

威綸科技股份有限公司

# JS 物件

顯示 XY 曲線圖物件參數

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 .....	7

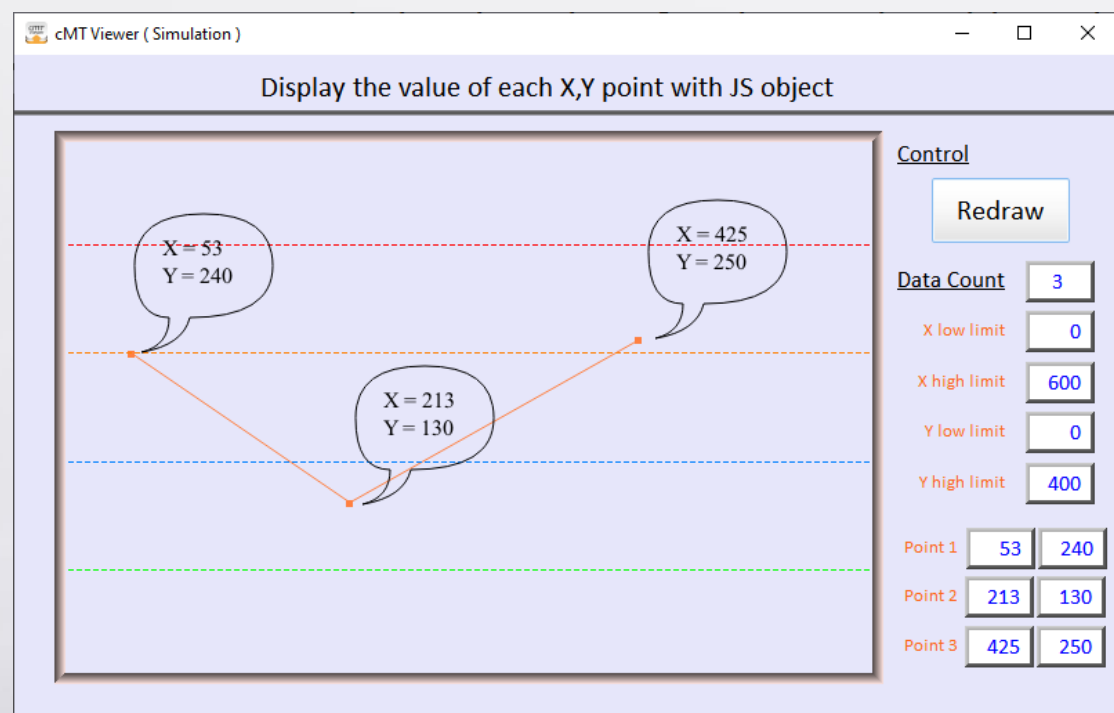
## 1. 簡介與操作

### 簡介

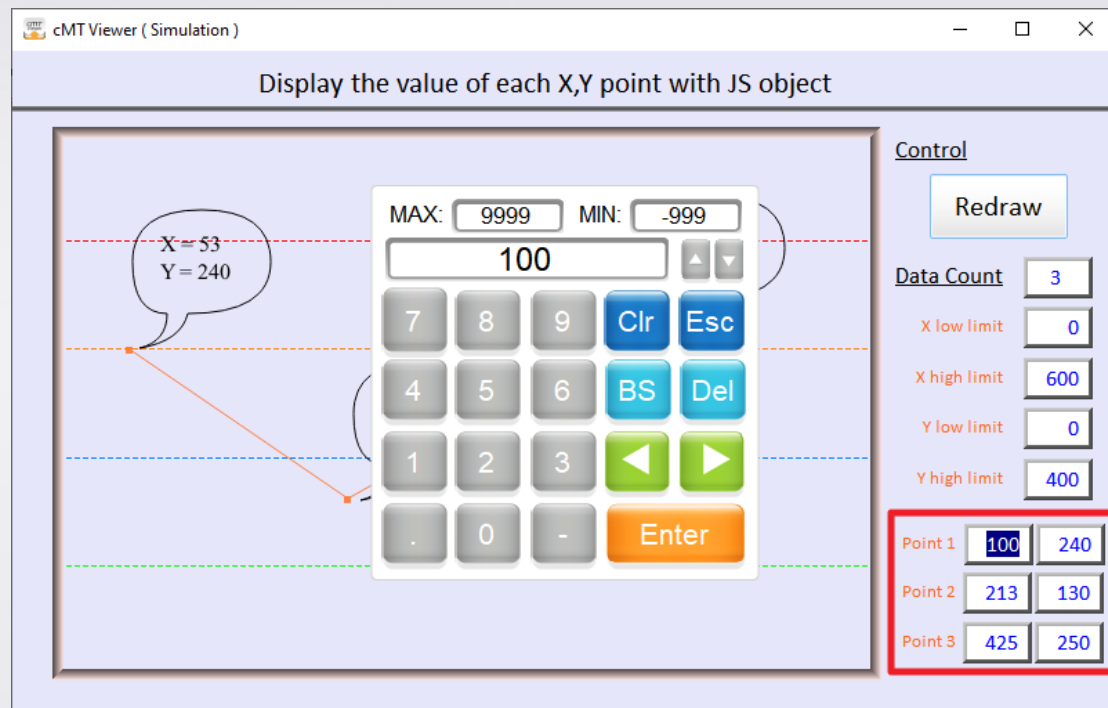
為了方便檢視 XY 圖上點的數據，透過 JS 物件可清楚指出 XY 圖上點的位置。

### 操作

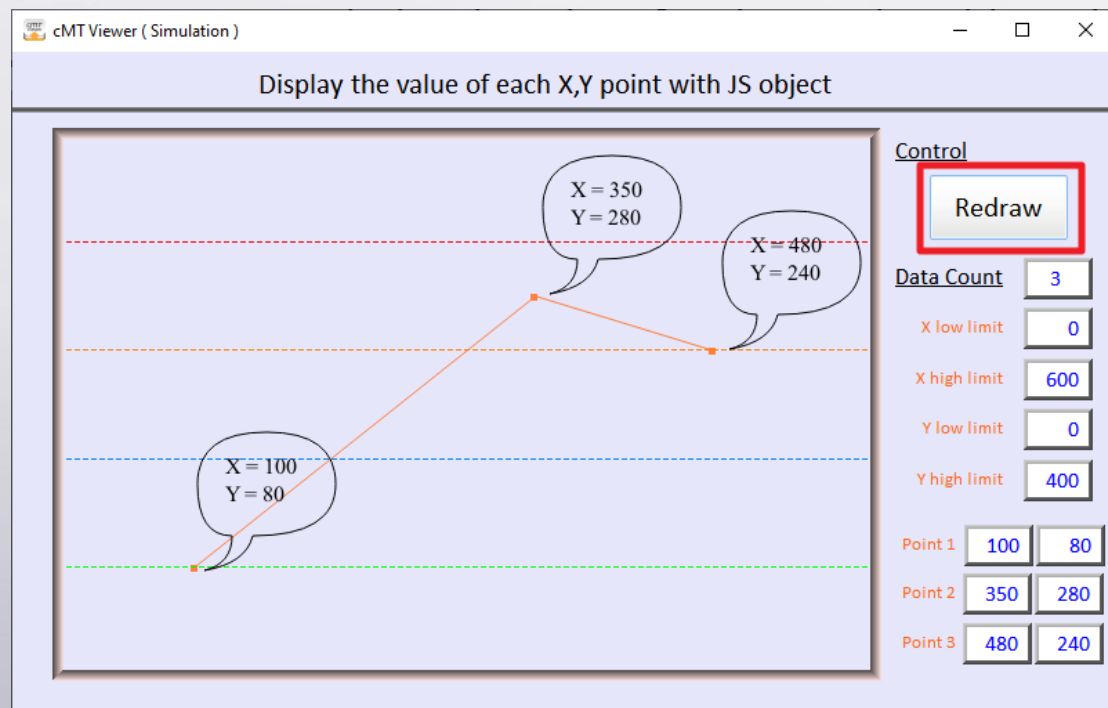
**步驟 1.** 開啟並執行工程檔案，並將工程檔案下載至人機。



步驟 2. 可手動調整 Point 1, 2, 3 的座標。



步驟 3. 觸發 [Redraw] 讓 XY 曲線圖物件與 JS 物件重新繪圖。



## 2. 設定程序

**步驟 1.** 建立一個 XY 曲線圖物件，控制位址為 LW-10，讀取的資料位址來源為 LW-104。



XY 曲線圖物件屬性

一般屬性 顯示區域 安全 圖片 輪廓

描述:

方向: 往右移動 通道數目: 1

控制

設備: Local HMI 位址: LW 10 16-bit Unsigned [使用方式...](#)

通道: 0

讀取

設備: Local HMI 位址: LW 104 16-bit Signed

☐ X 軸數據與 Y 軸數據取自不同位址

範圍上下限 ☐ 上下限值取自暫存器

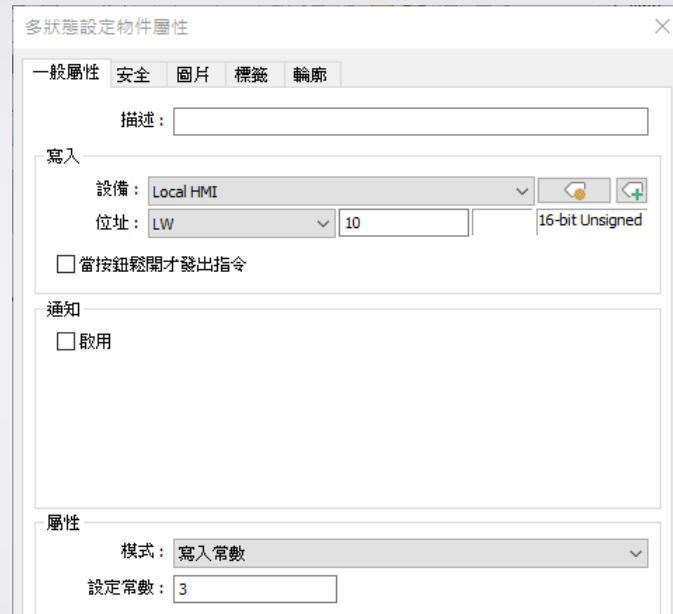
X 軸 下限: 0 上限: 600

Y 軸 下限: 0 上限: 400

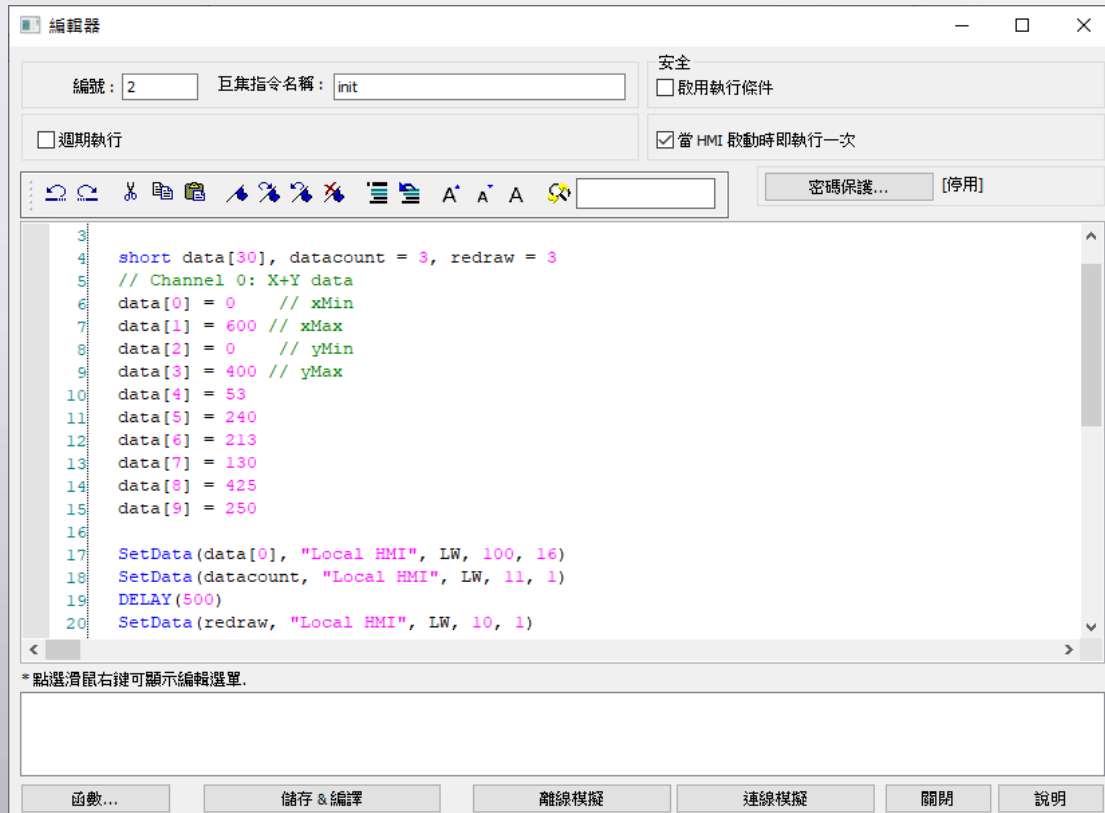
OK Cancel Apply Help

**步驟 2.** 建立數個數值物件，位址設定為 XY 曲線圖物件所用的地址。本範例使用各三個 XY 資料，因此需要六個數值物件。

步驟 3. 建立一個多狀態設定物件，控制 XY 曲線圖物件的重繪功能。當輸入常數 3 時，XY 曲線圖物件會清除舊圖並重新繪製。



步驟 4. 建立巨集，將 XY 曲線圖物件的資料預先輸入。啟用 [當 HMI 啟動時即執行一次]，則此巨集一開機就會執行一次。

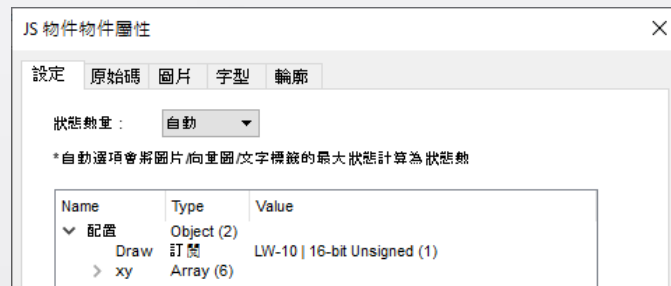


步驟 5. 建立 JS 物件，物件大小、位置與 XY 曲線圖物件相同。

步驟 6. 由於 JS 物件需要與 XY 曲線圖物件同步同時重新繪製，因此需訂閱 XY 曲線圖物件的控制地址。重繪時需要座標資訊，因此需要 XY 曲線圖物件的資料地址。

Draw：訂閱 XY 曲線圖物件的控制地址，LW-10。

xy：XY 曲線圖物件繪製的 XY 資料地址。



步驟 7. 建立 JS 指令。

- 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) (`'Draw'` and `'xy'`)。
- Line 2~3: 新增一個 Canvas 物件並加入到 Widget。
- Line 5~15: 使用 Subscription 的 onResponse Method 讓 LW-10 在有變化時執行接續的 callback 函式。callback 函式中，當訂閱的 LW-10 在有變化時，會主動通知。當 LW-10 的數據為 3 (XY 圖重繪命令) 時，JS 物件會去讀取 XY 物件的 X、Y 資料並重繪。
- Line 18~37: 繪製對話窗及顯示文字的函式。
- Line 39~41: 清除 Canvas 的函式。

```
1 var self = this;
2 var ctx = new Canvas();
3 self.widget.add(ctx);
4
5 self.config.Draw.onResponse(async(err,data)=>{
6     if (data.values[0] == 3){
7         clear();
8         var data = await driver.promises.getData(self.config.xy[0], 6);
9
10        for (let i = 0; i < 3; i++) {
11            let j=i*2;
12            draw(data.values[j],data.values[j+1]);
13        }
14    }
15 });
16
17
18 function draw(newX, newY) {
19
20     ctx.fillStyle = 'black';
21     ctx.beginPath();
22     ctx.moveTo(newX+55, 300-newY);
23     ctx.quadraticCurveTo(newX+5, 300-newY, newX+5, 300-newY+37.5);
24     ctx.quadraticCurveTo(newX+5, 300-newY+75, newX+30, 300-newY+75);
25     ctx.quadraticCurveTo(newX+30, 300-newY+95, newX+10, 300-newY+100);
26     ctx.quadraticCurveTo(newX+40, 300-newY+95, newX+45, 300-newY+75);
27     ctx.quadraticCurveTo(newX+105, 300-newY+75, newX+105, 300-newY+37.5);
28     ctx.quadraticCurveTo(newX+105, 300-newY, newX+55, 300-newY);
29     ctx.stroke();
30
31
32     ctx.fillStyle = 'black';
33     ctx.font = '16px serif';
34     ctx.fillText('X = ' + newX, newX+25, 300-newY+30);
35     ctx.fillText('Y = ' + newY, newX+25, 300-newY+50);
36
37 };
38
39 function clear(){
40     ctx.clearRect(0, 0, ctx.width, ctx.height);
41 }
42
```



### 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			
XY 物件	LW-10、 LW-104	XY_0	控制位址與資料讀取位址。
多狀態設定物件	LW-10	SW_0	XY 軸的控制位址。
數值物件	LW-11	ND_0	顯示 XY 軸物件的資料數量。
數值物件	LW-100	ND_1	顯示 X 下限。
數值物件	LW-101	ND_2	顯示 X 上限。
數值物件	LW-102	ND_3	顯示 Y 下限。
數值物件	LW-103	ND_3	顯示 Y 上限。
數值物件	LW-104 ~ LW-109	NE_10 ~ NE-5	XY 軸的資料位址。
JS 物件	LW-10、 LW-104	CO_0	