31. **cMT Diagnoser**

This chapter explains cMT Diagnoser which allows diagnostic operations in cMT / cMT X Series Model.
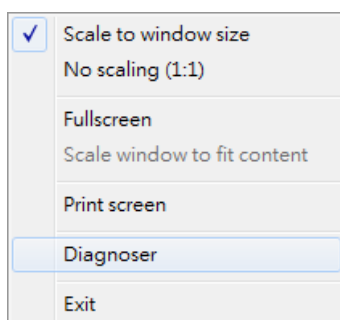
WEINTEK

EasyBuilder Pro V6.09.01

31.1.   Overview

cMT Diagnoser can monitor data and status of all objects in the current window when it is connected to an HMI, show the status of individual communication packet, show communication status with MQTT server, and debug macros.

31.2.   How to Start Diagnoser

cMT Diagnoser can be used in simulation mode or on HMI. When it is used on HMI, it can be opened from within cMT Viewer, or run an independent program named as cMT Diagnoser be directly opened on HMI.

1.   In on-line/off-line simulation, cMT Diagnoser can be opened by clicking the right mouse button in cMT Viewer simulation screen and then click Diagnoser.

✓ Scale to window size
No scaling (1:1)
Fullscreen
Scale window to fit content
Print screen
Diagnoser
Exit

2.   To permit connection to HMI from cMT Diagnoser, enable the Diagnoser option which is found in the [System Parameters] » [Remote] tab in EasyBuilder Pro.

WEINTEK

EasyBuilder Pro V6.09.01
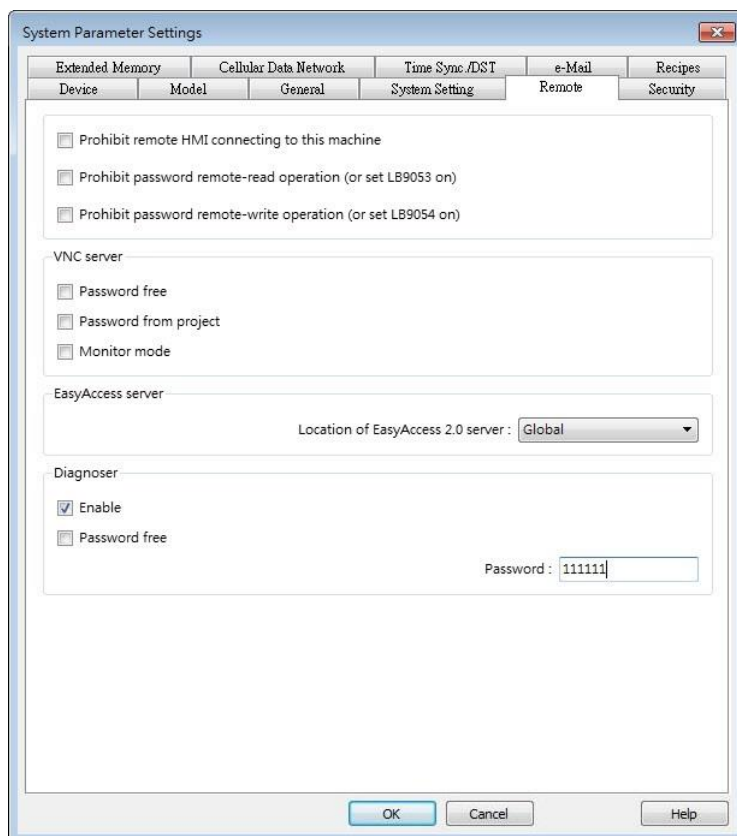
After downloading the project to HMI, cMT Viewer can open cMT Diagnoser. If [Password free] is not selected in Remote tab in previous step, a password is required for running cMT Diagnoser.
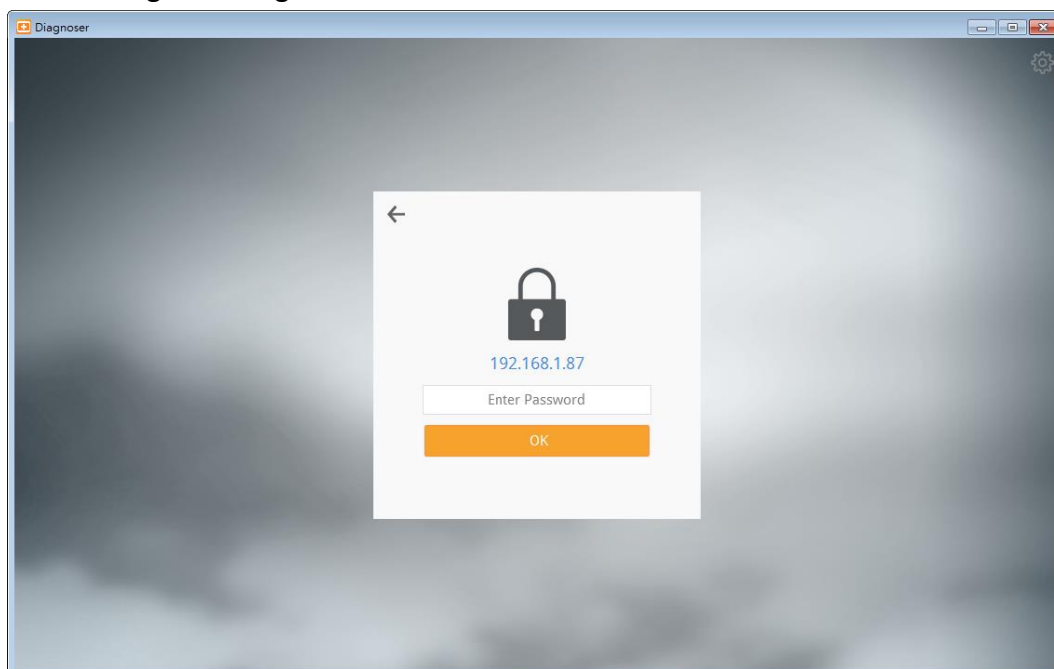

3. Execute cMTDiagnoser.exe, which can be run independently, in EasyBuilder Pro installation directory, and find an HMI to connect to.
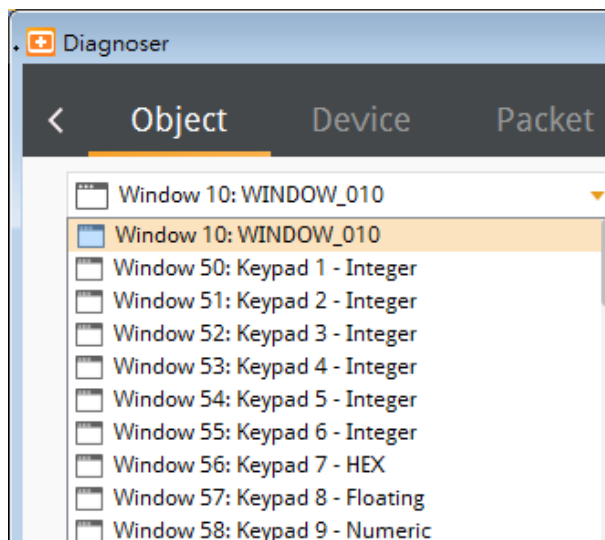
WEINTEK

EasyBuilder Pro V6.09.01
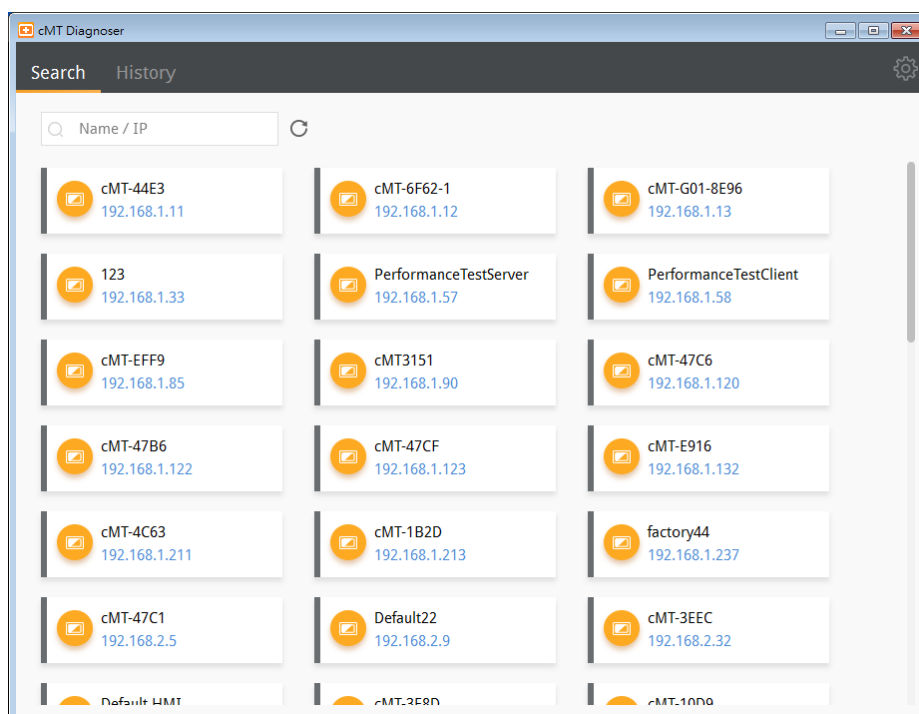
31.3.   cMT Diagnoser User Interface

31.3.1.   Object Settings

1. In the Object tab in cMT Diagnoser user interface, the objects used in the current window and their addresses, properties, and statuses are shown in a table.
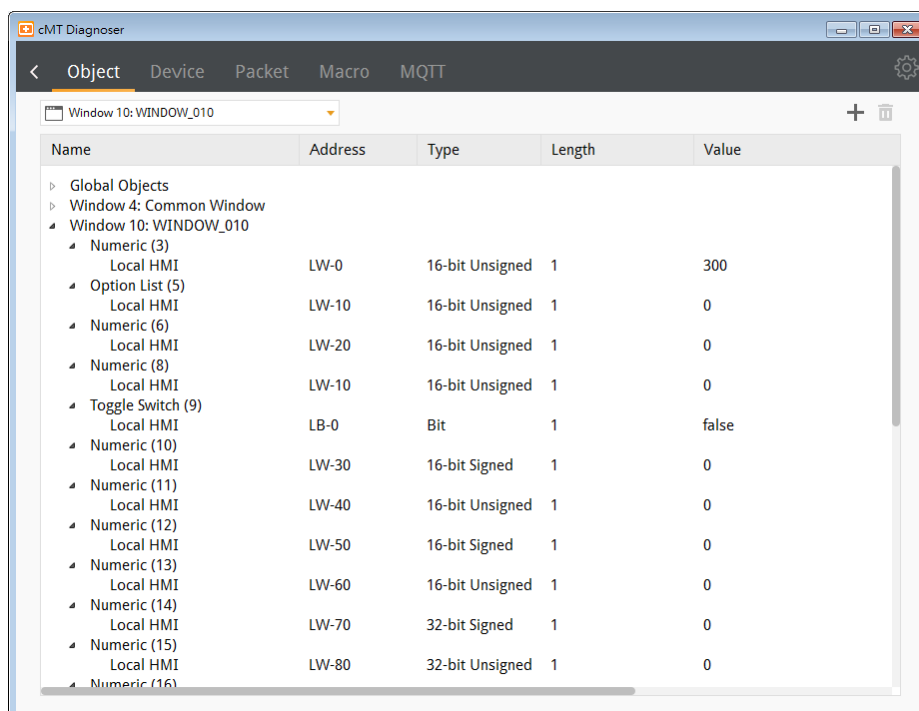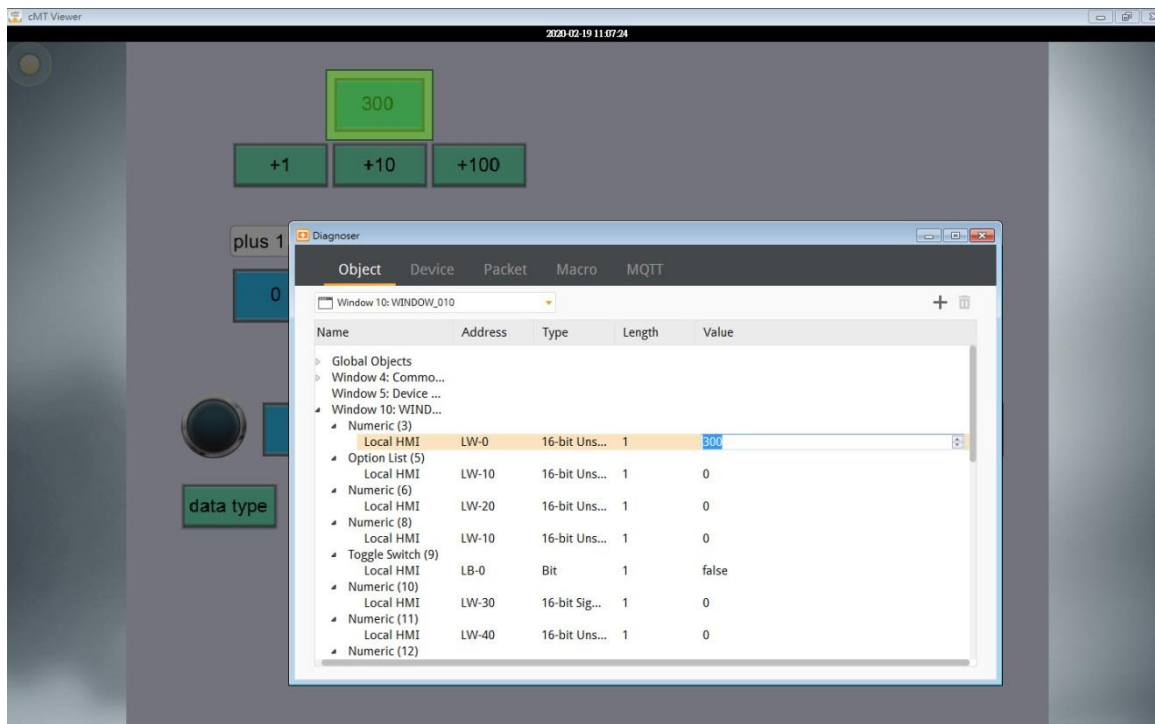

2. Selecting a window from the drop-down list can quickly change to another window.

WEINTEK   EasyBuilder Pro V6.09.01
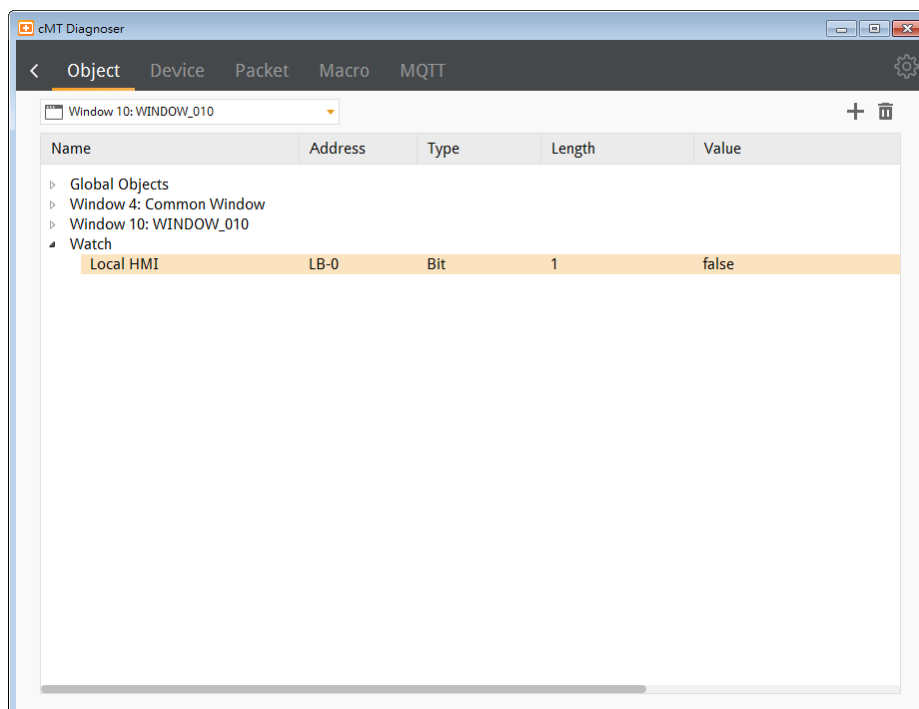
3. The object selected in the cMT Diagnoser window will be highlighted as shown below.
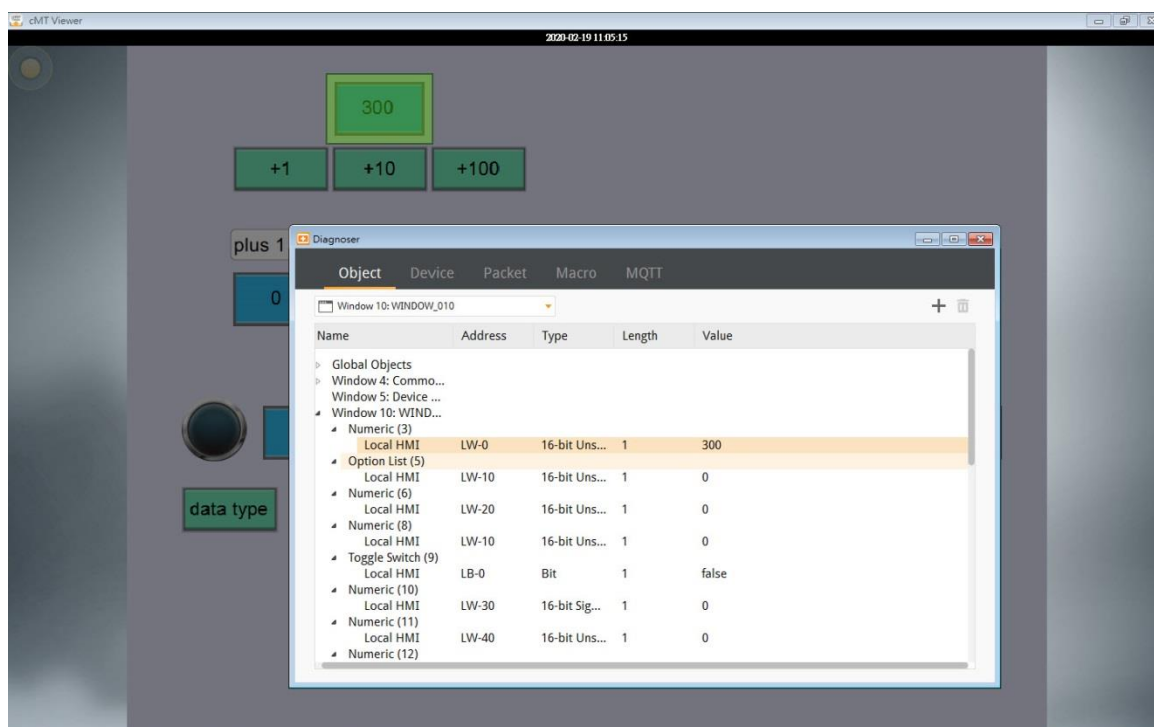
4. The value of an object can be changed by double-clicking on it in cMT Diagnoser window.
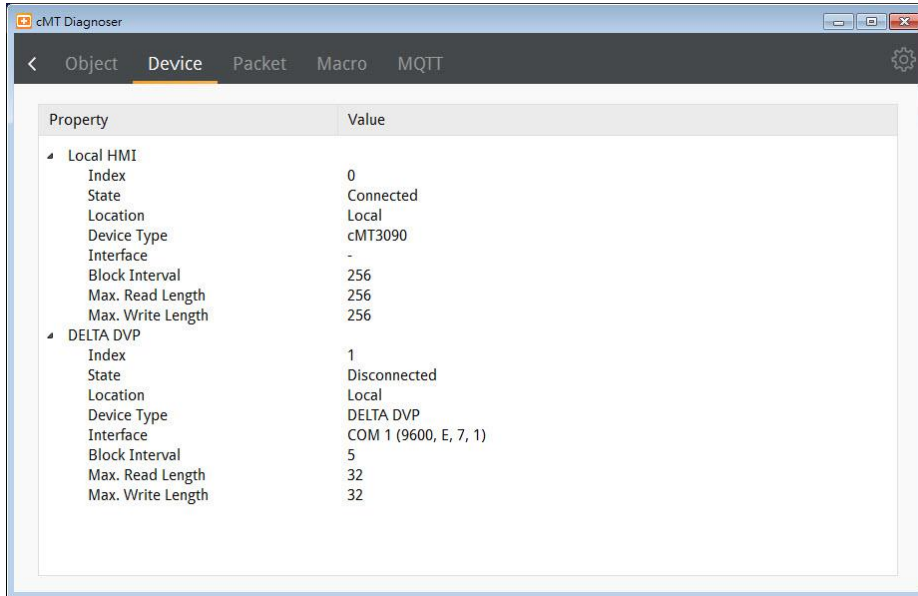
5. In cMT Diagnoser, watch addresses can be added or deleted. Click ➕ to add a new address, or select an address and then click 🗑 to delete it.

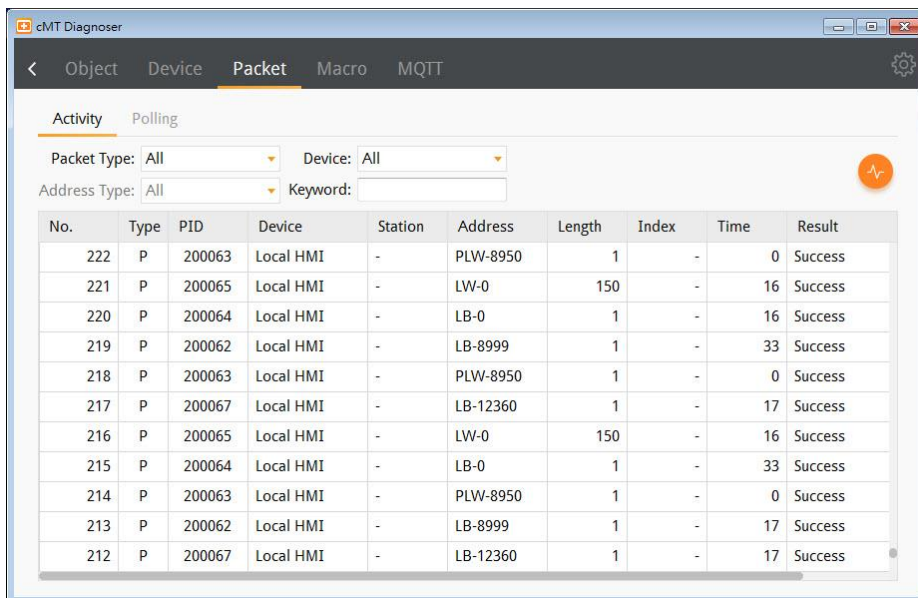### 31.3.2. Device Settings

1. The state and property of the device communicating with HMI is shown in the Device tab.

31.3.3. Packet Settings

1. In the Packet tab, select the ranges and then click ⚡ to observe communication status between HMI and PLC.
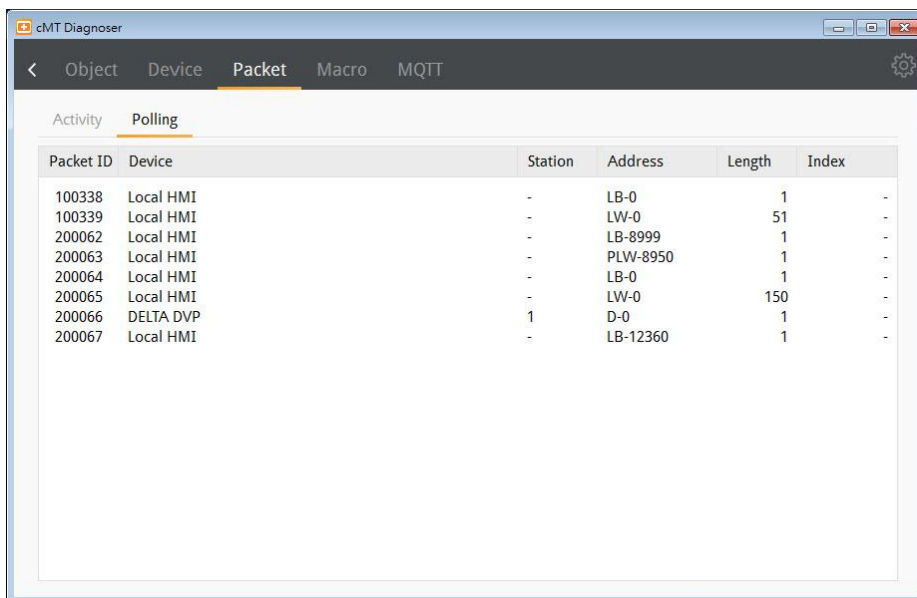

Item	Description
Packet	**All**
Type	Displays all packets in Activity area.
	Read

		Displays only Read packets in Activity area.
	Write	
		Displays only Write packets in Activity area.
	Polling	
		Displays only Polling packets in Activity area.
Device	**All**	
		Displays the information of Local HMI and PLC.
	Local HMI	
		Displays the information of Local HMI.
	PLC	
		Displays the information of PLC.
Station	Selects the PLC station number to be displayed.	
	(This function is disabled when selecting [All] in [Device].)	
Address Type	Selects all or a preferred address type to be displayed.	
	(This function is disabled when selecting [All] in [Device].)	
Keyword	Displays information that contains the keyword entered.	

When communication error occurs, information relating to the error will be highlighted in red in the Activity page, and the cause of error is shown in the Result column.
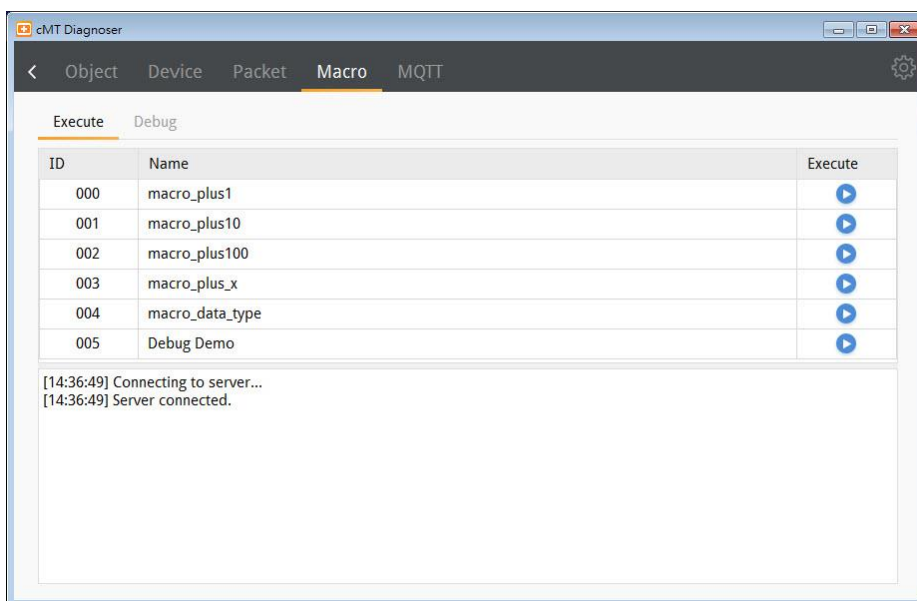

2. The Polling page shows the information of the polling packets periodically sent from HMI, including the start address, length…etc.

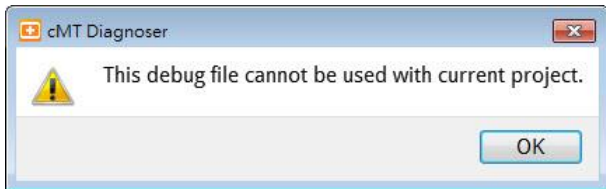| Item	Description
Packet ID	Use the Packet ID to find the invalid object.
Device	Displays HMI and PLC type.
Station	Displays PLC station number.
Address/Length	Displays the device type and the size of the packet.
Index	Displays the index register number of the object.

31.3.4. Macro Settings

1. Execute and test macros.

2. Macro Debugger Tool.
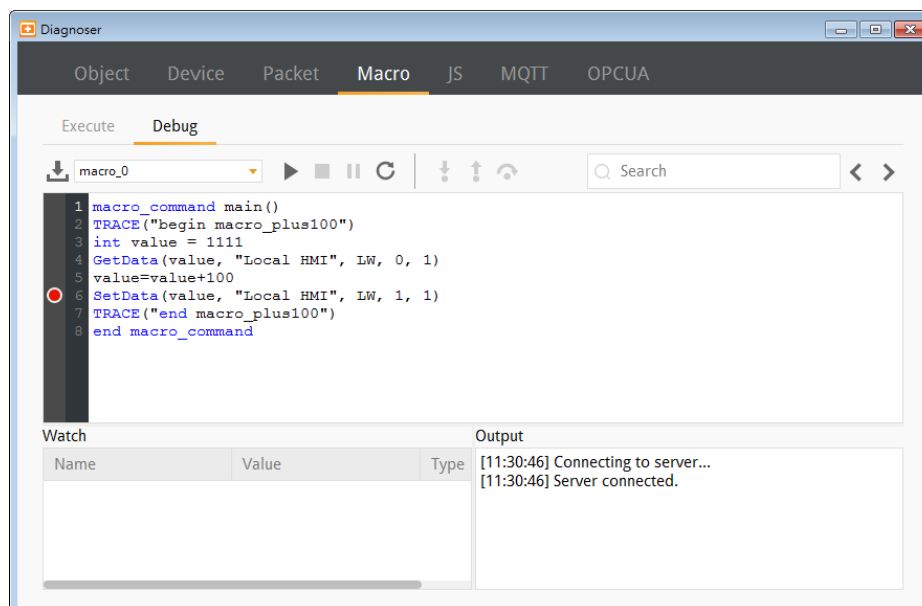
| Item	Description
Load macro debug file	Macro Debug File can be generated when compiling a project file in EasyBuilder Pro, and it is placed in the same directory as the compilation file. Macro Debug File has a file name extension of ".debug". When an unmatched debug file is loaded, the following message window shows.
Start debugging	Start debugging or execute macro from a breakpoint.
Stop debugging	Stop debugging.
Break	Pause macro debugging manually. A Breakpoint can be set at a statement by clicking the grey margin on the left-hand side of the statement. When a macro is launched, it will run until the Breakpoint, and wait for the next command.
Restart simulation	After running on-line / off-line simulation in macro work space, the macro can be revised directly in the Diagnoser window. After revising the macro, the simulation can be restarted by clicking Restart Simulation, and then the following confirmation window pops up.

EasyBuilder Pro V6.09.01
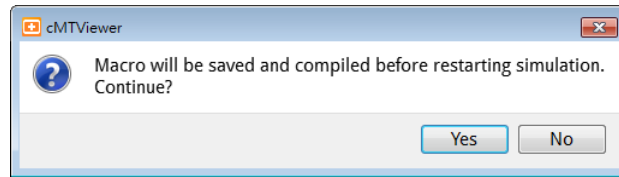
Step Into	Run the macro line by line. If the line contains a function, the diagnoser will enter the called function and continue line-by-line debugging from there.
Stop Out	Return to the line where the current function was called.
Step Over	Run the macro line by line. If the line contains a function, the function will be executed and the result is returned without debugging each line.
Search	Search for a macro by entering keywords.
Watch	When debugging, variables can be added here for watching the change of data.
Output	Shows system information or TRACE output.

Macro Debug in cMT Diagnoser can be opened in the following ways, each with different privilege requirements and operation steps.

Way1. [On-line Debug] or [Off-line Debug] within macro editing work space.
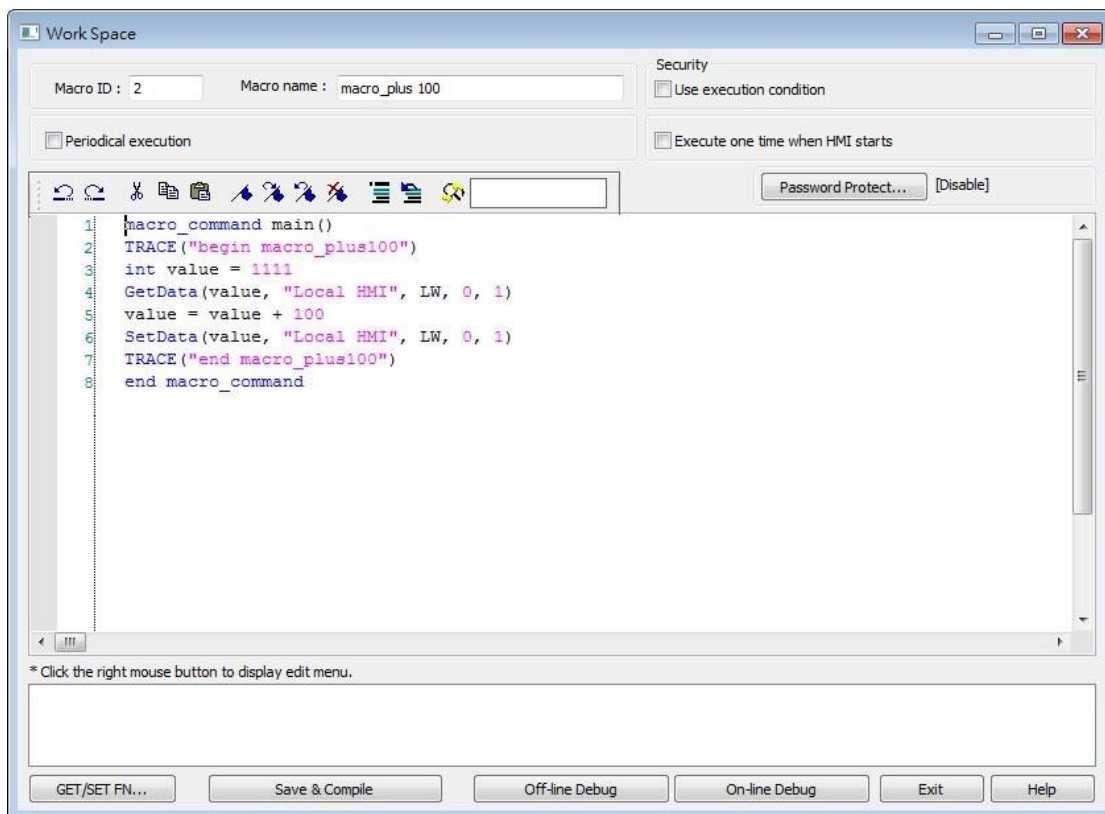
Click [On-line Debug] or [Off-line Debug]. Software will generate a Macro Debug File. Also,

macro editing work space window will close and be replaced by cMT Viewer simulation and cMT Diagnoser, the latter of which loads the Macro Debug File automatically. When the macro is being debugged in cMT Diagnoser's, the macro codes can actually be modified.

Upon completing macro debugging, close the cMT Viewer simulation. The macro editing work space should appear again, and it will ask whether you want to update the macro with the changes made in cMT Diagnoser.

Way2. On-line simulation / off-line simulation in EasyBuilder Pro.
The Macro Debug File is automatically generated after compiling the project file. When cMT Viewer simulation window is shown by running simulation in EasyBuilder Pro, open cMT Diagnoser from the right-click menu. The system imports the debug file automatically when it detects one. In simulation mode, macro can only be debugged but not changed.

Note: There must be at least one macro in the project for the Macro Debug File to be generated during compilation.
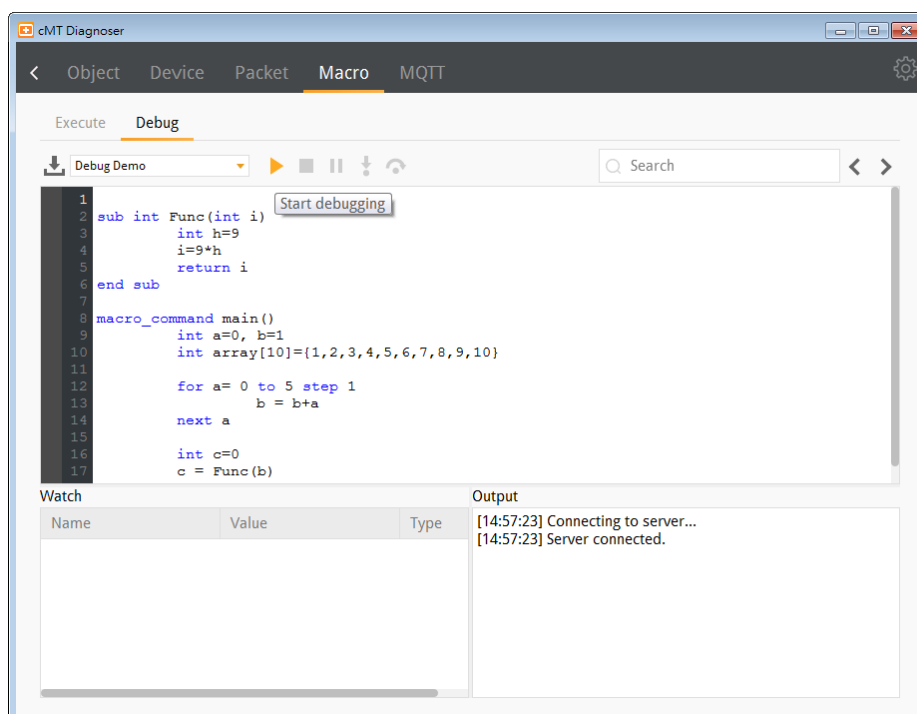
Way3. Downloading the project file to HMI and then running cMT Diagnoser.
The Macro Debug File should be manually imported in the Macro Debugger, and under this mode the macro can only be debugged but not changed.
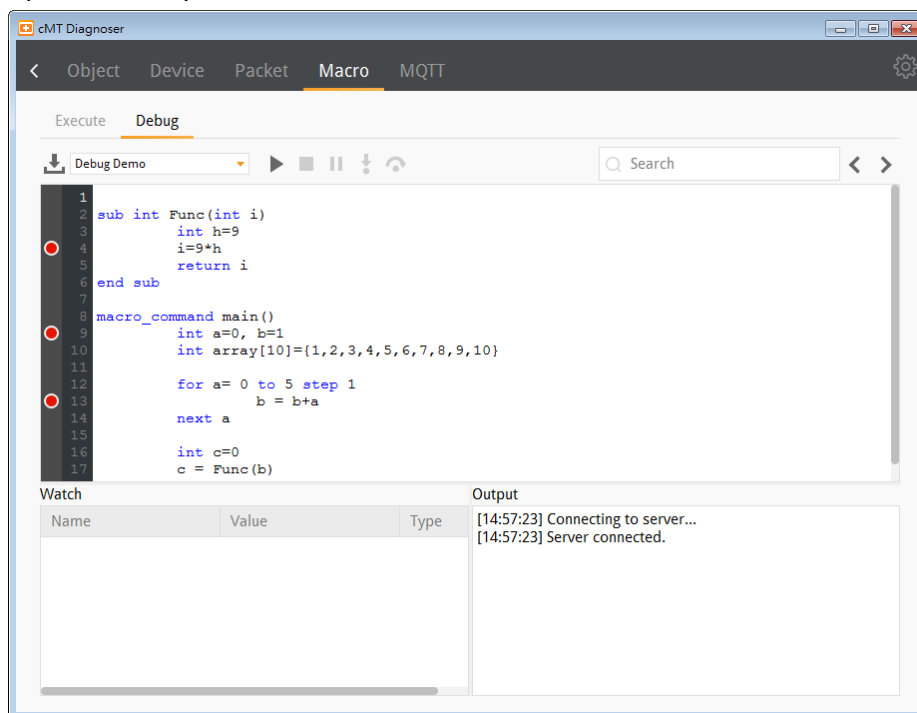
An Example of Macro Debugging
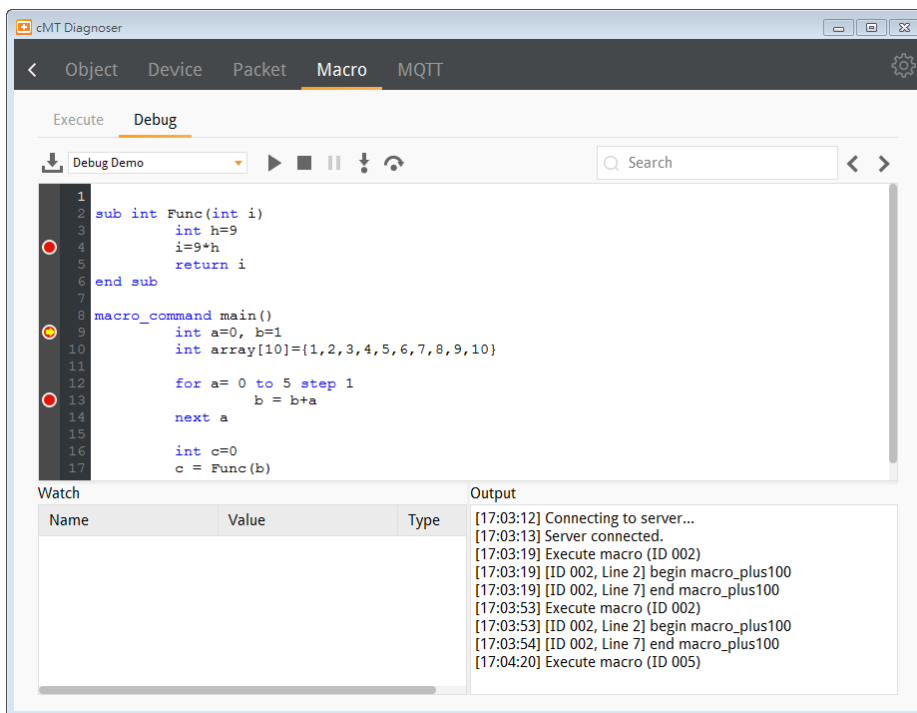The following steps show an example about how to debug macro.
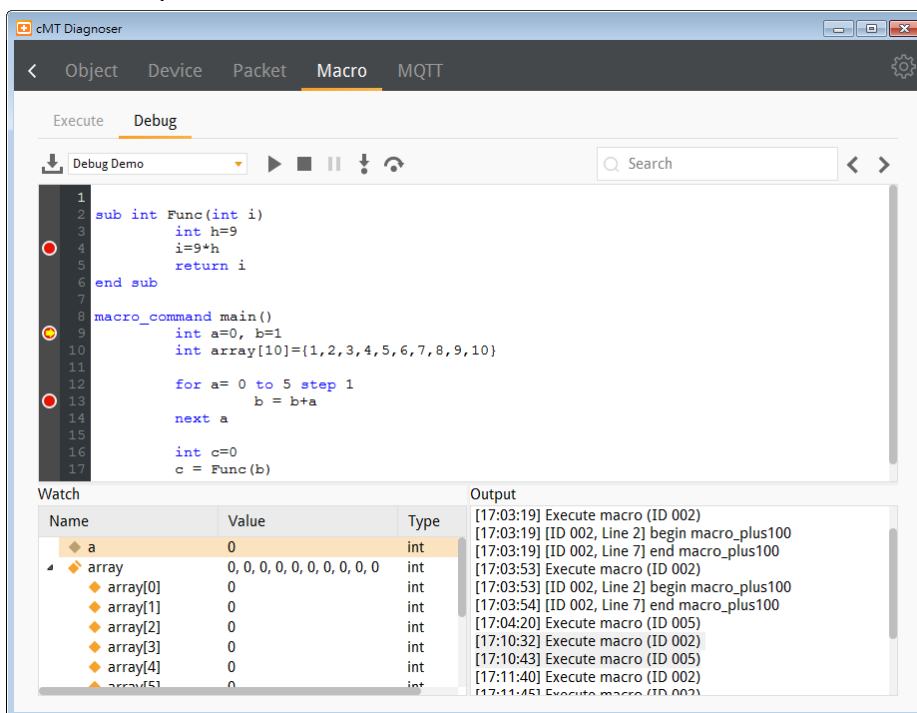
1. Click [Start Debugging].

2. Set Breakpoints in the grey margin on the left-hand side of the lines. The Breakpoints are represented by red dots.


3. After debugging is started, it will stop at the first breakpoint, now marked by a yellow arrow.

4. At this moment, variables can be added monitored in the Watch table as shown below. Contents of arrays can be monitored as well.


5. After the Start Debugging button is clicked once again, macro will run and stop at the next breakpoint. The changes of values are shown in the Watch list.
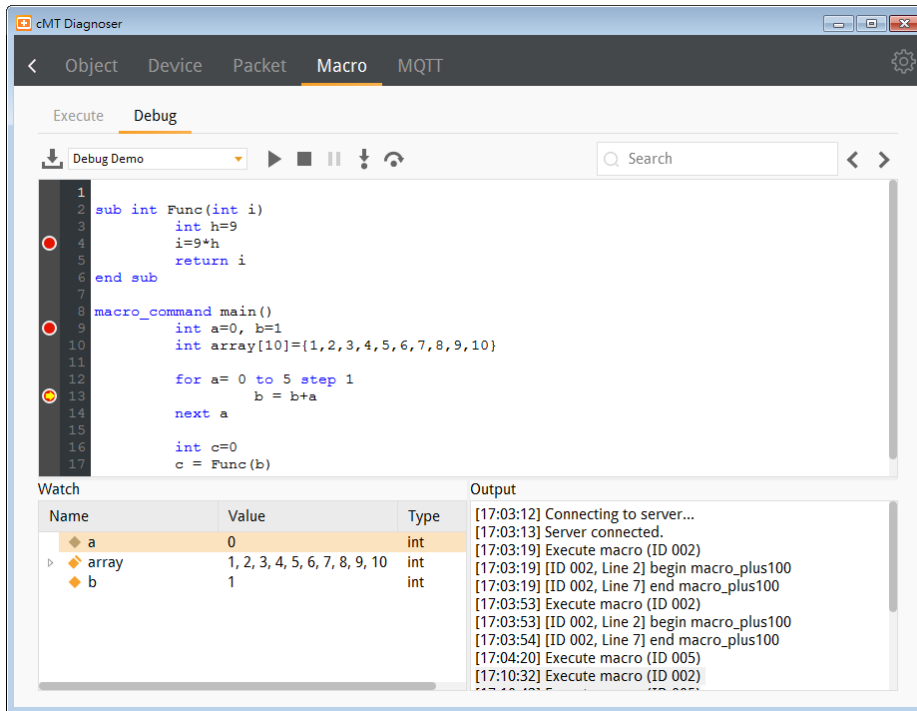
6. Click the Step Into button to run macro debugging line by line. If the line contains a function, cMT Diagnoser will enter the called function and continue line-by-line debugging from there. Changes of values can be observed in the Watch table.
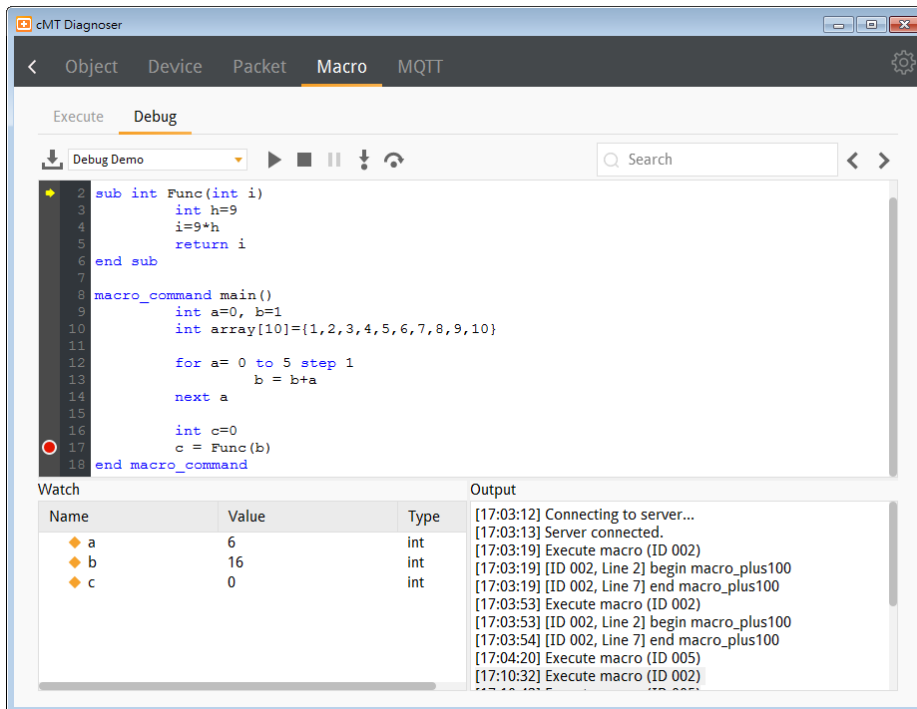
7. Click the Step Over button to run the macro line by line. Now, if the line contains a function, the result will be returned without debugging each line within the function call. When debugging meets a Breakpoint within the function call though, it stops there.
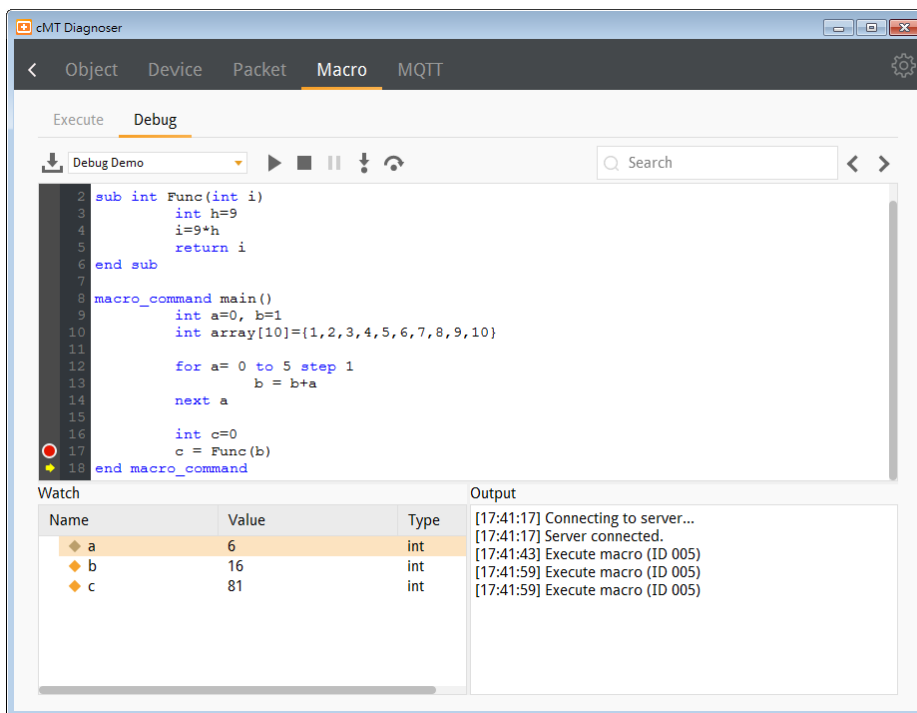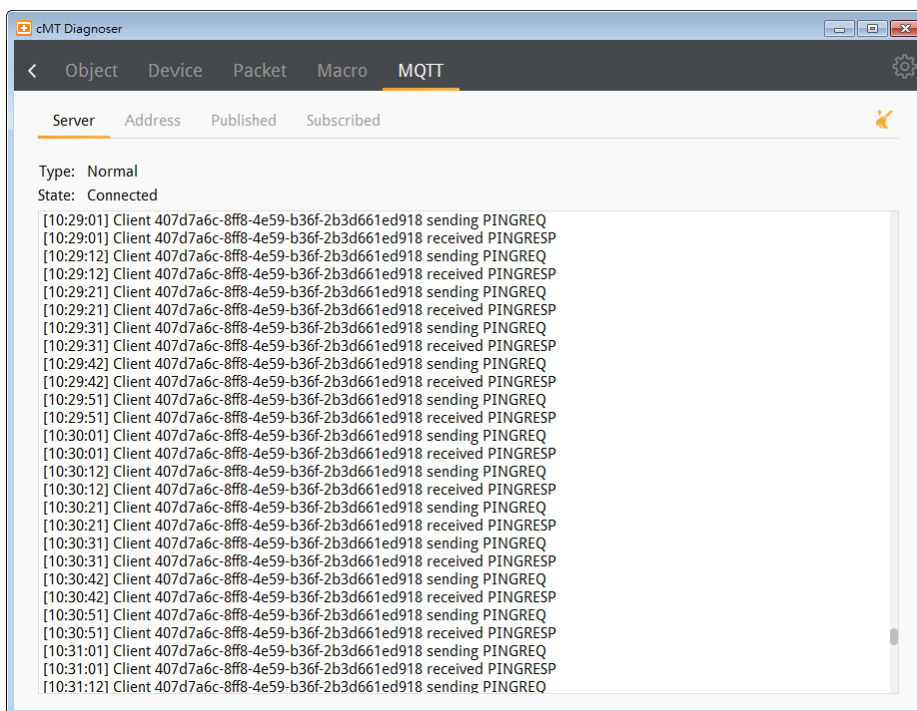
8. Click Stop Debugging button to stop.
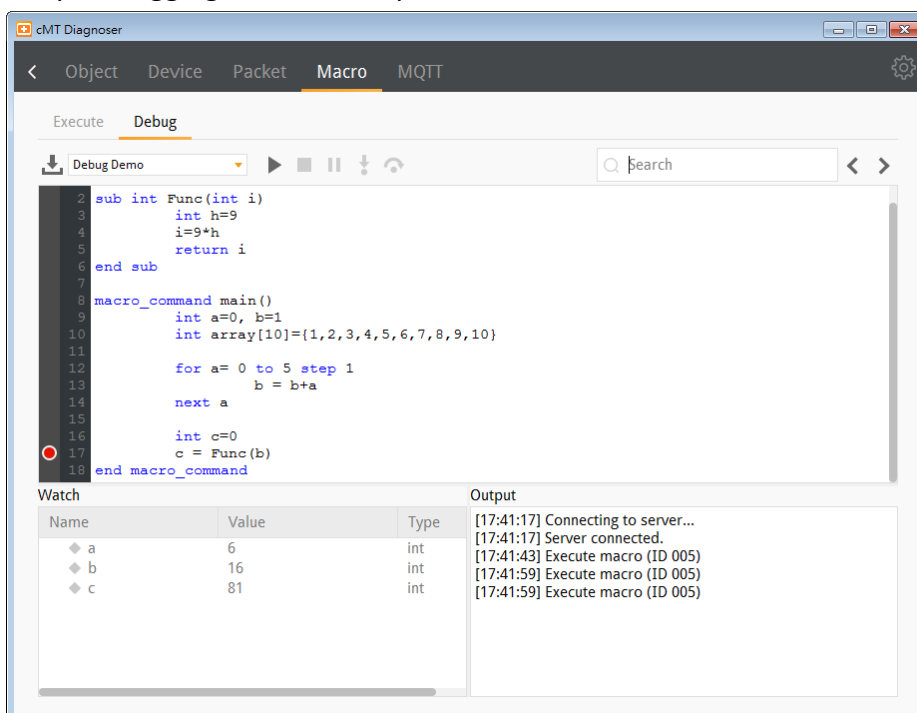

31.3.5. MQTT Settings

1. In Server tab the type and state of MQTT server is shown. Supported cloud services include: Normal / Azure IoT Hub / Sparkplug / Google Cloud IoT Core. Please note that AWS IoT is not supported.

2. In Address tab the Topics published and subscribed by the HMI are shown. Selecting the checkbox of a topic in this tab can make cMT Diagnoser receive all MQTT messages published and subscribed, allowing cMT Diagnoser to change address data for easier MQTT debugging. Only normal servers are supported.
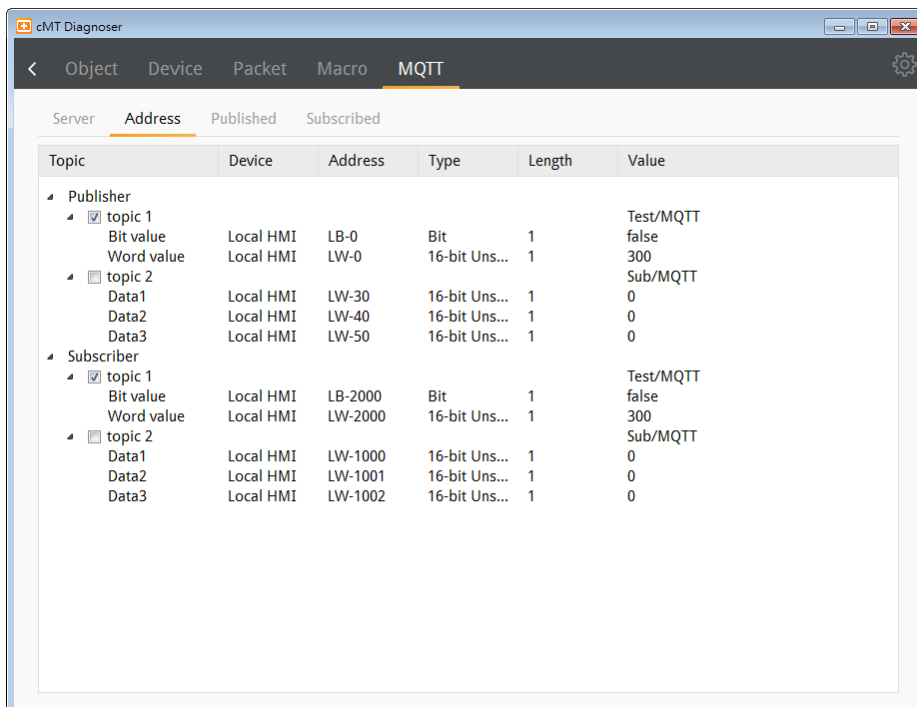
3. In Published tab the message published from HMI is shown. When Raw Data is used as content format, data will be shown in hexadecimal. Only normal servers are supported.
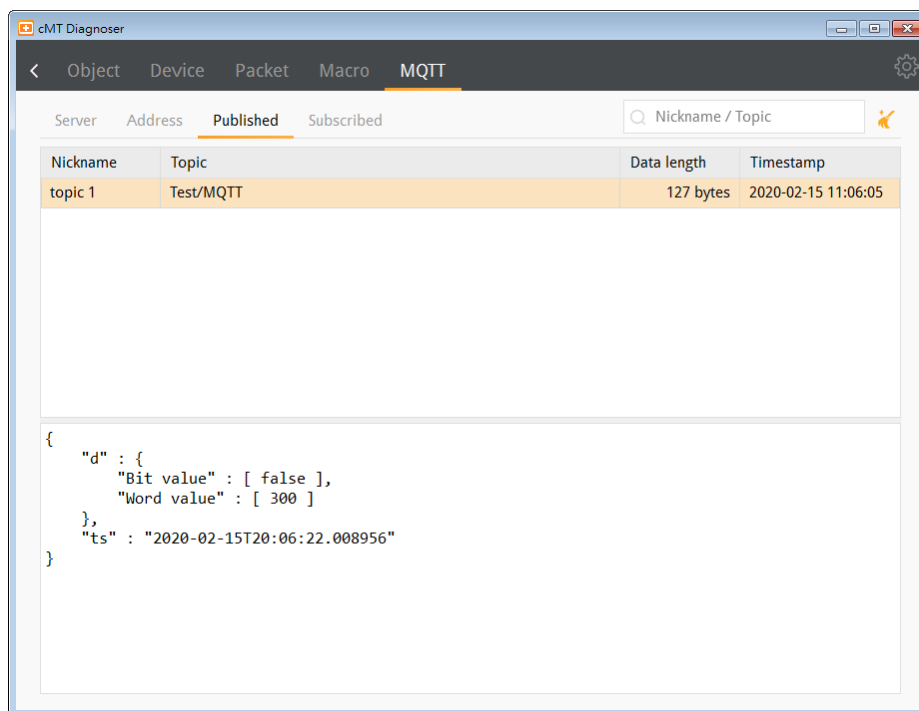
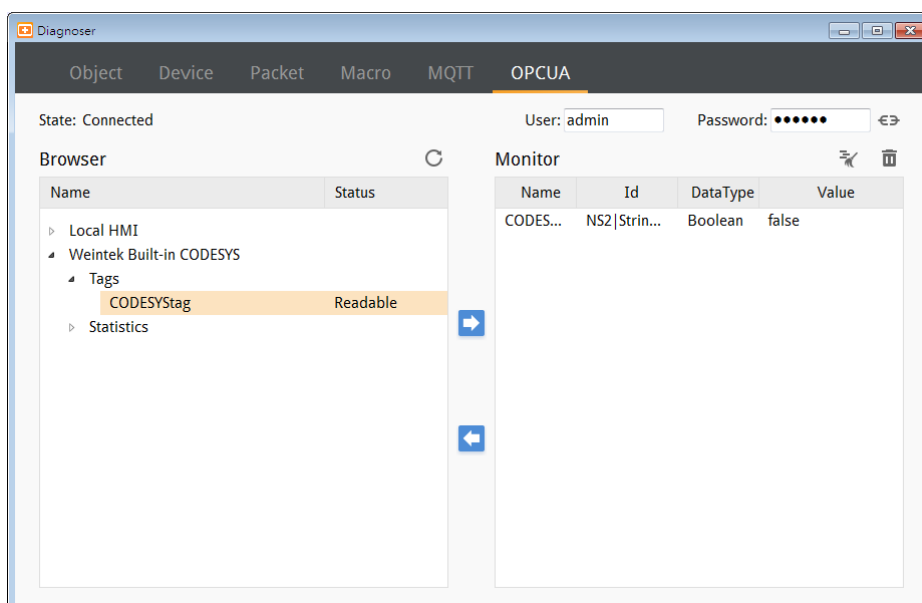4. In Subscribe tab the message subscribed by HMI is shown. When Raw Data is used as content format, data will be shown in hexadecimal. Only normal servers are supported.
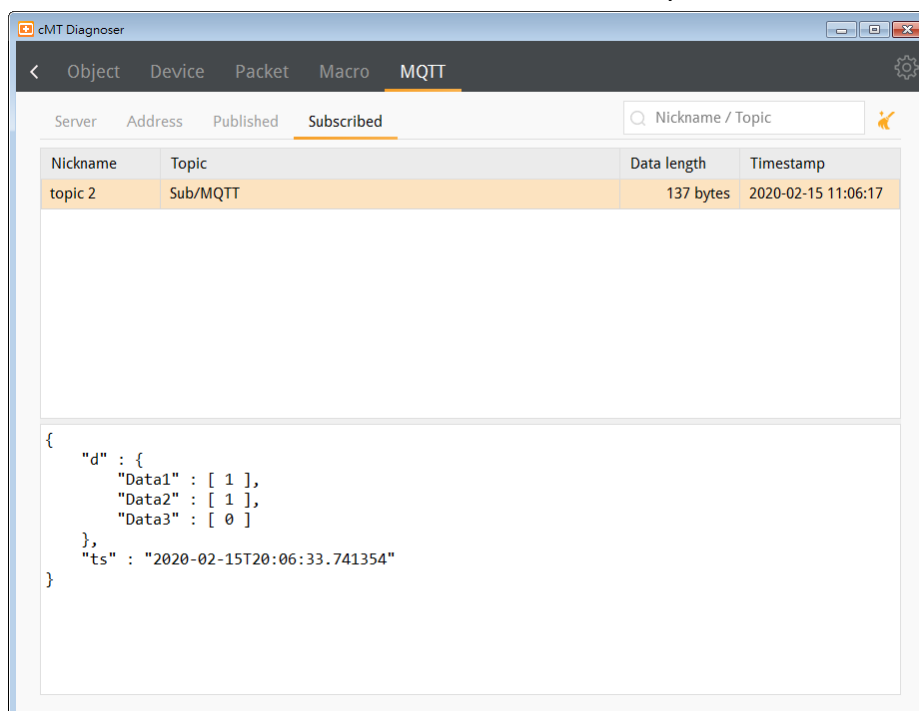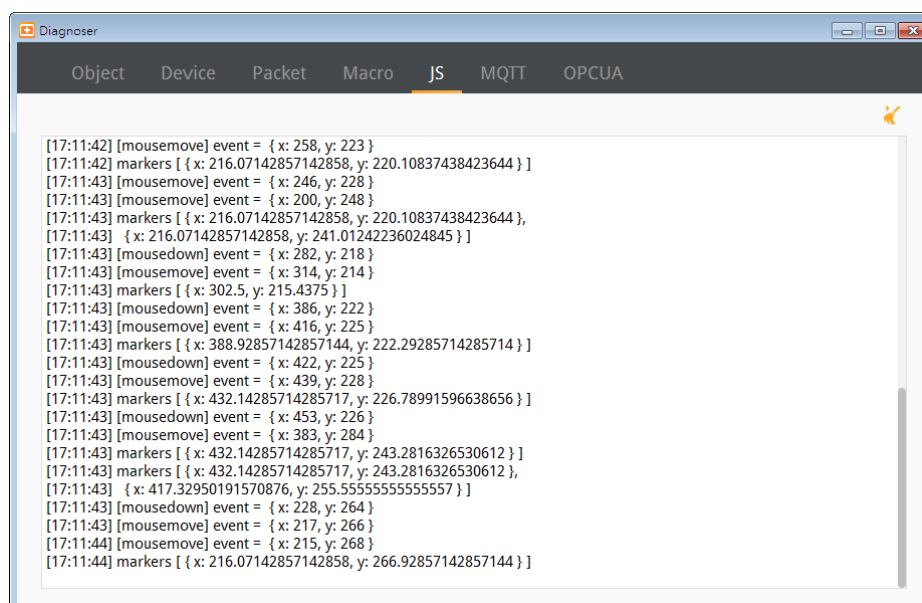

31.3.6. OPC UA Settings

1. Users can log in by entering the credentials to monitor values of OPC UA nodes.

31.3.7. JS Settings

1. In this tab users can view console messages of JS objects.

31.4. Additional Notes

1. cMT Diagnoser supports the monitoring all PLC addresses that HMI can access, including Tag PLC addresses.
2. Address values can be displayed in Decimal/HEX/Binary formats. Right-click anywhere on a monitor page to bring the menu up for the change.
3. When closing cMT Viewer with cMT Diagnoser opened, cMT Diagnoser will resume last state and pops up when cMT Viewer simulation window opens again. If Diagnoser is

closed before closing cMT Viewer, users should manually open Diagnoser in next simulation.

4. When monitoring cMT-SVR/cMT-SVRX models, connecting directly with cMT Diagnoser displays only global object information. To view information for objects within the current window, connect through cMT Viewer and access Diagnoser with a right-click.

5. When [Password Protect] is enabled in Macro Manager, the Macro Debug File cannot be generated after compilation.

6. Password Protect can be enabled for an individual macro, and two modes are available: Encrypted and Read Only. To debug a password protected macro, a password is required to unlock the macro in cMT Diagnoser. If the macro is protected in Read Only mode, it can only be viewed but not changed when debugging. If the macro is protected in Encrypted mode, it cannot be viewed at all.

7. When a macro is not successfully compiled, it will be put into the Macro Under Development list and will not be contained in the Macro Debug File.

8. cMT Diagnoser can only debug one macro at a time. When debugging a macro, executing another macro will not be effective.

9. When the breakpoint is placed on irrelevant lines, such as beside an empty line , variable declaration without initial value assignment, the breakpoint will be represented by a hollow red dot as line 7 below shows.