

威綸科技股份有限公司

# JS 物件

## 動態顯示趨勢圖取樣數值

工程檔案範例

## 目錄

1. 簡介與操作.....	1
2. 設定程序.....	2
3. 位址設定.....	8

## 1. 簡介與操作

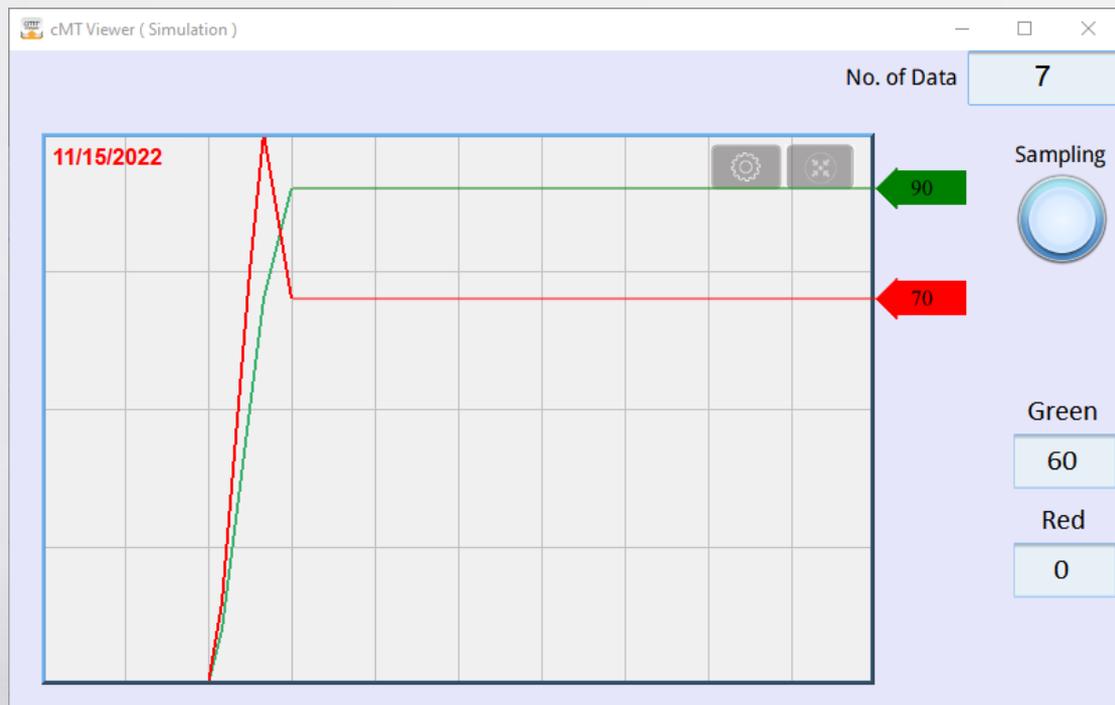
### 簡介

為了方便檢視趨勢圖上目前取樣數據的數值，可透過 JS 物件指出取樣數值。

### 操作

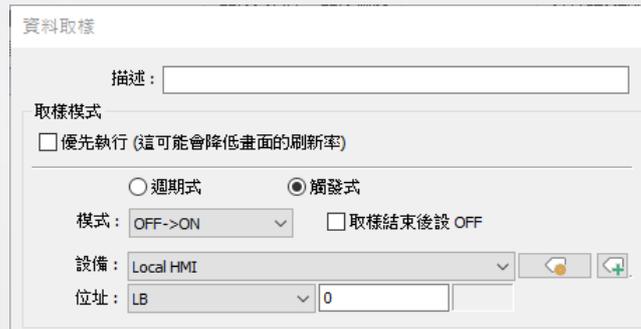
**步驟 1.** 開啟並執行工程檔案，並將工程檔案下載至人機。

**步驟 2.** 觸發 [Sampling] 進行資料取樣，右邊會顯示當下取樣數值。



## 2. 設定程序

**步驟 1.** 建立一個資料取樣物件，設定為觸發模式，控制地址為 LB-0。



資料取樣

描述:

取樣模式

優先執行 (這可能會降低畫面的刷新率)

週期式  觸發式

模式: OFF->ON  取樣結束後設 OFF

設備: Local HMI

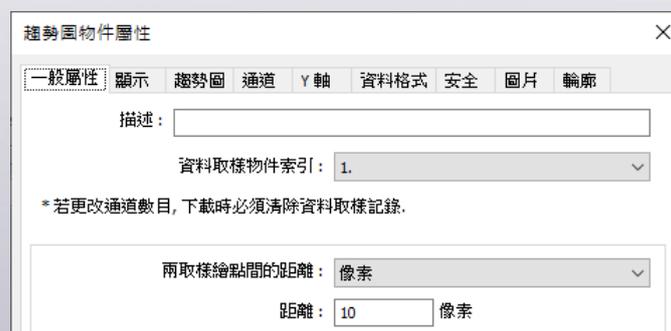
位址: LB 0

**步驟 2.** 設定取樣的資料，這裡使用從 LW-0 開始，取樣 2 個 16-bit Unsigned 地址。



區段	描述	讀取位址	資料類型	大小
1	16-bit Unsigned	LW-0	16-bit Unsigned	1字組
2	16-bit Unsigned		16-bit Unsigned	1字組

**步驟 3.** 建立趨勢圖物件。由於趨勢圖畫圖時，前幾個點位位置會有變動，為了方便計算線段繪製長度，繪點方式使用 [像素]。



趨勢圖物件屬性

一般屬性 | 顯示 | 趨勢圖 | 通道 | Y 軸 | 資料格式 | 安全 | 圖片 | 輪廓

描述:

資料取樣物件索引: 1

\*若更改通道數目, 下載時必須清除資料取樣記錄。

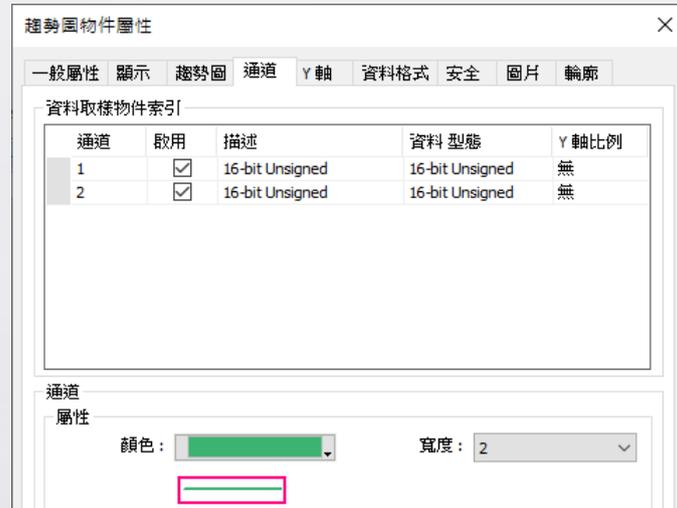
兩取樣繪點間的距離: 像素

距離: 10 像素

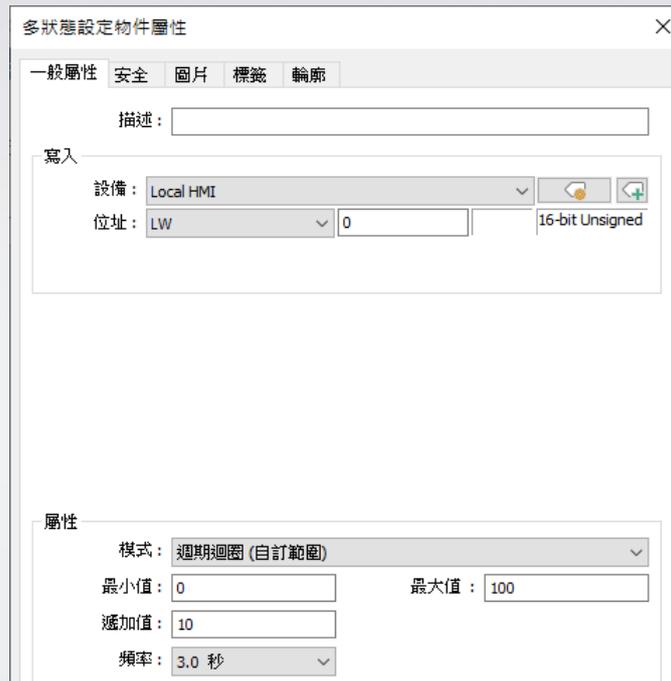
**步驟 4.** 為了避免繪圖過程中，X 軸或 Y 軸變動造成使用 JS 物件繪製線段位置錯誤，在趨勢圖的 [顯示] 分頁中取消 X 軸與 Y 軸縮放。



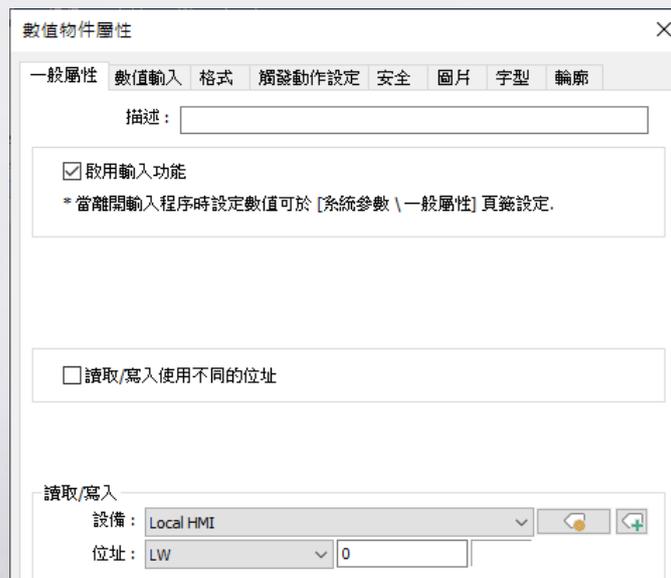
步驟 5. 在 [通道] 分頁選擇啟用 2 個通道。



步驟 6. 建立 2 個多狀態設定物件自動變換數據。



步驟 7. 建立 2 個數值物件地址監看 LW-0、LW-1 取樣數值。



步驟 8. 建立位元狀態切換開關觸發資料取樣。

位元狀態切換開關/位元狀態指示燈物件屬性

一般屬性 安全 圖片 標籤 輪廓

描述: \_\_\_\_\_

位元狀態指示燈  位元狀態切換開關

讀取/寫入使用不同的位址

讀取/寫入

設備: Local HMI

位址: LB 0

輸出反向

屬性

操作模式: 復歸型

步驟 9. 由於前幾個取樣數據在趨勢圖上出現的位置會略有不同，因此透過 [事件登錄] 記錄已取樣的數量以利調整起始畫線位置。

事件登錄

一般屬性 訊息 統計

類別: 0: Category 0 子類別...

等級: 低

HMI 重置時監視事件的延遲時間: 1 秒

推播通知 (EasyAccess 2.0)

類型

位元  字組

讀取

設備: Local HMI

位址: LB 0

通知

啟用

觸發條件

觸發: OFF->ON

事件登錄

一般屬性 訊息 統計

讀取和重置發生次數

啟用

設備: Local HMI

位址: RW 100 16-bit Unsigned

讀取和重置累積時間

啟用

**步驟 10.** 建立 JS 物件放置於趨勢圖上。由於 JS 物件需要在觸發資料取樣時畫取線段，因此需訂閱資料取樣的觸發地址、資料取樣的數據、以及已取樣數量。

trigger：資料取樣的觸發地址

datalog：資料取樣的數據

LogOccurance：已取樣的數量



**步驟 11.** 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 2~3: 新增一個 Canvas 物件並加入到 Widget。
- Line 5~7: 設定繪圖的初始參數。
- Line 9~17: 使用 Subscription 的 onResponse Method 讓 LB-0 在有變化時執行接續的 callback 函式。
- Line 19~80: 繪製對話窗及顯示文字的函式。
- Line 82~88: 為確保步驟間能順序進行而建立的 delay 函數。
- Line 90~106: 繪製右方標示數值的圖案。
- Line 108~118: 繪製線段。
- Line 120~126: 繪製右方欲標示的數值。
- Line 128~131: 清除繪圖。

```
1 var self = this;
2 var ctx = new Canvas();
3 this.widget.add(ctx);
4
5 var color;
6 ctx.fillStyle = "red";
7 ctx.strokeStyle = "red";
8
9 this.config.trigger.onResponse(async (err, data) => {
10
11     if (data.values[0]) {
12
13         clear();
14         myAsyncFunction();
15     }
16 });
17
18 async function myAsyncFunction() {
19
20     var datalog, ch1, ch2;
21     var trigger, triggerCount;
22     var x;
23
24     datalog = driver.getData(self.config.datalog[0], 2, (err, data) => {
25
26         if (err) {
27
28             console.log('Error:', err.message);
29
30         } else {
31
32             ch1 = data.values[0];
33             ch2 = data.values[1];
34
35         }
36     });
37
38     await delay(3);
39
40     trigger = driver.getData(self.config.LogOccurance, 1, (err, data) => {
41
42         if (err) {
43             console.log('Error:', err.message);
44
45         } else {
46             triggerCount = data.values;
47
48         }
49     });
50
51     await delay(3);
52
```

(僅擷取部分指令，詳細請參考工程檔案)

### 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
<b>視窗 10</b>			
多狀態設定物件	LW-0	SW_0	自動變換 LW-0 數據
多狀態設定物件	LW-1	SW_1	自動變換 LW-1 數據
數值物件	LW-0	NE_0	監看 LW-0 數據
數值物件	LW-1	NE_1	監看 LW-1 數據
數值物件	RW-100	NE_2	監看 RW-100 數據(取樣累計次數)
位元切換開關	LB-0	TS_0	觸發資料取樣
JS 物件	LB-0、 LW-0、 LW-1、 RW-100	CO_0	訂閱 LB-0 狀態、 讀取 LW-0、LW-1 數據、 讀取 RW-100 取樣次數
<b>視窗 11</b>			
趨勢圖		TD_0	繪製趨勢圖
<b>背景</b>			
資料取樣	LB-0、 LW-0、 LW-1、		LB-0 觸發資料取樣、 取樣 LW-0、LW-1 資料
事件登錄	LB-0、 RW-100		偵測 LB-0 觸發、 將觸發次數寫入 RW-100