

威綸科技股份有限公司

JS 物件

在畫面截圖上加上註解

工程檔案範例

目錄

1. 簡介與操作.....	1
2. 設定程序.....	3
3. 位址設定.....	9

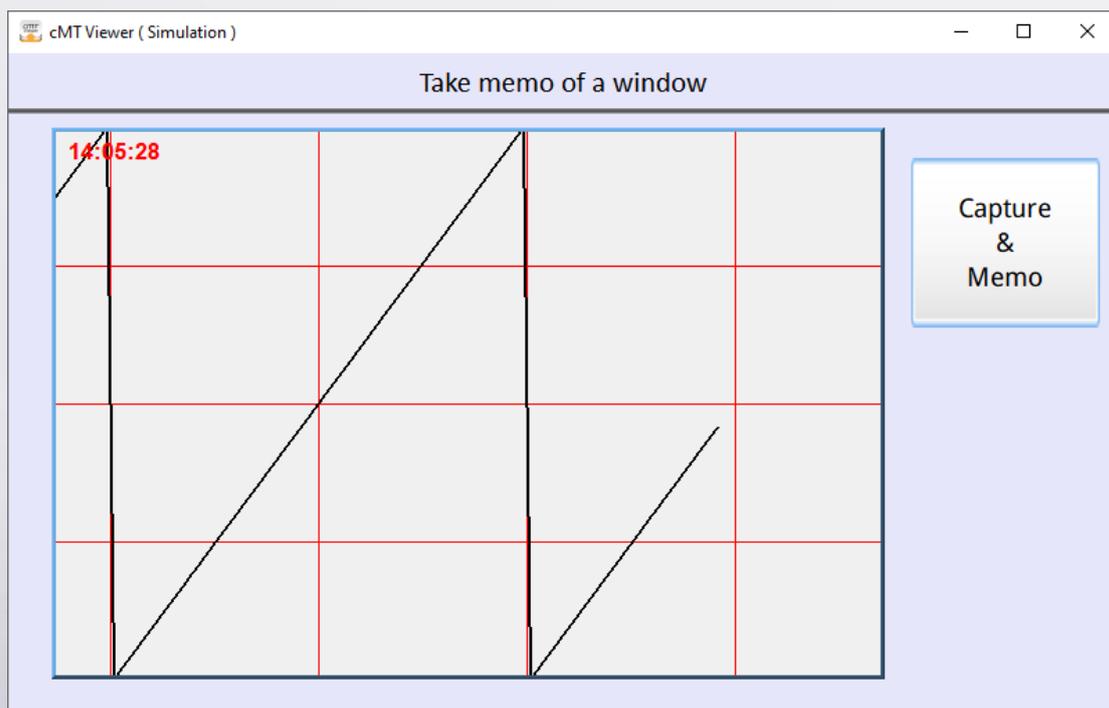
1. 簡介與操作

簡介

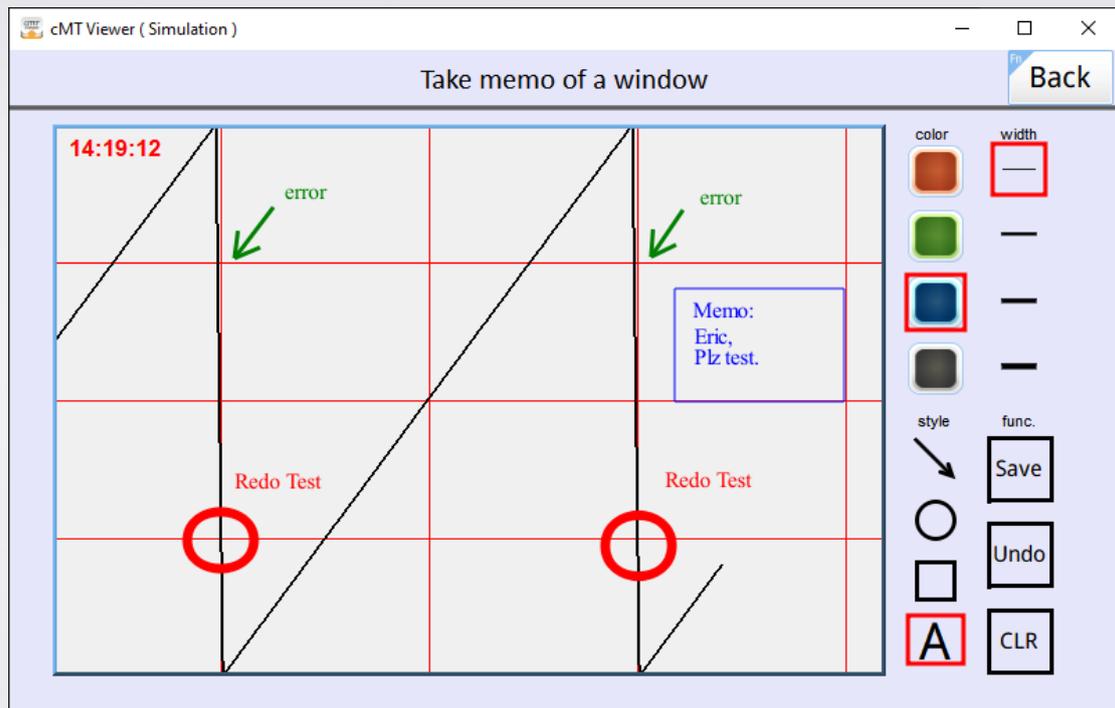
畫面截圖將 HMI 運行的畫面保存，此時若想額外截圖上加上簡易註解及標示，可以搭配 JS 物件達成。

操作

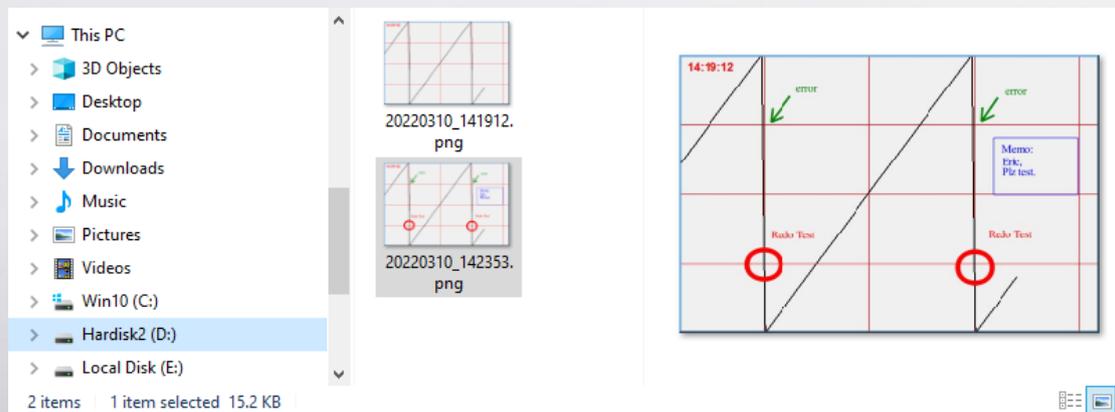
步驟 1. 開啟並執行工程檔案離線模擬，或將工程檔案下載至人機。讓 HMI 短暫運行採集資料取樣，接著按右邊的 [Capture & Memo] 按鈕。



步驟 2. 加上註解與標示。

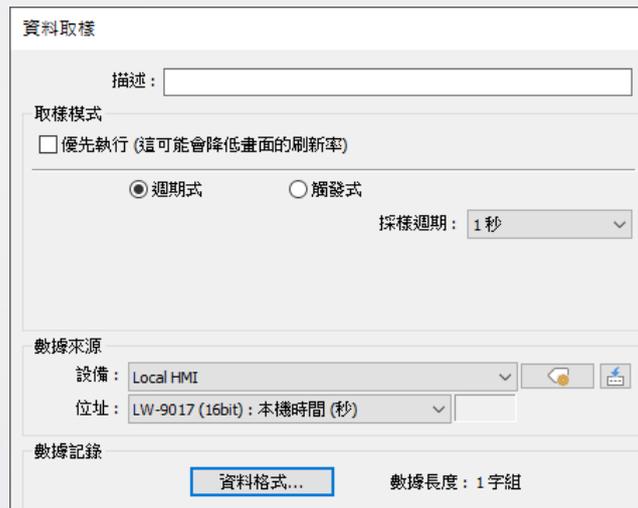


步驟 3. 觸發 [Save]，可在 C:\EBpro\usb1 資料夾或 HMI 上的 USB 隨身碟內找到註解後的圖片。



2. 設定程序

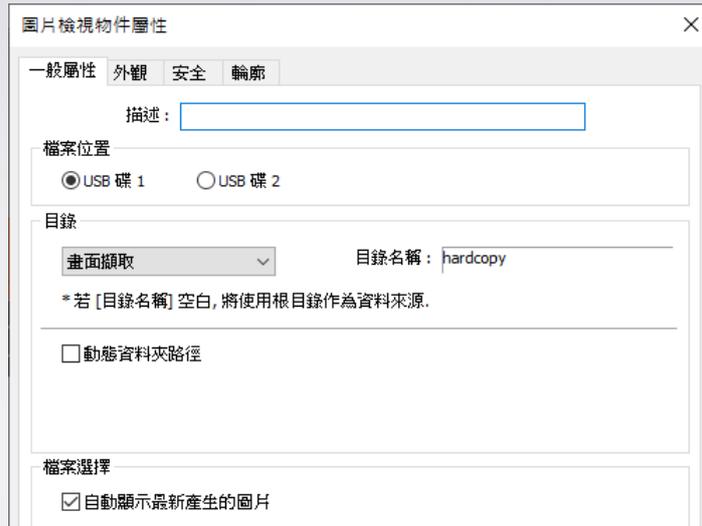
步驟 1. 建立一個資料取樣物件，地址設定為 LW-9017。此地址是讀取 HMI 時間的秒，因此無須額外控制數據也能自動變化。



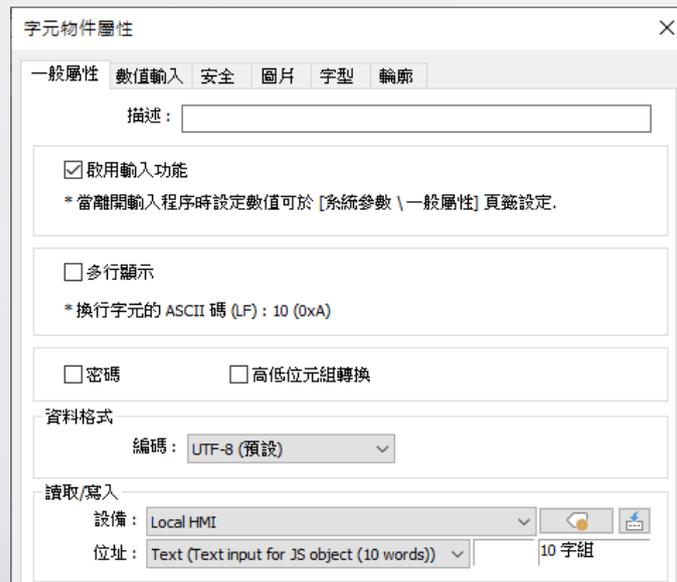
步驟 2. 在視窗 14 建立一個趨勢圖以及一個複合式切換開關。複合式切換開關會觸發 LB-0 (搭配 PLC 控制物件擷取視窗 14 的趨勢圖)，擷取完畢後換頁。



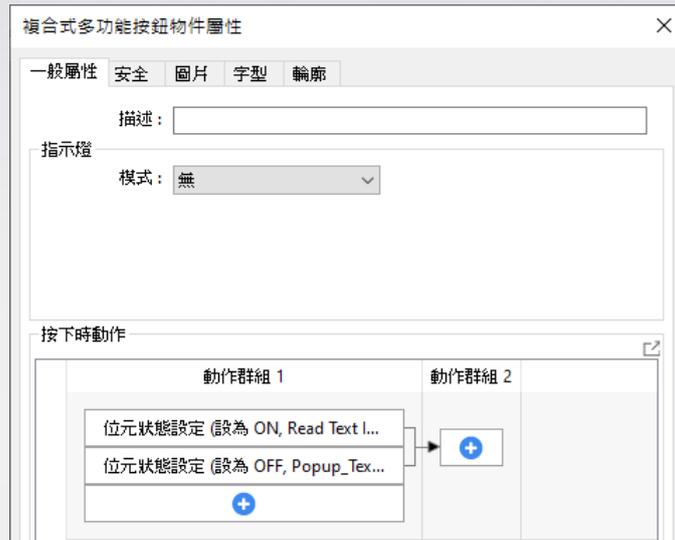
步驟 3. 在視窗 11 建立一個圖片檢視物件。圖片檢視物件與視窗 14 的趨勢圖大小及位置相同。檔案位置設定 USB1，並勾選 [自動顯示最新產生的圖片]。



步驟 4. 將視窗 12 建立為彈出視窗，放一個字元地址供 JS 物件讀取輸入字串。



再建立一個複合式切換開關，觸發此複合式切換開關會關閉彈出視窗及觸發 JS 物件存取字串。



步驟 5. 將視窗 13 建立為彈出視窗，放一個載入的圖片。此視窗會被用來當視窗 14（趨勢圖視窗）切換到視窗 10（JS 物件視窗）的緩衝畫面。



步驟 6. 建立兩個直接視窗物件，用來顯示視窗 12 與視窗 13。在視窗 10 建立 JS 物件，及數個圖片顯示動作。



設定重疊視窗 11。JS 物件的大小位置與視窗 11 的圖片檢視物件相同。

視窗設定

視窗名稱: Main
視窗編號: 10

大小
寬度: 800 高度: 480

外框
寬度: 0 顏色: [Color Picker]

背景
顏色: [Color Picker] 填色

透明度: [Slider] 0%

* [透明度] 只適用於彈出視窗, 直接/間接視窗和鍵盤視窗。

重疊視窗
 底層: 11. Picture Browser
 中層: 無
 頂層: 無

* 使用 [檢視] > [階層不透明度] 可於編輯時使重疊視窗呈現半透明。

步驟 7. 在 PLC 控制物件中，建立兩個截圖指令，一個截圖視窗 14 的趨勢圖，一個截圖視窗 10 編輯完的畫面。由於只需要趨勢圖繪圖的部分，因此兩個都啟用截圖範圍。

PLC 控制

描述: [Text Box]
設備: Local HMI

屬性
控制類型: 畫面擷取
 只在指定的視窗被啟動時才執行
 圖形旋轉 90 度

觸發
設備: Local HMI
位址: Capture Trend

畫面擷取
觸發模式: OFF->ON

來源視窗
 當前的基本視窗 視窗編號取自暫存器 指定視窗編號
視窗編號: 14. Trend

輸出: USB 碟 1
 自訂檔案名稱管理
 裁切視窗

設定...

確定 取消

裁切視窗

14. Trend

Take memo of a window

HR.UNI 95

Capture & Memo

X: 31 Y: 54
寬度: 600 高度: 400

確定 取消

步驟 8. 建立 JS 指令。(僅擷取部分指令，詳細請參考工程檔案)

- Line 1: `'this'` 即代表 JS 物件。透過 `'this.config' <object>` 可以取得在 Config 設定頁面中所加入的值(/values)。

- Line 2~20: 新增多個 Canvas、MouseArea 物件。
- Line 22~40: 將 Canvas、MouseArea 物件加入到 Widget。
- Line 42~57: 定義不同的 MouseArea 物件的範圍。
- Line 59~70: 建立變數並設定變數與函數預設值。
- Line 72~80: 使用 Subscription 的 onResponse Method 讓 LB-4 在有變化時執行接續的 callback 函式。當 LB-4 為 True 時，JS 物件會執行 binary2str 函數將 LW-10~LW-19 的資料轉換為字串並儲存到 Trace 變數中。
- Line 82~93: 讀取 LW-10~LW-19 的資料並轉換為字串。
- Line 96~109: 設定滑鼠觸發時的動作。
- Line 111~128: 設定滑鼠移動時的動作。
- Line 133~152: 設定滑鼠放開時的動作。
- Line 154~227: 設定不同的 MouseArea 物件的動作。
- Line 229~249: show_selected_color 函數圈選出選取的顏色。
- Line 251~271: show_selected_width 函數圈選出選取的線段寬度。
- Line 273~293: show_selected_style 函數圈選出選取的繪圖圖案。
- Line 295~316: show_selected_func 函數圈選出欲執行的功能。
- Line 319~324: set_dimension 函數定義各 MouseArea 物件的範圍。
- Line 326~341: draw_arrow 函數繪製箭頭圖案。
- Line 343~349: draw_rectangle 函數繪製矩形圖案。
- Line 351~359: draw_circle 函數繪製橢圓形圖案。
- Line 361~365: draw_text 函數繪製文字圖案。
- Line 367~420: redraw 函數參照 trace 陣列的數據繪製對應的圖案。
- Line 422~433: trace_push 函數將繪製所需的參數，例如：起始位置、結束位置、圖案、顏色、寬度、字串內容 push 到 trace 變數。

```
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. 位址設定

此範例所使用之物件位址皆列於下表，請依實際需要調整。

物件	位址	物件 ID	描述
視窗 10			
JS 物件	LB-2、 LB-3、 LB-4、 LW-10 ~ LW-19	CO_0	觸發截圖視窗 10、 觸發彈出視窗 12、 讀取字元物件的字串、 字元物件的地址來源
直接視窗	LB-1、 LB-3	WC_1 WC_0	用來顯示彈出視窗 13、 用來顯示彈出視窗 12
功能鍵		FK_0	將視窗切換回視窗 14
動作觸發	LB-0、 LB-1	AP_0	重置觸發截圖位元與直接 視窗 13 位元
視窗 11			
圖片檢視		PV_0	
視窗 12			
字元	LW-10 ~ LW-19	AE_0	
複合式切換開關	LB-3、 LB-4	CB_0	關閉彈出視窗 12 及觸發 JS 物件存取字串
視窗 13			
圖片		GP_0	
視窗 14			
趨勢圖		TD_0	顯示資料取樣的採樣結果
複合式切換開關	LB-0	CB_0	觸發截圖並切換視窗