

WEINTEK LABS., INC.

JS Object

Add Comments to Screenshot

Demo Project

Contents

1. Overview & Operation	1
2. Setting up the Screen	3
3. Addresses	9

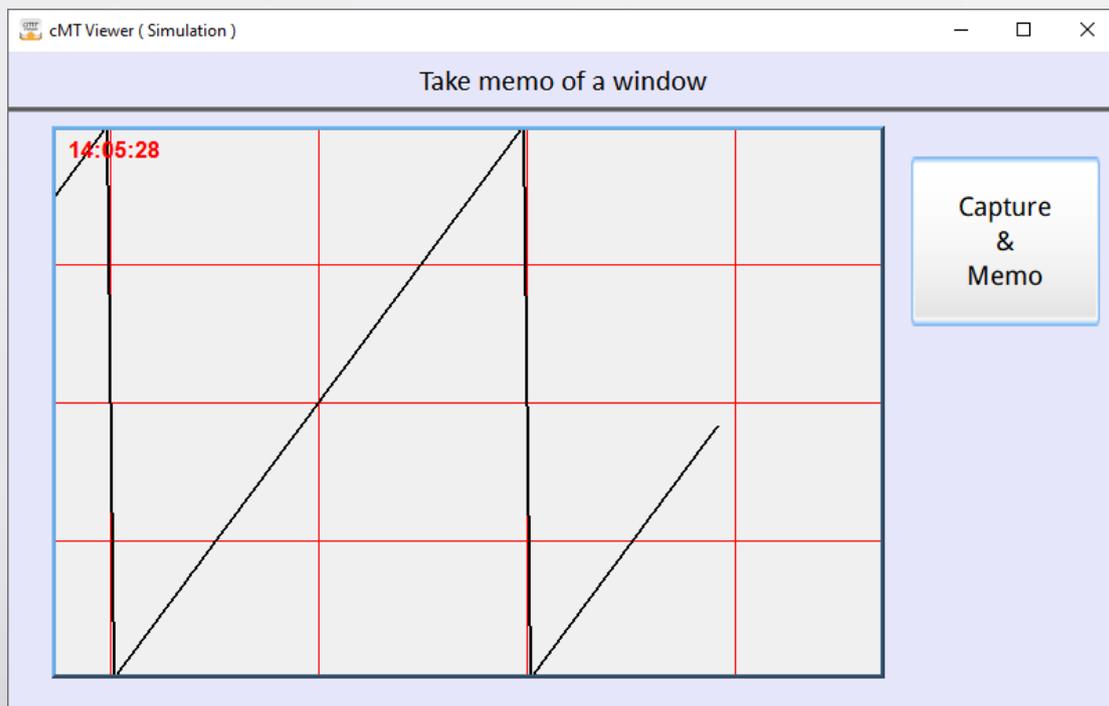
1. Overview & Operation

Overview

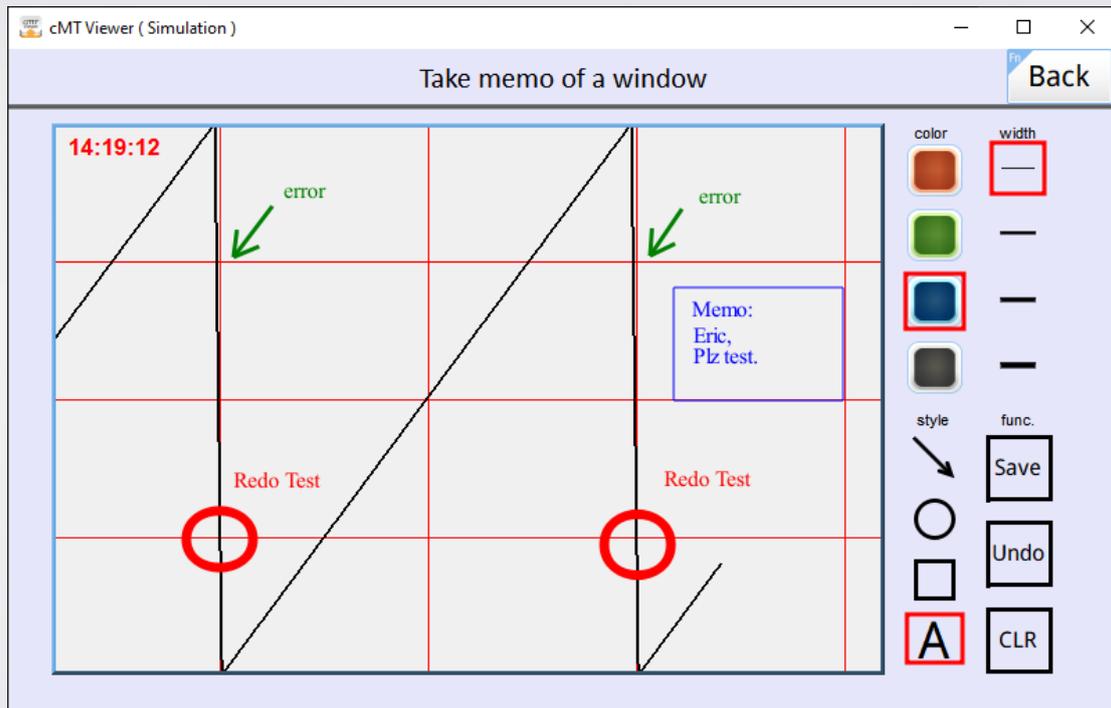
This demo project explains how to add notes or graphical elements to a screenshot of HMI project by using JS object.

Operation

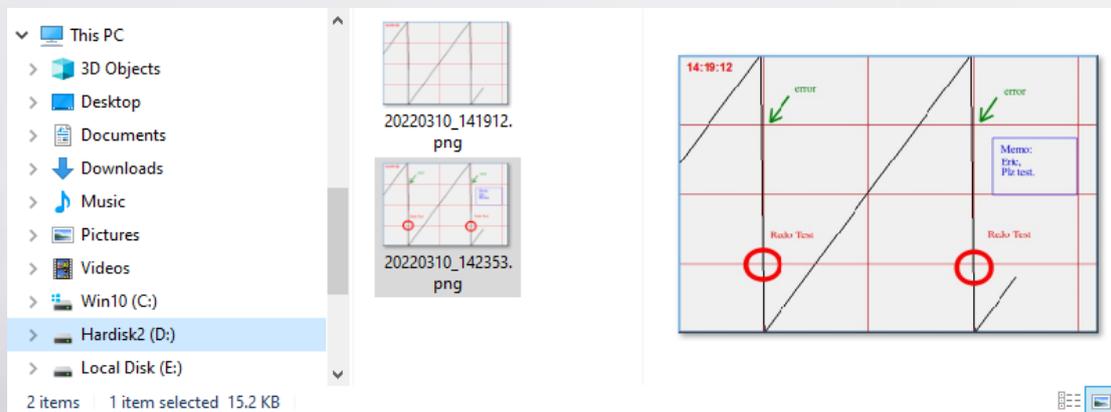
Step 1. Run off-line simulation or run the project on HMI. Wait a few seconds for data sampling and then click [Capture & Memo] button on the right hand side.



Step 2. Add notes, texts, and graphical elements.

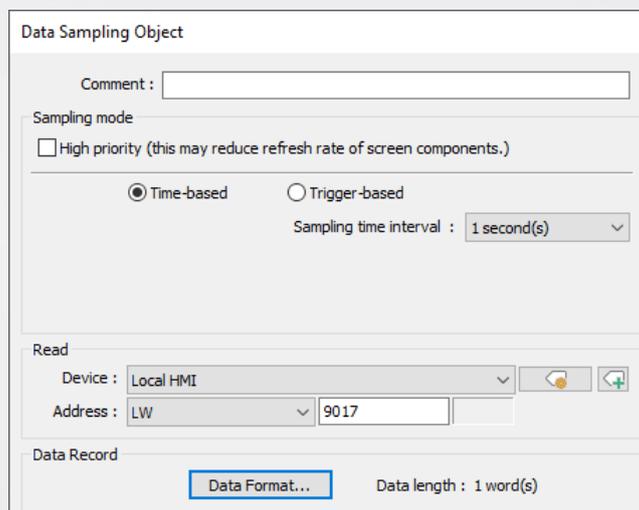


Step 3. Click [Save] and then the screenshot with the comments can be found in C:\EBpro\usb1 folder or in the USB disk on HMI.

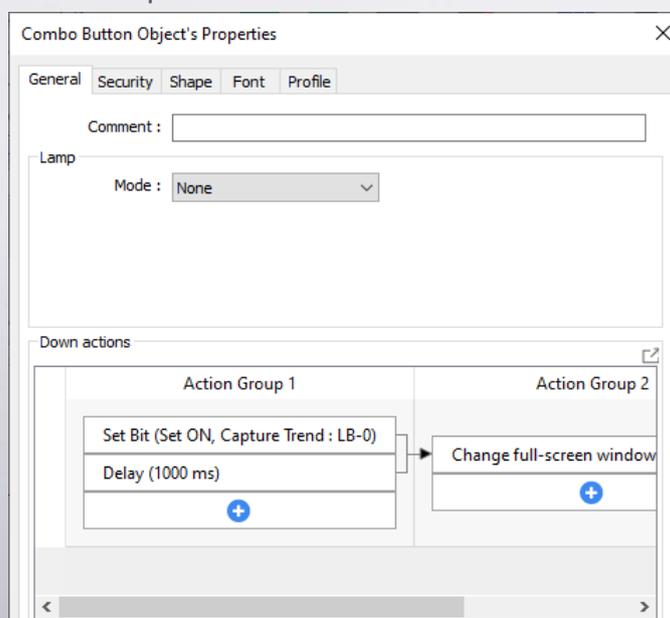


2. Setting up the Screen

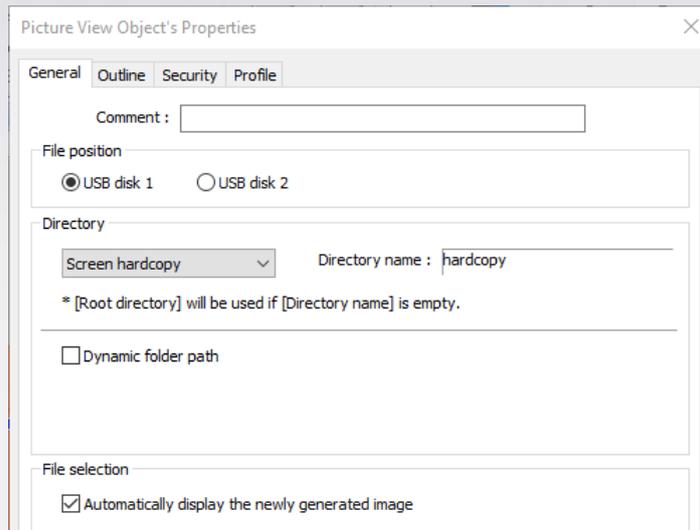
Step 1. Create a Data Sampling object and set address to LW-9017 to read the local second. The data changes every second.



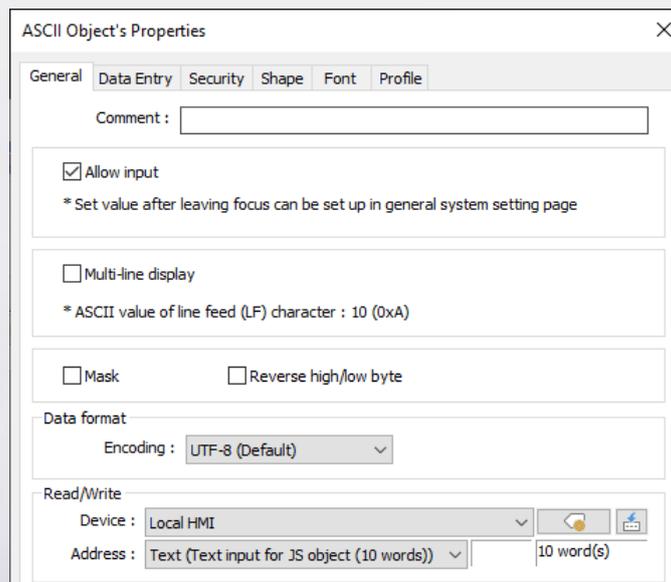
Step 2. In Window 14 create a Trend Display and a Combo Button. The Combo Button can trigger LB-0 to make PLC Control object take a screenshot of Trend Display, and then change window after the screenshot is captured.



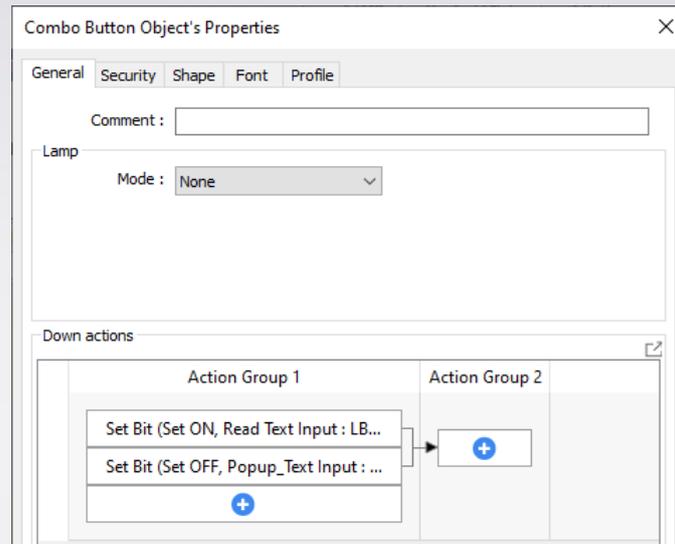
Step 3. In Window 11 create a Picture View object whose size and position are the same as of the Trend Display in Window 14. Select USB disk 1 as file position and select [Automatically display the newly generated image].



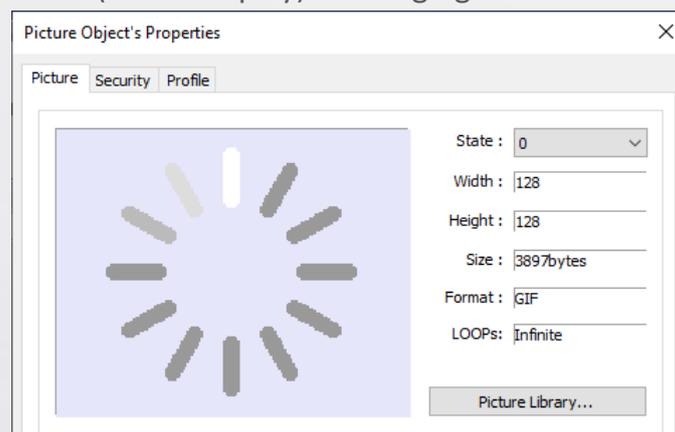
Step 4. Add Window 12 as a popup window and place an ASCII object in this window for JS object to read the entered string.



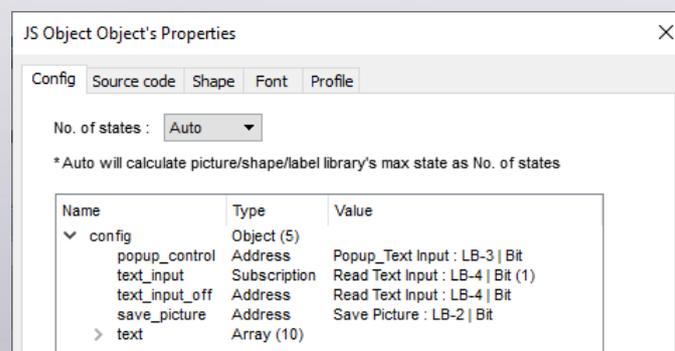
Create a Combo Button in this window. When this Combo Button is triggered, the popup window is closed and then the JS object reads the string.



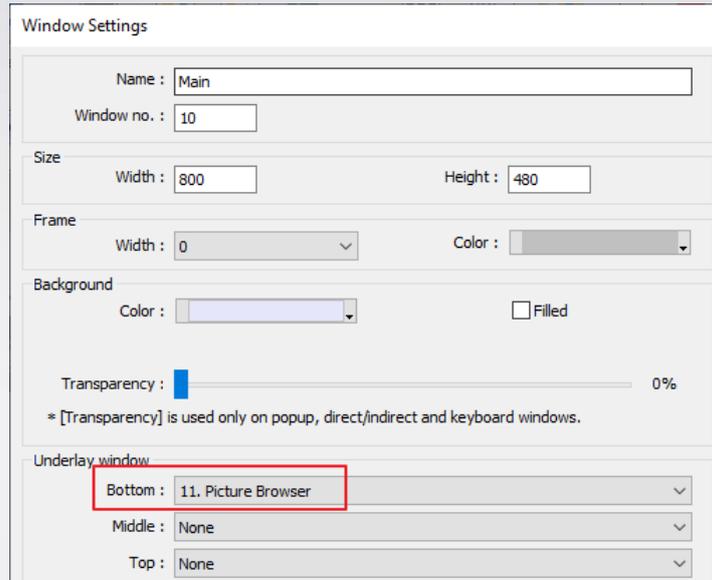
Step 5. Add Window 13 as a popup window and place a loading symbol image in this window. This window shows up during the time Window 14 (Trend Display) is changing to Window 10 (JS Object).



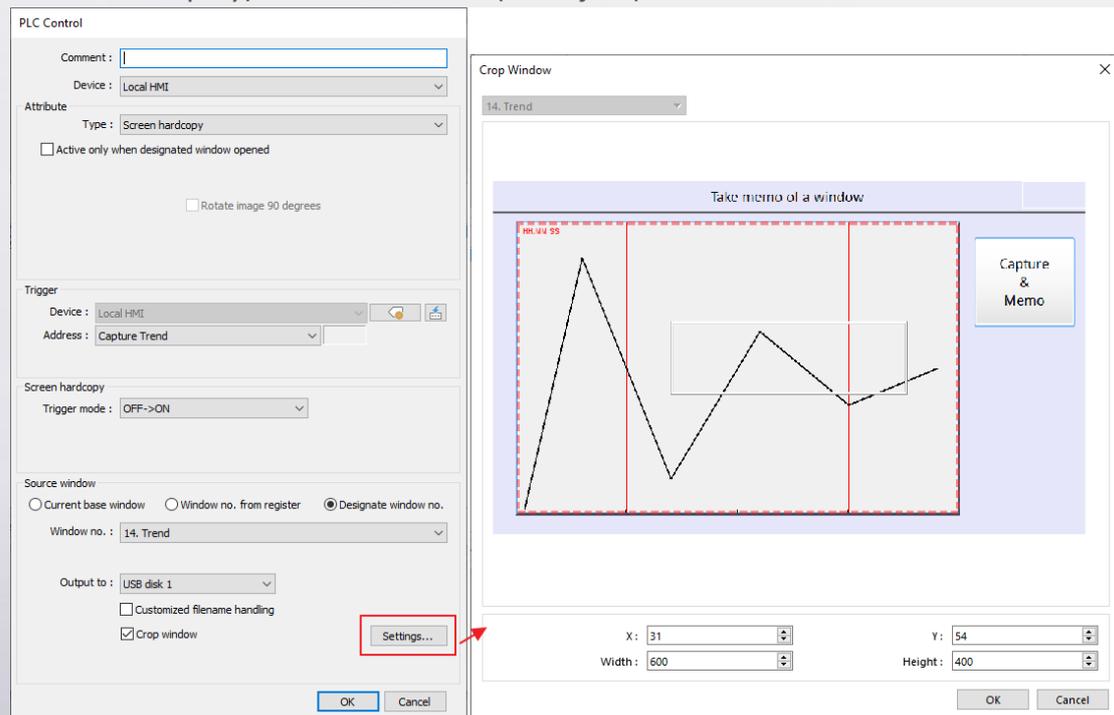
Step 6. Create two Direct Window objects to display Window 12 and Window 13. Create a JS object in Window 10 and set it as below.



Select Window 11 as underlay window. The size and position of the JS object should be the same as that of the Picture View object in Window 11.



Step 7. Create a PLC Control object and add two screen hardcopy commands to respectively take cropped screenshots of Window 14 (Trend Display) and Window 10 (JS Object).



Step 8. Create a JS object. (Only selected commands are shown below, please see the project for details.)

Line 1: *'this'* is the JS object. Through *'this.config'*<object> to obtain the value (/values) added in [Config] tab.

Line 2~20: Create multiple Canvases and MouseArea objects.

Line 22~40: Add Canvases and MouseArea objects to Widget.

JS Object

Line 42~57: Define each MouseArea object.

Line 59~70: Set variables and default values.

Line 72~80: Call 'onResponse' function of the '*self.config.text*' < Subscription > and register response callback to get notification when the state of LB-4 changes. When LB-4 is True, the JS object will execute binary2str function to convert data in LW-10~LW-19 to string and the data will be stored in Trace variable.

Line 82~93: Read data in LW-10~LW-19 and convert the data to string.

Line 96~109: Mousedown event.

Line 111~128: Mousemove event.

Line 133~152: Mouseup event.

Line 154~227: Set the Function to be executed when each MouseArea object takes input.

Line 229~249: The show_selected_color function shows the selected color.

Line 251~271: The show_selected_width function shows the selected width.

Line 273~293: The show_selected_style function shows the selected graphical style.

Line 295~316: The show_selected_func function shows the function to be executed.

Line 319~324: The set_dimension function defines the dimensions of mouse areas.

Line 326~341: The draw_arrow function draws an arrow.

Line 343~349: The draw_rectangle function draws a rectangle.

Line 351~359: The draw_circle function draws a circle.

Line 361~365: The draw_text function draws text.

Line 367~420: The redraw function. This function is used to draw a graphical style according to the data in Trace array.

Line 422~433: The trace_push function. This function pushes drawing parameters (e.g. start position, end position, style, color, width, string) to trace variable.

```
VAR SELF = THIS;  
VAR SELECT_CANVAS = NEW CANVAS();  
VAR CTX = NEW CANVAS();  
VAR REDRAW_CTX = NEW CANVAS();  
VAR MOUSEAREA = NEW MOUSEAREA();  
VAR RED_MOUSEAREA = NEW MOUSEAREA();  
VAR GREEN_MOUSEAREA = NEW MOUSEAREA();  
VAR BLUE_MOUSEAREA = NEW MOUSEAREA();  
VAR BLACK_MOUSEAREA = NEW MOUSEAREA();  
VAR WIDTH1_MOUSEAREA = NEW MOUSEAREA();  
VAR WIDTH2_MOUSEAREA = NEW MOUSEAREA();  
VAR WIDTH3_MOUSEAREA = NEW MOUSEAREA();  
VAR WIDTH4_MOUSEAREA = NEW MOUSEAREA();  
VAR ARROW_MOUSEAREA = NEW MOUSEAREA();  
VAR CIRCLE_MOUSEAREA = NEW MOUSEAREA();  
VAR RECTANGLE_MOUSEAREA = NEW MOUSEAREA();  
VAR TEXT_MOUSEAREA = NEW MOUSEAREA();  
VAR SAVE_MOUSEAREA = NEW MOUSEAREA();  
VAR UNDO_MOUSEAREA = NEW MOUSEAREA();  
VAR CLEAR_MOUSEAREA = NEW MOUSEAREA();
```

```
THIS.WIDGET.ADD(SELECT_CANVAS);  
THIS.WIDGET.ADD(CTX);  
THIS.WIDGET.ADD(REDRAW_CTX);  
THIS.WIDGET.ADD(MOUSEAREA);  
THIS.WIDGET.ADD(RED_MOUSEAREA);  
THIS.WIDGET.ADD(GREEN_MOUSEAREA);  
THIS.WIDGET.ADD(BLUE_MOUSEAREA);  
THIS.WIDGET.ADD(BLACK_MOUSEAREA);  
THIS.WIDGET.ADD(WIDTH1_MOUSEAREA);  
THIS.WIDGET.ADD(WIDTH2_MOUSEAREA);  
THIS.WIDGET.ADD(WIDTH3_MOUSEAREA);  
THIS.WIDGET.ADD(WIDTH4_MOUSEAREA);  
THIS.WIDGET.ADD(ARROW_MOUSEAREA);  
THIS.WIDGET.ADD(CIRCLE_MOUSEAREA);  
THIS.WIDGET.ADD(RECTANGLE_MOUSEAREA);  
THIS.WIDGET.ADD(TEXT_MOUSEAREA);  
THIS.WIDGET.ADD(SAVE_MOUSEAREA);  
THIS.WIDGET.ADD(UNDO_MOUSEAREA);  
THIS.WIDGET.ADD(CLEAR_MOUSEAREA);
```

```
SET_DIMENSION(MOUSEAREA, 0, 0, 610, CTX.HEIGHT);  
SET_DIMENSION(RED_MOUSEAREA, 615, 19, 42, 40);  
SET_DIMENSION(GREEN_MOUSEAREA, 615, 65, 42, 40);  
SET_DIMENSION(BLUE_MOUSEAREA, 615, 114, 42, 40);
```

3. Addresses

The addresses of key objects used in this demonstration are listed below, please adjust as necessary.

Object	Address	Object ID	Description
Window 10			
JS Object	LB-2, LB-3, LB-4, LW-10 ~ LW-19	CO_0	Takes screenshot of Window 10. Pops up Window 12. Reads ASCII object's string. ASCII object addresses.
Direct Window	LB-1 LB-3	WC_1 WC_0	Displays Window 13. Displays Window 12.
Function Key		FK_0	Returns to Window 14.
Action Trigger	LB-0, LB-1	AP_0	Resets bits in Capture Trend address and Popup Loading Icon address.
Window 11			
Picture View		PV_0	
Window 12			
ASCII	LW-10 ~ LW-19	AE_0	
Combo Button	LB-3, LB-4	CB_0	Closes Window 12 and triggers JS object to read the string.
Window 13			
Picture		GP_0	
Window 14			
Trend Display		TD_0	Displays sampling result.
Combo Button	LB-0	CB_0	Takes screenshot and changes window.