:40		0	0	
5267	22:33:39		0	0	
5266	22:33:38		0	0	
5265	22:33:37		0	0	
5264	22:33:36		0	0	
5263	?? ·33·35	16/00/07	0	0	

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	

	t	Color :	I b	
Title name	Title	Label library	Label tag	^
Sequence no.	No.	None		
Time	Time	None		
Date	Date	None		
Channel O	ch.O	None		
Channel 1	ch.1	None		
Channel 2	ch.2	None		
Channel 3	ch.3	None		1
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		20

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]		



	Set the background color of title.
[Setting]	No. Time Date Ch.0 5272 22:43:09 16/09/07 4 5271 22:43:08 16/09/07 2
	You can use label tag library for title with multi-language. Go to [setting] and select one from label library.
	Title : No. Label tag : no. label Itse label library OK Cancel

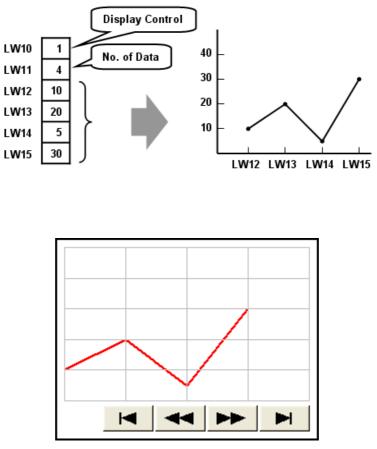
Note:

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:

eneral Display Are	a Shape		
Description :			
No. of channel :	1		
Cursor line		-	
	🔽 Enable	Color : 🗾	l 🕶
PLC name :	Local HMI	*	Setting
Address :	LW	0	
Channel :	0		
Control address -			
PLC name :	Local HMI	v	Setting
Address :		/ 10	
No. of dat	a address : 10 + 1	Offset to st	art address
Data storage sta	t address :		
PLC name ;	Local HMI	~	Setting
Address :	LW	/ 12	16-bit Unsigned
Limit			
Min. :	0	Max. : 32767	

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 [PLC name] , [Device type] , [Address] ,			
	[System tag], [Index register] of Data storage start address.			
	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.

Ob	jects

eneral Display	Area Shape Profile		
Data samples		mples to scroll : 10	\$
Profile color -	📝 Enable scroll switch		
	🔲 Transparent		
Frame	•	Background : 💻	
Grid		<u></u>	
	Enable	Color : 📕	
Horiz.	: 5 😒 division(s)	Verti. : 5	📚 division(s)
Channel			
Channel	1. Beckler		
Pen property Color	the second se	Width : 3	
_			<u></u>
-			

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



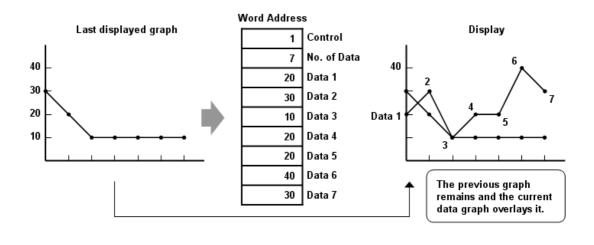
	Enable ba	ackground					a a a a a a a a a a a a a a a a a a a
			_	_			
		-	1	-			
		_	_	_			
		207	Here I.				i.
	Disable b	line - in d		~			
	Disable b	ackground		~			
	Disable b	ackground					
	Disable b	ackground		▲ ∫	▶►]	H	
	Disable b	ackground					
	Disable b	ackground					
	Disable b	ackground					
	Disable b	ackground					
	Disable b	ackground					
	Disable b	Packground					

	C Obj	ects
Grid	Enable Grid	
Channel	Set the color and width of each trend curve.	
	Channel : 0 Vidth : 1 Vidt	

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.

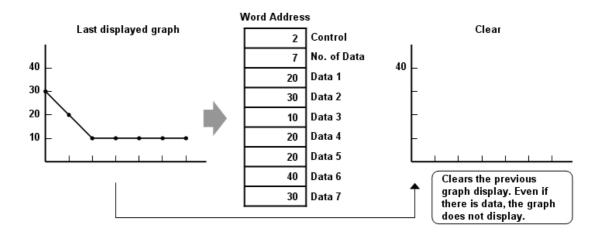


NOTE : During the period between c and d, do not change the content of [Control], [No. of Data] and [Data], it might cause error for trend curve plot.

How to clear a trend curve

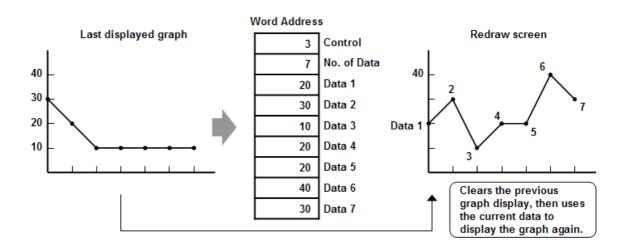
a. Write "2" to [Control word address], all the trend curves will be cleared.

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.

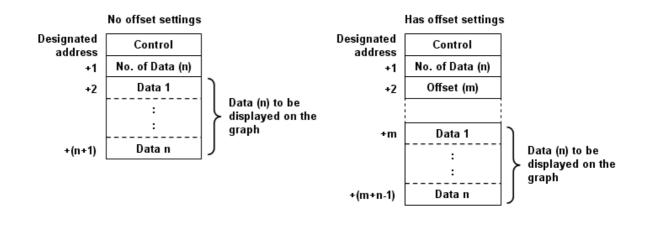




How to use offset mode

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

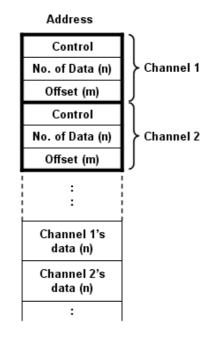
If the control register is 32 bits device, only bit 0-15 will be used as control purpose, bit 16-31 will be ignored. (as illustration below)

		32 bit	device	
3	1	16	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.



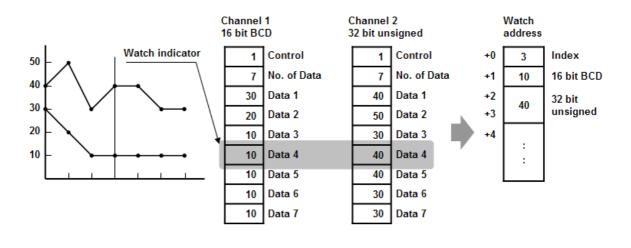
How to use watch (Cursor Line) feature



	🗹 Enable 🦳 📃	
PLC name :	Local HMI	~
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.

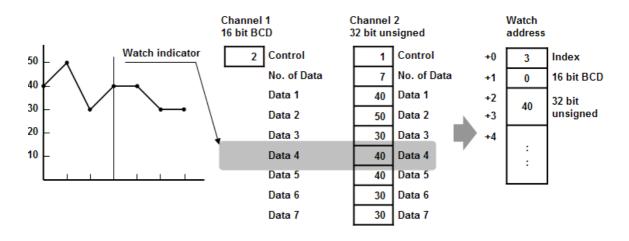
In the following example, the data block display contains two data blocks. The data format of channel 1 is 16 bit BCD and that of channel 2 is 32 bit unsigned. The cursor is positioned in data index 3 which is corresponding to the fourth data in data block. The system writes "data index" and the content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



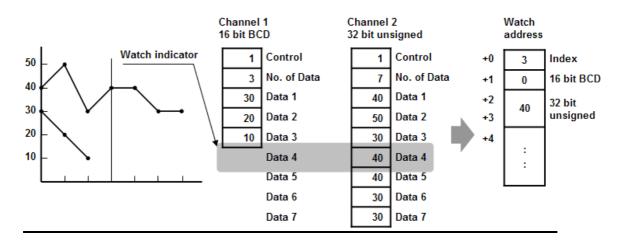
- **NOTE** 1. [Data Index] is a 16 bit unsigned integer; when the designated register of cursor line is 32 bit device, it will be stored in the bit 0-15.
 - 2. The watch function can only inspect current value in the data block. If there are multiple trend curves of the same channel on the screen, the 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.



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.

Objects

Obi	ects
~~	

eneral Display Are	a Shape		
Description :			
Direction :	Right 😽	No. of channels : 2	*
Control Address	- 0:		
PLC name :	Local HMI	~	Setting
Address :	LW	10	
No, of data a	address : [10 + 1		
Channel :	0		
Read address	N		
PLC name :	Local HMI		*
X data	✓ Separated address for Σ	(and Y data	
PLC name :	Local HMI	Setting	
Address :	LW 🔽	100	16-bit Unsigned
Y data			
PLC name :	Local HMI	4	Setting
Address :	LW	200	16-bit Unsigned
Limits			
X axis	Dynamic limits		
Low :	0	High : 32767	
Y axis	6)		
Low :	0	High : 32767	

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



	Up: Down:
	Op direction origin→ × ↓ origin→ Y × Down direction
	 b. No. of channel. Set the no. of channels of the XY plot. Each channel may conduct the draw operation alone.
Control	[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 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
	Address : LW V 100 16-bit Unsigned
	Click Cotting 1 to Coloct the IDLO normal (Device true) [Address]
	Click [Setting] to Select the [PLC name], [Device type], [Address], ,
	[Index register], for read address.
	 The usage of each address as follows, (Dynamic limits is not enabled.)
	For example:
	The Read address is LW100.
	X data 0 reads value from reading address LW100.
	X data 1 reads value from reading address LW101.
	X data 2 reads value from reading address LW102.
	X data 3 reads value from reading address LW103.
	X data 4 reads value from reading address LW104.
	X data 5 reads value from reading address LW105 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 dat	a 0 reads valu	e from rea	iding address L	.W104.	
Y dat	a 0 reads valu	e from rea	ding address L	.W105.	
X dat	a 1 reads valu	e from rea	iding address L	.W106.	
			iding address L		
lf voi	i check "Sena	rated add	ress for X and	Y data"	it allows you to s
•	•		axis respective		
amor	Read address			· y .	
	PLC name	Local HMI			~
	200	Separated	address for X and Y da	ıta	
	- X data	Local HMI		-	Setting
	Address :		✓ 100		16-bit Unsigned
			100	11/	
	- Y data	Local HMI			
			- Incor	10	Setting
	Address	. LW	200		to-pic onsigned
	xample: Read address i	is LW100	and LW200.		
X dat	a				
X low	/ limit reads va	lue from re	eading address	LW100.	
X hig	h limit reads va	alue from	reading addres	s LW101	
X dat	a O roade valu				•
X dat	a u reaus vaiu	e from rea	ding address L	.W102.	
			-		
X dat	a 1 reads valu	e from rea	iding address L	.W103.	
	a 1 reads valu a 2 reads valu	e from rea e from rea	iding address L iding address L	.W103. .W104.	
	a 1 reads valu a 2 reads valu a 3 reads valu	e from rea e from rea	iding address L iding address L iding address L	.W103. .W104.	
X dat Ydata	a 1 reads valu a 2 reads valu a 3 reads valu a	e from rea e from rea e from rea	iding address L iding address L iding address L	.W103. .W104. .W105 ar	
X dat Ydata Y low	a 1 reads valu a 2 reads valu a 3 reads valu a / limit reads va	e from rea e from rea e from rea lue from re	iding address L iding address L iding address L iding address L	.W103. .W104. .W105 ar .LW200.	nd so on…
X dat Ydata Y low Y hig	a 1 reads valu a 2 reads valu a 3 reads valu d imit reads va h limit reads va	e from rea e from rea e from rea lue from re alue from r	ading address L ading address L ading address L ading address L eading address reading address	.W103. .W104. .W105 ar .LW200. s LW201	nd so on…
X dat Ydata Y low Y hig Y dat	a 1 reads valu a 2 reads valu a 3 reads valu ia / limit reads va h limit reads va a 0 reads valu	e from rea e from rea e from rea lue from re alue from rea	ading address L ading address L ading address L ading address L eading address	.W103. .W104. .W105 ar .LW200. s LW201 .W202.	nd so on…
X dat Ydata Y low Y hig Y dat Y dat	a 1 reads valu a 2 reads valu a 3 reads valu ia / limit reads va h limit reads va a 0 reads valu a 1 reads valu	e from rea e from rea e from rea lue from rea alue from rea e from rea	ading address L ading address L ading address L ading address L eading address reading address ading address L	.W103. .W104. .W105 ar .LW200. s LW201 .W202. .W203.	nd so on…

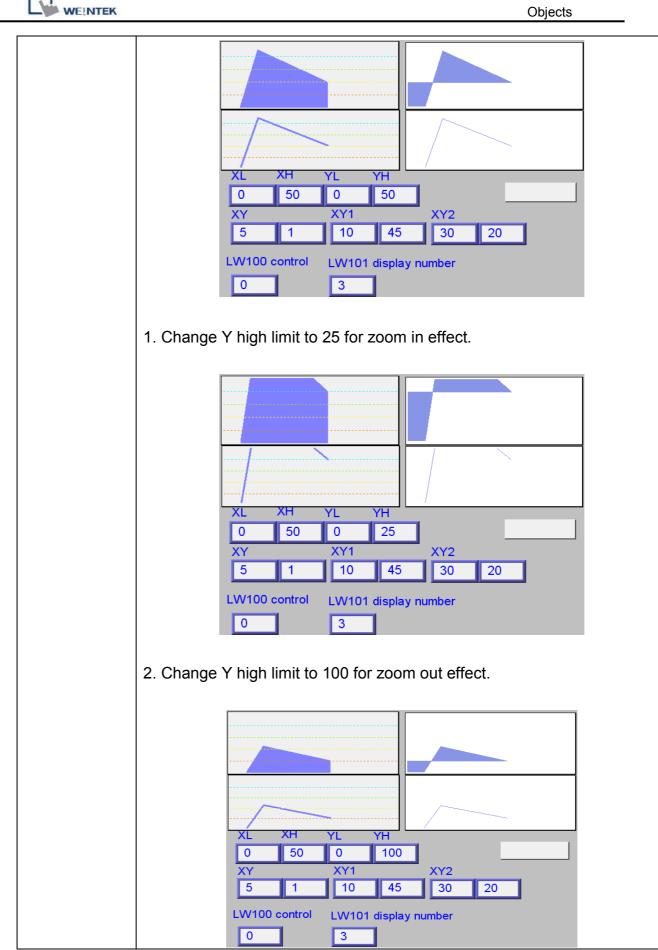


Limits	The above settings are based on dynamic limits, you can also have dynamic limits disable and set the fix high and low limits.
	Limits Dynamic limits
	Low : 0 High : 32767
	Y axis Low : 0 High : 32767
	The high and low limits is used as scale to calculate the percentage of X
	and Y axis. i.e. X or Y % = (X or Y reading value – low limit) / (high limit – low limit)
	Based on your settings, the memory allocation for limit and XY data will be as follows.
	The following setting is for 16-bit signed data format and dynamic limits.
	PLC name : Local HMI
	PLC name : Local HMI Setting
	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 V 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)
NOTE
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 ad	dress for X and Y da its	ta Dynamic limits		
	X	Y	X	Y	
	Data 0	, Data 0	Min	Min	
	Data 1	Data 1	Max	Max	
	Data 2	Data 2	Data 0	Data0	
	Data 3	Data 3	Data 1	Data1	
	:	:	Data 2	Data2	
	Separated ad	dress for X and Y da	ita:		
	Dynamic lim		Dynamic limits		
	X	+ Y	X +	Y	
)ata ()	X Mir		
	ΥE)ata 0	XMa		
)ata 1	Y Min		
)ata 1)ata 2	Y Max XDat		
)ata 2	YDat		
)ata 3	XDat	a1	
	Y Data 3		YDat		
		:	XDat YDat		
			T Dai	a∠	
	If dynamic limit is che	cked, you ma	ly change the	high and	low limits to
	realize zoom in and	zoom out fu	nction. (Pleas	e refer t	rend display
	object's dynamic limit.)				
	In the following examp	le, the dynam	nic limit is sele	cted, whe	ere XL=X low
	limit, XH=X high limit, א	L=Y low limit	, YH=Y high li	mit, and X	(Y, XY1, XY2
	are three XY data. Nov	v we change	the high limit c	of X and Y	′ respectively
	and you may observe t	_	-		· · · · · · · · · · · · · · · · ·



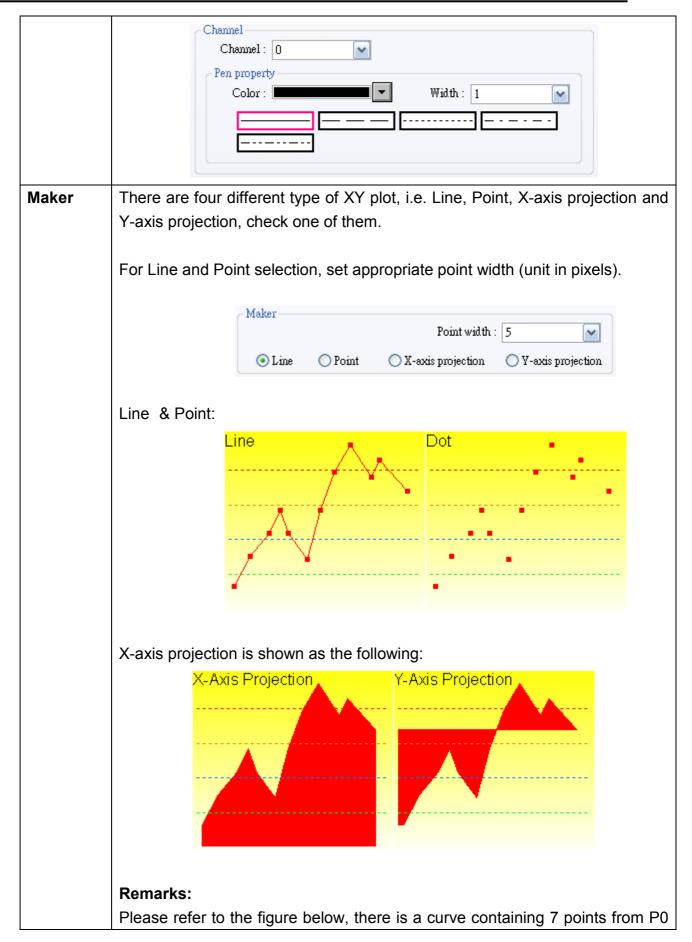


[Display Area tab]

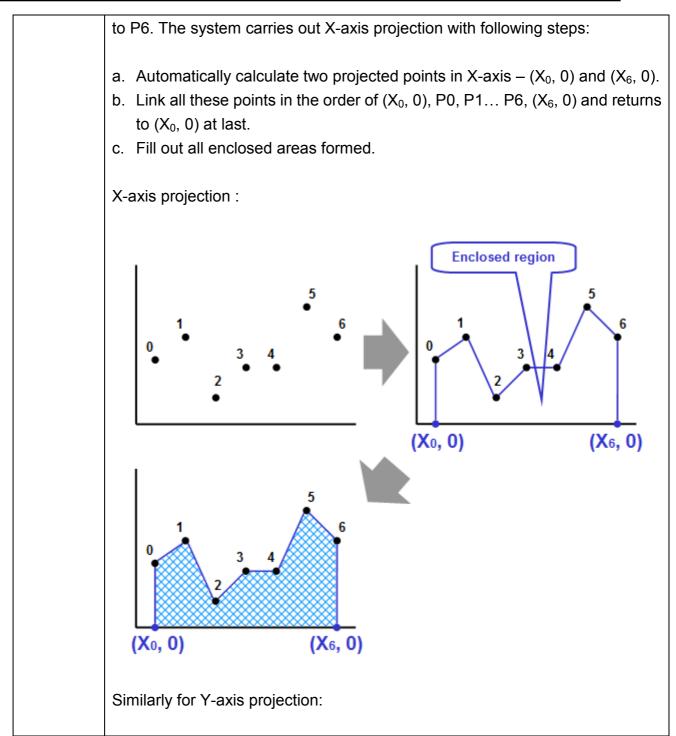
	rea Shape		
Profile color			
	🗹 Transparent		
Curve			
Channel	0		
Pen property —	<u>.</u>	1.	
Color	:	Width : 1	*
Maker			
Maker		Point width : 5	
100	2 2		
💽 Line	○ Point ○ X-ax	cis projection 🛛 🔿 Y-axi:	s projection
D.C			
Reference line			
	Limit from PLC		
Reference line Low limit		High limit : 100	
] High limit : 100	
		High limit : 100	
	0	· · · · · · · · ·	
	C Reference line 1	20	
	© Reference line 1	20	

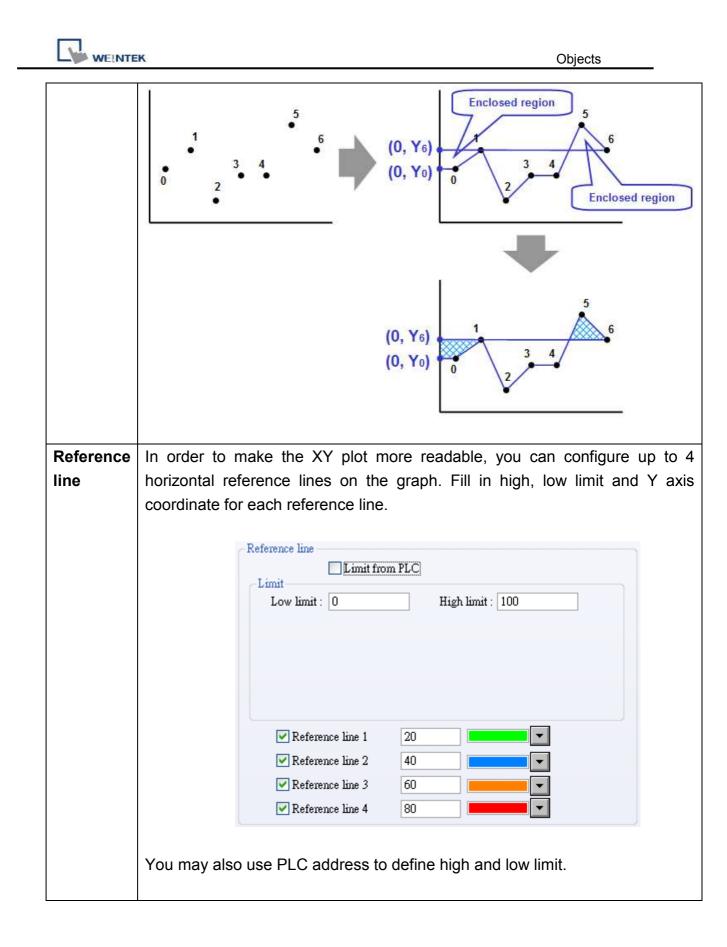
Setting	Description
Profile	Enable Transparent: It will not display the background color.
color	Disable Transparent: It will display the background color
Curve	Set the attribute of XY curve (color and width) for each channel.

Objects











Objects

Reference line	from PLC		
PLC name : Local Hr Address : LW	1I 🔽 O	~	Setting 16-bit Unsigned
✓ Refer ✓ Refer	ence line 1 20 ence line 2 40 ence line 3 60 ence line 4 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.

t (When LW 1 >= 10) 13:21:06 Event 0 (when LW0

3112106	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)

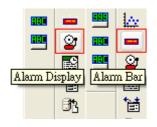


Alarm display object

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.





larm	Shar	pe Font			
I	nclude	categories : 0 thru 0	{see	Alarm (E	vent) Log object}
	Sc	roll speed : Speed 6	Y A	lcknowled	ge style : Click
Color					
		🔲 Transparent			
		Frame :	() + ()	Backg	round : 🚺 🗖
	Time a	scending 💿 Time de naracters	scending:		1
2011		Display items			Display order
	4	Event trigger date Event trigger time			Event trigger date Event trigger time
	¥	Event message		•	Event message
				•	
If "I	Display	chars" is 0, it means that the	system will] display al	l of characters.
Date	ММ	/DD/YY V Time	: HH:MM	:55	~
2.00					

Setting	Description
Include	Select category of events that belongs to the alarm display
categories	or alarm 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.
	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.

Attribute				Shape	Alarm
Font Comic Sans MS	~		Comic Sans MS	ute	Attrib
Color : Size : 12	~	Size : 12		a constraint in the	

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.

	Event 1 (LB10 = ON)		15:35:22	01/20/09	
	Event 0 (LB0=ON)		15:35:22	01/20/09	17
_Multi-tex	Event 3 (LW20 <= 5)	15:35:23	15:35:20	01/20/09	16
	Event 0 (LB0=ON)		15:35:19	01/20/09	15
	Event 1 (LB10 = ON;		15:35:18	01/20/09	14
	Event ?	15.25.16	15.25.15	01/20/00	12
		a la c a suiz			

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.

	Event Displ	ay Shape	Font			
	Description :					
		Real-time	*			
Write	e address :					
	PLC name :				*	Setting
	Address :	LW	*	0		16-bit Unsigned
		🔲 Enable ever	ut managemo	ent		

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 the [write	address] of [event display] object.				
Write value for event display —					
Write value : 200					
b. History Control					
• [Enable reading multip	ble histories] not selected.				
In this mode it displays event log from history record. The system save the event history in daily basis. The event history of each date is saved in separated files with date tags attached. The "History control" is used to select one history record file.					
The picture below sho designates a word device	ows the "History control" setting, which ce for "History control".				
History control)				
Device type : LW	×				
Address : 100	System tag				
	Index register				
16-bit Unsigned					
1 refers to the history re	(normally it is history record today). Index cord one day before the latest, and so on. story control" register is used as the index history record.				
	explain how to use "History control". The r is [LW100], supposed that the history nare				
Where 2006xxxx is the date of system saved history record. The following table shows the corresponding historical record displayed be event display object according to the value of [LW100].					
Value of [LW100]	Corresponding Historical Record				
	EL 20061202 out				
0	EL_20061203.evt				



EK			Obje
	1	EL 20061127.ev	vt
	2	 EL 20061123.ev	vt
	3	 EL 20061120.ev	vt
• [E	inable reading multip	le histories] selec	ted.
Def	inition: Displays a lis	t of events trigger	ed in multiple days.
Illus	tration: Take LW0 to	be the [History (Control] [Address] as
		•	ayed will be formed by
		ue in LW0 represe	ents the first history da
to s	tart with.		
Evo	mple: As illustrated	bolow for obowing	a it alcoror, the history
	a is numbered accord		g it clearer, the history
uate			
(No	.0、No.1、No.2).	If the value in LW	0 is "3", the first data to
be d	displayed will be data	a No. 3.	
	EL_20100604	No.4	1 KB EVT 檔案
	受EL_20100605 受EL_20100608	No.3 No.2	6KB EVT檔案 17KB EVT檔案
	📴 EL_20100609	No.1	4KB EVT檔案
	🕎 EL_20100610	No.0	12 KB EVT 檔案
AS 1	or LW1, 2 modes ca	n de selected.	
a.	Number of days		
	History control		
	PLC name : Local HMI		Setting
	Address : LW	✓ 0	16-bit Unsigned
	Enable p	eading multiple histories	
	Mode : Number of		× 1
6	0		
	•		will start from number
		•	nany days to be includ
tron	n the start to days be	store.	
Eve	mplo: An illustrated	bolow if volue of l	
	•		LW0 is "1", LW1 is "3",
uier	i the range of data w	mi start form 2010	0609, and include dat

of 2 days before (while 20100609 itself is counted). We can see





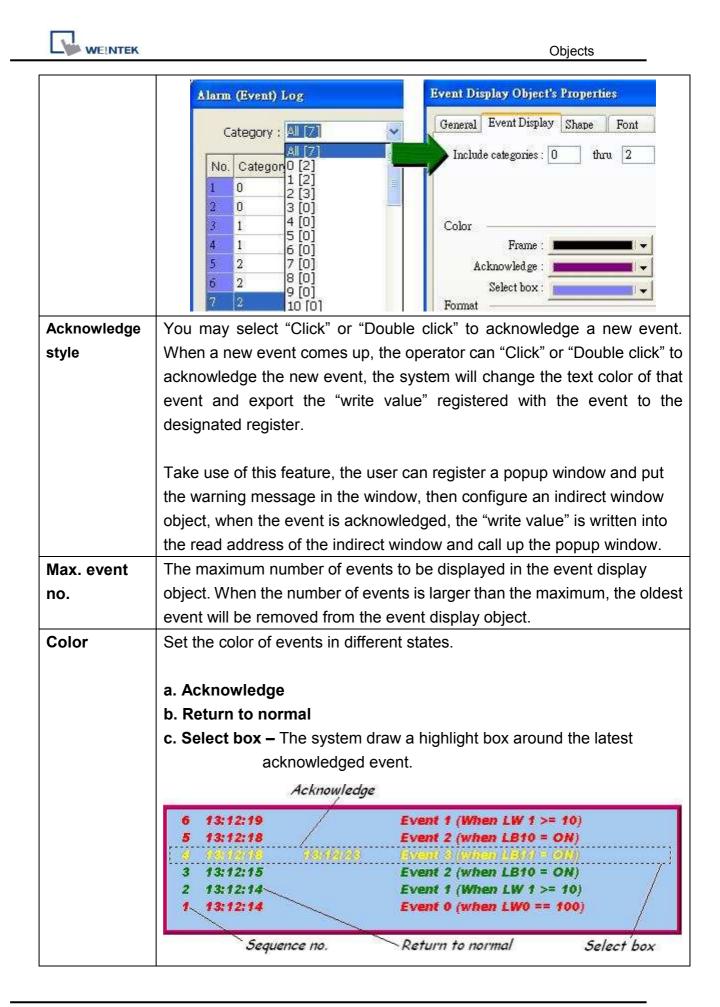
that in this example, since data of 20100607 does not exist, the
data displayed will only include 20100609 and 20100608.
[1] EL_20100604 No.4 1 KB EVT 檔案
「WEL_20100605 No.3 6 KB EVT 檔案 No.2 17 KB EVT 檔案
第EL_20100608 No.2 17 KB EVT 檔案 第EL_20100609 No.1 4 KB EVT 檔案
[[] EL_20100610 No.0 12 KB EVT 檔案
b. 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).
[]] EL_20100604 No.4 1 KB EVT 檔案
PEL_20100605 No.3 6 KB EVT 檔案
[9] EL_20100608 No.2 17 KB EVT 檔案
[1] EL_20100609 No.1 4 KB EVT 檔案
[禁 EL_20100610 No.0 12 KB EVT 檔案]] [] [] [] [] []]] []
The maximum size of data that can be displayed by system is
4MB; the exceeding part will be ignored.
The following shows how data will be stored while the data size is
too big.
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 hide	den.
2. In [Re	al- time] mode, select events to be deleted.
Illustration:	
- Control address -	
PLC name :	Local HMI Setting
Address :	LW V 100 16-bit Unsigned
	Enable event management
1. When the va	of History control is set LW100: alue in [LW100+0] is "0" \rightarrow All events will be
 When the va displayed. 	alue in [LW100+0] is "0" → All events will be alue in [LW100+0] is "1" →The confirmed events
 When the variable of the variable	alue in [LW100+0] is "0" → All events will be alue in [LW100+0] is "1" →The confirmed events en. alue in [LW100+0] is "2" →The recovered events
 When the variable of the displayed. When the variable of the displayed. When the variable of the displayed. When the variable of the displayed of the displayed. 	alue in [LW100+0] is "0" → All events will be alue in [LW100+0] is "1" →The confirmed events en. alue in [LW100+0] is "2" →The recovered events
 When the variable of the displayed. When the variable of the variable of the displayed. 	alue in [LW100+0] is "0" → All events will be alue in [LW100+0] is "1" →The confirmed events en. alue in [LW100+0] is "2" →The recovered events en.
 When the variable of the displayed. 	alue in [LW100+0] is "0" → All events will be alue in [LW100+0] is "1" →The confirmed events en. alue in [LW100+0] is "2" →The recovered events en. alue in [LW100+0] is "3" →The confirmed and

16	940337 572	vent Display Shape	Font (mo	Alarma /E	vent) Log object}
104	nciu	the categories . U	ru O (see	Alaim (E	vent) Log object}
			А	.cknowled	lge style : Click 🛛 🖌
				Max. e	vent no. : 200
Color	-	100 - 100			0
		📃 Transpa	rent		
		Frame :	-	Backs	ground :
	Ac	knowledge :	R	etum to r	
		Select box :			
		Select DOX .			
orma Sort	÷.				
ALC: NO		e ascending 💿 Ti	ime descending		
0	тшк	e ascenantă 🔍 Li	nue descending		
Ord	er & (Characters		10	×
		Display items	Display chars		Display order
Þ	4	Sequence no.	0		Event trigger time
	4	Event trigger date	0		Event message Return to normal time
	4	Event trigger time	0		Acknowledge time
	4	Acknowledge time Return to normal time	0		Event trigger date
	Y	Event message	0	-	Sequence no.
		PAcifi message	U.		
If."	Displ	ay chars" is 0, it means th	et the curstern will	dienlast al	l of characters
	Dispi	ay chors is 0, it means in	or the system will	uispiay a	a or characters.
Date	M	M/DD/YY 🔽	Time : HH:MM	:SS	~
Date	1.733		200000 - 10000	10.05	NUMBER OF THE OWNER

Setting	Description
Include categories	Select category of events that belongs to the event display object. (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.





Format	trigger date trigger time notification time return to normal time				
	0 12/14/06 15:26:21 15:26:31 15:26:36 Event 0 (when LV				
	2 12/14/06 15:26:47 16:26:50 Event 1 (When L) 2 12/14/06 15:26:48 Event 2 (when LE				
	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.				
	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				

The font tab sets the font size and italic attribute. The font of event message is set with the event log object.

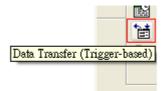
13.23 Data Transfer (Trigger-based)

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.



L	WEINTEK

eneral ;	Security S	hape Label	
D	escription :	t	
Source	address —		
F	LC name :	Local HMI	Setting
	Address :	LW 💟 D	
Destinat	ion addres	ананананананананананананананананананан	
F	LC name :	Local HMI	Setting
	Address :	LW 🔽 100	
Attribut		01	
No	. of word :		
	Mode :		ON->OFF 💽
Trigger	address	Touch trigger External trigger	
P	LC name :	Local HMI	Setting
	Address :	LB 🔽 0	

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 triggor					
	a. Touch trigger Press the object to activate data transfer operation.					
	b. External trigger					
	Register a bit device to trigger the data transfer operation.					
	$[ON \rightarrow OFF]$ Bit device change from ON to OFF to estimate data transfer					
	Bit device change from ON to OFF to activate data transfer operation.					
	$[OFF \rightarrow ON]$					
	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 Setting					
	Address : LB 🔽 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.



L	WEINTEK

	Security 3	Shape Lab	el			
D	escription :	(17
Source					772)	
○R	W ():	RW_A 🧿) Historical eve:	nt log	🔘 Historical (lata sampling
Backup	position –					
💿 U	SB 1	🔿 USB 2	⊖ SD c	ard		
OR	emote print	er/backup serv	/er			
Note	: Use LW-9	9032~9039 to	change the bac	kup folder	name.	
Note in [S]	: Use [Rem ystem Paran	ote printer/bac neter][Printer/l	kup server] to Backup Server]	store data] settings.	to a remote PC.	Enable the server
Save fo	rmat					
Save fo	1000 A.	MT8000 Ev	ent Log File (*	.evt)		*
Save fo	1000 A.	MT8000 Ev	ent Log File (*	.evt)		v
	Format :	MT8000 Ev	ent Log File (*) Yesterday			•
	Format :	💿 Today				~
Range	Format : Start : Within :	💿 Today				v
	Format : Start : Within :	⊙ Today All	🔿 Yesterday			
Range Trigger	Format : Start : Within : Mode :	💿 Today	🔿 Yesterday			
Range Trigger	Format : Start : Within : Mode :	 Today All External trig 	🔿 Yesterday			
Range Trigger Trigger	Format : Start : Within : Mode : Condition :	 Today All External trig OFF->ON 	🔿 Yesterday			Setting

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 :
	RW ORW_A OHistorical event log OHistorical data log Data Sampling object index : 1.
	Backup position
Backup Position	Select the destination where the source files will be copied to.
	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 <i>MT remote printer/backup</i>
	server from:
	Menu
Save	User can select the desired format to back up the file.
format	a. MT8000 Event Log File (*.evt) / MT8000 Data Log File (*.dtl)
	b. Comma Separated Values (*.csv)
	 Event Log saved as csv file
	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.

Objects

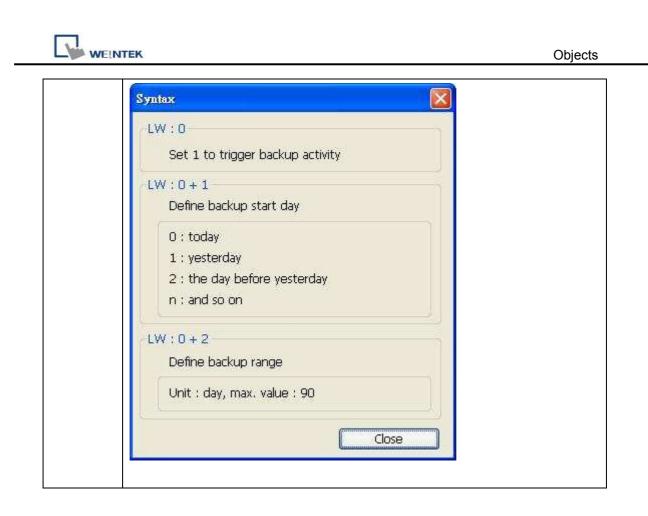
|--|

		12 13 1	1 AIA		遇 + 🥩 🕒	7 - 1 🧕	
			1 16 201		8 9 02 1		
		D20		÷	fx		
		A	B	C	D	E	
	1	[Creation t	ime]				
	2	and the second se	10:59:28 2	010			
	3	[Data field:	s]				
	4	event	category	time	message		
	5	[Data]					
	6	0	2	11:19:42	"Emergeno	sy"	
	7	0	8	11:19:43	"5"		
	8	2	(m.20)	11:19:46	"LOW"		
	9	2	2601	11:19:49	"5"		
	10	1	54761	11:19:52	Contraction and American Contraction of Contractio		
	11	2	0	11:19:52	"Word"	1	
	12						
	13				-		
	14						
		event is	triggorod				
	0	EVENUIS	ulyyeleu				
	1 -						
	1 -> event is acknowledged						
	2 >	ovent ref	turns to n	ormol			
ange	[Sta	rt] from [Today] o	r [Yeste	rday]		
	[Within]						
	Sele	Select the range of time period, for example, Select [Yesterday] in					
			•	•		• • •	
						o save the file	
	and	the day	before y	vesterday	/. Select	"All" to save	all the files
	avai	available in the system.					
		· · · · ·	- ,				
			Ran	ge			
				Start : 💿	Today 🤇) Yesterday	
				_			
	1		1	Within : 🔝		<u>~</u>	
				Al			
					lay(s)		



Attribut	There are two ways to activate Backup function.						
е	-						
	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.						
	$[OFF\toON]$						
	Bit device change from OFF to ON to activate backup operation.						
	Bit device change state to activate backup operation.						
	_						
	Trigger 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 Setting						
	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].						

- Trig	ger	12	200			1
	Mode :	External trigge	r (word) 🛛 👻	2	yntax)
Trig	ger address —					
	PLC name :	Local HMI			Setting	
	Address :	LW	0	5		
set to LW-0 back	o LW-0) : D: When t up.	he value of	suppose the of this address of the of this address is for sp	changes	from 0 to 7	1, trigger
	•	ta in this ad	dress is for sp	ecifying	the numbe	r of days
	·	shown belo)W:			



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.)



eneral 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
Ext. device			
O SD @	USB1 🔘 USB2	Folder name :	example
Attribute			
📃 Auto, repea	t	Background :	

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)

n e E	12			11	
PLC name :	Local HMI			× (Setting.
Address :	L₩	~	0		
Command :[0		Statu	s: 0+3	
Parameter 1 : [0+1		File index	c : 0 + 4	
Parameter 2 :	0+2		Start time	0+5	

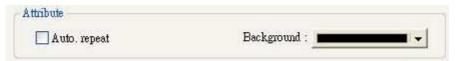
b. In [Control address], unselect the [Update video playing time]

🔲 Update video playing time

c. In [Ext. device], select [USB1] and input "example" as [Folder name].

Ext. device -			
O 2D	💽 USB1	🔿 USB2	Folder name : example

d. In [Attribute], unselect [Auto. repeat] and choose black as the background color.



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 :

eneral Preview			
Description :	-		
Control address —			
	💌 Enable		
PLC name	Local HMI		*
Device type	LW		¥
Address	0	System tag	
Address format	: DDDDD [range : 0 ~ 1	0500]	
		🔲 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 playi	ng time	
Ext. device			
OSD @	USB1 🔿 USB2	Folder name : example	
Attribute			
		Background :	

Setting		Description
	Enable control	• Enable
Control address	address	 a. You can use "Control address" to control the operation of media player b. Register a device address for "Control address".
		 Disable There is no manual control of video play operation. The system will start to play the first



		video at designated folder when the window is
1		popup.
Comma	nd	Users set this address to control the operation of
		media player.
		 Command (control address + 0)
Paramet	er 1	Parameter 1 for control operation.
		 Parameter 1 (control address + 1)
Paramet	or 2	Parameter 2 for control operation
		 Parameter 2 (control address + 2)
Status		The system will turn bits ON when state changes or
		malfunctions.
		 Status (control address + 3)
File inde	X	The system will write file index when starting to play
	<i>,</i> ,,	a video.
		 File index (control address + 4)
Start tim	10	The system will write video start time when starting
		to play a video. (unit = sec) (Always 0)
		 Start time (control address + 5)
End time	2	The system will write video end time when starting to
		play a video. (unit = sec)
		 End time (control address + 6)
	Update	Enable
	-	The system will write video elapsed time into
		[playing time] register in every [update period]
Video	time	seconds.
		Update period of [playing time], range between 1 to
time	•	60 sec.
	Playing	Update the video elapsed time periodically. (unit =
	time	sec)
		Playing time (control address + 7)
SD	1	Play video files in SD card.
USB1		Play video files in USB1.
		Play video files in USB2.
	ame	The name of the folder storing video files. Users
		must put video files in a folder (e.g. <i>"/example"</i>)
		instead of root directory.
	Paramet Status File inde Start time Video playing time SD USB1 USB2	File index Start time End time Video playing time Update video playing time Update playing time Playing time Playing time SD USB1



		Note
		1. [Folder name] couldn't be empty.
		 [Folder name] couldn't include /\:*?"<> .
		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.
Attribute		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)

Note 1. 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)

```
Note Parameter 1 (target location) should less than end time. If it is over end time, the system play video from last second.
```

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 [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 ~ 128) [Parameter 2] = ignore (set 0)

Note Default volume is 128.

j. Set video display size [Command] = 10 [Parameter 1] = display size (0 ~ 16) [Parameter 2] = ignore (set 0)

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.

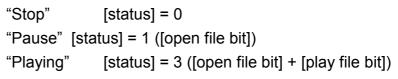
k. Status (control address + 3)

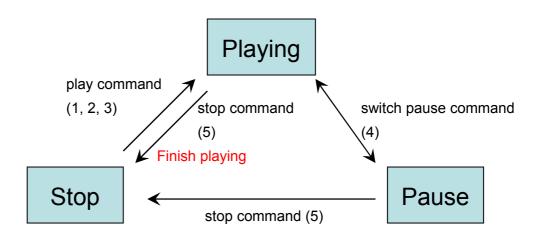
	15	09	08		02	01	00	bit
	Reserved (all 0)	0	0			0	0	
Bit 00: open fi	le bit (0: file closed ;	1: 1	file	opened)				
Bit 01: play file	e bit (0: not playing v	vide	о;	1: playing video)				
Bit 08: comma	and error bit (0: comr	nar	nd a	ccepted ;				

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 $(0\rightarrow 1)$.

Note 1. If file format is unsupported or disk I/O error happens during playing (e.g. 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:





* 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.

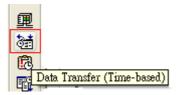
13.26 Data Transfer (Time-based)

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:



[Local HMI:LW250] -	> [Local HMI:LW260],	Mode : Word, Time	interval=2.0 second(s)	ansfer length=10 bit(s)), transfer length=1 wi	ord(s)

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.

L	WEINTEK

Description :				
Attribute				
Address type :	Bit	*	Interval :	3.0 second(s)
No. of bit :	1			
	Active actives	on decignate	d window one	here
	Active only wh	en designate	a williaow ope	ancu -
		en designate		aleu -
	Active only with	en designate	a window ope	
iource address		en uesignate		areu
iource address PLC name :		en uesignate		Setting
	Local HMI			
PLC name : Address :	Local HMI LB			
	Local HMI LB			

Setting	Description					
Attribute	[Address type]					
	Select the bit or word device.					
	[No. of words] or [No. of bits]					
1	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 bit, set the					
	number of data to transfer. See the picture below.					
	Attribute					
	Address type : Bit V 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
	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.
Destinatio	Set destination address.
n 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

1: [Local HMI:LB200] -> [Local HMI:LB210], Mode : Bit, Time interval=2.0 second(s), transfer length=10 bit(s 2: [Local HMI:LW250] -> [Local HMI:LW260], Mode : Word, Time interval=2.0 second(s), transfer length=1 \ 3: [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.



PLC	Control Objec	
1: 2: 3: 4:	LW150 LW151 LW400 LB300	Change window Write data to FLC(current base window) General PLC control Back light control(write back)
	New	Delete Settings Exit

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.

Description :					
					1700
PLC name :	Local HMI				×
Attribute					
Type of control :					~
Active only w	Change window Write data to P	v I Cícurrent	base window)		
Turn on back	Back light contr Back light contr	ntroi ol (write ba	ck)		
Trigger address	Sound control Screen hardcop	1 17			
PLC name :	Local HMI	2		X	Setting
Address :	LW	*	0		16-bit Unsigned

Setting	Description			
Attribut	[Type of control]			
e &	To set the type of control. Click the select button and you can drag			
Trigger	down a list of all available PLC control functions			
address	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 [Trigge address] is written in a valid window number, the system will close the current window and open the window designated by the			



PLC Control		
Description :		
PLC name : Lo	ocal HMI	
Attribute		
Type of control : Ch	hange window	-0
Active only whe	n designated window opened	10. WINDOW_010
Turn on back lig	ht 🗹 Clear data afte	r window changed
Trigger address		
PLC name : Lo	ocal HMI	Setting
Address : L	W 🗸 0	16-bit Uns
window number – window and open w If you use 32-bit d the trigger address	the above configur 11 into LW0, the sy window 11, then write evice as trigger addr s is in word basis, th to [Trigger address +	ystem will close the e 11 into LW1 (LW0- ress, and the device ien the system will w
window number – window and open w If you use 32-bit d the trigger address window number inf	11 into LW0, the sy window 11, then write evice as trigger addu s is in word basis, th	ystem will close the e 11 into LW1 (LW0- ress, and the device ien the system will w ·2].
window number – window and open window and open window and open with the trigger address window number informat.	11 into LW0, the sy window 11, then write evice as trigger address is in word basis, th to [Trigger address + write address for eac Trigger address	ystem will close the e 11 into LW1 (LW0- ress, and the device en the system will w ·2]. ch different type of da
window number – window and open window and open window and open with the trigger address window number interpret address windo	11 into LW0, the sy window 11, then write evice as trigger addr s is in word basis, th to [Trigger address + write address for eac	ystem will close the e 11 into LW1 (LW0- ress, and the device ien the system will w ·2]. ch different type of da <u>Write address</u> Address + 1
window number – window and open window and open window and open window use 32-bit dathe trigger address window number international backwork of the list of format.	11 into LW0, the sy window 11, then write evice as trigger address is in word basis, the to [Trigger address + write address for each Address Address	ystem will close the e 11 into LW1 (LW0- ress, and the device ien the system will w ·2]. ch different type of d <u>Write address</u> Address + 1 Address + 2
window number – window and open window and open window and open with the trigger address window number interpret address windo	11 into LW0, the sy window 11, then write evice as trigger address is in word basis, th to [Trigger address + write address for each Trigger address Address	ystem will close the e 11 into LW1 (LW0- ress, and the device ien the system will w ·2]. ch different type of d Write address Address + 1
window number – window and open window and open window and open window and open with the trigger address window number informat. Below is the list of format. Data Format 16-bit BCD 16-bit BCD 16-bit Unsigned 16-bit Signed	11 into LW0, the sy window 11, then write evice as trigger address is in word basis, th to [Trigger address + write address for each Address Address Address Address	ystem will close the e 11 into LW1 (LW0- ress, and the device ien the system will w 2]. ch different type of da <u>Write address</u> Address + 1 Address + 1 Address + 1
window number – window and open window and open window and open window and open with the trigger address window number interview address address window number interview address addre	11 into LW0, the sy window 11, then write evice as trigger address is in word basis, the to [Trigger address + write address for each Address Address Address	ystem will close the e 11 into LW1 (LW0- ress, and the device hen the system will w 2]. ch different type of da <u>Write address</u> Address + 1 Address + 2 Address + 1



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 → 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 lis
address]		in the above table. When a r
		control code is written into
		register, the system will cond
		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	

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)"
Set [Trigger address] to "ON", the system will turn on/off the
backlight and reset the [Trigger address]. Any touch on the screen
will turn the backlight on.
e. "Back light control"
This operation is the same as "Back light control (write back)"
except the system would not reset the [Trigger address].
e. "Sound control"
Sound
Sound Library Beep
Play Trigger mode : OFF->ON
 Select a sound from sound library for the PLC Control. You may configure three different ways to activate the [Trigger address]: (1) State change from OFF to ON (OFF->ON) (2) State change from ON to OFF (ON->OFF) (3) State change (either from ON->OFF or OFF->ON)
f. "Execute macro program"
Trigger address
PLC name : Local HMI Setting
Address : LB 🗸 0
Trigger mode : OFF->ON



٦

A	Activate the [Trigger address], the system will execute the Macro
	You may configure three different ways to activate the [Trigger address]:
(1) State change from OFF to ON (OFF->ON)
(2) State change from ON to OFF (ON->OFF)
(3) State change (either from ON->OFF or OFF->ON)
(4) Always active when ON
<u> </u>	
ł	n. "Screen hardcopy"
	Activate the [Trigger address], the system will have designated window printed out.
	You may configure three different ways to activate the [Trigger address]:
(1) State change from OFF to ON (OFF->ON)
(2) State change from ON to OFF (ON->OFF)
(3) State change (either from ON->OFF or OFF->ON)
	The designated window can be one of following three different
	ypes:
	Source window for print
	OCurrent base window OCurrent base window OCurrent base window no. from register OCurrent base window no.
	PLC name : Local HMI Setting
	Address : LW 0 16-bit Unsigned



[Window no. from register]

Print the window designated by a PLC device when the operation is activated, if [LW0] = 14, the window no.14 will be printed out.

[Designate window no.]

Select a base window to be printed out when the operation is activated.

Note

- 1. The system performs a *background printing process* 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

Overview

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	2
R.	E
æ	
	4

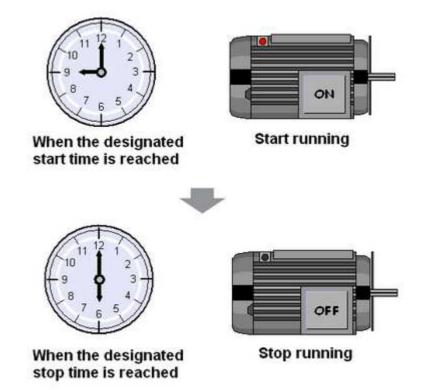
Scheduler	
New	Exit

Jeneral Time Set	Prohibit		
Description :	Scheduler 1		
Action mode	Power-ON st	art/end action	
inclusii incluc	💿 Bit ON	O Bit OFF	🚫 Word write
Action address			
PLC name :	Local HMI		Setting
Address :	LB	🖌 100	

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],

-Action mode				
	💿 Bit ON	🔘 Bit OFF	🔘 Word write	

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
⊙ Cor	istant	◯ Address

4. Unselect [Setting on individual day]. In [Start], adjust time as 8:00:00 and select Monday to Friday.

📃 Setting	on individu	al day				
tart						
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	ation action		
17 🗘	0	0	

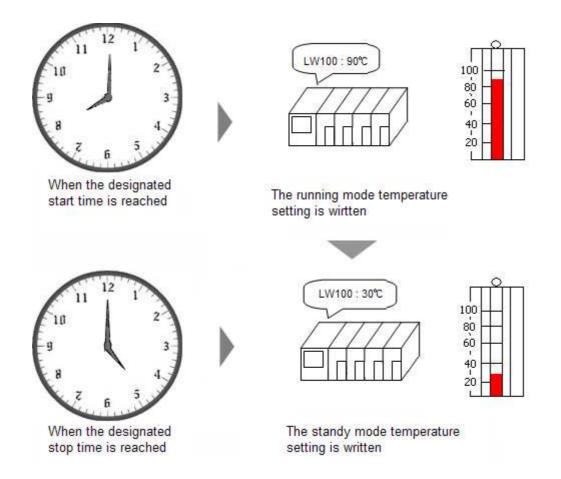
6. Click [OK], a new schedule object is created and display on the schedule list.





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.

	Time Set	Prohibit		
1	Description :	Scheduler 1		
Action	mode	🔲 Power-ON st	art/end action	
		🚫 Bit ON	O Bit OFF	💿 Word write
Action	address			
	PLC name :	Local HMI		Setting
	Address :	LW	✓ 100	16-bit Unsigned
Word	write value s	ettings		
		Constant	🔘 Address	
	Start value :	90		

[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],

⊂ Word write value settings
Onstant O Address
Write start value : 90

[Time Set tab]

5. Select [Time Set] tab, check [Constant]

General Time Set		Prohibit
⊙ Cor	istant	🔘 Address

6. Unselect [Setting on individual day]. In [Start], adjust time as 8:00:00 and select Monday to Friday.

art		a survey.				
8	\$	*	0	*		
Sun	🔽 Mon	🔽 Tue	🔽 Wen	💌 Thu	Fri	Sat

7. In [End], select [Enable termination action] and adjust time as 17:00:00.

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

	Time Set	Prohibit			
	Description :	Scheduler 1			
		Power-ON	start/end action		
Action	ı mode ——	🔿 Bit ON	◯ Bit OFF	💿 Word w	rite
Action	address				
	PLC name :	Local HMI		~	Setting
	Address :	LW	🖌 100		16-bit Unsigned

Setting	Description
Action Mode	Select the type of operation performed at designated time.



	At start time, turn ON the sp Example: Start time = 09:00 End time = 17:00	
	ON OFF 09:00:00	
	At start time, turn OFF the s Example: Start time = 09:00 End time = 17:00	
	ON OFF 09:00:00	
	At start time, the specific [W address. At end time, [Write address. Example: Device address =	Vrite start value] is written to the action e end value] is written to the action = LW100
	Start time = 09:00 End time = 12:00 Write start value = Write end value =	:00 = 10
	LW100 10	LW100 0
Action address	09:00:00	12:00:00 the scheduler performs actions 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				
	Π				
	Start action				
	Outside the scheduled range:				
	Power ON Start time				
	Termination action Start action				
	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.				
Word write	These settings are active only when Action Mode is set to [Word				
value Settings	Write].				
- and extingo					



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)

General Time Set Prohibit • Constant • Address Setting on individual day Start • O • O • O • O • O • O	heduler
Setting on individual day Start Sun Mon Tue Wen Thu Fri Sat End	eneral Time Set Prohibit
Start O O O O O O Sun Mon Tue Wen Thu Fri Sat End	⊙ Constant ○ Address
0 2 0 2 Sun Mon Tue Wen Thu Fri Sat	
End	
	🕼 Sun 🦳 Mon 💭 Tue 💭 Wen 💭 Thu 💭 Fri 💭 Sat
OK Cancel Help	

Setting	Description		
Constant/Addr	Select the method to set the start time and end time.		
ess	Constant		
	Specifies a fixed time and day.		
	Address		
	The start/end time is retrieved from the device address		
	at on 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.		



[]	
	Start actionTermination action09:0017:00
	Monday
	Tuesday
	NOTE 1. You must enter settings for the Start Time and End Time.
	You cannot set the Start Time and End Time to the exact same day and time.
	• Disable A schedule that is 1 day (Start and End times are within 24 hours) can be entered. Multiple Start and End days can be selected. You can perform actions at the same time on multiple days.
	To specify an End Time, you must select [Enable termination action]
	Start actionTermination action09:0017:00
	Monday
	Tuesday
	NOTE
	• You cannot set the Start Time and End Time to the
	exact same day and time.
	• The time scheduler is for one day only, so if the End



	Time is earlier than the Start Time, the operation of End Time will be performed on the next day.				
	(For example) Monday Tueso				
	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] is disabled, user can designate more than one day.				
End	Set the end time and day. When [Enable termination a can be specified. The day settings can on individual day] is enabled.	-			



■ 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.

eneral Time Set	Prohibit	
O Constant 🤇	⊙ Address	
Time setting addres	8	
PLC name :	Local HMI	Setting
Address :	LW 🗸 0	
	Control : 0	
	Status : 0 + 1	
A	ction mode : 0 + 2	
Start	t time (day) : 0 + 3	1
Start t	time (hour) : 0 + 4	1
Start tin	ae (minute) : 0 + 5	
Start tin	ne (second) : 0 + 6	
	time (day) : 0 + 7	
	time (hour) : 0 + 8	1
	ne (minute) : 0 + 9	
End tin	ne (second) : 0 + 10	

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		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.

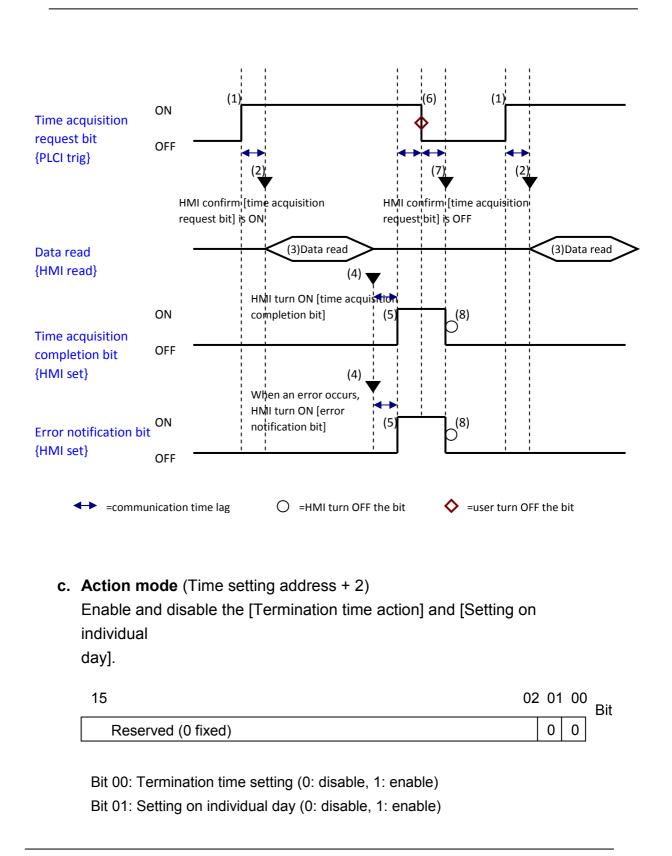
When the system competes the read operation, it will turn the [time acquisition complete bit] ON $(0\rightarrow 1)$. Also, if the read time data is incorrect, the [error notification bit] will be turned ON $(0\rightarrow 1)$.

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.





- **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

heduler				
General Time Set	Prohibit			
Prohibit				
🗹 Use prohibit	function			
PLC name :			*	Setting,
Address :	LB	✓ 0		
Sound	New Triber	www. Th		1
🔲 Enable	Sound Libre	<u>ay</u>]		
	Play			
	10			12
	01	K Cance	el	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	Security S	Shape Lab	el		
De	escription :					
Attribute	-					
	Mode :	Drop-down	List 🛛 💌	Backgro	ound : 📃	
				Sele	ction : 💻	
	Direction :	Down	~			
	Source	of item data :	Item addres	\$		*
Monitor	address					
P	LC name :	Local HMI			~	Setting
	Address :	1W	~	0		16-bit Unsigned
- Control a		()			12788	
	PLC name :	n de anne en com en la			~	Setting
		LW	~	0	· · · · · · · · · · · · · · · · · · ·	Setting
	PLC name :	n de anne en com en la		item data		Setting
	PLC name : Address :	LW [Address] : se		item data		Setting
Ρ	PLC name : Address : ress	LW [Address] : se	1 : item count	item data	ach item : 5	Setting
P Item addu O ASC	PLC name : Address : ress II (LW [Address] : se [Address] + :	1 : item count	item data	ach item : 5	

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]



	Coloct bookground color for the chiest			
	Select background color for the object.			
	[Selection] Select background color for the selected/highlighted 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.			
	06/08/09 - MULTIN WIRTH CANNER THE AND A MANAGEMENT OF A CONTRACT OF A C			
	13:14:18 13:14:38 13:14:58 13:15:18 13:15:38 13:15:58 13:16:18			
	[Туре]			
	Alarm (Event) log is used to display Historical Event-Display.			
	Data sampling is used to display Historical Trend-Display or			
	Data-Display.			



ed when
e one
en
nable ate.
the
dress for
ontent of
ata in this
ta in this



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].
Note : The system will automatically disable Mapping table when Item address mode is selected.

Mapping tab

tion	list Mapp	ing Security Shape	Label	
T	tem	Value	Item data	
0	0.311	0	test1	* *
1		1	test2	
2		2	test3	-
3		3	test4	
4		4	test5	
5	(error)		?	
				Set default
Ino	r notificatio			
Ino	r notificatio	n I Enable	Set ON	Set default
Ino			Set ON	

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 represents a state that will be displayed in the list. This field is read-only.
	[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



	 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). b. [For writing]: The system writes this value to [Monitor
	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]
	 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.
	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.

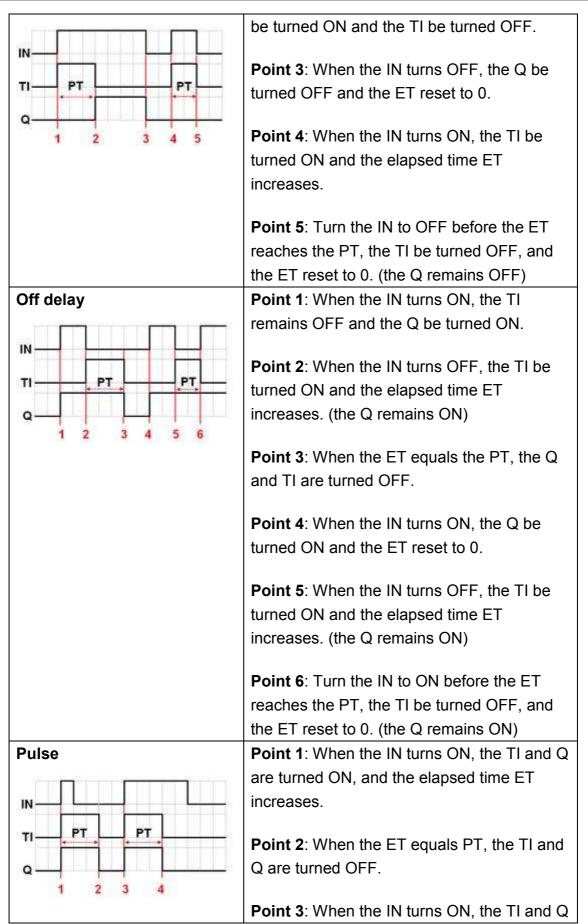
Timer Variable	Variables Type	Description
Input bit (IN)	Bit type	The master switch of timer.
Measurement bit	Bit type	Turn ON when the timer begin
(TI)		counting.
Output bit (Q)	Bit type	Activate when the timer finish
		counting.
Preset time (PT)	Word type	Set the timer value.
Elapsed time	Word type	Display current elapsed value
(ET)		of timer.
Reset bit (R)	Bit type	Reset the elapsed time (ET) to
		0.

Configuration

Click the "Timer" icon ⁽¹⁾, "Timer object properties" dialogue box appears as follows:

ner		
Description :		
Mode :	Accumulated OFF delay 🛛 🗸 T	ime base : 0.1 second (s) 🛛 🐱
-		
	Прт	РТ
	╹┼╎┥─₽╹	
	R	
	<u>السا</u> الد	
Input bit (IN)		
PLC name :	Local HMI	Setting
Address :	LB 🔽 0	
Measurement bit (1)	
PLC name :	Local HMI	Setting
Address :	LB 🗸 1	
Output bit (Q)	i dida	
PLC name :	Local HMI	Setting
Address :	LB 🔽 2	
Preset time (PT)		
PLC name :	Local HMI	Setting
Address :	LW 🗸 O	16-bit Unsigned
Elapsed time (ET)		
	🖌 Enable	
PLC name :	Local HMI	Setting
Address :	LW 😼 1	16-bit Unsigned
Reset bit (R)		
PLC name :	Local HMI	Setting
Address :	LB 😼 3	

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







	are turned ON, and the elapsed time ET
	increases.
	Point 4 : When the ET equals the PT, the TI
	and Q are turned OFF.
Accumulated On delay	Point 1: When the IN turns ON, the TI be
	turned ON and the elapsed time ET
	increases. (the Q remains OFF)
IN	
TI-PT-PT-PT-PT-	Point 2: When the IN turns OFF, and if the
R	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	
	Point 3: 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.
	Doint 4: When the ET reaches the DT the TL
	Point 4 : When the ET reaches the PT, the TI
	be turned OFF and the Q be turned ON.
	Point 5 : When the IN turns OFF, the Q be
	turned OFF. (Reset the ET to 0 by using
	Reset bit (R).)
Accumulated Off delay	Point 1 : When the IN turns ON, the Q be
	turned ON and TI remains OFF.
	Point 2 : When the IN turne OFF, the TL he
	Point 2 : When the IN turns OFF, the TI be
R	turned ON and the elapsed time ET
Q┦ <u>↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ </u>	increases. (the Q remains ON)
1 2 3 4 5 6 7 8 9 10	
	Point 3 : When the IN turns ON, the timer
	measurement pauses.
	Doint 4: When the IN turne OFF the record
	Point 4 : When the IN turns OFF, the paused
	timer measurement continues.
	Deint 5. When the ET equals the DT the T
	Point 5 : 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.

chieror	Profile					
	Description :					
				Encode for	nat: N	rsc 💌
Capta	ure address —				ा	
V 1	Use capture fu	unction				
	PLC name :	Local HMI			-	Setting
	Address :	LB	✓ 0			
Sto	rage medium		~~~~~			
(🔵 SD	💿 USB 1	O USB 2	1		
Rec	ord time					
	Before	: 5 📚 sec	conds	After: 5	\$	seconds
-						
Cont	rol address					
	rol address Jæ control fu	nction				
		nction				
					~ (Setting

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. $I = 0 \rightarrow Step Playing$
	$[LW100] = 0 \rightarrow \text{Stop Playing.}$
	[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.

		Objects
	Video In Object's Properties	
	General Profile	
	Description :	
	Input channel : 🚺 💉 Encode format : PAL 💌	
	Capture address 12	
	Use capture function	
	Control address	
	Use control function	
	Definition: Conturn the image of the imput video	
Use	Definition: Capture the image of the input video.	
Capture	Illustration:	
Function	A. [Capture address] the Control Address that trigge	ers system
	to capture the image of video.	
	B. [Storage medium] To choose where to save the v	
	image. Available storage: SD card, USB1 or USB2	•
	 VIP 1 video image will be saved in file VIP 1 ir 	n the
	chosen storage and VIP 2 video image in file V	VIP2.
	C. [Record time] To set a period of time for image ca	apturing.
	- The longest period can be set starts from 10 s	econds
	before triggering [Capture address] to 10 sec	conds
	after triggering. In this case there will be 21 im	ages
	captured, including the one captured at the trig	gering
	moment.	
	- The time interval for capturing is once in each	second.
	- The captured .jpg file will be named in the follo	
	format:	
	Before or after [Capture address] is triggered:	
	YYYYMMDDhhmmss.jpg	
	The moment that[Capture address] is triggere	d:
	YYYYMMDDhhmmss@.jpg	

-	E.		
PLC name :	Local HMI		Settin
Address :	LB	✓ 0	
Storage medium			
🔘 SD	💽 USB 1	OUSB 2	
Before	: 5 🔅 sec	onds Af	ter: 5 🔝 second
			ter : 5 📚 second
ake the illusti	ration above	as sample, s	
ake the illusti Before" and "/	ration above After" to "5" :	as sample, sesconds, whe	et [Record time
ke the illusti efore" and "/ anges from	ration above After" to "5" s OFF to ON,	e as sample, so seconds, whe system will be	et [Record time] n [Capture addi

Note:

WEINTEK

- 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

Use this utility to	o edit messages	that displays in	popup message boxes
---------------------	-----------------	------------------	---------------------

ystem Message	
Confirmation require	d
Message :	Please confirm the operation
	Use label library
ок ; [ок
	Use label library
Cancel :	Cancel
	Use label library
Font : [Arial
Deny write-comman	d
Message :	The system is being prohibited from writing device registers!
Font :	Arial
	Use label library
Allow write-comman	id
Message :	The system is now allowed to write device registers.
Font :	Arial
	Use label library
	OK Cancel

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.
	System Message Confirmation required Message : Please confirm the operation Use label library OK : OK OK Cancel : Cancel Use label library Font : Times New Roman
Deny write-command Allow	Display when system tag LB-9196 (local HMI supports monitor function only) is turned ON. Display when system tag LB-9196 (local HMI supports monitor function
write-command	only) is turned OFF.



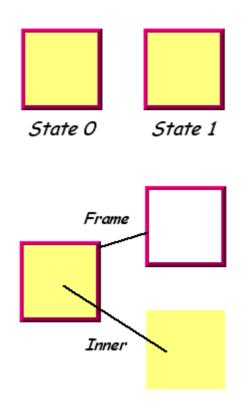
Chapter 14 Shape Library and Picture Library

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:



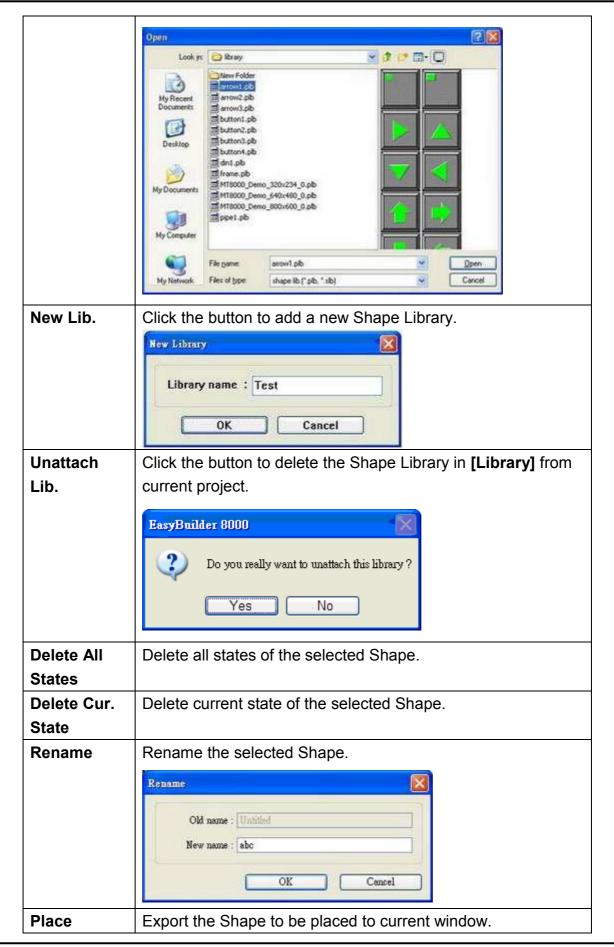
The frame and inner of a shape can be used separately or together by an object. Click **[Call up Shape Library]**, and the **[Shape Library]** dialogue appears as below:





Shape Library					
Library :	Sta	ite : O	• 012	3456(7
Frame arrow1 arrow2 arrow3 <u>button1</u> button2 button3					Select Lib New Lib Unattach Lib.
button4 System Frame System Button System Lamp	0: Untitled States : 3 Frame	1: Untitled States : 3 Frame	2: States : 3 Frame	3: States : 5 Frame	Delete All States Delete Cur. State
					Rename Place
State 0 ~ State 11 :	4: Untitled	5: Untitled	6: Untitled	7: Untitled	
	States : 3	States : 3	States : 3	States : 3	
	Frame	Frame	Frame	Frame Display	
				🗹 Inner 🗹 Fram	Close

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.



WE!NTEK



•			Reni	OM mane : [Untile]
Shape Library				New name : obc
Library : buttoni	State : [C REAL PROVIDE T
System Finane System Detton System Lemp				New Lab
	Stoles : 2 St	Shape1 2 Shape2 des 2 States 2 ner Inner	3: Shape3 States: 2 Inner	Delete All States
	Long Con			

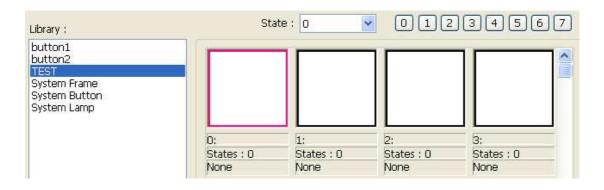
The following shows how to create a new Shape Library and add a Shape with two states to it.

Step 1

Click **[New Lib.]** and input the name of the new Shape Library.

New Library	X
Library name:	TEST
ОК	Cancel

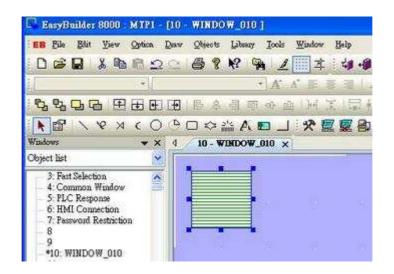
A new Shape Library "TEST" will be added to the **[Shape Library]** dialogue. At this moment, no Shape is in the library.





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.

+)) 2	Shape librar	y TEST
	Descriptio	n Untitled
	Shape no	o.: 0 🔽 State : 0 🤙
		Width: 77 Height: 73 Frame : undefined Inner : undefined
		miler . differinge

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.

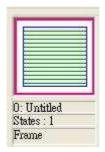


State	Select the state of the Shape which this graph represents. In this case the state is set "0". EB8000 provides 256 states for each Shape.
Frame	If [Frame] is selected, the graph will become a frame of the Shape.
Inner	If [Inner] is selected, the graph will become an inner part of the Shape.

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.



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]**:



Shape Horary	🖅 TEST 🛛 💌
Description	ı : Untitled
Shape no	.: 0 🐱 State : 1 💌
	Width:77 Height:73 Frame: undefined
	Frame 1 underined

A complete Shape with two states is created. See the following picture.

batton1 batton2	
HEST System Frame System Botton System Lamp	0. Untitled States : 2 Frame
State 0 - State 11 :	4: States : 0 None



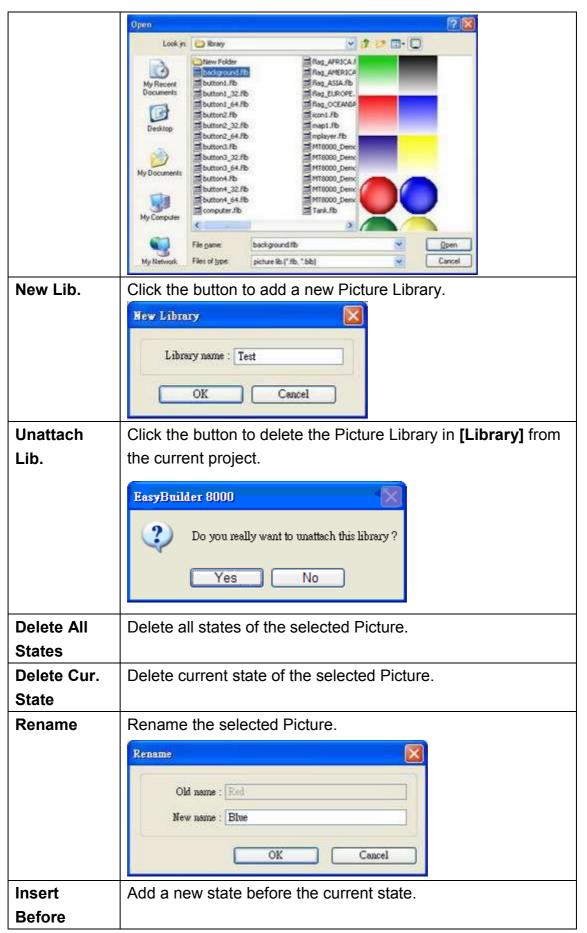
14.2 Creating Picture Library

Click the **[Call up Picture Library]** button in toolbar, and the **[Picture Library]** dialogue appears.

) 🛃 🖪 🖡	3 🕽 📢 🕒	1	
	Call up	Picture Library		
Picture Library				<u> </u>
Library :		State : 🚺		12345
icon1 Button2				
Demo graph0				
graph1				
map1 Computer				
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
IIII	BMP	BMP	BMP	BMP
Select Lib New Lib	Unattach Lib.	Rename	Export	
Import Picture Modify Picture]	Delete All States	Delete Cur. Stat	te 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.



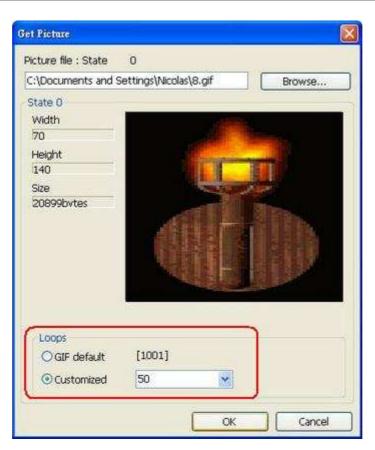




Insert After	Add a new state after the current state.				
Import Picture	Add a new picture to the Picture Library.				
	Import Picture Picture Picture no.: 0 Picture name: 123 Next Cancel				
Modify Picture	Modify the selected picture.				
Export	Export the selected picture to the appointed place. As shown below, users can get the original picture.				
	File name: Save My Network Save as type: Bitmap file.(*.bmp) Save				

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.

ew Library	
Library name:	TEST
ОК	Cancel

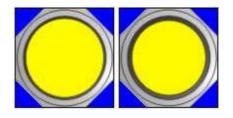
A new Picture Library "TEST" will be added to the **[Picture Library]** dialogue. At this moment, there is no Picture in the library.



	· · · · · ·		-10	
	1		2:	3:
Graph name :	0:			
Graph name : Total states :	0:	1:	0	0

Step 2

Prepare the pictures to be added; suppose the two graphs below are used to represent state 0 and state 1 respectively.

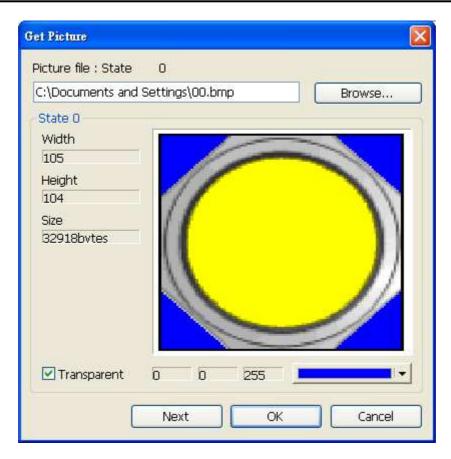


Click [Import Picture] and a dialogue appears as below. Set [Picture no.] and [Picture name] for it, and then click [Next].

Picture no. :	0 🖌
Picture name :	F YELLOW
Picture name :	FYELLOW

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.



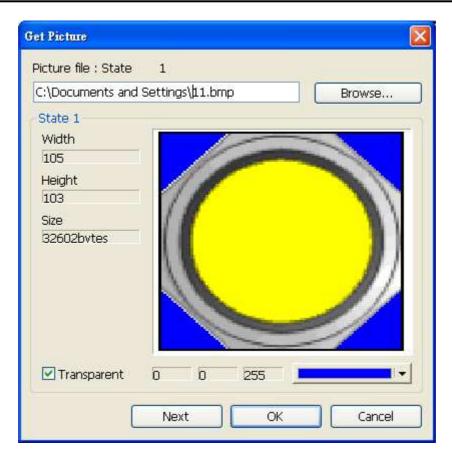
Before choosing transparent color, check **[Transparent]** box first and then left click on location-to-be of the graph. At this time, EB8000 will automatically display RGB value of the transparent color. Take above as an example, the actual shape shown as below:



Step 4

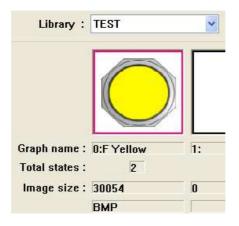
WEINTEK

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.

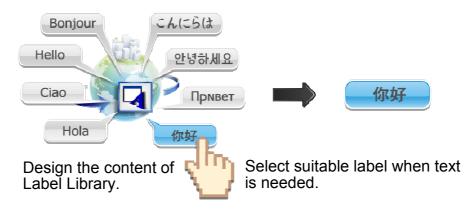


WEINTEK

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.

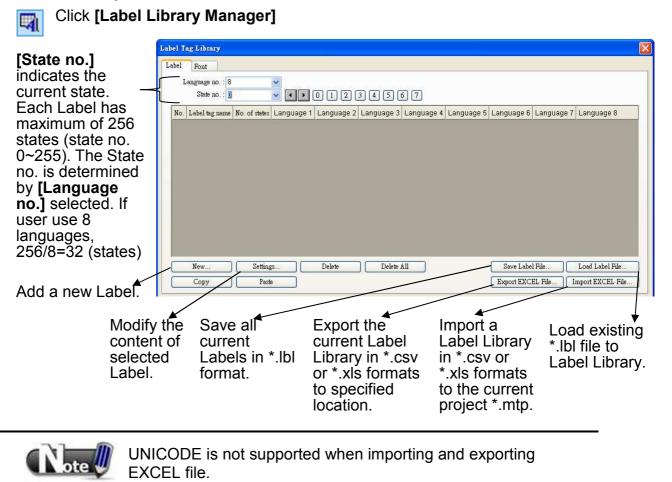


Chapter 15 Label Library and Multi-Language Usage



15.1 Introduction

The system in operation will display the corresponding text to the language in use according to the settings. EasyBuilder Pro supports 8 different languages simultaneously.



the Label.

15.2 Building Label Library



Label [Label name] User can specify the name of Label name : Pump Alarm [No. of states] No. of states : 2 The number of states can be shown by this Label. OK Cancel

2. Click [OK] a new Label "Pump Alarm" with 2 states will be added to the Label Library, select it and click [Settings].

el L	Font	3	~			- 27 - 22				
	State no. : ()	✓	012(3 4 5 (6 7				
lo,	Label tag name	No. of states	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7	Language 8
	Pump Alarm	2	()		0.				С.	
	New)Setting	ş)	Delete) Delete 1	<u>All</u>		Save Labe	1 File)	Load Label File.

3. Set up the corresponding language contents.

l Content Settings	Alarm	State no. : 0 💌 🖣	• 0 1 2 3 4	56
anguage 1	Language 2	Language 3	Language 4	
How are you?	你好			~
	×	1	~	
<	2	2	> <	2
⊿anguage 5	Language 6	Language 7	Language 8	
	2	~		0
	×	2	2	~
<	5 5	5 4	2	2



15.3 Setting Label Font

[Label Tag Library] / **[Font]** see the languages the current Label contains and set the font. Different languages can use different font.

Label Tag Library			
Label Font			
Font		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] When using a Label to show different languages, different fonts can be selected for each language.

[Comment] The memo for each font.

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.

General Security Shape Label Use label library Label tag : Use bitmap font Language : 1 State : 0 ♥ ♥ ♥ 0 1 Attribute Font : Arial Black Color : I talic Underline Duplicate these attributes to Every state Every language All Movement Direction : No movement ♥ Trucking	ew Bit Lamp Object	
✓ Use label library Label tag :: □ump Alarm □ Use bitmap font Label 0 □ Use bitmap font □ump Alarm Language :: 1 Attribute Font : Arial Black ✓ Color :: ✓ Italic Underline Duplicate these attributes to Every state Every state Every language Movement Direction :: No movement ✓	General Security Shape Label	
State : 0 • • • 0 1 Attribute Font : Arial Black Color : • · · Size : 16 • · · Align : Left • Blink : None • · Italic • Underline Duplicate these attributes to Every state • Every language • All Movement Direction : No movement • ·	₩ Use label library La	Label O
Color: Size : 16 Align: Left Blink : None Italic Underline Duplicate these attributes to Every state Every language All Movement Direction : No movement How are You	State : 0 🔽 📢	• 0 1
Align : Left Blink : None Italic Underline Duplicate these attributes to Every state Every language All Movement Direction : No movement How are You	Font : Arial Black	×
Italic Underline Duplicate these attributes to Every state Every language All Movement Direction : No movement Content : How are You	Color:	- Size : 16
Duplicate these attributes to Every state Every language All Movement Direction : No movement Content : How are You	Align : Left	Slink : None
Direction : No movement	Duplicate these attrib	utes to
How are You	11.12.0000000	*
3	Content :	
	How are You	
	Tracking	Duplicate this label to every state

When **[Use label library]** is selected, **[Content]** field shows the content of selected Label Tag and the settings of **[Font]** are also included in the Label Library. Please note that languages 2 ~ 8 can only set the Font **[Size]**, other settings for example: **[Color]**, **[Align]**, **[Blink]** etc. will follow the settings of language 1.

15.5 Settings of Multi-Language (System Register LW-9134)

When users would like to have the object's text to show multi-language, except for using Label Library, the system reserved register [LW-9134]: language mode can be used. The value of [LW-9134] can be set from 0 to 7. Different data of [LW-9134] corresponds to different Languages1 to 8. The way of using [LW-9134] will differ if the languages are not all chosen when compiling and downloading the project.

lf user defines 5 languages	When compiling if only languages 1, 3, 5 are chosen	For [LW-9134] the corresponding values are:
1 English	<u> </u>	→ 0
2 Traditional Chinese		
3 Simplified Chinese		→ 1
4 French		
5 Korean		→ 2
Use bitmap font Language : 1 Attribute Font : Arial Color : Align : Left	Blink : D Description	ntry Numeric Format Security Shape Font
Movement Direction : No movement	Every language PLC nan	ne : Local HMI Setting ss : LW-9134 (16bit) : language mode V
Content : Enstish	PLC nam	ne ; Local HMI 🔮
LITERST.		ss : LW9134 System tag
8	Address form	at : DDDDD [range : 0 ~ 10500]

When compiling, tick the defined and needed languages.

Compiling					X
Project name :	C:\EB8000\pr	oject\MTP1.mtp			
XOB file name :	C:\EB8000\pr	oject\MTP1.xob			
XOB password :	Set) (used in decompiler)	Decompilation	is prohibited	
Select the languages		MI Jage after redownloadin	ng the project : Language b	L 💌	
🗹 Languag	je 1	Language 2	Language 3	🗹 Language 4	
🗹 Langua	je 5	🗹 Language 6	🗹 Language 7	🗹 Language 8	

The simulation is shown below, if we change the value of [LW-9134], the content of the "Text" object will be changed.



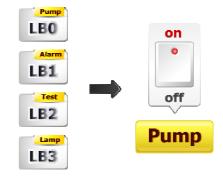
Please confirm your internet connection before downloading the demo project.

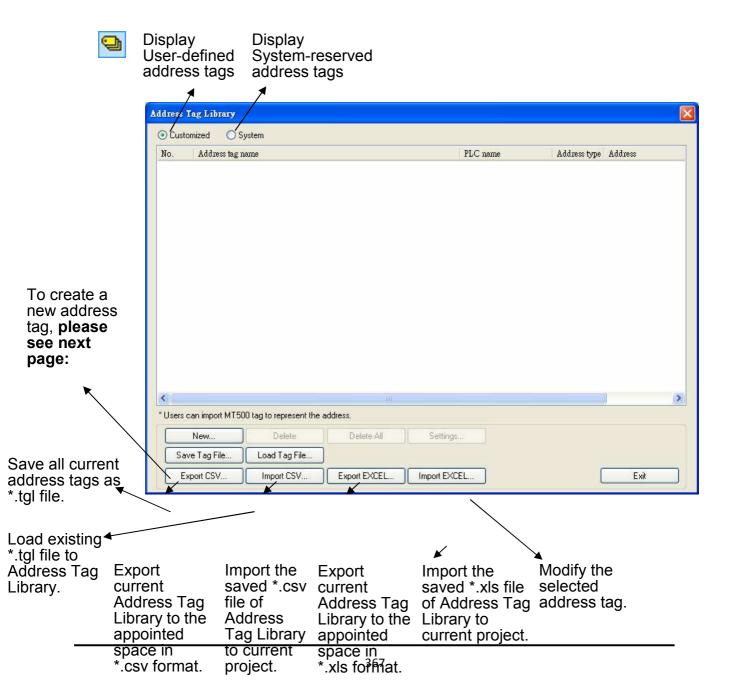


Chapter 16 Address Tag Library

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 [New]

Name of the address tag	Address Tag	
Selected from [Device list]	Tag name : pump	
The type of address; [bit] / [word] types available	PLC name : MITSUBISHI FX0n/FX2 Address type : ③ Bit ③ Word Device type : T	*
The device type; the available types are related to [PLC name] & [Address type]	Address : 20 Address format : DDD [range : 0 ~ 255] OK Cancel	
Content of the Kaddress		

Click [OK]

A new tag will be found in the **[Customized]** library.

No.	Address tag name			Address type	Address	Read/Write
1	pump	MITSUB	BISHI FX0n/FX2	Bit	T-20	Read/Write
<						
	can import MT500 tag to represe	it the address.				





16.2 Using Address Tag Library

General Security Description	Shape Label	
Read address -		
PLC name Address		
ldress		
PLC name :	MITSUBISHI FX0n/FX2	~
Device type :	pump	~
Address :	T-20 User-defined to	ag)
Address format :	DDD [range : 0 ~ 255]	

- 1. Define Address Tag Library
- 2. Create an object, select [General] / [PLC name]
- 3. Click [Setting]
- 4. Tick [User-defined tag]
- 5. From **[Device type]** select the defined tag.
- Upon completion, the window tree will show the address tag name used for the object.

Windows ×
Object list 💌
3: Fast Selection
5: PLC Response
6: HMI Connection 7: Password Restriction
9 *10: WINDOW 010
BL_0 (pump : T20) (Bottom Layer)
11
13
- 14

Chapter 17 Transferring Recipe Data

Recipe Data refers to data stored at RW and RW_A addresses. The way of reading and writing Recipe Data is nothing different from operating a word register. The difference is that Recipe Data is stored in flash memory, when restarting HMI, the latest data records in RW and RW_A are kept the same.

The size of Recipe Data in RW is 512K words, and RW_A is 64K words. Users can update Recipe Data with SD Card, USB disk, USB cable or Ethernet and use this data to update data in PLC. Recipe Data can also be uploaded to the designated address; furthermore, PLC data can be saved in recipe memory. The following explains the ways of operating Recipe Data.





17.1 Updating Recipe Data with Ethernet or USB cable

- 1. Open Project Manager and click [Download].
- 2. Select [RW] and [RW_A] and designate the directory of the source file.
- 3. After downloading, restart HMI, RW and RW_A will be updated.

	Download			
Select the source file	Firmware	MT6000/8000 i Series	~	
directory.	I RW I RW_A □ Data log	C:\EB8000\recipe\recipe.rcp C:\EB8000\recipe\recipe_a.rcp		Browse Browse
Tick [Reboot HMI	Install X-series media	-player drivers		
after download]	error : C:\EB8000\recipe\n scan font : finished reset HMI : finished	ecipe_a.rcp not found		
instead of rebooting	finished Connection ⓒ Ethernet	O USB cable (i series o	nly)	×
HMI manually.	 IP Name <u>H</u>MI Name: + OS 20091002 or later ✓ Reboot HMI after dow □ Reset data log □ Download 	nicolas_8100 ▼ 192.168 Search 192.168 192.168 192.168 192.168 192.168 192.168 192.168 192.168 192.168 192.168 192.168 192.168	1.111 (Default HMI) 1.112 (tony_8150x) 1.115 (nicolas_8050) 1.118 (nicolas_8050) 1.118 (ricolas_8100) 1.15 (tony) 1.20 (q21) 1.208 (kewin_MT8070iH) Reset event lo	₽g Exit

When [Reset recipe] is selected, before start downloading, EasyBuilder Pro

will set all the data of [RW] and [RW_A] to "0" first.



17.2 Updating Recipe Data with CF/SD Card or USB Disk

1. Open Project Manager and click [Build Download Data for CF/SD Card or

USB Disk].

2. Insert SD card or USB disk into PC

3. Click [Browse] to designate the file path.

4. Click **[Build]**, EasyBuilder Pro will then build the sources into SD card or USB disk.

G:\test	Browse
Gources	
Recipe (RW)	
C:\EB8000\HMI_memory\recipe\recipe.rcp	Browse
Recipe A (RW_A)	
C:\EB8000\HMI_memory\recipe\recipe_a.rcp	Browse
Data log	
Build	Exit

Select the source file directory.

When download data is successfully built, two folders can be

found: *history* and *mt8000. mt8000* is for storing project files;

history is for storing Recipe Data and Data Sampling / Event Log records.

17.3 Transferring Recipe Data

Use **[Data Transfer (Trigger-based) object]** to transfer Recipe Data to the appointed address, or save the data of the designated address to [RW] and [RW_A].

New	v Data Transfer (Trigger-based) Object	
Gen	neral Security Shape Label	
The starting address of the data is to be transferred from.	Description : Source address PLC name : MODBUS RTU	Setting
The starting address of the	Address : 3x 1	
data is to be transferred to.	PLC name : Local HMI	Setting
	Address : RW V 100	
The number of words of	Attribute	
the data from Source to Destination.	No. of word : 120 Mode : Touch trigger	
90 i +	OK Cancel	Help

17.4 Saving Recipe Data Automatically

In order to prolong HMI flash memory life span, EasyBuilder Pro will save Recipe Data automatically **every minute** to HMI. To avoid losing data when switching HMI off during the interval between saving operations, system register [LB-9029: Save all recipe data to machine (set ON)] is provided. Sending ON signal to [LB-9029] will make EasyBuilder Pro save Recipe Data once. Sending ON signal to [LB-9028: Reset all recipe data (set ON)], EasyBuilder Pro will clear all Recipe Data 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

- 1. 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.

2. Open the macro editor; you'll see the line numbers displayed on the left-hand side of the edit area.

WEINTEK	Macro Reference
Macro Macro list : [ID: 2] 10000 [ID: 3] 5000	
WorkSpace Macro ID: 4	Macro name : macro_4
	edit area
Click the right mouse button to display edit menu. GET/SET FN Compile	Exit Help

3. Right click on the edit area to open the pop-up menu as shown below:

Undo	Ctd+Z
<u>R</u> edo	Ctd+¥
Cut	Ctrl+X
Copy.	Ctrl+C
Paste	Ctil+∛
Select <u>A</u> ll	Ctrl+A
Toggle <u>B</u> ookmark	Ctrl+F2
<u>N</u> ext Bookmark	F2
Pre <u>v</u> ious Bookmark	Shft+F2
Clear All Bookmarks	
Toggle All Outlining	
Update All Outlining	

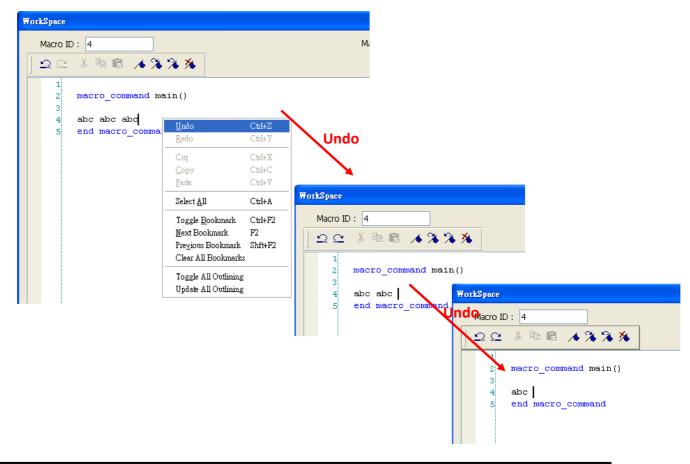
The disabled items are colored grey, which indicates that it is not possible to use that function in the current status of the editor. For example, you should mark a selected area to enable the copy function, otherwise it will be disabled.

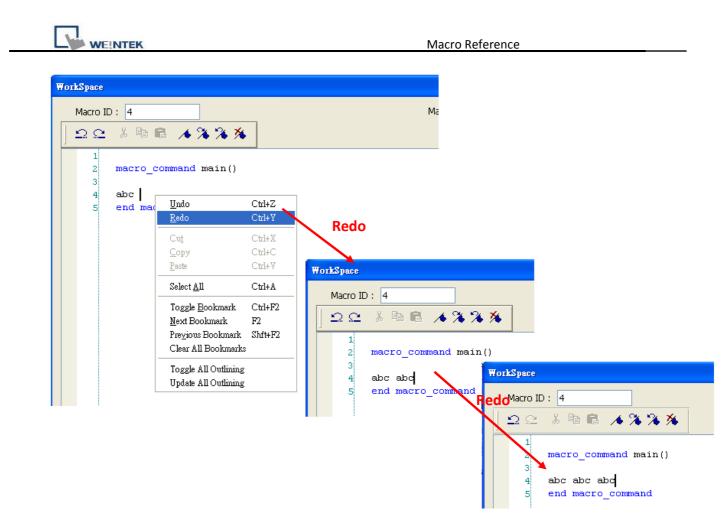
Accelerators are supported as described in the menu.

 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.

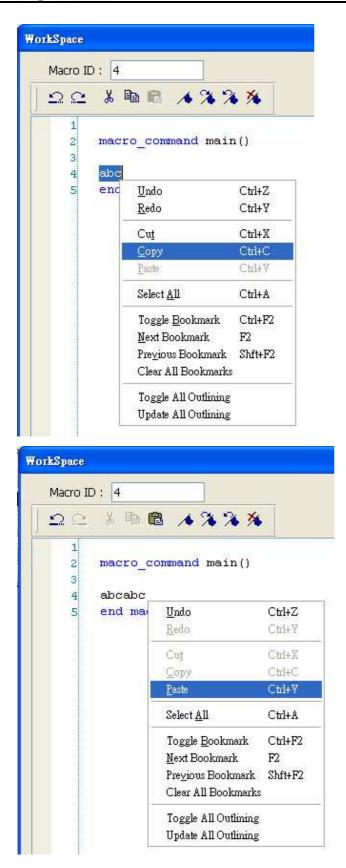
] 2 2 % **B 6** 🔺 🛪 🛪 🎋

5. 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).

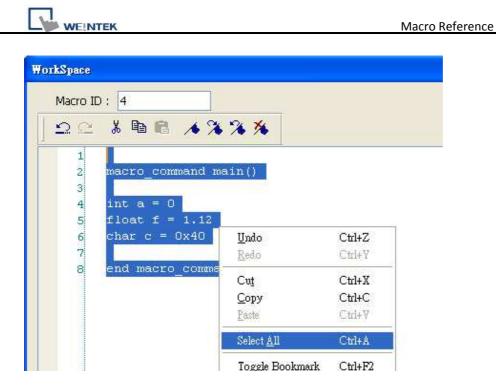




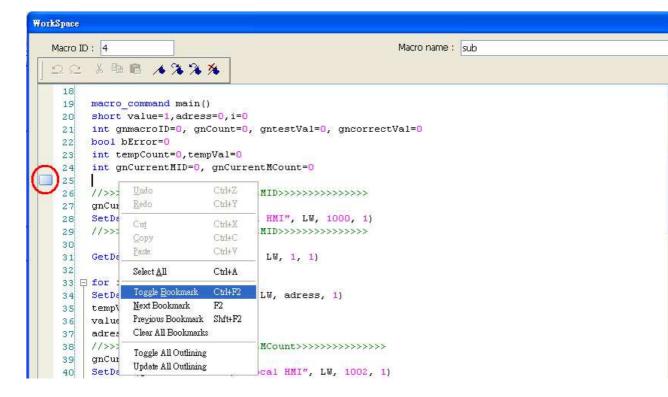
6. Select a word in the editor to enable the cut and copy function. After cut or copy is performed, the paste function is enabled.



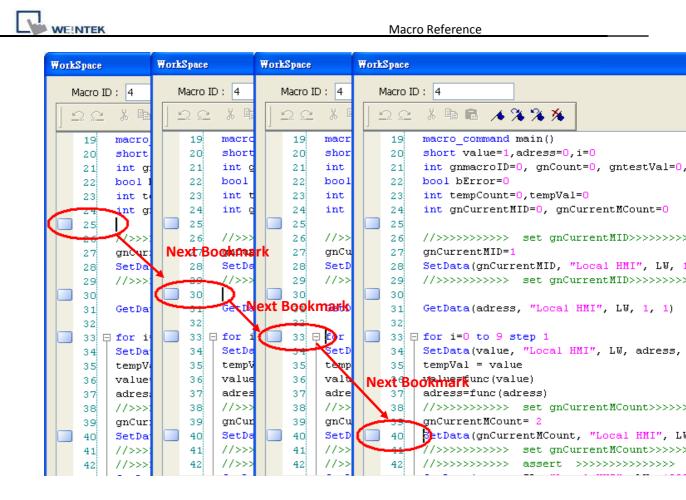
7. Use "Select All" to include all the content in the edit area.



- 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.



- b. If there's already a bookmark where the cursor is placed, select "Toggle
 Bookmark" to close it, otherwise to open it.
- c. 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.



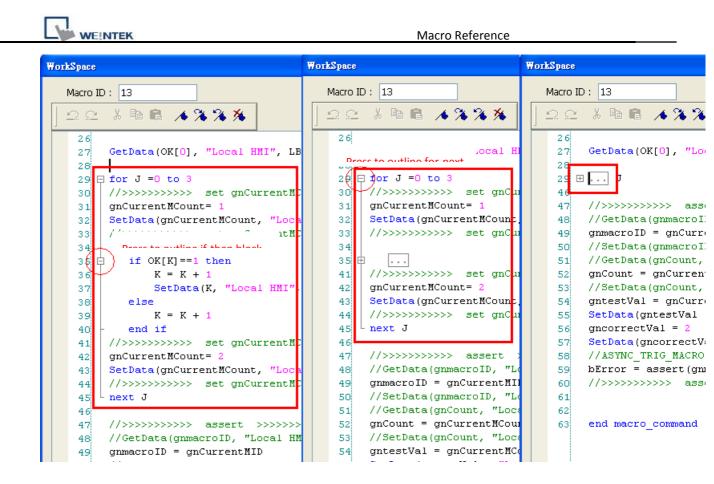
d. Selecting "Clear All Bookmarks" will close all bookmarks.

9. 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 🗄 📖 icon. There will be a tree diagram on the left side of edit area. Users

can click \Box to hide the block or \boxdot to open as shown below:



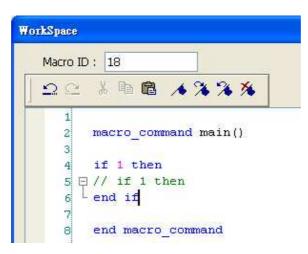
10. Right click to select "Toggle All Outlining" to open all macro code blocks.

VorkSpace			WorkSpace
Macro ID : 13			Macro ID: 13
<u> X</u> ħ	B 🔺 🕉 🕉	*	<u>⊃</u> ⊆ % B B A % % % }
26 27 GetDat 28 29 ⊞]	ca(OK[O], "Loc Undo	al HMI", LB,	28 29 ₽ for J = 0 to 3
45 47 //>>>	<u>R</u> edo	Ctrl+Y >>>>	<pre>30 //>>>>> set gnCurrentMCount>>>> 31 gnCurrentMCount= 1</pre>
48 //Get	Cut	Ctrl+X HMI	
49 gnmac	Сору	Ctrl+C	33 //>>>>>> set gnCurrentMCount>>>>
50 //Set	<u>P</u> aste	Ctrl+V HMI	
51 //Get 52 gnCou	Select All	Ctrl+A	LW, $35 \ominus$ if $OK[K] ==1$ then 36 $K = K + 1$
53 //Set	Select <u>N</u> II	I",	
54 gntes	Toggle <u>B</u> ookmark	Ctrl+F2 / gn(our 38 else
55 SetDa	<u>N</u> ext Bookmark	F2 MI"	LI 39 K = K + 1
56 gncor	Pre⊻ious Bookmark	Shft+F2	40 - end if
57 SetDa	Clear All Bookmark	s 1 HI	<pre>[I", 41 //>>>>>>> set gnCurrentMCount>>>></pre>
58 //ASY	Toggle All Outlinin;	~	42 gnCurrentMCount= 2
59 bErro	Update All Outlinin;	mCO1	nt, 43 SetData(gnCurrentMCount, "Local HMI", 1
60 //>>>	o puale All Outhini;	\$>>>:	>>> 44 //>>>>>>> set gnCurrentMCount>>>>
			45 - next J



11. Sometimes the outlining might be incorrect since that the keywords are

misjudged. For example:



To solve this problem, right click to select "Update All Outlining" to retrieve correct

outlining.



12. 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 End Sub	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
declaration
```

// local variable

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	'a', 'data', 'name'
	single 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	Description	Example
Operators		
<	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	if A >= 10 then B = 5
	equal to	
==	Equal to	if A == 10 then B = 5
<>	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	if A xor 256 then B = 5
	OR	
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	A = B << 8
	the left a specified number	
	of positions	
>>	Shifts the bits in a bitset to	A = B >> 8
	the right a specified number	
	of positions	

Bitwise Operators	Description	Example
&	Bitwise AND	A = B & 0xf
1	Bitwise OR	A = B C
۸	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] where define a variable B as a one-dimensional array of size 10

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
11	
<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:

<u> </u>	
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>
[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

WEINTEK

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
	end macro_command



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
	COT(source, result)// result is 0.5774
	end macro_command



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



Macro Reference

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

Name	RAND
Syntax	RAND(result)
Description	Calculates a random integer saved into result.
	Result must be a variable.
Example	macro_command main()
	short result
	RAND (result)// result is not a fixed value when executes macro every time end macro_command

Γ

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

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>
	<pre>// the length of the string (result1) is 4 bytes(= 1 * 4)</pre>
	DEC2ASCII(source, result2[0], 4)
	<pre>// result2[0] is '5', result2[1] is '6', result2[2] is '7', result2[3] is '8'</pre>
	<pre>// the length of the string (result2) is 8 bytes(= 2 * 4)</pre>
	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
	<pre>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</pre>

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
	0.400
	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
	GETBIT(9, result, 3)// result is 1
	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
	end macro_command



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
	end macro_command

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 Model General System Setting Security 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 Docal Free Protocol COM1 (19200, E, 8, 1) RS232
	Device Properties
	Name : MODBUS RTU Device
	Location : Local Settings
	PLC type : Free Protocol
	V.1.00, FREE_PROTOCOL.so
	PLC I/F : RS-232
	COM : COM1 (19200,E,8,1)
	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")
	Below is an example of executing an action of writing single coil (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])
<pre>// send out a "Write Single Coil" command OUTPORT(command[0], "MODBUS RTU Device", 8)</pre>
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[2]) LOBYTE(address, command[2]) ILOBYTE(address, command[2]) LOBYTE(read_no, command[4]) LOBYTE(read_no, command[5]) CRC(command[0], checksum, 6) LOBYTE(checksum, command[6]) HIBYTE(checksum, command[6]) HIBYTE(checksum, command[7]) // send out a 'Read Holding Registers' command OUTPORT(command[0], "MODBUS RTU Device", 8) // read responses for a 'Read Holding Registers' command INPORT(response[0], "MODBUS RTU Device", 9, returm_value) if returm_value > 0 then read_data[0] = response[4] + (response[3] << 8)// data in 4x_1 read_data[0] = response[6] + (response[5] << 8)// data in 4x_2 SetData(read_data[0], "Local HMI", LW, 100, 2) and if	
<pre>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[6]) HIBYTE(checksum, command[7]) // send out a 'Read Holding Registers" command 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)</pre>	
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 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	
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 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	HIBYTE(address, command[2])
LOBYTE(checksum, command[6]) HIBYTE(checksum, command[7]) // send out a 'Read Holding Registers" command 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)	HIBYTE(read_no, command[4])
HIBYTE(checksum, command[7]) // send out a 'Read Holding Registers" command 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	CRC(command[0], checksum, 6)
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)	· · · ·
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)	
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)	
	read_data[0] = response[4] + (response[3] << 8)// data in 4x_1
	SetData(read_data[0], "Local HMI", LW, 100, 2) end if
end macro_command	end macro_command

Name	GetData
Syntax	GetData(read_data[start], device_name, device_type, address_offset,
	data_count)



	or GetData(read_data, device_name, device_type, address_offset, 1)
Description	Receives data from the PLC. Data is stored into read_data[start]~ read_data[start + data_count - 1]. 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 :
	No. Name Location Device type Interf I/F 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.



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.

	V.1.10, FATEK_FB.so		
PLC I/F	: RS-232 👻	PLC default statio	n no.: 2
COM :	: COM1 (9600,E,7,1)		Settings

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 executed using a 32-bit data type (int or float), the function will automatically convert the data. For example,
	<pre>macro_command main() float f GetData(f, "MODBUS", 6x, 2, 1) // f will contain a floating point value end macro_command</pre>
Example	<pre>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 GetData(a, "Local HMI", LB, 2, 1) // get 30 states of LB0 ~ LB29 to the variables b[0] ~ b[29] GetData(b[0], "Local HMI", LB, 0, 30) // get one word from LW2 to the variable c GetData(c, "Local HMI", LW, 2, 1) // get 50 words from LW0 ~ LW49 to the variables d[0] ~ d[49] GetData(d[0], "Local HMI", LW, 0, 50) // get 2 words from LW6 ~ LW7 to the variable e // note that the type of e is int GetData(e, "Local HMI", LW, 6, 1) // get 20 words (10 integer 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)</pre>
	<pre>// get 2 words from LW2 ~ LW3 to the variable f GetData(f, "Local HMI", LW, 2, 1)</pre>



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	<pre>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) // get one word from LW2 to the variable c GetDataEx (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) // get 2 words from LW6 ~ LW7 to the variable e // note that he type of e is int GetDataEx (e, "Local HMI", LW, 6, 1)</pre>



Γ

 // get 20 words (10 integer values) from LW0 ~ LW19 to f[0] ~ f[9] // since each integer value occupies 2 words GetDataEx (f[0], "Local HMI", LW, 0, 10)
<pre>// get 2 words from LW2 ~ LW3 to the variable f GetDataEx (f, "Local HMI", LW, 2, 1) end macro command</pre>

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 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, 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

When a SetData() is executed using a 32-bit data type (int or float), the 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
	end macro_command
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
	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)
	// act the velues of LN/C LN/Z meta that the time of a is int
	// set the values of LW6 ~ 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", LW, 0, 10)
	end macro_command

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

Tiext
// set the state of LB2 SetDataEx (a, "Local HMI", LB, 2, 1)
// set the states of LB0 ~ LB29 SetDataEx (b[0], "Local 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)
<pre>// set the values of LW0 ~ LW19 // 10 integers equal to 20 words, since each integer value occupies 2 words.</pre>
SetDataEx (f[0], "Local HMI", LW, 0, 10) end macro_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



Macro Reference

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):		
	System Parameter Settings		
	Font Extended Memory Printer/Backup Server Device Model General System Setting Security		
	Device list :		
	No. Name Location Device type Interf I/F 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 Series", RT, 5, 1		
	represents that the address offset is 5.		
	 If address offset uses the format – "N#AAAAA", N indicates that PLC's statior		
	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		
	StringGet(read_data_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the		

			gGet() uses the default station number is not necessary to define station number
	PLC type : FATE	K FB Series	•]
	V.1.1	0, FATEK_FB.so	
	PLC I/F : RS-23	32 🔹	PLC default station no. : 2
	COM : COM1	(9600,E,7,1)	Settings
	🔲 Use	e broadcast command	
	number of data_coun	-	ead from depends on the value of the read_data is restricted to char array.
	read_data		read
	char (8-bit)	1	1
	char (8-bit)	2	1
	• •	ading 2 ASCII o	he size of 2 ASCII characters. According characters is actually reading the content
Example	macro_command ma char str1[20]	lin()	
	// read 10 words fro	m LW0~LW9 to	the variables str1[0] to str1[19]
			CII characters, reading 20 ASCII
	<pre>// characters is actu StringGet(str1[0], "Lo</pre>	, ,	•
	end macro_command	d	



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
	<pre>// macro will continue executing test = 1 even if the MODBUS device is</pre>
	// not responding
	StringGetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 1
	<pre>// macro will not continue executing test = 2 until MODBUS device responds</pre>
	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:

abco

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.	
	StringSet(str1[0], "Local HMI", LW, 0, 10)	
	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
	// macro will continue executing test = 1 even if the MODBUS device is
	// not responding
	StringSetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 1
	// macro will not continue executing test = 2 until MODBUS device responds
	StringSet(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 2



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]="abcdefghij"	
	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"

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'	



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]
	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)
	// success2=true, result2=3c4d.The result exceeds the data range of
	// 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("abcdefghijkImnopqrst", 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.
Example	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])
	// success1=true, dest1="123456790abcdefghij"
	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])



	nonition - Otring Find ("nonvent" to reat [atort])
	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	<pre>position = StringReverseFind (source[start], target[start])</pre>
	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"
	bool pos2
	pos2= StringReverseFind("abcdeabcde", target2[0])
	// pos2=-1
	char src3[20]="abcdeabcde"
	bool pos3
	pos3= StringReverseFind(src3[6], "ab")
	// pos3=-1
	end macro_command
1	

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.
Example	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	<pre>success = StringTrimLeft (source[start], set[start], destination[start])</pre>
	success = StringTrimLeft ("source", set[start], destination[start])
	success = StringTrimLeft (source[start], "set", destination[start])
	<pre>success = StringTrimLeft ("source", "set", destination[start])</pre>
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.

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])
	// success=true, dest="to be or not to be the question"
	success = StringInsert(18, str2[0], dest[0])
	// success=true, dest="to be or not to be, that is the question"
	success = StringInsert(0, "Hamlet:", dest[0])
	// success=false, dest remains the same.
	end macro_command



459



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, 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			
	When TRACE encounters the first format specification (if any), it converts the value of the first argument after format and outputs it accordingly. <i>format</i> 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] type			
	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.			
	precision (optional):			
	A nonnegative decimal integer which specifies the precision and			
	the number of characters to be printed.			
	<i>type</i> : 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> e [<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 [–]dddd.dddd,			
	where <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 <i>argument</i> 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, d					
	or					
	FindDataSampli	ngDate (data	_log_numbe	er, index, year	, month, day	·)
Description	A query function	A query function for finding the date of specified data sampling file according to				
	sampling no. ar	nd the file in	ndex. The c	late is stored	d into "year"	', "month" a
	respectively in th	e format of N	YYYY, MM a	nd DD.		
	Data Sampling Obje	ct				
	No. Description	Read address	Sample mode	Trigger address		
		Local HMI : L WO Local HMI : L WO		Disable Disable	Disable Disable	Disable l Disable l
	data sampling					
	data sampling The directory o					
	sampling files under the same directory are sorted according to the file name					
	indexed starting from 0. The most recently saved file has the smallest file index					
	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 ->					
	20101230.dtl -> index is 2					
	20110110.dtl -> i					
	20110111.dtl -> i	ndex is 0				



	"return_value" equals to 1 if referred data sampling file is successfully found, oth
	equals to 0.
	"data_log_number" and "index" can be constant or variable. "year", "month", "
	"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 // r and file index 2.
	// the result after execution: success == 1, year == 2010, month == 12 and //day success = FindDataSamplingDate(data_log_number, index, year, month, day)
	end macro_command

Name	FindDataSamplingIndex				
Syntax	return value = FindDataSamplingIndex (data log number, year, month, day, in				
oyntax	or				
	FindDataSamplingIndex (data_log_number, year, month, day, index)				
Description					
Description	A query function for finding the file index of specified data sampling file accord				
	data sampling no. and the date. The file index is stored into "index". "year", "m				
	"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 address 1 Local HMI : LWO Periodical Disable Disable Disable				
	1 Local HMI : LWO Periodical Disable Disable Disable 2 Local HMI : LWO Periodical Disable Disable Disable				
	data sampling no. The directory of saved data: [Storage location]\[filename]\yyyymmdd.dtl. T				
	sampling files under the same directory are sorted according to the file name				
	indexed starting from 0. The most recently saved file has the smallest file index				
	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				



r	
	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, oth
	equals to 0.
	"data_log_number", "year", "month" and "day" can be constant or variable. "in
	"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 // r
	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		
	// the result after execution: success == 1, index == 2		
	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.



Animation	New
Auto change remote 1 win change remote 2 win	Delete
change remote 3 win COS	Edit
INVBIT OnAuto OnAutoEnd	Сору
OnAutoStart OnChangeBrent	Paste
On Change Dubai On Change WTI	ОК
OnInit OnShowBrent OnShowDubai OnShowWTI OnTime_500ms OnTimer0 OnUpdateChart Scaling Scan Alarm screen saver Macro under development :	Cancel

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.

orkSpace			<u>_</u> 2
Macro ID : 29	Macro name : m	acro_test	
macro_command main()			
end macro_command			
8			
GET/SET FN Co	ompile	Exit	Help

Step 3:

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.

WEINTEK

Function name :	GetData	*	
	GetData(data[0], "Loca	al HMI", LW, 0, 4)	
Variable 1	<u></u>		
Variable type :	short (16-bit)		
Variable :	data 🔹	Array index : 0	
Read address PLC name : Device type :	Local HMI		•
)		
Address :	1		
Address format :	ddddd [range : 0 ~ 103	17	
	BIN	Data count : 4	
Description] Read data from a	device.		
Read data from a Usage]	device. .C name, device type, ad	dress, data count)	E
Read data from a Usage]		dress, data count)	E

Step 4:

After the completion of a new macro, press 'Compile" button to compile the macro.

WorkSpace			×
Macro ID : 29	Macro name : macro	ro_test	
macro_command main() short data[4]			
GetData(data[0], "Local	HMI", LW, 0, 4)		
end macro_command			
0 error(s)			
GET/SET FN	ompile	Exit Help	

If there is no error, press "Exit" button and find that a new macro "macro_test" exists in "Macro list".

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		Const
OnAuto		Сору
OnAutoEnd OnAutoStart		Paste
OnChangeBrent OnChangeDubai		ОК
OnChangeWTI OnInit		Cancel
OnShowBrent OnShowDubai		
OnShowWTI OnTime_500ms		
On Timer0		
OnUpdateChart Scaling		
Scan Alarm	*	
Macro under development :		

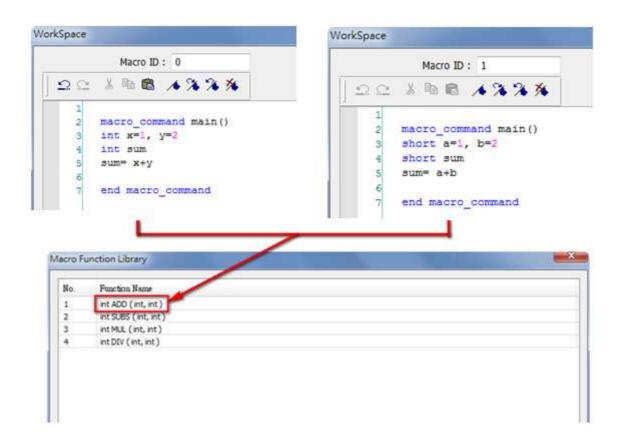
18.7.2 Execute a Macro

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".
 - 2. 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 User Defined Macro Function

When editing Macro, to save time of defining functions, user may search for the needed from built-in Macro Function Library. However, certain functions, though frequently used, may not be found there. In this case, user may define the needed function and save it for future use. Next time when the same function is required, the saved functions can be called from Macro Function Library for easier editing. Additionally, Macro Function Library greatly enhances the portability of user-defined functions. Before building a function please check the built-in functions or online function library to see if it exists.



18.8.1 Import Function Library File

Open a project in HMI programming software, the default Function Library File will be read automatically and the function information will be loaded in. At this moment if a user-defined function is called, the relevant *.mlb file must be imported first.

- 1. Default Function Library File Name: MacroLibrary (without filename extension)
- 2. Function Library Directory: HMI programming software installation directory\library (folder)
- 3. \library (folder) contains two types of function library files:
 - Without filename extension: MacroLibrary, the Default Function Library for HMI programming software to read at the beginning.
 - With filename extension (*.mlb): Such as "math.mlb". The files to be read / written when users import / export. These files are portable and can be called from the folder when needed.
- 4. When opening HMI programming software, only the functions in Default Function Library will be loaded in, to use functions in *.mlb files, please import them first.

C:\EB800	D\library	
	17	
Organize 🔻 New fol		
🚖 Favorites	Name	Date modified
📃 Desktop	picture	2011/10/13 上午1
📕 Downloads	🕌 shape	2011/10/12 上午 0
🖳 Recent Places	J sound	2011/10/12 上午 0
	length 0926.mlb	2008/7/16 下午 02:.
词 Libraries	MacroLibrary	2007/8/5 上午 01:39
Documents	map1.tlb	2007/8/5 上午 01:39
👌 Music	🗋 math.mlb	2007/8/5 上午 01:39
Pictures	menu01.flb	2007/8/5 上午 01:39
Videos		
🜉 Computer		
🏭 Local Disk (C:)		
CD Drive (D:) STEP	7	
FAE (\\vboxsrv) (E:	0	



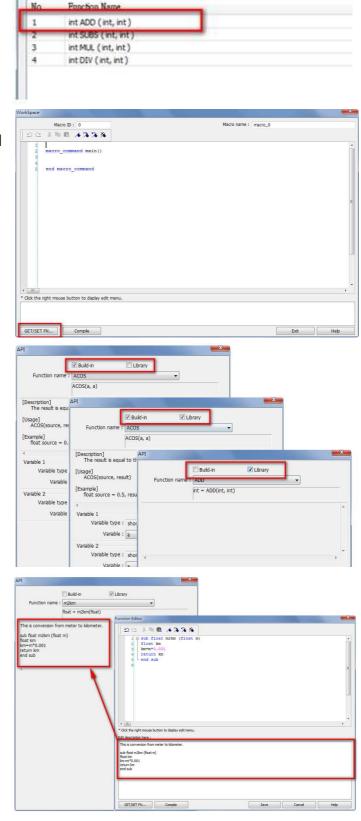
18.8.2 How to Use Macro Function Library

Macro Function Library

- 1. Select the function directly from Macro Function Library.
- In WorkSpace click [GET/SET FN...] to open API dialog box.

 At least check one from [Library] or [Build-in] and select the function to be used.

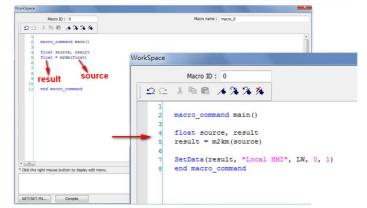
4. The description displayed in API dialog is the same as written in Function Editor.





Macro Reference

 Select the function to be used, fill in the corresponding variables according to the data type.



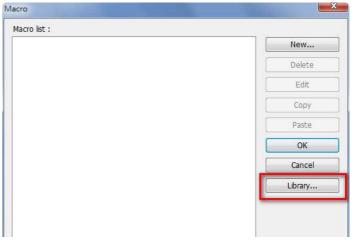
Upon completion of the steps above, user-defined functions can be used freely without defining the same functions repeatedly.



 Open Macro management dialog, click [Library] to enter Macro Function Library interface.

WEINTER

2 A list of functions will be shown, when the project is opened, the software will load in all the functions in the Default Function Library.



No.	Function Name		
1	int ADD (int, int)		
2	int SUBS (int, int)	Lists Default Functions	
3 4	int MUL (int, int) int DIV (int, int)	LISts Delautt Functions	
ADD: a	+b	Displays Function Description	
ADD: a	+b	Displays Function Description	
ADD: a		Displays Function Description	

3. The format of each line in function list:

```
return_type function_name ( parameter_type1, ..., parameter_typeN)
```

return_type indicates the type of the return value. If this value does not exist, this column will be omitted. function_name indicates the name of the function. "N" in *parameter_typeN* stands for the number of parameter types. If this function does not accept any parameters, this column will be omitted.

```
1 □ sub int ADD(int a, int b)

2 | int ret

3 | ret = a+b

4 | return ret

5 | end sub

6
```



18.8.3.1 Create a Function

1. Click [New] to enter Function Editor.

2. Edit function in Function Editing Field.

3. Edit function description here:

specifications, usages, editor's

statement etc.

No.	Function Name	
1	int ADD (int, int)	
2	int SUBS (int, int)	
3	int MUL (int, int)	
4	int DIV (int, int)	
ADD: a	a+b	
ł Nev	w Delete Edt	,
Exp	port	ОК
iction	n Editor	
9		
	Eurotian Editing Field	
	Function Editing Field	
	Function Editing Field	
- (11)		к
		,
		,
Click t	the right mouse button to display edit menu.	
Click t	the right mouse button to display edit menu. scription here :	

4. After editing a function, click [Compile] and [Save] to save this function to the Library. If it is not compiled, a warning dialog will be shown.



5. Successfully added into Macro Function Library.

No.	Function Name	
1	int ADD (int, int)	
2	int SUBS (int, int)	
3	int MUL (int, int)	
4	medav (me, me)	
5	short test1 (short)	
_		
this is a	a macro about square	
this is a	a macro about square	
this is a	a macro about square	
this is a	a macro about square	
this is a	a macro about square	
this is a	a macro about square	
this is a	a macro about square	
this is a	a macro about square	,
4		,
		,
4		,
4	w Delete Edit	, ,





- 1. The total size of data type can be declared in a function is 4096 bytes.
- 2. Function name must only contain alphanumeric characters, and cannot start with a number.

Macro Function Library



18.8.3.2 Delete a Function

1. In function list select the function to be deleted and click [Delete].

	Function Name		
1	int ADD (int, int)		
2	int SUBS (int, int)		
3	int MUL (int, int)		
4	int DIV (int, int) short test1 (short)		
	and clears (and c)		
this is a	a macro about square		
4			,
Nev	v Delete	Edit	
Exp			ОК
Exp	Import		OK
acro Fu	unction Library		
No.	Function Name		
1	int ADD (int, int)		
2	int SUBS (int, int)		
2 3	int SUBS (int, int) int MUL (int, int)		
2 3 4	int SUBS (int, int) int MUL (int, int) int DIV (int, int)		
2 3	int SUBS (int, int) int MUL (int, int)		
2 3 4	int SUBS (int, int) int MUL (int, int) int DIV (int, int)	Delete	
2 3 4	int SUBS (int, int) int MUL (int, int) int DIV (int, int)		
2 3 4	int SUBS (int, int) int MUL (int, int) int DIV (int, int)	Delete	
2 3 4	int SUBS (int, int) int MUL (int, int) int DIV (int, int)		
2 3 4	int SUBS (int, int) int MUL (int, int) int DIV (int, int)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int) short test1 (short)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int) short test1 (short)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int) short test1 (short)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int) short test1 (short)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int) short test1 (short)	Delete	
2 3 4 5	int SUBS (int, int) int MUL (int, int) int DIV (int, int) short test1 (short)	Delete	5
2 3 4 5	In SUBS (Int, Int) In IMUL (Int, Int) In DIV (Int, Int) In DIV (Int, Int) short test1 (short)	Delete	, ,
2 3 4 5 this is a	Int SUBS (Ht, Int) Int MUL (ht, Int) Int DU (ht, Int) short test1 (short)	Delete	, ,

- 2. Click [Yes] to confirm, [No] to cancel the deletion.
- 3. Click [Yes] to delete MAX_SHORT function.

Macro Function Librar

GET/SET FN...

Compile

18.8.3.3 Modify a Function

- 1. Users can modify the functions exist in the Library.
- 2. Select a function to modify by clicking [Edit] to enter Function Editor
- Double click on the function to be modified can also enter Function Editor.
- 4. After modifying, [Compile] then [Save] before leaving.

No.	Function Name	
1	int ADD (int, int)	
2	int SUBS (int, int)	
3	int MUL (int, int)	
4	int DIV (int, int)	
5	short test1 (short)	
this is a 4 New	macro about square	F
Exp		ок
nction		
	A The Constraint of the second s	
•		,
Click th	e right mouse button to display edit menu.	•
Click th	ription here :	•
Click th		

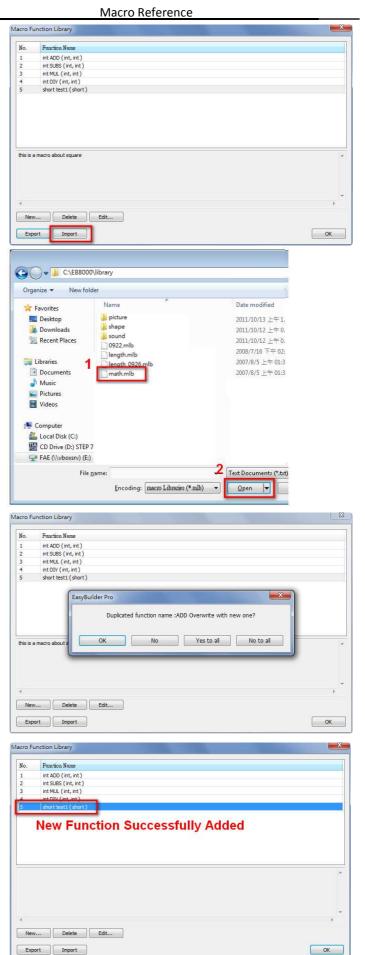
Cancel

Help



18.8.3.4 Import a Function

- 1. Functions can be imported using an external *.mlb file.
- EX: To import a function library "math.mlb" which contains a function "test1".
- 3. Click [Open].
- When importing a function with a name which already exists in the Library, a message will popup.
 - OK: Overwrite the existing function with the imported one.
 - NO: Cancel the importing of the function with the same name.
 - Yes to all: Overwrite using all the imported functions with the same name.
 - No to all: Cancel the importing of all the functions with the same name.
- The imported functions will be saved in Default Function Library, so if "math.mlb" file is deleted, "test1" will still exist in the Library, even when restart software.

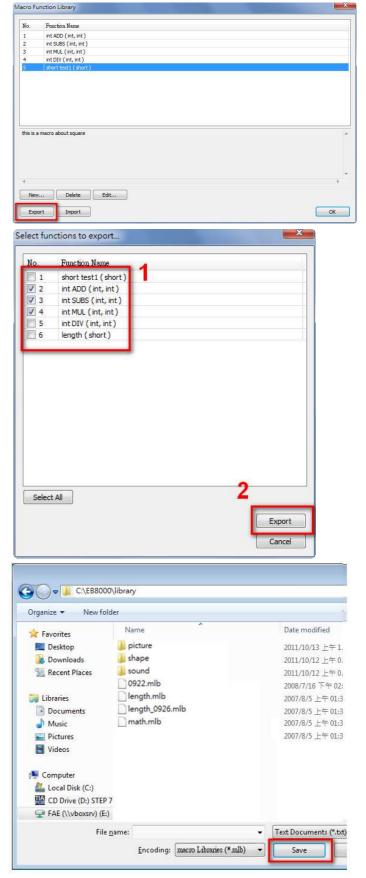




18.8.3.5 Export a Function

- Export the function from Function Library and save as *.mlb file.
- 2. Click [Export].

- Select the function to be exported, and click [Export].
- A "math.mlb" file can be found under export directory. This file contains 4 functions: ADD, SUBS, MUL, and DIV.



 The exported *.mlb file can be imported to another PC. Open HMI programming software, import, then the functions in this file can be used.



18.9 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.

18.10 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:

No.		Name	Location	Device type	Interface
Loca	IHMI	Local HMI	Local	MT8121T (800 x 600)	Disable
Loca	I PLC 1	MODBUS RTU Device	Local	Free Protocol	COM1(19200,E,8,1)
ce Prope	rties				
Name	MOR	BUS RTU Device			
Marrie		<u> </u>		<u>n</u>	
	© HN				
Location	: Loca	I → S	ettings		
		Brotocol			
LC type	1140	0, FREE_PROTOCOL.so			
LC type	V.1.0				
PLC I/F		32 -			

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	





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).

Reque	st		
•	Function code	1 Byte	0x03
	Starting Address	2 Bytes	0x0000 to 0xFFFF
]	Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Response			
	Function code	1 Byte	0x03
	Byte count	1 Byte	2 x N*
	Register value	N* x 2 Bytes	
	*N = Quantity of Registers		
Error			
	Error code	1 Byte	0x83
	Exception code	1 Byte	01 or 02 or 03 or 04

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])

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])

Lastly, use OUPORT to send out this read request to PLC
```

```
OUTPORT(command[0], "MODBUS RTU Device", 8)// send read 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</pre>
```

```
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</pre>
```

```
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()

```
WEINTEK
```

```
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

18.11 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

For example :

macro_command main() int a, b for a = 0 To 2 b = 4 + a int h , k // $\,$ illegal – definitions must occur before any statements or expressions

// for example, b = 4 + a
next a
end macro command

(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.

18.12 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
```

WEINTEK

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
```

WEINTEK

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

18.13 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)

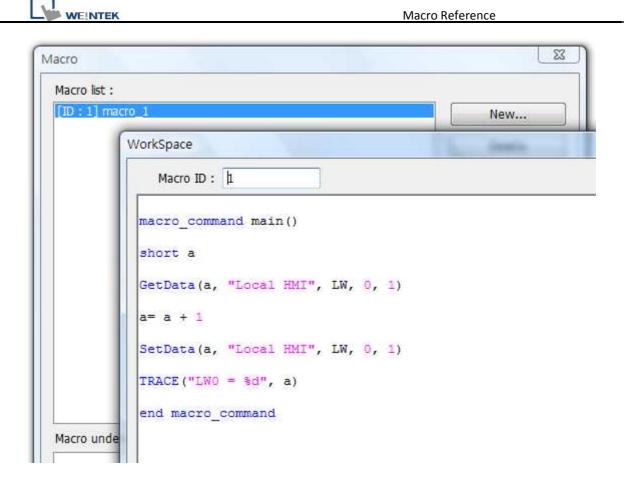
a= a + 1

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.



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.

Numeric Display	General Numeric Format Security Shape Font Profile
0 ####	Description :
Function Key	Read address
	PLG name : Local HMI Address : LW O

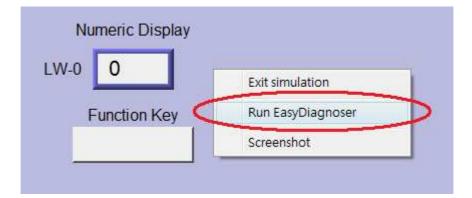


Lastly, compile the completed project and execute Off-line or On-line simulation.



When processing simulation on PC, right click and select "Run

EasyDiagnoser" in the pop-up menu.



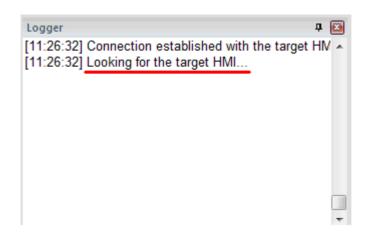
Afterwards, EasyDiagnoser will be started. [Logger] window displays whether EasyDiagnoser is able to connect with the HMI to be watched or not. [Output] window displays the output of the TRACE function. The illustration below shows that EasyDiagnoser succeeds in connecting with HMI.



🤯 Weintek MT Diagno	stic Tool - EasyDiagnose	
i 🖬 🔹 🖄 📝		
<u>F</u> ile <u>V</u> iew <u>O</u> ptions	<u>H</u> elp	
Command: Read + W	rite 🔻 Device: All	Output 🛛 🗜 🖸
Address Type: All	Range:	
No Cm PID	Device St.	
		-
<	۱. E	K F
Devices	д 📧	Logger 🛛 🗘 💌
Local HMI		[11:27:33] Looking for the target HMI
Index	0	[11:27:33] Connection established with the tar
Type Name	MT8000 Series HMI	
Location	Local	
Block Interval	5 words	
Max. Read Length	256 words	
Max. Write Length	256 words	

When EasyDiagnoser is not able to connect with HMI, [Logger] window

displays content as shown below:





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	E	xtended Memory	Printer/Backup Server	
Device	Model	General	System Setting	Securit
HMI ma	del : MTGO70;HA	(T8070)H/M T6100/M	(78100; /800 v 480)	•
HMI mo	odel : MT6070iH/M	1T8070iH/MT6100i/M	1T8100i (800 x 480)	•
HMI mo HMI station	1	1 T8070iH/M T6100i/N	1T8100i (800 x 480)	•

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		Þ
HMI Name: Default HMI v Search Search All	192.168.1.103 (Tina-MT8100i) 192.168.1.117 (nicolas_mt8104×h) 192.168.1.118 (nicolas_8100i) 192.168.1.208 (kevin_MT8070iH) 192.168.1.226 (susan-hmi)	
* OS 20091002 or later supports	192.168.1.233 (Demo-Joey) 192.168.1.237 (Demo-S7, 1200)	<u>~</u>
Project Port: 8005	ОК	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] type
	Each field of the format specification is described as below:
	flags (optional):
	-
	+
	width (optional):
	A nonnegative decimal integer controlling the minimum
	number of characters printed.
	precision (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.
	u : unsigned decimal integer. X or x : unsigned hexadecimal integer.
	E or e : Signed value having the form.
	[–]d.dddd e [sign]ddd where d 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 –.
	e_{A} and e_{A



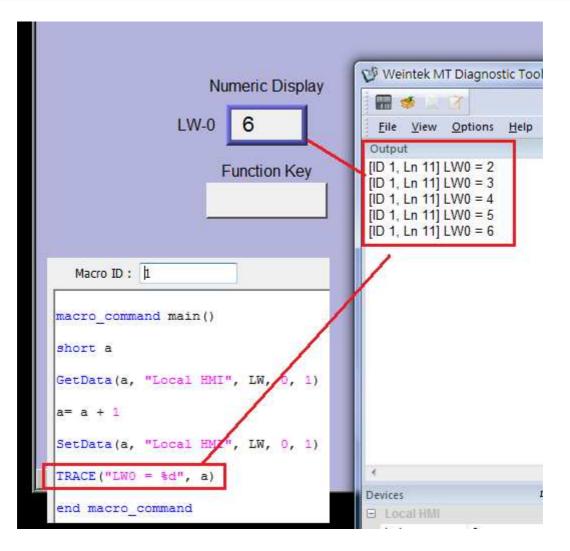
	f : Signed value having the form [–] <i>dddd.dddd</i> ,
	where <i>dddd</i> is one or more decimal digits.
	The length of output string is limited to 256 characters.
	The <i>argument</i> part is optional.
Example	macro_command main()
	char c1 = 'a'
	short s1 = 32767
	float f1 = 1.234567
	TRACE("The results 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	Macro Reference
Project Manager	
HMI IP, Password Type : MT8000 X Se Settings	eries
Connection ③ Ethernet HMI IP :	HMI Name: Default HMI 192.168.1.103 (Tina-MT8100i) Search 192.168.1.117 (nicolas_mt8104xh) 192.168.1.117 (nicolas_mt8104xh) 192.168.1.118 (nicolas_8100i) 192.168.1.208 (kevin_MT8070iH) 192.168.1.211 (MT8070iH - Katte) 192.168.1.226 (susan-hmi) 192.168.1.233 (Demoslow)
Data/Event Lo	Project Port: 8000 V OK Exit
and the second se	ilder8000
EasyConverter	EasyAddressViewer
EasyPrinter	EasyDiagnoser
Recipe/Extende	ed Memory Editor
Build Download Da	ta 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.



WEINTER

18.14 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:

Feneral	Data Entry	Security	Shape	Font					
i	Description :							1	
	🔲 Mask		🗌 Use U	NICODE	🔲 Rever	se high/lo	w byte		
Read	address								
Acou .	PLC name :	Local HMI				~	Setting		
	Address :			✓ 0		-			
ddress									1
aaress									
	PLC name :	Local HMI							~
	evice type :								~
	Address :]System tag		User-de	fined tag	
	ess format : E	DDDDD [ran	nge : 0 ~	10799]					
Addre				Г]Index registe	ər			
Addre									
Addre						of word	3		

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'	Έ'
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:

Аасто	E
Macro list :	New
	Delete

Edit the content:

WorkSpace	
	Macro ID : 1
128	ä 🖻 🖻 🔺 🔧 🌤 🎋
1	
2	macro_command main()
3	
4	char str[20]
5	
6	StringGet(str[0], "Local HMI", LW, 0, 20)
7	StringSet(str[0], "Local HMI", LW, 50, 20)
8	
9	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.



AAAAAAA	
• Execute macro	Macro : [[ID:001] macro_1
Read address	
PLC name : L	.ocal HMI Setting
Address : L	w v 0
Address	
Autoress	
PLC name :	Local HMI
Device type :	LW
Address ;	0 System tag User-defined
Address format :	DDDDD [range : 0 ~ 10799]
	Index register

		Macro Reference
Read address PLC name : Local HMI Address : LW	50	Setting
Address PLC name : Local HM Device type : LW Address : 50 Address format : DDDDD [System tag	View of the second seco
Tag Library	Index register No. c	of word : 10 OK Cancel
_		nd execute Off-line Ze or w to operate the executing
project:		
Test 1:	ABCDE G	Step 1: input string
Test 1:	ABCDE G	Step 2: press "GO" button
Test 1:	ABCDE ABCDE	Step 3: output string

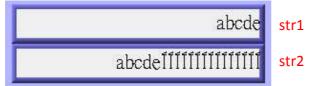


2. Initialization of a string.

Create a new macro and edit the content:

WorkSpace	
	Macro ID : 1
20	X 🖻 🛍 🔺 🛠 🌤 🎋
1	
2 n 3	macro_command main()
4 0	har str1[20]="abcde"
5 00	har str2[20]={'a','b','c','d','e'}
6	
	StringSet(str1[0], "Local HMI", LW, 0, 20)
8 8	StringSet(str2[0], "Local HMI", LW, 50, 20)
8 2	
10 e	nd 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:



	Macro ID : 1 M					
2 2	X 🖻 🛍 🔺 🛠 🌂 🛠					
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	char message_success[40] ="Success! Access Accepted."					
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	<pre>name_match = StringCompare(name_input[0], name[0])</pre>					
15	<pre>password_match = StringCompare(password_input[0], password[0])</pre>					
16						
17	FILL(message_clear[0], 0x20, 40)// FILL with white space					
18	StringSet(message_clear[0], "Local HMI", LW, 100, 40)					
19 I	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.

WEINTEK	Macro Reference	
Accor Pass	Object 2 unt Name word: Dbject 3 Maaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	化二乙基 医化二乙酸 医化二乙酸
AD_9 AAAAA	АААААААААААААААААААААААААААААААААААА	
		8
Object settings:		
Object 1: Function Ke	y <u></u>	

ct 2: ASCII Inp	
address	
PLC name : Local HM	II Setting
Address : LW	V 0
PLC name : Local Device type : LW	ID TAX
and the second se	System tag User-defined ta
Address : 0	
san and so thereas) [range : 0 ~ 10799]
Address : 0 ddress format : DDDDD) [range : 0 ~ 10799]



Object 3: ASCII Input

)ata Entry 🛛 Secu	urity Shape	Font	Profile			
De	scription :						
E	🗹 Mask	🗌 Use	UNICODE	Reve	erse high/lo	w byte	
Read add							
PL	.C name : Loca	HMI			~ [Setting	
ŝ.	Address : LW		~ 50)			
					10		
dress							
PLC	I name : Local	HMI					
Devic	e type : LW						
	CONTRACT OF STREET, ST			-			<i>c</i> . 11
A	ddress : 50		,	System tag		User-di	efined tag
	format : DDDDI) [range : 0 ~	10799]				
Address				1000 /A 100			
Address			1	Index reais	ter		
Address			I	Index regis	ter . of word :	10	

Object 4: ASCII Display

PLC name : Lo Address : Lo	ocal HMI W	✓ 100	Setting	
dress				
PLC name :	Local HMI			~
Device type :	LW	1		~
Address :	100	System tag	User-defined tag	
Address format : I	DDDDD [range : 0 ~	10799] Index register No. of wor	d: 20	
Tag Library		No. of wor	d : 20	



Lastly, compile $\overset{\frown}{\sim}$ the completed project and execute Off-line $\overset{\blacksquare}{=}$ or

On-line \blacksquare simulation. Follow the steps below to operate the executing

project:

© *	*	
	Account Name: adm9ten 1: input	
	Password: account name	
	Login	
! @ #	\$ % ^ & * () BS	
~ q w e	r t y u i o p { }	
Caps Z X	d f g h j k l Enter c v b n m < > ?	
Clear	SPACE + =	
		_

Account Name: admin Password: ****** Step 2: input password Login
1 2 3 4 5 6 7 8 9 0 BS `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 , / \lambda Clear SPACE + - * * *
© * #
Account Name: admin
Password: ****** Step 3: press
Password: ****** Step 3: press Login Login button

L	WEINTEK	

0 *	*
	Account Name: admin Password: ***** Login
Fast Sel	Success! Accepted.
	EasyView
© *	*
*	Account Name: ADMIN Password: ****** Login
C *	Account Name: ADMIN Password: *****



18.15 Macro Password Protection

Pass	vord
	Password : [11111] (max : 10 characters)
	OK Cancel

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.

ord	
Password :	(max : 10 characters)
ОК	Cancel

EasyBuilder8000 should be rebooted for typing the password again after 3 incorrect attempts.

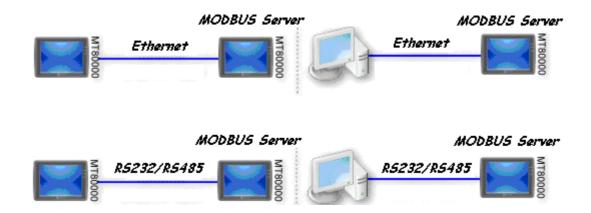


[Caution] When MACRO is password protected, decompilation of XOB file will not be able to restore MACRO contents.

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 Memo	ry	Printer/Ba	ckup Server
Device	Model	General	l Sys	tem Setting	Security
Device list :					
No Na	me Loca	tion Device type	Interfa	ce I/E Protoco	I Station no
roperties					
Name :	MODBUS Ser	vor			
1					
	⊖ HMI	● PLC			
Location :	Local	Settin	gs		
PLC type :	MODBUS Ser	ver			
	V.1.UU, MOD	BUS_SERVER.so			
PLC I/F :	RS-485 2W	*		Station n	0.: 1
	RS-232				
	RS-485 2W RS-485 4W				
	Ethernet				Settings
COM :					11 12 12 12 12 12 12 12 12 12 12 12 12 1
COM :	USB	cast command			
COM :	USB	cast command			
COM :	USB	cast command			
COM :	USB Use broad	cast command	~		
COM :	USB Use broad	ack (words) : 5	6.0		
COM :	USB Use broad		6.0		
COM : Interv Max. rea	USB Use broad al of block pa d-command s	ack (words) : 5			

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.

Device Model General System Setting Set HMI model: MT6056T/MT8056T (320 x 234)	General	34-3-1	
HMI model : [MT6056T/MT8056T (320 x 234)		Model	Device
	 M 160301 (520 X 234)		
HMI station no : 1 🗸 🗸	172	. no : 1	HMI station

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.

Font		Extended Memory		Printer/Backup Server				
Device	Model		General	Sys	tem Setting	Security	Security	
NI-	blama	Location	Device type		Interface	-	I/F	
No.	Name	Location	Dence gpc		THE WEIGHT OF WEIGHT			
NO. Local HMI	Local HMI	Local	MT6056T/MT	8056T	Disable		N/	

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	
	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 Addres	25 Settings	
	IP address : 192 . 168 . 1 . 111	
	Port no. : 8000	
	Timeout (sec) : 1.0 Turn around delay (ms) : 0	
Seno	ACK delay (ms) : 0 Parameter 1 : 0	
	Parameter 2 : 0 Parameter 3 : 0	
	ОК	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.

100	WE!NTEK

Name : MODE	BUS RTU	
	I <mark>⊙</mark> PLC	
Location : Local	Sett	ngs
PLC type : MOD	BUS RTU	
V.1.9), MODBUS_RTU.so	
		and a second second
PLC I/F : RS-44 COM : COM1 Port Settings	95 2W 💽 (9600,E,8,1)	PLC default station no. : 1
сом : сом1	(9600,E,8,1)	Settin
COM : COM1 Port Settings	(9600,E,8,1)	Settin
COM : COM1 Port Settings COM : COM	(9600,E,8,1)	Settin Timeout (sec) : 1.0
COM : COM1 Port Settings COM : COM Baud rate : 960	(9600,E,8,1)	Settin Timeout (sec) : 1.0 Turn around delay (ms) :
COM : COM1 Port Settings COM : COM Baud rate : 960 Data bits : 8 Bit	(9600,E,8,1)	Timeout (sec) : 1.0 Turn around delay (ms) : 0 Send ACK delay (ms) : 0

Set and click **[OK]**, a new device"MODBUS RTU" will be listed in the **[Device]** tab.

-		Extended Memory		Printer/Backup Server		
Device	Model		General	System Setting	Securit	y
Device list :	Name	Location	Device type	Interface	I/F Protocol	Sta

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.





PLC name :	MODBUS RTU			-	 Sett
Address :	0x	~	0		
	0x				
	1× 3×_Bit 4×_Bit				
	I3X_BIC 4√ BiF				
	6x_Bit		-		
	0x_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:

 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.
- (3) 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

Barcode interfaces:



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.

				🛄 System Parame	ter Settings			X
Click [Setti	inas1	and		I Font		Extended Memory		ckup Server
-				B E Device	Model	General	System Setting	Security
finish [Bar		evice	_	Device list :	Location	Device type	Interface I/F Protoco	
Settings]:			De	vice Properties			Timenare de emilio	n stann n
				Name	: Barcode (USB/C	OM)		1
The setting	s are de	etailed	d k	, and the	- <u></u>			
respectivel	•			Location				
respectiver	y below	•		Location	Local	Settings		
				PLC type	: Barcode (USB/C	OM)		~
	A				 Independent of a start of the local data of the local			
Barcode Device	Settings							
		-	Read	byte limit				
	COM 1	~						
Baud rate :	9600	*	10			•		Settings
Data bits :	8 Bits	~	-					
Parity :	None	~	✓ Use a	. start code				
Stop bits :	1 Bit	~		Start code : 2	55			
	-		Terminate	nr.				
			⊙ CR/L		O Other 0			
			CRIL	r USIA/EIA				
			○ None				ОК	Cancel
				OK		ancel		
					200	5		

100	WEINTEK	

СОМ :	COM 1	~	Read byte limit
Baud rate :	9600	~	10 📚
Data bits :	8 Bits	~	
Parity :	None	~	Use a start code
Stop bits :	1 Bit	~	Start code : 255
			Terminator CR/LF OSTX/ETX OOther O None

[COM]、[Baud rate]、[Data bits]、[Parity]、[Stop bits]

Barcode device can be connected to any of COM 1 ~ COM 3 or USB. 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. Enter the decimal ASCII value of the character.

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 0 indicates	the status of data reading. When reading			
Bit FLAG	data is complete, the status of FLAG 0 will be changed					
	FLAG	from OFF to ON. It will not return to OFF automatically,				
		users are free to s	et base on actual usage.			
Word	BARCODE	BARCODE 0	Number of bytes currently read.			
word	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

Read byte limit	(bytes) and BARCODE 1 ~ n is the address of "ASCII Display" object (barcode).
Use a start code Start code : 0	Address : BARCODE 0 BYTES : 13 Address : BARCODE 1~n
 OR/LF OSTX/ETX OOther ○None 	BARCODE : 9421007480830



In the example the data stored by barcode device corresponding address are listed below:

Barcode	Data
corresponding address	Data
	13 bytes (decimal)
BARCODE 0	The data saved in this address is 14 bytes = 7 words. If
DANCODE U	the number of byte is odd, system will add a byte (0x00)
	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.



ote

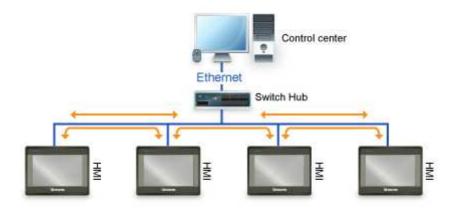
Please confirm your Internet connection before downloading the demo project.



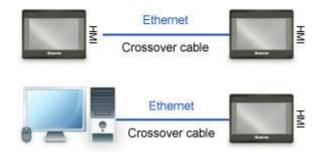
Chapter 21 Ethernet Communication and Multi-HMI Connection

There are two ways of Ethernet communication:

1. Use RJ45 straight through cable + hub



2. Use RJ45 crossover cable and without hub, but this is limited to the condition of point to point connection (HMI to HMI or PC to HMI).



Through Ethernet network, EasyBuilder Pro provides the following methods for data transmission:

- 1. HMI to HMI communication
- 2. PC to HMI communication
- 3. Operating the PLC connected to other HMI



21.1 HMI to HMI Communication



In the communication between HMI A and HMI B, when using [set bit] object on HMI A to control [LB-0] of HMI B:

1. Set the IP address of the two HMI, example: HMI A = 192.168.1.1, HMI B = 192.168.1.2

Add a remote HMI B. IP 192.168.1.2			Device Properties	2. HMI A project /
Add a remote HMI B. IP 192.168.1.2		НМІВ	Name :	[System Parameter Settings]
Port no. : 8000	3000)	: Remote Settings JP : 192.168.1.2 (Port = 8000) IP Address Settings Ethernet IP address : 192 . 168 . 1 . 2 Port no. : 8000	Location :	Add a remote HMI B.

2. Set Bit / [PLC name]

select "HMI B" to control the address of remote HMI.

neral Security :	Shape 📗 La	abel				
Description :	[
Write address —						
PLC name :	HMI B				*	Setting
Address :	LB		v 0			
				Write after 1	button is rel	eased
Attribute						
Set style :	Set ON					*



One HMI can handle requests from a maximum of other 32
 HMI simultaneously.

21.2 PC to HMI Communication



With On-line Simulation Function, PC can collect data of HMI through Ethernet network and save the data files on PC.

PC can control HMI by operating system reserved register.

HMI can control PC, for example, commanding PC to save data from HMI or PLC.

Suppose PC is going to communicate with two HMI (HMI A and HMI B), the MTP project on PC setting procedure:

1. Set the IP address of the two HMI, example: HMI A = 192.168.1.1, HMI B = 192.168.1.2

2. PC project/	System Parameter Settings						X	
[System Parameter] /	Font		Exte	nded Memory		Printer/Backup Server		r]
[Device List], add remote	Device Device list :	Mo	del	General	S	ystem Setting	Sect	arity
HMI A &HMI B.	No.	Name	Location			Device type	Interface	I/F Pro
	Local HMI	Local HMI				MT6050i/MT	Disable	N/A
	Remote HMI 1	1200000000	Contraction of the local division of the	° 192 168 1 1,		a work to see a set	Ethemet	TCP/IP
3. 🖾 Set Bit /	Remote HMI 2	HMI B	Remote (II	?:192.168.1.2,	Port=8000)	MT8xxx	Ethernet	TCP/IP
[PLC name], select the device to be controlled, if it's HMI A [LB], select "HMI A".	Descri - Write addre PLC r Add	urity Shap ption :			- Stranger and the	Ter button is rele	Setting.	
	Attribute	t style : Set	ON					

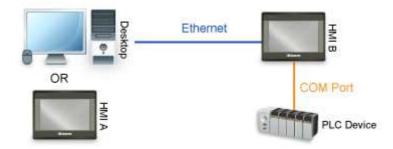


The number of HMI that a PC can control is not limited.HMI can control data on PC by considering PC another HMI.



Add PC as a new Remote HMI device to the HMI MTP project and set the IP address of the Remote HMI pointing to the PC.

21.3 Operate the PLC Connected with Other HMI



Through Ethernet network, PC or HMI can operate PLC that is connected to other HMI; for example, a Mitsubishi PLC connected to COM 1of HMI B. When using PC or HMI A to read PLC data, the procedure for setting PC or HMI A MTP projects:

OHMI

⊙PLC

V.1.10, MITSUBISHI_FXON.so

PLC default station no. : 0

Use broadcast command

~

Default station no. use station no. variable

Settings ... IP : 0.0.0.0 (Port = 8000)

v

1. Set HMI B IP, for example: 192.168.1.2

Device Properties 2. PC or HMI A project / Name : PLC on HMI B [System Parameter] / Location : Remote [Device list], add a remote PLC, and set PLC type : MITSUBISHI FX0n/FX2 correct parameters. Since this PLC is connected to PLC I/F: RS-485 4W remote HMI B, set IP the same as HMI B (192.168.1.2). COM : COM1 3. Set Bit/ [PLC name]

select "PLC on HMI B" (remote PLC) to control the PLC connected to HMI B.

neral	Security	Shape	Label					
- 3	Description	ı:[
Write	address —							
	PLC name	PLC o	n HMI B				× [Setting
	Address	: x		*	0			
					🔲 Write (after butto:	n is release	d
Attrib	ute							



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:

No.	Address tag n	ame			PLC name	Address type	Address	1
1	LB-9000 ; ini	tialized as ON			Local HMI	Bit	LB-9000	
2	LB-9001 : ini	tialized as ON		(Local HMI	Bit	LB-9001	
3	LB-9002 : ini	tialized as ON			Local HMI	Bit	LB-9002	
4	LB-9003 : ini	tialized as ON			Local HMI	Bit	LB-9003	
5	LB-9004 : ini	tialized as ON			Local HMI	Bit	LB-9004	
6	LB-9005 : ini	tialized as ON			Local HMI	Bit	LB-9005	
7	LB-9006 : ini	tialized as ON			Local HMI	Bit	LB-9006	
8	LB-9007 : ini	tialized as ON			Local HMI	Bit	LB-9007	
Э	LB-9008 : ini	tialized as ON			Local HMI	Bit	LB-9008	
10	LB-9009 : ini	tialized as ON			Local HMI	Bit	LB-9009	
11	LB-9010 : da	ta download indicator			Local HMI	Bit	LB-9010	
12	LB-9011 : da	ta upload indicator			Local HMI	Bit	LB-9011	
13	LB-9012 : da	ta download/upload in	dicator		Local HMI	Bit	LB-9012	
14	LB-9013 : FS	window control[hide(C	IN)/show(OFF)]		Local HMI	Bit	LB-9013	
15	LB-9014 : FS	button control[hide(O)	V)/show(OFF)]		Local HMI	Bit	LB-9014	
16	LB-9015 : FS	window/button contro	l[hide(ON)/show(OFF)]		Local HMI	Bit	LB-9015	
17	LB-9016 : sta	atus is on when a client	connects to this HMI		Local HMI	Bit	LB-9016	
18	LB-9017 : dis	able write-back in PLC	control's [change wind	ow]	Local HMI	Bit	LB-9017	
19	LB-9018 : dis	able mouse cursor (set	ON)		Local HMI	Bit	LB-9018	
20	LB-9019 : dis	able/enable buzzer			Local HMI	Bit	LB-9019	
21	LB-9020 : sh	ow (set ON)/ hide (set	OFF) system setting bar		Local HMI	Bit	LB-9020	
22	LB-9021 : re:	set current event log (si	et ON)		Local HMI	Bit	LB-9021	18
n.	1.0.0000	Tìr	- 24° 100 - 2			C BAL	1.0.000	>
								1000
Users	can import M i Su	10 tag to represent the	address.					
	New	Delete	Delete All	Settings				
Sa	ve Tag File	Load Tag File						
E	xport CSV	Import CSV	Export EXCEL	Import EXCE		[Exit	



22.1 The Address Ranges of Local HMI Memory 22.1.1 Bits

Memory	Device Type	Range	Format
Local Memory	LB	0 ~ 12095	DDDDD
Bits			
Local Word Bits	LW_BIT	0 ~ 1079915	DDDDDdd
			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	DDDDDD			
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		1073741823	Limited by device, max.			
			2G			



22.2 HMI Time

		Read(R)/Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LW-9010	(16bit-BCD) : local second	R/W	R/Y	R/Y
LW-9011	(16bit-BCD) : local minute	R/W	R/Y	R/Y
LW-9012	(16bit-BCD) : local hour	R/W	R/Y	R/Y
LW-9013	(16bit-BCD) : local day	R/W	R/Y	R/Y
LW-9014	(16bit-BCD) : local month	R/W	R/Y	R/Y
LW-9015	(16bit-BCD) : local year	R/W	R/Y	R/Y
LW-9016	(16bit-BCD) : local week	R	R	R
LW-9017	(16bit) : local second	R/W	R/Y	R/Y
LW-9018	(16bit) : local minute	R/W	R/Y	R/Y
LW-9019	(16bit) : local hour	R/W	R/Y	R/Y
LW-9020	(16bit) : local day	R/W	R/Y	R/Y
LW-9021	(16bit) : local month	R/W	R/Y	R/Y
LW-9022	(16bit) : local year	R/W	R/Y	R/Y
LW-9023	(16bit) : local week	R	R	R
LW-9030	(32bit) : system time (unit : 0.1 second)	R	R	R
LW-9048	(16bit) : time (0 : AM, 1 : PM)	R/W	R/Y	R/Y
LW-9049	(16bit) : local hour (12-hour format)	R/W	R/Y	R/Y



22.3 User Name and Password

		Read(R)/Write(W)/	Control(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9050	user logout	W	Y	Y
LB-9060	password error	R	R	R
LB-9061	update password (set ON)	W	Y	Y
LW-9219	(16bit) : user no. (1~12)	R/W	R/Y	R/Y
LW-9220	(32bit) : password	R/W	R/Y	R/Y
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/Y	R/Y
LW-9502	(32bit) : user 2's password	R/W	R/Y	R/Y
LW-9504	(32bit) : user 3's password	R/W	R/Y	R/Y
LW-9506	(32bit) : user 4's password	R/W	R/Y	R/Y
LW-9508	(32bit) : user 5's password	R/W	R/Y	R/Y
LW-9510	(32bit) : user 6's password	R/W	R/Y	R/Y
LW-9512	(32bit) : user 7's password	R/W	R/Y	R/Y
LW-9514	(32bit) : user 8's password	R/W	R/Y	R/Y
LW-9516	(32bit) : user 9's password	R/W	R/Y	R/Y
LW-9518	(32bit) : user 10's password	R/W	R/Y	R/Y
LW-9520	(32bit) : user 11's password	R/W	R/Y	R/Y
LW-9522	(32bit) : user 12's password	R/W	R/Y	R/Y





22.4 Data Sampling

	Read(R)/Write(W)/Control(Y)				
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LB-9025	delete the earliest data sampling file on HMI memory (set ON)	W	Y	Y	
LB-9026	delete all data sampling files on HMI memory (set ON)	W	Y	Y	
LB-9027	refresh data sampling information on HMI memory (set ON)	W	Y	Y	
LB-9034	save event/data sampling to HMI, USB disk, SD card (set ON)	W	Y	Y	
LB-11949	delete the earliest data sampling file on SD card (set ON)	W	Y	Y	
LB-11950	delete all data sampling files on SD card (set ON)	W	Y	Y	
LB-11951	refresh data sampling information on SD card (set ON)	W	Y	Y	
LB-11952	delete the earliest data sampling file on USB 1 (set ON)	W	Y	Y	
LB-11953	delete all data sampling files on USB 1 (set ON)	W	Y	Y	
LB-11954	refresh data sampling information on USB 1 (set ON)	W	Y	Y	
LB-11955	delete the earliest data sampling file on USB 2 (set ON)	W	Y	Y	
LB-11956	delete all data sampling files on USB 2 (set ON)	W	Y	Y	
LB-11957	refresh data sampling information on USB 2 (set ON)	W	Y	Y	
LW-9063	(16bit) : no. of data sampling files on HMI memory	R	R	R	
LW-9064	(32bit) : size of data sampling files on HMI memory	R	R	R	
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

				ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9021	reset current event log (set ON)	W	Y	Y
LB-9022	delete the earliest event log file on HMI memory (set ON)	W	Y	Y
LB-9023	delete all event log files on HMI memory (set ON)	W	Y	Y
LB-9024	refresh event log information on HMI memory (set ON)	W	Y	Y
LB-9034	save event/data sampling to HMI, USB disk, SD card (set ON)	W	Y	Y
LB-9042	acknowledge all alarm events (set ON)	W	Y	Y
LB-9043	unacknowledged events exist (when ON)	R	R	R
LB-11940	delete the earliest event log file on SD card (set ON)	W	Y	Y
LB-11941	delete all event log files on SD card (set ON)	W	Y	Y
LB-11942	refresh event log information on SD card (set ON)	W	Y	Y
LB-11943	delete the earliest event log file on USB 1 (set ON)	W	Y	Y
LB-11944	delete all event log files on USB 1 (set ON)	W	Y	Y
LB-11945	refresh event log information on USB 1 (set ON)	W	Y	Y
LB-11946	delete the earliest event log file on USB 2 (set ON)	W	Y	Y
LB-11947	delete all event log files on USB 2 (set ON)ON)	W	Y	Y
LB-11948	refresh event log information on USB 2 (set ON)	W	Y	Y
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 *Note1	R/W	R/Y	R/Y
LW-9451	(16bit) : time tag of event log – minute*Note1	R/W	R/Y	R/Y
LW-9452	(16bit) : time tag of event log – hour *Note1	R/W	R/Y	R/Y
LW-9453	(16bit) : time tag of event log – day *Note1	R/W	R/Y	R/Y



LW-9454	(16bit) : time tag of event log – month *Note1	R/W	R/Y	R/Y
LW-9455	(16bit) : time tag of event log – year*Note1	R/W	R/Y	R/Y
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(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9018	disable mouse cursor (set ON)	R/W	R/Y	R/Y
LB-9019	disable/enable buzzer	R/W	R/Y	R/Y
LB-9020	show (set ON)/ hide (set OFF) system setting bar	R/W	R/Y	R/Y
LB-9033	disable(when on)/enable (when off) HMI upload function(i series only) *Note1	R/W	R/Y	R
LB-9040	backlight up (set ON) *Note2	W	Y	Y
LB-9041	backlight down (set ON) *Note2	W	Y	Y
LB-9047	reboot HMI (set ON when LB9048 is on)	W	Y	Y
LB-9048	reboot-HMI protection	R/W	R/Y	R/Y
LB-9062	open hardware setting dialog (set ON)	W	Y	Y
LB-9063	disable(set ON)/enable(set OFF) popuping information dialog while finding an USB disk (i series support only)	R/W	R/Y	R/Y
LW-9008	(32bit-float) : battery voltage (i series supports only) *Note3	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 *Note2	R	R	R
LW-9080	(16bit) : backlight saver time (unit : minute)	R/W	R/Y	R/Y
LW-9081	(16bit) : screen saver time (unit : minute)	R/W	R/Y	R/Y



1. After changing the settings, please reboot HMI for updating. 2. LW-9040 used together with LB-9040 ~ LB-9041 can adjust the backlight brightness with level $0 \sim 31$.

3. For LW-9008, when the battery voltage level goes below 2.89V, it is recommended to replace the battery.



Read(R)/Write(W)/Control(Y) Remote Address Description Local MACRO HMI R/Y HMI R/Y LW-9125 (16bit) : HMI ethernet gateway 0 (machine R/W R/Y R/Y used only) LW-9126 (16bit) : HMI ethernet gateway 1 (machine R/W R/Y R/Y used only) LW-9127 (16bit) : HMI ethernet gateway 2 (machine R/W R/Y R/Y used only) LW-9128 (16bit) : HMI ethernet gateway 3 (machine R/W R/Y R/Y used only) LW-9129 (16bit) : HMI ethernet IP 0 (machine used only) R/W R/Y R/Y R/Y LW-9130 R/W R/Y (16bit) : HMI ethernet IP 1 (machine used only) LW-9131 (16bit) : HMI ethernet IP 2 (machine used only) R/W R/Y R/Y LW-9132 (16bit) : HMI ethernet IP 3 (machine used only) R/W R/Y R/Y 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 LW-1075 (16bit) : HMI ethernet Mask 0 (machine used R/W R/Y R/Y 0 only) LW-1075 (16bit) : HMI ethernet Mask 0 (machine used R/W R/Y R/Y 1 only) LW-1075 (16bit) : HMI ethernet Mask 0 (machine used R/W R/Y R/Y 2 only) LW-1075 (16bit) : HMI ethernet Mask 0 (machine used R/W R/Y R/Y 3 only)

22.7 Local HMI Network Information



22.8 Recipe and Extended Memory

		Read(R)/Write(W)/Control(
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LB-9028	reset all recipe data (set ON)	W	Y	Y	
LB-9029	save all recipe data to machine (set ON)	W	Y	Y	
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	R	R	R	



	exist (when ON)			
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(R)/Write(W)/Control(Y		
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
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?





22.10 Touch Position

		Read(R)/Write(W)/Control(Y)		
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
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?





22.11 Station Number Variables

		Read(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LW-10000	(16bit) : var0 - station no variable (usage : var0#address)	R/W	R/Y	R/Y
LW-10001	(16bit) : var1 - station no variable (usage : var1#address)	R/W	R/Y	R/Y
LW-10002	(16bit) : var2 - station no variable (usage : var2#address)	R/W	R/Y	R/Y
LW-10003	(16bit) : var3 - station no variable (usage : var3#address)	R/W	R/Y	R/Y
LW-10004	(16bit) : var4 - station no variable (usage : var4#address)	R/W	R/Y	R/Y
LW-10005	(16bit) : var5 - station no variable (usage : var5#address)	R/W	R/Y	R/Y
LW-10006	(16bit) : var6 - station no variable (usage : var6#address)	R/W	R/Y	R/Y
LW-10007	(16bit) : var7 - station no variable (usage : var7#address)	R/W	R/Y	R/Y
LW-10008	(16bit) : var8 - station no variable (usage : var8#address)	R/W	R/Y	R/Y
LW-10009	(16bit) : var9 - station no variable (usage : var9#address)	R/W	R/Y	R/Y
LW-10010	(16bit) : var10 - station no variable (usage : var10#address)	R/W	R/Y	R/Y
LW-10011	(16bit) : var11 - station no variable (usage : var11#address)	R/W	R/Y	R/Y
LW-10012	(16bit) : var12 - station no variable (usage : var12#address)	R/W	R/Y	R/Y
LW-10013	(16bit) : var13 - station no variable (usage : var13#address)	R/W	R/Y	R/Y
LW-10014	(16bit) : var14 - station no variable (usage : var14#address)	R/W	R/Y	R/Y
LW-10015	(16bit) : var15 - station no variable (usage : var15#address)	R/W	R/Y	R/Y





22.12 Index Register

		Read(R)/Write(W)/Control(Y			
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LW-9200	(16bit) : address index 0	R/W	R/Y	R/Y	
LW-9201	(16bit) : address index 1	R/W	R/Y	R/Y	
LW-9202	(16bit) : address index 2	R/W	R/Y	R/Y	
LW-9203	(16bit) : address index 3	R/W	R/Y	R/Y	
LW-9204	(16bit) : address index 4	R/W	R/Y	R/Y	
LW-9205	(16bit) : address index 5	R/W	R/Y	R/Y	
LW-9206	(16bit) : address index 6	R/W	R/Y	R/Y	
LW-9207	(16bit) : address index 7	R/W	R/Y	R/Y	
LW-9208	(16bit) : address index 8	R/W	R/Y	R/Y	
LW-9209	(16bit) : address index 9	R/W	R/Y	R/Y	
LW-9210	(16bit) : address index 10	R/W	R/Y	R/Y	
LW-9211	(16bit) : address index 11	R/W	R/Y	R/Y	
LW-9212	(16bit) : address index 12	R/W	R/Y	R/Y	
LW-9213	(16bit) : address index 13	R/W	R/Y	R/Y	
LW-9214	(16bit) : address index 14	R/W	R/Y	R/Y	
LW-9215	(16bit) : address index 15	R/W	R/Y	R/Y	
LW-9230	(32bit) : address index 16	R/W	R/Y	R/Y	
LW-9232	(32bit) : address index 17	R/W	R/Y	R/Y	
LW-9234	(32bit) : address index 18	R/W	R/Y	R/Y	
LW-9236	(32bit) : address index 19	R/W	R/Y	R/Y	
LW-9238	(32bit) : address index 20	R/W	R/Y	R/Y	
LW-9240	(32bit) : address index 21	R/W	R/Y	R/Y	
LW-9242	(32bit) : address index 22	R/W	R/Y	R/Y	
LW-9244	(32bit) : address index 23	R/W	R/Y	R/Y	
LW-9246	(32bit) : address index 24	R/W	R/Y	R/Y	
LW-9248	(32bit) : address index 25	R/W	R/Y	R/Y	
LW-9250	(32bit) : address index 26	R/W	R/Y	R/Y	
LW-9252	(32bit) : address index 27	R/W	R/Y	R/Y	
LW-9254	(32bit) : address index 28	R/W	R/Y	R/Y	
LW-9256	(32bit) : address index 29	R/W	R/Y	R/Y	
LW-9258	(32bit) : address index 30	R/W	R/Y	R/Y	
LW-9260	(32bit) : address index 31	R/W	R/Y	R/Y	



22.13 MTP File Information

		Read(R)	/Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
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(R)/	R)/Write(W)/Control(
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
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/Y	R/Y	



	(COM 1)			
LW-9542	(16bit) : MODBUS/ASCII server station no. (COM 2)	R/W	R/Y	R/Y
LW-9543	(16bit) : MODBUS/ASCII server station no. (COM 3)	R/W	R/Y	R/Y
LW-9544	(16bit) : MODBUS/ASCII server station no. (ethernet)	R/W	R/Y	R/Y
LW-9570	(32bit) : received data count (bytes) (COM 1 MODBUS server)	R	R	R
LW-9572	(32bit) : received data count (bytes) (COM 2 MODBUS server)	R	R	R
LW-9574	(32bit) : received data count (bytes) (COM 3 MODBUS server)	R	R	R
LW-9576	(32bit) : received data count (bytes) (Ethernet MODBUS server)	R	R	R



		Read(R)/	ontrol(Y)	
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9030	update COM 1 communication parameters (set ON)	R/W	R/Y	R/Y
LB-9031	update COM 2 communication parameters (set ON)	R/W	R/Y	R/Y
LB-9032	update COM 3 communication parameters (set ON)	R/W	R/Y	R/Y
LB-9065	disable/enable COM1 broadcast station no.	R/W	R/Y	R/Y
LB-9066	disable/enable COM2 broadcast station no.	R/W	R/Y	R/Y
LB-9067	disable/enable COM3 broadcast station no.	R/W	R/Y	R/Y
LW-9550	(16bit) : COM 1 mode(0:RS232,1:RS485 2W,2:RS485 4W)	R/W	R/Y	R/Y
LW-9551	(16bit) : COM 1 baud rate(7:1200,8:2400,0:4800,1:9600,2:19200,3: 38400,4:57600,)	R/W	R/Y	R/Y
LW-9552	(16bit) : COM 1 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/Y	R/Y
LW-9553	(16bit) : COM 1 parity (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/Y	R/Y
LW-9554	(16bit) : COM 1 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/Y	R/Y
LW-9555	(16bit) : COM 2 mode(0:RS232,1:RS485 2W,2:RS485 4W)	R/W	R/Y	R/Y
LW-9556	(16bit) : COM 2 baud rate(7:1200,8:2400,0:4800,1:9600,2:19200,3: 38400,4:57600,)	R/W	R/Y	R/Y
LW-9557	(16bit) : COM 2 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/Y	R/Y
LW-9558	(16bit) : COM 2 parity (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/Y	R/Y
LW-9559	(16bit) : COM 2 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/Y	R/Y
LW-9560	(16bit) : COM 3 mode(0:RS232,1:RS485 2W)	R/W	R/Y	R/Y
LW-9561	(16bit) : COM 3 baud rate(7:1200,8:2400,0:4800,1:9600,2:19200,3: 38400,4:57600,)	R/W	R/Y	R/Y
LW-9562	(16bit) : COM 3 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/Y	R/Y
LW-9563	(16bit) : COM 3 parity (0:none, 1:even, 2:odd,	R/W	R/Y	R/Y

22.15 Communication Parameters Settings



		-1		
	3:mark, 4:space)			
LW-9564	(16bit) : COM 3 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/Y	R/Y
LW-9565	(16bit) : COM 1 broadcast station no.	R/W	R/Y	R/Y
LW-9566	(16bit) : COM 2 broadcast station no.	R/W	R/Y	R/Y
LW-9567	(16bit) : COM 3 broadcast station no.	R/W	R/Y	R/Y
LW-10500	(16bit) : PLC 1 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10501	(16bit) : PLC 1 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10502	(16bit) : PLC 1 send ACK delay (unit : ms)	R/W	R/Y	R/Y
LW-10503	(16bit) : PLC 1 parameter 1	R/W	R/Y	R/Y
LW-10504	(16bit) : PLC 1 parameter 2	R/W	R/Y	R/Y
LW-10505	(16bit) : PLC 2 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10506	(16bit) : PLC 2 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10507	(16bit) : PLC 2 send ACK delay (unit : ms)	R/W	R/Y	R/Y
LW-10508	(16bit) : PLC 2 parameter 1	R/W	R/Y	R/Y
LW-10509	(16bit) : PLC 2 parameter 2	R/W	R/Y	R/Y
LW-10510	(16bit) : PLC 3 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10511	(16bit) : PLC 3 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10512	(16bit) : PLC 3 send ACK delay (unit : ms)	R/W	R/Y	R/Y
LW-10513	(16bit) : PLC 3 parameter 1	R/W	R/Y	R/Y
LW-10514	(16bit) : PLC 3 parameter 2	R/W	R/Y	R/Y
LW-10515	(16bit) : PLC 4 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10516	(16bit) : PLC 4 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10517	(16bit) : PLC 4 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10518	(16bit) : PLC 4 parameter 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10519	(16bit) : PLC 4 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10520	(16bit) : PLC 5 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10521	(16bit) : PLC 5 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10522	(16bit) : PLC 5 send ACK delay (unit : ms)			
	(SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10523	(16bit) : PLC 5 parameter 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10524	(16bit) : PLC 5 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10525	(16bit) : PLC 6 timeout (unit : 100ms)	R/W	R/Y	R/Y
		l	1	l



		•		•
LW-10526	(16bit) : PLC 6 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10527	(16bit) : PLC 6 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10528	(16bit) : PLC 6 parameter 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10529	(16bit) : PLC 6 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10530	(16bit) : PLC 7 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10531	(16bit) : PLC 7 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10532	(16bit) : PLC 7 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10533	(16bit) : PLC 7 parameter 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10534	(16bit) : PLC 7 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y
LW-10535	(16bit) : PLC 8 timeout (unit : 100ms)	R/W	R/Y	R/Y
LW-10536	(16bit) : PLC 8 turn around delay (unit : ms)	R/W	R/Y	R/Y
LW-10537	(16bit) : PLC 8 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/Y	R/Y
LW-10538	(16bit) : PLC 8 parameter 1 (SIEMENS S7/400 rack)	R/W	R/Y	R/Y
LW-10539	(16bit) : PLC 8 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/Y	R/Y



22.16 Communication Status with PLC (COM)

		Read(R))/Write(W)/Control(Y)				
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y			
LB-9150	auto. connection for PLC 1 (COM1) (when ON)	R/W	R/Y	R/Y			
LB-9151	auto. connection for PLC 2 (COM2) (when ON)	R/W	R/Y	R/Y			
LB-9152	auto. connection for PLC 3 (COM3) (when ON)	R/W	R/Y	R/Y			
LB-9200	PLC 1 status (SN0, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9201	PLC 1 status (SN1, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9202	PLC 1 status (SN2, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9203	PLC 1 status (SN3, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9204	PLC 1 status (SN4, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9205	PLC 1 status (SN5, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9206	PLC 1 status (SN6, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9207	PLC 1 status (SN7, COM1), set on to retry connection	R/W	R/Y	R/Y			
LB-9500	PLC 2 status (SN0, COM2), set on to retry connection	R/W	R/Y	R/Y			
LB-9501	PLC 2 status (SN1, COM2), set on to retry connection	R/W	R/Y	R/Y			
LB-9502	PLC 2 status (SN2, COM2), set on to retry connection	R/W	R/Y	R/Y			
LB-9503	PLC 2 status (SN3, COM2), set on to retry connection	R/W	R/Y	R/Y			
LB-9504	PLC 2 status (SN4, COM2), set on to retry connection	R/W	R/Y	R/Y			
LB-9505	PLC 2 status (SN5, COM2), set on to retry connection	R/W	R/Y	R/Y			
LB-9506	PLC 2 status (SN6, COM2), set on to retry connection	R/W	R/Y	R/Y			



LB-9507	PLC 2 status (SN7, COM2), set on to retry connection	R/W	R/Y	R/Y
LB-9800	PLC 3 status (SN0, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9801	PLC 3 status (SN1, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9802	PLC 3 status (SN2, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9803	PLC 3 status (SN3, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9804	PLC 3 status (SN4, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9805	PLC 3 status (SN5, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9806	PLC 3 status (SN6, COM3), set on to retry connection	R/W	R/Y	R/Y
LB-9807	PLC 3 status (SN7, COM3), set on to retry connection	R/W	R/Y	R/Y



22.17 Communication Status with PLC (Ethernet)

		Read(R)/Write(W)/Control(Y			
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LB-9153	auto. connection for PLC 4 (ethernet) (when ON)	R/W	R/Y	R/Y	
LB-9154	auto. connection for PLC 5 (ethernet) (when ON)	R/W	R/Y	R/Y	
LB-9155	auto. connection for PLC 6 (ethernet) (when ON)	R/W	R/Y	R/Y	
LB-9156	auto. connection for PLC 7 (ethernet) (when ON)	R/W	R/Y	R/Y	
LB-9157	auto. connection for PLC 8 (ethernet) (when ON)	R/W	R/Y	R/Y	
LB-9158	auto. connection for PLC 9 (ethernet) (when ON)	R/W	R/Y	R/Y	
LB-10070	forced to reconnect PLC 4 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/Y	R/Y	
LB-10071	forced to reconnect PLC 5 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/Y	R/Y	
LB-10072	forced to reconnect PLC 6 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/Y	R/Y	
LB-10073	forced to reconnect PLC 7 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/Y	R/Y	
LB-10074	forced to reconnect PLC 8 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/Y	R/Y	
LB-10075	forced to reconnect PLC 9 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/Y	R/Y	
LB-10100	PLC 4 status (ethernet), set on to retry connection	R/W	R/Y	R/Y	
LB-10400	PLC 5 status (ethernet), set on to retry	R/W	R/Y	R/Y	



	connection			
LB-10700	PLC 6 status (ethernet), set on to retry	R/W	R/Y	R/Y
	connection	R/W	R/ I	R/ I
LB-11000	PLC 7 status (ethernet), set on to retry	R/W	N R/Y	R/Y
	connection			
LB-11300	PLC 8 status (ethernet), set on to retry	R/W	R/Y	R/Y
	connection			
LB-11600	PLC 9 status (ethernet), set on to retry	R/W	R/Y	R/Y
	connection			
LB-11900	PLC 10 status (ethernet), set on to retry	R/W	R/Y	R/Y
	connection			
LB-11901	PLC 11 status (ethernet), set on to retry	R/W	R/Y	R/Y
	connection			
LB-11902	PLC 12 status (ethernet), set on to retry	R/W	R/Y	R/Y
LB-11903	PLC 13 status (ethernet), set on to retry	R/W	R/Y	R/Y
	connection			
LB-11904	PLC 14 status (ethernet), set on to retry	R/W	R/Y	R/Y
LB-11905	Connection PLC 15 status (ethernet), set on to retry			
LD-11905	connection	R/W	R/Y	R/Y
LB-11906	PLC 16 status (ethernet), set on to retry			
	connection	R/W	R/Y	R/Y
LW-9600	(16bit) : PLC 4's IP0 (IP address =			
	IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9601	(16bit) : PLC 4's IP1 (IP address =			
	IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9602	(16bit) : PLC 4's IP2 (IP address =			
	IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9603	(16bit) : PLC 4's IP3 (IP address =	.	5.87	
	IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9604	(16bit) : PLC 4's port no.	R/W	R/Y	R/Y
LW-9605	(16bit) : PLC 5's IP0 (IP address =		DM	DA
	IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9606	(16bit) : PLC 5's IP1 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)	rt/ VV	rs/ f	
LW-9607	(16bit) : PLC 5's IP2 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			



LW-9608	(16bit) : PLC 5's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9609	(16bit) : PLC 5's port no.	R/W	R/Y	R/Y
LW-9610	(16bit) : PLC 6's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9611	(16bit) : PLC 6's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9612	(16bit) : PLC 6's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9613	(16bit) : PLC 6's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9614	(16bit) : PLC 6's port no.	R/W	R/Y	R/Y
LW-9615	(16bit) : PLC 7's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9616	(16bit) : PLC 7's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9617	(16bit) : PLC 7's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9618	(16bit) : PLC 7's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9619	(16bit) : PLC 7's port no.	R/W	R/Y	R/Y
LW-9620	(16bit) : PLC 8's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9621	(16bit) : PLC 8's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9622	(16bit) : PLC 8's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9623	(16bit) : PLC 8's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9624	(16bit) : PLC 8's port no.	R/W	R/Y	R/Y
LW-9625	(16bit) : PLC 9's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9626	(16bit) : PLC 9's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9627	(16bit) : PLC 9's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9628	(16bit) : PLC 9's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
	······································			



LW-9629 (16bit) : PLC 9's port no.	R/W	R/Y	R/Y
------------------------------------	-----	-----	-----



22.18 Communication Status with PLC (USB)

		Read(R)/	Read(R)/Write(W)/Control(Y)		
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LB-9190	auto. connection for PLC (USB 1) (when ON)	R/W	R/Y	R/Y	
LB-9191	PLC status (USB 1), set on to retry connection	R/W	R/Y	R/Y	
LB-9193	auto. connection for PLC (USB 2) (when ON)	R/W	R/Y	R/Y	
LB-9194	PLC status (USB 2), set on to retry connection	R/W	R/Y	R/Y	



22.19 Communication Status with Remote HMI

		Read(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9068	auto. connection for remote HMI 1 (when ON)	R/W	R/Y	R/Y
LB-9069	auto. connection for remote HMI 2 (when ON)	R/W	R/Y	R/Y
LB-9070	auto. connection for remote HMI 3 (when ON)	R/W	R/Y	R/Y
LB-9071	auto. connection for remote HMI 4 (when ON)	R/W	R/Y	R/Y
LB-9072	auto. connection for remote HMI 5 (when ON)	R/W	R/Y	R/Y
LB-9073	auto. connection for remote HMI 6 (when ON)	R/W	R/Y	R/Y
LB-9074	auto. connection for remote HMI 7 (when ON)	R/W	R/Y	R/Y
LB-9075	auto. connection for remote HMI 8 (when ON)	R/W	R/Y	R/Y
LB-9100	remote HMI 1 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9101	remote HMI 2 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9102	remote HMI 3 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9103	remote HMI 4 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9104	remote HMI 5 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9105	remote HMI 6 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9106	remote HMI 7 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9107	remote HMI 8 status (set on to retry connection)	R/W	R/Y	R/Y
LB-9149	forced to reconnect remote HMI when IP changed on-line (set ON)	R/W	R/Y	R/Y
LW-9800	(16bit) : remote HMI 1's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9801	(16bit) : remote HMI 1's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9802	(16bit) : remote HMI 1's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9803	(16bit) : remote HMI 1's IP3 (IP address =	R/W	R/Y	R/Y



	IP0:IP1:IP2:IP3)			
LW-9804	(16bit) : remote HMI 1's port no.	R/W	R/Y	R/Y
LW-9805	(16bit) : remote HMI 2's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9806	(16bit) : remote HMI 2's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9807	(16bit) : remote HMI 2's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9808	(16bit) : remote HMI 2's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9809	(16bit) : remote HMI 2's port no.	R/W	R/Y	R/Y
LW-9810	(16bit) : remote HMI 3's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9811	(16bit) : remote HMI 3's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9812	(16bit) : remote HMI 3's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9813	(16bit) : remote HMI 3's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9814	(16bit) : remote HMI 3's port no.	R/W	R/Y	R/Y
LW-9815	(16bit) : remote HMI 4's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9816	(16bit) : remote HMI 4's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9817	(16bit) : remote HMI 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9818	(16bit) : remote HMI 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9819	(16bit) : remote HMI 4's port no.	R/W	R/Y	R/Y
LW-9820	(16bit) : remote HMI 5's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9821	(16bit) : remote HMI 5's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9822	(16bit) : remote HMI 5's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9823	(16bit) : remote HMI 5's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9824	(16bit) : remote HMI 5's port no.	R/W	R/Y	R/Y



				-
LW-9825	(16bit) : remote HMI 6's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9826	(16bit) : remote HMI 6's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9827	(16bit) : remote HMI 6's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9828	(16bit) : remote HMI 6's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9829	(16bit) : remote HMI 6's port no.	R/W	R/Y	R/Y
LW-9830	(16bit) : remote HMI 7's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9831	(16bit) : remote HMI 7's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9832	(16bit) : remote HMI 7's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9833	(16bit) : remote HMI 7's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9834	(16bit) : remote HMI 7's port no.	R/W	R/Y	R/Y
LW-9835	(16bit) : remote HMI 8's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9836	(16bit) : remote HMI 8's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9837	(16bit) : remote HMI 8's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9838	(16bit) : remote HMI 8's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9839	(16bit) : remote HMI 8's port no.	R/W	R/Y	R/Y
LW-9905	(16bit) : remote HMI 21's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9906	(16bit) : remote HMI 21's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9907	(16bit) : remote HMI 21's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9908	(16bit) : remote HMI 21's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9909	(16bit) : remote HMI 21's port no.	R/W	R/Y	R/Y
LW-9910	(16bit) : remote HMI 22's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y



LW-9911	(16bit) : remote HMI 22's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9912	(16bit) : remote HMI 22's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9913	(16bit) : remote HMI 22's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9914	(16bit) : remote HMI 22's port no.	R/W	R/Y	R/Y
LW-9915	(16bit) : remote HMI 23's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9916	(16bit) : remote HMI 23's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9917	(16bit) : remote HMI 23's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9918	(16bit) : remote HMI 23's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9919	(16bit) : remote HMI 23's port no.	R/W	R/Y	R/Y
LW-9920	(16bit) : remote HMI 24's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9921	(16bit) : remote HMI 24's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9922	(16bit) : remote HMI 24's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9923	(16bit) : remote HMI 24's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9924	(16bit) : remote HMI 24's port no.	R/W	R/Y	R/Y
LW-9925	(16bit) : remote HMI 25's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9926	(16bit) : remote HMI 25's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9927	(16bit) : remote HMI 25's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9928	(16bit) : remote HMI 25's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9929	(16bit) : remote HMI 25's port no.	R/W	R/Y	R/Y
LW-9930	(16bit) : remote HMI 26's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9931	(16bit) : remote HMI 26's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y



(16bit) : remote HMI 26's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 26's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 26's port no.	R/W	R/Y	R/Y
(16bit) : remote HMI 27's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 27's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 27's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 27's port no.	R/W	R/Y	R/Y
(16bit) : remote HMI 28's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 28's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 28's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 28's IP3 (IP address =	R/W	R/Y	R/Y
(16bit) : remote HMI 28's port no.	R/W	R/Y	R/Y
(16bit) : remote HMI 29's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 29's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 29's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 29's port no.	R/W	R/Y	R/Y
(16bit) : remote HMI 30's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 30's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
(16bit) : remote HMI 30's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3) (16bit) : remote HMI 26's IP3 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 27's IP0 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 27's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 27's IP2 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP0 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP2 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 29's IP2 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 29's IP3 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 30's IP0 (IP address = IP0:IP1:IP2:IP3) (16bit) : remote HMI 30's IP1 (IP address = IP0:IP1:IP2:IP3) (16bit) : re	IP0:IP1:IP2:IP3)RW(16bit) : remote HMI 26's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 26's port no.R/W(16bit) : remote HMI 27's IP0 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 27's IP1 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 27's IP2 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 28's IP0 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 28's IP1 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 28's IP2 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 29's IP3 (IP address = IP0:IP1:IP2:IP3)R/W(16bit) : remote HMI 30's IP1 (IP addres	IP0:IP1:IP2:IP3)R.WR.Y(16bit) : remote HMI 26's IP3 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 26's port no.R.WR/Y(16bit) : remote HMI 27's IP0 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 27's IP1 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 27's IP2 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 27's port no.R.WR/Y(16bit) : remote HMI 27's port no.R.WR/Y(16bit) : remote HMI 28's IP0 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 28's IP1 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 28's IP2 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 29's IP0 (IP address = IP0:IP1:IP2:IP3)R.WR/Y(16bit) : remote HMI 29's IP3 (IP address =



LW-9953	(16bit) : remote HMI 30's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9954	(16bit) : remote HMI 30's port no.	R/W	R/Y	R/Y
LW-9955	(16bit) : remote HMI 31's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9956	(16bit) : remote HMI 31's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9957	(16bit) : remote HMI 31's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9958	(16bit) : remote HMI 31's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9959	(16bit) : remote HMI 31's port no.	R/W	R/Y	R/Y
LW-9960	(16bit) : remote HMI 32's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9961	(16bit) : remote HMI 32's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9962	(16bit) : remote HMI 32's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9963	(16bit) : remote HMI 32's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9964	(16bit) : remote HMI 32's port no.	R/W	R/Y	R/Y



22.20 Communication Status with Remote PLC

		Read(R)/	Read(R)/Write(W)/Con		
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LW-10050	(16bit) : IP0 of the HMI connecting to remote PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10051	(16bit) : IP1 of the HMI connecting to remote PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10052	(16bit) : IP2 of the HMI connecting to remote PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10053	(16bit) : IP3 of the HMI connecting to remote PLC 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10054	(16bit) : port no. of the HMI connecting to remote PLC 1	R/W	R/Y	R/Y	
LW-10055	(16bit) : IP0 of the HMI connecting to remote PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10056	(16bit) : IP1 of the HMI connecting to remote PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10057	(16bit) : IP2 of the HMI connecting to remote PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10058	(16bit) : IP3 of the HMI connecting to remote PLC 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10059	(16bit) : port no. of the HMI connecting to remote PLC 2	R/W	R/Y	R/Y	
LW-10060	(16bit) : IP0 of the HMI connecting to remote PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10061	(16bit) : IP1 of the HMI connecting to remote PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10062	(16bit) : IP2 of the HMI connecting to remote PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10063	(16bit) : IP3 of the HMI connecting to remote PLC 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10064	(16bit) : port no. of the HMI connecting to remote PLC 3	R/W	R/Y	R/Y	
LW-10065	(16bit) : IP0 of the HMI connecting to remote PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y	
LW-10066	(16bit) : IP1 of the HMI connecting to remote	R/W	R/Y	R/Y	



	PLC 4 (IP address = IP0:IP1:IP2:IP3)			
LW-10067	(16bit) : IP2 of the HMI connecting to remote			
	PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10068	(16bit) : IP3 of the HMI connecting to remote	DAA		
	PLC 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10069	(16bit) : port no. of the HMI connecting to	R/W	R/Y	R/Y
	remote PLC 4	R/W	R/ 1	K/ I
LW-10300	(16bit) : remote PLC 1's IP0 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10301	(16bit) : remote PLC 1's IP1 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10302	(16bit) : remote PLC 1's IP2 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)	1000		
LW-10303	(16bit) : remote PLC 1's IP3 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10304	(16bit) : remote PLC 1's port no.	R/W	R/Y	R/Y
LW-10305	(16bit) : remote PLC 2's IP0 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10306	(16bit) : remote PLC 2's IP1 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)	10,00		
LW-10307	(16bit) : remote PLC 2's IP2 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10308	(16bit) : remote PLC 2's IP3 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10309	(16bit) : remote PLC 2's port no.	R/W	R/Y	R/Y
LW-10310	(16bit) : remote PLC 3's IP0 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)	1000		
LW-10311	(16bit) : remote PLC 3's IP1 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)	1011		
LW-10312	(16bit) : remote PLC 3's IP2 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10313	(16bit) : remote PLC 3's IP3 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)			
LW-10314	(16bit) : remote PLC 3's port no.	R/W	R/Y	R/Y
LW-10315	(16bit) : remote PLC 4's IP0 (IP address =	R/W	R/Y	R/Y
	IP0:IP1:IP2:IP3)	1.7.4.4		
LW-10316	(16bit) : remote PLC 4's IP1 (IP address =	R/W	R/Y	R/Y



System Reserved Words/Bits

	IP0:IP1:IP2:IP3)			
LW-10317	(16bit) : remote PLC 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10318	(16bit) : remote PLC 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-10319	(16bit) : remote PLC 4's port no.	R/W	R/Y	R/Y



22.21 Communication Error Messages & No. of Pending Cmd.

		Read(R)/Write(W)/Control(
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
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(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9000 ~ LB-9009	initialized as ON	R/W	R/Y	R/Y
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/Y	R/Y
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) *Note1	R/W	R/Y	R/Y
LB-9059	disable MACRO TRACE function (when ON) *Note2	R/W	R/Y	R/Y
LB-9064	enable USB barcode device (disable keyboard) (when ON) *Note3	R/W	R/Y	R
LW-9006	(16bit) : connected client no.	R	R	R
LW-9024	(16bit) : memory link system register	R/W	R/Y	R/Y
LW-9032	(8 words) : folder name of backup history files to SD, USB memory	R/W	R/Y	R/Y
LW-9050	(16bit) : current base window ID	R	R	R
LW-9134	(16bit) : language mode <mark>*Note4</mark>	R/W	R/Y	R/Y
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/Y	R/Y





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.



		Read(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-10069	forced to reconnect remote printer/backup server when IP changed on-line (set ON)	R/W	R/Y	R/Y
LW-9770	(16bit) : remote printer/backup server IP0 (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9771	(16bit) : remote printer/backup server IP1 (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9772	(16bit) : remote printer/backup server IP2 (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9773	(16bit) : remote printer/backup server IP3 (IP0:IP1:IP2:IP3)	R/W	R/Y	R/Y
LW-9774	(6 words) : remote printer/backup server user name*Note1	R/W	R/Y	R/Y
LW-9780	(6 words) : remote printer/backup server password*Note1	R/W	R/Y	R/Y

22.23 Remote Print/Backup Server



1. When change settings using LW-9774 and LW9780, please reboot HMI to enable the new settings.



22.24 EasyAccess

		Read(R)/Wr		V)/Control(Y)	
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LB-9051	disconnect (set OFF)/connect (set ON)	R/W	R/Y	R/Y	
	EasyAccess server				
LB-9052	status of connecting to EasyAccess server	R	R	R	
LB-9196	local HMI supports monitor function only (when ON)	R/W	R/Y	R/Y	
LB-9197	support monitor function only for remote HMIs (when ON)	R/W	R/Y	R/Y	

For further information on EasyAccess, please visit <u>http://www.ihmi.net/</u>.



22.25 Pass-Through Settings

	Read		R)/Write(W)/Control(Y		
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LW-9900	(16bit) : HMI run mode (0 : normal mode, 1~3 :	R/W	R/Y	R/Y	
	test mode (COM 1~COM 3)	1.7.4.4			
LW-9901	(16bit) : pass-through source COM port (1~3 :	R/W	R/Y	R/Y	
	COM 1~COM 3)	17/14			
LW-9902	(16bit) : pass-through destination COM port	R/W	R/Y	R/Y	
	(1~3 : COM 1~COM 3)				



		Read(R)/Write(W)/Control		
Address	dress Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9192	disable USB 1 PLC's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-9195	disable USB 2 PLC's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11960	disable PLC 1's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11961	disable PLC 2's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11962	disable PLC 3's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11963	disable PLC 4's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11964	disable PLC 5's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11965	disable PLC 6's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11966	disable PLC 7's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y
LB-11967	disable PLC 8's "PLC No Response" dialog (when ON)	R/W	R/Y	R/Y

22.26 Disable PLC No Response Dialog Box



22.27 HMI and Project Key

	Read(R)		d(R)/Write(W)/Control(Y)		
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y	
LB-9046	project key is different from HMI key (when ON)	R	R	R	
LW-9046	(32bit) : HMI key (i series only) *Note1	R/W	R/Y	R	



1. When change HMI key using LW-9046, please reboot HMI to enable the new settings.



Please confirm your Internet connection before downloading the demo project.



22.28 Fast Selection Window Control

		Read(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9013	FS window control[hide(ON)/show(OFF)]	R/W	R/Y	R/Y
LB-9014	FS button control[hide(ON)/show(OFF)]	R/W	R/Y	R/Y
LB-9015	FS window/button control[hide(ON)/show(OFF)]	R/W	R/Y	R/Y



22.29 Input Object Function

		Read(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
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



		Read(R)/	Write(W)/C	ontrol(Y)
Address	Description	Local HMI	MACRO R/Y	Remote HMI R/Y
LB-9044	disable remote control (when ON)	R/W	R/Y	R/Y
LB-9053	prohibit password remote-read operation (when ON)	R/W	R/Y	R/Y
LB-9054	prohibit password remote-write operation (when ON)	R/W	R/Y	R/Y
LB-9196	local HMI supports monitor function only (when ON)	R/W	R/Y	R/Y
LB-9197	support monitor function only for remote HMIs (when ON)	R/W	R/Y	R/Y
LB-9198	disable local HMI to trigger a MACRO (when ON)	R/W	R/Y	R/Y
LB-9199	disable remote HMI to trigger a MACRO (when ON)	R/W	R/Y	R/Y

22.30 Local/Remote Operation Restrictions



Chapter 23 HMI Supported Printers

23.1 The Supported Printer Types

HMI supported printer drivers include the following types:



EPSON compatible serial printers, please configure communication parameters to match the printer. The EPSON ESC/P2 printer protocol is used. Impact Printer: LQ-300, LQ-300+, LQ-300K+ (RS232) LQ-300+II (RS232) Inkjet Printer: Stylus Photo 750 Laser Printer: EPL-5800

HP PCL Series (USB)



HP compatible USB printers that support HP PCL5 level 3 protocol.

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.

Please check if HP printer supports PCL5 before connecting with HMI, otherwise HMI black screen may occur.



HMI Supported Printers



Serial printers, please configure communication parameters to match the printer. The Pixels of Width must be correctly set and can't exceed printer default setting: 100 pixels for 1610 220 pixels for 2407, 4004 **EPSON ESC Protocol Serial** Micro Printer: SIUPO (Beijing) http://www.siupo.com SP-M, D, E, F Series SP-E1610SK (paper width 45mm), SP-E400-4S (paper width 57.5mm) Recommended SP printer type

for customers outside China.



Micro printer from France connects via serial port; please configure communication parameters to match the printer.

EPSON TM-L90



Serial printers, please configure communication parameters to match the printer. The **Pixels of Width** must be correctly set and can't exceed printer default setting "576":



Serial printers, please configure communication parameters to match the printer. The **Pixels of Width** must be correctly set and can't exceed printer default setting "100":

SP-DN40SH Dot Matrix Printer SP-RMDIII40SH Thermal



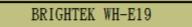
HMI Supported Printers



Use EasyPrinter to start printing for the printers connected with PC via Ethernet. This works under MS Windows so the most printers on market are supported.



Serial printers, please configure communication parameters to match the printer. The paper cutting mode can be selected: [No cut], [Half cut], and [Full cut].





Serial printers, please configure communication parameters to match the printer.



23.2 How to Add a New Printer and Start Printing 23.2.1 Add Printer Type

[System Parameter Settings] / [Model] select printer type and set relevant parameters.

tem Parameter S	ettings					
Font		Extended Memory		Printer/Backu	p Server	
Device	Model	General	System Se		Security	
HMI model : HMI station no : Port no. :	0	I/MT8070iH/MT6100i/M	IT8100i/WT3010 ODBUS server's p		~	
Timer Clock source :						
Printer Type : COM : Baud rate : Parity : Pixels of width :	HP PCL Se Axiohm A	E, F C/P2 Series sries (USB) 630 DIII, DIV, D5, D6, A, I 1-L90	DN, T) bits :	8 Bits 1 Bit 100%	>	To connect Remote Printer Server, go [System Parameter
	BRIGHTEN	(WH-C1/C2 Hs (for 1610 type) or 2) rameter Set	Settings] /
		ОК	Cancel	Device		Model General System Setting Security Extended Memory Printer/Backup Server
rameters o	-	Server], an	id set	Use I Note: U Output s	MT Remote Prin se EasyPinter t ettings Prientation : (Printer/Backup Server er to configure PC for printing screen hardcopy and storing backup data. Image: Imag
				ä		ngs - 0 . 0 . 0 . 0 - 8005 - 111111
						OK Cancel Help





23.2.2 Start Printing

Function Key Object		
neral Security Shape Label		
Description :		
Activate after button is released		
Change full-screen window O Change common Display popup window	vindow	
O Return to previous window O Close window		
ASCII/UNICODE mode	0	
O[Enter] O[Backspace] O[Clear]	O[Esc]	
O[ASCII]/[UNICODE]		
◯ Execute macro	Or, use	PLC
🔘 Window title bar	Control [Sc	
Hard copy screen to USB disk, SD card or printer	hardcopy]	
Screen hard copy Printer : SP-M, D, E, F		
Rotate image 90 degrees Mode : grayscale	printing by	-
Notification	bit registers	S.
	PLC Control	
OK Cancel	Description :	
	PLC name : Local HMI	
	Attribute	
	Type of control : Screen hardcopy	
	Active only when designated window op	ened
		age 90 degrees
	Trigger address	
	PLC name : Local HMI	Setting,.
	Address : LB	0
		Trigger mode : OFF->ON
	Source window for print	
	100000000	
	Current base window 💿 Window no. f	rom register O Designate window
	O Current base window O Window no. f PLC name : Local HMI Address : LW	Setting



Chapter 24 Recipe Editor

24.1 Introduction

Recipe Editor is used to create, view, and edit Recipe (*.rcp) and EMI (*.emi) files for HMI. Open Project Manager and click [Recipe/Extended Memory Editor].

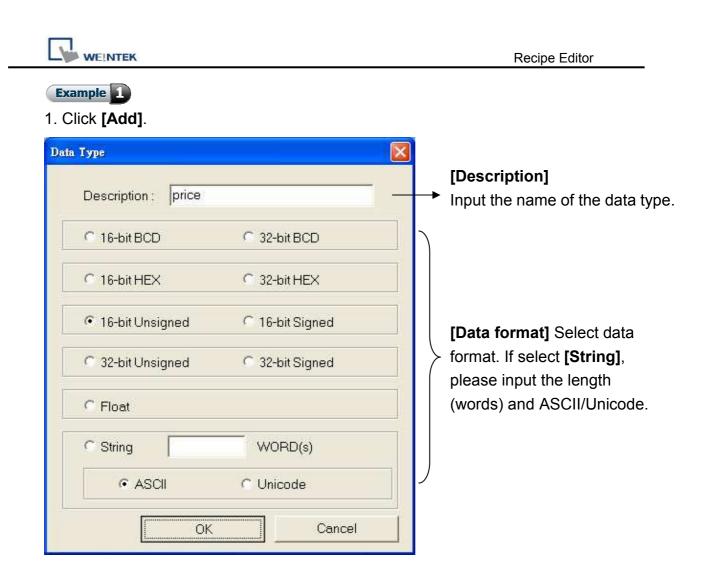


24.2 Recipe Editor Setting How to add new *.rcp / *.emi files? Set Address Range -> Select Data Format

[Select your data format]

Save the specified data format for next time loading. The saved file name: "dataEX.fmt" under EasyBuilder Pro installation directory.

	_		/	
	Set Data Format			
[Address range] Fill in address range,	Address range		Select your data format	
the unit is "word".	Data format		Save Format	Delete Format
	Size	Туре	Description	Add
[Data format] Edit new data format in this field.	1 WORD 10 WORDs 2 WORDs 10 WORDs	16-bit Unsigned String [ASCII] 32-bit Unsigned String [ASCII]	recipe no. product price barcode	Delete Clear All
				Modify
	<	- Internet)	Ok Cancel



2. After setting, click **[OK]** to start editing recipe data.

)ata format	то 2300	Save Format De	elete Format
Size	Туре	Description	Add
1 WORD 10 WORDs 2 WORDs 10 WORDs	16-bit Unsigned String [ASCII] 32-bit Unsigned String [ASCII]	recipe no. product price barcode	Delete Clear All
			Modify
			ОК

e	<u>E</u> dit <u>V</u> iew	Window	<u>H</u> elp			
3	🗃 🖬 X	Pa 💼	3			
ID	ADDRESS	recipe no.	product	price	barcode	
0	0	0	bread	13	1547965231	
1	23	1	coke	20	1437286591	
2	46	2	coffee	245	3265479210	
3	69	3	chair	599	6454789321	
4	92	4	desk	750	1464545441	
5	115	5	tea	15	1234534444	
6	138	6	egg	8	4568974164	
7	161	7	coke	28	1246634344	
8	184	8	coke	29	4644245645	
9	207	9	lemon	35	4645244575	
10	230	10	tea bag	68	5676454567	
11	253	11	book	245	4564676454	
12	276	0	0.000013	0		
13	299	0		0		
14	322	0		0		

3. In this example, the total length of data format is 23 words and will be seen as one set of recipe data.

The first set: "recipe no." = address 0, "product" = address 1 ~ 10, "price" = address $11 \sim 12$, "barcode" = address $13 \sim 22$;

The second set: "recipe no." = address 23, "product" = address 24 ~ 33, "price" = address 34 ~ 35, "barcode" = address 36 ~ 45...and so on.



■ After editing recipe data, it can be saved as *.rcp, *.emi, or *.csv. *.rcp can be downloaded to HMI using Project Manager or external devices (USB disk or SD card). *.emi can be saved directly to external device and insert to HMI for reading (EM register).



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.

25.1 How to Export to Excel

- 1. [EasyConverter] / [Open] / [OK]
- 2. Click [Export to Microsoft Excel]

ct number of digi	Туре	Word Size	Digito	čcalina		-xcc (3.5-50,0350)	× • *	
	19pe 16-bit Unsigne		Digits 0	No 💌		reation til	mej 10:51:06 2	0000
o-pir o usigner	To-bit offstglie	4	U.	NO _	22.8	nu Nov 27)ata]	10:51:06 2	008
							bit Unsigned	.
						0:51:06,0	A CONTRACT OF A DESCRIPTION OF A DESCRIP	
					10 March 10	0:51:07,0		
					1	0:51:08,1		
					01/02/	0:51:23,0		
					010201	0:51:24,0		
					100	0:51:25,1		
	100							
					1	0:51:26,1		
ling & Offset					ר 1	0:51:26,1 0:51:27,2		
ing & Offset					ר 1	0:51:26,1	۰,	
					ר 1	0:51:26,1 0:51:27,2	в	с
					ר 1	0:51:26,1 0:51:27,2 0:51:28,3	в	с
ng & Offset					1	0:51:26,1 0:51:27,2 0:51:28,3 A [Creation]	в	
					1	0:51:26,1 0:51:27,2 0:51:28,3 A [Creation]	B time]	с 2008
etting)					1 1 1 2	0:51:26,1 0:51:27,2 0:51:28,3 A [Creation Thu Nov 2	B time]	:008
etting)		OK	Ca	incel	1 1 1 2 3	0: 51: 26, 1 0: 51: 27, 2 0: 51: 28, 3 A [Creation Thu Nov 2 [Data]	B time] 27 10:51:06 2	:008
etting)		OK	Са		1 1 1 2 3 4	0:51:26,1 0:51:27,2 0:51:28,3 A [Creation Thu Nov 2 [Data] "Time"	B time] 27 10:51:06 2 "16-bit Unsi	:008
g & Offset ietting) ietting)		OK	Ce		1 1 1 2 3 4 5	0: 51: 26, 1 0: 51: 27, 2 0: 51: 28, 3 Creation Thu Nov 2 [Data] "Time" 10:51:06	B time] 27 10:51:06 2 "16-bit Unsi 0	:008

9 10:51:24

10 10:51:25

11 10:51:26

0

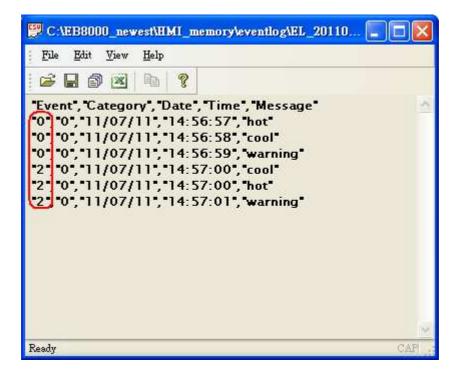
1

1



When converting event log into Excel, users can find an **[Event]** field in EasyConverter as below.

0 -> Event triggered; 1 -> Event acknowledged; 2 -> Event returns to normal





25.2 How to Use Scaling Function

Scaling is utilized to offset data:

new value = [(value + A) x B] + C, users can set values of A, B, and C.

A: lower limit of the value ; C: engineering low

B: [(engineering high) - (engineering low) / (upper limit) - (lower limit)]

Sele	ct number of digi	is after decimal p	ooint:		\bigcap
No	Name	Туре	Nord Size	Digit	Scaling
1	16-bit Unsigned	16-bit Unsigned	1	0	No 💙
					No Yes
				3	6

Example 1

For example, here is a voltage data with a format of 16-bit unsigned (range: $0 \sim 4096$).

If users want to convert those data to volt range form -5V to +5V:

new value = $[(value + 0) \times 0.0024] + (-5):$

Before:	After:
K.11/20081203.dtl - EasyConvert Edit Yiew Options Help Edit Yiew Options Help Wed Dec 03 08:47:15 2008 [Data] Time", "16-bit Unsigned" 08:47:16 0.000 08:47:17 300.000 08:47:20 1200.000 08:47:21 1500.000 08:47:21 1800.000 08:47:22 1800.000 08:47:22 1800.000 08:47:22 100.000 08:47:23 2100.000 08:47:25 2700.000 08:47:24 2400.000 08:47:25 2700.000 08:47:25 2700.000 08:47:28 3600.000 08:47:29 3900.000 08:47:31 3796.000 08:47:31 3796.000 08:47:33 3196.000 08:47:32 296.000 08:47:35 2596.000	Ele Edit View Options Help Ele Edit View Options Help Image: Contract of the state of



o Name	Туре	Word Size	Digits	Scaling	
1 16-bit U	nsigned 16-bit Uns	i; 1	3	Yes 💌	Settings of data
					above can be saved
					as a sample and loaded next time.
					The file name of the sample:
					*.LGS.
					After setting the values for
					Scaling, click [Save Setting]
					-
Scaling & Of	iset	-		_	Scaling, click [Save Setting] In a new sample, click [Load
A	B		с		Scaling, click [Save Setting] In a new sample, click [Load Setting] to use the sample saved
	В		C -5.000	00	Scaling, click [Save Setting] In a new sample, click [Load
A 0.000	B	24		00	Scaling, click [Save Setting] In a new sample, click [Load Setting] to use the sample saved



25.3 How to Use Multi-File Conversion

Example 1

1. Click [File] / [Multi-File] / [Add File] to combine multiple added files into one Excel file.

2. Click [Combine to a file], files will be separated into sheets of one EXCEL

(*.XLS) file labeled with the dated it is created. If users don't check this box, the files will be exported to Excel individually.

Multi-File		A	В	С	D
	7	##########	11:02:32	620	0
Convert file list:	8	##########	11:32:33	680	0
C:\EB8000\eng1.34\datalog\123\20081127.dtl C:\EB8000\eng1.34\datalog\trend\20081128.dtl	9	###########	11:32:34	680	0
	10	###########	11:32:35	680	0
	11	#######################################	11:32:36	680	0
	12	##########	11:32:37	680	0
	B	##########	11:32:38	680	0
	14	##########	11:32:39	680	0
	15	##########	11:32:40	680	0
Add File Delete File	16	###########	11:32:41	680	0
Enable Setting file	17	##########	11:32:42	700	0
	18	##########	11:32:43	680	0
Combine to a file	19	##########	11:32:44	680	0
C:\Documents and Settings\user\test.xls	20	###########	11:32:45	- 680	0
Cancel	I4 - •	• • • • • 200	<u>)81127</u> 020	081128/	
	就編	2			

The saved setting files can be loaded for combining:

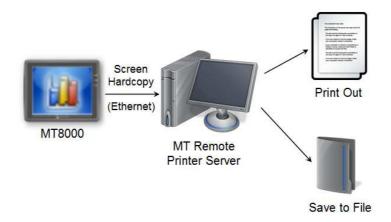
Check both **[Enable Setting file]** and **[Combine to a file]** boxes and select the files to be combined then click **[OK]**.

Convert file list:			
C:\Documents and Settings\user\20081127. C:\Documents and Settings\user\20081128.	dti dti		
✓ Enable Setting file		Add File	Delete File
Enable Setting file C:\EB8000\datalog\pressure\1.lgs		Add File	Delete File
		Add File	Delete File



Chapter 26 EasyPrinter

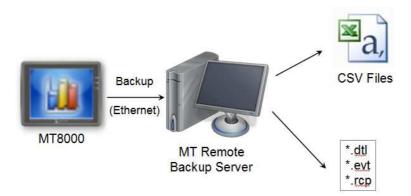
EasyPrinter is a Win32 application and can only run on MS Windows 2000 / XP / Vista / 7. It enables MT8000 Series to output screen hardcopies to a remote PC via Ethernet. Please see the following illustration:



Here are some advantages of using EasyPrinter:

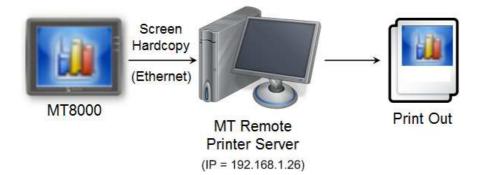
- EasyPrinter provides two modes of hardcopy output: Print-Out and Save-to-File. Users can use either way or both ways.
- Since EasyPrinter is running on MS Windows system, it supports most of the printers available in the market.
- Multiple MT8000 HMI can share one printer via EasyPrinter. Users don't 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:

General	Server	
Hardcopy	Port number of the server socket:	8005
Backup	User name: [Max. length = 12 characters]	admin
Баскир	<u>P</u> assword: [Max. length = 12 characters]	111111
	Naming Convention for HMI Folder (when w	
	Use IP address	
	⊚ Use IP address ⊘ Use HMI name (assign HMI name by L W	19032~L W9039)
	 ● Use IP address ● Use HMI name (assign HMI name by LW Prefix: 	19032~L W9039)
	 Use IP address Use HMI name (assign HMI name by L W Prefix: IP	19032~L W9039)

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:

General	Output	
Hardcopy	Print out to:	
191 B	hp LaserJet 3380 PCL 5	•
Backup	Save to files in:	
	D:\MT8000	

- 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:

Device	Model	General	System Setting	Security
Font	E	xtended Memory	Printer/B	ackup Server
	ote Printer/Backup nter to configure P		hardcopy and storing ba	ckup data.
	m : 💿 Horizonta		cal 🛛	
Printer siz	ze 🗄 💿 Original s	ize 🚫 Fit to	printer margins	
Marş	gin :	15 🗢 m	m	
	15) mm	15 🔹 mm	
Communication s	ettings			
IP addre	ss : 192 . 10	68 . 1 . 26		
P	ort: 8005		Λ÷	
User nai	ne : admin			
	rd : 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].

Hard copy screen to USB disk or	printer		_
Screen hard copy	Printer :	MT Remote Printer Server	-
		MT Remote Printer Server	
		USB disk 1	
- Notification		USB disk 2	

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

5.	Users can also use a [PLC Control] object to make screen
	hardcopies.
6.	Users cannot print alarm information via EasyPrinter.
7.	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:

General	Server	
Hardcopy	Port number of the server socket:	
	User name: [Max. length = 12 characters] admin	
Backup	Password: [Max. length = 12 characters] 111111	
	Naming Convention for HMI Folder (when w	177 - 187
	Ose IP address	177 - 187
	⊚ Use IP address ⊘ Use HMI name (assign HMI name by L W	79032~L W9039)
	 ● Use IP address ● Use HMI name (assign HMI name by LW Prefix: 	79032~L W9039)
	 O Use IP address O Use HMI name (assign HMI name by L W Prefix: IP(Ex: IP_192.168.1 	79032~L W9039)

- 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 [Backup] tab on the left side in the dialogue box as follows:

General	Output
lardcopy	Backup files in:
aracopy	D:\MT8000
Backup	
	When target file has existed:
	Overwrite it. (The content will be destroyed)
	Append .BAK to the file name.
	Convert Batch File
	Enable
	D.\MT8000\convert2csy.def

- 4. In **[Output]**, click the **(i)** button to browse and select a path for storage of the incoming history files.
- 5. Click **[OK]** to apply the settings.
- 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:

WEINTEK

Device	Model	General	System Setting	Security
Font	E	ktended Memory	Printer/Bac	kup Server
utput settings Orientation	ter to configure P : ③ Horizonta : ③ Original s	C for printing screen	hardcopy and storing back ical printer margins m	up data.
communication set	L			
IP address	: 192 . 16	58 . 1 . 26		
	C DOOD		10s	
Por	t: 8005			

 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:

WEINTEK

eneral Secu	rity Shape 1	Label	
Descri	ption :		
Source			
ORW	○RW_A	💿 Historical event log	🚫 Historical data log
Backup pos	ition		
OUSB 1	OUSB 2	Remote printer/backup s	erver
		Satemore printer backap c	
Note : Use		to change the backup folder nar	
Note : Use	LW9032~9039 Remote printer		ne.
Note : Use	LW9032~9039 Remote printer	to change the backup folder nar /backup server] to store data to :	ne.
Note : Use	LW9032~9039 Remote printer	to change the backup folder nar /backup server] to store data to :	ne.
Note : Use in [System	LW9032~9039 Remote printer	to change the backup folder nav /backup server] to store data to s ter/Backup Server] settings.	ne.
Note : Use in [System Range	: LW9032~9039 : [Remote printer 1 Parameter][Prin	to change the backup folder nav /backup server] to store data to s ter/Backup Server] settings.	ne.

- 8. In [Source], select [Historical event log].
- 9. In [Backup position], select [Remote printer server].
- 10. In [Range], select [Today] and [All].
- 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.
- [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.

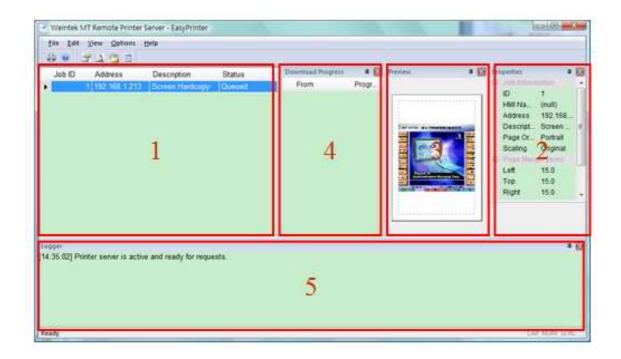
NOTE 8. The **[Backup]** object can be triggered via a bit signal.

9. 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	Job List	This window lists all incoming tasks, i.e. screen
		hardcopy and backup requests.
2	Droporty Window	This window shows the information about the
2	Property Window	task selected from "Job List."
3	Due in Minde	This window shows the preview image of the
S S	Preview Window	screen hardcopy task selected from "Job List."
4	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 \rightarrow File	Description
Enable Output	 Selected
	EasyPrinter processes the tasks one by one.
	 Unselected
	EasyPrinter arranges the incoming tasks in memory.

NOTE	10. 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 Eit To Margin
	<u>T</u> op: 15 <u>B</u> ottom: 15
	Users can freely change the properties of [Orientation],
	[Scaling] and [Margins] here.
Delete	To delete the selected tasks permanently.
Select All	To select all tasks from "Job List."

NOTE

11. The backup task is not editable.

- 12. [Edit] is available only when a task is selected.
- 13. **[Delete]** is available when at least one task is selected.

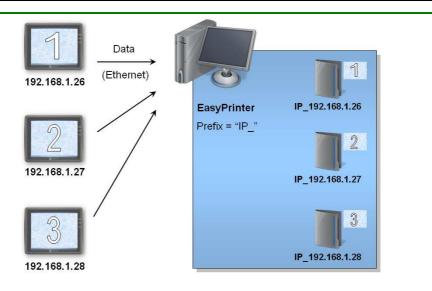


Menu → View	Description
Properties Bar	To show or hide the Property Window.
Preview Bar	To show or hide the Preview Window.
Download Bar	To show or hide the Download Progress Window.
Logger Bar	To show or hide the Message Window.

NOTE	show dowr	nload p	rogres	s by clicking	isers can select the mode the header of the following illustration:	e to
	Download Progres	s Progr		itage Display ength Display		
	,			•	0 messages in Message n, the oldest message will	be

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:	8005
Backup	User name: [Max. length = 12 characters]	admin
	Password: [Max. length = 12 characters]	111111
	Naming Convention for HMI Folder (when wr	riting files)
	Use IP address	
	🔿 Use HMI name (assign HMI name by L WS	9032~L W903
	Prefix:	
	IP(Ex: IP_192.168.1.2	25)
	Properties	
	Minimize to system tray	
	Detailed message	
et the Et	→ [Port number of the server hernet socket number for HMI	to conr
Set the Et range goe	hernet socket number for HMI s from 1 to 65535 and 8005 is	to conr the de
Set the Et range goe	hernet socket number for HMI	to con the de
Set the Et range goe [Server] -	hernet socket number for HMI s from 1 to 65535 and 8005 is	to con the de
Set the Et range goe [Server] - Set the us	hernet socket number for HMI es from 1 to 65535 and 8005 is → [User name] & [Password]	to conr the de trict tha
Set the Et range goe [Server] - Set the us authorized	hernet socket number for HMI es from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea	to con the de trict tha
Set the Et range goe [Server] - Set the us authorized [Naming	hernet socket number for HMI es from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder]	to con the de trict tha asyPrin
Set the Et range goe [Server] - Set the us authorized [Naming EasyPrint	hernet socket number for HMI es from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder] er creates different folders to s	to con the de trict tha asyPrin
Set the Et range goe [Server] - Set the us authorized [Naming EasyPrint hardcopy	hernet socket number for HMI es from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder] er creates different folders to s bitmap files, backup files) from	to conr the de trict that asyPrint store file n differe
Set the Et range goe [Server] - Set the us authorized [Naming EasyPrint hardcopy There are	hernet socket number for HMI s from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder] er creates different folders to s bitmap files, backup files) from two ways to name the folders:	to con the de trict tha asyPrint
Set the Et range goe [Server] - Set the us authorized [Naming EasyPrint hardcopy There are a. Use IP	hernet socket number for HMI s from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder] er creates different folders to s bitmap files, backup files) from two ways to name the folders: address	to con the de trict that asyPrin store file differe
Set the Et range goe [Server] - Set the us authorized [Naming EasyPrint hardcopy There are a. Use IP EasyPrint	hernet socket number for HMI s from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder] er creates different folders to s bitmap files, backup files) from two ways to name the folders: address er names the folder after the IF	to con the de trict that asyPrin store file differe P addre
Set the Et range goe [Server] - Set the us authorized [Naming EasyPrint hardcopy There are a. Use IP EasyPrint HMI send	hernet socket number for HMI s from 1 to 65535 and 8005 is → [User name] & [Password] ser name and password to rest d HMI can send requests to Ea Convention for HMI Folder] er creates different folders to s bitmap files, backup files) from two ways to name the folders: address	to con the de trict that asyPrin store fil differe : P addre



b. Use HMI name

WEINTEK

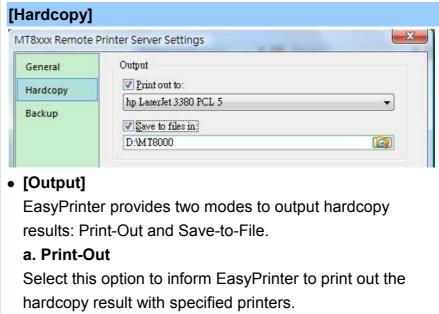
EasyPrinter names the folder after the name of the HMI sending the request. (i.e. [Prefix] + [HMI name])

• [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.



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.

	Printer Server Settings
General	Output
Hardcopy	Backup files in:
Backup	D:\MT8000
	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 C
[Output] EasyPrinte	er stores the backup files to the specified p
	er stores the backup files to the specified p
EasyPrinte	er stores the backup files to the specified p Log historical data files:
EasyPrinte	Log historical data files:
EasyPrinte For Event- [Specified	Log historical data files:
EasyPrinte For Event- [Specified	Path] →
EasyPrinte For Event- [Specified	Log historical data files: Path] → Folder] → [eventlog] →
EasyPrinte For Event- [Specified	Log historical data files: Path] → Folder] →
EasyPrinte For Event- [Specified [HMI	Log historical data files: Path] → Folder] → [eventlog] →
EasyPrinte For Event- [Specified [HMI	Log historical data files: Path] → Folder] → [eventlog] → EL_yyyymmdd.evt Sampling historical data file:



 $[datalog] \rightarrow$ [Folder name of the Data-Sampling object]→ yyyymmdd.dtl For Recipe: [Specified Path] → [HMI Folder] → [recipe] \rightarrow 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 16. Users can assign HMI names from LW9032 to LW9039.

17. 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.



Convert Batch File + EasyConverter

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

20. 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

- 21. EasyPrinter interprets the Convert Batch File on a line basis, i.e. each line forms a criterion.
- 22. Any two arguments should be separated by a comma.
- 23. Every argument should be put in double quotes.
- 24. Do not put any comma inside an argument.
- 25. 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 5th criterion can only be performed on the history files uploaded from the **[Data Sampling]** objects with the folder name "Voltage". The 3rd argument ("*") indicates this criterion accepts the qualified Data-Sampling files from any HMI. Users can also change the 3rd 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	 If the file type is "dtl"
		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

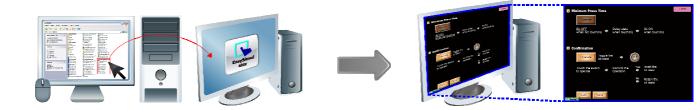
EasyPrinter examines criteria in ascending order every time a file is uploaded. Once the file is qualified for a criterion, it stops the examination and starts over for next file. Therefore, users should place the criteria with more specification upward in the Convert Batch File and place the less-specific 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)"

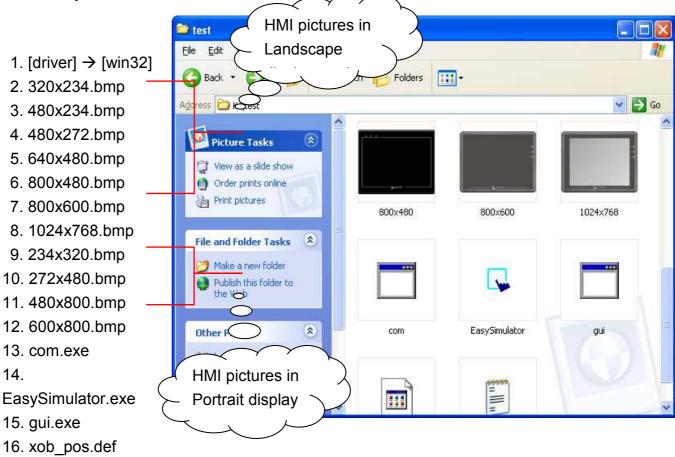


Chapter 27 EasySimulator

EasySimulator enables users to perform On-line/Off-line Simulation without installing EasyBuilder Pro software. To achieve that, users have to prepare the following files in one folder.



27.1 Prepare Needed Files









■ Users can find all the above files in EasyBuilder Pro installation directory, which means users have to install EasyBuilder Pro

software package on a PC first then copy the needed 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.

🖡 xob_pos - Notepad	
<u>File E</u> dit F <u>o</u> rmat <u>V</u> iew <u>H</u> elp	
"O" "D:\test" "D:\EasyAccess.xob"	operation mode 0: off-line 1: on-line define the directory of com.exe and gui.exe define the directory of xob file
	4
٢	

Line no.	Description
1	["0"] Perform Off-line Simulation; ["1"] Perform On-line Simulation
2	Specify the full path where the files locate. (e.g. com.exe, gui.exe,
2	EasySimulator.exeetc.)
3	Specify the full path of the project file. (*.xob)

Step 2

Double click on EasySimulator.exe to start simulation.



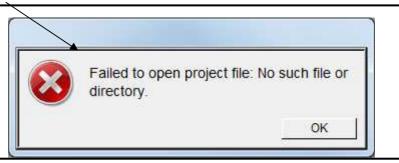
Step 3

On-line / Solution Simulation is displayed on the screen.



If EasySimulator.exe can't be activated, please check if the relevant directories are correctly defined.

If the window below is shown, it indicates there's an error in *.xob file directory, please check if it is correctly defined.





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:

The 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.

The following are examples of how to use EasyBuilder Pro to create projects used on HMI 1(Master) and HMI 2 (Slave).



28.1 How to Create a Project of Master HMI

The following is the project content of HMI 1 in [System Parameter Settings] / [Device].

Font		Extended 1	Memory		Printer/Back	up Serve	r
Device	Model	G	eneral	Syster	n Setting	Sec	urity
No.	Name	Location	Device type	в	Interface	1	I/F Pro
			a function to the second second second		CT111020110361646;		Territoria Constanti
Local HMI	Local HMI	Local	MT6070iH/I	MT8070	Disable		N/A
	Local HMI FATEK FB Series	Local Local	MT6070iH/I FATEK FB :		Disable COM1 (9600,	E,7,1)	N/A RS232

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 parameters must be set correctly.

2. Due to COM 2 of HMI 1 is used to receive commands from HMI 2; a new device must be added– **[Master-Slave Server]** for setting communication properties of COM 2. The picture above shows the parameters of COM 2-"115200, E, 8, 1", and uses RS232. These parameters are not required to be the same as PLC, but the **[Data bits]** must be set to **8**. In general, a higher baud rate for COM 2 is recommended for a more efficient communication with PLC.

28.2 How to Create a Project of Slave HMI

The project content of HMI 2 in [System Parameter Settings] / [Device].

Font	I	Extended Memory	Printer/Ba	ckup Server
Device	Model	General	System Setting	Security
Device fist.				
	Name	Location	Device type	Interface
Device list : No.	Name	Location	Device type	Interface
	Name Local HMI	Location	Device type MT6070iH/MT8070	Interface Disable

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:

1. Create a new device"FATEK FB Series". [PLC default station no.] must be the same as the connected PLC.

PLC type: FATEK FB Series	~
V.1.10, FATEK_FB.∞	
PLC I/F : RS-232	PLC default station no 1

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 - [115200, E, 8, 1].

PLC type :	FATEK FB Series				
	V.1.10, FATEK_FB.so				
PLC I/F 🤇	RS-232	С	OM Port Settin	gs	
COM	СОМ1				
~	Use broadcast comm		COM :	COM 1	*
			Baud rate :	115200	*
			Data bits :	8 Bits	~
Interv	al of block pack (word/		Parity :	Even	*
Max. rea	id-command size (word		Stop bits :	1 Bit	~
11	<u>//</u> 1				



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).

O HMI	
PLC type : FATE	IP Address Settings
V.1.10	O Ethernet O COM port (use master-slave protocol)
PLC I/F : RS-23	
COM : COMI	
Uæ	OK Cancel

Device list :

No.	Name	Location	Device type	Interface
Local HMI	Local HMI	Local	MT6056T/MT8056T	Disable
Remote PLC 1	FATEK FB Series	COM 1 (master-slave mode)	FATEK FB Series	COM1(11520

4. Upon completion of the settings, users can find a new device named **[*Remote PLC 1]** in the **[Device List]**. This device has a "*" symbol, which means, even if it contains "Remote" in the name, it actually gives commands and gets replies through a local COM port, and therefore the connection with PLC can be viewed form a local system reserved register, that is, [*Remote PLC 1], [*Remote PLC 2], [*Remote PLC 3] and [Local PLC 1], [Local PLC 2], [Local PLC 3] use the same system reserved register from the listed below:

Tag	Description
	When ON, auto. connection with PLC (COM 1) when
LB-9150	disconnected.
	When OFF, ignore disconnection with PLC.
	When ON, auto. connection with PLC (COM 2) when
LB-9151	disconnected.
	When OFF, ignore disconnection with PLC.
	When ON, auto. connection with PLC (COM 3) when
LB-9152	disconnected.
	When OFF, ignore disconnection with PLC.

	These local registers indicate the connection states with PLC
	(through COM1).
	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 with MT500 Project of Slave HMI

It allows MT500 using EB500 Master-Slave protocol to read EB Pro HMI Local Data and the data of the connected PLC.

EB Pro Settings

1. Select	System Parameter Settings			X
"Master-Slave	Font	Extended Memory	Printer/Backt	up Server
	Device Model	General	System Setting	Security
Server" driver and	Device list :			
click [Settings] . If	No. Name	Location Device type	Interface	I/F Protoc
a PLC is	Local HMI Local HMI Local Server Master-Slave Se.	Local MT6070iH/MT		N/A 9.11 DS222
connected, follow	Local Server Intester-Stave Se.	Theat Master-stave s	COMI(115200,E,	0,1) [5:0222
the original settings.	New Du	elete Settings		
0. Salaat IBS 2221	Device Properties	, and the second se		
2. Select [RS-232] ,		ive Server		
click [Settings].	OHMI	⊙PLC		
	Location : Local	Settings		
	PLC type : Master-Sla	ave Server		~
		ASTER_SLAVE.so		
	1			
	PLC I/F : RS-232	×		
	COM : COM1 (11	5200,E,8,1)		Settings
3. Fill in MT500				
PLC ID No. in	COM Port Settings			
[Parameter 1]			2 A 1	
	COM : COM 1	×	Timeout (sec) : 1.0	
(Refer to MT500	Baud rate : 115200	Turn	around delay (ms) : 0	
settings).	Data bits : 8 Bits	Ser Ser	nd ACK delay (ms) : 0	
	Parity : Even	~	Parameter 1 : 10	
	Stop bits : 1 Bit	~	Parameter 2 : 0	
			Parameter 3 : 0	
			ок	Cancel

■ EB500 Settings

WE!NTEK

1. In EB500 System Parameter Settings, set Multiple HMI: Slave, HMI-HMI link speed: 115200

 [Baud rate] must be the same in EB500 and EB Pro.

Double click on PLC
 Address View.exe to check
 PLC ID No. and fill in
 [Parameter 1] of EB Pro.

3. Connect COM ports RS232 of each HMI, the

LC General Indic	ator Security	Editor Hardw	are Aux.		
20535 00		10 W1000			
PLC type :	MITSUBISHI F>	KOn/FX2	~		
HMI model : [MT510T/MT50	8T (640 x 480)	~		
PLC I/F port :	RS-485 4W	~	Baud rate :	9600	5
Data bits :	7 Bits	V	Parity :	Even	1.
Stop bits :	1 Bit	~			
10 A			-	_	
Parameter 1 : [8		Turn around delay : [0	
Parameter 3 :	0		Parameter 4 :	Û	
Parameter 5 :	0		Parameter 6 :	0	
HMI station no. : [0	•	PLC station no. ; [0	
Multiple HMI :	Slave) (HMI-HMI link speed :	115200	5
Connect I/F :	Serial				
Lanal ID	address : 👔				
Locality	address : 0	. 0 .			
Server IP	address : 👔	0	0 . 0		
Subnetwo	ırk mask : 👔	<u>.</u>	0.0		
	Sector Sector				
Default route IP PLC time out const					8
PLC time out consta			PLC block pack : [3 pply]	Help
PLC time out consta LCAddressView			PLC block pack : [10
			PLC block pack : [oply)	10
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID	ant (sec) : 3.0	OK (PLC block pack : [oply) Max	Help Min
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID	ant (sec) : 3.0	OK (PLC block pack : [Cancel Addressing Format	Max 99999	Help Min
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID	Bit/Word PLC ID=10 Bit(HMI) Bit(PLC)	OK (Address Type LB X	PLC block pack : [Cancel Addressing Format ddd	Max 99999 377	Min 0
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID	ant (sec) : 3.0	OK (PLC block pack : [Cancel Addressing Format	Max 99999	Help Min
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID	Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC)	Address Type LB X Y M T	Addressing Format	Max 99999 377 377	Min 0 0
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 C/Address Type ID	Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC) Bit(PLC)	Address Type LB X Y M I C	Addressing Format ddd ooo ddd	Max 99999 377 377 9999 255 255	Help Min 0 0 0 0 0 0 0
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID	Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC)	Address Type LB X Y M T C LW	Addressing Format Addressing Format	Max 99999 377 377 9999 255 255 9999	Help Min 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2	Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC)	Address Type LB X Y M I C LW IV	Addressing Format Addressing Format	Max 99999 377 377 9999 255 255 9999 255	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressYiew TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2	Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (PLC) Word (PLC)	Address Type LB X Y M I C LW IV CV	PLC block pack : [Cancel Addressing Format ddd ddd ddd ddd ddd ddd ddd	Max 99999 377 377 9999 255 255 9999 255 199	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2	Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) Word (PLC)	Address Type LB X Y M I C LW IV CV D	PLC block pack : [Cancel A Addressing Format ddd ddd ddd ddd ddd ddd ddd ddd	Max 99999 377 377 9999 255 255 9999 255 199 9999	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2	Bit/Word Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) DWord (PLC)	Address Type LB X Y M I C LW IV CV D CV2	Addressing Format Cancel Addressing Format ddd ddd ddd ddd ddd ddd ddd ddd ddd	Max 99999 377 377 9999 255 255 9999 255 199 9999 255	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2	Bit/Word Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) DWord (PLC) Word (PLC)	Address Type LB X Y M T C LW TV CV D CV2 SD	PLC block pack : [Cancel Addressing Format ddd ddd ddd ddd ddd ddd ddd ddd ddd d	Max 99999 377 377 9999 255 255 99999 255 199 9999 255 199 9999	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2 	Bit/Word Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC)	Address Type LB X Y M T C LW TV CV D CV2 SD RWI	PLC block pack : [Cancel Addressing Format ddd ddd ddd ddd ddd ddd ddd ddd ddd d	Max 99999 377 377 9999 255 255 9999 255 199 9999 255 199 9999 255 9999 32767	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2 	Bit/Word Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC)	Address Type LB X Y M T C LW TV CV D CV2 SD RWI RBI	PLC block pack : [Cancel Addressing Format ddd ddd ddd ddd ddd ddd ddd ddd ddd d	Max 99999 377 377 255 255 9999 255 199 9999 255 199 9999 255 9999 32767 2047f	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2 	Bit/Word Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) Bit(HMI) Bit(HMI)	OK OK LB X Y M I C LW IV CV D CV2 SD RWI RBI RB	PLC block pack : [Cancel Addressing Format ddd ddd ddd ddd ddd ddd ddd ddd ddd d	Max 99999 377 377 255 255 9999 255 199 9999 255 199 9999 255 9999 32767 2047f 2047f	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2 	Bit/Word Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Bit(HMI) Bit(HMI) Bit(HMI)	OK OK LB X Y M I C LW IV CV D CV2 SD RWI RBI RB RW	PLC block pack : Cancel	Max 99999 377 377 255 255 255 9999 255 199 9999 255 199 9999 255 9999 32767 2047f 2047f 65535	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2 	Bit/Word Bit/Word PLC ID=10 Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (PLC) Word (PLC) Bit(HMI) Bit(HMI) Bit(HMI)	OK OK LB X Y M T C LW TV CV D CV2 SD RWI RBI RB RW Ms_RB	PLC block pack : Cancel	Max 99999 377 377 255 255 255 199 9999 255 199 9999 255 9999 32767 2047f 2047f 65535 4095f	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PLC time out consta LCAddressView TSUBISHI FX0n/FX2 .C/Address Type ID ITSUBISHI FX0n/FX2 	Bit/Word Bit/Word PLC ID=10 Bit(HMI) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Bit(PLC) Word (HMI) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Word (PLC) Bit(HMI) Bit(HMI) Bit(HMI)	OK OK LB X Y M I C LW IV CV D CV2 SD RWI RBI RB RW	PLC block pack : Cancel	Max 99999 377 377 255 255 255 9999 255 199 9999 255 199 9999 255 9999 32767 2047f 2047f 65535	Help Min 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0



communication is then enabled.



■ There will always be a PLC selected in EB500 system parameter settings, in this case, even to read/write EB Pro HMI Local Data only, the ID of the selected PLC in EB500 system

parameters must also be filled in EB Pro parameter 1.

■ 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 EB Pro HMI Local data.

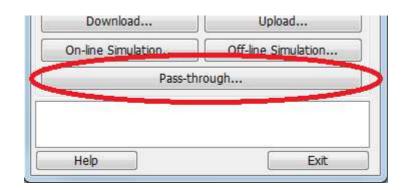
Device address:

Bit/Word	EB500	EB Pro	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	

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		
© Ethernet	O COM port	
Virtual COM I	Port (PC <-> PLC)	
	Please install 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].

Ethernet	O COM port
Virtual COM F	Port (PC <-> PLC)
	Please install weintek virtual serial port driver
PLC Connecti	on Port (HMI <-> PLC)
1 20 00111000	
HMLIP	
HMLIP	

If the dialogue below pops up during installation, please click [Continue Anyway].



After process is completed, the virtual COM port is displayed as follow.

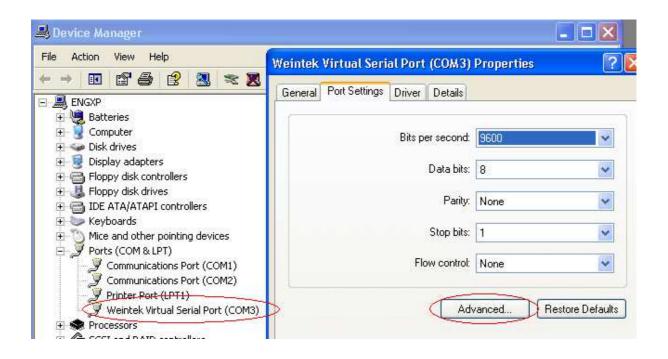
Virtual COM Port (PC <-> PLC)					
	СОМЗ				

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.

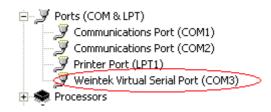


If users want to change the number of virtual serial port, please click [Weintek Virtual Serial Port] to open [Port Settings] / [Advanced...], as follows:



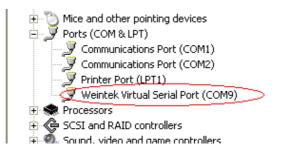
For example, user changes virtual serial port from COM 3 to COM 9.

WEINTEK



	Bits per second: 9	1600	~
anced Settings fo	or COM3	<i>51</i> .	
V Use FIED buff	ers (requires 16550 c	ompatible UAR	ті
2010/0 10			
Select lower se	ettings to correct con		ns.
Coloot biobox	abbinan far fastar narl		
	settings for faster perl	formance.	
Select highers Receive Buffer: Lo		formance.	4
	w (1)	formance.	ġ.

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 Por	t (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.

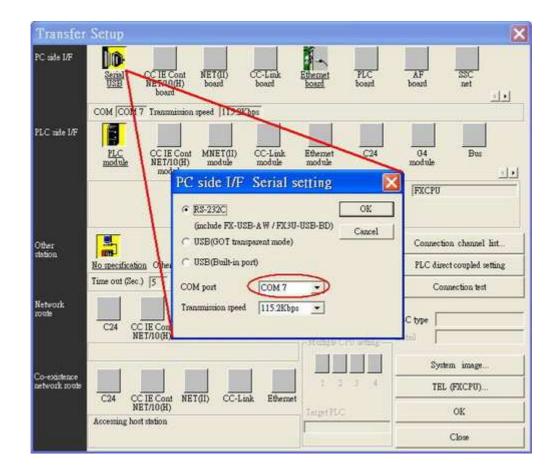
PLC	COM 2 Ethernet Weintek Virtual Serial Port (COM D	7)
в	Virtual COM Port (PC <-> PLC) COM7 PLC Connection Port (HMI <-> PLC)	
-	HMI IP: 192, 168, 1, 206 COM 2 RS232	
	Install Uninstall Apply C	

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:



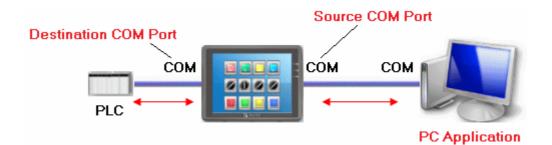
After completing all settings, when users execute PLC application on PC, the HMI will be switched automatically to pass-through mode (the communication between HMI and PLC will be suspended this moment and it will be resumed if the application closes), as follows:



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.

Ethernet	⊙ COM port			
HMLIP:	192.168.1.37		~	
	Get HMI Comm	unication Parameters		
H	HMI work mode: Ur	nknown		
Source COM Po	ort (PC-> HMI)			
	СОМ 1 👻		RS232	~
Baud rate :	9600 💌	Data bits :	8 Bits	~
Parity :	Even 💌	Stop bits :	1 Bit	~
Destination COM	M Port (HMI -> PLC)			
	СОМ 2 💌		RS232	~
Baud rate :	9600	Data bits :	8 Bits	*
Parity :	Even 💌	Stop bits :	1 Bit	~
-				
Start Pass-thro	ough Stop P	Pass-through		

[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.

Source COM port and Destination COM port

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

0 : RS232	1 : RS485/2W	2 : RS485/4W
0 : 4800	1:9600 2	2 : 19200 3 :
38400		
4 : 57600	5 : 115200)
7 : 7 bits	8 : 8 bits	
0 : none	1 : even 2	: odd
	0 : 4800 38400 4 : 57600 7 : 7 bits	0 : 4800 1 : 9600 2 38400 4 : 57600 5 : 115200 7 : 7 bits 8 : 8 bits



LW-9554 (stop bits)	1:1 bit	2 : 2 bits	

COM 2 mode settings

LW-9555 (PLC I/F)	0 : RS232	1 : RS485/2\	N	2 : RS48	35/4W	/
LW-9556 (baud rate)	0 : 4800	1:9600	2 : 1920	0	3	:
	38400					
	4 : 57600	5 : 1152	00			
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 ● PLC 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
PLC l/F : RS-485 2W PLC default station no. : 2 COM : COM3 (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 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:

⊖hmi ⊙pl	5		
Location : Local	Settings		
PLC type : SIEMENS S7/200			
V.1.90, SIEMENS ST	200.so		
PLC MF : RS-232	PL	C default station no	.: 2
COM: COM1 (9600,E,8,1)			Settings .
Use broadcast com	mand		
		2	
	ds): 5 😽 🛩		
Interval of block pack (wor			
Interval of block pack (wor Max. read-command size (wor	ds) : 32 👻		

After that, press **[Pass-through]** to assign HMI IP address; for example, 192.168.1.37. Finally, press **[Get HMI Communication Parameters]**, as follows:

HMHP:	192.168.1.37		*	
	Get HMI Commu	unication Parameters		
	HMI work mode : N	ormal		
Source COM Po	nt (PC -> HMI)			
	СОМ 1 💌		RS232	~
Baud rate :	9600	Data bits :	8 Bits	~
Parity :	Even 💌	Stop bits :	1 Bit	~
Destination CON	I Port (HMI -> PLC)			
	СОМ 3		RS485 2W	~
Baud rate :	9600	Data bits :	8 Bits	~
Parity :	Even 💌	Stop bits :	1 Bit	~

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.

Тад	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. EasyBuilder Pro supports protection functions for project files to ensure users' design achievement.





• The following protection functions can't be decrypted by factory since they are encrypted by users, therefore, please remember your password.



30.1 XOB Password

After project (MTP) is completed, users can compile the file to XOB format that can be downloaded to HMI. Password can be set to protect the XOB file in **[Compiling]** window.

A password will be required when attempting to decompile the XOB file to MTP. (XOB password range: 0 ~ 4294901760)

Compiling			×
Project name : C:\Docume	ents and Settings\Administrate	pr\桌面\MTP1.mtp	
XOB file name : C:\Docume	ents and Settings\Administrate	or\桌面\MTP1.xob	
XOB password : Set	(used in decompiler)	Decompilation is prohibited	
Select the languages used on the Startup			
Startup P	assword		
Language 1 O error(s), 0 warning(s)	Password :	(range : 0 ~ 4294901760)	
Object size : 26094 by Font size : 314444 b Picture size : 495032 b Shape size : 292 byte Sound size : 63164 by Macro size : 14 bytes Address tag size : 10 bytes	iytes iytes s		
Total size : 899050 b	oytes (0.86M)		
			<u>×</u>
Double click error messages to mo	/	objects !	
Compile Buil	d font files		Close

If the password is input incorrectly for three times when decompiling, please reset the decompilier.

roject ne name , c. (Docu	ments and Settings\Administrator\桌面\MTP1.mtp	Browse.,
XOB password : ***	syBuilder8000	
	Password error three times !! Please restart this application !!	2 8
	OK	



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.

Project name : C:\Documents and Settings\Administrator\桌面\MTP1.mtp XOB file name : C:\Documents and Settings\Administrator\桌面\MTP1.xob C:\Documents and Settings\Administrator\桌面\MTP1.xob Select the languages used on the HMI Startup language after redownloading the project : Language 1 C:\EB8000CHT\font\MTP1\$1.ttf (Times New Roman) 0 error(s), 0 warning(s) Object size : 25526 bytes Font size : 314444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
C:\EB8000CHT\font\MTP1\$1.ttf (Times New Roman) 0 error(s), 0 warning(s) Object size : 25526 bytes Font size : 314444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
Select the languages used on the HMI Startup language after redownloading the project : Language 1 Language 1 C:\EB8000CHT\font\MTP1\$1.ttf (Times New Roman) 0 error(s), 0 warning(s) Object size : 25526 bytes Font size : 314444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
Startup language after redownloading the project : Language 1 Language 1 C:\EB8000CCHT\font\MTP1\$1.ttf (Times New Roman) 0 error(s), 0 warning(s) Object size : 25526 bytes Font size : 314444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
C:\EB8000CHT\font\MTP1\$1.ttf (Times New Roman) 0 error(s), 0 warning(s) Object size : 25526 bytes Font size : 314444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
C:\EB8000CHT\font\MTP1\$1.ttf (Times New Roman) D error(s), 0 warning(s) Object size : 25526 bytes Font size : 314444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
0 error(s), 0 warning(s) Object size : 25526 bytes Font size : 31444 bytes Picture size : 495032 bytes Shape size : 292 bytes	
Picture size : 495032 bytes / Shape size : 292 bytes /	
Picture size : 495032 bytes / Shape size : 292 bytes /	
ihape size : 292 bytes /	
"otal size : 898458 bytes (0.86M) /	
ucceeded	
ouble click error messages to modify the attributes of relative objects !	
	lose

When attempting to decompile a XOB file that is already set to

[Decompilation is prohibited], an error message "This xob is set to not decompile!!" will be shown.

XOB file name :	C:\Documents and	Settings\Adminis	strator\桌面\MTI	P1.xob	Browse
Project file name :	C:\Documents and	Settings\Adminis	strator\桌面\MTI	P1.mtp	Browse
XOB password :	***				
ecompiling					
	decompile !!				
ecompiling his xob is set to not	decompile !!				
	decompile !!				



30.3 Disable HMI Upload Function [LB-9033]

EasyBuilder Pro provides system reserved address [LB-9033]. When this address is set to ON, HMI will disable upload function of XOB file. HMI needs to be rebooted to activate [LB-9033].

No.	Address tag r	name			PLC name	Address type	Address	Read/Write	1
30	LB-9029 : sa	ave all recipe data to m	achine (set ON)		Local HMI	Bit	LB-9029	Read/Write	
31	LB-9030 : up	odate COM 1 communi	cation parameters (set OI	4)	Local HMI	Bit	LB-9030	Read/Write	1
32	LB-9031 : up	odate COM 2 communi	cation parameters (set OI	N)	Local HMI	Bit	LB-9031	Read/Write	
33		pdate COM 3 communication parameters (set ON)			Local HMI	Bit	LB-9032	Read/Write	
34			e (when off) HMI upload f		Local HMI	Bit	LB-9033	Read/Write	
35	LB-9034 : sa	ave event/data samplir	ng to HMI, USB disk, SD	card (set ON)	Local HMI	Bit	LB-9034	Read/Write	
36	LB-9035 : HI	MI free space insufficie	ency alarm (when ON)		Local HMI	Bit	LB-9035	Read/Write	
37	LB-9036 : SI	D card free space insu	fficiency alarm (when ON)	Local HMI	Bit	LB-9036	Read/Write	
38			ciency alarm (when ON)		Local HMI	Bit	LB-9037	Read/Write	
39	LB-9038 : U	SB 2 free space insuffi	ciency alarm (when ON).		Local HMI	Bit	LB-9038	Read/Write	
40	LB-9039 : st	atus of file backup acti	ivity (backup in process il	ON)	Local HMI	Bit	LB-9039	Read/Write	
41		acklight up (set ON)			Local HMI	Bit	LB-9040	Read/Write	
42	LB-9041 : ba	acklight down (set ON)		/	Local HMI	Bit	LB-9041	Read/Write	
43	LB-9042 ; ad	cknowledge all alarm e	vents (set ON)		Local HMI	Bit	LB-9042	Read/Write	
44		hacknowledged events		/	Local HMI	Bit	LB-9043	Read/Write	
45		sable remote control (w			Local HMI	Bit	LB-9044	Read/Write	
46	LB-9045 : m	emory-map communica	ation fails (when ON)	/	Local HMI	Bit	LB-9045	Read/Write	
47	LB-9046 : pr	oject key is different fro	om HMI key (when ON)		Local HMI	Bit	LB-9046	Read/Write	
48		boot HMI (set ON whe	n LB9048 is on)		Local HMI	Bit	LB-9047	Read/Write	
49		boot-HMI protection			Local HMI	Bit	LB-9048	Read/Write	
50			(set OFF) watch dog (i se	ries support only)	Local HMI	Bit	LB-9049	Read/Write	
51	LB-9050 : us	ser logout		/	Local HMI	Bit	LB-9050	Read/Write	
6	LE SAFA LE	7	17 LOUGE 1	/		1.60	1.0.0004		5
Heare	can import MT5	00 tag to represent the	address						-
1	New	Delete	Delete All	Settings					
			Delete Mil	acungsin					
Sa	ave Tag File	Load Tag File							
E	Export CSV	Import CSV	Export EXCEL	Import EXCEL			P	Exit	

When attempting to upload a XOB file set to this function, the XOB file gained after uploading will be 0 bytes.



30.4 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].

Project protection (i series only)				
, 1 , 1 , 1 ,		1		
🔽 Enable Project key	: 111111	(range : 0 ~ 4294901750)		
* If this key is different from HMI key, the project won't be executed normally.				
* Use L W9046~9047 to change HMI key. LB9046 indicates check result (key error when status is on).				

LW-9046 ~ LW-9047 (32-bit) can be used to set the **[HMI key]**. The value can't be read or written into these two registers by remote HMI. While using this function, set the password (**[Project key]** password range: 0 ~ 4294901750), and the XOB file can only be executed on specific HMI when [HMI key] and [Project key] are identical. If they are different, the system will turn LB-9046 ON. HMI needs to be rebooted every time when revising [HMI key].



■ When [HMI key] and [Project key] are different, HMI and PLC won't be able to communicate.



Please confirm your Internet connection before downloading the demo project.



30.5 Project Password (MTP file)

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

Enabling this, password will be required if attempting to edit MTP file. (MTP password range: 1 ~ 4294967295)

□ Enable User 10 □ Enable User 11 □ Enable User 12 □ Enable Project password (M TI ♥ Enable	IP file Password : [19854561] OK ? file) Setting	(range: 1 ~ 4294967295) Cancel	► opening pop-up	tting, when this project, a window requires to assword will be
🔽 EasyBuilder80	000			
<u>File Tool H</u> elp	P			
0 🖨 🖬 🛛	EasyBuilder8000		3	
1 1 1 2 2 2	Password : ***	(Limited to 3 tim	es)	
Windows	ОК	Cancel		

When using "Window Copy" function, if the source file is protected by MTP password, please input correct password for EasyBuilder Pro to execute window copy.

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(#)

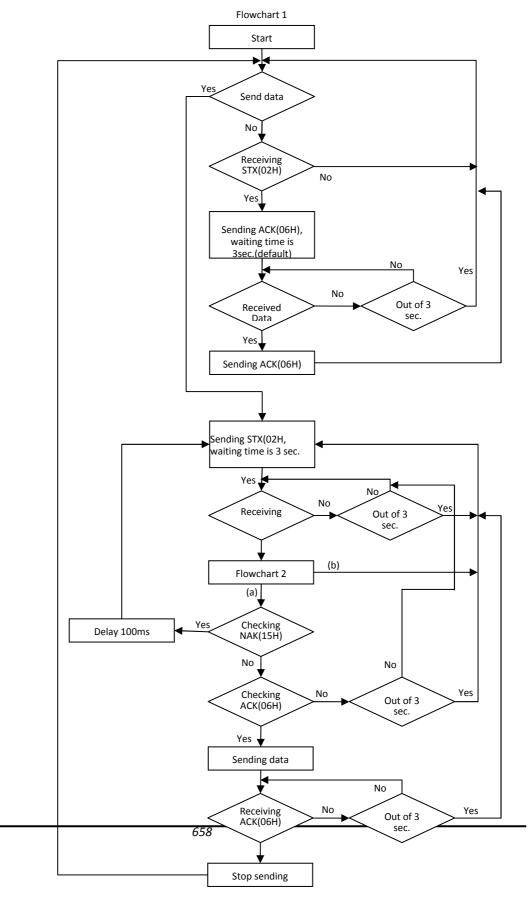
RS485 (4W)		
Master	Slaver	
TX+(#)	RX+(#)	
TX-(#)	RX-(#)	
RX+(#)	TX+(#)	

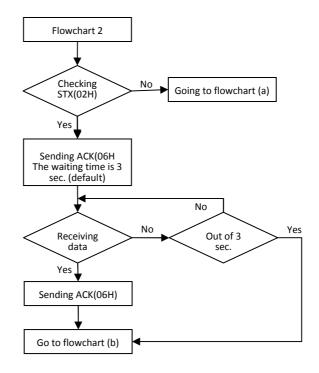


RX-(#)	TX-(#)
GND(#)	GND(#)

Note: # means being decided by PLC or controller.

The flowchart of communication as following:





Caution:

WEINTEK

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 comn	nand	
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 comm	and	
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
3	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
4~4+n-1	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
4+n , 4+n+1	0x10 , 0x03	End sign
4+n+2	0x##	checksum, Xor check-up and bytes in the front

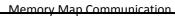
Below is an example for observation process of communication. If Master has a 0x0a in MW3, according to this protocol, master will communicate with slave immediately, and slave will put the 0x0a in corresponding MW3, the procedure is as following:

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,0x00
	0x00	
6,7	0x10,	End sign





	0x03	
8		checksum, 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	Address(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 ,
		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	ip.			
	OHMI	⊙ PLC			
Location :	Local	~	Settings		
PLC type :	: Memory Ma	p			
	V.1.00, MEN	ORY_MAP.	50		
PLC I/F :	RS-232	~	PLC default s	tation no. :	0
COM :	: COM1 (115)	200,E,8,1)			Settings.
		drast comm	and		67
M Fort Settings					V
COM :	COM 1	~	Timeout (sec) : 0.5	×
COM : Baud rate :	1	• •	Timeout (Turn around delay (and a street me	*
	115200	~		ms) : 0	
Baud rate :	115200 8 Bits	× × ×	Turn around delay (Send ACK delay (ms) : 0	~
Baud rate : Data bits :	115200 8 Bits Even	× × ×	Turn around delay (Send ACK delay (Paramet	ms): 0 ms): 0	
Baud rate : Data bits : Parity :	115200 8 Bits Even		Tum around delay (Send ACK delay (Paramet Paramet	ms): 0 ms): 0 er 1: 0	

June .	
	WEINTEK

	ne : Memory	Мар		
	⊖ HMI	⊙ PLC		
Locatio	on : Local	*	Settings	
PLC typ	pe : Memory	Мар		
	V.1.00, M	MEMORY_MAP.s	0	
PLC I	/F : RS-232	~	PLC default station no. : 0	
ort Settings				
			Treesuit (sec) - 0.8	
	COM 1	-	Timeout (sec) : 0.8	
COM :	COM 1	v v	and the second se	
COM : Baud rate :	COM 1 115200 8 Bits	~	Turn around delay (ms) : 0	
COM : Baud rate : Data bits :	COM 1 115200 8 Bits Even		Turn around delay (ms) : 0 Send ACK delay (ms) : 0	
сом :	COM 1	-	and the second se	.

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:

eneral Secur	ity Shape Label
Descrip	tion :
Read address	
PLC na	ame : Memory Map Setting
Addr	ess: MB
	Invert signal
Write address	
PLC na	ame : Memory Map Setting
Addr	ess: MB 👽 0
	Write when button is released
Attribute	
	style : Toggle 🛛 🔛

A multistate switch object setting is as following:

Jeneral	Security	Shape 1	Label				
	Description						
	Mode	Value		/	Offset :	0	
Read	address				0.07537693	1.5	
	PLC name :	Memory	Мар			~	Setting
				- 14 M	1.11	1	
Write	Address : address :	MW		0			16-bit Unsigned
Write	address :		Man	0			
Write	address : PLC name :	Memory	Мар			~	Setting
Write	address :	Memory MW	Map when button	v 0		•	
- Write	address : PLC name : Address :	Memory MW		v 0		~	Setting
Attrib	address : PLC name : Address :	Memory MW		V 0	lo. of states :		Setting

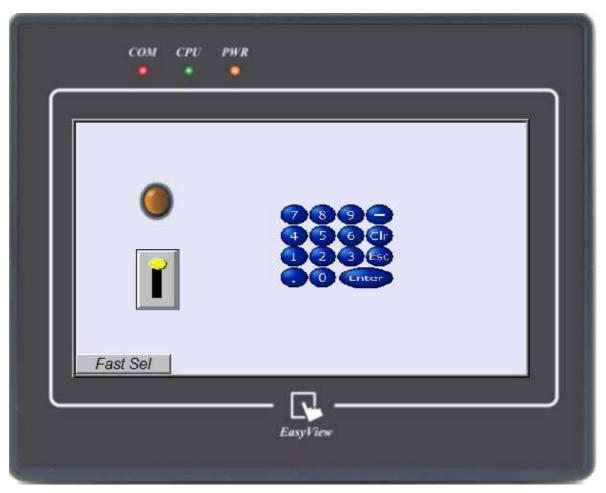
[Save],[Compile],[Download]

WEINTEK

Change parameter in [System Parameter Setting]/[PLC] and download to another HMI.

The HMI display is as following:





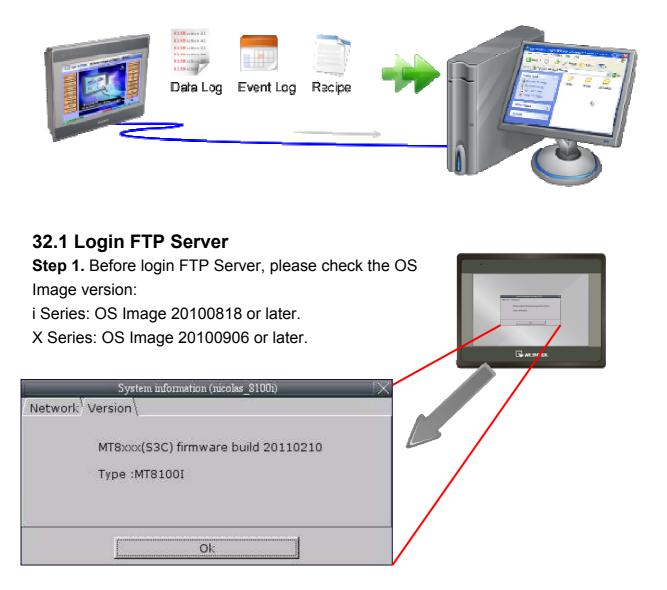
Users may try to touch the screen; the other HMI will act the same as current HMI.

The communicating way is the same as above-mentioned. The point is to keep the same data in the same register.



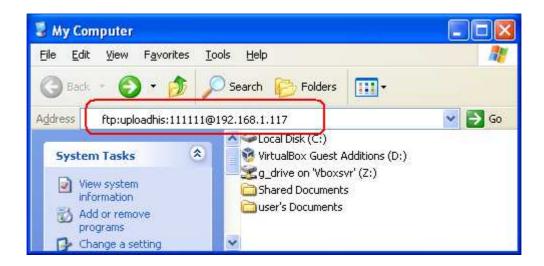
Chapter 32 FTP Server Application

In addition to backup history data from HMI to PC by SD card, 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.

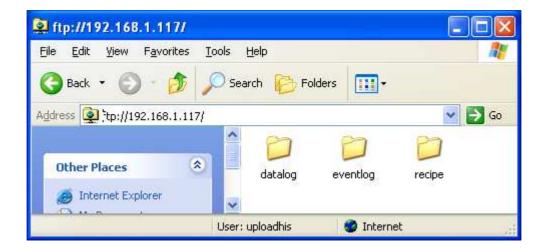


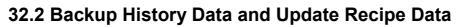


Step 2. Enter HMI IP: <u>ftp://192.168.1.117/</u> (example), login user name: uploadhis, and the HMI history upload password (if not changed, the default is 111111). Or, to directly enter <u>ftp://uploadhis:111111@192.168.1.117/</u>



Step 3. After entering IP, <u>ftp://192.168.1.117/</u> is shown, and the "datalog", "eventlog", and "recipe" folders can be seen.





• To backup "Data

WEINTEK

Sampling" records

 Click "datalog" folder to check the file names set by EasyBuilder Pro.
 Click on file names to check content.
 Copy and paste to save files on PC.

🕸 ftp://192.168.1.117/de	atalog/123/		
<u>File E</u> dit <u>V</u> iew F <u>a</u> vorites	<u>T</u> ools <u>H</u> elp		
🕝 Back 🝷 🕥 🕘 🥠	🔎 Search 🔀 Fol	Iders	
Address 👰 ftp://192.168.1.11	7/datalog/123/	✓ >	Go
Other Places	20110411.dtl	20110412.dtl 20110413.dtl	ñ
latalon	~		
	User: uploadhis	🎯 Internet	

To backup "Event

(Alarm) Log" records

1. Click "eventlog" folder to check the files.

2. Copy and paste to save files on PC.

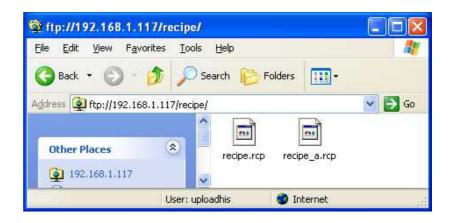
🕸 ftp://192.168.1.117	/eventlog/	
<u>File E</u> dit <u>V</u> iew F <u>a</u> voriti	es <u>T</u> ools <u>H</u> elp	
🕝 Back 🔹 🕥 🕘 💋	Search 😥 Folders	-
Address 👰 ftp://192.168.1	117/eventlog/	🝸 🛃 Go
Other Places	 €L_20110411. EL_201104 evt 	EL_201104
	User: uploadhis 🛛 🎯 Internet	

• To backup and update

"Recipe" records

1. Click "recipe" folder to check the files.

2. To update "recipe" data on HMI, overwrite "recipe.rcp" with new data and restart HMI in one minute.









■ Since recipe data is automatically saved once every minute, after updating "recipe.rcp" or "recipe_a.rcp", HMI must be

restarted in one minute otherwise the new updated recipe data will be overwritten by the former data. [LB-9047] and [LB-9048] can also be used to restart HMI. Set [LB-9048] to ON and then set [LB-9047] to ON to successfully restart HMI.

System Registers:

[LB-9047] reboot HMI (set ON when LB9048 is ON) [LB-9048] reboot-HMI protection

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



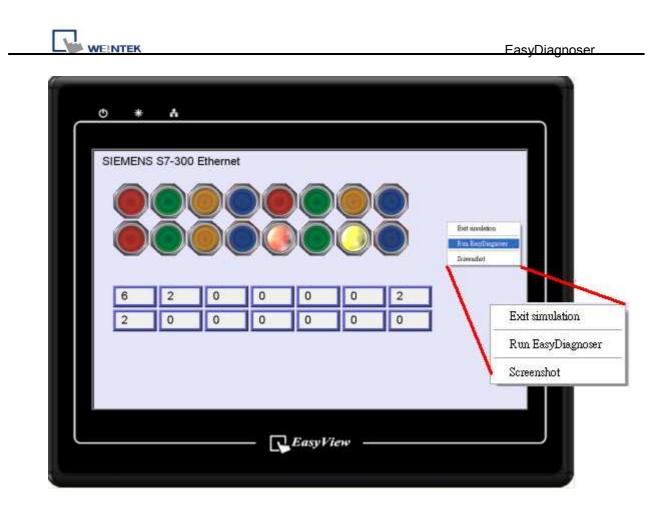
Step 2.

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.

Select HMI		nput HMI IP address	¥
Project Port:	8000		OK Exit
Select HMI			×
IP Name HMI Name: * OS 20091002 or lat	Tina-MT8100i Search Search All er supports	Search HMI 6.1.233 (Demo- 2.168.1.237 (Demo- 192.168.1.39 (test) 192.168.1.44 (Default 192.168.1.47 (Tina-M	-S7_1200) HMI)
<u>P</u> roject Port:	8000		OK Exit

It is also available to right click and select "Run EasyDiagnoser" for entering the setting window when executing On-Line Simulation in EB8000.



After setting completed, click OK, EasyDiagnoser operation window appears as below:

command: Read + V ddress Type: All No Cmd.	Vrite <u>D</u> evice: Rango PID Device		~ 99	Station:	
		50 <u>5</u>		566	
No Cmd.	PID Device			499	Capture
		St.	Index	Address / Length Ti	Fime Error
ling Packages				g×	Logger
mfg I acyages					
Package ID	Device	Station	Index	Address / Length	[15:15:22] Looking for the target HMI
Package ID 4 (1)	Local HMI	Station 	Index 	[LB] 00562/1	
Package ID 4 (1) 8 (0)	Local HMI Local HMI) 10	The contraction of the second	[LB] 00562/1 [LB] 00574/1	[15:15:22] Looking for the target HMI
Package ID 4 (1) 8 (0) 67 (32)	Local HMI Local HMI SIEMENS S7/300	 1		[LB] 00562/1 [LB] 00574/1 [M] 00000/1	[15:15:22] Looking for the target HMI
Package ID 4 (1)	Local HMI Local HMI) 10	67 22	[LB] 00562/1 [LB] 00574/1	[15:15:22] Looking for the target HMI
Package ID 4 (1) 8 (0) 67 (32)	Local HMI Local HMI SIEMENS S7/300	 1		[LB] 00562/1 [LB] 00574/1 [M] 00000/1	[15:15:22] Looking for the target HMI
Package ID 4 (1) 8 (0) 67 (32)	Local HMI Local HMI SIEMENS S7/300	 1		[LB] 00562/1 [LB] 00574/1 [M] 00000/1	[15:15:22] Looking for the target HMI

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 File View Options Help Save Asd + Write Egit Accuress Type.			
	Exit			
	Exit current file.			
View				
🔚 Device Bax Ctrl+Alt+D	Click [Device Bar] to display Device window.			
🐗 <u>P</u> ackage Bar Ctrl+Alt+P	Click [Package Bar] to display Package window.			
📄 Logger Bar 🛛 Ctil+Alt+L	Click [Logger Bar] to display Logger window.			
Output Bax Ctul+Alt+O	Click [Output Bar] to display Output window.			





• Activity area

In the activity area, users can observe the communication between HMI and PLC.

Con	nmand:	Read + 1	Write	<u>D</u> evice:	All			ation: 0		
<u>A</u> dd	ress Typ	e: All		Range	0		99999		apture	
	No	Cmd.	PID	Device	St.	Index	Address / Length	Time	Error	1
•	139	R	68	SIEMENS 87/300	1	255	[DB10] 00000 / 14	50	0	-
	138	R	4	Local HMI	3949	322	[LB] 00562/1	20	0	
	137	R	8	Local HMI	37555	(455)	[LB] 00574/1	10	0	
	136	R	67	SIEMENS \$7/300	1	255	[M] 00000 / 1	40	0	
	135	R	4	Local HMI	3223	(22)	[LB] 00562/1	20	0	
	134	R	8	Local HMI	1000	(177)	[LB] 00574/1	20	0	
	133	R	68	SIEMENS S7/300	া	255	[DB10] 00000 / 14	30	0	
	132	R	4	Local HMI	3222	1231	[LB] 00562 / 1	20	0	
	131	R	8	Local HMI	2.000	177	[LB] 00574/1	20	0	
	130	R	67	SIEMENS S7/300	1	255	[M] 00000 / 1	40	0	
	129	R	4	Local HMI	1.77	0.579	[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.
	 If command is set Read + Write, the Read and Write
	information of Local HMI and PLC will be displayed in activity

	area.						
	 If command is set Read, the Read information of Local HMI 						
	and PLC will be displayed in activity area.						
	• If command is set Write, 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.						
	 If command is set Read + Write, the Read and Write 						
	information of Local HMI will be displayed in activity area.						
	• If command is set Read , the Read information of Local HMI						
	will be displayed in activity area.						
	• If command is set Write, the Write information of Local HMI						
	will be displayed in activity area.						
	c. PLC						
	Display information of PLC, it depends on the setting of						
	command as following.						
	 If command is set Read + Write, the Read and Write 						
	information of PLC will be displayed in activity area.						
	• If command is set Read , the Read information of PLC will be						
	displayed in activity area.						
	• If command is set Write , the Write information of PLC will be						
	displayed 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						
Туре	on the 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								
	Package ID	Device	Station	Index	Address / Length			
Đ	4 (1)	Local HMI	19 14	8 97	[LB] 00562/1			
	8 (0)	Local HMI	(22)	122	[LB] 00574/1			
Đ	67 (32)	SIEMENS S7/300 Ethernet	1	255	[M] 00000/1			
Đ	68 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000/3			
Đ	69 (3)	SIEMENS S7/300 Ethernet	্ৰ	11	[DB10] 00003/3			
Đ	70 (3)	SIEMENS S7/300 Ethernet	1	12	[DB10] 00006/3			
Đ	71 (5)	SIEMENS S7/300 Ethernet	S 1 S	322	[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 wo	
	the Package.

	Object		Screen	ID	Address
Ŧ	4 (1)	Local HMI	32	25	[LB] 00562/1
	8 (0)	Local HMI	8		[LB] 00574/1
	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 3rd column.



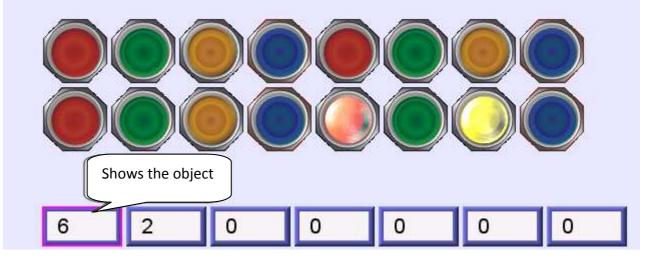
Poll	Polling Packages											
	Package ID	Device	Station	Index	Address / Length							
±	4 (1)	Local HMI			[LB] 00562/1							
	8 (0)	Local HMI			[LB] 00574/1							
±	67 (32)	SIEMENS S7/300 Ethernet	1		[M] 0000071							
Đ	68 (3)	SIEMENS S7/300 Ethernet	1	10	[DB10] 00000/3							

b. Double click **[Package ID]** then select **[object]**, the 1st 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.

Poll	ling l	Packages				C 🛛
	Ok	oject		Screen	ID	Address
Ŧ	4 (1)	Local HMI		(##)	[LB] 00562/1
	8 (0)	Local HMI	22	2574	[LB] 00574/1
Ŧ	67	(32)	SIEMENS S7/300 Ethernet	1	(##)	[M] 00000/1
	68	(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.

nices	9	X		
Local HMI		1		
Index	0			
Type Name	MT8000 Series HMI			
Location	Local 5 words			
Block Interval				
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	1		
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 7th and 8th lines of Macro.

File View Options Help	
📾 🥌 🔟 🗹	
Output	9×
[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.

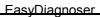
- 0: Normal
- 1: Time out
- 2: Fail Error
- 12: Ignore

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.

File <u>V</u> iew	Options	Help							
Co <u>m</u> mand:	Read + \	Write	Device:	SIEMEN	S S7/300 I	Ethernet 🔽 🗌 S	tation: 0		
Address Typ	e: All		No. 10 Range	0	~ 9	9999		≥apture	
No	Cmd.	PID	Device	St.	Index	Address / Length	Time	Error	1
691	R	71	SIEMENS S7/300 Et	1	265	[DB10] 00009/5	318	12	
590	R	67	SIEMENS S7/300 Et	1	255	(M) 00000 / 1	310	12	
589	R	66	SIEMENS S7/300 Et.	1	10	[DE10] 0009073	300	12	
588	R	69	SIEMENS 87/300 Et	1	11	[DB10] 00003/3	31.0	12	
587	R	78	SIEMENS S7/300 EL.	1	12	[DB10] 00086 / 3	310	12	
588	R	71	SIEMENS S7/300 Et	3	255	[DB10] 0000915	1.21 B	12	
585	R	67	SIEMENS S7/300 Et	. 1	255	[M] 0000071	1120	12	
584	R	68	SIEMENS S7/300 Et	1	10	[0810] 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

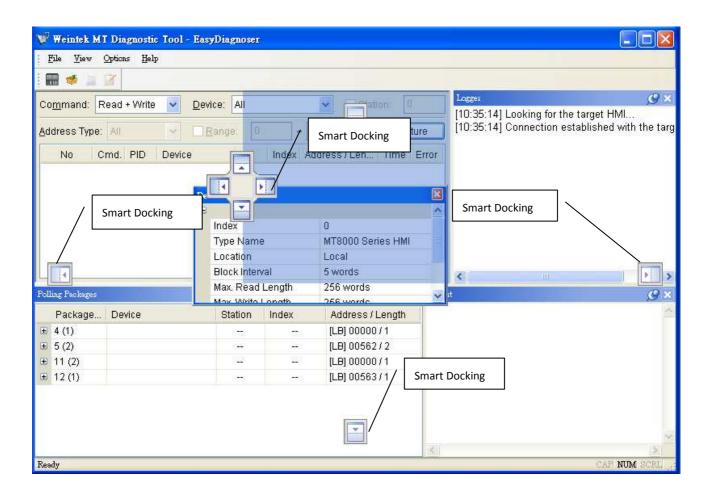
The captured information of Easy Diagnoser can be saved as *.xls which can be read in Excel.

File <u>V</u> iew Save <u>/</u>	-	Help						
Exit	d + V	Vrite		Device:	-	S S7/300		~
Idress Ty	pe: All		×	Range	s 8	~ 9	99999	
		T Indexed	The second second second					
No	Cmd.	PID	Device		St.	Index	Addres	s / Leng
		PID 68	The second second second			Index 10		and the section
No	Cmd.	1.1.1	Device	7300 Et	St.		Addres	00000/
No 176	Cmd. R	68	Device SIEMENS S7	/300 Et /300 Et	St.	10	Address (DB10) ()))))))))))))))))))



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_OB16:0:0		
13	TAG		Array2D		DINT[25,5]		(RADIX := Decimal, Cons
14	TAG		ArrayBool		BOOL[256]		(RADIX := Decimal, Cons
15	TAG		ArrayDIN1	•	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	TAC		L1		νοσ		(PADIX - Decimal Cone

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.

Pile Edit View Search Logic Communications	loois Window Help				
	🥁 🦀 🕰 (R R Select a	a Language	Y Ø
Rem Run 📜 🗖 Run Mode 🔤 🔜	Path: AB_ETH-1\192.168.1.1	30\Backplane\0*	- 品		
o Forces 💦 📮 Controller OK					
o Edits Battery OK		-()(U)(L)-	•		
	C > Favorites Add-On A	Safety 🔏 Alarms 🔏	Bit 🔏 Timer/O		
ontroller Organizer 🗸 🗸 🗙	Scope: 🛐 AB 🛛 🖌	how: All Tags			
Controller AB	Name _==	🛆 🛛 Value 🔶 🔶	Force Mask 🛛 🔦	Style	Data Type
Controller Fault Handler	+ ABC	56		Decimal	DINT
🔤 Power-Up Handler	+ Array2D	{}	{}	Decimal	DINT[25,5]
🖹 😁 Tasks	+ ArrayBool	{}	()	Decimal	BOOL[256]
🖻 🤯 MainTask	+ ArrayDINT	{}	{}	Decimal	DINT[130]
🕀 🕞 MainProgram	🕂 ArrayReal	{}	{}	Float	REAL[125]
Grand and a riggiants / ringes	b1	0		Decimal	BOOL
🔲 Ungrouped Axes	+ INT	{}	()	Decimal	INT[360]
🔁 Add-On Instructions	+ Local:1:C	{}	{}		AB:Embedded_I
🖹 😁 Data Types	+ Local:1:I	{}	{}		AB:Embedded_I
🕀 🙀 User-Defined	+ Local:2:C	{}	{}		AB:Embedded (
Grings Add-On-Defined	+ Local:2:1	{}	()		AB:Embedded
	+ Local:2:0	{}	()	1	AB:Embedded_(
표 🙀 Module-Defined	VarBool	0		Decimal	BOOL
- 📴 Trends	+ VarDint	21862		Decimal	DINT
😑 📇 I/O Configuration		0		Decimal	INT
	L Ut Vatint				
☐ ∰ CompactLogix5323E-QB1 System ☐ ∰ 1769-L23E-OB1 AB	T + VarInt VarBeal	0.0		Float	BEAL

Step 2. Export Tags data to CSV file.

👸 RSLogix 5000 - AB [1769-L23E-QB1 18.1	11]* - [Co	ontroller Tags - AB(contro	olle	a)]	
🃝 File Edit View Search Logic Communica	tions Too	ls Window Help			
Image: Constraint of the second se		Options Security Documentation <u>L</u> anguages Iranslate PLC5/SLC	-	▶ 🙀 📴 📝 🛒 🔍 (1.130\Backplane\0* //(.)(0)(L)- // Satety / Alarma / El	Q
Controller Organizer.		Import Export		Tags and Logic Commen	- A
Controller AB	^	Motion		Component	
Controller Fault Handler		Monitor Equipment Phases		56	
🖨 😁 Tasks		Custom Tools		{} ()	
🖨 🤯 MainTask 😨 🎝 MainProgram	E	ControlFLASH		{···}	
🦳 🦳 Unscheduled Programs / Phases 🚍 🦰 Motion Groups	Ç2	Online Books		0	

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.



Font		Extended	l Memory	Printer/Backup Server	Look in:	C EB8000	S 🖉 💌 🖽 -
Device Device list :	Model		General Sy	stem Setting Secur	ty My Recent Documents	Contension Contension	
No.	Name	Location	Device type	Interface		library	
Local HMI	Local HMI	Local	MT6070iH/MT8070	. Disable	Desktop	project	
Local PLC 4	Allen-Bradley E	Local	Allen-Bradley E	Ethemet(IP=192.168.1.130,	Port=4	usb1	
					My Computer My Network	-	AB-Tags AB-Tags AB-Tags AB-Tags AB-Tags Can Can
					EasyBuil	10-2000	

Step 4. In object dialog, select PLC, click Tag and select a controller tag.

Jeneral	Security	Shape	Label			
	Description	1:				
Read	address					
	PLC name	Allen-	Bradley EtherNe	t/IP-Tag (CompactLo	gix/ 😽 📒	Setting
	Tag	1: 0			~ ?	
		Nau	me	Data Type	Descripti	on
			Controller Tags			
		_		BOOL[256] BOOL		
			VarBool	BOOL		
Blinki	ng					
	Mode	•				

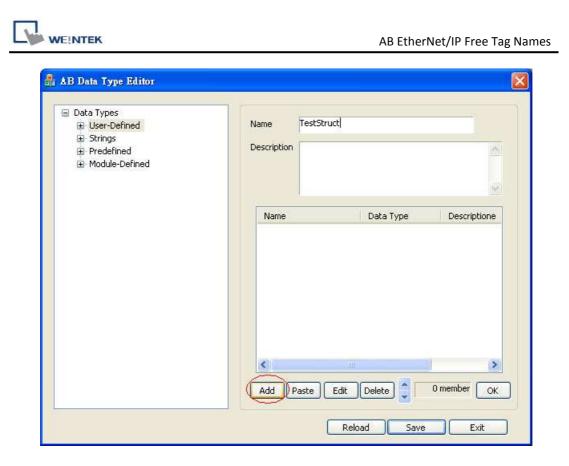


34.2 Adding New Data Type

Step 1. Right click on the assigned data type (usually labeled as [User-Defined]), then click [New Data Type] to start editing.

 Data Types User Defined Strings New Data Type Predefined Module-Defined 	Name Description		
	Name	Data Type	Descriptione
	Add Paste	Edit Delete	0 member OK

Step 2. Input the [Name] of the data type. [Description] can be skipped. For adding data member, click [Add].



Step 3. Input in [Name] and [Data Type] then click [OK] to leave.

Add data men	ıber	×
Name 🤇	Data1	
Description		<u>a</u>
Data Type		Binary Access
	 	
	mensions	
0	Dim <u>2</u> Dim <u>1</u>	Dim Q
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.

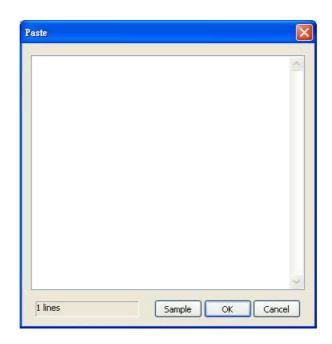
 Data Types User-Defined TestTypeA TestTypeB TestTypeC TestTypeD TestTypeE 	Name TestSt Description	ruct	
TestTypeF TestStruct Strings Predefined Module-Defined	Name Data1	Data Type INT	Descriptione
	Add Paste	Edit Delete	1 member OK

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.



lame:	TestTypeA			
Description:	I			
1embers:	Data Type	ata Type Size: 60 byte Style	(s)	External Access
VarBool	BOOL	Decimal	Description	Read/Write
BoolArray	BOOL[32]	Decimal		Read/Write
BoolArray VarReal	BOOL[32] REAL	Decimal Float		Read/Write Read/Write
VarReal	REAL	Float		Read/Write
VarReal RealArray	REAL REAL[5]	Float Float		Read/Write Read/Write
VarReal RealArray VarInt	REAL REAL[5] INT	Float Float Decimal		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

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.

iste					
VarBool BOOL BoolArrayBOOL[32] VarReal REAL RealArray REA VarInt INT IntArray INT[3] VarDint DINT DintArray DINT[3]	.[5]				
10 lines	-	ample	ок	Cancel	



Step 4. At this moment press [OK] to finish operating then return to the main window to view the successfully added multiple data.

Data Types User-Defined TestTypeA TestTypeB TestTypeC TestTypeD TestTypeE	Name TestTyp Description	A
 TestTypeF Strings Predefined Module-Defined 	Name VarBool BoolArray VarReal RealArray VarInt IntArray VarDint DintArray	Data Type Descriptione BOOL BOOL[32] REAL REAL[5] INT INT[3] DINT DINT[3]
	Add Paste (Edit Delete 8 member OK



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.

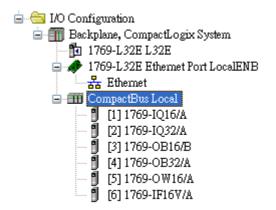
• To Re-edit:

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.



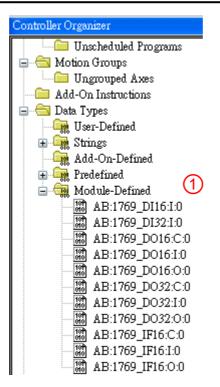
The Tags of these modules won't list the structure when exported to CSV file. Therefore, users should build it first.

	А	В	С	D	Е	F	G	Н
7	TYPE	SCOPE	NAME	DESCRIPT	DATATYPE	SPECIFIER	ATTRIBU	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

In [AB Data Type Editor.exe] in EB8000, right click on [Module-Defined], and then click [New Data Type...].

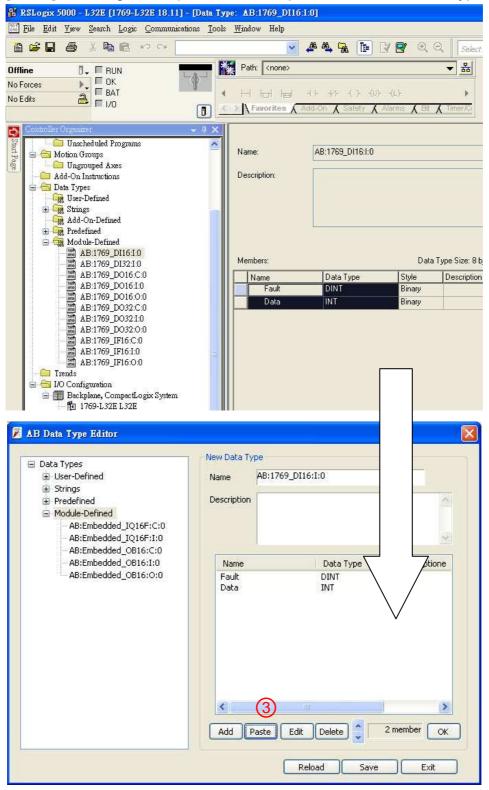
Data Types User-Defined Strings Predefined Module-Defined AB:Embedc New Data Type AB:Embedded_IQ16F:I:0	Name Description		
AB:Embedded_OB16:C:0 AB:Embedded_OB16:I:0 AB:Embedded_OB16:0:0	Name	Data Type	Descriptione
	Add Paste	Edit Delete	0 member OK

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].

Name	Data	
Description		
Data Type	INT	Binary Acces
	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	
Array [Dimensions	
0	Dim <u>2</u> Dim <u>1</u>	Dim <u>0</u>

Click [OK] to finish setting.