



# EasyBuilder Pro

Ver. 6.10.01

# EasyBuilder Pro 使用手冊

1.	關於 EasyBuilder Pro 安裝.....	1-1
1.1.	安裝環境要求 .....	1-2
1.2.	安裝步驟 .....	1-3
1.3.	軟體更新檢查 .....	1-7
2.	Utility Manager .....	2-1
2.1.	概要 .....	2-2
2.2.	設計 .....	2-3
2.3.	分析測試工具 .....	2-5
2.4.	傳輸 .....	2-7
2.4.1.	下載 (PC->HMI) .....	2-8
2.4.2.	上傳 (HMI->PC) .....	2-10
2.4.3.	建立使用在 USB 碟與 SD 卡所需的下載資料.....	2-11
2.4.4.	透過 USB 碟 / SD 卡下載程式到 HMI 的步驟 .....	2-11
2.5.	維護 .....	2-12
2.6.	資料轉換 .....	2-14
3.	建立簡單的工程檔案.....	3-1
3.1.	概要 .....	3-2
3.2.	建立新的工程檔案 .....	3-2
3.3.	儲存和編輯工程檔案 .....	3-4
3.4.	開始編譯和快速編譯 .....	3-5
3.5.	離線模擬和連線模擬 .....	3-6
3.6.	cMT Viewer.....	3-6
3.7.	下載工程檔案至 HMI .....	3-7
3.7.1.	自 EasyBuilder Pro 設定 .....	3-7
3.7.2.	使用 HMI 名稱 .....	3-9
3.7.3.	使用 USB 下載線 .....	3-11
3.7.4.	使用 USB 碟 / SD 卡 .....	3-11
3.8.	從 HMI 上傳工程檔案 .....	3-12
4.	硬體設定.....	4-1
4.1.	概要 .....	4-2
4.2.	I/O 埠 .....	4-2
4.3.	LED 指示燈 .....	4-2
4.4.	系統設置 .....	4-2
4.5.	系統工具列 .....	4-3
4.5.1.	系統設定 .....	4-4
4.5.2.	系統資訊 .....	4-16
5.	系統參數設定.....	5-1
5.1.	概要 .....	5-2

5.2.	設備清單	5-2
5.2.1.	本機 HMI 設定	5-2
5.2.2.	如何控制一台本機 PLC	5-3
5.2.3.	如何控制一台遠端設備	5-8
5.2.4.	如何控制一台遠端 HMI	5-9
5.3.	HMI 屬性	5-11
5.4.	一般屬性	5-15
5.5.	系統設定	5-22
5.6.	遠端	5-26
5.7.	使用者密碼	5-27
5.7.1.	一般模式	5-28
5.7.2.	進階安全模式	5-28
5.8.	擴展記憶體	5-30
5.9.	列印/備份伺服器	5-34
5.10.	時間同步/夏令時間	5-35
5.11.	郵件	5-37
5.12.	行動網路	5-39
5.13.	FTP	5-41
5.13.1.	FTP 伺服器	5-41
5.13.2.	FTP 客戶端	5-41
6.	視窗	6-1
6.1.	概要	6-2
6.2.	視窗類型	6-2
6.2.1.	基本視窗	6-2
6.2.2.	快選視窗	6-2
6.2.3.	公共視窗	6-3
6.2.4.	系統訊息視窗	6-3
6.3.	視窗的建立、設定與刪除	6-4
6.3.1.	視窗的建立與設定	6-5
6.3.2.	視窗的開啟、關閉或刪除	6-7
6.4.	視窗透明度	6-7
6.5.	編輯視窗應用	6-11
6.5.1.	編輯視窗	6-11
6.5.2.	尺規功能	6-12
6.5.3.	快速複製功能	6-13
7.	事件登錄	7-1
7.1.	概要	7-2
7.2.	事件登錄管理	7-2

7.2.1.	eMT、iE、XE、mTV、iP 系列.....	7-2
7.2.2.	cMT、cMT X 系列 .....	7-3
7.2.3.	Excel 編輯 .....	7-8
7.2.4.	快速查閱無效的事件 .....	7-8
7.3.	建立一個新的事件記錄 .....	7-9
8.	資料取樣.....	8-1
8.1.	概要 .....	8-2
8.2.	資料取樣記錄管理 .....	8-2
8.3.	新增一個資料取樣 .....	8-2
8.3.1.	自動停止選項範例 .....	8-9
8.3.2.	自訂檔案管理 .....	8-10
8.4.	外部裝置或資料庫同步 cMT Viewer 資料 .....	8-12
8.5.	查看 cMT Viewer 特定日期或特定檔案的歷史資料 .....	8-12
9.	物件一般屬性.....	9-1
9.1.	概要 .....	9-2
9.2.	選擇 PLC 並設定讀寫位址 .....	9-2
9.3.	使用向量圖庫與圖片庫 .....	9-3
9.3.1.	圖片 .....	9-3
9.3.2.	圖案 .....	9-4
9.3.3.	圖片資訊 .....	9-5
9.4.	設定標籤內容 .....	9-7
9.5.	輪廓調整 .....	9-9
9.6.	重疊物件 (cMT / cMT X 系列支援) .....	9-10
9.6.1.	啟用觸控區域 .....	9-10
9.6.2.	尋找重疊物件 .....	9-10
9.6.3.	轉換至複合式多功能按鈕 .....	9-11
10.	使用者密碼與物件安全防護.....	10-1
10.1.	概要 .....	10-2
10.2.	使用者密碼與可操作物件類別設定 .....	10-2
10.2.1.	一般模式 .....	10-2
10.2.2.	進階安全模式 .....	10-3
10.2.3.	LDAP .....	10-4
10.2.4.	遠端 HMI .....	10-5
10.3.	進階安全模式之控制位址 .....	10-6
10.3.1.	控制位址使用說明 .....	10-6
10.3.2.	命令功能說明 .....	10-7
10.3.3.	結果輸出說明 .....	10-9
10.4.	進階安全模式之應用 .....	10-10

10.4.1. 汇入使用者帳號 .....	10-10
10.4.2. 使用 USB 安全金鑰登入 .....	10-11
10.4.3. 使用 USB 安全金鑰自動登入/登出 .....	10-13
10.4.4. 進階安全模式搭配項目選單物件 .....	10-14
10.4.5. LDAP 模式 .....	10-14
10.4.6. 遠端 HMI 模式 .....	10-21
10.4.7. 使用插件自動登入/登出 .....	10-23
10.5. 物件安全設定頁 .....	10-25
10.5.1. 安全控制 .....	10-25
10.5.2. 開啟/關閉 .....	10-26
10.5.3. 使用者限制 .....	10-28
10.6. 物件安全防護範例 .....	10-28
10.7. 工程檔案之使用者密碼編輯保護 .....	10-31
10.8. 批次修改物件安全屬性 .....	10-32
11. 索引暫存器 .....	11-1
11.1. 概要 .....	11-2
11.2. 使用索引暫存器範例 .....	11-4
11.3. 使用標籤式 PLC 與索引暫存器範例 .....	11-6
12. 鍵盤的設計與使用 .....	12-1
12.1. 概要 .....	12-2
12.2. 設計彈出鍵盤 .....	12-2
12.3. 使用直接視窗的方式設計鍵盤 .....	12-5
12.4. 將鍵盤固定在需要輸入的視窗上 .....	12-6
12.5. 製作 Unicode 鍵盤 .....	12-6
13. 物件 .....	13-1
13.1. 位元狀態指示燈 .....	13-3
13.1.1. 概要 .....	13-3
13.1.2. 設定 .....	13-3
13.2. 多狀態指示燈 .....	13-5
13.2.1. 概要 .....	13-5
13.2.2. 設定 .....	13-5
13.3. 位元狀態設定 .....	13-10
13.3.1. 概要 .....	13-10
13.3.2. 設定 .....	13-10
13.4. 多狀態設定 .....	13-13
13.4.1. 概要 .....	13-13
13.4.2. 設定 .....	13-13
13.5. 功能鍵 .....	13-21

13.5.1. 概要 .....	13-21
13.5.2. 設定 .....	13-21
13.6. 位元狀態切換開關 .....	13-28
13.6.1. 概要 .....	13-28
13.6.2. 設定 .....	13-28
13.7. 多狀態切換開關 .....	13-31
13.7.1. 概要 .....	13-31
13.7.2. 設定 .....	13-31
13.8. 滑動開關 .....	13-34
13.8.1. 概要 .....	13-34
13.8.2. 設定 .....	13-34
13.8.3. 複合物件 .....	13-37
13.9. 數值 .....	13-40
13.9.1. 概要 .....	13-40
13.9.2. 設定 .....	13-40
13.10. 字元 .....	13-59
13.10.1. 概要 .....	13-59
13.10.2. 設定 .....	13-59
13.11. 間接視窗 .....	13-63
13.11.1. 概要 .....	13-63
13.11.2. 設定 .....	13-63
13.12. 直接視窗 .....	13-72
13.12.1. 概要 .....	13-72
13.12.2. 設定 .....	13-72
13.13. 移動/旋轉圖形 .....	13-78
13.13.1. 概要 .....	13-78
13.13.2. 設定 .....	13-78
13.14. 動畫 .....	13-83
13.14.1. 概要 .....	13-83
13.14.2. 設定 .....	13-83
13.15. 棒圖 .....	13-87
13.15.1. 概要 .....	13-87
13.15.2. 設定 .....	13-87
13.16. 錶針 .....	13-94
13.16.1. 概要 .....	13-94
13.16.2. 設定 .....	13-94
13.17. 趨勢圖 .....	13-102
13.17.1. 概要 .....	13-102

13.17.2. 設定 .....	13-102
13.18. 歷史數據顯示 .....	13-124
13.18.1. 概要 .....	13-124
13.18.2. 設定 .....	13-124
13.19. 數據群組顯示 .....	13-133
13.19.1. 概要 .....	13-133
13.19.2. 設定 .....	13-133
13.20. XY 曲線圖 .....	13-141
13.20.1. 概要 .....	13-141
13.20.2. 設定 .....	13-141
13.21. 報警條與報警顯示 .....	13-148
13.21.1. 概要 .....	13-148
13.21.2. 設定 .....	13-148
13.22. 事件顯示 .....	13-157
13.22.1. 概要 .....	13-157
13.22.2. 設定 .....	13-157
13.23. 資料傳輸 .....	13-171
13.23.1. 資料傳輸 (視窗) .....	13-171
13.23.2. 資料傳輸 (背景) 定時式 .....	13-174
13.23.3. 資料傳輸 (背景) 位元觸發 .....	13-177
13.24. 備份 .....	13-179
13.24.1. 概要 .....	13-179
13.24.2. 設定 .....	13-179
13.25. 媒體播放器 .....	13-188
13.25.1. 概要 .....	13-188
13.25.2. 設定 .....	13-188
13.26. BACnet Schedule .....	13-195
13.26.1. 概要 .....	13-195
13.26.2. 設定 .....	13-195
13.27. PLC 控制 .....	13-199
13.27.1. 概要 .....	13-199
13.27.2. 設定 .....	13-199
13.28. 排程 .....	13-206
13.28.1. 概要 .....	13-206
13.28.2. 設定 .....	13-206
13.29. 項目選單 .....	13-218
13.29.1. 概要 .....	13-218
13.29.2. 設定 .....	13-218

13.30. 計時器 .....	13-226
13.30.1. 概要 .....	13-226
13.30.2. 設定 .....	13-226
13.31. 影像輸入 .....	13-230
13.31.1. 概要 .....	13-230
13.31.2. 設定 .....	13-230
13.32. 系統訊息 .....	13-236
13.32.1. 概要 .....	13-236
13.32.2. 設定 .....	13-236
13.33. 配方檢視 .....	13-238
13.33.1. 概要 .....	13-238
13.33.2. 設定 .....	13-238
13.34. 流動塊 .....	13-245
13.34.1. 概要 .....	13-245
13.34.2. 設定 .....	13-245
13.35. 操作記錄 .....	13-249
13.35.1. 操作記錄設定 .....	13-249
13.35.2. 操作記錄檢視 .....	13-254
13.35.3. 操作記錄列印 .....	13-256
13.36. 複合式多功能按鈕 .....	13-263
13.36.1. 概要 .....	13-263
13.36.2. 設定 .....	13-263
13.37. 圓盤曲線圖 .....	13-271
13.37.1. 概要 .....	13-271
13.37.2. 設定 .....	13-271
13.38. 圖片檢視 .....	13-280
13.38.1. 概要 .....	13-280
13.38.2. 設定 .....	13-280
13.39. 檔案瀏覽器 .....	13-283
13.39.1. 概要 .....	13-283
13.39.2. 設定 .....	13-283
13.40. 匯入/匯出 .....	13-286
13.40.1. 概要 .....	13-286
13.40.2. 設定 .....	13-286
13.41. 圓餅圖 .....	13-290
13.41.1. 概要 .....	13-290
13.41.2. 設定 .....	13-290
13.41.3. 複合物件 .....	13-292

13.42. 條碼 .....	13-294
13.42.1. 一維條碼顯示 .....	13-294
13.42.2. 二維條碼顯示 .....	13-295
13.42.3. 推播通知二維條碼顯示 .....	13-297
13.42.4. 條碼掃描器(Android 攝影機) .....	13-300
13.43. 字串表 .....	13-306
13.43.1. 概要 .....	13-306
13.43.2. 設定 .....	13-306
13.44. 資料庫 .....	13-309
13.44.1. 資料庫伺服器 .....	13-309
13.44.2. SQL 查詢 .....	13-315
13.44.3. SQL 查詢檢視 .....	13-322
13.45. 動態刻度 .....	13-324
13.45.1. 概要 .....	13-324
13.45.2. 設定 .....	13-324
13.46. 動態繪圖 .....	13-327
13.46.1. 概要 .....	13-327
13.46.2. 設定 .....	13-327
13.47. PDF 檢視器 .....	13-332
13.47.1. 概要 .....	13-332
13.47.2. 設定 .....	13-332
13.48. 表格 .....	13-334
13.48.1. 概要 .....	13-334
13.48.2. 設定 .....	13-334
13.49. VNC Viewer .....	13-336
13.49.1. 概要 .....	13-336
13.49.2. 設定 .....	13-336
13.50. 連絡人編輯器 .....	13-340
13.50.1. 設定 .....	13-340
13.51. 事件甘特圖 .....	13-343
13.51.1. 概要 .....	13-343
13.51.2. 設定 .....	13-343
13.51.3. 事件甘特圖 .....	13-345
13.52. 動作觸發 .....	13-349
13.52.1. 概要 .....	13-349
13.52.2. 設定 .....	13-349
13.53. 月曆 .....	13-355
13.53.1. 概要 .....	13-355

13.53.2. 設定 .....	13-355
13.54. 觸控手勢 .....	13-357
13.54.1. 概要 .....	13-357
13.54.2. 設定 .....	13-357
13.55. PLC 網頁瀏覽器 .....	13-360
13.55.1. 概要 .....	13-360
13.55.2. 設定 .....	13-360
13.55.3. 注意事項 .....	13-362
13.56. 觸發條件 .....	13-364
13.56.1. 概要 .....	13-364
13.56.2. 設定 .....	13-364
13.57. FTP 連線 .....	13-367
13.57.1. 概要 .....	13-367
13.57.2. 設定 .....	13-367
13.58. 日期/時間 .....	13-370
13.58.1. 概要 .....	13-370
13.58.2. 設定 .....	13-370
13.59. 模板 .....	13-372
13.59.1. 模板視窗 .....	13-372
13.59.2. 模板 .....	13-373
14. 向量圖庫與圖片庫的建立 .....	14-1
14.1. 概要 .....	14-2
14.2. 向量圖庫的建立 .....	14-2
14.2.1. 向量圖管理 .....	14-2
14.2.2. 建立向量圖庫的步驟 .....	14-4
14.3. 圖片庫的建立 .....	14-9
14.3.1. 圖片管理 .....	14-9
14.3.2. 建立圖片庫的步驟 .....	14-11
14.3.3. 從剪貼簿匯入圖片 .....	14-13
14.4. 向量圖庫 / 圖片庫批次管理 .....	14-15
14.4.1. 圖片切換 .....	14-16
14.4.2. 顏色切換 .....	14-16
14.4.3. 特殊功能 .....	14-17
14.4.4. 視窗調整 .....	14-18
15. 文字標籤庫與多國語言使用 .....	15-1
15.1. 概要 .....	15-2
15.2. 文字標籤庫管理 .....	15-2
15.3. 文字標籤庫的建立 .....	15-3

15.4. 文字標籤庫的使用 .....	15-4
15.5. 語言與字型選擇 .....	15-5
15.5.1. 語言 .....	15-5
15.5.2. 預設字型 .....	15-6
15.5.3. 字型管理 .....	15-8
15.5.4. 文字對應 .....	15-8
15.5.5. 完整字型檔案下載 .....	15-9
15.6. 多國語言的使用 .....	15-10
15.7. 快速新增到文字標籤庫 .....	15-13
15.7.1. 物件標籤 .....	15-13
15.7.2. 非物件文字 .....	15-16
16. 位址標籤庫的建立與使用 .....	16-1
16.1. 概要 .....	16-2
16.2. 位址標籤庫的建立 .....	16-2
16.3. 位址標籤庫的使用 .....	16-5
17. 配方資料傳送 .....	17-1
17.1. 概要 .....	17-2
17.2. 使用乙太網路或 USB 線更新配方資料 .....	17-2
17.3. 使用 SD 卡或 USB 碟更新配方資料 .....	17-3
17.4. 配方資料傳輸 .....	17-4
17.5. 配方資料儲存機制 .....	17-5
18. 巨集指令說明 .....	18-1
18.1. 概要 .....	18-2
18.2. 巨集指令編輯器功能使用說明 .....	18-2
18.3. 巨集指令的結構 .....	18-8
18.4. 巨集指令的語法 .....	18-9
18.4.1. 常數和變數 .....	18-9
18.4.2. 運算符號 .....	18-11
18.5. 語句 .....	18-13
18.5.1. 定義語句 .....	18-13
18.5.2. 賦值語句 .....	18-14
18.5.3. 邏輯運算語句 .....	18-14
18.5.4. 多重判斷語句 .....	18-16
18.5.5. 迴圈語句 .....	18-17
18.6. 子函數 .....	18-19
18.7. 內置函數功能 .....	18-21
18.7.1. 函數一覽表 .....	18-21
18.7.2. 設備 .....	18-27

18.7.3.	Free Protocol.....	18-35
18.7.4.	巨集控制 .....	18-40
18.7.5.	資料操作 .....	18-41
18.7.6.	資料轉換 .....	18-45
18.7.7.	字串處理 .....	18-54
18.7.8.	數學運算 .....	18-74
18.7.9.	統計 .....	18-80
18.7.10.	配方資料庫 .....	18-81
18.7.11.	資料取樣/事件記錄.....	18-85
18.7.12.	校驗 .....	18-89
18.7.13.	其他 .....	18-92
18.8.	怎樣建立和執行巨集指令 .....	18-95
18.8.1.	怎樣建立一個巨集指令 .....	18-95
18.8.2.	執行巨集指令 .....	18-99
18.9.	使用者自定義函數功能 .....	18-99
18.9.1.	匯入函數庫檔案 .....	18-100
18.9.2.	如何使用巨集函數庫 .....	18-101
18.9.3.	函數庫管理介面 .....	18-103
18.10.	使用巨集指令時的注意事項 .....	18-110
18.11.	使用自由協定去控制一個設備 .....	18-111
18.12.	編譯錯誤提示資訊 .....	18-116
18.13.	巨集指令範例程式 .....	18-127
18.14.	巨集指令 TRACE 函數.....	18-132
18.15.	字串處理函式使用方法 .....	18-136
18.16.	巨集指令密碼保護 .....	18-142
18.17.	巨集支援使用變數讀寫 CANbus 地址 .....	18-144
19.	如何將 HMI 設定成 MODBUS 裝置.....	19-1
19.1.	概要 .....	19-2
19.2.	建立一個 MODBUS Server 裝置.....	19-2
19.3.	讀寫一個 MODBUS Server 裝置.....	19-4
19.4.	線上更改 MODBUS Server 站號 .....	19-7
19.5.	關於 MODBUS 各位址的說明 .....	19-7
20.	如何使用條碼掃描器.....	20-1
20.1.	概要 .....	20-2
20.2.	連接條碼掃描器的步驟 .....	20-2
20.3.	CDC/POS 模式 .....	20-6
21.	乙太網路通訊與多台人機連線.....	21-1
21.1.	概要 .....	21-2

21.2. HMI 與 HMI 間的通訊.....	21-2
21.3. PC 與 HMI 間的通訊.....	21-3
21.4. 控制連接在其他 HMI 上的 PLC .....	21-4
21.4.1. eMT / iE / XE / mTV / iP 系列的設定方法 .....	21-5
21.4.2. cMT / cMT X 系列的設定方法.....	21-5
22. 位址暫存器.....	22-1
22.1. 概要 .....	22-3
22.2. 本機 HMI 記憶體位址範圍 .....	22-4
22.2.1. 位元位址 .....	22-4
22.2.2. 字組位址 .....	22-4
22.3. 系統暫存器 .....	22-5
22.3.1. HMI 時間 .....	22-5
22.3.2. HMI 操作 .....	22-6
22.3.3. 觸碰位置 .....	22-7
22.3.4. 本機 HMI 網路資訊 .....	22-8
22.3.5. 工程檔案資訊 .....	22-10
22.3.6. 儲存空間管理 .....	22-11
22.3.7. 配方及擴展記憶體 .....	22-12
22.3.8. 資料取樣 .....	22-15
22.3.9. 事件記錄 .....	22-16
22.3.10. 站號變數 .....	22-19
22.3.11. 索引暫存器 .....	22-20
22.3.12. MODBUS Server 通訊.....	22-22
22.3.13. 通訊參數設定 .....	22-23
22.3.14. 與設備 (COM) 的通訊狀態與控制.....	22-26
22.3.15. 與設備 (乙太網路) 的通訊狀態與控制 .....	22-28
22.3.16. 與設備 (USB) 的通訊狀態與控制 .....	22-38
22.3.17. 與設備 (CAN Bus) 的通訊狀態與控制 .....	22-38
22.3.18. 與遠端 HMI 的通訊狀態與控制 .....	22-38
22.3.19. 與遠端設備的通訊狀態與控制 .....	22-52
22.3.20. 本機/遠端操作限制.....	22-54
22.3.21. 通訊錯誤碼 .....	22-54
22.3.22. 驅動程式 ID.....	22-56
22.3.23. DLT645 控制器 .....	22-56
22.3.24. [Device No Response] 視窗控制.....	22-56
22.3.25. [快選] 視窗控制.....	22-57
22.3.26. EasyAccess .....	22-57
22.3.27. EasyAccess 2.0 .....	22-58

22.3.28. 遠端列印/備份伺服器.....	22-59
22.3.29. 穿透通訊設定 .....	22-59
22.3.30. VNC 控制 .....	22-60
22.3.31. HMI 和工程檔案識別碼 .....	22-61
22.3.32. USB 安全金鑰.....	22-62
22.3.33. 使用者名稱和密碼 .....	22-62
22.3.34. 巨集 .....	22-63
22.3.35. 輸入物件功能 .....	22-65
22.3.36. 時間同步/夏令時間.....	22-65
22.3.37. 行動網路 .....	22-67
22.3.38. WiFi 設定.....	22-68
22.3.39. OPC UA 伺服器.....	22-70
22.3.40. 郵件 .....	22-70
22.3.41. cMT Viewer 與診斷器.....	22-72
22.3.42. CODESYS 資訊.....	22-74
22.3.43. 其它功能 .....	22-74
23. HMI 支援的印表機類型.....	23-1
23.1. 支援的印表機類型 .....	23-2
23.2. 如何新增一台印表機設備並觸發列印 .....	23-5
23.3. cMT / cMT X 安裝網路印表機.....	23-7
23.4. cMT / cMT X 透過安裝 PPD 使用印表機 .....	23-9
23.5. cMT X 系列透過 IPP 協定使用印表機 .....	23-11
23.6. 確認印表機是否支援 CUPS.....	23-13
24. Recipe Editor .....	24-1
24.1. 概要 .....	24-2
24.2. 配方資料 / 擴展記憶體編輯器設定 .....	24-2
24.3. 配方資料庫記錄的設定 .....	24-4
24.3.1. 定義 .....	24-4
24.3.2. 傳送 .....	24-6
24.3.3. 資料 .....	24-7
24.3.4. 系統暫存器 .....	24-9
25. EasyConverter .....	25-1
25.1. 概要 .....	25-2
25.2. 將資料取樣記錄檔案輸出至 Excel .....	25-2
25.3. 將事件記錄檔案輸出至 Excel .....	25-4
25.4. 將操作記錄檔案輸出至 Excel .....	25-6
25.5. 多檔案轉換 .....	25-8
25.6. 比例轉換功能 .....	25-9

25.7. 批次檔規則 .....	25-10
25.8. 檢查歷史記錄檔案完整性 .....	25-13
26. EasyPrinter .....	26-1
26.1. 概要 .....	26-2
26.2. 使用 EasyPrinter 為列印伺服器 .....	26-3
26.2.1. EasyPrinter 設定程序 .....	26-3
26.2.2. EasyBuilder Pro 設定程序 .....	26-4
26.3. 使用 EasyPrinter 為備份伺服器 .....	26-6
26.3.1. EasyPrinter 設定程序 .....	26-6
26.3.2. EasyBuilder Pro 設定程序 .....	26-7
26.4. EasyPrinter 操作說明 .....	26-10
26.4.1. 視窗介面 .....	26-10
26.4.2. 選單項目 .....	26-11
26.5. 轉換批次檔 .....	26-14
26.5.1. 轉換批次檔預設值 .....	26-16
26.5.2. 特定標準 .....	26-16
26.5.3. 轉換批次檔格式 .....	26-17
26.5.4. 執行順序 .....	26-17
27. EasySimulator/cMT Viewer 模擬 .....	27-1
27.1. EasySimulator .....	27-2
27.1.1. 概要 .....	27-2
27.1.2. 設定 EasySimulator 的步驟 .....	27-2
27.2. cMT Viewer .....	27-3
27.2.1. 概要 .....	27-3
27.2.2. 設定 cMT Viewer 的步驟 .....	27-3
28. EasySystemSetting .....	28-1
28.1. EasySystemSetting .....	28-2
28.2. 操作流程範例 .....	28-5
29. 穿透通訊功能 .....	29-1
29.1. 概要 .....	29-2
29.2. 序列埠穿透的乙太網路模式 .....	29-2
29.2.1. 安裝虛擬序列埠驅動程式的步驟 .....	29-2
29.2.2. 更改虛擬序列埠的步驟 .....	29-3
29.2.3. 移除虛擬序列埠的步驟 .....	29-3
29.2.4. 更新虛擬序列埠驅動程式的步驟 .....	29-4
29.2.5. 乙太網路模式設定 .....	29-5
29.3. 序列埠穿透的序列埠模式 .....	29-7
29.3.1. 序列埠設定 .....	29-7

29.3.2. 使用 Utility Manager .....	29-7
29.3.3. 使用系統暫存器 .....	29-9
29.4. 穿透通訊控制 .....	29-9
29.5. SIEMENS S7-200 PPI 與 S7-300 MPI 穿透功能設定 .....	29-10
29.5.1. EasyBuilder Pro 設定 .....	29-10
29.5.2. S7-200 PPI 連線方式 .....	29-10
29.5.3. S7-300 MPI 連線方式 .....	29-11
29.5.4. SIEMENS 穿透相關暫存器 .....	29-13
29.6. 乙太網路穿透 .....	29-15
30. 工程檔案保護功能 .....	30-1
30.1. 概要 .....	30-2
30.2. EXOB/CXOB 密碼 .....	30-2
30.3. 禁止反編譯 .....	30-3
30.4. 取消 HMI 上傳功能 .....	30-3
30.5. 工程檔案識別碼 .....	30-3
30.6. MTP 檔案密碼 .....	30-4
31. cMT Diagnoser .....	31-1
31.1. 概要 .....	31-2
31.2. 如何啟用 cMT 診斷器 .....	31-2
31.3. cMT 診斷器使用者介面 .....	31-3
31.3.1. 物件設定 .....	31-3
31.3.2. 設備設定 .....	31-6
31.3.3. 封包設定 .....	31-7
31.3.4. 巨集設定 .....	31-9
31.3.5. MQTT 設定 .....	31-15
31.3.6. OPC UA 設定 .....	31-18
31.3.7. JS 設定 .....	31-18
31.4. 補充說明 .....	31-19
32. FTP 啟用之運用 .....	32-1
32.1. 概要 .....	32-2
32.2. 登入 FTP 啟用的步驟 .....	32-2
32.3. 備份歷史資料及更新配方資料 .....	32-3
33. EasyDiagnoser .....	33-1
33.1. 概要 .....	33-2
33.2. 設定步驟 .....	33-2
33.3. EasyDiagnoser 設定 .....	33-3
33.3.1. 主要選單 .....	33-3
33.3.2. 通訊記錄區 .....	33-4

33.3.3. 輪詢封包 .....	33-6
33.3.4. 設備 .....	33-8
33.3.5. 輸出 (Macro debug) .....	33-8
33.4. 錯誤代碼 .....	33-9
33.5. 視窗調整 .....	33-10
34. 控制權(cMT/cMT X 系列) .....	34-1
34.1. 概要 .....	34-2
34.2. 設定 .....	34-2
34.3. 範例講解 .....	34-4
35. EasyWatch .....	35-1
35.1. 概要 .....	35-2
35.2. 設定 .....	35-3
35.2.1. 基本功能 .....	35-3
35.2.2. 快速工具 .....	35-3
35.3. 監視物件設定 .....	35-4
35.3.1. 新增監視物件 .....	35-4
35.3.2. 監視物件設定 .....	35-5
35.3.3. 新增監視物件的步驟 .....	35-6
35.4. 巨集物件設定 .....	35-9
35.4.1. 新增巨集物件 .....	35-9
35.4.2. 巨集物件設定 .....	35-9
35.4.3. 新增巨集設定 .....	35-10
35.5. HMI 設定 .....	35-11
35.5.1. 開啟 HMI 設定 .....	35-11
35.5.2. HMI 管理器 .....	35-11
35.6. 物件顯示列表 .....	35-12
35.6.1. 物件顯示欄位 .....	35-12
35.6.2. 頁面設定 .....	35-12
36. 管理員工具 .....	36-1
36.1. 概要 .....	36-2
36.2. 使用者帳號 .....	36-2
36.2.1. 使用者帳號介紹 .....	36-2
36.2.2. 使用者帳號設定 .....	36-4
36.2.3. 使用 EasyBuilder Pro 匯入帳戶 .....	36-5
36.2.4. 使用 EasyWeb 2.0 匯入帳戶到 cMT/cMT X 人機 .....	36-6
36.3. USB 安全金鑰 .....	36-8
36.3.1. USB 安全金鑰介紹 .....	36-8
36.3.2. USB 安全金鑰設定 .....	36-8

36.3.3. 使用 EasyBuilder Pro 設定 USB 安全金鑰.....	36-9
36.4. e-Mail SMTP 伺服器.....	36-10
36.4.1. e-Mail SMTP 伺服器設定.....	36-11
36.5. e-Mail 聯絡人.....	36-12
36.5.1. e-Mail 聯絡人介紹.....	36-12
36.5.2. e-Mail 聯絡人設定.....	36-14
36.5.3. 使用 EasyBuilder Pro 匯入 e-Mail 設定和連絡人.....	36-15
37. MODBUS 閘道功能.....	37-1
37.1. 概要 .....	37-2
37.2. 如何建立一個位址對應表 .....	37-2
37.3. 位址對應設定須知 .....	37-5
38. EasyDownload .....	38-1
38.1. 概要 .....	38-2
38.2. 設定 .....	38-2
39. 資料保護 .....	39-1
39.1. 概要 .....	39-2
39.2. 設定 .....	39-2
39.2.1. 字組位址屬性設定 .....	39-3
39.2.2. 位元位址屬性設定 .....	39-5
40. 影像串流 .....	40-1
40.1. 概要 .....	40-2
40.2. 設定 .....	40-2
40.3. 操作方式 .....	40-2
40.4. 支援機型 .....	40-3
41. 能源管理 .....	41-1
41.1. 能源需量設定 .....	41-2
41.1.1. 概要 .....	41-2
41.1.2. 設定 .....	41-2
41.2. 能源需量顯示設定 .....	41-6
41.2.1. 概要 .....	41-6
41.2.2. 設定 .....	41-6
42. IIoT .....	42-1
42.1. MQTT .....	42-2
42.1.1. 概要 .....	42-2
42.1.2. 設定 .....	42-2
42.2. OPC UA 伺服器 .....	42-26
42.2.1. 概要 .....	42-26
42.2.2. 設定 .....	42-26

42.2.3. 裝置統計資料 .....	42-37
42.2.4. 支援及限制 .....	42-38
43. JS 物件和 JS 庫 .....	43-1
43.1. 概要 .....	43-2
43.1.1. 使用建議 .....	43-2
43.1.2. 環境規格 .....	43-2
43.1.3. 警告 .....	43-2
43.1.4. JS 物件 SDK .....	43-2
43.2. JS 物件 .....	43-3
43.3. JS Resource .....	43-6
43.4. 範例講解 .....	43-7
43.4.1. 位元切換開關 .....	43-7
43.4.2. JS Resource 使用範例 .....	43-15
43.5. 規格限制 .....	43-18
43.6. JS 物件的運作流程 .....	43-18
附錄 A. 各系列 HMI 軟體功能差異 .....	A-1
附錄 B. cMT Viewer 使用限制 .....	B-1
附錄 C. PC 模擬使用限制 .....	C-1
附錄 D. 64-bit 資料型態使用注意事項 .....	D-1

Windows 是美國 Microsoft 公司在美國以及其他國家的商標或註冊商標。

Android 是 Google, Inc. 在美國以及其他國家的註冊商標或商標。

IOS 是 Cisco 公司在美國以及某些其他國家的註冊商標。

本文中出現的其他公司名、產品名或商標均為各公司的商標或註冊商標。

本文件中的資訊可能隨時變更，本公司將不另行通知。

# 1. 關於 EasyBuilder Pro 安裝

本章節說明如何安裝 EasyBuilder Pro。

1.1.	安裝環境要求 .....	1-2
1.2.	安裝步驟 .....	1-3
1.3.	軟體更新檢查 .....	1-7

## 1.1. 安裝環境要求

### 軟體來源:

進入威綸公司網站 <http://www.weintek.com> 下載所有可用軟體語言版本 (包括簡體中文、繁體中文、英文、日文、德文、義大利文、韓文、西班牙文、俄羅斯文、法文、波蘭文及土耳其文版本) 及最新軟體更新資訊。

### 作業系統:

Windows® 7 SP1 (32bit / 64bit)

Windows® 8 (32bit / 64bit)

Windows® 8.1 (32bit / 64bit)

Windows® 10 (32bit / 64bit)

Windows® 11 (32bit / 64bit)

## 1.2. 安裝步驟

- 當由官網下載 EasyBuilder Pro 安裝檔案後，點選 [Install]，選擇所需語言版本後，點選 [下一步]。

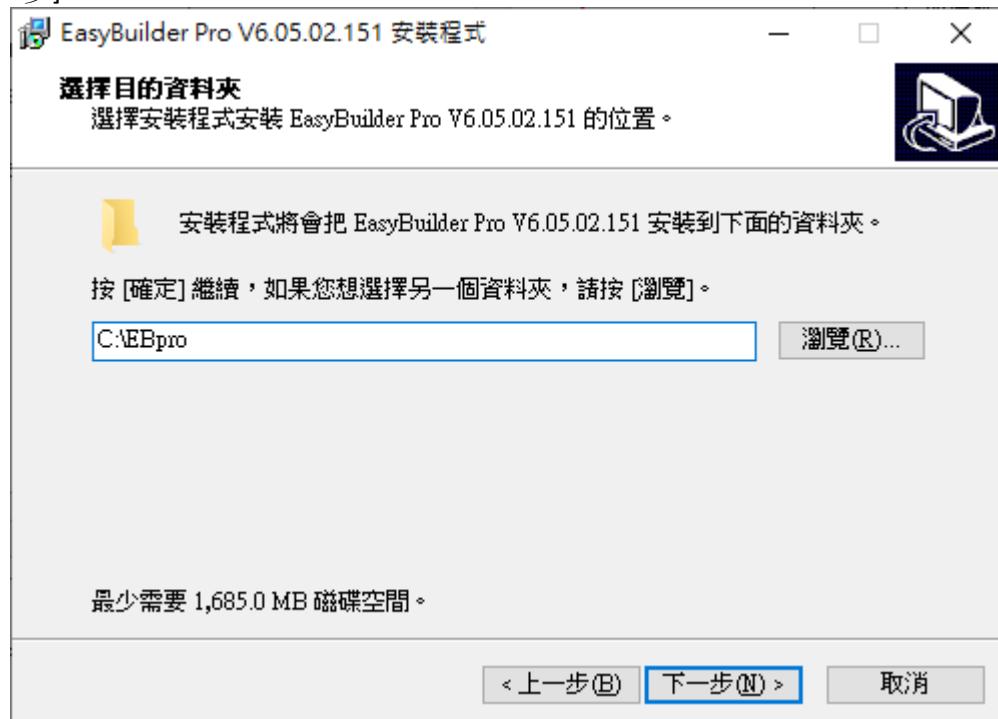


- 系統將會詢問使用者是否想要移除電腦中已安裝過的 EasyBuilder Pro 版本，請根據需求選擇後，點選 [下一步]。
- 請閱讀授權合約，同意後將繼續安裝。

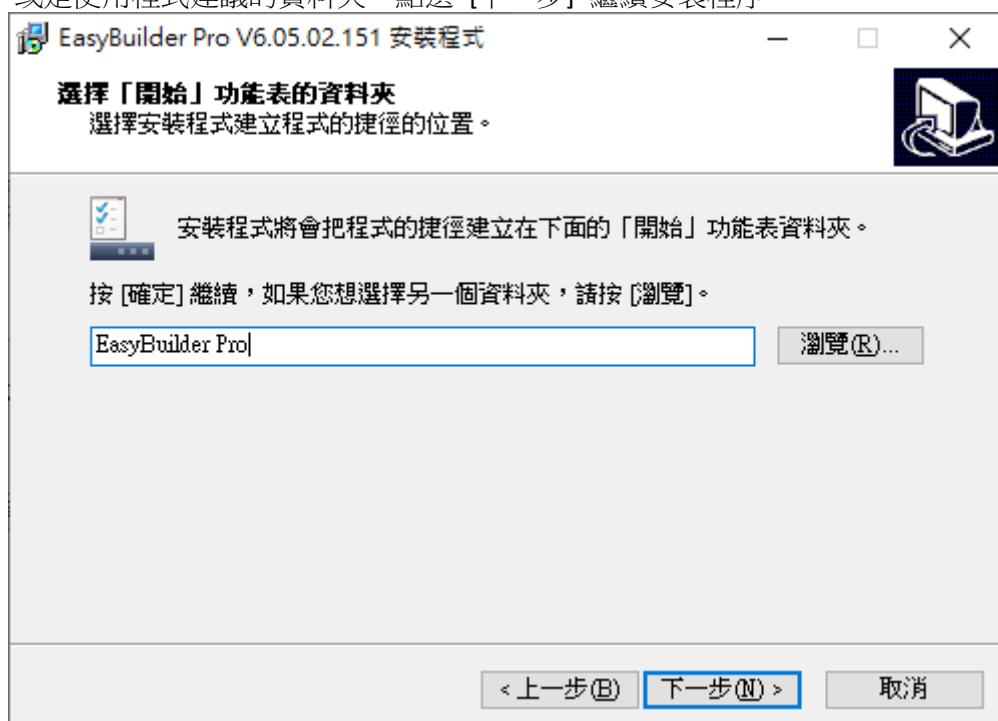


## 關於 EasyBuilder Pro 安裝

4. 指定一個全新的資料夾給 EasyBuilder Pro 安裝資料，或是直接使用系統建議的資料夾，點選 [下一步]。

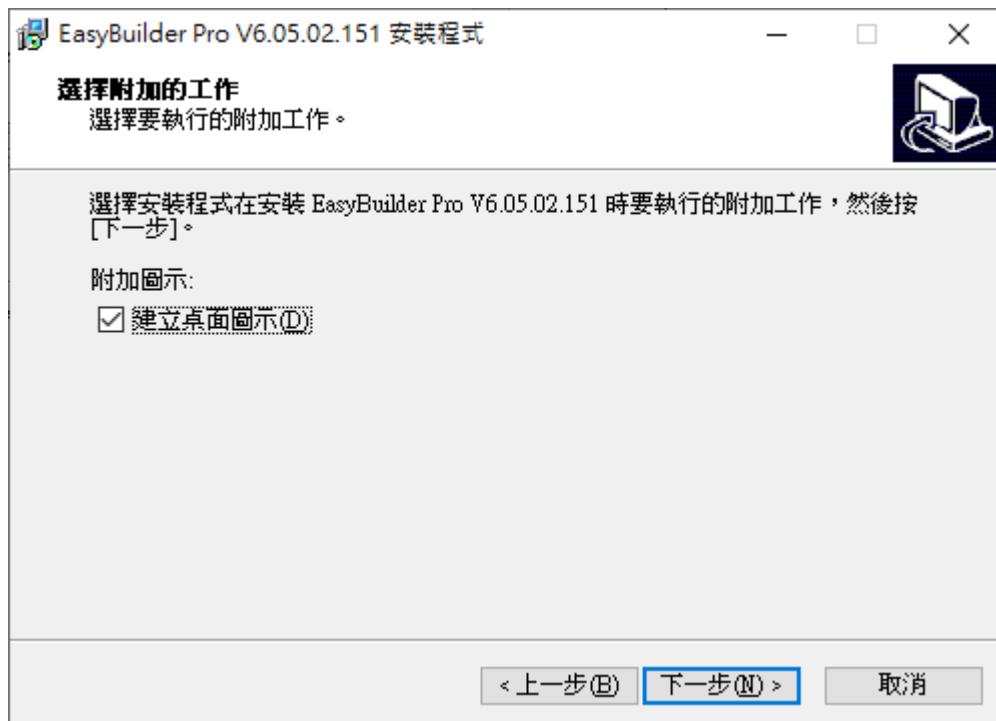


5. 系統會請使用者指定一個開始功能表資料夾用來建立程式捷徑，點選 [瀏覽] 指定一個資料夾，或是使用程式建議的資料夾，點選 [下一步] 繼續安裝程序。

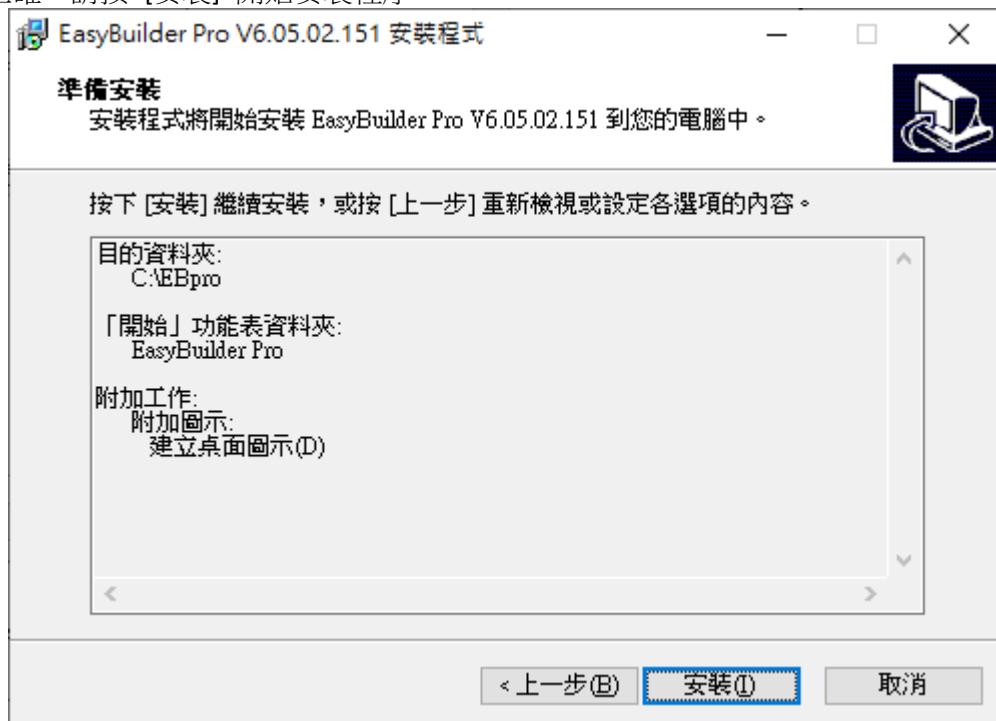


6. 系統會詢問使用者是否要執行附加工作，例如 [建立桌面圖示]，若需要請勾選，然後按 [下一步] 繼續安裝程序。

## 關於 EasyBuilder Pro 安裝

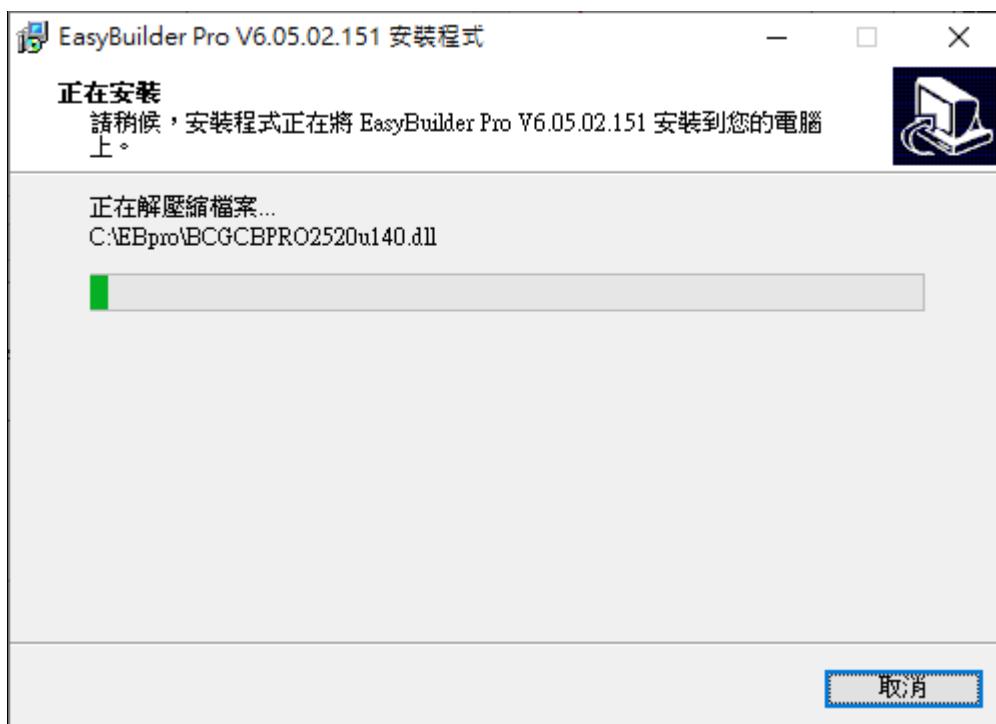


7. 在此階段所有設定已完成，請檢查是否無誤，若有需要改選的部份，請按 [上一步]，若全部正確，請按 [安裝] 開始安裝程序。

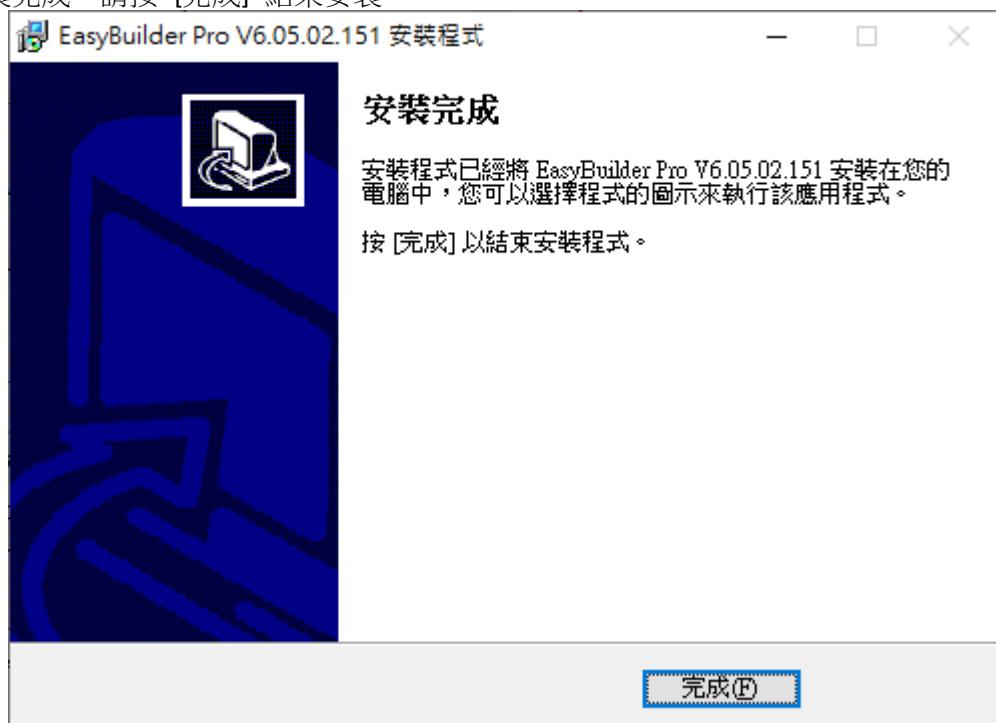


8. 安裝程序執行中。

## 關於 EasyBuilder Pro 安裝



9. 安裝完成，請按 [完成] 結束安裝。



10. 在 [開始] » [所有程式] » [EasyBuilder Pro] 目錄下可看到 EasyBuilder 各功能捷徑。

軟體目錄下各選項的涵義如下：

項目	描述
<b>Administrator</b>	可將 [使用者帳號]、[USB 安全金鑰]、[e-Mail SMTP 伺服器設定] 及 [e-Mail 聯絡人] 四種資料儲存於外部裝置 (USB 碟或 SD 卡)，並匯入 HMI，大幅增加資料的可攜性與便利性。
<b>cMT Viewer</b>	透過網路連接至 cMT / cMT X 系列，並可於 PC 上顯示及操作 cMT / cMT X 系列的 HMI。
<b>EasyAccess</b>	HMI 可藉由 Internet 登入 EasyAccess 伺服器，並實現遠端操作 HMI。
<b>EasyBuilder Pro</b>	EasyBuilder Pro 程式編輯軟體。
<b>EasyConverter</b>	資料取樣與事件記錄轉檔工具。
<b>EasyDiagnoser</b>	線上監控程式通訊除錯工具。
<b>EasyPrinter</b>	允許 HMI 透過乙太網路，將資料輸出至遠端 PC 的伺服器，實現遠端列印/備份的功能。
<b>EasySimulator</b>	執行程式模擬。
<b>EasySystemSetting</b>	提供使用者利用 USB 碟或 SD 卡更新硬體設定資訊的功能。
<b>EasyWatch</b>	透過 PC 監看或設定 HMI 與 PLC 位址內的數值，同時也可以進行巨集的呼叫，更方便使用者進行除錯與遠端監控使用。
<b>Recipe Editor</b>	可以設定配方數據格式，開啟配方數據檔案及外部記憶體檔案。
<b>Release Note</b>	軟體版本及相關最新資訊說明。
<b>UtilityManagerEx</b>	EasyBuilder Pro 綜合管理軟體。

### Note

- 部分 HMI 支援使用 USB 線下載/上傳程式，完成安裝 EasyBuilder Pro 後會自動安裝 USB 驅動程式，若彈出“Windows 無法驗證驅動程式的發行者”屬正常現象，請選擇繼續安裝，結束後可至 [電腦管理] » [裝置管理員] 確認 USB 驅動是否安裝完成。

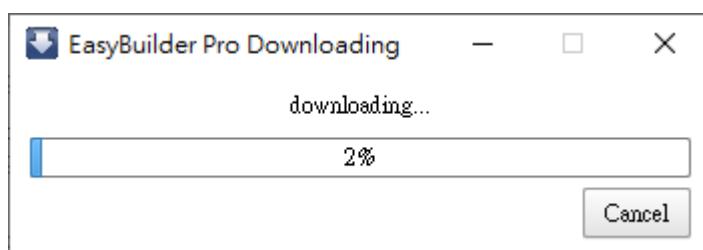
## 1.3. 軟體更新檢查

開啟 EasyBuilder Pro 後，系統會檢查電腦是否已經安裝最新版的 EasyBuilder Pro，如有版本提供更新，可以選擇是否下載最新版本或稍後提醒。

## 關於 EasyBuilder Pro 安裝



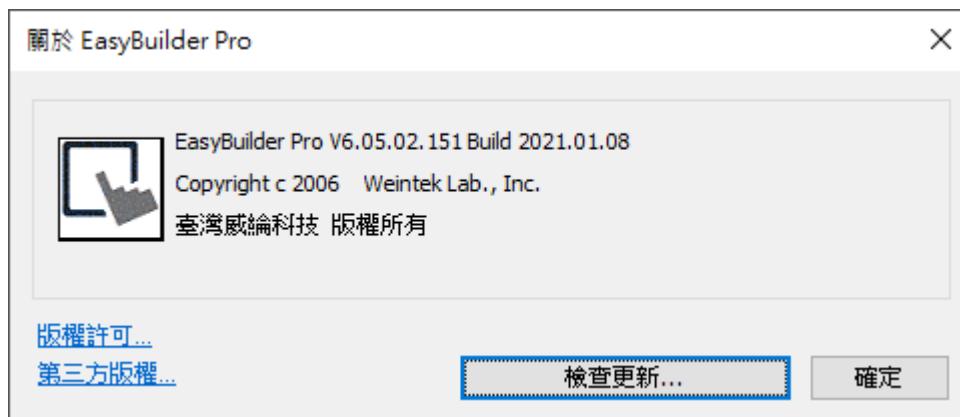
點選下載後，將自動下載新版本 EasyBuilder Pro 至暫存資料夾，過程中顯示更新狀態與進度條，下載完成後，請參考 1.2.安裝步驟完成安裝。



### Note

- 可於[檔案]/[偏好設定]/[其他]選擇是否當新版本 EasyBuilder Pro 推出時自動通知。

使用者也可以透過檔案/關於查看 EasyBuilder 版本並檢查更新。



## 2. Utility Manager

本章節說明如何使用 Utility Manager。

2.1.	概要 .....	2-2
2.2.	設計 .....	2-3
2.3.	分析測試工具 .....	2-5
2.4.	傳輸 .....	2-7
2.5.	維護 .....	2-12
2.6.	資料轉換 .....	2-14

## 2.1. 概要

在 EasyBuilder Pro 軟體安裝完成後，雙擊 PC 桌面上的 UtilityManagerEx 捷徑即可開啟。這是 EasyBuilder Pro 軟體的綜合管理器，可當成獨立的程式來操作。請選擇您所使用的機型，請注意，若未選擇正確，部分功能可能無法操作或正確運作。



設定	描述
	將 Utility Manager 程式對話窗最小化。
	關閉 Utility Manager。
	將常用的工具加入設定對話窗下方的工具列。
<b>Run</b>	執行在工具列中選擇的程式。
<b>Edit</b>	刪除在工具列中選擇的程式。

## 2.2. 設計



設定	描述
<b>EasyBuilder Pro</b>	啟動 EasyBuilder Pro 程式編輯器。
<b>位址瀏覽程式</b>	檢視各個 PLC 的設備類型與位址範圍。
<b>模擬</b>	在 PC 上模擬工程檔案的運行。執行連線模擬和離線模擬功能，需先選擇 .exob/.cxob 檔案的來源位置。 <b>離線模擬</b> 在 PC 上模擬工程檔案的運行，不與任何裝置連線。 <b>連線模擬</b> 在 PC 上模擬工程檔案的運行，此時 PLC 是直接與 PC 連接。
<b>Note</b>	<ul style="list-style-type: none"> <li>■ 在 PC 上進行 [連線模擬] 時，若監控設備是接在本地 PC 上的 PLC，監控時間有 1 小時的限制。</li> </ul>



## 2.3. 分析測試工具



設定	描述
<b>cMT Diagnoser</b>	cMT 系列用於監控物件的位址狀態、查詢通訊封包狀態，及 MQTT 與巨集的除錯。
	詳細資訊請參考《31 cMT Diagnoser》。
<b>EasyDiagnoser</b>	eMT、iE、XE、mTV 系列偵測 HMI 與 PLC 之間的通訊狀況。
	詳細資訊請參考《33 EasyDiagnoser》。
<b>EasyWatch</b>	可以透過 PC 監看或設定 HMI 和 PLC 內的位址數值。
	詳細資訊請參考《35 EasyWatch》。
<b>重新啟動 HMI</b>	不需拔除電源即可重新啟動 HMI，並恢復到一開機時的狀態。
<b>序列埠穿透</b>	允許 PC 上的應用程式透過 HMI 直接連結串列型 PLC。穿透通訊功能包含 [串列埠] 模式與 [乙太網路] 模式。在使用 [乙太網路] 穿透通訊功能前，請先安裝虛擬串列埠驅動程式。



詳細資訊請參考《29 Pass-through》。

#### 乙太網路穿透

當 HMI 為雙網路口機型，且 PC 與 PLC 分別連接到 HMI 上兩個不同的網路口時，允許 PC 上的應用程式透過 HMI 直接控制 PLC；僅支援 cMT 系列人機。



詳細資訊請參考《29 Pass-through》。

穿透通訊功能允許 PC 上的應用程式透過 HMI 直接連結 PLC，此時 HMI 所扮演的角色類似轉接器。



## 2.4. 傳輸



設定	描述
下載	使用乙太網路或 USB 線下載工程檔案到 HMI。
上傳	使用乙太網路或 USB 線上傳 HMI 的工程檔案到 PC。
建立 SD 卡或 USB 碟的下載資料	建立的資料可使用 SD 卡/USB 碟下載到 HMI。cMT / cMT X 系列需在 EasyBuilder Pro 軟體內建立 SD 卡或 USB 碟的下載檔案。

### 2.4.1. 下載 (PC->HMI)

透過此功能可以使用乙太網路或 USB 線下載檔案到 HMI 上。

eMT、iE、XE、mTV 系列

cMT、cMT X 系列



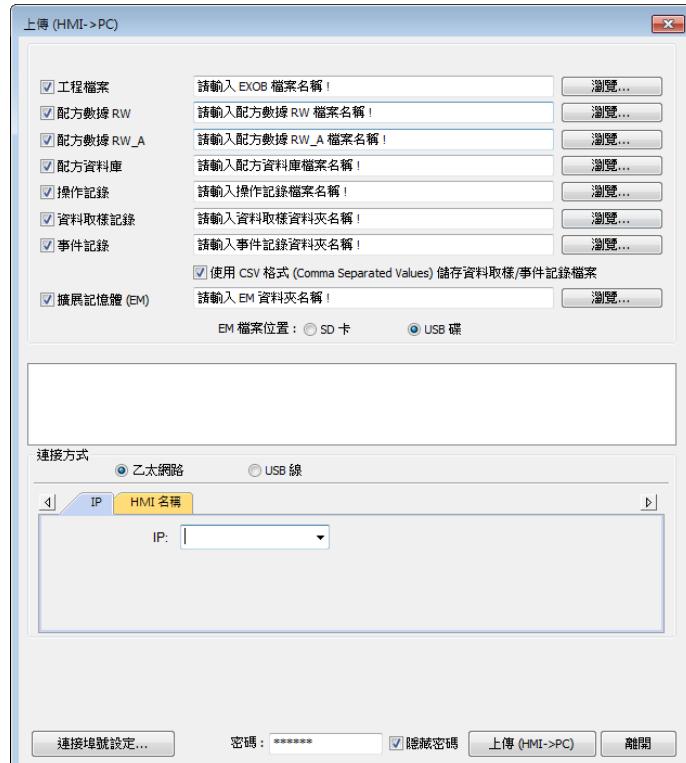
設定	描述
<b>工程檔案</b>	選擇 .exob, .cxob 格式的工程檔案。若工程檔案為 cMT/cMT X 系列，路徑下方會顯示該工程檔案的校驗和，可藉此比對工程檔案是否修改過。
<b>韌體</b>	若勾選此項，表示要更新 HMI 所有核心程式。第一次下載檔案至 HMI 時，一定要下載韌體。
<b>MQTT</b>	若檔案內有使用 MQTT 並選擇 HMI 當 Broker 時，則必須要勾選此項下載後才能使用。
<b>診斷工具</b>	啟用 HMI 診斷資料。當 HMI 發生異常運作時，透過蒐集診斷資料，可讓研發人員快速分析問題原因。
<b>USB 音源驅動程式</b>	啟用後，HMI 上接 USB 音效卡及喇叭即可輸出聲音。
<b>配方數據 RW / RW_A</b>	選擇 .rcp 格式的配方檔案。
<b>配方資料庫</b>	選擇 .db 格式的配方資料庫檔案。
<b>系統設定</b>	選擇 .conf 格式的系統設定檔案。
<b>開機畫面</b>	將指定的 .bmp 圖片下載到 HMI，HMI 啟動時，就會先顯示此圖片，再載入下載的程式。當

	cMT/cMTX 系列勾選下載工程檔案時，無法同時勾選下載使用者自訂畫面。
下載後啟動程式畫面	若勾選此項，HMI 將會在下載檔案成功後自動重新啟動。
連接埠號設定	設定透過乙太網路下載工程檔案所使用的連接埠。
同步 PC 時間至 HMI	將電腦時間同步至 HMI 上。
清除 配方數據 / 配方資料庫 / 事件記錄 / 資料取樣記錄 / 操作記錄 / / 字串表 / OPC UA history (HDA) 刪除開機畫面 / 刪除現存的使用者帳號 / 郵件聯絡人和 SMTP 設定	下載檔案前會先清除勾選的檔案。
CODESYS	透過 Utility Manager 下載 CODESYS boot project。

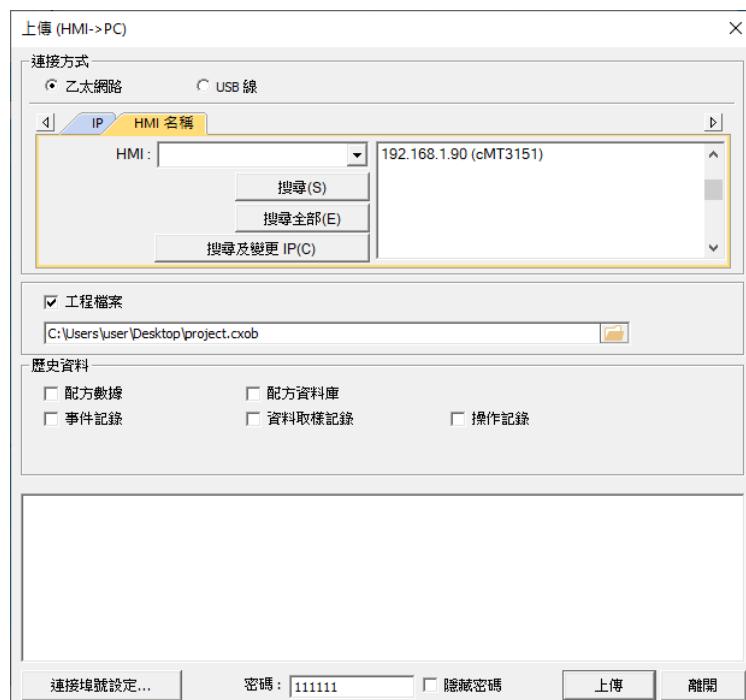
### 2.4.2. 上傳 (HMI->PC)

透過此功能可以使用乙太網路或 USB 線上傳 HMI 的檔案到 PC。上傳前須先選擇存放檔案的路徑。按下 [瀏覽]，指定上傳檔案所要存放的位置。

eMT、iE、XE、mTV 系列



cMT、cMT X 系列



設定	描述
事件記錄	將 HMI 的 .evt 檔案上傳到 PC。
擴展記憶體 (EM)	將 HMI 上的 SD 卡、USB 碟內的 .emi 檔案上傳到 PC。

關於 [工程檔案]、[RW / RW\_A]、[配方資料庫]、[資料取樣記錄] 請見本章《2.4.1 下載》。

### Note

- 若將工程檔案上傳至 PC，由於上傳回來的檔案格式為 .exob/ .cxob 檔案，使用者需先反編譯為 .emtp/ .cmtip 檔案，才能透過 EasyBuilder Pro 編輯。
- 上傳功能無法支援上傳儲存於外部裝置的歷史資料，但仍可透過 FTP 伺服器的方式，詳細資訊請參考《32 FTP 伺服器之運用》。

#### 2.4.3. 建立使用在 USB 碟與 SD 卡所需的下載資料



- 將 SD 卡或 USB 碟接上 PC。
- 指定檔案資料所要存放的路徑位置。
- 指定所要建立的來源資料檔案存放位置。
- 點選 [建立]。

所要建立的來源資料檔案將寫入所指定的外部裝置中，讓使用者可以透過該裝置下載工程檔案至 HMI，可以不需要透過乙太網路或 USB 線下載工程檔案。

#### 2.4.4. 透過 USB 碟 / SD 卡下載程式到 HMI 的步驟

假設已經把來源資料檔案建立在 USB 碟裡的資料夾名稱 “123” (K:\123)

1. 將 USB 碟 (工程檔案已包含在內) 接上 HMI。
2. 彈出 [Download / Upload] 視窗，請選擇 [Download]。
3. 輸入下載密碼。
4. 在 [Download Settings] 視窗，勾選 [Download project files] 以及 [Download history files]。
5. 點擊 [OK]。
6. 在 [Pick a Directory] 視窗，選擇路徑：usbdisk\disk\_a\_1\123。
7. 點擊 [OK]。

工程檔案將被自動更新。

### Note

- 若沒有下載工程檔案而只有下載歷史資料，必須手動去重新啟動 HMI 更新檔案。

## 2.5. 維護

此分頁根據選擇的機型不同有較大差異，請以該機型為主。

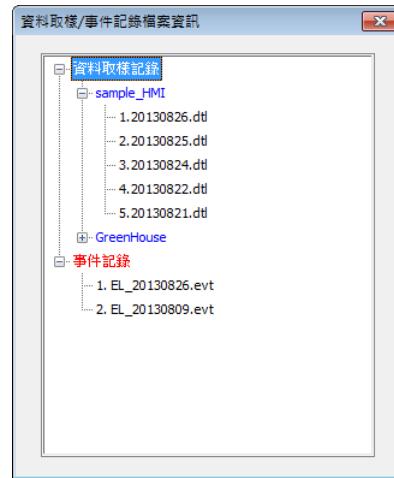


設定	描述
管理員工具	提供將 [使用者帳號]，[USB 安全金鑰]，[e-mail SMTP 同伺服器設定]，[e-Mail 連絡人] 四種資料儲存於 USB

碟。

 詳細資訊請參考《36 管理員工具》。

<b>cMT Viewer</b>	連線至 cMT / cMT X 系列，並顯示 cMT / cMT X 的工程檔案及數據資料。
<b>cMT-iV5/iV6 韌體更新</b>	更新 cMT-iV5/iV6 的 Runtime (韌體) 或 OS。
<b>cMT-Server OS 更新</b>	更新 cMT-SVR/SVR-200 的 OS。
<b>cMT-Gateway/CTRL OS 更新</b>	更新 cMT-Gateway/CTRL 的 OS。
<b>CODESYS 韌體更新</b>	更新 cMT/cMT X 的 CODESYS 韌體。
<b>SSL/TLS 憑證更新</b>	在 HMI 上更新內建已過期的 SSL/TLS �凭證。
<b>FTP 伺服器憑證更新</b>	更新 HMI 上 FTP 伺服器的憑證及私鑰。
<b>HMI 資訊收集</b>	從 HMI 端採集診斷相關資訊。
<b>EasyPrinter/Backup Server</b>	啟動遠端備份/列印伺服器。
<b>資料取樣/事件記錄檔 案資訊</b>	透過 USB 線或乙太網路連線查看 HMI 內的歷史資料記錄檔案個數。cMT / cMT X 系列不支援此功能。



**EasyAccess 1.0** 經由區域網路或網際網路來遠端遙控 HMI，詳細說明可至 [www.ihmi.net](http://www.ihmi.net) 查詢。

 **Note**

- HMI 資訊收集，僅 cMT/cMT-X 系列支援，HMI 需使用 OS 版本 20210705 或更新的版本。

## 2.6. 資料轉換



設定	描述
配方資料庫編輯器	可編輯配方資料。 請點選此圖示下載關於配方資料庫的文件。
EasyConverter	可讀取由 HMI 摷取的資料取樣記錄、事件記錄、操作記錄，並轉換成 Excel (.xlsx) 格式。 詳細資訊請參考《25 EasyConverter》。
配方資料/擴展記憶體編輯器	可建立 HMI 所使用的配方資料檔案，也可開啟及編輯現有的配方資料檔案。 詳細資訊請參考《24 Recipe Editor》。

# 3. 建立簡單的工程檔案

本章節說明如何建立一個工程檔案。

3.1.	概要 .....	3-2
3.2.	建立新的工程檔案 .....	3-2
3.3.	儲存和編輯工程檔案 .....	3-4
3.4.	開始編譯和快速編譯 .....	3-5
3.5.	離線模擬和連線模擬 .....	3-6
3.6.	cMT Viewer .....	3-6
3.7.	下載工程檔案至 HMI .....	3-7
3.8.	從 HMI 上傳工程檔案 .....	3-12

### 3.1. 概要

建立一個工程檔案的基本步驟如下：

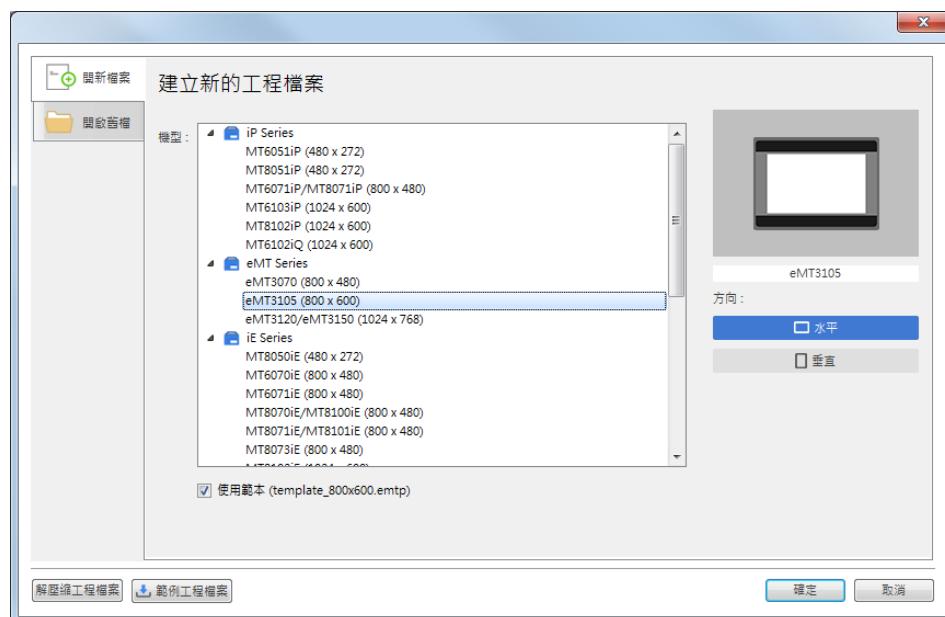
1. 建立新的工程檔案。
2. 儲存 / 編輯工程檔案。
3. 執行連線模擬 / 離線模擬。
4. 下載工程檔案至 HMI。

以下將說明每個步驟的設定方法。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

### 3.2. 建立新的工程檔案

1. 進入 EasyBuilder Pro 並開新檔案。
2. 選擇 [機型]，並勾選 [使用範本]。

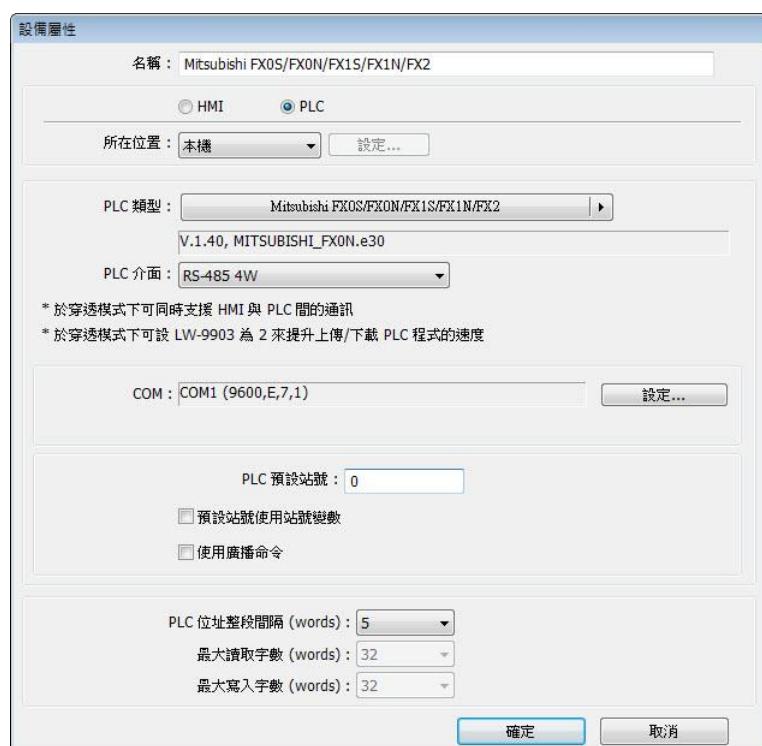


3. 在 [設備清單] 頁籤中點選 [新增]，設定欲連接的設備。

## 建立簡單的工程檔案



### 4. 設定正確參數。



### 5. [設備清單] 頁籤中增加了一個新的裝置。

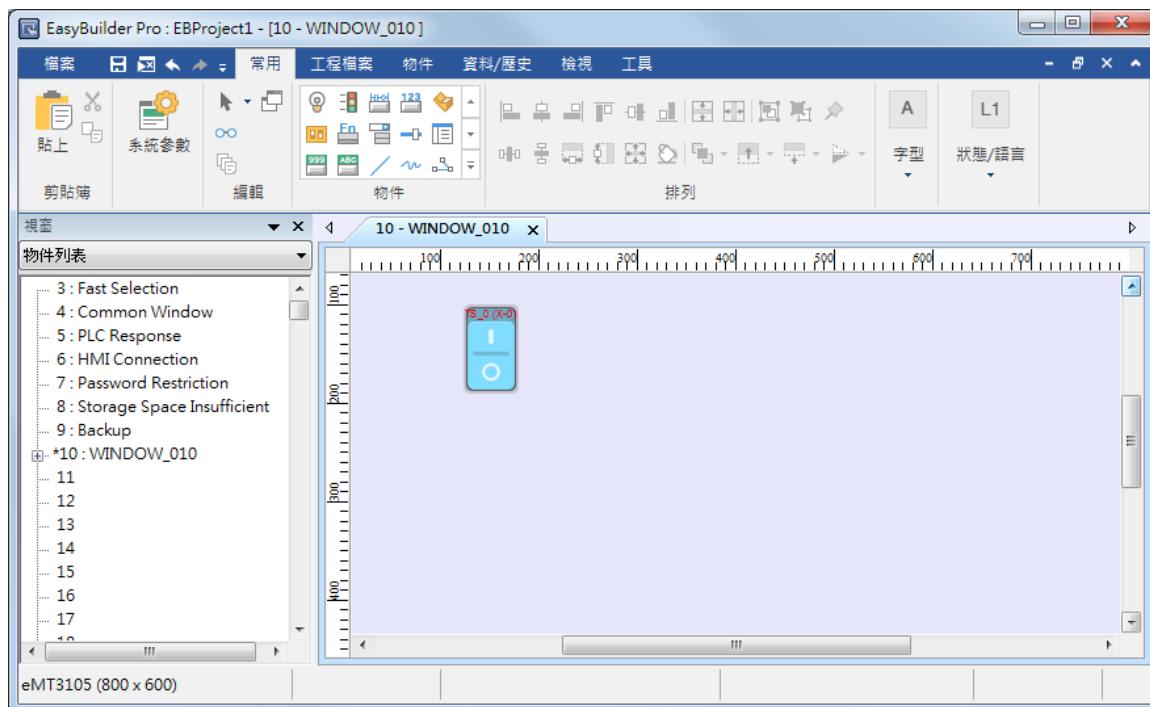


### 6. 建立一個新物件，以 [位元狀態切換開關] 物件為例，設定位址。

## 建立簡單的工程檔案



7. 將物件放置於編輯視窗中，即完成一個簡單的工程檔案。



### 3.3. 儲存和編輯工程檔案

1. 在 EasyBuilder Pro 的工具列上，點擊 [檔案] » [儲存檔案] 存成 .emtp 檔案。
2. 在 EasyBuilder Pro 的工具列上，點擊 [工程檔案] » [編譯] 將檔案編譯為 .exob 檔案。將此檔案下載至 HMI，並檢查工程檔案可否正常運行。



## 建立簡單的工程檔案

- cMT / cMT X 系列儲存的檔案格式為 .cmtp 檔案，編譯的檔案格式為 .cxob 檔案。
- 3. 使用者需事先於文字標籤庫設定多國語言後，再選擇工程檔案所需要的語言並下載至 HMI。編譯成功的對話視窗如下圖所示：



### 3.4. 開始編譯和快速編譯

編譯視窗的下方支援 [開始編譯] 和 [快速編譯] 兩種編譯模式。若工程檔案較大，但只有微調整幾個視窗時，使用 [快速編譯] 可節省編譯時間。

開始編譯：

完整編譯整個工程檔案。

快速編譯：

僅針對修改過的視窗或事件登錄進行編譯。若事件登錄中的其中一個項目曾被調整過，則會將整個事件登錄進行編譯。

#### Note

- EasyBuilder Pro 開啟檔案後的第一次編譯僅可支援 [開始編譯]。
- 若系統設定、圖庫、聲音庫、文字庫等有做過調整，調整後的第一次編譯僅支援 [開始編譯]。
- 若要進行下載工程檔案至 HMI，必須再次進行 [開始編譯] 才可下載。
- Utility Manager 不支援下載使用 [快速編譯] 的工程檔案。若要使用 Utility Manager 下載工程

檔案，則需要完整編譯。

### 3.5. 離線模擬和連線模擬



離線模擬：在 PC 上模擬工程檔案的運行，不需與任何裝置連線。



連線模擬：在 PC 上模擬工程檔案的運行，不需將程式下載到 HMI。此時 PLC 是直接與 PC 連接，請設定正確參數。

#### Note

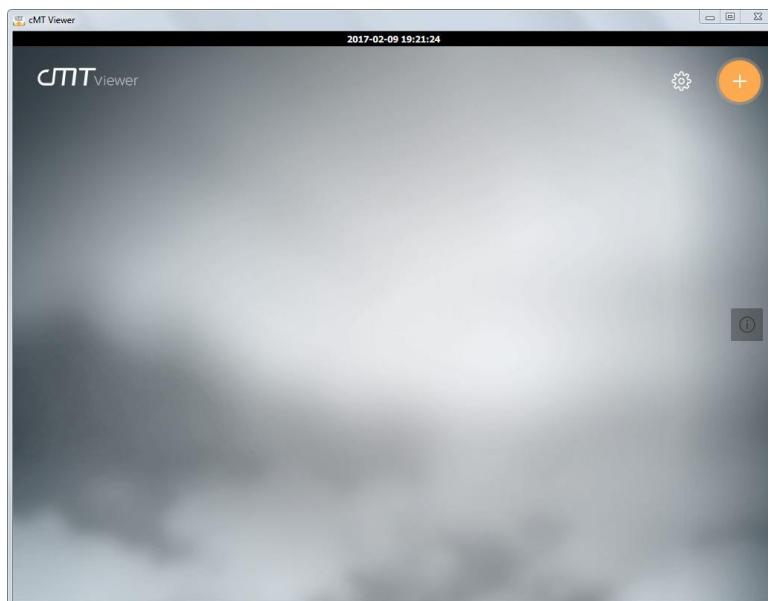
- iE/XE/eMT 系列在 PC 上進行 [連線模擬] 時，若監控設備是接在本地 PC 上的 PLC，監控時間會有 10 分鐘的限制。
- cMT/cMT X 系列在 PC 上進行 [連線模擬] 時，監控時間有 1 小時的限制且每 15 分鐘會有提示視窗。

### 3.6. cMT Viewer

cMT Viewer 可以利用網路連接到 cMT 系列，欲執行此程式請到安裝目錄下執行 cMTViewer.exe，或是在 EasyBuilder Pro 的工具列上，點擊 [工具] » [cMT Viewer]。



請點選此圖示觀看教學影片，請先確定已連上網路線。



### 3.7. 下載工程檔案至 HMI

以下介紹四種下載工程檔案至 HMI 的方法。

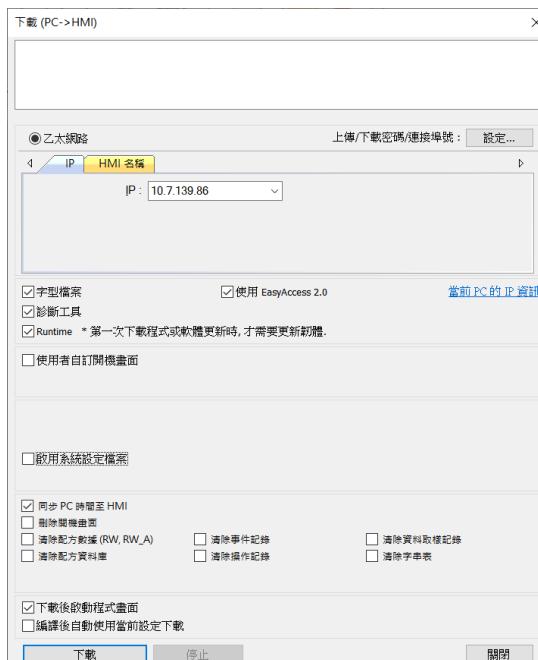


- cMT-SVR 只適用 3.6.1 的下載方法。
- 使用 mini USB 線下載工程檔案至 HMI 時，請不要同時連接 PLC 以避免 PLC 的雜訊影響到 HMI。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

#### 3.7.1. 自 EasyBuilder Pro 設定

1. 在 EasyBuilder Pro 的工具列上，點擊 [工程檔案] » [下載]。請先確認所有設定是否正確。
2. 選擇 [乙太網路]，設定 [密碼] 並指定 [HMI IP]。

eMT、iE、XE、mTV 系列



cMT、cMT X 系列



#### 設定

#### 描述

##### 字型檔案

將工程檔案中選用的字型下載至 HMI。

##### Runtime

勾選此選項表示要更新 HMI 上的所有核心程式。第一次下載工程檔案或更新 EasyBuilder Pro 版本並下載檔案至 HMI 時，一定要下載此韌體。

##### EasyAccess 2.0

下載 EasyAccess 2.0 的驅動程式至 HMI。(僅 MT8000iE 機型有此選項，其他機型則會自動下載)

##### 使用者自訂開機畫面

將指定的 bmp 圖檔下載至 HMI 作為啟動時的開機畫面。cMT/cMT X 系列請參考《5.4 一般屬性》在工程

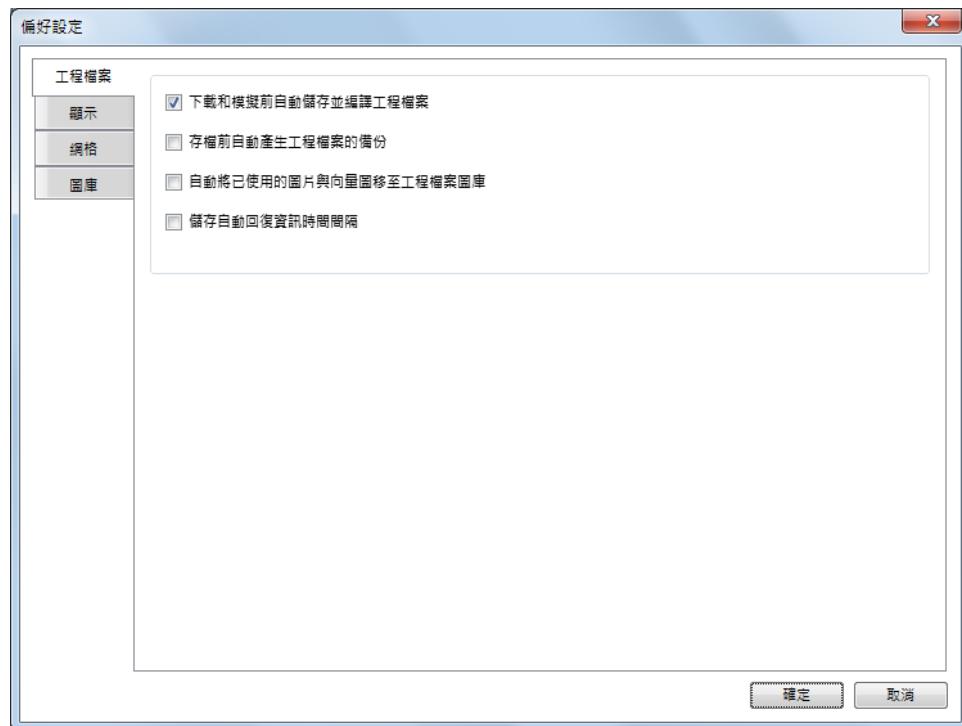
	檔案內更新 HMI 的開機畫面設定。
<b>啟用系統設定檔案</b>	下載可更新硬體設定資訊的檔案。
<b>同步 PC 時間至 HMI</b>	下載工程檔案時，將 HMI 的時間與電腦同步。
<b>刪除現存的使用者帳號、郵件連絡人和 SMTP 設定</b>	此選項如被勾選，下載程式前會先清除 HMI 上現有的使用者帳號、郵件連絡人和 SMTP 設定。此選項在 [系統參數設定] » [進階安全] 啟用 [在 HMI 上使用現有的使用者帳號] 或在 [郵件] 啟用 [在 HMI 上使用現有的連絡人設定] 時，才會有效。
<b>清除配方數據 / 事件記錄 / 資料取樣記錄 / 配方資料庫 / 操作記錄 / 字串表 / 刪除開機畫面</b>	選項如被勾選，下載程式前會先清除機器上所選取存在的檔案。
<b>下載後啟動程式畫面</b>	此選項如被勾選，下載程序完成後會自動重新啟動 HMI。
<b>編譯後自動使用當前設定下載</b>	如果勾選此項，下一次只要點選 [下載]，EasyBuilder Pro 將自動編譯程式並下載到上次下載的目標 HMI，請見下方說明。

**[編譯後自動使用當前設定下載]**

如果勾選此項，下一次只要點選 [下載]，EasyBuilder Pro 將自動編譯程式並下載到上次下載的目標 HMI。

1. 在 EasyBuilder Pro 的工具列上，點擊 [檔案] » [偏好設定]。
2. 勾選 [下載和模擬前自動儲存並編譯]。

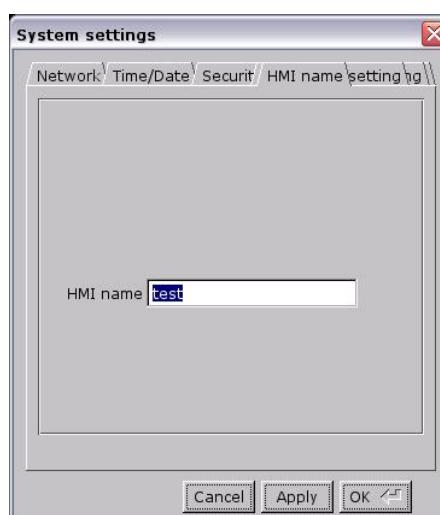
## 建立簡單的工程檔案



3. 在 EasyBuilder Pro 的工具列上，點選 [儲存檔案]，再點選 [下載]。
4. 勾選擬話窗上 [編譯後自動使用當前設定下載]。
5. 點擊 [下載]。
6. 完成以上設定後，下一次只要點選 [下載]，EasyBuilder Pro 將自動編譯程式並下載到上次下載的目標 HMI。

### 3.7.2. 使用 HMI 名稱

1. 在 HMI 上的 System settings 先設定 HMI name。



2. 在電腦上，選擇先前設定的 HMI 名稱並開始下載。若使用 [搜尋]，請在 [HMI 名稱] 中輸入要搜尋的特定 HMI 名稱。若使用 [搜尋全部]，則搜尋同網域內的所有 HMI。



### 3.7.3. 使用 USB 下載線



選擇 USB 線下載程式，其餘設定解說同《3.6.1 自 EasyBuilder Pro 設定》。使用 USB 線傳輸程式前，可至 [電腦管理] » [裝置管理員] 確認 USB 驅動是否安裝完成，若尚未被安裝，請參閱 [安裝步驟](#) 手動完成安裝。

### 3.7.4. 使用 USB 碟 / SD 卡

以下說明使用 USB 碟或 SD 卡下載工程檔案的步驟。

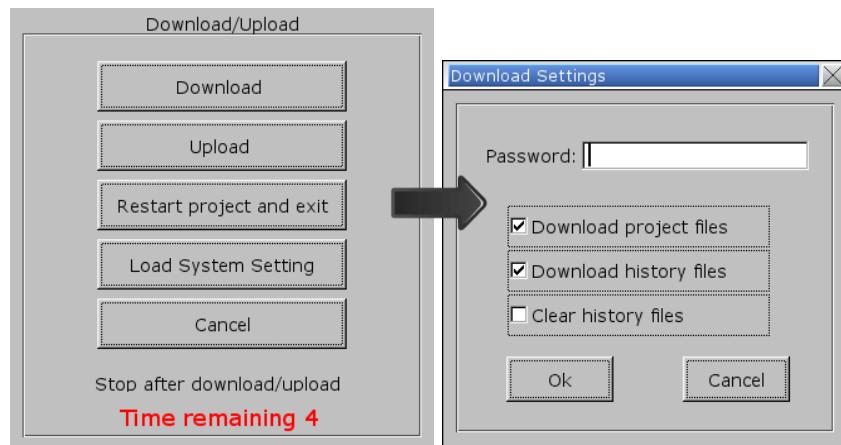
- 在 EasyBuilder Pro 的工具列上，點選 [工程檔案] » [建立使用在 USB 碟與 SD 卡所需的下載資料]。瀏覽欲下載的工程檔案後點選 [建立]，將該資料建立於外部裝置。



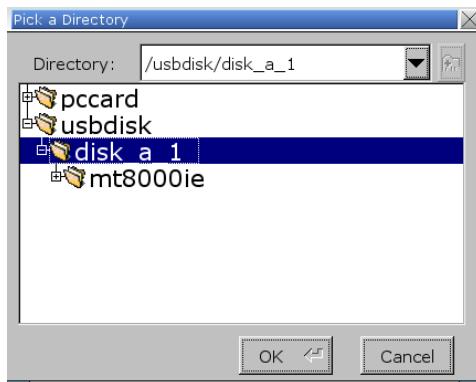
- 將外部裝置插入至 HMI。

## 建立簡單的工程檔案

3. 在 HMI 上選擇 [Download]，輸入密碼。



4. 密碼確認後會顯示外部裝置下的目錄名稱。(pccard : SD 卡；usbdisk : USB 碟)



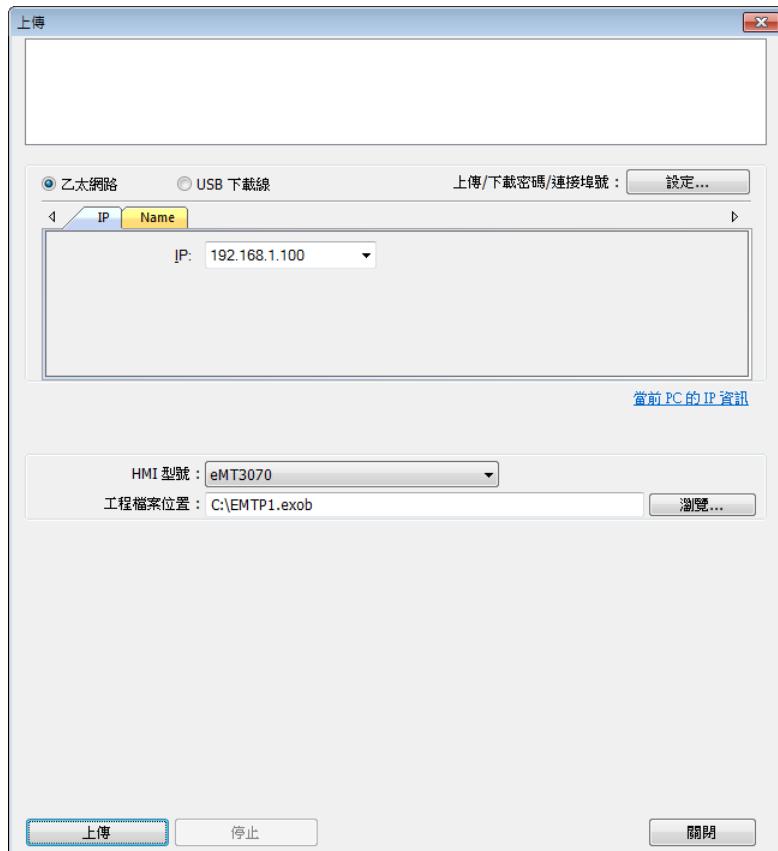
5. 選擇工程檔案的存放路徑，按下 [OK] 即開始下載。

 **Note**

- 此時下載檔案時必須選擇存放下載資料的上一層路徑，以上圖為例，必須選擇 disk\_a\_1 而非 mt8000ie。
- [系統設定] 是在 EasyBuilder Pro 軟體設定好 HMI 硬體的各項參數後，利用 USB 碟/SD 卡儲存設定並下載到 HMI 中。詳細使用說明請參考《4 硬體設定》。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

### 3.8. 從 HMI 上傳工程檔案

1. 在 EasyBuilder Pro 的工具列上，點選 [檔案] » [上傳]。
2. 設定欲上傳的 HMI 的 IP、型號、工程檔案名稱後，點選 [上傳] 即可。



# 4. 硬體設定

本章節說明硬體相關設定。

4.1.	概要 .....	4-2
4.2.	I/O 埠 .....	4-2
4.3.	LED 指示燈 .....	4-2
4.4.	系統設置 .....	4-2
4.5.	系統工具列 .....	4-3

## 4.1. 概要

本章節主要說明各系列硬體相關設定。

## 4.2. I/O 埠

HMI 支援的通訊介面，依不同機種而有所差異，詳細規格請參閱各機型的規格表。

- SD 卡插槽：提供工程檔案上傳及下載，包括配方數據、事件記錄與資料取樣記錄等，亦可備份或記錄歷史資料。
- 序列埠：連接 PLC 或其他設備，介面規格為：RS-232 / RS-485 2W / RS-485 4W / CAN Bus。
- 乙太網路：提供工程檔案上傳及下載，包括配方數據、事件記錄與資料取樣記錄。亦可連接具網路通訊功能之設備，如 PLC、PC 等。
- USB Host：支援各種 USB 介面的設備，如滑鼠鍵盤、隨身碟、印表機、條碼機等。
- USB Client：提供工程檔案上傳及下載，包括配方數據、事件記錄與資料取樣記錄等，亦可備份或記錄歷史資料。

當首次操作 HMI 前，必須在 HMI 上完成以下各系統設定，設定完成後即可使用 EasyBuilder Pro 軟體開發工程檔案。

## 4.3. LED 指示燈

LED 指示燈用於顯示 HMI 的操作狀態：

**機種: MT8121XE, MT8150XE, MT8121iE, MT8150iE**

LED	描述
PWR (橘燈)	表示電源狀態。
CPU (綠燈)	讀寫 Flash memory 時會閃爍。
COM (藍燈)	表示 COM port 通訊狀態，每一次通訊時都會閃爍，通訊良好時 LED 燈可能會保持恆亮。(不包含網路通訊)

**其他機種：**

LED	描述
PWR (橘燈)	表示電源狀態。
CPU (綠燈)	表示 CPU 狀態，閃爍或熄滅表示 CPU 異常。
COM (藍/紅燈)	表示通訊狀態，每一次通訊時都會閃爍，通訊良好時 LED 燈可能會保持恆亮。

## 4.4. 系統設置

部分 HMI 背後有一組重置按鈕及指撥開關，做不同模式切換時，將可觸發對應功能。若遺失 HMI

## 硬體設定

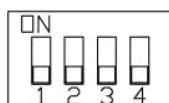
的系統設定密碼時，可以藉由調整指撥開關將 HMI 恢復成出廠預設值。詳細設定步驟如下：

1. 將 Dip Switch 1 切至 ON，其餘指撥開關保持為 OFF，然後重新啟動 HMI。此時 HMI 將進入觸控校正模式。
2. 在螢幕會出現 “+” 游標。使用觸控筆或者手指點選 “+” 游標的中心點進行五點校正。所有十字皆被準確觸控之後，“+” 游標會消失。校準參數會保留在系統裡。
3. 完成校正動作後，系統會詢問使用者是否將 HMI 的系統設定密碼回復為出廠設定，選擇 [Yes]。
4. 再次確認使用者是否要將 HMI 的系統設定密碼回復為出廠設定。當輸入 [yes] 按下 [OK] 後，HMI 內所有的程式檔案及歷史資料將全部被清除。  
(出廠時的 [Local Password] 預設密碼為 111111；但其他密碼，包括下載與上傳所使用的密碼於選擇恢復出廠值設定後，皆需重新輸入)。

以下為各機型指撥開關設置，請參閱相關安裝說明書。

### cMT / cMT X / eMT / iE

#### Dip Switch



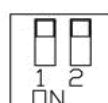
SW1	SW2	SW3	SW4	模式
ON	OFF	OFF	OFF	螢幕觸控校正模式
OFF	ON	OFF	OFF	隱藏系統工具列
OFF	OFF	ON	OFF	更新 OS
OFF	OFF	OFF	ON	保留
OFF	OFF	OFF	OFF	正常模式

#### Note

- 每台 HMI 的 Dip Switch 4 的開關位置可能有所不同。若 Dip Switch 4 的開關出廠時已被剪掉，代表此台 HMI 的 Dip Switch 4 必須設定為 ON。若 Dip Switch 4 的開關出廠時被保留，代表此台 HMI 的 Dip Switch 4 必須設定為 OFF。

### mTV

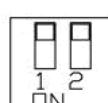
#### Dip Switch



SW1	SW2	模式
ON	ON	恢復原廠設定
ON	OFF	隱藏系統工具列
OFF	ON	更新 OS
OFF	OFF	正常模式

### cMT-SVR

#### Dip Switch



SW1	SW2	模式
ON	ON	恢復原廠設定
ON	OFF	恢復乙太網路 IP 設定
OFF	ON	更新 OS
OFF	OFF	正常模式

## 4.5. 系統工具列

啟動 HMI 後可利用在螢幕下方的 [工具列] 做系統設定，一般情況下它是自動隱藏的，使用者只

## 硬體設定

需點擊螢幕右下角的箭頭圖示即會彈出工具列，如下圖所示，由左而右為：系統設定、系統資訊、文字鍵盤、數字鍵盤。



隱藏 HMI 系統設定列的方法：

- 將 Dip Switch 2 設為 ON，系統設定列會被隱藏，設為 OFF，系統設定列便可被顯示並控制。使用者需重啟 HMI 來啟用/停止這個功能。
- mTV 系列則需將 Dip Switch 1 設為 ON 來隱藏系統設定列。
- 另外可使用系統暫存器 [LB-9020] 來顯示/隱藏系統設定列，當 [LB-9020] 設為 ON，此工具列會被顯示，當設為 OFF，則會被隱藏。

### 4.5.1. 系統設定

#### cMT、cMT X 系列

自 OS 版本 20231201 起，cMT X 系統頁面經過了重大的改版，因此存在兩種不同的設定方式。以下將分別介紹這兩種設定方法。

設定或變更 HMI 的各項系統參數，基於安全考量必須進行密碼確認。出廠時的預設密碼為 111111。

**20231201** 之後的版本



#### 主題切換

提供淺、深色模式，用戶可以依據環境搭配適當主題。

#### 系統屬性

##### [Model]

顯示您的 HMI 型號。

##### [HMI 名稱]

設定 HMI 名稱以便於下載/上傳工程檔案時辨識。

##### [OS 版本]

顯示當前 OS 的版本號。

##### [硬體金鑰]

顯示 HMI 的硬體金鑰。

##### [LAN1 實體位址]

LAN1 MAC 地址。

##### [LAN2 實體位址]

LAN2 MAC 地址。

##### [時間]

設定 HMI 內本地的時間。

**[日期]**

設定 HMI 內本地的日期

**[亮度]**

調整螢幕亮度。

**[聲音]**

調整系統音量。

**[語言]**

選擇系統介面語言。

**[輸入法]**

選擇系統鍵盤的輸入法，可以多選。

**[顯示 Start Button]**

啟用以顯示 Start 按鈕。

如果不啟用，可以選擇按鈕的位置，只要長按 1.5 秒就可以顯示。

**[顯示下載彈出視窗]**

啟用或禁用下載彈出窗口。

**[顯示滑鼠游標]**

啟用或禁用滑鼠游標。

**[系統鍵盤聲音]**

啟用或禁用系統鍵盤聲音。

**[方向]**

選擇顯示方向，可選擇 0 度、90 度、180 度或是 270 度。

**[系統指標]**

查看 CPU、記憶體、eMMC 和網路的使用情況。

**[重置 HMI 為預設]**

重置 HMI 至出廠設定。

**[OS 更新]**

選擇 SD 卡或是 USB 碟中的 OS 檔案後開始更新系統韌體。更新時，請勿將電源關閉，詳細更新步驟及注意事項，請參閱各機型的 OS 更新說明。

**[第三方版權]**

查看並管理第三方授權。



## CODESYS

### [開通狀態]

查看 CODESYS 的開通狀態。

### [CODESYS 狀態]

查看 CODESYS 的狀態。

### [應用程式狀態]

查看應用程式的運行狀態。。

### [登入狀態]

查看登入狀態。

### [版本]

查看 CODESYS 的版本號。

### [Ethernet (LAN1)]

設定 IP 地址、遮罩和閘道。



## 工程檔案

### [清除歷史]

清除與工程檔案相關的各種歷史數據，如配方、事件記錄、資料取樣等。

### [重啟 EB 工程檔案]

重新啟動 EB 工程檔案。



## 網路

用乙太網路下載工程檔案到 HMI 上，需要正確設定 HMI 的 IP 位址。您可以選擇自動分配 IP 位址或手動輸入。

### [Ethernet LAN1]

一般用於連接外部網絡，需要設定閘道 (Gateway)和網域名稱系統(DNS)。

### [Ethernet LAN2]

一般用於內部網絡連接，如連接 PLC。為避免網路衝突，LAN1 和 LAN2 應設置在不同的網域。

### [網路橋接]

雙網路孔機種，可以啟用橋接模式使 LAN1 以及 LAN2 在同一個網段，達到交換機功能使用。

## 硬體設定



### 安全

管理不同使用者的密碼。

#### [Admin]

最高權限，可以操作所有功能與設定，包含上傳下載檔案，備份歷史資料，修改各權限的密碼。

#### [Update Engineering Files]

可使用此密碼上傳下載檔案，備份歷史資料。

#### [History]

此密碼用於 FTP Server 登入與備份歷史資料。

#### [Users]

使用於 cMT Viewer/診斷器登入，上方三個權限的密碼也都可以在 cMT Viewer 上來使用。

### EasyAccess 2.0

EasyAccess 2.0 的資訊頁面。

請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。



### VNC

#### [啟用]

當啟用 VNC 功能後，您可以允許 VNC Client 裝置連接到 HMI。這使得遠程用戶可以查看和操作 HMI 界面。

#### [多台連線]

允許多個 VNC Client 裝置連接。

請注意，同時連接的 VNC Client 裝置越多，可能會對 HMI 的通訊速度產生影響。

#### [需要密碼]

為了保護系統安全，您可以設定一個密碼，只有輸入正確密碼的 VNC Client 才能連接到 HMI。

### VNC 互鎖

#### [啟用]

啟用 VNC 互鎖功能，以增強系統安全性。

## 硬體設定

和管理性。

### [超時]

設定 VNC 互鎖的超時時間，超過設定時間無操作，則操作權將自動斷開。

### [狀態欄模式]

選擇狀態欄的顯示模式，可根據使用需求進行調整。



## FTP

### [啟用]

啟用 FTP 服務後，用戶可以通過網絡使用 FTP 客戶端來訪問 HMI 上的文件。

### [FTP 埠號]

設定 FTP 服務的埠號。預設埠號是 21。

### [FTP Client 端可以修改 USB/SD 的資料]

此選項允許 FTP 客戶端修改連接到 HMI 的 USB 碟或 SD 卡上的文件。啟用此功能時，請確保適當的安全措施已經到位，以防止未經授權的訪問。



## 印表機

### [名稱]

顯示連接到 HMI 的印表機名稱。

### [版本]

顯示當前印表機驅動的版本。

### [URI]

顯示印表機的 URI，用於定位印表機。

### [列印測試頁]

點擊此選項可列印測試頁，以檢查印表機是否正常運作。

### [備份 PPD 檔案]

備份 PPD 檔案，這是一個描述印表機功能的文件。

**20231201 之前的版本****[安裝印表機]**

進行印表機的安裝設定。

詳細設定可參考章節 23.3 安裝印表機。

**CODESYS**

請輸入啟用碼來啟用 CODESYS

**一般屬性**

可在此頁面調整 FTP 設定以及其他一般設定。

**[顯示方式]**

可選擇 0 度、90 度、180 度或是 270 度。

**[亮度]**

可調整螢幕顯示亮度。

**[彈出下載視窗]**

啟用後可彈出下載視窗。

**[顯示 Start button]**

啟用以顯示 Start 按鈕。

**[顯示滑鼠游標]**

啟用以顯示滑鼠游標。

**網路**

用乙太網路下載工程檔案到 HMI 上，需正確設定操作對象 (HMI) 的 IP 位址。可選擇自動分配 IP 位址或自行輸入 IP 位址。若需使用 Email 及 EasyAccess2.0 功能，需正確設定網域名稱系統。若 HMI 具備兩個乙太網路端口，為避免網路發生衝突，請將兩個 IP 位址設成不同的網域。

**[啟用橋接模式]**

雙網路孔機種，可以啟用橋接模式使 LAN1 以及 LAN2 在同一個網段，達到交換機功能使用。

## 硬體設定



### 時間 / 日期

設定 HMI 內本地的日期與時間。



### HMI 名稱

設定 HMI 名稱以便於下載/上傳工程檔案時辨識。



### 清除歷史紀錄

清除儲存於 HMI 內的歷史記錄。



### 工程檔案管理

點擊重啟將 HMI 重新開機。

## 硬體設定



### 系統密碼

HMI 的密碼防護，預設密碼為 111111，請使用者設定自己的密碼，完成後始可使用該密碼。

### 進階安全模式

可新增、刪除使用者帳號，並且管理不同使用者的權限。

### VNC 設定

啟用後，可使用 VNC 透過乙太網路監控遠端 HMI。

#### [開啟 VNC]

允許 VNC Client 裝置連接。

#### [VNC 多台連線]

允許多個 VNC Client 裝置連接。同時連接越多 VNC Client 裝置可能影響 HMI 的通訊速度。

### EasyAccess 2.0

EasyAccess 2.0 的設定頁。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 硬體設定



### 印表機

詳細設定可參考章節 23.3 安裝印表機。



### OS 更新

選擇 SD 卡或是 USB 碟中的 OS 檔案後開始更新系統韌體。更新時，請勿將電源關閉，詳細更新步驟及注意事項，請參閱各機型的 OS 更新說明。



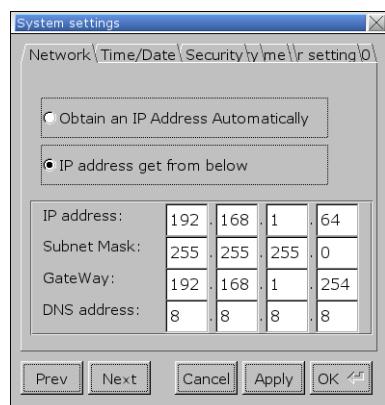
### 重置選項

將 HMI 重置為出廠設定。

## 硬體設定

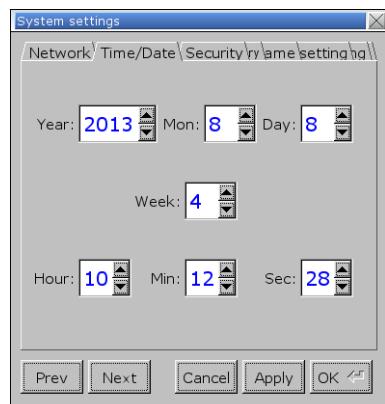
### eMT、iE、XE、mTV 系列

設定或變更 HMI 的各項系統參數，基於安全考量必須進行密碼確認。出廠時的預設密碼為 111111。



#### Network

用乙太網路下載工程檔案到 HMI 上，需正確設定操作對象 (HMI) 的 IP 位址。可選擇自動取得 IP 位址或自行輸入 IP 位址。若需使用 Email 及 EasyAccess2.0 功能，需正確設定 DNS address。若 HMI 具備兩個乙太網路端口，為避免網路發生衝突，請將兩個 IP 位址設成不同的網域。



#### Time / Date

設定 HMI 內本地的日期與時間。



#### Security

HMI 的密碼防護，預設密碼為 111111，請使用者設定自己的密碼，完成後始可使用該密碼。

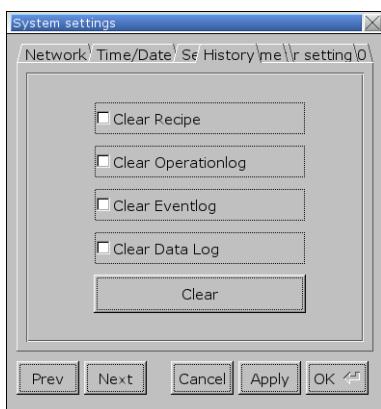
[進入系統設定的密碼]

[上傳工程檔案的密碼]

[下載工程檔案的密碼]

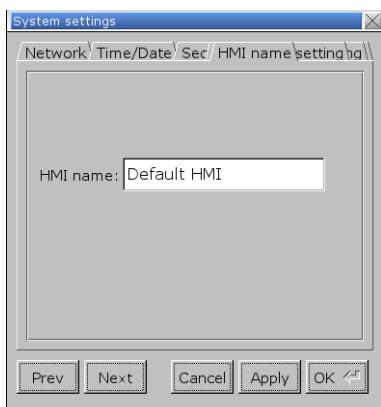
[上傳歷史記錄的密碼]

## 硬體設定



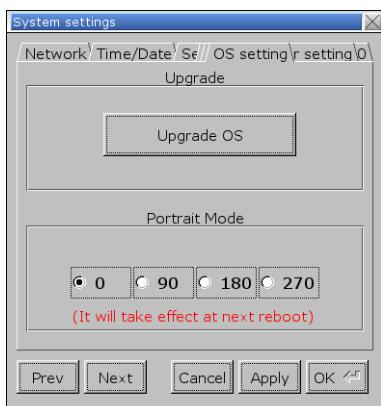
### History

清除儲存於 HMI 內的歷史記錄：  
 [清除配方數據]  
 [清除操作紀錄]  
 [清除事件記錄]  
 [清除資料取樣]



### HMI name

設定 HMI 名稱以便於下載/上傳工程檔案時辨識。



### OS setting

[Upgrade OS] 更新系統韌體。更新時，請勿將電源關閉，詳細更新步驟及注意事項，請參閱各機型的 OS 更新說明。

[Portrait Mode] 調整顯示模式。當調整了顯示模式後，需將 HMI 重新上電才可生效。也就是說，HMI 的電源需完全移除再重新上電。若是使用直立式顯示模式（90 度和 270 度），則工程檔案也必須為垂直的工程檔案，否則無法正確顯示。

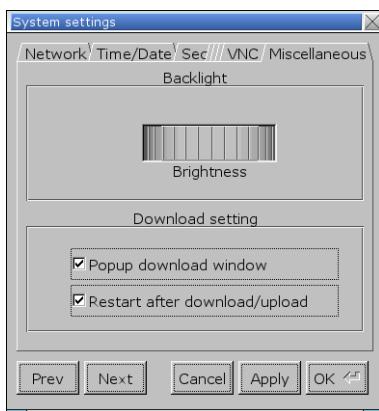
### VNC server setting

啟用後，可使用 VNC 透過乙太網路監控遠端 HMI。

[Start VNC Single-connection]  
 允許一台 VNC Client 裝置連接。

[Start VNC multi-connection]  
 允許多個 VNC Client 裝置連接。  
 同時連接越多 VNC Client 裝置可能影響 HMI 的通訊速度。  
 設定步驟請見下頁說明。

請點選此圖示觀看教學影片，請先確定已連上網路線。

**Misc 1**

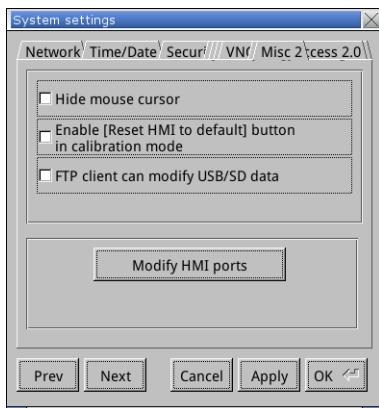
旋鈕可調整 LCD 畫面亮度。

**[Popup download window]**

啟用後，接上 USB 碟或 SD 卡後，可顯示上傳/下載選項視窗。

**[Restart after download/upload]**

啟用後，當工程檔案被上傳或下載後會重新啟動 HMI。

**Misc 2****[Hide mouse cursor]**

隱藏滑鼠游標。

**[Enable [Reset HMI to default] button in calibration mode]**

啟用後，在開機時長按螢幕可以進入觸控模式，觸控校正完成後，會顯示 [Reset HMI to default] 選項。

**[FTP client can modify USB/SD data]**

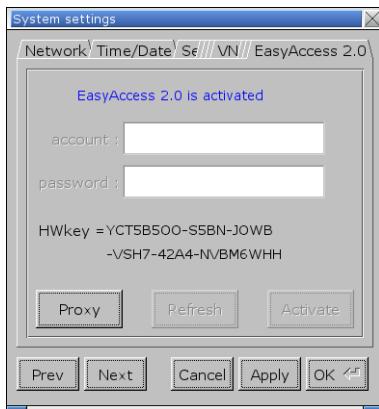
啟用後，可透過 FTP 伺服器修改 USB 碟/SD 卡中的資料。

**[Modify HMI ports]**

修改上傳/下載的埠號及 FTP 的埠號。

**EasyAccess 2.0**

開啟 EasyAccess 2.0 的設定頁。



 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

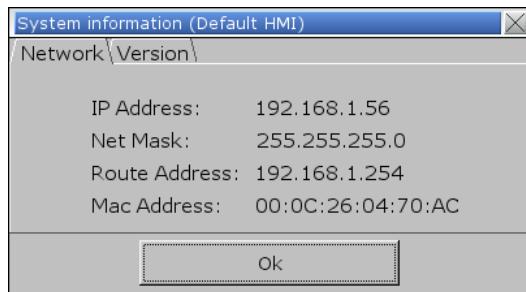
以下為設定 VNC Server 的步驟。

1. 開啟 HMI 的 VNC server 並設置登入密碼。
2. 安裝 Java IE 或 VNC viewer 於 PC。
3. 安裝 Java IE 後可透過網路瀏覽器輸入遠端 HMI 的 IP 位址。  
或者透過 VNC viewer 輸入遠端 HMI 的 IP 位址和密碼。

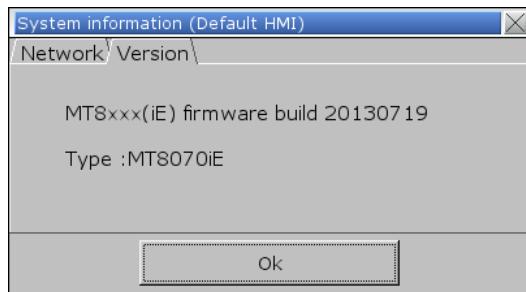


#### 4.5.2. 系統資訊

**Network:** 顯示網路資訊，包含 HMI 的 IP 位址等。



**Version:** 顯示 HMI 系統版本及機型資訊。



# 5. 系統參數設定

本章節說明 EasyBuilder Pro 各項系統參數設定。

5.1.	概要 .....	5-2
5.2.	設備清單 .....	5-2
5.3.	HMI 屬性 .....	5-11
5.4.	一般屬性 .....	5-15
5.5.	系統設定 .....	5-22
5.6.	遠端 .....	5-26
5.7.	使用者密碼 .....	5-27
5.8.	擴展記憶體 .....	5-30
5.9.	列印/備份伺服器 .....	5-34
5.10.	時間同步/夏令時間 .....	5-35
5.11.	郵件 .....	5-37
5.12.	行動網路 .....	5-39
5.13.	FTP .....	5-41

## 5.1. 概要

在 EasyBuilder Pro 表單點選 [常用] 下的 [系統參數] 後，將顯示 [系統參數設定] 對話視窗。下面將說明各頁籤的內容。

## 5.2. 設備清單

用來設定 HMI 欲連接裝置之屬性，這些裝置包括本機或遠端的 HMI 或設備。當開啟新的工程檔案時，會預設一個“本機 HMI”裝置，用來辨識此機型。

### 5.2.1. 本機 HMI 設定



設定	描述
啟用安全通訊	當啟用安全通訊功能時，HMI 會進行加密通訊，以確保資料傳輸的安全。此功能僅 cMT X 系列 HMI 支援，且只有同樣支援安全通訊的設備才能進行互相通訊。
LW 保護、RW 保護、資料保護	關於資料保護的具體設定方法，請參閱本手冊的《第 39 章：資料保護》。

### 5.2.2. 如何控制一台本機 PLC



“本機設備”是指與本機 HMI 直接連接的設備，若要控制本機設備時，需先新增此種類型的裝置。點選 [設備清單] 頁籤中的 [新增]，即可開啟 [設備屬性] 視窗。

以下將以 MODBUS RTU 做為本機設備：



設定	描述
<b>名稱</b>	顯示裝置名稱。
<b>HMI 或設備</b>	此範例的連接裝置為設備，所以此時選擇 [設備]。
<b>所在位置</b>	可以選擇 [本機] 或 [遠端]，此範例設備連接在本機 HMI 上，所以選擇 [本機]。
<b>設備類型</b>	選擇設備的型號。
<b>介面</b>	設備所使用的介面類型，可以選擇 [RS-232]、[RS-485 2W]、[RS-485 4W]、[乙太網路]、[USB] 以及 [CAN Bus]。 <ul style="list-style-type: none"> <li>● 介面類型如果為 [RS-232]、[RS-485 2W]、[RS-485 4W]，點選 [設備屬性] 對話窗中的 [設定]，可以開啟 [通訊埠設定] 對話視窗，並設定通訊參數。</li> </ul>



### 超時

通訊中斷超過此項設定值 (單位為秒)，HMI 會使用 5 號視窗 “Device No Response” 做為提示。

### 通訊延時

HMI 在送出下一個命令給設備前，會先延遲此項設定值 (單位為毫秒)，再送出命令。此項設定值會降低 HMI 與設備間的通訊效率，若無特殊需求，設定為 “0” 即可。

若使用的設備為 Siemens S7-200 系列，則不能忽略此項設定值，建議將 [通訊延時] 設定為 “5”，[ACK 訊號延時] 設定為 “30”。

- 介面類型如果為 [乙太網路]，點選 [設備屬性] 對話窗中的 [設定] 可以開啟 [IP 位址設定] 對話視窗，使用者必須正確設定設備的 IP 位址與連接埠。



- 介面類型如果為 [USB]，就不需再設定，檢查 [設備屬性] 的各設定值是否正確即可。
- 介面類型如果為 [CAN Bus]，請參照《設備連接手冊》中關於“CANopen”的說明，並匯入 .eds 檔案。

#### 設備預設站號

設定設備位址所使用的預設站號。當位址內容不包括站號資訊時，將使用此項設定值做為設備的站號。

也可將設備站號資訊直接設定在位址內容中，此時位址格式為 ABC#Addr

其中 ABC 表示設備所使用的站號，必須大於等於 0，且小於等於 255。Addr 指定設備的位址，兩個數據之間以“#”做為區隔。如下圖所示，顯示此時將讀取設備站號為 1 的 0x-20 位址之內容。



## 系統參數設定

### 預設站號使用站號變數

設備預設站號可以使用站號變數。利用 LW-10000~LW-10015 (var 0 ~ var 15) 來設定站號變數。當設備預設站號選擇使用站號變數時，若在設備位址中未指定所使用的站號，則站號一律由預設站號所指定的站號變數來決定。  
若設備預設站號已選擇 var3：



以下使用幾個例子說明。

- 所操作的設備站號為 5。



- 所操作的設備站號由 var7 (LW-10007) 來決定。



- 設備的位址為 "111"，此時不指定設備站號，但因為已使用預設站號 var3，所以所操作的設備站號由 var3 (LW-10003) 來決定。



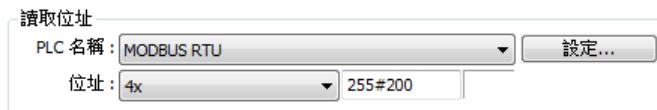
 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 使用廣播命令

當勾選 [使用廣播命令] 後，依照所使用的設備定義之廣播站號，填入至 [廣播命令所使用的站號]。當 HMI 使用廣播站號發送命令時，設備將只接收命令而不回覆 HMI。



如下圖所示：



假設廣播站號為 255，當 HMI 發送命令至 255#200 這個位址時，所有的設備會接收這個命令但不回覆 HMI。

## 系統參數設定

有支援廣播命令的設備才適用此功能。

<b>位址整段間隔 (words)</b>	不同讀取命令的讀取位址之間距若小於此項設定值，這些命令可以合併為同一個命令。此項設定值如果為“0”，將取消命令合併功能。 若此項設定值為“5”，當分別從 LW-3 讀取 1 個 word 與從 LW-6 讀取 2 個 word 的數據 (即 LW-6 與 LW-7 的內容) 時，因 LW-3 與 LW-6 的位址差距小於 5，此時可以將兩個命令合併為一個命令，合併後的命令內容為從 LW-3 開始連續讀取 5 個 words 的數據 (讀取 LW-3 ~ LW-7)。需注意，可以被合併的命令之讀取數據大小將不會大於 [最大讀取字數 (words)]。
<b>最大讀取字數 (words)</b>	一次可以從裝置讀取數據的最大量，單位為 word。
<b>最大寫入字數 (words)</b>	一次可以寫入到裝置的數據最大量，單位為 word。

完成上述的各項設定後，在 [設備清單] 中可以發現新增了一個名稱為“本機 PLC 1”的裝置。



### Note

- 使用 cMT / cMT X 時，可以在 [系統參數設定] 對話窗中，選擇“本機 HMI”，按下 [新增設備] 按鈕，便會在“本機 HMI”之下，新增一個“本機 PLC 1”，如下圖所示。



請點選此圖示觀看教學影片，請先確定已連上網路線。

### 5.2.3. 如何控制一台遠端設備



“遠端設備”是指與遠端 HMI 連接的設備。若要控制遠端設備需先增加此種類型的裝置。點選 [設備清單] 頁籤的 [新增]，即可開啟 [設備屬性] 對話窗設定各項屬性。

以下將以 MODBUS RTU 做為遠端設備：



設定	描述
HMI 或設備	連接裝置為設備，所以此時選擇 [設備]。
所在位置	可以選擇 [本機] 或 [遠端]。因設備連接在遠端 HMI 上，所 以此時選擇 [遠端]，並且設定遠端 HMI 的 IP 位址及連接埠。 請在 [設備屬性] 對話窗按下 [所在位置] 旁的 [設定]。

## 系統參數設定



<b>設備類型</b>	選擇遠端設備的型號。
<b>介面</b>	遠端設備所使用的介面類型，若遠端設備使用序列埠時，介面需選擇 [RS-232]，[RS-485 2W]，[RS485 4W] 任一種。
<b>COM</b>	遠端設備連接遠端 HMI 時所使用的序列埠。此項內容必須正確設定。
<b>設備預設站號</b>	遠端設備所使用的站號。

完成上述的各項設定後，在 [設備清單] 中可以發現新增了一個名稱為“遠端 設備 1”的裝置。



### Note

- 使用 cMT / cMT X 系列時，可以在 [設備清單] 頁籤中，選擇已建立的“遠端 HMI 1”，按下 [新增設備] 按鈕，便會在“遠端 HMI 1”之下，新增一個“遠端 設備 1”，如下圖所示。



請點選此圖示觀看教學影片，請先確定已連上網路線。

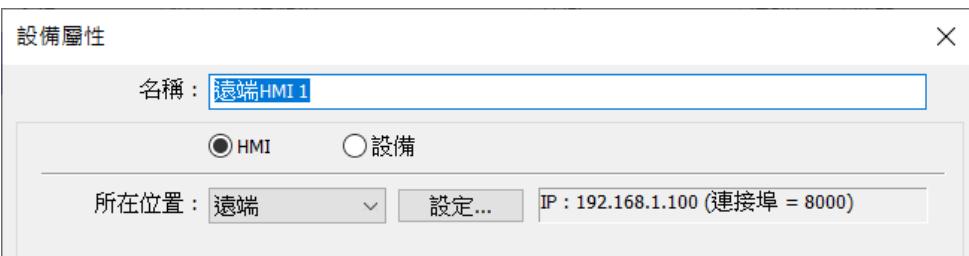
### 5.2.4. 如何控制一台遠端 HMI



“遠端 HMI”是指非本機 HMI 的所有其他 HMI，PC 也被視為遠端 HMI 的一種。若要控制遠端 HMI

## 系統參數設定

需先新增此種類型的裝置。點選 [設備清單] 頁籤的 [新增]，即可開啟 [設備屬性] 對話窗設定各項屬性。



設定	描述
HMI 或設備	連接裝置為 HMI，所以此時選擇 [HMI]。
所在位置	可以選擇 [本機] 或 [遠端]，因為使用遠端 HMI，此時選擇 [遠端]，並且必須設定遠端 HMI 的 IP 位址及連接埠。請在 [設備屬性] 對話窗按下 [所在位置] 旁的 [設定]。

完成上述的各項設定後，在 [設備清單] 即新增了一個為“遠端 HMI 1”的裝置。

系統參數設定						
設備	HMI 屬性	一般屬性	系統	遠端	使用者密碼	擴展記憶體
設備清單:						當前 PC 的 IP 資訊
本機 HMI	名稱 Local HMI	位置 本機	設備類型 MT8073IE (800 x 480)	介面 -	通訊協議 -	站號 0
遠端 HMI 1	遠端HMI 1	遠...	eMT/XE/IE/IP/mTV 系列	乙太網路	TCP/IP	N/A

### Note

- 使用 cMT / cMT X 系列時，可以在 [設備清單] 頁籤中，按下 [新增 HMI] 按鈕，便會新增一個“遠端 HMI 1”，如下圖所示。

系統參數設定						
設備	HMI 屬性	一般屬性	系統	遠端	使用者密碼	擴展記憶體
設備清單:						當前 PC 的 IP 資訊
本機 HMI	名稱 Local HMI	位置 本機	設備類型 cMT3072XH/cMT3072XH2/cMT3072XHT (1024 x 600)	介面 -	通訊協議 -	
遠端 HMI 1	HMI	遠... HMI		乙太網路	TCP/IP	

- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

### 5.3. HMI 屬性

[HMI 屬性] 設定頁用來設定 [HMI 型號]、[時鐘來源]、[印表機] 及 [捲動軸] 寬度等。

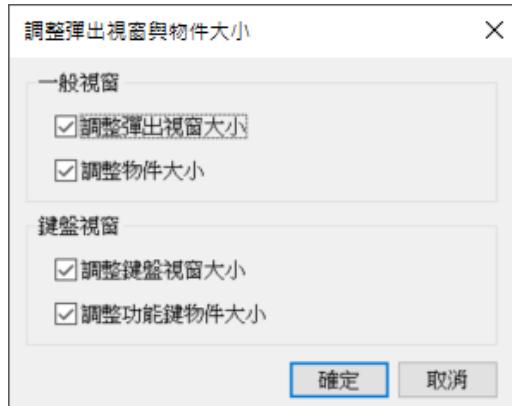
#### eMT、iE、XE、mTV 系列



## cMT、cMT X 系列



設定	描述
<b>HMI 型號</b>	<p>請選擇欲使用的 HMI 型號。</p> <p>當使用者更換其他的 HMI 型號並按確認鍵時，若更換後的機型之解析度與前者不同，將會彈出 [調整彈出視窗與物件大小] 對話窗如下圖所示。按下確認後即完成變更 HMI 型號的設定。</p>

**水平/垂直模式**

設定工程檔案於 HMI 端的顯示模式。

<b>HMI 站號</b>	選擇 HMI 所使用的站號，若無特殊目的，採用預設值即可。														
<b>連接埠</b>	設定 HMI 所使用的通訊埠號碼，若無特殊目的，採用預設值即可。														
<b>支援 cMT 通訊協議</b>	支援與 cMT / cMT X 系列的 HMI 相互通訊。cMT / cMT X 系列的工程檔案也需設定相同的連接埠。														
<b>印表機 (eMT、iE、XE、mTV 系列)</b>	<p><b>型號</b> 顯示目前支援的印表機類型，其中 HP PCL 系列需使用 USB 介面連接，其他類型印表機則需使用序列埠連接。</p> <p> 詳細資訊請參考《23 HMI 支援的印表機類型》。 使用序列埠連接的印表機需正確設定序列埠的通訊參數。當印表機的型號為 [SP-M, D, E,F] 時，需設定【每行點數】，此設定值不可超過印表機每行可以列印的點數，否則將造成錯誤的列印結果。</p>														
<b>印表機 (cMT、cMT X 系列)</b>	<p><b>可在 HMI 上安裝印表機驅動程式。</b></p> <p><b>類型</b> 選擇印表機的型號。</p> <p><b>狀態位址</b> 顯示印表機的運作情形。</p> <table border="1" data-bbox="539 1594 1333 1987"> <thead> <tr> <th>LW-n</th> <th>狀態</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>印表機的驅動程式尚未安裝</td> </tr> <tr> <td>1</td> <td>正在安裝印表機的驅動程式</td> </tr> <tr> <td>2</td> <td>印表機已就緒</td> </tr> <tr> <td>3</td> <td>印表機正在列印中</td> </tr> <tr> <td>4</td> <td>印列工作被取消</td> </tr> <tr> <td>LW-</td> <td>錯誤</td> </tr> </tbody> </table>	LW-n	狀態	0	印表機的驅動程式尚未安裝	1	正在安裝印表機的驅動程式	2	印表機已就緒	3	印表機正在列印中	4	印列工作被取消	LW-	錯誤
LW-n	狀態														
0	印表機的驅動程式尚未安裝														
1	正在安裝印表機的驅動程式														
2	印表機已就緒														
3	印表機正在列印中														
4	印列工作被取消														
LW-	錯誤														

n+1	
0	無
1	找不到印表機
2	未知的錯誤

**控制位址**

設定印表機的連線參數。

LW-n	命令
0	無
1	更新印表機連線參數
2	重啟印表機驅動
LW-n+1	介面類型
0	乙太網路
1	USB
LW-n+2	IP 位址 (共 4 個字組)
LW-n+6	連接埠 (預設為 9100)

注意：若使用他牌印表機，即便有 ppd 檔案，仍可能因為沒有對應的驅動而無法列印。請以實際測試結果為準。

**捲動軸**

可設定捲動軸的尺寸。當物件的尺寸不足以顯示其內容時，捲動軸將顯示於該物件上。此功能可被套用至 [報警顯示]、[事件顯示]、[歷史數據控制]、[項目選單] 等可以上下捲動的物件。

**穿透 (虛擬序  
列埠)**

可自訂網路穿透通訊時使用的連接埠。

## 5.4. 一般屬性

[一般屬性] 設定頁用來設定與畫面操作有關的各項屬性。

eMT、iE、XE、mTV 系列



## 系統參數設定

## cMT、cMT X 系列



設定	描述
其他選項	<p><b>快速換頁顯示模式</b></p> <p>在此模式下，換頁後物件會先使用上一次正確讀取的數據，預先顯示物件的內容，以達到較快的換頁顯示速度。</p> <p>可控物件：可輸入變數的物件，如切換開關、數值輸入等。</p> <p>不可控物件：僅顯示狀態的物件，如位元狀態指示燈、多狀態指示燈等。</p> <p>支援快速顯示模式的物件包含：位元狀態指示燈、多狀態指示燈、字元、數值、多功能按鈕、項目選單、直接/間接視</p>

窗、棒圖、錶針、動畫、移動圖形、PDF 檢視器、圖片檢視、能源需量顯示等物件。

#### 啟用 RW\_A 暫存器

可勾選是否啟用配方資料 RW\_A。在啟用 RW\_A 後，物件才可以操作 RW\_A 的內容。RW\_A 的大小為 64K。

#### 若觸摸範圍超出按鈕，仍觸發釋放動作(cMT / cMT X 系列功能)

當勾選此項功能後，按下按鈕，手指仍壓在觸控螢幕上，但滑至螢幕超出按鈕範圍的地方才放開，仍被視為放開該按鈕，並會觸發按鈕釋放動作。

#### 當離開輸入程序時設定數值 (cMT / cMT X 系列功能)

勾選此選項後，在輸入物件(例如數值物件)輸入完數值後，不須再按 Enter 鍵，即可輸入。例如：當進入輸入程序後，再點選其他輸入物件，原先的輸入程序將仍視為輸入完成，即會設定數值。若輸入內容不合法，則會跳出警示訊息。



1. 透過 ESC 離開物件，數值不會保持。
2. 若使用系統鍵盤，當關閉鍵盤時，數值也會直接儲存。
3. 若有使用 [輸入次序]，使用 ENTER 離開該物件時，[輸入次序] 才會正常運作。若中途點選了其他物件，則 [輸入次序] 會停止運作。

#### CSV 分隔符號 (cMT / cMT X 系列功能)

對備份與匯入/匯出的檔案設定分隔符號為逗號(,)或分號(:)。

### 視窗

#### 初始視窗編號

選擇 HMI 開機後的起始頁面。

#### 公共視窗屬性

公共視窗 (4 號視窗) 內的物件會出現在每個基本視窗中，此選項可設定公共視窗內的物件，將出現在基本視窗物件的上層或下層。

#### 物件動畫

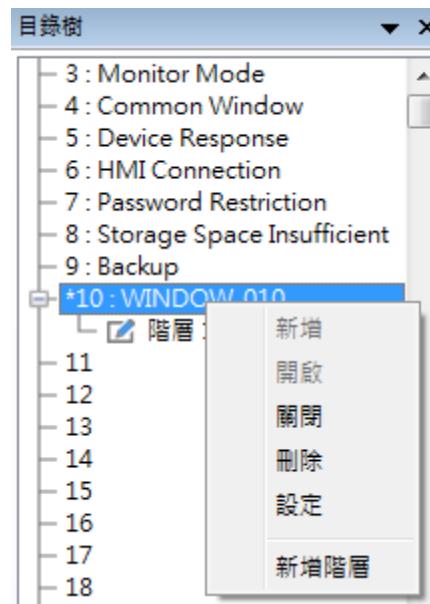
如果選擇 [保持] 模式，則 HMI 在運行時，[動畫] 與 [移動圖形] 物件將顯示在其他類型物件的上方，與物件的建立順序無關。如果選擇 [不保持] 模式，則物件的顯示順序依照物件建立的次序先後，先建立者先顯示。

#### 啟用視窗階層(可用於組織化/顯示/隱藏物件)

當勾選此項功能後，在目錄樹下，同視窗可以產生不同階

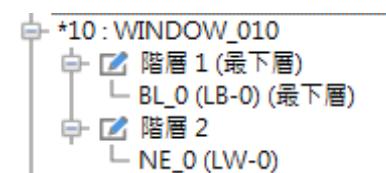
層。一個視窗最多可分為 10 個階層。

如下圖，在視窗編號上按右鍵，選擇[新增階層]。

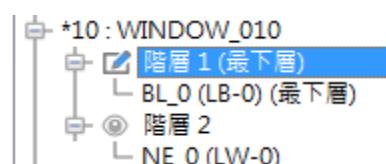


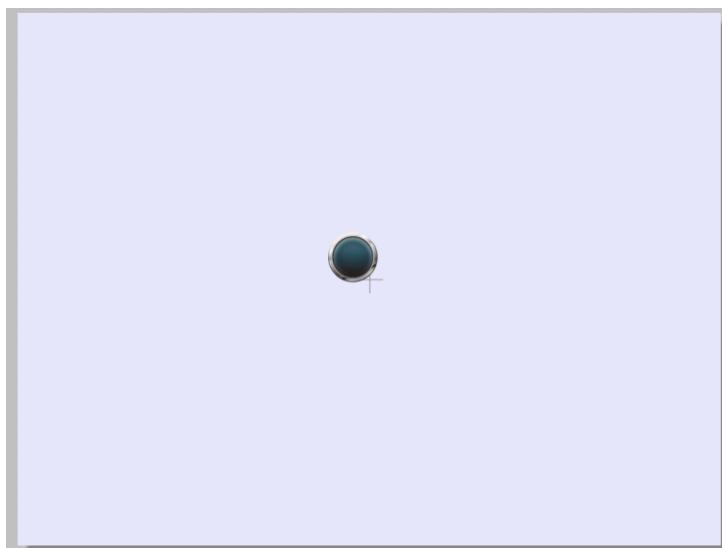
以下為使用範例：

在階層一放置一個位元燈物件，階層二放置一個數值物件。

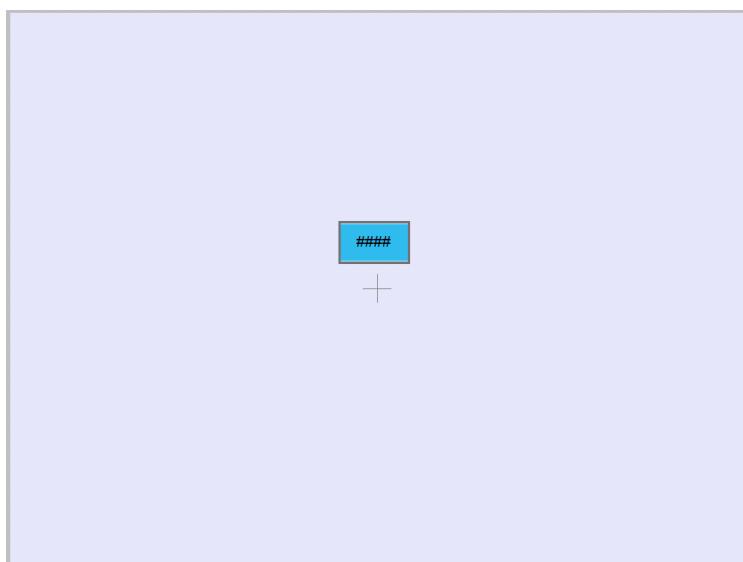
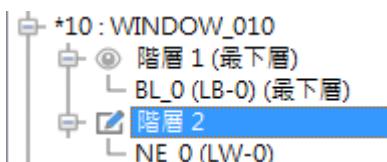


點選階層一，EasyBuilder Pro 編輯主畫面只會顯示位元燈物件。

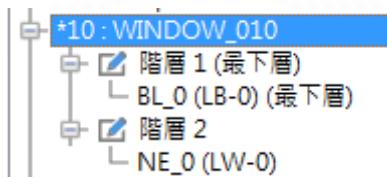


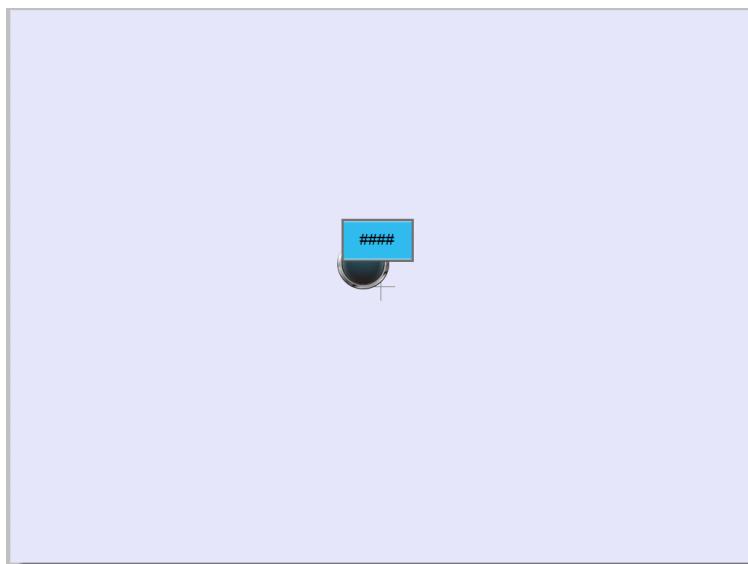


點選階層二，EasyBuilder Pro 編輯主畫面只會顯示數值物件。



點選視窗 10，則會顯示所有物件。





### 螢幕保護設定

#### 背光節能時間

當未碰觸螢幕的持續時間等於此設定值時，將關閉背光燈，設定的時間單位為分鐘。關閉背光燈後只需碰觸到螢幕，即可重開背光燈。當設定值選擇 [無] 時，HMI 將不使用背光節能的功能。

#### 螢幕保護時間

當未碰觸螢幕的持續時間等於此設定值時，將自動切換到 [螢幕保護視窗] 所指定的視窗，設定的時間單位為分鐘。當設定值選擇 [無] 時，HMI 將不使用螢幕保護的功能。

#### 螢幕保護視窗

指定螢幕保護功能啟動時所顯示的頁面。

注意：因螢幕保護視窗採用功能鍵之“返回上一個視窗”作為跳離螢幕保護視窗模式，若工程檔案規劃如下案例，恐會造成錯誤。

案例：螢幕保護視窗 -> 某視窗 [返回上一個視窗]-> 螢幕保護視窗

說明：假設視窗 11 上有一“返回上一個視窗”按鍵，此用來返回上一個視窗，但若在視窗 11 時進入螢幕保護視窗模式，於螢幕保護視窗點擊任一處跳離螢幕保護視窗模式後回到視窗 11，此時若再度點選視窗 11 上的“返回上一個視窗”按鍵，將再度回到螢幕保護視窗。

解決：1. 螢幕保護視窗中若使用“返回上一個視窗”，其他頁面請一律採用“切換基本視窗”。2. 螢幕保護視窗中使用“切換基本視窗”至固定頁面（例如首頁），並確認此頁面沒有使用“返回上一個視窗”，其他頁面仍可以使用“返回上一個視

窗”。

#### 快選視窗按鍵

##### 設定

僅適用 eMT、iE、XE、mTV、iP 系列。

設定快選視窗 (3 號視窗) 的各項屬性，若要使用快選視窗前，需先建立 3 號視窗。

##### 屬性

選擇是否使用快選視窗，在選擇 [啟用] 後，可以點選 [設定] 功能，設定快選視窗按鍵的各項屬性，包括標示於快選視窗按鍵上的圖片及文字。

##### 位置

選擇快選視窗按鈕的出現位置，選擇 [左] 則快選視窗按鈕出現在畫面的左下角；選擇 [右] 則快選視窗按鈕出現在畫面的右下角。

##### 設定

設定快選視窗的圖片及文字屬性。

##### HMI 啟動時隱藏按鍵

若勾選，當 HMI 啓動時將隱藏快選視窗按鈕。若要呼叫快選視窗，必須透過系統暫存器 LB-9013~LB-9015。

#### 監控模式

僅適用 cMT、cMT X 系列。

啟用後，cMT Viewer 的工程檔案預覽畫面會改為顯示工程檔案中 3 號視窗的畫面，且可即時監看 3 號視窗的數據變化。提供 3x3 及 5x4 兩種方格顯示方式，最多可顯示 50 台 HMI 的畫面。

#### 事件

##### 使用 LW9450~9455 作為事件記錄的時間標籤

若勾選，事件登錄將使用以下標籤的時間作為事件觸發的時間。

[LW-9450: 事件登錄的時間標籤-秒] (限制範圍: 0 ~ 59)

[LW-9451: 事件登錄的時間標籤-分] (限制範圍: 0 ~ 59)

[LW-9452: 事件登錄的時間標籤-時] (限制範圍: 0 ~ 23)

[LW-9453: 事件登錄的時間標籤-日] (限制範圍: 1 ~ 31)

[LW-9454: 事件登錄的時間標籤-月] (限制範圍: 1 ~ 12)

[LW-9455: 事件登錄的時間標籤-年] (限制範圍: 1970 ~ 2037)

注意：若 LW-9450 ~ LW-9455 的時間有其一數值超出限制範圍，系統將無法採用此功能作為事件記錄的時間標籤。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 鍵盤

顯示作為鍵盤視窗的列表，這些視窗代表在使用 [數值輸入]

與【字元輸入】物件時，可以選擇的鍵盤類型，最多可以新增到 32 個鍵盤。使用者要建立自訂的鍵盤時，需先在已存在的視窗規劃好要使用的鍵盤，並使用【新增】功能選擇這些視窗並加入到列表中即可。

 詳細資訊請參考《12 鍵盤的設計與使用》。

#### 外部鍵盤編排模式

支援 QWERTY、AZERTY 與 Cyrillic 的 USB 鍵盤，HMI 上可使用 LW-9199 切換鍵盤模式。

#### 游標顏色及選擇顏色

僅適用 eMT、iE、XE、mTV、iP 系列。

設定使用【數值輸入】或【字元輸入】物件時，在輸入欄位中出現的游標顏色及選擇顏色。

---

#### 工程檔案保護

使用者的工程檔可被限定在特定的 HMI 上執行。

 詳細資訊請參考《30 工程檔案保護功能》。

---

#### 更新 HMI 上 的開機畫面

僅適用 cMT、cMT X 系列。

#### 預設

如果 HMI 上已有設定的開機畫面，選擇預設將會把該開機畫面清除。

#### 自定義

使用者可以自行選擇圖片作為開機畫面。支援格式:**bmp**

---

## 5.5. 系統設定

[系統設定] 控制 EasyBuilder Pro 中各種功能。

## 系統參數設定



有些功能從系統暫存器複製而來，例如：[隱藏系統設定列 (LB-9020)]、[隱藏滑鼠遊標 (LB-9018)]、[取消聲音輸出 (LB-9019)] 及 [取消上傳功能 (LB-9033)]，使用者亦可選擇系統暫存器來使用這些功能。

要選擇系統暫存器，使用者可在新增物件時，於設定頁勾選 [系統暫存器] 並選擇 [位址類型]。要檢視所有的系統暫存器，可於 **EasyBuilder Pro** 點選 [工程檔案] » [位址標籤] » [系統暫存器]。

設定	描述
下載工程檔案後所顯示的語言	選擇在重新下載工程檔案並啟動 HMI 時所顯示的語言。
HMI 啟動後開始與設備通訊的延遲時間	部分 PLC 裝置的開機速度較慢，為避免 HMI 一開始在通訊時，因 PLC 裝置的開機速度較慢產生通訊錯誤警 示，可使用此功能延遲 HMI 啟動後開始與裝置通訊的

時間。

**當 HMI 啟動後包含延遲初始化巨集指令的執行**

啟用後，巨集中設定為 [當 HMI 啟動時即執行一次] 的巨集也會延遲執行。

#### 開機後使用初始化巨集指令

指定當 HMI 開機後所執行的巨集指令。

#### 自動登出

如果閒置 HMI 的時間超過此設定值，HMI 中有設定安全等級的物件將無法使用，需要再次輸入使用者 ID 及密碼才能操作物件。

#### 隱藏系統設定列

##### /Start button

將 HMI 的系統設定頁關閉。

#### 隱藏滑鼠游標

將 HMI 上滑鼠游標關閉。

#### 滑鼠游標尺寸

可調整滑鼠游標尺寸。

#### 聲音控制

**當觸碰物件時即發出聲音：**當物件啟用聲音時，觸控按鍵即發出聲音。

**當物件動作時才發出聲音：**當物件啟用聲音並啟用 [最少按鍵時間] 時，碰觸物件與物件動作之間有時間差，物件動作時才發出聲音。

**取消聲音輸出：**將 HMI 聲音關閉。(不包含系統設定按鍵的聲音)

**停止上一個播放後再播放下一個聲音：**此功能為 cMT / cMT X 系列專用。當聲音尚未播放完成時就觸發下一個聲音，將會停止當前聲音並開始播放下一個聲音。

#### 取消上傳功能 (重新啟動 HMI 後生效)(或設 LB9033 為 ON)

將 HMI 上傳工程檔案的功能關閉。

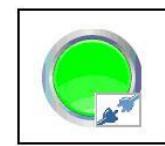
#### 當與設備通訊失敗時，顯示斷線圖示在相關的物件上

選擇當與設備通訊失敗時，是否使用斷線符號標示在相關物件上。



當使用此項功能且物件無法與設備通訊時，斷線符號會顯示在物件右下角。

斷線圖示只會出現在曾經與設備成功通訊過但後來通訊失敗的物件上。



#### 啟動看門狗

當 HMI 超過指定的時間仍無法正常運作，會自動重啟系統。

#### 啟用字型預先載入 (可優化性能，但

此功能為 cMT / cMT X 系列專用。勾選此選項後，針對較多文字的頁面可優化換頁速度，但 HMI 啟動可能會

<b>HMI 啟動速度可能 較慢)</b>	延遲幾秒。
<b>啟用計時器來顯示最 少按鍵時間</b>	此功能為 cMT / cMT X 系列專用。勾選此選項後，使用最少按鍵時間設定的物件，觸發時將會有倒數計時提醒使用者。
<b>自適應 DPI 縮放</b>	預設啟用此功能，系統會根據連接的裝置的螢幕像素調整文字大小與位置、Y 軸的間隔等。 所適用的物件包含趨勢圖、圓盤曲線圖、郵件聯絡人編輯器、項目選單、媒體播放器、報警顯示及事件顯示。 若取消勾選，則基於長度的設定可能無法正確地在不同的螢幕上反映。

## 5.6. 遠端

此頁籤用來設定遠端連線時的相關功能，部分功能與系統暫存器相同。



設定	描述
禁止遠端 HMI 連接	可禁止遠端 HMI 或 cMT viewer 連接至本機，LB-9044 亦可 用來控制遠端連接。
禁止遠端讀取密碼操作 (或設 LB9053 為 ON)	禁止遠端 HMI 讀取本機 HMI 的密碼，包含工程檔案與使用者 密碼等資料。

<b>禁止遠端寫入 密碼操作 (或 設 LB9054 為 ON)</b>	禁止遠端 HMI 寫入本機 HMI 的密碼，包含工程檔案與使用者密碼等資料。
<b>VNC 伺服器</b>	<p><b>不須密碼即可登入：</b>當透過 VNC 連線至 HMI 時，無需輸入密碼即可連結，LB-12089 亦可用來控制輸入密碼與否。</p> <p><b>監視模式：</b>當透過 VNC 連線至 HMI 時，將只能監看而無法操作，LB-12088 亦可用來控制監視模式。</p> <p><b>從工程檔案中設定密碼：</b>設定登入 VNC 伺服器所使用的密碼。</p>
<b>EasyAccess 伺 服器</b>	<p><b>EasyAccess 2.0 伺服器位置</b></p> <p>可選擇全球或中國地區。</p> <p> 請點選此圖示觀看教學影片，請先確定已連上網路線。</p>
<b>診斷器</b>	<p><b>啟用：</b>若啟用，當透過 cMT Viewer 連接到 cMT / cMT X 時，可開啟診斷器功能，LB-12656 亦可用來控制診斷器功能。</p> <p><b>不須密碼即可登入：</b>當欲登入診斷器時，無需輸入密碼即可連結，LB-12657 亦可用來控制輸入密碼與否。</p> <p><b>密碼：</b>設定登入診斷器的密碼，LW-11756 (4 words) 亦可用來設定密碼。</p>
<b>cMT Viewer</b>	<p><b>最大連線數量</b></p> <p>設定 cMT / cMT X 系列可允許被 cMT Viewer 連線之數量。</p>

## 5.7. 使用者密碼

[使用者密碼] 設定頁用來設定使用者的密碼以及可操控的物件類別。共有兩種認證模式：一般模式和進階安全模式。

 詳細資訊請參考《10 物件安全防護》。

### 5.7.1. 一般模式



可設定 12 組使用者密碼，且密碼須為非負整數。

HMI 運行時，使用者在成功輸入密碼後，系統會依照使用者的設定內容決定使用者可以操作的物件類別。物件的類別被區分為 [類別 A] 至 [類別 F] 共 6 種。

類別屬於 [無] 的物件，開放給任何使用者使用。

當 “使用者 3” 的設定內容如上圖時，則此使用者只被允許使用類別屬於 “無” 與 A、B、C 的物件。

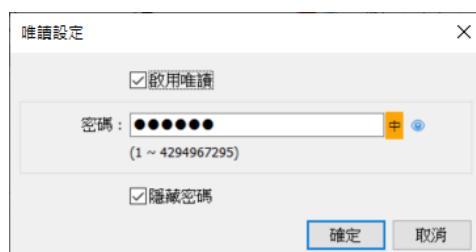
### 5.7.2. 進階安全模式

進階安全模式可規劃的使用者為 11 組，另外提供 [管理員] 使用模式，此管理員有最大使用權，任何物件的安全等級皆可操作。不同的使用者密碼可由英文字母或數字所組成，並可規劃每個使用者可操作的物件類別分為 [類別 A] 至 [類別 L] 等共 12 個類別。

## 系統參數設定

設定描述編輯

設定是否禁止使用者修改此頁面的密碼。



[啟用唯讀]：啟用後，使用者無法修改此頁面的設置。

[隱藏密碼]：啟用後，密碼欄位會以 \* 符號顯示。

設定使用者可

若勾選 [在 HMI 上使用現有的使用者帳號] 時，當下載工程

<b>以操作的物件 類別與密碼</b>	檔案至 HMI 時，原本 HMI 上的使用者帳號將不會被清除。
<b>管理員</b>	為內定管理員帳號，不可被刪除，且權限全開不得修改權限。  進階安全模式可搭配 [項目選單] 物件來顯示帳號名稱和權限。  若勾選 [隱藏使用者] 則帳號名稱和權限等資料不會顯示在 [項目選單] 物件上。
<b>控制位址</b>	進階安全模式提供一組 [控制位址] 供使用者登入和管理帳號。
<b>工程檔案密碼</b>	若有設此密碼，當使用者想要編輯工程檔案時，必須輸入此密碼才能編輯。勾選 [啟用] 再按下 [設定] 就可以設定密碼。完成設定後，在編輯工程檔案前，將要求使用者輸入密碼，輸入正確才能進入工程檔案。
<b>當 USB 安全 金鑰插入 HMI 時自動執行登 入/登出</b>	勾選後可使用 USB 安全金鑰自動登入/登出，而登入/登出的狀態會自動寫入至指定的 [狀態位址]。當 USB 碟插入時，將執行自動登入，當 USB 碟拔出時，將自動執行登出。狀態位址的數值意義為：0x00: 無動作, 0x01: 登入成功, 0x04: 登入失敗, 0x08: 登出成功, 0x10: 登出失敗。

## 5.8. 擴展記憶體

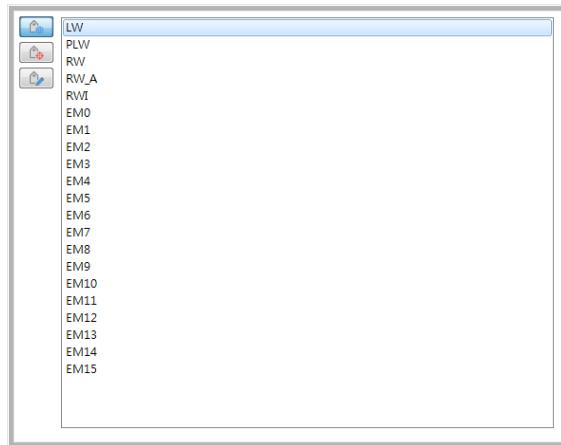
[擴展記憶體] 設定頁用來設定擴展記憶體的儲存位置。

## 系統參數設定

系統參數設定

	設備	HMI 屬性	一般屬性	系統	遠端	使用者密碼	擴展記憶體	行動網路	時間同步/夏令時間	郵件	FTP
EM0	檔案名稱: em0.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM1	檔案名稱: em1.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM2	檔案名稱: em2.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM3	檔案名稱: em3.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM4	檔案名稱: em4.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM5	檔案名稱: em5.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM6	檔案名稱: em6.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM7	檔案名稱: em7.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM8	檔案名稱: em8.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM9	檔案名稱: em9.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM10	檔案名稱: em10.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM11	檔案名稱: em11.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM12	檔案名稱: em12.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM13	檔案名稱: em13.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM14	檔案名稱: em14.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
EM15	檔案名稱: em15.emi	<input checked="" type="radio"/> USB 磁 1	<input type="radio"/> USB 磁 2								
<input type="checkbox"/> 啟用動態檔案名稱 (不包含副檔名)											
<input type="button" value="確定"/> <input type="button" value="取消"/> <input type="button" value="說明"/>											

擴展記憶體包含 EM0 ~ EM15，使用方式與其他 HMI 上的設備類型相似 (類似使用 LW 或 RW 位址類型)，使用者只需在建立物件時自 [位址] 中指定使用 EM0 ~ EM15 即可，每個擴展記憶體最多可以存放 2G word 的資料。



擴展記憶體中的數據使用檔案的形式存放在 [SD 卡]、[USB 碟] 上，[EM0] ~ [EM15] 所使用的檔案名稱預設分別為 em0.emi ~ em15.emi，使用者可以自行更改其檔案名稱，並使用 RecipeEditor.exe 開啟這些檔案並編輯擴展記憶體中的數據。

擴展記憶體中的數據不會因 HMI 斷電而消失，也就是下次開機後，擴展記憶體中的數據會恢復為關機前的狀態，與配方數據 (RW、RW\_A) 類似。

當做為擴展記憶體的裝置不存在時，若讀取擴展記憶體中的數據，內容將一律為 “0”；當做為擴展記憶體的裝置不存在時，若要將數據寫到擴展記憶體中，HMI 將會顯示 “Device no response” 訊息。

HMI 在機器不需斷電的情況下，可以隨時插上或移除外接裝置。使用者可以利用這項特性，更新或擷取擴展記憶體中的數據。

設定	描述
啟用動態檔案 名稱	若啟用，EM0 ~ EM15 的檔案名稱將可動態更改，先藉由“EM 檔案名稱索引(0~15)”挑選欲修改的 EM 編號，並輸入“EM 檔案名稱”，並設定“控制”為 1，即可查看“結果”。成功執行後，EM 位址將立即使用新的檔案名稱作為資料來源。 注意：當動態修改 EM 檔案名稱後，若重新啟動 HMI，EM 檔案資料來源將以最後修改的檔案名稱為主，而非工程檔案內的設定。

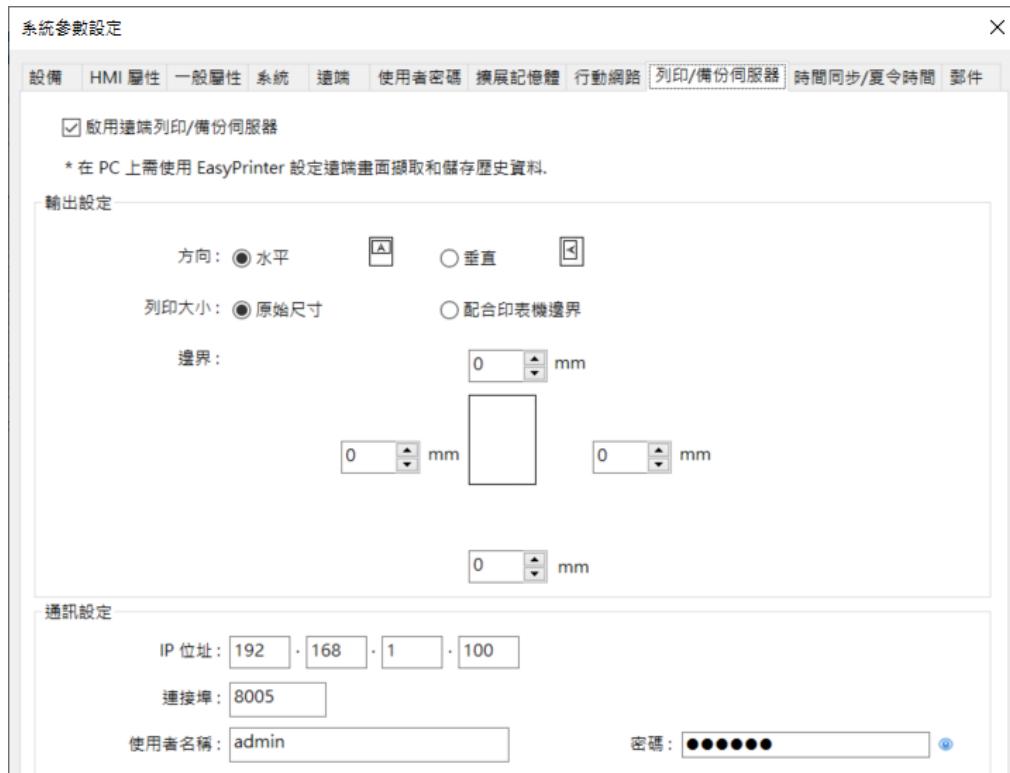
## 系統參數設定



請點選此圖示觀看教學影片，請先確定已連上網路線。

## 5.9. 列印/備份伺服器

此設定頁用來設定遠端列印/備份伺服器的相關設定。此功能不支援 cMT / cMT X 系列。



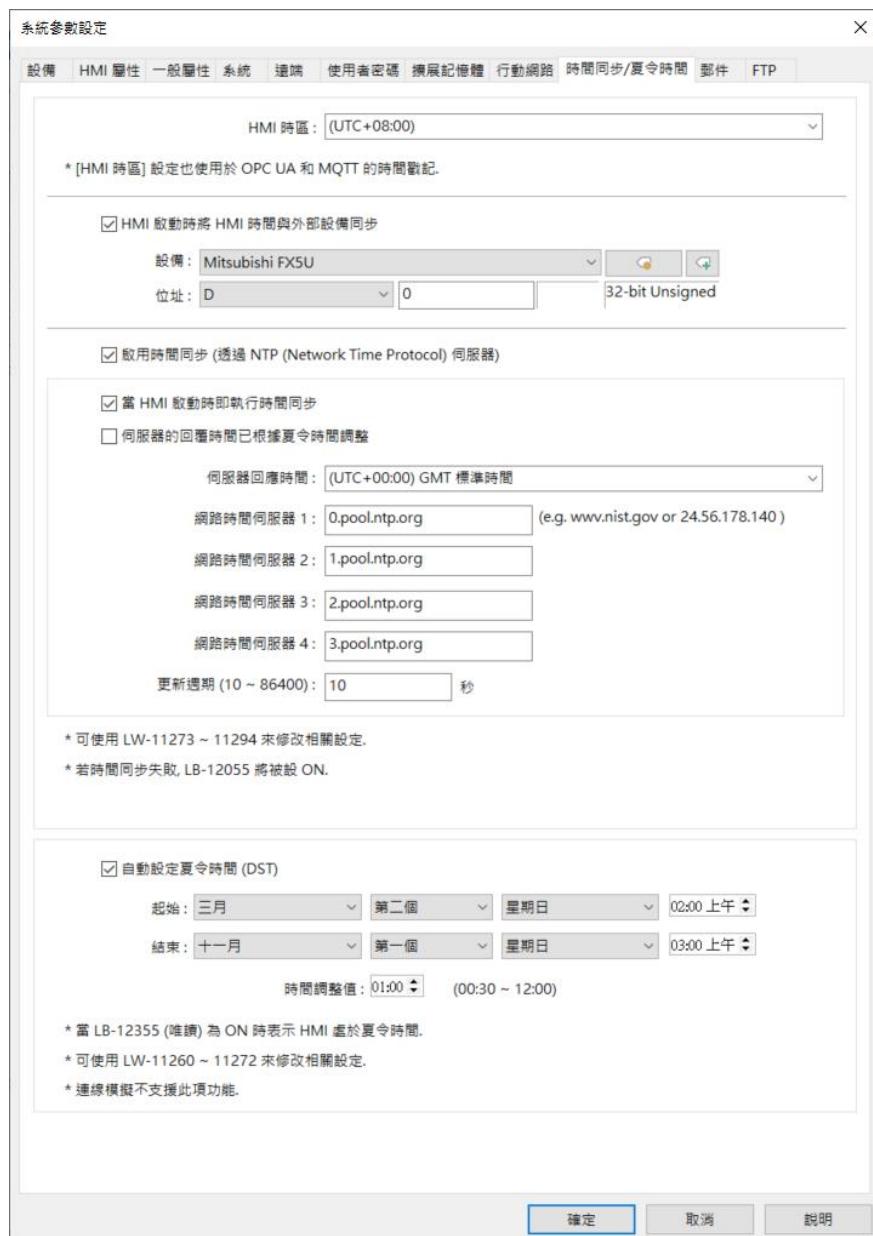
設定	描述
輸出設定	<b>方向</b> 設定圖文要 [水平] 或 [垂直] 列印。 <b>列印大小</b> 設定要照 [原始尺寸] 或 [配合印表機邊界] 列印。 <b>邊界</b> 設定上下左右的邊界寬度。
通訊設定	<b>IP 位址</b> 透過網路指定印表機的 IP 位址。 <b>連接埠、使用者名稱、密碼</b> 指定登入伺服器的資訊。 連接埠可設定為 1 ~ 65535。 使用者名稱或密碼最長可為 12 個字。

詳細資訊請參考《26 EasyPrinter》。

請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 5.10. 時間同步/夏令時間

此設定頁用來設定時間同步與夏令時間。



設定	描述
<b>HMI 時區</b>	設定 HMI 的時區。
<b>HMI 啟動時將 HMI 時間與外部設備同步</b>	當 HMI 啓動時，會先自動與指定的來源位址同步時間。以上圖所示的設定為例，表示時間來自設備“Mitsubishi FX5U”的 D 位址，此時位址 D-0 開始的連續 6 個字組暫存器分別存放下列資訊： D-0 -> 秒 (限制範圍： 0~59)

- 
- D-1 -> 分 (限制範圍： 0~59)  
D-2 -> 時 (限制範圍： 0~23)  
D-3 -> 日 (限制範圍： 1~31)  
D-4 -> 月 (限制範圍： 1~12)  
D-5 -> 年 (限制範圍： 1970~2037)
- 

**啟用時間同步****當 HMI 啟動時即執行時間同步**

當 HMI 啟動時，會先自動與指定的 NTP 伺服器同步時間。

**伺服器回應時間**

設定 NTP 伺服器的時區。

**網路時間伺服器**

提供四個網路時間伺服器欄位供使用者設置。當第一個伺服器同步失敗時，會嘗試與第二個伺服器同步。依此類推依序往下嘗試同步。當 HMI 無法與任一伺服器同步時，系統暫存器 LB-12055 會設為 ON。

**更新週期**

與 NTP 伺服器同步時間的頻率。頻率範圍為 10 ~ 86400 秒。

**自動設定夏令****起始 / 結束****時間 (DST)**

設定夏令時間的起始與結束時間。其中週的欄位，[最後的]

代表當月最後一週，可能為當月的第五週或第六週。

**時間調整值**

設定時間的偏移量。

**Note**

- 當 [網路時間伺服器] 設定為靜態 IP 地址時，可以通過 LW-11278~11293 位址進行寫入/控制操作。如果 [網路時間伺服器] 設定為域名，則不支援通過 LW-11278~11293 位址修改網路時間伺服器的 IP 地址。
- 當夏令時間結束時，時間會向前調整並開始時間重複的過渡時期，此時趨勢圖物件之即時模式會停止更新畫面，但歷史數據不受影響。
- 當夏令時間結束時，即使 HMI 時間被外部調整（手動修改或從網路校時等）回夏令時間內，系統仍不會啟動夏令時間，LB-12355 仍會保持為 OFF。
- 當夏令時間結束時，HMI 時間看似回到夏令時間內，系統仍不會啟動夏令時間，LB-12355 仍會保持為 OFF。
- 當尚未進入夏令時間前，外部調整 HMI 時間至夏令時間內，則視為進入夏令時間，LB-12355 會顯示為 ON。此時 HMI 將以外部調整的時間為準，不會將時間再加上 [時間調整值]。
- 當已進入夏令時間後，外部調整 HMI 時間至夏令時間外，則視為結束夏令時間，LB-12355 會顯示為 OFF。此時 HMI 將以外部調整時間為準，不會將時間再減去 [時間調整值]。
- 此功能尚不支援南半球國家的夏令時間。



夏令時間的相關暫存器請參考《22 位址暫存器》。

## 5.11. 郵件

[郵件] 設定頁用來設定 e-Mail 的相關資料。

若勾選 [在 HMI 上使用現有的連絡人設定]，當下載工程檔案至 HMI 時，將優先使用原本 HMI 上的連絡人設定，若不存在，則使用以下設定。



設定	描述
<b>SMTP 設定</b>	伺服器：設定郵件伺服器。 連接埠：設定連接埠。 使用者名稱：設定電子郵件地址。 密碼：設定郵件密碼。

**密碼確認：**確認設定的郵件密碼。

**SMTP 伺服器需要認證：**使用安全密碼驗證才能登入。

**使用下列加密類型連結：**寄出時，選擇是否需要加密連線 (TLS、SSL)。TLS 僅支援 TLS 1.0。

#### 寄件者資訊

##### 名稱

可選擇使用“設備清單”中的本機 HMI 名稱、系統暫存器 LW-10884 (HMI 名稱)或自行輸入寄件者名稱。

##### 郵件地址

設定電子郵件地址。

#### 錯誤訊息

郵件發送異常時，將 SMTP 伺服器回傳的錯誤訊息顯示在指定的暫存器位址。

👉 郵件寄送的相關暫存器請參考《22 位址暫存器》。

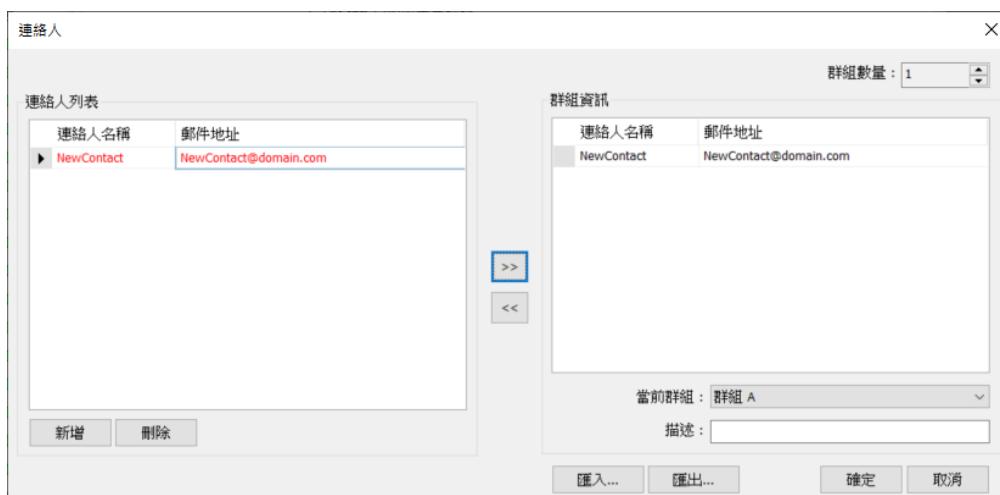
#### 錯誤通知

啟用後，在寄送郵件前，會先將 LB-12053 或 LB-12054 設為 OFF。否則，當有郵件寄送失敗，則 LB-12053 或 LB-12054 會一直保持為 ON。

#### 測試 SMTP 設定

點選 [測試] 後，EasyBuilder Pro 將自動發出一封郵件至指定的收件者群組，以驗證 SMTP 伺服器的設定是否正確。

按下 [連絡人] 後：



#### 設定

#### 描述

##### 連絡人列表

新增或刪除連絡人清單。

##### 群組資訊

將多個連絡人設定成群組。

##### 群組數量

設定群組數量，群組名稱依數量，依序命名為群組 A ~ P，最多共可設定 16 個群組。

**當前群組**

顯示目前清單中的連絡人所屬群組。

**描述**

使用者可以輸入關於該群組的描述。



傳送事件記錄的詳細資訊請參考《7 事件登錄》。



郵件發送相關的系統暫存器資訊請參考《22.3.40 位址暫存器》。



請點選此圖示觀看教學影片，請先確定已連上網路線。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 5.12. 行動網路

[行動網路] 設定頁用來設定行動網路的相關內容。

支援的機型：

cMT X 系列

cMT-SVR (OS 版本須為 20151127 或更新的版本)

cMT3071/cMT3072/cMT3090/cMT3103/cMT3151 (OS 版本須為 20180723 或更新的版本)

cMT-HDM/cMT-FHD/cMT-FHDX (OS 版本須為 20190130 或更新的版本)

MT8051iP/MT8071iP/MT8102iP (OS 版本須為 20201007 或更新的版本)

iE 系列 (OS 版本須為 20231215 或更新的版本)

將 3G/4G 網卡插入 USB 埠即可連線到網際網路。

**設定****描述**

**在 HMI 上使用現有的設定**

勾選後，行動網路將使用 HMI 現有的參數。

備註：除了特殊的網卡之外，一般的行動網路的 [個人識別碼 (PIN)] 為 0000，[存取點名稱 (APN)] 為 internet，不需設定 [使用者名稱]、[密碼]、[電話號碼]。

相關的系統暫存器：

- LW-11297: SIM 卡的個人識別碼 PIN (行動網路)
- LW-11313: 存取點名稱 APN (行動網路)
- LW-11329: 使用者名稱 (行動網路)
- LW-11345: 密碼 (行動網路)
- LW-11361: 電話號碼 (行動網路)
- LW-11377: 停止 (設 0)/啟動 (設 1) 連接 (行動網路)
- LW-11378: 最後錯誤碼 (0:成功, 1:錯誤的 PIN 碼, 2:無 SIM 卡, 3:無設備, 4:puk 碼已鎖住, 5:其他) (行動網路)
- LW-11379: 連接狀態 (0:無設備, 1:斷線, 2:連線中, 3:已連線) (行動網路)

### USB 數據連線

透過 Android® 手機的 USB 數據連線功能，讓 HMI 可連線到網際網路。若欲啟動 USB 數據連線功能，只需透過 micro USB 線或者任何可連接手機的 USB 傳輸線，連接 HMI 與 Android 手機，並開啟 Android 手機的 USB 數據連線，如下圖的範例所示。



並透過以下系統暫存器監控狀態：

- LW-11380: 停止 (設 0)/啟動 (設 1) 連接 (USB 數據連線)
- LW-11381: 連接狀態 (0:無設備, 1:斷線, 2:已連線, 3:失敗, 4:OS 不支援, 5:HMI 不支援) (USB 數據連線)



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 5.13. FTP



### 5.13.1. FTP 伺服器

在 OS 版本 20240308 之後的 cMT / cMT X 系列 HMI 開始支援顯式 FTPS。用戶可以在此設定 HMI FTP 伺服器的憑證，具有以下選項：

1. 保持 HMI 設定：繼續使用 HMI 當前的憑證設定。
2. 預設憑證：使用系統提供的預設憑證。
3. 自訂憑證：用戶可以匯入自己的憑證以及私鑰，以滿足特定的安全需求。

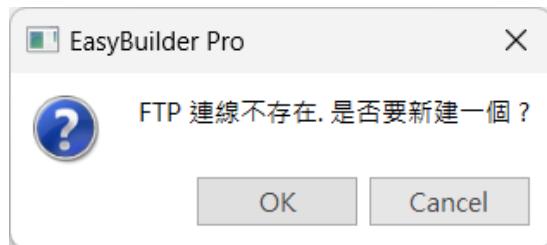
### 5.13.2. FTP 客戶端

FTP 客戶端功能允許用戶進行畫面擷取和歷史資料的備份。啟用此功能後，用戶可以將檔案從 HMI 備份到 FTP 伺服器。此功能類似於 eMT/iE/XE/mTV 系列通過 EasyPrinter 進行歷史資料備份。此功能僅 cMT / cMT X 系列支援。

設定	描述
啟用 FTP 功能	啟用後，可透過 FTP 功能將擷取畫面或歷史資料上傳到 FTP 伺服器。
FTP 連線	設定備份資料的 FTP 伺服器資訊。
HMI 資料夾命名規則	目錄名稱的命名方式。可選擇使用 IP、備份資料夾或使用 HMI 名稱作為資料夾名稱。
字首:	置於文件名稱字首的文字。

 Note

- 當啟用 [FTP 客戶端] 功能但未設定任何 FTP 連線時，系統將引導用戶至 [資料/歷史] » [FTP 連線] 物件，以建立 FTP 伺服器連線參數。



# 6. 視窗

本章節說明視窗的種類，以及如何建立與使用視窗。

6.1.	概要 .....	6-2
6.2.	視窗類型 .....	6-2
6.3.	視窗的建立、設定與刪除 .....	6-4
6.4.	視窗透明度 .....	6-7
6.5.	編輯視窗應用 .....	6-11

## 6.1. 概要

視窗是 EasyBuilder Pro 編輯軟體裡最重要的元素之一。所有需要顯示在 HMI 上的各種物件、圖形、文字等必須透過視窗才能呈現。EasyBuilder Pro 內建 1997 個視窗，其範圍為視窗 3 至視窗 1999。

## 6.2. 視窗類型

依照功能與使用方式的不同，可將視窗分為下列四種類型：

- 基本視窗
- 快選視窗
- 公共視窗
- 系統訊息視窗

### 6.2.1. 基本視窗

基本視窗是最常被使用的視窗類型，除了可當作主畫面的用途之外，也被用在：

- 底層畫面，可提供其他視窗作為背景畫面。
- 鍵盤視窗。
- 功能鍵物件所選用的彈出視窗。
- 間接視窗與直接視窗物件所選用的彈出視窗。
- 螢幕保護視窗畫面。



#### Note

- 由於基本視窗的尺寸大小必須與 HMI 顯示螢幕相同，所以其解析度設定也必須與所使用的 HMI 解析度一致。

### 6.2.2. 快選視窗

3 號視窗為預設的快選視窗，此種視窗可以與基本視窗同時存在，一般被用在置放常用的工作按鈕，位置為畫面的左下角或右下角。要使用快選視窗除了需先建立 3 號視窗外，需再設定快選視窗按鈕的各項屬性，快選視窗按鈕的各項設定在 [系統參數設定] » [一般屬性] 頁籤中。除了可以使用快選視窗按鈕來切換快選視窗的顯示與隱藏之外，系統暫存器也提供以下位址可用來控制快選視窗：

- [LB-9013] 隱藏 (設 ON)/顯示 (設 OFF) 快選視窗
- [LB-9014] 隱藏 (設 ON)/顯示 (設 OFF) 快選按鍵
- [LB-9015] 隱藏 (設 ON)/顯示 (設 OFF) 快選視窗/按鍵



#### Note

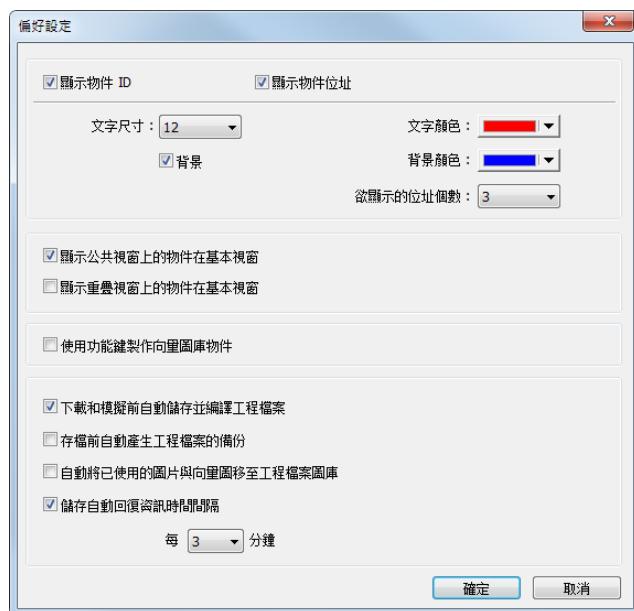
- cMT / cMT X 系列不支援快選視窗功能。

### 6.2.3. 公共視窗

4 號視窗為預設的公共視窗，此視窗中的物件也會出現在其他基本視窗中，但不包含彈出視窗，因此通常會將各視窗共用的物件放置在公共視窗中。

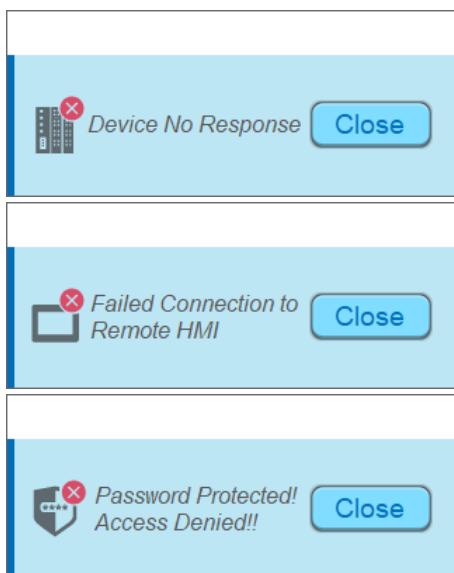
HMI 上的程式運行時，可以使用功能鍵物件的【切換公共視窗】模式，線上更改公共視窗的來源。

在【偏好設定】可設定當編輯程式時，公共視窗上的物件是否會被顯示於基本視窗。有了此預覽功能，可避免當編輯程式時，將基本視窗的物件重疊到公共視窗的物件。



### 6.2.4. 系統訊息視窗

5、6、7、8 號視窗為預設的系統提示訊息視窗：



#### 5 號視窗為 Device Response 視窗

當 HMI 與設備通訊中斷時，系統將自動彈出 Device no response 的警告視窗。可使用系統暫存器所提供的相關位址來禁止彈出此視窗。

#### 6 號視窗為 HMI Connection 視窗

當本地 HMI 無法連接到遠端的 HMI 時，系統將自動彈出 Failed connection to remote HMI 的警告視窗。

#### 7 號視窗為 Password Restriction 視窗

當使用者無權限操作某物件時，可依設定決定是否彈出 Password Protected! Access Denied! 的警告視窗。



### 8 號視窗為 Storage Space Insufficient 視窗

當 HMI 記憶體、USB 碟或 SD 卡上的可用空間不足以儲存新的資料時，系統將自動彈出 Storage Space Insufficient 的警告視窗。(當系統偵測到記憶體剩 4 MB 以下)

下列的系統暫存器可檢視 HMI、USB 碟或 SD 卡上目前可用的儲存空間：

[LW-9072] HMI 當前的剩餘空間 (K bytes)

[LW-9074] SD 卡當前的剩餘空間 (K bytes)

[LW-9076] USB 碟當前的剩餘空間 (K bytes)

並可透過下列的系統暫存器檢視儲存裝置空間是否足夠，若系統偵測到記憶體剩 4 MB 以下，會將相關位址設為 ON：

[LB-9035] HMI 剩餘空間不足警示 (當狀態為 ON)

[LB-9036] SD 卡剩餘空間不足警示 (當狀態為 ON)

[LB-9037] USB 碟剩餘空間不足警示 (當狀態為 ON)

 詳細資訊請參考《22 位址暫存器》。

視窗 5 ~ 視窗 8 的內容提示，可以根據實際需要來修改，方便操作人員容易讀懂和識別故障資訊。

#### Note

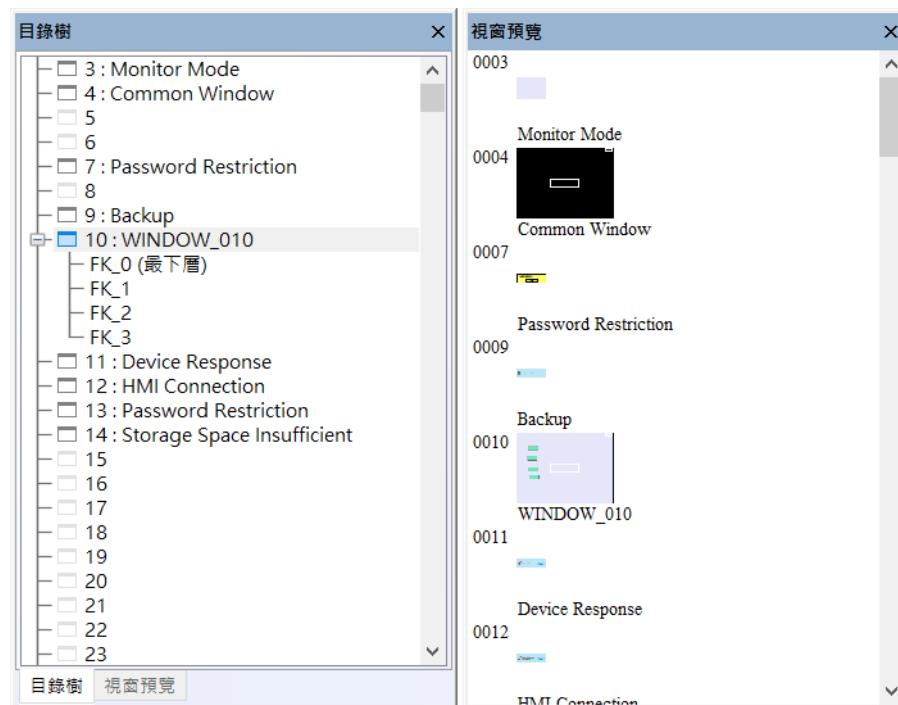
- eMT、iE、XE、mTV 系列最多可同時開啟 24 個彈出視窗，包含系統訊息視窗、直接視窗和間接視窗；cMT、cMT X 系列則無彈出視窗的數量上限。
- 系統不允許在一個基本視窗上使用 2 個直接(或間接) 視窗彈出同一個視窗(含重疊視窗)。
- 視窗 3 ~ 視窗 9 為系統內部使用，視窗 10 ~ 1999 為使用者可自行運用。

## 6.3. 視窗的建立、設定與刪除

可透過 [目錄樹] 查看已建立的視窗。

[目錄樹] 顯示視窗編號，已定義的視窗名稱將被顯示。目前被開啟編輯的視窗編號前會有個  符號，按下視窗編號旁的 (+) 號可看到視窗內使用的物件，包含物件 ID、位址與描述。

[視窗預覽] 用整體視窗外型的小圖來預覽視窗。



### 6.3.1. 視窗的建立與設定

在目錄樹下，選擇欲建立的視窗編號後按下滑鼠右鍵選擇【新增】。



設定	描述
視窗名稱	輸入的名稱將顯示在視窗的控制條，也會顯示在目錄樹中。
視窗編號	由 3 ~ 1999。
大小	設定視窗的大小，一般基本視窗的解析度與所選用的 HMI 的解析度一樣。
重疊視窗	重疊視窗功能可視為另一種附加的公共視窗，在設計程式時，同一物件可能被放置於許多的視窗中，但不是所有視窗時，便可使用重疊視窗。 每一個基本視窗最多可選擇三個視窗作為背景，從 [底層] 開始到 [頂層] 結束，這些背景視窗內的物件將在基本視窗中依序出現。
彈出視窗	基本視窗也可當作彈出視窗，[X] 與 [Y] 用來設定基本視窗在畫面彈跳出的座標位置。座標原點為畫面的左上角。
壟斷	若勾選，此視窗的顯示將獨佔所有的操作權。例如，當一個設定壟斷的視窗彈出，其他彈出視窗與背景視窗的操作將完全暫停，直到壟斷的視窗被關閉才可操作其他視窗。當基本視窗作為鍵盤視窗時，自動具有此屬性。
視窗控制條	設定系統訊息視窗彈出時是否包含視窗控制條屬性。 當設定的視窗為 5 ~ 8 時，才可設定此功能。
巨集指令	設定當視窗開啟或關閉時，及欲循環執行 (每 500ms 執行一次) 的巨集指令。 使用此功能前需先建立巨集指令。

### Note

- 在重疊視窗中的物件無法從顯示它們的基本視窗上編輯，若要編輯重疊視窗的物件，均需開啟該視窗編輯。
- 若基本視窗所使用的重疊視窗編號與欲彈出的視窗編號相同，將無法彈出該視窗。
- 當基本視窗與彈出視窗皆使用相同的重疊視窗時，彈出的視窗將無法顯示重疊視窗上的物件。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。  
或是在 [檢視] » [視窗列表] 選擇 [新增]，並選擇欲建立的視窗類型。



呼叫 [視窗設定] 對話窗有以下方式：

- 於目錄樹選擇視窗編號，按下滑鼠右鍵選擇 [設定]。
- 在 [檢視] » [視窗列表] 選擇欲設定的視窗後選擇 [設定]。
- 在該視窗中，未選擇任何物件時按下滑鼠右鍵選擇 [屬性]。

### 6.3.2. 視窗的開啟、關閉或刪除

開啟現有的視窗有以下方式：

- 雙擊點選目錄樹上的視窗編號。
- 在目錄樹上選擇欲開啟的視窗編號後，按下滑鼠右鍵選擇 [開啟]。
- 在 [檢視] » [視窗列表] 選擇欲開啟的視窗編號後選擇 [開啟]。

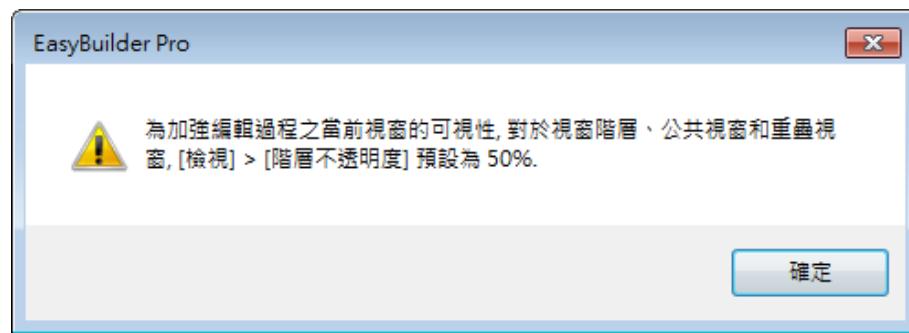
關閉或刪除現有的視窗有以下方式：

- 在目錄樹上選擇欲關閉或刪除的視窗編號後，按下滑鼠右鍵選擇 [關閉] 或 [刪除]。
- 在 [檢視] » [視窗列表] 選擇欲刪除的視窗編號後選擇 [刪除]。

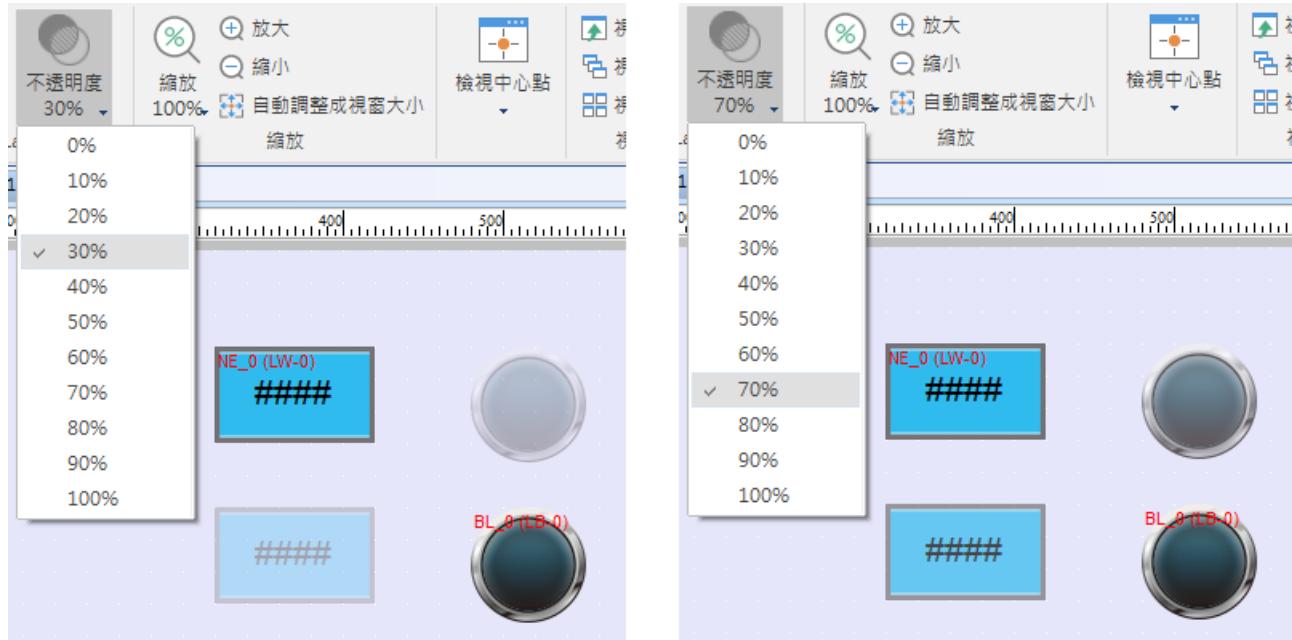
## 6.4. 視窗透明度

EasyBuilder Pro 編輯檔案時的視窗設定支援 [不透明度] 調整，可應用於視窗階層、公共視窗和重疊視窗。

## 視窗



如下圖所示，當 [不透明度] 的數值越大時，相關視窗的物件越清晰。



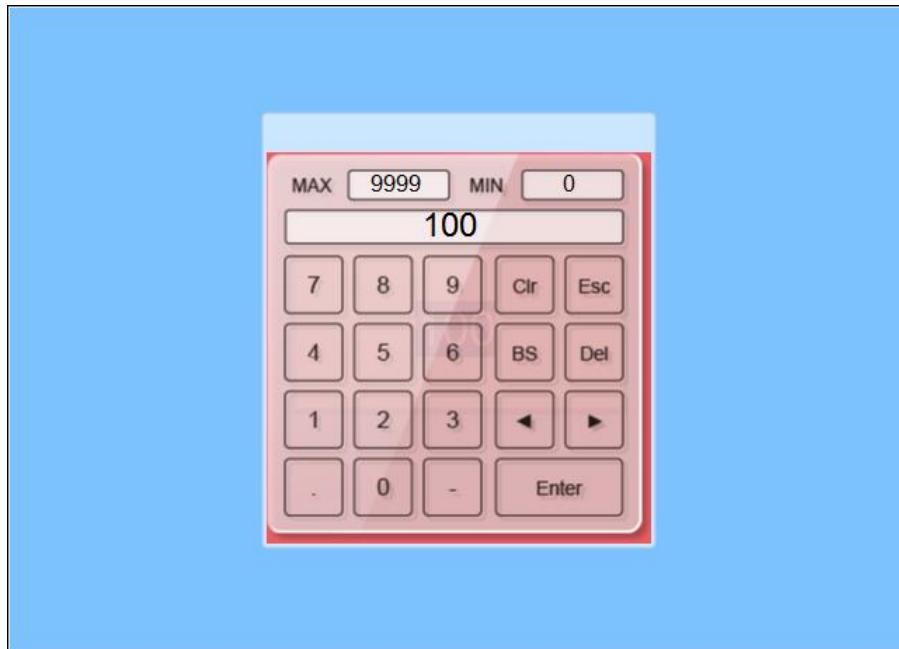
另外，cMT / cMT X 系列的視窗設定支援 [透明度] 調整，可應用於彈出視窗、直接/間接視窗和鍵盤視窗。

如下圖所示，當 [透明度] 的數值越大時，基本視窗與彈出視窗重疊的物件之可見度越清晰。

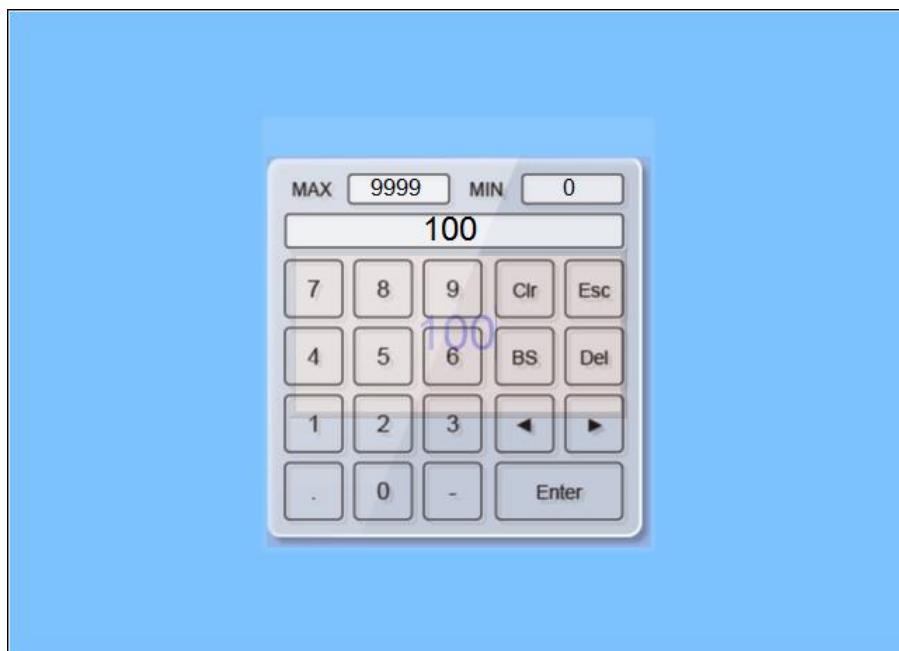


## 範例

當 [透明度] 設定為 40%時:



當【透明度】設定為 90%時：



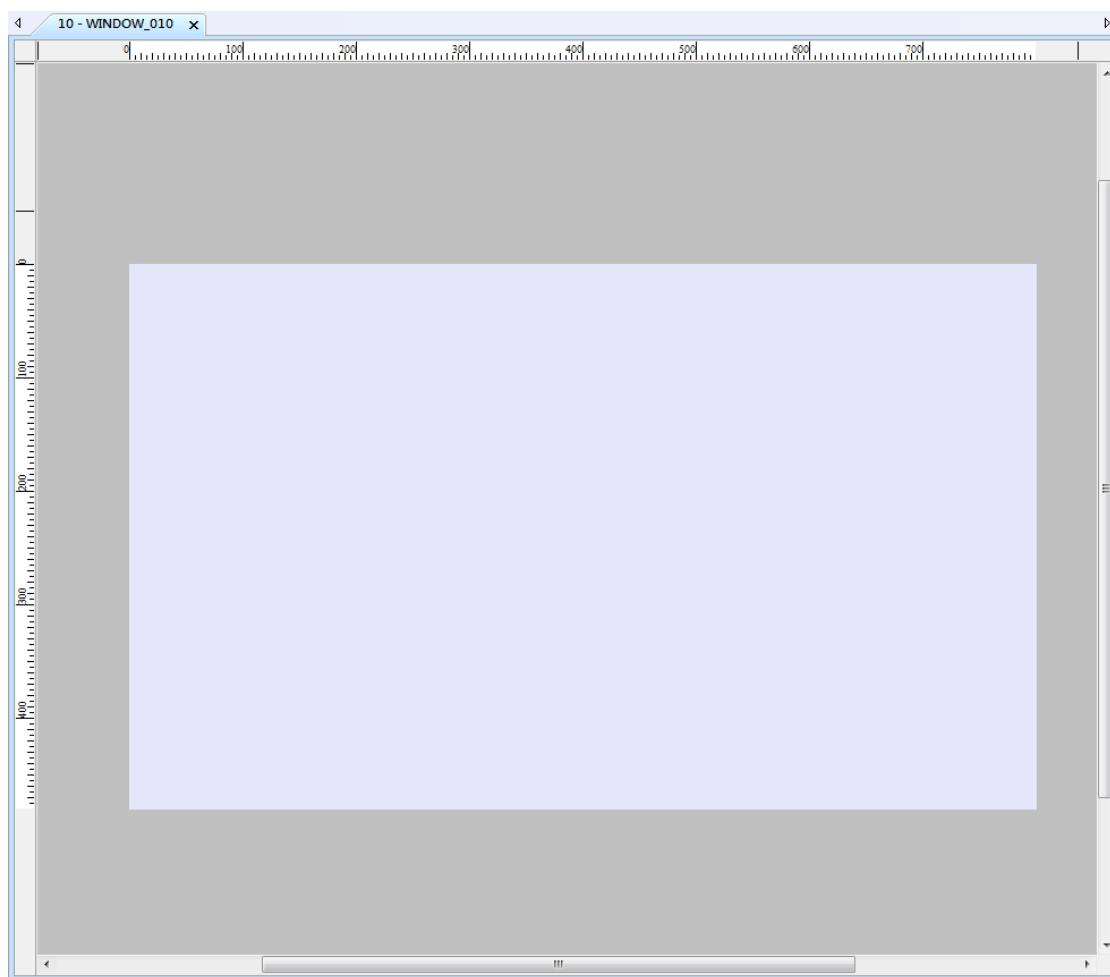
## 6.5. 編輯視窗應用

EasyBuilder Pro 在編輯視窗的操作具有尺規與快速複製物件等功能，提供人性化的編輯方式。

### 6.5.1. 編輯視窗

如下圖所示，編輯視窗可分成淡紫色的可視區域以及灰色的非可視區域。可視區域即是下載至 HMI 後會顯示在螢幕上的部份；反之，放置在非可視區域的物件，將不會顯示在 HMI 畫面上，但仍會與設備通訊。

在應用上可將背景物件（計時器、定時式資料傳輸等）放在非可視區域，維持編輯畫面的整齊；或是直接使用文字物件下註解。



#### 選擇物件

當使用選擇物件時，按下滑鼠左鍵拉取可一次選擇多個物件。

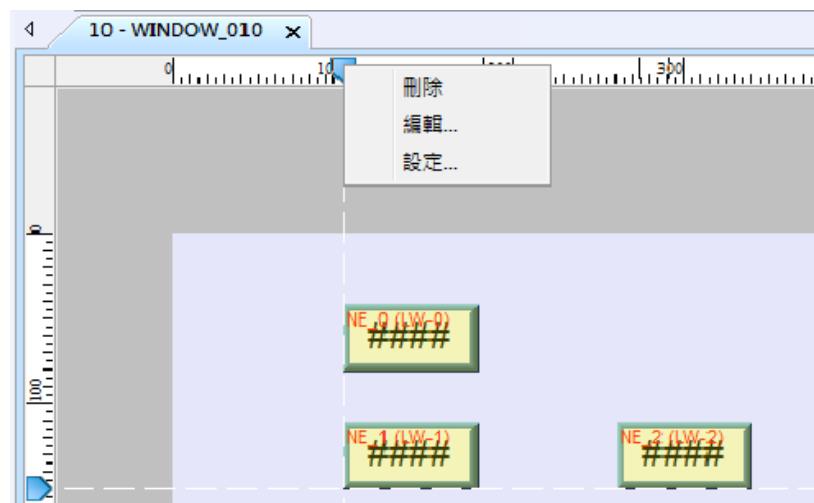


#### 畫面平移

當使用畫面平移時，按下滑鼠左鍵就可以像瀏覽 PDF 檔案一樣直接抓取畫面並任意移動。

### 6.5.2. 尺規功能

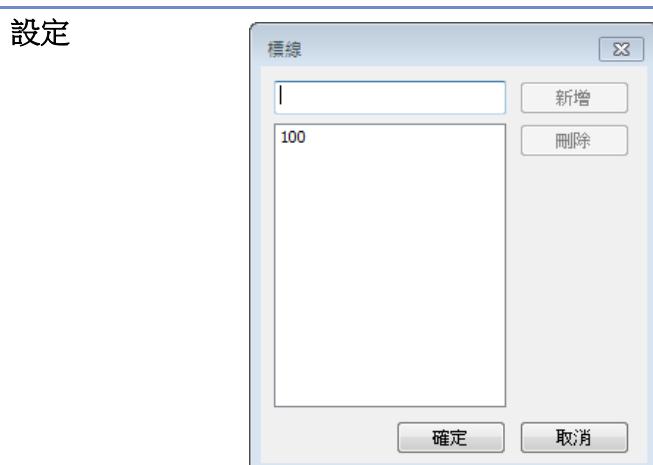
若勾選 [檢視] » [尺規]，編輯視窗的上方與左方會出現尺規，可直接在尺規上方按下右鍵新增對齊標線，對於編排畫面與物件對齊有很大的幫助。在拖曳移動或縮放物件時，物件位置會自動吸附對齊標線。



設定	描述
<b>刪除</b>	刪除標線。
<b>編輯</b>	



手動編輯標線位置。



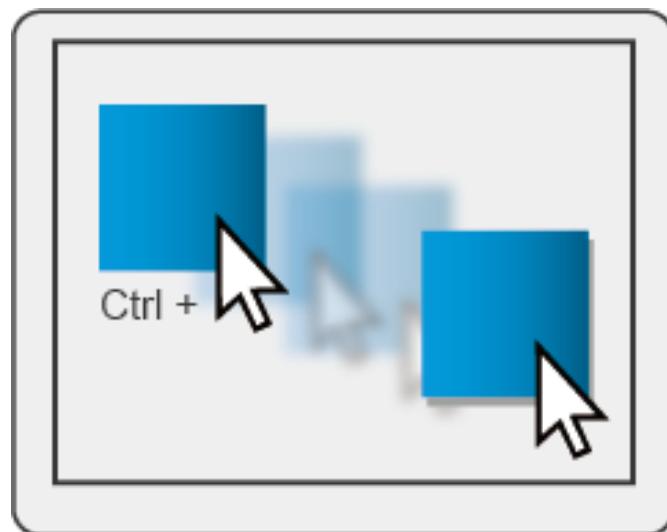
可在此新增或刪除多個標線。

此外，也可以在主選單中選擇 [檔案] » [網格/對齊/標線設定] 視窗中自由更改標線的顏色及樣式。



### 6.5.3. 快速複製功能

在物件上按住 Ctrl 按鍵並拖曳物件即可複製物件到指定的位置。



# 7. 事件登錄

本章節說明如何設定與使用事件登錄。

7.1.	概要 .....	7-2
7.2.	事件登錄管理 .....	7-2
7.3.	建立一個新的事件記錄 .....	7-9

## 7.1. 概要

使用事件登錄的基本程序如下：

1. 定義事件觸發條件與內容。
2. 依條件觸發事件。
3. 可將事件記錄儲存到指定位置。
4. 可使用物件檢視事件的完整處理週期。

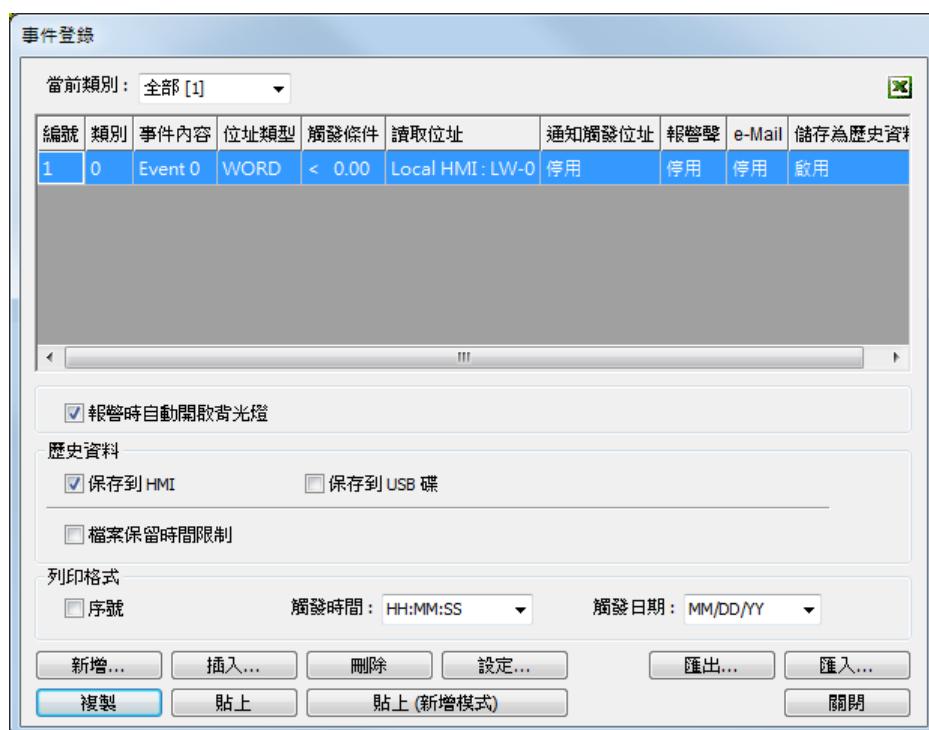
本章節說明如何設定與使用事件登錄。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 7.2. 事件登錄管理

透過報警條  、報警顯示  、事件顯示  等物件，可以得知事件從發生 → 等待處理 → 警報解除的時間。首先需定義事件的內容。事件登錄筆數最大上限為 10000 筆。

### 7.2.1. eMT、iE、XE、mTV、iP 系列



#### 設定

#### 描述

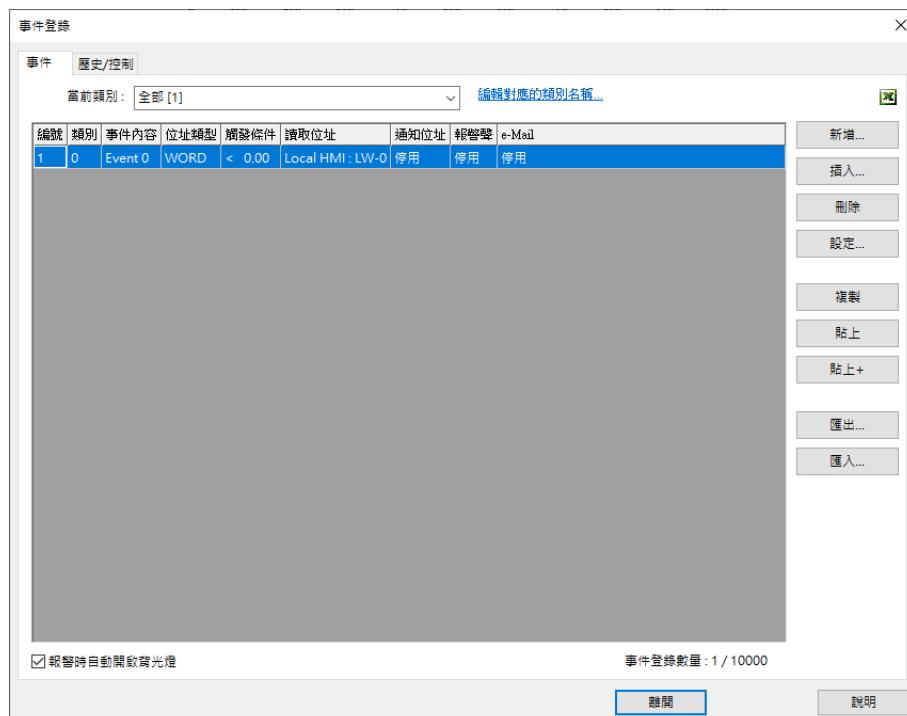
##### 當前類別

提供事件分類功能，將事件分成 0 ~ 255 個類別，可選擇一個類別來輸入或列出事件資料。[] 中顯示此類別的事件資料數量。

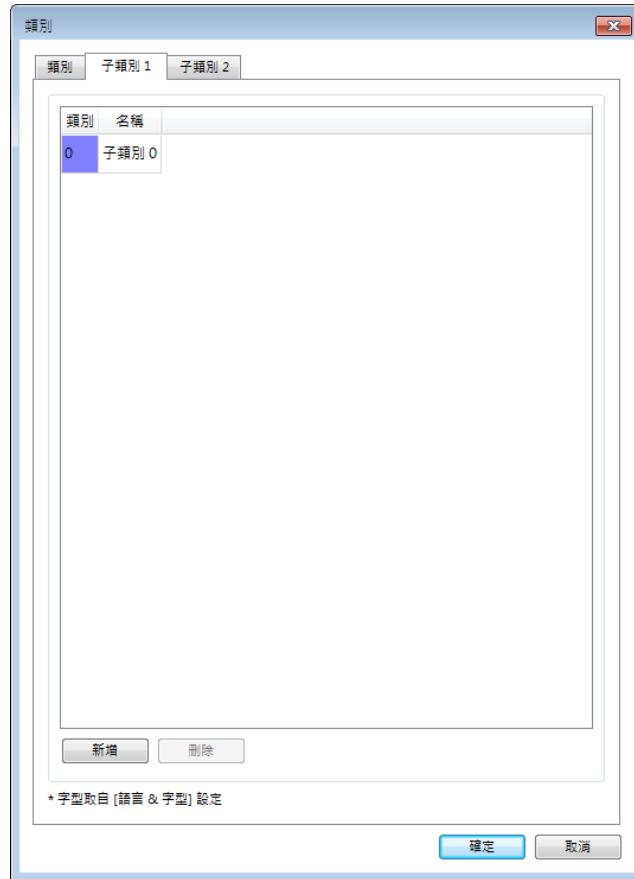
## 事件登錄

<b>歷史資料</b>	指定事件記錄檔案的儲存位置。當事件一發生，HMI 即會立即儲存到歷史資料中。若使用連線或離線模擬功能時，檔案一律存放在安裝目錄下的 HMI_memory / SD_card / USB 資料夾內。
<b>檔案保留時間限制</b>	此項設定值決定事件記錄檔案保留在 HMI 記憶體內的最大數量(不包含今天)。也就是說，若 [保留時間] 設定為兩天，系統將只保留除了今天以外的最新的兩個事件記錄檔案，之前的檔案將自動被刪除，以避免儲存空間被耗盡。
<b>列印格式</b>	需先在 [系統參數設定] » [HMI 屬性] 頁籤選擇印表機型號，才允許設定當事件觸發時，欲列印的格式。
<b>複製</b>	複製所選擇的事件項目。
<b>貼上</b>	貼上時，會用所複製的資料取代所選取的事件。操作前，會彈出警示確認是否覆蓋。
<b>貼上 (新增模式)</b>	貼上時，會將所複製的資料置於列表最後。

## 7.2.2. cMT、cMT X 系列



設定	描述
<b>當前類別</b>	提供事件分類功能，將事件分成 0 ~ 255 個類別，可選擇一個類別來輸入或列出事件資料。[] 中顯示此類別的事件資料數量。 <b>編輯對應的類別名稱：</b> 可編輯各類別的對應名稱。在此可新增子類別，子類別 1 與子類別 2 分別可再新增 0~255 的類別區分，用於更細節的事件分類。

**複製**

複製所選擇的事件項目。

**貼上**

貼上時，會用所複製的資料取代所選取的事件。操作前，會彈出警  
示確認是否覆蓋。

**貼上+**

貼上時，會將所複製的資料置於列表最後。

**設定****描述****歷史**

保存事件記錄檔案的儲存位置。可選擇儲存至 HMI 記憶體 (10000 限制)、HMI 記憶體 (直到空間存滿)、SD 卡、USB 碟，另外可啟用同步儲存事件記錄檔案至資料庫伺服器。儲存規則如下：

- 當選擇為 HMI 記憶體(10000 限制)，HMI 的事件記錄至 10000 筆時，會自動刪除最早的 1000 筆資料後繼續儲存資料。
- 如果選擇 HMI 記憶體 (直到空間存滿)，系統會持續將資料存進人機的記憶體內，但無法同步至資料庫伺服器。cMT / cMT X 的事件登錄至 HMI 記憶體滿時，會自動刪除最早的 1000 筆資料後繼續儲存資料。
- 當選擇為 USB 碟與 SD 卡，HMI 的事件記錄至 10000 筆時，會自動將資料儲存至指定的外接儲存裝置，刪除 HMI 裡最早的 1000 筆資料。
- 當選擇使用同步至資料庫伺服器時，須選擇預先設定的資料庫。
- 若外接儲存裝置內已經有事件記錄，當每次同步時，新的事件記錄將附加在原有的檔案。
- 若將外接儲存裝置從 HMI 上移除，或是中斷與資料庫伺服器的連接，之後在新增的事件記錄不超過 9000 筆時就插回外部裝置/回復連接，則在中斷的期間，事件記錄仍然會保存在 HMI

內不會被清除。若在中斷期間，事件記錄已經超過 9000 筆，則較舊的資料就會被刪除，即使之後將外接儲存裝置插回或是回復連接也無法同步到被刪除的資料。

<b>歷史資料顯示來源</b>	指定事件記錄從選擇的路徑讀取歷史資料。																
<b>檔案保留時間限制</b>	事件記錄檔案保留在 HMI 記憶體內的最大天數 (不包含今天)。 在進行事件記錄同步時才會檢查檔案的時間並清除較舊的檔案。 若 [保留時間] 設定為兩天，則系統將只保留今天以外，最新兩天的事件記錄檔案，之前的檔案將自動被刪除。																
<b>限制寫入 HMI 快閃記憶體的頻率</b>	啟用後，系統將以 10 秒的時間間隔將操作記錄儲存在 HMI 上，為了避免操作紀錄在兩次儲存動作間因關機而造成資料的流失，EasyBuilderPro 提供系統暫存器 LB-9034，只需對其送出 ON 的訊號，系統即會執行一次儲存動作。																
<b>啟動時，將關機之前觸發的事件標記為已恢復</b>	啟用後，系統將自動恢復關機前已觸發的事件。 適用範圍：儲存至 HMI 記憶體的末 10000 筆事件。																
<b>週期性自動同步</b>	啟用後，HMI 不只會參照 [歷史資料] 的同步規則，資料也會依據設定的週期保存到外部裝置。 單位：分鐘。 數值：1 ~ 1440。																
<b>控制位址</b>	若勾選 [控制位址] 並啟用 [保存檔案]，使用者可以在控制位址輸入數值，來同步或清除裝置中的事件記錄檔案。 當寫入特定數值於控制位址時，可觸發相關命令。																
<table border="1"> <thead> <tr> <th>數值</th> <th>命令</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>清除 HMI 上的資料</td> </tr> <tr> <td>2</td> <td>將資料同步到外接裝置</td> </tr> <tr> <td>3</td> <td>先將資料同步到外接裝置後，再清除 HMI 上的資料</td> </tr> <tr> <td>4</td> <td>在更換 HMI 後，延用原先儲存在 USB 碟、SD 卡或資料庫上的歷史資料的功能</td> </tr> <tr> <td>6</td> <td>釋放事件登錄的歷史資料 (*.db) 未用到的空間。當歷史資料選擇儲存到 HMI 記憶體 (直到空間存滿) 時，可使用此命令。</td> </tr> <tr> <td>11</td> <td>事件記錄的訊息內容更新為新匯入的 [字串表]</td> </tr> <tr> <td>12</td> <td>重置事件發生次數的記錄 (顯示於事件顯示與報警顯示)</td> </tr> </tbody> </table>		數值	命令	1	清除 HMI 上的資料	2	將資料同步到外接裝置	3	先將資料同步到外接裝置後，再清除 HMI 上的資料	4	在更換 HMI 後，延用原先儲存在 USB 碟、SD 卡或資料庫上的歷史資料的功能	6	釋放事件登錄的歷史資料 (*.db) 未用到的空間。當歷史資料選擇儲存到 HMI 記憶體 (直到空間存滿) 時，可使用此命令。	11	事件記錄的訊息內容更新為新匯入的 [字串表]	12	重置事件發生次數的記錄 (顯示於事件顯示與報警顯示)
數值	命令																
1	清除 HMI 上的資料																
2	將資料同步到外接裝置																
3	先將資料同步到外接裝置後，再清除 HMI 上的資料																
4	在更換 HMI 後，延用原先儲存在 USB 碟、SD 卡或資料庫上的歷史資料的功能																
6	釋放事件登錄的歷史資料 (*.db) 未用到的空間。當歷史資料選擇儲存到 HMI 記憶體 (直到空間存滿) 時，可使用此命令。																
11	事件記錄的訊息內容更新為新匯入的 [字串表]																
12	重置事件發生次數的記錄 (顯示於事件顯示與報警顯示)																

只有當事件本身沒有啟用 [讀取和重置發生次數] 位址時才有效。

若使用者沒有手動在控制位址上輸入以上數值，系統會參照 [歷史資料] 說明中描述的同步規則保存事件記錄。

#### 狀態位址

若控制位址是 LW-n，則控制位址的連續暫存器 (LW-n+1 ~ LW-n+4)

#### 錯誤位址

用來表示歷史資料狀態及錯誤訊息。詳細位址對應請參考事件登錄的提示。



數值	狀態位址 : LW-n+1 及 LW-n+3
0	未連接外部裝置或資料庫
1	正在連接外部裝置或資料庫
2	已連接外部裝置或資料庫
3	正在將事件記錄儲存到外部裝置或資料庫，完成後，會回復至數值 2
數值	錯誤位址 : LW-n+2 及 LW-n+4
0	沒有錯誤
1	未知錯誤
2	連接外部裝置或資料庫失敗
3	操作權限不足
4	錯誤的資料庫名稱
5	資料格式不一致
6	無法連接資料庫的表格
7	無法建立表格
8	無法寫入表格
9	無法開啟資料庫
10	資料庫損毀

#### Note

- 若使用者需將 SD 卡或 USB 碟拔除，或中斷與資料庫伺服器連接時，可以先使用控制位址將事件記錄同步。
- 使用 cMT Viewer 監控多台人機時，當下操作人機畫面的 Start Button 閃爍表示其他人機有事

## 事件登錄

件發生，用來提醒使用者切換人機並確認事件。

- 儲存於 HMI 記憶體超過 10000 筆的事件，不適用於 [啟動時，將關機之前觸發的事件標記為已恢復] 選項。

### 7.2.3. Excel 編輯

點擊事件登錄視窗右上角的 Excel 小圖示，可直接開啟 Excel 表格做為編輯時的範例參考。

此範例為安裝目錄下的 EventLogExample.xls 檔案，範例檔中有設計好的驗證機制和下拉式選單。

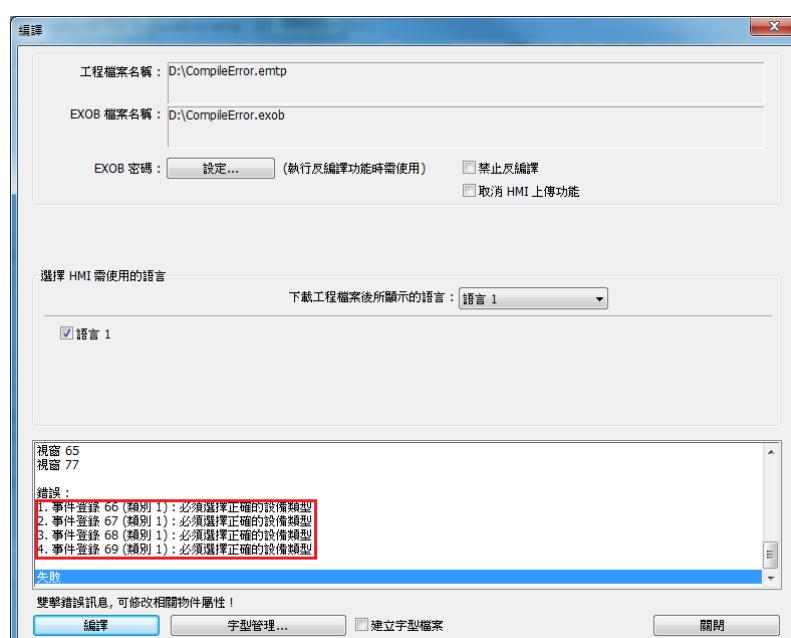
	A	B	C	D	E	F	G	H	I	J	K
1	Category	Priority level	Address type	PLC name	Device type	System tag	User-defined tag	Address	Index	Data Format	Enable
2	0	Middle	Word	Local HMI	LW	False	False	100	null	32-bit Signed	True
3	1	Low	Bit	Local HMI	LB-9009	True	False	9009	IDX 5	16-bit BCD	Use
4										16-bit BCD	
5										32-bit BCD	
6										16-bit Unsigned	
7										16-bit Signed	
										32-bit Unsigned	
										32-bit Signed	
										32-bit Float	



- [System tag] 與 [User-defined tag] 請勿同時設為 true，否則系統會將其視為 [System tag]，並將 [User-defined tag] 視為 false。若在 [Device type] 所輸入的資料為 [User-defined tag]，請將 [System tag] 設為 false。
- 當在 Excel 表中將 [User-defined tag] 設為 true，但若系統將 [Device type] 與系統中使用者自訂的 tag 做比對，卻找不到合適的 tag，則系統會自動將事件登錄中的 [User-defined tag] 設為 false。
- [Color] 格式為 R:G:B，各介於 0 ~ 255 之間的整數。
- 在匯入文字標籤庫或聲音庫前，請確認系統中已存在對應的名稱。

### 7.2.4. 快速查閱無效的事件

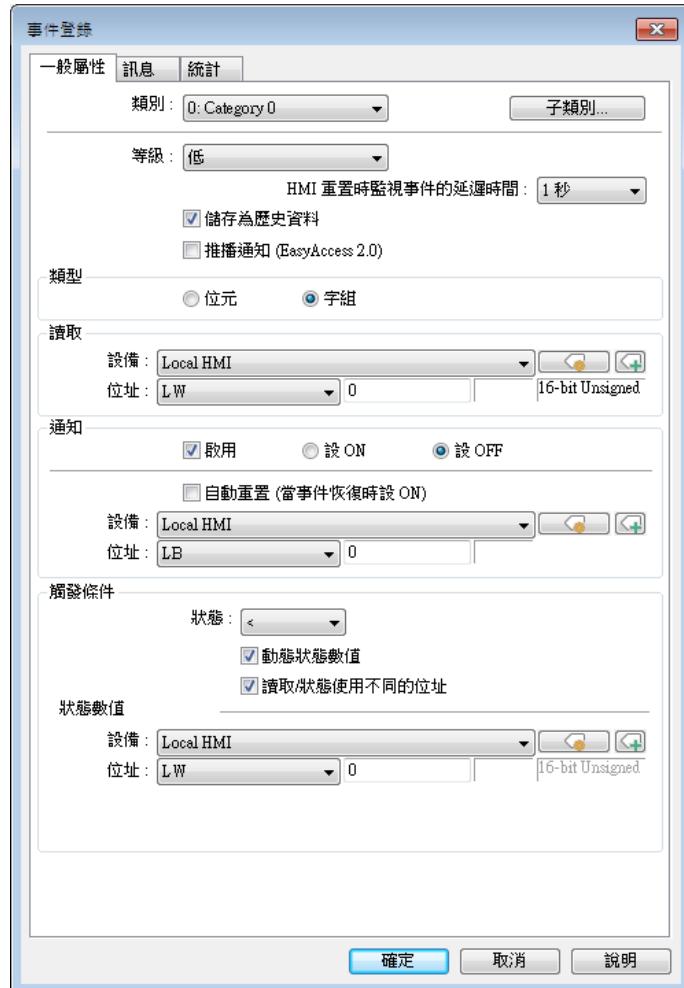
在編譯工程檔案時，若有無效的事件登錄，在編譯對話窗中會顯示無效的事件登錄項目。雙點擊錯誤的項目，即會自動開啟 [事件登錄] 物件並指向錯誤的項目。



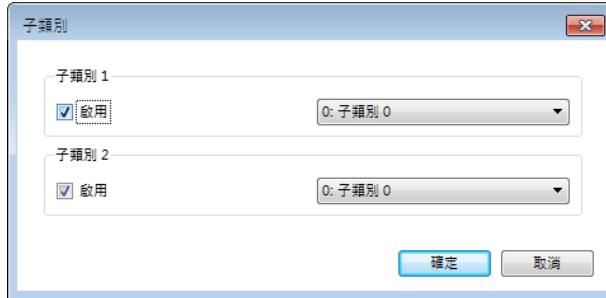
### 7.3. 建立一個新的事件記錄

#### 一般屬性設定

點選【事件登錄】視窗的【新增】，將出現【一般屬性】設定頁。

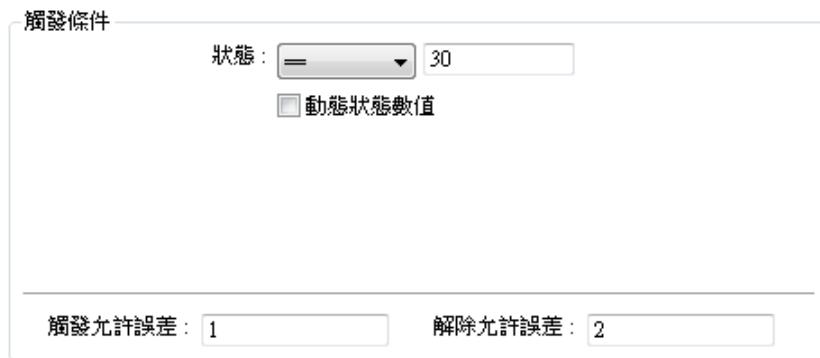


設定	描述
類別	選擇事件類別，從 0 ~ 255。
子類別	可選擇是否要啟用定義於【編輯對應的類別名稱】的子類別 1 與子類別 2。



<b>等級</b>	報警條與報警顯示物件上的事件會依照等級先執行排序後，其次再依照時間排序。
<b>儲存為歷史資料</b>	當在事件登錄的主設定頁中，啟用了 [歷史資料] 下的保存功能，則可分別設定每個事件是否要保存為歷史資料。
<b>推播通知 (EasyAccess 2.0)</b>	當事件發生時，在 iOS®/Android®裝置上可接收來自 EasyAccess 2.0 的訊息推播。
<b>HMI 重置時監視事件的延遲時間</b>	與 [動態狀態數值] 搭配使用。當 HMI 啟動時，動態狀態數值可能尚未設定，使用 [HMI 重置時監視事件的延遲時間] 可讓系統延遲讀取事件的時間。此功能僅在 HMI 啟動時執行一次。
<b>讀取位址</b>	系統將讀取此位址所獲得的數據，來檢查事件是否滿足觸發條件。
<b>通知</b>	若勾選，系統會在事件發生時，將指定暫存器狀態設為 [ON] 或 [OFF]。 <b>自動重置</b> 當警報條件解除後，恢復 [通知] 位址至原狀態。例如：當警報觸發時，通知指定位元為 ON，則當警報解除時，若有勾選 [自動重置]，系統會將指定位元設為 OFF。
<b>觸發條件</b>	當選擇位元時，事件登錄將偵測一個位元位址的狀態。 當選擇字組時，事件登錄將偵測一個字組位址的值是否等於、大於或小於一個特定數值。請見以下範例 1 與範例 2。 <b>動態狀態數值</b> 啟用後，觸發條件會參照指定位址的數值。若未勾選 [讀取/狀態使用不同的位址]，則觸發條件的位址為 [讀取位址 + 1]。 <b>讀取/狀態使用不同的位址</b> 啟用後，可自行設定觸發條件參照的來源位址。

## 範例 1



上面的設定內容表示

當 [讀取位址] 中的數據大於等於  $29 (= 30 - 1)$

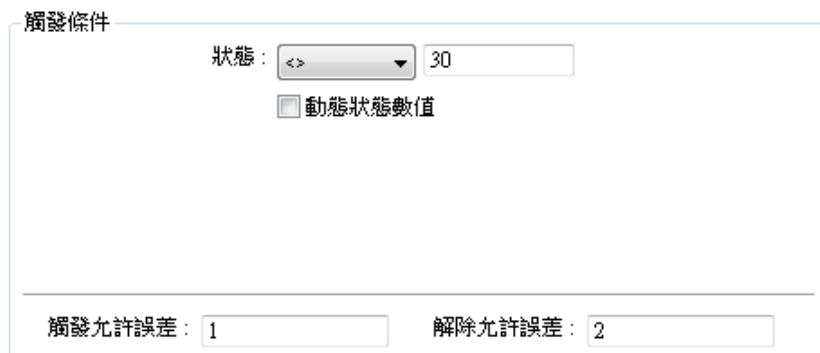
或小於等於  $31 (= 30 + 1)$  時，事件將被觸發。也就是事件被觸發的條件為：

$$29 \leq [\text{讀取位址}] \text{ 中的數據} \leq 31$$

事件被觸發後，當 [讀取位址] 中的數據大於  $32 (= 30 + 2)$  或小於  $28 (= 30 - 2)$  時，系統將恢復為正常狀態。也就是系統恢復為正常狀態的條件為：

$$[\text{讀取位址}] \text{ 中的數據} < 28 \text{ 或 } [\text{讀取位址}] \text{ 中的數據} > 32$$

## 範例 2



上面的設定內容表示

當 [讀取位址] 中的數據小於  $29 (= 30 - 1)$

或大於  $31 (= 30 + 1)$  時，事件將被觸發。也就是事件被觸發的條件為：

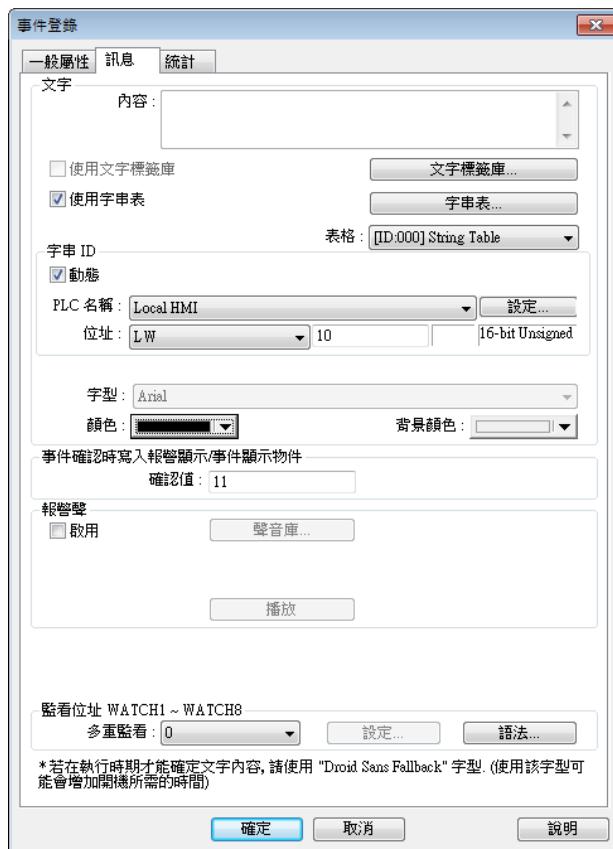
$$[\text{讀取位址}] \text{ 中的數據} < 29 \text{ 或 } [\text{讀取位址}] \text{ 中的數據} > 31$$

事件被觸發後，當 [讀取位址] 中的數據大於等於  $28 (= 30 - 2)$  或小於等於  $32 (= 30 + 2)$  時，系統將恢復為正常狀態。也就是系統恢復為正常狀態的條件為：

$$28 \leq [\text{讀取位址}] \text{ 中的數據} \leq 32$$

## 事件登錄

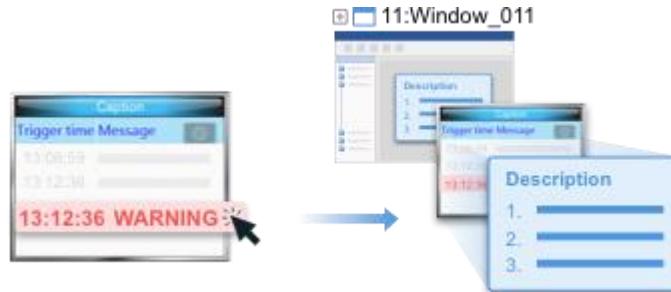
## 訊息設定



設定	描述
內容	事件記錄在 [報警條]、[報警顯示] 與 [事件顯示] 物件中顯示的訊息內容。並可在內容中使用 [監看位址] WATCH1 至 WATCH8 來引用資料。文字內容來源亦可為文字標籤庫或是字串表。
字串 ID	事件顯示/報警顯示物件依據字串 ID 的暫存器地址顯示對應訊息內容。  [記錄字串 ID] 勾選後，事件產生時，會同時存下字串編號，先前觸發的事件仍可以維持發生當下的訊息。
字型 / 顏色 / 背 景顏色	每一個事件可以分別設定字型、顏色、背景顏色。[報警條] 物件會採用字型及顏色的設定。[報警顯示] 與 [事件顯示] 物件則會採用字型、顏色、背景顏色的設定。歷史模式的 [事件顯示] 物件則不採用此項設定。
事件確認 時寫入	當報警顯示或事件顯示物件中的事件訊息被確認時，系統會將此確認值寫入至物件指定的 [確認] 位址中。可搭配間接視窗物件的讀取位址，將確認值寫入後，彈出一視窗藉以說明報警

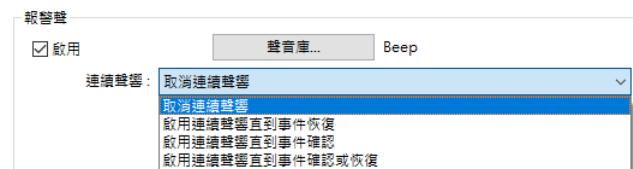
## 事件登錄

事件的處置方式。如下圖所示，當點選某一事件，若確認值設為 11，此時將彈出視窗 11。



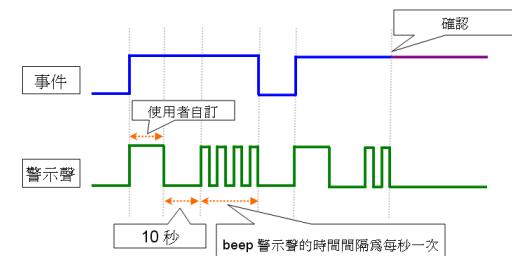
請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 報警聲



若啟用，當事件發生時會播放指定的聲音，並且可以選擇持續發出警報聲響，直到該事件被確認或恢復正常時，才會停止發出聲音。

若勾選使用持續警報聲響時，可以選擇在警報被觸發後，延遲所指定的時間後才開始發出 Beep 聲音。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

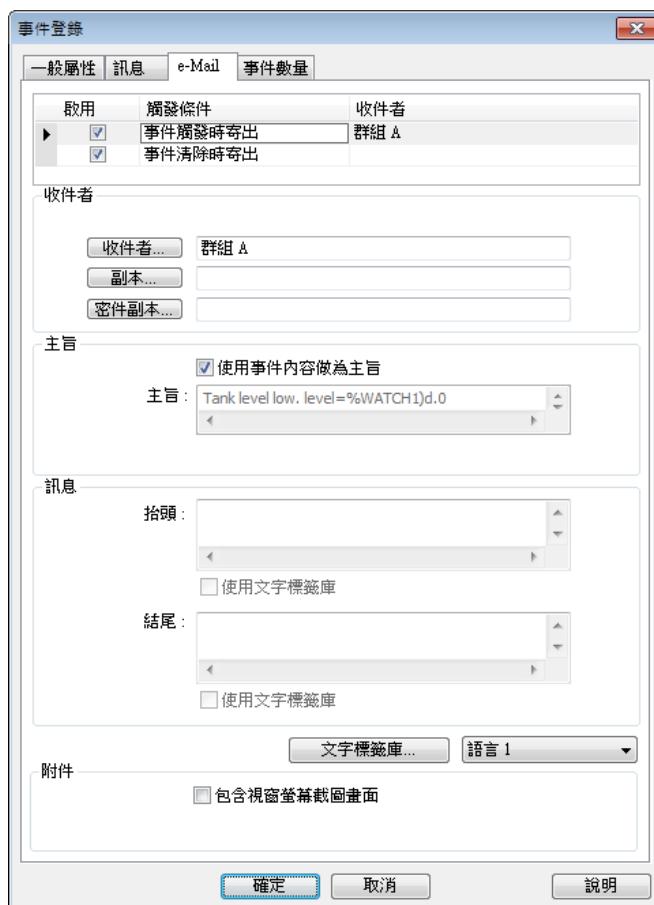
### 監看位址

使用者可以設定當事件觸發時，顯示監看位址所設定暫存器內的數值。最多可同時監看八個位址。點選 [語法] 可查看事件內容套用監看位址數據的語法。

請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## e-Mail 設定

請先在 [系統參數設定] » [郵件] 啟用此功能。



設定	描述
收件者	可分別在收件者、副本、密件副本欄位，選擇收件者。
主旨	可輸入信件顯示的主題內容。
訊息	可輸入信件顯示的抬頭與結尾的內容。
附件	若勾選，可指定某一視窗的螢幕截圖當作附件。

### Note

- 郵件的寄件者所定義事件的等級優先順序，會影響信箱伺服器上的傳輸服務對郵件的處理。事件等級設定為緊急，等同於高重要性；事件等級設定為高與一般，等同於一般重要性；事件等級設定為低，等同於低重要性。

## 事件登錄

### 統計設定



設定	描述
發生次數 讀取和重 置位址	若啟用，可將 HMI 自開機後所發生的事件次數寫入至字組位址。 此字組位址可以讀取 / 寫入。
累積時間 讀取和重 置位址	若啟用，可將該筆事件從發生後至恢復期間所累積的秒數寫入至字組位址。此字組位址可以讀取 / 寫入。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 安全設定

啟用此功能後，可限制事件只有在指定暫存器符合特定條件時，才會被記錄。



設定	描述
使用暫存 器狀態/數 值	當指定暫存器符合條件時，事件才會被記錄。



#### Note

- 事件若在啟用安全設定時被觸發，則之後事件在觸發狀態下被停用安全設定時，仍可以在 [事件顯示]、[報警顯示] 物件上被確認。
- 當事件有啟用安全設定，且條件未達到啟用狀態時，該事件的地址不會與裝置通訊。

# 8. 資料取樣

本章節說明如何設定與使用資料取樣。

8.1.	概要 .....	8-2
8.2.	資料取樣記錄管理 .....	8-2
8.3.	新增一個資料取樣 .....	8-2
8.4.	外部裝置或資料庫同步 cMT Viewer 資料 .....	8-12
8.5.	查看 cMT Viewer 特定日期或特定檔案的歷史資料 .....	8-12

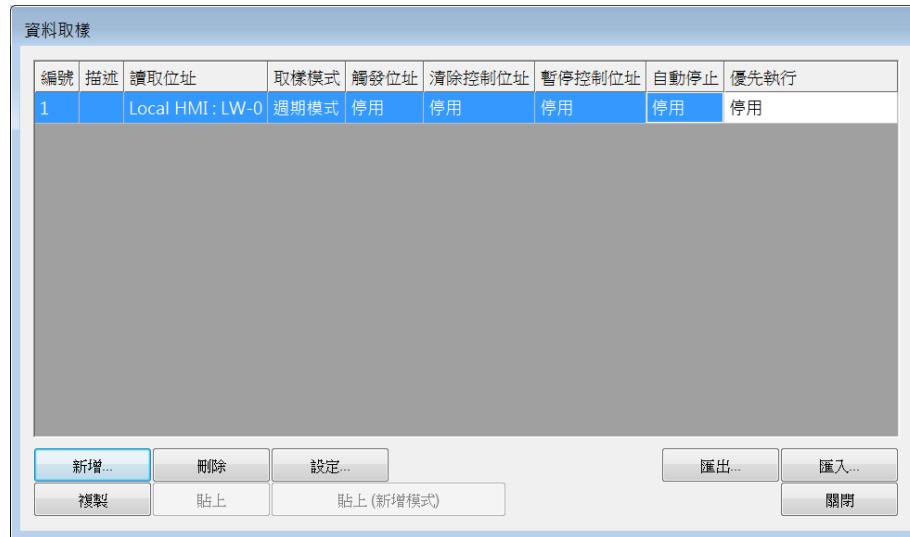
## 8.1. 概要

定義「資料取樣」的取樣方式，例如：取樣時間，取樣位址，及字組長度後，可將已獲得的取樣資料儲存到指定的位置，如 HMI 記憶體、SD 卡或 USB 碟。資料取樣可搭配使用趨勢圖或歷史數據顯示物件檢視資料取樣記錄的內容。

## 8.2. 資料取樣記錄管理

新增一個資料取樣，請依照下列步驟：

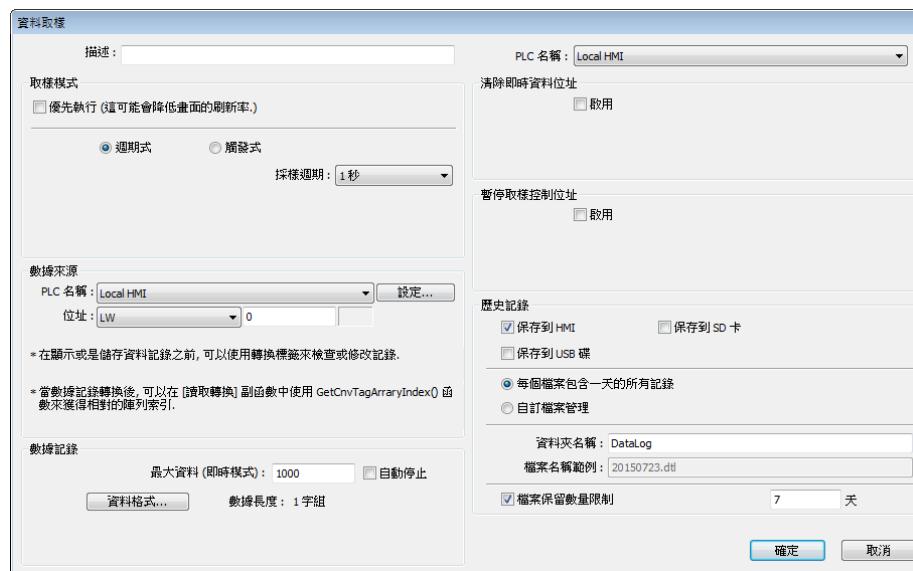
1. 點選功能表 [資料/歷史]，再點擊 [資料取樣]。
2. 點擊 [新增] 開始相關設定，如下圖所示：



## 8.3. 新增一個資料取樣

以下介紹如何設定一筆資料取樣物件。

eMT、iE、XE、mTV、iP 系列

**設定****描述****取樣模式****優先執行**

優先執行資料取樣功能。

請注意，由於 HMI 會優先執行資料取樣的動作，因此 HMI 畫面的物件更新速度會減慢，請避免太多優先執行的取樣設定。

**週期式**

用固定的時間頻率進行資料取樣，[採樣週期] 可設定範圍從 0.1 秒至 120 分鐘。

**觸發式**

利用一個特定位元位址的狀態，來觸發取樣動作。

**[模式] 可為：**

**[OFF -> ON]** 當指定位址的狀態從 OFF 變為 ON，會觸發資料取樣。

**[ON -> OFF]** 當指定位址的狀態從 ON 變為 OFF，會觸發資料取樣。

**[OFF <-> ON]** 只要指定位址的狀態改變，就會觸發資料取樣。

**觸發後設為 ON/OFF**

若勾選，在觸發資料取樣後，系統會將觸發位元復歸為 ON/OFF。

**數據來源**

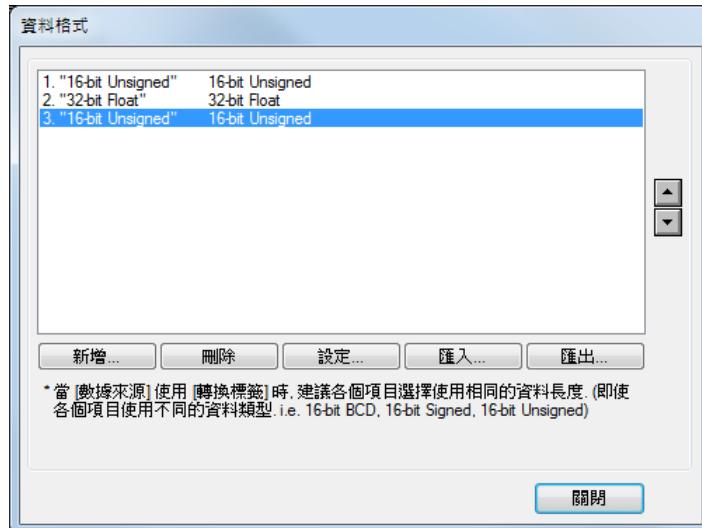
選擇一個設備位址作為取樣數據的來源。如需對數據進行運算時，請使用具有 [讀取轉換] 功能的使用者定義標籤。陣列位址可搭配 GetCnvTagArrayIndex 函數取得相對的陣列索引後再進行運算。

**數據記錄  
(即時模式)**

在即時模式下，如無勾選 [自動停止]，HMI 最多可保留 86400 筆資料，超過 86400 筆後，會從最舊的資料取樣開始刪除。

**資料格式**

可設定讀取多個不同格式的連續位址的資料。例如將位址及格式分別設定為 LW-0 (16-bit Unsigned)、LW-1 (32-bit Float)、LW-3 (16-bit Unsigned)。上限為 1000 筆。



### 自動停止

當資料取樣達到 [最大資料] 數量後，即停止資料取樣動作。

請見《8.3.1 自動停止選項範例》。

**清除即時  
資料位址** 當指定位元位址的狀態由 [OFF -> ON] 或 [ON -> OFF] 時，將清除在趨勢圖 [即時模式] 下已取樣的資料，取樣資料的數目也會被歸零，但不影響已存成檔案中的歷史取樣資料。

**暫停取樣  
控制位址** 當指定位元位址的狀態被觸發時，將暫停取樣動作，直到指定位址的狀態恢復。

**歷史記錄**

**保存到 HMI**  
將取樣資料儲存在 HMI 裡。資料每 10 秒儲存一次；若少於 10 秒，可以使用系統暫存器 [LB-9034] 來強迫儲存。(關於 LB-9034 的執行限制，請見下方 Note 第六點)

#### 保存到 SD 卡 / USB 碟

將資料取樣儲存到指定的外部裝置中。

#### 每個檔案包含一天的所有記錄

資料取樣將以一天為單位，按日期將檔案儲存於指定檔名的資料夾內，且檔案名稱為 *yyyymmdd.dtl*。

#### 資料夾名稱

設定取樣資料夾的名稱，必須全部由 ASCII 字元所組成。

儲存的方式為：[保存位置] \ [取樣資料夾名稱] \ [檔案名稱]

#### 檔案保存數量限制

此項設定值用來決定資料取樣記錄檔案被保留的數量 (不包含當下使用的.dtl 檔案)。也就是說，若設定保留為兩天，系統則會保留除了目前正在記錄的檔案以外，最新的兩個資料取樣檔案。

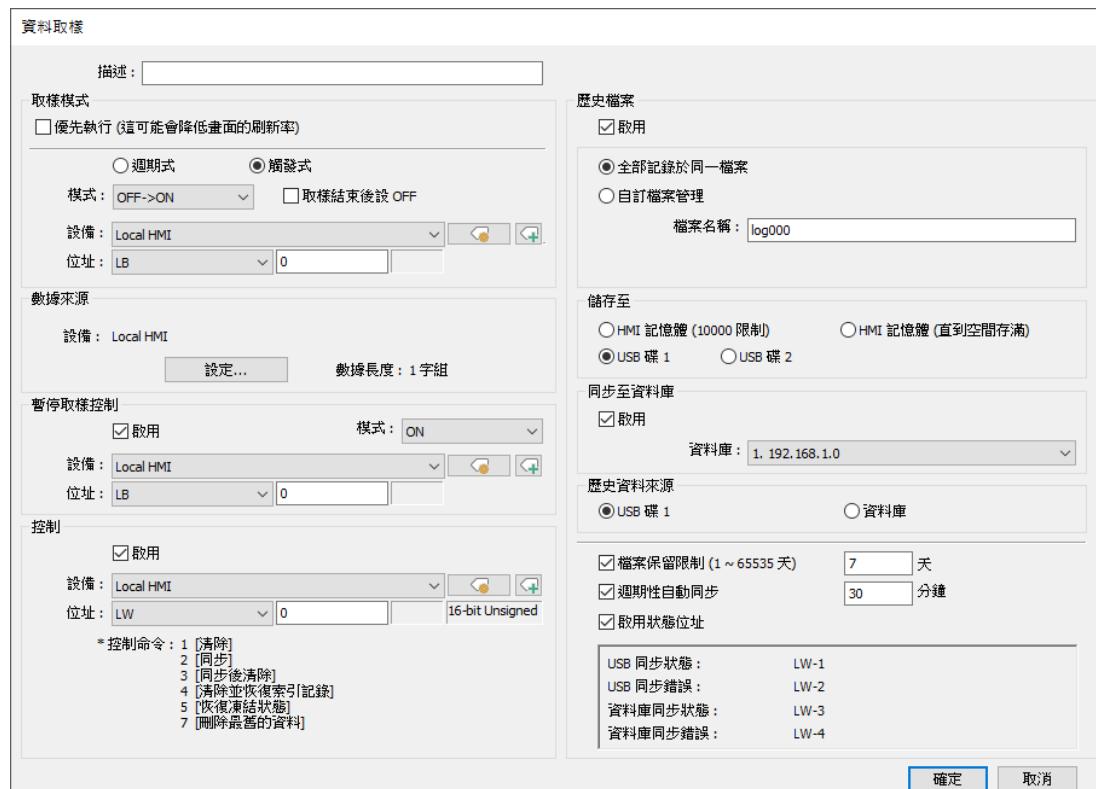
**自訂檔案  
管理** 提供自行設定資料取樣檔案名稱 (\*.dtl) 與檔案劃分的方式。

詳細規則請見《8.3.2 自訂檔案管理》。

 Note

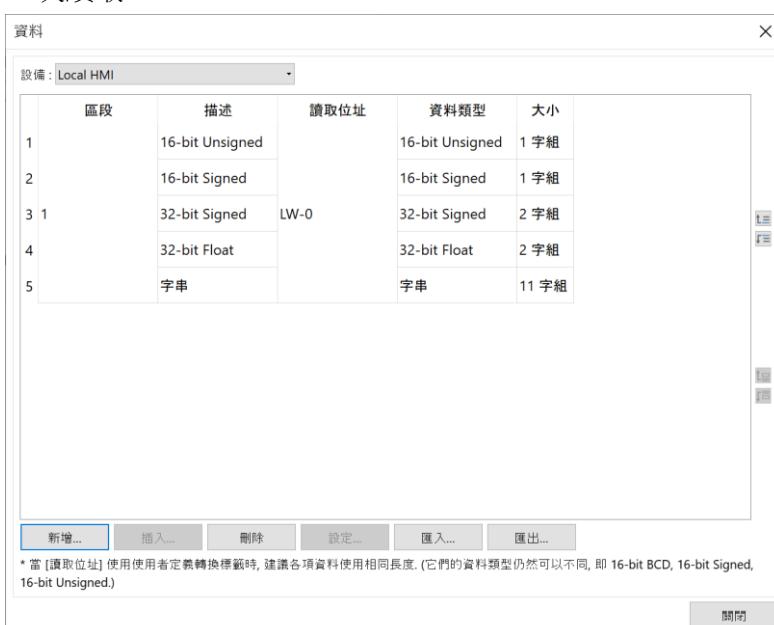
- 一筆取樣資料可包含超過一項以上的數據，資料取樣動作可以同時擷取不同型態的數據。您可以自行定義一筆取樣資料的內容。例如：您總共定義了三筆數據，總長度為 4 words，即每次的取樣動作，系統會從指定的數據來源位址擷取長度為 4 words 的數據，作為一筆取樣資料的內容。
  - 當使用 [每個檔案包含一天的所有記錄] 且假設 [檔案保存數量限制] 設定保留數量為 2 個檔案，則系統將僅保留昨天與前天的資料取樣記錄檔案，超過這個時間範圍的檔案將自動被刪除，以避免儲存空間被耗盡。
  - 當使用 [自訂檔案管理] 且 [檔案保存數量限制] 設定保留數量為 2 個檔案，則系統除了保留目前正在取樣的檔案之外，將另外保留最新的 2 個檔案，總計為 3 個檔案。其餘的檔案將自動被刪除，以避免儲存空間被耗盡。
  - 在 PC 使用模擬功能時，資料取樣檔案一律儲存在安裝目錄下的「保存位置」下的 **datalog** 子目錄。路徑為：**C:\EBpro\ [保存位置]\datalog**。此時若要改變資料取樣的格式，須先刪除安裝目錄下的舊資料取樣記錄，以避免系統誤讀舊記錄。
  - 當儲存檔案至 USB 碟或 SD 卡時，一個 FAT32 資料夾可儲存的檔案數量取決於檔案的名稱長短，當檔名越長，則資料夾可儲存的檔案數量就越少。
  - LB-9034 的觸發間隔最快 2 秒可執行 1 次。也就是說，當觸發了 LB-9034 後，需等待 2 秒後再次觸發才有作用。
-  請點選此圖示觀看教學影片，請先確定已連上網路線。

**cMT、cMT X 系列** (cMT、cMT X 系列部分欄位的使用方式與 eMT、iE、XE、mTV、iP 系列相同，若無特別說明，請直接參閱即可)

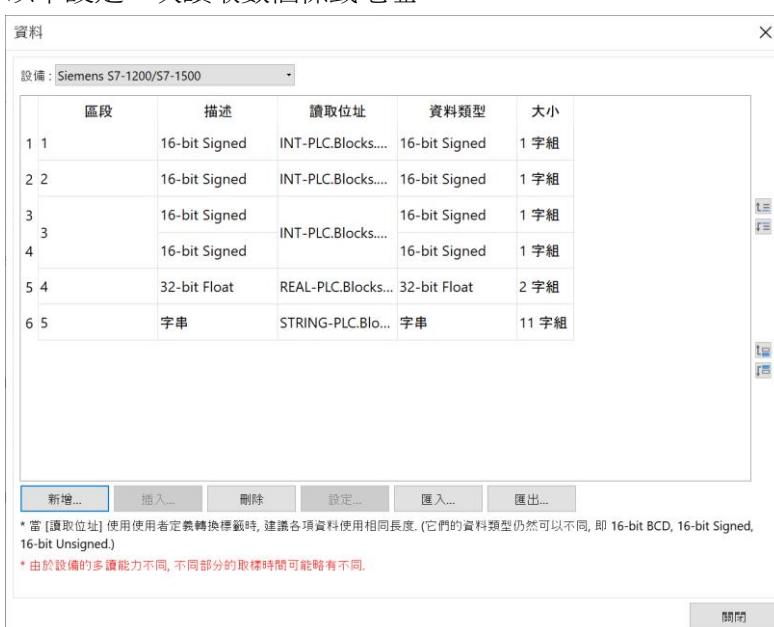


## 資料取樣

設定	描述
<b>數據來源</b>	可設定讀取多個不同格式的資料，上限為 1000 筆。在 [設定...] 中，可設定此資料取樣欲批次採樣的位址格式。依照 PLC 的特性，有些驅動可支援採樣非連續地址，但有些僅支援採樣連續地址。僅支援採樣連續位址的驅動，則只要設定位址的格式即可。此顯示格式可以直接套用在趨勢圖與歷史數據顯示。
<b>採樣連續位址</b> (勾選 [下一個連續位址] 選項)	僅須設定後續的地址的格式，這些地址會在同一個區段中。以下設定一次讀取 LW-0~LW-17。

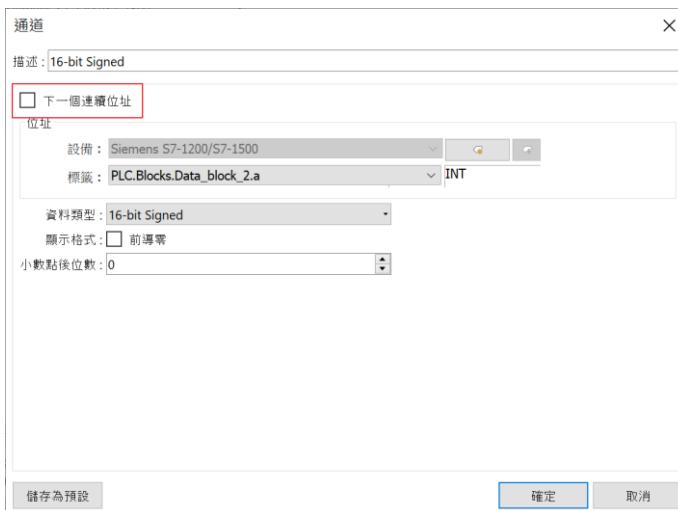
**採樣非連續位址** (不勾選 [下一個連續位址] 選項)

以下設定一次讀取數個標籤地址。



點擊 [新增]，取消勾選 [下一個連續位址]，接著就可以在地址的部

分選擇其他地址。



備註：標籤形式的驅動程式使用連續地址時，僅支援一維線性陣列。

#### 控制位址

對控制位址輸入指令數值時，即對歷史資料發送特定的命令。

數值	命令
1	清除 HMI 裡的資料
2	將資料同步到外接儲存裝置
3	先將資料同步到外接儲存裝置，接著清除 HMI 裡的資料
4	在更換 HMI 後，延用原先儲存在 USB 碟、SD 卡或資料庫上的歷史資料的功能
5	當 HMI 儲存空間不足時會停止資料取樣，在刪除不需要的歷史記錄並騰出空間後，使用此控制命令即可恢復資料取樣
6	釋放該資料取樣的歷史資料 (*.db) 未用到的空間。當歷史資料選擇儲存到 HMI 記憶體 (直到空間存滿) 時，可使用此命令。
7	刪除最舊的資料取樣的歷史資料 (*.db)。

#### 狀態位址

若控制位址是 LW-n，則控制位址的連續暫存器 (LW-n+1 ~ LW-n+4) 用

#### 錯誤位址

來表示歷史資料狀態及錯誤訊息。詳細位址對應請參考資料取樣的提示。



數值	狀態位址：LW-n+1 及 LW-n+3
0	未連接外部裝置或資料庫

1	正在連接外部裝置或資料庫
2	已連接外部裝置或資料庫
3	正在將資料取樣儲存到外部裝置或資料庫，完成後會回復至數值 2
數值	錯誤位址：LW-n+2 及 LW-n+4
0	沒有錯誤
1	未知的錯誤
2	連接外部裝置或資料庫失敗
3	操作權限不足
4	錯誤的資料庫名稱
5	資料格式不一致
6	無法連接資料庫的表格
7	無法建立表格
8	無法寫入表格
9	無法開啟資料庫
10	資料庫損毀
11	內部儲存的資料與外部裝置儲存的資料的 Row ID 衝突

**歷史記錄**

歷史資料儲存的路徑，可選擇儲存至 HMI 記憶體(10000 限制)、HMI 記憶體(直到空間存滿)、USB 碟、SD 卡，另外可啟用同步儲存歷史資料檔案至資料庫伺服器。

- 當選擇為 HMI 記憶體(10000 限制)，cMT 的資料取樣至 10000 筆時，會自動刪除最早的 1000 筆資料後繼續儲存資料。
- 如果選擇 HMI 記憶體(直到空間存滿)，系統會持續將資料存進人機的記憶體內，但無法同步至資料庫伺服器。cMT 的資料取樣至 HMI 記憶體滿時，會自動刪除最早的 1000 筆資料後繼續儲存資料。
- 當選擇為 USB 碟與 SD 卡，cMT 的資料取樣至 10000 筆時，會自動將取樣資料儲存至指定的外接儲存裝置，刪除 HMI 裡最早的 1000 筆資料。
- 當選擇使用同步至資料庫伺服器時，須選擇預先設定的資料庫。

 同步規則請見《8.4 外接儲存裝置同步 cMT Viewer 資料》。

 資料庫伺服器使用說明請見《13.44 資料庫伺服器》。

**自訂檔案  
管理**

提供自行設定資料取樣檔案名稱 (\*.db) 與檔案劃分的方式。當前正在取樣的檔案皆保存在 HMI 記憶體內。若採用同步至 USB 碟/SD 卡，請注意以下兩點：

1. 當儲存資料取樣的 db 檔案名稱改變時，舊 db 檔會被同步至 USB 碟/SD 卡。
  2. 當資料取樣產生新檔案時，USB 碟/SD 卡若未接上 HMI，則舊的資料取樣檔案會直接被刪除。例如：目前資料取樣的資料是儲存在 20161218.db 的檔案中，若此時產生 20161219.db 的檔案且 USB 碟/SD 卡未接上 HMI，則 20161218.db 的檔案會直接被刪除，不會保留。
- 另外，當使用自訂檔案管理時，無法啟用同步至資料庫伺服器功能。

## 資料取樣

	 自訂檔案管理設定視窗請見《8.3.2 自訂檔案管理》。
週期性自動同步	啟用後，HMI 不只會參照 [歷史資料] 的同步規則，資料也會依據設定的週期保存到外部裝置。請注意，週期性自動同步的計時器在使用者操作控制位址時會重置。  請見《外接儲存裝置同步 cMT Viewer 資料》。
檔案保留限制	使用 <b>[全部記錄於同一檔案]</b> 當同步位置為 SD 卡或 USB 碟時，檔案保存天數可設定為 1~65535 天。 使用 <b>[自訂檔案管理]</b> 當同步位置為 SD 卡或 USB 碟時，檔案保存數量可設定為 1~65535 個； 當同步位置為 HMI 記憶體時，檔案保存數量 1~1000 個。
歷史資料來源	讀取歷史資料時可選擇來源為 USB 碟、SD 卡，或是資料庫。

## 8.3.1. 自動停止選項範例

搭配不同物件，[自動停止] 產生的效用就不同，如下表：(假設最大資料設為 n)

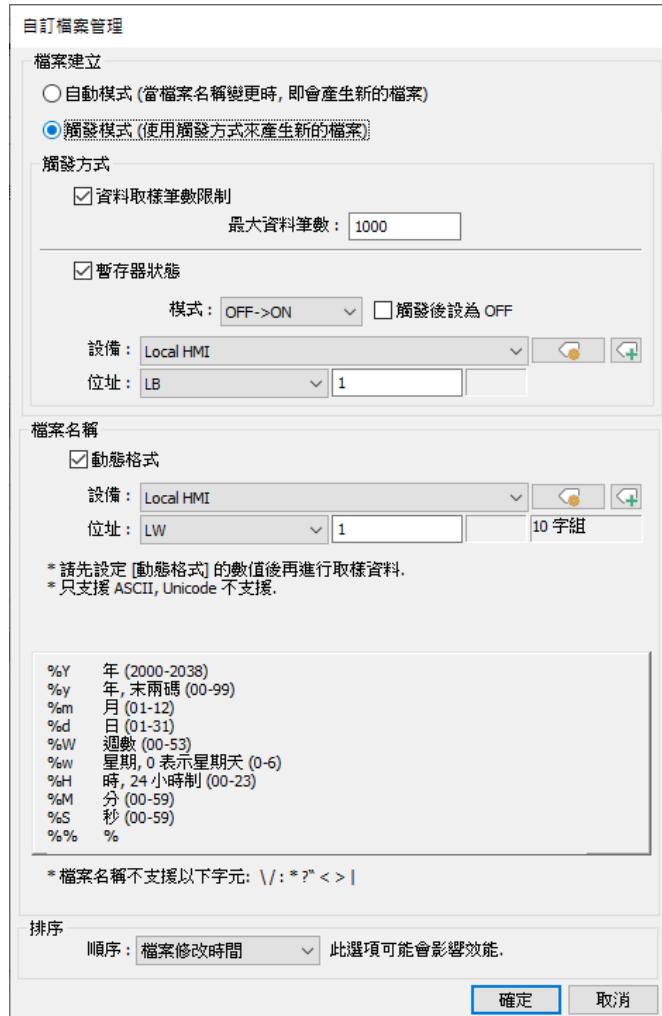
搭配物件	未勾選 [自動停止]	勾選 [自動停止]
趨勢圖-即時模式	將刪除較舊的取樣資料，並顯示剛獲得最新的 n 筆資料在趨勢圖上。請參考下列圖解。	至第 n 筆資料後停止動作。
趨勢圖-歷史模式	資料持續被取樣，並顯示所有歷史資料在趨勢圖上。	至第 n 筆資料後停止動作。
歷史資料顯示	資料持續被取樣，並顯示所有歷史資料在歷史資料顯示上。	至第 n 筆資料後停止動作。
資料取樣	持續取樣新記錄。	至第 n 筆資料後停止動作。

範例：設定資料長度數為 10 個，當第 11 個資料產生時，在未勾選 [自動停止] 的狀況下，最舊的資料記錄將會被刪除，並增加最新的記錄，如下圖所示。

Record Number	Data	Not selecting [Auto. stop]
1	101	102
2	102	103
3	103	104
4	104	105
5	105	106
6	106	107
7	107	108
8	108	109
9	109	110
10	110	111
11	111	

### 8.3.2. 自訂檔案管理

提供自行設定資料取樣檔案名稱 (\*.dtl, \*.db) 與檔案劃分的方式。



設定	描述
檔案建立	<b>自動模式</b> 當檔案名稱變更時，即會產生新的檔案。
觸發模式	<b>根據 [觸發方式] 的設定，產生新的檔案。</b>
觸發方式	<b>資料取樣筆數限制</b> 當取樣數量達到 [最大資料筆數] 的設定值時，即會產生新的檔案。 <b>暫存器狀態</b> 可指定一個位元位址的狀態變化來產生新的檔案。當其狀態符合 [模式] 的設定時，即會產生新的檔案。 <b>觸發後設為 ON/OFF</b> 當指定的位元位址的狀態變化，並產生新的檔案後，回復成原狀態。 <b>重新啟動 HMI 後，僅在滿足觸發條件後才生成並寫入新檔案</b>

若 HMI 重啟時，沒有滿足觸發條件產生新檔案，新的資料會寫入重啟前的最後一個檔案內。下載新專案至 HMI 視為重啟。

<b>檔案名稱</b>	可使用預設的控制碼、英數字及部分半形符號。
<b>動態格式</b>	可指定一組字組位址來設定檔案的名稱，亦可輸入時間方塊的控制碼來帶入系統時間。 請注意，字組長度為 1 ~ 25，且不支援的半形符號有 \/: * ? < >   。
<b>排序</b>	檔案排序會影響系統認定新/舊檔案的方式，進而影響保留檔案限制及備份檔案時的邏輯。
<b>檔案名稱</b>	檔案排序方式以檔名排序遞減。
<b>檔案修改時間</b>	檔案排序方式依照內容實際時間排序。(因會需要檢查檔案內的內容，此選項可能會影響效能) 舉例： 當檔案產生的順序為 filename_05、filename_04、filename_03、filename_02、filename_01 時，也就是說 filename_05 為最舊的檔案，filename_01 為最新的檔案，且檔案保留限制為 5 個檔案，在新增第六個檔案時： 選擇檔案名稱排序：系統會認定 filename_05 為較新產生的檔案，刪除 filename_01 以維持檔案數量為五個。 選擇檔案修改時間排序：系統會認定 filename_01 為較新產生的檔案，刪除 filename_05 以維持檔案數量為五個。 請注意檔案命名： 當使用檔案名稱排序時，檔名若為 filename_1、filename_2 及 filename_10，系統認定檔案的新舊排序為 filename_2 為最新，filename_10 次之，filename_1 為最舊；若需要系統認定 filename_10 為最新的檔案，則需將檔名補 0，也就是 filename_01、filename_02 及 filename_10 。

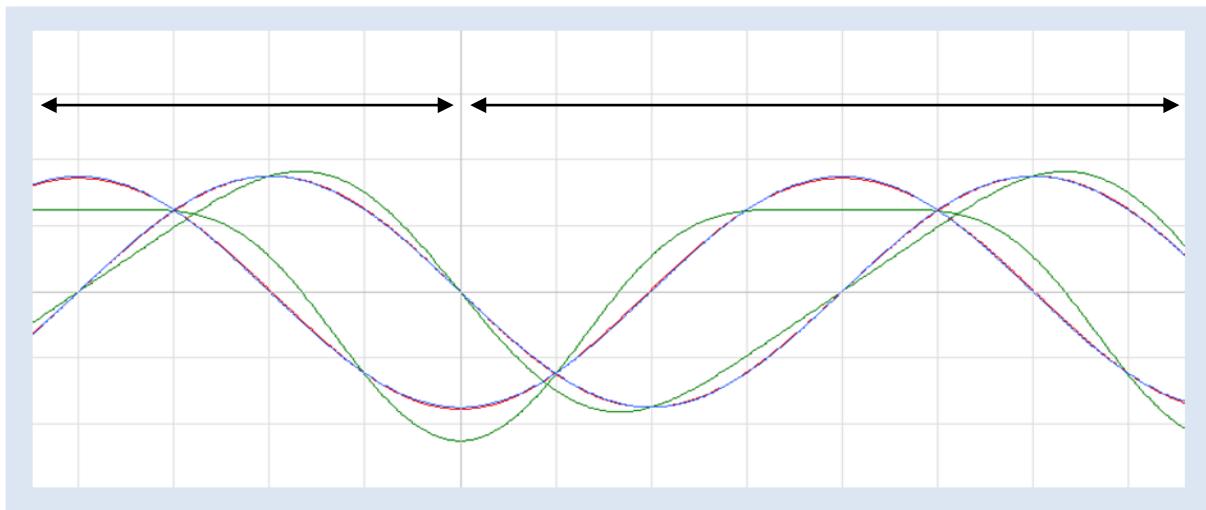
### Note

- 當數據來源使用採樣非連續位址，區段的上限為 8 組。
- 當檔案名稱使用 [動態格式] 且勾選 [資料取樣筆數限制] 來觸發產生新的檔案，則在 HMI 開機前，動態格式的來源暫存器就需先填入名稱，否則將一直無法達至指定取樣筆數，也不會產生取樣檔案。
- 當產生新的檔案時，系統會先偵測新檔案的檔案名稱是否存在。若已存在，則會將取樣資料直接附加於存在的檔案中。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 8.4. 外部裝置或資料庫同步 cMT Viewer 資料



以往資料取樣的資料在 [趨勢圖] 物件上顯示時，必須設定是即時模式或是歷史模式，兩者模式不可共存於同一個趨勢圖物件上。cMT / cMT X 系列將歷史模式與即時模式結合，達到無縫連接，讓所有資料取樣可在同一個 [趨勢圖] 物件或 [歷史數據顯示] 物件上顯示，畫面也會自動的更新。另外可透過外接儲存裝置將資料與之同步。

同步規則：

1. 每當取樣的資料達至 10000 筆時，HMI 便會自動將資料寫入至外接儲存裝置，接著刪除 HMI 裡最早期的 1000 筆資料。
2. 若將外接儲存裝置從 HMI 上移除，之後在新產生的取樣的資料不超過 9000 筆時就插回 HMI，在外接儲存裝置移除的期間，取樣資料仍然會保存在 HMI 裡不會被清除。若在外部裝置移除的期間，取樣的資料已經超過 9000 筆，較舊的資料就會被刪除，即使之後將外接儲存裝置插回也無法同步到被刪除的資料。
3. 若外接儲存裝置內已經有取樣資料，則每次同步時，新的取樣資料將會附加在原有的資料。

## 8.5. 查看 cMT Viewer 特定日期或特定檔案的歷史資料

若想查看歷史的資料，請參照以下步驟（以趨勢圖為例）：

1. 點選趨勢圖右上方的  按鈕。
2. 跳出趨勢圖設定對話窗。

取消	選項	完成
起始日期	<input checked="" type="button"/>	
結束日期	<input checked="" type="button"/>	
趨勢圖設定		
通道顯示控制		
Y 軸	關	
	預設值	

取消	選項	完成
檔案選擇		
(當前)	<input checked="" type="checkbox"/>	
20161219_140822		
20161219_140812		
20161219_140802		
20161219_140750		
20161219_140740		
趨勢圖設定		
通道顯示控制		
Y 軸	關	
	預設值	

3. 指定欲查詢的日期或檔案。

取消	選項	完成
起始日期	<input checked="" type="button"/>	
2016-12-19		
結束日期	<input checked="" type="button"/>	
2016-12-19		

取消	選項	完成
檔案選擇		
(當前)		
20161219_141014		
20161219_141004	<input checked="" type="checkbox"/>	
20161219_140956		
20161219_140946		
20161219_140936		

4. 按下 [完成] 完成設定。

# 9. 物件一般屬性

本章節說明一個物件的基本設定。

9.1.	概要 .....	9-2
9.2.	選擇 PLC 並設定讀寫位址 .....	9-2
9.3.	使用向量圖庫與圖片庫 .....	9-3
9.4.	設定標籤內容 .....	9-7
9.5.	輪廓調整 .....	9-9
9.6.	重疊物件 (cMT / cMT X 系列支援) .....	9-10

## 9.1. 概要

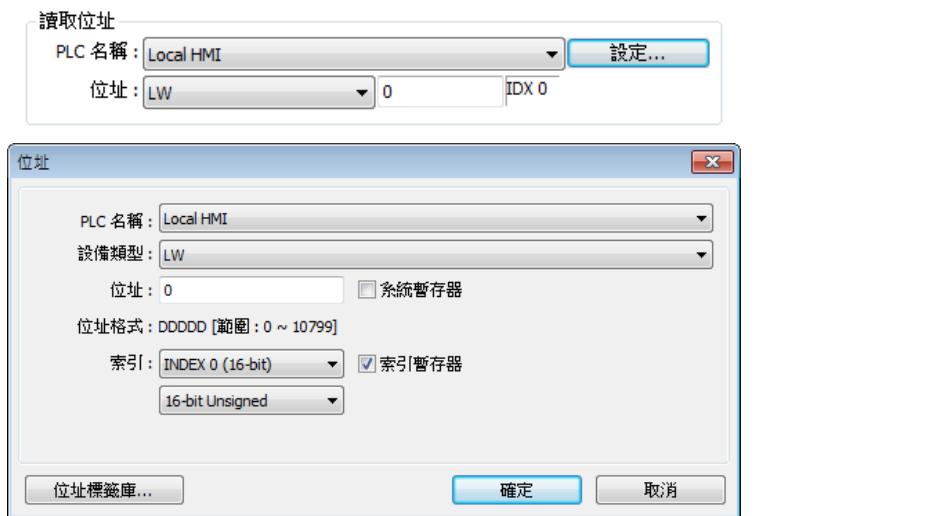
一般建立一個物件的步驟如下：

1. 選擇 PLC 裝置並設定讀寫位址。
2. 使用向量圖庫或圖片庫。
3. 設定標籤內容。
4. 調整輪廓。

本章將說明物件一般屬性的內容。

## 9.2. 選擇 PLC 並設定讀寫位址

某些物件的使用需選擇要操作的 PLC 對象。[PLC 名稱] 用來表示要控制的 PLC，這些 PLC 名稱來自 [系統參數] 中 [設備清單] 的內容。



設定	描述
PLC 名稱	選擇 PLC 的型號。
設備類型	選擇位址類型，當 PLC 型號不同時，將出現不同的位址類型。
位址	設定讀寫的位址。
系統暫存器	位址標籤包含 [系統暫存器] 與 [使用者定義標籤]。此項目用來選擇是否使用 [系統標籤]。系統標籤為系統保留作為特殊用途的位址，分為 bit 位址系統暫存器與 word 位址系統暫存器 (LB 或 LW)。在選擇使用 [系統暫存器] 後，除了 [設備類型] 將顯示系統暫存器的內容之外， [位址] 將顯示目前所選用的系統暫存器。
索引暫存器	選擇是否使用 [索引暫存器]。

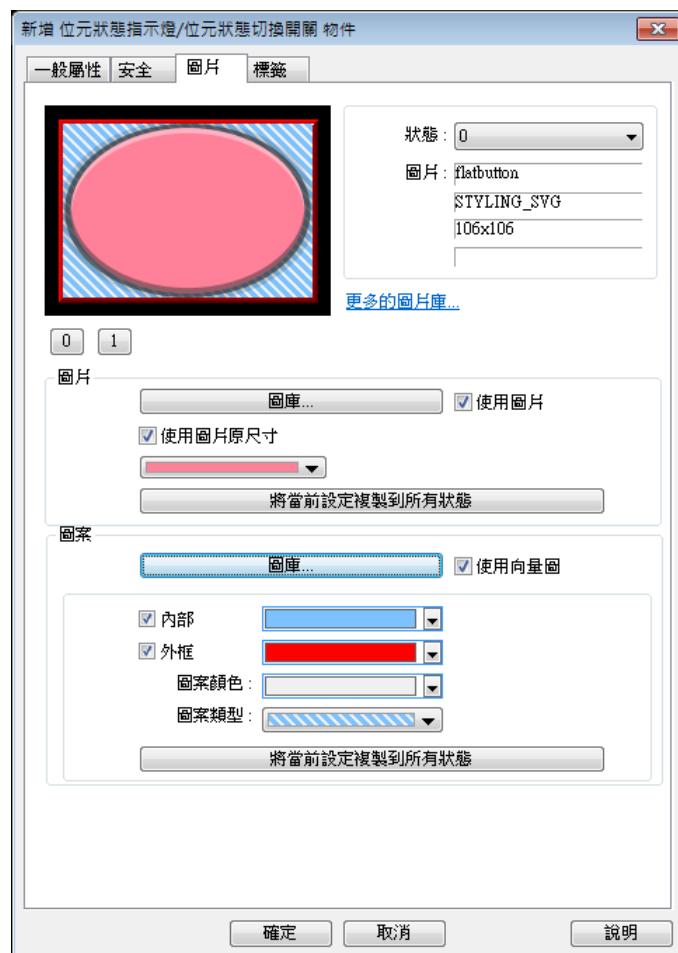
關於系統暫存器的詳細資訊請參考《22 位址暫存器》。

關於索引暫存器的詳細資訊請參考《11 索引暫存器》。

關於位址標籤庫的詳細資訊請參考《16 位址標籤庫》。

### 9.3. 使用向量圖庫與圖片庫

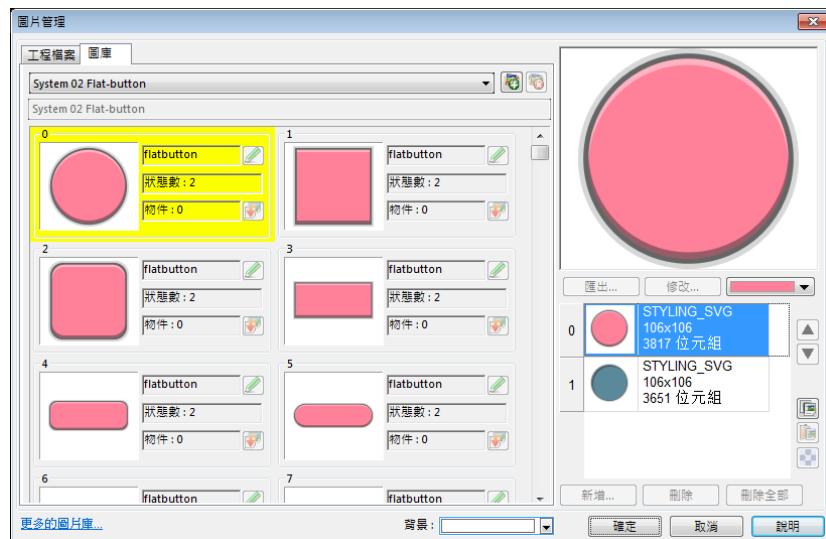
部分物件有 [圖片] 設定分頁可以使用 [向量圖庫] 與 [圖片庫] 的圖形，增加物件的視覺效果。向量圖庫與圖片庫的使用在物件屬性頁中的 [圖片] 分頁中設定，如下圖所示。



#### 9.3.1. 圖片

勾選 [使用圖片]，在按下 [圖庫] 按鈕後可以得到下面的 [圖片管理] 對話窗，可從中選擇圖片。

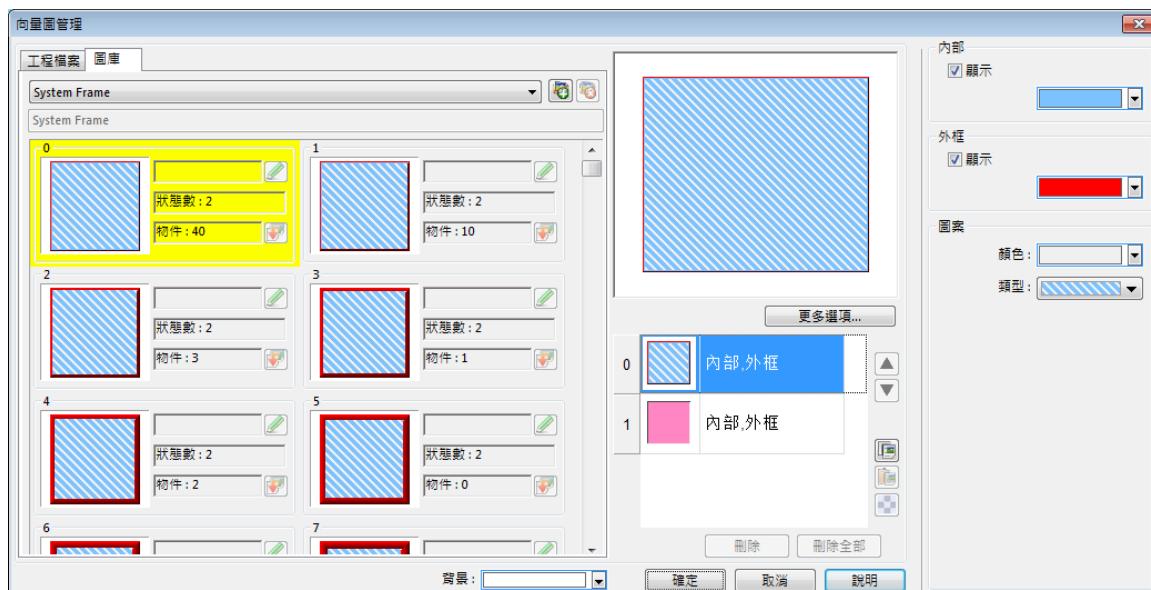
## 物件一般屬性



設定	描述
圖庫	點擊 [圖片] 欄位的 [圖庫] 可開啟圖片庫管理視窗。
使用圖片原尺寸	勾選後，EasyBuilder 將會將圖片調整為原始尺寸以提高效能。
將當前設定複製到所有狀態	使用系統圖片庫時才有此選項。點擊 [將當前設定複製到所有狀態] 後會將顏色設定套用至所有狀態。

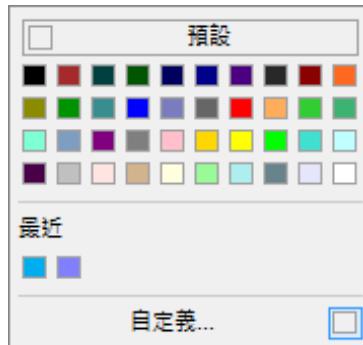
## 9.3.2. 圖案

勾選 [使用向量圖]，在按下 [圖庫] 按鈕後可以得到下面的 [向量圖管理] 對話窗，可從中選擇向量圖。



設定	描述
內部	選擇是否使用圖案的內部底色，按下顏色設定鈕後所出現的

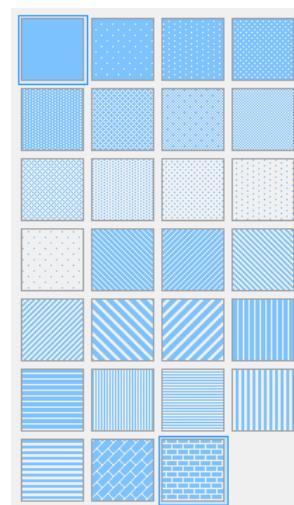
[色彩] 對話窗，可用來設定顏色。



**外框** 選擇是否使用圖案的外框，按下顏色設定鈕後所出現的 [色彩] 對話窗，可用來設定外框的顏色。

**圖案顏色** 設定內部填充的顏色。

**圖案類型** 設定內部填充的圖案式樣。



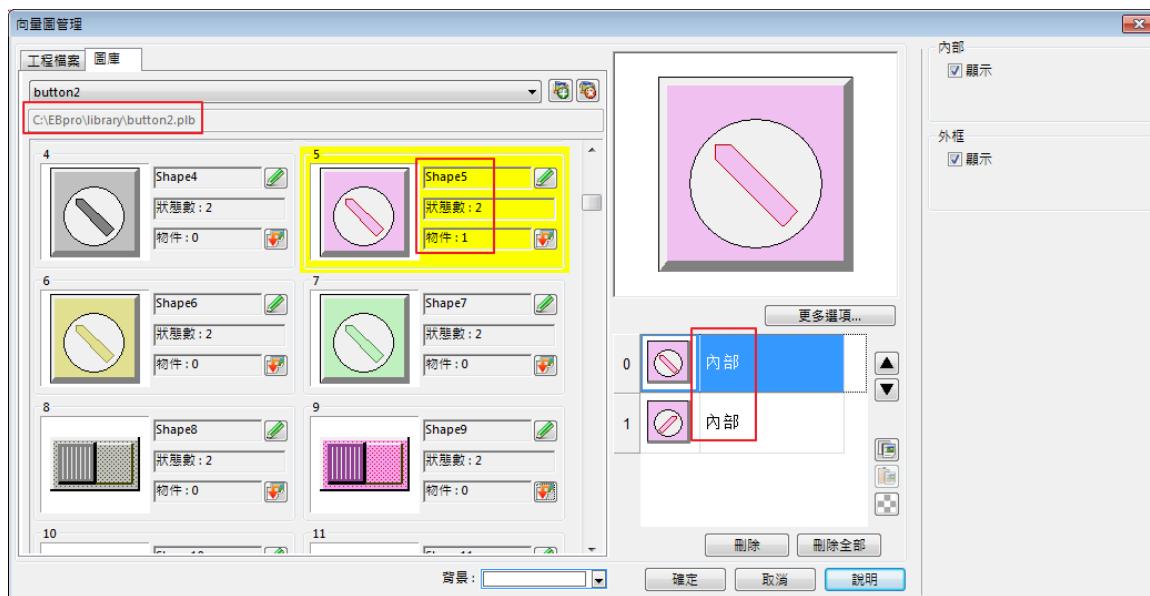
**將當前設定**

**複製到所有  
狀態** 將目前狀態各項屬性設置到其他狀態。

### 9.3.3. 圖片資訊

在 [向量圖] 或 [圖片] 管理視窗中，目前選擇的樣式會使用黃色背景來加以標示，此外，對話窗中亦包含以下的資訊。

## 物件一般屬性



路徑	表示目前圖庫檔案的來源路徑
Shape5	表示此向量圖的名稱
狀態數 : 2	此向量圖的狀態個數
物件 : 1	表示此向量圖被 1 個物件所使用
內部、外部	此向量圖的狀態 0 與狀態 1 皆只具備內部，不具備外框。



關於向量圖庫與圖片庫詳細資訊請參考《14 向量圖庫與圖形庫的建立》。

## 9.4. 設定標籤內容



### 設定

### 描述

**使用文字標籤**

勾選此選項物件才允許使用文字標籤。

**新增至文字標籤庫**

快速新增物件的標籤內容(包含各狀態)至文字標籤庫中。

**使用文字標籤庫**

勾選此選項表示文字內容將來自文字標籤庫。

**文字標籤庫**

檢視文字庫的內容



詳細資訊請參考《15 文字標籤庫與多國語言使用》。

### 內容

設定文字內容。如使用 **【文字標籤庫】**，此項內容將來自文字標籤庫。

**標籤位置連動**

勾選此選項時，移動某個狀態的文字將連帶移動其他狀態的文字。

**複製文字到所有的狀態**

將當前的文字內容複製到其他所有的狀態。

## 物件一般屬性

字型	選擇文字所使用的字型。EasyBuilder Pro 支援 WINDOWS 的 true-font 字型。
尺寸	選擇文字的大小。
顏色	選擇文字的顏色。
閃爍	選擇文字閃爍方式，可選擇不閃爍 [無]，或閃爍時間間隔為 [1 秒] 或 [0.5 秒] 的閃爍方式。
斜體	使用斜體字體。 <i>Italic Label</i>
底線	文字加上底線。 <u>Underline Label</u>
文字對齊	文字對齊時的排列方式，等同於快捷鍵的對齊按鈕。
	   左對齊                  置中對齊                  右對齊 <b>111</b> <b>111</b> <b>111</b> <b>222222</b> <b>222222</b> <b>222222</b> <b>3333333333</b> <b>3333333333</b> <b>3333333333</b>
文字區域位置	文字在物件中的位置。水平與垂直對齊分別對應以下快捷鍵的按鈕。
	 
複製此屬性至每個狀態	將當前的屬性複製到其他所有的狀態。
更多...	<p><b>文字圖片化 (cMT/cMT X 系列不支援)</b>          勾選此選項，文字將使用 bitmap 圖形顯示。</p> <p><b>ON = OFF(使用狀態 0)</b>          顯示內容不會隨著狀態改變。</p> <p><b>跑馬燈</b>          移動方向 – 設定跑馬燈的效果並選擇文字的移動方向，可選擇 [不移動]、[向左]、[向右]、[向上]、[向下]。          持續移動 – 未勾選此選項，則文字需在全部消失後才出現後續的文字，如下圖。</p>
	
	有勾選此選項，則文字會連續出現，如下圖。



速度 – 選擇文字的移動速度。

### Note

- 若 [文字/註解] 物件使用 [註解] 功能，其支援文字內容僅作為註解使用，在 HMI 畫面上不會顯示。
- 若欲使用的字型，在編輯畫面與人機上的位置有垂直方向的偏移，可以透過 %appdata%\Weintek\EasyBuilder 資料夾中的 FontRules.xml 檔案進行字型的偏移調整。藉由新增該字型名稱於 FontRules.xml 檔案，即可事先調整特定字型的垂直指標(vertical metrics)，以避免顯示偏移的現象。

```
<?xml version="1.0"?>
<FontRules>
    <Version data="1.1"/>
    <Rule data="[\w]*algerian[.\w]*"/>
    <Rule data="[\w]*castellar[.\w]*"/>
    <Rule data="^Wingdings$"/>
    <Rule data="fix_vertical_metrics = ^Gotham Pro Black$"/>
    <Rule data="fix_vertical_metrics = ^Gotham Pro Bold$"/>
    <Rule data="fix_vertical_metrics = ^Gotham Pro$/>
</FontRules>
```

## 9.5. 輪廓調整

將物件放置於編輯視窗後，再點擊物件可啟用 [輪廓] 設定頁來調整物件的外型大小。



設定	描述
位置	<b>圖釘</b> 鎖定設定，勾選此選項後將無法改變物件的位置與大小。 <b>[X]、[Y]</b> 此座標為物件放置於編輯視窗中的位置。
尺寸	設定物件的 [寬度] 和 [高度]。

## 9.6. 重疊物件 (cMT / cMT X 系列支援)

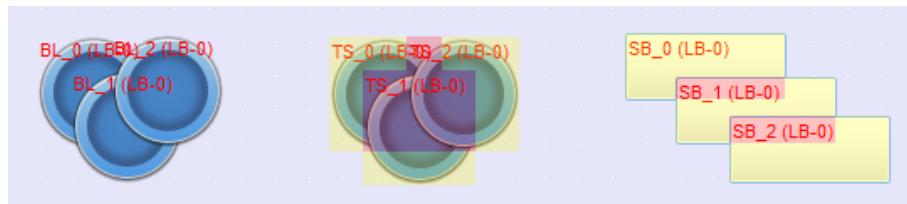
當有多個物件重疊在一起時，可以透過轉換至複合式多功能物件將多個物件整合成單一物件。以下將說明如何尋找重疊物件與轉換方法。該複合式多功能物件的一般屬性將以最上層的物件為主。

### 9.6.1. 啟用觸控區域

在 [檢視] 底下的 [顯示] 部分勾選 [觸控區域]。



可以觸控的物件將會以黃色區域顯示，重疊的物件區域則會額外用粉紅色來顯示，無法觸控的物件則會維持原樣。

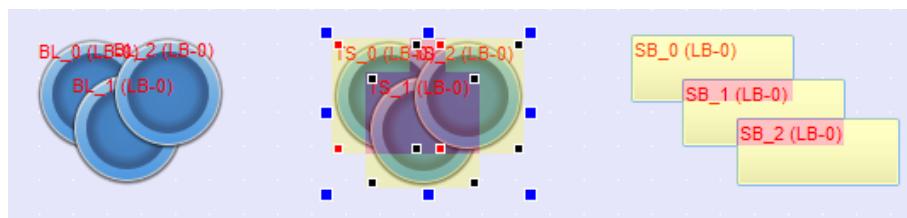


### 9.6.2. 尋找重疊物件

[工具]下[升級]部分的[尋找重疊物件]可以協助尋找在工程檔案中所有的可觸控重疊物件。



第一次使用將會從該視窗開始尋找第一組重疊物件，再次使用則會尋找下一組重疊物件。若該視窗已無重疊物件，將會自動往後面的視窗尋找，若有發現則會自動開啟。



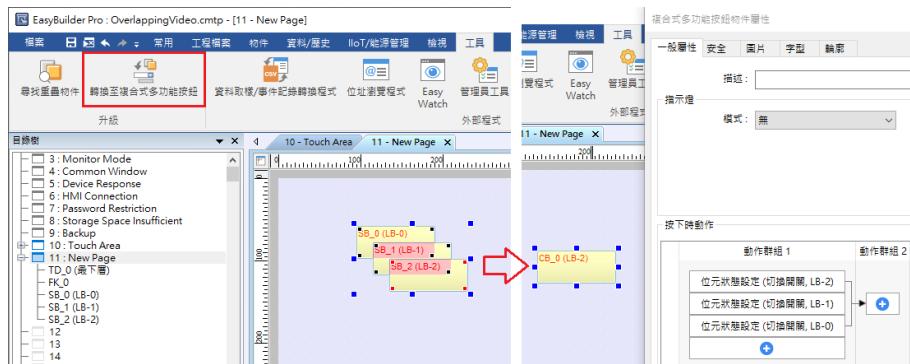
#### Note

- 要執行 [尋找重疊物件]，視窗所有的視窗階層要為可編輯狀態。

### 9.6.3. 轉換至複合式多功能按鈕

透過前面兩個方法找到的重疊物件，若皆是位元狀態設定、位元狀態切換開關、多狀態設定、多狀態切換開關與功能鍵的物件，則可以使用此功能將多個物件合成為一個按鈕物件，有兩種操作方法如下。

1. [工具] 下 [升級] 部分，點選轉換至複合式多功能按鈕。

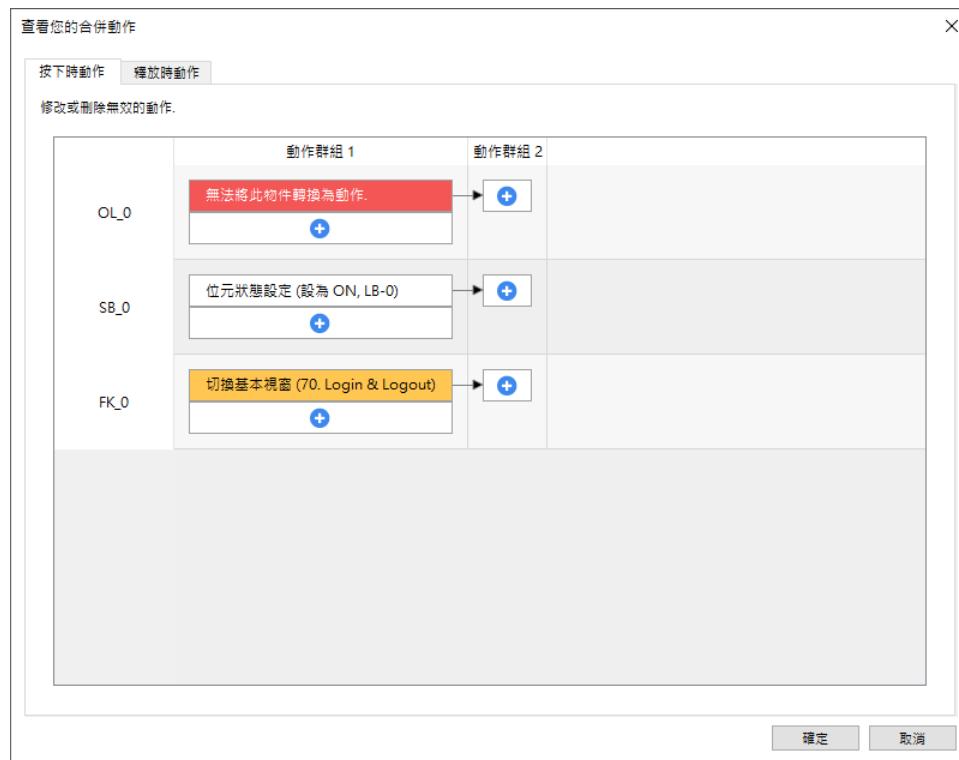


2. 全選重疊物件後按右鍵，選擇轉換至複合式多功能按鈕。



當確認執行之後，將會彈出一個合併動作確認視窗。可在此查看並且編輯各個物件轉換後的動作，若背景沒有任何顏色，表示此物件可以合併；若背景為黃色，則表示此物件無法直接合併，將自動調整成可以合併的條件；若背景為紅色，則表示此物件無法執行合併。出現黃色或紅色背景的物件皆可將滑鼠移動至上方查看錯誤原因。

## 物件一般屬性


 Note

- 新產生的複合式多功能物件的描述，將會使用由最上層往下排列第一個有描述的物件。
- 新產生的複合式多功能物件的安全設定，將會使用最上層物件設定。後面物件若碰到設定不同，合併完成的記錄會有警示。
- 新產生的複合式多功能物件的聲音設定，將會使用由最上層往下排列第一個有聲音設定的物件。後面物件若碰到設定不同，合併完成的記錄會有警示。
- 新產生的複合式多功能物件的外觀設定，將會使用由最上層往下排列第一個有使用圖片或向量圖的外觀與文字標籤設定。若全部物件都沒有使用圖片或向量圖，則複合式多功能物件會使用能被轉換的最下層物件的文字標籤。
- 任何物件若有設定震動，合併後的新物件會啟用震動。
- 當物件有使用操作記錄，合併後的新物件會沿用設定。若有多個物件皆使用操作記錄，合併完成的記錄會有警示。

# 10. 使用者密碼與物件安全防護

本章節說明如何在 EasyBuilder Pro 設定使用者的安全等級與密碼。

10.1. 概要 .....	10-2
10.2. 使用者密碼與可操作物件類別設定 .....	10-2
10.3. 進階安全模式之控制位址 .....	10-6
10.4. 進階安全模式之應用 .....	10-10
10.5. 物件安全設定頁 .....	10-25
10.6. 物件安全防護範例 .....	10-28
10.7. 工程檔案之使用者密碼編輯保護 .....	10-31
10.8. 批次修改物件安全屬性 .....	10-32

## 10.1. 概要

在 EasyBuilder Pro 設定使用者的安全等級與密碼，提供以下模式：

- 一般模式
- 進階安全模式

另外，cMT / cMT X 系列提供使用 LDAP 進行使用者登入。

物件安全防護須完成以下兩項設定：

1. 使用者密碼與可操作物件類別設定
2. 物件操作安全防護

一個物件只能屬於一個安全等級，或將安全等級設定為“無”，使得任何人皆可操作該物件。

## 10.2. 使用者密碼與可操作物件類別設定

在 [系統參數設定] » [使用者密碼] 頁籤可設定相關參數。

### 10.2.1. 一般模式

一般模式可設定最多 12 個使用者，各別設定不同的密碼，密碼需為非負整數，並規劃每個使用者可操作的物件類別分為“A ~ F”等共 6 個類別。

HMI 運作時，使用者在成功輸入密碼後，系統會依照設定內容決定使用者可以操作的物件類別。如下圖，“使用者 1”只被允許操作物件類別為“A、B、C”。



- cMT / cMT X 系列不支援使用者密碼之一般模式。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 10.2.2. 進階安全模式

進階安全模式可設定的使用者數為 11 組，另外提供 [管理員] 使用模式，此管理員有最大使用權限，任何物件的安全等級皆可操作。使用者名稱可由中文、英文以及數字組成，不同的使用者密碼可由英文或數字所組成，並可設定每個使用者可操作的物件類別分為 “A ~ L” 等共 12 個類別。(使用管理員工具，可設定最多 127 組使用者。此功能介紹於《10.4 進階安全模式之應用》)

進階安全模式提供一組 [控制位址] 機制，供使用者登入和管理帳號，請參考《10.3 進階安全模式之控制位址》的說明。或者可使用 USB 安全金鑰自動登入，當插入的 USB 碟裝置含有 USB 安全金鑰時，將自動登入指定的帳號，請參考《10.4.3 使用 USB 安全金鑰登入 / 登出》的說明。亦可使用指紋或 RFID 自動登入，當指紋辨識或是 RFID 掃描成功後，將自動登入所綁定的帳號，請參考《10.4.7 使用插件登入 / 登出》的說明。



#### Note

- 當使用 cMT / cMT X 系列搭配進階安全模式時，控制位址的字組位址來源只能為 Local HMI。使用 LW 做為控制位址時，僅 HMI 本機上可進行登入，cMT Viewer 上無法進行使用者登入。
- eMT、iE、XE、mTV 系列機型若有使用 EasyAccess 2.0，請注意 EasyAccess 2.0 的 HMI Viewer 無

法支援進階安全功能，請改用 VNC Viewer。

- 當使用者名稱含有中文時，僅支援使用 EasyBuider Pro 編輯使用者名稱，無法透過 EasyWeb 或是系統設定修改。

 請點選此圖示下載範例程式。此範例程式說明更多關於進階安全模式的使用方式。下載範例程式前，請先確定已連上網路線。

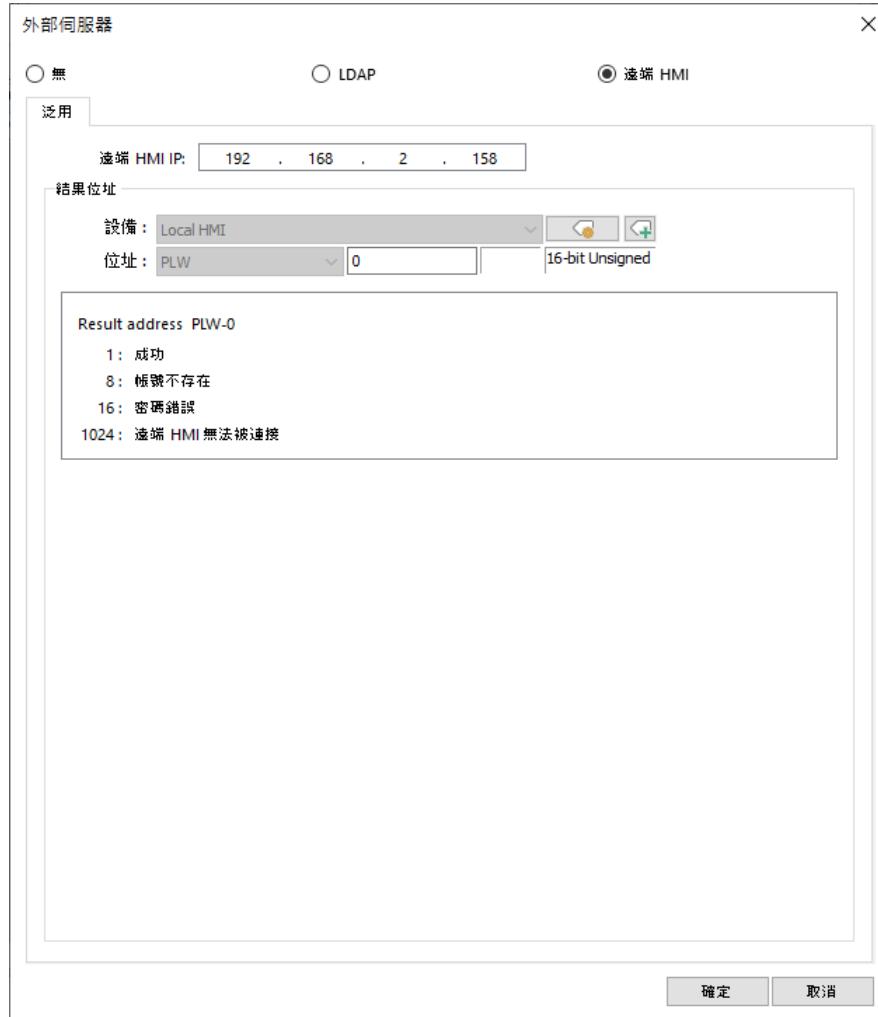
### 10.2.3. LDAP

LDAP (Lightweight Directory Access Protocol)，中文為輕量目錄存取協定，可用於存取目錄服務。目錄服務提供以類似資料庫的架構集中儲存資訊，在此主要用於帳號整合及驗證，以集中管理使用者帳戶。使用 LDAP 模式時，帳號管理不必需要在 HMI (EBPro) 端進行，而是可交由目錄服務伺服器管理，而 HMI 則是透過 LDAP 協定與伺服器進行帳號的認證。HMI 上僅需設定目錄服務伺服器的位置、組織架構等必要資訊且設定各群組可使用的權限類別，無須管理個別使用者的帳號密碼。



#### 10.2.4. 遠端 HMI

使用 [遠端 HMI] 模式時，帳號管理可不需要在本機 HMI 端進行，而是交由遠端 HMI 管理。本機 HMI 可透過 [遠端 HMI] 上的帳號密碼進行登入，無須自行管理使用者的帳號密碼。



### 10.3. 進階安全模式之控制位址

控制位址可用於登入和管理帳號，並使用連續 20 個位址作為參數。當選擇 cMT/cMT X series，系統將提供 LW 與 PLW 暫存器供挑選。LW 與 PLW 不同處在於 LW 是指 HMI 本機上的位址，而 PLW 則是指 Client 端 (例如: cMT-iV5、cMT-iV6、iOS、Android 設備) 上操作這些功能的位址。因每台 cMT/cMT X series 可供多台 Client 端連接，因此以上這些功能的系統暫存器將由 Client 端個別運行。使用 [控制位址] 執行登入時，需選擇 [使用者名稱] 或 [使用者索引] 其中一種方式登入。[使用者名稱] 及 [密碼] 需先至 [系統參數設定] » [使用者密碼] » [進階安全模式] 設定。

#### 10.3.1. 控制位址使用說明

當控制位址設為 LW/PLW-n 時，n 為任一數字，則將使用以下位址：

控制位址	標籤名稱	描述
LW/PLW-n (佔 1 個字組)	命令	控制各項操作命令 (例如: 登入，登出，新增/修改/刪除帳號...等等)。
LW/PLW-n + 1 (佔 1 個字組)	命令執行結果	顯示執行命令的結果。
LW/PLW-n + 2 (佔 1 個字組)	使用者索引	帳號索引 (配合項目選單物件使用)。
LW/PLW-n + 3 (佔 1 個字組)	使用者權限	權限值 (Level A = bit0, Level B = bit1...等等)。
LW/PLW-n + 4 (佔 8 個字組)	使用者名稱	帳號名稱 (可為中文、英文或數字，大小寫視為不同)。
LW/PLW-n + 12 (佔 8 個字組)	密碼	帳號密碼 (可為英文或數字符號，大小寫視為不同)。

當控制位址設定後，可至 [位址標籤庫] » [使用者定義標籤] 查詢其他相關功能的位址。

例如，當 [控制位址] 為 LW/PLW-0 時，則

LW/PLW-0 → [命令]

LW/PLW-1 → [命令執行結果]

LW/PLW-2 → [使用者索引]

LW/PLW-3 → [使用者權限]

LW/PLW-4 ~ LW/PLW-11 → [使用者名稱]

LW/PLW-12 ~ LW/PLW-20 → [密碼]

#### Note

- 當使用 cMT / cMT X 系列搭配進階安全模式時，控制位址的字組位址來源只能為 Local HMI。使用 LW 做為控制位址時，僅 HMI 本機上可進行登入，cMT Viewer 上無法進行使用者登入。
- eMT、iE、XE、mTV 機型若有使用 EasyAccess 2.0，請注意 EasyAccess 2.0 的 HMI Viewer 無法支

援進階安全功能，請改用 VNC Viewer。

### 10.3.2. 命令功能說明

當在 [命令] LW-n 輸入特定 [數值] 時,-> 可操作的功能如下：

設定數值	命令	搭配位址
1	使用名稱登入帳號	需先定義 [使用者名稱] 和 [密碼]，輸入名稱及密碼後，系統會去比對是否和 [系統參數設定] » [使用者密碼] » [進階安全模式] 設定相符。
2	使用索引登入帳號	需先定義 [使用者索引] 和 [密碼]，請參考《10.4.4 進階安全模式搭配項目選單物件》說明。
3	登出帳號	
4	更改目前已登入帳號的密碼	需先定義 [使用者名稱] 和 [密碼]，其中 [使用者名稱] 需填入原密碼，[密碼] 填入新密碼。
5	新增帳號	需先定義 [使用者名稱]，[密碼] 和 [使用者權限]。
6	新增臨時帳號 (單位: 分鐘)	需先定義 [使用者名稱]，[密碼]，[使用者權限] 和 [使用者索引]，其中 [使用者索引] 用來指定有效時間 (單位:分鐘)，0 表示永久有效。
7	使用名稱刪除現有帳號	需先定義 [使用者名稱]。
8	使用索引刪除現有帳號	需先定義 [使用者索引]。
9	使用名稱設定現有帳號的權限	需先定義 [使用者名稱] 和 [使用者權限]。
10	使用索引設定現有帳號的權限	需先定義 [使用者索引] 和 [使用者權限]。
11	使用名稱設定現有帳號的密碼	需先定義 [使用者名稱] 和 [密碼]。
12	使用索引設定現有帳號的密碼	需先定義 [使用者索引] 和 [密碼]。
13	使用名稱讀取現有帳號的權限	需先定義 [使用者名稱]，若成功執行則將帳號權限顯示於 [使用者權限]。
14	使用索引讀取現有帳號的權限	需先定義 [使用者索引]，若成功執

		行則將帳號權限顯示於 [使用者權限]。
15	新增臨時帳號 (單位: 天數)	需先定義 [使用者名稱]、[密碼]、[使用者權限] 和 [使用者索引]，其中 [使用者索引] 用來指定有效天數，0 表示永久有效。
16	新增限期帳號 (單位: 分鐘)	需先定義 [使用者名稱]、[密碼]、[使用者權限] 和 [使用者索引]，其中 [使用者索引] 用來指定有效時間，且不得為 0。
17	新增限期帳號 (單位: 天數)	需先定義 [使用者名稱]、[密碼]、[使用者權限] 和 [使用者索引]，其中 [使用者索引] 用來指定有效天數，且不得為 0。
18	使用名稱讀取剩餘分鐘	需先定義 [使用者名稱]，若成功執行則將剩餘時間顯示於 [使用者索引]。
19	使用索引讀取剩餘分鐘	需先定義 [使用者索引]，若成功執行則將剩餘時間顯示於 [使用者索引]。
20	使用名稱讀取剩餘天數	需先定義 [使用者名稱]，若成功執行則將剩餘天數顯示於 [使用者索引]。
21	使用索引讀取剩餘天數	需先定義 [使用者索引]，若成功執行則將剩餘天數顯示於 [使用者索引]。

### Note

- 新增臨時帳號 / 限期帳號: 臨時帳號與限期帳號的差別在於臨時帳號不會被保存於系統裡，故 HMI 斷電後即失效，但臨時帳號與限期帳號在超出有效時間後都會自動被刪除。
- 刪除現有帳號: 不可刪除目前登入帳號。
- 離線模擬/連線模擬: 皆使用程式內的帳號設定，進行模擬時，使用者對帳號內容的修改不會保留到下一次模擬。
- admin: 被內定為管理員帳號，不可被刪除，且權限全開不得修改權限。
- 系統暫存器 PLW-10754: 顯示目前登入的使用者名稱 (只支援於 cMT / cMT X 系列)。
- [使用者權限] 無法用來顯示當前登入使用者的權限，需透過系統暫存器 LW-9222 來顯示。
- LDAP 模式不支援使用 [使用者索引] 方式登入。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

### 10.3.3. 結果輸出說明

每當執行命令後，系統自動將執行結果輸出值傳送到控制位址的  $LW-n + 1$  位址。下列結果輸出值為 16 進制數值：

結果輸出值	訊息
(0x001)	命令執行成功
(0x002)	錯誤命令
(0x004)	帳戶已存在 (發生於新增帳號時)
(0x008)	帳戶不存在
(0x010)	密碼錯誤
(0x020)	目前命令無法被執行
(0x040)	不合法的帳戶名稱
(0x080)	密碼含有無效字元
(0x100)	匯入資料無效
(0x200)	超出有效期限 (當使用 USB 安全金鑰登入時，有效期限可設定於管理員工具)

#### Note

- 使用者可將 [命令執行結果]  $LW-n + 1$  與輸出值的對應訊息預先建立於事件登錄物件裡，並藉由事件顯示物件直接顯示命令的執行結果。

## 10.4. 進階安全模式之應用

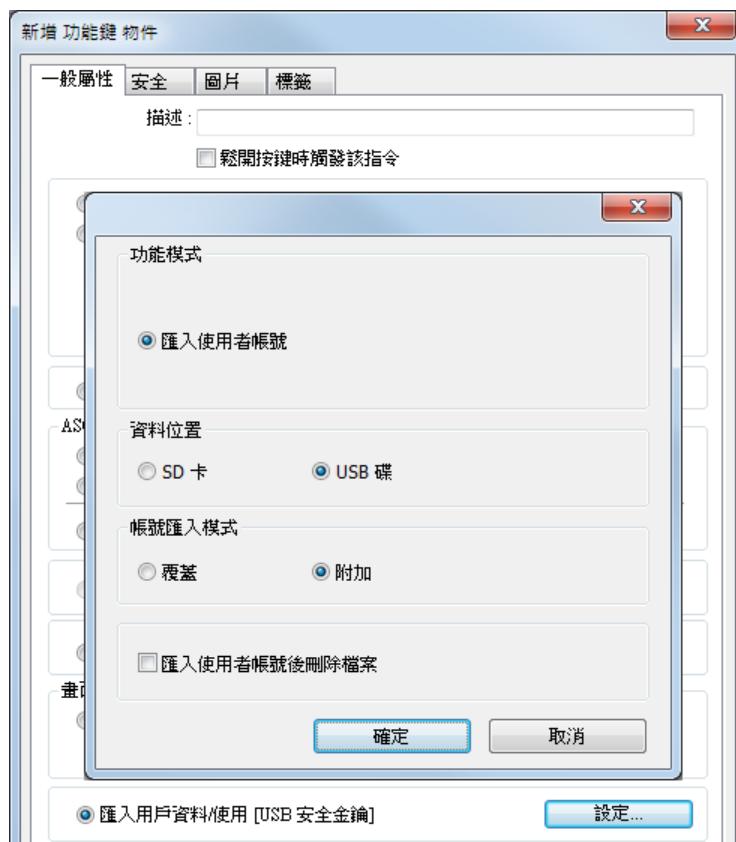
### 10.4.1. 匯入使用者帳號

除了 [系統參數設定] » [使用者密碼] 頁籤可以設定使用者帳號之外，從 EasyBuilder Pro 安裝目錄下開啟管理員工具後，勾選 [使用者帳號]，亦可進行使用者帳戶資料的設定。使用管理員工具最多可新增 127 筆帳戶資料。使用畫面如下：



關於管理員工具的詳細資訊請參考《36 管理員工具》。

新增帳號完畢後，可儲存至 USB 碟或 SD 卡，並預先在程式內建立 [功能鍵] » [匯入使用者資料]，設定如下：



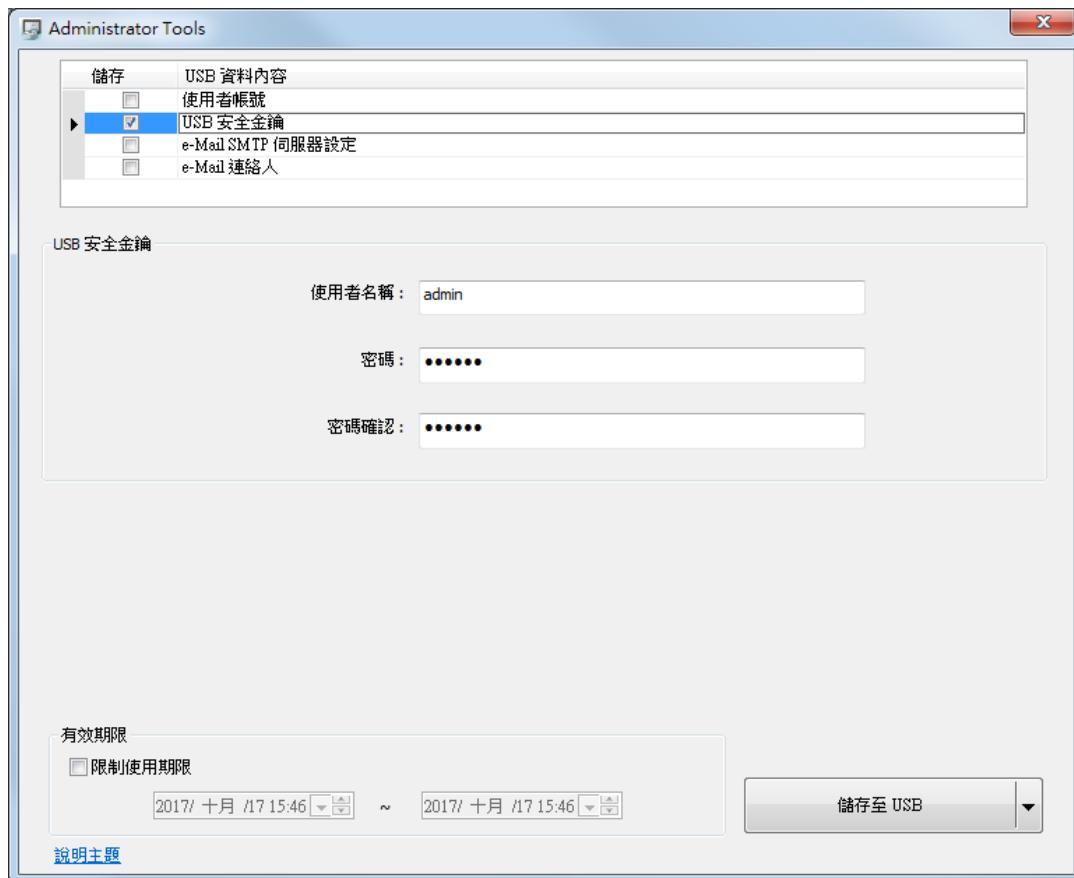
設定完成後，即可將外部裝置插入 HMI，並執行此功能鍵來匯入使用者帳號。

若帳號匯入方式選擇 [覆蓋]，則匯入前會先清除系統中所有帳戶，匯入完成後執行登出。若選擇 [匯入使用者帳號後刪除檔案]，在完成匯入帳號後系統將自動刪除儲存於外部裝置的匯入資料。若有在管理員工具 中指定 [有效期限]，則使用者只能在指定的期限內執行匯入動作。匯入的帳號並不會因為超過有效期限而被系統刪除。

#### 10.4.2. 使用 USB 安全金鑰登入

除了手動輸入帳號及密碼進行帳戶登入外，使用者亦可藉由一鍵登入來完成登入的動作，從 EasyBuilder Pro 安裝目錄下開啟管理員工具後，勾選 [USB 安全金鑰]，即可進行相關設定，透過預先設定好使用者的登錄資訊，即可利用 USB 安全金鑰直接登入帳戶，設定畫面如下：

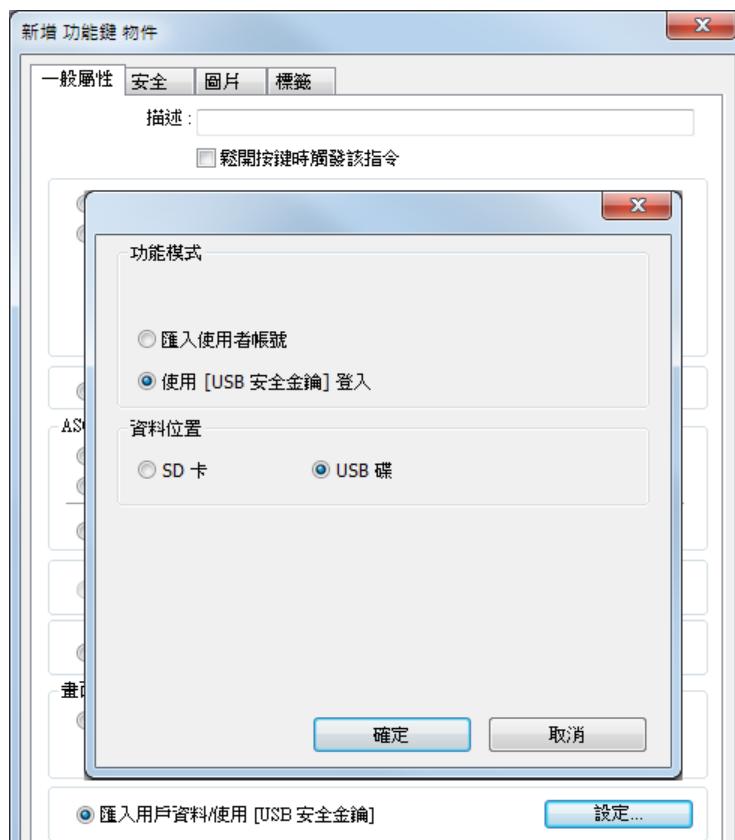
## 使用者密碼與物件安全防護



### Note

- USB 安全金鑰所欲設定的使用者帳號須為已存在於 HMI 程式上的使用者帳號。
- 👉 關於管理員工具的詳細資訊請參考《36 管理員工具》。

設定完 USB 安全金鑰後可儲存至 USB 碟或 SD 卡，在程式內建立 [功能鍵] » [使用 [USB 安全金鑰]]，設定如下：



設定完成後，即可將外部裝置插入 HMI，並執行此功能鍵來一鍵登入 USB 安全金鑰。若有在管理員工具中指定 [有效期限]，則使用者只能在指定的期限內執行登入動作，且系統會在金鑰超出有效期限後自動登出。

#### 10.4.3. 使用 USB 安全金鑰自動登入/登出

如下圖，在 [系統參數設定] » [使用者密碼] 頁籤勾選 [啟用] 便可啟用 USB 安全金鑰自動登入及登出：



啟用後，只要使用者插入含有 USB 安全金鑰的外部裝置，就可以直接使用設定的帳號登入。拔出裝置後，帳號將自動登出。登入/登出狀態會自動寫入至指定的 [狀態位址]，其數值代表的意義為：

- 0x00: 無動作
- 0x01: 登入成功
- 0x04: 登入失敗
- 0x08: 登出成功

0x10: 登出失敗

 設定 USB 安全金鑰的方法請參考《36 管理員工具》。

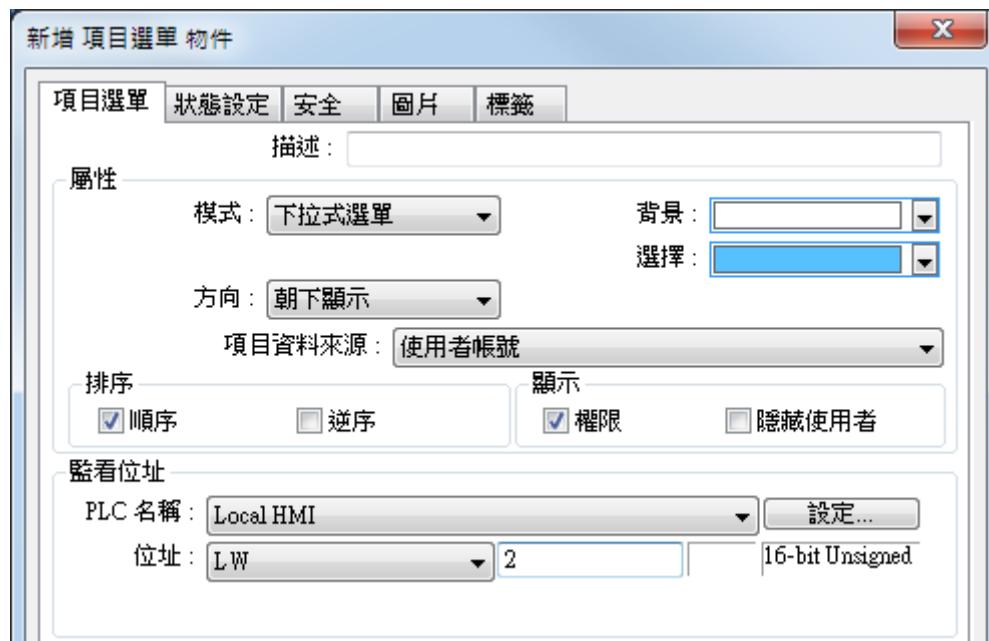
### Note

- 當啟用 [當 USB 安全金鑰插入 HMI 時自動執行登入/登出] 時，將無法使用 [功能鍵] 登入，但仍可使用控制位址登入及登出。
- 此功能無法在 PC 模擬模式下使用。
- 此功能只能使用儲存在 USB 碟中的 USB 安全金鑰。

 請點選此圖示下載範例程式。此範例程式說明如何使用 USB 安全金鑰自動登入/登出。下載範例程式前，請先確定已連上網路線。

#### 10.4.4. 進階安全模式搭配項目選單物件

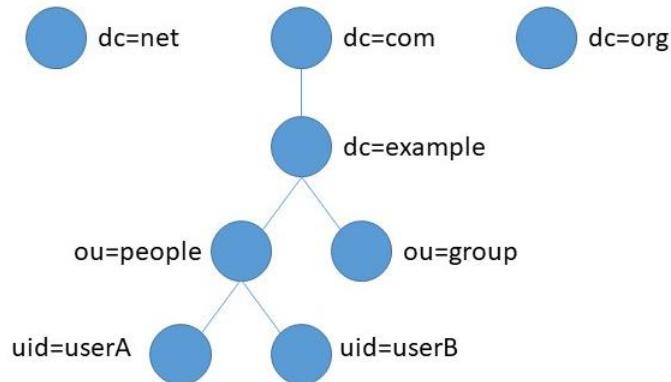
進階安全模式的控制位址 LW-n+2 為使用者索引，可搭配項目選單物件的使用者帳號來顯示帳號名稱和權限。使用者可以選擇是否要在項目選單物件上顯示帳號權限和隱藏使用者。隱藏使用者是指在 [系統參數設定] » [使用者密碼] » [進階安全模式] 下，可將使用者帳號名稱隱藏起來，增加安全性。假設進階安全模式的控制位址設定為 LW-0，則在此監看位址即為 LW-2。



#### 10.4.5. LDAP 模式

LDAP (Lightweight Directory Access Protocol)，中文為輕量目錄存取協定，可用於存取目錄服務。目錄服務提供以類似資料庫的架構集中儲存資訊，在此主要用於帳號整合及驗證，以集中管理使用者帳戶。使用 LDAP 模式時，帳號管理不必需要在 HMI(EBPro)端進行，而是可交由目錄服務伺服

器管理，而 HMI 則是透過 LDAP 協定與伺服器進行帳號的認證。HMI 上僅需設定目錄服務伺服器的位置、組織架構等必要資訊且設定各群組可使用的權限類別，無須管理個別使用者的帳號密碼。



LDAP 使用的控制位址與進階安全模式相同。控制位址的使用方法請參考《10.3. 進階安全模式之控制位址》。由於項目選單物件無法取得 LDAP 的使用者名稱，因此不支援使用 [索引|登入帳號]。

### Note

- 若多個群組都有相同的使用者，則使用者可擁有各個群組的權限。如下圖所示，若 Engineer 和 Sales 群組內都有同樣的使用者 Angela，則 Angela 可以擁有 Class A ~ F 權限。

群組名稱	Class A	Class B	Class C	Class D	Class E	Class F
1 Engineer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 Sales	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

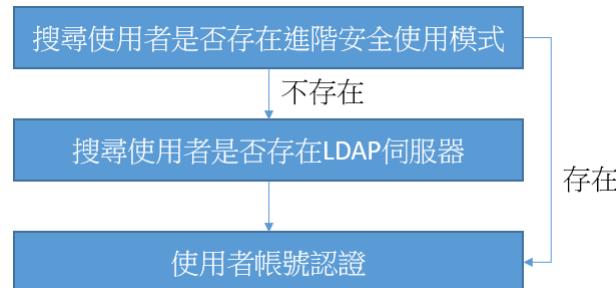
- 進階安全模式功能的使用者帳號密碼與 LDAP 模式使用的認證可同時使用。然而，若進階安全模式功能中的使用者名稱亦同時存在於 LDAP 伺服器中時，則僅會使用進階安全模式功能方式進行使用者帳號密碼驗證，並不會採用 LDAP 伺服器中的帳戶來認證。如下表所示，使用者 Angela 同時存在於 LDAP 伺服器與進階安全模式使用者中，則 HMI 僅會採用進階安全模式使用者帳號進行帳號驗證。

LDAP 伺服器使用者

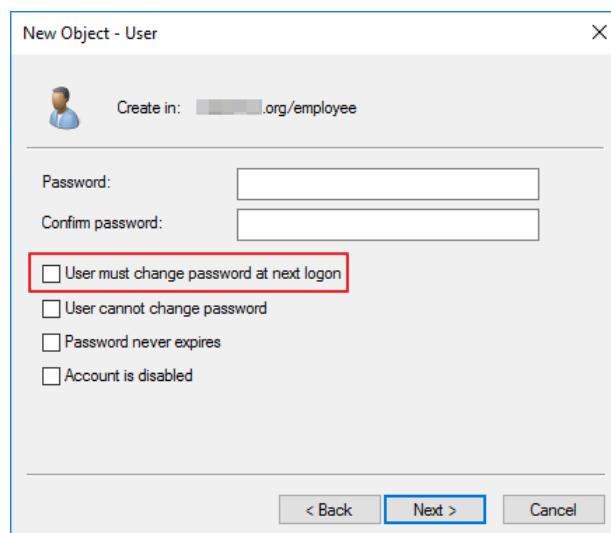
Name	Type
Angela	User
Bella	User
Cindy	User
Dora	User
Elly	User
Fanny	User

進階安全使用者

編號	啟用	隱藏使用者	使用者名稱	密碼
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Angela	1
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Gina	2
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Helen	3



- LDAP 模式不支援使用 [使用者索引] 功能登入。
- LDAP 目前僅支援 Active Directory。
- 目前 HMI 無法更改使用者的密碼，因此在 LDAP 伺服器建立使用者時，必須取消勾選 [User must change password at next logon]。



#### 10.4.5.1. 一般屬性設定頁

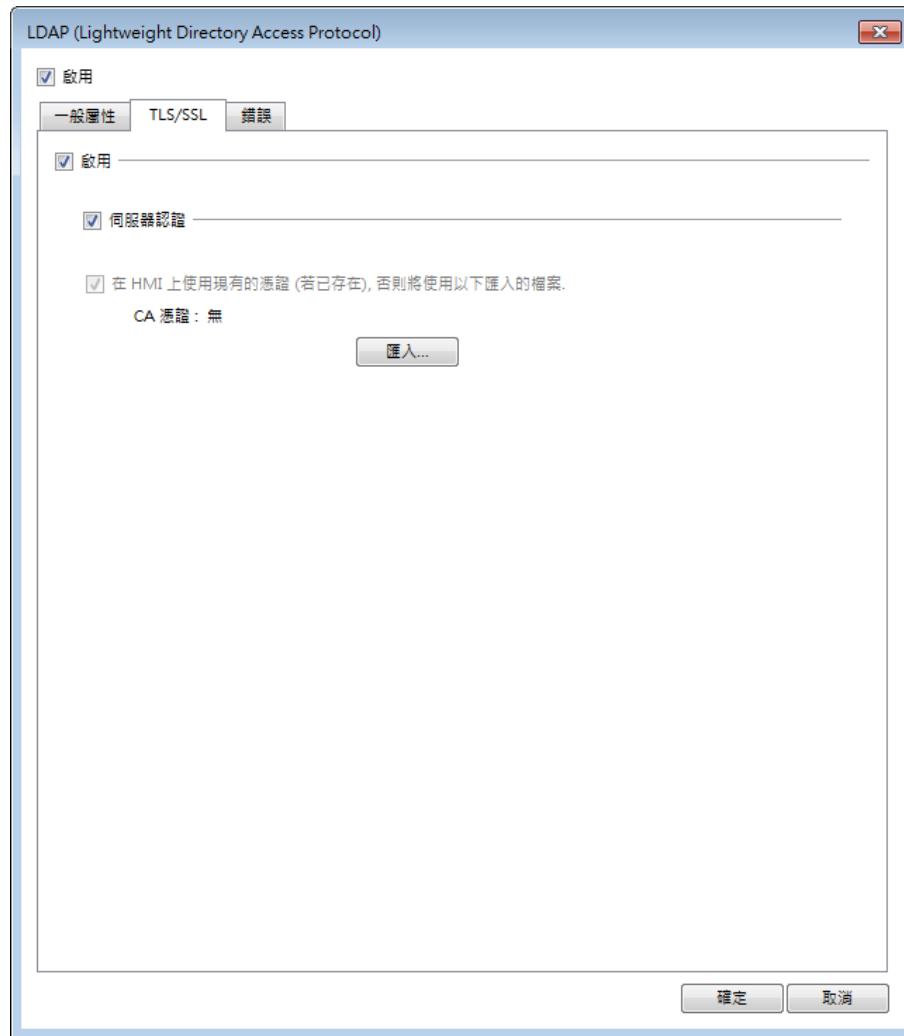
設定 LDAP 伺服器的連線資訊及群組可使用的權限類別。



設定	描述
<b>主機</b>	LDAP 伺服器的主機名稱或 IP 位址。
<b>連接埠</b>	LDAP 預設為 389，LDAPS 預設為 636。
<b>Base DN</b>	LDAP 伺服器的網域 (由 Domain Component DC 組成)。
<b>Users base DN</b>	使用者資訊的組織單元(OU)。
<b>Group base DN</b>	群組資訊的組織單元(OU)。
<b>新增</b>	新增一個群組。
<b>刪除</b>	刪除一個群組。
<b>從伺服器匯入</b>	透過帳號密碼登入 LDAP 伺服器取得所有可連接的群組。
<b>群組名稱及 Class</b>	各個群組可以使用的物件權限類別。群組名稱長度最多可使用 64 個字組，可使用英文 (有區分大小寫)/數字/符號 /Unicode。

#### 10.4.5.2. TLS/SSL 設定頁

使用 LDAPS (LDAP over SSL) 與 AD Server 進行連線時，需要使用加密的連線，必須啟用此分頁內的設定。



設定	描述
啟用	啟用後，LDAP 通訊連線會經過加密。
伺服器認證	連線時，HMI 會比對 AD Server 的憑證。
在 HMI 上使用 現有的憑證	設定使用現有 HMI 上的憑證或匯入新的憑證。

### 10.4.5.3. 錯誤設定頁

當無法連接 LDAP 伺服器時的錯誤碼。



#### 設定

#### 描述

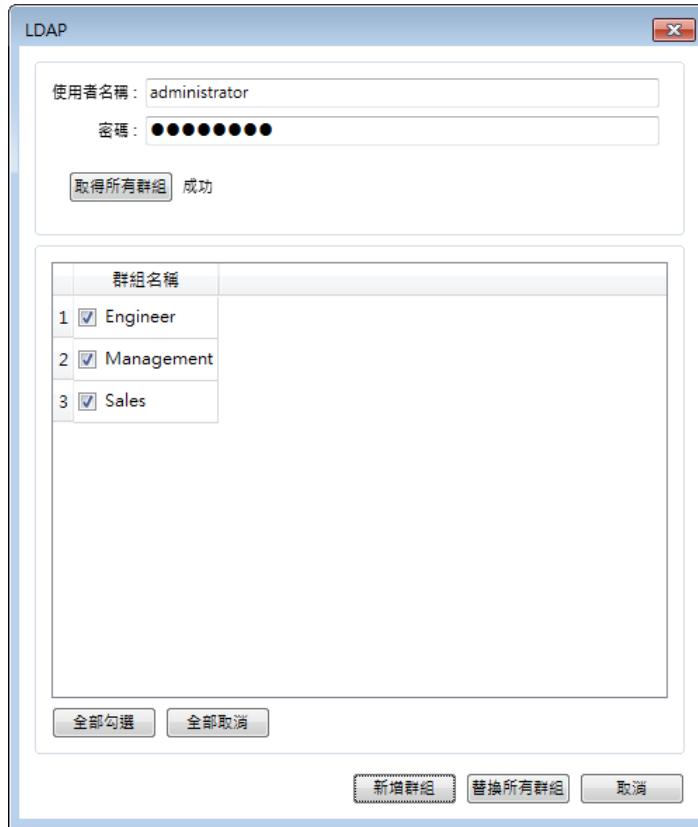
##### 錯誤位址

登入命令的執行結果會輸出到此位址。

數值	描述
0	無錯誤
1	LDAP 伺服器端異常或無輸入密碼
2	未知的錯誤
257	無法連上遠端 LDAP 伺服器
258	登入帳號或密碼錯誤
259	驗證憑證失敗
512	未知的 TLS
513	網域名稱與 CN 不符合

#### 10.4.5.4. LDAP 設定頁 ([從伺服器匯入] 設定頁)

從 LDAP 伺服器取得群組資訊。



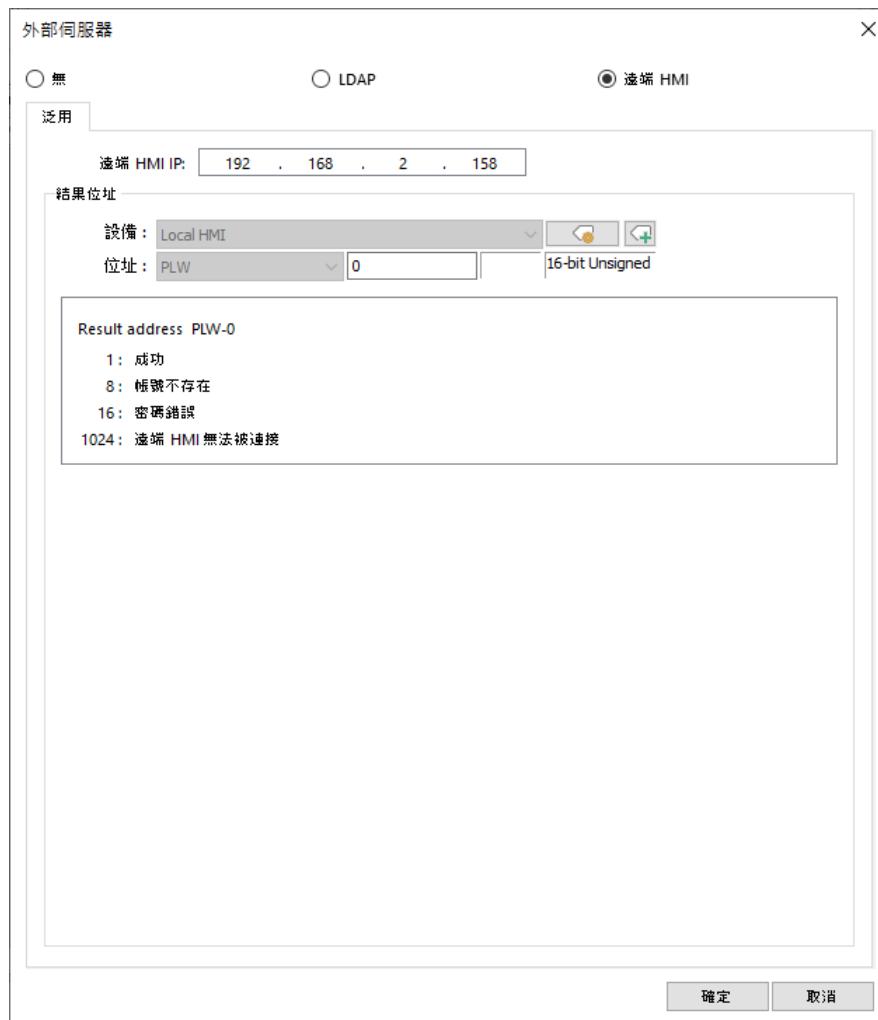
設定	描述								
使用者名稱	登入 LDAP 伺服器使用的帳號。								
密碼	登入 LDAP 伺服器使用的密碼。								
取得所有群組	取得 LDAP 伺服器中指定 DN 的所有群組。								
	<table border="1"> <thead> <tr> <th>錯誤訊息</th><th>描述</th></tr> </thead> <tbody> <tr> <td>Can't contact LDAP server</td><td>無法連上 LDAP 伺服器</td></tr> <tr> <td>Invalid Credentials</td><td>登入 LDAP 伺服器的帳號或密碼錯誤</td></tr> <tr> <td>Unknown</td><td>LDAP 伺服器異常或無輸入密碼</td></tr> </tbody> </table>	錯誤訊息	描述	Can't contact LDAP server	無法連上 LDAP 伺服器	Invalid Credentials	登入 LDAP 伺服器的帳號或密碼錯誤	Unknown	LDAP 伺服器異常或無輸入密碼
錯誤訊息	描述								
Can't contact LDAP server	無法連上 LDAP 伺服器								
Invalid Credentials	登入 LDAP 伺服器的帳號或密碼錯誤								
Unknown	LDAP 伺服器異常或無輸入密碼								

##### Note

- 最多可使用 128 個群組。匯入時，系統會先確認 LDAP 伺服器的群組數量，若超過 128 個，則會匯入失敗。
- 若匯入名稱重複時，不會清除原先該群組名稱的權限類別。

#### 10.4.6. 遠端 HMI 模式

使用 [遠端 HMI] 模式時，帳號管理可不需要在本機 HMI 端進行，而是交由遠端 HMI 管理。本機 HMI 可透過 [遠端 HMI] 上的帳號密碼進行登入，無須自行管理使用者的帳號密碼。



設定	描述
遠端 HMI IP	欲取得使用者資訊的遠端 HMI 伺服器的 IP。
結果地址	連接遠端 HMI 伺服器或取得使用者資訊有異常時，會將錯誤碼輸出到此地址。

#### Note

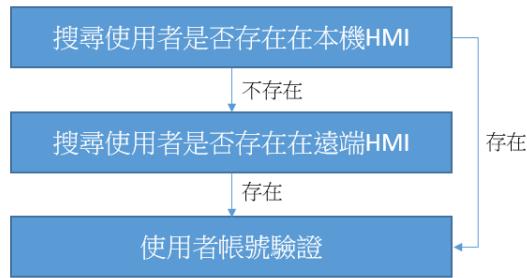
- 本機 HMI 的使用者帳號密碼與遠端 HMI 模式的認證可同時使用。然而，若本機 HMI 中的使用者名稱亦同時存在於遠端 HMI 中時，則僅會使用本機 HMI 中的使用者帳號進行使用者帳號密碼驗證，而不會採用遠端 HMI 的帳戶來認證。如下表所示，使用者 Angela 同時存在於遠端 HMI 與本機使用者中，則 HMI 僅會採用本機使用者帳號進行帳號驗證。

遠端 HMI 中的使用者

	啟用	隱藏使用者	使用者名稱	密碼
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Angela	111 <span style="background-color: red;">弱</span> 
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bella	222 <span style="background-color: red;">弱</span> 
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Gigi	333 <span style="background-color: red;">弱</span> 

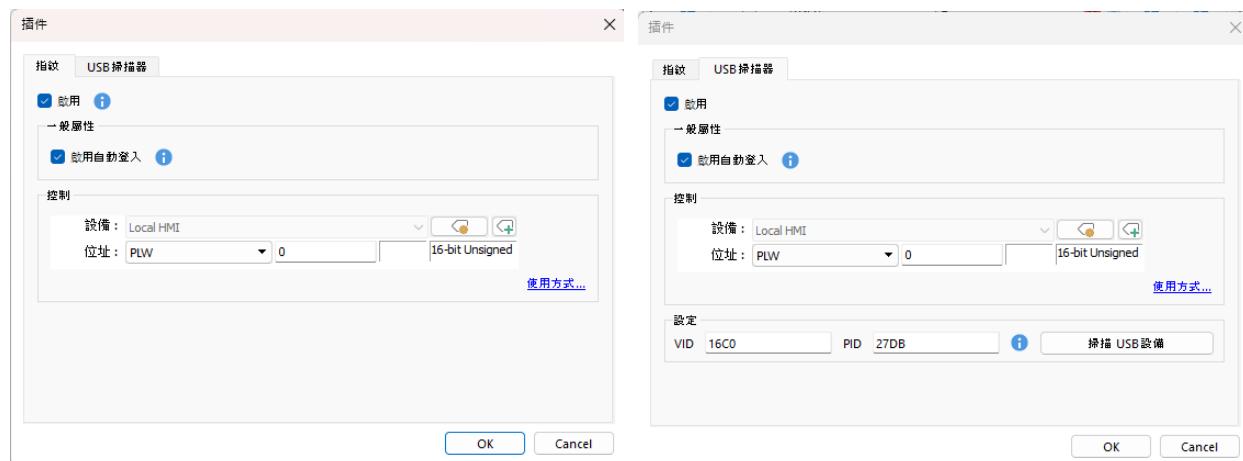
本機 HMI 中的使用者

	啟用	隱藏使用者	使用者名稱	密碼
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Angela	1 <span style="background-color: red;">弱</span> 
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Amy	2 <span style="background-color: red;">弱</span> 
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Allen	3 <span style="background-color: red;">弱</span> 



#### 10.4.7. 使用插件自動登入/登出

啟用插件後，使用者可透過指紋辨識器登入用指紋綁定的使用者帳號，或透過 USB 掃描器登入用卡片/條碼綁定的使用者帳號。



在設定 USB 掃描器時，為避免其他 USB 裝置影響登入運作，需先設定 USB 掃描器的 VID 及 PID。在點擊 [掃描 USB 裝置] 後，系統將提示“請插入 USB 設備”訊息，使用者將 USB 掃描器插入 PC 後，系統將會取得 USB 掃描器的一組特有的 VID 及 PID，點擊 [儲存] 後系統將會自動將該 VID 及 PID 代入設定。

[啟用自動登入]：勾選 [啟用自動登入] 模式，指紋/USB 掃描器保持在隨時可以登入的狀態，無須透過命令啟動掃描。若希望恢復預設透過輸入命令啟動掃描的狀態，可以輸入命令功能的命令 7 (暫停自動登入) 或命令 8 (恢復自動登入) 暫停/恢復自動登入狀態。

#### 控制位址使用說明

當控制位址設為 PLW-n 時，n 為任一數字，則將使用以下位址：

控制位址	標籤名稱	描述
PLW-n (佔 1 個字組)	命令	控制各項操作命令 (例如: 登入，綁定指紋/卡片，刪除指紋/卡片...等)。
PLW-n + 1 (佔 1 個字組)	命令執行結果	顯示執行命令的結果。
PLW-n + 2 (佔 1 個字組)	狀態	插件伺服器的初始化狀態。
PLW-n + 3 (佔 1 個字組)	錯誤	設備伺服器錯誤碼。

#### 命令功能說明

當在 [命令] PLW-n 輸入特定 [數值] 時,-> 可操作的功能如下：

設定數值	命令	搭配位址
1	通過指紋/RFID/條碼登入	
2	按使用者名稱添加指紋/RFID/條碼	需先定義 [使用者名稱]。

## 使用者密碼與物件安全防護

3	按使用者索引添加指紋/RFID/條碼	需先定義 [使用者索引]，請參考《10.4.4 進階安全模式搭配項目選單物件》說明。
4	按使用者名稱刪除所有指紋/RFID/條碼	需先定義 [使用者名稱]。
5	按使用者索引刪除所有指紋/RFID/條碼	需先定義 [使用者索引]。
6	刪除所有指紋/RFID/條碼	
7	暫停自動登入	
8	恢復自動登入	

## 命令執行結果說明

每當執行命令後，系統自動將執行結果輸出值傳送到控制位址的 PLW-n + 1 位址。

結果輸出值	訊息
0	命令執行成功
1	未知的錯誤
6	取消
101	無法配對的使用者帳戶
115	驗證失敗
其他	系統異常

## 錯誤碼說明

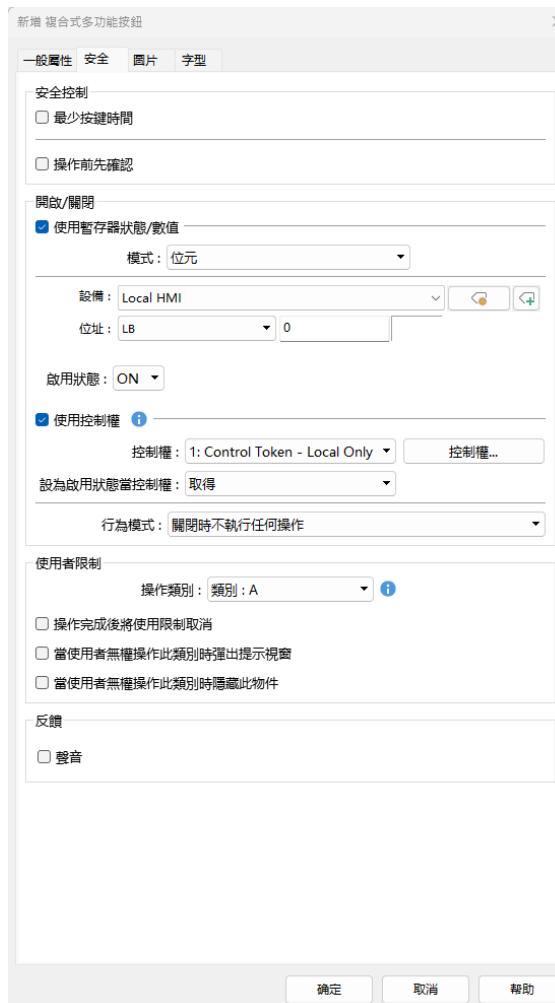
當插件伺服器初始化後，系統自動將執行結果輸出值傳送到控制位址的 PLW-n + 3 位址。

錯誤碼	訊息
0	初始化成功
1	未知的錯誤
2 或其他	系統異常

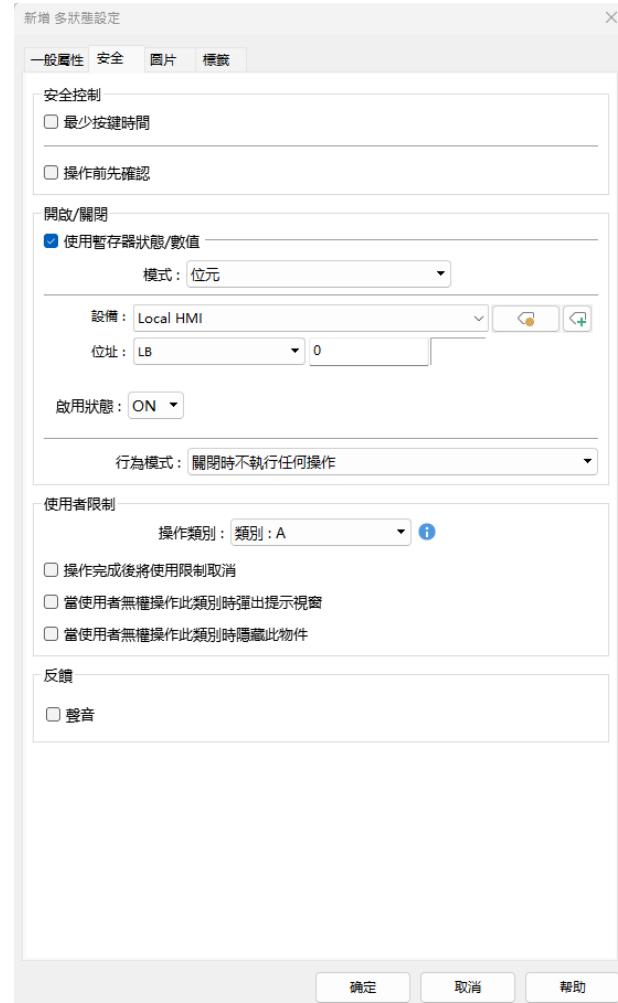
## 10.5. 物件安全設定頁

透過使用物件的【安全】設定頁的設置，可限制物件只有在符合特定條件時，才可進行操作。另外，此設定頁亦可設定操作物件時的聲音。

cMT / cMT X 系列

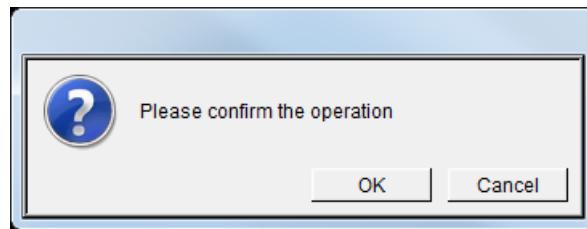


eMT、iE、XE、mTV 系列



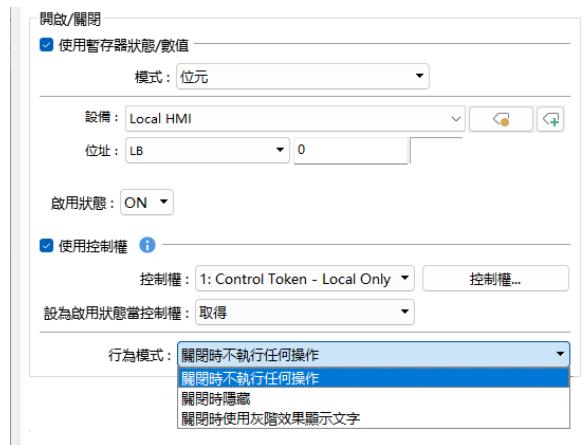
### 10.5.1. 安全控制

設定	描述
<b>最少按鍵時間</b>	只有當持續按壓物件時間大於此設定值才能成功啟動操作。
<b>操作前先確認</b>	按壓一個物件後，會出現一個確認對話窗，需點選 OK 來確定操作。如果等待確認操作的時間超過 [確認等待時間]，對話窗會自動消失並取消動作。



### 10.5.2. 開啟/關閉

物件是否允許被操作，將決定於一個指定位址的狀態或是是否取得控制權。如下圖所示，必須在 LB-0 狀態為 OFF 且取得第 2 組別的控制權時，才允許操作此物件。



以下說明當無法對物件進行操作時，物件的行為模式。

設定	描述
關閉時不執行任何操作	當物件無法被操作時，物件仍顯示於視窗。
關閉時隱藏	當物件無法被操作時，物件將被隱藏。
關閉時使用灰階效果顯示文字	當物件無法被操作時，物件的標籤文字會以灰階樣式顯示。

#### 10.5.2.1. 使用暫存器狀態/數值

物件根據指定位元/字組的數據而決定是否允許被操作。

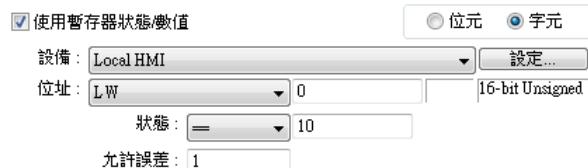
模式	描述
位元	當指定的位元位址於 On/Off 狀態時物件才可被操作。
字組	若勾選 [使用] 並選擇 [字組]，此物件是否允許被操作，將決定於一個指定字組位址的 [狀態]。
狀態	

可設定指定字組位址的條件，有 > 、 < 、 == 、 <> 、 >= 或 <= 可以選擇。其中 == 與 <> 可以設定 [允許誤差]。

<>：當暫存器數值 > [暫存器數值+允許誤差] 或 暫存器數值 < [暫存器數值-允許誤差] 時可以操作。

==：當暫存器數值介於 [暫存器數值-允許誤差] 與 [暫存器數值+允許誤差] 時可以操作 (包含暫存器數值±允許誤差)。

舉例來說：



當指定字組位址的數值介於 9~11 之間可以操作。

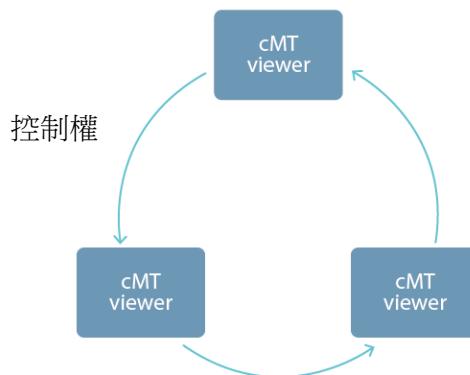
#### 條件物件

若勾選 [使用] 並選擇 [條件物件]，此物件是否允許被操作，將決定於被指定到的 [觸發條件] 物件所輸出的 On/Off 狀態。

#### Note

- cMT / cMT X 系列可支援 [字組] 的物件有多狀態設定物件、數值物件、字元物件、複合式切換開關物件；iE/XE/eMT/mTV 可支援 [字組] 的物件有多狀態設定物件、數值物件。

#### 10.5.2.2. 控制權 (cMT / cMT X 系列)



由於 cMT / cMT X 系列可同時有多台 cMT Viewer 進行操作，為了避免同時有多台的 cMT viewer 裝置操作同個物件而影響到工作流程的操作安全，可以使用控制權進行物件的保護。簡單來說，只有拿到控制權的 cMT Viewer 裝置，才能對物件進行操作。啟用控制權時，同時間一個控制權僅有一台裝置可以取得，其餘的裝置則必須依序等待取得物件的控制權。

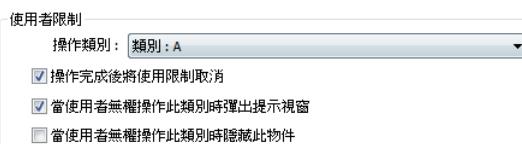
目前支援控制權的物件有：複合式切換開關物件、數值物件、字元物件、直接視窗物件、間接視

窗物件。

設定	描述
控制權	設定控制權的組別。
控制權...	可新增/刪除控制權的組別。詳細設定內容請參考第 34 章_控制權。
設為啟用狀態	若使用 [取得]，則拿到控制權的裝置可操作物件。若使用
當控制權	[未取得]，則未拿到控制權的裝置可操作物件。

### 10.5.3. 使用者限制

設定物件類別，只允許可操作此類別的使用者操作。

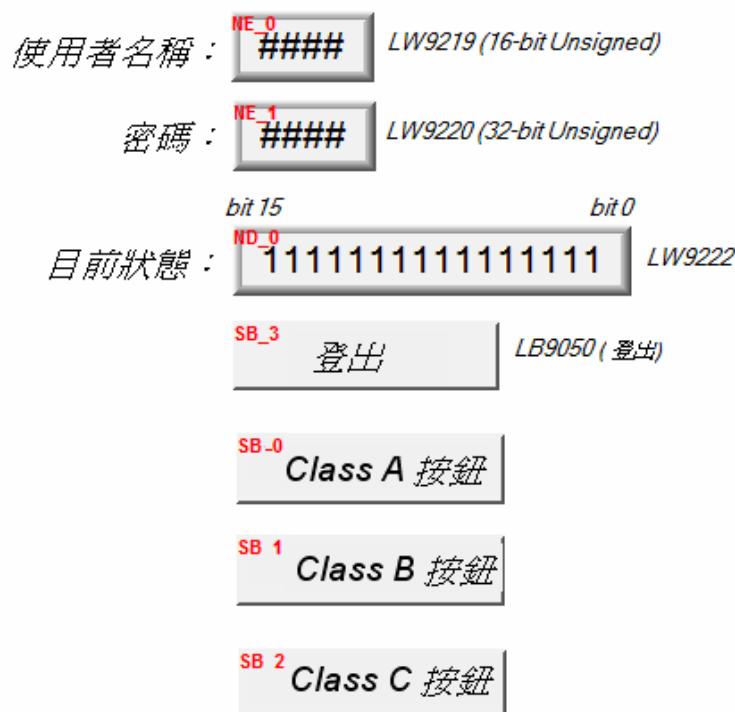


設定	描述
操作類別	“無”表示任何使用者皆可操作。“管理者”表示只有 admin 帳號可以操作。
操作完成後將使用限制取消	一旦使用者的操作等級被允許操作該物件，系統便不再檢查該物件的安全等級，也就是說，即使是別的使用者，該物件也可被隨意操作。
當使用者無權操作此類別時彈出提示視窗	當使用者操作身份不符合此物件的操作等級時，將彈出警告視窗 7 號。使用者可自行設定此視窗上的提示文字。
當使用者無權操作此類別時隱藏此物件	當使用者操作身份不符合此物件操作的等級時，物件會被隱藏。

### 10.6. 物件安全防護範例

一般模式之物件安全防護的使用範例：

- 建立一個工程檔案，[系統參數設定] » [使用者密碼] » [一般模式] 中啟用三個使用者，例如：
  - 使用者 1 = 操作物件類別 A
  - 使用者 2 = 操作物件類別 A, B
  - 使用者 3 = 操作物件類別 A, B, C
- 在視窗 10 設計如下圖所示：



建立兩個 [數值輸入] 物件：

[LW-9219] 使用者編號 (1~12)，長度 = 1 word

[LW-9220] 輸入使用者密碼，長度 = 2 words

建立一個 [數值顯示] 物件：

[LW-9222] 顯示當前登入使用者的權限狀態，格式為 16-bit Binary

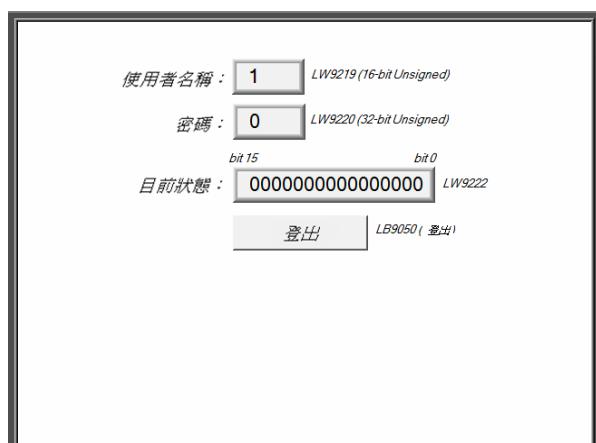
建立一個 [位元狀態設定] 物件：

[LB-9050] 使用者登出

建立三個 [位元狀態設定] 物件：

選擇不同的物件類別，但皆設定 [當使用者無權操作此類別時隱藏此物件]。

3. 完成上述的各項設計並在存檔與編譯後即可執行離線模擬功能，下圖為離線模擬功能的起始畫面。

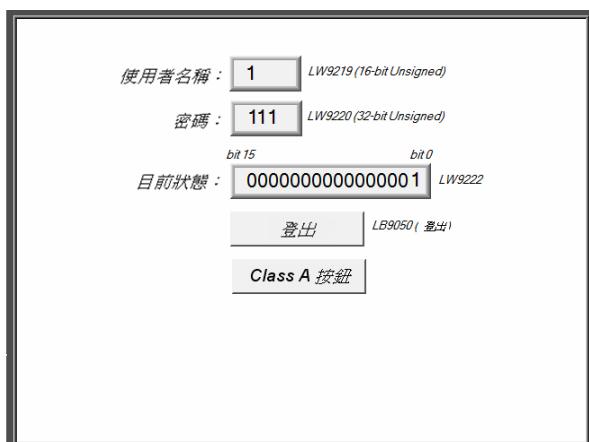


因尚未輸入密碼，數值物件顯示

“0000000000000000”，使用物件類別 “無”。

[Class A 按鈕] ~ [Class C 按鈕] 物件分別屬於類別 “A ~ C” 並設定 [當使用者無權操作此類別時隱藏此物件]，所以皆被系統所隱藏。

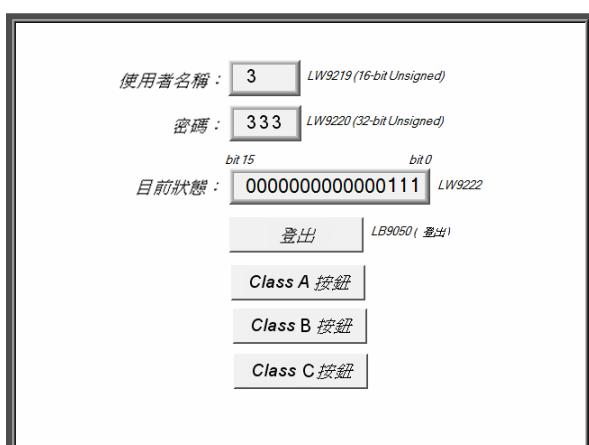
## 使用者密碼與物件安全防護



輸入“使用者 1”的密碼(111)：

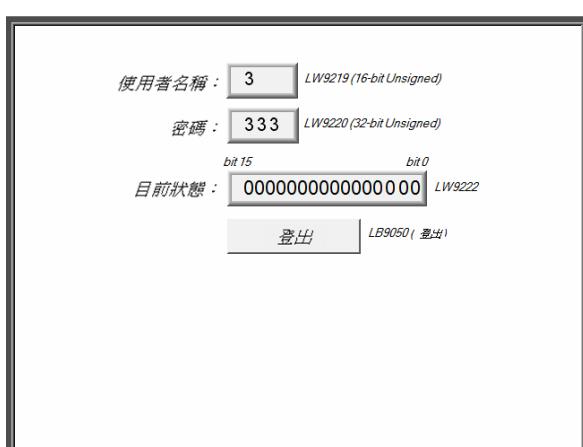
因設定“使用者 1”允許操作類別 A 物件，所以此時 [Class A 按鈕] 物件將出現並允許操作。

[LW-9222] 的 bit 0 變為“1”，表示此時的使用者允許使用類別“A”的物件。



輸入“使用者 3”的密碼(333)，

因設定允許操作類別 A, B, C 物件，[LW-9222] 的 bit 0 ~ bit 2 的值皆變為“1”，表示此時的使用者允許使用類別“A ~ C”的物件。



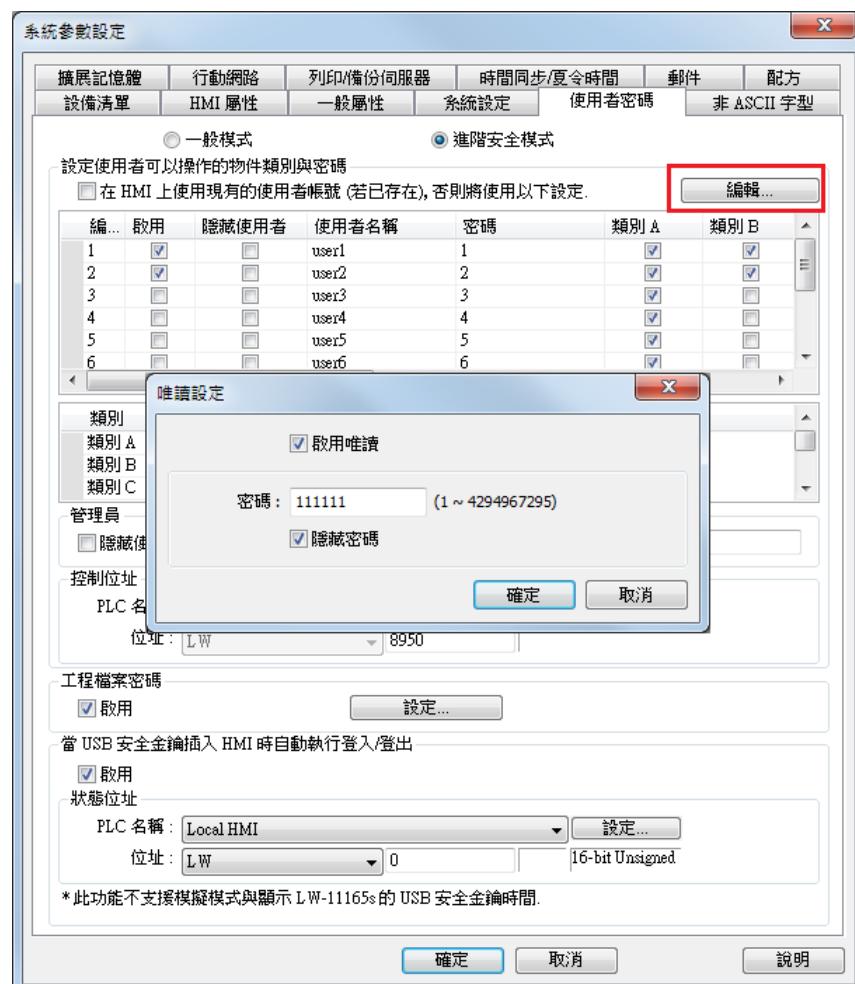
此時若按下 [登出] 強迫使用者登出，可以發現系統將回復到起始狀態，此時只允許操作類別為“無”的物件。

### Note

- 密碼輸入：當密碼輸入錯誤時，[LB-9060] 的狀態將被設定為 ON 狀態；當密碼輸入成功時，[LB-9060] 的狀態將自動被恢復為 OFF 狀態。使用者 1 至使用者 12 的密碼可以利用讀取系統暫存器 [LW-9500] 至 [LW-9522]，共 24 words 的內容取得。
- 線上更改密碼：當 [LB-9061] 的狀態設定為 ON 時，系統將讀取 [LW-9500] 至 [LW-9522] 內的數值，更新使用者的密碼，往後並使用這些新的密碼。此時使用者可操作類別的物件並不會因密碼的變更而改變。

## 10.7. 工程檔案之使用者密碼編輯保護

若您的工程檔案可能需要傳給其他人編輯，但又擔心使用者密碼的設定外流，可以藉由點選 [編輯] 按鍵來啟用唯讀模式。



若勾選 [啟用唯讀]，欲編輯使用者密碼的相關設定前需先輸入密碼。

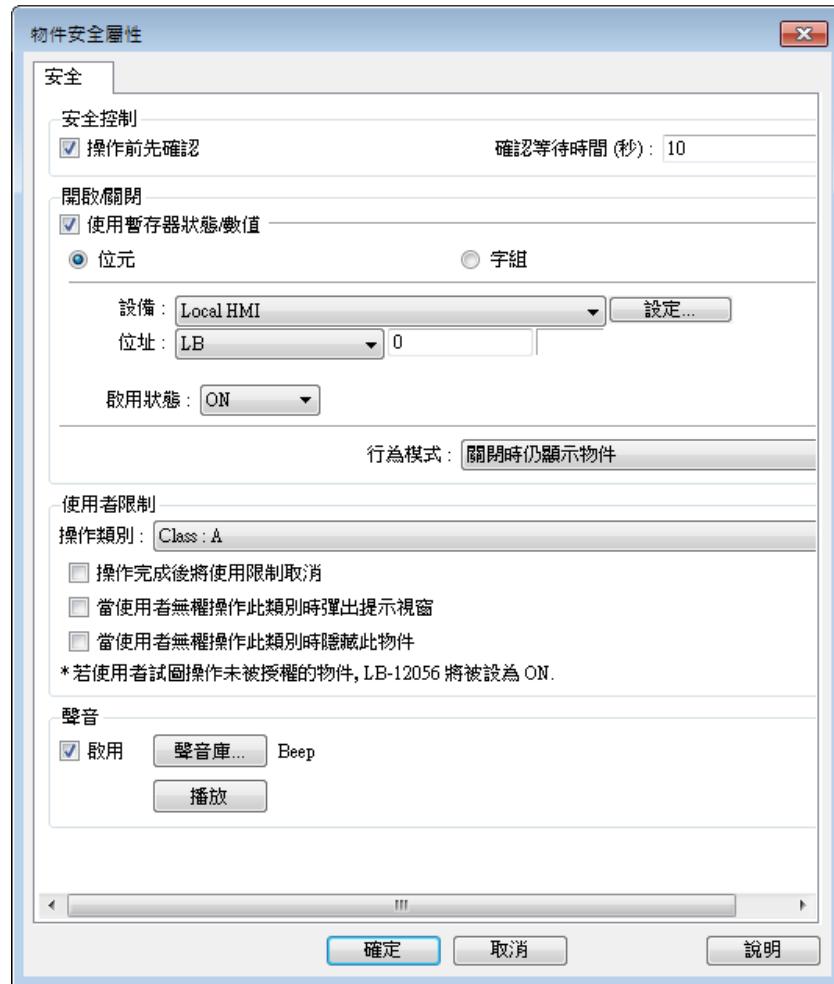
若勾選 [隱藏密碼]，密碼將以 \* 符號顯示。

### Note

- 若不慎忘記密碼，因已加密所以原廠也無法解開，請妥善保管密碼。

## 10.8. 批次修改物件安全屬性

圈選多個物件後按下右鍵選擇安全設定即可進入物件安全屬性視窗，介面與單一物件的安全屬性完全相同，可以一次性地修改大量物件的安全屬性。



### Note

- 若圈選的多數物件有不同安全屬性介面，批次修改的介面則會有些微調整。舉例如下，同時選擇位元指示燈物件與數值物件的物件安全屬性介面就只會出現可以設定的部分，其餘會隱藏或使用反灰顯示。



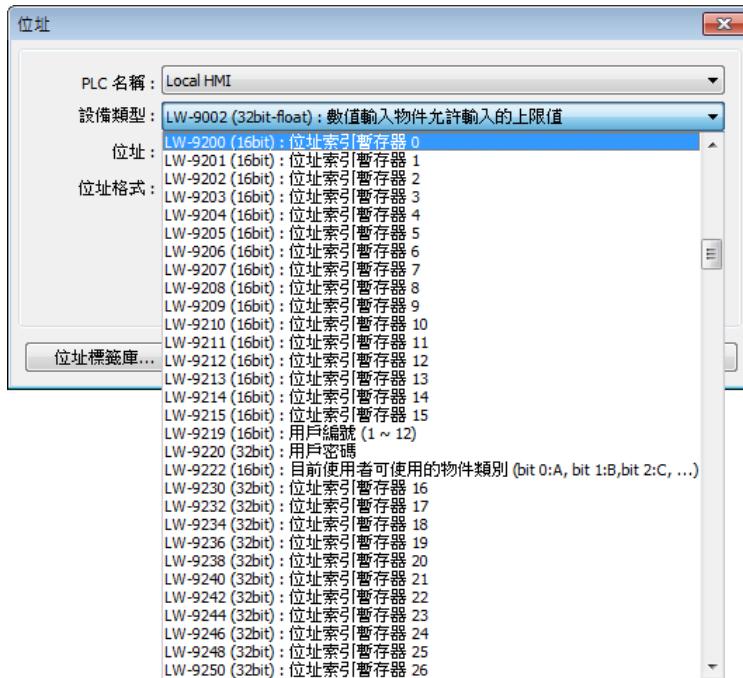
# 11. 索引|暫存器

本章節說明如何使用索引|暫存器。

11.1. 概要 .....	11-2
11.2. 使用索引 暫存器範例 .....	11-4
11.3. 使用標籤式 PLC 與索引 暫存器範例 .....	11-6

## 11.1. 概要

索引暫存器是 EasyBuilder Pro 提供用於變換位址的暫存器。有了索引暫存器後，使用者可以在不改變物件位址內容的情況下，在 HMI 上直接修改物件的讀取與寫入位址。EasyBuilder Pro 提供 32 組索引暫存器，分別為 16 組 16-bit 的索引暫存器和 16 組 32-bit 的索引暫存器。cMT/cMT X 系列的機型額外提供 16 組 16-bit 的私有索引暫存器和 16 組 32-bit 的私有索引暫存器。



16-bit 位址索引暫存器 0 至 15 的對應位址為 LW-9200 (16-bit)至 LW-9215 (16-bit)，其最大偏移量為 65536 words。

32-bit 位址索引暫存器 16 至 31 的對應位址為 LW-9230 (32-bit)至 LW-9260 (32-bit)，其最大偏移量為 4294967296 words。

16-bit 私有位址索引暫存器 0 至 15 的對應位址為 PLW-9200 (16-bit)至 PLW-9215 (16-bit)，其最大偏移量為 65536 words。

32-bit 私有位址索引暫存器 16 至 31 的對應位址為 PLW-9230 (32-bit)至 PLW-9260 (32-bit)，其最大偏移量為 4294967296 words。

使用 [索引暫存器] 後，所使用 [設備類型] 的位址則由下列公式決定：

“設定的常數位址 + 所選擇索引暫存器中的值”

私有索引暫存器的用法與一般索引暫存器相同，該操作只會作用於目前使用的 cMT Viewer，其餘裝置的 cMT Viewer 或人機本體的畫面並不會受到影響。

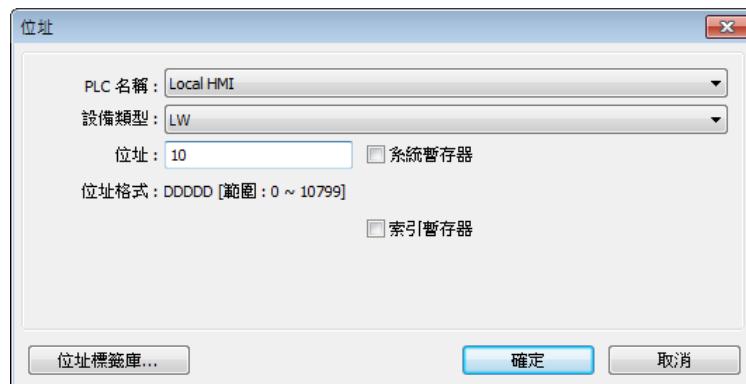
 Note

- 索引暫存器可使用於所有設備的字組格式的位址暫存器。若使用於位元格式的位址暫存器，則當索引暫存器中的數據每改變 1 會偏移 16 個位元位址。
- PLB, PLW 不支援索引暫存器。

## 11.2. 使用索引暫存器範例

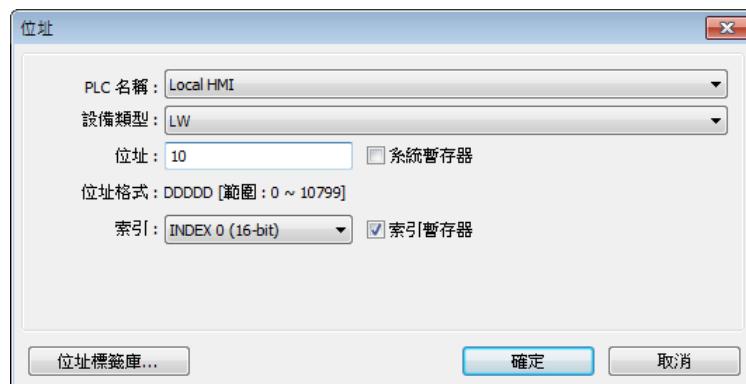
以下為實際存取位址的計算過程：

若未勾選 [索引暫存器] 並設定位址為 LW-10，系統則對此位址做讀取/寫入的動作。

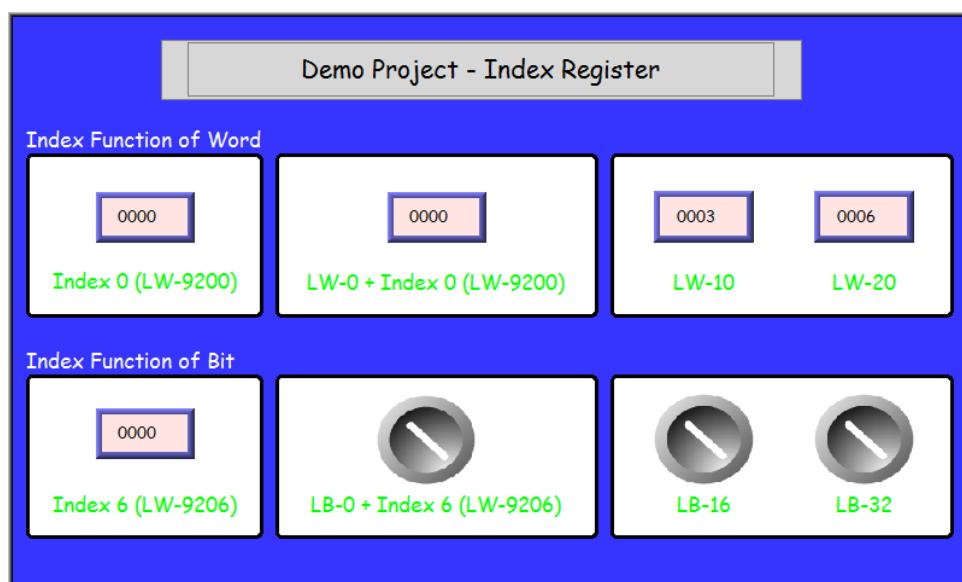


若勾選 [索引暫存器] 並選擇 [索引] 為 [INDEX 0 (16-bit)]，則存取位址為 [LW-(10 + 位址索引暫存器 0 的值)]。

例如: [LW-9200] 位址中的數值為 “5”，根據計算公式可得出實際存取位址為 [LW-(10+5)]，即 [LW-15]。

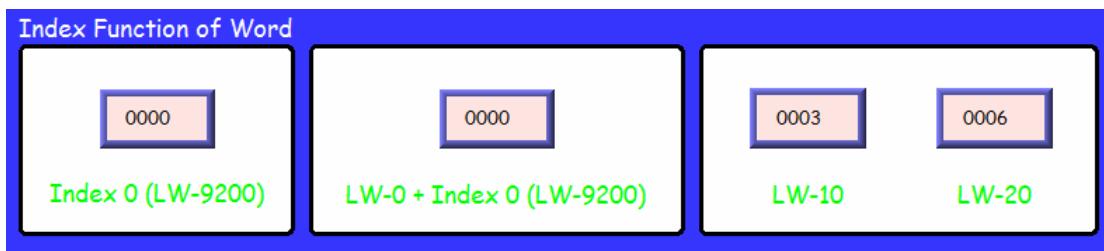


以實際範例作進一步說明：

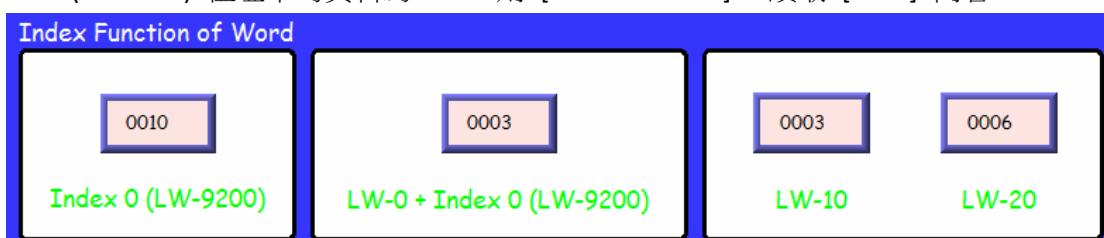


## 範例 1

下圖表示使用索引暫存器的字組格式位址。假設 [LW-0] 的值為 “0”，[LW-10] 的值為 “3”，[LW-20] 的值為 “6”，則結果如下：



若 Index 0 (LW-9200) 位址中的資料為 “0”，則  $[LW-0 + Index 0] = \text{讀取 } [LW-0]$  內容。

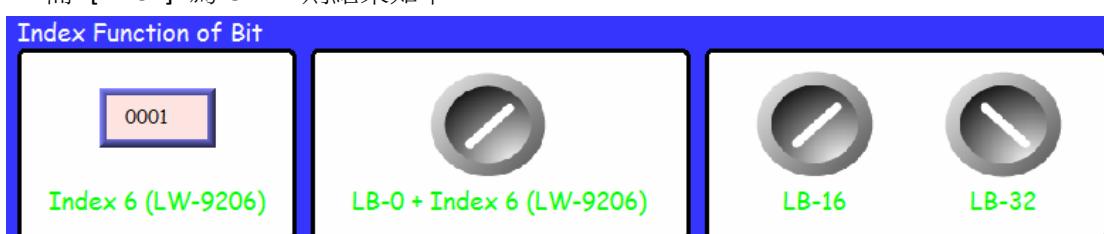


若 Index 0 (LW-9200) 位址中的資料設為 “10”，則  $[LW-0 + Index 0] = \text{讀取 } [LW-10] = "3"$ 。

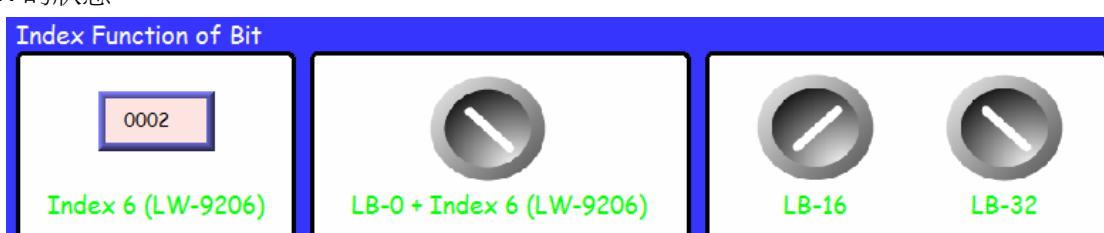
## 範例 2

下圖表示使用索引暫存器的位元格式位址。

由於 1 個字組等於 16 個位元，所以索引暫存器數值改變 1 相當於偏移 16 個位元。假設 [LB-16] 為 ON，而 [LB-32] 為 OFF，則結果如下：



若 Index 6 (LW-9206) 位址中的資料設為 “1”，則開關  $[LB-0 + Index 6]$  讀取 LB-16 位址狀態，也就是 ON 的狀態。



若 Index 6 (LW-9206) 位址中的資料設為 “2”，則開關  $[LB-0 + Index 6]$  讀取 LB-32 位址狀態，也就是 OFF 的狀態。

 Note

- 使用索引暫存器於位元位址時，所設定的位元位址將會以 16 個位元位址為一個偏移單位。假設以 LB-0 為範例且使用索引暫存器，若是索引暫存器裡的數值為 1，則 LB-16 將會動作，若是索引暫存器裡的數值為 2，則 LB-32 會動作。



下載範例程式前，請先確定已連上網路線。

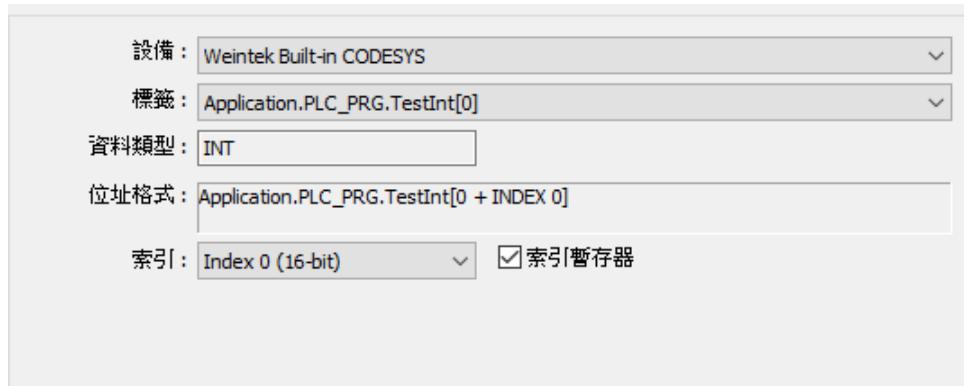
### 11.3. 使用標籤式 PLC 與索引暫存器範例

索引暫存器支援一維陣列的標籤。

#### 範例 1: INT

若勾選 [索引暫存器] 並選擇 [索引] 為 [INDEX 0 (16-bit)]，則存取位址為 Application.PLC\_PRG.TestInt[0 + 位址索引暫存器 0 的值]。

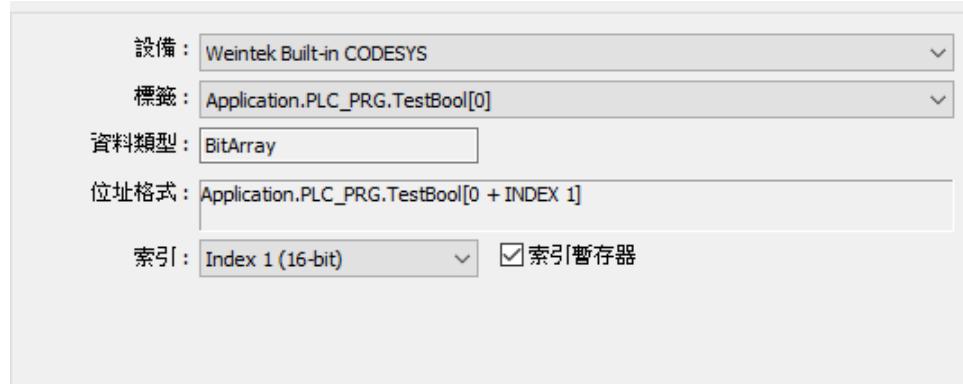
例如: [LW-9200] 位址中的數值為 “5”，根據計算公式可得出實際存取位址為 Application.PLC\_PRG.TestInt[5]。



## 範例 2: 位元

若勾選 [索引[暫存器]] 並選擇 [索引[]] 為 [INDEX 1 (16-bit)]，則存取位址為 Application.PLC\_PRG.TestBool[0 + 位址索引[暫存器 1 的值]]。

例如: [LW-9201] 位址中的數值為 “7”，根據計算公式可得出實際存取位址為 Application.PLC\_PRG.TestBool[7]。



值得注意的是，和非標籤式地址的不同，索引[暫存器]用於位元標籤陣列時，就不需要進行修正。

# 12. 鍵盤的設計與使用

本章節說明如何設計與使用鍵盤。

12.1.	概要 .....	12-2
12.2.	設計彈出鍵盤 .....	12-2
12.3.	使用直接視窗的方式設計鍵盤 .....	12-5
12.4.	將鍵盤固定在需要輸入的視窗上 .....	12-6
12.5.	製作 Unicode 鍵盤.....	12-6

## 12.1. 概要

數值輸入與字元輸入物件都需要使用鍵盤做為輸入工具，而數字鍵盤及字元鍵盤均是使用功能鍵物件來製作的。除了使用 EasyBuilder Pro 提供的內建鍵盤外，您也可以自行設計鍵盤，而 cMT 及 cMT X 系列另提供系統鍵盤可選擇。

鍵盤種類可分為：

- 彈出鍵盤（可選擇是否使用視窗控制條）
- 固定鍵盤
- Unicode 文字鍵盤
- 系統鍵盤（cMT/ cMT X 系列支援）

## 12.2. 設計彈出鍵盤

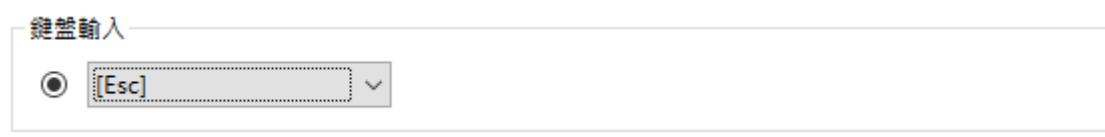
1. 先建立並開啟要作為鍵盤的視窗，假設為視窗 200。



2. 調整視窗 200 的長度與寬度，建立各個功能鍵物件，並用 [ASCII/Unicode 模式]。

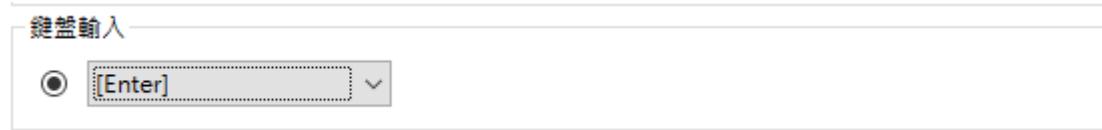


將其中一個功能鍵用來觸發取消訊號 [ESC]。

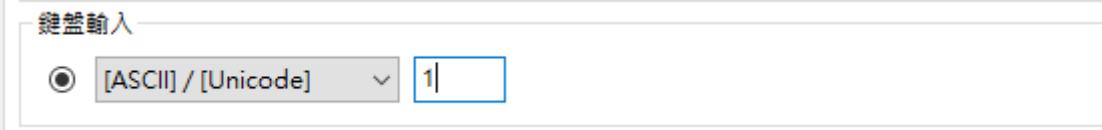


## 鍵盤的設計與使用

另一個功能鍵用來觸發輸入訊號 [Enter]。



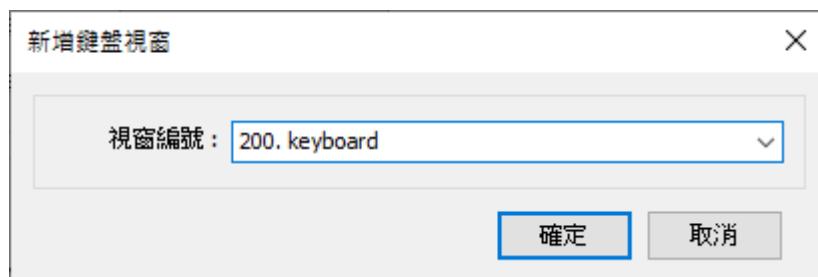
其他大部分功能鍵用來觸發數值輸入訊號，例如用來觸發數值 1 的輸入訊號。



3. 最後為功能鍵挑選適合的圖形做為顯示。



4. 在 [系統參數設定] » [一般屬性] » [鍵盤] 設定中，按下 [新增] 後選擇加入 [視窗 200]。最多可新增 32 個鍵盤視窗。



5. 在完成上述的所有步驟後，當使用數值輸入與字元輸入物件的設定頁時，即可發現在 [數值輸入] » [鍵盤] 設定中的 [視窗編號]，增加了“200. Keyboard”的選項。[鍵盤彈出位置] 可用來選擇鍵盤在 HMI 的出現位置，系統將 HMI 螢幕劃分為 9 個區域，如下圖所示。



6. 在選擇 [200. Keyboard] 後，當按下數值輸入或字元輸入物件時，將自動彈出視窗 200，按鍵盤上的功能鍵就可以輸入數值。



### 12.3. 使用直接視窗的方式設計鍵盤

- 新增一個直接視窗物件，設定讀取位址 LB-0 來啟動直接視窗。在 [屬性] 內勾選 [視窗控制條] 及設定鍵盤所在 [視窗編號]。



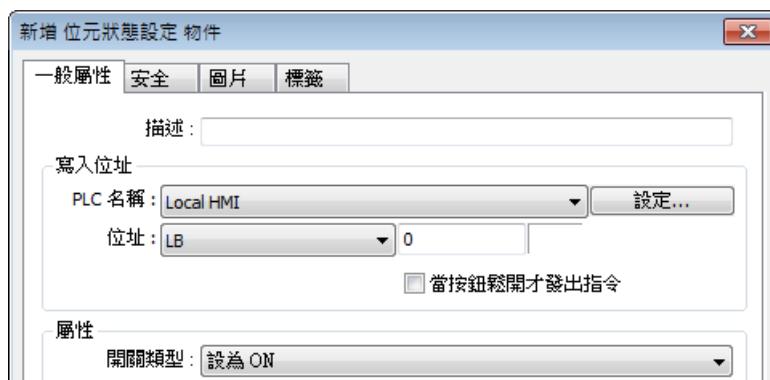
- 在 [位置] 頁籤將物件設定自動調整視窗尺寸。



- 新增數值輸入物件，在數值輸入頁籤內不要勾選 [使用彈出鍵盤]。



4. 設定一個位元狀態設定物件，寫入位址為 LB-0，開關類型為 [設為 ON]，並重疊在數值輸入物件上。當按下數值輸入物件的同時，也會將鍵盤視窗開啟。



5. 在鍵盤的 [Enter] 功能鍵和 [ESC] 功能鍵上，分別放置一個位元狀態設定物件，寫入位址為 LB-0，開關類型為 [設為 OFF]。在按下這兩個鍵的任意一個鍵時，可將鍵盤視窗關閉。

#### 12.4. 將鍵盤固定在需要輸入的視窗上

若不採用彈出鍵盤方式或是使用直接視窗來預設鍵盤所在位置，可採用此固定鍵盤方式，但這方式將無法移動或關閉鍵盤。

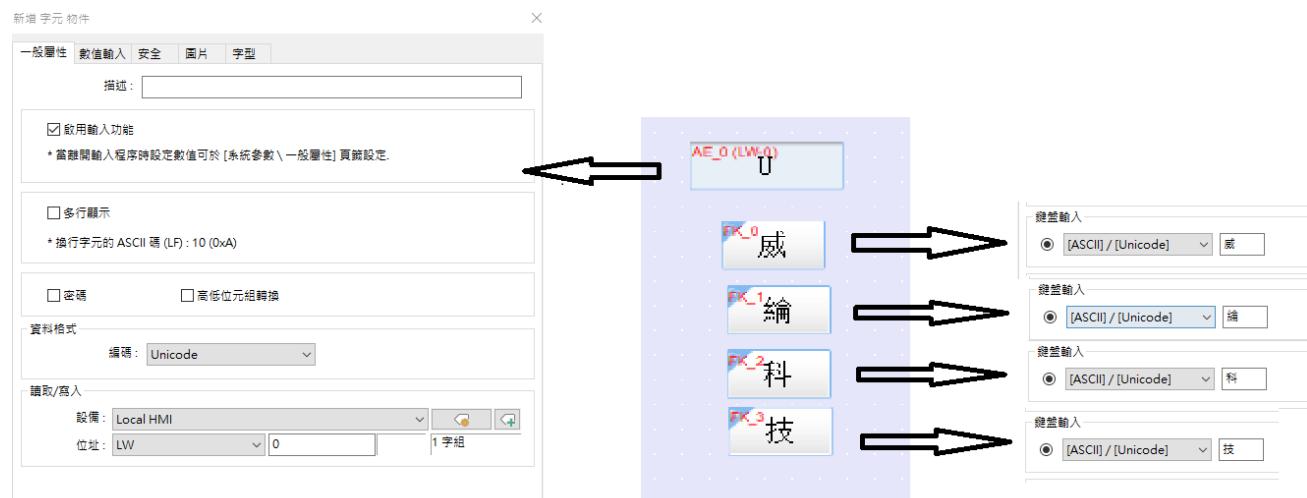
1. 新增數值輸入物件，在 [數值輸入] » [鍵盤] 屬性中不要勾選 [使用彈出鍵盤]。
2. 使用功能鍵物件將鍵盤按鍵設計好後，放置於視窗上即可使用。
3. 當按下數值輸入物件時，可以藉由鍵盤上的功能鍵來輸入數值。

#### 12.5. 製作 Unicode 鍵盤

本節說明如何使用功能鍵物件製作 Unicode 鍵盤：

1. 放置一個字元輸入物件在視窗上，並勾選 [Unicode]。

## 2. 製作「威」、「綸」、「科」、「技」這四個文字輸入功能鍵，即完成一個簡單的文字鍵盤。



### Note

■ 您可以將自製的鍵盤設定群組為【群組圖片】並添加到【群組圖庫】中，以便於後續的調用。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

# 13. 物件

本章節說明如何設計與使用各種物件。

13.1.	位元狀態指示燈 .....	13-3
13.2.	多狀態指示燈 .....	13-5
13.3.	位元狀態設定 .....	13-10
13.4.	多狀態設定 .....	13-13
13.5.	功能鍵 .....	13-21
13.6.	位元狀態切換開關 .....	13-28
13.7.	多狀態切換開關 .....	13-31
13.8.	滑動開關 .....	13-34
13.9.	數值 .....	13-40
13.10.	字元 .....	13-59
13.11.	間接視窗 .....	13-63
13.12.	直接視窗 .....	13-72
13.13.	移動/旋轉圖形 .....	13-78
13.14.	動畫 .....	13-83
13.15.	棒圖 .....	13-87
13.16.	錶針 .....	13-94
13.17.	趨勢圖 .....	13-102
13.18.	歷史數據顯示 .....	13-124
13.19.	數據群組顯示 .....	13-133
13.20.	XY 曲線圖 .....	13-141
13.21.	報警條與報警顯示 .....	13-148
13.22.	事件顯示 .....	13-157
13.23.	資料傳輸 .....	13-171
13.24.	備份 .....	13-179
13.25.	媒體播放器 .....	13-188
13.26.	BACnet Schedule .....	13-195
13.27.	PLC 控制 .....	13-199
13.28.	排程 .....	13-206
13.29.	項目選單 .....	13-218
13.30.	計時器 .....	13-226
13.31.	影像輸入 .....	13-230
13.32.	系統訊息 .....	13-236

13.33. 配方檢視 .....	13-238
13.34. 流動塊 .....	13-245
13.35. 操作記錄 .....	13-249
13.36. 複合式多功能按鈕 .....	13-263
13.37. 圓盤曲線圖 .....	13-271
13.38. 圖片檢視 .....	13-280
13.39. 檔案瀏覽器 .....	13-283
13.40. 匯入/匯出 .....	13-286
13.41. 圓餅圖 .....	13-290
13.42. 條碼 .....	13-294
13.43. 字串表 .....	13-306
13.44. 資料庫 .....	13-309
13.45. 動態刻度 .....	13-324
13.46. 動態繪圖 .....	13-327
13.47. PDF 檢視器 .....	13-332
13.48. 表格 .....	13-334
13.49. VNC Viewer .....	13-336
13.50. 連絡人編輯器 .....	13-340
13.51. 事件甘特圖 .....	13-343
13.52. 動作觸發 .....	13-349
13.53. 月曆 .....	13-355
13.54. 觸控手勢 .....	13-357
13.55. PLC 網頁瀏覽器 .....	13-360
13.56. 觸發條件 .....	13-364
13.57. FTP 連線 .....	13-367
13.58. 日期/時間 .....	13-370
13.59. 模板 .....	13-372

## 13.1. 位元狀態指示燈

### 13.1.1. 概要

[位元狀態指示燈] 物件用來顯示位元暫存器的狀態。狀態 0 代表位元的狀態為 OFF；狀態 1 代表位元的狀態為 ON。

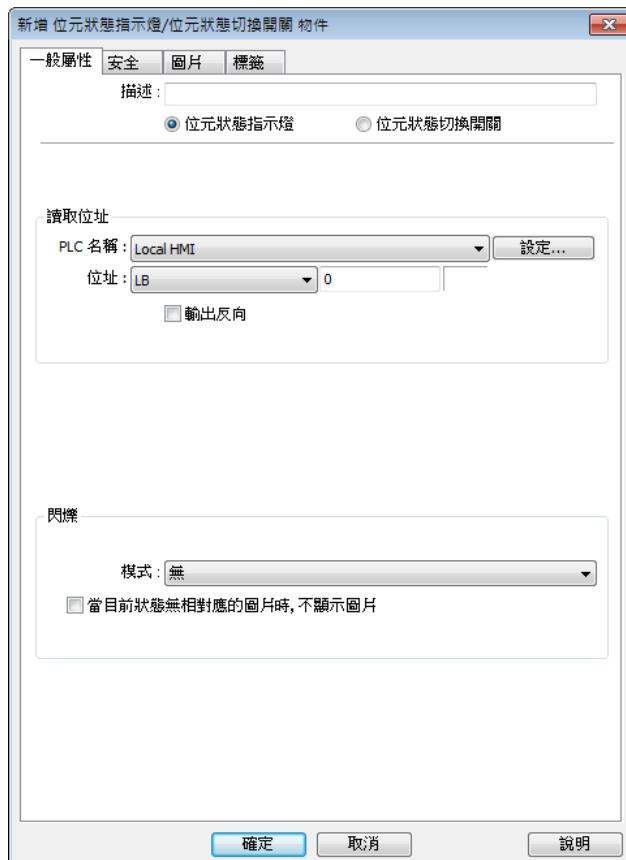


### 13.1.2. 設定



按下工作列的 [物件] » [位元狀態指示燈] 按鈕後即會開啟 [位元狀態指示燈] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [位元狀態指示燈] 物件。

#### 一般屬性設定



設定	描述
描述	使用者可為此物件描述相關訊息。 <b>位元狀態指示燈 / 位元狀態切換開關</b> 可與 [位元狀態切換開關] 功能互相轉換。
讀取位址	點選 [設定] 後選擇位元暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制位元狀態指示燈物件。 使用者也可在 [一般屬性] 頁中設定位址。
輸出反向	可以將讀取的狀態作反向顯示，例如位元的狀態實際上為 OFF，但勾選了 [輸出反向] 後會顯示為 ON。
閃爍	設定物件在位元狀態為 ON 或 OFF 的顯示方式。 <b>模式：</b> <b>無</b> 不閃爍。 <b>狀態為 0 時顯示圖片</b> 狀態為 OFF 時，圖片會進行狀態 0 及狀態 1 交互閃爍。 <b>狀態為 1 時顯示圖片</b> 狀態為 ON 時，圖片會進行狀態 0 及狀態 1 交互閃爍。 <b>狀態為 0 時閃爍</b> 狀態為 OFF 時，圖形 0 會進行出現與消失交互動作。 <b>狀態為 1 時閃爍</b> 狀態為 ON 時，圖形 1 會進行出現與消失交互動作。 <b>當目前狀態無相對應的圖片時，不顯示圖片</b> 若勾選，當圖片數目不足以顯示全部狀態時，將不顯示圖片。反之則顯示最後一個狀態。

 **Note**

- 在標籤頁中，若勾選 [ON=OFF(使用狀態)]，則狀態 0 與 1 都使用狀態 0 的設定屬性內容。

## 13.2. 多狀態指示燈

### 13.2.1. 概要

[多狀態指示燈] 物件利用字組暫存器內的數據，顯示相對的狀態與圖形(最高可支援 256 種狀態的顯示)。當暫存器內的數值為 0 時，顯示 [狀態 0]；當數值為 1 時，則顯示 [狀態 1]，依此類推。

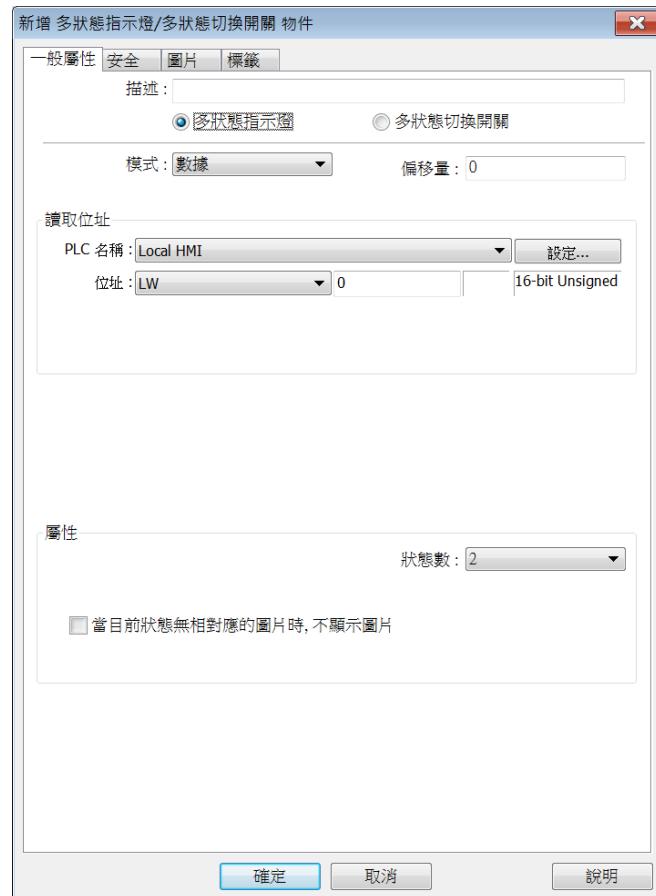


### 13.2.2. 設定



按下工作列的 [物件] » [多狀態指示燈] 按鈕後即會開啟 [多狀態指示燈] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [多狀態指示燈] 物件。

## 一般屬性設定



設定	描述
描述	使用者可為此物件描述相關訊息。 <b>多狀態指示燈 / 多狀態切換開關</b> 可與 [多狀態切換開關] 功能互相轉換。
模式 / 偏移量	[多狀態指示燈] 物件提供三種不同的數據顯示模式。 <b>數據</b> 直接利用暫存器內的數據加減 [偏移量] 的結果做為物件目前的狀態。如下圖所示，當寫入一個數值 3 至暫存器地址 LW-200 時，因為有偏移量 3，所以地址 LW-200 的物件圖片會顯示狀態 0 (數值 3 - 偏移量 3)。



### LSB

此模式首先會將暫存器內的數據先轉換為 2 進制，接著使用不為 0 的最低位元決定物件目前的狀態。以下表數據為例：

十進制	二進制	顯示的狀態
0	00000000000000000000	全部 bit 皆為 0，則顯示狀態 0
1	00000000000000000001	不是 0 的最低位元為 bit 0，此時顯示狀態 1
2	00000000000000000010	不是 0 的最低位元為 bit 1，此時顯示狀態 2
3	00000000000000000011	不是 0 的最低位元為 bit 0，此時顯示狀態 1
4	00000000000000000100	不是 0 的最低位元為 bit 2，此時顯示狀態 3
7	00000000000000000111	不是 0 的最低位元為 bit 0，此時顯示狀態 1
8	00000000000000001000	不是 0 的最低位元為 bit 3，此時顯示狀態 4
16	000000000000100000	不是 0 的最低位元為 bit 4，此時顯示狀態 5
32	00000000001000000000	不是 0 的最低位元為 bit 5，此時顯示狀態 6
64	00000000010000000000	不是 0 的最低位元為 bit 6，此時顯示狀態 7
128	00000000100000000000	不是 0 的最低位元為 bit 7，此時顯示狀態 8
256	00000001000000000000	不是 0 的最低位元為 bit 8，此時顯示狀態 9
512	00000010000000000000	不是 0 的最低位元為 bit 9，此時顯示狀態 10
1024	00000100000000000000	不是 0 的最低位元為 bit 10，此時顯

		示狀態 11
2048	0000100000000000	不是 0 的最低位元為 bit 11，此時顯示狀態 12
4096	0001000000000000	不是 0 的最低位元為 bit 12，此時顯示狀態 13
8192	0010000000000000	不是 0 的最低位元為 bit 13，此時顯示狀態 14
16384	0100000000000000	不是 0 的最低位元為 bit 14，此時顯示狀態 15
32768	1000000000000000	不是 0 的最低位元為 bit 15，此時顯示狀態 16

### 位元組合

指示燈的狀態根據位元的狀態顯示，更改屬性的狀態數可增加不同裝置的讀取位址。PLC 1 表示最低位，PLC 2 次之，以此類推。最多可讀取 4 個位址，共 16 個狀態數。



### 週期轉換狀態

物件的狀態會依照固定的頻率依序變換狀態。使用者可以利用 [頻率] 設定狀態改變頻率。

#### 讀取位址

點選 [設定] 後選擇字組暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制多狀態指示燈物件。使用者也可在 [一般屬性] 頁中設定位址。

#### 屬性

#### 狀態數

物件顯示的狀態數目。狀態從 0 開始編號，能顯示的最大狀態編號為設定的 [狀態數] - 1，當要求顯示超過設定的狀態數時，系統會顯示最後一個狀態。例如設定 [狀態數] 為 8，則顯示的狀態依序為 0，1，2，...，7，若要求暫存器顯示狀態 8(含) 以上的狀態時，顯示的圖片僅顯示狀態 7。

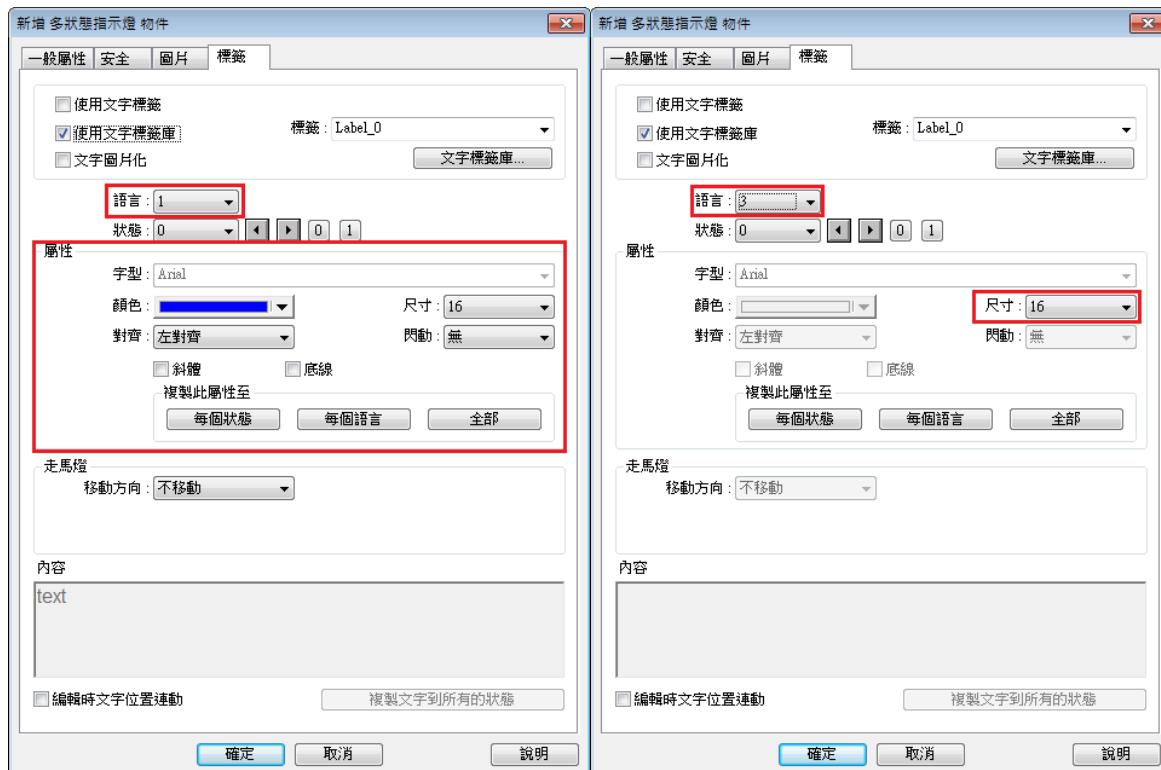
## 物件

當目前狀態無相對應的圖片時，不顯示圖片

若勾選，當圖片數目不足以顯示全部狀態時，將不顯示圖片。反之則顯示最後一個狀態。



- 在標籤頁中，語言 1 能夠改變字型相關屬性設定，但語言 2~8 只能改變字的尺寸，其他屬性設定皆與語言 1 相同。



### 13.3. 位元狀態設定

#### 13.3.1. 概要

[位元狀態設定] 用於設定位元暫存器的狀態。此物件提供手動操作與自動執行兩種操作模式。使用手動操作模式，按壓此按鈕可以將暫存器的狀態設定為 ON 或 OFF。

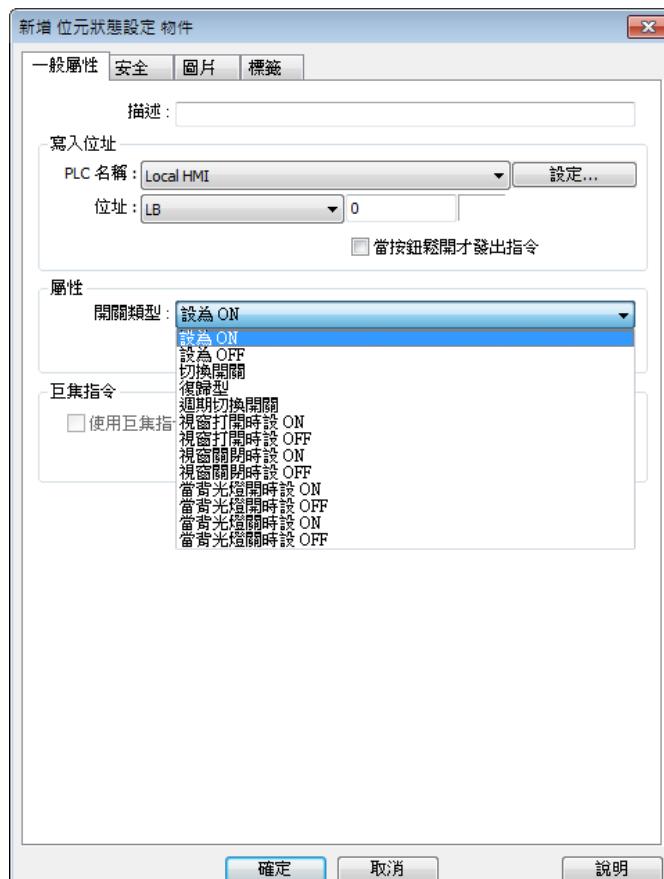
若使用自動執行模式，則在某些特定條件下會自動執行指定的動作，使用此種操作模式，即使按壓此按鈕也不會有任何影響。

#### 13.3.2. 設定



按下工作列的 [物件] » [位元狀態設定] 按鈕後即會開啟 [位元狀態設定] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [位元狀態設定] 物件。

#### 一般屬性設定



設定	描述
寫入位址	點選 [設定] 後選擇位元暫存器設備類型的 [PLC 名稱]，[位址]，[設

[備類型]，[系統暫存器]，[索引暫存器]來控制位元狀態設定物件。使用者也可在 [一般屬性] 頁中設定位址。

#### 當按鈕鬆開才發出指令

使用此設定表示在按下物件後，必須完全鬆開按壓動作，物件定義的操作模式才會被執行。如未使用此項設定，只要一碰觸此區域，將立刻執行物件的動作。若選擇使用復歸型模式，將不支援此項功能。

屬性	開關類型	描述
	設為 ON	按壓此物件後，所指定暫存器的狀態將被設定為 ON。
	設為 OFF	按壓此物件後，所指定暫存器的狀態將被設定為 OFF。
	切換開關	按壓此物件後，所指定暫存器的狀態將被反向設定，即 ON→OFF, OFF→ON。
	復歸型	按壓此物件後，所指定暫存器的狀態將先被設定為 ON，但手放開後，狀態將被設定為 OFF。
	週期切換開關	所指定暫存器的狀態將在 ON 與 OFF 間週期性切換，此模式為自動執行。可設定的週期為 0.1 秒 ~ 25.5 秒。
	視打開時設 ON	物件所在位置的視窗被打開時，所指定暫存器的狀態將自動被設定為 ON。
	視窗打開時 OFF	物件所在位置的視窗被打開時，所指定暫存器的狀態將自動被設定為 OFF。
	視窗關閉時設 ON	物件所在位置的視窗被關閉時，所指定暫存器的狀態將自動被設定為 ON。
	視窗關閉時設 OFF	物件所在位置的視窗被關閉時，所指定暫存器的狀態將自動被設定為 OFF。
	當背光燈開時設 ON (不支援 cMT-SVR)	當背光燈打開時，所指定暫存器的狀態將自動被設定為 ON。
	當背光燈開時設 OFF (不支援 cMT-SVR)	當背光燈打開時，所指定暫存器的狀態將自動被設定為 OFF。
	當背光燈關時設 ON (不支援 cMT-SVR)	當背光燈關閉時，所指定暫存器的狀態將自動被設定為 ON。
	當背光燈關時設 OFF (不支援 cMT-SVR)	當背光燈關閉時，所指定暫存器的狀態將自動被設定為 OFF。
巨集指令	[位元狀態設定] 物件可以搭配執行巨集命令。選擇此項功能前需先建立巨集命令。	



如何建立巨集命令請參考《18 巨集指令說明》

#### 觸發模式

當物件的操作模式，選擇【切換開關】時，設定執行巨集命令的條件，可以選擇狀態由 OFF 變為 ON 或由 ON 變為 OFF 時，才執行巨集命令，也可選擇狀態改變時(OFF<->ON)，即執行巨集命令。



#### Note

- 在標籤頁中，若勾選【ON=OFF(使用狀態)】，則狀態 0 與 1 都使用狀態 0 的設定屬性內容。
- cMT-SVR 不支援位元狀態設定的【週期切換開關】屬性使用 PLB 與 PLW\_Bit 以外的地址類型。
- 【安全】設定頁的選項會因【屬性】設定而有所不同，當位元狀態設定物件使用【視窗開啟時設 ON/OFF】功能時，不支援【安全】設定頁中的任何選項。
- 若有啟用執行【巨集指令】，物件在執行【寫入位址】的動作與執行【巨集指令】兩者間並無絕對的執行順序。若要確保【寫入位址】動作與【巨集指令】的執行順序，請改用複合式多功能按鈕物件。

## 13.4. 多狀態設定

### 13.4.1. 概要

【多狀態設定】用於設定字組暫存器的數據。此物件提供手動操作與自動執行兩種操作模式。使用手動操作模式，按壓此按鈕可以設定暫存器內的數據。

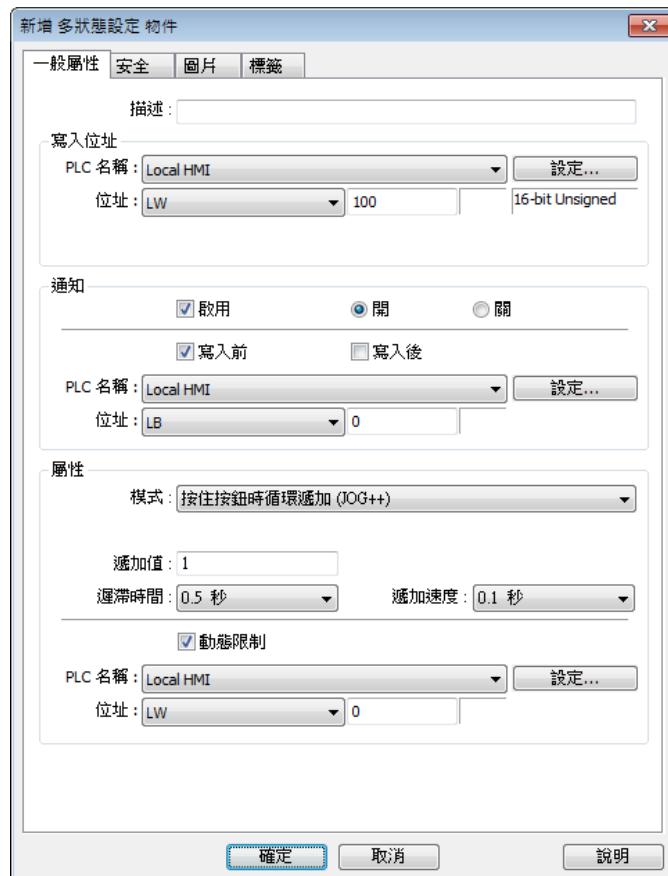
若使用自動執行模式，則在某些特定條件下會自動執行指定的動作，使用此種操作模式，即使按壓此按鈕也不會有任何影響。

### 13.4.2. 設定



按下工作列的【物件】»【多狀態設定】按鈕後即會開啟【多狀態設定】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【多狀態設定】物件。

#### 一般屬性設定



設定	描述
寫入位址	<p>點選 [設定] 後選擇字組暫存器設備類型的[PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制多狀態設定物件。使用者也可在 [一般屬性] 頁中設定位址。</p> <p><b>當按鈕鬆開才發出指令</b></p> <p>使用此項設定表示在按壓此按鈕後，必須完全離開此區域才會執行物件定義的動作。如未使用此項設定，則只要一按壓此鈕，將立刻執行物件定義的動作。</p>
通知	<p>使用此項設定，則在使用手動操作模式時，在完成動作後可以連帶設定此項目所指定暫存器的狀態，使用 [開] 與 [關] 選擇要設定的狀態。</p> <p><b>[寫入前] / [寫入後]</b></p> <p>在寫入動作前 / 後設定所指定暫存器的狀態。</p>
屬性	<p><b>模式</b></p> <p>選擇物件的動作模式。可以選擇的模式請見以下範例 2。</p> <p><b>動態限制</b></p> <p>上下限可由指定暫存器設定，請見以下範例 1。</p>

## 範例 1

上下限可由指定暫存器設定。當寫入位址為 LW-n，則上/下限會根據以下的規則自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT/cMT X)
動態限制位址	LW-n	LW-n	LW-n
下限	LW-n	LW-n	LW-n
上限	LW-n+1	LW-n+2	LW-n+4

以下表為例，當 [暫存器位址] 為 LW-100 時，則上/下限的位址會自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT/cMT X)
動態限制位址	LW-100	LW-100	LW-100
下限	LW-100	LW-100	LW-100
上限	LW-101	LW-102	LW-104

## 範例 2

可以選擇模式如下：

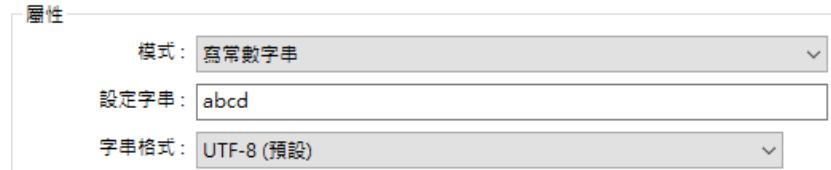
- 寫入常數

設定常數功能。每按壓一次物件，[設定常數] 中的設定值將寫至指定的暫存器中。常數的型態可為 16-bit BCD、32-bit BCD、...、64-bit Double 等。以下圖為例，當按壓此按鈕後，會將數值 12 寫入指定的暫存器中。



### ● 寫常數字串 (限 cMT/cMT X 機型)

設定常數字串功能。每按壓一次物件，[設定字串] 中的設定值將寫至指定的暫存器中。常數字串的型態可為 UTF-8、Unicode 與 Latin-1 等。以下圖為例，當按壓此按鈕後，會將字串 abcd 寫入指定的暫存器中。



### ● 遞加 (JOG+)

加值功能。每按壓一次物件，所指定暫存器內的數據將加上 [遞加值] 中設定的增量值，但增值的結果將不超過 [上限值] 中的設定值。以下圖為例，每按壓一次此按鈕後，會將指定的暫存器中的數值+1，直至抵達上限值 10。



### ● 遞減 (JOG-)

減值功能。每按壓一次物件，所指定暫存器內的數據將減去 [遞減值] 中設定的減量值，但是減值的結果不會低於 [下限值] 中的設定值。以下圖為例，每按壓一次此按鈕後，會將指定的暫存器中的數值-1，直至抵達下限值 0。



### ● 迴圈 (自訂範圍) (限 cMT / cMT X 機型)

手動週期性迴圈功能。每按壓一次物件，將所指定暫存器內的數據加上 [遞加值] 中的設定值，直到暫存器內的數據等於 [最大值]；接著按壓一次物件，將暫存器內的數據減去 [遞加值] 中的設定值，直到暫存器內的數據等於 [最小值]。以下圖為例，系統會自動將指定的暫存器中的數值以手動按壓的方式+1，直到抵達最大值 10，接著再以相同的手動按壓方式-1 直到等於最小值 0，如此週而復始的執行。



### ● 按住按鈕時遞加 (JOG++)

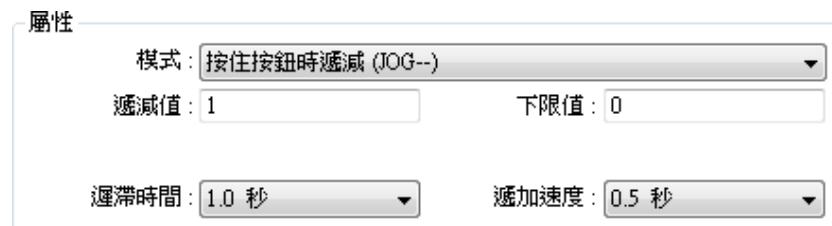
按住按鈕時遞加功能。若按壓物件超過 [遲滯時間] 設定時間，則所指定暫存器內的數據將以 [遞加速度] 所設定的速度，每次增加 [遞加值] 中設定的增量值，但增量的結果將不超過 [上限

值] 中的設定值。以下圖為例，每按壓一次此按鈕後，會將指定的暫存器中的數值+1，若按住此按鈕的時間超過 1.0 秒後，會以每 0.5 秒的速度持續+1 直到抵達上限值 10。



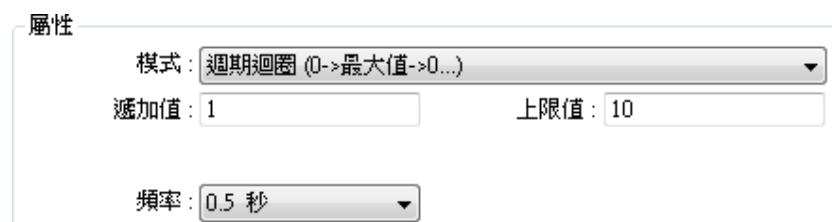
#### ● 按住按鈕時遞減 (JOG--)

按住按鈕時遞減功能。若按壓物件超過 [遲滯時間] 的設定時間，則所指定暫存器內的數據將以 [遞加速度] 所設定的速度，每次減少 [遞減值] 中設定的減量值，但減值的結果不會低於 [下限值] 中的設定值。以下圖為例，每按壓一次此按鈕後，會將指定的暫存器中的數值-1，若按住此按鈕的時間超過 1.0 秒後，會以每 0.5 秒的速度持續-1 到抵達下限值 0。



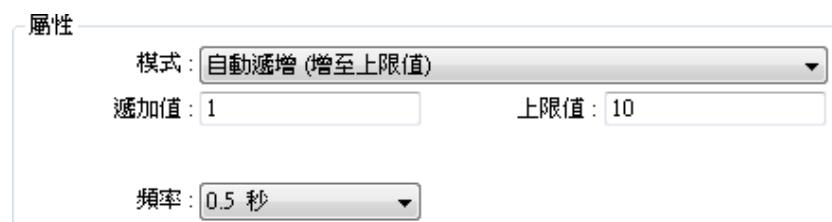
#### ● 週期迴圈 (0->最大值->0)

週期性遞加功能。[多狀態設定] 物件會使用 [頻率] 設定的週期與 [遞加值] 中設定的增量值，自動增量所指定暫存器內的數據，但增量的結果將不超過 [上限值] 中的設定值。以下圖為例，系統會自動將指定的暫存器中的數值以每 0.5 秒的頻率+1，直到抵達上限值 10，接著數值會返回 0 再重新持續+1。



#### ● 自動遞增 (增至上限值)

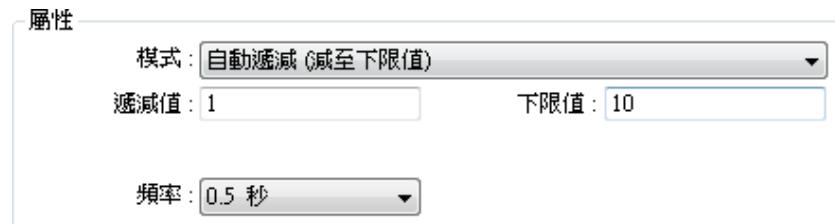
週期性遞增功能。[多狀態設定] 物件會使用 [頻率] 設定的週期，自動將所指定暫存器內的數據加上 [遞加值] 中設定的增量值，當結果等於 [上限值] 時自動停止。以下圖為例，系統會自動將指定的暫存器中的數值以每 0.5 秒的頻率+1，直到抵達上限值 10 後停止。



#### ● 自動遞減 (減至下限值)

週期性遞減功能。[多狀態設定] 物件會使用 [頻率] 設定的週期，自動將所指定暫存器內的數據減去 [遞減值] 中設定的減量值，當結果等於 [下限值] 時自動停止。以下圖為例，系統會自動將

指定的暫存器中的數值以每 0.5 秒的頻率-1，直到抵達下限值 10 後停止。



#### ● 週期迴圈 (自訂範圍)

週期性迴圈功能。[多狀態設定] 物件會使用 [頻率] 設定的週期，每次將所指定暫存器內的數據加上 [遞加值] 中的設定值，直到暫存器內的數據等於 [上限值]；接著使用相同的週期，將暫存器內的數據減去 [遞加值] 中的設定值，直到暫存器內的數據等於 [下限值]。以下圖為例，系統會自動將指定的暫存器中的數值以每 0.5 秒的頻率+1，直到抵達上限值 10，接著再以相同的頻率-1 直到等於下限值 0，如此週而復始的執行不停止。



#### ● 週期遞加 (從低到高)

步進功能。[多狀態設定] 物件會使用 [頻率] 設定的週期，每次將所指定暫存器內的數據加上 [遞加值] 中的設定值，直到暫存器內的數據等於 [最大值]，接著會將暫存器內的數據復歸為 [最小值]，並重複先前的動作，讓數據一直保持動態變化。以下圖為例，系統會自動將指定的暫存器中的數值以每 0.5 秒的頻率+1，直到抵達上限值 10，接著再返回下限值 0 重新遞增，如此週而復始的執行不停止。



#### ● 週期遞減 (從高到低)

步退功能。[多狀態設定] 物件會使用 [頻率] 設定的週期，每次將所指定暫存器內的數據減去 [遞減值] 中的設定值，直到暫存器內的數據等於 [最小值]，接著會將暫存器內的數據復歸為 [最大值]，並重複先前的動作，讓數據一直保持動態變化。以下圖為例，系統會自動將指定的暫存器中的數值以每 0.5 秒的頻率-1，直到抵達下限值 0，接著再返回上限值 10 重新遞減，如此週而復始的執行不停止。



- 視窗打開時設定 / 視窗關閉時設定

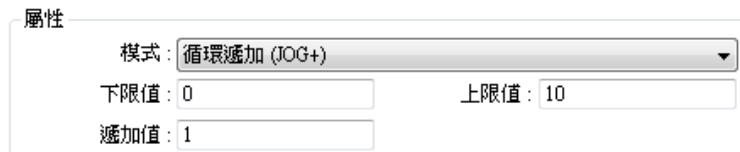
開啟 / 關閉物件所在位置的視窗時，會將 [設定常數] 中的設定值自動寫至指定的暫存器中。若 [設定常數] 設為 5，當該頁的視窗被開啟 / 關閉時，系統會自動將數值 5 寫入暫存器中。

- 當背光燈開時設定 / 當背光燈關時設定 (不支援 cMT-SVR)

當背光燈原處在關閉 / 開啟狀態，若恢復為相反狀態時，會將 [設定常數] 中的設定值自動寫至指定的暫存器中。若 [設定常數] 設為 5，當該頁的背光燈狀態改變時，系統會自動將數值 5 寫入暫存器中。

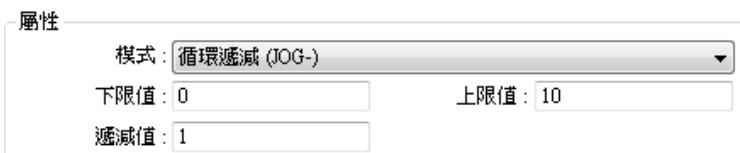
- 循環遞加 (Cyclic JOG+)

加值功能。每按壓一次物件，所指定暫存器內的數據將加上 [遞加值] 中設定的增量值。當增量值達到上限時，會復歸回下限再重新遞增。以下圖為例，每按壓一次此物件後，會將指定的暫存器中的數值 +1，當抵達上限值 10 後會自動復歸回 0 再遞增執行。



- 循環遞減 (Cyclic JOG-)

減值功能。每按壓一次物件，所指定暫存器內的數據將減去 [遞減值] 中設定的減量值。當減量值達到下限時，會復歸回上限再重新遞減。以下圖為例，每按壓一次此物件後，會將指定的暫存器中的數值 -1，當抵達下限值 0 後會自動復歸回 10 再遞減執行。



- 按住按鈕時循環遞加 (JOG++)

持續遞加功能。當按住按鈕的時間超過 [遲滯時間] 時，此物件會根據 [遞加速度] 的設定將指定暫存器的數據持續的遞加至上限值，之後會復歸回下限值。以下圖為例，每按壓一次此按鈕後，會將指定的暫存器中的數值 +1，若按住此按鈕的時間超過 0.5 秒後，會以每 0.1 秒的速度持續 +1 直到抵達上限值 10，接著會復歸回 0 再遞增執行。



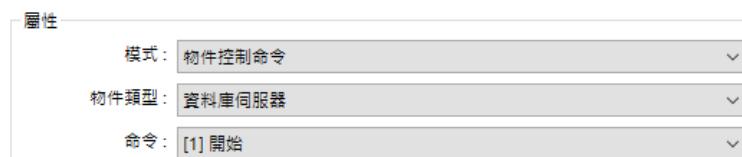
- 按住按鈕時循環遞減 (JOG- -)

持續遞減功能。當按住按鈕的時間超過 [遲滯時間] 時，此物件會根據 [遞減速度] 的設定將指定暫存器的數據遞減至下限值，之後會復歸回上限值。以下圖為例，每按壓一次此按鈕後，會將指定的暫存器中的數值-1，若按住此按鈕的時間超過 0.5 秒後，會以每 0.1 秒的速度持續-1 直到抵達下限值 0，接著會復歸回 10 再遞增執行。



#### ● 物件控制命令 (限 cMT / cMT X 機型)

提供各功能控制位址可使用的命令供選擇 (同時顯示命令對應的數值)。將 [寫入地址] 設定為功能的控制地址，此設定物件便能依照命令設定寫入對應值，達到功能控制目的。

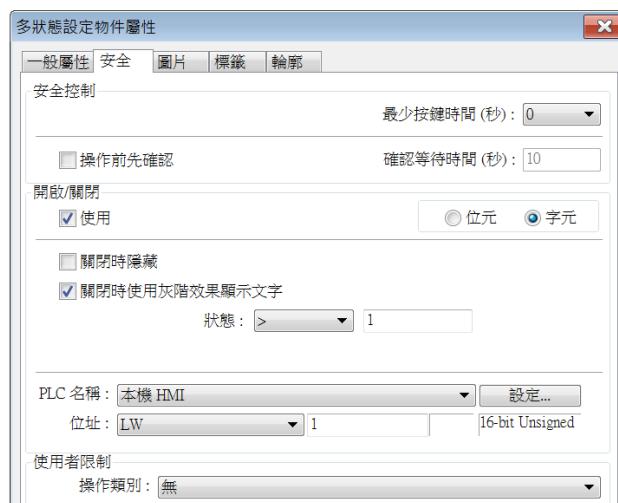


#### ● 系統標籤命令 (限 cMT / cMT X 機型)

當寫入位址為特定系統標籤時，例如多國語言設定 LW-9134，即可選擇欲設定的命令。



## 安全設定



設定	描述
開啟/關閉	若勾選 [使用] 並選擇 [字組]，此物件是否允許被操作，將決定於一個指定字組位址的 [狀態]。如圖中所示，則必須在 LW-1 的數值大於

1 時，才允許操作此物件。

#### 關閉時隱藏

當指定的字組位址的數值不符合 [狀態] 時，物件會被隱藏。

#### 關閉時使用灰階效果顯示文字

物件的標籤文字會在指定的字組位址的數值不符合 [狀態] 時，以灰階樣式顯示。



#### 狀態

可設定指定字組位址的條件，有 > 、 < 、 == 、 <> 、 >= 或 <= 可以選擇。其中 == 與 <> 可以設定 [允許誤差]。

舉例來說：

關閉時隱藏

關閉時使用灰階效果顯示文字

狀態 :	<input type="button" value="=="/>	<input type="text" value="10"/>
允許誤差 :	<input type="text" value="1"/>	

當指定字組位址的數值大於等於 11，或小於等於 9 時，物件就會被關閉並隱藏。

#### Note

- cMT-SVR 不支援多狀態設定的 [週期迴圈]、[自動遞增]、[自動遞減]、[週期遞加]、[週期遞減] 等屬性使用 PLW 以外的地址類型。

## 13.5. 功能鍵

### 13.5.1. 概要

[功能鍵] 物件提供視窗切換、鍵盤製作、巨集執行及畫面列印等功能，同時也可用於設定 USB 安全金鑰。

若使用 cMT / cMT X 系列 HMI，無法透過 cMT Viewer 使用 [畫面擷取]、[匯入使用者資料/使用 USB 安全金鑰] 的功能鍵物件。

### 13.5.2. 設定



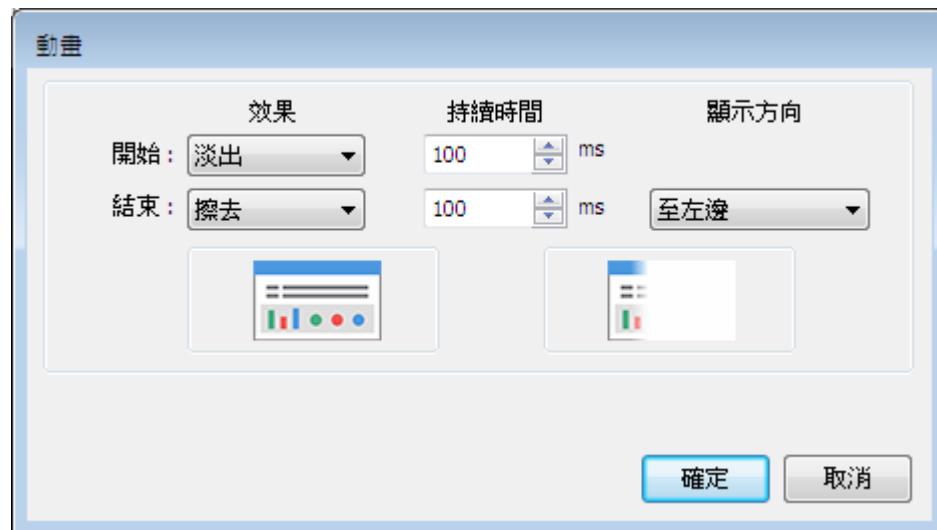
按下工作列的 [物件] » [功能鍵] 按鈕後即會開啟 [功能鍵] 物件屬性對話窗，正確設定各項屬性後按下確定鍵，即可新增一個 [功能鍵] 物件。

#### 一般屬性設定



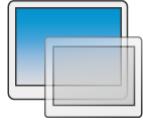
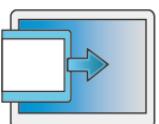
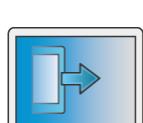
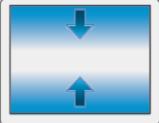
設定	描述
當按鈕鬆開才發出指令	使用此選項表示必須在釋放按壓物件的動作後，選擇的動作才會被執行。若未選擇，則在碰觸物件後，將立刻執行選擇的動作。
視窗切換	<p>切換基本視窗：切換基本視窗。</p> <p>切換公共視窗：切換公用視窗。</p> <p>彈出視窗：呼叫其他視窗。此時呼叫出的視窗必定在基本視窗的上面。使用此功能可以選擇是否使用 [當父視窗被關閉時結束彈出視窗]，參考下圖。選擇此屬性則呼叫出的視窗會在發生換頁動作時自動消失，否則使用者必須自行在被呼叫出的視窗上設計 [關閉視窗] 功能鍵來關閉此視窗。</p>
	

**動畫設定 (cMT / cMT X 系列)：**cMT / cMT X 系列使用功能鍵呼叫其他視窗時，能夠設定動畫效果。點選 [動畫設定]，能夠設定不同的視窗彈出效果。



可設定視窗 [開始] 與 [結束] 的動畫效果。

效果	樣式
----	----

淡出	
飛入	
飄入	
擦去	
分割	
環狀	
時鐘	
縮放	
旋轉	
推入	

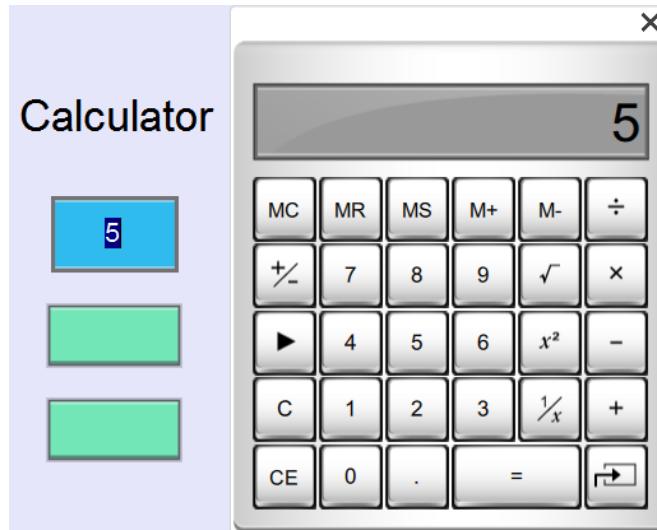
[持續時間] 可以設定視窗開始與結束的速度。

[顯示方向] 可以設定視窗開始與結束的出入方向。

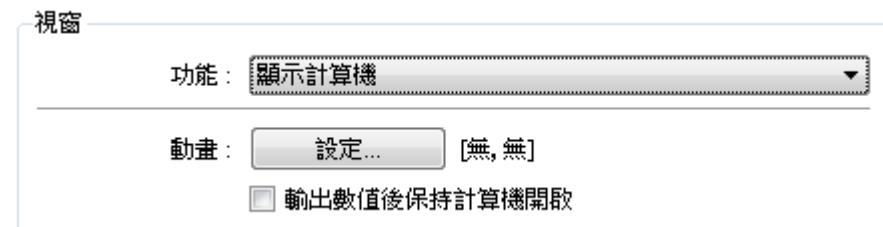
**返回上一個視窗**：返回前一頁基本視窗。例如當由“視窗 10”切換到“視窗 20”時，使用此功能可以再返回“視窗 10”。此功能只對基本視窗有效。

**關閉視窗**：關閉在基本視窗上被呼叫出的視窗，包括訊息視窗。

**顯示計算機視窗(cMT / cMT X 系列)**：可直接呼叫出一個外觀固定的計算機物件，可以在人機上執行簡單計算並輸出至目標數值物件上。



使用此功能可以選擇是否使用 [輸出數值後保持計算機開啟]，參考下圖。



請點選此圖示觀看教學影片，請先確定已連上網路線。

### 鍵盤輸入

用來作為鍵盤的輸入訊號，主要用在 [數值] 與 [字元] 物件需要使用鍵盤來輸入數字或文字的場合。

**Enter:** 與鍵盤的輸入 (Enter) 動作相同。

**Backspace:** 與鍵盤的後退刪除 (Backspace) 動作相同。

**Clear:** 清除暫存器中已輸入的資料。

**Esc:** 與使用 [關閉視窗] 功能相同，可用來關閉彈跳出的鍵盤視窗。

**Delete:** 與鍵盤的刪除 (Delete) 動作相同，可將游標右方的一個字元刪除。

**Left:** 與鍵盤的←動作相同，可將游標向左移動一個字元。

**Right:** 與鍵盤的→動作相同，可將游標向右移動一個字元。

**Line feed:** 换行。

**Inc:** 將數值+1。

**Dec:** 將數值-1。

**ASCII/UNICODE:** 設定鍵盤的輸入字元。

**觸發巨集指令**

選擇此項功能，將執行指定的巨集命令，選擇此項功能前需先建立巨集命令。

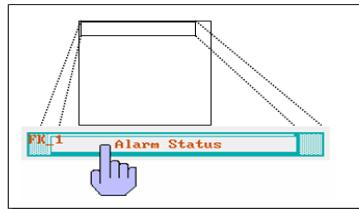
 如何建立巨集命令請參考《18 巨集指令說明》

觸發巨集指令

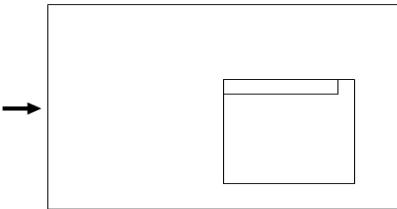
巨集指令 : [ID:000] macro\_0

**視窗控制條**

當彈出的視窗無視窗控制條時，若需要移動視窗，則先點一下此物件，在移動的目的地再點一下，則視窗就會被移動到指定的位置。



首先啓動視窗控制列



接著在要移向的地方觸控一下，該視窗就移動到該位置了



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

**畫面擷取至  
USB 磁碟, SD 卡  
或印表機**

此項功能用來擷取當前的畫面。要選擇此項功能前需先在【系統參數設定】»【HMI 屬性】中選擇所使用的印表機類型。使用單色印表機時，勾選【灰階效果】可以提升畫面的辨識度，但也會影響文字的顯示效果，因此如果是強調文字的列印效果，請不要使用灰階功能。

**畫面列印至 USB 磁碟, SD 卡或印表機**

畫面列印

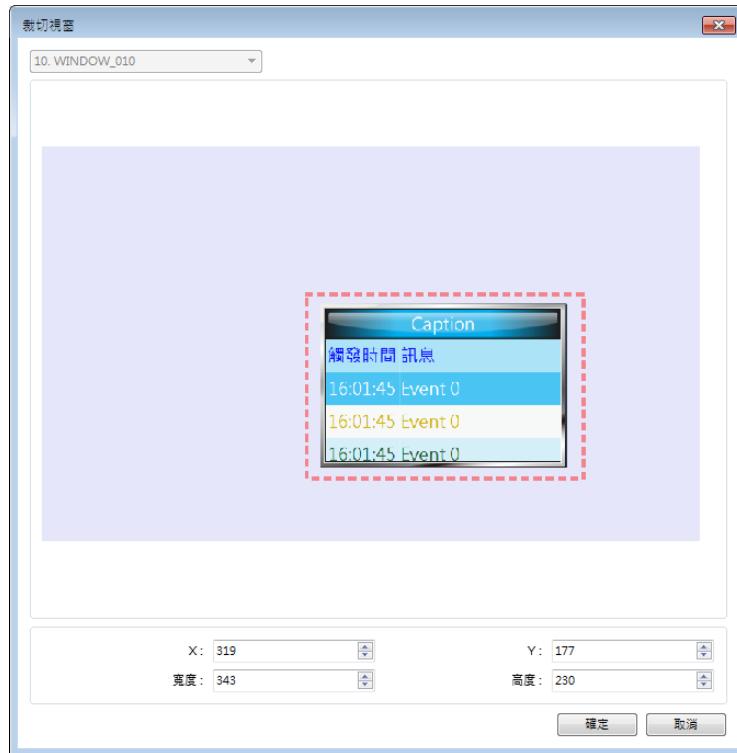
印表機 : HP PCL 系列 (USB)

圖形旋轉 90 度

模式 : 彩色

**裁切視窗**

畫面擷取可自訂輸出畫面的範圍。



### 確認所有事件

**(警報)  
(cMT / cMT X  
系列)**

按下此功能鍵時將一次確認所有事件。

**匯入使用者資料/使用 [USB  
安全金鑰]**

用來匯入進階安全的使用者帳號或 e-mail 的連絡人。也可設定為使用 USB 金鑰登入。



**資料位置：**提供從 USB 碟、SD 卡讀取兩個選項。

**帳號匯入方式：**選擇 [覆蓋]，HMI 內將只保存此次匯入的帳號資料，若是選擇 [附加]，HMI 內帳號資料將保留，並加入此次匯入的新帳號資料。

**注意：**當選用 [附加] 模式，且欲匯入的使用者名稱已存在於工程檔案內，該匯入動作將無法被執行。

---

**匯入使用者帳號後刪除檔案**：將 USB 內的使用者帳號匯入後即刪除來源資料，可確保資料不洩漏。

---

**通知**

使用此項設定，則在完成動作後可以連帶設定此項目所指定暫存器的狀態，使用 [開] 與 [關] 選擇要設定的狀態。

---



■ 當匯入 e-mail 的聯絡人時，僅會以 [覆蓋] 的方式匯入，因此已存在的聯絡人皆會先被刪除後才匯入新的聯絡人。



詳細資訊請參考《6 視窗》、《12 鍵盤的設計與使用》、《36 管理員工具》。

## 13.6. 位元狀態切換開關

### 13.6.1. 概要

【位元狀態切換開關】為【位元狀態指示燈】物件與【位元狀態設定】物件的組合。此物件除了可以用來顯示暫存器的狀態外，也可以利用這個物件在視窗上定義一個碰觸區域，按壓此區域可以設定所指定暫存器的狀態為 ON 或 OFF。

### 13.6.2. 設定



按下工作列的【物件】»【位元狀態切換開關】按鈕後即會開啟【位元狀態切換開關】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【位元狀態切換開關】物件。

#### 一般屬性設定



設定	描述
描述	使用者可為此物件描述相關訊息。 <b>位元狀態指示燈 / 位元狀態切換開關</b> 可與 [位元狀態指示燈] 功能互相轉換。
讀取/寫入使用 不同位址	可以分開設定數據的讀取位址與寫入位址。
讀取位址	點選 [設定] 後選擇位元暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制顯示位元狀態的切換開關狀態物件。使用者也可在 [一般屬性] 頁中設定位址。
	<b>輸出反向</b> 可以將讀取的狀態作反向顯示，例如位元的狀態實際上為 OFF，但勾選了 [輸出反向] 後會顯示為 ON。 當未勾選 [讀取/寫入使用不同的位址] 時，讀取位址欄位名稱會顯示為 [讀取/寫入位址]。
寫入位址	點選 [設定] 後選擇位元暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制位元狀態切換開關物件。使用者也可在 [一般屬性] 頁中設定位址。此暫存器可以與 [讀取位址] 所指定的暫存器相同亦或不同。
	<b>當按鈕鬆開才發出指令</b> 使用此設定表示在按下物件後，必須完全鬆開按壓動作，物件定義的操作模式才會被執行。如未使用此項設定，只要一碰觸此區域，將立刻執行物件的動作。若選擇使用復歸型模式，將不支援此項功能。
屬性	開關類型
	設為 O
	按壓此物件後，所指定暫存器的狀態將被設定為 ON
	設為 OF
	按壓此物件後，所指暫存器的狀態將被設定為 OFF。
	切換開關
	按壓此物件後，所指定暫存器的狀態將被反相。當狀態為 ON 時，會被顯示為 OFF。當狀態為 OFF 時，則會被顯示為 ON。
	復歸型
	按壓此物件後，所指定暫存器的狀態將先被設定為 ON，但手放開後，狀態將被設定為 OFF。
巨集指令	[位元狀態切換開關] 物件可以搭配執行巨集命令。選擇此項功能前需先建立巨集命令。  如何建立巨集命令請參考《18 巨集指令說明》

 **Note**

- 若有啟用執行 [巨集指令]，物件在執行 [寫入位址] 的動作與執行 [巨集指令] 兩者間並無絕對的執行順序，若要確保 [寫入位址] 動作與 [巨集指令] 的執行順序，請改用複合式多

功能按鈕物件。

## 13.7. 多狀態切換開關

### 13.7.1. 概要

[多狀態切換開關] 物件為 [多狀態指示燈] 物件與 [多狀態設定] 物件的組合。此物件除了可以利用暫存器內的數據顯示不同的狀態外，也可以利用這個物件在視窗上定義一個碰觸區域，按壓此區域可以設定所指定暫存器內的數據。

### 13.7.2. 設定



按下工作列的 [物件] » [多狀態切換開關] 按鈕後即會開啟 [多狀態切換開關] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [多狀態切換開關] 物件。

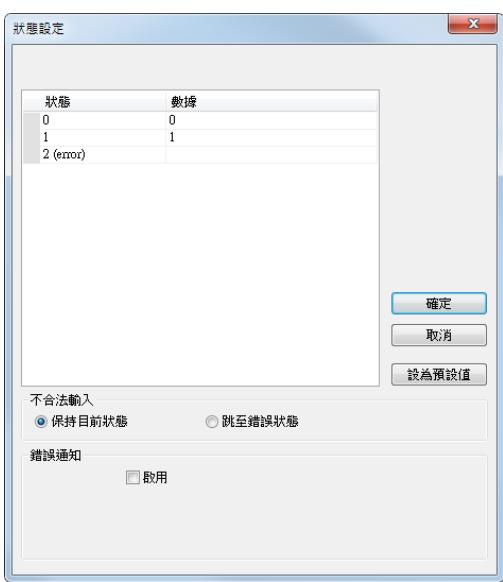
#### 一般屬性設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



設定	描述								
<b>描述</b>	使用者可為此物件描述相關訊息。 <b>多狀態指示燈 / 多狀態切換開關</b> 可與 [多狀態指示燈] 功能互相轉換。								
<b>模式 / 偏移量</b>	提供不同的數據顯示模式： 數據、LSB。  詳細資訊請參考《13.2 多狀態指示燈》。								
<b>讀取/寫入使用不同位址</b>	使用者可以分開設定數據的讀取位址與寫入位址。								
<b>讀取位址</b>	點選 [設定] 後選擇字組暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制顯示多狀態切換開關狀態物件。使用者也可在 [一般屬性] 頁中設定位址。 當未勾選 [讀取/寫入使用不同的位址] 時，讀取位址欄位名稱會顯示為 [讀取/寫入位址]。								
<b>寫入位址</b>	點選 [設定] 後選擇字組暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制多狀態切換開關物件。使用者也可在 [一般屬性] 頁中設定位址。此暫存器可以與 [讀取位址] 所指定的暫存器相同亦或不同。 <b>當按鈕鬆開才發出指令</b> 使用此項設定表示在按壓此按鈕後，必須完全離開此區域才會執行物件定義的動作。如未使用此項設定，則只要一按壓此鈕，將立刻執行物件定義的動作。								
<b>屬性</b>	<p><b>模式</b> 選擇物件的操作方式。可以選擇的模式請見以下範例 1。</p> <p><b>使用狀態設定</b> 使用者可修改狀態對應的數值，亦可使用當有不合法的數值輸入時的動作狀態和通知指定位元切換狀態。</p>  <table border="1" data-bbox="516 1545 865 1650"> <thead> <tr> <th>狀態</th> <th>數據</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> <tr> <td>2 (error)</td> <td></td> </tr> </tbody> </table> <p>不合法輸入  <input checked="" type="radio"/> 保持目前狀態   <input type="radio"/> 跳至錯誤狀態</p> <p>錯誤通知  <input type="checkbox"/> 啟用</p>	狀態	數據	0	0	1	1	2 (error)	
狀態	數據								
0	0								
1	1								
2 (error)									

**保持目前狀態**

若輸入超出範圍的數值，多狀態切換開關會保持目前狀態。

**跳至錯誤狀態**

若輸入超出範圍的數值，多狀態切換開關會跳到錯誤狀態。

**錯誤通知**

當輸入無效的數值時，可以自動設定所指定位址的狀態。

**寫入成功後傳送通知**

當寫入 PLC 的動作成功後，將指定位元暫存器的狀態設為開 / 關。

**錯誤管理  
(cMT / cMT X  
系列)**

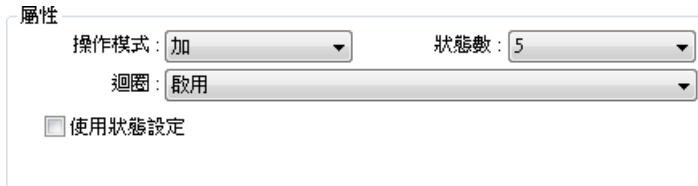
當有不合法的數值輸入時的動作狀態和通知指定位元切換狀態。用途與 [使用狀態設定] 雷同，但可不必設置各狀態對應的數據。

**範例 1**

可以選擇模式如下：

**● 加 (JOG+)**

遞加功能。每按壓一次物件，所指定暫存器內的數據+1，但增值的結果將不超過 [狀態數]。若 [啟用] 迴圈，則抵達最大狀態後會復歸回最低狀態 0。以下圖為例，若操作模式選擇 [加]，狀態數為 5 且 [啟用] 迴圈，則每按壓一次此物件，狀態會從狀態 0 會往上+1 直至狀態 4 ([狀態數] - 1)，然後復歸回狀態 0 重新遞加。

**● 減 (JOG-)**

遞減功能。每按壓一次物件，所指定暫存器內的數據-1 直至 0。若 [啟用] 迴圈，則抵達最大狀態後會復歸回最高狀態。以下圖為例，若操作模式選擇 [減]，狀態數為 5 且 [啟用] 迴圈，則每按壓一次此物件，會往下-1 直至狀態 0，然後復歸回最高狀態 4 ([狀態數] - 1) 重新遞減。



## 13.8. 滑動開關

### 13.8.1. 概要

【滑動開關】物件是用來建立一個滑塊區域顯示數值或藉由拖曳滑軌改變指定暫存器內的數值。

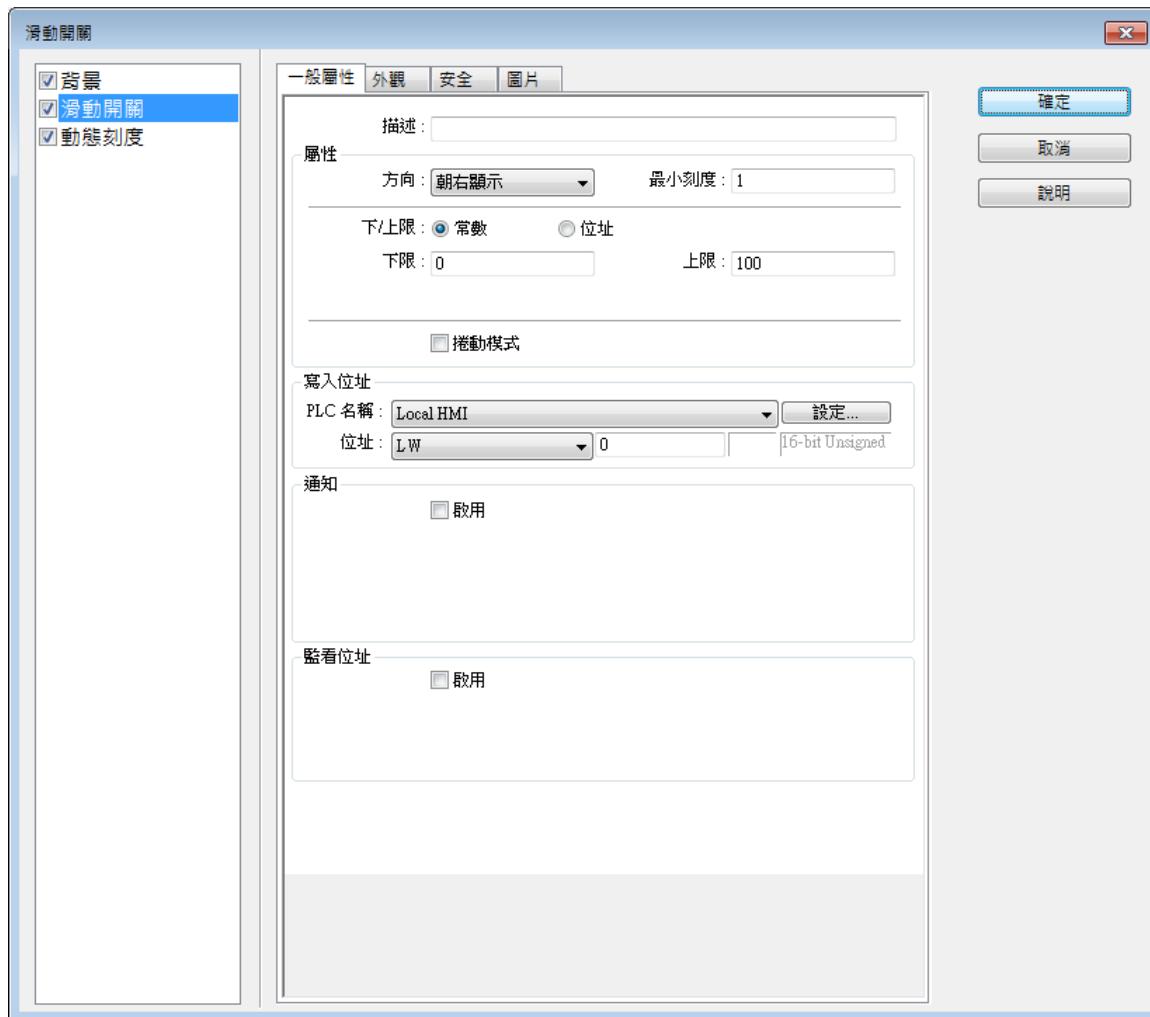
### 13.8.2. 設定



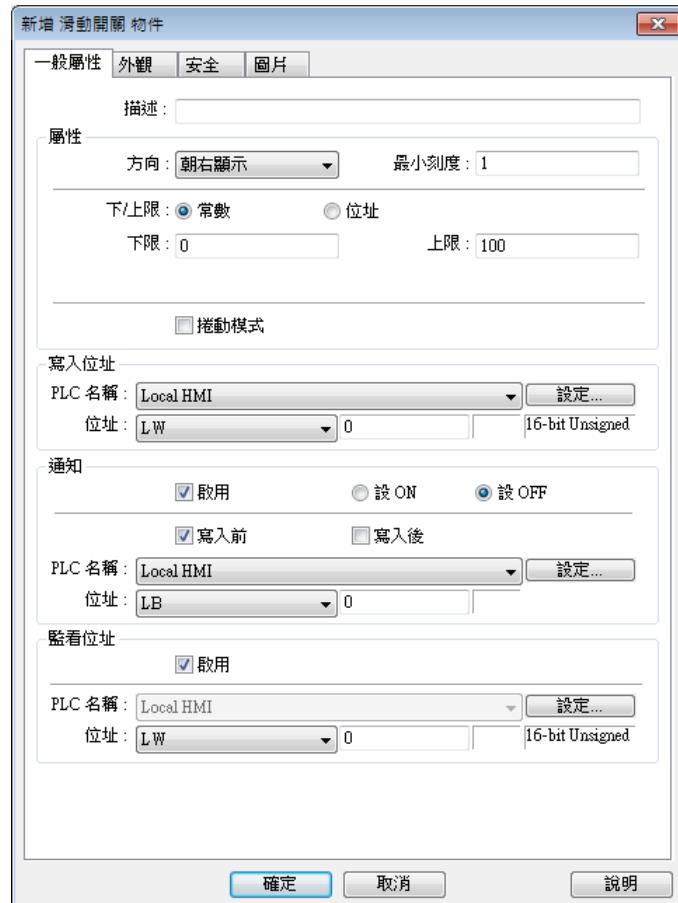
按下工作列的【物件】»【滑動開關】按鈕後即會開啟【滑動開關】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【滑動開關】物件。

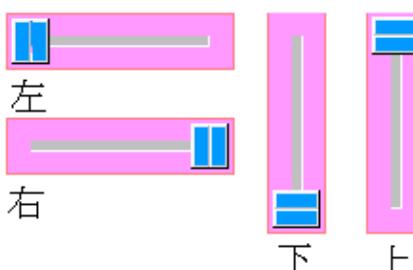
#### 一般屬性設定

cMT / cMT X 系列



## eMT、iE、XE、mTV 系列



設定	描述
屬性	<b>方向</b> 滑動開關物件可以四個方向來顯示 (朝右顯示，朝上顯示，朝左顯示，朝下顯示)。
	
	<b>最小刻度</b> 依照所填入之最小刻度值來顯示。例如：設 [最小刻度] 為 10，數值顯示為每一次都是依據 10 的刻度來跳動。
	<b>常數</b>

可直接設定字組暫存器的上下限常數值。例如：設【下限】為 5 和【上限】為 100，則設定的數值範圍為 5 ~ 100。

### 位址

上下限可由指定暫存器設定，請見以下範例 1。

### 捲動模式

不同於【最小刻度】拖拉滑動開關改變數值，只要輕觸一下【滑動開關】物件時，數值會被遞增/遞減，增減的多寡會根據【捲動值】的設定。

#### 寫入位址

點選【設定】後選擇字組暫存器設備類型的【設備】，【位址】，【設備類型】，【系統暫存器】，【索引暫存器】來控制滑動開關物件。使用者也可在【一般屬性】頁中設定位址。

#### 通知

使用此項設定，則在使用手動操作模式時，在完成動作後可以連帶設定此項目所指定暫存器的狀態，使用【開】與【關】選擇要設定的狀態。

點選【設定】後選擇位元暫存器設備類型的【設備】，【位址】，【設備類型】，【系統暫存器】，【索引暫存器】來控制通知位元項目。使用者也可在【一般屬性】頁中設定位址。

#### 【寫入前】/【寫入後】

在寫入動作前 / 後設定所指定暫存器的狀態。

#### 監看位址

在滑塊被拖曳時，可以即時顯示當前寫入位址的設定值。

## 範例 1

上下限可由指定暫存器設定。當寫入位址為 LW-n，則上/下限會根據以下的規則自動被設定為：

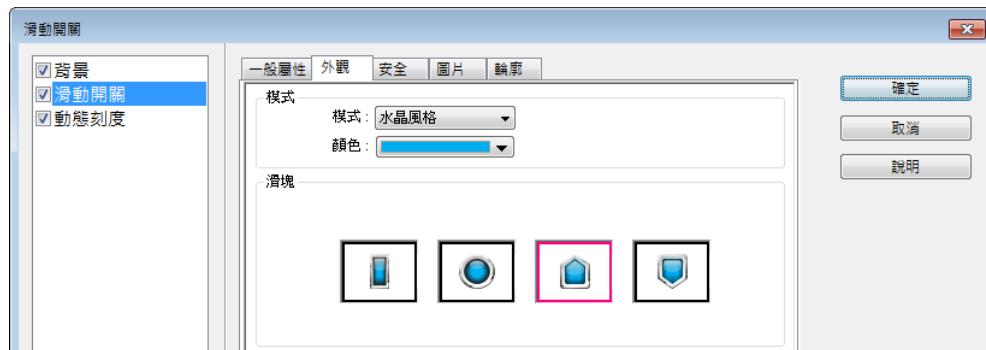
位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
位址	LW-n	LW-n	LW-n
下限	LW-n	LW-n	LW-n
上限	LW-n+1	LW-n+2	LW-n+4

以下表為例，當【暫存器位址】為 LW-100 時，則上/下限的位址會自動被設定為：

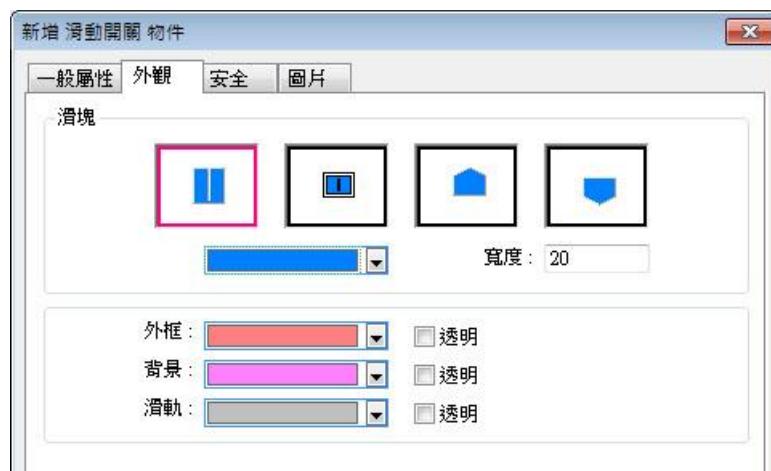
位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
位址	LW-100	LW-100	LW-100
下限	LW-100	LW-100	LW-100
上限	LW-101	LW-102	LW-104

## 外觀設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



### 設定

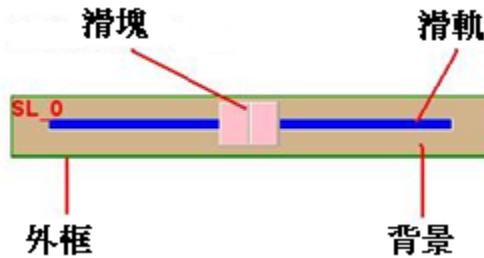
### 描述

**樣式 (只支援  
cMT / cMT X  
系列)**

選擇滑塊的樣式。若選用預設，可從圖片庫挑選一圖片作為滑塊的顯示圖示。

**滑塊**

系統共有四種預設滑塊外觀可供選擇，也可調整滑塊寬度、外框、背景和滑軌顏色。

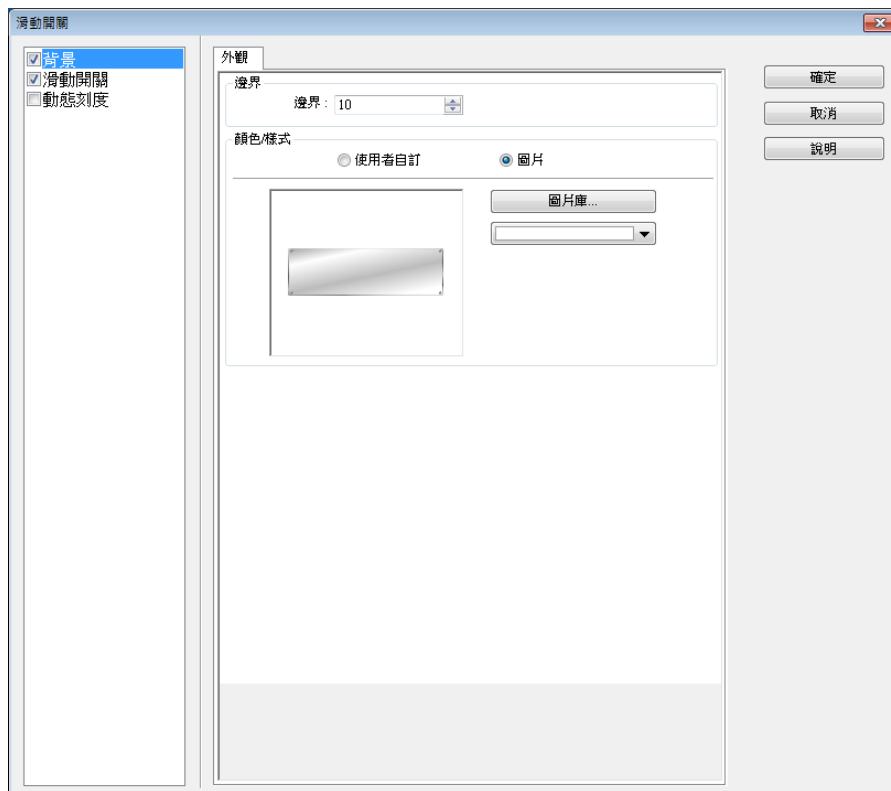


### 13.8.3. 複合物件

cMT / cMT X 系列提供一次性設定相關元件的功能。除了滑動開關以外，增加背景與動態刻度的

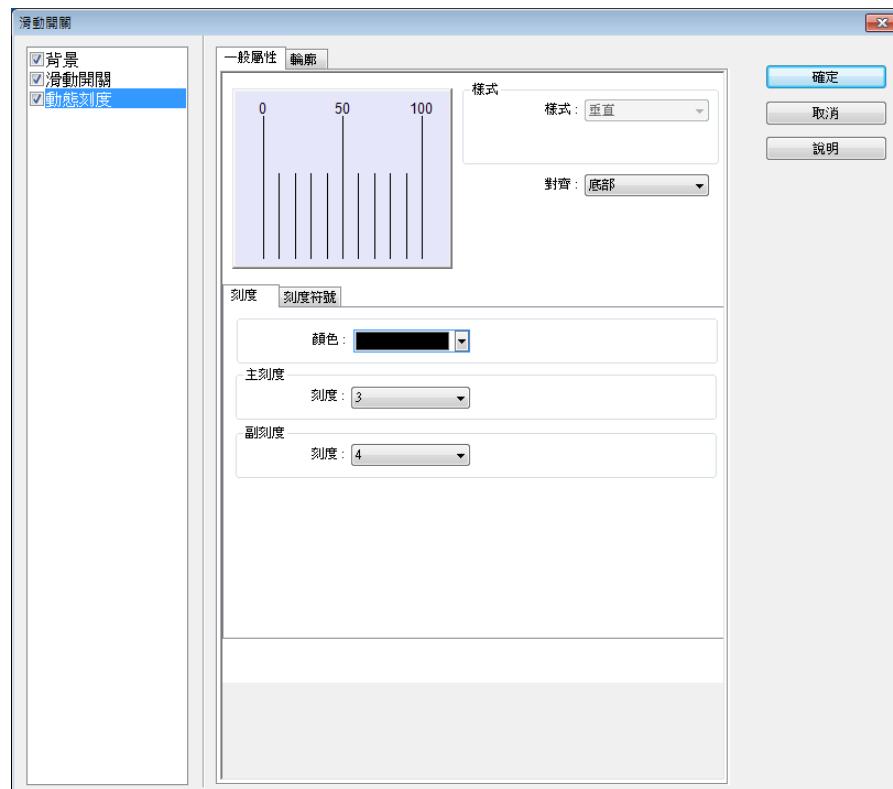
元素供使用者更能活用及美化滑動開關的設計。

## 背景設定



設定	描述
邊界	邊緣與物件的留白距離。
顏色/樣式	<p>使用者自訂</p> <p>根據圖案樣式與顏色挑選自訂背景。</p> <p>圖片</p> <p>可使用內建的背景圖片或是圖片庫中的檔案。</p>

## 動態刻度設定



設定	描述
樣式	將依動態刻度所設定的類型顯示。
對齊	設定刻度對齊的相對位置。
刻度	選擇 [主刻度] 與 [副刻度] 的分割數量及刻度顏色。
刻度符號	<p>設定刻度符號顯示時的字型、顏色、尺寸與其他屬性。</p> <p>在 [滑動開關] 若將上下限設定成 [位址]，則 [動態刻度] 的[上下限取自暫存器] 會被自動設定。在 [滑動開關] 的 [方向] 若為向左顯示，則 [動態刻度] 的 [轉換(左對齊/右對齊)] 會被自動設定。</p>

## 13.9. 數值

### 13.9.1. 概要

[數值] 物件可以用來輸入或顯示所指定字組暫存器內的數值。

### 13.9.2. 設定



按下工作列的 [物件] » [數值] 按鈕後即會開啟 [數值] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [數值] 物件。

#### 一般屬性設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



設定	描述
啟用輸入功能	若勾選，可開啟與輸入功能相關的屬性設定。
讀取/寫入使用不同位址	使用者可以分開設定數據的讀取位址與寫入位址。
讀取位址	<p>點選 [設定] 後選擇字組暫存器設備類型的 [設備], [位址], [設備類型], [系統暫存器], [索引暫存器] 來顯示數值。使用者亦可選用 [位址標籤庫] 裡設定好的位址標籤。</p> <p>當未勾選 [讀取/寫入使用不同的位址] 時，讀取位址欄位名稱會顯示為 [讀取/寫入位址]。</p>
寫入位址	選擇字組相關的 [設備], [設備類型], [位址] 作為數值寫入目標。
通知	<p>使用此項設定，則在完成動作之前 / 之後可以連帶設定此項目所指定暫存器的狀態，使用 [開] 與 [關] 選擇要設定的狀態。</p> <p>寫入前的通知功能在 cMT、cMT X 系列上與 eMT、iE、XE、mTV 系列上略有不同。</p> <p><b>cMT、cMT X 系列</b></p> <p>觸發物件開啟鍵盤視窗 » 寫入前的通知 » 輸入數值，按下 ENTER » 正式寫入 » 寫入後的通知。</p> <p><b>eMT、iE、XE、mTV 系列</b></p> <p>觸發物件開啟鍵盤視窗 » 輸入數值，按下 ENTER» 寫入前的通知 » 正式寫入 » 寫入後的通知。</p>
輸入無效通知	當輸入無效的數值時，通知指定暫存器的狀態 [開 / 關]。

## 數值輸入設定

cMT、cMT X 系列



eMT、iE、XE、mTV 系列



### 設定

#### 模式

### 描述

#### 觸控

藉由觸碰物件來啟動輸入程序。

#### 位元控制

藉由指定位元暫存器的開或關來啟動及結束輸入程序。

#### 輸入控制

指定控制輸入啟動及結束的位元暫存器位址。輸入順序必須遵照 [輸入次序] 的設定，輸入時須搭配外接 USB 鍵盤，不可使用螢幕觸控鍵盤。

cMT-SVR 則需使用 cMT Viewer 的內建鍵盤。

#### 輸入次序

設定輸入次序及輸入次序群組達成多個輸入物件連續輸入。

#### 使用準則：

- 輸入次序範圍: 1 ~ 511。群組範圍: 1 ~ 15。
- 若無勾選 [群組] 時，輸入次序群組為 0。

- 系統只尋找同一個輸入次序群組中的輸入物件。
- 愈小的輸入次序數值代表輸入順序排在愈前面，反之則愈後面。
- 若多個輸入物件有同樣的輸入次序群組及輸入次序，則較下層的輸入物件將優先輸入。



請點選此圖示下載範例程式。下載範例程式前，  
請先確定已連上網路線。

## 鍵盤

### 使用彈出鍵盤

勾選：指定鍵盤視窗及彈出位置。當啟動輸入時，系統將在指定位置彈出鍵盤視窗，並於結束輸入時關閉。

不勾選：啟動輸入時系統將不會彈出鍵盤視窗，使用者必須以下列方法進行輸入動作：

- 自行在視窗中設計鍵盤。
- 使用外接鍵盤。

### 系統鍵盤

當使用 cMT/cMT X 系列時，可以使用 cMT/cMT X 預設的鍵盤，不需自行設計鍵盤視窗。

### 動畫設定

當使用 cMT/cMT X 系列且使用 [自訂鍵盤] 時，可以設定鍵盤視窗彈出時的動畫效果。

動畫效果請參考《Ch13.5 功能鍵-視窗切換》。

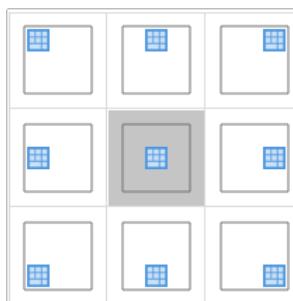
### 隱藏視窗控制條

數值 / 字元鍵盤可選擇不使用視窗控制條。

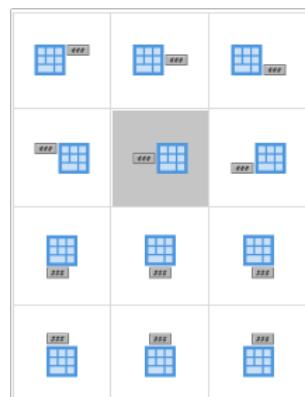
### 鍵盤彈出位置

當使用 cMT/cMT X 系列且使用 [自訂鍵盤] 時，鍵盤的彈出位置可選擇相對於 HMI 螢幕或相對於物件。

相對於 HMI 螢幕



相對於物件



**若輸入數值超出範圍時重新啟動鍵盤**

當使用輸入物件時，若輸入的數值超出設定範圍時系統將自動重新啟動鍵盤。

**其他選項**

(適用於 cMT / cMT X  
系列)

**顯示上下限值**

勾選此選項，在輸入數值時，物件旁會顯示該物件位址的上下限值。

**顯示前一數值**

勾選此選項，在輸入數值時，物件旁會顯示該物件位址更改前的數值。



若欲於當前視窗內嵌鍵盤，請參考《第十二章:鍵盤設計與應用》。



請點選此圖示觀看教學影片，請先確定已連上網路線。

## 範例 1

設計群組式的數值物件。

藉由設定輸入次序及輸入次序群組達成多個輸入物件連續輸入。當完成目前物件輸入動作後，系統將自動跳至下一個同一群組的輸入物件繼續輸入。

- 建立三個數值物件，皆使用 [輸入次序]，次序分別為次序 1、次序 2、及次序 3，並設定為 [群組 1]。則輸入順序如下圖：

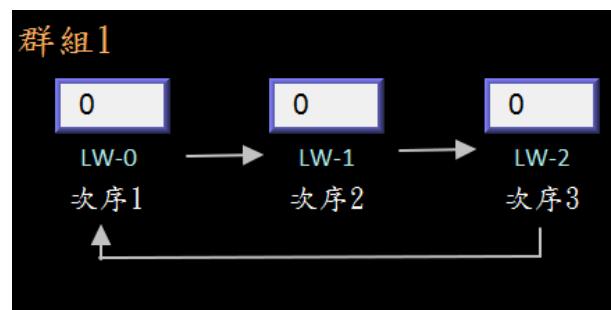
LW-0



LW-1



LW-2



- 若在輸入完最後一個數值物件後，即停止跳出鍵盤結束所有的輸入動作，則勾選 [輸入完成後不再按次序輸入] 即可。



## 格式設定



設定	描述
資料格式	支援二進碼十進數 (BCD)、二進制 (Binary)、十進制帶號/未帶號整數 (Signed/Unsigned)、十六進制 (Hex)、浮點數 (Float) 的讀取與寫入。使用 16-bit 格式時，會佔用暫存器一個字組，使用 32-bit 格式時，則是佔用兩個字組。
顯示格式	<p><b>密碼</b> 數值顯示時將使用 “*” 符號代替所有數字。</p> <p><b>預設</b> 一般的數值格式。</p> <p><b>自訂</b> [格式] 中的 每個 “*” 符號代表顯示於數值物件的每一位整數，“#” 符號代表顯示於數值物件的小數，其中僅 cMT / cMT X</p>

系列支援 float 或 double 自訂格式。除了數值以外，[格式] 內也可以輸入額外的文字，例如 “kg” 等文字。字型對齊屬性可以選擇 [左對齊]、[置中對齊]、[右對齊] 與 [前導零]。

數值物件所顯示的位數，會依 “\*” 符號的數目，由高位至低位開始顯示。

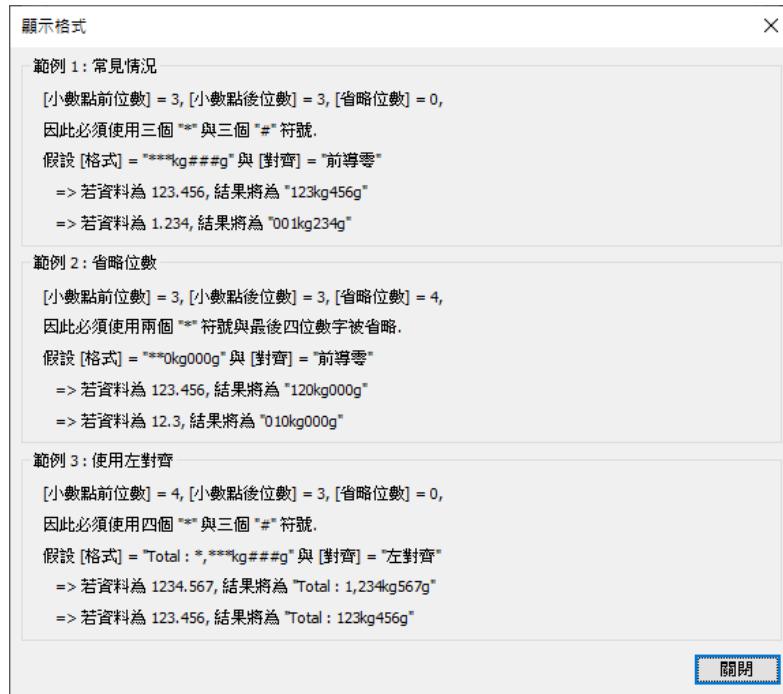
### 省略位數

可決定數值顯示的省略位數，由低位至高位開始省略。

所需 “\*” 符號的數量 = [小數點前位數] - [省略位數] - [小數點後位數]，所需的 “#” 符號需要的數量 = [小數點後位數] - [省略位數]

請見以下範例：





當使用非 [前導零] 時，將會忽略未顯示的 \* 中間的文字。

例如：

[小數點前位數] = 5, [省略位數] = 0

顯示格式為 "Total=\*\*kg\*\*\*g"

若讀取資料為 255 時，將會顯示 "Total=255g"

若讀取資料為 1000 時，將會顯示 "Total=1kg000g"

#### 科學符號

使用科學符號表示數值，例如：數值 100 以 1e+2 表示。BCD、HEX、Binary 不支援此格式。僅 cMT / cMT X 系列支援科學符號。

### 數字位元數

#### 小數點前位數

小數點前的顯示位數。

#### 小數點後位數

小數點後的顯示位數。

### 比例轉換

#### 內插法

所顯示的數據是利用暫存器中的原始數據經過換算後所獲得。

選擇此項功能必須設定 [比例最小值]、[比例最大值] 與 [限制] 項目中的 [輸入下限]、[輸入上限]。請見以下範例 2。

測試：

測試設定的轉換比例是否正確。請見以下範例 2。

動態比例：

比例轉換的上下限可由指定暫存器設定。請見以下範例 4。

#### 巨集副函式

物件的讀取/寫入數據將經過巨集函數庫的函式運算而獲得。

讀取轉換：物件讀取的數值將經過巨集運算後顯示。

寫入轉換：寫入物件的數值將經過巨集運算後回傳。

使用此功能，請見《13.9.2.1 數值物件使用巨集副函式規則》。

**限制** 用來設定輸入數值上下限的來源，並可設定警示顏色與警示效果。

#### 輸入常數

選擇輸入數值的上下限分別來自【設備下限】與【設備上限】中的設定值。若輸入值不在上下限定義的範圍內，將無法更改暫存器內的數值。

#### 取自暫存器

上下限可由指定暫存器設定。

使用連續位址，請見以下範例 5。

cMT / cMT X 系列支援使用不同位址，使用者可以分別設定上限及下限暫存器位址。當比例轉換方式為內插法時，上下限位址均必需要設定。

**使用警示** 下限

**色彩** 當暫存器內的數值小於下限值時，物件會使用此項顏色顯示數值。

#### 上限

當暫存器內的數值大於上限值時，物件會使用此項顏色顯示數值。

#### 閃爍

當暫存器內的數值小於下限值或大於上限值時，物件會使用閃爍的效果加以警示。

### 觸發動作設定 (僅 cMT / cMT X 支援此物件)

使用此項設定，則在完成動作之前 / 之後可以分批執行一系列的動作。設定動作時，可以先將動作進行分組，同一個群組中的動作會一起執行。當前一個群組中的所有動作都觸發後，下一個群組的動作才會開始進行。相較於【一般屬性】設定的通知，【觸發動作設定】分頁的設置可使用較多樣且較靈活的設計方式。動作群組的注意事項請參考手冊 Ch13.53 說明。



設定	描述
寫入前動作	一旦觸發寫入動作 (即使尚未寫入完成)，就會分批執行一系列的動作。
寫入後動作	寫入完成後會分批執行一系列的動作。

### 13.9.2.1. 數值物件使用巨集副函式規則

- 必須有回傳值且只能有一個參數。

例如：

sub char test (short a) // (正確)

sub test (char a) // (錯誤，沒有回傳值)

sub char test (char a, char b) // (錯誤，有兩個參數)

- 數值物件的資料格式須對應到特定的參數類型。

如下表所示：

巨集參數類型	數值物件的資料格式
short	16-bit Signed
Int	32-bit Signed
unsigned short	16-bit BCD, 16-bit HEX, 16-bit Binary, 16-bit Unsigned
unsigned int	32-bit BCD, 32-bit HEX, 32-bit Binary, 32-bit Unsigned
float	32-bit Float
long	64-bit Signed (只支援於 cMT / cMT X)
unsigned long	64-bit Unsigned (只支援於 cMT / cMT X)
double	64-bit Double (只支援於 cMT / cMT X)

假設一數值物件的資料格式為 16-bit Unsigned 時，只能選擇參數類型為 unsigned short 的巨集副函式，例如：

```
sub char test(unsigned short a) // (正確)
sub char test(char a) // (錯誤)
```

- 僅可存取本地 HMI 的位址。

例如：

```
GetData(var, "Local HMI", LB, 0, 1) // (正確)
GetData(var, "MODBUS RTU", 0x, 0, 1) // (錯誤)
```

- 無法呼叫下列函數：

ASYNC\_TRIG\_MACRO, SYNC\_TRIG\_MACRO, DELAY, FindDataSamplingDate, FindDataSamplingIndex, FindEventLogDate, FindEventLogIndex, IMPORT, IMPORT2, OUTPORT, PURGE, TRACE

- 無法使用下列迴圈語句：

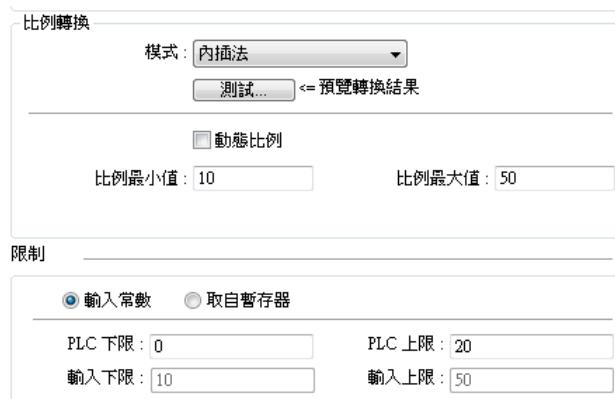
For-Next, While-Wend

## 範例 2

比例轉換使用 [內插法] 公式如下：

$$\text{新數值} = \frac{\text{比例最小值} + (\text{原數值} - PLC\text{下限}) \times \text{比例最大值} - \text{比例最小值}}{PLC\text{上限} - PLC\text{下限}}$$

以下圖的設定為例，當原始數據是 15 時，則經過換算得到的數值為 40。



### [測試]

利用測試功能檢查設定的轉換比例是否設定正確。輸入欲測試數值至 [PLC] 欄位，以下圖所示，輸入 15 為例，經過比例轉換後可得計算結果 40。



### 範例 3

當數值物件使用 Float 以外的變數型態且使用小數點，則內插法運算後的結果不會自動調整小數點的位數。請參考下述的說明：

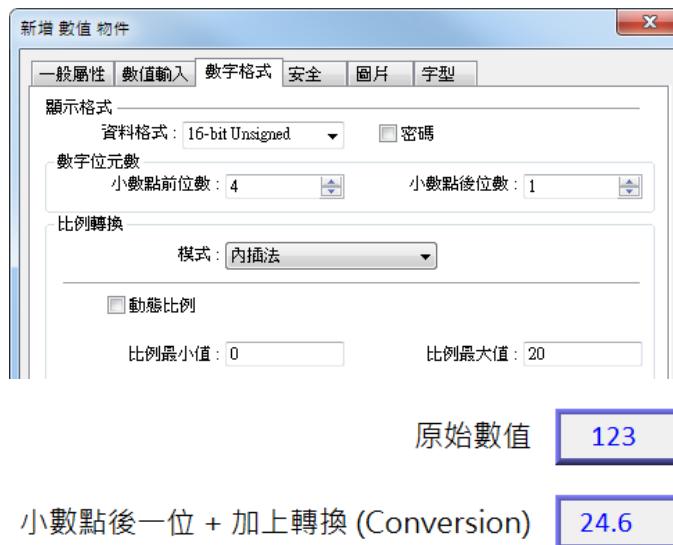
- 建立兩個數值物件，其中一數值物件設定顯示小數點後 1 位數，且使用內插法，如下圖所示。



- 輸入一數值 “123” 至未使用內插法的物件，則另一物件顯示數值為 “246.0” 而非 “24.6”。



3. 若欲使數值向左移一位數，則需調整內插法的比例最大值，如下圖所示。



#### 範例 4

比例轉換使用 [內插法] 時，比例最小值/比例最大值可由指定暫存器設定。當寫入位址為 LW-n，則比例最小值/比例最大值會根據以下的規則自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
動態比例位址	LW-n	LW-n	LW-n
比例最小值	LW-n	LW-n	LW-n
比例最大值	LW-n+1	LW-n+2	LW-n+4

以下表為例，當 [動態限制位址] 為 LW-100 時，則比例最小值/比例最大值位址會自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
動態比例位址	LW-100	LW-100	LW-100
比例最小值	LW-100	LW-100	LW-100
比例最大值	LW-101	LW-102	LW-104

#### 範例 5

上下限可由指定暫存器設定。當寫入位址為 LW-n，並使用連續位址，則上/下限會根據以下的規則自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
暫存器位址	LW-n	LW-n	LW-n
下限	LW-n	LW-n	LW-n
上限	LW-n+1	LW-n+2	LW-n+4

以下表為例，當 [暫存器位址] 為 LW-100 時，則上/下限的位址會自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
暫存器位址	LW-100	LW-100	LW-100

下限	LW-100	LW-100	LW-100
上限	LW-101	LW-102	LW-104

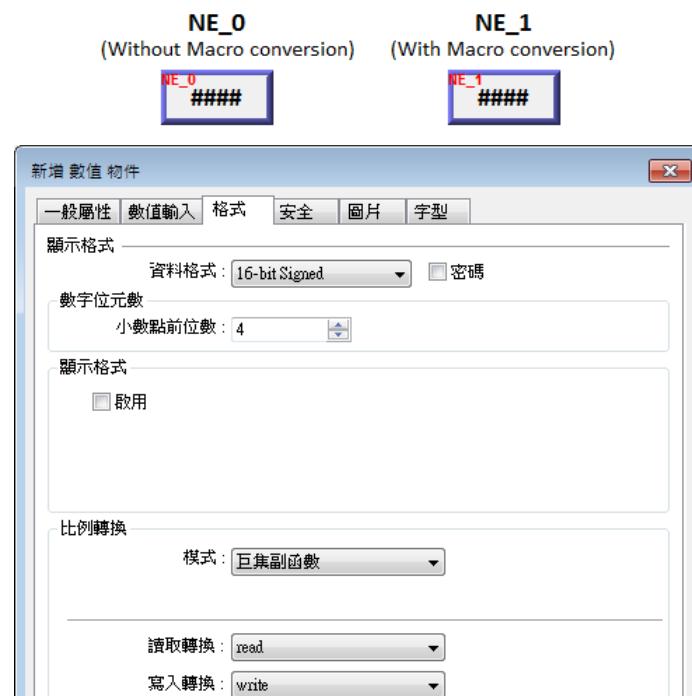
## 範例 6

此範例說明 [數值] 物件如何使用 [巨集副函式] 執行比例轉換。

假設有兩個函式庫分別如下：一個執行 [讀取轉換]，另一個執行 [寫入轉換]。

Read Conversion	Write Conversion
<pre>sub short read(short a) short b b = a + 10 return b end sub</pre>	<pre>sub short write(short a) short b b = a - 10 return b end sub</pre>

- 建立兩個控制位址相同的 [數值] 物件 NE\_0 及 NE\_1，令 NE\_1 使用 [巨集副函式] 執行讀取/寫入比例轉換。



- 當在 NE\_0 寫入 0 時，NE\_1 會執行 [讀取轉換] 換算為 10。

NE_0 (Without Macro conversion)	NE_1 (With Macro conversion)
0	10

Read Conversion	Write Conversion
sub short read(short a)  short b b = a + 10 return b  end sub	sub short write(short a)  short b b = a - 10 return b  end sub

3. 當在 NE\_1 輸入 80 時，會執行 [寫入轉換] 換算後為 70，故 NE\_0 顯示為 70。



Read Conversion	Write Conversion
sub short read(short a)  short b b = a + 10 return b  end sub	sub short write(short a)  short b b = a - 10 return b  end sub

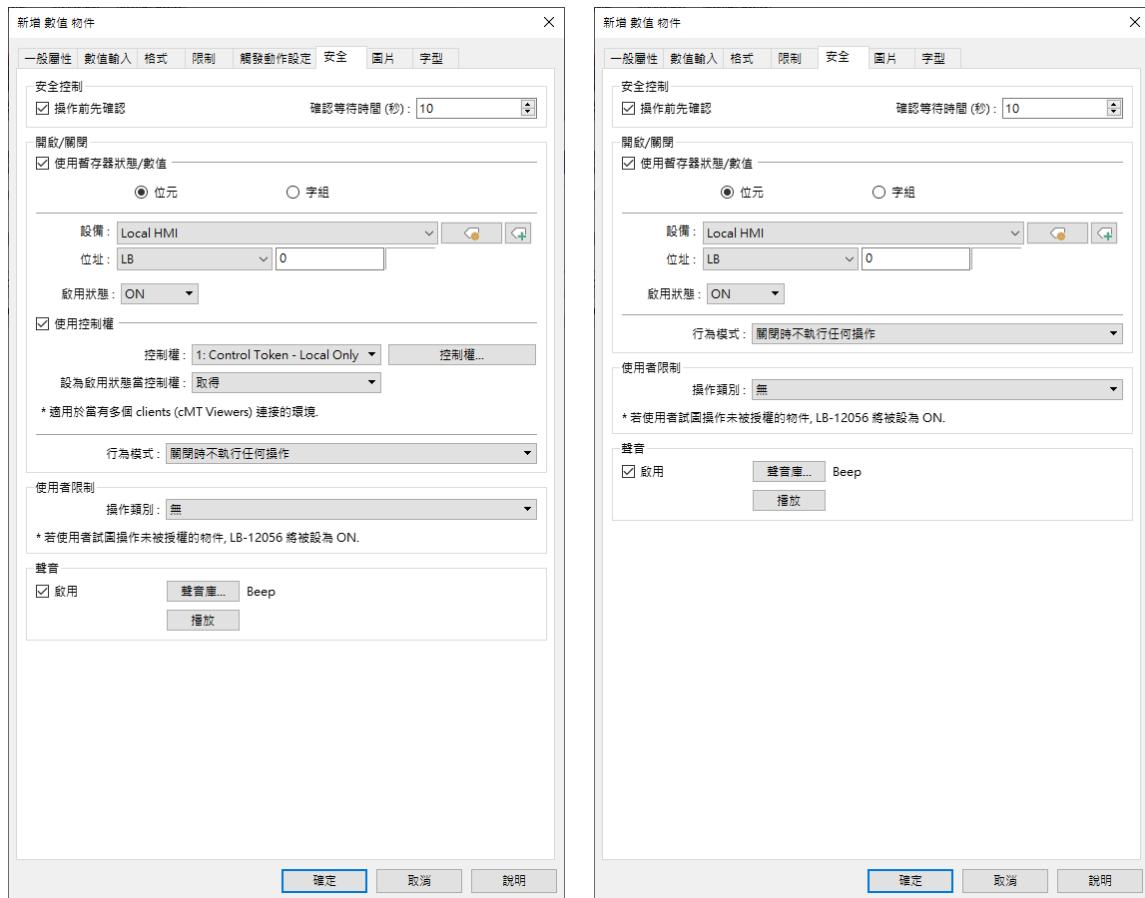
### Note

- 當一數值物件同時使用 [讀取轉換] 及 [寫入轉換] 時，若在此物件輸入一數值，則物件顯示的數值會先執行 [寫入轉換]，再根據轉換算後的數值執行 [讀取轉換]。若在此範例中，[寫入轉換] 設定為  $b = a - 20$ ，則於 NE\_1 寫入 80 後，會先執行 [寫入轉換] 回傳數值 60，再執行 [讀取轉換] 顯示數值 70。

## 安全設定

cMT、cMT X 系列

eMT、iE、XE、mTV 系列

**設定****描述****安全控制**

執行操作前會提示確認視窗。

**開啟/關閉**

若勾選 [使用暫存器狀態/數值]，此物件是否允許被操作，將決定於一個指定位元或字組位址的 [狀態] 是否被啟用。

若勾選 [使用控制權]，此物件能否被操作需依照客戶端是否取得控制權而定。

以下為行為模式：

**關閉時不執行任何操作**

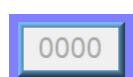
當指定的位元或字組位址的數值不符合 [狀態] 時，物件將無法被操作但不會有任何變化。

**關閉時隱藏**

當指定的字組位址的數值不符合 [狀態] 時，物件會被隱藏。

**關閉時使用灰階效果顯示文字**

物件的數值會在指定的字組位址的數值不符合 [狀態] 時，以灰階樣式顯示。

**使用者限制**

根據使用者密碼權限來限制對數值物件的操作。

**聲音**

啟用時可選擇觸發數值物件時的聲音。

## 字型設定



設定	描述
顏色	當數值在上下限的範圍內時，使用此項顏色顯示。
對齊	<p><b>左對齊</b>：數值靠左顯示。</p> <p><b>置中對齊</b>：數值置中顯示。</p> <p><b>右對齊</b>：數值靠右顯示。</p> <p><b>前導零</b>：數值不滿設定的位數前會補零。</p>
	<p><b>左對齊</b> <span style="border: 1px solid blue; padding: 2px;">66</span></p> <p><b>置中對齊</b> <span style="border: 1px solid blue; padding: 2px;">66</span></p> <p><b>右對齊</b> <span style="border: 1px solid blue; padding: 2px;">66</span></p> <p><b>前導零</b> <span style="border: 1px solid blue; padding: 2px;">0066</span></p>

---

**尺寸**

設定字型大小。

---

## 13.10. 字元

### 13.10.1. 概要

[字元] 物件使用 ASCII 編碼的方式顯示所指定暫存器中的數據。

### 13.10.2. 設定



按下工作列的 [物件] » [字元] 按鈕後即會開啟 [字元] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [字元] 物件。

#### 一般屬性設定

cMT、cMT X 系列



eMT、iE、XE、mTV 系列

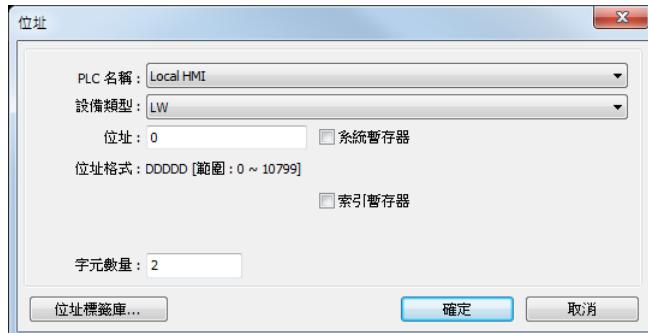


設定	描述
啟用輸入功能	若勾選，可開啟與輸入功能相關的屬性設定。
密碼	字元顯示時將使用“*”符號代替所有字元。
多行顯示	顯示多行文字。當字串中有使用到換行的 ASCII 碼 LF (0xA) 時，顯示時即會換行。
垂直對齊	當啟用【多行顯示】時，可選擇多行文字的對齊方式。
編碼	編碼可選擇 UTF-8 (預設)、Unicode 或 Latin-1。
字串終止	可選擇忽略空白字元，或是當遇到第一個 \0 時，視為字串結束。
高低位元組轉換	正常情況下，ASCII code 的顯示順序為 [高位元組] + [低位元組]。勾選此功能後，則顯示順序改為 [低位元組] + [高位元組]。



高低位元不互換      高低位元互換

讀取位址	點選【設定】後，可以選擇字組暫存器設備類型的【PLC 名稱】，【位址】，【設備類型】，【系統暫存器】，【索引暫存器】來顯示字元。使用者亦可選用【位址標籤庫】裡設定好的位址標籤，也可在【一般屬性】頁中直接設定 PLC 名稱、暫存器名稱及位址。
------	--------------------------------------------------------------------------------------------------------------------------



#### 字組數量

選擇文字最多可顯示的資料長度，單位為 word。

cMT / cMT X 支援最多 160 words，eMT/iE/XE/mTV/iP 則為 64 words。

通知	使用此項設定，則在完成動作之前 / 之後可以連帶設定此項目所指定暫存器的狀態，使用【開】與【關】選擇要設定的狀態。 寫入前的通知功能在 cMT、cMT X 系列上與 eMT、iE、XE、mTV 系列上略有不同。
----	--------------------------------------------------------------------------------------------------------------

#### cMT、cMT X 系列

觸發物件開啟鍵盤視窗 » 寫入前的通知 » 輸入數值，按下 ENTER » 正式寫入 » 寫入後的通知。

#### eMT、iE、XE、mTV 系列

觸發物件開啟鍵盤視窗 » 輸入數值，按下 ENTER » 寫入前的通知 » 正式寫入 » 寫入後的通知。

 Note

- 使用 UNICODE 時，一個 UNICODE 文字等於一個字組 (word)；而使用 ASCII 時，一個 ASCII 文字等於一個位元組 (byte)，所以一個字組 (word) 可以有兩個 ASCII 文字。(1 個字組 (word) 等於 2 個位元組 (byte))

## 字型設定

cMT、cMT X 系列



eMT、iE、XE、mTV 系列



## 設定

## 描述

## 預覽實際的文字設定尺寸

根據屬性中設定的字型與尺寸，在內容欄位中預覽。

## 鎖定內容的最小尺寸

勾選後，ASCII 物件寬度自動匹配所有文字能顯示的最小寬度。

## 屬性

可設定文字顯示時所使用的字型、字型大小與顏色，另外也包括文字對齊的方式。

## 文字對齊

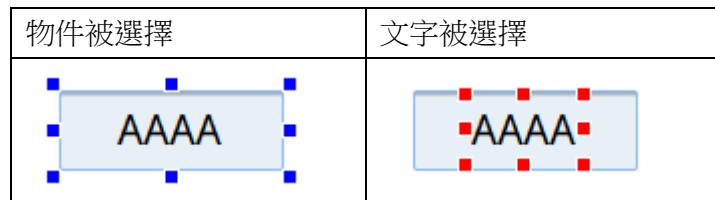
## 水平

當字元物件使用多行文字時，文字的水平對齊方式。

## 文字區域位置

文字在物件中的顯示位置。當物件使用了文字，且文字被選擇時，會使用紅框表示；當物件被選擇時，會使用藍框表示。若啟用了【多行】

顯示]，則此功能無效。



### 觸發動作設定 (僅 cMT / cMT X 支援此物件)

使用此項設定，則在完成動作之前 / 之後可以分批執行一系列的動作。設定動作時，可以先將動作進行分組，同一個群組中的動作會一起執行。當前一個群組中的所有動作都觸發後，下一個群組的動作才會開始進行。相較於 [一般屬性] 設定的通知，[觸發動作設定] 分頁的設置可使用較多樣且較靈活的設計方式。動作群組的注意事項請參考手冊 Ch13.53 說明。



設定	描述
寫入前動作	一旦觸發寫入動作 (即使尚未寫入完成)，就會分批執行一系列的動作。
寫入後動作	寫入完成後會分批執行一系列的動作。

## 13.11. 間接視窗

### 13.11.1. 概要

**[間接視窗]** 物件為使用字組暫存器控制指定編號的視窗的開啟及關閉。彈出視窗的顯示範圍有兩種方式，第一種是先在視窗上定義一個顯示區域，在此顯示區域內顯示彈出視窗的內容。所顯示的彈出視窗的寬度與高度不會大於此顯示區域；第二種是使用**【自動調整視窗尺寸】**功能，啟用此功能後不需事先定義彈出視窗的區域，系統會自動根據對應的彈出視窗尺寸調整其顯示區域。欲關閉彈出視窗只需將控制的字組暫存器的內容設定為 0 即可。**[直接視窗]** 與 **[間接視窗]** 的差別在於直接視窗是利用位元狀態進行視窗控制，而間接視窗則是利用字組數值進行視窗控制。

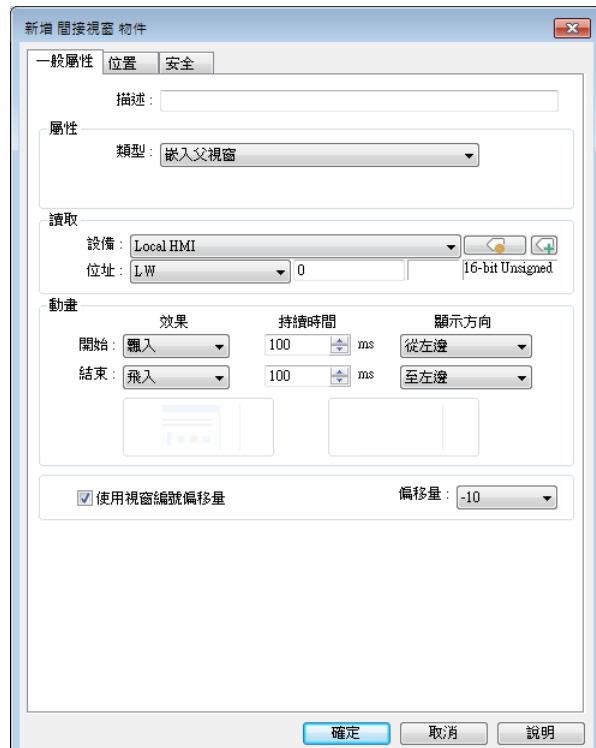
### 13.11.2. 設定



按下工作列的**【物件】** » **【嵌入視窗】** » **【間接視窗】** 按鈕後即會開啟**【間接視窗】** 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個**【間接視窗】** 物件。

#### 一般屬性設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



設定	描述
類型	僅支援 cMT / cMT X 系列。此功能設置會影響當基本視窗有使用重疊

視窗時，重疊視窗上的間接視窗物件的顯示方式。請參考本節的範例 2。

### 嵌入父視窗

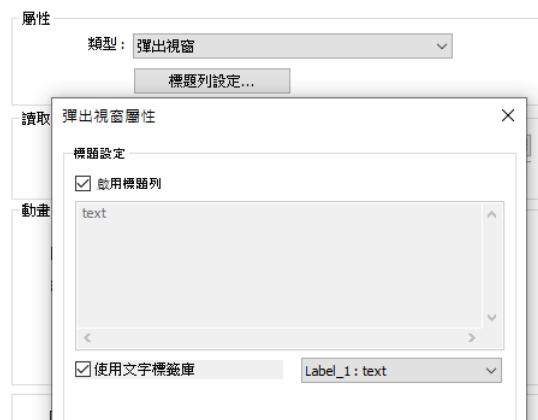
間接視窗物件顯示在該父視窗上。若間接視窗物件是放在重疊視窗上，在基本視窗上呼叫此重疊視窗的間接視窗物件時，物件是顯示在該重疊視窗上。

### 彈出視窗

間接視窗物件顯示在基本視窗上。若間接視窗物件是放在重疊視窗上，在基本視窗上呼叫此重疊視窗的間接視窗物件時，物件是顯示在基本視窗上。

### 標題列設定

選擇是否啟用標題列並設定標題名稱。



#### 讀取位址

點選 [設定] 後選擇字組暫存器設備類型的 [PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制視窗彈出。使用者也可在 [一般屬性] 頁中設定位址。

#### 屬性

#### 類型

設定彈出視窗樣式。支援兩種樣式：

- 隱藏視窗控制條  
彈出的子視窗不包含視窗控制條，無法拖曳移動視窗。



- 顯示視窗控制條  
彈出的子視窗包含視窗控制條，它的視窗位置可藉由控制條任意被拖曳。



**動畫  
(cMT / cMT X  
系列)**

**效果**  
可設定視窗 [開始] 與 [結束] 的動畫效果。

效果	樣式
淡出	
飛入	
飄入	
擦去	
分割	
環狀	
時鐘	
縮放	
旋轉	

**持續時間**

可以設定視窗開始與結束的速度。

**顯示方向**

可以設定視窗開始與結束的出入方向。

**使用視窗編號**

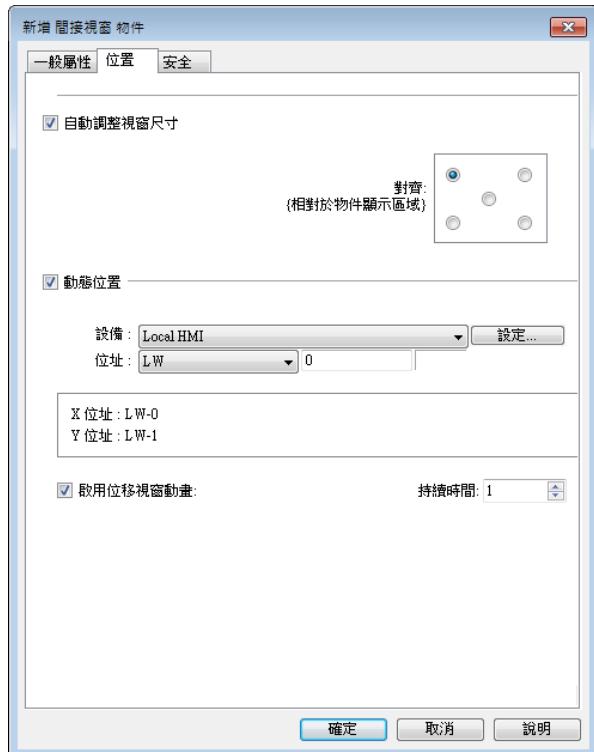
彈出的視窗的編號會等於暫存器中的數據加上偏移量。例如：控制字

**偏移量**

組暫存器中的數據為 20，偏移量為 5，則會彈出視窗編號 25。

## 位置設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



設定	描述
<b>自動調整視窗尺寸</b>	系統會根據彈出的視窗的尺寸調整其顯示範圍，並自動調整相對位置。 <b>(相對於物件顯示區域)</b> 彈出的視窗的位置基準點 (對應於間接視窗物件)。如：若設定右下角為基準點，則任一視窗彈出時，會以自身的右下角對準此基準點。若設定左上角為基準點，則任一視窗彈出時，會以自身的左上角對準此基準點，依此類推。請見以下範例 1。
<b>動態位置</b>	啟用此功能後，可在 HMI 上透過指定位址調整視窗的位置。 <b>啟用位移視窗動畫</b> 設定當視窗位置移動時，是否使用動畫。

## 範例 1

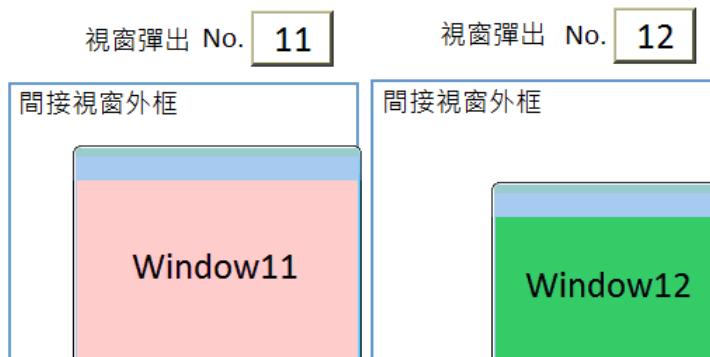
若有兩個彈出視窗編號 11 及編號 12，使用字組暫存器位址 LW-0 控制，勾選 [自動調整視窗尺寸] 功能並設定右下角為彈出基準點。

1. 建立一個 [間接視窗] 物件，讀取位址為 LW-0，勾選 [自動調整視窗尺寸]。
2. 在 [間接視窗] 物件上調整好彈出視窗的顯示區塊。





3. 將數值 11 寫入 LW-0，則彈出視窗編號 11。
4. 將數值 12 寫入 LW-0，則彈出視窗編號 12。
5. 將數值 0 寫入 LW-0，則關閉彈出視窗。

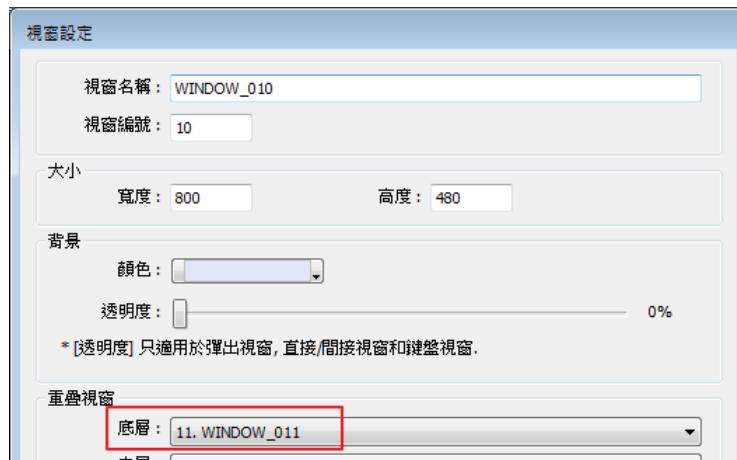


要關閉彈出視窗除了可以對控制字組暫存器寫入數值 0 之外，也可在彈出視窗上設計一個 [功能鍵] 物件，選擇 [關閉視窗] 模式，在按下此物件後即可關閉彈出視窗。

## 範例 2

此範例主要說明若間接視窗物件是放在重疊視窗上，在基本視窗上呼叫此重疊視窗的間接視窗物件時，物件類型選擇嵌入父視窗與彈出視窗時的顯示方式。

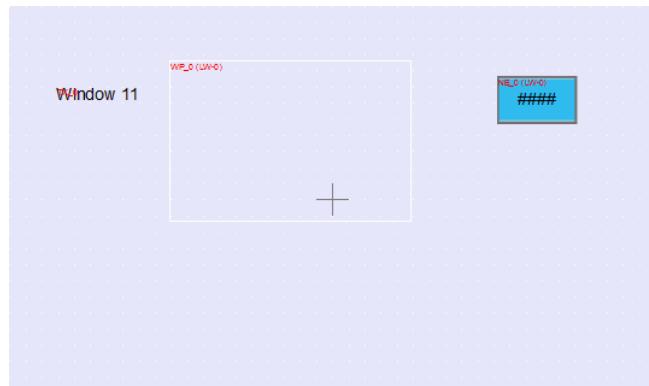
1. 建立兩個基本視窗編號 10 及編號 11，在視窗 10 啟用重疊視窗，並選擇視窗 11。



2. 在視窗 10 任意建立數個物件。

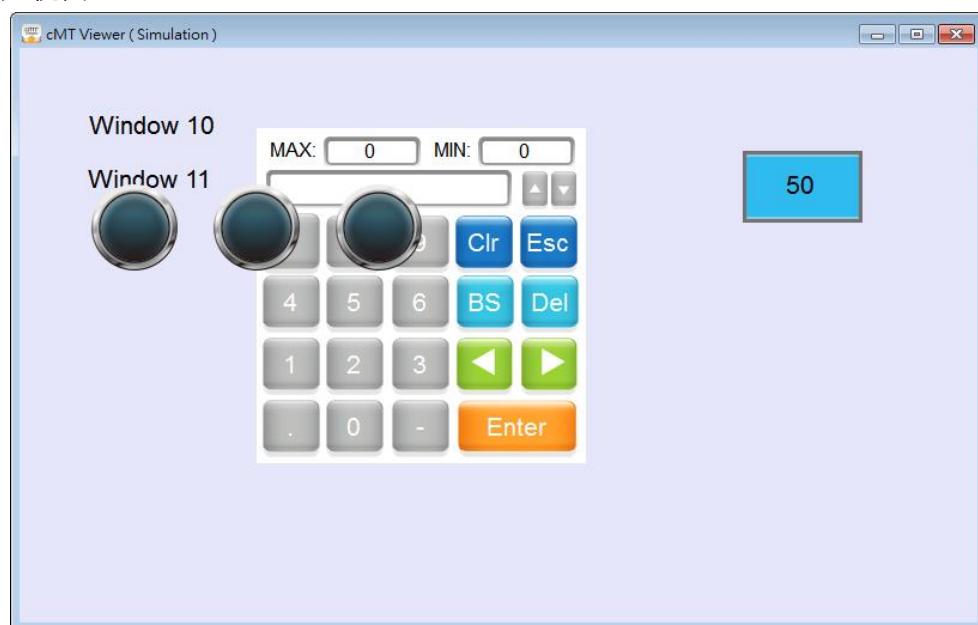


3. 在視窗 11 建立一個 [間接視窗] 物件與一個 [數值輸入] 物件，讀取位址皆為 LW-0。



4. 在 [數值輸入] 物件輸入彈出視窗的編號，可看到當類型選擇為 [嵌入父視窗] 與 [彈出視窗] 使用以下方式呈現。

#### 嵌入父視窗



### 彈出視窗



#### Note

- eMT、iE、XE、mTV 系列最多可同時開啟 24 個彈出視窗，包含系統訊息視窗、直接視窗和間接視窗；cMT / cMT X 系列則無彈出視窗的數量上限。
- 系統不允許在一個基本視窗上使用 2 個直接 (或間接) 視窗彈出同一個視窗。
- 如果彈出的視窗有 [壟斷] 屬性，當視窗彈出後，背景視窗的操作將完全暫停，直到壟斷的視窗被關閉才可操作其他視窗。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.12. 直接視窗

### 13.12.1. 概要

**[直接視窗]** 物件是用位元暫存器去控制彈出視窗的開啟及關閉。首先，在視窗上定義一個顯示區域，當所指定的位元暫存器的狀態改變時，將在此顯示區域內顯示此視窗的內容。所顯示視窗的寬度與高度不會大於此顯示區域。將控制此彈出視窗的位元暫存器狀態恢復即可關閉此彈出視窗。

**[直接視窗]** 與 **[間接視窗]** 的差別在於直接視窗是利用位元狀態進行視窗控制，而間接視窗則是利用字組數值進行視窗控制。

### 13.12.2. 設定



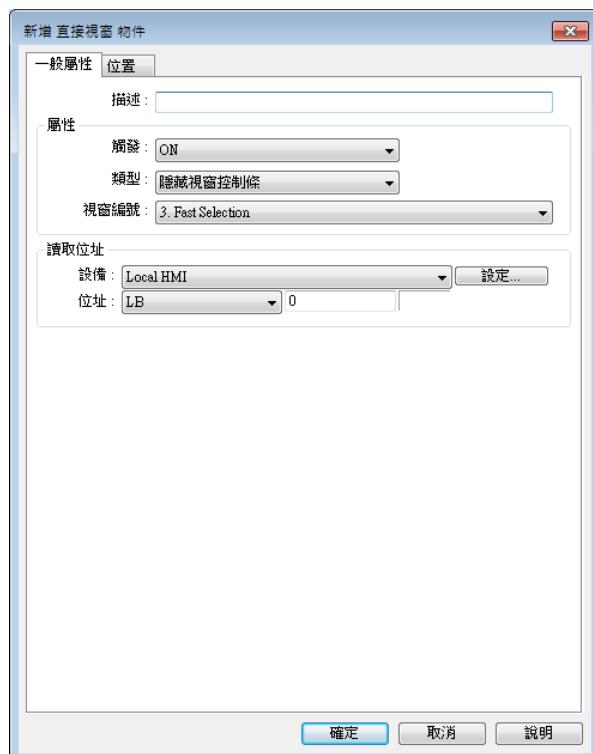
按下工作列的 **[物件] » [嵌入視窗] » [直接視窗]** 按鈕後即會開啟 **[直接視窗]** 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 **[直接視窗]** 物件。

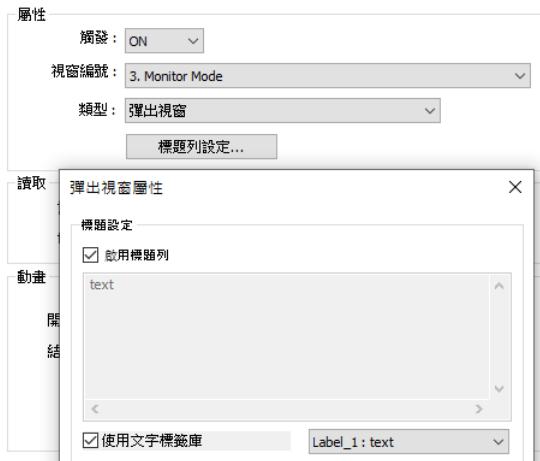
#### 一般屬性設定

cMT / cMTX 系列



eMT、iE、XE、mTV 系列



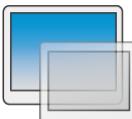
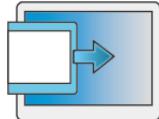
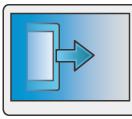
設定	描述
類型	僅支援 cMT / cMT X 系列。此功能設置會影響當基本視窗有使用重疊視窗時，重疊視窗上的直接視窗物件的顯示方式。請參考 13.11 間接視窗的範例 2。
嵌入父視窗	直接視窗物件顯示在該父視窗上。若直接視窗物件是放在重疊視窗上，在基本視窗上呼叫此重疊視窗的直接視窗物件時，物件是顯示在該重疊視窗上。
彈出視窗	直接視窗物件顯示在基本視窗上。若直接視窗物件是放在重疊視窗上，在基本視窗上呼叫此重疊視窗的直接視窗物件時，物件是顯示在基本視窗上。
標題列設定	選擇是否啟用標題列並設定標題名稱。
讀取位址	<p>點選 [設定] 後選擇位元暫存器設備類型的[PLC 名稱]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來控制視窗彈出。使用者也可在 [一般屬性] 頁中設定位址。</p>
屬性	<p>類型</p> <p>設定彈出視窗樣式。支援兩種樣式：</p> <ul style="list-style-type: none"> <li>● 隱藏視窗控制條 彈出的子視窗不包含視窗控制條，無法拖曳移動視窗。</li> </ul>  <ul style="list-style-type: none"> <li>● 顯示視窗控制條 彈出的子視窗包含視窗控制條，它的視窗位置可藉由控制條任意被拖曳。</li> </ul>

**視窗編號**

設定您彈出的視窗號碼。

**動畫  
(cMT / cMT X  
系列)**

**效果**  
可設定視窗 [開始] 與 [結束] 的動畫效果。

效果	樣式
淡出	
飛入	
飄入	
擦去	
分割	
環狀	
時鐘	
縮放	

旋轉	
----	--

**持續時間**

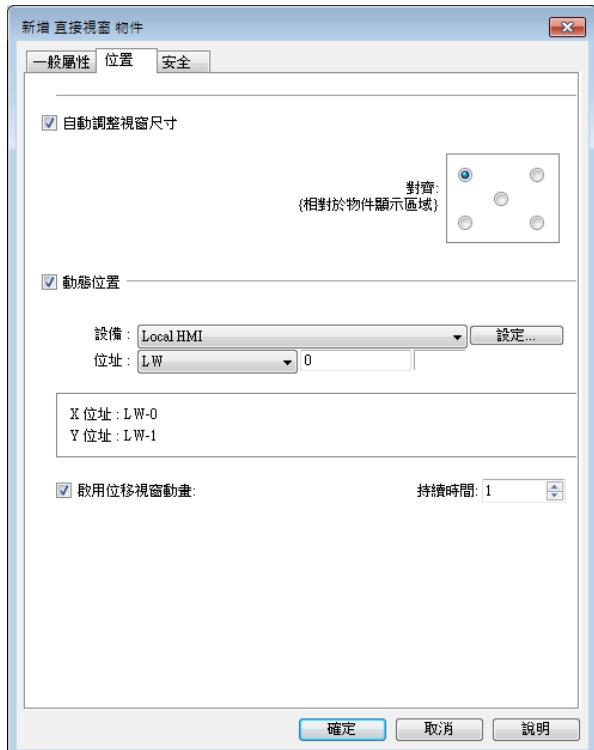
可以設定視窗開始與結束的速度。

**顯示方向**

可以設定視窗開始與結束的出入方向。

## 位置設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



設定	描述
<b>自動調整視窗尺寸</b>	系統會根據彈出的視窗的尺寸調整其顯示範圍，並自動調整相對位置。 <b>(相對於物件顯示區域)</b> 彈出的視窗的位置基準點 (對應於間接視窗物件)。如：若設定右下角為基準點，則任一視窗彈出時，會以自身的右下角對準此基準點。若設定左上角為基準點，則任一視窗彈出時，會以自身的左上角對準此基準點，依此類推。請見以下範例 1。
<b>動態位置</b>	啟用此功能後，可在 HMI 上透過指定位址調整視窗的位置。 <b>啟用位移視窗動畫</b> 設定當視窗位置移動時，是否使用動畫。

## 範例 1

若有一個彈出視窗編號 11，使用 [位元狀態切換開關] 控制，位址設定為 LB-0。

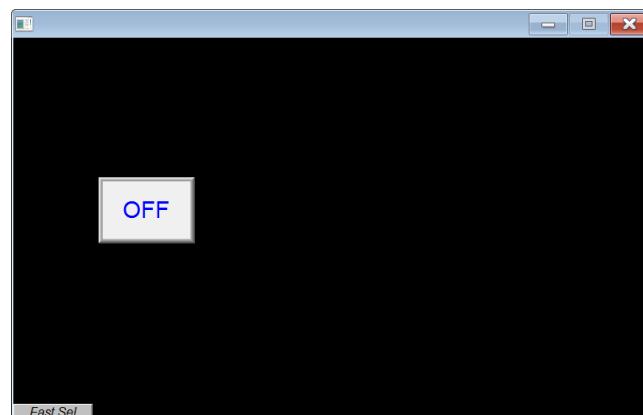
1. 建立一個 [直接視窗] 物件，讀取位址為 LB-0。
2. 勾選 [自動調整視窗尺寸] 並調整好彈出視窗的顯示區塊，此範例將基準點設為右下角。



3. LB-0 狀態為 ON，則彈出視窗編號 11。



4. LB-0 狀態為 OFF，則關閉彈出視窗。



### Note

- eMT、iE、XE、mTV 系列最多可同時開啟 24 個彈出視窗，包含系統訊息視窗、直接視窗和間接視窗；cMT / cMT X 系列則無彈出視窗的數量上限。
- 系統不允許在一個基本視窗上使用 2 個直接 (或間接) 視窗彈出同一個視窗。
- 如果彈出的視窗有 [壟斷] 屬性，當視窗彈出後，背景視窗的操作將完全暫停，直到壟斷的視窗被關閉才可操作其他視窗。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.13. 移動/旋轉圖形

### 13.13.1. 概要

**[移動/旋轉圖形]** 物件可定義物件的狀態，移動距離與旋轉角度。物件會根據讀取位址及連續的暫存器內的數據，改變物件的狀態，移動距離與旋轉角度。

### 13.13.2. 設定



按下工作列的 **[物件] » [動畫] » [移動/旋轉圖形]** 按鈕後即會開啟 **[移動/旋轉圖形]** 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 **[移動/旋轉圖形]** 物件。

### 一般屬性設定



設定	描述
讀取位址	點選 <b>[設定]</b> 後選擇字組暫存器設備類型的 <b>[設備]</b> ， <b>[位址]</b> ， <b>[設備類</b>

型]，[系統暫存器]，[索引暫存器]來做為控制移動圖形狀態和移動位置的目標。使用者也可在 [一般屬性] 頁中設定位址。

**屬性** 選擇物件的移動方式及移動的範圍，請見下方《13.13.2.1 屬性模式說明》。

**顯示比例** 物件各個狀態的圖形在顯示時，可以分開設定縮放比例，參考下圖。



**旋轉** 選擇物件的旋轉方式，以及範圍。可選擇順時針或逆時針旋轉。最小/最大角度的單位為度，設定範圍可以為 0 度到 360 度，最大角度必須大於最小角度。請見下方《13.13.2.2 旋轉模式說明》。

**限制值位址** 利用 [限制值位址] 內的數據動態調整物件的顯示區域，請見範例 1。

## 範例 1

利用 [限制值位址] 內的數據動態調整物件的顯示區域。當限制值位址為 LW-n，則 X 軸、Y 軸與旋轉角度的上下限會根據以下的規則自動被設定為：

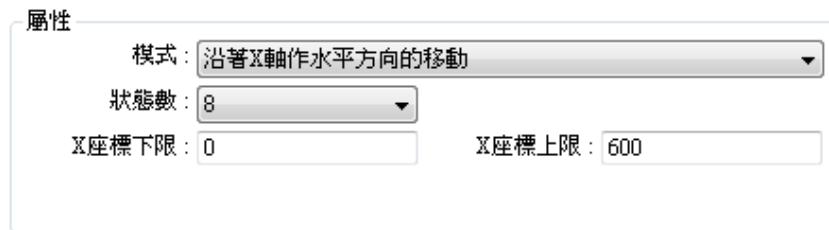
資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
位置-X 軸座標下限	LW-n	LW-n	LW-n
位置-X 軸座標上限	LW-n+1	LW-n+2	LW-n+4
位置-Y 軸座標下限	LW-n+2	LW-n+4	LW-n+8
位置-Y 軸座標上限	LW-n+3	LW-n+6	LW-n+12
旋轉-輸入下限	LW-n+4	LW-n+8	LW-n+16
旋轉-輸入上限	LW-n+5	LW-n+10	LW-n+20
旋轉-比例下限	LW-n+6	LW-n+12	LW-n+24
旋轉-比例上限	LW-n+7	LW-n+14	LW-n+28

### 13.13.2.1. 屬性模式說明

(以下假設讀取地址使用 LW-n)

- 沿著 X 軸作水平方向的移動

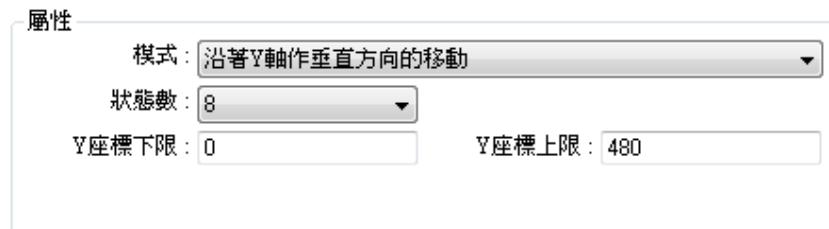
只允許物件沿著 X 軸作水平方向的移動。移動範圍由 [X 軸座標下限] 與 [X 軸座標上限] 來決定。



資料格式	16-bit	32-bit	64-bit (只支援於 cMT)
控制物件狀態位址	LW-n	LW-n	LW-n
控制物件 X 軸移動距離位址	LW-n+1	LW-n+2	LW-n+4

- 沿著 Y 軸作垂直方向的移動

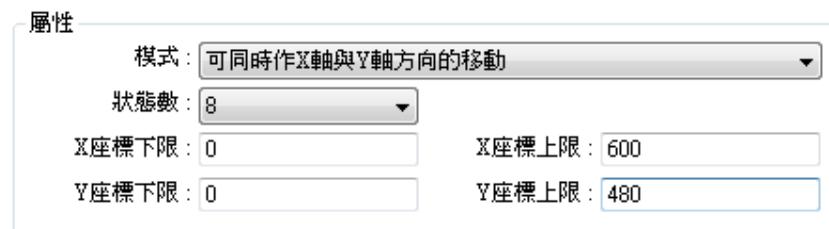
只允許物件沿著 Y 軸作垂直方向的移動。移動範圍由 [Y 軸座標下限] 與 [Y 軸座標上限] 來決定。



資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
控制物件狀態位址	LW-n	LW-n	LW-n
控制物件 Y 軸移動距離位址	LW-n+1	LW-n+2	LW-n+4

- 可同時作 X 軸與 Y 軸方向的移動

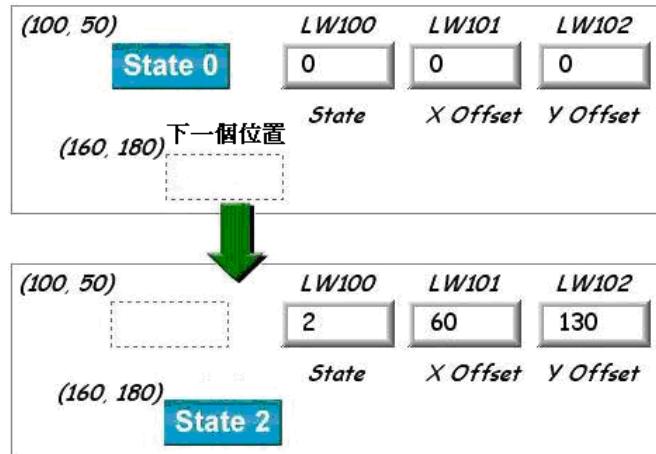
允許物件沿著 X 軸與 Y 軸移動。移動範圍由 [X 軸座標下限]、[X 軸座標上限] 與 [Y 軸座標下限]、[Y 軸座標上限] 來決定。



資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
控制物件狀態位址	LW-n	LW-n	LW-n
控制物件 X 軸移動距離位址	LW-n+1	LW-n+2	LW-n+4
控制物件 Y 軸移動距離位址	LW-n+2	LW-n+4	LW-n+8

舉例來說，若暫存器為 LW-100，且變數型態使用[16-bit Unsigned]，則 LW-100 存放物件的狀態，LW-101 存放 X 軸方向的移動距離，LW-102 存放 Y 軸方向的移動距離。

以下圖為例，物件的位址為 LW-100 且起始位址為(100, 50)，假使現在要移動物件至(160, 180)且顯示狀態 2 的圖形，則 LW-100 需設定為 2， $LW-101 = 160 - 100 = 60$ ， $[LW102] = 180 - 50 = 130$



- 沿著 X 軸按比例作水平方向的移動

只允許物件沿著 X 軸、按比例作水平方向的移動。

$$\text{公式：位移距離} = (\text{讀取位址數據} - \text{輸入下限}) \times \frac{\text{比例上限} - \text{比例下限}}{\text{輸入上限} - \text{輸入下限}}$$

資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
控制物件狀態位址	LW-n	LW-n	LW-n
控制物件 X 軸移動距離位址	LW-n+1	LW-n+2	LW-n+4

- 沿著 Y 軸按比例作垂直方向的移動

只允許物件沿著 Y 軸、按比例作垂直方向的移動。公式與 [沿著 X 軸按比例作水平方向的移動] 相同。

資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
控制物件狀態位址	LW-n	LW-n	LW-n
控制物件 Y 軸移動距離位址	LW-n+1	LW-n+2	LW-n+4

- 沿著 X 軸按反比例作水平方向的移動

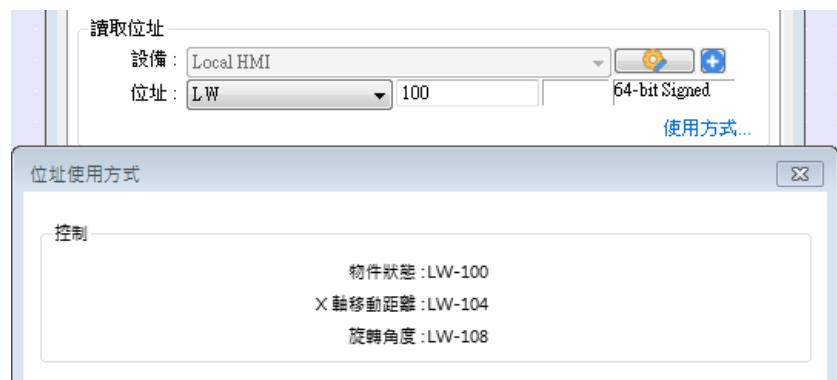
此項功能與 [沿著 X 軸按比例作水平方向的移動] 相同，但移動方向相反。

- 沿著 Y 軸按反比例作垂直方向的移動

此項功能與 [沿著 Y 軸按比例作垂直方向的移動] 相同，但移動方向相反。

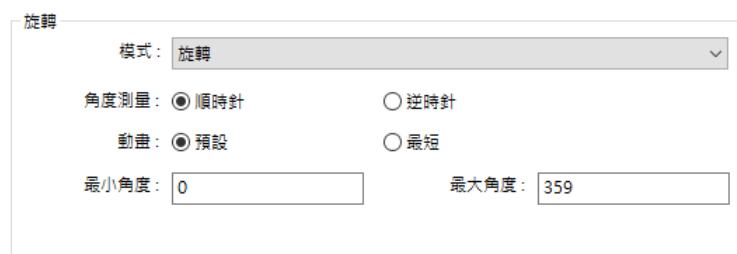
### 13.13.2.2. 旋轉模式說明

控制物件旋轉的位址為始自讀取位址 (LW-n) 的連續位址，這些位址可能因選擇的模式不同而變化，其對應的位址請參考讀取位址底下的 [使用方式...]。



### ● 旋轉

將物件以順時針或逆時針的方式旋轉。旋轉角度範圍由 [最小角度] 與 [最大角度] 來決定。動畫可以選擇預設或是最短的方式來呈現。預設是依照兩角度間之線性插值旋轉，舉例來說當角度測量方式設定為順時針，旋轉角度從 10 度轉到 350 度時會順時針旋轉 340 度；而最短會沿產生最短動畫路徑方向旋轉，也就是說當 10 度轉到 350 度時會逆時針旋轉 20 度。



### ● 按比例旋轉

將物件以順時針或逆時針的方式旋轉。根據下方比例公式來決定旋轉角度。

$$\text{旋轉角度} = (\text{讀取位址數據} - [\text{輸入下限}]) \times \frac{[\text{比例上限}] - [\text{比例下限}]}{[\text{輸入上限}] - [\text{輸入下限}]}$$

### ● 按反比例旋轉

此項功能與 [按比例旋轉] 相同，但旋轉方向相反。

## 13.14. 動畫

### 13.14.1. 概要

使用者可以預先定義 [動畫] 物件的移動軌跡，並利用更改暫存器內的數據，控制物件的狀態與物件在移動軌跡上的位置。系統將使用連續兩個暫存器內的數據來控制動畫物件，第一個暫存器為控制物件的狀態，第二個為控制物件的位置。

### 13.14.2. 設定

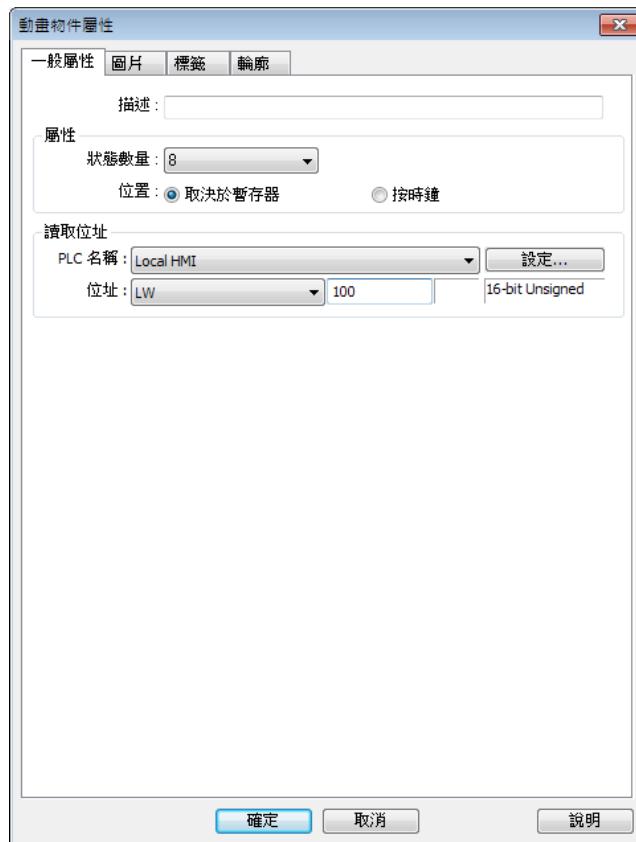


按下工作列的 [物件] » [動畫] » [動畫] 按鈕。首先，設定物件的移動路徑。使用滑鼠在編輯畫面上點擊左鍵一一指定每個移動位置，然後點擊右鍵即會開啟 [動畫] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [動畫] 物件。



要更改物件的屬性，可以使用滑鼠左鍵雙擊物件所在位置，利用出現的 [動畫] 物件屬性對話窗，即可更改物件的各項屬性。

## 一般屬性設定



設定	描述						
<b>屬性</b> <p><b>狀態數量</b> 設定物件狀態數目。</p> <p><b>取決於暫存器</b> 由暫存器數據控制物件的狀態和位置，此時必須正確設定物件狀態與位置的讀取位址。請見以下範例 1。</p> <p><b>按時鐘</b> 物件自動改變狀態與顯示位置，[自動控制位置] 項目用來設定狀態與顯示位置改變方式。</p> <p><b>自動控制位置</b></p> <table border="1"> <tr> <td>速度: 10</td> <td>*0.1 秒</td> </tr> <tr> <td>狀態轉換: 基於時間</td> <td><input checked="" type="checkbox"/> 返回</td> </tr> <tr> <td>轉換週期: 5</td> <td>*0.1 秒</td> </tr> </table> <p><b>速度</b>: 位置改變的速度，單位為 0.1 秒。例如設定為 10，則物件每隔 1 秒鐘變換一個位置。</p> <p><b>狀態轉換</b>: 狀態改變的方式，可以選擇 [基於位置] 與 [基於時間]。選擇 [基於位置] 表示位置改變，狀態也隨著改變。若選擇 [基於時間]，表示位置使用固定的頻率自動變換，變換頻率在 [轉換週期] 中設定。</p>	速度: 10	*0.1 秒	狀態轉換: 基於時間	<input checked="" type="checkbox"/> 返回	轉換週期: 5	*0.1 秒	
速度: 10	*0.1 秒						
狀態轉換: 基於時間	<input checked="" type="checkbox"/> 返回						
轉換週期: 5	*0.1 秒						

返回：假設物件有 4 個位置，分別為 position 0、position 1、position 2、position 3。若未選擇此項設定，當移動到最後一個位置(position 3)後，物件將移動到初始位置 position 0，再重複原來位置改變方式，移動位置整理順序如下。

position 0 → position 1 → position 2 → position 3 → position 0 → position 1 → position 2...

若選擇此項設定，當移動到最後一個位置 (position 3) 後，物件將使用反向的移動方式，移動到初始位置 position 0，再重複原來位置改變方式，移動位置整理順序如下。

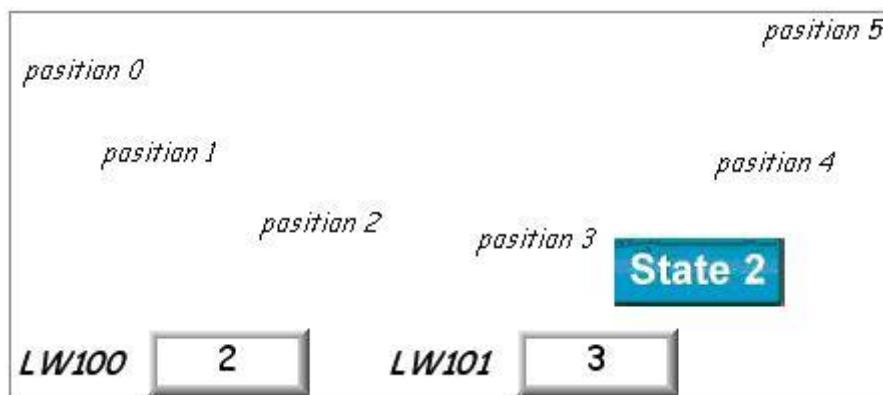
position 0 → position 1 → position 2 → position 3 → position 2 → position 1 → position 0...

## 範例 1

如果物件的狀態與位置由暫存器中的數據決定，就必須正確設定物件狀態與位置的讀取位址。讀取位址格式如下表。

資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
控制物件狀態位址	LW-n	LW-n	LW-n
控制物件位置位址	LW-n+1	LW-n+2	LW-n+4

舉例來說，若暫存器為 LW-100，且格式使用[16-bit Unsigned]，則 LW-100 存放物件的狀態，LW-101 存放物件的顯示位置。以下圖為例，LW-100 = 2，LW-101 = 3，所以物件顯示狀態 2，並出現在位置 3。



## 輪廓設定



設定	描述
向量圖尺寸	設定物件所顯示圖形的大小。
軌跡	設定移動軌跡上各點的位置。

### Note

- 因為一個動畫物件可設定多個不同的圖片，因此無法使用【使用原尺寸】將所有的圖片還原成原始尺寸。

## 13.15. 棒圖

### 13.15.1. 概要

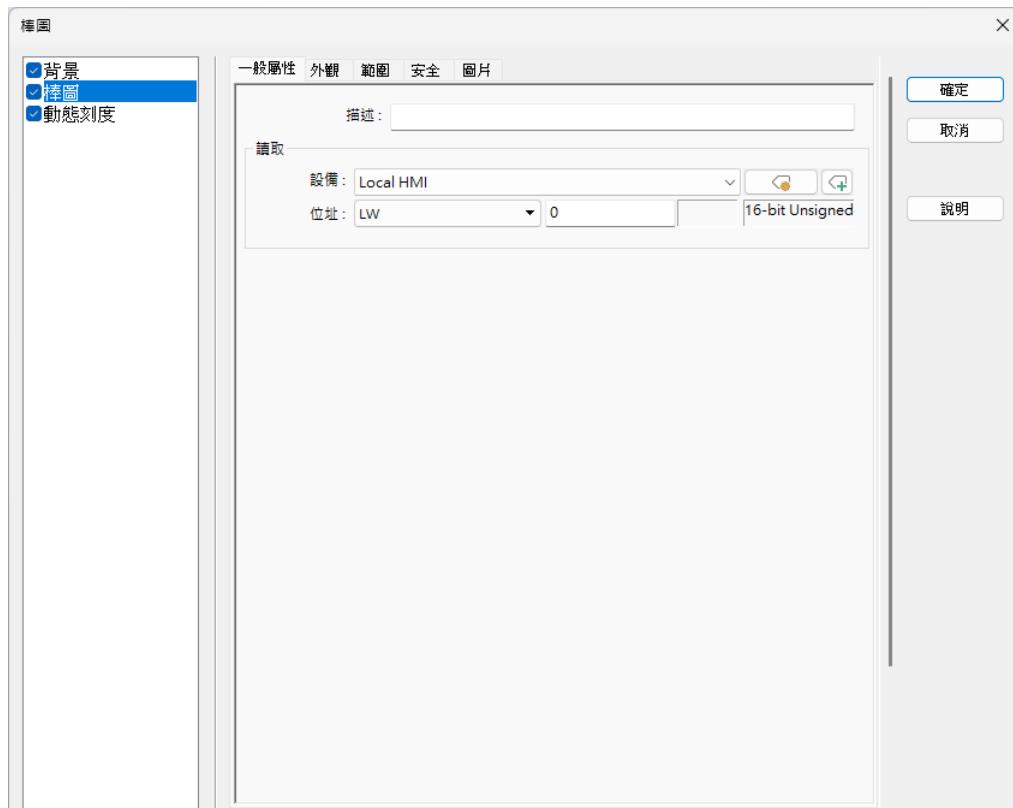
[棒圖] 物件使用百分比例與棒圖的方式，顯示暫存器中的數據。

### 13.15.2. 設定



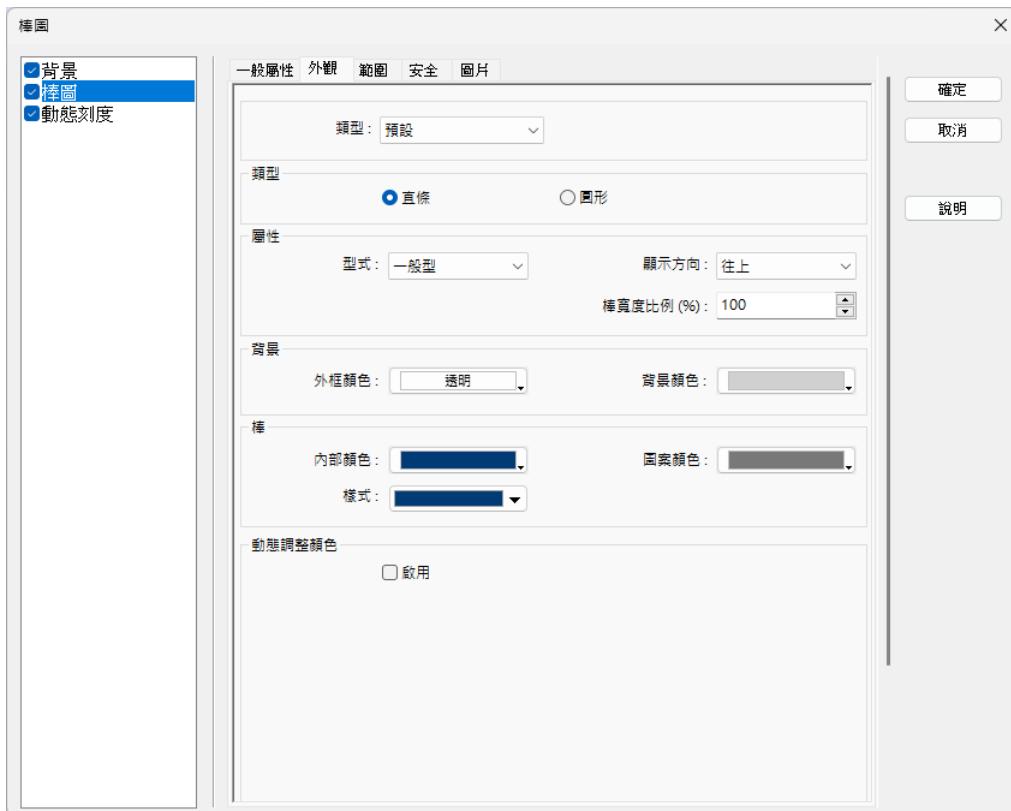
按下工作列的 [物件] » [曲線圖] » [棒圖] 按鈕後即會開啟 [棒圖] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [棒圖] 物件。

#### 一般屬性設定



設定	描述
讀取位址	點選 [設定] 後選擇字組暫存器設備類型的 [設備]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來做為棒圖顯示的數據依據。

## 外觀設定



### 設定

### 描述

#### 類型

可選擇 [直條] 或 [圓形] 種類的棒圖。

#### 屬性

#### 型式

可選擇 [一般型] 或 [偏差型]。當選擇 [偏差型] 時，需設定原點位置。

#### 顯示方向/角度

[直條] 棒圖可以選擇棒圖的顯示方向，分別為 [朝上顯示]、[朝下顯示]、[朝右顯示]、[朝左顯示]。

[圓形] 棒圖可以選擇棒圖的 [移動方向] 與 [角度]。



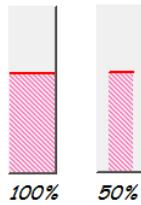
[移動方向] 可以選擇 [順時針] 或 [逆時針]。

[角度] 若選擇 [全圓]，則須設定 [起始角度]。

若未選擇 [全圓]，則須設定 [起始角度] 與 [結束角度]。

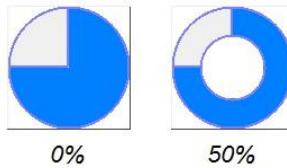
#### 棒寬度比例(%)

[直條] 棒圖可以設定棒圖的顯示寬度與物件寬度間的百分比率。以下圖示顯示不同比例，100% 以及 50%。



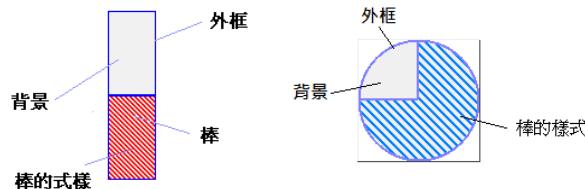
#### 圓心半徑(%)

[圓形] 棒圖可以設定棒圖圓心中空的比例。以下圖示顯示不同比例，0% 以及 50%。



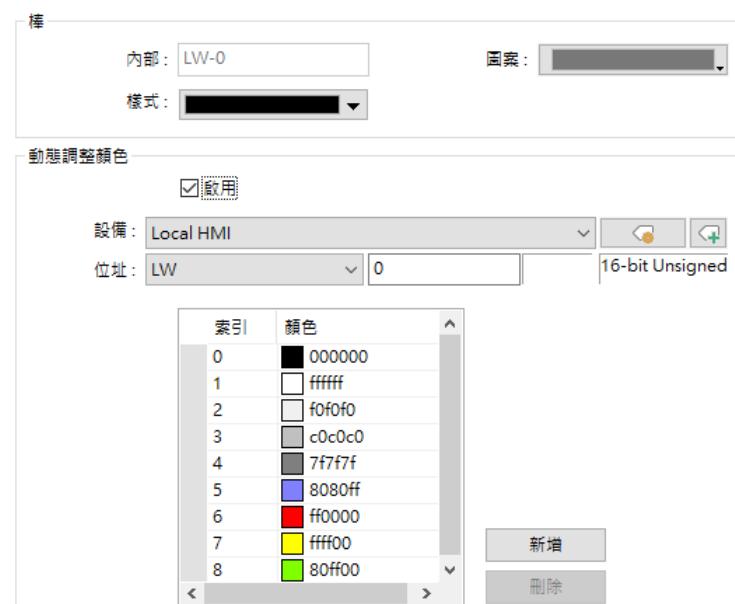
#### 顏色/樣式

設定棒圖外框、背景顏色與填充區域的樣式與顏色，參考下圖。

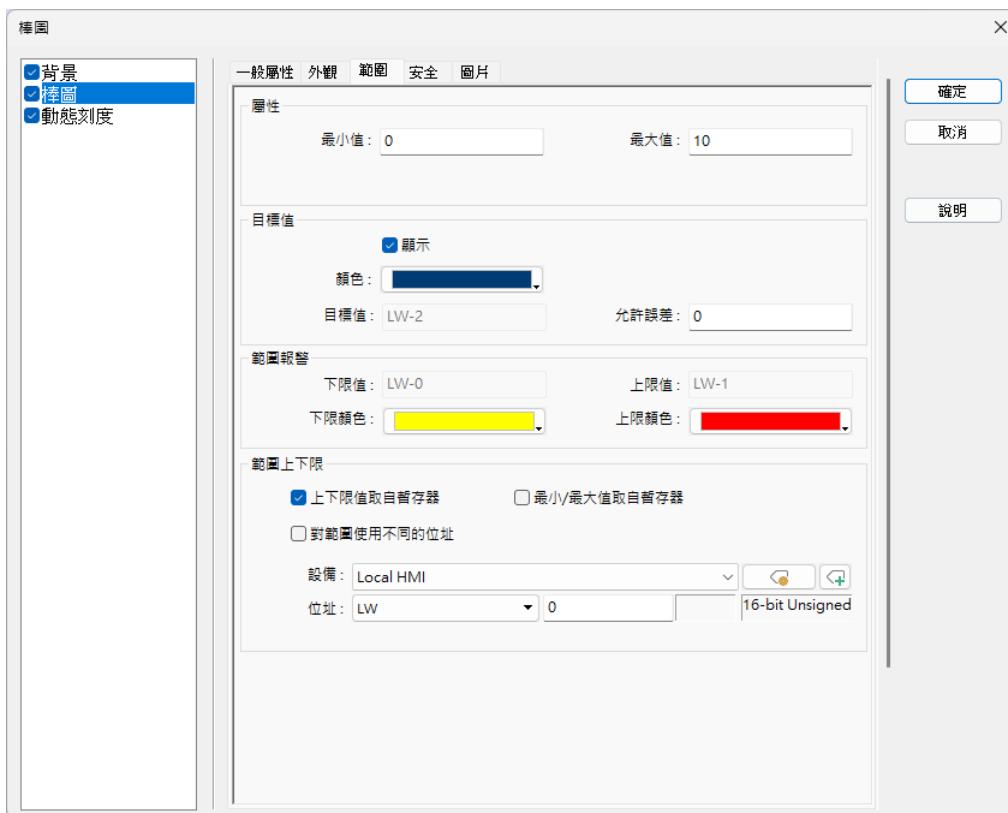


#### 動態調整顏色

cMT / cMT X 系列支援動態調整顏色，動態調整顏色啟用後，棒內部顏色可由設定位址決定。



## 範圍設定



設定	描述
<b>最小值/最大值</b>	棒圖填充的百分比可以利用公式換算而得，請見以下範例 1。
<b>目標值</b>	當暫存器內的數據符合條件時，填充區域的顏色可以變更為此項目所定義的顏色。請見以下範例 2。
<b>範圍報警</b>	當數據大於【上限值】時，填充區域的顏色可以變更為【上限顏色】所定義的顏色；若當數據小於【下限值】時，填充區域的顏色可以變更為【下限顏色】所定義的顏色。 請注意，cMT / cMT X 系列動態調整顏色啟用後，目標值顏色、警報下限顏色及上限顏色設定將會關閉，由動態調整顏色位置管理。
<b>範圍上下限</b>	當選擇【上下限值取自暫存器】，【範圍報警】中所使用的【下限值】、【上限值】與【目標值】皆讀取自指定的暫存器。指定之暫存器會顯示在輸入的欄位之中。請見以下範例 3。 當選擇【最小/最大值取自暫存器】，【屬性】中所使用的【最小值】、【最大值】皆讀取自指定的暫存器。指定之暫存器會顯示在輸入的欄位之中。 當選擇【對範圍使用不同的位址時】，上述的位址皆可獨立設定，讀取自不同的暫存器。此選項僅支援於 cMT / cMT X。

## 範例 1

棒圖填充的百分比可以利用下列的公式換算而得：

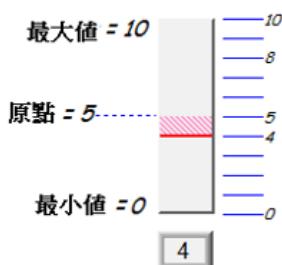
$$\text{顯示區域百分比(一般型)} = \frac{\text{暫存器數據} - \text{最小值}}{\text{最大值} - \text{最小值}} \times 100\%$$

$$\text{顯示區域百分比(偏差型)} = \frac{\text{暫存器數據} - \text{原點位置}}{\text{最大值} - \text{最小值}} \times 100\%$$

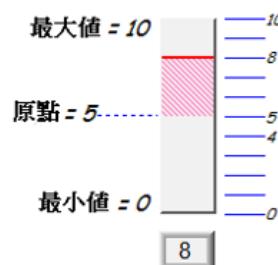
若 (暫存器數據 - [原點位置]) 大於 0，則棒圖將由 [原點位置] 的位置往上填充；若 (暫存器數據 - [原點位置]) 小於 0，則棒圖將由 [原點位置] 的位置往下填充。

下圖顯示在 [原點位置] 設定為 5，[最大值] 為 10，[最小值]為 0 並使用不同數據時，棒圖的填充情形。

當讀取數值為 4



當讀取數值為 8

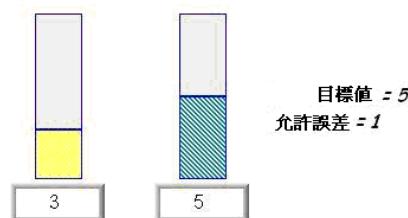


## 範例 2

當暫存器內的數據符合下列的條件時，填充區域的顏色可以變更為此項目所定義的顏色。

$$[\text{目標值}] - [\text{允許誤差}] \leq \text{暫存器內的數據} \leq [\text{目標值}] + [\text{允許誤差}]$$

參考下圖，此時 [目標值] = 5，[誤差值] = 1，則暫存器的值大於或等於  $5-1=4$ ，且小於或等於  $5+1=6$ ，填充區域的部分將改變為“目標值顏色”。



## 範例 3

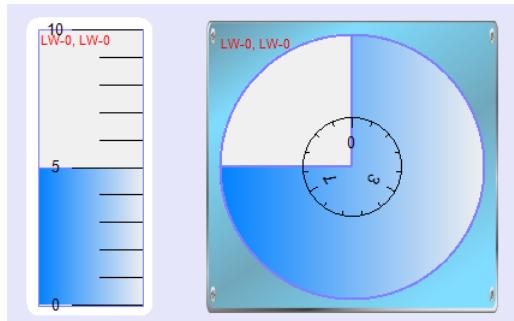
範圍報警的上下限可由指定暫存器設定。當寫入位址為 LW-n，則上/下限及相關設定的位址會根據以下的規則自動設定：(假設 [最小/最大值取自暫存器] 已勾選)

資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
警報下限值	LW-n	LW-n	LW-n

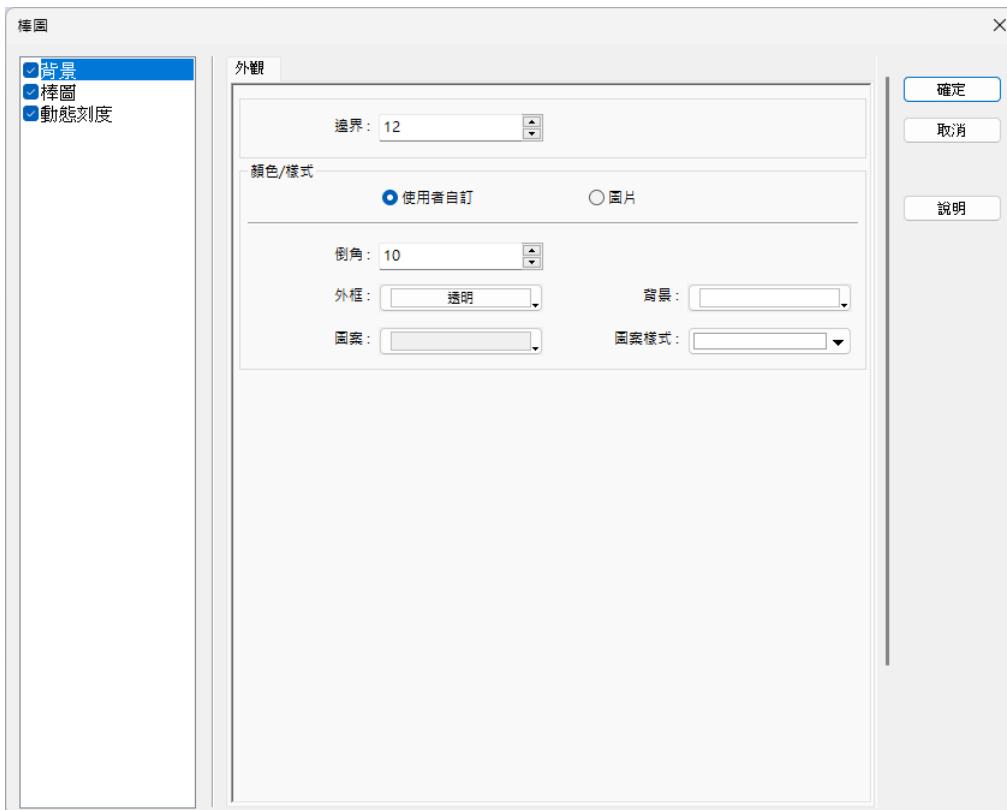
<b>警報上限值</b>	LW-n+1	LW-n+2	LW-n+4
<b>目標值</b>	LW-n+2	LW-n+4	LW-n+8
<b>最小值</b>	LW-n+3	LW-n+6	LW-n+12
<b>最大值</b>	LW-n+4	LW-n+8	LW-n+16
<b>原點位置</b>	LW-n+5	LW-n+10	LW-n+20

## 複合物件

cMT / cMT X 系列提供一次性設定相關元件的功能。除了棒圖以外，增加背景與動態刻度的元素供使用者更能活用及美化棒圖的設計。



## 背景設定



設定	描述
<b>邊界</b>	邊緣與物件的留白距離。
<b>顏色/樣式</b>	使用者自訂



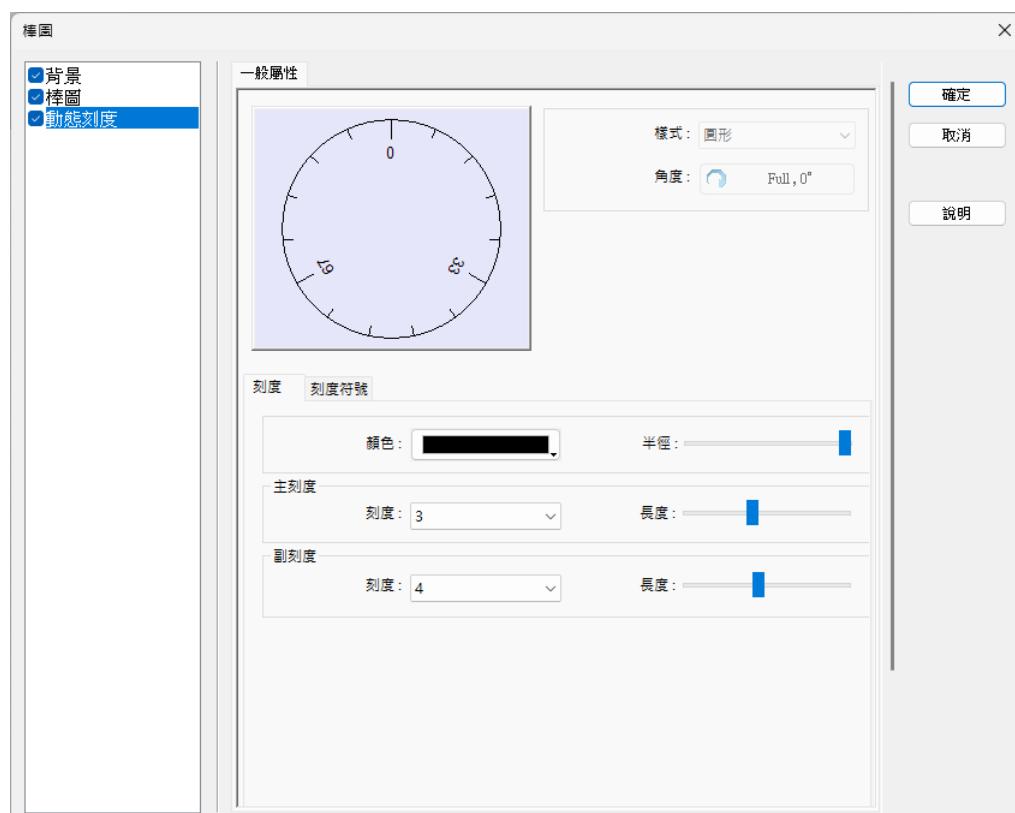
根據圖案樣式與顏色挑選自訂背景。

### 圖片



可使用內建的背景圖片或是圖片庫中的檔案。

## 動態刻度



### 設定

### 描述

#### 樣式

將依棒圖所設定的類型顯示。

#### 刻度

選擇 [主刻度] 與 [副刻度] 的分割數量，若樣式為圓形時，可設定刻度的半徑與長度。

#### 刻度符號

設定刻度顯示時的字型、顏色、尺寸與其他屬性。

## 13.16. 錶針

### 13.16.1. 概要

[錶針] 物件會使用儀表圖的方式，指示目前暫存器中的數據。

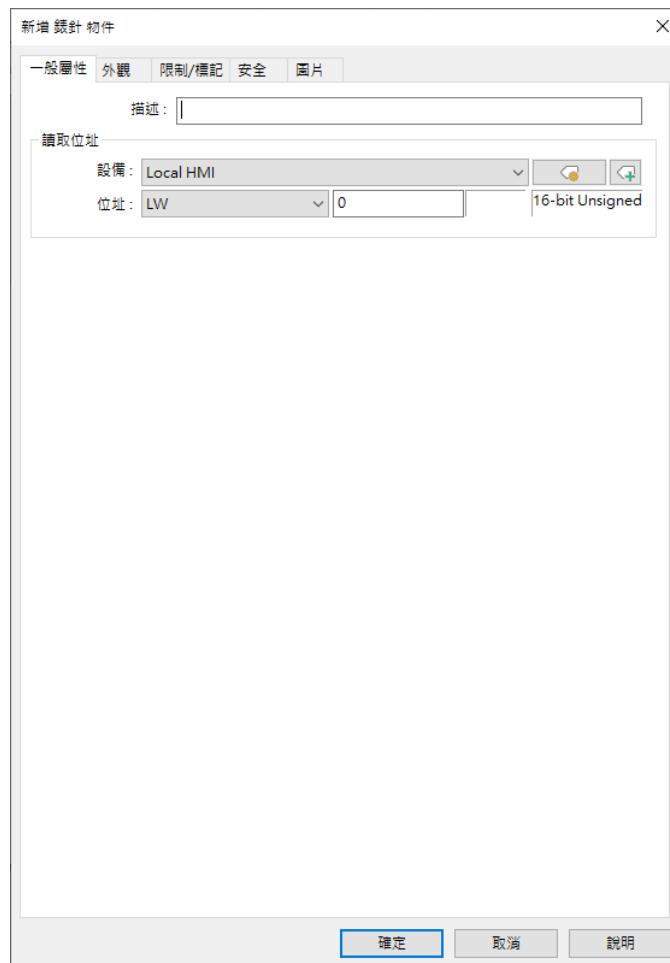
### 13.16.2. 設定



按下工作列的 [物件] » [曲線圖] » [錶針] 按鈕後即會開啟 [錶針] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [錶針] 物件。

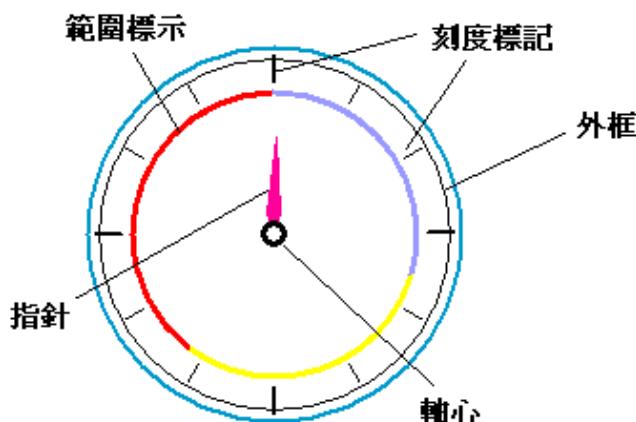
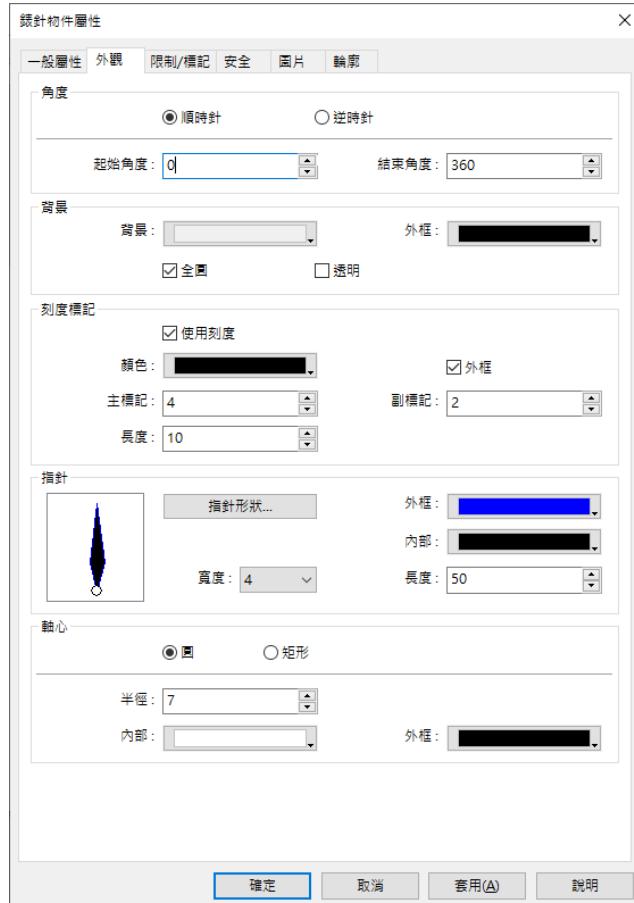
#### 13.16.2.1. eMT、iE、XE、mTV 系列

##### 一般屬性設定



設定	描述
讀取位址	點選 [設定] 後選擇字組暫存器設備類型的 [設備]，[位址]，[設備類型]，[系統暫存器]，[索引暫存器] 來做為錶針顯示的數據依據。使用者也可在 [一般屬性] 頁中設定位址。

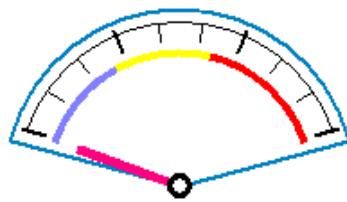
## 外觀設定



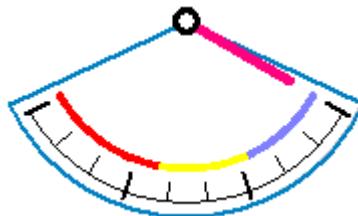
設定	描述
角度	設定指針轉動方向為 [順時針] 或 [逆時針]。 錶針物件以圖形中點上方為起始點，表示為 0 度或 360 度。向左為逆時針，向右為順時針。角度可設定範圍皆為 0~360 度。不同的設定值

所顯示的結果，可參考下面的幾種不同的設定。

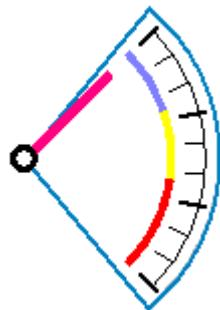
- [起始角度] = 290°  
[結束角度] = 70°



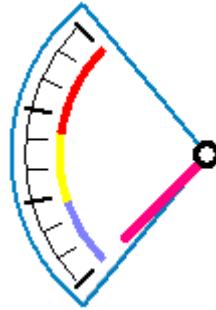
- [起始角度] = 120°  
[結束角度] = 240°



- [起始角度] = 40°  
[結束角度] = 140°



- [起始角度] = 225°  
[結束角度] = 315°

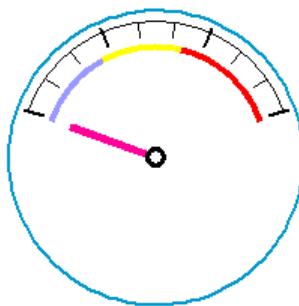


## 背景

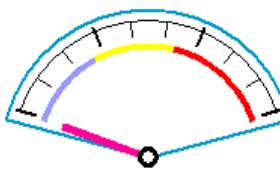
設定物件的背景與圓周的顏色。

## 全圓

當選擇【全圓】時，錶針物件將顯示整個圓形，反之則顯示被定義的角度範圍。



使用全圓



非使用全圓

## 透明

當選擇【透明】時，錶針物件將不會顯示背景與外框顏色。

## 刻度標記

設定標記的數量與顏色。

## 指針

設定指針的形狀，長寬度和顏色。

## 軸心

設定軸心的樣式與顏色。

## 限制 / 標記設定



### 設定

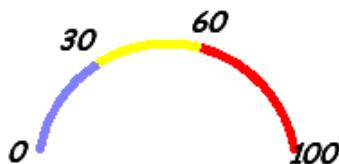
### 描述

#### 數值

設定物件所要顯示的數值範圍。請見以下範例 1。

#### 範圍

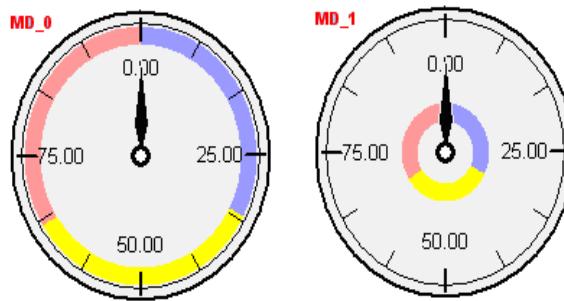
設定上、下限值及指示的顏色與寬度。



#### 使用者自定半徑

數值範圍顯示色彩離圓心的距離。

例如，設為 80： 設為 30：

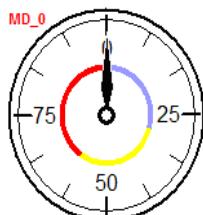


### 上下限取自暫存器

上下限可由指定暫存器設定。請見以下範例 2。

#### 刻度符號

設定是否使用刻度符號於錶針上。



### 範例 1

設定物件所要顯示的數值範圍。指針的指示角度計算：

$$\text{角度(度)} = \frac{\text{讀取數據} - \text{最小值}}{\text{最大值} - \text{最小值}} \times (\text{結束角度} - \text{起始角度})$$

假設讀取的數據為 30，起始角度  $0^\circ$ ，結束角度  $360^\circ$ ，最小值 0，最大值 100，則指針的指示角度為：

$$\text{角度(度)} = \frac{30 - 0}{100 - 0} \times (360 - 0) = 108(\text{度})$$

### 範例 2

上下限可由指定暫存器設定。當寫入位址為 LW-n，則上/下限會根據以下的規則自動被設定為：

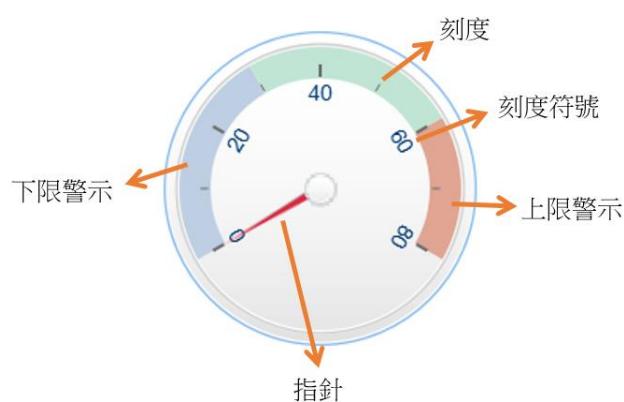
位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
下限	LW-n	LW-n	LW-n
上限	LW-n+1	LW-n+2	LW-n+4

以下表為例，當 [暫存器位址] 為 LW-100 時，則上/下限的位址會自動被設定為：

位址格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
下限	LW-100	LW-100	LW-100
上限	LW-101	LW-102	LW-104

### 13.16.2.2. cMT / cMT X 系列

#### 一般屬性設定



**樣式**

可自下拉選單選擇 [自訂]、[樣式 1]、[樣式 2]、[標準]。

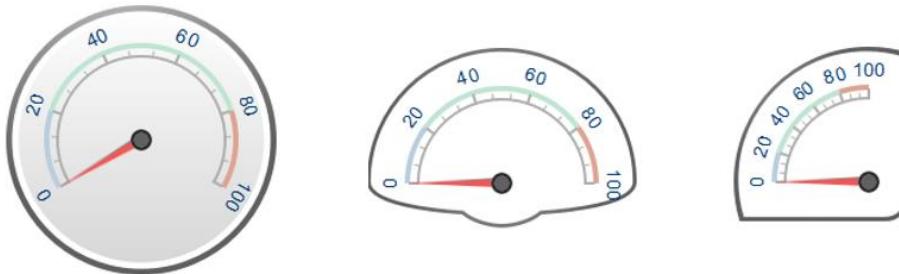
選擇 [自訂] 時，需自訂錶針的各種屬性，包含指針長寬、軸心半徑等。選擇 [樣式 1]、[樣式 2]、[標準] 樣式時，可自行設定以下背景選項。

**顏色**

錶針圖片的背景色。

**外觀**

下圖是樣式 1 的全圓、半圓及四分之一圓的顯示樣式。



下圖是樣式 2 的全圓、半圓及四分之一圓的顯示樣式。

**旋轉**

錶針的背景圖片會依照角度順時針旋轉指定的度數。

**角度****移動方向**

調整指針的刻度範圍。錶針物件以圖形中點上方為起始點，表示為 0 度，向右為順時針，向左為逆時針。

**角度**

可以選擇[預設]或是[自訂]。選擇預設時，可以透過圖示選項快速地設定選擇的角度及旋轉方向。



### 全圓

勾選後，錶針會從 [起始角度] 的角度作為原點，根據 [順時針 / 逆時針] 方向畫一個全圓。數據的最小值及最大值於 [數值] 欄位中的 [最小值] 及 [最大值] 設置。

<b>啟用動畫</b>	設定錶針移動時是否有滑動式移動至指定位置。若不勾選則錶針在數據變換後會直接跳到指定位置。
<b>數值</b>	設定錶針的上下限。
<b>讀取位址</b>	錶針顯示的數據來源位址。
<b>指針</b>	設定指針及軸心的樣式。若採用 [自訂] 模式，指針的指向方向必須為朝上才可正確顯示。
<b>範圍</b>	設定上下限數值與警示的顏色。
	<b>動態限制</b>
	上下限警示的範圍由指定暫存器設定。請見上一小節之範例 2。 若選擇使用不同位址，上下限範圍可設定成讀取自不同的暫存器。
<b>刻度</b>	設定刻度標記間隔及顏色。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.17. 趨勢圖

### 13.17.1. 概要

[趨勢圖] 物件會將設定在 [資料取樣] 中的資料利用連續的線段描繪出圖，以利資料分析。

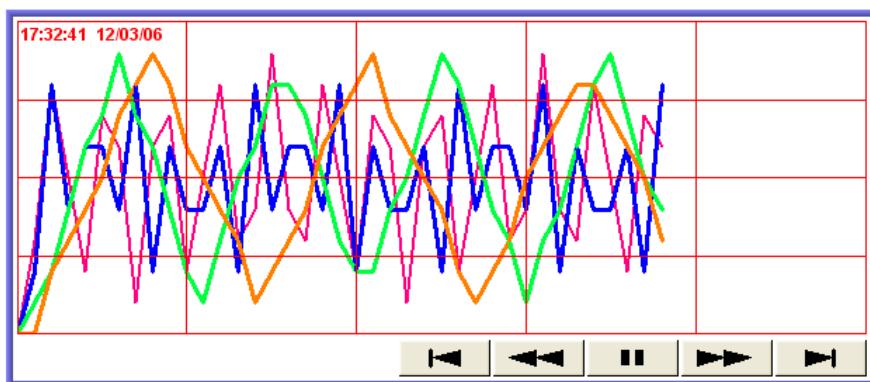
### 13.17.2. 設定



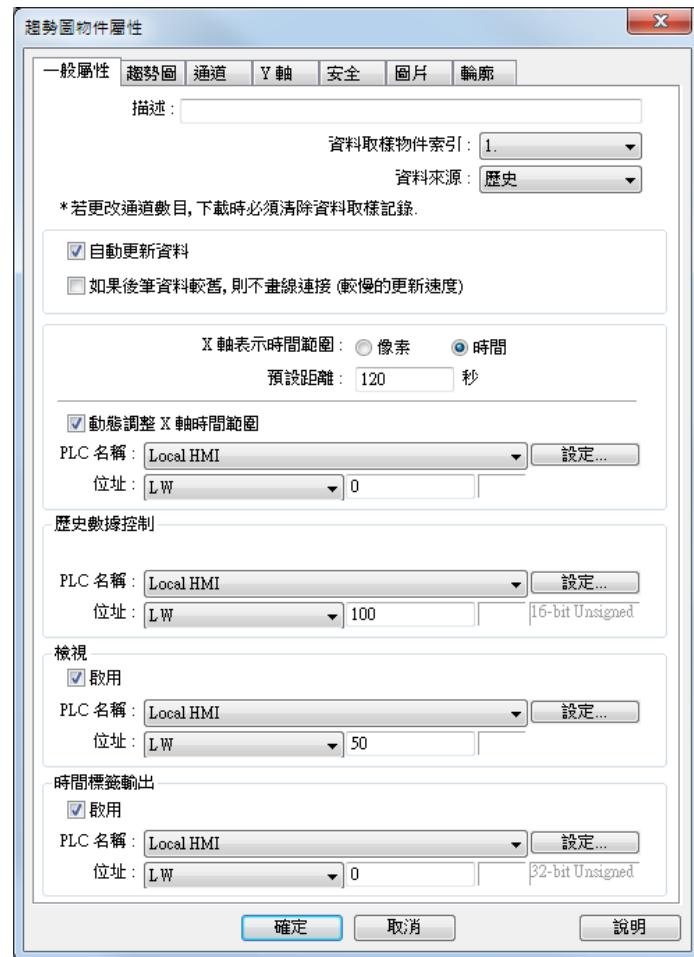
按下工具列的 [資料/歷史] » [趨勢圖] 按鈕後即會出現 [趨勢圖] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [趨勢圖] 物件。

#### 13.17.2.1. eMT、iE、XE、mTV 系列

##### 一般屬性設定



按鈕	描述
	顯示最初的取樣資料。
	顯示往前一個間隔的取樣資料。
	暫停畫面自動捲動功能。當新的取樣資料產生時，畫面不捲動，也不顯示超出畫面範圍的最新取樣資料。
	開啟畫面自動捲動功能。當新的取樣資料產生時，畫面會捲動來顯示最新的取樣資料。
	顯示往後一個間隔的取樣資料。
	顯示最新的取樣資料。



設定	描述
<b>資料取樣物件索引</b>	選擇 [資料取樣] 物件作為繪圖所需的數據來源。
<b>自動更新資料</b>	若勾選，於 [歷史] 模式下系統將每 10 秒自動更新所檢視的內容。反之則需透過視窗的刷新才會更新所檢視的內容。
<b>如果後筆資料較舊，則不畫線連接</b>	若勾選，當時間被往前調整且資料持續取樣，系統將不會於兩段資料間畫線連接。(將降低繪圖速度)
<b>資料來源</b>	選擇數據來源的形式，可以選擇 [即時] 或 [歷史]。
<b>即時</b>	可顯示來自 [資料取樣] 物件從人機開機後，固定筆數的取樣資料。取樣資料的顯示數量於 [資料取樣] 物件的 [最大資料 (即時模式)] 中設定。當超過此設定的數量，則較舊的資料會從畫面上刪除。若需顯示他日或較舊的資料，需使用 [歷史] 模式。 可以利用 [暫停控制] 功能暫停物件畫面更新的動作，但僅指暫停畫面刷新，並不會暫停 [資料取樣] 物件的取樣動作。
<b>歷史</b>	歷史記錄來自 [資料取樣] 物件使用日期來分類並儲存的取樣資料。

使用 [歷史] 模式可以利用 [資料取樣物件索引] 選定要顯示的歷史記錄，並利用 [歷史數據控制] 查看不同日期的歷史記錄。

HMI 會將取樣資料的歷史記錄檔案依時間先後排序，以日期最新的檔案為記錄 0 (一般是今日已存檔的取樣資料)，日期次新的檔案為記錄 1，其餘記錄依此類推。

在 [歷史控制] 中所指定暫存器中的數據如果為 0，[趨勢圖] 物件將顯示記錄 0 的數據；暫存器中的數據如果為 1，將顯示記錄 1 的數據，也就是說暫存器中的數據如果為 n，將顯示記錄 n 的數據。

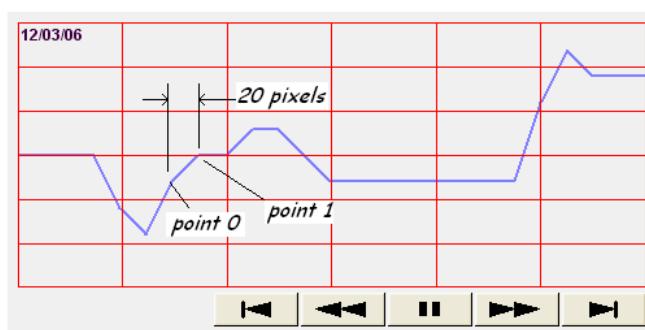
範例：若歷史控制暫存器為 LW-0，假使目前的 [資料取樣] 物件已儲存的取樣資料檔案依時間先後分別為 pressure\_20061120.dtl、pressure\_20061123.dtl、pressure\_20061127.dtl、pressure\_20061203.dtl，共 4 筆檔案，並且今日時間為 2006/12/3，則依照 LW-0 中的數據內容，[趨勢圖] 所顯示的取樣資料檔案整理如下：

LW-0 之數值	顯示的歷史資料取樣檔案
0	pressure_20061203.dtl
1	pressure_20061127.dtl
2	pressure_20061123.dtl
3	pressure_20061120.dtl

可搭配 [項目選單] 物件，資料來源選擇 [歷史數據日期]，則所有的歷史資料會依照日期分類並顯示於項目選單物件上。詳細可參考手冊《13.29 項目選單》。

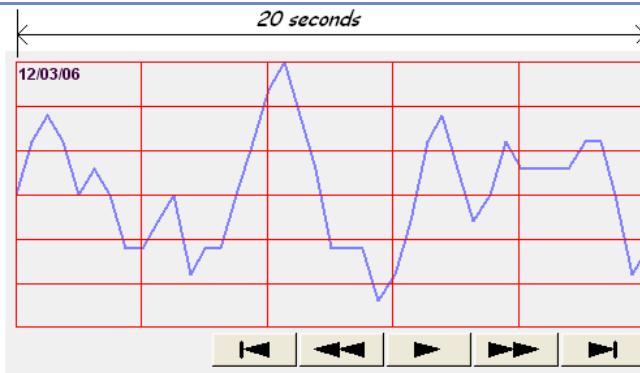
## 像素

設定兩取樣繪點間的距離，如下所示。



## 時間

設定 X 軸表示的時間範圍，如下所示。



可在 [趨勢圖] 頁面的 [網格] 項目啟用 [時間刻度] 功能。

#### 動態調整兩取

#### 樣繪點間距 /

#### 動態調整 X 軸

#### 時間範圍

指定一個 32-bit 格式的字元暫存器來線上調整 [像素] 或 [時間] 的距離。啟用此功能後，若暫存器內無輸入任何數值，則距離會採用預設值。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 自動更新資料

若啟用，則每次開啟 [歷史模式] 的 [趨勢圖] 物件所在的視窗時，物件畫面將會每 10 秒自動更新。請注意：

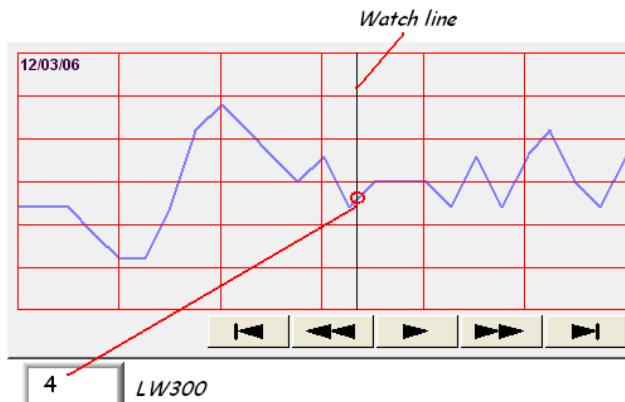
- 自動更新功能的狀態可由畫面捲動控制按鈕查看：  
當圖示為  表示趨勢圖物件的資料會自動更新。  
當圖示為  表示趨勢圖物件的資料停止更新。
- 當往前捲動查看較舊的資料時，會取消 [自動更新資料] 功能。  
此時控制按鈕的圖示為 。
- 當勾選 [自動更新資料] 時，無論先前是否曾藉由畫面捲動控制按鈕啟用或停用自動更新，切換回此視窗時，必定會自動更新畫面。  
範例：假設物件已啟用 [自動更新資料]，則往前捲動查看舊資料會停止自動更新功能。此時若切換至他頁再換回，則物件畫面仍會自動更新。
- 若製作工程檔案時，未啟用 [自動更新資料] 功能，而之後在人機上欲啟用時，只要按  按鈕即可。請注意此時的自動更新功能在切換視窗後就會被停用，也就是說，若切換至他頁再換回，則物件畫面仍停止更新。

#### 暫停控制

當暫停位元暫存器設為 ON 時，將暫停趨勢圖畫面刷新，但不會暫停資料取樣物件的取樣動作。當 [資料來源] 使用 [即時] 模式時才會顯示此選項。

#### 檢視

啟用後，使用者碰觸 [趨勢圖] 物件會產生一垂直檢視線，並將檢視線上的取樣數據輸出到指定的位址，再顯示於 [數值] 物件，如下所示。



[檢視] 功能也可以輸出多個通道的取樣數據，系統會依照 [資料取樣] 物件中所定義的取樣資料數據格式，依序將標記所在位置的取樣數據，從 [檢視] 功能所定義的起始位址依序寫入。例如 [資料取樣] 物件的取樣資料包含數個數據，格式皆不同，假設此時 LW-300 為 [檢視] 功能所定義的暫存器，則檢視線所標記的取樣資料的輸出位置如下所示。

暫存器	通道	資料格式
LW-300	0	16-bit Unsigned (1 word)
LW-301	1	32-bit Unsigned (2 words)
LW-303	2	32-bit float (2 words)
LW-305	3	16-bit Signed (1 word)

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 時間標籤輸出

若啟用，系統將會以第一個取樣點的取樣時間作為時間原點並開始計數，並將最新取樣點之累計秒數輸出至 [時間標籤輸出位址 + 2]。當點選物件上的曲線時，可將觸碰處最接近的取樣點之累計秒數輸出至 [時間標籤輸出位址]。

若觸發資料取樣物件的 [清除即時資料位址]，除了可以清除目前的取樣數據，也可以重置取樣時間原點。

注意: [時間標籤輸出位址] 與 [時間標籤輸出位址 + 2] 皆須為 32-bit 格式。[時間標籤輸出位址 + 2] 只適用於即時模式，而 [時間標籤輸出位址] 適用於即時模式及歷史模式。

當趨勢圖頁籤的 [相對時間模式] 被勾選時，才可啟用此功能。

## 趨勢圖設定



### 設定

### 描述

**透明 / 外框 /  
背景**

設定物件的外框與背景顏色。

**使用畫面捲動  
控制按鈕**

啟用 / 取消 畫面捲動控制按鈕，如下所示。



### 網格

設定格線的數目與顏色。會根據 [一般屬性] 設定頁的 [兩取樣繪點間的距離] 或是 [X 軸表示時間範圍] 設定不同而有差異。系統會利用這些設定，自動計算垂直格線的數目。

**X 軸-間隔** 設定格線垂直線的數目。

- 依照 [兩取樣繪點間的距離]：  
選擇每兩條垂直格線間所包含的取樣點數目。
- 依照 [X 軸表示時間範圍]：  
選擇每兩條垂直格線間所顯示的時間間隔。

**X 軸-等份** 設定格線垂直線的數目。

**Y 軸-等份** 設定格線水平線的數目。

**時間刻度**

選擇 [時間] 或 [日期] 來顯示時間刻度於趨勢圖的 X 軸，有不同的格式提供選擇，如下所示。

**字型 / 顏色 / 尺寸**

選擇時間刻度顯示文字之字型、字體顏色、字體尺寸。  
字體尺寸的預設值為 8。

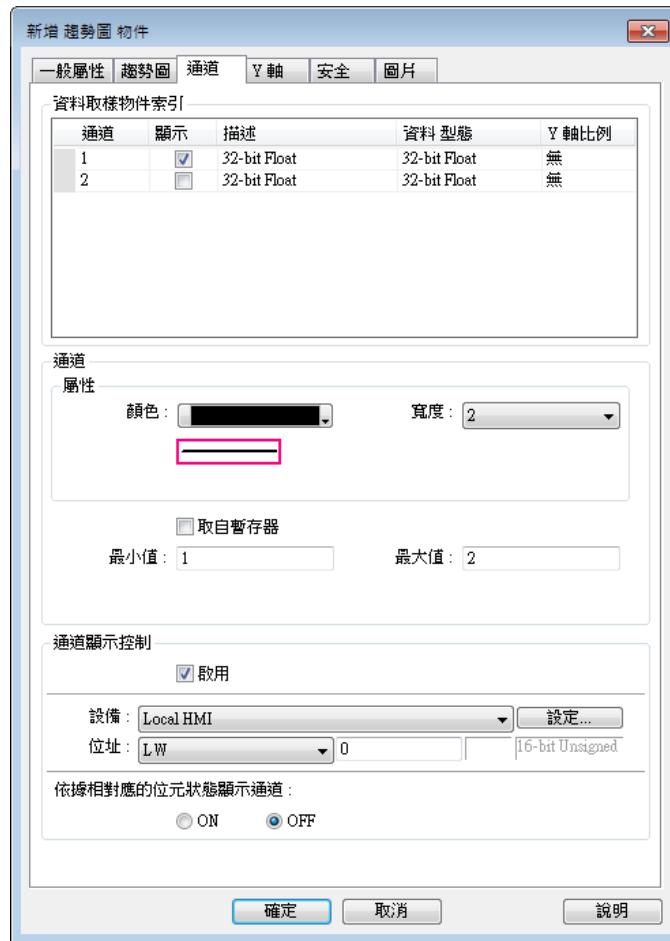
**時間 / 日期****相對時間模式**

勾選後 HMI 會將第一筆資料的時間記錄下來，搭配時間刻度的 SSSSS 使用，可表示相對的時間。此時時間刻度的日期無法使用。



最新的取樣資料的時間資訊將標示於物件的左上角，此項目用來設定時間 / 日期的顯示格式與顏色。

## 通道設定



### 設定

### 描述

#### Y 軸比例

顯示 Y 軸被設定為主軸或輔助軸。請見《Y 軸設定》說明。

通道	顯示	描述	資料型態	Y 軸比例
1	True	channel A	16-bit Unsigned	輔助軸
2	True	channel B	16-bit Unsigned	主軸
3	True	channel C	16-bit Unsigned	輔助軸

#### 通道

設定各條曲線的顏色，寬度，與樣式。

#### 取自暫存器

- 不勾選 [取自暫存器]

[最小值] 與 [最大值] 用來設定各曲線所描繪的取樣數據的最小值與最大值。也就是說如果存在某一曲線所描繪的取樣數據最小值為 50，最大值為 100，則 [最小值] 與 [最大值] 需設定為 50 與 100，如此所有的取樣數據才會完全被描繪在物件中。

- 勾選 [取自暫存器]

上下限可由指定暫存器設定。當寫入位址為 LW-n，則上/下限會根據以下的規則自動被設定為：

位址格式	16-bit	32-bit	64-bit
			(只支援於 cMT / cMT X)
下限	LW-n	LW-n	LW-n
上限	LW-n+1	LW-n+2	LW-n+4

以下表為例，當 [暫存器位址] 為 LW-100 時，則上/下限的位址會自動被設定為：

位址格式	16-bit	32-bit	64-bit
			(只支援於 cMT / cMT X)
下限	LW-100	LW-100	LW-100
上限	LW-101	LW-102	LW-104

此設定常用於放大與縮小趨勢圖 (不適用於 cMT / cMT X 系列)。請見以下範例 1。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 通道顯示控制

若勾選 [啟用]，則此位址中的各個位元將會被用來控制各個通道的顯示與否。Bit-1 控制通道 1，Bit-2 控制通道 2，依此類推。舉例來說，建立 5 個通道，並設定通道控制的位址為 LW-0，則各個通道會被以下位址控制：

通道	控制位址	位元狀態	是否顯示
1	LW_bit-000	OFF	YES
2	LW_bit-001	ON	NO
3	LW_bit-002	ON	NO
4	LW_bit-003	OFF	YES
5	LW_bit-004	OFF	YES

若通道選擇不被顯示，則被取消的通道不會佔用到通道控制位址。假設以上表為例：總共有 5 個通道，但第 3 個通道未勾選顯示於趨勢圖上，所以最多會有 4 個通道同時顯示於趨勢圖上，控制通道的位址只會使用 LW\_bit-000~003。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

 Note

- 趨勢圖最多可同時顯示 64 個通道。
- 趨勢圖最多只能顯示到通道編號 256，因此通道編號 257 後無提供挑選。

## Y 軸設定



設定	描述
Y 軸比例	可將 Y 軸設定為主軸或輔助軸，或是選擇 [無]，則不顯示 Y 軸。總共可以顯示 32 個 Y 軸。
刻度字型	設定 Y 軸刻度的字型、字型顏色、字型尺寸。
動態調整 Y 軸	可控制 Y 軸為顯示或不顯示。若控制位址為 LW-50，則控制通道 1 的位址為 LW_Bit 5000，控制通道 2 的位址為 LW_Bit 5001，依此類推。
動態調整主軸	可變更主軸。若在 LW-80 寫入 1，則主軸即為通道 1，若寫入 2，則主軸即為通道 2，依此類推。

 Note

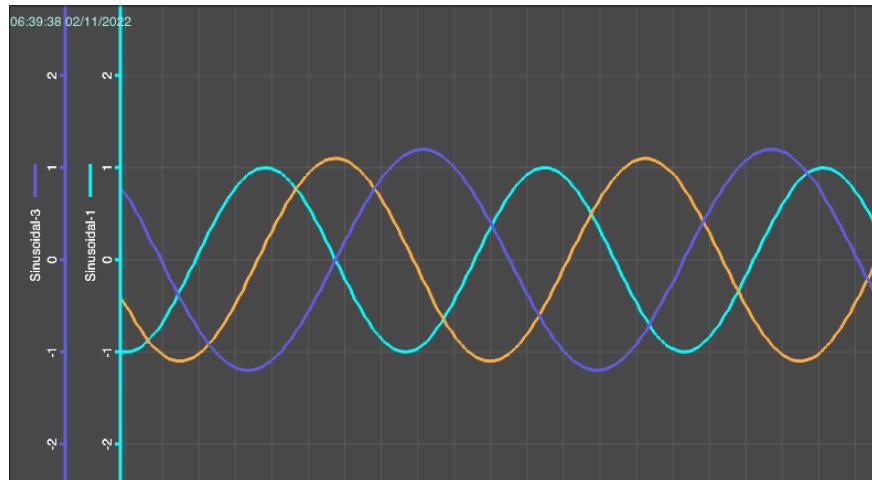
- eMT、iE、XE、mTV 系列的趨勢圖最大能顯示 6 MB 的資料。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

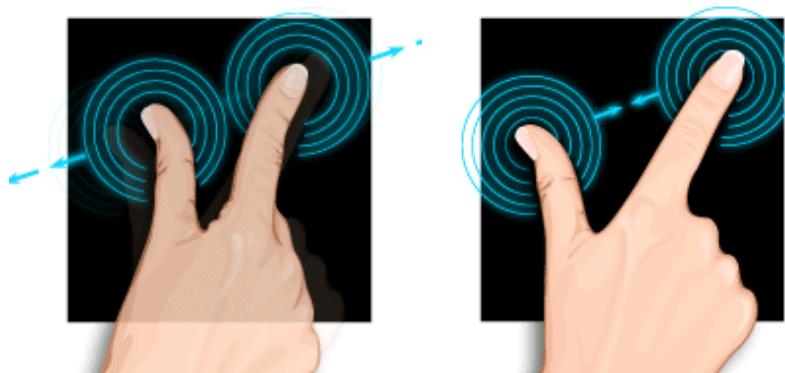
 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 13.17.2.2. cMT、cMT X 系列

#### 一般屬性設定



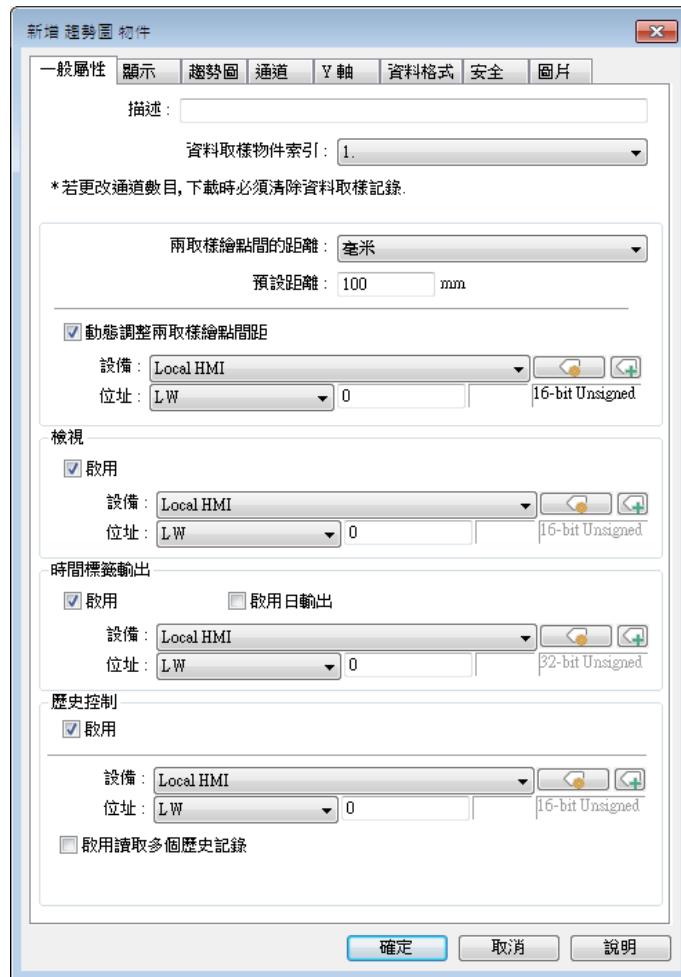
cMT / cMT X 系列的趨勢圖結合了 [即時] 模式和 [歷史] 模式，在物件上往左側拖曳即可查看歷史資料，往右側拖曳可查看最新的取樣資料，並可用雙指拖曳來放大或縮小趨勢圖。



放大趨勢圖

縮小趨勢圖

 cMT / cMT X 系列資料取樣儲存機制的詳細資訊請參考《8 資料取樣》。



設定	描述
<b>資料取樣物件索引</b>	選擇【資料取樣】物件作為繪圖所需的數據來源。
<b>毫米</b>	設定兩取樣繪點間的距離，上限為 500 mm。
<b>時間</b>	設定 X 軸表示的時間範圍。
<b>像素</b>	設定兩取樣繪點間的距離，上限為 1920 pixels。
<b>動態調整兩取樣繪點間距</b>	指定一個 32-bit 格式的字元暫存器來線上調整 [毫米]、[像素] 或 [時間] 的距離。啟用此功能後，若暫存器內無輸入任何數值，則距離會採用預設值。
<b>檢視</b>	啟用後，使用者碰觸【趨勢圖】物件會產生一垂直檢視線，並將檢視線上的取樣數據輸出到指定的位址，再顯示於【數值】物件。

[檢視] 功能也可以輸出多個通道的取樣數據，系統會依照 [資料取樣] 物件中所定義的取樣資料數據格式，依序將標記所在位置的取樣數據，從 [檢視] 功能所定義的起始位址依序寫入。例如 [資料取樣] 物件的取樣資料包含數個數據，格式皆不同，假設此時 LW-300 為 [檢視] 功能所定義的暫存器，則檢視線所標記的取樣資料的輸出位

置：

暫存器	通道	資料格式
LW-300	0	16-bit Unsigned (1 word)
LW-301	1	32-bit Unsigned (2 words)
LW-303	2	32-bit float (2 words)
LW-305	3	16-bit Signed (1 word)

#### 時間標籤輸出

##### 啟用

從第一次取樣資料點的時間開始計算，單位為秒。

[時間標籤輸出位址] 為目前選到的取樣資料點的秒數。

[時間標籤輸出位址 + 2] 為最新取樣資料點的秒數。

##### 啟用日輸出

從 1970/1/1 開始計算，單位為天。

[時間標籤輸出位址 + 4] 為目前選到的取樣資料點的天數。

[時間標籤輸出位址 + 6] 為最新取樣資料點的天數。

#### 歷史控制

歷史控制功能允許使用者根據特定控制位址來決定顯示的資料。這樣可以靈活地管理和顯示不同時間段的歷史數據。以下是詳細的設定步驟和範例說明。

歷史數據的儲存方式有兩種：全部記錄於同一檔案和自訂檔案管理。

##### 全部記錄於同一檔案

當資料取樣的儲存方式為全部記錄於同一檔案時，[歷史控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。

- 啟用讀取多個歷史記錄 - 天數

[歷史控制位址+1] 為總共顯示的天數，計算方式為從起始日索引往前計算。

- 啟用讀取多個歷史記錄 - 指定天數

[歷史控制位址+1] 為結束日索引，數值需設定比起始日索引大。若起始日索引為 4，結束日索引為 7，則代表要顯示 4 天前到 7 天前的資料。

##### 自訂檔案管理

當資料取樣的儲存方式為自訂檔案管理時，檔案會依照自訂檔案管理的排序設定決定檔案新舊，使用者可以透過歷史控制位址來控制顯示的數據。在 [歷史控制] 中所指定的暫存器中的數據如果為 0，則顯示最新的數據；如果為 1，則顯示次新的數據，依此類推。

## 顯示設定



### 設定

### 描述

#### 選項按鈕顯示

設定是否顯示物件上的選項按鈕 .

● 預定義：

預先設定是否顯示選項按鈕。

● 動態：

設定一個位元位址動態決定是否顯示選項按鈕。

預定義

動態

設備 :	Local HMI			
位址 :	LB		0	

#### 顯示選項 “通道顯示”

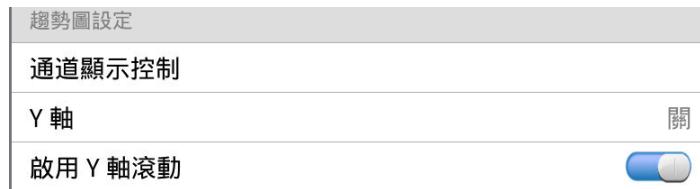
若勾選，則會顯示“通道顯示控制”選項。

#### 顯示選項 “Y 軸刻度”

若勾選，則會顯示“Y 軸”選項。

顯示選項 “啟用 Y 軸滾動”。

若勾選，則會顯示 “啟用 Y 軸滾動” 選項。



#### 縮放按鈕顯示

設定是否隱藏物件上的縮放按鈕 。

#### 顯示選項

#### 顯示自訂檔案名稱

若資料取樣使用 [自訂檔案管理]，啟用 [顯示自訂檔案名稱] 後，趨勢圖左上角將會顯示自訂的檔案名稱。

#### 連接資料缺口

在 PLC 斷線又重新連接上的情況下，斷線期間未取樣的時間區段，會繪一直線讓斷線前最後一筆資料與重連後的第一筆資料連接起來，不會有空白狀況。

#### 觸碰

#### 取消透過上滑/下滑滾動 Y 軸

啟用後，手指拖曳縮放趨勢圖物件時，Y 軸會一直保持在物件中間。

#### 取消 X 軸捲動

啟用後，手指拖曳縮放趨勢圖物件時，只會改變 Y 軸範圍。

#### 取消 Y 軸捲動

啟用後，手指拖曳縮放趨勢圖物件時，只會改變 X 軸範圍。

## 趨勢圖設定

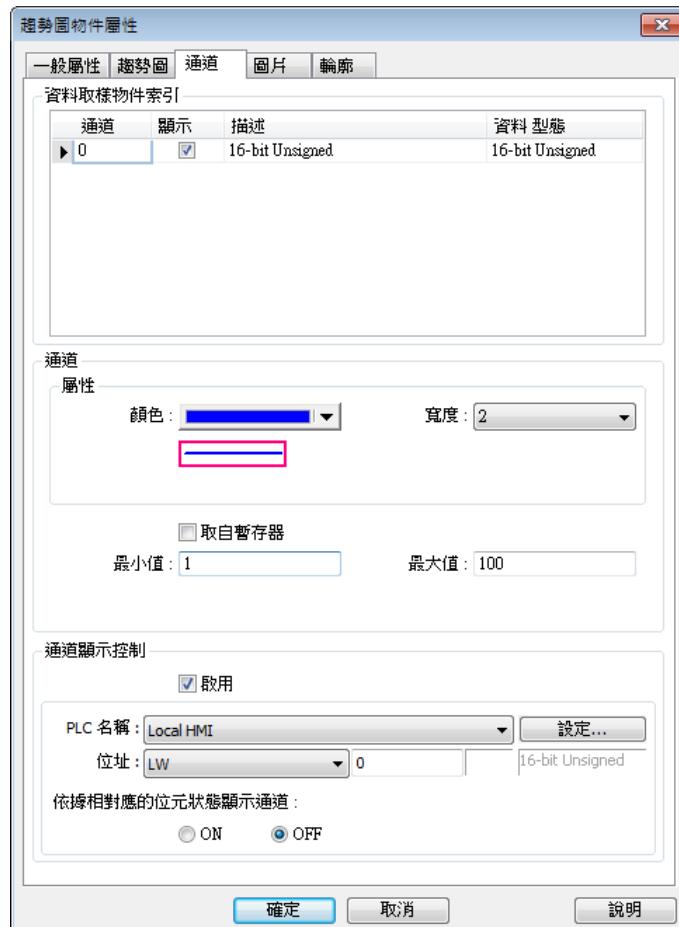


設定	描述
透明 / 外框 / 背景	設定物件的外框與背景顏色。
網格	設定格線的數目與顏色。會根據 [一般屬性] 設定頁的 [兩取樣繪點間的距離] 或是 [X 軸表示時間範圍] 設定不同而有差異。系統會利用這些設定，自動計算垂直格線的數目。 <b>X 軸-間隔</b> 設定格線垂直線的數目。 <ul style="list-style-type: none"><li>● 依照 [兩取樣繪點間的距離]： 選擇每兩條垂直格線間所包含的取樣點數目。</li><li>● 依照 [X 軸表示時間範圍]： 選擇每兩條垂直格線間所顯示的時間間隔。</li></ul> <b>X 軸-等份</b> 設定格線垂直線的數目。 <b>Y 軸-等份</b> 設定格線水平線的數目。
時間刻度	選擇 [時間] 或 [日期] 來顯示時間刻度於趨勢圖的 X 軸，有不同的格式提供選擇，另可選擇時間刻度擺放的位置。 字體尺寸的預設值為 8。

**時間 / 日期****相對時間模式**

勾選後 HMI 會將第一筆資料的時間記錄下來，搭配時間刻度的 SSSSS 使用，可表示相對的時間。此時時間刻度的日期無法使用。

最新的取樣資料的時間資訊將標示於物件的左上角，此項目用來設定時間 / 日期的顯示格式與顏色。

**通道設定**

最大值。也就是說如果存在某一曲線所描繪的取樣數據最小值為 50，最大值為 100，則 [最小值] 與 [最大值] 需設定為 50 與 100，如此所有的取樣數據才會完全被描繪在物件中。

● 勾選 [取自暫存器]

上下限可由指定暫存器設定。當寫入位址為 LW-n，則上/下限會根據以下的規則自動被設定為：

位址格式	16-bit	32-bit	64-bit
(只支援於 cMT / cMT X)			
下限	LW-n	LW-n	LW-n
上限	LW-n+1	LW-n+2	LW-n+4
以下表為例，當 [暫存器位址] 為 LW-100 時，則上/下限的位址會自動被設定為：			
位址格式	16-bit	32-bit	64-bit
(只支援於 cMT / cMT X)			
下限	LW-100	LW-100	LW-100
上限	LW-101	LW-102	LW-104

#### 通道顯示控制

若勾選 [啟用]，則此位址中的各個位元將會被用來控制各個通道的顯示與否。Bit-1 控制通道 1，Bit-2 控制通道 2，依此類推。舉例來說，建立 5 個通道，並設定通道控制的位址為 LW-0，則各個通道會被以下位址控制：

通道	控制位址	位元狀態	是否顯示
1	LW_bit-000	OFF	YES
2	LW_bit-001	ON	NO
3	LW_bit-002	ON	NO
4	LW_bit-003	OFF	YES
5	LW_bit-004	OFF	YES

若通道選擇不被顯示，則被取消的通道不會佔用到通道控制位址。假設以上表為例：總共有 5 個通道，但第 3 個通道未勾選顯示於趨勢圖上，所以最多會有 4 個通道同時顯示於趨勢圖上，控制通道的位址只會使用 LW\_bit-000~003。

#### Note

- 趨勢圖最多可同時顯示 64 個通道。
- 趨勢圖最多只能顯示到通道編號 256，因此通道編號 257 後無提供挑選。

## Y 軸設定



物件上的 Y 軸會顯示該通道的通道刻度。當 [趨勢圖] 設定頁的網格選擇 [顯示] 時，才可使用此功能。

設定	描述
Y 軸	<p><b>通道資料不顯示時隱藏通道 Y 刻度</b> 該設定由使用者決定 Y 軸是否與通道顯示控制連動顯示或隱藏，預設為開啟。</p> <p><b>數值對齊</b> 該設定決定 Y 軸數值顯示的對齊方式。</p> <p>以刻度為中心                      完整顯示</p>

Y 軸比例也可在 cMT Viewer 中調整，請參考以下步驟。

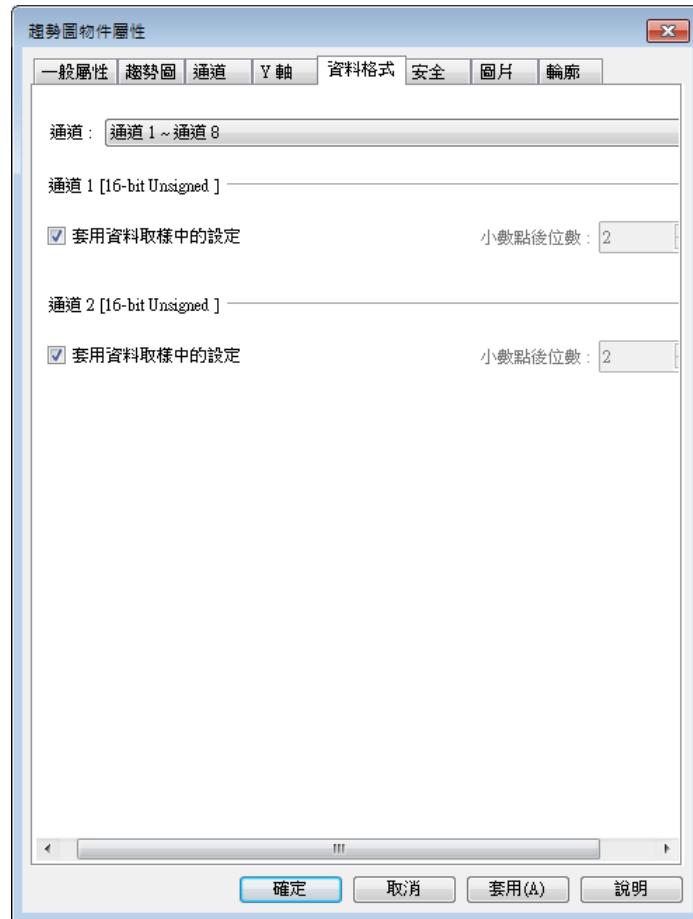
- 點選 [趨勢圖] 物件右上方的 按鈕。
- 點選 [趨勢圖設定] 下的 Y 軸。



3. 選擇要顯示 Y 軸的通道。



## 資料格式設定



設定	描述
套用資料取樣 中的設定	該通道的小數點後位數直接套用資料取樣中的設定。
小數點後位數	表示該通道會將數值轉換後以新的數值表示。 例如：使用小數點後位數 1，資料取樣得到的原始數值為 45，在趨勢圖上將以 4.5 為新的數值顯示。

## 範例 1

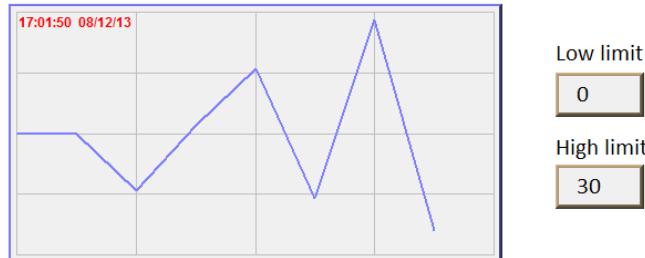
此範例說明如何縮放趨勢圖。此範例不適用於 cMT / cMT X 系列。

使用者需在通道的上限/下限設定欄位勾選 **【取自暫存器】** 以實現此功能。當最大/最小值取自暫存器設定為 LW-n，則 LW-n 控制最小值，LW-n+1 控制最大值。



設定最大/最小值取自暫存器 LW-0，建立兩個【數值】物件控制最小值及最大值，位址分別為 LW-0 和 LW-1。

當有一組數據的大小皆介於 0 至 30，則在控制最小值位址輸入 0，控制最大值位址輸入 30，趨勢圖呈現如下所示。



若要縮小趨勢圖，則可以在最大值輸入較大的數據。例如：在控制最小值位址輸入 0，控制最大值位址輸入 100，趨勢圖呈現如下所示。



若要放大趨勢圖，則可以在最大值輸入較小的數據。例如：在控制最小值位址輸入 0，控制最大值位址輸入 20，趨勢圖呈現如下所示。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.18. 歷史數據顯示

### 13.18.1. 概要

【歷史數據顯示】物件用來顯示已經儲存的資料取樣數據，跟趨勢圖不同的是【歷史數據顯示】物件使用表列的方式直接顯示這些數據的內容。歷史數據範例表，如下圖所示。

編號	時間	日期	Ch.0	Ch.1	Ch.2
3577	21:52	16/09/07	0	0	0
3576	21:52	16/09/07	0	0	0
3575	21:52	16/09/07	0	0	0
3574	21:52	16/09/07	0	0	0
3573	21:52	16/09/07	0	0	0
3572	21:52	16/09/07	0	0	0
3571	21:52	16/09/07	0	0	0
3570	21:52	16/09/07	0	0	0
3569	21:52	16/09/07	0	0	0
3568	21:52	16/09/07	0	0	0

eMT/iE/XE/mTV 最大可顯示 6MB 資料，超過部分 HMI 不會顯示。

### 13.18.2. 設定



按下工作列上的【資料/歷史】»【歷史數據顯示】按鈕後即會出現【歷史數據顯示】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【歷史數據顯示】物件。

## 一般屬性設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



### 設定

### 描述

#### 資料取樣物件索引

選擇【資料取樣】物件作為所需的數據來源。

#### 自動更新資料

若勾選，系統將每十秒鐘自動更新所檢視的內容。反之則需要透過視窗的刷新才會更新所檢視的內容。

#### 樣式

歷史數據顯示物件的顯示樣式。

#### 網格

選擇物件是否使用格線區分每個欄位。

#### 顏色

設定格線所使用的顏色。

#### 欄位距離

此項設定值用來調整各欄位間的距離，下圖為使用不同【欄位距離】設定時的顯示情形。

No.	Time	Date	Ch.0	Ch.1	Ch.2
3667	21:57	16/09/07	1	0	0
3666	21:57	16/09/07	1	0	0
3665	21:57	16/09/07	1	0	0
3664	21:57	16/09/07	1	0	0
3663	21:57	16/09/07	1	0	0
3662	21:57	16/09/07	1	0	0
3661	21:57	16/09/07	1	0	0
3660	21:56	16/09/07	0	0	0
3659	21:56	16/09/07	0	0	0
3658	21:56	16/09/07	0	0	0

No.	Time	Date
3667	21:57	16/09/07
3666	21:57	16/09/07
3665	21:57	16/09/07
3664	21:57	16/09/07
3663	21:57	16/09/07
3662	21:57	16/09/07
3661	21:57	16/09/07
3660	21:56	16/09/07
3659	21:56	16/09/07
3658	21:56	16/09/07

外觀	設定物件的外框與背景顏色，若勾選 [透明] 表示不使用外框與背景顏色。
文字	設定欲顯示的字型與尺寸。
時間 / 日期	<b>按時間順序</b> 資料將由舊到新依序顯示，最近取樣的資料顯示於物件底部。 <b>按時間逆序</b> 資料將由舊到新依序顯示，最近取樣的資料顯示於物件頂部。
歷史數據控制 (eMT、iE、 XE、mTV 系 列)	系統會將取樣資料的歷史記錄檔案依時間先後排序，日期最新的檔案為記錄 0 (一般是今日已存檔的取樣資料)，日期次新的檔案為記錄 1，其餘記錄依此類推。[歷史數據控制] 項目則用來指定要顯示的記錄。
檢視 (cMT、cMT X 系列)	在歷史數據物件上點擊一筆資料，會將該筆資料內容輸出至指定的位址。
歷史數據控制 (cMT、cMT X 系列)	歷史控制功能允許使用者根據特定控制位址來決定顯示的資料。這樣可以靈活地管理和顯示不同時間段的歷史數據。以下是詳細的設定步驟和範例說明。 歷史數據的儲存方式有兩種：全部記錄於同一檔案和自訂檔案管理。
<b>全部記錄於同一檔案</b>	
當資料取樣的儲存方式為全部記錄於同一檔案時，[歷史控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。	
<ul style="list-style-type: none"><li>● 啟用讀取多個歷史記錄 - 天數</li></ul>	
[歷史控制位址+1] 為總共顯示的天數，計算方式為從起始日索引往前計算。	
<ul style="list-style-type: none"><li>● 啟用讀取多個歷史記錄 - 指定天數</li></ul>	
[歷史控制位址+1] 為結束日索引，數值需設定比起始日索引大。若起始日索引為 4，結束日索引為 7，則代表要顯示 4 天前到 7 天前的資料。	
<b>自訂檔案管理</b>	
當資料取樣的儲存方式為自訂檔案管理時，檔案會依照自訂檔案管理的排序設定決定檔案新舊，使用者可以透過歷史控制位址來控制顯示的數據。在 [歷史控制] 中所指定的暫存器中的數據如果為 0，則顯示最新的數據；如果為 1，則顯示次新的數據，依此類推。	

 Note

- 使用 cMT / cMT X 系列時，請直接於各裝置的 cMT Viewer 畫面上點選歷史數據顯示物件右上角的齒輪圖示來指定日期並顯示數據。



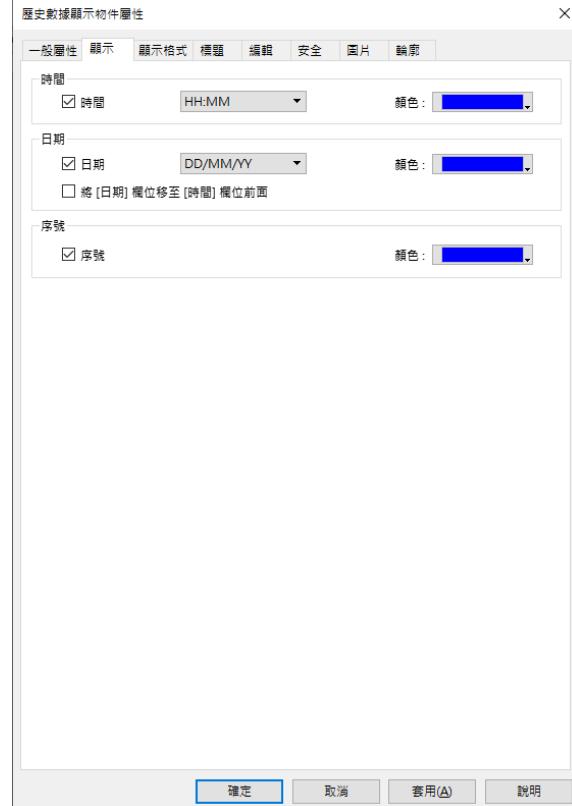
 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 顯示設定

### cMT / cMT X 系列



### eMT、iE、XE、mTV 系列



設定	描述
時間 / 日期	用來選擇是否顯示資料的取樣時間與日期，並決定時間與日期的顯示顏色與格式。 <b>將 [日期] 欄位移至 [時間] 欄位前面</b>

若勾選，此兩欄位的顯示順序將交換。

**序號** 若勾選，可顯示資料的編號欄位。

**行列** **多行顯示**

若勾選，可設定資料顯示的行數。設定後無論歷史數據的內容有幾行，將一律顯示相同的設定行數。

#### 選項按鈕顯示

##### (cMT / cMT X

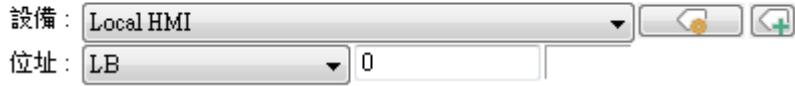
##### 系列)

設定是否顯示物件上的選項按鈕 。

- 預定義：

預先設定是否顯示選項按鈕。

- 動態：



設定一個位元位址動態決定是否顯示選項按鈕。

## 數據顯示格式設定

### cMT / cMT X 系列



### eMT、iE、XE、mTV 系列



設定	描述
通道	最多可同時顯示 64 個通道。由此設定頁可得知目前選取的【資料取樣】物件一次取樣的數據長度、各通道的數據格式、顯示於此物件上的通道編號。如上圖：【資料取樣】物件執行一次將存取讀取 4 個數據 (通道 1~通道 4)，各通道的數據格式皆為 16-bit Unsigned。由於只勾選通道 1 及通道 4，歷史數據顯示物件顯示的資料方式如下圖所示。

編號	日期	時間	ch.1	ch.4
16	09/03/17	14:03	0	0
15	09/03/17	14:02	0	0
14	09/03/17	14:02	0	0
13	09/03/17	14:02	0	0
12	09/03/17	14:02	0	0
11	09/03/17	14:02	0	0
10	09/03/17	14:02	0	0
9	09/03/17	14:02	0	0
8	09/03/17	14:02	0	0

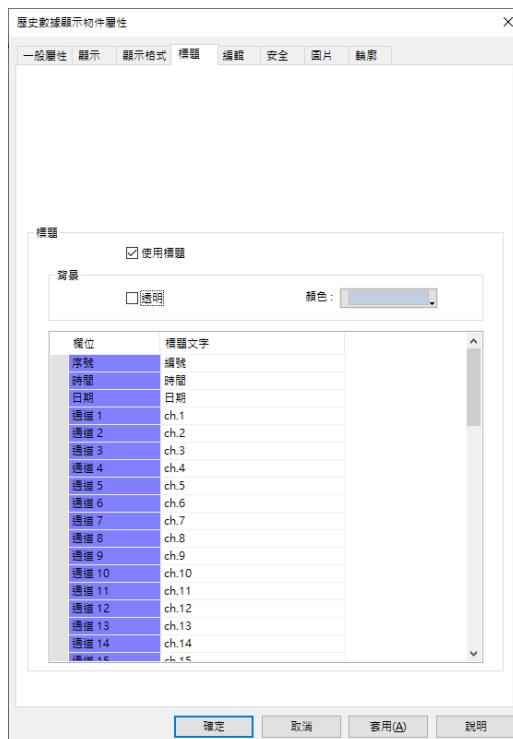
cMT / cMT X 系列的數據格式可以直接選擇套用資料取樣中的設定。

當使用歷史數據顯示物件顯示【字串】格式時，可以選擇：



- 使用不同的字串格式顯示。
- 將數據的高位元組與低位元組資料互換後，再加以顯示。

## 標題設定



### 設定

### 描述

#### 使用標題

選擇是否使用標題。

編號	日期	時間	ch.1	ch.4
1	09/03/17	14:06	#####	##

### 背景

### 透明

勾選 [透明] 表示不使用標題文字的背景色。

### 顏色

設定標題文字的背景色。

### 設定

### 文字

標題的文字來源也可以取自文字標籤庫。需先定義文字標籤庫後，於相對應的標題名稱勾選 [文字標籤庫]，並選擇文字標籤即可。

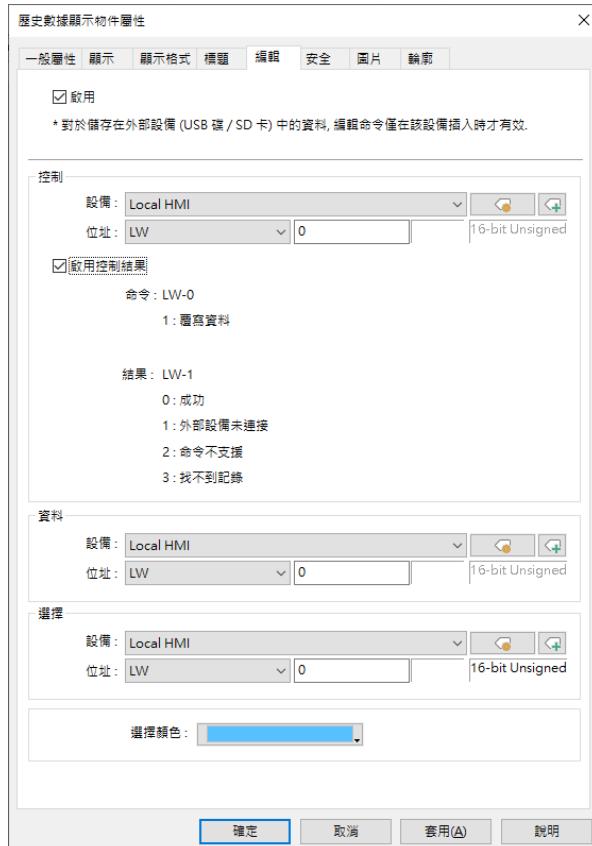
標題名稱	文字標籤庫	文字標籤	標題
▶ 序號	<input checked="" type="checkbox"/>	Label_1	No.
時間	<input type="checkbox"/>		時間
日期	<input type="checkbox"/>		日期
通道 1	<input type="checkbox"/>		ch.1
通道 2	<input type="checkbox"/>		ch.2
通道 3	<input type="checkbox"/>		ch.3



- 當於 PC 執行過模擬後，若相同工程檔案欲改變資料取樣內的資料格式並再次執行模擬，請先將 EasyBuilder 安裝資料夾內的 HMI\_memory、SD\_card、usb1 資料夾的舊資料取樣記錄檔案刪除，避免系統誤讀舊檔案。

## 編輯設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



### 設定

### 描述

#### 控制位址

LW-n: 命令

數值	描述
1	將 [資料位址] 內的數據覆寫至 資料取樣。
2	刪除所選擇的資料。

LW-n+1: 顯示控制結果 (cMT / cMT X 系列)

數值	描述
0	執行成功。
1	外部裝置未連接。
2	命令不支援。
3	找不到記錄。

#### 資料位址

當選取歷史數據顯示物件中的某一行數時，系統會將該行數的數據依

序讀出，此時即可修改資料取樣中的數據。

注意：此位址的編排格式需配合資料取樣中的資料格式設定。

#### 選擇位址

指示當前歷史數據顯示物件被選取的行數，修改此位址內的數值亦可移動選擇行數。

注意：

1. 當選擇位址內的數值為 0 時，將不會有行數被選擇，且 [資料位址] 內的數值將維持前一個選擇行數之數據。
2. 當選擇位址內的數值大於實際行數時，將選擇當前最大值之行數。



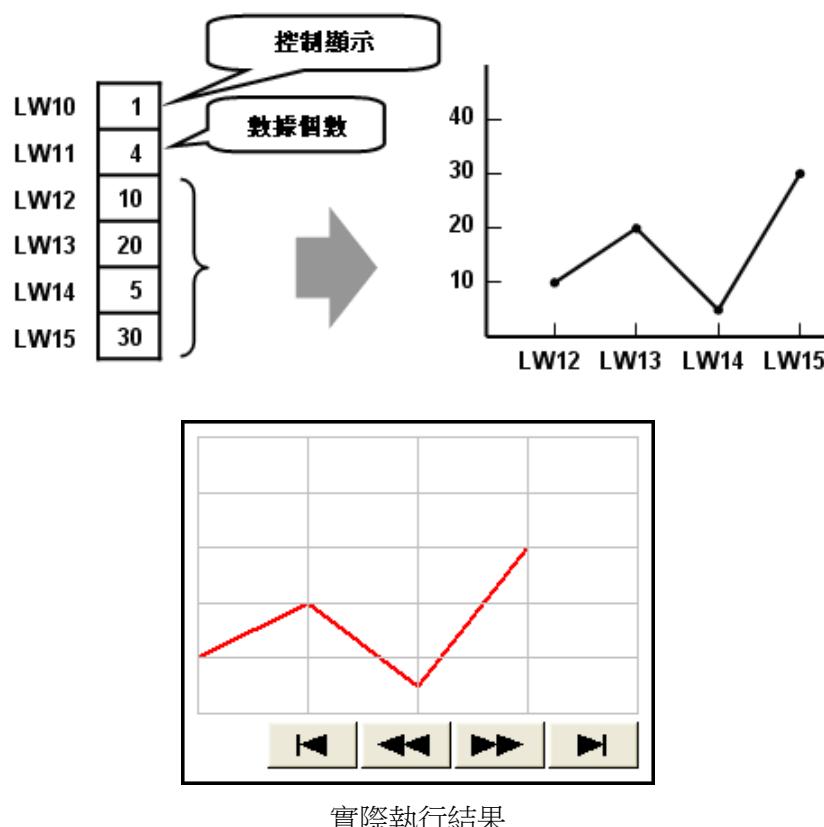
#### Note

- 當資料取樣的儲存位置為外部裝置時，若將外部裝置移除將無法正確的使用此功能，須等到外部裝置再次插入且系統讀取到歷史數據後，才可繼續動作。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.19. 數據群組顯示

### 13.19.1. 概要

一個數據群組(或區塊)是指一組連續位址中的數據，X 軸代表位址，Y 軸代表數據，如下圖為使用數據群組顯示物件顯示單一數據群組 LW-12~LW-15 中的數據。數據群組顯示物件亦可同時顯示多個數據群組的內容，例如同時顯示 LW-12~LW-15 與 RW-12~RW-15 兩個數據群組，使用者可藉由此方式來觀察及比較各暫存器中的資料。



### 13.19.2. 設定



按下工作列的【物件】»【曲線圖】»【數據群組顯示】按鈕後即會開啟【數據群組顯示】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【數據群組顯示】物件。

## 一般屬性設定



設定	描述
描述	使用者可為此物件描述相關訊息。
通道數目	設定物件的通道數目。每個通道表示一組群組數據。最多可同時支援 12 組。
檢視	<p>若啟用，可顯示垂直檢視線上的數據索引和數值，請見以下範例 1。</p> <p><b>啟用輸入功能 (cMT / cMT X 系列)</b></p> <p>可在 [檢視位址] 輸入數值來移動垂直檢視線。</p>
通道	選擇一個通道來設定控制位址的相關數據群組屬性。
控制位址	<p>選擇數據群組的控制位址及數據來源。</p> <p>當控制位址設定為 LW-n 時，輸入特定數值至 LW-n 來控制圖形的顯示及清除。當執行完命令後，系統會將控制位址重設為 0。</p> <p>輸入 “0”：無動作 (預設值)</p> <p>輸入 “1”：繪圖</p> <p>輸入 “2”：清除</p> <p>輸入 “3”：重新繪圖</p>

### 數據個數位址

當控制位址設定為 LW-n 時，寫入特定數值至 LW-n+1 做為存放群組的數據數量，最多支援 1024 個。

### 數據儲存起始位址

若啟用 [使用位址偏移]，[數據儲存偏移位址] 定義為 LW-n+2。

若選擇 16-bit 格式，每個起始數據的位址間隔為 1。

例如：起始位址 +1，起始位址 +2，等等。

若選擇 32-bit 格式，每個起始數據的位址間隔為 2。

例如：起始位址 +2，起始位址 +4，等等。

關於控制位址各項設定，請見以下範例 2~5。

### 限制

用來設定所顯示圖形之最大值與最小值。



- [數據群組顯示] 物件未清除繪圖 (一直對控制位址輸入數值 1) 的情況下，可重複繪圖的次數為 “32 次除以通道數” 。
 

例如：

通道數目為 1 的情況 → 透過輸入 1 到控制位址可重複繪圖 32 次  
 通道數目為 2 的情況 → 透過輸入 1 到控制位址可重複繪圖 16 次  
 通道數目為 16 的情況 → 透過輸入 1 到控制位址可重複繪圖 2 次

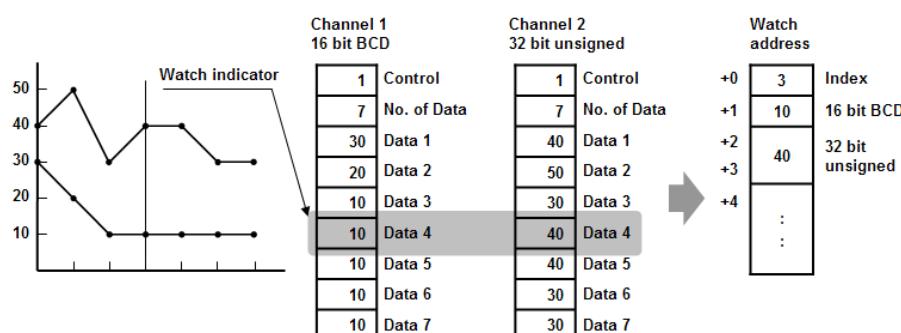
## 範例 1

### 數據檢視功能

啟用 [檢視] 功能，則當觸控此物件上任一點時，會顯示一條垂直的檢視線，曲線與該檢視線交會形成的點所相對應的數值將會寫入至指定的暫存器中。

資料格式	數據索引	通道 1 數值	通道 2 數值
<b>16-bit</b>	位址	位址 +1	位址 +2
<b>32-bit</b>	位址	位址 +2	位址 +4
<b>64-bit</b> (只支援於 cMT / cMT X)	位址	位址 +4	位址 +8

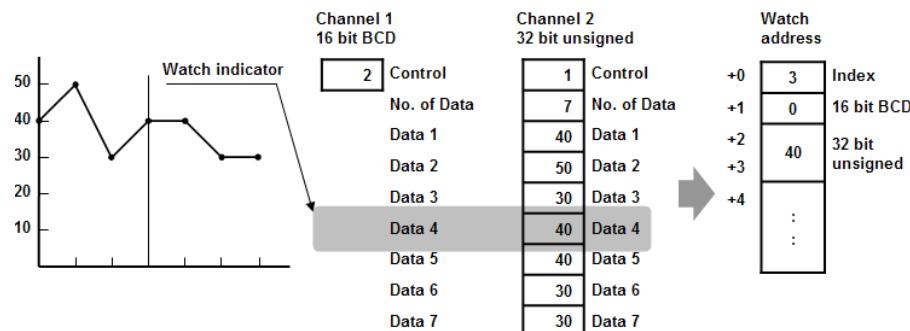
若設定檢視位址為 LW-n，則在 LW-n 寫入數值代表從各通道欲呼叫的索引編號 (從 0 計算)，如下圖：



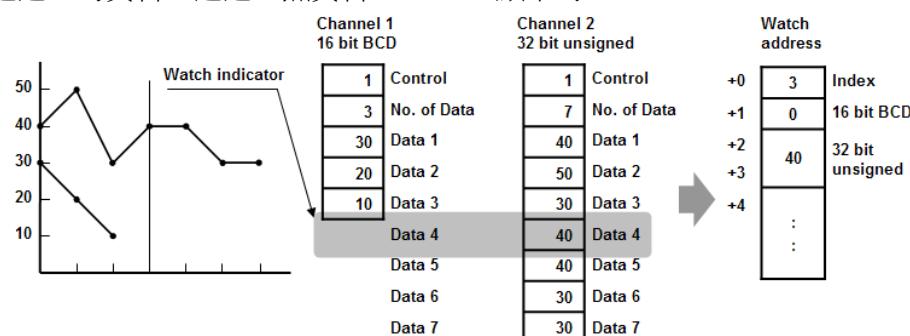
 Note

- 索引編號為 16 位元無號數整數。若指定之暫存器為 32 位元時，只有較低的 16 位元可作用，並忽略使用較高的 16 位元。
- 當檢視的通道無資料時，則會以 0 代替。

EX: 僅有通道二的資料，通道一無資料，LW-n+1 顯示為 0。



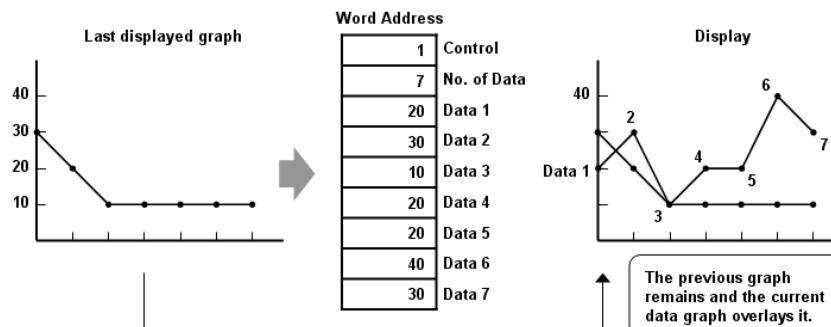
EX: 僅有通道二的資料，通道一無資料，LW-n+1 顯示為 0。



## 範例 2

### 如何顯示數據群組的內容

- 在 [數據個數位址] 輸入欲顯示的數據筆數，也就是“控制位址 +1”。
- 在 [數據儲存起始位址] 依序填入數據內容。
- 在 [控制位址] 輸入“1”；此時 HMI 將以折線圖畫出目前暫存器的內容 (並保留先前圖形)。
- HMI 在完成前項動作後將對[控制位址] 寫入“0”。


 Note

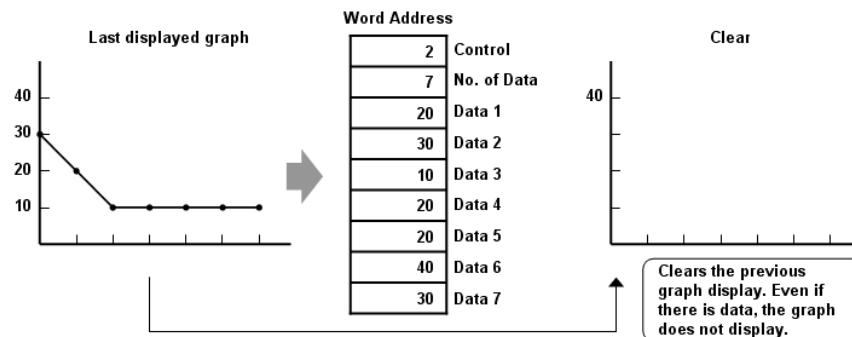
- 在上述動作 3 和 4 之間，請勿更改 [控制位址]、[數據個數位址] 及 [數據儲存起始位址] 內

容，否則可能產生非預期結果。

### 範例 3

#### 如何清除已顯示的圖形

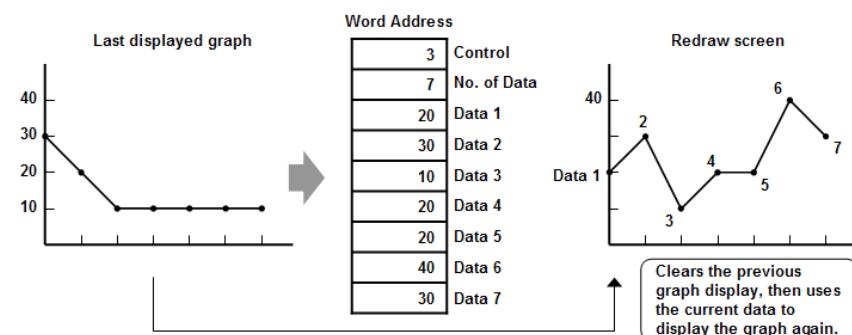
1. 在 [控制位址] 輸入“2”；將清除先前所畫之線圖。
2. HMI 在完成前項動作後將於 [控制位址] 寫入“0”。



### 範例 4

#### 清除已顯示的圖形並顯示新數據的圖形

1. 在 [數據個數位址] 輸入欲顯示的數據筆數，也就是“控制位址 +1”。
2. 在 [數據儲存起始位址] 依序填入數據內容。
3. 在 [控制位址] 輸入“3”；此時 HMI 會先將先前的折線圖清除，再畫出目前位址內的內容。
4. HMI 在完成前項動作後將於 [控制位址] 寫入“0”。



### 範例 5

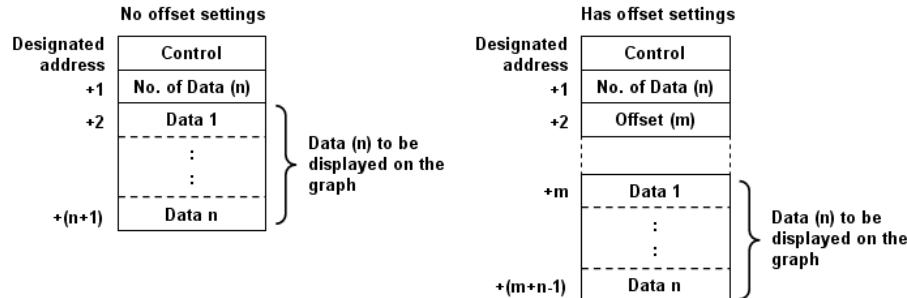
#### 位址偏移模式

啟用後，則各個通道的 [控制位址]、[數據個數位址]、[數據儲存偏移位址] 會使用連續的位址。例如有 3 個通道(通道 0 至通道 2)，且 [控制位址] 分別為 LW-0、LW-100 和 LW-200，則各個通道的 [控制位址]、[數據個數位址]、[數據儲存偏移位址] 如下：(下表使用 3 個通道數，格式皆為 16-bit Unsigned 且數據儲存偏移位址內的數值設為 m)

項目	通道 0	通道 1	通道 2
控制位址	LW-0	LW-100	LW-200

數據個數位址	LW-1	LW-101	LW-201
數據儲存偏移位址	LW-2 (=m)	LW-102 (=m)	LW-202 (=m)
資料 1	LW-0+m	LW-100+m	LW-200+m
資料 2	LW-1+m	LW-101+m	LW-201+m
...	...	...	...

下圖左側代表未使用【偏移模式】的讀取方式，右側則是使用位址偏移模式的讀取方式。

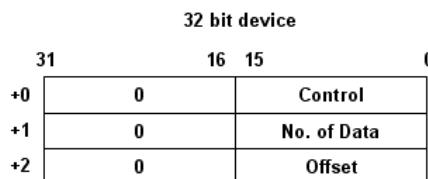


### Note

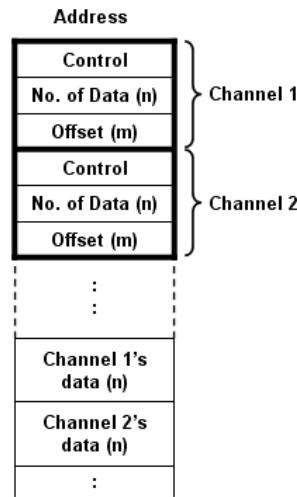
- 當【控制位址】設定為 LW-n 時，【數據個數位址】及【數據儲存偏移位址】會根據以下規則設定：

資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
控制位址	LW-n	LW-n	LW-n
數據個數位址	LW-n+1	LW-n+2	LW-n+4
數據儲存偏移位址	LW-n+2	LW-n+4	LW-n+8

- 當【控制位址】為 32 位元時，只有較低的 16 位元產生作用，請將較高的 16 位元內容設為 0。



- 系統會在【控制位址】的數據不為 0 時讀取【數據個數位址】及【數據儲存偏移位址】的內容。
- 當使用了兩個同類型暫存器以上的通道，則啟用【使用位址偏移】並使用連續的位址做控制位址可減少系統讀取數據的時間。如下圖。當使用 16 位元格式時，設定通道 1 的控制位址為 LW-n，通道 2 的控制位址為 LW-n+3，依此類推。



## 顯示區域設定



設定	描述
格線	<b>顯示點數</b> 設定圖形一頁所能顯示最大資料筆數。
	<b>捲動量</b> 左右捲動的資料筆數。

### 使用畫面捲動控制按鈕

按下 後，畫面將顯示往前或往後一個點的資料。

按下 後，畫面將顯示最初或最後的資料。

#### 外觀

物件的邊緣線顏色及背景顏色。

#### 透明

若勾選透明，則物件就不會有背景顏色，也不會出現【顏色】的選項。

#### 網格

物件上分隔水平及垂直區塊的格線。

#### 通道

設定各數據群組圖形之線條顏色、粗細及樣式。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.20. XY 曲線圖

### 13.20.1. 概要

[XY 曲線圖] 物件用來顯示二維座標的 XY 資料點，每個數據包含 X 值和 Y 值，皆從暫存器中讀取。

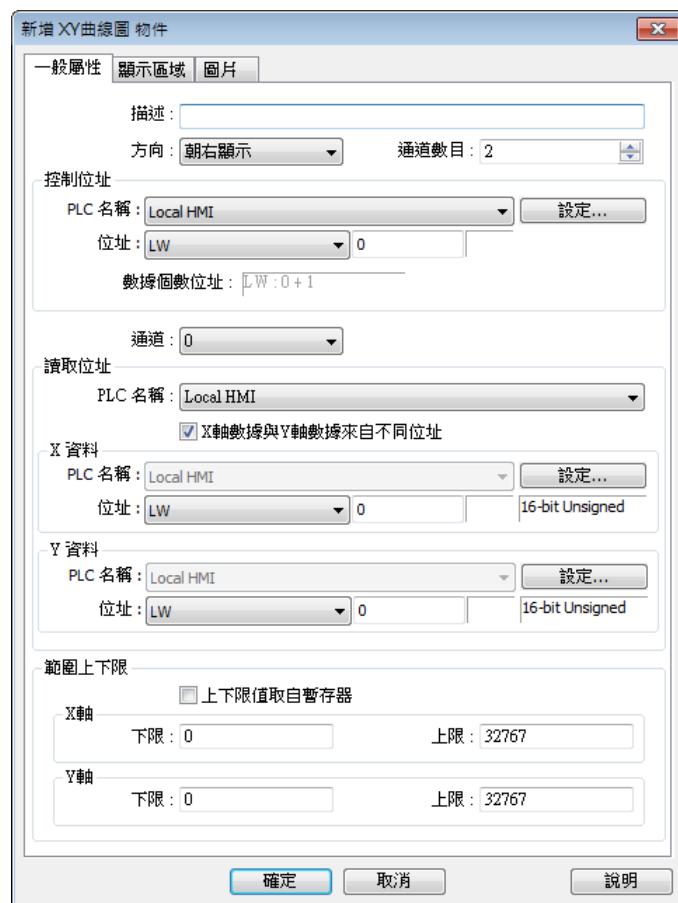
同時可顯示最多 16 組曲線。此功能可讓使用者觀察及分析各暫存器中的資料。負數亦可使用。

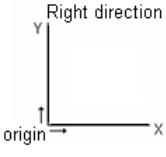
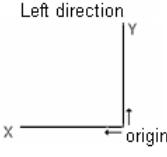
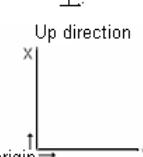
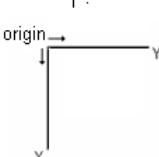
### 13.20.2. 設定



按下工作列的 [物件] » [曲線圖] » [XY 曲線圖] 按鈕，隨即出現物件屬性對話窗。

#### 一般屬性設定



設定	描述													
方向	XY 軸方向可選擇 [朝右]、[朝左]、[朝上] 或 [朝下] 顯示，如下圖：													
	右：  左：  上：  下： 													
通道數目	欲觀察的通道資料筆數。													
控制位址	用來控制 XY 曲線圖的顯示或清除，當控制位址設定為 LW-n，則對 LW-n 寫入不同的數值代表不同的命令。同時，LW-n+1 會被用來調整顯示的數據個數。當 HMI 完成指定的動作後會將 [控制位址] 之值設為 0。													
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>控制位址</th><th>數值</th><th>結果</th></tr> </thead> <tbody> <tr> <td rowspan="3">LW-n</td><td>1</td><td>顯示目前圖形 (保留已繪製的圖)</td></tr> <tr> <td>2</td><td>清除圖</td></tr> <tr> <td>3</td><td>清除所有圖形，並重新繪製圖形</td></tr> <tr> <td>LW-n+1</td><td>任意數</td><td>顯示的數據個數</td></tr> </tbody> </table>	控制位址	數值	結果	LW-n	1	顯示目前圖形 (保留已繪製的圖)	2	清除圖	3	清除所有圖形，並重新繪製圖形	LW-n+1	任意數	顯示的數據個數
控制位址	數值	結果												
LW-n	1	顯示目前圖形 (保留已繪製的圖)												
	2	清除圖												
	3	清除所有圖形，並重新繪製圖形												
LW-n+1	任意數	顯示的數據個數												
數據個數位址	此位址是用來儲存資料顯示的數量。每個通道可以有高達 1023 個 XY 數據。													
通道	指定一個通道並設定讀取的相關屬性。													
讀取位址	<p><b>設備</b> 選擇讀取的資料來源裝置。 存取暫存器資料時，須同時考慮是否啟用 <b>[X 軸數據和 Y 軸數據來自不同位址]</b> 和 <b>[上下限值取自暫存器]</b>。請見以下範例 1。</p>													
範圍上下限	<ul style="list-style-type: none"> <li>● 未勾選時： 上限 / 下限為常數。上下限是用於計算 X，Y 軸的刻度百分比，請見以下範例 2。</li> <li>● 勾選時： 使用者可改變上下限來達到縮放效果。請見以下範例 3。</li> </ul>													

## 範例 1

存取暫存器資料時，須同時考慮是否啟用【X 軸數據和 Y 軸數據來自不同位址】和【上下限值取自暫存器】。以下以實例說明各情況（假設皆使用 16-bit 暫存器）：

- 假設停用【X 軸數據和 Y 軸數據來自不同位址】，當【讀取位址】設為 LW-n 時：

啟用【上下限值取自暫存器】		停用【上下限值取自暫存器】	
X 資料	Y 資料	X 資料	Y 資料
下限	LW-n	LW-n+2	常數
上限	LW-n+1	LW-n+3	常數
第一筆數據	LW-n+4	LW-n+5	LW-n+1
第二筆數據	LW-n+6	LW-n+7	LW-n+3
第三筆數據	LW-n+8	LW-n+9	LW-n+5
第四筆數據	LW-n+10	LW-n+11	LW-n+7

- 假設啟用【X 軸數據和 Y 軸數據來自不同位址】，當【X 資料】為 LW-m，【Y 資料】為 LW-n：

啟用【上下限值取自暫存器】		停用【上下限值取自暫存器】	
X 資料	Y 資料	X 資料	Y 資料
下限	LW-m+0	LW-n+0	常數
上限	LW-m+1	LW-n+1	常數
第一筆數據	LW-m+2	LW-n+2	LW-n+0
第二筆數據	LW-m+3	LW-n+3	LW-n+1
第三筆數據	LW-m+4	LW-n+4	LW-n+2
第四筆數據	LW-m+5	LW-n+5	LW-n+3

## 範例 2

當【上下限值取自暫存器】未勾選時，上限 / 下限為常數。上下限用於計算 X，Y 軸的刻度百分比：

$$\text{刻度百分比} (\%) = \frac{\text{暫存器數據} - \text{下限}}{\text{上限} - \text{下限}}$$

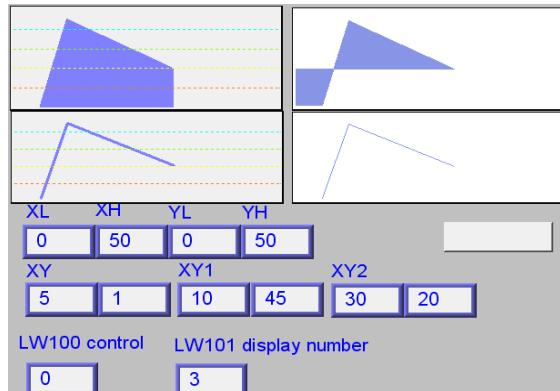
假設暫存器為 LW-n 且停用【X 軸數據和 Y 軸數據來自不同位址】，則上限 / 下限的數據來源會根據以下方式設定：

資料格式	16-bit	32-bit	64-bit (只支援於 cMT / cMT X)
X 軸下限	LW-n	LW-n	LW-n
X 軸上限	LW-n+1	LW-n+2	LW-n+4
Y 軸下限	LW-n+2	LW-n+4	LW-n+8
Y 軸上限	LW-n+3	LW-n+6	LW-n+12

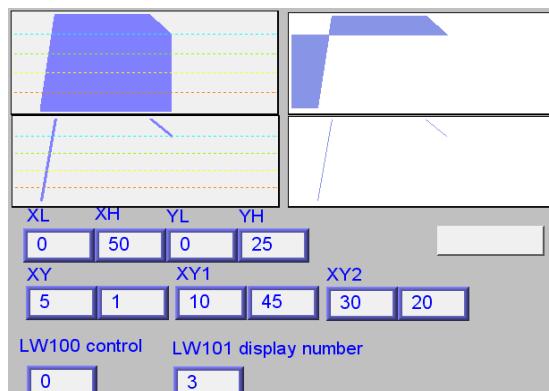
### 範例 3

若勾選【上下限值取自暫存器】，使用者可改變上下限來達到縮放效果。

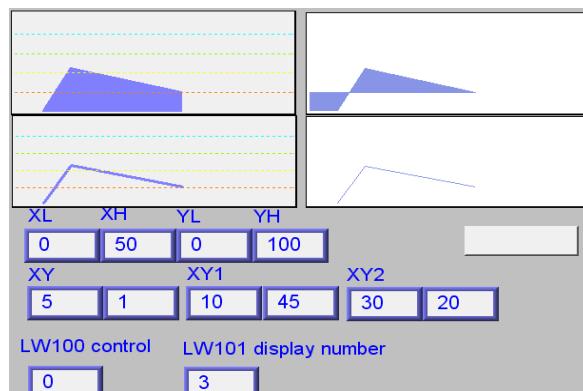
如下範例， $XL=X$  下限， $XH=X$  上限， $YL=Y$  下限， $YH=Y$  上限， $XY$ ， $XY1$ ， $XY2$  為三個 XY 數據。此時改變 Y 軸的上限，即可觀察縮放效果。效果如下：



原圖



改變 Y 軸上限為 25 (放大效果)



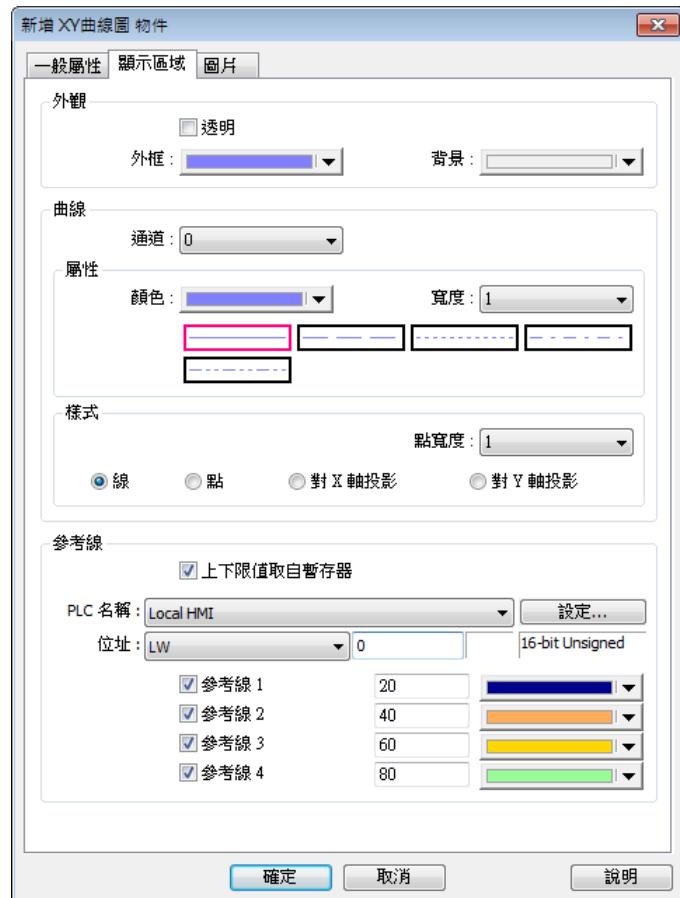
改變 Y 軸上限為 100 (縮小效果)

👉 詳細資訊請參考《13.17 趨勢圖》。



- X 和 Y 資料可使用不同格式，例如 X 資料使用 16-bit unsigned 而 Y 資料使用 32-bit signed，此時需特別留意位址的設置。
- 當 PLC 是 Tag PLC 時，例如 AB tag PLC，則 X 和 Y 一定要使用相同的位址格式。若選擇不同的格式會出現警示訊息。

## 顯示區域設定



### 設定

### 描述

**外觀**

勾選 [透明] 時背景為透明，無勾選則依照所選擇的色彩來表現外框及背景。

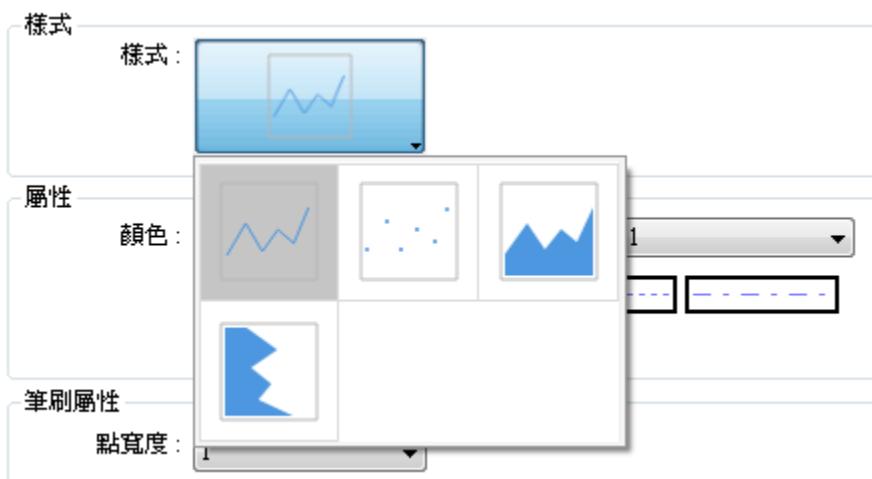
**曲線**

可在此設定通道所要顯示的屬性。

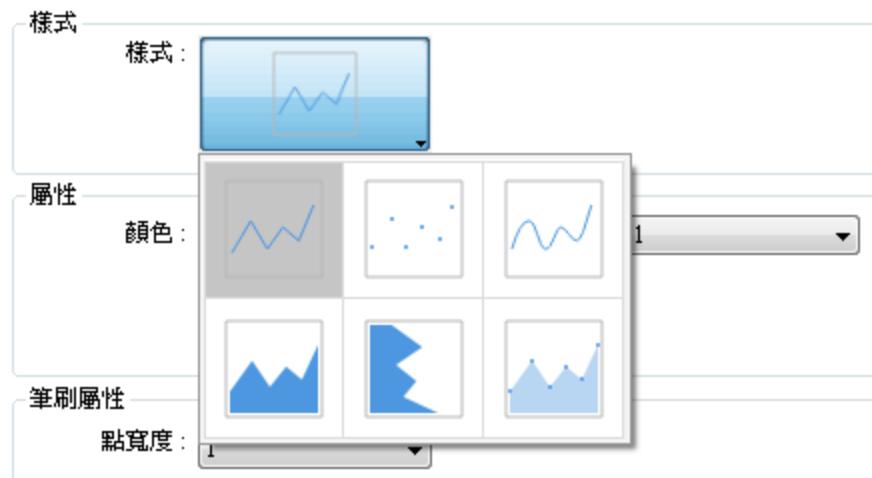
**樣式**

設定螢幕顯示方式。

eMT、iE、XE、mTV 系列支援四種樣式：



cMT / cMT X 系列支援六種樣式：



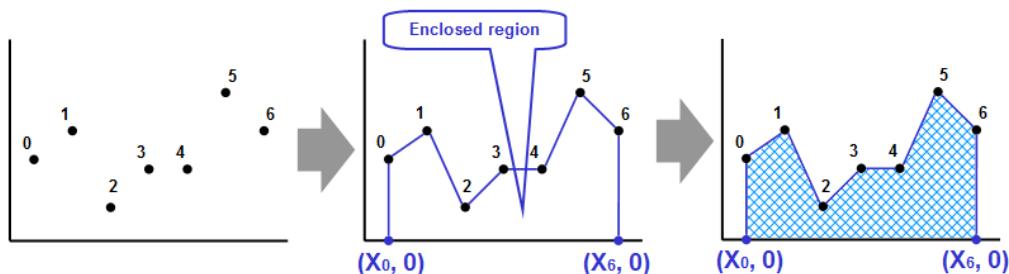
請見以下範例 4。

<b>屬性</b>	可設定線的 [顏色] 與 [寬度]。
<b>筆刷屬性</b>	可設定點寬度。
<b>參考線</b>	最多可畫四條參考線在曲線圖上，使用者可以自行選擇線條的色彩及參考的數值，並且依據所設定數值來顯示在螢幕上。若勾選上 [下限值取自暫存器]，則需設定一個參考線之讀取位址。

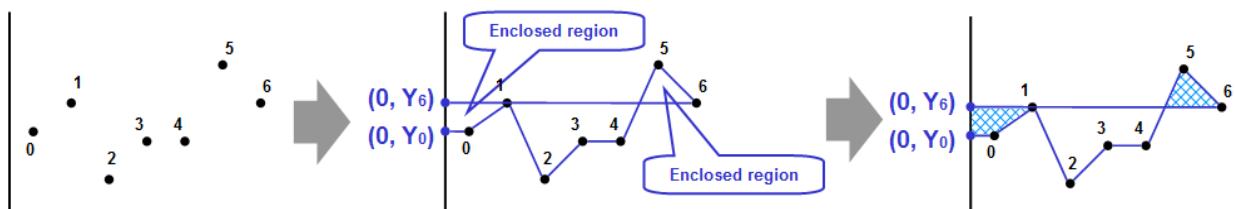
#### 範例 4

下圖中的曲線由 7 個點構成，由 P0 到 P6。系統繪製 [對 X 軸投影] 方式如以下步驟：

1. 自動計算出二個投影的點： X 軸 –  $(X_0, 0)$  和  $(X_6, 0)$
2. 依照點出現的順序，連結所有的點： $(X_0, 0)$ , P0... P6,  $(X_6, 0)$  並且最後連結到第一個點  $(X_0, 0)$
3. 填滿封閉區域，結果如下：



同樣的 [對 Y 軸投影] 可得：



 Note

- [XY 曲線圖] 物件未清除繪圖 (一直對控制位址輸入數值 1) 的情況下，可重複繪圖的次數為“32 次除以通道數”。

例如：

通道數目為 1 的情況 → 透過輸入 1 到控制位址可重複繪圖 32 次

通道數目為 2 的情況 → 透過輸入 1 到控制位址可重複繪圖 16 次

通道數目為 16 的情況 → 透過輸入 1 到控制位址可重複繪圖 2 次

## 13.21. 報警條與報警顯示

### 13.21.1. 概要

[報警條] 與 [報警顯示] 物件可以用來顯示已被定義在 [事件登錄] 中，且系統目前狀態滿足觸發條件的事件，此時這些事件也被稱為警示。[報警條] 與 [報警顯示] 物件將利用事件被觸發的時間先後，依序顯示這些警示，其中 [報警條] 物件使用單行跑馬燈型式呈現警示內容；[報警顯示] 物件則可同時顯示多行警示內容。下圖顯示不同物件對警示的表示方式。

 有關事件登錄詳細資訊請參考《7 事件登錄》。

**I (When LW 1 >= 10) 13:21:06 Event 0 (when LW0)**

[報警條] 物件，單行顯示多個事件

13/12/06	13:21:38	Event 2 (when LB10 = ON)
13/12/06	13:21:38	Event 3 (when LB11 = ON)
13/12/06	13:21:38	Event 0 (when LW0 == 100)
13/12/06	13:21:38	Event 1 (When LW 1 >= 10)

[報警顯示] 物件，可顯示多行

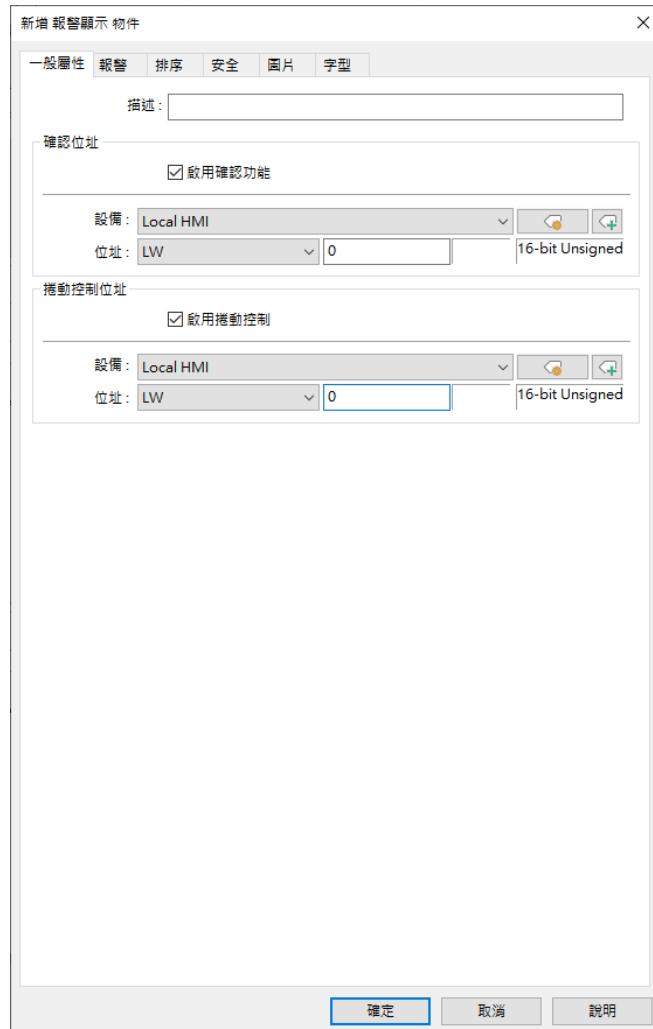
### 13.21.2. 設定



按下工具列上的 [資料/歷史] » [報警條] 按鈕後，即會出現物件屬性對話窗；相同方式，按下工具列上的 [資料/歷史] » [報警顯示] 按鈕後，即會出現物件屬性對話窗，設定各項屬性後按下確定鍵，即可新增一個物件。

#### 一般屬性設定

[報警條] 和 [報警顯示] 的設定不同之處在於，[報警顯示] 可以設定 [確認位址] 及 [捲動控制位址]，如下圖所示：



設定	描述
啟用確認功能	<p>當事件被確認時，在 [事件登錄] » [訊息] 頁設定的 [事件確認時寫入報警顯示/事件顯示物件] 中的數值會被輸出到 [事件顯示物件] 的 [確認位址]。詳細資訊請參考《7 事件登錄》。</p> <p><b>事件確認時寫入報警顯示/事件顯示物件</b></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">確認值 : 11</div>
捲動控制位址	<p>啟用捲動控制位址後，物件將根據控制位址的數值，向下捲動指定的行數。此數值由 0 開始計數。請注意，當啟用此功能，物件的上下與左右之捲動控制條將被取消，請務必確認物件寬度可以完整顯示內容。</p>

### Note

- 使用 cMT / cMT X 系列時，當手指觸壓螢幕且不移動位置時會被視為確認當筆事件。若手指觸壓螢幕且滑動則視為拖曳捲軸。

- 使用 cMT / cMT X 系列時，若啟用 [確認功能]，為了避免誤觸控到錯誤的項目，則項目的欄高會有最小固定欄高。因此在小尺寸的 HMI 上，有啟用 [確認功能] 的報警顯示的欄高會比無啟用 [確認功能] 的報警顯示略高。

## 報警設定

### 報警條



### 報警顯示



### 設定

### 描述

#### 顯示

僅 [報警條] 可設定。

#### iP/iE/XE/eMT/mTV 系列

速度 : 速度 5

文字平滑移動 (可能會提高 CPU 使用率)

文字平滑移動若啟用，報警訊息將會移動得更為平穩，但同時可能提高 CPU 的負載。

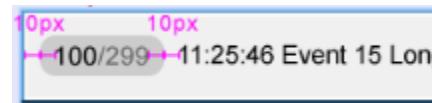
#### cMT / cMT X 系列

模式分為輪播與跑馬燈。

模式 :	輪播	模式 :	跑馬燈	
間隔 :	3.0 秒	速度 :	速度 5	

顯示當前和報警總數     多行顯示

選擇輪播時，可啟用當前和報警總數與間隔秒數



選擇跑馬燈時，速度設定 [報警條] 物件中所顯示文字的移動速度。啟用多行顯示可在 [報警條] 物件中將事件內容用多行顯示。此模式會影響效能，請斟酌使用。

#### 顯示的類別範圍

被觸發事件的 [類別] 需符合此處設定的顯示範圍才會被顯示(事件的類別在 [事件登錄] 中設定)。例如當 [報警條] 物件的 [類別] 被設定為 2~4，則僅有 [類別] 為 2、3 或 4 的事件才會被顯示在 [報警條] 物件中。詳細說明請參考《7 事件登錄》。

#### cMT / cMT X 系列

可選擇動態類別範圍。物件將根據暫存器的數值動態篩選顯示的事件類別。

##### 顯示的類別範圍

預定義  動態類別範圍

設備 :	Local HMI	設定...
位址 :	LW	0
		16-bit Unsigned

起始範圍 : LW-0  
結束範圍 : LW-1

#### 物件標題

僅 cMT / cMT X 系列 [報警顯示] 可設定。

##### 物件標題

使用物件標題

文字尺寸 : 16

顏色 :

	名稱
物件標題	Caption

#### 選項按鈕顯示 (cMT / cMT X 系列)

設定是否顯示物件上的選項按鈕 是否顯示。

##### ● 預定義：

預先設定好是否顯示選項按鈕。

● 動態：

預定義

動態

設備： Local HMI

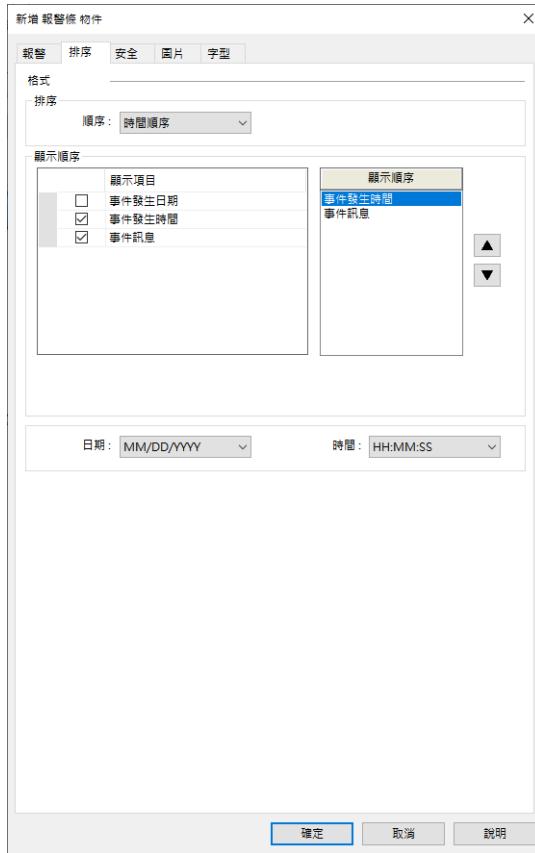
位址： LB ▾ 0



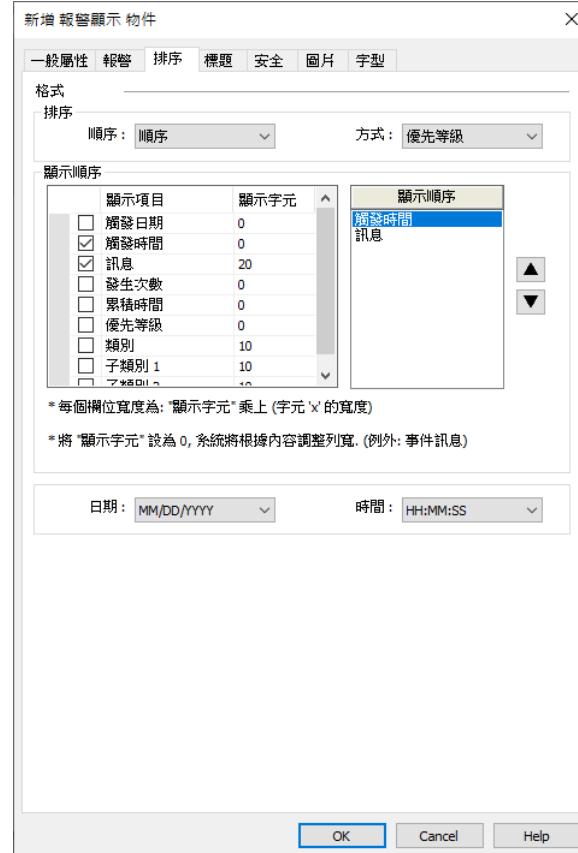
設定一個位元位址動態決定是否顯示選項按鈕。

## 排序設定

### 報警條



### 報警顯示



### 設定

### 描述

#### 排序 (報警條)

#### 時間順序

較晚發生的警示被排列在後 (或在下)

#### 時間逆序

較晚發生的警示被排列在前 (或在上)

#### 排序 (報警顯示)

#### 順序 - 方式 :

#### 順序 - 優先等級 (僅支援 cMT/cMT X)

優先等級較高的事件被排列在後 (或在下)

#### 逆序 - 優先等級 (僅支援 cMT/cMT X)

優先等級較高的事件被排列在前 (或在上)

#### 順序 - 發生次數

較多發生次數的事件被排列在後 (或在下)

#### 逆序 - 發生次數

較多發生次數的事件被排列在前 (或在上)

#### 順序 - 時間

較晚發生的事件被排列在後 (或在下)

**逆序 - 時間**

較晚發生的事件被排列在前（或在上）

**顯示順序** 使用者可勾選擬要顯示的資訊以及設定顯示順序。

**日期** 選擇顯示事件發生日的格式，共有以下 4 種模式。

MM/DD/YY、DD/MM/YY、DD.MM.YY、YY/MM/DD

**時間** 選擇顯示事件發生時間的格式，共有以下 4 種模式。

HH:MM:SS、HH:MM、DD:HH:MM、HH

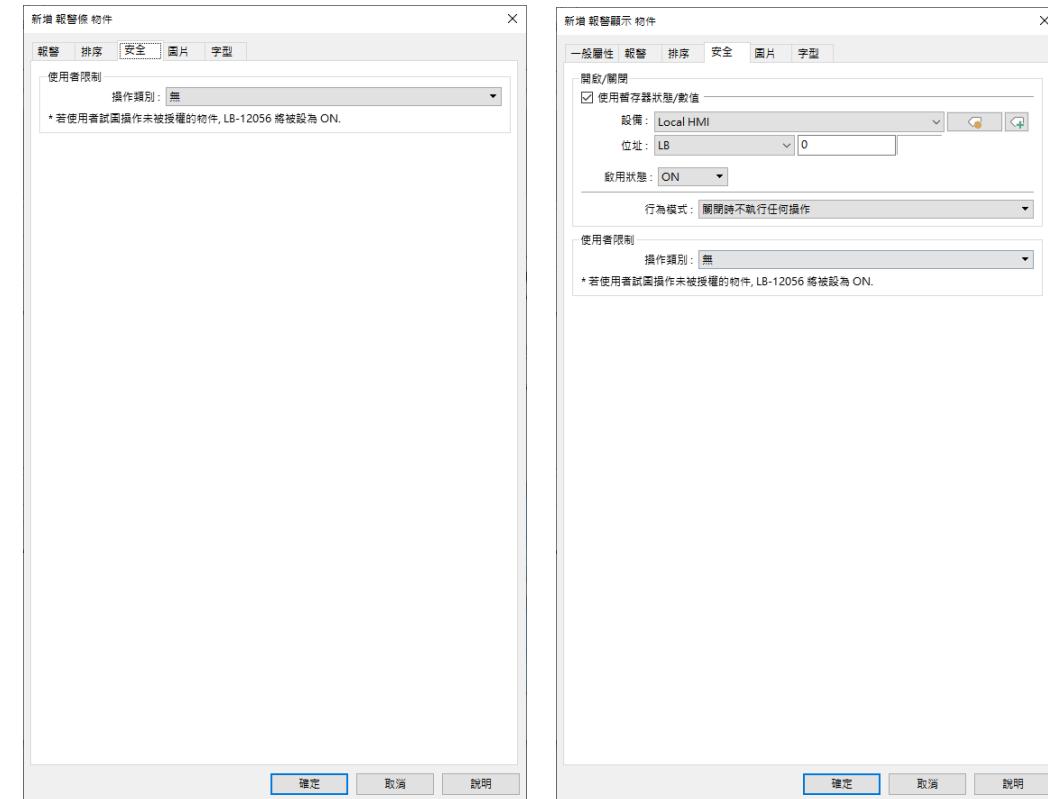
**標題設定**

cMT 系列支援使用標題，使用者在啟用使用標題後，可以設定文字顏色並分別設定各標題欲顯示之名稱。

## 安全設定

### 報警條

### 報警顯示



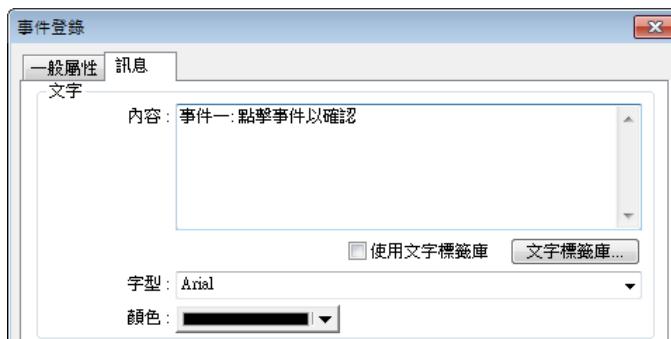
設定	描述
<b>開啟/關閉</b>	若勾選 [使用]，則此物件是否允許被操作，將決定於一個指定位元位址的狀態。如圖中設定，則必須在 LB-0 狀態為 ON 時，才允許操作此物件。
<b>使用者限制</b>	<p><b>操作類別</b> “無”表示任何用戶皆可操作。“管理者”表示只有 admin 帳號可以操作。</p> <p><b>當使用者無權操作此類別時隱藏此物件</b> 當用戶操作身份不符合此物件操作的等級時，物件會被隱藏。</p> <p>當使用報警顯示物件時，若反勾選此項目，當用戶無權限操作此類別物件時，用戶仍然能夠看到報警顯示物件，但無法變更或觸發物件。</p> <p>當使用報警條物件時，則無法反勾選此項目。</p>

## 字型設定

設定物件文字之尺寸與斜體效果。



而【報警條】和【報警顯示】中事件所顯示的訊息內容、字型與顏色是根據【事件登錄】中的設定，如下圖：



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

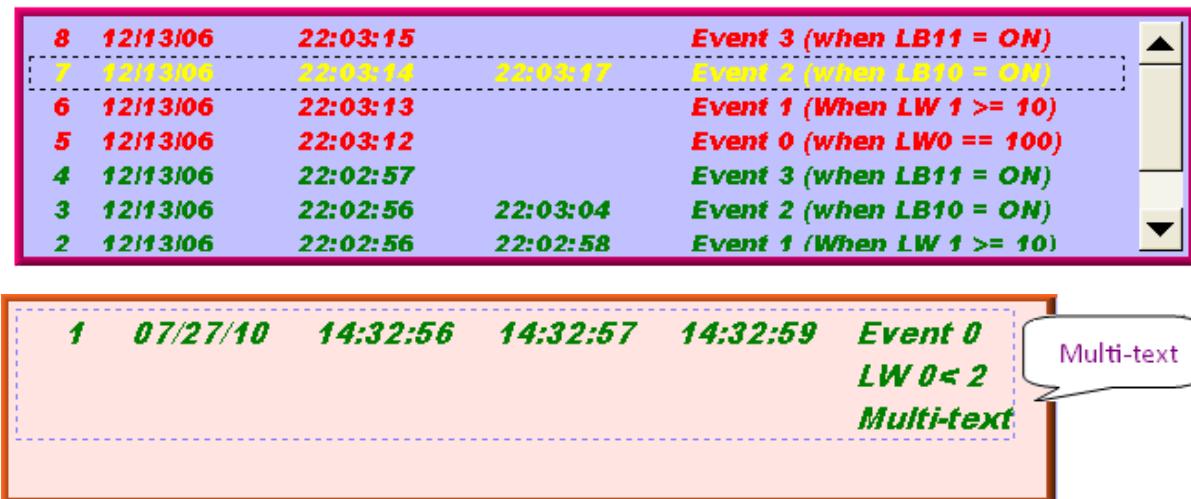
## 13.22. 事件顯示

### 13.22.1. 概要

[事件顯示] 物件可以用來顯示已被定義在 [事件登錄] 中，且曾經滿足觸發條件的事件。

[事件顯示] 物件將根據設定依據事件被觸發的時間順序、發生次數或是優先等級，依序顯示這些事件。

[事件顯示] 物件可以顯示事件發生日期、事件發生時間、事件確認時間、恢復正常時間、事件訊息、發生次數、累積時間、優先等級、類別以及子類別等內容。訊息內容可以用多行的方式顯示。



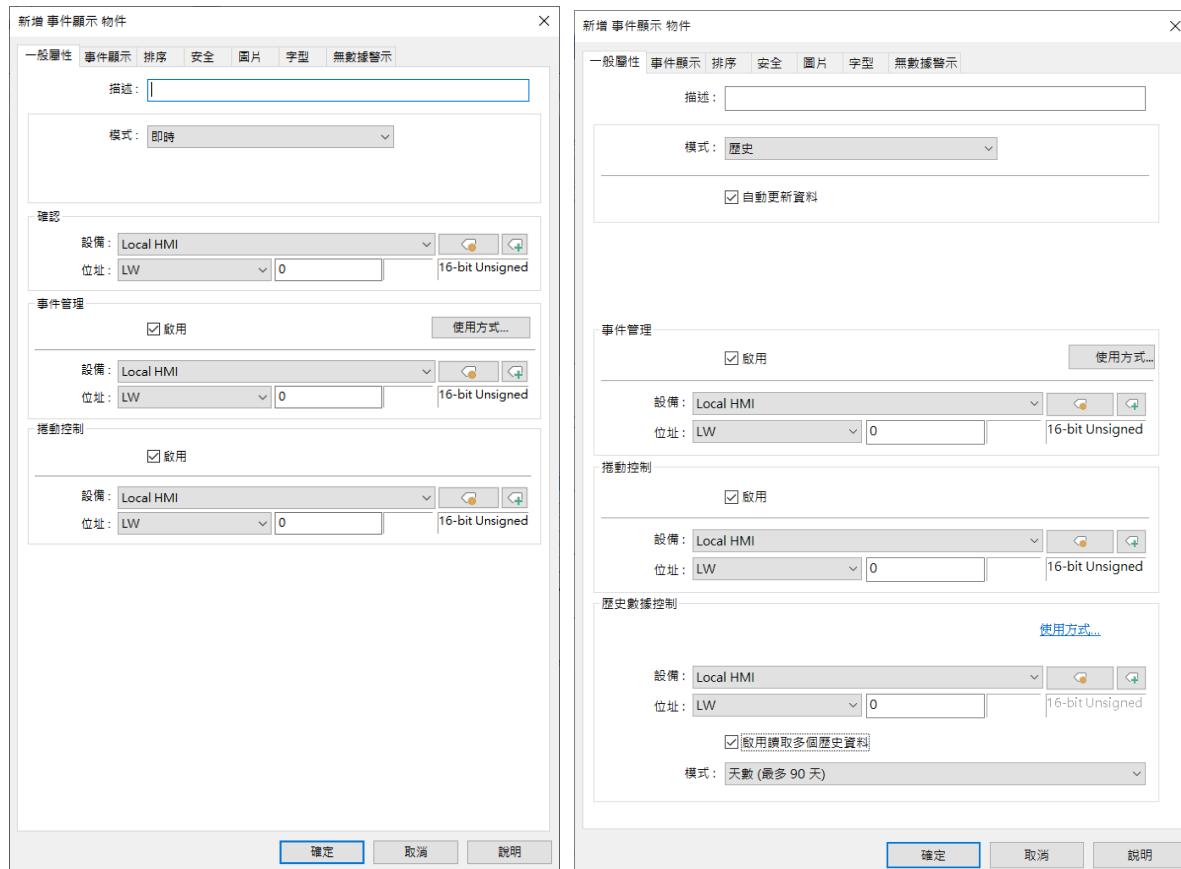
### 13.22.2. 設定



按下工具列上的 [資料/歷史] » [事件顯示] 按鈕後，即會出現 [事件顯示] 物件屬性對話窗，正確設定各項屬性後按下確定鍵，即可新增一個 [事件顯示] 物件。

## 一般屬性設定

### 13.22.2.1. eMT、iE、XE、mTV 系列



設定	描述
<b>模式</b>	<p>選擇事件來源的形式，可以選擇 [即時] 或 [歷史]。</p> <ul style="list-style-type: none"> <li>● 即時 將會顯示所有自開機以來被觸發的事件內容。</li> <li>● 歷史 系統會讀取記憶體內的事件檔案，顯示資料於物件中。若檢視的內容為當天的歷史記錄，將會每秒鐘自動更新檢視的內容。</li> </ul>
<b>確認</b>	<p>選擇 [即時] 模式時可設定此位址。</p> <p>當事件被確認時，在 [事件登錄] » [訊息] 頁設定的 [事件確認時寫入報警顯示/事件顯示物件] 中的數值會被輸出到 [事件顯示物件] 的 [確認位址]。詳細資訊請參考《7 事件登錄》</p> <p><b>事件確認時寫入報警顯示/事件顯示物件</b></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">確認值 : 11</div>
<b>歷史數據控制</b>	<p>選擇 [歷史] 模式時可設定此位址。</p> <ul style="list-style-type: none"> <li>● 無勾選 [啟用讀取多個歷史資料] 歷史模式僅可顯示單一天的歷史事件記錄。當歷史數據控制位址為</li> </ul>

LW-n 時，於 LW-n 輸入特定數值可顯示對應的歷史資料。

系統透過索引來選擇歷史記錄 .evt 檔：

輸入 0 則顯示最近一日的歷史資料

輸入 1 顯示第二近期的一筆歷史資料

輸入 2 顯示第三近期的一筆歷史資料

以此類推。

假設歷史數據控制位址是 LW-100，而目前有以下四筆歷史資料：

EL\_20100720.evt、EL\_20100723.evt、EL\_20100727.evt、

EL\_20100803.evt。當輸入數值到 LW-100 時，所顯示的歷史資料如下表：

LW-100 數值	相對應的歷史資料
0	EL_20100803.evt
1	EL_20100727.evt
2	EL_20100723.evt
3	EL_20100720.evt

#### ● 勾選 [啟用讀取多個歷史資料]

勾選後，可於一個事件顯示物件上顯示多天的歷史事件資料。使用時，會佔用兩個連續位址。請注意選擇 [天數] 和 [最後歷史資料索引] 對於控制位址暫存器定義的些微差異。

#### 選擇 [天數] :

[歷史數據控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。

[歷史數據控制位址+1] 為總共顯示的天數，計算方式為從起始日索引往前計算。

也就是當 [歷史數據控制位址] 為 LW-n 時，則歷史資料顯示範圍由 [LW-n] 內的數值所代表的日期開始算起，往前推算 [LW-n+1] 天。假設現在時間是 2010/6/10，LW-n 輸入數值為“1”，LW-n+1 輸入數值為“3”，則表示顯示的歷史資料範圍由前 1 天 (LW-n 的值) 20100609 開始，到往前推算三天 (包括 20100609)，所以應該顯示 20100607~20100609 範圍內的資料。



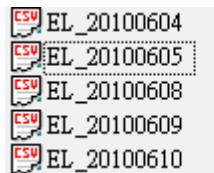
CSV	EL_20100604	No.4	1 KB EVT 檔案
CSV	EL_20100605	No.3	6 KB EVT 檔案
CSV	EL_20100608	No.2	17 KB EVT 檔案
CSV	EL_20100609	No.1	4 KB EVT 檔案
CSV	EL_20100610	No.0	12 KB EVT 檔案

#### 選擇 [最新歷史資料索引] :

[歷史控制位址] 為目前選擇的起始資料索引。0 代表第一個標記索引，1 代表第二個標記索引，以此類推。

[歷史控制位址+1] 為結束資料索引，數值需設定比起始資料索引大。

也就是當 [歷史數據控制位址] 為 LW-n 時，LW-n 輸入數值為“1”，  
LW-n+1 輸入數值為“3”，顯示為索引 No.1 至 No.3 的歷史資料，即為  
下圖中 No. 1、No. 2、No. 3 的歷史資料。



No.4	1 KB	EVT 檔案
No.3	6 KB	EVT 檔案
No.2	17 KB	EVT 檔案
No.1	4 KB	EVT 檔案
No.0	12 KB	EVT 檔案

注意: [LW-n] 內的數值需小於 [LW-n + 1] 內的數值，才可成立一個合法的查詢範圍。

系統最多可顯示 4MB 歷史資料，超出部份系統將略過。

以下為顯示資料過大的例子。

5 個歷史資料，每個 0.5 MB → 最多可顯示:  $8 \times 0.5\text{MB}$

5 個歷史資料，每個 1 MB → 最多可顯示:  $4 \times 1\text{MB}$

5 個歷史資料，每個 1.5 MB → 最多可顯示:  $2 \times 1.5\text{MB} + 1 \times 1\text{MB}$  (部分)

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 事件管理

### 啟用事件管理

啟用後，將特定的數值寫入暫存器 LW-n 及 LW-n+1 可對「事件顯示」物件給予不同的命令，命令條件及內容如下表：

位址	數值	命令內容
LW-n	0	顯示所有事件
	1	隱藏「已確認」事件
	2	隱藏「已恢復」事件
	3	隱藏「已確認」或「已恢復」事件
	4	隱藏「已確認」與「已恢復」事件
LW-n+1	1	可在 [即時模式] 下，刪除選擇的單一事件

## 捲動控制

啟用後，物件將根據控制位址的數值，向下捲動指定的行數。此數值由 0 開始計數。

例：如下圖所示，物件內總共有 10 項事件。捲動控制位址設定為 3 時，上圖因按時間順序排列，最上端事件為序號 4 事件；下圖因按時間逆序排列，最上端事件為序號 7 事件。



當啟用捲動控制功能時，捲動條已無控制功能，僅能表示目前內容的相對位置。若控制位址數據大於物件內的總行數，則會顯示最後面的內容。請注意，當啟用此功能，物件的上下與左右之捲動控制條將被取消，請務必確認物件寬度可以完整顯示內容。



請點選此圖示觀看教學影片，請先確定已連上網路線。

### 13.22.2.2. cMT、cMT X 系列

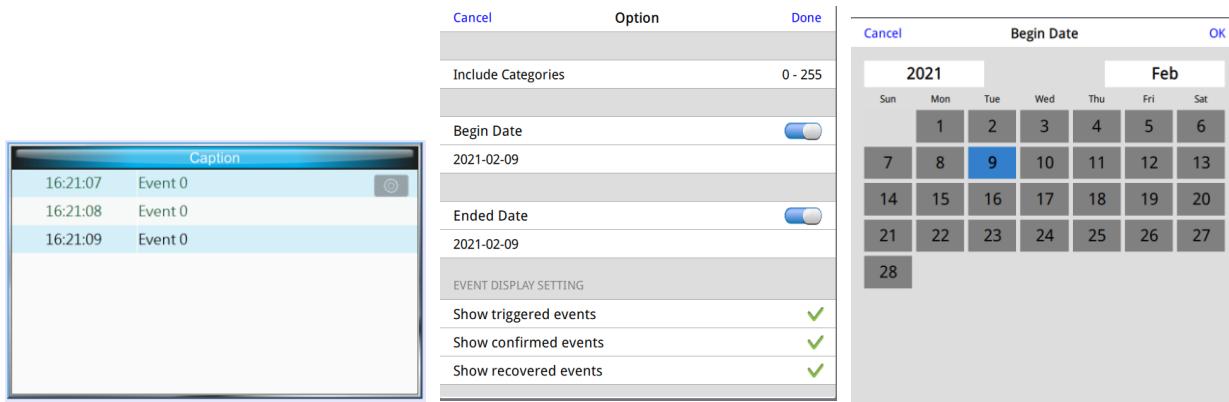


cMT/cMT X 系列的事件顯示物件僅顯示有在事件登錄設定勾選擬儲存為歷史資料的事件，如未勾選擬儲存為歷史資料，請使用報警條或報警顯示。

設定	描述
確認	<p>當事件被確認時，在 [事件登錄] » [訊息] 頁設定的 [事件確認時寫入報警顯示/事件顯示物件] 中的數值會被輸出到 [事件顯示物件] 的 [確認位址]。</p> <p>詳細資訊請參考《7 事件登錄》。</p> <p><b>事件確認時寫入報警顯示/事件顯示物件</b></p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <b>確認值 :</b> 11       </div>
捲動控制位址	<p>啟用後，物件將根據控制位址的數值，向下捲動指定的行數。此數值由 0 開始計數。</p>
歷史數據控制	<ul style="list-style-type: none"> <li>● 無勾選 [啟用讀取多個歷史資料] 顯示單一天歷史數據。歷史數據控制位址輸入 0 代表今天，1 代表昨天，以此類推。</li> <li>● 勾選 [啟用讀取多個歷史資料] 勾選後，可選擇顯示多天的歷史事件資料。使用時，會佔用兩個連續位址。請注意選擇 [天數] 和 [指定天數] 對於控制位址暫存器定義的些微差異。</li> </ul> <p><b>選擇 [天數] :</b></p> <p>[歷史數據控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。</p> <p>[歷史數據控制位址+1] 為總共顯示的天數，計算方式為從起始日索引往前計算。</p> <p><b>選擇 [指定天數] :</b></p> <p>[歷史控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。</p> <p>[歷史控制位址+1] 為結束日索引，數值需設定比起始日索引大。 例如當起始日索引為 4，結束日索引為 7，則代表要顯示 4 天前到 7 天前的資料。</p>

cMT / cMT X 中的事件顯示物件，會顯示所有已發生的事件並即時更新。

點選物件右上方的選項按鈕  可以設定開始和結束的日期。如未特別設定，則會顯示所有事件。

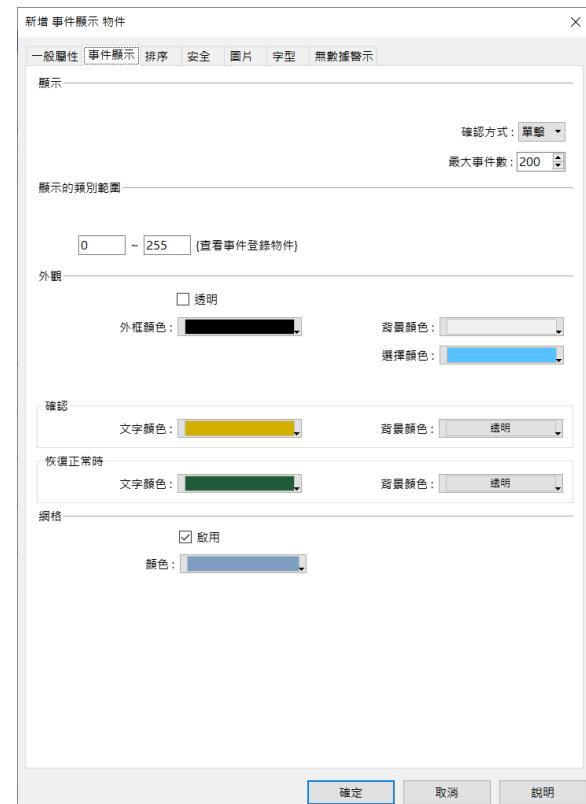


## 事件顯示設定

### cMT 系列



### eMT、iE、XE、mTV 系列



設定	描述
顯示的類別範圍	事件的 [類別] 需滿足此項設定範圍才會被顯示 (事件的 [類別] 在 [事件登錄] 中設定)。假設 [顯示的類別範圍] 被設定為 2 到 4，則僅有 [類別] 2 或 3 或 4 的事件，才會被顯示在 [事件顯示] 物件中。詳細資訊請參考《7 事件登錄》中有關 [類別] 的解釋。
確認方式	可選擇 [單擊] 或 [雙擊] 的確認方式。當事件發生時，使用者可依設定方式來做確認的動作。此處所謂“確認”的動作是指使用者對於已發生並顯示在 [事件顯示] 物件上的事件，此時除了會將該事件的顯示顏色轉變為 [確認] 的顏色之外，也會將此事件預先設定的輸出

值，寫至 [確認位址] 所設定的地址上。

當輸出地址為 LW-100，且事件確認時的寫入值為 31，則當使用者使用【確認】的動作時，LW-100 中的數據將被設定為 31，利用此項功能搭配【間接視窗】物件，可以在不同事件發生時彈跳出不同的視窗並說明事件的內容。

最大事件數  
(eMT、iE、  
XE、mTV 等  
列)

**顏色** 設定事件在確認後、恢復正常後、以及被點選時等狀態下的顯示顏色。系統將繪出虛線顯示剛被點選的事件。

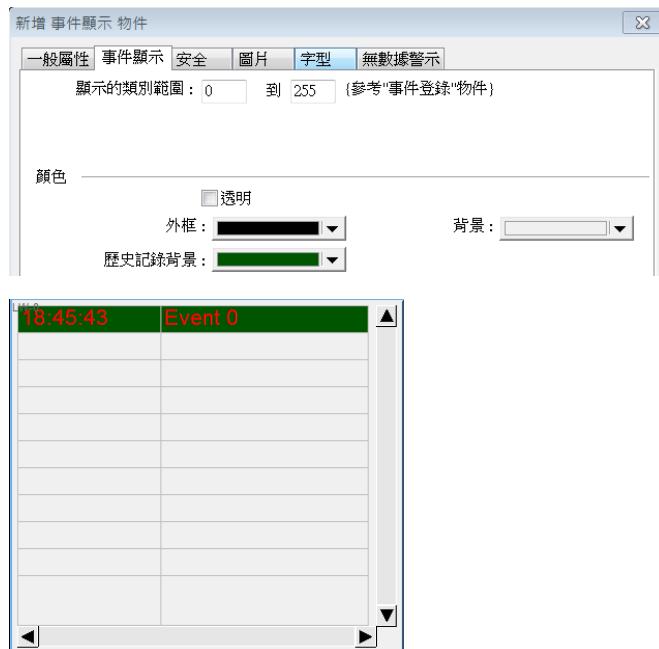
**顏色** 設定事件在確認後、恢復正常後、以及被點選時等狀態下的顯示顏色。系統將繪出虛線顯示剛被點選的事件。

確認



## 歷史記錄背景 (eMT, iE, XE, mTV)

使用事件顯示物件並搭配歷史模式時，可設定歷史記錄的背景顏色。檔案歷史記錄產生時，背景會依據選定顏色顯示。



網格

可以在物件上繪製格線，並選擇格線顏色。

自動調整欄寬(僅適用於 cMT 預設模式)

依照內容自動調整欄寬。

樣式

可選擇[預設]、[水晶風格]、[扁平化風格]。

**(cMT 系列)****使用物件標題**

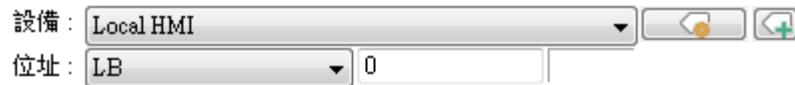
若勾選，則會在[配方檢視]物件顯示標題。可設定 [文字尺寸]、[顏色]、[名稱]。

**選項按鈕顯示****(cMT / cMT X  
系列)**

設定是否顯示物件上的選項按鈕 。

**● 預定義：**

預先設定好是否顯示選項按鈕。

**● 動態：**

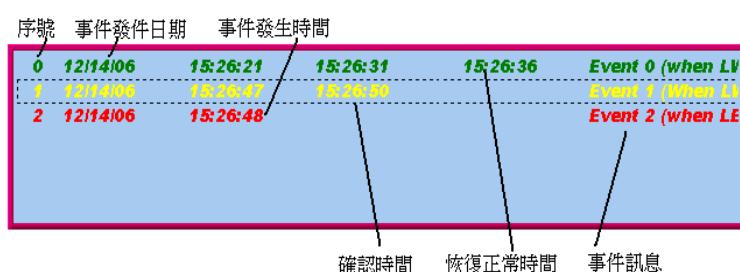
設定一個位元位址動態決定是否顯示選項按鈕。

**過濾器 (cMT/ cMT X 系列)****設定****描述****顯示的類別範**

事件的 [類別] 需滿足此項設定範圍才會被顯示 (事件的 [類別] 在

**圓**

[事件登錄] 中設定)。假設 [顯示的類別範圍] 被設定為 2 到 4，則僅有 [類別] 2 或 3 或 4 的事件，才會被顯示在 [事件顯示] 物件中。詳細資訊請參考《7 事件登錄》中有關 [類別] 的解釋。  
可選擇動態範圍，物件將根據暫存器的數值動態篩選顯示的事件類別。

**排序****cMT、cMT X 系列****eMT、iE、XE、mTV 系列****排序****順序 - 方式：****順序 - 優先等級 (僅支援 cMT/cMT X)**

在概觀頁面依優先度等級群組排序，將優先等級較高的事件群組排列在後 (或在下)，並在細節頁面依據時間逆序顯示該群組所有事件。

**逆序 - 優先等級 (僅支援 cMT/cMT X)**

在概觀頁面依優先度等級排序，將優先等級較高的事件群組排列在前（或在上），並在細節頁面依據時間逆序顯示該群組所有事件。

#### 順序 - 發生次數

eMT、iE、XE、mTV 系列將較多發生次數的事件排列在後（或在下）。

cMT/ cMT X 系列在概觀頁面依發生次序排序，將較多發生次數的事群組排列在後（或在下），並在細節頁面依據時間逆序顯示該群組所有事件。

#### 逆序 - 發生次數

eMT、iE、XE、mTV 系列將較多發生次數的事件排列在前（或在上）。

cMT/ cMT 系列在概觀頁面依發生次序排序，將較多發生次數的事件群組排列在前（或在上），並在細節頁面依據時間逆序顯示該群組所有事件。

#### 順序 - 時間

較晚發生的事件被排列在後（或在下）

#### 逆序 - 時間

較晚發生的事件被排列在前（或在上）

顯示順序	使用者可勾選擬想要顯示的資訊以及設定顯示順序。
日期	選擇顯示事件發生日的格式，共有以下 4 種模式。 MM/DD/YY、DD/MM/YY、DD.MM.YY、YY/MM/DD
時間	選擇顯示事件發生時間的格式，共有以下 4 種模式。 HH:MM:SS、HH:MM、DD:HH:MM、HH

## 標題設定



cMT / cMT X 系列支援使用標題，使用者在啟用使用標題後，可以設定文字顏色並分別設定各標題欲顯示之名稱。

## 安全設定

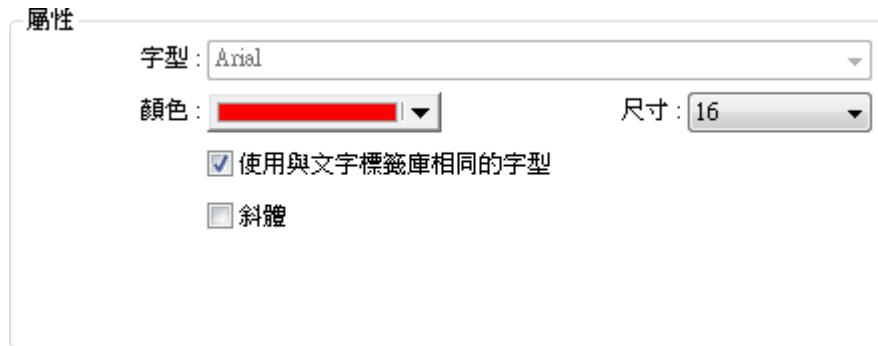


設定	描述
開啟/關閉	<p>若勾選 [使用]，則此物件是否允許被操作，將決定於一個指定位元位址的狀態。如圖中設定，則必須在 LB-0 狀態為 ON 時，才允許操作此物件。</p> <p><b>關閉時隱藏</b></p> <p>當指定的位元位址於關閉狀態時物件會被隱藏。</p>
使用者限制	<p><b>操作類別</b></p> <p>“無”表示任何用戶皆可操作。“管理者”表示只有 admin 帳號可以操作。</p> <p><b>當使用者無權操作此類別時隱藏此物件</b></p> <p>當用戶操作身份不符合此物件操作的等級時，物件會被隱藏。</p> <p>若反勾選此項目，當用戶無權限操作此類別物件時，用戶仍然能夠看到事件顯示物件，但無法變更或觸發物件。</p>

## 字型

即時模式：可以設定斜體及尺寸。字型將根據 [事件登錄] 中的設定顯示

歷史模式：可以設定斜體及尺寸，字型及顏色，或是勾選 [使用文字標籤庫中相同的字型]。



## 無數據警示



啟用無數據警示，可以設定當尚未有事件觸發前，顯示在物件上的文字。cMT / cMT X 系列不支援此功能。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.23. 資料傳輸

### 13.23.1. 資料傳輸 (視窗)

#### 13.23.1.1. 概要

[資料傳輸 (視窗)] 物件可以於物件所屬的視窗開啟時，將指定位址中的數據傳送到其他位址中。  
[資料傳輸 (視窗)] 可以使用手動按鈕的方式啟用數據傳送，也可以利用特定位址的狀態改變，來觸發數據傳輸的動作。



- 使用 [資料傳輸 (視窗)] 物件的位元觸發功能時，若將物件置於公用視窗，則無論基本視窗目前為何，當觸發條件滿足時，即會啟動資料傳輸。

#### 13.23.1.2. 設定



按下工作列的 [物件] » [資料傳輸] » [資料傳輸 (視窗)] 按鈕後，即會出現 [資料傳輸 (視窗)] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個物件。

## 一般屬性設定



設定	描述
來源位址	設定被傳送數據的來源位址。
目標位址	設定數據傳送的目的位址。
屬性	<p><b>字組數量</b> 數據的傳送數量，單位為字組。</p> <p><b>模式</b></p> <p><b>手動模式</b> 需使用手動的方式按下資料傳輸物件，才會進行資料傳送的動作。</p> <p><b>觸發模式</b> 利用所指定暫存器狀態的改變來觸發資料傳送的動作，利用【觸發模式】選擇需要的觸發方式，這些觸發方式包括： 可以選擇狀態由 OFF 變為 ON 或由 ON 變為 OFF 時，執行資料傳輸動作。也可選擇狀態改變時 (OFF&lt;-&gt;ON)，即執行資料傳輸動作。</p>
觸發位址	觸發模式所使用的暫存器位址在【觸發位址】中設定。
通知	若勾選，系統會在準備開始資料傳輸時，將指定暫存器狀態設為 [開] 或 [關]。

---

**自動重置**

當系統資料傳輸結束後，恢復【通知】位址至原狀態。

---

### 13.23.2. 資料傳輸 (背景) 定時式

#### 13.23.2.1. 概要

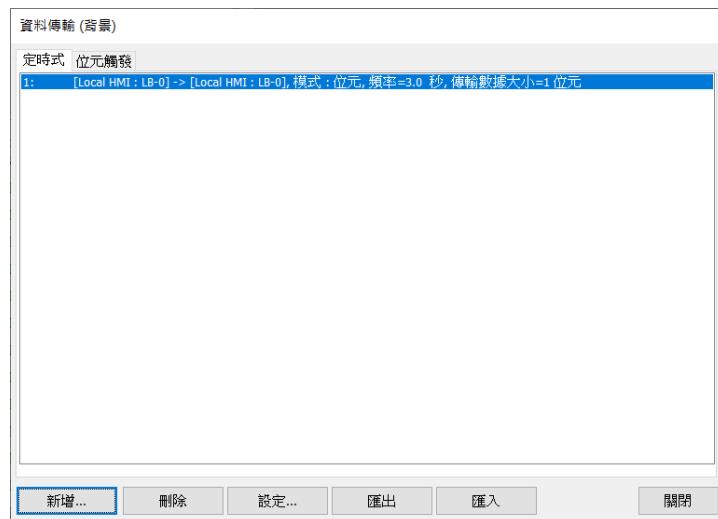
[資料傳輸 (背景)] 物件的定時式模式可以將指定位址中的數據根據設定的觸發時間定期傳送到其他位址中。

#### 13.23.2.2. 設定

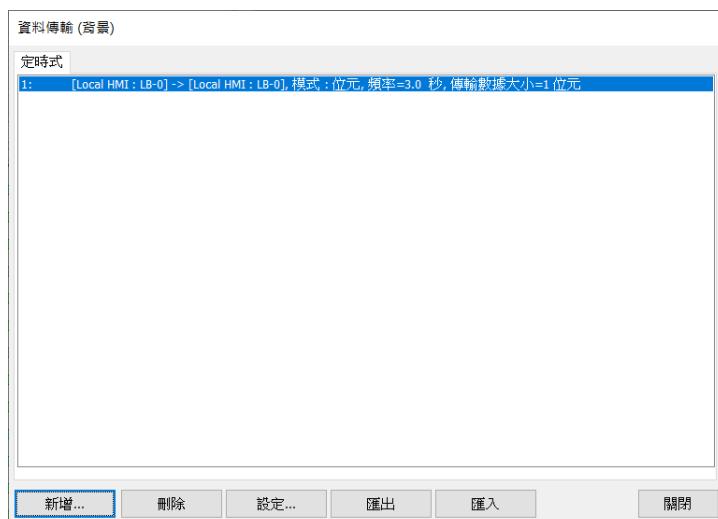


按下工作列的 [物件] » [資料傳輸] » [資料傳輸 (背景)] » 開啟 [定時式] 分頁 » [新增] 按鈕後，即會出現 [定時式資料傳輸] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個物件。資料傳輸(背景)定時式與資料傳輸(背景)位元觸發匯入/匯出功能需分別使用。

##### cMT / cMT X 系列



##### eMT、iE、XE、mTV 系列



## 一般屬性設定



### 設定

### 描述

#### 屬性

#### 位址類型

選擇被傳送數據的類型，可以選擇 [Bit] 或 [Word] 的數據。

#### 位元數量 / 字組數量

當 [位址類型] 選擇 [Bit] 型態時，數據傳送單位為 bit，使用 [位元數量] 設定傳送數量。

當 [位址類型] 選擇 [Word] 型態時，數據傳送單位為 word，使用 [字組數量] 設定傳送數量。

#### 間隔

數據傳送頻率，例如選擇 3 秒，則每隔 3 秒，將傳送數據到指定的位址中。

較小的間隔或是大量的資料傳輸可能會導致系統執行速度變慢，建議使用者拉長傳送的間隔或是一次傳送小量的資料，以避免系統執行速度變慢。

當需要設定短時間的傳輸時，請注意設定間隔的時間要大於資料傳輸的時間。例如：單次數據傳輸操作需要 2 秒，則間隔時間須設置超過

2 秒。

**只在指定的視窗被開啟時才執行**

若勾選，資料傳輸將只在所選擇的視窗被開啟時才執行。

**優先執行**

將此資料傳輸的優先順序提高，可能會降低畫面的刷新率。

---

**來源位址**

設定數據傳送的來源位址。

---

**目標位址**

設定數據傳送的目標位址。

---

**通知**

若勾選，系統會在準備開始資料傳輸時，將指定暫存器狀態設為 [開] 或 [關]。

**自動重置**

當系統資料傳輸結束後，恢復 [通知] 位址至原狀態。

---

### 13.23.3. 資料傳輸 (背景) 位元觸發

#### 13.23.3.1. 概要

[資料傳輸 (背景)] 物件的位元觸發模式可以讓 HMI 在背景持續偵測指定的位元位址狀態，進而觸發資料傳送到其他位址中。

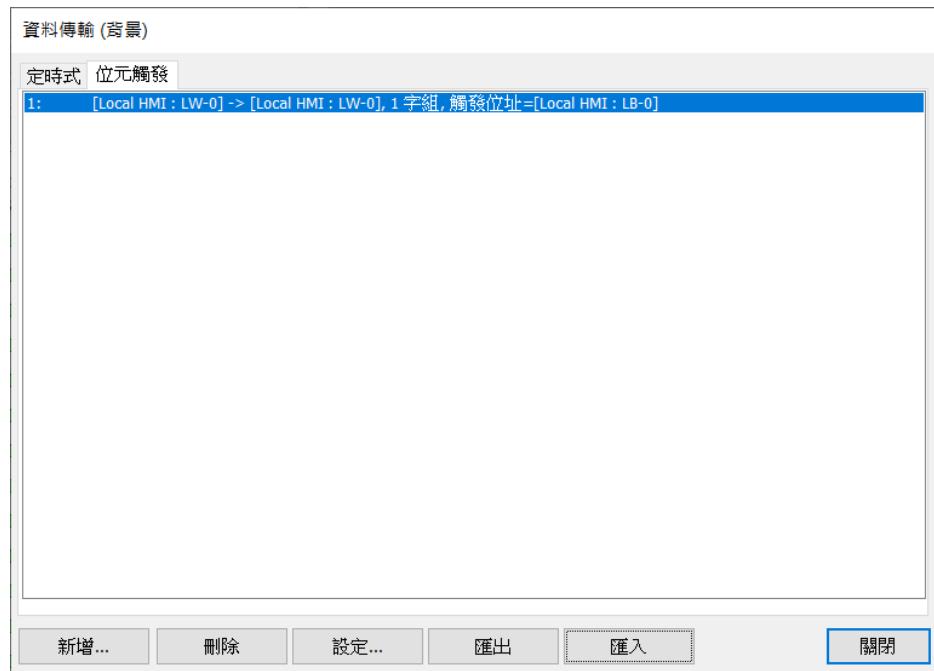
eMT、iE、XE、mTV 系列不支援此功能。

#### 13.23.3.2. 設定

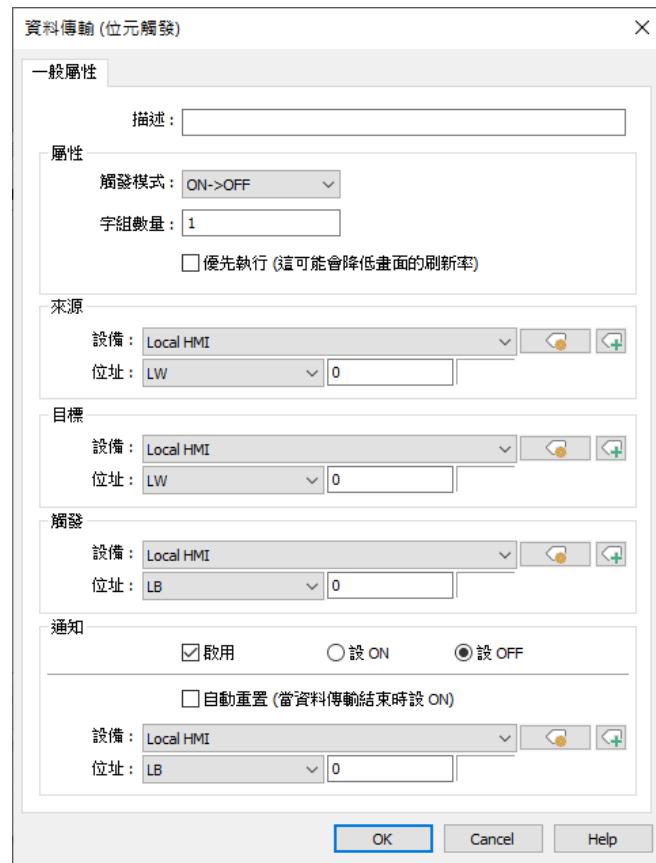


按下工作列的 [物件] » [資料傳輸] » [資料傳輸 (背景)] » 開啟 [位元觸發] 分頁 » [新增] 按鈕後，即會出現 [資料傳輸 (位元觸發)] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個物件。資料傳輸(背景)定時式與資料傳輸(背景)位元觸發匯入/匯出功能需分別使用。

#### cMT / cMT X 系列



## 一般屬性設定



設定	描述
<b>屬性</b>	<p><b>觸發模式</b> 利用所指定暫存器狀態的改變來觸發資料傳送的動作，利用【觸發模式】選擇需要的觸發方式，這些觸發方式包括： 可以選擇狀態由 OFF 變為 ON 或由 ON 變為 OFF 時，執行資料傳輸動作。也可選擇狀態改變時 (OFF&lt;-&gt;ON)，即執行資料傳輸動作。</p> <p><b>字組數量</b> 數據的傳送數量，單位為字組。</p> <p><b>優先執行</b> 將此資料傳輸的優先順序提高，可能會降低畫面的刷新率。</p>
<b>來源位址</b>	設定被傳送數據的來源位址。
<b>目標位址</b>	設定數據傳送的目的位址。
<b>觸發位址</b>	設定觸發模式所使用的暫存器位址。
<b>通知</b>	若勾選，系統會在準備開始資料傳輸時，將指定暫存器狀態設為【開】或【關】。
<b>自動重置</b>	當系統資料傳輸結束後，恢復【通知】位址至原狀態。

## 13.24. 備份

### 13.24.1. 概要

利用 [備份(觸發式)] 或 [備份(背景)] 物件可以將配方資料(RW, RW\_A)、事件記錄、配方資料庫、指定的資料取樣記錄及操作記錄複製到擴充裝置 (SD 卡或 USB 碟)，並可以指定備份的時間範圍或格式。例如事件記錄原來儲存在 SD 卡，此時可以在不需關機的情形下插上 USB 碟，並利用 [備份] 物件複製一份相同的資料到 USB 碟，並在不需關機的情形下，直接拔取 USB 碟，這些數據即可移至 PC 做進一步的分析。當備份動作進行中時，[LB-9039] 的狀態將維持在 ON。另外搭配 [郵件] 的設定，可以透過郵件功能將資料以夾帶方式透過郵件發送至指定的收件人信箱。[備份(觸發式)] 與 [備份(背景)] 物件的差別在於，[備份(觸發式)] 物件僅在放置物件的視窗上執行，[備份(背景)] 物件則在每一頁視窗均會執行。[備份(背景)] 物件僅支援 cMT / cMT X 系列。

### 13.24.2. 設定



按下工作列上的 [物件] » [檔案操作]，即可選擇 [備份(觸發式)] 或 [備份(背景)] 物件。

選擇 [備份(觸發式)] 後即會出現物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [備份(觸發式)] 物件。

使用 cMT / cMT X 系列時，可以選擇 [備份(背景)] 物件。在 [備份(背景)] 物件管理對話窗中按下 [新增] 按鍵，正確設定各項屬性後按下確認鍵，即可新增一個 [備份(背景)] 物件。

物件

## 一般屬性設定

[備份(視窗)] 物件

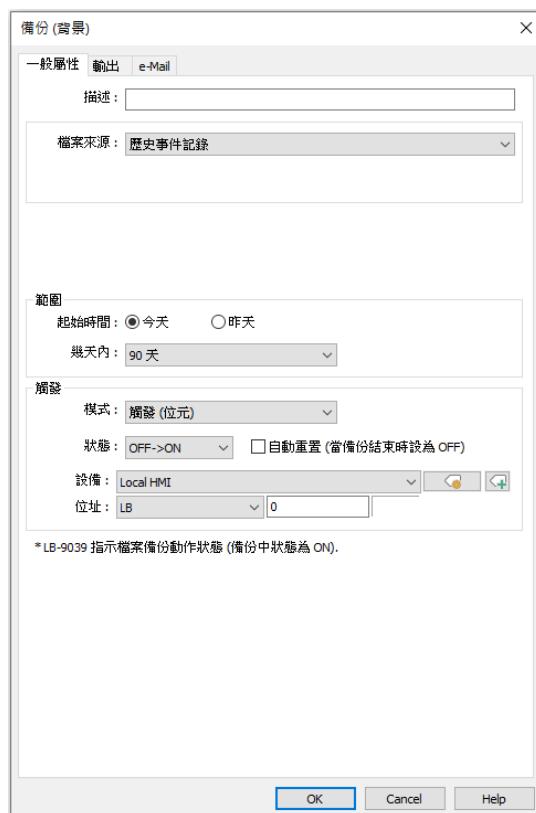
cMT / cMT X 系列



eMT、iE、XE、mTV 系列



[備份(背景)] 物件 : cMT / cMT X 系列



設定	描述
來源	<p><b>[RW]、[RW_A]、[配方資料庫]、[歷史事件記錄]、[歷史資料取樣]、[操作記錄]、[SQL 查詢]</b></p> <p>這些選項用來選擇要複製的檔案來源，當選擇檔案來源為 [歷史資料取樣] 時，需使用 [資料取樣物件索引] 選擇要複製的資料取樣記錄。</p> <p>除了[RW]、[RW_A]之外，其他選項僅在 EasyBuilder Pro 的專案內使用到時，才會顯示以供選擇。</p>
事件類別範圍 (需搭配歷史事件記錄)	<p>當備份歷史事件記錄時，若 [儲存格式] 為 CSV 檔案，才能使用此選項。類別範圍分成 [全部] 與 [部分]。舉例來說，若在 [部分] 的欄位輸入 3,5,8，備份的檔案將包含類別 3、5、8。若輸入 3~8，備份的檔案將包含類別 3~8。</p>
範圍	<p><b>資料取樣記錄</b></p> <p>設定儲存的檔案數量。例如，起始時間設定為 [當前] 並選擇 5 檔案，表示備份記憶體中最新的五筆檔案。</p> <p><b>事件記錄</b></p> <p>設定儲存天數。例如，起始時間設定為 [昨天] 並選擇 2 天。表示儲存昨天及前天的資料。選擇全部，可儲存最多 90 天的資料。</p>
觸發	<p><b>模式</b></p> <p>選擇物件的執行方式。</p> <p><b>手動</b></p> <p>使用者只需按壓備份物件，即可執行檔案複製動作。</p> <p><b>觸發 (位元)</b></p> <p>當指定的暫存器狀態改變符合觸發條件時，物件將執行檔案複製動作。</p> <p>可以選擇狀態由 OFF 變為 ON 或由 ON 變為 OFF 時，執行檔案複製動作，也可選擇狀態改變時(OFF&lt;-&gt;ON)，即執行檔案複製動作。</p> <p> 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。</p> <p><b>觸發 (字組)</b></p> <p>可以透過 [觸發位址] 設定所需備份資料的時間範圍。</p> <p>[觸發位址] 的用法如下(假使目前觸發位址被設定為 LW-0)：</p> <p>LW-0: 當此位址所指定暫存器中的數據由 0 變為 1，將觸發備份動作。</p> <p>LW-1: 此位址所指定暫存器中的數據用來指定備份的起始時間。</p> <p>LW-2: 此位址所指定暫存器中的數據用來指定備份的天數 (最多 90 天)。</p>



cMT-SVR 的 [備份(觸發式)] 物件僅支援 [手動]，不支援 [觸發(位元)] 與 [觸發(字組)]。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 觸發位址

當指定的暫存器狀態設為 ON 時，物件將執行檔案複製動作。備份完成後，該暫存器將被設為 OFF。

#### Note

- 請注意所有的資料必須要被儲存在一任意記憶體之中 (如：HMI memory 或 USB 碟或 SD 卡)，否則無法使用備份功能。
- 單次備份最大天數為 90 天。
- 當儲存檔案至 USB 碟或 SD 卡時，一個 FAT32 資料夾可儲存的檔案數量取決於檔案的名稱長短，當檔名越長，則資料夾可儲存的檔案數量就越少。

 關於 cMT / cMT X 系列同步資料取樣記錄及事件記錄至外部裝置的方式請參考《7 事件登錄》、《8 資料取樣》。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 輸出設定

### cMT / cMT X 系列



### eMT、iE、XE、mTV 系列



#### 設定

#### 描述

##### 備份位置

設定來源檔案的備份位置。

##### SD 卡 / USB 碟

備份至 SD 卡或 USB 碟。外部裝置需事先連接在人機上。

##### 遠端列印/備份伺服器 (僅 eMT, iE, XE, mTV 系列)

備份至遠端備份伺服器。若使用此選項，使用者需先在 [系統參數設定] » [列印/備份伺服器] 中設定。

請注意，操作記錄僅能備份至遠端列印/備份伺服器。欲備份至 SD 卡 /USB 碟，請使用操作記錄的控制位址。

詳細資訊請參考《26 EasyPrinter》。

##### 郵件

將備份資料以郵件寄出。

若使用此選項，使用者需先在 [系統參數設定] » [郵件] 中設定，再至 [備份] 物件屬性 » [e-Mail] 設定收件者、主旨、訊息等郵件設定。

##### FTP (僅支援 cMT/ cMT X 系列)

將檔案備份到 FTP 伺服器，須設定 [FTP 路徑] 及 [重複的檔案名稱] 的處理方式。使用此功能前，需先在 [系統參數設定] 的 [FTP] 中啟用 [FTP 伺服器] 功能。

#### 儲存格式

選擇想要儲存的備份檔案格式。

##### eMT, iE, XE, mTV 系列：

- HMI 事件記錄檔(.evt) / HMI 資料記錄檔(.dtl)
- Comma Separated Values (.csv)

另外，事件記錄的備份檔案中有事件類別欄位。

	A	B	C	D	E
1	Event	Category	Date	Time	Message
2	0	1	2013/7/4	16:12:11	Event A
3	2	1	2013/7/4	16:12:12	Event A
4	0	0	2013/7/4	16:12:33	Event B
5	2	0	2013/7/4	16:12:36	Event B
6	0	0	2013/7/4	16:12:37	Event B
7	1	0	2013/7/4	16:12:37	Event B
8	2	0	2013/7/4	16:12:39	Event B
9	0	0	2013/7/4	16:12:40	Event B

0: 事件觸發

1: 事件確認

2: 事件恢復正常

HMI 事件記錄檔 (.evt) 和 HMI 資料記錄檔 (.dtl) 都可以利用 EasyConverter 輕鬆轉成 .xls 或 .csv 格式。

- SQLite 資料庫檔案 (.db)

##### cMT / cMT X 系列：

- SQLite 資料庫檔案 (.db) (限備份到郵件時)
- Comma Separated Values (.csv)

#### 劃分方式

不劃分	歷史資料備份全部產生至同一個檔案中。
日期	歷史資料備份檔案依日期劃分。 如資料取樣使用 [自訂檔案管理]，則會以檔案為單位，和檔案管理邏輯相同。
行數	歷史資料備份檔案依設定的行數劃分。

#### 將 BOM 加入檔案開頭，使 EXCEL 能夠正確解釋非 ASCII 字串

當備份資料取樣或事件記錄檔為 CSV 格式時，可選擇在檔案開頭加入 BOM (Byte Order Mark)，讓 Excel 可以直接開啟包含非 ASCII 內容的.csv 檔案，並可選擇是否包含其他資訊如標題、輸出時間、發生次數，以及累積時間資訊。



### 啟用校驗和以確保資料完整性

備份的歷史記錄可以透過 EasyConverter 判斷檔案校驗和，以確保資料完整性。

## 進階設定 (eMT, iE, XE, mTV 系列)



### 設定

### 描述

#### 序號

若啟用，當備份歷史資料時，可在檔案名稱結尾上附加自訂的序號。當備份歷史資料時，會使用設定的 LW 位址的數值附加五碼的號碼在檔案名稱結尾上。備份完成後，系統會自動將 LW 位址加 1。

序號的範圍是 0~65535。

例如：當序號為 123，會附加 00123 在檔案結尾。

資料取樣檔案 - 20140407.dtl 將被備份成 2014040700123.dtl

事件記錄檔案 - 20140407.evt 將被備份成 2014040700123.evt

### 選項

#### 備份後刪除舊的檔案

啟用此功能時，在備份後歷史檔案將被自動刪除。



- cMT / cMT X 系列不支援進階設定。

### e-Mail 設定



### 設定

### 描述

於檔案名稱加入.txt 副檔名  
可以避免信件被防毒軟體阻擋

若啟用，當備份資料以郵件附檔傳送時，會在檔案名稱結尾加入.txt 副檔名，以避免遭受郵件伺服器或防毒軟體阻擋。

---

**擋****收件者、主旨、訊息**

備份所寄出的郵件的收件者電子郵件地址，郵件的主旨及訊息內容。

---

## 13.25. 媒體播放器

### 13.25.1. 概要

媒體播放器提供在 HMI 上播放影片檔案的功能。舉例來說，現場以動態影片來提供維修保養作業指示，將使維修保養工作更為簡易。

若使用 cMT / cMT X 系列 HMI，無法透過 cMT Viewer 檢視此物件。

### 13.25.2. 設定



按下工作列的 [物件] » [媒體播放器] 按鈕後即會開啟 [媒體播放器] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [媒體播放器] 物件。

#### 一般屬性設定

cMT / cMT X(cMT-SVR 除外)



設定

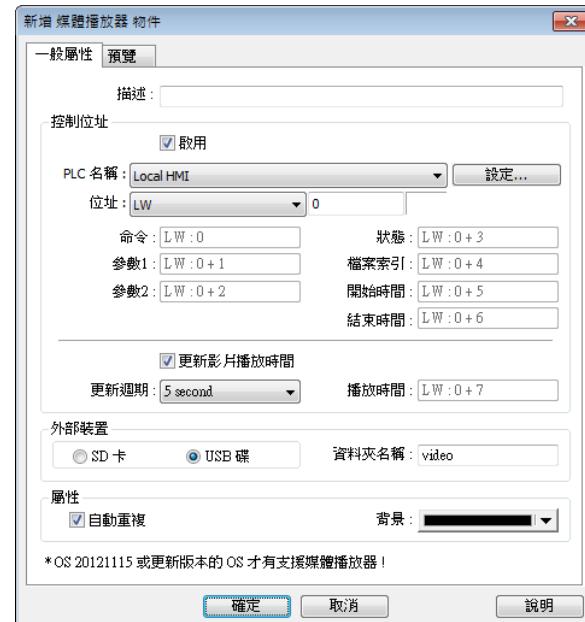
##### 控制模式

**不控制**：無法手動控制影片的播放狀態。在視窗開啟時，影片會自動播放。

**使用使用者介面控制**：使用系統提供的使用介面控制媒體播放。

**使用命令控制**：使用控制位址來控制媒體播放。

eMT、XE 系列



描述

恢復模式	定義從其他頁面跳回至有媒體播放器的視窗時，影片恢復播放時的位置。可設定 <b>恢復先前播放位置</b> 或 <b>重新播放</b> 。
控制位址	<ul style="list-style-type: none"><li>● 啓用時 使用者可針對【媒體播放器】進行控制並且得到播放資訊。必須指定一位址用來控制物件行為。</li><li>● 未啓用時 無法手動控制影片的播放狀態。在視窗開啟時，系統會自動播放影片。</li></ul> <p><b>命令(控制位址 + 0)</b> 控制【媒體播放器】的動作模式。</p> <p><b>參數 1(控制位址 + 1)</b> 相對於特定命令所傳入的參數 1。</p> <p><b>參數 2(控制位址 + 2)</b> 相對於特定命令所傳入的參數 2。</p> <p><b>狀態(控制位址 + 3)</b> 記錄檔案狀態、播放情況及錯誤代碼。</p> <p><b>檔案索引(控制位址 + 4)</b> 播放的檔案位於指定目錄下的索引 (以檔名排序，建議以數字為起始檔名)。</p> <p><b>開始時間(控制位址 + 5)</b> 影片開始時間(秒)。(通常為 0)</p> <p><b>結束時間(控制位址 + 6)</b> 影片結束時間(秒)。(影片的時間長度)</p> <p><b>更新影片播放時間</b> 啓用時，每隔【更新週期】(秒) 會將影片已播放時間寫入【播放時間】暫存器中。</p> <p><b>更新週期</b> 【播放時間】的更新週期，範圍由 1 至 60 秒。</p> <p><b>播放時間(控制位址 + 7)</b> 影片已播放時間(秒)(介於【開始時間】與【結束時間】之間)。</p>
外部裝置	<p>選擇播放 SD / USB 裡的檔案。</p> <p><b>資料夾名稱</b> 影片檔案放置的資料夾名稱。檔案必須被放置於資料夾中且資料夾只能為一層，多層資料夾將不會被接受 (例如指定【資料夾名稱】為"example\ex"將會出現錯誤)。</p> <p>【資料夾名稱】不可空白，必須為英數字，且全部由 ASCII 字元所組成。</p>
屬性	<b>自動重複</b>

當所有的影片播放結束，會自動跳回第一個影片從頭播放。

例如：Video 1 > Video 2 > Video 1 > Video 2

**背景** 指定物件背景顏色。

### Note

- 預設的暫存器格式為 16 位元無正負號；當指定之暫存器為 32 位元時，只有較低的 16 位元產生作用，並請將較高的 16 位元固定為 0。

## 控制命令

以下說明各種控制命令的設定。

- **播放索引檔案**

[命令] = 1

[參數 1] = 檔案索引

[參數 2] = 忽略 (應設為 0)

### Note

- 檔案以檔名排序。
- 假如找不到檔案，則將 [狀態] 的位元 8 設為 ON。
- 若需中途切換影片，請先將正在播放的影片停止。

- **播放上一個檔案**

[命令] = 2

[參數 1] = 忽略(應設為 0)

[參數 2] = 忽略(應設為 0)

### Note

- 若 [檔案索引] 為 0，則輸入命令 2 會從頭播放原檔案。
- 假如找不到檔案，則將 [狀態] 的位元 8 會被設為 ON，代表命令錯誤。

- **播放下一個檔案**

[命令] = 3

[參數 1] = 忽略(應設為 0)

[參數 2] = 忽略(應設為 0)

### Note

- 如果找不到檔案，則播放索引值 0 的檔案。
- 假如找不到檔案，則將 [狀態] 的位元 8 會被設為 ON，代表命令錯誤。

- **暫停/播放 切換**

[命令] = 4

[參數 1] = 忽略(應設為 0)

[參數 2] = 忽略(應設為 0)

- 停止播放並關閉檔案

[命令] = 5

[參數 1] = 忽略 (應設為 0)

[參數 2] = 忽略 (應設為 0)

- 從指定位置開始播放

[命令] = 6

[參數 1] = 目標時間 (以秒為單位)

[參數 2] = 忽略 (應設為 0)

 Note

■ 參數 1 (目標時間) 應小於結束時間，若超出結束時間則由結束時間前 1 秒開始播放。

- 往後跳躍 (秒)

[命令] = 7

[參數 1] = 目標時間 (以秒為單位)

[參數 2] = 忽略 (應設為 0)

 Note

■ 從目前時間往後跳躍 [參數 1] 指定秒數後開始播放。若系統目前為暫停播放影片狀態，則跳躍動作會在開始播放後才會進行。

■ 若播放時間超過結束時間，則由結束時間前 1 秒開始播放。

- 往前跳躍 (秒)

[命令] = 8

[參數 1] = 目標時間 (以秒為單位)

[參數 2] = 忽略 (應設為 0)

 Note

■ 從目前時間往前跳躍 [參數 1] 指定秒數後開始播放。若系統目前為暫停播放影片狀態，則跳躍動作會在開始播放後才會進行。

■ 若播放時間少於開始時間，系統會從頭播放影片。

- 設定音量

[命令] = 9

[參數 1] = 音量 (0 ~ 128)

[參數 2] = 忽略(應設為 0)

 Note

■ 預設為最大音量 (128)。



- 設定影像放大倍率

[命令] = 10

[參數 1] = 影像大小 (0 ~ 16)

[參數 2] = 忽略(應設為 0)

 Note

- [參數 1 = 0] : 配合物件大小。
- [參數 1 = 1 ~ 16] : 放大倍率範圍 25% ~ 400%，設定 1 為放大 25%，2 為 50%，3 為 75% 以此類推。

- 狀態(控制位址 + 3)

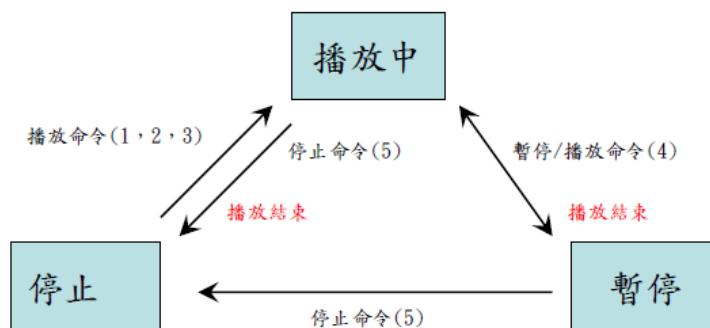
當 HMI 正在播放影片，則 [檔案開啓位元] 位元 00 及 [檔案播放位元] 位元 01 將被同時設為 ON (0 → 1)。相反地，若找不到檔案或輸入的命令不正確，則 [錯誤命令位元] 位元 08 將會被設為 ON (0 → 1)。若在播放過程中發現檔案格式錯誤或任何磁碟 I/O 錯誤，則 [檔案錯誤位元] 位元 09 將被設為 ON (0 → 1)。

15	09 08	02 01 00	位元
保留 (0 固定)	0 0		0 0

位元 00： 檔案開啓位元(0：未開啓檔案；1：已開啓檔案)  
 位元 01： 檔案播放位元(0：未播放檔案；1：檔案播放中)  
 位元 08： 錯誤命令位元(0：正確命令格式；1：錯誤命令或參數)  
 位元 09： 檔案錯誤位元(0：檔案格式及讀取正確；1：檔案格式或讀取錯誤)

 Note

- 參考下圖 [媒體播放器] 之狀態轉換圖可知：  
 “停止” 時狀態 = 0, “暫停” 時狀態 = 1, “播放” 時狀態 = 3



- 請透過設定 [命令]，[參數 1] 及 [參數 2] 來操作物件，並將其它位址視為唯讀。

## 預覽設定

使用者可利用預覽功能來檢查 HMI 是否支援欲播放的影片格式。



設定	描述
前進<< / 後退 >>	往前或往後快轉 (以 1 分鐘為單位)。
播放 / 暫停	可選擇影片開始播放或暫停播放。
停止	停止播放影片並關閉檔案。若需測試另一影片，必須先停止播放目前影片。
載入	選擇要預覽的影片。

### Note

- 使用者需注意 HMI 上同一時間只能有一個影片檔被開啟。
- 假如使用者沒有啓用 [控制位址] 且沒有設定 [自動重複]，則指定目錄下的第一個檔案播完一遍後，系統會自行將影片關閉。
- 當沒有啓用 [控制位址] 時，物件生成後自動到指定目錄下找尋第一個檔案 (以檔名排序)開始播放。
- 當影片可以使用媒體播放器的預覽功能，表示人機支援此影片格式並且可以播放。若是在人機上有播放品質不佳的狀況請調整影片的解析度。
- 支援檔案格式有: mpeg4, xvid, flv...等等。
- 影片的建議幀率為 30fps。

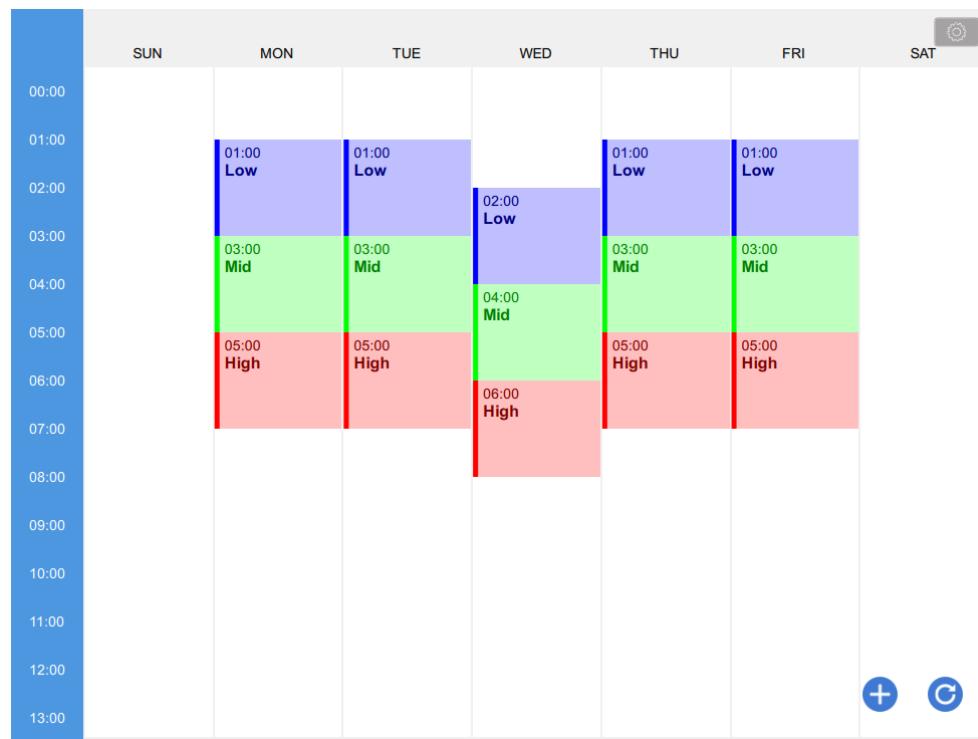


請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.26. BACnet Schedule

### 13.26.1. 概要

[BACnet Schedule] 物件提供圖形化介面存取 BACnet IP 設備中排程(Schedule)的資訊。僅 cMT / cMT X 系列支援此功能。



### 13.26.2. 設定



在 [系統參數] 的 [設備]，新增 BACnet/IP 驅動，並確認掃描的 BACnet/IP 裝置或匯入的標籤檔案中含有 Schedule。

按下工作列的 [物件] » [時間相關] » [BACnet Schedule] 按鈕後即會開啟 [BACnet Schedule] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [BACnet Schedule] 物件。

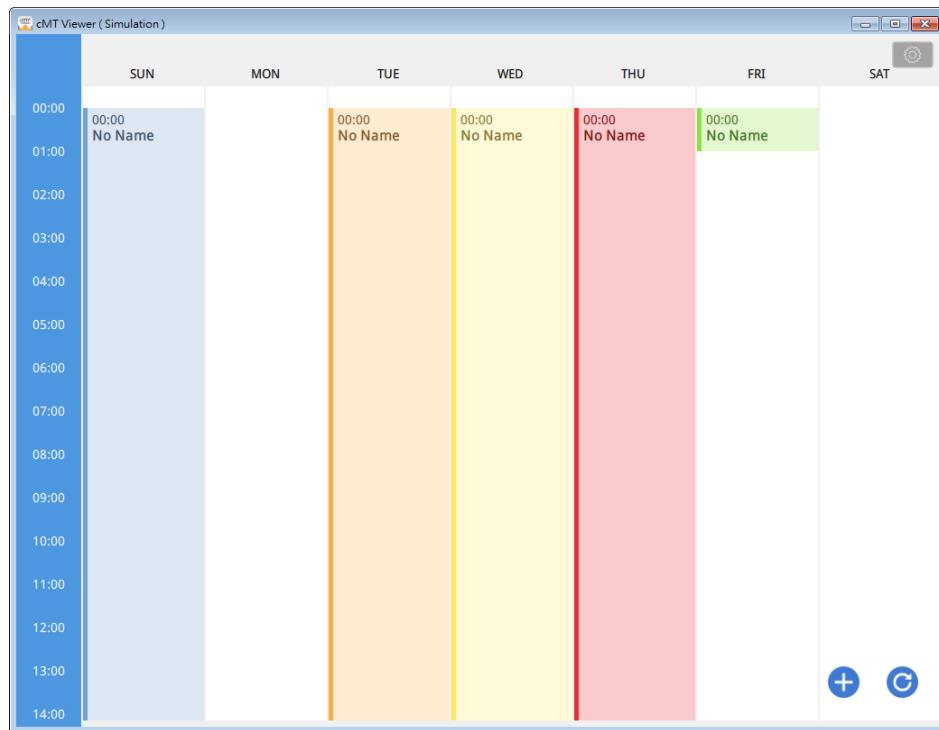
## 一般屬性設定



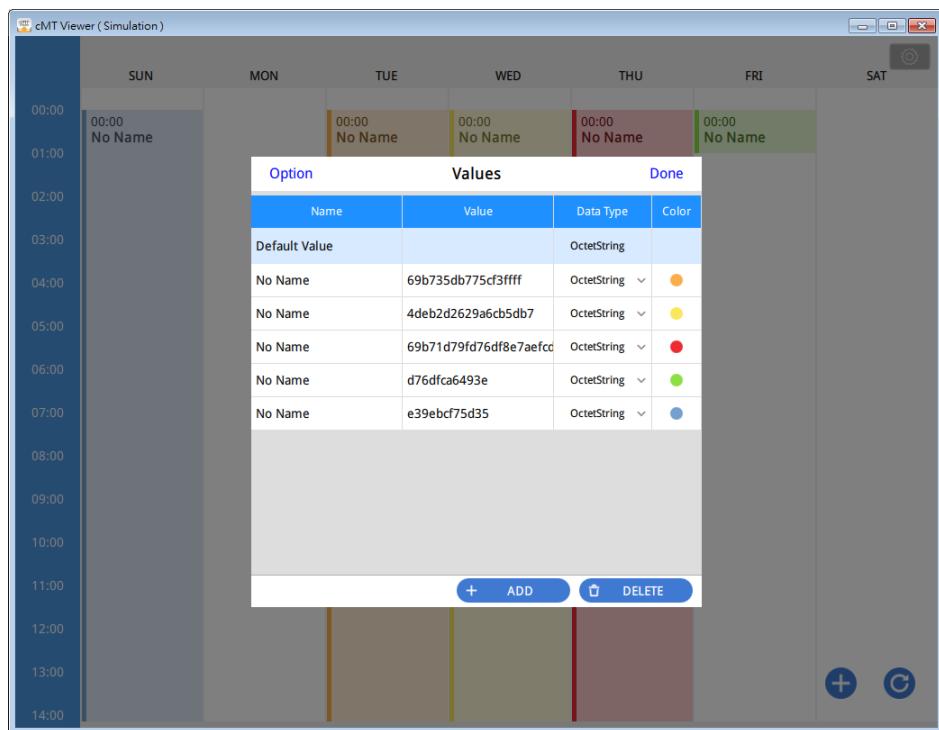
設定	描述
設備	選擇欲讀取的 BACnet/IP 設備。
位址	選擇欲讀取的 Schedule 標籤位址。
選項按鈕顯示 (cMT / cMT X 系列)	<p>設定是否顯示物件上的選項按鈕 。</p> <ul style="list-style-type: none"> <li>● 預定義：</li> <li>預先設定好是否顯示選項按鈕。</li> <li>● 動態：</li> </ul> <p style="text-align: center;"><input type="radio"/> 預定義      <input checked="" type="radio"/> 動態</p>
	<p>設備 : Local HMI</p> <p>位址 : LB</p> <p>設定一個位元位址動態決定是否顯示選項按鈕。</p>

## 範例 1

當工程檔案被下載至人機介面後，將會自動讀取 BACnet IP 裝置裡的排程設定。以下圖為例：



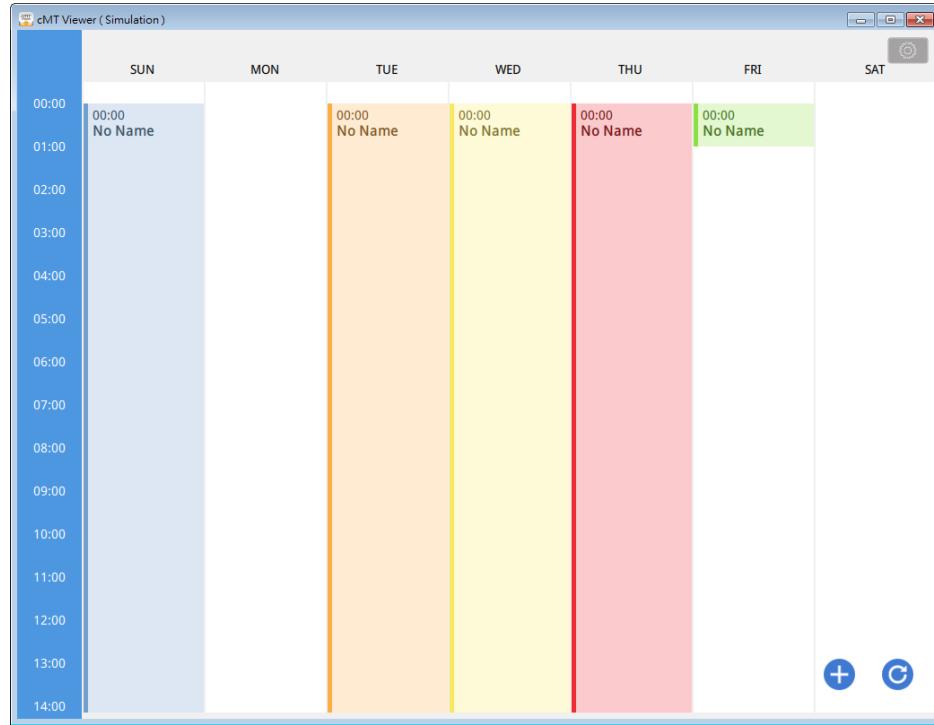
點選 ，再點選 Value Editor，可以建立數值表，包含名稱 (Name)、數值 (Value)、資料類型 (Data Type) 與顏色 (Color)。如下：



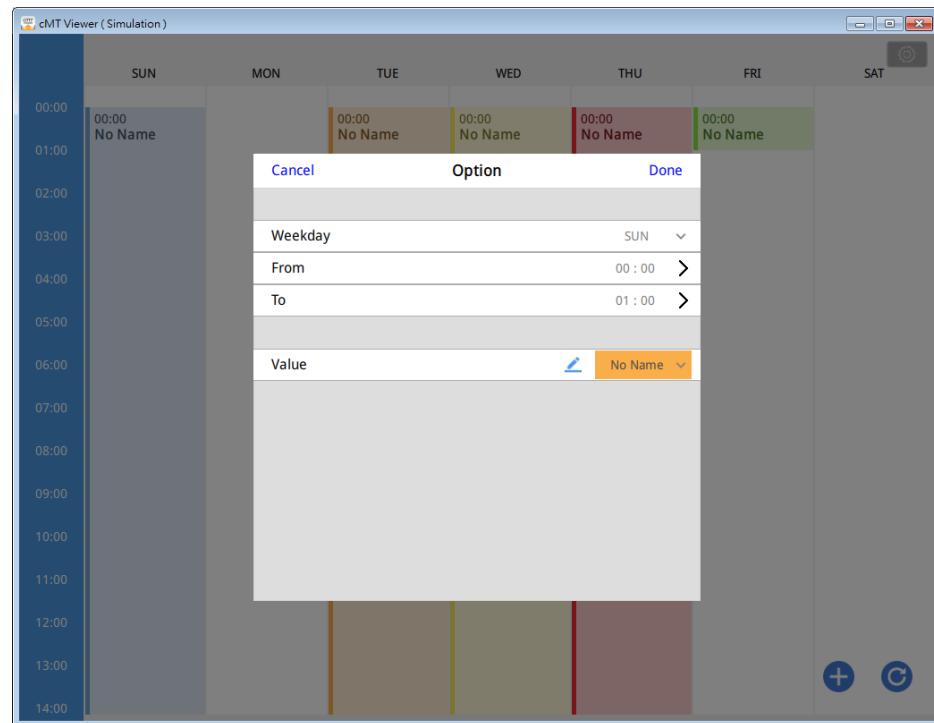
點擊右下的  可以新增一組排程時間設定。而按下  可重新自 BACnet 設備讀取排程的資

## 物件

料。



如下圖所示，一組設定中，需要設定星期幾 (Weekday)、開始時間 (From)、結束時間 (To)，和數值(Value)。Value 欄位由數值表中選擇。(點下 也可以至 Value Editor 畫面編輯數值表)



當新增時間設定完成後，數值即會被寫入 BACnet 設備中的對應排程位址中。

## 13.27. PLC 控制

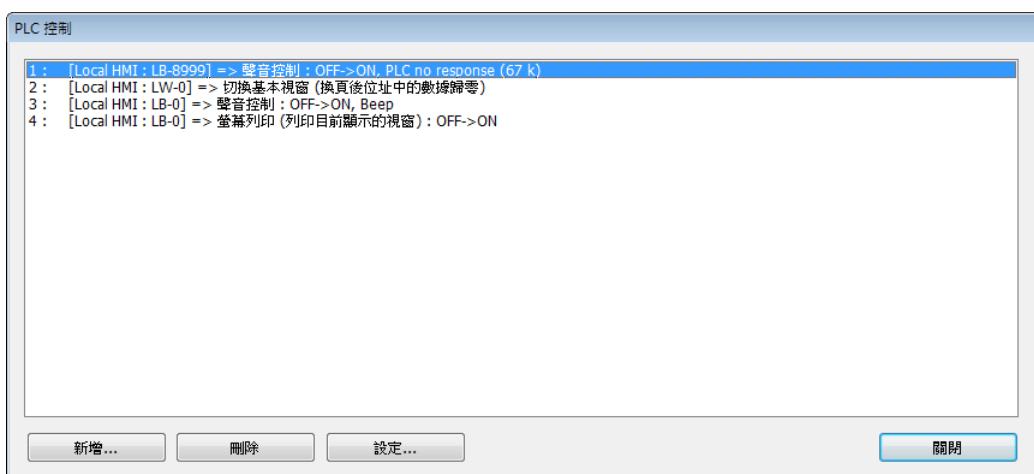
### 13.27.1. 概要

當相應的控制命令被觸發時，[PLC 控制] 物件能啟動某個特定的動作。

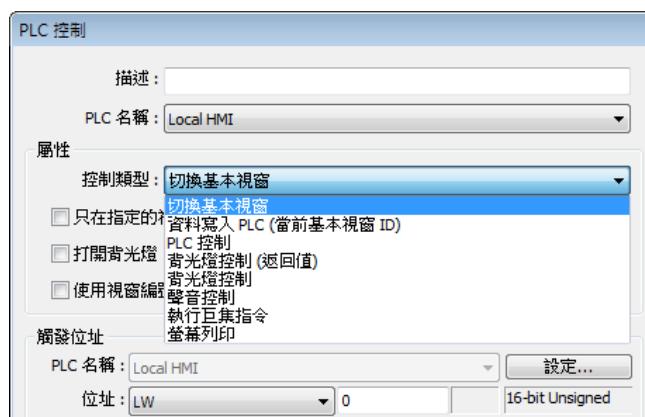
### 13.27.2. 設定



按下工作列的 [物件] » [PLC 控制] 按鈕後即會出現 [PLC 控制] 物件管理對話窗，接著可按下 [新增] 按鍵，並利用出現的 [PLC 控制] 物件設定對話窗正確設定物件的各項屬性，最後按下確定鍵即可新增一個 [PLC 控制] 物件。



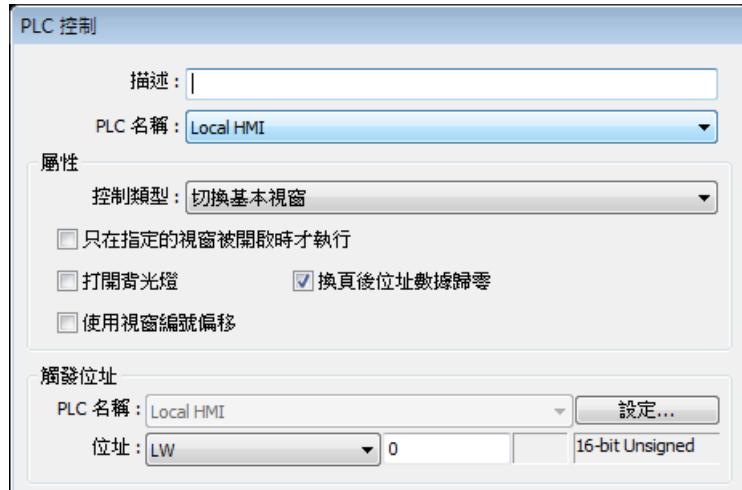
下圖為按下 [新增] 按鍵後所出現的設定對話窗。請見下面《13.27.2.1 控制類型說明》。



- cMT / cMT X 系列不支援 [PLC 控制] 功能。
- 若使用 cMT / cMT X 系列 HMI，透過 cMT Viewer 觸發 [螢幕列印] 時，檔案會儲存到 HMI 上的 USB/SD 卡裝置。若觸發 [背光燈控制] 功能，則是控制 HMI 上的背光燈開關。

### 13.27.2.1. 控制類型說明

- 切換基本視窗



設定	描述
只在指定的視窗被開啟時才執行	此切換視窗的功能將只在指定的視窗內才有作用。
打開背光燈	若啟用此選項，當背光燈為關閉狀態時，切換視窗成功後會自動開啟背光燈。
換頁後位址數據歸零	若啟用此選項，切換視窗成功後會將觸發位址中的數據歸零。如勾選 [使用視窗編號偏移]，當偏移量設定為負值時，此選項才會出現。
使用視窗編號偏移	當勾選此選項時，[觸發位址] 所指定暫存器中的數據，再加上 [視窗編號偏移]，才是實際切換的目的視窗號碼。例如：當觸發位址為 LW-0，偏移量為 -10，當 LW 0 的數值為 25 時，視窗將切換到視窗編號 15 (數值 25 + 偏移量-10)。偏移量範圍為 -1024 至 1024。

#### Note

- 當 [LB - 9017] 的狀態被設定為 ON 時，切換後的視窗編號將不再寫至特定的位址中。
- 切換基本視窗的功能。當 [觸發位址] 中的數據改變，且改變後的數據為一個有效的視窗編號時，將關閉目前的視窗並切換至 [觸發位址] 中數據所指定的視窗，並將此時切換後的視窗編號寫至 [觸發位址 + 1 (16bit)] 或是 [觸發位址 + 2 (32bit)] 中。  
例如：目前的視窗編號為 10，觸發位址為 LW-0：  
當 LW-0 中的數據由其他數據改變為 11 時，EasyBuilder Pro 除了會將基本視窗切換到視窗 11 之外，也會將 LW-1 中的數據更改為 11。  
▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。
- 當切換視窗成功時，切換後的視窗編號的寫入位址與 [觸發位址] 中設定的讀取位址、變數型態皆有關係，如下表所示：

資料型態	目的視窗編號讀取位址 (觸發位址)	切換後視窗編號的寫入 位址
16-bit BCD	位址	位址 + 1
32-bit BCD	位址	位址 + 2
16-bit Unsigned	位址	位址 + 1
16-bit Signed	位址	位址 + 1
32-bit Unsigned	位址	位址 + 2
32-bit Signed	位址	位址 + 2
64-bit Unsigned	位址	位址 + 4
64-bit Signed	位址	位址 + 4



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### ● 資料寫入 PLC (當前基本視窗)

當切換基本視窗時，會將基本視窗的編號寫至 [觸發位址] 中。如果設定了 [使用視窗編號偏移]，會將基本視窗的編號+視窗編號偏移的數值寫至 [觸發位址] 中。

#### ● PLC 控制 (eMT, iE, XE, mTV)

此項功能提供使用者利用暫存器中的數據，控制 PLC 與 HMI 之間的資料傳輸，資料傳輸方向包含四種類型，參考下表的內容：

資料傳輸類型	資料傳輸方向
1	PLC 暫存器中的數據 → HMI 上的 RW 暫存器。
2	PLC 暫存器中的數據 → HMI 上的 LW 暫存器
3	HMI 上的 RW 配方資料 → PLC 上的暫存器
4	HMI 上的 LW 暫存器 → PLC 上的暫存器

使用此項功能時，由 [觸發位址] 所設定的位址連續四個暫存器中的數據，決定資料傳輸類型、資料傳送數量、資料來源位址與資料傳送目的位址等。下表表示各暫存器中數據所表示的意義：

位址	用途	說明
[觸發位址]	存放資料傳輸類型，並決定資料傳輸的方向。	用來決定資料傳輸類型，如上表所述，共有四種類型。當暫存器被寫入新的數據時，HMI 即執行相應的傳輸，傳輸完成後會將暫存器中的數據設為 0。
[觸發位址] + 1	存放欲傳輸資料大小	單位為字組(word)。
[觸發位址] + 2	存放傳輸過程中數據來源的位址偏移量	傳輸的數據來源的起始位址為：[觸發位址] + 4 + 位址偏移量

以 OMRON PLC 為例，如果此時設定的 [觸發位址] 為 DM-100，而在暫存器 [觸發位址] +2 也就是 DM-102 中的資料為“5”，則傳輸的數據來源的起始位址為 DM-109，其中  $109 = (100 + 4) + 5$ 。

[觸發位址] + 3

存放傳輸過程中配方資料暫存器 (RW) 或者本地資料暫存器 (LW) 的起始位址

以 OMRON PLC 為例，如果此時設定的 [觸發位址] 為 DM-100，而在暫存器 [觸發位址] + 3 也就是 DM-103 中的資料為“100”，則傳輸過程中操作的 RW 或 LW 的起始位址為 RW-100 或 LW-100。

## 範例 1

假使現在需要使用 [PLC 控制] 的功能，將 OMRON PLC 中從 DM-100 起始的 16 words 的資料，傳輸到 HMI 配方記憶體 RW-200 開始的位址中，設定方法如下：

1. 首先，假設用 DM-10 起始的四個資料暫存器來控制傳輸。先建立一個 [PLC 控制] 物件，選擇類型為 [PLC 控制]，讀取位址設定為 DM-10。
2. 確定操作資料的大小和位址的偏移量。  
將 DM-11 設定為 16，表示傳輸資料的大小為 16 words；將 DM-12 設定為 86，表示數據的來源位址為 DM-100 ( $100 = 10 + 4 + 86$ )；將 DM-13 設定值為 200，表示目標位址為 RW-200。
3. 最後，依照資料傳輸的方向，設定傳輸類型。  
將 DM-10 設定為 1，表示將傳輸 PLC 暫存器中的數據到 HMI 上的 RW 暫存器中。  
如果設定 DM-10 值為 3，則傳輸方向相反。

### ● 背光燈控制 (返回值)

當 [觸發位址] 的狀態由 OFF 變為 ON 時，HMI 將打開/關閉背光燈，此時也會將 [觸發位址] 的狀態設定為 OFF。背光燈關閉時，使用者只需碰觸螢幕，背光燈即會再度打開。

### ● 背光燈控制

當 [觸發位址] 的狀態由 OFF 變為 ON 時，HMI 將打開/關閉背光燈。但因不具備 "返回值" (write back) 功能，此時並不會將 [觸發位址] 的狀態設定為 OFF。

### ● 聲音控制

當 [觸發位址] 的狀態改變符合觸發條件時，將播放預先指定的聲音檔案。

可以選擇狀態由 OFF 變為 ON 或由 ON 變為 OFF 時，播放聲音檔案。也可選擇狀態改變時 (OFF<->ON)，即播放聲音檔案。

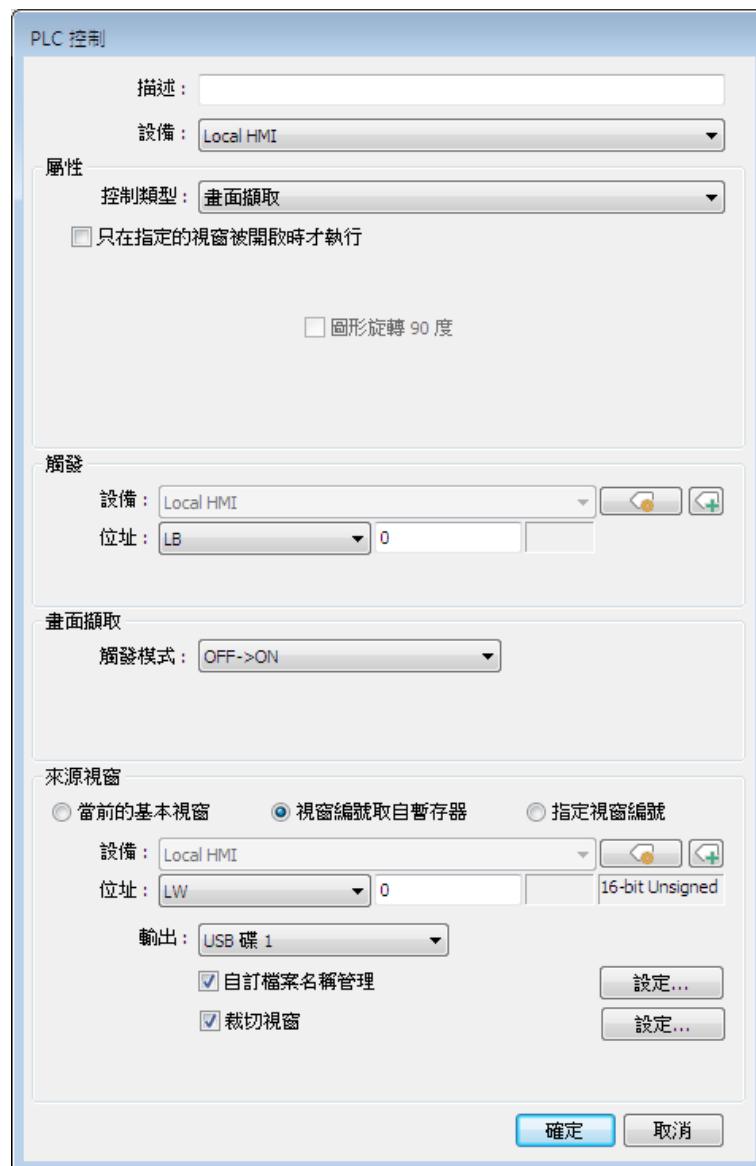
### ● 執行巨集指令

檔案中若有編輯完成的巨集指令，此選項便會出現。

當 [觸發位址] 的狀態改變符合觸發條件時，將執行指定的巨集指令。

可以選擇狀態由 OFF 變為 ON 或由 ON 變為 OFF 時，執行巨集指令。也可選擇狀態改變時 (OFF<->ON)，即執行巨集指令。或是當狀態為 ON 時即執行：只需狀態維持在 ON，即可持續執行指定的巨集指令 (最快為每 0.5 秒執行一次)。

### ● 畫面擷取



當 [觸發位址] 的狀態被觸發時，將列印指定的視窗畫面。

若裝置選擇為 [USB 碟] 或 [SD 卡]，當觸發此功能時，系統將於外接儲存裝置建立 **hardcopy** 資料夾來儲存圖片，圖片格式為 JPG，圖片名稱將由 **yymmdd\_0000** 開始編號。

若欲選擇的裝置為印表機，請於 [系統參數設定 \ HMI 屬性] 頁籤設定欲連接的印表機類型。

若欲選擇的裝置為遠端印表機，請於 [系統參數設定 \ 列印/備份伺服器] 頁籤設定相關參數。

可藉由以下三種方式指定欲列印的視窗編號：

#### 當前的基本視窗

列印目前開啟的基本視窗畫面。

#### 視窗編號取自暫存器

欲列印的視窗編號來源將取自字組位址中的數值。

#### 指定視窗編號

直接指定欲列印的視窗編號。

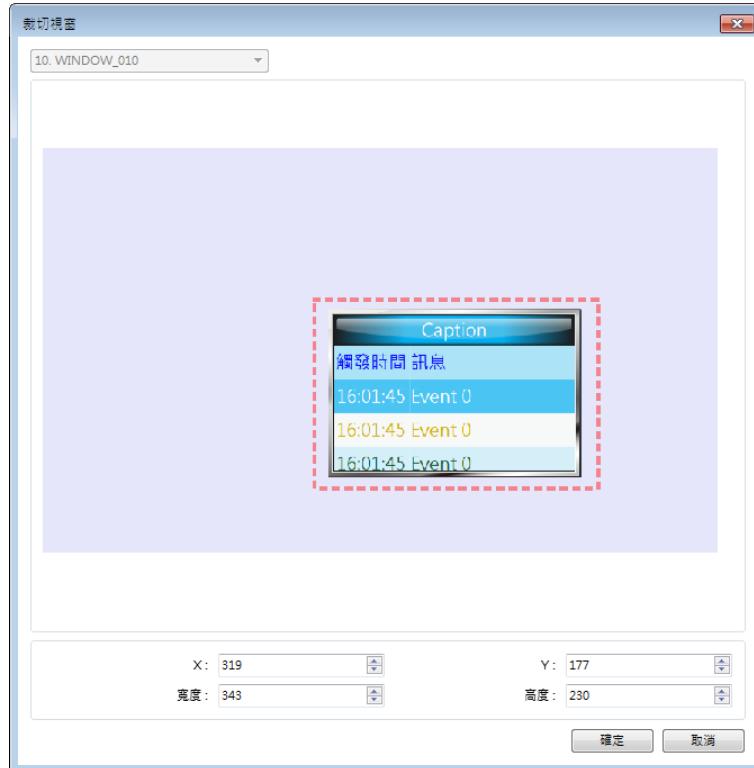
### 自訂檔案名稱管理

若勾選，可更改畫面擷取時的資料夾名稱和檔案名稱。

設定	描述
資料夾名稱	<p>可使用預設的控制碼、英數字及部分的半形符號： !@#\$%^&amp;()_+{}`-=;'. <b>動態格式</b> 若勾選，可指定一組字組位址來設定資料夾的名稱，亦可輸入時間方塊的控制碼來帶入系統時間，字組長度為 1~25。 <b>注意:</b> 最多可建立 10 個層級的資料夾，多餘的層級將被忽略。</p>
檔案名稱	<p>可使用預設的控制碼、英數字及部分的半形符號： !@#\$%^&amp;()_+{}`-=;'. <b>動態格式</b> 若勾選，可指定一組字組位址來設定檔案的名稱，亦可輸入時間方塊的控制碼來帶入系統時間，字組長度為 1~25。 <b>注意:</b> 當相同名稱的檔案已經存在時，系統將以 “_0001” 依序命名，直到 “_9999” 為止，便忽略之後所觸發的畫面擷取。 假設已有檔案名稱 “A.jpg”， “A_0001.jpg”， “A_0003.jpg”，當再次以相同名稱觸發畫面擷取時，將會優先產生 “A_0002.jpg”，再產生 “A_0004.jpg”， “A_0005.jpg”...。</p>

### 裁切視窗

若勾選，畫面擷取可自訂輸出畫面的範圍。



#### Note

- 當指定被列印的視窗不是當前視窗時，系統提供背景列印。
- 指定背景視窗時，該視窗之 [直接視窗] 或 [間接視窗] 將不會被列印。
- 若使用動態格式來命名資料夾或檔案名稱時，當輸入不支援的符號時，系統將以底線符號“\_”取代顯示。
- 若使用動態格式，當沒有輸入任何字串即觸發畫面擷取時，系統將以預設的路徑儲存圖片，即為 hardcopy\yyymmdd\_0000.JPG。
- 當狀態觸發後，cMT/cMT X 機型會自動復歸該位元狀態，而 eMT/iP/iE/XE/mTV 機型則保持該位元之狀態。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.28. 排程

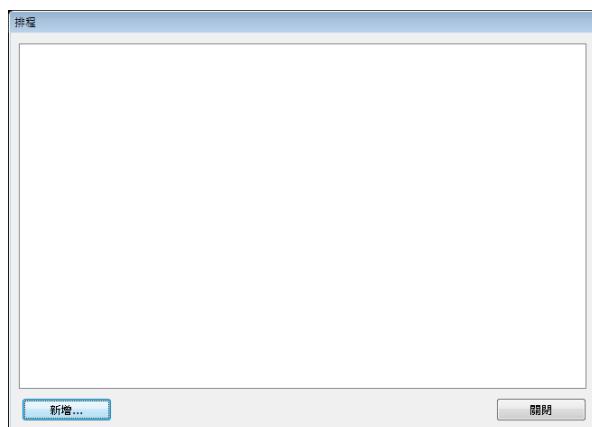
### 13.28.1. 概要

[排程] 可用來設定時刻表，將位元設為 ON / OFF 或在字組位址寫入數值，適合用來規劃一週內的例行程序。

### 13.28.2. 設定



按下工作列的 [物件] » [時間相關] » [排程] 按鈕後即會出現對話窗，按下 [新增] 鍵，即可進入排程的設定頁。



先介紹兩個範例再詳細說明各項功能：

#### 範例 1

馬達 (位址：LB - 100) 從星期一直運轉到星期五，時間由每天上午 9 點到下午 6 點。設定程序為在起始時間 (早上 9 點) 將位址 LB-100 設為 ON，在結束時間 (下午 6 點) 將位址 LB-100 設為 OFF。

1. 按下工作列上的 [排程] 按鈕後即會出現對話窗，按下 [新增] 鍵，即可進入排程的設定頁。
2. 選擇 [一般屬性] 頁籤，選定 [行動模式] 為 [位元設為 ON]，並設定 [目標位址] 為 LB-100。



3. 選擇 [時間設定] 頁籤，接著選擇 [常數]。



4. 設定 [開始]。將時間設為 9 點 0 分 0 秒，接著勾選星期一到星期五，不勾選 [設定為單一日期]。
5. 設定 [結束]。勾選 [啟用結束行動]，將結束時間設定為 18 點 0 分 0 秒。
6. 按下 [確定] 鍵後，即可看到排程的日程表。

## 範例 2

從星期一到星期五，在起始時間 8 點把溫度設定值 90 度寫入字組位址 LW-100，此時系統進入運轉模式。在結束時間 17 點把溫度設定值 30 度寫入字組位址 LW-100，此時系統進入等待模式。

1. 按下工作列上的 [排程] 按鈕後即會出現對話窗，按下 [新增] 鍵，即可進入排程的設定頁。
2. 選擇 [一般屬性] 頁籤，選定 [行動模式] 為 [字組寫入]。設定 [目標位址] 為 LW-100。
3. 選擇 [常數]，設定 [開始欲寫入數值] 為 90。

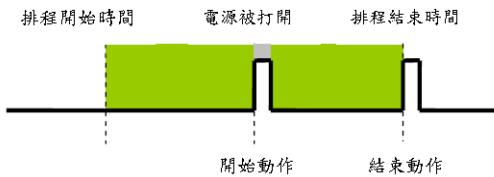
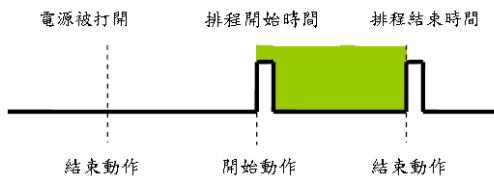


4. 選擇 [時間設定] 頁籤，接著選擇 [常數]。
5. 設定 [開始]。將時間設為 8 點 0 分 0 秒，接著勾選星期一到星期五，不勾選 [設定為單一日期]。
6. 設定 [結束]。勾選 [啓用結束行動]，將結束時間設定為 17 點 0 分 0 秒。
7. 選擇 [一般屬性] 頁籤，設定 [結束欲寫入數值] 為 30。
8. 按下 [確定] 鍵後，即可看到排程的日程表。

## 一般屬性設定



設定	描述
電源開啟時執行開啟 / 結束動作	當電源打開時，執行已設定的動作。 ● 啓用時 假如 HMI 的電源在排程區間內被打開，則開始動作會被執行。 假如 HMI 的電源在排程區間外被打開，則結束動作會被執行。

**在排程區間內****在排程區間外****● 停用時**

假如電源打開時晚於排程開始時間，則開始動作不會自動執行。然而結束動作依然會執行。假如結束動作未被設定，由於無法正確地判定排程區間，結束動作將不會被執行。

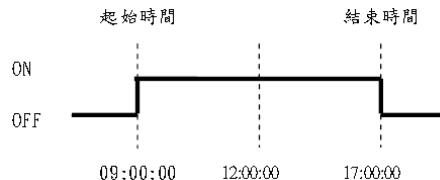
**行動模式**

選擇在設定的時間要操作的類型。

**位元設為 ON**

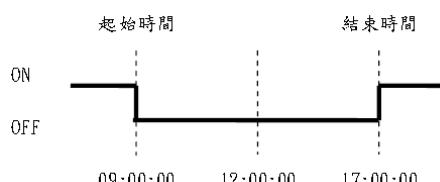
在排程開始時，將指定位元位址的狀態設為 ON；在排程結束時，將指定位元位址的狀態設為 OFF。

例如：起始時間：09:00:00 結束時間：17:00:00

**位元設為 OFF**

在排程開始時，將指定位元位址的狀態設為 OFF；在排程結束時，將指定位元位址的狀態設為 ON。

例如：起始時間：09:00:00 結束時間：17:00:00

**字組寫入**

在排程開始時，將 [開始欲寫入數值] 寫入指定字組位址；在排程結束時，將 [結束欲寫入數值] 寫入指定字組位址。使用者可以直接輸入常數，或是用 [位址] 模式設定數值。若使用 [位址] 模式，則 [控制位址] 內的數值為開始欲寫入的數值，[控制位址 + 1] 內的數值為

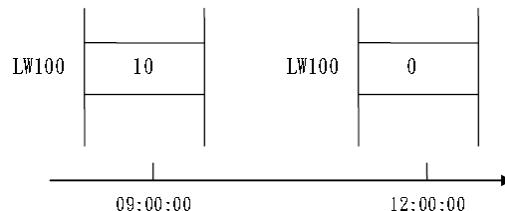
結束欲寫入的數值。

例如：字組寫入值設定位址：`LW-100`

起始時間：`09:00:00` 結束時間：`17:00:00`

使用常數：開始欲寫入數值：`10` 結束欲寫入數值：`0`

使用位址：若控制位址設定為 `LW-n`，則在 `LW-n` 內輸入 `10`，在 `LW-(n+1)` 內輸入 `0`。

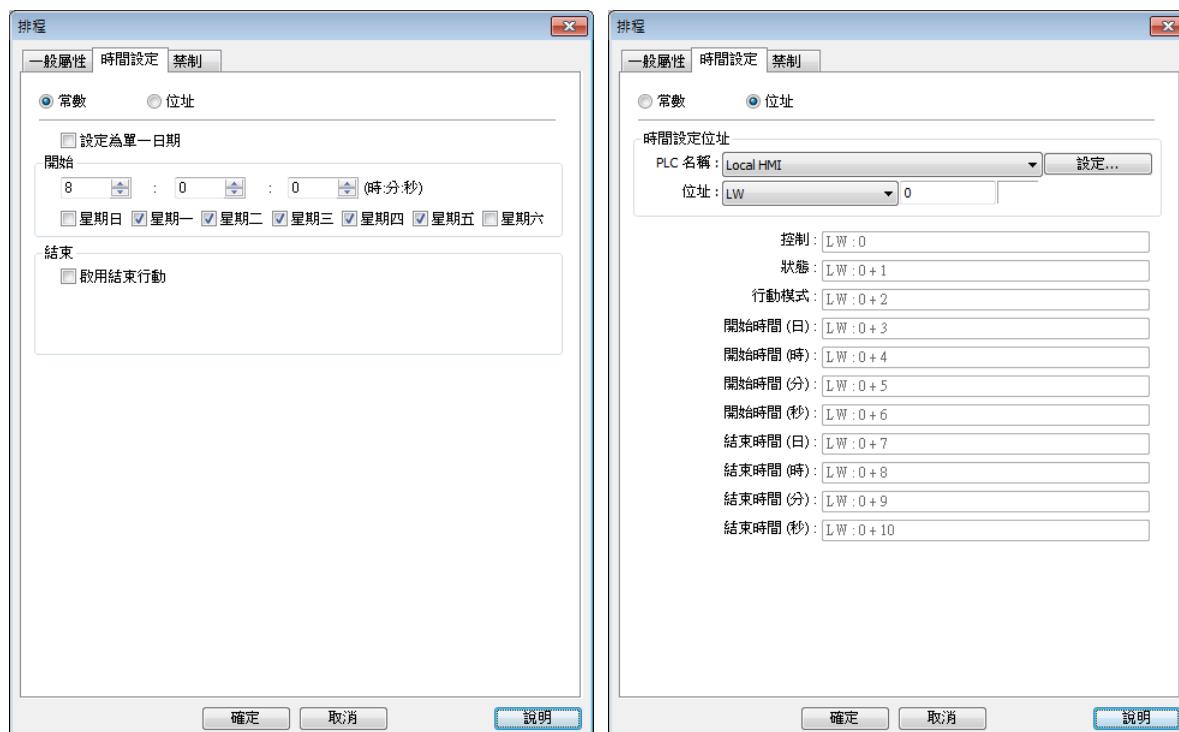


### Note

- 必須在 [時間設定] 頁籤中勾選 [啓用結束動作] 才能使用 [結束欲寫入數值]。

## 時間設定

選擇設定起始時間和結束時間的方法。[常數] 可指定一個固定的時間和日期，而 [位址] 將指定特定位址作為時間和日期的資訊。



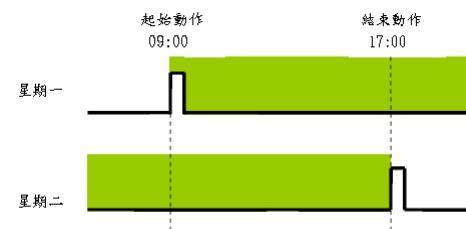
- 常數

### [設定為單一日期]

啓用時

## 物件

動作可在一周內指定的日期及時間被執行。當啟用後，則必須設定所有日期有相同的動作開始時間及結束時間。

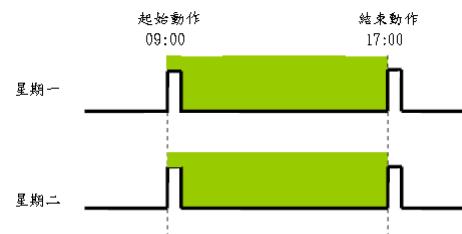


### Note

- 必須輸入起始時間和結束時間。
- 不能在起始時間和結束時間欄位裡輸入一模一樣的時間和日期。

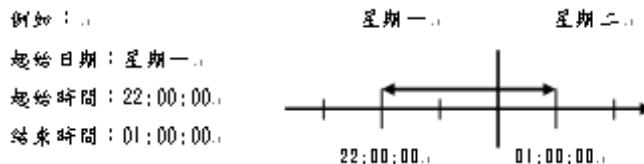
### 停用時

排程時間必須被限定在一天之內 (起始時間和結束時間必須在 24 小時內)。



### Note

- 不能在起始時間和結束時間欄位裡輸入一模一樣的時間和日期。
- 此種時間排程只適用於一天之內的排程，因此如果所鍵入的結束時間早於起始時間，則結束動作將會等到下一天才會執行。



### ● 位址

開始時間、結束時間、執行命令、執行結果皆由指定位址控制。使用者只需定義 [時間設定位址]，其餘的 11 個控制字組會自動產生並列示出來。圖上之資料長度皆以 16 位元為例；當指定之暫存器為 32 位元時，只有較低的 16 位元產生作用，並請將較高的 16 位元固定為 0。

以下說明各位址之使用說明：

#### 控制 (時間設定位址 + 0)

當 [更新時間位元] (見下圖) 被偵測為 ON ( $0 \rightarrow 1$ ) 時，則讀出 [模式]、[開始時間] 和 [結束時間]。



位元 00 : 更新時間位元(0:無動作, 1:讀取排程時間資料)

### Note

- HMI 並不會定期地讀取時間設定位址的 [模式] (位址 + 2) 到 [結束時間(秒)] (位址 + 10) 裡的資料。所以，當排程時間資料改變時，請務必把 [控制] 中的 [更新時間位元] 設為 ON ( $0 \rightarrow 1$ )。



### 狀態(時間設定位址 + 1)

在 [控制] 中的時間資料讀取完成之後，HMI 將會把 [時間讀取完成位元] 設為 ON ( $0 \rightarrow 1$ )。

同樣地，若輸入的時間資料不正確，[錯誤通知位元] 將會同時被設為 ON ( $0 \rightarrow 1$ )。

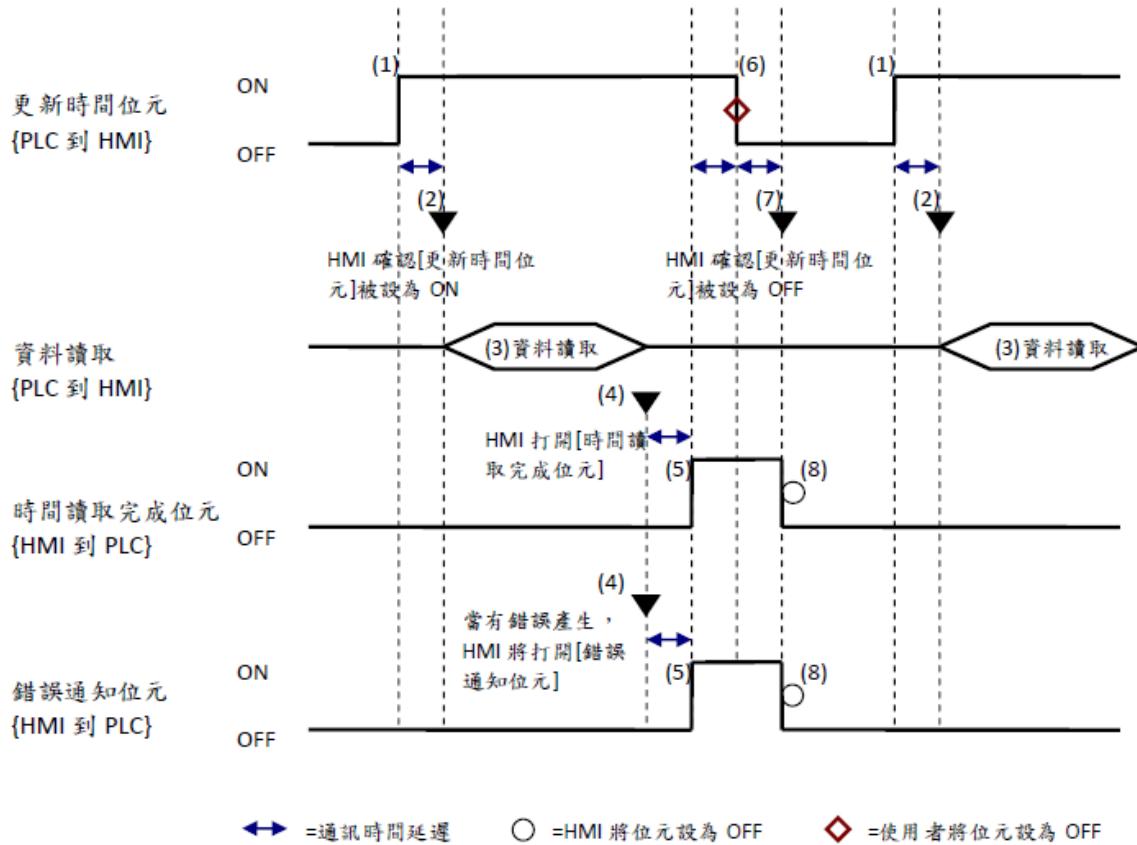
15	02 01 00 位元
保留 (0 固定)	0 0

位元 00： 時間讀取完成位元(0：還沒開始或是正在讀取時間資料；1：時間資料讀取完成)

位元 01： 錯誤通知位元(0：時間資料被正確更新；1：時間資料中包含錯誤)



- 一旦發現 [時間讀取完成位元] 被觸發，請務必把 [控制] 中的 [更新時間位元] 設為 OFF( $1 \rightarrow 0$ )。一旦這個位元被設為 OFF( $1 \rightarrow 0$ )，則 [狀態] 中的 [時間讀取完成位元] 及 [錯誤通知位元] 將同時被設為 OFF( $1 \rightarrow 0$ )。



### 模式(時間設定位址 + 2)

啓用或停用 [結束時間動作設定] 和 [單一日期指定模式]。不管 [結束時間動作設定] 的狀態如何，所有的時間資料 ([時間設定位址] 中的 11 個字組位址) 都會被讀取。

15	02 01 00	位元
保留 (0 固定)	0   0	

位元 00 : 結束時間動作設定(0：停用；1：啓用)

位元 01 : 單一日期指定模式(0：停用；1：啓用)

### Note

- 若 [結束時間動作設定] 輸入 0(停用)，仍會讀取結束時間資料但忽略其內容。
- 若 [單一日期指定模式] 輸入 1(啓用)，請確認是否已輸入開始及結束時間資訊。假如有 2 個以上的開始/結束日期位元被同時設為 ON，則會產生錯誤。

### 開始 / 結束日期 (開始日期 : 時間設定位址 + 3 ; 結束日期 : 時間設定位址 + 7)

指定觸發開始/結束動作的日期。

15	07 06 05 04 03 02 01 00	位元
保留 (0 固定)	Sat   Fri   Thu   Wen   Tue   Mon   Sun	

位元 00 : 星期日(0：無；1：指定)

位元 01 : 星期一(0：無；1：指定)

位元 02 : 星期二(0：無；1：指定)

位元 03 : 星期三(0：無；1：指定)

位元 04 : 星期四(0：無；1：指定)

位元 05 : 星期五(0：無；1：指定)

位元 06 : 星期六(0：無；1：指定)

### 開始/結束時間 (開始時間 : 時間設定位址 + 4 到 + 6 ; 結束時間 : 時間設定位址 + 8 到 + 10)

時 : 0 - 23 分 : 0 - 59 秒 : 0 - 59

假如所指定的值超出上面的範圍，將會產生錯誤。

### Note

- 使用者所輸入的時間資料應為 16 位元無正負號 (unsigned) 格式，系統不接受 BCD 格式的時間資料。
- 當時間設定使用 [位址模式] 時，若 HMI 重啟後，仍需使用 [更新時間位元] 來載入排程時間。
- 即使是用 RW 位址，HMI 重啟後仍需使用 [更新時間位元] 來載入排程時間。建議可於公共視窗放置位元狀態設定物件，開關類型設為“當視窗開時設 ON”來自動載入設定，達成重啟 HMI 仍可以保持前一次設定的目的。
- 結束時間取決於 [模式](位址+2) 設定。同樣地，[結束時間動作設定] (位元 00)有效與否取決於 [單一日期指定模式] (位元 01)的使用。

單一日期指定模式	使用	不使用
結束時間動作設定	使用	不使用

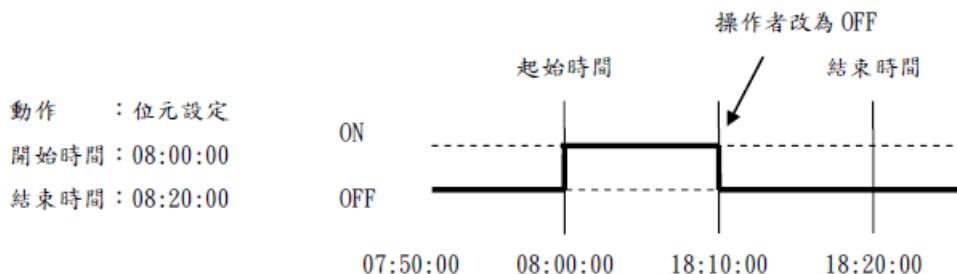
## 禁制



啓用時，在執行開始動作前 HMI 將讀取此位元狀態，若為 ON，則略過此次開始及結束動作(若存在)；反之則正常執行設定動作。

### Note

- 最多可註冊 64 個 [排程] 物件。
- 時間排程的特性為一次動作。當開始時間到達時，特定的設備位址只會被寫入一次，這個寫入的動作將不會重複。



- [開始/結束寫入數值] 和 [禁制位元] 只會在執行開始動作前讀取一次。所以當開始動作執行後，就算再去改變 [禁制位元] 狀態或 [結束寫入數值] 都無法改變結束動作的執行與否及寫入數值。另外，為了讀取 [開始/結束寫入數值] 和 [禁制位元] 資料，起始動作可能因資料通訊而有少許延遲。
- 當使用者改變 HMI 的系統時間，系統將會重新確認排程中起始與結束時間的範圍。假如編輯的物件位於新範圍中，則開始動作會被執行。假如結束動作未被設定，系統無法確認新範圍，則這個動作將不會被執行。
- 當相同的起始和結束時間出現在多個排程物件中，將依其編號由小到大順序被處理。
- 當 [時間設定] 指定為 [位址]，系統將會定期去讀取 [控制] 位址，時間長短視系統忙碌程度而定。
- 當 [時間設定] 指定為 [位址]，且指定開始時間和結束時間超過時間合法的範圍，則設定的時間可能不會正確地運作。注意：不能使用 BCD 當成輸入值。

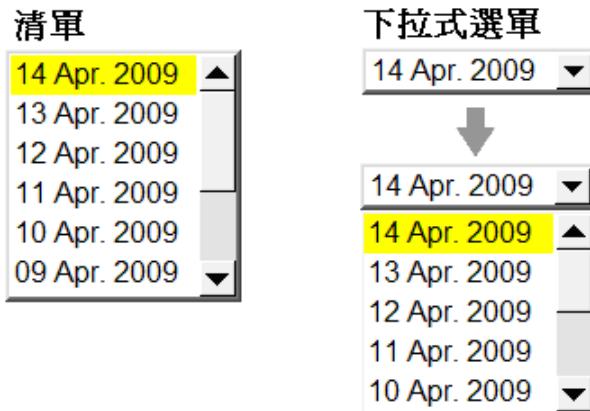
- 當 [時間設定] 指定為 [位址]，排程物件直到第一次成功更新時間資料，才開始運作。
-  請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.29. 項目選單

### 13.29.1. 概要

[項目選單] 物件可以顯示多樣項目成一列表，使用者可以藉此檢視並選擇。一旦使用者選擇了某一項目，相對應的項目數據將被寫入到字組暫存器。

[項目選單] 有兩種顯示模式：[清單] 和 [下拉式選單]。清單可以完整顯示所有的項目，並把目前所選擇的項目標示出來。此外，下拉式選單在一般情況下只顯示目前所選擇之項目。但是當使用者點選下拉式選單時，系統則會列出所有完整項目(類似於清單的顯示法) 如下所示：



### 13.29.2. 設定



按下工作列的 [物件] » [項目選單] 按鈕後即會開啟 [項目選單] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [項目選單] 物件。

## 項目選單設定

eMT、iE、XE、mTV 系列



cMT / cMT X 系列



### 設定

### 描述

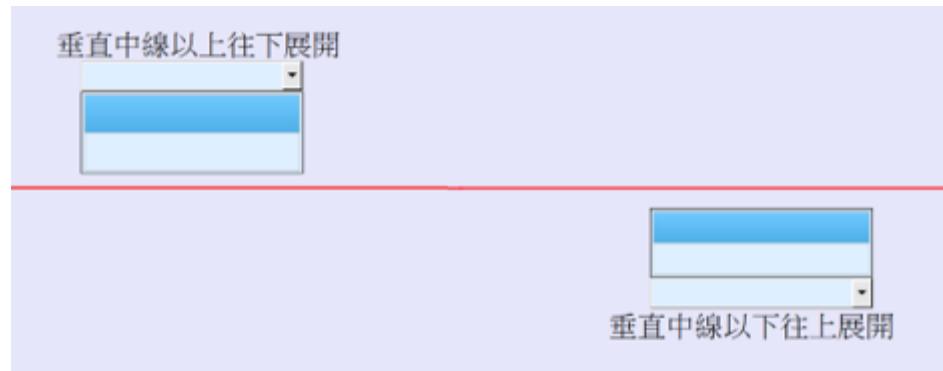
#### 屬性

**模式**：可選擇 [清單] 或 [下拉式選單]。

**背景顏色**：設定背景的顏色。

**選擇顏色**：設定選擇的項目的背景顏色。

**方向**：選擇 [自動]，則選單內容會根據物件置放位置，並以視窗的垂直中線劃分。當物件置於垂直中線以上，選單內容一律往下展開，反之，置於垂直中線以下，選單內容一律往上展開。



選擇 [往下]，則選單內容會固定向下展開。選擇 [往上]，則選單內容會固定向上展開。

注意：若物件放置於視窗邊緣處且選擇往邊緣處展開，將造成選單內容顯示被壓縮。

**類型**：cMT / cMT X 系列的 [下拉式選單] 專用，選擇 [標準]，則物件外觀為 cMT / cMT X 樣式，選擇 [簡易]，則物件外觀為 iE/eMT/XE 樣式。若將 iE/eMT/XE 的工程檔案的機型改為 cMT / cMT X 系列，物件外觀的預設類型為 [簡易]。

**字型尺寸**：選擇 [與標籤尺寸相同] 時，下拉式選單中的文字尺寸會參照 [標籤] 分頁中的文字尺寸顯示。若選擇 [固定尺寸] 時，下拉式選單中的文字則會使用系統預設的尺寸顯示。

**欄位高度**：下拉式選單中項目欄位的高度。

---

**監看位址**

系統會將已選擇項目的相對應數據寫入 [監看位址] 中。

**當按鈕鬆開才發出指令**

若啟用，當按鈕鬆開時才會將指定的數據寫入 [監看位址] 中。

---

**寫入成功後傳送通知**

當寫入 PLC 的動作成功後，將指定位元暫存器的狀態設為開 / 關。

---

**Note**

- cMT / cMT X 系列的項目資料來源不支援 [當按鈕鬆開才發出指令]。
- 基於安全考量，在 HMI 上顯示項目選單的選項，有最小 7mm 的保護高度，避免使用者誤觸其他選項。

### 13.29.2.1. 項目資料來源說明

#### ● 預設

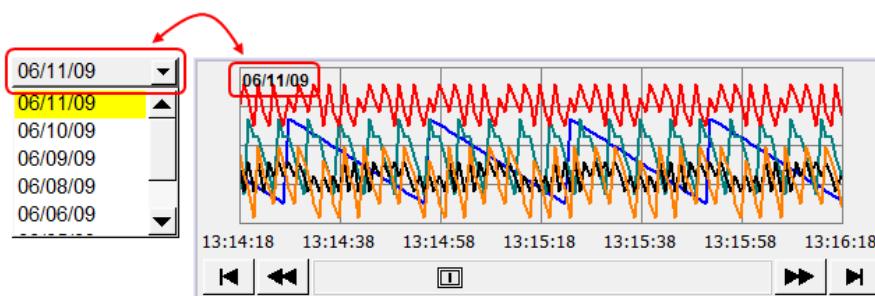
顯示的選項由使用者在 [狀態設定] 分頁中手動輸入。

可調整欲使用的 [項目數]。每一個項目表示一個狀態並會顯示在列表上，且相對應的數值可被寫入至 [監看位址]。

#### ● 歷史數據日期



可與歷史資料顯示物件搭配使用，例如 [趨勢圖]、[歷史數據顯示]、[事件顯示] 物件。當在項目選單上選擇一日期後，歷史資料顯示物件會顯示其對應日期的資料。顯示方式如下圖所示：



設定	描述
類型	可選擇 [事件登錄] 或是 [資料取樣]。
日期	共有 8 種日期模式可選擇，YYYY 代表四位數年份 (例如 2012)、YY 代表二位數年份 (例如 12)、MM 代表月份、DD 代表日期。
資料取樣	若 [類型] 選擇 [資料取樣]，需在 [資料取樣] 設定欲顯示的資料取樣物件。一般來說，選擇與之搭配的趨勢圖的歷史模式或歷史數據物件相同即可。
啟用 [刪除歷史檔案] 功能	如勾選此功能，可以設定一個控制位址。將此控制位址設定為 1，即可刪除該日的歷史資料。

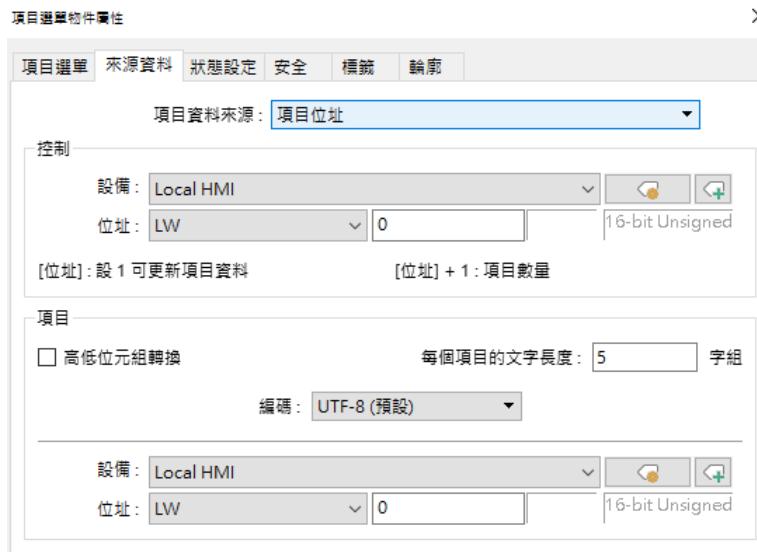
#### Note

- 當選用歷史數據來源(歷史資料日期)，由於系統將自動讀取歷史檔案並產生日期資料，故狀態設定中使用者不需再填寫。
- 使用者可以在狀態設定中設定當項目選單進入錯誤狀態時，項目選單將顯示的資料。

項目	數據	項目資料
0 (error)		Error!!!

### ● 項目位址模式

[項目位址] 模式可載入 [項目位址] 的文字並將其文字顯示於項目選單上。當選擇 [項目位址] 模式後，下方會出現 [控制位址] 和 [項目位址]，如下圖所示：



設定	描述
控制	<b>[位址]</b> : 若將此位址所指定的暫存器中數據設定為 1，將更新物件所顯示的項目為 [項目位址] 的內容。更新完成後暫存器中的數據會被恢復為 0。 <b>[位址] + 1</b> : 此位址中的數據用來設定項目的個數。
項目	設定用來存放項目內容的起始位址。
UNICODE	選項的內容使用 UNICODE 文字，例如中文字。
每個項目的文字長度	每一個項目的文字長度，單位是字組。

### Note

- 使用到的 UNICODE 文字必須先用 [文字] 物件輸入好，EasyBuilder Pro 才會預先編譯所需要的字型檔案，在下載時一併存放到 HMI 上，如此才能正確顯示 UNICODE 文字。
- (項目的個數) × [每個項目的文字長度] 不得超過 4096。
- 若選擇 [項目位址] 模式，系統將自動取消 [狀態設定] 頁面的設定欄位。
- 當切換視窗後，項目位址的內容將會移除。若希望切換視窗後仍可保持內容，可以透過多狀態設定物件的“當視窗開啟時設定”來達成自動更新項目位址內容。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### ● 使用者帳號

當啟用 [進階安全模式] 後，則資料來源將有 [使用者帳號] 選項。此時的項目選單會顯示使用者的名稱。



設定	描述
排序	決定使用者帳號的排序方式。
顯示	若勾選 [權限]，將顯示各個用戶的使用權限。若勾選 [隱藏用戶]，將顯示已隱藏的使用者資料。

#### Note

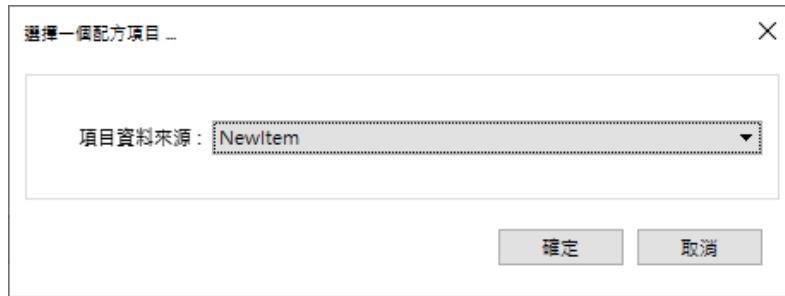
- 使用者索引的位址為 [進階安全模式] » [使用者密碼] » [進階安全模式] 中的 [控制位址+2 (LW-n + 2)]。
- cMT / cMT X 系列不支援 [權限] 選項。

### 狀態設定

此設定頁顯示所有狀態的項目、文字和數值，如果要改變項目數，請點選 [項目選單] » [屬性] » [項目數]。



設定	描述
項目	系統會列出目前所有使用的項目，每一個項目表示一個狀態並且會顯示在列表。此欄為唯讀。
數據	<p>使用者可為每一個項目設定數值，但須注意：</p> <p>讀取監看位址：如果系統偵測到 [監看位址] 的數值有改變，物件會根據其數值並選擇第一個吻合的項目。如果沒有項目吻合，將跳至錯誤狀態；如果已經設定錯誤通知位元，該位元會被觸發。</p> <p>寫入監看位址：當使用者選擇某項目，系統將數據寫入至 [監看位址]。</p>
項目資料	使用者可為每一個項目設定顯示文字，項目選單物件將顯示所有項目的文字在列表上供使用者檢視和選擇。
從配方資料庫 記錄匯入項目 資料	<p>當項目選單的 [監看位址] 選擇 Recipe-Selection 時才會啟用此功能。</p> <p>點選 [從配方記錄匯入項目資料] 會開啟 [配方資料庫記錄] 設定，於 [項目資料來源] 選擇資料來源後，該欄的所有資料將自動匯入到 [項目選單]。</p>



匯入時，會根據配方資料的數目自動調整 [項目選單] 物件的 [項目數]。

匯入後，若 [配方資料庫記錄] 物件的內容有所改變也不會影響項目選單的內容。例如，當在 [配方資料庫記錄] 中修改配方資料時，已匯入完畢的 [項目選單] 的項目資料不會隨之改變。

#### 錯誤狀態

錯誤狀態的文字只能應用於 [下拉式選單] 模式，[清單] 模式無法使用錯誤狀態文字。

在錯誤狀態發生時，[清單] 模式將不會選取任一個項目來表示錯誤狀態，而 [下拉式選單] 模式則會顯示錯誤狀態的文字。

例如，當 [項目數] 設 8 時，項目編號 8 即為錯誤狀態 (因為第 1 個項目是編號 0)。

#### 設為預設值

將所有項目數據還原為預設值，例如：將項目 0 的數據還原成 0，項目 1 的數據還原為 1...等等。

#### 錯誤通知

當項目選單偵測到不合法的數據寫入時，會觸發指定暫存器的狀態 [開] 或 [關] 以示警告。此外，使用者亦可搭配其他物件，例如 [事件登錄]、[報警條]、[彈出視窗] 來提示錯誤。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.30. 計時器

### 13.30.1. 概要

計時器相當於一計時開關，可做為延時開關、脈衝開關及累加式延時開關，其包含下列六項變數：

計時器變數	變數類型	描述
輸入位元 (IN)	位元變數	計時器的總開關
測量位元 (TI)	位元變數	計時開始時設 ON
輸出位元 (Q)	位元變數	計時結束後啟動相關設定
預設時間 (PT)	字組變數	設定計時器時間數值
已計時間 (ET)	字組變數	顯示計時器目前已計時間
重置位元 (R)	位元變數	將目前計時器已計時間 (ET) 歸零

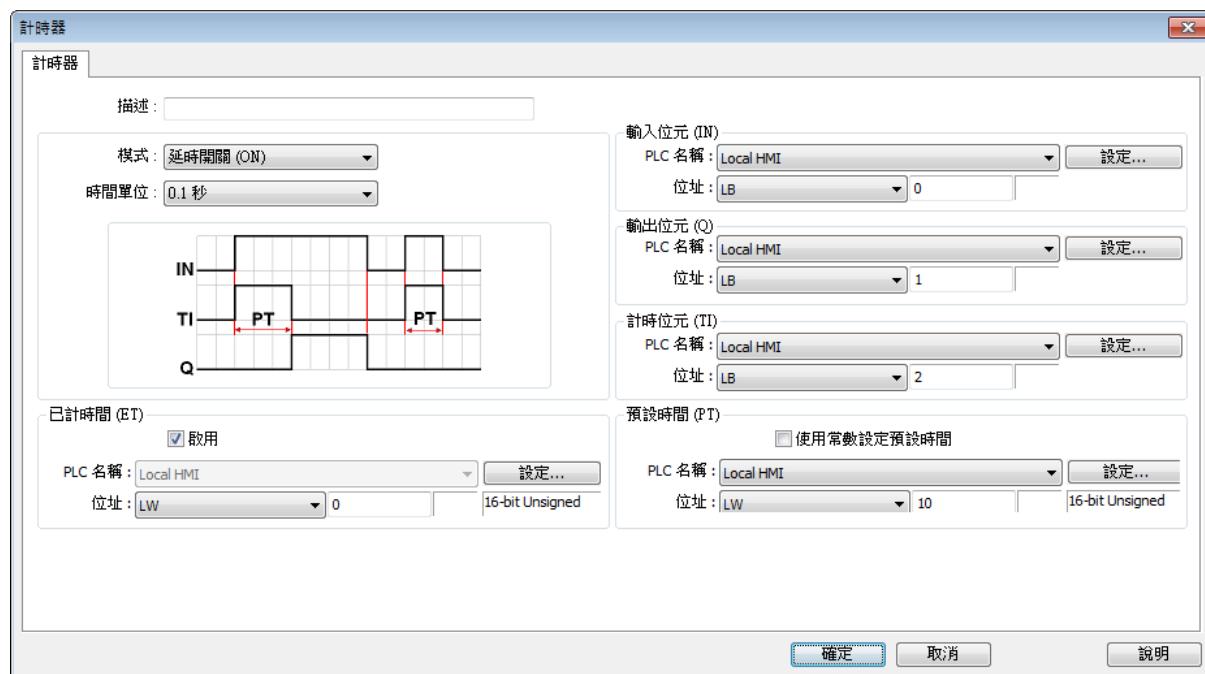
[計時器 (視窗)] 物件可以放在視窗內，僅在該視窗可觸發計時器計數。

[計時器 (背景)] 則是無論目前 HMI 顯示任何視窗，只要輸入位元為 ON，即會進行計時器計數。

### 13.30.2. 設定

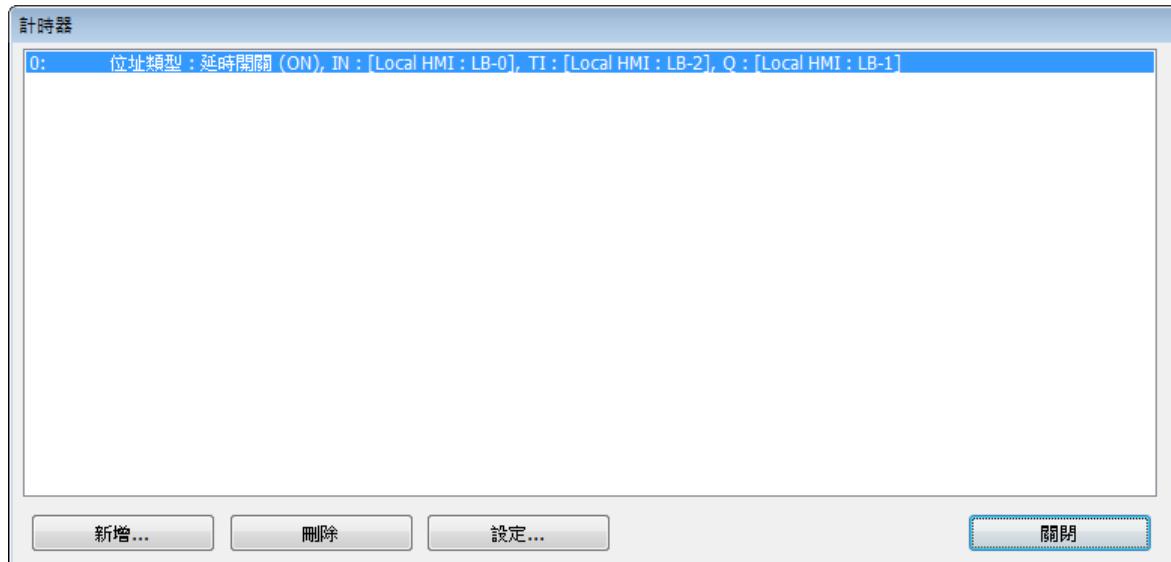


按下工作列的 [物件] » [時間相關] » [計時器] 圖示，其 [計時器] 物件屬性對話窗顯示如下。



### Note

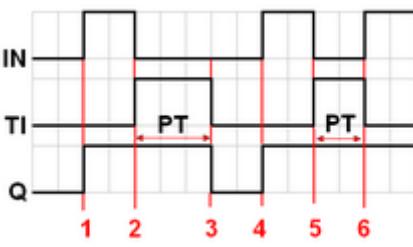
- [使用常數設定預設時間] 僅適用於 cMT / cMT X 系列。
- 若使用 cMT / cMT X 系列的 [計時器 (背景)]，點選 [計時器 (背景)] 圖示，會先開啟計時器管理視窗。點選 [新增] 可建立 [計時器] 物件。



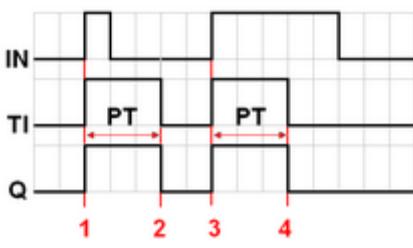
#### ● 延時開關 (ON)

電位圖	暫存器
	輸入位元 (IN)：計時器的總開關。 測量位元 (TI)：計時開始設 ON。 輸出位元 (Q)：計時結束後設 ON。 預設時間 (PT)：設定計時器時間數值。 已計時間 (ET)：顯示計時器目前已計時間。
說明 (參照上圖)	
<b>時段 1：</b> 輸入位元 IN 設 ON 時，測量位元 TI 開啟，已計時間 ET 開始計數，輸出位元 Q 保持 OFF。 <b>時段 2：</b> 當已計時間 ET 等於預設時間 PT 時，測量位元 TI 被關閉，同時輸出位元 Q 被開啟。 <b>時段 3：</b> 輸入位元 IN 設 OFF 時，輸出位元 Q 被關閉，同時已計時間 ET 歸零。 <b>時段 4：</b> 輸入位元 IN 設 ON 時，測量位元 TI 被開啟，已計時間 ET 開始計數，輸出位元 Q 保持 OFF。 <b>時段 5：</b> 在已計時間 ET 到達預設時間 PT 之前，將輸入位元 IN 設為 OFF，測量位元 TI 將被關閉，同時已計時間 ET 歸零。因為 ET 仍小於 PT，輸出位元 Q 保持在 OFF。	

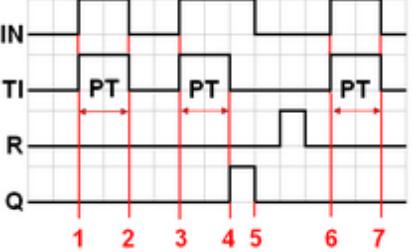
### ● 延時開關 (OFF)

電位圖	暫存器
	<p>輸入位元 (IN)：計時器的總開關。</p> <p>測量位元 (TI)：計時開始設 ON。</p> <p>輸出位元 (Q)：計時結束後設 OFF。</p> <p>預設時間 (PT)：設定計時器時間數值。</p> <p>已計時間 (ET)：顯示計時器目前已計時間。</p>
說明 (參照上圖)	
<b>時段 1：</b> 輸入位元 IN 設 ON 時，測量位元 TI 保持 OFF，輸出位元 Q 被開啟，已計時間 ET 歸零。	
<b>時段 2：</b> 輸入位元 IN 設 OFF 時，測量位元 TI 被開啟，輸出位元 Q 保持 ON，已計時間 ET 開始計數。	
<b>時段 3：</b> 當已計時間 ET 等於預設時間 PT 時，輸出位元 Q 和測量位元 TI 被關閉。	
<b>時段 4：</b> 輸入位元 IN 設 ON 時，測量位元 TI 保持 OFF，輸出位元 Q 被開啟，已計時間歸零。	
<b>時段 5：</b> 當輸入位元 IN 設 OFF 時，測量位元 TI 被開啟，輸出位元 Q 保持 ON，已計時間 ET 開始計數。	
<b>時段 6：</b> 當在已計時間 ET 到達預設時間 PT 的數值前設輸入位元 IN 為 ON，測量位元 TI 被關閉，同時輸出位元 Q 保持 ON，已計時間 ET 歸零。	

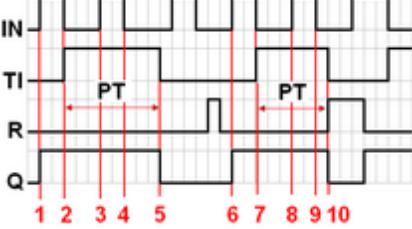
### ● 脈衝啟動開關

電位圖	暫存器
	<p>輸入位元 (IN)：計時器的總開關。</p> <p>測量位元 (TI)：計時開始設 ON；計時結束後設 OFF。</p> <p>輸出位元 (Q)：計時開始設 ON；計時結束後設 OFF。</p> <p>預設時間 (PT)：設定計時器時間數值。</p> <p>已計時間 (ET)：顯示計時器目前已計時間。</p>
說明 (參照上圖)	
<b>時段 1：</b> 當輸入位元 IN 設 ON 時，測量位元 TI 和輸出位元 Q 同時被開啟，已計時間 ET 開始計數。	
<b>時段 2：</b> 當已計時間 ET 等於預設時間 PT 時，輸出位元 Q 和測量位元 TI 同時被關閉。(因為在計數同時已先將輸入位元 IN 設 OFF，所以已計時間 ET 將被自動歸零。)	
<b>時段 3：</b> 當輸入位元 IN 設 ON 時，測量位元 TI 和輸出位元 Q 同時被開啟，已計時間 ET 開始計數。	
<b>時段 4：</b> 當已計時間 ET 等於預設時間 PT 時，輸出位元 Q 和測量位元 TI 同時被關閉。	

### ● 累加式延時開關 (ON)

電位圖	暫存器
	輸入位元 (IN)：計時器的總開關。 測量位元 (TI)：計時開始時設 ON。 輸出位元 (Q)：計時結束後設 ON。 預設時間 (PT)：設定計時器時間數值。 已計時間 (ET)：顯示計時器目前已計時間。 重置位元 (R)：將目前已計時間 (ET) 歸零。
說明 (參照上圖)	
時段 1：當輸入位元 IN 設 ON 時，測量位元 TI 被開啟，已計時間 ET 開始計數，輸出位元 Q 保持 OFF。	
時段 2：當輸入位元 IN 設 OFF 時，如果已計時間 ET 未到達預設時間 PT，測量位元 TI 被關閉，同時輸出位元 Q 保持 OFF。已計時間 ET 保持現在的狀態數值。	
時段 3：當輸入位元 IN 再度設 ON 時，測量位元 TI 被開啟，同時已計時間 ET 再次由剛剛保持的狀態數值開始計數，同時輸出位元 Q 保持 OFF。	
時段 4：當已計時間 ET 等於預設時間 PT 時，測量位元 TI 被關閉，同時輸出位元 Q 被開啟。	
時段 5：設輸入位元 IN 為 OFF，同時輸出位元 Q 也被關閉。(此時設重置位元 ON 使已計時間 ET 歸零後再設為 OFF。)	

### ● 累加式延時開關 (OFF)

電位圖	暫存器
	輸入位元 (IN)：計時器的總開關。 測量位元 (TI)：計時開始時設 ON。 輸出位元 (Q)：計時結束後設 OFF。 預設時間 (PT)：設定計時器時間數值。 已計時間 (ET)：顯示計時器目前已計時間。 重置位元 (R)：將目前已計時間 (ET) 歸零。
說明 (參照上圖)	
時段 1：當輸入位元 IN 設 ON 時，測量位元 TI 保持 OFF，同時輸出位元 Q 被開啟。	
時段 2：當輸入位元 IN 設 OFF 時，測量位元 TI 被開啟，同時輸出位元 Q 保持 ON。已計時間 ET 開始計數。	
時段 3：當輸入位元 IN 再度設 ON 時，測量位元 TI 和輸出位元 Q 保持 ON，同時已計時間 ET 暫停計數。	
時段 4：當輸入位元 IN 再度設 OFF 時，已計時間 ET 再次由剛剛保持的狀態數值開始計數。	
時段 5：當已計時間 ET 等於預設時間 PT 時，測量位元 TI 和輸出位元 Q 同時被關閉。(此時設重置位元 ON 使已計時間 ET 歸零後再設為 OFF。)	



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.31. 影像輸入

### 13.31.1. 概要

人機提供影像輸入功能，使用者加裝監視鏡頭後，透過監視鏡頭即可即時監看現場狀況，也能將畫面記錄到儲存裝置並且在電腦上做分析。

此功能可應用在各個層面，除了可監看現場狀況外，也能應用在行車裝置或是大樓監控。

各系列支援的設備類型與影像擷取方式如下表所示。

機型系列		iP	iE	eMT	mTV	XE	cMT-SVR	cMT-FHD cMT-FHDX cMT-HDM	Other cMT/cMT X
支援設備類型	網路攝影機	N/A	N/A	Y	N/A	Y	N/A	Y	Y
	USB 攝影機	N/A	N/A	Y	Y	Y	N/A	Y	Y
	類比影像輸入	N/A	N/A	eMT3121A eMT3151A	N/A	N/A	N/A	N/A	N/A

機型系列		非 cMT 系列 - 截圖	cMT / cMT X 系列 - 截圖/錄影
影像擷取	網路攝影機	位址觸發後拍攝當下一張圖片。	截圖: 位址觸發後拍攝當下一張圖片。
	USB 攝影機		錄影: 位址觸發後將根據指定的記錄時間區間進行影片錄製。
	類比影像輸入	位址觸發後根據設定拍攝當下前後多張圖片	

### 13.31.2. 設定



按下工作列上的【影像輸入】按鈕後即會開啟【影像輸入】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【影像輸入】物件。

## 一般屬性設定

### eMT、iE、XE、mTV 系列

#### 類比格式影像輸入



#### USB 攝影機影像輸入



#### 網路攝影機影像輸入



**cMT / cMT X 系列****USB 攝影機影像輸入****網路攝影機影像輸入****設定****描述****輸入通道**

可選擇使用影像輸入通道 1 或影像輸入通道 2。(限類比輸入)

**編碼格式**

可選擇 NTSC 或 PAL 訊號。(限類比輸入)

**RTSP**

網路攝影機的 RTSP 地址。

若該網路攝影機設有存取權限之帳號與密碼，可將帳號與密碼內嵌於 RTSP 地址中。

例如:

admin:admin@192.168.1.119:554/cam/realmonitor?channel=1&subtype=0

**透過 TCP 串流影像**

勾選後，可支援透過 TCP 串流影像的網路攝影機。

**緩衝**

當網路不穩定時，合適的緩衝時間設定，可以提供更穩定的影像播放。可設定的範圍為 200 毫秒至 5 秒。

**影像擷取位址**

勾選 [使用影像擷取功能] 來設定擷取輸入影像畫面之功能。非 cMT 系列機種為擷取圖片，而 cMT / cMT X 系列機種為擷取位址觸發擷取圖片或是擷取當下前後指定時間的 avi 影片。

**影像擷取位址**

觸發系統擷取影像的控制位址。

**儲存空間**

設定擷取圖片的儲存裝置。

根據像輸入類型，儲存裝置的相關資訊如下：

格式	儲存裝置
類比格式	可儲存於 SD 卡或 USB 碟。
USB 攝影機	只可儲存於 SD 卡。
網路攝影機	可儲存於 SD 卡或 USB 碟。

### 記錄時間 (cMT / cMT X 系列)

位址觸發後將根據指定的前後紀錄時間進行影片錄製，並存在指定的儲存裝置內。請注意，錄製完成後，HMI 會需要一點時間進行壓縮。壓縮時間長短會根據 HMI 的 CPU 能力有所不同。錄製與壓縮過程中，物件右下角會出現紅點表示正在進行錄製或壓縮。在壓縮過程中，無法進行下一個影片的錄製。

### 記錄時間 (eMT 系列)

設定擷取畫面之時間範圍。

格式	儲存方式
類比格式	<ul style="list-style-type: none"> <li>● 最大擷取範圍為 [影像擷取位址] 觸發時之前後 10 秒。</li> <li>● 系統每秒擷取圖片一次。</li> <li>● 圖片檔案命名規則： [影像擷取位址] 觸發前後： YYYYMMDDhhmmss.jpg [影像擷取位址] 觸發當下： YYYYMMDDhhmmss@.jpg 例如，設定記錄時間為前後 5 秒，當 [影像擷取位址] 的狀態由 OFF 轉為 ON 時，系統將從觸發時間點起算，每秒 1 張，擷取前後 5 秒之輸入畫面，共 11 張圖。</li> </ul>
USB 攝影機	僅擷取控制位址觸發當下的影像畫面。圖片檔案命名規則：YYYYMMDDhhmmss.png。
網路攝影機	僅擷取控制位址觸發當下的影像畫面。圖片檔案命名規則：YYYYMMDDhhmmss.png。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 控制位址

啟用後，將特定的數值寫入 [控制位址] 及連續的暫存器可控制影像輸出。假設控制位址設定為 LW-n (n 為任意位址)，將特定數值寫入指定

位址的執行命令如下表所示。

#### 類比格式影像輸入控制功能

位址	數值	命令內容
LW-n	0	停止播放影像
	1	開啟輸入通道 1 的影像並顯示於屏幕上
	2	開啟輸入通道 2 的影像並顯示於屏幕上
	3	開啟輸入通道 1 的影像但不顯示 (仍可執行影像擷取功能)
	4	開啟輸入通道 2 的影像但不顯示 (仍可執行影像擷取功能)
LW+*	1	暫停/繼續播放影像
LW-n+2	1~100	對比調整 (限類比輸入)
LW-n+3	1~100	亮度調整 (限類比輸入)

#### USB 攝影機/網路攝影機控制功能

位址	數值	命令內容
LW-n	0	停止播放影像
	1	開啟播放影像
LW-n+1	1	暫停/繼續播放影像
LW-n+2	字串	網路攝影機的 RTSP 地址(64 字組)

- 在變更 [控制位址 (LW-n)] 的數值後，系統將保留變更後的值。
- 在變更 [控制位址 + 1 (LW-n+1)] 的數值後，系統將在執行對應命令結束後將其清除為 0。
- 若不啟用 [使用控制功能]，系統將自動播放 [輸入通道] 指定之影像輸入。
- 勾選 [顯示調整] 後，才可調整對比及亮度 (限類比輸入)



#### Note

- 無論使用何種攝影機，系統中任何時間點只能有一組影像輸入通道開啟影像輸入。

#### 關於類比格式影像輸入：

- 影像擷取功能不受暫停播放控制之影響，所擷取的圖片仍是外部影像輸入之即時畫面。
- 類比輸入推薦的格式類型和解析度：

	1:1	50%
NTSC	720 x 480	360 x 240
PAL	720 x 576	360 x 288



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 關於 cMT / cMT X 系列影像擷取：

- 影片錄製完畢後，會需要一段時間進行壓縮，壓縮時間會依照錄影時間長短而定。
- 當人機正在壓縮影片檔案的時候，觸發新的事件錄影不會執行。

### 關於使用 USB 攝影機：

- 若使用 cMT / cMT X 系列 HMI，無法透過 cMT Viewer 檢視 USB 攝影機影像。
- 使用 USB 攝影機時，若在影像播放中途移除 USB 攝影機，則影像不會在插回攝影機時重新載入。若有勾選控制位址，則先用控制位址停止影像載入再重新開始。若無使用控制位址，則需要切換視窗或重啟 HMI。
- eMT3070A 機型的影像輸入物件的最大尺寸為 340\*240，eMT3105P、eMT3120A、eMT3150A、XE 及 mTV 系列機型則為 640\*480。
- 使用 USB 攝影機時，實際的顯示尺寸會根據 USB 攝影機支援的解析度中尋找一個最接近規劃的物件大小的解析度。也就是說，影像顯示的尺寸不一定會剛好與物件設計的尺寸相同。建議將物件尺寸調整成與 USB 攝影機支援的解析度。
- 使用 USB 攝影機時，為了避免攝影機的實際顯示畫面超出視窗的大小，系統會保留右邊及底部 50 像素。也就是說，影像輸入物件的右邊及下方邊緣皆會距離視窗邊緣 50 像素。
- 使用 USB 攝影機時，影像輸入物件的背景色為黑色。若輸入影像的解析度小於物件設計時的尺寸，則會顯示黑色邊框，建議將物件尺寸調整成與 USB 攝影機相同的解析度。
- 目前已測試可成功顯示影像的 USB 攝影機為 Logitech C170, Logitech C310, Logitech C910, LifeCam VX-2000。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 關於使用網路攝影機：

- 若使用 cMT / cMT X 系列 HMI，可能可以在 Windows PC 上透過 cMT Viewer 檢視網路攝影機影像。該電腦須另安裝第三方解碼器。
- 由於相容性與使用的設備相關，可否成功檢視網路攝影機影像，需以實際測試結果為主。
- 使用網路攝影機需設定攝影機的 RTSP 地址。RTSP 地址可於該網路攝影機之設定頁面中查詢，或可於網路上的相關資料庫中查詢。
- 使用網路攝影機時，若在影像播放中途與網路攝影機連線中斷，則影像不會在攝影機重新上線時繼續播放。因此於設計工程檔案時需注意，若有勾選使用控制位址，則可先用控制位址停止影像載入再重新開始。若無使用控制位址，則需要切換視窗或重啟 HMI 以重新播放影像。
- 使用網路攝影機時，畫面上實際的顯示尺寸即為物件的尺寸。若原影像解析度與物件大小不同，影像會自動調整為與物件設定大小相同。影像調整之際，可能會有失真情形，故仍建議將物件尺寸調整成與網路攝影機影像之解析度相同。
- 請使用符合 ONVIF 規範的網路攝影機並使用 RTSP 方式串流。
- 為確保影像串流的品質及影像流暢度，若有影像動作延遲，或 CPU 負載過高造成人機反應緩慢等情形，請調整攝影機輸出影像之設定，以維持 HMI 正常運作。另外，因各機型硬體規格不同，以及各工程檔案也有差異，最佳參數需讓使用者微調。
- 下表為當前建議的影像規格：

解析度	最大 960x544
影像壓縮格式	H.264, MJPEG
畫面更新率	15 fps
位元傳輸率	最大 800kbps

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

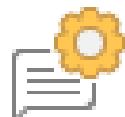
 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.32. 系統訊息

### 13.32.1. 概要

當物件選擇被使用前需彈出確認視窗或是否可使用遠端登入時，系統會先彈出訊息視窗：【操作確認提示】、【禁止寫入命令】、【允許寫入命令】的訊息，此三項訊息的內容可在【系統訊息】中編輯。

### 13.32.2. 設定



按下工作列上的【系統訊息】按鈕後即會開啟【系統訊息】對話窗，可以設定系統訊息。

#### 系統訊息設定



設定	描述
視窗尺寸	選擇提示的視窗和字體尺寸。
操作確認提示	要操作受保護的物件時，顯示訊息向使用者確認這項操作。您可以設定【操作確認提示】中的訊息與【確認】、【取消】兩個按鈕上的文字。

[確認] 與 [取消] 兩個按鈕上的文字，需使用相同的字型。另外，只有在 [訊息] 選擇使用文字標籤庫時，[確認] 與 [取消] 兩個按鈕才允許使用文字標籤庫。

**禁止寫入命令** 當系統暫存器 LB-9196 (本地 HMI 只支援檢視功能) 設為 ON 時，顯示此訊息。

**允許寫入命令** 當系統暫存器 LB-9196 (本地 HMI 只支援檢視功能) 設為 OFF 時，顯示此訊息。

## 13.33. 配方檢視

### 13.33.1. 概要

【配方檢視】可用來檢視特定的配方資料，使用者可在【配方檢視】上觀察到該筆配方的所有項目及數值。

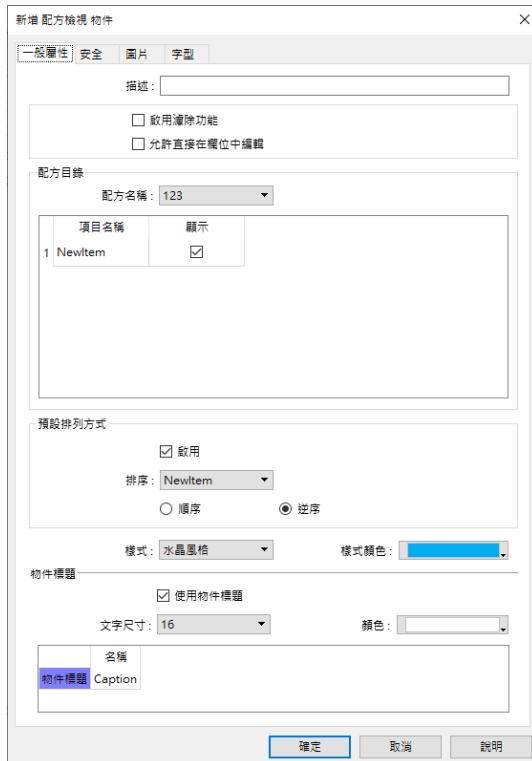
### 13.33.2. 設定



按下工作列上的【配方檢視】按鈕後即會出現【配方檢視】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【配方檢視】物件。

#### 一般屬性設定

cMT / cMT X 系列



eMT、iE、XE、mTV 系列



【配方檢視】各部份的名稱請參考下圖：



設定	描述
啟用濾除功能	若勾選，即可輸入欲過濾的關鍵字，只讀取有包含關鍵字的資料。
允許直接在欄位中編輯	若勾選，即可在配方檢視物件上直接編輯配方資料庫。
自動更新資料	若勾選，修改配方後系統將自動更新所檢視的內容。反之則需透過視窗的切換才會更新所檢視的內容。
配方目錄	<p><b>配方名稱</b> 選擇欲檢視的配方名稱，可點選下拉式選單尋找其他配方。</p> <p><b>選擇</b> 可設定欲顯示的配方項目欄位。</p>
預設排列方式	設定 [配方檢視] 項目的排序方式。提供 [順序] 與 [逆序] 兩種排序方式。
<b>樣式</b>	
(僅適用 cMT / cMT X)	可選擇 [預設]、[水晶風格]、[扁平化風格]。
使用物件標題 (僅適用 cMT / cMT X)	若勾選，則會在 [配方檢視] 物件顯示標題，標題文字亦可自訂。並可設定 [文字尺寸]、[顏色]、[名稱]。(只有當選擇的樣式為水晶風格或是扁平化風格時，才能設定物件標題。)
標題	<p><b>每個項目的標題</b>，根據 [資料/歷史] » [配方資料庫] 中的設定。</p> <p><b>透明</b> 若勾選，則標題就不會有背景顏色，並且不會出現 [顏色] 的選項。</p>
外觀	<p>物件的邊緣線顏色及背景顏色。</p> <p><b>透明</b> 若勾選，則物件就不會有背景顏色，並且不會出現 [顏色] 的選項。</p>
<b>自動調整欄寬(僅適用於 cMT / cMT X 預設模式)</b>	
網格	<p>依照內容自動調整欄寬</p> <p><b>區分配方每個資料的間隔線。</b></p> <p><b>透明</b> 若勾選，則網格就不會有顏色，並且不會出現 [顏色] 的選項。</p>

**選擇控制**

當點選到特定一行的資料時，所顯示的顏色。

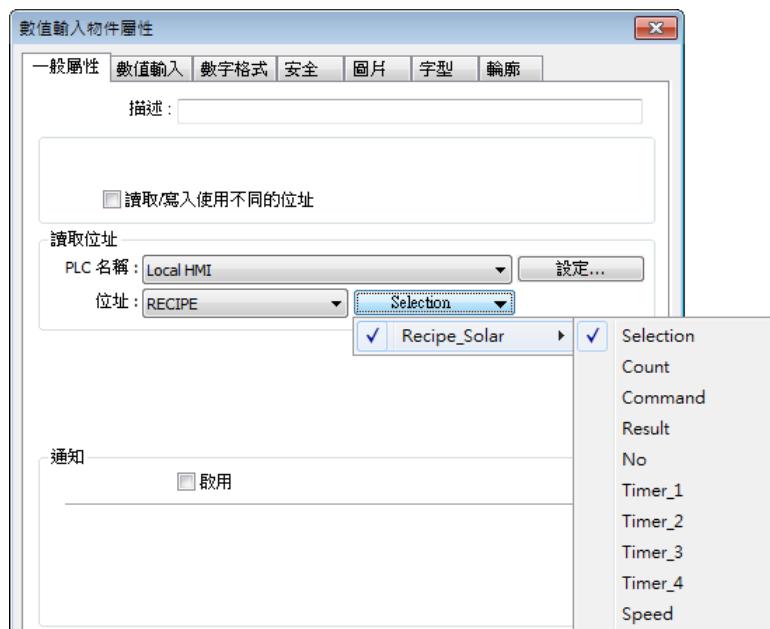


- 可使用四個系統暫存器來查詢/更新/增加/刪除配方資料庫的內容：

**Selection**

目前所選擇的配方編號。編號為從 0 開始計算，因此若點選第一筆，則 Selection 的數值會顯示 0，依此類推。

當 [Selection] 的值改變時，相對應的暫存器也會跟著改變，例如下圖中的 No., Timer\_1, Timer\_2。

**Count**

目前配方中的資料筆數。

**Command**

輸入特定的數值可對選取的配方資料下執行命令。(數字為執行的命令數值)

輸入 “1” 新增配方記錄

輸入 “2” 更新目前選擇的配方記錄

輸入 “3” 刪除目前選擇的配方記錄

輸入 “4” 刪除所有配方記錄

輸入 “5” 寫入所選擇的記錄至 PLC

輸入 “6” 更新由 PLC 所選擇的記錄

**Result**

可監看命令的執行結果。(數字為執行命令後的結果數值)

數值 “1” 代表成功執行命令

數值 “2” 代表選擇的記錄不存在

數值 “4” 代表未知的命令

數值 “8” 代表記錄已達上限(10000 筆)，無法再加入新記錄

數值 “16” 代表其他命令正在被執行

數值 “32” 代表傳送命令失敗

 使用此功能需要至 [資料/歷史] » [配方資料庫] 建立配方資料，請參考《24 Recipe Editor》。

## 範例 1

以下示範 [配方檢視] 和配方資料庫的簡易使用方法。

在範例中會建立配方資料庫，並且使用 [配方檢視] 來檢視和選擇配方資料。當您點選了 [配方檢視] 中的任一筆記錄時，[Selection] 及對應的暫存器數值都會改變。最後，可以使用 [Command] 來編輯修改配方資料庫。

*Simple Recipe View & Recipe Database*

No	Name	Timer_1	Timer_2	Timer_3	Timer_4	Speed
0	Mercury	10	1	11	12	28.500
1	Venus	20	1	21	22	33.500
2	Mars	30	2	32	35	41.500
3	Jupiter	50	3	53	56	50.500
4	Saturn	80	5	85	90	60.500

System Registers:

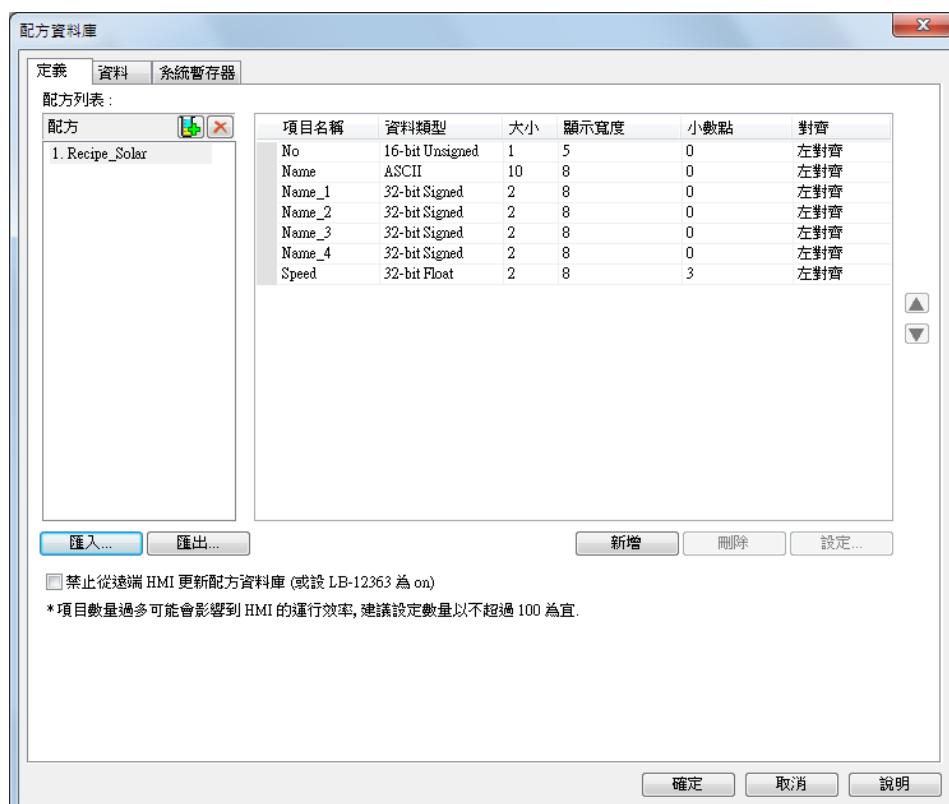
Selection:  Count:  Command:  Result:

Selected records: (modify here)

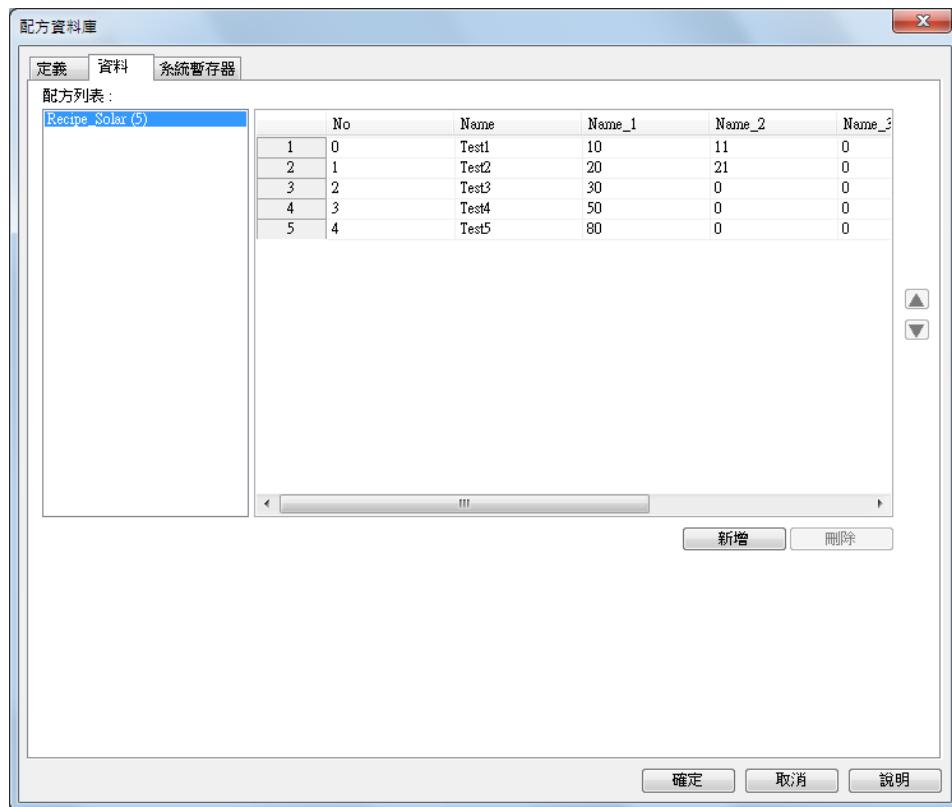
No.	Name:			
<input type="text" value="2"/>	<input type="text" value="Mars"/>			
Timer_1:	Timer_2:	Timer_3:	Timer_4:	Speed:
<input type="text" value="30"/>	<input type="text" value="2"/>	<input type="text" value="32"/>	<input type="text" value="35"/>	<input type="text" value="41.500"/>

Edited by Mac

1. 建立一個配方，如下圖：



2. 使用 [資料] 增加任意數筆的資料，或如下圖所示：



3. 新增一個 [配方檢視] 物件並使用剛建立的配方資料庫。
4. 建立四個使用 [Selection]、[Count]、[Command]、[Result] 的 [數值] 物件。
5. 建立 No, Name, Timer\_1, Timer\_2, ..., Timer\_4, Speed 項目對應的物件。例如使用 [字元] 並設定位址為 “Recipe” 中的 “Name”，如下圖所示：



6. 如此便完成設計。
7. 畫面中已選取 Mars，並且對應的 Timer\_1, Timer\_2, ..., Speed 亦更新至對應的物件。配方記錄中共有 5 筆記錄，所以 Count 顯示為 5。可以試著選擇 [配方檢視] 的其他筆記錄，對應的 Name, Timer\_1, ... 等資訊也會同時更新。
8. 亦可試著操作：
  - 新增：在 Command 中輸入 1，將目前輸入的資料加至配方資料庫成為一筆新的記錄。
  - 更新：在 Command 中輸入 2，將 [數值] 及 [字元] 的資料更新至配方資料庫。
  - 刪除：

在 Command 中輸入 3 即可刪除目前選擇的記錄。

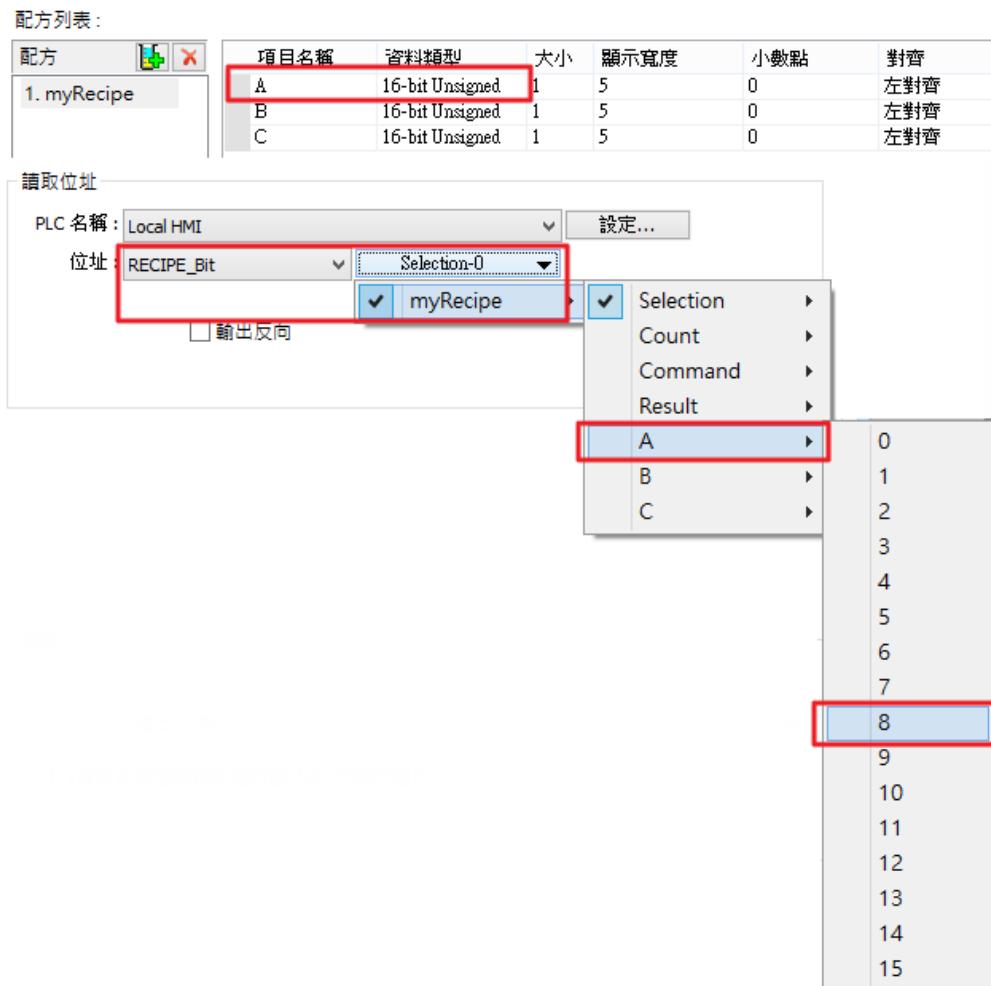
- 排序不同的項目  
點選 [標題] 即可對不同的項目排序。

## 範例 2

以下示範如何使用 RECIPE\_Bit 讀取配方資料的位元位址。

資料庫本身無法新增 BOOL 類型的項目，但是使用者仍然可以直接讀寫 16bit, 32bit 資料的個別位元資料。

如下圖所示，在位元物件的讀取位址，選擇 RECIPE\_Bit，並指向欲讀寫的項目，會展開選擇位元的選單。如此，可以運用配方資料庫記錄並讀寫位元資料。



## 13.34. 流動塊

### 13.34.1. 概要

**[流動塊]** 物件可表示導管內的滑塊移動或運輸線的情況。不同於以往使用 [移動圖形] 物件繪製流動圖形時，需自行丈量及確認兩點之間的位置是否齊整，流動塊的每一段區塊必為精準的水平或垂直線段且流動間隔固定，cMT / cMT X 系列則可繪製多角度，不限定水平或垂直。

以下列舉 **[流動塊]** 物件的特點：

- 每個線段必為垂直或水平的直線，且流動間隔固定，cMT / cMT X 系列則可繪製多角度 (限制為 5 度)，不限定水平或垂直。
- 支援動態調整流速及方向 (流速及方向可用指定暫存器調整)。
- 可使用安全機制。利用指定位元的狀態作為流動塊顯示與否的依據。

### 13.34.2. 設定



請直接點擊 **[流動塊]** 圖示建立此物件，或點選工具列上的 **[物件] » [動畫] » [流動塊]** 新增此物件。

#### 一般屬性設定



設定	描述
反向	流動塊的流動方向會依照建立時的順序方向移動 (藍色箭頭方向)。 勾選反向後，流動的方向將以反向移動。
動態速度	讀取位址 流動塊的速度及方向可用指定暫存器調整。使用範圍為 -25 ~ 25。當輸入負值時代表流動反向。

## 設定

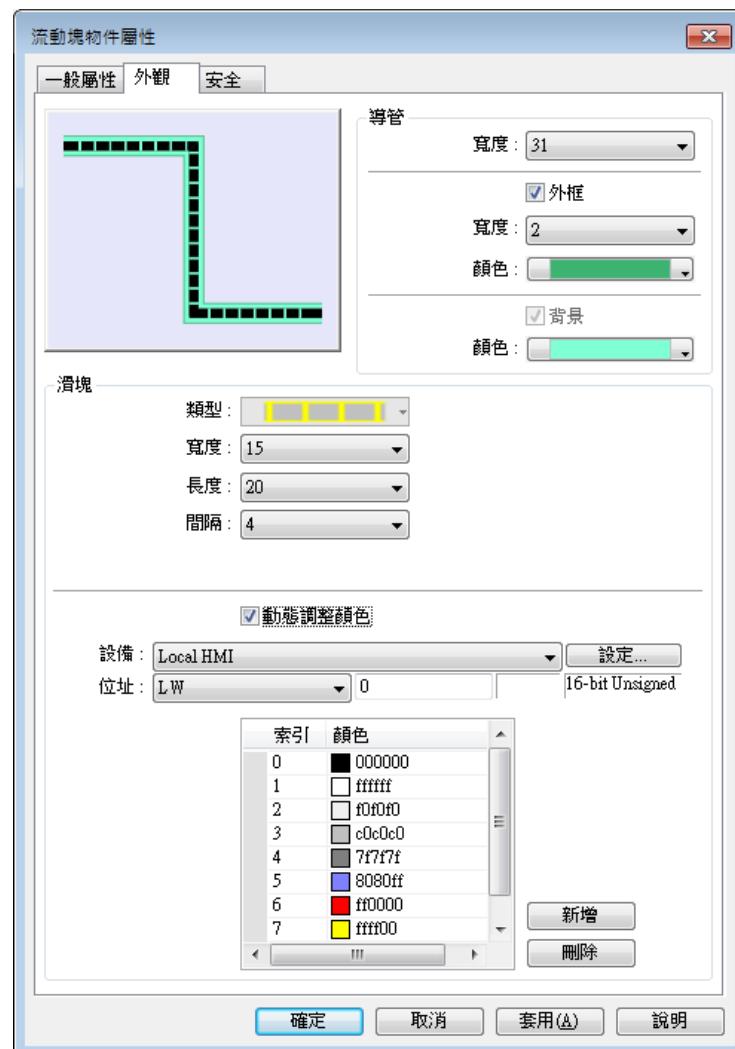
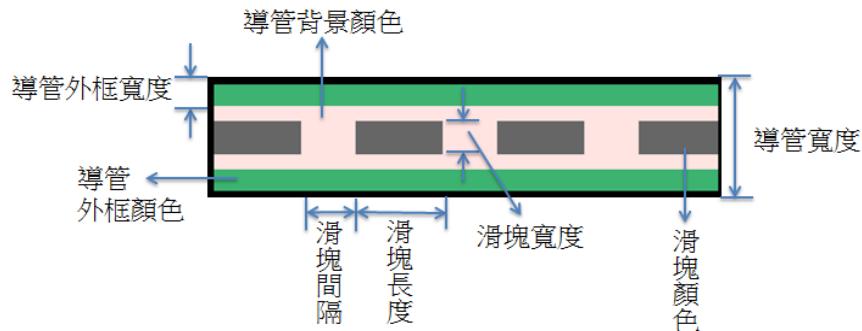
顯示目前選擇的位址及使用格式。並可在此進階使用【系統暫存器】、  
【索引暫存器】、【位址標籤庫】來設定位址。

## 流動速度

分為 25 個級速，數值越大，流速越快。若未使用【動態速度】時，  
則流動速度可設定的範圍僅為正向 0 ~ 25。

## 外觀設定

設定流動塊物件的外觀，各部位的名稱請參考下圖：



設定	描述
導管	流動塊外圍的屬性設定。可設定外圍的背景顏色、邊界顏色及寬度。 當勾選 [外框] 時，則必定要使用背景。
滑塊	流動塊內部的屬性設定。可設定流動塊的類型、長、寬、間隔及顏色。類型分為矩形與箭頭。箭頭指向為流動塊流動方向。 矩形： 箭頭：
動態調整顏色	可在人機上動態設定流動塊的顏色，提供 256 種自訂顏色，編號 0~255。透過指定的位址，可進行顏色切換。若在指定位址輸入的值大於自訂顏色範圍的上限，將使用編號最大的顏色。

### Note

- 若在 [一般屬性] 設定頁中同時勾選[ 反向] 及 [動態速度]，則當在動態速度的控制位址中輸入負值時，流動塊流動方向將為順向流動。
- 若滑塊類型選擇 [箭頭] 且搭配動態速度功能，未給定初始速度值時不會顯示箭頭。
- 繪製流動塊物件時，為避免轉折處重疊面積過大，在轉折處會有最小的規劃量。如圖 34.1 的十字下方的圖形。圖 34.2 的每一線段皆為使用最小規劃量繪製出的圖形。

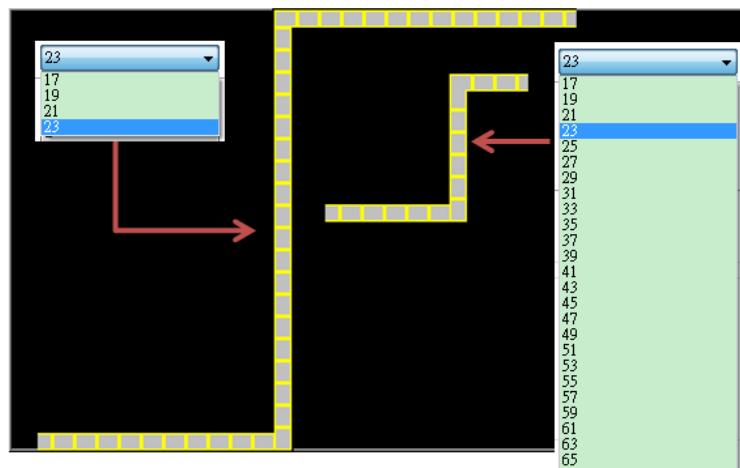


(圖 34.1)

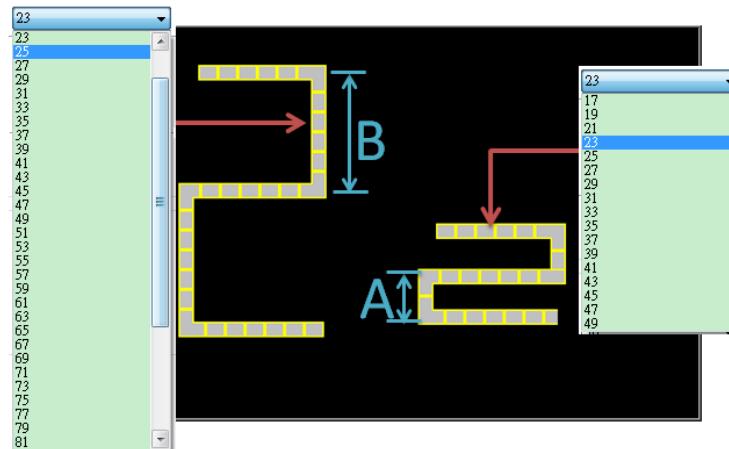
(圖 34.2)

流動塊可調整的長寬高參數會根據已繪製的圖形間隔尺寸及視窗尺寸改變。

如下圖，當流動塊整體尺寸越大，則為避免參數調整後會超出視窗尺寸，可設定的長寬參數則越少。反之當整體尺寸越小，可設定的長寬參數則越多。



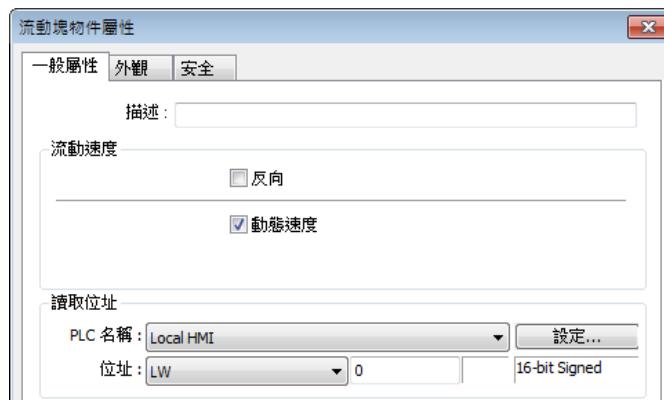
如下圖，當流動塊轉折處的間隔越小，則為了避免參數調整後，會致使流動塊上邊與下邊觸及 (如圖 34.4 A 區段)，可調整參數會越少；反之當流動塊轉折處間隔越大 (如圖 34.4 B)，可調參數越多。



## 範例 1

以下範例展示如何藉由【動態速度】調整流動塊的速度及方向。

- 建立一個流動塊物件，並使用動態速度，位址設為 LW-0，格式為 16-bit Signed。



- 建立一個數值物件，位址設為 LW-0。上限為 25，下限為 -25，格式為 16-bit Signed。



- 執行模擬或下載至人機驗證操作。當在 LW-0 輸入正值時，流動方向為順向，且數值越大，流動速度越快。當輸入負值時，流動為反向，且數值越小，流動速度越快。若輸入值為 0，則停止流動。



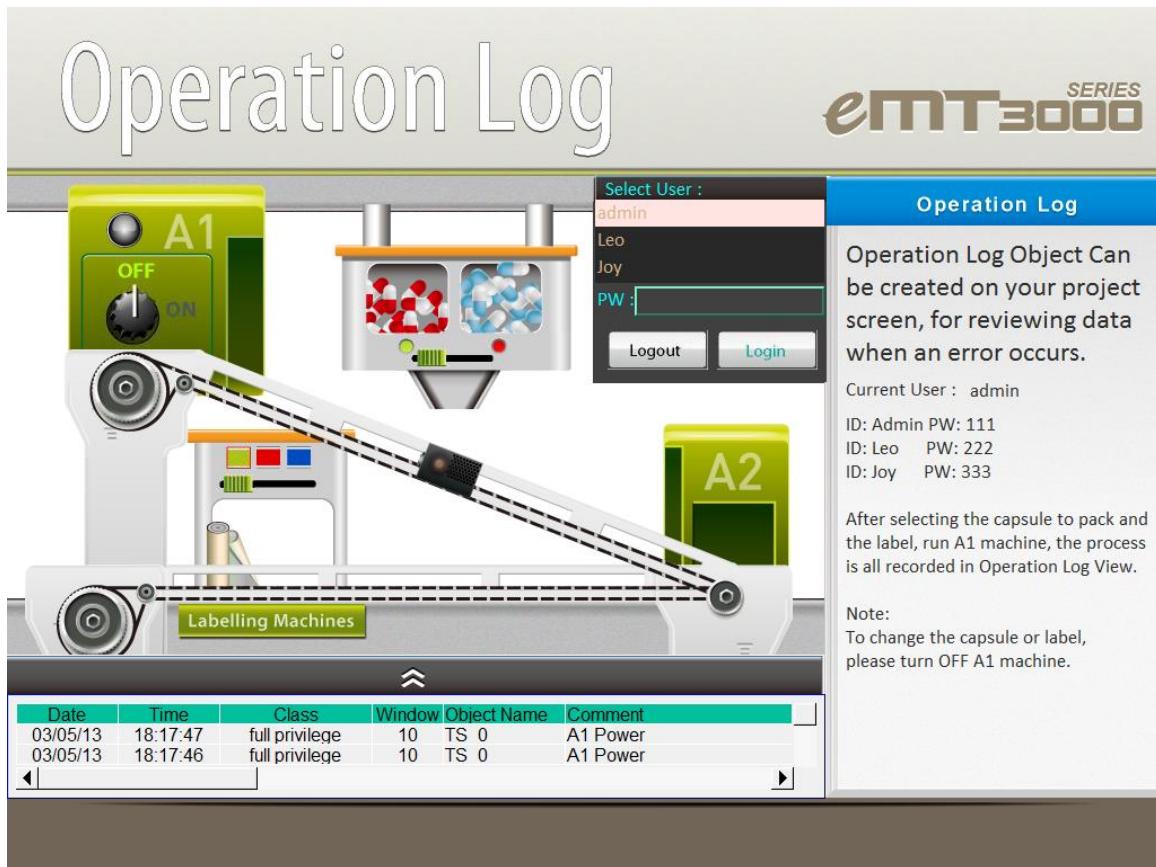
請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.35. 操作記錄

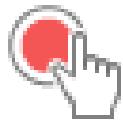
### 13.35.1. 操作記錄設定

#### 13.35.1.1. 概要

透過操作記錄的功能，能夠即時將使用者所操作的步驟顯示出來，當出現異常狀況時，可用來分析操作記錄，從備份出來的報表可以得知操作過程是否符合作業流程，進一步能針對有問題的部份進行檢討及修正。操作記錄若設定適當，也能用於符合審計追蹤與電子簽名之要求，保障資料完整性。

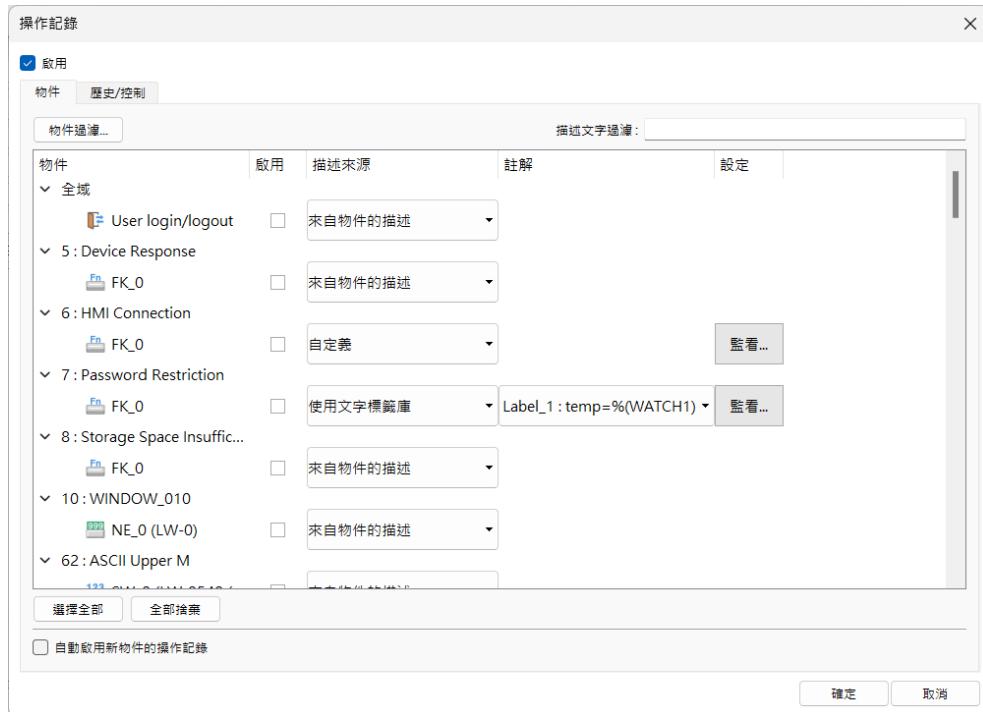


### 13.35.1.2. 設定

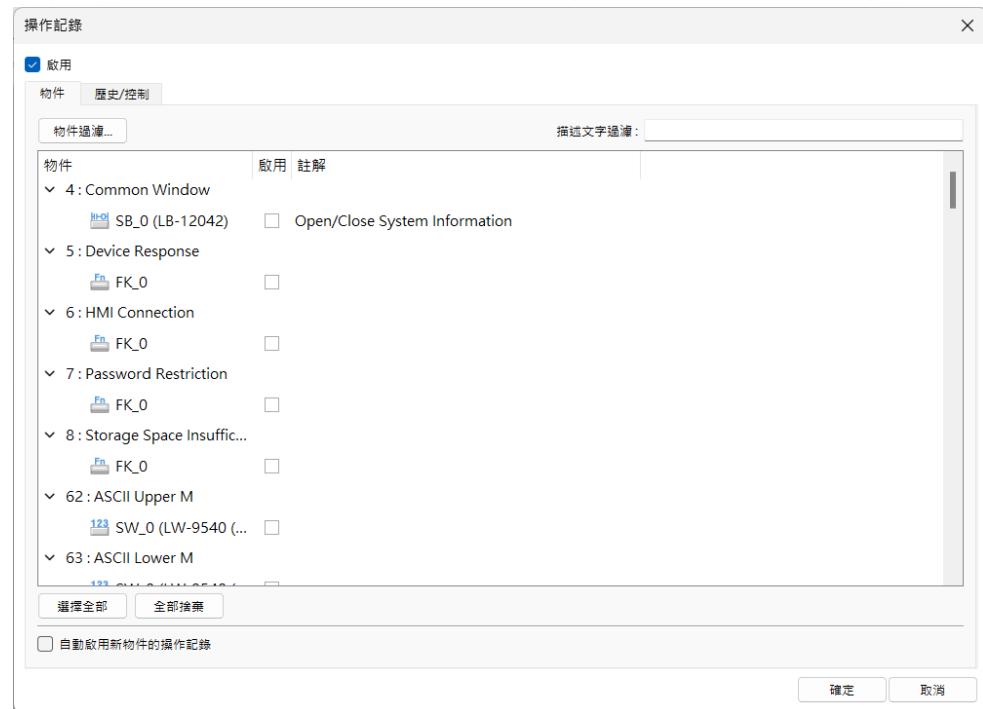


設定需要被記錄的物件。點選 [資料/歷史]，點選 [操作記錄設定] 並勾選 [啟用操作記錄功能]。

cMT、cMT X 系列



eMT、iE、XE、mTV 系列



## 物件設定

設定	描述
物件	啟用操作記錄功能後，可選擇記錄使用者登入登出資訊或視窗的物件。系統會依照視窗的編號自動列出所有已經建立，且能被記錄的物件。點選 [物件過濾...] 會列出所有能被記錄的物件種類。使用者可藉由物件過濾功能，快速找到希望設定記錄的物件。
啟用	選擇物件是否要被記錄。
描述來源	<p><b>來自物件的描述</b></p> <p>操作記錄的 [描述] 欄位內容來源為該物件的描述設定。</p> 
自定義	自定義操作記錄的 [描述] 欄位內容，點擊 [監看...] 設定描述內容。
使用文字標籤庫	[描述] 欄位支援文字標籤庫，[描述] 欄位內容在不同語言下記錄翻譯文字。
設定	當描述來源為 [自定義] 或 [使用文字標籤庫] 時，點擊 [監看...] 使用者可以設定監看位址對應的暫存器數值。使用文字標籤庫為描述來源時，監看位址語法需定義於文字標籤內容中。最多可同時監看四個位址。
	
選擇全部	選擇記錄所有物件。

若有使用 [篩選] 功能，則 [選擇全部] 只會全選顯示於列表上的物件。

#### 清除全部

清除所有已勾選的物件。

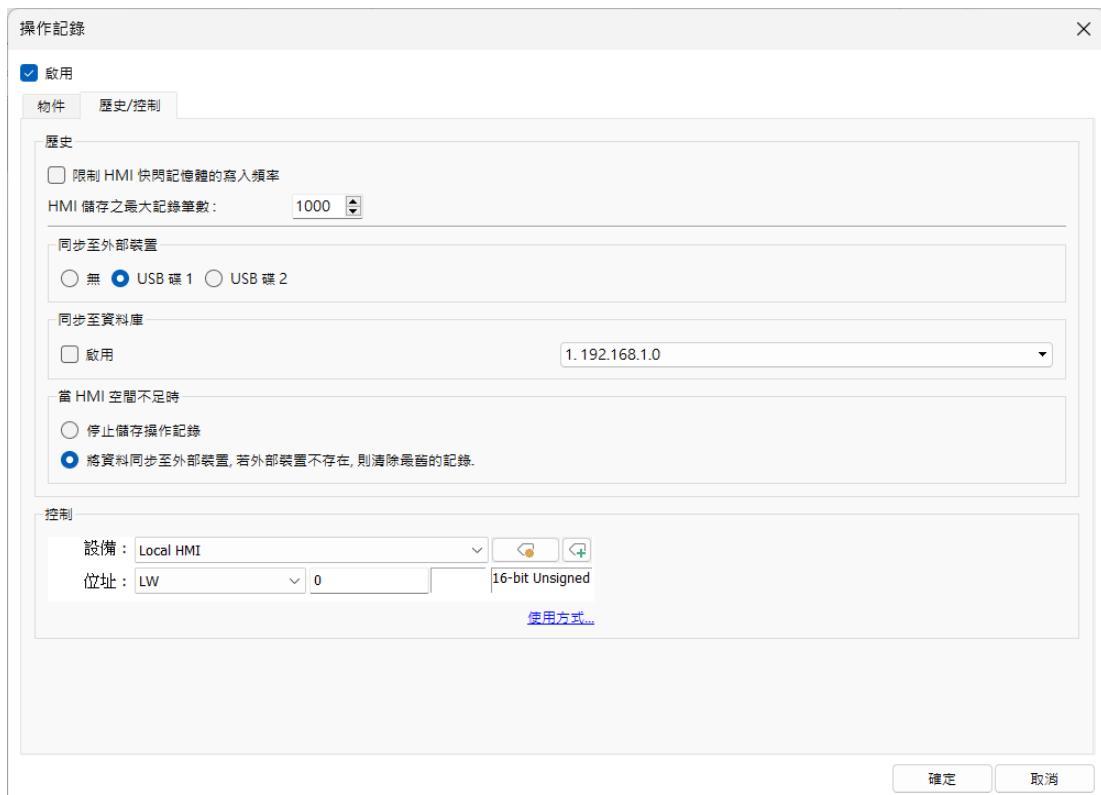
若有使用 [篩選] 功能，則 [清除全部] 只會取消顯示於列表上的物件。

#### 自動啟用操作

##### 記錄於新增的 物件

啟用此選項後，後續新增的物件會自動開啟操作記錄。

## 歷史/控制



#### 設定

#### 描述

##### 儲存設定

設定操作記錄的資料儲存方式。

##### 限制HMI快閃記憶體的寫入頻率

啟用後，系統將以10秒的時間間隔將操作記錄儲存在HMI上，為了避免操作紀錄在兩次儲存動作間因關機而造成資料的流失，EasyBuilder Pro 提供系統暫存器 LB-9034，只需對其送出 ON 的訊號，系統即會執行一次儲存動作。

##### HMI記憶體之最大記錄筆數

設定HMI記憶體可儲存的最大資料筆數。

##### 同步 / 備份至外部裝置 / 資料庫

選擇將資料備份至SD卡或USB碟，也可同步更新到資料庫上(cMT / cMT X系列)。

#### 當HMI空間不足時

當人機內的儲存空間不足時，選擇【停止儲存操作記錄】以保留較早的操作記錄資料或選擇【將資料同步至外部裝置】。若選擇儲存至外部的裝置，當裝置不存在時，則人機會自動清除HMI記憶體內最舊的記錄。

#### 控制位址

輸入特定的數值可對選取的操作記錄下執行命令並回傳命令執行結果。

假設控制位址為LW-n (n為任意地址)，則顯示執行結果的位址為LW-n+1。

控制位址 (LW-n)：

- (1)：清除所有記錄
- (2)：複製資料到USB碟
- (3)：複製資料到SD卡
- (4)：複製資料到USB碟並清除人機內的操作記錄
- (5)：複製資料到SD卡並清除人機內的操作記錄
- (6)：在HMI上開啟操作記錄功能
- (7)：在HMI上關閉操作記錄功能
- (8)：在更換HMI後，延用原先儲存在USB碟上之歷史資料的功能。
- (9)：在更換HMI後，延用原先儲存在SD卡上之歷史資料的功能。
- (10)：複製資料到資料庫伺服器。(cMT / cMT X系列)
- (11)：複製資料到資料庫伺服器並清除HMI記憶體內的資料。(cMT / cMT X系列)
- (12)：在更換HMI後，延用原先儲存在資料庫上之歷史資料的功能。  
(cMT/ cMT X系列)

命令執行結果 (LW-n+1)：

- (0)：處理中
- (1)：執行成功
- (2)：裝置不存在
- (3)：操作記錄不存在
- (4)：無法判讀的錯誤

#### Note

- 操作記錄僅記錄可被手動觸發的物件，被動的物件不會被記錄，如：定時式資料傳輸物件。
- 若使用離線 / 連線模擬時，操作記錄會被存放於 EasyBuilder 安裝資料夾下的 HMI\_memory\operationlog\operationlog.db
- 若使用位元設定物件觸發巨集時，則操作記錄會記錄兩筆資料，位元的觸發及巨集的觸發。

### 13.35.2. 操作記錄檢視

#### 13.35.2.1. 概要

[操作記錄檢視] 物件可用來檢視操作記錄。

#### 13.35.2.2. 設定



使用此物件前，需先至 [操作記錄設定] 設定須檢視的物件。點選 [資料/歷史] » [操作記錄檢視]。

#### 一般屬性設定

cMT、cMT X 系列



eMT、iE、XE、mTV 系列





設定	描述
樣式	可選擇操作記錄檢視表格的樣式：預設、水晶風格、扁平化風格
標題/外觀/網格 /選擇控制	當樣式選擇【預設】時，可設定此顯示參數。
字型	[操作記錄檢視] 物件上文字的字型、尺寸、顏色。
選項按鍵 (cMT / cMT X 系列)	<p>設定是否顯示物件上的選項按鍵 .</p> <ul style="list-style-type: none"> <li>● 預定義： 預先設定好是否顯示選項按鈕。</li> <li>● 動態： <input checked="" type="radio"/> 預定義 <input type="radio"/> 動態</li> </ul> <hr/> <p>設備 : Local HMI 位址 : LB 0 </p> <p>設定一個位元位址動態決定是否顯示選項按鈕。</p>

## 標籤設定

cMT、cMT X 系列



eMT、iE、XE、mTV 系列



設定	描述
物件標題	顯示於 [操作記錄檢視] 物件上的標題。
欄位標題	顯示於 [操作記錄檢視] 物件的欄位的標題。
排序	設定操作記錄排序方式。
顯示順序	設定當操作記錄產生時，項目資訊的顯示順序。若 [顯示字元] 設定為 0，系統將顯示所有字元。
日期/時間	設定 [操作記錄檢視] 物件上日期及時間的格式。

### 13.35.3. 操作記錄列印

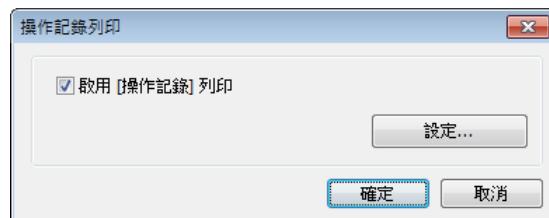
#### 13.35.3.1. 概要

[操作記錄列印] 可將操作記錄的內容轉換成報表的形式輸出至外接儲存裝置或印表機。當使用外接儲存裝置時，報表會輸出成 JPEG/PNG/PDF 格式。使用此功能前，需先啟用 [操作記錄設定]。若使用 cMT / cMT X 系列 HMI，透過 cMT Viewer 觸發列印時，報表會儲存到 HMI 上的 USB/SD 卡裝置。

### 13.35.3.2. 操作



勾選 [啟用 [操作記錄] 列印] 後，點選 [設定] 進入列印設定。

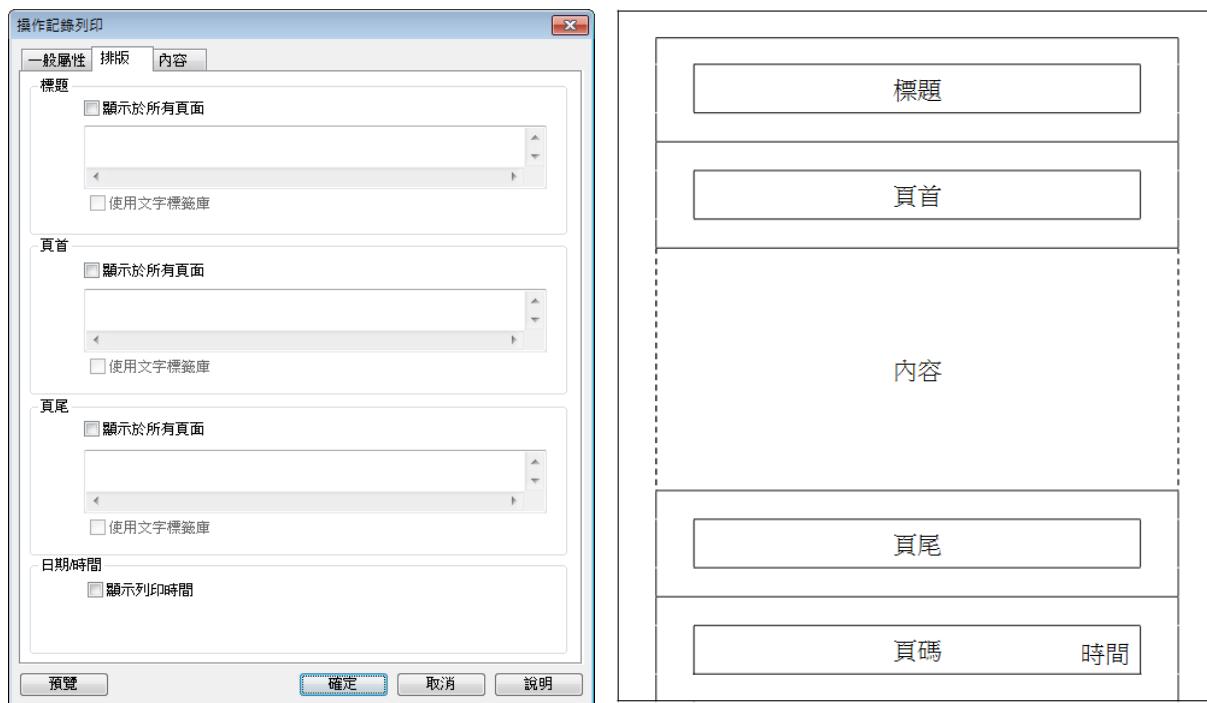


#### 一般屬性設定



設定	描述												
印表機	<p>選擇輸出 [操作記錄] 檔案時的外部儲存裝置。當選擇輸出於印表機時，印表機必須為 A4 格式。若選擇儲存於外部儲存裝置時，系統會產生一個 <b>operationlogsheet</b> 資料夾，將操作記錄的報表轉成 JPEG/PNG 圖檔，以 "列印的日期_列印序號" 的方式命名並儲存於此資料夾。</p> <p>例如：2013/05/08 列印的第一張圖檔，則 JPEG 檔名為 130508_0000，依此類推。</p>												
格式	<p>可選擇格式為 [圖像] 或 [PDF]。</p> <p>當選擇 [圖像] 時，cMT/cMT X HMI 會產生 PNG 格式，其他機種則為 JPEG 格式。</p> <p>[PDF] 選項僅支援於 cMT/cMT X HMI。</p>												
方向	選擇 [操作記錄] 檔案輸出時的方向。												
字型	選擇 [操作記錄] 檔案輸出時的字型及尺寸。當字型設定為大中小時的對應尺寸如下表：												
	<table border="1"><thead><tr><th>尺寸</th><th>標題尺寸</th><th>內容尺寸</th></tr></thead><tbody><tr><td>大</td><td>20點</td><td>16點</td></tr><tr><td>中</td><td>16點</td><td>12點</td></tr><tr><td>小</td><td>12點</td><td>8點</td></tr></tbody></table>	尺寸	標題尺寸	內容尺寸	大	20點	16點	中	16點	12點	小	12點	8點
尺寸	標題尺寸	內容尺寸											
大	20點	16點											
中	16點	12點											
小	12點	8點											
範圍	選擇 [操作記錄] 檔案輸出時的資料範圍。												
日期	使用日期決定輸出的資料範圍時，會從 [起始時間] 往前推算 [幾天內] 做輸出報表。最大為 180 天。												
筆數	使用筆數決定輸出的總數時，最大為 10000 筆記錄。												
觸發位址	設定控制 [操作記錄列印] 的暫存器位址來源。當暫存器被設定成 ON 時會觸發輸出，在輸出工作完成後，暫存器會自動重置成 OFF 狀態。												
預覽	檢視欲輸出的畫面。												

## 排版設定



排版設定頁各部分的位置分配如右圖。

設定	描述
<b>標題</b>	設定欲顯示標題的內容，標題只可為一行。 <b>顯示於所有頁面</b> 勾選後，則標題內容將顯示於每一頁，反之，只顯示於第一頁。
<b>頁首</b>	設定欲顯示頁首的內容，頁首內容最多支援列印五行文字。 <b>顯示於所有頁面</b> 勾選後，則頁首內容將顯示於每一頁，反之，只顯示於第一頁。
<b>頁尾</b>	設定欲顯示頁尾的內容，頁尾內容最多支援列印五行文字。 <b>顯示於所有頁面</b> 勾選後，則頁尾內容將顯示於每一頁，反之，只顯示於最後一頁。
<b>日期/時間</b>	勾選後，列印時的日期/時間會顯示於每一頁右下角，反之，則不顯示。
<b>頁碼</b>	每一頁都會顯示。

## 內容設定



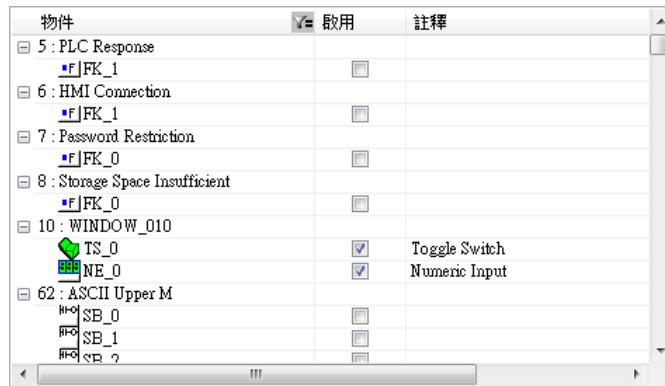
設定	描述
標題列表	設定標題名稱的顯示文字。
排序	<b>按時間順序</b> 操作記錄將由舊到新依序顯示，最近的操作記錄顯示於底部。 <b>按時間逆序</b> 操作記錄將由新到舊依序顯示，最近的操作記錄顯示於頂部。
日期/時間	設定時間資訊的顯示格式。

### 13.35.3.3. 範例講解

#### 範例 1

本範例講解如何建立一個操作記錄。

1. 建立一個 [位元狀態切換開關] 物件和 [數值] 物件於視窗 10。
2. 開啟 [操作記錄設定] 物件，啟用視窗 10 下的 [位元狀態切換開關] 物件和 [數值] 物件。



3. 建立一個 [操作記錄檢視] 物件。設定好各項屬性後關閉。
4. 執行離線模擬，觸發 [位元狀態切換開關] 物件和 [數值] 物件，操作記錄會顯示於 [操作記錄檢視] 物件上。

日期	時間	物件名稱	動作	位址	資訊
03/30/17	18:23:52	TS_0	Toggle	Local HMI : LB-0	bit set OFF->ON
03/30/17	18:22:19	NE_0	Set word	Local HMI : LW-0	write 0->5



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

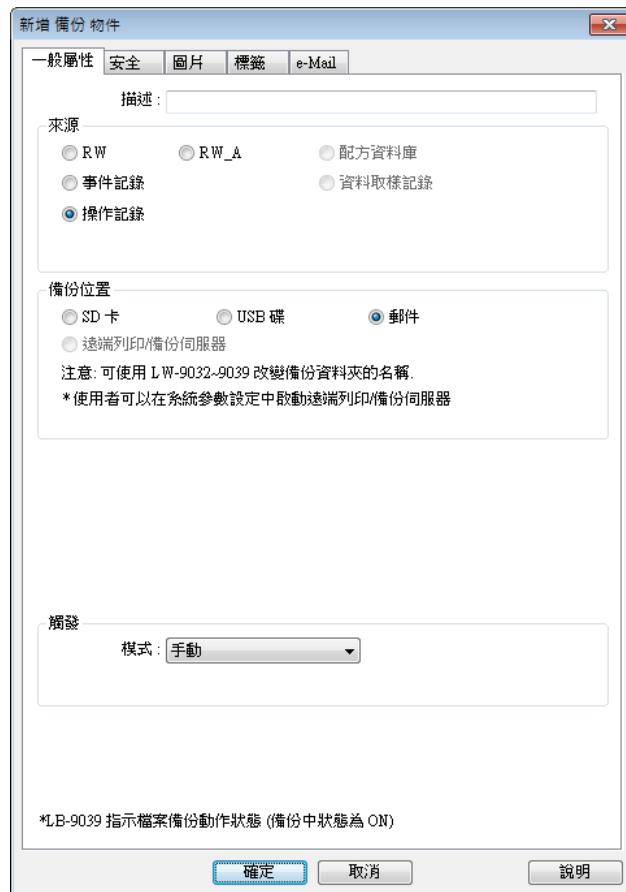
## 範例 2

操作記錄檔案可以透過 Utility Manager 上傳到電腦或是利用 [備份] 物件將檔案透過郵件傳送至信箱。

- 使用 Utility Manager 上傳
- 1. 開啟 Utility Manager，點選 [上傳]。
- 2. 勾選 [操作記錄]，輸入檔名及人機的 IP，點選 [上傳]。



- 使用郵件傳送到信箱
- 1. 點選 [系統參數設定] » [郵件] 設定郵件的伺服器及收/寄件者郵件地址。
- 2. 建立 [備份] 物件，來源設定為 [操作記錄]，備份位置設定為 [郵件]。



👉 郵件伺服器設定方法請參考《5 系統參數設定》。

⬇️ 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.36. 複合式多功能按鈕

### 13.36.1. 概要

複合式多功能按鈕物件可以執行多重指令。以往同一區塊若要執行多個指令時，必須將物件疊加在同一位置，且執行時會根據疊加的順序執行指令。使用者必須根據指令執行的順序疊加物件，規劃程式時需花費較多時間測試執行的順序。複合式多功能按鈕可以直接讓使用者設定多重指令的執行，並調整其順序。

以下列舉複合式多功能按鈕物件的特點：

- 可執行多重指令。
- 自行調整多重指令的執行順序。
- 可使用位元或字元作其物件顯示的狀態。

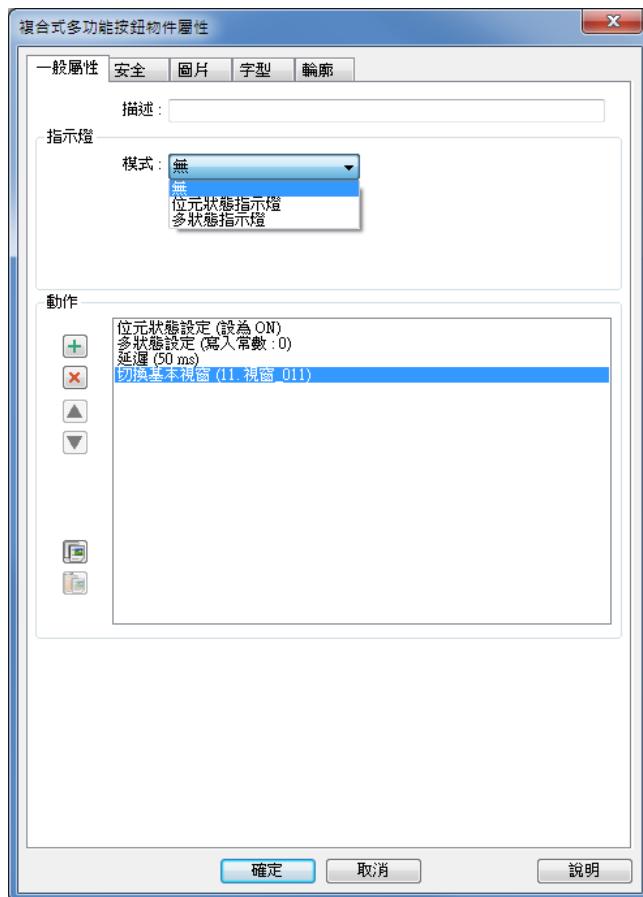
### 13.36.2. 設定



按下工作列的 [物件] 工 [複合式多功能按鈕] 按鈕後即會開啟 [複合式多功能按鈕] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [複合式多功能按鈕] 物件。

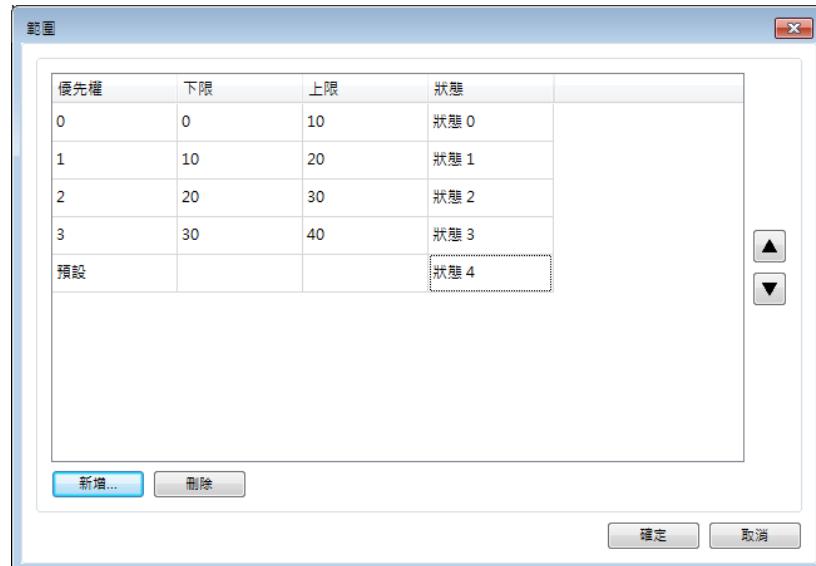
### 13.36.2.1. eMT、iE、XE、mTV 系列

#### 一般屬性設定



設定	描述
指示燈	物件顯示的狀態模式，可選擇是否使用多狀態顯示。
無	不使用多種狀態。
位元狀態指示燈	讀取位元暫存器的數據來顯示狀態。[輸出反向] 可以將讀取的狀態作反向顯示，例如位元的狀態實際上為 OFF，但勾選了 [輸出反向] 後會顯示為 ON。
多狀態指示燈	讀取字元暫存器的數據來顯示狀態。 [狀態數] 為物件顯示的狀態數目。狀態從 0 開始編號，能顯示的最大狀態編號為設定的 [狀態數] - 1，當要求顯示超過設定的狀態數時，系統會顯示最後一個狀態。例如設定 [狀態數] 為 8，則顯示的狀態依序為 0, 1, 2, ..., 7，若要求暫存器顯示狀態 8 以上的狀態時，顯示的圖片僅顯示狀態 7。 [按範圍更改狀態] 若勾選，將會按照設定的範圍來決定顯示狀態。若

設定的範圍有重疊的部分，將以優先權決定顯示的狀態。優先權為預設代表字元暫存器的數據不在任何設定的範圍之內。



#### 功能鍵說明

一個複合式多功能按鈕的清單最多可執行 20 個指令。

功能鍵	描述
	新增動作。
	對選取的指令進行刪除的動作。
	改變選取的指令的執行順序。
	對選取的指令進行複製的動作。
	對選取的指令進行貼上的動作。

#### 動作

##### 延遲

延遲 n 毫秒後才往下執行指令。一個複合式多功能按鈕僅可建立一個 [延遲] 指令。

##### 位元設定

將暫存器的狀態設定為 ON 或 OFF。

開關類型	描述
設為 ON	所指定暫存器的狀態將為 ON
設為 OF	所指定暫存器的狀態將被設定為 FF。
切換開關	按壓此物件後，所指定暫存器的狀態將被設定為反向。

##### 多狀態設定

可改變指定暫存器的數據。

開關類型	描述
設定常數	將常數寫入指定暫存器。
遞加 (JOG+)	加值功能。每按壓一次物件，所指定暫存器內的數據將加上 [遞加值] 中設定的增量值，但增值

		的結果將不超過 [上限值] 中的設定值。
遞減 (JOG-)		減值功能。每按壓一次物件，所指定暫存器內的數據將減去 [遞減值] 中設定的減量值，但減值的結果不會低於 [下限值] 中的設定值。
動態限制		上限 / 下限由指定暫存器設定。 假設動態限制位址為 LW-n，則若使用 [遞加 (JOG+)] 時，只需設定上限；反之當使用 [遞減 (JOG-)] 時，只需設定下限。

切換基本視窗

換頁至指定的視窗。

### 13.36.2.2. cMT / cMT X 系列

#### 一般屬性設定



[指示燈] 動作大致與 eMT、iE、XE、mTV 系列相同，請參考手冊 Ch13.36.2.1 說明。

設定動作時，可以先將動作進行分組，同一個群組中的動作會一起執行。當目前群組中的所有動作都觸發後，下一個群組的動作才會開始進行。動作群組的注意事項請參考手冊 Ch13 說明。

設定	描述
按下時動作	當手觸碰到按鈕時，立刻執行的動作。
釋放時動作	當手離開按鈕時，才執行的動作。
[延遲] 動作	延遲 n 毫秒後才往下執行指令。
[位元狀態設定]	將暫存器的狀態設定為 ON 或 OFF。
動作	<p><b>設為 ON</b></p> <p>所指定暫存器的狀態將被設定為 ON。</p> <p><b>設為 OFF</b></p> <p>所指定暫存器的狀態將被設定為 OFF。</p> <p><b>切換開關</b></p> <p>所指定暫存器的狀態將被設定為反向。</p> <p><b>復歸型</b></p> <p>當選擇復歸型時，會同時增加一個位元狀態設為 ON 的動作在 [按下時動作]，及一個將該位元狀態設為 OFF 的動作在 [釋放時動作]。</p>
[多狀態設定] 動作	<p>可改變指定暫存器的數據。</p> <p><b>寫入常數</b></p> <p>將常數寫入指定暫存器。</p> <p><b>遞加 (JOG+)</b></p> <p>加值功能。所指定暫存器內的數據將加上 [遞加值] 中設定的增量值，但增值的結果將不超過 [上限值] 中的設定值。</p> <p><b>遞減 (JOG-)</b></p> <p>減值功能。所指定暫存器內的數據將減去 [遞減值] 中設定的減量值，但減值的結果不會低於 [下限值] 中的設定值。</p> <p><b>動態限制 (JOG+,JOG-)</b></p> <p>JOG+上限 / JOG-下限由指定暫存器設定。</p> <p><b>寫常數字串</b></p> <p>將特定字串寫入指定暫存器。</p>
[切換基本視窗] 動作	<p><b>切換基本視窗</b></p> <p>換頁至指定的視窗。</p> <p><b>切換公共視窗</b></p> <p>切換公用視窗。</p> <p><b>返回上一個視窗</b></p> <p>返回前一頁基本視窗。例如當由 “視窗 10” 切換到 “視窗 20” 時，使用此功能可以再返回“視窗 10”。此功能只對基本視窗有效。</p> <p><b>動畫設定</b></p>

cMT / cMT X 系列可支援視窗的動畫效果，點選 [動畫設定] 後，能夠選擇的動畫效果有：淡出、飛入、飄入、擦去、分割、環狀、時鐘、縮放、旋轉、推入等，並可設定持續時間與顯示方向。

<b>[觸發巨集指令]</b>	觸發指定巨集。若群組中有巨集指令，只要巨集指令一被觸發，即會視為動作執行成功。即使巨集指令執行的時間較長，也不會等到該巨集全部執行完成才進行觸發下一個群組的動作。
<b>[彈出視窗] 動作</b>	彈出指定的視窗。
<b>[關閉當前開啟的視窗] 動作</b>	可關閉當前開啟的視窗。
<b>[顯示計算機視窗] 動作</b>	可直接呼叫出一個外觀固定的計算機物件，可以在人機上執行簡單計算並輸出至目標數值物件上。
<b>[鍵盤輸入] 動作</b>	<p>用來作為鍵盤的輸入訊號，主要用在 [數值] 與 [字元] 物件需要使用鍵盤來輸入數字或文字的場合。</p> <p><b>Enter:</b> 與鍵盤的輸入 (Enter) 動作相同。</p> <p><b>Backspace:</b> 與鍵盤的後退刪除 (Backspace) 動作相同。</p> <p><b>Clear:</b> 清除暫存器中已輸入的資料。</p> <p><b>Esc:</b> 與使用 [關閉視窗] 功能相同，可用來關閉彈跳出的鍵盤視窗。</p> <p><b>Delete:</b> 與鍵盤的刪除 (Delete) 動作相同，可將游標右方的一個字元刪除。</p> <p><b>Left:</b> 與鍵盤的←動作相同，可將游標向左移動一個字元。</p> <p><b>Right:</b> 與鍵盤的→動作相同，可將游標向右移動一個字元。</p> <p><b>Inc:</b> 將數值+1。</p> <p><b>Dec:</b> 將數值-1。</p> <p><b>ASCII/UNICODE:</b> 設定鍵盤的輸入字元。</p>
<b>[畫面擷取] 動作</b>	<p>可擷取當前的畫面，並選擇將畫面儲存至 USB 碟或 SD 卡。</p> <p>若 HMI 為無螢幕機型(cMT-SVR/cMT-SVRX)，搭配以下介面結果如下：</p> <p><b>cMT-iV6:</b> 可擷取當前的畫面至 cMT-iV6 的 SD 卡。</p> <p><b>cMT Viewer on PC:</b> 不支援 [畫面擷取] 動作。請直接使用滑鼠右鍵的[畫面擷取] 功能。</p> <p><b>cMT Viewer:</b> 不支援 [畫面擷取] 動作。</p>
<b>[確認所有事件 (報警)] 動作</b>	執行時可一次確認所有報警事件。
<b>[匯入資料] 動作</b>	<p>用來匯入進階安全的使用者帳號或 e-mail 的連絡人。也可設定為使用 USB 金鑰登入。</p> <p><b>資料位置：</b>提供從 USB 碟、SD 卡讀取兩個選項。</p> <p><b>帳號匯入方式：</b>選擇 [覆蓋]，HMI 內將只保存此次匯入的帳號資</p>

料，若是選擇 [附加]，HMI 內帳號資料將保留，並加入此次匯入的新帳號資料。

**匯入使用者帳號後刪除檔案**：將 USB 內的使用者帳號匯入後即刪除來源資料，可確保資料不洩漏。

<b>[等待觸發條件] 動作</b>	使用位元/字元設定不同觸發條件。當滿足觸發條件時，[複合式多功能按鈕] 才會執行下一個 [動作群組] 的動作。
<b>[資料傳輸 (背景)] 動作</b>	將指定位址中的數據傳送到其他位址中。
<b>檔案傳輸</b>	利用 FTP 傳輸檔案。HMI 將以被動模式連線 FTP 伺服器。 <b>[一般屬性] 設定頁</b>

### 設定傳輸方向

[下載] 將檔案從 FTP 伺服器下載至本機 HMI。

[上傳] 將檔案從本機 HMI 傳送至 FTP 伺服器。

### 設定伺服器位址

**[靜態]** 直接設定伺服器位址相關設定。

**[動態]** 指定一個動態位址，以在 HMI 執行時可透過特定位址動態調整伺服器設定。假設動態位址為 LW-n:

LW-n: 主機語法 (0:IP, 1:網域名稱)

LW-n+1: IP (4 字組)

LW-n+5: 連接埠號

LW-n+6: 認證 (0:停用, 1:啟用)

LW-n+7: 使用者名稱 (16 字組)

LW-n+23: 密碼 (16 字組)

LW-n+39: 網域名稱 (64 字組)

### 成功傳輸後刪除來源檔案

檔案傳輸後，會刪除來源檔案。當傳輸方向為下載(FTP -> HMI)時，需確保該帳號具有 FTP 伺服器寫入權限。

### [檔案設定] 設定頁

可設定檔案位置、FTP 伺服器端檔案路徑、本機 HMI 端檔案路徑。當路徑為資料夾時，則將傳輸資料夾內的所有檔案，但不包括子資料夾內的檔案。若檔案名稱相同時，一律覆蓋檔案。

### [狀態] 設定頁

設定顯示檔案傳輸執行結果及回應的位址。

關於 FTP 伺服器的回應列表可參考：

[https://en.wikipedia.org/wiki/List\\_of\\_FTP\\_server\\_return\\_codes](https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes)。

<b>[控制權] 動作</b>	取得或釋放控制權。 <b>控制權</b> 選擇控制權。 <b>行為模式</b> 設定取得或釋放控制權。 <b>最大等待時間</b> 當持續無法取得控制權，即會根據 [若未取得控制權] 執行設定的動作。
<b>[備份] 動作</b>	備份歷史資料或配方資料至外接儲存裝置，或是透過電子郵件或FTP 伺服器傳輸。
<b>[安全移除 USB 碟/SD 卡] 動作</b>	可安全的移除外接儲存裝置，確保資料的完整性。

### Note

- 視窗相關功能在一個複合式多功能按鈕僅可擇一使用，並且只能使用一次。視窗相關功能有[切換基本視窗]、[彈出視窗]、[關閉當前開啟的視窗]
- 一個複合式多功能按鈕最多可建立 20 個群組，每一個群組最多 20 個動作。
- 檔案傳輸功能的錯誤碼列表：

錯誤碼	說明
0	執行成功
1	下載時，HMI 路徑不存在
3	偵測不到 USB 隨身碟或 SD 卡
4	未輸入 HMI 或 FTP 路徑
5	上傳檔案不存在
8	操作被 FTP 服務器拒絕
9	USB 隨身碟或 SD 卡容量已滿
10	未知的錯誤

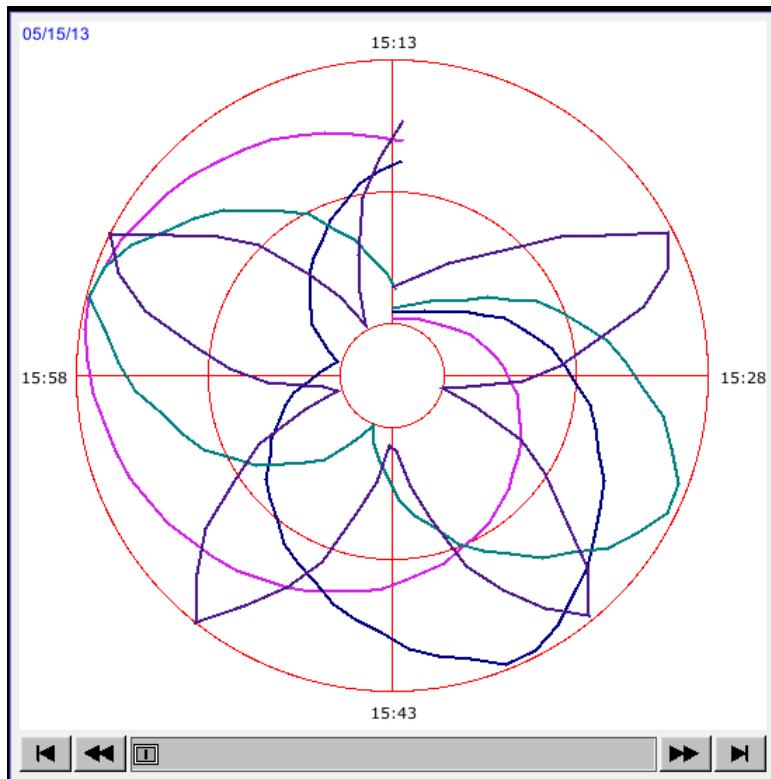


請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.37. 圓盤曲線圖

### 13.37.1. 概要

【圓盤曲線圖】可將【資料取樣】的取樣資料以極座標系統繪成圓盤曲線圖，半徑代表 y 分量，夾角代表 x 分量。使用上與【趨勢圖】雷同。



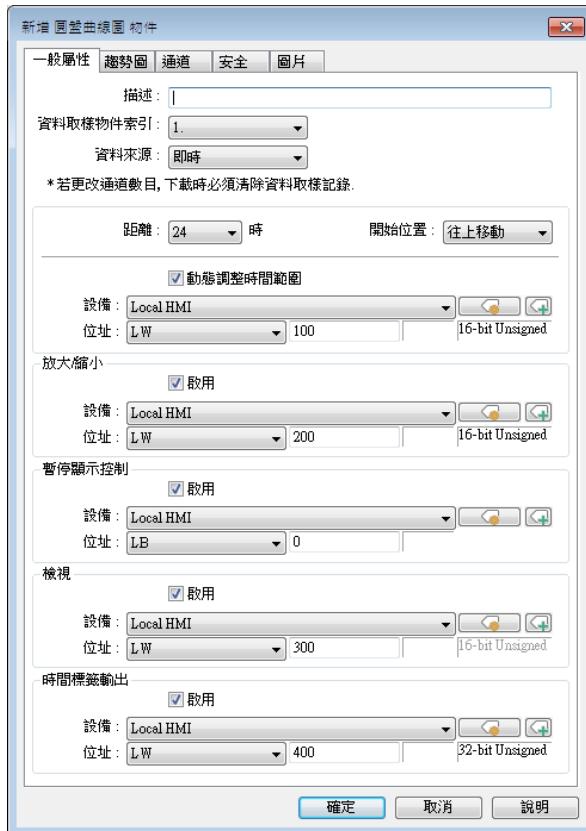
### 13.37.2. 設定



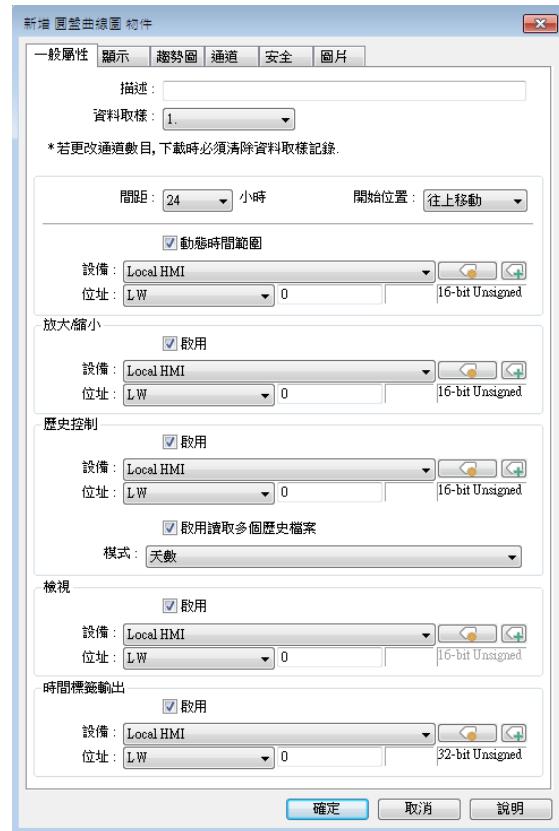
按下工作列上的【圓盤曲線圖】按鈕後即會開啟【圓盤曲線圖】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【圓盤曲線圖】物件。

## 一般屬性設定

eMT, iE, XE, mTV 系列



cMT, cMT X 系列



### 設定

### 描述

#### 資料取樣物件

##### 索引

選擇繪圖的數據來源。

#### 資料來源

##### (eMT, iE, XE,

##### mTV 系列支援)

選擇數據來源的形式，可以選擇 [即時] 或 [歷史]。

##### 即時

可顯示來自 [資料取樣] 物件從人機開機後，固定筆數的取樣資料。取樣資料的顯示數量於 [資料取樣] 物件的 [最大資料 (即時模式)] 中設定。當超過此設定的數量，則較舊的資料會從畫面上刪除。若需顯示他日或較舊的資料，需使用 [歷史] 模式。

可以利用 [暫停控制] 功能暫停物件畫面更新的動作，但僅暫停畫面刷新，並不會暫停 [資料取樣] 物件的取樣動作。

##### 歷史

歷史記錄來自 [資料取樣] 物件使用日期來分類並儲存的取樣資料。使用 [歷史] 模式可以利用 [資料取樣物件索引] 選定要顯示的歷史記錄，並利用 [歷史數據控制] 位址查看不同日期的歷史記錄。

##### 注意

若無啟用圓盤曲線圖設定頁 [使用畫面捲動控制按鈕] 功能，則當欲

顯示的取樣資料時間超過【距離】的時間時，即無法檢視之前的即時或歷史資料。例如：當【距離】設定為 1 小時，則此圓盤曲線圖即無法顯示 1 小時前的取樣資料。

#### 自動更新資料

若啟用，則每次開啟【歷史模式】的【圓盤曲線圖】物件所在的視窗時，物件畫面將會每秒自動更新。請注意：

- 自動更新功能的狀態可由畫面控制按鈕查看：

當圖示為 表示圓盤曲線圖物件的資料會自動更新。

當圖示為 表示圓盤曲線圖物件的資料停止更新。

- 當往前捲動查看較舊的資料時，會取消【自動更新資料】功能。此時控制按鈕的圖示為 .

- 當勾選【自動更新資料】時，無論先前是否曾藉由畫面捲動控制按鈕啟用或停止更新，切換回此視窗時，必定會自動更新畫面。

範例：假設物件已啟用【自動更新資料】，則往前捲動查看舊資料會停止自動更新功能。此時若切換至他頁再換回，則物件畫面仍會自動更新。

若製作工程檔案時，未啟用【自動更新資料】功能，而之後在人機上欲啟用時，只要按 按鈕即可。請注意此時的自動更新功能在切換視窗後就會被停用，也就是說，若切換至他頁再換回，則物件畫面仍停止更新。

#### 距離

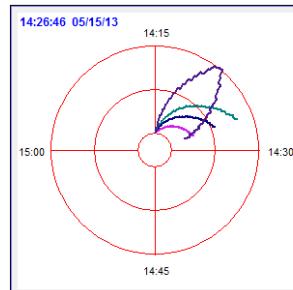
圓周一圈的時間長度。以小時為單位，長度範圍 1~24 (小時)。

#### 開始位置

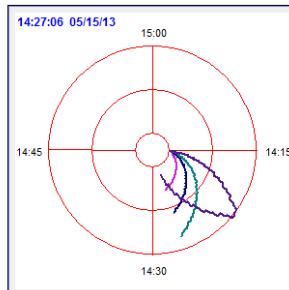
繪圖時的起始位置。

(往上移動)

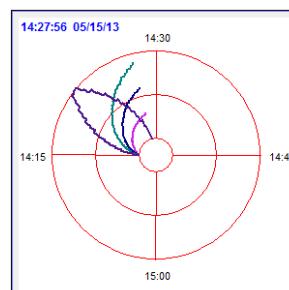
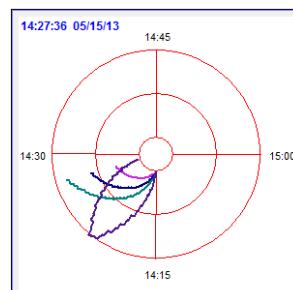
(往右移動)



(往下移動)



(往左移動)



#### 動態調整 X 軸

若啟用，可指定一個字組暫存器來線上動態調整【圓盤曲線圖】的時間範圍。

#### 時間範圍

調整的數據以小時為單位。若暫存器內無輸入任何數值，則

距離會採用預設值。

#### 放大/縮小

放大 / 縮小物件顯示畫面，最大可放大 10 倍。當暫存器內數值為 0 時，效果等同於 1，會顯示原尺寸。

#### 暫停控制

當暫停位元暫存器設為 ON 時，將暫停圓盤曲線圖畫面刷新，但不會暫停資料取樣物件的取樣動作。當 [資料來源] 使用 [即時] 模式時才有此選項。

#### 歷史數據控制

##### (eMT, iE, XE,

##### mTV 系列支

##### 援)

系統將歷史記錄檔案依時間先後順序編號，歷史控制即用來指定欲顯示的歷史記錄檔案。當歷史控制位址內的數值為 0，將顯示最新的檔案，若為 1，則顯示次新的檔案，依此類推。當 [資料來源] 使用 [歷史] 模式時才有此選項。可搭配 [項目選單] 物件，資料來源選擇 [歷史數據日期]，則所有的歷史資料會依照日期分類並顯示於項目選單物件上。詳細可參考手冊《13.29 項目選單》。

#### 範例

當設定歷史數據控制位址為 LW-n，且 [資料取樣] 物件已儲存的取樣資料檔案有四筆，分別為 20061120.dtl、20061123.dtl、0061127.dtl、20061203.dtl，則控制數據依序如下表所示：

LW-n 之數據	顯示的歷史資料取樣檔案
0	20061203.dtl
1	20061127.dtl
2	20061123.dtl
3	20061120.dtl

#### 歷史控制

##### (cMT, cMT X

##### 系列支援)

#### 啟用

與上方歷史數據控制相同。

#### 啟用讀取多個歷史記錄 - 天數

[歷史控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。

[歷史控制位址+2] 為總共顯示的天數，計算方式為從起始日索引往前計算。

#### 啟用讀取多個歷史記錄 - 指定天數

[歷史控制位址] 為目前選擇的起始日索引。0 代表今天，1 代表昨天，以此類推。

[歷史控制位址+2] 為結束日索引，數值需設定比起始日索引大。若起始日索引為 4，結束日索引為 7，則代表要顯示 4 天前到 7 天前的資料。

#### 檢視

可檢視當觸控在圓盤曲線圖物件上時，會產生一檢視線，並將檢視線上的數據輸出到指定的位址。若欲檢視多個通道的數據時，下一個通道點的數據將自動被載入至連續的字組暫存器中。若每個通道的資料格式不同，須依通道相對應的暫存器格式排列。

### 範例

當設定檢視位址為 LW-n，且資料取樣有以下四筆資料格式，分別為 16-bit Unsigned、32-bit Unsigned、32bit Signed、16-bit Signed，則檢視位址依序如下表：

通道	資料格式	資料長度	檢視位址
0	16-bit Unsigned	1 Word	LW-n
1	32-bit Unsigned	2 Words	LW-n+1
2	32bit Signed	2 Words	LW-n+3
3	16-bit Signed	1 Word	LW-n+5

### 時間標籤輸出

若啟用，系統以第一個取樣點的取樣時間作為時間原點並開始計數，並將最新取樣點之累計秒數輸出至 [時間標籤輸出位址 + 2]。當點選圓盤曲線圖物件上的曲線時，可將觸碰處最接近的取樣點之累計秒數輸出至 [時間標籤輸出位址]。

### 注意

[時間標籤輸出位址] 與 [時間標籤輸出位址 + 2] 皆須為 32-bit 格式。[時間標籤輸出位址 + 2] 只適用於即時模式的圓盤曲線圖，而 [時間標籤輸出位址] 適用於即時模式及歷史模式的圓盤曲線圖。

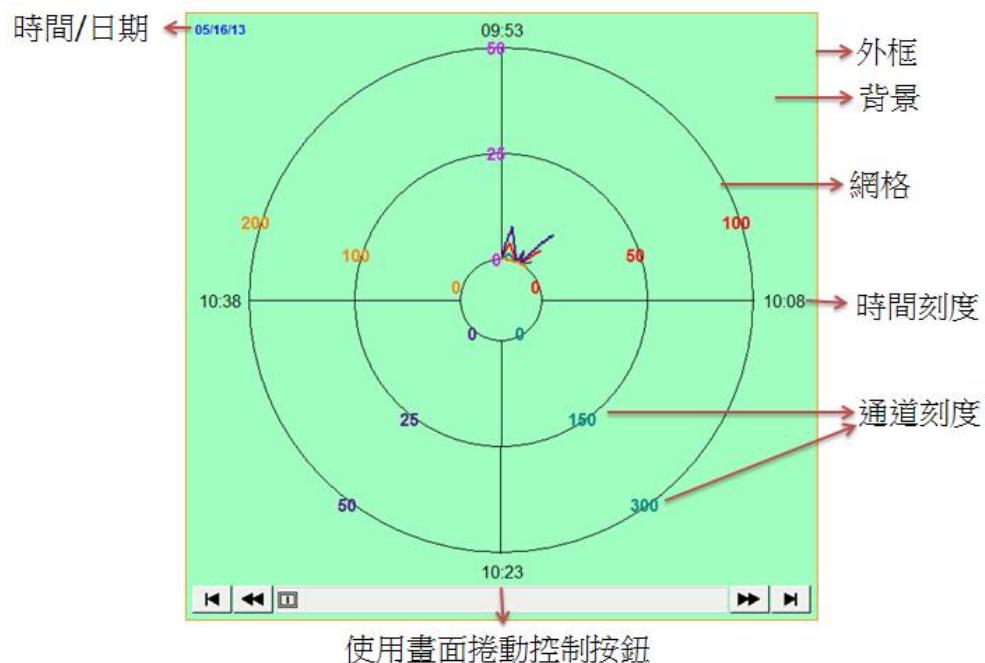
## 顯示設定



本頁設定僅支援於 cMT / cMT X 系列。

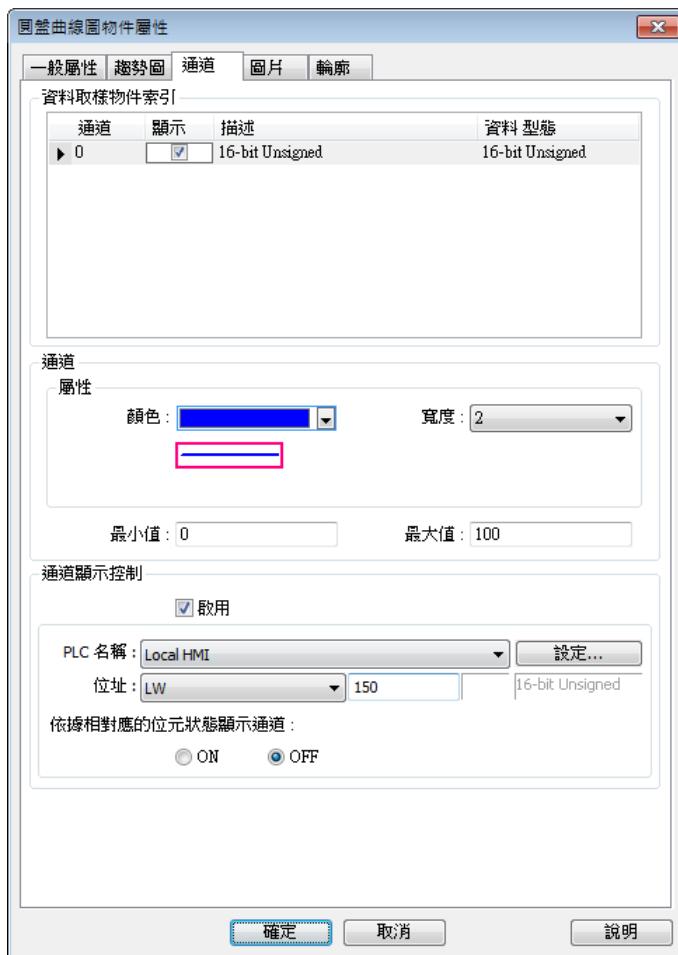
設定	描述
選項按鈕顯示	設定是否顯示物件上的選項按鈕
	設定是否顯示物件上的選項按鈕 。
	<ul style="list-style-type: none"> <li>● 預定義： 預先設定是否顯示選項按鈕。</li> <li>● 動態： 設定一個位元位址動態決定是否顯示選項按鈕。</li> </ul>
	<input checked="" type="radio"/> 預定義 <input type="radio"/> 動態
	設備 : Local HMI 位址 : LB 0

## 圓盤曲線圖設定



設定	描述
外觀	<p><b>外框</b> 物件的外框顏色。</p> <p><b>背景</b> 物件的背景顏色。</p> <p><b>使用畫面捲動控制按鈕</b> 可查看其他時間範圍的取樣資料，捲動時的最小單位是根據一般屬性設定頁中的【距離】。若無啟用【使用畫面捲動控制按鈕】功能，則當欲顯示的取樣資料時間超過一般屬性設定頁的【距離】的時間時，即無法檢視之前的即時或歷史資料。例如：當【距離】設定為 1 小時，則此圓盤曲線圖即無法顯示 1 小時前的取樣資料。</p>
網格	<p>設定格線的數目與顏色。</p> <p><b>Y 軸</b> 設定 Y 軸的區隔數目。</p> <p><b>字型尺寸</b> 時間標籤及通道刻度標籤的文字尺寸。</p> <p><b>時間刻度</b> 設定是否顯示時間標籤。當時間範圍大於 1 小時，時間標籤的間隔以 1 小時為單位；當時間範圍小於一小時，時間標籤的間隔以 15 分鐘為單位。</p> <p><b>通道刻度</b> 設定是否顯示通道刻度。各通道刻度的文字顏色會依照通道線條所設定的顏色來顯示。</p>
時間 / 日期	<p><b>時間</b> 設定顯示的時間格式。</p> <p><b>日期</b> 設定顯示的日期格式。</p>

## 通道設定



### 設定

### 描述

#### 通道

設定各個曲線的樣式與顏色，與曲線所能描繪數據的上下限值。最多可同時支援 8 個通道。

##### 未勾選 [取自暫存器]

數據的上限與下限由常數設定。

##### 勾選 [取自暫存器]

數據的上限與下限由指定暫存器設定。當寫入位址為 LW-n，則上/下限會根據以下的規則自動被設定為：

位址格式	16-bit	32-bit
下限	LW-n	LW-n
上限	LW-n+1	LW-n+2

#### 通道顯示控制

當選擇使用 [通道顯示控制後]，則此位址中的各個位元將會被用來控制各個通道的顯示與否。Bit-0 控制通道 0，Bit-1 控制通道 1，依此類推。

##### 依據相對應的位元狀態顯示通道

若設定為 ON，則當相對應的位元狀態設為 OFF 時隱藏通道；若設定

為 OFF，則相對應的位元狀態設為 ON 時隱藏通道。

### 範例

當設定通道顯示控制位址為 LW-0 且當相對應的位元狀態設為 OFF 時開啟通道，此時若有五個通道，則會依以下的規則被控制是否顯示：

通道編號	控制地	位狀態	是否顯示
0	LW_bit-000	OFF	YES
1	LW_bit-001	ON	NO
2	LW_bit-002	ON	NO
3	LW_bit-003	OFF	YES
4	LW_bit-004	OFF	YES

## 13.38. 圖片檢視

### 13.38.1. 概要

【圖片檢視】物件可播放外接裝置，例如 USB 碟或 SD 卡內的圖片檔案。

若使用 cMT/cMT X 系列 HMI，無法透過 cMT Viewer 檢視此物件。

### 13.38.2. 設定



按下工作列上的【圖片檢視】按鈕後即會開啟【圖片檢視】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【圖片檢視】物件。

#### 一般屬性設定



設定	描述
<b>檔案位置</b>	選擇檔案的來源位置為 USB 碟或 SD 卡。
<b>目錄</b>	設定圖片檔案存放的目錄，可搭配影像輸入物件的影像擷取功能，或是功能鍵物件的畫面擷取功能，並立即播放儲存於外接儲存裝置的圖片。
	<b>動態資料夾路徑</b>
	可使用本機位址指定目錄路徑名稱。
<b>檔案選擇</b>	<p><b>自動顯示最新產生的圖片</b> 當指定的目錄夾內有新圖片產生時，圖片檢視物件將會自動顯示最新的圖片。</p> <p><b>當切換至新圖片時發送通知</b> 當啟用【自動顯示最新產生的圖片】，若 HMI 切換至最新的圖片，指定的暫存器會被設為 On/Off。</p> <p><b>由位址指定檔案(隱藏工具列)</b> 啟用此選項，將由本機位址指定欲讀取圖片之檔案名稱。工具列將被隱藏。</p>

### Note

- 圖片的檔案名稱僅能使用 ASCII 字元，不支援 Unicode。
- 支援的圖片檔案格式為.jpg, .bmp, .gif, .png。
- iP/iE/eMT/XE/mTV 機型可顯示的圖片尺寸大小，為  $(\text{image.width} \times \text{image.height}) < (\text{screen.width} \times \text{screen.height}) \times 4$ ，不符合的圖片尺寸大小將無法顯示於 HMI。
- 具有電容式螢幕的人機，支援兩指放大縮小功能。

## 外觀設定



設定	描述
<b>外觀</b>	調整工具列的位置、物件的背景顏色、文字的顯示字型。(只支援於 iP/iE/eMT/XE/mTV 機型)
	<b>隱藏刪除按鈕</b> 啟用後，將不會顯示刪除按鈕於工具列上。刪除按鈕可用來刪除檢視

的圖片。

**當圖片尺寸小於物件尺寸時，以圖片原尺寸顯示**

啟用後，當圖片小於此物件時，會使用圖片原尺寸顯示，可避免圖片因為被放大導致圖像顯示失真。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.39. 檔案瀏覽器

### 13.39.1. 概要

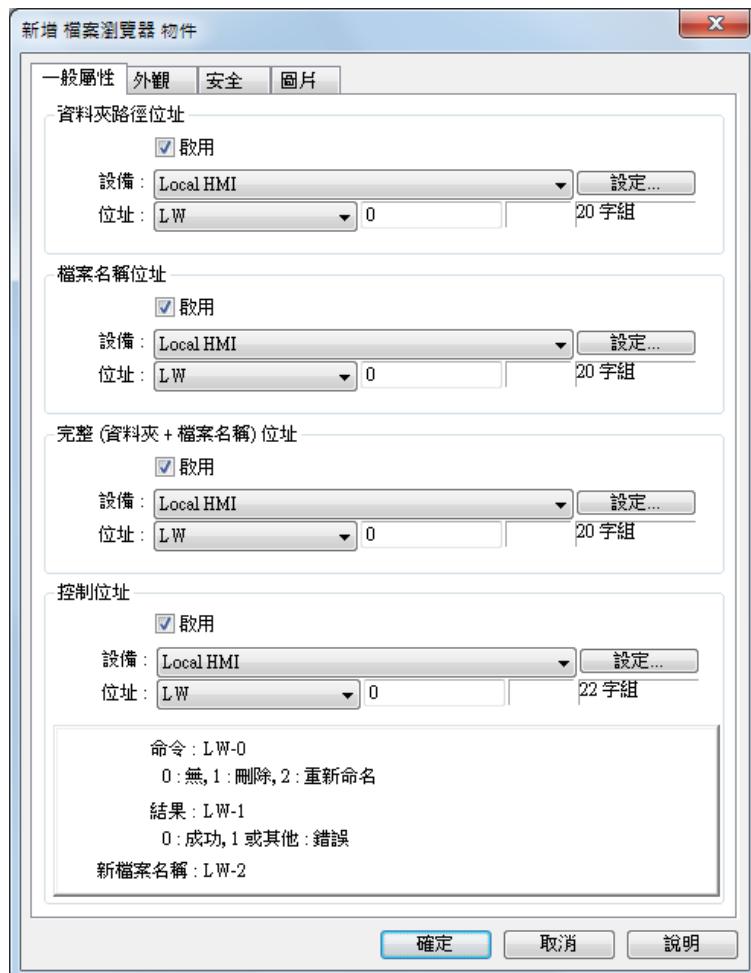
[檔案瀏覽器] 物件可用於顯示 SD 卡或 USB 碟中的檔案名稱及其資料夾路徑。除瀏覽裝置中的檔案之外，在 [檔案瀏覽器] 中點選的檔案的檔案名稱及路徑會寫入特定的位址。  
若使用 cMT / cMT X 系列 HMI，無法透過 cMT Viewer 檢視此物件。

### 13.39.2. 設定



請直接點擊 [檔案瀏覽器] 圖示建立此物件，或點選工具列上的 [物件] » [檔案瀏覽器] 新增此物件。

#### 一般屬性設定



設定	描述
資料夾路徑位址	當前的資料夾路徑。
檔案名稱位址	當前檔案瀏覽器選取的檔案名稱。
完整(資料夾 + 檔案 名稱) 位址	當前所選取的檔案所在的完整路徑 + 檔案名稱。
控制位址	<p>設定相關的控制位址用來刪除或重新命名檔案瀏覽器中的檔案項目。</p> <p><b>命令：控制位址</b></p> <p>0: 無 1: 刪除 2: 重新命名 3: 選擇 (只支援於 cMT / cMT X 系列)</p> <p><b>結果：控制位址 + 1</b></p> <p>0: 成功 1 或其他: 錯誤</p> <p><b>新檔案名稱：控制位址 + 2</b></p>

## 外觀屬性設定



設定	描述
檔案位置	選擇讀取自 SD 卡或 USB 碟。
檔案類型	選擇可顯示的檔案。可選擇顯示全部檔案或是 CSV 檔案或是圖像檔案。cMT / cMT X 系列另可選擇 PDF 檔案，用於搭配 PDF 檢視器使用。
字型、背景、顏色	設定物件及文字的屬性。



### Note

- 被點選之檔案的檔案名稱及路徑會寫入設定的位址。若欲透過更改設定的位址來改變檔案瀏覽器上的檔案選擇，可先輸入檔案所在位置後，再輸入選擇控制命令(=3)，即可讓檔案瀏覽器指向該檔案。
- 重新開機或者剛插入儲存裝置時，系統會嘗試讀取資料夾路徑位址以及檔案名稱位址，並且切換目錄然後選擇設定位址所指向的檔案。若未啟用資料夾路徑位址，則會讀取完整位址。



請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.40. 汇入/汇出

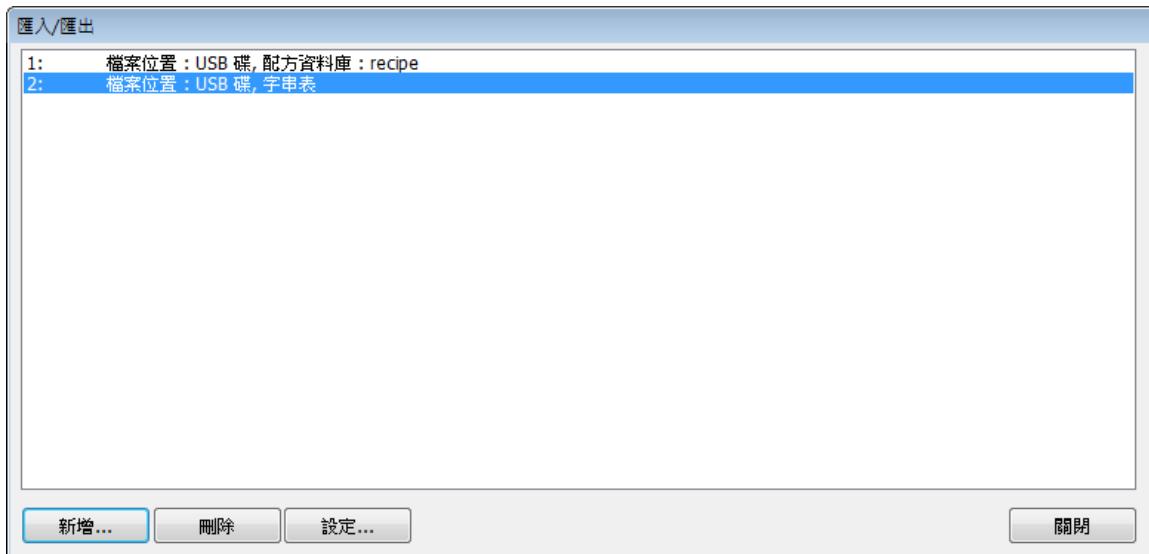
### 13.40.1. 概要

[汇入/汇出] 物件可以汇入/汇出配方资料库或是字串表。

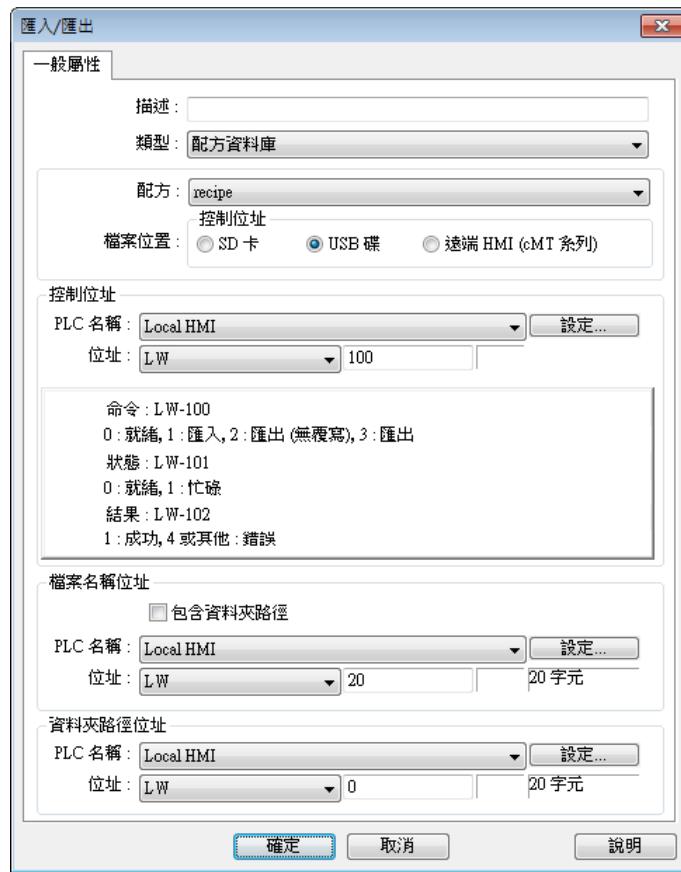
### 13.40.2. 設定



按下工具列上的 [汇入/汇出] 按钮后即会出现 [汇入/汇出] 物件管理对话窗，接著可按下 [新增] 按键，並利用出現的 [汇入/汇出] 物件設定对话窗正確設定物件的各項屬性，最後按下確定鍵即可新增一個 [汇入/汇出] 物件。



## 一般屬性設定



設定	描述
類型	檔案來源，可選擇為配方資料庫或是字串表。
檔案位置	選擇匯入/匯出自 SD 卡、USB 碟或是遠端 HMI (cMT / cMT X 系列)。當檔案位置設定為遠端 HMI 時，僅支援遠端 HMI 為 cMT / cMT X 系列。
配方	選擇使用的配方。若是選擇字串表，則無此選項。
控制位址	<p>設定相關的控制位址用來執行匯入/匯出，以及顯示執行的結果。</p> <p><b>控制:</b> 控制位址</p> <p>配方資料庫:</p> <ul style="list-style-type: none"> <li>0: 就緒</li> <li>1: 匯入</li> <li>2: 匯出 (無覆寫)</li> <li>3: 匯出</li> </ul> <p>字串表:</p> <ul style="list-style-type: none"> <li>0: 就緒</li> <li>1: 刪除</li> </ul>

- 2: 汇入  
 4: 汇出 (無覆寫)  
 5: 汇出  
**狀態: 控制位址+1**  
 0: 就緒  
 1: 忙碌  
**結果: 控制位址+2**  
 1: 成功  
 4: 檔案已存在；未執行覆寫  
 其他: 錯誤

檔案名稱位址	匯入/匯出使用的檔案名稱。若勾選包含資料夾路徑，則此位址將包含完整的路徑及檔案名稱。
資料夾路徑位址	匯入/匯出使用的檔案所在的檔案路徑。
遠端 HMI 位址	當檔案位置設定為遠端 HMI (cMT / cMT X 系列) 時，需要輸入遠端 HMI 的 IP 至此控制位址。

## 範例 1

以下範例展示如何設定配方匯入/匯出功能。依下表所示輸入設定：

欄位	設定值
檔案位置	USB 碟
配方	Recipe_A (或其他配方)
控制位址	LW-100
檔案名稱位址	LW-200
資料夾路徑位址	LW-250

- 加入兩個 [字元輸入] 物件，位址分別設定為 LW-200 和 LW-250。
- 在 LW-200 輸入檔案名稱：2015\_recipe.csv。
- 在 LW-250 輸入資料夾路徑：Setting。
- 使用 [多狀態設定] 物件，寫入數值 3 至 LW-100，即可匯出配方 Recipe\_A 至 USB 碟中的 Setting/2015\_recipe.csv 檔案中。

### Note

- 執行匯出 (無覆寫) 時，若目標檔案已經存在，則取消此次匯出，結果數值會被設定為 4。以下表格列出各錯誤碼及其表示的狀態：

錯誤碼 (HEX)	狀態
0x1	成功執行
0x4	匯出時，指定的檔案已存在
0x10	無效的指令
0x100	欄位含有非數字的資料
0x101	路徑不能包含 “..”(上一層目錄)
0x102	更新 Recipe DB 資料時發生通訊錯誤
0x103	從 Project 檔案讀取 Recipe DB 資訊時發生錯誤
0x200	執行期間未知的錯誤
0x201	執行期間未知的內部錯誤
0x202	匯入資料的格式未知
0x203	Recipe DB 的定義表格檢查錯誤
0x204	Recipe DB 的資料表格檢查錯誤
0x205	Recipe DB 的定義表格寫入錯誤
0x206	Recipe DB 的資料表格寫入錯誤
0x300	檔案錯誤：未知的錯誤
0x301	檔案錯誤：檔名為空白
0x302	檔案錯誤：外部裝置不存在
0x303	檔案錯誤：指定位置並非檔案（目錄或特殊檔案）或檔案名稱與現有的資料夾名稱相同
0x304	檔案錯誤：無法刪除指定檔案
0x305	檔案錯誤：檔案開啟錯誤
0x306	檔案錯誤：無法判讀的 BOM 檔頭
0x307	讀取 CSV 檔案錯誤（不正確的資料格式）
0x308	檔案錯誤：外部裝置空間不足
0x309	檔案錯誤：找不到檔案
0x30A	檔案錯誤：CSV 檔案中的資料超過 10000 行
0x400	資料庫未知的錯誤
0x401	資料庫錯誤：無法開啟表格
0x402	資料庫錯誤：無法開啟資料列
0x403	DB 和匯入 CSV 資料欄位數量不一致
0x501	連接遠端 HMI 失敗
0x503	遠端 HMI 資料庫禁止匯入
0x504	遠端 HMI 資料庫不支援匯入



請點選此圖示觀看教學影片，請先確定已連上網路線。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.41. 圓餅圖

### 13.41.1. 概要

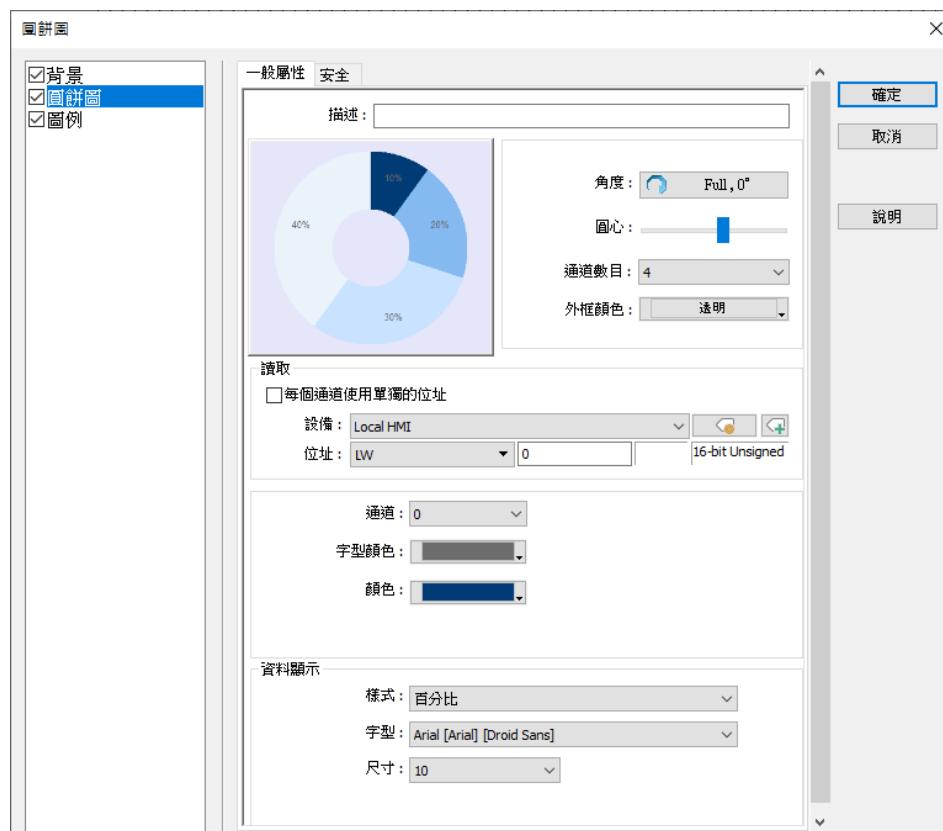
透過輸入資料於指定的字組位址後，以圓餅圖的方式來顯示各通道所佔之比例。

### 13.41.2. 設定



按下工作列上的【圓餅圖】按鈕後設定各項屬性並按下確定，即可新增一個【圓餅圖】物件。

#### 一般屬性設定



#### 設定

##### 角度

#### 描述

##### 預設

可以從縮圖中輕鬆選擇配置方向和角度範圍。

##### 自訂

必須自行設定配置方向以及角度。若未選擇【全圓】，則必需設定【結束角度】。

**圓心**

設定圓心的範圍尺寸。

**通道數目**

設定欲檢視的通道數目，範圍: 2 ~ 16。

**外框顏色**

設定圓餅圖外框的顏色。

**讀取位址**

此設定為通道 1 的位址，一個通道表示一個位址，且位址須為連續性的。若讀取位址設定為 LW-0，則通道 2 的讀取位址將為 LW-1，通道 3 的讀取位址將為 LW-2，以此類推。

**每個通道使用單獨的位址**

啟用後可於各通道設定讀取位址。

**通道**

選擇通道編號來設定各通道區塊的顯示樣式，可設定的有：[字型顏色]、[背景顏色]、[圖案顏色]、[圖案樣式]。此處的 [背景顏色] 是指圖案樣式的背景顏色。

**資料顯示**

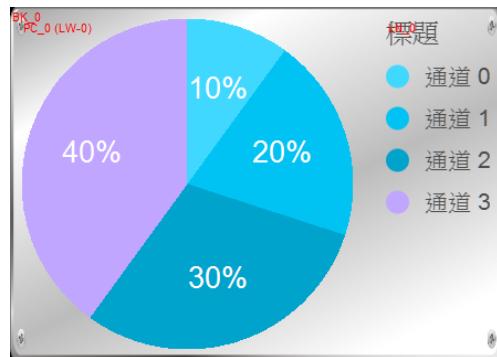
設定資料於圓餅圖的顯示樣式。可選擇的有：[無]、[數據]、[百分比]。

在圓餅圖顯示文字資料時，可以設定欲顯示資料的字型與字型尺寸。

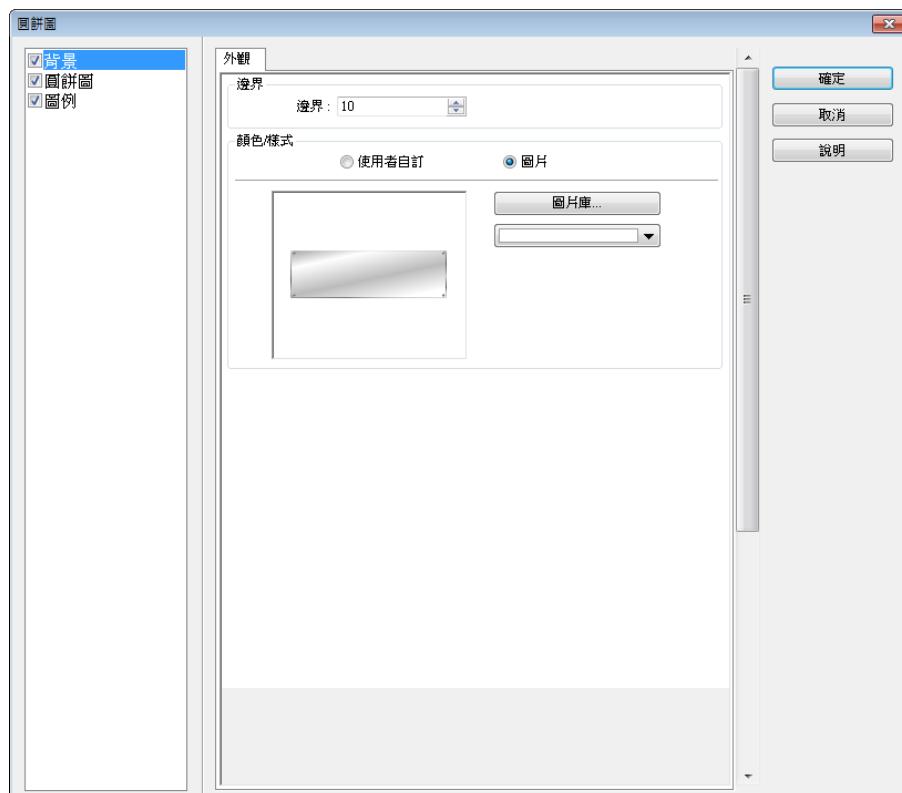
在圓餅圖顯示 [數據] 時，可以設定欲顯示的小數點後個位數。

### 13.41.3. 複合物件

cMT / cMT X 系列人機的複合式設定，提供一次性設定相關元件的功能。除了圓餅圖以外，增加背景與圖例的元素供使用者更能活用及美化圓餅圖的設計。



#### 背景設定



設定	描述
邊界	邊緣與物件的留白距離。
顏色/樣式	使用者自訂

使用者自訂

顏色樣式

倒角 : 10

外框 : 透明

背景 :

圖案 :

圖案樣式 :

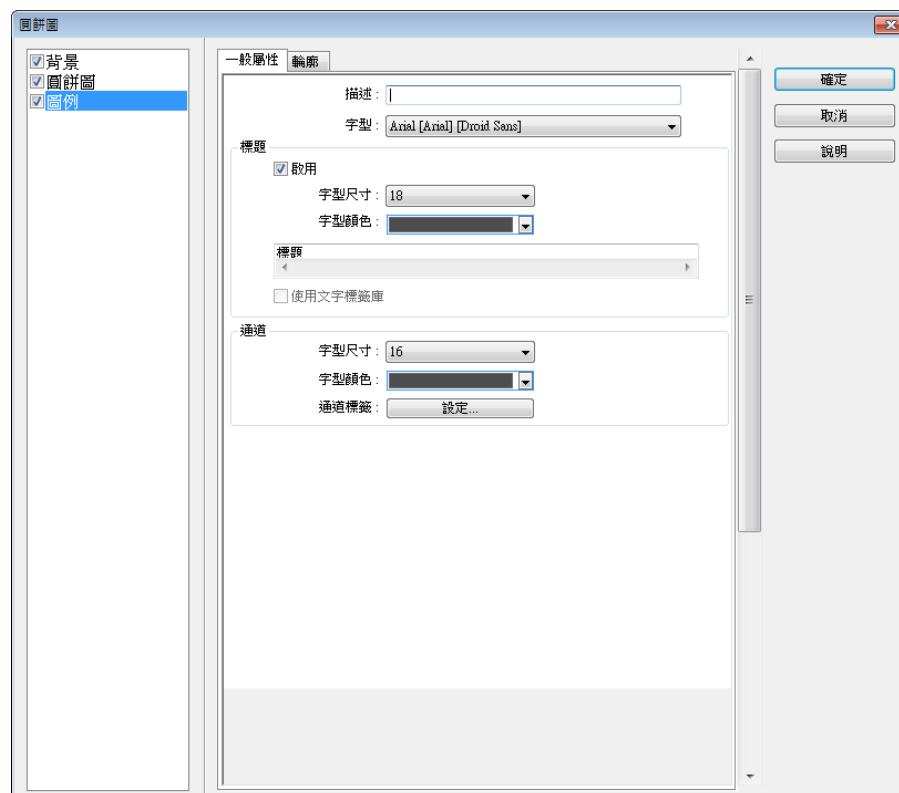
根據圖案樣式與顏色挑選自訂背景。

### 圖片



可使用內建的背景圖片或是圖片庫中的檔案。

## 圖例設定



### 設定

### 描述

#### 標題

可選擇是否要顯示標題，且可用文字標籤庫。

#### 通道

各通道顯示的名稱可在此設定，使用文字標籤庫時，通道數正好對應狀態數。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.42. 條碼

### 13.42.1. 一維條碼顯示

#### 13.42.1.1. 概要

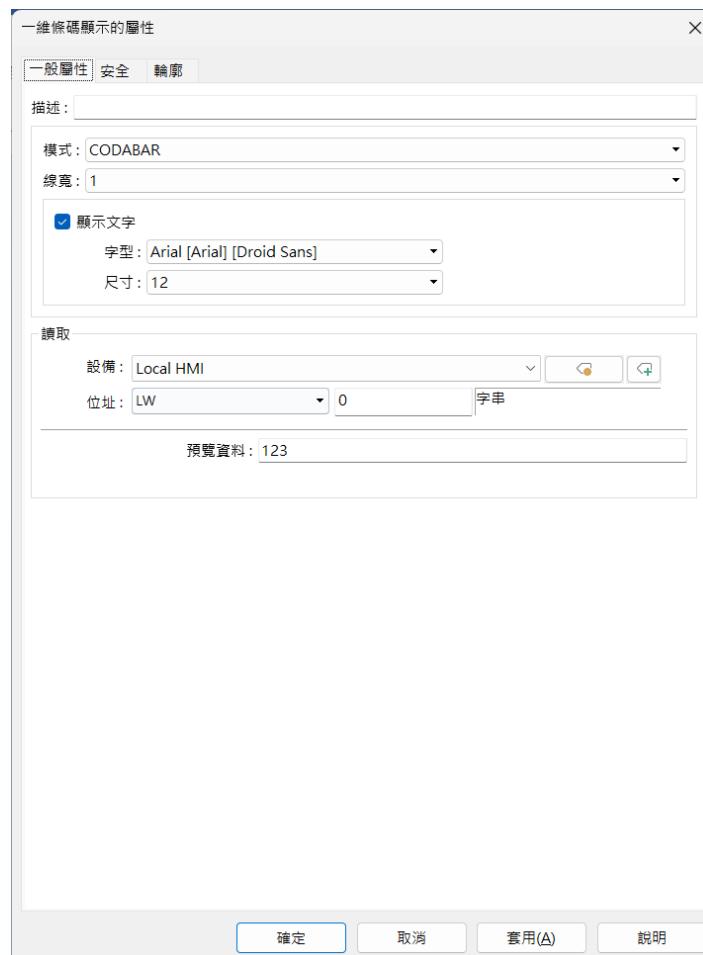
在指定的字組位址輸入資料後，系統將生成相應的一維條碼於視窗中，以供掃描使用。

#### 13.42.1.2. 設定



按下工作列上的【一維條碼顯示】按鈕後即會開啟【一維條碼顯示】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【一維條碼顯示】物件。

#### 一般屬性設定



#### 設定

#### 描述

#### 模式

支援以下條碼類型：CODABAR, CODE 39, CODE 93, CODE 128, EAN

	8, EAN 13。
線寬	設定一維條碼的線條寬度。
顯示文字	可選擇是否在一維條碼下方展示對應的文字內容。
字型	設定顯示文字的字型。
尺寸	設定顯示文字的尺寸大小。
讀取位址	在指定的字組位址輸入資料後，系統將生成相應的一維條碼於視窗中，以供掃描。字組資料長度範圍為 1 至 80。
預覽資料	此預覽資料將被轉換為條碼，方便專案畫面的設計與預覽。

## 13.42.2. 二維條碼顯示

### 13.42.2.1. 概要

透過輸入資料於指定的字組位址後，可產生相對於資料的二維條碼於視窗供掃描。

### 13.42.2.2. 設定



按下工作列上的 [二維條碼顯示] 按鈕後即會開啟 [二維條碼顯示] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [二維條碼顯示] 物件。

## 一般屬性設定



設定	描述										
<b>模式</b>	支援條碼類型：QR code, Aztec code, Data Matrix										
<b>校正標準</b>	二維條碼有容錯能力，條碼圖形如果有破損，仍然可以被機器讀取內容。 <b>QR code</b> 錯誤修正容量分為四種：L、M、Q、H，請見以下說明：										
	<table border="1"> <thead> <tr> <th colspan="2">錯誤修正容量</th> </tr> </thead> <tbody> <tr> <td>L</td><td>7% 的字碼可被修正</td></tr> <tr> <td>M</td><td>15% 的字碼可被修正</td></tr> <tr> <td>Q</td><td>25% 的字碼可被修正</td></tr> <tr> <td>H</td><td>30% 的字碼可被修正</td></tr> </tbody> </table> <b>Aztec coed</b> 錯誤修正容量依比例計算，可調整範圍是 5%~95%。當錯誤修正容量設定愈高，條碼的像素尺寸相對也需更大，才能有準確的辨識率。	錯誤修正容量		L	7% 的字碼可被修正	M	15% 的字碼可被修正	Q	25% 的字碼可被修正	H	30% 的字碼可被修正
錯誤修正容量											
L	7% 的字碼可被修正										
M	15% 的字碼可被修正										
Q	25% 的字碼可被修正										
H	30% 的字碼可被修正										
<b>顏色</b>	設定二維條碼顯示的顏色。										
<b>UNICODE</b>	在預設中，二維條碼的內容是以 ASCII 編碼形式產生。若勾選此選項，則二維條碼的內容會以 UNICODE 的編碼形式產生。										

預設 ASCII 編碼適用於一般英文字母與數字形式的內容；勾選 UNICODE 則可以使用符合 UNICODE 編碼的其他文字，例如中文、韓文等。

**高低位元組轉換** 正常情況下，二維條碼的顯示順序為 [高位元組] + [低位元組]。

勾選此功能後，則顯示順序改為 [低位元組] + [高位元組]。

**讀取位址** 透過輸入資料於指定的字組位址後，可產生相對於資料的二維條碼於視窗供掃描。QR code 可輸入資料的字組長度為 1 ~ 1024，Aztec code 與 Data Matrix 則為 1 ~ 2048。

### 13.42.3. 推播通知二維條碼顯示

#### 13.42.3.1. 概要

**[推播通知二維條碼顯示]** 物件可透過二維條碼連結到通訊軟體帳號後，將該人機的所有觸發事件推播到手機的通訊軟體上。

#### 13.42.3.2. 設定



按下工作列的 **[物件] » [條碼] » [推播通知二維條碼顯示]** 按鈕後即會開啟 **[新增推播通知條碼顯示]** 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 **[推播通知二維條碼顯示]** 物件。

## 一般屬性設定



設定	描述										
<b>模式</b>	支援條碼類型：QR code, Aztec code, Data Matrix										
<b>校正標準</b>	二維條碼有容錯能力，條碼圖形如果有破損，仍然可以被機器讀取內容。										
<b>QR code</b>	錯誤修正容量分為四種：L、M、Q、H，請見以下說明： <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">錯誤修正容量</th> </tr> </thead> <tbody> <tr> <td>L</td><td>7% 的字碼可被修正</td></tr> <tr> <td>M</td><td>15% 的字碼可被修正</td></tr> <tr> <td>Q</td><td>25% 的字碼可被修正</td></tr> <tr> <td>H</td><td>30% 的字碼可被修正</td></tr> </tbody> </table>	錯誤修正容量		L	7% 的字碼可被修正	M	15% 的字碼可被修正	Q	25% 的字碼可被修正	H	30% 的字碼可被修正
錯誤修正容量											
L	7% 的字碼可被修正										
M	15% 的字碼可被修正										
Q	25% 的字碼可被修正										
H	30% 的字碼可被修正										
<b>Aztec code</b>	錯誤修正容量依比例計算，可調整範圍是 5%~95%。當錯誤修正容量設定愈高，條碼的像素尺寸相對也需更大，才能有準確的辨識率。										
<b>顏色</b>	設定二維條碼顯示的顏色。										
<b>UNICODE</b>	預設無法修改。										
<b>高低位元組轉換</b>	預設無法修改。										

讀取位址 預設無法修改。

## 系統參數設定



設定	描述
<b>EasyAccess 伺服器</b>	全球伺服器支援微信、Facebook 以及 LINE 的事件推播功能。中國地區的伺服器只支援微信事件推播功能。

## 推播通知設定

將檔案下載至人機後會顯示推播通知二維條碼，請參考以下連結在微信、Facebook 以及 LINE 軟體上設定推播通知 (V2.8 與更新版本)。<https://support.ihmi.net/zh-TW/ea20/release-notes>

#### 13.42.4. 條碼掃描器(Android 攝影機)

##### 13.42.4.1. 概要

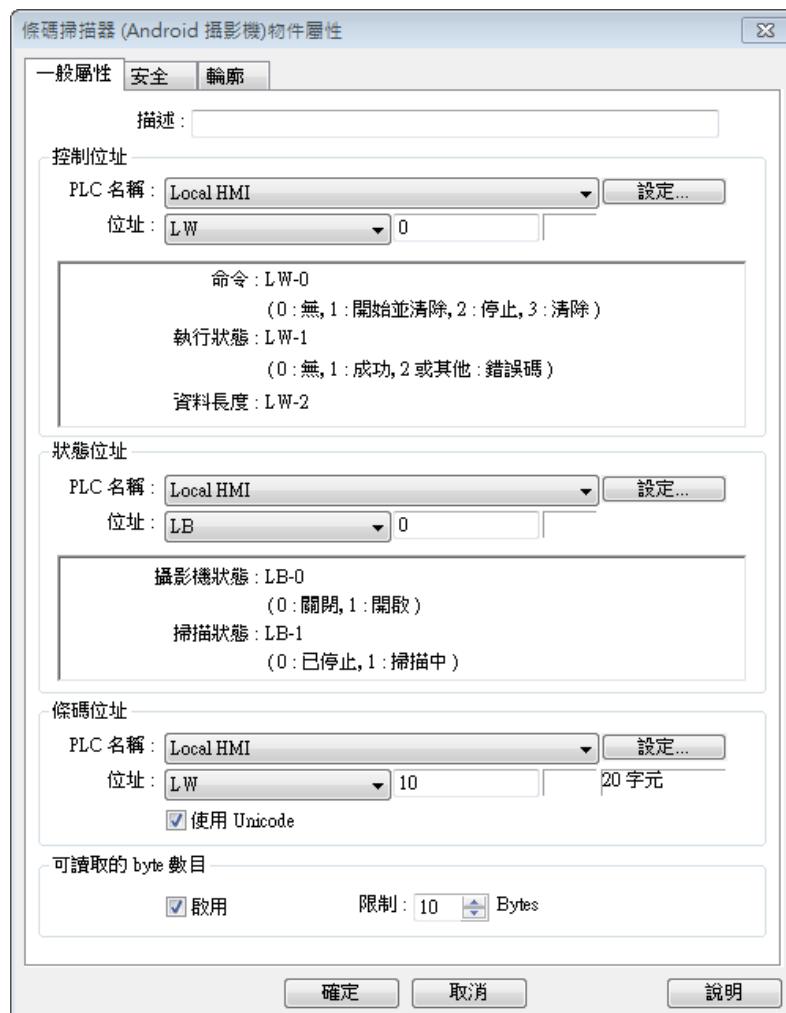
部分人機型號支援透過安卓系統裝置 (智慧手機/平板電腦) 的鏡頭，並將裝置上的 cMT Viewer 連上 cMT / cMT X 後，即可用行動裝置掃描一維以及二維條碼。

##### 13.42.4.2. 設定



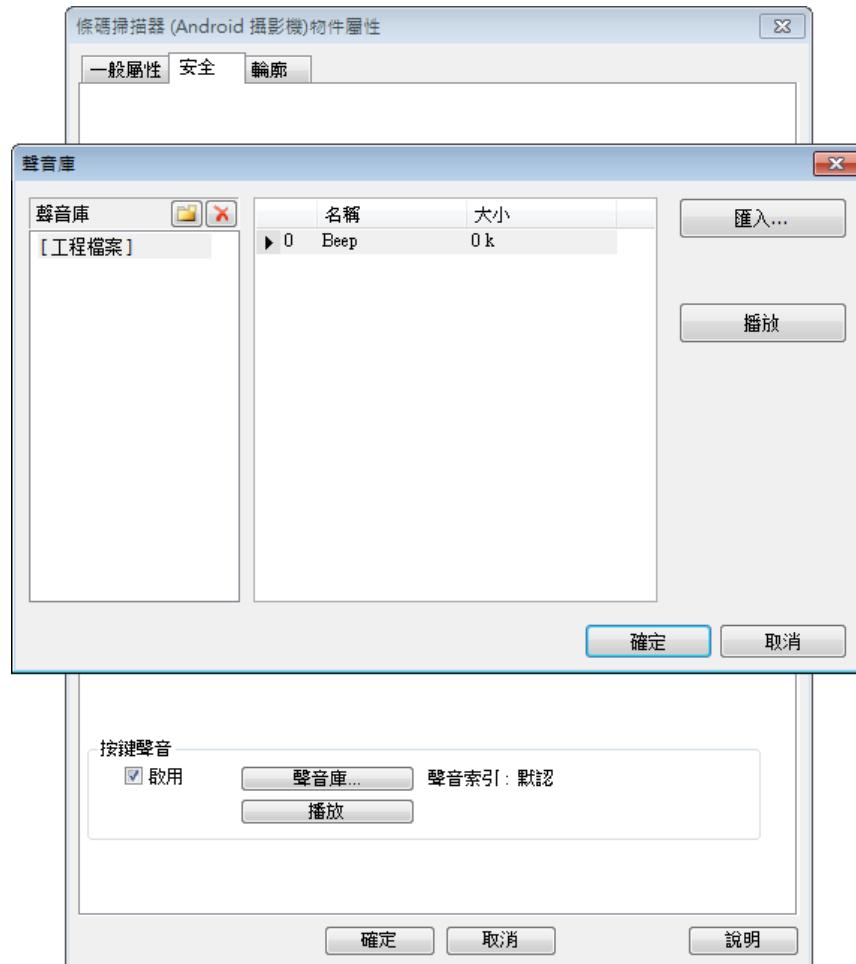
按下工具列上的 [條碼掃描器] 按鈕，或是從[物件] » [條碼] 選單中點選 [條碼掃描器] 開啟物件設定視窗，設定適當的屬性按下確定鍵，即完成一個 [條碼掃描器] 物件。

#### 一般屬性設定



設定	描述
控制位址	<p>控制位址：定義下指令的位址 0: 無 1: 開始並清除 2: 停止 3: 清除</p> <p>控制位址 +1：顯示執行狀態 0: 無 1: 成功 2 或其他: 錯誤碼</p> <p>控制位址 +2：顯示資料長度資訊</p>
狀態位址	<p>狀態位址：表示相機鏡頭為開啟或關閉 0: 關閉 1: 開啟</p> <p>狀態位址 +1：表示是否可以開始掃描 0: 已停止 1: 掃描中</p>
條碼位址	選擇讀取條碼資料存放的位址，可使用 Unicode。
可讀取的 byte 數目	若讀取大小超過此設定，執行狀態的值將變為 2(錯誤碼)。

## 安全設定



### 設定

#### 聲音

### 描述

可設定讀取到結果時是否要播放聲音。支援音檔為.wav。

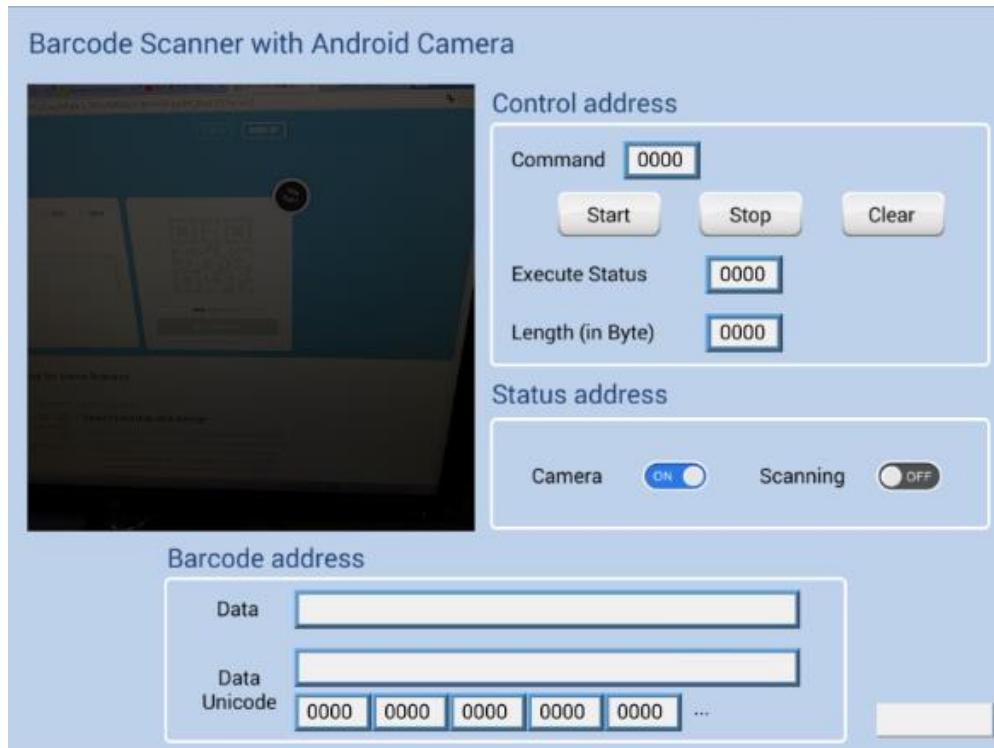
#### Note

- 條碼掃描器物件無法使用模擬或是 cMT Viewer 開啟。
- 支援 EAN/UPC, Code 128, Code 39, Interleaved 2 of 5 and QR Code。
- 行動裝置上，如果其他應用程式使用錄影或鎖定相機，可能會造成 cMT Viewer 不正常運作。
- 設計工程檔案時，若有多台 cMT Viewer 連線，由於地址是共用的，多台 cMT Viewer 觀看同一頁時會同時進行掃描。

## 範例

以下範例展示實際使用平板電腦掃描 QR 碼的過程。

1. 初始狀態畫面，鏡頭是偏暗的狀態。



2. 按下 Start 按鈕後，Scanning 狀態將會開啟，鏡頭由暗轉亮時就可以開始掃描。



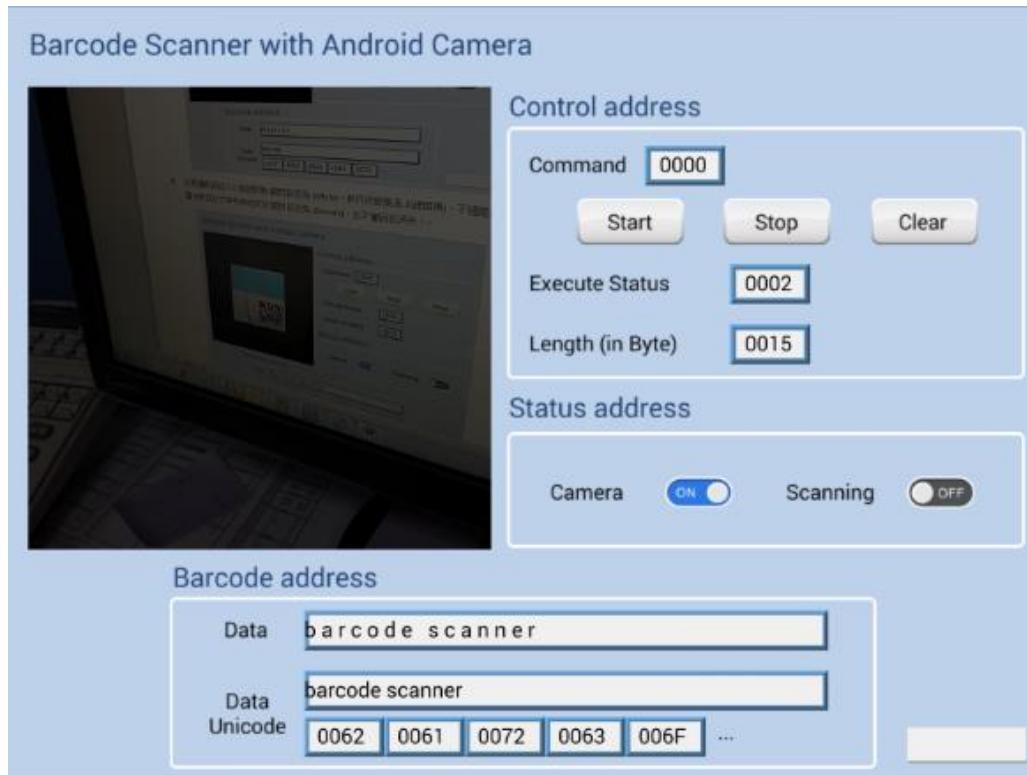
3. 成功讀取到資料後 (執行狀態為 1)，條碼圖案會被截取，內容會顯示在條碼位址，另外也可以使用 Unicode 編碼。



4. 如果讀取資訊大小超過限制 (範例設定為 10 bytes)，執行狀態將變為 2 (錯誤碼)，不過超過限制的顯示取決於字元物件的設定 (範例設定為 20 words)，並不會因此消失。



5. 換頁之後，Scanning 會轉為關閉，參數會保留上一次的結果直到下一次按下開啟 (Start) 或清除 (Clear) 按鈕。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.43. 字串表

### 13.43.1. 概要

透過字串表可在 HMI 運行時，利用預先定義的編號來動態調整文字，也可應用於多國語言的環境。

### 13.43.2. 設定



按下 [工程檔案] » [字串表] 後即會開啟 [字串表] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [字串表] 物件。

#### 一般屬性設定



設定	描述
表格	字串表目錄。 [新表格] 建立一個新的字串表。 [刪除表格] 刪除選擇的字串表。
新增	新增一筆字串。
設定	設定所選字串的內容。
匯出為 CSV 檔案	儲存所有文字表格為 .CSV 格式檔案。
匯入 CSV 檔案	將現存文字表格 .CSV 檔案匯入字串表。
匯出為 EXCEL 檔案	儲存所有文字表格為 .XLS 格式檔案。
匯入 EXCEL 檔案	將現存文字表格 .XLS 檔案匯入字串表。



#### Note

- 字串表多國語言的文字字型需在 [文字標籤庫] 中設定。

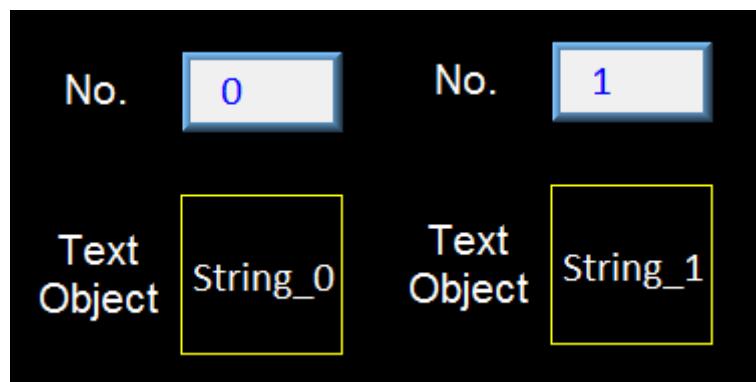
- 所有表格的字串總筆數上限為 10000。

## 範例 1

1. 建立一個字串表如上圖所示。
2. 建立一個 [文字] 物件，並使用字串表。字串 ID 設定為動態且讀取位址設定為 LW-0。



3. 建立一個 [數值] 物件，位址設定為 LW-0。
4. 當 LW-0 = 0，顯示編號 0 的字串；當 LW-0 = 1，顯示編號 1 的字串。



## 13.44. 資料庫

### 13.44.1. 資料庫伺服器

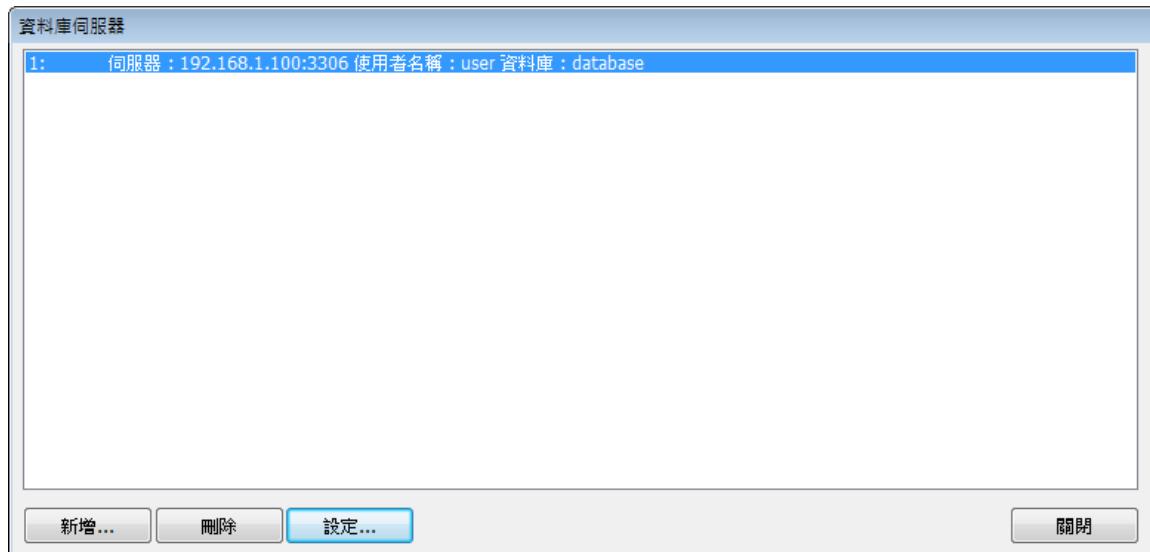
#### 13.44.1.1. 概要

[資料庫伺服器] 物件可連結 MySQL 或 MS SQL 資料庫。使用者可將資料取樣/事件記錄透過 [資料庫伺服器] 傳送至資料庫或是搭配 SQL 查詢功能存取資料庫中的數據。

#### 13.44.1.2. 設定



請直接點擊 [資料庫伺服器] 圖示建立此物件，或點選工具列上的 [資料/歷史] » [資料庫伺服器] 新增此物件。



## 一般屬性設定



設定	描述
伺服器系統	支援 MySQL 與 MS SQL。
使用 IP	輸入資料庫的 IP 位址。
使用網域名稱	<p>使用網域名稱 <input type="button" value="▼"/></p> <p>網域名稱: 127.0.0.1</p> <p>支援使用網域名稱指定伺服器。</p>
使用伺服器名稱	<p>伺服器系統: MS SQL 伺服器 <input type="button" value="▼"/></p> <p>使用伺服器名稱 <input type="button" value="▼"/> e.g. HOSTSQLEXPRESS</p> <p>伺服器名稱: PC_NAME\INSTANCE_NAME</p> <p>此功能只有 MS SQL 支援。Instance 代表通訊的連接埠號，可藉此分辨一台電腦上的多台資料庫伺服器。 伺服器名稱支援下列格式：</p> <ul style="list-style-type: none"> <li>● &lt;Computer name&gt;\&lt;Instance name&gt;</li> <li>● &lt;Computer name&gt; (連到預設 Instance : MSSQLSERVER)</li> <li>● &lt;IP address&gt;\&lt;Instance name&gt;</li> <li>● &lt;IP address&gt; (連到預設 Instance : MSSQLSERVER)</li> </ul>

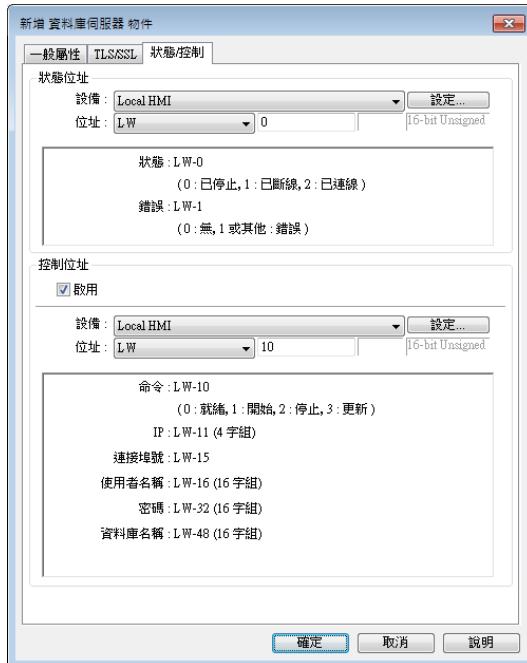
<b>連接埠號</b>	輸入資料庫的連接埠號。
<b>使用者名稱</b>	連線至資料庫所需的使用者名稱，最多可達 32 個字。
<b>密碼</b>	連線至資料庫所需的密碼，最多可達 32 個字。
<b>資料庫名稱</b>	輸入欲收集歷史資料的資料庫名稱。
<b>自訂超時</b>	可自訂以下操作的超時設定。
<b>連線超時</b>	嘗試連接到資料庫伺服器的超時設定。嘗試連線至指定時間後，就停止嘗試連線。
<b>讀取超時</b>	當讀取資料庫伺服器時的超時設定。
<b>寫入超時</b>	當寫入資料庫伺服器時的超時設定。

**TLS/SSL**

設定	描述
<b>啟用</b>	啟用 TLS/SSL 安全傳輸機制。可手動選擇加密版本 TLS 1.0, TLS 1.1 或 TLS 1.2。
<b>伺服器認證</b>	若勾選，驗證該伺服器憑證是否曾經經過憑證授權機構 (CA) 認證。伺服器憑證是在建立連線時由伺服器送出。 <b>伺服器名稱需與憑證資訊相符</b> 驗證該網域名或 IP 是否與憑證內的紀錄符合。網域

名與 IP 紀錄是記載於憑證內的 Subject Alternative Name 中。

## 狀態/控制設定



### 設定

#### 狀態位址

### 描述

LW-n: 顯示 [資料庫伺服器] 連線狀態

數值	描述
0	停止連線資料庫
1	無法連線資料庫
2	成功連線資料庫

LW-n+1: 錯誤提示

數值	描述
0	無錯誤
1	不明原因錯誤
2	無法連線資料庫
3	權限不足，資料庫拒絕連線
4	資料庫名稱錯誤
5	無效的網域名稱
6	超過連線數錯誤

#### 控制位址

LW-n: 控制 [資料庫伺服器] 執行或停止

數值	描述
0	就緒

1	開始
2	停止
3	更新

LW-n+1: 設定資料庫的 IP 位址

LW-n+5: 設定資料庫的連接埠號

LW-n+6: 設定連線至資料庫所需的使用者名稱

LW-n+22: 設定連線至資料庫所需的密碼

LW-n+38: 設定欲收集歷史資料的資料庫名稱

### Note

- 若資料取樣成功同步到 SQL 資料庫，則資料庫會產生三個資料表，資料取樣則是存放在 \*\_data。

產生的資料表	描述
<HMI NAME>_<DATALOG NAME>_data	存放資料取樣的數據
<HMI NAME>_<DATALOG NAME>_data_format	系統內部使用
<HMI NAME>_<DATALOG NAME>_data_section	系統內部使用

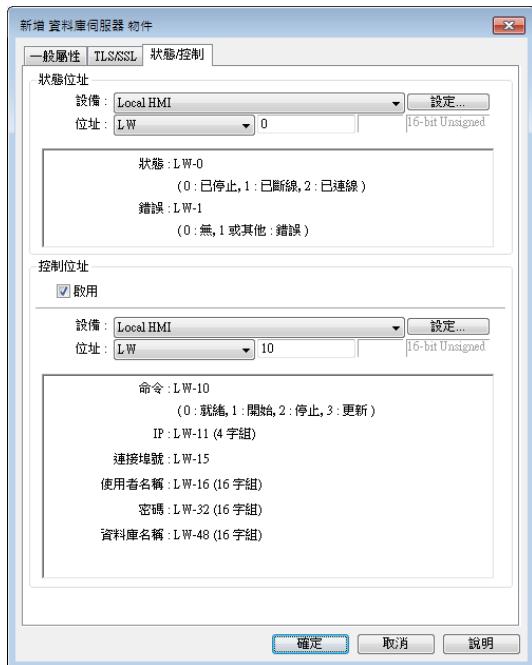
- 若是使用事件記錄，則資料庫產生的三個資料表如下，事件記錄是存放在 \*\_event。

產生的資料表	描述
<HMI NAME>_event	存放事件記錄的數據
<HMI NAME>_event_log	系統內部使用
<HMI NAME>_event_update_time	系統內部使用

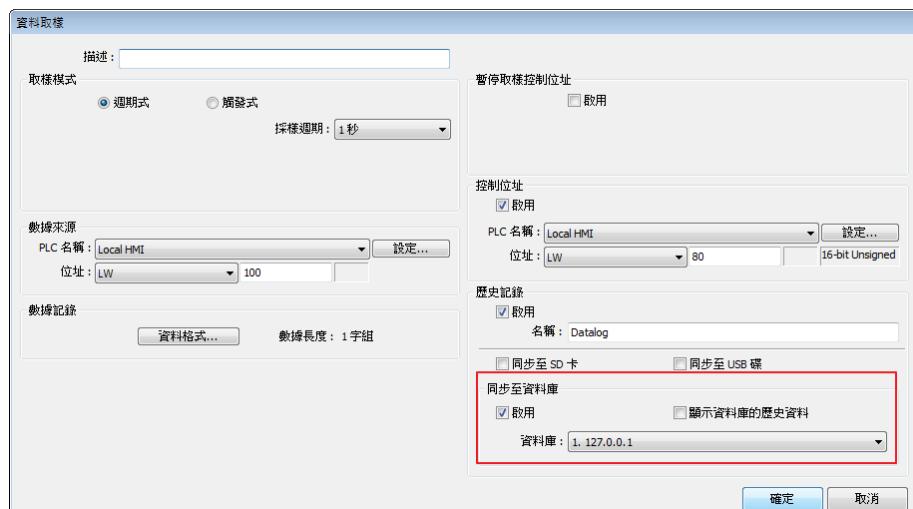
- 當再次異動資料取樣/事件記錄的內容，例如資料格式或是事件訊息內容，並下載至 HMI，則在 SQL 資料庫需先刪除已建立的三個同步資料表，新的資料取樣/事件記錄才有效。
- 支援 Azure SQL Server，伺服器系統需選擇 MS SQL 並使用網域名稱。Azure SQL Server 有防火牆相關設定，須將連入的 Client 加入白名單。請參考以下連結設定：  
<https://docs.microsoft.com/zh-tw/azure/azure-sql/database/single-database-manage#manage-an-existing-server>

### 範例 1

- 啟用 [資料庫伺服器]，狀態位址設定為 LW-0，控制位址設定為 LW-10。



- 2.** 建立一個資料取樣，歷史記錄欄位選擇 [同步至資料庫]，並啟用 [控制位址] LW-80 以控制 HMI 歷史資料的同步與清除。



- 3.** 若資料庫正確連結，則狀態位址 LW-0 會顯示 2 (已成功連接)，錯誤位址 LW-1 會顯示為 0 (無錯誤)。
  - 4.** 在 LW-80 寫入 2 (同步資料)。開啟 SQL 資料庫，在<HMI NAME>\_<DATALOG NAME>\_data 資料表即可看到數據。

	資料表	執行	記錄	型態	校對	大小	多餘
<input type="checkbox"/>	hostname_datalog_data		6	MyISAM	utf8_unicode_ci	2.1 KB	-
<input type="checkbox"/>	hostname_datalog_data_format		1	MyISAM	utf8_unicode_ci	2.0 KB	-
<input type="checkbox"/>	hostname_datalog_data_section		0	MyISAM	utf8_unicode_ci	1.0 KB	-
3 資料表		總計	7	MyISAM	utf8_unicode_ci	5.2 KB	0 Bytes



請點選此圖示觀看教學影片，請先確定已連上網路線。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.44.2. SQL 查詢

### 13.44.2.1. 概要

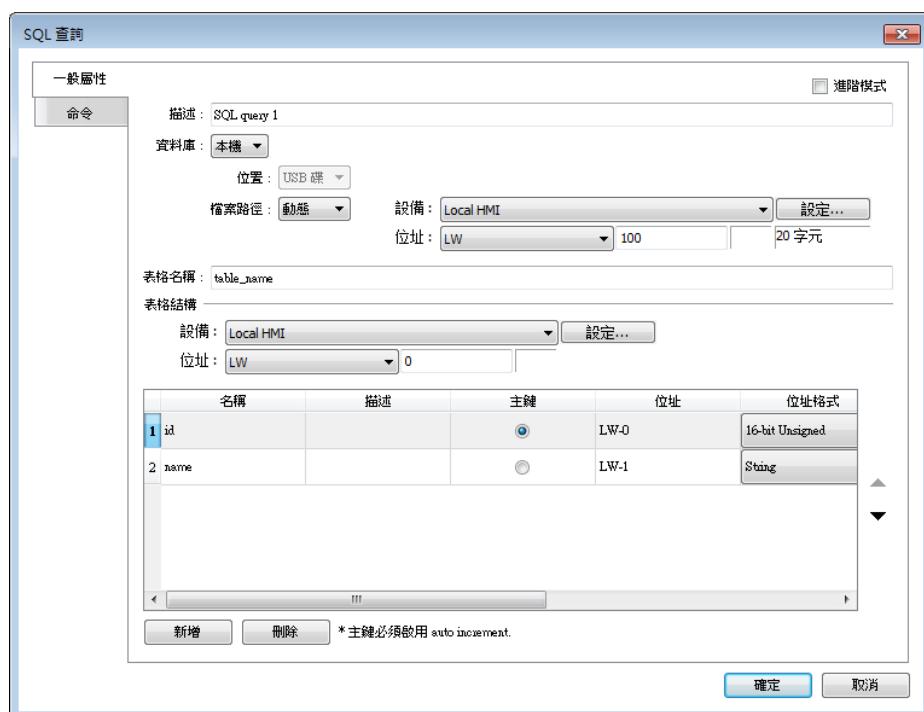
[SQL 查詢] 功能可與 SQL 上的資料庫進行資料交換。啟用 [SQL 查詢] 的功能之前，必須先設定 [資料庫伺服器]。

### 13.44.2.2. 設定



按下工作列的 [資料/歷史] » [SQL 查詢] 按鈕後即會出現 [SQL 查詢] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [SQL 查詢] 項目。

#### 一般屬性設定



設定	描述
進階模式	<b>不勾選 [進階模式]</b> 點選 [新增] 後可新增一筆欄位，或使用 [從伺服器匯入] 從資料庫直接匯入所有欄位。
<b>勾選 [進階模式]</b>	若勾選進階模式，則必須在 [命令] 分頁手動輸入語法以操作 MySQL/MS SQL 資料庫上的記錄。請注意，當勾選進階模式時，將無法再重設為一般模式。
描述	使用者可為此物件描述相關訊息。
資料庫	選擇資料庫的來源。

本機：來源為連接在 HMI 上的 USB 隨身碟或 SD 卡裡的 SQLite 資料庫。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

遠端：來源為 [資料庫伺服器] 物件中所設定的資料庫。

#### 檔案路徑

當選擇使用本機的資料庫時，可設定使用 [靜態] 路徑或 [動態] 路徑。使用靜態路徑時，需先輸入路徑至此設定對話窗。若使用動態路徑，則路徑來源會參考指定的位址。

#### 表格名稱

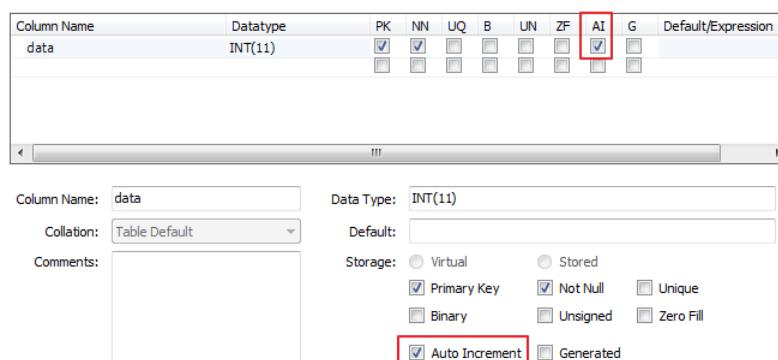
欲讀取的資料庫的表格名稱。

#### 表格結構

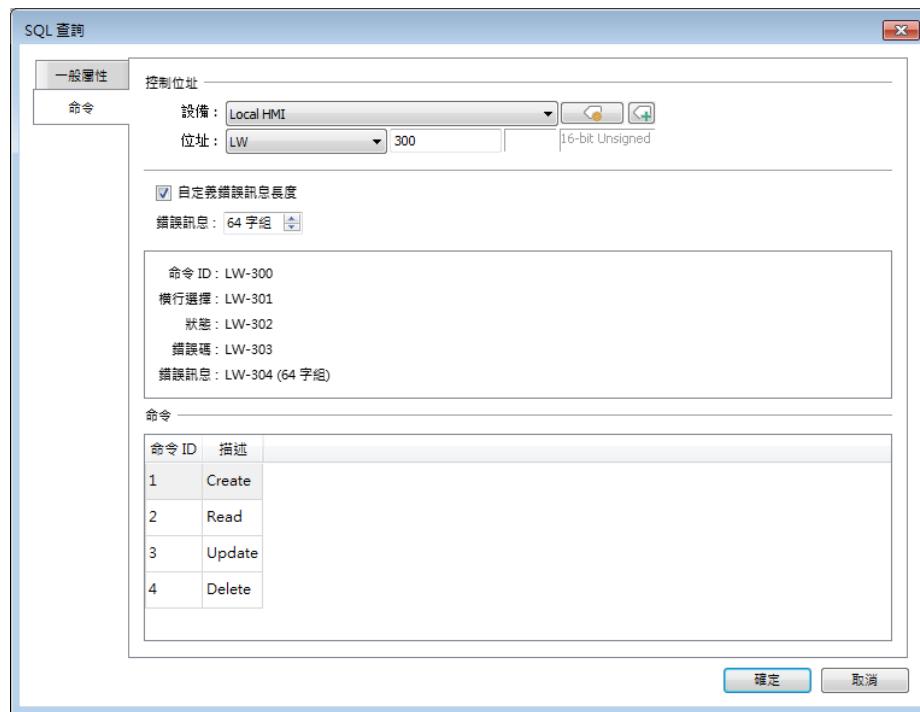
從資料庫讀取到表格後，會將欄位的資訊依序填入 [表格結構] 中設定的位址。當從資料庫讀取完畢後，[位址格式] 需要再手動調整。

### Note

- SQL 查詢物件的主鍵必須為數值資料型態，不支援字元型態。
- 在 MySQL 軟體中，主鍵必須啟用 Auto Increment。



## 命令設定



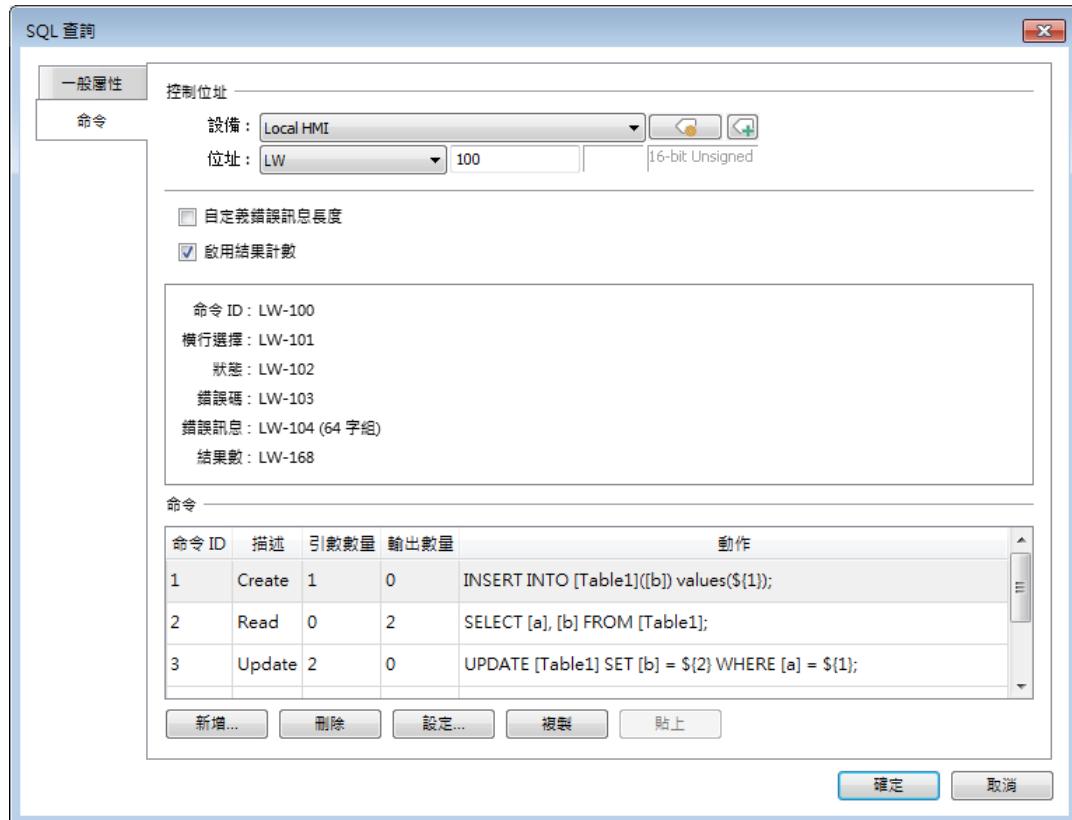
設定	描述
控制位址	控制位址的連續五個暫存器將用來執行 SQL 的命令及顯示執行結果，其中錯誤訊息的長度可以自行定義。
命令	在一般模式之下(不勾選進階模式)，命令視窗顯示 SQL 四種基本命令 (Create, Read, Update, Delete)，以及對應的命令 ID。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

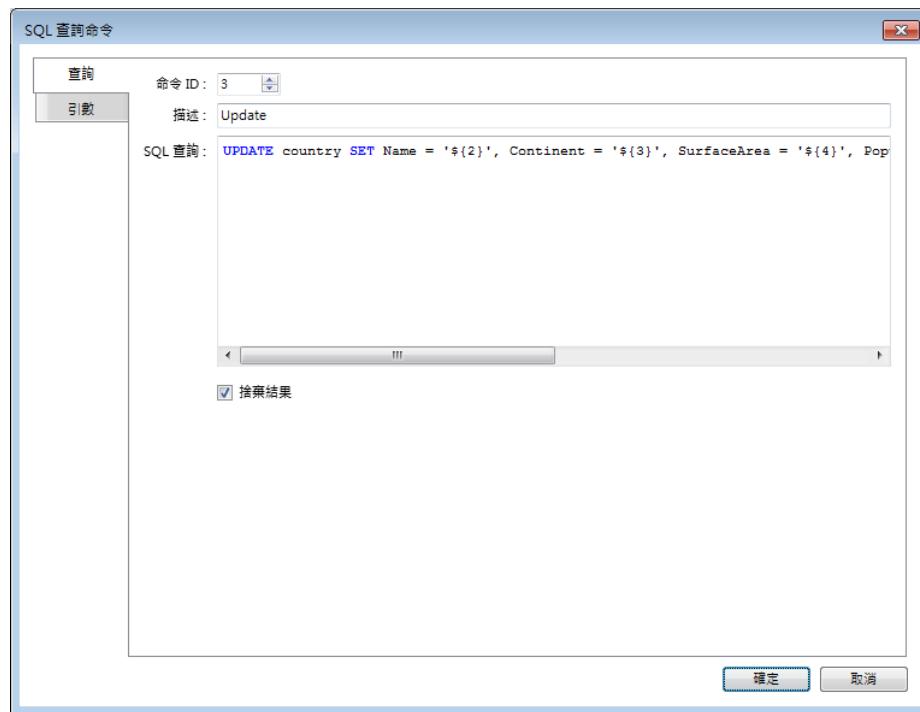
### 13.44.2.3. 進階模式

若勾選使用進階模式，在 [命令] 欄位會增加 [弓[數數量]、[輸出數量]、[動作] 三個欄位，亦可選擇是否批次輸出結果到連續位址中。



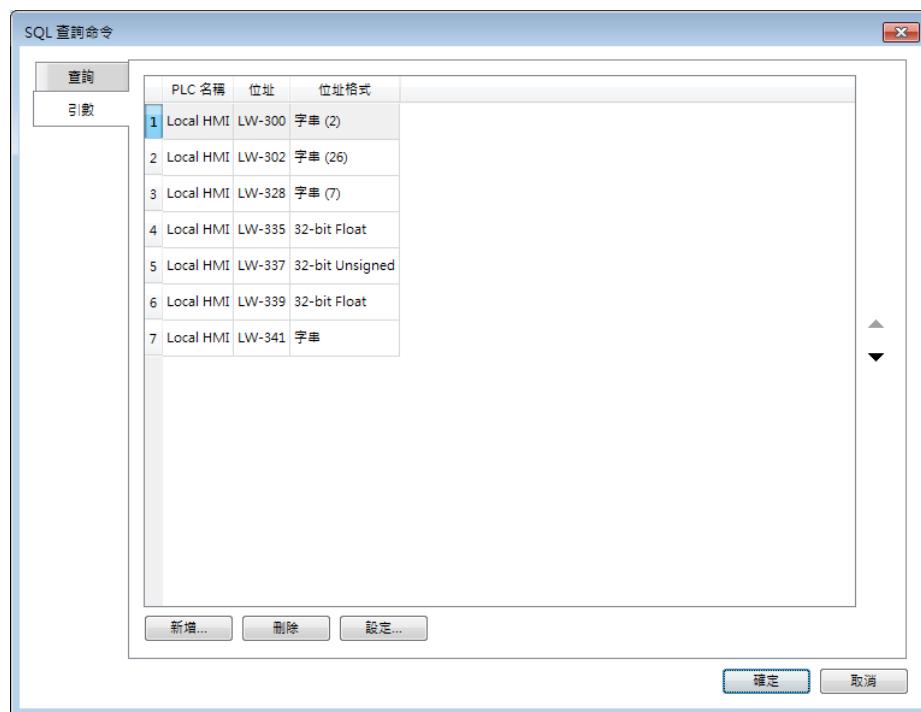
設定	描述
控制位址	輸入欲執行的命令 ID。
橫行選擇	指向特定橫向欄位。
狀態	顯示執行的狀態。參考 CH13.44.2.4。
錯誤碼	顯示執行的結果。參考 CH13.44.2.5。
錯誤訊息	顯示資料庫伺服器回傳的錯誤訊息，預設長度為 64 字組。亦可勾選 [字定義錯誤訊息長度]，最大可達 200 字組。
結果數	顯示實際輸出的筆數。
新增/刪除/設定/複製	新增/刪除/設定/複製一筆 SQL 命令。

## 查詢設定頁



設定	描述
命令 ID	控制此 SQL 語法的命令數值。
描述	此 SQL 語法的名稱。
SQL 啟動	欲執行的 SQL 語法。搜尋引數的方式為 \${引/數編號}。
捨棄結果	若勾選，執行完的 SQL 指令不會反應在 [SQL 查詢檢視] 物件上。例如：INSERT INTO, UPDATE, DELETE 等語句是直接操作資料庫，不需再傳回資料。此時，便可勾選 [捨棄結果] 使 [SQL 查詢檢視] 物件不進行更新。

## 引數設定頁



當 [查詢] 設定頁有使用到引數時，系統會根據 \${引數編號} 的設置參照到此設定頁的位址。

## 輸出設定頁

當使用進階模式進行資料庫查詢時，查詢得到的結果會依序儲存到 [輸出] 設定頁中指定的位址。若勾選 [多筆結果]，系統將會將查詢結果批量輸出到一系列連續的位址。

未啟用 [多筆結果]	啟用 [多筆結果]

設定	描述
批量位址	資料庫查詢結果的起始輸出位址。
最大結果數	可以透過 [最大結果數] 限制資料庫查詢輸出的數量，上限為 1000 筆。在設定畫面需要定義每筆查詢結果中所包含的欄位，並指定其位址格式及字組長度。(最大結果數) x (輸出數量)不能超過 10000 筆，也就

是最多輸出至 10000 個位址。

若設定一次輸出 100 筆，但 Query 回傳的結果有 300 筆，則：

當 [欄位選擇] 設定為 20，則輸出第 20 ~ 119 筆結果到批量位址。

當 [欄位選擇] 設定為 260，則輸出第 260 ~ 299 筆結果到批量位址。

<b>顯示第 N 筆結果的位址</b>	將第 N 筆資料使用的位址顯示在下方的表格。若一筆輸出結果的資料長度為 OFFSET，第一筆資料設定輸出至 LW-200，則第 N 筆位址的計算方式為 LW-200 + OFFSET * N。
---------------------	--------------------------------------------------------------------------------------------------



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 13.44.2.4. 狀態

狀態數值	意義
0	正常
1	回傳資料超過 1000 筆，可搭配 SQL 查詢語法 LIMIT 進行換頁

#### 13.44.2.5. 錯誤碼

錯誤碼數值	意義
0	無錯誤
1	未知的錯誤
2	不合法的指令
3	資料庫伺服器連線未就緒
4	無法讀取引數
5	無法寫入輸出
6	引數數量錯誤
7	MySQL 錯誤，請參考錯誤訊息內容。
8	不支援的資料型態
9	欄位數量超過限制
10	橫行選擇超過範圍
11	無法讀取本機資料庫的路徑
12	本機資料庫名稱不存在
13	內部錯誤

#### 13.44.2.6. 資料型態轉換

由 MySQL 資料庫讀取資料後，將按照以下表格進行資料型別轉換，若無法轉換則會跳錯誤碼 5，例如將 MySQL 的 INT 轉換成 16-bit Unsigned 時，若數值內容大於 16-bit Unsigned 可表示的範圍，也同樣會跳錯誤碼 5。

MySQL 資料格式	EasyBuilder Pro 資料型態
TINYINT	16/32-bit BCD

SMALLINT MEDIUMINT INT BIGINT BIT	16/32-bit HEX 16/32-bit Binary 16/32-bit Signed 16/32-bit Unsigned
FLOAT DOUBLE DECIMAL	32-bit Float
DATETIME CHAR, BINARY VARCHAR, VARBINARY TINYBLOB, TINYTEXT BLOB, TEXT MEDIUMBLOB, MEDIUMTEXT LONGBLOB, LONGTEXT	String

### 13.44.3. SQL 檢視查詢

#### 13.44.3.1. 概要

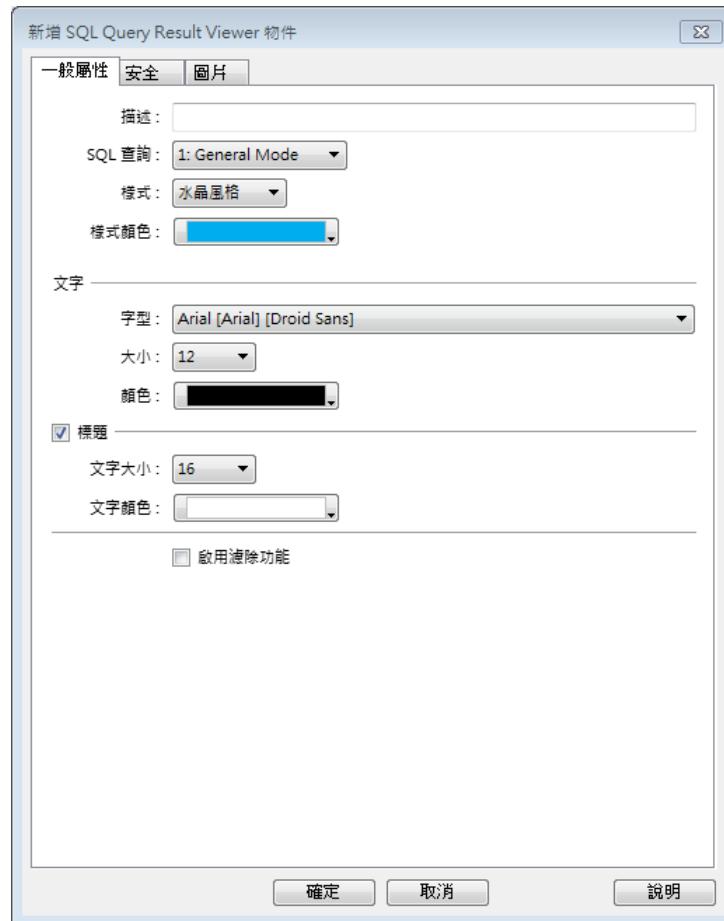
[SQL 檢視查詢] 物件主要是與 [SQL 查詢] 功能搭配使用。當設定好 [SQL 查詢] 功能的 SQL 讀取設定時，可使用 [SQL 檢視查詢] 物件顯示 SQL 資料。

#### 13.44.3.2. 設定



按下工作列的 [資料/歷史] » [SQL 檢視查詢] 按鈕後即會出現 [SQL 檢視查詢] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [SQL 檢視查詢] 項目。

## 一般屬性設定



設定	描述
描述	使用者可為此物件描述相關訊息。
SQL 查詢	選擇欲顯示的 [SQL 查詢] 編號。
樣式	選擇物件的樣式及色彩。
文字	設定此 [SQL 查詢檢視] 物件顯示時的文字參數。
標題	設定此 [SQL 查詢檢視] 物件的標題。
表格	當樣式使用為 [預設] 時，才有此設定欄位。可設定表格的顯示參數。
啟用過濾功能	啟用後，即可在 [SQL 查詢檢視] 物件上輸入字串搜尋指定文字。

## 13.45. 動態刻度

### 13.45.1. 概要

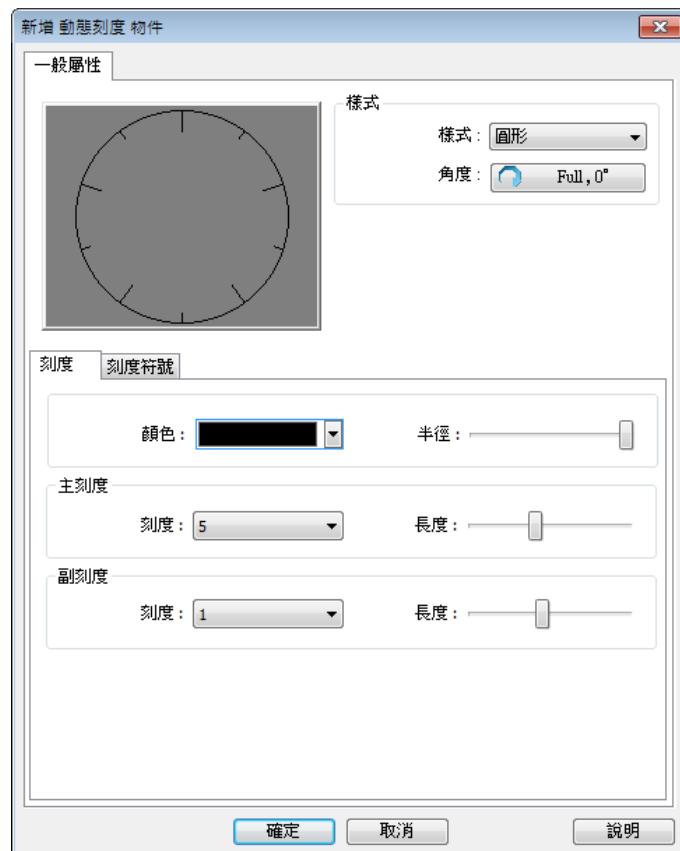
【動態刻度】物件可呈現不同樣式的刻度，並可以調整刻度內容，替其他物件如【趨勢圖】、【棒圖】等提供刻度。

### 13.45.2. 設定

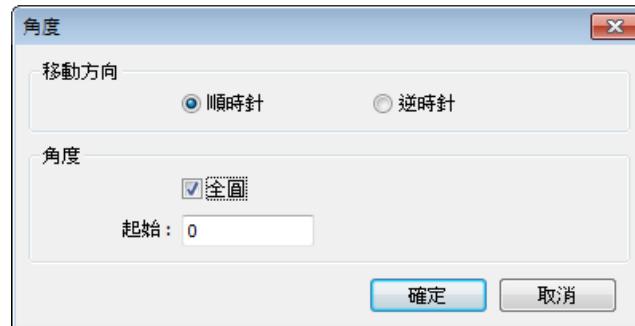


按下工作列上的【動態刻度】按鈕後即會開啟【動態刻度】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【動態刻度】物件。

#### 一般屬性設定



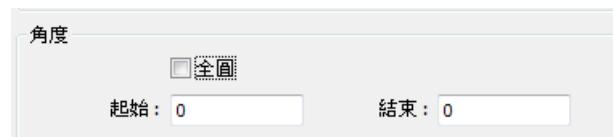
設定	描述
樣式	可選擇【圓形】、【水平】、【垂直】三種樣式。其中【圓形】可以設定【移動方向】與【角度】。



[移動方向] 可以選擇 [順時針]、[逆時針]。

[角度] 可以選擇 [全圓]，並設定 [起始角度]。

若不選擇 [全圓]，則須設定 [起始角度] 與 [結束角度]。



## 刻度

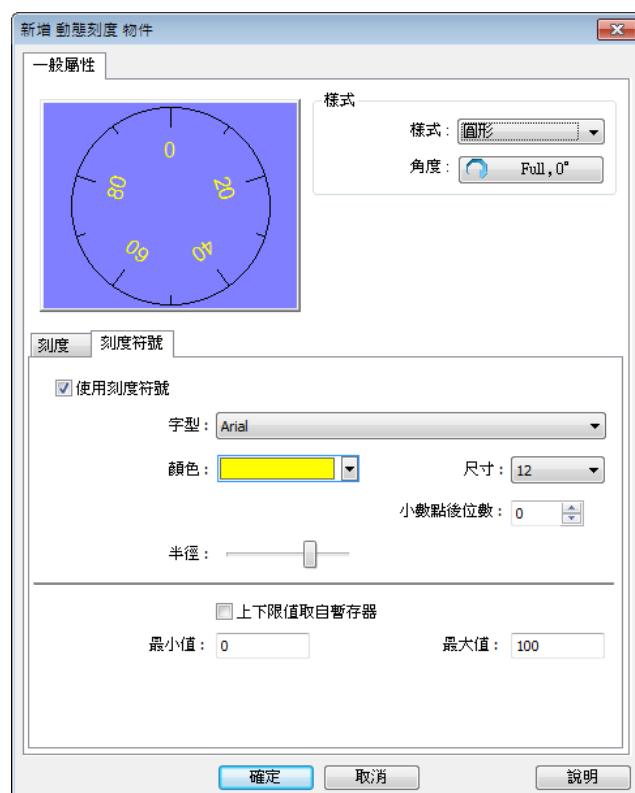
可設定 [動態刻度] 物件的 [顏色]、[主刻度] 與 [副刻度] 的刻度數目。

若 [動態刻度] 的樣式為 [圓形]，則可設定 [圓形] 的半徑與 [主、副刻度] 的長度。

## 刻度符號

可在 [主刻度] 上顯示 [刻度符號]，分為 [圓形] 刻度與非圓形刻度 ([垂直]、[水平]) 兩類，請見以下說明。

### 圓形

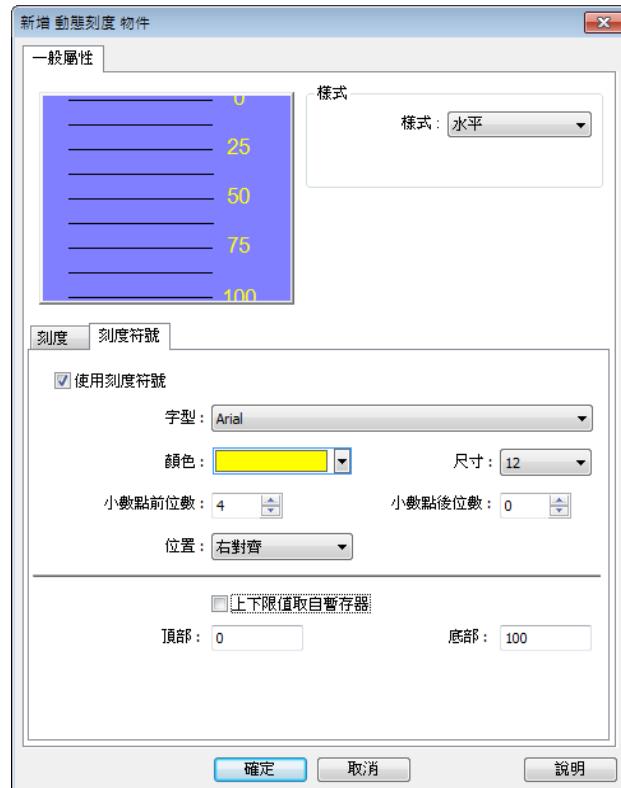


可設定刻度數字標示的字型、字型顏色、字型尺寸，與小數點後位

數。

可由動態刻度物件的中心點作為原點，設定刻度標記顯示的半徑範圍。

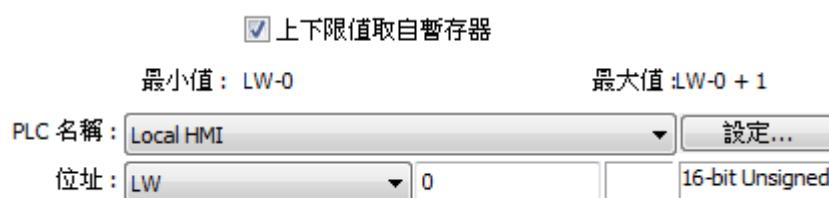
### 垂直、水平



可設定刻度數字標示的字型、字型顏色、字型尺寸，與小數點前/後位數。

可設定刻度標記顯示的位置。

刻度符號可以設定 [最大值]、[最小值]。若勾選 [上下限值取自暫存器]，則可以使用暫存器設定上下限。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.46. 動態繪圖

### 13.46.1. 概要

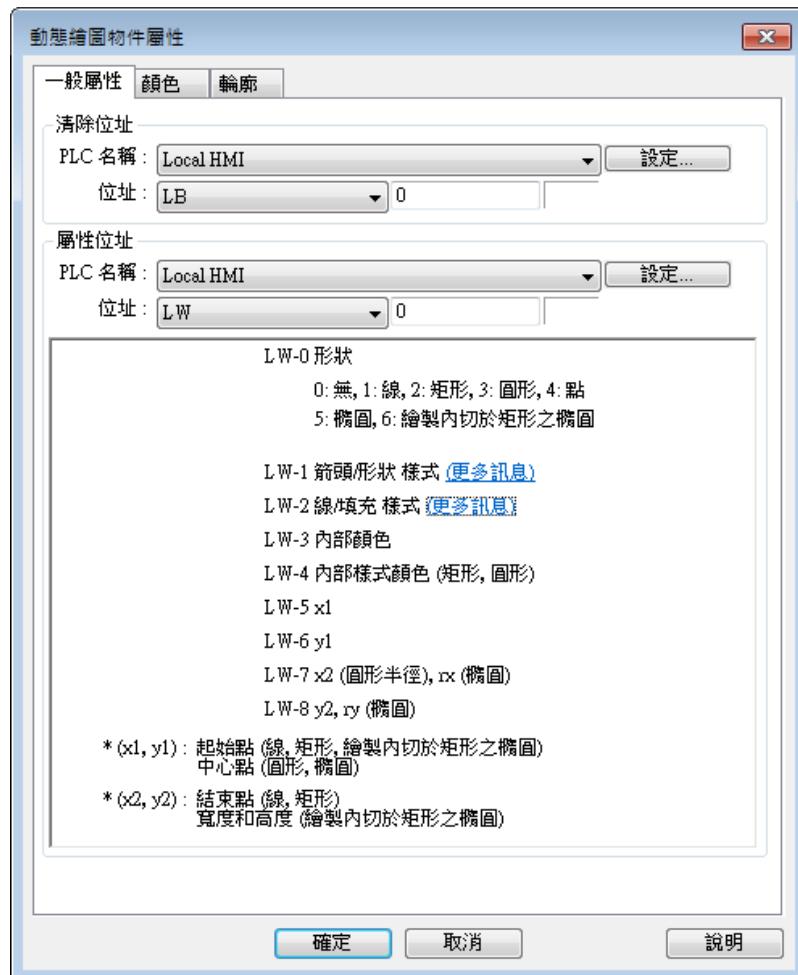
**[動態繪圖]** 物件可在人機介面運作時，在一定區域內顯示線段、矩形、圓形以及點等圖案。配合地址設定可畫出不同的樣式，另外還可以自定顏色。

### 13.46.2. 設定



按下工具列上的 **[動態繪圖]** 按鈕後，即會出現 **[動態繪圖]** 物件屬性對話窗，正確設定各項屬性後按下確定鍵，即可新增一個 **[動態繪圖]** 物件。

#### 一般屬性設定



設定	描述
清除位址	設定清除位址來清除動態繪圖。
屬性位址	設定相關的屬性位址來控制動態繪圖，以及顯示執行的結果。

各個屬性位址對應的樣式如下表所示。

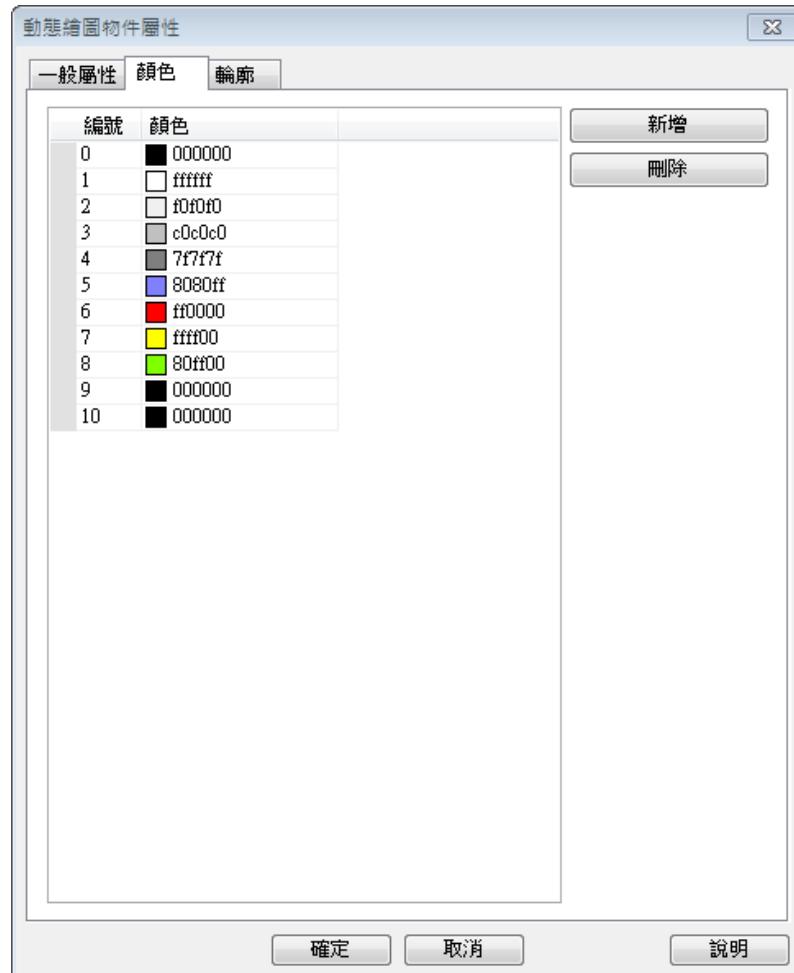
屬性位址	屬性位址+0	屬性位址+1	屬性位址+2	屬性位址+3	屬性位址+4
預設	0	個位數	十位數		自定義
線	1	0:無箭頭	0:小型	0:實線	線顏色
		1:單邊空心箭頭	1:大型	1:虛線	
		2:雙邊空心箭頭		2:點線	
		3:單邊實心箭頭		3:虛點線	
		4:雙邊實心箭頭		4:虛點點線	
				5 或以上:實線且寬度為 2 或以上	
矩形	2	0:空心		對應至線的樣式	矩形顏色
		1:實心		對應至填充的樣式	
圓形	3	0:空心		對應至線的樣式	圓形顏色
		1:實心		對應至填充的樣式	
點	4				點顏色
橢圓	5	0:空心		對應至線的樣式	橢圓顏色
		1:實心		對應至填充的樣式	
繪製內切於矩形的橢圓	6	0:空心		對應至線的樣式	橢圓顏色
		1:實心		對應至填充的樣式	
弧	7			對應至線的樣式	弧顏色
扇形	8	0:空心		對應至線的樣式	扇形顏色
		1:實心		對應至填充的樣式	
移動起始點	21				

屬性位址	屬性位址+0	屬性位址+5	屬性位址+6	屬性位址+7	屬性位址+8	屬性位址+9
預設	0					
線	1	起始點 X	起始點 Y	終點 X	終點 Y	
矩形	2	左上角 X	左上角 Y	右下角 X	右下角 Y	
圓形	3	圓中心點 X	圓中心點 Y	圓半徑		
點	4	點 X	點 Y			
橢圓	5	橢圓中心點 X	橢圓中心點 Y	橢圓 X 軸半徑	橢圓 Y 軸半徑	
繪製內切於矩形的橢圓	6	左上角 X	左上角 Y	寬度	高度	
弧	7	弧中心點 X	弧中心點 Y	弧半徑	起始角度	結束角度
扇形	8	扇形中心點 X	扇形中心點 Y	扇形半徑	起始角度	結束角度
移動起始點	21	新起始點 X	新起始點 Y			

【屬性位址+2】於不同樣式下的數值如下表所示。



## 色彩設定

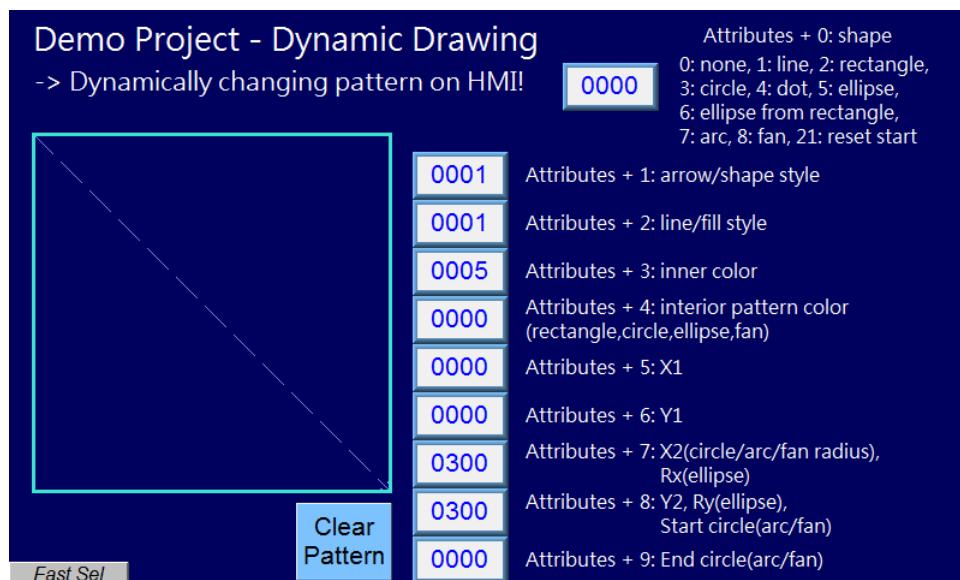


設定	描述
新增	新增動態繪圖所需要使用的顏色。
刪除	刪除所選擇的顏色。

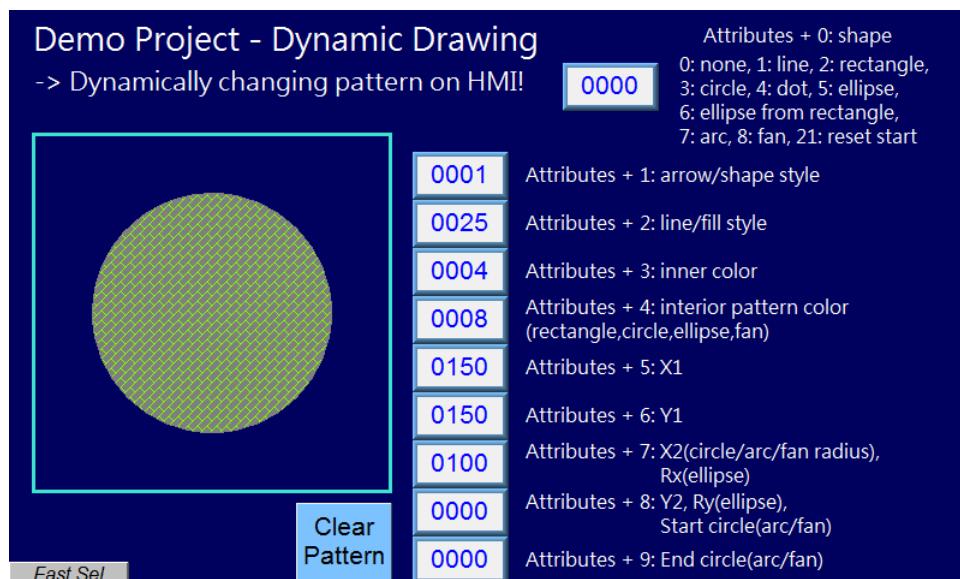
## 範例 1

以下範例展示如何藉由 [動態繪圖] 畫出線段以及圓形的範例。

1. 建立一個動態繪圖物件，清除繪圖位址設為 LB-0，屬性位址設為 LW-0。
2. 建立一個位元狀態切換開關物件，位址設為 LB-0，並選擇 [切換開關] 為開關類型，用於清除繪圖。
3. 建立十個數值輸入物件，位址設為 LW-0~LW-9，用於設定繪圖的屬性樣式。
4. 執行模擬或下載至人機驗證操作。當屬性位址數值設定如下圖所示，於 LW-0 位址輸入 1 就可得到線段繪圖。



5. 按下清除繪圖。把屬性位址數值設定變更如下圖所示，於 LW-0 位址輸入 3 就可得到圓形繪圖。



Note

- 需先定義 [屬性位址+1] ~ [屬性位址+8] 後，再執行 [屬性位址]。當執行完操作後，系統將自動重置 [屬性位址]。
- 若不清除繪圖，圖形會一直覆蓋疊加，在一個動態繪圖物件中，最多存在 1000 個圖形。
- 線的樣式數值最大為 19，等同於實線且寬度為 16，數值超過 19 則會以 19 來顯示。
- 顏色編號請至顏色頁籤設定。
- 弧跟扇形的起始角度與結束角度設定為 0 至 360。
- 原始起始點為左上角(0,0)，下移動起始點的命令會讓(X1,Y1)變成新的起始點，並且起始點座標會持續累加，直到清除繪圖才會將起始點重新設定成(0,0)。
- cMT/cMT X 機型若有需要使用 cMT Viewer 連線監控 [動態繪圖] 物件時，[屬性位址] 建議改用 PLW 位址控制，繪製時需對各 cMT Viewer 的 PLW 地址下指令，以避免 cMT Viewer 與 HMI 之間有繪圖結果不同步的情形。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.47. PDF 檢視器

### 13.47.1. 概要

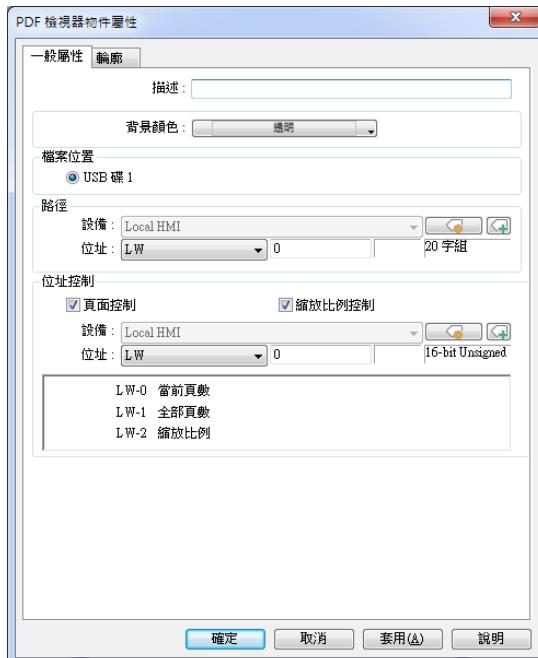
PDF 檢視器物件能讓使用者在 HMI 上瀏覽 PDF 文件。

### 13.47.2. 設定



請直接點擊 [PDF 檢視器] 圖示建立此物件，或點選工具列上的 [物件] » [媒體] » [PDF 檢視器] 新增此物件。

#### 一般屬性設定



設定	描述
檔案位置	選擇存放 PDF 檔案的位置。
路徑	PDF 檔案在外部儲存裝置中的路徑。
頁面控制	<b>當前頁數</b> 顯示當前顯示的頁碼。也可在此位址輸入頁碼，切換當前頁面。 <b>全部頁數</b> 顯示該份 PDF 檔案的全部頁數。
縮放比例控制	顯示當前縮放比例。也可調整縮放比例。縮放比例的單位為百分比，100 代表 100% (1:1) 顯示。

 Note

- PDF 檢視器物件無法使用模擬或是 cMT Viewer 開啟。
- 設有密碼保護或是權限的 PDF 檔案無法使用 PDF 檢視器物件開啟。
- 如果一次開啟多個 PDF 檢視器物件，可能提高 CPU 負載。
- 在多頁瀏覽模式下使用頁面控制位址更換頁面，將退出多頁瀏覽，而以單頁顯示所選的頁面。
- 具有電容式螢幕的人機，支援兩指放大縮小功能。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 13.48. 表格

### 13.48.1. 概要

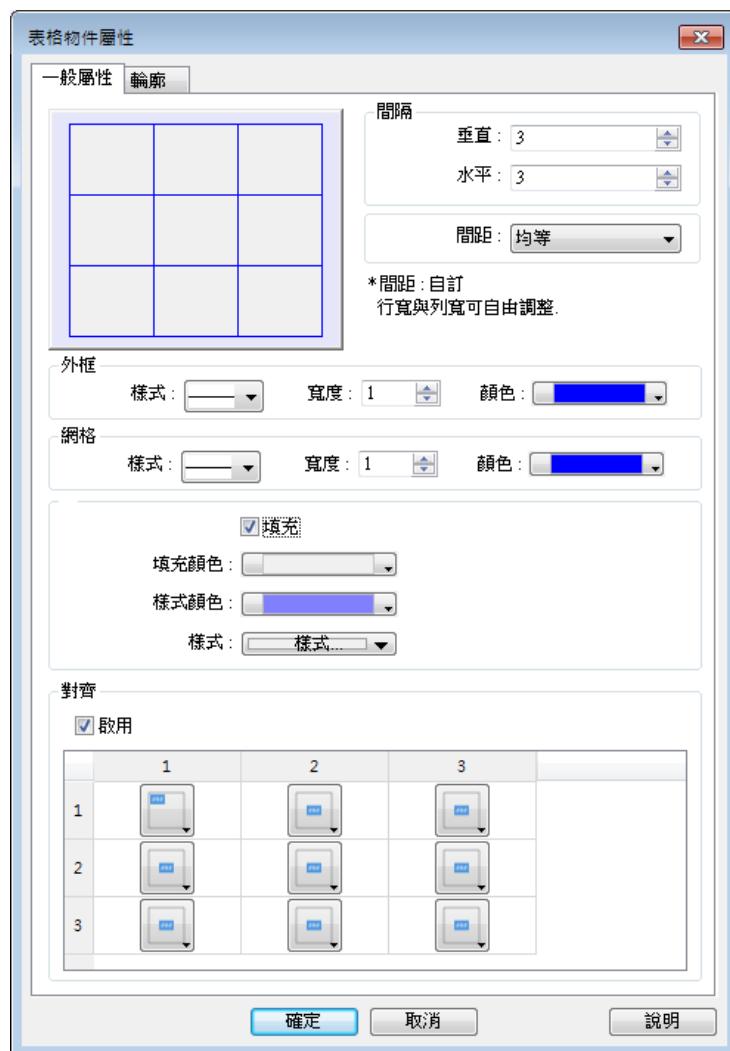
[表格] 物件可於人機畫面上繪製表格。使用者可設定表格的外框、網格及填充的樣式。

### 13.48.2. 設定



請直接點擊 [表格] 圖示建立此物件，或點選工具列上的 [物件] » [表格] 新增此物件。

#### 一般屬性設定



設定	描述
預覽視窗	顯示表格設定的樣式。
垂直間隔	設定垂直間隔的數量，範圍 1 ~ 255。
水平間隔	設定水平間隔的數量，範圍 1 ~ 255。
間距	可選擇 [均等] 或 [自訂]，若選擇自訂，則可於編輯視窗手動調整垂直線與水平線之間距。
外框	設定表格外框的樣式、寬度及顏色。當線樣式選擇 [實線] 時，才可設定線寬，且其範圍為 0 ~ 8。當線寬設為 0 時，將不顯示線條。
網格	設定表格網格的樣式、寬度及顏色。當線樣式選擇 [實線] 時，才可設定線寬，且其範圍為 0 ~ 8。當線寬設為 0 時，將不顯示線條。
填充	設定表格填充的樣式及顏色。
對齊	將其他物件拖曳至格子中，會依照對齊設定自動調整物件位置。表格內每一格可以各自設定對齊方式。可設定水平、垂直對齊方式，預設為置中對齊。 當啟用對齊，垂直間隔/水平間隔數量只支援達 32。 僅支援單一物件的對齊，不支援群組物件的對齊。

## 13.49. VNC Viewer

### 13.49.1. 概要

[VNC Viewer] 物件可以在人機上運行 VNC 功能連接到有安裝 VNC server 的電腦或有支援 VNC server 功能的人機。使用者可以在人機上監看或操控被連線端的畫面。

### 13.49.2. 設定



請直接點擊 [VNC Viewer] 圖示建立此物件，或點選工具列上的 [物件] » [媒體] » [VNC Viewer] 新增此物件。

### 一般屬性設定

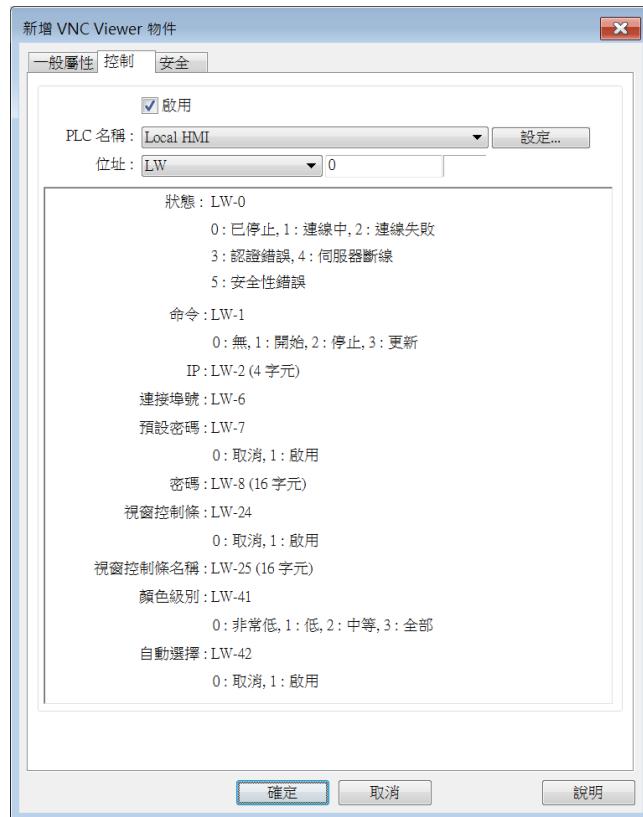


設定	描述
IP	連線對象的 IP 位址。
連接埠號	連線對象的連接埠號。
預設密碼	可直接輸入連線對象的 VNC 預設密碼，當切換到 VNC Viewer 所在的頁面時，VNC Viewer 就會直接與連線對象連線，不需要另外輸入密碼。
顏色級別	<p>可以選擇 [全部(所有可用顏色)]、[中等(256 色)]、[低(64 色)]、[非常低(8 色)]。</p> <p><b>自動選擇</b></p> <p>可讓 VNC Viewer 自動偵測並選擇 VNC 伺服器支援的顏色級別。</p> <p><b>啟用線上修改功能</b></p> <p>可使用 [控制] 功能對應的位址來選擇顏色級別與是否要啟用自動選擇，請見 [控制設定]。</p>
視窗控制條	<p><b>啟用</b></p> <p>勾選後在方框內可以輸入要在 VNC Viewer 的控制條上顯示的名稱，僅支援 ASCII 文字，無法設定字型。可搭配 [文字標籤庫] 使用。同時需要勾選 [啟用] 才能在畫面上自由拖曳與縮放 VNC Viewer 視窗。</p>
選項	<p><b>啟用系統鍵盤輸入</b></p> <p>啟用後，操作 VNC Viewer 物件，畫面右上角會出現鍵盤圖示，點擊圖示會呼叫出系統鍵盤。</p>

 **Note**

- eMT、iE、XE、mTV: 需使用 OS 20160418 或更新版本的 OS。
- cMT / cMT X 系列: 僅支援人機本地畫面上執行，需使用 OS 20180928 或更新版本的 OS。
- 不支援連線模擬功能。
- VNC Viewer 僅顯示 VNC 伺服器實際的解析度大小，且不支援縮放功能。
- 勾選 [預設密碼] 時，人機的 Virtual Keyboard 需要手動呼叫。若不勾選 [預設密碼]，Virtual Keyboard 才會自動彈出。

## 控制設定



設定	描述
<b>控制</b>	指定相關的控制位址來設定 [VNC Viewer] 的各項參數，以及顯示狀態。
<b>控制位址 : 狀態</b>	<p>0 : 已停止      1 : 連線中      2 : 連線失敗      3 : 認證錯誤      4 : 伺服器斷線      5 : 安全性錯誤</p>
<b>控制位址+1 : 命令</b>	<p>0 : 無      1 : 開始      2 : 停止</p>
<b>控制位址+2~控制位址+5 : IP</b>	
<b>控制位址+6 : 連接埠號</b>	
<b>控制位址+7 : 預設密碼</b>	
	0 : 取消

1：啟用  
**控制位址+8：密碼 (16 字組)**  
**控制位址+24：視窗控制條**  
0：取消  
1：啟用  
**控制位址+25：視窗控制條名稱 (16 字組)**  
**控制位址+41：顏色級別**  
0：非常低  
1：低  
2：中等  
3：全部  
**控制位址+42：自動選擇**  
0：取消  
1：啟用

### Note

- 控制位址+6 的連接埠號輸入範圍若為 0~99 時，實際埠號為輸入值再加上 5900。舉例來說，若在控制位址+6 輸入 1，則實際的連接埠號為 5901。但若在控制位址+6 輸入 100，實際的連接埠號為 100。
- 當開啟 VNC Viewer 時，若要使用 HMI 內建的 Virtual Keyboard 輸入，需要先按一下 Virtual Keyboard 上的 [Focus] 按鍵，再點一下 VNC Viewer 畫面，才能切換輸入目標。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。



請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.50. 連絡人編輯器

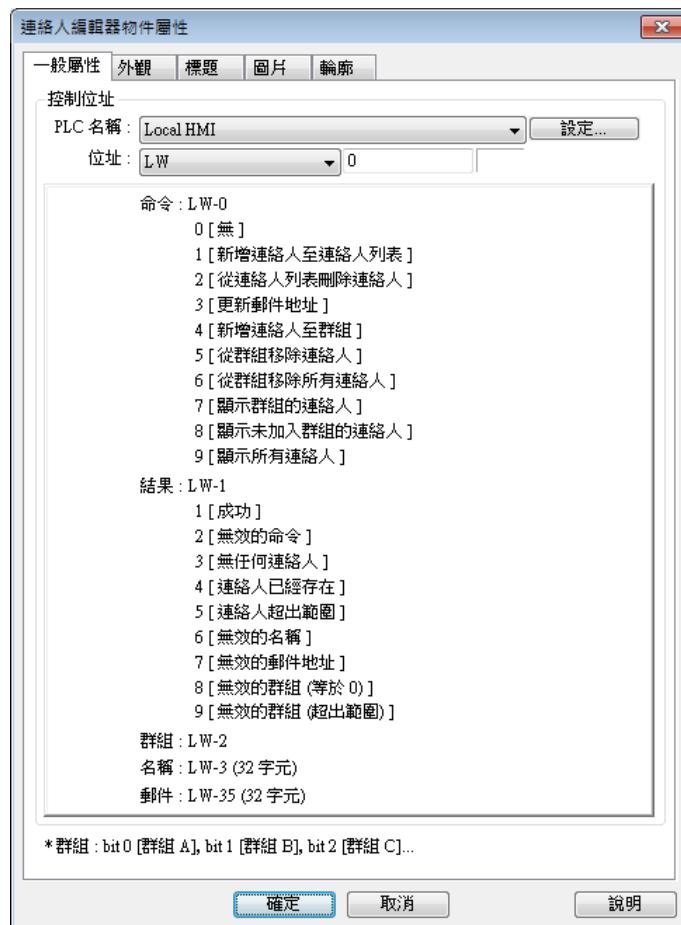
[連絡人編輯器] 物件支援使用者在 HMI 上即時增減郵件寄送對象的連絡資訊。

### 13.50.1. 設定



請直接點擊 [連絡人編輯器] 圖示建立此物件，或點選工具列上的 [物件] » [連絡人編輯器] 新增此物件。

#### 一般屬性設定



設定	描述
控制位址	指定的暫存器及其連續的位址會做為 [連絡人編輯器] 物件的參數使用。
命令位址 (LW-n) :	

數值	內容
0	無
1	新增連絡人至連絡人列表
2	從連絡人列表刪除連絡人
3	更新郵件地址
4	新增連絡人至群組
5	從群組移除連絡人
6	從群組移除所有連絡人
7	顯示群組的連絡人
8	顯示未加入群組的連絡人
9	顯示所有連絡人

#### 結果位址 (LW-n+1) :

數值	內容
1	執行成功
2	無效的命令
3	找不到連絡人
4	連絡人已經存在
5	連絡人數量已達上限，無法再加入
6	無效的名稱
7	無效的郵件地址
8	無效的群組
9	無效的群組 (超出範圍)

#### 連絡人群組 (LW-n+2) : 以位元代表群組

位元	內容
0	群組 A
1	群組 B
2	群組 C
3~15	依此類推

使用者名稱 (LW-n+3) : 共 32 個字組

郵件地址 (LW-n+35) : 共 32 個字組



- cMT / cMT X 系列無一般屬性設定頁。

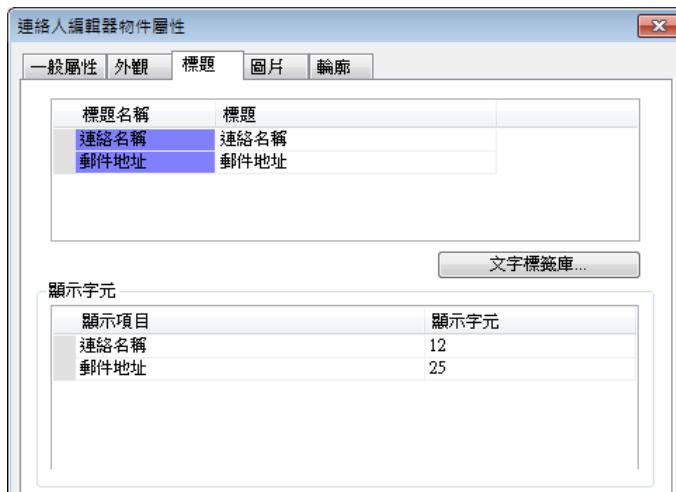
- 連絡人名稱不支援 Unicode 文字。
- 群組數量設定於 [系統參數設定] » [郵件] » [收信人]，群組數量無法線上更改。

## 外觀設定



設定	描述
字型 & 顏色	調整 [連絡人編輯器] 物件顯示的字型及物件顏色。

## 標題設定



設定	描述
標題	設定標題的顯示文字。
顯示字元	設定連絡人名稱及郵件地址的文字長度。範圍：1~60

## 13.51. 事件甘特圖

### 13.51.1. 概要

事件甘特圖是一種常見的時程管理工具，被廣泛應用在專案管理。當事件甘特圖應用在 HMI 的事件警報上時，可以幫助使用者更清楚地了解警報的發生區間、持續時間長短等。使用事件甘特圖前，需先設定事件登錄物件。

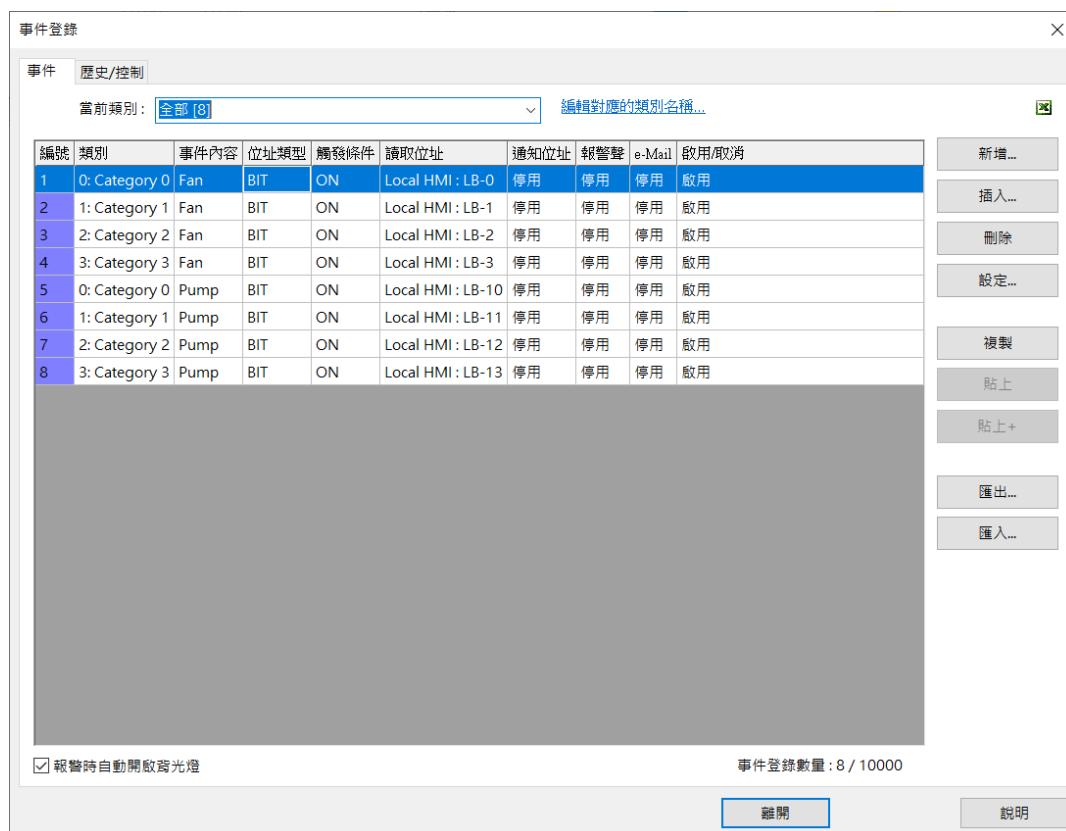
本功能僅支援於 cMT / cMT X 系列人機 (cMT-Gateway 除外)。

### 13.51.2. 設定

#### 13.51.2.1. 事件登錄

請點選工具列上的 [資料/歷史] » [事件登錄] 建立數個事件登錄物件。

事件甘特圖在繪製圖形時，部分的參數設定在 [事件登錄] 中。以下介紹會影響事件甘特圖物件繪製圖形時，需使用的參數。



設定	描述
編輯對應的類別名稱	事件甘特圖繪製圖形時，會顯示各類別的名稱。

### 13.51.2.2. 一般屬性設定



#### 設定

#### 描述

##### 類別

事件甘特圖繪製圖形時，是直接繪製類別中的所有事件，因此個別事件的類別需正確設定。

### 13.51.2.3. 訊息設定



#### 設定

#### 描述

##### 文字內容

項目名稱。

##### 顏色

項目色彩。

### 13.51.3. 事件甘特圖

#### 13.51.3.1. 設定



請點選工具列上的 [資料/歷史] » [事件甘特圖] 建立事件甘特圖物件。

#### 一般屬性設定

**新增 事件甘特圖 物件**

一般屬性	外觀
描述 : <input type="text"/>	
顯示的類別範圍	
<input checked="" type="radio"/> 預定義 <input type="radio"/> 動態類別範圍 <input type="text"/> 0 ~ <input type="text"/> 255 * 參閱事件登錄物件。	
選項按鈕顯示	
<input checked="" type="radio"/> 預定義 <input type="radio"/> 動態 <input checked="" type="checkbox"/> 啟用	

設定	描述
----	----

**顯示的類別範圍**  
預定義  
預先設定欲顯示在事件甘特圖上的事件類別。

**動態類別範圍**  
物件將根據暫存器的數值動態篩選顯示的事件類別。

顯示的類別範圍
<input type="radio"/> 預定義 <input checked="" type="radio"/> 動態類別範圍
設備 : Local HMI
位址 : LW
起始範圍 : LW-0
結束範圍 : LW-1

#### 選項按鈕顯示



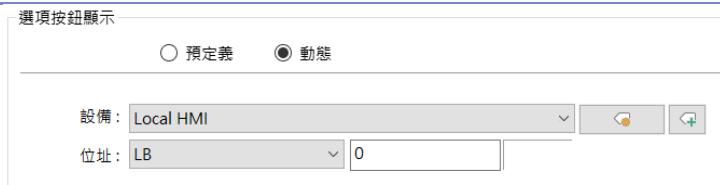
設定是否顯示物件上的選項按鈕。

##### 預定義

預先設定是否顯示選項按鈕。

##### 動態

設定一個位元位址動態決定是否顯示選項按鈕。



## 外觀設定



### 設定

### 描述

#### 顯示的時間間隔

隔

顯示的時間範圍。此範圍在 HMI 亦可透過物件右上方的  
調整。



按鈕

取消	選項	完成
起始日期與時間	<input type="button" value="..."/>	
結束日期與時間	<input type="button" value="..."/>	
事件過濾		
關鍵字		
大小寫須相符	<input checked="" type="checkbox"/>	
顯示時間跨度 (小時)	1 ▼	

樣式	事件甘特圖物件的樣式。系統預設提供三種樣式，客戶亦可選用客製樣式。
文字尺寸	事件甘特圖上文字的尺寸，範圍為 50%~200%。
標題格式	日期 + 時間 : 08/16/2018 13:55 ~ 08/16/2018 14:55 只顯示日期 : 09/03/2018 ~ 09/04/2018 只顯示時間 : 13:57 ~ 14:57
檢視線	在事件甘特圖的條狀圖上點擊任一處時，會出現的一條檢視線。檢視線最頂端並會顯示此位置的時間。
分割數量	X 軸的網格數量。事件甘特圖會將 X 軸分成 [分割數量] 等份。
時間標籤	時間刻度的表示方式。

## HMI 上設定

在 HMI 或是於 cMT Viewer 中可透過物件右上方的  按鈕調整物件顯示。

取消	選項	完成
起始日期與時間	<input type="button" value="..."/>	
結束日期與時間	<input type="button" value="..."/>	
事件過濾		
關鍵字		
大小寫須相符	<input checked="" type="checkbox"/>	
顯示時間跨度 (小時)	1 ▼	

設定	描述
起始日期與時間	設定顯示資料的開始時間。 預設為未開啟，圖表資料自現存資料的起始時間開始顯示。
結束日期與時間	設定顯示資料的結束時間。 預設為未開啟，圖表資料顯示至當前時間為止。
事件過濾	關鍵字：可輸入關鍵字來過濾特定事件。 大小寫須相符：使用英文搜尋時是否要將大小寫視為不同字元。
顯示時間跨度 (小時)	可線上更改圖上顯示的時間範圍(1~96)。All 表示現有資料範圍即為顯示的時間範圍。

 **Note**

- HMI 斷電期間，無法明確得知各個警報的狀態，因此斷電重開後，並不會繪製斷電期間的圖形。

## 13.52. 動作觸發

### 13.52.1. 概要

**[動作觸發]** 物件可以根據指定的觸發條件用來分批執行一系列的動作。設定動作時，可以先將動作進行分組，同一個群組中的動作會一起執行。當前一個群組中的所有動作都觸發後，下一個群組的動作才會開始進行。

使用 **[動作觸發 (視窗)]** 時，觸發方式為 **[視窗開啟/關閉]**、**[背光燈開啟/關閉]**、**[數值變化]**。

使用 **[動作觸發 (背景)]** 時，觸發方式為 HMI 偵測 **[閒置時間]**、**[數值變化]**、**[控制權]**、**[使用者安全存取]**、**[條件物件]**。

僅 cMT / cMT X 系列支援此物件。

### 13.52.2. 設定



按下工作列的 **[物件] » [動作相關] » [動作觸發 (視窗)]** 或 **[動作觸發 (背景)]** 按鈕後即會開啟 **[動作觸發]** 物件屬性對話窗，點選 **[新增]** 新增一個動作觸發項目。

#### 一般屬性設定

**動作觸發 (視窗)**



**動作觸發 (背景)**



設定	描述
<b>觸發模式 (視窗)</b>	<p><b>視窗開啟</b> 當視窗開啟時，觸發該物件設定的項目。使用此模式時，不支援 [安全] 分頁功能。</p> <p><b>視窗關閉</b> 當視窗關閉時，觸發該物件設定的項目。</p>
<b>背光燈開啟</b>	背光燈開啟時，觸發該物件設定的項目。
<b>背光燈關閉</b>	背光燈關閉時，觸發該物件設定的項目。
<b>數值變化</b>	當指定位址的數據達成觸發條件後，觸發該物件設定的項目。
<b>[閒置時間] 模式 (背景)</b>	當 HMI 在無人操作的情況下，超過 [閒置時間] 設定的時間後，即會觸發該項目。範圍: 1 ~ 43200 秒。
<b>[數值變化] 模式 (背景)</b>	<p>當指定位址的數據達成觸發條件後，即會觸發該項目。</p> <p><b>當動作結束時檢查觸發條件</b> 執行完所有動作後，物件會再度檢查觸發條件是否成立，若成立將再度執行。</p> <p><b>動態狀態數值</b> 使用 [字組] 類型時支援動態狀態數值。當觸發條件使用動態狀態數值且無啟用 [讀取/狀態使用不同的位址] 時，觸發條件會使用 [數值位址] 的下一個連續暫存器。</p>
<b>[控制權] 模式 (背景)</b>	當取得或釋放控制權時，即會觸發該項目。
<b>[使用者安全存取] 模式 (背景)</b>	當安全等級設定為 [任何] 時，只要任一權限符合 [觸發當存取等級] 時都會觸發該物件設定的項目。若安全等級設定為特定權限，只有當特定權限符合 [觸發當存取等級] 時，才會觸發該物件設定的項目。
<b>[條件物件] 模式 (背景)</b>	<p>搭配 [觸發條件] 物件使用。當條件符合 [觸發] 設定時，會執行 [動作觸發] 物件。</p> <p><b>當動作結束時檢查觸發條件</b> 執行完所有動作後，物件會再度檢查觸發條件是否成立，若成立將再度執行。</p>
<b>動作群組</b>	將動作進行分組。同一個群組中的動作會同時開始執行，當該群組中的動作全部觸發完畢後，才會執行下一個群組中的動作。
<b>[延遲] 動作</b>	延遲 n 毫秒後才往下執行指令。

**[位元設定] 動作**

將暫存器的狀態設定為 ON 或 OFF。

**設為 ON**

所指定暫存器的狀態將被設定為 ON。

**設為 OF**

所指定暫存器的狀態將被設定為 OFF。

**切換開關**

所指定暫存器的狀態將被設定為反向。

**[多狀態設定] 動作**

可改變指定暫存器的數據。

**寫入常數**

將常數寫入指定暫存器。

**遞加 (JOG+)**

加值功能。所指定暫存器內的數據將加上 [遞加值] 中設定的增量值，但增值的結果將不超過 [上限值] 中的設定值。

**遞減 (JOG-)**

減值功能。所指定暫存器內的數據將減去 [遞減值] 中設定的減量值，但減值的結果不會低於 [下限值] 中的設定值。

**動態限制 (JOG+,JOG-)**

JOG+上限 / JOG-下限由指定暫存器設定。

**寫常數字串**

將特定字串寫入指定暫存器。

**物件控制命令 (限 cMT / cMT X 機型)**

提供各功能控制位址可使用的命令供選擇 (同時顯示命令對應的數值)。

**系統標籤命令 (限 cMT / cMT X 機型)**

當寫入位址為特定系統標籤時，例如多國語言設定 LW-9134，即可選擇欲設定的命令。

**[切換基本視窗] 動作**

可進行切換基本視窗、公共視窗或是返回上一個視窗。切換基本視窗只能為最後一個動作。

**切換基本視窗**

換頁至指定的視窗。

**切換公共視窗**

切換公用視窗。

**返回上一個視窗**

返回前一頁基本視窗。例如當由”視窗 10”切換到”視窗 20”時，使用此功能可以再返回”視窗 10”。此功能只對基本視窗有效。

**動畫設定**

能夠選擇的動畫效果有：淡出、飛入、飄入、擦去、分割、

環狀、時鐘、縮放、旋轉、推入等，並可設定持續時間與顯示方向。

#### [觸發巨集指令] 動作

觸發指定巨集。若群組中有巨集指令，且巨集指令執行的時間較長時，只要巨集指令一旦被觸發，即會視為動作執行成功，而不會等到該巨集全部執行完成才進行觸發下一個群組的動作。

#### [彈出視窗] 動作

彈出指定的視窗。

#### [關閉當前開啟的視窗]

##### 動作

可關閉當前開啟的視窗。

#### [顯示計算機視窗] 動作

可直接呼叫出一個外觀固定的計算機物件，可以在人機上執行簡單計算並輸出至目標數值物件上。

#### [鍵盤輸入] 動作

用來作為鍵盤的輸入訊號，主要用在 [數值] 與 [字元] 物件需要使用鍵盤來輸入數字或文字的場合。

**Enter:** 與鍵盤的輸入 (Enter) 動作相同。

**Backspace:** 與鍵盤的後退刪除 (Backspace) 動作相同。

**Clear:** 清除暫存器中已輸入的資料。

**Esc:** 與使用 [關閉視窗] 功能相同，可用來關閉彈跳出的鍵盤視窗。

**Delete:** 與鍵盤的刪除 (Delete) 動作相同，可將游標右方的一個字元刪除。

**Left:** 與鍵盤的←動作相同，可將游標向左移動一個字元。

**Right:** 與鍵盤的→動作相同，可將游標向右移動一個字元。

**Inc:** 將數值+1。

**Dec:** 將數值-1。

**ASCII/UNICODE:** 設定鍵盤的輸入字元。

#### [畫面擷取] 動作

可擷取當前的畫面，並選擇將畫面儲存至 USB 碟或 SD 卡。

#### [確認所有事件(報警)] 動

##### 作

執行時可一次確認所有報警事件。

#### [匯入資料] 動作

用來匯入進階安全的使用者帳號或 e-mail 的連絡人。也可設定為使用 USB 金鑰登入。

**資料位置：**提供從 USB 碟、SD 卡讀取兩個選項。

**帳號匯入方式：**選擇 [覆蓋]，HMI 內將只保存此次匯入的帳號資料，若是選擇 [附加]，HMI 內帳號資料將保留，並加入此次匯入的新帳號資料。

**匯入使用者帳號後刪除檔案：**將 USB 內的使用者帳號匯入後即刪除來源資料，可確保資料不洩漏。

#### [等待觸發條件] 動作

使用位元/字元設定不同觸發條件。當滿足觸發條件時，才會

執行下一個【動作群組】的動作。若條件在超時等待時間內沒有達成，可選擇繼續進行下一個群組的動作或者不繼續執行動作，提前結束。

<b>[資料傳輸 (背景)] 動作</b>	將指定位址中的數據傳送到其他位址中。
<b>[檔案傳輸] 動作</b>	利用 FTP 傳輸檔案。HMI 將以被動模式連線 FTP 伺服器。

#### **[一般屬性] 設定頁**

##### **設定傳輸方向**

**[下載]** 將檔案從 FTP 伺服器下載至本機 HMI。

**[上傳]** 將檔案從本機 HMI 傳送至 FTP 伺服器。

##### **設定伺服器位址**

**[靜態]** 直接設定伺服器位址相關設定。

**[動態]** 指定一個動態位址，以在 HMI 執行時可透過特定位址動態調整伺服器設定。假設動態位址為 LW-n:

LW-n: 主機語法 (0:IP, 1:網域名稱)

LW-n+1: IP (4 字組)

LW-n+5: 連接埠號

LW-n+6: 認證 (0:停用, 1:啟用)

LW-n+7: 使用者名稱 (16 字組)

LW-n+23: 密碼 (16 字組)

LW-n+39: 網域名稱 (64 字組)

##### **成功傳輸後刪除來源檔案**

檔案傳輸後，會刪除來源檔案。當傳輸方向為下載(FTP -> HMI)時，需確保該帳號具有 FTP 伺服器寫入權限。

#### **[檔案設定] 設定頁**

可設定檔案位置、FTP 伺服器端檔案路徑、本機 HMI 端檔案路徑。當路徑為資料夾時，則將傳輸資料夾內的所有檔案，但不包括子資料夾內的檔案。若檔案名稱相同時，一律覆蓋檔案。

#### **[狀態] 設定頁**

設定顯示檔案傳輸執行結果及回應的位址。

關於 FTP 伺服器的回應列表可參考：

[https://en.wikipedia.org/wiki/List\\_of\\_FTP\\_server\\_return\\_codes](https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes)。

<b>[控制權] 動作</b>	取得或釋放控制權。

#### **控制權**

選擇控制權。

#### **行為模式**

設定取得或釋放控制權。

#### **最大等待時間**

當持續無法取得控制權，即會根據 [若未取得控制權] 執行設定的動作。

**[備份] 動作**

備份歷史資料或配方資料至外接儲存裝置，或是透過電子郵件或 FTP 伺服器傳輸。

**[安全移除 USB 碟/SD 卡] 動作**

可安全的移除外接儲存裝置，確保資料的完整性。

**Note**

- 工程檔案中最多可建立 1000 個動作觸發項目。
- 一個動作觸發項目最多可建立 20 個群組，每一個群組最多 20 個動作。
- 如欲修改動作群組名稱，可在該動作群組名稱按下右鍵，並按下 [重新命名]。
- 檔案傳輸功能的錯誤碼列表：

錯誤碼	說明
0	執行成功
1	下載時，HMI 路徑不存在
3	偵測不到 USB 隨身碟或 SD 卡
4	未輸入 HMI 或 FTP 路徑
5	上傳檔案不存在
8	操作被 FTP 服務器拒絕
9	USB 隨身碟或 SD 卡容量已滿
10	未知的錯誤

## 13.53. 月曆

### 13.53.1. 概要

【月曆】物件可以在 cMT / cMT X 系列的人機上顯示月曆資訊。

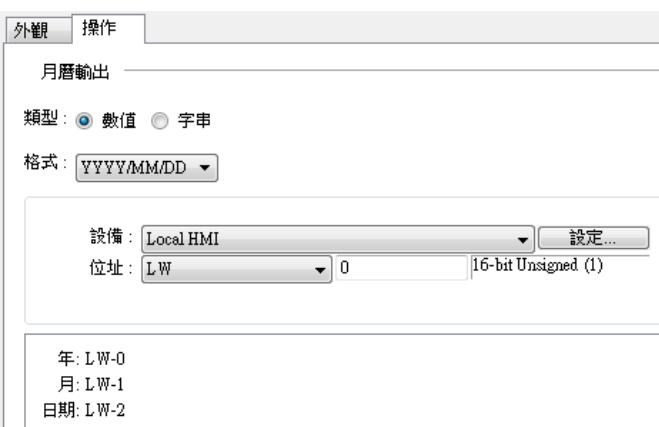
### 13.53.2. 設定



按下【物件】頁籤»【時間相關】»【月曆】按鈕後即會開啟【月曆】物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個【月曆】物件。

#### 一般屬性設定



設定	描述
外觀	導航欄：可設定背景與文字顏色。 星期：可設定背景與文字顏色。 日期：可設定背景、網格與文字顏色。 今天：可設定背景與文字顏色。
操作	可將日期資訊輸出到指定的位址，輸出格式可指定為數值或字串。 

 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 13.54. 觸控手勢

### 13.54.1. 概要

現在的智慧型手機、平板電腦可透過多點觸控的方式操作，例如同時五指滑動可關閉應用程式等，而 [觸控手勢] 物件可以讓 HMI 也有如同在智慧型裝置上操作的體驗。[觸控手勢] 物件支援多達 23 種的手勢。

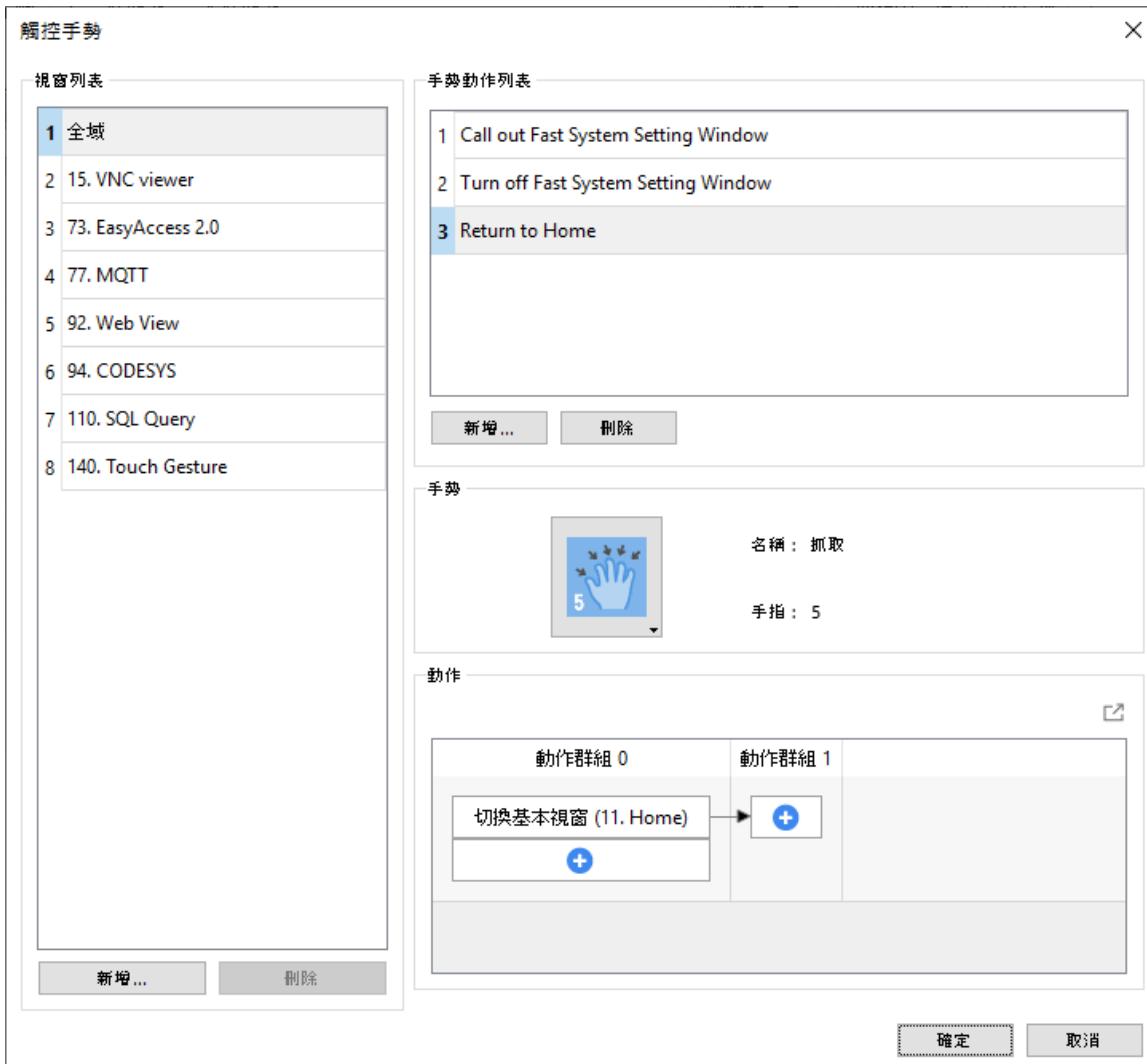
觸控規格為最高同時五指。

### 13.54.2. 設定



按下工作列的 [物件] » [動作相關] » [觸控手勢] 按鈕後即會開啟 [觸控手勢] 物件屬性對話窗，先選擇手勢執行的視窗後，點選 [新增] 新增一個觸控手勢項目。

## 一般屬性設定



設定	描述
視窗列表	設定觸控手勢僅在特定視窗運作或所有視窗都會運作。選擇 [全域] 時，不支援一指操作的觸控手勢。
	<b>新增/刪除</b> 新增/刪除一個欲使用觸控手勢的視窗。
手勢動作列表/手勢	顯示在該視窗使用的手勢。
	<b>新增/刪除</b> 新增/刪除一個手勢。
動作	使用該手勢時，可觸發的一連串動作。關於動作的說明，可參考手冊 CH13.36 或 CH13.52。

 Note

- 每一視窗中(全域視窗或個別視窗)，可使用最多 16 個手勢動作。
- 如果在全域視窗和個別視窗中使用到重複的手勢，僅會執行個別視窗中的動作。
- 如果在手勢操作途中有觸碰到物件，物件本身還是會被觸發。
- 個別視窗的觸控手勢動作僅在視窗是基本視窗時有效。若該視窗用於彈出視窗彈出時，或重疊視窗時，觸控手勢都無效，相關動作也不會被執行。
- 當已觸發之手勢指定的動作都結束後才會執行下一次的手勢偵測。

## 13.55. PLC 網頁瀏覽器

### 13.55.1. 概要

[PLC 網頁瀏覽器] 物件可以用來瀏覽 PLC 的網頁。此功能目前僅在 cMT X 進階型系列支援，支援列表請參考 [附錄 A 各系列 HMI 軟體功能差異](#)。

### 13.55.2. 設定



按下工作列的 [物件] » [媒體] » [PLC 網頁瀏覽器] 按鈕後即會開啟 [PLC 網頁瀏覽器] 物件屬性對話窗，正確設定各項屬性後按下確認鍵，即可新增一個 [PLC 網頁瀏覽器] 物件。

#### 一般屬性設定



設定	描述																																										
預設 URL	可指定欲開啟的網頁。																																										
顯示網址列	勾選此選項後可以在人機上直接更改網址、重整網頁、縮放網頁大小。																																										
	<table border="1"> <thead> <tr> <th>圖示</th><th>描述</th></tr> </thead> <tbody> <tr> <td>←</td><td>回上一頁</td></tr> <tr> <td>→</td><td>回下一頁</td></tr> <tr> <td>Q</td><td>縮小網頁</td></tr> <tr> <td>Q</td><td>放大網頁</td></tr> <tr> <td>⌂</td><td>返回首頁</td></tr> <tr> <td>C</td><td>重新載入此頁</td></tr> </tbody> </table>	圖示	描述	←	回上一頁	→	回下一頁	Q	縮小網頁	Q	放大網頁	⌂	返回首頁	C	重新載入此頁																												
圖示	描述																																										
←	回上一頁																																										
→	回下一頁																																										
Q	縮小網頁																																										
Q	放大網頁																																										
⌂	返回首頁																																										
C	重新載入此頁																																										
圓角	瀏覽器視窗預設為直角，若啟用圓角功能，將會修改為圓角。																																										
控制位址	<p>LW-n: 命令 ID</p> <table border="1"> <thead> <tr> <th>數值</th><th>描述</th></tr> </thead> <tbody> <tr> <td>0</td><td>無作用</td></tr> <tr> <td>1</td><td>前往目標網址</td></tr> <tr> <td>2</td><td>重新載入此頁</td></tr> <tr> <td>3</td><td>停止</td></tr> <tr> <td>4</td><td>回上一頁</td></tr> <tr> <td>5</td><td>回下一頁</td></tr> <tr> <td>6</td><td>清除快取</td></tr> <tr> <td>7</td><td>清除 cookie</td></tr> </tbody> </table> <p>LW-n+1: 狀態</p> <table border="1"> <thead> <tr> <th>數值</th><th>描述</th></tr> </thead> <tbody> <tr> <td>0</td><td>開始載入</td></tr> <tr> <td>1</td><td>停止載入</td></tr> <tr> <td>2</td><td>載入成功</td></tr> <tr> <td>3</td><td>載入失敗</td></tr> </tbody> </table> <p>LW-n+2: 錯誤碼</p> <table border="1"> <thead> <tr> <th>數值</th><th>描述</th></tr> </thead> <tbody> <tr> <td>0</td><td>無錯誤</td></tr> <tr> <td>1</td><td>內部錯誤</td></tr> <tr> <td>2</td><td>連線錯誤</td></tr> <tr> <td>3</td><td>憑證錯誤</td></tr> <tr> <td>4</td><td>HTTP 錯誤</td></tr> <tr> <td>5</td><td>FTP 錯誤</td></tr> </tbody> </table>	數值	描述	0	無作用	1	前往目標網址	2	重新載入此頁	3	停止	4	回上一頁	5	回下一頁	6	清除快取	7	清除 cookie	數值	描述	0	開始載入	1	停止載入	2	載入成功	3	載入失敗	數值	描述	0	無錯誤	1	內部錯誤	2	連線錯誤	3	憑證錯誤	4	HTTP 錯誤	5	FTP 錯誤
數值	描述																																										
0	無作用																																										
1	前往目標網址																																										
2	重新載入此頁																																										
3	停止																																										
4	回上一頁																																										
5	回下一頁																																										
6	清除快取																																										
7	清除 cookie																																										
數值	描述																																										
0	開始載入																																										
1	停止載入																																										
2	載入成功																																										
3	載入失敗																																										
數值	描述																																										
0	無錯誤																																										
1	內部錯誤																																										
2	連線錯誤																																										
3	憑證錯誤																																										
4	HTTP 錯誤																																										
5	FTP 錯誤																																										

	6 DNS 錯誤
目標頁面 URL 位址	可輸入欲前往的網址，長度上限為 255 字組。輸入完畢後，在 [控制位址] 的 [命令 ID] 輸入數值 1 即可前往。
當前頁面 URL 位址	顯示當前顯示的網址 URL，長度上限為 255 字組。

### 使用者腳本(背景)



使用者選擇一個 Userscript 的 JS 檔案，可以用於修改網頁代碼。此 User Script JS 檔案是整個專案共享的，也就是專案內所有的 PLC 網頁瀏覽器物件都會發生相同效果。

此功能 HMI 需使用 20230201 或更新的 OS 版本。

#### 13.55.3. 注意事項

1. 支援機型與 OS 版本請參考 EasyBuilder Pro 手冊附錄 A。
2. PLC 網頁瀏覽器是設計專門用於連接/顯示/操作 PLC 設備的網頁介面，若使用 PLC 網頁瀏覽器連接其它類型的網頁，可能會產生非預期結果或錯誤。
3. 當 PLC 網頁瀏覽器顯示內容較複雜的網頁時會占用較多的記憶體，並耗費大量的 CPU 運算能力，而影響到 HMI 的操作反應速度。建議用戶可以使用換頁功能或搭配直接視窗/間接視窗物件，在需要時才顯示 PLC 網頁瀏覽器物件，以避免影響系統效能。
4. 因 HMI 系統記憶體有限，當 PLC 網頁瀏覽器 (連接的網頁) 占用過多記憶體時，系統可能會主動關閉 PLC 網頁瀏覽器，避免影響 HMI 正常運作。

5. 因考量到效能因素，PLC 網頁瀏覽器只能在尚未開通 CODESYS 的人機上運行。
6. 因網頁技術更新太快，EasyBuilder Pro 無法確保 PLC 網頁瀏覽器物件可支援所有的 PLC 的網頁伺服器，可能有無法支援或無法正確顯示該 PLC 網頁伺服器情形，請以實體測試的結果為準。
7. 若單一視窗內放置多個 PLC 網頁瀏覽器物件，則只有一個會運行，其餘的則無作用。
8. 即便有其他物件設計疊加在 PLC 網頁瀏覽器物件上，實際運作時，PLC 網頁瀏覽器仍會在物件的最上層顯示。
9. 若使用 PC 版本、手機與平板裝置的 cMT Viewer，點擊 [PLC 網頁瀏覽器] 時，會用該平台的預設瀏覽器開啟。因此請注意，如希望開啟位於 HMI 本地的 PLC 網頁時，須確定運行 cMT Viewer 的裝置與 PLC 在同一網域，方可開啟。
10. cMT-iV5/cMT-iV6 無法使用 [PLC 網頁瀏覽器]。
11. 瀏覽器不支援的功能：
  - a. 另存檔案
  - b. 文字選取
  - c. 呼叫右鍵選單
  - d. 拖放
  - e. Windows prompt 彈出的帳號密碼輸入框 (例:FTP 登入)
  - f. 影片播放，例如：Youtube 影片等
  - g. 上傳或下載檔案
  - h. 由 Java 語言製作的網頁
12. 當 PLC 網頁瀏覽器無法正確顯示網頁上的文字時，請參考《15.5.5 完整字型檔案下載》，使 PLC 網頁瀏覽器支援非 OS 內建語系的字型。

## 13.56. 觸發條件

### 13.56.1. 概要

【觸發條件】物件可用來取代巨集 if-else 的判斷式，當條件符合時，可藉由輸出位元應用在其他物件上，例如：物件的安全設定、事件記錄、備份物件、資料傳輸等。【動作觸發(背景)】物件的觸發模式也可直接選用條件物件。僅 cMT/cMT X 系列支援此物件。

### 13.56.2. 設定



按下工作列上的【觸發條件】按鈕後即會出現【觸發條件】物件屬性對話窗，使用新增並正確設定條件內容後按下確認鍵，即可新增一個【觸發條件】物件。

#### 一般屬性設定



設定	描述
名稱	設定觸發條件物件名稱。
條件	設定觸發條件。
條件類型	條件類型可為位元、字組。



或是另一個條件物件。



#### 邏輯表達式

當運算子為 **AND** 時，代表上方所有條件皆須成立，此觸發條件才成立；當運算子為 **OR** 時，代表上方所有條件只須一個成立此觸發條件就算成立。



當使用自訂時，可使用更為複雜的邏輯表達式，**AND/OR** 以及 **( )** 的混用，會依照邏輯表達式的判斷當成觸發條件。

#### 輸出

若勾選，該觸發條件物件成立時，會將此位元位址設為 **ON**；該觸發條件物件不成立時，會將此位元位址設為 **OFF**。



- [觸發條件] 物件最多可新增 128 個。
- 一個 [觸發條件] 物件的條件類型最多可設定 8 個。

## 範例 1

以下示範 [觸發條件] 物件來取代巨集 if-else 判斷式的簡易使用方法。

如果有一個巨集內容如下，if-else 判斷式內包含三種條件，則可以透過 [觸發條件] 物件取代此巨集。

```

macro_command main()
    bool condition_A
    short condition_B, condition_C

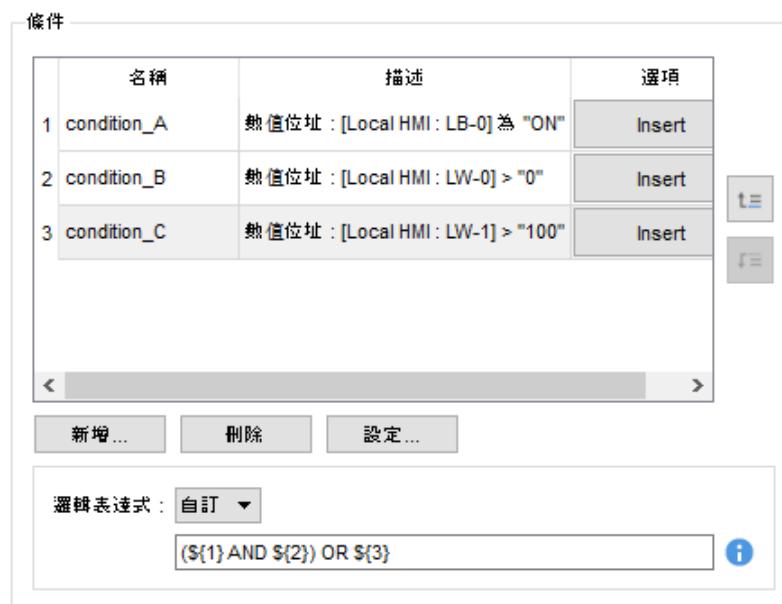
    GetData(condition_A, "Local HMI", LB, 0, 1)
    GetData(condition_B, "Local HMI", LW, 0, 1)
    GetData(condition_C, "Local HMI", LW, 1, 1)

    if (condition_A == 1 and condition_B > 0) or condition_C > 100 then
        //Execute content
    end if
end macro_command

```

\* 點選滑鼠右鍵可顯示編輯選單。  
注意 W2: 變數未初始化 : condition\_A  
注意 W2: 變數未初始化 : condition\_B  
注意 W2: 變數未初始化 : condition\_C  
0 error(s) 3 warning(s)

1. 建立一個觸發條件物件並新增 condition\_A，condition\_B 與 condition\_C 三個條件，運算子使用自訂並使用邏輯表達式 \${1} AND \${2}) OR \${3}，如下圖所示。



2. 此時觸發條件物件已經成功取代巨集指令，可依需求設定成功觸發後的動作內容。

## 13.57. FTP 連線

### 13.57.1. 概要

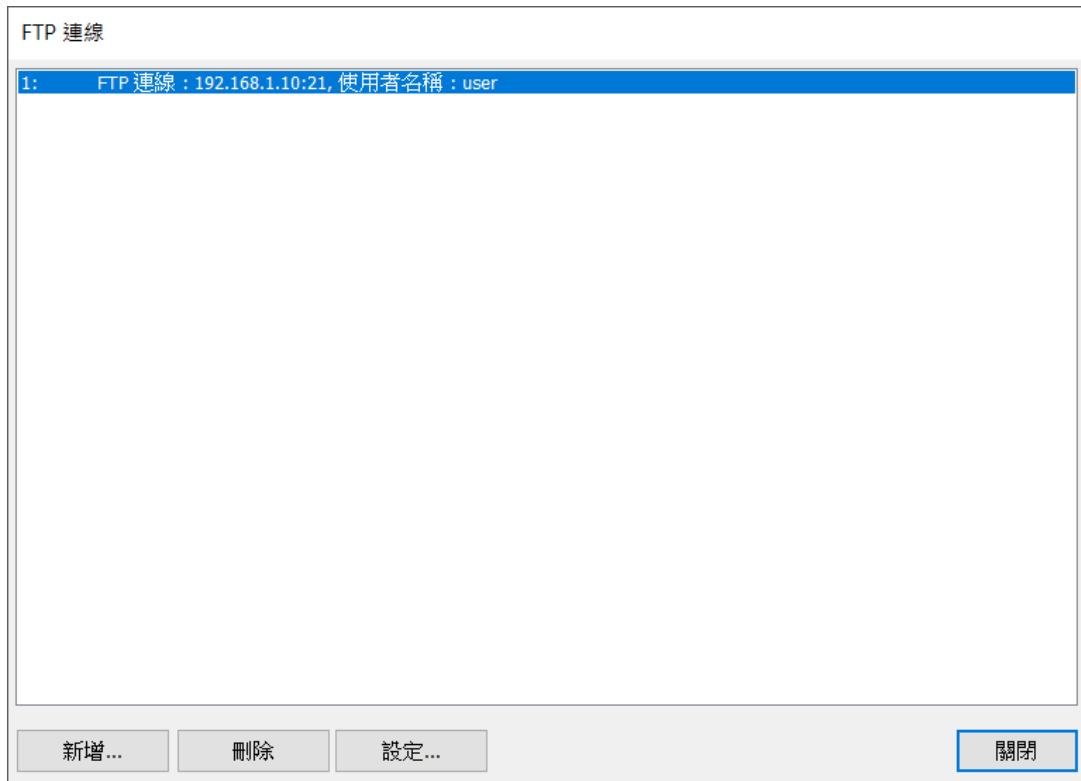
設定 FTP 伺服器的連線資訊。啟用此功能後，HMI 上可將檔案備份到 FTP 伺服器，亦可從 FTP 接收檔案。使用 [FTP 連線] 上傳備份檔案（畫面擷取或歷史資料）到電腦，可以達到類似透過 EasyPrinter 備份歷史資料的功能。

僅 cMT/cMT X 系列支援此功能。

### 13.57.2. 設定



按下工作列上的 [資料/歷史] » [FTP 連線] 按鈕後即會出現 [FTP 連線] 屬性對話窗，使用新增並正確設定後按下確認鍵，即可新增一個 [FTP 連線] 項目。



## 一般屬性設定



設定	描述
伺服器位址	<p><b>靜態</b> 使用預先設置的連線參數，例如：類型、IP、連接埠等。</p> <p><b>動態</b> 透過控制地址設置 FTP 的連線參數。</p>

**類型****純粹 FTP**

使用明碼通訊。

**透過隱含式 TLS 的 FTP**

全程使用加密通訊。預設使用 TCP 通訊埠 990。

**透過外顯式 TLS 的 FTP**

提出加密要求後(AUTH)，開始進行加密通訊。預設使用 TCP 通訊埠 21。

**主機**

FTP 伺服器的 IP 位址或網域名稱。

**連接埠號**

FTP 伺服器的通訊埠。

**使用者名稱**

FTP 伺服器的使用者名稱。

**密碼**

FTP 伺服器的使用者密碼。

**Note**

- [FTP 連線] 刪除項目時，若出現 [此伺服器已被使用，無法刪除] 的錯誤訊息，請至 [常用] 選單 » [系統參數] 物件 » [FTP] 分頁停用 FTP 功能。

## 13.58. 日期/時間

### 13.58.1. 概要

【日期/時間】物件可於人機上顯示日期與時間，使用者可設定顯示格式與字型。

### 13.58.2. 設定



請直接點擊【日期/時間】圖示建立此物件，或點選工具列上的【物件】»【時間相關】»【日期/時間】新增此物件。

#### 一般屬性設定



設定	描述
日期	設定顯示日期格式。
星期	設定星期顯示文字。
時間	設定顯示時間格式。
字型	選擇顯示的字型/顏色/尺寸與對齊方式。

## 13.59. 模板

### 13.59.1. 模板視窗

#### 13.59.1.1. 概要

[模板視窗] 物件可以用來設定 [模板] 物件的來源視窗。

#### 13.59.1.2. 設定



按下工作列的 [物件] » [模板] » [模板視窗] 按鈕後即會開啟 [模板視窗] 物件對話窗，按下 [新增] 後設定模板視窗範圍按下 [確認] 鍵，即可新增一個 [模板] 物件。



## 13.59.2. 模板

### 13.59.2.1. 概要

[模板] 物件將根據 [模板視窗] 內的物件替換物件的地址或文字減少重複性的介面設計操作。

### 13.59.2.2. 設定

#### 一般屬性設定



設定	描述
描述	自定義物件的描述。
視窗編號	選擇來源模板視窗。
標籤名稱對應	在標籤名稱中尋找自訂的字串並替換為另一自訂的字串。
文字對應	在文字物件、複合式多功能按鈕以及操作記錄的

文字標籤中尋找自訂的字串並替換為另一自訂的字串。



### Note

- 標籤名稱對應替換僅支援使用者定義標籤及標籤型態(Symbolic Addressing)的地址。
- 標籤名稱對應功能支持在多個裝置上進行標籤的批量替換。例如，您可以將裝置 1 的標籤 A 更改為標籤 B，同時將裝置 2 的標籤 C 更改為標籤 D。但請注意，每個替換動作僅限於同一裝置內的標籤。
- 文字對應不支援使用文字標籤庫的物件。
- 模板物件僅在編輯時期有效，編譯後將產生獨立物件，也就是說當反編譯.exob/.cxob 檔案時，反編譯後的工程檔案將只有獨立的物件，而不是模板物件。
- 模板視窗內不能包含模板物件。
- 因模板物件是在編譯時才生成獨立物件，位址檢視及位址標籤庫無法顯示模板物件內替換地址後的使用狀態。
- 超出模板視窗範圍的物件會被移除不被模板物件顯示，包含替換文字內容後超出模板視窗的物件。

### 範例

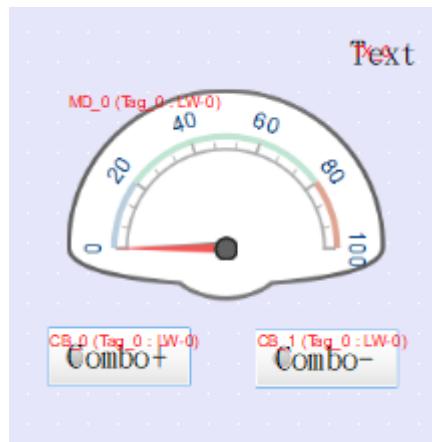
1. 在位址標籤庫新增兩個標籤並分別命名為 Tag\_0, Tag\_1。

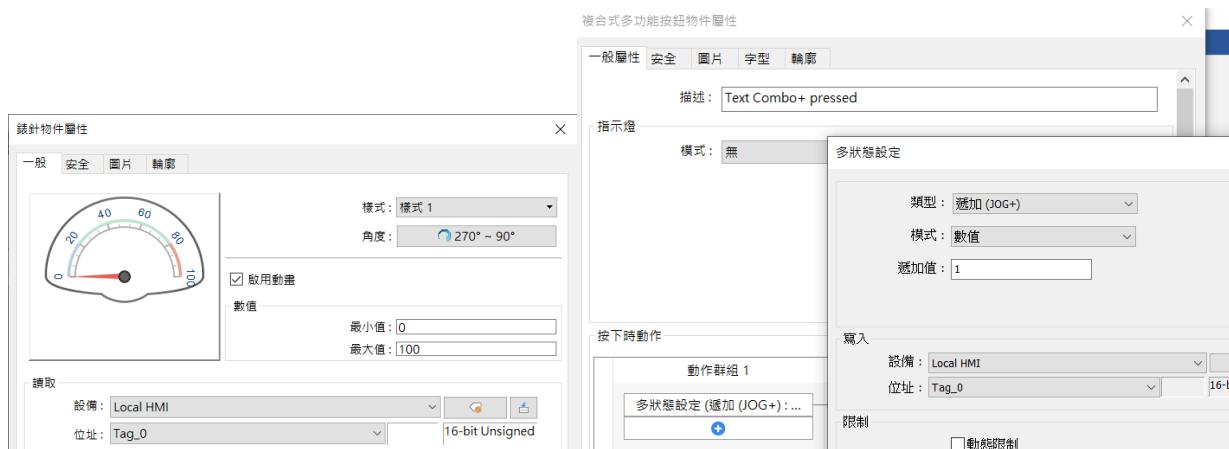
Tag_0	Local HMI	LW-0	字組
Tag_1	Local HMI	LW-1	字組

2. 新增一視窗並設定其視窗大小，此設定將影響模板物件的大小。



3. 在步驟 2 新增的視窗內，放置錶針物件、複合式多功能按鈕物件，讀取/寫入位址設置為步驟 1 新增的 Tag\_0。





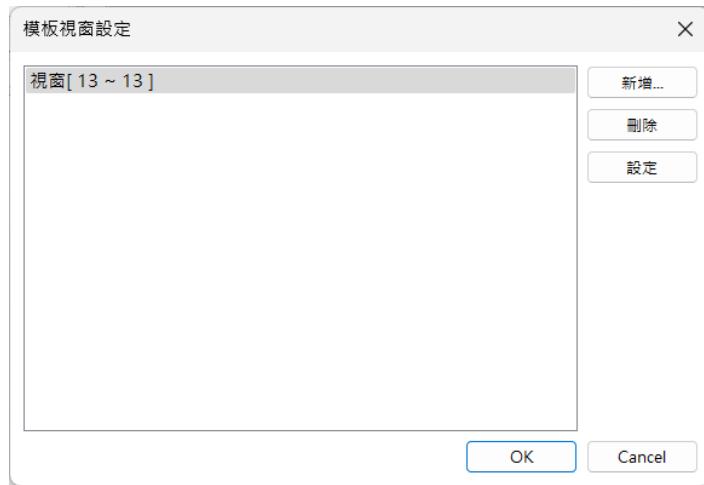
4. 設定複合式多功能按鈕物件的描述為 Text Combo+ pressed，以及設定其文字標籤為 Combo+。



5. 在步驟 2 新增的視窗內放置一文字物件，並將其內容設定為 Text。



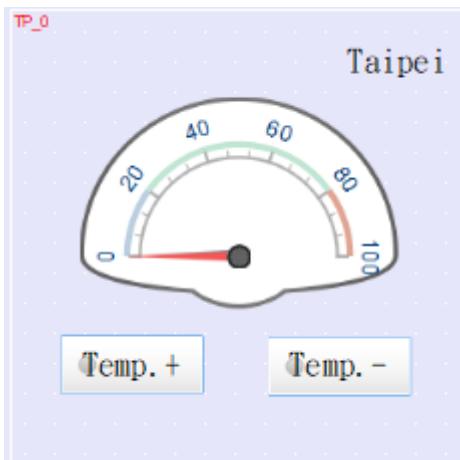
6. 將步驟 2 新增的視窗設定為模板視窗。



7. 在畫面上新增一模板物件，並參考以下設定。



8. 將專案編譯後，EasyBuilder Pro 會將模板視窗內的物件使用到的位址含有 0 的標籤名稱替換為 1，因此 Tag\_0 將會被替換為 Tag\_1，文字對應則會根據模板物件設定對物件描述、文字內容以及文字標籤等作相對應的替換。



# 14. 向量圖庫與圖片庫的建立

本章節說明如何建立向量圖庫與圖片庫。

14.1.	概要 .....	14-2
14.2.	向量圖庫的建立 .....	14-2
14.3.	圖片庫的建立 .....	14-9
14.4.	向量圖庫 / 圖片庫批次管理.....	14-15

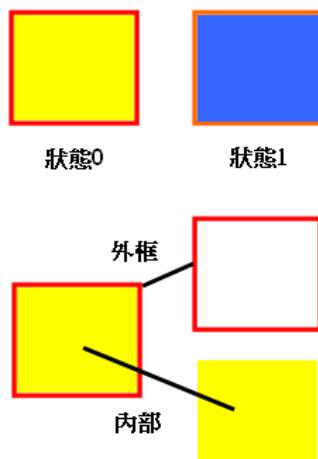
## 14.1. 概要

EasyBuilder Pro 提供向量圖庫與圖片庫的使用。在圖庫管理上提供 [工程檔案] 及 [圖庫] 二種模式。[工程檔案] 的圖片被儲存在 .emtp 的工程檔案內，[圖庫] 的圖片則會被儲存在 EasyBuilder Pro 圖庫目錄或是使用者自訂的路徑下。圖片可以增加物件在視覺上的效果，每個向量圖或圖片最多可包含 256 個狀態。下文將說明如何建立向量圖庫與圖片庫。

👉 向量圖庫與圖片庫的使用請參考《9 物件一般屬性》。

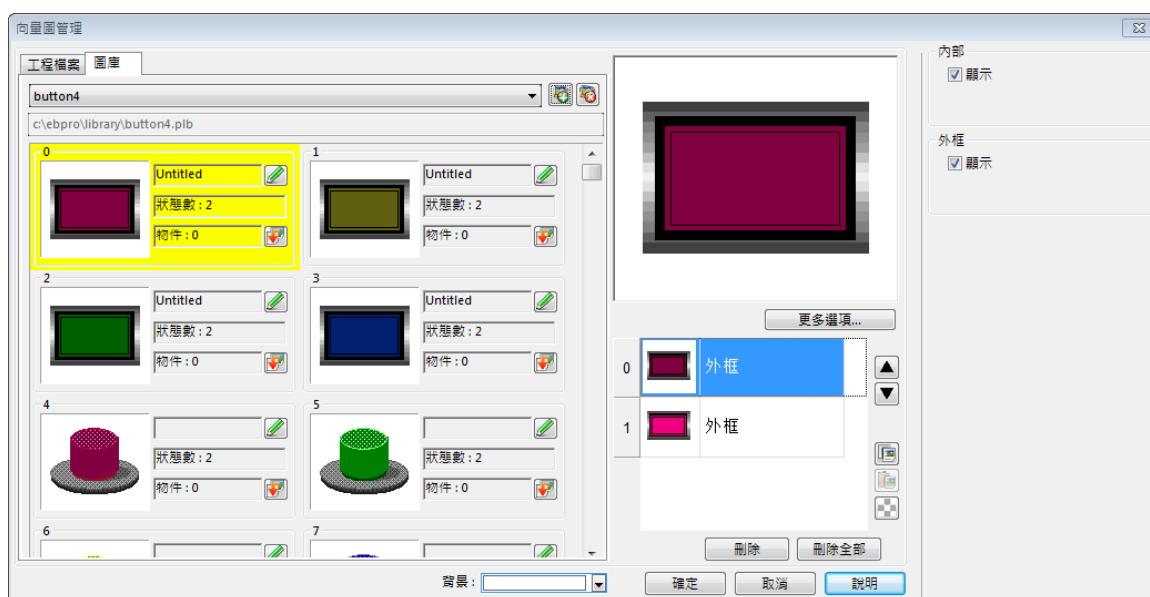
## 14.2. 向量圖庫的建立

向量圖是由直線、曲線、多邊形等繪圖元件所構成的圖形；一個完整的向量圖可能具有一個以上的狀態，每個狀態都可包含兩個部分：外框與內部，如下圖所示。



### 14.2.1. 向量圖管理

物件可以單獨選擇使用向量圖的外框及內部，或者同時使用。在按下工具列的 [工程檔案] » [向量圖] 按鈕，即可進入 [向量圖管理] 對話窗，如下圖所示。



設定	描述
<b>工程檔案</b>	在此頁籤編輯的向量圖，將儲存於 .emtp 工程檔案中，最多可新增 1000 個向量圖。
<b>圖庫</b>	在此頁籤編輯的向量圖，將儲存於圖庫目錄下，但並不會儲存於 .emtp 工程檔案中。
<b>新增圖庫</b>	新增已存在的 .plib 向量圖庫檔案。若要新增一個全新的圖庫，則輸入不存在的檔名後按 [開啟]，將會建立一個空白的圖庫檔案，最多可新增 40 個圖庫。
<b>刪除圖庫</b>	刪除當前選擇的圖庫。
<b>複製至工程檔案</b>	將此向量圖複製到 [工程檔案] 中。僅支援非系統向量圖庫的圖形複製。以下 4 種系統向量圖庫不支援複製功能 System Frame/System Button / System Lamp / System Pipe。
<b>背景</b>	選擇向量圖的背景顏色，此顏色只會在 [向量圖管理] 中顯示，實際圖片並不會有此背景顏色。
<b>更多選項</b>	可以設定 [內部]、[外框] 以及 [圖案] 的顏色及樣式。
<b>上移 / 下移</b>	將此向量圖形往前 / 往後移一個狀態。
<b>複製</b>	複製此向量圖形。
<b>貼上</b>	貼上已複製的向量圖形。

## 向量圖庫與圖片庫的建立



在此向量圖形後插入一個空白狀態。

**刪除**

刪除此向量圖形的某一個狀態。

**刪除全部**

刪除所選擇的向量圖形的全部狀態。

**確定**

確定儲存此次編輯結果。

**取消**

取消此次編輯結果。

**說明**

開啟說明文件。



### Note

- 向量圖庫中可選擇 [內部]、[外框] 的色彩，以及 [圖案式樣] 的功能只開放給 System Frame/System Button 圖庫。



- cMT / cMT X 系列的 [圖案式樣] 功能支援漸層樣式，如下圖所示：

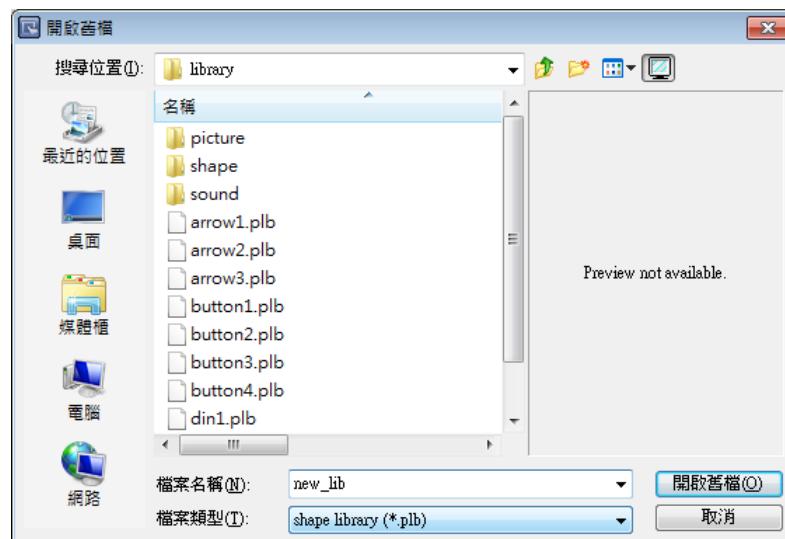


#### 14.2.2. 建立向量圖庫的步驟

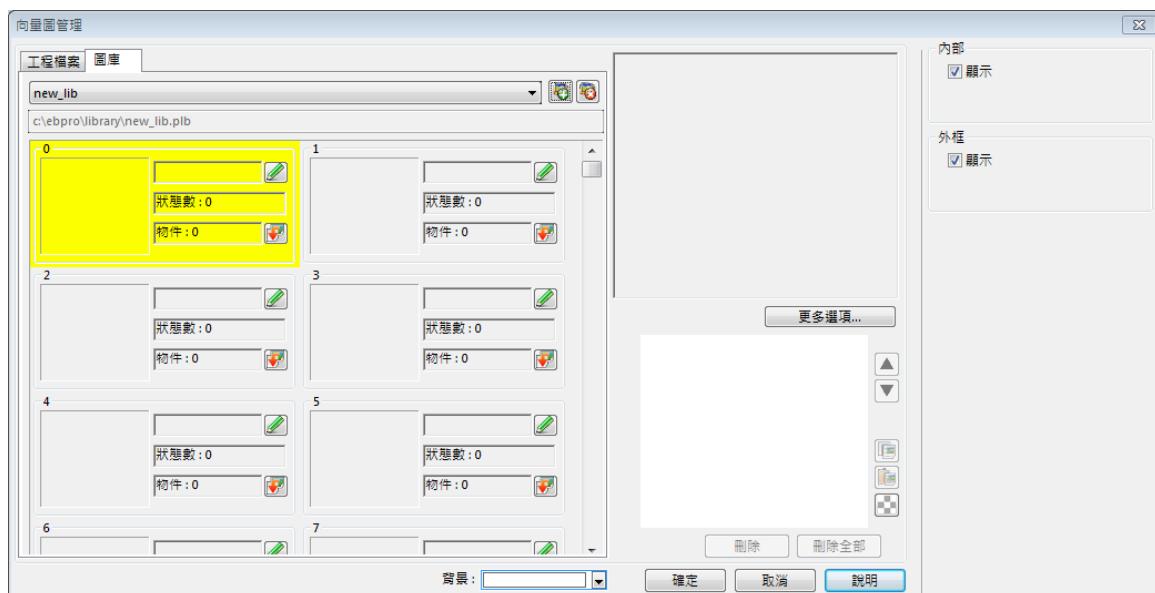
下面說明如何建立一個新的向量圖庫，並在此圖庫中加入一個具有兩個狀態的向量圖。

1. 按下 [新增圖庫] 後，在對話窗中輸入新的向量圖庫名稱 “new\_lib”。

## 向量圖庫與圖片庫的建立

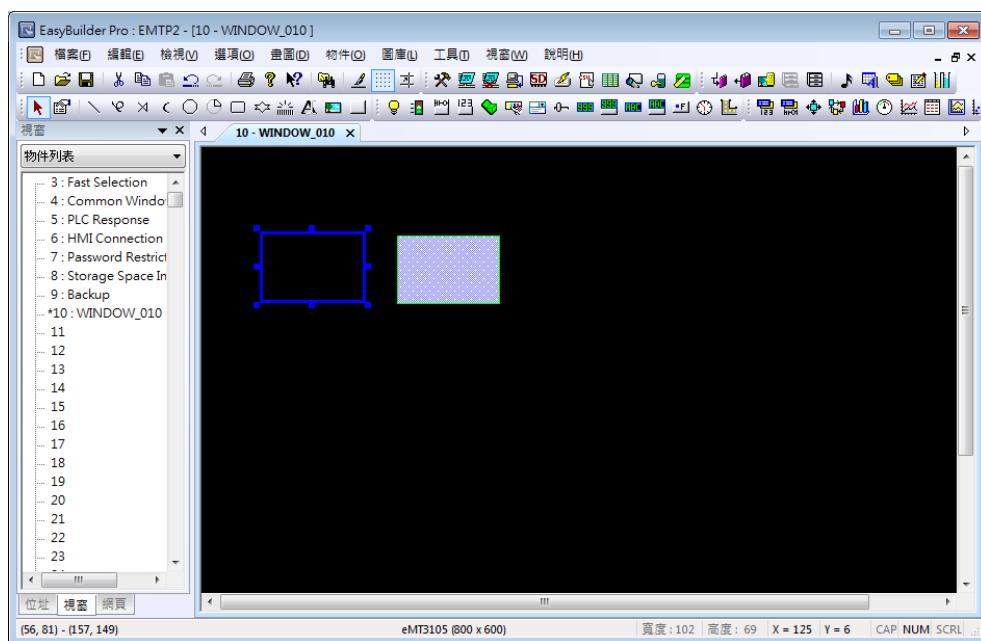


2. 點選 [開啟舊檔] 後，會彈出詢問是否要建立檔案的對話框，接著按 [是] 進行下一步。
3. 此時可以發現 [向量圖管理] 中增加一個新的向量圖庫 [new\_lib]，在圖庫名稱的下方會顯示此圖庫的路徑，此新的向量圖庫中並未包含任何向量圖，如下圖所示。

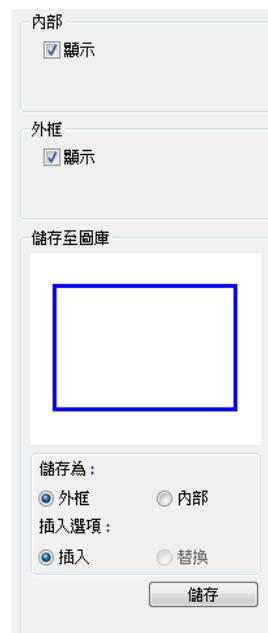


4. 對特定向量圖，加入一個狀態。首先使用繪圖工具在視窗上繪出需要的外框與內部圖形，並圈選要加到向量圖庫的外框圖形。

## 向量圖庫與圖片庫的建立



5. 接著在圈選圖形元件後，按下工具列上的 [工程檔案] » [向量圖] » [儲存至向量圖庫] 按鈕，選擇已建立的 [new\_lib] 圖庫，接著選擇要儲存的向量圖編號，被選擇的向量圖會顯示黃色的背景。
6. 接下來設定將此向量圖形儲存為此狀態的 [外框]，插入選項選擇 [插入] 並點選 [儲存]。



設定	描述
內部	設定是否顯示向量圖形內部。
外框	設定是否顯示向量圖形外框。
儲存至圖庫	<b>儲存為外框</b> 將此圖形儲存為向量圖外框。 <b>儲存為內部</b>

## 向量圖庫與圖片庫的建立

將此圖形儲存為向量圖內部。

### 插入

選擇插入一個新的向量圖狀態。

### 替換

選擇替換此向量圖狀態。

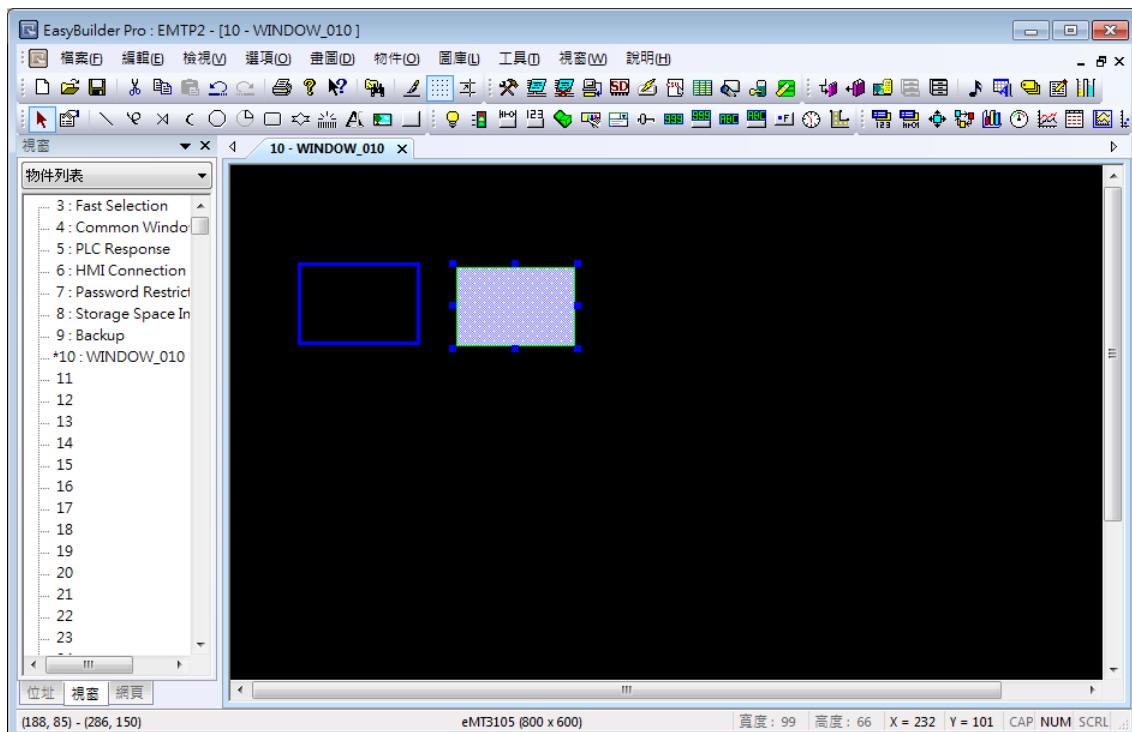
### 儲存

確定儲存以上設定。

7. 由下圖可以看到此向量圖已新增了一個狀態，它的外框已被定義完成。

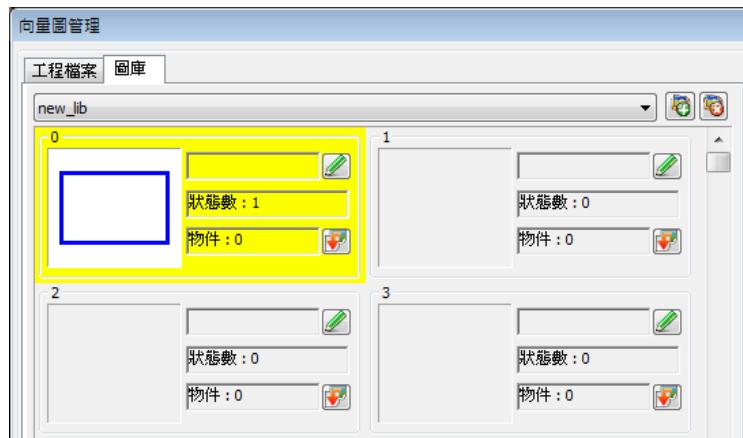


8. 設定內部圖形。在視窗上圈選已繪好的內部圖形。

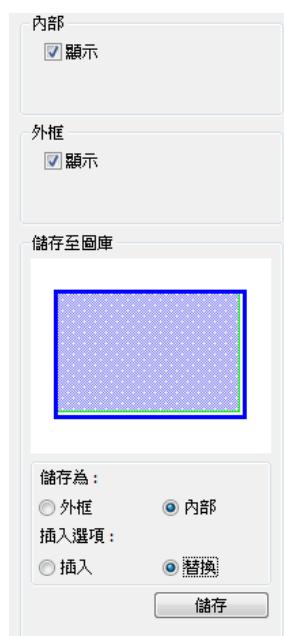


9. 按下工具列上的 [儲存至向量圖庫] 按鈕，選擇已建立的 [new\_lib] 圖庫，接著選擇儲存與外框相同的向量圖編號。

## 向量圖庫與圖片庫的建立



- 10.** 設定將此向量圖形儲存為此狀態的 [內部]，插入選項選擇 [替換]，點選 [儲存]。

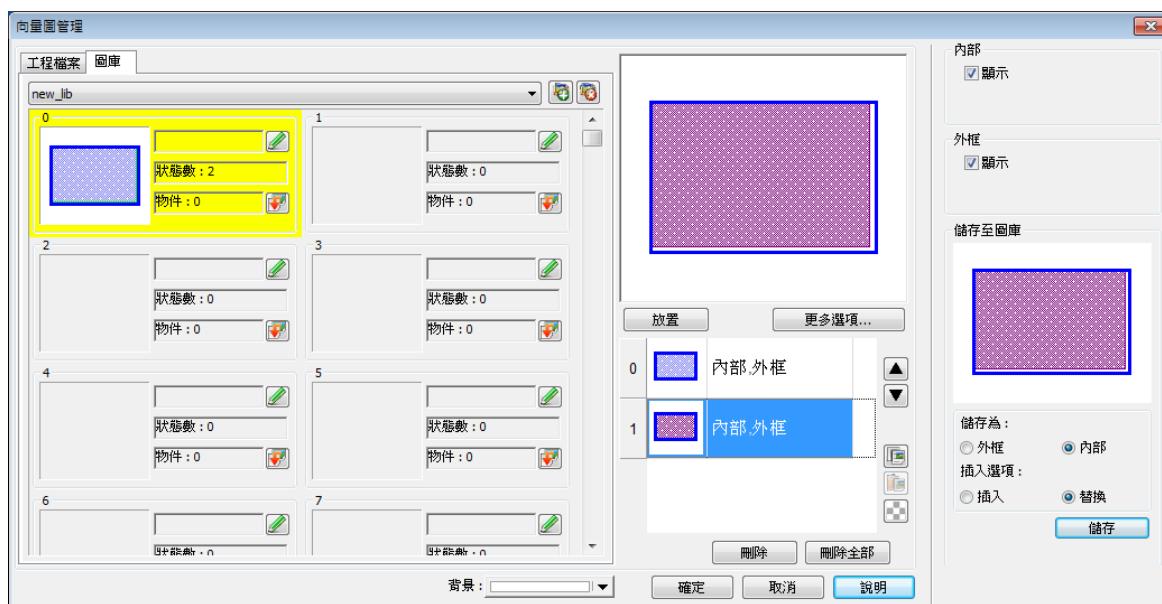


- 11.** 一個狀態的向量圖形可以只有 [內部] 或是 [外框]，也可以同時存在。由下圖可看到狀態 0 的向量圖形已具備 [內部] 與 [外框] 的設定。按下 [確定] 後，狀態 0 的圖形即建立完成。



- 12.** 使用與狀態 0 相同的方式，插入一個向量圖形做為狀態 1，如下圖所示，此向量圖編號已建立包含兩個狀態的向量圖形，最後按下 [確定] 即完成設定。

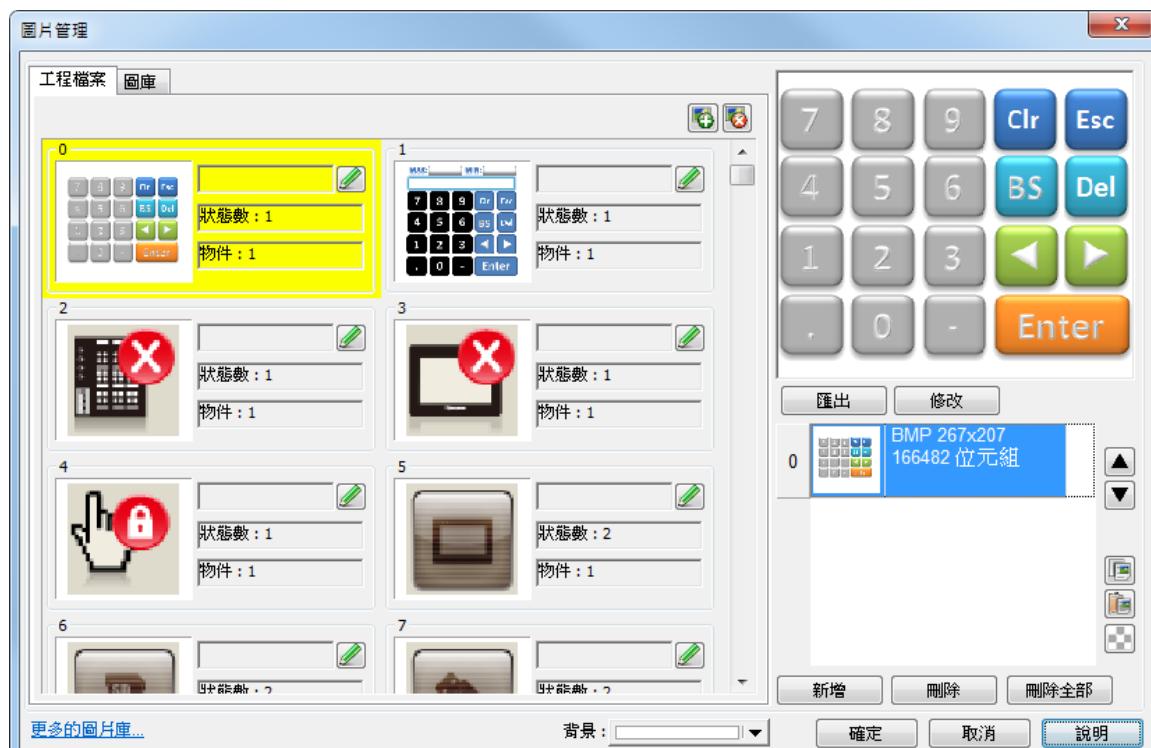
## 向量圖庫與圖片庫的建立



### 14.3. 圖片庫的建立

#### 14.3.1. 圖片管理

按下工具列上的 [工程檔案] » [圖片] 按鈕後將出現 [圖片管理] 對話窗，如下圖所示。



#### 設定

#### 描述

#### 工程檔案

在此頁籤編輯的圖片，將儲存於 .emtp 工程檔案中。最

多可新增 1000 個圖片。

**圖庫**

在此頁籤編輯的圖片，將儲存於圖庫目錄下，不會儲存於 .emtp 工程檔案中。

**新增圖庫**

新增已存在的 .flbx & .flb 圖庫檔案，若要新增一個全新的圖庫，則輸入不存在的檔名後按 [開啟]，將會建立一個空白的圖庫檔案，最多可新增 40 個圖庫。

**刪除圖庫**

刪除當前選擇的圖庫。

**瀏覽圖庫**

可設定多個路徑後預覽圖庫的內容。

**複製至工程檔案**

將此圖片複製到 [工程檔案] 中。

**背景**

選擇圖片的背景顏色，此顏色只會在 [圖片管理] 中顯示，實際圖片並不會有此背景顏色。

**更多的圖片庫**

連結至 Weintek 官方網站的圖片庫下載區，登入帳號後即可下載更多的圖片庫。

**匯出**

將此圖片匯出。

**修改**

修改此圖片設定。

**複製**

複製此圖片。

**貼上**

貼上已複製的圖片。支援將剪貼簿中的圖片，利用 [貼上] 功能將圖片匯入至圖庫。

**插入透明狀態**

在此圖片後插入一個空白狀態。

**新增**

新增一個圖片。

**刪除**

刪除此圖片。

**刪除全部**

刪除全部的圖片。

**確定**

確定儲存此次編輯結果。

**取消**

取消此次編輯結果。

**說明**

開啟說明文件。

**Note**

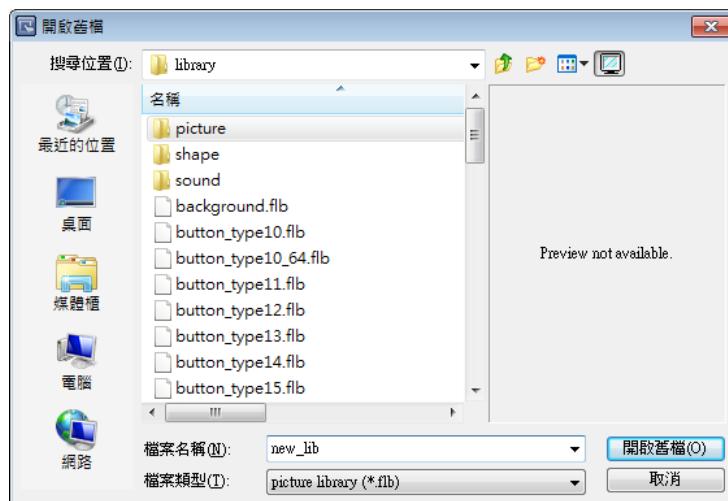
- 圖片庫支援的圖片格式為 .bmp、.jpg、.gif、.dpd、.svg 和 .png。在圖片庫新增 .gif 格式的圖檔時，如果圖檔屬於動畫類型的檔案，使用者可以設定動畫的播放次數。如下圖所示



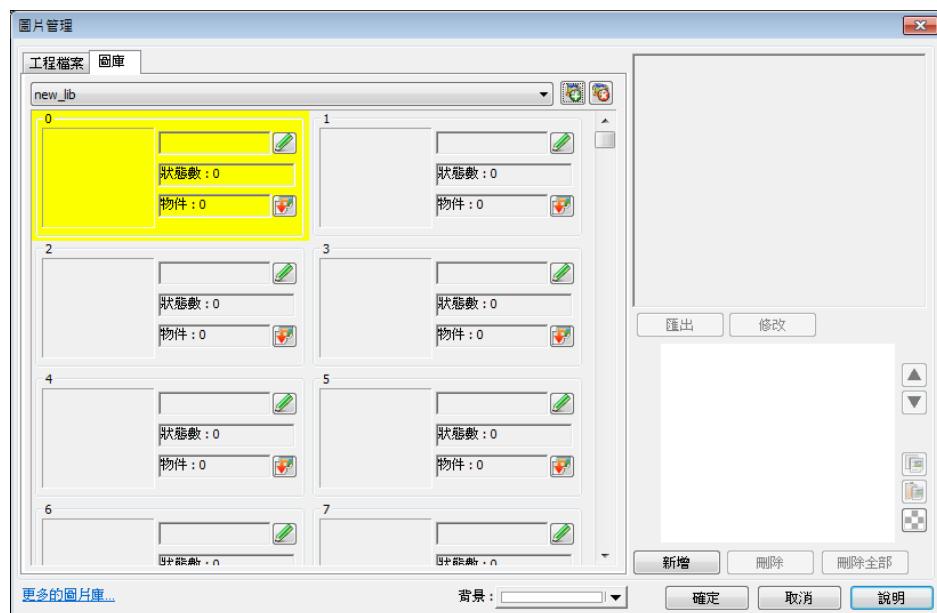
### 14.3.2. 建立圖片庫的步驟

下面說明如何建立一個新的圖片庫，並在此圖庫中加入一個具有兩個狀態的圖片。

- 按下 [新增圖庫] 後，在對話窗中輸入新的圖片庫名稱。

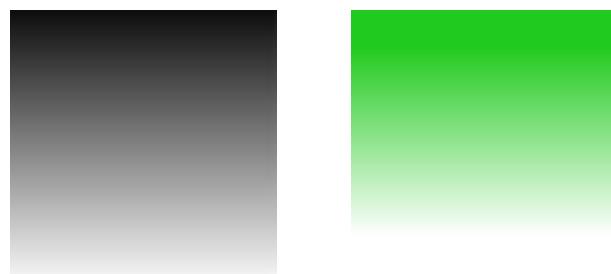


- 點選 [開啟舊檔] 後，會彈出詢問是否要建立檔案的對話框，接著按 [是] 進行下一步。
- 此時可以發現 [圖片管理] 對話窗中增加一個新的圖庫 [new\_lib]，且此新的圖庫中並未包含任何圖片，如下圖所示。

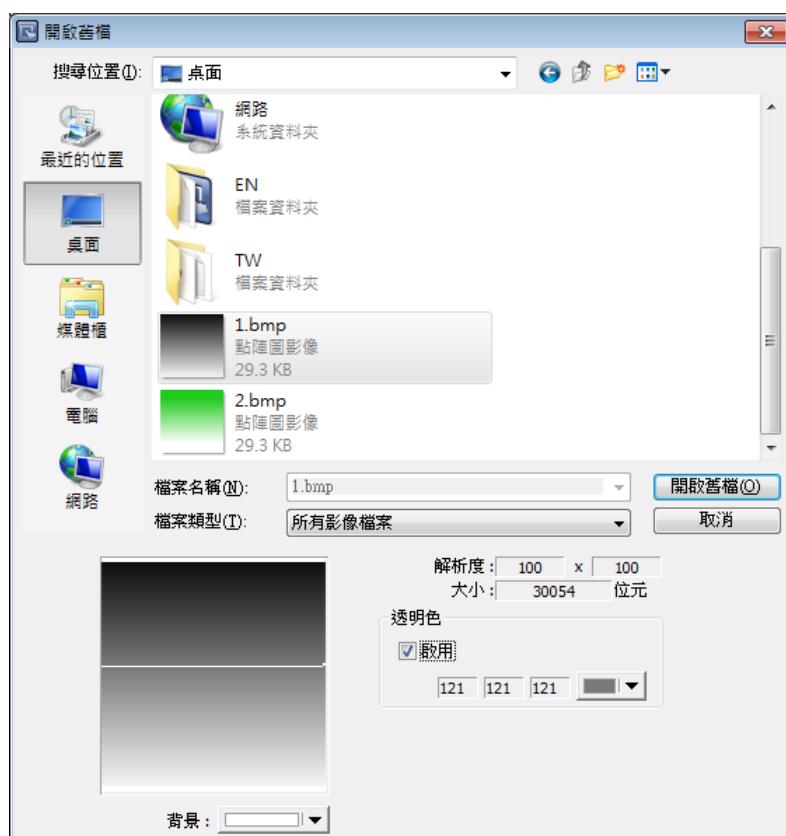


## 向量圖庫與圖片庫的建立

4. 假設要將下面兩張圖片分別用來表示狀態 0 與狀態 1。

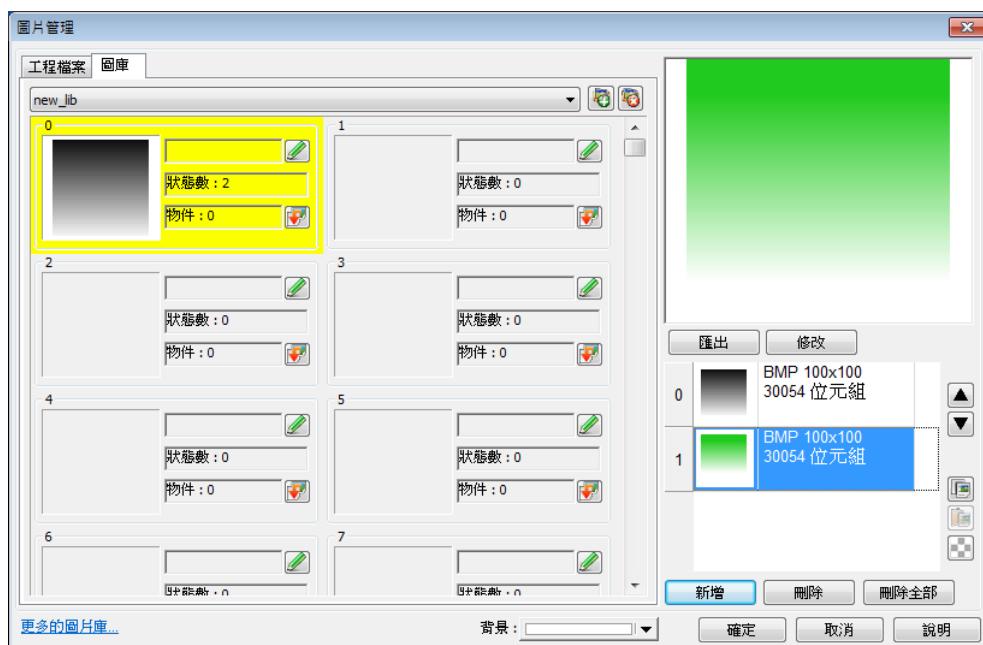


5. 首先選擇已建立的 [new\_lib] 圖庫，接著選擇圖片欲儲存的圖片編號，被選擇的圖片編號會顯示黃色的背景。
6. 按 [新增] 後，選擇狀態 0 的圖片檔案來源。
7. 此時出現下圖的對話窗時，若勾選 [啟用] 來使用透明色，並設定 RGB (121, 121, 121)，會將此顏色區域設定為透明色，或是使用滑鼠點擊想要作為透明色的位置，系統將自動判斷 RGB 的數值。



8. 此時已完成狀態 0 的圖片設定，接著 [新增] 狀態 1 的圖片，步驟與狀態 0 相同。在 [圖片管理] 對話窗中新加入的圖片為編號 0，由圖片資訊中也可看出此圖片為 bitmap 形式，且包含兩個狀態，如下圖所示。

## 向量圖庫與圖片庫的建立



- 在完成上述的各項動作後，按下 [確定] 即可建立一個完整的圖片。

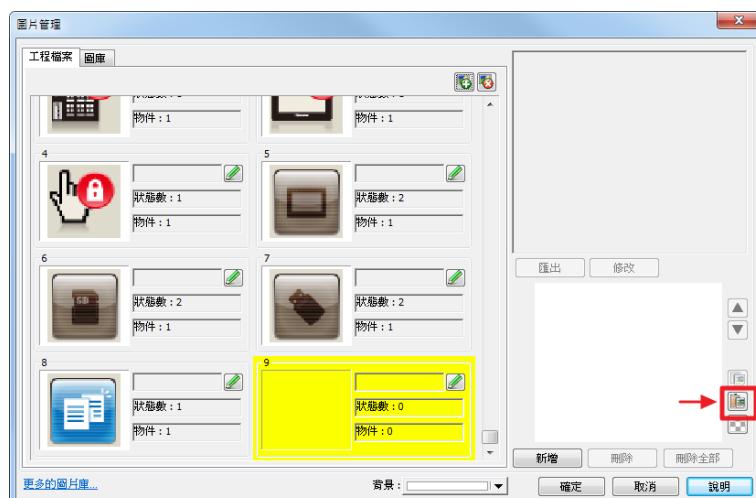
### 14.3.3. 從剪貼簿匯入圖片

下面說明如何從剪貼簿中將圖片匯入至圖庫。

- 假設剪貼簿中有以下這張圖片。

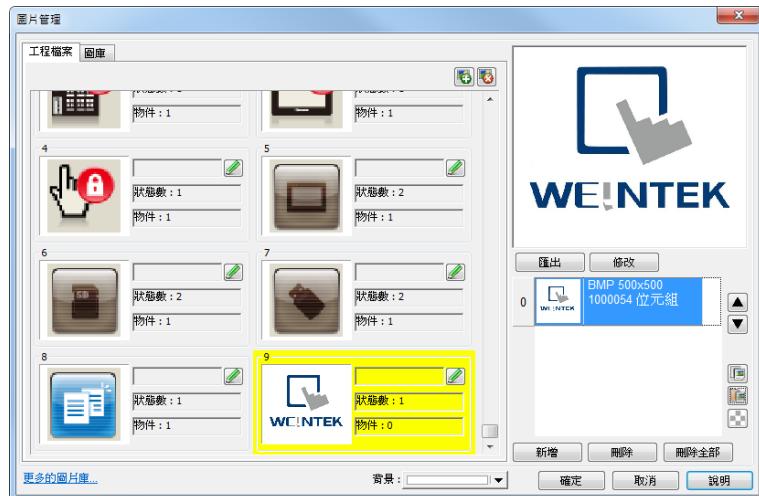


- 點選圖片庫右方的 [貼上] 按鈕。



- 圖片就可輕鬆匯入。

## 向量圖庫與圖片庫的建立

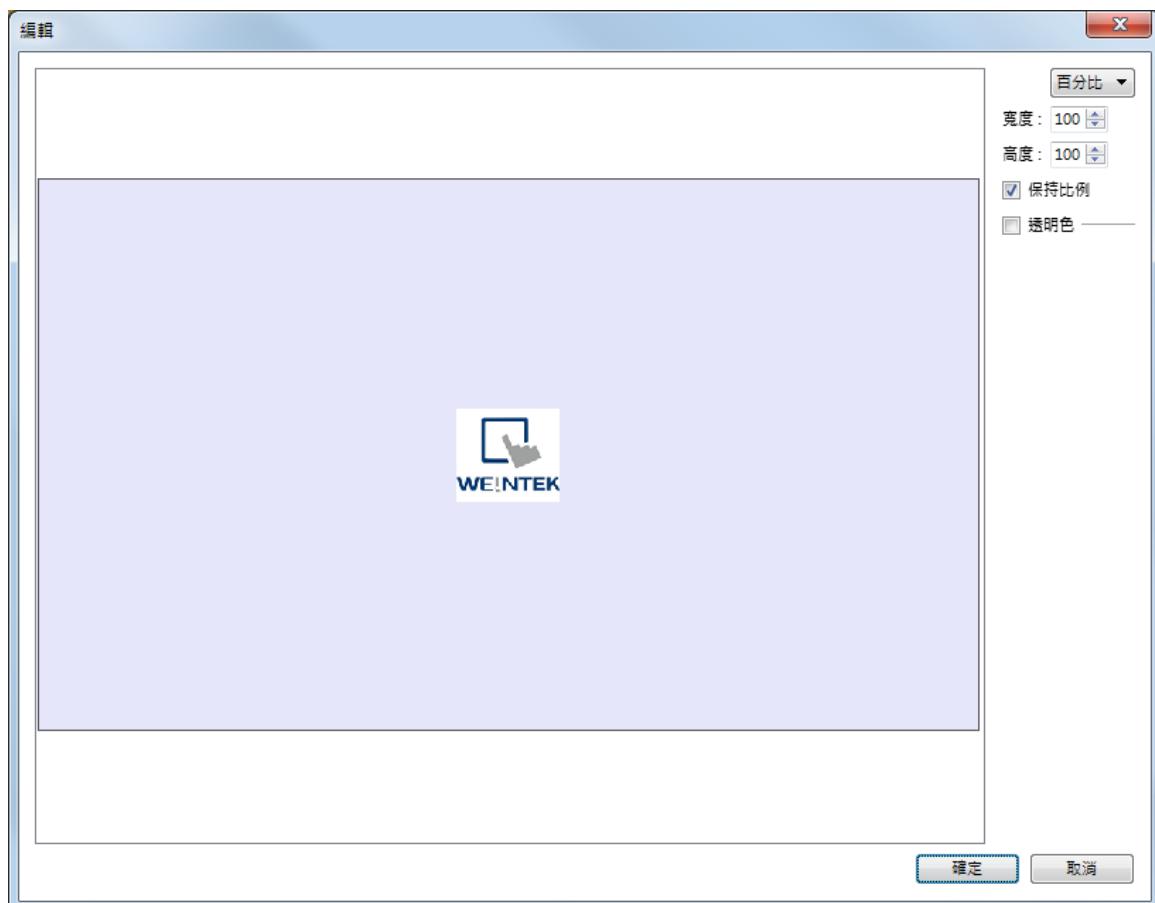


此外，亦可直接從工程檔案編輯視窗貼上來自剪貼簿的圖片。

1. 假設剪貼簿中有以下這張圖片。



2. 於編輯視窗按 Ctrl + V，可彈出以下視窗。



3. 點選 [確定] 後，圖片即被放置於視窗上。
4. 點選該圖片查看屬性，可發現該圖片被存放於 [工程檔案] 內。



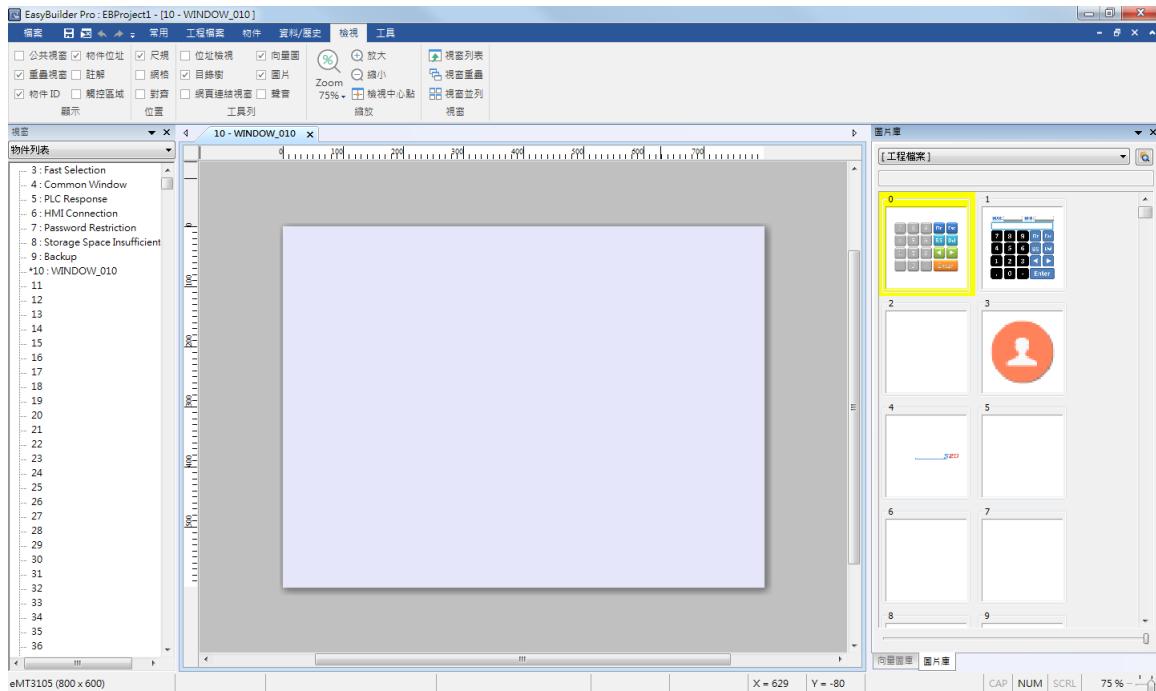
### Note

- 僅 .bmp、.dpd 和 .jpg 檔案可使用透明色。

## 14.4. 向量圖庫 / 圖片庫批次管理

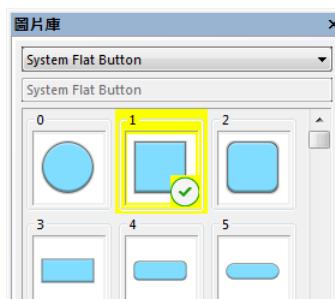
向量圖庫、圖片庫的管理視窗，可用來即時修改單張圖片，或同時修改多張圖片。部分圖庫亦支援快速地改變圖片顏色功能。

## 向量圖庫與圖片庫的建立



### 14.4.1. 圖片切換

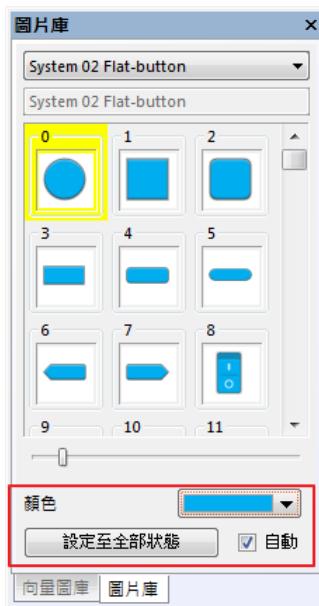
1. 選擇要更換圖片的物件，可選擇一個或多個物件。
2. 在圖片庫管理視窗選擇新圖片，點擊該圖片右下角套用 。



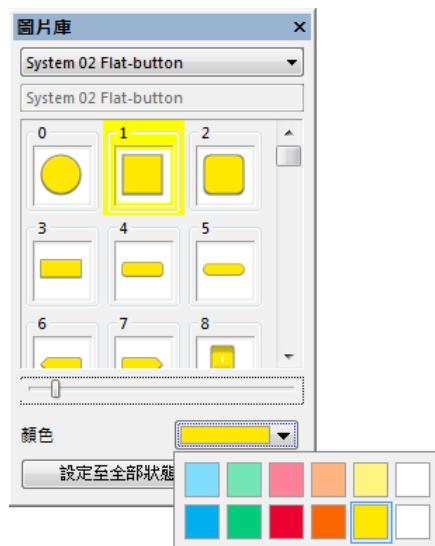
### 14.4.2. 顏色切換

部分圖庫支援快速切換顏色的功能。支援此功能的圖庫，在圖庫管理視窗的下方會有 [顏色] 的設定欄位。

## 向量圖庫與圖片庫的建立



1. 選擇要更換圖片顏色的物件，同樣的圖片可批次切換顏色。
2. 在圖片庫管理視窗的【顏色】下拉選單中選擇圖片的顏色。

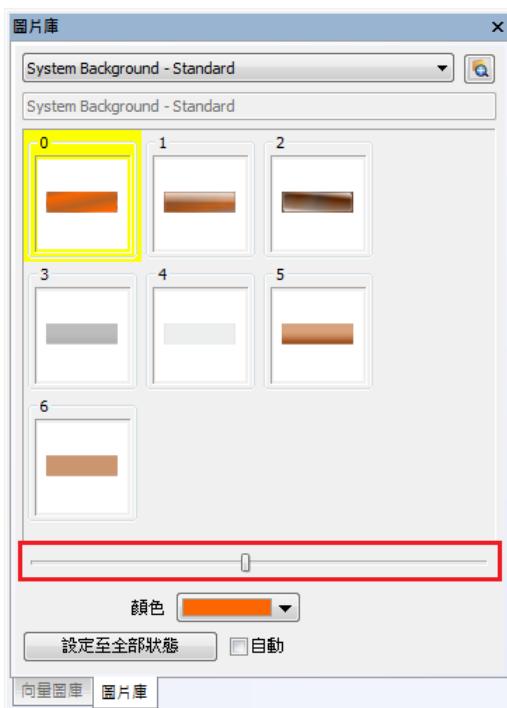


3. 點擊欲使用的新的圖片的右下角套用。

設定	描述
設定至全部狀態	將選取的顏色套用至此物件的所有狀態。(需選擇同一圖片)
自動	自動執行【設定至全部狀態】功能。

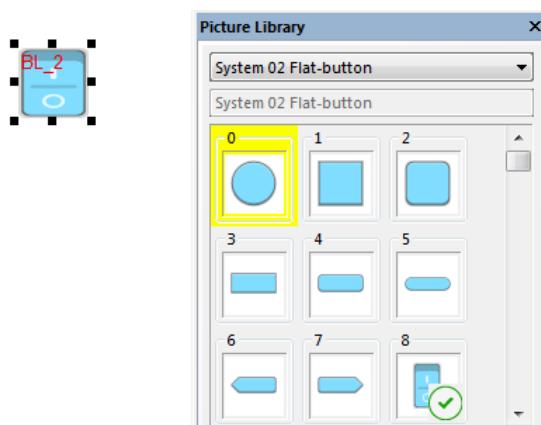
## 14.4.3. 特殊功能

- 在預覽視窗中，可透過調整滑動條改變預覽圖的大小。



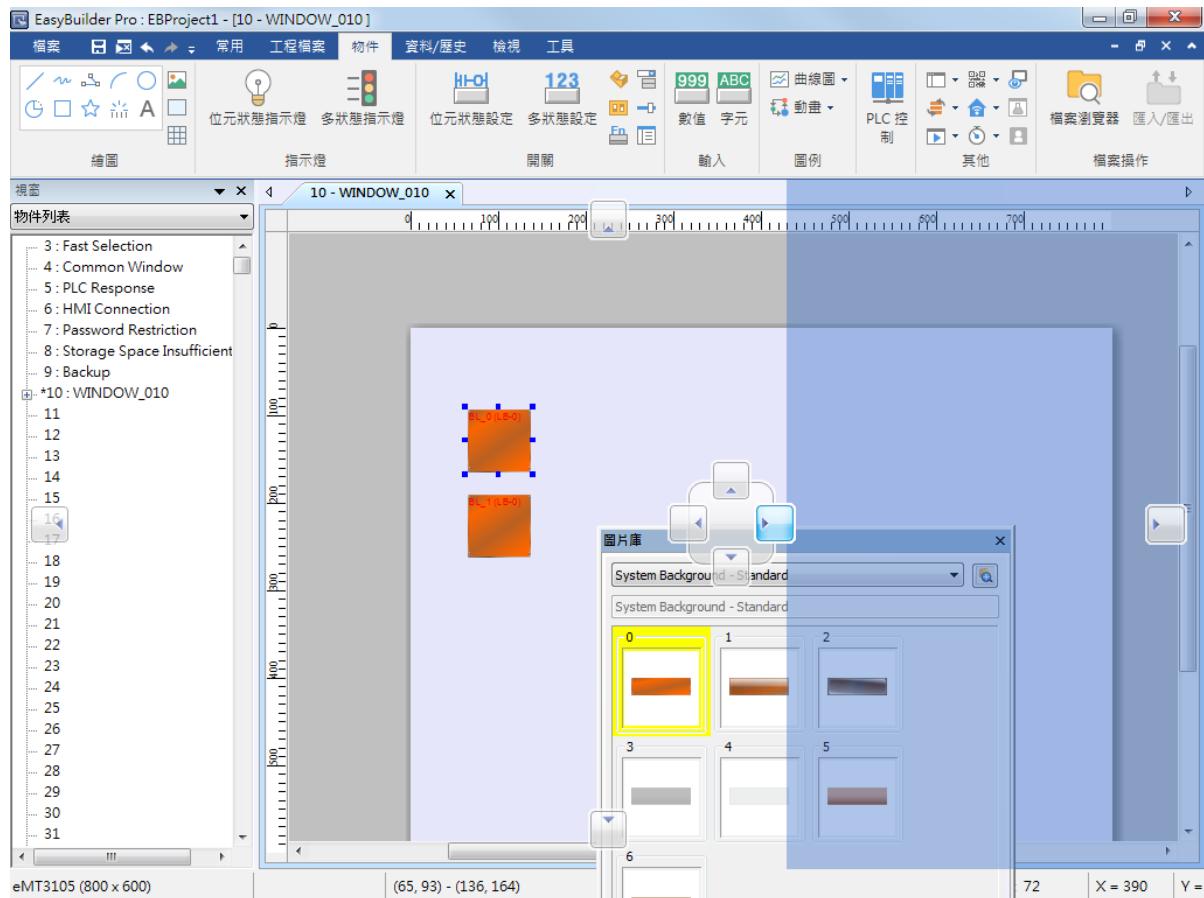
- 即時預覽圖片套用在物件上的結果。步驟如下：

  1. 先點選物件。
  2. 按住鍵盤的 [Shift] 按鍵。
  3. 瀏覽圖片庫管理視窗選擇欲使用的圖片，預覽的圖片會被暫時套用在此物件上。



#### 14.4.4. 視窗調整

使用者可以使用拖曳功能及顯示在編輯畫面上的多個定位點圖示，來放置管理視窗到喜愛的位置。



# 15. 文字標籤庫與多國語言使 用

本章節說明如何建立與使用文字標籤庫。

15.1. 概要 .....	15-2
15.2. 文字標籤庫管理 .....	15-2
15.3. 文字標籤庫的建立 .....	15-3
15.4. 文字標籤庫的使用 .....	15-4
15.5. 語言與字型選擇 .....	15-5
15.6. 多國語言的使用 .....	15-10
15.7. 快速新增到文字標籤庫 .....	15-13

## 15.1. 概要

當需要在工程檔案中使用多國語言時，需先建立文字標籤庫後，再從中選擇需要的標籤。HMI 在運行時會依照語言模式的設定，於工程檔案中顯示所選擇語言模式相對應的文字。EasyBuilder Pro 同時支援 24 種不同語言的文字顯示。本章節將說明如何建立文字標籤庫並使用多國語言。

## 15.2. 文字標籤庫管理

按下工具列上的 [工程檔案] » [文字標籤] 即可進入 [文字標籤庫] 對話窗，如下圖所示。



設定	描述
當前狀態	一個文字標籤最多可擁有 256 個狀態 (0 ~ 255)。 狀態數量將受 [語言數量] 使用的限制，若使用 1 ~ 3 個語言，每個語言最多可有 256 個狀態，若使用 4 個以上的語言數量，則可用 768去除以語言數量後，即可得到最多狀態數。 例如：語言數量為 24，則 $768/24 = 32$ (狀態數)。
新增	新增一筆文字標籤。
設定	設定所選擇的文字標籤內容。
匯出文字庫檔案	儲存所有文字標籤為 .lbl 格式檔案。
匯入文字庫檔案	將現存文字標籤 .lbl 檔案匯入文字標籤庫。
匯出為 EXCEL 檔案	儲存所有文字標籤為 .csv, .xls 或 .xlsx 格式檔案。
匯入 EXCEL 檔案	將現存文字標籤 .csv, .xls 或 .xlsx 檔案匯入文字標籤庫。

**編譯所有項目標籤以供將來反編譯**

開啟此選項將在編譯時將所有標籤加入編譯檔.xob 中，使未來在反編譯時可以完整恢復文字標籤庫。請注意，這可能會顯著增加.xob 檔案大小。

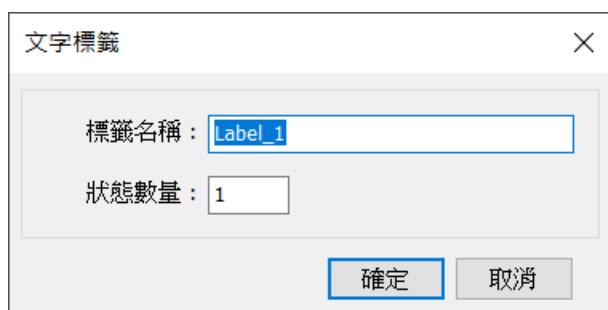
### Note

- 匯入或匯出 Excel 檔案均不支援 Unicode。

## 15.3. 文字標籤庫的建立

請依照以下步驟建立文字標籤庫。

1. 開啟 [文字標籤庫] » [新增]。定義文字標籤的名稱並設定文字標籤所要表現的狀態數。



2. 按下 [確定] 後會顯示一個新的空白標籤，選擇該標籤後按 [設定] 即可編輯標籤內容。



3. 設定相關語言的內容。



#### 15.4. 文字標籤庫的使用

當文字標籤庫存在已定義完成的標籤，在物件的 [標籤] 頁籤中可以勾選 [使用文字標籤庫]，在 [標籤] 的下拉選單會列示出已定義的文字標籤。

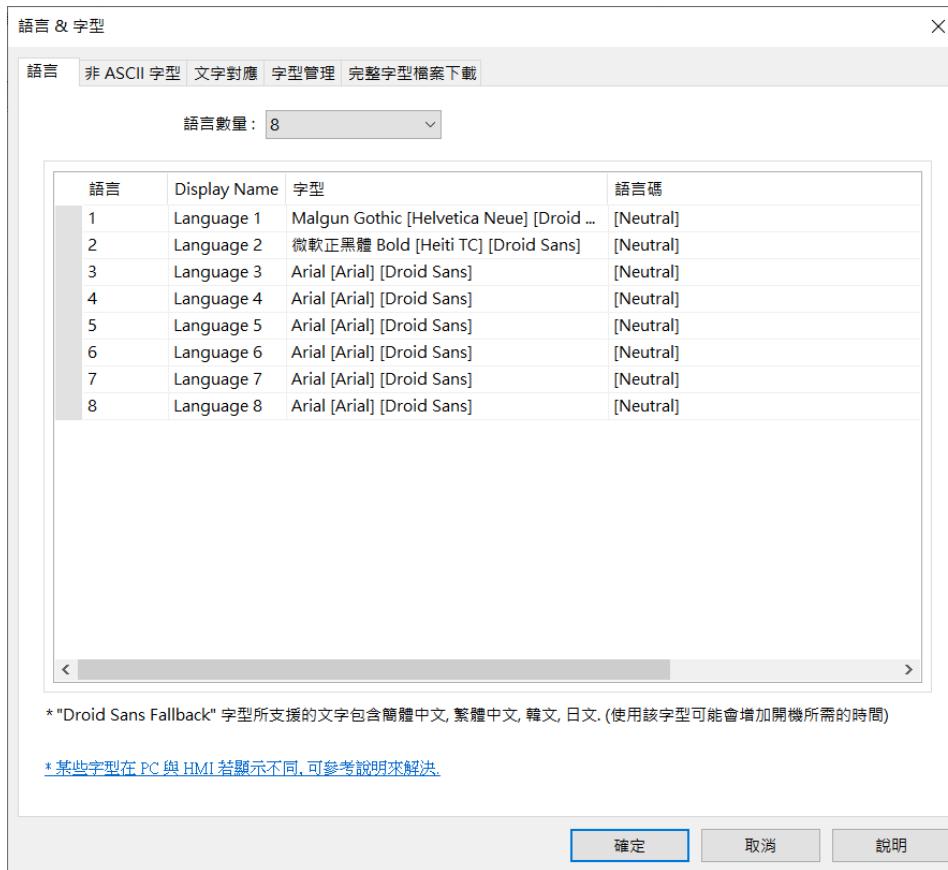


選擇標籤後，可以發現 [內容] 中的文字來源所顯示為文字標籤的內容，所使用的字型為文字標籤庫的設定內容。請注意，語言 2 ~ 24 的文字屬性設定只有文字尺寸是可被單獨設定的，其餘屬性設定，包含文字顏色、對齊和閃爍等，均與語言 1 相同。

## 15.5. 語言與字型選擇

### 15.5.1. 語言

按下工具列上的 [工程檔案] » [語言&字型] 即可進入 [語言&字型] 對話窗，如下圖所示。



設定	描述
語言編號	此處設定檔案的語言個數。
字型	<p>頁籤中可以設定語言顯示的名稱與字型，不同語言可選擇不同的字型，如有需要，亦可為每種字型寫下註釋。[語言碼] 則是用來設定當事件登錄有啟用 [推播通知 (EasyAccess 2.0)] 功能時，於 EasyAccess 2.0 APP 上可以更換事件推播的語言。</p> <p> 請點選此圖示觀看教學影片，請先確定已連上網路線。</p>

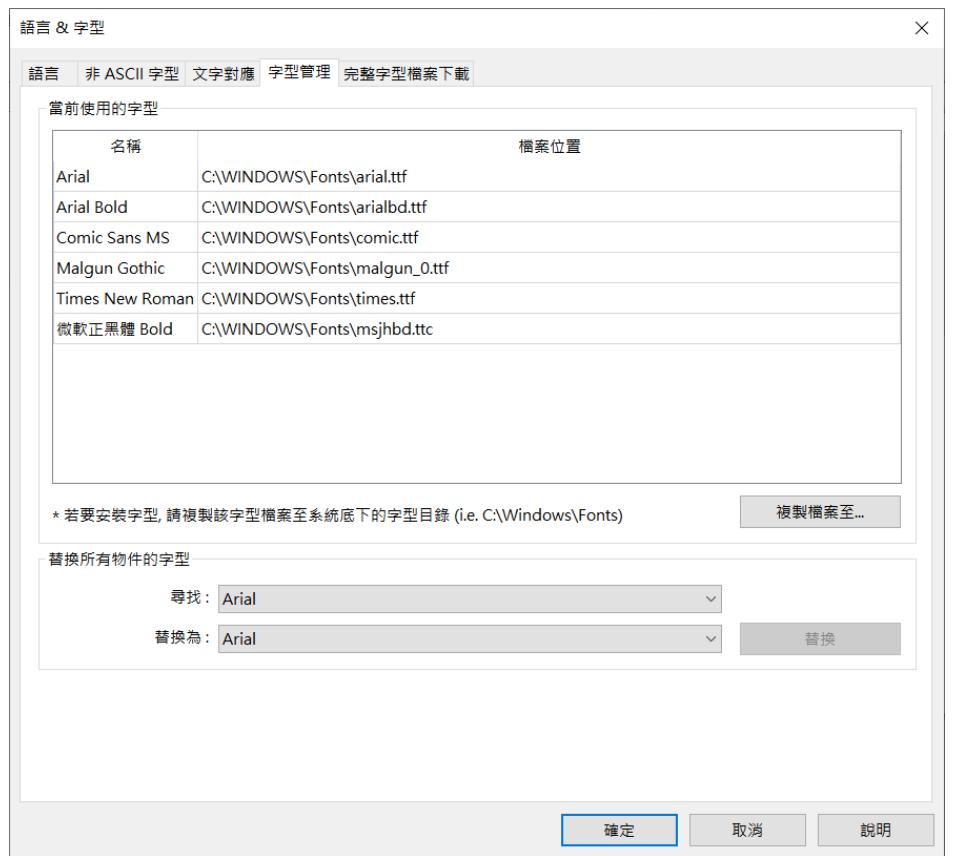
### 15.5.2. 預設字型

[預設字型] 設定頁用來設定字串中不被支援的文字所使用的字型。當使用者使用非所選字體支援的文字 (例如：簡體中文字、繁體中文字、日文、韓文等) 時，EasyBuilder Pro 會自動為這類文字選用項目表中的字型。



設定	描述
<b>字串的預設字型</b>	當使用非所選字型支援的文字時，EasyBuilder Pro 會自動將字型改為 [字串的預設字型]。

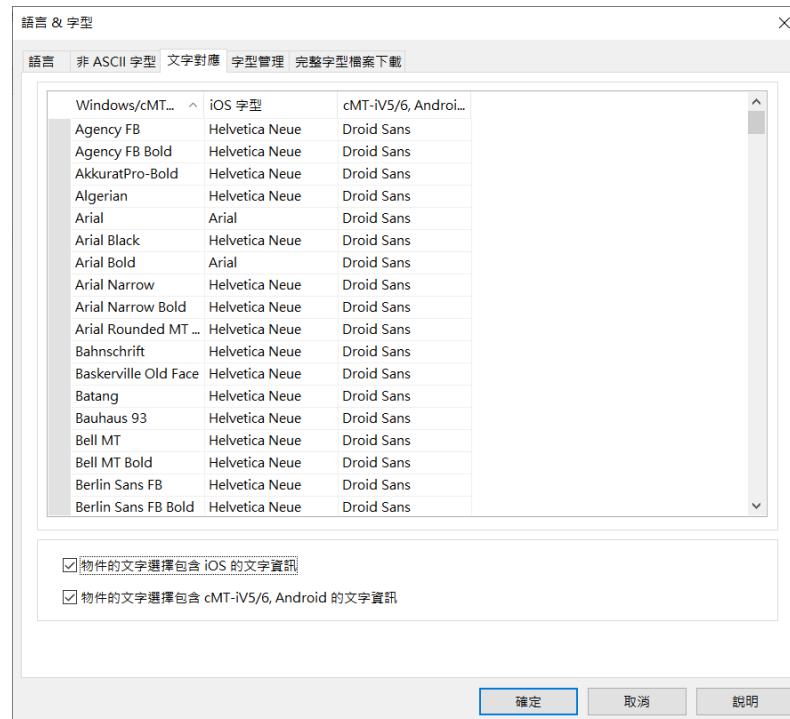
### 15.5.3. 字型管理



設定	描述
當前使用的字型	包含目前檔案中有使用的字型以及檔案所在位置。
複製檔案至	複製字型檔案到選定的路徑。
替換所有物件的字型	可使用此功能將字型一次性變更。

### 15.5.4. 文字對應

當機型為 cMT / cMT X 系列時，[文字對應] 設定頁中將列出在 WINDOWS/cMT3000 中使用的字型，所對應在 iOS / cMT-iV5 / cMT-iV6 / Android 設備上可以顯示的字型。



## 設定 描述

**物件的文字選擇包含 iOS / cMT-iV5 / Android 的文字資訊**

啟用後，物件在挑選字型時，對應的 iOS/cMT-iV5/cMT-iV6/Android 字型會顯示在 [字型] 欄位。



### 15.5.5. 完整字型檔案下載

當 PLC 網頁瀏覽器無法正確顯示網頁上的文字時，EasyBuilder Pro 可以將完整字型檔案下載至 HMI 上，使 PLC 網頁瀏覽器支援此列出的字型。



## 15.6. 多國語言的使用

當物件的文字內容要求表現出多國語言的效果時，除了使用文字標籤外，也需搭配系統暫存器 [LW-9134: 當前使用的語言] 的使用。

[LW-9134] 的有效設定值範圍為 0 ~ 23，不同的數據對應到需顯示的語言。

當編譯下載的檔案沒有勾選全部語言時，[LW-9134] 使用方式將有改變。

例如使用者於文字標籤庫建立 5 種語言：

1: 英文, 2: 繁體中文, 3: 簡體中文, 4: 法文, 5: 韓文

而編譯時只勾選語言 1, 3, 5 則 [LW-9134] 對應數值為：

0: 英文, 1: 簡體中文, 2: 韓文

請依照下列步驟使用多國語言。

1. 先建立一個 [文字/註解] 物件，並勾選 [使用文字標籤庫]。

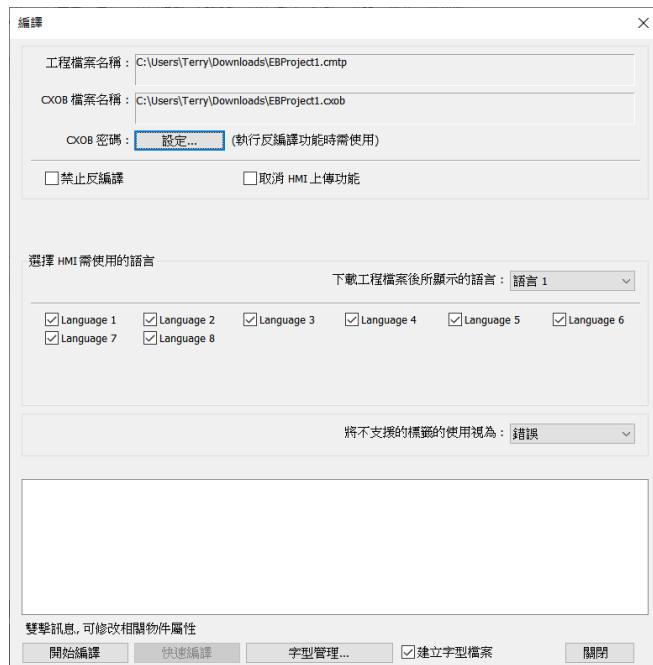
## 文字標籤庫與多國語言使用



2. 再建立一個 [數值物件]，位址設為系統暫存器 [LW-9134]。



3. 編譯時勾選需要並已定義的語言。



4. 模擬如下，當更改 [LW-9134] 的內容時，即可變換文字物件所顯示的內容。

### English

LW9134 : 目前所使用的語言 0

### 简体中文 (SIMPLE)

LW9134 : 目前所使用的語言 2

### 한국어 (KOREAN)

LW9134 : 目前所使用的語言 4

### Note

- 若 HMI 型號選擇 cMT/cMT X 系列時，[LW-9134] 是指伺服器的語言模式，可用來切換 cMT/cMT X 系列上的語言，而 [PLW-9134] 則是切換手持平板裝置上的語言。
- 編譯工程檔案時，只有使用到的文字標籤會被編譯至 \*.cxob/ \*.exob 中。因此透過\*.cxob/ \*.exob 反編譯回的工程檔案，不會有未使用到的文字標籤。

 請點選此圖示下載範例程式。此範例程式說明如何用項目選單物件切換多國語言。

下載範例程式前，請先確定已連上網路線。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 15.7. 快速新增到文字標籤庫

[新增到文字標籤庫] 功能可以將文字以簡便的操作新增至文字標籤庫。

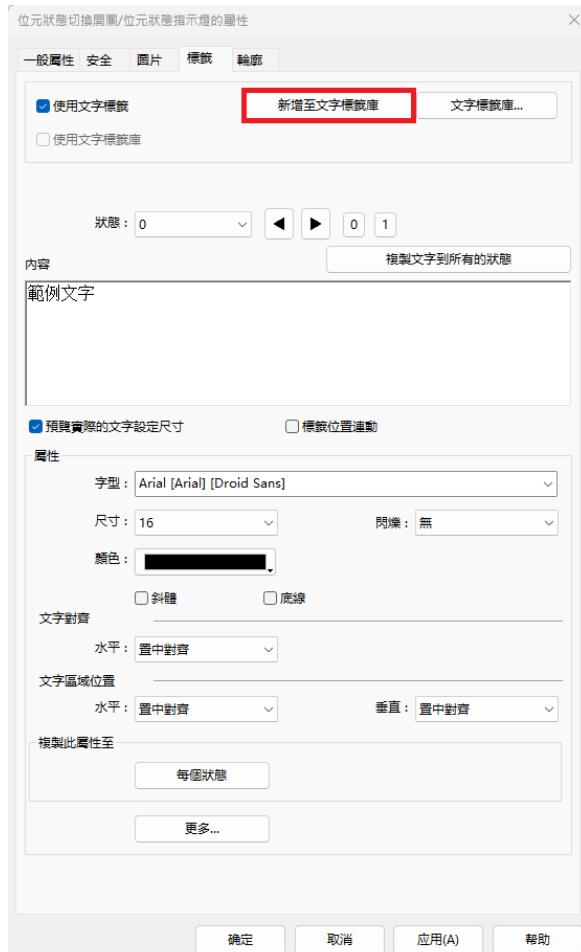
本章節將展示如何將不同的文字快速新增到文字標籤庫。

### 15.7.1. 物件標籤

若要快速將文字標籤加入至文字標籤庫，請依照以下三種方法：

方法一：透過物件屬性設定頁

1. 開啟物件屬性設定頁，並選取標籤頁面。
2. 點擊 [新增至文字標籤庫] 按鈕。



方法二：透過編輯視窗

1. 在編輯視窗上，選擇欲新增至文字標籤庫的一個或多個物件。
2. 點擊滑鼠右鍵，在彈出的選單中，選擇 [新增到文字標籤庫]。



### 方法三：透過 [新增到文字標籤庫] 介面

- 在 EasyBuilder Pro 工作列的工程檔案分頁，找到文字標籤中的 [新增到文字標籤庫] 功能。



- 在 [尋找位置] 中選擇物件所在的頁面，並點擊 [尋找]。

舉例來說：若要將視窗 10 的物件標籤加入文字標籤庫，請選擇視窗 10，並點擊 [尋找]。



3. 操作介面會列出該頁面尚未新增至文字標籤庫的選項，請依需求對個別項目進行 [新增]，或是 [新增全部]。

新增到文字標籤庫			
尋找		新增	新增全部
Location	Name	Type	Text
視窗_10	TS_0	文字	"範例文字" / ""
視窗_10	TS_0	描述 (操作記錄)	描述
視窗_10	BL_0	文字	"Example text" / ""

雙擊選擇以尋找物件.

關閉

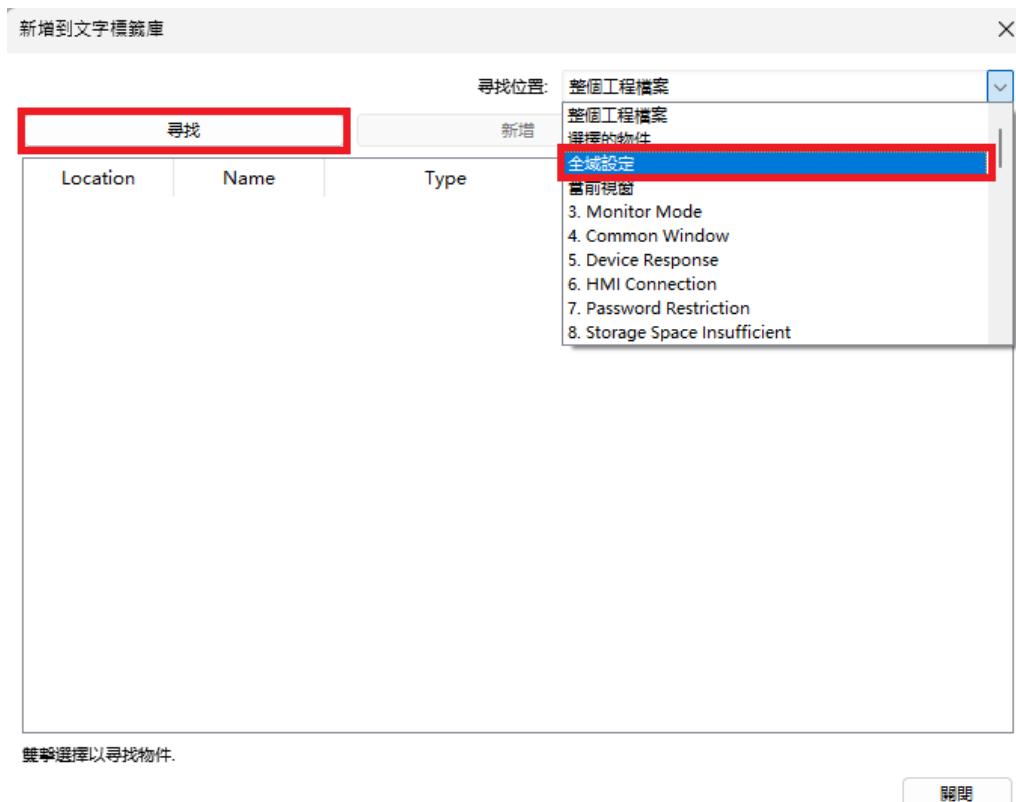
### 15.7.2. 非物件文字

若要快速新增物件標籤以外的文字到文字標籤庫，請依照以下步驟。

- 在 EasyBuilder Pro 工作列的工程檔案分頁，找到文字標籤中的 [新增到文字標籤庫] 功能。



- 在 [尋找位置] 中選擇 [全域設定]，並點擊 [尋找]。



- 點擊 [尋找] 後，操作介面會列出全域設定中可以進行新增的文字。

目前可新增的類別為以下：

- 操作記錄描述
- 事件訊息
- 事件(子)分類自定義名稱
- 事件(子)分類名稱

請依需求對個別項目進行 [新增]，或是 [新增全部]。

The screenshot shows a dialog box titled "新增到文字標籤庫" (Add to Text Label Library). At the top right is a close button (X). Below it is a search bar with the placeholder "尋找位置: 全域設定" (Search location: Global Setting) and a dropdown arrow. The main area contains a table with the following data:

尋找		新增	新增全部
Location	Name	Type	Text
操作記錄	UAC	描述 (操作記錄)	
事件登錄	事件登錄測試 (...)	文字	事件登錄測試
事件登錄	類別	檔頭名稱	自定義類別名稱
事件登錄	類別	名稱	Category 0
事件登錄	類別	名稱	Category 1
事件登錄	類別	名稱	Category 2
事件登錄	類別	名稱	Category 3
事件登錄	類別	名稱	Category 4
事件登錄	類別	名稱	Category 5
事件登錄	類別	名稱	Category 6
事件登錄	類別	名稱	Category 7
事件登錄	類別	名稱	Category 8

下方提示文字：雙擊選擇以尋找物件。

右下角有 "關閉" 按鈕。

# 16.位址標籤庫的建立與使用

本章節說明如何建立與使用位址標籤庫。

16.1.	概要 .....	16-2
16.2.	位址標籤庫的建立 .....	16-2
16.3.	位址標籤庫的使用 .....	16-5

## 16.1. 概要

一般來說，建議使用者在程式設計開始時，先將常用的位址定義在位址標籤庫中，除了可以省去繁複的位址輸入外，也可以增加物件位址資訊的可讀性。

## 16.2. 位址標籤庫的建立

按下工具列上的 [工程檔案] » [位址] 即可進入 [位址標籤庫] 對話窗如下圖所示。



設定	描述
<b>使用者定義標籤</b>	顯示使用者自訂標籤。
<b>系統暫存器</b>	顯示系統保留位址標籤，系統暫存器不能被刪除或修改。
<b>分類</b>	若勾選，可將系統暫存器依功能類型分類，亦可自行編輯 <code>system_tag.xml</code> 檔案設定分類的內容， <code>system_tag.xml</code> 檔案存放於 EasyBuilder Pro 安裝目錄下。
<b>搜尋</b>	輸入關鍵字來搜尋特定標籤。
<b>新增</b>	如何增加一個位址標籤，請見下頁說明。
<b>設定</b>	修改所選位址標籤的內容。
<b>刪除全部</b>	刪除所有位址標籤。
<b>刪除</b>	刪除所選擇的位址標籤。
<b>匯出為 CSV</b>	使用 .CSV 格式將位址標籤庫的所有內容匯出並儲存。
<b>匯入自 CSV</b>	將已存在且為 .CSV 格式的位址標籤檔案匯入到當前的工程檔案。

<b>匯出為 EXCEL</b>	使用 .xls 格式將位址標籤庫的所有內容匯出並儲存。 若是在工程檔案內有被使用到的標籤，匯出後的標籤名稱會在 Excel 中顯示為紅色。
<b>匯入自 EXCEL</b>	將已存在且為 .xls 格式的位址標籤檔案匯入到當前的工程檔案。
<b>使用 UTF-8 格式</b>	若勾選，所匯出的 CSV 格式檔案將為 UTF-8 格式，反之則為 ANSI。
<b>匯出 CSV 檔案</b>	
<b>批次重新命名</b>	可尋找設備內的標籤名稱並批次重新命名。

1. 按下 [新增] 後，即可設定相關屬性。

eMT、iE、XE、mTV 系列



cMT / cMT X 系列



設定	描述
<b>名稱</b>	設定標籤的名稱。
<b>描述</b>	為各個標籤輸入可讀資訊。
<b>設備</b>	可設定的內容來自 [系統參數設定] » [設備] 頁籤。

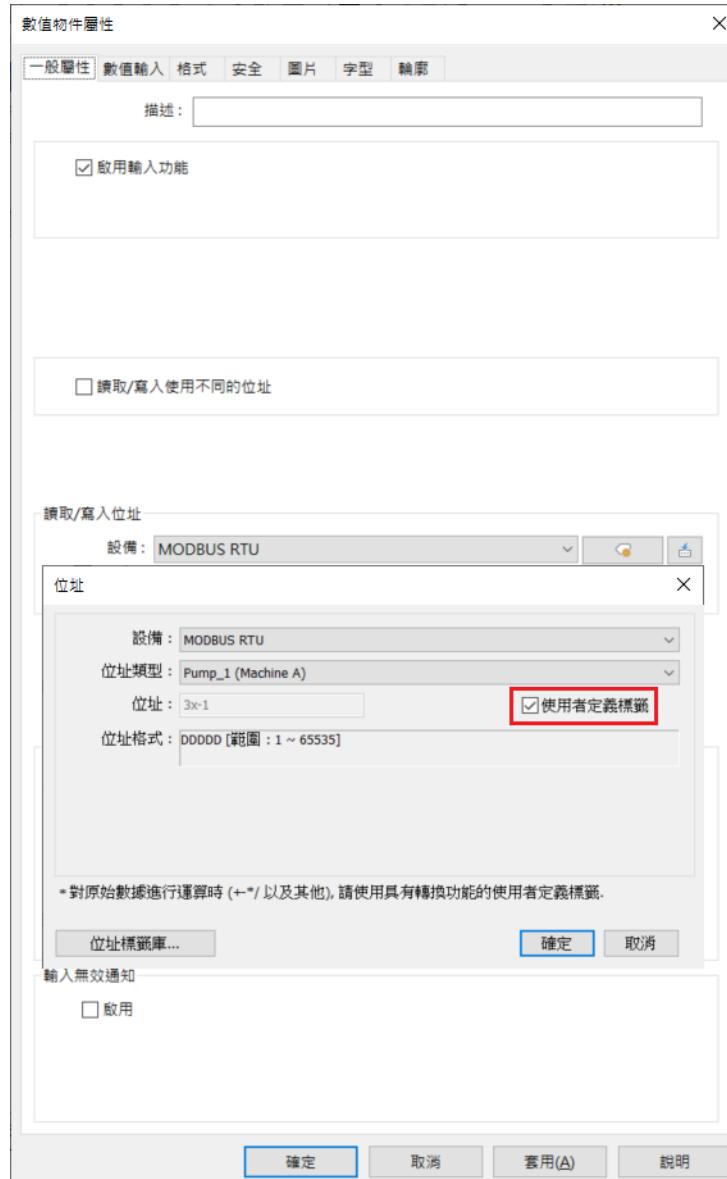
<b>位址模式</b>	可選擇 [位元] 或 [字組] 類型。
<b>位址類型</b>	可選擇類型與 [設備]、[位址模式] 的設定有關。
<b>位址</b>	設定標籤位址。
<b>原始格式</b>	當選擇位址模式為 [字組] 時，可預先定義資料格式的型態。若有預先指定 [原始格式]，當選擇此標籤時，系統將會限制只能使用預設的格式。
<b>轉換/運算 (使用巨集指令副函數)</b>	若啟用，可為位址標籤選擇欲轉換的資料格式，或是選擇巨集指令副函數進行讀取/寫入數據轉換功能。
<b>模式 (僅 cMT/ cMT X 系列支援)</b>	可以選擇巨集副函數或是四則運算模式。
<b>讀取轉換/寫入轉換</b>	<p>選擇讀取/寫入時欲帶入的巨集指令副函式名稱。            資料格式需與副函數的宣告型別相同，才能選擇相對應的副函數。</p> <p>當模式為四則運算時，需要在讀取轉換及寫入轉換分別填上有效的四則運算公式。有效的四則運算公式最少需要包含設備值變數\${v}，設備值變數\${v}在函數中至多出現一次，四則運算公式提供以下運算符號 +、-、*、/、( )。</p>

2. 按下 [確定] 後，可在 [使用者定義標籤] 中發現一筆新增的位址標籤。



### 16.3. 位址標籤庫的使用

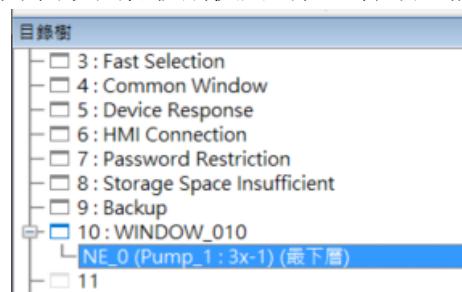
1. 設定位址標籤庫的相關定義。
2. 新增物件，於位址屬性選擇相關的 [設備]。
3. 點選 [設定] 可彈出位址設定視窗。
4. 勾選 [使用者定義標籤]。



5. 在 [位址類型] 選用自訂標籤。
6. 若所定義之字組類型有預先指定資料格式，則系統將會自動限制只能使用選定的格式。

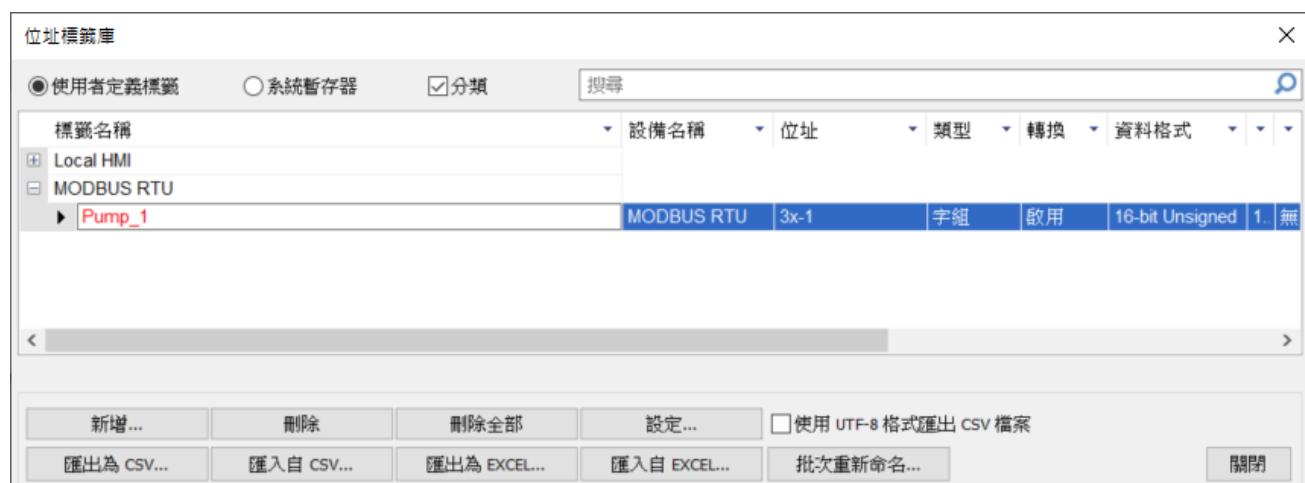


7. 設定完成後，可在物件資訊樹狀圖檢視所使用的位址標籤名稱。



### Note

- 已被使用的位址標籤，其位址標籤名稱於位址標籤庫將被標示為紅色字體。



 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

# 17. 配方資料傳送

本章節說明如何傳送配方資料。

17.1.	概要 .....	17-2
17.2.	使用乙太網路或 USB 線更新配方資料 .....	17-2
17.3.	使用 SD 卡或 USB 碟更新配方資料 .....	17-3
17.4.	配方資料傳輸 .....	17-4
17.5.	配方資料儲存機制 .....	17-5

## 17.1. 概要

所謂配方資料是指存在 **RW** 與 **RW\_A** 位址上的數據，讀寫這些位址的方式與讀寫一般字組位址的方式並無不同，配方資料的特性在於關機後這些數據將保存在 **HMI** 快閃記憶體上，重新開機後 **RW** 與 **RW\_A** 位址上的數據將維持前一次記錄的內容。

**RW** 位址可儲存的配方資料大小為 **512 K words**，而 **RW\_A** 位址則為 **64 K words**，使用者可以透過 SD 卡、USB 碟、USB 線或乙太網路更新配方資料，並利用這些資料更新 **PLC** 上的數據，同樣地也可以上傳配方資料至電腦；此外，使用者亦可以將 **PLC** 上的數據保存在配方資料中。下文將針對配方資料的各種操作做說明。

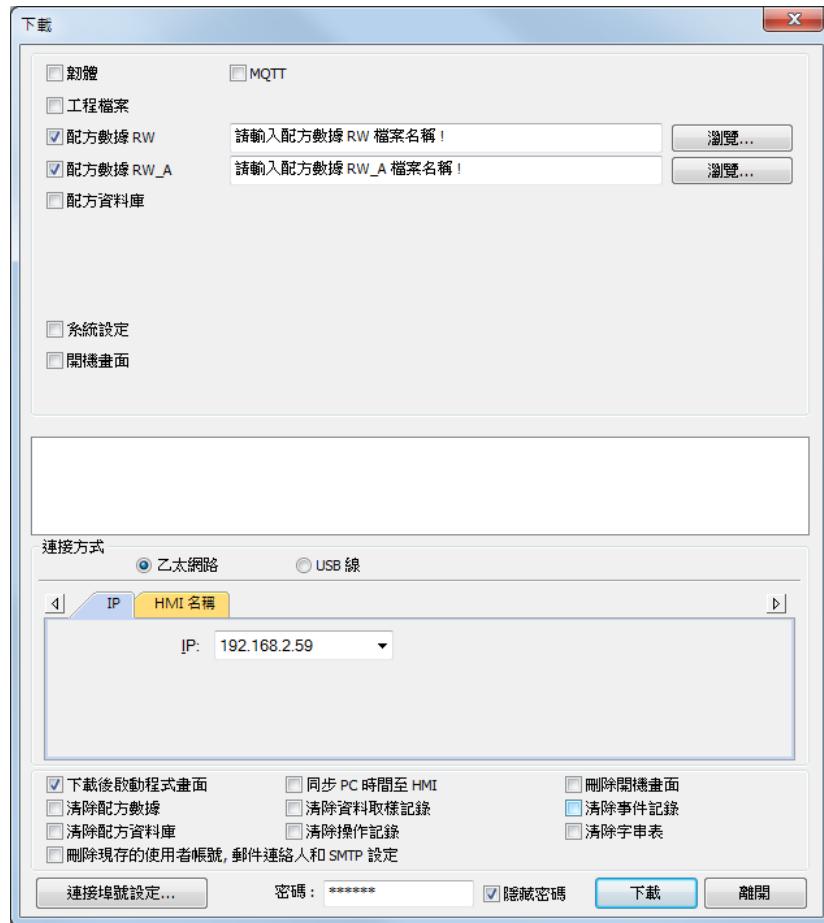
 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 17.2. 使用乙太網路或 USB 線更新配方資料

1. 透過 Utility Manager 選擇 [傳輸] -> [下載] 功能。
2. 勾選 **[RW]** 與 **[RW\_A]** 後選擇欲下載的檔案來源。
3. 下載成功後重新啟動 **HMI**，即可更新 **RW** 與 **RW\_A** 的內容。

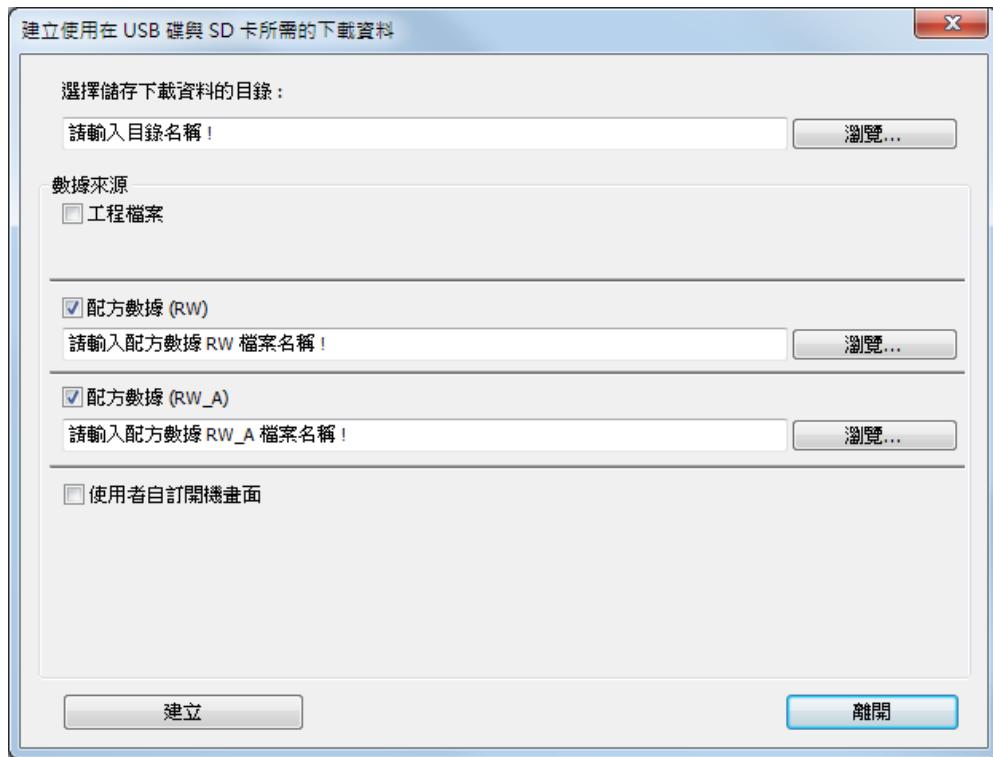
若勾選 **[下載後啟動程式畫面]**，可免去手動重啟 **HMI** 的步驟。

若勾選 **[清除配方數據]**，在進行任何下載動作前，系統會先將 **[RW]** 與 **[RW\_A]** 上的數據內容全部清除。



### 17.3. 使用 SD 卡或 USB 碟更新配方資料

1. 透過 Utility Manager 選擇 [傳輸] -> [建立 SD 卡或 USB 碟的下載資料]。
2. 將 SD 卡或 USB 碟插入電腦。
3. 點選 [瀏覽] 指定檔案資料所要存放的路徑名稱。
4. 點選 [建立]，EasyBuilder Pro 會自動把所要建立的來源資料檔案寫入到 SD 卡或 USB 碟裡。

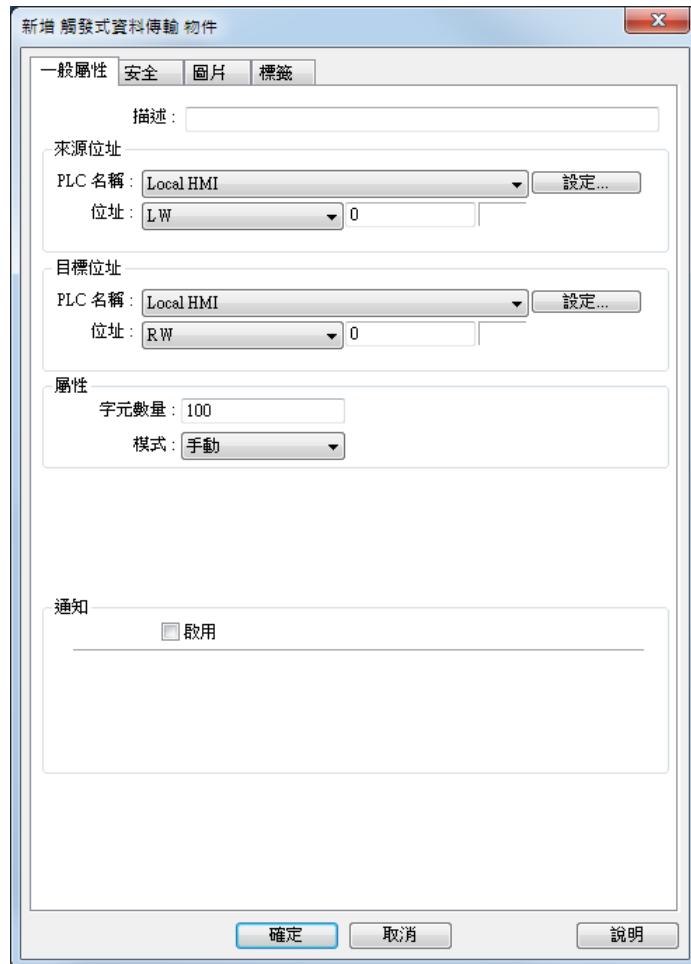


#### Note

- 建立完成後將可發現兩個新資料夾，分別是 history 和 project 資料夾 (資料夾名稱根據機型系列不同，例如：emt3000、mt8000ie、mt8000xe、mtv、mtip 等)。project 資料夾是存放工程檔案的資料夾，而 history 則是存放配方資料及資料取樣/事件記錄的資料夾。
- 需手動重新啟動 HMI 以更新配方資料內容。

## 17.4. 配方資料傳輸

可以藉由 [觸發式資料傳輸] 物件，將配方資料傳送到特定位址；也可以將特定位址的數據保存在 [RW] 與 [RW\_A] 中。



設定	描述
來源位址	設定數據傳送的來源位址。
目標位址	設定數據傳送的目標位址。
屬性	設定自來源位址傳送到目標位址的數據之字組個數。

## 17.5. 配方資料儲存機制

為了延長 HMI 上的快閃記憶體使用壽命，系統以每隔 1 分鐘的時間間隔將配方資料儲存在 HMI 上，且為了避免配方資料在兩次儲存動作間因關機而造成資料的流失，EasyBuilder Pro 提供系統暫存器 [LB-9029: 強迫儲存配方資料到 HMI]，只需對其送出 ON 的訊號，系統即會執行一次配方資料儲存動作。另外如果對 [LB-9028: 重置配方資料] 送出 ON 的訊號，則會將所有的配方數據清除。

# 18. 巨集指令說明

本章介紹巨集指令的語法、編輯、及使用方法。

18.1.	概要 .....	18-2
18.2.	巨集指令編輯器功能使用說明 .....	18-2
18.3.	巨集指令的結構 .....	18-8
18.4.	巨集指令的語法 .....	18-9
18.5.	語句 .....	18-13
18.6.	子函數 .....	18-19
18.7.	內置函數功能 .....	18-21
18.8.	怎樣建立和執行巨集指令 .....	18-95
18.9.	使用者自定義函數功能 .....	18-99
18.10.	使用巨集指令時的注意事項 .....	18-110
18.11.	使用自由協定去控制一個設備 .....	18-111
18.12.	編譯錯誤提示資訊 .....	18-116
18.13.	巨集指令範例程式 .....	18-127
18.14.	巨集指令 TRACE 函數 .....	18-132
18.15.	字串處理函式使用方法 .....	18-136
18.16.	巨集指令密碼保護 .....	18-142
18.17.	巨集支援使用變數讀寫 CANbus 地址 .....	18-144

## 18.1. 概要

巨集指令提供了應用程式之外所需的附加功能。在 HMI 人機界面運行時，巨集指令可以自動的執行這些命令。它可以擔負執行譬如複雜的運算、字串處理，和使用者與工程之間的交流等功能。本章主要介紹巨集指令的語法、如何使用和編輯方法等功能。希望通過本章的說明，能夠使各位能夠快速的掌握 EasyBuilder Pro 軟體提供的強大的巨集指令功能。

## 18.2. 巨集指令編輯器功能使用說明

巨集指令編輯器提供下列新功能：

- 顯示行號
- 復原 (Undo) / 重複 (Redo)
- 剪下 (Cut) / 複製 (Copy) / 貼上 (Paste)
- 全選 (Select All)
- 建立 / 取消書籤 (Toggle Bookmark) / 上一個書籤 (Previous Bookmark) / 下一個書籤 (Next Bookmark) / 清除全部書籤 (Clear All Bookmarks)
- 程式碼摺疊 (Toggle All Outlining)
- 安全 -> 啟用執行條件
- 週期執行
- 當 HMI 啟動時即執行一次

以下將詳細描述如何使用各項功能。

1. 打開巨集指令編輯器，可以看到編輯區左邊將自動顯示行號。



2. 編輯區中按滑鼠右鍵，呼叫出右鍵選單如下圖。目前的狀態無法使用的功能將顯示灰色。例如必須在編輯區中選取一段文字才會開啟複製功能，因此未選取任何文字的狀態下，複製功能暫不開啟。提供快速鍵，如選單內所提示。



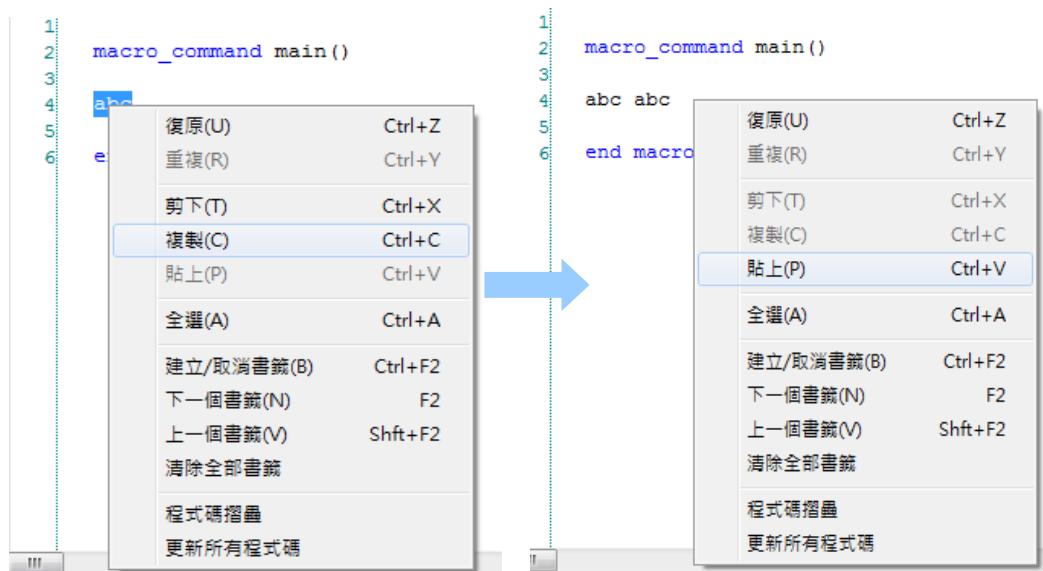
3. 編輯區上方有工具列，提供 [復原]、[重複]、[剪下]、[複製]、[貼上]、[建立/取消書籤]、[下一個書籤]、[上一個書籤]、[清除全部書籤] 等按鈕方便快速選取。



4. 改變編輯區內容將開啟 [復原] 功能，使用者執行復原後可用 [重複] 復原。使用者可從右鍵選單或是利用熱鍵 (Undo : Ctrl+Z, Redo : Ctrl+Y) 執行此功能。



5. 在編輯區選取一段文字後可進行 [剪下]、[複製]，之後可用 [貼上] 將選取的文字貼上。



6. 選擇 [全選] 可選取編輯區全部內容。



7. 當程式碼很長的時候，為方便使用者閱讀，提供了書籤功能。下面說明如何使用此功能。  
 8. 將游標移至編輯區中想要插入書籤的位置，按右鍵，選擇 [建立 / 取取消書籤]。編輯區左邊將會看到一個代表書籤的藍色小方塊。



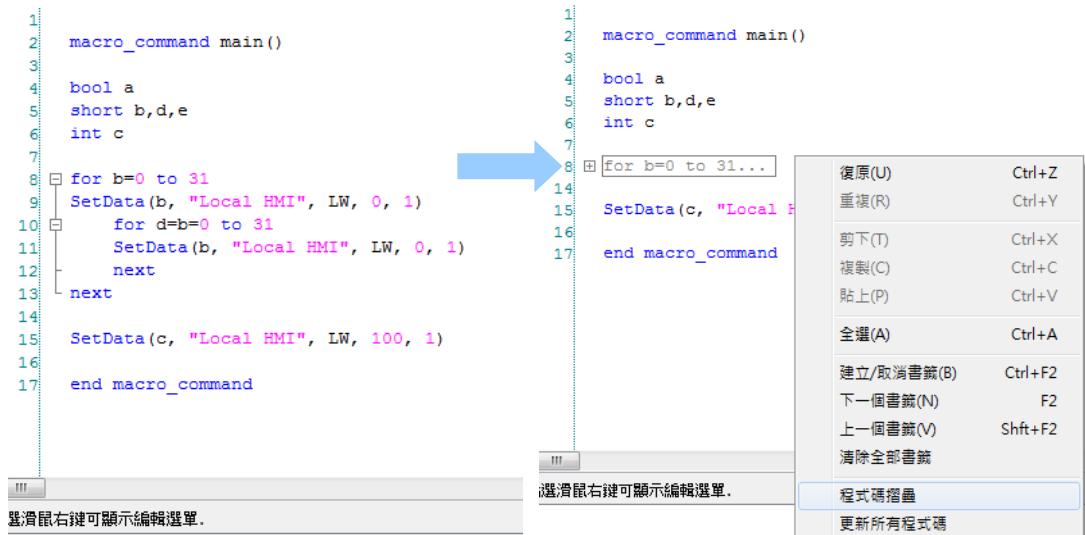
9. 若游標所在位置已存在書籤，選擇 [建立 / 取消書籤] 可將其關閉，反之，選擇 [建立 / 取消書籤] 可將其開啟。
10. 右鍵選擇 [下一個書籤] 游標將會移至下一個書籤所在位置。選擇 [上一個書籤] 游標將會移至上一個書籤所在位置。



11. 選擇 [清除全部書籤] 將關閉所有書籤。
12. 巨集指令編輯器提供程式碼摺疊功能，方便使用者瀏覽程式碼。所謂程式碼摺疊，是指編輯器可將屬於同一區塊的程式碼隱藏起來，被隱藏起來的程式碼在編輯區裡會顯示成。編輯區左側會顯示樹狀圖，使用者可按下 隱藏程式區塊，按下 展開程式區塊。如下圖所示：

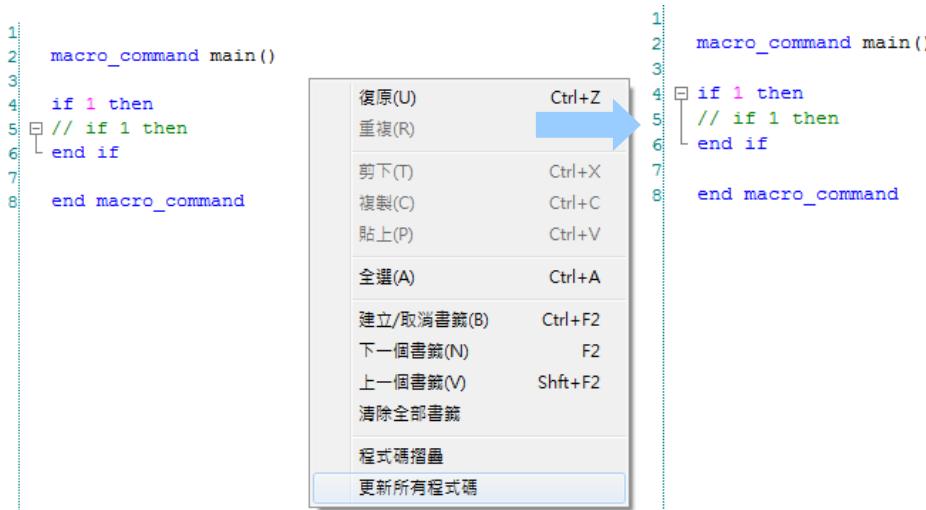


13. 右鍵選擇【程式碼折疊】可展開所有程式碼區塊。



14. 有時候程式碼區塊可能會誤判。這種誤判起因於編輯器沒有辦法區分當前輸入的關鍵字是否存在於注釋中。例如下圖。使用者可以從右鍵選單選擇【更新所有程式碼】來更正這個錯誤。

## 巨集指令說明



**15.** 包圍在特定關鍵字內的程式碼稱為一程式碼區塊。內定的程式碼區塊如下列：

- 子函數： sub – end sub
- 迴圈語句：
  - i. for – next
  - ii. while – wend
- 邏輯運算語句：
  - i. if – end if
- 多重判斷語句： select case – end select

**16.** 巨集指令編輯窗口非壟斷屬性，開啟巨集編輯窗口後，可繼續回到主畫面同時進行編輯，也可直接在巨集視窗中進行連線/離線模擬。



**17.** 巨集編輯器提供尋找 / 替換文字的功能。

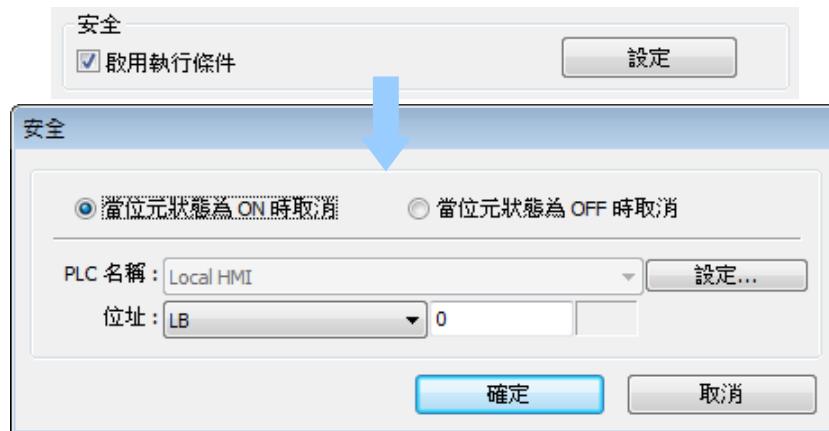


**18.** 使用者勾選 [週期執行] 時，會週期性的觸發此巨集。



**19.** 使用者勾選 [安全] -> [啟用執行條件] -> [設定] 後，可以進行安全設定：

- 當位元狀態 ON 時取消：當位元狀態 ON 時禁止執行此巨集。
- 當位元狀態 OFF 時取消：當位元狀態 OFF 時禁止執行此巨集。



20. 使用者勾選 [當 HMI 啟動時即執行一次] 時，在 HMI 啟動時會自動執行巨集一次。

### 18.3. 巨集指令的結構

巨集指令是由各種語句組成的。這些語句包含常數、變數和各種運算符號。這些語句放置在特定的順序位置以便執行後達到一個希望的執行結果。

巨集指令的結構一般為以下格式：

總體變數聲明-----	可選
Sub Function Block Declarations (子函數聲明)-----	可選
局部變數聲明	
End Sub (結束子函數)	
macro_command main() [主函數]-----	必須
局部變數聲明	
[各式語句]	
end macro_command [結束主函數]-----	必須

一個巨集指令必須有一個且只有一個主函數，用來開始巨集指令的執行。格式為：

**macro\_command** 函數名稱()

**end macro\_command**

變數聲明必須放在巨集指令語句的前面，否則如果語句放置在變數聲明的前面，將會造成巨集指令無法編譯通過。

局部變數一般用在巨集指令主函數或者自定義的子函數中。它的合法性只在指定的函數中有效。總體變數一般是定義在所有巨集指令函數的前面，且它在整個巨集指令中均具有有效性。當局部變數和總體變數被定義為相同的名稱時，只有局部變數有效。

下面就是一個簡單的巨集指令，其中就包含了變數聲明和函數呼叫。雙斜線 “//” 代表程式註解，在它後面的文字不會被執行。

```
macro_command main()
    short pressure = 10                      // 局部變數聲明
    SetData(pressure, "Allen-Bradley DF1", N7, 0, 1) // 函數呼叫
end macro_command
```

## 18.4. 巨集指令的語法

### 18.4.1. 常數和變數

#### 18.4.1.1. 常數

常數是一個可以被各式語句直接使用的固定的資料。有如下格式：

常數類型	使用說明	舉例
十進位整數		345, -234, 0, 23456
十六進位數	必須以 0x 開頭	0x3b, 0xffff, 0x237
字元型	字元必須使用單引號；字串使用雙引號， 雙引號內如果要顯示"請使用\"表示。	'a', "data", "函數名稱"
布林型		true, false

下面即為一個簡單的常數使用的範例。

```
macro_command main()
    short A, B // 聲明 A 和 B 為短整型變數
    A = 1234
    B = 0x12 // 1234 和 0x12 即為常數
end macro_command
```

#### 18.4.1.2. 變數

變數是一個代表著各種資料的名稱。在巨集指令中，這些資料可以隨著巨集指令語句執行的結果改變而改變。

### 變數的命名規則

- 必須以英文字母開頭
- 變數名稱長度不超過 32 個字元
- 系統保留暫存器名稱不能作為變數名稱。

下面為 8 種不同的變數類型，前 5 種為有號數值型態，後 3 種為無號數值型態：

變數類型	描述	範圍
bool 布林型	1 bit (一個位)	0, 1
char 字元型	8 bits (一個位元組)	+127 ~ -128
short 短整型	16 bits (一個字組)	+32767 ~ -32768
int 雙整型	32 bits (雙字組)	+2147483647 ~ -2147483648
float 浮點型	32 bits (雙字組)	
unsigned char 字元型	8 bits (一個位元組)	0 ~ 255
unsigned short 短整型	16 bits (一個字組)	0 ~ 65535
unsigned int 雙整型	32 bits (雙字組)	0 ~ 4,294,967,295
long 長整數	64 bits (四個字組) (只支援於 cMT / cMT X)	+281474976710655 ~ - 281474976710655
unsigned long 長整數	64 bits (四個字組) (只支援於 cMT / cMT X)	0 ~ 281474976710655
double 雙精度浮點數	64 bits (四個字組) (只支援於 cMT / cMT X)	

## 變數聲明

變數必須在使用前聲明。所以，在巨集指令，所有的變數都必須在語句使用前都被聲明完成。聲明變數時，先定義變數的類型，後面再跟著變數名稱。

如下範例:

```
int      a
short   b, switch
float    pressure
unsigned short  c
```

## 陣列聲明

巨集指令支援一維陣列 (下標從 0 開始)。聲明陣列變數時，先定義陣列變數的類型，變數名稱，接著就是該陣列變數的個數，變數個數必須放置在“[]”符號中。陣列變數的長度為 1 ~ 4096。一個巨集指令中最多只支援 4096 個變數。

如下範例:

```
int      a[10]
short   b[20], switch[30]
float    pressure[15]
```

陣列的下標最小為 0，最大下標為(陣列的長度-1)

如下範例:

```
char data[100]           // 陣列變數的長度是 100
所以：最小的陣列為 “data[0]”，最大的陣列為 “data[99]”，即  $100 - 1 = 99$ 。
```

## 變數和陣列初始化

有兩種方法可以讓變數初始化：

- 使用語句中的賦值語句 (=)

如下範例：

```
int a
float b[3]
a = 10
b[0] = 1
● 聲明變數時直接賦值
char a = '5', b = 9
```

陣列變數的聲明是一個特殊的情況。一個完整的陣列被初始化時，可以在陣列變數聲明時，將資料放置在波形括弧“{}”裡面，各資料使用逗號分開。

如下所示：

```
float data[4] = {11, 22, 33, 44} // 這樣 data[0] = 11, data[1] = 22....
```

### 18.4.2. 運算符號

運算符通常被用來指定資料是如何被操作和運算，如下：(在任何一個語句中，運算符左邊的變數結果均依據運算符右邊的條件而獲得。)

運算符號	描述	舉例
=	賦值運算符號	pressure = 10

數學運算符號	描述	舉例
+	加	A = B + C
-	減	A = B - C
*	乘	A = B * C
/	除	A = B / C
% 或 mod	求餘 (返回剩餘數)	A = B % 5 或 A = B mod 5

注意：

由於整數的預設變數類型為整數型，因此在使用除法時，若除數與被除數皆是整數，且計算出的結果含有小數點，則小數點會自動被捨去。若要避免小數點被捨去，在除數或被除數加上.0 即可將數值的變數類型轉為浮點型。

範例：

A = 3 / 2 = 1      » 3 和 2 皆是整數型，因此運算結果也是整數型。

$B = 3 / 2.0 = 1.5 \rightarrow 3$  是整數型， $2.0$  是浮點型，因此運算結果是浮點型。

$C = 3.0 / 2 = 1.5 \rightarrow 3.0$  是浮點型， $2$  是整數型，因此運算結果是浮點型。

比較運算符號	描述	舉例
<	小於	if A < 10 then B = 5
<=	小於或者等於	if A <= 10 then B = 5
>	大於	if A > 10 then B = 5
>=	大於或者等於	if A >= 10 then B = 5
==	等於	if A == 10 then B = 5
<>	不等於	if A <> 10 then B = 5

邏輯運算符號	描述	舉例
And	與	if A < 10 and B > 5 then C = 10
Or	或	if A >= 10 or B > 5 then C = 10
Xor	異或	if A xor 256 then B = 5
Not	非	if not A then B = 5

移位元和位元運算符號通常被用來操作字元型變數、短整型變數和雙整型變數的位元。在一個語句中，這些運算符號的優先權是在從該語句的左邊到右邊依此執行的。即在語句中左邊位置的優先執行，依次從左到右執行。

移位運算符號	描述	舉例
<<	往左移動指定的位數	A = B << 8
>>	往右移動指定的位數	A = B >> 8

位運算符號	描述	舉例
&	位與運算	A = B & 0xf
	位或運算	A = B   C
^	位異或運算	A = B ^ C
~	位取反運算	A = ~B

## 所有運算符號的優先權

上述所有運算符號的優先權從高到低詳細如下所述：

1. 位於圓括號裡面的運算符號最優先
2. 數學運算符號
3. 移位和位元運算符號
4. 比較運算符號
5. 邏輯運算符號
6. 賦值運算符號

## 關鍵字

下面的關鍵字為巨集指令保留使用。這些均不能用來作為變數名稱、陣列名稱或者函數名稱等。

+, -, \*, /, %, >=, >, <=, <, <>, ==, and, or, xor, not, <<, >>, =, &, |, ^, ~  
 exit, macro\_command, for, to, down, step, next, return, bool, short, int, char, float, void, if, then, else,  
 break, continue, set, sub, end, while, wend, true, false  
 SQRT, CUBERT, LOG, LOG10, SIN, COS, TAN, COT, SEC, CSC, ASIN, ACOS, ATAN, BIN2BCD, BCD2BIN,  
 DATE2ASCII, DATE2DEC, DEC2ASCII, FLOAT2ASCII, HEX2ASCII, DOUBLE2ASCII, ASCII2DEC, ASCII2FLOAT,  
 ASCII2HEX, ASCII2DOUBLE, FILL, RAND, DELAY, SWAPB, SWAPW, LOBYTE, HIBYTE, LOWORD, HIWORD,  
 GETBIT, SETBITON, SETBITOFF, INVBIT, ADDSUM, XORSUM, CRC, CRC8, CRC16\_CCITT,  
 CRC16\_CCITT\_FALSE, CRC16\_X25, CRC16\_XMODEM, INPORT, OUTPORT, POW, GetCnvTagArrayIndex ,  
 GetLastError, GetData, GetDataEx, SetData, SetDataEx, SetRTS, GetCTS, Beep, SYNC\_TRIG\_MACRO,  
 ASYNC\_TRIG\_MACRO, TRACE, FindDataSamplingDate, FindDataSamplingIndex, FindEventLogDate,  
 FindEventLogIndex  
 StringGet, StringGetEx, StringSet, StringSetEx, StringCopy, StringMid, StringMD5, StringDecAsc2Bin,  
 StringBin2DecAsc, StringDecAsc2Float, StringFloat2DecAsc, StringHexAsc2Bin, StringBin2HexAsc,  
 StringLength, StringCat, StringCompare, StringCompareNoCase, StringFind, StringReverseFind,  
 StringFindOneOf, StringIncluding, StringExcluding, StringToUpper, StringToLower, StringToReverse,  
 StringTrimLeft, StringTrimRight, StringInsert, String2Unicode, Unicode2Utf8, UnicodeCat,  
 UnicodeCompare, UnicodeCopy, UnicodeExcluding, Utf82Unicode

## 18.5. 語句

### 18.5.1. 定義語句

這個定義語句包含了變數和陣列的聲明。正式的格式如下：

類型	名稱
----	----

定義一個變數的名稱為"名稱"且類型為"類型"。

舉例:

int A //定義了變數 A 為雙整型格式

類型	陣列名稱[陣列長度]
----	------------

定義一個陣列變數為"名稱"，大小為"陣列長度"且類型為"類型"時。

舉例:

int B[10] //定義了一維陣列變數 B 的長度為 10，類型為雙整型

### 18.5.2. 賦值語句

賦值語句使用賦值運算符號將賦值運算符號右邊運算式運算的結果放置到運算符號左邊的變數中。一個運算式是由變數、常數和各種運算符號組成，執行後產生一個新的資料。

類型	運算式
----	-----

舉例：

A = 2 //這樣變數 A 就被賦值為 2

### 18.5.3. 邏輯運算語句

邏輯運算語句是根據邏輯（布林）運算式的結果來執行相應的動作。它的語句如下所示：

單行格式

<b>If</b> <Condition> <b>then</b> [Statements] <b>else</b> [Statements] <b>end if</b>
---------------------------------------------------------------------------------------------------

舉例:

```
if a == 2 then
    b = 1
else
    b = 2
end if
```

## 區塊格式

```
If <Condition> then
    [Statements]
else if <Condition-n> then
    [Statements]
else
    [Statements]
end if
```

舉例:

```
if a == 2 then
    b = 1
else if a == 3 then
    b = 2
else
    b = 3
end if
```

## 語法描述

<b>if</b>	必須用在該語句的開始部分。
<b>&lt;Condition&gt;</b>	必要條件。 這是一個控制語句。當 <Condition> 為 0 時，即為“FALES”，(條件為假)；當 <Condition> 為非 0 時，即為“True”(條件為真)。
<b>then</b>	當 <Condition> 執行為 “TRUE”(真) 時，必須放置在需要執行的語句之前。
<b>[Statements]</b>	在區塊形式中是可選擇的參數，在單行形式中，且沒有 else 子句時，為必要參數，該語句在 <Condition> 為真時執行。
<b>else if</b>	可選，一條或多條語句，在相對應的 <Condition - n> 為 true 時執行。
<b>&lt;Condition-n&gt;</b>	可選，解釋同 Condition
<b>else</b>	可選，在上述 Condition 和 Condition - n 都不為 true 時執行。

**end if** 必須。在一個 if-then 語句中使用這個來結束 if-then 語句。

#### 18.5.4. 多重判斷語句

Select-case 可用來處理多重判斷的敘述，其功能類似 if-else 語句。根據所指定變數的值，分別對應到符合該值的 case，並執行 case 下面的敘述，直到遇到 break 敘述時，才跳到結束符號 end select 處。語法結構如下：

沒有預設 case 的形式：

```
Select Case [variable]  
Case [value]  
[Statements]  
break  
end Select
```

舉例：

```
Select Case A  
Case 1  
b=1  
break  
end Select
```

有預設 case 的形式：

```
Select Case [variable]  
Case [value]  
[Statements]  
break  
Case else  
[Statements]  
break  
end Select
```

舉例：

```
Select Case A  
Case 1  
b=1  
break  
Case else  
b=0
```

```

break
end Select

```

多個不同 case 對應到相同區塊：

```
Select Case [variable]
```

```
Case [value1]
```

```
[Statements]
```

```
Case [value2]
```

```
[Statements]
```

```
break
```

```
end Select
```

舉例：

```
Select Case A
```

```
Case 1
```

```
break
```

```
Case 2
```

```
b=2
```

```
break
```

```
Case 3
```

```
b=3
```

```
break
```

```
end Select
```

語法描述

<b>Select Case</b>	必須用在該語句的開始部分。
<b>[variable]</b>	必要條件。此變數將會與每一個 case 做比較。
<b>Case else</b>	可選。代表預設 case。當 [variable] 的值不符合任何一個 case 時，將會執行此敘述下面的區塊。在沒有預設 case 的情況，當 [variable] 的值不符合任何一個 case 時，將不會做任何動作而直接跳出 select 控制結構。
<b>break</b>	可選。跳到某一個 case 下面執行時，將一句一句執行 case 語句下面的敘述直到遇到 break 命令才結束，並跳到 end select 敘述。當 case 敘述下面沒有任何 break 命令時，流程將不斷往下執行，直到遇到 end select 敘述，才結束並跳出 select 控制結構。
<b>end Select</b>	select-case 語句的結束標誌。

## 18.5. 迴圈語句

迴圈語句依據迴圈條件來反復的執行一個任務。迴圈語句有兩種表達方式。

### 18.5.5.1. for next 語句

For-next 語句通常用來執行次數固定的迴圈任務。一個變數用作為任務執行次數的計數器和結束迴圈任務執行的條件。這個變數為固定執行的次數。語法結構如下：

```
for [Counter] = <StartValue> to <EndValue> [step <StepValue>]  
[Statements]  
next [Counter]
```

或者

```
for [Counter] = <StartValue> down <EndValue> [step <StepValue>]  
[Statements]  
next [Counter]
```

舉例：

```
for a = 0 to 10 step 2  
    b = a  
next a
```

語法描述

<b>for</b>	必須用在該語句的開始部分。
<b>[Counter]</b>	必要，迴圈計數器的數值變數，該變數的結果用來計數迴圈的次數。
<b>&lt;StartValue&gt;</b>	必要，Counter 的初值。
<b>to/down</b>	必要。用來決定步長是遞增還是遞減。 “to” 以 <StepValue> 為步長遞增 <Counter> “down” 以 <StepValue> 為步長遞減 <Counter>
<b>&lt;EndValue&gt;</b>	必要，Counter 的終值、測試點。當 <Counter> 大於該值時，巨集指令將結束這個迴圈任務。
<b>step</b>	可選，指定 <Step Value> 的步長，指定為 1 以外的數值。
<b>[StepValue]</b>	可選，Counter 的步長，只能是數值，如果沒有指定，則預設為 1。
<b>[Statements]</b>	可選，for 和 next 之間的語句區塊，該語句區塊將執行所指定的次數。
<b>next</b>	必須的。
<b>[Counter]</b>	可選。

### 18.5.5.2. while-wend 語句

While-wend 語句是用來執行不確定次數的迴圈任務。設置一個變數用來判斷結束迴圈的條件。當條件為“True”時，該語句將一直迴圈執行直到條件變為“False”。語法結構如下：

```
while <Condition>
    [Statements]
wend
```

舉例:

```
while a < 10
    a = a + 10
wend
```

語法描述

<b>while</b>	必須用在該語句的開始部分。
<b>continue</b>	必要條件。 這是一個控制語句。當為 “True” 時，開始執行迴圈命令，當為 “False” 時，結束執行迴圈命令。
<b>wend</b>	While-wend 語句的結束標誌。

### 18.5.5.3. 其他控制命令

<b>break</b>	用在 for-next 和 while-wend 語句中。當遇到此語句時，立即跳到語句的結束部分。
<b>continue</b>	用在 for-next 和 while-wend 語句中。當遇到此語句時，立即結束當前迴圈命令而開始執行下一個迴圈命令。
<b>return</b>	可用在自訂 function 的回傳值敘述。寫在主函數裡面時，用來強制跳出主函數。

## 18.6. 子函數

使用子函數可以有效的減少迴圈命令的代碼，子函數必須在使用前被定義，且可以使用任何變數和語句類型。在主函數中，將子函數的參數放置在子函數名稱後面的圓括號中，即可調用子函數。子函數被執行後，將執行後的結果返回到主函數需要的賦值語句或者條件中。定義子函數時，不一定要有返回值，且參數部分可以為空。在主函數中調用子函數時，調用方式應符合其定義。語法結構如下：

有返回值的子函數語法：

```
sub type <函數名稱> [(parameters)]
    Local variable declarations
    [Statements]
    [return [value]]
end sub
```

舉例:

```
sub int Add(int x, int y)
    int result
    result = x + y
    return result
end sub

macro_command main()
    int a = 10, b = 20, sum
    sum = Add(a, b)
end macro_command

或:

sub int Add()
    int result, x=10, y=20
    result = x + y
    return result
end sub

macro_command main()
    int sum
    sum = Add()
end macro_command
```

沒有返回值的子函數語法：

```
sub <函數名稱> [(parameters)]
    Local variable declarations
    [Statements]
end sub
```

舉例：

```
sub Add(int x, int y)
    int result
    result = x + y
end sub

macro_command main()
    int a = 10, b = 20
    Add(a, b)
end macro_command
```

或：

```
sub Add()
    int result, x=10, y=20
    result = x +y
end sub

macro_command main()
    Add()
end macro_command
```

語法描述

<b>sub</b>	必須用在該子函數的開始部分。
<b>type</b>	可選。用來定義子函數執行後返回的資料類型。子函數也可以不回傳任何值。
<b>(parameters)</b>	可選。這些參數保留了從主函數傳入的數值。這些被傳入的參數必須使用與在參數變數聲明的類型一致。 舉例： sub int MyFunction(int x, int y). x 和 y 必須為從主函數中傳過來的雙整型資料格式的資料。調用此子函數的語句格式大致為這樣： ret = MyFunction(456, pressure) , 其中 pressure 需為雙整型資料格式方符合子函數參數變數的聲明。 請注意調用語句的參數部分可以是常數也可以是變數。當執行這個子函數後，一個雙整型資料將會返回給變數 “ret”。
<b>Local variable declaration</b>	除了被傳遞的參數之外，子函數中使用的變數必須事先聲明。在上面的“舉例”中，X 和 Y 就是子函數可以使用的變數。總體變數也可以用在子函數中。
<b>[Statements]</b>	需要執行的語句。
<b>[return value]]</b>	可選。用來將執行的結果返回給調用語句。這個結果可以是一個常數或者變數。返回後同時也結束了子函數的執行。子函數也可以不回傳任何值，但是當 type 部分有定義時，則必須加上此 return 敘述。
<b>end sub</b>	必須的。用來結束子函數。

## 18.7. 內置函數功能

EasyBuilder Pro 軟體巨集指令中本身提供了一些內建的函數用來從設備獲取資料和傳輸資料到設備、資料處理和數學運算等。

### 18.7.1. 函數一覽表

點選以下表格中的函數名稱可檢視其相關詳細資訊。

函數名稱	簡述
設備函數	
<a href="#">GetData</a>	讀取設備的資料。
<a href="#">GetDataEx</a>	讀取設備的資料，不等待設備回應，逕自往下執行。
<a href="#">GetError</a>	取得錯誤碼。
<a href="#">SetData</a>	將數據寫到設備中。
<a href="#">SetDataEx</a>	將數據寫到設備中，不等待設備回應，逕自往下執行。
Free Protocol 函數	
<a href="#">GetCTS</a>	偵測 RS-232 之 CTS 訊號。
<a href="#">INPORT</a>	從串列埠/乙太網口讀取數據。
<a href="#">INPORT2</a>	從串列埠/乙太網口讀取數據並在讀取完畢後，等待指定的時間。
<a href="#">INPORT3</a>	從串列埠/乙太網口讀取數據讀取指定的長度。
<a href="#">INPORT4</a>	從串列埠/乙太網口讀取數據，直到讀取到特定結尾符號後即停止讀取。
<a href="#">OUTPORT</a>	將數據從串列埠/乙太網口傳送給設備或者控制器。
<a href="#">PURGE</a>	清空 COM port 的輸出入緩衝區。
<a href="#">SetRTS</a>	拉高或拉低 RS-232 之 RTS 訊號。
巨集控制函數	
<a href="#">ASYNC_TRIG_MACRO</a>	觸發指定巨集指令並同時執行。
<a href="#">SYNC_TRIG_MACRO</a>	觸發指定巨集指令執行，並等待其執行完畢後，才往下執行。
<a href="#">DELAY</a>	讓巨集指令暫停執行指定時間。
資料操作函數	
<a href="#">FILL</a>	將數據放置到陣列中。
<a href="#">SWAPB</a>	將一個 16 bit 字的高低位元組顛倒。
<a href="#">SWAPW</a>	將一個 32 bit 雙整型資料的高位字組和低位字組顛倒。
<a href="#">LOBYTE</a>	讀取一個 16 bit 資料的低位元組。
<a href="#">HIBYTE</a>	讀取一個 16 bit 資料的高位元組。
<a href="#">LOWORD</a>	讀取一個 32 bit 資料的低位字組。
<a href="#">HIWORD</a>	讀取一個 32 bit 資料的高位字組。
<a href="#">INVBIT</a>	將資料的指定位元狀態相反。
<a href="#">SETBITON</a>	將資料的指定位元設置為 1。
<a href="#">SETBITOFF</a>	將資料的指定位元設置為 0。
<a href="#">GETBIT</a>	讀取資料的指定位元的狀態。
資料轉換函數	
<a href="#">ASCII2DEC</a>	將字元型 ASCII 資料轉換為十進位格式的資料。

<a href="#">ASCII2FLOAT</a>	將字元型 ASCII 資料轉換為浮點數格式的資料。
<a href="#">ASCII2HEX</a>	將 ASCII 字元型資料轉換為十六進位的資料。
<a href="#">ASCII2DOUBLE</a>	將 ASCII 字元型資料 (source) 轉換為雙精度浮點數的資料。 此函數僅支援 cMT/cMT X 系列。
<a href="#">BIN2BCD</a>	將 BIN 格式的資料轉換為 BCD 格式的資料。
<a href="#">BCD2BIN</a>	將 BCD 格式的資料轉換為 BIN 格式的資料。
<a href="#">DATE2ASCII</a>	將現在日期資料轉換為 ASCII 格式的資料。
<a href="#">DATE2DEC</a>	將現在日期資料轉換為十進位格式的資料。
<a href="#">DEC2ASCII</a>	將十進位的資料轉換為 ASCII 格式的資料。
<a href="#">FLOAT2ASCII</a>	將浮點數格式資料轉換為 ASCII 格式的資料。
<a href="#">HEX2ASCII</a>	將十六進位格式資料轉換為 ASCII 格式的資料。
<a href="#">DOUBLE2ASCII</a>	雙精度浮點數格式資料 (source) 轉換為 ASCII 格式的資料。 此函數僅支援 cMT/cMT X 系列。
<a href="#">StringDecAsc2Bin</a>	將十進位字串轉換成整數。
<a href="#">StringBin2DecAsc</a>	將整數轉換成十進位字串。
<a href="#">StringDecAsc2Float</a>	將十進位字串轉換成浮點數。
<a href="#">StringFloat2DecAsc</a>	將浮點數轉換成十進位字串。
<a href="#">StringHexAsc2Bin</a>	將十六進位字串轉換成整數。
<a href="#">StringBin2HexAsc</a>	將整數轉換成十六進位字串。
字串處理函數	
<a href="#">String2Unicode</a>	將字串轉換為 Unicode 字串。
<a href="#">StringCat</a>	連接兩字串。
<a href="#">StringCompare</a>	比較兩字串的內容是否相等。此函數將大小寫視為不同。
<a href="#">StringCompareNoCase</a>	比較兩字串的內容是否相等。此函數將大小寫視為相同。
<a href="#">StringCopy</a>	複製字串。
<a href="#">StringExcluding</a>	從某字串第一個字元開始提取不包含特定字元的一段字串。
<a href="#">StringFind</a>	在某個字串中尋找另一個字串。
<a href="#">StringFindOneOf</a>	尋找某字串中任一個字元在另一個字串中第一次出現的位置。
<a href="#">StringGet</a>	讀取設備的字元陣列資料。
<a href="#">StringGetEx</a>	讀取設備的字元陣列資料，不等待設備回應，逕自往下執行。
<a href="#">StringIncluding</a>	從某字串第一個字元開始提取包含特定字元的一段字串。
<a href="#">StringInsert</a>	將字串插入到另一字串中的特定位置。
<a href="#">StringLength</a>	取得字串的長度。

<a href="#">StringMD5</a>	產生一組透過 MD5 訊息摘要演算法的字串。
<a href="#">StringMid</a>	利用此函數可將一個字串中的某一段子字串提取出來。
<a href="#">StringReverseFind</a>	在某個字串中尋找另一個字串，從後面開始找。
<a href="#">StringSet</a>	將字元陣列資料寫到設備中。
<a href="#">StringSetEx</a>	將字元陣列資料寫到設備中，不等待設備回應，逕自往下執行。
<a href="#">StringToUpper</a>	將字串的字元全部轉換成大寫。
<a href="#">StringToLower</a>	將字串的字元全部轉換成小寫。
<a href="#">StringToReverse</a>	將字串反轉。
<a href="#">StringTrimLeft</a>	裁剪字串開頭的特定字元。
<a href="#">StringTrimRight</a>	裁剪字串尾端的特定字元。
<a href="#">Unicode2Utf8</a>	將 Unicode 字串轉換為 UTF8 格式字串。
<a href="#">UnicodeCat</a>	連接兩 Unicode 字串。
<a href="#">UnicodeCompare</a>	比較兩 Unicode 字串的內容是否相等。大小寫視為不同。
<a href="#">UnicodeCopy</a>	複製 Unicode 字串。
<a href="#">UnicodeExcluding</a>	從某 Unicode 字串第一個字元開始提取不包含特定字元的一段字串。
<a href="#">UnicodeLength</a>	取得 Unicode 字串長度。
<a href="#">Utf82Unicode</a>	將 Utf8 格式字串轉換為 Unicode 字串。

## 數學運算函數

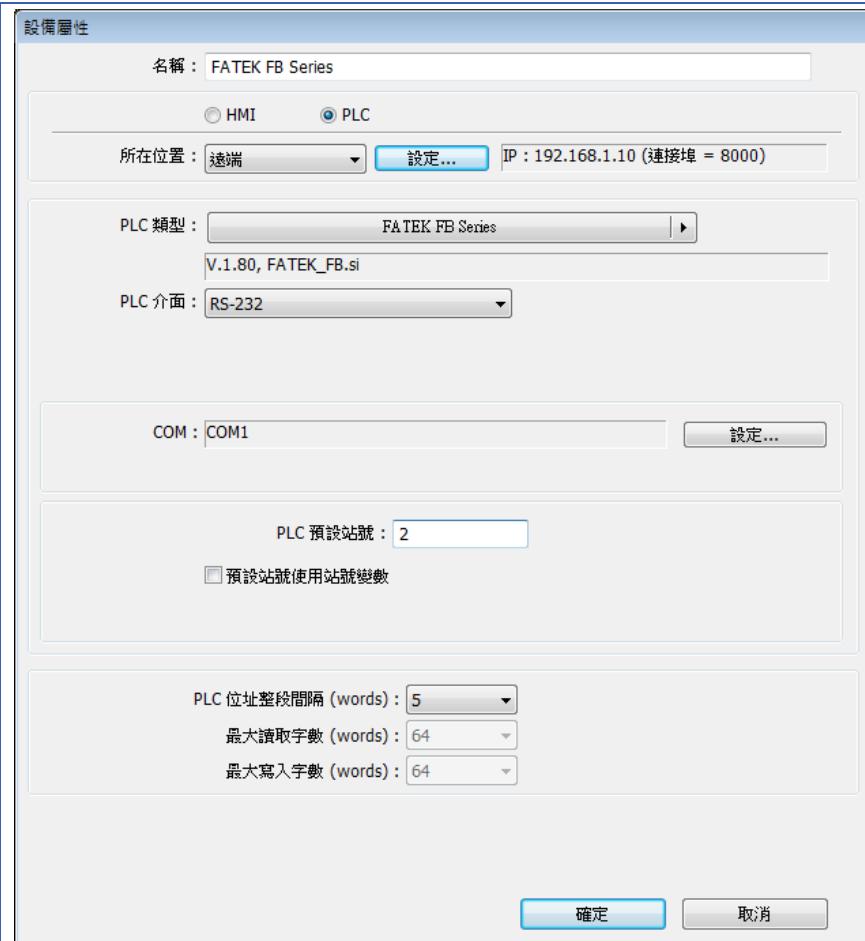
<a href="#">SQRT</a>	開平方根。
<a href="#">CUBERT</a>	開三次方根。
<a href="#">POW</a>	指數計算。
<a href="#">SIN</a>	三角函數的正弦計算。
<a href="#">COS</a>	三角函數的餘弦計算。
<a href="#">TAN</a>	三角函數的正切計算。
<a href="#">COT</a>	三角函數的餘切計算。
<a href="#">SEC</a>	三角函數的正割計算。
<a href="#">CSC</a>	三角函數的餘割計算。
<a href="#">ASIN</a>	反三角函數中反正弦計算。
<a href="#">ACOS</a>	反三角函數中反餘弦計算。
<a href="#">ATAN</a>	反三角函數中反正切計算。
<a href="#">LOG</a>	取得自然對數。
<a href="#">LOG10</a>	取得以 10 為基底的對數。
<a href="#">RAND</a>	產生一個隨機數。
<a href="#">CEIL</a>	向上取得整數。
<a href="#">FLOOR</a>	向下取得整數。

<a href="#">ROUND</a>	取得最接近整數。
統計函數	
<a href="#">AVERAGE</a>	從陣列中計算平均值。
<a href="#">HARMEAN</a>	從陣列中計算調和平均值。
<a href="#">MAX</a>	從陣列中取最大值。
<a href="#">MEDIAN</a>	從陣列中取中位數。
<a href="#">MIN</a>	從陣列中取最小值。
<a href="#">STDEVP</a>	從陣列中計算標準差。
<a href="#">STDEVS</a>	從陣列中計算樣本標準差。
配方資料庫函數	
<a href="#">RecipeGetData</a>	取得配方中的資料。
<a href="#">RecipeQuery</a>	查詢配方中的資料。
<a href="#">RecipeQueryGetData</a>	取得 RecipeQuery 的查詢結果。
<a href="#">RecipeQueryGetRecordID</a>	取得 RecipeQuery 查詢結果的資料列編號。
<a href="#">RecipeSetData</a>	將資料寫入配方資料庫中。
<a href="#">RecipeTransactionBegin</a>	開啟批次寫入配方的功能。需與 RecipeTransactionCommit 或 RecipeTransactionRollback 搭配使用。 此函數僅支援 cMT/cMT X 系列。
<a href="#">RecipeTransactionCommit</a>	執行批次寫入配方。 此函數僅支援 cMT/cMT X 系列。
<a href="#">RecipeTransactionRollback</a>	取消執行批次寫入配方。 此函數僅支援 cMT/cMT X 系列。
資料取樣/事件記錄函數	
<a href="#">FindDataSamplingDate</a>	查詢資料取樣檔的日期。
<a href="#">FindDataSamplingIndex</a>	查詢資料取樣檔的檔案索引值。
<a href="#">FindEventLogDate</a>	查詢事件登錄檔的日期。
<a href="#">FindEventLogIndex</a>	查詢事件登錄檔的檔案索引值。
校驗函數	
<a href="#">ADDSUM</a>	計算一維陣列累加的 checksum。
<a href="#">XORSUM</a>	計算一維陣列異或運算的 checksum。
<a href="#">BCC</a>	等同 XORSUM 函數。
<a href="#">CRC</a>	計算一維陣列的資料進 16-bit CRC 計算，以獲得 checksum (校驗和)。
<a href="#">CRC8</a>	計算一維陣列的資料進 8-bit CRC 計算，以獲得 checksum (校驗和)。
<a href="#">CRC16_CCITT</a>	計算一維陣列的資料進 CRC16_CCITT 計算，以獲得 checksum (校驗和)。
<a href="#">CRC16_CCITT_FALSE</a>	計算一維陣列的資料進 CRC16_CCITT_FALSE 計算，以獲得 checksum (校驗和)。
<a href="#">CRC16_X25</a>	計算一維陣列的資料進 CRC16_X25 計算，以獲得 checksum

	(校驗和)。
<a href="#"><u>CRC16_XMODEM</u></a>	計算一維陣列的資料進 CRC16_XMODEM 計算，以獲得 checksum (校驗和)。
其他函數	
<a href="#"><u>Beep</u></a>	發出系統警笛音。
<a href="#"><u>Buzzer</u></a>	開啟 / 關閉蜂鳴器。
<a href="#"><u>TRACE</u></a>	監視並列印巨集中的變數內容。
<a href="#"><u>GetCnvTagArrayIndex</u></a>	使用者定義標籤使用 [讀取轉換] 並為陣列時，在巨集副函數中，使用此函數可取得陣列的索引。

### 18.7.2. 設備

函數名稱	GetData
語法	<pre>GetData(read_data[start], device_name, device_type, address_offset, data_count) or GetData(read_data, device_name, device_type, address_offset, 1)</pre>
描述	<p>獲取設備的資料，若讀取失敗，後續的程式碼將不會繼續執行。資料是存儲在 <code>read_data[start]~read_data[start+data_count-1]</code> 這些一維陣列變數中。</p> <p><code>data_count</code> 是設定的讀取數據的個數。一般來說，<code>read_data</code> 是一個一維陣列，但是如果 <code>data_count</code> 是 1，<code>read_data</code> 可以是一個一維陣列，也可以是一個普通的變數。下面是兩種從設備中讀取一個字的方法。</p> <pre>macro_command main()     short read_data_1[2], read_data_2     GetData(read_data_1[0], "FATEK FB Series", RT, 5, 1)     GetData(read_data_2, "FATEK FB Series", RT, 5, 1) end macro_command</pre> <p>此處的 <code>device_name</code>，即為在 “系統參數” 中建立設備類型時，設定的“設備名稱”。在此，設備名稱被設定為 “FATEK FB Series”，如下圖所示。</p>  <p><code>device_type</code> 是設備類型和設備中資料的編碼方式。例如：如果 <code>device_type</code> 是 <code>LW_BIN</code>，那麼讀取的設備類型為 <code>LW</code>，資料編碼方式為 <code>BIN</code>。如果使用 <code>BIN</code> 編碼方式，“<code>_BIN</code>”可以忽略。</p> <p>如果 <code>device_type</code> 是 <code>LW_BCD</code>，表示設備類型 <code>LW</code>，資料的編碼方式為 <code>BCD</code> 格式。<code>address_offset</code> 是設備中的地址偏移量。</p> <p>例如，<code>GetData(read_data_1[0], "FATEK FB Series", RT, 5, 1)</code> 代表讀取的設備位址偏移量為 5。</p> <p>如果 <code>address_offset</code> 使用格式為 "<code>N#AAAAA</code>"，<code>N</code> 表示設備的站號，<code>AAAAA</code> 表示位址偏移量。此情況一般使用在同一個串列埠上連接有多台設備或者控制器的情況下。例如：<code>GetData(read_data_1[0], "FATEK FB Series", RT, 2#5, 1)</code> 表示讀取站號為 2 的設備的資料。如果 <code>GetData()</code> 使用 “系統參數 / 設備列表” 中設定的默認的站號，在此可以不填這個站號。</p>



從設備中讀取的資料個數，根據 `read_data` 變數的類型和 `data_count` 的值來決定的。如下表所示：

<code>read_data</code> 的類型	<code>data_count</code> 的值	讀取16位元數據的個數
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bol (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4
float (32-bit)	1	2
float(3-bit)	2	4

當 `Getdata()` 函數讀取 32 位元的資料類型 (int 或者 float 型) 時，此函數會自動的轉換這個資料。例如：

	<pre>macro_command main()     float f     GetData(f, "MODBUS", 6x, 2, 1) // f 中將會是浮點型的數據 end macro_command</pre>
舉例	<pre>macro_command main()     bool a     bool b     bool b_array[30]     char c     char c_array[20]     short s     short s_array[50]     int i     int i_array[10]     float f     float f_array[15]      // 讀取 LB2 的狀態到變數 a 中     GetData(a, "Local HMI", LB, 2, 1)      // 讀取 LB0 ~ LB29 共 30 個狀態，到變數 b_array[0] ~ b_array[29] 中     GetData(b_array[0], "Local HMI", LB, 0, 30)      // 讀取 LW-0 的低位元組到變數 c 中     // 注意char為1 Byte，LW一個定址佔2 Byte(1字組)，讀取字組暫存器的第一個位     // 元組會讀取到該字組的低位元組     // Ex: 假設LW-0的值為0x0201，則c會讀取到0x01     GetData(c, "Local HMI", LW, 0, 1)      // 讀取 LW-1 ~ LW-10的資料到c_array[0] 至 c_array[19] 中     GetData(c_array[0], "Local HMI", LW, 0, 20)      // 讀取 LW-2 的資料到變數 s 中     GetData(s, "Local HMI", LW, 2, 1)      // 讀取 LW-0 ~ LW-49 共 50 個字組到變數 s_array[0] 到 s_array[49] 中     GetData(s_array[0], "Local HMI", LW, 0, 50)      // 讀取兩個字 LW-6 ~ LW-7 到變數 e 中     // Ex: 假設LW-6的值為0x0002, LW-7的值為0x0001，則e會讀取到0x00010002(65538)     // 注意一個 int 資料將佔據 2 個字組(32-bit)     GetData(i, "Local HMI", LW, 6, 1)      // 讀取 LW-0 ~ LW-19 共 20 個字組到變數 i_array[0] ~ i_array[9] 中 (共 10 個     // int 型變數)，陣列 i_array[10] 的變數類型定義為 int。     GetData(i_array[0], "Local HMI", LW, 0, 10)      // 讀取 LW-10 ~ LW-11 到變數 f 中</pre>

```

// 注意此時變數 f 的類型為 float
GetData(f, "Local HMI", LW, 10, 1)

// 讀取 LW-0 ~ LW-29 共 30 個字到變數 f_array[0] ~ f_array[14] 中 (共 15 個
// float 型變數)，陣列 f_array[15] 的變數類型定義為 float。
// 注意一個 float 資料將佔據 2 個字組(32-bit)
GetData(f_array[0], "Local HMI", LW, 0, 15)

end macro_command

```

函數名稱	GetDataEx
語法	<pre>GetDataEx (read_data[start], device_name, device_type, address_offset, data_count) or GetDataEx (read_data, device_name, device_type, address_offset, 1)</pre>
描述	獲取設備的資料，若讀取失敗，仍會繼續執行後續的程式碼。 read_data、device_name、device_type、address_offset 和 data_count 的說明和 GetData 相同。
舉例	<pre>macro_command main() bool a bool b bool b_array[30] char c char c_array[20] short s short s_array[50] int i int i_array[10] float f float f_array[15]  // 讀取 LB2 的狀態到變數 a 中 GetDataEx(a, "Local HMI", LB, 2, 1)  // 讀取 LB0 ~ LB29 共 30 個狀態，到變數 b_array[0] ~ b_array[29] 中 GetDataEx(b_array[0], "Local HMI", LB, 0, 30)  // 讀取 LW-0 的低位元組到變數 c 中 // 注意char為1 Byte，LW一個定址佔2 Byte(1字組)，讀取字組暫存器的第一個位 元組會讀取到該字組的低位元組 // Ex: 假設LW-0的值為0x0201，則c會讀取到0x01 GetDataEx(c, "Local HMI", LW, 0, 1)  // 讀取 LW-1 ~ LW-10的資料到c_array[0] 至 c_array[19] 中 GetDataEx(c_array[0], "Local HMI", LW, 0, 20)  // 讀取 LW-2 的資料到變數 s 中 GetDataEx(s, "Local HMI", LW, 2, 1)</pre>

```

// 讀取 LW-0 ~ LW-49 共 50 個字組到變數 s_array[0] 到 s_array[49] 中
GetDataEx(s_array[0], "Local HMI", LW, 0, 50)

// 讀取兩個字 LW-6 ~ LW-7 到變數 e 中
// Ex: 假設LW-6的值為0x0002, LW-7的值為0x0001, 則會讀取到0x00010002(65538)
// 注意一個 int 資料將佔據 2 個字組(32-bit)
GetDataEx(i, "Local HMI", LW, 6, 1)

// 讀取 LW-0 ~ LW-19 共 20 個字組到變數 i_array[0] ~ i_array[9] 中 (共 10 個
int 型變數), 陣列 i_array[10] 的變數類型定義為 int。
GetDataEx(i_array[0], "Local HMI", LW, 0, 10)

// 讀取 LW-10 ~ LW-11 到變數 f 中
// 注意此時變數 f 的類型為 float
GetDataEx(f, "Local HMI", LW, 10, 1)

// 讀取 LW-0 ~ LW-29 共 30 個字到變數 f_array[0] ~ f_array[14] 中 (共 15 個
float 型變數), 陣列 f_array[15] 的變數類型定義為 float。
// 注意一個 float 資料將佔據 2 個字組(32-bit)
GetDataEx(f_array[0], "Local HMI", LW, 0, 15)

end macro_command

```

函數名稱	GetError
語法	GetError(err)
描述	取得錯誤碼。
舉例	<pre> macro_command main() short err char byData[10]  GetDataEx(byData[0], "MODBUS RTU", 4x, 1, 10) // 讀取 10 byte資料  // 當錯誤碼 (err) 為 0，表示 GetDataEx 成功執行 GetErr(err) // 讀取錯誤碼，存入變數err  end macro_command  錯誤碼: 0: 執行正常 1: GetDataEx 錯誤 2: SetDataEx 錯誤 </pre>

函數名稱	SetData
語法	<pre> SetData(send_data[start], device_name, device_type, address_offset,        data_count) or SetData(send_data, device_name, device_type, address_offset, 1) </pre>

**描述**

將數據寫到設備中，若寫入失敗，後續的程式碼將不會繼續執行。資料保存在 send\_data[start]~send\_data[start+data\_count-1] 中。  
data\_count 是寫入到設備中資料的個數。一般來說，send\_data 是一個陣列。但是如果 data\_count 是 1，send\_data 可以是一個陣列也可以是一個普通的變數。下面是寫一個資料到設備中的方法。

```
macro_command main()
    short send_data_1[2] = { 5, 6}, send_data_2 = 5
    SetData(send_data_1[0], "FATEK FB Series", RT, 5, 1)
    SetData(send_data_2,      "FATEK FB Series", RT, 5, 1)
end macro_command
```

device\_name 詳見上面的說明，在此不在說明。

device\_type 是設備類型和設備中資料的編碼方式。例如：如果 device\_type 是 LW\_BIN，那麼讀取的設備類型為 LW，資料編碼方式為 BIN。如果使用 BIN 編碼方式，“\_BIN”可以忽略。

如果 device\_type 是 LW\_BCD，表示設備類型 LW，資料的編碼方式為 BCD 格式。

address\_offset 是設備中的地址偏移量。

例如，SetData (read\_data\_1[0], "FATEK FB Series", RT, 5, 1) 代表讀取的設備位址偏移量為 5。

如果 address\_offset 使用格式為 "N#AAAAA"，N 表示設備的站號，AAAAA 表示位址偏移量。此情況一般使用在同一個串列埠上連接有多台設備或者控制器的情況下。例如：SetData(send\_data\_1[0], "FATEK FB Series", RT, 2#5, 1) 表示設定站號為 2 的設備的資料。如果 SetData() 使用“系統參數 / 設備列表”中設定的默認的站號，在此可以不填這個站號。

設定到設備的資料個數，根據 send\_data 變數的類型和 data\_count 的值來決定的。如下表所示：

send_data 的類型	data_count 的值	設定 16 位元數據的個數
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)		1
bol (8-bit)	2	
short 16-bit)	1	1
shrt (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4
float (32-bit)	1	2
float (32-bit)	2	4

當 Setdata() 函數寫入 32 位元的資料類型 (int 或者 float 型) 到設備時，此函數會自動的轉換這個資料。例如：

	<pre> macro_command main() float f = 2.6 SetData(f, "MODBUS", 6x, 2, 1) // 在此將會設定一個浮點數到設備中 end macro_command </pre>
舉例	<pre> macro_command main() int i bool a = true bool b[30] short c = false short d[50] int e = 5 int f[10]  for i = 0 to 29 b[i] = true next i  for i = 0 to 49 d[i] = i * 2 next i  for i = 0 to 9 f [i] = i * 3 next i  // 變數 a 的數值設定到 LB2 中 SetData(a, "Local HMI", LB, 2, 1)  // 設定 LB0 ~ LB29 共 30 個位的狀態 SetData(b[0], "Local HMI", LB, 0, 30)  // 將變數 c 的值設定到 LW-2 中 SetData(c, "Local HMI", LW, 2, 1)  // 設定 LW-0 ~ LW-49 共 50 個數據 SetData(d[0], "Local HMI", LW, 0, 50)  // 將變數 e 的值寫入到 LW-6 ~ LW-7 兩個暫存器中，注意變數 e 的類型為int。 SetData(e, "Local HMI", LW, 6, 1)  // 設定 LW-0 ~ LW-19 共 20 個字的數據 // 10 個雙整型數據等於 20 個 16 位元整型數 (1個字組)，因一個 int 資料將佔據 2 個字組 SetData(f[0], "Local HMI", LW, 0, 10) </pre>

	end macro_command
函數名稱	SetDataEx
語法	SetDataEx (send_data[start], device_name, device_type, address_offset, data_count) or SetDataEx (send_data, device_name, device_type, address_offset, 1)
描述	將數據寫到設備中，若寫入失敗，仍會繼續執行後續的程式碼。 send_data、device_name、device_type、address_offset和data_count的說明和 SetData 相同。
舉例	<pre>macro_command main() int i bool a = true bool b[30] short c = false short d[50] int e = 5 int f[10]  for i = 0 to 29 b[i] = true next i  for i = 0 to 49 d[i] = i * 2 next i  for i = 0 to 9 f [i] = i * 3 next i  // 將變數 a 的數值設定到 LB2 中 SetDataEx(a, "Local HMI", LB, 2, 1)  // 設定 LB0 ~ LB29 共 30 個位的狀態 SetDataEx (b[0], "Local HMI", LB, 0, 30)  // 將變數 c 的值設定到 LW-2 中 SetDataEx (c, "Local HMI", LW, 2, 1)  // 設定 LW-0 ~ LW-49 共 50 個數據 SetDataEx (d[0], "Local HMI", LW, 0, 50)  // 將變數 e 的值寫入到 LW-6 ~ LW-7 兩個暫存器中，注意變數 e 的類型為int。 SetDataEx (e, "Local HMI", LW, 6, 1)</pre>

	<pre>// 設定 LW-0 ~ LW-19 共 20 個字的數據 // 10 個雙整型數據等於 20 個 16 位元整型數 (1個字組)，因一個 int 資料 將佔據 2 個字組 SetDataEx (f[0], "Local HMI", LW, 0, 10)  end macro_command</pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------

### 18.7.3. Free Protocol

函數名稱	GetCTS
語法	GetCTS(com_port, result)
描述	<p>偵測 RS-232 之 CTS 訊號。</p> <p>com_port 表示序列埠編號，支援 COM1。此參數可以為變數或常數。result 參數必須為變數。</p> <p>此函數可偵測 CTS 電位值，當 CTS 在高電位時，result 將寫入 1，否則寫入 0。</p>
舉例	<pre>macro_command main() char com_port=1 char result  GetCTS(com_port, result) // 偵測 COM1 之 CTS 電位值  GetCTS (1, result) // 偵測 COM1 之 CTS 電位值  end macro_command</pre>

函數名稱	INPORT
語法	INPORT(read_data[start], device_name, read_count, return_value)
描述	<p>從串列埠或者乙太網口讀數據到人機界面上。這些資料保存在read-data[start]~read-data[start+read-count-1] 這個一維陣列中。</p> <p>device_name是在“設備列表”中定義的“設備名稱”，而這個 device 必須選擇為“Free Protocol”這個設備類型。</p> <p>read-count 是設定的需要讀取的命令的位元組長度，它可以是一個常數或變數。如果這個函數能夠成功的從設備或者控制器中讀取到資料，則return_value將回傳讀取到的word長度。</p>
舉例	<p>下面就是一個使用 INPORT 函數讀取一個 MODBUS 設備保持暫存器資料的範例。</p> <pre>// 讀取保持暫存器資料 macro_command main() char command[32], response[32] short address, checksum short read_no, return_value, read_data[2]  FILL(command[0], 0, 32) // 命令初始化</pre>

```

FILL(response[0], 0, 32)

command[0] = 0x1    // 站號
command[1] = 0x3    // 功能碼：讀取保持暫存器

address = 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

read_no = 2    // read 2 words (4x_1 and 4x_2)
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])

CRC(command[0], checksum, 6)

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

// 使用 OUTPORT 函數將命令送出去
OUTPORT(command[0], "MODBUS RTU Device", 8)

// 使用 INPORT 函數讀取返回的命令
INPORT(response[0], "MODBUS RTU Device", 9, return_value)

if return_value > 0 then
    read_data[0] = response[4] + (response[3] << 8)    // data in 4x_1
    read_data[1] = response[6] + (response[5] << 8)    // data in 4x_2

    SetData(read_data[0], "Local HMI", LW, 100, 2)
end if

end macro_command

```

函數名稱	INPORT2
語法	INPORT2(response[start], device_name, receive_len, wait_time)
描述	從串列埠或者乙太網口讀數據到人機界面上。這些資料保存在 response 這個一維陣列中。device_name是在“設備列表”中定義的“設備名稱”，而這個device 必須選擇為“Free Protocol”這個設備類型。 receive_len 存放所接收到的資料位元組長度，必須為變數。receive_len 的最大長度將不會超過 response 陣列宣告的大小。wait_time 表示等待時間（單位是 millisecond），可以是一個常數或變數。當資料讀取完畢後，若在指定的等待時間內未再收到任何資料，此函數便結束執行並回傳結果。
舉例	<pre> macro_command main()  short wResponse[6], receive_len, wait_time=20  INPORT2(wResponse[0], "Free Protocol", receive_len, wait_time) </pre>

	<pre>// wait_time 單位 : millisecond  if receive_len &gt; 0 then     SetData(wResponse[0], "Local HMI", LW, 0, 6)     // 把回應的資料寫入LW0 end if  end macro_command</pre>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	INPUT3
語法	INPUT3(response[start], device_name, read_count, receive_len)
描述	從串列埠或者乙太網口讀數據到人機界面上。這些資料保存在 response 這個一維陣列中，讀取數據時會讀取指定的數量，未讀取的數據會保留於HMI記憶體緩衝區中，留待下次讀取，避免數據遺失。device_name是在“設備列表”中定義的“設備名稱”，而這個 device 必須選擇為“Free Protocol”這個設備類型。read_count是每次讀取的數據的長度。receive_len 存放所接收到的資料位元組長度，必須為變數。receive_len 的最大長度將不會超過 response 陣列宣告的大小。
舉例	<pre>macro_command main()  short wResponse[6], receive_len  INPUT3(wResponse[0], "Free Protocol", 6, receive_len) // 讀取 6 words  if receive_len &gt;= 6 then      SetData(wResponse[0], "Local HMI", LW, 0, 6) // 把回應的資料寫入LW0  end if  end macro_command</pre>

函數名稱	INPUT4
語法	INPUT4(response[start], device_name, receive_len, tail_ascii)
描述	從串列埠或者乙太網口讀數據到人機界面上。這些資料保存在 response 這個一維陣列中。tail_ascii 是結尾符號。讀取數據時，只要讀取到結尾符號後即停止讀取。device_name是在“設備列表”中定義的“設備名稱”，而這個 device 必須選擇為“Free Protocol”這個設備類型。receive_len 存放所接收到的資料位元組長度，必須為變數。receive_len 的最大長度將不會超過 response 陣列宣告的大小。
舉例	<pre>macro_command main()  char tail_ascii = 0x03// == ETX  short wResponse[1024], receive_len</pre>

	<pre> IMPORT4(wResponse[0], "Free Protocol", receive_len, 0x0d)// 0x0d == CR  IMPORT4(wResponse[0], "Free Protocol", receive_len, tail_ascii)  if receive_len &gt;= 6 then      SetData(wResponse[0], "Local HMI", LW, 0, 6)// set responses to LW0  end if  end macro_command </pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	OUTPORT
語法	OUTPORT(source[start], device_name, data_count)
描述	<p>將放置在從 source[start] 到 source[start+data_count-1] 的所有資料通過串列埠或者乙太網口傳送給設備或者控制器中。</p> <p>device_name 是在“設備列表”中定義的“設備名稱”，而這個 device 必須選擇為“Free Protocol”這個設備類型。</p> <p>Data_count 是發送資料的個數，可以是常數或變數。</p>
舉例	<p>要使用 OUTPORT 函數，必須要在設備類型中選擇“Free Protocol”，如下圖所示。</p>  <p>這裡的 device_name 即為“MODBUD RTU Device”。埠的屬性也是依據在這個“系統參數”中的設定，譬如，在此設定為 (9600, E, 8, 1…)</p> <p>下面是一個範例程式，使用自由協定，以 MODBUS RTU 的協定格式，將單個暫存器設置為 ON。</p> <pre> macro_command main()  char command[32] short address, checksum  FILL(command[0], 0, 32) // 初始化命令  command[0] = 0x1 // 站號 command[1] = 0x5 // 功能碼：寫單個位  address = 0 HIBYTE(address, command[2]) </pre>

	<pre> LOBYTE(address, command[3])  command[4] = 0xff // 使該bit設置為ON command[5] = 0  CRC(command[0], checksum, 6)  LOBYTE(checksum, command[6]) HIBYTE(checksum, command[7])  // 將命令通過串列埠送出去 OUTPORT(command[0], "MODBUS RTU Device", 8)  end macro_command </pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	PURGE
語法	PURGE (com_port)
描述	com_port 表示序列埠編號，支援 COM1 ~ COM3。此參數可以為變數或常數。 可利用這個指令來清空 COM port 的輸出入緩衝區。
舉例	<pre> macro_command main() int com_port=3 PURGE (com_port) PURGE (1) end macro_command </pre>

函數名稱	SetRTS
語法	SetRTS(com_port, source)
描述	拉高或拉低 RS-232 之 RTS 訊號。 com_port 表示序列埠編號，支援 COM1。此參數可以為變數或常數。source 參數也可以為變數或常數。 當傳入的 source 參數值大於 0 時，將拉高 RTS 電位，當 source 參數值等於 0 時，將拉低 RTS 電位。
舉例	<pre> macro_command main() char com_port=1 char value=1  SetRTS(com_port, value) // value &gt; 0，拉高 COM1 之 RTS 電位  SetRTS(1, 0) // 拉低 COM1 之 RTS 電位  end macro_command </pre>

#### 18.7.4. 巨集控制

函數名稱	ASYNC_TRIG_MACRO
語法	ASYNC_TRIG_MACRO(macro_id or name)
描述	<p>一個執行中的巨集指令可以使用此函數利用非同步的方式觸發執行其他的巨集指令。可使用其他巨集指令的 <code>macro_id</code> 或名稱作為參數，指定被觸發的巨集指令 [編號] 或 [名稱]。</p> <p>巨集指令在使用此函數後將立即執行後續的命令，不需等待被觸發的巨集指令執行完成。</p> <p><code>macro_id</code> 可以是常數或使用變數來表示。</p>
舉例	<pre>macro_command main() char ON = 1, OFF = 0  SetData(ON, "Local HMI", LB, 0, 1)  ASYNC_TRIG_MACRO(5)    // call a macro (its ID is 5)  ASYNC_TRIG_MACRO("macro_1")    // call a macro (its name is macro_1)  SetData(OFF, "Local HMI", LB, 0, 1)  end macro_command</pre>

函數名稱	DELAY
語法	DELAY(time)
描述	讓巨集指令暫停執行，持續的時間至少是指定的這個時間。時間的單位為毫秒。 <code>time</code> 可以是常數或者變數。
舉例	<pre>macro_command main() int time = 500  DELAY(100)    // delay 100 ms DELAY(time)    // delay 500 ms  end macro_command</pre>

函數名稱	SYNC_TRIG_MACRO
語法	SYNC_TRIG_MACRO(macro_id or name)
描述	<p>一個執行中的巨集指令可以使用此函數利用同步的方式觸發執行其他的巨集指令。可使用其他巨集指令的 <code>macro_id</code> 或名稱作為參數，指定被觸發的巨集指令 [編號] 或 [名稱]。</p> <p>使用此函數的巨集指令將暫停執行，直到被觸發的巨集指令執行完成才會繼續執行後續的命令。</p> <p><code>macro_id</code> 可以是常數或使用變數來表示。</p>
舉例	<pre>macro_command main() char ON = 1, OFF = 0</pre>

	<pre> SetData(ON, "Local HMI", LB, 0, 1)  SYNC_TRIG_MACRO(5) // call a macro (its ID is 5)  SYNC_TRIG_MACRO("macro_1") // call a macro (its name is macro_1)  SetData(OFF, "Local HMI", LB, 0, 1)  end macro_command </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 18.7.5. 資料操作

函數名稱	FILL
語法	<code>FILL(source[start], preset, count)</code>
描述	依序將預設值 (preset) 放置到一維陣列 source[start] 開始的陣列中，放置的資料個數由 count 決定。 source 和 start 必須為變數，preset 可以為一個常數或者變數。
舉例	<pre> macro_command main() char result[4] char preset  FILL(result[0], 0x30, 4) // result[0] is 0x30, result[1] is 0x30, , result[2] is 0x30, , result[3] is 0x30  preset = 0x31 FILL(result[0], preset, 2) // result[0] is 0x31, result[1] is 0x31  end macro_command </pre>

函數名稱	SWAPB
語法	<code>SWAPB(source, result)</code>
描述	將一個 16 位元字的高低位元組顛倒，並將結果存放在 result 變數中。 source 可以是常數或者是變數，單數 result 必須為變數。
舉例	<pre> macro_command main() short source, result  SWAPB(0x5678, result) // result is 0x7856  source = 0x123 SWAPB(source, result) // result is 0x2301  end macro_command </pre>

函數名稱	SWAPW
語法	<code>SWAPW(source, result)</code>

<b>描述</b>	將一個 32 位元雙整型資料的高位字組和低位字組顛倒，並將結果存放在 <code>result</code> 變數中。 <code>source</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
<b>舉例</b>	<pre>macro_command main() int source, result  SWAPW(0x12345678, result) // result is 0x56781234  source = 0x12345 SWAPW(source, result) // result is 0x23450001  end macro_command</pre>

<b>函數名稱</b>	LOBYTE
<b>語法</b>	<code>LOBYTE(source, result)</code>
<b>描述</b>	獲取一個 16 位元資料的低位元組，並且放置在 <code>result</code> 變數中。 <code>source</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
<b>舉例</b>	<pre>macro_command main() short source, result  LOBYTE(0x1234, result) // result is 0x34  source = 0x123 LOBYTE(source, result) // result is 0x23  end macro_command</pre>

<b>函數名稱</b>	HIBYTE
<b>語法</b>	<code>HIBYTE(source, result)</code>
<b>描述</b>	獲取一個 16 位元資料的高位元組，並且放置在 <code>result</code> 變數中。 <code>source</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
<b>舉例</b>	<pre>macro_command main() short source, result  HIBYTE(0x1234, result) // result is 0x12  source = 0x123 HIBYTE(source, result) // result is 0x01  end macro_command</pre>

<b>函數名稱</b>	LOWORD
<b>語法</b>	<code>LOWORD(source, result)</code>
<b>描述</b>	獲取一個 32 位元資料的低位字組，並將結果放置在 <code>result</code> 變數中。 <code>source</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
<b>舉例</b>	<pre>macro_command main() int source, result</pre>

	<pre>LOWORD(0x12345678, result) // result is 0x5678  source = 0x12345 LOWORD(source, result) // result is 0x2345  end macro_command</pre>
--	-------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	HIWORD
語法	HIWORD(source, result)
描述	獲取一個 32 位元資料的高位字組，並將結果放置在 <code>result</code> 變數中。 <code>source</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
舉例	<pre>macro_command main() int source, result  HIWORD(0x12345678, result) // result is 0x1234  source = 0x12345 HIWORD(source, result) // result is 0x0001  end macro_command</pre>

函數名稱	INVBIT
語法	INVBIT(source, result, bit_pos)
描述	將資料或者變數 ( <code>source</code> ) 指定的位元位址狀態相反，並將改變後的資料存放在 <code>result</code> 變數中。 <code>source</code> 和 <code>bit_pos</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
舉例	<pre>macro_command main() int source, result short bit_pos  INVBIT(4, result, 1) // result = 6  source = 6 bit_pos = 1 INVBIT(source, result, bit_pos) // result = 4  end macro_command</pre>

函數名稱	SETBITON
語法	SETBITON(source, result, bit_pos)
描述	將資料或者變數 ( <code>source</code> ) 指定的位元位址設置為 1，並將改變後的資料存放在 <code>result</code> 變數中。 <code>source</code> 和 <code>bit_pos</code> 可以是常數或者變數，但是 <code>result</code> 必須為變數。
舉例	<pre>macro_command main() int source, result</pre>

	<pre> short bit_pos  SETBITON(1, result, 3) // result is 9  source = 0 bit_pos = 2 SETBITON (source, result, bit_pos) // result is 4  end macro_command </pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	SETBITOFF
語法	SETBITOFF(source, result, bit_pos)
描述	將資料或者變數 (source) 指定的位元位址設置為 0，並將改變後的資料存放在 result 變數中。 source 和 bit_pos 可以是常數或者變數，但是 result 必須為變數。
舉例	<pre> macro_command main() int source, result short bit_pos  SETBITOFF(9, result, 3) // result is 1  source = 4 bit_pos = 2 SETBITOFF(source, result, bit_pos) // result is 0  end macro_command </pre>

函數名稱	GETBIT
語法	GETBIT(source, result, bit_pos)
描述	獲取資料或者變數 (source) 指定的位元的狀態，並將結果放置在 result 變數中。 result 的資料將為 1 或者 0。 source 和 bit_pos 可以是常數或者變數，但是 result 必須為變數。
舉例	<pre> macro_command main() int source, result short bit_pos  GETBIT(9, result, 3) // result is 1  source = 4 bit_pos = 2 GETBIT(source, result, bit_pos) // result is 1  end macro_command </pre>

### 18.7.6. 資料轉換

函數名稱	ASCII2DEC
語法	ASCII2DEC(source[start], result, len)
描述	將字元型 ASCII 資料 (source) 轉換為十進位格式的資料，並存放在 result 變數中。ASCII 的長度即為 len，第一個字元的位置即為 source[start] 的資料。 source 和 len 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。
舉例	<pre>macro_command main() char source[4] short result  source[0] = '5' source[1] = '6' source[2] = '7' source[3] = '8'  ASCII2DEC(source[0], result, 4)    // result is 5678  end macro_command</pre>

函數名稱	ASCII2FLOAT
語法	ASCII2FLOAT (source[start], result, len)
描述	將字元型 ASCII 資料 (source) 轉換為浮點數格式的資料，並存放在 result 變數中。ASCII 的長度即為 len，第一個字元的位置為 source[start] 的資料。 source 和 len 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。
舉例	<pre>macro_command main() char source[4] float result  source[0] = '5' source[1] = '6' source[2] = '.' source[3] = '8'  ASCII2FLOAT(source[0], result, 4)    // result is 56.8  end macro_command</pre>

函數名稱	ASCII2HEX
語法	ASCII2HEX (source[start], result, len)
描述	將 ASCII 字元型資料 (source) 轉換為十六進位的資料，並存放在 result 變數中。字元的長度即為 len 的資料。第一個字元存放在 source[start] 中。 source 和 len 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。
舉例	<pre>macro_command main() char source[4] short result</pre>

	<pre> source[0] = '5' source[1] = '6' source[2] = '7' source[3] = '8'  ASCII2HEX(source[0], result, 4) // result is 0x5678  end macro_command </pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	ASCII2DOUBLE
語法	ASCII2DOUBLE(source[start], result, count)
描述	將 ASCII 字元型資料 (source) 轉換為雙精度浮點數的資料，並存放在 result 變數中。字元的長度即為 count 的資料。第一個字元存放在 source[start] 中。 source 和 count 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。 此函數僅支援 cMT/cMT X 系列。
舉例	<pre> macro_command main()  char source[4] = {'5', '6', '.', '8'} double result  ASCII2DOUBLE(source[0], result, 4)// result == 56.8 SetData(result, "Local HMI", LW, 100, 1)  end macro_command </pre>

函數名稱	BIN2BCD
語法	BIN2BCD(source, result)
描述	將 BIN 格式的資料 (source) 轉換為 BCD 格式的資料 (result)。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre> macro_command main()  short source, result  BIN2BCD(1234, result) // result is 0x1234  source = 5678 BIN2BCD(source, result) // result is 0x5678  end macro_command </pre>

函數名稱	BCD2BIN
語法	BCD2BIN(source, result)
描述	將 BCD 格式的資料 (source) 轉換為 BIN 格式的資料 (result)。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre> macro_command main() </pre>

	<pre>short source, result  BCD2BIN(0x1234, result) // result is 1234  source = 0x5678 BCD2BIN(source, result) // result is 5678  end macro_command</pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	DATE2ASCII
語法	DATE2ASCII(day_offset, date[start], count, [separator])
描述	將現在日期資料經過 day_offset 的時間調整後轉換為 ASCII 格式的資料，並存放在一個一維陣列 (date) 中。count 表示這個轉換後的字串的長度，同時這個長度也取決於存放結果。separator 則是分隔年月日的分隔符號，預設為 “/” 。day_offset 和 count 可以是常數或者變數，start 與 separator 必須為常數。
舉例	<pre>macro_command main() char result[10]  DATE2ASCII(5, result[0], 10) // result[0]~[9] == "2019/02/16"// today is 2019/02/11  DATE2ASCII(5, result[0], 10, " - ") // result[0]~[9] == "2019-02-16"// today is 2019/02/11  end macro_command</pre>

函數名稱	DATE2DEC
語法	DATE2DEC(day_offset, date)
描述	將現在日期資料經過 day_offset 的時間調整後轉換為十進位格式的資料，並存放在 date 變數中。 day_offset 可以是常數或者變數，date 必須為變數。
舉例	<pre>macro_command main() int day_offset = 5, date  DATE2DEC(0, date) // date == 20190211 (Today is 2019/02/11) DATE2DEC(day_offset, date) // date == 20190216 (20190211 + 5)  end macro_command</pre>

函數名稱	DEC2ASCII
語法	DEC2ASCII(source, result[start], len)
描述	將十進位的資料 (source) 轉換為 ASCII 格式的資料，並存放在一個一維陣列 (result) 中。len 表示這個轉換後的字串的長度，同時這個長度也取決於存放結果的一維陣列的資料格式。例如：如果 result 一維陣列的格式為 “char” (字元型，長度為一個位元組)，則長度為 “位元組數*len” 。如果 result 一維陣列的格式

	<p>為 “short” (短整型資料，2 個位元組) ，則長度為 “word*len” 。依此類推。轉換後的第一個字元放在 result[start] 中，第二個字元放在 result[start+1] 中，最後一個字元放在 result[start+(len-1)] 中。</p> <p>source 和 len 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。</p>
舉例	<pre>macro_command main() short source char result1[4] short result2[4] char result3[6] source = 5678  DEC2ASCII(source, result1[0], 4) // result1[0] is '5', result1[1] is '6', result1[2] is '7', result1[3] is '8' // the length of the string (result1) is 4 bytes( = 1 * 4) DEC2ASCII(source, result2[0], 4) // result2[0] is '5', result2[1] is '6', result2[2] is '7', result2[3] is '8' // the length of the string (result2) is 8 bytes( = 2 * 4)  source=-123 DEC2ASCII(source, result3[0], 6) // result1[0] is '-', result1[1] is '0', result1[2] is '0', result1[3] is '1' // result1[4] is '2', result1[5] is '3' // the length of the string (result1) is 6 bytes( = 1 * 6)  end macro_command</pre>

函數名稱	FLOAT2ASCII
語法	FLOAT2ASCII (source, result[start], len)
描述	浮點數格式資料 (source) 轉換為 ASCII 格式的資料，並將結果存放在一個一維陣列 (result) 中。len 表示這個轉換後的字串的長度，同時這個長度也取決於存放結果的一維陣列的資料格式。例如：如果 result 一維陣列的格式為 “char” (字元型，長度為一個位元組)，則長度為 “位元組數*len” 。如果 result 一維陣列的格式為 “short” (短整型資料，2 個位元組)，則長度為 “word*len” 。依此類推。source 和 len 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。
舉例	<pre>macro_command main() float source char result[4]  source = 56.8 FLOAT2ASCII (source, result[0], 4) // result[0] is '5', result[1] is '6', result[2] is '.', result[3] is '8'  end macro_command</pre>

函數名稱	HEX2ASCII
------	-----------

語法	<code>HEX2ASCII(source, result[start], len)</code>
描述	十六進位格式資料 (source) 轉換為 ASCII 格式的資料，並將結果存放在一個一維陣列 (result) 中。len 表示這個轉換後的字串的長度，同時這個長度也取決於存放結果的一維陣列的資料格式。例如：如果 result 一維陣列的格式為 “char” (字元型，長度為一個位元組)，則長度為 “位元組數*len” 。如果 result 一維陣列的格式為 “short” (短整型資料，2 個位元組)，則長度為 “word*len” 。依此類推。source 和 len 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。
舉例	<pre>macro_command main() short source char result[4]  source = 0x5678 HEX2ASCII(source, result[0], 4) // result[0] is '5', result[1] is '6', result[2] is '7', result[3] is '8'  end macro_command</pre>

函數名稱	DOUBLE2ASCII
語法	<code>DOUBLE2ASCII(source, result[start], count)</code>
描述	雙精度浮點數格式資料 (source) 轉換為 ASCII 格式的資料，並將結果存放在一個一維陣列 (result) 中。count 表示這個轉換後的字串的長度，同時這個長度也取決於存放結果的一維陣列的資料格式。例如：如果 result 一維陣列的格式為 “char” (字元型，長度為一個位元組)，則長度為 “位元組數*count” 。如果 result 一維陣列的格式為 “short” (短整型資料，2 個位元組)，則長度為 “word* count” 。依此類推。source 和 count 可以是常數或者變數，單數 result 必須為變數。start 必須為常數。此函數僅支援 cMT/cMT X 系列。
舉例	<pre>macro_command main()  double source = 56.8 char result[4]  DOUBLE2ASCII(source, result[0], 4) // result[0] == '5', result[1] == '6', result[2] == '.', result[3] == '8'  end macro_command</pre>

函數名稱	StringDecAsc2Bin
語法	<code>success = StringDecAsc2Bin(source[start], destination)</code> 或 <code>success = StringDecAsc2Bin("source", destination)</code>
描述	<p>此函數將十進位字串轉換成整數。</p> <p>來源字串 source 可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。</p> <p>destination 必須為一變數，用以存放轉換後的整數值。</p>

	<p>執行完畢會回傳一 <b>bool</b> 型態的值給 <b>success</b> 欄位。當轉換成功，<b>success</b>等於 <b>true</b>，否則等於 <b>false</b>。</p> <p>來源字串必需為十進位字串，若其包含正負號或 ‘0’ ~ ‘9’ 以外的字元，函數將會回傳<b>false</b>。</p> <p><b>success</b> 欄位可寫可不寫。</p>
<b>舉例</b>	<pre>macro_command main() char src1[5] = "12345" int result1 bool success1 success1 = StringDecAsc2Bin(src1[0], result1) // success1 = true , result1 為 "12345"  char src2[5] = "-6789" short result2 bool success2 success2 = StringDecAsc2Bin(src2[0], result2) // success2 = true , result2 為 "-6789"  char result3 bool success3 success3= StringDecAsc2Bin("32768", result3) // success3 = true , 但結果超出 result3 所能表達的範圍  char src4[2] = "4b" char result4 bool success4 success4 = StringDecAsc2Bin (src4[0], result4) // success4= false , 因 src4 包含正負號 或 ‘0’ ~ ‘9’ 以外的字元  end macro_command</pre>

<b>函數名稱</b>	StringBin2DecAsc
<b>語法</b>	<b>success</b> = StringBin2DecAsc ( <b>source</b> , <b>destination</b> [ <b>start</b> ])
<b>描述</b>	<p>此函數將整數轉換成十進位字串。</p> <p>來源 <b>source</b> 可以為常數或變數。</p> <p><b>destination</b>[<b>start</b>] 必須為一維字元陣列變數，用以存放轉換後的十進位字串。</p> <p>執行完畢會回傳一 <b>bool</b> 型態的值給 <b>success</b> 欄位。當轉換成功，<b>success</b>等於 <b>true</b>，否則等於 <b>false</b>。</p> <p>若轉換後的十進位字串長度大於目標陣列的大小，函數將會回傳 <b>false</b>。</p> <p><b>success</b> 欄位可寫可不寫。</p> <p>注意：此函數不能轉換負值。</p>
<b>舉例</b>	<pre>macro_command main() int src1 = 2147483647 char dest1[20] bool success1</pre>

	<pre> success1 = StringBin2DecAsc(src1, dest1[0]) // success1 = true , dest1為 "2147483647"  short src2 = 0x3c char dest2[20] bool success2 success2 = StringBin2DecAsc(src2, dest2[0]) // success2 = true , dest2為 "60"  int src3 = 2147483647 char dest3[5] bool success3 success3 = StringBin2DecAsc(src3, dest3[0]) // success3 = false , dest3 內容不變  end macro_command </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringDecAsc2Float
語法	<pre> success = StringDecAsc2Float (source[start], destination) 或 success = StringDecAsc2Float ("source", destination) </pre>
描述	<p>此函數將十進位字串轉換成浮點數。</p> <p>來源字串 <code>source</code> 可以為靜態字串 (如 <code>"source"</code>) 或是一維字元陣列變數 (如 <code>source[start]</code> ) 。</p> <p><code>destination</code> 必須為一變數，用以存放轉換後的浮點數值。</p> <p>執行完畢會回傳一 <code>bool</code> 型態的值給 <code>success</code> 欄位。當轉換成功，<code>success</code> 等於 <code>true</code>，否則等於 <code>false</code> 。</p> <p>來源字串必需為十進位字串，若其包含 <code>'0' ~ '9'</code> 或 <code>'.'</code> 以外的字元，函數將會回傳 <code>false</code> 。</p> <p><code>success</code> 欄位可寫可不寫。</p>
舉例	<pre> macro_command main() char src1[10] = "12.345" float result1 bool success1 success1 = StringDecAsc2Float(src1[0], result1) // success1 = true , result1 為 "12.345"  float result2 bool success2 success2 = StringDecAsc2Float("1.234567890", result2) // success2 = true , 但結果超出 result2 所能表達的範圍，可能喪失精確度  char src3[2] = "4b" float result3 bool success3 success3 = StringDecAsc2Float(src3[0], result3) // success3 = false , 因 src3 包含 '0' ~ '9' 或 '.' 以外的字元 </pre>

	end macro_command
--	-------------------

函數名稱	StringFloat2DecAsc
語法	success = StringFloat2DecAsc(source, destination[start])
描述	<p>此函數將浮點數轉換成十進位字串。</p> <p>來源 source 可以為常數或變數。</p> <p>destination[start] 必須為一維字元陣列變數，用以存放轉換後的十進位字串。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當轉換成功，success等於 true，否則等於 false。</p> <p>若轉換後的十進位字串長度大於目標陣列的大小，函數將會回傳 false。</p> <p>success 欄位可寫可不寫。</p>
舉例	<pre>macro_command main() float src1 = 1.2345 char dest1[20] bool success1 success1 = StringFloat2DecAsc(src1, dest1[0]) // success1 = true , dest1為 "1.2345"  float src2 = 1.23456789 char dest2 [20] bool success2 success2 = StringFloat2DecAsc(src2, dest2 [0]) // success2 = true , 但可能喪失精確度  float src3 = 1.2345 char dest3[5] bool success3 success3 = StringFloat2DecAsc(src3, dest3 [0]) // success3 = false , dest3 內容不變  end macro_command</pre>

函數名稱	StringHexAsc2Bin
語法	<p>success = StringHexAsc2Bin (source[start], destination)</p> <p>或</p> <p>success = StringHexAsc2Bin ("source", destination)</p>
描述	<p>此函數將十六進位字串轉換成整數。</p> <p>來源字串 source 可以為靜態字串 (如 "source" )或是一維字元陣列變數 (如 source[start])。</p> <p>destination 必須為一變數，用以存放轉換後的整數值。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當轉換成功，success等於 true，否則等於 false。</p>

	來源字串必需為十六進位字串，若其包含 ‘0’ ~ ‘9’ 或 ‘a’ ~ ‘f’ 或 ‘A’ ~ ‘F’ 以外的字元，函數將會回傳 false。 success 欄位可寫可不寫。
舉例	<pre>macro_command main() char src1[5]="0x3c" int result1 bool success1 success1 = StringHexAsc2Bin(src1[0], result1) // success1 = true , result1 為 3c  short result2 bool success2 success2 = StringDecAsc2Bin("1a2b3c4d", result2) // success2 = true , 但結果超出 result2 所能表達的範圍 , result2 = 3c4d  char src3[2] = "4g" char result3 bool success3 success3 = StringDecAsc2Bin (src3[0], result3) // success3=false , 因 src3包含 ‘0’ ~ ‘9’ 或 ‘a’ ~ ‘f’ 或 ‘A’ ~ ‘F’ 以外的字元  end macro_command</pre>

函數名稱	StringBin2HexAsc
語法	success = StringBin2HexAsc (source, destination[start])
描述	<p>此函數將整數轉換成十六進位字串。</p> <p>來源source可以為常數或變數。</p> <p>destination[start] 必須為一維字元陣列變數，用以存放轉換後的十六進位字串。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當轉換成功，success等於 true，否則等於 false。</p> <p>若轉換後的十六進位字串長度大於目標陣列的大小，函數將會回傳 false。</p> <p>success 欄位可寫可不寫。</p> <p>注意：此函數不能轉換負值。</p>
舉例	<pre>macro_command main() int src1 = 20 char dest1[20] bool success1 success1 = StringBin2HexAsc(src1, dest1[0]) // success1 = true , dest1 為 "14"  short src2 = 0x3c char dest2[20] bool success2 success2 = StringBin2HexAsc(src2, dest2[0]) // success2 = true , dest2 為 "3c"</pre>

	<pre> int src3 = 0x1a2b3c4d char dest3[6] bool success3 success3 = StringBin2HexAsc(src3, dest3[0]) // success3 = false , dest3 內容不變  end macro_command </pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 18.7.7. 字串處理

函數名稱	String2Unicode
語法	result = String2Unicode("source", destination[start])
描述	將source字串轉換為Unicode字串，存放到destination，轉換後的Unicode字串長度會存到result。Source來源必須為常數，不可為變數。
舉例	<pre> macro_command main()  char dest[20] int result  result = String2Unicode("abcde", dest[0]) // "result" will be set to 10. result = String2Unicode("abcdefghijklmno", dest[0]) // "result" will be set to 20. // "result" will be the length of converted Unicode string  end macro_command </pre>

函數名稱	StringCat
語法	success = StringCat (source[start], destination[start]) 或 success = StringCat ("source", destination[start])
描述	利用此函數將來源字串銜接於目標字串之後。此函數執行成功後，目標字串將等於兩字串銜接後的結果。 來源字串source可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。 destination[start]必須為一維字元陣列變數。 執行完畢會回傳一 bool 型態的值給 success 欄位。當執行成功，success等於 true，否則等於 false。當兩字串銜接後的長度超過目標陣列的長度時，目標字串將保留銜接以前的內容，不做任何改變，並回傳 false。
舉例	<pre> macro_command main() char src1[20] = "abcdefghijkl" char dest1[20] = "1234567890" bool success1 success1= StringCat(src1[0], dest1[0]) </pre>

	<pre>// success1 = true , dest1 = "123456790abcdefghijklm" char dest2 [10] = "1234567890" bool success2 success2= StringCat("abcde", dest2 [0]) // success2 = false , dest2 內容不變  char src3[20] = "abcdefghijklm" char dest3[20] bool success3 success3 = StringCat(src3[0], dest3[15]) // success3 = false , dest3 內容不變  end macro_command</pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringCompare
語法	<pre>ret = StringCompare (str1[start], str2[start]) ret = StringCompare ("string1", str2[start]) ret = StringCompare (str1[start], "string2") ret = StringCompare ("string1", "string2")</pre>
描述	<p>比較兩字串的內容是否相等。此函數將大小寫視為不同。</p> <p>兩個傳入的字串皆可以為靜態字串 (如 "source")或是一維字元陣列變數 (如 source[start])。</p> <p>執行完畢會回傳一 bool型態的值給 ret 欄位。若兩字串相等，ret 為 true，否則為 false。</p>
舉例	<pre>macro_command main() char a1[20] = "abcde" char b1[20] = "ABCDE" bool ret1 ret1 = StringCompare(a1[0], b1[0]) // ret1 = false  char a2[20] = "abcde" char b2[20] = "abcde" bool ret2 ret2 = StringCompare(a2[0], b2[0]) // ret2 = true  char a3 [20] = "abcde" char b3[20] = "abcdefg" bool ret3 ret3 = StringCompare(a3[0], b3[0]) // ret3 = false  end macro_command</pre>

函數名稱	StringCompareNoCase
------	---------------------

語法	<pre>ret = StringCompareNoCase(str1[start], str2[start]) ret = StringCompareNoCase("string1", str2[start]) ret = StringCompareNoCase(str1[start], "string2") ret = StringCompareNoCase("string1", "string2")</pre>
描述	<p>比較兩字串的內容是否相等。此函數將大小寫視為相同。</p> <p>兩個傳入的字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數(如 source[start])。</p> <p>執行完畢會回傳一 bool 型態的值給 ret 欄位。若兩字串相等，ret 為 true，否則為 false。</p>
舉例	<pre>macro_command main() char a1[20] = "abcde" char b1[20] = "ABCDE" bool ret1 ret1 = StringCompareNoCase(a1[0], b1[0]) // ret1 = true  char a2[20] = "abcde" char b2[20] = "abcde" bool ret2 ret2 = StringCompareNoCase(a2[0], b2[0]) // ret2 = true  char a3 [20] = "abcde" char b3[20] = "abcdefg" bool ret3 ret3 = StringCompareNoCase(a3[0], b3[0]) // ret3 = false  end macro_command</pre>

函數名稱	StringCopy
語法	<pre>success = StringCopy ("source", destination[start]) 或 success = StringCopy (source[start], destination[start])</pre>
描述	<p>利用此函數進行字串的複製。此函數可以將傳入的靜態字串 (以雙引號" "括起來的字串)，或是存放在字元陣列中的字串複製到目的 buffer。</p> <p>來源字串 source可以為靜態字串 (如 "source" )或是一維字元陣列變數 (如 source[start])。</p> <p>destination[start] 必須為一維字元陣列變數。</p> <p>複製完畢會回傳一 bool 型態的值給 success 欄位。當複製成功，success等於 true，否則等於 false。當來源字串的長度大於目的 buffer 的大小時，將不做任何處理，並回傳 false 到 success 欄位。</p> <p>success 欄位可寫可不寫。</p>
舉例	<pre>macro_command main() char src1[5] = "abcde" char dest1[5]</pre>

```

bool success1
success1 = StringCopy(src1[0], dest1[0])
// success1=true , dest1為 "abcde"
char dest2[5]
bool success2
success2 = StringCopy("12345", dest2[0])
// success2 = true , dest2為 "12345"
char src3[10] = "abcdefghijklm"
char dest3[5]
bool success3
success3 = StringCopy(src3[0], dest3[0])
// success3 = false , dest3 內容不變
char src4[10] = "abcdefghijklm"
char dest4[5]
bool success4
success4 = StringCopy(src4[5], dest4[0])
// success4=true , dest4為 "fghij"

end macro_command

```

函數名稱	StringExcluding
語法	<pre> success = StringExcluding (source[start], set[start], destination[start]) success = StringExcluding ("source", set[start], destination[start]) success = StringExcluding (source[start], "set", destination[start]) success = StringExcluding ("source", "set", destination[start]) </pre>
描述	<p>提取 <code>source</code> 字串中某個以索引值 0 的字元 (第一個字元) 開頭的子字串，而且此子字串的每個字元必不存在於<code>set</code>字串中。此函數將會從 <code>source</code>字串的第一個字元開始尋找，直到找到存在於 <code>set</code> 字串中的字元為止。</p> <p><code>source</code> 字串與 <code>set</code> 字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 <code>source[start]</code>)。</p> <p>執行完畢會回傳一 <code>bool</code> 型態的值給 <code>success</code> 欄位。當執行成功，<code>success</code>等於 <code>true</code>，否則等於 <code>false</code>。當提取出來的字串長度大於目標陣列的大小，將會回傳 <code>false</code>。</p>
舉例	<pre> macro_command main() char src1[20] = "cabbageabc" char set1[20] = "ge" char dest1[20] bool success1 success1 = StringExcluding(src1[0], set1[0], dest1[0]) // success1 = true , dest1 = "cabba"  char src2[20] = "cabbage" char dest2[20] bool success2 success2 = StringExcluding(src2[0], "abc", dest2[0]) // success2 = true , dest2 = "" </pre>

	<pre> char set3[20] = "ge" char dest3[4] bool success3 success3 = StringExcluding("cabbage", set3[0], dest3[0]) // success3 = false , dest3 內容不變  end macro_command </pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringFind
語法	<pre> position = StringFind (source[start], target[start]) position = StringFind ("source", target[start]) position = StringFind (source[start], "target") position = StringFind ("source", "target") </pre>
描述	<p>尋找某一字串 (target) 在另一個字串 (source) 裡第一次出現的位置。</p> <p>兩個傳入的字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數(如 source[start])。</p> <p>執行完畢會回傳target字串在 source 字串裡出現的位置。Source 字串由 0 開始遞增為字元編索引值。若 source 字串存在一個子字串，其所包含的字元與排列順序跟 target 字串完全相等，則函數會回傳此子字串開頭的索引值，若沒有找到，則回傳 -1 。</p>
舉例	<pre> macro_command main() char src1[20] = "abcde" char target1[20] = "cd" short pos1 pos1 = StringFind(src1[0], target1[0]) // pos1 = 2  char target2[20] = "ce" short pos2 pos2 = StringFind( "abcde", target2[0]) // pos2 = -1  char src3[20] = "abcde" short pos3 pos3 = StringFind(src3[3], "cd") // pos3 = -1  end macro_command </pre>

函數名稱	StringFindOneOf
語法	<pre> position = StringFindOneOf (source[start], target[start]) position = StringFindOneOf ("source", target[start]) position = StringFindOneOf (source[start], "target") position = StringFindOneOf ("source", "target") </pre>
描述	尋找 target 字串中任一個字元在 source 字串中第一次出現的位置。

	<p>兩個傳入的字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數(如 source[start])。</p> <p>執行完畢會回傳 target 字串中任一個字元在 source 字串裡第一次出現的位置，即 source 字串中為該字元編的索引值 (由 0 開始)。若沒有找到，則回傳 -1。</p>
舉例	<pre>macro_command main() char src1[20] = "abcdeabcde" char target1[20] = "sdf" short pos1 pos1 = StringFindOneOf(src1[0], target1[0]) // pos1 = 3  char src2[20] = "abcdeabcde" short pos2 pos2 = StringFindOneOf(src2[1], "agi") // pos2 = 4  char target3 [20] = "bus" short pos3 pos3 = StringFindOneOf("abcdeabcde", target3[1]) // pos3 = -1  end macro_command</pre>

函數名稱	StringGet																				
語法	<code>StringGet(read_data[start], device_name, device_type, address_offset, data_count)</code>																				
描述	<p>獲取設備的資料。字串的資料型態為字元陣列，是存儲在 <code>read_data[start]~read_data[start+data_count-1]</code> 這些一維陣列變數中。<code>read_data</code> 必須為一維字元陣列。</p> <p><code>data_count</code> 是設定的讀取字元的個數，可以是常數也可以是變數。</p> <p>此處的 <code>device_name</code>，即為在“系統參數”中建立設備類型時，設定的“設備名稱”。在此，設備名稱被設定為“FATEK FB Series”，如下圖所示。</p> <table border="1"> <thead> <tr> <th>編號</th> <th>名稱</th> <th>位置</th> <th>設備類型</th> <th>介面類型</th> </tr> </thead> <tbody> <tr> <td>本機 HMI</td> <td>Local HMI</td> <td>本機</td> <td>MT8104iH (800 x...)</td> <td>-</td> </tr> <tr> <td>本機 同伺服器</td> <td>Free Protocol</td> <td>本機</td> <td>Free Protocol</td> <td>COM 1 (9600)</td> </tr> <tr> <td>遠端 PLC 1</td> <td>FATEK FB Series</td> <td>遠端 (IP:192.168.1.10, ...)</td> <td>FATEK FB Series</td> <td>COM 1 (9600)</td> </tr> </tbody> </table> <p><code>device_type</code> 是設備類型和設備中資料的編碼方式。例如：如果 <code>device_type</code> 是 <code>LW_BIN</code>，那麼讀取的設備類型為 <code>LW</code>，資料編碼方式為 <code>BIN</code>。如果使用 <code>BIN</code> 編碼方式，“_BIN”可以忽略。</p> <p>如果 <code>device_type</code> 是 <code>LW_BCD</code>，表示設備類型 <code>LW</code>，資料的編碼方式為 <code>BCD</code> 格式。</p>	編號	名稱	位置	設備類型	介面類型	本機 HMI	Local HMI	本機	MT8104iH (800 x...)	-	本機 同伺服器	Free Protocol	本機	Free Protocol	COM 1 (9600)	遠端 PLC 1	FATEK FB Series	遠端 (IP:192.168.1.10, ...)	FATEK FB Series	COM 1 (9600)
編號	名稱	位置	設備類型	介面類型																	
本機 HMI	Local HMI	本機	MT8104iH (800 x...)	-																	
本機 同伺服器	Free Protocol	本機	Free Protocol	COM 1 (9600)																	
遠端 PLC 1	FATEK FB Series	遠端 (IP:192.168.1.10, ...)	FATEK FB Series	COM 1 (9600)																	

`address_offset` 是設備中的地址偏移量。  
 例如，`StringGet(read_data_1[0], "FATEK FB Series", RT, 5, 1)` 代表讀取的設備位址偏移量為 5。  
 如果 `address_offset` 使用格式為 "N#AAAAA"，N 表示設備的站號，AAAAA 表示位址偏移量。此情況一般使用在同一個串列埠上連接有多台設備或者控制器的情況下。例如：`StringGet(read_data_1[0], "FATEK FB Series", RT, 2#5, 1)` 表示讀取站號為 2 的設備的資料。如果 `StringGet()` 使用 “系統參數 / 設備列表” 中設定的默認的站號，在此可以不填這個站號。



從設備中讀取的資料個數，由 `data_count` 的值來決定，因 `read_data` 變數僅接受 `char` 陣列型態。如下表所示：

<code>read_data</code> 的類型	<code>data_count</code> 的值	讀取 16 位元數據的個數
<code>char (8-bit)</code>	1	1
<code>char (8-bit)</code>	2	1

因為一個 WORD (16 位元) 等於 2 個 ASCII 字元的長度，當設備類型長度為 WORD 時，根據上表，讀取 2 個 ASCII 字元實際上是讀 1 個 WORD 的數據。

#### 舉例

```
macro_command main()
char str1[20]

// 讀取 LW-0 ~ LW-9 共 10 個 WORD 到變數 str1[0] 到 str1[19] 中
// 因 1 WORD 可存放 2 個 ASCII 字元，欲讀取 20 個 ASCII 字元
```

	<pre>// 實際上共讀取了 10 個WORD StringGet(str1[0], "Local HMI", LW, 0, 20)  end macro_command</pre>
--	----------------------------------------------------------------------------------------------

函數名稱	StringGetEx
語法	<pre>StringGetEx (read_data[start], device_name, device_type, address_offset, data_count)</pre>
描述	<p>獲取設備的資料，不等待設備回應，逕自往下執行。</p> <p>read_data、device_name、device_type、address_offset 和 data_count 的說明和 GetData 相同。</p>
舉例	<pre>macro_command main() char str1[20] short test=0  // 當 MODBUS 設備未回應，test = 1 將照常執行 StringGetEx(str1[0], "MODBUS RTU", 4x, 0, 20) test = 1  // 當 MODBUS 設備未回應，test = 2 將不會被執行，直到得到回應 StringGet(str1[0], "MODBUS RTU", 4x, 0, 20) test = 2  end macro_command</pre>

函數名稱	StringIncluding
語法	<pre>success = StringIncluding (source[start], set[start], destination[start]) success = StringIncluding ("source", set[start], destination[start]) success = StringIncluding (source[start], "set", destination[start]) success = StringIncluding ("source", "set", destination[start])</pre>
描述	<p>提取 source 字串中某個以索引值 0 的字元 (第一個字元) 開頭的子字串，而且此子字串的每個字元都能在 set 字串中找到相同的字元。此函數將會從 source 字串的第一個字元開始尋找，直到找到不存在於 set 字串中的字元為止。</p> <p>source 字串與 set 字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當執行成功，success 等於 true，否則等於 false。當提取出來的字串長度大於目標陣列的大小，將會回傳 false。</p>
舉例	<pre>macro_command main() char src1[20] = "cabbageabc" char set1[20] = "abc" char dest1[20] bool success1 success1 = StringIncluding(src1[0], set1[0], dest1[0]) // success1 = true, dest1 = "cabba"</pre>

	<pre> char src2[20] = "gecabba" char dest2[20] bool success2 success2 = StringIncluding(src2[0], "abc", dest2[0]) // success2 = true , dest2 = "  "  char set3[20] = "abc" char dest3[4] bool success3 success3 = StringIncluding("cabbage", set3[0], dest3[0]) // success3 = false , dest3內容不變  end macro_command </pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringInsert
語法	<pre> success = StringInsert (pos, insert[start], destination[start]) success = StringInsert (pos, "insert", destination[start]) success = StringInsert (pos, insert[start], length, destination[start]) success = StringInsert (pos, "insert", length, destination[start]) </pre>
描述	<p>將 <code>insert</code> 字串插入到目標字串中的特定位置，插入位置由 <code>pos</code> 所指定。  <code>Insert</code> 字串可以為靜態字串 (如 "insert") 或是一維字元陣列變數 (如 <code>insert[start]</code>)。</p> <p>使用者亦可以在 <code>length</code> 欄位指定 <code>insert</code> 字串的長度。      執行完畢會回傳一 <code>bool</code> 型態的值給 <code>success</code> 欄位。當執行成功，<code>success</code> 等於 <code>true</code>，否則等於 <code>false</code>。當插入完成後的字串長度大於目標陣列的大小，將會回傳 <code>false</code>。</p>
舉例	<pre> macro_command main()  char str1[20] = "but the question is" char str2[10] = ", that is" char dest[40] = "to be or not to be" bool success  success = StringInsert(18, str1[3], 13, dest[0]) // success = true, dest = "to be or not to be the question"  success = StringInsert(18, str2[0], dest[0]) // success=true, dest="to be or not to be, that is the question"  success = StringInsert(0, "Hamlet:", dest[0]) // success = false, dest 內容不變  end macro_command </pre>

函數名稱	StringLength
語法	<pre> length = StringLength (source[start]) 或 </pre>

	<code>length = StringLength ("source")</code>
<b>描述</b>	<p>取得字串的長度。</p> <p>來源字串 <code>source</code> 可以為靜態字串 (如 "source") 或是一維字元陣列變數(如 <code>source[start]</code>)。</p> <p>函數回傳值代表來源字串的長度。</p>
<b>舉例</b>	<pre>macro_command main() char src1[20] = "abcde" int length1 length1= StringLength(src1[0]) // length1 = 5  char src2[20] = { 'a', 'b', 'c', 'd', 'e' } int length2 length2= StringLength(src2[0]) // length2 = 5  char src3[20] = "abcdefghijklm" int length3 length3 = StringLength(src3 [2]) // length3 = 8  end macro_command</pre>

<b>函數名稱</b>	<code>StringMD5</code>
<b>語法</b>	<code>result = StringMD5(source[start], destination[start])</code> 或 <code>result = StringMD5("source", destination[start])</code>
<b>描述</b>	<p>利用此函數可產生一組透過MD5訊息摘要演算法的字串。</p> <p>來源字串 <code>source</code> 可以為靜態字串 (如 "source")或是一維字元陣列變數 (如 <code>source[start]</code>)。當來源字串為字元陣列變數時，由陣列下標決定子字串起始位置。</p> <p><code>destination[start]</code> 必須為一維字元陣列變數，用以存放產生出來的子字串。</p> <p>執行完畢會回傳MD5字串長度給 <code>result</code> 欄位。</p>
<b>舉例</b>	<pre>macro_command main() char source[32] = "password", dest[32] int result  result = StringMD5(source[0], dest[0]) result = StringMD5("password", dest[0]) // "result" will be set to 32. // "result" will be the length of MD5 string. // dest[] = 5f4dcc3b5aa765d61d8327deb882cf99  end macro_command</pre>

<b>函數名稱</b>	<code>StringMid</code>
<b>語法</b>	<code>success = StringMid (source[start], count, destination[start])</code>

	<p>或  <code>success = StringMid ("string", start, count, destination[start])</code></p>
描述	<p>利用此函數可將一個字串中的某一段子字串提取出來。</p> <p>來源字串 <code>source</code> 可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 <code>source[start]</code>)。當來源字串為字元陣列變數時，由陣列下標決定子字串起始位置。當來源字串為靜態字串時，由第二個參數 <code>start</code> 決定子字串起始位置。</p> <p><code>count</code> 決定要提取的子字串長度。</p> <p><code>destination[start]</code> 必須為一維字元陣列變數，用以存放提取出來的子字串。執行完畢會回傳一 <code>bool</code> 型態的值給 <code>success</code> 欄位。當執行成功，<code>success</code> 等於 <code>true</code>，否則等於 <code>false</code>。</p> <p>若提取出來的子字串長度大於目標陣列的大小，函數將會回傳 <code>false</code>。</p> <p><code>success</code> 欄位可寫可不寫。</p>
舉例	<pre>macro_command main() char src1[20] = "abcdefghijklmnpqrst" char dest1[20] bool success1 success1 = StringMid(src1[5], 6, dest1[0]) // success1 = true , dest1 為 "fghijk"  char src2[20] = "abcdefghijklmnpqrst" char dest2[5] bool success2 success2 = StringMid(src2[5], 6, dest2[0]) // success2 = false , dest2 內容不變  char dest3[20] = "12345678901234567890" bool success3 success3 = StringMid("abcdefghijklmnpqrst", 5, 5, dest3[15]) // success3 = true , dest3 = "123456789012345fghij"  end macro_command</pre>

函數名稱	<code>StringReverseFind</code>
語法	<code>position = StringReverseFind (source[start], target[start])</code> <code>position = StringReverseFind ("source", target[start])</code> <code>position = StringReverseFind (source[start], "target")</code> <code>position = StringReverseFind ("source", "target")</code>
描述	<p>尋找某一字串 (<code>target</code>) 在另一個字串 (<code>source</code>) 裡最後一次出現的位置。</p> <p>兩個傳入的字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 <code>source[start]</code>)。</p> <p>執行完畢會回傳 <code>target</code> 字串在 <code>source</code> 字串裡最後一次出現的位置。<code>source</code> 字串由 0 開始遞增為字元編索引值。若 <code>source</code> 字串存在一個子字串，其所包含的字元與排列順序跟 <code>target</code> 字串完全相等，則函數會回傳此子字串開頭的索引值，若沒有找到，則回傳 -1。若 <code>source</code> 字串中存在多個與 <code>target</code> 字串相等的子字串，則回傳最後一個出現的子字串的位置。</p>

舉例	<pre> macro_command main() char src1[20] = "abcdeabcde" char target1[20] = "cd" short pos1 pos1 = StringReverseFind(src1[0], target1[0]) // pos1 = 7  char target2[20] = "ce" short pos2 pos2 = StringReverseFind( "abcdeabcde" , target2[0]) // pos2 = -1  char src3[20] = "abcdeabcde" short pos3 pos3 = StringReverseFind(src3[6], "ab") // pos3 = -1  end macro_command </pre>
----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

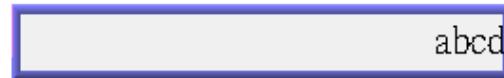
函數名稱	StringSet			
語法	StringSet (send_data[start], device_name, device_type, address_offset, data_count)			
描述	<p>將數據寫到設備中。字串資料保存在 <code>send_data[start]~send_data[start+data_count-1]</code> 中，<code>send_data</code> 必須為一維字元陣列型態。</p> <p><code>data_count</code> 是寫入到設備中字元資料的個數，可以是常數也可以是變數。</p> <p><code>device_name</code> 詳見上面的說明，在此不在說明。</p> <p><code>device_type</code> 是設備類型和設備中資料的編碼方式。例如：如果 <code>device_type</code> 是 <code>LW_BIN</code>，那麼讀取的設備類型為 <code>LW</code>，資料編碼方式為 <code>BIN</code>。如果使用 <code>BIN</code> 編碼方式，"<code>_BIN</code>" 可以忽略。</p> <p>如果 <code>device_type</code> 是 <code>LW_BCD</code>，表示設備類型 <code>LW</code>，資料的編碼方式為 <code>BCD</code> 格式。</p> <p><code>address_offset</code> 是設備中的地址偏移量。</p> <p>例如，<code>StringSet(read_data_1[0], "FATEK FB Series", RT, 5, 1)</code> 代表讀取的設備位址偏移量為 5。</p> <p>如果 <code>address_offset</code> 使用格式為 "<code>N#AAAAAA</code>"，<code>N</code> 表示設備的站號，<code>AAAAAA</code> 表示位址偏移量。此情況一般使用在同一個串列埠上連接有多台設備或者控制器的情況下。例如：<code>StringSet(send_data_1[0], "FATEK FB Series", RT, 2#5, 1)</code> 表示設定站號為 2 的設備的資料。如果 <code>StringSet()</code> 使用“系統參數 / 設備列表”中設定的默認的站號，在此可以不填這個站號。</p> <p>設定到設備的資料個數，根據 <code>data_count</code> 的值來決定，因 <code>send_data</code> 變數僅接受 <code>char</code> 陣列型態。如下表所示：</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="padding: 5px;"><code>send_data</code> 的類型</td> <td style="padding: 5px;"><code>data_count</code> 的值</td> <td style="padding: 5px;">設定 16 位元數據的個數</td> </tr> </table>	<code>send_data</code> 的類型	<code>data_count</code> 的值	設定 16 位元數據的個數
<code>send_data</code> 的類型	<code>data_count</code> 的值	設定 16 位元數據的個數		

char (8-bit)	1	1
char (8-bit)	2	1

因為一個 WORD (16 位元) 等於 2 個 ASCII 字元的長度，當設備類型長度為 WORD 時，根據上表，寫入 2 個 ASCII 字元實際上是寫 1 個 WORD 的數據。巨集指令會以先寫 low byte 再寫 hight byte 的順序，依序將 ASCII 字元寫入。使用 [字元顯示] 物件顯示數據時，data\_count 必須填入 2 的倍數才能正確顯示。例如：

```
macro_command main()
char src1[10]="abcde"
StringSet(src1[0], "Local HMI", LW, 0, 5)
end macro_command
```

[字元顯示] 物件顯示如下：



abcd

當 data\_count 為一個大於或等於字串長度的偶數時，可以完整顯示整個字串：

```
macro_command main()
char src1[10]="abcde"
StringSet(src1[0], "Local HMI", LW, 0, 6)
end macro_command
```



abcde

舉例	<pre>macro_command main() char str1[10]= "abcde"  // 字串 str1 寫入 LW-0 ~ LW-2 共三個 WORD // 即使 data_count 為 10，寫到第 3 個 WORD 時發現字串已結束， // 便不再寫入後面的數據 StringSet(str1[0], "Local HMI", LW, 0, 10)  end macro_command</pre>
----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringSetEx
語法	StringSetEx(send_data[start], device_name, device_type, address_offset, data_count)
描述	將數據寫到設備中，不等待設備回應，逕自往下執行。 send_data、device_name、device_type、address_offset 和 data_count 的說明和 StringSet 相同。

舉例	<pre> macro_command main() char str1[20]= "abcde" short test=0  // 當 MODBUS 設備未回應，test = 1 將照常執行 StringSetEx(str1[0], "MODBUS RTU", 4x, 0, 20) test = 1  // 當 MODBUS 設備未回應，test = 2 將不會被執行，直到得到回應 StringSet(str1[0], "MODBUS RTU", 4x, 0, 20) test = 2  end macro_command </pre>
----	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringToUpper
語法	<pre> success = StringToUpper (source[start], destination[start]) success = StringToUpper ("source", destination[start]) </pre>
描述	<p>將 source 字串的字元全部轉換成大寫，並寫入 destination 字串。</p> <p>source 字串可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當執行成功，success等於 true，否則等於 false。source 字串長度大於目標陣列的大小，將會回傳 false。</p>
舉例	<pre> macro_command main() char src1[20] = "aBcDe" char dest1[20] bool success1 success1 = StringToUpper(src1[0], dest1[0]) // success1 = true , dest1 = "ABCDE"  char dest2[4] bool success2 success2 = StringToUpper("aBcDe", dest2[0]) // success2 = false , dest2 內容不變  end macro_command </pre>

函數名稱	StringToLower
語法	<pre> success = StringToLower (source[start], destination[start]) success = StringToLower ("source", destination[start]) </pre>
描述	<p>將 source 字串的字元全部轉換成小寫，並寫入 destination 字串。</p> <p>source 字串可以為靜態字串 (如 "source")或是一維字元陣列變數 (如 source[start])。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當執行成功，success等於 true，否則等於 false。source 字串長度大於目標陣列的大小，將會回傳 false。</p>

舉例	<pre>macro_command main() char src1[20] = "aBcDe" char dest1[20] bool success1 success1 = StringToLower (src1[0], dest1[0]) // success1 = true , dest1 = “abcde”  char dest2[4] bool success2 success2 = StringToLower ("aBcDe", dest2[0]) // success2 = false , dest2 內容不變  end macro_command</pre>
----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	StringToReverse
語法	<pre>success = StringToReverse (source[start], destination[start]) success = StringToReverse ("source", destination[start])</pre>
描述	<p>將 source 字串反轉，並寫入 destination 字串。</p> <p>source 字串可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。</p> <p>執行完畢會回傳一 bool 型態的值給 success 欄位。當執行成功，success 等於 true，否則等於 false。source 字串長度大於目標陣列的大小，將會回傳 false。</p>
舉例	<pre>macro_command main() char src1[20] = "abcde" char dest1[20] bool success1 success1 = StringToReverse (src1[0], dest1[0]) // success1 = true , dest1 = “edcba”  char dest2[4] bool success2 success2 = StringToReverse ("abcde", dest2[0]) // success2 = false , dest2 內容不變  end macro_command</pre>

函數名稱	StringTrimLeft
語法	<pre>success = StringTrimLeft (source[start], set[start], destination[start]) success = StringTrimLeft ("source", set[start], destination[start]) success = StringTrimLeft (source[start], "set", destination[start]) success = StringTrimLeft ("source", "set", destination[start])</pre>
描述	<p>從 source 字串的第一個字元開始往後尋找，若找到與 set 字串相同的字元便裁剪掉該字元，直到遇到不存在於 set 字串中的字元為止。</p> <p>source 字串與 set 字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。</p>

	執行完畢會回傳一 <code>bool</code> 型態的值給 <code>success</code> 欄位。當執行成功， <code>success</code> 等於 <code>true</code> ，否則等於 <code>false</code> 。當裁剪完後的字串長度大於目標陣列的大小，將會回傳 <code>false</code> 。
舉例	<pre>macro_command main() char src1[20] = "# *a*#bc" char set1[20] = "# *" char dest1[20] bool success1 success1 = StringTrimLeft (src1[0], set1[0], dest1[0]) // success1 = true , dest1 = "a*#bc"  char set2[20] = {'#', ' ', '*'} char dest2[4] bool success2 success2 = StringTrimLeft ("# *a*#bc", set2[0], dest2[0]) // success2 = false , dest2 內容不變  char src3[20] = "abc *#" char dest3[20] bool success3 success3 = StringTrimLeft (src3[0], "# *", dest3[0]) // success3 = true , dest3 = "abc *#"  end macro_command</pre>

函數名稱	<code>StringTrimRight</code>
語法	<pre>success = StringTrimRight (source[start], set[start], destination[start]) success = StringTrimRight ("source", set[start], destination[start]) success = StringTrimRight (source[start], "set", destination[start]) success = StringTrimRight ("source", "set", destination[start])</pre>
描述	<p>從 <code>source</code> 字串的最後一個字元開始往前尋找，若找到與 <code>set</code> 字串相同的字元便裁剪掉該字元，直到遇到不存在於 <code>set</code> 字串中的字元為止。</p> <p><code>source</code> 字串與 <code>set</code> 字串皆可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 <code>source[start]</code>)。</p> <p>執行完畢會回傳一 <code>bool</code> 型態的值給 <code>success</code> 欄位。當執行成功，<code>success</code> 等於 <code>true</code>，否則等於 <code>false</code>。當裁剪完後的字串長度大於目標陣列的大小，將會回傳 <code>false</code>。</p>
舉例	<pre>macro_command main() char src1[20] = "# *a*#bc# * " char set1[20] = "# *" char dest1[20] bool success1 success1 = StringTrimRight(src1[0], set1[0], dest1[0]) // success1 = true , dest1= "# *a*#bc"  char set2[20] = {'#', ' ', '*'} char dest2[20]</pre>

	<pre> bool success2 success2 = StringTrimRight("# *a*#bc", set2[0], dest2[0]) // success2 = true , dest2 = "# *a*#bc"  char src3[20] = "ab**c *#" char dest3[4] bool success3 success3 = StringTrimRight(src3[0], "# *", dest3[0]) // success3 = false , dest3 內容不變  end macro_command </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	Unicode2Utf8
語法	<code>result = Unicode2Utf8(source[start], destination[start])</code>
描述	將source的Unicode字串轉換為Utf8字串，存放到destination。 執行完畢會回傳一 bool 型態的值給 result 欄位。當執行成功，result 等於 true，否則等於 false。
舉例	<pre> macro_command main()  char unicode_str[20] char utf8_str[20] String2Unicode("ABC", unicode_str[0]) bool result  result = Unicode2Utf8(unicode_str[0], utf8_str[0]) // "result" will be set to true. "utf8_str" will equal "ABC" encoded in UTF8 StringCat("DEF", utf8_str[0]) // "utf8_str" will equal "ABCDEF" encoded in UTF8  char dst[20] bool result2  result2 = Utf82Unicode(utf8_str[0], dst[0]) // "result" will be set to true. "dst" will equal "ABCDEF" encoded in Unicode.  end macro_command </pre>

函數名稱	UnicodeCat
語法	<code>result = UnicodeCat(source[start], destination[start])</code> 或 <code>result = UnicodeCat("source", destination[start])</code>
描述	利用此函數將來源字串銜接於目標字串之後。此函數執行成功後，目標字串將等於兩字串銜接後的結果。 來源字串source可以為靜態字串 (如 "source") 或是一維字元陣列變數 (如 source[start])。 destination[start]必須為一維字元陣列變數。

	執行完畢會回傳一 <b>bool</b> 型態的值給 <b>result</b> 欄位。當執行成功， <b>result</b> 等於 <b>true</b> ，否則等於 <b>false</b> 。當兩字串銜接後的長度超過目標陣列的長度時，目標字串將保留銜接以前的內容，不做任何改變，並回傳 <b>false</b> 。
舉例	<pre>macro_command main()  char strSrc[12]="α θ β γ θ δ" char strDest[28]="ζ η θ λ 1234" bool result  result = UnicodeCat(strSrc[0], strDest[0]) // "result" will be set to true // "strDest" will be set to "ζ η θ λ 1234 α θ β γ θ δ"  result = UnicodeCat("ζ η θ λ ", strDest[0]) // the function fails. // "result" will be set to false due to insufficient destination buffer size. // In this case, the content of "strDest" remains the same.  end macro_command</pre>

函數名稱	UnicodeCompare
語法	<pre>result = UnicodeCompare(string1[start], string2[start]) result = UnicodeCompare("string1", string2[start]) result = UnicodeCompare(string1[start], "string2") result = UnicodeCompare("string1", "string2")</pre>
描述	<p>比較兩字串的內容是否相等。此函數將大小寫視為不同。</p> <p>兩個傳入的字串皆可以為靜態字串 (如 "string")或是一維字元陣列變數 (如 <b>string[start]</b>)。</p> <p>執行完畢會回傳一 <b>bool</b>型態的值給 <b>result</b> 欄位。若兩字串相等，<b>result</b> 為 <b>true</b>，否則為 <b>false</b>。</p>
舉例	<pre>macro_command main() char str1[10]="θ α β θ γ" char str2[8]="α β γ δ" bool result  result = UnicodeCompare(str1[0], str2[0]) // "result" will be set to false. result = UnicodeCompare(str1[0], "θ α β θ γ") // "result" will be set to true.  end macro_command</pre>

函數名稱	UnicodeCopy
語法	<pre>result = UnicodeCopy("source", destination[start]) 或 result = UnicodeCopy(source[start], destination[start])</pre>
描述	<p>利用此函數進行字串的複製。此函數可以將傳入的靜態字串 (以雙引號" "括起來的字串)，或是存放在字元陣列中的字串複製到目標陣列。</p> <p>來源字串 <b>source</b>可以為靜態字串 (如 "source" )或是一維字元陣列變數 (如 <b>source[start]</b>)。</p>

	<p><code>destination[start]</code> 必須為一維字元陣列變數。</p> <p>複製完畢會回傳一 <code>bool</code> 型態的值給 <code>result</code> 欄位。當複製成功，<code>result</code>等於 <code>true</code>，否則等於 <code>false</code>。當來源字串的長度大於目標陣列的大小時，將不做任何處理，並回傳 <code>false</code> 到 <code>result</code> 欄位。</p> <p><code>result</code> 欄位可寫可不寫。</p>
舉例	<pre>macro_command main() char strSrc[14]="α β θ γ δ θ ε"    // α β θ γ δ θ ε char strDest[14] bool result  result = UnicodeCopy(strSrc[0], strDest[0]) // "result" will be set to true. result = UnicodeCopy("α β θ γ δ θ ε", strDest[0]) // "result" will be set to true, strDest = α β θ γ δ θ ε" result = UnicodeCopy("α β γ δ ε ζ α β γ δ ε ζ", strDest[0]) // "result" will be set to false. // The size of source string exceeds the size of destination string.  end macro_command</pre>

函數名稱	UnicodeExcluding
語法	<pre>result = UnicodeExcluding(source[start], set[start], destination[start]) result = UnicodeExcluding("source", set[start], destination[start]) result = UnicodeExcluding(source[start], "set", destination[start]) result = UnicodeExcluding("source", "set", destination[start])</pre>
描述	<p>提取 <code>source</code> 字串中某個以索引值 <code>0</code> 的字元 (第一個字元) 開頭的子字串，而且此子字串的每個字元必不存在於<code>set</code>字串中。此函數將會從 <code>source</code>字串的第一個字元開始尋找，直到找到存在於 <code>set</code> 字串中的字元為止。</p> <p><code>source</code> 字串與 <code>set</code> 字串皆可以為靜態字串 (如 "<code>source</code>") 或是一維字元陣列變數 (如 <code>source[start]</code>)。</p> <p>執行完畢會回傳一 <code>bool</code> 型態的值給 <code>result</code> 欄位。當執行成功，<code>result</code>等於 <code>true</code>，否則等於 <code>false</code>。當提取出來的字串長度大於目標陣列的大小，將會回傳 <code>false</code>。</p>
舉例	<pre>macro_command main() char source[14]="γ δ ξ κ θ λ θ , dest[8] char set[4]="λ θ " bool result  result = UnicodeExcluding(source[0], set[0], dest[0]) // the function succeeds. // "result" will be set to true and "dest" will be set to "γ δ ξ κ".  result = UnicodeExcluding(source[0], set[0], dest[4]) // the function fails. // "result" will be set to false due to insufficient destination buffer size.  end macro_command</pre>

函數名稱	UnicodeLength
------	---------------

語法	<code>result = UnicodeLength(source[start])</code> 或 <code>result = UnicodeLength("source")</code>
描述	取得 Unicode 字串長度。 來源字串 source 可以為靜態字串 (如 "source") 或是一維字元陣列變數(如 source[start])。 函數回傳值代表來源 Unicode 字串的長度。
舉例	<pre>macro_command main()      char strSrc[6] = "ABC"     int result1, result2      result1 = UnicodeLength(strSrc[0]) // "result1" is equal to 3     result2 = UnicodeLength("ABC") // "result2" is equal to 3  end macro_command</pre>

函數名稱	<code>Utf82Unicode</code>
語法	<code>result = Utf82Unicode(source[start], destination[start])</code>
描述	將source的Utf8字串轉換為Unicode字串，存放到destination。 執行完畢會回傳一 bool 型態的值給 result 欄位。當執行成功，result 等於 true，否則等於 false。
舉例	<pre>macro_command main()      char unicode_str[20]     char utf8_str[20]     String2Unicode("ABC", unicode_str[0])     bool result      result = Unicode2Utf8(unicode_str[0], utf8_str[0])     // "result" will be set to true. "utf8_str" will equal "ABC" encoded in UTF8     StringCat("DEF", utf8_str[0]) // "utf8_str" will equal "ABCDEF" encoded in UTF8      char dst[20]     bool result2      result2 = Utf82Unicode(utf8_str[0], dst[0])     // "result" will be set to true. "dst" will equal "ABCDEF" encoded in Unicode.  end macro_command</pre>



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 18.7.8. 數學運算

函數名稱	SQRT
語法	SQRT(source, result)
描述	開平方根。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。資料來源必須為一個正數。
舉例	<pre>macro_command main() float source, result  SQRT(15, result)  source = 9.0 SQRT(source, result)// 執行後 result = 3.0  end macro_command</pre>

函數名稱	CUBERT
語法	CUBERT (source, result)
描述	開三次方根。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。資料來源必須為一個正數。
舉例	<pre>macro_command main() float source, result  CUBERT (27, result) // 執行後 result = 3.0  source = 27.0 CUBERT (source, result) // 執行後 result = 3.0  end macro_command</pre>

函數名稱	POW
語法	POW (source1, source2, result)
描述	計算 source1 的某次方 (source2)。資料來源 source1 和 source2 可以是常數或者變數，但是存放結果的 result 必須為變數。資料來源必須為一個正數。
舉例	<pre>macro_command main() float y, result y = 0.5 POW (25, y, result) // 執行後 result = 5 end macro_command</pre>

函數名稱	SIN
語法	SIN(source, result)
描述	三角函數的正弦計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	macro_command main()

	<pre>float source, result SIN(90, result) // result is 1  source = 30 SIN(source, result) // result is 0.5  end macro_command</pre>
--	-------------------------------------------------------------------------------------------------------------------------------------

函數名稱	COS
語法	COS(source, result)
描述	三角函數的餘弦計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source, result  COS(90, result) // result is 0  source = 60 COS(source, result) // result is 0.5  end macro_command</pre>

函數名稱	TAN
語法	TAN(source, result)
描述	三角函數的正切計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source, result  TAN(45, result) // result is 1  source = 60 TAN(source, result) // result is 1.732  end macro_command</pre>

函數名稱	COT
語法	COT(source, result)
描述	三角函數的餘切計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source, result  COT(45, result) // result is 1</pre>

	<pre>source = 60 COT(source, result) // result is 0.5774  end macro_command</pre>
--	-----------------------------------------------------------------------------------

函數名稱	SEC
語法	SEC(source, result)
描述	三角函數的正割計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source, result  SEC(45, result) // result is 1.414  source = 60 SEC(source, result) // if source is 60, result is 2  end macro_command</pre>

函數名稱	CSC
語法	CSC(source, result)
描述	三角函的餘割計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source, result  CSC(45, result) // result is 1.414  source = 30 CSC(source, result) // result is 2  end macro_command</pre>

函數名稱	ASIN
語法	ASIN(source, result)
描述	反三角函數中反正弦計算。資料來源 source 可以是常數或者變數，但是存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source, result  ASIN(0.8660, result) // result is 60  source = 0.5 ASIN(source, result) // result is 30</pre>

	<code>end macro_command</code>
--	--------------------------------

函數名稱	ACOS
語法	<code>ACOS(source, result)</code>
描述	反三角函數中反餘弦計算。資料來源 <code>source</code> 可以是常數或者變數，但是存放結果的 <code>result</code> 必須為變數。
舉例	<pre>macro_command main() float source, result  ACOS(0.8660, result) // result is 30  source = 0.5 ACOS(source, result) // result is 60  end macro_command</pre>

函數名稱	ATAN
語法	<code>ATAN(source, result)</code>
描述	反三角函數中反正切計算。資料來源 <code>source</code> 可以是常數或者變數，但是存放結果的 <code>result</code> 必須為變數。
舉例	<pre>macro_command main() float source, result  ATAN(1, result) // result is 45  source = 1.732 ATAN(source, result) // result is 60  end macro_command</pre>

函數名稱	LOG
語法	<code>LOG (source, result)</code>
描述	取得 <code>source</code> 的自然對數，並存入 <code>result</code> 變數。 <code>source</code> 可為變數或常數。 存放結果的 <code>result</code> 必須為變數。
舉例	<pre>macro_command main() float source=100, result  LOG (source, result) // result 約等於 4.6052 end macro_command</pre>

函數名稱	LOG10
語法	<code>LOG10 (source, result)</code>
描述	取得 <code>source</code> 的以 10 為基底的對數，並存入 <code>result</code> 變數。 <code>source</code> 可為變數或常數。

	存放結果的 result 必須為變數。
舉例	<pre>macro_command main() float source=100, result  LOG10 (source, result) // result 等於 2 end macro_command</pre>

函數名稱	RAND
語法	RAND(result)
描述	產生一個隨機數。(範圍: 0 ~ 32766) 存放結果的 result 必須為變數。
舉例	<pre>macro_command main() short result  RAND (result) // result is not a fixed value when executes macro every time  end macro_command</pre>

函數名稱	CEIL
語法	result=CEIL(source)
描述	取得不小於 source 的最小整數值。
舉例	<pre>macro_command main()  float x = 3.8 int result  result = CEIL(x) // result = 4  end macro_command</pre>

函數名稱	FLOOR
語法	result=FLOOR(source)
描述	取得不大於 source 的最大整數值。
舉例	<pre>macro_command main()  float x = 3.8 int result  result = FLOOR(x) // result = 3  end macro_command</pre>

函數名稱	ROUND
語法	result=ROUND(source)
描述	取得最接近 source 的整數數值。

舉例	macro_command main()  float x = 5.55 int result  result = ROUND(x) // result = 6  end macro_command
----	--------------------------------------------------------------------------------------------------------------------------



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 18.7.9. 統計

函數名稱	AVERAGE
語法	AVERAGE(source[start], result, count)
描述	從陣列中計算平均值
舉例	<pre>int data[5] = {1, 2, 3, 4, 5} float result  AVERAGE(data[0], result, 5)      // result 等於3 AVERAGE(data[2], result, 3)      // result 等於4</pre>

函數名稱	HARMEAN
語法	HARMEAN(source[start], result, count)
描述	從陣列中計算調和平均值
舉例	<pre>int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result  HARMEAN(data[0], result, 10) // result 等於3.414</pre>

函數名稱	MAX
語法	MAX(source[start], result, count)
描述	從陣列中取最大值
舉例	<pre>int data[5] = {1, 2, 3, 4, 5} int result  MAX(data[0], result, 5) // result 等於5 MAX(data[1], result, 3) // result 等於4</pre>

函數名稱	MEDIAN
語法	MEDIAN(source[start], result, count)
描述	從陣列中取中位數
舉例	<pre>int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result  MEDIAN(data[0], result, 10) // result 等於 5.5</pre>

函數名稱	MIN
語法	MIN(source[start], result, count)
描述	從陣列中取最小值
舉例	<pre>int data[5] = {1, 2, 3, 4, 5} int result  MIN(data[0], result, 5)</pre>

	MIN(data[0], result, 5) // result 等於 1 MIN(data[1], result, 3) // result 等於 2
--	----------------------------------------------------------------------------------

函數名稱	STDEVP
語法	STDEVP(source[start], result, count)
描述	從陣列中計算標準差
舉例	<pre>int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result</pre> <pre>STDEVP(data[0], result, 10) // result 等於2.872</pre>

函數名稱	STDEVS
語法	STDEVS(source[start], result, count)
描述	從陣列中計算樣本標準差
舉例	<pre>int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result</pre> <pre>STDEVS(data[0], result, 10) // result 等於 3.027</pre>



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 18.7.10. 配方資料庫

函數名稱	RecipeGetData
語法	RecipeGetData (destination, recipe_address, record_ID)
描述	取得配方中的資料。所取得的資料存放在destination且必須為變數。 recipe_address由配方名稱與項目名稱組成："recipe_name.item_name"。 record_ID指定欲取得配方中的第幾筆資料，也就是資料列編號。
舉例	<pre>macro_command main() int data = 0 char str[20] int recordID bool result</pre> <pre>recordID = 0 result = RecipeGetData(data, "TypeA.item_weight", recordID) // 從配方 "TypeA" 中取得第0筆，且 item name為 "item_weight" 的資料</pre> <pre>recordID = 1 result = RecipeGetData(str[0], "TypeB.item_name", recordID) // 從配方 "TypeB" 中取得第1筆，且 item name為 "item_name" 的資料</pre> <pre>end macro_command</pre>

函數名稱	RecipeQuery
語法	RecipeQuery (SQL command, destination)
描述	<p>透過SQL語句，對配方中的資料進行查詢。查詢結果的資料筆數存放在 destination，必須為變數。SQL command 的部份可為靜態文字或傳入字元陣列，例如：</p> <pre>RecipeQuery("SELECT * FROM TypeA", destination)</pre> <p>或</p> <pre>RecipeQuery(sql[0], destination)</pre> <p>SQL 語句必須以 "SELECT * FROM" 開頭，後面接配方名稱及查詢條件。</p>
舉例	<pre>macro_command main()  int total_row = 0 char sql[100] = "SELECT * FROM TypeB" short var bool result  result = RecipeQuery("SELECT * FROM TypeA", total_row) // 查詢配方 "TypeA"。查詢結果的資料筆數存放在 total_row.  result = RecipeQuery(sql[0], total_row) // 查詢配方 "TypeB"。查詢結果的資料筆數存放在 total_row.  result = RecipeQuery("SELECT * FROM Recipe WHERE Item &gt;%(var)", total_row) // 查詢配方 "Recipe" 裡的 "Item" 欄位數據大於 var 的資料筆數。查詢結果的資料筆數存放在 total_row.  end macro_command</pre>

函數名稱	RecipeQueryGetData
語法	RecipeQueryGetData (destination, recipe_address, result_row_no)
描述	<p>取得 RecipeQuery 的查詢結果。呼叫此函數前必須先呼叫 RecipeQuery，且 recipe_address 中需指定與 RecipeQuery 相同的配方名稱。</p> <p>result_row_no 指定欲取得查詢結果中的第幾筆資料。</p>
舉例	<pre>macro_command main()  int data = 0 int total_row = 0 int row_number = 0 bool result_query bool result_data  result_query = RecipeQuery("SELECT * FROM TypeA", total_row) // 查詢配方 "TypeA"。查詢結果的資料筆數存放在 total_row.  if (result_query) then</pre>

	<pre> for row_number = 0 to total_row - 1 result_data = RecipeQueryGetData(data, "TypeA.item_weight", row_number) next row_number end if  end macro_command </pre>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	RecipeQueryGetRecordID
語法	RecipeQueryGetRecordID (destination, result_row_no)
描述	<p>取得 RecipeQuery 查詢結果的資料列編號。呼叫此函數前必須先呼叫 RecipeQuery。</p> <p>result_row_no 指定欲取得查詢結果中的第幾筆資料的資料列編號，並將資料列編號寫入 destination。</p>
舉例	<pre> macro_command main()  int recordID = 0 int total_row = 0 int row_number = 0 bool result_query bool result_id  result_query = RecipeQuery("SELECT * FROM TypeA", total_row) // 查詢配方 "TypeA"。查詢結果的資料筆數存放在 total_row.  if (result_query) then for row_number = 0 to total_row - 1 result_id = RecipeQueryGetRecordID(recordID, row_number) next row_number end if  end macro_command </pre>

函數名稱	RecipeSetData
語法	RecipeSetData(source, recipe_address, record_ID)
描述	<p>將資料寫入配方資料庫中。如果成功將傳回 True，否則將傳回 False。</p> <p>recipe_address 由配方名稱與項目名稱組成："recipe_name.item_name"。</p> <p>record_ID 指定欲修改配方中的第幾筆資料，也就是資料列編號。</p>
舉例	<pre> macro_command main()  int data=99 char str[20]="abc" int recordID bool result  recordID = 0 result = RecipeSetData(data, "TypeA.item_weight", recordID) </pre>

	<pre>// set data to recipe "TypeA", where item name is "item_weight" and the record ID is 0.  recordID = 1 result = RecipeSetData(str[0], "TypeB.item_name", recordID) // set data to recipe "TypeB", where item name is "item_name" and the record ID is 1.  end macro_command</pre>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	RecipeTransactionBegin
語法	RecipeTransactionBegin ()
描述	<p>開啟批次寫入配方的功能。需配合RecipeTransactionCommit或RecipeTransactionRollback進行使用。</p> <p>RecipeTransactionBegin 到 RecipeTransactionCommit之間的所有Recipe寫入動作將會等到Commit命令後一次執行。</p> <p>RecipeTransactionBegin 到 RecipeTransactionRollback之間的所有Recipe寫入動作將會等到Rollback命令後全數取消。</p> <p><b>警告提示</b></p> <p>若在巨集結束前，沒有呼叫RecipeTransactionCommit或RecipeTransactionRollback，則系統將會自動呼叫RecipeTransactionRollback進行寫入取消，並在cMT診斷器上出現以下警示：DB Transaction ended without commit/rollback, and rolled back all changes automatically.</p> <p>若重複呼叫RecipeTransactionBegin()，則會在cMT診斷器上出現以下警示：Cannot start a transaction within a transaction.</p> <p><b>注意</b></p> <p>當使用RecipeTransactionBegin時，請盡量縮短RecipeTransactionBegin至RecipeTransactionCommit/RecipeTransactionRollback的執行時間，避免其他物件同時操作配方資料庫造成系統異常。</p>
舉例	<pre>macro_command main() int data=99 char str[20]="abc" int recordID = 0 bool result result = RecipeSetData(data, "TypeA.item_weight", recordID) //將資料寫入配方 “TypeA” 其中的欄位 “item_weight” ，Record ID為0。 RecipeTransactionBegin() recordID = 1 result = RecipeSetData(str[0], "TypeB.item_name", recordID) RecipeTransactionCommit() //將資料寫入配方 “TypeB” 其中的欄位 “item_name” ，Record ID為1。</pre>

	<pre> RecipeTransactionBegin() recordID = 2 result = RecipeSetData(str[0], "TypeB.item_name", recordID) RecipeTransactionRollback() //因為取消批次寫入配方，所以Record ID仍為1。 end macro_command </pre>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

函數名稱	RecipeTransactionCommit
語法	RecipeTransactionCommit ()
描述	<p>執行批次寫入配方。需配合RecipeTransactionBegin使用。</p> <p>RecipeTransactionBegin 到 RecipeTransactionCommit之間的所有Recipe寫入動作將會等到Commit命令後一次執行。</p> <p><b>警告提示</b></p> <p>若在沒有呼叫RecipeTransactionBegin就先呼叫RecipeTransactionCommit，則系統將會在cMT診斷器上出現以下警示：Cannot commit - no transaction is active.</p>
舉例	請參考RecipeTransactionBegin範例。

函數名稱	RecipeTransactionRollback
語法	RecipeTransactionRollback ()
描述	<p>取消執行批次寫入配方。需配合RecipeTransactionBegin使用。</p> <p>RecipeTransactionBegin 到 RecipeTransactionRollback之間的所有Recipe寫入動作將會等到Rollback命令後全數取消。</p> <p><b>警告提示</b></p> <p>若在沒有呼叫RecipeTransactionBegin就先呼叫RecipeTransactionRollback，則系統將會在cMT診斷器上出現以下警示：Cannot rollback - no transaction is active.</p>
舉例	請參考RecipeTransactionBegin範例。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 18.7.11. 資料取樣/事件記錄

函 數 名 稱	FindDataSamplingDate
語 法	<pre> return_value = FindDataSamplingDate (data_log_number, index, year, month, day) or FindDataSamplingDate (data_log_number, index, year, month, day) </pre>
描 述	利用此函數查詢資料取樣檔的日期。根據所輸入的資料取樣編號 (data_log_number) 與資料取樣檔案索引 (index) 可查詢該資料取樣的日期，依照年、月、日的順序寫入 year、month、day 的欄位中。

	 <p>資料取樣檔的儲存方式為：[保存位置]\[取樣資料夾名稱]\yyymmdd.dtl。而資料取樣檔案索引 (index) 指的是取樣資料夾下面所有資料取樣檔案依檔名排序後的順位值 (從0開始)。Index 值越小者，日期越新。例如假設某取樣資料夾下面有四個資料取樣檔：</p> <p>20101210.dtl 20101230.dtl 20110110.dtl 20110111.dtl</p> <p>則index依序為：</p> <p>20101210.dtl -&gt; index 為 3 20101230.dtl -&gt; index 為 2 20110110.dtl -&gt; index 為 1 20110111.dtl -&gt; index 為 0</p> <p>當成功搜尋到所欲查詢的資料取樣檔時，將回傳 1 至 return_value，否則將回傳 0。</p> <p>data_log_number 與 index 可以為常數或是變數，但 year、month、day 與 return_value 必須為變數。 return_value 為可選。</p>
舉例	<pre>macro_command main() short data_log_number = 1, index = 2, year, month, day short success  // 若存在一資料取樣檔 20101230.dtl，其資料取樣編號為 1，檔案索引為 2 // 則success == 1, year == 2010, month == 12, day == 30 success = FindDataSamplingDate(data_log_number, index, year, month, day)  end macro_command</pre>



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

函數名稱	FindDataSamplingIndex
語法	<pre>return_value = FindDataSamplingIndex (data_log_number, year, month, day, index) or FindDataSamplingIndex (data_log_number, year, month, day, index)</pre>
描述	利用此函數查詢資料取樣檔的檔案索引值。根據所輸入的資料取樣編號 (data_log_number) 與日期可查詢該資料取樣檔的檔案索引，並將其寫入 index。year、month、day 三個欄位依序代表年、月、日，其輸入格式為 YYYY(年)、MM(月)、DD(日)。

資料取樣							
編號	描述	讀取位址	取樣模式	觸發位址	清除控制位址	暫停控制位址	自動停止
1		Local HMI : LW0	週期模式	停用	停用	停用	停用
2		Local HMI : LW100	週期模式	停用	Local HMI : LB0	Local HMI : LB0	停用

資料取樣檔的儲存方式為：[保存位置]\[取樣資料夾名稱]\yyymmdd.dtl。而資料取樣檔案索引 (index) 指的是取樣資料夾下面所有資料取樣檔案依檔名排序後的順位值 (從 0 開始)。Index 值越小者，日期越新。例如假設某取樣資料夾下面有四個資料取樣檔：

20101210.dtl

20101230.dtl

20110110.dtl

20110111.dtl

則 index 依序為：

20101210.dtl -> index 為 3

20101230.dtl -> index 為 2

20110110.dtl -> index 為 1

20110111.dtl -> index 為 0

當成功搜尋到所欲查詢的資料取樣檔時，將回傳 1 至 return\_value，否則將回傳 0。

data\_log\_number 與 year、month、day 可以為常數或是變數，但 index 與 return\_value 必須為變數。

return\_value 為可選。

### 舉例

```
macro_command main()
short data_log_number = 1, year = 2010, month = 12, day = 10, index
short success

// 若存在一資料取樣檔 20101210.dtl，其資料取樣編號為 1，檔案索引為 2
// 則success == 1，index == 2
success = FindDataSamplingIndex (data_log_number, year, month, day, index)

end macro_command
```



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

函數名稱	FindEventLogDate
語法	return_value = FindEventLogDate (index, year, month, day) or FindEventLogDate (index, year, month, day)
描述	利用此函數查詢事件登錄檔的日期。根據所輸入的事件登錄檔案索引 (index) 可查詢該事件登錄的日期，依照年、月、日的順序寫入 year、month、day 的欄位中。

	<p>事件登錄檔案索引 (index) 指的是指定的保存位置 (HMI、SD 卡或 USB) 下事件登錄檔案依檔名排序後的順位值 (從 0 開始)。Index 值越小者，日期越新。例如假設有四個事件登錄檔：</p> <p>EL_20101210.evt EL_20101230.evt EL_20110110.evt EL_20110111.evt</p> <p>則 index 依序為：</p> <p>EL_20101210.evt -&gt; index 為 3 EL_20101230.evt -&gt; index 為 2 EL_20110110.evt -&gt; index 為 1 EL_20110111.evt -&gt; index 為 0</p> <p>當成功搜尋到所欲查詢的事件登錄檔時，將回傳 1 至 return_value，否則將回傳 0。</p> <p>Index 可以為常數或是變數，但 year、month、day 與 return_value 必須為變數。</p> <p>return_value 為可選。</p>
舉例	<pre>macro_command main() short index = 1, year, month, day short success  // 若存在一事件登錄檔 EL_20101230.evt，檔案索引為 1 // 則 success == 1, year == 2010, month == 12, day == 30 success = FindEventLogDate (index, year, month, day)  end macro_command</pre>



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

函數名稱	FindEventLogIndex
語法	<pre>return_value = FindEventLogIndex (year, month, day, index) or FindEventLogIndex (year, month, day, index)</pre>
描述	<p>利用此函數查詢事件登錄檔的檔案索引值。根據所輸入的日期可查詢該事件登錄檔的檔案索引，並將其寫入 index。year、month、day 三個欄位依序代表年、月、日，其輸入格式為 YYYY (年)、MM (月)、DD (日)。</p> <p>事件登錄檔案索引 (index) 指的是指定的保存位置 (HMI、SD 卡或 USB) 下事件登錄檔案依檔名排序後的順位值 (從 0 開始)。Index 值越小者，日期越新。例如假設有四個事件登錄檔：</p> <p>EL_20101210.evt EL_20101230.evt EL_20110110.evt EL_20110111.evt</p> <p>則 index 依序為：</p> <p>EL_20101210.evt -&gt; index 為 3</p>

	<p>EL_20101230.evt -&gt; index 為 2      EL_20110110.evt -&gt; index 為 1      EL_20110111.dtl -&gt; index 為 0</p> <p>當成功搜尋到所欲查詢的事件登錄檔時，將回傳 1 至 return_value，否則將回傳 0。</p> <p>year、month、day 可以為常數或是變數，但 index 與 return_value 必須為變數。</p> <p>return_value 為可選。</p>
舉例	<pre>macro_command main() short year = 2010, month = 12, day = 10, index short success  // 若存在一事件登錄檔 EL_20101210.evt，檔案索引為 2 // 則success == 1，index == 2 success = FindEventLogIndex (year, month, day, index)  end macro_command</pre>



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 18.7.12. 校驗

函數名稱	ADDSUM
語法	ADDSUM(source[start], result, data_count)
描述	<p>將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料累加起來，以獲得 checksum (校驗和)，並將結果存放在 result 變數中。</p> <p>result 必須為變數，data_count 是進行累加的資料的個數，可以是常數或者是變數。</p>
舉例	<pre>macro_command main() char data[5] short checksum  data[0] = 0x1 data[1] = 0x2 data[2] = 0x3 data[3] = 0x4 data[4] = 0x5 ADDSUM(data[0], checksum, 5) // checksum is 0xf  end macro_command</pre>

函數名稱	XORSUM
語法	XORSUM(source[start], result, data_count)
描述	<p>將 source[start] 到 source[start+data_count-1] 的所有一維陣列的資料進行異或運算，以獲得 checksum (校驗和)，並將結果存放在 result 變數中。</p> <p>result 必須為變數，data_count 是進行異或計算的資料的個數，可以是常數或者是</p>

	變數。
舉例	<pre>macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} short checksum  XORSUM(data[0], checksum, 5) // checksum is 0x1  end macro_command</pre>

函數名稱	BCC
語法	BCC(source[start], result, data_count)
描述	等同 XORSUM 函數。
舉例	<pre>macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} char checksum  BCC(data[0], checksum, 5) // checksum is 0x1  end macro_command</pre>

函數名稱	CRC
語法	CRC(source[start], result, data_count)
描述	將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料進 16-bit CRC 計算，以獲得 checksum (校驗和)，並將結果存放在 result 變數中。 result 必須為變數，data_count 是進行計算的資料的個數，可以是常數或者是變數。
舉例	<pre>macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} short checksum  CRC(data[0], checksum, 5) // checksum is 0xbb2a, 16-bit CRC  end macro_command</pre>

函數名稱	CRC8
語法	CRC8(source[start], result, data_count)
描述	將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料進 8-bit CRC 計算，以獲得 checksum (校驗和)，並將結果存放在 result 變數中。 result 必須為變數，data_count 是進行計算的資料的個數，可以是常數或者是變數。
舉例	<pre>macro_command main() char source[5] = {1, 2, 3, 4, 5} short CRC8_result  CRC8(source[0], CRC8_result, 5) // CRC8_result = 188  end macro_command</pre>

函數名稱	CRC16_CCITT
語法	CRC16_CCITT(source[start], result, data_count)
描述	將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料進 16-bit CRC 計算，以獲得 CRC-16/CCITT 的 checksum (校驗和)，並將結果存放在 result 變數中。result 必須為變數，data_count 是進行計算的資料的個數，可以是常數或者是變數。
舉例	<pre>macro_command main()      char source[5] = "12345"     short crc_result     CRC16_CCITT(source[0], crc_result, 5) //crc_result = 0xA5A2  end macro_command</pre>

函數名稱	CRC16_CCITT_FALSE
語法	CRC16_CCITT_FALSE(source[start], result, data_count)
描述	將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料進 16-bit CRC 計算，以獲得 CRC-16/CCITT-FALSE 的 checksum (校驗和)，並將結果存放在 result 變數中。result 必須為變數，data_count 是進行計算的資料的個數，可以是常數或者是變數。
舉例	<pre>macro_command main()      char source[5] = "12345"     short crc_result     CRC16_CCITT_FALSE(source[0], crc_result, 5) //crc_result = 0x4560  end macro_command</pre>

函數名稱	CRC16_X25
語法	CRC16_X25(source[start], result, data_count)
描述	將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料進 16-bit CRC 計算，以獲得 CRC-16/X25 的 checksum (校驗和)，並將結果存放在 result 變數中。result 必須為變數，data_count 是進行計算的資料的個數，可以是常數或者是變數。
舉例	<pre>macro_command main()      char source[5] = "12345"     short crc_result     CRC16_X25(source[0], crc_result, 5) //crc_result = 0xBB40  end macro_command</pre>

函數名稱	CRC16_XMODEM
語法	CRC16_XMODEM(source[start], result, data_count)
描述	將 source[start] 到 source[start+data-count-1] 的所有一維陣列的資料進 16-bit CRC 計算，以獲得 CRC-16/XMODEM 的 checksum (校驗和)，並將結果存放在 result 變數

	<p>中。</p> <p>result 必須為變數, data_count 是進行計算的資料的個數, 可以是常數或者是變數。</p>
舉例	<pre>macro_command main()      char source[5] = "12345"     short crc_result     CRC16_XMODEM(source[0], crc_result, 5) //crc_result = 0x546C  end macro_command</pre>

### 18.7.13. 其他

函數名稱	Beep
語法	Beep ()
描述	<p>發出系統警笛音。</p> <p>此函數可發出 800 赫茲，30 毫秒的系統警笛音。</p>
舉例	<pre>macro_command main()      Beep()  end macro_command</pre>

函數名稱	Buzzer
語法	Buzzer (state)
描述	開啟 / 關閉 蜂鳴器
舉例	<pre>macro_command main()      char on = 1, off = 0     Buzzer(on) // turn on the buzzer      DELAY(1000) // delay 1 second      Buzzer(off) // turn off the buzzer      DELAY(500) // delay 500ms      Buzzer(1) // turn on the buzzer      DELAY(1000) // delay 1 second      Buzzer(0) // turn off the buzzer  end macro_command</pre>

函數名稱	TRACE
語法	TRACE(format, argument)

<b>描述</b>	<p>一個執行中的 巨集指令可以使用此函數，監視變數內容的變化，並列印字串，以協助除錯。使用者應開啟 EasyDiagnoser/cMT Diagnoser 觀看此函數的輸出結果。</p> <p>當 TRACE 函數抓取一個%開頭的特殊字元，將同時從 argument 抓取一個參數做格式化後輸出。</p> <p>Format 代表列印格式，支援 % 開頭的特殊字元。特殊字元格式如下，其中方括號內的欄位為可選，紅色文字欄位為必需：</p> <p><b>%[flags] [width] [.precision] type</b></p> <p>每個欄位的意義如下所述：</p> <p><b>flags (可選) :</b></p> <ul style="list-style-type: none"> <li>- : 靠左對齊。指定寬度後，不足的左邊補空格</li> <li>+ : 輸出正負號</li> </ul> <p><b>width (可選) :</b></p> <p>十進位正整數，指定應預留的字元寬度，不足部份補空白字元。</p> <p><b>precision (可選) :</b></p> <p>十進位正整數，指定精確度，以及輸出字元數。</p> <p><b>type :</b></p> <table border="0" style="width: 100%;"> <tr> <td>C 或 c :</td> <td>以字元方式輸出</td> </tr> <tr> <td>d :</td> <td>以 signed 十進位整數輸出</td> </tr> <tr> <td>i :</td> <td>以 signed 十進位整數輸出</td> </tr> <tr> <td>o :</td> <td>以 unsigned 八進位整數輸出</td> </tr> <tr> <td>u :</td> <td>以 unsigned 十進位整數輸出</td> </tr> <tr> <td>X 或 x :</td> <td>以 unsigned 十六進位整數輸出</td> </tr> <tr> <td>lld :</td> <td>以長整數 (64bit) signed 變數輸出 (只支援於cMT / cMT X)</td> </tr> <tr> <td>llu :</td> <td>以長整數 (64bit) unsigned 變數輸出 (只支援於cMT / cMT X)</td> </tr> <tr> <td>f :</td> <td>以單倍精確度浮點數輸出。</td> </tr> <tr> <td>lf :</td> <td>以雙倍精確度浮點數輸出。</td> </tr> <tr> <td>E 或 e :</td> <td>以科學表示法輸出。格式為 “[ - ] dddd e[sign]ddd” 。</td> </tr> </table> <p>其中欄位d是十進位數字，欄位ddd是一至多個十進位數字，欄位ddd必須是三個十進位數字。sign是+或-。</p> <p><i>Format</i> 字串最長支援 256 個字元，多出的字元將被忽略。</p> <p><i>Argument</i> 部份可寫可不寫。但一個特殊字元應搭配一個變數。</p>	C 或 c :	以字元方式輸出	d :	以 signed 十進位整數輸出	i :	以 signed 十進位整數輸出	o :	以 unsigned 八進位整數輸出	u :	以 unsigned 十進位整數輸出	X 或 x :	以 unsigned 十六進位整數輸出	lld :	以長整數 (64bit) signed 變數輸出 (只支援於cMT / cMT X)	llu :	以長整數 (64bit) unsigned 變數輸出 (只支援於cMT / cMT X)	f :	以單倍精確度浮點數輸出。	lf :	以雙倍精確度浮點數輸出。	E 或 e :	以科學表示法輸出。格式為 “[ - ] dddd e[sign]ddd” 。
C 或 c :	以字元方式輸出																						
d :	以 signed 十進位整數輸出																						
i :	以 signed 十進位整數輸出																						
o :	以 unsigned 八進位整數輸出																						
u :	以 unsigned 十進位整數輸出																						
X 或 x :	以 unsigned 十六進位整數輸出																						
lld :	以長整數 (64bit) signed 變數輸出 (只支援於cMT / cMT X)																						
llu :	以長整數 (64bit) unsigned 變數輸出 (只支援於cMT / cMT X)																						
f :	以單倍精確度浮點數輸出。																						
lf :	以雙倍精確度浮點數輸出。																						
E 或 e :	以科學表示法輸出。格式為 “[ - ] dddd e[sign]ddd” 。																						
<b>舉例</b>	<pre>macro_command main() char c1 = 'a' short s1 = 32767 float f1 = 1.234567  TRACE("The results are") // 輸出 : The results are TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1) // 輸出 : c1 = a, s1 = 32767, f1 = 1.234567  end macro_command</pre>																						



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

函數名稱	GetCnvTagArrayIndex
語法	GetCnvTagArrayIndex(array_index)
描述	使用者定義標籤使用 [讀取轉換] 並為陣列時，在巨集副函數中，使用此函數可取得陣列的索引。 當資料取樣使用 [讀取轉換] 時，可用來處理陣列索引的數據後再取樣。
舉例	Sub short newfun(short param) Int index GetCnvTagArrayIndex(index) //如果index的數據是2，表示將使用 [讀取轉換] 陣列中的第三個資料 return param end sub

## 18.8. 怎樣建立和執行巨集指令

### 18.8.1. 怎樣建立一個巨集指令

按照以下步驟以建立一個巨集指令。

- 點選 [工程檔案] 上的巨集指令圖示  打開巨集指令管理對話窗。

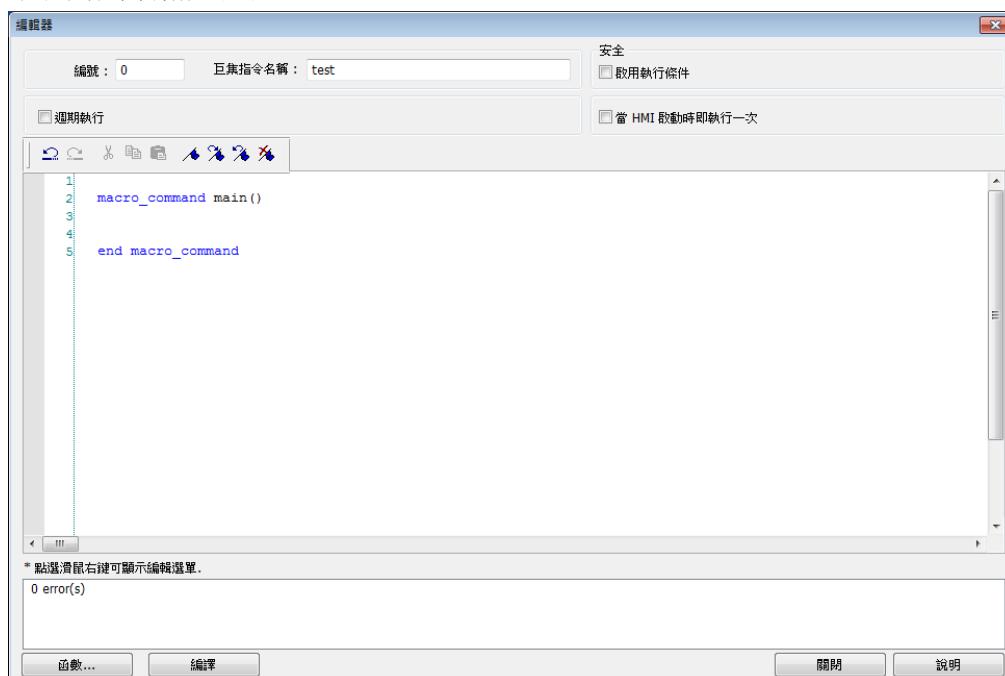


在巨集指令管理對話方塊中，已經編譯成功的巨集指令會出現在 [已編譯成功] 列表上，未完成編譯的會出現在 [未完成編譯] 列表中。下面是巨集指令管理對話方塊中各按鍵的功能描述。

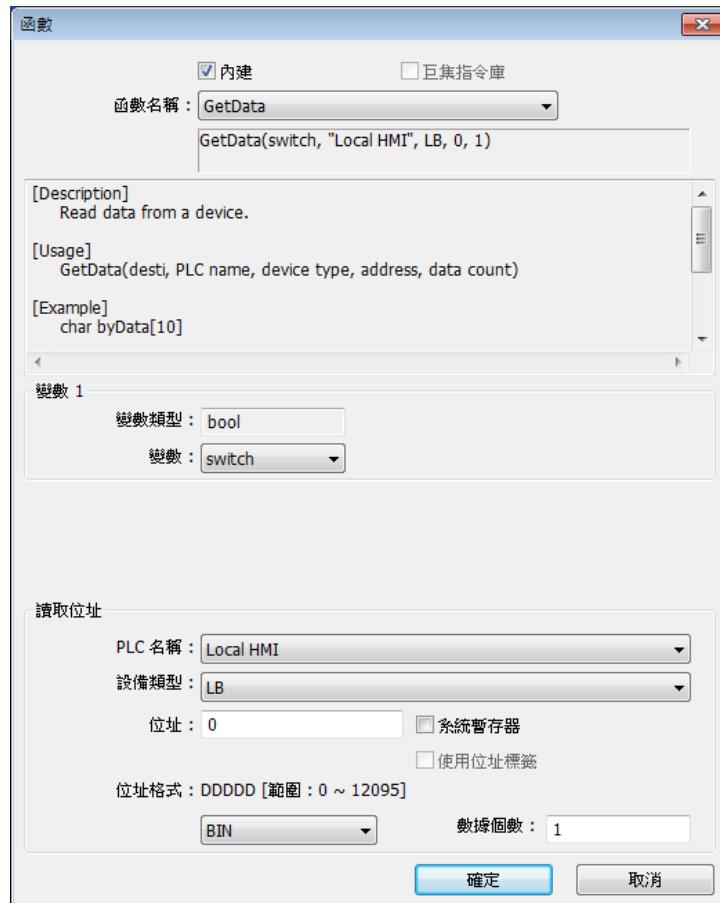
設定	描述
新增	新增一個巨集指令，並開啟新建的巨集指令編輯區。
刪除	刪除選擇的巨集指令。
編輯	打開巨集指令編輯區，並開啟選擇的巨集指令。
複製	複製選擇的巨集指令。
貼上	將剛剛選擇需要複製的巨集指令，貼至“已編譯完成”列表區，並產生一個新的巨集指令名稱。

<b>匯出</b>	將勾選的巨集指令儲存為*.ebm 檔案。
<b>匯入</b>	將*.ebm 檔案匯入至此工程檔案。
<b>確認</b>	確認此次所編輯的所有巨集指令後離開此巨集指令管理對話方塊，必須按此鍵才會儲存 此次編輯內容。
<b>取消</b>	取消此次所編輯的所有巨集指令，離開巨集指令管理對話方塊。
<b>巨集指令庫</b>	進入巨集函數庫管理對話方塊。

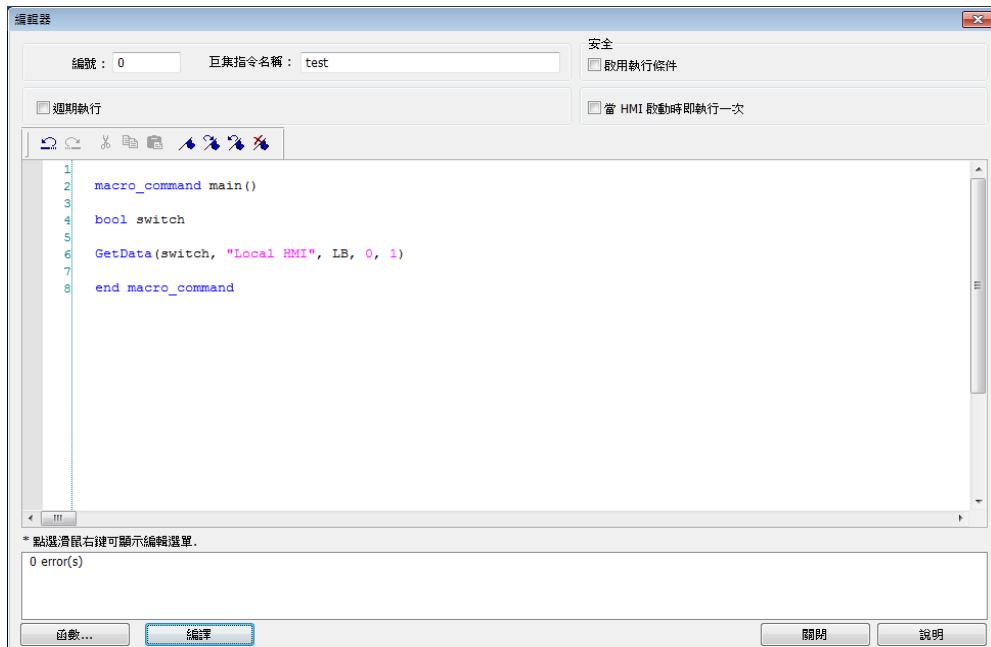
2. 按下 [新增] 按鈕，打開一個新增的巨集指令編輯區。每一個巨集指令都有一個唯一的編號，定義在 [編號] 這個位置。在 [巨集指令名稱] 這個欄目中也必須輸入巨集指令的名稱，否則編譯將無法通過。



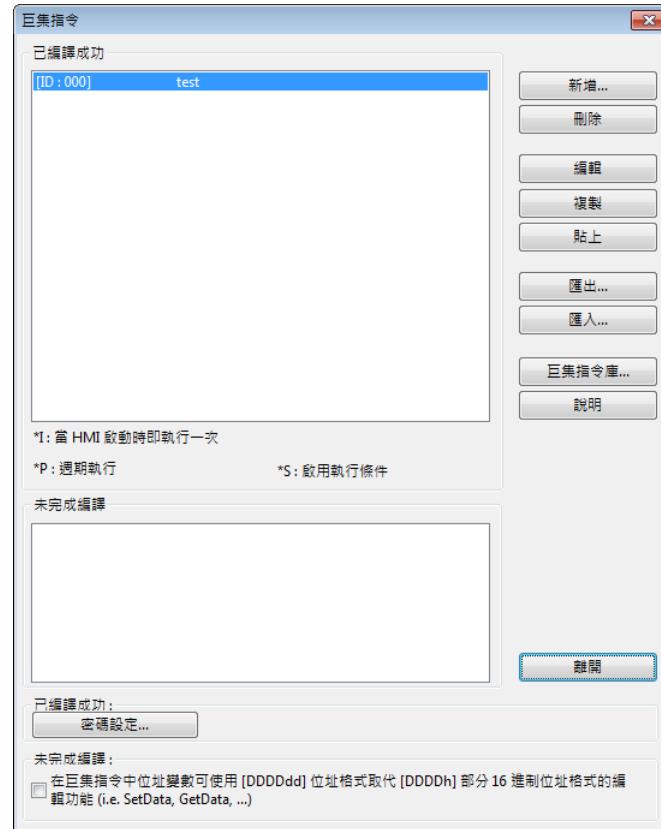
3. 設計屬於您的巨集指令程式。如果要使用內建的函數，譬如 Setdata() 或者 Getdata() 等函數，單擊 [函數] 按鍵開啟函數列表對話方塊，選擇需要的函數，並設定必要的參數。



4. 編輯完成一個新建的巨集指令程式後，單擊 [編譯] 按鍵，對該巨集指令進行編譯工作。



5. 如果沒有錯誤，單擊 [關閉] 按鍵，這樣在 [已編譯成功] 區會發現新增了一個 “test” 這個名稱的巨集指令。



### 18.8.2. 執行巨集指令

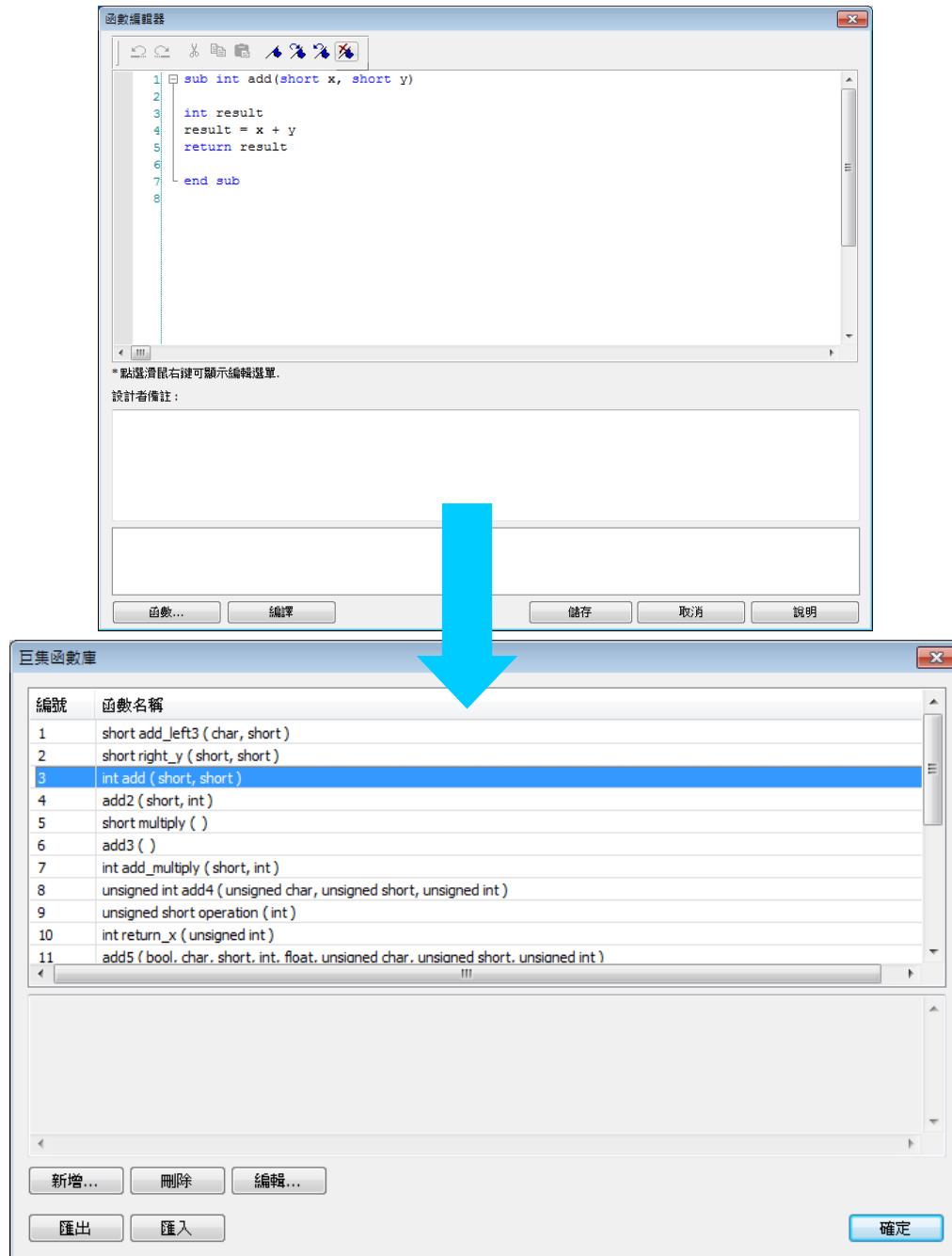
執行巨集指令有多種不同的方法，下面分別說明。

- 使用 [PLC 控制] 物件
  - 1. 打開 [PLC 控制] 物件，並設定屬性為 [執行巨集指令]。
  - 2. 選擇需要執行的巨集指令名稱。選擇一個位元作為觸發巨集指令並設定觸發巨集指令的條件。在條件滿足時，該巨集指令將會被重複執行。為了每次只讓巨集指令執行一次，設計時需在巨集指令將該觸發位元復位。
  - 3. 使用一個 [位元狀態設定] 物件或者 [位元狀態切換開關] 物件作為這個位元的控制開關。
- 
- 使用 [位元狀態設定] 物件或者 [位元狀態切換開關] 物件
  - 1. 在 [位元狀態設定] 物件或者 [位元狀態切換開關] 物件的一般屬性頁中，勾選 [使用巨集指令]。
  - 2. 選擇要執行的巨集指令。當這個物件被執行時，選擇的巨集指令就會被執行一次。
- 
- 使用 [功能鍵]
  - 1. 在 [功能鍵] 的一般屬性頁對話方塊中，勾選 [觸發巨集指令]。
  - 2. 選擇需要執行的巨集指令。每按一下這個功能鍵時，選擇的巨集指令就會被執行一次。
- 
- 使用 [巨集指令編輯器] 設定條件
  - 1. [週期執行]：可以設定每間隔幾秒自動執行巨集指令一次。
  - 2. [當 HMI 啟動時即執行一次]：當 HMI 重新上電或重新啟動時會執行此巨集指令一次。
- 
- 在 [視窗設定] 設定巨集的執行時機
  - 1. [開啟視窗時執行]：當開啟此視窗時即執行指定的巨集指令一次。
  - 2. [循環執行]：當開啟此視窗時，即每 0.5 秒循環執行指定的巨集指令。
  - 3. [關閉視窗時執行]：當離開此視窗時即執行指定的巨集指令一次。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

### 18.9. 使用者自定義函數功能

在使用巨集編輯器時，為了減少定義函數的時間，使用者可從內建的函數庫搜尋所需的函數。然而，當使用者編輯巨集時，若有某些特定的函數常常使用卻無法從內建函數庫搜尋到時，就可以自行定義所需的函數並儲存起來。當下次需要再定義相同的函數時，可由 [巨集指令庫] 呼叫出已儲存的函數，方便函數編輯。另外，[巨集指令庫] 也大幅提升了使用者自訂函數之可攜性。建立函數前可先查看現有內建函數或線上函數庫是否有現成函數可使用。



### 18.9.1. 汇入函数库档案

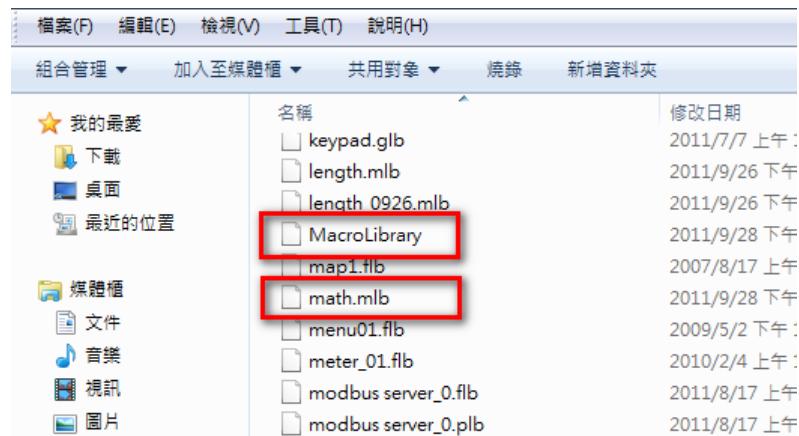
HMI 編輯軟體開啟一個工程檔時，會自動去讀入一個預設函數庫檔案，並載入函數資訊。當開啟一專案有呼叫到使用者定義函數時必須先載入相關的 .mlb 檔案。

1. 預設函數庫檔名：MacroLibrary (沒有副檔名)
2. 函數庫路徑：HMI 編輯軟體的安裝目錄的 \library 資料夾下面。
3. \library 資料夾下面可以看到兩種函數庫檔案：

沒有副檔名： MacroLibrary，為預設函數庫， HMI 編輯軟體指定讀入此檔案。

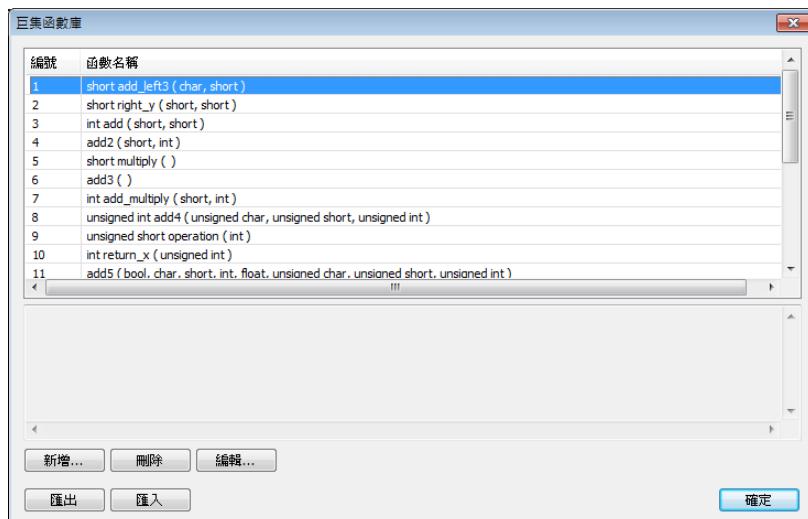
有副檔名 (.mlb)：例如 math.mlb，使用者匯入/匯出時會去讀 / 寫的檔案，具可攜性，欲使用時再從資料夾呼叫出檔案即可。

4. EasyBuilder Pro 開啟時，只會去載入預設函數庫中的函數，使用者若需要用到 .mlb 檔內的函數時，必須自行將其匯入。



### 18.9.2. 如何使用巨集函數庫

1. 在巨集編輯器中直接呼叫函數。



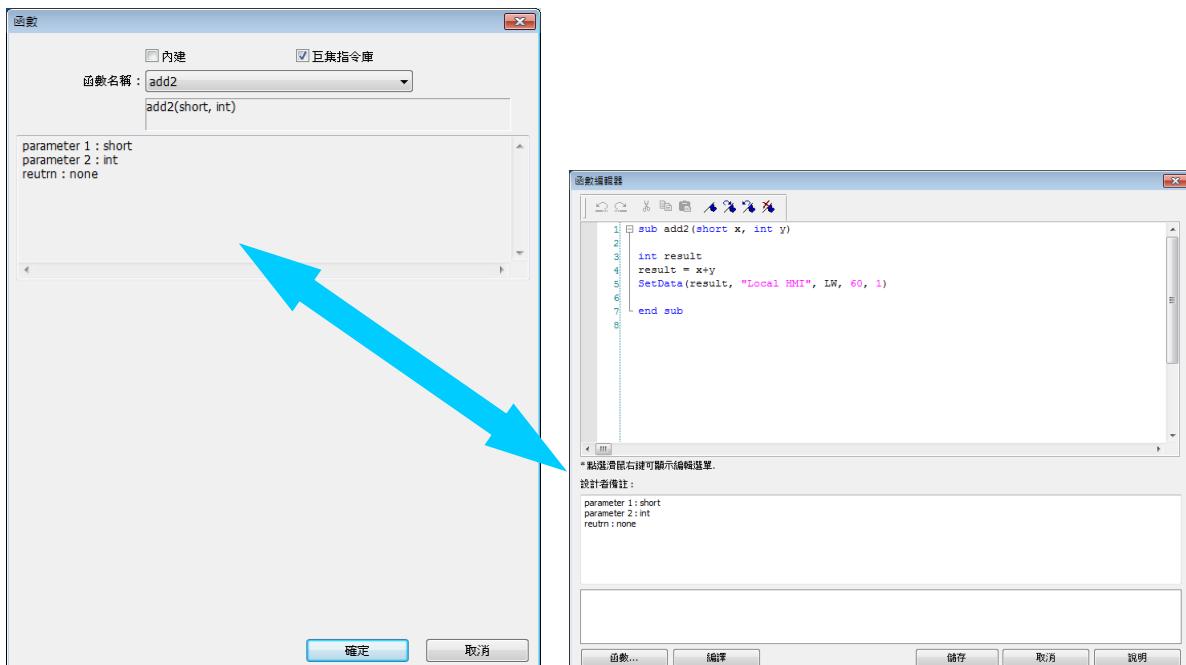
2. 按下巨集編輯器左下角的 [函數] 按鈕開啟函數對話框。



3. 須至少勾選一項 [巨集指令庫] 或 [內建]，並選擇欲使用的函數。



4. 顯示函數說明文字，即是使用者在函數編輯器中編輯的說明文字。



5. 選取欲使用的函數，將函數庫中的 [變數類型] 修改為指定的變數名稱。

```

1  macro_command main()
2
3
4  short a
5  int b,result
6
7  add2(short, int)
8
9  end macro_command

```

```

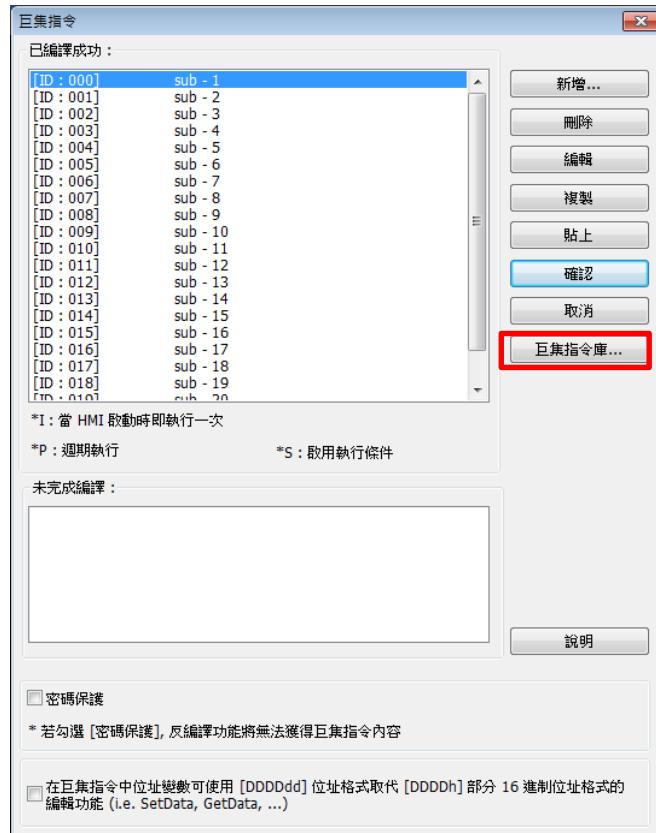
1  macro_command main()
2
3
4  short a
5  int b,result
6
7  result = add2(a, b)
8
9  end macro_command

```

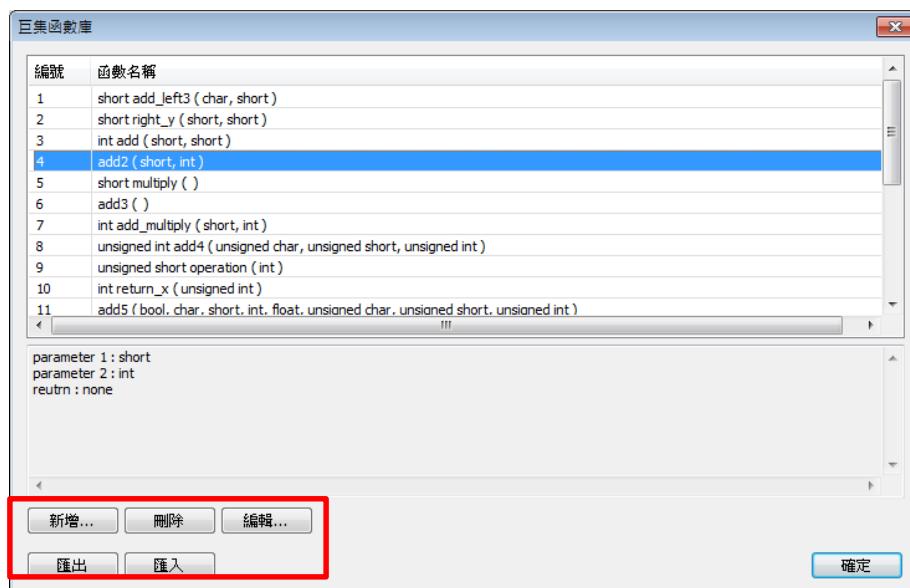
6. 使用者根據以上的步驟，即可完成使用自訂函數的功能，有效節省了重複定義相同的函數。

### 18.9.3. 函數庫管理介面

- 開啟巨集管理對話框，按下右下角【巨集指令庫】按鈕，進入函數庫管理對話框介面。



- 函數庫管理對話框中有函數列表。此列表列出工程檔開啟時，HMI 編輯軟體會從預設函數庫載入所有公共巨集庫。



- 函數列表中每一行的格式如下：

```
return_type function_name ( parameter_type1, ..., parameter_typeN)
```

`return_type` 表示回傳值型態，若無回傳值則此欄位省略；`function_name` 表示函數名稱。

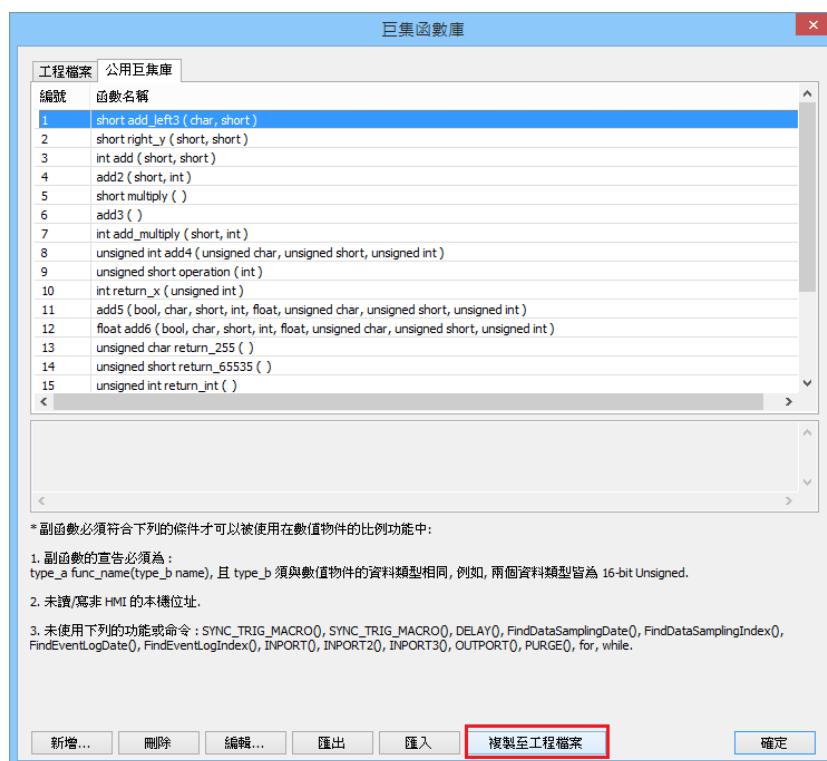
`parameter_typeN` 表示第 N 個參數型態，若此函數不接受任何參數，此欄位省略。

```

1. sub int ADD(int a, int b)
2.     int ret
3.     ret = a+b
4.     return ret
5. end sub
6.

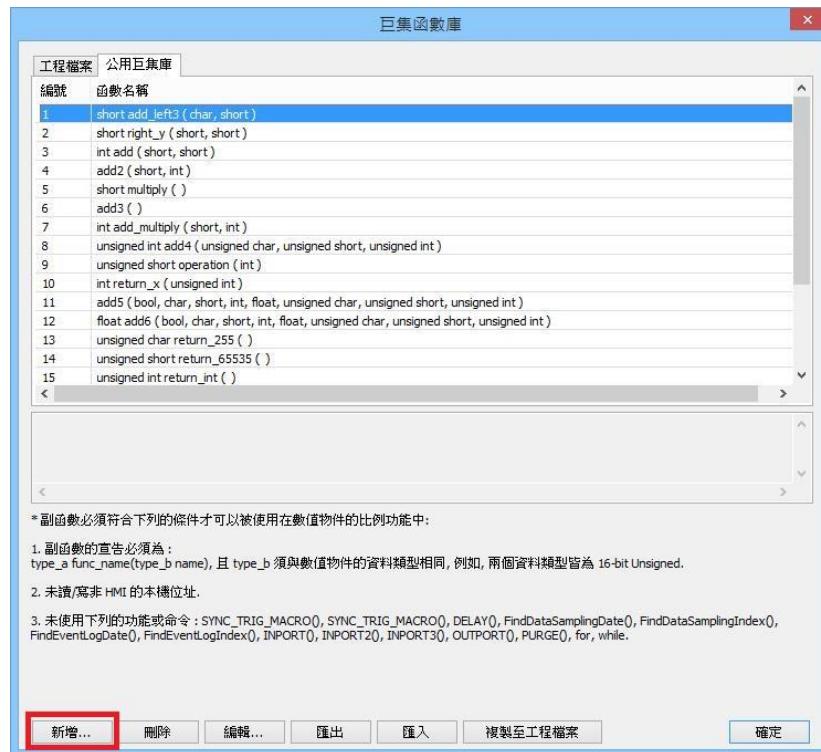
```

- 函數可以內嵌到工程檔案中。選擇函數項目再點選 [複製至工程檔案]，在 [工程檔案] 標籤中就可以找到此函數，如此當使用者將專案拿到別台電腦中開啟，仍然可使用此函數。編譯工程檔案時會將工程檔案中有使用到的內嵌函數編譯進 .exob 檔案中，若工程檔案中都未使用的函數，反編譯後會發現找不到此函數內容。

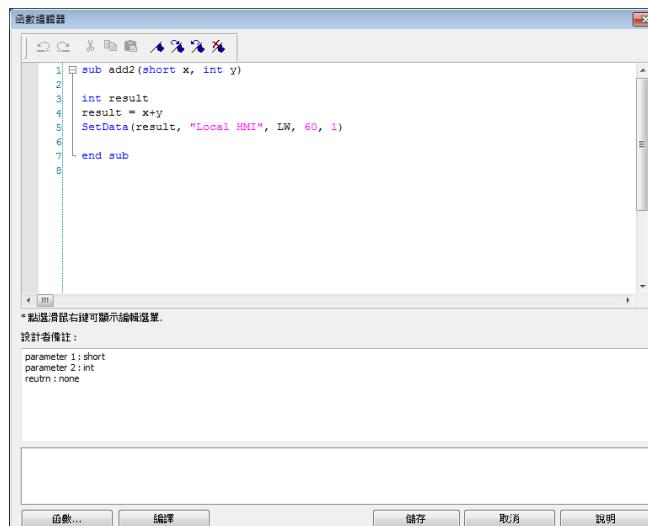


### 18.9.3.1. 新建函數

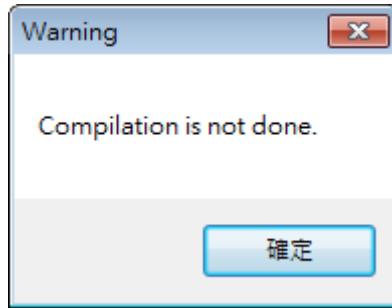
- 按下 [新增] 進入函數編輯器。



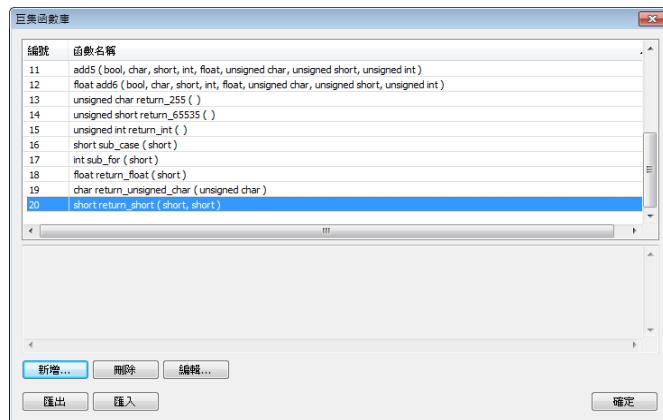
- 在函數編輯區編輯函數，接著按 [編譯] 再按 [儲存]。



- 使用者可以在函數說明編輯區中編輯說明文字，說明此函數的規格、使用方法、作者聲明等等。
- 函數編輯完成後，必須通過編譯，才可以按下 [儲存] 寫入函數庫。否則會出現下圖彈出視窗，警告使用者此函數未完成編譯。



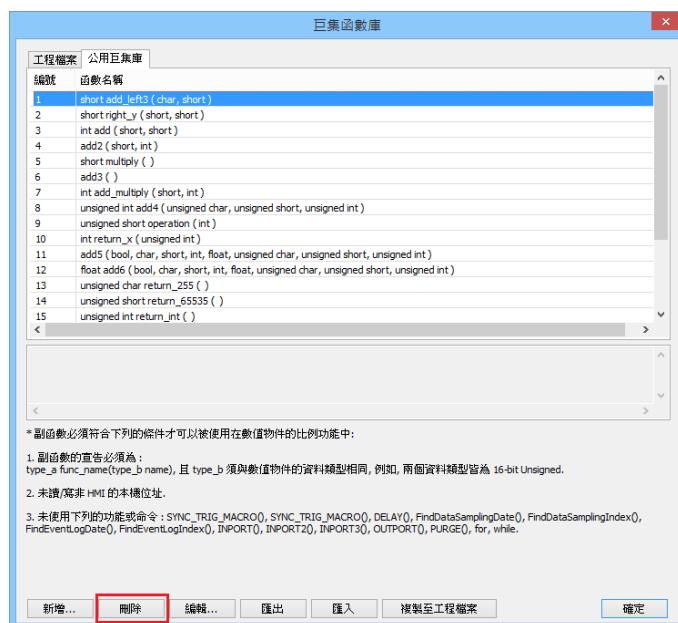
### 5. 成功加入函數庫。



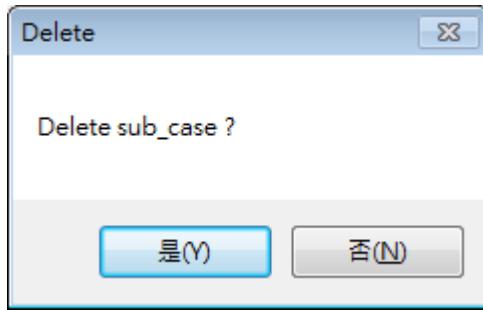
- 函數中可宣告的資料型態總數為 4096 個位元組。
- 函數名稱必須為英數字元，且不可為數字開頭。

#### 18.9.3.2. 刪除函數

##### 1. 在函數列表中選定要刪除的函數，按下 [刪除] 按鈕，即可刪除該函數。

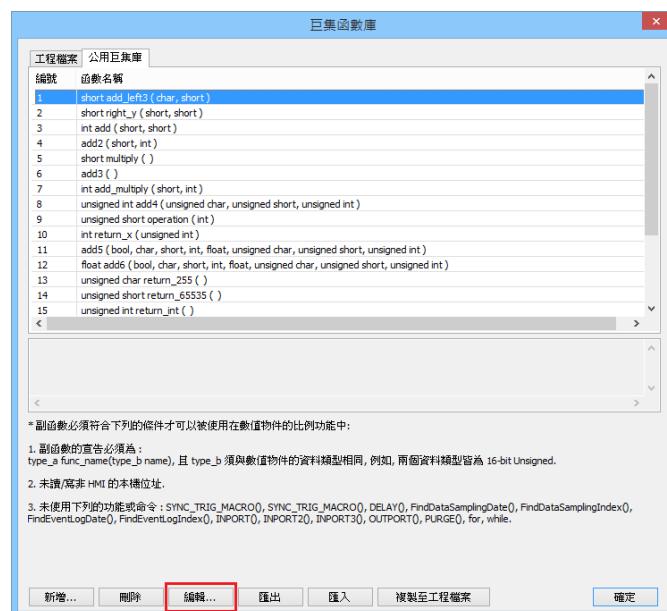


2. 按下 [是] 確認刪除，按下 [否] 取消刪除。按下 [是] 刪除 MAX\_SHORT 此函數。

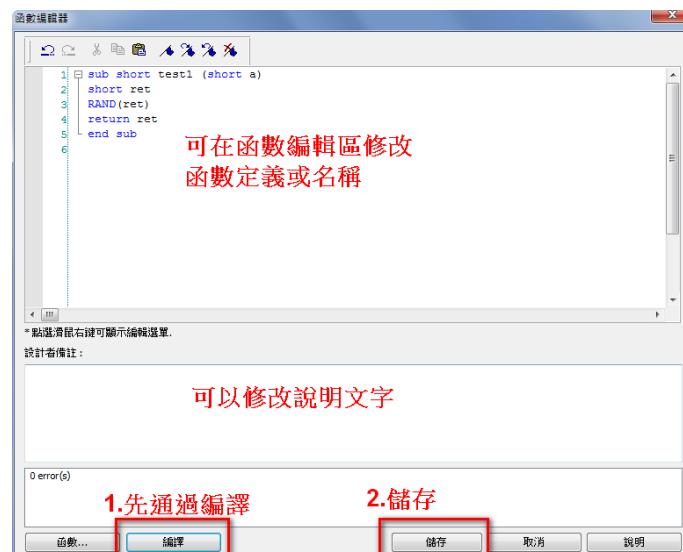


### 18.9.3.3. 修改函數

1. 使用者可以修改函數庫中的函數。
2. 選定要修改的函數，按下 [編輯]按鈕，進入函數編輯器。



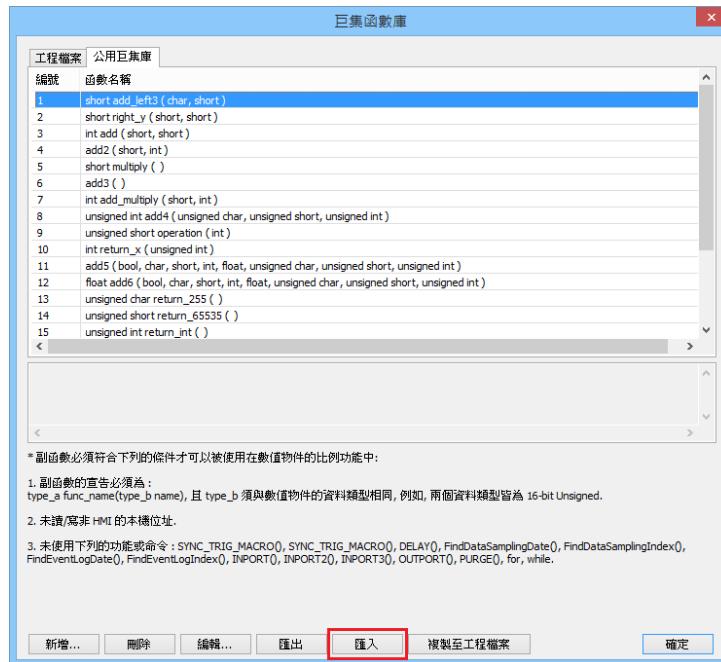
3. 也可直接在該函數上雙點滑鼠左鍵，進入函數編輯器。



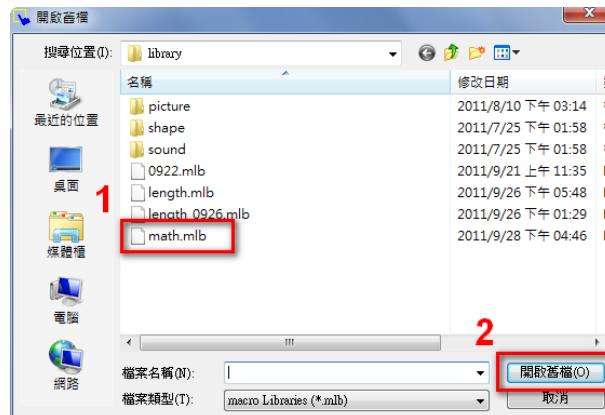
4. 修改完成後，一樣要先通過編譯，才可以儲存離開。

#### 18.9.3.4. 匯入函數

1. 使用者可以從外部的 .mlb 檔案將函數匯入。



2. 假設欲加入函數庫 math.mlb，此函數庫內含有一函數 test1。按下[開啟舊檔]按鈕。



3. 匯入函數時，若檔案庫中已存在相同名稱的函數，會出現下面的彈出視窗。



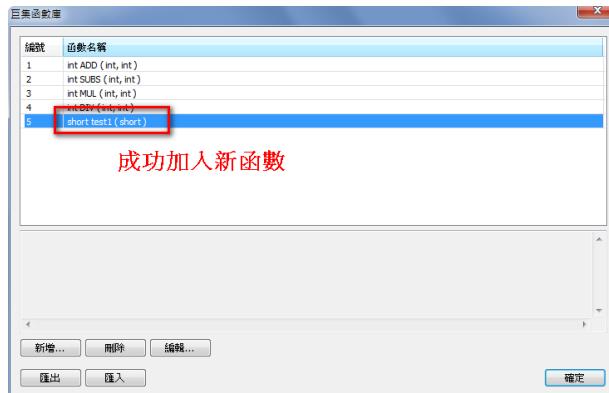
[確定]：用外部匯入的函數覆蓋函數庫中現有的同名函數。

[否]：放棄外部匯入此同名函數。

[全部確定]：全部用外部匯入的同名函數覆蓋所有同名函數。

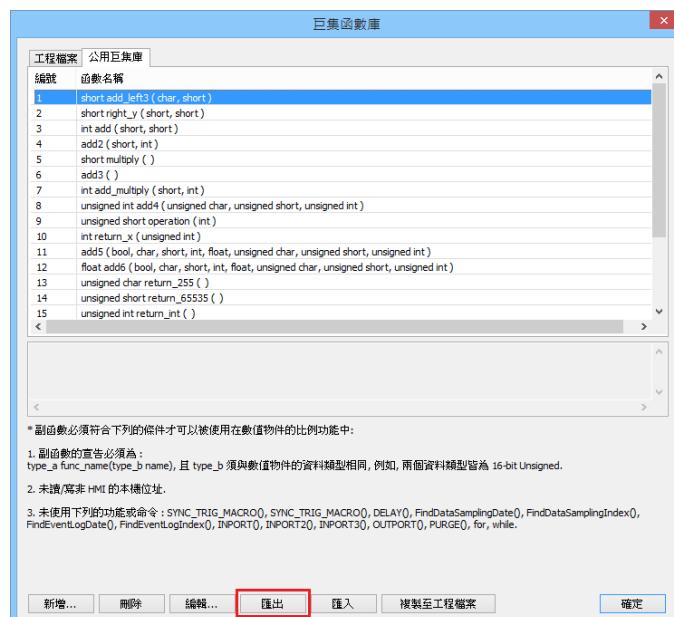
[全否]：全部放棄覆蓋匯入同名的函數。

4. 完成匯入後，匯入的函數都已寫入到預設函數庫中，因此使用者可以將 `math.mlb` 檔案刪除，而 `test1` 仍會存在函數庫中，即使重新開啟 HMI 編輯軟體亦然。

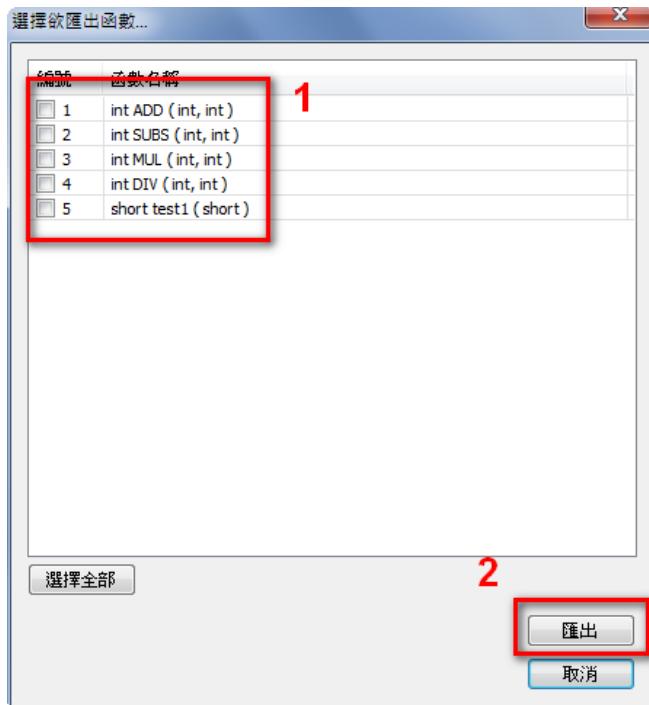


### 18.9.3.5. 匯出函數

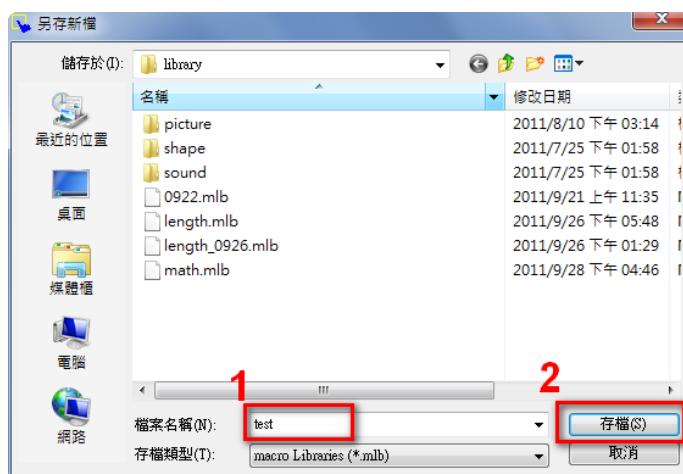
1. 使用者可以將函數庫中的函數匯出到 `.mlb` 檔。按下 [匯出] 按鈕。



2. 選擇要匯出的函數，然後按 [匯出]。



3. 匯出的目錄下出現一個 `math.mlb` 的函數庫檔案，其中包含 ADD、SUBS、MUL、DIV 這四支函數。
4. 匯出的 `.mlb` 檔可以攜帶到別的電腦上，只要使用者開啟 HMI 編輯軟體並完成匯入動作後，即可使用此檔案內所提供的所有函數。



## 18.10. 使用巨集指令時的注意事項

1. 儲存局部變數的空間是 4KB，所以各種不同變數類型的最大陣列大小為如下：

```
char    a[4096]
bool   b[4096]
short  c[2048]
int    d[1024]
float  e[1024]
```

```
long    f[512]
```

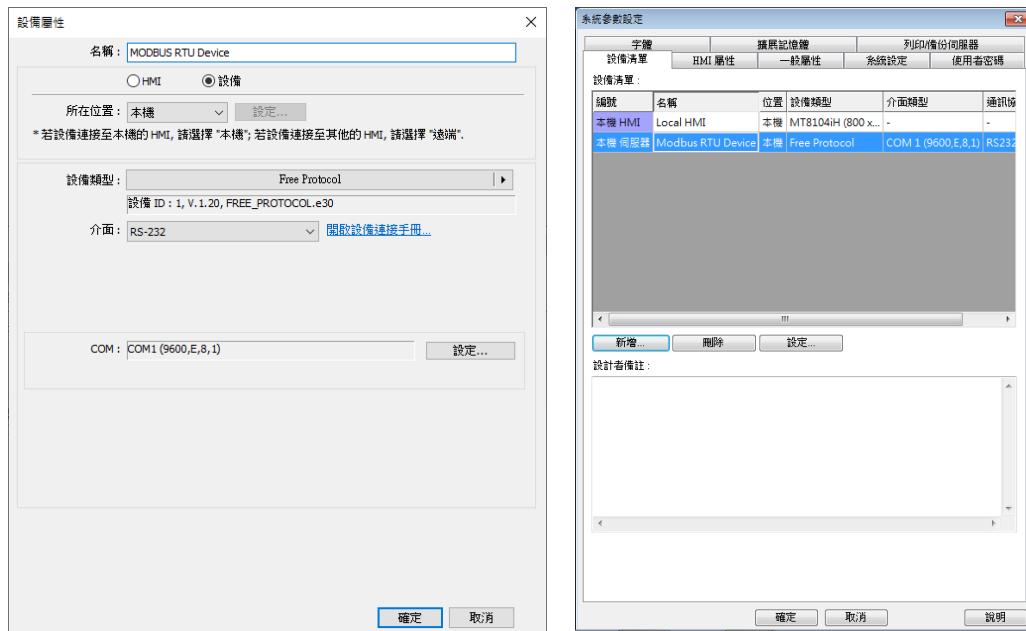
```
double   g[512]
```

2. 一個 EasyBuilder Pro 工程檔案中最多包含 255 個巨集指令，但 cMT X 系列可以支援到 500 個巨集指令。
3. 巨集指令有可能造成 HMI 當機，可能的原因為：
  - 巨集指令中執行了一個不正確的無限迴圈
  - 陣列的大小超過了巨集指令的變數容量
4. 設備的通訊速度可能影響巨集指令的執行速度。相對的，使用過多的巨集指令，可能會造成與設備的通訊速度變慢。

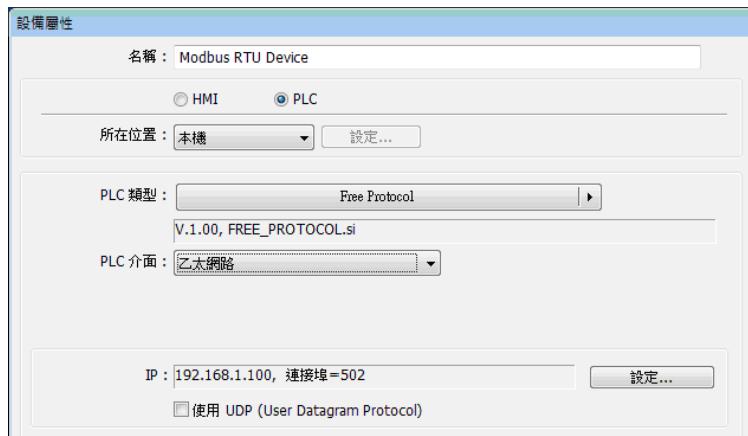
## 18.11. 使用自由協定去控制一個設備

當 EasyBuilder Pro 沒有內建與某一個設備通訊的驅動程式時，使用者也可以使用巨集指令中的 OUTPORT 和 INPORT 函數來實現與該設備的通訊。使用 OUTPORT 和 INPORT 函數發送和接收的資料，必須遵行該設備的通訊協定。下面的範例程式說明了如何使用這兩個函數來控制一個 MODBUS RTU 設備。

1. 首先，在系統參數/設備列表中建立一個新的設備。這個新建的 [設備類型] 設置為 “Free Protocol”，[設備名稱] 設置為 “MODBUS RTU Device”，如下圖所示。



2. 這裡的設備連接使用 RS232，如果連接一個 MODBUS TCP / IP 設備，這個介面類型必須設定為 “乙太網路”，同時必須設定正確的 IP 位址和連接埠號，如下圖所示。



假設 HMI 人機界面將要讀取設備中的 4x\_1 和 4x\_2 兩個資料暫存器。首先，使用 OUTPORT 函數發送讀命令給這個設備，OUTPORT 函數的寫法為：

`OUTPORT(command[start], device_name, cmd_count)`

因為 “MODBUS RTU Device” 是一個 MODBUS RTU 設備，讀數據的命令必須遵行 MODBUS RTU 協定的命令規則。所以必須使用 0x03 這個命令去讀取 4x\_1 和 4x\_2 這兩個資料暫存器的資料。下圖說明瞭讀命令的格式。(省略了設備的站號和最後的兩個 CRC 位元組)

Request		
Function code	1 Byte	0x03
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response		
Function code	1 Byte	0x03
Byte count	1 Byte	2 x N*
Register value	N* x 2 Bytes	

\*N = Quantity of Registers

Error		
Error code	1 Byte	0x83
Exception code	1 Byte	01 or 02 or 03 or 04

根據這個 MODBUS RTU 協定，發送命令的內容如下所示：(總共 8 個位元組)

response[0]: station number	(BYTE 0) 站號
response[1]: function code	(BYTE 1) 功能碼
response[2]: high byte of starting address	(BYTE 2) 起始位址高位元
response[3]: low byte of starting address	(BYTE 3) 起始位址低位元
response[4]: high byte of quantity of registers (BYTE 4)	資料暫存器的高位元
response[5]: low byte of quantity of registers	(BYTE 5) 資料暫存器的低位元
response[6]: low byte of 16-bit CRC	(BYTE 6) CRC 的低位元組
response[7]: high byte of 16-bit CRC	(BYTE 7) CRC 的高位元組

所以讀數據的命令巨集指令程式設計如下：

```

char command[32]
short address, checksum

FILL(command[0], 0, 32) // initialize command[0]~command[31] to 0

command[0] = 0x1 // station number
command[1] = 0x3 // read holding registers (function code is 0x3)

address = // starting address (4x_1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

read_no = 2 // the total words of reading is 2 words
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])

CRC(command[0], checksum, 6) // calculate 16-bit CRC

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

```

最後，使用 OUTPORT 函數將這個讀命令發送給這個 MODBUS RTU 設備。

```
OUTPORT(command[0], "MODBUS RTU Device", 8) // 發送命令
```

發送完這個命令後，使用 IMPORT 函數讀取該 MODBUS RTU 設備返回的命令。根據 MODBUS RTU 協定，這個回復的命令內容為如下 (總共 9 個位元組)：

command[0]: station number	(BYTE 0) 站號
command[1]: function code	(BYTE 1) 功能碼
command[2]: byte count	(BYTE 2) 位元組長度
command[3]: high byte of 4x_1	(BYTE 3) 第一個資料的高位元組
command[4]: low byte of 4x_1	(BYTE 4) 第一個資料的低位元組
command[5]: high byte of 4x_2	(BYTE 5) 第二個資料的高位元組
command[6]: low byte of 4x_2	(BYTE 6) 第二個資料的低位元組
command[7]: low byte of 16-bit CRC	(BYTE 7) CRC 的低位元組
command[8]: high byte of 16-bit CRC	(BYTE 8) CRC 的高位元組

此時，IMPORT 函數的語句如下：

```
IMPORT(response[0], "MODBUS RTU Device", 9, return_value) // 讀取回復命令
```

函數中，實際讀取到的位元組長度存放在變數 `return_value` (變數類型為位元組) 中。如果 `return_value` 的資料為 0，則表示使用 IMPORT 讀取命令失敗。

根據 MODBUS RTU 協定，如果命令人回復的正確，則 `response[1]` 必須為 0x03。當讀取到正確的命令後，計算出 `4x_1` 和 `4x_2` 這兩個暫存器的值，並將這兩個資料送到人機界面的 LW-100 和

LW-101 暫存器中。

```
If (return_value) >0 and response[1] == 0x3) then  
    read_data[0] = response[4] + (response[3] << 8) // 計算 4x_1 的資料  
    read_data[1] = response[6] + (response[5] << 8) // 計算 4x_2 的資料  
  
    SetData(read_data[0], "Local HMI", LW, 100, 2) // 將資料存至 HMI 上  
endif
```

完整的巨集指令程式如下：

```
// Read Holding Registers
macro_command main()

char command[32], response[32]
short address, checksum
short read_no, return_value, read_data[2], i

FILL(command[0], 0, 32)//  initialize command[0]~command[31] to 0
FILL(response[0], 0, 32)

command[0] = 0x1//  station number
command[1] = 0x3//  read holding registers (function code is 0x3)

address = 0
address = 0//  starting address (4x_1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

read_no = 2//  the total words of reading is 2 words
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])

CRC(command[0], checksum, 6)//  calculate 16-bit CRC

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

OUTPORT(command[0], "MODBUS RTU Device", 8 )//  send request
INPORT(response[0], "MODBUS RTU Device", 9, return_value)//  read response

if (return_value > 0 and response[1] == 0x3) then
    read_data[0] = response[4] + (response[3] << 8)//  4x_1
    read_data[1] = response[6] + (response[5] << 8)//  4x_2

    SetData(read_data[0], "Local HMI", LW, 100, 2)
end if

end macro_command
```

下面的舉例說明如何使用自由協定設定 MODBUS RTU 設備中 0x\_1 的狀態。這個是使用 MODBUS RTU 協議中的“寫單個暫存器”的功能碼“0x05”來實現的。

**Request**

Function code	1 Byte	<b>0x05</b>
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

**Response**

Function code	1 Byte	<b>0x05</b>
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

**Error**

Error code	1 Byte	<b>0x85</b>
Exception code	1 Byte	01 or 02 or 03 or 04

完整的巨集指令程式如下：

```
// Write Single Coil (ON)
macro_command main()

char command[32], response[32]
short address, checksum
short i, return_value

FILL(command[0], 0, 32)// initialize command[0]~ command[31] to 0
FILL(response[0], 0, 32)

command[0] = 0x1// station number
command[1] = 0x5// function code : write single coil

address = 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

command[4] = 0xff// force 0x_1 on
command[5] = 0

CRC(command[0], checksum, 6)

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

OUTPORT(command[0], "MODBUS RTU Device", 8)// send request
IMPORT(response[0], "MODBUS RTU Device", 8, return_value)// read response

end macro_command
```



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 18.12. 編譯錯誤提示資訊

- 錯誤資訊格式：

**error C# : 錯誤描述**

(# 是錯誤資訊編號)

舉例：error C37 : undeclared identifier : i

當編譯後提示有錯誤資訊時，這個錯誤的描述內容可以參考錯誤資訊編號。

● Error 描述

**(C1) syntax error : 'identifier'**

出現這個資訊時，有許多種可能。

舉例：

```
macro_command main()
char i, 123xyz // 不支援這個變數類型
end macro_command
```

**(C2) 'identifier' used without having been initialized**

巨集指令必須定義聲明的陣列變數的大小。

舉例：

```
macro_command main()
char i
int g[i] // i 必須為一個數值常數
end macro_command
```

**(C3) redefinition error : 'identifier'**

函數名稱和變數名稱在有效範圍內，必須是唯一的。

舉例：

```
macro_command main()
int g[10], g // 重複定義錯誤
end macro_command
```

**(C4) function name error : 'identifier'**

保留的關鍵字和常數，不能被定義為函數名稱。

舉例：

```
sub int if() // 函數名稱定義錯誤
```

**(C5) parentheses have not come in pairs**

語句中缺少了 “(“ or ”)” 。

舉例：

```
macro_command main() // 圓括號沒有成對的出現
```

**(C6) illegal expression without matching 'if'**

‘if’ 語句中沒有合法的運算式。

**(C7) illegal expression (no ‘then’) without matching ‘if’**

‘if’ 語句中缺少了 ‘then’，也就是 ‘if’ 和 ‘then’ 沒有成對。

**(C8) illegal expression (no ‘end if’)**

‘if’ 語句中缺少了 ‘end if’。

**(C9) illegal ‘end if’ without matching ‘if’**

‘end if’ 語句前缺少了 ‘if’ 語句。

**(C10) illegal ‘else’**

‘if’ 語句的標準格式請參考 CH 18.5.3 邏輯運算語句，任何與以上格式不符合的語句，在編譯時就會錯誤。

**(C11)'case' expression not constant**

‘case’ 後方的數值判斷必須為一個定值常數。

舉例：

```
macro_command main()
int a = 0
int b
select case a
case b // case 內容非常數
break
end select
end macro_command
```

**(C12) 'select' statement contains no 'case'**

‘select’ 後方缺少了 ‘case’ 語句。

舉例：

```
macro_command main()
int a = 0
int b
select a // 'select' 後方缺少了 'case' 語句
case 1
break
end select
end macro_command
```

**(C13) illegal expression without matching 'select case'**

'select' 和 'case' 沒有成對。

**(C14) 'select' statement contains no 'end select'**

'select' 語句中缺少了 'end select'。

**(C15) illegal 'case'**

'case' 語句的標準格式請參考 CH 18.5.4 多重判斷語句，任何與以上格式不符合的語句，在編譯時就會錯誤。

**(C16) illegal 'case else'**

'case else' 語句的標準格式請參考 CH 18.5.4 多重判斷語句，任何與以上格式不符合的語句，在編譯時就會錯誤。

**(C17) illegal expression (no 'for') without matching 'next'**

'for' 語句錯誤：在 'next' 前，缺少了 'for' 語句。

**(C19) variable data type error**

語句中變數的資料型態錯誤。

**(C20) must be keyword 'to' or 'down'**

缺少了關鍵字 'to' 或者 'down'。

**(C21) illegal expression (no 'next')**

'for' 語句的標準格式為：

for [變數] = [初始值] to [結束值] [step]

next [變數]

任何與上述格式不符合的語句，編譯時會錯誤。

**(C22) 'wend' statement contains no 'while'**

'while' 語句錯誤：在 'wend' 前，缺少了 'while' 語句。

**(C23) illegal expression without matching 'wend'**

缺少 'wend' 關鍵字。

'while' 語句的標準格式為：

while [邏輯運算式]

wend

任何與上述格式不符合的語句，編譯時會錯誤。

**(C24) syntax error : 'break'**

不合法的 'break' 語句。'break' 語句只會在 'for' 或者 'while' 語句中出現。

**(C25) syntax error : 'continue'**

不合法的 'continue' 語句。'continue' 語句只會在 'for' 或者 'while' 語句中出現。

**(C28) must be 'macro\_command'**

此處應該為 'macro\_command' 。

**(C29) must be key word 'sub'**

子函數的定義格式為：

sub [data type] function\_函數名稱 (...)

.....

end sub

舉例：

sub int pow (int exp)

.....

end sub

任何不符合上述語法結構的，在編譯時會錯誤。

**(C30) number of parameters is incorrect**

參數數量不匹配。

**(C31) parameter type is incorrect**

參數資料類型不相配。調用函數時，參數必須在資料類型、個數上一一對應才能通過編譯，否則編譯時將出現此項錯誤訊息。

**(C33) function name : undeclared function**

沒有定義的函數名稱。

**(C34) expected constant expression**

不合法的陣列下標表達形式。

**(C35) invalid array declaration**

不合法的陣列定義。

**(C37) undeclared identifier : i 'identifier'**

使用沒有定義的變數。只能使用已經定義的變數和函數，否則編譯時將出現此項錯誤訊息。

**(C38) device encoding method is not supported**

通訊函數 GetData( ... )、SetData( ... )的參數中有包含設備位址類型資訊，當設備位址類型不是此種設備支援的位址類型時，編譯時將出現此項錯誤訊息。

**(C39) array index must be integer, short, char or constant**

陣列的格式為：

宣告：函數名稱[constant] (constant is the size of the array)

使用：函數名稱[integer, character or constant]

任何不符合上述規則的陣列運算式，編譯時將會錯誤。

**(C40) execution syntax should not exist before variable declaration or constant definition**

變數定義語句的前面不能有執行語句。

舉例：

```
macro_command main()
```

```
int a, b
```

```
for a = 0 To 2
```

```
b = 4 + a
```

int h , k // 定義變數語句在此處是錯誤的，在一個函數內定義變數語句的前面  
不能有執行語句，例如 b = 4 + a

```
next a
```

```
end macro_command
```

**(C41) float variables cannot be contained in shift calculation**

移位運算中，運算元不能為浮點數。

**(C42) function must return a value**

函數應有返回值。

**(C43) function should not return a value**

函數不應有返回值。

**(C44) float variables cannot be contained in calculation**

運算中不能有浮點數的資料。

**(C45) device address error/tag name does not exist**

設備位址錯誤。

**(C46) size of function variables is too large (max. 4k bytes)**

一維陣列的大小超過 4k。

**(C47) macro command entry function is not only one**

巨集指令入口只能有一個，形式為：

```
macro_command function_函數名稱()
```

```
end macro_command
```

**(C49) an extended addressee's station number must be between 0 and 255**

在巨集指令中，擴展地址內的站號大小只能從 0 到 255。

舉例：

```
SetData(bits[0] , "PLC 1", LB , 300#123, 100)
```

// illegal : 300#123 意思是站號為 300, 但是最大值是 255

**(C50) an invalid device name**

在巨集指令中，設備的名稱並未定義在系統參數的設備列表中。

**(C51) macro command do not control a remote device**

巨集指令只能控制本機連接的設備。

舉例：

SetData(bits[0], "PLC 1", LB, 300#123, 100)

"PLC1" 連接在遠端的 HMI 上，所以它不能被執行。

**(C52) GetData/GetDataEx/StringGet/StringGetEx cannot use a broadcast station no.**

上述語法無法使用廣播站號。

**(C53) IMPORT() must use a "Free Protocol" device**

IMPORT 語法必須使用在 "Free Protocol" 的設備上。

**(C54) OUTPORT() must use a "Free Protocol" device**

OUTPORT 語法必須使用在 "Free Protocol" 的設備上。

**(C55) Recipe Database is not supported on this HMI model**

該機型不支援配方資料庫功能。

**(C56) the data type of 'identifier' must be "unsigned"**

資料型態必須為 "unsigned"。

**(C57) Recipe bit position is out of range**

使用到的配方資料的 Recipe\_bit 設定超出範圍。

**(C58) assignment is out of range**

變數的賦值超出資料型態定義的上下限範圍。

**(C59) declaration of global variables in macro library is not allowed**

巨集指令庫中不允許宣告全域變數。

**(C60) illegal expression following the keyword "step" in the for-loop**

在 'for' 語句中的關鍵字 'step' 後方有不合法的表達式。

**(C61) nested call to sub function is not allowed**

不允許子函數間的巢狀呼叫。

**(C62) case else must be placed at the end of the select case**

'case else' 必須位在 'select case' 語句中的最後一項。

**(C63) array index exceeds array size**

陣列索引超出陣列定義的大小。

**(C64) data count exceeds the size of read/write buffer**

讀/寫指令超過 4k 位元組。

**(C65) SQL syntax not accepted**

不支援的 SQL 語法。

**(C66) recipe tag not found**

配方標籤名稱不存在於配方資料庫。

**(C67) counter variable of for-loop doesn't support unsigned data type**

'for' 語句中的計數器變數不支援 unsigned 資料型別。

舉例：

```
macro_command main()
unsigned int i
for i = 5 down 0 step 1 // 不支援 unsigned 資料型別
next
end macro_command
```

**(C68) Conversion Tag size error**

使用轉換標籤相關語法時長度錯誤。

**(C69) Macro name : 'identifier' not found**

使用的巨集名稱不存在。

**(C70) Macro undefined : Macro ID = 'identifier'**

使用的巨集 ID 不存在。

**(C71) syntax error (or number of characters exceeds 2048)**

語法錯誤（或字元數超過 2048）。

**(C72) parameter value is out of range : 'identifier'**

參數值超出範圍。

**(C73) 'identifier' does not support**

**GetData/setData/GetDataEx/setDataEx/StringGet/StringGetEx/StringSet/StringSetEx**

識別字不支援使用於上述語法。

**(C74) station no. variable must be between var0 ~ var15**

站號變數必須在 var0 ~ var15 之間。

**(C75) Macro function is not supported on this HMI model**

該機型不支援巨集相關功能。

**(C76) the "unsigned" keyword must be followed by a data type**

關鍵字 "unsigned" 後面必須跟著資料類型。

**(C77) index register syntax error**

索引|暫存器語法錯誤。

**(C78) this tag does not support index register**

此標籤不支援索引|暫存器。

**(C79) index register is not supported on this HMI model**

該機型不支援索引|暫存器。

**(C80) function does not support if/while/for/switch statement**

函數不支援 if/while/for/switch 語句。

**(C82) string must be declared as a char or unsigned char array**

字串必須宣告為 char 或 unsigned char 陣列。

**(C83) 'identifier' must be a constant data**

識別字必須是常數資料。

**(C84) array data exceeds array size**

陣列資料超出陣列大小。

**(C85) illegal expression (no 'select') without matching 'end select'**

'end select' 缺少 'select' 關鍵字。

**(C95) total number of characters exceeds 100000**

巨集指令總字元數不得超過 100,000。

若使用者欲使用超過字元數量限制的巨集指令時，需要先將該巨集指令的總字元數調整至 100,000 以下，並透過 SYNC\_TRIG\_MACRO 函數利用同步的方式觸發執行其他的巨集指令完成總字元數超過 100,000 的巨集指令。

### 18.13. 巨集指令範例程式

- for 迴圈，各種運算式 (算術，移位元，邏輯，關係運算式)

```
macro_command main()
```

```
    int a[10], b[10], i
```

```
    b[0] = (400 + 400 << 2) / 401
    b[1] = 22 * 2 - 30 % 7
    b[2] = 111 >> 2
    b[3] = 403 > 9 + 3 >= 9 + 3 < 4 + 3 <= 8 + 8 == 8
    b[4] = not 8 + 1 and 2 + 1 or 0 + 1 xor 2
    b[5] = 405 and 3 and not 0
    b[6] = 8 & 4 + 4 & 4 + 8 | 4 + 8 ^ 4
    b[7] = 6 - (~4)
    b[8] = 0x11
    b[9] = 409
```

```
    for i = 0 to 4 step 1
        if (a[0] == 400) then
            GetData(a[0],"Device 1", 4x, 0,9)
            GetData(b[0],"Device 1", 4x, 11,10)
        end If
        next i
    end macro_command
```

- while, if, break 語句

```
macro_command main()
```

```
    int b[10], i
```

```
    i = 5
```

```
    while i == 5 - 20 % 3
```

```
        GetData(b[1], "Device 1", 4x, 11, 1)
```

```
        if b[1] == 100 then
```

```
            break
```

```
        end if
```

```
    wend
```

```
end macro_command
```

- 總體變數和子函數調用

```
char g
sub int fun(int j, int k)
    int y

    SetData(j, "Local HMI", LB, 14, 1)
    GetData(y, "Local HMI", LB, 15, 1)
    g = y

    return y
end Sub

macro_command main()
    int a, b, i

    a = 2
    b = 3
    i = fun(a, b)
    SetData(i, "Local HMI", LB, 16, 1)
end macro_command
```

- if 結構語句

```
macro_command main()
    int k[10], j

    for j = 0 to 10
        k[j] = j
    next j

    if k[0] == 0 then
        GetData(k[1], "Device 1", 4x, 0, 1)
    end if

    if k[0] == 0 then
        GetData(k[1], "Device 1", 4x, 0, 1)
    else
        GetData(k[2], "Device 1", 4x, 0, 1)
    end if
```

```
if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 1, 1)
else if k[2] == 1 then
    SetData(k[3], "Device 1", 4x, 2, 1)
end If
```

```
if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 3, 1)
else if k[2] == 2 then
    SetData(k[3], "Device 1", 4x, 4, 1)
else
    SetData(k[4], "Device 1", 4x, 5, 1)
end If
end macro_command
```

- while 和 wend 結構語句

```
macro_command main()
    char i = 0
    int a[13], b[14], c = 4848

    b[0] = 13

    while b[0]
        a[i] = 20 + i * 10

        if a[i] == 120 then
            c = 200
            break
        end if

        i = i + 1
    wend

    SetData(c, "Device 1", 4x, 2, 1)
end macro_command
```

- break 和 continue 語句結構

```
macro_command main()
    char i = 0
    int a[13], b[14], c = 4848
    b[0] = 13

    while b[0]
        a[i] = 20 + i * 10

        if a[i] == 120 then
            c = 200
            i = i + 1
            continue
        end if
        i = i + 1

        if c == 200 then
            SetData(c, "Device 1", 4x, 2, 1)
            break
        end if
    wend

end macro_command
```

- 陣列結構

```
macro_command main()
    int a[25], b[25], i

    b[0] = 13

    for i = 0 to b[0] step 1
        a[i] = 20 + i * 10
    next i

    SetData(a[0], "Device 1", 4x, 0, 13)
end macro_command
```

- 字串相關語法內使用”符號的方法，也可使用在變數宣告及函數參數的時候

```
macro_command main()
    char data[40]= "\""Note\" "
```

```
StringCopy("This is a \"test\" for weintek", data[7])
//字串內容為"Note" This is a "test" for weintek
end macro_command
```

## 18.14. 巨集指令 TRACE 函數

巨集指令的 TRACE 函數，搭配使用 EasyDiagnoser/cMT Diagnoser，可用來檢視所使用變數目前的內容。使用 cMT / cMT X 系列的用戶，建議使用 cMT Diagnoser 的內建巨集除錯工具進行除錯，無需特別使用 TRACE 函數。

下面示範如何在巨集指令中利用 TRACE 命令並使用 EasyDiagnoser 進行監控。

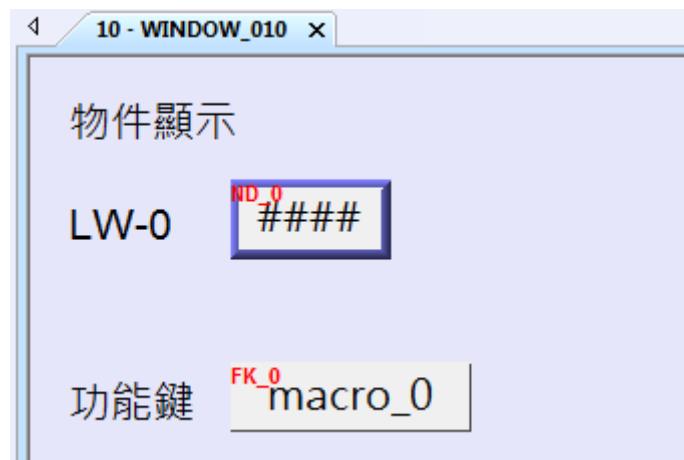
- 首先，請在工程檔案中新增 macro\_0，並在 macro\_0 的內容中加入 `TRACE("LW = %d", a)`，"%d"表示使用 10 進制顯示 LW 目前的數值。macro\_0 的內容如下：

```

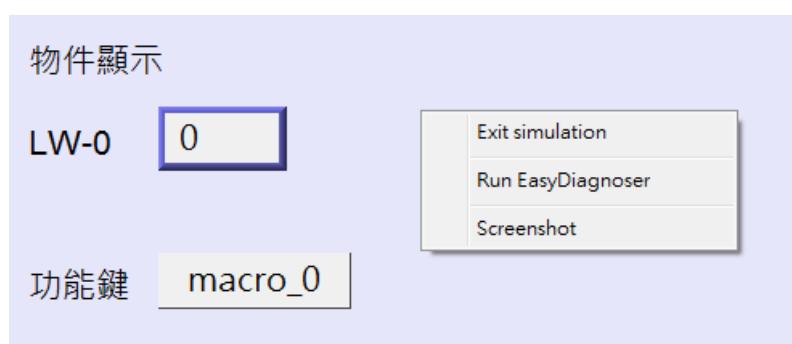
1  macro_command main()
2
3
4  short a
5  GetData(a, "Local HMI", LW, 0, 1)
6  a=a+1
7  SetData(a, "Local HMI", LW, 0, 1)
8  TRACE ("LW0 = %d" , a)
9
10 end macro_command

```

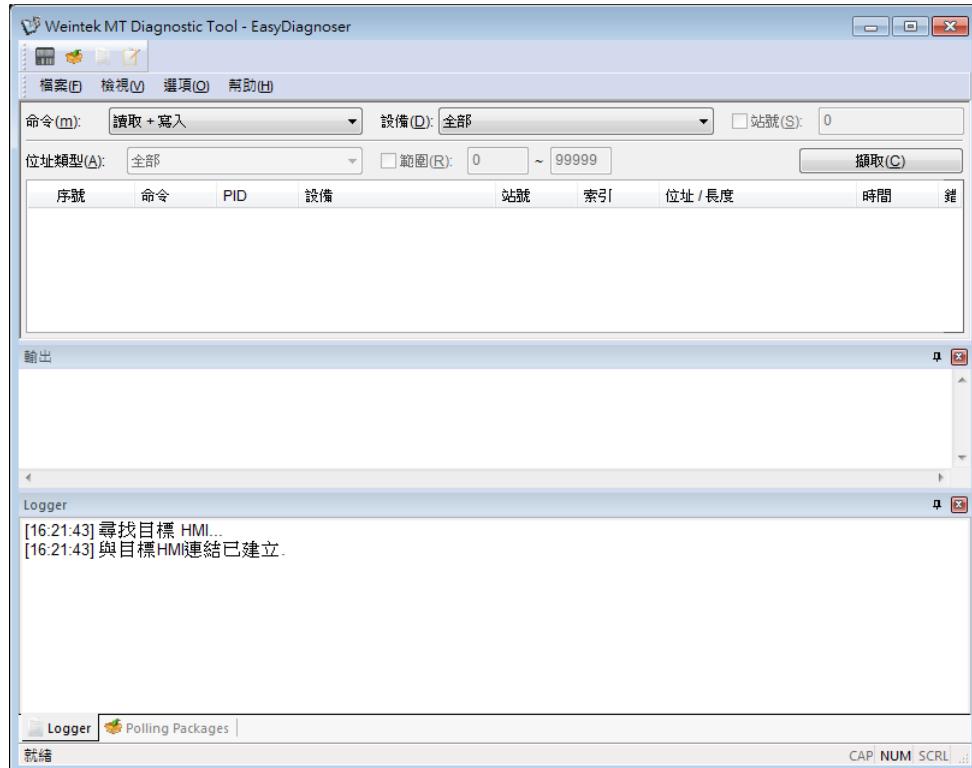
- 接著在工程檔案的第 10 頁分別加上 [數值顯示] 與 [功能鍵] 物件，[功能鍵] 物件用來執行 macro\_0。



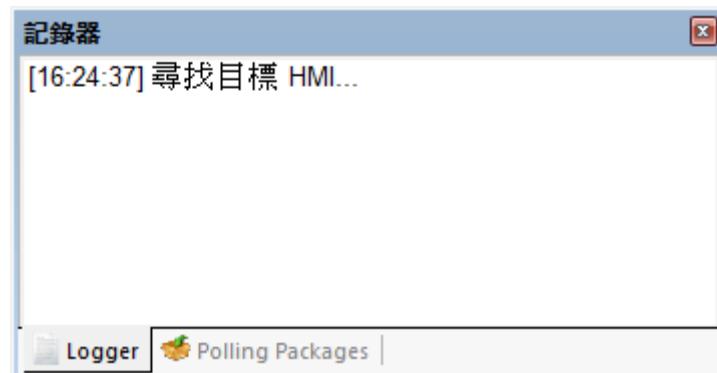
- 最後編譯已完成的工程檔案並執行離線或線上模擬。
- 在 PC 上進行模擬功能時，點擊滑鼠右鍵後選擇選單上的 "Run EasyDiagnoser"。



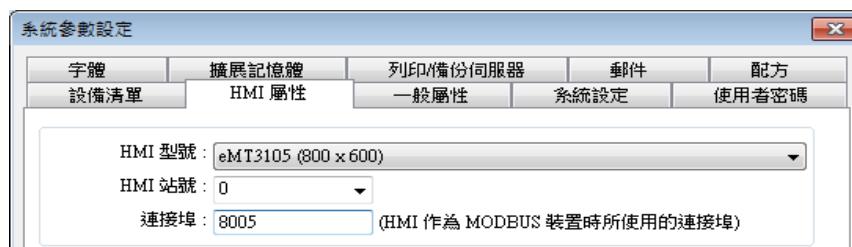
5. 此時即會出現 EasyDiagnoser 的畫面，[記錄器] 窗口用來顯示 EasyDiagnoser 是否可以連接上需要監視的 HMI，[輸出] 窗口用來顯示 TRACE 的執行結果，下圖表示 EasyDiagnoser 已成功連接上 HMI。



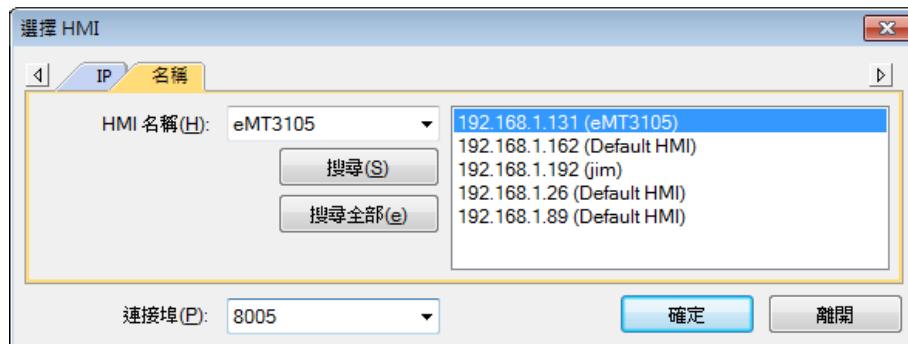
若未成功連接上 HMI，[記錄器] 的窗口會顯示下面的內容：



6. 未連線成功的可能原因是 PC 未成功執行模擬功能；另一個原因是在 PC 執行模擬功能的工程檔案所使用的 [連接埠] 不正確 (可能已被系統佔用)，此時請更改工程檔案的 [連接埠] (請參考下圖)，再次編譯重新執行模擬功能即可。



**7.** 開啟 EasyDiagnoser 時，應設定與工程檔案相同的連接埠。



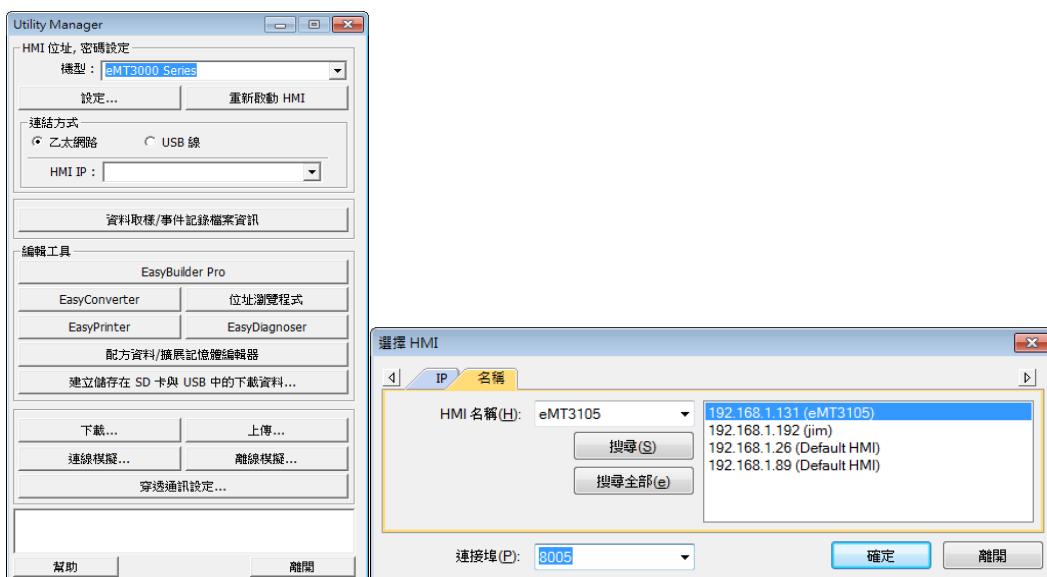
從所設定的 [連接埠] 算起連續 3 個 port 將保留給 HMI 做通訊用。例如，上圖中設定 [連接埠] 為 8005，則 8005、8006、8007 三個 port 將被保留。因此在 PC 上模擬時，應確定這些被保留的 port 未被其他程式所佔用。

TRACE 命令語法如下：

函數名稱	TRACE
語法	TRACE(format, argument)
描述	<p>一個執行中的 巨集指令可以使用此函數，監視變數內容的變化，並列印字串，以協助除錯。使用者應開啟 EasyDiagnoser /cMT Diagnoser 觀看此函數的輸出結果。</p> <p>當 TRACE 函數抓取一個%開頭的特殊字元，將同時從 argument 抓取一個參數做格式化後輸出。</p> <p>Format 代表列印格式，支援 % 開頭的特殊字元。特殊字元格式如下，其中方括號內的欄位為可選，紅色文字欄位為必需：</p> <p><b>%[flags] [width] [.precision] type</b></p> <p>每個欄位的意義如下所述：</p> <p><b>flags (可選) :</b></p> <ul style="list-style-type: none"> <li>- : 靠左對齊。指定寬度後，不足的左邊補空格</li> <li>+ : 輸出正負號</li> </ul> <p><b>width (可選) :</b></p> <p>十進位正整數，指定應預留的字元寬度，不足部份補空白字元。</p> <p><b>precision (可選) :</b></p> <p>十進位正整數，指定精確度，以及輸出字元數。</p> <p><b>type :</b></p> <ul style="list-style-type: none"> <li>C 或 c : 以字元方式輸出</li> <li>d : 以 signed 十進位整數輸出</li> <li>i : 以 signed 十進位整數輸出</li> <li>o : 以 unsigned 八進位整數輸出</li> <li>u : 以 unsigned 十進位整數輸出</li> <li>X 或 x : 以 unsigned 十六進位整數輸出</li> <li>lld : 以長整數 (64bit) signed 變數輸出 (只支援於cMT / cMT X)</li> <li>llu : 以長整數 (64bit) unsigned 變數輸出 (只支援於cMT / cMT X)</li> </ul>

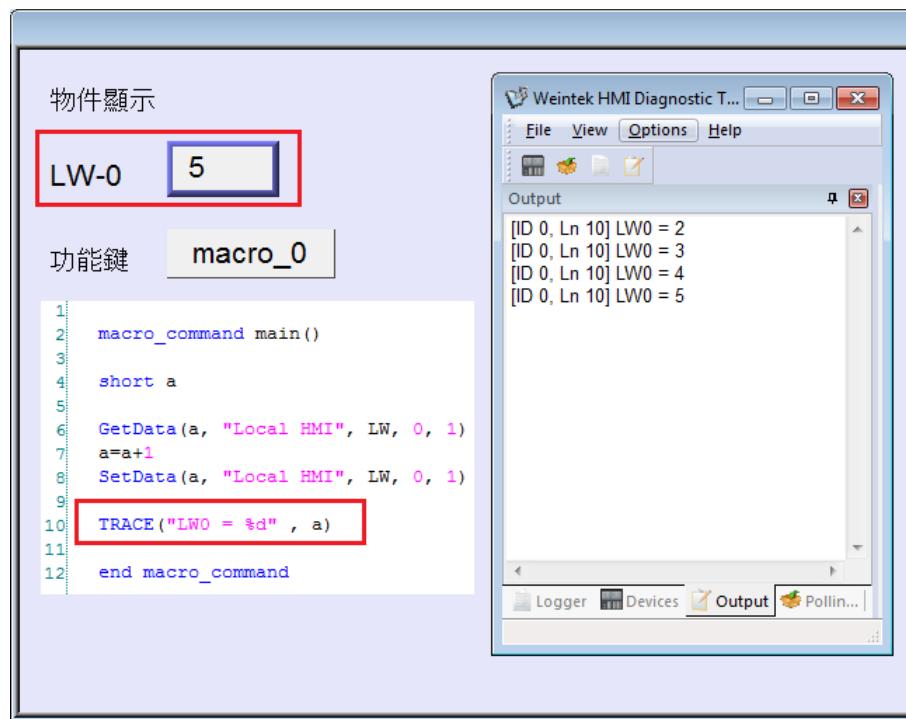
	<p><b>f</b> : 以單倍精確度浮點數輸出。</p> <p><b>lf</b> : 以雙倍精確度浮點數輸出。</p> <p><b>E</b> 或 <b>e</b> : 以科學表示法輸出。格式為[ - ] d.dddd e [sign]ddd。其中欄位d是十進位數字，欄位ddd是一至多個十進位數字，欄位ddd必須是三個十進位數字。<b>sign</b>是+或-。</p> <p><i>Format</i> 字串最長支援 256 個字元，多出的字元將被忽略。</p> <p><i>Argument</i> 部份可寫可不寫。但一個特殊字元應搭配一個變數。</p>
舉例	<pre>macro_command main() char c1 = 'a' short s1 = 32767 float f1 = 1.234567  TRACE("The results are") // 輸出 : The results are TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1) // 輸出 : c1 = a, s1 = 32767, f1 = 1.234567  end macro_command</pre>

8. 新增 LB-9059 – 關閉巨集指令 TRACE 功能 (當狀態為 ON)。當設定為 ON 時，TRACE 將不會把數據輸出到 EasyDiagnoser。
9. 也可以直接利用 Utility Manager 執行 EasyDiagnoser.exe，Utility Manager 將會顯示網路上目前存在的 HMI，此時只要選擇要監看通訊狀態的 HMI 即可。請注意 Project Port 的部份應設定與工程檔案所使用的【連接埠】相同。



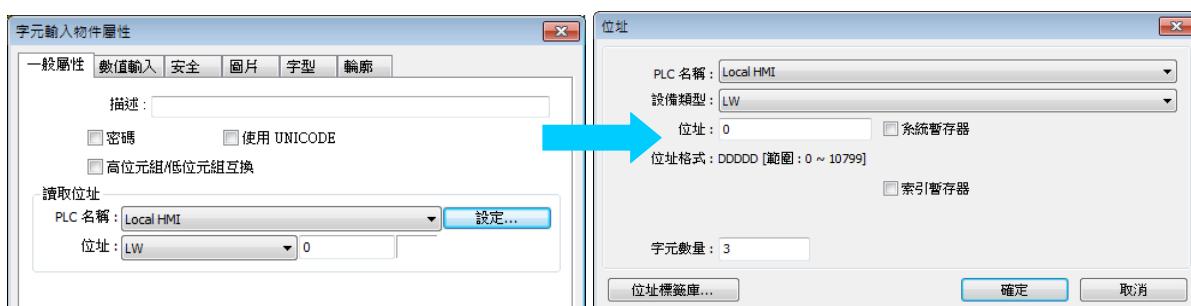
10. 將工程檔下載到 HMI 實際操作。當 EasyDiagnoser 無法連接上需要監視的 HMI 時，一般可能的原因是 HMI 未上電。或是【連接埠】不正確，可能發生 EasyDiagnoser 不斷連上 HMI 又斷線的情況。請確定 EasyDiagnoser 應設定與工程檔案相同的 Port No.，更改方式如前面所述。
11. 當 EasyDiagnoser 成功與 HMI 連結後，只要執行 macro\_0，即可發現【輸出】窗口顯示目

前 TRACE 的執行結果。



## 18.15. 字串處理函式使用方法

巨集指令提供字串處理函式，令使用者可以很方便的對字串進行各種操作。所謂字串，是由一連串的 ASCII 字元組成，每一個 ASCII 字元佔用 1 位元組 (byte)，在 16 位元暫存器中是以低位元組優先方式儲存。舉例來說，我們利用一個【文字輸入】物件在 LW-0 ~ LW-2 的位置，寫入一條字串 "abcdef"，設定如下：



在此【文字輸入】物件輸入"abcdef"：



此字串在 LW-0 ~ LW-2 中的存放方式如下圖所示 (LB 代表低位元組，HB 代表高位元組)：

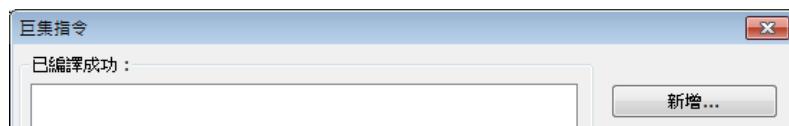
	HB	LB
LW0	'B'	'A'
LW1	'D'	'C'
LW2	'F'	'E'
LW3		
LW4		
LW5		

由於 [文字輸入] 物件顯示的資料長度單位為 word，而每個 ASCII 字元長度為一個 byte，所以每次最少會顯示兩個字元，此部份在 [文字輸入] 物件的章節有詳細的說明。因此上面的例子中，設定 [文字輸入] 物件的字組數量為 3，表示最多可輸入 / 顯示 6 個 ASCII 字元。

下面將舉例說明如何從新增一個巨集指令的方法開始，一直到執行模擬為止，帶您一步步建立一個可執行的工程檔，以實際體會字串處理函數強大的功能。

### 1. 以下說明如何從暫存器讀入與寫入一個字串。

請新增一個巨集指令：



在巨集指令編輯器編輯以下內容：

```

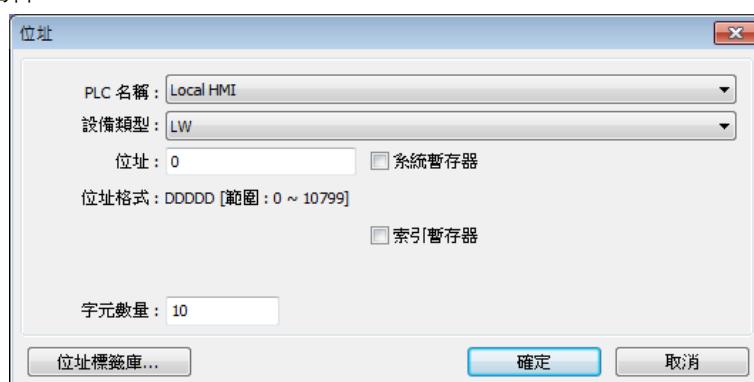
1  macro_command main()
2
3
4  char str[20]
5
6  StringGet(str[0], "Local HMI", LW, 0, 20)
7  StringSet(str[0], "Local HMI", LW, 50, 20)
8
9  end macro_command

```

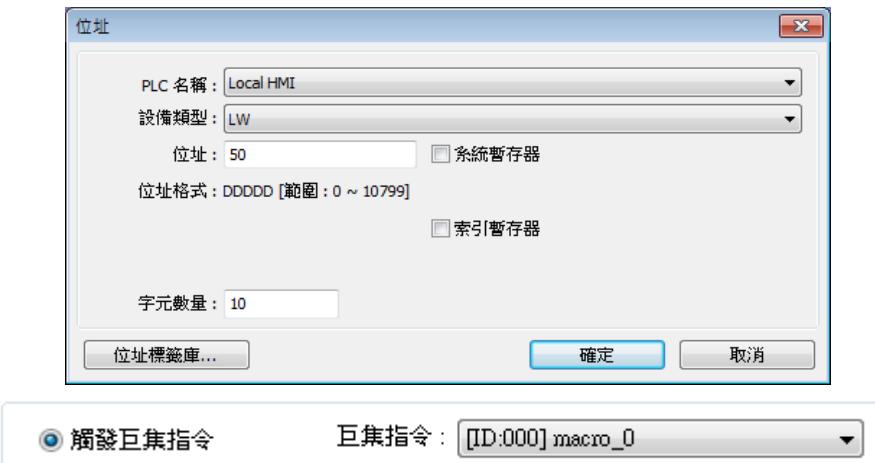
此巨集指令的目的是利用 StringGet 函數從暫存器中讀入一條字串放入字元陣列 str 中，並利用 StringSet 輸出 str 的內容。

接著在工程檔案的第 10 頁分別加上 [文字輸入] 與 [功能鍵] 物件，物件的設定內容請參考下圖，[功能鍵] 物件用來執行 macro\_0。

[文字輸入] 物件



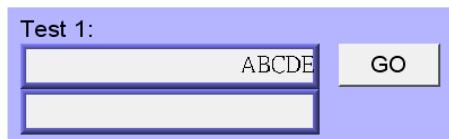
[功能鍵] 物件



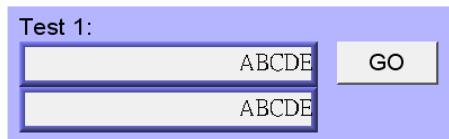
最後編譯 已完成的工程檔案並執行離線 (off-line) 或線上 (on-line) 模擬，並在模擬畫面按照以下步驟操作：

**Step 1.** 輸入字串

**Step 2.** 按下“GO”按鈕



**Step 3.** 顯示結果



**2.** 字串之初始化。

在巨集指令編輯器編輯以下內容：

```

1
2     macro_command main()
3
4     char str1[20]="abcde"
5     char str2[20]={'a','b','c','d','e'}
6
7     StringSet(str1[0], "Local HMI", LW, 0, 20)
8     StringSet(str2[0], "Local HMI", LW, 50, 20)
9
10    end macro_command

```

用雙引號 (" ") 刮起來的內容視為一個字串。str1 是以字串方式初始化，而 str2 是以字元陣列方式初始化。



以字串方式初始化時巨集編譯器會在字串結尾後面加上結束符號 '\0' 代表字串結束。利用

**StringSet** 將字串寫入暫存器時遇到結束符號便會停止繼續寫入陣列後面的內容。即使 **data count** 被設為大於字串長度的數值，顯示字串時只會顯示結束符號之前的內容。

以字元陣列方式初始化時陣列內容並不會被視為一個字串，因此也不會加上結束符號 '\0'。寫入暫存器的字元數會依據 **data count** 所設定的數值決定。

### 3. 一個簡單的帳號密碼登入頁面。

在巨集指令編輯器編輯以下內容(巨集指令 [ID:001] macro\_1)：

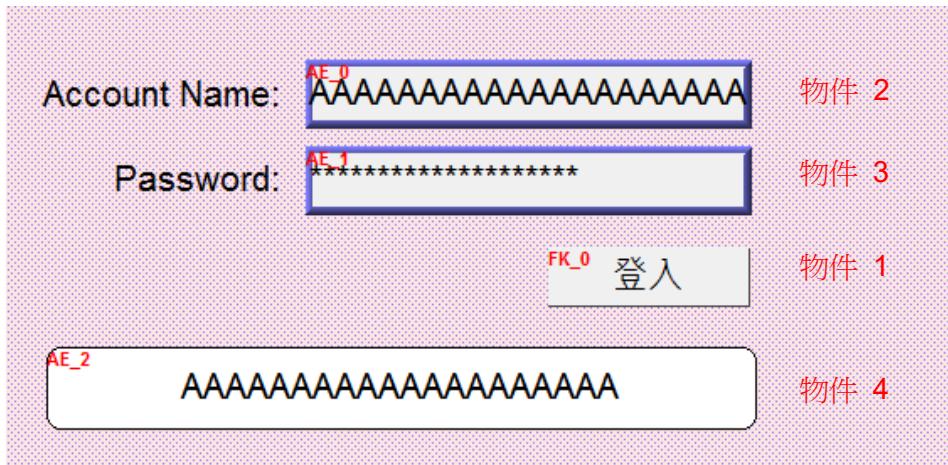
```

1  macro_command main()
2
3
4  char name[20]="admin"
5  char password[20]="123456"
6  char name_input[20], password_input[20]
7  char message_success[40]="Success! Access Accepted."
8  char message_fail[40]="Fail! Access Denied."
9  char message_clear[40]
10 bool name_match=false, password_match=false
11
12 StringGet(name_input[0], "Local HMI", LW, 0, 20)
13 StringGet(password_input[0], "Local HMI", LW, 50, 20)
14
15 name_match = StringCompare(name_input[0], name[0])
16 password_match = StringCompare(password_input[0], password[0])
17
18 FILL(message_clear[0], 0x20, 40) //FILL with white space
19 StringSet(message_clear[0], "Local HMI", LW, 100, 40)
20
21 if (name_match==true and password_match==true) then
22     StringSet(message_success[0], "Local HMI", LW, 100, 40)
23 else
24     StringSet(message_fail[0], "Local HMI", LW, 100, 40)
25 end if
26
27 end macro_command

```

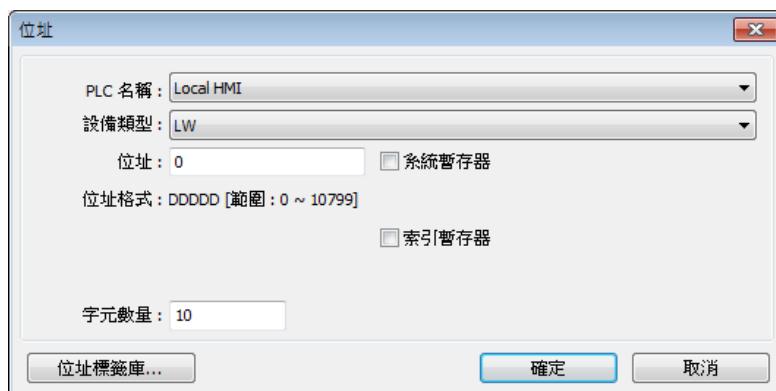
此巨集將利用 **[StringGet]** 取得使用者輸入的帳號密碼的字串，分別放入 **name\_input** 與 **password\_input** 兩個陣列中。接著利用 **[StringCompare]** 比對帳號密碼，若帳號相符，則 **name\_match** 設為 **true**；若密碼相符，則 **password\_match** 設為 **true**。最後檢查 **name\_match** 與 **password\_match** 的值，若同時為 **true**，表示登入成功，並印出 "Success! Access Accepted." 字串。若其中之一不相符，則印出 "Fail! Access Denied." 字串。

接著在工程檔案的第 10 頁分別加上 **[文字輸入]**  與 **[功能鍵]**  物件，物件的設定內容請參考下圖，**[功能鍵]** 物件用來執行 **macro\_1**。

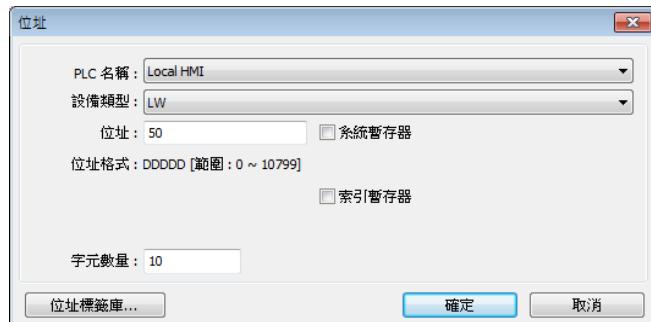


物件 1：[功能鍵] 物件 ，點選 [觸發巨集指令] 並選擇巨集指令 [ID:001] macro\_1。

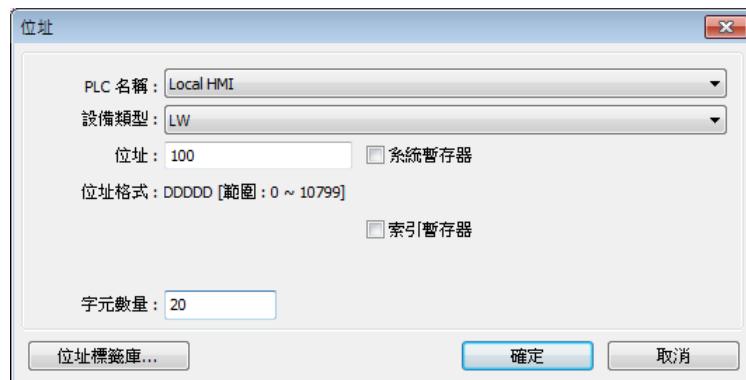
物件 2：[文字輸入] 物件



物件 3：[文字輸入] 物件

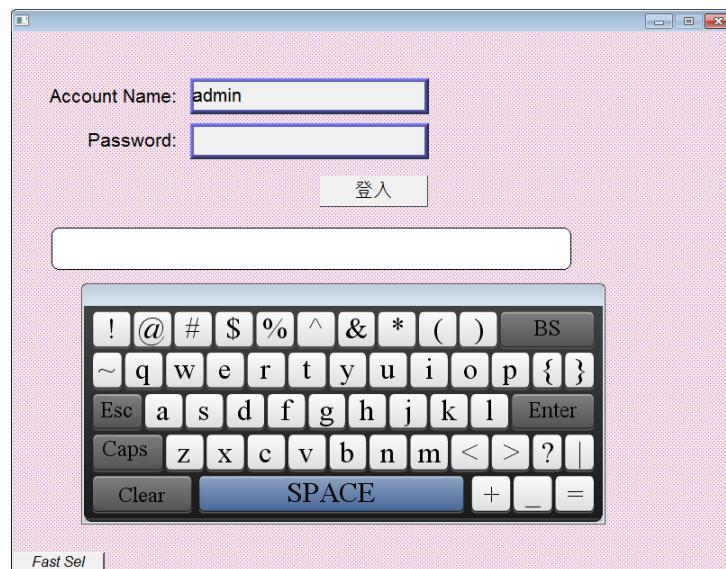


物件 4：[文字顯示] 物件 

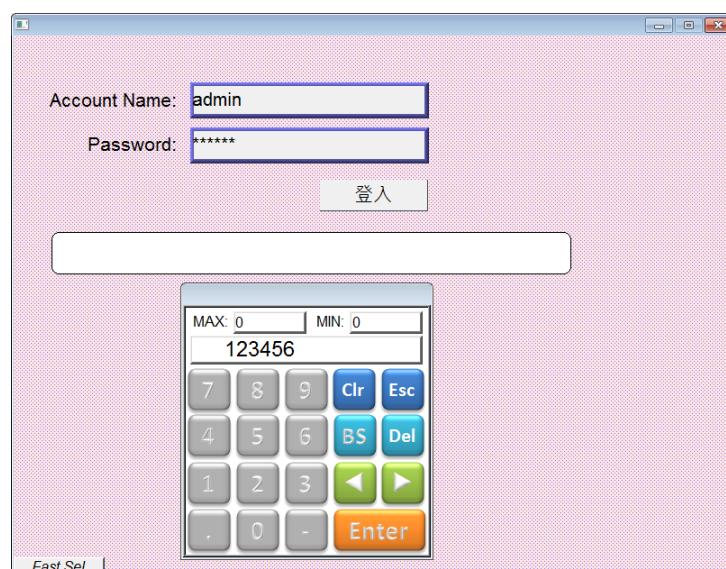


最後編譯  已完成的工程檔案並執行離線 (off-line)  或線上(on-line)  模擬，並在模擬畫面按照以下步驟操作：

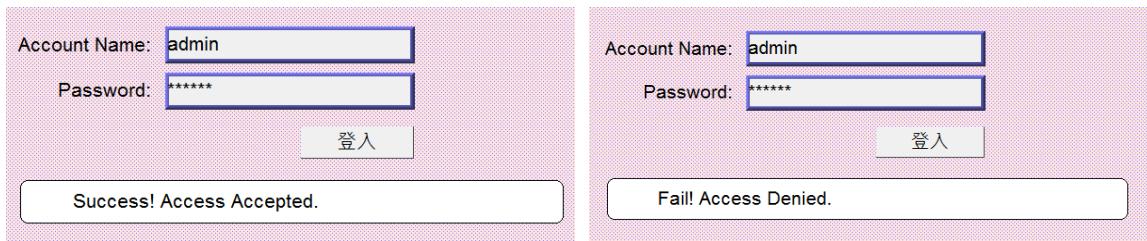
**1.** 輸入帳號。



**2.** 輸入密碼並按下登入鍵。

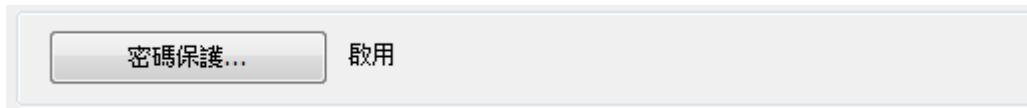


3. 顯示登入成功或登入失敗。



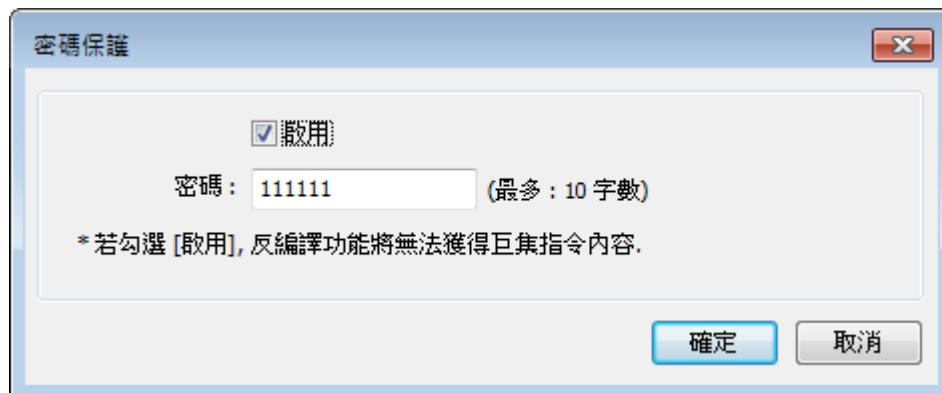
## 18.16. 巨集指令密碼保護

巨集指令密碼保護密碼保護分為兩層：所有巨集密碼保護與單一巨集密碼保護  
所有巨集保護：

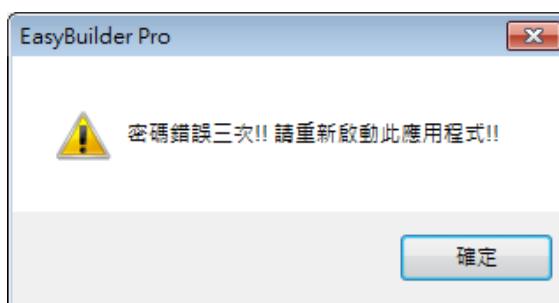


在所有巨集列表提供【密碼保護】選項，當點選【密碼保護】後勾選【啟用】按鈕可以設定密碼，密碼最多不可超過 10 個字元 (只支援 ASCII 文字，並區分大小寫，例如可以輸入 "a\$#\*hFds")。

當設定所有巨集的【密碼保護】功能後，用戶欲開啟巨集時，需先輸入正確的密碼。



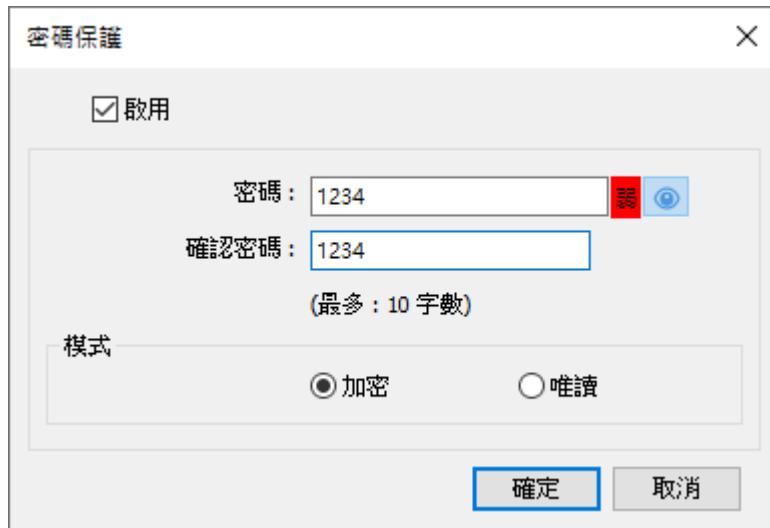
當輸入不正確的密碼三次，需重新啟動 EasyBuilder Pro，才能重新輸入密碼。



### 單一巨集密碼保護：

在巨集指令編輯器裡提供【密碼保護】選項，當點選【密碼保護】後勾選【啟用】按鈕可以設定密

碼。密碼最多不可超過 10 個字元 (只支援 ASCII 文字，並區分大小寫，例如可以輸入 "a\$#\*hFds")。



密碼保護方式分為 [加密] 與 [唯讀]。

#### [加密]

對巨集內容加密，從巨集列表點選巨集欲進入編輯窗口時需先正確輸入密碼。

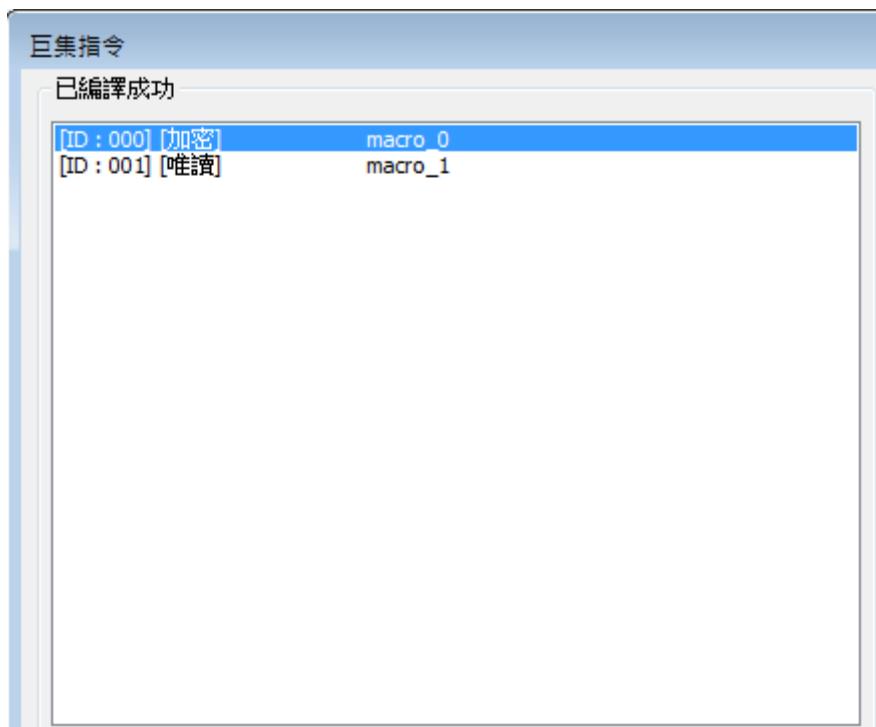
密碼輸入錯誤三次，將提醒需重啟 EasyBuilder Pro (每一項巨集所需次數各自獨立)。

#### [唯讀]

用戶將僅能檢視巨集的內容，無法編輯。

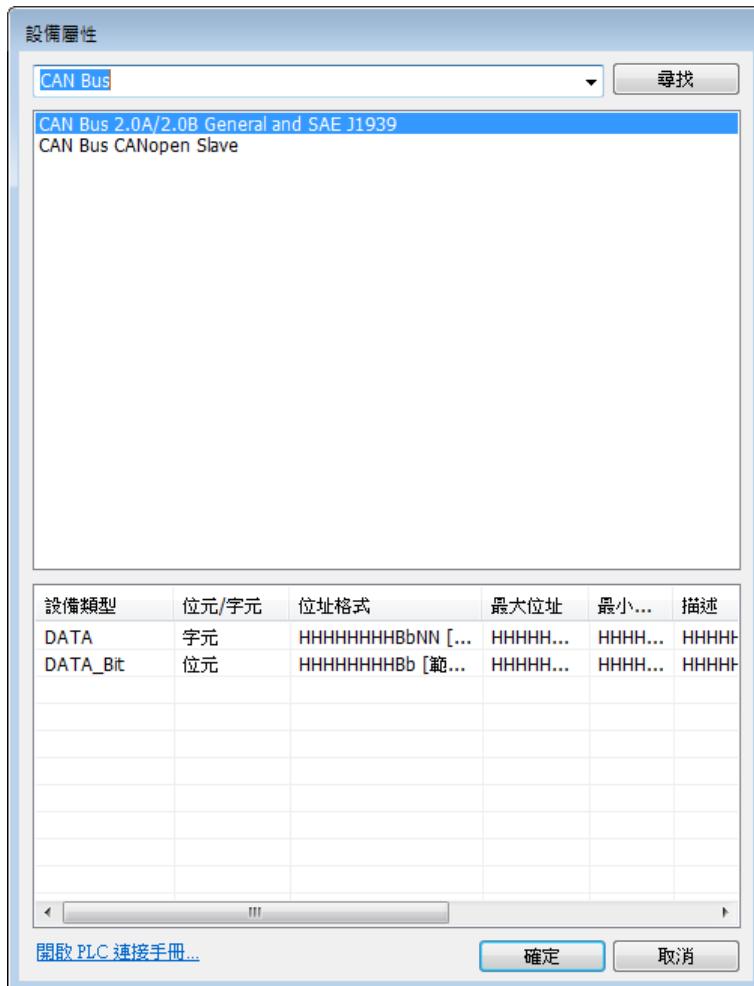
選用此模式時可以由巨集列表直接進入巨集指令編輯器，但在進到[密碼保護]設定窗口時就需要輸入正確的密碼。密碼的輸入規則與 [加密] 相同，限制只能錯誤輸入三次。

巨集列表中會顯示單一巨集所使用的保護模式。



## 18.17. 巨集支援使用變數讀寫 CANbus 地址

當使用 CAN Bus 2.0A/2.0B General and SAE J1939 驅動程式時，會發現該驅動程式擁有兩種地址類型：DATA 與 DATA\_Bit，資料格式的描述如下表所示。



資料格式	描述
<b>DATA</b>	HHHHHHHH: ID B: Byte 位置(1~8)
HHHHHHHHBbNN	b: Bit 位置(1~8) NN: Bit 數量(1~64)
<b>DATA_Bit</b>	HHHHHHHH: ID B: Byte 位置(1~8)
HHHHHHHHBb	b: Bit 位置(1~8)

由於 ID 是使用 16 進制表示，位置與數量則是使用 10 進制表示，請參考下方的使用方式。

## 範例

未採用變數地址的表示法

```
short f  
GetData(f, "CAN Device", DATA, 4e55108, 1)  
GetData(f, "CAN Device", DATA, 4e65108, 1)
```

採用變數地址的表示法

```
short f  
unsigned int address = 0x4e55108  
GetData(f, "CAN Device", DATA, address, 1)  
address = address + 0x10000// == 0x4e65108  
GetData(f, "CAN Device", DATA, address, 1)
```

注意事項：

1. 將變數宣告為 Unsigned int，並使用 16 進制來表示地址。
2. 因為 Unsigned int 的長度為 4 bytes，而且 Bb 與 NN 分別都需要 1 byte 的空間。所以當使用變數作為地址參數時，讀寫 DATA\_Bit 地址類型時，地址格式將變動為 HHHHHH~~Bb~~ (最大 ID 只能到 0xffff)；讀寫 DATA 地址類型時，地址格式將變動為 HHHH~~Bb~~NN (最大 ID 只能到 0xffff)。

# 19. 如何將 HMI 設定成 MODBUS 裝置

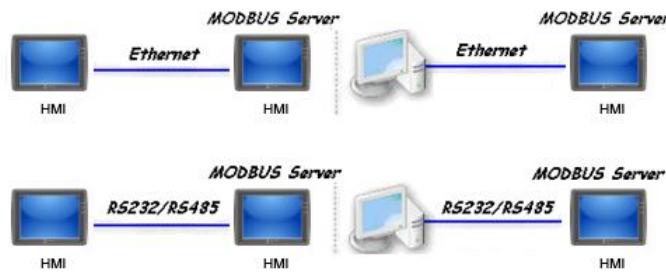
本章節說明如何將 HMI 設定成 MODBUS 裝置。

19.1. 概要 .....	19-2
19.2. 建立一個 MODBUS Server 裝置.....	19-2
19.3. 讀寫一個 MODBUS Server 裝置.....	19-4
19.4. 線上更改 MODBUS Server 站號.....	19-7
19.5. 關於 MODBUS 各位址的說明 .....	19-7

## 19.1. 概要

將 HMI 設定成 MODBUS 裝置後，透過 MODBUS 協議即可讀寫 HMI 上的數據。

下圖顯示 HMI 被設定成 MODBUS 裝置 (又稱為 MODBUS Server)，HMI、PC 或其它裝置只需使用 MODBUS 協議，透過 Ethernet 或 RS-232 / RS-485 介面，即可讀寫 HMI 上的數據。



## 19.2. 建立一個 MODBUS Server 裝置

- 要將 HMI 設定為 MODBUS 裝置，首先需在 [系統參數設定] » [設備清單] 中增加一個新的設備，此時 PLC 類型需選擇 MODBUS Server，可以選擇的 [PLC 介面] 如下圖所示。

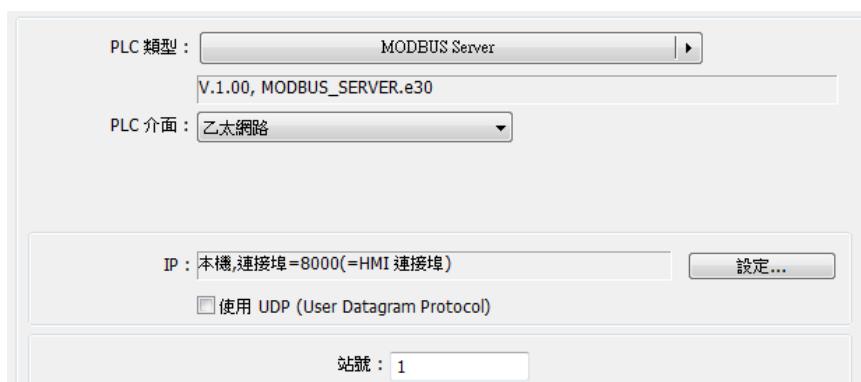


## 如何將 HMI 設定成 MODBUS 裝置

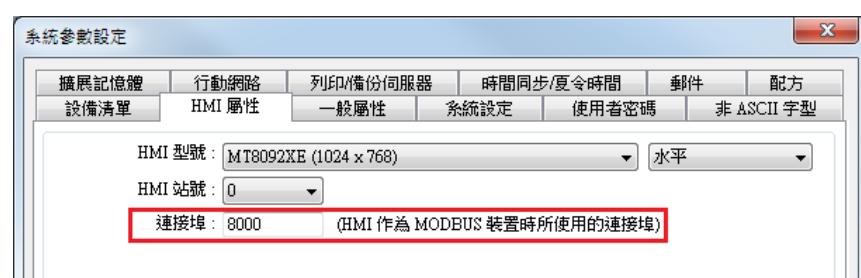
2. 當 PLC 介面選擇 [RS-232] 或 [RS-485] 時，需選擇使用的 [COM] (COM 1 ~ COM 3)，並設定正確的通訊參數。如下圖所示，此時 MODBUS Server 的 [站號] 設定為 1。  
點選 [設定]，可以設定 [限制 LW 最大讀取/寫入位址]。當工程檔案的物件使用 LW 暫存器時，超過此範圍的位址將不會被 Modbus 客戶端讀/寫。



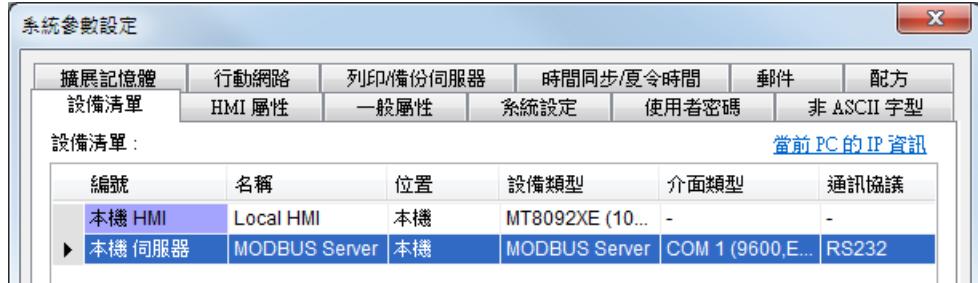
當 PLC 介面選擇 [乙太網路] 時，需設定 [連接埠號]。



因 MODBUS Server 與 HMI 須使用相同的 [連接埠]，若要更改 MODBUS Server 的連接埠，需在 [系統參數設定] » [HMI 屬性] 頁籤中修改。



3. 在按下確定鍵後，即可在【設備清單】中發現一個新的裝置：MODBUS Server，此時即完成 MODBUS 裝置的設定，在完成 .emtp 檔案的編譯並將獲得的 .exob 檔案下載到 HMI 後，即可透過 MODBUS 協議讀寫 HMI 上的數據。



### Note

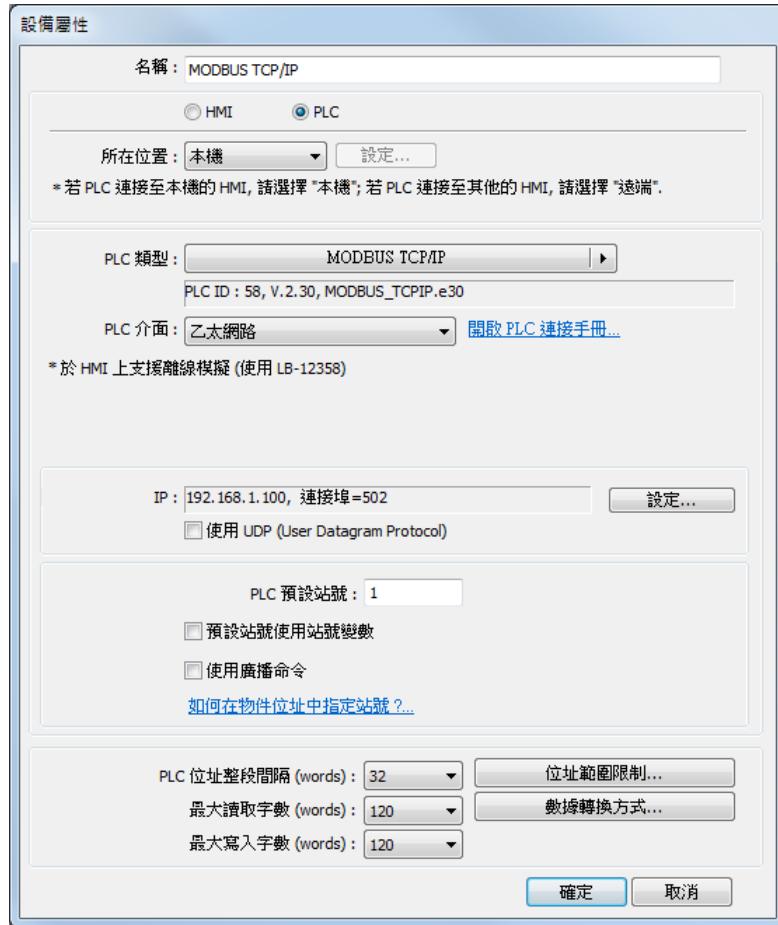
- cMT / cMT X 系列在 PLC 介面選擇 [乙太網路] 時，連接埠可自行輸入。



## 19.3. 讀寫一個 MODBUS Server 裝置

兩台 HMI 可以透過設定成 MODBUS Client (客戶端) 和 Server (伺服器) 相互通訊。

- 在 Client 端的設備清單中，需增加一個新的設備。若 Client 端使用 [乙太網路] 介面，則 [PLC 類型] 需挑選 MODBUS TCP/IP，並正確設定 [IP 位址] (即 Server 端所在位置的 IP)、[連接埠] 與 [站號]。

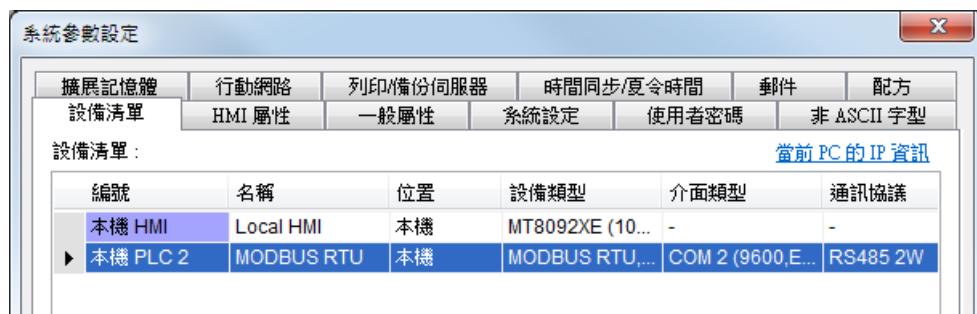


若 Client 端要使用 [RS-232] 或 [RS-485] 介面。則 [PLC 類型] 需挑選 MODBUS RTU，並正確設定各項通訊參數。

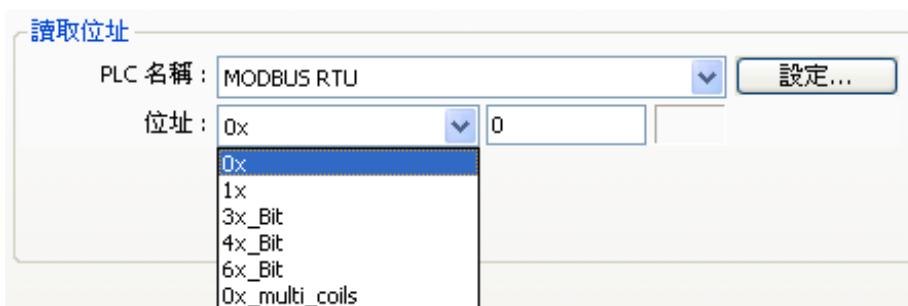
## 如何將 HMI 設定成 MODBUS 裝置



2. 完成各項設定並按下確定鍵後，即可在 [設備清單] 中發現一個新的設備 “MODBUS RTU”。



3. 開啟各個物件的設定頁，在 [PLC 名稱] 選擇 MODBUS RTU 後，即可設定 MODBUS 裝置的各項讀寫位址。



此時因被讀寫的裝置 (Server 端) 為 HMI，所以實際讀寫的位置之對應關係如下：

讀寫 0x/1x (1 ~ 12800)	對應到 讀寫 LB (0 ~ 12799)
讀寫 3x/4x/5x (1 ~ 9999)	對應到 讀寫 LW (0 ~ 9998)
讀寫 3x/4x/5x (10000 ~ 65535)	對應到 讀寫 RW (0 ~ 55535)

## 19.4. 線上更改 MODBUS Server 站號

EasyBuilder Pro 提供下列系統暫存器，讓使用者可以線上更改 MODBUS Server 所使用的站號。

LW-9541	MODBUS/ASCII server 站號 (COM 1)
LW-9542	MODBUS/ASCII server 站號 (COM 2)
LW-9543	MODBUS/ASCII server 站號 (COM 3)
LW-9544	MODBUS/ASCII server 站號 (乙太網路)

## 19.5. 關於 MODBUS 各位址的說明

EasyBuilder Pro 中 MODBUS 協定的設備類型為 0x、1x、3x、4x、5x、6x，還有 3x\_bit、4x\_bit 等等，下面將分別說明這些設備類型在 MODBUS 協定中支援的功能碼。

0x	是個可讀可寫的設備類型，相當於操作 PLC 的輸出點。該設備類型讀位元狀態時發出的功能碼為 01H，寫位元狀態時發出的功能碼為 05H。寫多個位元暫存器時發出的功能碼為 0fH。
1x	是個唯讀的設備類型，相當於讀 PLC 的輸入點。讀位元狀態時發出的功能碼為 02H。
3x	是個唯讀的設備類型，相當於讀 PLC 的唯讀資料暫存器。讀數據時發出的功能碼為 04H。
4x	是個可讀可寫的設備類型，相當於操作 PLC 的資料暫存器。當讀數據時發出的功能碼為 03H，當寫資料時發出的功能碼為 10H。
5x	該設備類型與 4x 的設備類型屬性是一樣的。即發出讀寫的功能碼完全一樣。不同之處在於，當為雙字組時，若 32_bit unsigned 格式的資料，使用 5x 和 4x 兩種設備類型分別讀取資料時，高字組和低字組的位置是顛倒的。若使用 4x 設備類型讀到的資料是 0x1234，那麼使用 5x 設備類型讀取的資料即為 0x3412。
6x	是一個可讀可寫的設備類型，讀數據時發出的功能碼也是 03H，與 4x 不同之處在於寫資料的時候，發出的功能碼為 06H，即寫單個暫存器的資料。
3x_bit	該設備類型支援的功能碼與 3x 設備類型完全一致，不同之處在於 3x 是讀數據，而 3x_bit 是讀數據中的某一個 bit 的狀態。

## 如何將 HMI 設定成 MODBUS 裝置

**4x\_bit** 該設備類型支援的功能碼與 4x 設備類型完全一致，不同之處在於 4x 是讀數據，而 4x\_bit 是讀數據中的某一個 bit 的狀態。

**6x\_bit** 該設備類型支援的功能碼與 6x 設備類型完全一致，不同之處在於 6x 是讀數據，而 6x\_bit 是讀數據中的某一個 bit 的狀態。



更多資訊請參考《37 MODBUS TCP/IP 閘道功能》。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

# 20. 如何使用條碼掃描器

本章節說明如何使用條碼掃描器及連接步驟。

20.1. 概要 .....	20-2
20.2. 連接條碼掃描器的步驟 .....	20-2
20.3. CDC/POS 模式 .....	20-6

## 如何使用條碼掃描器

### 20.1. 概要

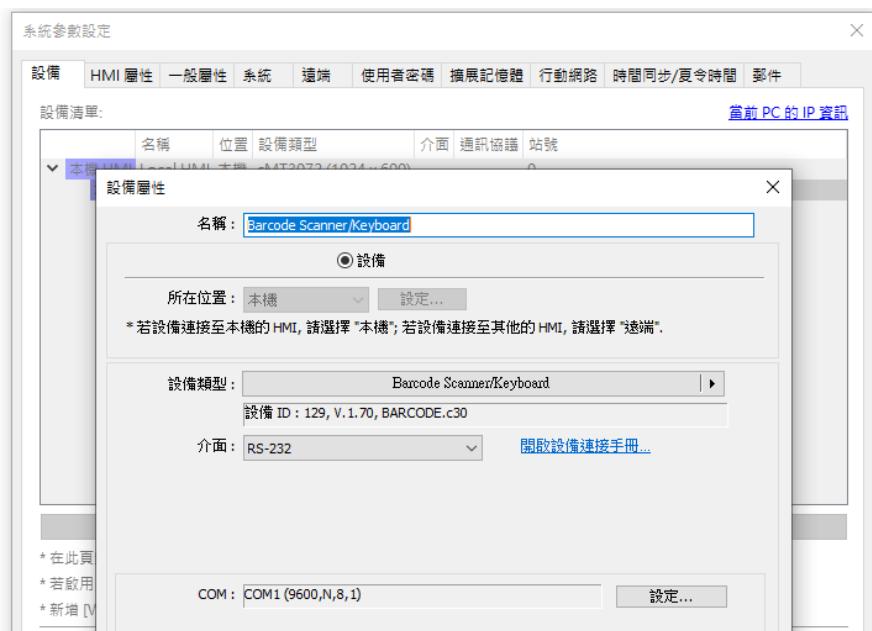
HMI 支援透過下列通訊埠連接條碼掃描器：

- USB
- COM Port

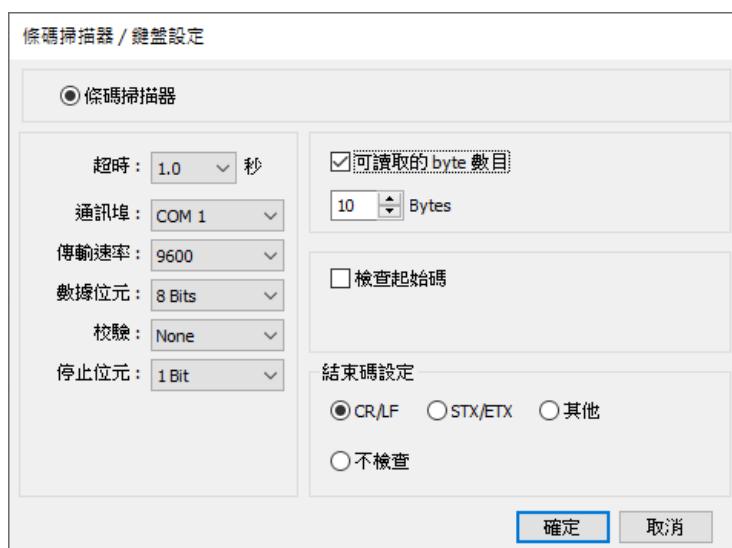
欲連接條碼掃描器，請先在設備清單中增加一個新裝置。

### 20.2. 連接條碼掃描器的步驟

1. 在 [系統參數設定] » [設備清單] 頁籤中增加一個裝置。



2. 按下 [設定] 按鈕並完成設定。



設定	描述
<b>超時</b>	當勾選 [條碼掃描器] 時，若該裝置讀取速度較慢，可將超時設定加長以讀取完整的資料。 當勾選 [鍵盤] 時，可設定透過鍵盤輸入資料的有效時間範圍，系統將於開始輸入資料時才計時。
<b>通訊埠</b>	
<b>傳輸速率</b>	
<b>數據位元</b>	當選用 COM 介面時，須正確設定條碼掃描器的通訊參數；若選用 USB 介面，則無須設定通訊參數。
<b>校驗</b>	
<b>停止位元</b>	
<b>可讀取的 byte 數目</b>	若勾選，則可以限制條碼掃描器讀取的 byte 數目，以避免裝置讀取過多的數據。此項設定值範圍為 10 ~ 1024。 注意：若實際讀取的 byte 數目超過設定值，將無法讀取。
<b>起始碼</b>	條碼掃描器所讀取到的第一個數據必須與起始碼相同，系統才會將讀取的數據視為是合法的輸入，否則將會忽略讀取的數據。 起始碼不會被存放在條碼掃描器所對應的位址中。
<b>不檢查</b>	若不使用起始碼，系統會將全部讀取到的數據存放在條碼掃描器對應的位址中。
<b>STX</b>	以 0x02 做為起始碼。
<b>ETX</b>	以 0x03 做為起始碼。
<b>其他</b>	由使用者自訂數據的起始碼。 例如：起始碼為 255 (0xff)，且讀取到的數據為 <b>0xff 0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37</b> 但實際存放在條碼掃描器對應位址中的數據為 <b>0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37</b>
<b>結束碼設定</b>	結束碼設定 結束碼可用來標示數據的結尾，當讀取到結束碼時，表示讀取到一筆完整的數據。 <b>不檢查</b> 若不使用結束碼，系統會將全部讀取到的數據存放在條碼掃描器對應的位址中。

**STX**

以 0x02 做為結束碼。

**ETX**

以 0x03 做為結束碼。

**STX/ETX**

以 0x02 或 0x03 做為結束碼。

**CR/LF**

以 0x0a 或 0x0d 做為結束碼。

**其他**

由使用者自訂數據的結束碼。

例如: 結束碼為 55 (0x37)，且讀取到的數據為

0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37

但實際存放在條碼掃描器對應位址中的數據為

0x34 0x39 0x31 0x32 0x30 0x30 0x34

完成以上各項設定後，即可在 [設備清單] 中發現一個新的條碼掃描器裝置。

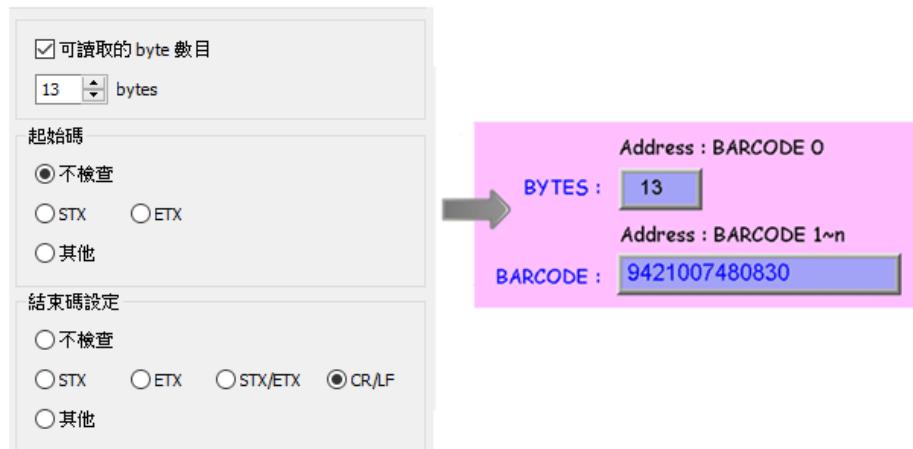
此時在物件的設定屬性頁中的 [設備] 即可選擇條碼掃描器，並可使用相關的位址類型。

位址類型	位址名稱	描述
位元	FLAG	<b>FLAG 0:</b> 指示數據是否讀取完成。在讀取到數據時，系統會自動將 FLAG 先設定為 OFF，待讀取成功後再設定為 ON。
	RESET	<b>RESET 0:</b> 當設為 ON 時，可清除 BARCODE 和 RESULT 內的數據。
	CONNECT_STA	<b>CONNECT_STATUS 0:</b>
	TUS	指示是否接上 USB 介面的條碼掃瞄器設備，當狀態為 ON 時表示已接上。
字組	BARCODE	<b>BARCODE 0:</b> 記錄目前讀取到的 byte 數目。 <b>BARCODE 1 ~ n:</b> 存放裝置讀取的數據。
	RESULT	<b>RESULT 0:</b> 指示 BARCODE 的讀取結果。各項數據的表示意義如下： 0x00: 等待讀取 BARCODE。 0x01: 讀取 BARCODE 成功。 0x02: BARCODE 格式錯誤。 0x03: 在啟用 [可讀取的 byte 數目] 時，所讀取的數據長度超過所設定的大小。 0x04: 在啟用 [檢查起始碼] 時，所讀取的數據不符合設定值。

0x05: 在啟用 [結束碼] 時，所讀取的數據不符合設定值。

## 範例 1

假設目前條碼掃描器的設定如下圖，且讀取到的條碼為 9421007480830，圖中的數值物件 (BYTES) 的位址為 BARCODE 0，字元物件 (BARCODE) 的位址為 BARCODE 1 ~ n。



此時條碼掃描器裝置對應的位址所存放的數據如下：

條碼掃描器對應位址	數據
BARCODE 0	13 bytes (十進制) 但實際存入位址中的數據為 14 bytes = 7 words 也就是當讀取 byte 數目為奇數時，系統會自動加上一個 byte 的數據 (0x00)
BARCODE 1	3439 (HEX)
BARCODE 2	3132 (HEX)
BARCODE 3	3030 (HEX)
BARCODE 4	3437 (HEX)
BARCODE 5	3038 (HEX)
BARCODE 6	3338 (HEX)
BARCODE 7	0030 (HEX)

### Note

- 每台 HMI 只支援連接一台 USB 介面的條碼掃描器裝置。當工程檔案的裝置列表中包含 USB 條碼掃描器裝置時，系統暫存器 LB-9064 [啟用 USB 條碼掃描器裝置 (鍵盤功能關閉) (當狀態為 ON)] 將自動被設定為 ON。若此時需恢復 USB 鍵盤的功能並暫停使用 USB 條碼掃描器，可以將 LB-9064 設定為 OFF。

## 如何使用條碼掃描器

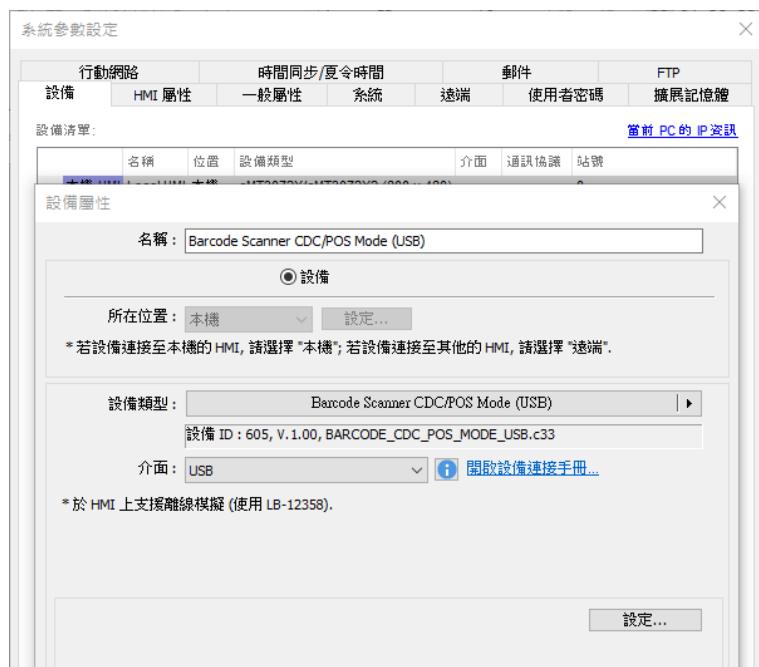


請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 20.3. CDC/POS 模式

若條碼掃描器有支援 CDC 或 POS 模式，可選擇 Barcode Scanner CDC/POS Mode (USB)驅動程式通訊，優化條碼掃描器的讀取時間。

- 在 [系統參數設定] » [設備清單] 頁籤中增加一個裝置。



- 按下 [設定] 按鈕並完成設定。



設定	描述
超時	當條碼掃描器沒有回應 HMI 的命令超過超時設定，則視為該設備斷線。 CDC 模式：此模式沒有結束字元，無法確認數據是否已完整讀取，因此超時的設定決定數據收取結束的時間。
通訊延時 模式	增加通訊延時於 HMI 與條碼掃描器的通訊之中。 可選擇 CDC 或 POS 模式。

完成以上各項設定後，即可在 [設備清單] 中發現一個新的條碼掃描器裝置。

此時在物件的設定屬性頁中的 [設備] 即可選擇條碼掃描器，並可使用相關的位址類型。

位址類型	位址名稱	描述
位元	flag	指示數據是否讀取完成。在讀取到數據時，系統會自動將 flag 先設定為 OFF，待讀取成功後再設定為 ON。
字組	data	<b>data 0~1999:</b> 存放裝置讀取的數據。
	state	<b>state 0:</b> 數值為 0 代表系統未偵測到條碼掃描器；數值為 1 代表條碼掃描器正常連接。
	len	<b>len 0:</b> 記錄目前讀取到的 byte 數目。

# 21. 乙太網路通訊與多台人機連線

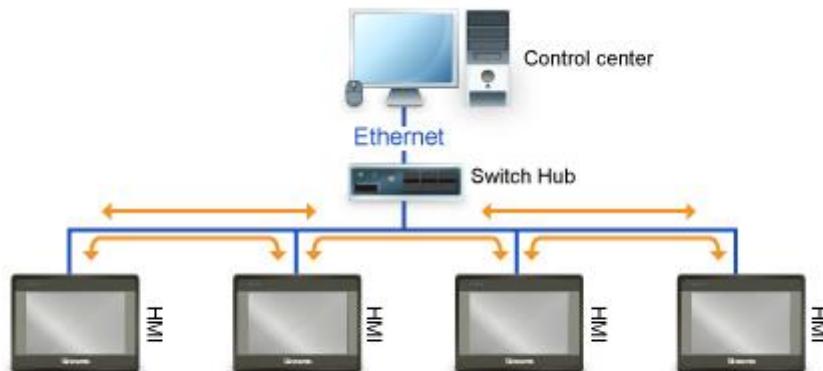
本章節說明如何使用乙太網路連接多種設備。

21.1. 概要 .....	21-2
21.2. HMI 與 HMI 間的通訊 .....	21-2
21.3. PC 與 HMI 間的通訊 .....	21-3
21.4. 控制連接在其他 HMI 上的 PLC .....	21-4

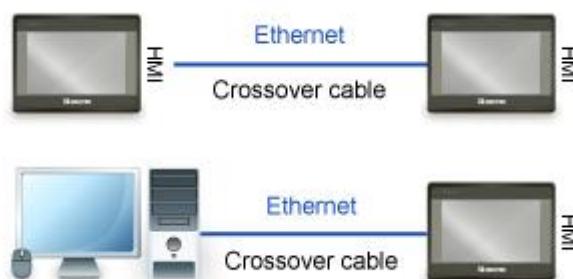
## 21.1. 概要

乙太網路連線的方式分為兩種：

1. 使用 RJ45 平行網路線與集線器。



2. 使用 RJ45 跳接網路線，不需使用集線器，但只限使用在一對一連線的情況下 (HMI 對 HMI，或 PC 對 HMI)。



透過乙太網路連線，系統提供了下列三種數據交換的方式：

1. HMI 與 HMI 間的通訊。
2. PC 與 HMI 間的通訊。
3. 控制連接在其他 HMI 上的 PLC。

## 21.2. HMI 與 HMI 間的通訊

HMI 之間通訊可在 [系統參數設定] 中新增一個遠端 HMI 裝置即可。

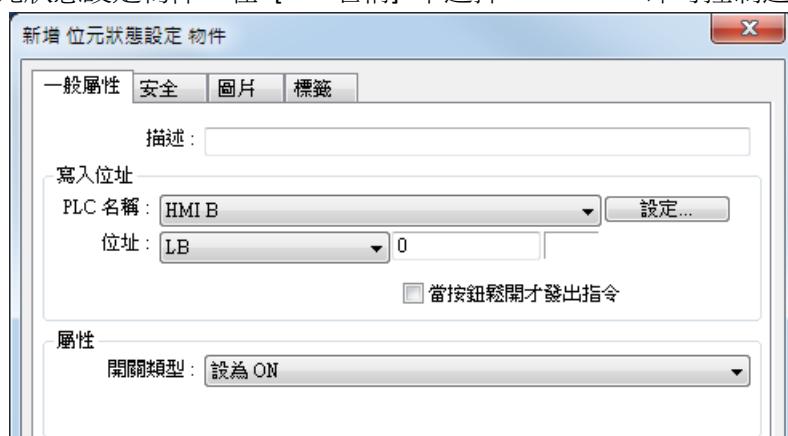
以兩台 HMI 的通訊為例 (HMI A 與 HMI B)，假設 HMI A 欲使用位元狀態設定物件控制 HMI B 的 [LB-0] 位址的內容，則 HMI A 工程檔案的設定步驟如下。



1. 設定各台 HMI 的 IP 位址，假設 HMI A: 192.168.1.1，HMI B: 192.168.1.2。
2. 自 [系統參數設定] » [設備清單]，新增一台遠端 HMI，即為 HMI B (IP: 192.168.1.2)。



3. 設定一個位元狀態設定物件，在 [PLC 名稱] 中選擇 “HMI B”，即可控制遠端 HMI 的位址。

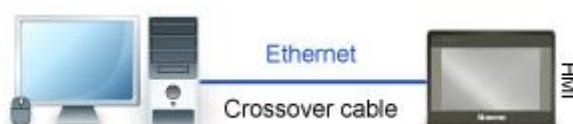


### Note

- 一台 HMI 最多可同時處理來自 64 個不同 HMI 的訪問要求。

## 21.3. PC 與 HMI 間的通訊

透過連線模擬功能，PC 可以藉由乙太網路擷取 HMI 上的數據，並保存在 PC 上。假設 PC 欲通訊的設備為兩台 HMI (HMI A 與 HMI B)，則 PC 端所使用工程檔案的設定步驟如下。

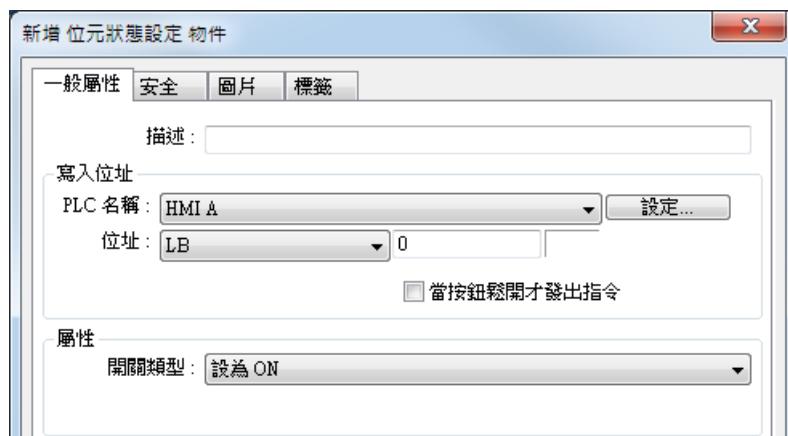


1. 設定各台 HMI 的 IP 位址，假設 HMI A: 192.168.1.1，HMI B: 192.168.1.2。
2. 自 [系統參數設定] » [設備清單]，新增兩台遠端 HMI，分別為 HMI A (IP:192.168.1.1)，與 HMI

B (IP:192.168.1.2)。



- 設定一個位元狀態設定物件，在 [PLC 名稱] 中選擇 “HMI A”，即可控制遠端 HMI A 的位址。同樣的方式也可用於 HMI B。

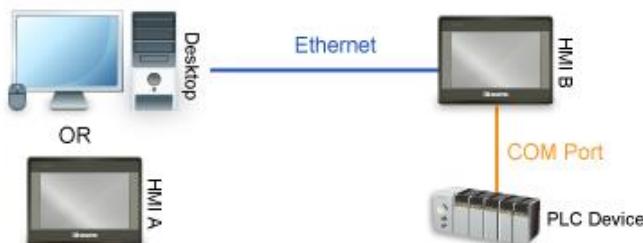


### Note

- 一台 PC 最多可同時控制 64 台遠端 HMI。
- 如上面的例子，HMI 也允許操作 PC 上的數據，此時只需將 PC 視為另一台 HMI 即可，也就是必須在 HMI A / HMI B 使用的工程檔案中新增一台遠端 HMI，並將此遠端 HMI 的 IP 位址指向 PC。

## 21.4. 控制連接在其他 HMI 上的 PLC

透過乙太網路連線，PC 或 HMI 可以操作連接在其他 HMI 上的遠端 PLC。假設現在有一台 PLC 連接到 HMI B 的 COM 1，當 PC 或 HMI A 欲讀取此台 PLC 上的數據，則 PC 端或 HMI A 上所使用的工程檔案設定步驟如下。

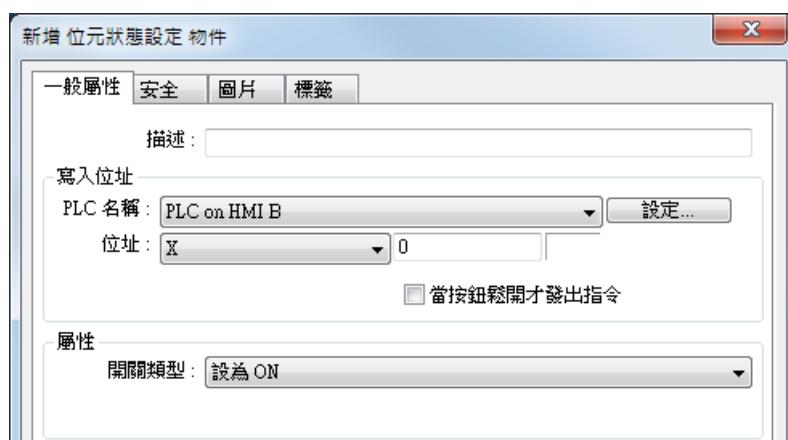


#### 21.4.1. eMT / iE / XE / mTV / iP 系列的設定方法

1. 設定 HMI B 的 IP 位址，假設 HMI B: 192.168.1.2。
2. [系統參數設定] » [設備清單]，新增一台遠端 PLC，將名稱設為 “PLC on HMI B” 並正確設定 PLC 的相關通訊參數。因此台 PLC 是連接在遠端 HMI B 上，所以將遠端 IP 位址指向 HMI B (IP: 192.168.1.2)。

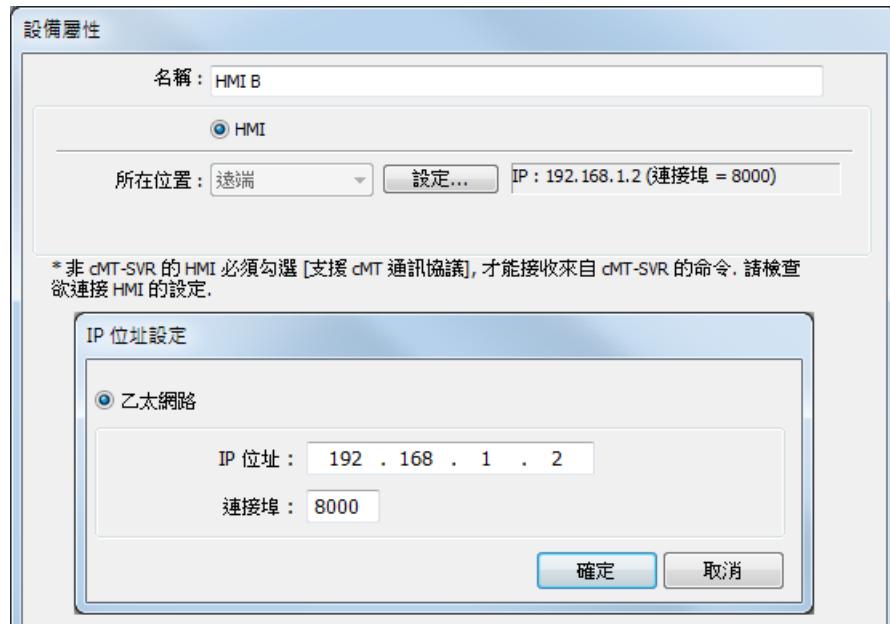


3. 設定一個位元狀態設定物件，在 [PLC 名稱] 中選擇 “PLC on HMI B”，即可控制遠端 HMI B 上的 PLC。

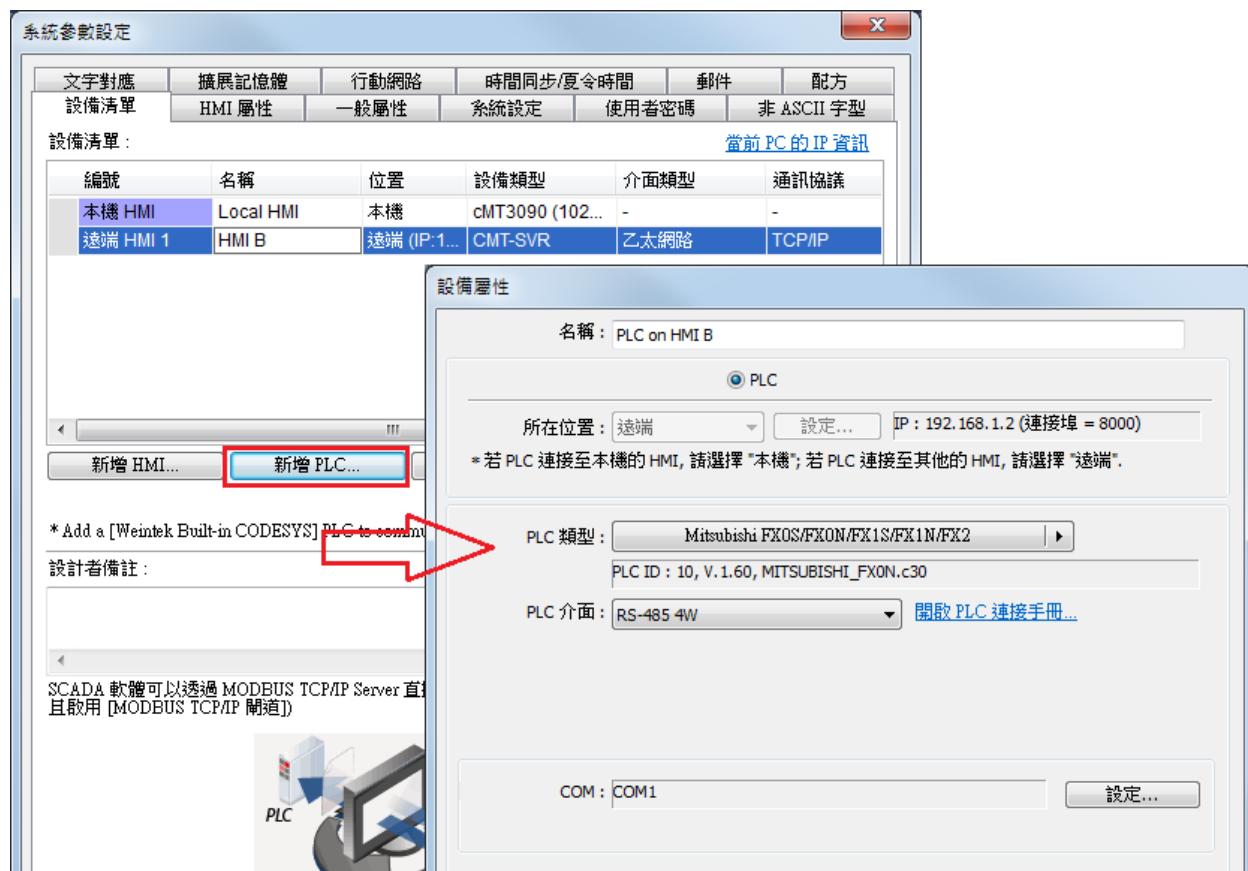


#### 21.4.2. cMT / cMT X 系列的設定方法

1. 設定 HMI B 的 IP 位址，假設 HMI B: 192.168.1.2。
2. [系統參數設定] » [設備清單]，點選 [新增 HMI] 並設定 HMI B 的 IP 位址，假設 HMI B: 192.168.1.2。



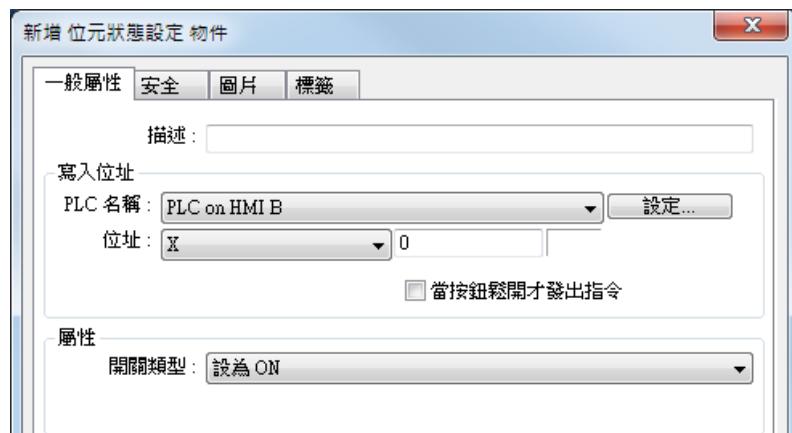
3. 在 HMI B 底下點選 [新增 PLC]，新增一台遠端 PLC，將名稱設為 “PLC on HMI B” 並正確設定 PLC 的相關通訊參數。



4. 建立完成後，可以看到一台遠端的 PLC 設定被建立在遠端 HMI 下面，本機 HMI 代表的是 HMI A，遠端 HMI 1 是 HMI B，遠端 PLC 1 則是 HMI B 所連接的 PLC。



5. 設定一個位元狀態設定物件，在 [PLC 名稱] 中選擇 “PLC on HMI B”，即可控制遠端 HMI B 上的 PLC。



### Note

- cMT / cMT X 系列的遠端 HMI 若為 eMT/iE/XE/mTV，須於 [系統參數設定] » [HMI 屬性] 勾選 [支援 iE/XE/eMT/mTV HMI 通訊協議和 EasyWatch]。同樣地，eMT/iE/XE/mTV 系列的工程檔案上的 [系統參數設定] » [HMI 屬性] 也需勾選 [支援 cMT 通訊協議]，如此 cMT / cMT X 系列與 eMT/iE/XE/mTV 才能通訊。

# 22.位址暫存器

本章節說明各類位址暫存器。

22.1. 概要 .....	22-3
22.2. 本機 HMI 記憶體位址範圍 .....	22-4
22.2.1. 位元位址 .....	22-4
22.2.2. 字組位址 .....	22-4
22.3. 系統暫存器 .....	22-4
22.3.1. HMI 時間 .....	22-5
22.3.2. HMI 操作 .....	22-6
22.3.3. 觸碰位置 .....	22-7
22.3.4. 本機 HMI 網路資訊 .....	22-8
22.3.5. 工程檔案資訊 .....	22-10
22.3.6. 儲存空間管理 .....	22-11
22.3.7. 配方及擴展記憶體 .....	22-12
22.3.8. 資料取樣 .....	22-15
22.3.9. 事件記錄 .....	22-16
22.3.10. 站號變數 .....	22-19
22.3.11. 索引暫存器 .....	22-20
22.3.12. MODBUS Server 通訊 .....	22-22
22.3.13. 通訊參數設定 .....	22-23
22.3.14. 與設備 (COM) 的通訊狀態與控制 .....	22-26
22.3.15. 與設備 (乙太網路) 的通訊狀態與控制 .....	22-28
22.3.16. 與設備 (USB) 的通訊狀態與控制 .....	22-38
22.3.17. 與設備 (CAN Bus) 的通訊狀態與控制 .....	22-38
22.3.18. 與遠端 HMI 的通訊狀態與控制 .....	22-38
22.3.19. 與遠端設備的通訊狀態與控制 .....	22-52
22.3.20. 本機/遠端操作限制 .....	22-54
22.3.21. 通訊錯誤碼 .....	22-54
22.3.22. 驅動程式 ID .....	22-56
22.3.23. DLT645 控制器 .....	22-56
22.3.24. [Device No Response] 視窗控制 .....	22-56
22.3.25. [快選] 視窗控制 .....	22-57
22.3.26. EasyAccess .....	22-57
22.3.27. EasyAccess 2.0 .....	22-58

22.3.28. 遠端列印/備份伺服器.....	22-59
22.3.29. 穿透通訊設定 .....	22-59
22.3.30. VNC 控制 .....	22-60
22.3.31. HMI 和工程檔案識別碼 .....	22-61
22.3.32. USB 安全金鑰.....	22-62
22.3.33. 使用者名稱和密碼 .....	22-62
22.3.34. 巨集 .....	22-63
22.3.35. 輸入物件功能 .....	22-65
22.3.36. 時間同步/夏令時間.....	22-65
22.3.37. 行動網路 .....	22-67
22.3.38. WiFi 設定.....	22-68
22.3.39. OPC UA 伺服器.....	22-70
22.3.40. 郵件 .....	22-70
22.3.41. cMT Viewer 與診斷器.....	22-72
22.3.42. CODESYS 資訊.....	22-74
22.3.43. 其它功能 .....	22-74

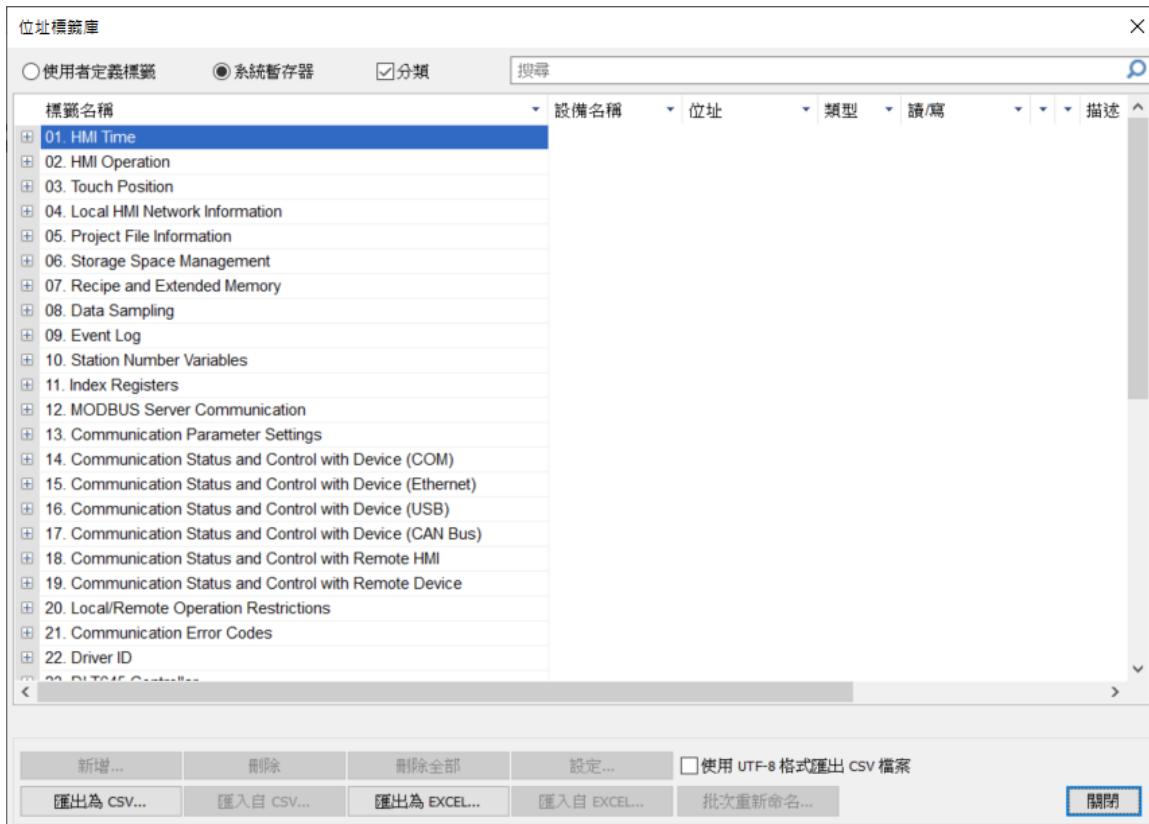
## 22.1. 概要

EasyBuilder Pro 編輯軟體保留了一些位元位址和字組位址的暫存器供給系統使用，這些系統保留暫存器分別有不同的功用，而我們將系統保留暫存器位址分類如下。

因某些系統暫存器即使可透過 [巨集] 或 [遠端 HMI] 寫入數值，亦不表示可以控制該暫存器的功能。若該系統暫存器可透過 [巨集] 或 [遠端 HMI] 寫入及控制，則標示為 "控制"。

當選擇 cMT/cMT X series，系統將提供 PLW 與 PLB 暫存器供挑選。LW / LB 與 PLW / PLB 不同處在於 LW / LB 是指 HMI 本機上的位址，而 PLW / PLB 則是指 Client 端 (例如: cMT-iV5、cMT-iV6、iOS、Android 設備) 上操作這些功能的位址。因每台 cMT/cMT X series 可供多台 Client 端連接，因此以上這些功能的系統暫存器將由 Client 端個別運行。

**注意:** 各機型所支援的系統暫存器，將以挑選該機型後於 EasyBuilder Pro 上的位址標籤庫是否有顯示支援為準，不支援的系統暫存器將反灰。



## 22.2. 本機 HMI 記憶體位址範圍

### 22.2.1. 位元位址

暫存器	設備類型	範圍	格式
本機位元位址 (使用者使用)	LB	0 ~ 8999	DDDDD
本機位元位址 (系統暫存器使用)	LB	9000 ~ 12895	DDDDD
Client 端位元位址	PLB	0 ~ 12399	DDDDD
本機字元位址取 位元位址	LW_Bit	0 ~ 1290015	DDDDDDdd DDDD: 位址 dd: 位元位址 (00 ~ 15)
Client 端字元位址 取位元位址	PLW_Bit	0 ~ 1230015	DDDDDDdd DDDD: 位址 dd: 位元位址 (00 ~ 15)
配方暫存器的位 元位址索引偏移 量	RBI	0 ~ 65535f	DDDDDh DDDD: 位址 h: 位元位址 (0 ~ f) 透過 LW-9000 來當作 索引暫存器，並對應 到 RW_Bit
配方暫存器 RW 的位元位址	RW_Bit	0 ~ 524287f	DDDDDh DDDD: 位址 h: 位元位址 (0 ~ f)
配方暫存器 RW_A 的位元位 址	RW_A_Bit	0 ~ 65535f	DDDDDh DDDD: 位址 h: 位元位址 (0 ~ f)
Client 端配方暫存 器 PRW 取位元位 址	PRW_Bit	0 ~ 52428715	DDDDDDdd DDDD: 位址 dd: 位元位址 (00 ~ 15)

### 22.2.2. 字組位址

暫存器	設備類型	範圍	格式
本機字元位址	LW	0 ~ 8999	DDDDD

(使用者使用)				
本機字元位址	LW	9000 ~ 12900	DDDDD	
(系統暫存器使用)				
Client 端字元位址	PLW	0 ~ 12300	DDDDD	
配方暫存器 RW	RW	0 ~ 524287	DDDDDD	
配方暫存器 RW_A	RW_A	0 ~ 65535	DDDDD	
配方暫存器的字元 位址索引偏移量	RWI	0 ~ 65535	DDDDD	透過 LW-9000 來當作 索引暫存器，並對應到 RW
Client 端配方暫存 器 PRW	PRW	0 ~ 524287	DDDDDD	
擴展記憶體暫存器	EM0 ~ EM15	0 ~ 1073741823	DDDDDDDDDDDD	

## 22.3. 系統暫存器

### 22.3.1. HMI 時間

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-11958	時間設定錯誤 (當狀態為 ON) *註 3	讀	讀	讀	是
LW-9010	(16bit-BCD) : 本機時間 (秒)	讀/寫	讀/控制	讀/控制	是
LW-9011	(16bit-BCD) : 本機時間 (分)	讀/寫	讀/控制	讀/控制	是
LW-9012	(16bit-BCD) : 本機時間 (時)	讀/寫	讀/控制	讀/控制	是
LW-9013	(16bit-BCD) : 本機時間 (日)	讀/寫	讀/控制	讀/控制	是
LW-9014	(16bit-BCD) : 本機時間 (月)	讀/寫	讀/控制	讀/控制	是
LW-9015	(16bit-BCD) : 本機時間 (年)	讀/寫	讀/控制	讀/控制	是
LW-9016	(16bit-BCD) : 本機時間 (星期)	讀	讀	讀	是
LW-9017	(16bit) : 本機時間 (秒)	讀/寫	讀/控制	讀/控制	是
LW-9018	(16bit) : 本機時間 (分)	讀/寫	讀/控制	讀/控制	是
LW-9019	(16bit) : 本機時間 (時)	讀/寫	讀/控制	讀/控制	是
LW-9020	(16bit) : 本機時間 (日)	讀/寫	讀/控制	讀/控制	是
LW-9021	(16bit) : 本機時間 (月)	讀/寫	讀/控制	讀/控制	是
LW-9022	(16bit) : 本機時間 (年) *註 1	讀/寫	讀/控制	讀/控制	是
LW-9023	(16bit) : 本機時間 (星期) *註 2	讀	讀	讀	是
LW-9030	(32bit) : 系統時間 (單位 : 0.1 秒)	讀	讀	讀	是
LW-9048	(16bit) : 時間 (0 : AM, 1 : PM)	讀/寫	讀/控制	讀/控制	是

LW-9049	(16bit)：本機時間 (12 小時制)	讀/寫	讀/控制	讀/控制	是
---------	-----------------------	-----	------	------	---

 Note

1. 數值範圍為 2000 ~ 2035。
2. 數值範圍為 0 ~ 6，即為星期日 ~ 星期六。
3. 當透過 LW-9010 ~ LW-9023 更新 HMI 時間時，若數值設置為非法範圍，系統會將時間回復為設定前的時間，並將 LB-11958 設定為 ON。若使用 LW-9010 ~ LW-9023 於 PC 模擬時修改時間，將無法作用。

### 22.3.2. HMI 操作

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9018	隱藏 (設 ON)/顯示 (設 OFF) 滑鼠游標	讀/寫	讀/控制	讀/控制	是
LB-9019	取消 (設 ON)/開啟 (設 OFF) 聲音輸出	讀/寫	讀/控制	讀/控制	是
LB-9020	顯示 (設 ON)/隱藏 (設 OFF) 系統設定列	讀/寫	讀/控制	讀/控制	是
LB-9033	取消 (設 ON)/開啟 (設 OFF) HMI 上傳功能 * <small>註 1</small>	讀/寫	讀/控制	讀	是
LB-9040	背光燈調亮 (設定為 ON) * <small>註 2</small>	讀/寫	讀/控制	讀/控制	是
LB-9041	背光燈調暗 (設定為 ON) * <small>註 2</small>	寫	控制	控制	是
LB-9047	重新啟動 HMI (設定為 ON, 並當 LB-9048 狀態為 ON 時)	寫	控制	控制	是
LB-9048	重啟機制保護	讀/寫	讀/控制	讀/控制	是
LB-9062	開啟硬體設定對話窗 (設定為 ON)	寫	控制	控制	是
LB-9063	隱藏 (設 ON)/顯示 (設 OFF) 當插上 USB 碟時彈出下載視窗	讀/寫	讀/控制	讀/控制	是
LB-9064	啟用 USB 條碼掃瞄器裝置 (鍵盤功能關閉) (當狀態為 ON) * <small>註 5</small>	讀/寫	讀/控制	讀	是
LB-11959	LED 指示燈控制 * <small>註 4</small>	讀/寫	讀/控制	讀/控制	否
LB-12042	開啟/關閉 [System information] 對話窗 (設定為 ON/設定為 OFF)	讀/寫	讀/控制	讀/控制	否
LB-12051	蜂鳴器控制 (當狀態為 ON 時啟動)	讀/寫	讀/控制	讀/控制	是
LB-12360	CPU 使用率警報 (> 95%) * <small>註 6</small>	讀	讀	讀	是
LB-12364	顯示 (設 ON)/隱藏 (設 OFF) [初始化 HMI] 選項於觸控校正模式	讀/寫	讀/控制	讀/控制	否
LB-12660	重置按鈕的狀態 (當按鈕按下時狀態為 ON)	讀	讀	讀	是
LB-12665	開啟 USB/SD 卡的上傳/下載視窗	讀/寫	讀/控制	讀/控制	是
LB-12752	取消(設 OFF)/啟用(設 ON) 震動. (支援的 OS 版本 : 20200618 或更新的版本) * <small>註 7</small>	讀/寫	讀/控制	讀/控制	是
LW-9007	(16bit)：硬體索引編號	讀	讀	讀	否

LW-9008	(32bit-float) : 電池電壓 *註 3	讀	讀	讀	否
LW-9025	(16bit) : CPU 使用率	讀	讀	讀	是
LW-9026	(16bit) : OS 版本 (年)	讀	讀	讀	是
LW-9027	(16bit) : OS 版本 (月)	讀	讀	讀	是
LW-9028	(16bit) : OS 版本 (日)	讀	讀	讀	是
LW-9040	(16bit) : 背光燈亮度值 *註 2	讀/寫	讀/控制	讀/控制	是
LW-9051	(16bit) : 音量調整 (0~100)	讀/寫	讀/控制	讀/控制	是
LW-9054	(32bit) : HMI 機型 ID	讀	讀	讀	是
LW-9080	(16bit) : 背光節能時間 (單位 : 分鐘)	讀/寫	讀/控制	讀/控制	是
LW-9081	(16bit) : 螢幕保護時間 (單位 : 分鐘)	讀/寫	讀/控制	讀/控制	是
LW-9141	(16bit) : HMI 站號	讀/寫	讀/控制	讀/控制	是
LW-9199	(16bit) : 外部鍵盤編排模式 :0: (QWERTY), 1: (AZERTY), 2: (Cyrillic)	讀/寫	讀/控制	讀/控制	是
LW-9350	(16bit) : 本機尚未處理的命令數目	讀	讀	讀	是
LW-10884	(16 words) : HMI 名稱	讀/寫	讀/控制	讀/控制	是
LW-11155	(32bit) : HMI 上記憶體總數 (K bytes)	讀	讀	讀	是
LW-11157	(32bit) : HMI 上當前剩餘的記憶體總數 (K bytes)	讀	讀	讀	是
LW-11159	(16bit) : 記憶體使用率 (x 100%)	讀	讀	讀	是
LW-11382	(16bit) : 指撥開關狀態 (bit 0 : DIP 1, bit 1 : DIP 2, bit 2 : DIP 3)	讀	讀	讀	是
LW-12834	(16bit) : 系統設定主題 (0: 淺色 , 1: 深色)	讀/寫	讀/控制	讀/控制	是

### Note

- 當變更設定時，需重啟 HMI 以更新設定值。
- LW-9040 可搭配 LB-9040 ~ LB-9041 來調整背光亮度，亮度值為 0 ~ 31。
- 只支援 eMT 系列，建議當 LW-9008 的電池電壓值少於 2.80V 時，可更換電池。
- 可用於辨識 mTV 與 cMT-SVR/cMT-HDM/cMT-FHD 的裝置。當有多台裝置時，可觸發此位址讓 LED 燈的狀態閃爍。
- LB-9064 啟用 USB 條碼掃瞄器裝置 (鍵盤功能關閉) 範例如下：



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

- 當 CPU 負載連續 30 秒高於 95% 時，將被設定為 ON。
- 只支援於 cMT3162X。

#### 22.3.3. 觸碰位置

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9041	(16bit) : 觸控狀態 (bit 0 on = 正在觸碰螢幕)	讀	讀	讀	是

## 位址暫存器

LW-9042	(16bit) : 觸碰時, X 的位置	讀	讀	讀	是
LW-9043	(16bit) : 觸碰時, Y 的位置	讀	讀	讀	是
LW-9044	(16bit) : 離開時, X 的位置	讀	讀	讀	是
LW-9045	(16bit) : 離開時, Y 的位置	讀	讀	讀	是



請點選此圖示下載範例程式。此範例程式說明如何透過觸碰位置相關的位址暫存器來達到隱密性的換頁動作。下載範例程式前，請先確定已連上網路線。

## 22.3.4. 本機 HMI 網路資訊

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12041	刷新 HMI 乙太網路資訊 (DHCP, 閘道, 遮罩) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12094	更新乙太網路 1 設定 (IP, 遮罩, 閘道) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12095	更新乙太網路 2 設定 (IP, 遮罩) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12822	取消 (設為 OFF)/啟用 (設為 ON) 乙太網路橋接模式	讀/寫	讀/控制	讀/控制	是
LW-9125	(16bit) : HMI 乙太網路 1 所使用的閘道 0 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9126	(16bit) : HMI 乙太網路 1 所使用的閘道 1 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9127	(16bit) : HMI 乙太網路 1 所使用的閘道 2 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9128	(16bit) : HMI 乙太網路 1 所使用的閘道 3 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9129	(16bit) : HMI 乙太網路 1 所使用的 IP 0 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9130	(16bit) : HMI 乙太網路 1 所使用的 IP 1 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9131	(16bit) : HMI 乙太網路 1 所使用的 IP 2 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9132	(16bit) : HMI 乙太網路 1 所使用的 IP 3 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-9133	(16bit) : 乙太網路所使用的連接埠 (只在 HMI 上有效)	讀	讀	讀	是
LW-9135	(16bit) : HMI 實體位址 (MAC) 0	讀	讀	讀	是

## 位址暫存器

LW-9136	(16bit) : HMI 實體位址 (MAC) 1	讀	讀	讀	是
LW-9137	(16bit) : HMI 實體位址 (MAC) 2	讀	讀	讀	是
LW-9138	(16bit) : HMI 實體位址 (MAC) 3	讀	讀	讀	是
LW-9139	(16bit) : HMI 實體位址 (MAC) 4	讀	讀	讀	是
LW-9140	(16bit) : HMI 實體位址 (MAC) 5	讀	讀	讀	是
LW-10750	(16bit) : HMI 乙太網路 1 所使用的遮罩 0(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10751	(16bit) : HMI 乙太網路 1 所使用的遮罩 1(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10752	(16bit) : HMI 乙太網路 1 所使用的遮罩 2(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10753	(16bit) : HMI 乙太網路 1 所使用的遮罩 3(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10786	(16bit) : HMI 乙太網路 2 所使用的 IP 0(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10787	(16bit) : HMI 乙太網路 2 所使用的 IP 1(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10788	(16bit) : HMI 乙太網路 2 所使用的 IP 2(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10789	(16bit) : HMI 乙太網路 2 所使用的 IP 3(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10790	(16bit) : HMI 乙太網路 2 所使用的遮罩 0(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10791	(16bit) : HMI 乙太網路 2 所使用的遮罩 1(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10792	(16bit) : HMI 乙太網路 2 所使用的遮罩 2(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10793	(16bit) : HMI 乙太網路 2 所使用的遮罩 3(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10794	(16bit) : HMI 乙太網路 2 所使用的閘道 0(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10795	(16bit) : HMI 乙太網路 2 所使用的閘道 1(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10796	(16bit) : HMI 乙太網路 2 所使用的閘道 2(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-10797	(16bit) : HMI 乙太網路 2 所使用的閘道 3(只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是

LW-10798	(16bit) : 乙太網路 2 實體位址 (MAC) 0	讀	讀	讀	是
LW-10799	(16bit) : 乙太網路 2 實體位址 (MAC) 1	讀	讀	讀	是
LW-10800	(16bit) : 乙太網路 2 實體位址 (MAC) 2	讀	讀	讀	是
LW-10801	(16bit) : 乙太網路 2 實體位址 (MAC) 3	讀	讀	讀	是
LW-10802	(16bit) : 乙太網路 2 實體位址 (MAC) 4	讀	讀	讀	是
LW-10803	(16bit) : 乙太網路 2 實體位址 (MAC) 5	讀	讀	讀	是
LW-10804	(16bit) : HMI 乙太網路 1 網域名稱系統 (DNS) 伺服器 IP0	讀/寫	讀/控制	讀/控制	是
LW-10805	(16bit) : HMI 乙太網路 1 網域名稱系統 (DNS) 伺服器 IP1	讀/寫	讀/控制	讀/控制	是
LW-10806	(16bit) : HMI 乙太網路 1 網域名稱系統 (DNS) 伺服器 IP2	讀/寫	讀/控制	讀/控制	是
LW-10807	(16bit) : HMI 乙太網路 1 網域名稱系統 (DNS) 伺服器 IP3	讀/寫	讀/控制	讀/控制	是
LW-10808	(16bit) : HMI 乙太網路 2 網域名稱系統 (DNS) 伺服器 IP0	讀/寫	讀/控制	讀/控制	是
LW-10809	(16bit) : HMI 乙太網路 2 網域名稱系統 (DNS) 伺服器 IP1	讀/寫	讀/控制	讀/控制	是
LW-10810	(16bit) : HMI 乙太網路 2 網域名稱系統 (DNS) 伺服器 IP2	讀/寫	讀/控制	讀/控制	是
LW-10811	(16bit) : HMI 乙太網路 2 網域名稱系統 (DNS) 伺服器 IP3	讀/寫	讀/控制	讀/控制	是
LW-10812	(16bit) : 自動分配 IP 位址 (DHCP => 0 : off, 1 : on)	讀/寫	讀/控制	讀/控制	是
LW-10813	(16bit) : 自動分配乙太網路 2 的 IP 位址 (DHCP => 0 : off, 1 : on)	讀/寫	讀/控制	讀/控制	是
LW-10815	(16bit) : 乙太網路 1 的連線速度 (0:失敗, 10 (10M), 100 (100M), 1000 (1G))	讀	讀	讀	是
LW-10816	(16bit) : 乙太網路 2 的連線速度 (0:失敗, 10 (10M), 100 (100M), 1000 (1G))	讀	讀	讀	是

### 22.3.5. 工程檔案資訊

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9100	(16bit) : 工程檔案的名稱 (14 字組)	讀	讀	讀	是
LW-9116	(32bit) : 工程檔案的大小 (單位 : byte)	讀	讀	讀	是
LW-9118	(32bit) : 工程檔案的大小 (單位 : K bytes)	讀	讀	讀	是

LW-9120	(32bit) : EasyBuilder Pro 版本	讀	讀	讀	是
LW-9122	(16bit) : 工程檔案編譯日期 [年]	讀	讀	讀	是
LW-9123	(16bit) : 工程檔案編譯日期 [月]	讀	讀	讀	是
LW-9124	(16bit) : 工程檔案編譯日期 [日]	讀	讀	讀	是
LW-11440	(16bit) : 工程檔案編譯時間 [時] (24 小時制)	讀	讀	讀	是
LW-11441	(16bit) : 工程檔案編譯時間 [分]	讀	讀	讀	是
LW-11442	(16bit) : 工程檔案編譯時間 [秒]	讀	讀	讀	是
LW-12752	(16 words) : 工程檔案校驗和	讀	讀	讀	是
LW-12825	(32bit) : EasyBuilder Pro 版本 (完整)	讀	讀	讀	是

### 22.3.6. 儲存空間管理

特定有支援 SD 卡槽或可外接兩個 USB 硬碟的機型才能使用相關的系統暫存器。

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9035	HMI 剩餘空間不足警示 (當狀態為 ON)	讀	讀	讀	是
LB-9036	SD 卡剩餘空間不足警示 (當狀態為 ON)	讀	讀	讀	是
LB-9037	USB 碟 1 剩餘空間不足警示 (當狀態為 ON)	讀	讀	讀	是
LB-9038	USB 碟 2 剩餘空間不足警示 (當狀態為 ON) <i>*註 1</i>	讀	讀	讀	是
LB-12048	USB 碟 1 狀態偵測 (當狀態為 ON 時表示存在)	讀	讀	讀	是
LB-12049	USB 碟 2 狀態偵測 (當狀態為 ON 時表示存在) <i>*註 1</i>	讀	讀	讀	是
LB-12050	SD 卡狀態 (當狀態為 ON 時表示存在)	讀	讀	讀	是
LW-9070	(16bit) : 剩餘空間警報下限 (Mega bytes)	讀	讀	讀	否
LW-9071	(16bit) : 系統保留的剩餘空間 (Mega bytes)	讀	讀	讀	否
LW-9072	(32bit) : HMI 當前的剩餘空間 (K bytes)	讀	讀	讀	是
LW-9074	(32bit) : SD 卡當前的剩餘空間 (K bytes)	讀	讀	讀	是
LW-9076	(32bit) : USB 碟 1 當前的剩餘空間 (K bytes)	讀	讀	讀	是
LW-9078	(32bit) : USB 碟 2 當前的剩餘空間 (K bytes) * <i>註 1</i>	讀	讀	讀	否
LW-11458	(32bit) : HMI 歷史資料總儲存空間 (K bytes) * <i>註 2</i>	讀	讀	讀	是
LW-11460	(32bit) : HMI 當前剩餘的歷史資料儲存空間 (K bytes) * <i>註 2</i>	讀	讀	讀	是
LW-12492	(4 words) : 最後偵測 USB 分割 ID * <i>註 3</i>	讀	讀	讀	是
LW-12496	(32-bit float) : 最後偵測 USB 分割大小 (GB) *	讀	讀	讀	是

	<b>註 3</b>				
LW-12498	(4 words) : 設定作為 USB1 的分割 ID * <b>註 3</b>	讀/寫	讀/控制	讀/控制	是
LW-12502	(4 words) : 設定作為 USB2 的分割 ID * <b>註 3</b>	讀/寫	讀/控制	讀/控制	是
LW-12829	(32bit) : HMI 可用的歷史資料儲存空間 (K bytes)	讀	讀	讀	是

 **Note**

1. USB 碟 2 的相關暫存器只支援於 iP / iE, cMT X 系列。
2. 只支援於 cMT / cMT X 系列。
3. 將特定 USB 碟的 ID 手動寫入至 LW-12498/LW-12502 後，該 USB 碟未來將會識別為 USB1/USB2，也僅該 USB 碟會識別為 USB1/USB2。如要解除 USB 識別限制，需將 LW-12498/LW-12502 重新設定為空值。若 USB1 ID(LW-12498)與 USB2 ID(LW-12502)寫入相同的 USB ID，該 USB ID 將預設識別為 USB1。



請點選此圖示下載範例程式。此範例程式說明如何使用 LW-9072 ~ LW-9076 來搭配備份物件的應用。下載範例程式前，請先確定已連上網路線。

### 22.3.7. 配方及擴展記憶體

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9028	重置配方資料 (設定為 ON)	寫	控制	控制	是
LB-9029	儲存配方資料到 HMI (設定為 ON)	寫	控制	控制	是
LB-9460	EM0 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9461	EM1 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9462	EM2 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9463	EM3 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9464	EM4 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9465	EM5 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9466	EM6 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9467	EM7 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9468	EM8 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是

	ON)				
LB-9469	EM9 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9470	EM0 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9471	EM1 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9472	EM2 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9473	EM3 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9474	EM4 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9475	EM5 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9476	EM6 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9477	EM7 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9478	EM8 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9479	EM9 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9480	EM0 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9481	EM1 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9482	EM2 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9483	EM3 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9484	EM4 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9485	EM5 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9486	EM6 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9487	EM7 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是

	為 ON)				
LB-9488	EM8 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-9489	EM9 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12363	禁止從遠端 HMI 更新配方資料庫 (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12801	EM10 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12802	EM11 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12803	EM12 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12804	EM13 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12805	EM14 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12806	EM15 的儲存裝置 (SD 卡) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12807	EM10 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12808	EM11 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12809	EM12 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12810	EM13 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12811	EM14 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12812	EM15 的儲存裝置 (USB 碟) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12813	EM10 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12814	EM11 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12815	EM12 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12816	EM13 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是

	態為 ON)				
LB-12817	EM14 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是
LB-12818	EM15 的儲存裝置 (USB 碟 2) 不存在 (當狀態為 ON)	讀	讀	讀	是

### 22.3.8. 資料取樣

特定有支援 SD 卡槽或可外接兩個 USB 硬碟的機型才能使用相關的系統暫存器。

cMT 系列不支援透過系統暫存器來刪除檔案或查看更新關於資料取樣的檔案資訊。但可透過資料取樣設定下的控制位址來刪除檔案，詳情可參考第八章。

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9025	刪除 HMI 記憶體裡日期最早的資料取樣檔案 (設定為 ON)	寫	控制	控制	否
LB-9026	刪除 HMI 記憶體裡全部資料取樣檔案 (設定為 ON)	寫	控制	控制	是
LB-9027	更新 HMI 記憶體裡資料取樣統計資訊 (設定為 ON)	寫	控制	控制	否
LB-9034	儲存事件記錄/資料取樣記錄/操作記錄至 HMI, USB 碟, SD 卡 (設定為 ON) *註 1	寫	控制	控制	是
LB-11949	刪除 SD 卡裡日期最早的資料取樣檔案 (設定為 ON)	寫	控制	控制	否
LB-11950	刪除 SD 卡裡全部資料取樣檔案 (設定為 ON)	寫	控制	控制	是
LB-11951	更新 SD 卡裡資料取樣統計資訊 (設定為 ON)	寫	控制	控制	否
LB-11952	刪除 USB 碟 1 裡日期最早的資料取樣檔案 (設定為 ON)	寫	控制	控制	否
LB-11953	刪除 USB 碟 1 裡全部資料取樣檔案 (設定為 ON)	寫	控制	控制	是
LB-11954	更新 USB 碟 1 裡資料取樣統計資訊 (設定為 ON)	寫	控制	控制	否
LB-11955	刪除 USB 碟 2 裡日期最早的資料取樣檔案 (設定為 ON) *註 3	寫	控制	控制	否
LB-11956	刪除 USB 碟 2 裡全部資料取樣檔案 (設定為 ON) *註 3	寫	控制	控制	是
LB-11957	更新 USB 碟 2 裡資料取樣統計資訊 (設定為 ON) *註 3	寫	控制	控制	否
LW-9063	(16bit) : HMI 記憶體裡存在的資料取樣檔案數	讀	讀	讀	否

	目				
LW-9064	(32bit) : HMI 記憶體裡存在的資料取樣檔案大小 (bytes)	讀	讀	讀	否
LW-10489	(16bit) : SD 卡裡存在的資料取樣檔案數目	讀	讀	讀	否
LW-10490	(32bit) : SD 卡裡存在的資料取樣檔案大小 (bytes)	讀	讀	讀	否
LW-10492	(16bit) : USB 碟 1 裡存在的資料取樣檔案數目	讀	讀	讀	否
LW-10493	(32bit) : USB 碟 1 裡存在的資料取樣檔案大小 (bytes)	讀	讀	讀	否
LW-10495	(16bit) : USB 碟 2 裡存在的資料取樣檔案數目 *註 3	讀	讀	讀	否
LW-10496	(32bit) : USB 碟 2 裡存在的資料取樣檔案大小 (bytes) *註 3	讀	讀	讀	否

### Note

1. 最快執行時間間隔為每 2 秒。
2. 刪除或更新資料取樣的相關暫存器，於 PC 模擬時皆無作用。
3. USB 碟 2 的相關暫存器只支援於 iP / iE 系列。

#### 22.3.9. 事件記錄

特定有支援 SD 卡槽或可外接兩個 USB 硬碟的機型才能使用相關的系統暫存器。

cMT 系列不支援透過系統暫存器來刪除檔案或查看更新關於事件記錄的檔案資訊。但可透過事件登錄設定下的控制位址來刪除檔案，詳情可參考第七章。

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9021	重置當前的事件記錄 (OFF->ON)	寫	控制	控制	否
LB-9022	刪除 HMI 記憶體裡日期最早的事件記錄檔案 (設定為 ON)	寫	控制	控制	否
LB-9023	刪除 HMI 記憶體裡全部事件記錄檔案 (設定為 ON)	寫	控制	控制	是
LB-9024	更新 HMI 記憶體裡事件記錄統計資訊 (設定為 ON)	寫	控制	控制	否
LB-9034	儲存事件記錄/資料取樣記錄/操作記錄至 HMI, USB 碟, SD 卡 (設定為 ON) *註 2	寫	控制	控制	是
LB-9042	確認全部事件 (設定為 ON)	寫	控制	控制	是
LB-9043	存在未確認的事件 (當狀態為 ON)	讀	讀	讀	是

LB-11940	刪除 SD 卡裡日期最早的事件記錄檔案 (設定為 ON)	寫	控制	控制	否
LB-11941	刪除 SD 卡裡全部事件記錄檔案 (設定為 ON)	寫	控制	控制	是
LB-11942	更新 SD 卡裡事件記錄統計資訊 (設定為 ON)	寫	控制	控制	否
LB-11943	刪除 USB 碟 1 裡日期最早的事件記錄檔案 (設定為 ON)	寫	控制	控制	否
LB-11944	刪除 USB 碟 1 裡全部事件記錄檔案 (設定為 ON)	寫	控制	控制	是
LB-11945	更新 USB 碟 1 裡事件記錄統計資訊 (設定為 ON)	寫	控制	控制	否
LB-11946	刪除 USB 碟 2 裡日期最早的事件記錄檔案 (設定為 ON) *註 4	寫	控制	控制	否
LB-11947	刪除 USB 碟 2 裡全部事件記錄檔案 (設定為 ON) *註 4	寫	控制	控制	是
LB-11948	更新 USB 碟 2 裡事件記錄統計資訊 (設定為 ON) *註 4	寫	控制	控制	否
LB-12024	取消報警聲 (設定為 ON)	寫	控制	控制	是
LB-12399	當有報警存在於任一類別時設狀態為 ON *註 5	讀	讀	讀	是
LB-12400	當類別 0 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12401	當類別 1 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12402	當類別 2 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12403	當類別 3 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12404	當類別 4 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12405	當類別 5 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12406	當類別 6 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12407	當類別 7 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LB-12655	當類別 255 有報警存在時設狀態為 ON *註 5	讀	讀	讀	是
LW-9060	(16bit) : HMI 記憶體裡存在的事件記錄檔案數目	讀	讀	讀	否
LW-9061	(32bit) : HMI 記憶體裡存在的事件記錄檔案大小 (bytes)	讀	讀	讀	否
LW-9450	(16bit) : 事件記錄的時間標籤 - 秒 *註 1	讀/寫	讀/控制	讀/控制	是
LW-9451	(16bit) : 事件記錄的時間標籤 - 分 *註 1	讀/寫	讀/控制	讀/控制	是
LW-9452	(16bit) : 事件記錄的時間標籤 - 時 *註 1	讀/寫	讀/控制	讀/控制	是
LW-9453	(16bit) : 事件記錄的時間標籤 - 曰 *註 1	讀/寫	讀/控制	讀/控制	是
LW-9454	(16bit) : 事件記錄的時間標籤 - 月 *註 1	讀/寫	讀/控制	讀/控制	是

LW-9455	(16bit) : 事件記錄的時間標籤 - 年 *註 1	讀/寫	讀/控制	讀/控制	是
LW-10480	(16bit) : SD 卡裡存在的事件記錄檔案數目	讀	讀	讀	否
LW-10481	(32bit) : SD 卡裡存在的事件記錄檔案大小 (bytes)	讀	讀	讀	否
LW-10483	(16bit) : USB 碟 1 裡存在的事件記錄檔案數目	讀	讀	讀	否
LW-10484	(32bit) : USB 碟 1 裡存在的事件記錄檔案大小 (bytes)	讀	讀	讀	否
LW-10486	(16bit) : USB 碟 2 裡存在的事件記錄檔案數目 *註 4	讀	讀	讀	否
LW-10487	(32bit) : USB 碟 2 裡存在的事件記錄檔案大小 (bytes) *註 4	讀	讀	讀	否
LW-11443	(16bit) : 推播通知報警狀態 (0: 無; 1: 綠色; 2: 黃色; 3: 紅色)	讀	讀	讀	是
LW-11499	報警總數*註 5	讀	讀	讀	是
LW-11500	類別 0 的報警數量 *註 5	讀	讀	讀	是
LW-11501	類別 1 的報警數量*註 5	讀	讀	讀	是
LW-11502	類別 2 的報警數量 *註 5	讀	讀	讀	是
LW-11503	類別 3 的報警數量 *註 5	讀	讀	讀	是
LW-11504	類別 4 的報警數量*註 5	讀	讀	讀	是
LW-11505	類別 5 的報警數量 *註 5	讀	讀	讀	是
LW-11506	類別 6 的報警數量 *註 5	讀	讀	讀	是
LW-11507	類別 7 的報警數量 *註 5	讀	讀	讀	是
LW-11755	類別 255 的報警數量 *註 5	讀	讀	讀	是
LW-11763	(16bit) : 低等級警報總數 *註 5	讀	讀	讀	是
LW-11764	(16bit) : 正常等級警報總數 *註 5	讀	讀	讀	是
LW-11765	(16bit) : 高等級警報總數 *註 5	讀	讀	讀	是
LW-11766	(16bit) : 緊急等級警報總數 *註 5	讀	讀	讀	是

### Note

1. 若欲使用 LW-9450 ~ LW-9455 作為事件登錄的時間來源標籤，請先在 [系統參數設定] » [一般屬性] 頁籤設定相關屬性。
2.  請點選此圖示下載範例程式。此範例程式說明如何使用 LW-9450 ~ LW-9455 作為事件記錄的時間來源標籤。下載範例程式前，請先確定已連上網路線。
3. 最快執行時間間隔為每 2 秒。
4. 刪除或更新事件記錄的相關暫存器，於 PC 模擬時皆無作用。
5. USB 碟 2 的相關暫存器只支援於 iP / iE 系列。
6. 只支援於 cMT/ cMT-X 系列。

### 22.3.10. 站號變數

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-10000	(16bit) : var0 - 站號變數 (語法 : var0#位址)	讀/寫	讀/控制	讀/控制	是
LW-10001	(16bit) : var1 - 站號變數 (語法 : var1#位址)	讀/寫	讀/控制	讀/控制	是
LW-10002	(16bit) : var2 - 站號變數 (語法 : var2#位址)	讀/寫	讀/控制	讀/控制	是
LW-10003	(16bit) : var3 - 站號變數 (語法 : var3#位址)	讀/寫	讀/控制	讀/控制	是
LW-10004	(16bit) : var4 - 站號變數 (語法 : var4#位址)	讀/寫	讀/控制	讀/控制	是
LW-10005	(16bit) : var5 - 站號變數 (語法 : var5#位址)	讀/寫	讀/控制	讀/控制	是
LW-10006	(16bit) : var6 - 站號變數 (語法 : var6#位址)	讀/寫	讀/控制	讀/控制	是
LW-10007	(16bit) : var7 - 站號變數 (語法 : var7#位址)	讀/寫	讀/控制	讀/控制	是
LW-10008	(16bit) : var8 - 站號變數 (語法 : var8#位址)	讀/寫	讀/控制	讀/控制	是
LW-10009	(16bit) : var9 - 站號變數 (語法 : var9#位址)	讀/寫	讀/控制	讀/控制	是
LW-10010	(16bit) : var10 - 站號變數 (語法 : var10#位址)	讀/寫	讀/控制	讀/控制	是
LW-10011	(16bit) : var11 - 站號變數 (語法 : var11#位址)	讀/寫	讀/控制	讀/控制	是
LW-10012	(16bit) : var12 - 站號變數 (語法 : var12#位址)	讀/寫	讀/控制	讀/控制	是
LW-10013	(16bit) : var13 - 站號變數 (語法 : var13#位址)	讀/寫	讀/控制	讀/控制	是
LW-10014	(16bit) : var14 - 站號變數 (語法 : var14#位址)	讀/寫	讀/控制	讀/控制	是
LW-10015	(16bit) : var15 - 站號變數 (語法 : var15#位址)	讀/寫	讀/控制	讀/控制	是



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.11. 索引暫存器

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9000	(16bit) : RWI 索引暫存器	讀/寫	讀/控制	讀/控制	是
LW-9200	(16bit) : 位址索引暫存器 0	讀/寫	讀/控制	讀/控制	是
LW-9201	(16bit) : 位址索引暫存器 1	讀/寫	讀/控制	讀/控制	是
LW-9202	(16bit) : 位址索引暫存器 2	讀/寫	讀/控制	讀/控制	是
LW-9203	(16bit) : 位址索引暫存器 3	讀/寫	讀/控制	讀/控制	是
LW-9204	(16bit) : 位址索引暫存器 4	讀/寫	讀/控制	讀/控制	是
LW-9205	(16bit) : 位址索引暫存器 5	讀/寫	讀/控制	讀/控制	是
LW-9206	(16bit) : 位址索引暫存器 6	讀/寫	讀/控制	讀/控制	是
LW-9207	(16bit) : 位址索引暫存器 7	讀/寫	讀/控制	讀/控制	是
LW-9208	(16bit) : 位址索引暫存器 8	讀/寫	讀/控制	讀/控制	是
LW-9209	(16bit) : 位址索引暫存器 9	讀/寫	讀/控制	讀/控制	是
LW-9210	(16bit) : 位址索引暫存器 10	讀/寫	讀/控制	讀/控制	是
LW-9211	(16bit) : 位址索引暫存器 11	讀/寫	讀/控制	讀/控制	是
LW-9212	(16bit) : 位址索引暫存器 12	讀/寫	讀/控制	讀/控制	是
LW-9213	(16bit) : 位址索引暫存器 13	讀/寫	讀/控制	讀/控制	是
LW-9214	(16bit) : 位址索引暫存器 14	讀/寫	讀/控制	讀/控制	是
LW-9215	(16bit) : 位址索引暫存器 15	讀/寫	讀/控制	讀/控制	是
LW-9230	(32bit) : 位址索引暫存器 16	讀/寫	讀/控制	讀/控制	是
LW-9232	(32bit) : 位址索引暫存器 17	讀/寫	讀/控制	讀/控制	是
LW-9234	(32bit) : 位址索引暫存器 18	讀/寫	讀/控制	讀/控制	是
LW-9236	(32bit) : 位址索引暫存器 19	讀/寫	讀/控制	讀/控制	是
LW-9238	(32bit) : 位址索引暫存器 20	讀/寫	讀/控制	讀/控制	是
LW-9240	(32bit) : 位址索引暫存器 21	讀/寫	讀/控制	讀/控制	是
LW-9242	(32bit) : 位址索引暫存器 22	讀/寫	讀/控制	讀/控制	是
LW-9244	(32bit) : 位址索引暫存器 23	讀/寫	讀/控制	讀/控制	是
LW-9246	(32bit) : 位址索引暫存器 24	讀/寫	讀/控制	讀/控制	是
LW-9248	(32bit) : 位址索引暫存器 25	讀/寫	讀/控制	讀/控制	是
LW-9250	(32bit) : 位址索引暫存器 26	讀/寫	讀/控制	讀/控制	是
LW-9252	(32bit) : 位址索引暫存器 27	讀/寫	讀/控制	讀/控制	是
LW-9254	(32bit) : 位址索引暫存器 28	讀/寫	讀/控制	讀/控制	是
LW-9256	(32bit) : 位址索引暫存器 29	讀/寫	讀/控制	讀/控制	是
LW-9258	(32bit) : 位址索引暫存器 30	讀/寫	讀/控制	讀/控制	是
LW-9260	(32bit) : 位址索引暫存器 31	讀/寫	讀/控制	讀/控制	是

PLW-9200	(16bit)：私有位址索引暫存器 0	讀/寫	N/A	N/A	是
PLW-9201	(16bit)：私有位址索引暫存器 1	讀/寫	N/A	N/A	是
PLW-9202	(16bit)：私有位址索引暫存器 2	讀/寫	N/A	N/A	是
PLW-9203	(16bit)：私有位址索引暫存器 3	讀/寫	N/A	N/A	是
PLW-9204	(16bit)：私有位址索引暫存器 4	讀/寫	N/A	N/A	是
PLW-9205	(16bit)：私有位址索引暫存器 5	讀/寫	N/A	N/A	是
PLW-9206	(16bit)：私有位址索引暫存器 6	讀/寫	N/A	N/A	是
PLW-9207	(16bit)：私有位址索引暫存器 7	讀/寫	N/A	N/A	是
PLW-9208	(16bit)：私有位址索引暫存器 8	讀/寫	N/A	N/A	是
PLW-9209	(16bit)：私有位址索引暫存器 9	讀/寫	N/A	N/A	是
PLW-9210	(16bit)：私有位址索引暫存器 10	讀/寫	N/A	N/A	是
PLW-9211	(16bit)：私有位址索引暫存器 11	讀/寫	N/A	N/A	是
PLW-9212	(16bit)：私有位址索引暫存器 12	讀/寫	N/A	N/A	是
PLW-9213	(16bit)：私有位址索引暫存器 13	讀/寫	N/A	N/A	是
PLW-9214	(16bit)：私有位址索引暫存器 14	讀/寫	N/A	N/A	是
PLW-9215	(16bit)：私有位址索引暫存器 15	讀/寫	N/A	N/A	是
PLW-9230	(32bit)：私有位址索引暫存器 16	讀/寫	N/A	N/A	是
PLW-9232	(32bit)：私有位址索引暫存器 17	讀/寫	N/A	N/A	是
PLW-9234	(32bit)：私有位址索引暫存器 18	讀/寫	N/A	N/A	是
PLW-9236	(32bit)：私有位址索引暫存器 19	讀/寫	N/A	N/A	是
PLW-9238	(32bit)：私有位址索引暫存器 20	讀/寫	N/A	N/A	是
PLW-9240	(32bit)：私有位址索引暫存器 21	讀/寫	N/A	N/A	是
PLW-9242	(32bit)：私有位址索引暫存器 22	讀/寫	N/A	N/A	是
PLW-9244	(32bit)：私有位址索引暫存器 23	讀/寫	N/A	N/A	是
PLW-9246	(32bit)：私有位址索引暫存器 24	讀/寫	N/A	N/A	是
PLW-9248	(32bit)：私有位址索引暫存器 25	讀/寫	N/A	N/A	是
PLW-9250	(32bit)：私有位址索引暫存器 26	讀/寫	N/A	N/A	是
PLW-9252	(32bit)：私有位址索引暫存器 27	讀/寫	N/A	N/A	是
PLW-9254	(32bit)：私有位址索引暫存器 28	讀/寫	N/A	N/A	是
PLW-9256	(32bit)：私有位址索引暫存器 29	讀/寫	N/A	N/A	是
PLW-9258	(32bit)：私有位址索引暫存器 30	讀/寫	N/A	N/A	是
PLW-9260	(32bit)：私有位址索引暫存器 31	讀/寫	N/A	N/A	是



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.12. MODBUS Server 通訊

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9055	MODBUS server (COM 1) 接收到合法的命令 (當狀態為 ON)	讀	讀	讀	是
LB-9056	MODBUS server (COM 2) 接收到合法的命令 (當狀態為 ON)	讀	讀	讀	是
LB-9057	MODBUS server (COM 3) 接收到合法的命令 (當狀態為 ON)	讀	讀	讀	是
LB-9058	MODBUS server (乙太網路) 接收到合法的命令 (當狀態為 ON)	讀	讀	讀	是
LB-12052	MODBUS server 狀態 (當狀態為 ON 時關閉 server 功能)	讀/寫	讀/控制	讀/控制	是
LW-9270	(16bit)：請求的功能碼 - MODBUS server (COM 1)	讀	讀	讀	是
LW-9271	(16bit)：請求的開始位址 - MODBUS server (COM 1)	讀	讀	讀	是
LW-9272	(16bit)：請求的位址數目 - MODBUS server (COM 1)	讀	讀	讀	是
LW-9275	(16bit)：請求的功能碼 - MODBUS server (COM 2)	讀	讀	讀	是
LW-9276	(16bit)：請求的開始位址 - MODBUS server (COM 2)	讀	讀	讀	是
LW-9277	(16bit)：請求的位址數目 - MODBUS server (COM 2)	讀	讀	讀	是
LW-9280	(16bit)：請求的功能碼 - MODBUS server (COM 3)	讀	讀	讀	是
LW-9281	(16bit)：請求的開始位址 - MODBUS server (COM 3)	讀	讀	讀	是
LW-9282	(16bit)：請求的位址數目 - MODBUS server (COM 3)	讀	讀	讀	是
LW-9285	(16bit)：請求的功能碼 - MODBUS server (乙太網路)	讀	讀	讀	是
LW-9286	(16bit)：請求的開始位址 - MODBUS server (乙太網路)	讀	讀	讀	是
LW-9287	(16bit)：請求的位址數目 - MODBUS server (乙太網路)	讀	讀	讀	是
LW-9288	(16bit)：最後通訊錯誤碼 - MODBUS server (乙太網路)	讀	讀	讀	是
LW-9289	(16bit)：最後通訊錯誤碼 - MODBUS server (COM)	讀	讀	讀	是

	1)				
LW-9290	(16bit) : 最後通訊錯誤碼 - MODBUS server (COM 2)	讀	讀	讀	是
LW-9291	(16bit) : 最後通訊錯誤碼 - MODBUS server (COM 3)	讀	讀	讀	是
LW-9541	(16bit) : MODBUS/ASCII server 站號 (COM 1)	讀/寫	讀/控制	讀/控制	是
LW-9542	(16bit) : MODBUS/ASCII server 站號 (COM 2)	讀/寫	讀/控制	讀/控制	是
LW-9543	(16bit) : MODBUS/ASCII server 站號 (COM 3)	讀/寫	讀/控制	讀/控制	是
LW-9544	(16bit) : MODBUS/ASCII server 站號 (乙太網路)	讀/寫	讀/控制	讀/控制	是
LW-9570	(32bit) : 已接收的數據 (bytes) (COM 1 MODBUS server)	讀	讀	讀	是
LW-9572	(32bit) : 已接收的數據 (bytes) (COM 2 MODBUS server)	讀	讀	讀	是
LW-9574	(32bit) : 已接收的數據 (bytes) (COM 3 MODBUS server)	讀	讀	讀	是
LW-9576	(32bit) : 已接收的數據 (bytes) (乙太網路 MODBUS server)	讀	讀	讀	是

### 22.3.13. 通訊參數設定

位址	描述	讀 / 寫 / 控制			CMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9030	更新 COM 1 通訊參數 (LW-9550~9554) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9031	更新 COM 2 通訊參數 (LW-9555~9559) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9032	更新 COM 3 通訊參數 (LW-9560~9564) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9065	停用/啟用 COM 1 廣播站號	讀/寫	讀/控制	讀/控制	是
LB-9066	停用/啟用 COM 2 廣播站號	讀/寫	讀/控制	讀/控制	是
LB-9067	停用/啟用 COM 3 廣播站號	讀/寫	讀/控制	讀/控制	是
LW-9550	(16bit) : COM 1 模式 (0:RS232,1:RS485 2W,2:RS485 4W) (使用 LB-9030 來更新所有通訊設定)	讀/寫	讀/控制	讀/控制	是
LW-9551	(16bit) : COM 1 串列傳輸速率 (7:1200,8:2400,0:4800,1:9600,10:14400,2:19200, 11:28800,3:38400,4:57600,..) *註 1	讀/寫	讀/控制	讀/控制	是
LW-9552	(16bit) : COM 1 數據位元 (7 : 7 bits, 8 : 8 bits)	讀/寫	讀/控制	讀/控制	是
LW-9553	(16bit) : COM 1 校驗 (0:none, 1:even, 2:odd,	讀/寫	讀/控制	讀/控制	是

	3:mark, 4:space)				
LW-9554	(16bit) : COM 1 停止位元 (1 : 1 bit, 2 : 2 bits)	讀/寫	讀/控制	讀/控制	是
LW-9555	(16bit) : COM 2 模式 (0:RS232,1:RS485 2W,2:RS485 4W) (使用 LB-9031 來更新所有通訊設定)	讀/寫	讀/控制	讀/控制	是
LW-9556	(16bit) : COM 2 串列傳輸速率 (7:1200,8:2400,0:4800,1:9600,10:14400,2:19200, 11:28800,3:38400,4:57600,..) *註 1	讀/寫	讀/控制	讀/控制	是
LW-9557	(16bit) : COM 2 數據位元 (7 : 7 bits, 8 : 8 bits)	讀/寫	讀/控制	讀/控制	是
LW-9558	(16bit) : COM 2 校驗 (0:none, 1:even, 2:odd, 3:mark, 4:space)	讀/寫	讀/控制	讀/控制	是
LW-9559	(16bit) : COM 2 停止位元 (1 : 1 bit, 2 : 2 bits)	讀/寫	讀/控制	讀/控制	是
LW-9560	(16bit) : COM 3 模式 (0:RS232, 1:RS485 2W) (使用 LB-9032 來更新所有通訊設定)	讀/寫	讀/控制	讀/控制	是
LW-9561	(16bit) : COM 3 串列傳輸速率 (7:1200,8:2400,0:4800,1:9600,10:14400,2:19200, 11:28800,3:38400,4:57600,..) *註 1	讀/寫	讀/控制	讀/控制	是
LW-9562	(16bit) : COM 3 數據位元 (7 : 7 bits, 8 : 8 bits)	讀/寫	讀/控制	讀/控制	是
LW-9563	(16bit) : COM 3 校驗 (0:none, 1:even, 2:odd, 3:mark, 4:space)	讀/寫	讀/控制	讀/控制	是
LW-9564	(16bit) : COM 3 停止位元 (1 : 1 bit, 2 : 2 bits)	讀/寫	讀/控制	讀/控制	是
LW-9565	(16bit) : COM 1 廣播站號	讀/寫	讀/控制	讀/控制	是
LW-9566	(16bit) : COM 2 廣播站號	讀/寫	讀/控制	讀/控制	是
LW-9567	(16bit) : COM 3 廣播站號	讀/寫	讀/控制	讀/控制	是
LW-10500	(16bit) : 設備 1 超時 (單位 : 100ms, 0 : 50ms)	讀/寫	讀/控制	讀/控制	是
LW-10501	(16bit) : 設備 1 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10502	(16bit) : 設備 1 ACK 訊號延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10503	(16bit) : 設備 1 參數 1	讀/寫	讀/控制	讀/控制	是
LW-10504	(16bit) : 設備 1 參數 2	讀/寫	讀/控制	讀/控制	是
LW-10505	(16bit) : 設備 2 超時 (單位 : 100ms, 0 : 50ms)	讀/寫	讀/控制	讀/控制	是
LW-10506	(16bit) : 設備 2 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10507	(16bit) : 設備 2 ACK 訊號延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10508	(16bit) : 設備 2 參數 1	讀/寫	讀/控制	讀/控制	是
LW-10509	(16bit) : 設備 2 參數 2	讀/寫	讀/控制	讀/控制	是
LW-10510	(16bit) : 設備 3 超時 (單位 : 100ms, 0 : 50ms)	讀/寫	讀/控制	讀/控制	是
LW-10511	(16bit) : 設備 3 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10512	(16bit) : 設備 3 ACK 訊號延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10513	(16bit) : 設備 3 參數 1	讀/寫	讀/控制	讀/控制	是

LW-10514	(16bit) : 設備 3 參數 2	讀/寫	讀/控制	讀/控制	是
LW-10515	(16bit) : 設備 4 超時 (單位 : 100ms)	讀/寫	讀/控制	讀/控制	是
LW-10516	(16bit) : 設備 4 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10517	(16bit) : 設備 4 ACK 訊號延時 (單位 : ms) (SIEMENS S7/400 連接類型)	讀/寫	讀/控制	讀/控制	是
LW-10518	(16bit) : 設備 4 參數 1 (SIEMENS S7/400 機座)	讀/寫	讀/控制	讀/控制	是
LW-10519	(16bit) : 設備 4 參數 2 (SIEMENS S7/400 CPU 插槽)	讀/寫	讀/控制	讀/控制	是
LW-10520	(16bit) : 設備 5 超時 (單位 : 100ms)	讀/寫	讀/控制	讀/控制	是
LW-10521	(16bit) : 設備 5 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10522	(16bit) : 設備 5 ACK 訊號延時 (單位 : ms) (SIEMENS S7/400 連接類型)	讀/寫	讀/控制	讀/控制	是
LW-10523	(16bit) : 設備 5 參數 1 (SIEMENS S7/400 機座)	讀/寫	讀/控制	讀/控制	是
LW-10524	(16bit) : 設備 5 參數 2 (SIEMENS S7/400 CPU 插槽)	讀/寫	讀/控制	讀/控制	是
LW-10525	(16bit) : 設備 6 超時 (單位 : 100ms)	讀/寫	讀/控制	讀/控制	是
LW-10526	(16bit) : 設備 6 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10527	(16bit) : 設備 6 ACK 訊號延時 (單位 : ms) (SIEMENS S7/400 連接類型)	讀/寫	讀/控制	讀/控制	是
LW-10528	(16bit) : 設備 6 參數 1 (SIEMENS S7/400 機座)	讀/寫	讀/控制	讀/控制	是
LW-10529	(16bit) : 設備 6 參數 2 (SIEMENS S7/400 CPU 插槽)	讀/寫	讀/控制	讀/控制	是
LW-10530	(16bit) : 設備 7 超時 (單位 : 100ms)	讀/寫	讀/控制	讀/控制	是
LW-10531	(16bit) : 設備 7 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10532	(16bit) : 設備 7 ACK 訊號延時 (單位 : ms) (SIEMENS S7/400 連接類型)	讀/寫	讀/控制	讀/控制	是
LW-10533	(16bit) : 設備 7 參數 1 (SIEMENS S7/400 機座)	讀/寫	讀/控制	讀/控制	是
LW-10534	(16bit) : 設備 7 參數 2 (SIEMENS S7/400 CPU 插槽)	讀/寫	讀/控制	讀/控制	是
LW-10535	(16bit) : 設備 8 超時 (單位 : 100ms)	讀/寫	讀/控制	讀/控制	是
LW-10536	(16bit) : 設備 8 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10537	(16bit) : 設備 8 ACK 訊號延時 (單位 : ms) (SIEMENS S7/400 連接類型)	讀/寫	讀/控制	讀/控制	是
LW-10538	(16bit) : 設備 8 參數 1 (SIEMENS S7/400 機座)	讀/寫	讀/控制	讀/控制	是
LW-10539	(16bit) : 設備 8 參數 2 (SIEMENS S7/400 CPU 插槽)	讀/寫	讀/控制	讀/控制	是
LW-10655	(16bit) : 設備 32 超時 (單位 : 100ms)	讀/寫	讀/控制	讀/控制	是

LW-10656	(16bit) : 設備 32 通訊延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10657	(16bit) : 設備 32 ACK 訊號延時 (單位 : ms)	讀/寫	讀/控制	讀/控制	是
LW-10658	(16bit) : 設備 32 參數 1	讀/寫	讀/控制	讀/控制	是
LW-10659	(16bit) : 設備 32 參數 2	讀/寫	讀/控制	讀/控制	是

 Note

1. 傳輸速率分別為 0:4800, 1:9600, 2:19200, 3:38400, 4:57600, 5:115200, 6:187.5K, 7:1200, 8:2400, 10:14400, 11:28800, 12:76800。

#### 22.3.14. 與設備 (COM) 的通訊狀態與控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9150	自動連結設備 1 (COM 1) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9151	自動連結設備 2 (COM 2) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9152	自動連結設備 3 (COM 3) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9200	與設備 1 的通訊狀態 (站號 0, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9201	與設備 1 的通訊狀態 (站號 1, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9202	與設備 1 的通訊狀態 (站號 2, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9203	與設備 1 的通訊狀態 (站號 3, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9204	與設備 1 的通訊狀態 (站號 4, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9205	與設備 1 的通訊狀態 (站號 5, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9206	與設備 1 的通訊狀態 (站號 6, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9207	與設備 1 的通訊狀態 (站號 7, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9455	與設備 1 的通訊狀態 (站號 255, COM 1), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9500	與設備 2 的通訊狀態 (站號 0, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9501	與設備 2 的通訊狀態 (站號 1, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LB-9502	與設備 2 的通訊狀態 (站號 2, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9503	與設備 2 的通訊狀態 (站號 3, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9504	與設備 2 的通訊狀態 (站號 4, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9505	與設備 2 的通訊狀態 (站號 5, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9506	與設備 2 的通訊狀態 (站號 6, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9507	與設備 2 的通訊狀態 (站號 7, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9755	與設備 2 的通訊狀態 (站號 255, COM 2), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9800	與設備 3 的通訊狀態 (站號 0, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9801	與設備 3 的通訊狀態 (站號 1, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9802	與設備 3 的通訊狀態 (站號 2, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9803	與設備 3 的通訊狀態 (站號 3, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9804	與設備 3 的通訊狀態 (站號 4, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9805	與設備 3 的通訊狀態 (站號 5, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9806	與設備 3 的通訊狀態 (站號 6, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9807	與設備 3 的通訊狀態 (站號 7, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10055	與設備 3 的通訊狀態 (站號 255, COM 3), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12030	COM 1 開啟狀態指示 (OFF : 正常, ON : 開啟失敗) *註 1	讀	讀	讀	是
LB-12031	COM 2 開啟狀態指示 (OFF : 正常, ON : 開啟失敗)	讀	讀	讀	是
LB-12032	COM 3 開啟狀態指示 (OFF : 正常, ON : 開啟失敗)	讀	讀	讀	是
LB-12033	COM 4 開啟狀態指示 (OFF : 正常, ON : 開啟失敗)	讀	讀	讀	是

LB-12034	COM 5 開啟狀態指示 (OFF：正常, ON：開啟失敗)	讀	讀	讀	是
LB-12035	COM 6 開啟狀態指示 (OFF：正常, ON：開啟失敗)	讀	讀	讀	是
LB-12036	COM 7 開啟狀態指示 (OFF：正常, ON：開啟失敗)	讀	讀	讀	是
LB-12037	COM 8 開啟狀態指示 (OFF：正常, ON：開啟失敗)	讀	讀	讀	是
LB-12038	COM 9 開啟狀態指示 (OFF：正常, ON：開啟失敗)	讀	讀	讀	是
LW-9351	(16bit)：設備 1 (COM 1) 尚未處理的命令數目	讀	讀	讀	是
LW-9352	(16bit)：設備 2 (COM 2) 尚未處理的命令數目	讀	讀	讀	是
LW-9353	(16bit)：設備 3 (COM 3) 尚未處理的命令數目	讀	讀	讀	是



1. COM 的開啟狀態指示可使用於 PC 模擬時，查看 COM 是否被其他程式占用。

### 22.3.15. 與設備 (乙太網路) 的通訊狀態與控制

位址	描述	讀 / 寫 / 控制			CMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9153	自動連結設備 4 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9154	自動連結設備 5 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9155	自動連結設備 6 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9156	自動連結設備 7 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9157	自動連結設備 8 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9158	自動連結設備 9 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9189	自動連結設備 40 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-10070	當線上更改設備 4 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-10071	當線上更改設備 5 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-10072	當線上更改設備 6 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-10073	當線上更改設備 7 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-10074	當線上更改設備 8 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-10075	當線上更改設備 9 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-10099	當線上更改設備 33 (乙太網路) 的 IP 或系統參數時，設 ON 重新連結	讀/寫	讀/控制	讀/控制	是

LB-10100	與設備 4 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10101	與設備 4 的通訊狀態 (站號 0, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10102	與設備 4 的通訊狀態 (站號 1, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10103	與設備 4 的通訊狀態 (站號 2, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10104	與設備 4 的通訊狀態 (站號 3, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10105	與設備 4 的通訊狀態 (站號 4, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10106	與設備 4 的通訊狀態 (站號 5, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10107	與設備 4 的通訊狀態 (站號 6, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10108	與設備 4 的通訊狀態 (站號 7, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10356	與設備 4 的通訊狀態 (站號 255, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10400	與設備 5 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10401	與設備 5 的通訊狀態 (站號 0, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10402	與設備 5 的通訊狀態 (站號 1, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10403	與設備 5 的通訊狀態 (站號 2, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10404	與設備 5 的通訊狀態 (站號 3, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10405	與設備 5 的通訊狀態 (站號 4, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10406	與設備 5 的通訊狀態 (站號 5, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10407	與設備 5 的通訊狀態 (站號 6, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10408	與設備 5 的通訊狀態 (站號 7, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是

LB-10656	與設備 5 的通訊狀態 (站號 255, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10700	與設備 6 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10701	與設備 6 的通訊狀態 (站號 0, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10702	與設備 6 的通訊狀態 (站號 1, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10703	與設備 6 的通訊狀態 (站號 2, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10704	與設備 6 的通訊狀態 (站號 3, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10705	與設備 6 的通訊狀態 (站號 4, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10706	與設備 6 的通訊狀態 (站號 5, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10707	與設備 6 的通訊狀態 (站號 6, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10708	與設備 6 的通訊狀態 (站號 7, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-10956	與設備 6 的通訊狀態 (站號 255, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11000	與設備 7 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11001	與設備 7 的通訊狀態 (站號 0, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11002	與設備 7 的通訊狀態 (站號 1, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11003	與設備 7 的通訊狀態 (站號 2, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11004	與設備 7 的通訊狀態 (站號 3, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11005	與設備 7 的通訊狀態 (站號 4, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11006	與設備 7 的通訊狀態 (站號 5, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11007	與設備 7 的通訊狀態 (站號 6, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是

LB-11008	與設備 7 的通訊狀態 (站號 7, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11256	與設備 7 的通訊狀態 (站號 255, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11300	與設備 8 的通訊狀態 (乙太網路), 設 ON 重連 一次	讀/寫	讀/控制	讀/控制	是
LB-11301	與設備 8 的通訊狀態 (站號 0, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11302	與設備 8 的通訊狀態 (站號 1, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11303	與設備 8 的通訊狀態 (站號 2, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11304	與設備 8 的通訊狀態 (站號 3, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11305	與設備 8 的通訊狀態 (站號 4, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11306	與設備 8 的通訊狀態 (站號 5, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11307	與設備 8 的通訊狀態 (站號 6, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11308	與設備 8 的通訊狀態 (站號 7, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11556	與設備 8 的通訊狀態 (站號 255, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11600	與設備 9 的通訊狀態 (乙太網路), 設 ON 重連 一次	讀/寫	讀/控制	讀/控制	是
LB-11601	與設備 9 的通訊狀態 (站號 0, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11602	與設備 9 的通訊狀態 (站號 1, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11603	與設備 9 的通訊狀態 (站號 2, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11604	與設備 9 的通訊狀態 (站號 3, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11605	與設備 9 的通訊狀態 (站號 4, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11606	與設備 9 的通訊狀態 (站號 5, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LB-11607	與設備 9 的通訊狀態 (站號 6, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11608	與設備 9 的通訊狀態 (站號 7, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11856	與設備 9 的通訊狀態 (站號 255, 乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11900	與設備 10 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11901	與設備 11 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11902	與設備 12 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11903	與設備 13 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11904	與設備 14 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11905	與設備 15 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11906	與設備 16 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-11939	與設備 49 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12670	自動連結設備 41 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12671	自動連結設備 42 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12672	自動連結設備 43 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12673	自動連結設備 44 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12693	自動連結設備 64 (乙太網路) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12700	與設備 50 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12701	與設備 51 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12702	與設備 52 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12703	與設備 53 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12714	與設備 64 的通訊狀態 (乙太網路), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是

LB-12720	當線上更改設備 34(乙太網路) 的 IP 或系統參數時, 設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-12721	當線上更改設備 35(乙太網路) 的 IP 或系統參數時, 設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-12722	當線上更改設備 36(乙太網路) 的 IP 或系統參數時, 設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-12723	當線上更改設備 37(乙太網路) 的 IP 或系統參數時, 設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LB-12750	當線上更改設備 64(乙太網路) 的 IP 或系統參數時, 設 ON 重新連結	讀/寫	讀/控制	讀/控制	是
LW-9354	(16bit)：設備 4(乙太網路) 尚未處理的命令數目	讀	讀	讀	是
LW-9355	(16bit)：設備 5(乙太網路) 尚未處理的命令數目	讀	讀	讀	是
LW-9356	(16bit)：設備 6(乙太網路) 尚未處理的命令數目	讀	讀	讀	是
LW-9357	(16bit)：設備 7(乙太網路) 尚未處理的命令數目	讀	讀	讀	是
LW-9389	(16bit)：設備 39(乙太網路) 尚未處理的命令數目	讀	讀	讀	是
LW-9600	(16bit)：設備 4 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9601	(16bit)：設備 4 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9602	(16bit)：設備 4 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9603	(16bit)：設備 4 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9604	(16bit)：設備 4 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9605	(16bit)：設備 5 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9606	(16bit)：設備 5 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9607	(16bit)：設備 5 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9608	(16bit)：設備 5 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9609	(16bit)：設備 5 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9610	(16bit)：設備 6 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9611	(16bit)：設備 6 的 IP1 (IP 位址 =	讀/寫	讀/控制	讀/控制	是

	IPO:IP1:IP2:IP3)				
LW-9612	(16bit)：設備 6 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9613	(16bit)：設備 6 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9614	(16bit)：設備 6 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9615	(16bit)：設備 7 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9616	(16bit)：設備 7 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9617	(16bit)：設備 7 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9618	(16bit)：設備 7 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9619	(16bit)：設備 7 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9620	(16bit)：設備 8 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9621	(16bit)：設備 8 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9622	(16bit)：設備 8 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9623	(16bit)：設備 8 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9624	(16bit)：設備 8 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9625	(16bit)：設備 9 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9626	(16bit)：設備 9 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9627	(16bit)：設備 9 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9628	(16bit)：設備 9 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9629	(16bit)：設備 9 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9765	(16bit)：設備 37 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9766	(16bit)：設備 37 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-9767	(16bit)：設備 37 的 IP2 (IP 位址 = IP0:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9768	(16bit)：設備 37 的 IP3 (IP 位址 = IP0:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9769	(16bit)：設備 37 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-11472	(16bit)：設備 4 的 ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11473	(16bit)：設備 4 的 ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11474	(16bit)：設備 4 的 ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11475	(16bit)：設備 4 的 ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11476	(16bit)：設備 4 的 ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11477	(16bit)：設備 4 的 ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11478	(16bit)：設備 5 的 ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11479	(16bit)：設備 5 的 ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11480	(16bit)：設備 5 的 ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11481	(16bit)：設備 5 的 ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11482	(16bit)：設備 5 的 ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11483	(16bit)：設備 5 的 ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11484	(16bit)：設備 6 的 ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11485	(16bit)：設備 6 的 ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11486	(16bit)：設備 6 的 ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11487	(16bit)：設備 6 的 ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11488	(16bit)：設備 6 的 ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

	ID0:ID1:ID2:ID3:ID4:ID5)				
LW-11489	(16bit)：設備 6 的 ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11490	(16bit)：設備 7 的 ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11491	(16bit)：設備 7 的 ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11492	(16bit)：設備 7 的 ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11493	(16bit)：設備 7 的 ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11494	(16bit)：設備 7 的 ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-11495	(16bit)：設備 7 的 ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	讀/寫	讀/控制	讀/控制	是
LW-12110	(16bit)：設備 38 的 IP0 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12111	(16bit)：設備 38 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12112	(16bit)：設備 38 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12113	(16bit)：設備 38 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12114	(16bit)：設備 38 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12115	(16bit)：設備 39 的 IP0 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12116	(16bit)：設備 39 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12117	(16bit)：設備 39 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12118	(16bit)：設備 39 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12119	(16bit)：設備 39 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12120	(16bit)：設備 40 的 IP0 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12121	(16bit)：設備 40 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

LW-12122	(16bit)：設備 40 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12123	(16bit)：設備 40 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12124	(16bit)：設備 40 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12125	(16bit)：設備 41 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12126	(16bit)：設備 41 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12127	(16bit)：設備 41 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12128	(16bit)：設備 41 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12129	(16bit)：設備 41 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12240	(16bit)：設備 64 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12241	(16bit)：設備 64 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12242	(16bit)：設備 64 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12243	(16bit)：設備 64 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12244	(16bit)：設備 64 的連接埠	讀/寫	讀/控制	讀/控制	是



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.16. 與設備 (USB) 的通訊狀態與控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9190	自動連結設備 (USB) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9191	與設備的通訊狀態 (USB), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LW-9390	(16bit)：設備 (USB) 尚未處理的命令數目	讀	讀	讀	是

### 22.3.17. 與設備 (CAN Bus) 的通訊狀態與控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12080	自動連結設備 (CAN Bus) (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12081	與設備的通訊狀態 (CAN Bus), 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12100	暫停 CAN Bus 設備 1 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12101	暫停 CAN Bus 設備 2 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12102	暫停 CAN Bus 設備 3 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12103	暫停 CAN Bus 設備 4 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12104	暫停 CAN Bus 設備 5 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12105	暫停 CAN Bus 設備 6 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12106	暫停 CAN Bus 設備 7 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12107	暫停 CAN Bus 設備 8 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12108	暫停 CAN Bus 設備 9 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12109	暫停 CAN Bus 設備 10 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12354	暫停 CAN Bus 設備 255 的通訊 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LW-9392	(16bit)：設備 (CAN Bus) 尚未處理的命令數目	讀	讀	讀	是

### 22.3.18. 與遠端 HMI 的通訊狀態與控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9068	自動連結遠端 HMI 1 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9069	自動連結遠端 HMI 2 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9070	自動連結遠端 HMI 3 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9071	自動連結遠端 HMI 4 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9072	自動連結遠端 HMI 5 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9073	自動連結遠端 HMI 6 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9074	自動連結遠端 HMI 7 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9075	自動連結遠端 HMI 8 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是

LB-9099	自動連結遠端 HMI 32 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9100	與遠端 HMI 1 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9101	與遠端 HMI 2 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9102	與遠端 HMI 3 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9103	與遠端 HMI 4 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9104	與遠端 HMI 5 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9105	與遠端 HMI 6 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9106	與遠端 HMI 7 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9107	與遠端 HMI 8 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-9148	與遠端 HMI 49 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	否
LB-9149	當線上更改遠端 HMI 的 IP 時, 設 ON 重新連結遠端 HMI	讀/寫	讀/控制	讀/控制	是
LB-12754	自動連結遠端 HMI 33 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12755	自動連結遠端 HMI 34 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12756	自動連結遠端 HMI 35 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12757	自動連結遠端 HMI 36 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12758	自動連結遠端 HMI 37 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12759	自動連結遠端 HMI 38 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12760	自動連結遠端 HMI 39 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12761	自動連結遠端 HMI 40 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12762	自動連結遠端 HMI 41 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12763	自動連結遠端 HMI 42 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12764	自動連結遠端 HMI 43 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12765	自動連結遠端 HMI 44 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12766	自動連結遠端 HMI 45 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12767	自動連結遠端 HMI 46 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12768	自動連結遠端 HMI 47 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12769	自動連結遠端 HMI 48 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12770	自動連結遠端 HMI 49 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12771	自動連結遠端 HMI 50 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12772	自動連結遠端 HMI 51 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12773	自動連結遠端 HMI 52 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12774	自動連結遠端 HMI 53 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12775	自動連結遠端 HMI 54 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12776	自動連結遠端 HMI 55 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12777	自動連結遠端 HMI 56 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12778	自動連結遠端 HMI 57 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是

LB-12779	自動連結遠端 HMI 58 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12780	自動連結遠端 HMI 59 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12781	自動連結遠端 HMI 60 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12782	自動連結遠端 HMI 61 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12783	自動連結遠端 HMI 62 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12784	自動連結遠端 HMI 63 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12785	自動連結遠端 HMI 64 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12786	與遠端 HMI 50 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12787	與遠端 HMI 51 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12788	與遠端 HMI 52 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12789	與遠端 HMI 53 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12790	與遠端 HMI 54 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12791	與遠端 HMI 55 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12792	與遠端 HMI 56 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12793	與遠端 HMI 57 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12794	與遠端 HMI 58 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12795	與遠端 HMI 59 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12796	與遠端 HMI 60 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12797	與遠端 HMI 61 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12798	與遠端 HMI 62 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12799	與遠端 HMI 63 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LB-12800	與遠端 HMI 64 的通訊狀態, 設 ON 重連一次	讀/寫	讀/控制	讀/控制	是
LW-9800	(16bit) : 遠端 HMI 1 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9801	(16bit) : 遠端 HMI 1 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9802	(16bit) : 遠端 HMI 1 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9803	(16bit) : 遠端 HMI 1 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9804	(16bit) : 遠端 HMI 1 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9805	(16bit) : 遠端 HMI 2 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9806	(16bit) : 遠端 HMI 2 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9807	(16bit) : 遠端 HMI 2 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-9808	(16bit) : 遠端 HMI 2 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9809	(16bit) : 遠端 HMI 2 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9810	(16bit) : 遠端 HMI 3 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9811	(16bit) : 遠端 HMI 3 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9812	(16bit) : 遠端 HMI 3 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9813	(16bit) : 遠端 HMI 3 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9814	(16bit) : 遠端 HMI 3 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9815	(16bit) : 遠端 HMI 4 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9816	(16bit) : 遠端 HMI 4 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9817	(16bit) : 遠端 HMI 4 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9818	(16bit) : 遠端 HMI 4 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9819	(16bit) : 遠端 HMI 4 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9820	(16bit) : 遠端 HMI 5 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9821	(16bit) : 遠端 HMI 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9822	(16bit) : 遠端 HMI 5 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9823	(16bit) : 遠端 HMI 5 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9824	(16bit) : 遠端 HMI 5 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9825	(16bit) : 遠端 HMI 6 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9826	(16bit) : 遠端 HMI 6 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9827	(16bit) : 遠端 HMI 6 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9828	(16bit) : 遠端 HMI 6 的 IP3 (IP 位址 =	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

	IPO:IP1:IP2:IP3)				
LW-9829	(16bit)：遠端 HMI 6 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9830	(16bit)：遠端 HMI 7 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9831	(16bit)：遠端 HMI 7 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9832	(16bit)：遠端 HMI 7 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9833	(16bit)：遠端 HMI 7 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9834	(16bit)：遠端 HMI 7 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9835	(16bit)：遠端 HMI 8 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9836	(16bit)：遠端 HMI 8 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9837	(16bit)：遠端 HMI 8 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9838	(16bit)：遠端 HMI 8 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9839	(16bit)：遠端 HMI 8 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9895	(16bit)：遠端 HMI 20 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9896	(16bit)：遠端 HMI 20 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9897	(16bit)：遠端 HMI 20 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9898	(16bit)：遠端 HMI 20 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9899	(16bit)：遠端 HMI 20 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9905	(16bit)：遠端 HMI 21 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9906	(16bit)：遠端 HMI 21 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9907	(16bit)：遠端 HMI 21 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9908	(16bit)：遠端 HMI 21 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-9909	(16bit) : 遠端 HMI 21 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9910	(16bit) : 遠端 HMI 22 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9911	(16bit) : 遠端 HMI 22 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9912	(16bit) : 遠端 HMI 22 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9913	(16bit) : 遠端 HMI 22 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9914	(16bit) : 遠端 HMI 22 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9915	(16bit) : 遠端 HMI 23 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9916	(16bit) : 遠端 HMI 23 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9917	(16bit) : 遠端 HMI 23 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9918	(16bit) : 遠端 HMI 23 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9919	(16bit) : 遠端 HMI 23 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9920	(16bit) : 遠端 HMI 24 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9921	(16bit) : 遠端 HMI 24 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9922	(16bit) : 遠端 HMI 24 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9923	(16bit) : 遠端 HMI 24 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9924	(16bit) : 遠端 HMI 24 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9925	(16bit) : 遠端 HMI 25 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9926	(16bit) : 遠端 HMI 25 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9927	(16bit) : 遠端 HMI 25 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9928	(16bit) : 遠端 HMI 25 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9929	(16bit) : 遠端 HMI 25 的連接埠	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-9930	(16bit) : 遠端 HMI 26 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9931	(16bit) : 遠端 HMI 26 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9932	(16bit) : 遠端 HMI 26 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9933	(16bit) : 遠端 HMI 26 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9934	(16bit) : 遠端 HMI 26 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9935	(16bit) : 遠端 HMI 27 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9936	(16bit) : 遠端 HMI 27 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9937	(16bit) : 遠端 HMI 27 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9938	(16bit) : 遠端 HMI 27 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9939	(16bit) : 遠端 HMI 27 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9940	(16bit) : 遠端 HMI 28 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9941	(16bit) : 遠端 HMI 28 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9942	(16bit) : 遠端 HMI 28 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9943	(16bit) : 遠端 HMI 28 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9944	(16bit) : 遠端 HMI 28 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9945	(16bit) : 遠端 HMI 29 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9946	(16bit) : 遠端 HMI 29 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9947	(16bit) : 遠端 HMI 29 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9948	(16bit) : 遠端 HMI 29 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9949	(16bit) : 遠端 HMI 29 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9950	(16bit) : 遠端 HMI 30 的 IPO (IP 位址 =	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

	IPO:IP1:IP2:IP3)				
LW-9951	(16bit)：遠端 HMI 30 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9952	(16bit)：遠端 HMI 30 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9953	(16bit)：遠端 HMI 30 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9954	(16bit)：遠端 HMI 30 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9955	(16bit)：遠端 HMI 31 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9956	(16bit)：遠端 HMI 31 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9957	(16bit)：遠端 HMI 31 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9958	(16bit)：遠端 HMI 31 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9959	(16bit)：遠端 HMI 31 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9960	(16bit)：遠端 HMI 32 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9961	(16bit)：遠端 HMI 32 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9962	(16bit)：遠端 HMI 32 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9963	(16bit)：遠端 HMI 32 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9964	(16bit)：遠端 HMI 32 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-9995	(16bit)：遠端 HMI 39 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9996	(16bit)：遠端 HMI 39 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9997	(16bit)：遠端 HMI 39 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9998	(16bit)：遠端 HMI 39 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-9999	(16bit)：遠端 HMI 39 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12506	(16bit)：遠端 HMI 40 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-12507	(16bit) : 遠端 HMI 40 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12508	(16bit) : 遠端 HMI 40 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12509	(16bit) : 遠端 HMI 40 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12510	(16bit) : 遠端 HMI 40 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12511	(16bit) : 遠端 HMI 41 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12512	(16bit) : 遠端 HMI 41 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12513	(16bit) : 遠端 HMI 41 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12514	(16bit) : 遠端 HMI 41 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12515	(16bit) : 遠端 HMI 41 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12516	(16bit) : 遠端 HMI 42 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12517	(16bit) : 遠端 HMI 42 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12518	(16bit) : 遠端 HMI 42 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12519	(16bit) : 遠端 HMI 42 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12520	(16bit) : 遠端 HMI 42 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12521	(16bit) : 遠端 HMI 43 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12522	(16bit) : 遠端 HMI 43 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12523	(16bit) : 遠端 HMI 43 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12524	(16bit) : 遠端 HMI 43 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12525	(16bit) : 遠端 HMI 43 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12526	(16bit) : 遠端 HMI 44 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12527	(16bit) : 遠端 HMI 44 的 IP1 (IP 位址 =	讀/寫	讀/控制	讀/控制	是

	IPO:IP1:IP2:IP3)				
LW-12528	(16bit)：遠端 HMI 44 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12529	(16bit)：遠端 HMI 44 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12530	(16bit)：遠端 HMI 44 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12531	(16bit)：遠端 HMI 45 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12532	(16bit)：遠端 HMI 45 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12533	(16bit)：遠端 HMI 45 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12534	(16bit)：遠端 HMI 45 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12535	(16bit)：遠端 HMI 45 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12536	(16bit)：遠端 HMI 46 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12537	(16bit)：遠端 HMI 46 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12538	(16bit)：遠端 HMI 46 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12539	(16bit)：遠端 HMI 46 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12540	(16bit)：遠端 HMI 46 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12541	(16bit)：遠端 HMI 47 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12542	(16bit)：遠端 HMI 47 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12543	(16bit)：遠端 HMI 47 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12544	(16bit)：遠端 HMI 47 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12545	(16bit)：遠端 HMI 47 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12546	(16bit)：遠端 HMI 48 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12547	(16bit)：遠端 HMI 48 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-12548	(16bit) : 遠端 HMI 48 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12549	(16bit) : 遠端 HMI 48 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12550	(16bit) : 遠端 HMI 48 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12551	(16bit) : 遠端 HMI 49 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12552	(16bit) : 遠端 HMI 49 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12553	(16bit) : 遠端 HMI 49 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12554	(16bit) : 遠端 HMI 49 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12555	(16bit) : 遠端 HMI 49 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12556	(16bit) : 遠端 HMI 50 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12557	(16bit) : 遠端 HMI 50 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12558	(16bit) : 遠端 HMI 50 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12559	(16bit) : 遠端 HMI 50 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12560	(16bit) : 遠端 HMI 50 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12561	(16bit) : 遠端 HMI 51 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12562	(16bit) : 遠端 HMI 51 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12563	(16bit) : 遠端 HMI 51 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12564	(16bit) : 遠端 HMI 51 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12565	(16bit) : 遠端 HMI 51 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12566	(16bit) : 遠端 HMI 52 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12567	(16bit) : 遠端 HMI 52 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12568	(16bit) : 遠端 HMI 52 的 IP2 (IP 位址 =	讀/寫	讀/控制	讀/控制	是

	IPO:IP1:IP2:IP3)				
LW-12569	(16bit)：遠端 HMI 52 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12570	(16bit)：遠端 HMI 52 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12571	(16bit)：遠端 HMI 53 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12572	(16bit)：遠端 HMI 53 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12573	(16bit)：遠端 HMI 53 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12574	(16bit)：遠端 HMI 53 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12575	(16bit)：遠端 HMI 53 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12576	(16bit)：遠端 HMI 54 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12577	(16bit)：遠端 HMI 54 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12578	(16bit)：遠端 HMI 54 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12579	(16bit)：遠端 HMI 54 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12580	(16bit)：遠端 HMI 54 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12581	(16bit)：遠端 HMI 55 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12582	(16bit)：遠端 HMI 55 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12583	(16bit)：遠端 HMI 55 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12584	(16bit)：遠端 HMI 55 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12585	(16bit)：遠端 HMI 55 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12586	(16bit)：遠端 HMI 56 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12587	(16bit)：遠端 HMI 56 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12588	(16bit)：遠端 HMI 56 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

LW-12589	(16bit) : 遠端 HMI 56 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12590	(16bit) : 遠端 HMI 56 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12591	(16bit) : 遠端 HMI 57 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12592	(16bit) : 遠端 HMI 57 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12593	(16bit) : 遠端 HMI 57 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12594	(16bit) : 遠端 HMI 57 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12595	(16bit) : 遠端 HMI 57 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12596	(16bit) : 遠端 HMI 58 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12597	(16bit) : 遠端 HMI 58 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12598	(16bit) : 遠端 HMI 58 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12599	(16bit) : 遠端 HMI 58 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12600	(16bit) : 遠端 HMI 58 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12601	(16bit) : 遠端 HMI 59 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12602	(16bit) : 遠端 HMI 59 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12603	(16bit) : 遠端 HMI 59 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12604	(16bit) : 遠端 HMI 59 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12605	(16bit) : 遠端 HMI 59 的連接埠	讀/寫	讀/控制	讀/控制	是
LW-12606	(16bit) : 遠端 HMI 60 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12607	(16bit) : 遠端 HMI 60 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12608	(16bit) : 遠端 HMI 60 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12609	(16bit) : 遠端 HMI 60 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

## 位址暫存器

	IPO:IP1:IP2:IP3)				
LW-12610	(16bit)：遠端 HMI 60 的連接埠 IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12611	(16bit)：遠端 HMI 61 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12612	(16bit)：遠端 HMI 61 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12613	(16bit)：遠端 HMI 61 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12614	(16bit)：遠端 HMI 61 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12615	(16bit)：遠端 HMI 61 的連接埠 IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12616	(16bit)：遠端 HMI 62 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12617	(16bit)：遠端 HMI 62 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12618	(16bit)：遠端 HMI 62 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12619	(16bit)：遠端 HMI 62 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12620	(16bit)：遠端 HMI 62 的連接埠 IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12621	(16bit)：遠端 HMI 63 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12622	(16bit)：遠端 HMI 63 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12623	(16bit)：遠端 HMI 63 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12624	(16bit)：遠端 HMI 63 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12625	(16bit)：遠端 HMI 63 的連接埠 IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12626	(16bit)：遠端 HMI 64 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12627	(16bit)：遠端 HMI 64 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12628	(16bit)：遠端 HMI 64 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是
LW-12629	(16bit)：遠端 HMI 64 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	是

LW-12630	(16bit)：遠端 HMI 64 的連接埠	讀/寫	讀/控制	讀/控制	是
----------	------------------------	-----	------	------	---

### 22.3.19. 與遠端設備的通訊狀態與控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-10050	(16bit)：連接遠端設備 1 的 HMI 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10051	(16bit)：連接遠端設備 1 的 HMI 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10052	(16bit)：連接遠端設備 1 的 HMI 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10053	(16bit)：連接遠端設備 1 的 HMI 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10054	(16bit)：連接遠端設備 1 的 HMI 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10055	(16bit)：連接遠端設備 2 的 HMI 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10056	(16bit)：連接遠端設備 2 的 HMI 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10057	(16bit)：連接遠端設備 2 的 HMI 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10058	(16bit)：連接遠端設備 2 的 HMI 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10059	(16bit)：連接遠端設備 2 的 HMI 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10060	(16bit)：連接遠端設備 3 的 HMI 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10061	(16bit)：連接遠端設備 3 的 HMI 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10062	(16bit)：連接遠端設備 3 的 HMI 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10063	(16bit)：連接遠端設備 3 的 HMI 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10064	(16bit)：連接遠端設備 3 的 HMI 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10065	(16bit)：連接遠端設備 4 的 HMI 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10066	(16bit)：連接遠端設備 4 的 HMI 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10067	(16bit)：連接遠端設備 4 的 HMI 的 IP2 (IP 位	讀/寫	讀/控制	讀/控制	否

## 位址暫存器

	址 = IPO:IP1:IP2:IP3)				
LW-10068	(16bit)：連接遠端設備 4 的 HMI 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10069	(16bit)：連接遠端設備 4 的 HMI 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10205	(16bit)：連接遠端設備 32 的 HMI 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10206	(16bit)：連接遠端設備 32 的 HMI 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10207	(16bit)：連接遠端設備 32 的 HMI 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10208	(16bit)：連接遠端設備 32 的 HMI 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10209	(16bit)：連接遠端設備 32 的 HMI 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10300	(16bit)：遠端設備 1 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10301	(16bit)：遠端設備 1 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10302	(16bit)：遠端設備 1 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10303	(16bit)：遠端設備 1 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10304	(16bit)：遠端設備 1 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10305	(16bit)：遠端設備 2 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10306	(16bit)：遠端設備 2 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10307	(16bit)：遠端設備 2 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10308	(16bit)：遠端設備 2 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10309	(16bit)：遠端設備 2 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10310	(16bit)：遠端設備 3 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10311	(16bit)：遠端設備 3 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10312	(16bit)：遠端設備 3 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否

## 位址暫存器

LW-10313	(16bit)：遠端設備 3 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10314	(16bit)：遠端設備 3 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10315	(16bit)：遠端設備 4 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10316	(16bit)：遠端設備 4 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10317	(16bit)：遠端設備 4 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10318	(16bit)：遠端設備 4 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10319	(16bit)：遠端設備 4 的連接埠	讀/寫	讀/控制	讀/控制	否
LW-10455	(16bit)：遠端設備 32 的 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10456	(16bit)：遠端設備 32 的 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10457	(16bit)：遠端設備 32 的 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10458	(16bit)：遠端設備 32 的 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-10459	(16bit)：遠端設備 32 的連接埠	讀/寫	讀/控制	讀/控制	否

## 22.3.20. 本機/遠端操作限制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9044	禁止遠端控制 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9053	禁止遠端讀取密碼操作 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9054	禁止遠端寫入密碼操作 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9196	本機 HMI 只支援檢視功能 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9197	只允許遠端 HMI 使用檢視功能 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9198	禁止本機 HMI 觸發巨集 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-9199	禁止遠端 HMI 觸發巨集 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是

## 22.3.21. 通訊錯誤碼

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9400	(16bit)：與設備 1 通訊錯誤時產生的錯誤碼	讀	讀	讀	是

LW-9401	(16bit)：與設備 2 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9402	(16bit)：與設備 3 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9403	(16bit)：與設備 4 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9404	(16bit)：與設備 5 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9405	(16bit)：與設備 6 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9406	(16bit)：與設備 7 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9407	(16bit)：與設備 8 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9449	(16bit)：與設備 50 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9490	(16bit)：與設備 (USB) 通訊錯誤時產生的錯誤碼	讀	讀	讀	是
LW-9492	(16bit)：與設備 (CAN-Bus) 通訊錯誤時產生的錯誤碼	讀	讀	讀	是

### Note

#### 1. 通訊錯誤碼解釋如下：

錯誤碼	通訊錯誤原因
0	正常
1	裝置忙線無法再接收命令
2	通訊錯誤 (原因不明)
3	裝置不存在
4	指定站號的裝置不存在
5	位址格式錯誤
6	讀取/寫入不支援的位址
7	裝置使用的驅動程式不存在
8	序列埠(COM Port)不存在
9	裝置的 IP 位址錯誤或是無法連線到該裝置
10	裝置所回覆的內容檢核錯誤(checksum error)
11	無法辨識的命令
12	忽略此次通訊
20	未正確連接使用 USB 介面的裝置
21	未正確連接使用 CAN Bus 介面的裝置
22	未接受到來自裝置的任何回覆
23	未在指定的時間內(timeout)自裝置讀取到足夠數量的數據
24	物件所使用的轉換標籤(Conversion Tag)不存在或是內容錯誤
25	HMI 拒絕接收來自 Remote HMI 的命令
251	讀取/寫入 MODBUS 裝置暫存器的字數(word no.)超過允許值
252	MODBUS 裝置所回覆數據的格式不正確
253	MODBUS 裝置所回覆數據檢核錯誤(checksum error)

### 22.3.22. 驅動程式 ID

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9300	(16bit) : 連接在本機的設備 1 所使用的驅動程式編號	讀	讀	讀	是
LW-9301	(16bit) : 連接在本機的設備 2 所使用的驅動程式編號	讀	讀	讀	是
LW-9302	(16bit) : 連接在本機的設備 3 所使用的驅動程式編號	讀	讀	讀	是
LW-9303	(16bit) : 連接在本機的設備 4 所使用的驅動程式編號	讀	讀	讀	是
LW-9331	(16bit) : 連接在本機的設備 32 所使用的驅動程式編號	讀	讀	讀	是

### 22.3.23. DLT645 控制器

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-10700	(4 words) : DLT_645 使用者 (COM 1)	讀/寫	讀/控制	讀/控制	是
LW-10704	(4 words) : DLT_645 密碼 (COM 1)	讀/寫	讀/控制	讀/控制	是
LW-10708	(6 words) : DLT_645 位址 (COM 1)	讀/寫	讀/控制	讀/控制	是
LW-10715	(4 words) : DLT_645 使用者 (COM 2)	讀/寫	讀/控制	讀/控制	是
LW-10719	(4 words) : DLT_645 密碼 (COM 2)	讀/寫	讀/控制	讀/控制	是
LW-10723	(6 words) : DLT_645 位址 (COM 2)	讀/寫	讀/控制	讀/控制	是
LW-10730	(4 words) : DLT_645 使用者 (COM 3)	讀/寫	讀/控制	讀/控制	是
LW-10734	(4 words) : DLT_645 密碼 (COM 3)	讀/寫	讀/控制	讀/控制	是
LW-10738	(6 words) : DLT_645 位址 (COM 3)	讀/寫	讀/控制	讀/控制	是

### 22.3.24. [Device No Response] 視窗控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9192	禁止彈出設備 (USB) 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11960	禁止彈出設備 1 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11961	禁止彈出設備 2 的 "Device No Response" 視窗	讀/寫	讀/控制	讀/控制	是

	(當狀態為 ON)				
LB-11962	禁止彈出設備 3 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11963	禁止彈出設備 4 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11964	禁止彈出設備 5 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11965	禁止彈出設備 6 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11966	禁止彈出設備 7 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-11967	禁止彈出設備 8 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12023	禁止彈出設備 64 的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12082	禁止彈出 CAN Bus 設備的 "Device No Response" 視窗 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是

### 22.3.25. [快選] 視窗控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9013	隱藏 (設 ON)/顯示 (設 OFF) 快選視窗	讀/寫	讀/控制	讀/控制	否
LB-9014	隱藏 (設 ON)/顯示 (設 OFF) 快選按鍵	讀/寫	讀/控制	讀/控制	否
LB-9015	隱藏 (設 ON)/顯示 (設 OFF) 快選視窗/按鍵	讀/寫	讀/控制	讀/控制	否

### 22.3.26. EasyAccess

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9051	與 EasyAccess 伺服器斷線 (設 OFF)/連線 (設 ON)	讀/寫	讀/控制	讀/控制	否
LB-9052	與 EasyAccess 伺服器連線狀態 (當連線中狀態為 ON)	讀	讀	讀	否

 關於 EasyAccess 的更多詳情，請參閱網址 <http://www.ihmi.net/>。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.27. EasyAccess 2.0

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-10820	(16bit) : 取消 (設為 0)/啟用 (設為 1) (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-10821	(5 words) : session ID (EasyAccess 2.0)	讀	讀	讀	是
LW-10826	(2 words) : 密碼 (EasyAccess 2.0)	讀	讀	讀	是
LW-10828	(16bit) : 執行狀態 (EasyAccess 2.0)	讀	讀	讀	是
LW-10829	(16bit) : 最後錯誤碼 (EasyAccess 2.0)	讀	讀	讀	是
LW-11170	(16bit) : Proxy 代理伺服器 取消/啟用 (0:取消, 1:啟用) (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11171	(16bit) : Proxy 代理伺服器類型 (0:HTTP, 1:SOCKSv4, 2:SOCKSv5) (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11172	(16bit) : Proxy 代理伺服器 IPO (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11173	(16bit) : Proxy 代理伺服器 IP1 (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11174	(16bit) : Proxy 代理伺服器 IP2 (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11175	(16bit) : Proxy 代理伺服器 IP3 (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11176	(16bit) : Proxy 代理伺服器埠號 (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11177	(16bit) : Proxy 代理伺服器認證 (0:取消, 1:啟用) (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11178	(16 words) : Proxy 代理伺服器使用者名稱 (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11194	(16 words) : Proxy 代理伺服器密碼 (EasyAccess 2.0)	讀/寫	讀/控制	讀/控制	是
LW-11210	(20 words) : 硬體金鑰 (EasyAccess 2.0)	讀	讀	讀	是
LW-11296	(16bit) : EasyAccess 2.0 伺服器位置 (0 : 全球, 1 : 中國地區)	讀/寫	讀/控制	讀/控制	是
LW-11770	(64 words) : 微信推播通知的二維條碼 (URL) (EasyAccess 2.0)	讀	讀	讀	是
LW-11982	(16 words) : HMI 網域 (EasyAccess 2.0)	讀	讀	讀	是
LW-12773	(16 words) : 已連接用戶數量 (EasyAccess 2.0)	讀	讀	讀	是
LW-12774	(16 words) : 已連接用戶 1 (EasyAccess 2.0)	讀	讀	讀	是
LW-12790	(16 words) : 已連接用戶 2 (EasyAccess 2.0)	讀	讀	讀	是
LW-12806	(16 words) : 已連接用戶 3 (EasyAccess 2.0)	讀	讀	讀	是

### 22.3.28. 遠端列印/備份伺服器

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-10069	當線上更改遠端列印/備份伺服器的 IP 時，設 ON 重新連結遠端列印/備份伺服器	讀/寫	讀/控制	讀/控制	否
LB-12040	遠端列印/備份伺服器斷線警示 (當狀態為 ON)	讀	讀	讀	否
LW-9770	(16bit)：遠端列印/備份伺服器的 IPO (IP0:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-9771	(16bit)：遠端列印/備份伺服器的 IP1 (IP0:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-9772	(16bit)：遠端列印/備份伺服器的 IP2 (IP0:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-9773	(16bit)：遠端列印/備份伺服器的 IP3 (IP0:IP1:IP2:IP3)	讀/寫	讀/控制	讀/控制	否
LW-9774	(6 words)：登入遠端列印/備份伺服器所需的使用者名稱 *註 1	讀/寫	讀/控制	讀/控制	否
LW-9780	(6 words)：登入遠端列印/備份伺服器所需的密碼 *註 1	讀/寫	讀/控制	讀/控制	否

#### Note

1. 若欲使用 LW-9774 及 LW-9780 更改設定，必須重新啟動 HMI 此變更才有效。

請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.29. 穿透通訊設定

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9901	(16bit)：穿透通訊數據來源串列埠口 (1~3 : COM 1~COM 3)	讀/寫	讀/控制	讀/控制	是
LW-9902	(16bit)：穿透通訊數據目標串列埠口 (1~3 : COM 1~COM 3)	讀/寫	讀/控制	讀/控制	是
LW-9903	(16bit)：穿透通訊控制 (0: 正常, 1: 暫停, 2: 執行穿透功能時，停止 HMI 與設備間的通訊)	讀/寫	讀/控制	讀/控制	是
LW-9904	(16bit)：穿透伺服器連接埠 (2000~2100)	讀/寫	讀/控制	讀/控制	是
LW-10850	(16bit)：取消/啟用 (0: 取消, 1: 正常, 2: IP 限制) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是

LW-10851	(16bit)：目的端 COM 埠 (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10852	(16bit)：目的端 PLC 站號 (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10853	(16bit)：通訊協議 (0：未定義, 1：PPI, 2：MPI) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10854	(16bit)：連接 client 的 IP0 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10855	(16bit)：連接 client 的 IP1 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10856	(16bit)：連接 client 的 IP2 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10857	(16bit)：連接 client 的 IP3 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10858	(16bit)：指定的 client 的 IP0 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10859	(16bit)：指定的 client 的 IP1 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10860	(16bit)：指定的 client 的 IP2 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10861	(16bit)：指定的 client 的 IP3 (IP address = IP0:IP1:IP2:IP3) (siemens 穿透功能)	讀/寫	讀/控制	讀/控制	是
LW-10862	(16bit)：連接狀態 (0：就緒, 1：client 連接中) (siemens 穿透功能)	讀	讀	讀	是
LW-10863	(16bit)：執行狀態 (0：正常, 1：錯誤) (siemens 穿透功能)	讀	讀	讀	是
LW-10864	(16bit)：最後錯誤碼 (siemens 穿透功能)	讀	讀	讀	是



關於 Siemens 穿透功能的詳細說明，請參考《29 穿透通訊功能》。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.30. VNC 控制

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12088	啟用 VNC 監視模式 (當狀態為 ON) *註 1	讀/寫	讀/控制	讀/控制	是
LB-12089	VNC 不須密碼即可登入 (當狀態為 ON) *註 1	讀/寫	讀/控制	讀/控制	是
LB-12090	VNC 客戶端連接至 HMI (當狀態為 ON) (請使 用 OS 20120621 或更新版本的 OS)	讀	讀	讀	是

## 位址暫存器

LB-12091	當 VNC 客戶端連接至 HMI 時取消自動登出功能 (當狀態為 ON) (請使用 OS 20120621 或更新版本的 OS)	讀/寫	讀/控制	讀/控制	是
LB-12092	開啟 (設 ON)/取消 (設 OFF) VNC 功能	讀/寫	讀/控制	讀/控制	是
LB-12093	VNC 連線模式 (OFF：單台連線, ON：多台連線) *註 1	讀/寫	讀/控制	讀/控制	是
LB-12824	VNC 客戶端連接 HMI 時禁用觸控 (當狀態為 ON) (支援 OS 版本 20230526 或更新版本) *註 2	讀/寫	讀/寫	讀/寫	否
LB-12824 (cMT X 機種)	啟用/取消 VNC/WebView 互鎖	讀/寫	讀/控制	讀/控制	是
LW-9530	(4 words) : VNC 伺服器密碼	讀/寫	讀/控制	讀/控制	是
LW-12831	(32bit) : VNC/WebView 互鎖超時 (1 ~ 86400 秒)	讀/寫	讀/控制	讀/控制	是
LW-12833	(16bit) : VNC/WebView 互鎖狀態列 (0: 一般, 1: 最小化)	讀/寫	讀/控制	讀/控制	是

 Note

- 於 eMT、iE、XE、mTV、iP 系列，若欲更改 VNC 連線模式，必須使用 LB-12092 關閉 VNC 功能後再開啟此變更才有效。
- 當觸控控制權被切換至 VNC 客戶端時，視窗左上角將顯示圖示 。若啟用 VNC 監視模式，再啟用禁用 HMI 觸控，則控制權仍會保留在 HMI 端。在啟用禁用 HMI 觸控的當下，所接上的 USB 滑鼠控制也會被禁用，但重新插拔 USB 滑鼠將恢復其控制權。

## 22.3.31. HMI 和工程檔案識別碼

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9046	工程檔案識別碼與 HMI 識別碼不同 (當狀態為 ON)	讀	讀	讀	是
LW-9046	(32bit) : HMI 識別碼 *註 1	讀/寫	讀/控制	讀	是

 Note

- 若欲使用 LW-9046 更改 HMI 識別碼設定，必須重新啟動 HMI 此變更才有效。

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.32. USB 安全金鑰

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-11160	(16bit) : USB 安全金鑰啟始時間 - 年	讀	讀	讀	是
LW-11161	(16bit) : USB 安全金鑰啟始時間 - 月	讀	讀	讀	是
LW-11162	(16bit) : USB 安全金鑰啟始時間 - 日	讀	讀	讀	是
LW-11163	(16bit) : USB 安全金鑰啟始時間 - 時	讀	讀	讀	是
LW-11164	(16bit) : USB 安全金鑰啟始時間 - 分	讀	讀	讀	是
LW-11165	(16bit) : USB 安全金鑰結束時間 - 年	讀	讀	讀	是
LW-11166	(16bit) : USB 安全金鑰結束時間 - 月	讀	讀	讀	是
LW-11167	(16bit) : USB 安全金鑰結束時間 - 日	讀	讀	讀	是
LW-11168	(16bit) : USB 安全金鑰結束時間 - 時	讀	讀	讀	是
LW-11169	(16bit) : USB 安全金鑰結束時間 - 分	讀	讀	讀	是

### 22.3.33. 使用者名稱和密碼

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9050	使用者登出 *註 2	寫	控制	控制	否
LB-9060	密碼輸入錯誤指示 *註 2	讀	讀	讀	否
LB-9061	更新密碼 (設定為 ON) *註 2	寫	控制	控制	否
LB-12056	使用者操作未被授權的物件 (當狀態為 ON)	讀	讀	讀	是
PLB-12056	使用者操作未被授權的物件 (當狀態為 ON) (用於手持平板裝置)	讀	N/A	N/A	是
LW-9082	(16bit) : 自動登出時間 (單位 : 分鐘, 0 : 取消此功能)	讀/寫	讀/控制	讀/控制	是
LW-9219	(16bit) : 使用者編號 (1 ~ 12) *註 2	讀/寫	讀/控制	讀/控制	否
LW-9220	(32bit) : 密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9222	(16bit) : 當前使用者可使用的物件類別 (bit 0:A, bit 1:B, bit 2:C, ...)	讀	讀	讀	是
PLW-9222	(16bit) : 當前使用者 (用於手持平板裝置) 可使用的物件類別 (bit 0:A, bit 1:B, bit 2:C, ...)	讀	N/A	N/A	是
LW-9500	(32bit) : 使用者 1 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9502	(32bit) : 使用者 2 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9504	(32bit) : 使用者 3 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9506	(32bit) : 使用者 4 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9508	(32bit) : 使用者 5 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否

LW-9510	(32bit) : 使用者 6 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9512	(32bit) : 使用者 7 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9514	(32bit) : 使用者 8 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9516	(32bit) : 使用者 9 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9518	(32bit) : 使用者 10 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9520	(32bit) : 使用者 11 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-9522	(32bit) : 使用者 12 的密碼 *註 2	讀/寫	讀/控制	讀/控制	否
LW-10754	(8 words) : 當前登入的使用者名稱 *註 1	讀	讀	讀	是
PLW-10754	(8 words) : 當前使用者名稱 (用於手持平板裝置) *註 1	讀	N/A	N/A	是

 Note

1. 只支援於 [使用者密碼] » [進階安全模式]。
2. 只支援於 [使用者密碼] » [一般模式]。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.34. 巨集

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9059	關閉巨集指令 TRACE 功能 (當狀態為 ON) * 註 1	讀/寫	讀/控制	讀/控制	是
LW-10900	(16bit) : 巨集指令 0 狀態 (0:就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10901	(16bit) : 巨集指令 1 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10902	(16bit) : 巨集指令 2 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10903	(16bit) : 巨集指令 3 狀態(0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10904	(16bit) : 巨集指令 4 狀態(0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常	讀	讀	讀	是

	中止 (超出陣列範圍))				
LW-10905	(16bit) : 巨集指令 5 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10906	(16bit) : 巨集指令 6 狀態(0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10907	(16bit) : 巨集指令 7 狀態(0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10908	(16bit) : 巨集指令 8 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-10909	(16bit) : 巨集指令 9 狀態(0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-11154	(16bit) : 巨集指令 254 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12247	(16bit) : 巨集指令 255 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12248	(16bit) : 巨集指令 256 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12249	(16bit) : 巨集指令 257 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12250	(16bit) : 巨集指令 258 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12251	(16bit) : 巨集指令 259 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12252	(16bit) : 巨集指令 260 狀態 (0: 就緒, 3:執行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32:異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12253	(16bit) : 巨集指令 261 狀態 (0: 就緒, 3:執	讀	讀	讀	是

	行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32: 異常中止 (超出陣列範圍))				
LW-12254	(16bit) : 巨集指令 262 狀態 (0: 就緒, 3:執 行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32: 異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12255	(16bit) : 巨集指令 263 狀態 (0: 就緒, 3:執 行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32: 異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12256	(16bit) : 巨集指令 264 狀態 (0: 就緒, 3:執 行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32: 異常中止 (超出陣列範圍))	讀	讀	讀	是
LW-12491	(16bit) : 巨集指令 499 狀態 (0: 就緒, 3:執 行中, 5:等待回覆, 9:等待被同步, 17:延遲, 32: 異常中止 (超出陣列範圍))	讀	讀	讀	是

### Note

1. LB-9059 關閉巨集指令 TRACE 功能範例如下：

 請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

### 22.3.35. 輸入物件功能

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-9002	(32bit-float) : 數值輸入物件允許輸入的上限 值	讀	讀	讀	是
LW-9004	(32bit-float) : 數值輸入物件允許輸入的下限 值	讀	讀	讀	是
LW-9052	(32bit-float) : 數值輸入物件的前一次輸入值	讀	讀	讀	是
PLW-9052	(32bit-float) : 數值輸入物件的前一次輸入值	讀	N/A	N/A	是
LW-9150	(32 words) : 顯示當前鍵盤所輸入的資料 (ASCII)	讀	讀	讀	是
LW-9540	(16bit) : 鍵盤大小寫切換	讀/寫	讀/控制	讀/控制	是

### 22.3.36. 時間同步/夏令時間

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12055	時間同步失敗 (當狀態為 ON)	讀	讀	讀	是

LB-12355	夏令時間 (當狀態為 ON)	讀	讀	讀	是
LW-11260	(16bit) : 啟用/取消夏令時間 (DST) (0:取消, 1:啟用)	讀/寫	讀/控制	讀/控制	是
LW-11261	(16bit) : 時間調整值 (時)	讀/寫	讀/控制	讀/控制	是
LW-11262	(16bit) : 時間調整值 (分)	讀/寫	讀/控制	讀/控制	是
LW-11263	(16bit) : DST 的起始月份	讀/寫	讀/控制	讀/控制	是
LW-11264	(16bit) : DST 的起始週數 (1~5)	讀/寫	讀/控制	讀/控制	是
LW-11265	(16bit) : DST 的起始星期 (0~6)	讀/寫	讀/控制	讀/控制	是
LW-11266	(16bit) : 當 DST 啟用時的本機時間 (時)	讀/寫	讀/控制	讀/控制	是
LW-11267	(16bit) : 當 DST 啟用時的本機時間 (分)	讀/寫	讀/控制	讀/控制	是
LW-11268	(16bit) : DST 的結束月份	讀/寫	讀/控制	讀/控制	是
LW-11269	(16bit) : DST 的結束週數 (1~5)	讀/寫	讀/控制	讀/控制	是
LW-11270	(16bit) : DST 的結束星期 (0~6)	讀/寫	讀/控制	讀/控制	是
LW-11271	(16bit) : 當 DST 結束時的本機時間 (時)	讀/寫	讀/控制	讀/控制	是
LW-11272	(16bit) : 當 DST 結束時的本機時間 (分)	讀/寫	讀/控制	讀/控制	是
LW-11273	(16bit) : 啟用/取消透過 NTP (Network Time Protocol) 伺服器同步時間 (0:取消, 1:啟用)	讀/寫	讀/控制	讀/控制	是
LW-11274	(16bit) : 當 HMI 啟動時即執行時間同步 (0:取消, 1:啟用)	讀/寫	讀/控制	讀/控制	是
LW-11275	(16bit) : 伺服器的回覆時間已根據夏令時間調整 (0:取消, 1:啟用)	讀/寫	讀/控制	讀/控制	是
LW-11276	(16bit) : HMI 時區 (單位：分鐘)	讀/寫	讀/控制	讀/控制	是
LW-11277	(16bit) : 伺服器回應時間 (伺服器時區)(單位：分鐘)	讀/寫	讀/控制	讀/控制	是
LW-11278	(16bit) : 網路時間伺服器 1 的 IP 0 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11279	(16bit) : 網路時間伺服器 1 的 IP 1 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11280	(16bit) : 網路時間伺服器 1 的 IP 2 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11281	(16bit) : 網路時間伺服器 1 的 IP 3 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11282	(16bit) : 網路時間伺服器 2 的 IP 0 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11283	(16bit) : 網路時間伺服器 2 的 IP 1 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11284	(16bit) : 網路時間伺服器 2 的 IP 2 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是

	address = IPO:IP1:IP2:IP3) *註 1				
LW-11285	(16bit) : 網路時間伺服器 2 的 IP 3 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11286	(16bit) : 網路時間伺服器 3 的 IP 0 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11287	(16bit) : 網路時間伺服器 3 的 IP 1 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11288	(16bit) : 網路時間伺服器 3 的 IP 2 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11289	(16bit) : 網路時間伺服器 3 的 IP 3 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11290	(16bit) : 網路時間伺服器 4 的 IP 0 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11291	(16bit) : 網路時間伺服器 4 的 IP 1 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11292	(16bit) : 網路時間伺服器 4 的 IP 2 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11293	(16bit) : 網路時間伺服器 4 的 IP 3 (IP address = IPO:IP1:IP2:IP3) *註 1	讀/寫	讀/控制	讀/控制	是
LW-11294	(32bit) : 更新週期 (時間同步間隔) (10 ~ 86400, 單位 : 秒)	讀/寫	讀/控制	讀/控制	是

 Note

- 當系統參數設定 [時間同步/夏令時間] » [網路時間伺服器] 設定為靜態 IP 地址時，可以通過 LW-11278~11293 位址進行寫入/控制操作。如果 [網路時間伺服器] 設定為域名，則不支援通過 LW-11278~11293 位址修改網路時間伺服器的 IP 地址。

### 22.3.37. 行動網路

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-11297	(16 words) : SIM 卡的個人識別碼 PIN (行動 網路)	讀/寫	讀/控制	讀/控制	是
LW-11313	(16 words) : 存取點名稱 APN (行動網路)	讀/寫	讀/控制	讀/控制	是
LW-11329	(16 words) : 使用者名稱 (行動網路)	讀/寫	讀/控制	讀/控制	是
LW-11345	(16 words) : 密碼 (行動網路)	讀/寫	讀/控制	讀/控制	是
LW-11361	(16 words) : 電話號碼 (行動網路)	讀/寫	讀/控制	讀/控制	是
LW-11377	(16bit) : 停止 (設 0)/啟動 (設 1) 連接 (行	讀/寫	讀/控制	讀/控制	是

	動網路)				
LW-11378	(16bit) : 最後錯誤碼 (0:成功, 1:錯誤的 PIN 碼, 2:無 SIM 卡, 3:無設備, 4:puk 碼已鎖住, 5:其他) (行動網路)	讀	讀	讀	是
LW-11379	(16bit) : 連接狀態 (0:無設備, 1:斷線, 2:連線中, 3:已連線) (行動網路)	讀	讀	讀	是
LW-11380	(16bit) : 停止 (設 0)/啟動 (設 1) 連接 (USB 數據連線)	讀/寫	讀/控制	讀/控制	是
LW-11381	(16bit) : 連接狀態 (0:無設備, 1:斷線, 2:已連線, 3:失敗, 4:OS 不支援, 5:HMI 不支援) (USB 數據連線)	讀	讀	讀	是
LW-12631	(16bit) : 信號強度 (dBm) (0: 失敗, 其他: 信號強度) (行動網路)	讀	讀	讀	是
LW-12632	(8 words) : IMEI (行動網路)	讀	讀	讀	是
LW-12640	(10 words) : SIM 卡的 ICCID 代碼 (行動網路)	讀	讀	讀	是

### 22.3.38. WiFi 設定

特定有支援 WiFi 的機型才能使用相關的系統暫存器，例如: MT8103iE、cMT3103、cMT-SVR-200... 等等。

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12365	更新 wifi 設定 (IP, 子網路遮罩, 預設閘道, DNS) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12666	取消 (設 OFF)/開啟(設 ON) WiFi 热點	讀/寫	讀/控制	讀/控制	是
LB-12667	更新 WiFi 热點設定 (SSID, 密碼) (設定為 ON)	讀/寫	讀/控制	讀/控制	是
LB-12820	M02 狀態 (0: 未插入, 1: 已插入)	讀	讀	讀	是
LW-11383	(16bit) : WiFi 控制 (1: 斷線, 2: 連線, 3: 彈出設定視窗, 4: 透過 SSID 連線)	讀/寫	讀/控制	讀/控制	是
LW-11384	(16bit) : WiFi 錯誤碼 (0: 無錯誤, 1: 無裝置, 2: 訊號關閉, 3: 參數無效)	讀	讀	讀	是
LW-11385	(16bit) : WiFi 狀態 (0: 已停止; 1: 連接中; 2: 已連接)	讀	讀	讀	是
LW-11386	(16 words) : WiFi 已連接的 SSID	讀	讀	讀	是
LW-11402	(16bit) : WiFi 訊號等級 (0: 無, 1: 微弱, 2: 普通, 3: 良好, 4: 強) *註 2	讀	讀	讀	是
LW-11403	(16bit) : WiFi 國碼 *註 1	讀/寫	讀/控制	讀/控制	是

LW-11404	(16bit) : WiFi 訊號 (0: 關閉, 1: 開啟)	讀/寫	讀/控制	讀/控制	是
LW-11405	(16bit) : WiFi 訊號強度 (dBm) (0, 1, 2: 失敗, 其他: 訊號強度)	讀	讀	讀	是
LW-11410	(16bit) : HMI WiFi IP 0 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11411	(16bit) : HMI WiFi IP 1 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11412	(16bit) : HMI WiFi IP 2 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11413	(16bit) : HMI WiFi IP 3 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11414	(16bit) : HMI WiFi 遮罩 0 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11415	(16bit) : HMI WiFi 遮罩 1 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11416	(16bit) : HMI WiFi 遮罩 2 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11417	(16bit) : HMI WiFi 遮罩 3 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11418	(16bit) : HMI WiFi 閘道 0 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11419	(16bit) : HMI WiFi 閘道 1 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11420	(16bit) : HMI WiFi 閘道 2 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11421	(16bit) : HMI WiFi 閘道 3 (只在 HMI 上有效)	讀/寫	讀/控制	讀/控制	是
LW-11422	(16bit) : HMI WiFi 實體位址 (MAC) 0	讀	讀	讀	是
LW-11423	(16bit) : HMI WiFi 實體位址 (MAC) 1	讀	讀	讀	是
LW-11424	(16bit) : HMI WiFi 實體位址 (MAC) 2	讀	讀	讀	是
LW-11425	(16bit) : HMI WiFi 實體位址 (MAC) 3	讀	讀	讀	是
LW-11426	(16bit) : HMI WiFi 實體位址 (MAC) 4	讀	讀	讀	是
LW-11427	(16bit) : HMI WiFi 實體位址 (MAC) 5	讀	讀	讀	是
LW-11428	(16bit) : HMI WiFi 網域名稱系統 (DNS) 伺服器 IP 0	讀/寫	讀/控制	讀/控制	是
LW-11429	(16bit) : HMI WiFi 網域名稱系統 (DNS) 伺服器 IP 1	讀/寫	讀/控制	讀/控制	是
LW-11430	(16bit) : HMI WiFi 網域名稱系統 (DNS) 伺服器 IP 2	讀/寫	讀/控制	讀/控制	是
LW-11431	(16bit) : HMI WiFi 網域名稱系統 (DNS) 伺服器 IP 3	讀/寫	讀/控制	讀/控制	是
LW-11432	(16bit) : 自動分配 WiFi IP 位址 (DHCP => 0 : off, 1 : on)	讀/寫	讀/控制	讀/控制	是
LW-12006	(16 words) : WiFi 热點 SSID (啟用以套用)	讀/寫	讀/控制	讀/控制	是
LW-12022	(32 words) : WiFi 热點密碼 (啟用以套用)	讀/寫	讀/控制	讀/控制	是
LW-12703	(16 words) : WiFi 欲連接的 SSID	讀/寫	讀/控制	讀/控制	是
LW-12719	(32 words) : WiFi 欲使用的密碼	讀/寫	讀/控制	讀/控制	是
LW-12751	(16bit) : WiFi 欲使用的安全模式 (0: WPA/WPA2, 1: WEP, 2: 無)	讀/寫	讀/控制	讀/控制	是

LW-12768	(16bit) : WiFi 热點安全模式 (0: 無, 1: WPA-MIX-PSK)	讀/寫	讀/控制	讀/控制	是
LW-12769	(16bit) : WiFi 热點伺服器位址 IP0	讀	讀	讀	是
LW-12770	(16bit) : WiFi 热點伺服器位址 IP1	讀	讀	讀	是
LW-12771	(16bit) : WiFi 热點伺服器位址 IP2	讀/寫	讀/控制	讀/控制	是
LW-12772	(16bit) : WiFi 热點伺服器位址 IP3	讀	讀	讀	是

 Note

- 請使用 ASCII 輸入國碼且需為大寫，設定後需重啟 HMI。國碼將根據各地區的規範，設定 HMI 可連接或搜尋的 WiFi 頻道。
- 訊號強度分為 1: 微弱 (<-70 dBm), 2: 普通 (-60 ~ -70 dBm), 3: 良好 (-50 ~ -60 dBm), 4: 強 (>-50 dBm)

### 22.3.39. OPC UA 伺服器

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-11435	(16bit) : OPC UA 伺服器狀態 (0: 已停止, 1: 已啟用)	讀	讀	讀	是
LW-11436	(16bit) : OPC UA 伺服器錯誤碼 (0: 成功, 1 或其他: 錯誤) *註 1	讀	讀	讀	是
LW-11437	(16bit) : OPC UA 伺服器控制命令 (0: 無, 1: 開始, 2: 停止)	讀/寫	讀/控制	讀/控制	是
LB-12753	清除 OPC UA HDA 資料	讀/寫	讀/控制	讀/控制	是

 Note

- 失敗步驟碼解釋如下：

錯誤碼	錯誤原因
0	無錯誤
1	未知的錯誤
2	憑證無效

### 22.3.40. 郵件

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12053	[事件記錄] 郵件發送失敗 (當狀態為 ON)	讀	讀	讀	是
LB-12054	[備份物件] 郵件發送失敗 (當狀態為 ON)	讀	讀	讀	是

LW-9216	(16bit) : 匯入郵件資料的結果 *註 1	讀	讀	讀	是
LW-11444	(16bit) : 失敗步驟 (郵件) *註 2	讀	讀	讀	是
LW-11445	(16bit) : 錯誤碼 (郵件) *註 3	讀	讀	讀	是

 Note

1. 若匯入成功則輸出值為 1，若匯入失敗 (檔案不存在) 則輸出值為 2。, 若檔案格式錯誤或檔案無法被解密，則輸出值為 4。
2. 失敗步驟碼解釋如下：

錯誤碼	錯誤原因
0	CSMTP_NO_ERROR
100	WSA_STARTUP = Unable to initialise winsock2
101	WSA_VER = Wrong version of the winsock2
102	WSA_SEND = Function send() failed
103	WSA_RECV = Function recv() failed
104	WSA_CONNECT = Function connect failed
105	WSA_GETHOSTBY_NAME_ADDR = Unable to determine remote server
106	WSA_INVALID_SOCKET = Invalid winsock2 socket
107	WSA_HOSTNAME = Function hostname() failed
108	WSA_IOCTLSOCKET = Function ioctlsocket() failed
109	WSA_SELECT
110	BAD_IPV4_ADDR = Improper IPv4 address
200	UNDEF_MSG_HEADER = Undefined message header
201	UNDEF_MAIL_FROM = Undefined mail sender
202	UNDEF SUBJECT = Undefined message subject
203	UNDEF_RECIPIENTS = Undefined at least one recipient
204	UNDEF_RECIPIENT_MAIL = Undefined recipient mail
205	UNDEF_LOGIN = Undefined user login
206	UNDEF_PASSWORD = Undefined user password
207	BAD_LOGIN_PASSWORD = Invalid user login or password
208	BAD_DIGEST_RESPONSE = Server returned a bad digest MD5 response
209	BAD_SERVER_NAME = Unable to determine server name for digest MD5 response
300	COMMAND_MAIL_FROM = Server returned error after sending MAIL FROM
301	COMMAND_EHLO = Server returned error after sending EHLO
302	COMMAND_AUTH_PLAIN = Server returned error after sending AUTH PLAIN
303	COMMAND_AUTH_LOGIN = Server returned error after sending AUTH LOGIN
304	COMMAND_AUTH_CRAMMD5 = Server returned error after sending AUTH CRAM-MD5

305	COMMAND_AUTH_DIGESTMD5 = Server returned error after sending AUTH DIGEST-MD5
306	COMMAND_DIGESTMD5 = Server returned error after sending MD5 DIGEST
307	COMMAND_DATA = Server returned error after sending DATA
308	COMMAND_QUIT = Server returned error after sending QUIT
309	COMMAND_RCPT_TO = Server returned error after sending RCPT TO
310	MSG_BODY_ERROR = Error in message body
400	CONNECTION_CLOSED = Server has closed the connection
401	SERVER_NOT_READY = Server is not ready
402	SERVER_NOT RESPONDING = Server not responding
403	SELECT_TIMEOUT =
404	FILE_NOT_EXIST = File not exist
405	MSG_TOO_BIG = Message is too big
406	BAD_LOGIN_PASS = Bad login or password
407	UNDEF_XYZ_RESPONSE = Undefined xyz SMTP response
408	LACK_OF_MEMORY = Lack of memory
409	TIME_ERROR = time() error
410	RECVBUF_IS_EMPTY = RecvBuf is empty
411	SEND_BUF_IS_EMPTY = SendBuf is empty
412	OUT_OF_MSG_RANGE = Specified line number is out of message size
413	COMMAND_EHLO_STARTTLS = Server returned error after sending STARTTLS
414	SSL_PROBLEM = SSL problem
415	COMMAND_DATABLOCK = Failed to send data block
416	STARTTLS_NOT_SUPPORTED = The STARTTLS command is not supported by the server
417	LOGIN_NOT_SUPPORTED = AUTH LOGIN is not supported by the server

3. 可藉由勾選 [系統參數設定 \ 郵件] 頁籤底下的 [錯誤訊息]，並指定一個啟始字組位址 (長度可自行調整) 來顯示郵件伺服器所回傳的錯誤訊息。

#### 22.3.41. cMT Viewer 與診斷器

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-12656	啟用診斷器 (設定為 ON), 取消 (設定為 OFF)	讀/寫	讀/控制	讀/控制	是
LB-12657	診斷器不須密碼即可登入 (當狀態為 ON)	讀/寫	讀/控制	讀/控制	是
LW-11756	(4 words) : 診斷器密碼	讀/寫	讀/控制	讀/控制	是
LW-11839	(16bit) : 已連接的 clients 數量 (cMT Viewer)	讀	讀	讀	是
LW-11840	(8 words) : client 1 使用者名稱	讀	讀	讀	是

LW-11848	(8 words) : client 2 使用者名稱	讀	讀	讀	是
LW-11856	(8 words) : client 3 使用者名稱	讀	讀	讀	是
LW-11864	(8 words) : client 4 使用者名稱	讀	讀	讀	是
LW-11872	(8 words) : client 5 使用者名稱	讀	讀	讀	是
LW-11880	(8 words) : client 6 使用者名稱	讀	讀	讀	是
LW-11888	(8 words) : client 7 使用者名稱	讀	讀	讀	是
LW-11896	(8 words) : client 8 使用者名稱	讀	讀	讀	是
LW-11904	(8 words) : client 9 使用者名稱	讀	讀	讀	是
LW-11912	(8 words) : client 10 使用者名稱	讀	讀	讀	是
LW-11940	(16bit) : client 1 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11941	(16bit) : client 1 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11942	(16bit) : client 1 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11943	(16bit) : client 1 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11944	(16bit) : client 2 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11945	(16bit) : client 2 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11946	(16bit) : client 2 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11947	(16bit) : client 2 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11948	(16bit) : client 3 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11949	(16bit) : client 3 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11950	(16bit) : client 3 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11951	(16bit) : client 3 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11952	(16bit) : client 4 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11953	(16bit) : client 4 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11954	(16bit) : client 4 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11955	(16bit) : client 4 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11956	(16bit) : client 5 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11957	(16bit) : client 5 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11958	(16bit) : client 5 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11959	(16bit) : client 5 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11960	(16bit) : client 6 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11961	(16bit) : client 6 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11962	(16bit) : client 6 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11963	(16bit) : client 6 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11964	(16bit) : client 7 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11965	(16bit) : client 7 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11966	(16bit) : client 7 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11967	(16bit) : client 7 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是

LW-11968	(16bit) : client 8 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11969	(16bit) : client 8 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11970	(16bit) : client 8 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11971	(16bit) : client 8 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11972	(16bit) : client 9 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11973	(16bit) : client 9 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11974	(16bit) : client 9 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11975	(16bit) : client 9 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11976	(16bit) : client 10 IPO (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11977	(16bit) : client 10 IP1 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11978	(16bit) : client 10 IP2 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
LW-11979	(16bit) : client 10 IP3 (IP 位址 = IPO:IP1:IP2:IP3)	讀	讀	讀	是
PLW-11998	(8 words) : 控制權釋放倒數	讀	N/A	N/A	是

#### 22.3.42. CODESYS 資訊

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LW-11760	(16bit) : CODESYS 韌體狀態 (0:錯誤, 1:開始, 2:停止)	讀	讀	讀	是
LW-11761	(16bit) : CODESYS 應用程式狀態 (0:錯誤, 1:開始, 2:停止)	讀	讀	讀	是
LW-11762	(16bit) : CODESYS 登入狀態 (0:錯誤, 1:登入, 2:登出)	讀	讀	讀	是
LW-12059	(16bit) : CODESYS 韌體版本 (年)	讀	讀	讀	是
LW-12060	(16bit) : CODESYS 韌體版本 (月)	讀	讀	讀	是
LW-12061	(16bit) : CODESYS 韌體版本 (日)	讀	讀	讀	是

#### 22.3.43. 其它功能

位址	描述	讀 / 寫 / 控制			cMT 支援
		本機 HMI	巨集	遠端 HMI	
LB-9000~ LB-9009	重新開機時狀態為 ON	讀/寫	讀/控制	讀/控制	是
LB-9010	資料傳輸寫入指示	讀	讀	讀	是
LB-9011	資料傳輸讀取指示	讀	讀	讀	是
LB-9012	資料傳輸執行指示	讀	讀	讀	是
LB-9016	當任何遠端 HMI 客戶端連接到作為遠端的此 HMI 時, 狀態為 on	讀	讀	讀	是
LB-9017	取消 PLC 控制物件[切換視窗]的[寫回]功能	讀/寫	讀/控制	讀/控制	是
LB-9039	檔案備份動作狀態 (備份中狀態為 ON)	讀	讀	讀	是

LB-9045	memory-map 通訊失敗 (當狀態為 ON)	讀	讀	讀	是
LB-9049	看門狗功能 [開啟 (ON)/取消 (OFF)] (使用 LW-11456 設定看門狗超時) *註 1	讀/寫	讀/控制	讀/控制	是
LB-12356	啟用(設 ON)/停用(設 OFF) 影像串流	讀/寫	讀/控制	讀/控制	是
LB-12357	影像串流狀態 (ON: 啟用 / OFF: 停用)	讀	讀	讀	是
LB-12358	啟用 (設 ON) / 停用 (設 OFF) 於 HMI 上的離線模擬 *註 5	讀/寫	讀/控制	讀/控制	是
LB-12361	操作記錄功能的狀態 (OFF: 關閉, ON: 開啟)	讀	讀	讀	是
LB-12664	Cache Hit (cMT-G03)	讀	讀	讀	是
LB-12668	SECS 授權狀態 (0: 未開通, 1: 已開通)	讀	讀	讀	是
LB-12819	啟動/停止開關狀態 (0: 停止, 1: 啟動) *註 6	讀	讀	讀	是
LW-9006	(16bit) : 連接到本機的遠端 HMI 數目	讀	讀	讀	是
LW-9024	(16bit) : memory link 系統暫存器	讀/寫	讀/控制	讀/控制	是
LW-9032	(8 words) : 備份歷史記錄到 SD 卡, USB 碟的資料夾名稱 *註 3	讀/寫	讀/控制	讀/控制	是
LW-9050	(16bit) : 當前顯示的基本視窗編號	讀	讀	讀	是
PLW-9050	(16bit) : 當前顯示的基本視窗編號 (用於手持平板裝置)	讀	N/A	N/A	是
LW-9134	(16bit) : 當前所使用的語言 *註 2	讀/寫	讀/控制	讀/控制	是
PLW-9134	(16bit) : 當前所使用的語言 (用於手持平板裝置) *註 2	讀/寫	N/A	N/A	是
LW-9900	(16bit) : HMI 工作模式 (0: 正常模式, 1~3: 測試模式 (使用 COM 1~COM 3))	讀/寫	讀/控制	讀/控制	是
LW-10762	(8 words) : 插槽 1 使用者名稱	讀/寫	讀/控制	讀/控制	是
LW-10770	(8 words) : 插槽 2 使用者名稱	讀/寫	讀/控制	讀/控制	是
LW-10778	(8 words) : 插槽 3 使用者名稱	讀/寫	讀/控制	讀/控制	是
LW-10814	(16bit) : 連接至 Weintek HMI (0:無, 1:連接中) *註 4	讀	讀	讀	否
LW-11456	(16bit) : 看門狗超時 (3 ~ 10), 單位：秒 *註 1	讀/寫	讀/控制	讀/控制	是
LW-12245	(32bit) : BACnet/IP server 設備 id	讀/寫	讀/控制	讀/控制	是
LW-12824	(16 bit) : 快閃記憶體健康狀態 (0: 不可用或不支援, 1: 良好, 2: 一般, 3: 差) *註 7	讀	讀	讀	是
LW-12827	(16bit) : BACnet/IP server 埠號	讀/寫	讀/控制	讀/控制	是

 Note

1. 若啟用 LB-9049 看門狗功能，可以設定在 HMI 無法正常運作持續所指定的秒數後自動重啟 HMI。
2. 當物件的文字內容要求表現出多國語言的效果時，除了需使用文字標籤外，也需搭配系統保

留暫存器 LW-9134 的使用。LW-9134 的有效可設定值範圍為 0 ~ 23，此字組位址數值的映射方式將與下載至 HMI 的語言種類有關。當編譯下載的檔案沒有勾選全部語言時，LW-9134 使用方式將有所改變。例如：使用者在文字標籤庫建立了 5 種語言，分別是語言 1 (繁體中文)，語言 2 (簡體中文)，語言 3 (英文)，語言 4 (法文)，語言 5 (日文)。若使用者只下載語言 1、語言 3、語言 5，此時 LW-9134 裡的數值對應的語言種類為 0 → 語言 1 (繁體中文)，1 → 語言 3 (英文)，2 → 語言 5 (日文)。透過項目選單物件搭配 LW-9134 來切換語言範例如下：

-  請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。
- 3.  請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。
- 4. 當 USB Host 接上 Weintek HMI 時，此位址將被設為 1。藉此可用來測試另一台 Weintek HMI 的 USB Client 是否正常。
- 5. 透過此功能可以在 HMI 上將通訊模式切換為離線模擬。當切換至離線模擬模式時，即使 HMI 沒有正確連接設備，Device No Response 訊息也不會彈出，且與設備相關的物件仍可以顯示與操作，但讀取/寫入的值並不會真正被記錄至設備端。
- 6. 支援於 cMT-CTRL01 機型。
- 7. 當快閃記憶體的健康狀態為差時，請立即連繫當地經銷商尋求幫助，以避免 HMI 發生無法開機的狀況。

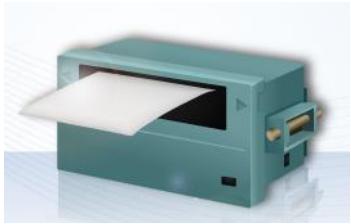
# 23. HMI 支援的印表機類型

本章節說明 HMI 支援的印表機類型，以及設定程序。

23.1. 支援的印表機類型 .....	23-2
23.2. 如何新增一台印表機設備並觸發列印 .....	23-5
23.3. cMT / cMT X 安裝網路印表機.....	23-7
23.4. cMT / cMT X 透過安裝 PPD 使用印表機 .....	23-9
23.5. cMT X 系列透過 IPP 協定使用印表機 .....	23-11
23.6. 確認印表機是否支援 CUPS.....	23-13

## 23.1. 支援的印表機類型

HMI 可支援的印表機驅動大致分為以下幾種。

印表機類型	描述
● <b>SP-M, D, E, F</b>	 <p>此驅動使用 EPSON ESC 串口微型印表機協議。使用串列埠連接，請配合印表機調整通訊參數。[每行點數] 必須正確設定，且不得超過印表機預設值： 100 pixels : 1610 型號之印表機； 220 pixels : 2407, 4004 型號之印表機。 SP-E1610SK (紙張寬度 45mm), SP-E400-4S (紙張寬度 57.5mm) 建議中國地區以外的用戶使用此種 SP 印表機型。 北京迅普: <a href="http://www.siupo.com">http://www.siupo.com</a></p>
● <b>EPSON ESC/P2 系列</b>	 <p>使用串列埠連接，請配合印表機調整通訊參數。此驅動使用 EPSON ESC/P2 印表機通訊協議。 點陣式印表機: LQ-300, LQ-300+, LQ-300K+ (RS-232), LQ-300+II (RS-232) 噴墨式印表機: Stylus Photo 750 雷射印表機: EPL-5800</p>
● <b>HP PCL 系列(USB)</b>	 <p>使用 USB 埠連接，可支援 HP 印表機的 PCL5 協議或 PostScript3 的印表機語言。 因 PCL 印表機語言的向下支援特性，支援 PCL5 或以上的印表機皆可以支援 PCL5 協議。</p>

**● Axiohm A630**

法國愛克勝微型印表機，使用串列埠連接，請配合印表機調整通訊參數。

**● SPRT**

使用串列埠連接，請配合印表機調整通訊參數。[每行點數] 必須正確設定，且不得超過印表機預設值“100”。

支援型號:

SP-DN40SH: 點陣式印表機

SP-RMDIII40SH: 熱感式印表機

**● EPSON TM-L90**

使用串列埠連接，請配合印表機調整通訊參數。[每行點數] 必須正確設定，且不得超過印表機預設值“576”。

**● EPSON TM-T70**

使用串列埠連接，請配合印表機調整通訊參數。[每行點數] 必須正確設定，且不得超過印表機預設值“576”。可選擇紙張切割模式，分為 [不切] 或 [半切]。

**● BRIGHTEK WH-A19**

支援型號：

A92R10-00E72A: 尾碼為 72 的是 16 位印表機，A 表示寬電壓 5~9V，此款與 A6 16 點陣相同。

- **BRIGHTEK WH-E19**

使用串列埠連接，請配合印表機調整通訊參數。



- **BRIGHTEK WH-22 (炜煌)**

支援型號：

E22R10-00E725: 與 A7 16 點陣相同，A7 型號為  
A72R90-31E72A

E221R90-00E11740GA: 此印表機為 485 介面，需要使  
用 232 轉 485 模組。



- **BRIGHTEK WH-C1/C2**

使用串列埠連接，請配合印表機調整通訊參數。可  
選擇紙張切割模式，分為 [不切]、[半切] 或 [全  
切]。



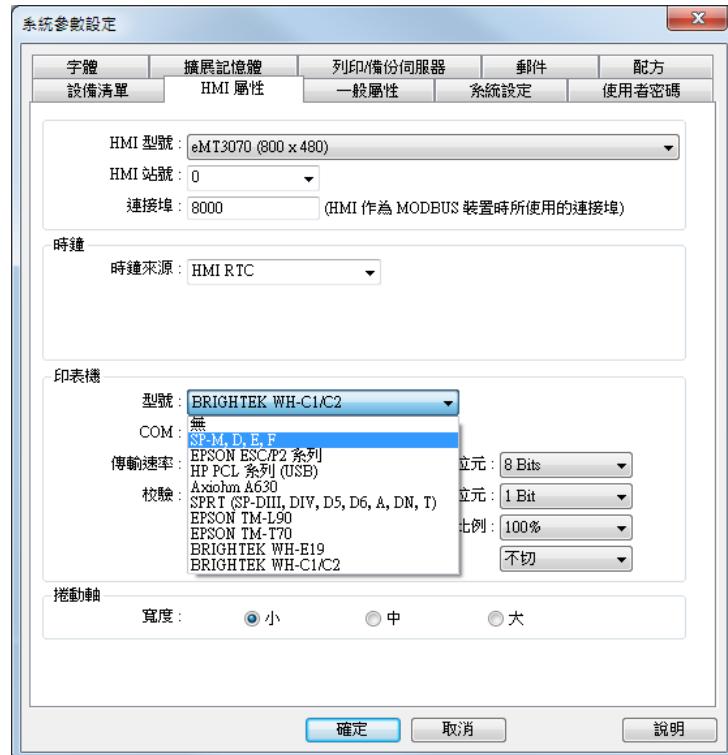
- **遠端列印伺服器**

HMI 可透過乙太網路連接在 PC 上的印表機，並使  
用 EasyPrinter 來執行列印動作。由於 EasyPrinter  
在 MS Windows 系統下運行，因此支援市面上大部  
分的印表機。



## 23.2. 如何新增一台印表機設備並觸發列印

1. 新增印表機設備。
2. 從 [系統參數設定] » [HMI 屬性] 頁籤選擇欲連接的印表機型號，並正確設定相關參數。



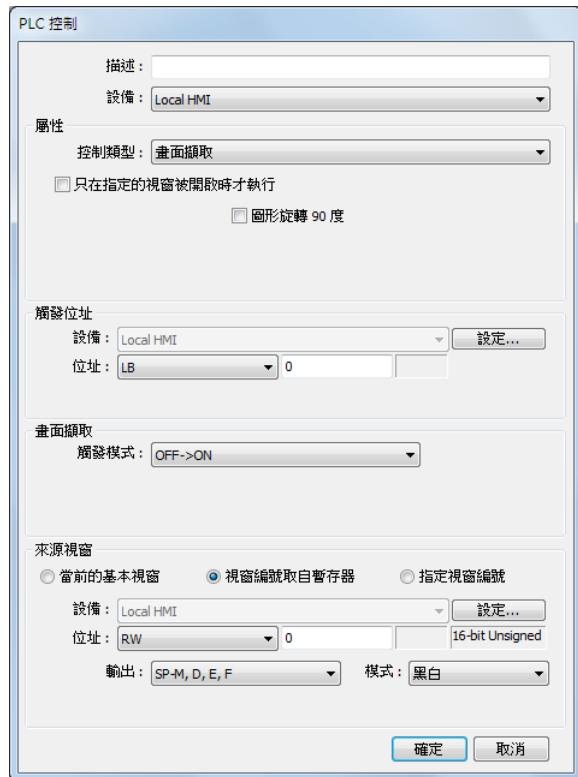
3. 若欲連接至遠端列印伺服器，則需啟用 [系統參數設定] » [列印/備份伺服器] 頁籤，並正確設定相關參數。



4. 觸發列印。
5. 使用者可透過“功能鍵”物件來直接觸發列印功能。



6. 或者，使用“PLC 控制”物件的【畫面擷取】，藉由預先定義好的位元位址來觸發列印功能。



### 23.3. cMT / cMT X 安裝網路印表機

除了上述的印表機以外，cMT / cMT X 亦可透過系統設定的 [安裝網路印表機] 功能使用更多印表機。

- 點選  開啟系統設定。登入 [設定] 頁後，找到 [印表機] » [安裝網路印表機]。

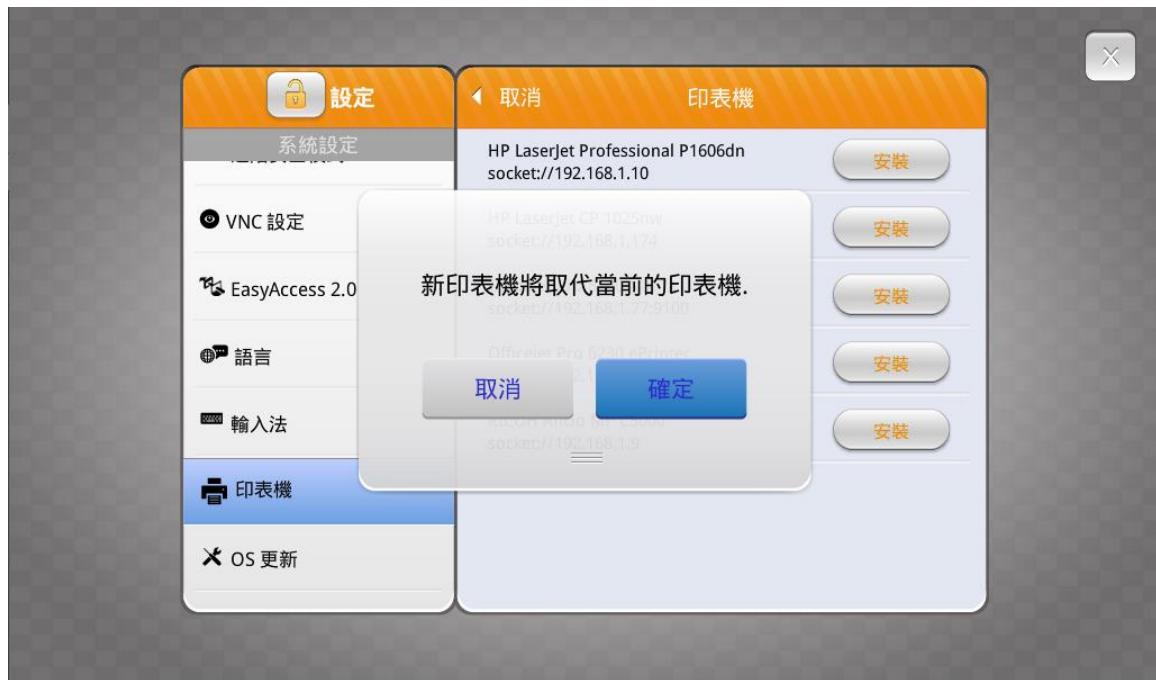


## HMI 支援的印表機類型

- HMI 會花數秒鐘搜尋網域內的網路型印表機。



- 選擇欲安裝的網路型印表機，點選 [安裝] 進行驅動的安裝。由於一次只能安裝一台網路型印表機，因此新的印表機的驅動將會取代舊的印表機的驅動。



- 安裝完成後，即可使用該網路型印表機。

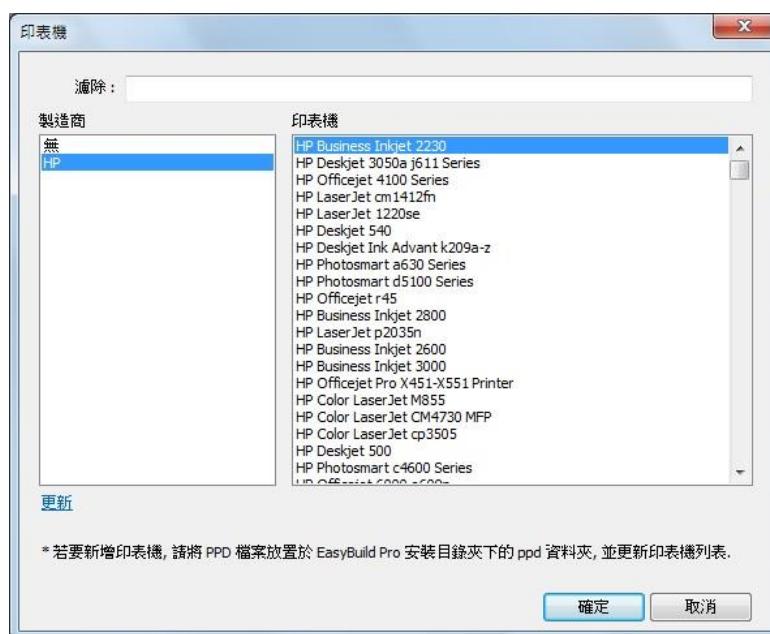
## 23.4. cMT / cMT X 透過安裝 PPD 使用印表機

除了上述的方式以外，cMT / cMT X 亦可透過安裝 PPD 功能使用更多印表機。

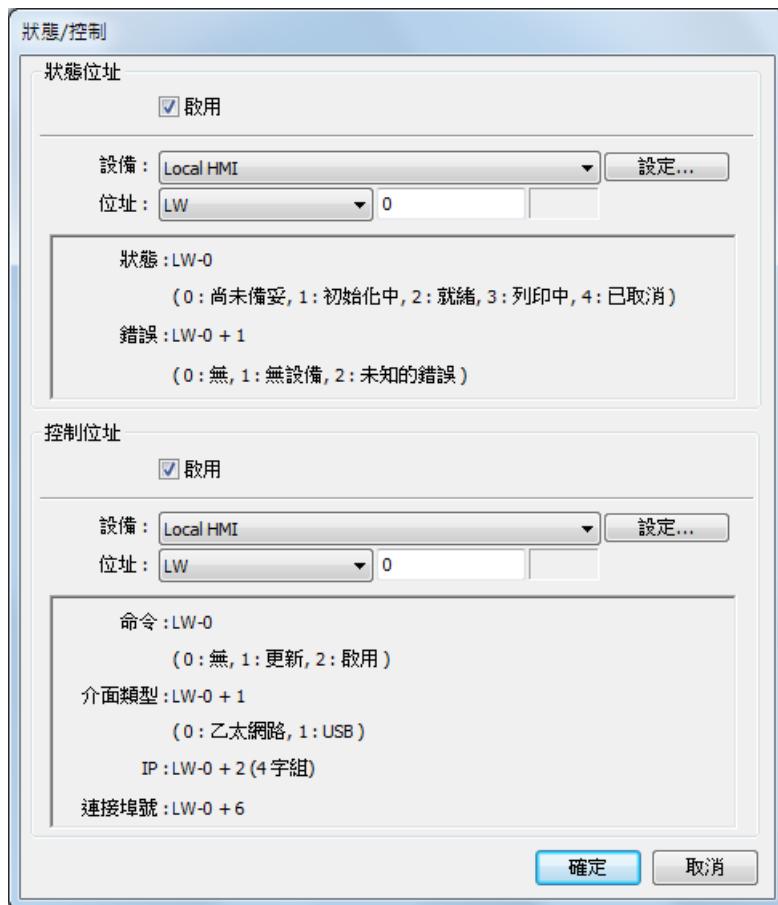
- 開啟 [系統參數設定] » [HMI 屬性] 頁籤設定。



- 選擇欲連接的印表機型號，若要新增印表機，請將 PPD 檔案放置於 EasyBuilder Pro 安裝目錄下的 ppd 資料夾，並點選 [更新] 即可。

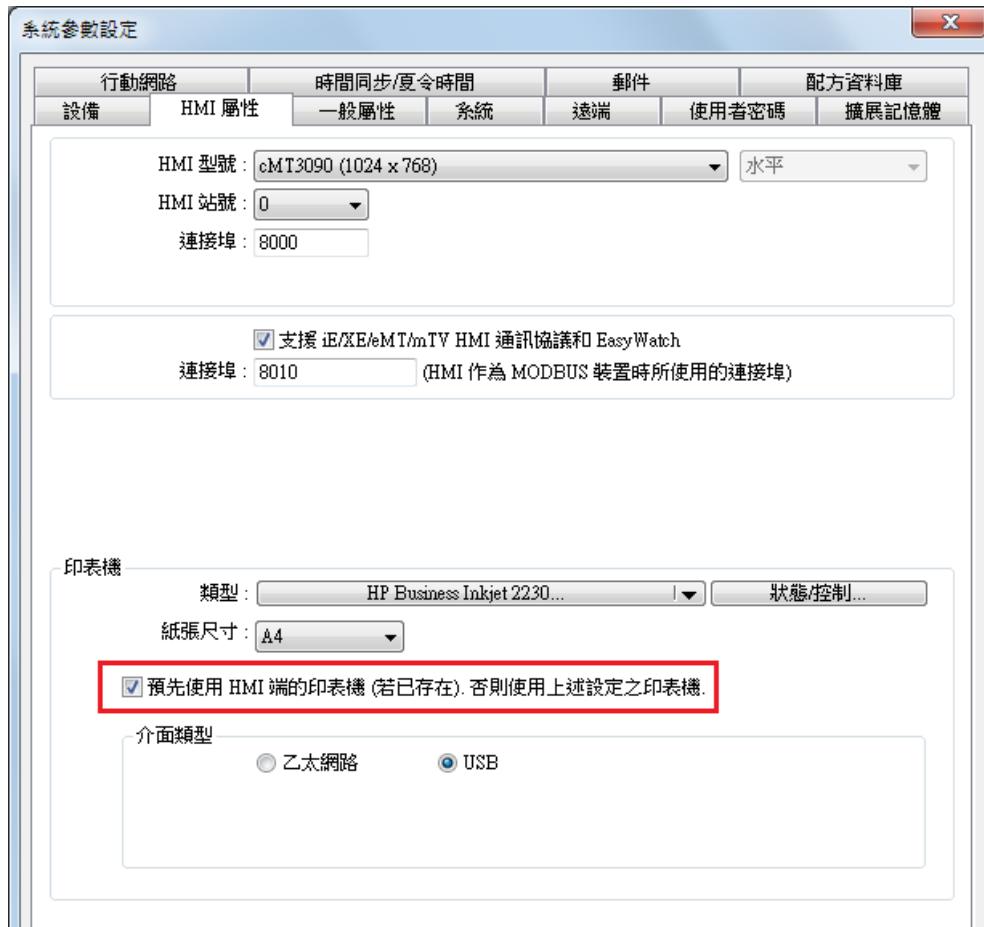


3. 可藉由啟用 [狀態位址] 來監看印表機狀態，或啟用 [控制位址] 來線上更新印表機的連線參數。



### Note

- 此功能不支援 cMT-SVR 及 cMT Gateway 系列機型。
- 因 cMT / cMT X 與 HP 印表機有較好的相容性，建議使用 HP 印表機，並請以實際測試結果為主。
- 使用 cMT / cMT X 系列 HMI 測試印表機時，請優先以 [安裝網路印表機] 方式安裝驅動。如無法使用，再使用 PPD 手動安裝驅動。
- 當 HMI 端已安裝網路印表機驅動，若欲使用工程檔案內設置的印表機，請勿勾選 [預先使用 HMI 端已安裝的印表機] 選項。並於 HMI 端使用控制位址下命令 1: 更新，即可置換印表機驅動來源為工程檔案內的設定。



### 23.5. cMT X 系列透過 IPP 協定使用印表機

cMT X HMI 支援使用 IPP 協議列印，前提是印表機也須支援 IPP 協議。關於 IPP 協議，請參考連結：[https://en.wikipedia.org/wiki/Internet\\_Printing\\_Protocol](https://en.wikipedia.org/wiki/Internet_Printing_Protocol)。

通常於印表機的網路設定可以確認是否支援 IPP 協議。以 HP 印表機為例，需要先啟用網際網路列印通訊協定(IPP)或是 Bonjour，如下圖所示。

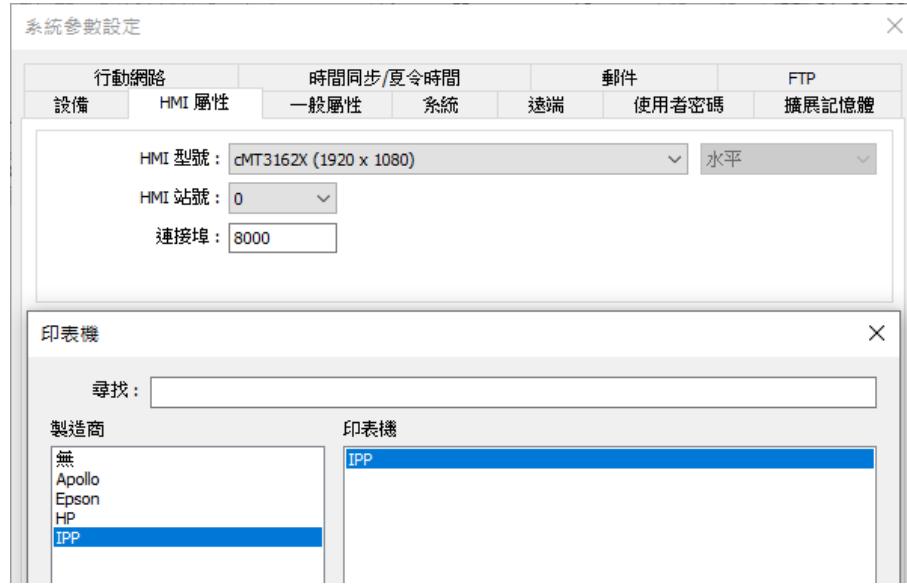
The screenshot shows the 'Networking' section of the HP Officejet Pro 6230 ePrinter's web interface. Under 'Network' settings, the 'IPP' (Internet Printing Protocol) section is selected. It includes options for enabling or disabling IPP and setting the printer location. A status message indicates that IPP is enabled.

The screenshot shows the 'Networking' tab of the HP LaserJet Professional P1606dn's web interface. Under the 'Bonjour' section, the 'Enable Bonjour' checkbox is checked. The printer's Bonjour name is listed as 'HP LaserJet Professional P1606dn' and its domain name as 'NP1B7C40C.local.'

啟用之後透過系統設定的安裝網路印表機列表即可看到能夠安裝的印表機，後方(IPP)代表使用此協議通訊。

The screenshot shows the 'EasyAccess 2.0' software's printer list. It lists three printers: 'HP LaserJet Professional P1606dn socket://192.168.1.10 (IPP)', 'RICOH MP C3504ex socket://192.168.1.9 (IPP)', and 'HP LaserJet CP 1025nw socket://192.168.2.247 (IPP)'. Each entry has an 'Install' button next to it.

在 EasyBuilder Pro 上的印表機設定選擇 IPP 即可正常列印。部分 cMT X HMI 需要透過手動加入 IPP 協議，如果有此需求，請洽詢 Weintek 經銷商。



## 23.6. 確認印表機是否支援 CUPS

因為 cMT/cMT X HMI 內部是透過 CUPS 執行列印動作，故印表機必須支援 CUPS，cMT / cMT X 才可支援。關於 CUPS，請參考連結：<https://en.wikipedia.org/wiki/CUPS>。

是否支援 CUPS 確認方法：

- 向印表機製造商詢問。
- 若用戶熟悉 Linux 操作系統，可在 (Linux) PC 環境下透過以下方式測試，並下載 PPD 檔：
  1. Linux 安裝 CUPS 或確認 CUPS 已經執行，透過瀏覽器輸入 <http://localhost:631> 。
  2. 在 Home 頁面選擇 [Adding Printers and Classes] 。



The screenshot shows the CUPS 1.4.2 documentation homepage. At the top, there is a navigation bar with links for Home, Administration, Classes, Online Help, Jobs, and Printers. Below the navigation bar, the title "CUPS 1.4.2" is displayed in a large, bold font. A subtitle below it states: "CUPS is the standards-based, open source printing system developed by Apple Inc. for Mac OS® X and other UNIX®-like operating systems." To the right of the title is the CUPS logo, which consists of a stylized letter 'C' with the text "UNIX PRINTING SYSTEM" inside it. The main content area is divided into three columns: "CUPS for Users", "CUPS for Administrators", and "CUPS for Developers". The "CUPS for Administrators" column contains a yellow-highlighted link titled "Adding Printers and Classes".

CUPS for Users	CUPS for Administrators	CUPS for Developers
Overview of CUPS	Adding Printers and Classes	Introduction to CUPS Programming
Command-Line Printing and Options	Managing Operation Policies	CUPS API
What's New in CUPS 1.4	Printer Accounting Basics	Filter and Backend Programming
User Forum	Server Security	HTTP and IPP APIs
	Using Kerberos Authentication	PPD API
	Using Network Printers	Raster API
	cupsd.conf Reference	PPD Compiler Driver Information File Reference
	Find Printer Drivers	Developer Forum

3. 在 Printers 選擇 [Add Printer]。

## Printers

[Add Printer](#) [Find New Printers](#) [Manage Printers](#)

## Classes

[Add Class](#) [Manage Classes](#)

4. 跳出安裝界面，可以先透過網路驗證，選擇 [AppSocket] 界面。

## Add Printer

### Add Printer

**Local Printers:**  CUPS-BRF (Virtual Braille BRF Printer)

Serial Port #1

Serial Port #5

USB Serial Port #1

LPT #1

HP Printer (HPLIP)

HP Fax (HPLIP)

**Discovered Network Printers:**  HP LaserJet CP 1025nw (Hewlett-Packard HP LaserJet CP 1025nw)

HP LaserJet Professional P1606dn (Hewlett-Packard HP LaserJet)

HP LaserJet Professional P1606dn (HP LaserJet Professional P1606dn)

RICOH MP C3504ex (RICOH MP C3504ex)

HP LaserJet CP 1025nw (HP LaserJet CP 1025nw)

Hewlett-Packard HP LaserJet CP 1025nw (driverless) (Hewlett-Packard HP LaserJet CP 1025nw)

Hewlett-Packard HP LaserJet Professional P1606dn (driverless) (Hewlett-Packard HP LaserJet Professional P1606dn)

**Other Network Printers:**  Internet Printing Protocol (http)

Internet Printing Protocol (https)

Internet Printing Protocol (ipp)

Internet Printing Protocol (ipps)

Backend Error Handler

AppSocket/HP JetDirect

LPD/LPR Host or Printer

Windows Printer via SAMBA

5. 輸入當下印表機 IP 位置，例如：socket://10.3.9.55:9100。

## Add Printer

**Connection:**

**Examples:**

`http://hostname:631/ipp/`  
 `http://hostname:631/ipp/port1`

`ipp://hostname/ipp/`  
`ipp://hostname/ipp/port1`

`lpd://hostname/queue`

`socket://hostname`  
`socket://hostname:9100`

See "Network Printers" for the correct URI to use.

[Continue](#)

6. 輸入印表機名稱。

## Add Printer

7. 在 [Make] 選擇印表機的廠牌，接著在 [Model] 選擇型號。若找不到您的型號，需與廠商確認可用哪個型號的 PPD 替代。

## Add Printer

**Name:** GK420D  
**Description:** Ethernet-10.3.9.55  
**Location:** LAB  
**Connection:** socket://10.3.9.55:9100  
**Sharing:** Do Not Share This Printer  
**Make:**

**Name:** CK420D

**Description:** Ethernet-10.3.9.55

**Location:** LAB

**Connection:** socket://10.3.9.55:9100

**Sharing:** Do Not Share This Printer

**Make:** Zebra [Select Another Make/Manufacturer](#)

**Model:** [Zebra CPCL Label Printer \(en\)](#)  
[Zebra EPL1 Label Printer \(en\)](#)  
[Zebra EPL2 Label Printer \(en\)](#)  
**Zebra ZPL Label Printer (en)**

**Create a PPD File:** [Choose File](#) no file selected

[Add Printer](#)

8. 添加完成之後在 Linux 下透過 `/etc/cups/ppd` 找到剛剛安裝過程中印表機名稱的 PPD 檔。成功取得 PPD 檔後，匯入到 EasyBuilder Pro (參考 CH23.4)，並重新下載工程檔案至 HMI 測試。
9. 若仍無法驅動印表機正確列印，請洽詢 Weintek 經銷商。

# 24. Recipe Editor

本章節說明如何使用 Recipe Editor。

24.1. 概要 .....	24-2
24.2. 配方資料 / 擴展記憶體編輯器設定.....	24-2
24.3. 配方資料庫記錄的設定 .....	24-4

## 24.1. 概要

Recipe Editor 可用來建立 HMI 所使用的配方資料檔案，也可開啟及編輯現有的配方資料檔案。此外，EasyBuilder Pro 提供另一個編輯配方資料檔案的工具 - 配方資料庫記錄，此功能需先在 EasyBuilder Pro [資料/歷史] » [配方資料庫] 定義配方，再使用 [配方檢視物件] 來顯示配方內容，以下將介紹此二種編輯器的使用方法。

## 24.2. 配方資料 / 擴展記憶體編輯器設定

1. 從 Utility Manager 點擊 [配方資料/擴展記憶體編輯器]。
2. 要新增新的 .rcp 或是 .emi 檔案，請點選 [檔案] » [開啟檔案]。
3. 設定存取位址範圍與資料格式。



設定	描述
存取範圍	填入起始位址和結束位址，以字組為單位。
選擇資料格式	定義完成的資料格式可儲存，並於下次需要時載入。 範本將存成 “dataEX(fmt” 檔案並存放在 EasyBuilder Pro 安裝目錄下。
資料格式	可在資料格式區域中編輯新的資料格式。

4. 點擊 [新增] 後，彈出資料類型編輯視窗如下，請在 [描述] 欄位輸入資料型態的名稱，並選擇資料格式。若選擇 [String]，需輸入字組長度及設定格式類型為 [ASCII] 或 [Unicode]。



5. 資料格式定義完成後，點選 [確定] 即可編輯配方資料。

ID	ADDRESS	Product No.	Name	Store No.	Category
0	0	0	shampoo	9	4
1	13	1	knife	1	5
2	26	2	chair	3	2
3	39	3	coffee	3	3
4	52	4	pencil	6	5
5	65	5	muffin	6	3
6	78	6	donut	5	3
7	91	7	DVD	9	6
8	104	8	postcard	4	5
9	117	9	maps	5	5
10	130	10	camera	2	1

此範例的資料格式長度為 13 個字組，因此可將每 13 個字組長度視為一組配方來使用。如上，第一組的 “Product no.” 為 address 0，“Name” 位址為 address 1 ~ 10，“Store No.” 為 address 11，“Category” 為 address 12;

第二組的 “Product no.” 為 address 13，“Name” 位址為 address 14 ~ 23，“Store No.” 為 address 24，“Category” 為 address 25...依此類推。

### Note

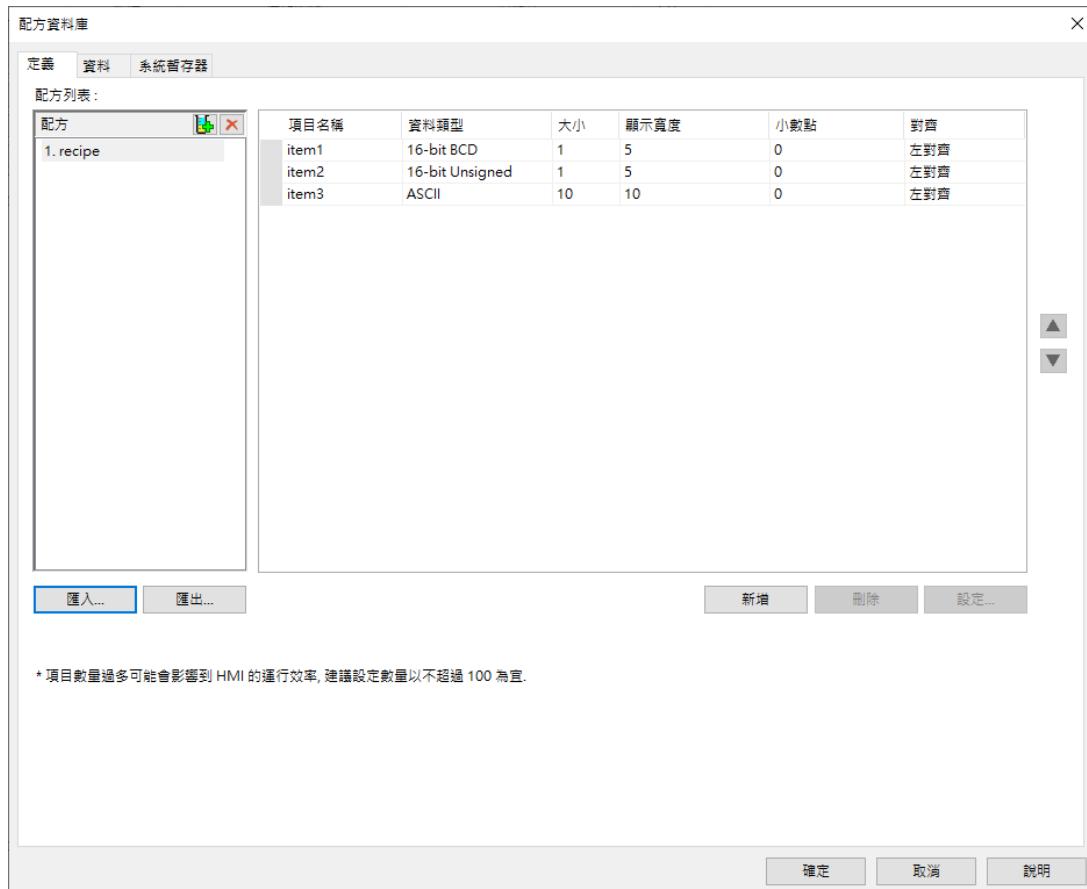
- 編輯完成的配方資料可儲存成 .rcp、.emi 或 .csv 檔案格式。儲存完成的 .rcp 檔案可藉由 Utility Manager 或外部裝置 (USB 碟或 SD 卡) 下載到 HMI，而 .emi 則可直接存放至外部裝置並插入至 HMI 讀取，即為擴展位址 EM。

## 24.3. 配方資料庫記錄的設定

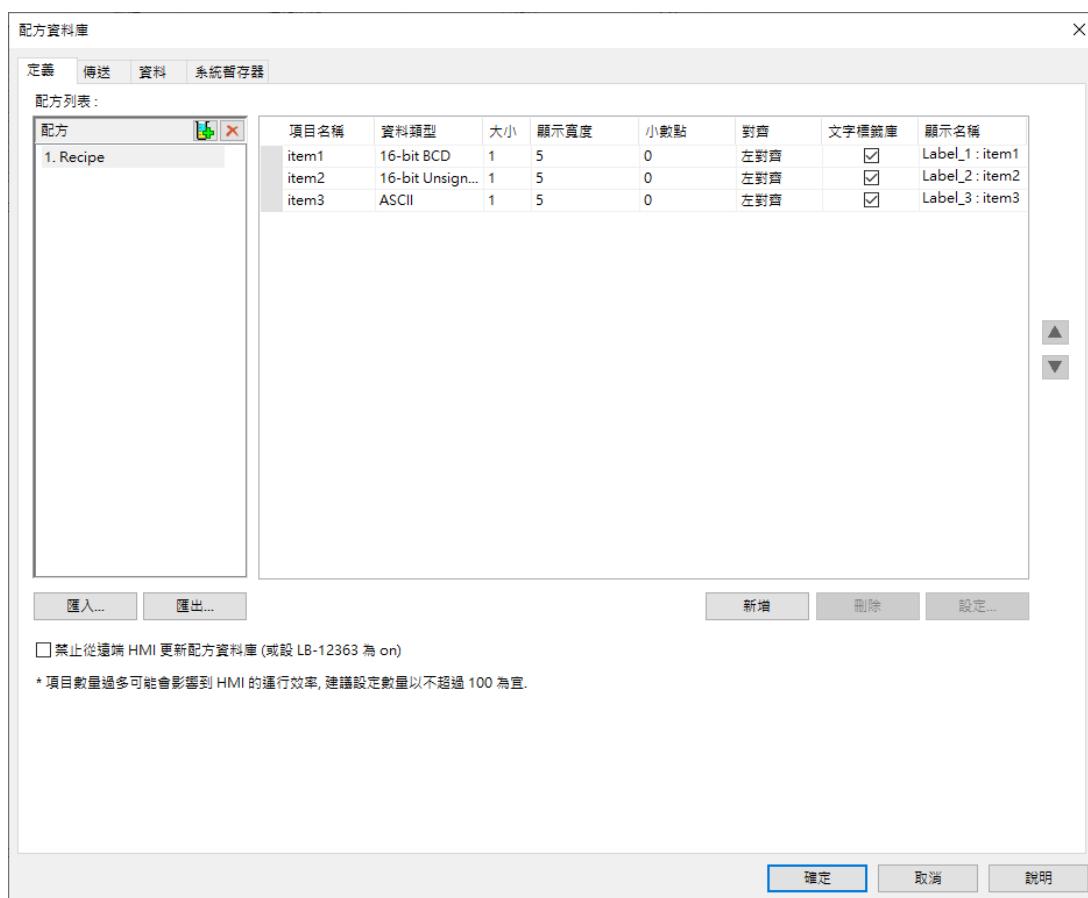
在 EasyBuilder Pro 的 [資料/歷史]，點選 [配方資料庫]，即可設定配方的項目內容、操作模式、與配方資料。

### 24.3.1. 定義

[定義] 設定頁用來設定配方資料庫的相關內容，包括配方名稱、項目名稱與資料格式。  
eMT、iE、XE、mTV 系列：



## cMT / cMT X 系列:



設定	描述
配方列表	新增配方或刪除配方，最多可達 100 個配方。
新增	新增一筆新項目，最多可達 1000 筆配方項目。
設定	設定所選擇配方項目的相關屬性，請見下方說明。
刪除	選擇要刪除的項目，即可進行刪除。
匯入	匯入配方資料定義檔。
匯出	匯出配方資料定義檔，匯出的檔案並不會包含配方資料庫記錄中的內容。

## Recipe Editor

按下 [設定] 後：

cMT / cMT X 系列

eMT、iE、XE、mTV 系列



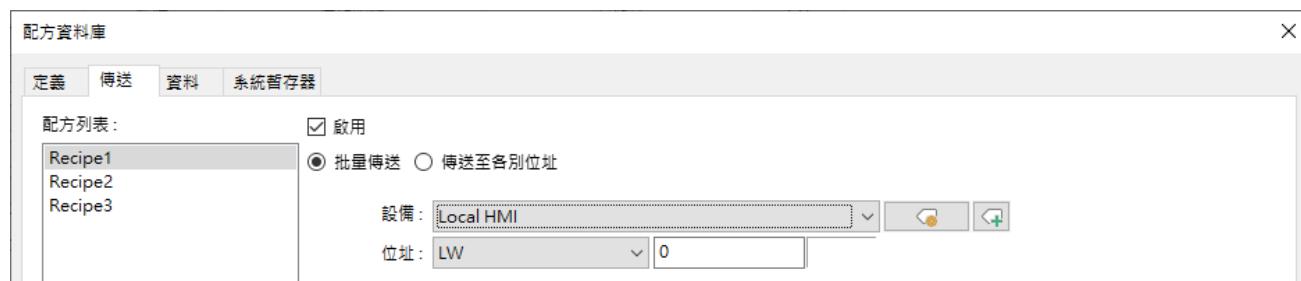
設定	描述
名稱	輸入配方項目名稱。
顯示類型	設定項目的資料類型。
項目大小 (字組)	設定項目大小，單位為字組。
顯示寬度 (字)	設定項目的顯示字數。
小數點後位數	設定數據顯示時的小數點後位數。
對齊	設定顯示時的對齊方式，可選擇左對齊、置中對齊、右對齊。
文字標籤	cMT/cMT X 系列支援設定顯示名稱。顯示名稱預設與該項目名稱相同，也可以設定使用文字標籤庫。 文字標籤庫使用方法請參考第 16 章。

### 24.3.2. 傳送

透過 [傳送] 頁內的設定，可更容易地在配方資料庫記錄與指定位址之間傳送資料。

#### 批量傳送

傳送目標為指定位址開始的數個連續位址。



使用此模式時，須注意目標位址的連續位址的格式、地址間隔等相關設定必須與該配方項目符合。

例如：若配方項目如下圖，使用批量傳送到指定位址 LW-0 時，將得到如下表格的結果。

項目名稱	資料類型	大小	顯示寬度	小數點	對齊	文字標籤庫	顯示名稱
item1	16-bit Unsigned	1	5	0	左對齊	<input checked="" type="checkbox"/>	Label_1 : item1
item2	16-bit Signed	1	5	0	左對齊	<input checked="" type="checkbox"/>	Label_2 : item2
item3	32-bit Float	2	5	0	左對齊	<input checked="" type="checkbox"/>	Label_3 : item3

配方項目名稱	傳送目標地址	傳送目標地址格式
item1	LW-0	16-bit Unsigned
item2	LW-1	32-bit Signed
item3	LW-3	32-bit Float

### 傳送至各別位址

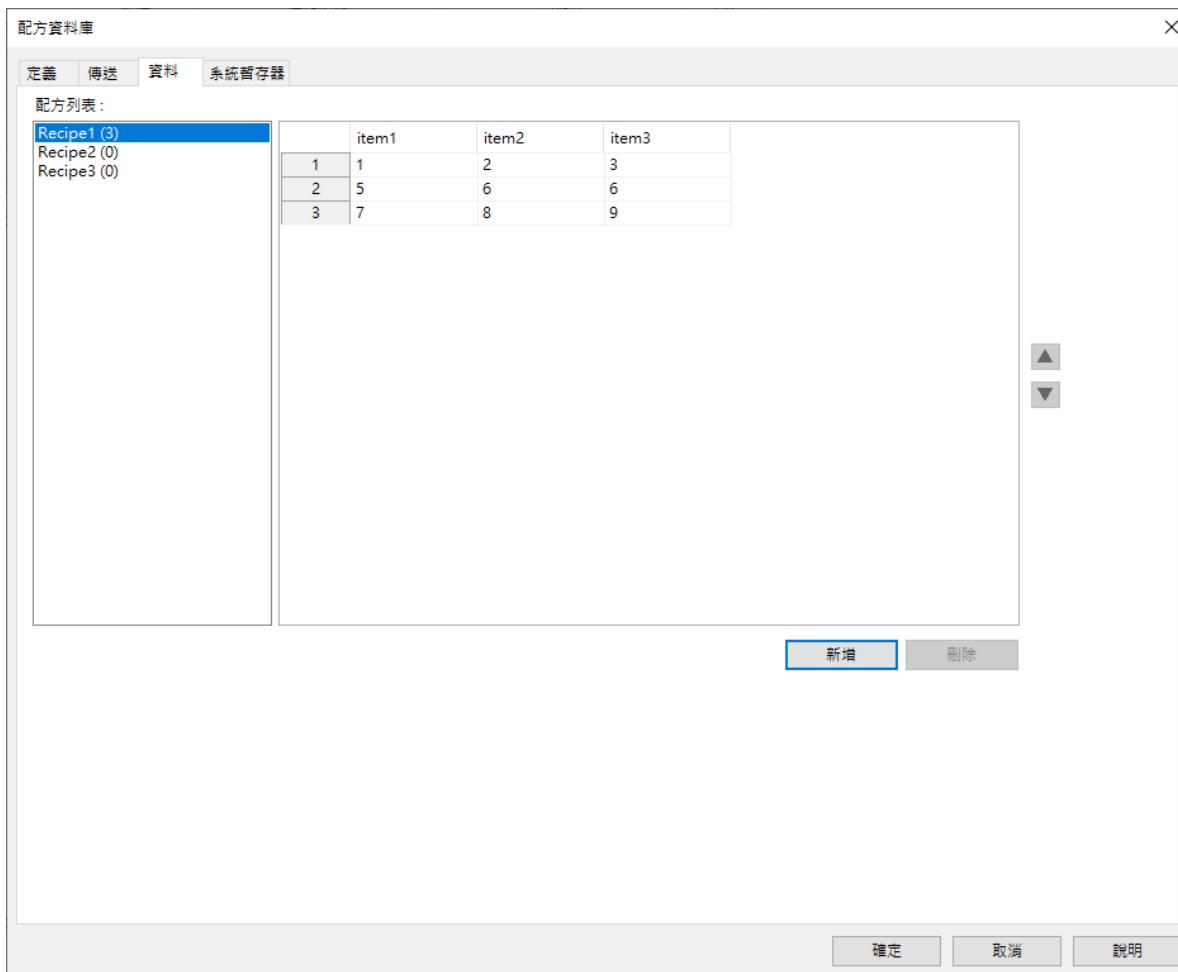
各個項目分別傳送至指定的位址。

項目名稱	資料類型	位址
item1	16-bit Unsigned	[MODBUS TCP/IP] 4x-1
item2	16-bit Signed	[MODBUS TCP/IP] 4x-100
item3	32-bit Float	[MODBUS TCP/IP] 4x-200

使用此模式時，仍須注意對應位址的資料格式是否與配方項目符合。以上圖為例：須注意 4x-1 是否為 16-bit Signed 格式，4x-100 是否為 32-bit Signed 格式，依此類推。

### 24.3.3. 資料

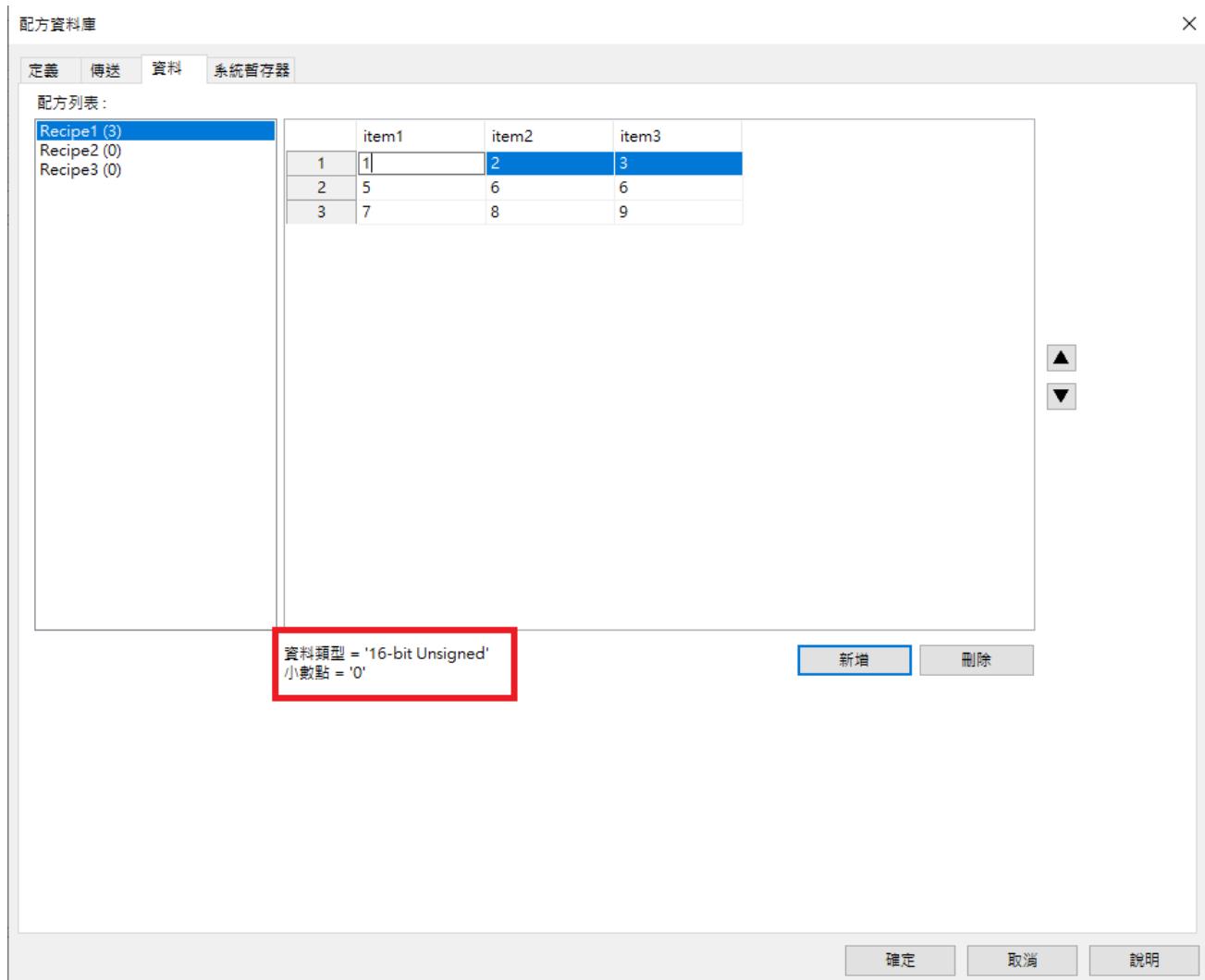
編輯配方資料的內容。使用者需於 [定義] 分頁中設定好配方格式後，再利用 [資料] 分頁來新增或編輯配方資料。



如上圖所示，左邊有配方列表，以及右邊上方的配方項目，名稱皆是在【定義】頁籤設定的配方而來的，接下來可分別針對配方名稱的格式來定義配方的資料。

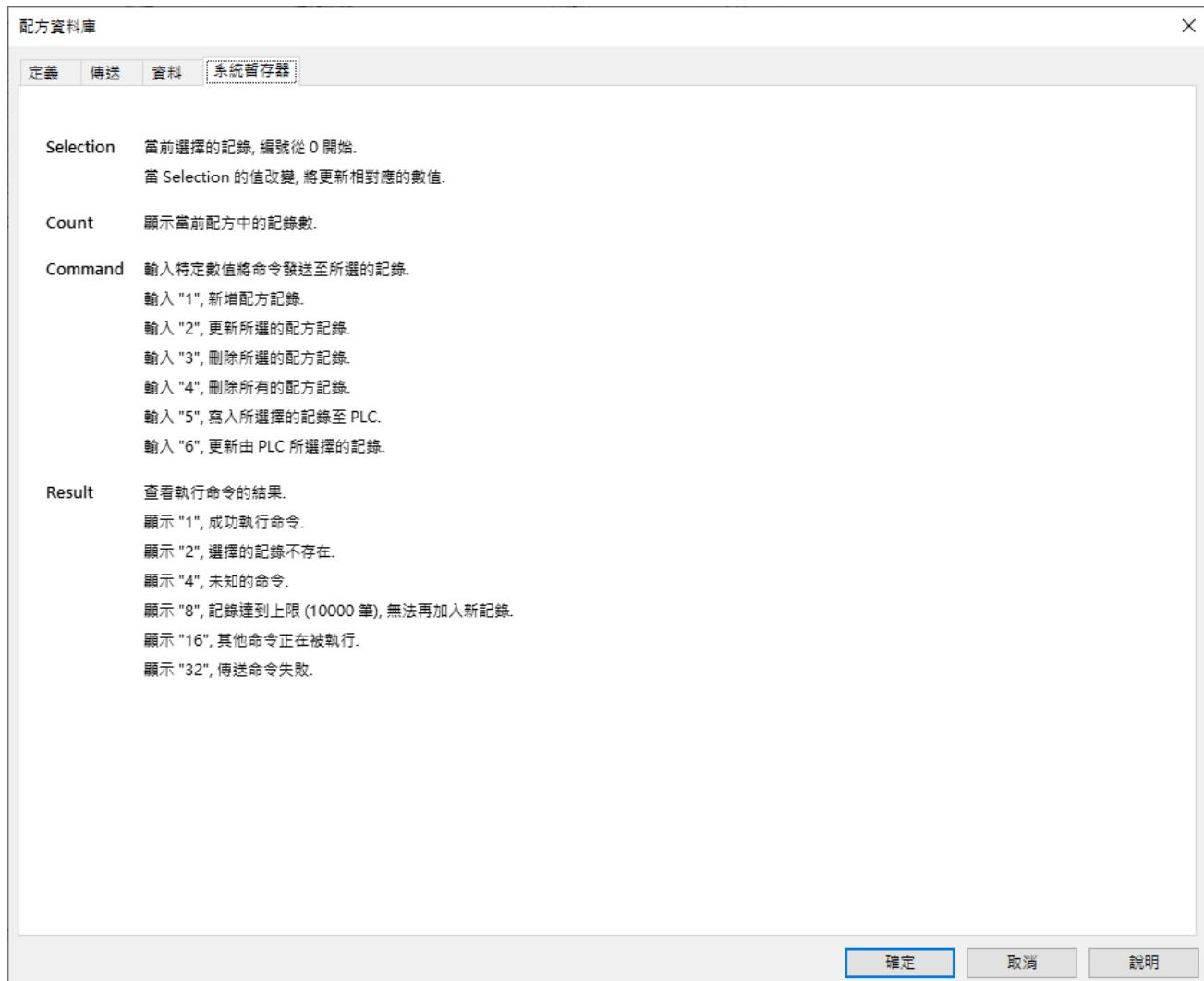
設定	描述
配方	配方列表，此內容可顯示【定義】頁籤中的配方，其括號內的數字可顯示出有幾筆配方資料。
新增	依照各項目的格式定義，可開始編輯配方內容。
刪除	可刪除所編輯配方的內容。
上下鍵	利用上下鍵將所選配方項目的資料往上或往下移動。

依照各項目的格式定義，按下【新增】按鈕後，可編輯配方項目的內容，每一個項目的格式在點選欄位後會顯示在下方，使用者可依項目的格式填入所需的內容，並按下【確定】來完成儲存。



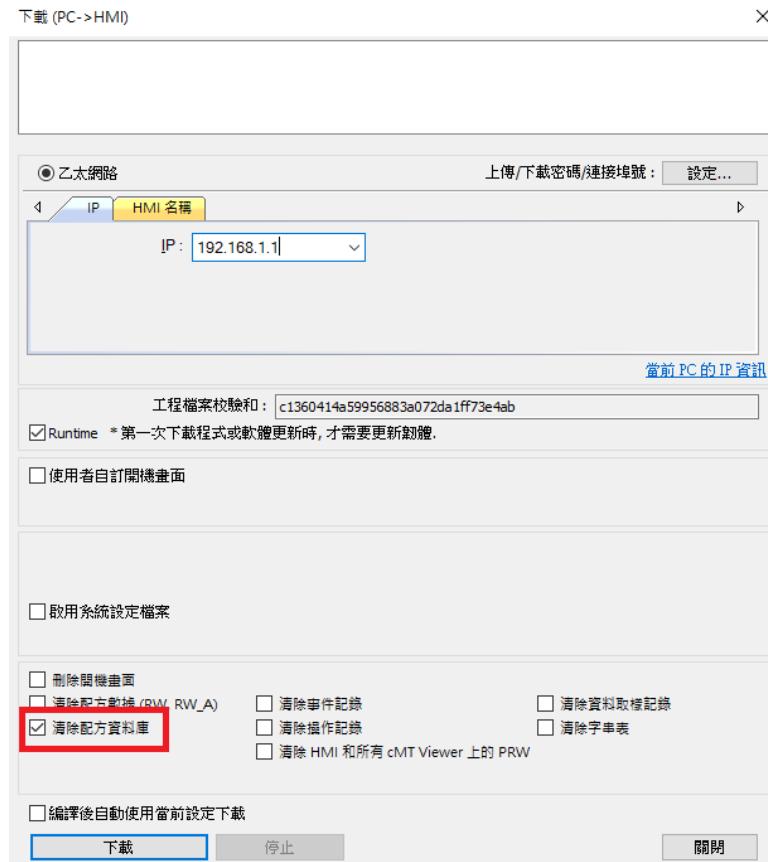
#### 24.3.4. 系統暫存器

在 [系統暫存器] 分頁，可以查看操作配方資料庫的控制位址及說明。



### Note

- 一個工程檔案最多可建立 100 個配方資料庫，一個配方資料庫最多可擁有 1000 筆配方項目。
- 每一個配方最多可以新增 10000 筆記錄。
- 一個配方資料庫最多可使用 2000 個字組的資料，若超過此限制，將無法成功編譯。
- 配方資料庫名稱及配方項目僅支援英文字母或數字。
- 配方資料在編譯後會儲存於 .exob 檔案內並且被下載到 HMI，此配方資料無法共用在其他工程檔案。若是在下載工程檔案後，再次使用配方資料庫記錄修改配方內容且需下載到 HMI，務必勾選 [清除配方資料庫]，若無勾選，則 HMI 還是會保持原本舊的配方資料庫內容。



 請點選此圖示觀看教學影片，請先確定已連上網路線。

# 25. EasyConverter

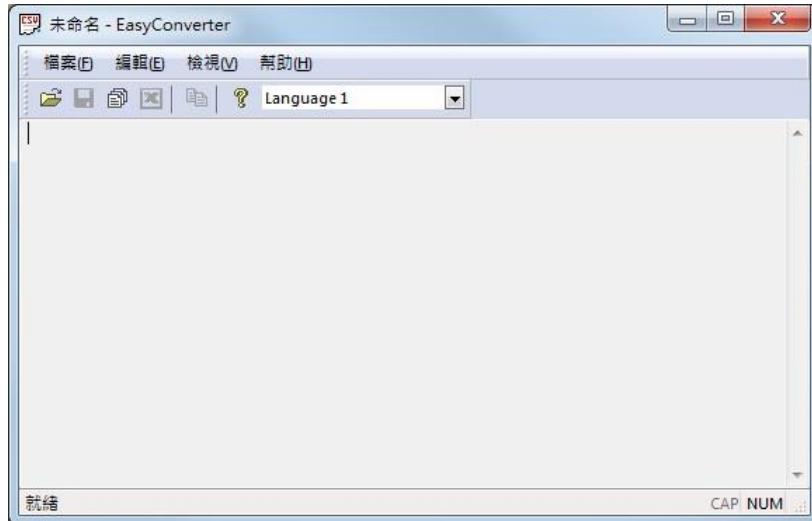
本章節說明如何使用 EasyConverter 與相關設定。

25.1.	概要 .....	25-2
25.2.	將資料取樣記錄檔案輸出至 Excel .....	25-2
25.3.	將事件記錄檔案輸出至 Excel .....	25-4
25.4.	將操作記錄檔案輸出至 Excel .....	25-6
25.5.	多檔案轉換 .....	25-8
25.6.	比例轉換功能 .....	25-9
25.7.	批次檔規則 .....	25-10
25.8.	檢查歷史記錄檔案完整性 .....	25-13

## 25.1. 概要

EasyConverter 程式可讀取由 HMI 撷取的資料取樣記錄、事件記錄或操作記錄檔案，並轉換成 Excel 或 PDF 格式的檔案。

- 從 Utility Manager 點擊 [資料轉換] » [EasyConverter]。
- 從 EasyBuilder Pro 工具選單下點擊 [事件記錄/資料取樣記錄轉換程式]。



▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 25.2. 將資料取樣記錄檔案輸出至 Excel

1. 當開啟的資料取樣記錄檔案格式為 DB，且檔案內含有一天以上的資料時，將可選擇欲檢視的日期範圍。(若開啟的檔案格式為 DTL，則略過此步驟)



2. 此時將彈出設定視窗如下，請依照需求作相關設定。



3. 按下 [確定] 後，資料取樣記錄將顯示如下，再按下 [匯出至 Excel] 即可轉換成 Excel 格式。

```

Date", "Time", "Millisecond", "temperature", "humidity"
"2014/4/5", "04:03:30", "46", "0", "0"
"2014/4/5", "04:03:33", "459", "0", "0"
"2014/4/5", "04:03:36", "456", "0", "0"
"2014/4/5", "04:03:39", "456", "0", "0"
"2014/4/5", "04:03:42", "457", "0", "0"
"2014/4/5", "04:03:45", "457", "0", "0"
"2014/4/5", "04:03:48", "457", "10", "0"
"2014/4/5", "04:03:51", "458", "10", "23"
"2014/4/5", "04:03:54", "457", "10", "23"
"2014/4/5", "04:03:57", "458", "42", "23"
"2014/4/5", "04:04:00", "456", "14", "67"
"2014/4/5", "04:04:03", "457", "14", "55"
"2014/4/5", "04:04:06", "459", "14", "55"
"2014/4/7", "04:04:09", "264", "14", "55"
"2014/4/7", "04:04:12", "209", "23", "55"
"2014/4/7", "04:04:15", "21", "23", "96"
"2014/4/7", "04:04:18", "209", "23", "96"

```

4. Excel 檔案顯示如下。

	A	B	C	D	E	F
1	Date	Time	Millisecond	temperature	humidity	
2	2014/4/5	4:03:30	46	0	0	
3	2014/4/5	4:03:33	459	0	0	
4	2014/4/5	4:03:36	456	0	0	
5	2014/4/5	4:03:39	456	0	0	
6	2014/4/5	4:03:42	457	0	0	
7	2014/4/5	4:03:45	457	0	0	
8	2014/4/5	4:03:48	457	10	0	
9	2014/4/5	4:03:51	458	10	23	
10	2014/4/5	4:03:54	457	10	23	
11	2014/4/5	4:03:57	458	42	23	
12	2014/4/5	4:04:00	456	14	67	
13	2014/4/5	4:04:03	457	14	55	
14	2014/4/5	4:04:06	459	14	55	
15	2014/4/7	4:04:09	264	14	55	
16	2014/4/7	4:04:12	209	23	55	
17	2014/4/7	4:04:15	21	23	96	

 Note

- 當 EasyConverter 開啟的檔案內容所使用的儲存格總數量超過 600 萬格時，將只顯示部分資料。(匯出至 xls /xlsx 時仍可完整顯示資料內容)
- 在匯出至 xls /xlsx 時，強制分頁的條件如下：
  - 單一分頁行數超過 6 萬行。
  - 單一分頁儲存格使用超過 150 萬格。
- 使用 [檔案] » [另存新檔] 可將記錄檔案輸出成 PDF 檔案。

### 25.3. 將事件記錄檔案輸出至 Excel

1. 當開啟的事件記錄檔案格式為 DB，且檔案內含有一天以上的資料時，將可選擇欲檢視的日期範圍。(若開啟的檔案格式為 EVT，則略過此步驟)



2. 若該事件記錄檔案 DB 含有多國語言，則可以選擇欲檢視的語言。(若開啟的檔案格式為 EVT，則略過此步驟)



3. 若該事件記錄 DB 有使用到字串表動態 ID 功能，則為了正確顯示事件觸發時對應的字串內容，需要手動挑選字串表的檔案路徑 (\*.csv)。



4. 按下 [確定] 後，事件記錄將顯示如下，再按下 [匯出至 Excel] 即可轉換成 Excel 格式。

Event	Category	Date	Time	Message
"0","0","2014/4/5","04:03:28","Bit OFF"				
"2","0","2014/4/5","04:03:31","Bit OFF"				
"0","0","2014/4/5","04:03:31","Bit ON"				
"2","0","2014/4/5","04:03:32","Bit ON"				
"0","0","2014/4/5","04:03:32","Bit OFF"				
"1","0","2014/4/5","04:03:33","Bit OFF"				
"2","0","2014/4/5","04:03:35","Bit OFF"				
"0","0","2014/4/5","04:03:35","Bit ON"				
"1","0","2014/4/5","04:03:36","Bit ON"				
"2","0","2014/4/5","04:03:38","Bit ON"				
"0","0","2014/4/5","04:03:38","Bit OFF"				
"2","0","2014/4/5","04:03:40","Bit OFF"				
"0","0","2014/4/5","04:03:40","Bit ON"				
"2","0","2014/4/7","04:04:15","Bit ON"				
"0","0","2014/4/7","04:04:15","Bit OFF"				
"2","0","2014/4/7","04:04:16","Bit OFF"				
"0","0","2014/4/7","04:04:16","Bit ON"				

若開啟的事件記錄檔案格式為 DB，使用者可以在 EasyConverter 工作列»編輯»偏好設定中的 SQLite 事件記錄設定選擇資料展示格式，選擇格式 v2 時，則可以將確認時間以及恢復時間置於同一行事件中。



類別	觸發日期	觸發時間	確認日期	確認時間	恢復日期	恢復時間	訊息	發生次數	累積時間
"0","2022/6/14","11:07:30","","","","2022/6/14","11:07:30","String Table 0-0","1","0"									
"0","2022/6/14","11:07:31","","","","2022/6/14","11:07:31","String Table 0-0","2","0"									
"0","2022/6/14","11:07:32","","","","2022/6/14","11:07:32","String Table 0-0","3","0"									
"0","2022/6/14","11:07:32","","","","2022/6/14","11:07:33","String Table 0-0","4","1"									
"0","2022/6/14","11:07:34","","","","2022/6/14","11:07:34","String Table 0-0","5","1"									

5. Excel 檔案顯示如下。

The screenshot shows two tables side-by-side. The top table is a raw event log with columns: A (Event), B (Category), C (Date), D (Time), E (Message). The bottom table is a summary table with columns: A (Date/Time), B (Category), C (Trigger Date), D (Trigger Time), E (Confirm Date), F (Confirm Time), G (Recovery Date), H (Recovery Time), I (Message), J (Occurrence Count), K (Cumulative Time). Both tables have a 'event' tab at the bottom.



### Note

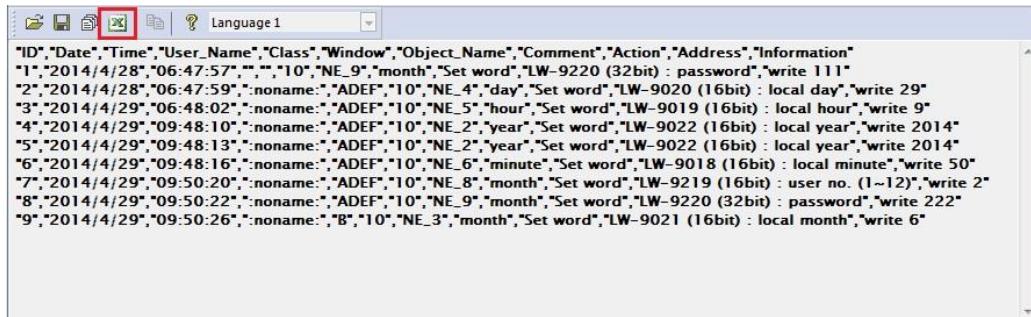
- 在表格的第一行可以發現 "Event" 欄位，**0** -> 表示事件觸發時；**1** -> 表示事件確認時；**2** -> 表示事件恢復正常時。
- 當 EasyConverter 開啟的檔案內容所使用的儲存格總數量超過 600 萬格時，將只顯示部分資料。(匯出至 xls /xlsx 時仍可完整顯示資料內容)
- 在匯出至 xls /xlsx 時，強制分頁的條件如下：
  - 單一分頁行數超過 6 萬行。
  - 單一分頁儲存格使用超過 150 萬格。
- 若有使用字串表，則在放置 eventlog db 的同一資料夾下，將字串表的 csv 檔命名與 eventlog 的 db 相同時，使用 EasyConverter 打開即會自動載入該字串表，不會再彈出詢問字串表的路徑視窗。例如：eventlog 的 db 檔案名稱為 event.db，則字串表可命名為 event.csv 讓 EasyConverter 自動載入。
- EasyConverter 最多只能開啟包含 28.5 萬列事件記錄的檔案，超過則無法開啟。
- 使用 [檔案] » [另存新檔] 可將記錄檔案輸出成 PDF 檔案。

## 25.4. 將操作記錄檔案輸出至 Excel

1. 當開啟的操作記錄檔案含有一天以上的資料時，將可選擇欲檢視的日期範圍。



2. 按下 [確定] 後，操作記錄將顯示如下，再按下 [匯出至 Excel] 即可轉換成 Excel 格式。



3. Excel 檔案顯示如下。

A	B	C	D	E	F	G	H	I	J	K	L
ID	Date	Time	User Name	Class	Window	Object Name	Comment	Action	Address	Information	
1	2014/4/28	6:47:57				10 NE_9	month	Set word	LW-9220 (32bit) : password	write 111	
2	2014/4/28	6:47:59	:noname:	ADEF		10 NE_4	day	Set word	LW-9020 (16bit) : local day	write 29	
3	2014/4/29	6:48:02	:noname:	ADEF		10 NE_5	hour	Set word	LW-9019 (16bit) : local hour	write 9	
4	2014/4/29	6:48:10	:noname:	ADEF		10 NE_2	year	Set word	LW-9022 (16bit) : local year	write 2014	
5	2014/4/29	6:48:13	:noname:	ADEF		10 NE_2	year	Set word	LW-9022 (16bit) : local year	write 2014	
6	2014/4/29	6:48:16	:noname:	ADEF		10 NE_6	minute	Set word	LW-9018 (16bit) : local minute	write 50	
7	2014/4/29	6:50:20	:noname:	ADEF		10 NE_8	month	Set word	LW-9219 (16bit) : user no. (1~12)	write 2	
8	2014/4/29	6:50:22	:noname:	ADEF		10 NE_9	month	Set word	LW-9220 (32bit) : password	write 222	
9	2014/4/29	6:50:26	:noname:	B		10 NE_3	month	Set word	LW-9021 (16bit) : local month	write 6	
10											
11											
12											

### Note

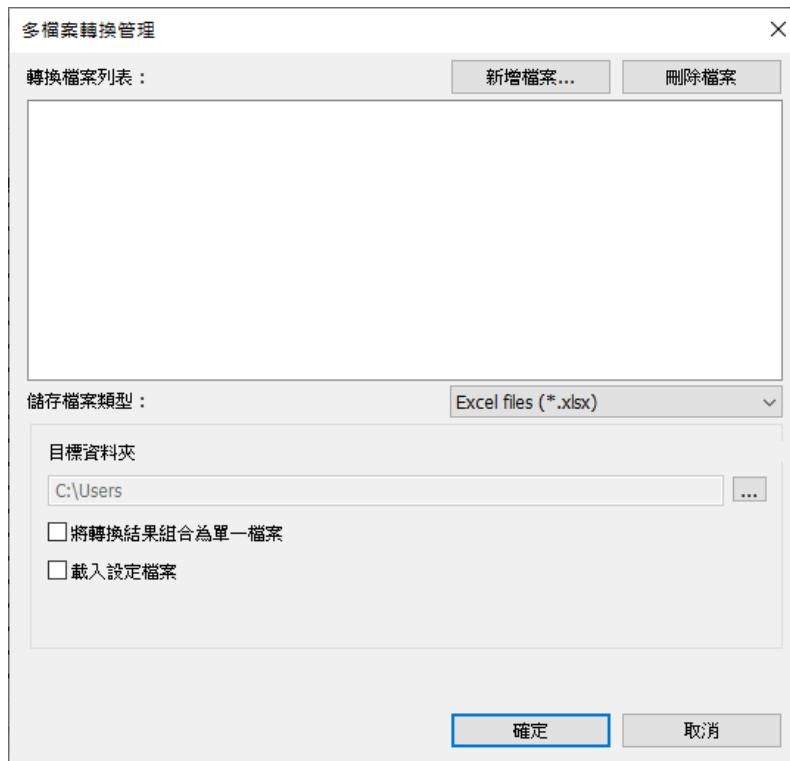
- 當 EasyConverter 開啟的檔案內容所使用的儲存格總數量超過 600 萬格時，將只顯示部分資料。(匯出至 xls /xlsx 時仍可完整顯示資料內容)
- 在匯出至 xls /xlsx 時，強制分頁的條件如下：
  - 單一分頁行數超過 6 萬行。
  - 單一分頁儲存格使用超過 150 萬格。
- 使用 [檔案] » [另存新檔] 可將記錄檔案輸出成 PDF 檔案。

## 25.5. 多檔案轉換

1. 點選 [多檔案轉換] 圖示可呼叫多檔案轉換管理視窗。



2. 點選 [新增檔案...] 可加入欲轉換的檔案名稱並請選擇儲存檔案類型(可選擇 Excel 檔案或 PDF 檔案)。若選擇 Excel 檔案且未勾選 [將轉換結果組合為單一檔案] 而按下 [確定]，檔案將個別被輸出至不同的 Excel 檔案。



3. 若勾選 [將轉換結果組合為單一檔案]，則所有的檔案將被輸出至單一個 Excel 檔案，且每個檔案分一頁籤，Excel 檔案顯示如下。

	A	B	C	D	E	F	G
1	Date	Time	Millisecond	temperature	humidity		
2	2014/3/22	6:36:52	260	2	1		
3	2014/3/22	6:36:55	250	6	3		
4	2014/3/22	6:36:58	250	10	6		
5	2014/3/22	6:37:01	300	13	8		
6	2014/3/22	6:37:04	280	17	10		
7	2014/3/22	6:37:07	250	21	13		
8							
9							
10							
11							
12							
13							
14							
15							



- 當欲合併的檔案總大小超過 32 MB 時，將無法執行合併功能。

## 25.6. 比例轉換功能

當開啟的檔案為資料取樣記錄時，可設定比例轉換功能。

比例轉換功能使用方式如下：

新數值 = [(數值 + A) × B] + C，使用者可以在 A、B 和 C 設定數值。

**A** -> 數值下限；**B** -> [(比例最大值) - (比例最小值)] / (數值上限) - (數值下限)；**C** -> 比例最小值。

範例：

有一電壓資料，其格式是 16-bit unsigned，電壓數值介於 0 ~ 4096，若要將其電壓數值轉換成伏特，介於 -5V ~ +5V 之間。

新數值 = [(數值 + 0) × 0.0024] + (-5)：



比例轉換前

Date	Time	Millisecond	sample
2014/06/30	23:02:50	80	0
2014/06/30	23:02:54	30	0
2014/06/30	23:02:57	990	55
2014/06/30	23:03:02	70	55
2014/06/30	23:03:06	20	89
2014/06/30	23:03:10	20	159
2014/06/30	23:03:14	30	530
2014/06/30	23:03:18	20	898
2014/06/30	23:03:22	40	1024
2014/06/30	23:03:26	0	2055
2014/06/30	23:03:30	30	2055

比例轉換後

Date	Time	Millisecond	sample
2014/06/30	23:02:50	80	-5.000
2014/06/30	23:02:54	30	-5.000
2014/06/30	23:02:57	990	-4.868
2014/06/30	23:03:02	70	-4.868
2014/06/30	23:03:06	20	-4.786
2014/06/30	23:03:10	20	-4.618
2014/06/30	23:03:14	30	-3.728
2014/06/30	23:03:18	20	-2.845
2014/06/30	23:03:22	40	-2.542
2014/06/30	23:03:26	0	-0.068
2014/06/30	23:03:30	30	-0.068

以上的資料設定可以儲存為範本 .lgs 格式檔案，並於下次需要時可以直接載入設定。

## 25.7. 批次檔規則

透過 EasyConverter 的 command line 功能，可以藉由執行批次檔 .bat 即可將副檔名為 .dtl 或 .evt 的來源檔案轉換成 .xls 或 .csv 的輸出檔案，也可以把存放在資料庫內的歷史資料備份到電腦上。於批次檔內亦可定義關於輸出檔案的格式 (如: ASCII, Unicode 或 UTF-8)、毫秒資訊與是否載入設定檔案等訊息。

以下說明如何建立批次檔案 .bat 及其規則說明。

指令參數說明

```
[/c{a,8,u}] [/t{0,1}] [/s "Format file"] ["Src file"] ["Dest file"]
[/d{0,1,2}] [/l{0,1,2}] [/h "Hostname"] [/n "Data Log Name"] [/g "Config File"] [/i "Time Interval"] ["Dest file"]
```

例如:

```
EasyConverter.exe /ca /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.csv"
EasyConverter.exe /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.xls"
EasyConverter.exe /d1 /l0 /h "hostname" /g "D:\config.ini" /i "5~2" "D:\test\
EasyConverter.exe /d1 /l1 /h "hostname" /n "log000" /g "D:\config.ini" "D:\test\"
```

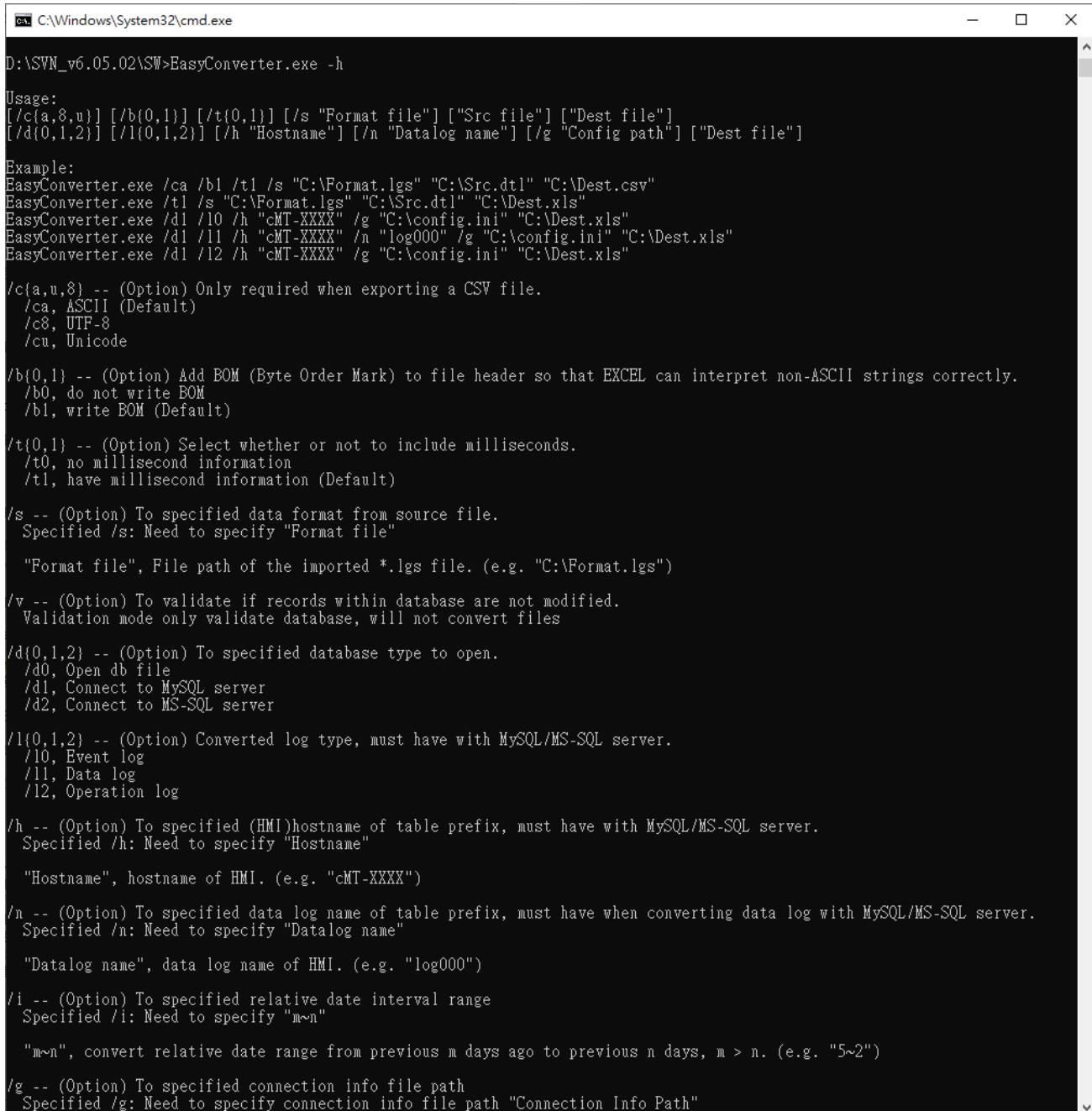
指令參數	說明
/c{a,8,u}	選擇性質，設定編碼格式，當匯出為 .csv 格式時才需要。 /ca : ASCII (預設) /c8 : UTF-8 /cu : Unicode
/t{0,1}	選擇性質，設定是否要包含毫秒資訊。 /t0 : 不包含毫秒資訊 /t1 : 包含毫秒資訊 (預設)

/s	選擇性質，設定是否要載入設定檔案。 若要載入設定檔案，需於 /s 後面指定 .lgs 的檔案路徑， 例如: /s "C:\Format.lgs"
/d{0,1,2}	選擇性質，指定資料庫類型。 /d0: 開啟 db 檔案 /d1: 開啟 MySQL Server /d2: 開啟 MS-SQL Server
/l{0,1,2}	選擇性質，指定 log 檔案類型，需搭配 MySQL 或 MS-SQL 轉檔時使用。 /l0: 事件記錄 /l1: 資料取樣 /l2:: 操作記錄
/h	選擇性質，指定主機名稱，需搭配 MySQL 或 MS-SQL 轉檔時使用。 例如: /h "cMT-XXXX"
/n	當要對資料庫中的資料取樣轉檔時，需指定資料取樣檔案名稱。 例如: /n "log000"
/g	選擇性質，指定資料庫連線資訊檔案路徑，需搭配 MySQL 或 MS-SQL 轉 檔時使用。 例如: /g "D:\config.ini"  請點選此圖示下載連線資訊範例。
/i	選擇性質，指定資料日期間格。 m: 從 m 天前 0:00 到 m 天前 23:59 m~n: 從 m 天前 0:00 到 n 天前 23:59 m~: 從 m 天前 0:00 到 db 最後一筆 ~n: 從 db 第一筆到 n 天前 23:59 例如: /i "m~n"
"Src file"	指定來源檔案路徑，檔案格式需為 .dtl, .evt, 或 .db
"Dest file"	指定輸出檔案路徑，可為 .xls, .xlsx, .csv 或 .pdf. * <a href="#">註 1</a>

註 1: 當 command line 中無指定 “Dest file” 的檔案名稱及路徑時，系統將輸出檔案至與 “Src file” 相同的目錄下。

以上說明亦可於 Windows 底下的 cmd.exe 輸入 EasyConverter.exe 的路徑來查看，例如:

D:\EasyBuilder\EB Pro>EasyConverter.exe -h



```

C:\Windows\System32\cmd.exe
D:\SVN_v6.05.02\SW>EasyConverter.exe -h

Usage:
[ /c{a,8,u} ] [ /b{0,1} ] [ /t{0,1} ] [ /s "Format file" ] [ "Src file" ] [ "Dest file" ]
[ /d{0,1,2} ] [ /l{0,1,2} ] [ /h "Hostname" ] [ /n "Datalog name" ] [ /g "Config path" ] [ "Dest file" ]

Example:
EasyConverter.exe /ca /b1 /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.csv"
EasyConverter.exe /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.xls"
EasyConverter.exe /d1 /l0 /h "cMT-XXXX" /g "C:\config.ini" "C:\Dest.xls"
EasyConverter.exe /d1 /l1 /h "cMT-XXXX" /n "log000" /g "C:\config.ini" "C:\Dest.xls"
EasyConverter.exe /d1 /l2 /h "cMT-XXXX" /g "C:\config.ini" "C:\Dest.xls"

/c{a,u,8} -- (Option) Only required when exporting a CSV file.
  /ca, ASCII (Default)
  /c8, UTF-8
  /cu, Unicode

/b{0,1} -- (Option) Add BOM (Byte Order Mark) to file header so that EXCEL can interpret non-ASCII strings correctly.
  /b0, do not write BOM
  /b1, write BOM (Default)

/t{0,1} -- (Option) Select whether or not to include milliseconds.
  /t0, no millisecond information
  /t1, have millisecond information (Default)

/s -- (Option) To specified data format from source file.
  Specified /s: Need to specify "Format file"

  "Format file", File path of the imported *.lgs file. (e.g. "C:\Format.lgs")

/v -- (Option) To validate if records within database are not modified.
  Validation mode only validate database, will not convert files

/d{0,1,2} -- (Option) To specified database type to open.
  /d0, Open db file
  /d1, Connect to MySQL server
  /d2, Connect to MS-SQL server

/l{0,1,2} -- (Option) Converted log type, must have with MySQL/MS-SQL server.
  /l0, Event log
  /l1, Data log
  /l2, Operation log

/h -- (Option) To specified (HMI)hostname of table prefix, must have with MySQL/MS-SQL server.
  Specified /h: Need to specify "Hostname"

  "Hostname", hostname of HMI. (e.g. "cMT-XXXX")

/n -- (Option) To specified data log name of table prefix, must have when converting data log with MySQL/MS-SQL server.
  Specified /n: Need to specify "Datalog name"

  "Datalog name", data log name of HMI. (e.g. "log000")

/i -- (Option) To specified relative date interval range
  Specified /i: Need to specify "m~n"

  "m~n", convert relative date range from previous m days ago to previous n days, m > n. (e.g. "5~2")

/g -- (Option) To specified connection info file path
  Specified /g: Need to specify connection info file path "Connection Info Path"

```

### 範例說明:

當欲轉換存放於 D:\EasyBuilder\EB Pro\HMI\_memory 目錄底下的 20150919.dtl 檔案成 20150919.xls 並存放至桌面時:

若批次檔 .bat 放置於與 EasyConverter 相同目錄下時，command line 為

EasyConverter.exe "D:\EasyBuilder\EB Pro\HMI\_memory\20150919.dtl" "C:\Users\Desktop\20150919.xls"

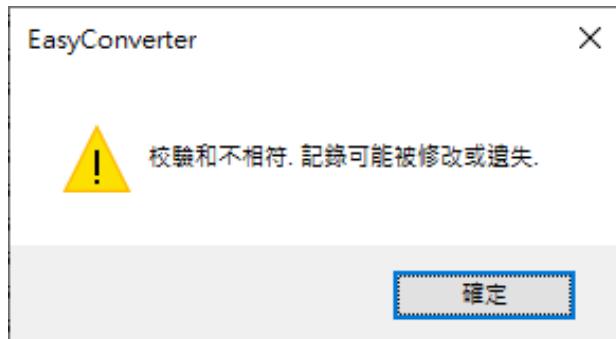
若批次檔 .bat 放置於與 EasyConverter 不同目錄下時，須另指定 EasyConverter.exe 存放路徑， command line 則為

"D:\EasyBuilder\EB Pro\EasyConverter.exe" "D:\EasyBuilder\EB Pro\HMI\_memory\20150919.dtl"

"C:\Users\Desktop\20150919.xls"

## 25.8. 檢查歷史記錄檔案完整性

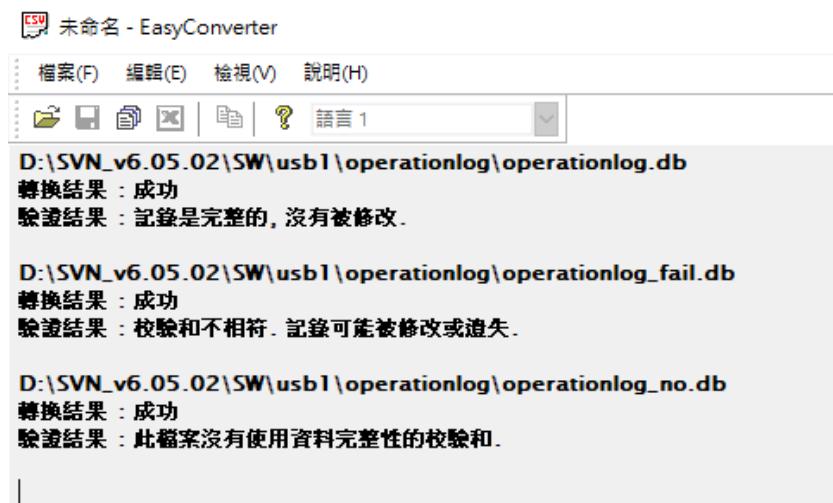
備份的資料取樣記錄、事件記錄和操作記錄可以透過 EasyConverter 判斷檔案校驗和，以確保資料完整性。EasyConverter 開啟檔案時，如偵測到記錄可能被修改，會彈出如下的警示。



在狀態列透過以下圖示顯示當下檔案的校驗和判斷結果：

- ?: 此檔案沒有使用資料完整性的校驗和。
- ✗: 校驗和不相符，記錄可能被修改或遺失。
- ✓: 記錄是完整的，沒有被修改過。

多檔案轉換也會顯示校驗和的驗證結果：



亦或可利用命令提示字元檢查檔案完整性，透過指令參數/v 即可進入驗證模式，驗證模式下不會有轉檔的行為。

例如：

EasyConverter.exe /v "C:\Src.db"

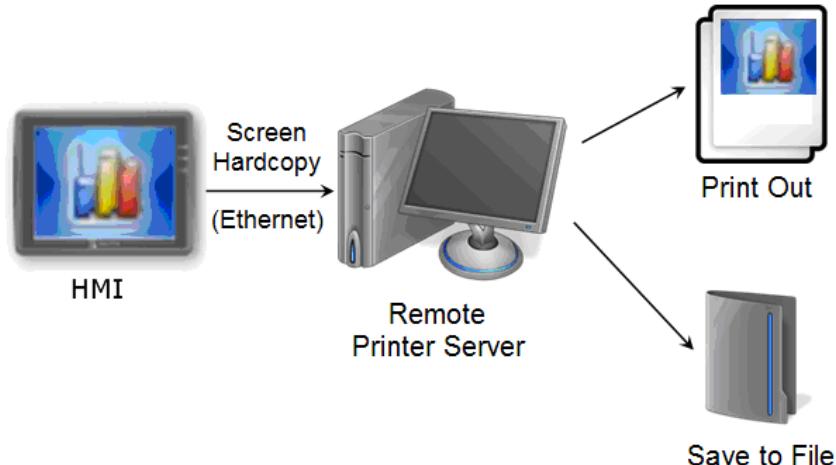
# 26. EasyPrinter

本章節說明如何設定 EasyPrinter。

26.1.	概要	26-2
26.2.	使用 EasyPrinter 為列印伺服器	26-3
26.3.	使用 EasyPrinter 為備份伺服器	26-6
26.4.	EasyPrinter 操作說明	26-10
26.5.	轉換批次檔	26-14

## 26.1. 概要

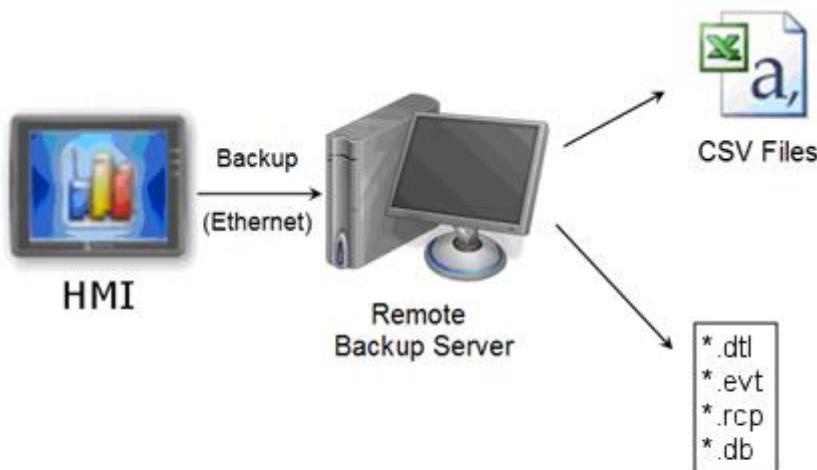
EasyPrinter 是屬於 Win 32 的應用程式，因此只能在 MS Windows 7 SP1 / 8 / 10 等系統下運行。此功能讓 HMI 可以透過乙太網路，輸出螢幕擷取畫面並列印於遠端電腦。



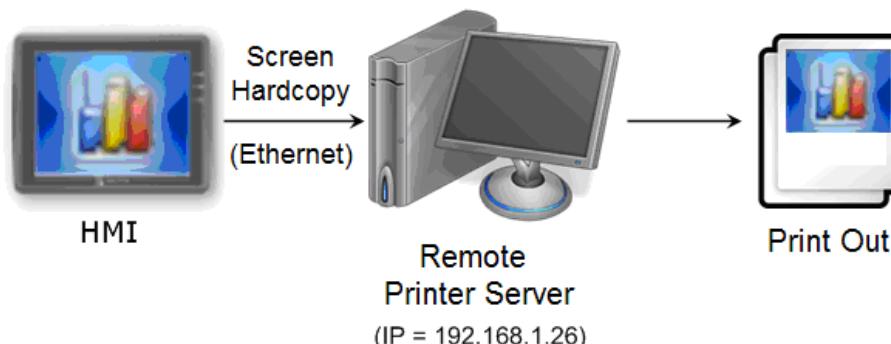
以下為使用 EasyPrinter 的優點：

- EasyPrinter 提供兩種螢幕列印輸出模式：[輸出至]及[儲存至]。使用者可使用其中一種或兩個都使用。
- 由於 Easy Printer 在 MS Windows 系統下運行，因此可支援市面上大部分的印表機。
- 此功能下多台 HMI 可以共用一台實體印表機，使用者不需為每台 HMI 各準備一台印表機。

另外，EasyPrinter 可以當作是一台備份伺服器。使用者可使用 HMI 上的備份物件，透過乙太網路，將取樣資料與事件記錄等歷史檔案備份至遠端 PC。請見下方說明：



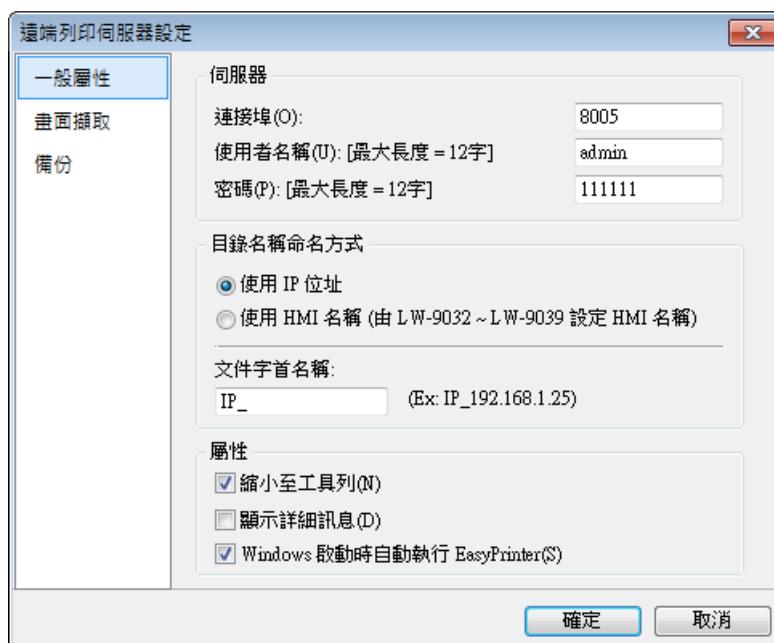
## 26.2. 使用 EasyPrinter 為列印伺服器



使用者可用 [功能鍵] 物件來操作螢幕列印。這些螢幕擷取畫面會透過乙太網路被傳送至遠端印表機伺服器，並列印出來。

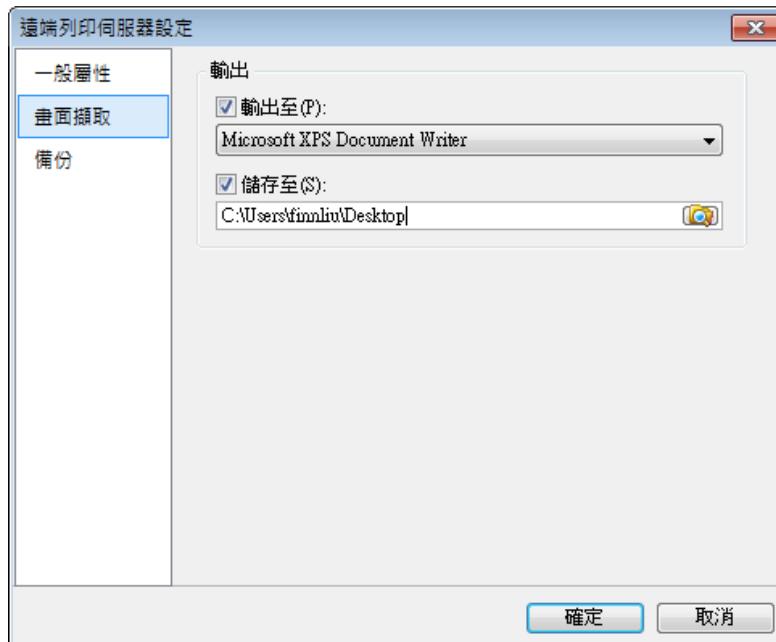
### 26.2.1. EasyPrinter 設定程序

在 EasyPrinter 設定頁下點選 [選項] » [設定] 會出現下面的對話窗：



- 點選左列 [一般屬性]
- 在 [伺服器]，設定 [連接埠] 為 “8005”，[使用者名稱] 為 “admin”，[密碼] 為“111111”。(以上皆為預設值)。
- 在 [目錄名稱命名方式]，點選 [使用 IP 位址] 並在 [文件名稱] 填入 “IP\_”。
- 在 [屬性]，選擇 [縮小至工具列]。可自行選擇在 Windows 啟動時是否自動執行 EasyPrinter 。

接著設定輸出位置。



1. 點選左列 [畫面擷取]
2. 在 [輸出]，點選 [輸出至] 並選擇一台印表機做為螢幕列印的輸出裝置。(注意：使用者只能選擇自己的系統中存在的印表機，上列印表機僅為參考。)
3. 點選 [儲存至] 並選擇檔案儲存路徑，EasyPrinter 將會把擷取檔案一併儲存在指定路徑裡。
4. 按下 [確定] 確認使用以上設定。
5. 在 EasyPrinter 設定頁下點選 [檔案] » [允許輸出]，EasyPrinter 會將這些列印指令輸出。

### 26.2.2. EasyBuilder Pro 設定程序

在 EasyBuilder Pro 設定 EasyPrinter 的設定程序：

1. 開啟 EasyBuilder Pro，並開啟新專案或已使用之專案。
2. 在 [常用] » [系統參數設定] » [列印/備份伺服器] 中，勾選 [使用遠端列印/備份伺服器]。



3. 在 [輸出設定]，指定適當的邊界，(在此例中上下左右邊界皆設為 15mm)。
4. 在 [通訊設定]，輸入列印伺服器 [IP 位址]，同 EasyPrinter 的設定。指定 [連接埠] 號碼 “8005”，[使用者名稱] 為 “admin”，[密碼] 為 “111111”。
5. 按下 [確定]。
6. 接著在功能表 [物件] » [開關] 選擇 [功能鍵] 並在物件設定頁中點選 [畫面列印] 並將 [印表機] 設定至 [遠端列印/備份伺服器]。

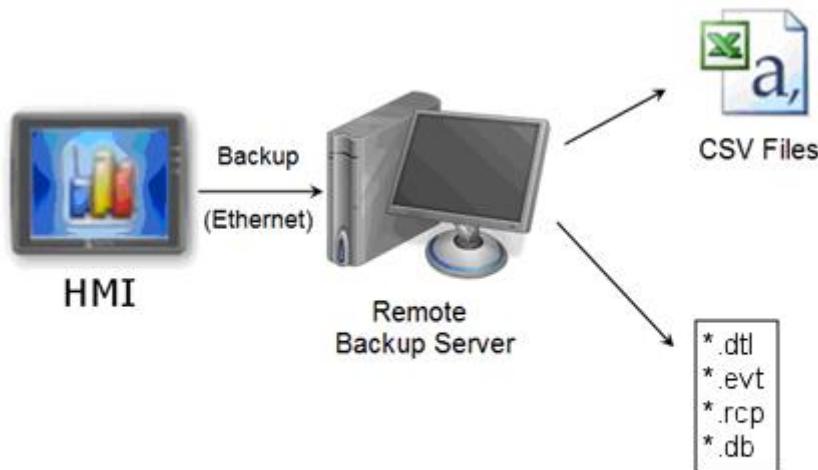


7. 將 [功能鍵] 物件置於 [公共視窗] (四號視窗)，使用者便可以隨時開始螢幕列印
8. [編譯] 及 [下載] 工程檔案至 HMI，按下已設定的 [功能鍵] 物件，開始列印。

### Note

- 使用者亦可透過 [PLC 控制] 物件來達成螢幕列印
- 警報資料無法透過 EasyPrinter 列印。
- EasyPrinter 只能透過乙太網路與 HMI 通訊，請確認所使用 HMI 網路是否設定正確。

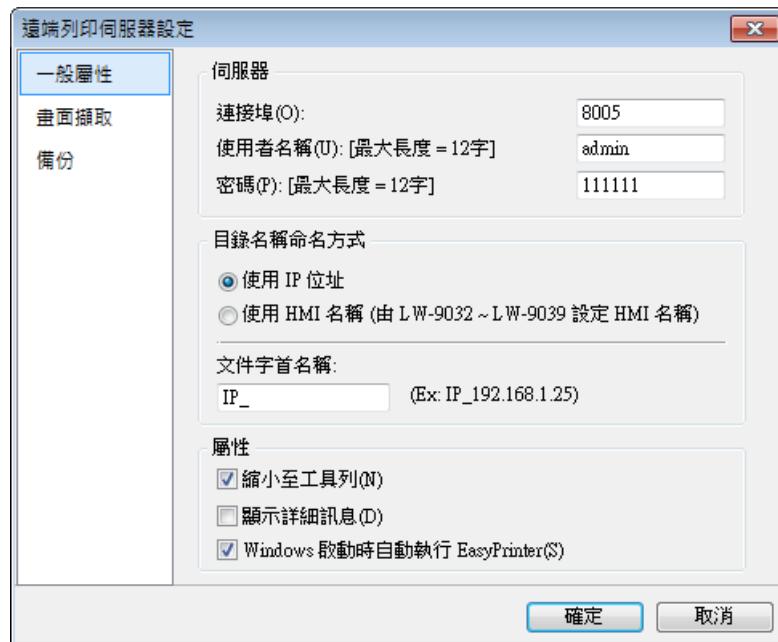
### 26.3. 使用 EasyPrinter 為備份伺服器



使用者可使用 [備份] 物件，將歷史資料，以及操作記錄等上傳至遠端備份伺服器。

#### 26.3.1. EasyPrinter 設定程序

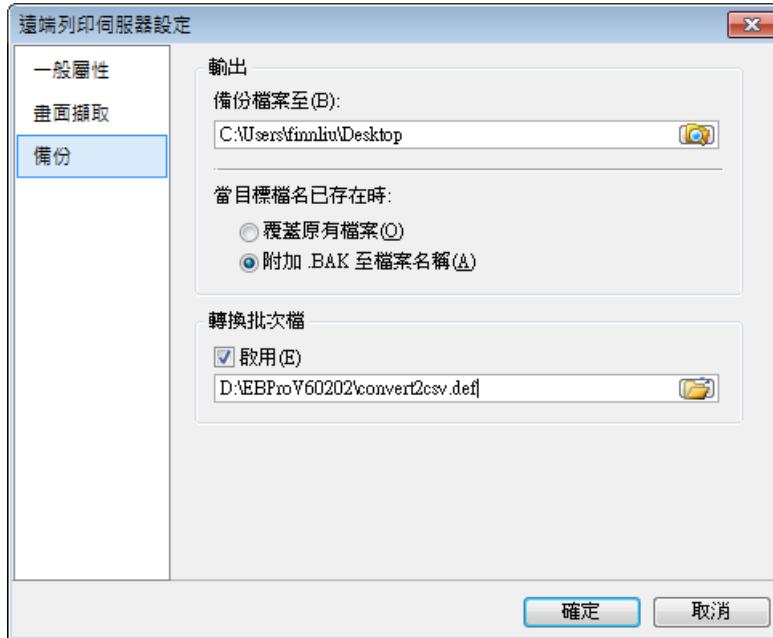
在 EasyPrinter 設定頁下點選 [選項] » [設定] 會出現下面的對話窗：



1. 在左列選擇 [一般屬性]
2. 在 [伺服器] 中，指定 [連接埠] 為 “8005”，[使用者名稱] 為 “admin”，[密碼] 為 “111111”。(以上皆為預設值)。
3. 在 [目錄名稱命名方式]，點選 [使用 IP 位址] 並指定“IP\_”為 [文件名稱]。

- 在 [屬性]，選擇 [縮小至工具列]。可自行選擇在 Windows 啟動時是否自動執行 EasyPrinter。

接著設定備份位置。



- 點選左列 [備份]
- 在 [輸出]，點擊 來瀏覽及選擇歷史資料的儲存路徑。
- 按下 [確定] 確認使用以上設定。
- 在 EasyPrinter 設定頁下點選 [檔案] » [允許輸出]，EasyPrinter 會將備份資料儲存在方才所選的路徑。

### 26.3.2. EasyBuilder Pro 設定程序

接著在 EasyBuilder Pro 工程檔案中加入相關設定：

- 開啟 EasyBuilder Pro，並開啟新專案或已使用之專案。
- 在 [常用] » [系統參數設定] » [列印/備份伺服器] 中，勾選 [使用遠端列印/備份伺服器]。



3. 在 [通訊設定]，輸入列印伺服器 [IP 位址] 同 EasyPrinter 的設定，指定 [連接埠] 號 “8005”，[使用者名稱] 為 “admin”，[密碼] 為 “111111”。(以上皆為預設值)。
4. 按下 [確定]。

接下來添加備份功能到視窗。

1. 接著在功能表 [資料/歷史] 選擇 [備份] 會出現下面的對話窗：



2. 在 [來源]，選擇 [事件記錄] (或照需求選擇 [RW]、[RW\_A])。
3. 在 [備份位置]，選擇 [遠端列印/備份伺服器]
4. 在 [範圍]，選擇 [今天] 和 [全部] (或照實際需求改變)。
5. 在 [觸發]，選擇 [手動]
6. 按下 [確定]
7. 將 [備份] 物件置於視窗中，例如 [4:共用視窗]，使用者便可隨時執行備份。
8. [編譯] 及 [下載] 工程檔至 HMI，按下前面設定的 [備份] 物件，開始備份歷史資料。

### Note

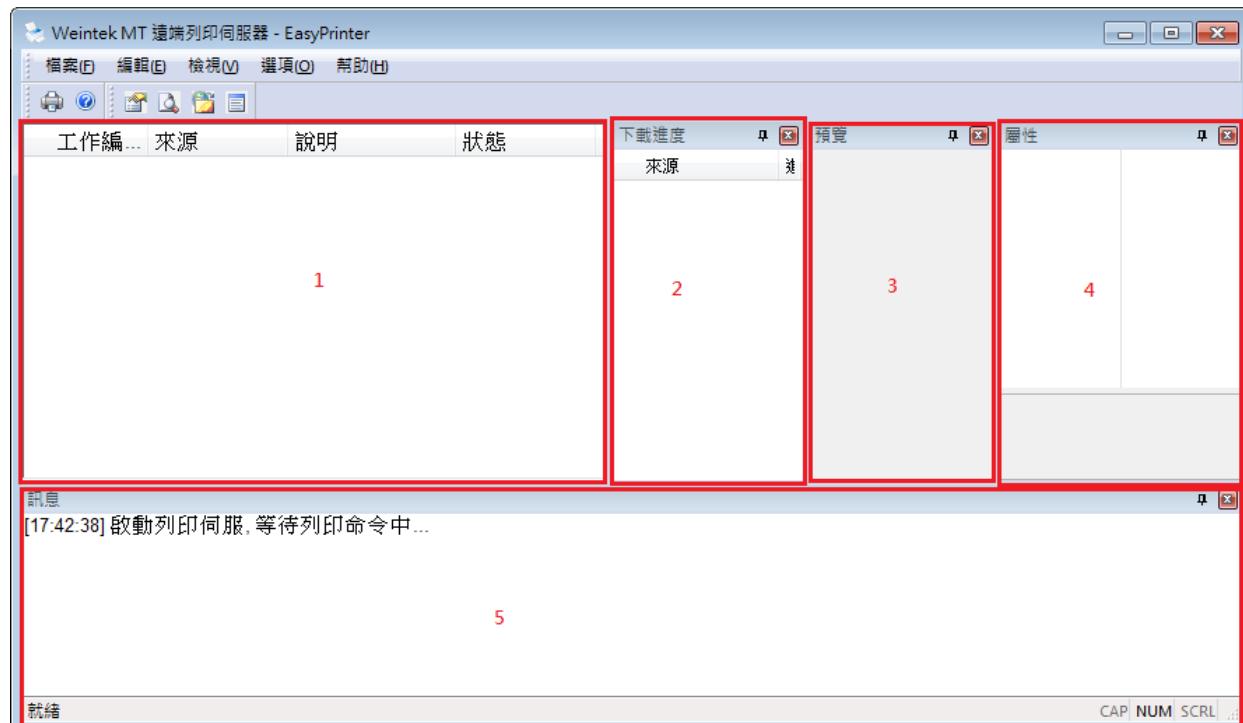
- [備份] 物件亦可用位元位址觸發。
- 使用者可以放置一個 [排程] 物件，在一週的最後一天轉為 ON，用以觸發 [備份] 物件自動備份所有歷史資料。

## 26.4. EasyPrinter 操作說明

以下介紹 EasyPrinter 視窗介面和操作說明。

### 26.4.1. 視窗介面

EasyPrinter 視窗可切分為五個區域，如下圖。



區域	名稱	描述
1	工作列表	此視窗顯示所有工作，包括螢幕列印與備份工作。
2	下載進度視窗	此視窗顯示新進工作之下載進度。
3	預覽視窗	此視窗顯示工作列表中所選螢幕列印之預覽影像。
4	屬性視窗	此視窗顯示工作列表中所選工作之資訊。
5	訊息視窗	此視窗顯示工作執行中的時間與訊息，例如密碼錯誤等等。

### 26.4.2. 選單項目

以下說明其他 EasyPrinter 功能。

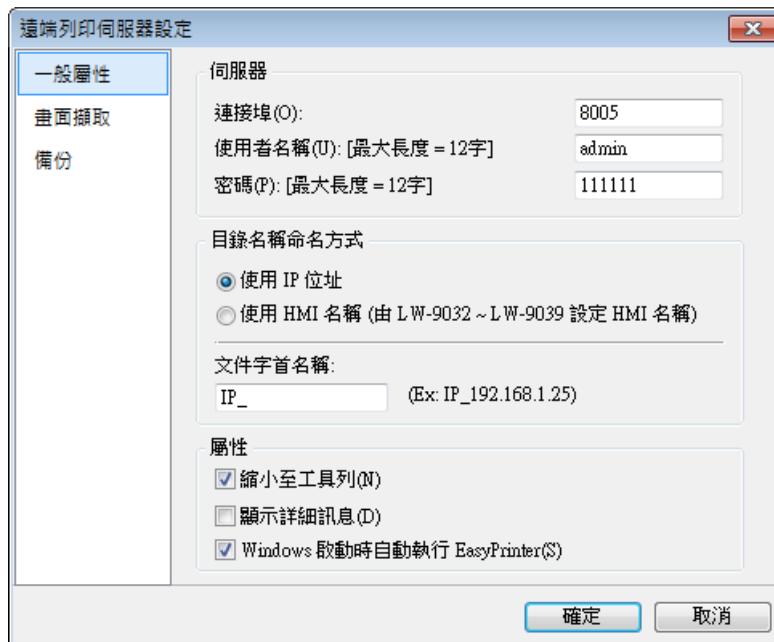
選單項目	描述
<b>檔案</b>	<b>允許輸出</b> 已選：EasyPrinter 執行工作 未選：EasyPrinter 將工作保留於記憶體中
<b>編輯</b>	<b>編輯</b> 編輯 [螢幕列印]。使用者可以在此自由設定方向、列印大小、邊界。 <b>刪除</b> 永久刪除所選工作。 <b>選擇全部</b> 選擇 [工作列表] 上的所有工作。
<b>檢視</b>	<b>屬性視窗</b> 顯示或隱藏屬性視窗。 <b>預覽視窗</b> 顯示或隱藏預覽視窗。 <b>下載進度視窗</b> 使用者可以選擇下載進度的顯示方式。在視窗中，可點擊 <b>【進度】</b> 選擇依百分比顯示或資料長度顯示。 
<b>選項</b>	下頁詳細說明各項設定和意義。

#### Note

- EasyPrinter 只能將最多至 128MB 的工作資料保留於記憶體中，若記憶體已滿，所有新進工作會被拒絕。使用者可以選擇 [允許輸出] 直接執行，或是刪除部分工作來清出記憶空間給新進工作。
- 備份工作不可編輯。
- 只有在選擇工作後才能 [編輯]。
- 選擇至少一樣工作方可 [刪除]。

以下詳細說明 [選項] » [設定] 的各項設定和意義。

- 一般屬性頁籤

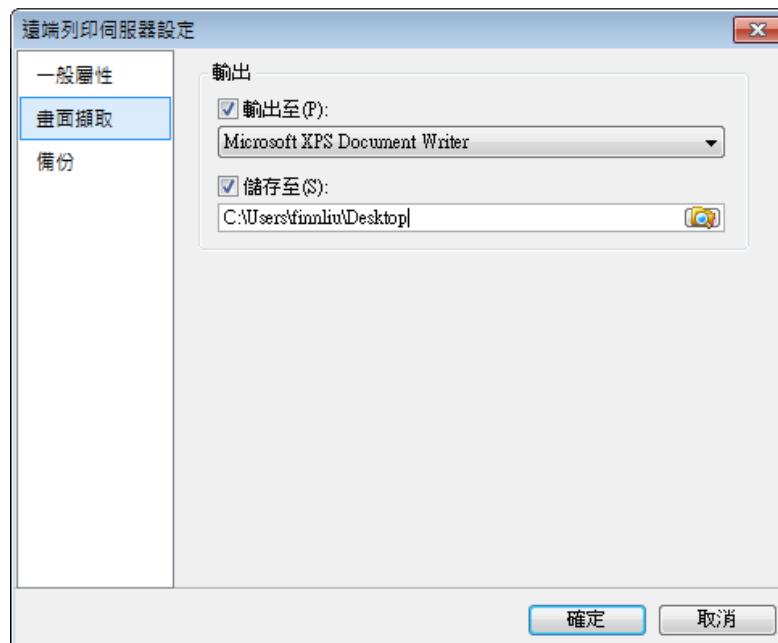


設定	描述
伺服器	<b>連接埠</b> 設定乙太網路接口讓 HMI 連結，範圍 1~65535，8005 是預設值。 <b>使用者名稱 / 密碼</b> 設定使用者名稱及密碼，讓有授權的 HMI 方能使用 EasyPrinter 功能。
目錄名稱命名方式	<b>EasyPrinter</b> 使用不同的資料夾儲存來自不同 HMI 的各種檔案(螢幕列印點陣圖檔、備份檔等等)。有兩種方式可以命名這些資料夾： <b>使用 IP 位址</b> 在使用所設 IP 位址的人機送出指令後，EasyPrinter 會為資料夾命名為文件名稱+IP 位址。請見下圖。 
	<b>使用 HMI 名稱</b>

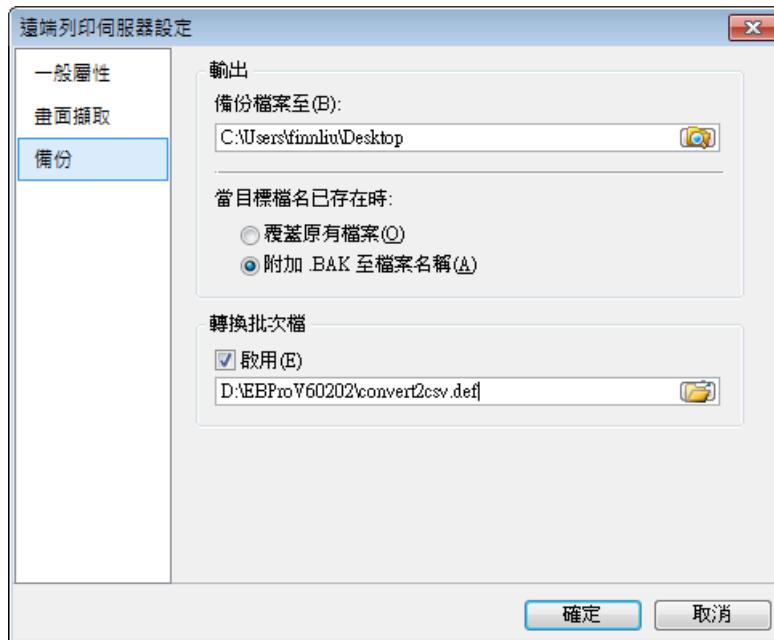
EasyPrinter 使用送出指令的 HMI 之名稱來為資料夾命名：[文件名稱]+[HMI 名稱]。

- 屬性
  - 縮小至工具列**  
勾選此項，EasyPrinter 會被縮小並置於工具列中，使用者只要雙擊工具列上的圖像，即可打開 EasyPrinter。
  - 詳細訊息**  
勾選此項，訊息視窗中會顯示更詳細的事件訊息。
  - Windows 啟動時自動執行 EasyPrinter**  
勾選此項，當 Windows 啟動時，EasyPrinter 會自動執行。

- 螢幕擷取頁籤：



設定	描述
輸出	<b>輸出至</b> 選擇此項告知 EasyPrinter 將螢幕列印結果透過特定印表機列印出來。 <b>儲存至</b> 選擇此項告知 EasyPrinter 將螢幕列印結果轉成點陣圖檔，並儲存於指定路徑。使用者可以在以下路徑找到點陣圖檔： <b>[使用者指定路徑] \ [HMI 資料夾] \ yyymmdd_hhmm.bmp</b> 舉例來說，當一個螢幕列印指示發生於 17:35:00 2009 年 1 月 12 日，點陣圖檔將被命名為"090112_1735.bmp"。如果同一分鐘有另一個點陣圖檔產生，新圖檔將被命名為"090112_1735_01.bmp"，以此類推。
備份頁籤	



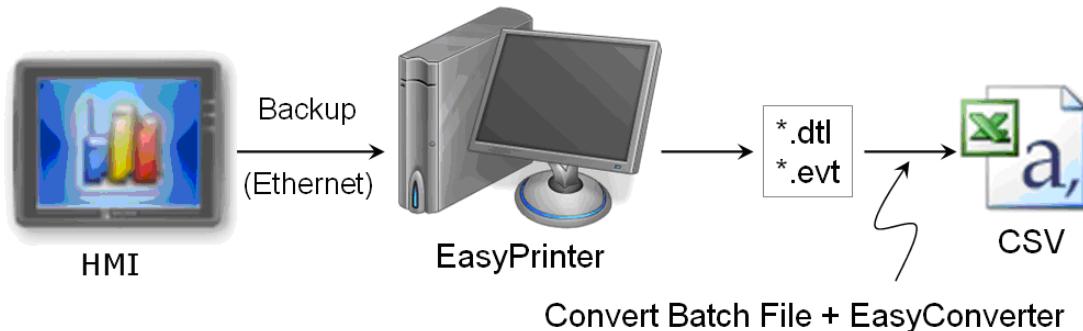
設定	描述
輸出	EasyPrinter 將備份檔案儲存於特定路徑下。  上層路徑：[使用者指定路徑] \ [HMI 名稱] 或 [IP 位址]
下層路徑：	<ul style="list-style-type: none"> <li>● 事件記錄歷史資料：\ eventlog\ EL_yyyymmdd.evt</li> <li>● 資料取樣歷史資料：\ datalog \ [資料取樣物件的檔案名稱]\ yyyymmdd.dtl</li> <li>● 配方數據：\ recipe \ recipe.rcp 或 recipe_a.rcp</li> <li>● 配方資料庫：\ recipe \ recipe.db</li> <li>● 操作記錄：\ operationlog\ operationlog.db</li> </ul>
轉換批次檔	勾選 [開啟] 指定自動將上傳之歷史檔案轉檔成.csv 或.xls (Excel) 檔之轉換批次檔。請參考下一小節。

### Note

- 使用者可以用暫存器 LW-9032 至 LW-9039 來指定 HMI 名稱。

## 26.5. 轉換批次檔

EasyPrinter 提供一個轉檔裝置，可將上傳之資料取樣與事件記錄等兩種歷史檔案自動存成 .csv 檔。使用者若需要此功能，須先準備一個轉換批次檔，告知 EasyPrinter 如何轉換歷史檔案。如下所示，此轉檔功能實際是由 EasyConverter 執行，EasyPrinter 只是遵照轉換批次檔的標準來用正確的參數啟動 EasyConverter 達成轉檔指示。

 **Note**

- EasyConverter 是另一個 Win32 應用程式，可將歷史資料轉換成.csv 或 Excel 的.xls 等檔案。使用者可在 EasyBuilder Pro 下載路徑中找到這個程式。
- 使用者若需使用此功能，須先確定 EasyPrinter 及 EasyConverter 皆被放置於相同路徑下。

### 26.5.1. 轉換批次檔預設值

以下為預設轉換批次檔。

程式碼 1 預設轉換批次檔 (convert2csv.def)

```
1: "dtl", "EasyConverter /c $(路徑名稱)"  
2: "evt", "EasyConverter /c $(路徑名稱)"
```

檔案文字會以兩行呈現，每行含有兩個參數，用逗號分隔，形成對應特定類型檔案(資料取樣與事件記錄)的處理標準。第一個參數顯示該檔案類型的副檔名，第二個參數顯示操作模式所需執行的命令。“\$(路徑名稱)”是關鍵字，告訴 EasyPrinter 需用轉檔的備份檔案名稱來取代之。例如，資料取樣檔案名稱為 20090112.dtl，已被上傳及儲存，EasyPrinter 會輸出以下指令於命令視窗。

```
1: EasyConverter /c 20090112.dtl
```

如此一個名稱為 20090112.csv 的檔案即被建立。

因此，轉換批次檔的預設標準如下：

1. 轉換所有資料取樣歷史檔案(.dtl)為.csv 檔。
2. 轉換所有事件記錄歷史檔案(.evt)為.csv 檔。

#### Note

- 事實上，第二個參數中的“\$(路徑名稱)”代表檔案的完整路徑名稱，在前面的例子中，EasyPrinter 以下面名稱取代：  
[使用者指定路徑]\[HMI 資料夾]\[資料記錄]\[資料取樣物件檔案名稱]\20090112.dtl
- EasyPrinter 以一行檔案文字為單位來解讀轉換批次檔，也就是說，一行文字形成一個標準。
- 任兩個參數皆須以逗號分隔。
- 任一參數皆須以雙引號標示。
- 同一參數中切勿放逗號。
- 目前支援的參數為\$(PathName)"、\$(HmiName)、\$(IP)，分別代表”檔案的完整路徑名稱”、“來源 HMI 名稱”、“來源 HMI IP 位址”。
- 當設定內有勾選[Windows 啟動時自動執行 EasyPrinter]，批次檔預設值的 EasyConverter 必須加上在電腦中的路徑才能正確轉換。

```
1: "dtl", "C:\EBPro\EasyConverter /c $(路徑名稱)"  
2: "evt", "C:\EBPro\EasyConverter /c $(路徑名稱)"
```

 詳細資訊請參考《25 EasyConverter》。

### 26.5.2. 特定標準

請參考以下特定標準：

- 可針對特定 HMI 上傳的檔案使用特別的操作，例如程式碼 2。
- 可用 HMI 的名稱來辨別該 HMI，例如程式碼 3。

- 或是針對不同的資料取樣歷史記錄需要不同的操作方式，例如程式碼 4。  
(只適用在 [資料取樣] 檔案，且名稱為 "Voltage" 時。)  
第三個參數 ("\*") 表示接受來自任何 HMI 的資料取樣當中符合標準者。  
使用者可以轉換第三個參數為 "192.168.1.26"，"192.168.1.\*"，HMI 名稱等等，用以減少目標 HMI 範圍。

程式碼 2 針對 HMI IP = 192.168.1.26 的特別定義標準

```
1: "dtl", "EasyConverter /c ${Pathname}", "192.168.1.26"
```

程式碼 3 針對 HMI 名稱為 Weintek\_01 的特別定義標準

```
1: "dtl", "EasyConverter /c ${Pathname}", "Weintek_01"
```

程式碼 4 針對資料取樣物件檔案名稱 = Voltage 的特別定義標準

```
1: "dtl", "EasyConverter /s Voltage.lgs ${Pathname}", "*", "Voltage"
```

### 26.5.3. 轉換批次檔格式

以下列出標準格式與各參數之說明。

檔案類型	指令(行)	HMI IP/名稱	條件 1	條件 2
------	-------	-----------	------	------

- 檔案類型  
這個參數特定出此標準所針對的上傳檔案類型之副檔名(e.g. ".dtl" 為資料取樣歷史檔案，".evt" 為事件記錄歷史檔案)
- 指令(行)  
當上傳檔案符合標準時，EasyPrinter 送至命令視窗的確切指令。
- HMI IP/名稱  
此參數指定這個標準所針對的 HMI。
- 條件 1  
如果檔案類型是 ".dtl"，此參數將這個標準所針對之 [資料取樣] 物件的檔案夾名稱指定出來。對其他檔案類型無效。
- 條件 2  
未使用 (保留為以後使用)

### 26.5.4. 執行順序

EasyPrinter 於每個檔案上傳後，由下往上驗證各標準。一旦符合標準，就會停止驗證，並開始處理下一個檔案。因此，使用者可將較廣泛的標準放在下方，而將較明確的標準置於上方。例如，以下為批次檔內容：

```
"dtl", "EasyConverter /c $(路徑名稱)"
"evt", "EasyConverter /c $(路徑名稱)"
"dtl", "EasyConverter /c $(路徑名稱)", "192.168.1.26"
"dtl", "EasyConverter /c $(路徑名稱)", "my_HMI_01"
"dtl", "EasyConverter /c $(路徑名稱)", "my_HMI_02"
"dtl", "EasyConverter /s Voltage.lgs $(路徑名稱)", "*", "Voltage"
```

其正確的順序為 (由最後一行往上執行)：

```
"dtl", "EasyConverter /s Voltage.lgs $(路徑名稱)", "*", "Voltage"
"dtl", "EasyConverter /c $(路徑名稱)", "my_HMI_02"
```

```
"dtl", "EasyConverter /c $(路徑名稱)", "my_HMI_01"  
"dtl", "EasyConverter /c $(路徑名稱)", "192.168.1.26"  
"dtl", "EasyConverter /c $(路徑名稱)"  
"evt", "EasyConverter /c $(路徑名稱)"
```

# 27. EasySimulator/cMT Viewer 模擬

本章節說明如何使用 EasySimulator/cMT Viewer 在不用運行 EBpro 的狀況下進行離線/連線模擬。

27.1. EasySimulator .....	27-2
27.2. cMT Viewer.....	27-3

## 27.1. EasySimulator

### 27.1.1. 概要

EasySimulator 功能允許使用者不用開啟 EasyBuilder Pro 軟體，即可執行 eMT/iE/XE/mTV/iP 系列的連線/離線模擬。使用此功能前，使用者須先將執行 EasySimulator 所需的相關檔案放置於同一資料夾。

### 27.1.2. 設定 EasySimulator 的步驟

1. 為了執行 EasySimulator，需準備相關檔案：

- [driver] → [win32]
- com\_e30.exe
- EasySimulator.exe
- gui\_e30.exe
- sqlite3.dll
- xob\_pos.def
- libcurl.dll
- libeay32.dll
- MFC71.dll
- mosquitto.dll
- mosquitopp.dll
- pthreadVC2.dll
- ssleay32.dll

2. 使用文字編輯工具開啟 xob\_pos.def (例如：記事本)，並正確設定相對內容。

```
"2" // operation mode 2: off-line 3: on-line
"C:\EBpro" // define the directory of com_e30.exe and gui_e30.exe
"C:\EBpro\project\emt_demo_800x600.exob" // define the directory of exob file
```

行數	描述
1	“2” 執行離線模擬；“3” 執行連線模擬。
2	指出相關檔案的存放路徑。 (例如：com_e30.exe, gui_e30.exe, EasySimulator.exe...等等)。
3	指出 .exob 檔案的存放路徑。

3. 雙擊 EasySimulator.exe 即可開始執行模擬。

4. 連線 / 離線模擬結果將顯示在螢幕上。



- 以上相關檔案皆可在 EasyBuilder Pro 安裝目錄資料夾裡找到，所以使用者必須在某部電腦上安裝 EasyBuilder Pro 軟體後，再將這些檔案複製到目標電腦上。
- 若執行 EasySimulator.exe 後沒有反應，請再次確認相關檔案路徑是否定義正確。
- 若彈出 “Failed to open project file : No such file or directory.” 的視窗，表示 .exob 檔路徑錯誤，請再次確認是否定義正確。

## 27.2. cMT Viewer

### 27.2.1. 概要

cMT Viewer 功能允許使用者不用開啟 EasyBuilder Pro 軟體，即可執行 cMT/cMT X 系列的連線/離線模擬。

### 27.2.2. 設定 cMT Viewer 的步驟

1. 使用 WinRAR 等第三方解壓縮軟體將 .cxob 解壓縮兩次，在 mt8000 資料夾內會產生一個 project 檔案。
2. 在命令提示字元中，使用以下命令。

連線模擬

```
cMTViewer.exe -ns "path/to/project/file" -allinone
```

離線模擬

```
cMTViewer.exe -fs "path/to/project/file" -allinone
```

# 28. EasySystemSetting

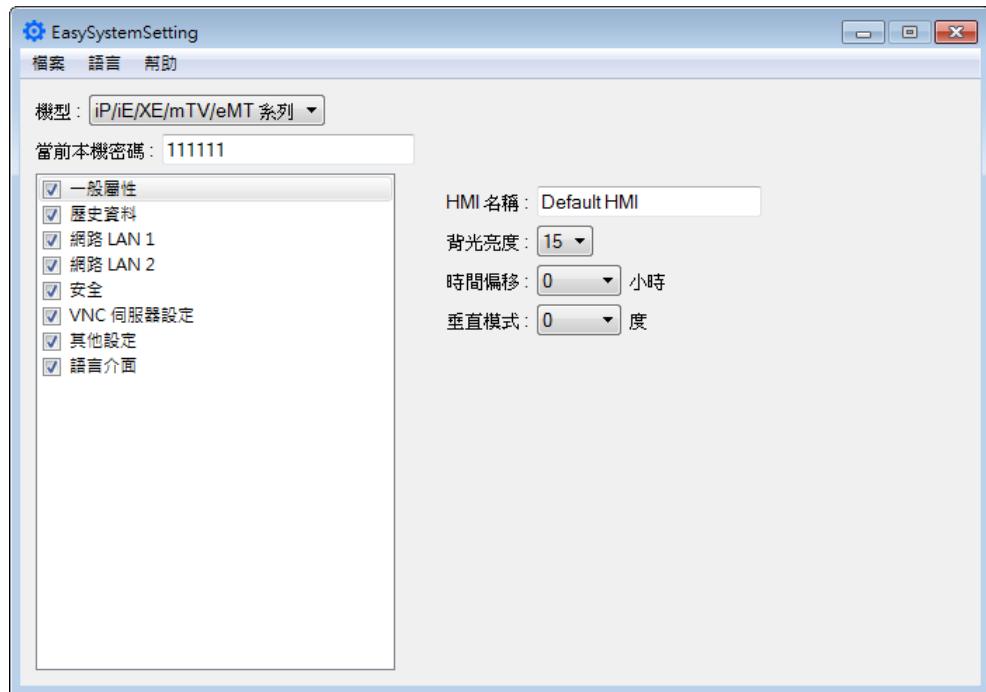
本章節說明如何使用 EasySystemSetting。

28.1. EasySystemSetting .....	28-2
28.2. 操作流程範例 .....	28-5

## 28.1. EasySystemSetting



EasySystemSetting 提供使用者利用 USB 碟或 SD 卡更新硬體設定資訊的功能。在 EasyBuilder Pro 安裝資料夾下點擊 EasySystemSetting.exe 開啟，或是在 EasyBuilder Pro 編輯軟體的 [工具] 頁籤下點選 [系統設定] 編輯器。HMI 的韌體版本需使用 OS 20131106 或更新版本。



設定	描述
檔案	[新增] 將所有設定回復為預設值。 [載入] 將已儲存的.conf 檔案匯入並編輯。 [儲存] 將完成設定的資訊匯出成.conf 檔案。
語言	變更顯示語言。
幫助	[關於] 版本資訊。

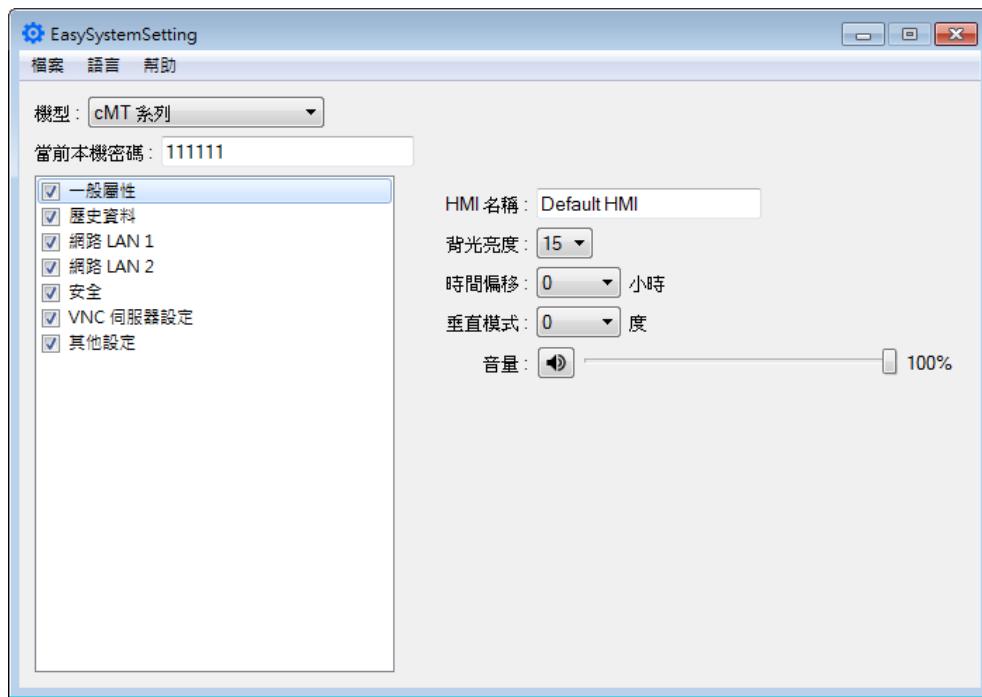
## iP/iE/XE/mTV/eMT 系列



以上設定皆與本手冊第四章描述的系統設定內容相同，以下將介紹一般屬性頁籤。

設定	描述
當前本機密碼	請填入登入系統參數設定的密碼。
一般屬性	<p>[HMI 名稱] 設定 HMI 名稱。只支援 [A~Z]、[a~z]、[0~9]、及[-]、[.] 字元。</p> <p>[背光亮度] 調整背光燈亮度。</p> <p>[時間偏移] 將 HMI 的 RTC 時間設定偏移。</p> <p>例如：目前 RTC 為 15:00:00，若設定時間偏移值為-3，更新後將為 12:00:00。</p> <p>[垂直模式] 設定顯示的旋轉模式。</p>

## cMT / cMT X 系列



以上設定皆與本手冊第四章描述的系統設定內容相同，以下將介紹一般屬性頁籤。

設定	描述
<b>當前本機密碼</b>	請填入登入系統參數設定的密碼。
<b>一般屬性</b>	<p>[HMI 名稱] 設定 HMI 名稱。只支援 [A~Z]、[a~z]、[0~9]、及 [-]、[.] 字元。</p> <p>[背光亮度] 調整背光燈亮度。</p> <p>[時間偏移] 將 HMI 的 RTC 時間設定偏移。</p> <p>例如：目前 RTC 為 15:00:00，若設定時間偏移值為 -3，更新後將為 12:00:00。</p> <p>[垂直模式] 設定顯示的旋轉模式。</p> <p>[音量] 設定音量大小。</p>

## 28.2. 操作流程範例

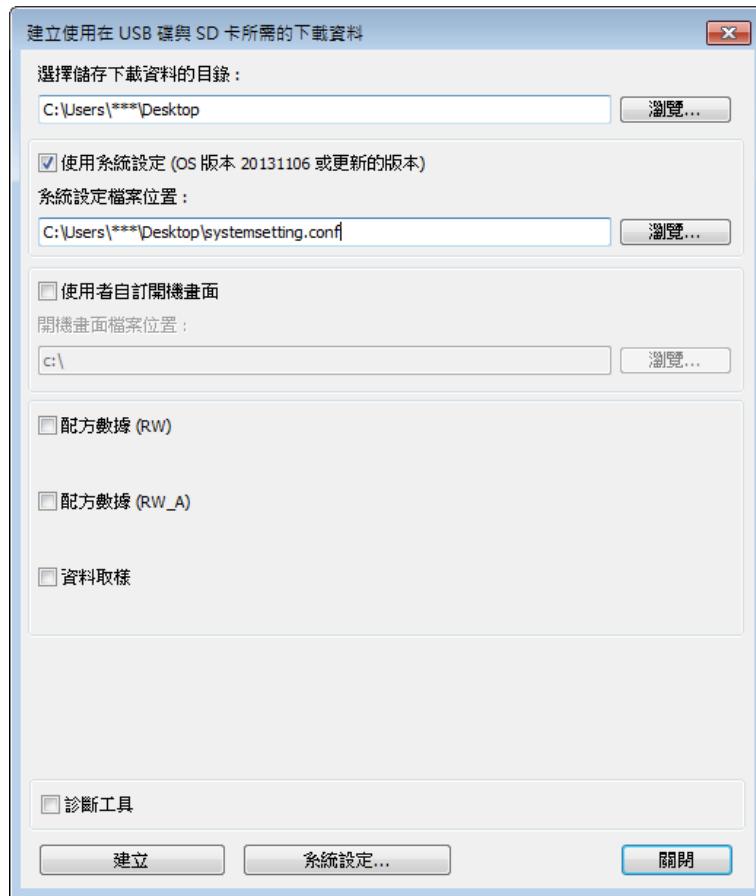
以下介紹如何使用 USB 碟或 SD 卡更新 HMI 的 IP 位址。

### iP/iE/XE/mTV/eMT 系列

- 開啟 EasySystemSetting 對話窗，將 HMI 的 Network 資訊更改如下所示。



- 點選 [檔案] » [儲存] 產生一個 systemsetting.conf 檔案。
- 關閉 EasySystemSetting。
- 點選 [建立使用在 USB 碟與 SD 卡所需的下載資料]，勾選 [使用系統設定] 並將.conf 檔案匯入。



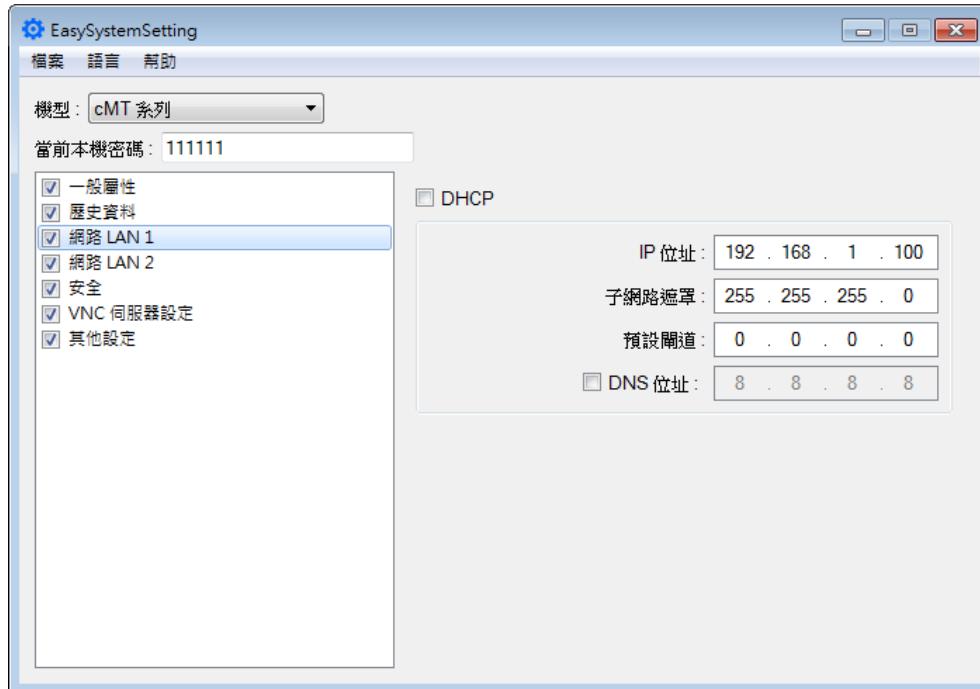
5. 點選 [建立使用在 USB 碟與 SD 卡所需的下載資料] 中的 [建立] 產生 USB 碟與 SD 卡下載檔案。
6. 將儲存該檔案的外部裝置插入 HMI，將彈出 Download/Upload 對話窗。



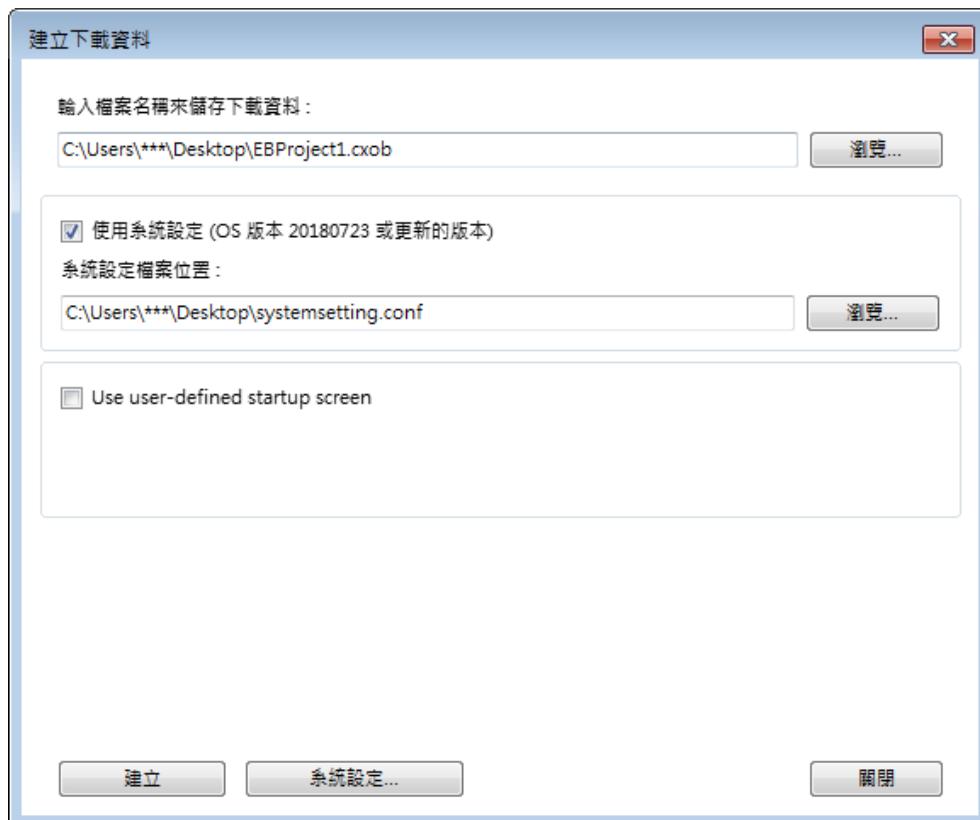
7. 點選 [Load System Setting] 將顯示 [Download Config Settings] 訊息。完成更新系統設定後，將繼續更新工程檔案。

## cMT / cMT X 系列

- 開啟 EasySystemSetting 對話窗，將 HMI 的 Network 資訊更改如下所示。



- 點選 [檔案] » [儲存] 產生一個 systemsetting.conf 檔案。
- 關閉 EasySystemSetting。
- 在 EasyBuilder Pro 點選 [建立下載資料]，勾選 [使用系統設定] 並將.conf 檔案匯入。



5. 按下建立後會產生相同名字的程式編譯檔與系統設定檔(EBProject1.cxb & EBProject1.conf)，將兩個檔案放在外部裝置的相同目錄下。
6. 將儲存該檔案的外部裝置插入 HMI，將彈出下載/上傳對話窗，點選下載。將工程檔案與下載密碼正確輸入後點選下載。



### Note

- Utility Manager 的下載功能與 EB Pro 的下載功能都可以直接下載系統設定檔案。

# 29. 穿透通訊功能

本章節說明如何設定穿透通訊功能。

29.1.	概要 .....	29-2
29.2.	序列埠穿透的乙太網路模式 .....	29-2
29.3.	序列埠穿透的序列埠模式 .....	29-7
29.4.	穿透通訊控制 .....	29-9
29.5.	SIEMENS S7-200 PPI 與 S7-300 MPI 穿透功能設定 .....	29-10
29.6.	乙太網路穿透 .....	29-15

## 29.1. 概要

穿透通訊功能允許 PC 上的應用程式透過 HMI 直接控制 PLC；此時 HMI 的功能類似轉接器。

序列埠穿透功能包含以下兩種模式：

- 乙太網路
- 序列埠

點擊 Utility Manager 的 [序列埠穿透] 按鈕，即可檢視這兩種模式的設定內容。

## 29.2. 序列埠穿透的乙太網路模式

### 29.2.1. 安裝虛擬序列埠驅動程式的步驟

在使用 [乙太網路] 穿透通訊功能前，要先安裝虛擬序列埠驅動程式。

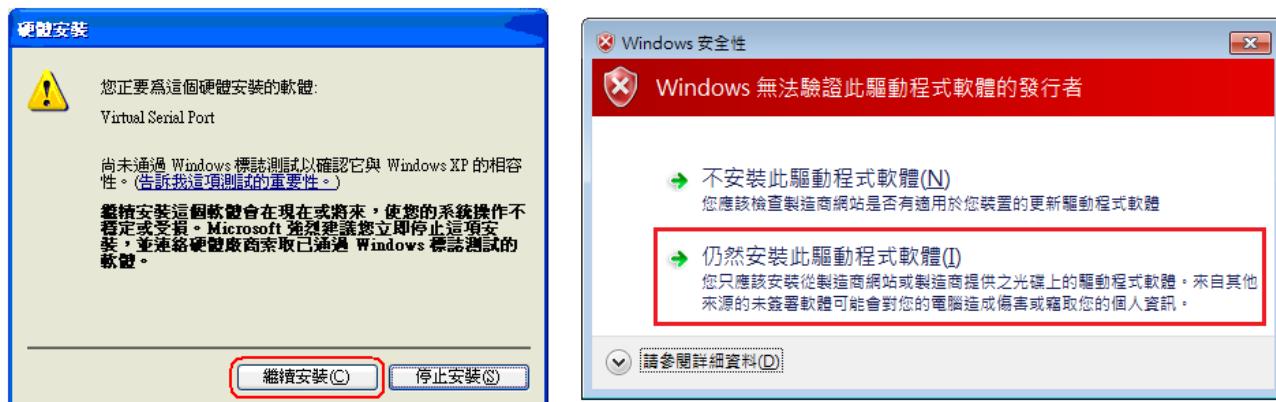
注意：請保持 Windows Update 更新至最新補丁，以確保虛擬序列埠驅動程式能正常運作。

1. 開啟 Utility Manager 檢視目前驅動程式的安裝狀態，若畫面顯示 [請安裝虛擬序列埠驅動程式]，請點擊 [安裝驅動] 按鈕。



2. 在安裝驅動程式的過程中可能被要求確認安裝，請選擇 [繼續安裝]。

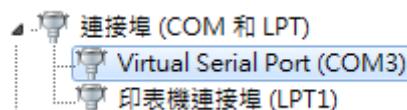
## 穿透過訊功能



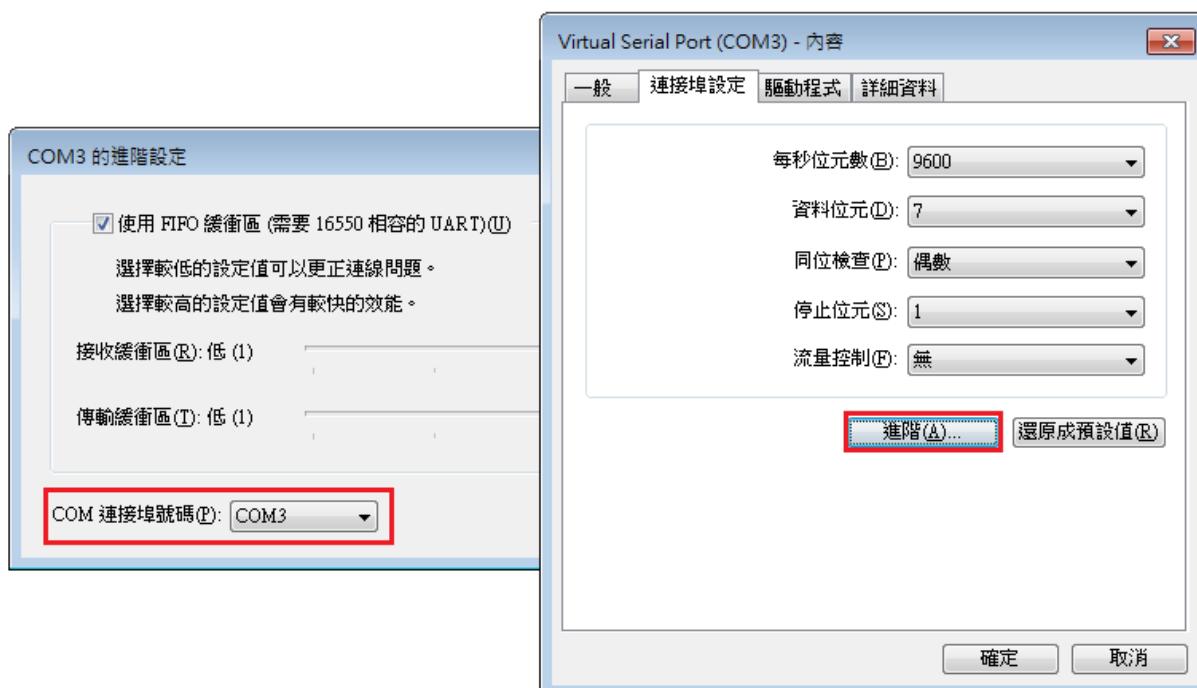
- 在完成安裝程序後，原先顯示 [請安裝虛擬序列埠驅動程式] 的位置將顯示目前所使用的虛擬序列埠號。

### 29.2.2. 更改虛擬序列埠的步驟

- 在 [裝置管理員] 中可發現已安裝完成的 [Virtual Serial Port]。



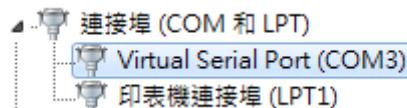
- 若要更改虛擬序列埠號，只需進入 [Virtual Serial Port] 的內容，並選擇 [連接埠設定] 下的 [進階]，即可更改虛擬序列埠號。



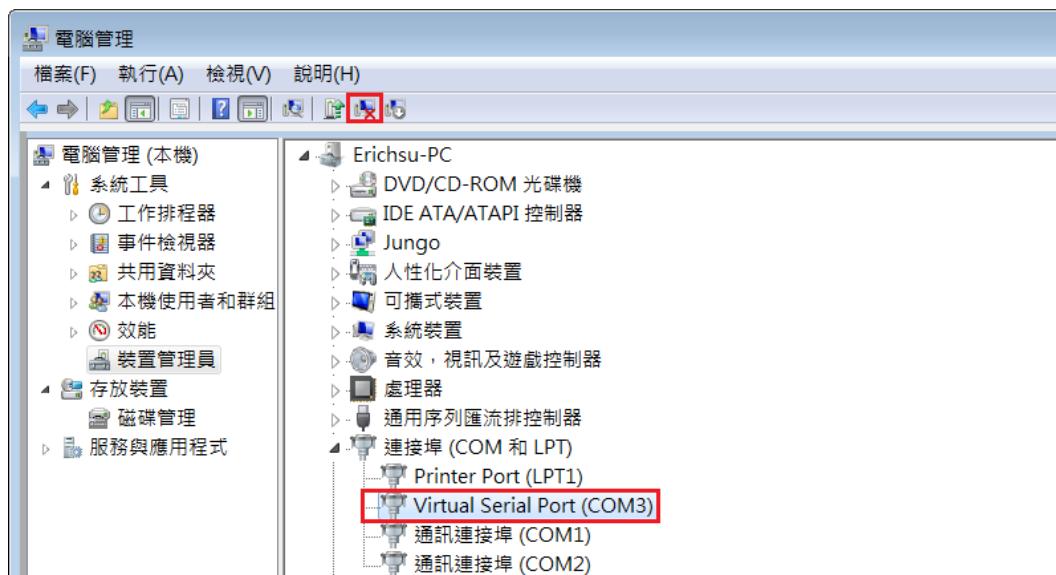
### 29.2.3. 移除虛擬序列埠的步驟

- 在 [裝置管理員] 中可發現已安裝完成的 [Virtual Serial Port]。

穿透通訊功能



- 若要移除虛擬序列埠號，請先點選欲刪除的 [Virtual Serial Port]，再點選 [裝置管理員] 視窗上方的 [移除安裝] 按鈕。



- 按下 [確定] 鍵，即可移除指定的虛擬序列埠。



#### 29.2.4. 更新虛擬序列埠驅動程式的步驟

- 在 [裝置管理員] 中可發現已安裝完成的 [Virtual Serial Port]。



- 若要更新虛擬序列埠驅動程式，請先點選欲更新的 [Virtual Serial Port]，再點選 [裝置管理員] 視窗上方的 [更新安裝] 按鈕。

## 穿透通訊功能



3. 使用瀏覽器選擇欲更新驅動程式所在位置，點選 [下一步] 後完成安裝。



### 29.2.5. 乙太網路模式設定

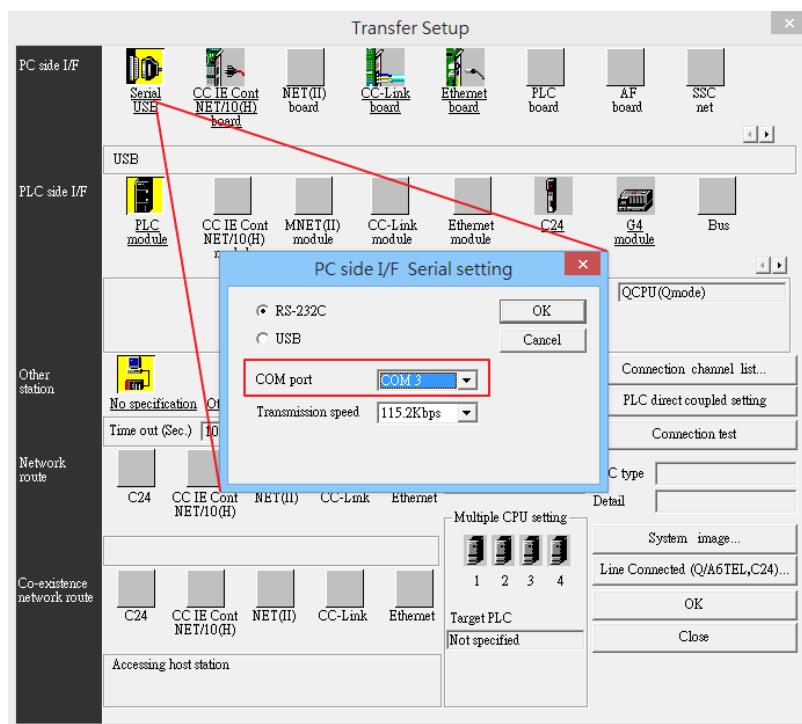
在安裝完成虛擬序列埠驅動程式後，只需依照下面步驟即可使用網路穿透通訊功能。

1. 請設定連接 PLC 的 HMI IP 位址。
2. 指定 HMI 通訊埠、HMI 連接 PLC 的序列埠與屬性。
3. 完成所有設定後需按下 [套用] 按鈕，所有屬性才會生效。

## 穿透通訊功能



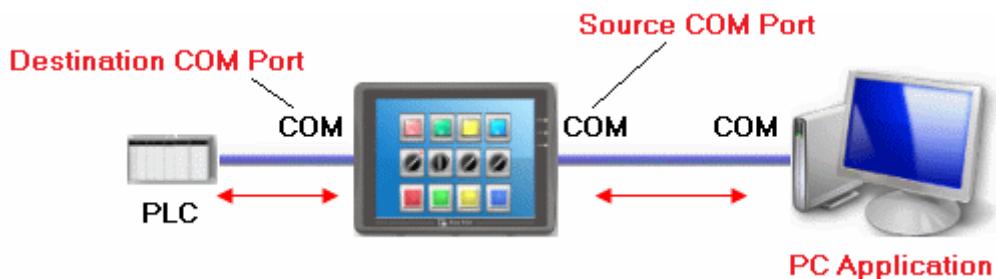
- 在執行 PC 上的應用程式時，所使用的序列埠號需指向虛擬序列埠號。以 Mitsubishi 的應用程式為例，若此時的虛擬序列埠為 COM 3，則在 [PC side I/F Serial setting] 視窗中的 [COM port] 需選擇 COM 3。



- 完成上述各項設定後，使用者在執行 PC 上的 PLC 應用程式時，HMI 會自動切換為穿透通

訊模式，此時可以將應用程式視為直接使用虛擬序列埠控制 PLC。在關閉應用程式時，HMI 也會自動關閉穿透通訊模式。

### 29.3. 序列埠穿透的序列埠模式



【數據來源串口】是指 HMI 與 PC 的連接埠。

【數據目標串口】是指 HMI 與 PLC 的連接埠。

在使用 [序列埠] 穿透通訊功能前，需先正確設定這兩個串口的屬性。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

#### 29.3.1. 序列埠設定

啟用 [序列埠] 穿透通訊功能的方式有兩種。

- 使用 Utility Manager
- 使用系統暫存器

LW-9901 設定數據來源串口 (1 ~ 3: COM 1 ~ COM 3)

LW-9902 設定數據目標串口 (1 ~ 3: COM 1 ~ COM 3)

#### 29.3.2. 使用 Utility Manager

1. 在 Utility Manager 按下 [序列埠穿透] 按鈕並設定通訊參數如下圖所示。



設定	描述
<b>HMI IP</b>	需指定 HMI 的 IP 位址。
<b>讀取 HMI 通訊參數設定</b>	讀取 HMI 上數據來源串口與數據目標串口的各項設定值。在按下 [讀取 HMI 通訊參數設定] 按鈕後，所有通訊參數將被更新。
<b>數據來源 COM 埠(PC-&gt;HMI) / 數據目標 COM 埠(HMI-&gt;PLC)</b>	顯示與設定數據來源串口與數據目標串口的通訊參數。當點選 [開始穿透通訊]，將根據 [數據來源 COM 埠]、[數據目標 COM 埠] 中所設定的內容，執行穿透通訊功能。
<b>傳輸速率 / 數據位元 / 校驗 / 停止位元</b>	通常 [數據來源 COM 埠] 與 [數據目標 COM 埠] 中的這些設定需相同。[數據來源 COM 埠] 因為是連接到 PC，通訊模式通常選擇為 “RS-232” 即可；[數據目標 COM 埠] 因為接到 PLC，通訊模式需依照 PLC 的通訊設定，可選擇：“RS-232”、“RS-485 2W” 或 “RS-485 4W”。

### Note

- 若不需要穿透通訊功能，需點選 [結束穿透通訊] 來關閉穿透通訊功能，此時 HMI 才會重新開啟和 PLC 的通訊。

共有三種模式可顯示目前 HMI 的工作模式。

模式	描述
----	----

未知	在未讀取 HMI 的設定值前，所顯示的 HMI 工作狀態。
正常模式	在讀取 HMI 的設定值後，顯示的狀態。HMI 處在正常通訊狀態，不接受來自數據來源串口的任何數據。
穿透模式	HMI 目前正處在穿透模式狀態。此時 PC 上的應用程式可以透過數據來源串口直接控制連接在數據目標串口上的 PLC。

### 29.3.3. 使用系統暫存器

另一種啟動 HMI 穿透通訊功能的方式為直接更改系統暫存器 LW-9901 (數據來源串口) 與 LW-9902 (數據目標串口) 中的數據內容。當 LW-9901 與 LW-9902 中的數據符合下列條件時，HMI 將自動啟動穿透通訊功能。

- LW-9901 與 LW-9902 中的數據需為 1 ~ 3 (1 ~ 3 分別表示 COM 1 ~ COM 3)。
- LW-9901 與 LW-9902 中的數據不可相同。

如有需要更改各串口的通訊參數，只需更改各參數相對應的系統暫存器中的數據，並對 [LB-9030：更新 COM 1 通訊參數]、[LB-9031：更新 COM 2 通訊參數]、[LB-9032：更新 COM 3 通訊參數] 送出 ON 的訊號即可。



#### Note

- 若要關閉 HMI 的穿透通訊功能，只需將 LW-9901 與 LW-9902 中的數據更改為 0。
- 若使用的機型 COM1 的第 7、8 腳與 COM3 的 RX、TX 共用 (例如 eMT 系列、cMT 系列)，在 [裝置清單] 中無 COM3 裝置的情況下，第 7、8 腳會被作為 COM1 的 RTS、CTS 使用。要啟用 COM3 的 RX、TX，[系統參數設定] » [裝置清單] 中須添加使用 COM3 的 Master-Slave 驅動或 Free Protocol 驅動。

### 29.4. 穿透通訊控制

一般來說，開啟穿透模式時，HMI 會關閉與 PLC 間的通訊，直到穿透模式結束。然而，特定的 PLC 驅動程式於穿透模式下可支援 HMI- PLC, PC-HMI 同時通訊。



要查詢支援的 PLC 驅動，請參閱《PLC 連結手冊》的相關章節。

同時通訊功能可利用系統暫存器 LW-9903 進行控制。

LW-9903 數值	描述
0 (預設)	正常模式。執行穿透功能時，HMI- PLC, PC-HMI 可

以同時通訊。

2

執行穿透功能時，將停止 HMI 與 PLC 間的通訊。



### Note

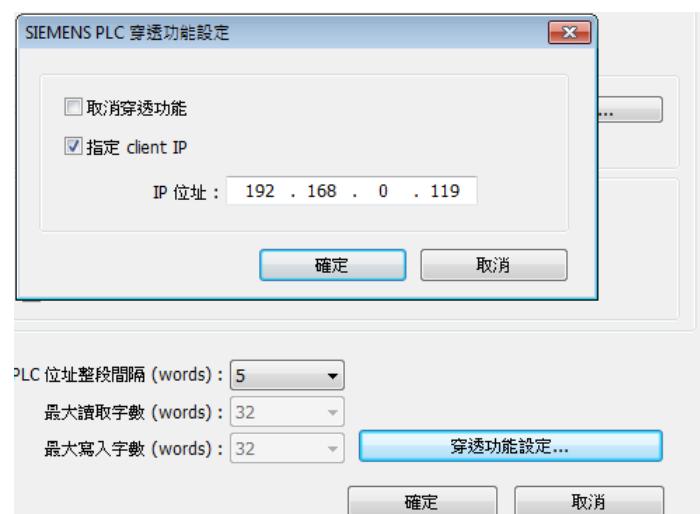
- 因序列埠通訊速度受限，可設定 LW-9903 為 2 關閉此功能來加快上傳/下載 PLC 程式的速度。

## 29.5. SIEMENS S7-200 PPI 與 S7-300 MPI 穿透功能設定

EasyBuilder Pro 支援 SIEMENS S7-200 PPI 與 S7-300 MPI 穿透功能。

### 29.5.1. EasyBuilder Pro 設定

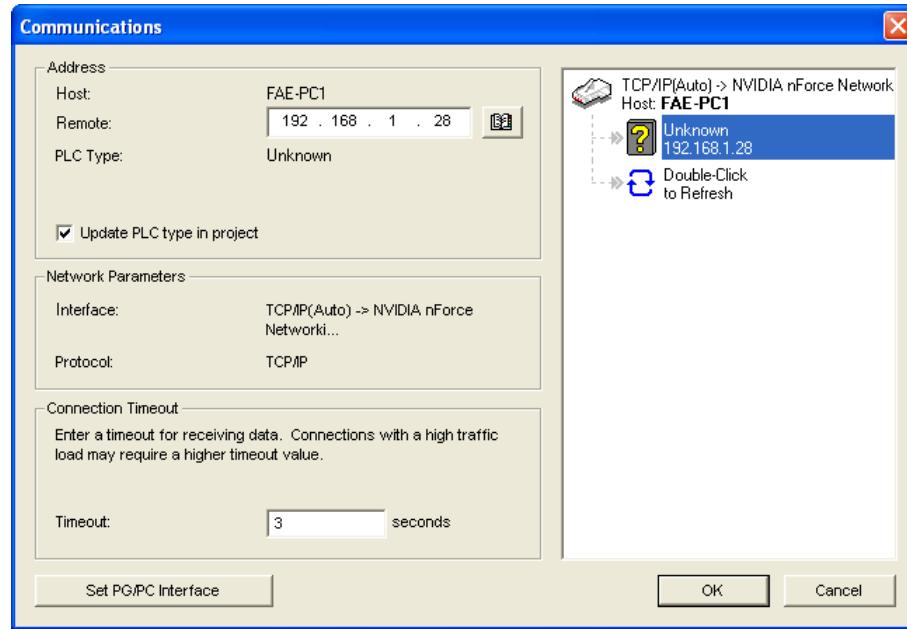
自 EasyBuilder Pro [系統參數設定] » [設備清單] 中新增 Siemens S7-200 PPI 或 S7-300 MPI 時，點選 [穿透功能設定] 會顯示以下對話窗。



設定	描述
取消穿透功能	勾選擬此項目將關閉穿透功能，預設為未勾選擬。
指定 client IP	當開啟穿透功能時，指定連接到 HMI 的客戶端 IP 位址。

### 29.5.2. S7-200 PPI 連線方式

確認開啟穿透功能的 HMI 已經開機並連上網路後，開啟 STEP7 Micro/Win，進入 Communications 設定視窗，搜尋欲執行穿透功能的 HMI 的 IP 位址，直接對從連線，即可以開始穿透通訊。

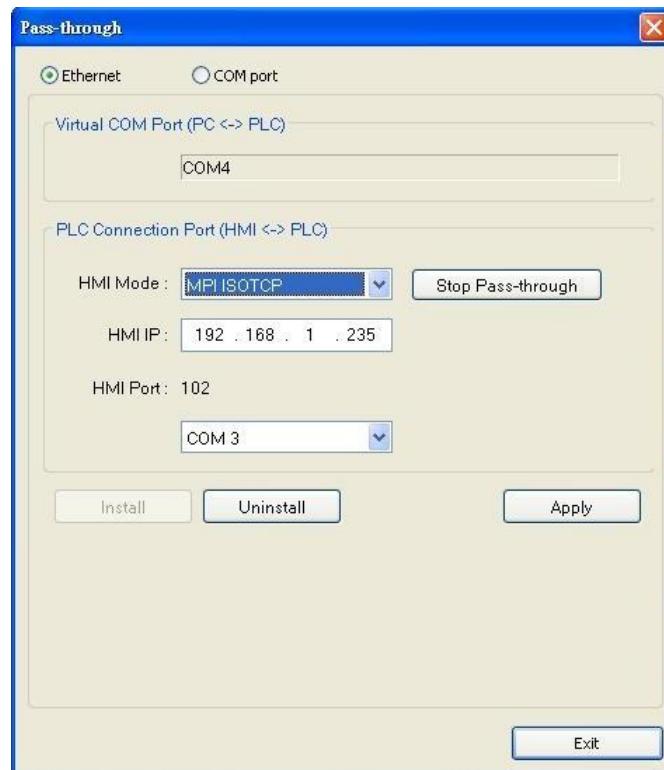


### 29.5.3. S7-300 MPI 連線方式

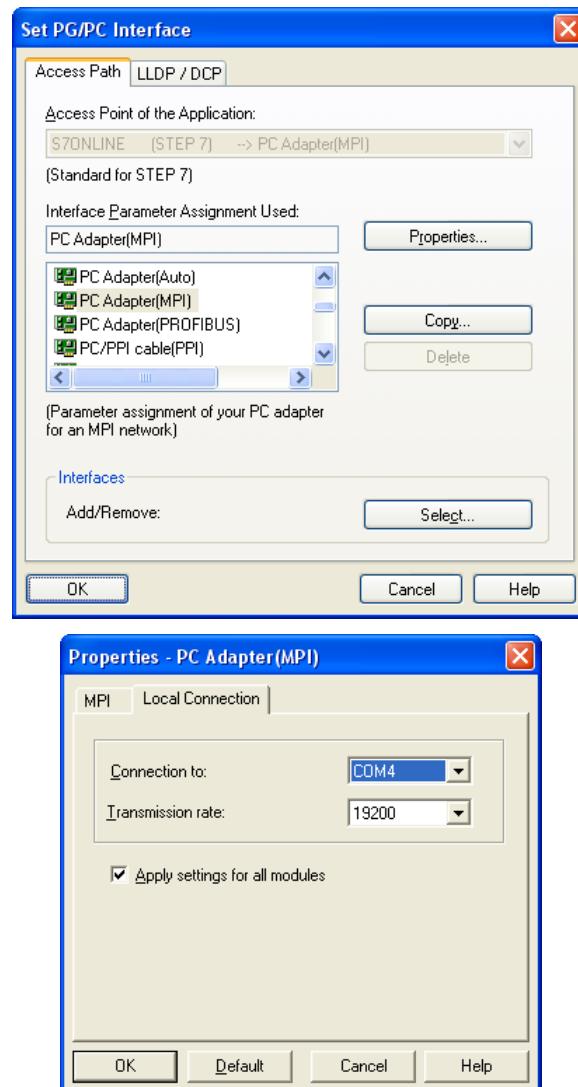
可透過虛擬序列埠或是乙太網路連線。

#### 29.5.3.1. 透過虛擬序列埠

- 在 Utility Manager 中，開啟 [穿透連接]，選擇 MPI ISOTCP 安裝 MPI 專用虛擬序列埠程式。同時設定 HMI IP 和 HMI 連接 PLC 的連接埠，並啟動穿透。



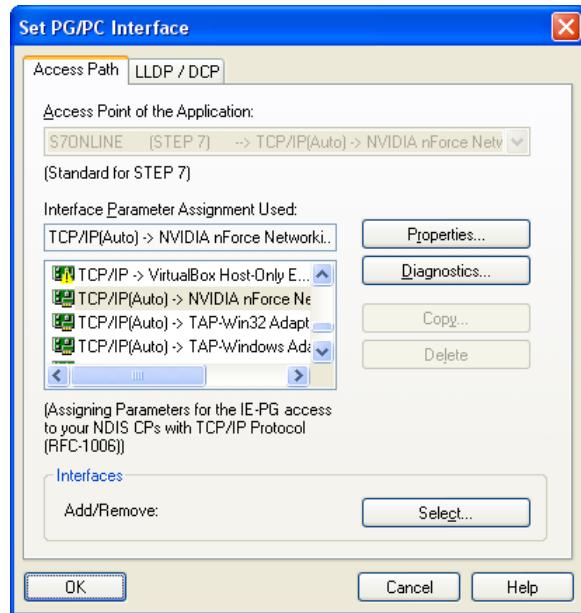
2. 在 STEP 7 中，進入 [Option] » [Set PG/PC Interface]，確認界面卡是 PC Adapter(MPI)，並且進入 [Properties] 中設定和虛擬序列埠同樣的連接埠，範例圖為 COM4。



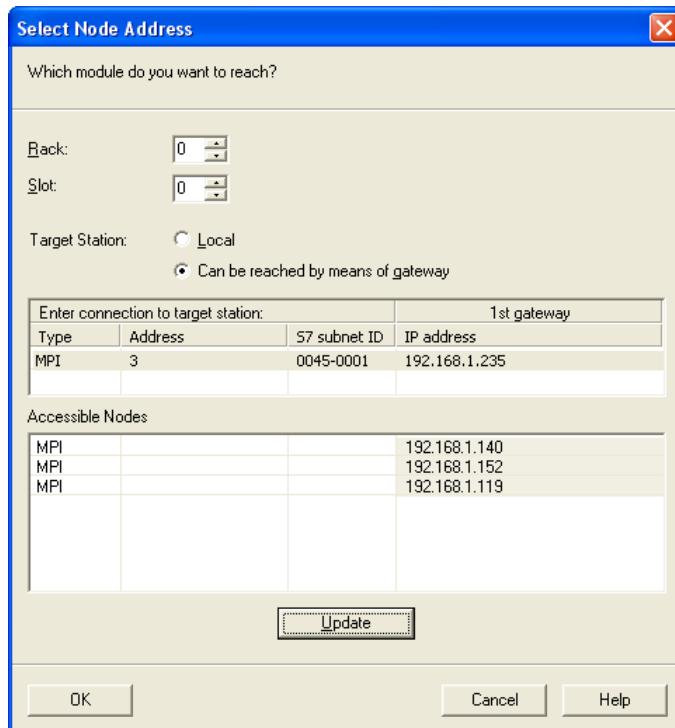
3. 完成以上設定，即能在 STEP 7 透過 HMI 穿透上傳/下載 PLC 程式。

### 29.5.3.2. 透過乙太網路連線

1. 在 STEP 7 中，進入 [Option] » [Set PG/PC Interface]，如下圖所示欄位選擇 “TCP/IP(Auto) -> 使用中電腦網路介面卡名稱”。



2. 在 [PLC] » [Update station to PG] 中，Target Station 選擇 [Can be reached by means of gateway]，並在下方欄位中，依左到右選擇 MPI 並輸入 PLC 站號、S7 Subnet ID、HMI IP 位址。按下 OK 後，S7 即能透過穿透模式，經由 HMI 將 PLC 程式上傳至 STEP 7 中。



#### 29.5.4. SIEMENS 穿透相關暫存器

系統暫存器 LW-10850 至 LW-10864 用於表示或設定 SIEMENS 穿透狀態。

 關於各暫存器功能，請參考《22 位址暫存器》。

執行穿透時，若錯誤發生，LW-10863 會提示錯誤。LW-10864 則會顯示錯誤碼，下表解釋錯誤碼所表示的內容及可能原因。(以下客戶端通常意指 STEP7 PLC 程式)

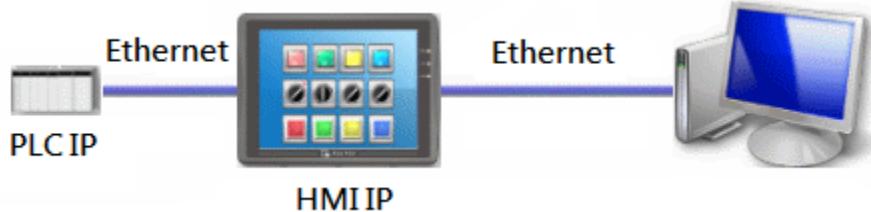
錯誤碼	描述	錯誤原因
<b>0</b>	執行正常。	
<b>1</b>	禁止 client 端連接到 HMI。	因 HMI 已在執行穿透功能，不再接受其他 client 端的命令。
<b>2</b>	禁止 client 端連接到 HMI。	當 LW-10850 被設為 1 時，欲連接到 HMI 的 client 端 IP 位址與 LW-10858 ~ LW-10861 所定義的不同。
<b>3</b>	通訊協議錯誤。	LW-10853 設定錯誤。
<b>4</b>	PLC 站號錯誤。	LW-10852 所指定站號的 PLC 不存在。
<b>5</b>	通訊延時。	PLC 未正確連接。
<b>6</b>	通訊忙碌。	PLC 不接受穿透命令，請檢查 PLC 的設定。
<b>7</b>	穿透命令錯誤。	穿透功能架設環境錯誤。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 29.6. 乙太網路穿透

乙太網路穿透通訊功能是在 PLC 與 PC 都使用乙太網路跟 HMI 連接時，允許 PC 上的應用程式透過 HMI 直接控制 PLC；僅支援 cMT / cMT X 系列人機。



- 開啟 Utility Manager 後選擇 cMT / cMT X Series 並開啟 [分析測試工具] 頁籤，即可找到 [乙太網路穿透]。



- 點選 [乙太網路穿透] 按鈕並填入正確通訊參數建立連線。



設定	描述
HMI IP	需指定 HMI 的 IP 位址。
HMI 連接埠號	需填入與 HMI 連接的正確埠號，預設值為 8000。
穿透 IP	需指定使用穿透方式控制的裝置的 IP 位址。
狀態	設定完成後點選 [連接] 可在此確認連線狀態。

### Note

- 一台 HMI 只能設定一組乙太網路穿透 IP。
- 系統參數 LB-9044 (禁止遠端控制) 開啟時無法穿透。

# 30. 工程檔案保護功能

本章節說明工程檔案保護功能的相關設定。

30.1. 概要 .....	30-2
30.2. EXOB/CXOB 密碼 .....	30-2
30.3. 禁止反編譯 .....	30-3
30.4. 取消 HMI 上傳功能 .....	30-3
30.5. 工程檔案識別碼 .....	30-3
30.6. MTP 檔案密碼 .....	30-4

### 30.1. 概要

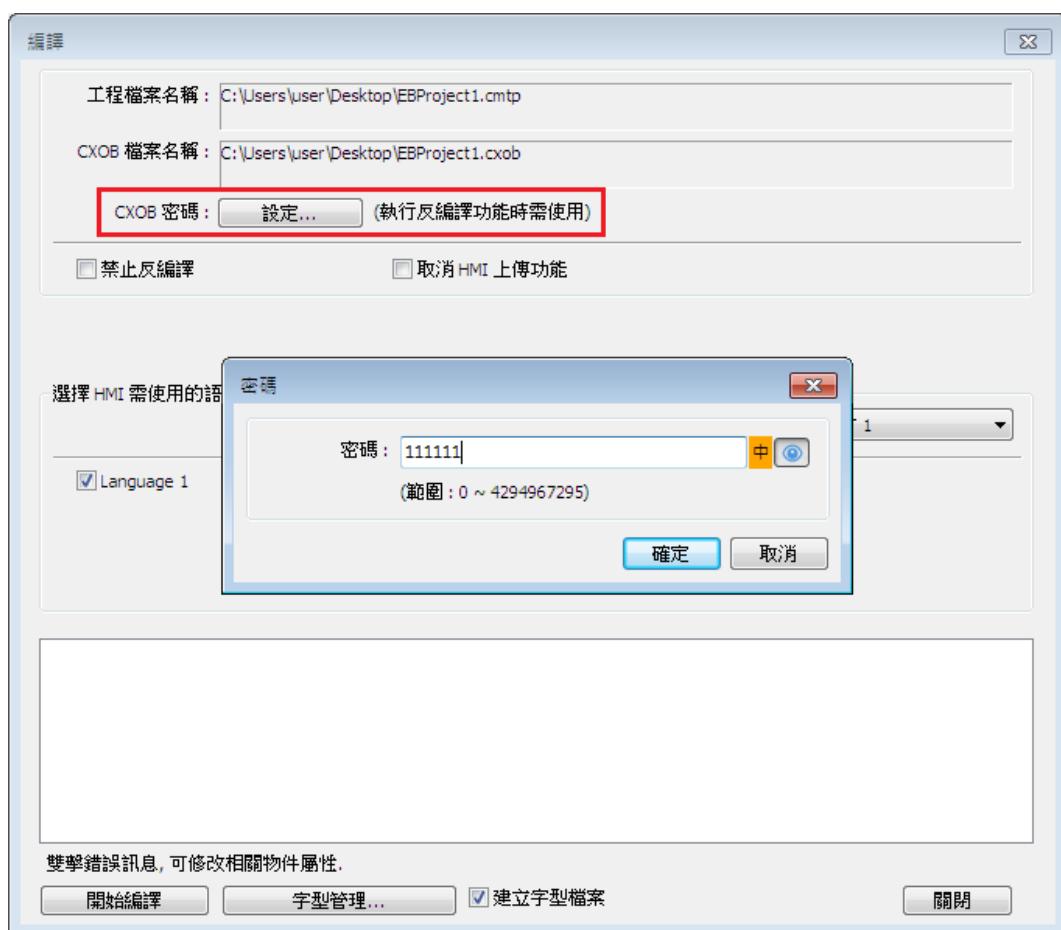
EasyBuilder Pro 提供許多工程檔案的安全加密功能，完整保護檔案。



- 以下介紹的安全加密功能，若不慎忘記密碼時，因已由用戶自行加密，所以原廠也無法解開，請妥善保管密碼。

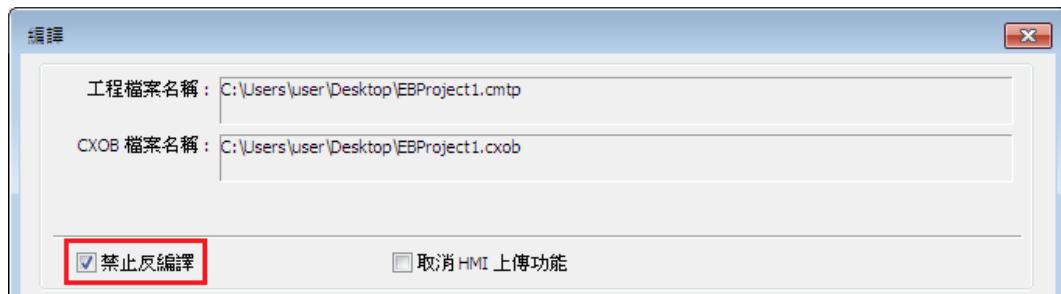
### 30.2. EXOB/CXOB 密碼

在編輯完成工程檔案 (.emtp/.cmt) 後，可透過系統提供的編譯功能，將 .emtp/.cmt 工程檔案編譯為下載至 HMI 所需的 .exob/.cxob 檔案。用戶可在編譯視窗設定 [EXOB/CXOB 密碼] (範圍: 0 ~ 4294967295)，若之後要將此 .exob/.cxob 反編譯成 .emtp/.cmt 工程檔案，必需輸入此密碼才能完成反編譯。若反編譯時密碼輸入錯誤三次，請重新啟動 EasyBuilder Pro。



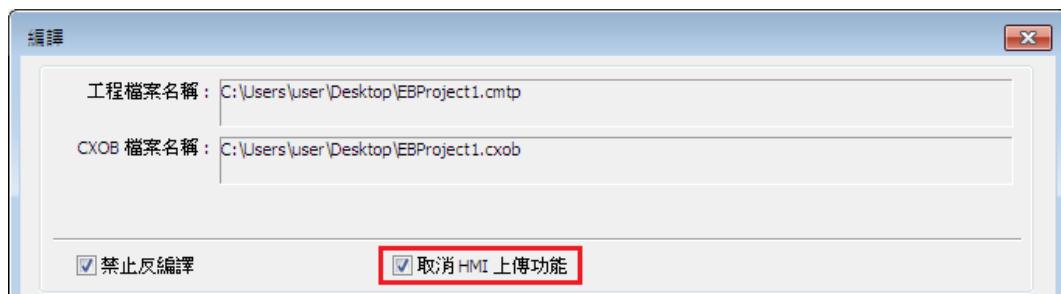
### 30.3. 禁止反編譯

在編輯完成工程檔案 (.emtp/.cmtp) 後，可透過系統提供的編譯功能，將 .emtp/.cmtp 工程檔案編譯為下載至 HMI 所需的 .exob/.cxob 檔案。用戶可在編譯視窗設定 [禁止反編譯]，系統將忽略 [EXOB/CXOB 密碼] 的設定，且此 .exob/.cxob 檔案將無法反編譯回 .emtp/.cmtp 工程檔案。



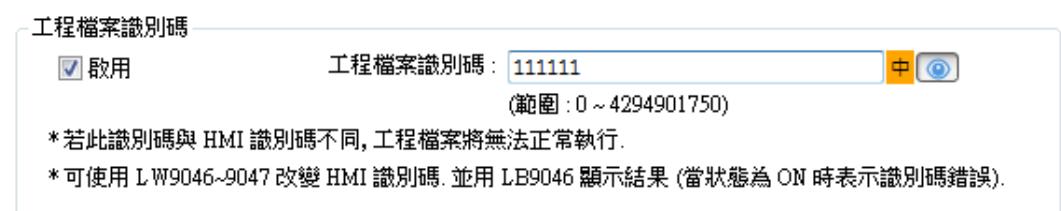
### 30.4. 取消 HMI 上傳功能

EasyBuilder Pro 提供系統暫存器 LB-9033 作為禁止 HMI 上傳的功能。當此位址被設定為 ON 時，即上傳後得到的 .exob/.cxob 檔案大小將為 0 位元組而無法反編譯。此外，當變更任何設定時，必須重新啟動 HMI 以更新設定值。



### 30.5. 工程檔案識別碼

用戶的工程檔案可以被限制只能在特定的 HMI 上執行，下圖為 [系統參數設定] » [一般屬性] 頁籤中 [工程檔案識別碼] 的設定畫面。



## 工程檔案保護功能

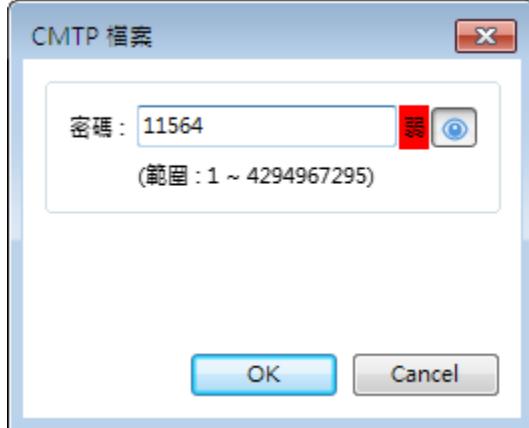
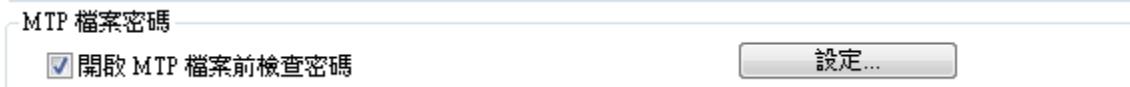
當啟用 [工程檔案識別碼] 功能時，可設定 [工程檔案識別碼] (範圍: 0 ~ 4294901750)，且需搭配系統暫存器 LW-9046 及 LW-9047 (32-bit) 設定 HMI 的 [人機識別碼]，其數據無法被讀取或遠端寫入。而編譯後所得到的 .exob/.cxob 檔案只能在 [人機識別碼] 與 [工程檔案識別碼] 相同的 HMI 上執行。若 [人機識別碼] 與 [工程檔案識別碼] 不相同時，LB-9046 的狀態將被設定為 ON。當變更 [人機識別碼] 設定時，必須重新啟動 HMI 以更新設定值。

### Note

- 若 [人機識別碼] 與 [工程檔案識別碼] 不相同時，HMI 與 PLC 之間將無法通訊。
-  請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 30.6. MTP 檔案密碼

在編輯完成工程檔案 (.emtp/.cmtip) 後，可以設定保護 .emtp/.cmtip 工程檔案的密碼，請至 [系統參數設定] » [使用者密碼] 頁籤設定此功能 (範圍: 1 ~ 4294967295)。啟用此功能後，每次欲開啟該 .emtp/.cmtip 工程檔案時，須先輸入正確的密碼後才能開啟並編輯。



### Note

- 在使用 “窗口複製” 功能時，若欲複製的來源工程檔案設有 MTP 密碼保護時，須先輸入正確的密碼後，系統才能執行窗口複製功能。

# 31. cMT Diagnoser

本章節說明 cMT / cMT X 系列支援物件狀態的監視與操控以及 cMT 診斷器的相關功能。

31.1.	概要 .....	31-2
31.2.	如何啟用 cMT 診斷器 .....	31-2
31.3.	cMT 診斷器使用者介面 .....	31-3
31.4.	補充說明 .....	31-19

### 31.1. 概要

cMT 診斷器可用於監控物件的位址狀態、查詢通訊封包狀態，及 MQTT 與巨集的除錯。

### 31.2. 如何啟用 cMT 診斷器

cMT 診斷器可分成模擬使用與 HMI 使用，而 HMI 使用又細分成透過 cMT Viewer 開啟 cMT 診斷器與直接使用 cMT 診斷器共三種不同使用狀況。以下依序說明：

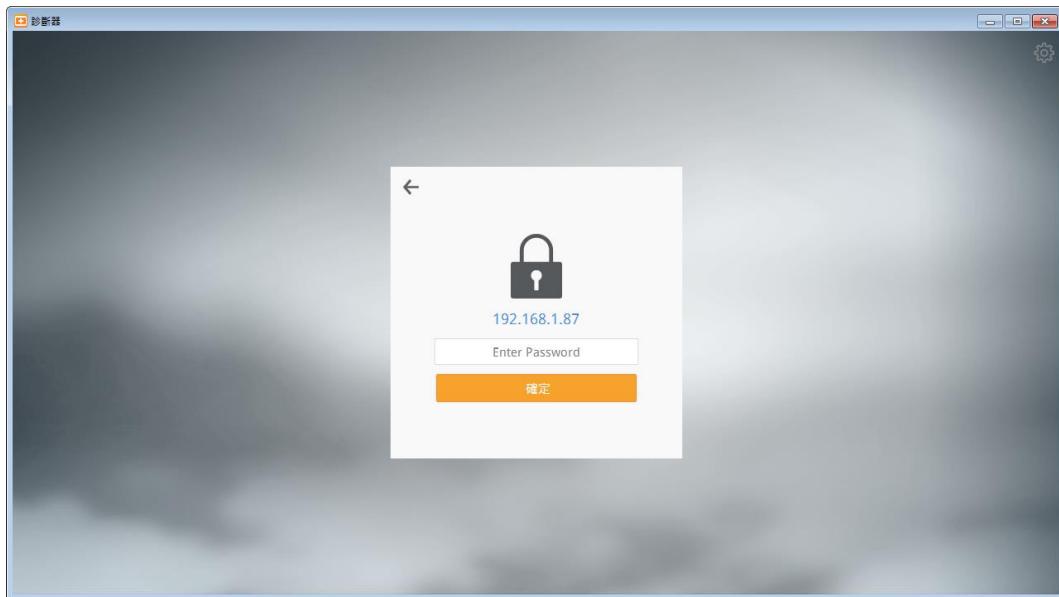
1. 於連線模擬或離線模擬階段時，在 cMT Viewer 的模擬畫面按下右鍵點選診斷器。



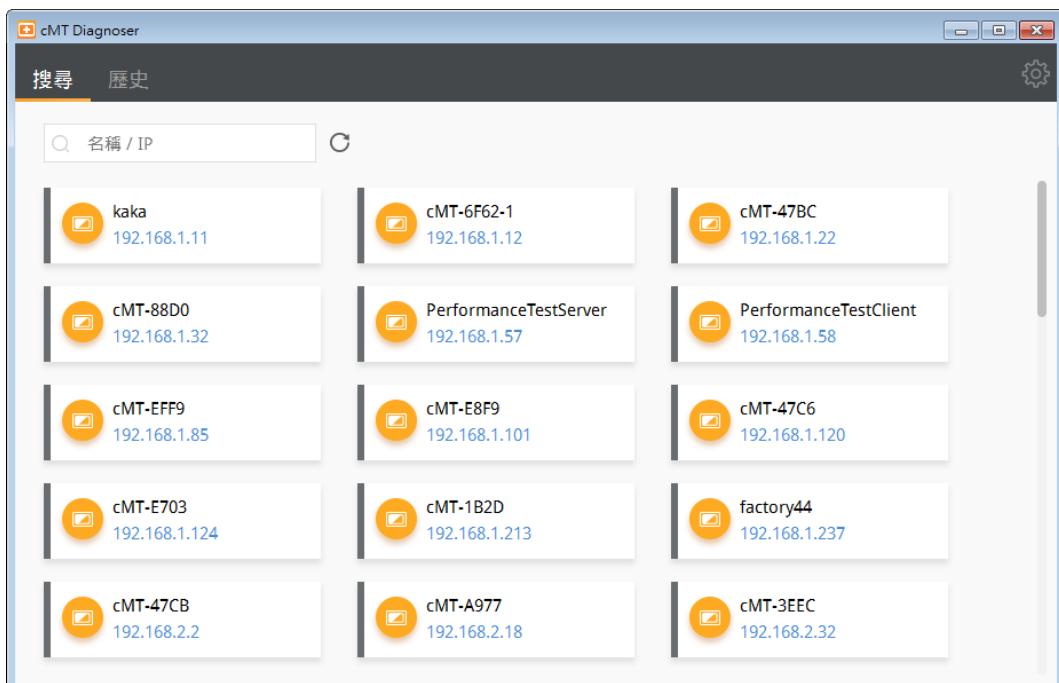
2. 如希望 HMI 可以使用 cMT 診斷器連接，首先必須在工程檔案的 [系統參數設定] » [遠端] 頁籤啟用 cMT 診斷器。



下載工程檔案至 HMI 後透過 cMT Viewer 開啟 cMT 診斷器。若沒勾選 [不須密碼即可登入] 的話，就必須先輸入預設的密碼才能執行 cMT 診斷器。



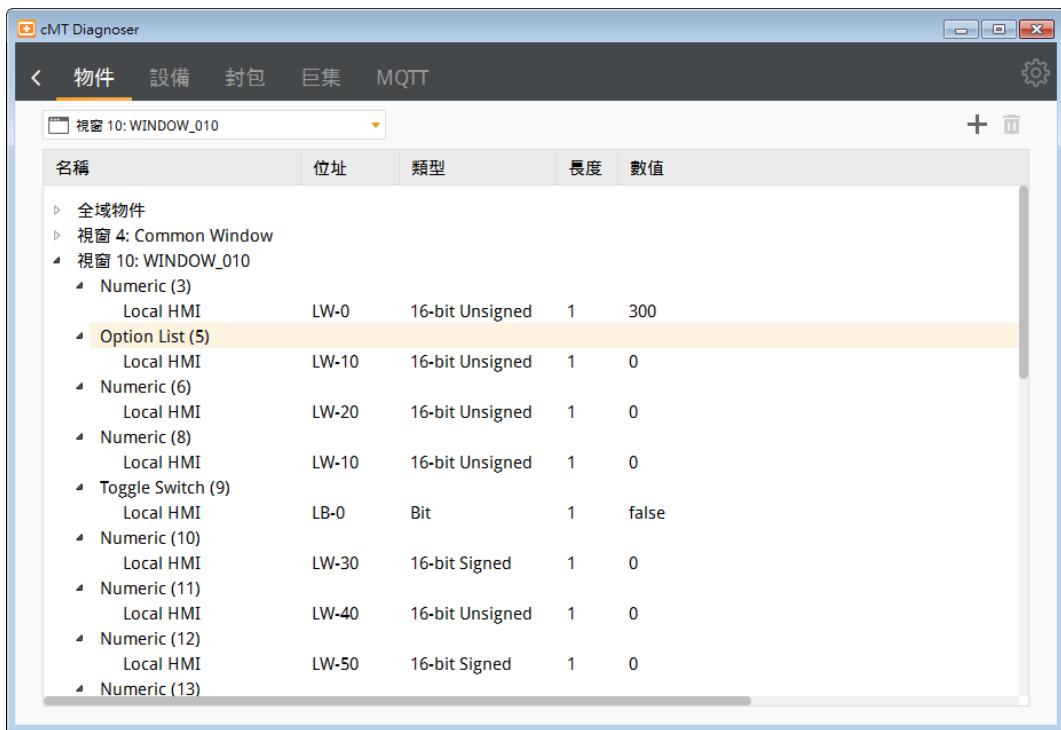
3. 在 EB Pro 安裝資料夾內執行 cMTDiagnoser.exe，是 cMT 診斷器的獨立程式。啟用後能搜尋欲連接的 HMI。



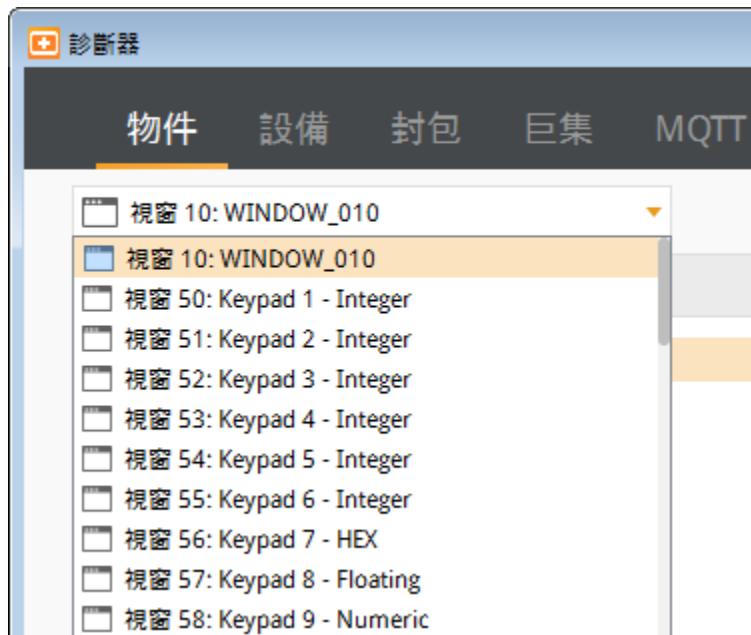
### 31.3. cMT 診斷器使用者介面

#### 31.3.1. 物件設定

1. 此畫面顯示該視窗上存在的所有物件、位址屬性與其狀態數值的內容。



2. 由下拉式選單可迅速切換不同的視窗。



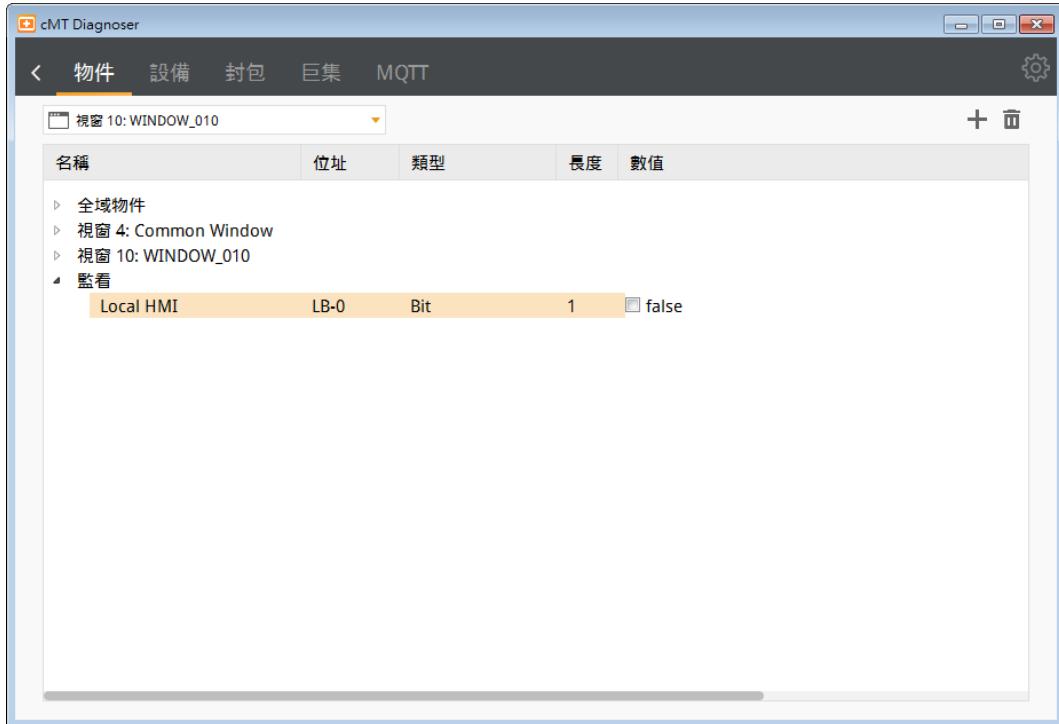
3. 在 cMT 診斷器視窗上點擊物件時，畫面上的對應物件會有額外標示。



4. 在 cMT 診斷器上雙擊物件時，則可以動態修改該物件的數值。



5. 在 cMT 診斷器上可增減欲監看的位址。點擊 新增位址，選擇特定監看位址後點擊 即可刪除。



### 31.3.2. 設備設定

1. 此畫面顯示與此 HMI 通訊的設備狀態和參數。



### 31.3.3. 封包設定

1. 設定好參數按下  可以觀察 HMI 和 PLC 之間的通訊。



項目	描述
<b>封包類型</b>	<b>全部</b> 在通訊記錄區顯示全部通訊的封包。 <b>讀取</b> 在通訊記錄區只顯示讀取命令的封包。 <b>寫入</b> 在通訊記錄區只顯示寫入命令的封包。 <b>輪詢</b> 在通訊記錄區只顯示輪詢的封包。
<b>設備</b>	<b>全部</b> 顯示本地 HMI 和 PLC 的資訊。 <b>Local HMI</b> 只顯示本地 HMI 的資訊。 <b>PLC</b> 只顯示 PLC 的資訊。
<b>站號</b>	選擇欲顯示之 PLC 站號。 (當 [設備] 選擇全部時無法使用此功能)
<b>位址類型</b>	使用者可以選擇全部或是其中的設備位址類型顯示在畫面上。 (當 [設備] 選擇全部時無法使用此功能)
<b>關鍵字</b>	可過濾並顯示包含此關鍵詞的位址資訊。

與設備的通訊若有異常，在通訊頁面會以紅色標示，結果欄位顯示錯誤原因。

No.	類型	PID	設備	站號	位址	長度	索引	時間	結果
46	P	200041	Local HMI	-	PLW-8950	1	-	0	Success
45	P	200036	Local HMI	-	LB-9039	3322	-	17	Success
44	P	200040	Local HMI	-	LB-8999	1	-	17	Success
43	P	200038	Local HMI	-	LW-0	150	-	33	Success
42	P	200037	Local HMI	-	LB-0	1	-	33	Success
41	P	200041	Local HMI	-	PLW-8950	1	-	0	Success
40	P	200039	DELTA DVP	1	D-0	1	-	1070	Ignore
39	P	200036	Local HMI	-	LB-9039	3322	-	16	Success
38	P	200040	Local HMI	-	LB-8999	1	-	33	Success
37	P	200041	Local HMI	-	PLW-8950	1	-	0	Success
36	P	200038	Local HMI	-	LW-0	150	-	16	Success

2. 輪詢封包顯示 HMI 定時發出詢問的封包，包含起始地址與長度等相關資訊。

封包 ID	設備	站號	位址	長度	索引
100309	Local HMI	-	LB-0	1	-
100310	Local HMI	-	LW-0	1	-
200036	Local HMI	-	LB-9039	3322	-
200037	Local HMI	-	LB-0	1	-
200038	Local HMI	-	LW-0	150	-
200039	DELTA DVP	1	D-0	1	-
200040	Local HMI	-	LB-8999	1	-
200041	Local HMI	-	PLW-8950	1	-

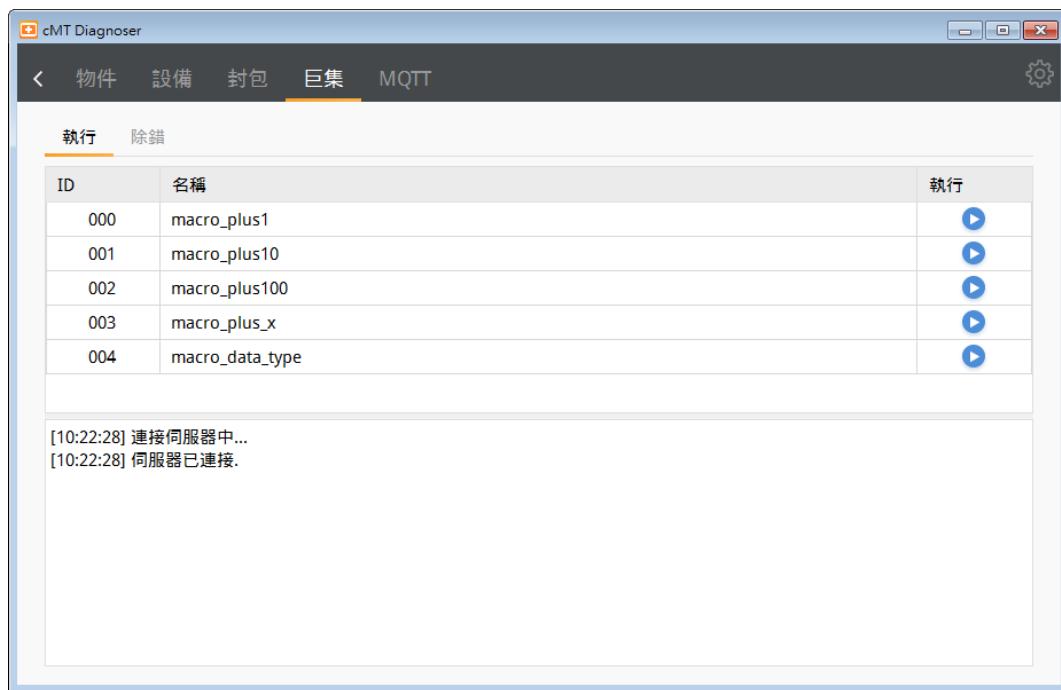
項目	描述
封包 ID	封包的 ID 編號，可由通訊記錄區看出有問題的封包之物件。
設備	顯示 HMI 和 PLC 的設備。
站號	顯示 PLC 的站號。
位址/長度	顯示設備類型位址及封包內的位址長度。

**索引**

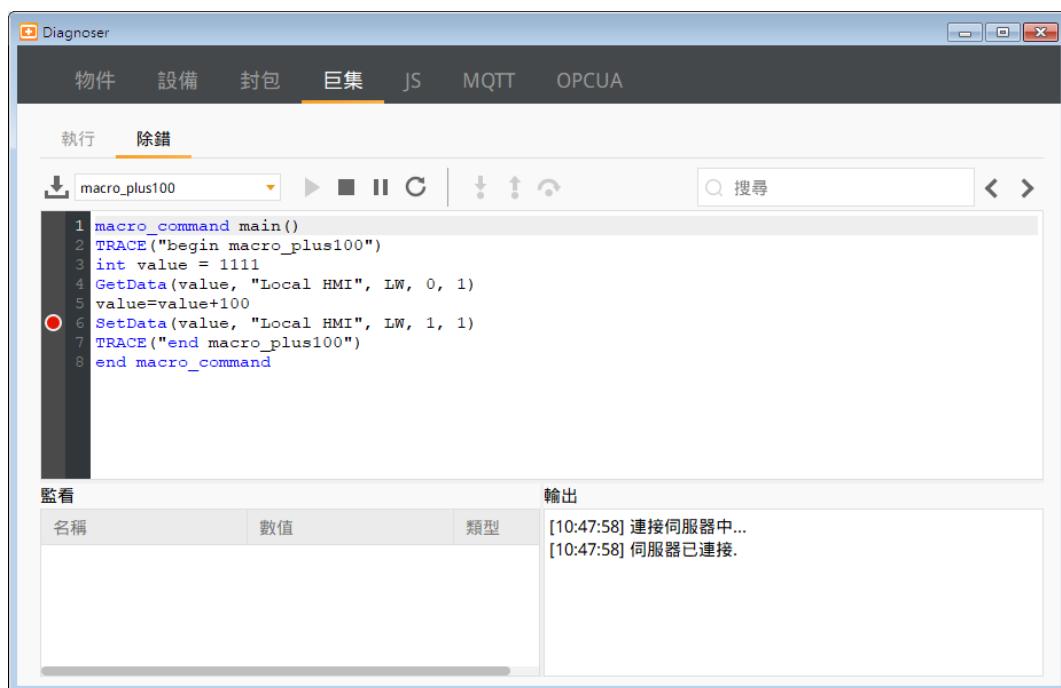
顯示物件所使用的索引暫存器編號。

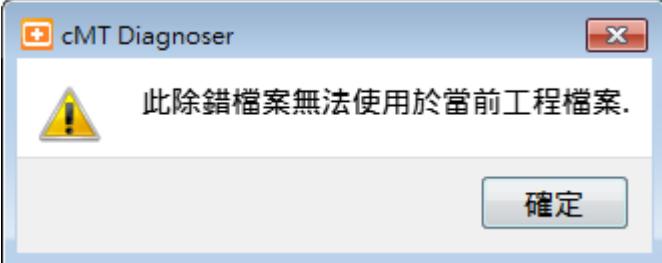
### 31.3.4. 巨集設定

- 可以在此執行並測試巨集。



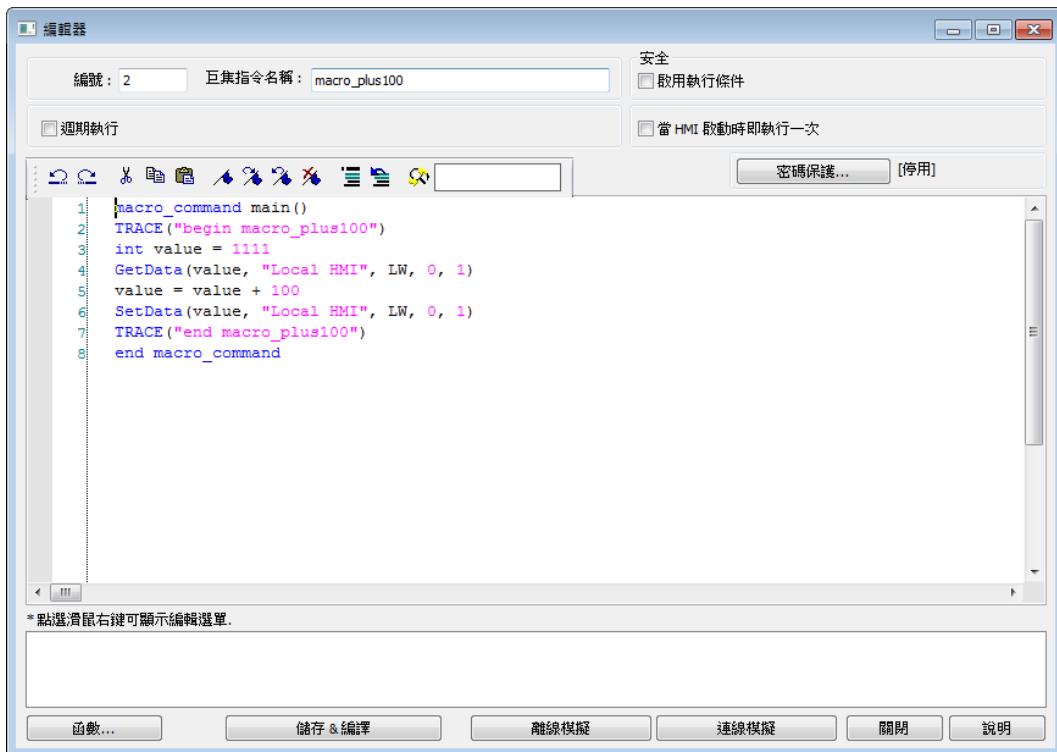
- 巨集除錯功能。



項目	描述
載入巨集	讀取巨集除錯檔案。巨集除錯檔案可由 EasyBuilder Pro 編譯工程
除錯檔案	檔案時產生，產生位置與編譯檔為相同路徑，副檔名為.debug。若載入不匹配的巨集除錯檔案，將會出現下圖警示。
	
開始除錯	開始巨集除錯功能或是從中斷點往下繼續執行。
停止除錯	停止巨集除錯功能。
暫停	手動暫停目前巨集執行進度。在巨集的行數前方點擊可以產生中斷點，巨集執行到中斷點會暫停並等待下一個執行命令。
重啟模擬	使用編輯巨集頁面上的連線/離線模擬時，可直接改寫巨集並使用重啟模擬完成一鍵重新模擬，按下後會有提示與確認視窗彈出。
	
進入	逐步執行。若下一步為函式，會進去函式內部。
跳離函式	可跳離目前執行所在的函式。
跳過	逐步執行。若下一步為函式，會執行函式所有內容，並回傳函式結果，但不逐行執行函式內的內容。
搜尋	可搜尋巨集中關鍵字。
監看	在除錯執行中，可以新增變數來觀察數據即時的變化。
輸出	顯示系統資訊與 Trace 函式輸出。

有三種方式開啟 cMT 診斷器巨集除錯功能，每一種方式皆有不同的使用權限以及操作方式，說明如下。

- A. 編輯巨集頁面上的連線/離線模擬。



按下模擬按鈕後自動產生巨集除錯檔案，軟體會關閉所有編輯巨集頁面並彈出 cMT Viewer 模擬頁面，同一時間 cMT 診斷器也一併執行並自動匯入巨集除錯檔案。

於除錯時，cMT Diagnoser 的巨集頁面亦可修改巨集內容。巨集除錯執行完畢後關閉 cMT Viewer，編輯巨集頁面會再度開啟並會確認是否要將在 cMT 診斷器上的變更更新到編輯巨集頁面上。

#### B. EasyBuilder Pro 的連線/離線模擬。

編譯時會自行產生巨集除錯檔案，當在 EasyBuilder Pro 執行模擬彈出 cMT Viewer 模擬頁面，必須使用右鍵開啟 cMT 診斷器，若系統有偵測到有巨集除錯檔案，則會自行匯入。此模式下的巨集只能除錯無法變更。

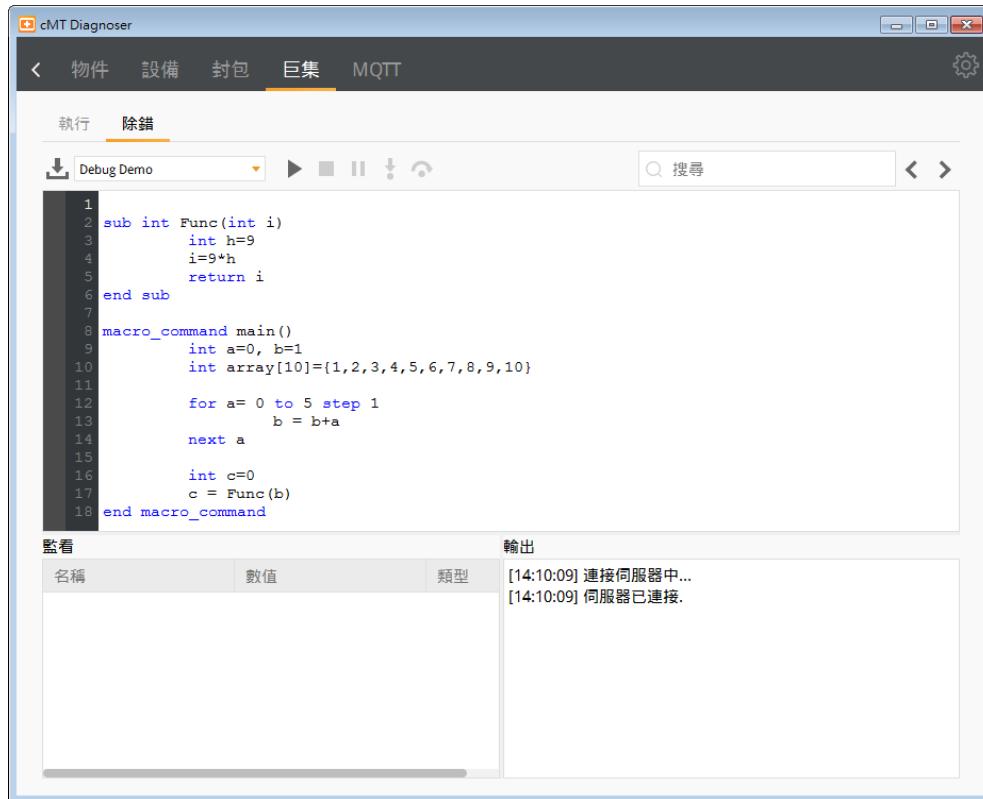
注意：工程檔案中至少要建立一個巨集指令，編譯後才會有巨集除錯檔案。

#### C. 將工程檔案下載至 HMI 後使用 cMT Diagnoser 操作。

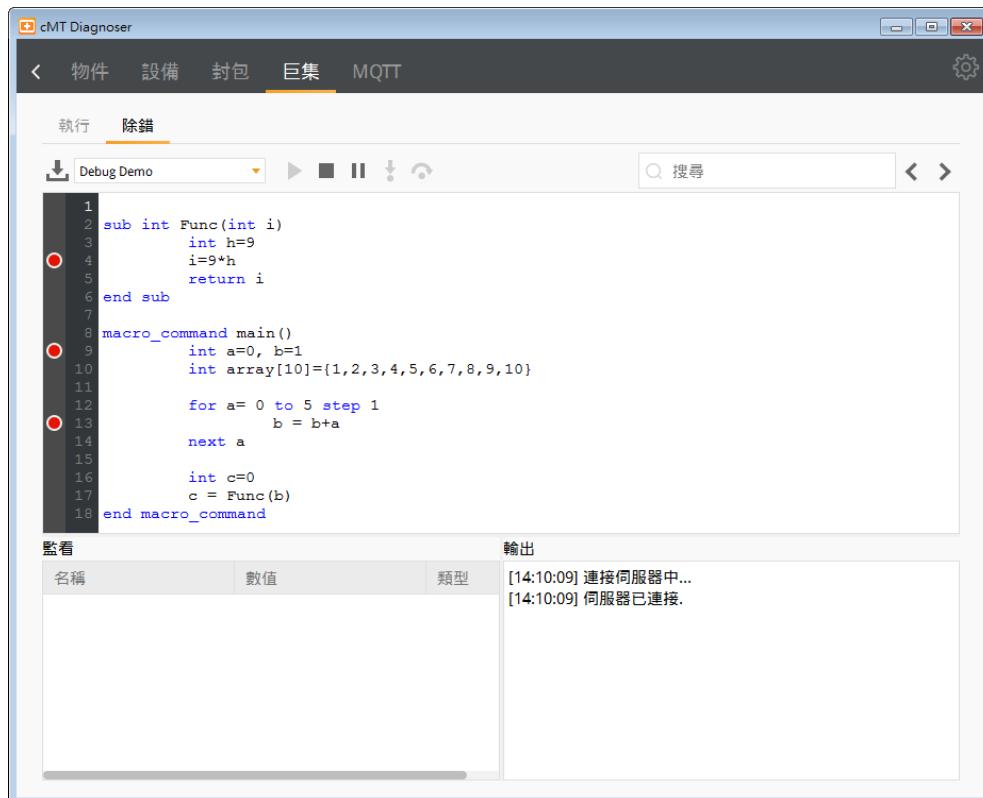
需要手動匯入巨集除錯檔案。此模式下的巨集只能除錯無法變更。

### 巨集除錯範例

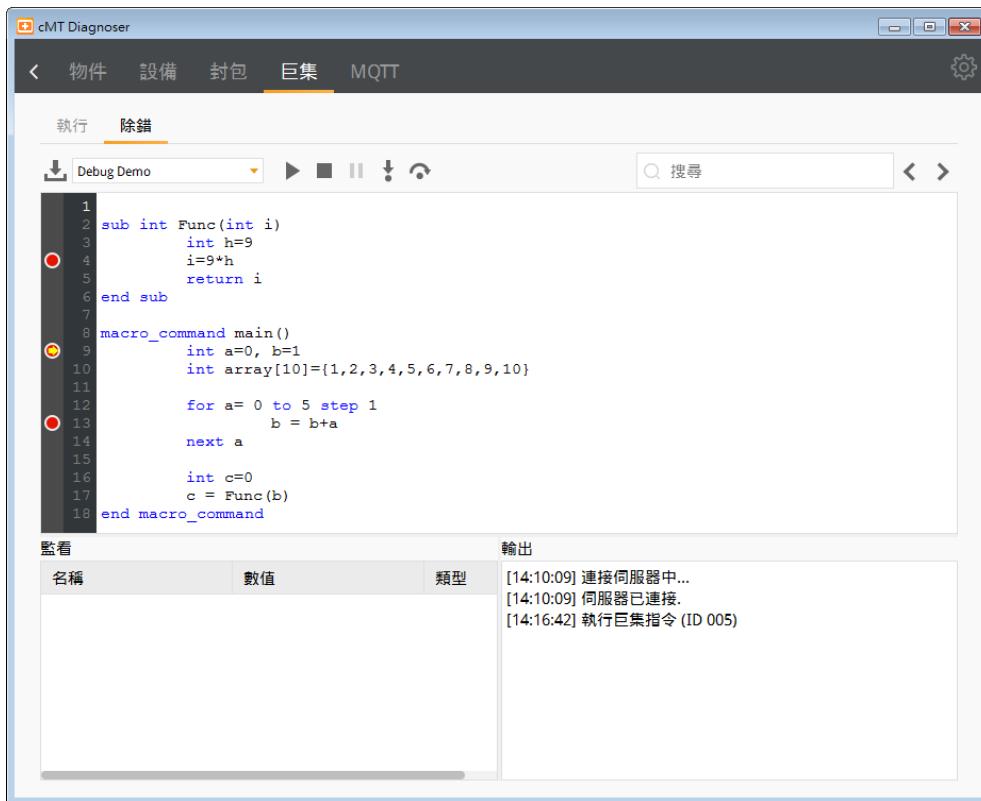
#### 1. 以下使用簡單範例說明巨集除錯流程與設定。點選箭頭開始除錯。



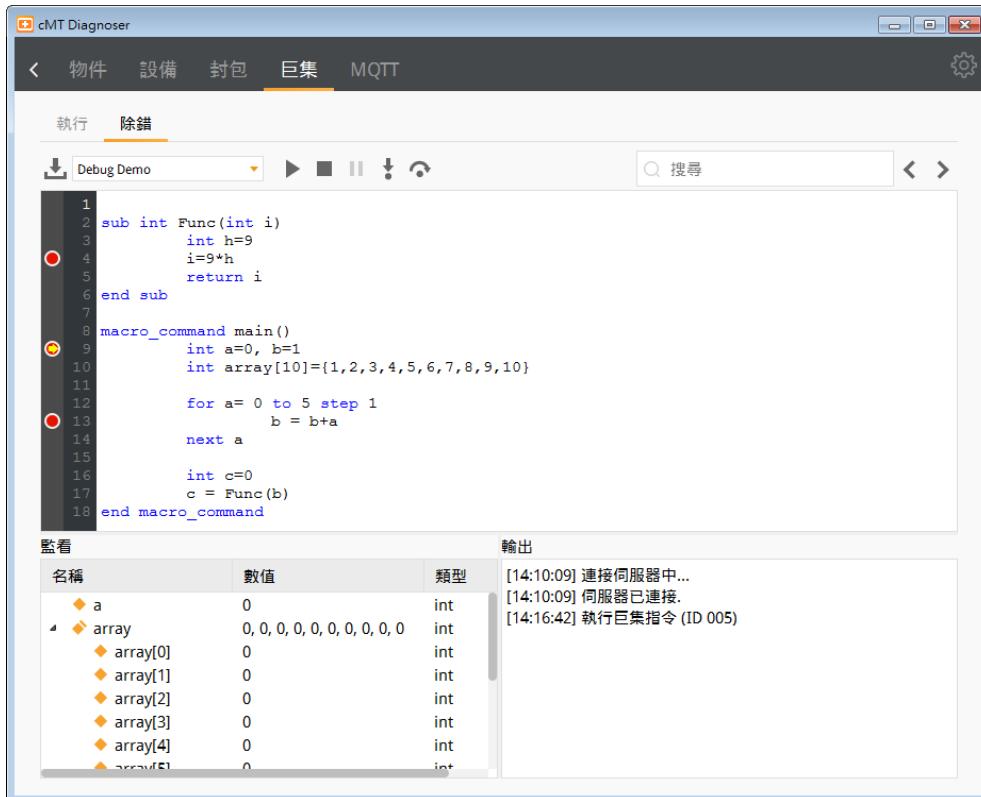
2. 從行數左邊加入或取消中斷點 (紅色圓圈)。



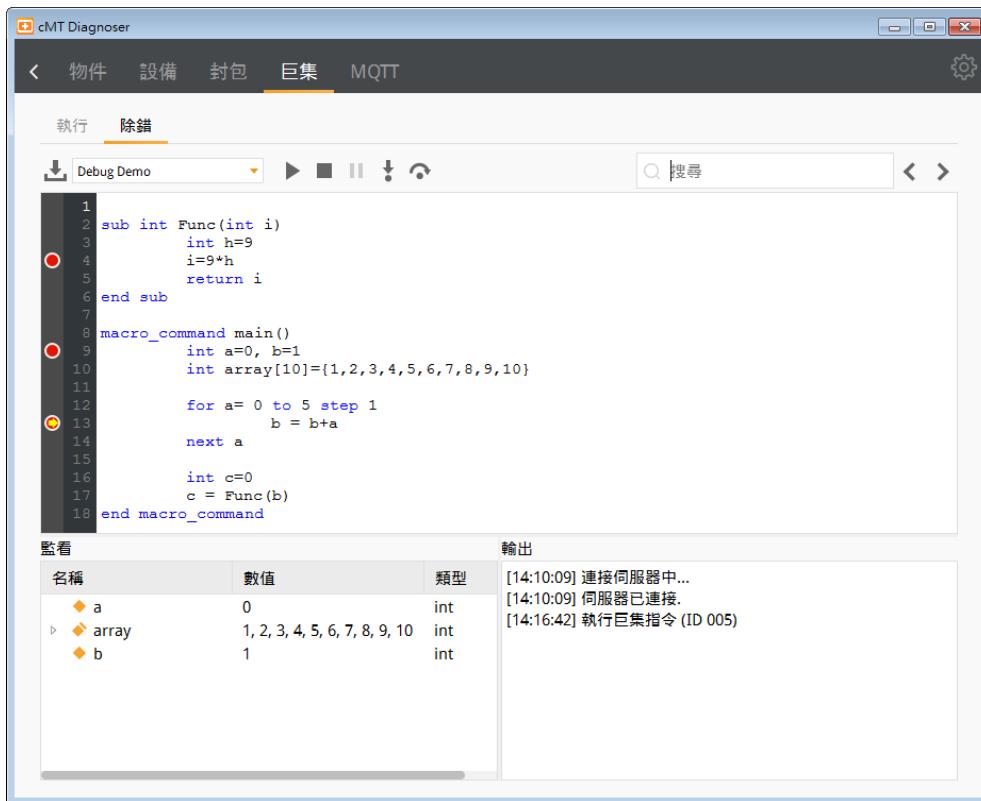
3. 觸發巨集之後，巨集會停留在第一個中斷點 (黃色箭頭)。



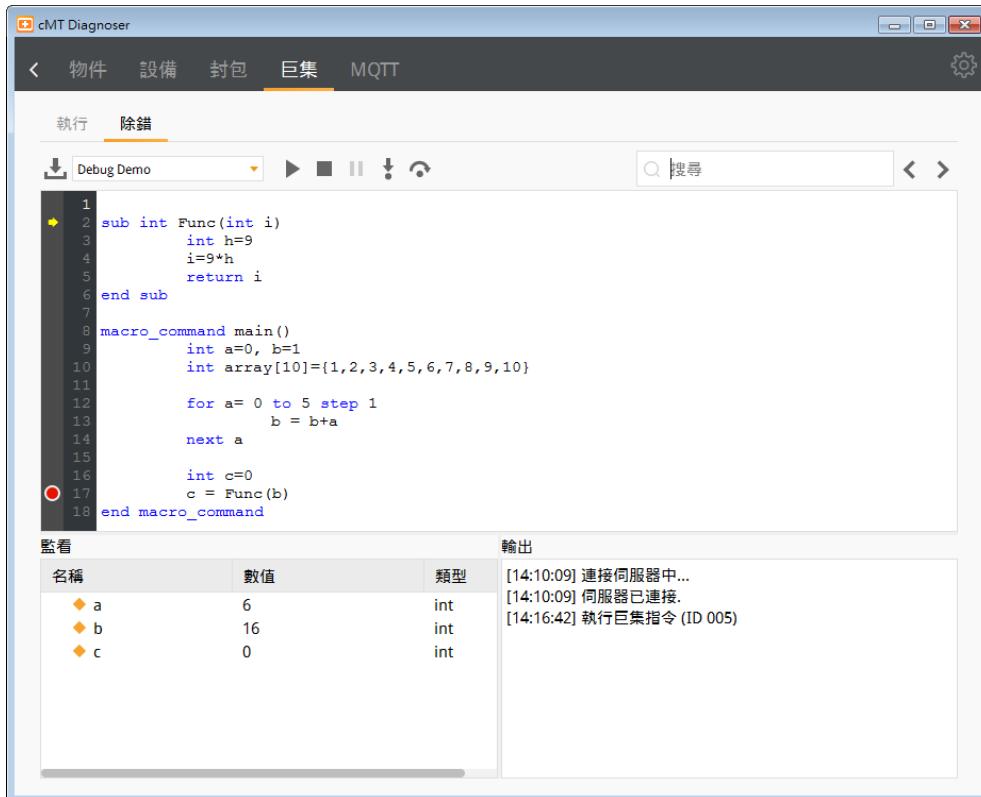
4. 此時可以在監看頁面新增想觀察的變數，若變數為陣列也可在監看視窗展開，查看陣列數值。



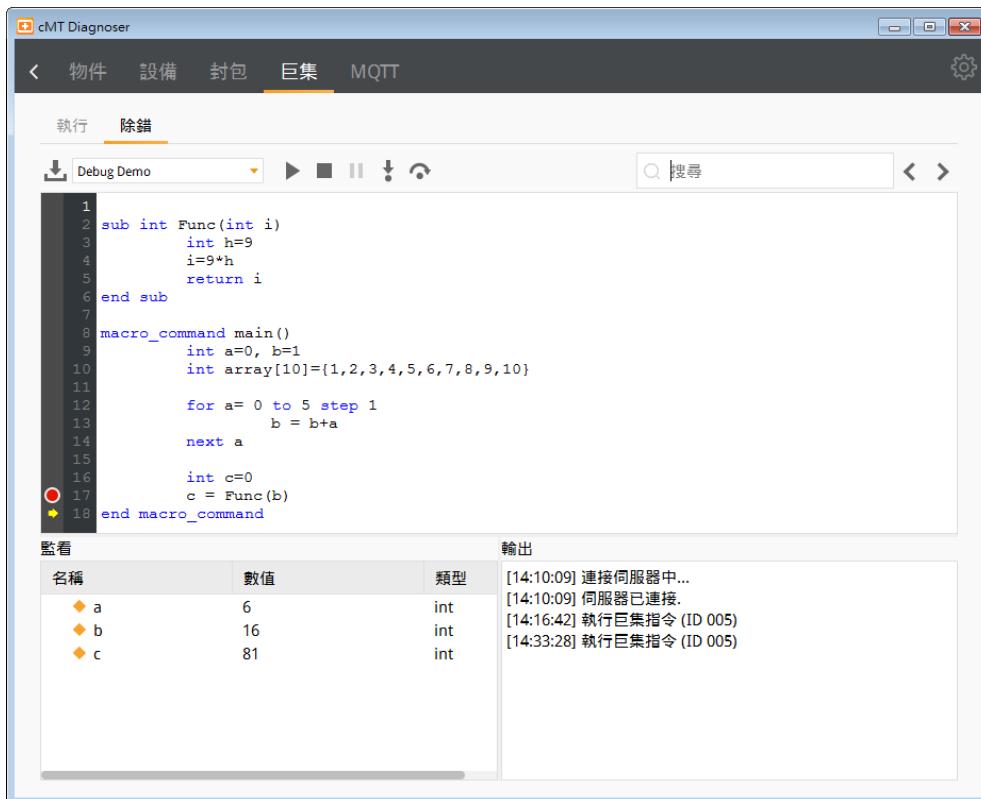
5. 按下開始除錯，巨集將會執行到下一個中斷點。變數的數值也會隨著改變。



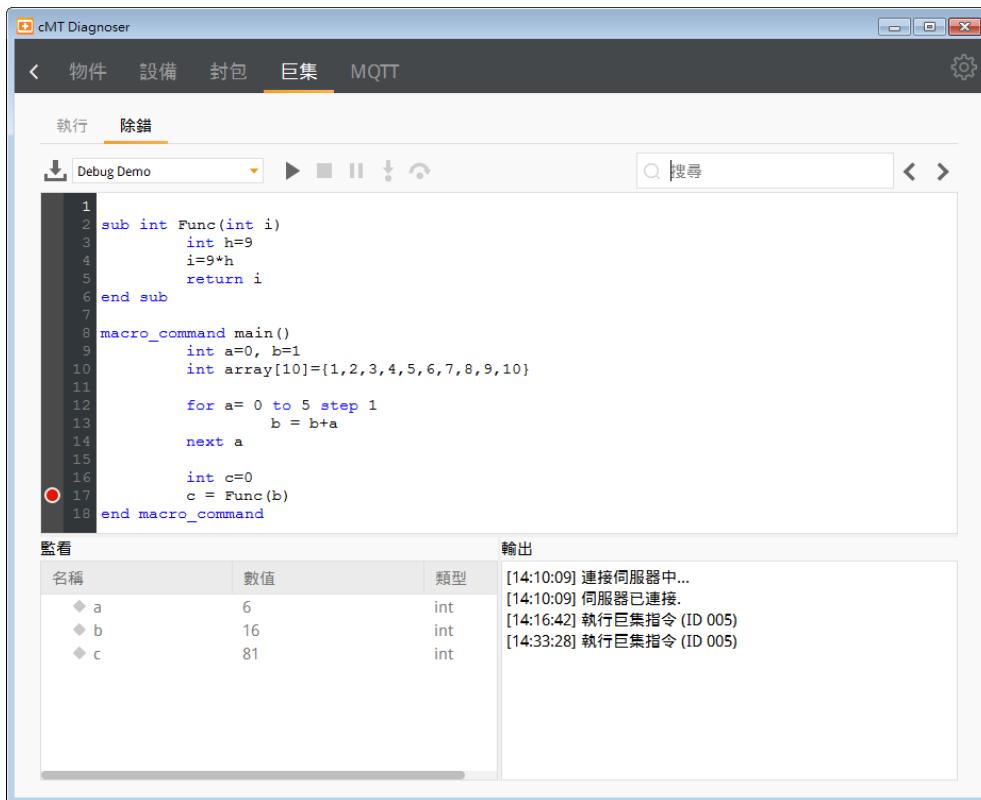
6. 進入(Step into)為逐步執行命令，若下一步為函式執行則會讓巨集進入該函式並依序更新變數的數值。



7. 跳過(Step over) 為逐步執行命令，若下一步為函式，不會進入函式逐步執行除錯而是直接回傳函式執行結果。但若函式內有設定中斷點，則還是會在函式中的中斷點停止。



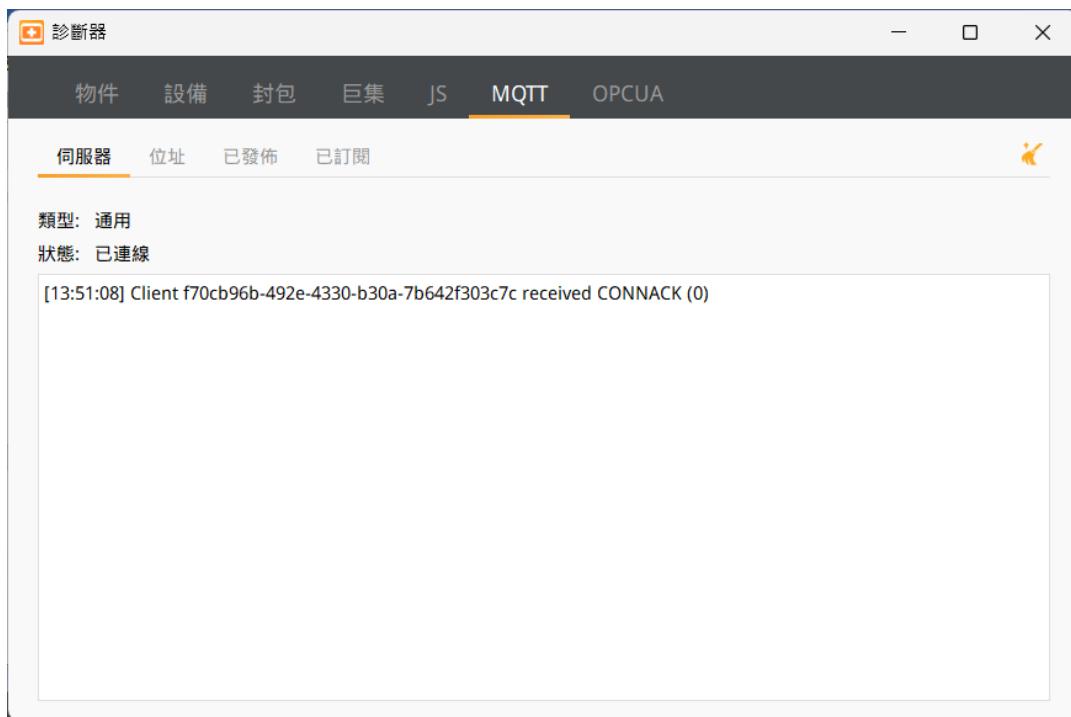
8. 按下停止除錯結束巨集除錯功能。



### 31.3.5. MQTT 設定

1. 伺服器頁面顯示 MQTT 伺服器的類型與目前狀態。目前支援的有通用/Azure IoT

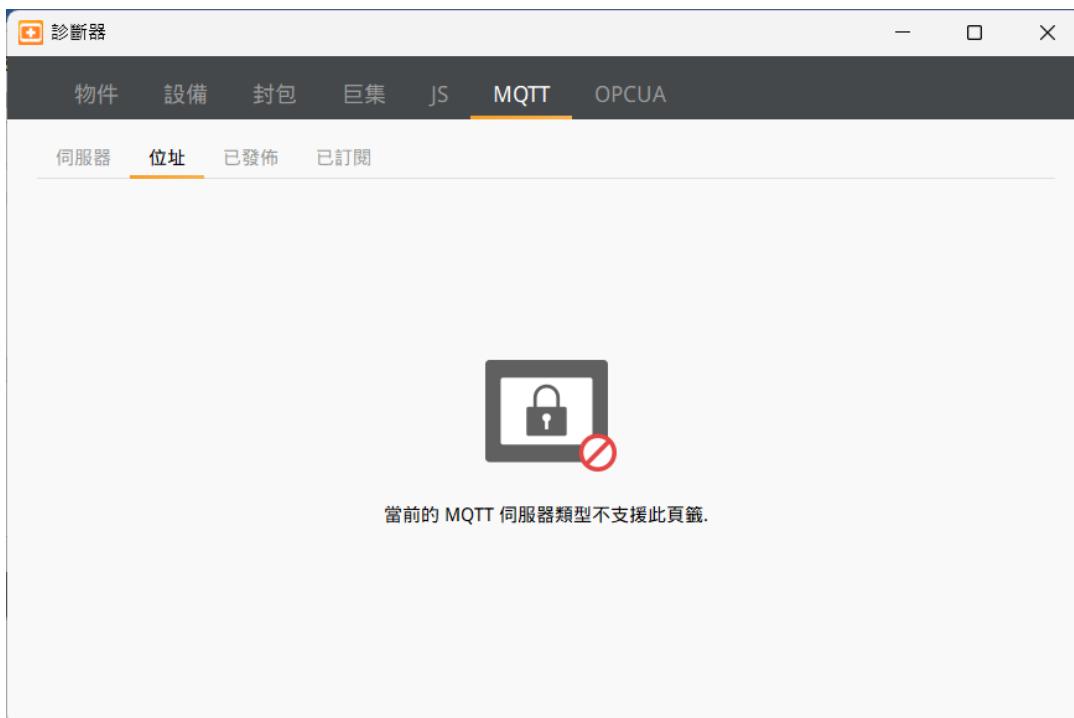
Hub/Sparkplug，AWS IoT 不支援。



- 位址頁面顯示此 HMI 設定所有發佈與訂閱的主題。若勾選主題，可讓 cMT 診斷器收到所有發佈與訂閱的 MQTT 訊息以及可以直接更改位址的數據，方便測試 MQTT 功能。只支援通用伺服器。

主題	設備	位址	類型	長度	數值
<b>發佈</b>					
topic 1	Local HMI	LB-0	Bit	1	test false
	Local HMI	LW-0	16-bit Unsigned	1	0
<b>訂閱</b>					
topic 1	Local HMI	LB-0	Bit	1	test false
	Local HMI	LW-0	16-bit Unsigned	1	0

如使用非通用伺服器將會出現以下畫面。



3. 發佈主題頁面顯示 HMI 發出的訊息內容。主題選擇壓縮格式或是使用 Raw Data 時，以 16 進位顯示訊息封包內容。只支援通用伺服器。

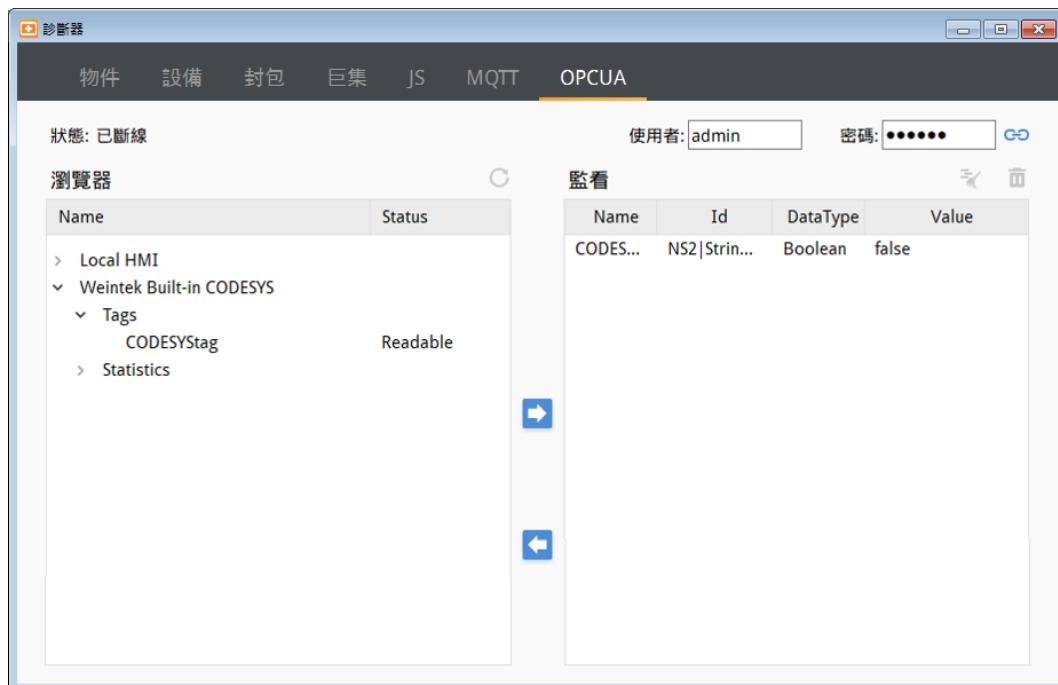


4. 訂閱主題頁面顯示 HMI 訂閱的訊息內容。主題選擇壓縮格式或是使用 Raw Data 時，以 16 進位顯示訊息封包內容。只支援通用伺服器。



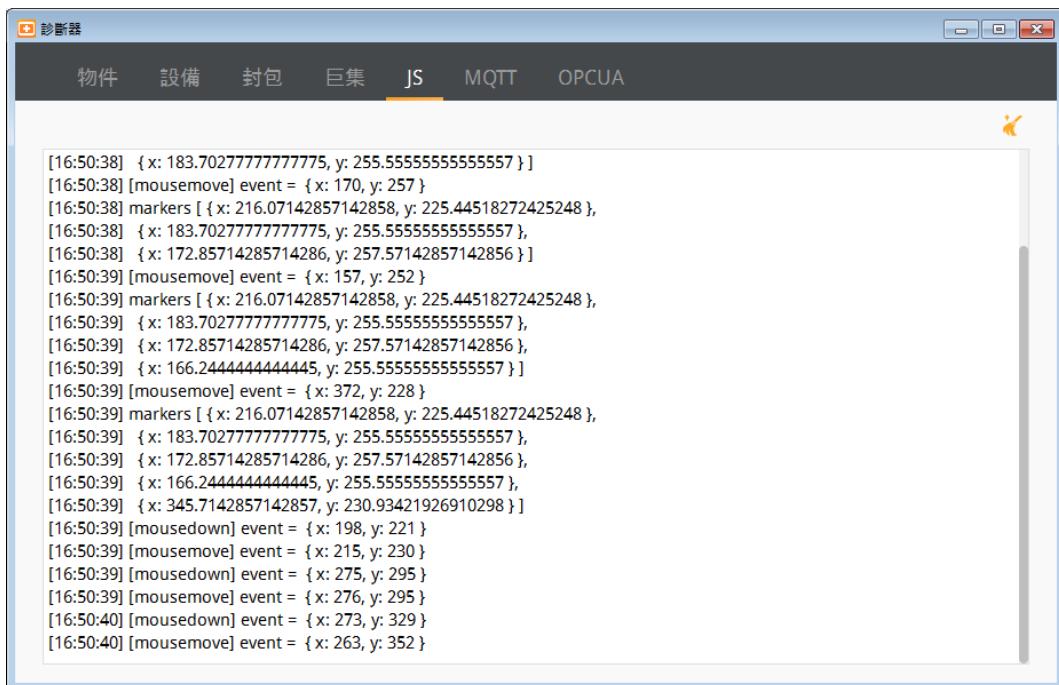
### 31.3.6. OPC UA 設定

- 可用使用者驗證的方式登入並監控 OPC UA 節點的數值。



### 31.3.7. JS 設定

- 在此可檢視 JS 物件所產生的 console 訊息。



### 31.4. 補充說明

1. 物件與監看位址支援所有 HMI 可通訊的 PLC 位址，包含 Tag PLC 位址。
2. 位址數值的顯示方式支援 Decimal/HEX/Binary。使用右鍵可呼叫選單替換顯示格式。
3. cMT 診斷器會記錄上一次關閉時的狀態與位置。如果直接關閉 cMT Viewer，下一次模擬時 cMT 診斷器將會自動彈出。反之，關閉 cMT Viewer 前就先將診斷器關閉，下一次模擬時就需手動開啟診斷器。
4. 當監看 cMT-SVR/cMT-SVRX 機種的資訊時，若直接使用 cMT Diagnoser 連到 cMT-SVR/cMT-SVRX 機種，僅能查看全域物件的資訊。若先透過 cMT Viewer 連線 cMT-SVR/cMT-SVRX，並藉由按下右鍵點選診斷器，則可以查看當前視窗的物件資訊。
5. 巨集指令若有設定密碼保護，將無法產生巨集除錯檔案。
6. 個別巨集若有設定密碼，在除錯頁面需要輸入密碼來解開。未解開時，該巨集若是唯讀，可以看到巨集內容，但在模擬時不能更改。若是加密，則看不到巨集內容。
7. 編譯失敗的巨集會被放到未完成編譯的列表，不會被包含到巨集除錯檔案裡。
8. 同一時間內巨集除錯功能只能對一個巨集執行。
9. 執行巨集除錯時，若中斷點設置在不需要執行的地方，例如：空白，無附值的變數宣告等，將會以反白的樣式呈現，如下圖第 7 行。

The screenshot shows the cMT Diagnoser software interface. The main window title is "Diagnoser". The top menu bar includes "Object", "Device", "Packet", "Macro" (which is highlighted in orange), and "MQTT". Below the menu is a toolbar with "Execute" and "Debug" tabs, where "Debug" is selected. A dropdown menu shows "Debug Demo". The central area displays a code editor with the following pseudocode:

```
1 |
2 sub int Func(int i)
3     int h=9
4     i=9*h
5     return i
6 end sub
7
8 macro_command main()
9     int a=0, b=1
10    int array[10]={1,2,3,4,5,6,7,8,9,10}
11
12    for a= 0 to 5 step 1
13        b = b+a
14    next a
15
16    int c=0
17    c = Func(b)
18 end macro_command
```

Below the code editor is a "Watch" table with three columns: "Name", "Value", and "Type". The table is currently empty. To the right of the table is an "Output" window containing the text:

```
[11:20:18] Connecting to server...
[11:20:18] Server connected.
```

# 32. FTP 啟服器之運用

本章節說明 FTP 啟服器之運用。

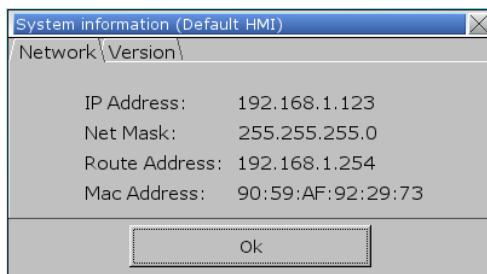
32.1. 概要 .....	32-2
32.2. 登入 FTP 啟服器的步驟 .....	32-2
32.3. 備份歷史資料及更新配方資料 .....	32-3

### 32.1. 概要

除了使用 SD 卡、USB 碟或 EasyPrinter 伺服器將歷史資料由 HMI 備份至 PC 之外，也能利用 FTP 伺服器達成這個目標。當工程檔案下載到 HMI 後，可透過 FTP 伺服器進行歷史資料備份、配方資料備份或更新配方資料的動作，但是無法刪除存在 FTP 伺服器內的檔案。拔除 USB 碟或 SD 卡前，請先中斷 FTP 連線，否則可能會導致下次插入 USB/SD 時無法使用。

### 32.2. 登入 FTP 伺服器的步驟

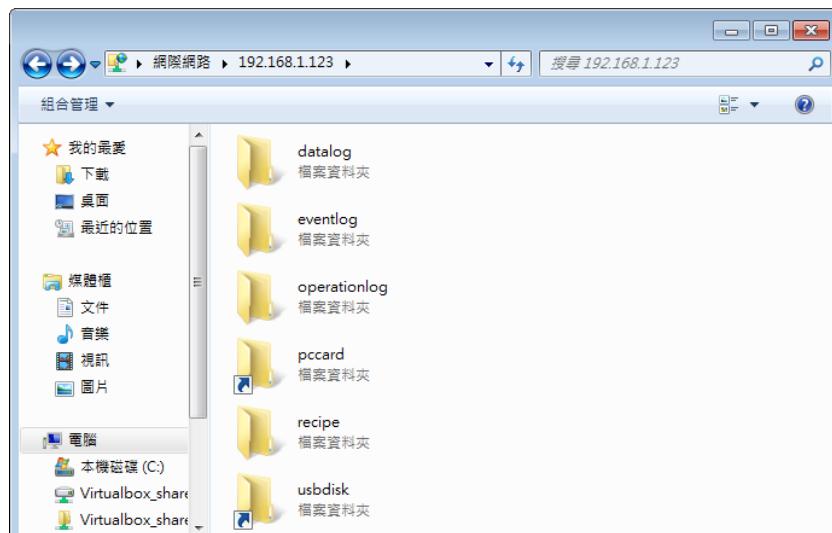
- 在登入 FTP 伺服器之前請先確認 HMI 的 IP 位址。



- 在 [我的電腦] 網址列輸入 HMI 的 IP 位址 <ftp://192.168.1.123> (範例) 然後登入帳號 uploadhis，密碼為 HMI 的 history upload password (若沒有更改過密碼，預設密碼為 111111)。或是直接輸入 <ftp://uploadhis:111111@192.168.1.123/>



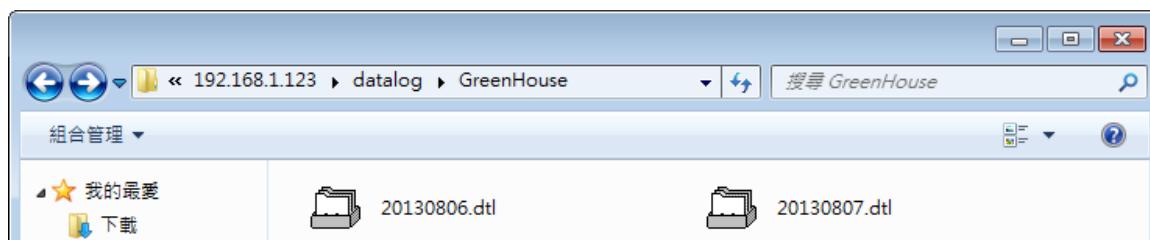
- 輸入 IP 位址後，網址列會顯示為 <ftp://192.168.1.123/>，並且可以看到下圖所示的資料夾。



### 32.3. 備份歷史資料及更新配方資料

- 備份資料取樣記錄

1. 點選 **datalog** 資料夾後，可看到 **datalog** 在 EasyBuilder Pro 設定的資料夾檔名。
2. 點選檔名後即可看到 **datalog** 的檔案。
3. 使用複製及貼上的功能，將資料取樣的記錄保存在 PC 上。



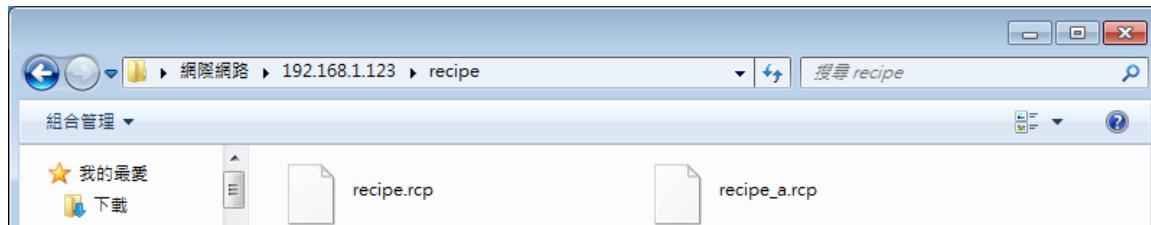
- 備份警報及事件記錄

1. 點選 **eventlog** 資料夾後，即可看到事件的記錄檔案。
2. 可使用複製及貼上的功能將事件記錄保存在 PC 上。

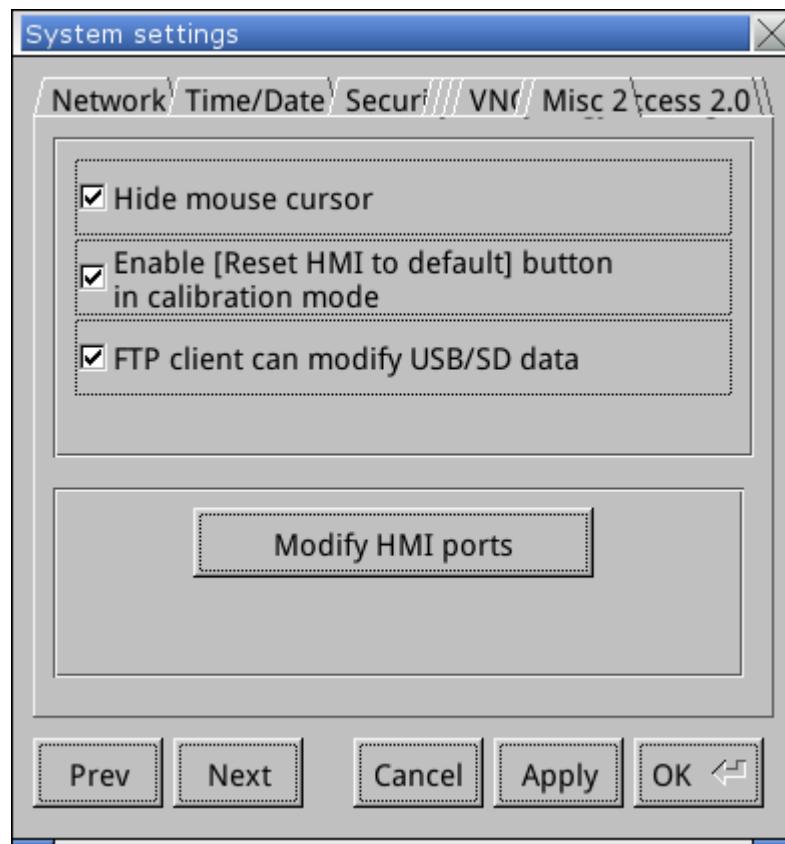


- 備份及更新配方資料

1. 點選 **recipe** 資料夾後，即可看到配方資料的檔案。
2. 可使用複製及貼上的功能將配方資料保存在 PC 上。



- FTP client 端可以修改 USB/SD 的資料
1. 預設為未啟用。啟用之後可以透過 FTP 功能刪除或寫入 USB/SD 內的檔案。
  2. 如下圖，MT 系列在系統設定的“Misc 2”頁面可以找到該設定，cMT / cMT X 系列則是在系統設定的“一般屬性”頁面。





### Note

- 因為配方資料每分鐘會自動儲存一次，若要更新 recipe.rcp 或 recipe\_a.rcp，請務必於 1 分鐘內將 HMI 重啟，否則新配方資料將被原有舊配方覆蓋過去。
- 使用者也可使用系統暫存器 LB-9047 (重新啟動 HMI) 及 LB-9048 (重啟機制保護) 將 HMI 重啟。若是使用 LB-9047 及 LB-9048，需先將 LB-9048 設 ON 後，再將 LB-9047 設 ON，即可重啟 HMI。
- 使用 FTP 伺服器功能的時候，人機中檔案的修改日期會被視為 UTC+0，所以在 PC 上查看檔案的時候，檔案的修改時間會被加上 PC 的時區設定，導致與實際的時間不符的現象。

# 33. EasyDiagnoser

本章節說明如何使用 EasyDiagnoser。

33.1.	概要	33-2
33.2.	設定步驟	33-2
33.3.	EasyDiagnoser 設定	33-3
33.4.	錯誤代碼	33-9
33.5.	視窗調整	33-10

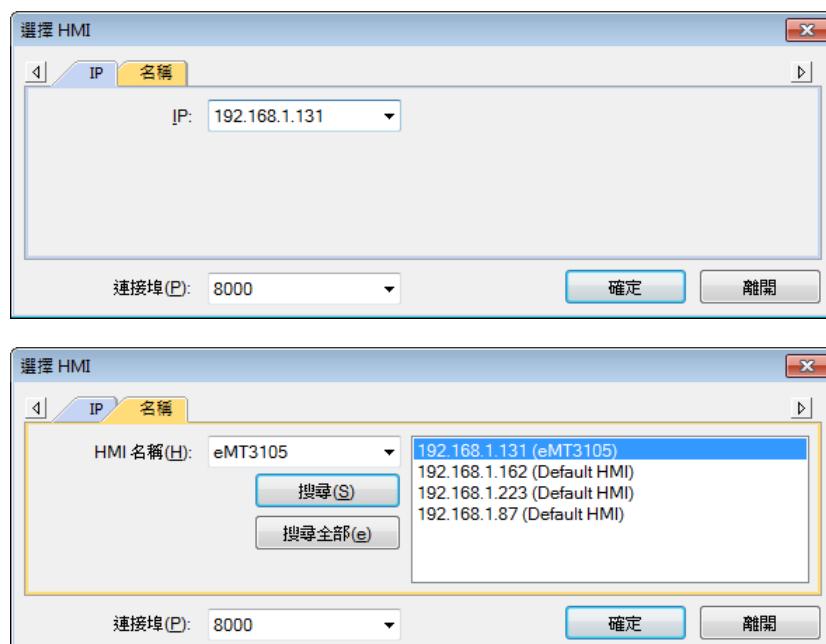
### 33.1. 概要

EasyDiagnoser 是用來偵測 HMI 與 PLC 之間通訊是否正常的工具。

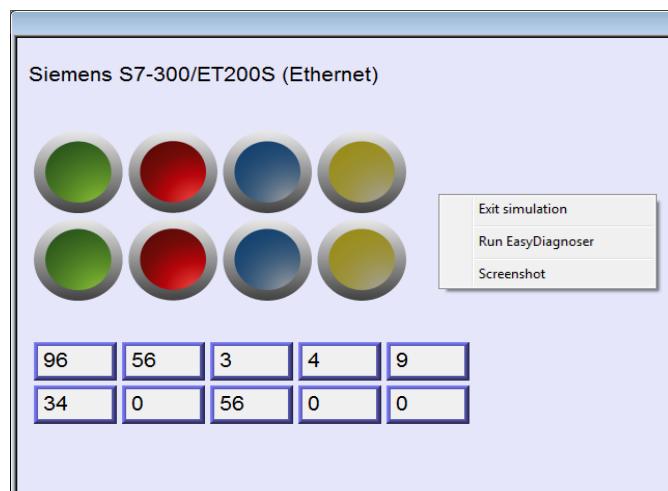
### 33.2. 設定步驟

以下為設定 EasyDiagnoser 的步驟：

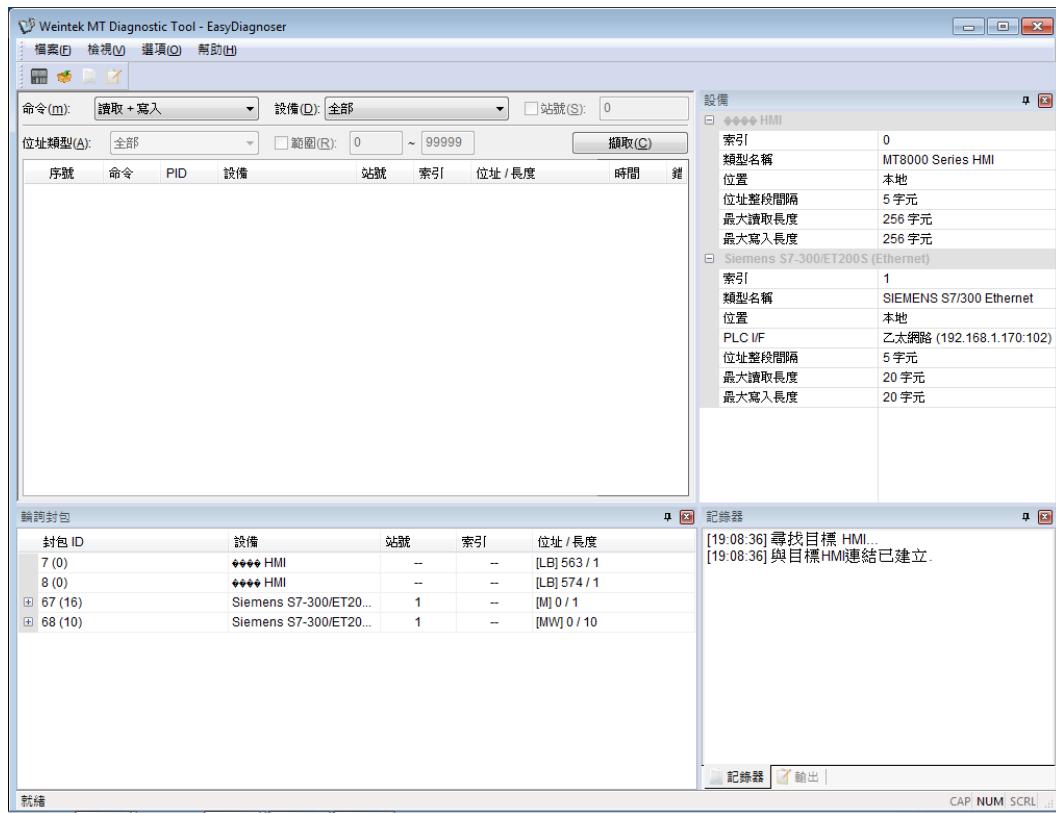
1. 開啟 Utility Manager 並且點選 EasyDiagnoser。
2. 設定欲進行通訊之 HMI 的 IP 位址，可選擇自行輸入 IP 位址或使用 [Search all] 功能，並輸入 [Project Port]。



另外在 EasyBuilder Pro 執行 On Line Simulation 時，按下右鍵可選擇 [Run EasyDiagnoser] 進入 EasyDiagnoser。



3. 完成以上設定之後，按下 [OK]，EasyDiagnoser 操作畫面如下圖所示：



### 33.3. EasyDiagnoser 設定

#### 33.3.1. 主要選單

項目	描述
檔案	<b>另存新檔</b> 可將擷取下來的通訊資料，儲存成 .xls 檔案，並可於 Excel 開啟。 <b>離開</b> 離開目前檔案。
檢視	<b>設備列表</b> 可顯示設備列表視窗。 <b>封包列表</b> 可顯示封包列表視窗。 <b>訊息視窗</b> 可顯示訊息視窗。 <b>輸出視窗</b> 可顯示輸出視窗。
選項	<b>工具列</b> 顯示 [設備列表]、[封包列表]、[訊息視窗]、[輸出視窗] 之工具列。 <b>狀態列</b> 在 EasyDiagnoser 視窗底端，顯示 CAP，NUM 或

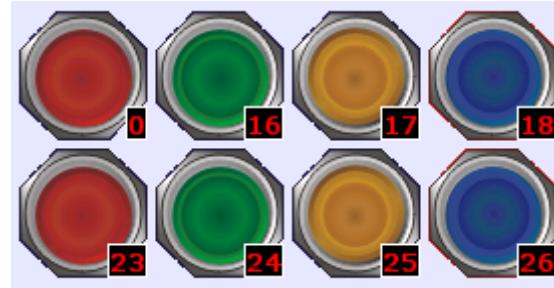
SCRL 的資訊。

#### 更新封包列表

顯示目前 HMI 頁面的封包。

#### 顯示物件 ID

顯示 HMI 上物件的 ID。



#### 清除通訊記錄

清除所有通訊中所記錄的資訊。

#### 幫助

顯示 EasyDiagnoser 版本資訊。

### 33.3.2. 通訊記錄區

使用者可以觀察 HMI 和 PLC 之間的通訊。

Weintek MT Diagnostic Tool - EasyDiagnoser								
命令(m): 讀取 + 寫入			設備(D): 全部		站號(S): 0			
位址類型(A):		全部	<input type="checkbox"/> 範圍(R): 0 ~ 99999		<input type="button" value="擷取(C)"/>			
序號	命令	PID	設備	站號	索引	位址 / 長度	時間	錯
239	R	8	Local HMI	--	--	[LB] 574 / 1	10 0	
238	R	7	Local HMI	--	--	[LB] 563 / 1	10 0	
237	R	68	Siemens S7-300/ET200S...	1	--	[MW] 0 / 10	40 0	
236	R	8	Local HMI	--	--	[LB] 574 / 1	20 0	
235	R	7	Local HMI	--	--	[LB] 563 / 1	30 0	
234	R	67	Siemens S7-300/ET200S...	1	--	[M] 0 / 1	40 0	
233	R	8	Local HMI	--	--	[LB] 574 / 1	10 0	
232	R	7	Local HMI	--	--	[LB] 563 / 1	20 0	
231	R	8	Local HMI	--	--	[LB] 574 / 1	20 0	
230	R	68	Siemens S7-300/ET200S...	1	--	[MW] 0 / 10	40 0	
229	R	7	Local HMI	--	--	[LB] 563 / 1	20 0	
228	R	8	Local HMI	--	--	[LB] 574 / 1	20 0	
227	R	67	Siemens S7-300/ET200S...	1	--	[M] 0 / 1	40 0	
226	R	7	Local HMI	--	--	[LB] 563 / 1	10 0	
225	R	8	Local HMI	--	--	[LB] 574 / 1	30 0	
224	R	68	Siemens S7-300/ET200S...	1	--	[MW] 0 / 10	30 0	
223	R	7	Local HMI	--	--	[LB] 563 / 1	20 0	
222	R	8	Local HMI	--	--	[LB] 574 / 1	10 0	
221	R	7	Local HMI	--	--	[LB] 563 / 1	10 0	
220	R	67	Siemens S7-300/ET200S...	1	--	[M] 0 / 1	30 0	
219	R	8	Local HMI	--	--	[LB] 574 / 1	20 0	
218	R	7	Local HMI	--	--	[LB] 563 / 1	20 0	
217	R	68	Siemens S7-300/ET200S...	1	--	[MW] 0 / 10	50 0	

項目	描述
命令	<p><b>讀取 + 寫入</b> 在通訊記錄區顯示讀和寫的命令。</p> <p><b>讀取</b> 在通訊記錄區只顯示讀的命令。</p> <p><b>寫入</b> 在通訊記錄區只顯示寫的命令。</p>
設備	<p><b>全部</b> 顯示本地 HMI 和 PLC 的資訊。</p> <ul style="list-style-type: none"> <li>● 如果設定 [命令 : 讀取 + 寫入]，在通訊記錄區會顯示本地 HMI 和 PLC 的讀和寫的資訊。</li> <li>● 如果 [命令 : 讀取]，在通訊記錄區會顯示本地 HMI 和 PLC 的讀的資訊。</li> <li>● 如果 [命令 : 寫入]，在通訊記錄區會顯示本地 HMI 和 PLC 的寫的資訊。</li> </ul> <p><b>Local HMI</b> 顯示本地 HMI 的資訊。</p> <ul style="list-style-type: none"> <li>● 如果設 [命令 : 讀取 + 寫入]，在通訊記錄區會顯示本地 HMI 的讀和寫的資訊。</li> <li>● 如果 [命令 : 讀取]，在通訊記錄區會顯示本地 HMI 的讀的資訊。</li> <li>● 如果 [命令 : 寫入]，在通訊記錄區會顯示本地 HMI 的寫的資訊。</li> </ul> <p><b>PLC</b> 顯示 PLC 的資訊。</p> <ul style="list-style-type: none"> <li>● 如果設 [命令 : 讀取 + 寫入]，在通訊記錄區會顯示 PLC 的讀和寫的資訊。</li> <li>● 如果 [命令 : 讀取]，在通訊記錄區會顯示 PLC 的讀的資訊。</li> <li>● 如果 [命令 : 寫入]，在通訊記錄區會顯示 PLC 的寫的資訊。</li> </ul>
站號	選擇想顯示的 PLC 之站號。 (當 [設備] 選擇 All 時無法使用此功能)
位址類型	使用者可以選擇全部或是其中的設備位址類型顯示在螢幕上。 (當 [設備] 選擇 All 時無法使用此功能)
範圍	設定要擷取的位址範圍。 (當 [設備] 選擇 All 時無法使用此功能)
擷取	點選 [擷取] 鈕開始或停止擷取通訊信息。
錯誤碼	請參考本章《33.4 錯誤代碼》。

### 33.3.3. 輪詢封包

輪詢封包				
封包 ID	設備	站號	索引	位址 / 長度
7 (0)	Local HMI	--	--	[LB] 563 / 1
8 (0)	Local HMI	--	--	[LB] 574 / 1
+ 67 (16)	Siemens S7-300/ET...	1	--	[M] 0 / 1
+ 68 (10)	Siemens S7-300/ET...	1	--	[MW] 0 / 10

項目	描述
封包 ID	封包的 ID 編號，可由通訊記錄區看出有問題的封包 ID 的物件。
設備	顯示 HMI 和 PLC 型號。
站號	顯示 PLC 站號。
索引	顯示物件所使用的索引暫存器編號。
位址/長度	顯示設備類型位址及封包內的 word 長度。

輪詢封包	物件	視窗	ID	位址
	8 (0)	Local HMI	--	[LB] 574 / 1
	67 (16)	Siemens S7-300/ET...	1	[M] 0 / 1
▶	位元狀態切換開關	10	1	[M] 0
	位元狀態切換開關	10	1	[M] 0
	位元狀態切換開關	10	2	[M] 0
	位元狀態切換開關	10	2	[M] 0
	位元狀態切換開關	10	3	[M] 0
	位元狀態切換開關	10	3	[M] 0
	位元狀態切換開關	10	4	[M] 0
	位元狀態切換開關	10	4	[M] 0
	位元狀態切換開關	10	5	[M] 0
	位元狀態切換開關	10	5	[M] 0
	位元狀態切換開關	10	6	[M] 0
	位元狀態切換開關	10	6	[M] 0
	位元狀態切換開關	10	7	[M] 0

項目	描述
物件	封包 ID 內的物件。
視窗	物件在程式中的所在視窗。
ID	物件的 ID 號碼。
位址	物件位址。



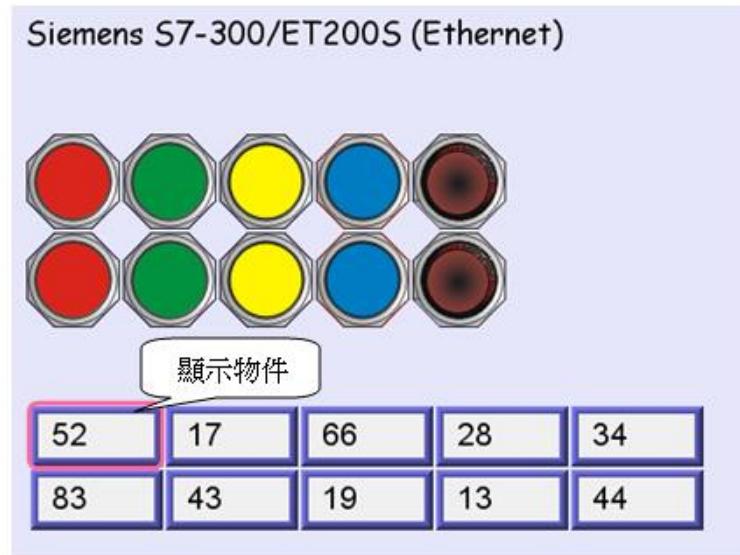
- 點選封包 ID 後，第 3 欄會顯示設備的站號：

封包 ID	設備	站號	索引	位址 / 長度
+ 131 (32)	SIEMENS S7/200	3	--	[MW_Bit] 0000..
+ 130 (32)	SIEMENS S7/200	3	--	[M] 00000 / 1
+ 129 (32)	SIEMENS S7/200	3	--	[Q] 00000 / 1
+ 128 (32)	SIEMENS S7/200	3	--	[I] 00000 / 1
+ 127 (16)	SIEMENS S7/200	3	--	[MW] 00000 / 16
... Numeric Input		10	0	[MW] 00000
... Numeric Input		10	65	[MW] 00001

- 雙擊封包 ID 之後點選物件，可顯示物件所在的位置。

例如，選擇數值輸入 同時 [視窗] 顯示 10，表示此物件在程式中的第 10 視窗中，同時此物件在 HMI 上會被粉紅色的方框標示出來，如下圖所示：

物件	視窗	ID	位址
7 (0)	Local HMI	--	[LB] 563 / 1
8 (0)	Local HMI	--	[LB] 574 / 1
+ 67 (16)	Siemens S7-300/ET...	1	--
... 68 (10)	Siemens S7-300/ET...	1	--
▶ 數值輸入		10	[MW] 0
數值輸入		10	[MW] 2
數值輸入		10	[MW] 4
數值輸入		10	[MW] 6
數值輸入		10	[MW] 8
數值輸入		10	[MW] 10
數值輸入		10	[MW] 12
數值輸入		10	[MW] 14
數值輸入		10	[MW] 16
數值輸入		10	[MW] 18



### 33.3.4. 設備

設備視窗顯示 HMI 及 PLC 的相關訊息。

設備	
<input type="checkbox"/> Local HMI	
索引	0
類型名稱	MT8000 Series HMI
位置	本地
位址整段間隔	5字元
最大讀取長度	256字元
最大寫入長度	256字元
<input type="checkbox"/> Siemens S7-300/ET200S (Ethernet)	
索引	1
類型名稱	SIEMENS S7/300 Ethernet
位置	本地
PLC I/F	乙太網路 (192.168.1.170:102)
位址整段間隔	5字元
最大讀取長度	20字元
最大寫入長度	20字元

### 33.3.5. 輸出 (Macro debug)

搭配使用 Macro 所提供的 Trace 函數，即可偵測 Macro 執行的狀態。

以下圖為例，[ID 1,Ln 7] 及 [ID 1,Ln 12]：

ID 1 表示 Macro 的名稱。

Ln 7 及 Ln 12 表示顯示在 Macro 中的第 7 行及第 12 行資料。

輸出	
[ID 1, Ln 7]	LW0 = 1
[ID 1, Ln 12]	LW0 = 2
[ID 1, Ln 7]	LW0 = 2
[ID 1, Ln 12]	LW0 = 3
[ID 1, Ln 7]	LW0 = 3
[ID 1, Ln 12]	LW0 = 4

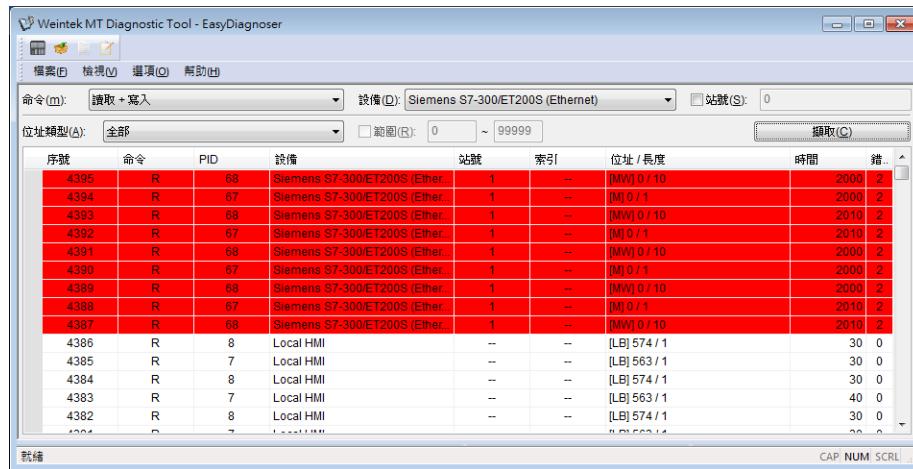
👉 詳細資訊請參考《18 巨集指令》。

### 33.4. 錯誤代碼

在通訊記錄區可從錯誤代碼找出錯誤原因。請參考下列錯誤代碼。

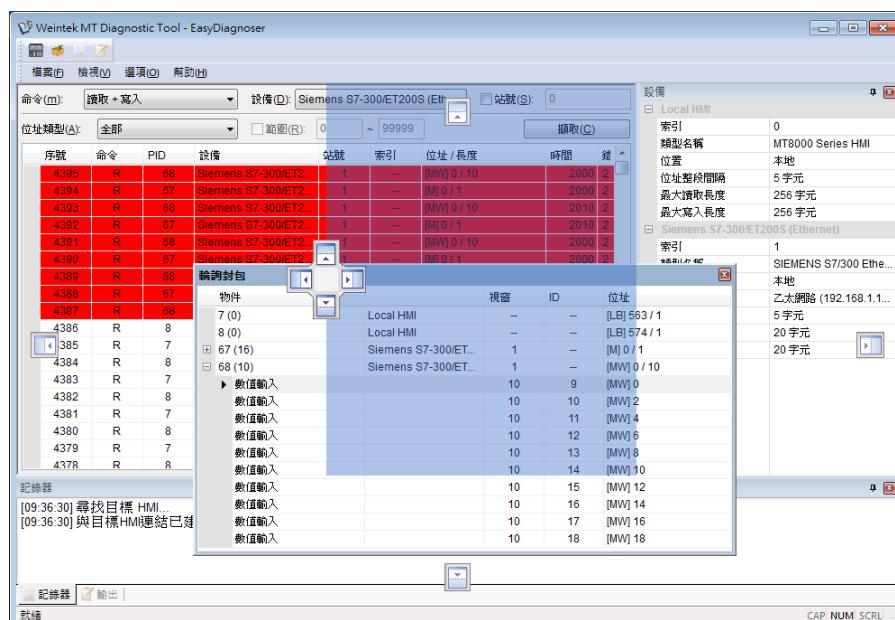
錯誤碼	通訊錯誤原因
0	正常
1	裝置忙線無法再接收命令
2	通訊錯誤 (原因不明)
3	裝置不存在
4	指定站號的裝置不存在
5	位址格式錯誤
6	讀取/寫入不支援的位址
7	裝置使用的驅動程式不存在
8	序列埠(COM Port)不存在
9	裝置的 IP 位址錯誤或是無法連線到該裝置
10	裝置所回覆的內容檢核錯誤(checksum error)
11	無法辨識的命令
12	忽略
20	未正確連接使用 USB 介面的裝置
21	未正確連接使用 CAN Bus 介面的裝置
22	未接受到來自裝置的任何回覆
23	未在指定的時間內 (timeout) 自裝置讀取到足夠數量的數據
24	物件所使用的轉換標籤 (Conversion Tag) 不存在或是內容錯誤
25	HMI 拒絕接收來自 Remote HMI 的命令
251	讀取/寫入 MODBUS 裝置暫存器的字數 (word no.) 超過允許值
252	MODBUS 裝置所回覆數據的格式不正確
253	MODBUS 裝置所回覆數據檢核錯誤 (checksum error)

當錯誤發生時，錯誤的訊息會標示成紅色，如下圖所示：



### 33.5. 視窗調整

使用者可以使用拖曳功能及顯示在編輯畫面上的多個定點圖示來放置視窗到喜愛的位置。



#### Note

- EasyDiagnoser 不支援使用 Siemens S7-1200 (Ethernet)、Rockwell EtherNet / IP (CompactLogix) – Free Tag Names 和 Rockwell EtherNet / IP (ControlLogix) – Free Tag Names 等使用 tag 的 PLC。

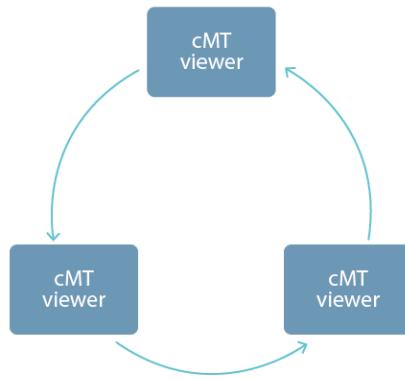
請點選此圖示觀看教學影片，請先確定已連上網路線。

# 34. 控制權(cMT/cMT X 系列)

本章節說明如何使用控制權。

34.1. 概要 .....	34-2
34.2. 設定 .....	34-2
34.3. 範例講解 .....	34-4

### 34.1. 概要

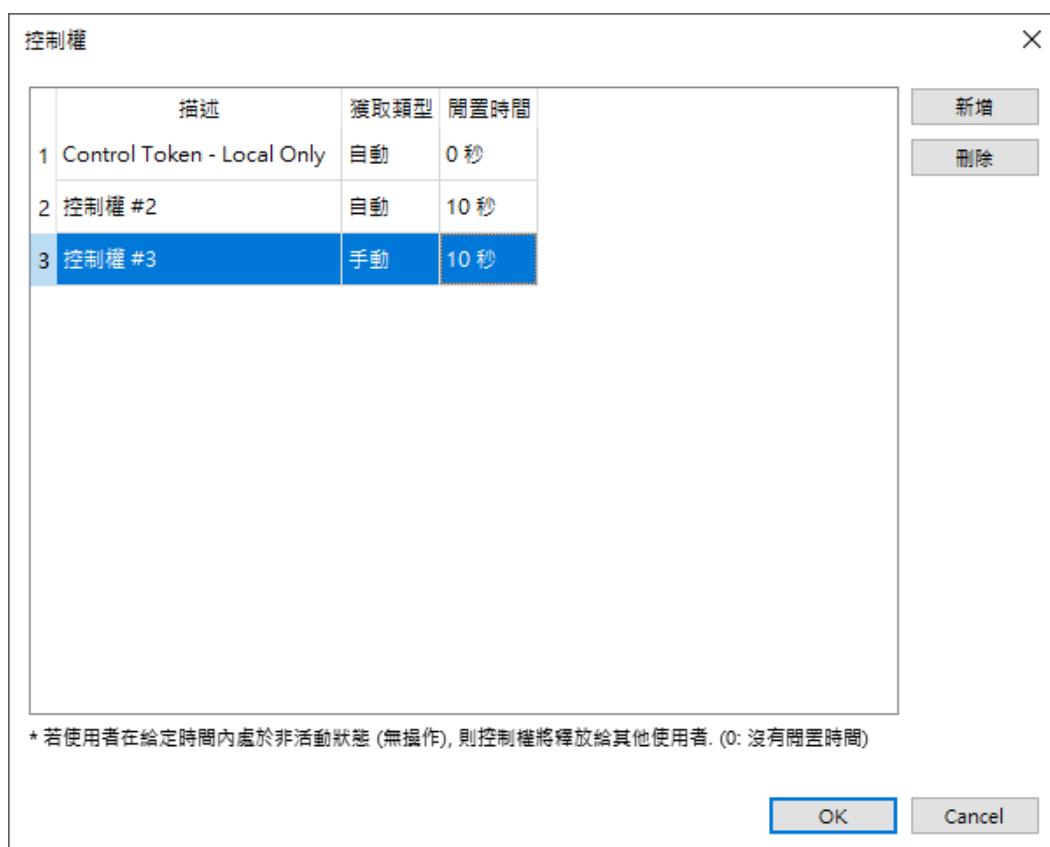


由於 cMT / cMT X 系列可同時有多台 cMT Viewer 進行操作，為了避免同時有多台的 cMT viewer 裝置操作同個物件，因為沒有意識到有其他人也在操作而影響到工作流程的操作安全，可以使用控制權進行物件的保護。簡單來說，只有拿到控制權的 cMT Viewer 裝置，才能對物件進行操作。啟用控制權時，同時間一個控制權僅有一台裝置可以取得，其餘的裝置則必須依序等待取得物件的控制權。

### 34.2. 設定



點選功能表列 [工程檔案] 內的 [控制權] 後會跳出設定視窗。



設定	描述
描述	<p>設定控制權的名稱。</p> <p>其中第一項 [Control Token – Local Only] 僅存在於內建 cMT Viewer 機型，選擇此項目時僅本機端可操作物件。(例如：cMT3072、cMT3090、cMT3151、cMT-HDM 等...)</p>
獲取類型	<p>可以選擇自動或是手動。</p> <p><b>自動</b> cMT Viewer 在進入視窗時，將自動嘗試取得該視窗內所有物件所設定的控制權；在離開視窗時即會自動釋出控制權。</p> <p><b>手動</b> 使用者須透過動作觸發或複合式多功能按鈕設定動作以取得或釋放控制權。</p>
閒置時間	<p>為避免某一使用者取得控制權並在操作完畢後，未主動釋出控制權導致其他使用者無法操作，可啟用 [閒置時間] 功能。當使用者取得一個控制權後，若在指定期間內無操作，則系統會釋放此控制權。</p> <p>閒置時間範圍可為 0 ~ 86400 秒；當值設定為 0 時，意即取消 [閒置時間] 功能。</p> <p>Control Token – Local Only 沒有閒置時間，設定固定為 0。</p>

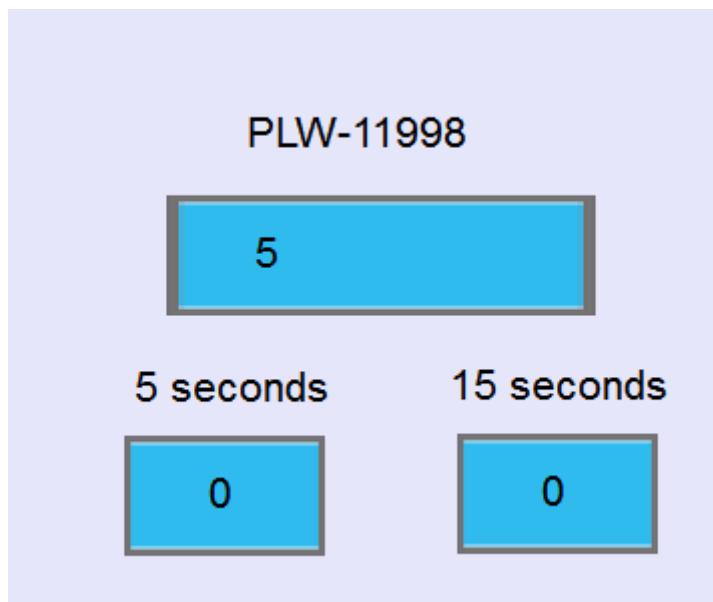
 Note

- 控制權取得機制為先到先得，且一個控制權僅可被一台 cMT Viewer 裝置取得。其餘裝置將依嘗試取得控制權的時間順序排入序列，依序等待取得控制權。
- 一台 cMT Viewer 裝置可以同時擁有多個控制權。
- 當獲取類型為自動時，即使換頁前後的視窗皆有設定相同控制權之物件，在進入新視窗後仍需要重新取得控制權。若控制權物件置放於重疊視窗或公用視窗，且換頁前後都使用同樣的重疊視窗，則換頁後無需重新取得控制權。
- 當 cMT Viewer 裝置與 Server 端斷線時，若斷線時間小於 30 秒又恢復通訊，則無須重新取得控制權；若斷線時間超過 30 秒，則需要重新取得控制權。

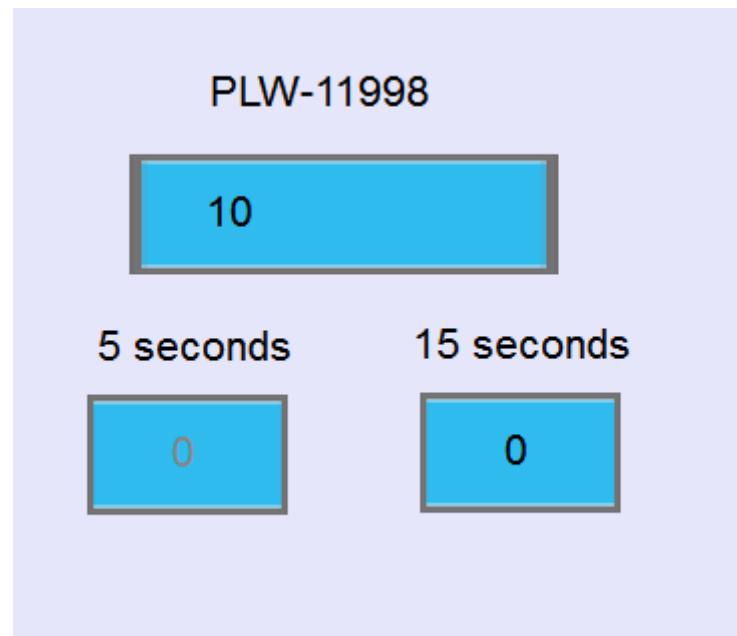
### 34.3. 範例講解

剩餘閒置時間可以透過 PLW-11998 搭配 [字元] 物件監看，該地址會倒數剩餘的閒置時間。若有兩個以上的控制權時，會先倒數完剩餘閒置時間少的控制權時間，再繼續倒數剩餘時間長的控制權。

如下圖所示，使用 [字元] 物件顯示 PLW-11998 來倒數剩餘閒置時間，兩個數值物件分別使用不同的控制權，一個閒置時間為 5 秒，一個閒置時間為 15 秒。執行模擬時，PLW-11998 會先倒數 5 秒的閒置時間。



當倒數完 5 秒後控制權被釋放，物件即轉為灰階。另一個閒置時間為 15 秒的控制權還剩下 10 秒即會釋放，此時 PLW-11998 會從 10 秒開始倒數。



# 35. EasyWatch

本章節說明如何使用 EasyWatch。

35.1.	概要	35-2
35.2.	設定	35-3
35.3.	監視物件設定	35-4
35.4.	巨集物件設定	35-9
35.5.	HMI 設定	35-11
35.6.	物件顯示列表	35-12

### 35.1. 概要

EasyWatch 可以透過 PC 監看或設定 HMI 和 PLC 內的位址數值，同時也可以進行巨集指令的呼叫，更方便使用者進行除錯及遠端監控使用。

以下範例說明，使用者如何透過 EasyWatch 查看設定值及資料的正確性。在 EasyBuilder Pro 新增 [數值輸入] 物件，位址設定成 LW-10，再到 EasyWatch 新增相同位址，當執行監視時，狀態為已連結，且顯示正確數值，表示已連線，即可開始監看。



#### Note

- 當系統暫存器 [LB-9044 (禁止遠端控制)] 或 [系統參數設定] » [系統設定] » [禁止遠端 HMI 連接] 被設定時，將無法使用 EasyWatch 功能監控。
- 使用 cMT / cMT X HMI 時，需於 [系統參數設定] » [HMI 屬性] 勾選“支援 iP/iE/XE/eMT/mTV 通訊協議和 EasyWatch”選項，並用該連接埠號與 EasyWatch 連線。

## 35.2. 設定

### 35.2.1. 基本功能

設定	描述
檔案	<b>新增</b> ：開啟一個新的 EasyWatch 檔案。 <b>開啟</b> ：開啟已編輯的 EasyWatch 檔案。 <b>儲存檔案</b> ：儲存 EasyWatch 檔案設定。 <b>另存新檔</b> ：可將 EasyWatch 檔案設定，儲存成 .ewt 格式。 <b>離開</b> ：關閉 EasyWatch。
編輯	<b>剪下</b> ：剪下選取的物件至剪貼簿中。 <b>複製</b> ：複製選取的物件至剪貼簿中。 <b>貼上</b> ：貼上剪貼簿中的物件。
物件	<b>新增物件</b> ：可新增監視物件或巨集物件。 <b>刪除物件</b> ：選擇欲刪除的物件，系統會彈出訊息，確認是否刪除。 <b>修改物件</b> ：選擇欲修改的物件，即可修改內容。 <b>HMI 管理器</b> ：對 HMI 進行新增、修改、移除的管理。 <b>執行</b> ：選擇欲執行的物件，即可執行此物件。 <b>停止</b> ：選擇執行中的物件，即可停止此物件。
說明	<b>說明主題</b> ：提供基本功能的操作方法，供使用者參考。 <b>關於 EasyWatch</b> ：顯示此版本資訊。

### 35.2.2. 快速工具



設定	描述
	開啟新增
	開啟舊檔
	儲存檔案

	剪下	剪下選取的物件至剪貼簿中。
	複製	複製選取的物件至剪貼簿中。
	貼上	貼上剪貼簿中的物件。
	執行	選擇欲執行的物件，即可執行物件。
	停止	選擇執行中的物件，即可停止物件。
	刪除物件	選擇欲刪除的物件，即可刪除物件。
	監視物件	新增監視物件。
	巨集物件	新增巨集物件。
	HMI 管理器	對 HMI 進行新增、修改、移除的管理。
	說明主題	提供基本功能的操作方法，供使用者參考。

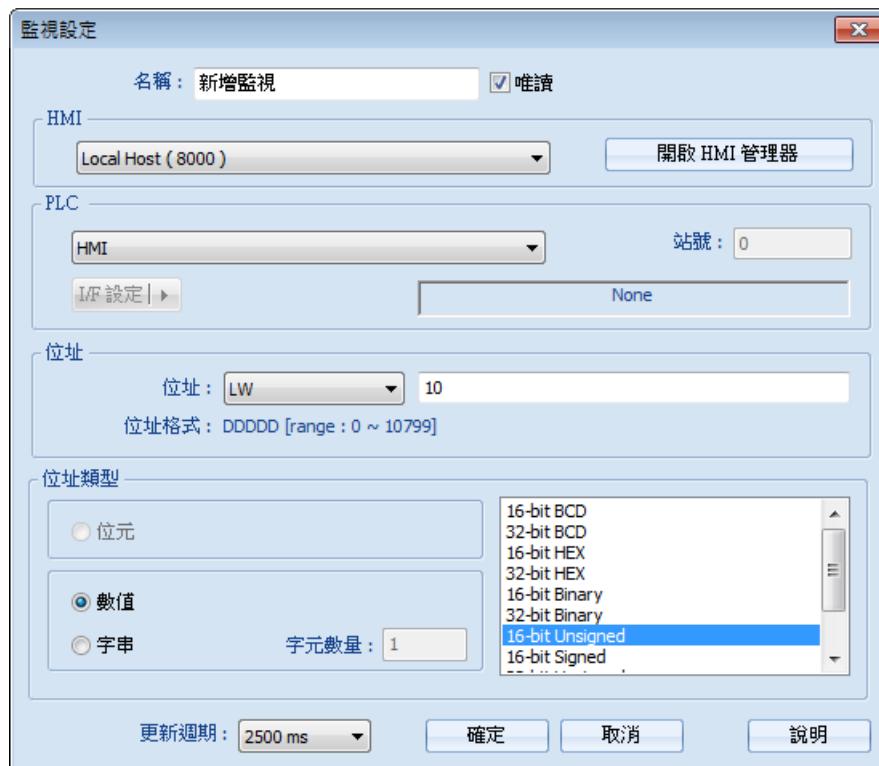
### 35.3. 監視物件設定

#### 35.3.1. 新增監視物件

系統提供兩種方式新增物件：

- 在基本工具列選擇 [物件] » [新增物件] » [新增監視物件]。
- 在快速工具列選擇新增監視物件圖示。

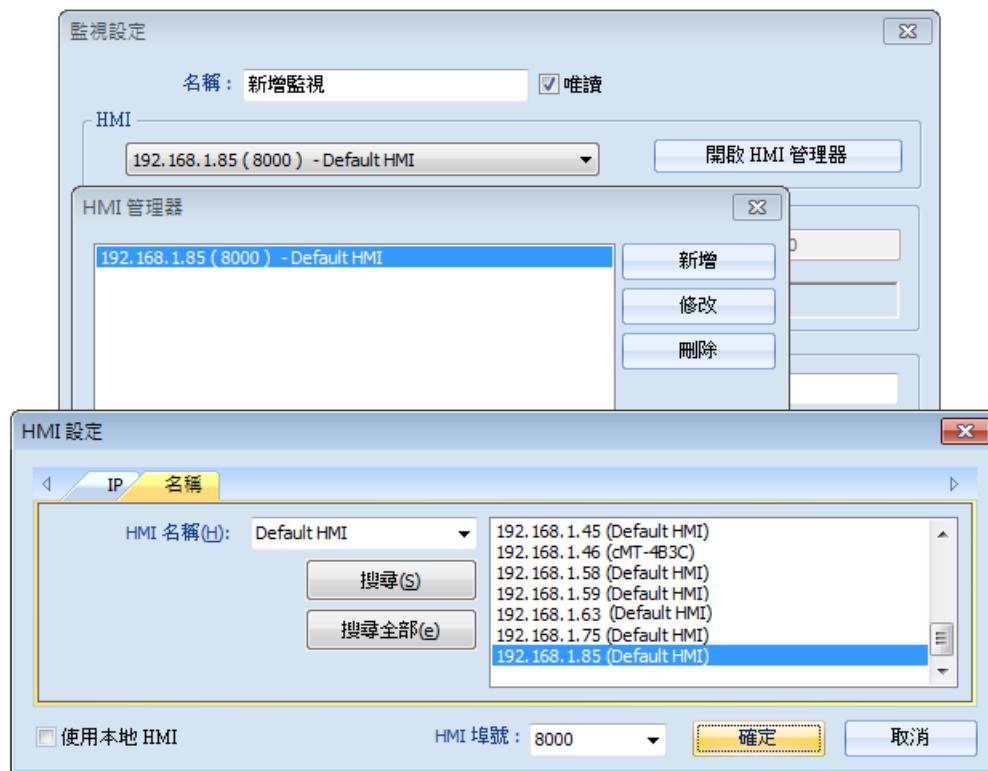
### 35.3.2. 監視物件設定



設定	描述
名稱	對物件命名，名稱不可重複。 <b>唯讀</b> ：若物件設為唯讀時，將不能設定該位址的數值。
HMI	選擇欲監視的 HMI。
PLC	設定欲監視位址所屬 PLC 的類型、站號及連線方式。
位址	設定欲監視位址的類型及位址。
位址類型	將依照位址類型列出可以選擇的顯示方式，執行時會依照顯示方式來解析並顯示該位址。
更新週期	設定監視物件的更新週期，若同時運行過多的物件將導致誤差與延遲。

### 35.3.3. 新增監視物件的步驟

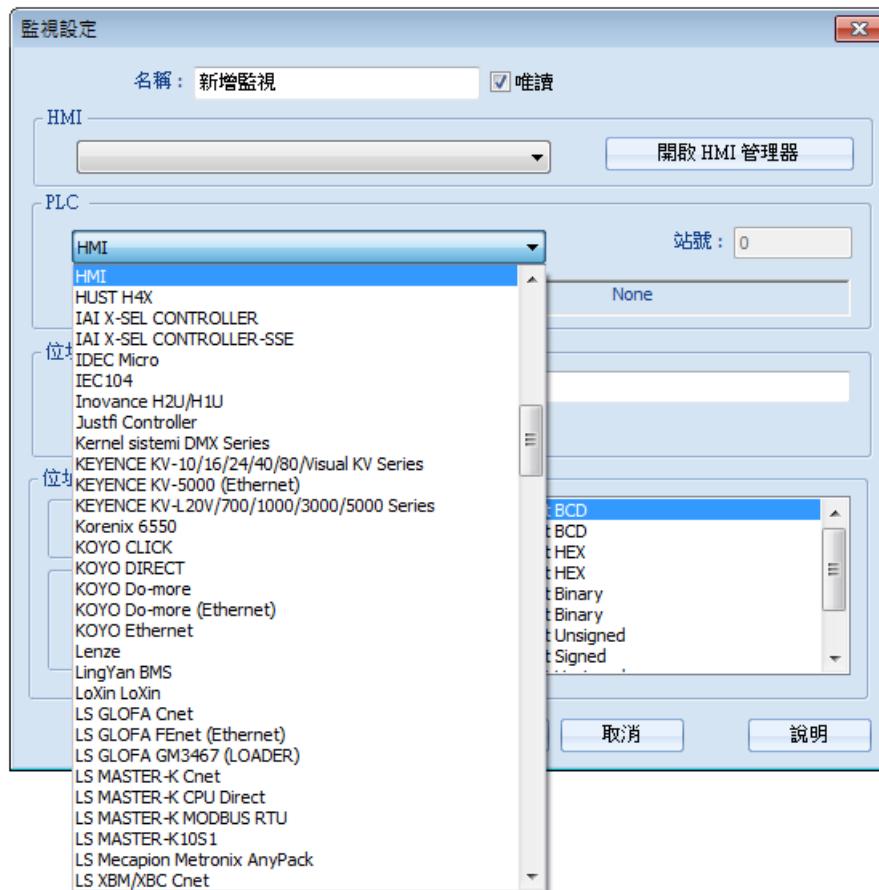
- 選擇欲操作的 HMI，若 HMI 不存在，點選 [開啟 HMI 管理器] » [新增]，可透過網路搜尋 HMI，選擇 [確定] 即可完成新增動作。



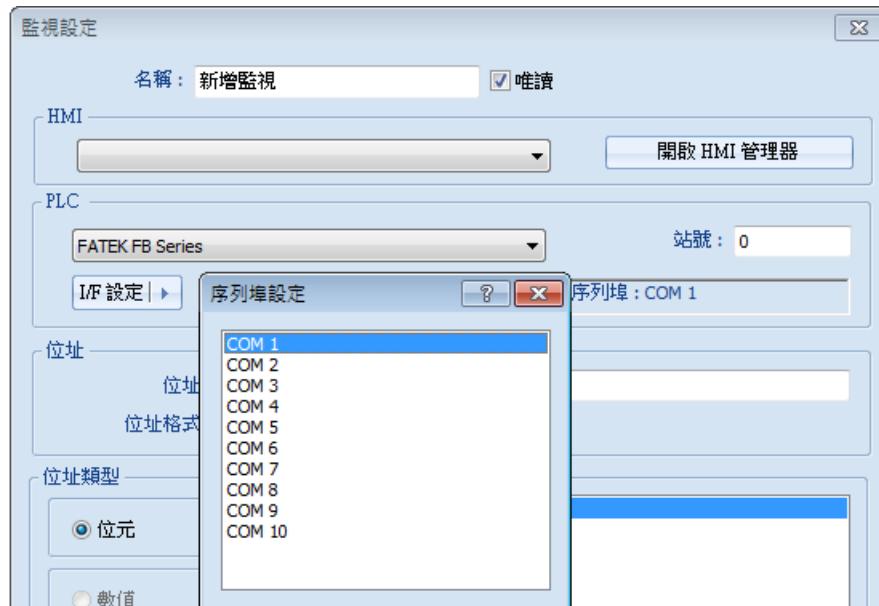
也可勾選 [使用本地 HMI]，將以 PC 上所模擬的程式為監視設備。



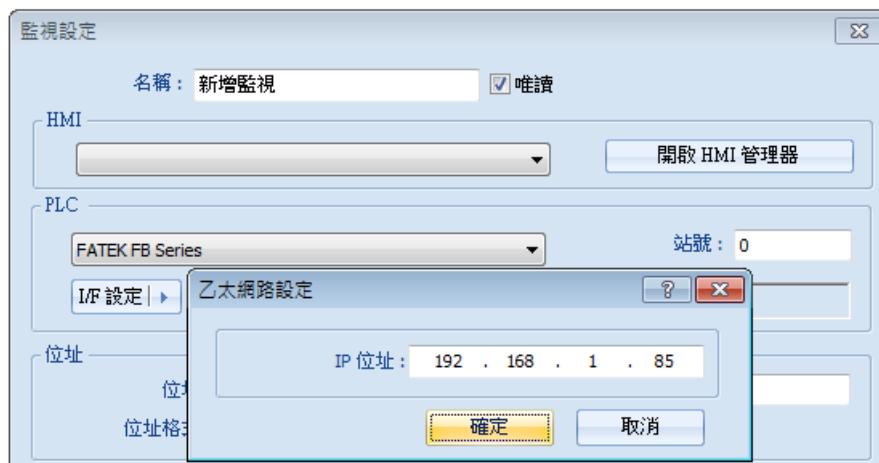
- 選擇直接操作 HMI 或 PLC。若選擇 HMI，即可直接對本機 HMI 操作。



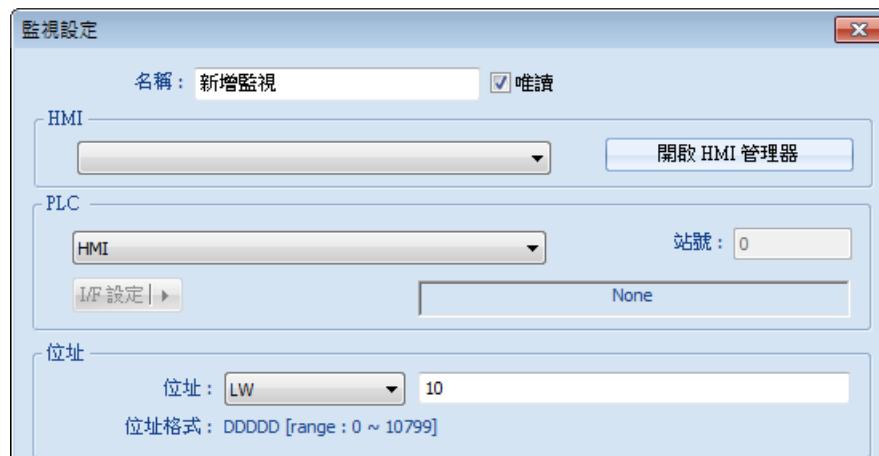
若選擇 PLC 時，PLC 連結方式 (I/F 設定) 可使用 [序列埠]。



也可以選擇 [乙太網路] 並設定 IP 位址。

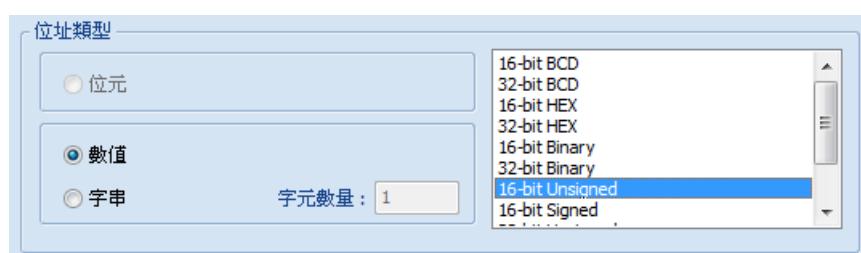


**3.** 設定欲監控的位址及類型。



**4.** 當選用字組類型時，可設定該位址為數值或字串。

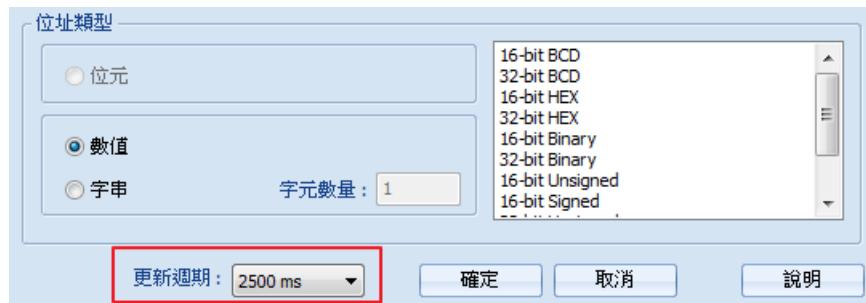
[數值]：可選擇欲監視的位址的資料格式。



[字串]：可設定 ANSI、UNICODE、高/低反向三種格式的資料。並可由 [字組數量] 設定欲讀取的字組數量。



5. 設定監視物件的更新週期。可設定範圍從 500ms 至 5000ms。



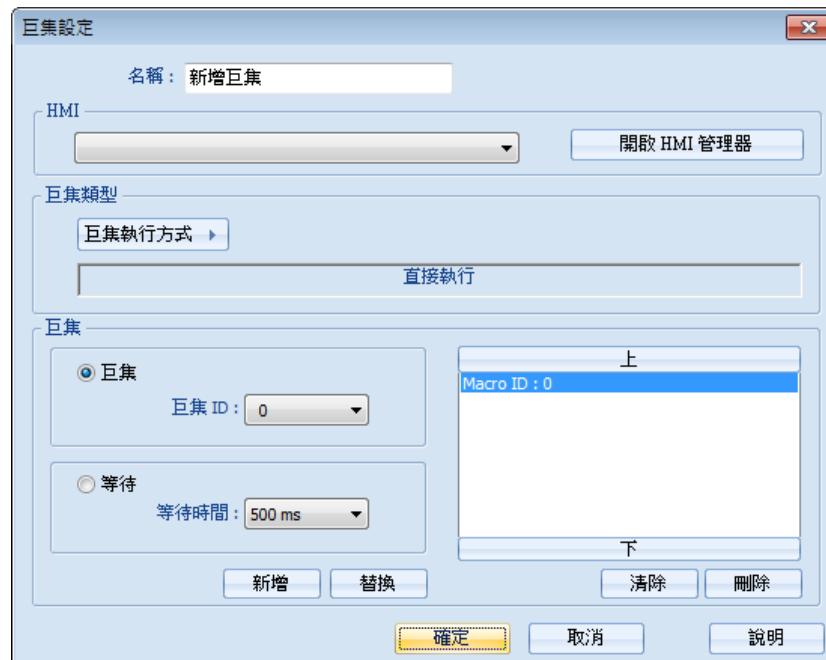
## 35.4. 巨集物件設定

### 35.4.1. 新增巨集物件

系統提供兩種方式新增物件：

- 在基本工具列選擇 [物件] » [新增物件] » [新增巨集物件]。
- 在快速工具列選擇新增巨集物件圖示。

### 35.4.2. 巨集物件設定



設定	描述
名稱	對物件命名，名稱不可重複。
HMI	選擇欲監視的 HMI 設備。
巨集類型	巨集執行的方式可分為直接呼叫和週期呼叫。
巨集	每個巨集物件可執行數個巨集指令，且巨集與巨集執行之間可設定間隔時間。

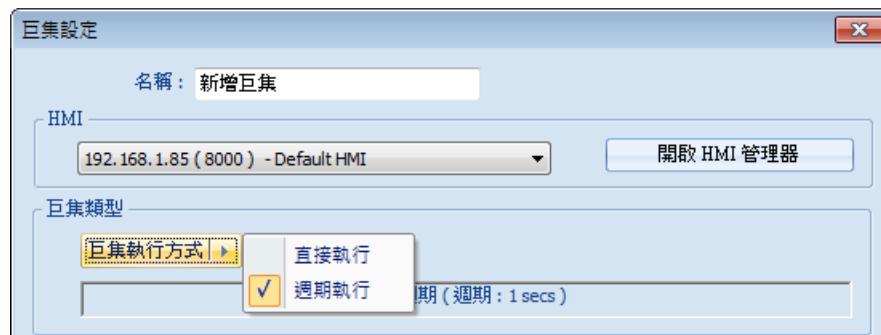
### 35.4.3. 新增巨集設定

1. 選擇 HMI，可參照本章節《35.3.3 新增監視物件的步驟》。

2. 選擇巨集執行方式，可選擇直接執行或週期執行。

[直接執行]：巨集指令會直接執行，並執行一次。

[週期執行]：可設定巨集指令執行的週期時間。



假設在 [執行週期] 設定為 5 秒，當執行完所有巨集指令，將於 5 秒後重新執行此巨集物件。



3. 設定巨集，包含執行 [巨集] 與 [等待] 時間。

[巨集]：選擇要執行的巨集 ID，點擊 [新增] 即可新增到巨集列表中。



[等待]：選擇等待時間，在執行完一個巨集指令後，等待所設定的時間，再執行下一個巨集指令。點選 [新增] 或 [替換] 可新增或取代巨集列表中所選取的巨集。



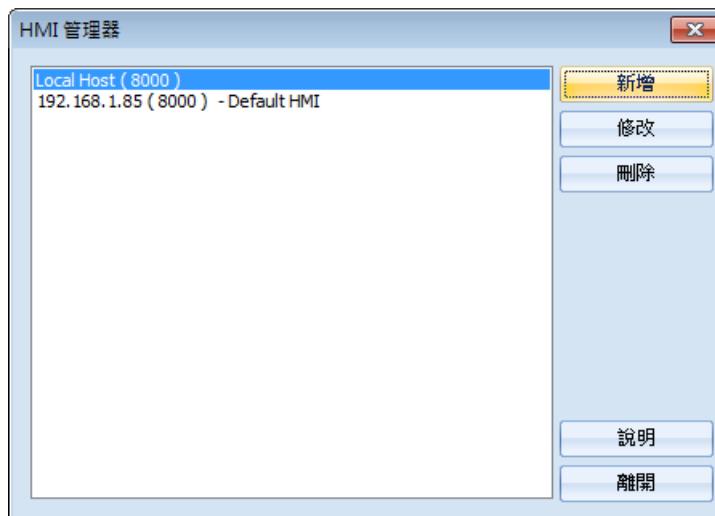
## 35.5. HMI 設定

### 35.5.1. 開啟 HMI 設定

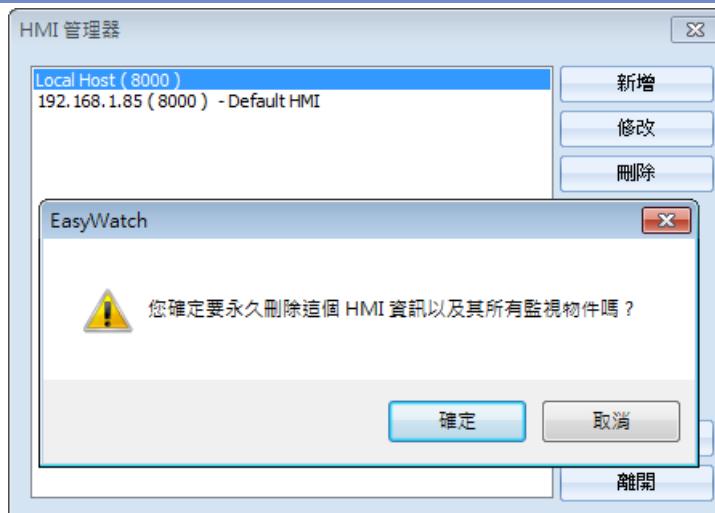
系統提供兩種方式開啟 HMI 設定：

- 在基本工具列選擇 [物件] » [HMI 管理器]。
- 在快速工具列選擇 HMI 管理器圖示。

### 35.5.2. HMI 管理器



設定	描述
新增	選擇 HMI，可參照本章節《35.3.3 新增監視物件的步驟》。.
修改	選擇欲修改 HMI 項目即可修改。
刪除	選擇欲移除 HMI 項目，確認後即可移除。



## 35.6. 物件顯示列表

### 35.6.1. 物件顯示欄位



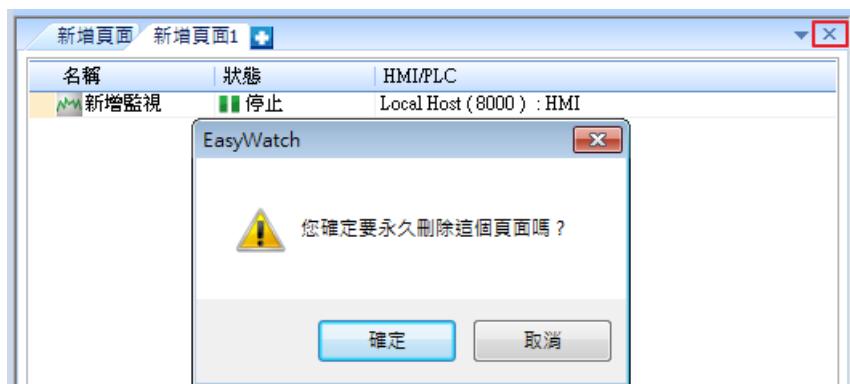
設定	描述
名稱	顯示物件名稱，並且透過圖形顯示，可以方便使用者識別物件的種類。.
狀態	顯示目前物件的執行狀態，分別有 [連結中]、[已連結] 或是 [停止]，同時也可顯示錯誤訊息。若該 HMI 不在線上或是連接埠號輸入錯誤，會顯示“無法發現 HMI”的訊息；若為監視物件且位址設定有誤，則顯示“位址錯誤”的訊息。
HMI/PLC	顯示目前物件所操作 HMI/PLC 的相關資訊。
位址	若為監視物件，將顯示其位址相關設定資料。
位址類型	
更新週期	設定監視物件的更新週期。
數值	若為監視物件，且狀態為已連結時，將顯示目前 HMI 上該位址的數值。當監視物件不具唯讀屬性時，也可透過修改該欄位來設定監視位址的內容。若為巨集物件，並且型態為 [直接執行]，在數值欄位上會顯示按鈕，點擊後可直接執行該巨集指令。

### 35.6.2. 頁面設定

- 新增頁面：點選下面圖示新增頁面。



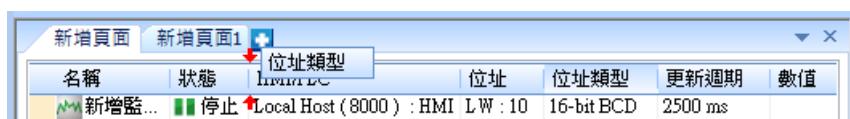
- 刪除頁面：點選下面圖示刪除頁面。



- 重新命名：在頁面的名稱上雙擊滑鼠左鍵後，即可重新命名。



- 變換欄位順序：物件顯示欄位順序可依使用者喜好自行排列選擇。



 請點選此圖示觀看教學影片，請先確定已連上網路線。

# 36. 管理員工具

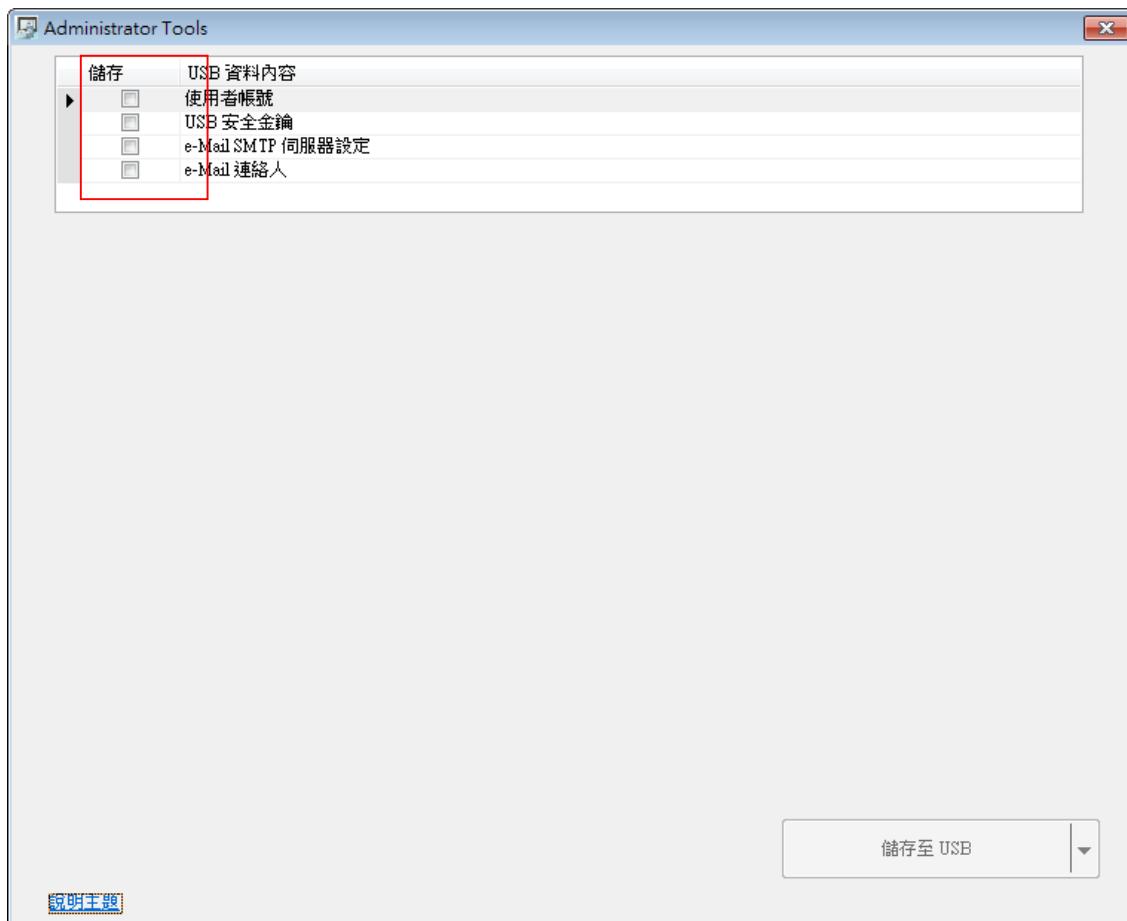
本章節說明如何設定管理員工具。

36.1.	概要	36-2
36.2.	使用者帳號	36-2
36.3.	USB 安全金鑰	36-8
36.4.	e-Mail SMTP 伺服器	36-10
36.5.	e-Mail 聯絡人	36-12

## 36.1. 概要

管理員工具提供將 [使用者帳號]，[USB 安全金鑰]，[e-mail SMTP 伺服器設定]，[e-Mail 連絡人] 四種資料儲存於 USB 碟，並配合 EasyBuilder Pro 使用者帳戶以及 e-Mail 的功能，利用功能鍵的匯入用戶資料/使用[USB 安全金鑰]，可將所建立的資料匯入 HMI，大幅增加資料的可攜性與便利性。

開啟管理員工具軟體後，在儲存核取方塊中打勾，即可開啟該項的編輯功能，後續章節將逐一介紹各個功能的設定。



## 36.2. 使用者帳號

### 36.2.1. 使用者帳號介紹

勾選使用者帳號的核取方塊，即可進行使用者帳戶資料的設定。



設定	描述
隱藏用戶	選擇此帳戶是否為隱藏用戶。
使用者名稱	使用者名稱。*Note 1
密碼	使用者密碼。*Note 1
類別 A ~ L	使用者權限。
新增	新增一筆帳戶資料。*Note 2
刪除	刪除一筆帳戶資料。
匯入	匯入使用者帳戶資料。
匯出	匯出使用者帳戶資料。
有效期限	<p>沒有勾選 [限制使用期限]，將資料匯入 HMI 後，資料是永久有效。</p> <p>若勾選[限制使用期限] 並且設定有效期限後，資料需要在有效期限內匯入 HMI，若是過了有效期限才匯入 HMI，則資料無法被匯入，請重新使用此工具制作一份新的資料。</p>
儲存至 USB	將資料儲存至 USB。若要儲存至目錄，可點選 “▼” 後選擇目錄。





1. 可以是字母，數字符號，“-”，“\_”所構成，大小寫視為不同。
2. 最多可新增 127 筆帳戶資料。

### 36.2.2. 使用者帳號設定

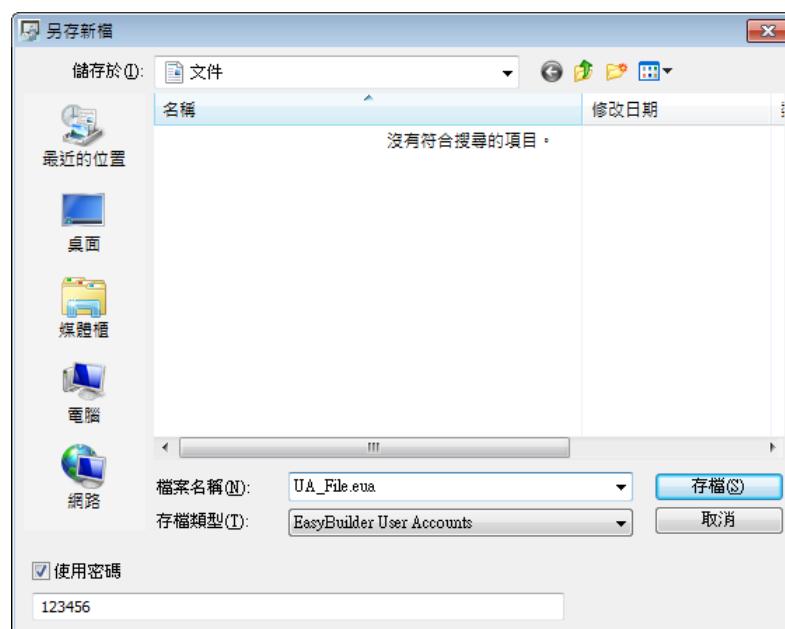
1. 點選 [新增] 可新增一筆使用者資料，選點 [刪除] 即可刪除該筆帳戶資料，勾選 [隱藏用戶] 則該帳號成為隱藏用戶，在 [使用者名稱] 輸入使用者名稱，[密碼] 輸入使用者密碼，並勾選該帳號可使用的權限 [類別 A] ~ [類別 L]。

使用者帳號設定

序號	隱藏...	使用者名稱	密碼	類別 A	類別 B	類別 C	類別 D	類別 E	類別 F	類別 G	類別 H	類別 I	類別 J	類別 K	類別 L
1	<input type="checkbox"/>	Kevin	001001	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	<input checked="" type="checkbox"/>	Fiona	002002	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	Katte	003003	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input checked="" type="checkbox"/>	Barry	004004	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	<input type="checkbox"/>	Susan	005005	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	<input type="checkbox"/>	Carey	006006	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	Paul	007007	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	Ted	008008	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	Jim	009009	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	Mark	010010	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

新增      刪除      匯入...      匯出...

2. 將帳戶資料建立完成之後，點選 [匯出] 可將資料匯出備份，左下角可選擇是否勾選 [使用密碼] 對此資料做保護，日後若需要重新建立或修改只要再點選 [匯入] 並輸入保護密碼，即可匯入該資料。



3. 若勾選了 [有效期限] » [限制有效期限] 則代表限制只有在此範圍時間內，才允許 USB 碟將使用者帳戶資料做匯入 HMI 的動作，若沒有勾選則代表沒有限制範圍，即任何時間皆可以做匯入 HMI 的動作。



4. 全部都設定完成後，點選 [儲存至 USB]，選擇要儲存的 USB 碟位置，點選 [建立]，建立成功會出現 [建立成功!] 的訊息。



### 36.2.3. 使用 EasyBuilder Pro 匯入帳戶

利用 EasyBuilder Pro 所建立的[功能鍵]物件，在動態操作面板觸發此物件時，方可執行此匯入的動作，以下介紹功能鍵建立的步驟：

1. 在 EasyBuilder Pro 選擇[功能鍵]物件，選擇[匯入用戶資料 / 使用 [USB 安全金鑰]，接著按 [設定]。



2. 在 [功能模式] 選擇 [匯入使用者帳號]，[資料位置] 依使用者欲匯入的裝置決定，[帳號匯入方式] 選擇 [覆蓋]， HMI 內將只保存此次匯入的帳號資料，若是選擇 [附加]， HMI 內

帳號資料將保留，並加入此次匯入的新帳號資料；勾選 [匯入使用者帳號後刪除檔案] 在匯入帳號後，將刪除來源資料檔案，接著按 [確定] 即完成設定。



 請點選此圖示下載範例程式。此範例程式說明如何使用功能鍵物件匯入使用者帳號。下載範例程式前，請先確定已連上網路線。

#### 36.2.4. 使用 EasyWeb 2.0 匯入帳戶到 cMT/cMT X 人機

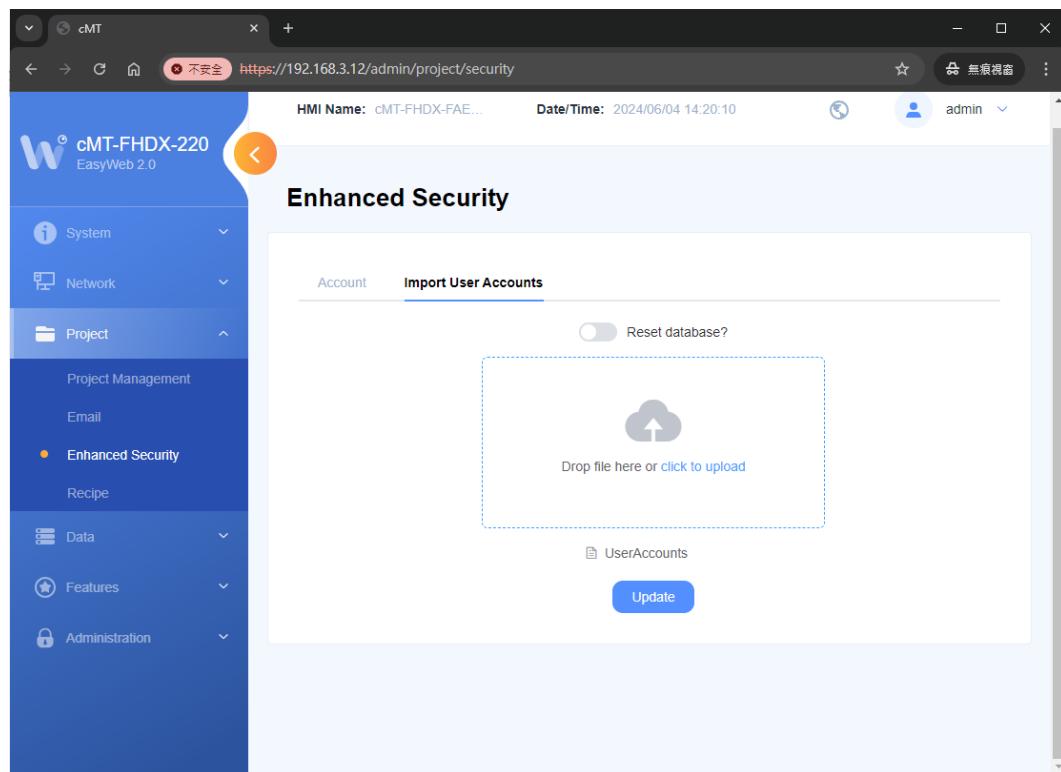
1. 將帳戶資料建立完成之後，點選 [儲存至資料夾]，存成 UserAccounts 檔案。



2. 在瀏覽器輸入人機 IP 進入 EasyWeb 2.0，在 Enhanced Security » Import User Accounts 頁面上傳 User Accounts 檔案，如果有重複的使用者匯入會失敗，可以勾選 [Reset database] 覆蓋就

## 管理員工具

可以成功匯入。



- 在 Account 頁面按下 [Save]，就可以新增使用者帳戶到人機。

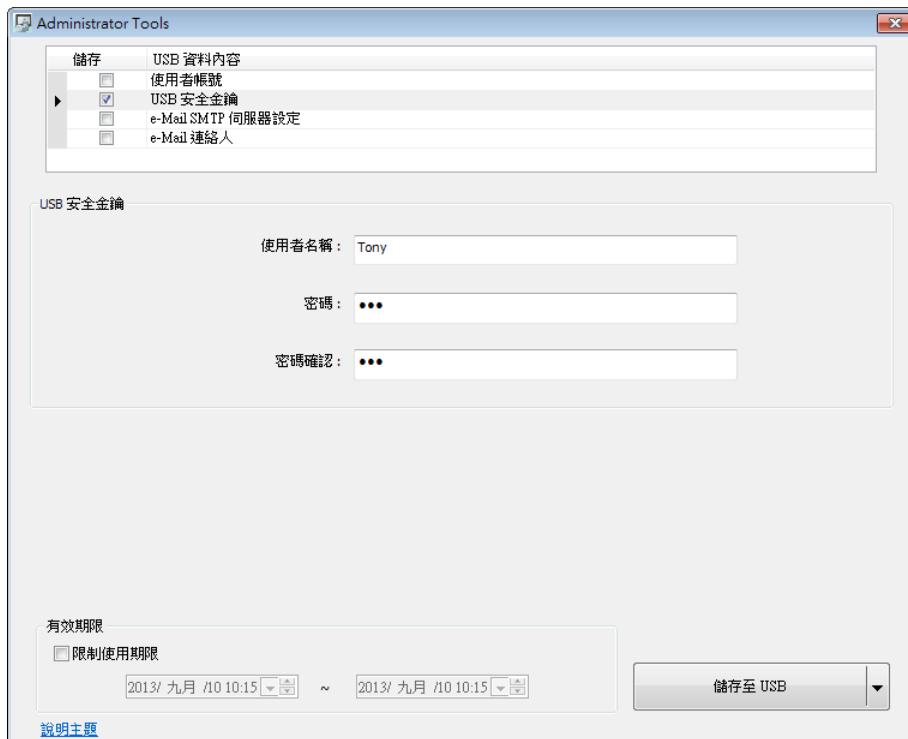
The screenshot shows the 'Account' page with the 'Add Account' and 'Save' buttons visible. A message at the top says 'Please click [Save] button after editing data'. Below is a table with columns: ID, Hide, Account, Password, A, B, C, D, E, F, G, H, I. The table contains 8 rows of user data:

ID	Hide	Account	Password	A	B	C	D	E	F	G	H	I
1		admin	111111	✓	✓	✓	✓	✓	✓	✓	✓	✓
2		001001	001001	✓								
3		002002	002002	✓	✓							
4		003003	003003	✓	✓	✓						
5		004004	004004	✓	✓	✓	✓					
6		005005	005005	✓	✓	✓	✓	✓	✓			
7		006006	006006	✓	✓	✓	✓	✓	✓	✓		
8		007007	007007	✓	✓	✓	✓	✓	✓	✓		

### 36.3. USB 安全金鑰

#### 36.3.1. USB 安全金鑰介紹

勾選 [USB 安全金鑰] 核取方塊即可進行使用者安全金鑰的設定，預先設定好使用者的登錄資訊，即可利用使用者安全金鑰設定直接登入帳戶，設定畫面如下所示：



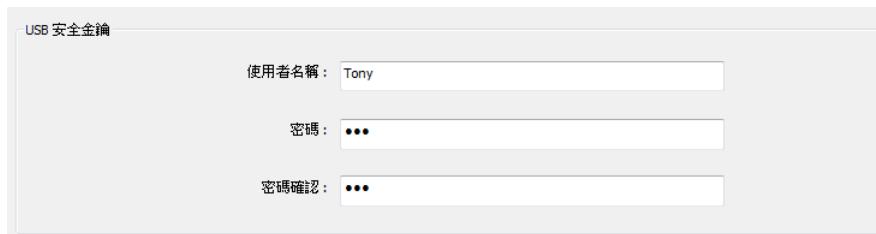
設定	描述
使用者名稱	使用者名稱。*Note 1
密碼	使用者密碼。*Note 1
密碼確認	確認使用者密碼。
有效期限	在此有效期限內可以在 HMI 上使用安全金鑰登入，若沒有勾選則代表無限制。
儲存至 USB	將資料儲存至 USB。

#### Note

- 可以是字母，數字符號，“-”，“\_”所構成，大小寫視為不同。

#### 36.3.2. USB 安全金鑰設定

- 在 [使用者名稱] 輸入使用者的帳戶名稱，[密碼] 輸入使用者的密碼，[密碼確認] 重覆輸入密碼確認。



2. 若勾選了 [有效期限] » [限制有效期限] 則代表限制只有在此範圍時間內，才允許使用者用此 USB 安全金鑰登入功能，若沒有勾選則代表沒有限制，任何時間皆可以執行此功能。



3. 全部都設定完成後，點選 [儲存至 USB]，選擇要儲存的 USB 碟位置，點選 [建立]，建立成功會出現 [建立成功!] 的訊息。



### 36.3.3. 使用 EasyBuilder Pro 設定 USB 安全金鑰

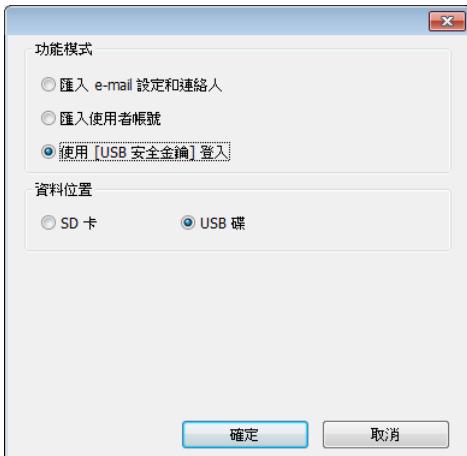
利用 EasyBuilder Pro 所建立的 [功能鍵] 物件，在動態操作面板觸發此物件時，方可執行 USB 安全金鑰登入的動作，以下介紹 [功能鍵] 建立的步驟。

1. 在 EasyBuilder Pro 選擇 [功能鍵] 的物件，選擇“匯入用戶資料/使用[USB 安全金鑰]”，接著按 [設定]。

**匯入用戶資料/使用 [USB 安全金鑰]**

**設定...**

2. 在 [功能模式] 選擇 “使用 [USB 安全金鑰登入]”，[資料位置] 依安全金鑰檔案存放的裝置決定，按 [確定] 即完成設定。

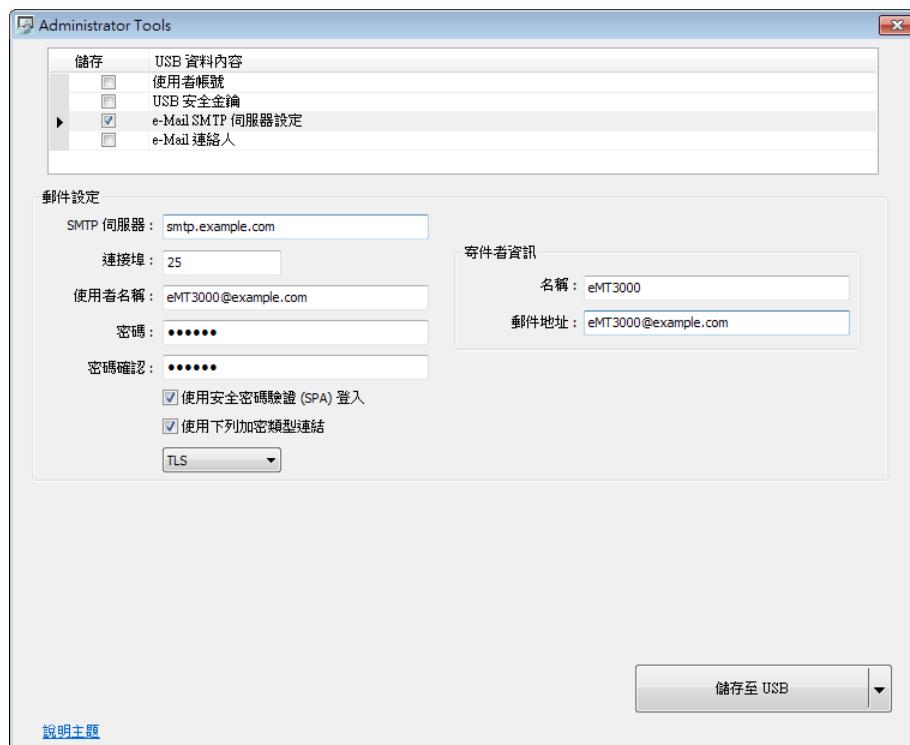


請點選此圖示下載範例程式。此範例程式說明如何透過功能鍵物件使用 USB 安全金鑰登入。

下載範例程式前，請先確定已連上網路線。

#### 36.4. e-Mail SMTP 伺服器

勾選 [e-Mail SMTP 伺服器設定] 的核取方塊，即可進行外寄郵件伺服器的設定。



郵件設定	描述
<b>SMTP 伺服器</b>	外寄郵件伺服器。
<b>連接埠</b>	外寄郵件伺服器連接埠號碼。
<b>使用者名稱</b>	使用者郵件帳號名稱。
<b>密碼</b>	使用者郵件帳號密碼。
<b>密碼確認</b>	確認使用者郵件帳號密碼。
寄件者設定	描述
<b>名稱</b>	收件時顯示寄件者的名稱。
<b>郵件地址</b>	收件時顯示寄件者的郵寄地址。
<b>儲存至 USB</b>	將資料儲存至 USB。

#### 36.4.1. e-Mail SMTP 伺服器設定

1. 以下依照 e-mail SMTP 設定內容。



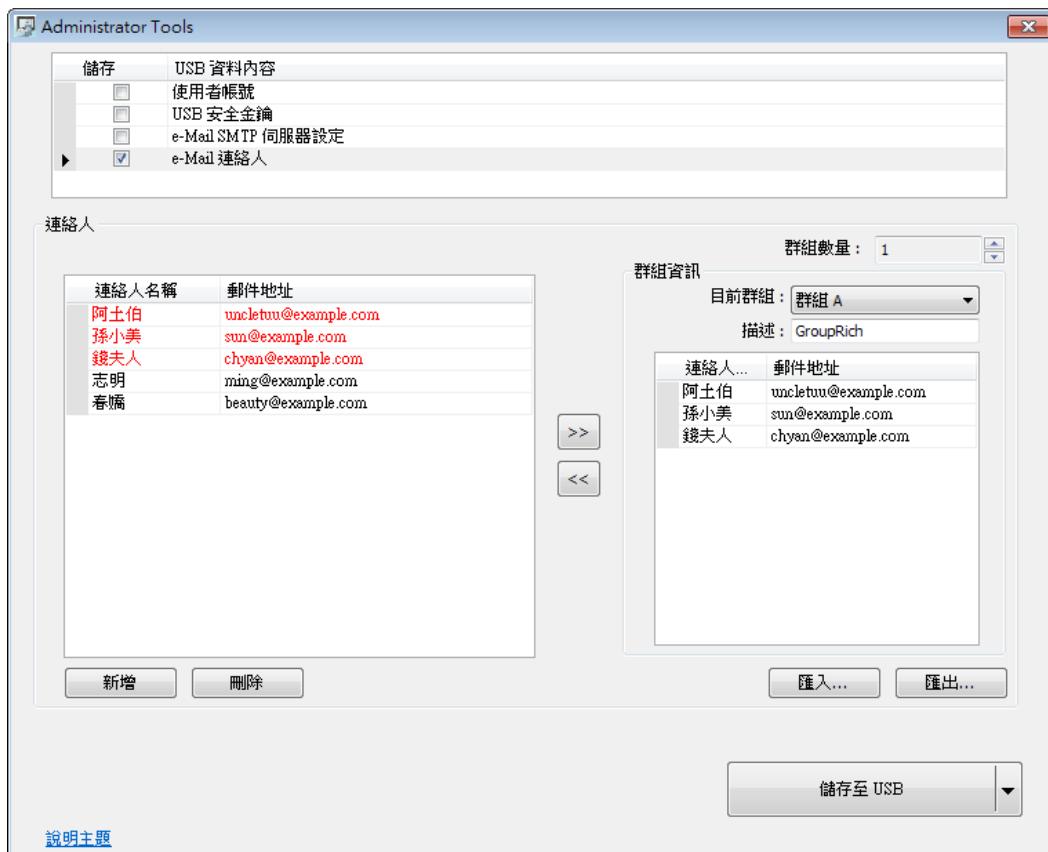
2. 全部都設定完成後，點選 [儲存至 USB]，選擇要儲存的 USB 碟位置，點選 [建立]，則會出現 [建立成功!] 的訊息。



## 36.5. e-Mail 聯絡人

### 36.5.1. e-Mail 聯絡人介紹

勾選 [e-Mail 連絡人] 的核取方塊，即可設定郵件聯絡人。



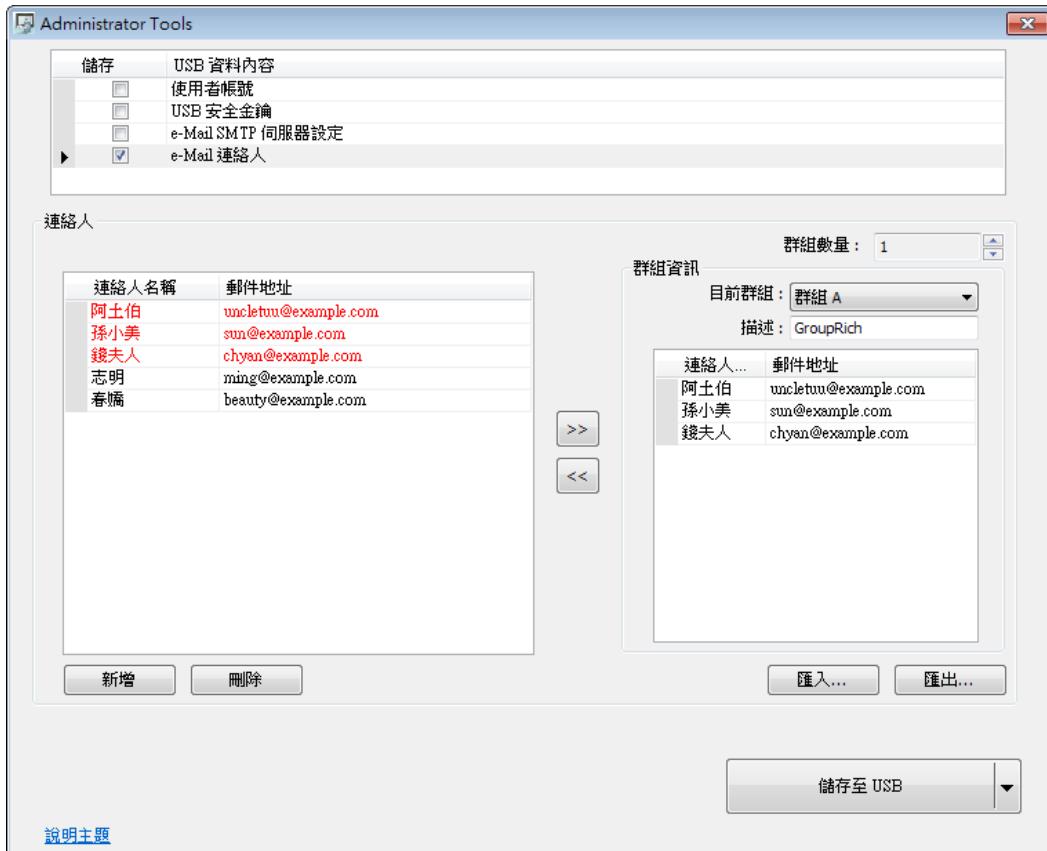
設定	描述
新增	新增一筆聯絡人資料。 *Note 1
刪除	移除一筆聯絡人資料。
群組數量	群組數目。 *Note 2
目前群組	目前群組名稱。 *Note 3
描述	群組描述。
匯入	匯入聯絡人資訊。
匯出	匯出聯絡人資訊。
儲存至 USB	將資料儲存至 USB。

### Note

1. 最多可新增 256 筆聯絡人資料
2. 最多可新增 16 個群組 (群組 A ~ 群組 P)
3. 群組 A ~ 群組 P , 當群組數量為 1 時，只會存在群組 A ，增加至 2 時，則會同時存在群組 A 與群組 B ，以此類推。
4. 聯絡人名稱可以輸入 UTF-8 的字型。

### 36.5.2. e-Mail 聯絡人設定

1. 按 [新增] 新增完成所有郵件通知對象。
2. 將連絡人加入到群組 A 之中，可以看到被加入到此群組的連絡人顯示的顏色是紅色。



3. 設定群組數量，即可新增群組，此時會新增一個群組 B。重覆步驟 a 與步驟 b 將聯絡人加入群組之中。



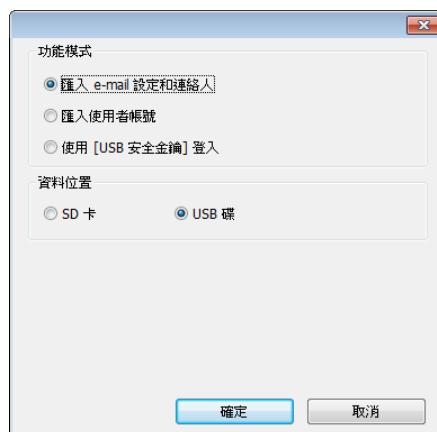
4. 將 e-Mail 連絡人資料建立完成之後，點選 [匯出] 可將資料匯出備份，日後若需要重新建立或修改只要再點選 [匯入] 匯入該資料即可。
5. 全部都設定完成後，點選 [儲存至 USB]，選擇要儲存的 USB 碟位置，點選 [建立]，建立成功會出現 [建立成功!] 的訊息。



### 36.5.3. 使用 EasyBuilder Pro 汇入 e-Mail 設定和連絡人

利用 EasyBuilder Pro 所建立的 [功能鍵] 物件，在動態操作面板觸發此物件時，方可執行此匯入的動作，以下介紹 [功能鍵] 建立的步驟。

1. 在 EasyBuilder Pro 選擇 [功能鍵] 的物件，選擇“匯入用戶資料/使用[USB 安全金鑰]”，接著按 [設定]。
2. 在 [功能模式] 選擇 [匯入 e-mail 設定和連絡人]，[資料位置] 依使用者欲匯入的裝置決定，接著按 [確定] 即完成設定。



 請點選此圖示下載範例程式。此範例程式介紹如何透過功能鍵匯入 e-mail 設定和連絡人。

下載範例程式前，請先確定已連上網路線。

# 37. MODBUS 閘道功能

本章節說明如何使用 MODBUS 閘道功能並建立位址對應表。

37.1. 概要 .....	37-2
37.2. 如何建立一個位址對應表 .....	37-2
37.3. 位址對應設定須知 .....	37-5

## 37.1. 概要

以往若要使用 SCADA (*Supervisory Control and Data Acquisition*) 軟體去存取與 HMI 連接的 PLC 資料時，需透過資料傳輸先將 PLC 資料傳送至 HMI 的本地位址，再於 PC 上使用 MODBUS TCP/IP 或 MODBUS RTU 通訊協議去讀取 HMI 的本地位址將 PLC 資料取回。

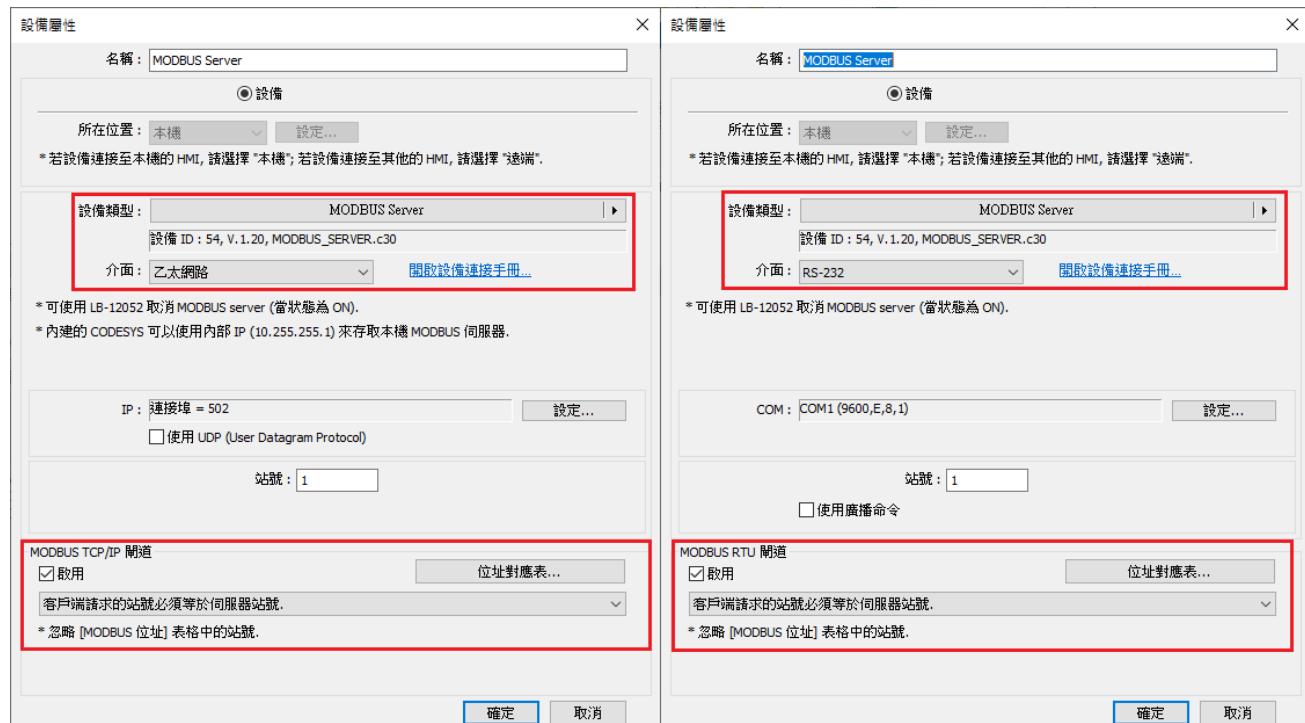
現在用戶可以透過 EasyBuilder Pro 提供的 MODBUS 閘道功能，將 MODBUS 與 PLC 的位址預先設定對應後，即可以直接利用 MODBUS TCP/IP 或 MODBUS RTU 通訊協議存取 PLC 上的資料。



## 37.2. 如何建立一個位址對應表

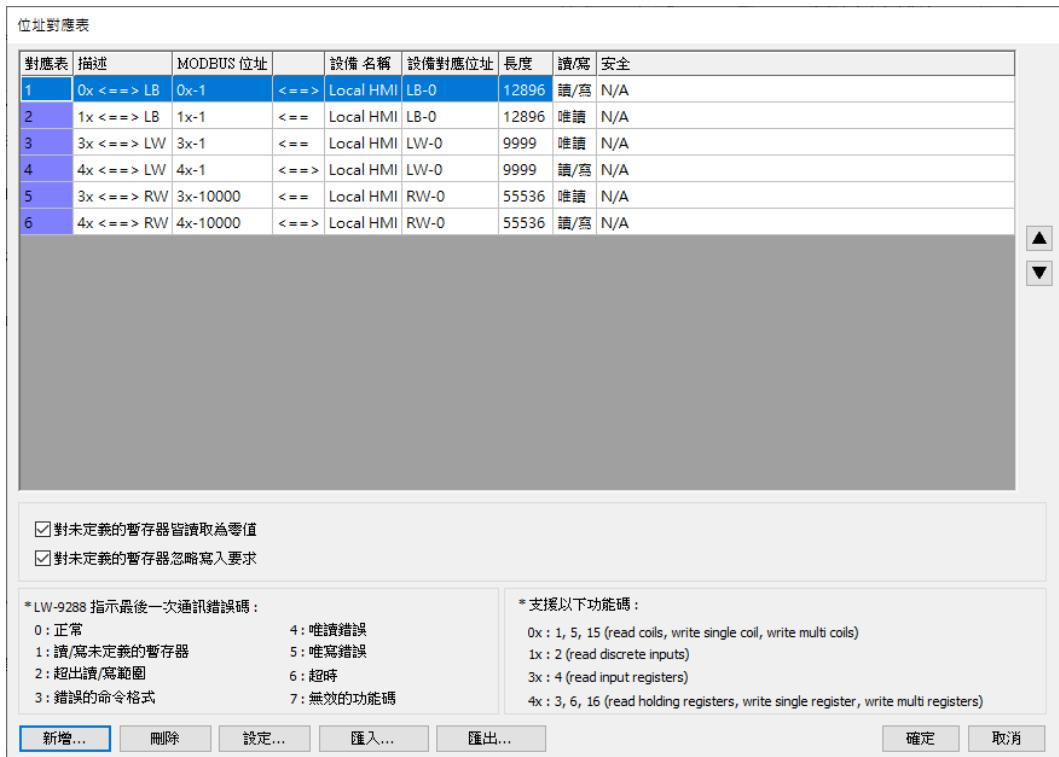
新增一個位址對應表，請依照下列步驟：

1. 於 [系統參數設定] » [設備清單] 新增欲監控的 PLC 設備。(以 FATEK FB Series 為例)
2. 新增一個 MODBUS Server，介面可根據 MODBUS Client 的通訊介面選擇使用乙太網路或 COM 通訊埠，並啟用 [MODBUS TCP/IP 閘道] 或 [MODBUS RTU 閘道]，如下圖所示：



## MODBUS 閘道功能

3. 點選【位址對應表】按鈕後，會顯示預設的對應表，用戶可以依需求修改並新增其他對應表。



4. 假設，SCADA 需存取 FATEK FB Series PLC 的 DO 暫存器開始的連續 50 個位址，設定如下：



- (1) 設定欲對應的暫存器之位址模式，此範例為 [字組]。
- (2) 設定欲對應的暫存器之存取模式，此範例為 [讀/寫]。
- (3) 設定欲對應的 MODBUS 起始位址，此範例為 [4x1]。
- (4) 設定欲對應的 PLC 起始位址，此範例為 [D0]。
- (5) 設定欲對應位址的範圍長度，此範例為 [50]。
- (6) 選擇是否要高/低位元組或高/低字組轉換。

對應表	描述	MODBUS 位址		PLC 名稱	PLC 對應位址	長度	讀/寫
1	Access D0 ~ D49	4x-1	<=>	FATEK FB Series	D-0	50 字元	讀/寫

上圖的設定內容說明 MODBUS Server 4x1 ~ 4x50 位址對應到 FATEK FB Series PLC 的 D0 ~ D49 位址。

5. 完成以上設定後，SCADA 只需利用 MODBUS 協議，發送讀/寫 4x1 ~ 4x50 位址的命令，即可以直接存取 FATEK FB Series PLC 的 D0 ~ D49 位址。

### 37.3. 位址對應設定須知

- [MODBUS TCP/IP 閘道] 功能不支援使用 UDP。
- [MODBUS RTU 閘道] 功能只支援於 cMT/cMT X 系列。
- MODBUS 客戶端請求命令中的通訊站號，根據設備配置與需求，在以下三種模式中選擇最適合的模式：  
**客戶端請求的站號必須等於伺服器站號**  
站號設定以 MODBUS Server 站號為主。

客戶端請求的站號必須等於伺服器站號。  
 \* 忽略 [MODBUS 位址] 表格中的站號。

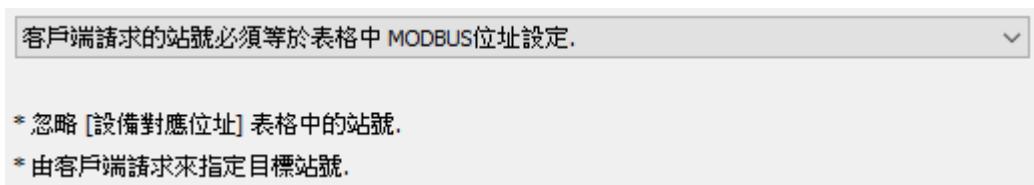
#### 使用客戶端請求的站號

站號設定以外部連線的 MODBUS Client 站號為主。不能同時有多台連到 MODBUS Server 且使用不同站號。

使用客戶端請求的站號。  
 \* 忽略 [MODBUS 位址] 表格中的站號。  
 \* 忽略 [設備對應位址] 表格中的站號。  
 \* 由客戶端請求來指定目標站號。

#### 客戶端請求的站號必須等於表格中 MODBUS 位址設定

站號設定以位址對應表的 MODBUS 位址為主。可設定成 4x-1#100, 4x-2#100 代表有兩台 MODBUS Client 使用站號 1 與站號 2 與 MODBUS Server 連線通訊，表示可以同時有多台連線。若在 MODBUS 位址沒有設定站號則會使用設備預設站號。



- 新增 MODBUS Server 的更新時間(秒)設定，讓 MODBUS Server 具備快取功能，可處理多台客戶端頻繁讀取的情況。預設 0 秒代表不啟用快取機制，設定時間大於 0 秒則表示啟用快取機制。假設更新時間設為 1 秒，當 Server 第一次收到客戶端的命令，會執行轉發去取得 PLC 資料，但在這 1 秒內，如果客戶端再次詢問相同暫存器的資料，Server 將會從快取中拿之前取得的 PLC 資料回覆給客戶端。



- 當客戶端設備的暫存器並非連續，但客戶又希望可以一次連續讀寫全部資訊的時候，針對在位址對應表中未定義的暫存器，新增以下兩個選項，讓讀寫命令可以正常運作。  
僅 cMT/cMT X 系列支援此功能。

- 對未定義的暫存器皆讀取為零值
- 對未定義的暫存器忽略寫入要求

- 當位址定義為 32-bit 暫存器，客戶端不支援單獨讀寫該暫存器中的 16-bit (高/低字組)資料。
- 系統提供暫存器 LW-9288，可用來指示此功能資料傳送是否正常。

各錯誤碼表示如下：

數值	定義
0	正常
1	讀取或寫入未定義在位址對應表中的暫存器
2	讀取或寫入的位址範圍超出單一位址對應表所定義的 數據長度 (或是讀取/寫入跨表格的暫存器)
3	命令格式未遵循 MODBUS TCP/IP 通訊協議
4	修改只允許讀取的暫存器

- 
- 5** 讀取只允許寫入的暫存器
- 
- 6** 在設定的時間內無法得到 PLC 的正確回應
- 
- 7** 使用了 MODBUS Server 不支援的功能碼
- 

- 啟用 [MODBUS 閘道] 功能後，EasyBuilder Pro 將取消 MODBUS Server 與 HMI 位址間原有的對應關係，包含：
- (1) 0x, 1x 對應到 LB
  - (2) 3x, 4x 對應到 LW, RW

因此如需透過 0x, 1x, 3x, 4x 的命令來存取 LB 或 LW 的資料，仍需先將位址對應關係重新設定於 [位址對應表] 中，可參考下列設定內容。

對應表	描述	MODBUS 位址		PLC 名稱	PLC 對應位址	長度	讀/寫
1	0x <==> LB	0x-1	<==>	Local HMI	LB-0	12400 位元	讀/寫
2	1x <==> LB	1x-1	<==>	Local HMI	LB-0	12400 位元	唯讀
3	3x <==> LW	3x-1	<==>	Local HMI	LW-0	9999 字元	唯讀
4	4x <==> LW	4x-1	<==>	Local HMI	LW-0	9999 字元	讀/寫
5	3x <==> RW	3x-10000	<==>	Local HMI	RW-0	55536 字元	唯讀
6	4x <==> RW	4x-10000	<==>	Local HMI	RW-0	55536 字元	讀/寫

# 38. EasyDownload

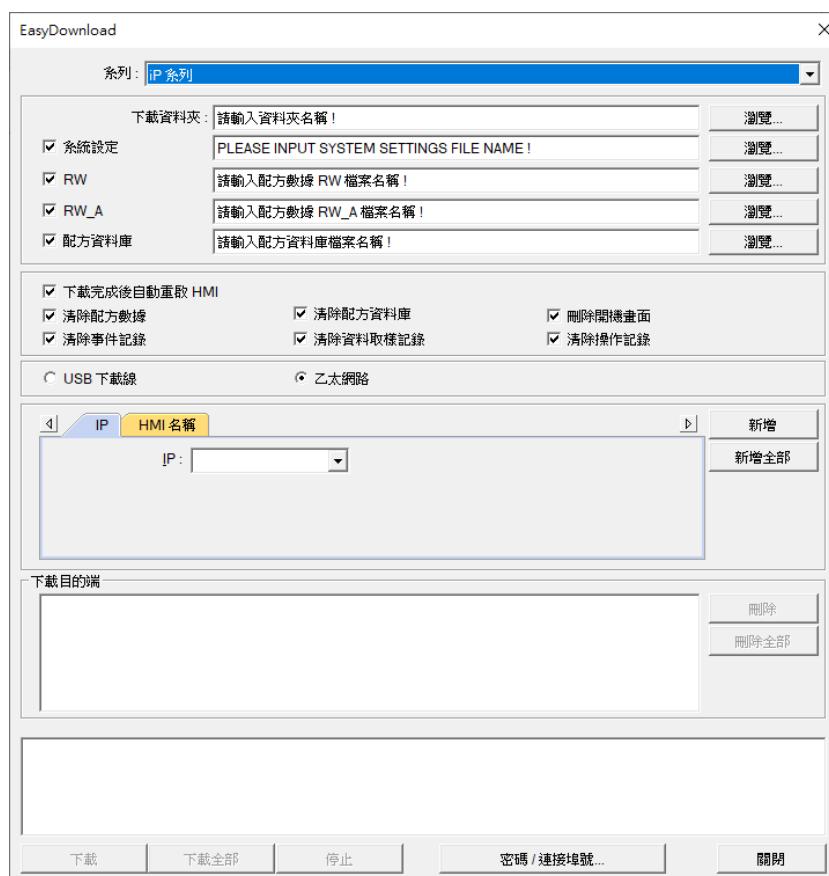
本章節說明如何使用 EasyDownload 程式。

38.1. 概要 .....	38-2
38.2. 設定 .....	38-2

## 38.1. 概要

EasyDownload 程式可以透過乙太網路或 USB 線來下載於 EasyBuilder Pro 編輯軟體建立的工程檔案。需先透過 EasyBuilder Pro 編輯軟體的 [工程檔案] 選單 » [建立使用在 USB 碟與 SD 卡所需的下載資料] 建立工程檔案下載資料。

## 38.2. 設定



設定	描述
<b>系列</b>	選擇欲下載的人機系列機型。
<b>下載資料夾</b>	選擇欲下載的工程檔案下載資料。 cMT Series / cMT X Series / IIoT Gateway Series 機型下載需選擇.cxob 檔；非 cMT 機型需選擇對應的資料夾。
<b>配方數據 RW</b>	若勾選，可瀏覽欲下載的配方數據 (.rcp) 檔案。
<b>配方數據 RW_A</b>	若勾選，可瀏覽欲下載的配方數據 (.rcp) 檔案。
<b>配方資料庫</b>	若勾選，可瀏覽欲下載的配方資料庫 (.db) 檔案。
<b>下載完成後自動啟動 HMI</b>	若勾選，下載程序完成後會自動重新啟動 HMI。

**清除配方數據****清除配方資料庫****刪除開機畫面**

若勾選，下載程序進行前會先清除 HMI 上所選取的資料。

**清除事件記錄****清除資料取樣記錄****清除操作記錄****USB 下載線**

使用 USB 線傳送檔案至 HMI。

請先確認 USB 驅動是否正確安裝至 PC。

**乙太網路**

使用乙太網路傳送檔案至 HMI。

**IP**

輸入下載目標 HMI 的 IP 位址。

**名稱**

輸入下載目標 HMI 的名稱。

**搜尋**

搜尋指定的 HMI 名稱。

**搜尋全部**

搜尋在同一區網上的所有 HMI 名稱。

**新增**

新增搜尋狀態內所選擇的 HMI 至下載目的端。

**新增全部**

新增搜尋狀態內所有 HMI 至下載目的端。

**下載目的端**

欲下載工程檔案的列表。

**刪除**

刪除下載目的端所選擇的 HMI。

**刪除全部**

刪除下載目的端所有 HMI。

**下載**

點選後將開始下載程序。

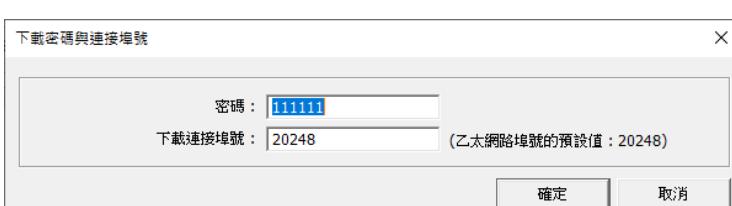
**下載全部**

將下載目的端的所有 HMI 依序執行下載程序。

**密碼/連接埠號**

輸入 HMI 系統設定裡所設定之下載密碼與連接埠號。

非 cMT 系列:

**cMT / cMT X 系列:****Note**

- 以 eMT3000 系列為範例，建立完成的下載資料夾格式如下所示，請選擇第一層資料夾名稱作為下載路徑。

第一層資料夾	第二層資料夾	第三層資料夾
emt3000	001	
	002	
	Pub	driver
		font

- 第一層資料夾的名稱將依型號不同而有所變化。
- 只有乙太網路支援多屏下載。
- 當一次下載工程檔案至多台 HMI 時，所有的 HMI 必須使用相同的密碼與連接埠號。
- 工程檔案是依序執行下載程序，因此必須在此刻被更新的 HMI 完成下載後，才會繼續執行下一個 HMI 的下載工作。
- 為了避免因為某一台設備不在線導致過久的等待時間，等待時間將為 3 秒。
- EasyDownload 軟體限制最多一次 512 台下載，也就是下載目的端內的人機數量最多可以新增 512 台。

# 39. 資料保護

本章節說明如何使用【資料保護】功能。

39.1. 概要 .....	39-2
39.2. 設定 .....	39-2

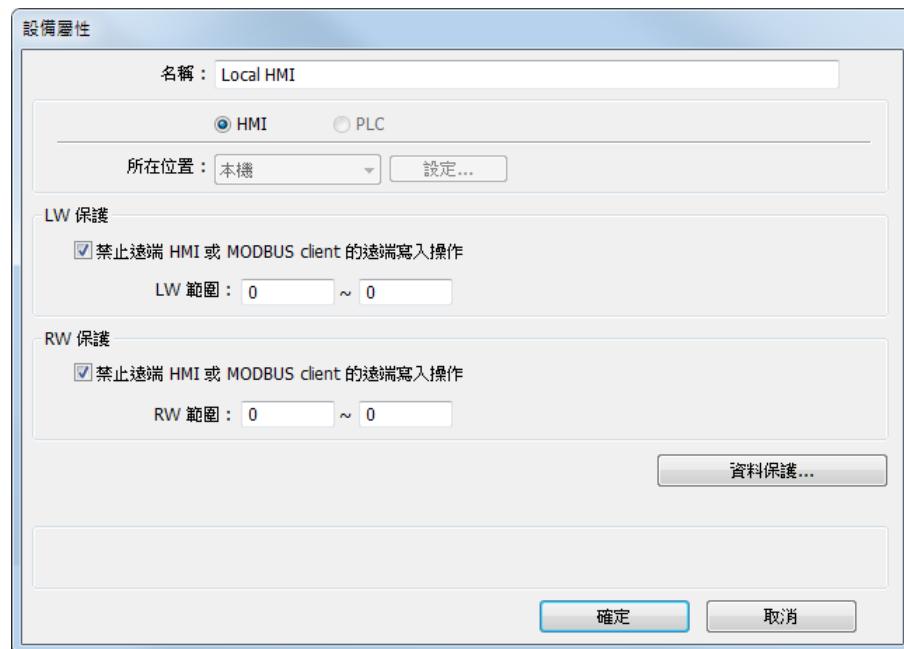
### 39.1. 概要

本機 HMI 的字組或位元位址可以自訂操作時的限制。若欲設定此功能，請在 [系統參數設定] » [設備清單] 頁籤中點選 [本機 HMI] 並點選 [保護] 按鈕。



### 39.2. 設定

設定對話窗如下：

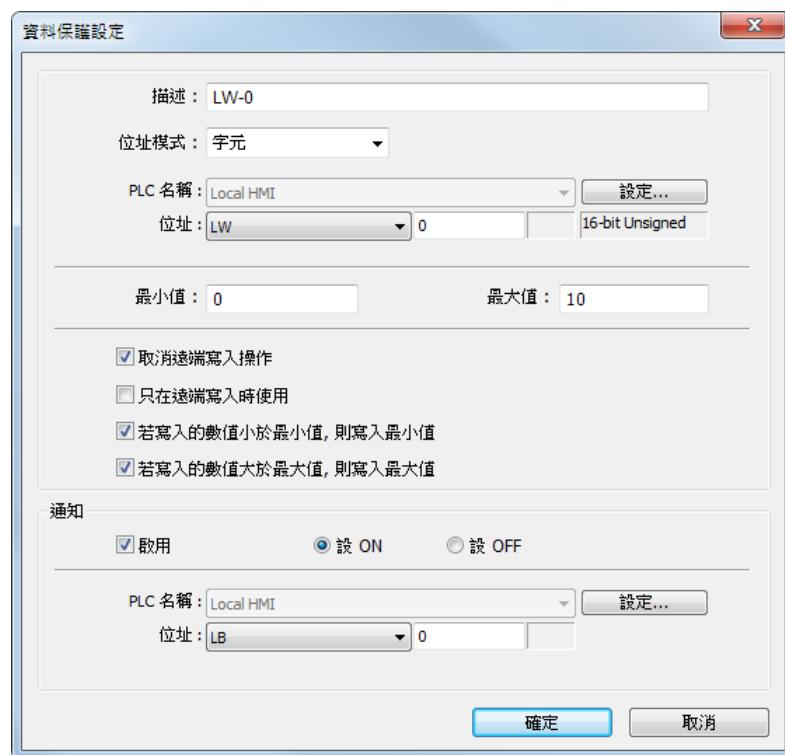


設定	描述
<b>LW 保護</b>	
<b>禁止遠端 HMI 或 MODBUS client 的遠端寫入操作</b>	可禁止遠端 HMI 或 MODBUS client 寫入指定的 LW 位址。
<b>RW 保護</b>	
<b>禁止遠端 HMI 或 MODBUS client 的遠端寫入操作</b>	可禁止遠端 HMI 或 MODBUS client 寫入指定的 RW 位址。

點選【資料保護】按鈕後，可針對本地字組位址或位元位址設定寫入操作時的限制。

### 39.2.1. 字組位址屬性設定

設定與本地字組位址相關的操作限制。

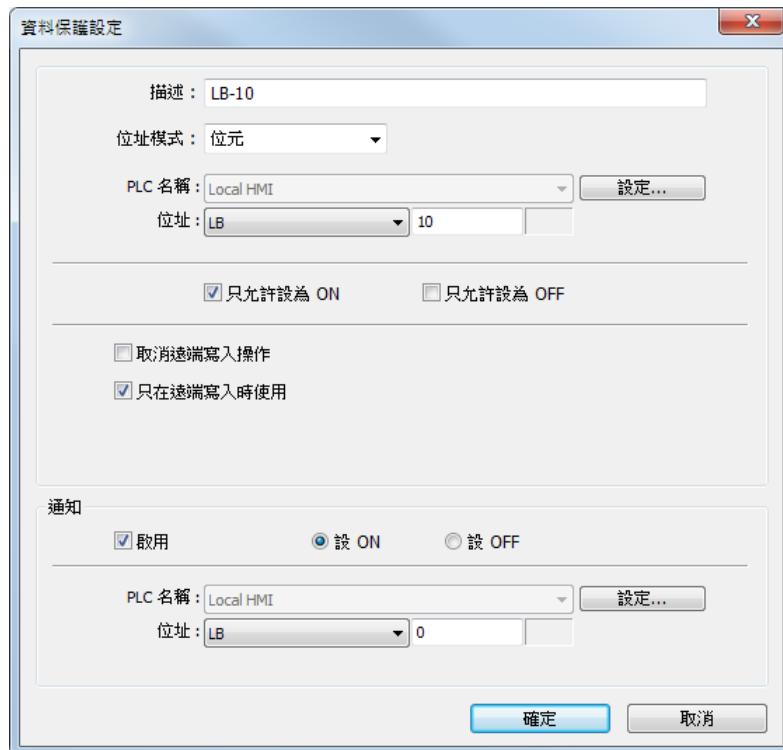


設定	描述
<b>描述</b>	輸入關於此項目的說明或備註。
<b>位址模式</b>	選擇 [字組] 模式後，可設定相關屬性。
<b>最小值</b>	設定此字組位址欲限制輸入的最小值。
<b>最大值</b>	設定此字組位址欲限制輸入的最大值。
<b>取消遠端寫入操作</b>	若勾選，遠端 HMI 將無法寫入此受保護的位址。
<b>只在遠端寫入時使用</b>	若勾選，將只有遠端寫入時的操作會有 [最小值] 與 [最大值] 的限制。
<b>若寫入的值小於最小值，則寫入最小值</b>	若勾選，當輸入值小於 [最小值] 的設定時，系統將寫入最小值。 若未勾選，當輸入值小於 [最小值] 的設定時，系統將保留原來的數值。
<b>若寫入的值大於最大值，則寫入最大值</b>	若勾選，當輸入值大於 [最大值] 的設定時，系統將寫入最大值。 若未勾選，當輸入值大於 [最大值] 的設定時，系統將保留原來的數值。
<b>通知</b>	當輸入的值小於 [最小值] 或大於 [最大值] 時，系統將觸發該通知位元位址。

以上圖為例，遠端 HMI 將無法寫入 LW-0，且當本地寫入的數值大於 10 時，則會寫入 10，同時觸發通知位元 LB-0 ON。

### 39.2.2. 位元位址屬性設定

設定與本地位元位址相關的操作限制。



設定	描述
<b>描述</b>	輸入關於此項目的說明或備註。
<b>位址模式</b>	選擇 [位元] 模式後，可設定相關屬性。
<b>只允許設為 ON</b>	可限制此位元位址只能被設為 ON。
<b>只允許設為 OFF</b>	可限制此位元位址只能被設為 OFF。
<b>取消遠端寫入操作</b>	若勾選，遠端 HMI 將無法寫入此受保護的位址。
<b>只在遠端寫入時使用</b>	若勾選，將只有遠端寫入時的操作會有 [只允許設為 ON] 與 [只允許設為 OFF] 的限制。
<b>通知</b>	若啟用，當勾選 [只允許設為 ON]，若欲將此受保護的位元位址設為 OFF，系統將觸發該通知位元位址。 相反的，當勾選 [只允許設為 OFF]，若欲將此受保護的位元位址設為 ON，系統將觸發該通知位元位址。

以上圖為例，遠端 HMI 將限制只能將 LB-10 設為 ON，而本地寫入操作則無限制。若遠端 HMI 欲將 LB-10 設為 OFF，系統將觸發通知位元 LB-0 ON。

# 40. 影像串流

本章節說明如何設定影像串流。

40.1.	概要	40-2
40.2.	設定	40-2
40.3.	操作方式	40-2
40.4.	支援機型	40-3

## 40.1. 概要

影像串流功能，可在第三方裝置上，透過網路瀏覽器串流顯示連接在 HMI 的 USB 攝影機的影像。

 請點選此圖示觀看教學影片，請先確定已連上網路線。

## 40.2. 設定

在 EasyBuilder Pro 中並無特定物件設定影像串流的功能。影像串流功能為透過系統暫存器控制。要使用影像串流功能前，請使用位元設定或巨集指令將以下相關系統暫存器加入工程檔案之中。

LB-

12356: 開啟 (設 ON) / 關閉 (設 OFF) 影像串流伺服器

LB-

12357: 影像串流伺服器狀態 (ON: 開啟 / OFF: 關閉)

LB-12356 設定為 ON 即開啟影像串流功能；LB12356 設定為 OFF 即關閉影像串流功能。

LB-12357 表示影像串流功能狀態：若為 ON 表示串流功能啟用中，若為 OFF 表示串流功能停用中。當 USB 攝影機未接上時，無法啟用影像串流功能。

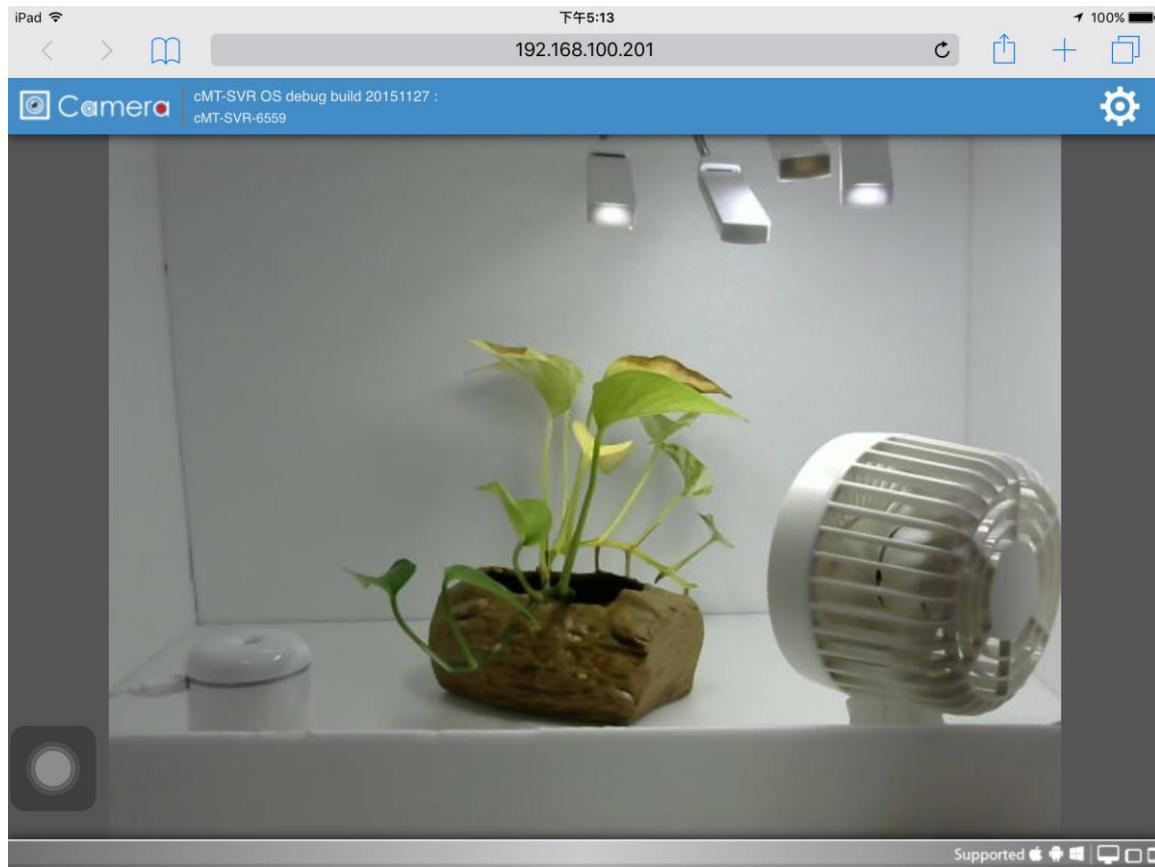
## 40.3. 操作方式

確認影像串流伺服器狀態為開啟後，要顯示串流影像，請在瀏覽器中輸入 HMI 的 IP 位址，並同時指定埠號 8080。如 IP 位址為 192.168.1.100，則位址欄輸入：

<http://192.168.1.100:8080>

### 範例 1

以下展示在 iPad 上使用 Safari 瀏覽器檢視植物監控的靜態影像。



### Note

- 如使用 EasyAccess 2.0，HMI 的 IP 位址即為軟體中所取得的虛擬 IP 位址。

#### 40.4. 支援機型

eMT 系列、mTV 系列、XE 系列、cMT 系列及 cMT-X 系列皆支援影像串流。

請注意功能是否支援與 OS 版本有關，請參考下列支援的 OS 版本，如未列表則為出廠即支援此功能。

機型	OS 版本
eMT 系列	20150525 或更新版本
XE 系列 /eMT3070B	20150225 或更新版本
mTV	20150508 或更新版本
cMT-SVR	20151117 或更新版本
cMT3090	20171024 或更新版本
cMT3151	20171129 或更新版本

# 41. 能源管理

本章節說明如何使用能源需量設定來監看與記錄用電量，並計算未來能源需量。

41.1. 能源需量設定 .....	41-2
41.2. 能源需量顯示設定 .....	41-6

## 41.1. 能源需量設定

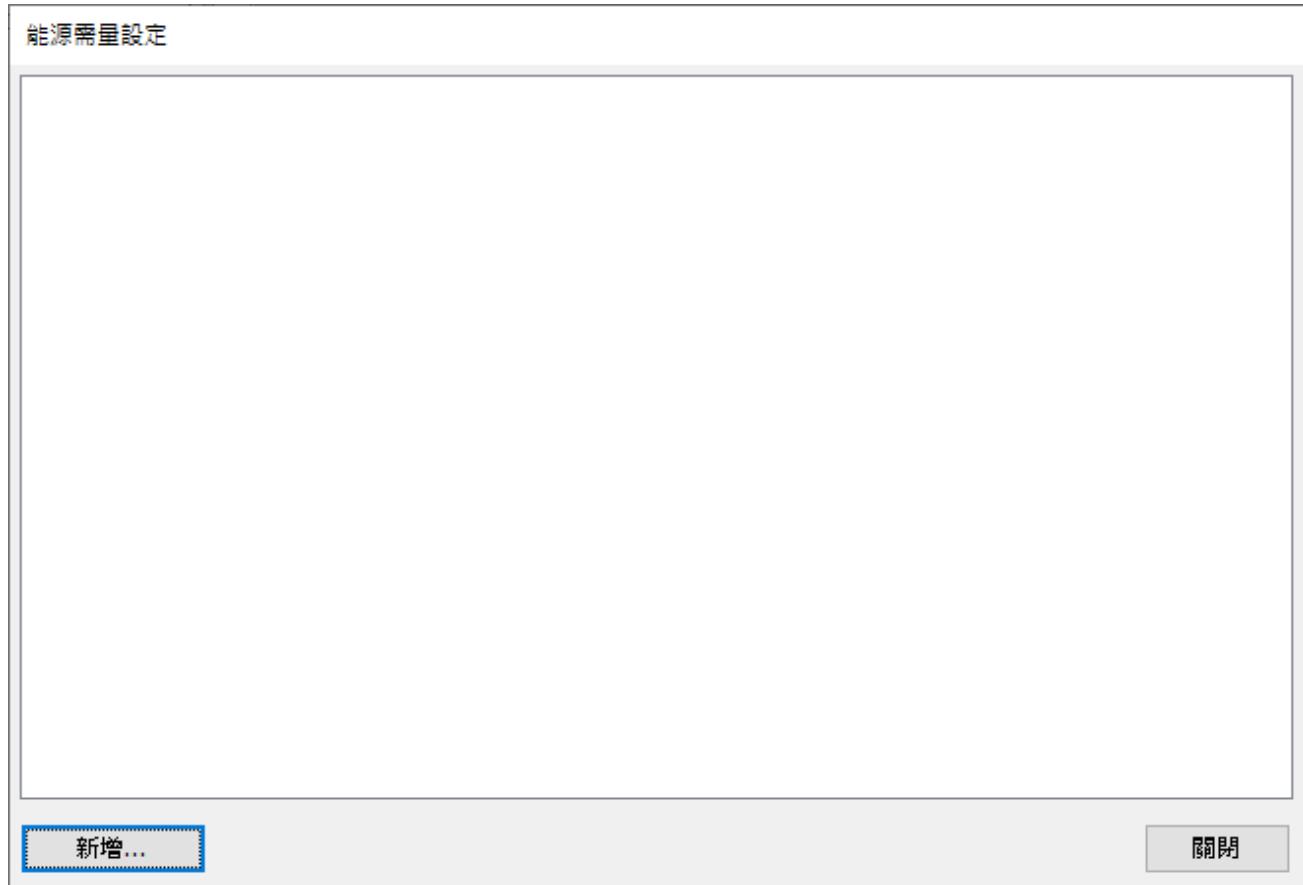
### 41.1.1. 概要

透過能源需量設定的功能，能夠利用需量週期 ( $T$ ) 與需量更新頻率 ( $t$ ) 的設定計算出需量，因此可以得知用電量的多寡以達到節約能源的目的。

### 41.1.2. 設定



點選功能表列 [資料/歷史]，並點選 [能源需量設定] 按鈕，即會出現 [能源需量設定] 物件視窗，選擇新增並正確設定一般屬性以及需量臨界值屬性後按下確定鍵，即完成一個 [能源需量設定] 物件。



## 一般屬性設定



設定	描述
累積用電量位址	用電量的記錄位址。單位可選擇 0.1/0.01/0.001 千瓦小時。
控制	在控制地址設定控制命令 1 時，將清除所記錄的需量輸出。
需量更新頻率(t)	設定記錄一次用電量的時間。範圍是 1~60 分鐘。
需量週期(T)	設定計算一次需量的週期。範圍是 1~60 分鐘。

### Note

- 需量週期(T)必須是需量更新頻率(t)的整數倍數。

## 需量臨界值設定



設定	描述
臨界值	可設定警示門檻與報警門檻數值，也可作動態設定。
通知	當需量數值超過警示臨界值或報警臨界值時，指定位址狀態會改變。
自動重置	若勾選，表示需量數值恢復時，指定位址狀態也會恢復成預設值。

## 需量輸出設定



設定	描述
啟用	開啟後可記錄當前需量值。
最大需量統計	可選擇是否要記錄當(前)日以及當(前)月的最大需量值。可選擇時間與日期格式以及計費週期的起始日。

## 41.2. 能源需量顯示設定

### 41.2.1. 概要

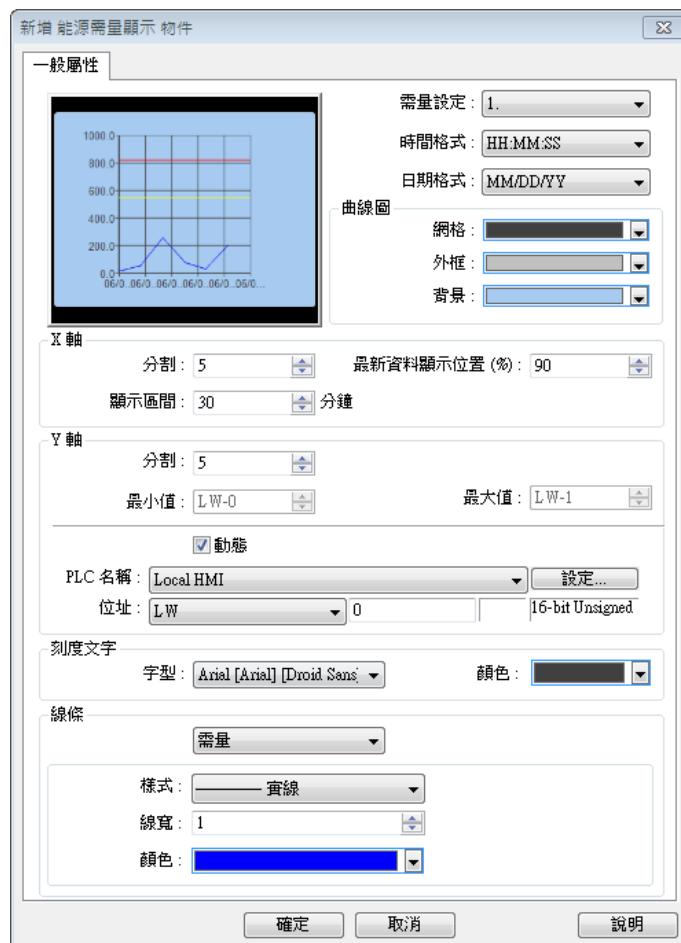
[能源需量顯示] 物件可在人機介面運作時，將特定能源需量的資訊用圖形的方式展示出來。可以自由調整字體、網格、線段的顏色與大小，警示與報警臨界線也可以在 [能源需量顯示] 物件上顯示。

### 41.2.2. 設定



點選功能表列 [資料/歷史]，並點選 [能源需量顯示] 按鈕後，即會出現 [能源需量顯示] 物件屬性對話窗，正確設定各項屬性後按下確定鍵，即可新增一個 [能源需量顯示] 物件。

#### 一般屬性設定



#### 設定

#### 描述

能源需量物件

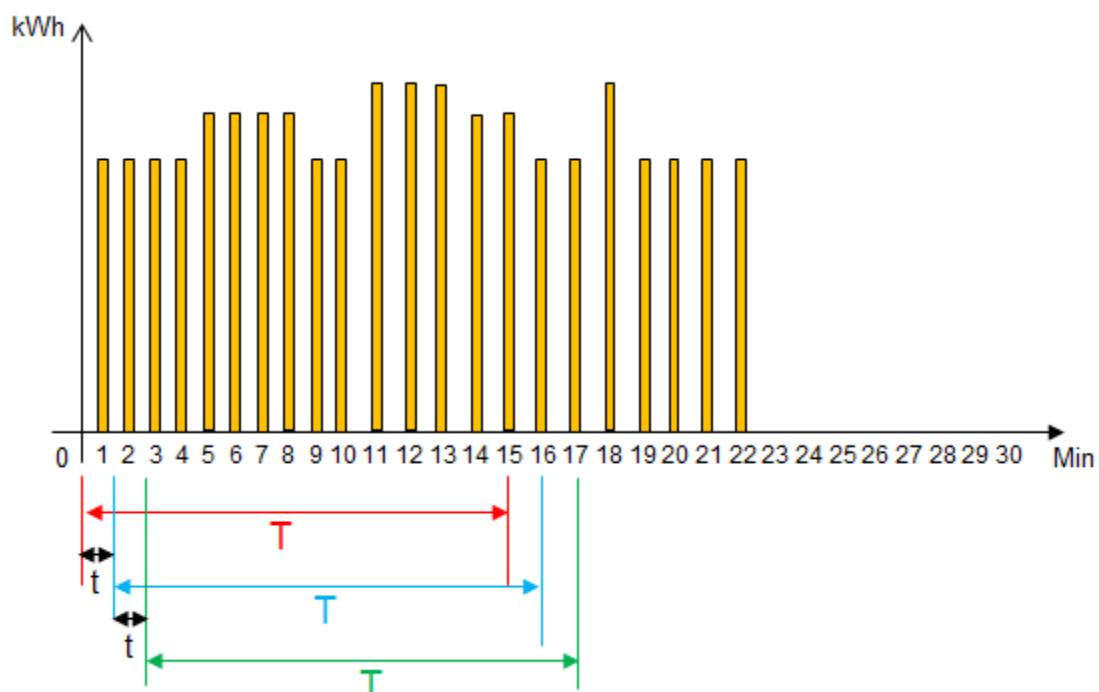
選擇欲檢視的能源需量資料。

**索引**

<b>時間/日期格式</b>	設定顯示的時間與日期格式。
<b>曲線圖</b>	設定網格、外框以及背景的顏色。
<b>X 軸</b>	設定分割數量、顯示區間以及最新資料的顯示位置 (50%~100%)。
<b>Y 軸</b>	設定分割數量以及上下限數值並且可動態設定。
<b>刻度文字</b>	可選擇文字字體以及顏色。
<b>線條</b>	需量、警報以及警告三條線段的樣式、寬度、顏色設定。

## 範例 1 能源需量算法

以下範例展示需量更新頻率( $t$ )和需量週期( $T$ )與需量的關係。



1. 若需量更新頻率( $t$ )為 1 分鐘，需量週期( $T$ )為 15 分鐘，如上圖所示。
2. 第 1~15 分鐘為紅色週期，將用電量全部加起來並乘上 4 (必須計算成 1 個小時)，就可得到一個需量值(kwh)。
3. 第 2~16 分鐘為藍色週期，將用電量全部加起來並乘上 4 (必須計算成 1 個小時)，就可得到一個需量值(kwh)。
4. 第 3~17 分鐘為綠色週期，將用電量全部加起來並乘上 4 (必須計算成 1 個小時)，就可得到一個需量值(kwh)。
5. 能源需量顯示物件會將獲得的需量值蒐集起來並作成圖表。
6. 如果  $t=3$ ， $T=15$ ，則是變成最近 5 次電量相加並乘上 4 (必須計算成 1 個小時)。
7. 如果  $t=5$ ， $T=30$ ，則是變成最近 6 次電量相加並乘上 2 (必須計算成 1 個小時)，以此類推。



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

# 42. IIoT

本章節說明如何使用 IIoT 的各種通訊協定。

42.1. MQTT .....	42-2
42.2. OPC UA 伺服器.....	42-26

## 42.1. MQTT

### 42.1.1. 概要

[MQTT] 物件可將訊息發送給 MQTT 伺服器，亦可從 MQTT 伺服器訂閱主題。HMI 也可做為 MQTT 伺服器，當 HMI 做為 MQTT 伺服器時，不會發送訊息至其他伺服器。

### 42.1.2. 設定



點選功能表列 [物件]，並點選 [IIoT] » [MQTT] 新增此物件。點選 [啟用] 後會跳出設定視窗。



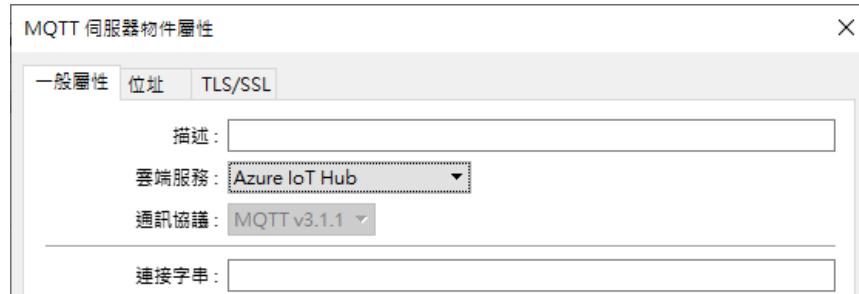
### 42.1.2.1. 飼服器設定

#### 一般屬性設定



設定	描述
雲端服務	<b>通用</b> 通用的 MQTT 發佈-訂閱主題模式。
	<b>AWS IoT</b> 以 AWS IoT 為 Broker，使用 Thing 傳遞資料並支援 Shadow 功能。詳細介紹請參考文件“AWS IoT 使用者手冊”。
	<b>Sparkplug B</b> Sparkplug B 是依據 IIoT 領域應用的使用特性，發展出標準 MQTT 未規範到的訊息主題和訊息內容，並能讓未支援 MQTT 的終端裝置也能夠透過 Edge of Network (HMI) 間接進行 MQTT 資料傳遞。詳細介紹請參考文件“Sparkplug B 快速入門手冊”。
	<b>Azure IoT Hub</b> 以 Microsoft Azure IoT Hub 為 Broker，簡化設定步驟，填入正確的連接字串即可通訊。連接字串可以在 Microsoft Azure IoT Hub 的裝置

細節內找到資訊。



<b>通訊協議</b>	支援 MQTT v3.1、v3.1.1、v5 (v5 僅支援 cMT/cMT X 系列)。
<b>自訂 Client ID/使用者名稱/密碼之長度</b>	Client ID：上限 128 字組。 使用者名稱/密碼：上限 256 字組。
<b>IP</b>	設定接收訊息的 MQTT 伺服器 IP。若輸入 IP 位址 127.0.0.1，將會啟動在人機上的 MQTT 伺服器。
<b>使用網域名稱</b>	支援使用網域名稱指定伺服器。
<b>連接埠號</b>	設定接收訊息的 MQTT 伺服器連接埠。
<b>Client ID</b>	登錄名稱。可使用變數做為登錄名稱。例如：輸入%0，則 Client ID 即為 HMI 的名稱。
<b>驗證</b>	選擇是否使用 [使用者名稱] 及 [密碼] 連接 MQTT 伺服器。
<b>使用者名稱</b>	連接 MQTT 伺服器使用的 [使用者名稱]。
<b>密碼</b>	連接 MQTT 伺服器使用的 [密碼]。
<b>測試連線時間</b>	當 MQTT 伺服器超過 [測試連線時間] 仍未收到 HMI 的測試訊息，會視為 HMI 已斷線。 備註：當使用模擬時，訊息傳遞可能會延遲，但延遲時間最長不會超過 [測試連線時間]。HMI 上的訊息則是會即時發送。
<b>時間戳記</b>	<b>本地時間</b> 使用 HMI 時間做為時間戳記。 <b>UTC 時間</b> 使用 UTC+0 時間。若時間戳記不正確，請檢查 [系統參數設定] » [時間同步/夏令時間] 頁籤的時區設定。
<b>正常斷線時清除訊息緩衝區</b>	選項啟用時，自行中斷連線時 (方式: 設定 MQTT 伺服器的命令控制位址為 2) 會清除緩衝區內的訊息。反之緩衝區內訊息會保留。預設為啟用。
<b>自動關閉非活動中的 MQTT 連線</b>	當閒置時間超過 [最大閒置時間] 時會自動斷線，直到下一次資料需要更新時才會再連線。 可選擇是否只有在第一次連線時才會更新初始數值以及主題列表。使用此項設定時，控制位址的開始/停止命令將無效。

## 位址設定



### 設定

#### 狀態位址

### 描述

LW-n: 顯示 [MQTT] 連線狀態

數值	描述
0	不啟用連線 MQTT 伺服器
1	已斷線或連線中
2	成功連線 MQTT 伺服器

LW-n+1: 錯誤提示

數值	描述
0	無錯誤
1	未知錯誤
2	連線失敗
3	伺服器拒絕存取
4	內建 MQTT 伺服器連接埠號不在允許範圍內
5	無法解析的伺服器名稱
6	緩衝區已滿

32	不正確的 Client ID
48	加密憑證驗證失敗
256	已在連線中

**緩衝區使用量地址**

發佈未成功的訊息會先儲存在記憶體內當作緩衝，最多為 10000 筆，地址數值顯示單位為%，無條件進位。

LW-n: 顯示緩衝區使用量

**控制位址**

LW-n: 控制 [MQTT 伺服器] 執行或停止

數值	描述
0	就緒
1	開始
2	停止
3	更新

LW-n+1: 設定 MQTT 伺服器的 IP 位址

LW-n+5: 設定 MQTT 伺服器的連接埠號

LW-n+6: 設定連線至 MQTT 伺服器的 Client ID

LW-n+26: 是否啟用驗證

數值	描述
0	停用
1	啟用

LW-n+27: 設定連線至 MQTT 伺服器的使用者名稱

LW-n+43: 設定連線至 MQTT 伺服器的密碼

LW-n+59: 設定連線至 MQTT 伺服器的網域名稱

當雲端服務選擇 Azure IoT Hub 時，控制地址如下

LW-n: 控制 [MQTT 伺服器] 執行或停止

數值	描述
0	就緒
1	開始
2	停止
3	更新

LW-n+1: 設定連接字串(128 words)

## TLS/SSL 設定



設定	描述
啟用	啟用 TLS/SSL 加密。可手動選擇加密版本 TLS 1.0, TLS 1.1 或 TLS 1.2。 使用 TLS 1.1 及 TLS 1.2 時，HMI 的 OS 必須使用 20180323 或更新的版本。
伺服器認證	<p><b>啟用</b></p> <p>驗證該伺服器憑證是否曾經經過憑證授權單位 (CA) 認證。伺服器憑證是在建立連線時由伺服器送出。</p> <p><b>伺服器名稱需與憑證資訊相符</b></p> <p>驗證該伺服器名稱或 IP 是否與憑證內的紀錄符合。Domain name 與 IP 紀錄是記載於憑證內的 Subject Alternative Name 中。</p>
客戶端驗證	私鑰與客戶端憑證為讓伺服器驗證客戶端所需的資料。

## 系統主題

目前有四種系統主題 HMI 可選擇是否發佈。若 HMI 有啟用系統主題，則訂閱者可透過訂閱該主題，得知該 HMI 的主題清單或連線狀態。



設定	描述
<b>Topic List</b>	HMI 內的主題清單。訊息在 HMI 與伺服器連線後發送至伺服器。
<b>Birth Topic</b>	HMI 與伺服器連線後，會發出的訊息。
<b>Close Topic</b>	HMI 主動與伺服器斷線前，最後發出的訊息。
<b>Last Will</b>	當 HMI 與伺服器之間發生異常斷線時，Last Will 的訂閱端會收到此訊息。HMI 在一開始建立連線時，就會將 Last Will 訊息同步設定至伺服器端。
<b>主題</b>	該系統主題的實際主題名稱。
<b>保留訊息</b>	MQTT 伺服器會保留最新的一筆資料。
<b>QoS</b>	MQTT 提供三個級別的可靠性，稱為服務質量。訊息傳送的服務質量決定了訊息是否保證送達。 QoS 0: 訊息只發送一次，不保證送達 QoS 1: 訊息送達至少要一次 QoS 2: 訊息送達剛好一次
<b>內容格式</b>	JSON (Default) : 使用預設值

各系統主題的預設值:

紅字表示為當下實際值

Topic list:

```
{
    "d": {
        "topics": [

```

```
{  
    "compression" : "壓縮類型",  
    "nickname" : "主題名稱",  
    "topic" : "主題"  
},  
{  
    "compression" : "壓縮類型",  
    "nickname" : "主題名稱",  
    "topic" : "主題"  
}  
]  
,  
"ts" : "當前時間"  
}
```

topics 內的消息依據實際主題設定有所不同。上為兩個主題的例子。

#### Birth Topic:

```
{  
    "d":{  
        "connected":true  
    },  
    "ts":"當前時間"  
}
```

#### Close Topic:

```
{  
    "d":{  
        "connected":false  
    },  
    "ts":"當前時間"  
}
```

#### Last Will:

```
{  
    "d":{  
        "connected":false  
    }  
}
```

}

JSON (Customized)：使用自訂內容。

### Note

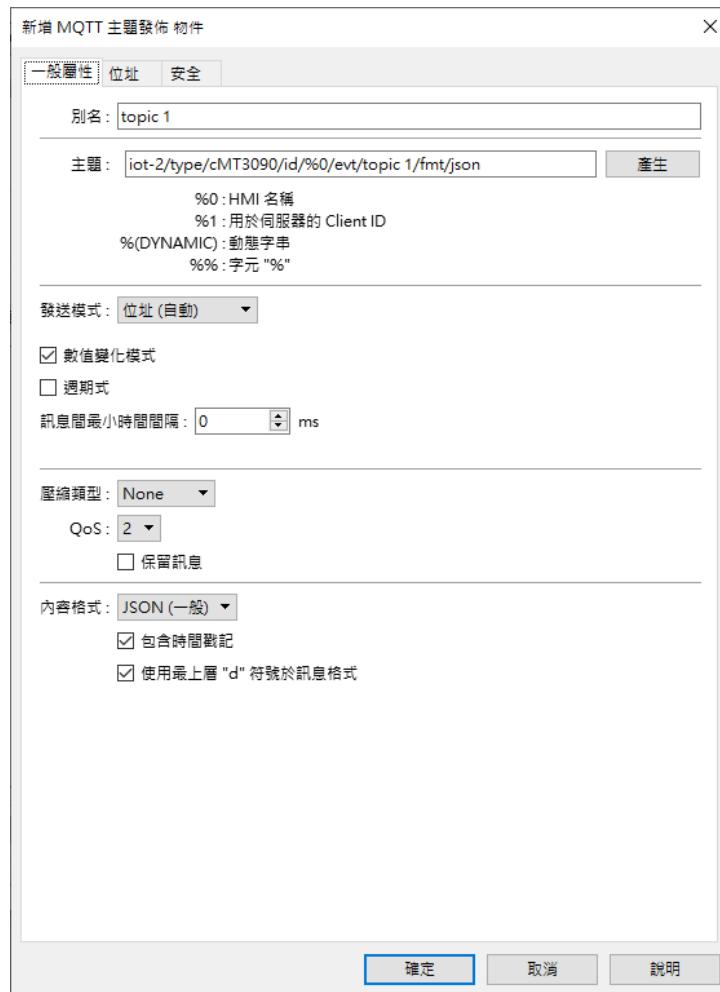
- 雲端服務選擇 Sparkplug B,Azure IoT Hub 時不支援【系統主題】分頁功能。
- ▶ 請點選此圖示觀看教學影片，請先確定已連上網路線。

#### 42.1.2.2. MQTT 主題發佈



選擇【新增】可進入一般屬性設定與位址設定，或是可直接使用【匯出】/[匯入] CSV 檔案功能來建立 MQTT 發佈主題。MQTT 發佈主題的數量上限為 255 件。

## 一般屬性設定



設定	描述
別名	設定 MQTT 主題的項目名稱。
主題	發送訊息時，MQTT 伺服器收到的主題。可使用變數做為主題內容。 %(DYNAMIC) 用法：在主題欄位輸入 %(DYNAMIC)，則下方會產生一個可設定動態地址的選項。%(DYNAMIC) 的字串內容亦可包含多個 topic level。例如：myhome/groundfloor。



當雲端服務選擇 Azure IoT Hub 時，主題發佈有固定格式，使用者只能自訂最後一個 level。



## 發送模式

### 位址(自動)

數值變化模式：

當任一數值有變化時，即發佈 MQTT 訊息。

週期式：

定時發佈 MQTT 訊息。

訊息間最短時間間隔：

可以設定訊息間最短時間間隔避免過度發布 MQTT 訊息。也就是當數值變化的間隔小於訊息間最短時間間隔時，則會將訊息放在緩存，待滿足訊息間最短時間間隔時才會再發布 MQTT 訊息。

### 位址(位元觸發)

指定位元被觸發時，即發佈 MQTT 訊息。

### 事件登錄

發佈來源可使用事件登錄的資料。可選擇當單一的事件登錄項目或當類別下的任一項目觸發時發佈 MQTT 訊息。

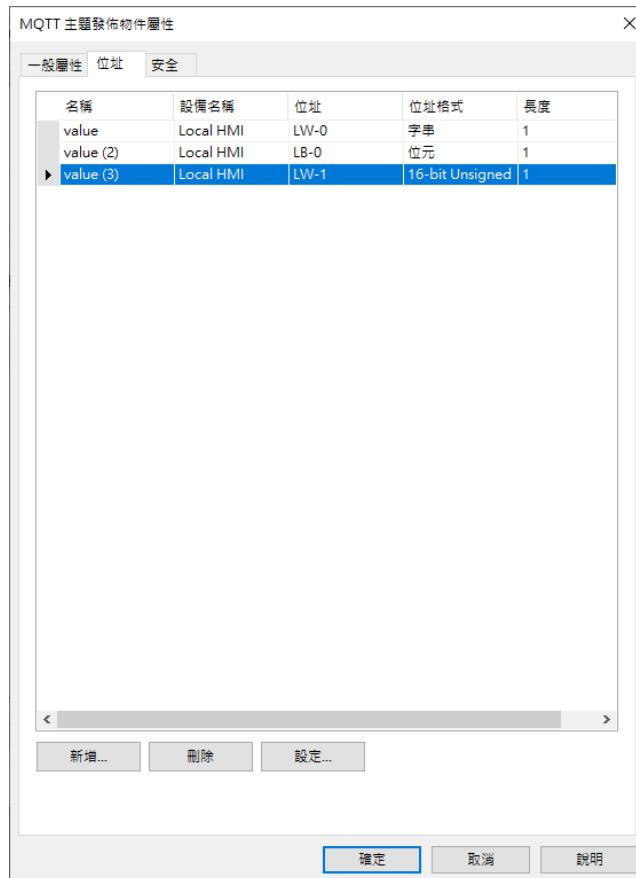
## 壓縮類型

傳輸訊息前，會先將訊息壓縮。被壓縮的訊息，在 MQTT 客戶端要讀取前，需先解壓縮。可選擇使用 zlib, gzip 或是 DEFLATE 演算法來壓縮或解壓縮訊息。

保留訊息	勾選後，MQTT 伺服器會保留最新的一筆資料。
包含時間戳記	當內容格式使用 [JSON (一般)] 時才支援此功能，可手動決定是否要包含時間戳記。
使用最上層 "d" 號於訊息格式	當內容格式使用 [JSON (一般)] 時才支援此功能。 勾選後，訊息格式如下：
	<pre>{   "d": {     "addressName1": ...,     "addressName2": ...   },   "ts": ... }</pre>
	未勾選，訊息格式如下：
	<pre>{   "addressName1": ...,   "addressName2": ...,   "ts": ... }</pre>
	如圖所示，未勾選的情況下，ts 和地址名稱都是同層級的鍵值(key)。因此，應避免將地址名稱取為 ts。
QoS	MQTT 提供三個級別的可靠性，稱為服務質量。訊息傳送的服務質量決定了訊息是否保證送達。 0: 訊息只發送一次，不保證送達 1: 訊息送達至少要一次 2: 訊息送達剛好一次
內容格式	Raw data：以 BYTE 數據組成的資料。 JSON (一般)：將所有資料放在成員 "d" 的 JSON 格式。 JSON (進階)：可彈性自訂巢狀結構的 JSON 格式。

## 位址設定

此章節說明當內容格式使用 [Raw Data] 與 [JSON (一般)] 時位址的設定。

**設定****描述****新增**

建立主題的位址來源。每一個位址可分別設定長度。

**刪除**

刪除位址。

**設定**

修改位址及名稱。

**類型/位址**

名稱：

類型  位元  字組

位址  
設備：     
位址：

\* 您可以使用標籤庫中的轉換標籤來轉換數據。

包括在發送的所有訊息中  
 數用小數點後位數之個數

**類型/位址**

名稱：

類型  位元  字組

位址  
設備：     
位址：

\* 您可以使用標籤庫中的轉換標籤來轉換數據。

包括在發送的所有訊息中  
 移除 JSON 陣列括號 '[' 和 ']'  
 自動跳脫特殊字元

**確定**    **取消**

**設定****描述****包括在發送的所有訊息中**

當該發送的其他位址數值有變化時，可以選擇此資料是否要包含在發送的訊息中。此功能僅支援 [JSON (一般)] 與 [JSON (進階)] 內容格式。

**移除 JSON 陣列括**

在使用非陣列位元或字元時可手動移除陣列符號。此

號 'T' 和 'Y'	功能僅支援 [JSON (一般)] 內容格式。
啟用小數點後位數之 個數	當位址格式使用 Float 浮點數時，可以選擇小數點後 的位數。此功能僅支援 [JSON (一般)] 與 [JSON (進 階)] 內容格式。
自動跳脫特殊字元	cMT/cMT X 系列支援當位址格式為字串時，可以選擇 自動跳脫特殊字元。  MQTT 字串可能含有 JSON 的關鍵字(像是 " 和 \ ) 導致 JSON 解析(JSON parsing)時失敗。啟用此功能 後，發送的字串內容如有關鍵字，將會自動跳脫，也 就是當有 " 符號時則發送 \"，當有 \ 符號時則發 送 \\，讓 MQTT 字串可以被正常解析。此功能僅支 援 [JSON (一般)] 內容格式。

### Note

- 一個 tag 長度最多可使用 255 個 word。

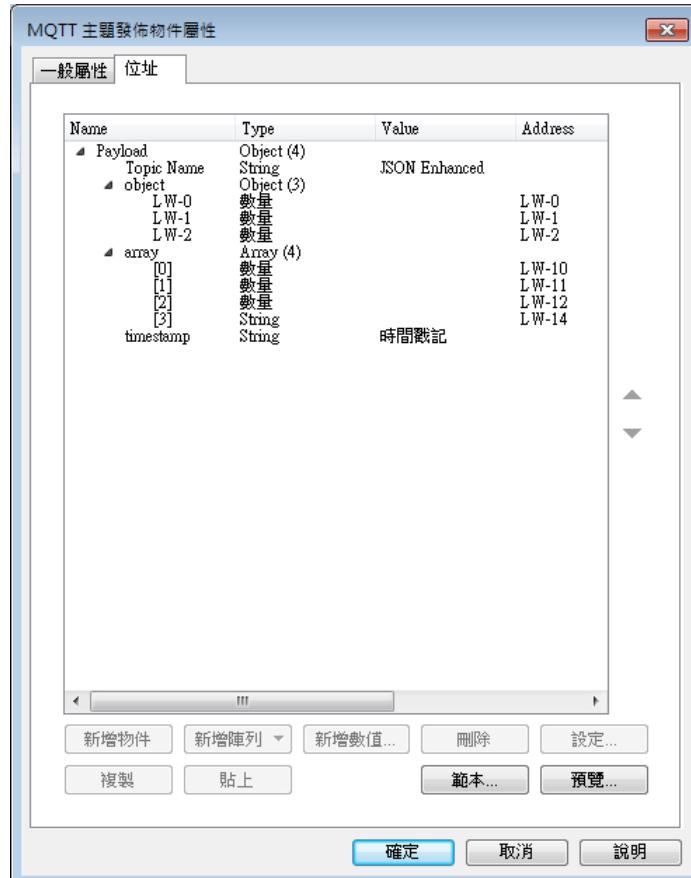
## 安全設定



當暫存器狀態符合設定時，才會發佈該主題。如上圖，當 LB-0 為 ON 時，此主題才會發布。

## 位址設定 [JSON (進階)]

此章節說明當內容格式使用 [JSON (進階)] 時位址的設定。[JSON (進階)] 支援巢狀結構，可使用物件、陣列等形式，時間戳記及資料名稱亦可自訂，具有較彈性的設計方式。



以上圖為例，若使用上圖的設定，在訂閱端將會收到以下形式的 MQTT 訊息。

```
{
  "Topic Name" : "JSON Enhanced",
  "Object" : {
    "LW-0" : [ 1 ],
    "LW-1" : [ 2 ],
    "LW-2" : [ 3 ]
  },
  "Array" : [ [ 4 ], [ 5 ], [ 6 ], [ "AABBCCDD" ] ],
  "timestamp" : "2019-02-19T06:52:13.846038"
}
```

設定	描述
新增物件	新增一個資料物件。物件中可有多個資料格式，每個資料格式各自有名稱及數值。物件的資料使用大括號 {} 包括。
新增陣列	新增一個資料陣列。陣列中可有多個資料格式，但只會有一個名稱。物件的資料使用中括號 [] 包括。
新增數值	新增一個數值、字串或時間戳記。當為數值或字串

時，可為固定數值或從指定位址讀取數據。

**刪除** 刪除選定的欄位。

**設定** 修改選定的欄位。若選擇的欄位為物件及陣列時，僅可修改名稱。但物件及陣列包括的數值可修改參數。

**複製** 複製選定的欄位。

**貼上** 在選擇的欄位貼上之前複製的部分。

**範本** 在此貼上 JSON 文字，系統會自動編排成符合此 JSON 格式的架構，省去自行定義的時間。



**預覽** 用容易閱讀的格式預覽 JSON 資料。

### Note

- 一個 Topic 最多可使用 512 個節點 (包含 payload)，一個 tag 長度最多可使用 255 個 word。

#### 42.1.2.3. MQTT 主題訂閱



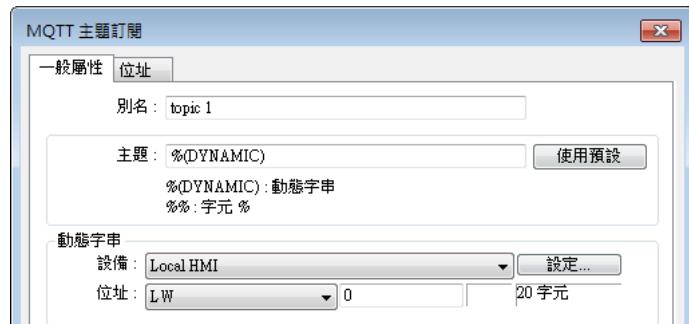
選擇 [新增] 可進入一般屬性設定與位址設定，或是可直接使用 [匯出]/[匯入] CSV 檔案功能來建立 MQTT 訂閱主題。MQTT 訂閱主題的數量上限為 255 件。

## 一般屬性設定

此章節說明當內容格式使用 [Raw Data] 與 [JSON (一般)] 時位址的設定。



設定	描述
別名	設定 MQTT 主題的項目名稱。
主題	從 MQTT 伺服器訂閱的主題。可使用動態字串訂閱。 %(DYNAMIC) 用法：在主題欄位輸入 %(DYNAMIC)，則下方會產生一個可設定動態地址的選項。 %(DYNAMIC) 的字串內容亦可包含多個 topic level。例如：myhome/groundfloor。



當雲端服務選擇 Azure IoT Hub 時，主題訂閱有固定格式，使用者只能自訂最後一個 level，必須與主題發佈對應。



#### 壓縮類型

訂閱主題的傳輸壓縮必須與發佈主題相同。

#### 驗證時間戳記

使用者可自行決定是否要驗證時間戳記。勾選表示收到的資料時間戳記必須嚴格遞增才會更新，否則會被判斷成為過時的資料而不更新。

#### 使用最上層“d”符號於 訊息格式

勾選後，訊息格式如下：

```
{
  "d": {
    "addressName1": ...,
    "addressName2": ...
  },
  "ts": ...
}
```

未勾選，訊息格式如下：

```
{
  "addressName1": ...,
  "addressName2": ...,
  "ts": ...
}
```

請依照資料來源選擇適當的設定。

#### QoS

MQTT 提供三個級別的可靠性，稱為服務質量。訊息傳送的服務質量決定了訊息是否保證送達。

0: 訊息只發送一次，不保證送達

1: 訊息送達至少要一次

2: 訊息送達剛好一次

#### 內容格式

Raw data：未有特定格式的原始數據。

JSON (一般)：單層結構的 JSON 格式。

JSON (進階)：可彈性自訂巢狀結構的 JSON 格式。

#### 操作模式

主題訂閱可以設定模式

自動：收到資料立即將數值寫至地址。

手動：收到訊息後，先將數值暫存至緩衝區，等到控制地址收到執行指令，才將數值寫至地址。緩衝區儲存上限為一百筆。

#### 控制地址 (手動模式)

LW- n: 命令

數值	描述
1	將緩衝區裡最舊一筆的資料寫入地址。緩衝區裡若有十筆資料，可下十次命令 1 依序寫入地址。
2	將緩衝區裡最新的一筆資料寫入地址，並清空緩衝區裡所有資料。

LW-n+1: 執行結果

數值	描述
0	目前暫存器中沒有資料
1	執行結果成功
2	此主題訂閱目前處於關閉狀態，執行失敗(見 MQTT 主題訂閱 安全設定)

LW- n+2: 未處理的訊息

緩衝區中的資料數量。

## 位址設定



### 設定

### 描述

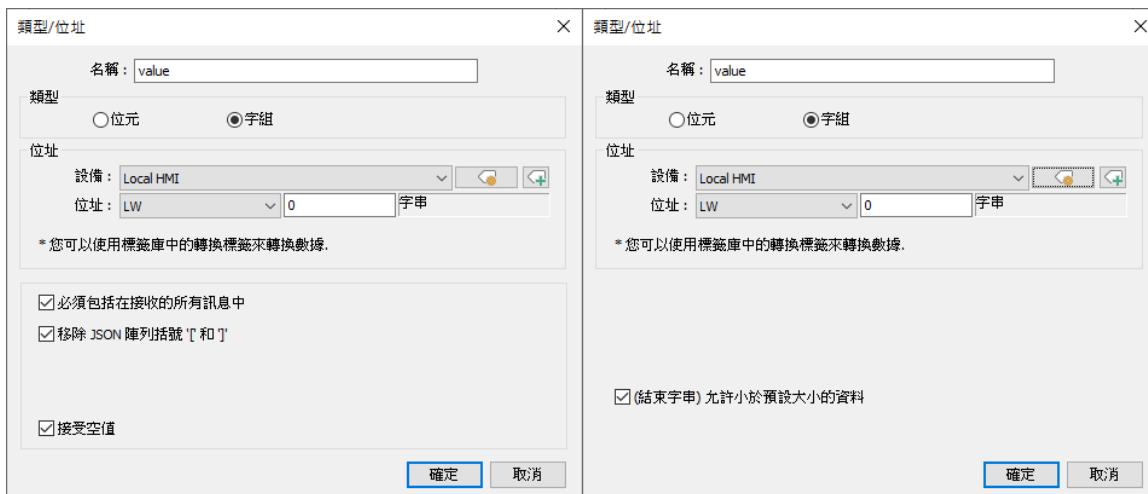
**新增** 建立訂閱主題後資料放置的位址。每一個位址可分別設定長度。

**刪除**

刪除位址。

**設定**

修改位址及名稱。

**設定****描述****必須包括在接收的所有訊息中**

當該訂閱的其他位址數值有變化時，可以選擇此資料是否要包含在接收的訊息中。此功能僅支援 [JSON (一般)] 與 [JSON (進階)] 內容格式。

**移除 JSON 陣列括號 '[' 和 ']'**

在使用非陣列位元或字元的時候可手動移除陣列符號。此功能僅支援 [JSON (一般)] 內容格式。

**接受空值**

可以接收空值(NULL)。此功能僅支援 [JSON (一般)] 與 [JSON (進階)] 內容格式。

**(結束字串)允許小於預設大小的資料**

當接收的字串長度少於此資料的字串設定長度時，仍可接收。目前限定只有該訂閱物件的最後一個字串資料可進行此設定，若字串資料後面還有使用其他數值或位元資料，則無法使用此功能。此功能僅支援 [Raw data] 內容格式。

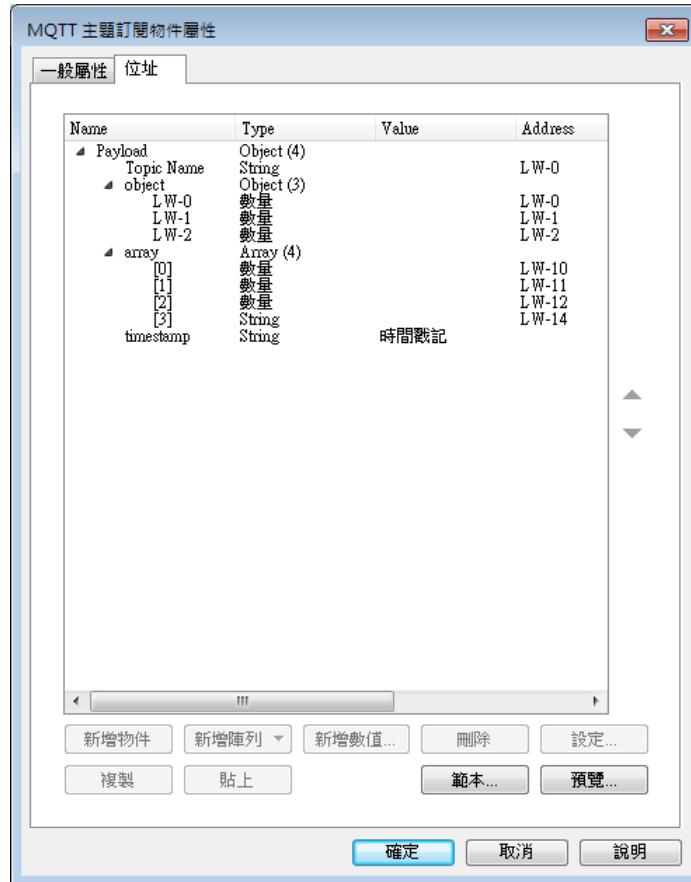
## 安全設定



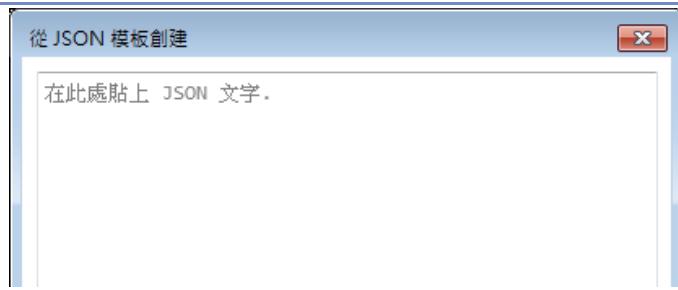
當暫存器狀態符合設定時，才會訂閱該主題。如上圖，當 LB-0 為 ON 時，才會訂閱該主題。

### 位址設定 [JSON (進階)]

此章節說明當內容格式使用 [JSON (進階)] 時位址的設定。[JSON (進階)] 支援巢狀結構，可使用物件、陣列等形式，時間戳記及資料名稱亦可自訂，具有較彈性的設計方式。



設定	描述
新增物件	新增一個資料物件。物件中可有多個資料格式，每個資料格式各自有名稱及數值。物件的資料使用大括號 {} 包括。
新增陣列	新增一個資料陣列。陣列中可有多個資料格式，但只會有一個名稱。物件的資料使用中括號 [] 包括。
新增數值	新增一個數值、字串或時間戳記。當為數值或字串時，可為固定數值或從指定位址讀取數據。
刪除	刪除選定的欄位。
設定	修改選定的欄位。若選擇的欄位為物件及陣列時，僅可修改名稱。但物件及陣列包括的數值可修改參數。
複製	複製選定的欄位。
貼上	在選擇的欄位貼上之前複製的部分。
範本	在此貼上 JSON 文字，系統會自動編排成符合此 JSON 格式的架構，省去自行定義的時間成本。



### Note

- Amazon Web Service(AWS) IoT Core 的可支援標準 MQTT 協議。使用上請注意：
  1. 主題最多僅能使用八層，例如 iot-2/type 為兩層。
  2. 不支援 [一般屬性] 的驗證功能，需使用 [TLS/SSL] 的憑證方式驗證。
  3. 僅支援 QoS 0 與 QoS 1。
  4. 不支援主題發佈的 [保留訊息] 功能。

#### 42.1.2.4. Sparkplug B

當雲端服務設定為 Sparkplug B 時，一般屬性及設備設定介紹如下。

#### 一般屬性設定



設定	描述
群組 ID	識別此 Edge Of Network Nodes 所屬的群組 ID。
Edge node ID	識別此 Edge Of Network Node 的 ID。
DDATA 最小時間	若偵測到資料變化，發送新的 DDATA (Device DATA) 訊息前的最少等待時間。
QoS	MQTT 提供三個級別的可靠性，稱為服務質量。訊息

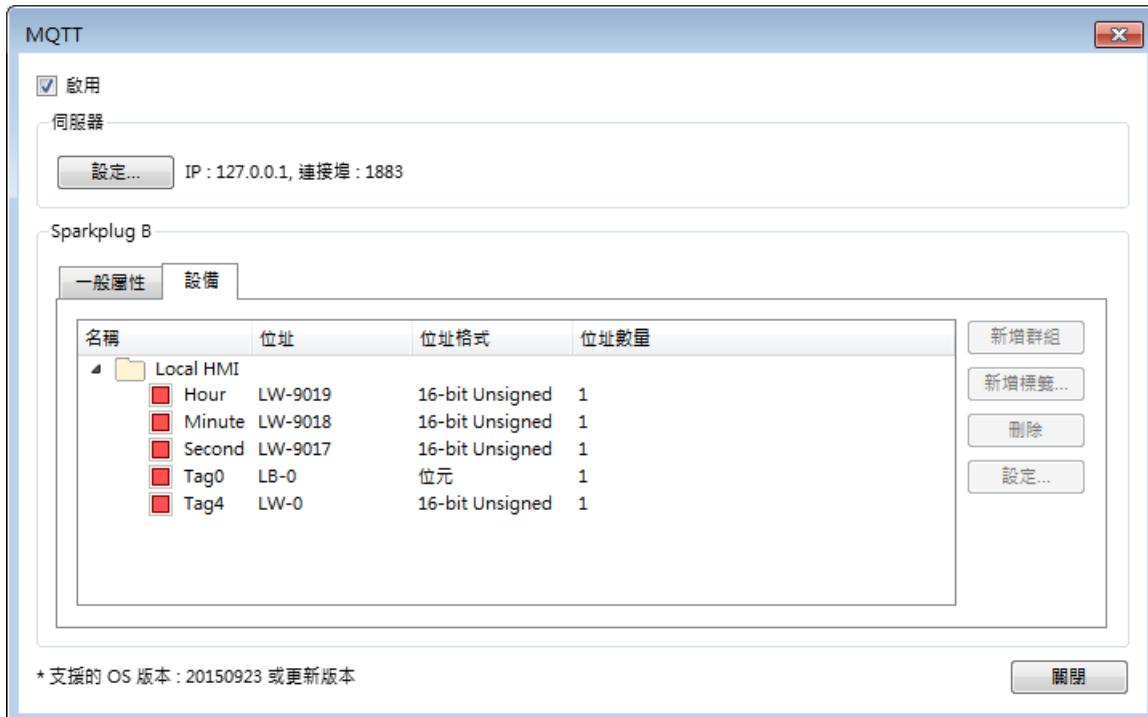
傳送的服務質量決定了訊息是否保證送達。

0: 訊息只發送一次，不保證送達

1: 訊息送達至少要一次

2: 訊息送達剛好一次

## 設備設定



\* 支援的 OS 版本 : 20150923 或更新版本

關閉

設定	描述
新增群組	新增群組來管理標籤。
新增標籤	新增此 EoN Node 在 MQTT Engine 監控的標籤。名稱不可為空白。
刪除	刪除已存在的群組或標籤。
設定	設定已存在的群組或標籤



請點選此圖示下載範例程式。下載範例程式前，請先確定已連上網路線。

## 42.2. OPC UA 伺服器

### 42.2.1. 概要

OPC UA(Unified Architecture)是在工業自動化產業的通訊標準。具有資料通訊不受限於平台、統一存取機制、通訊的標準化以及安全憑證機制的特性。cMT / cMT X 系列人機支援 OPC UA 伺服器的角色，可以利用 OPC UA 客戶端 (Client) 軟體存取人機或 PLC 上的地址標籤資訊，進一步達到垂直整合的成果。

軟硬體需求：

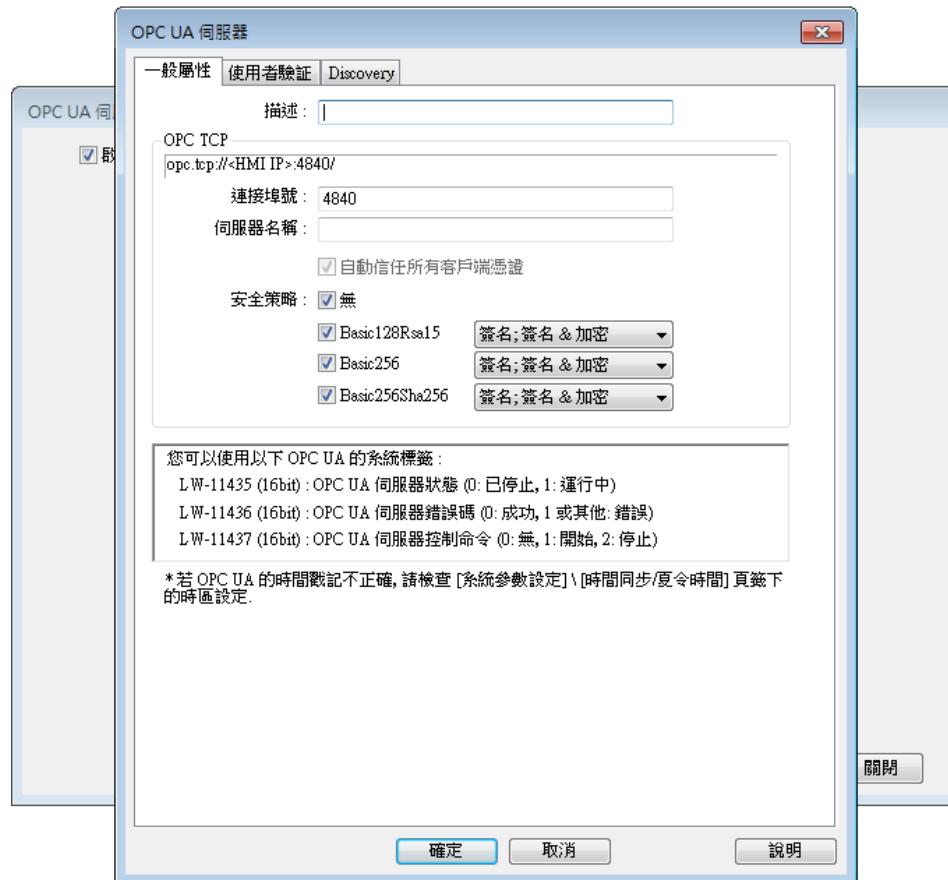
- 支援機型：cMT / cMT X 系列。\* cMT-SVR / cMT-SVR-200 和 cMT-HDM / cMT-FHD / cMT-FHDX 需額外載入授權。
- 支援軟體：Easy Builder Pro v5.06.01 or later version
- 建議 OPC UA 客戶端程式：Unified Automation UaExpert

### 42.2.2. 設定

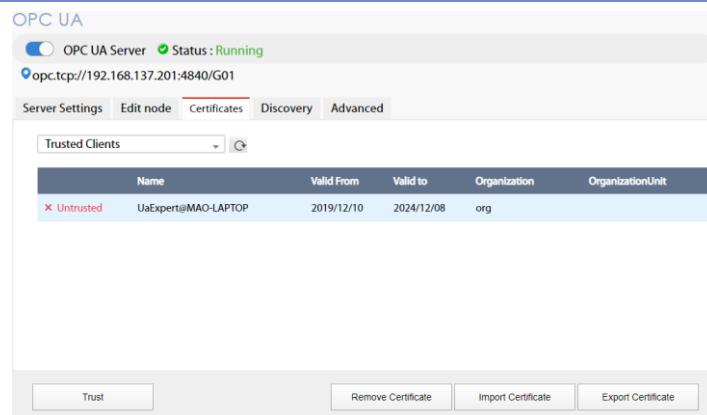


點選功能表列 [物件]，並點擊 [IIoT] » [OPC UA 伺服器] 新增此物件。點選啟用後會跳出設定視窗。

### 42.2.2.1. 一般屬性設定



設定	描述
描述	對於此物件的描述。
OPC TCP	伺服器的 URL 地址。
連接埠號	設定客戶端連進來需要的連接埠號，預設為 4840。
伺服器名稱	填寫伺服器的名稱，可空白。 <b>自動信任所有客戶端憑證</b> 此選項預設為啟用，僅 cMT Gateway 系列可選擇不啟用。未啟用此選項時，所有 OPC UA 客戶端皆會被拒絕連線，除非該客戶端的憑證已於 OPC UA 網頁介面中被設定為信任，如下所示：

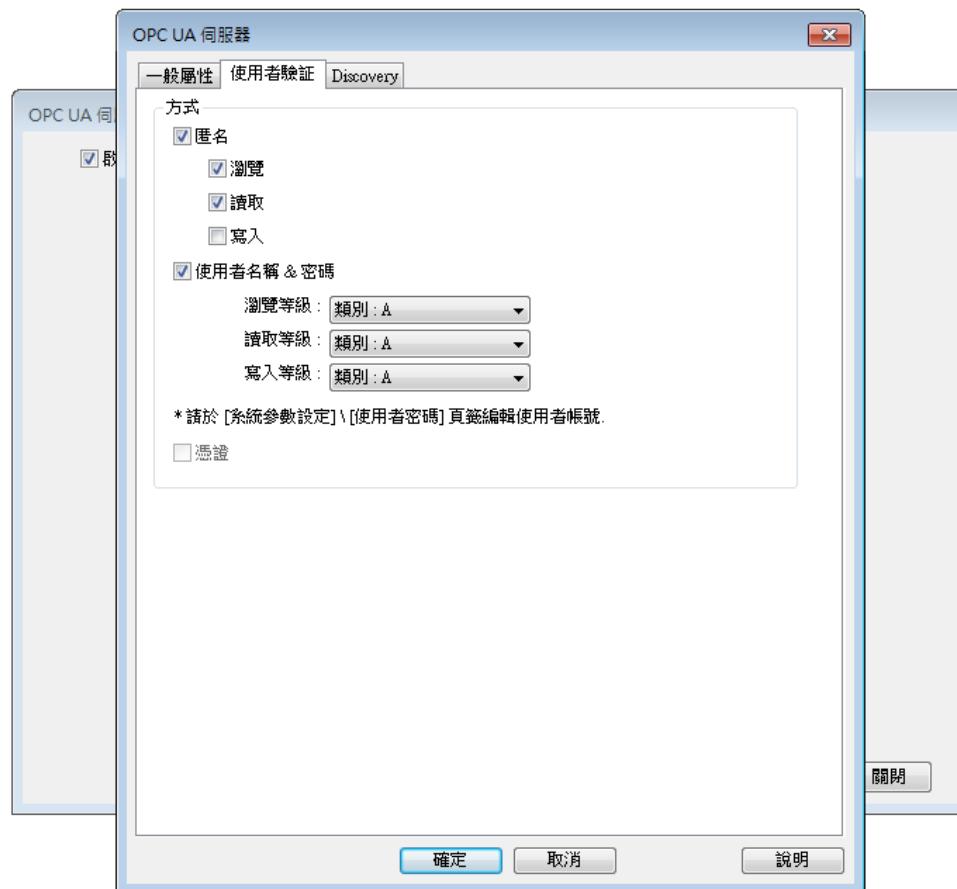


請注意，根據 OPC UA 規範，除非安全策略允許 [無] 的選項，OPC UA 客戶端必須使用客戶端憑證進行連線，且 OPC UA 伺服器會檢查憑證的正當性。

#### 安全策略

此為 OPC UA 提供的安全策略，和客戶端可選用的加密演算方式。

#### 42.2.2.2. 使用者驗證設定



#### 設定

方式

#### 描述

匿名

當客戶端軟體使用匿名登入時，資料存取的權限以瀏

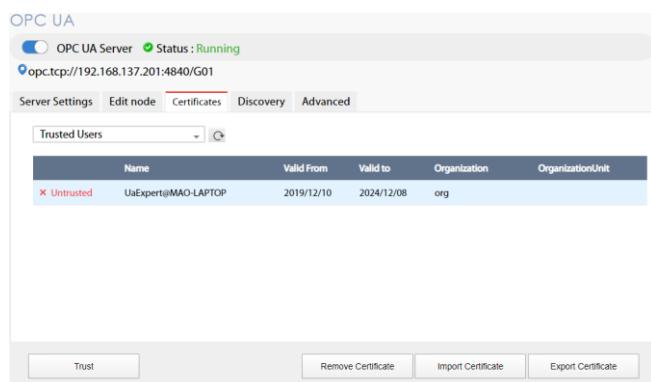
覽/讀取/寫入設定。

### 使用者名稱&密碼

與人機的使用者名稱&密碼共用。在客戶端軟體登入後，資料存取的權限則是以級別區分。

### 憑證

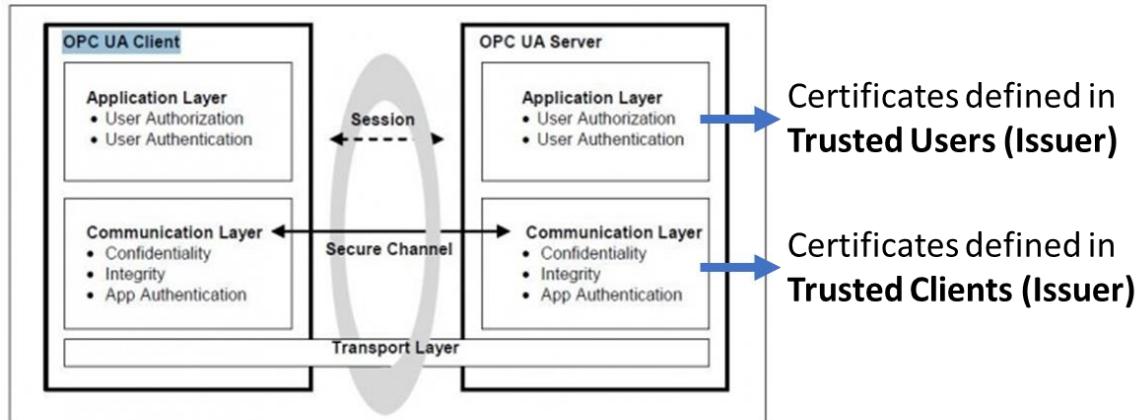
此選項僅支援於 cMT Gateway 系列。OPC UA 客戶端可使用憑證作為驗證方式，而不用使用者名稱與密碼登入。使用網路介面設定受信任/不受信任的使用者憑證，如下所示：



### Note

- OPC UA 的安全層級可分為：

- (1) Communication Layer (通訊層級，例如：安全策略)
- (2) Application Layer (應用層級)，如下所示：



安全層級 (請見連結說明：<http://wiki.opcfoundation.org/index.php/File:SecurityLayers.jpg>)

- 客戶端憑證位於通訊層級，使用【無】以外的安全策略時，憑證是必須的。
- 使用者憑證位於應用層級。使用使用者憑證是一種登入的方式。

### 42.2.2.3. Discovery 設定

開啟後，OPC UA 伺服器將連接並註冊至指定的 OPC UA Local Discovery Server (LDS)。

OPC UA Discovery 服務用於簡化大量 OPC UA 伺服器網路位置的維護工作；OPC UA 客戶端連接至 OPC UA Discovery 伺服器後，一次查詢區域網路中所有的 OPC UA 伺服器。



設定	描述
IP	OPC UA Local Discovery Server (LDS) 的 IP 地址。
連接埠號	OPC UA LDS 的連接埠號。
伺服器名稱	OPC UA LDS 的伺服器名稱。
描述	做為備註使用，不影響通訊。

### 範例

以下步驟說明如何設置 Discovery 功能：

- 先安裝 OPC UA Local Discovery Sever (LDS) (例如安裝至名稱為 DESKTOP-abcd 的電腦)。OPC UA LDS 可至 OPC UA 基金會下載：

<https://opcfoundation.org/developer-tools/developer-kits-unified-architecture/local-discovery-server-lds/>

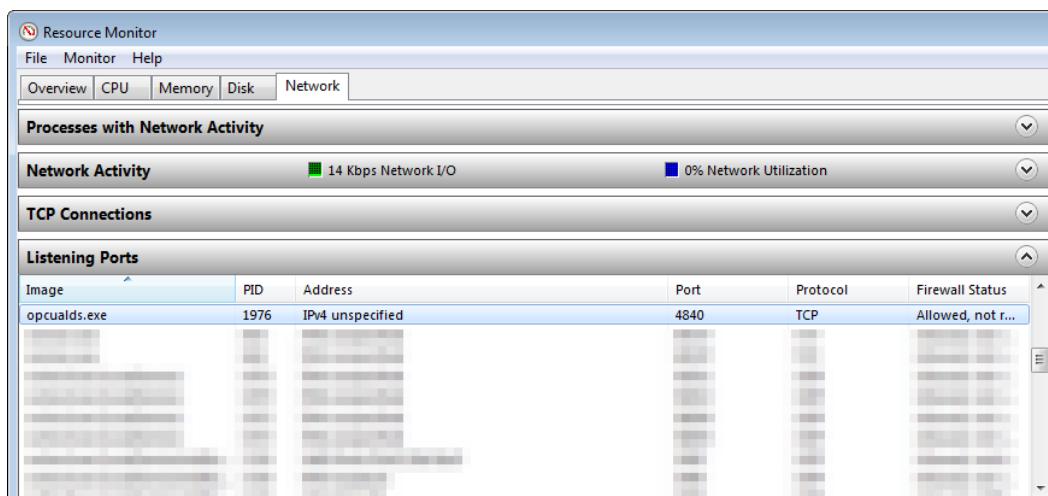
- 若 HMI 設定為自動取得 IP，則所處的 DHCP 伺服器與 DNS 伺服器需支援對應的功能，才能將

HMI 名稱解析至 IP 地址。若 HMI 為固定 IP，可將 HMI 的名稱改為 HMI 的 IP 位址。例如：HMI 的 IP 為 192.168.1.100，則 HMI 使用的名稱須使用 192.168.1.100 或可使用 0.0.0.0。

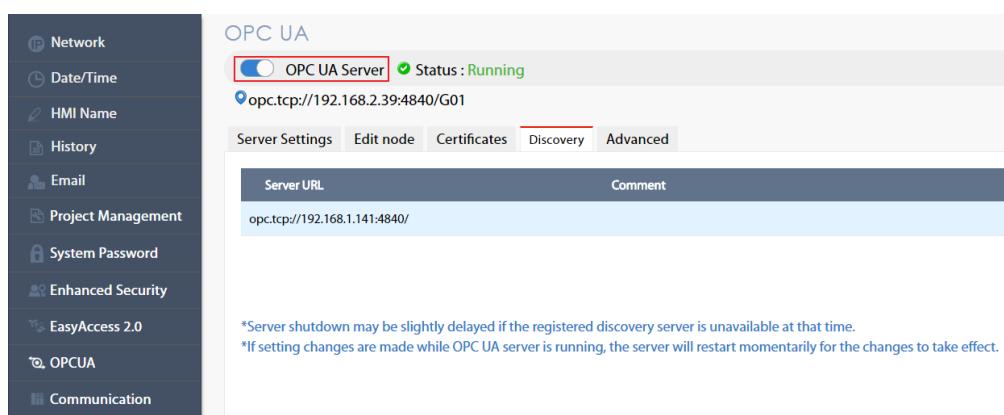
3. 安裝 LDS 的電腦中，將 OPC UA 伺服器的憑證由目錄 C:\ProgramData\OPC Foundation\UA\pki\rejected\certs (存放不信任的憑證) 複製到目錄 C:\ProgramData\OPC Foundation\UA\pki\trusted\certs (存放信任的憑證)。
4. 啟動 OPC UA 客戶端軟體，輸入第一步的電腦名稱 (例如 DESKTOP-abcd) 或 IP 位置，便可快速查閱已註冊的 OPC UA 伺服器。

如 Discovery 功能無法正常使用，請檢查以下項目：

1. 啟動 Windows 工作管理員 » [效能] » [資源監視器] » [網路] » [接聽連接埠] 查看 opcuauds.exe 使用的連接埠號。由下圖中可看到，目前這台電腦的 opcuauds.exe 使用的連接埠為 4840。



2. 在網頁瀏覽器輸入 HMI 的 IP，並輸入密碼進行登入。至 [OPCUA] 設定頁重新啟動 OPC UA 伺服器。註：此 OPC UA 頁籤只支援於 cMT Gateway 系列。



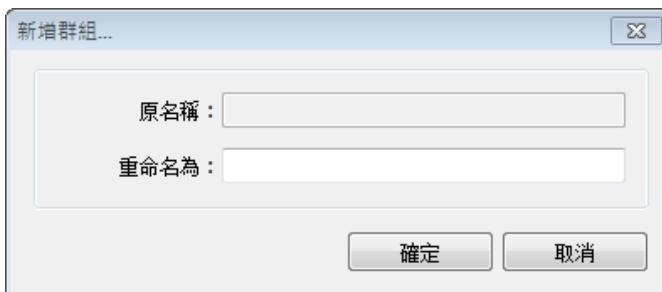
#### 42.2.2.4. 標籤設定



#### 設定

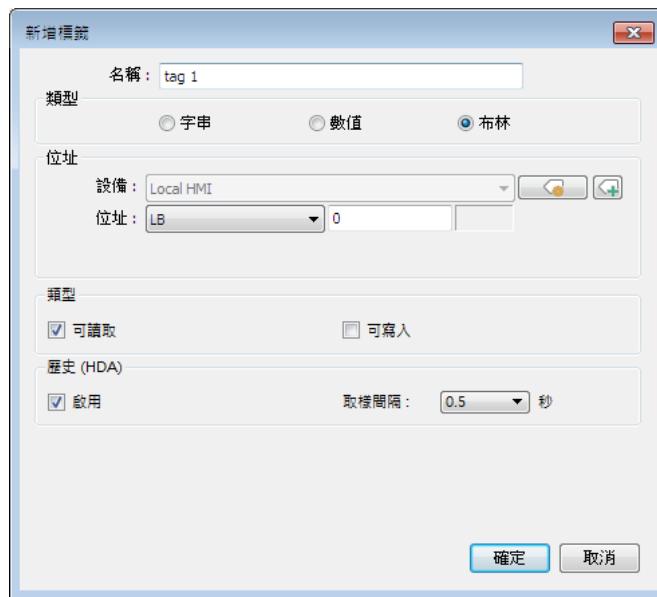
新增群組

#### 描述



新增群組來管理標籤。

### 新增標籤

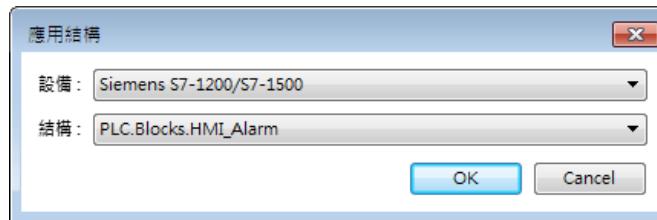


新增在客戶端監控的標籤。在此可選擇該地址是否可寫入且名稱不可空白。

#### 歷史(HDA)

啟用 OPC UA HDA 功能。

### 應用結構



使用者可以在設備下添加結構化節點集，前提是設備為標籤型 PLC 且已有事先定義的結構化資料類型。  
當按下 OK 後需選擇若 OPC UA 節點樹中不存在節點，是否要自行建立。



### 設定

設定已存在的群組或標籤。

### 刪除

刪除已存在的群組或標籤。

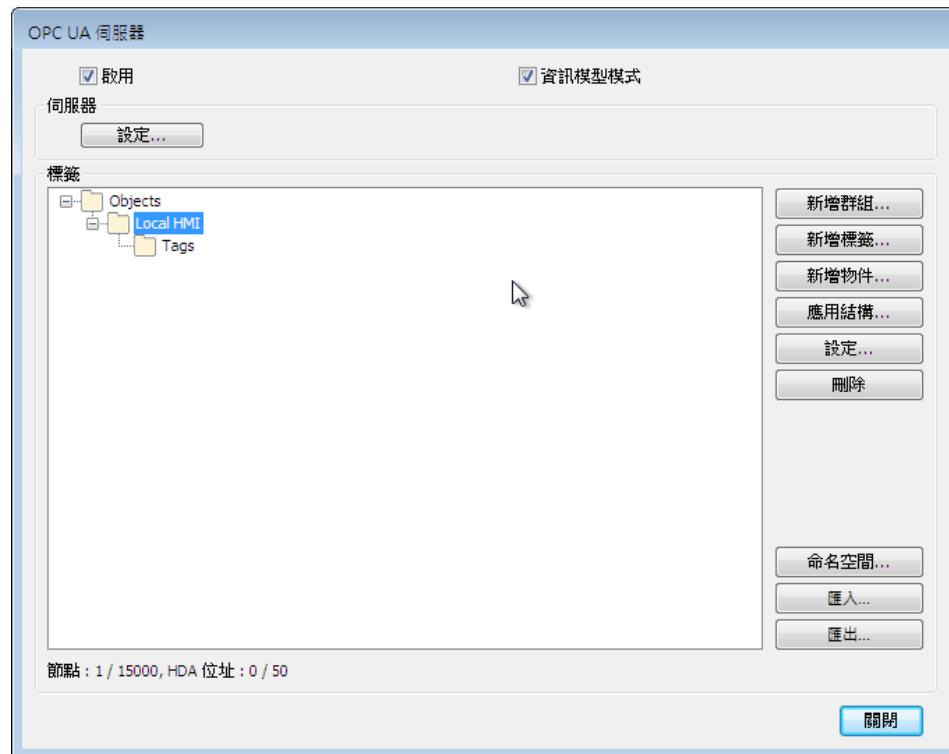
### 匯入

匯入先前設定的標籤。可匯入\*.xlsx, \*.xls, \*.csv, \*.xml 檔案。

### 匯出

將目前設定的標籤匯出。可匯出為 Excel 格式或是 XML 格式。

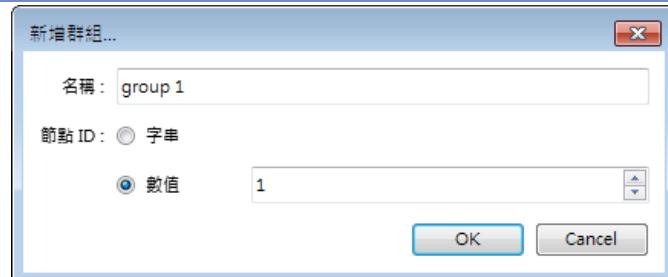
#### 42.2.2.5. 標籤設定 - 資訊模型模式



##### 設定

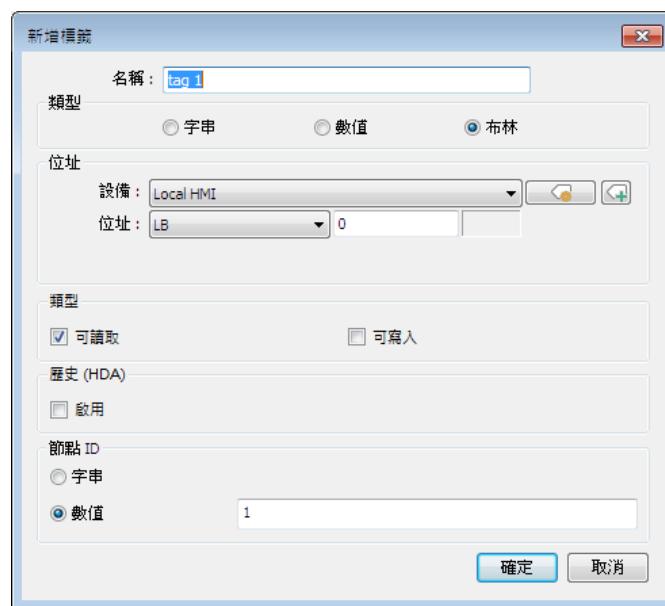
##### 描述

##### 新增群組



新增群組來管理標籤，可自行定義節點 ID。

## 新增標籤



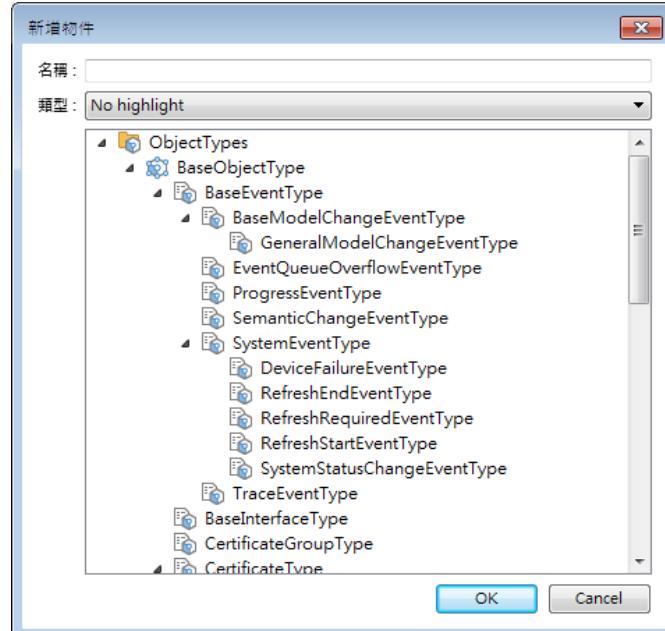
新增在客戶端監控的標籤，標籤分為資料變數與屬性。資料變數代表設備採集到的資料，其下可以再添加子資料變數與屬性；屬性代表設備的設定參數，其下不能再添加子節點。

標籤設定時可選擇該地址是否可寫入且名稱不可空白，可自行定義節點 ID。

### 歷史(HDA)

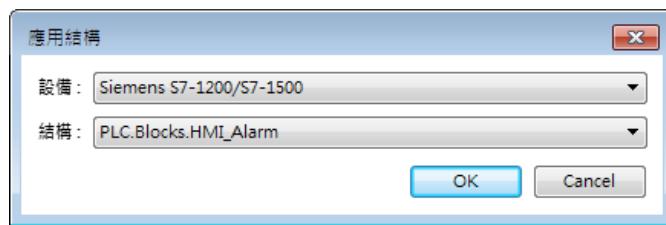
啟用 OPC UA HDA 功能。

## 新增物件

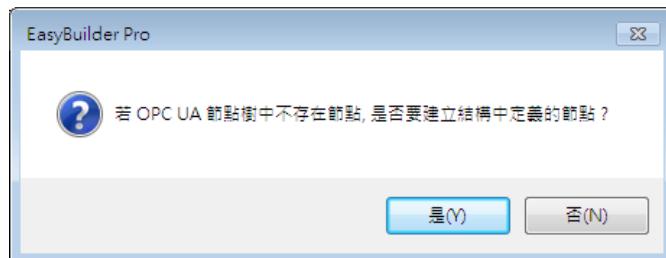


新增一個物件型態中的物件且名稱不可空白。

## 應用結構



使用者可以在設備下添加結構化節點集，前提是設備為標籤型 PLC 且已有事先定義的結構化資料類型。當按下 OK 後需選擇若 OPC UA 節點樹中不存在節點，是否要自行建立。



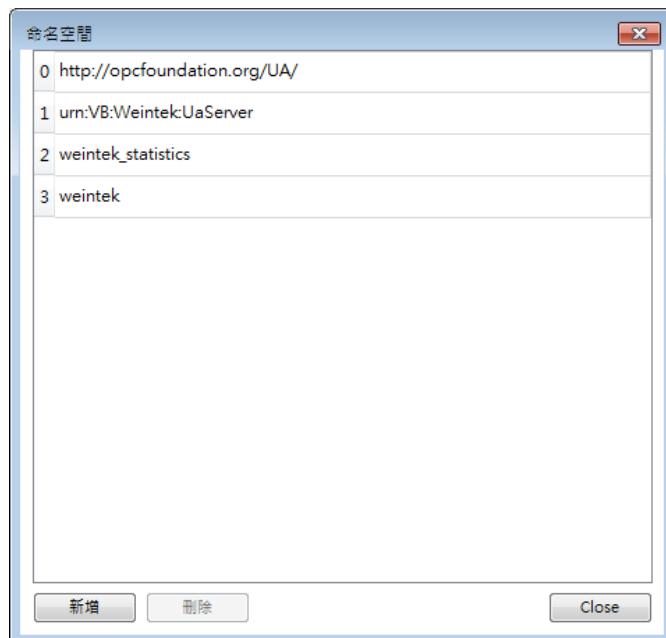
### 設定

設定已存在的群組或標籤。

### 刪除

刪除已存在的群組或標籤。

### 命名空間



在此可新增或刪除設備提供的物件型態。

### 匯入

匯入先前設定的標籤。可匯入\*.xlsx, \*.xls, \*.csv, \*.xml 檔案。

### 匯出

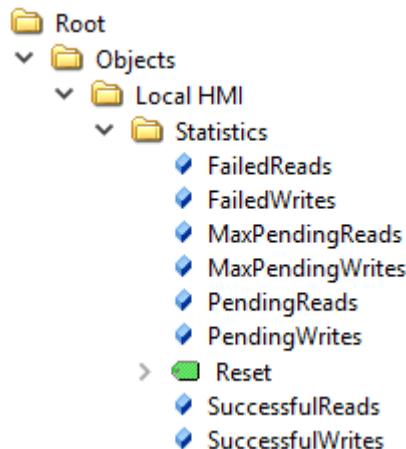
將目前設定的標籤匯出。可匯出為 Excel 格式或是 XML 格式。

 Note

- 將工程檔案下載至人機之前，請先確定人機時間及時區設定皆設定正確，避免 OPC UA 客戶端程式在連線時，因為產生的憑證時間錯誤，造成驗證憑證失敗，導致無法連到 OPC UA 伺服器。
- 可取消資訊模型模式回復為一般模式，但會喪失所有節點的定義。
-  請點選此圖示觀看教學影片，請先確定已連上網路線。

#### 42.2.3. 裝置統計資料

裝置的通訊統計資料可以在各別裝置 Statistics 節點下取得，如下圖：



各節點意義如下：

節點名稱	意義
FailedReads	失敗的讀取命令個數。當為非零時可能有通訊失敗的問題。
FailedWrites	失敗的寫入命令個數。當為非零時可能有通訊失敗的問題。
MaxPendingReads	發生過最大的待讀取命令個數。
MaxPendingWrties	發生過最大的待寫入命令個數。
PendingReads	待讀取的命令個數。當數值長時間持續在一定數量以上，代表通訊模組無法立即消化所有的命令。可能造成 OPC UA 節點更新變慢。在極端條件下，例如長時間在 30 以上，有可能 OPC UA 節點會一段時間不更新。
PendingWrites	待寫入的命令個數。寫入命令較讀取有較高優先權，若此數值長時間過高，將會影響讀取命令。
Reset	重置統計資料。
SuccessfulReads	成功讀取的命令個數。
SuccessfulWrites	成功寫入的命令個數。

#### 42.2.4. 支援及限制

以下簡列 OPC UA 伺服器支援的功能及限制。

項目	描述
OPC UA Profile	<p>Standard UA Server Profile 包含但不僅限於</p> <ul style="list-style-type: none"> <li>* Core Server Facet</li> <li>* UA-TCP UA-SC UA-Binary</li> <li>* SecurityPolicy – None</li> <li>* Enhanced DataChange Subscription Server Facet</li> <li>* Standard DataChange Subscription Server Facet</li> <li>* Embedded DataChange Subscription Server Facet</li> <li>* User Token – X509 Certificate Server Facet</li> <li>* User Token – User Name Password Server Facet</li> <li>* Standard DataChange Subscription Server Facet</li> <li>* Embedded DataChange Subscription Server Facet</li> </ul> <p>相關資訊可參考 <a href="#">Profile Reporting Visualization Tool</a> by OPC Foundation</p>
Security policies	<p>None</p> <p>Basic128Rsa15</p> <p>Basic256</p> <p>Basic256Sha256</p>
最大 OPC UA 節點數	15 000
單一節點最大陣列長度	255
讀取快取時間	<p>100ms</p> <p>(快取將維持 100ms，之後將重新讀取)</p>
Client Session 數量	100
單一 Client Session 可用的 Subscription 個數	64
Publishing Interval 最小值	100ms
OPC UA HDA	<p>*最多支援 50 個節點地址。</p> <p>*每個節點地址可儲存 10000 筆 HDA 資料。</p> <p>節點地址定義？</p> <p>每一個開啟 HDA 的節點，視為使用等同其長度的節點地址。資料類型如為字串，則為對應的字組數量。</p> <p>*當 HMI 記憶體剩餘空間少於 10%時，系統會開始刪除最早的 HDA 資料，以儲存新的資料。直到記憶體剩餘空間</p>

	多於 10%時，系統才會停止刪除舊的資料。
<b>效能</b>	
最大讀取吞吐量 (Security: None)	內建暫存器 (例如 LW): 27000 words/second (WPS) MODBUS RTU@9600bps: 500 WPS MODBUS RTU@115200bps: 4000 WPS MODBUS TCP/IP: 10000 WPS
測試環境 EBPro version: V6.02.02.242 cMT-G02 OS version: 20180917 測試時使用陣列節點以優化讀取效率。	

 **Note**

## ■ OPC UA HDA 節點地址範例:

若有 50 個節點 (node1、node2、...、node 50)，每個節點皆映射到長度為 1 的 bit 地址，共使用 50 個節點地址。

若是一個節點對應到一個長度為 50 的 16-bit Unsigned 整數陣列(長度設定為 50)，則該陣列的元素個別皆為一個節點地址，所以該節點計為使用 50 個節點地址。

若是一個節點為對應到字組數量為 50 的字串，該節點共使用 50 個節點地址。

# 43. JS 物件和 JS 庫

本章介紹 JS 物件和 JS 庫的語法、編輯、及使用方法。

43.1.	概要 .....	43-2
43.2.	JS 物件 .....	43-3
43.3.	JS Resource .....	43-6
43.4.	範例講解 .....	43-7
43.5.	規格限制 .....	43-18
43.6.	JS 物件的運作流程 .....	43-18

## 43.1. 概要

在實際 HMI 的應用中，有些需求無法透過 EasyBuilder Pro 內建的功能達成時，可以考慮使用 JS 物件。透過撰寫 JavaScript 程式，自由地控制物件行為及外觀，達成特定目標。

### 43.1.1. 使用建議

應儘可能使用 EasyBuilder Pro 其它內建功能或物件來達成需求，除非需求難以 (或完全無法) 透過 EasyBuilder Pro 內建功能來完成時，才考慮使用 JS 物件。

### 43.1.2. 環境規格

- EasyBuilder Pro V6.05.01 及之後的版本。
- 支援的 JavaScript 版本：ECMAScript 2017 (SharedArrayBuffer 和 Atomics 除外)。
- 支援的 HMI 型號：cMT X 系列。
- 32-bit Android 裝置不支援 JS Object。

### 43.1.3. 警告

JS 物件提供了強大的物件定制能力，但錯誤的使用方式可能造成系統行為異常或效能降低，請小心使用。

### 43.1.4. JS 物件 SDK

JS 物件支援的 SDK 請參考以下連結：

[https://dl.weintek.com/public/Document/JS\\_Object\\_SDK/Current/index.html](https://dl.weintek.com/public/Document/JS_Object_SDK/Current/index.html)

## 43.2. JS 物件

### 設定

JS 物件的屬性配置，包含下列幾種資料：

1. Address
2. Subscription
3. 非固定，且希望由 JS 物件的使用者，依使用情境決定其內容，進而影響 JS 物件執行期 (runtime) 行為的參數。

屬性配置在執行期會經由一個 JavaScript 物件 – ‘config’，注入到 JS 物件身上 ⇔ `this.config`。



設定	描述
新增物件	新增一個 JavaScript 物件。物件中可再建立多個屬性，每個

	屬性各自有名稱及值。
<b>新增陣列</b>	新增一個資料陣列。
<b>新增數值</b>	新增一個屬性，可以是下列型別： <b>String</b> 長度最大可支援 1000 個字元。 <b>Number</b> 可支援 64 位元浮點數的範圍。 <b>Boolean</b> True 或是 False。 <b>Address</b> 設備地址。 <b>Subscription</b> 地址訂閱。用於監測設備地址的資料變化，當資料改變時，系統會透過在 <code>Subscription.onResponse</code> 方法中註冊的回調 (callback) 函式通知。
<b>刪除</b>	刪除選定的欄位。
<b>設定</b>	修改選定的欄位。
<b>複製</b>	複製選定的欄位。
<b>貼上</b>	在選擇的欄位貼上之前複製的內容。
<b>範本</b>	將目前屬性配置以 JSON 格式呈現。修改 JSON 格式並在按下 [確認] 後，JSON 內容將轉換回屬性配置。 熟悉 JSON 的用戶可以此方式縮短屬性配置的設定時間。

## 原始碼

[原始碼] 為 JavaScript 程式，其內容決定了此 JS 物件的執行期行為。

新增 JS 物件

設定 原始碼 圖片 字型

原始碼 (JavaScript)

```
1 this.config.radiusSub.onResponse((err, data)
2   if (err) {
3     console.log('[response] error =' + err)
4   } else {
5     let radius = data.values[0];
6     let a = area(radius);
7     driver.setData(this.config.areaAddress)
8   }
9 });
10 );
```

JS Object SDK

確定 取消 說明

---

形式(Type)	描述
Compile	編譯程式碼，可確認程式碼內容是否有效。
	使用彈出視窗編寫程式碼。
JS Object SDK	開啟 JS Object SDK Documentation 連結，用於協助開發人員創建及編寫 JS 物件。

### 43.3. JS Resource

當在 JS 物件中需要使用外部 JS 模塊時，必須先將 JS 模塊加到 EBpro 專案的 JS Resource 中，然後才能在 JS 物件的 [原始碼] 中引入該模塊。

下圖為 JS Resource 對話窗，以下逐一解釋：



設定	描述
開啟包含資料夾	當開啟專案時，EBpro 會在電腦系統中暫時建立一資料夾，並將 JS Resource 內容匯出到此資料夾。當儲存專案時，EBpro 也會反向地將資料夾的內容同步回專案中。
新資料夾	建立新資料夾。
新增資料夾	匯入一整個資料夾。
新增檔案	匯入一個檔案。
刪除	刪除資料夾或檔案。
重新命名	修改資料夾或 JS Resource 名稱。
加密	可以將已匯入的.js 檔案加密成.js.enc 檔案，直接開啟加密後的檔案會顯示為亂碼。
解密	將.js.enc 檔案以加密時所設置的密碼解密。
複製路徑	複製選中的檔案或資料夾路徑到剪貼簿，方便在 JS 物件的 [原始碼] 中使用。



- JS Resource 內容上限為：10 MB。
- 包含資料夾在開啟專案時自動產生，在關閉專案時自動刪除。
- 包含資料夾名稱在每次開啟專案時以亂數產生，非固定。
- 包含資料夾的內容變化會讓專案變為 [已修改] 狀態，但直到用戶執行儲存專案時，此資料夾的內容才真正被同步回專案中。
- 加密的密碼長度上限為：32 Bytes。

## 43.4. 範例講解

此章節的目的為透過一系列的教學範例，讓使用者更熟悉如何使用 JS 物件實作。本章節程式碼所使用的函式及類別可參考 [JS Object SDK](#)。

### 43.4.1. 位元切換開關

此章節的目的為學習如何使用 JS 物件逐步實作出 [位元切換開關] 物件的功能與行為，設計流程如下：

1. 設計出可讀取指定位址及輸出反向的 JS 物件
2. 加上另可將指定位址設為 ON 的 JS 物件
3. 加上另可將指定位址設為 OFF 的 JS 物件
4. 加上另可將指定位址做狀態反向的 JS 物件
5. 加上另可將指定位址做復歸型的 JS 物件

為了設計出該物件的功能，在 JS 物件中建立的對應的屬性的名稱標示如下圖。



#### 43.4.1.1. 設計出可讀取指定位址及輸出反向的 JS 物件

本節會說明如何設計出可讀取指定位址及輸出反向的 JS 物件。為了達到 [可讀取指定位址及輸出反向] 的功能，JS 物件須建立 2 個屬性：*readAddressSub* 和 *invertSignal*。

### 設定



屬性	描述
<b>readAddressSub</b>	讀取指定位元位址的狀態。類型選擇 Subscription，Value Type 選擇 Bit，訂閱地址選擇 LB-100。
	<p>數值</p> <p>名稱: <input type="text" value="readAddressSub"/></p> <p>類型: <input type="button" value="Subscription"/></p> <p>默值:</p> <p>類型 <input checked="" type="radio"/> 位元 <input type="radio"/> 字組</p> <p>位址</p> <p>設備: <input type="button" value="Local HMI"/> <input type="button"/> <input type="button"/></p> <p>位址: <input type="button" value="ReadAddress"/></p> <p>* 您可以使用標籤庫中的轉換標籤來轉換數據。</p>
<b>invertSignal</b>	設定是否輸出反向。Type 選擇 Boolean，Value 設定為 False (不進行輸出反向)。
	<p>數值</p> <p>名稱: <input type="text" value="invertSignal"/></p> <p>類型: <input type="button" value="Boolean"/></p> <p>默值:</p> <p><input type="radio"/> true <input checked="" type="radio"/> false</p>

## 原始碼

本節將逐行解釋各程式碼。

- Line2：'this' 即代表 JS 物件本身。透過 '**this.config**' <object> 取得在 [Config] 設定頁中所加入的值('readAddressSub' 和 'invertSignal')。
- Line2：呼叫'**this.config.readAddressSub**'<Subscription> 的 'onResponse' 函式註冊 response callback，讓 [讀取位址] 在數據有變化時主動通知。
- Line 3 ~ 9：根據 response callback 內容進行後續動作。
- Line 3 ~ 4：判斷是否有錯誤。若 err 有值，代表讀取 [讀取位址] 發生錯誤，並簡單列印出錯誤訊息。
- Line 5 ~ 8：若 err 為空值，表示讀取 [讀取位址] 成功且資料有改變。根據讀取到的資料及 '**invertSignal**' 屬性設定 JS 物件狀態。

讀取位址數據	輸出反向	JS 物件狀態
0	False	0
1	False	1
0	True	1
1	True	0

### 43.4.1.2. 設計位元狀態切換開關設定為 ON 模式功能

承前一節範例，本節會再加上可將指定位址 [設為 ON] 模式的功能。如同位元切換開關使用設為 ON 模式功能。為了達到上述功能，JS 物件須再建立 2 個屬性：*writeAddress* 和 *switchStyle*。

## 設定



屬性	描述
<b>readAddressSub</b>	讀取指定位元位址的狀態。
<b>invertSignal</b>	是否輸出反向。
<b>writeAddress</b>	設定被寫入的位址。當使用者按下物件時，會根據 ' <b>switchStyle</b> ' 中的設定將數值寫入至此位址。
<b>switchStyle</b>	設定開關的模式，數值設定為 [Set ON]。
	<div style="border: 1px solid #ccc; padding: 5px;"> <p>數值</p> <p>名稱：<input type="text" value="switchStyle"/></p> <p>類型：<input type="button" value="String"/></p> <p>默值：<input type="text" value="Set ON"/></p> </div>

## 原始碼

```

1  this.config.readAddressSub.onResponse((err, data) => {
2      if (err) {
3          console.log('[response] error =', err.message);
4      } else if (this.config.invertSignal) {
5          this.state = (data.values[0]) ? 0 : 1;
6      } else {
7          this.state = (data.values[0]) ? 1 : 0;
8      }
9  });
10 });
11 var mouseArea = new MouseArea();
12 this.widget.add(mouseArea);
13
14 mouseArea.on('mousedown', (MouseEvent) => {
15     if (this.config.switchStyle === 'Set ON') {
16         driver.setData(this.config.writeAddress, 1);
17     }
18 });
19 });
20

```

透過使用 **MouseArea** 物件，接收手指按下/或釋放的事件，以進而後續的寫入動作。

- Line 1 ~ 11：參考 CH43.4.1.1。
- Line 12：建立一個名為 '**mouseArea**' 的 **MouseArea** 物件。
- Line 13：**'this.widget'**<Container>代表 JS 物件的 Graphical widget，負責畫面的呈現及與使用者的互動。
- Line 13：將 '**mouseArea**' 加入 JS 物件。
- Line 15：向 '**mouseArea**' 註冊 '**mousedown**' 事件的監聽者。當使用者手指按下時，系統會呼叫

此監聽者，並執行 MouseEvent。

- Line 15 ~ 19：'mousedown' 事件的監聽者。
- Line 16 ~ 18：若 [switchStyle] 的值為 'Set ON'，使用 driver module 的 setData Method 將數值 1 寫入到 [writeAddress]。

#### 43.4.1.3. 設計位元切換開關設為 OFF 模式功能

承前一節範例，本節會改為可將指定位址 [設為 OFF] 模式的功能。如同位元切換開關使用設為 OFF 模式功能。

#### 設定

JS 物件物件屬性			
設定			
Name	Type	Value	
config	Object (4)	ReadAddress : LB-100   位元 false	
readAddressSub	訂閱	WriteAddress : LB-100   位元	
invertSignal	布林	Set OFF	
writeAddress	位址標籤		
switchStyle	String		

屬性	描述
<b>readAddressSub</b>	讀取指定位元位址的狀態。
<b>invertSignal</b>	是否輸出反向。
<b>writeAddress</b>	設定被寫入的位址。當使用者按下物件時，會根據 ' <b>switchStyle</b> ' 中的設定將數值寫入至此位址。
<b>switchStyle</b>	設定開關的模式，數值設定為 [Set OFF]。

數值	X
名稱：	<input type="text" value="switchStyle"/>
類型：	<input type="button" value="String"/>
默值：	<input type="text" value="Set OFF"/>

## 原始碼

```

1  this.config.readAddressSub.onResponse((err, data) => {
2      if (err) {
3          console.log('[response] error =', err.message);
4      } else if (this.config.invertSignal) {
5          this.state = (data.values[0]) ? 0 : 1;
6      } else {
7          this.state = (data.values[0]) ? 1 : 0;
8      }
9  });
10 );
11
12 var mouseArea = new MouseArea();
13 this.widget.add(mouseArea);
14
15 mouseArea.on('mousedown', (MouseEvent) => {
16     if (this.config.switchStyle === 'Set ON') {
17         driver.setData(this.config.writeAddress, 1);
18     } else if (this.config.switchStyle === 'Set OFF') {
19         driver.setData(this.config.writeAddress, 0);
20     }
21 });
22

```

- Line 1 ~ 17：參考 CH43.4.1.2。
- Line 18 ~ 19：若 [switchStyle] 的值為 'Set OFF'，寫入 0 到 [writeAddress]。

### 43.4.1.4. 設計位元切換開關設為切換開關模式功能

承前一節範例，本節會改為可將指定位址做狀態反向的功能。如同位元切換開關使用設為切換開關模式功能。

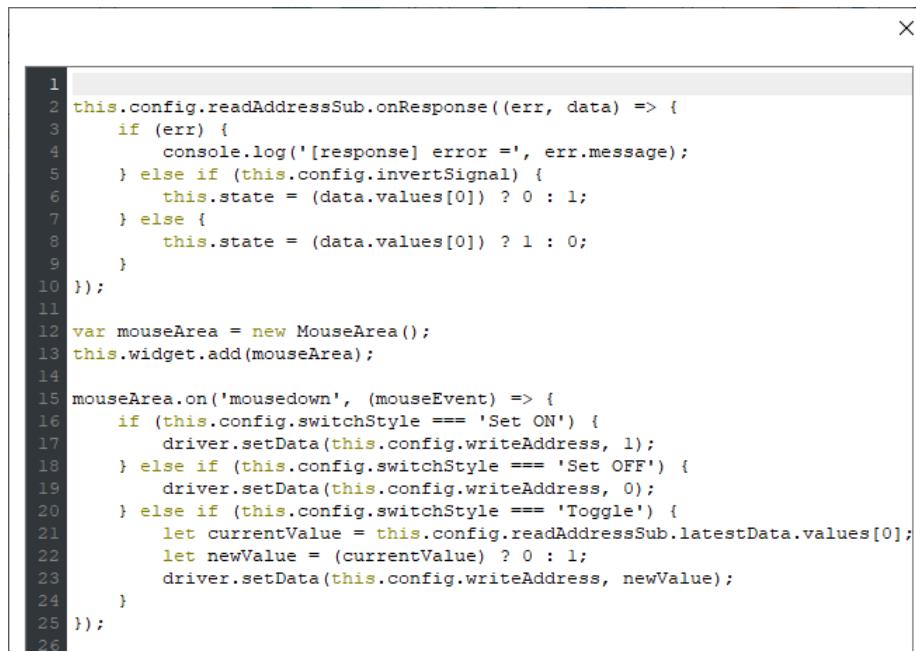
## 設定

JS 物件物件屬性																					
	設定	原始碼	安全																		
	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>config</td> <td>Object (4)</td> <td>ReadAddress : LB-100   位元 false WriteAddress : LB-100   位元 Toggle</td> </tr> <tr> <td>readAddressSub</td> <td>訂閱</td> <td></td> </tr> <tr> <td>invertSignal</td> <td>布林</td> <td>false</td> </tr> <tr> <td>writeAddress</td> <td>位址標籤</td> <td>WriteAddress : LB-100   位元</td> </tr> <tr> <td>switchStyle</td> <td>String</td> <td>Toggle</td> </tr> </tbody> </table>	Name	Type	Value	config	Object (4)	ReadAddress : LB-100   位元 false WriteAddress : LB-100   位元 Toggle	readAddressSub	訂閱		invertSignal	布林	false	writeAddress	位址標籤	WriteAddress : LB-100   位元	switchStyle	String	Toggle		
Name	Type	Value																			
config	Object (4)	ReadAddress : LB-100   位元 false WriteAddress : LB-100   位元 Toggle																			
readAddressSub	訂閱																				
invertSignal	布林	false																			
writeAddress	位址標籤	WriteAddress : LB-100   位元																			
switchStyle	String	Toggle																			

屬性	描述
<b>readAddressSub</b>	讀取指定位元位址的狀態。
<b>invertSignal</b>	是否輸出反向。
<b>writeAddress</b>	設定被寫入的位址。當使用者按下物件時，會根據 ' <b>readAddressSub</b> ' 的狀態和 ' <b>switchStyle</b> ' 中的設定將數值寫入至此位址。
<b>switchStyle</b>	設定開關的模式，數值設定為 [Toggle]。

數值
名稱： <input type="text" value="switchStyle"/>
類型： <input type="text" value="String"/>
默值： <input type="text" value="Toggle"/>

## 原始碼



```

1  this.config.readAddressSub.onResponse((err, data) => {
2    if (err) {
3      console.log('[response] error =', err.message);
4    } else if (this.config.invertSignal) {
5      this.state = (data.values[0]) ? 0 : 1;
6    } else {
7      this.state = (data.values[0]) ? 1 : 0;
8    }
9  });
10 });
11
12 var mouseArea = new MouseArea();
13 this.widget.add(mouseArea);
14
15 mouseArea.on('mousedown', (MouseEvent) => {
16   if (this.config.switchStyle === 'Set ON') {
17     driver.setData(this.config.writeAddress, 1);
18   } else if (this.config.switchStyle === 'Set OFF') {
19     driver.setData(this.config.writeAddress, 0);
20   } else if (this.config.switchStyle === 'Toggle') {
21     let currentValue = this.config.readAddressSub.latestData.values[0];
22     let newValue = (currentValue) ? 0 : 1;
23     driver.setData(this.config.writeAddress, newValue);
24   }
25 });
26

```

- Line 1 ~ 19：參考 CH43.4.1.3。
- Line 20 ~ 23：若 [switchStyle] 的值為 'Toggle'，將 [writeAddress] 的狀態反相。

### 43.4.1.5. 設計位元切換開關復歸型模式功能

承前一節範例，本節會改為可將指定位址使用【復歸型】模式的功能。如同位元切換開關使用復歸型模式功能。

## 設定



屬性	描述
<b>readAddressSub</b>	讀取指定位元位址的狀態。
<b>invertSignal</b>	是否輸出反向。
<b>writeAddress</b>	設定被寫入的位址。當使用者按下物件時，會將數值 1 寫入此位址；當使用者手放開物件時，會將 0 寫入此位址。
<b>switchStyle</b>	設定開關的模式，數值設定為 [Momentary]。



## 原始碼

```

1  this.config.readAddressSub.onResponse((err, data) => {
2    if (err) {
3      console.log('[response] error =', err.message);
4    } else if (this.config.invertSignal) {
5      this.state = (data.values[0]) ? 0 : 1;
6    } else {
7      this.state = (data.values[0]) ? 1 : 0;
8    }
9  });
10 );
11
12 var mouseArea = new MouseArea();
13 this.widget.add(mouseArea);
14
15 mouseArea.on('mousedown', (MouseEvent) => {
16   if (this.config.switchStyle === 'Set ON') {
17     driver.setData(this.config.writeAddress, 1);
18   } else if (this.config.switchStyle === 'Set OFF') {
19     driver.setData(this.config.writeAddress, 0);
20   } else if (this.config.switchStyle === 'Toggle') {
21     let currentValue = this.config.readAddressSub.latestData.values[0];
22     let newValue = (currentValue) ? 0 : 1;
23     driver.setData(this.config.writeAddress, newValue);
24   } else if (this.config.switchStyle === 'Momentary') {
25     driver.setData(this.config.writeAddress, 1);
26   }
27 });
28
29 mouseArea.on('mouseup', (MouseEvent) => {
30   if (this.config.switchStyle === 'Momentary') {
31     driver.setData(this.config.writeAddress, 0);
32   }
33 });
34 );

```

- Line 1 ~ 23：參考 43.4.1.4。
- Line 24 ~ 25：若 [switchStyle] 的值為 'Momentary'，寫入 1 到 [寫入位址]。
- Line 29 ~ 33：當使用者手指離開物件時，寫入 0 到 [writeAddress]。

### 43.4.2. JS Resource 使用範例

此章節的目的為學習 JS 物件如何匯入在 JS Resource 中的 JS 模塊。

JS 物件目前支援以下兩種引入模塊方式：

1. ES6 動態匯入 (dynamic import)：透過 `import()` 函式。注意：JS 物件不支援 ES6 靜態匯入 (static import)。
2. CommonJS：透過 `require()` 函式。

#### 43.4.2.1. 使用 ES6 動態匯入

此範例示範如何使用 ES6 動態匯入 JS 模塊。

在此 JS 物件中，當圓半徑改變時，將使用外部 JS 模塊 (*circle.js*) 裡的 `area` 及 `circumference` 函式，計算圓的面積及周長，並將結果輸出到指定地址。

*circle.js* 程式碼如下：

```
// circle.js

export function area(radius) {
    return Math.PI * radius * radius;
}

export function circumference(radius) {
    return 2 * Math.PI * radius;
}
```

1. 建立 JS 物件的屬性如下圖。



2. 在 [Source code] 設定頁編寫以下程式碼：

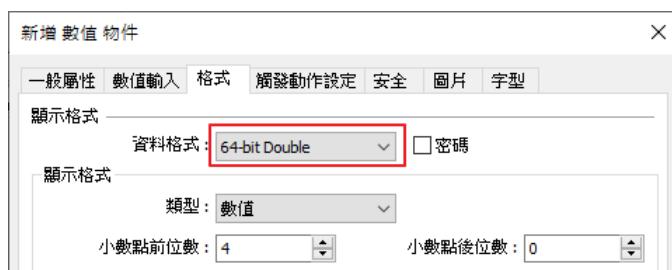
```

1 const circle = await import('/circle.js');
2
3 this.config.radiusSub.onResponse((err, data) => {
4     if (err) {
5         console.log('[response] error =', err.message);
6     } else {
7         let radius = data.values[0];
8         let area = circle.area(radius);
9         let circumference = circle.circumference(radius);
10        driver.setData(this.config.areaAddress, area);
11        driver.setData(this.config.circumferenceAddress, circumference);
12    }
13 });

```

- Line 1：匯入 JS Resource 中的 '/circle.js' 檔案，並將結果綁定到 **circle** 上。
- Line 3：呼叫 '**this.config.radiusSub**' 的 'onResponse' 函式註冊 response callback，讓 [circle radius] 在變化時通知 JS 物件。
- Line 8：呼叫 circle.js 的 area 方法計算圓面積，並將結果存在 area 變數。
- Line 9：呼叫 circle.js 的 circumference 方法計算圓周長，並將結果存在 circumference 變數
- Line 10 ~ 11：將圓面積與周長計算結果分別輸出到 areaAddress 與 circumferenceAddress 地址。

3. 在 Project 中建立三個 64 bit Double 數值物件，分別為 Radius (LW-100)、Area (LW-110)、circumference(LW-120)。



4. 輸入數值到 Radius (LW-100)、Area (LW-110) 和 circumference(LW-120) 會自動計算出數值。

### Note

- 若要匯入 JS 模塊的默認輸出 (/default export)，可以使用以下方式：

```

1 var myModule = (await import('my-module.js')).default;
2

```

#### 43.4.2.2. 使用 CommonJS 匯入

此範例示範如何使用 `require()` 函式匯入 JS 模塊。

此 JS 物件行為與前例相同，但使用 CommonJS 方式匯入外部 JS 模塊。

*circle.js* 程式碼如下：

```
// circle.js

exports.area = function (radius) {
    return Math.PI * radius * radius;
};

exports.circumference = function (radius) {
    return 2 * Math.PI * radius;
}
```

1. JS 物件屬性與前例相同。
2. 在 [原始碼] 設定頁編寫以下程式碼：

```
1 const circle = require('/circle.js');

2 this.config.radiusSub.onResponse((err, data) => {
3     if (err) {
4         console.log('[response] error =', err.message);
5     } else {
6         let radius = data.values[0];
7         let area = circle.area(radius);
8         let circumference = circle.circumference(radius);
9         driver.setData(this.config.areaAddress, area);
10        driver.setData(this.config.circumferenceAddress, circumference);
11    }
12 });
13 });
14 }
```

- Line 1：匯入 JS Resource 中的 '/circle.js' 檔案，並將結果綁定到 **circle** 上。
  - Line 3 ~ 13：參考 CH43.4.2.1。
3. 在 Project 中建立三個 64 bit Double 數值物件，分別為 Radius (LW-100)、Area (LW-110)、circumference(LW-120)。
  4. 輸入數值到 Radius (LW-100)，Area (LW-110) 和 circumference(LW-120) 會自動計算出數值。

### 43.5. 規格限制

1. JS 物件原始碼長度限制：100KB。
2. JS Resource 總計檔案大小限制：10MB。
3. JS Context (\*) 的記憶體用量限制：20MB。

(\*) JS context 為 Javascript 執行環境，有一些內建物件與函數。在人機上，同一個視窗的所有 JS 物件會共用同一個 JS context，因此也會共用 memory heap 及 global object。

### 43.6. JS 物件的運作流程

1. 建立 JS Context。
2. 建立 JS 物件。
3. 為 JS 物件初始化 ‘config’ ( $=>$  this.config)。
4. 為 JS 物件初始化 ‘widget’ ( $=>$ this.widget)。
5. 等待所有 subscriptions 的回應。
6. 將 JS 物件的 [原始碼] 包裝成一個非同步函數並呼叫它。

# 附錄A. 各系列 HMI 軟體功能差異

- eMT Series: eMT3070B, eMT3105P, eMT3120A, eMT3150A
- cMT Series: cMT-SVR-100, cMT-SVR-102, cMT-SVR-200, cMT-SVR-202, cMT-iV5, cMT3151, cMT3090, cMT3071, cMT3072, cMT3103, cMT-HDM, cMT-FHD
- cMT X Series: cMT3072X, cMT3072X2, cMT3072XH, cMT3072XH2, cMT3072XHT, cMT3072XP, cMT3092X, cMT3092X (V2), cMT3102X, cMT3102X (V2), cMT3108XH, cMT3108XP, cMT3152X, cMT3161X, cMT3162X, cMT3162X (V2), cMT2058XH, cMT2078X, cMT2108X, cMT2108X2, cMT2108X2 (V2), cMT2128X, cMT2158X, cMT2158X (V2), cMT2166X, cMT1106X, cMT-FHDX-220, cMT-FHDX-820, cMT-SVRX-820, cMT-SVRX-822
- cMT Series-Gateway: cMT-G01, cMT-G02, cMT-G03, cMT-G04, cMT-CTRL01
- mTV Series: mTV-100
- iE Series: MT8050iE, MT8051iE, MT8053iE, MT6070iE, MT6071iE, MT8070iE, MT8071iE, MT8073iE, MT8100iE, MT8101iE, MT8102iE, MT8103iE
- iER Series: MT8070iER1
- XE Series: MT8090XE, MT8091XE, MT8092XE, MT8121XE, MT8150XE
- iP Series: MT6051iP, MT8051iP, MT6071iP, MT8071iP, MT8071iP2, MT6103iP, MT8102iP

請注意，以下表格僅供參考，可能不包含所有功能。部分功能是否有效，與軟體版本有關，當有疑問時請再次確認所使用的軟體，或是向 Weintek 諮詢。

## iP/iE/eMT/mTV/XE 系列

類別	型號 功能	iP	iE	eMT	mTV	XE	備註
EasyBuilder	工程檔案上限	22.5MB	22.5MB	64MB	64MB	64MB	
	USB 碟下載工程	Y	Y	Y	Y	Y	
功能	64-bit 資料型態	N/A	N/A	N/A	N/A	N/A	
	3G/4G 行動網路	Y	N/A	N/A	N/A	N/A	1
	控制權	N/A	N/A	N/A	N/A	N/A	
	雙 USB	Y	Y	N/A	N/A	N/A	
	電子郵件	N/A	Y	Y	Y	Y	
	進階安全模式	Y	Y	Y	Y	Y	
	乙太網路印表機	N/A	N/A	N/A	N/A	N/A	
	LDAP	N/A	N/A	N/A	N/A	N/A	
	字串表	N/A	Y	Y	Y	Y	
	時間同步 (NTP)	N/A	Y	Y	Y	Y	
	USB 數據連線	Y	Y	Y	Y	Y	a, 2
	使用者自訂開機畫面	Y	Y	Y	Y	Y	
	VNC Server	Y	Y	Y	Y	Y	3
	影像串流	N/A	N/A	Y	Y	Y	4
硬體	網頁介面	N/A	N/A	N/A	N/A	N/A	
	EasyWeb 2.0	N/A	N/A	N/A	N/A	N/A	
	WebView	N/A	N/A	N/A	N/A	N/A	
	MPI	Y	Y	Y	Y	Y	
	聲音(喇叭)	N/A	N/A	eMT3120 eMT3150	N/A	N/A	
	聲音(其他)	N/A	N/A	N/A	HDMI	USB	10
	CAN Bus	N/A	N/A	Y	N/A	8091XE 8092XE	
	WiFi	N/A	8103iE	N/A	N/A	N/A	
	震動提示	N/A	N/A	N/A	N/A	N/A	

## 各系列 HMI 軟體功能差異

類別	型號 功能	iP	iE	eMT	mTV	XE	備註
物件	動作觸發	N/A	N/A	N/A	N/A	N/A	
	BACnet Schedule	N/A	N/A	N/A	N/A	N/A	
	條碼掃描器	N/A	N/A	N/A	N/A	N/A	5
	複合式多功能按鈕	Y	Y	Y	Y	Y	
	觸發條件	N/A	N/A	N/A	N/A	N/A	
	日期/時間	Y	Y	Y	Y	Y	
	動態繪圖	Y	Y	Y	Y	Y	
	動態刻度	Y	Y	Y	Y	Y	
	檔案瀏覽器	Y	Y	Y	Y	Y	
	流動塊	Y	Y	Y	Y	Y	
	JS 物件	N/A	N/A	N/A	N/A	N/A	
	媒體播放器	N/A	N/A	eMT3105 eMT3121 eMT3150	N/A	Y	6
	PDF 檢視器	N/A	N/A	N/A	N/A	N/A	
	圖片檢視	Y	Y	Y	Y	Y	
	圓餅圖	Y	Y	Y	Y	Y	
	PLC 網頁瀏覽器	N/A	N/A	N/A	N/A	N/A	11
	表格	Y	Y	Y	Y	Y	
	模板視窗	Y	Y	Y	Y	Y	
	觸控手勢	N/A	N/A	N/A	N/A	N/A	
IIoT/Energy	網路攝影機	N/A	N/A	Y	N/A	Y	7
	USB 攝影機	N/A	N/A	Y	Y	Y	12
	影像輸入	N/A	N/A	eMT3121 eMT3150	N/A	N/A	
	VNC Viewer	N/A	Y	Y	Y	Y	8
	能源需量顯示	N/A	N/A	N/A	N/A	N/A	
	能源需量設定	N/A	N/A	N/A	N/A	N/A	
	MQTT	N/A	Y	Y	Y	Y	
Data/History	MQTT-進階 JSON	N/A	N/A	N/A	N/A	N/A	
	MQTT-AWS IoT, Sparkplug B, Azure IoT Hub	N/A	N/A	N/A	N/A	N/A	
	OPC UA 客戶端	Y	Y	Y	Y	Y	
	OPC UA 伺服器	N/A	N/A	N/A	N/A	N/A	
	圓盤曲線	Y	Y	Y	Y	Y	
	資料庫伺服器	N/A	N/A	N/A	N/A	N/A	b
	事件甘特圖	N/A	N/A	N/A	N/A	N/A	
其他	FTP 伺服器	N/A	N/A	N/A	N/A	N/A	b
	操作記錄設定	Y	Y	Y	Y	Y	
應用	SQL 查詢	N/A	N/A	N/A	N/A	N/A	
	配方資料庫	Y	Y	Y	Y	Y	
	配方匯入/匯出	Y	Y	Y	Y	Y	
	配方檢視	Y	Y	Y	Y	Y	
	CODESYS	N/A	N/A	N/A	N/A	N/A	
應用	cMT Viewer App	N/A	N/A	N/A	N/A	N/A	
	cMT Diagnoser	N/A	N/A	N/A	N/A	N/A	
	EasyAccess 1.0	Y	Y	Y	Y	Y	
	EasyDiagnoser	Y	Y	Y	Y	Y	

## 各系列 HMI 軟體功能差異

類別	型號 功能	iP	iE	eMT	mTV	XE	備註
Weincloud	EasyPrinter	Y	Y	Y	Y	Y	
	EasySimulator	Y	Y	Y	Y	Y	
	EasySystemSetting	Y	Y	Y	Y	Y	9
	EasyWatch	Y	Y	Y	Y	Y	
Weincloud	Dashboard	N/A	N/A	N/A	N/A	N/A	
	EasyAccess 2.0	Y	Y	Y	Y	Y	
	EasyAccess 2.0 內建授權	N/A	8053iE 8073iE 8103iE	N/A	N/A	8092XE	

## cMT 系列

	系列	cMT						cMT Gateway	備註
類別	型號 功能	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD		
EasyBuilder	工程檔案上限	32MB	64MB	64MB	64MB	64MB	64MB	32MB	
	USB 碟下載工程	N/A	Y	Y	Y	Y	Y	N/A	
功能	64-bit 資料型態	Y	Y	Y	Y	Y	Y	Y	
	3G/4G 行動網路	Y	Y	Y	Y	Y	Y	N/A	1
	控制權	Y	Y	Y	Y	Y	Y	N/A	
	雙 USB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	電子郵件	Y	Y	Y	Y	Y	Y	Y	
	進階安全模式	Y	Y	Y	Y	Y	Y	Y	
	乙太網路印表機	N/A	Y	Y	Y	Y	Y	N/A	
	LDAP	Y	Y	Y	Y	Y	Y	Y	
	字串表	Y	Y	Y	Y	Y	Y	N/A	
	時間同步(NTP)	Y	Y	Y	Y	Y	Y	Y	
	USB 數據連線	Y	Y	Y	Y	Y	Y	N/A	a, 2
	使用者自訂開機畫面	N/A	Y	Y	Y	Y	Y	N/A	
	VNC Server	N/A	Y	Y	Y	Y	Y	N/A	3
	影像串流	Y	Y	Y	Y	Y	Y	N/A	4
	網頁介面	Y	Y	Y	Y	Y	Y	Y	
	EasyWeb 2.0	Y	N/A	N/A	N/A	N/A	N/A	N/A	
	WebView	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	MPI	Y	Y	Y	Y	Y	Y	Y	
硬體	聲音輸出(喇叭)	N/A	N/A	Y	N/A	Y	N/A	N/A	
	聲音輸出(其他)	N/A	N/A	N/A	N/A	N/A	HDMI	N/A	10
	CAN Bus	N/A	Y	Y	Y	Y	N/A	N/A	
	WiFi	cMT-SVR-200/202	N/A	N/A	Y	N/A	N/A	cMT-G02	
	震動提示	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
物件	動作觸發	Y	Y	Y	Y	Y	Y	N/A	
	BACnet Schedule	Y	Y	Y	Y	Y	Y	N/A	
	條碼掃描器	Y	Y	Y	Y	Y	Y	N/A	5
	複合式多功能按鈕	Y	Y	Y	Y	Y	Y	N/A	
	觸發條件	Y	Y	Y	Y	Y	Y	Y	
	日期/時間	Y	Y	Y	Y	Y	Y	N/A	

	系列	cMT						cMT Gateway	備註
類別	型號 功能	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD	cMT-G01 cMT-G02 cMT-G03 cMT-G04 cMT-CTRL01	
IIoT/Energy	動態繪圖	Y	Y	Y	Y	Y	Y	N/A	
	動態刻度	Y	Y	Y	Y	Y	Y	N/A	
	檔案瀏覽器	N/A	Y	Y	Y	Y	Y	N/A	
	流動塊	Y	Y	Y	Y	Y	Y	N/A	
	JS 物件	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	媒體播放器	N/A	Y	Y	Y	Y	Y	N/A	6
	PDF 檢視器	N/A	Y	Y	Y	Y	Y	N/A	
	圖片檢視	N/A	Y	Y	Y	Y	Y	N/A	
	圓餅圖	Y	Y	Y	Y	Y	Y	N/A	
	PLC 網頁瀏覽器	N/A	N/A	N/A	N/A	N/A	N/A	N/A	11
	表格	Y	Y	Y	Y	Y	Y	N/A	
	模板視窗	Y	Y	Y	Y	Y	Y	N/A	
	觸控手勢	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	網路攝影機	N/A	Y	Y	Y	Y	Y	N/A	7
	USB 攝影機	N/A	Y	Y	Y	Y	Y	N/A	12
	影像輸入	N/A	N/A	N/A	N/A	Y	N/A	N/A	
	VNC Viewer	N/A	Y	Y	Y	Y	Y	N/A	8
Data/History	能源需量顯示	Y	Y	Y	Y	Y	Y	N/A	
	能源需量設定	Y	Y	Y	Y	Y	Y	N/A	
	MQTT	Y	Y	Y	Y	Y	Y	Y	
	MQTT-進階 JSON	Y	Y	Y	Y	Y	Y	Y	
	MQTT - AWS IoT, Sparkplug B, Azure IoT Hub	Y	Y	Y	Y	Y	Y	Y	
	OPC UA 客戶端	Y	Y	Y	Y	Y	Y	Y	
	OPC UA 伺服器	License	Y	Y	Y	Y	License	Y	
	圓盤曲線	Y	Y	Y	Y	Y	Y	N/A	
應用	資料庫伺服器	Y	Y	Y	Y	Y	Y	Y	b
	事件甘特圖	Y	Y	Y	Y	Y	Y	N/A	
	FTP 伺服器	Y	Y	Y	Y	Y	Y	N/A	b
	操作記錄設定	Y	Y	Y	Y	Y	Y	N/A	
	SQL 查詢	Y	Y	Y	Y	Y	Y	Y	
	配方資料庫	Y	Y	Y	Y	Y	Y	N/A	
	配方匯入/匯出	Y	Y	Y	Y	Y	Y	N/A	
	配方檢視	Y	Y	Y	Y	Y	Y	N/A	
其他	CODESYS	N/A	License	License	N/A	License	N/A	cMT-CTRL01	
Weincloud	cMT Viewer App	Y	Y	Y	Y	Y	Y	N/A	
	cMT Diagnoser	Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 1.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasyDiagnoser	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasyPrinter	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasySimulator	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasySystemSetting	N/A	Y	Y	Y	Y	Y	N/A	9
	EasyWatch	Y	Y	Y	Y	Y	Y	Y	

	系列	cMT						cMT Gateway	備註
類別	型號 功能	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD	cMT-G01 cMT-G02 cMT-G03 cMT-G04 cMT-CTRL01	
	EasyAccess 2.0	Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 2.0 內建授權	cMT-SVR-102/202	cMT3071: N/A cMT3072: Y		Y	Y	N/A	N/A	

**cMT X 系列**

	型號 功能	cMT3092X cMT3092X (V2) cMT3102X cMT3102X (V2) cMT3108XH cMT3108XP cMT3152X cMT3162X cMT3162X (V2)	cMT3072X cMT3072X2 cMT3072XH cMT3072XH2 cMT3072XHT cMT3072XP	cMT2166X cMT3161X	cMT2058XH cMT2078X cMT2108X cMT2108X2 cMT2108X2 (V2) cMT2128X cMT2158X cMT2158X (V2)	cMT1106X	cMT-FHDX-220 cMT-FHDX-820	cMT-SVRX-820 cMT-SVRX-822	備註
EasyBuilder	工程檔案上限	64MB	64MB	64MB	64MB	64MB	64MB	64MB	
	USB 碟下載工程	Y	Y	Y	Y	Y	Y	Y	
功能	64-bit 資料型態	Y	Y	Y	Y	Y	Y	Y	
	3G/4G 行動網路	Y	Y	Y	Y	Y	Y	Y	1
	控制權	Y	Y	Y	Y	N/A	Y	Y	
	雙 USB	Y	Y	Y	Y	Y	Y	Y	
	電子郵件	Y	Y	Y	Y	Y	Y	Y	
	進階安全模式	Y	Y	Y	Y	Y	Y	Y	
	乙太網路印表機	Y	Y	Y	Y	Y	Y	N/A	
	LDAP	Y	Y	Y	Y	N/A	Y	Y	
	字串表	Y	Y	Y	Y	Y	Y	Y	
	時間同步(NTP)	Y	Y	Y	Y	Y	Y	Y	
	USB 數據連線	Y	Y	Y	Y	Y	Y	Y	a, 2
	使用者自訂開機畫面	Y	Y	Y	Y	Y	Y	N/A	
	VNC Server	Y	Y	Y	Y	Y	Y	N/A	3
	影像串流	Y	Y	Y	Y	N/A	Y	Y	4
	網頁介面	Y	Y	Y	Y	Y	Y	Y	
	EasyWeb 2.0	Y	Y	Y	Y	Y	Y	Y	
	WebView	Y	Y	Y	Y	N/A	Y	Y	
硬體	MPI	Y	Y	cMT2166X	cMT2058XH cMT2078X cMT2108X2(V2) cMT2128X cMT2158X(V2)	Y	Y	Y	
	聲音輸出(喇叭)	cMT3092X cMT3108XH cMT3108XP cMT3152X cMT3162X	N/A	Y	cMT2158X	N/A	N/A	N/A	
	聲音輸出(其他)	N/A	N/A	N/A	N/A	N/A	HDMI	N/A	10
	CAN Bus	cMT3092X (V2) cMT3102X cMT3102X (V2) cMT3108XH cMT3108XP cMT3162X cMT3162X (V2)	cMT3072X cMT3072XH cMT3072XH2 cMT3072XHT cMT3072XP	N/A	N/A	N/A	N/A	N/A	

類別	功能	型號	cMT3092X	cMT3072X	cMT2166X	cMT2058XH	cMT1106X	cMT-FHDX-220	cMT-SVRX-820	備註
			cMT3092X (V2)	cMT3102X	cMT3072X2	cMT2108X		cMT-FHDX-820	cMT-SVRX-822	
	WiFi	cMT3102X, cMT3108XH, cMT3108XP (M02 module)	N/A	N/A	cMT2108X2, cMT2108X2 (V2) (M02 module)	(M02 module)	N/A	N/A	N/A	
	震動提示	cMT3162X	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
物件	動作觸發	Y	Y	Y	Y	N/A	Y	Y		
	BACnet Schedule	Y	Y	Y	Y	Y	Y	Y		
	條碼掃描器	Y	Y	N/A	N/A	N/A	Y	Y	Y	5
	複合式多功能按鈕	Y	Y	Y	Y	Y	Y	Y		
	觸發條件	Y	Y	Y	Y	N/A	Y	Y		
	日期/時間	Y	Y	Y	Y	Y	Y	Y		
	動態繪圖	Y	Y	Y	Y	Y	Y	Y		
	動態刻度	Y	Y	Y	Y	Y	Y	Y		
	檔案瀏覽器	Y	Y	Y	Y	Y	Y	N/A		
	流動塊	Y	Y	Y	Y	Y	Y	Y		
	JS 物件	Y	Y	Y	Y	N/A	Y	Y		
	媒體播放器	Y	Y	Y	Y	N/A	Y	N/A	6	
	PDF 檢視器	Y	Y	Y	Y	Y	Y	N/A		
	圖片檢視	Y	Y	Y	Y	Y	Y	N/A		
	圓餅圖	Y	Y	Y	Y	Y	Y	Y		
	PLC 網頁瀏覽器	Y	Y	N/A	N/A	N/A	cMT-FHDX-820	N/A	11	
	表格	Y	Y	Y	Y	Y	Y	Y		
IIoT/Energy	模板視窗	Y	Y	Y	Y	Y	Y	Y		
	觸控手勢	cMT3108XP cMT3152X cMT3162X	cMT3072XP	Y	N/A	N/A	cMT-FHDX-820	N/A		
	網路攝影機	Y	Y	Y	Y	N/A	Y	N/A	7	
	USB 攝影機	Y	Y	Y	Y	N/A	Y	N/A	12	
	影像輸入	N/A	N/A	N/A	N/A	N/A	N/A	N/A		
	VNC Viewer	Y	Y	Y	Y	Y	Y	N/A	8	
	能源需量顯示	Y	Y	N/A	N/A	N/A	Y	Y		
	能源需量設定	Y	Y	N/A	N/A	N/A	Y	Y		
	MQTT	Y	Y	Y	Y	Y	Y	Y		
	MQTT - 進階 JSON	Y	Y	Y	Y	Y	Y	Y		
Data/History	MQTT - AWS IoT, Sparkplug B, Azure IoT Hub	Y	Y	Y	Y	N/A	Y	Y		
	OPC UA 客戶端	Y	Y	Y	Y	Y	Y	Y		
	OPC UA 雖服器	Y	Y	N/A	N/A	N/A	License	Y		
	圓盤曲線	Y	Y	Y	Y	Y	Y	Y		
	資料庫伺服器	Y	Y	N/A	N/A	N/A	Y	Y	b	
	事件甘特圖	Y	Y	N/A	N/A	N/A	Y	Y		
	FTP 伺服器	Y	Y	Y	Y	Y	Y	Y	b	
	操作記錄設定	Y	Y	Y	Y	Y	Y	Y		
	SQL 查詢	Y	Y	N/A	N/A	N/A	Y	Y		

類別	功能	型號	cMT3092X	cMT3072X	cMT2166X	cMT2058XH	cMT1106X	cMT-FHDX-220 cMT-FHDX-820	cMT-SVRX-820 cMT-SVRX-822	備註
			cMT3092X (V2)	cMT3102X	cMT3072X (V2)	cMT3072XH				
其他	配方資料庫		Y	Y	Y	Y	Y	Y	Y	
	配方匯入/匯出		Y	Y	Y	Y	Y	Y	Y	
	配方檢視		Y	Y	Y	Y	Y	Y	Y	
CODESYS	License	License	N/A	License	N/A	License	License	License	License	
應用	cMT Viewer App		Y	Y	Y	Y	N/A	Y	Y	
	cMT Diagnoser		Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 1.0		N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasyDiagnoser		N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasyPrinter		N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasySimulator		N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasySystemSetting		Y	Y	Y	Y	Y	Y	Y	9
Weincloud	EasyWatch		Y	Y	Y	Y	Y	Y	Y	
	Dashboard		Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 2.0		Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 2.0 內建授權	cMT3092X cMT3152X	N/A	N/A	N/A	N/A	N/A	N/A	cMT-SVR-822	

## 備註

- a. MT6071iE 不支援 USB 數據連線
- b. 連接至外部伺服器

## OS 支援版本

No	功能	OS 版本
1	3G/4G 行動網路	cMT-SVR 20151127 cMT3071/3072/3090/3103/3151: 20180723 cMT-HDM/FHD 20190130
2	USB 數據連線	iE Type A: 20160707 iE Type B: 20160325 eMT: 20160601 mTV: 20160527 XE: 20160503 cMT-SVR: 20160518 cMT3090/3103/3151/HDM: 20160412 或更新版本
3	VNC Server	cMT3090/3151: 20160412 或更新版本
4	影像串流	cMT-SVR: 20151127, cMT3151:20171129 或更新版本
5	條碼掃描器	Android cMTViewer V1.7.0 或更新版本
6	媒體播放器	cMT-HDM: 不支援 1920*1080 解析度 cMT3072X(H): 20200929
7	網路攝影機	eMT: 20160601 XE: 20160503 cMT3072X(H): 20200929
8	VNC Viewer	iE Type A: 20160707 iE Type B: 20160325 eMT: 20160601 mTV: 20160527 XE: 20160503 全 cMT 系列: 20181211 或更新版本
9	EasySystemSetting	全 cMT 系列: 20180723 或更新版本
10	聲音輸出	支援 USB 聲音輸出: XE: 20190325
11	PLC 網頁瀏覽器	cMT3072X(H):20200723, cMT3092X/3152X/3162X: 20201202
12	USB 攝影機	cMT3072X(H): 20200929

## 附錄B. cMT Viewer 使用限制

操作本地 cMT / cMT X 系列人機，可直接在其螢幕上操作，或是使用其他客戶端裝置，執行 cMT Viewer 或 VNC Viewer，進行遠端操作。然而，使用 cMT Viewer 客戶端裝置 (iOS, Android, PC, cMT-iV5, cMT-iV6) 進行遠端連線時，即使能操作本地 cMT / cMT X 人機，部分功能可能無法正常運作。在您設計工程檔案時，請將以上情況列入考慮。

以下列出使用 cMT Viewer 進行遠端操作時，無法正常運作的功能：

- 進階安全模式 - 插件
- 檔案瀏覽器\*註 2
- 媒體播放器\*註 2
- 觸控手勢
- 影像輸入：USB 攝影機
- 影像輸入：網路攝影機\*註 1
- 影像輸入：類比輸入
- 圖片檢視\*註 2
- PDF 檢視器\*註 2
- VNC Viewer\*註 2
- 功能鍵選擇的功能：畫面擷取，匯入使用者資料/使用[USB 安全金鑰] \*註 2
- 自訂鍵盤：iOS 系統不支援，一律使用系統鍵盤
- PLC 控制物件選擇的功能：
  - I. 畫面擷取：當檔案儲存在連接人機的外接裝置時
  - II. 背光燈控制：人機螢幕的背光控制
- 操作記錄選擇的功能：
  - I. 操作記錄列印：當檔案儲存在連接人機的外接裝置時

### Note

1. 在 Windows PC 上，如有安裝第三方解碼器，則可能可以透過 cMT Viewer 檢視網路攝影機影像。可否成功檢視影像，以實際測試結果為主。
2. cMT-iV6、cMT3108XP (iV) / cMT3162X (iV) 支援此功能。

## 附錄C. PC 模擬使用限制

在設計 EasyBuilder Pro 工程檔案時，常需要使用到連線模擬/離線模擬來測試程式運作是否合乎設計，但部分功能仍需要下載到 HMI 後才能測試以及使用。因此，在設計工程檔案時，請將此情況列入考慮。

於 PC 模擬時，歷史資料將儲存於 EB Pro 安裝目錄下的對應資料夾：

- HMI\_memory：當儲存位置設定為 HMI 記憶體。
- SD\_card：當儲存位置設定為 SD 卡。
- usb1：當儲存位置設定為 USB 碟。
- usb2：當儲存位置設定為 USB 碟 2。(部分機型支援連接 2 個 USB 碟)

以下為無法於 PC 連線模擬/離線模擬時運作的功能：

### cMT/cMT X 系列

- 進階安全模式 – 插件
- 媒體播放器
- 影像輸入：USB 攝影機
- 影像輸入：網路攝影機
- 影像輸入：類比輸入
- PDF 檢視器
- VNC Viewer
- PLC 網頁瀏覽器

### eMT/iE/XE/mTV 系列

- 影像輸入：USB 攝影機
- 影像輸入：網路攝影機
- 影像輸入：類比輸入
- VNC Viewer

# 附錄D. 64-bit 資料型態使用注意事項

cMT / cMT X 系列可以支援使用 64-bit 的資料型態，以下為其使用限制。

- 輸入限制 (數值物件、多狀態設定...，等等)

- I. uint64: 0 ~ 2<sup>48</sup>
- II. int64: -2<sup>48</sup> ~ 2<sup>48</sup>
- III. double: -2<sup>48</sup> ~ 2<sup>48</sup>

- int64 / uint64

因目前邏輯計算都以 double 在進行，double 能正確表示整數的部份有 52-bit，約略為 15 ~ 16 位有效數字，超過該範圍則會喪失精準度。

又因為 EasyBuilder Pro 架構的限制，部分輸入框超過 48-bit 可能導致異常，故目前將規則統一訂為 48-bit。

- double

double 類型原本在 EasyBuilder Pro 上的輸入限制是藉由取得 double 本身的最大值/最小值來做為輸入數值的上下限 ( $+/- 1.79 * 10^{308}$ )，但實際上該數值近乎無上下限，故統一將其輸入的上下限訂為  $-2^{48} \sim 2^{48}$ 。

## 範例

使用者透過 PLC 讀取了一數值 (64-bit unsigned int) 144,115,188,075,855,872，經過數值輸入可能會被錯誤顯示為 144,115,188,075,855,870。

## 其他狀況

除了 PLC 數值可能被外部修改導致數值超過 52-bit 而顯示異常外，如配方輸入 / SQL 查詢 / 資料庫 (資料取樣) 等可由外部修改數值的狀況導致其範圍超過，也都可能發生相同問題。

## 注意

double 原本即有精準度的問題，並不代表輸入在  $+2^{48} \sim -2^{48}$  則會正確表示，上述規則僅基於 int64 / uint64 使用上的說明。