

NEXIOT

neXVIC

VIC7000  
Training  
VIC7310

v 1.2.1.8

黃翊凱 Garry Huang



# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

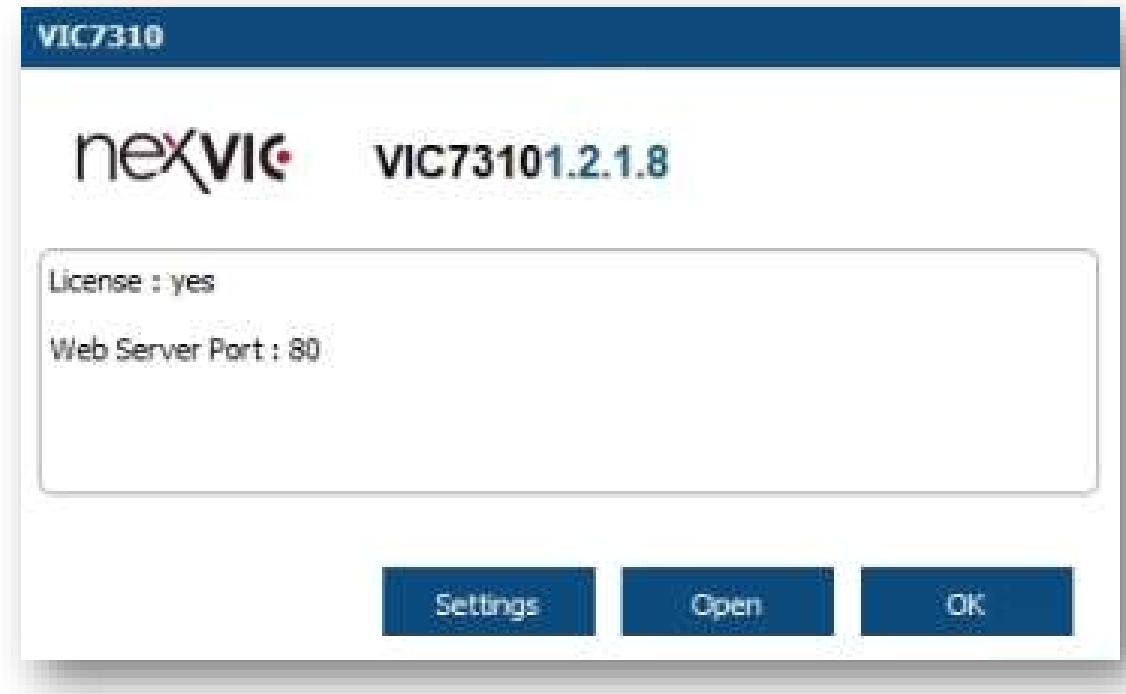
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# 軟體設定

- **開啟 VIC7000**

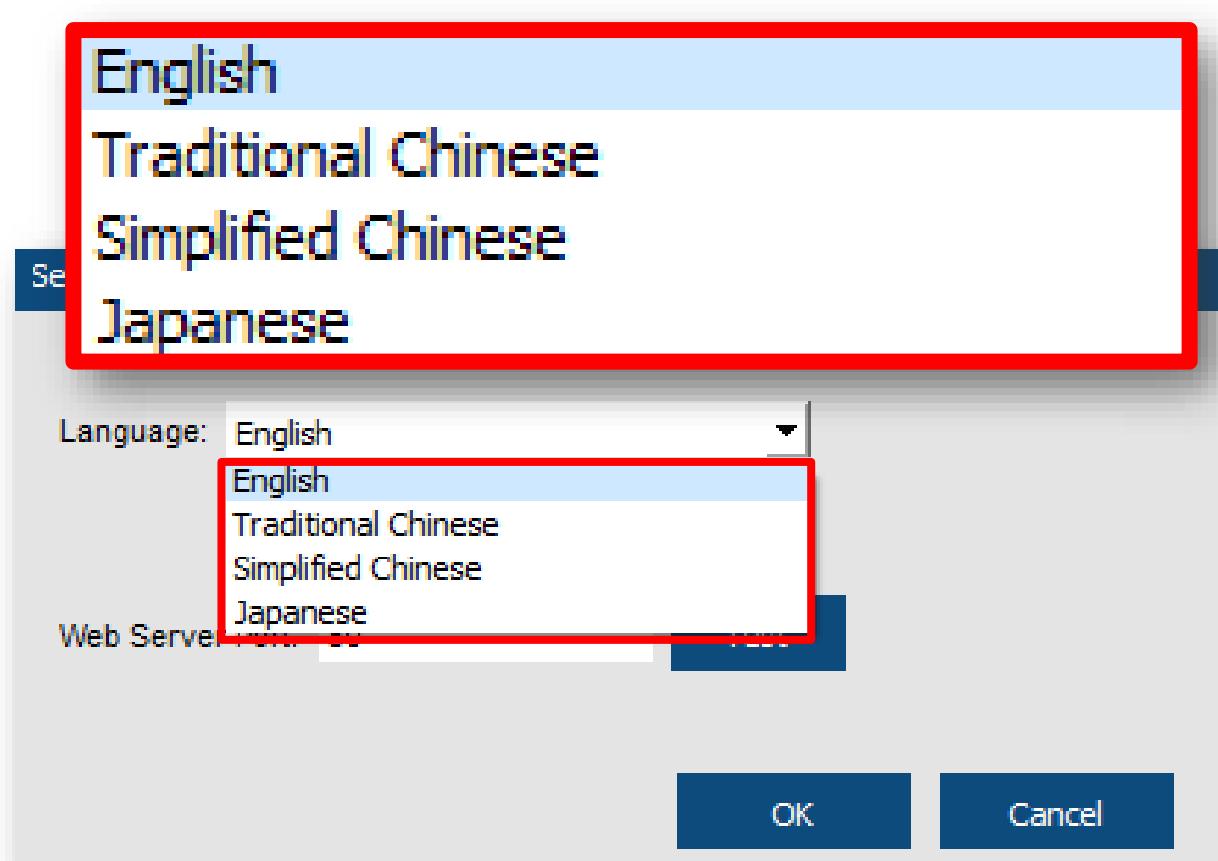
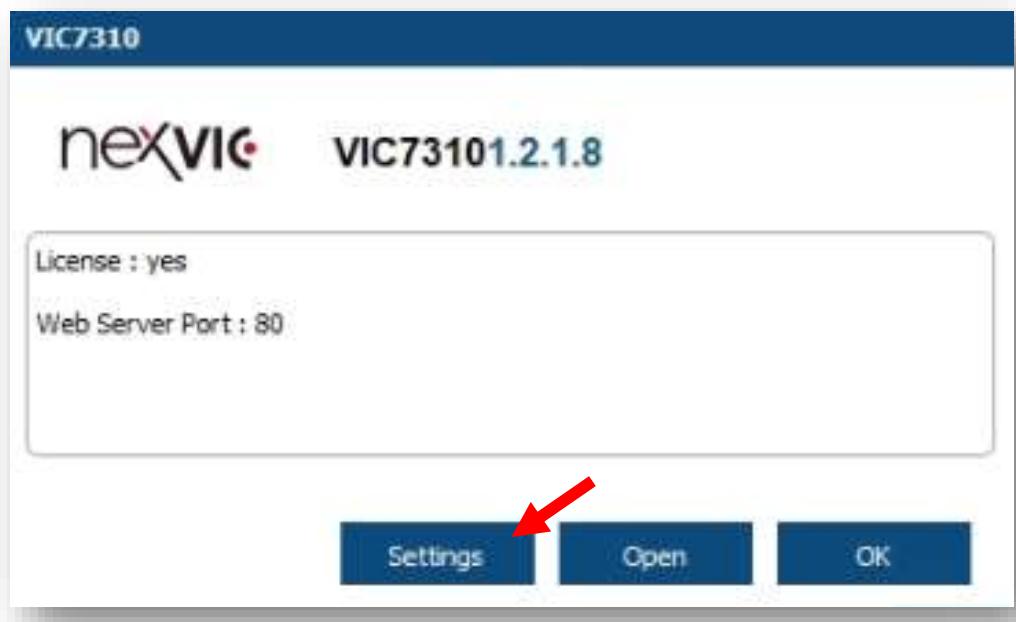
1. 開啟 VIC7000 程式：開機後過一陣子，VIC7000 就會自動啟動



# 軟體設定

- 語言設定

點擊設定，可以設定軟體語言

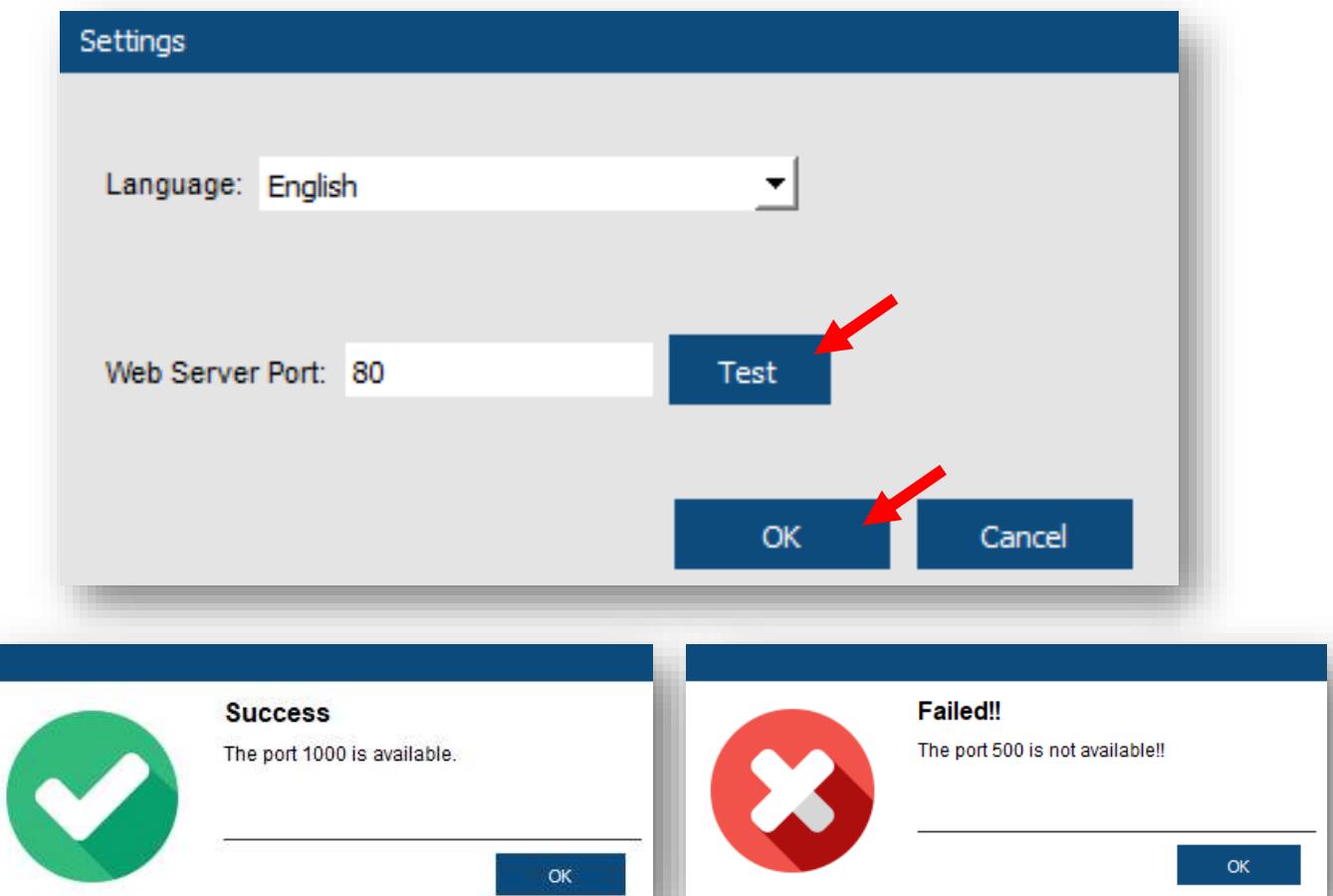
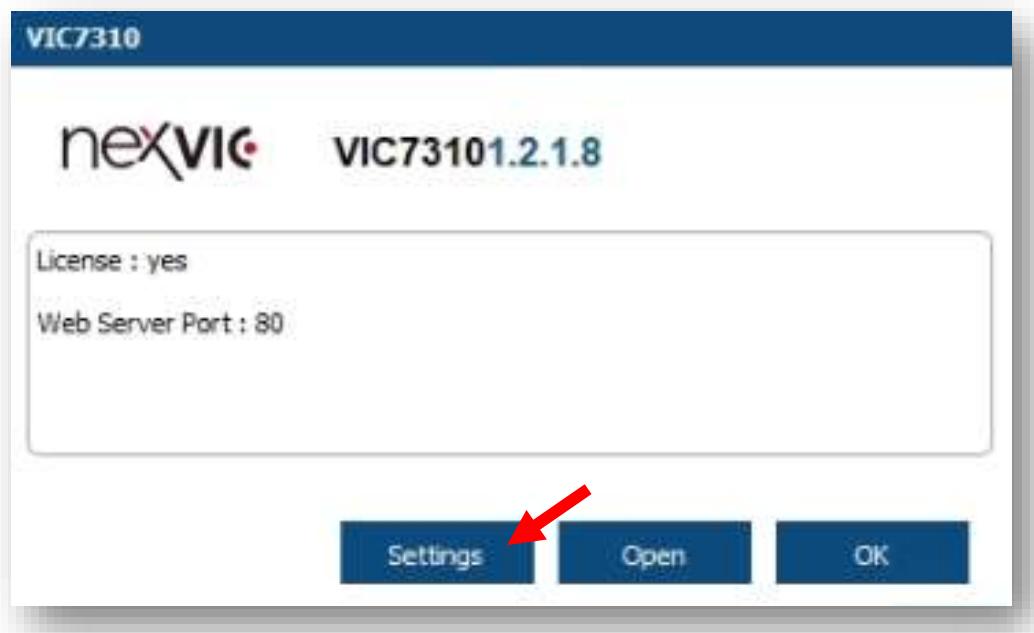


# 軟體設定

## • 連接埠設定

點擊設定，可以設定連接通訊埠，預設為 80

更改前須測試設定的通訊埠沒有被使用，確認通訊埠可用後點擊確認



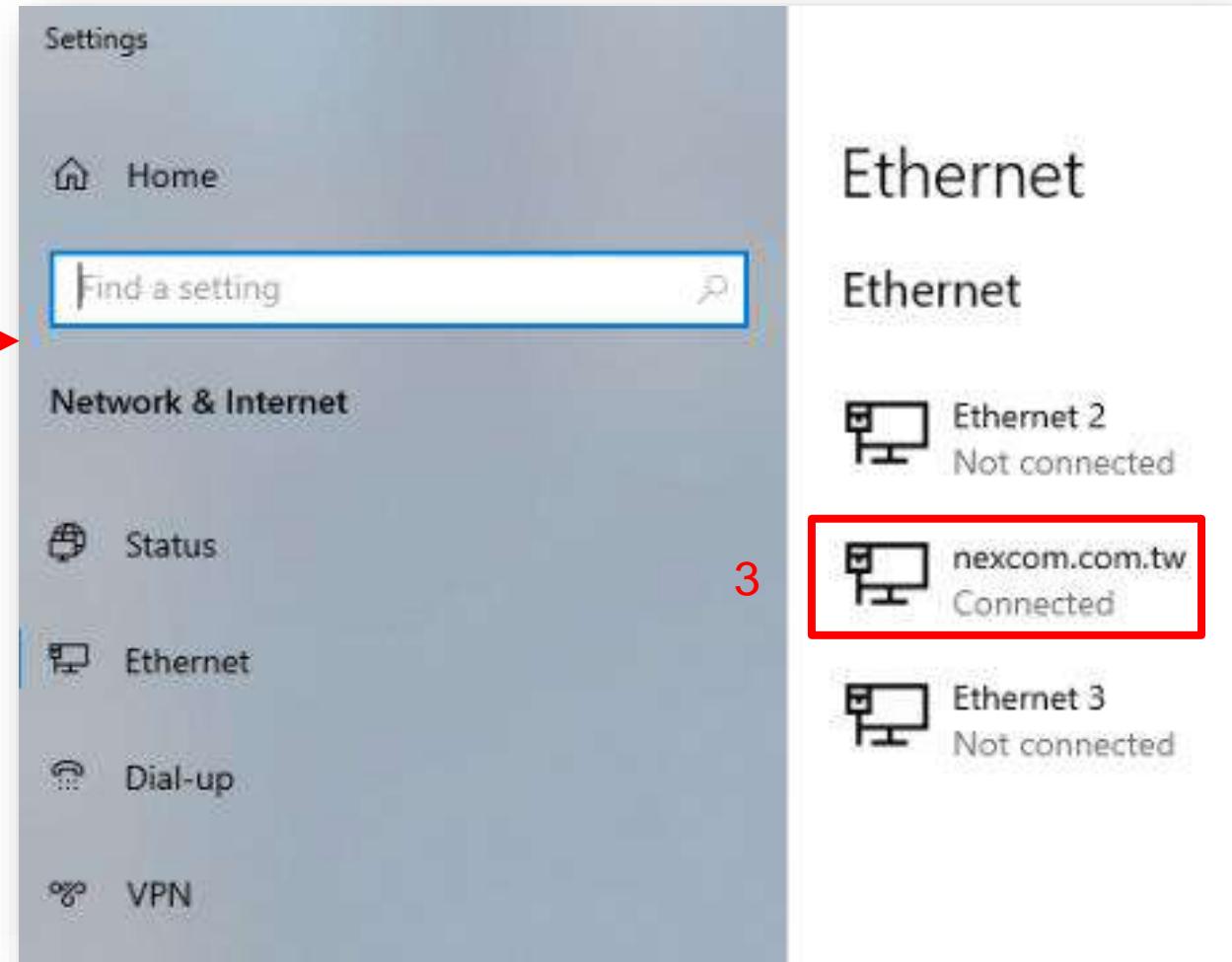
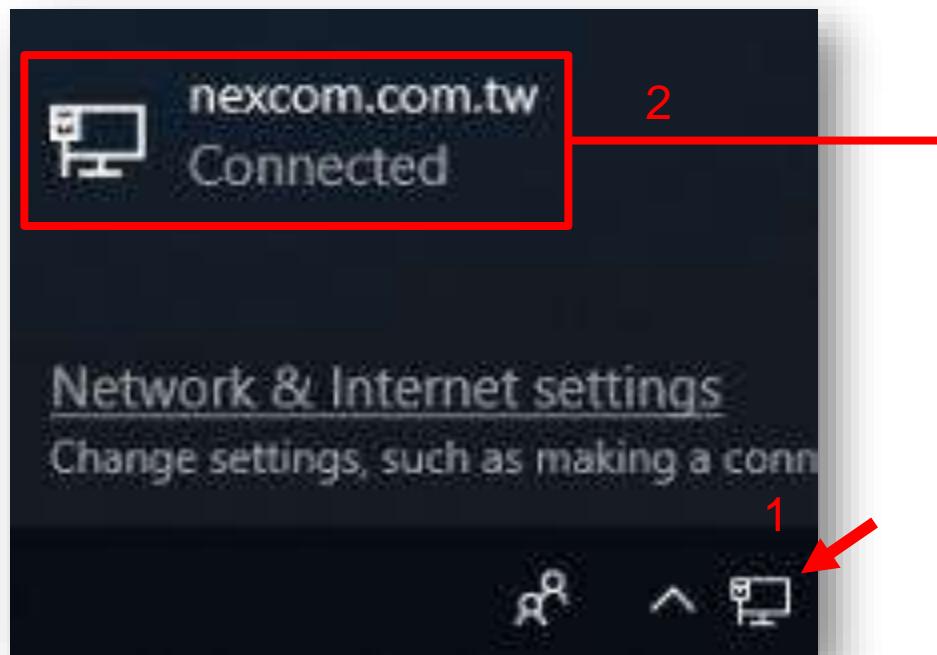
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# IP 設定

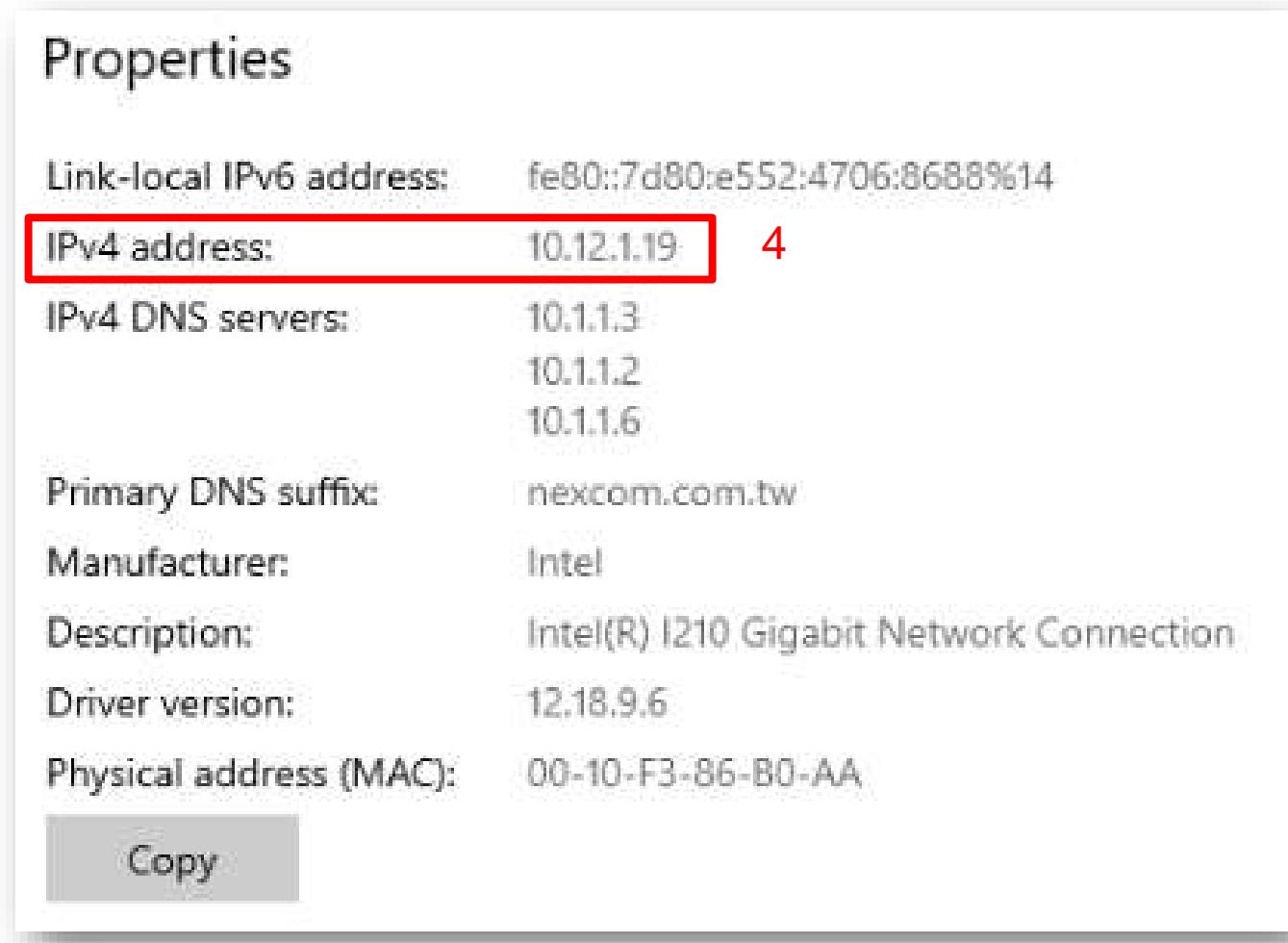
- 取得 VIC IP

進入乙太網路連線的狀態頁



- 取得 VIC IP

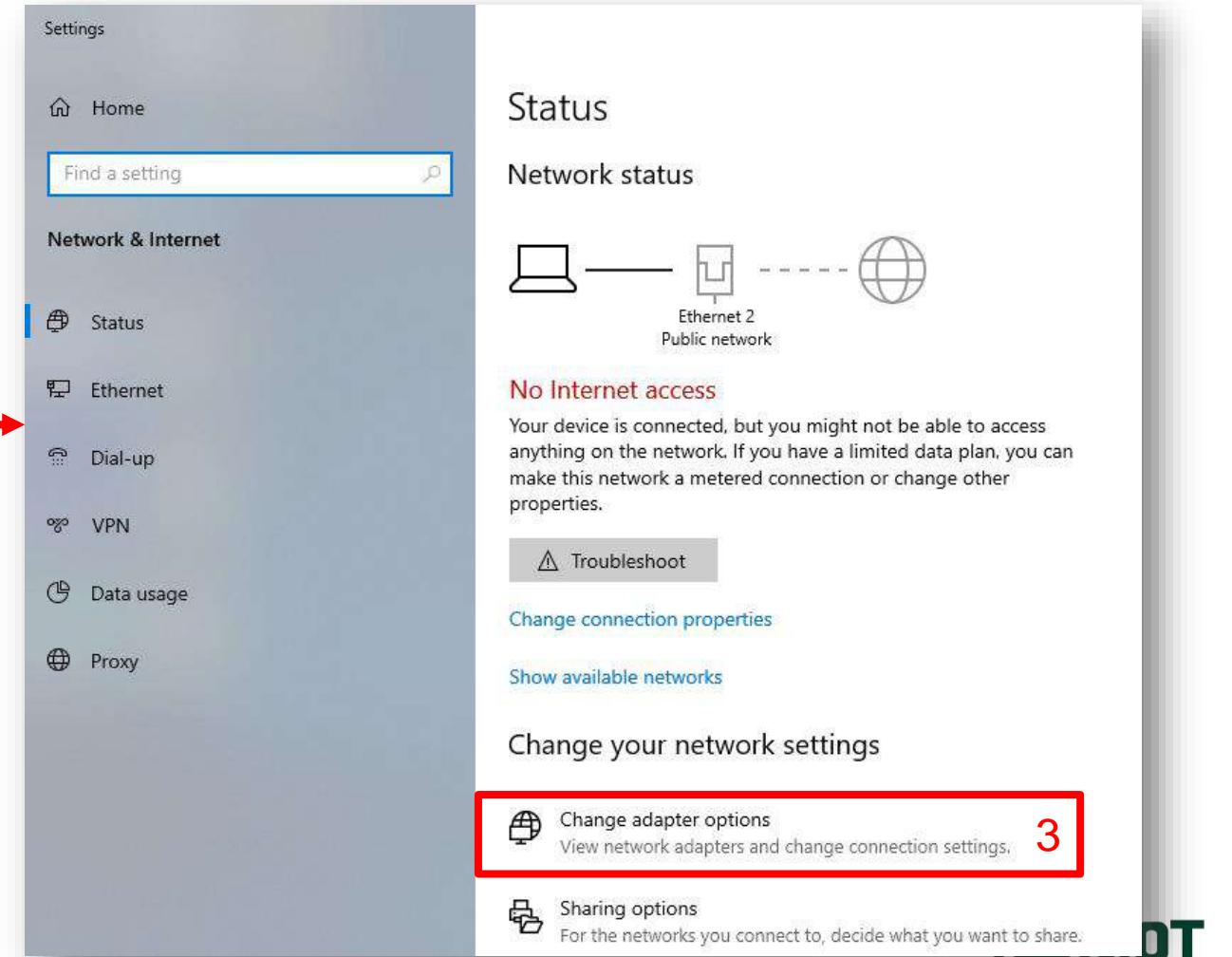
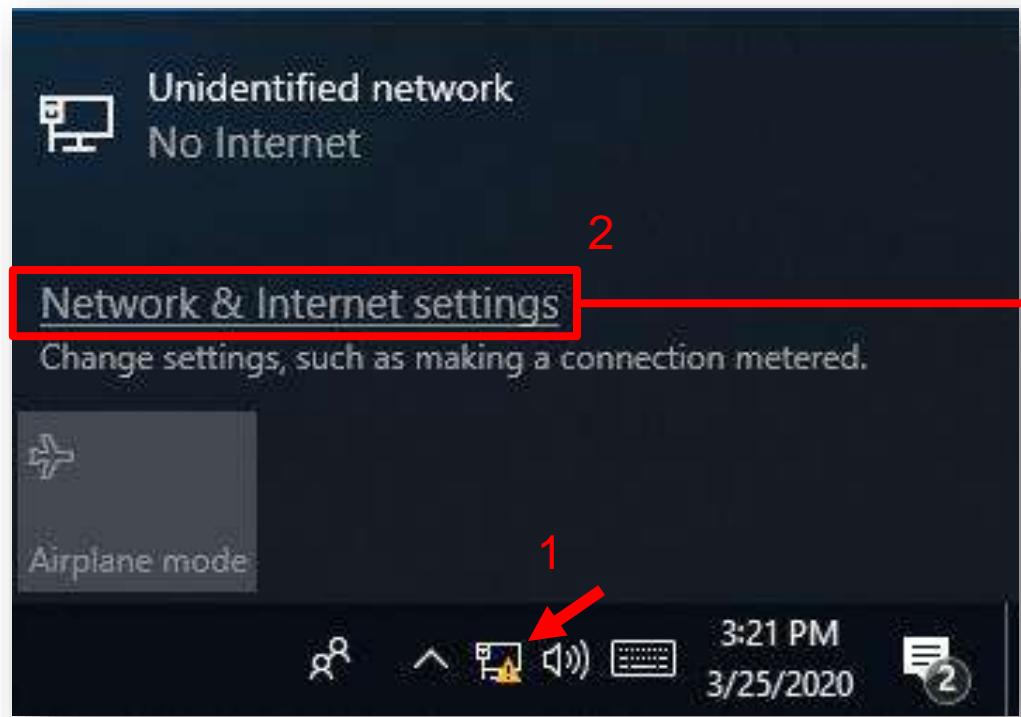
往下看到內容，IPv4 位址即為 VIC7000 的 IP 位址



# IP 設定

## • 設定 VIC IP

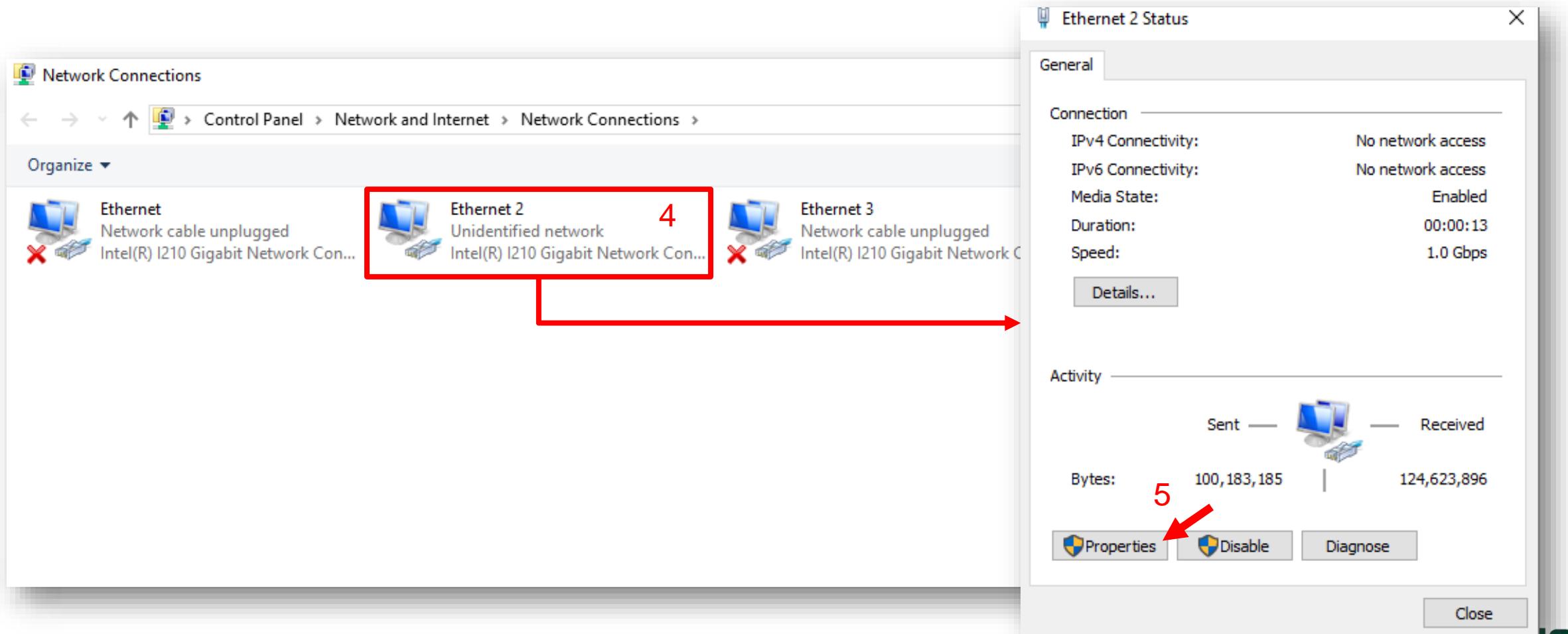
點擊網路和網際網路設定，會出現設定視窗，選擇變更介面卡選項



# IP 設定

## • 設定 VIC IP

點擊要使用的網卡兩下，會出現該網卡的狀態頁，點擊內容

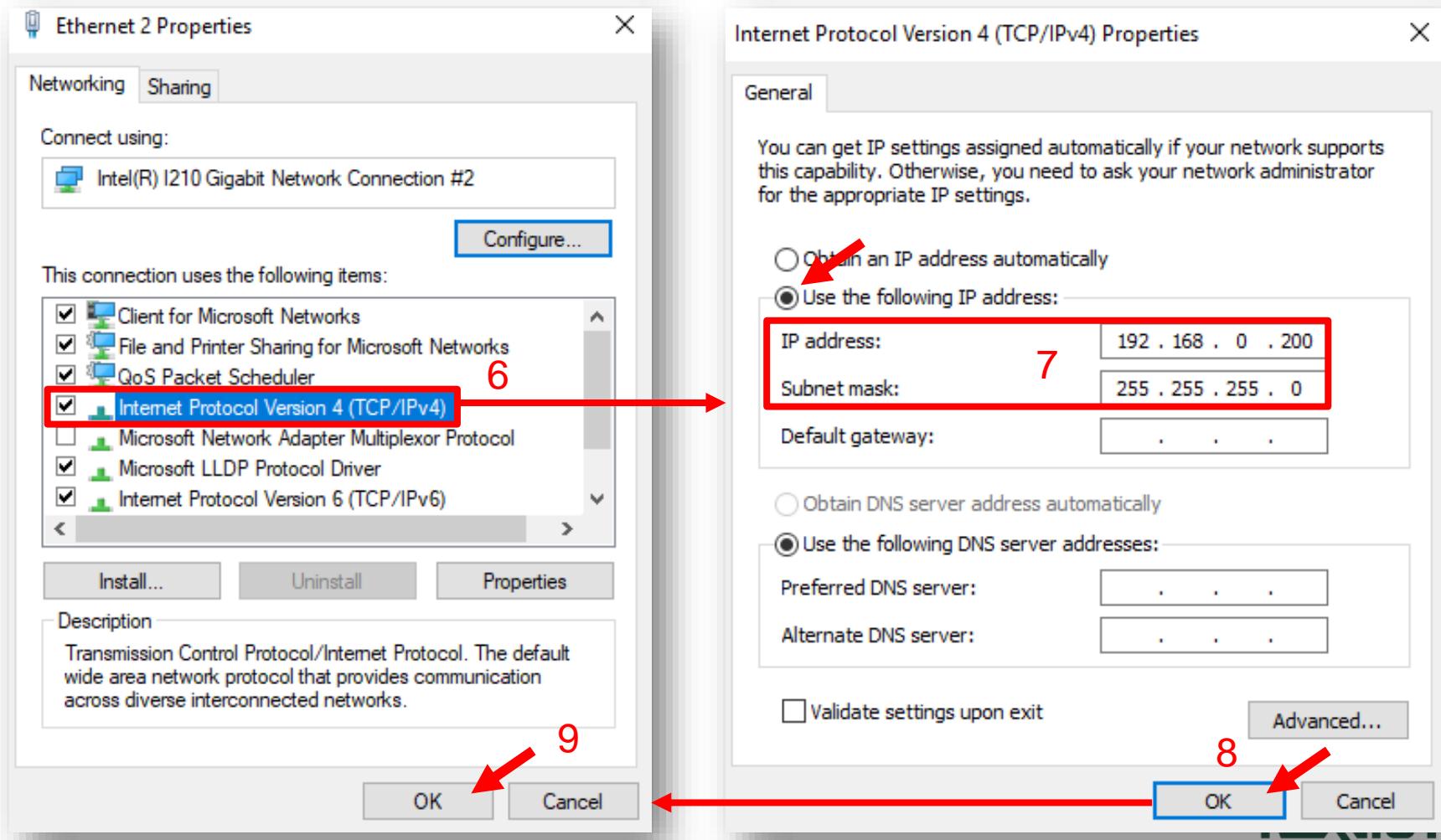


# IP 設定

## • 設定 VIC IP

點擊 網路網路通訊協定第4版(TCP/IPv4) 兩下，設定固定IP和子網路遮罩，點擊確定

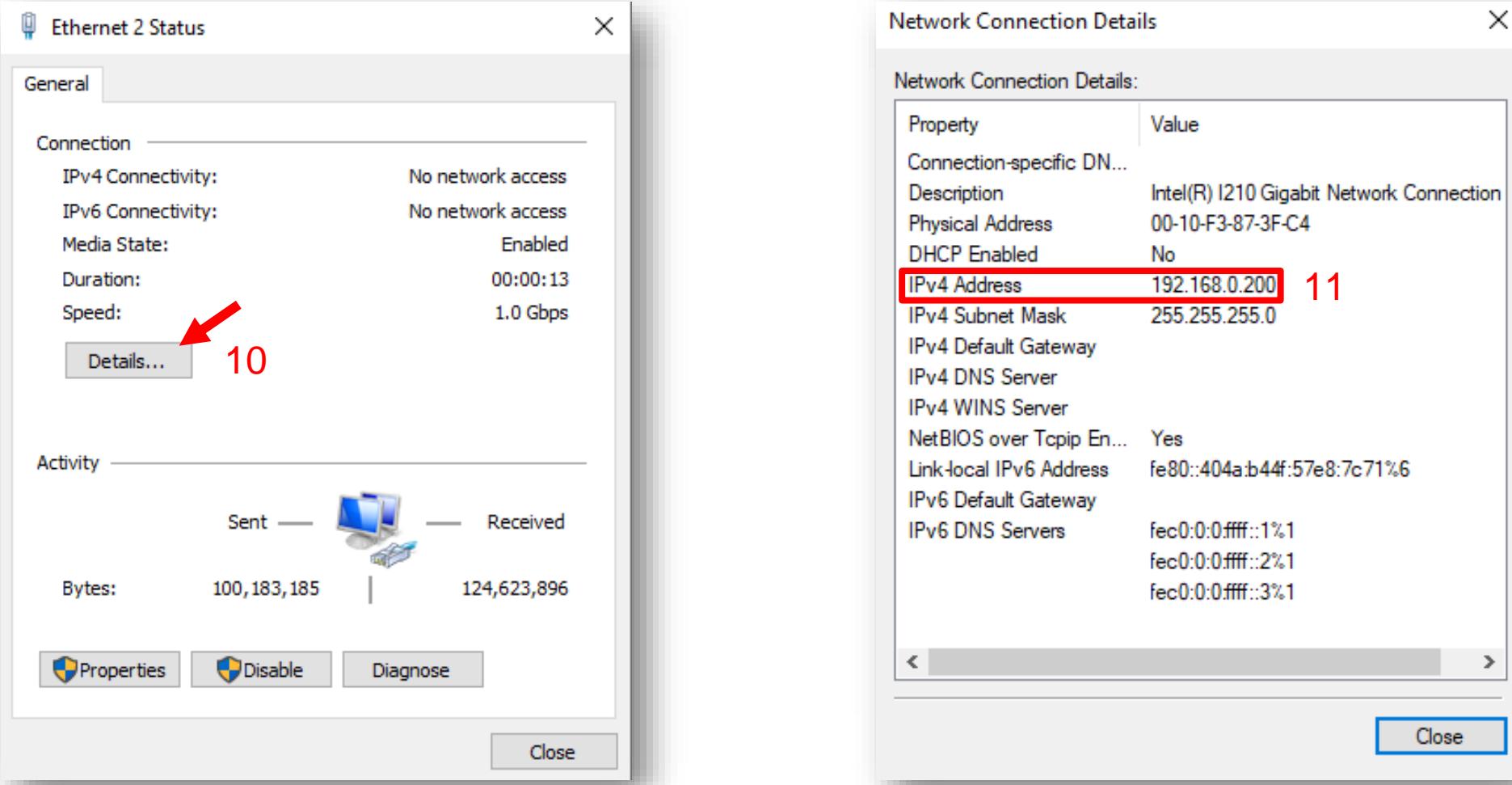
IP 位址必須和要連線的電腦 IP 位址  
在同一網域內  
簡單來說，IP 位址的前三碼必須相同  
範例中，要連線的電腦 IP 位址必須為  
192.168.0.n  
n 為 1~255 且不可和同網域中其他設  
備相同



# IP 設定

## • 設定 VIC IP

點擊狀態頁的詳細資料，可以看到設定好的固定 IP 位址



**Ethernet 2 Status**

**General**

**Connection**

- IPv4 Connectivity: No network access
- IPv6 Connectivity: No network access
- Media State: Enabled
- Duration: 00:00:13
- Speed: 1.0 Gbps

**Activity**

Sent —  — Received

Bytes: 100,183,185 | 124,623,896

**Properties** **Disable** **Diagnose** **Close**

**Network Connection Details**

**Network Connection Details:**

Property	Value
Connection-specific DN...	
Description	Intel(R) I210 Gigabit Network Connection
Physical Address	00-10-F3-87-3F-C4
DHCP Enabled	No
<b>IPv4 Address</b>	<b>192.168.0.200</b> <b>11</b>
IPv4 Subnet Mask	255.255.255.0
IPv4 Default Gateway	
IPv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes
Link-local IPv6 Address	fe80::404a:b4ff:57e8:7c71%6
IPv6 Default Gateway	
IPv6 DNS Servers	fec0::0:ffff::1%1 fec0::0:ffff::2%1 fec0::0:ffff::3%1

**Close**

# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# 登入

- 進入 VIC7000

1. 開啟瀏覽器：使用 Google Chrome

2. 網址列輸入 **Server IP:通訊埠**：預設不須加埠號



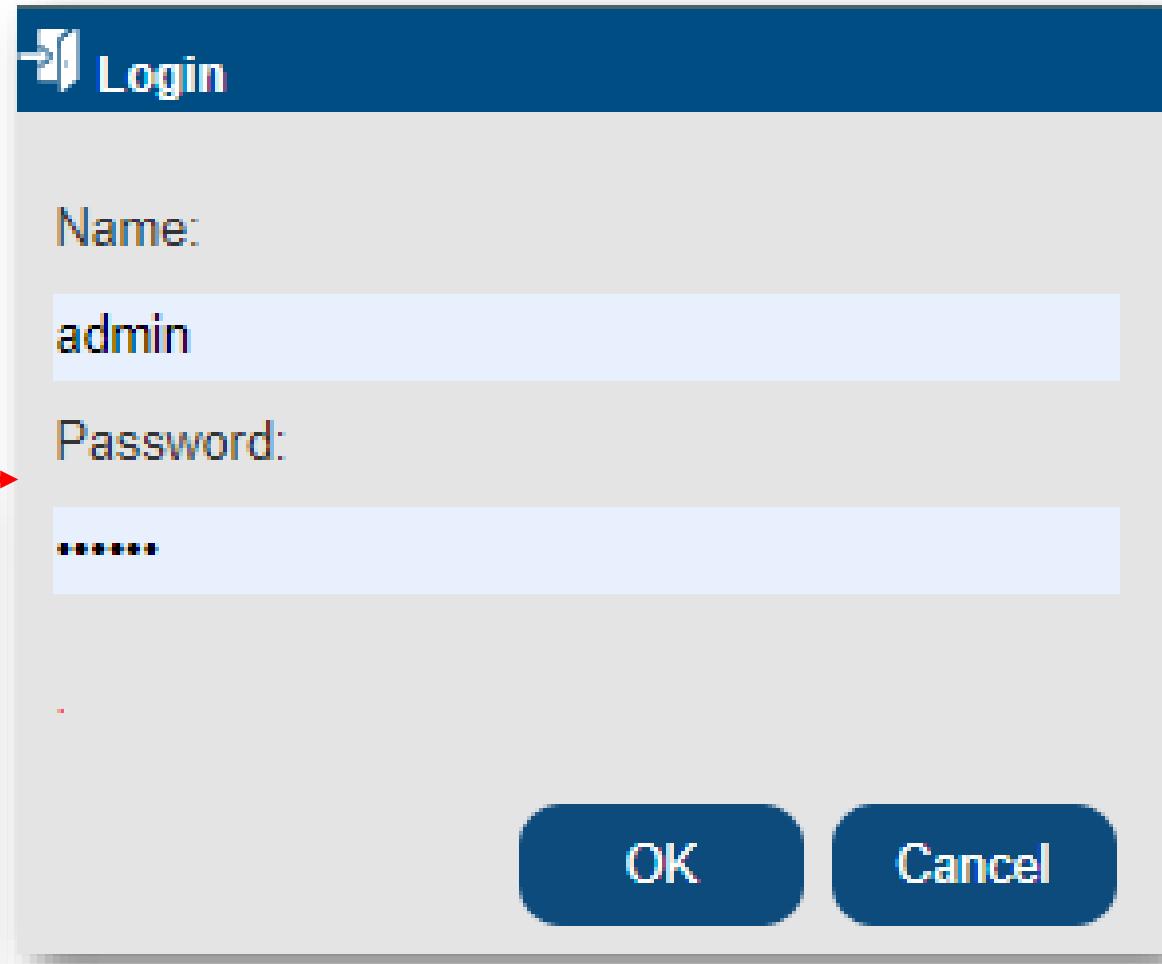
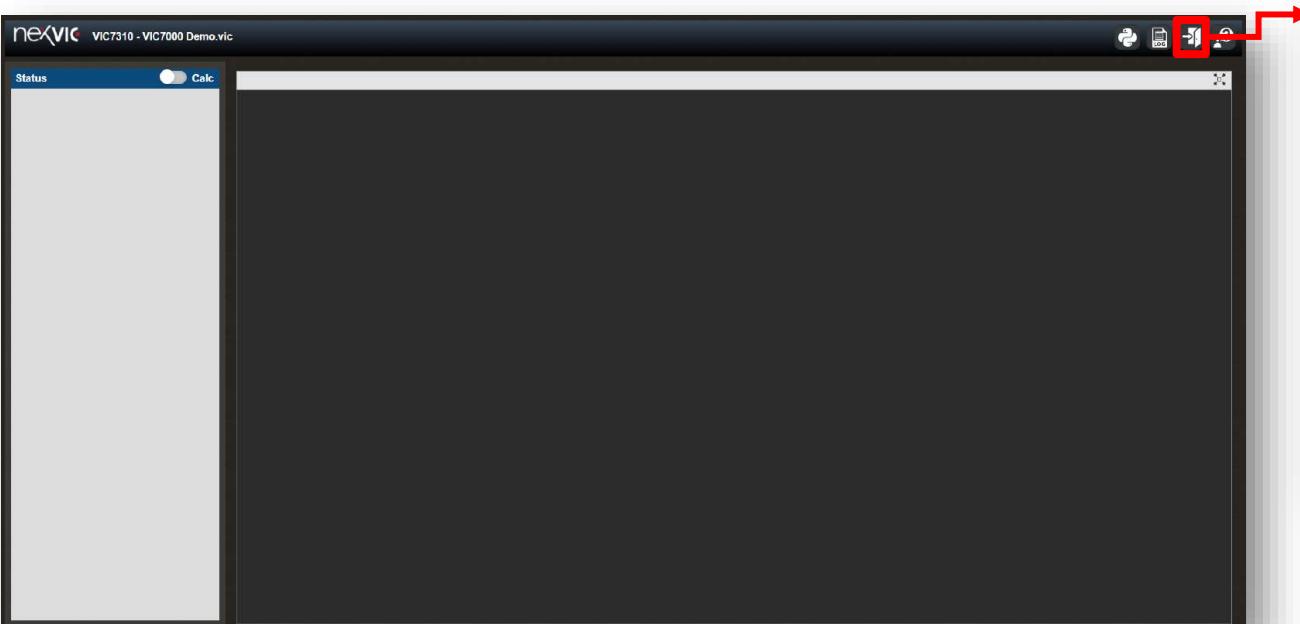
# 登入

- 登入 VIC7000

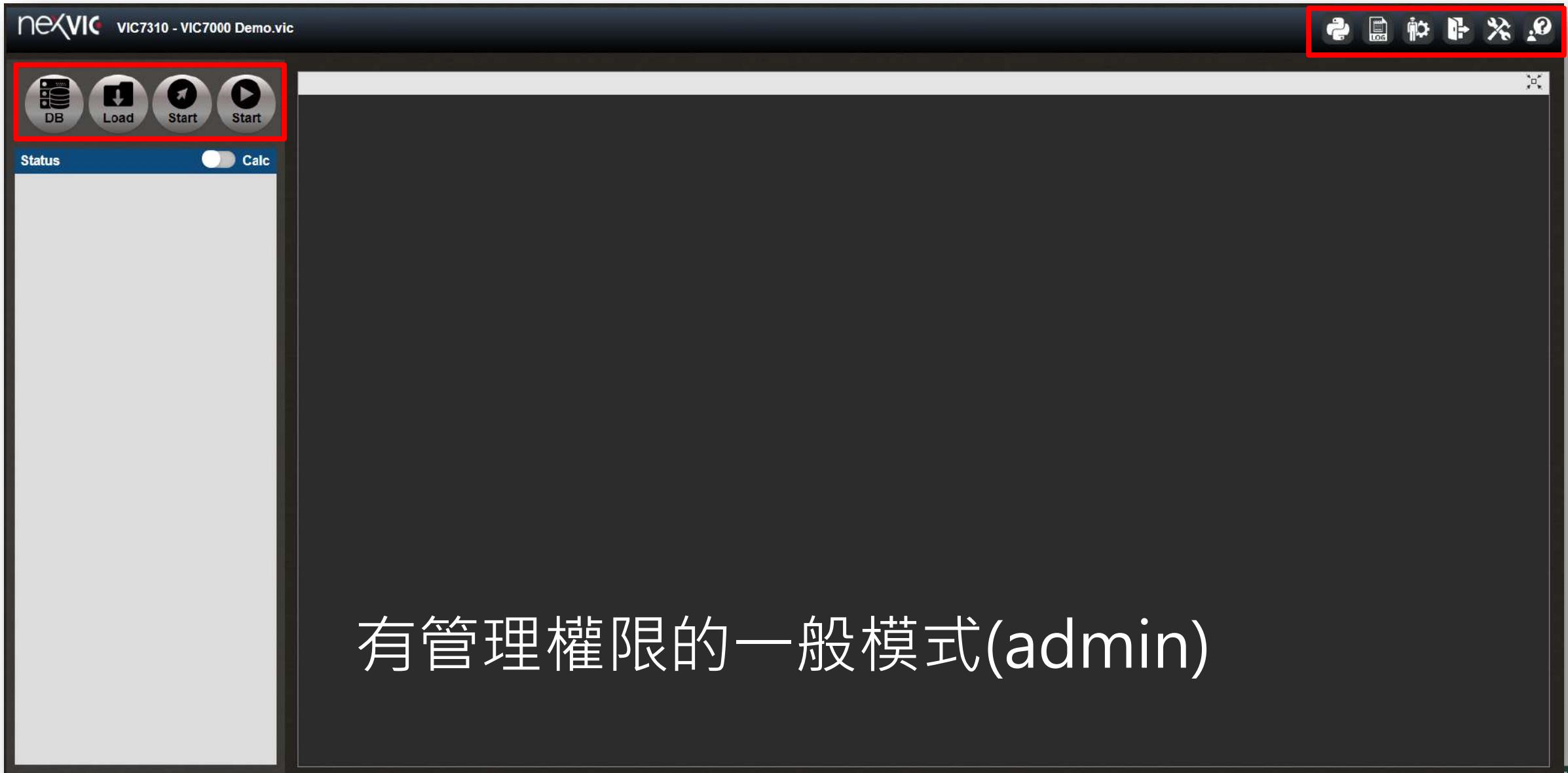
1. 點擊右上的登入按鈕
2. 輸入使用者或管理者名稱和密碼

使用者名稱為 user，管理者名稱為 admin，  
預設密碼皆為 123456

若登入後閒置超過半小時，將被強制登出



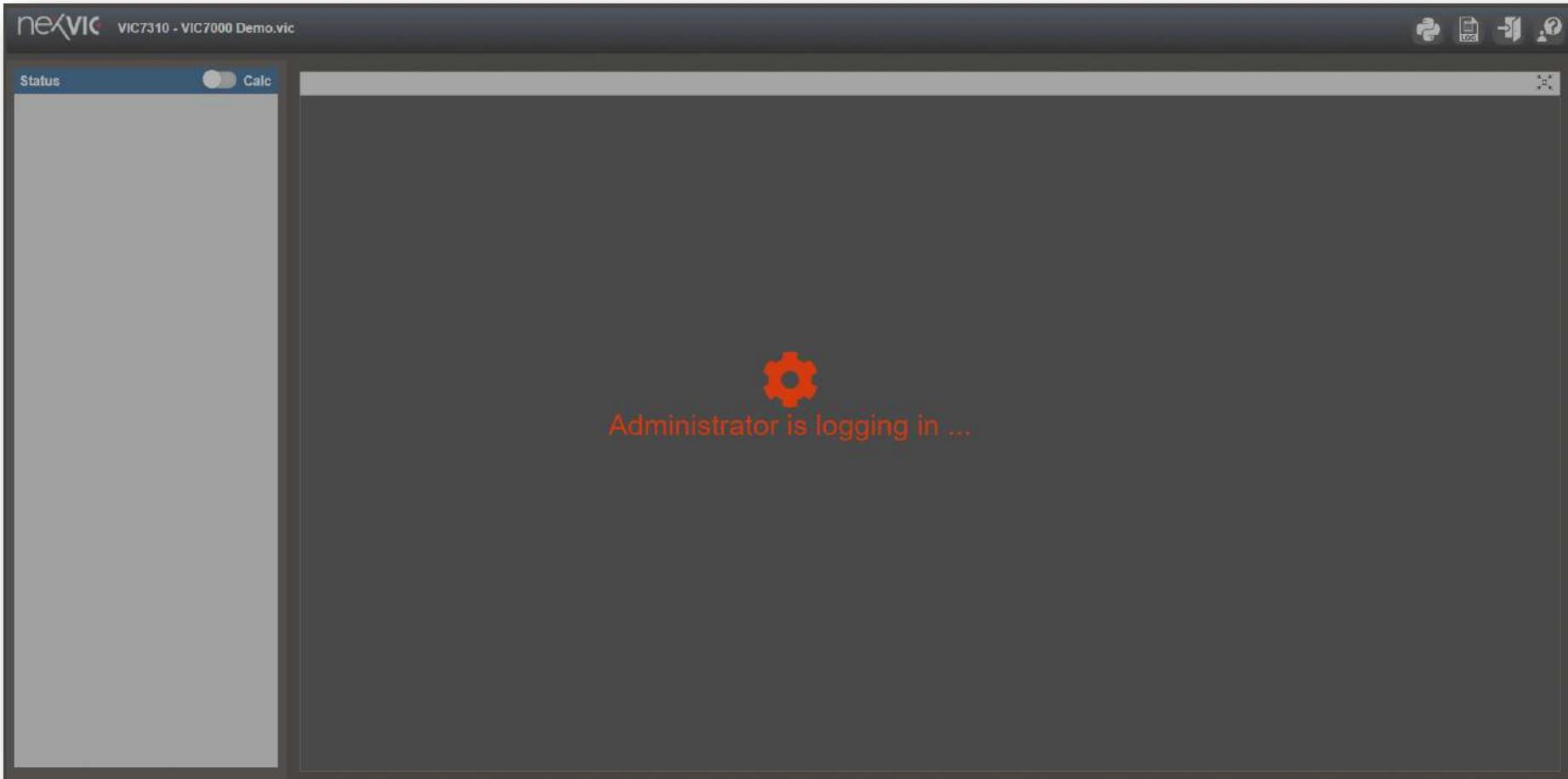




# 登入

- **登入限制**

- 允許最多同時 16 人同時在無管理權限的一般模式監看程式運作狀況
- 僅允許一個管理者登入系統，此時其他使用者無法登入或監看程式

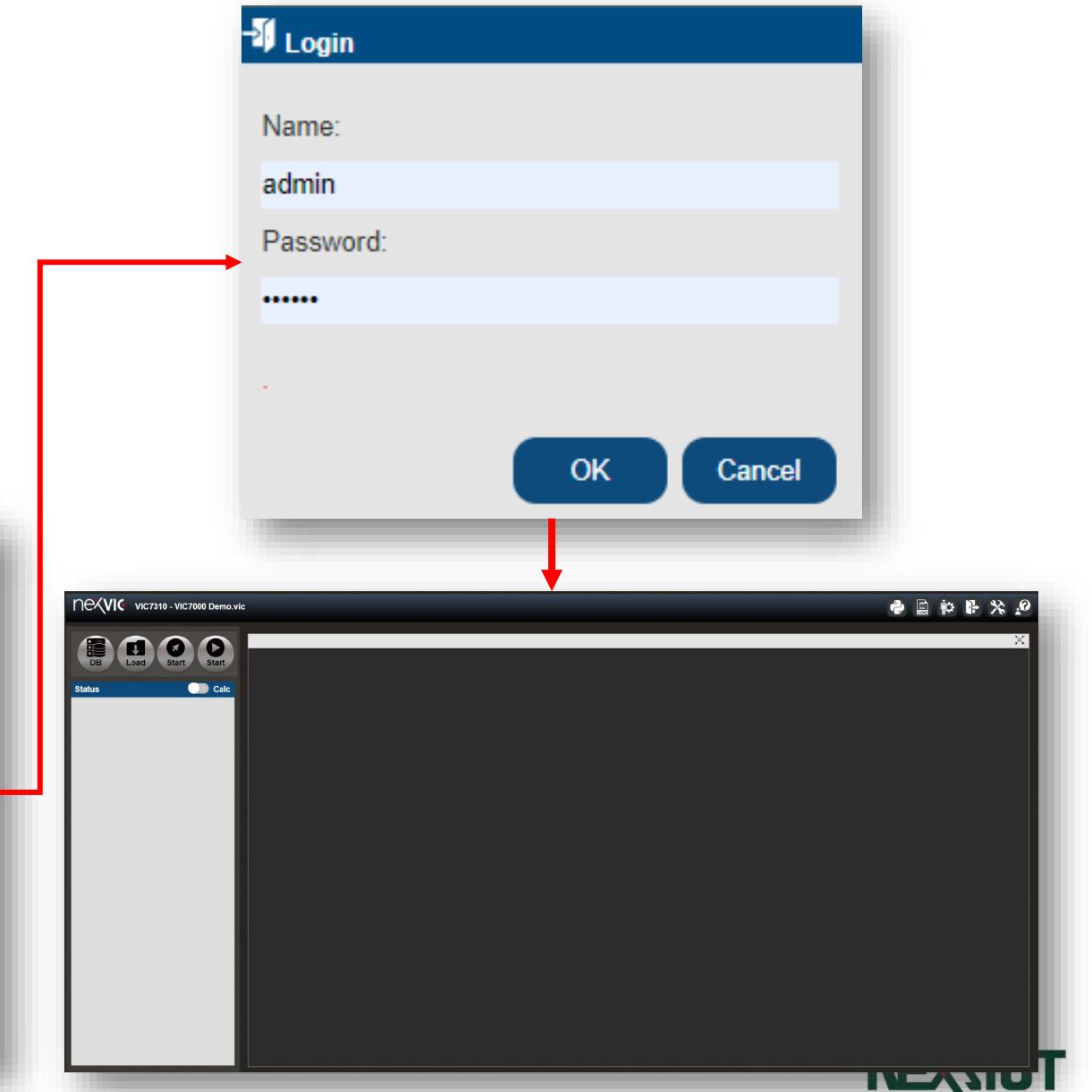
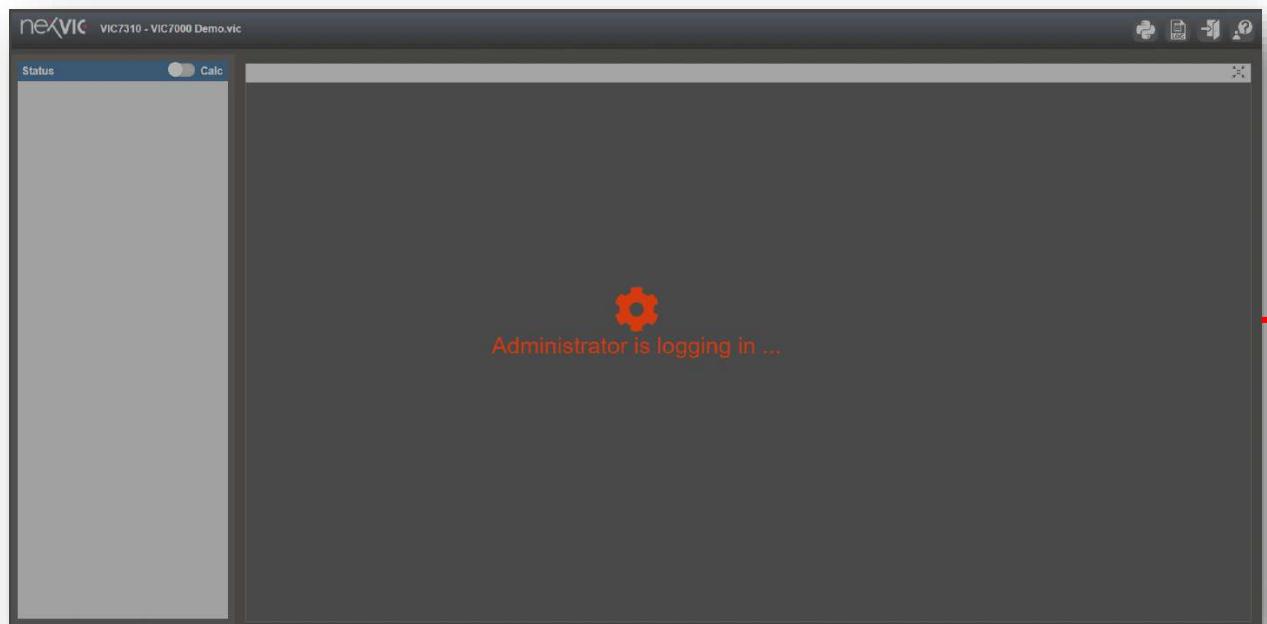


# 登入

- 取代登入

1. 點擊畫面任一處：出現登入視窗
2. 輸入管理者名稱和密碼
3. 點擊確定

登入後，原管理者會被**強制登出**



# 登入

- 頁面按鈕

1. 資料庫
2. 載入專案
3. 啟動/停止控制
4. 啟動/停止專案
5. Python 輸出
6. 系統日誌
7. 管理模式
8. 登出
9. 系統設置
10. 關於



## • 系統日誌

- 記錄軟體重要資訊
- 分類記錄
- 可匯出成 .txt
- 匯出路徑為瀏覽器下載路徑



System Log

Module: All

From: Pick a date To: Pick a date Reset

Module: All

Export Clear Cancel

System Log content (partial):

```

[6/28/2019, 12:37:19] {socket} A Client connected to server.(127.0.0.1, total connections: 2)
[6/28/2019, 12:37:19] {socket} A Client connected to server.(127.0.0.1, total connections: 1)
[6/28/2019, 12:57:16] {file} Acquisition starts
[6/28/2019, 12:57:20] {file} Acquisition stops
[6/28/2019, 13:00:59] {socket} A Client connected to server.(192.168.133.1, total connections: 1)
[6/28/2019, 13:07:17] {file} Acquisition starts
[6/28/2019, 13:07:29] {file} Acquisition stops
[6/28/2019, 13:07:47] {file} Acquisition starts
[6/28/2019, 13:08:20] {file} Acquisition stops
[6/28/2019, 13:10:48] {file} Acquisition starts
[6/28/2019, 13:23:10] {file} Acquisition stops
[6/28/2019, 13:23:32] {file} Acquisition starts
[6/28/2019, 13:23:56] {modbus} A modbus connected to server.(192.168.133.1, total connections: 1)
[6/28/2019, 13:24:04] {file} Acquisition stops

```

另存新檔

本機 下載 搜尋 下載

新增資料夾

名稱: log.Wed Jun 26 2019 16\_03\_01 GMT+0800 (台北標準時間).txt

修改日期: 2017/1/10 下午 0

類型: 編輯資料夾

組合管理

快速存取

桌面 下載 (選中)

文件 圖片 pserver DATA (D) Monitor Rec Application Sto Meeting

檔案名稱(N): log.Wed Jun 26 2019 16\_03\_01 GMT+0800 (台北標準時間).txt

存檔類型(T): Text Document (\*.txt)

存檔(S) 取消

log.Wed Jun 26 2019 16\_03\_01 GMT+0800 (台北標準時間).txt - 記事本

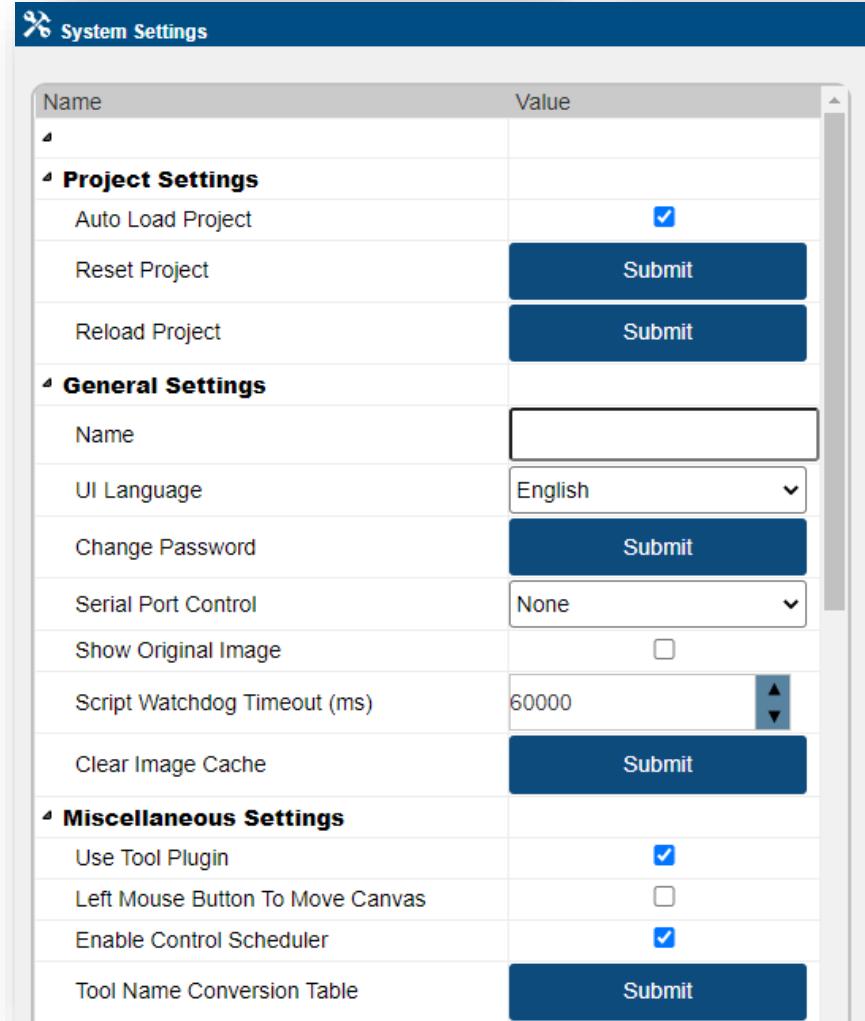
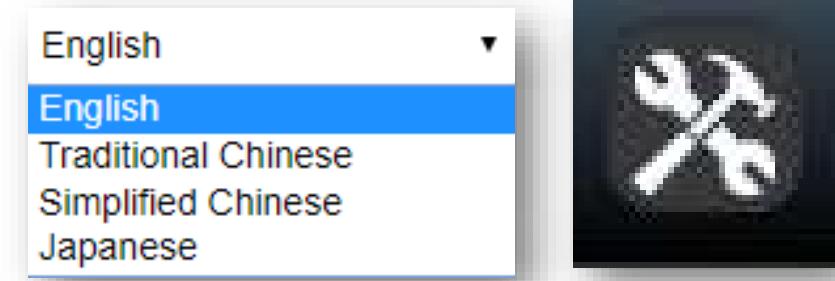
```

<>[2019/6/27 上午2:01:01]{webserver}WebServer start at port 80.
<>[2019/6/27 上午2:01:53]{system}System quits
<>[2019/6/27 上午2:02:03]{webserver}WebServer start at port 80.
<>[2019/6/27 上午2:02:49]{socket}A Client connected to server.(192.168.133.1, total connections: 1)
<>[2019/6/27 上午2:02:59]{ajax}Admin signed in.(192.168.133.1)
<>[2019/6/27 上午2:03:03]{project}Load project.(VIC7000 Demo.vic)
<>[2019/6/27 上午2:03:03]{project}Load project successfully.(VIC7000 Demo.vic)
<>[2019/6/27 上午2:03:04]{socket}A Client disconnected.(192.168.133.1, total connections: 0)
<>[2019/6/27 上午2:03:05]{socket}A Client connected to server.(192.168.133.1, total connections: 1)
<>[2019/6/27 上午2:03:06]{file}Acquisition starts
<>[2019/6/27 上午2:03:13]{file}Acquisition stops
<>[2019/6/27 上午2:05:45]{file}Acquisition starts
<>[2019/6/27 上午2:12:20]{file}Acquisition stops
<>[2019/6/27 上午3:09:14]{ajax}time out to force signing out
<>[2019/6/27 上午3:19:48]{ajax}Admin signed in.(192.168.133.1)
<>[2019/6/27 上午3:19:49]{file}Acquisition starts
<>[2019/6/27 上午3:50:43]{ajax}time out to force signing out
<>[2019/6/27 上午4:35:43]{ajax}Admin signed in.(192.168.133.1)
<>[2019/6/27 上午4:36:03]{socket}A Client connected to server.(192.168.133.1, total connections: 2)
<>[2019/6/27 上午4:41:26]{file}Acquisition stops
<>[2019/6/27 上午4:41:28]{project}Load project.(VIC7000 Demo.vic)
<>[2019/6/27 上午4:41:29]{project}Load project successfully.(VIC7000 Demo.vic)
<>[2019/6/27 上午4:41:29]{socket}A Client disconnected.(192.168.133.1, total connections: 1)
<>[2019/6/27 上午4:41:30]{socket}A Client disconnected.(192.168.133.1, total connections: 0)
<>[2019/6/27 上午4:41:31]{socket}A Client connected to server.(192.168.133.1, total connections: 1)
<>[2019/6/27 上午4:41:31]{socket}A Client connected to server.(192.168.133.1, total connections: 2)

```

- 系統設置 – 專案 & 一般 & 其他

- 自動載入專案
- 重置專案
- 重新載入專案
- 名稱
- 介面語言 – 英文、繁中、簡中、日語
- 改變密碼
- 序列埠控制
- 顯示原圖
- 腳本超時中斷時間(ms)
- 清除圖檔緩存
- 使用工具插件
- 使用滑鼠左鍵移動畫布
- 啟動控制排程
- 工具命名對照圖



- 系統設置 – 系統 & Flow設置 & 通訊埠

- 系統載入
- 系統儲存
- 系統重新啟動
- 顯示處理中的運算子
- 全部運算子的延遲時間
- UDP通訊埠
- UDP群組通訊埠
- TCP資料通訊埠
- TCP指令通訊埠



System Settings

**System**

- System Load
- System Save
- System Restart

**Flow Settings**

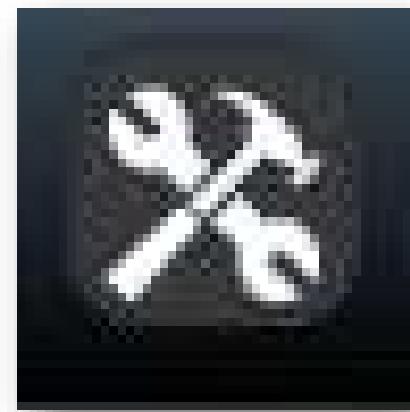
- Show Processing Operators
- Delay Time For All Operators (ms)

**Communication Port**

Port Type	Port Number	Action
UDP Port	7001	Setting
UDP Multicast Group Port	7002	Setting
TCP Data Port	7003	Setting
TCP Command Port	7004	Setting

- 系統設置 – 資料庫

- 儲存圖檔
- 儲存不重複資料
- 目標值數量
- 目標值(顏色)數量
- 目標值(樣式比對)數量
- OCR 數量
- 顏色數量
- 樣式比對數量
- 運算數量
- 汇出路徑
- 選擇匯出路徑
- 備份路徑
- 選擇備份路徑
- 備份時間(時)



System Settings

Database Settings

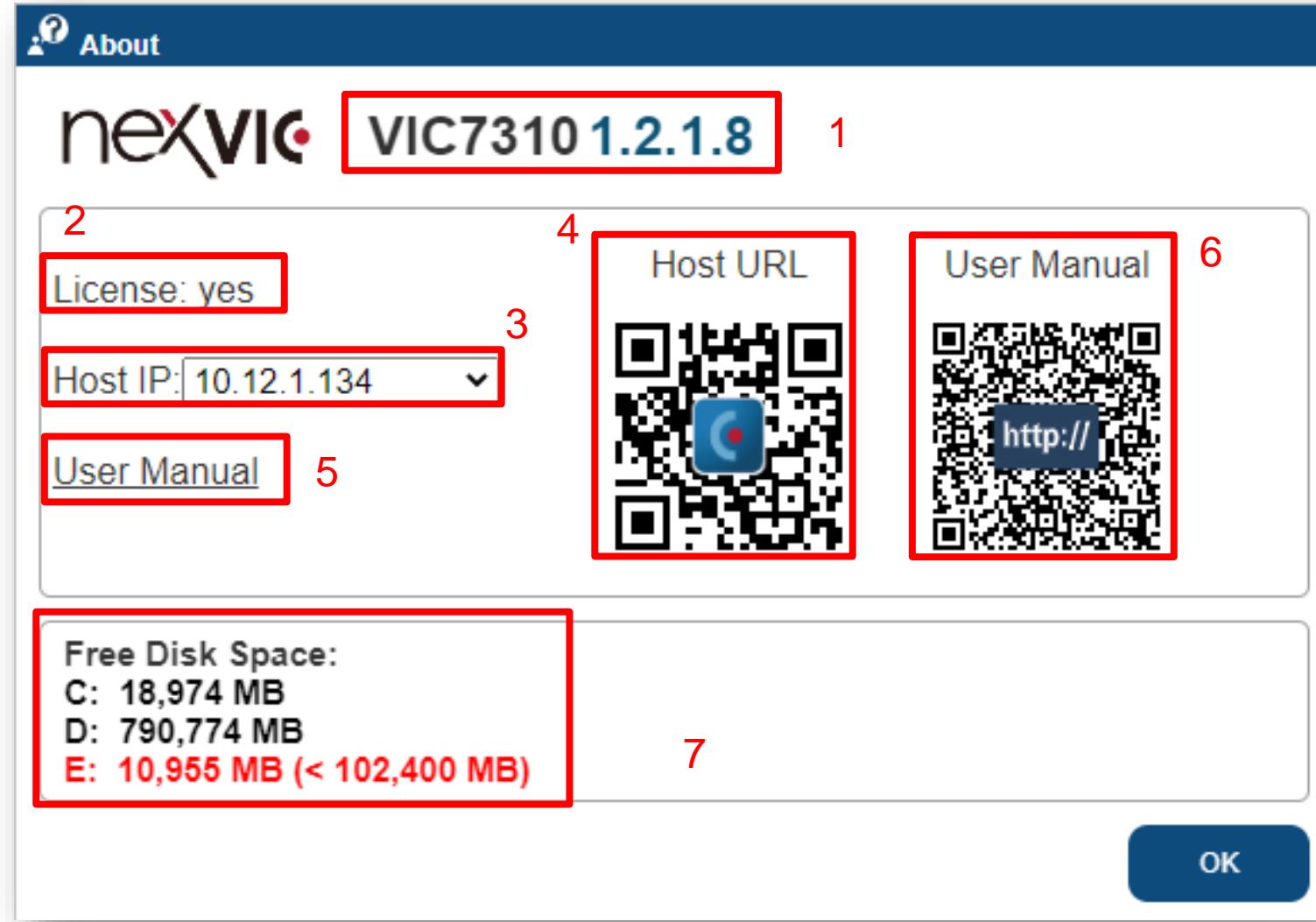
Save Image	<input type="checkbox"/>
Skip Repeated Data	<input type="checkbox"/>
Target Quantity	5
Target_color Quantity	5
Target_pattern Quantity	5
OCR Quantity	20
Color Quantity	20
Pattern Quantity	10
Calc Quantity	0
Export Path	D:/database/temp/
Select Export Path	<input type="button" value="Submit"/>
Backup Path	D:/database/backup/
Select Backup Path	<input type="button" value="Submit"/>
Backup time(hour)	0

Cancel

# 登入

## • 系統資訊

1. 版本資訊
2. 軟體 License 檢查
3. VIC7000 主機 IP 選擇欄位
4. VIC7000 主機 IP QR code
5. 本地使用手冊連結
6. 線上使用手冊 QR code
7. VIC7000 主機磁碟可用空間



## 管理模式



- 頁面按鈕

1. 新增專案
2. 載入專案
3. 儲存專案
4. 另存專案

5. 頁面
6. 腳本頁面
7. 監視頁面
8. 控制頁面
9. 連結頁面
10. SECS頁面

11. 精靈模式

1      2      3      4      5      6      7      8      9      10      11



# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

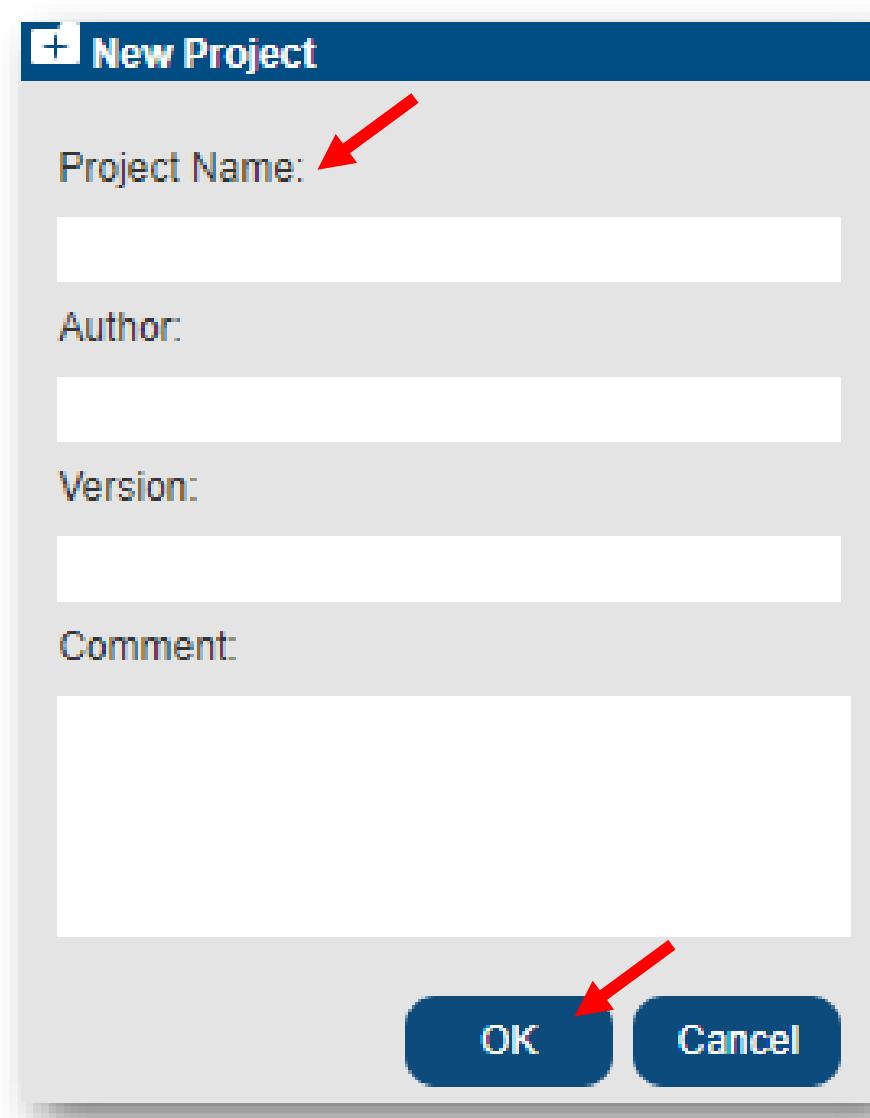
# 新增專案

- 新增專案

1. 輸入專案名稱 (必填，不可和已存在專案同名)

2. 點擊確定

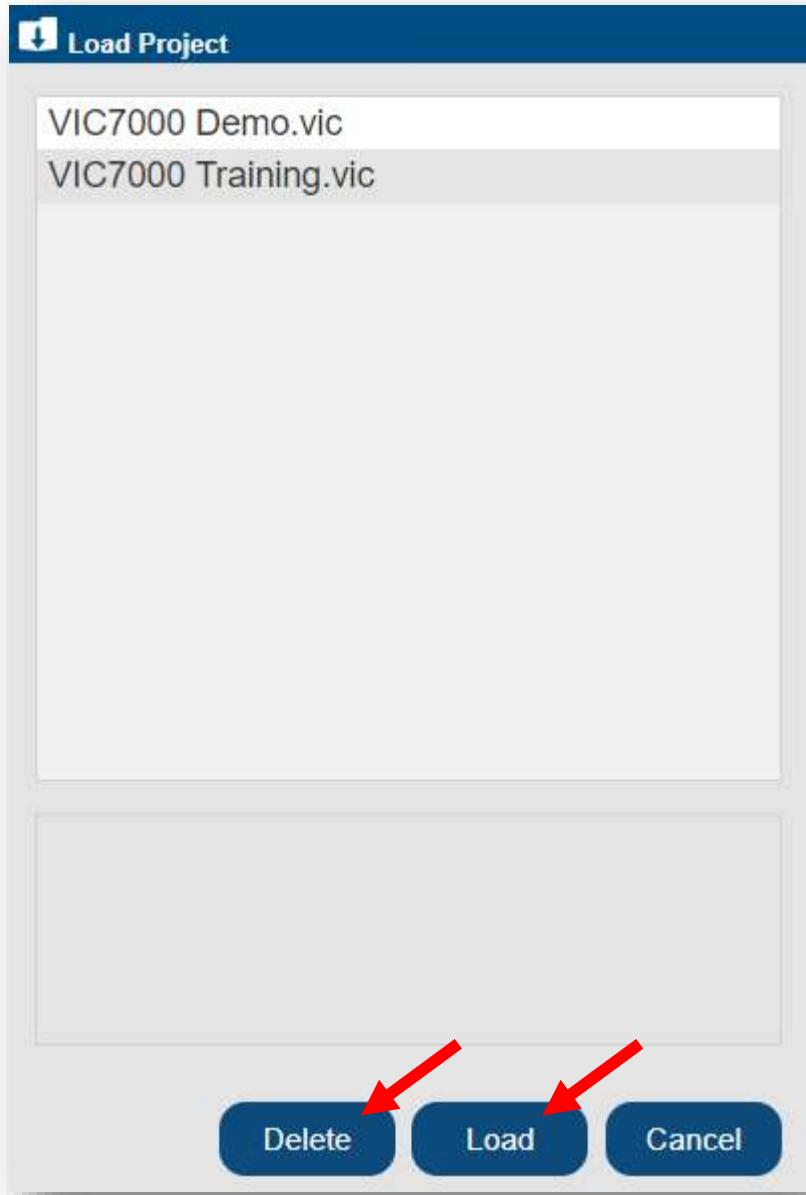
- 作者
- 版本
- 備註



# 新增專案

- 載入專案

- 載入專案
- 刪除專案



# 新增專案

- 頁面



01

Add Page

Tool

Data Source

From File

Settings

Name	Value
Loop	<input checked="" type="checkbox"/>
Sort By	Name
Select Image Folder	<input type="button" value="Submit"/>
Image Folder	C:/VIC7000/Demo_imag
Interval (ms)	1000

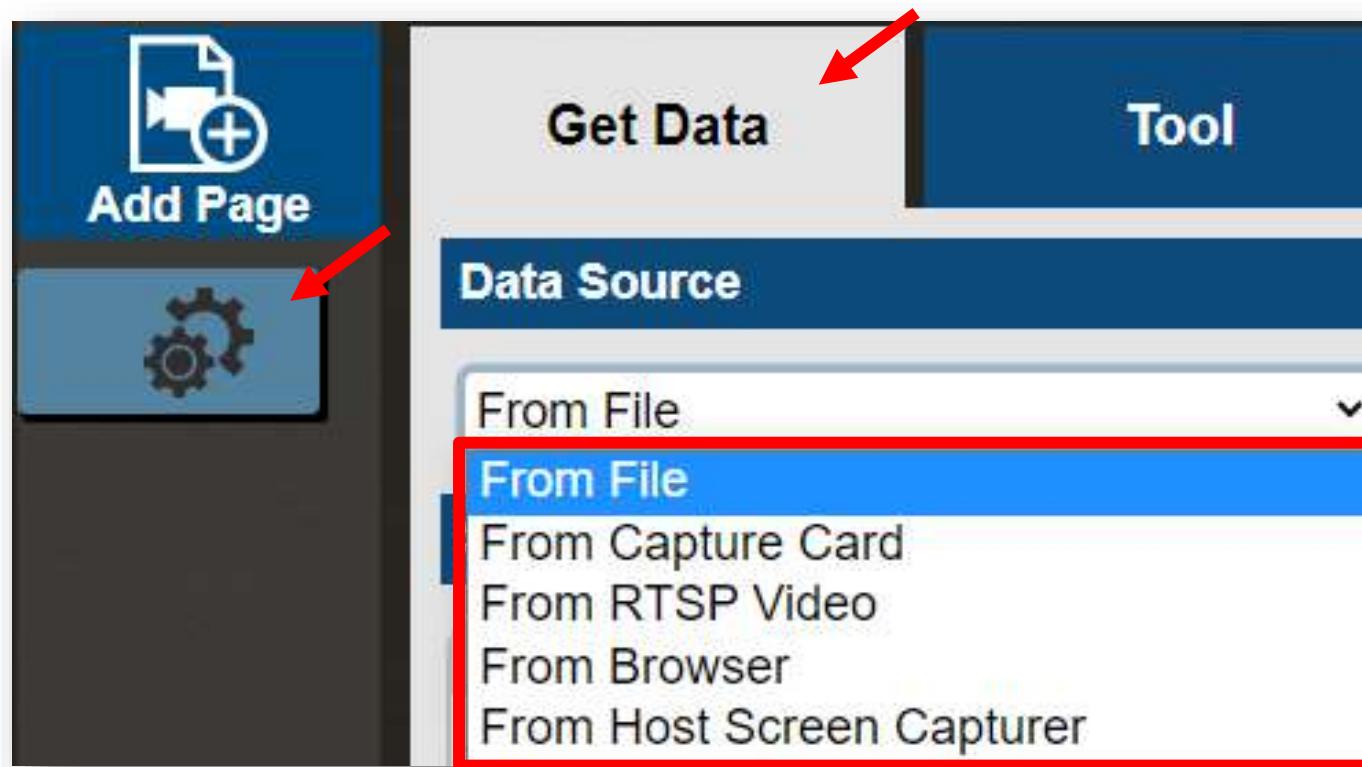
Get Image

# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

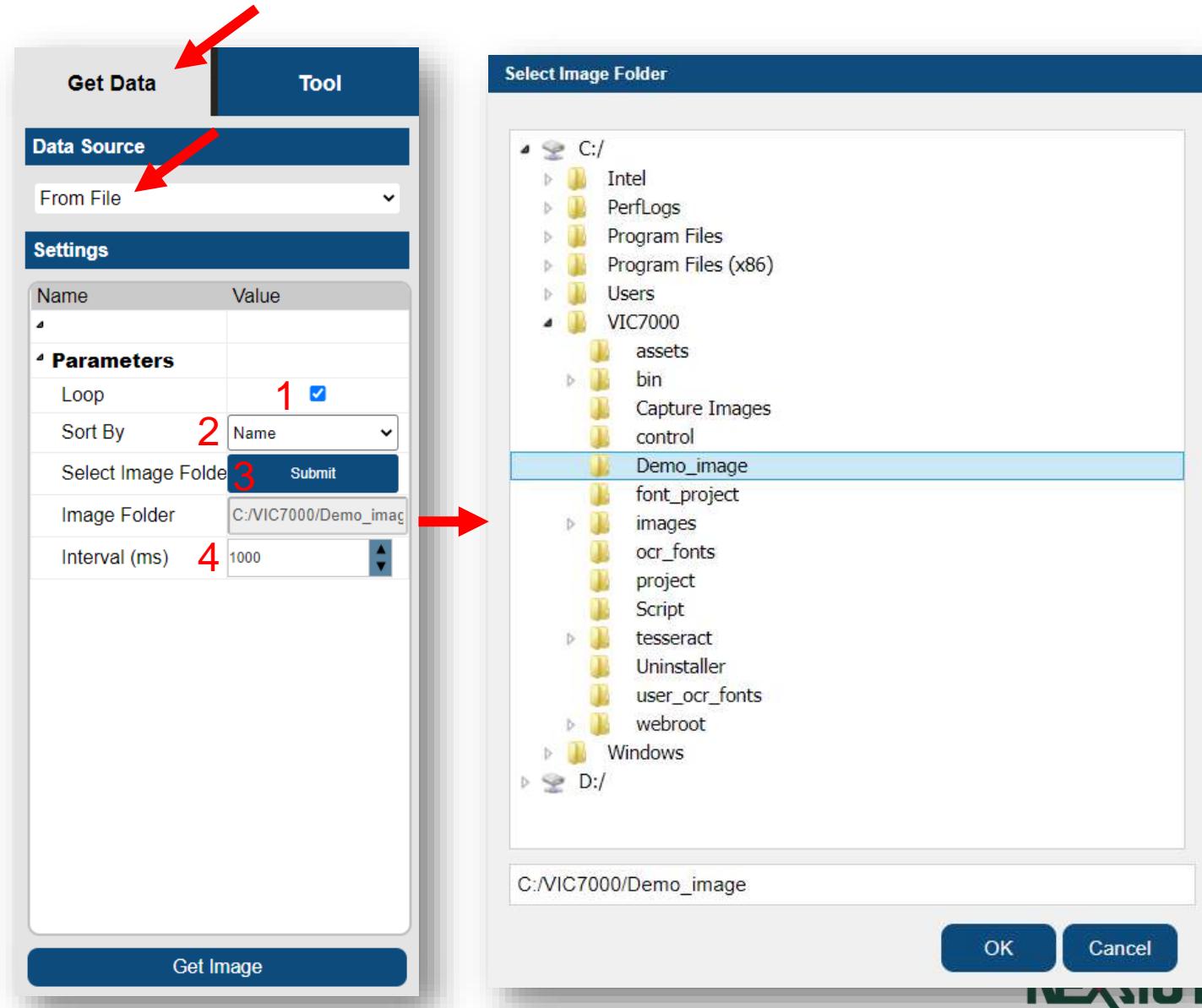
- 影像來源

- 來自檔案：讀取目的地資料夾中的每一張圖像( .bmp 或 .tif )
- 來自擷取卡：讀取擷取卡接收到的影像來源
- 來自 RTSP 影像：透過 RTSP 讀取影像輸入
- 來自瀏覽器：讀取透過 VIC7000 開啟的瀏覽器畫面
- 來自本機畫面擷取：讀取 VIC7000 機器本身畫面



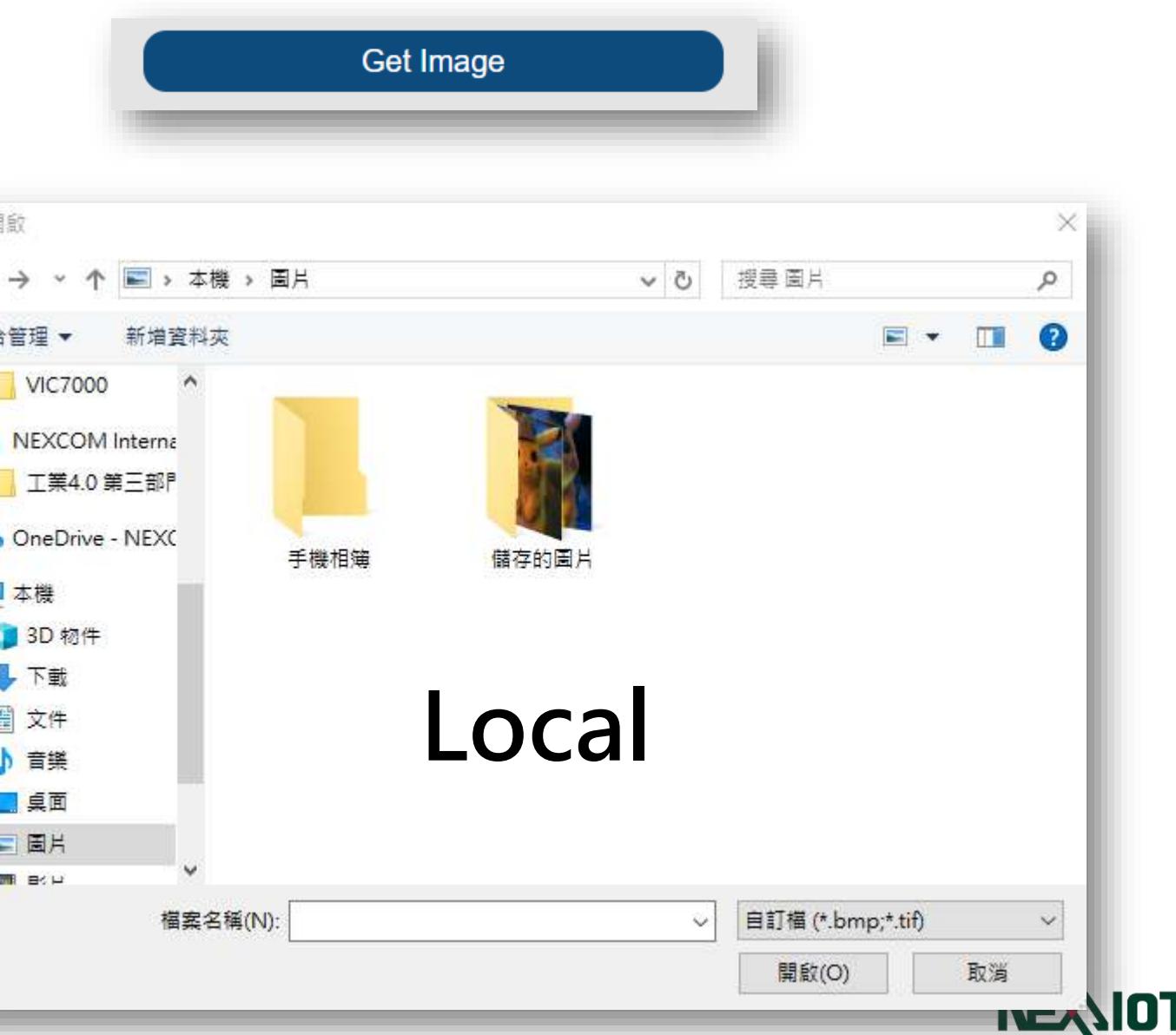
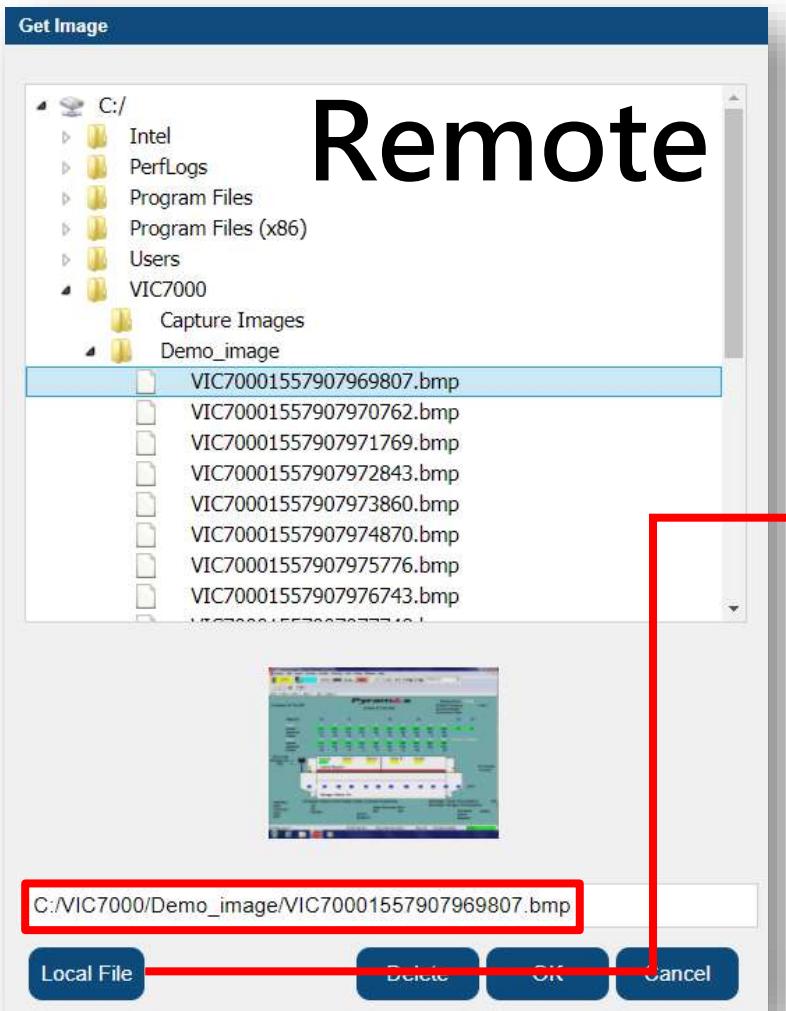
- 來自檔案

- 是否循環讀取資料夾內的圖片
- 檔案排序方式
- 選擇圖檔路徑
- 讀取時間間隔



- 來自檔案

- 取得圖像：讀取本機或遠端的圖像



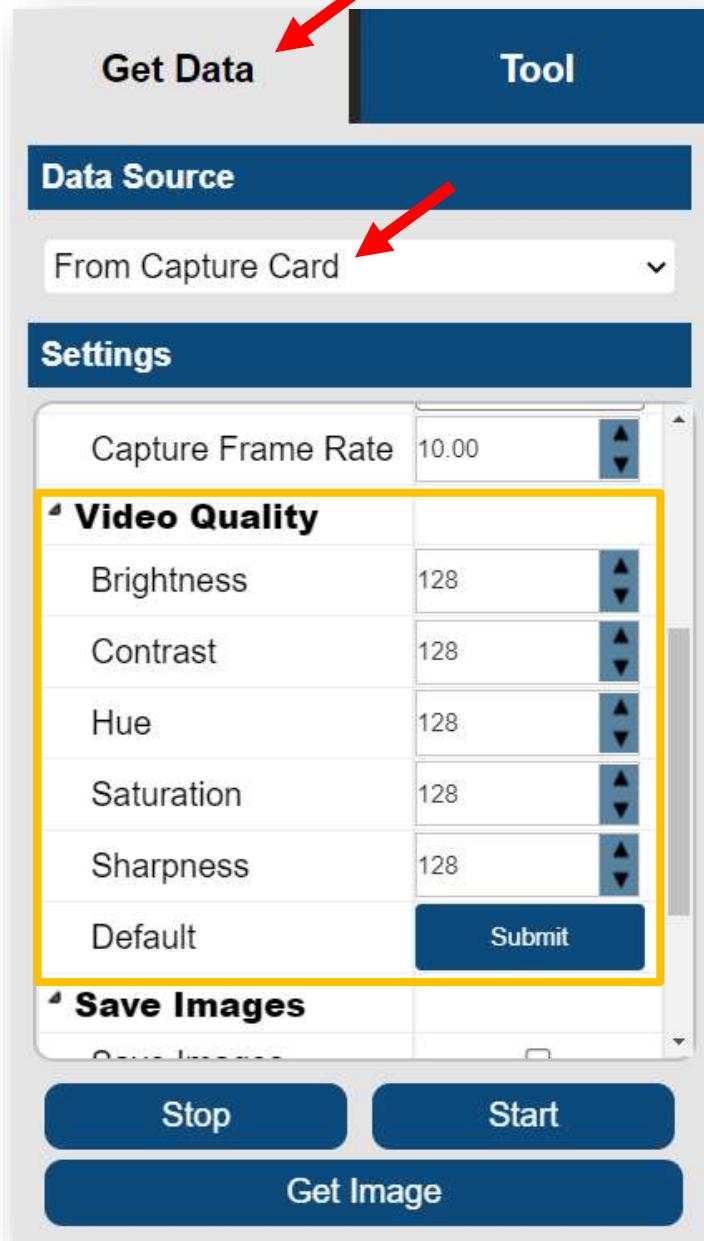
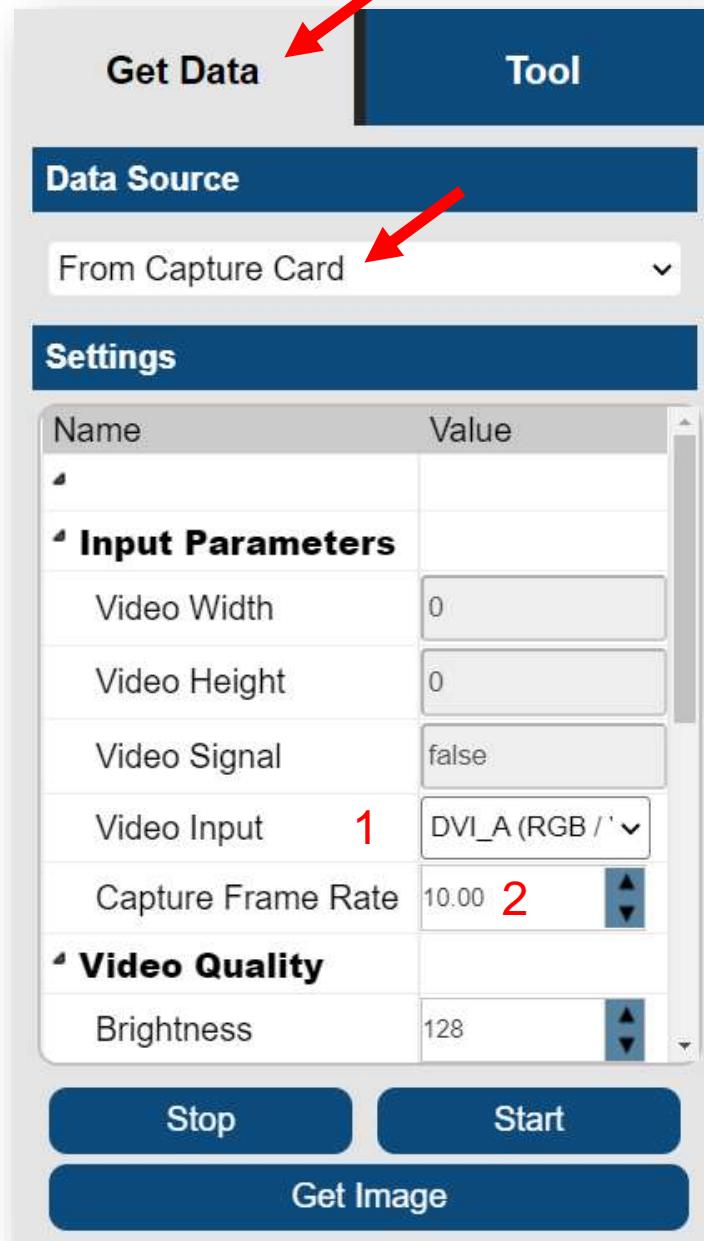
# 取像

- 來自擷取卡

- 影像輸入：通常為 DVI\_A (RGB / VGA)
- 擷取頻率 (fps)

- 取像參數

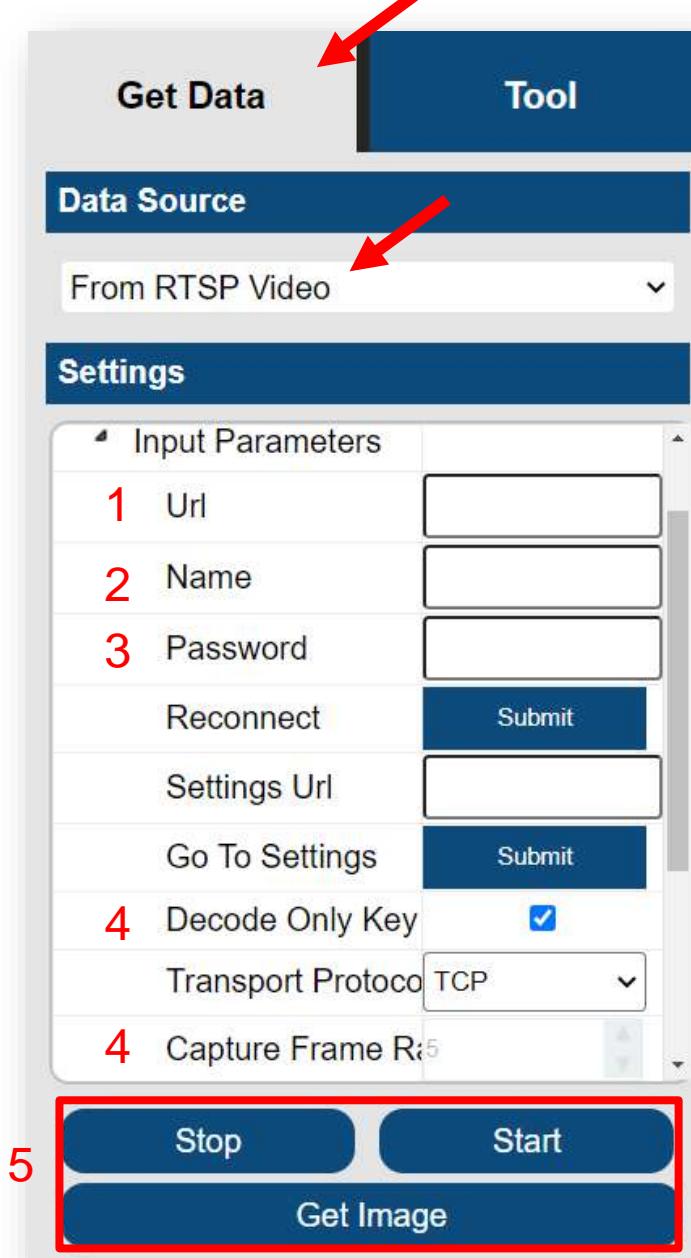
- 亮度
- 對比度
- 彩度
- 飽和度
- 銳利度
- 預設：恢復預設值



- 來自 RTSP 影像

1. RTSP 來源網址
2. 名稱
3. 密碼
4. 只解碼關鍵幀 or 摄取頻率
5. 啟動/停止：持續從 RTSP 來源取像

取得圖像：從 RTSP 來源取得一張圖像

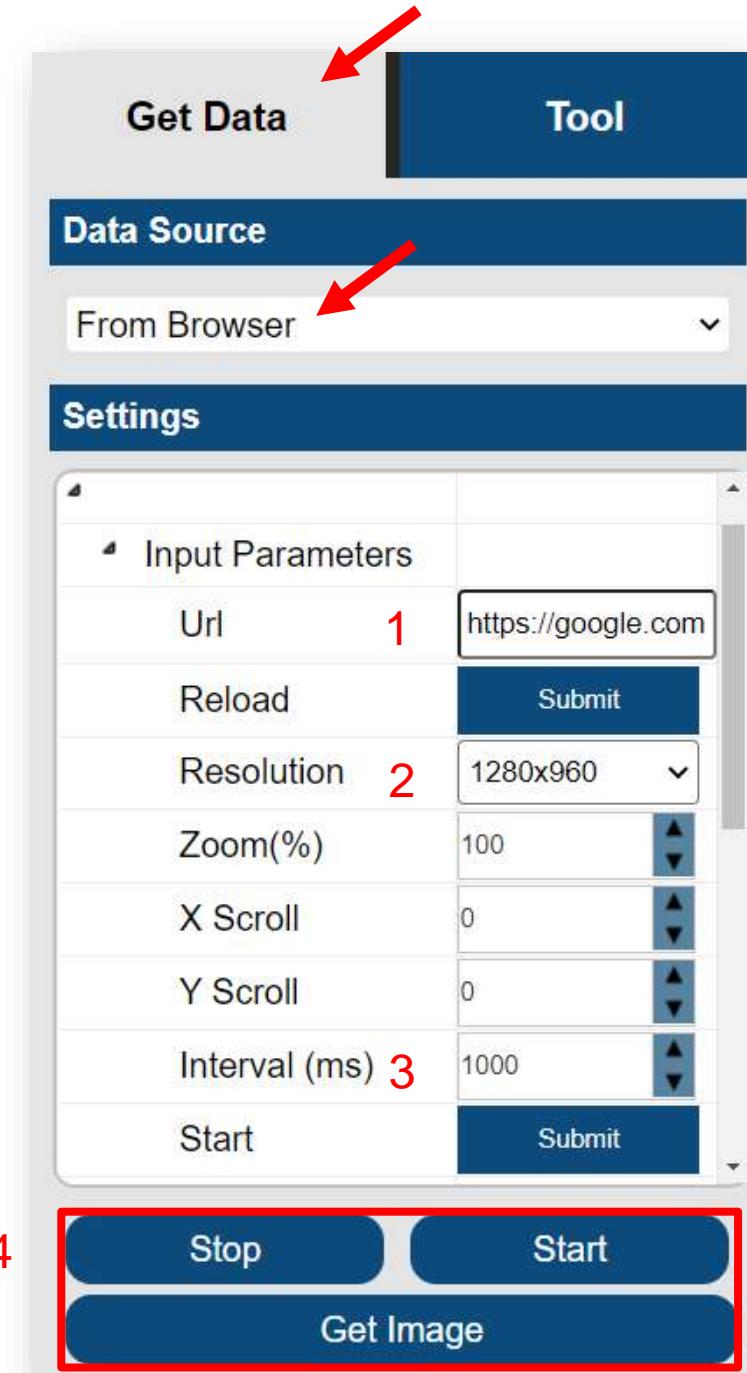


# 取像

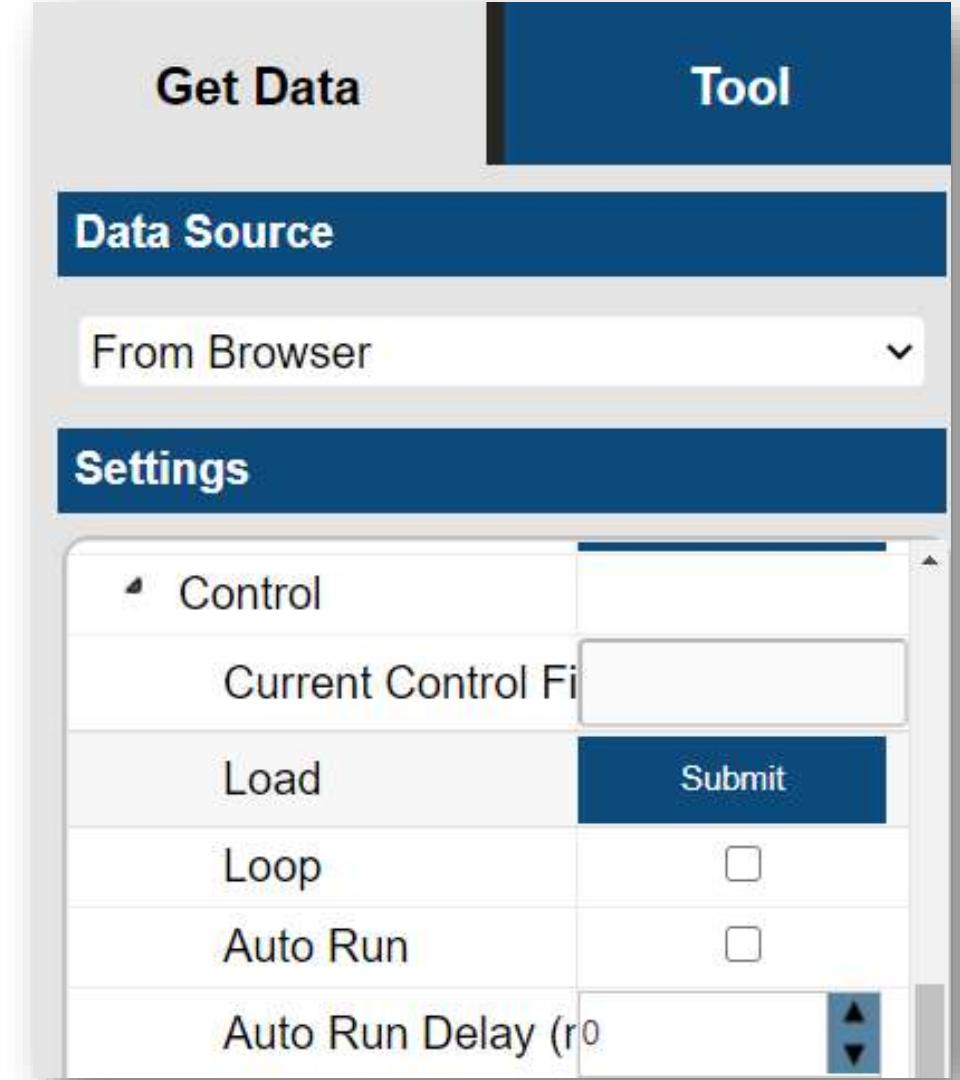
- 來自瀏覽器

1. 網址
2. 解析度
3. 讀取時間間隔
4. 啟動/停止：持續從瀏覽器來源取像

取得圖像：從瀏覽器來源取得一張圖像



- 來自瀏覽器
  - 控制：設定瀏覽器控制檔  
目前控制檔案  
載入控制檔  
是否循環執行控制檔  
是否在啟動時自動啟動控制檔  
自動啟動延遲時間



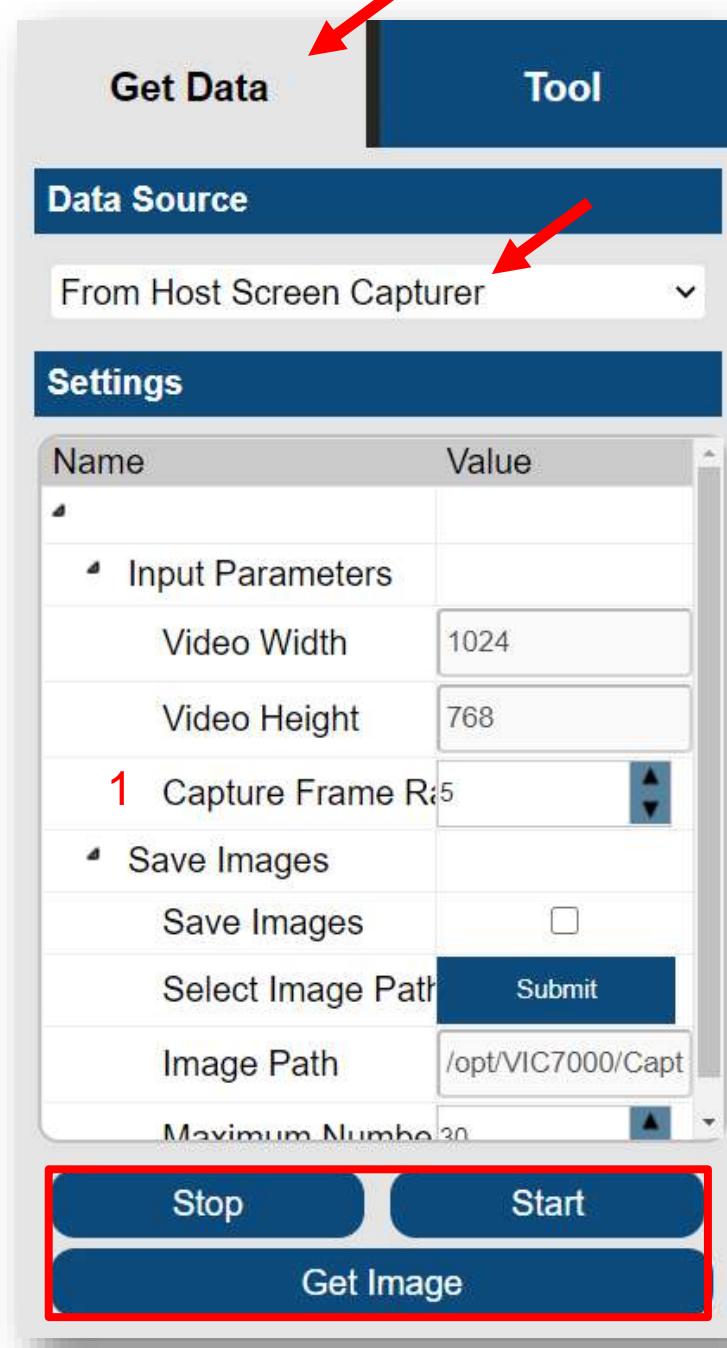
The screenshot shows the nexVIC software interface with the following configuration:

- Data Source:** Set to "From Browser".
- Settings:** Under the "Control" section, the "Current Control Fi" field is empty. The "Load" button is highlighted in blue. The "Loop" and "Auto Run" checkboxes are empty. The "Auto Run Delay (r0)" field has a value of 0 and includes up and down arrow buttons for adjustment.

# 取像

- 來自本機畫面擷取

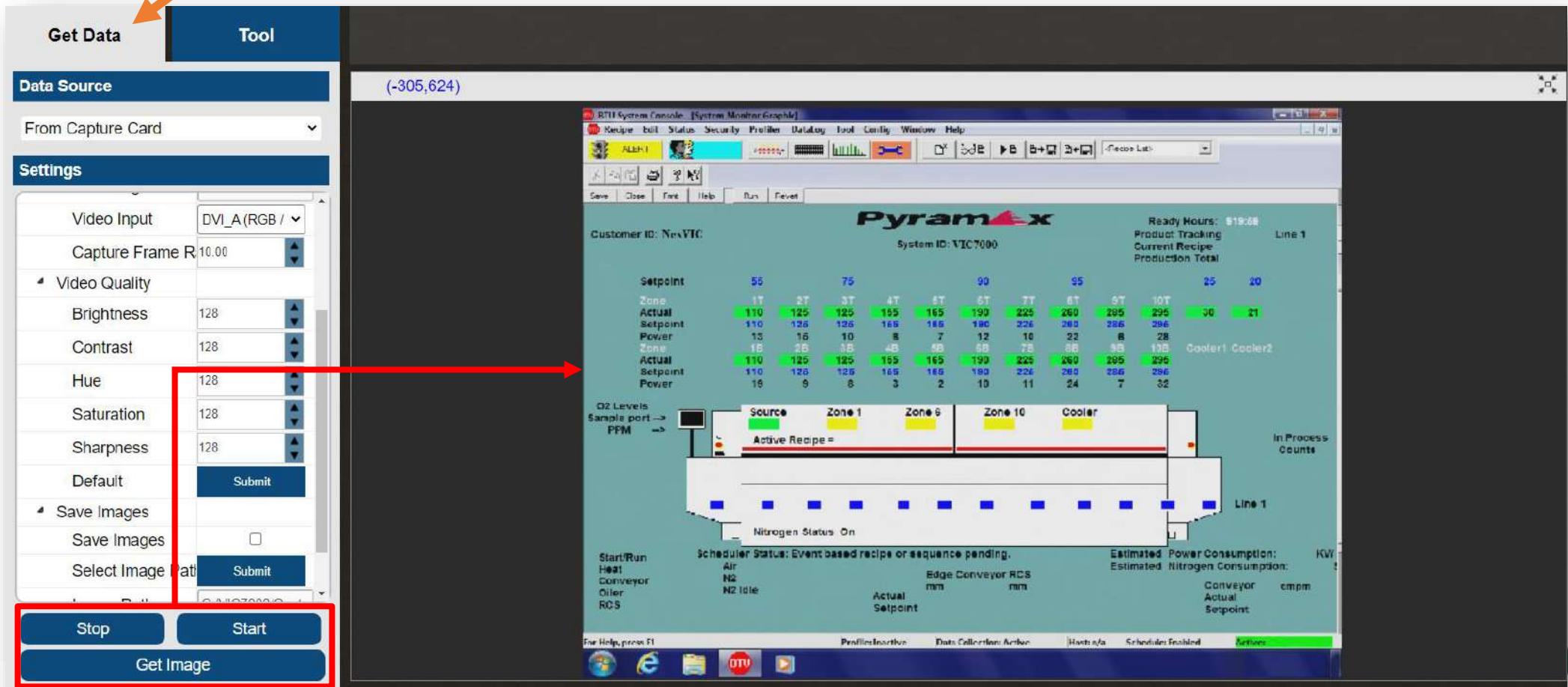
- 擷取頻率
- 啟動/停止：持續取得本機畫面  
取得圖像：取得一張本機畫面



2

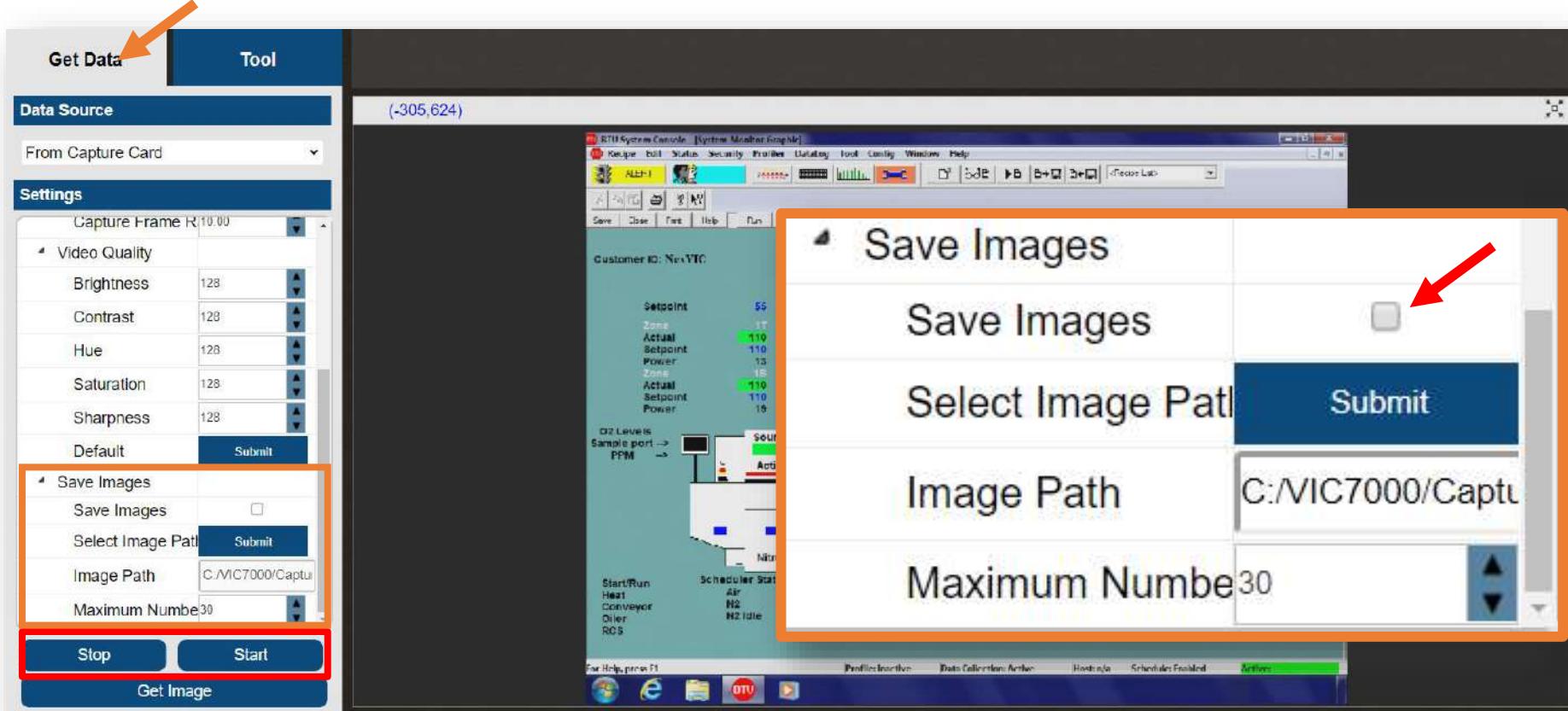
- 來自擷取卡 / RTSP 影像 / 瀏覽器 / 本機畫面擷取

- 啟動/停止：持續從擷取卡 / RTSP / 瀏覽器 / 本機畫面取得圖像
- 取得圖像：從擷取卡 / RTSP / 瀏覽器 / 本機畫面取得一張圖像



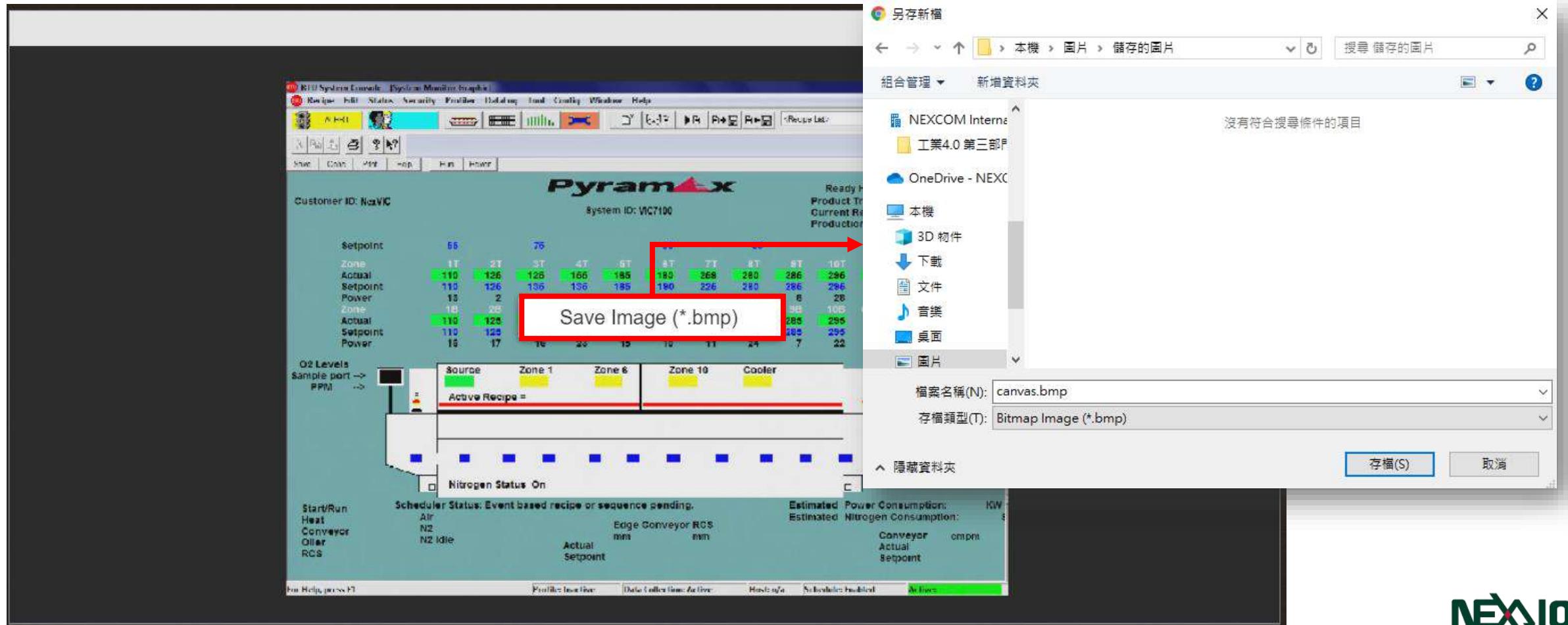
## • 邊讀取邊存圖

來源為來自擷取卡、來自 RTSP 影像、來自瀏覽器、來自本機畫面擷取時，可以邊取像邊存圖  
啟用儲存圖片功能，點擊啟動來持續從影像來源取得圖像，同時儲存圖片  
單一資料夾最多可儲存7200張圖片，儲存的圖片可用於字型學習或辨識設定



## • 儲存圖像

在圖像上點擊右鍵，可儲存圖像到本機



# Outline

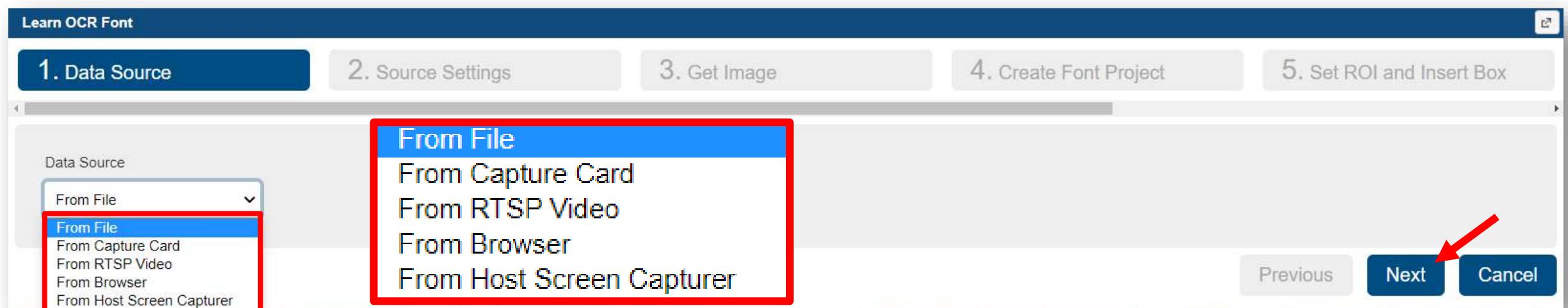
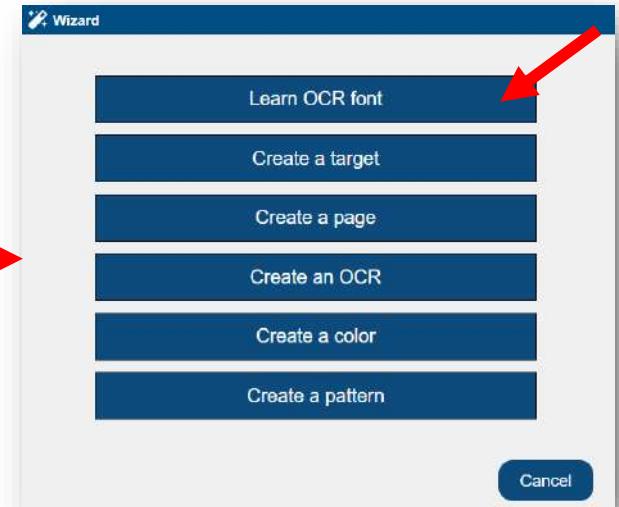
- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

- 精靈：辨識字型學習

1. 影像來源

來自檔案、來自擷取卡、來自 RTSP 影像

來自瀏覽器、來自本機畫面擷取

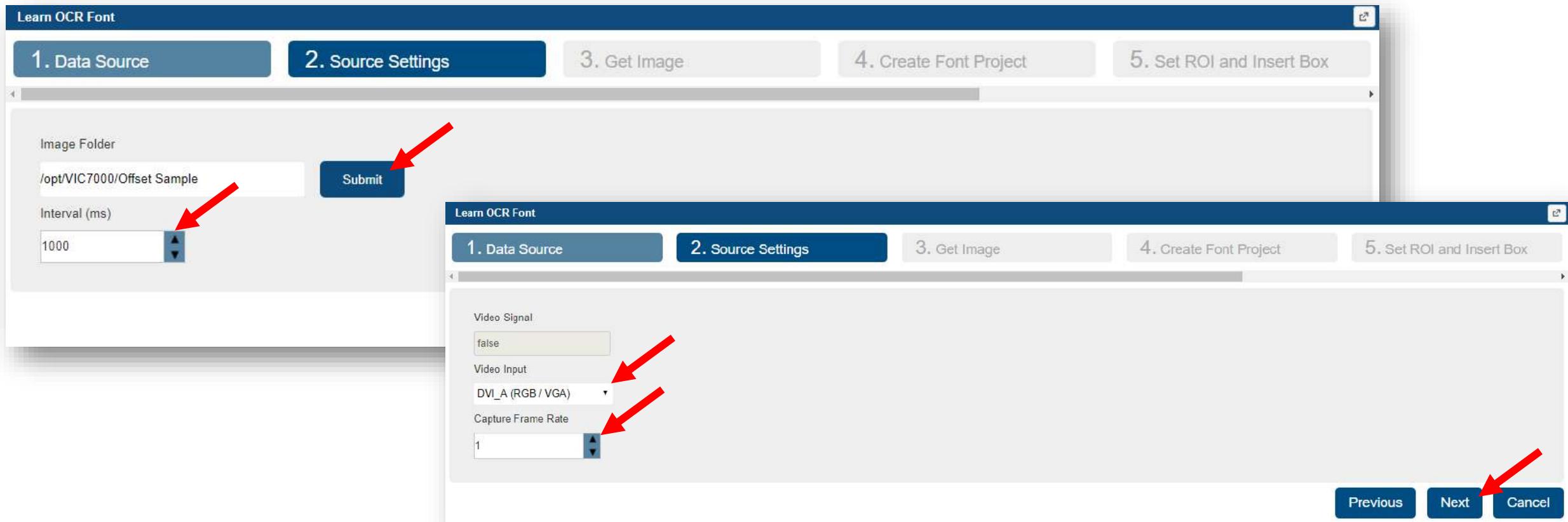


- 精靈：辨識字型學習

## 2. 影像來源設置

來自檔案：圖檔路徑、讀取時間間隔

來自擷取卡：輸入訊號類型、擷取頻率(fps)



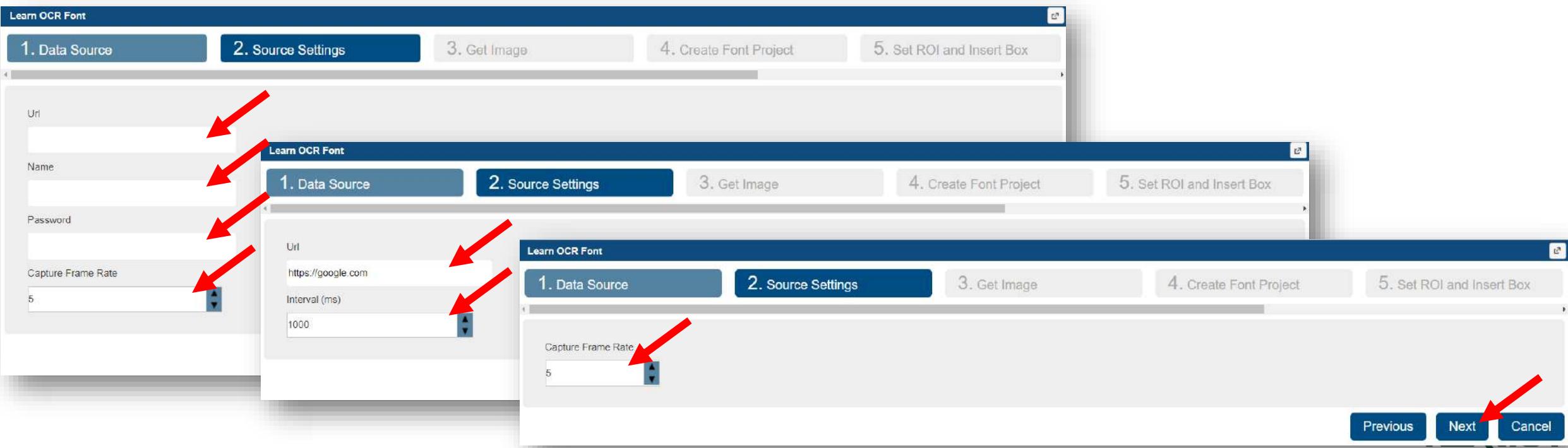
- 精靈：辨識字型學習

- 2. 影像來源設置

- 來自 RTSP 影像：RTSP 來源網址、名稱、密碼、擷取頻率

- 來自瀏覽器：網址、讀取時間間隔

- 來自本機畫面擷取：擷取頻率(fps)



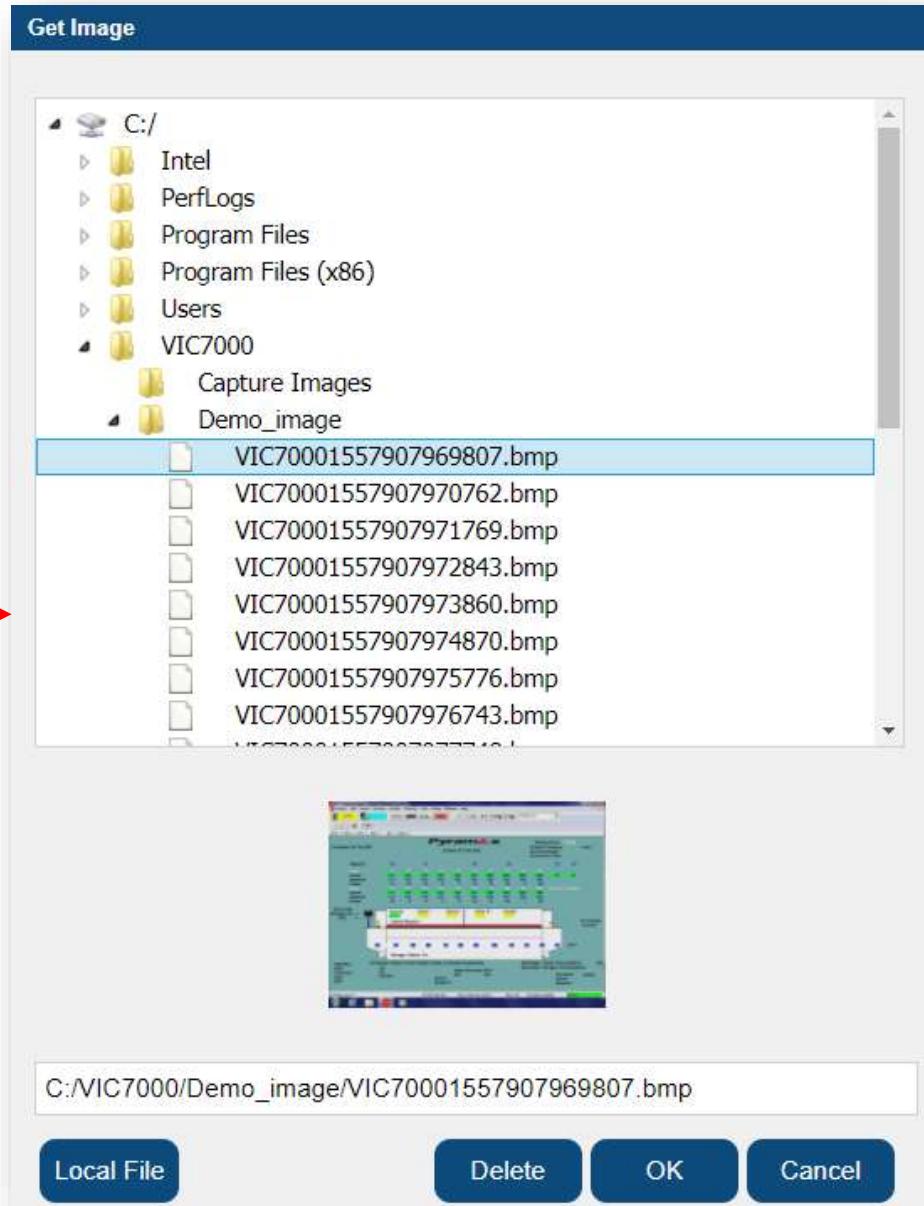
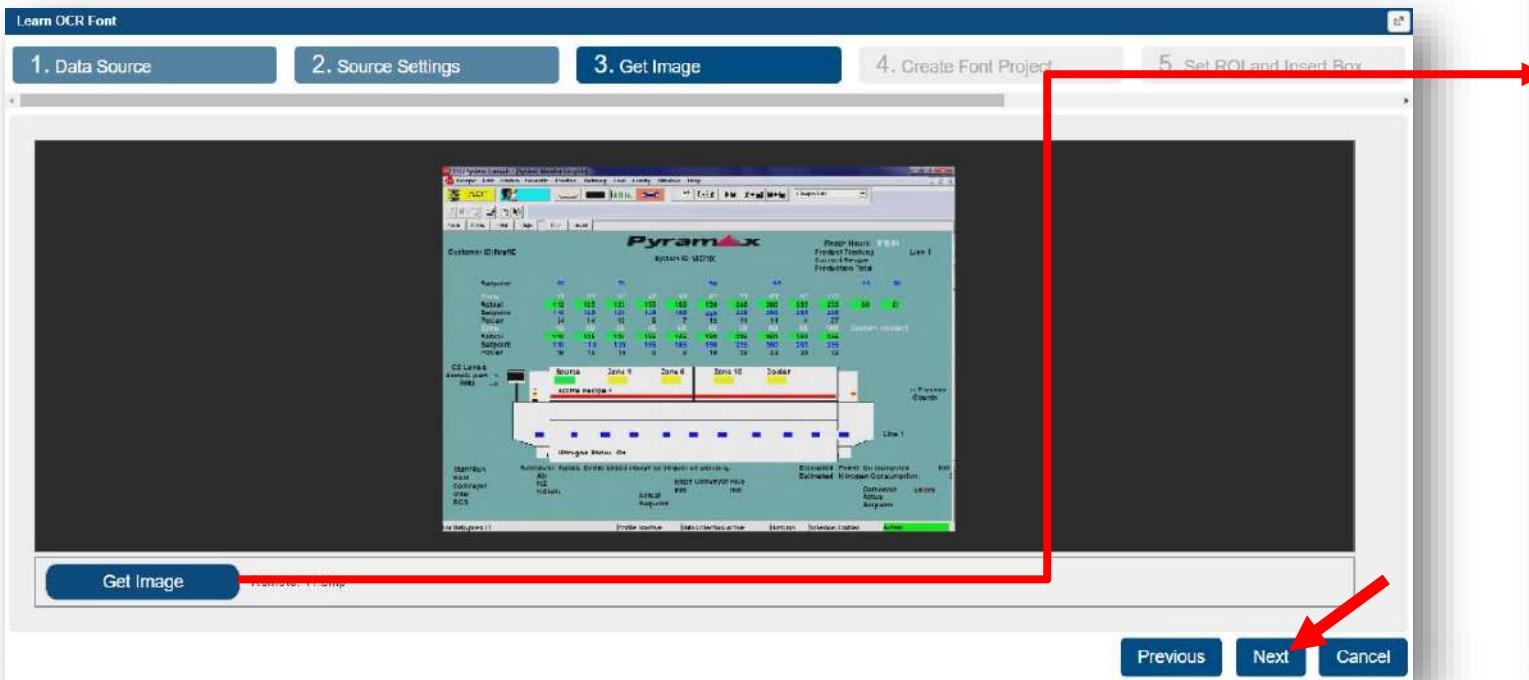
- 精靈：辨識字型學習

## 3. 取得圖像

來自檔案：讀取本機或遠端的圖像

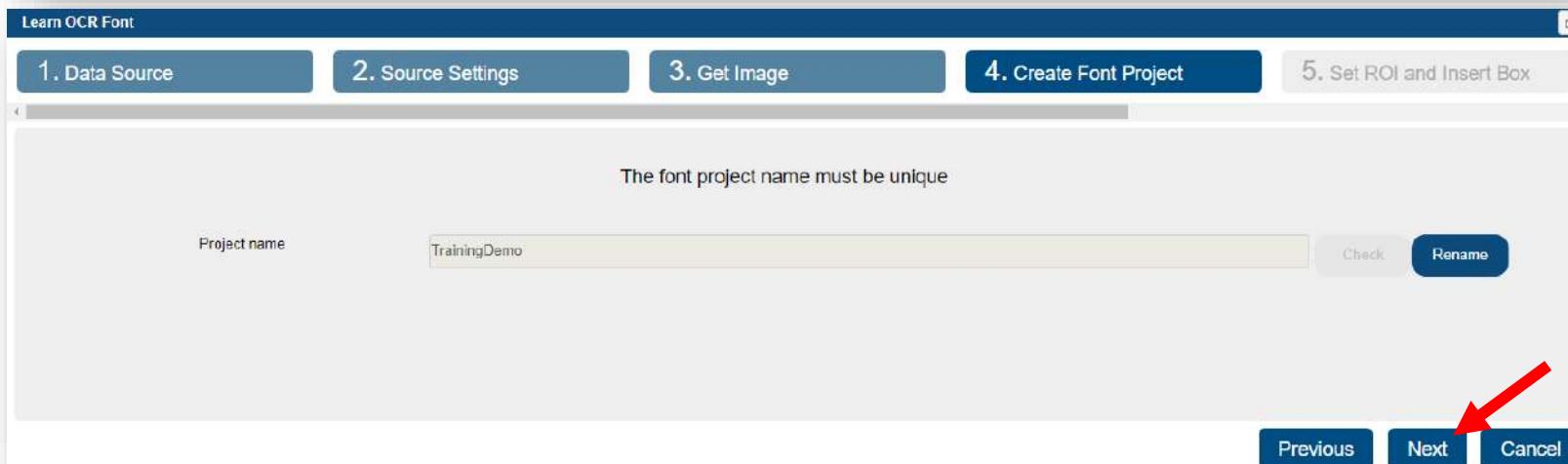
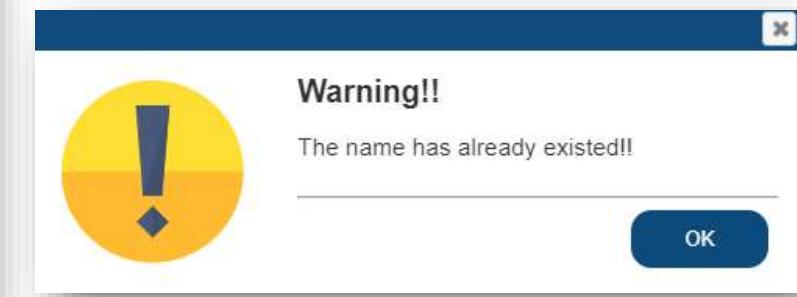
來自擷取卡、來自 RTSP 影像、來自瀏覽器、

來自本機畫面擷取：從取像來源取得一張圖像



- 精靈：辨識字型學習

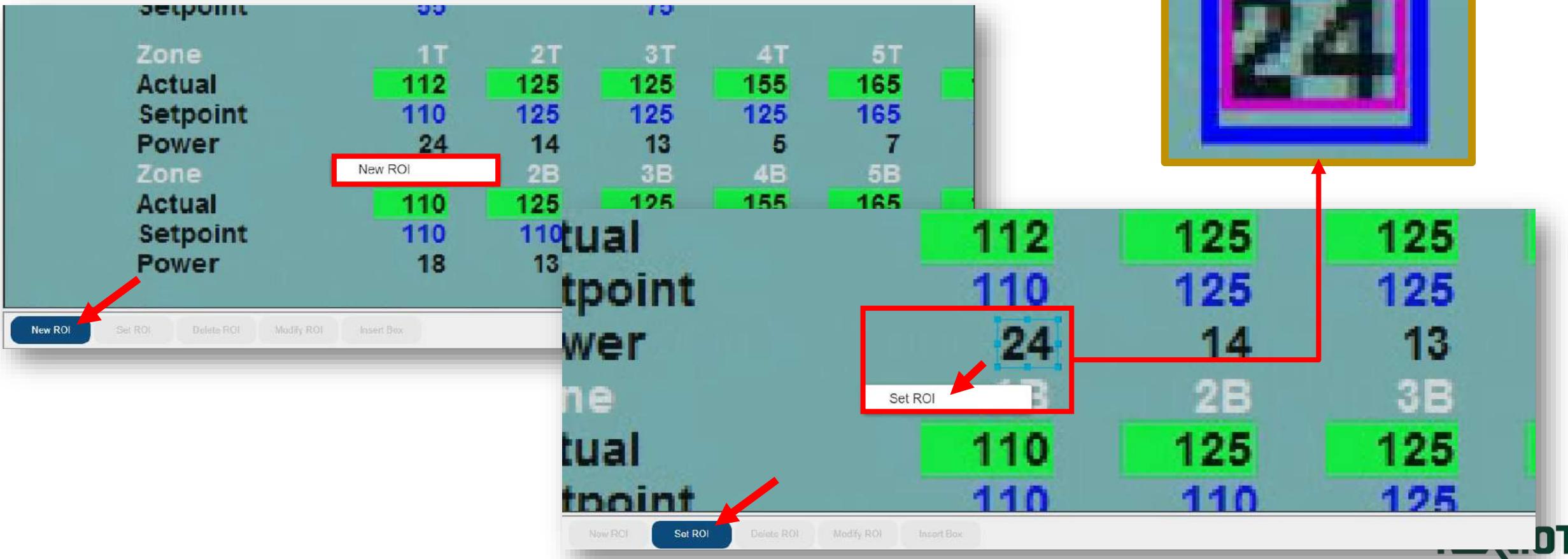
## 4. 建立字型專案：輸入專案名稱，字型檔專案名稱必須唯一，輸入名稱後需要檢查名稱是否重複



- 精靈：辨識字型學習

5. 設定ROI與新增字元框：右鍵新增ROI or 點擊新增ROI → 調整大小和位置 →

右鍵設置ROI or 點擊設置ROI



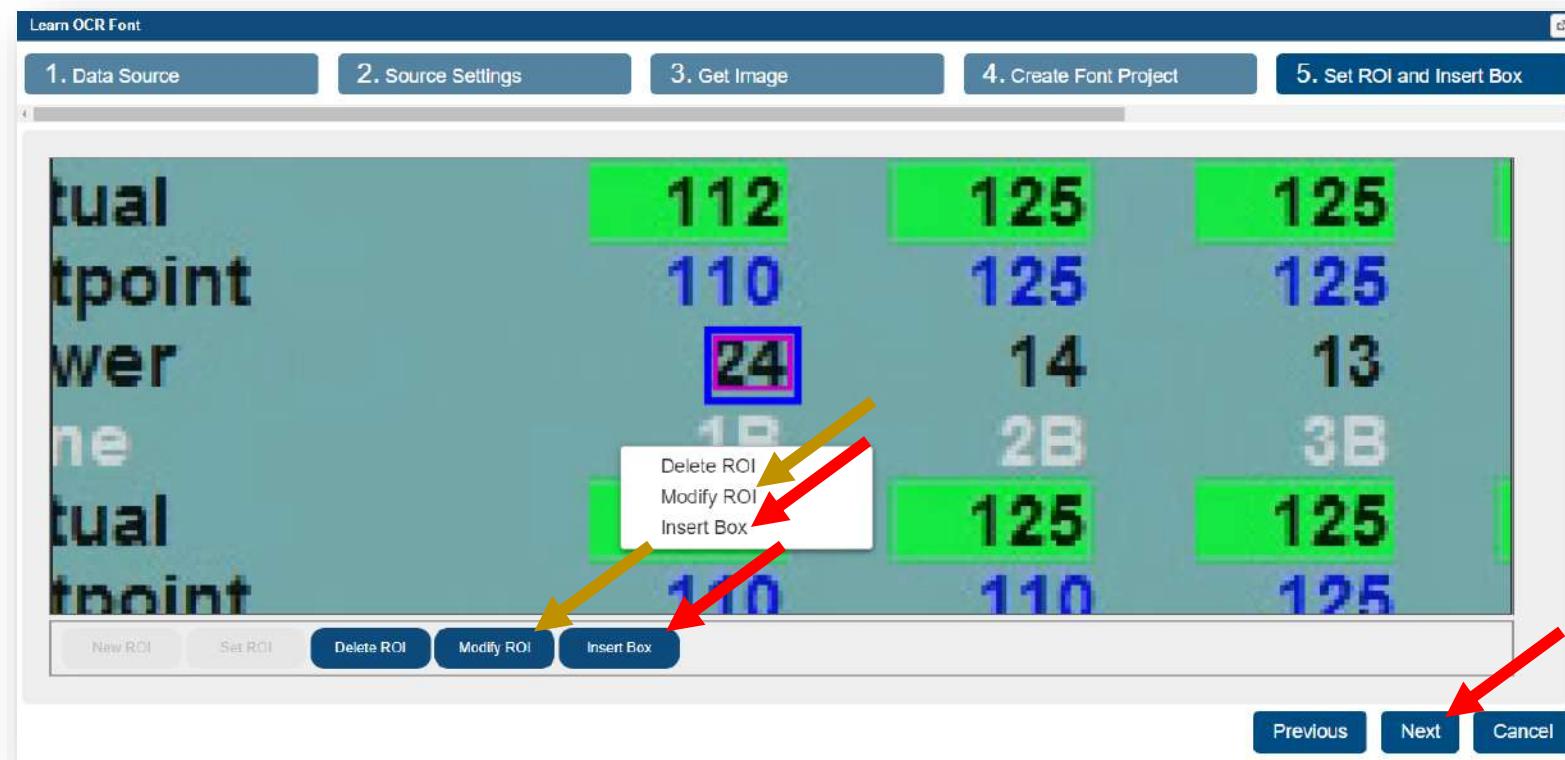
- 精靈：辨識字型學習

5. 設定ROI與新增字元框：右鍵新增字元框 or 點擊新增字元框 → 新增其他字元框

右鍵修改 ROI or 點擊修改 ROI → 調整 ROI 的大小和位置 → 右鍵設置 ROI or 點擊設置 ROI

→ 右鍵新增字元框 or 點擊新增字元框

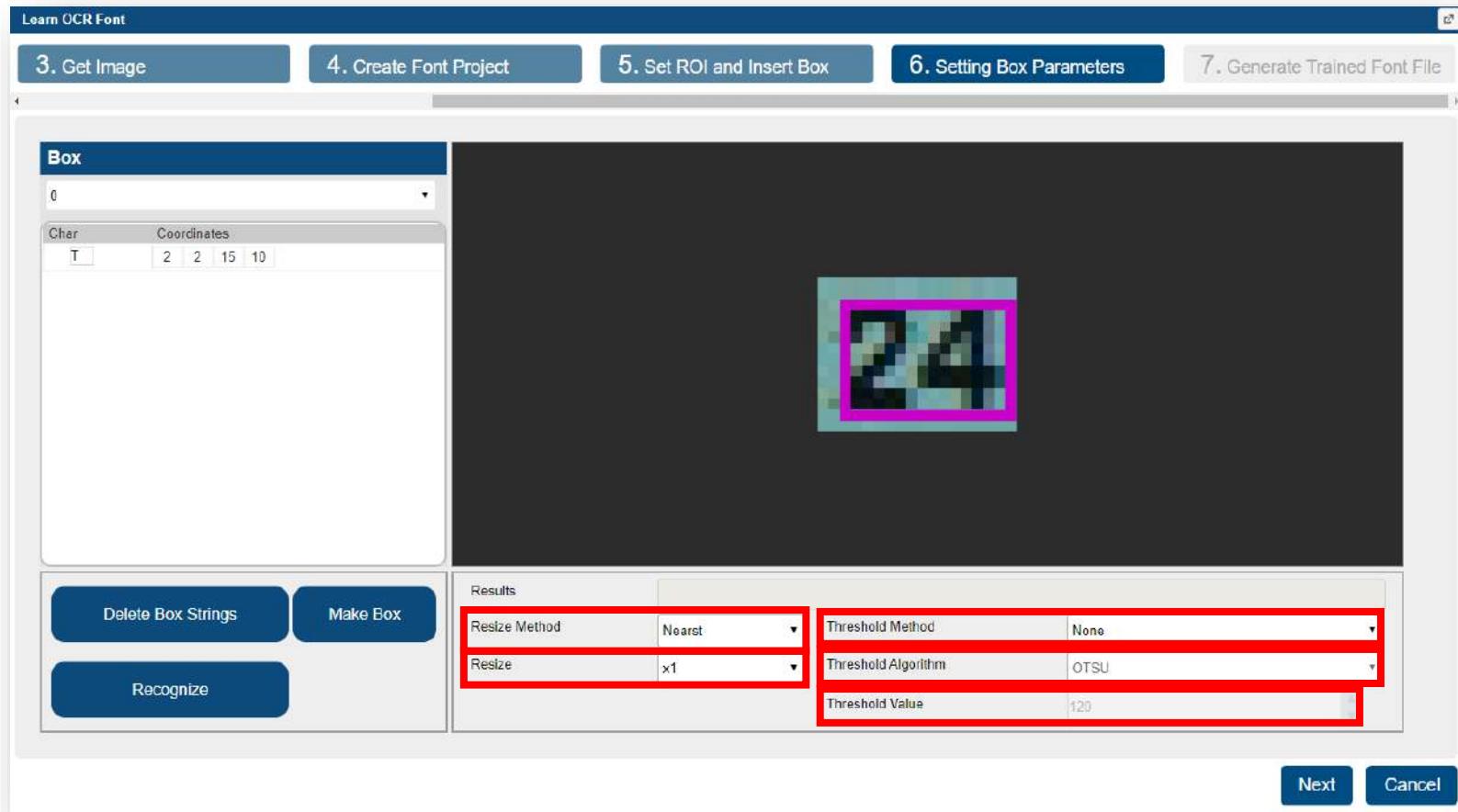
重複修改 ROI 和新增字元框直到所有要學習的字都有新增字元框



- 精靈：辨識字型學習

6. 設定字元框參數：設定調整大小方式和倍數 → 設定二值化方法、二值化演算法和閥值

讓每個字都可以被單一個粉色框圈住



- 字元框參數

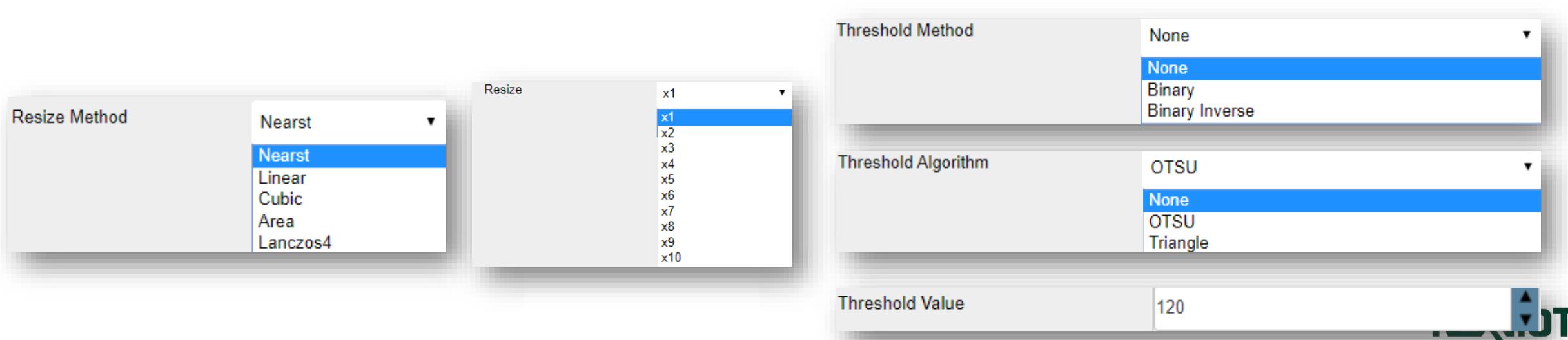
調整大小方式：最近鄰插值、雙線性插值、雙立方插值、區域取樣、Lanczos4

調整大小：x1 ~ x10

二值化方法：無、二分、二分反向

二值化演算法：無、OTSU、Triangle

閥值：若二值化方法選擇二分或二分反向 且 二值化演算法選擇無，就需要調整二值化閥值



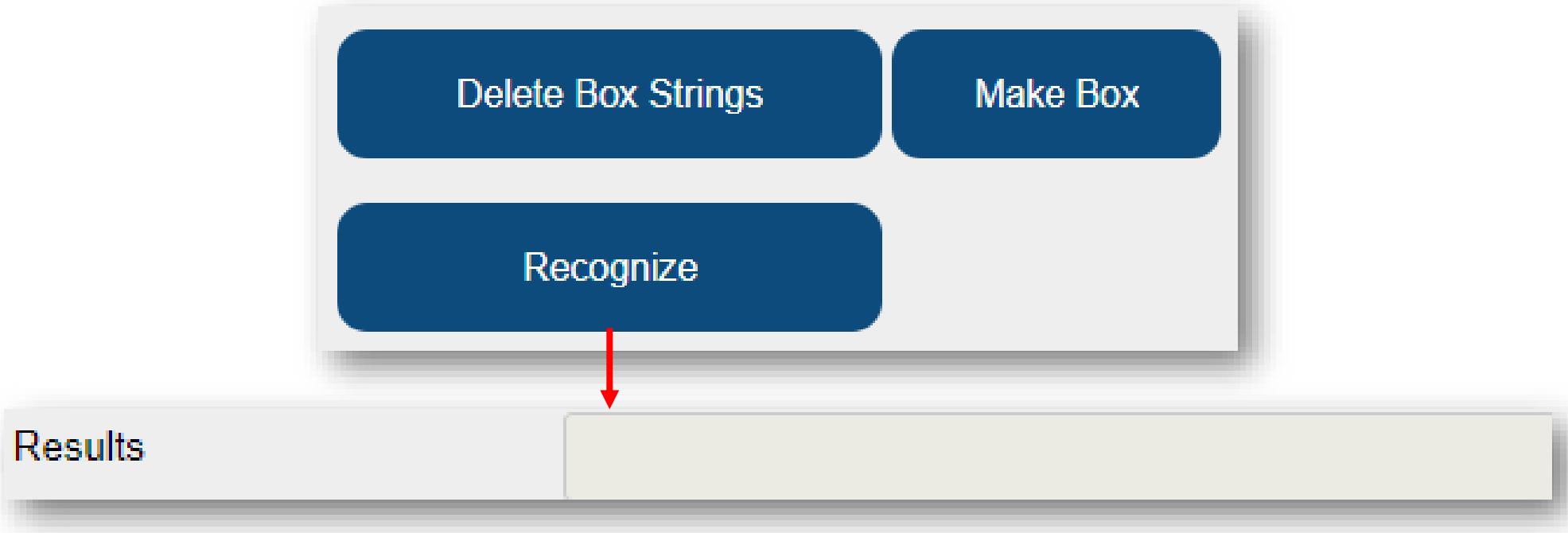
- 精靈：辨識字型學習

6. 設定字元框參數：

刪除字元框字串：刪除當前字元框

制作字元框：對當前畫面制作字元框

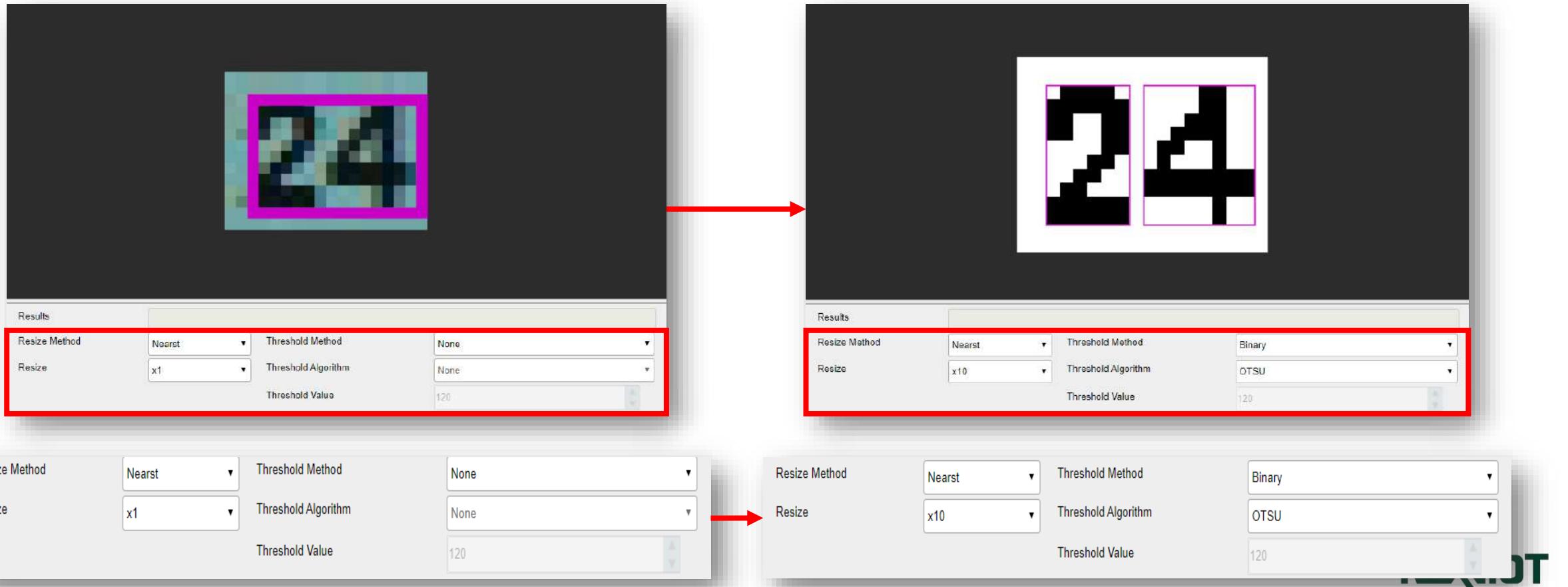
辨識：立即執行辨識，辨識結果會顯示在結果欄位中



- 精靈：辨識字型學習

- 6. 設定字元框參數：設定調整大小方式和倍數 → 設定二值化方法、二值化演算法和閥值

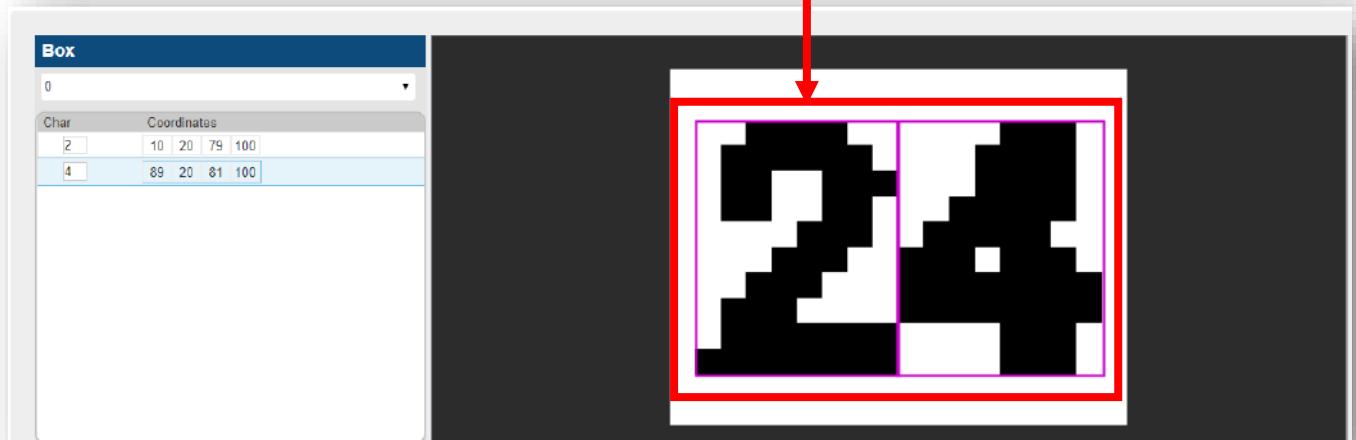
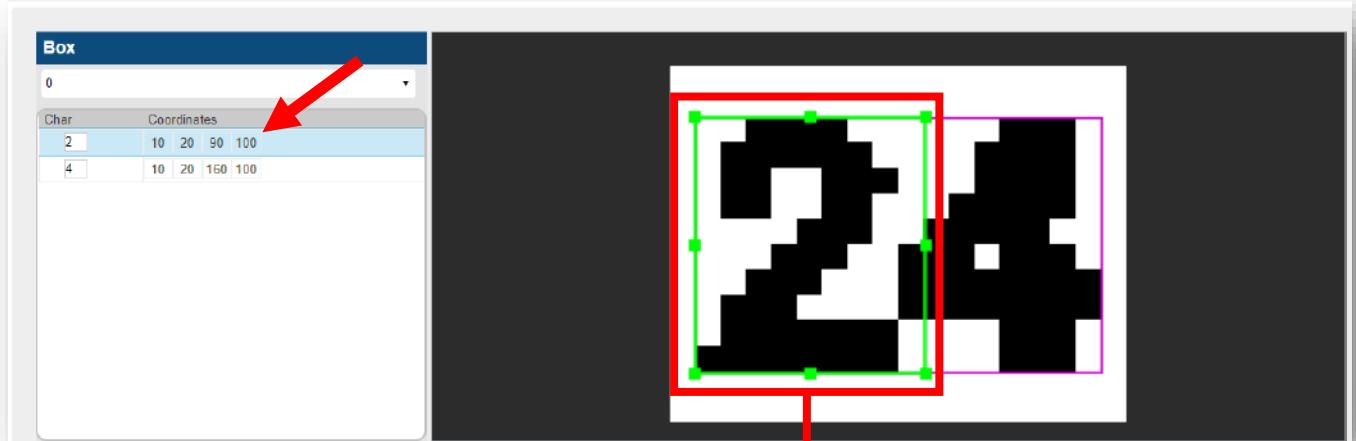
讓每個字都可以被單一個粉色框圈住



- 精靈：辨識字型學習

6. 設定字元框參數：點擊字元框內的字元，會顯示系統辨識的字元位置

若圈選範圍不符預期，就必須手動調整圈選位置，直接拖拉調整圈選位置



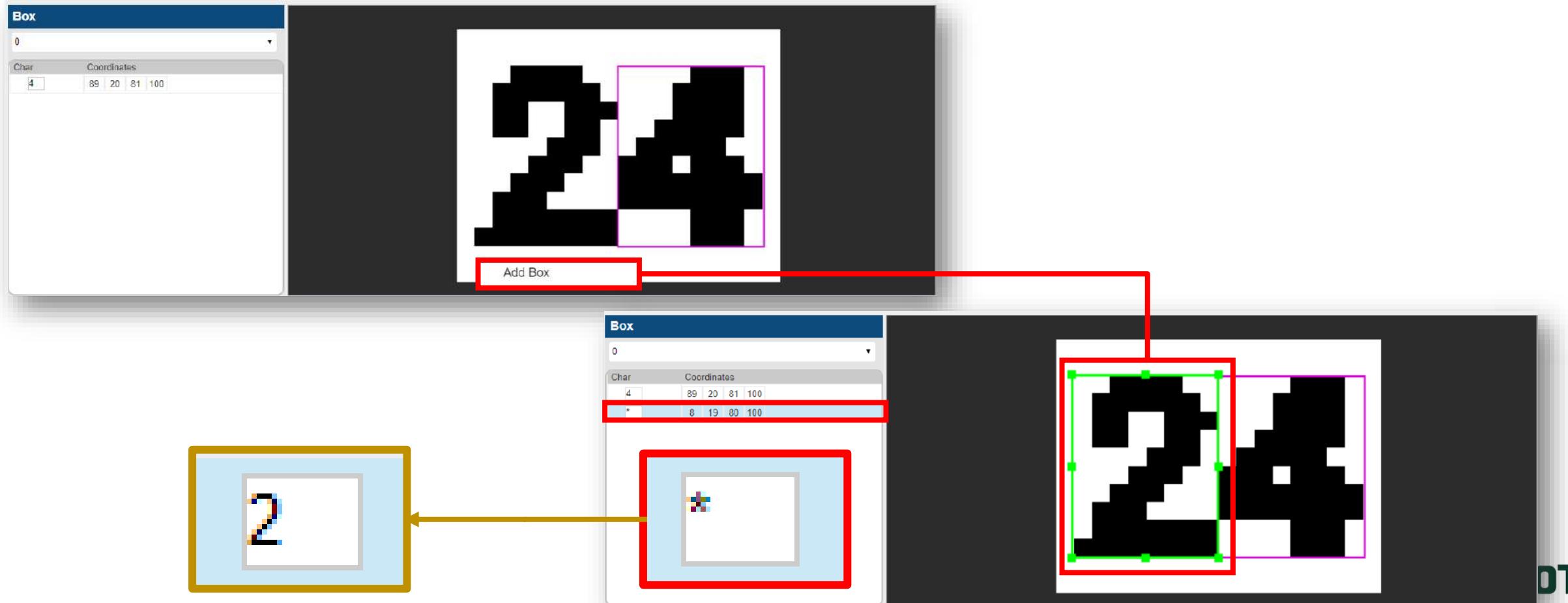
- 精靈：辨識字型學習

- 6. 設定字元框參數：若系統辨識結果不符預期，修改字元框內的字元



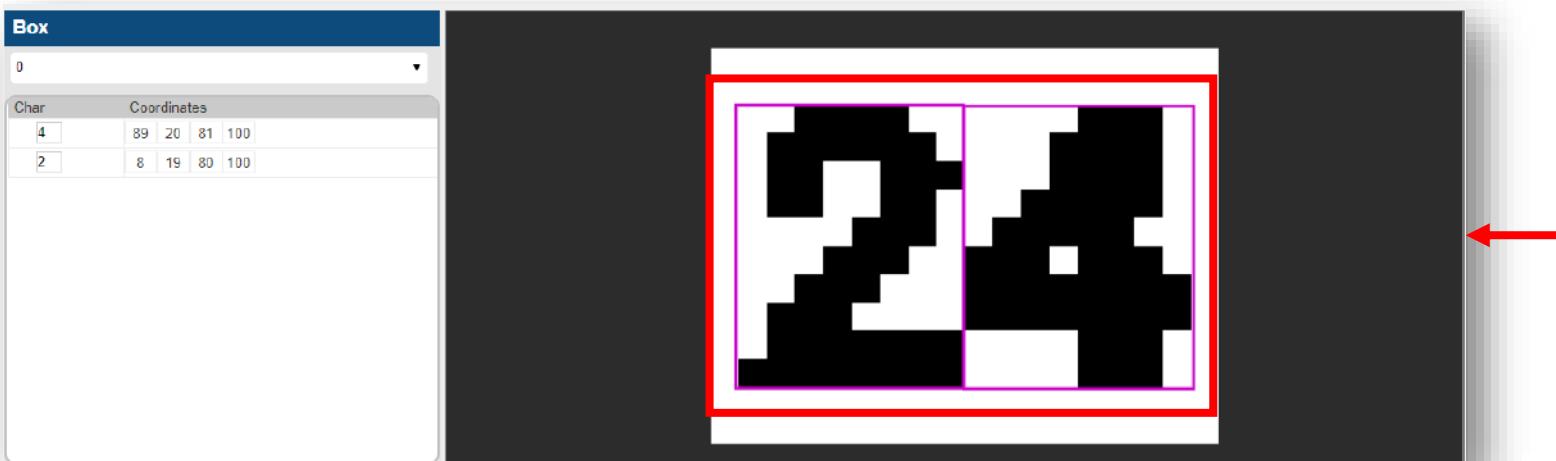
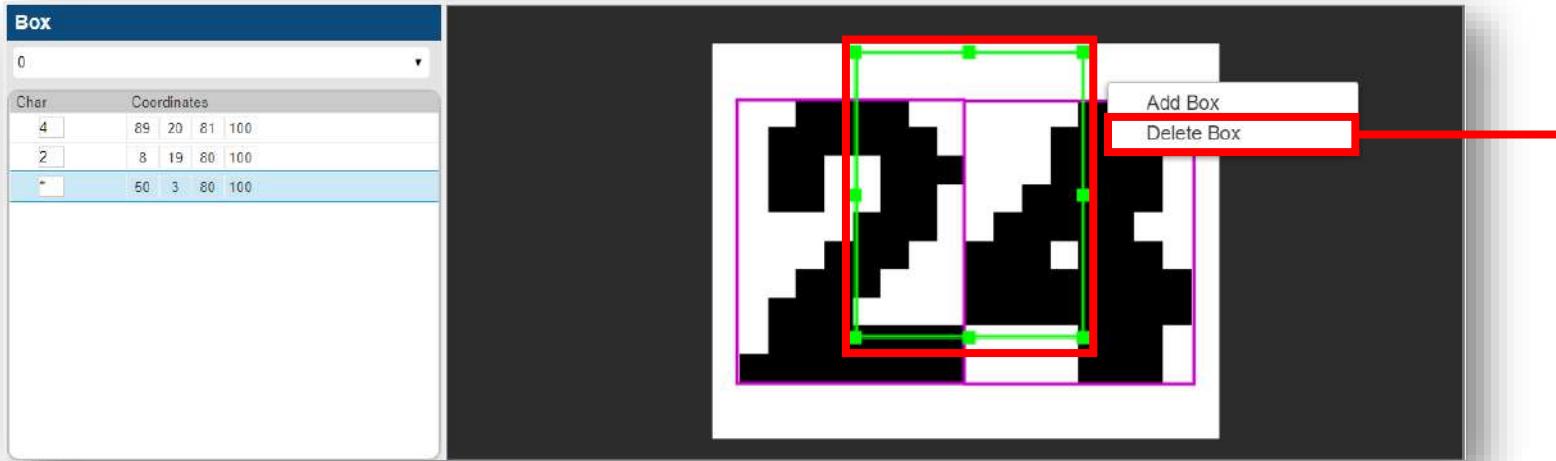
- 精靈：辨識字型學習

6. 設定字元框參數：若字元框內有少缺字元，在字元框內點擊右鍵加入字元框 → 調整字元框的大小和位置，圈選住要辨識的字元 → 修改字元框內的字元



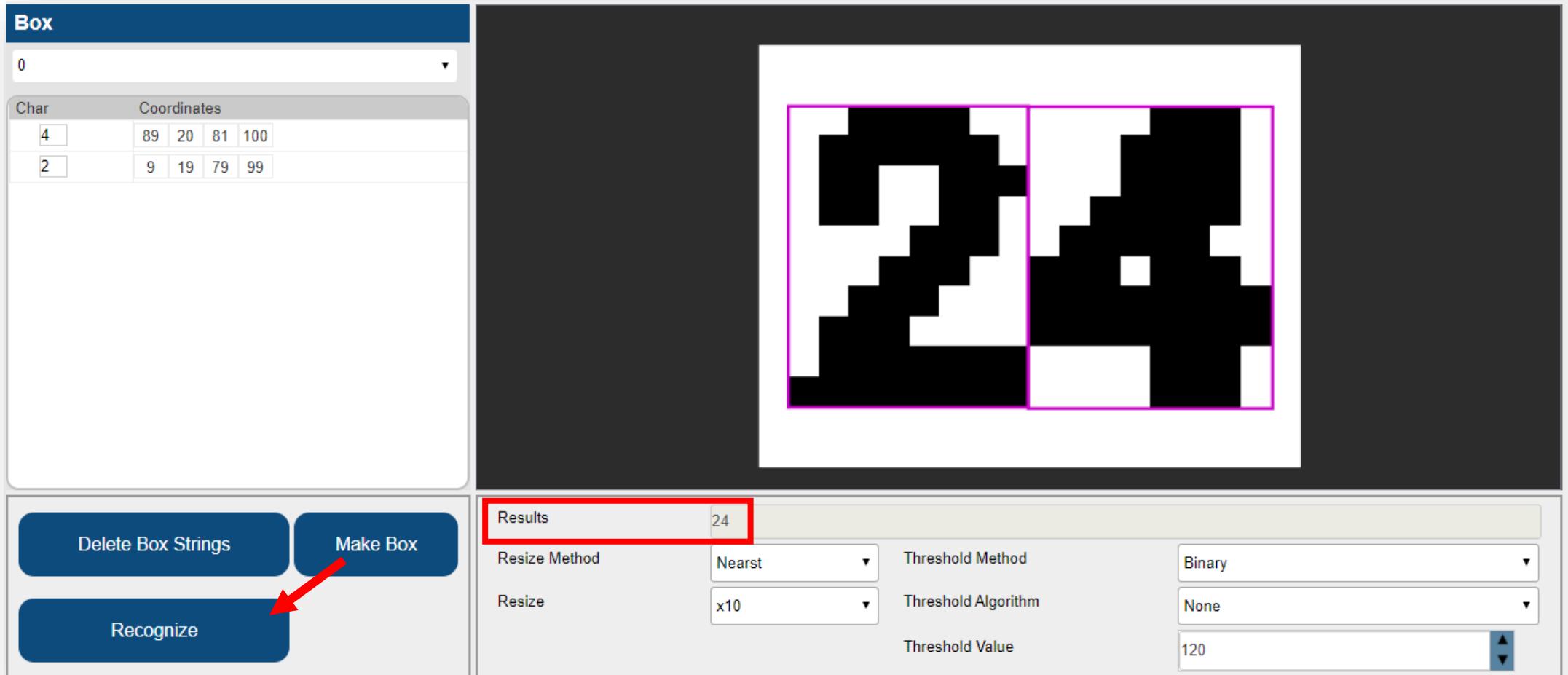
- 精靈：辨識字型學習

6. 設定字元框參數：若字元框內有多餘的圈選區域，選擇該字元框 → 右鍵選擇刪除字元框



- 精靈：辨識字型學習

6. 設定字元框參數：字元框設定結束後，點擊辨識 → 在結果欄位內確認辨識結果正確

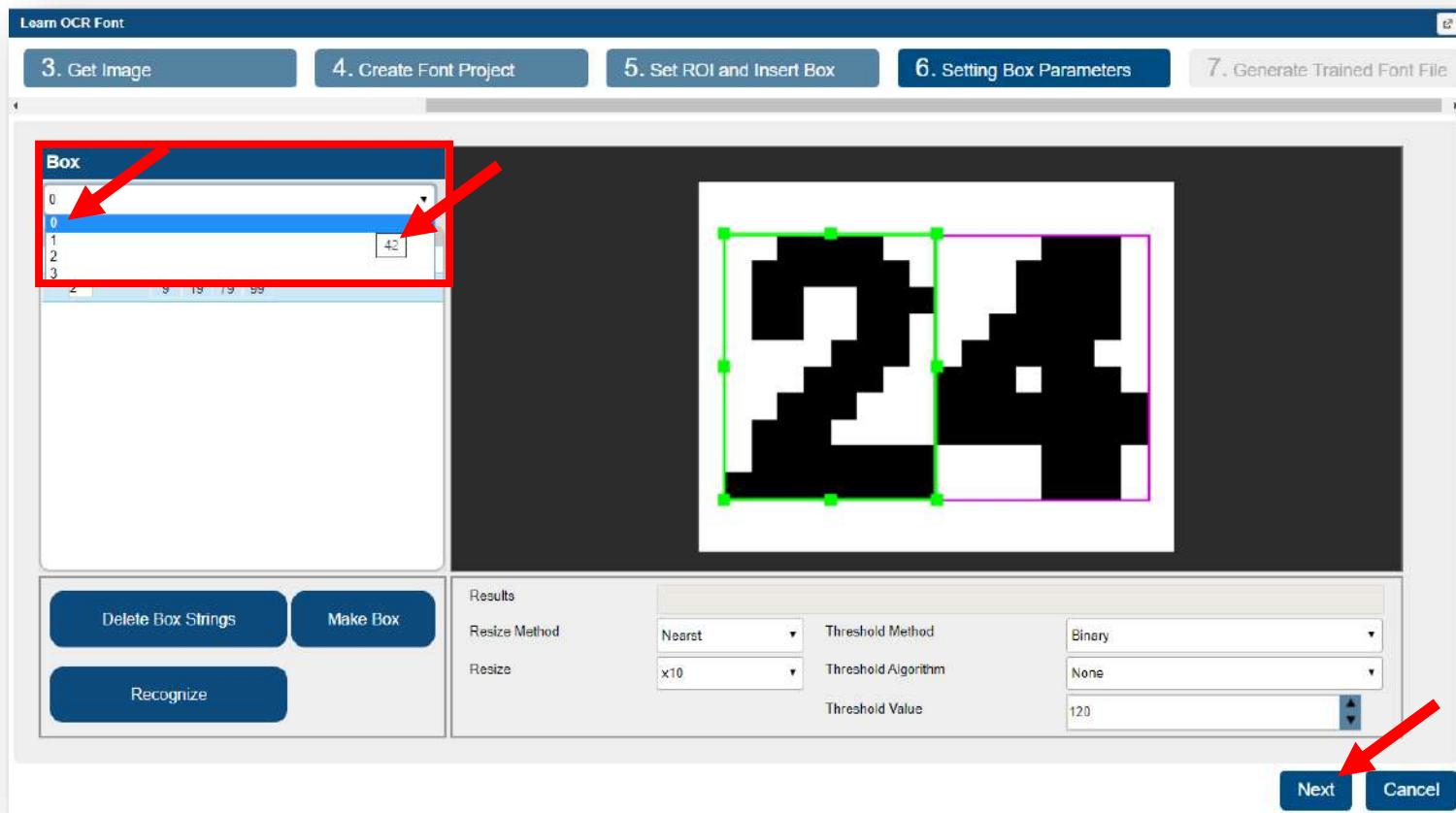


- 精靈：辨識字型學習

- 6. 設定字元框參數：開啟字元框選單，依照上述步驟設定每一個字元框

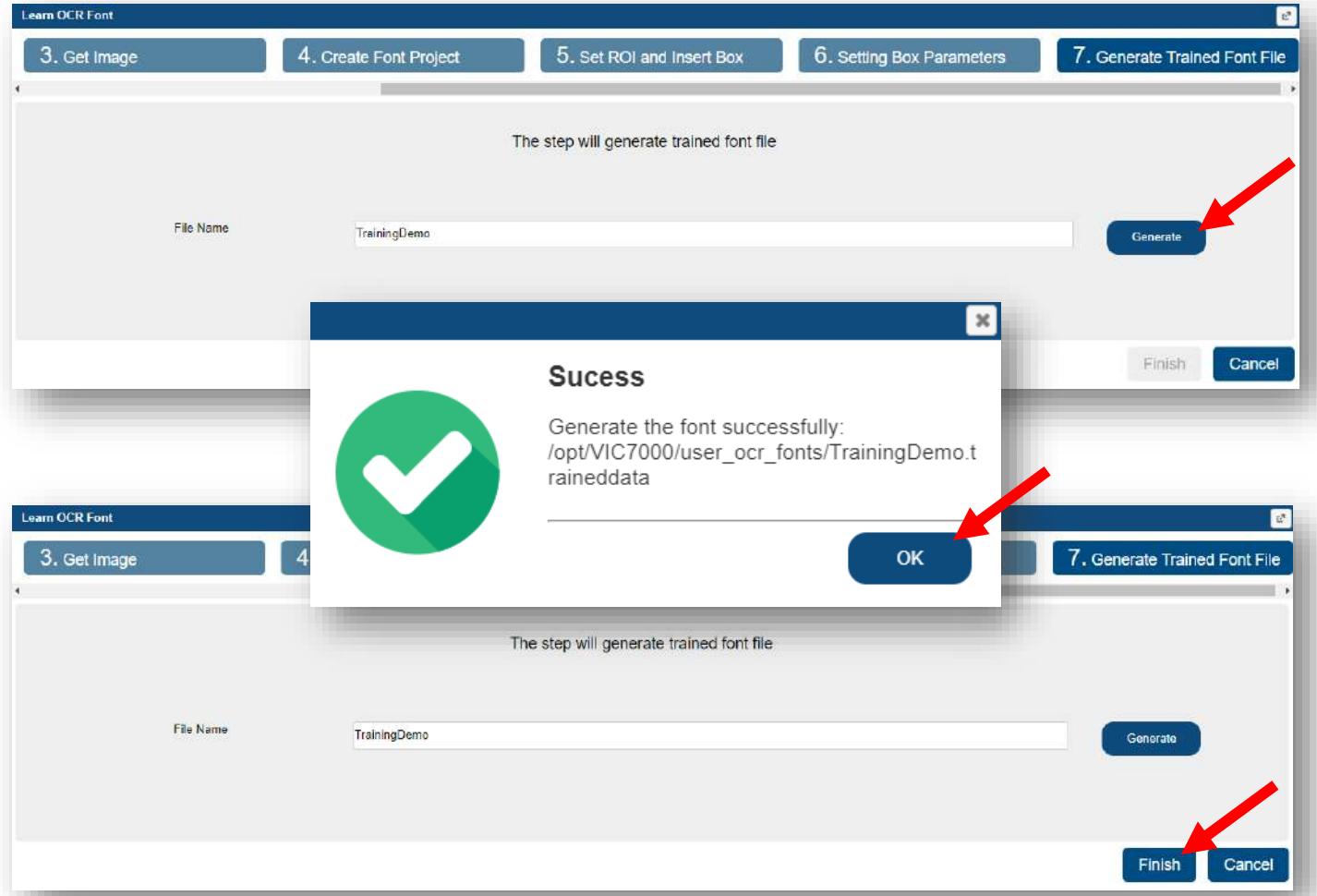
移動游標到字元框編號上，會顯示該字元框的字元內容

每個字元框的調整大小和二值化參數，盡量都要相同



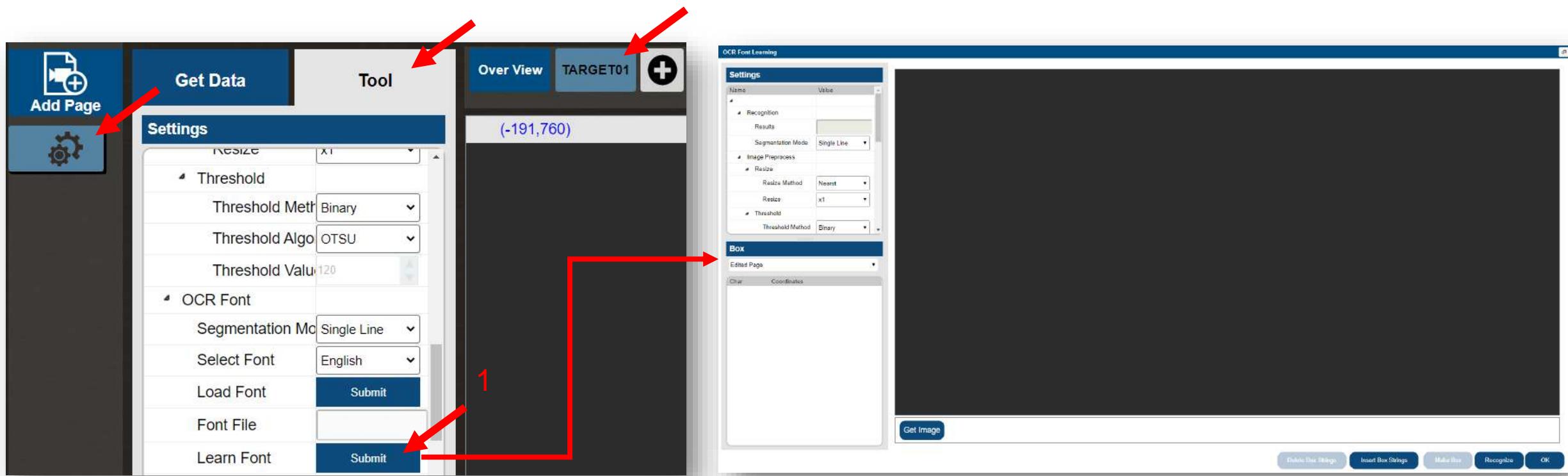
- 精靈：辨識字型學習

7. 生成訓練字型檔案：若必要，修改檔名 → 點擊產生字型 → 生成成功後，點擊完成



- 一般

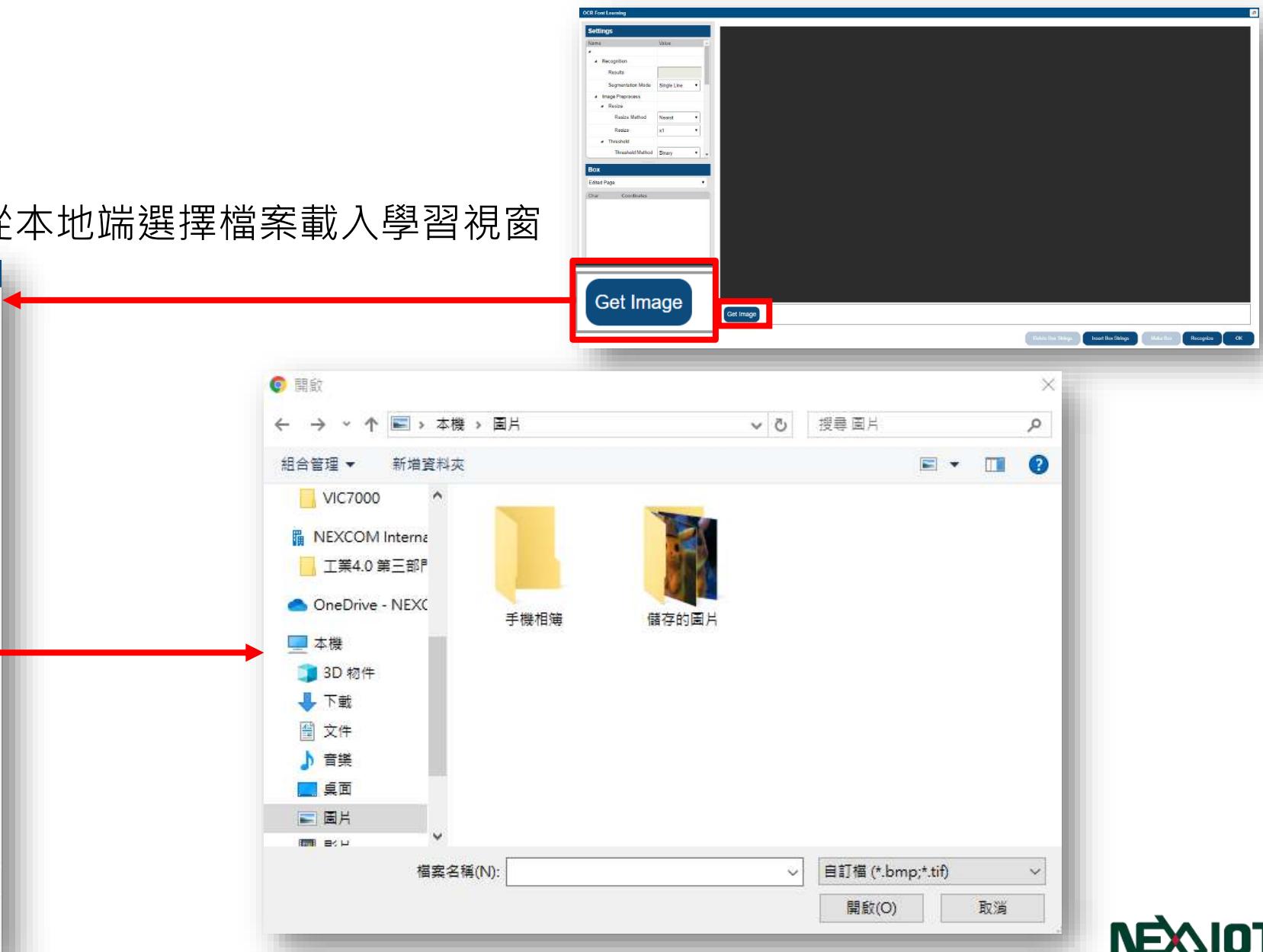
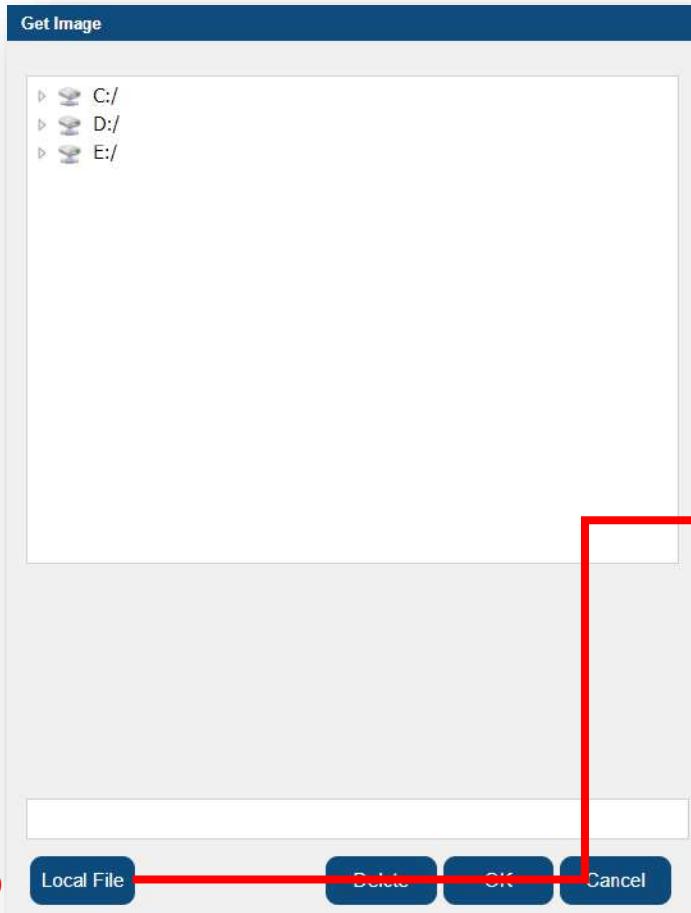
- 開啟字型學習視窗
- 載入圖像
  - 直接載入：取得的圖像直接載入學習視窗



- 一般

- 載入圖像

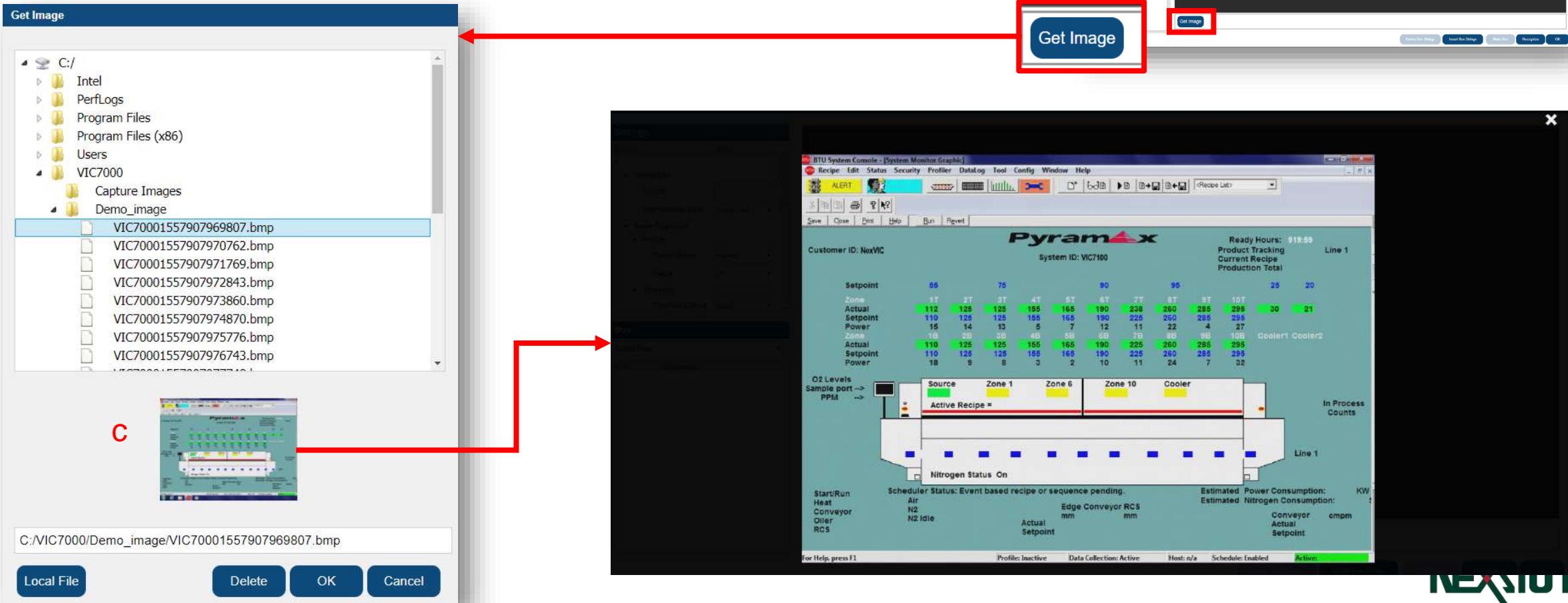
- 開啟本地圖檔：從本地端選擇檔案載入學習視窗



- 一般

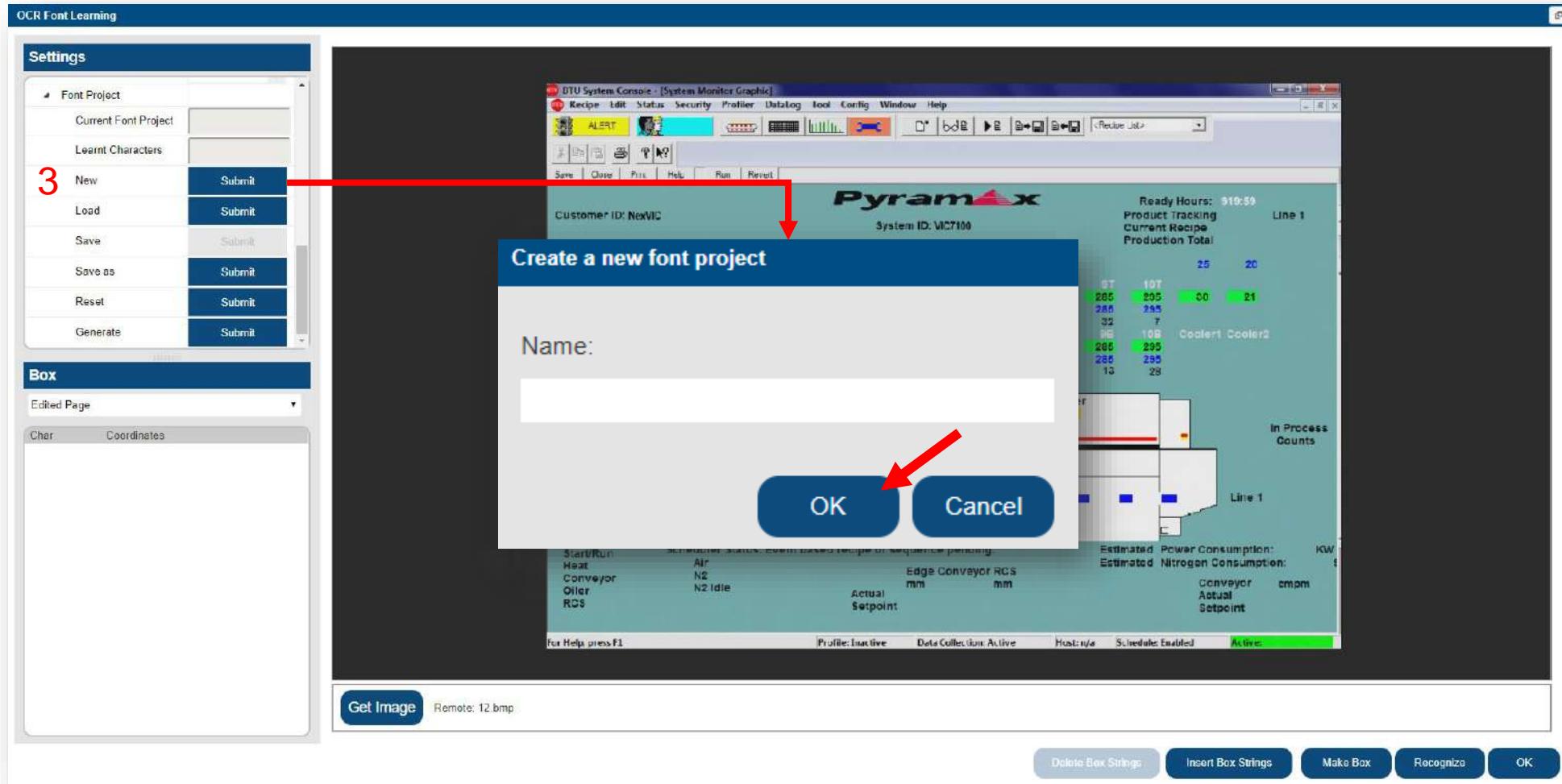
- 2. 載入圖像

- c. 開啟遠端圖檔：從 VIC7000 電腦選擇檔案載入學習視窗  
點擊預覽區塊，可放大圖像



- 一般

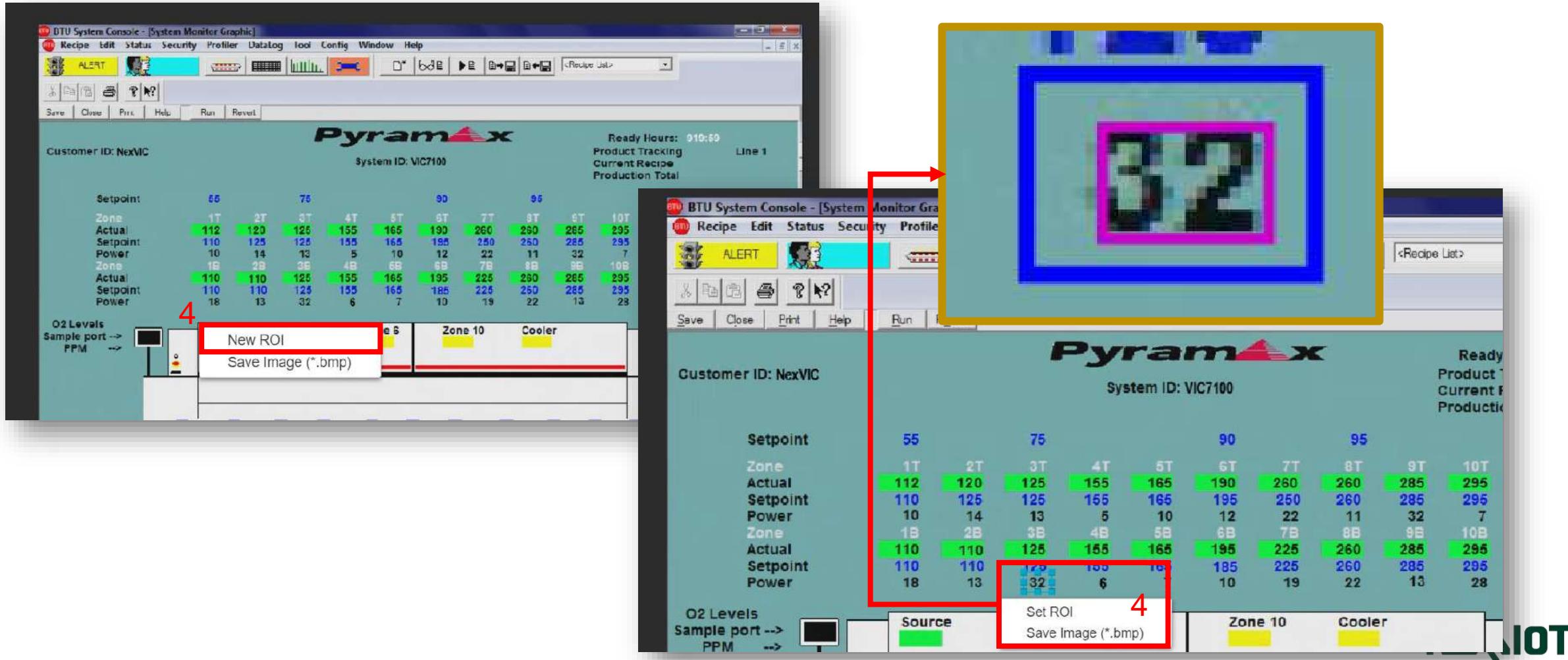
3. 建立字型專案：點擊新增按鈕 → 輸入字型專案的名稱，此名稱必須為唯一 → 點擊確定



# 字型學習

## ● 一般

4. 設定ROI與新增字元框：右鍵新增ROI → 調整大小和位置 → 右鍵設置ROI

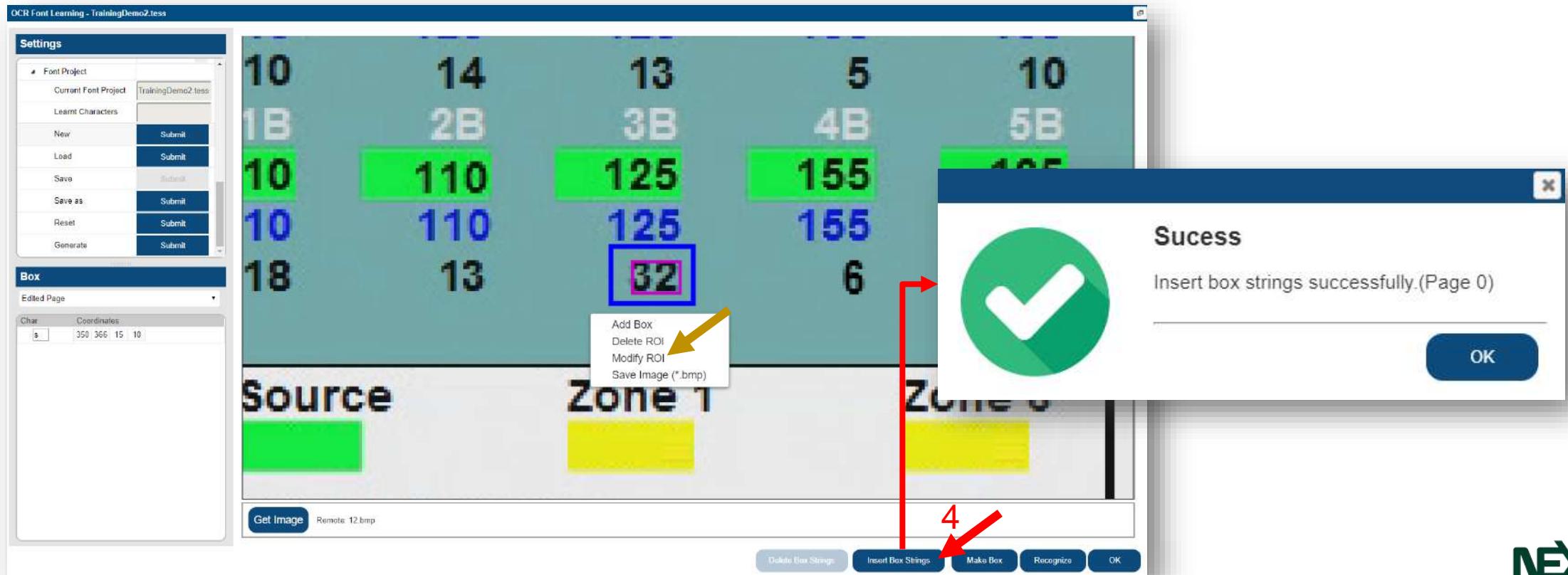


- 一般

4. 設定ROI與新增字元框：點擊加入字元框字串 → 新增其他字元框字串

右鍵修改 ROI → 調整ROI的大小和位置 → 右鍵設置ROI → 點擊加入字元框字串

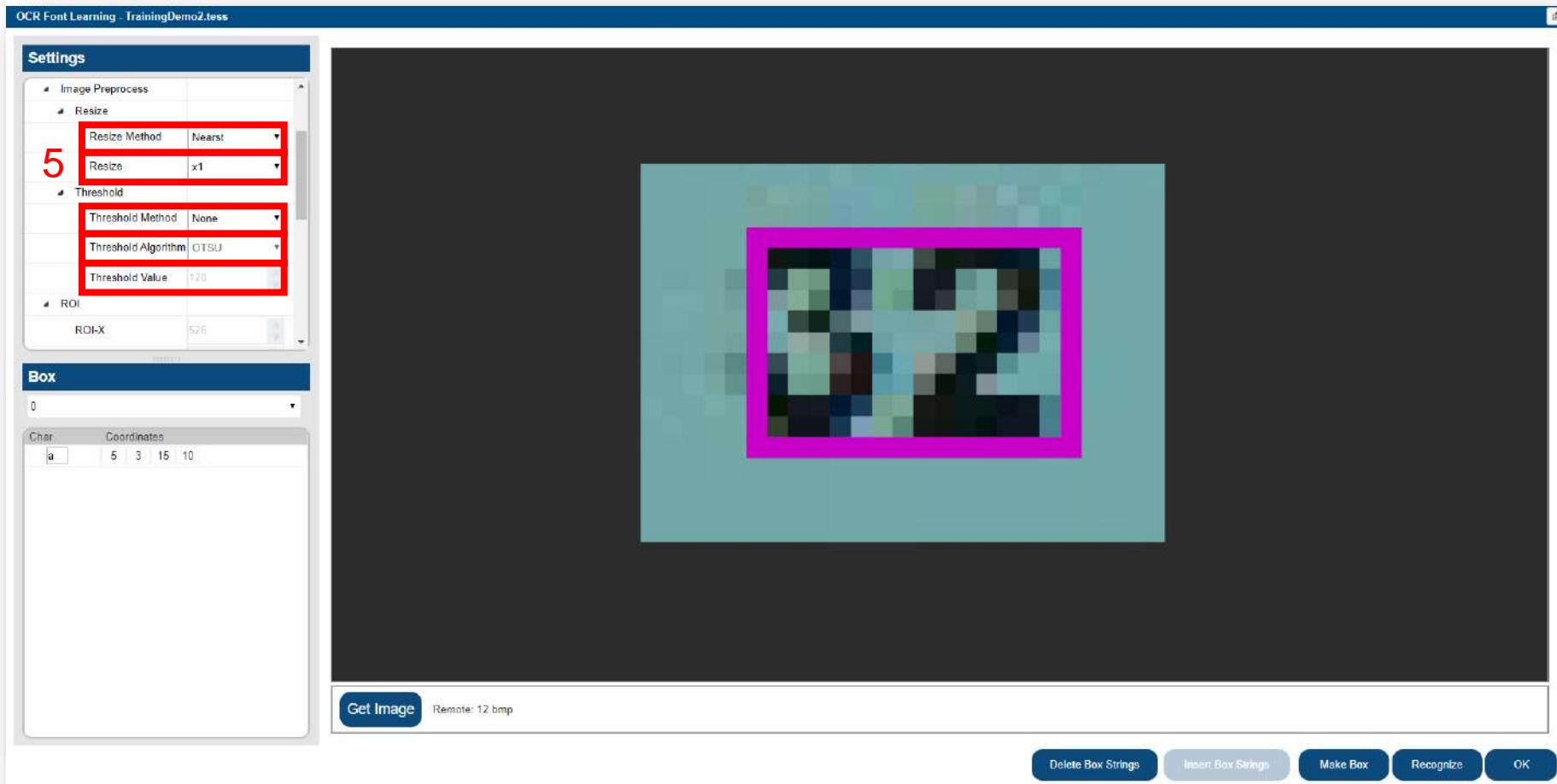
重複上述步驟直到所有要學習的字都有加入字元框字串



- 一般

5. 設定字元框參數：設定調整大小方式和倍數 → 設定二值化方法、二值化演算法和閥值

讓每個字都可以被單一個粉色框圈住



- 字元框參數

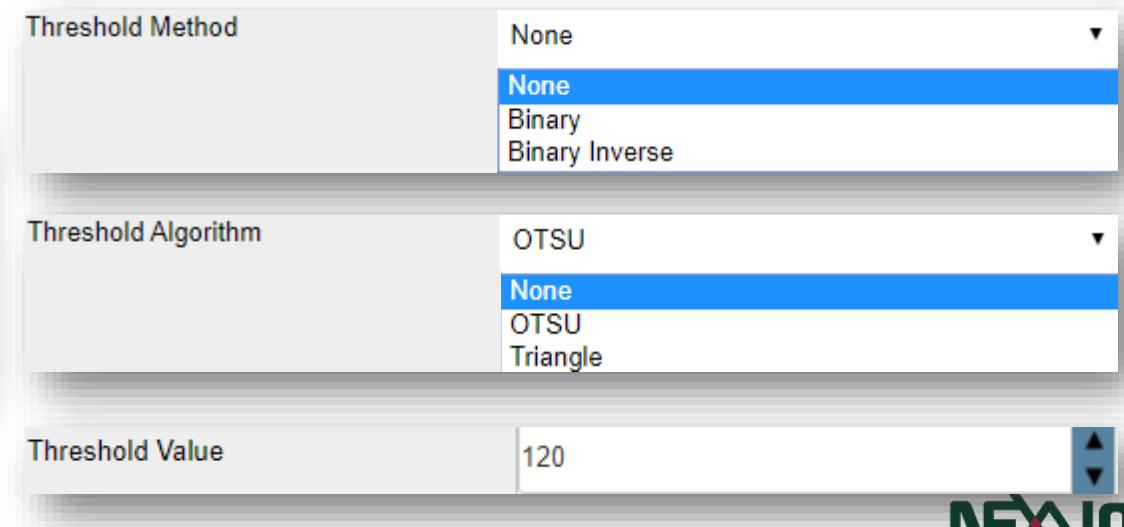
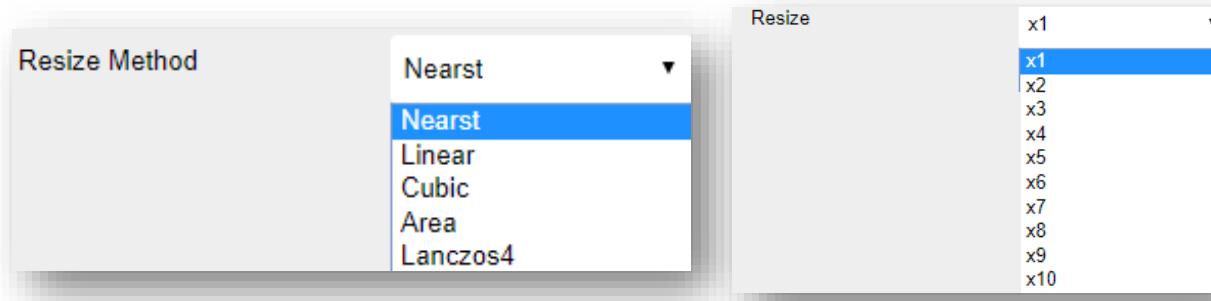
調整大小方式：最近鄰插值、雙線性插值、雙立方插值、區域取樣、Lanczos4

調整大小：x1 ~ x10

二值化方法：無、二分、二分反向

二值化演算法：無、OTSU、Triangle

閥值：若二值化方法選擇二分或二分反向 且 二值化演算法選擇無，就需要調整二值化閥值



- 一般

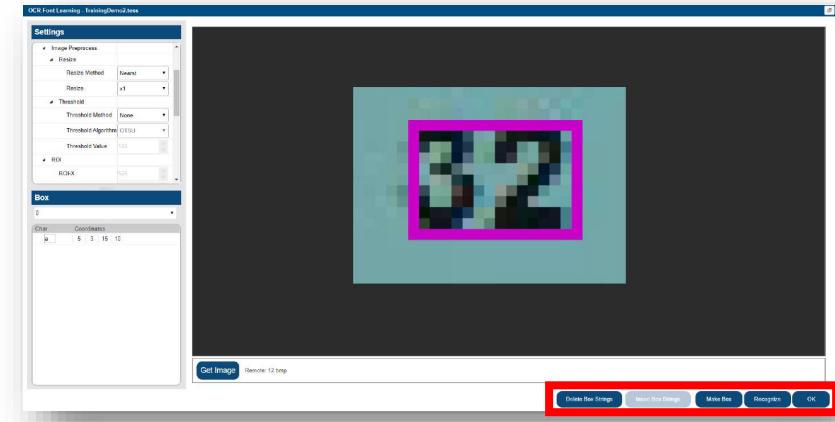
5. 設定字元框參數：

刪除字元框字串：刪除當前字元框

加入字元框字串：將ROI圈選的字元框字串加入字型專案

制作字元框：對當前畫面制作字元框

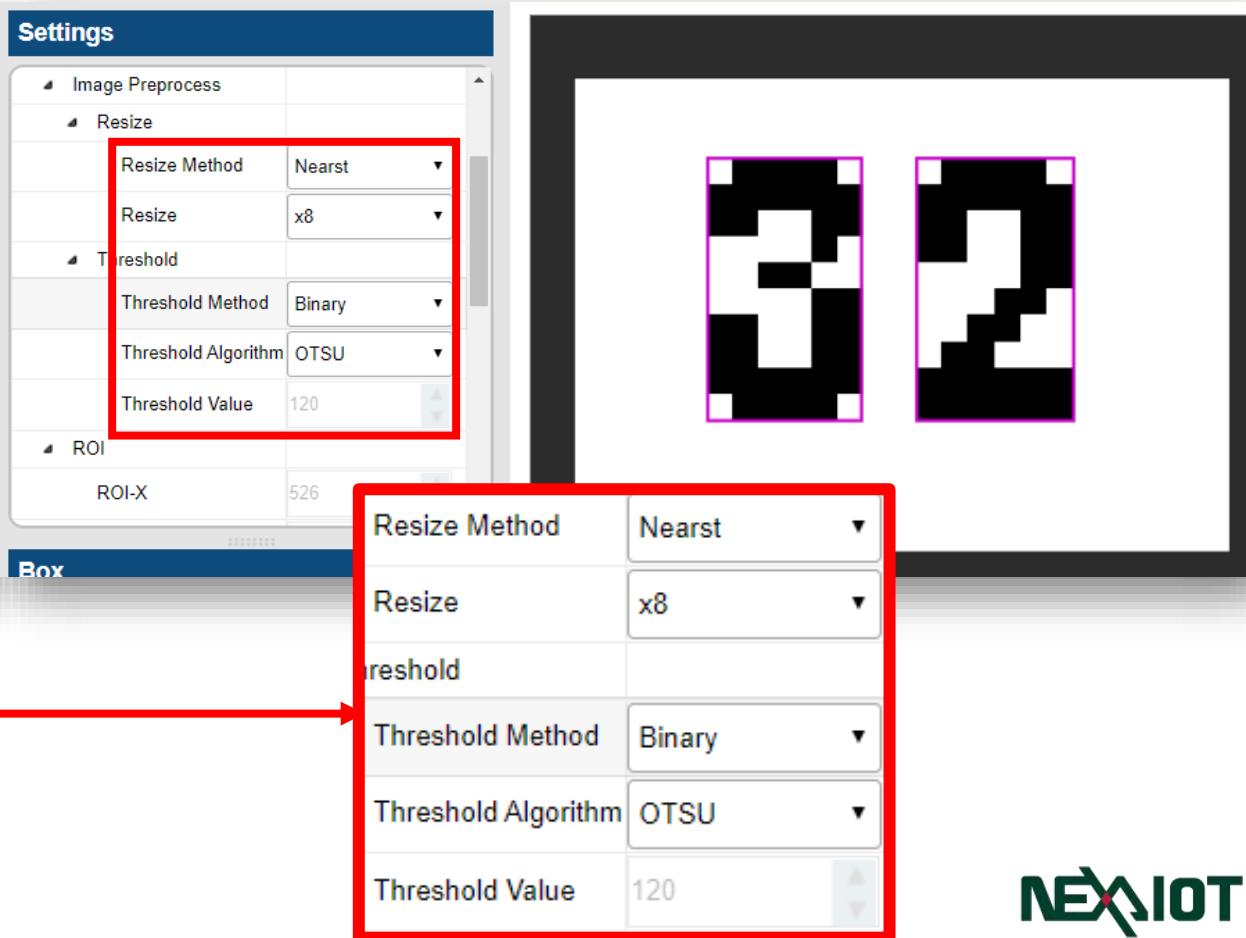
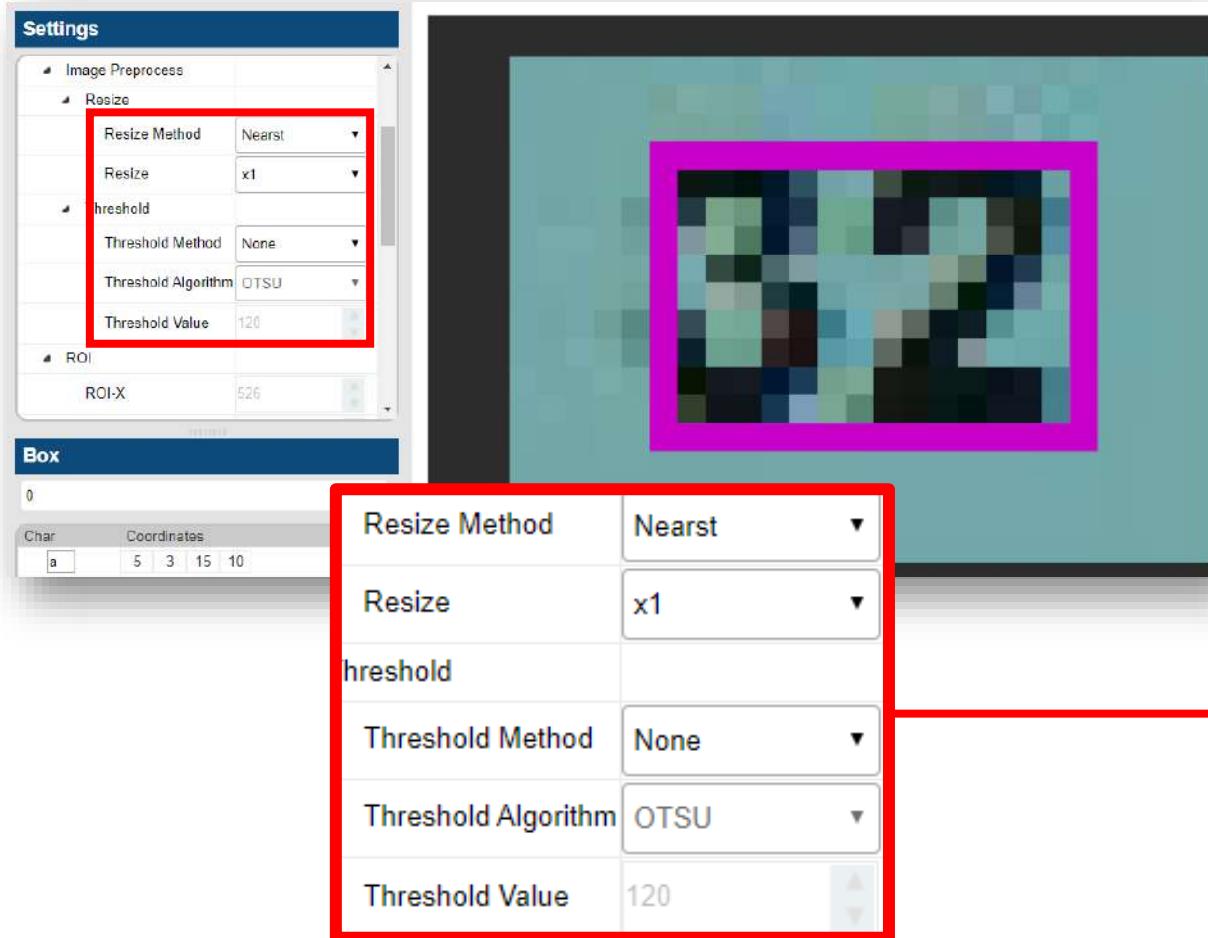
辨識：立即執行辨識，辨識結果會顯示在結果欄位中

[Delete Box Strings](#)[Insert Box Strings](#)[Make Box](#)[Recognize](#)

- 一般

5. 設定字元框參數：設定調整大小方式和倍數 → 設定二值化方法、二值化演算法和閥值

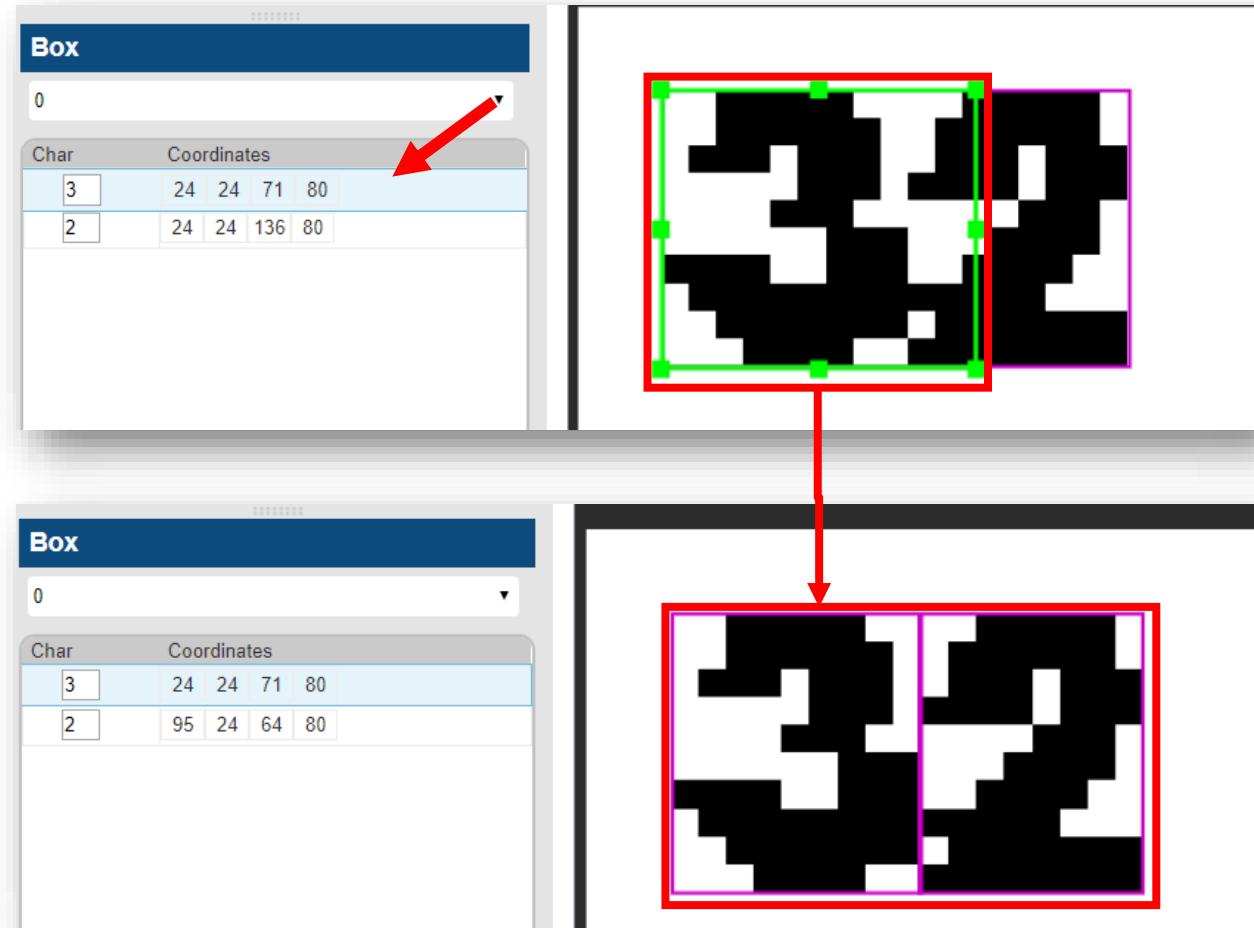
讓每個字都可以被單一個粉色框圈住



- 一般

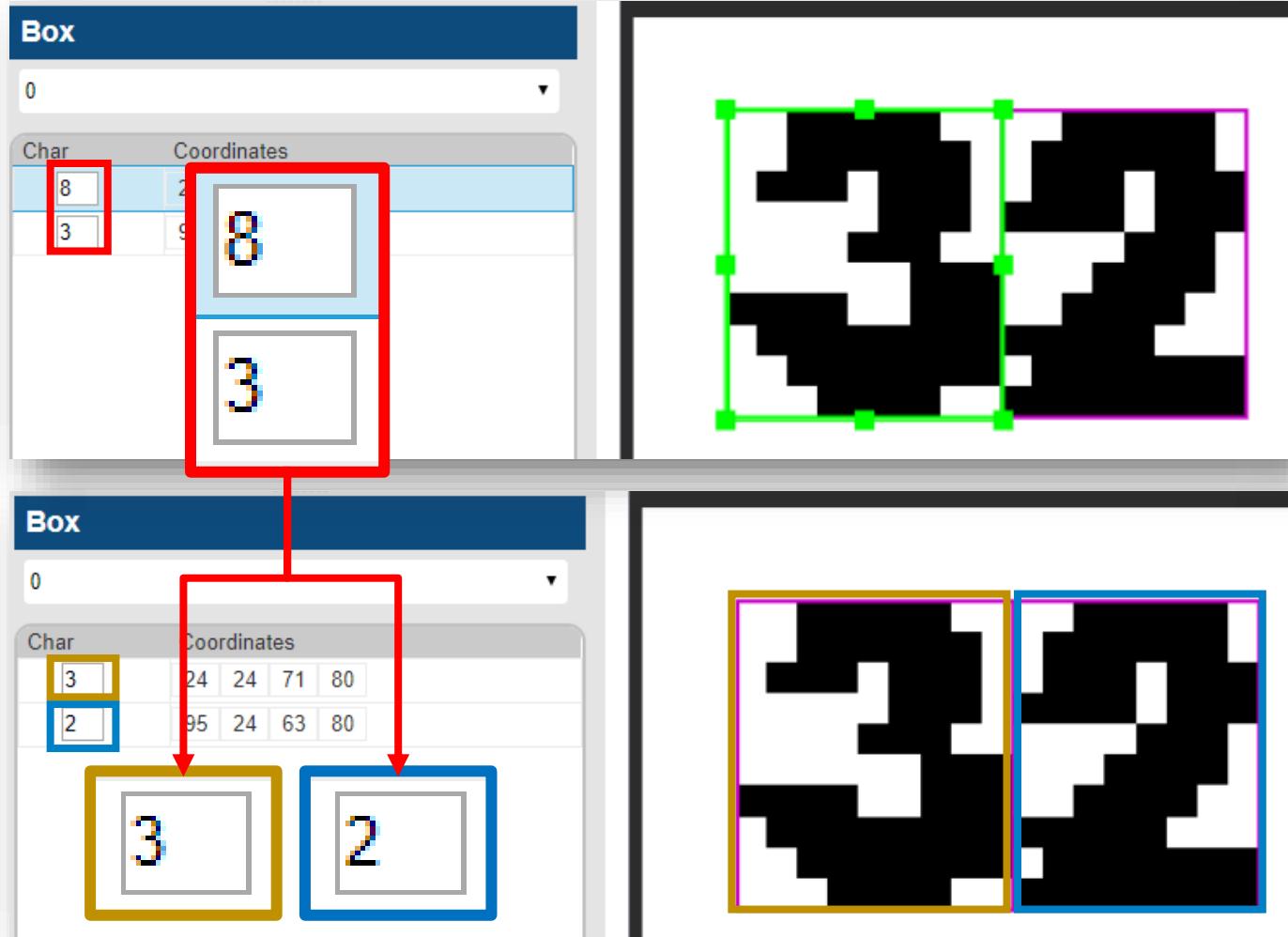
5. 設定字元框參數：點擊字元框內的字元，會顯示系統辨識的字元位置

若圈選範圍不符預期，就必須手動調整圈選位置，直接拖拉調整圈選位置



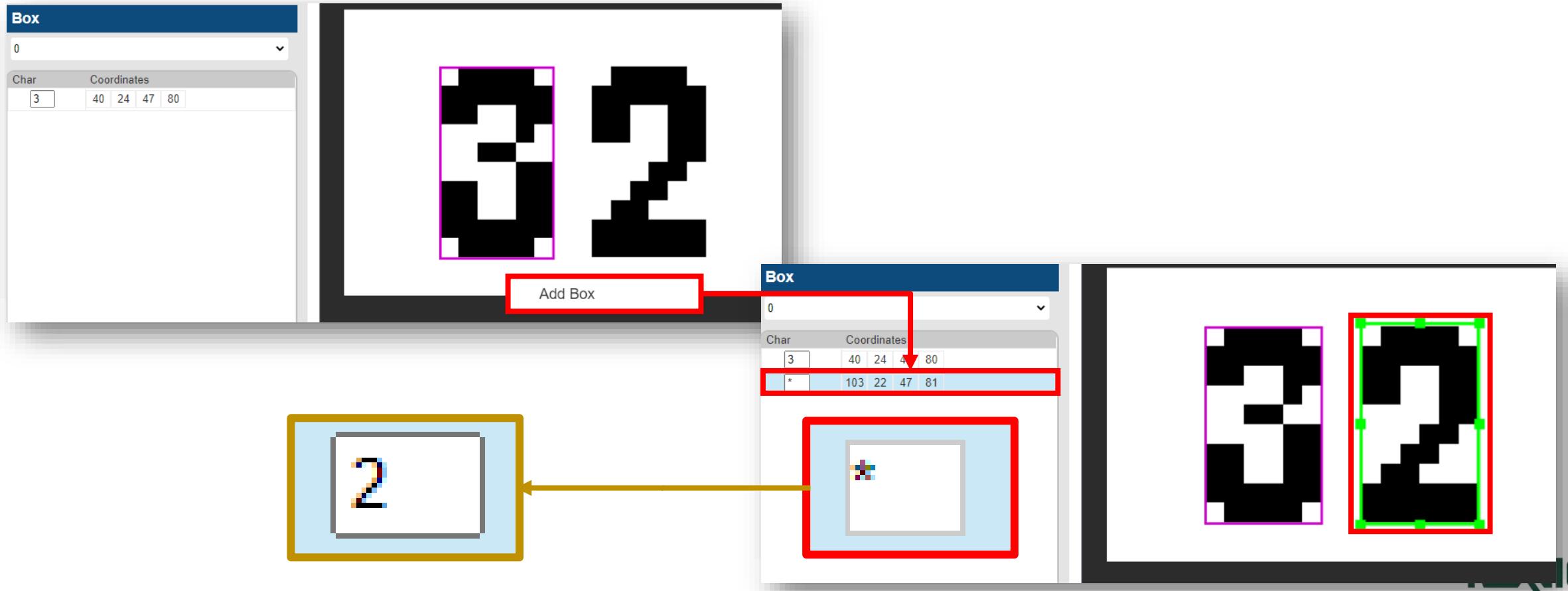
- 一般

5. 設定字元框參數：若系統辨識結果不符預期，修改字元框內的字元



- 一般

5. 設定字元框參數：若字元框內有少缺字元，在字元框內點擊右鍵加入字元框  
→ 調整字元框的大小和位置，圈選住要辨識的字元 → 修改字元框內的字元



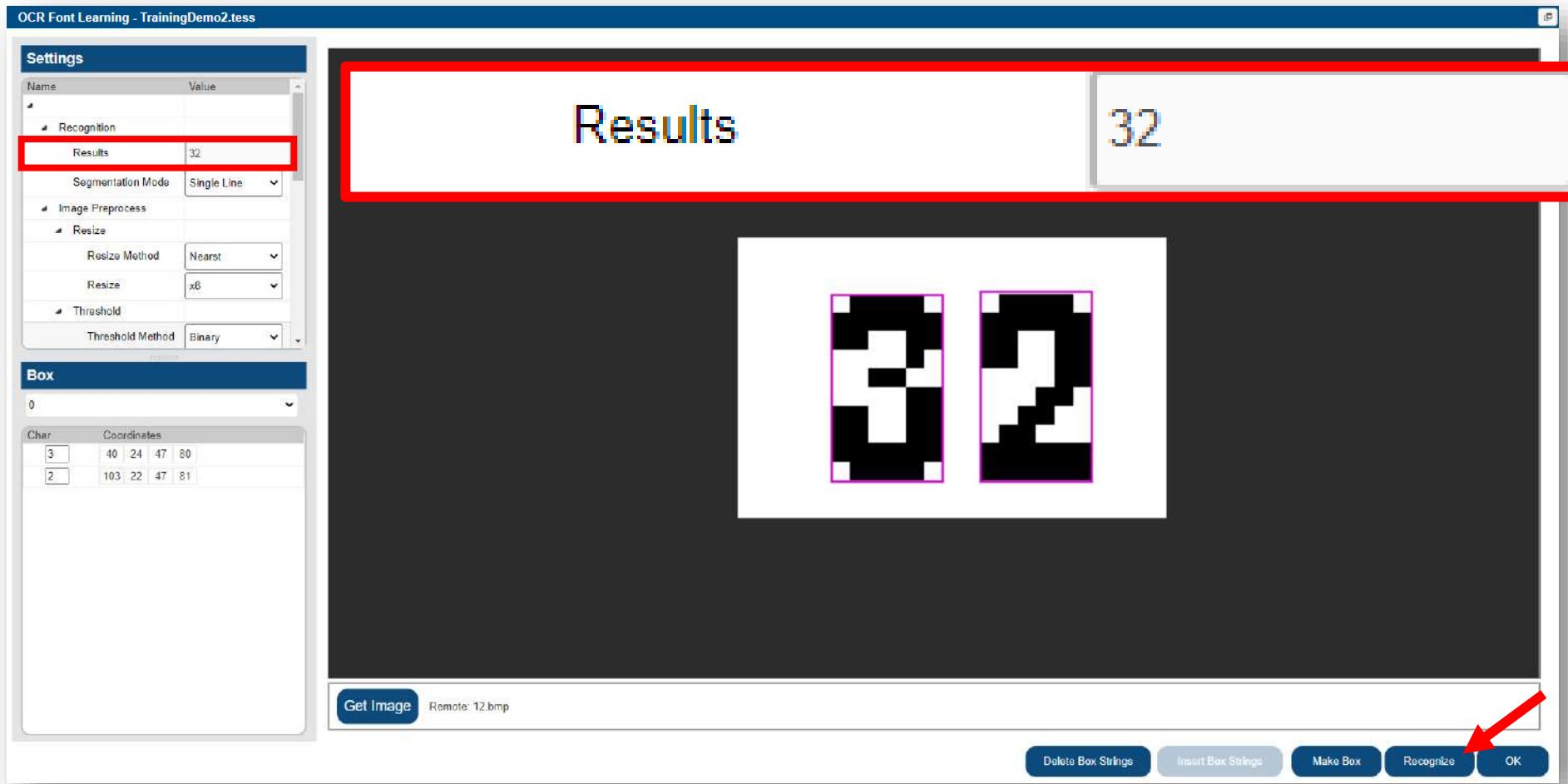
- 一般

5. 設定字元框參數：若字元框內有多餘的圈選區域，選擇該字元框 → 右鍵選擇刪除字元框



- 一般

5. 設定字元框參數：字元框設定結束後，點擊辨識 → 在結果欄位內確認辨識結果正確

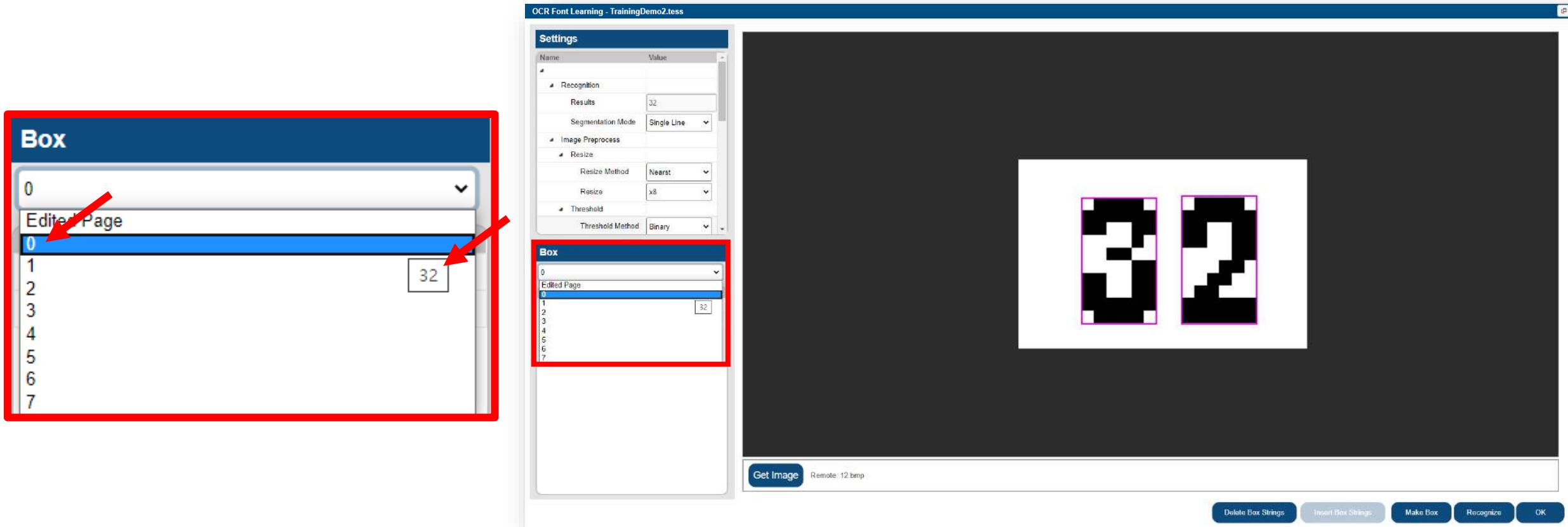


- 一般

- 5. 設定字元框參數：開啟字元框選單，依照上述步驟設定每一個字元框

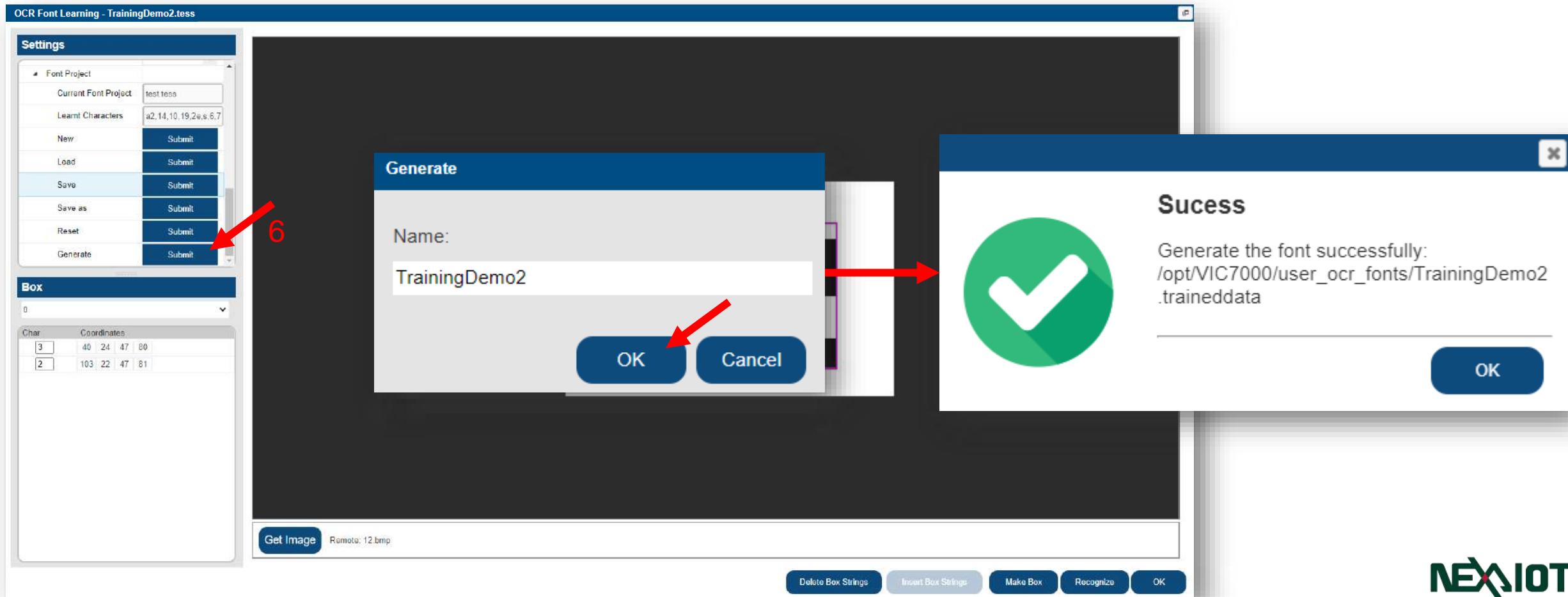
移動游標到字元框編號上，會顯示該字元框的字元內容

每個字元框的調整大小和二值化參數，盡量都要相同



- 一般

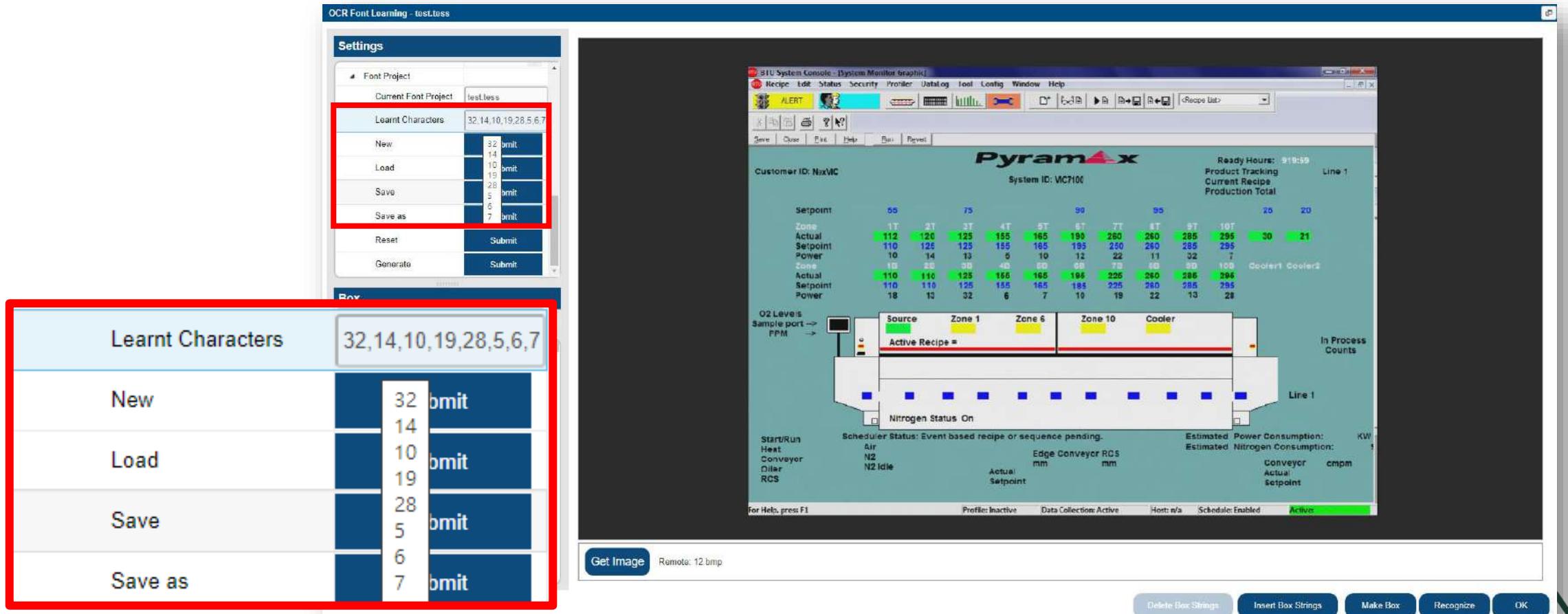
6. 生成訓練字型檔案：點擊產生字型 → 若必要，修改檔名 → 點擊產生字型 → 點擊確定 → 確認生成字型檔是否成功



# 字型學習

- 一般

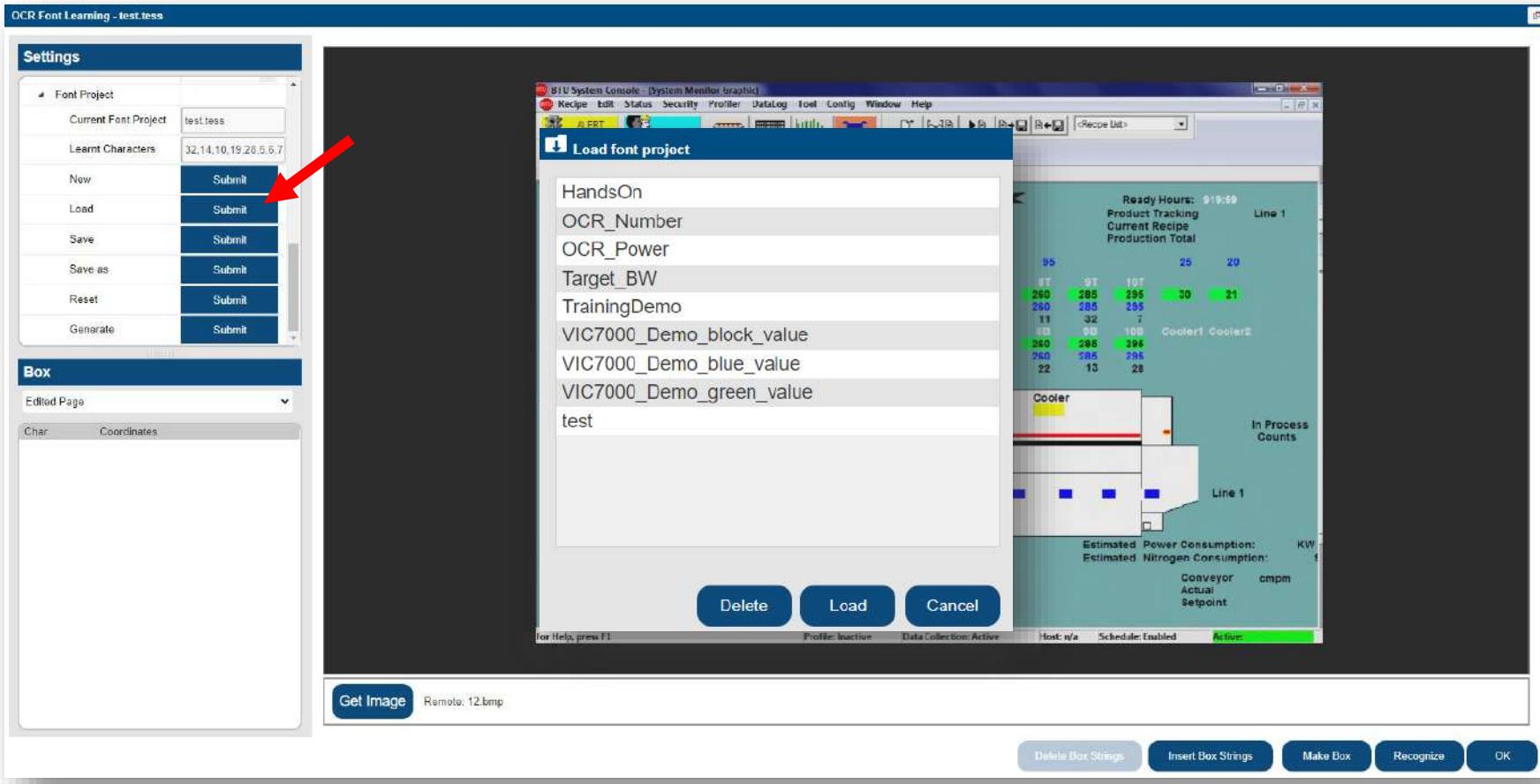
- 再次學習：儲存字型專案或生成字型檔後，把滑鼠游標移到已學習過字元欄位上，可以看到該字型專案學習過的字元。若需要學習更多字元，可以讓字型專案再次學習



- 一般

- 再次學習**：載入字型專案再次學習

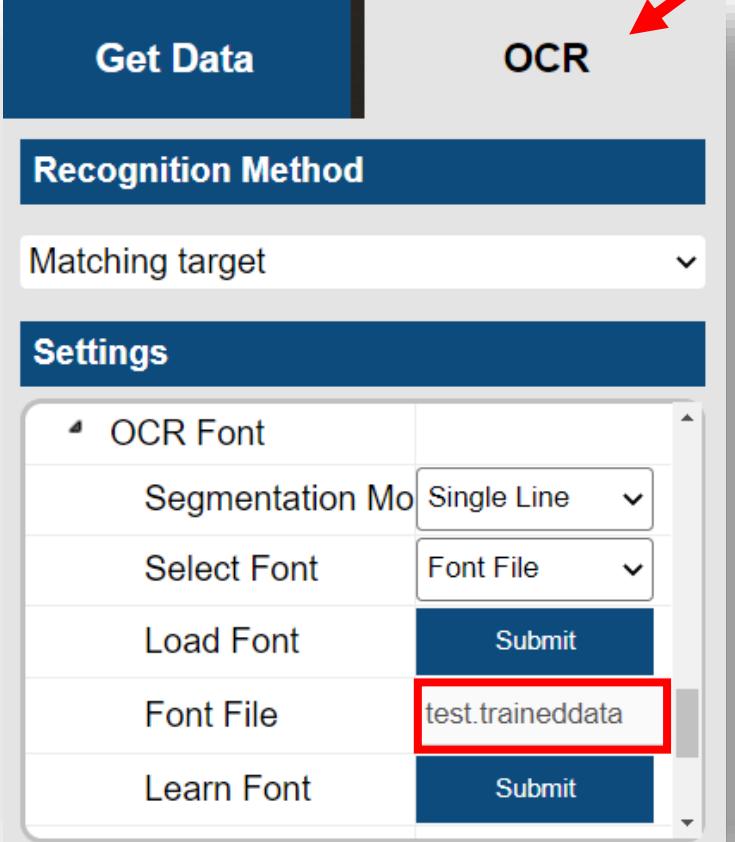
1. 載入按鈕：用字型學習視窗內的載入按鈕



- 一般

- 再次學習：載入字型專案再次學習

2. 直接載入：直接載入和文字辨識頁面設定的字型檔同名的字型專案



Get Data      OCR

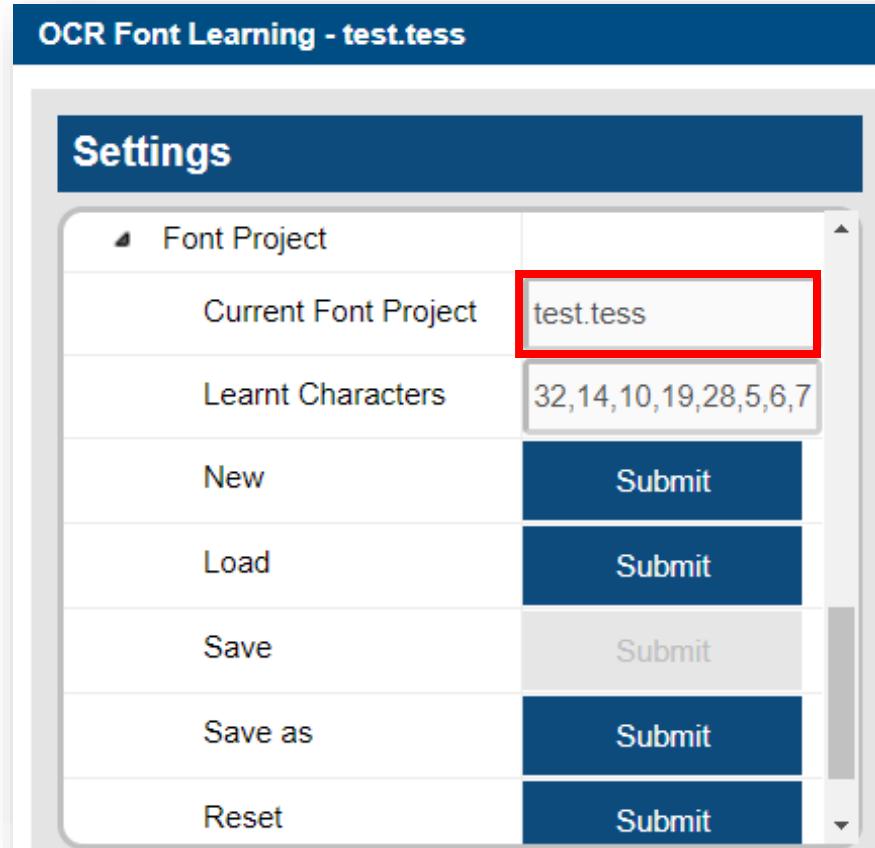
### Recognition Method

Matching target

### Settings

OCR Font

- Segmentation Mode: Single Line
- Select Font: Font File
- Load Font: Submit
- Font File: **test.traineddata**
- Learn Font: Submit



### OCR Font Learning - test.tess

### Settings

Font Project

Current Font Project	<b>test.tess</b>
Learnt Characters	32,14,10,19,28,5,6,7

New: Submit

Load: Submit

Save: Submit

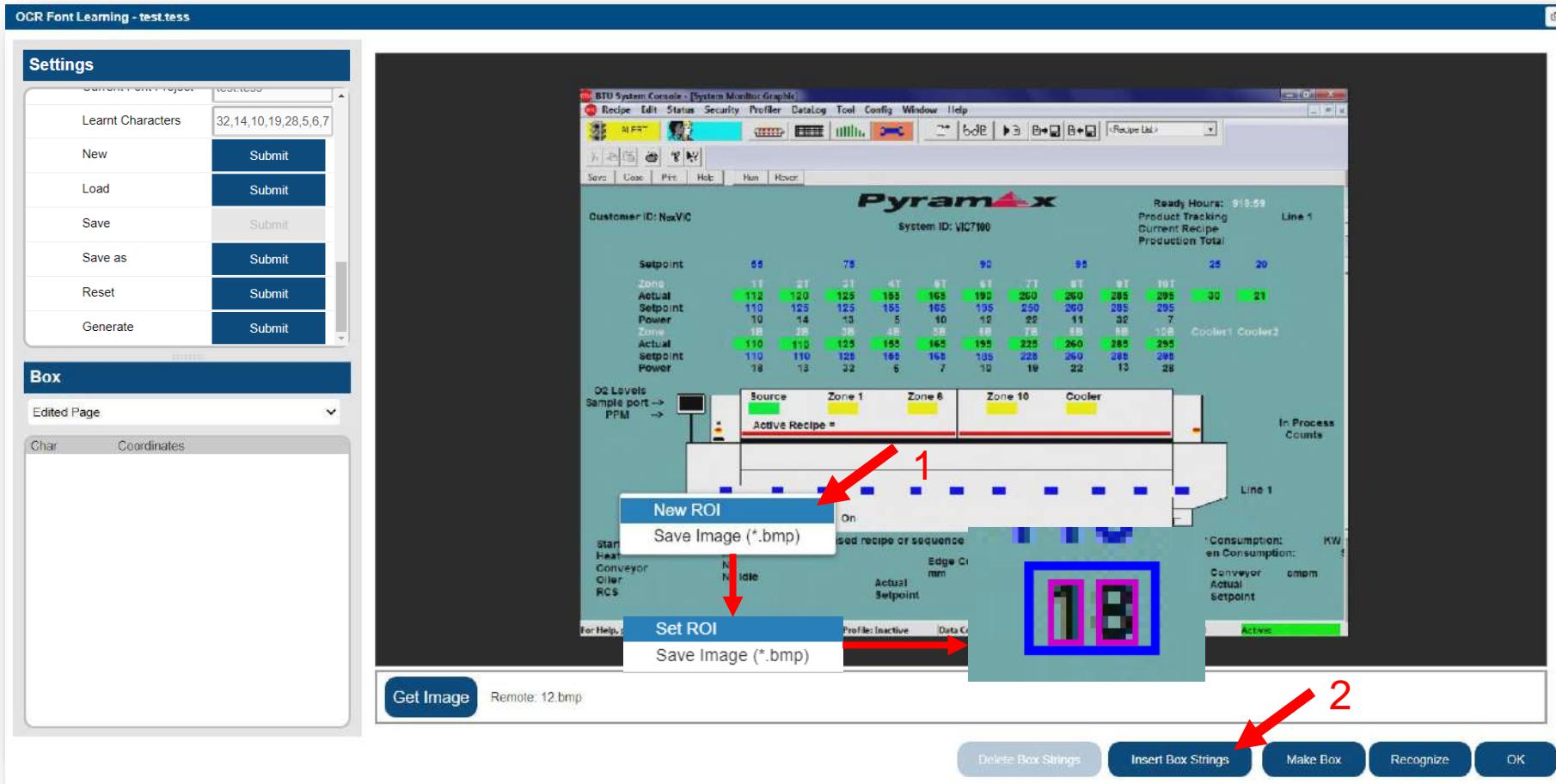
Save as: Submit

Reset: Submit

# 字型學習

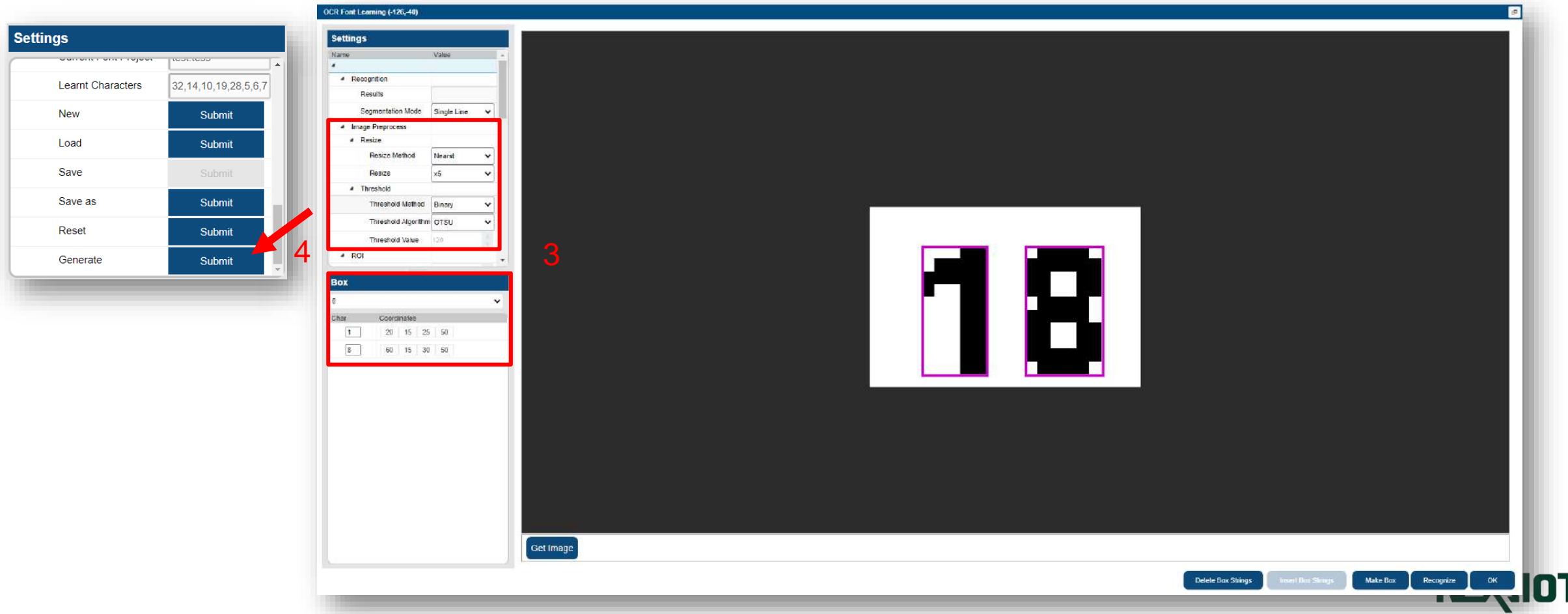
- 一般

- 再次學習：在編輯頁面新增和設置ROI → 加入字元框字串 → 編輯字元框字串 → 生成訓練字型檔案



- 一般

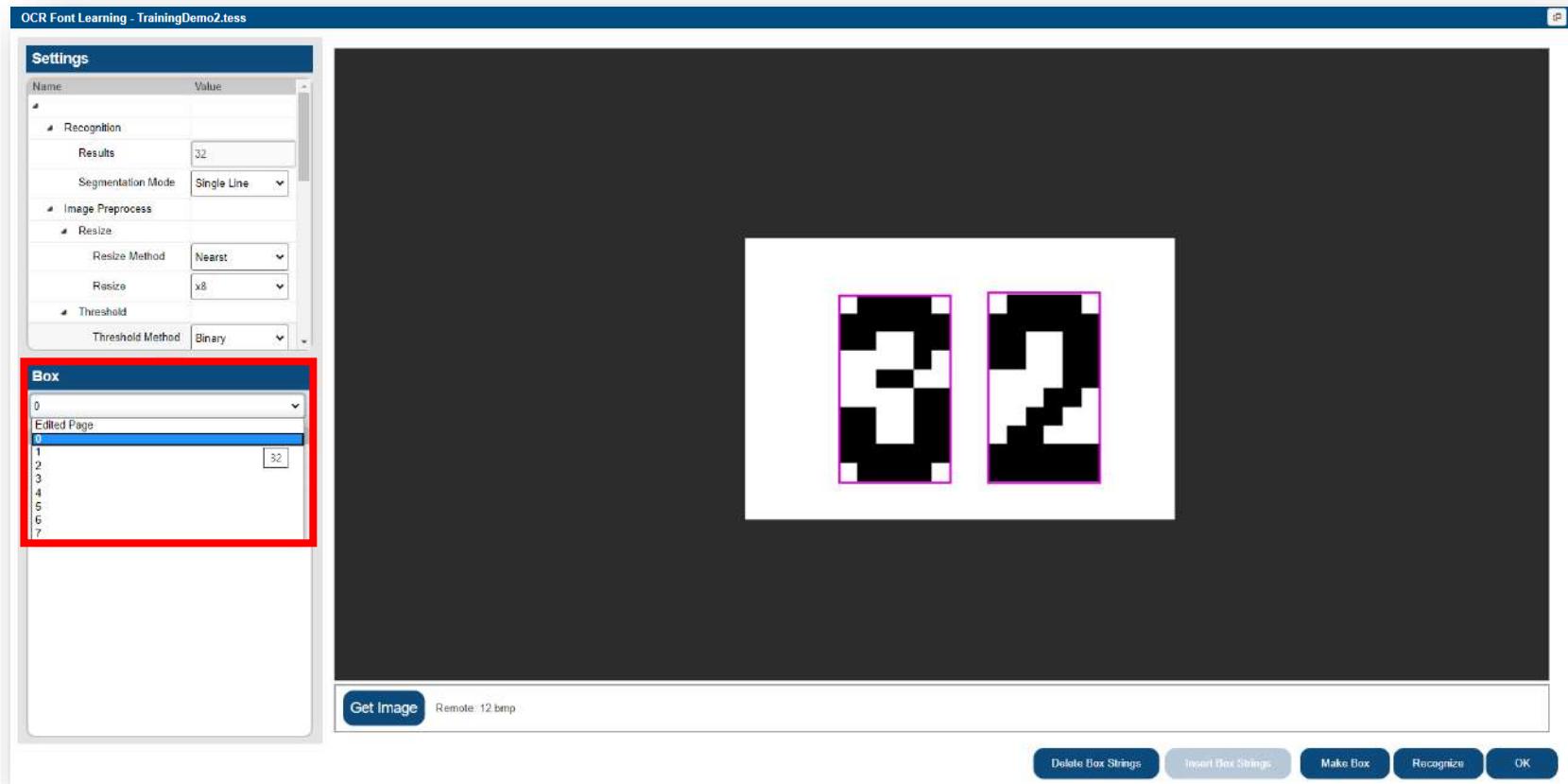
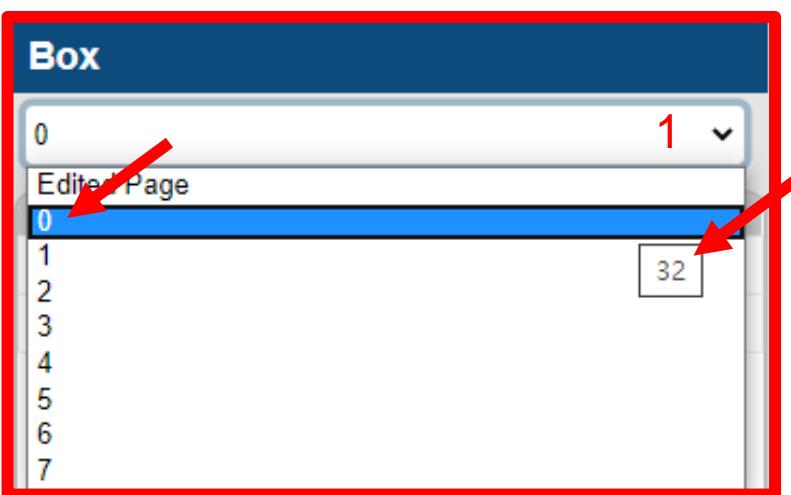
- 再次學習：在編輯頁面新增和設置ROI → 加入字元框字串 → 編輯字元框字串 → 生成訓練字型檔案



- 一般

- 編輯字型**：開啟字元框選單，移動游標到字元框編號上，會顯示該字元框的字元內容

選擇要編輯的字元框



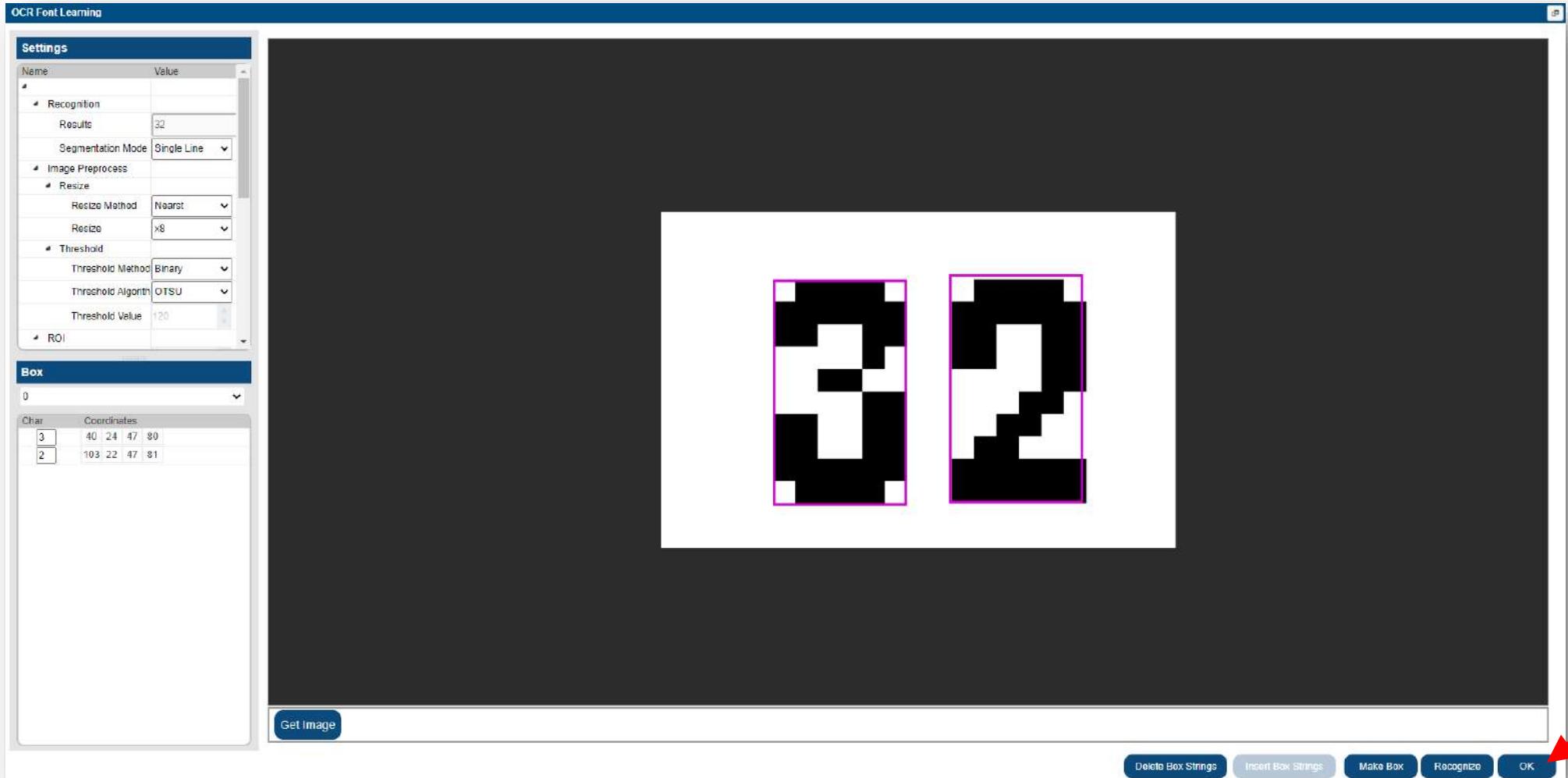
- 一般

- 編輯字型**：可以在字元框內修改字元 → 點擊辨識，確認辨識結果 → 點擊產生字型，生成訓練字型檔案



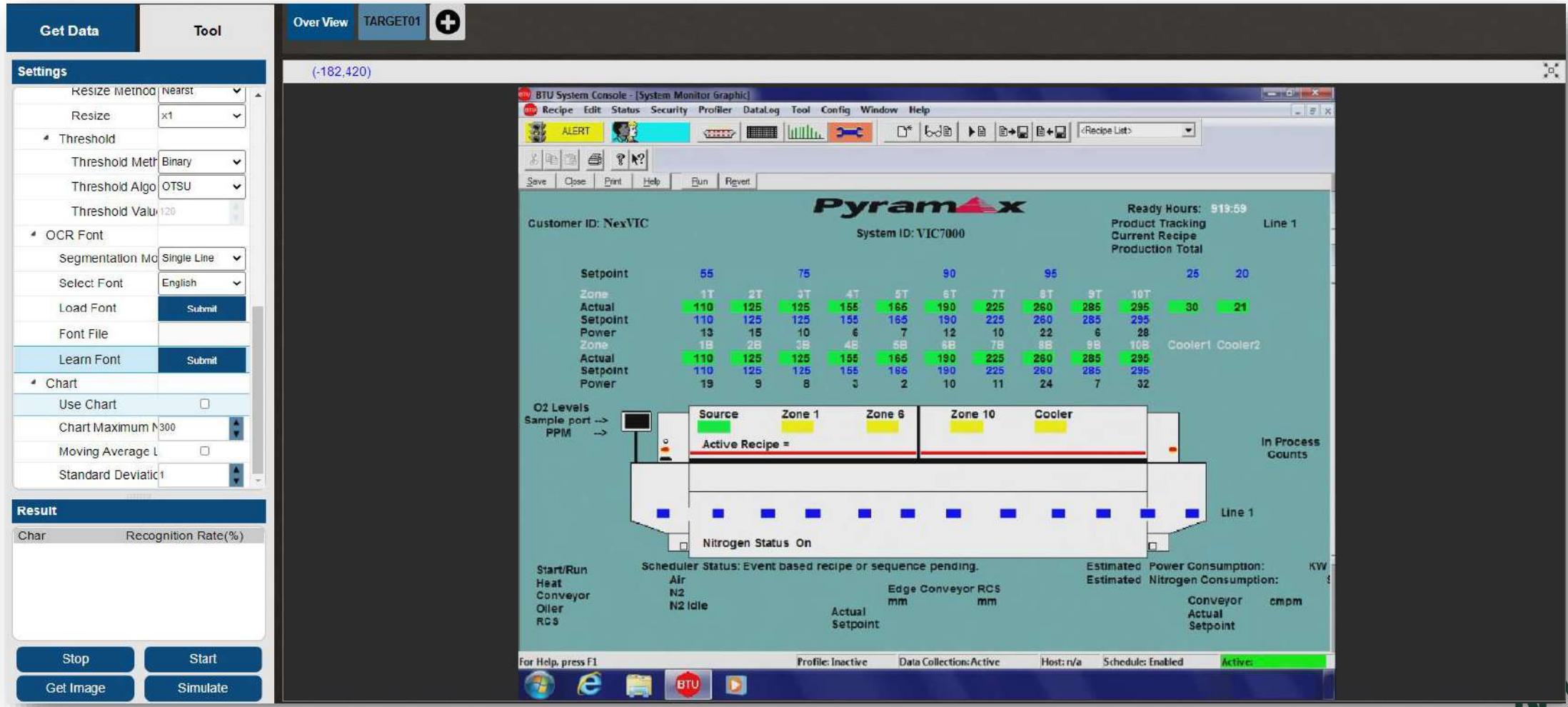
- 一般

7. 學習結束：點擊確定，離開字型學習視窗



- 一般

## 7. 學習結束：回到功能頁面



# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

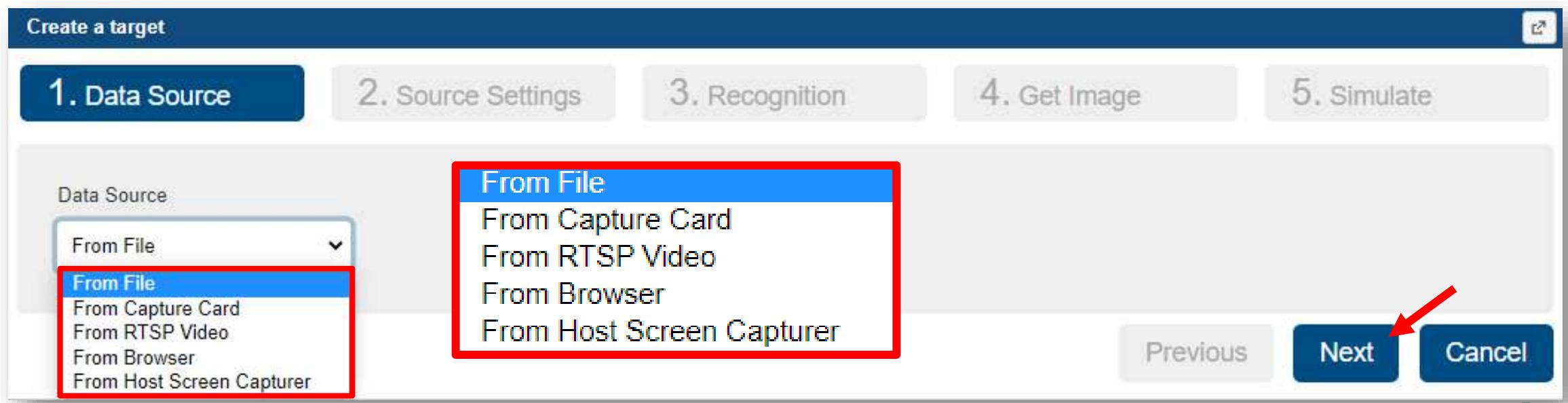
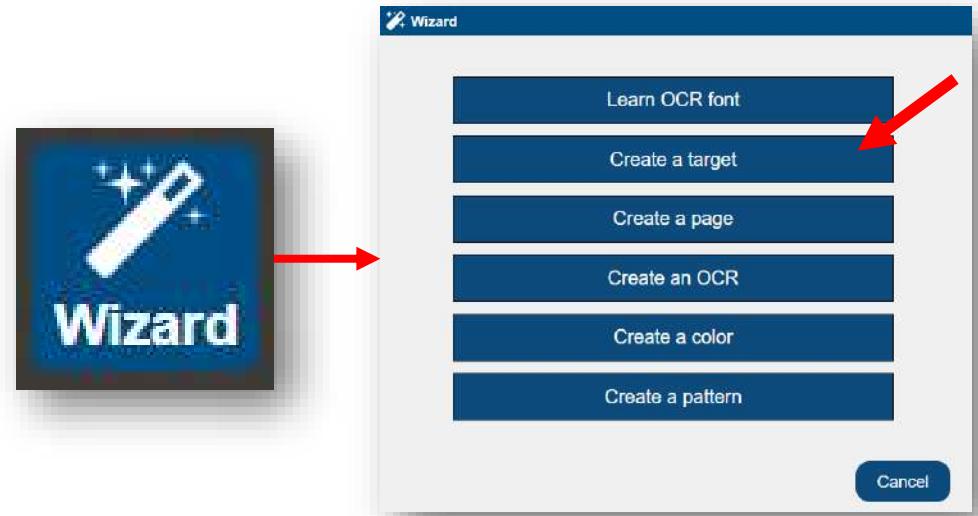
# 目標值

- 精靈：新增目標值

## 1. 影像來源

來自檔案、來自擷取卡、來自 RTSP 影像

來自瀏覽器、來自本機畫面擷取



# 目標值

- 精靈：新增目標值

## 2. 影像來源設置

來自檔案：圖檔路徑、讀取時間間隔

來自擷取卡：輸入訊號類型、擷取頻率(fps))

The screenshot shows the 'Create a target' software interface with two tabs: '1. Data Source' and '2. Source Settings'. The '2. Source Settings' tab is active. In the '1. Data Source' tab, there is an 'Image Folder' input field containing 'C:/VIC7000' and a 'Submit' button. A red arrow points to the 'Submit' button. In the '2. Source Settings' tab, there is a 'Video Signal' input field containing 'false', a 'Video Input' dropdown menu currently showing 'DVI\_A (RGB / VGA)', and a 'Capture Frame Rate' input field containing '10.00'. A red arrow points to the 'Capture Frame Rate' input field. In the bottom right corner of the interface, there are three buttons: 'Previous', 'Next', and 'Cancel'. A red arrow points to the 'Next' button.

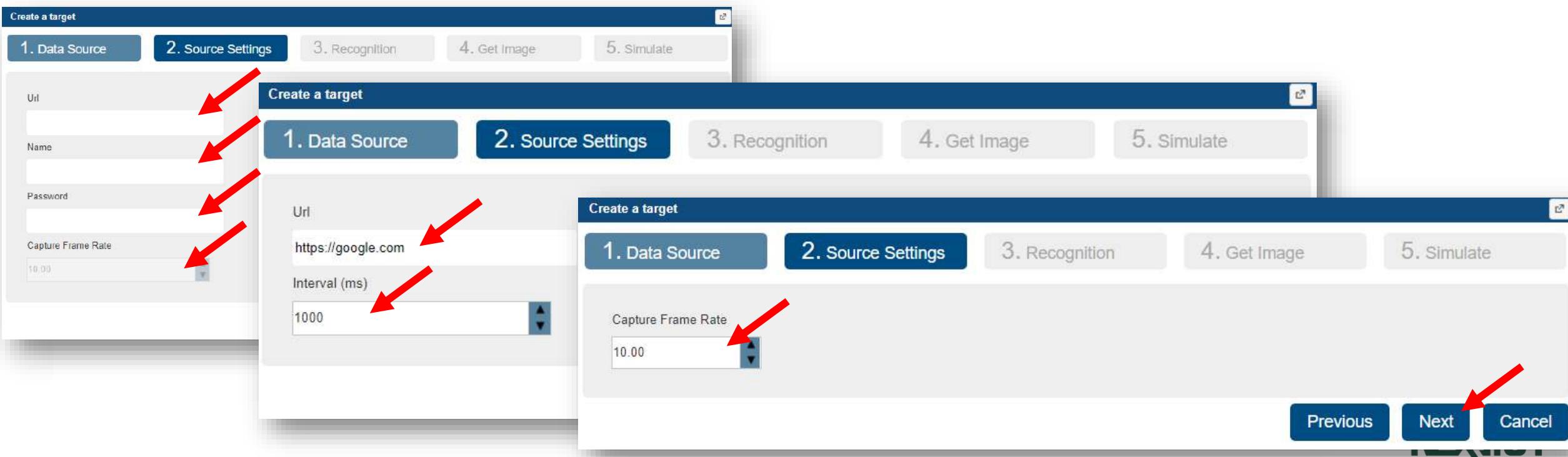
- 精靈：新增目標值

## 2. 影像來源設置

來自 RTSP 影像：RTSP 來源網址、名稱、密碼、擷取頻率

來自瀏覽器：網址、讀取時間間隔

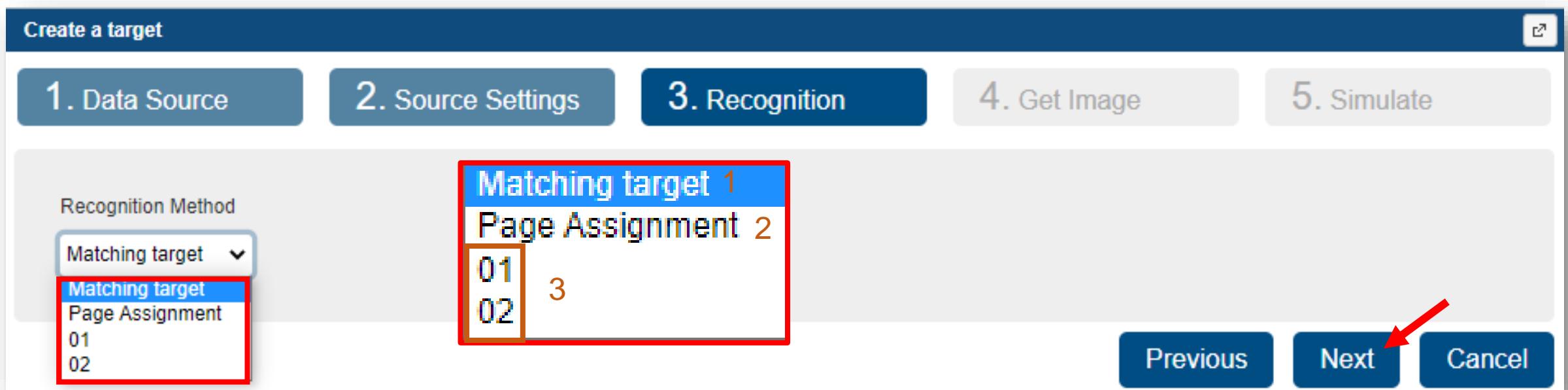
來自本機畫面擷取：擷取頻率(fps)



- 精靈：新增目標值

## 3. 辨識

1. 比對目標值：依據目標值比對條件自動選擇要使用的頁面
2. 頁面指定：透過腳本或外部連線選擇要使用的頁面
3. 選擇頁面：直接選擇要使用的頁面



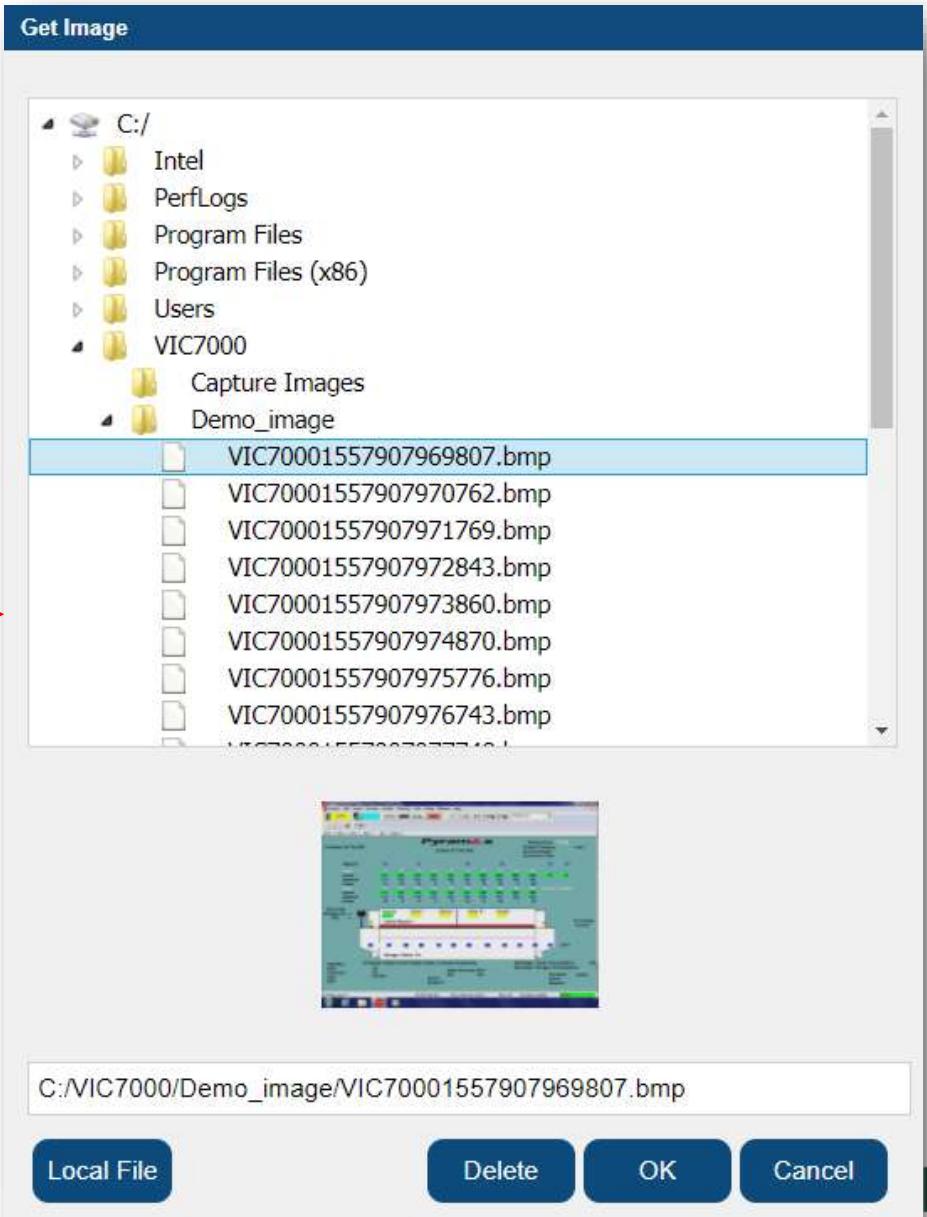
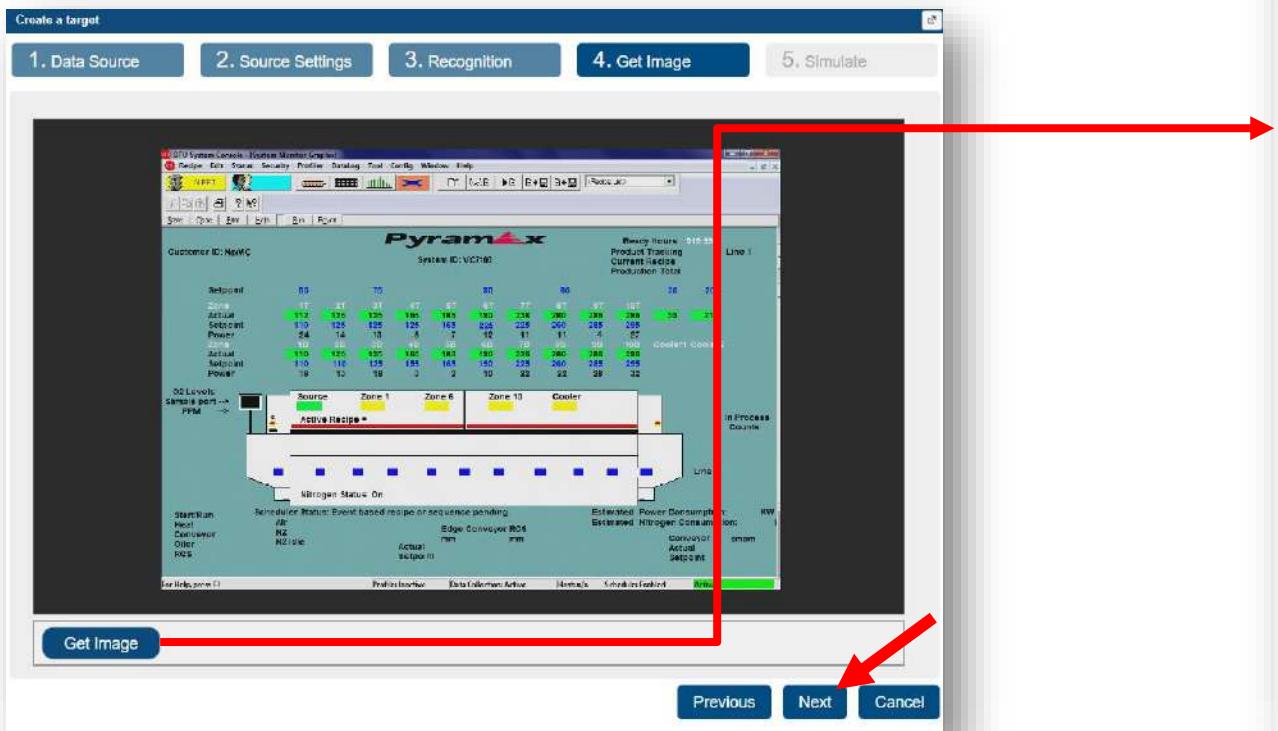
- 精靈：新增目標值

## 4. 取得圖像

來自檔案：讀取本機或遠端的圖像

來自擷取卡、來自 RTSP 影像、來自瀏覽器、

來自本機畫面擷取：從取像來源取得一張圖像

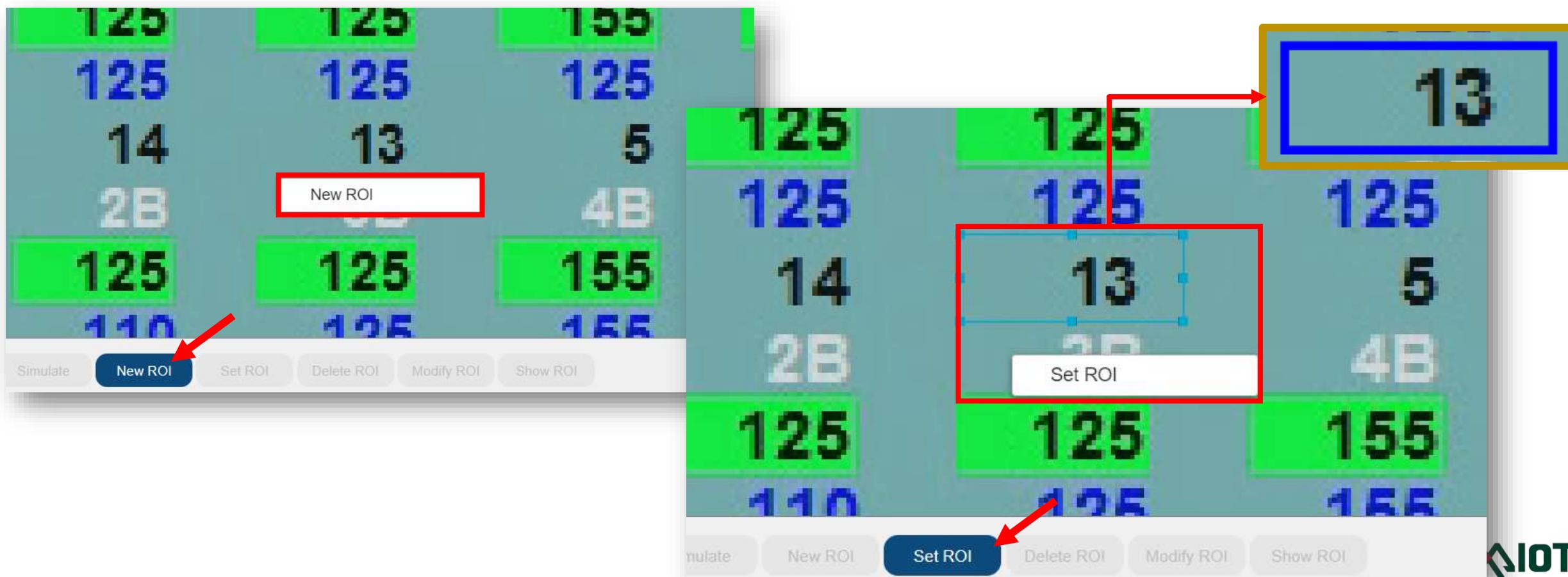


# 目標值

- 精靈：新增目標值

## 5. 模擬

右鍵新增ROI or 點擊 新增ROI → 調整大小和位置 → 右鍵設置ROI or 點擊設置ROI

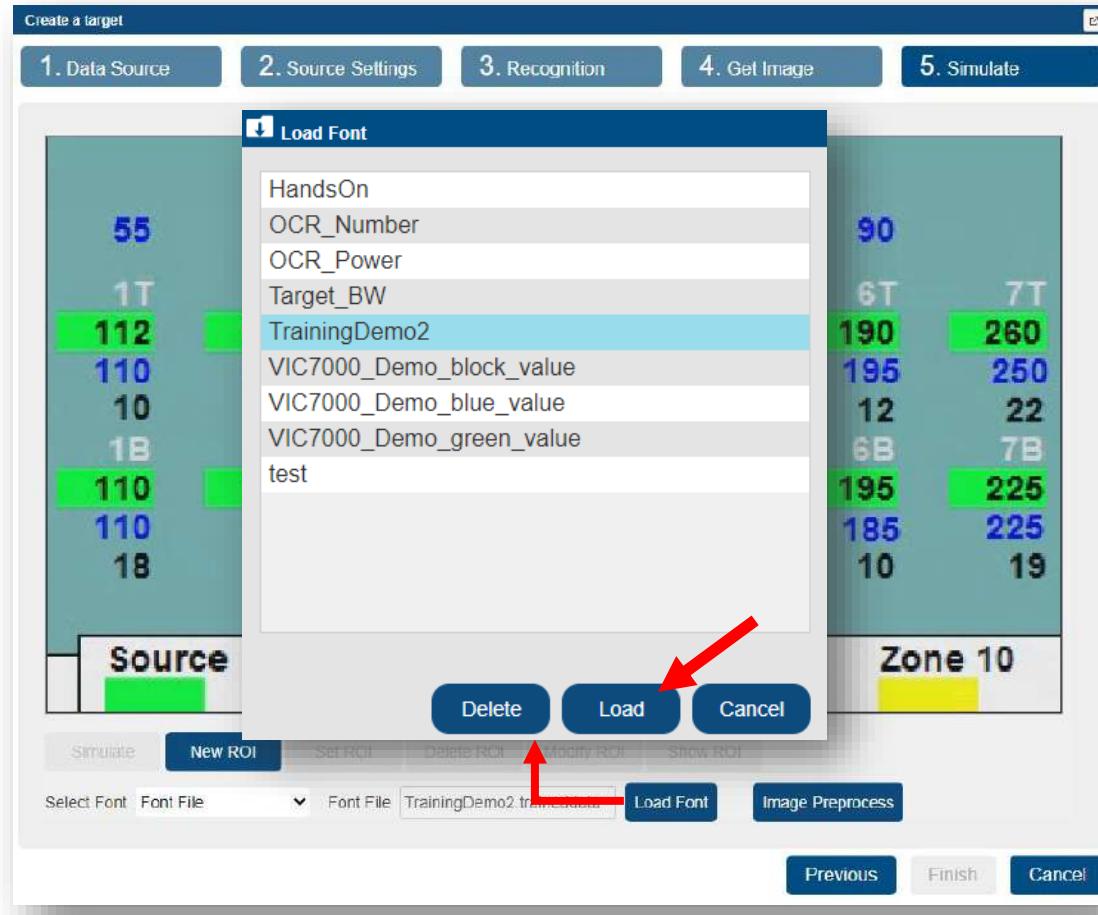


# 目標值

- 精靈：新增目標值

## 5. 模擬

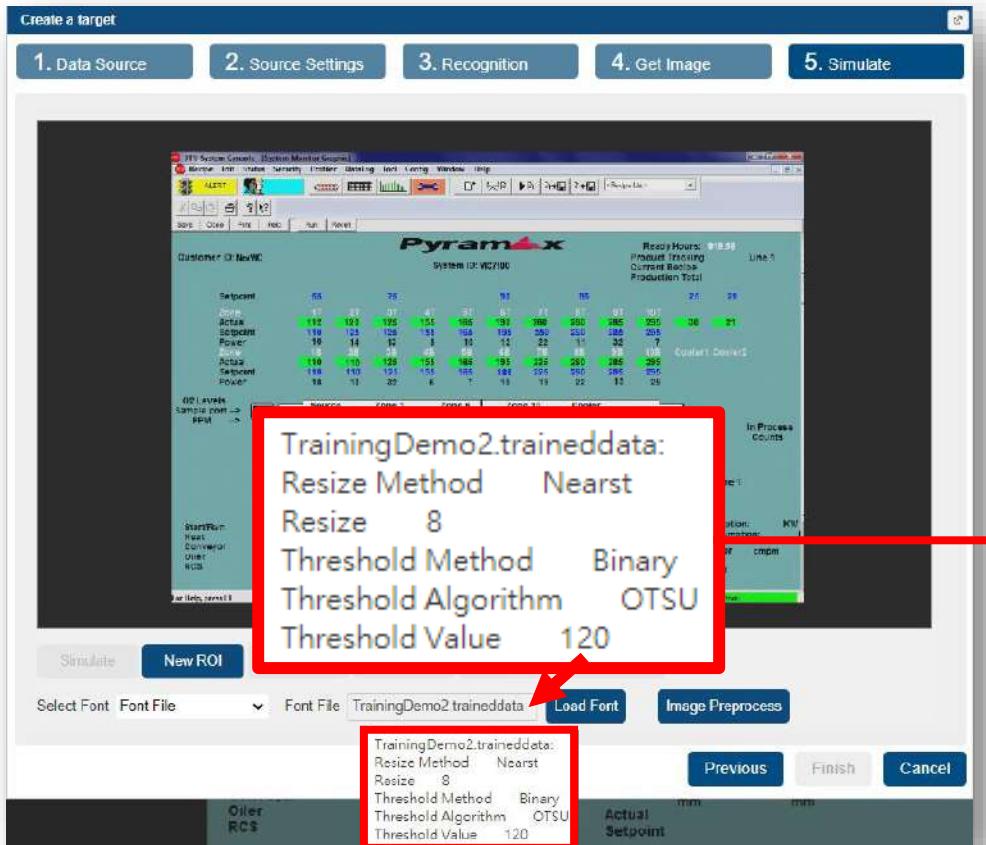
載入字型 → 選擇要使用的字型檔 → 點擊載入



- 精靈：新增目標值

## 5. 模擬

影像前處理 → 設定影像前處理參數，要和字型專案內的前處理參數相同 → 點擊確定  
將游標移到字型檔欄位上，會顯示該字型檔使用的前處理參數

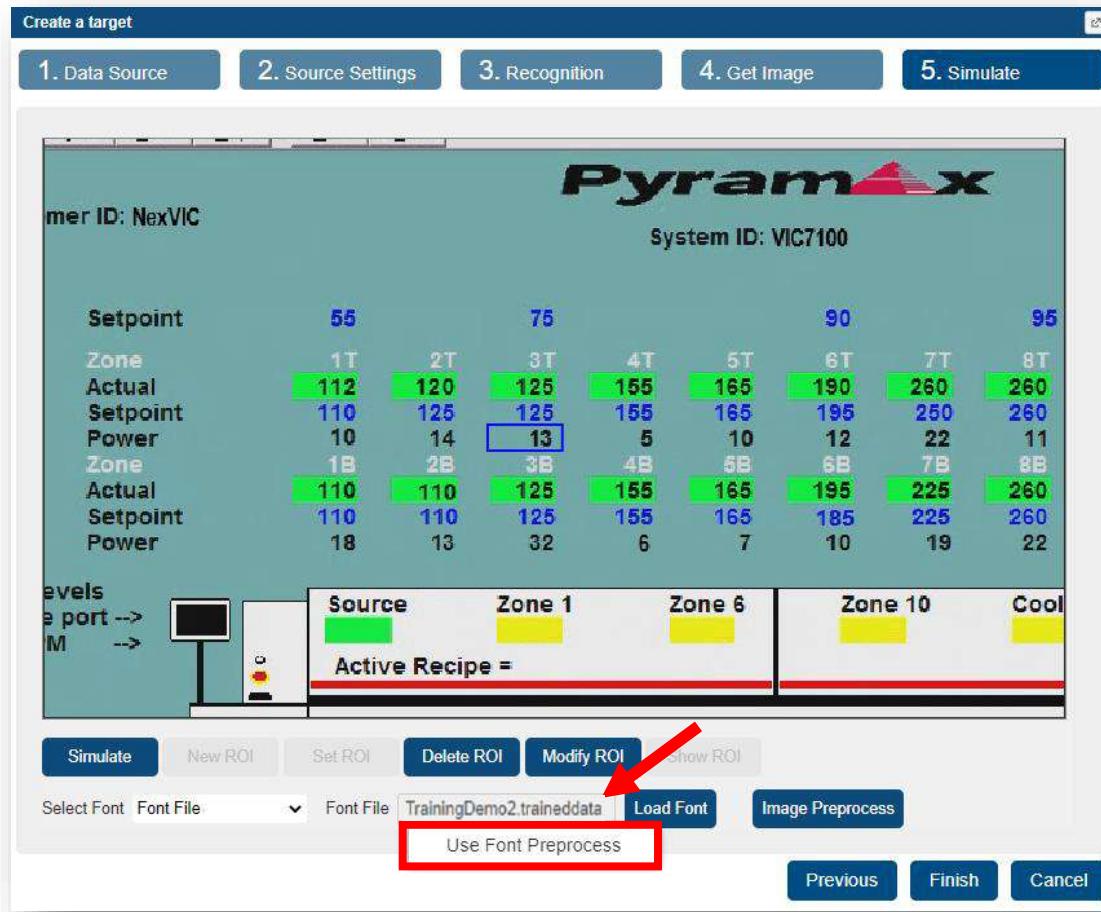


# 目標值

- 精靈：新增目標值

## 5. 模擬

在字型檔欄位點擊右鍵，選擇使用字型的前處理值，可以快速選用前處理參數



# 目標值

- 精靈：新增目標值

## 5. 模擬

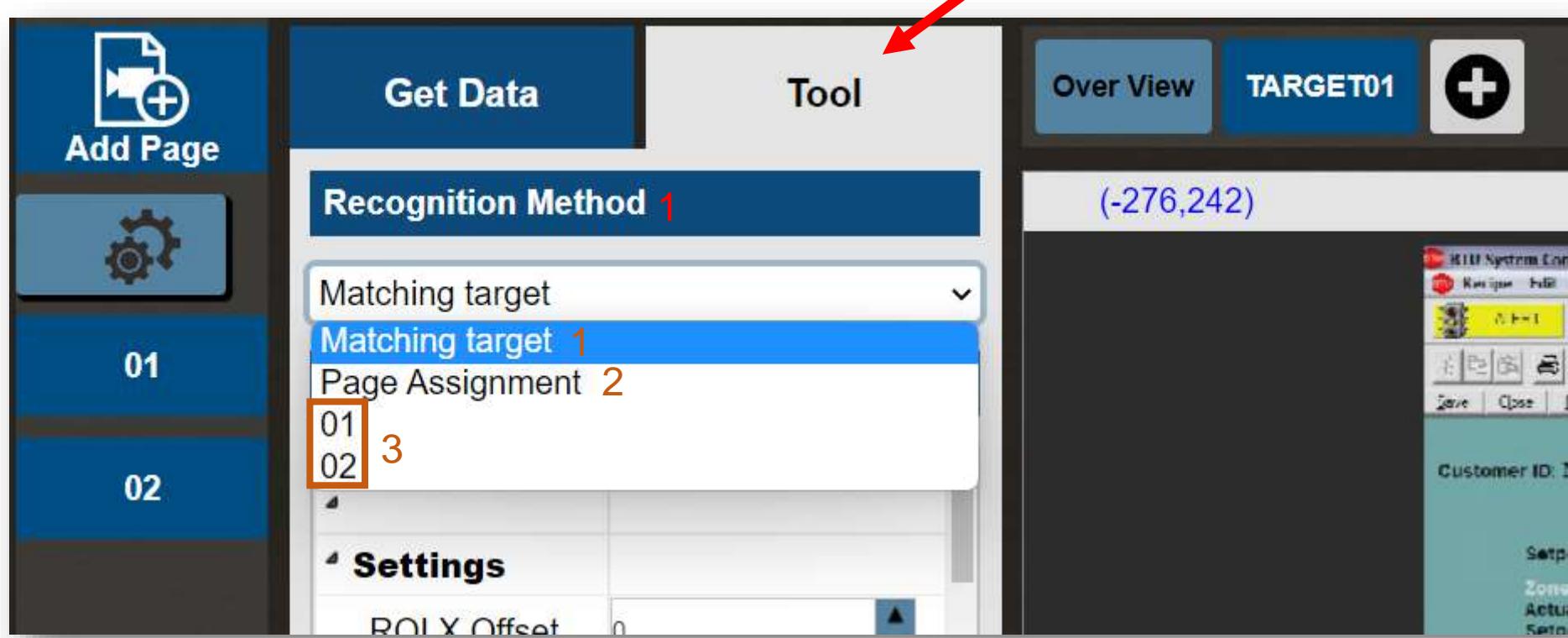
點擊模擬 → 確認辨識值



- 一般：新增目標值

- 選擇辨識方法

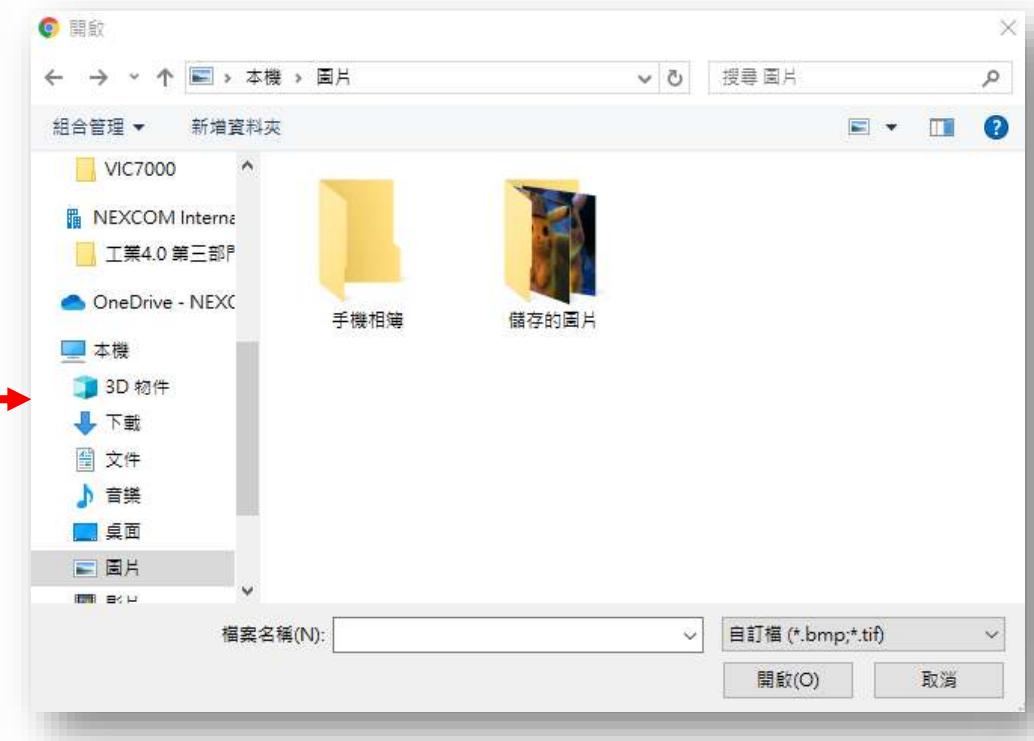
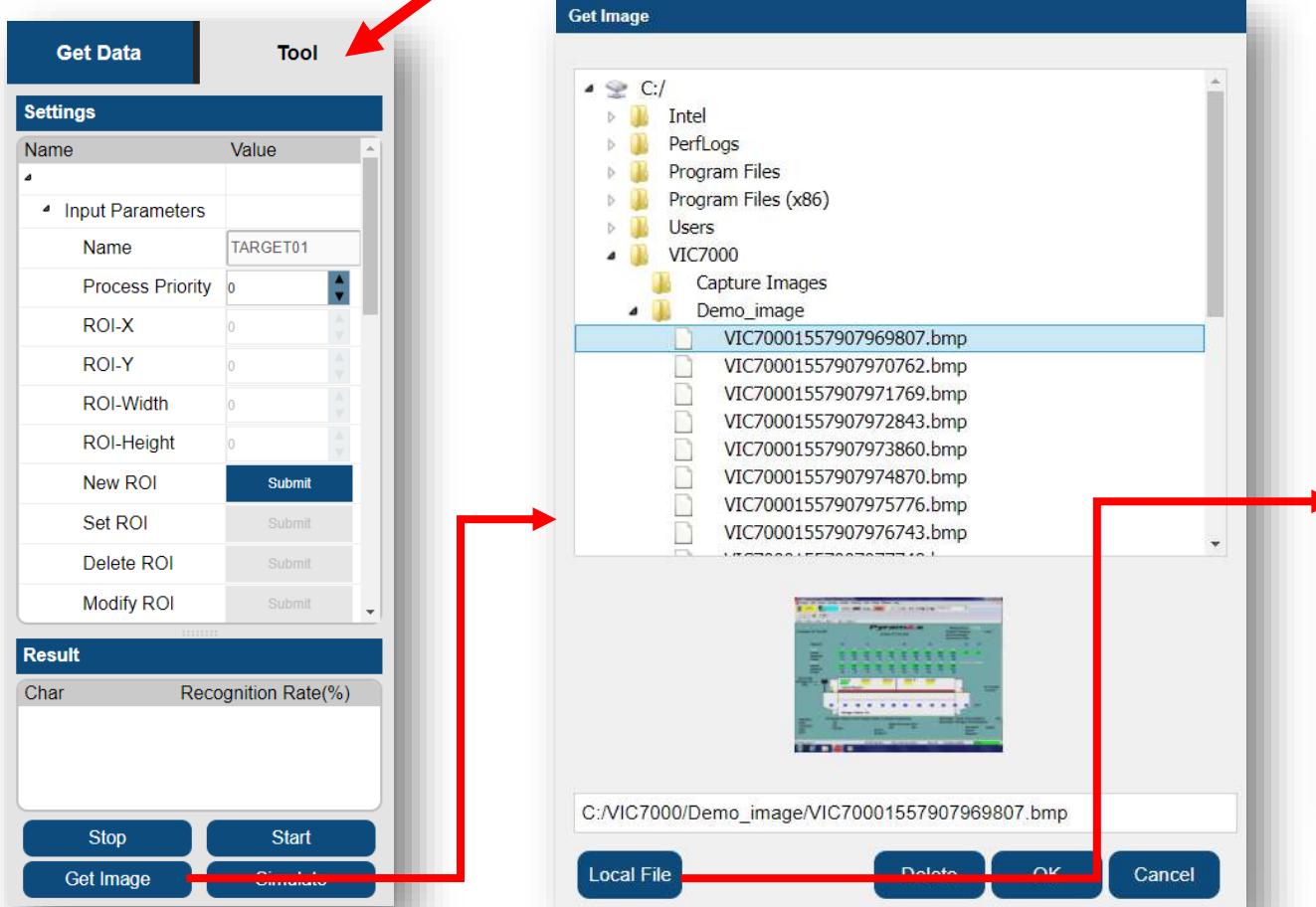
- 比對目標值：依據目標值比對條件自動選擇要使用的頁面
- 頁面指定：透過腳本或外部連線選擇要使用的頁面
- 選擇頁面：直接選擇要使用的頁面



- 一般：新增目標值

## 3. 取得圖像

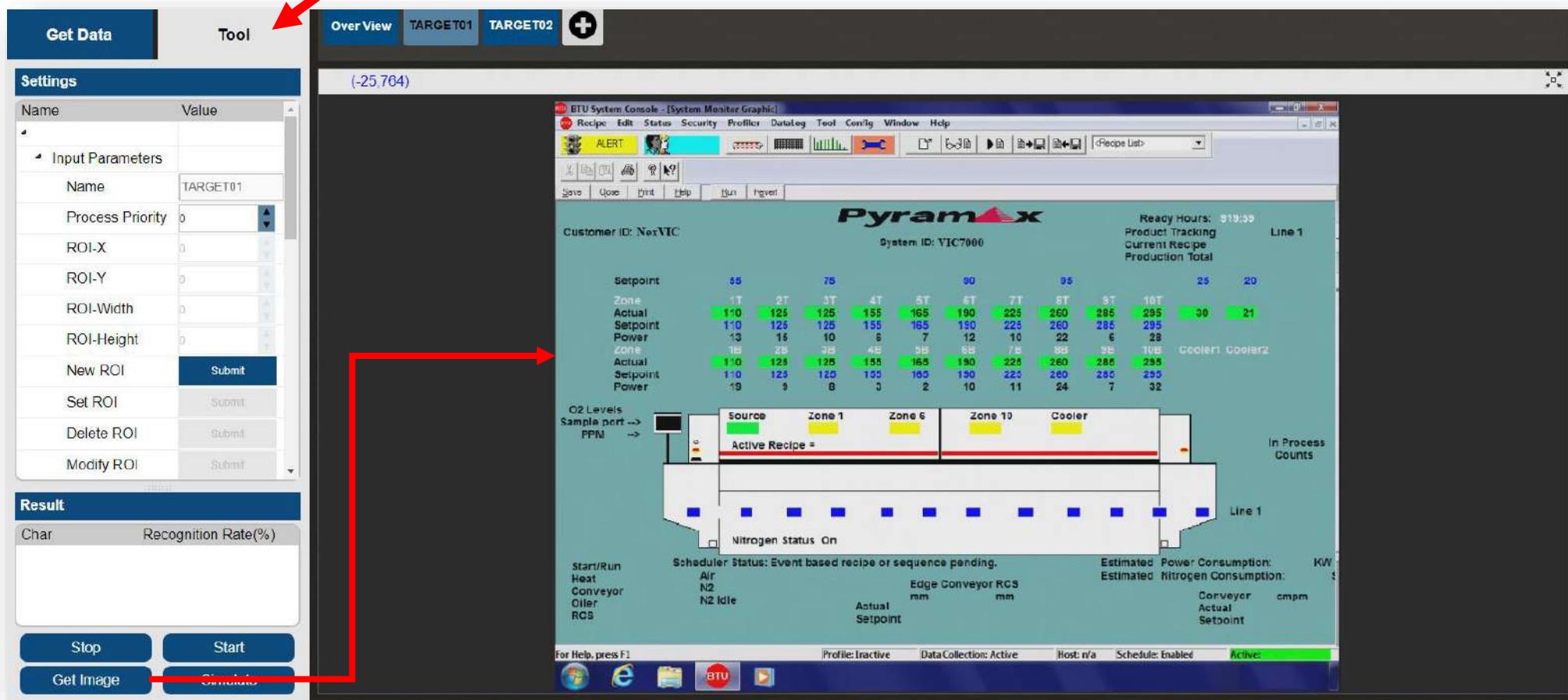
### a. 來自檔案：讀取本地或遠端的圖像



- 一般：新增目標值

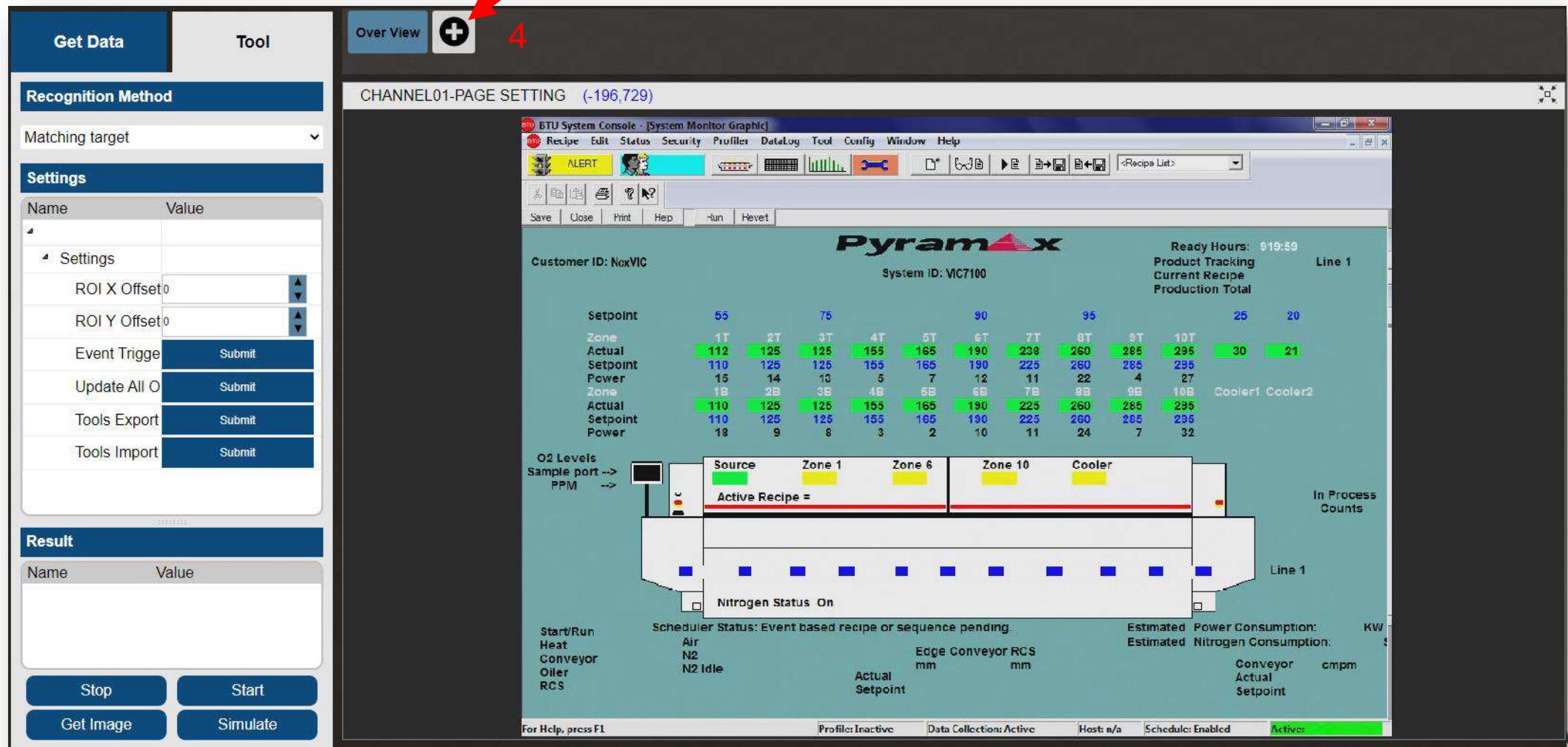
3. 取得圖像

b. 來自擷取卡、RSTP 影像、瀏覽器、本機畫面擷取：從影像來源取得一張圖像



- 一般：新增目標值

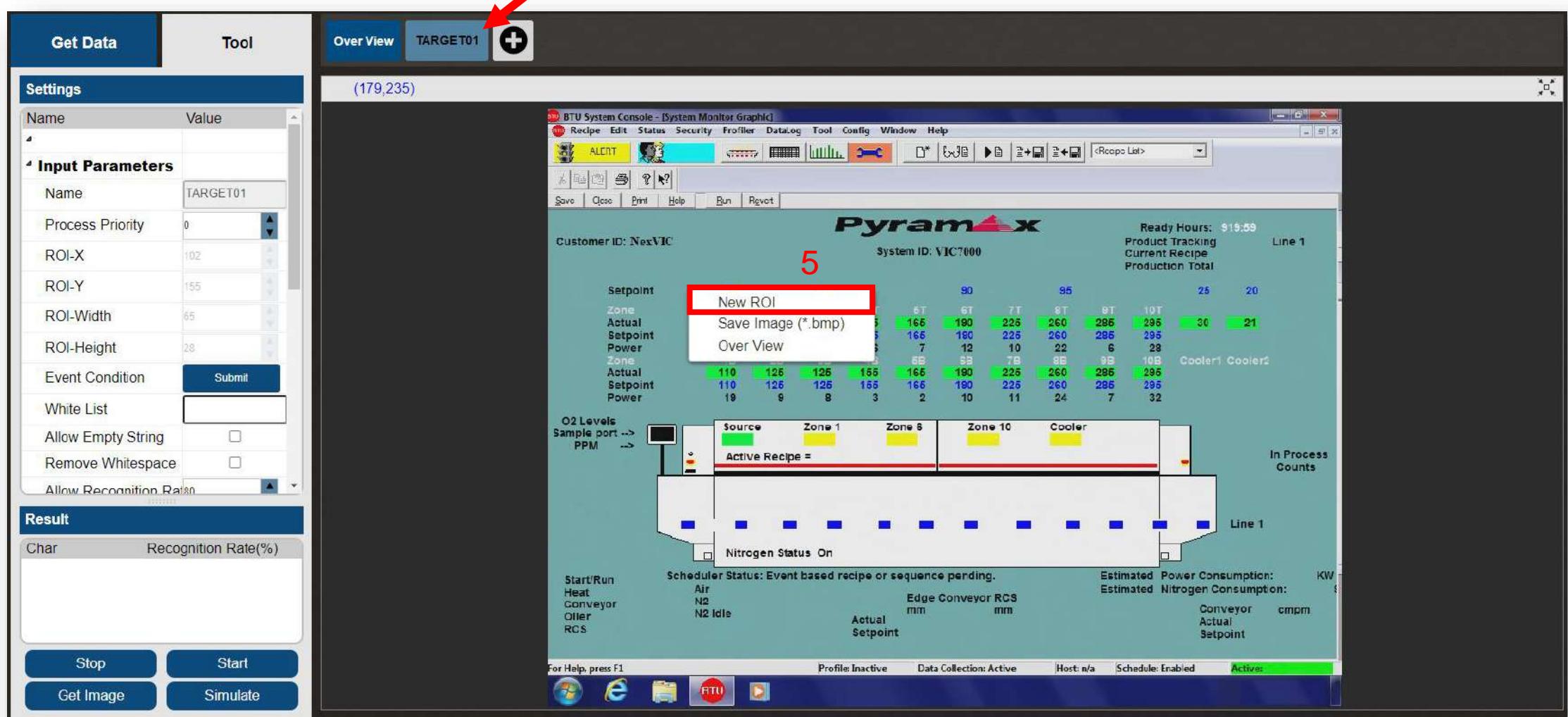
4. 新增目標值：點擊新增按鈕，會看到新增的目標值



# 目標值

- 一般：新增目標值

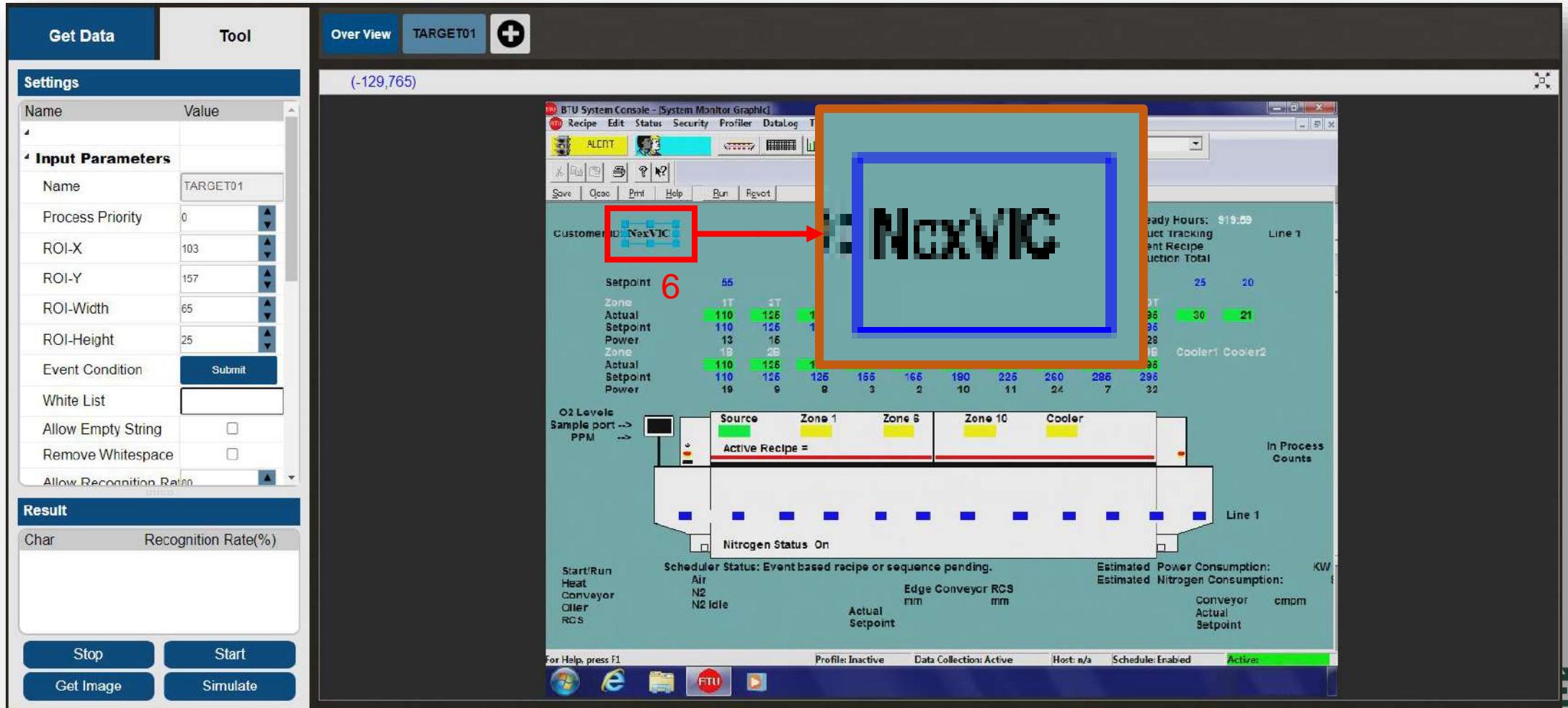
5. 新增ROI：在圖像上點擊右鍵 → 選擇新增ROI



# 目標值

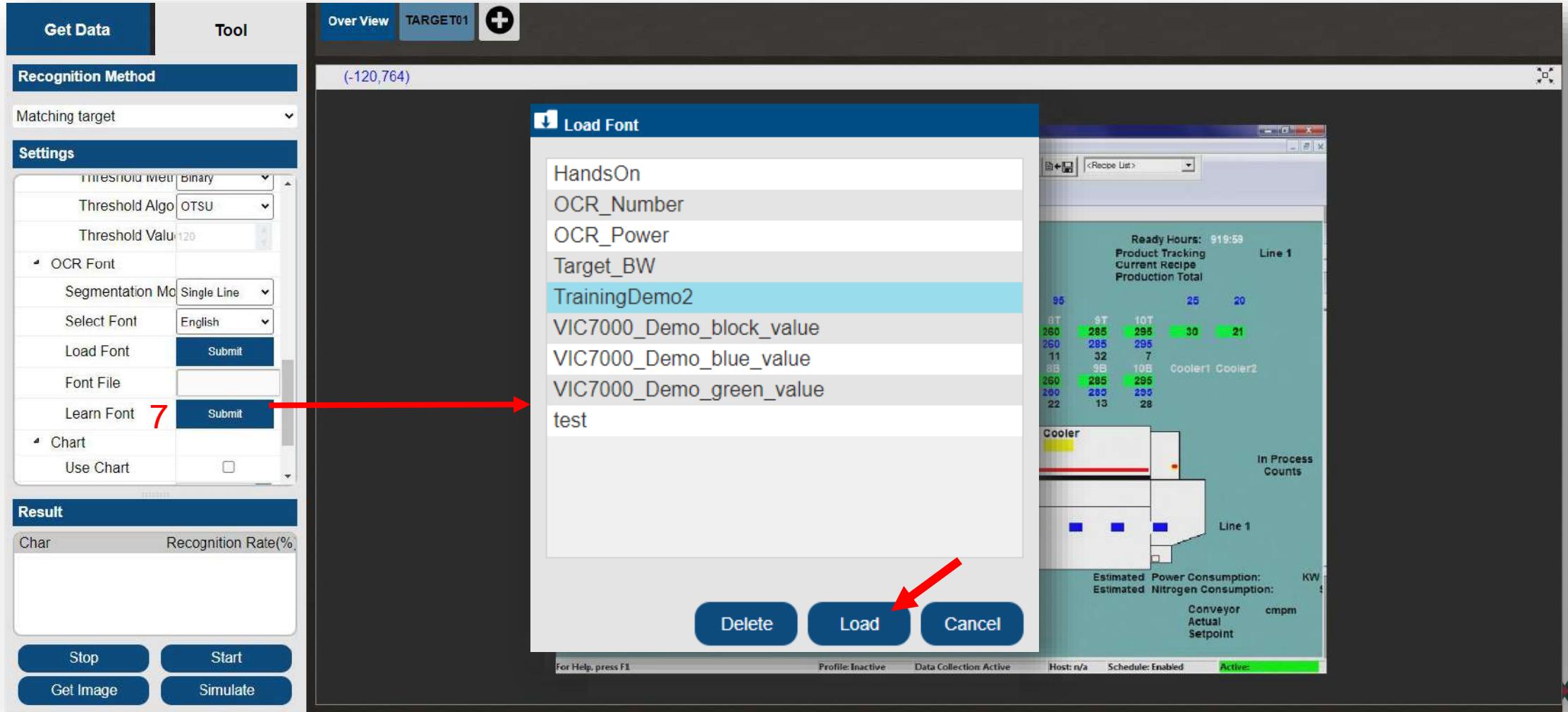
- 一般：新增目標值

6. 設置ROI：調整ROI大小和位置 → 在圖像任意處點擊左鍵



- 一般：新增目標值

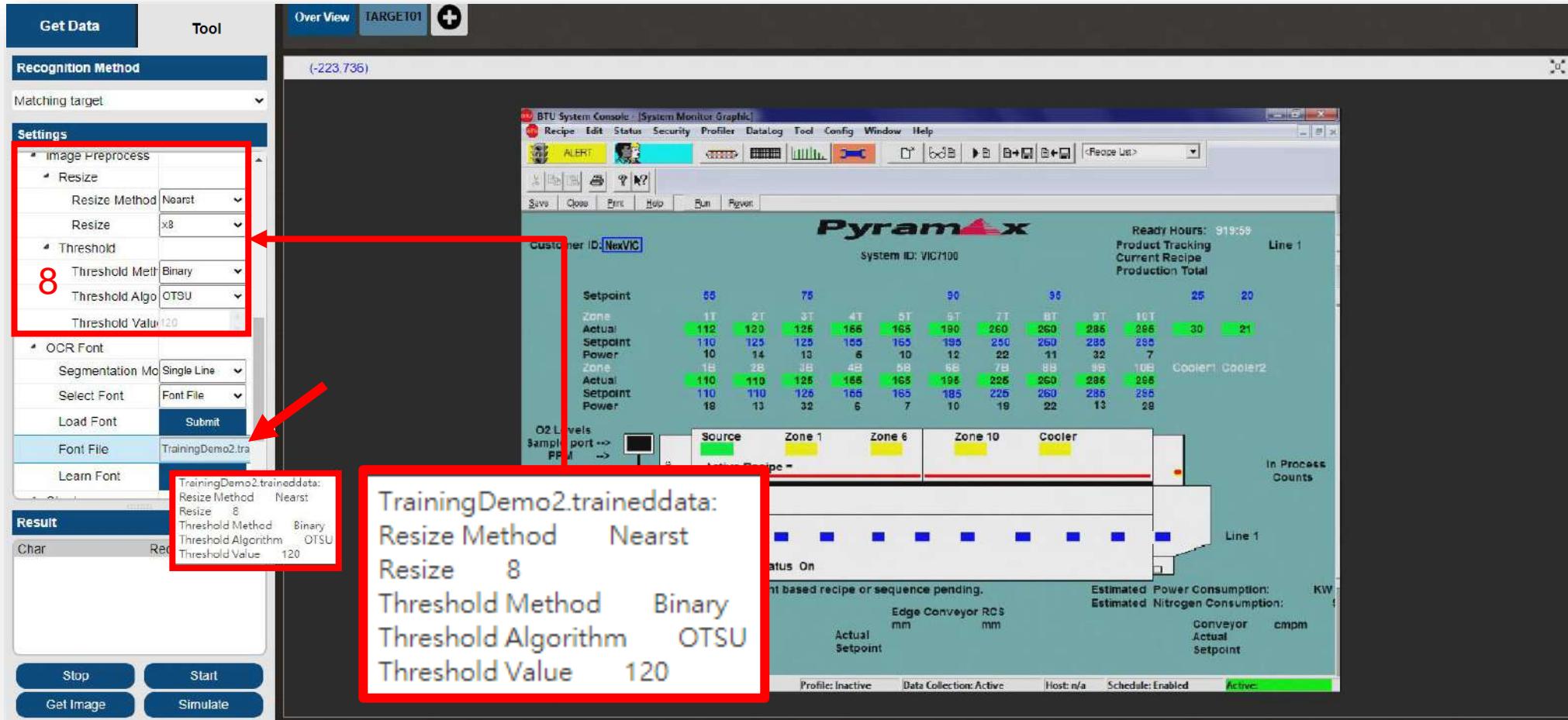
7. 選擇字型檔：載入字型 → 選擇要使用的字型檔 → 點擊載入



- 一般：新增目標值

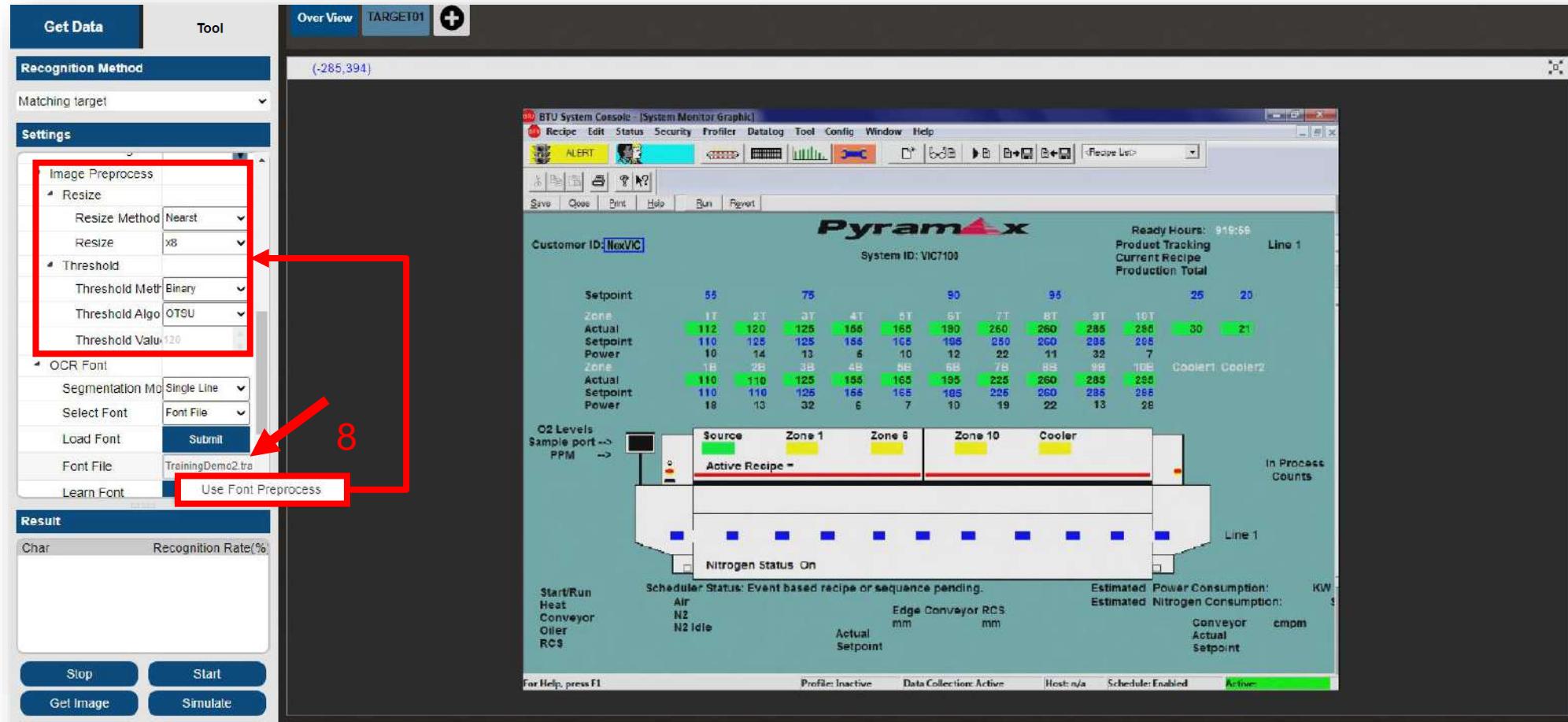
8. 設定前處理參數：設定影像前處理參數，要和字型專案內的前處理參數相同

將游標移到字型檔欄位上，會顯示該字型檔使用的前處理參數



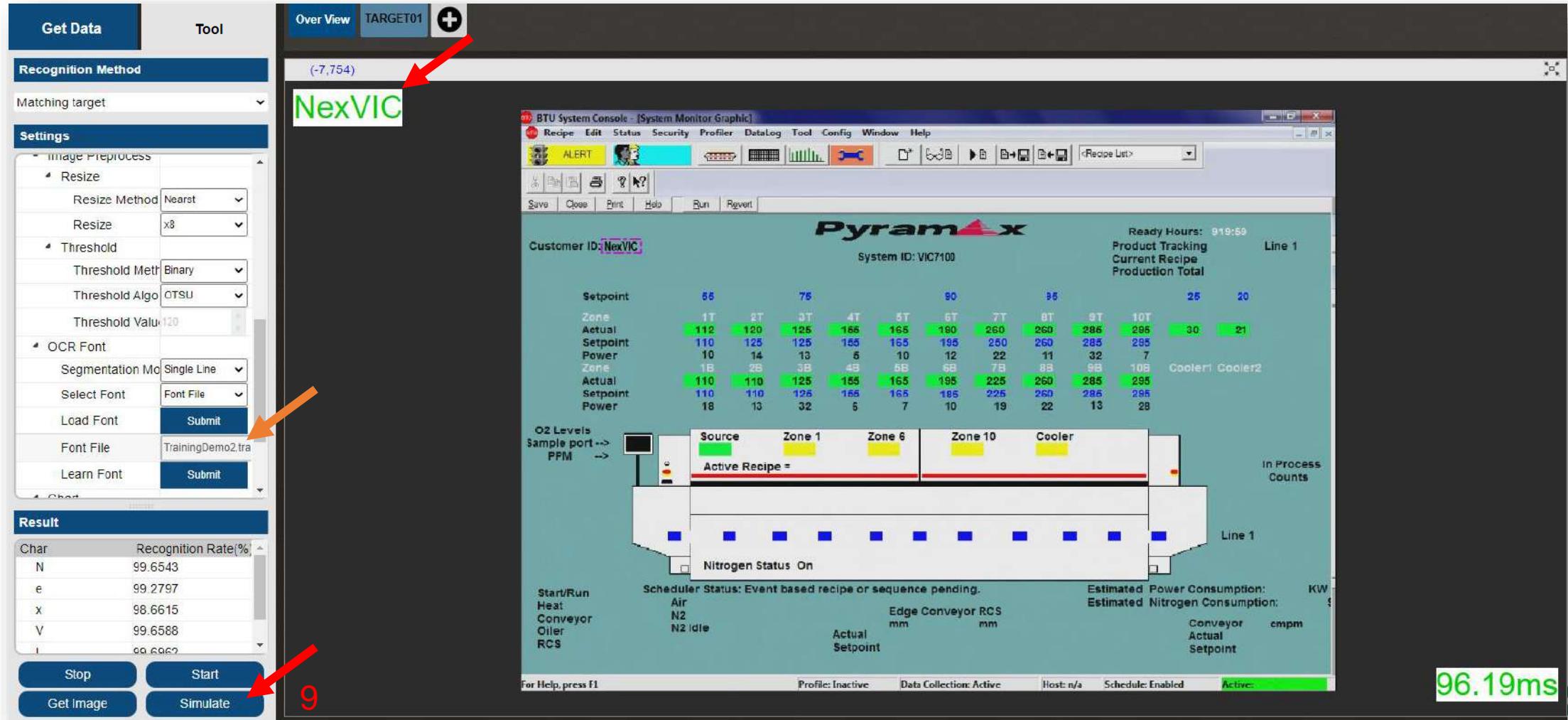
- 一般：新增目標值

8. 設定前處理參數：在字型檔欄位點擊右鍵，選擇使用字型的前處理值，可以快速套用前處理參數



- 一般：新增目標值

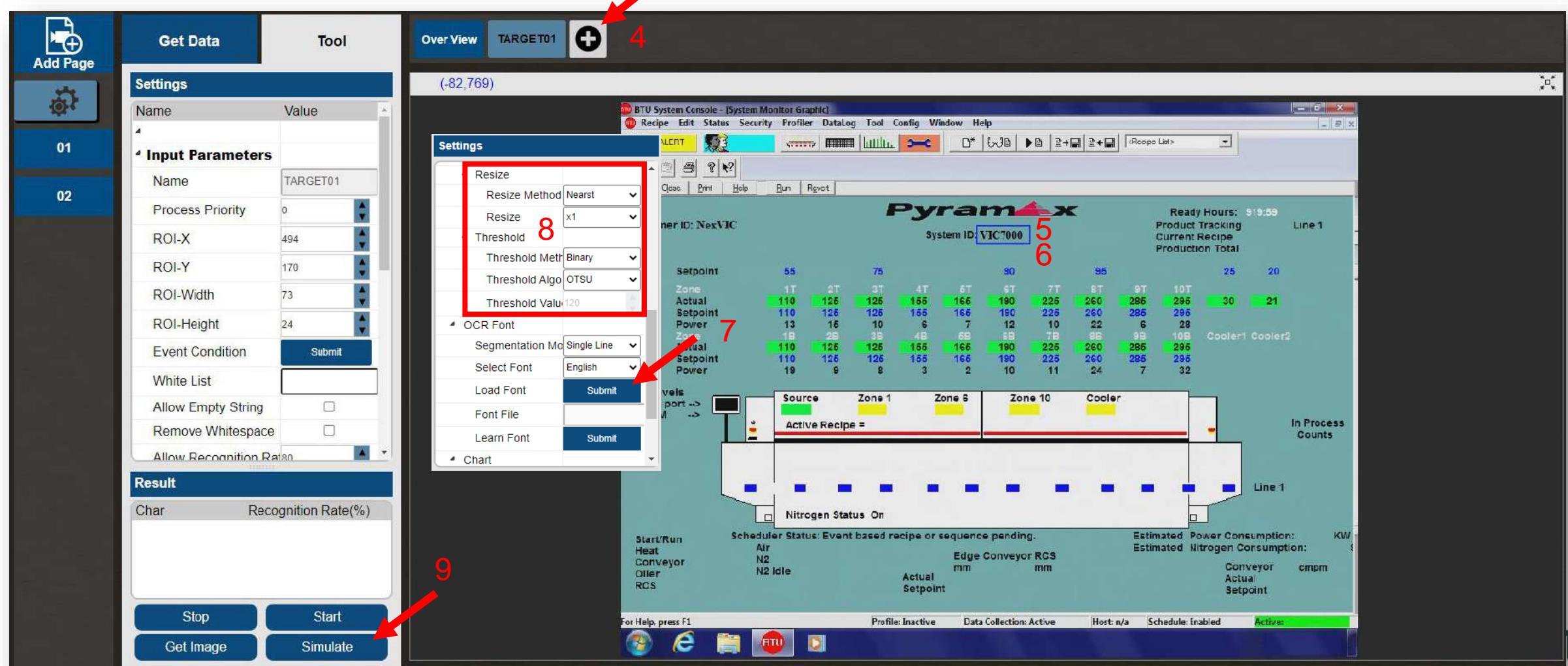
9. 模擬確認：點擊模擬 → ROI內的辨識值即為目標值



# 目標值

- 一般：新增目標值

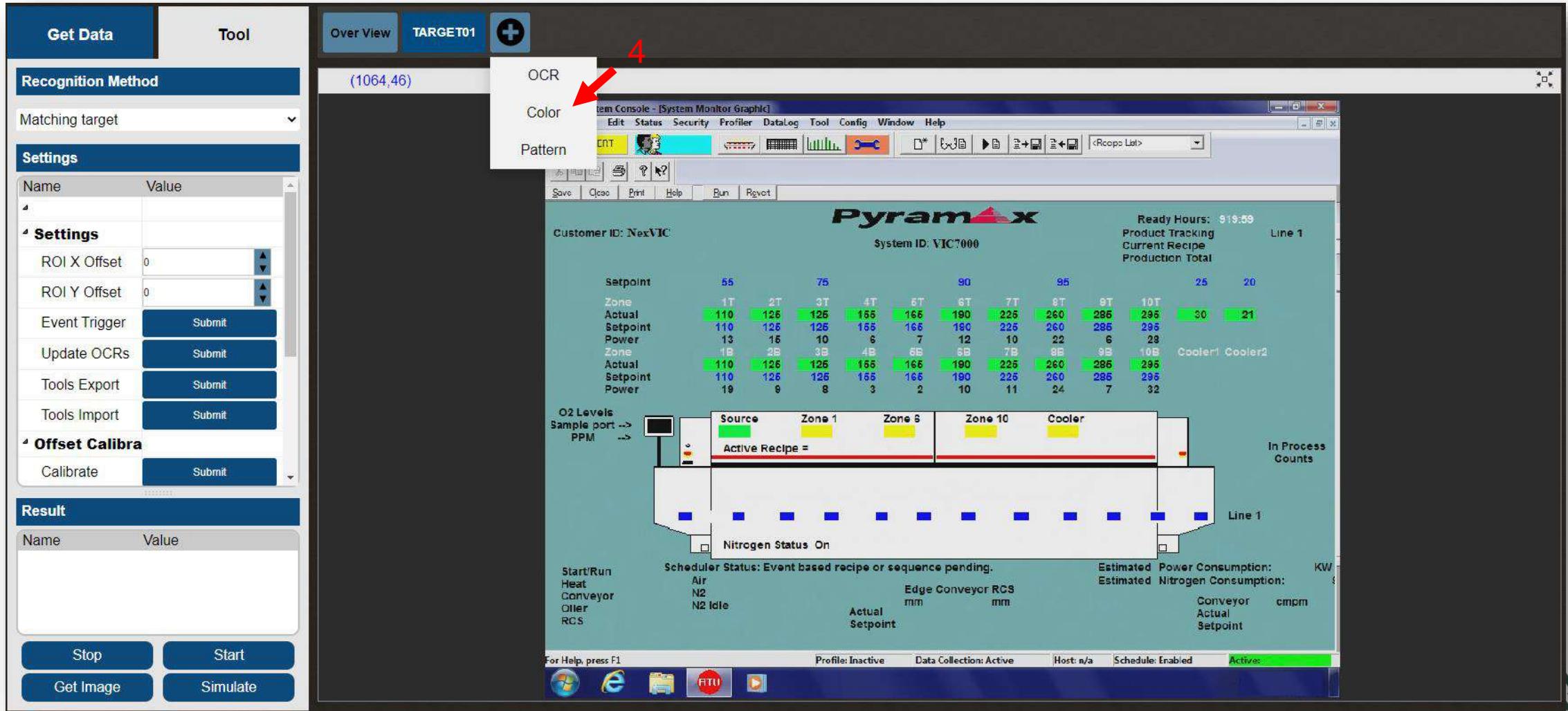
- 多個目標值：在同一頻道的頁面設定中點擊新增按鈕 → 依照前述步驟可新增更多目標值



# 目標值

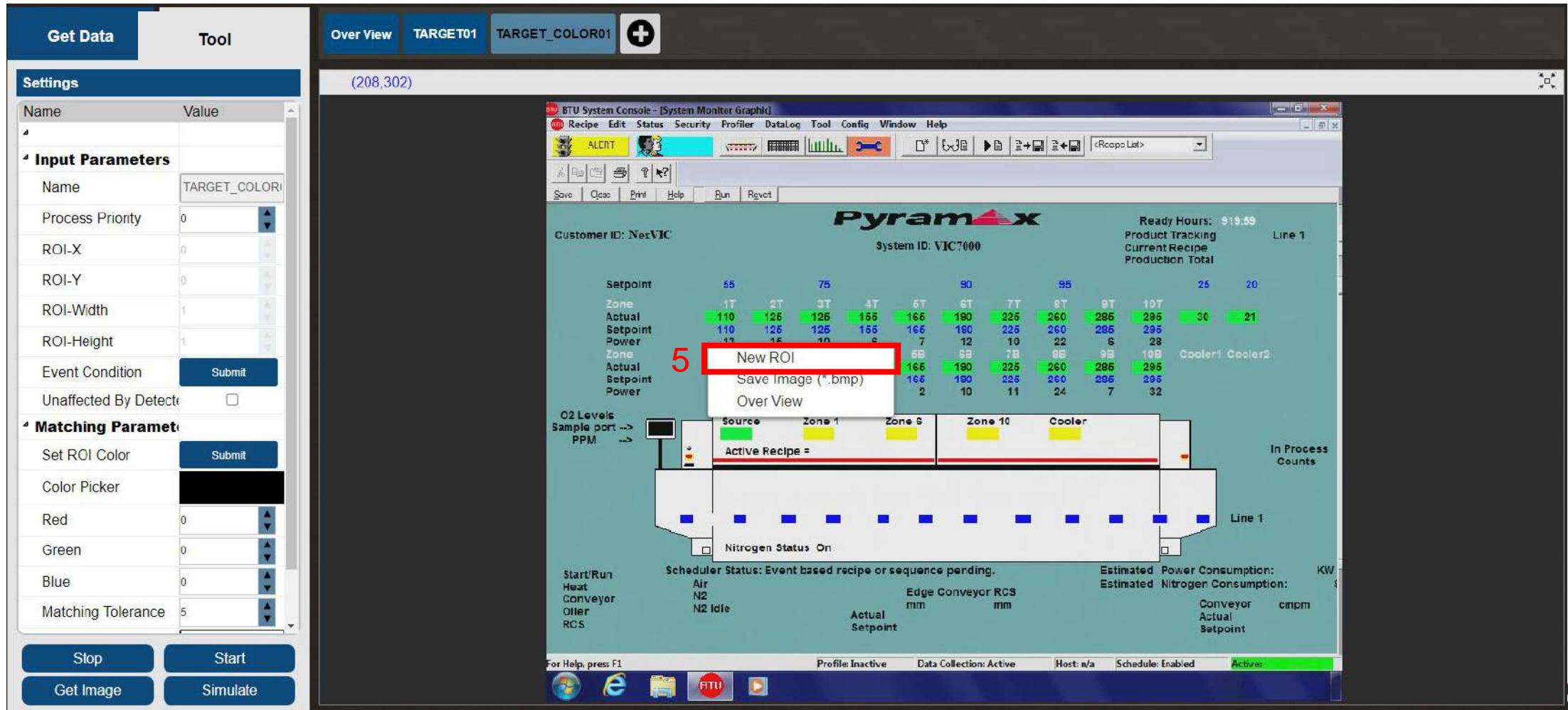
- 一般：新增顏色目標值

## 4. 新增顏色目標值：滑鼠移到新增按鈕 → 點擊 Color



- 一般：新增顏色目標值

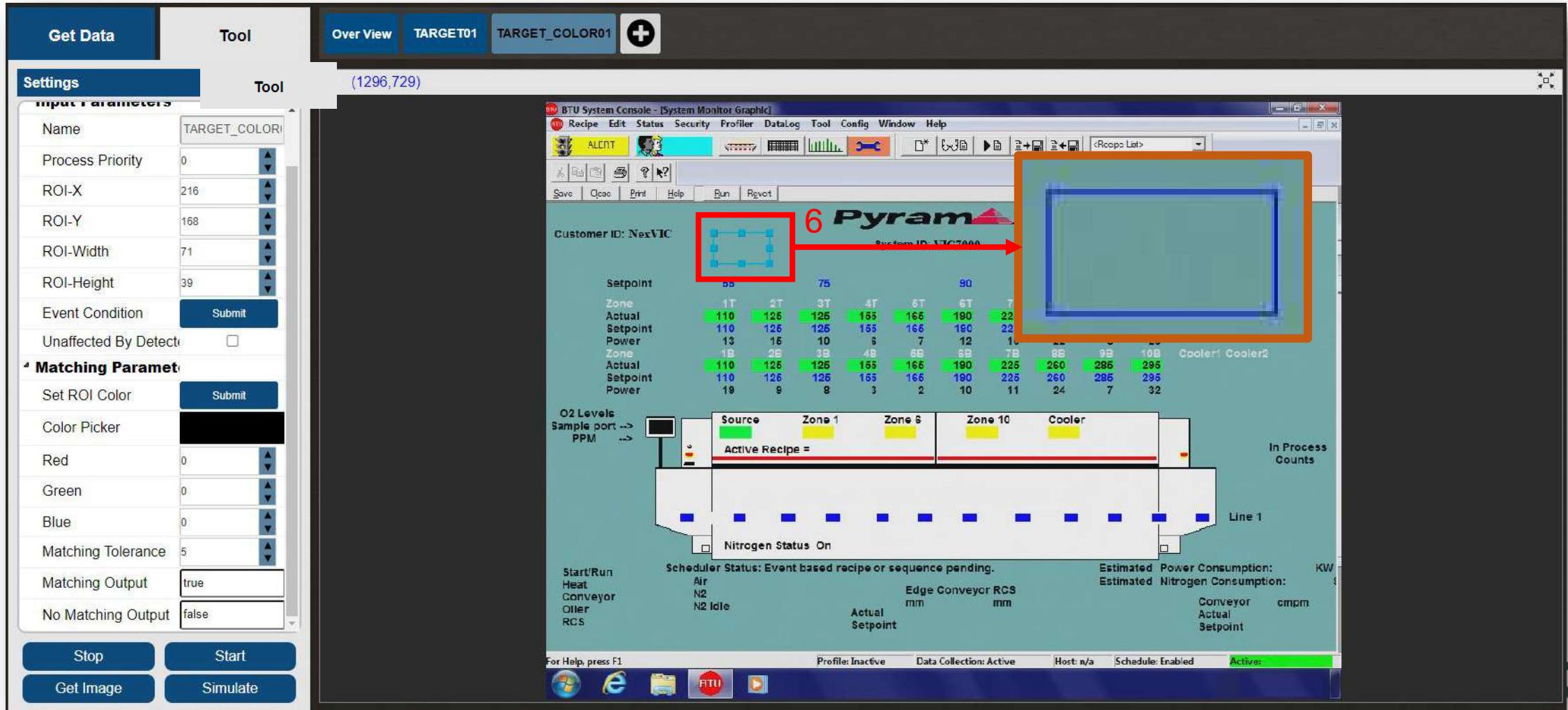
5. 新增ROI：在圖像上點擊右鍵 → 選擇新增 ROI



# 目標值

- 一般：新增顏色目標值

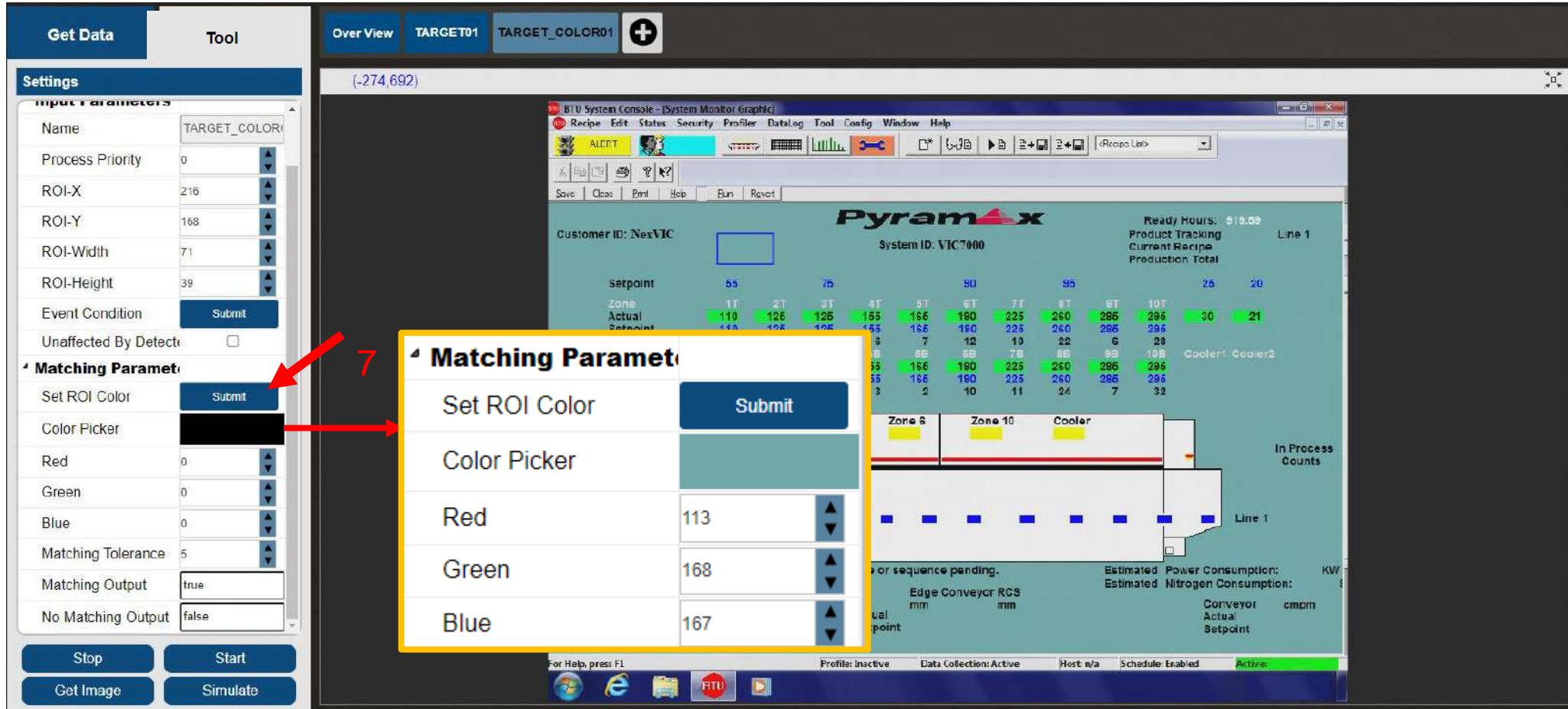
6. 設置ROI：調整ROI大小和位置 → 在圖像任意處點擊左鍵



- 一般：新增顏色目標值

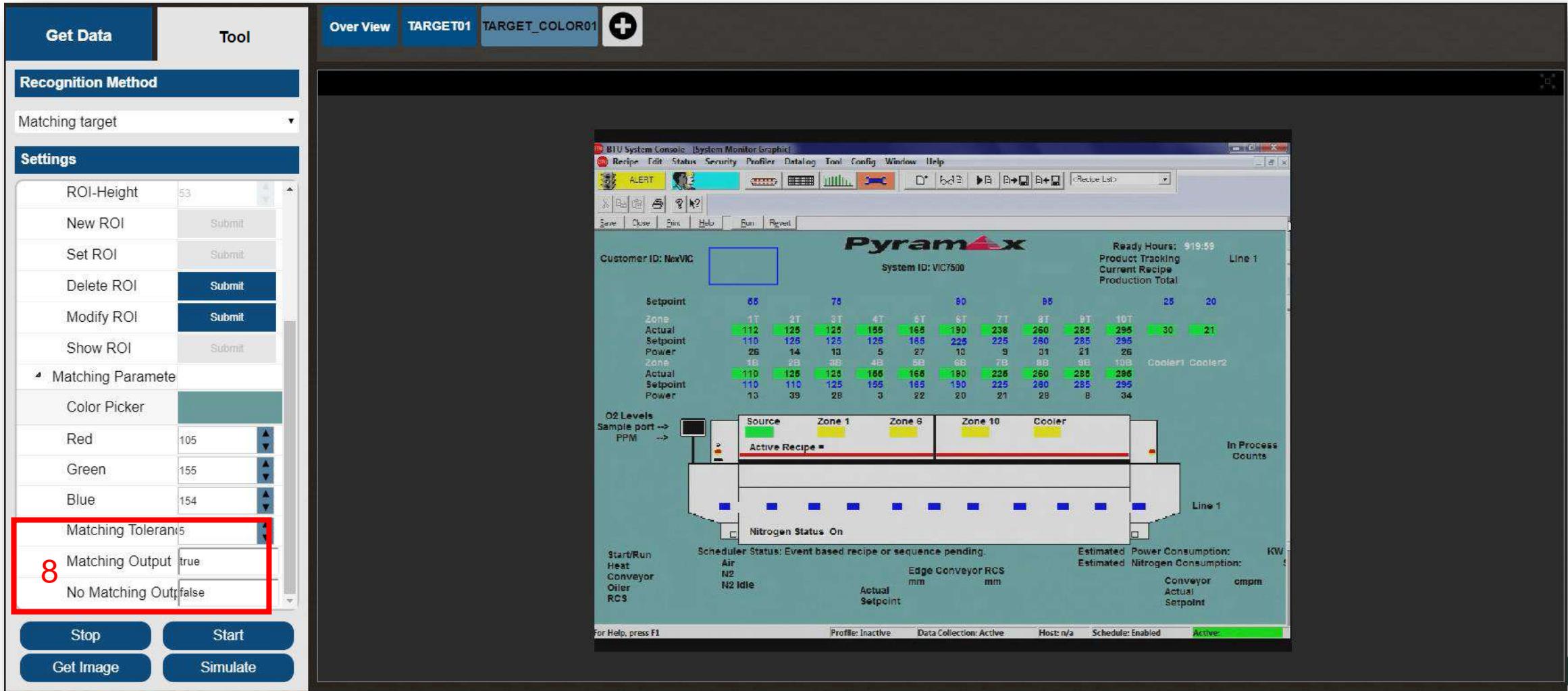
## 7. 設置比對顏色：點擊設置ROI顏色

系統會自動取得ROI內的 RGB 平均值，並將其設為顏色比對條件



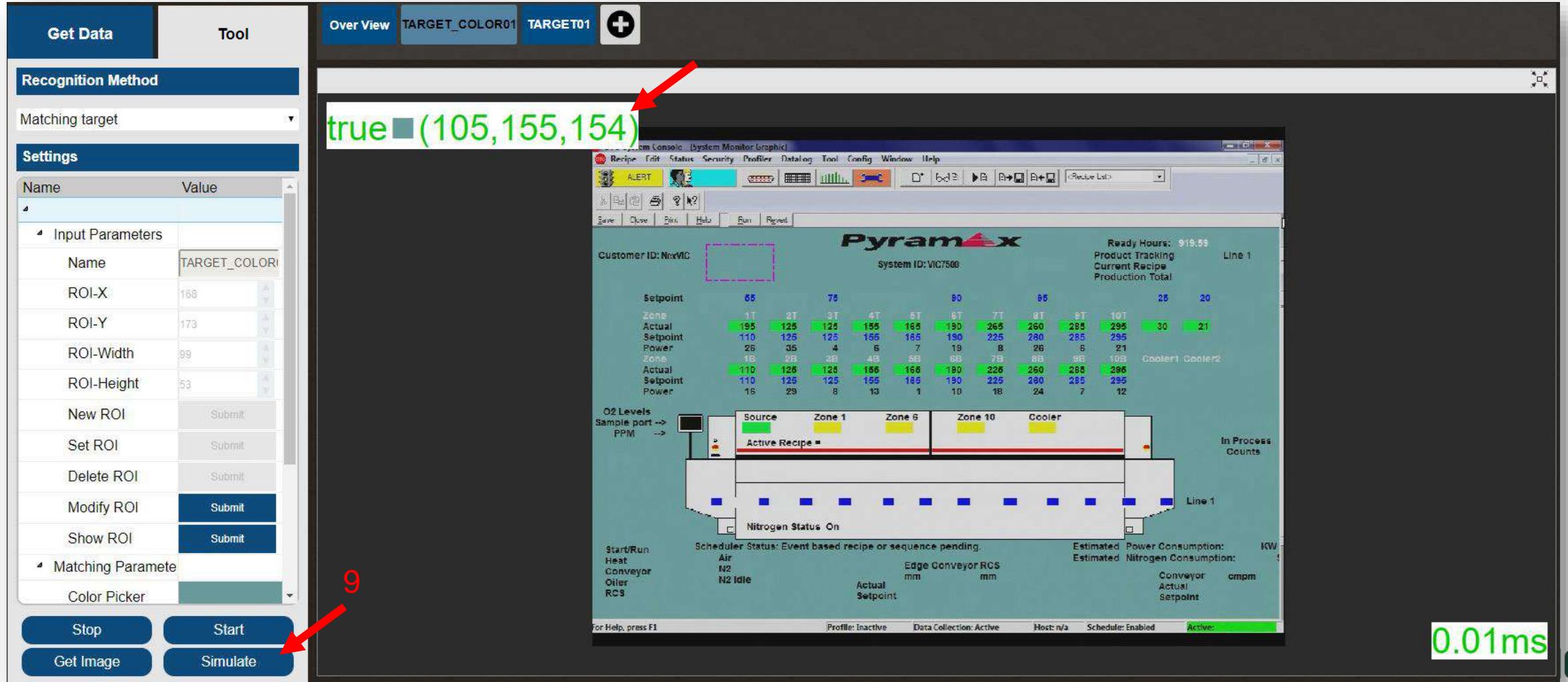
- 一般：新增顏色目標值

8. 設定比對容錯度和比對結果輸出：設定比對容錯度(RGB±) → 輸入比對結果輸出



- 一般：新增顏色目標值

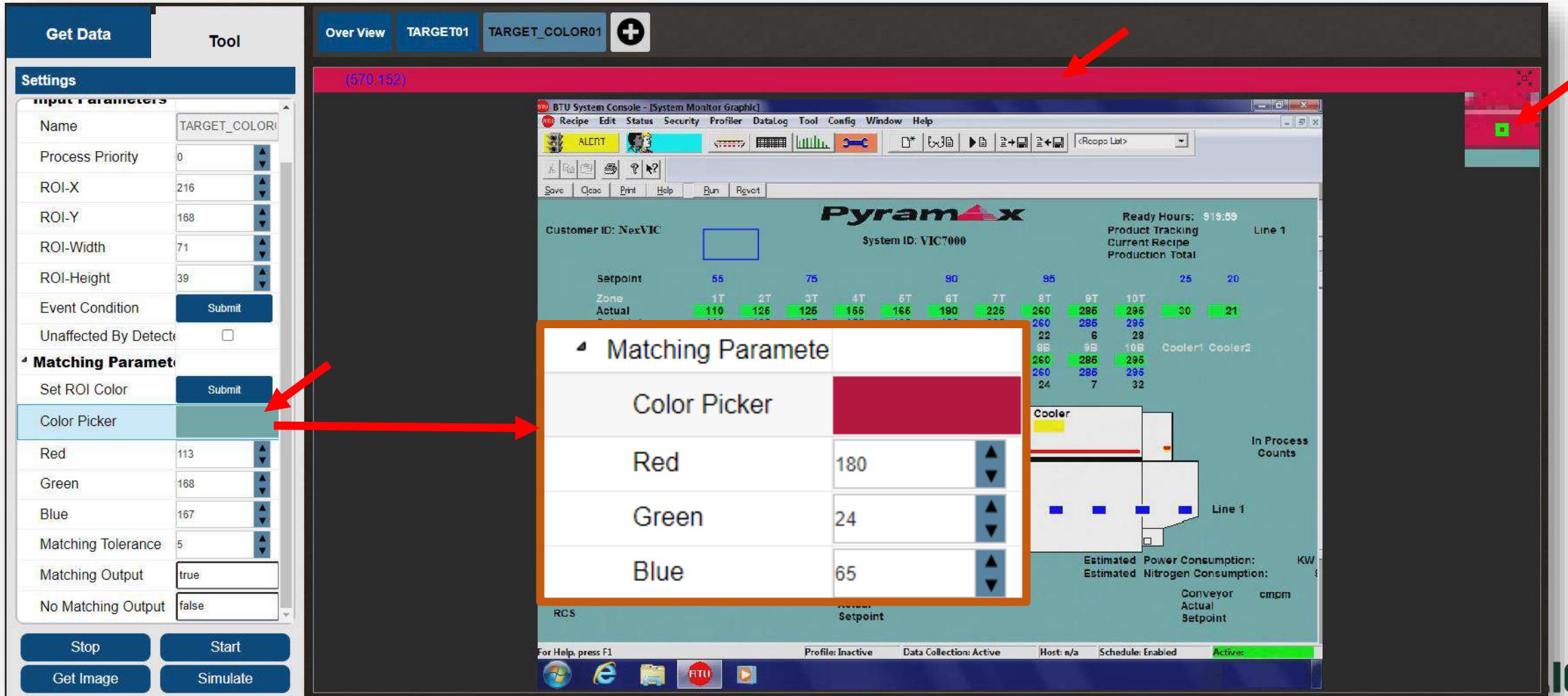
9. 模擬確認：點擊模擬 → 確認辨識結果，此結果即為顏色目標值



# 目標值

- 一般：新增顏色目標值

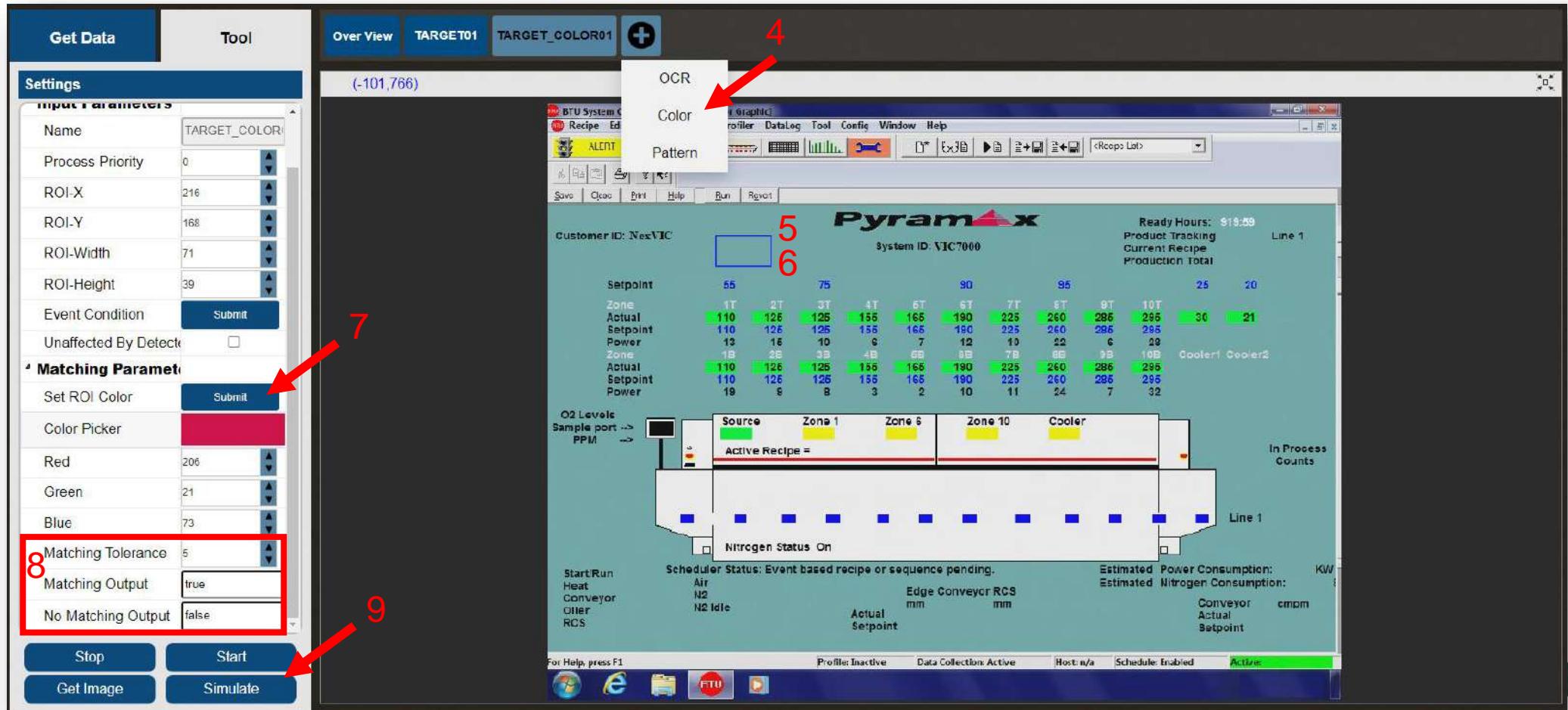
- 自選顏色：點擊顏色選擇 → 點擊圖像中設為比對條件的顏色 完成顏色比對條件設定



# 目標值

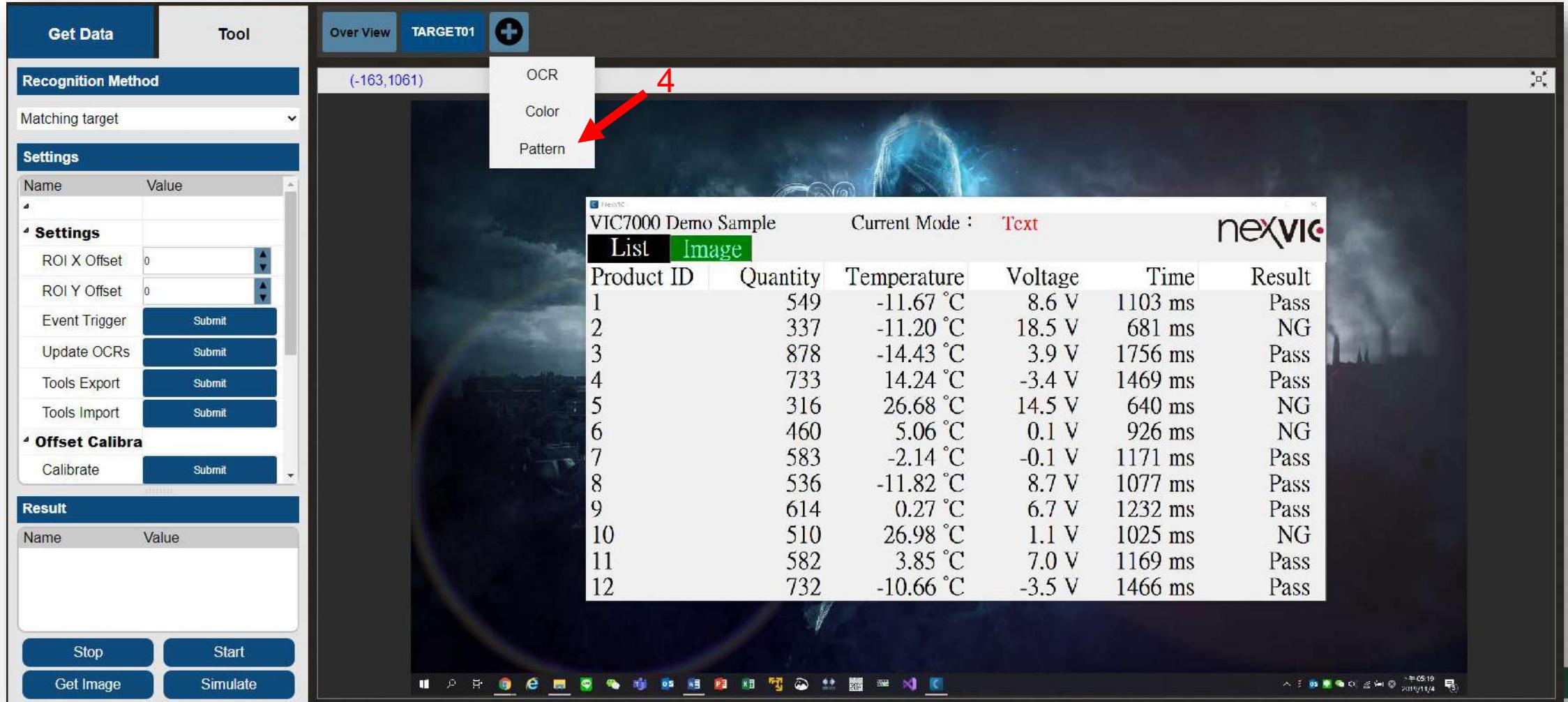
- 一般：新增顏色目標值

- 多個顏色目標值：在同一頻道的頁面設定中使用新增按鈕 → 依照前述步驟可新增更多顏色目標值



- 一般：新增樣式目標值

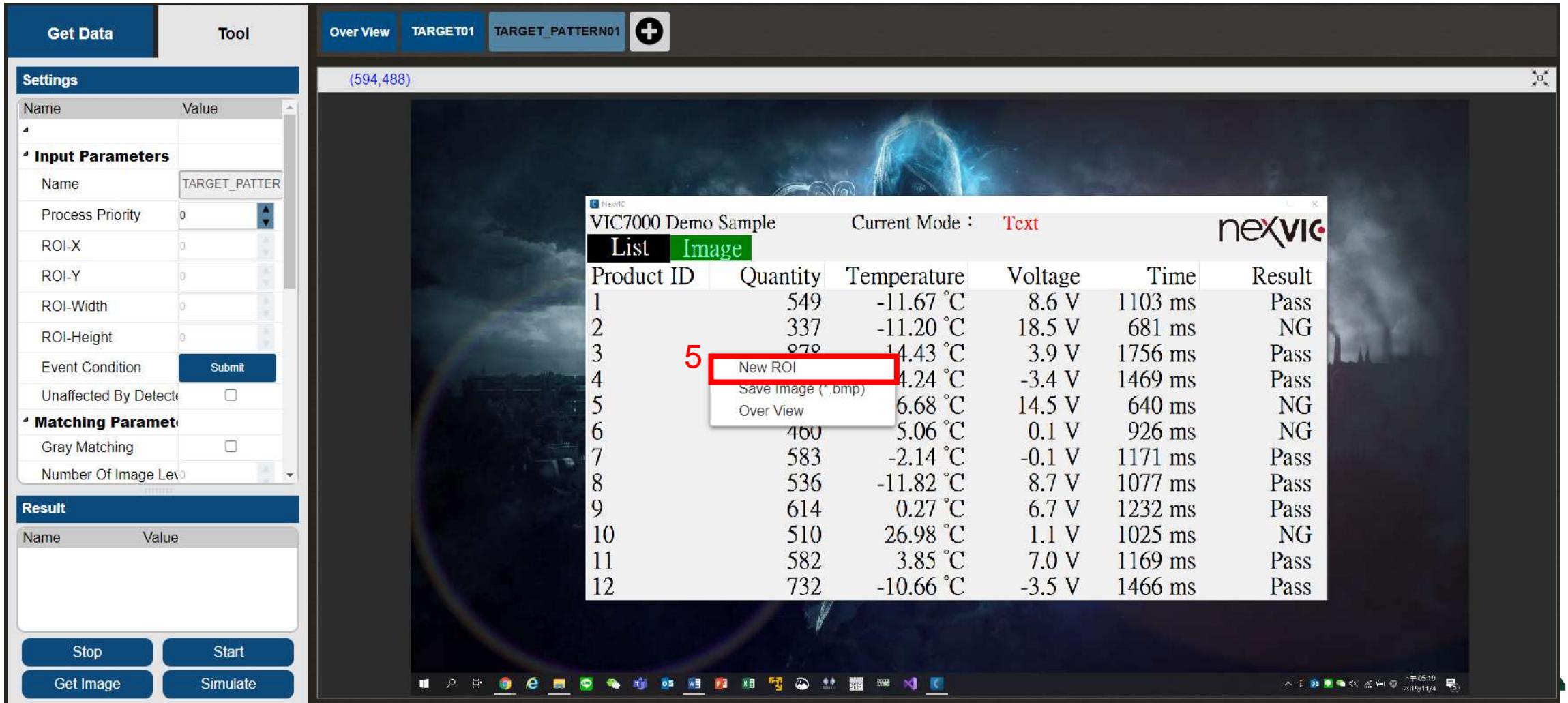
4. 新增樣式目標值：滑鼠移到新增按鈕 → 點擊 Pattern



# 目標值

- 一般：新增樣式目標值

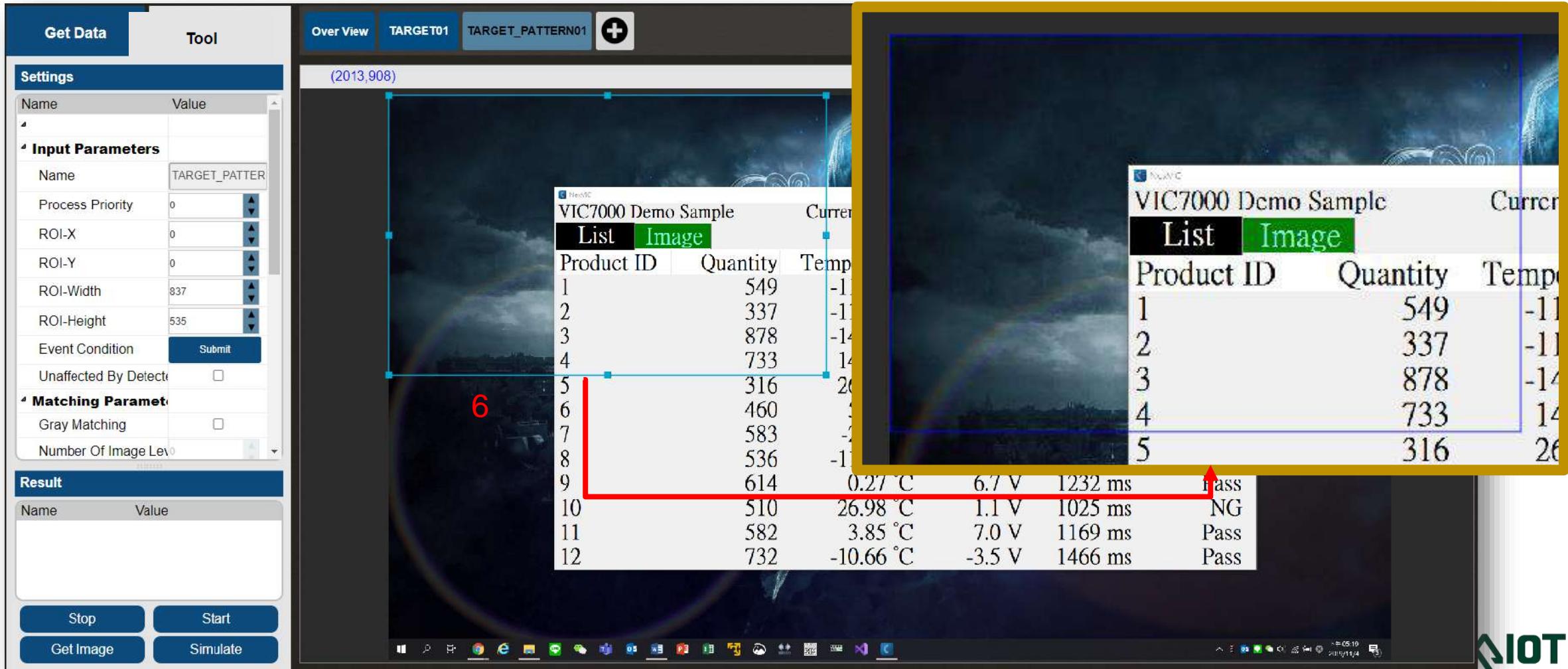
- 新增ROI：在圖像上點擊右鍵 → 選擇新增ROI



# 目標值

- 一般：新增樣式目標值

6. 設置ROI：調整ROI大小和位置 → 在圖像任意處點擊左鍵

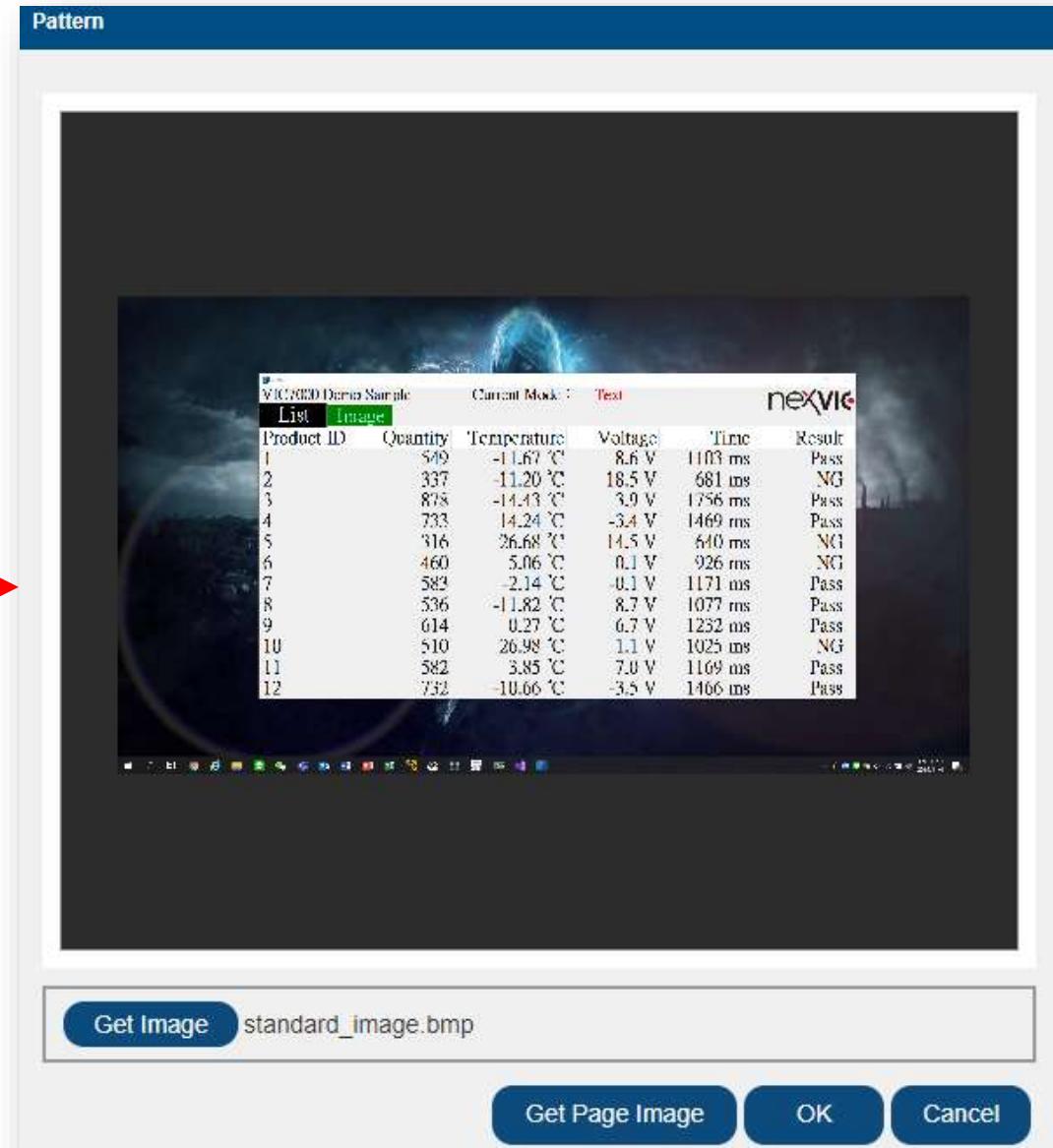
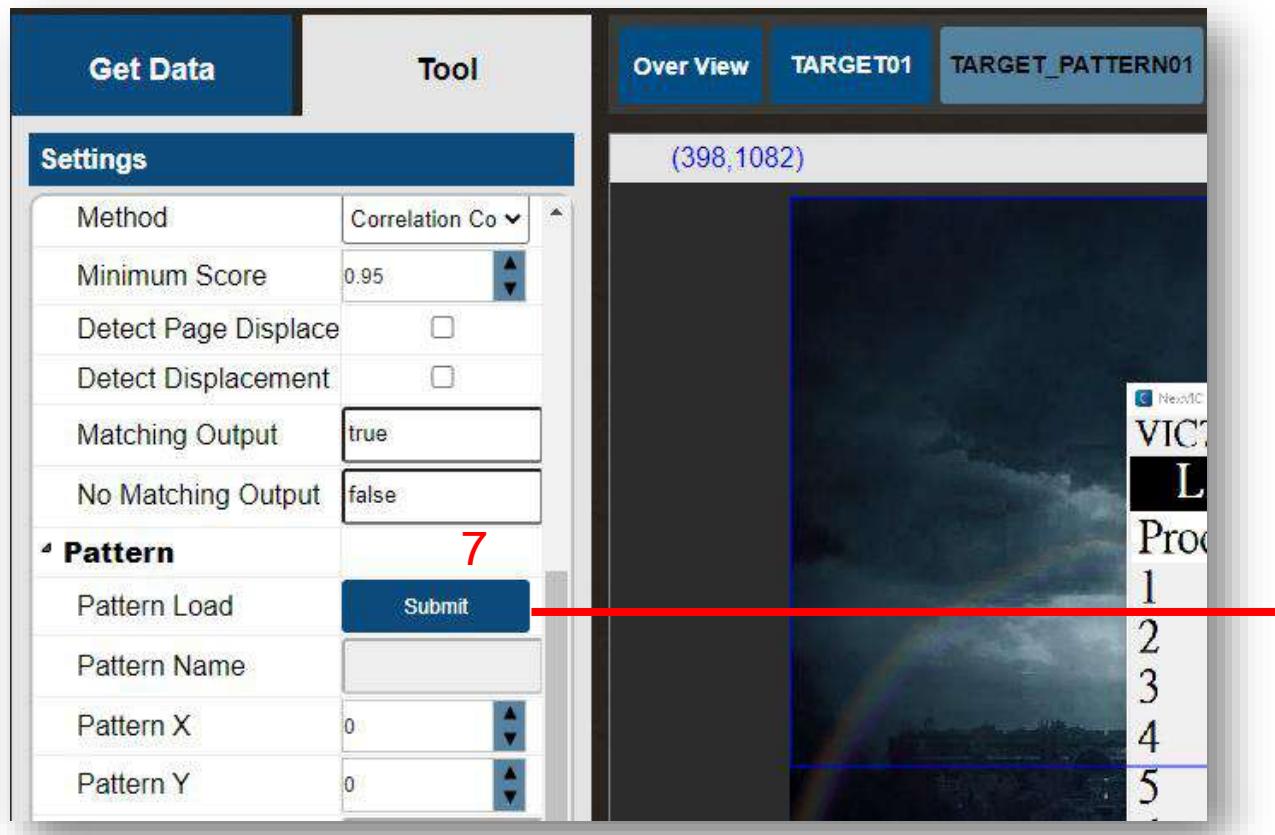


# 目標值

- 一般：新增樣式目標值

7. 設定比對樣式：點擊樣式載入，進入裁切視窗

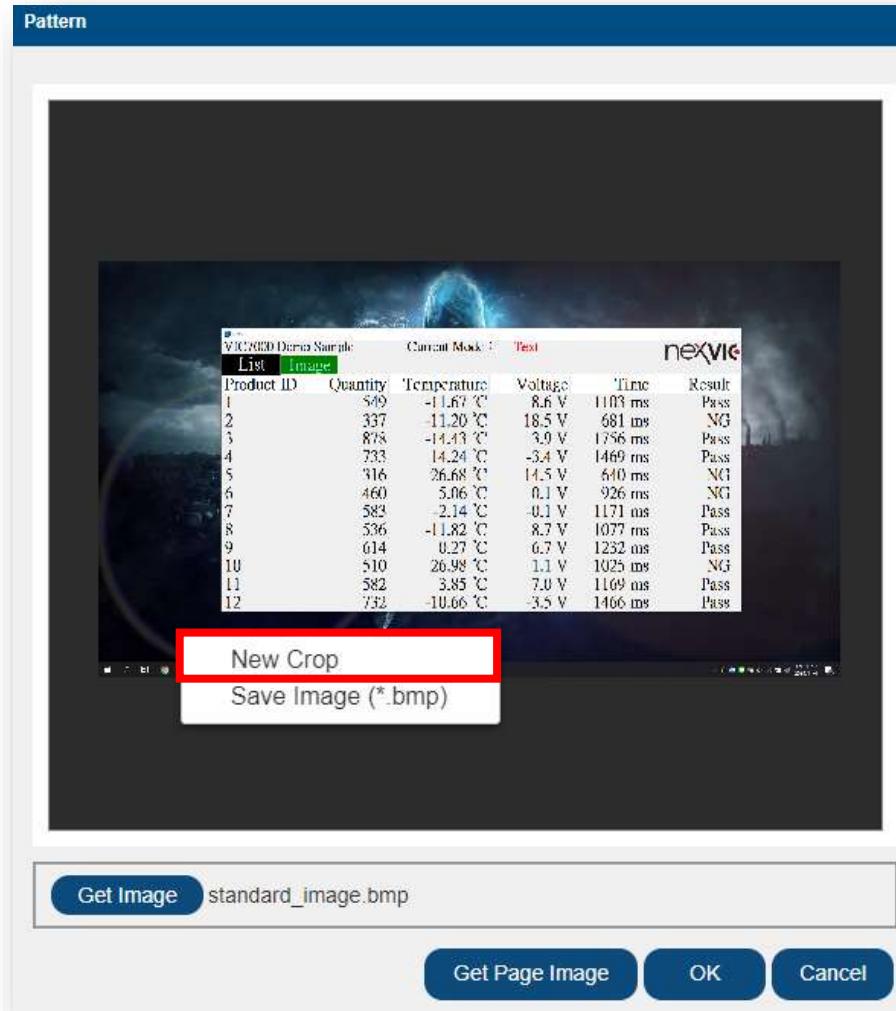
圖像會被自動載入



# 目標值

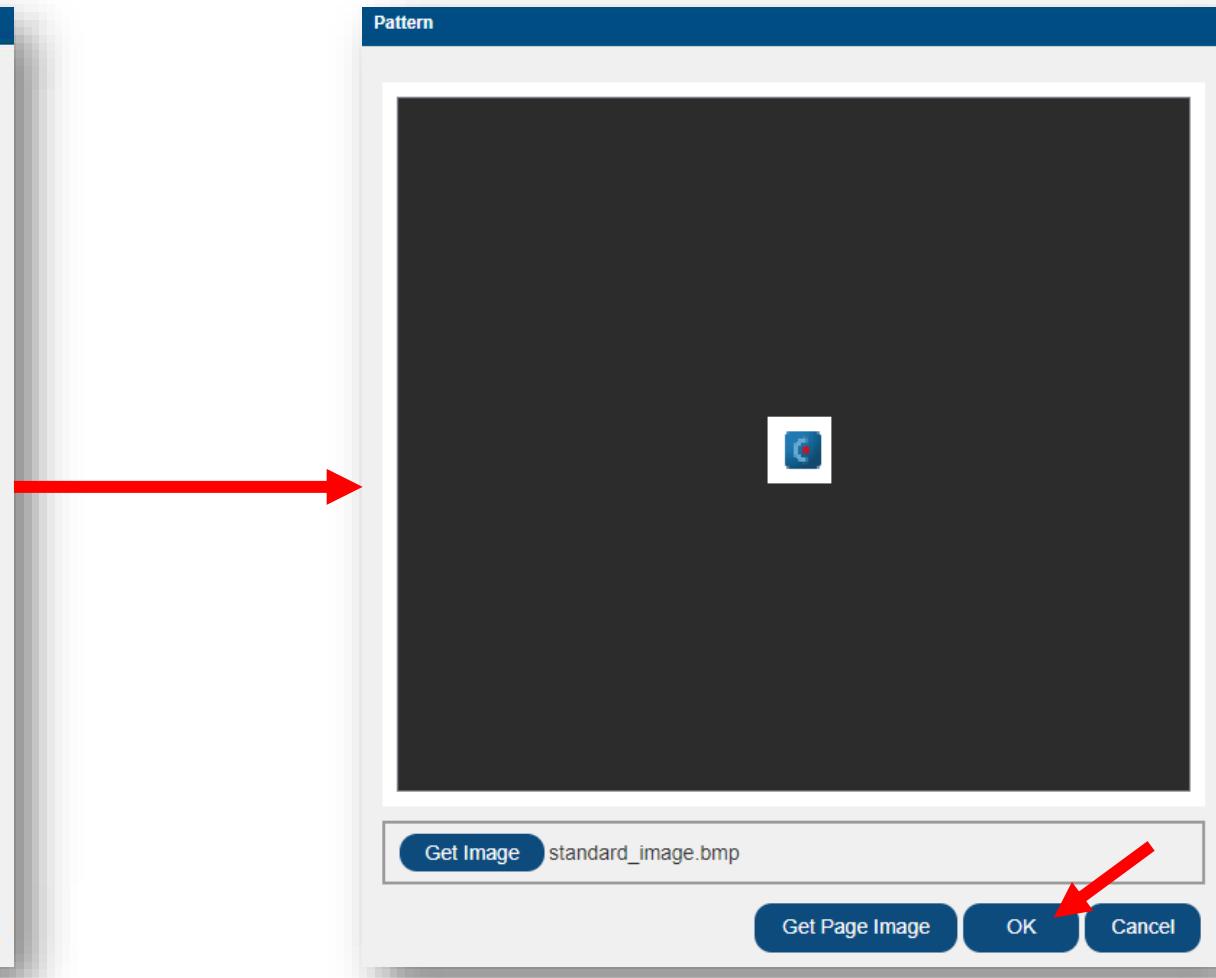
- 一般：新增樣式目標值

7. 設定比對樣式：在圖像上點擊右鍵 → 選擇建立裁切



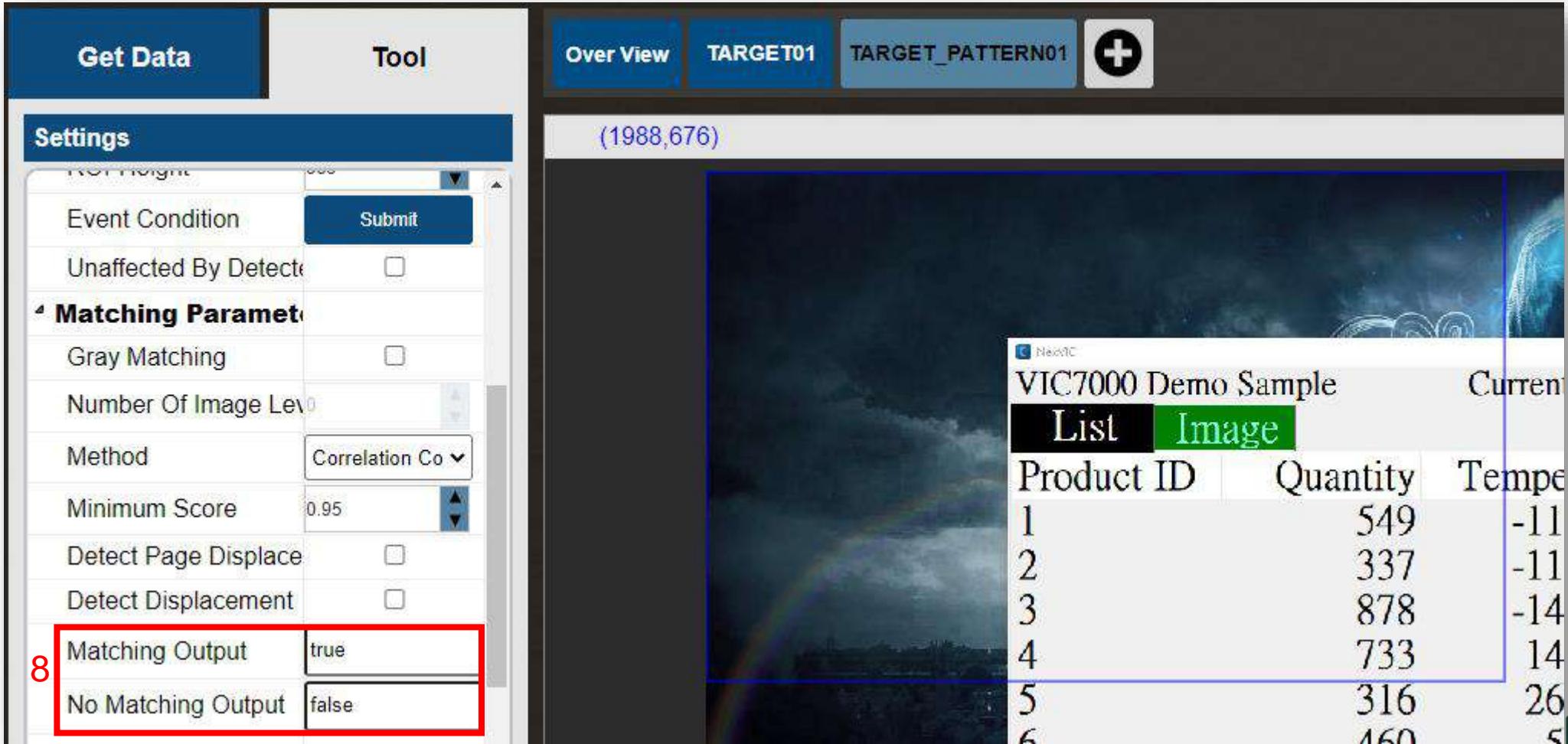
- 一般：新增樣式目標值

7. 設定比對圖案：選擇要比對的樣式 → 在圖像上點擊右鍵 → 選擇裁切 → 點擊確定



- 一般：新增樣式目標值

8. 設定比對結果輸出：輸入比對結果輸出



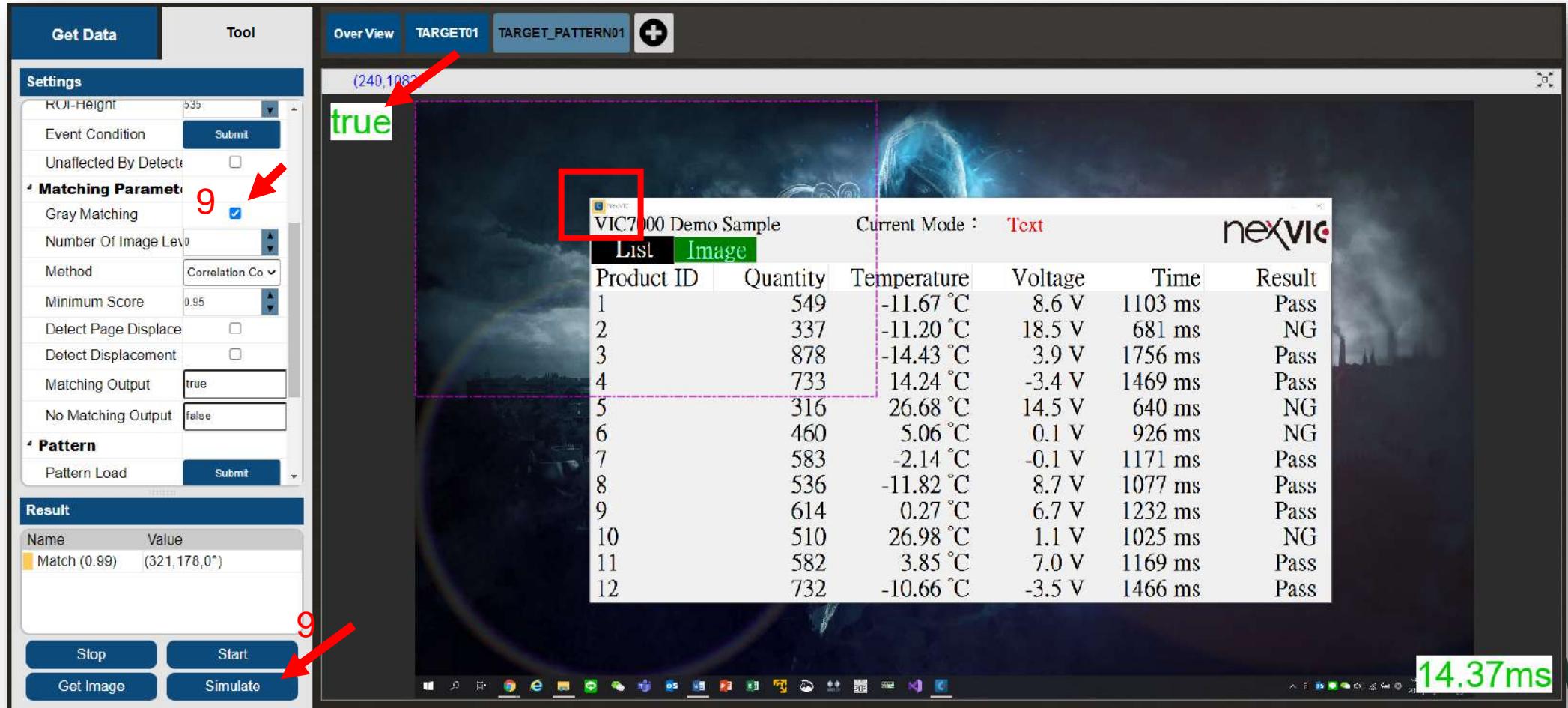
The screenshot shows the nexVIC software interface. On the left, the 'Settings' panel is open, displaying 'Matching Parameters'. A red box highlights the 'Matching Output' field, which is set to 'true'. Below it is the 'No Matching Output' field, set to 'false'. The main workspace shows a target image with a bounding box and coordinates (1988, 676). To the right is a table titled 'VIC7000 Demo Sample' with columns for Product ID, Quantity, and Temperature. The table lists six items with varying values.

Product ID	Quantity	Temperature
1	549	-11
2	337	-11
3	878	-14
4	733	14
5	316	26
6	460	5

- 一般：新增樣式目標值

9. 模擬確認：建議開啟灰階比對，加快辨識速度 → 點擊模擬 → 確認辨識結果

此結果即為樣式比對目標值



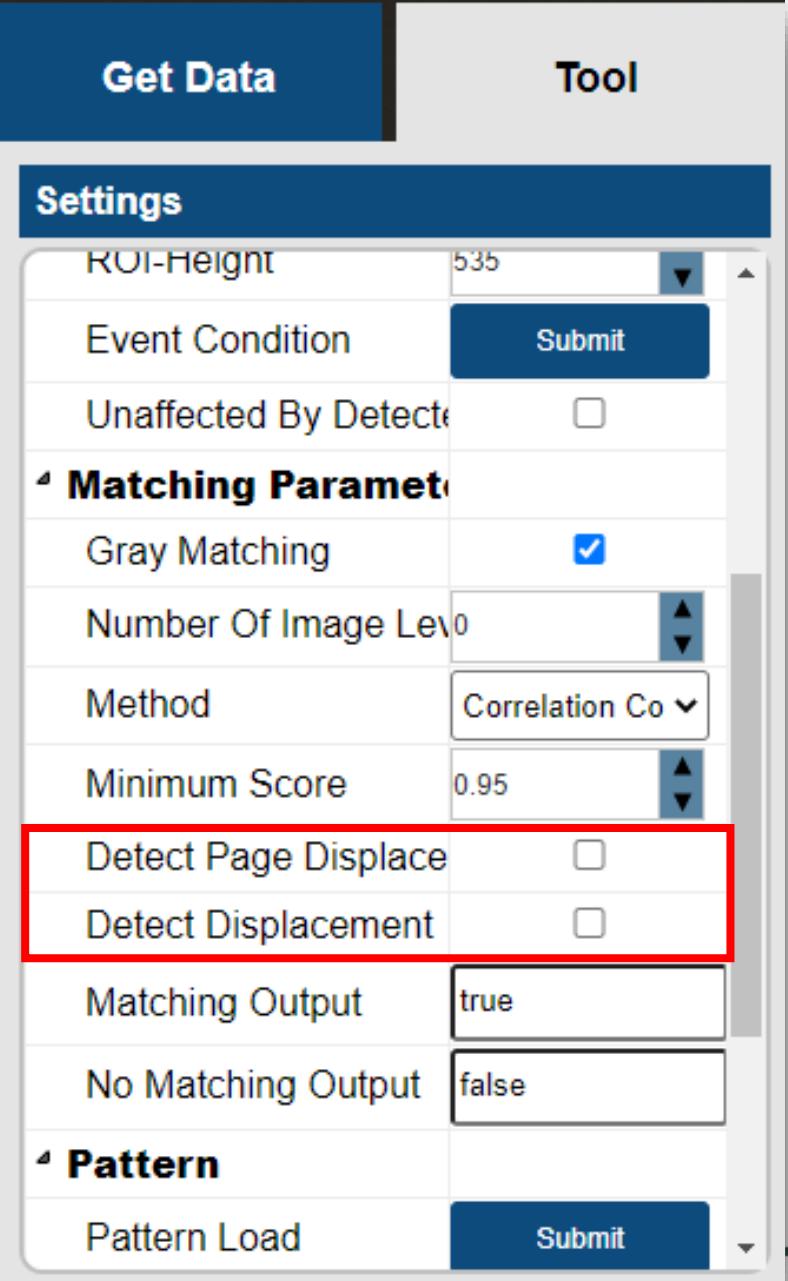
- 一般：新增樣式目標值

- 設定是否偵測位移：以設定樣式目標值的圖像上的樣式位置為基準

偵測頁面位移：讓目標值、顏色目標值的ROI隨著樣式目標值所在位置不同而移動

偵測位移：讓符合此樣式目標值的頁面內所有辨識工具的ROI隨著樣式目標值所在位置不同而移動

<b>Detect Page Displacement</b>	<input type="checkbox"/>
<b>Detect Displacement</b>	<input type="checkbox"/>

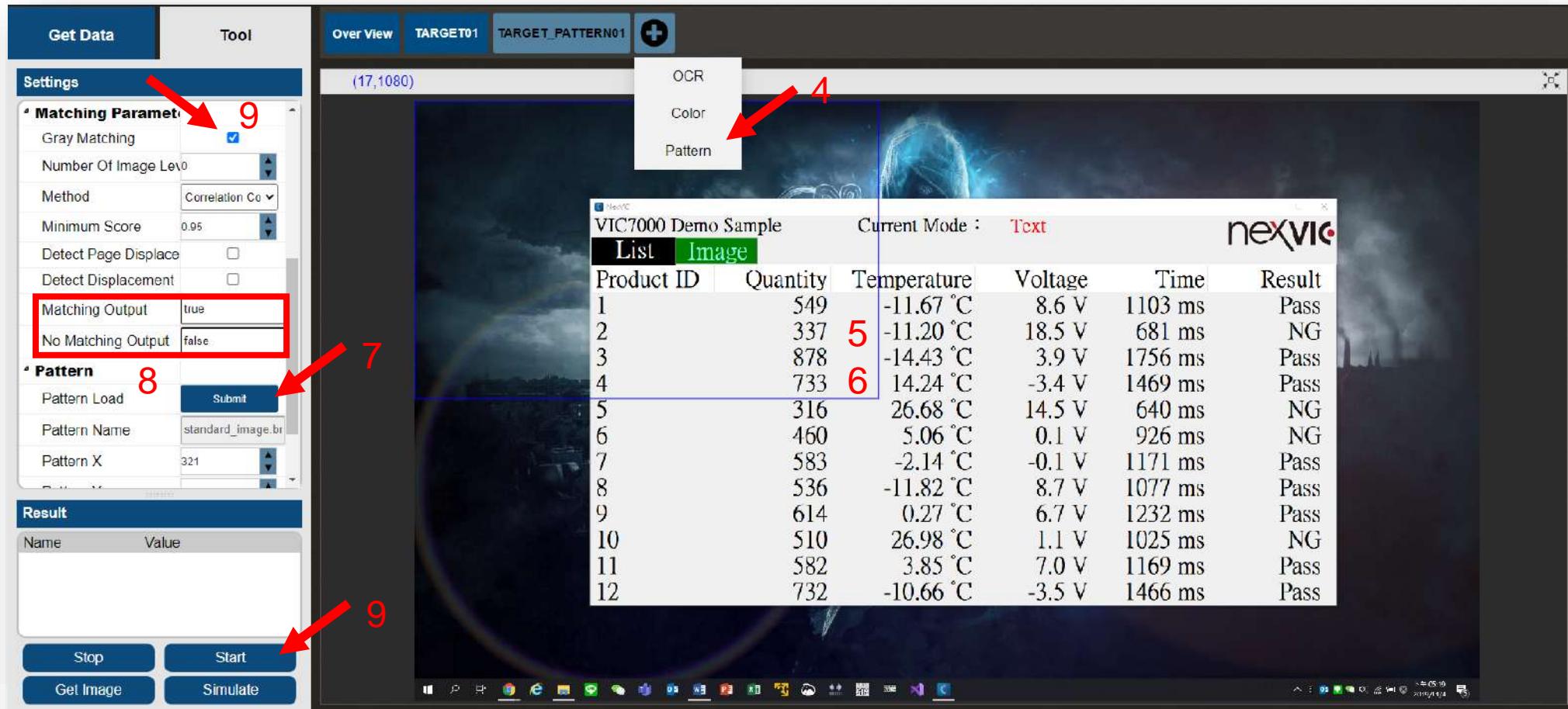


The screenshot shows the 'Settings' tab of the nexVIC software. The interface includes the following fields:

- ROI-Height: 535
- Event Condition:
- Unaffected By Detection:
- Matching Parameters:**
  - Gray Matching:
  - Number Of Image Levels:
  - Method: Correlation Co.
  - Minimum Score: 0.95
- Detect Page Displacement:**
- Detect Displacement:**
- Matching Output: true
- No Matching Output: false
- Pattern:**
  - Pattern Load:

- 一般：新增樣式目標值

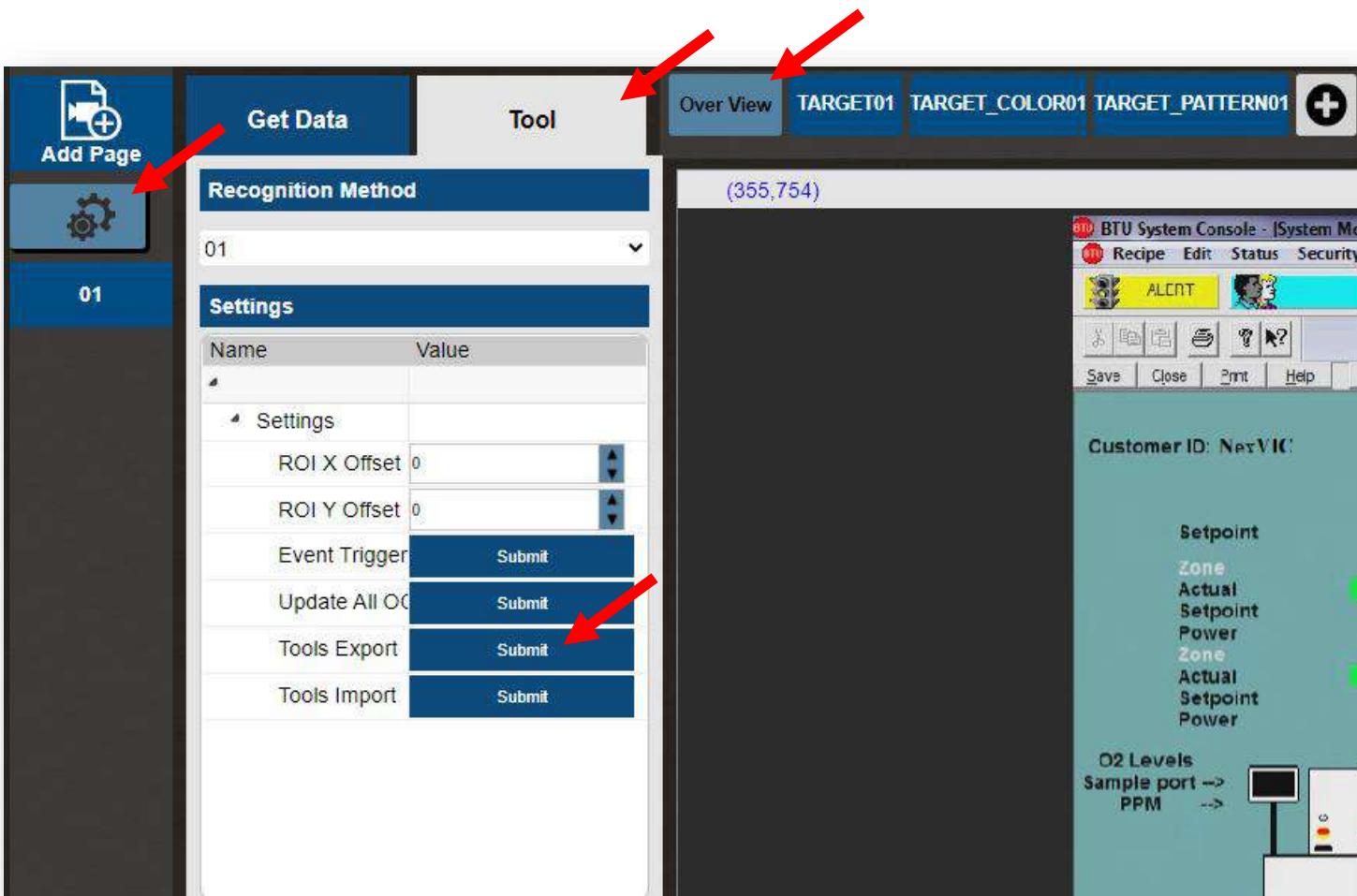
- 多個樣式目標值：在同一頻道的頁面設定中使用新增按鈕 → 依照前述步驟可新增更多樣式目標值



# 目標值

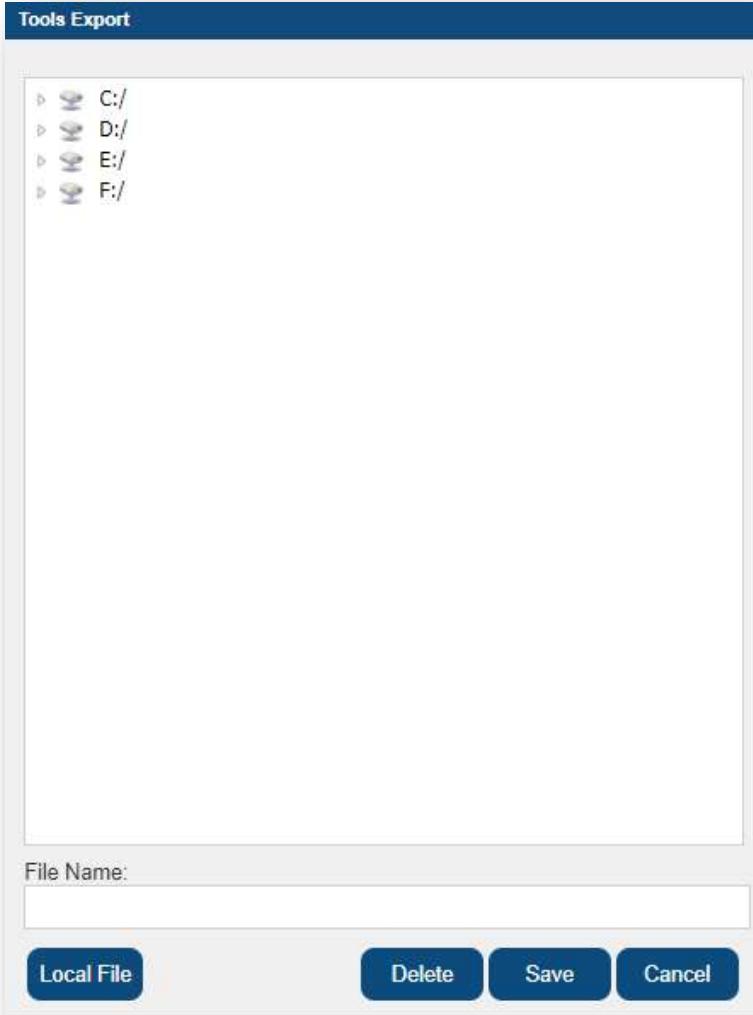
- 工具匯出：匯出辨識工具設定

點擊工具匯出按鈕 → 選擇儲存類型 → 點擊確定



- 工具匯出：匯出辨識工具設定

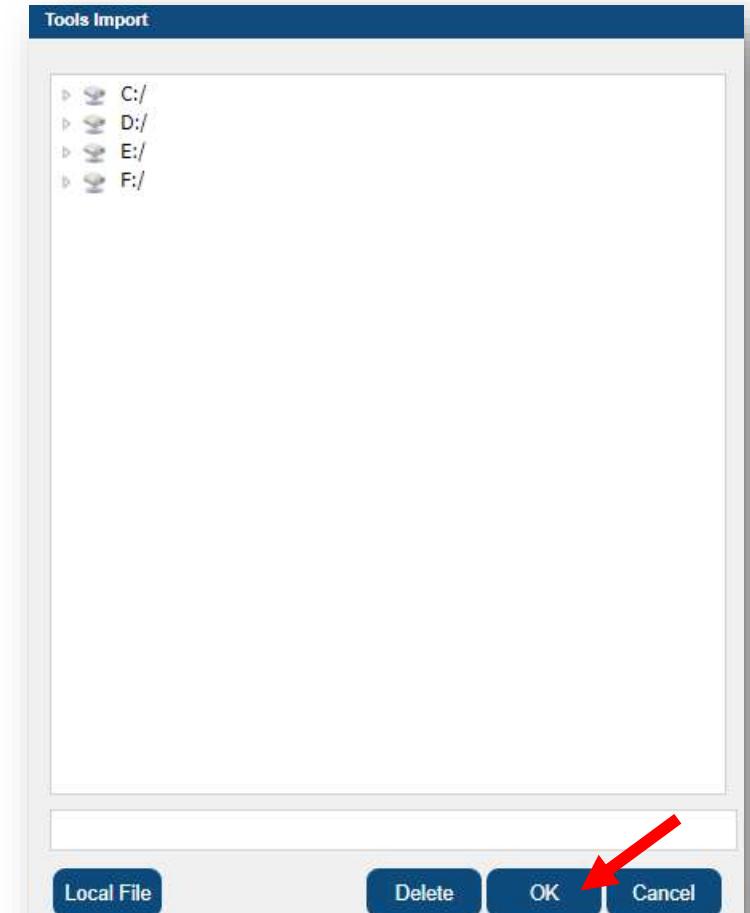
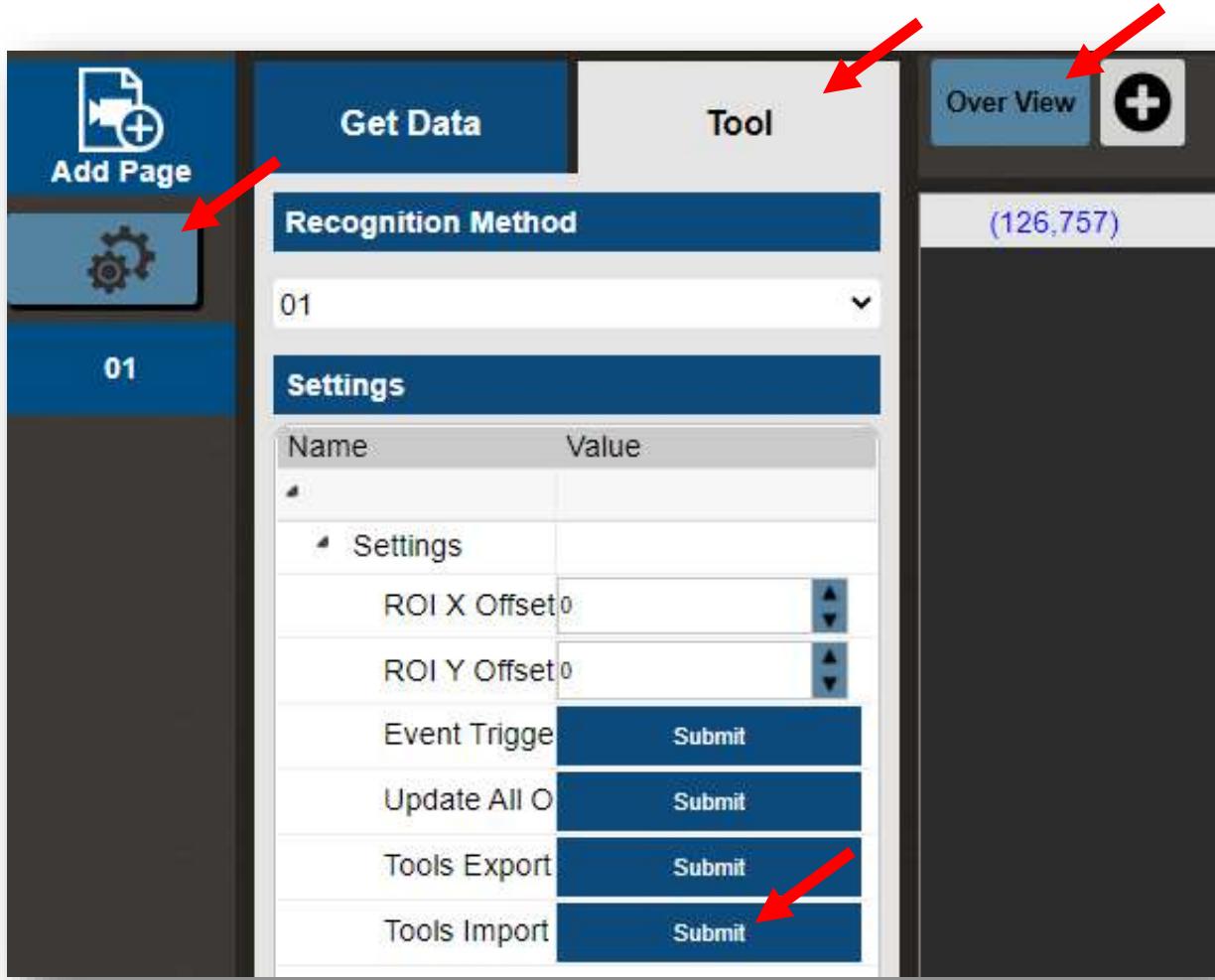
選擇儲存位址，匯出檔案的副檔名是 tools，內容為 json 字串



```
{  
  "tools": [  
    {  
      "roi": false,  
      "roi_x": 0,  
      "roi_y": 0,  
      "roi_width": 0,  
      "roi_height": 0,  
      "type": "OCR",  
      "id": "TARGET01",  
      "segmentation": 7,  
      "preprocess_resize": 0,  
      "preprocess_resize_method": 0,  
      "preprocess_threshold_method": 1,  
      "preprocess_threshold_algorithm":  
      "preprocess_threshold_value": 120,  
      "allow_recognition_rate": 80,  
      "allow_empty_string": 0,  
      "allow_remove_whitespace": 0,  
      "font_select": 1,  
      "font_file": ""  
    },  
    {  
      "roi": false,  
      "roi_x": 0,  
      "roi_y": 0,  
      "roi_width": 0,  
      "roi_height": 0,  
      "type": "COLOR",  
      "id": "TARGET_COLOR01",  
      "red": 0,  
      "green": 0,  
      "blue": 0,  
      "tolerance": 5,  
      "matching_output": "true",  
      "no_matching_output": "false"  
    },  
    {  
      "roi": false,  
      "roi_x": 0,  
      "roi_y": 0,  
      "roi_width": 0,  
      "roi_height": 0,  
      "type": "PATTERN",  
      "id": "TARGET_PATTERN01",  
      "gray_match": 0,  
      "pattern_name": "",  
      "pattern_x": 0,  
      "pattern_y": 0,  
      "pattern_width": 0,  
      "pattern_height": 0,  
      "min_score": 0.95,  
      "detect_page_displacement": 0,  
      "detect_displacement": 0,  
      "matching_output": "true",  
      "no_matching_output": "false",  
      "pattern_img": ""  
    }  
  ]  
}
```

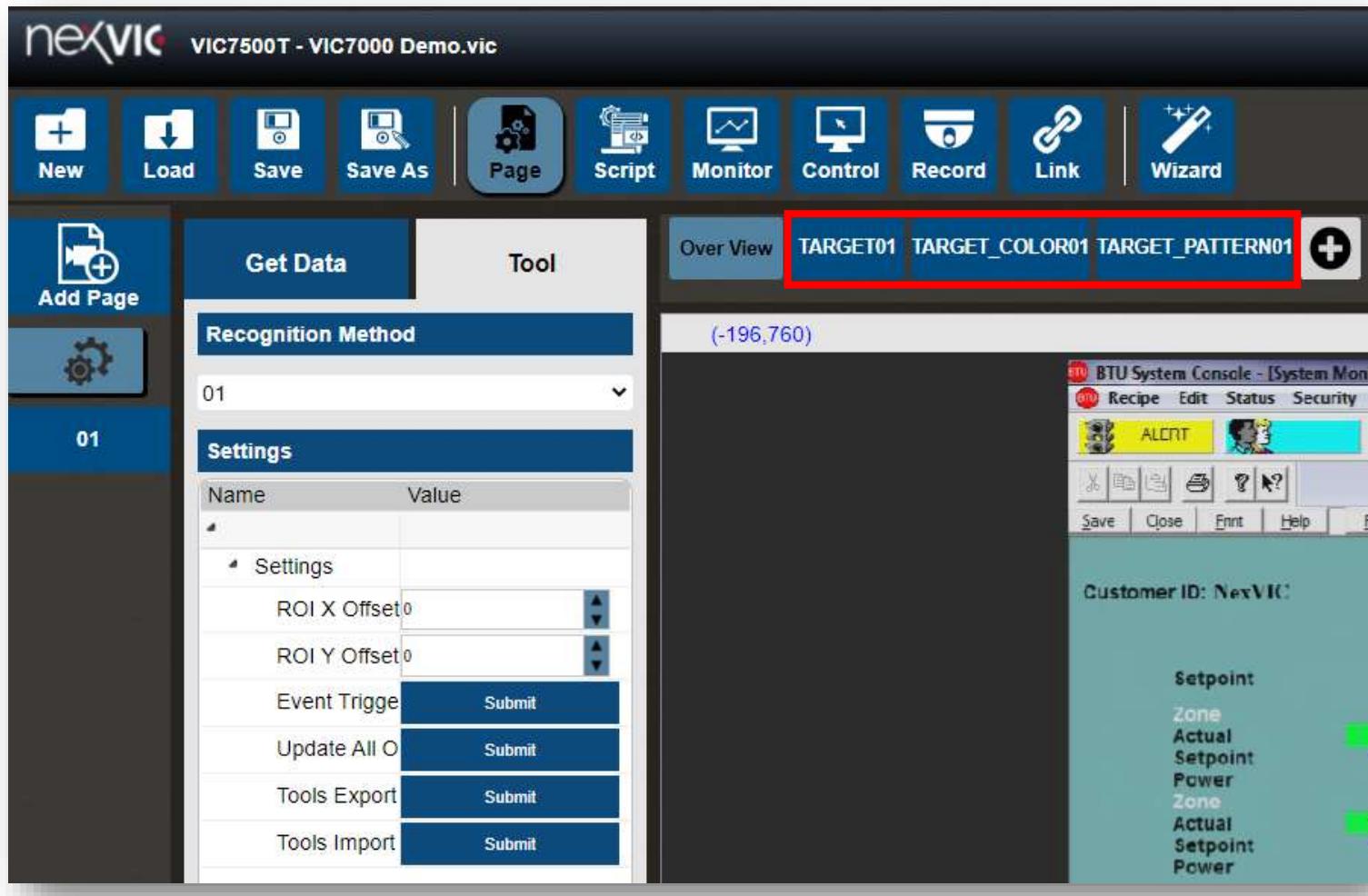
- 工具匯入：匯入辨識工具設定

點擊工具匯入按鈕 → 選擇編輯好的 tools 檔案 → 點擊確定



## • 工具匯入：匯入辨識工具設定

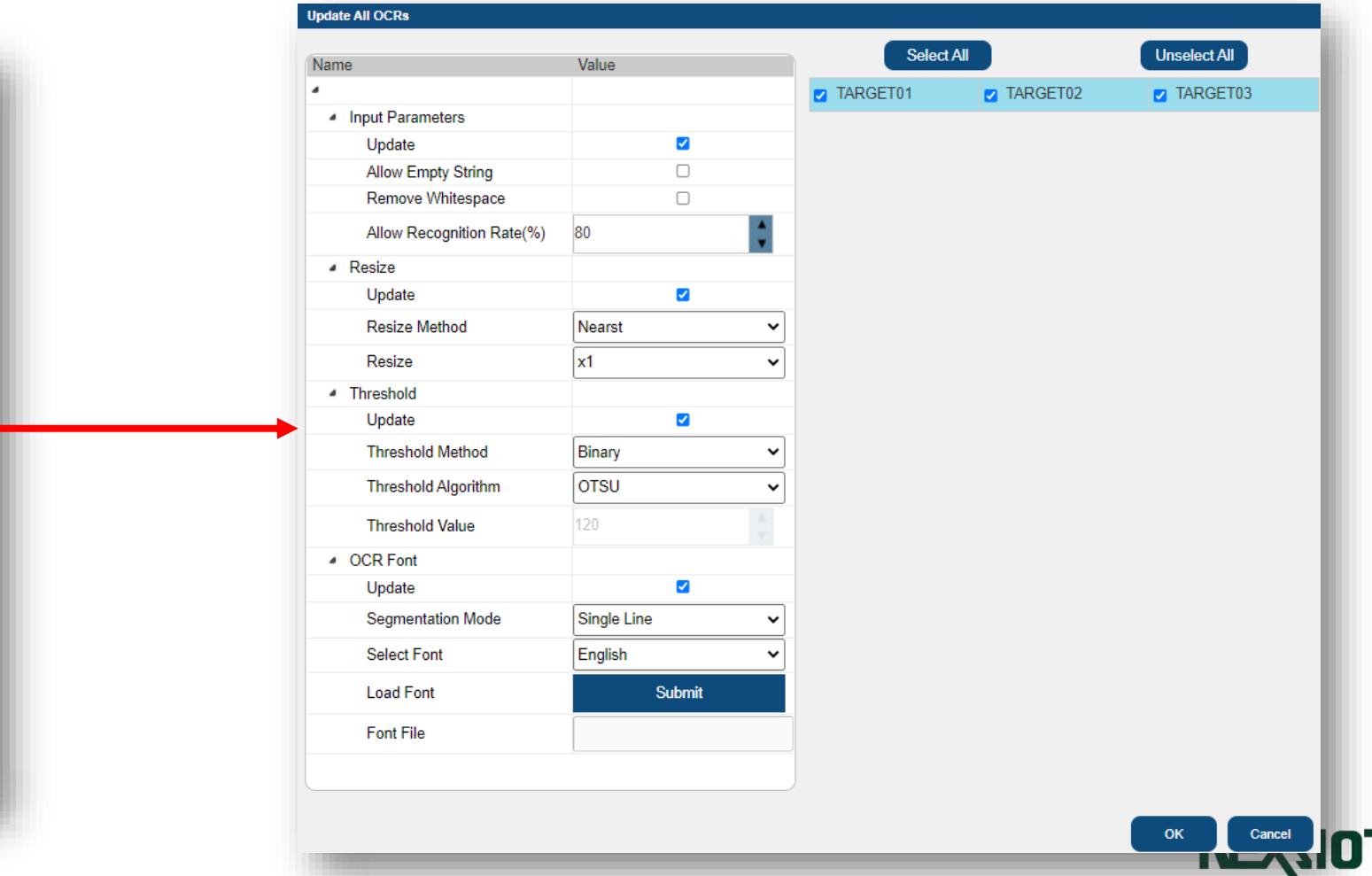
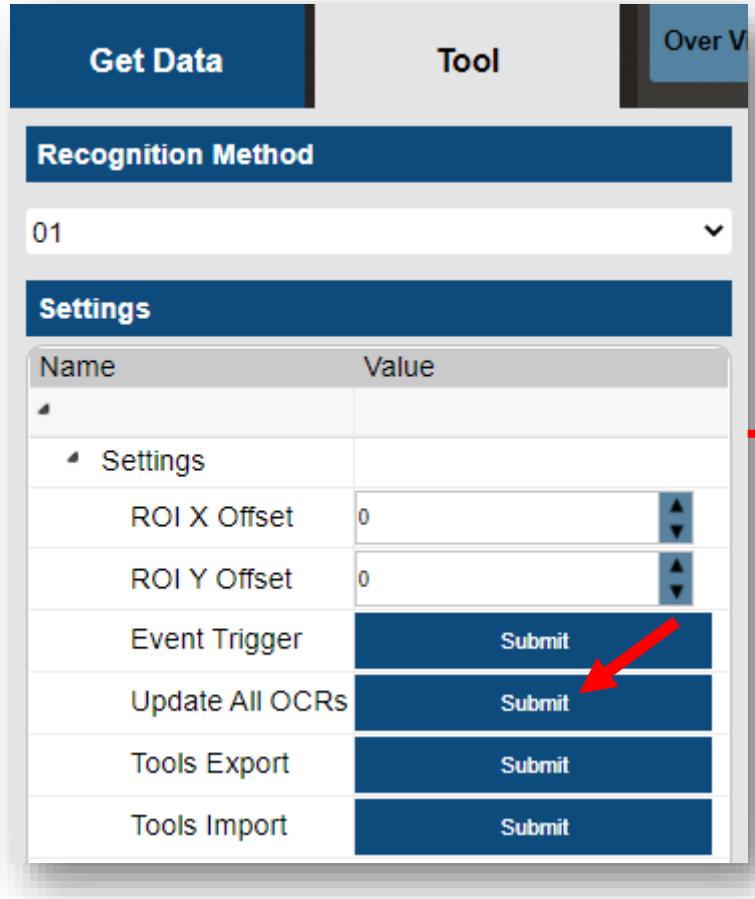
系統會依照匯入的 tools 檔案自動建立辨識工具



# 目標值

- 更新全部OCR

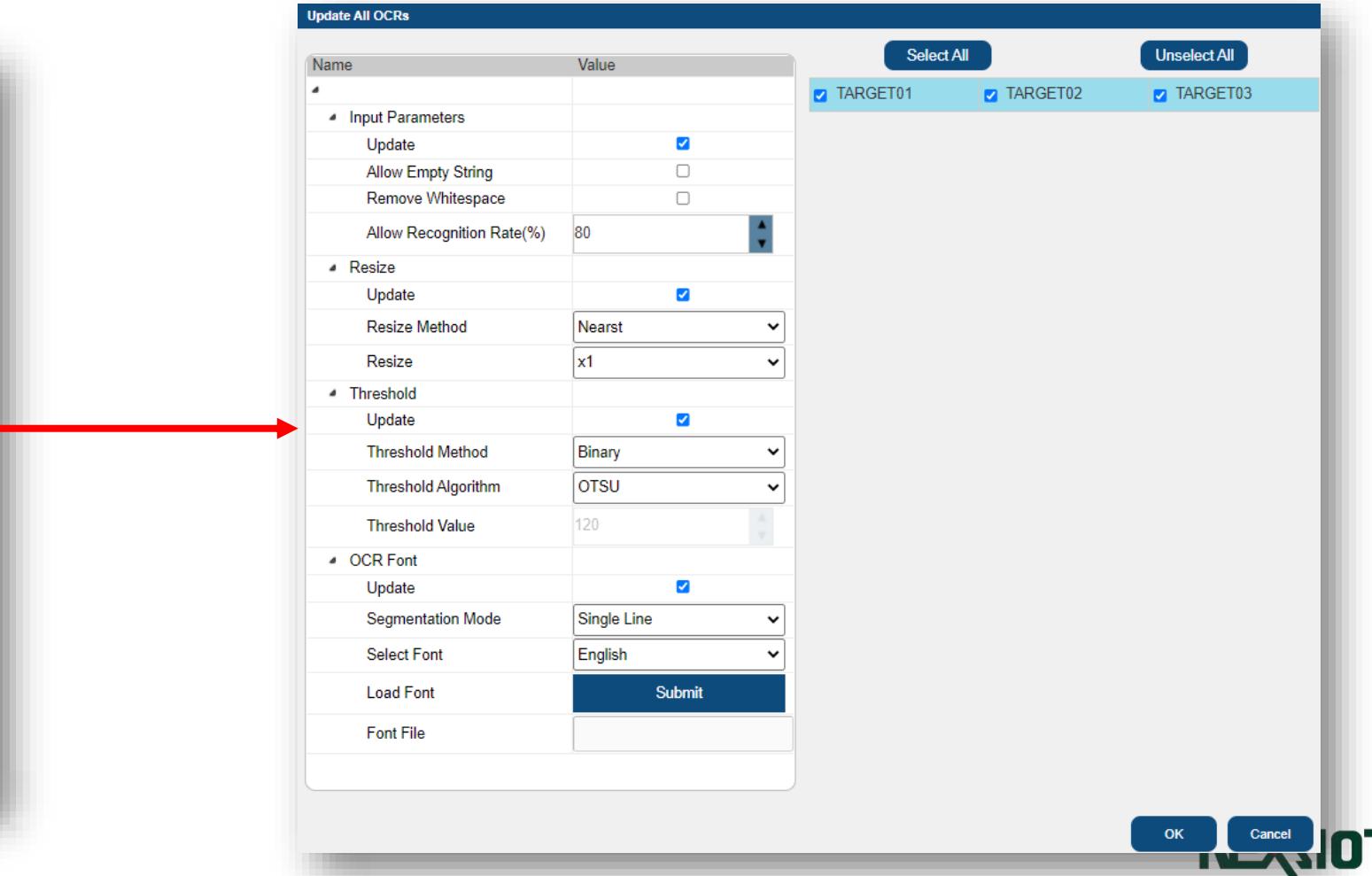
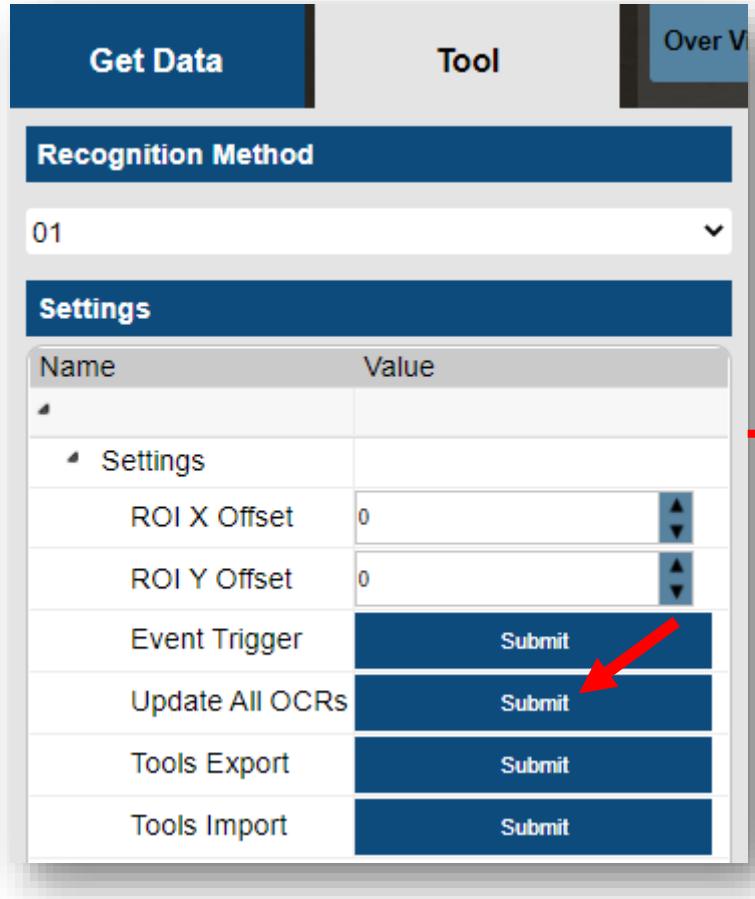
點擊更新全部OCR，可以一次更新全部的TARGET設定



# 目標值

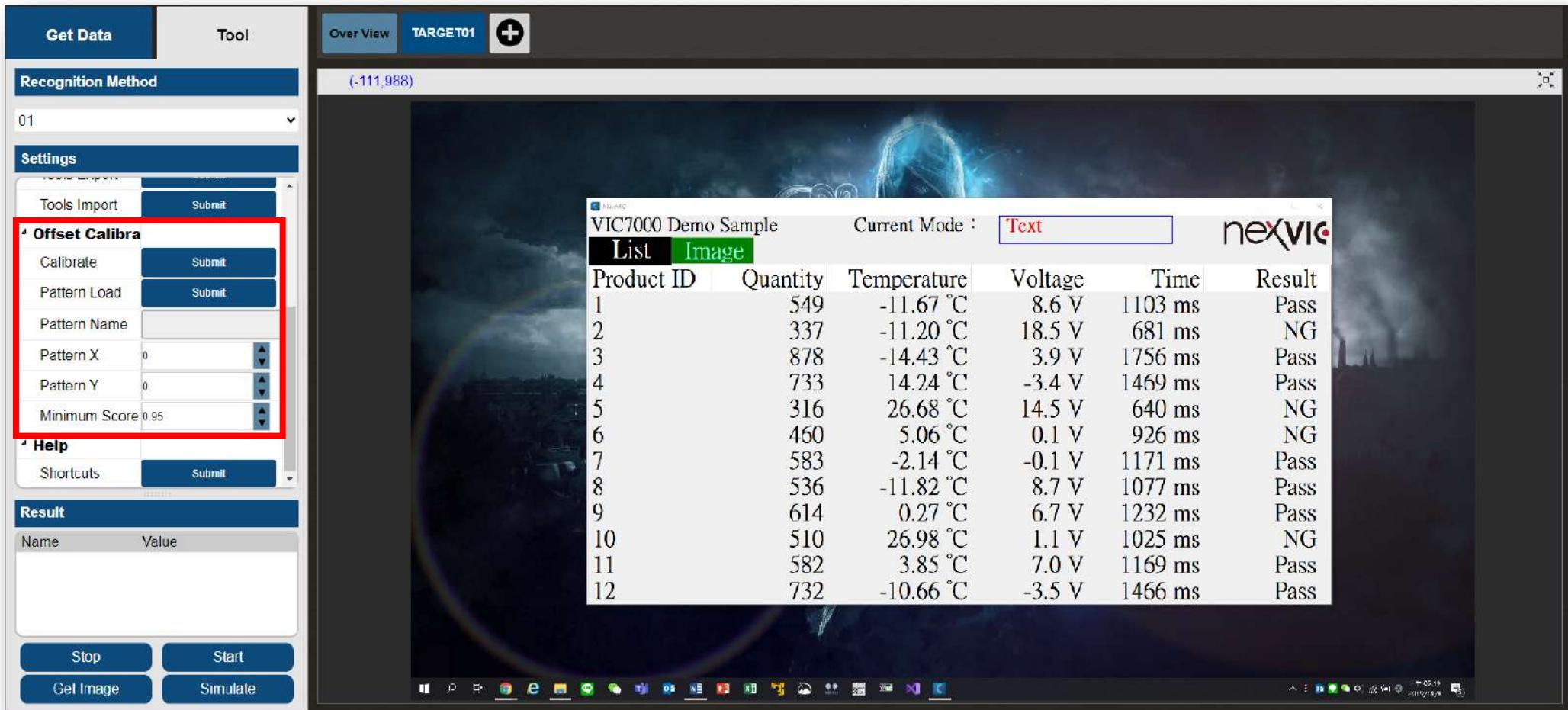
- 更新全部OCR

點擊更新全部OCR，可以一次更新全部的TARGET設定



- 位移校正

設置樣式比對標準，往後若載入辨識區域有位移的圖像，可使用位移校正來偵測ROI X位移和ROI Y位移，即可使用有位移的圖像進行辨識設定



# 目標值

- 位移校正

點擊樣式載入，載入欲比對的樣式

Get Data      Tool

Recognition Method

01

Settings

Tools Export   Tools Import

Tools Import   Submit

Offset Calibration

Calibrate   Submit

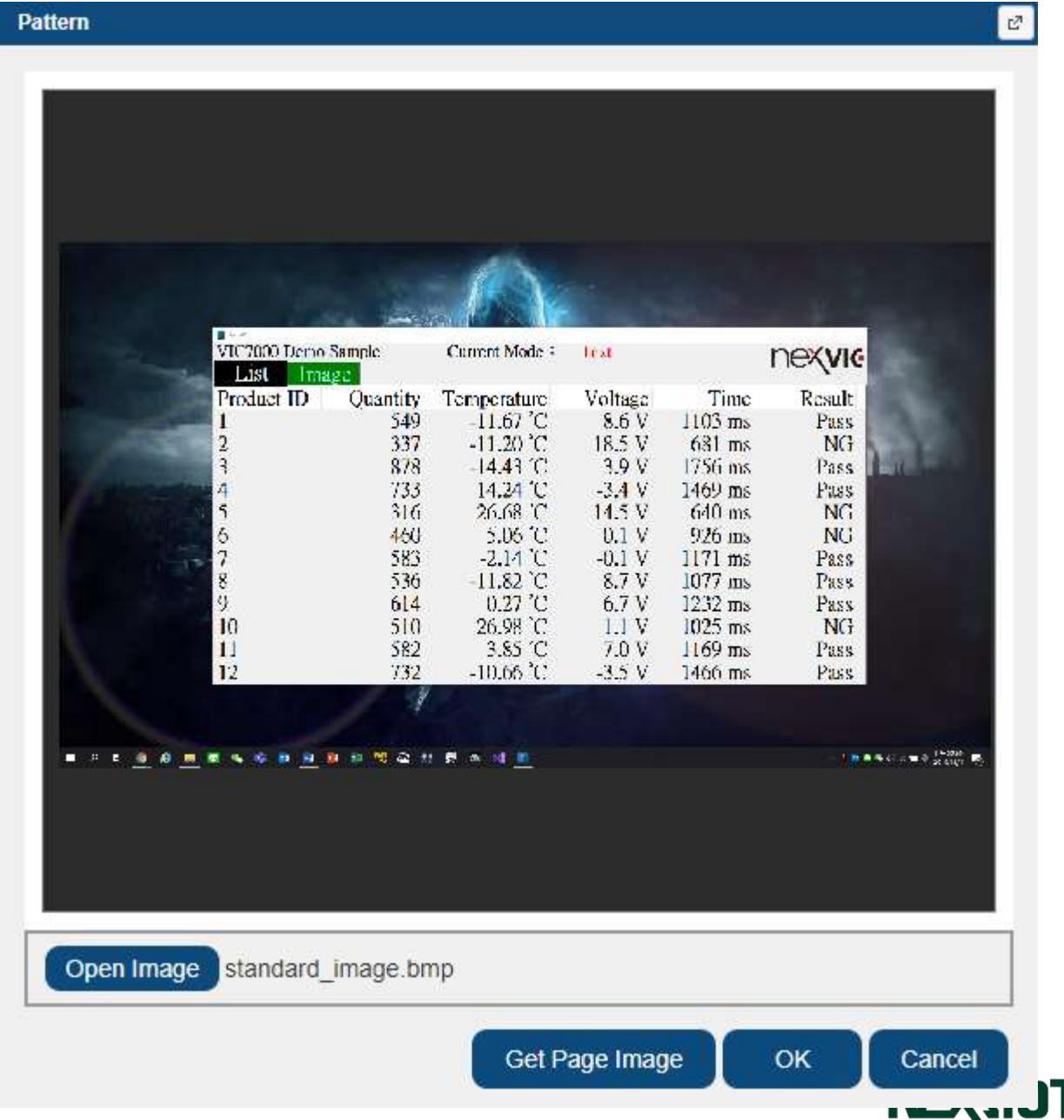
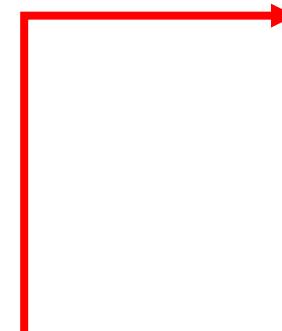
Pattern Load   Submit

Pattern Name

Pattern X 0

Pattern Y 0

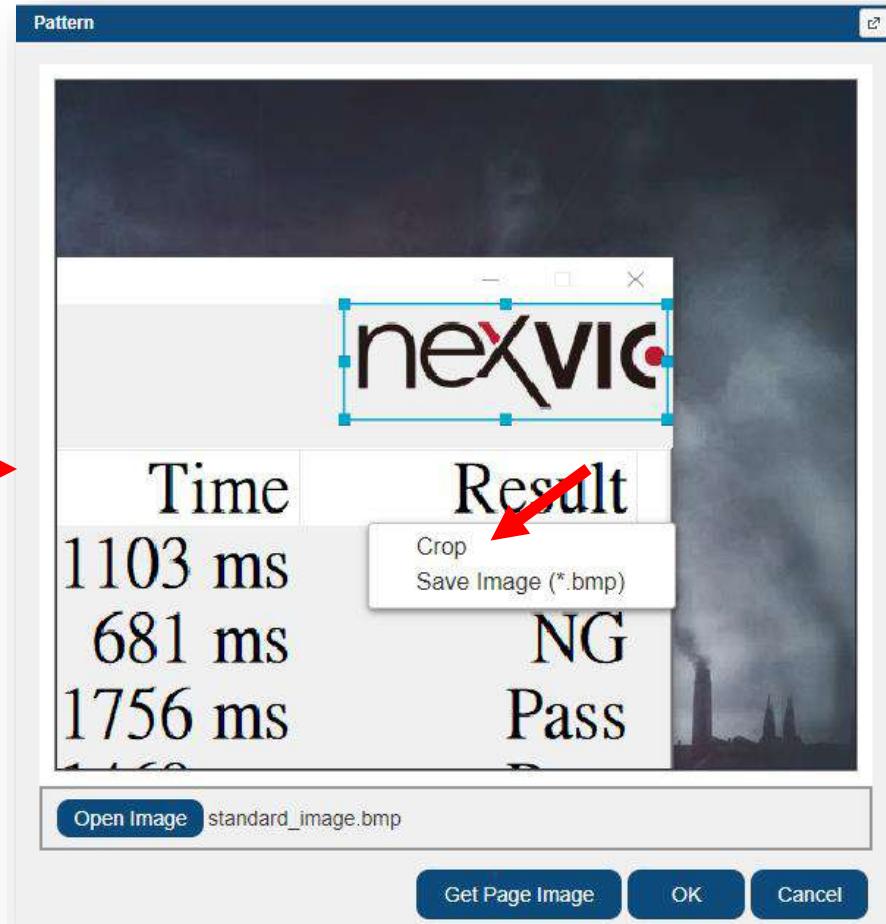
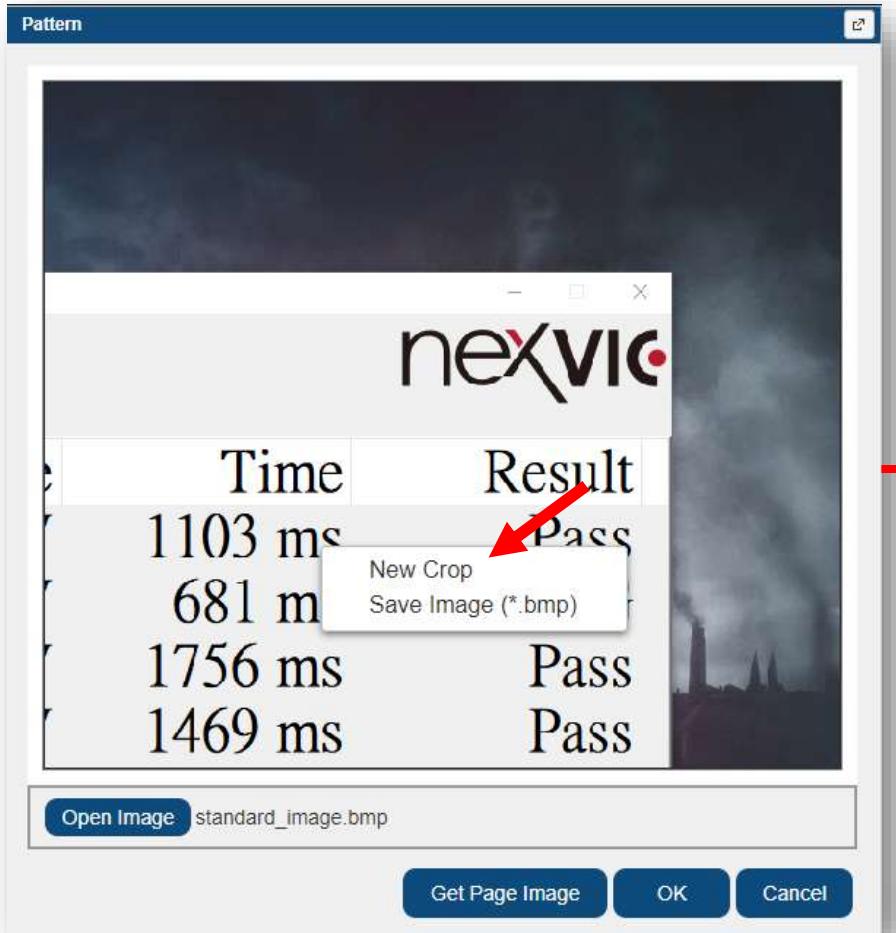
Minimum Score 0.95



# 目標值

- 位移校正

點擊右鍵並選擇建立裁切，選擇要使用的樣式後，點擊右鍵並選擇裁切



# 目標值

- 位移校正

確認比對樣式後，點擊確定，系統同時會載入樣式 X 和樣式 Y，即為比對樣式的座標



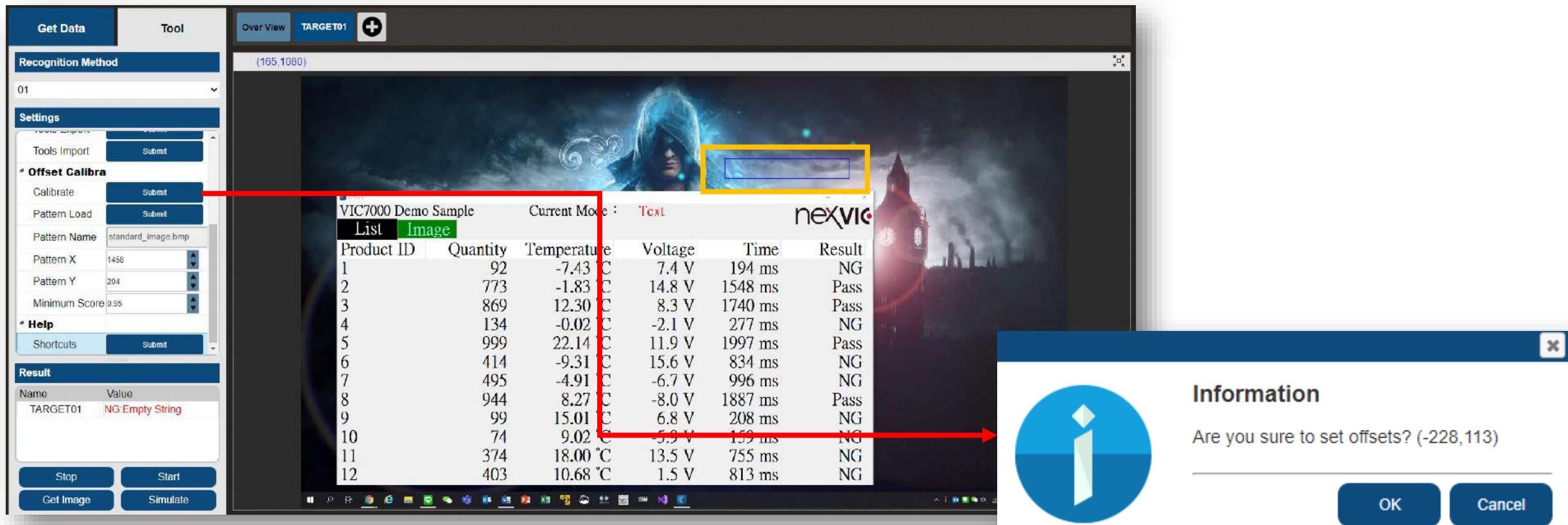
Offset Calibra

Calibrate	Submit
Pattern Load	Submit
Pattern Name	standard_image.bmp
Pattern X	1458
Pattern Y	204
Minimum Score	0.95

# 目標值

- 位移校正

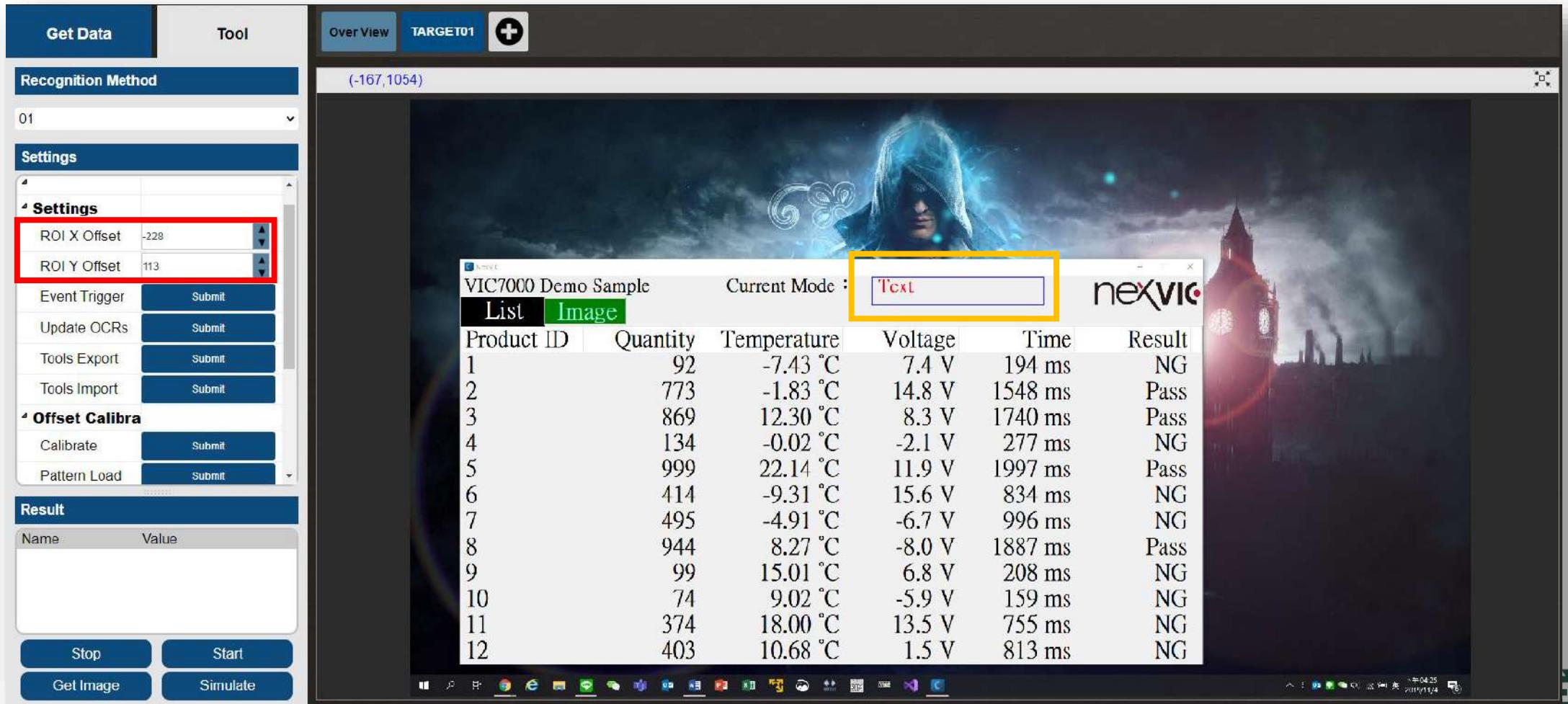
載入有辨識區域位移的圖像，點擊校正，系統會自動計算位移量，點擊確定



# 目標值

- 位移校正

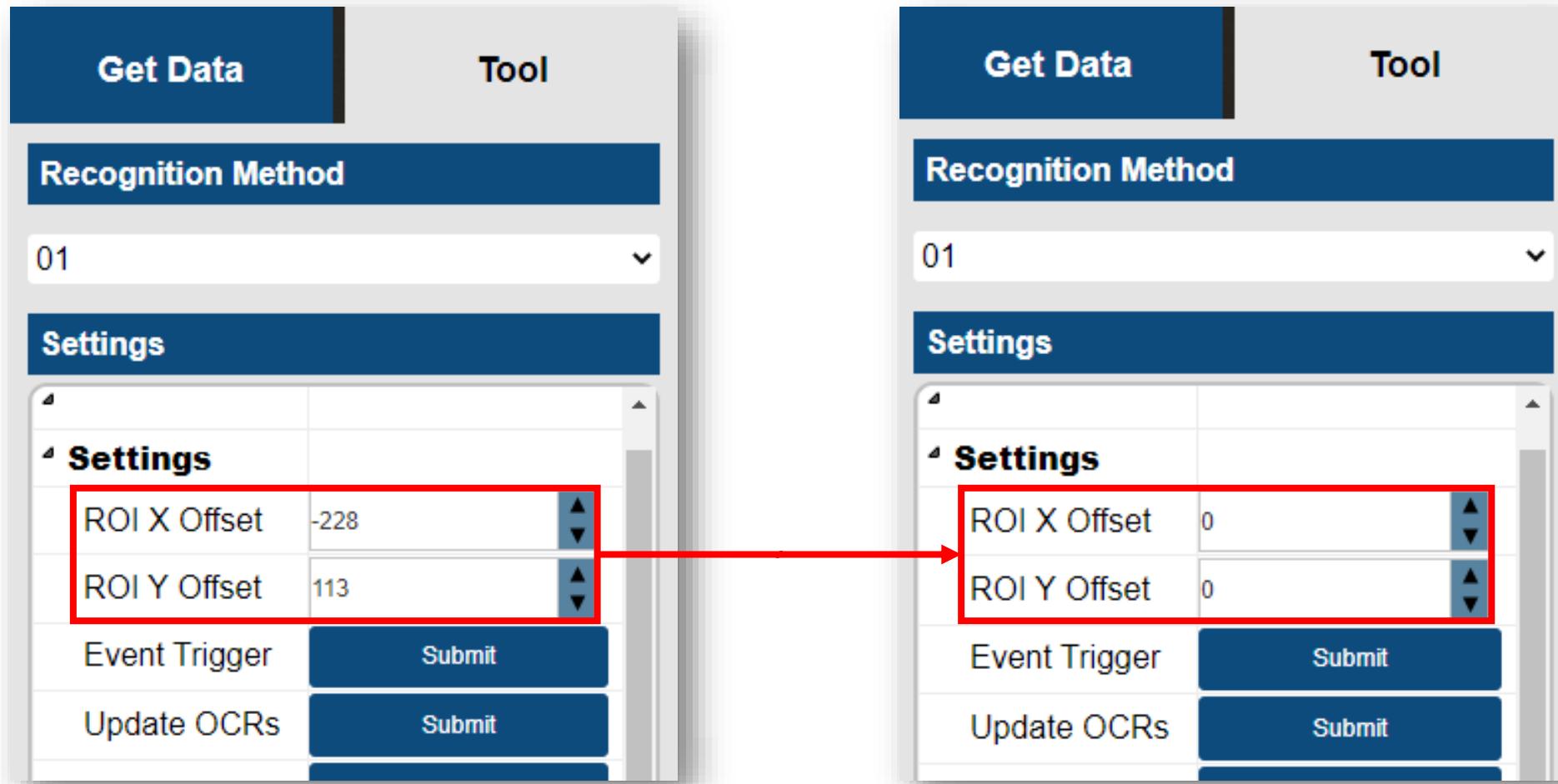
點擊確定後，系統會將位移量帶入ROI X位移和ROI Y位移，即可用有位移的圖像進行辨識設定



# 目標值

- 位移校正

辨識設定結束後，建議將ROI X位移和ROI Y位移復歸為0，避免實際辨識時錯誤



# Outline

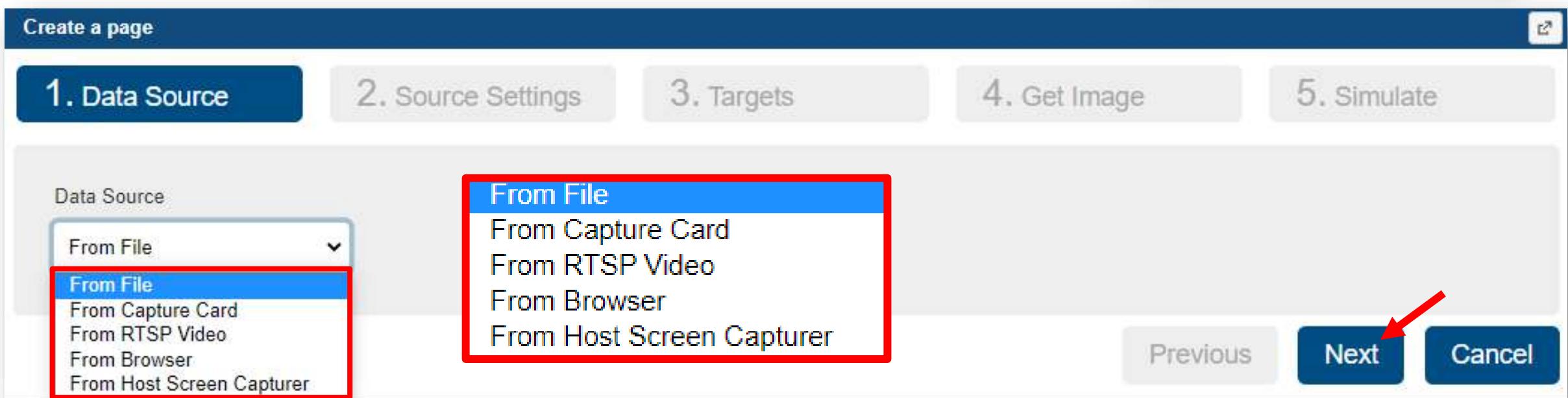
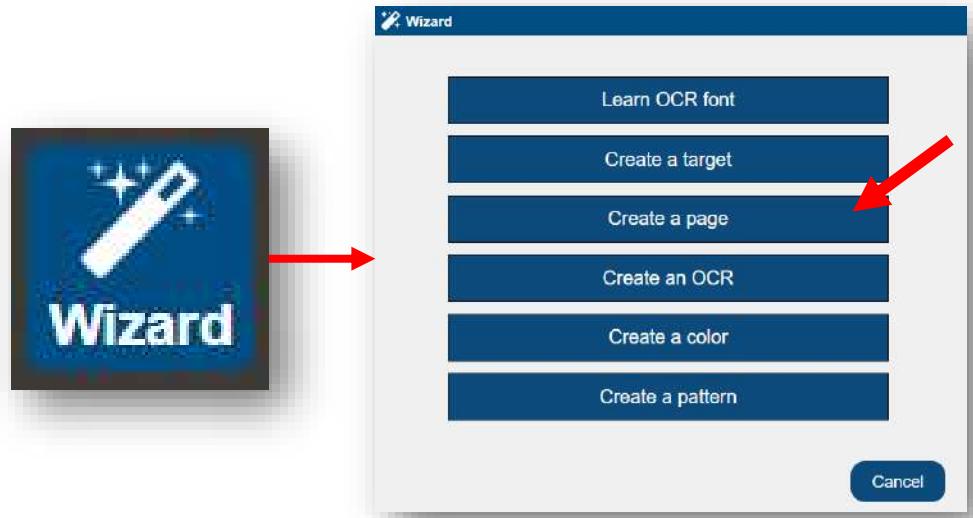
- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

- 精靈：新增頁面

1. 影像來源

來自檔案、來自擷取卡、來自 RTSP 影像

來自瀏覽器、來自本機畫面擷取

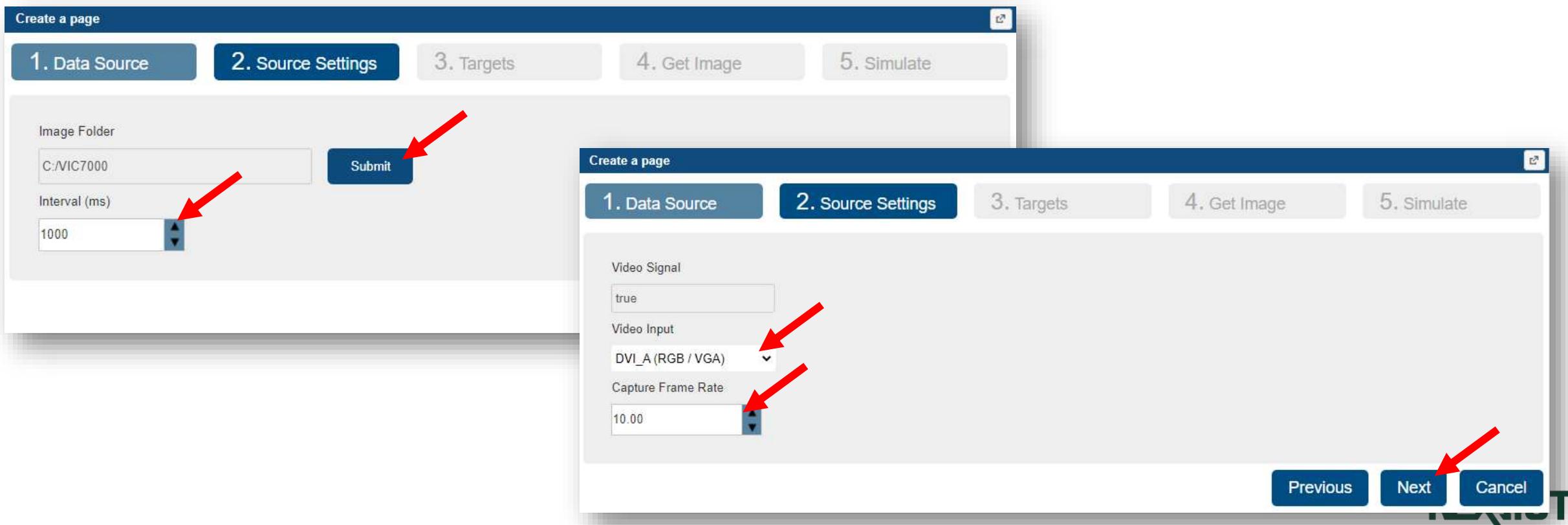


- 精靈：新增頁面

- 2. 影像來源設定

- 來自檔案：圖檔路徑、讀取時間間隔

- 來自擷取卡：輸入訊號類型、擷取頻率(fps)



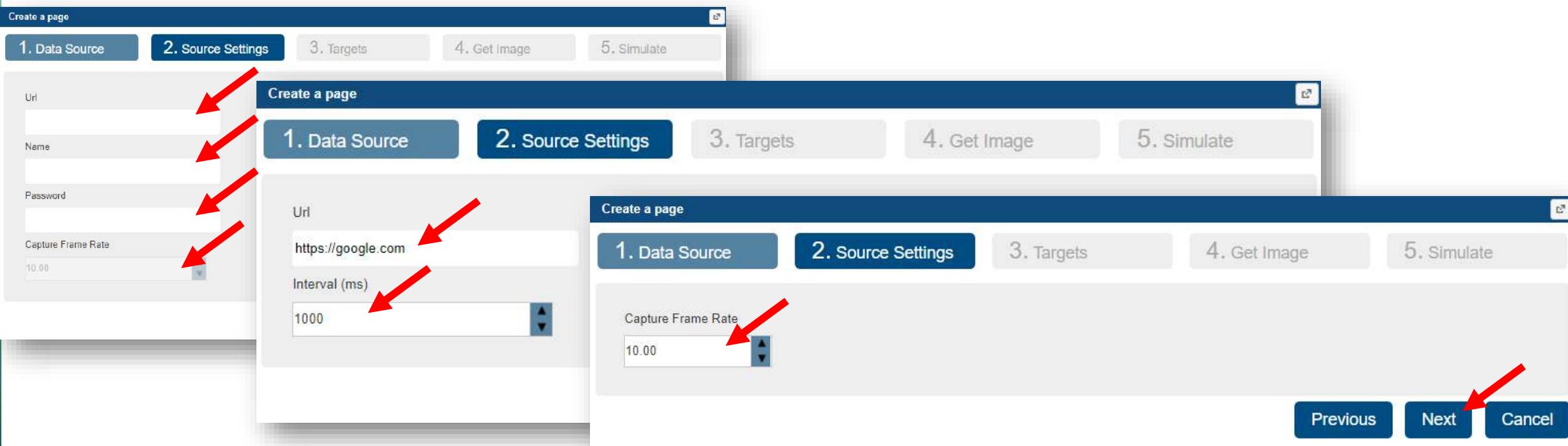
- 精靈：新增頁面

## 2. 影像來源設定

來自 RTSP 影像：RTSP 來源網址、名稱、密碼、擷取頻率

來自瀏覽器：網址、讀取時間間隔

來自本機畫面擷取：擷取頻率(fps)



- 精靈：新增頁面

- 3. 目標值：設定目標值比對條件

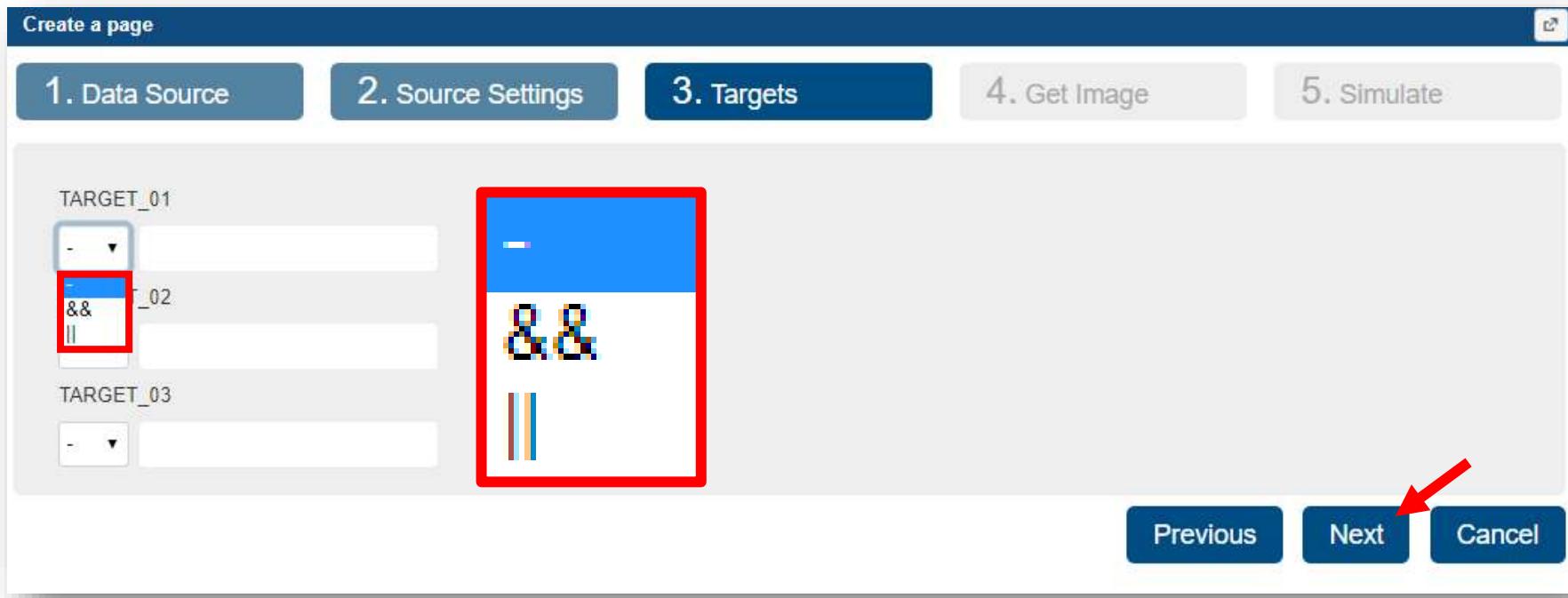
比對邏輯運算子：

若為  $-$ ，不比對此目標值

先比對  $\parallel$ ，若有一設定  $\parallel$  的目標值符合，則此頁面比對成功，使用此頁面

$\parallel$  比對失敗，再比對  $\&\&$ ，需全部設定  $\&\&$  的目標值符合，此頁面才比對成功，才使用此頁面

若有多個頁面同時比對成功，會使用編號較小的頁面



- 精靈：新增頁面

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Stop	&&	Operate	
PAGE02		Text	&&	Run	&&	Recipe	使用此頁面

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Stop	&&	Operate	
PAGE02		Image	&&	Run	&&	Operate	使用此頁面

- 精靈：新增頁面

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Run	&&	Operate	使用此頁面
PAGE02		Text	&&	Stop	&&	Recipe	

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Run	&&	Operate	使用此頁面
PAGE02		Image	&&	Run	&&	Operate	

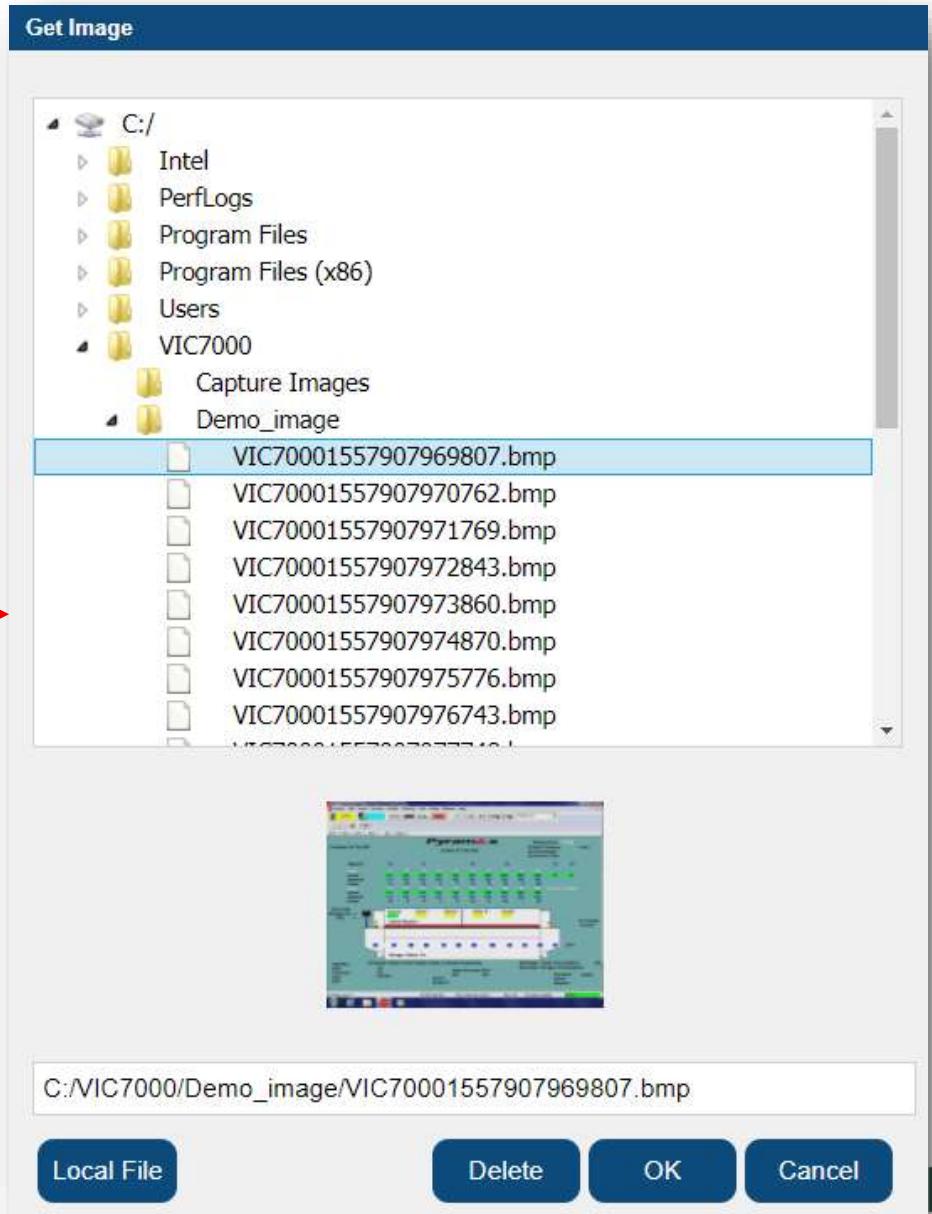
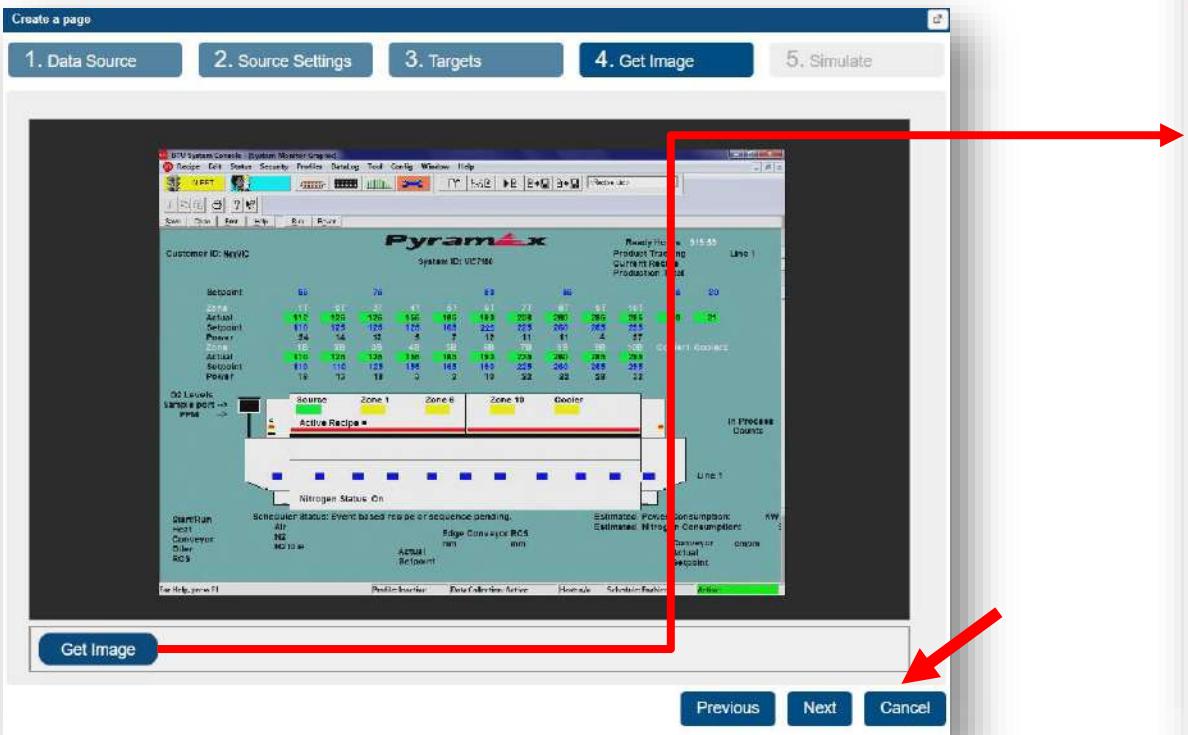
## • 精靈：新增頁面

### 4. 取得圖像

來自檔案：讀取本機或遠端的圖像

來自擷取卡、來自 RTSP 影像、來自瀏覽器、

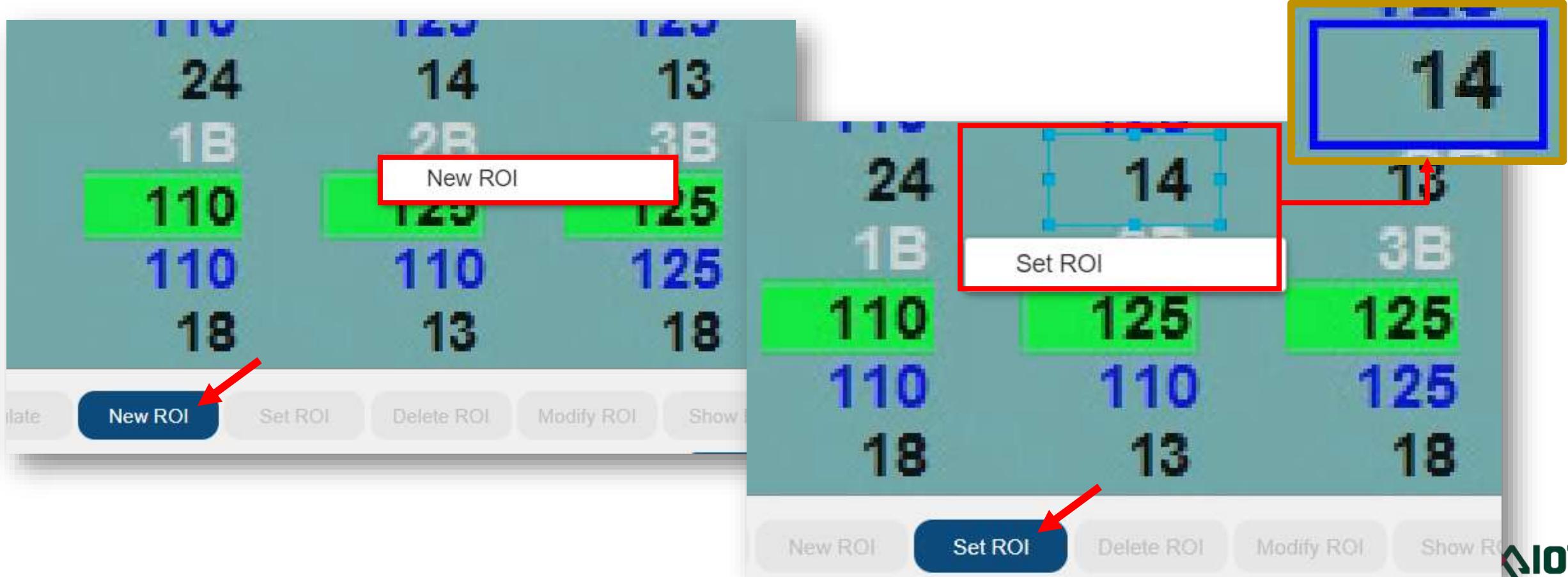
來自本機畫面擷取：從取像來源取得一張圖像



- 精靈：新增頁面

## 5. 模擬

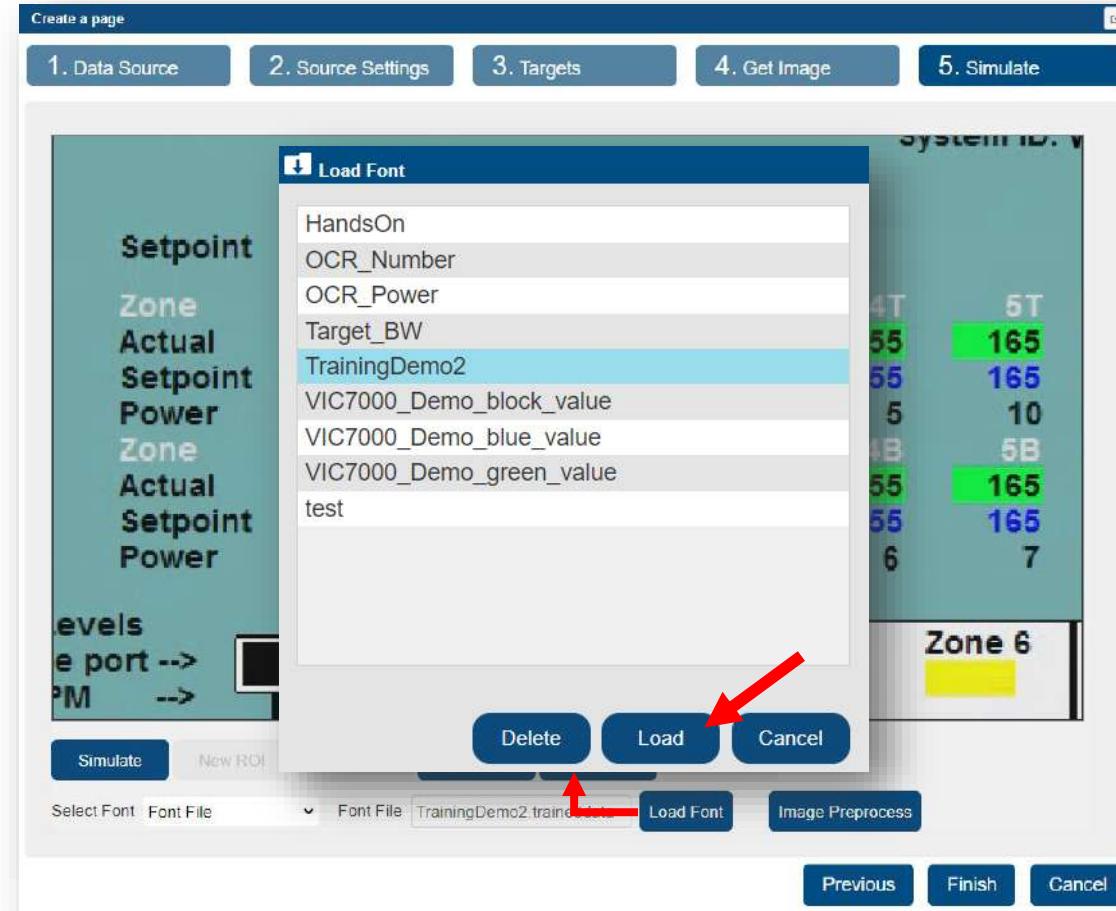
右鍵新增ROI or 點擊 新增ROI → 調整大小和位置 → 右鍵設置ROI or 點擊設置ROI



- 精靈：新增頁面

## 5. 模擬

載入字型 → 選擇要使用的字型檔 → 點擊載入

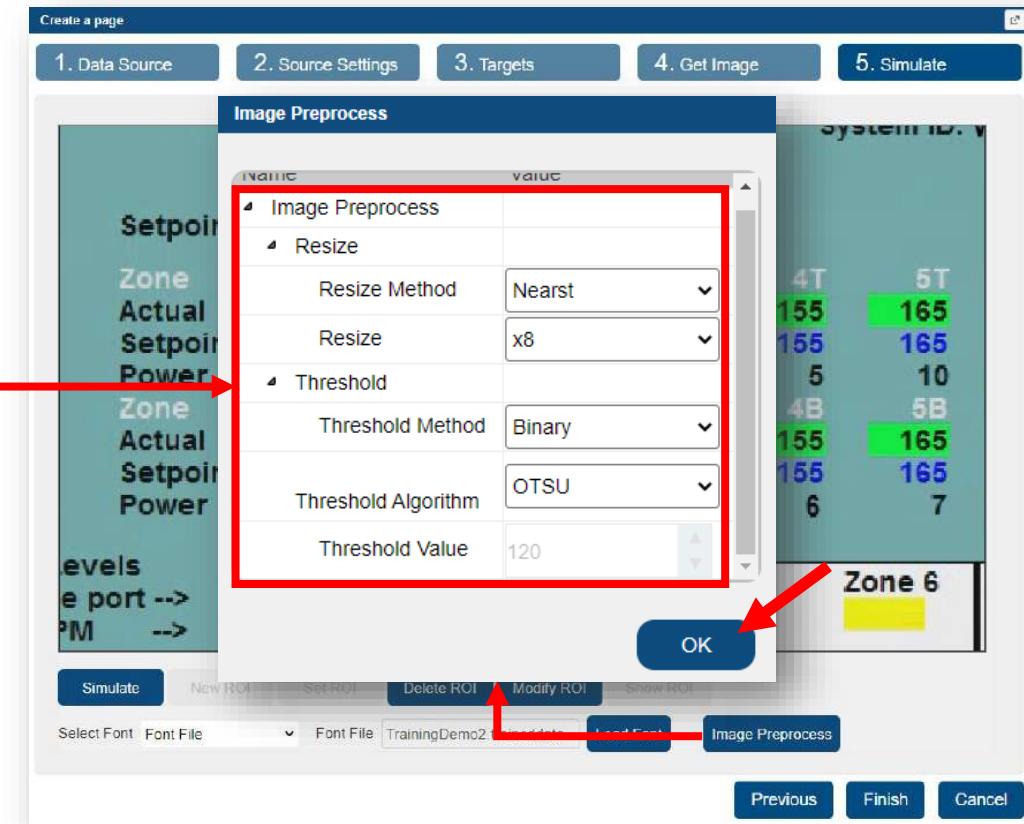
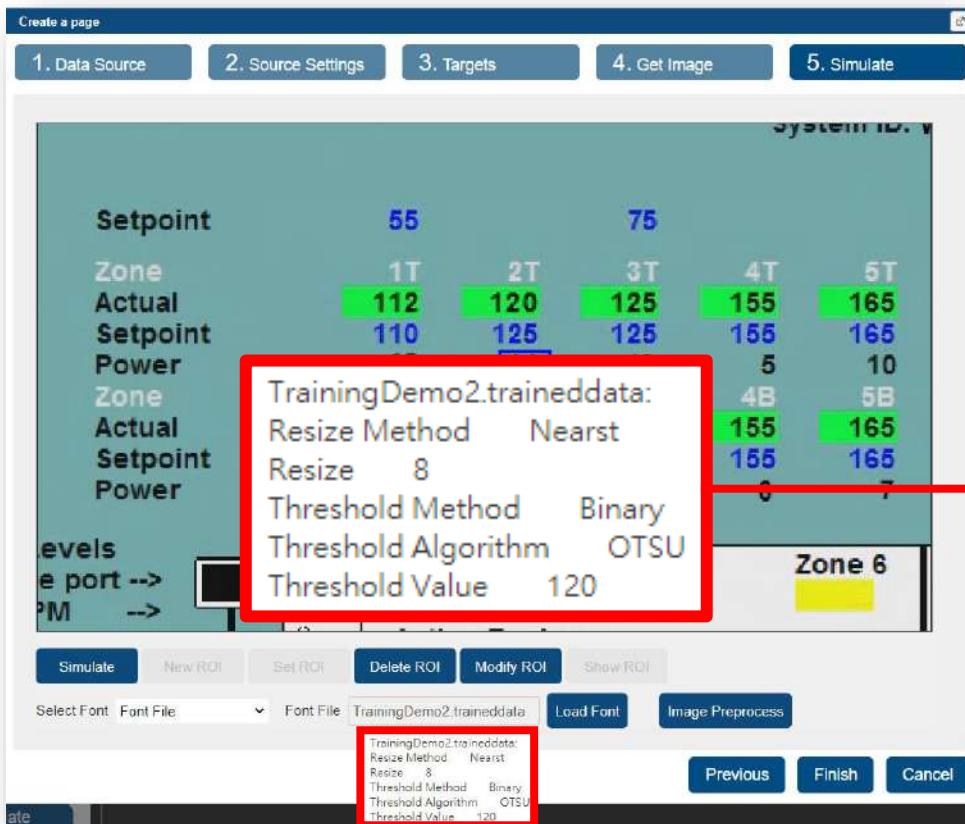


## • 精靈：新增頁面

### 5. 模擬

影像前處理 → 設定影像前處理參數，要和字型專案內的前處理參數相同 → 點擊確定

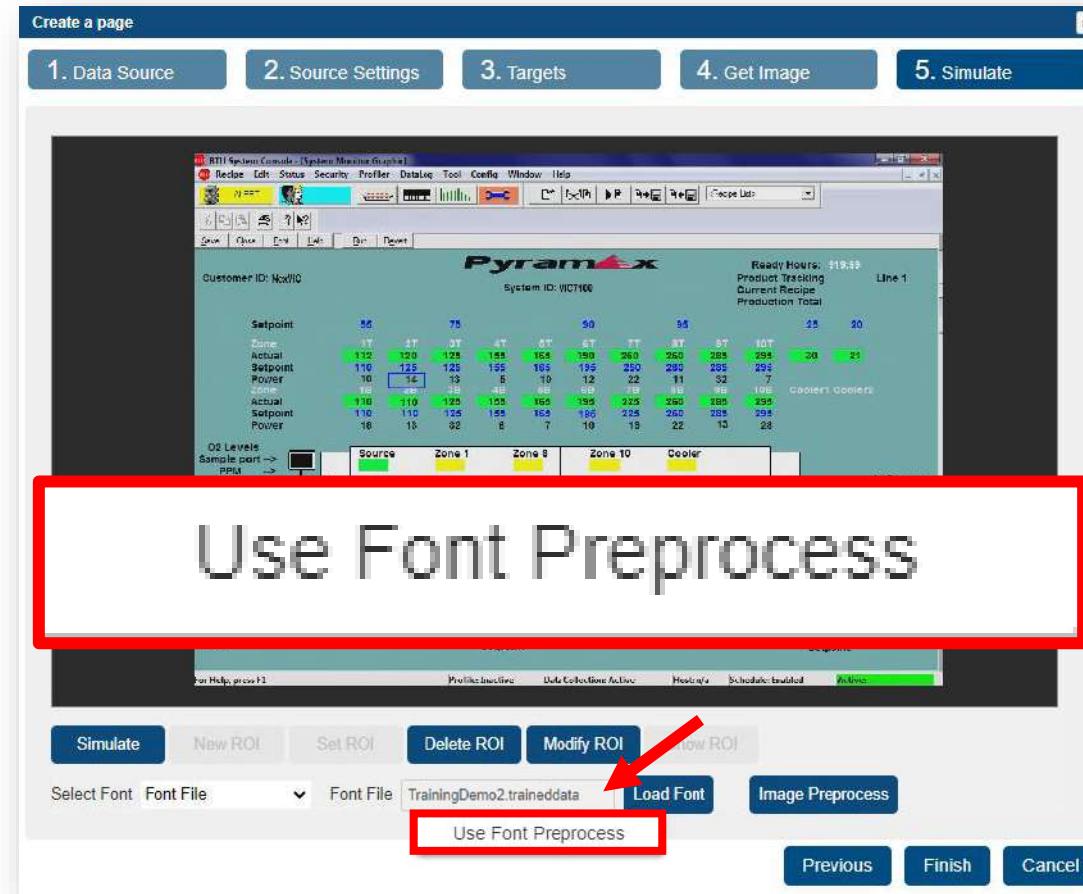
將游標移到字型檔上，會顯示該字型檔使用的前處理參數



- 精靈：新增頁面

## 5. 模擬

在字型檔欄位點擊右鍵，選擇使用字型的前處理值，可以快速選用前處理參數



- 精靈：新增頁面

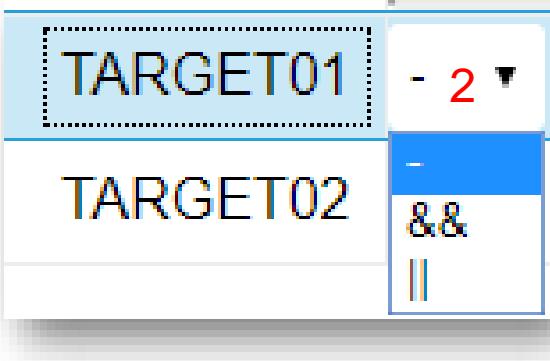
## 5. 模擬

點擊模擬 → 確認辨識值



## • 一般

1. 新增頁面：點擊新增頁面
2. 目標值：設定目標值比對條件



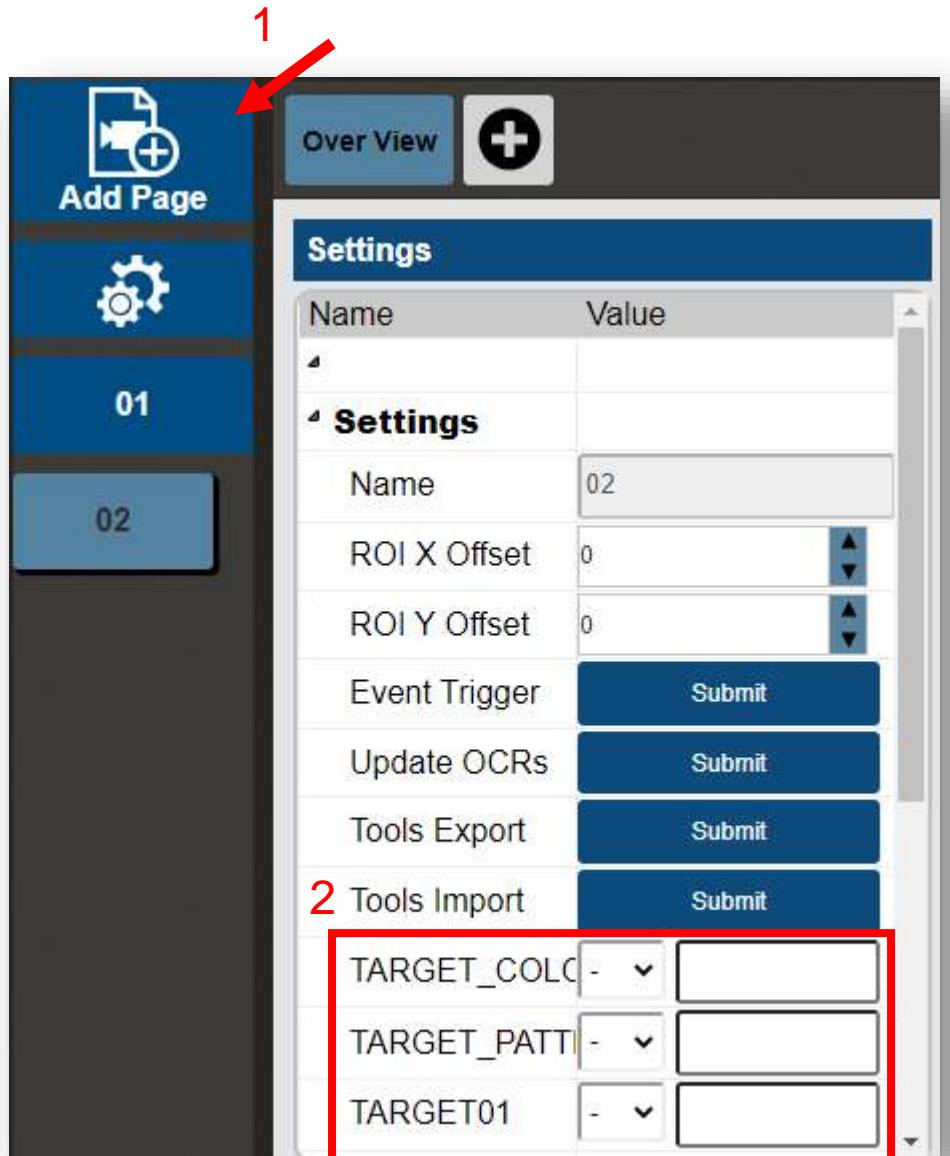
- : 不判定  
 && : AND  
 || : OR

比對邏輯運算子：

若為 -，不比對此目標值

先比對 ||，若有一設定 || 的目標值符合，則此頁面比對成功，  
 使用此頁面

||比對失敗，再比對 &&，需全部設定 && 的目標值符合，  
 此頁面才比對成功，才使用此頁面  
 若有多個頁面同時比對成功，會使用編號較小的頁面



## • 一般

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Stop	&&	Operate	
PAGE02		Text	&&	Run	&&	Recipe	使用此頁面

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Stop	&&	Operate	
PAGE02		Image	&&	Run	&&	Operate	使用此頁面

## • 一般

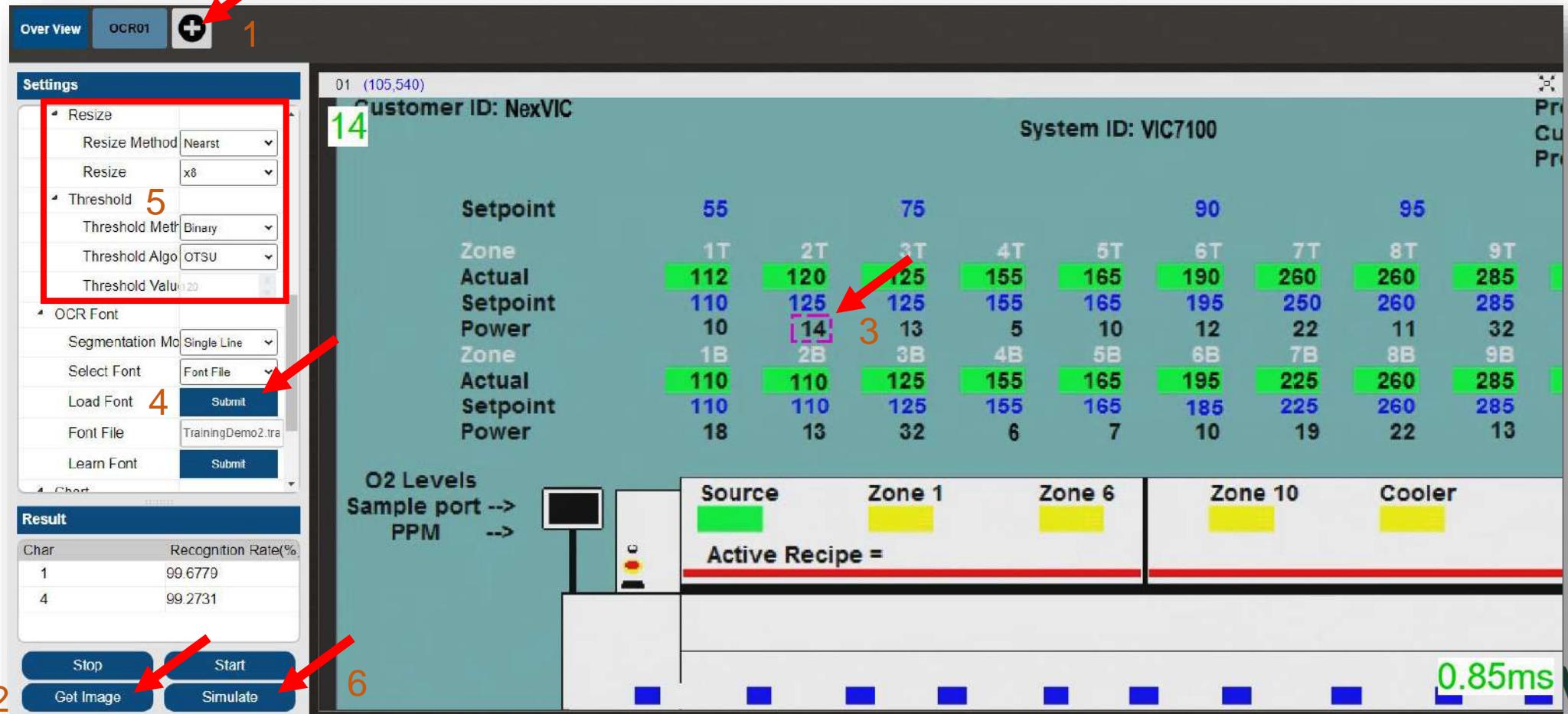
	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Run	&&	Operate	使用此頁面
PAGE02		Text	&&	Stop	&&	Recipe	

	TARGET01		TARGET02		TARGET03		使用的頁面
實際辨識結果	Text		Run		Operate		
PAGE01		Image	&&	Run	&&	Operate	使用此頁面
PAGE02		Image	&&	Run	&&	Operate	

## • 一般

4. 新增 OCR : 點擊新增按鈕 → 取得圖像 → 設置ROI → 選擇字型檔 → 設定前處理參數 → 模擬辨識

此OCR內的資料就是要擷取出的資料



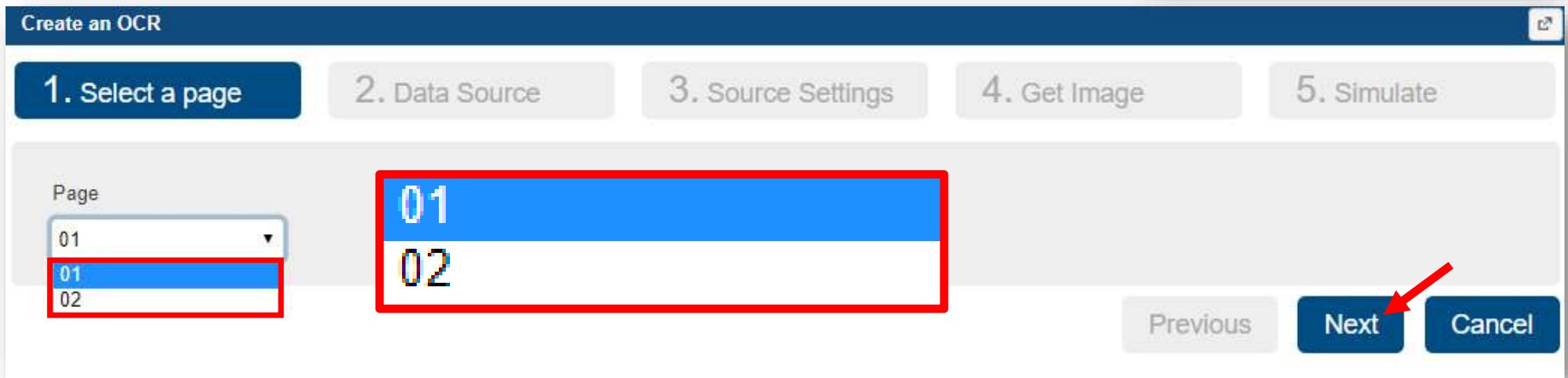
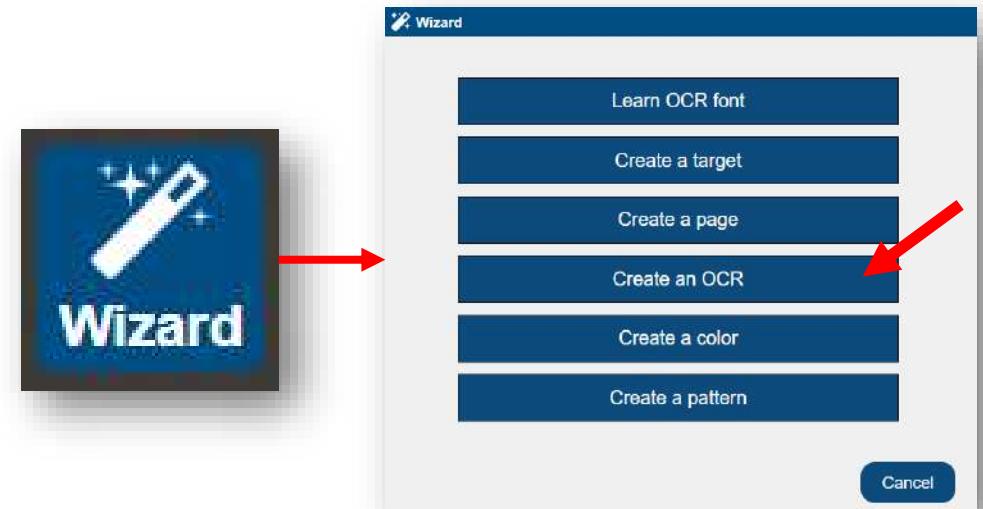
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

- 精靈：新增 OCR

1. 選擇頁面

選擇要新增 OCR 的頁面

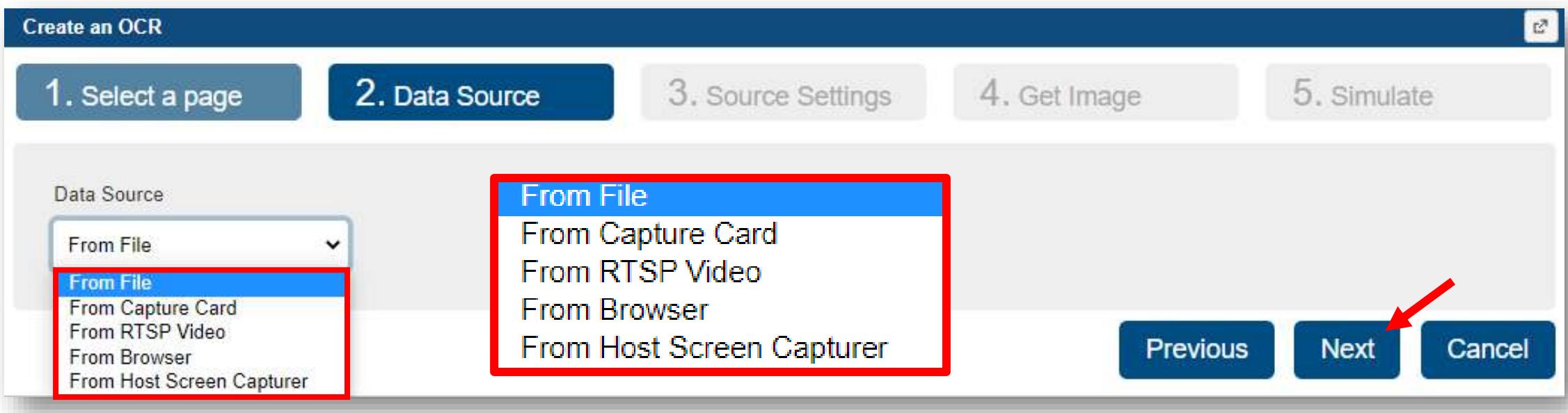


- 精靈：新增 OCR

## 2. 影像來源

來自檔案、來自擷取卡、來自 RTSP 影像

來自瀏覽器、來自本機畫面擷取

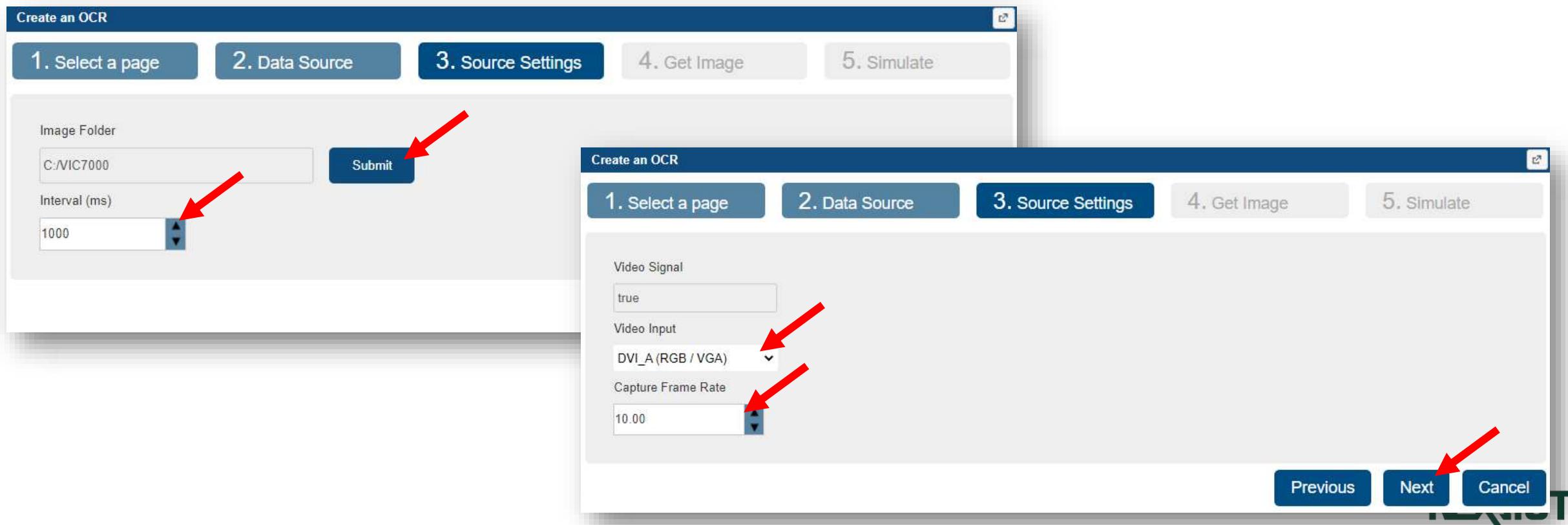


- 精靈：新增 OCR

## 3. 影像來源設定

來自檔案：圖檔路徑、讀取時間間隔

來自擷取卡：輸入訊號類型、擷取頻率(fps)



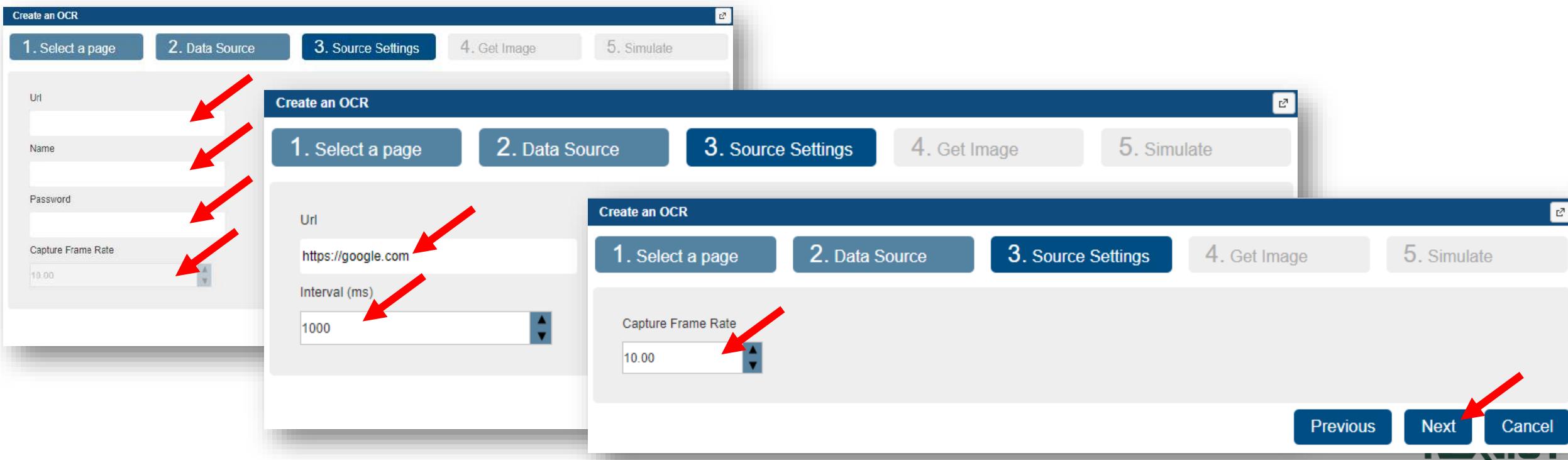
- 精靈：新增 OCR

## 3. 影像來源設定

來自 RTSP 影像：RTSP 來源網址、名稱、密碼、擷取頻率

來自瀏覽器：網址、讀取時間間隔

來自本機畫面擷取：擷取頻率(fps)



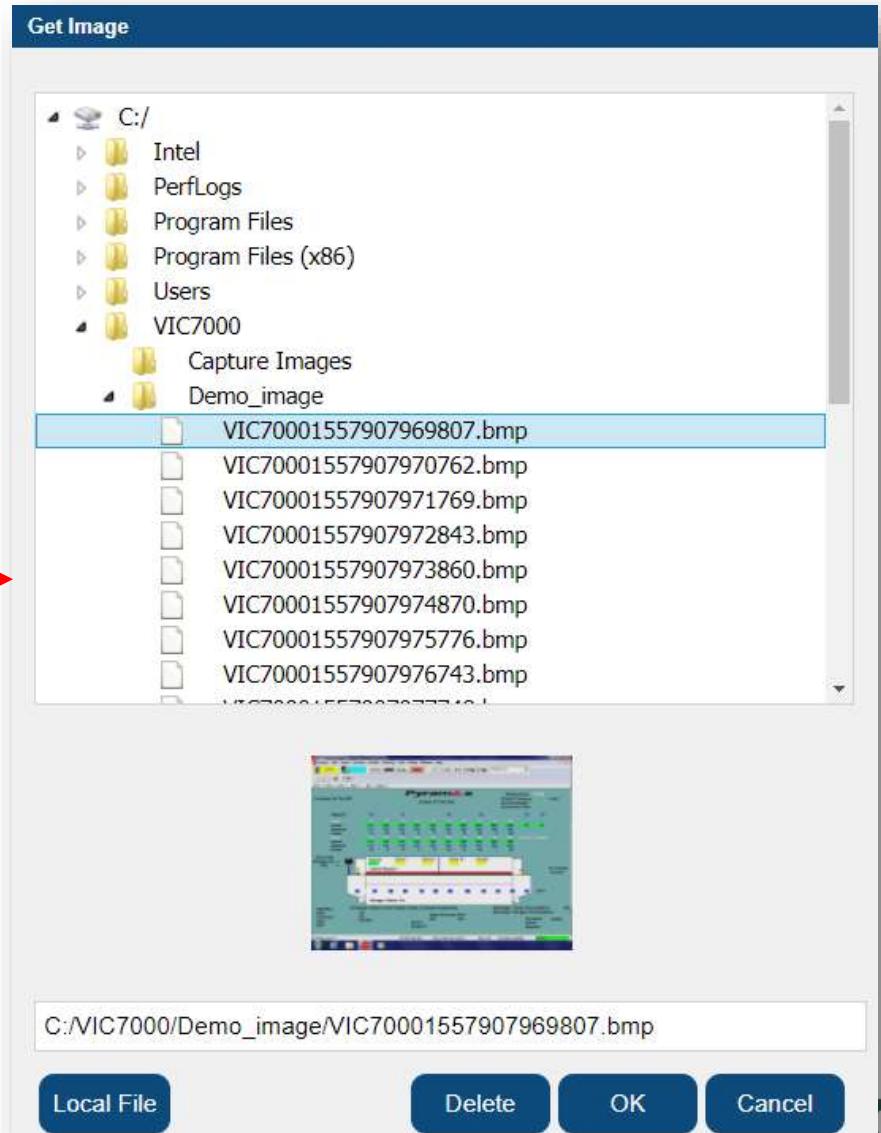
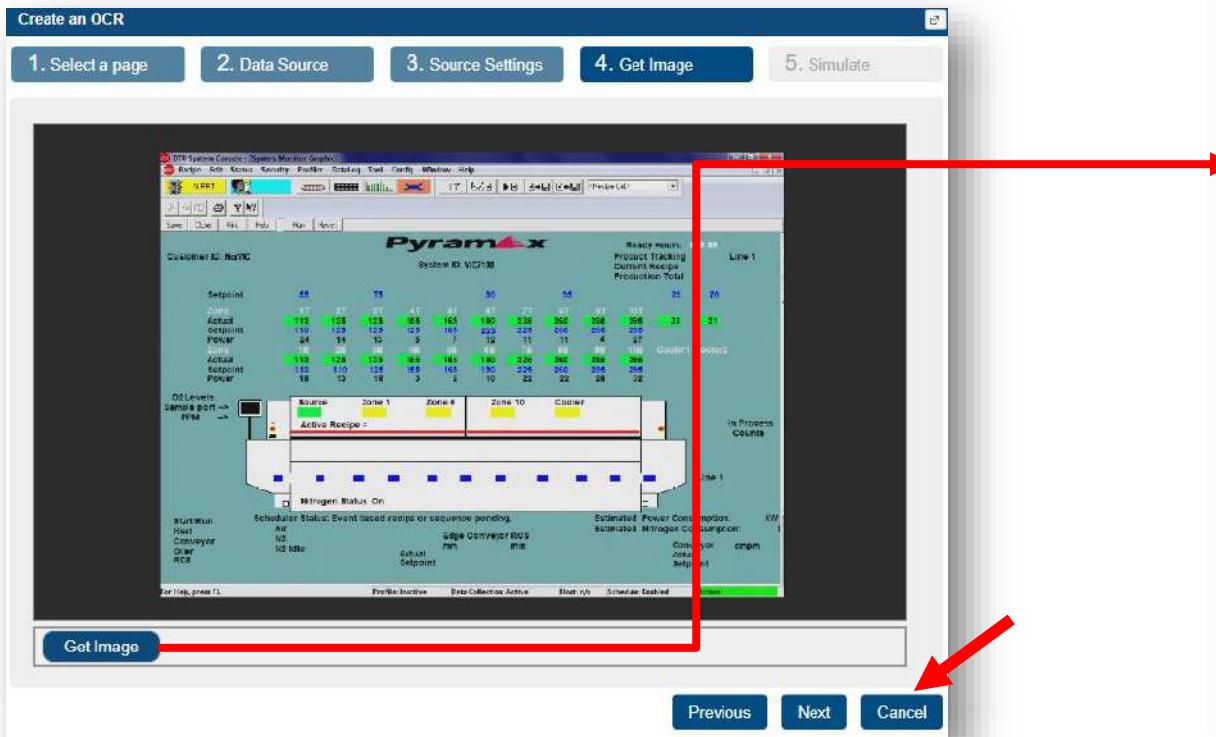
- 精靈：新增 OCR

## 4. 取得圖像

來自檔案：讀取本機或遠端的圖像

來自擷取卡、來自 RTSP 影像、來自瀏覽器、

來自本機畫面擷取：從取像來源取得一張圖像



- 精靈：新增 OCR

## 5. 模擬

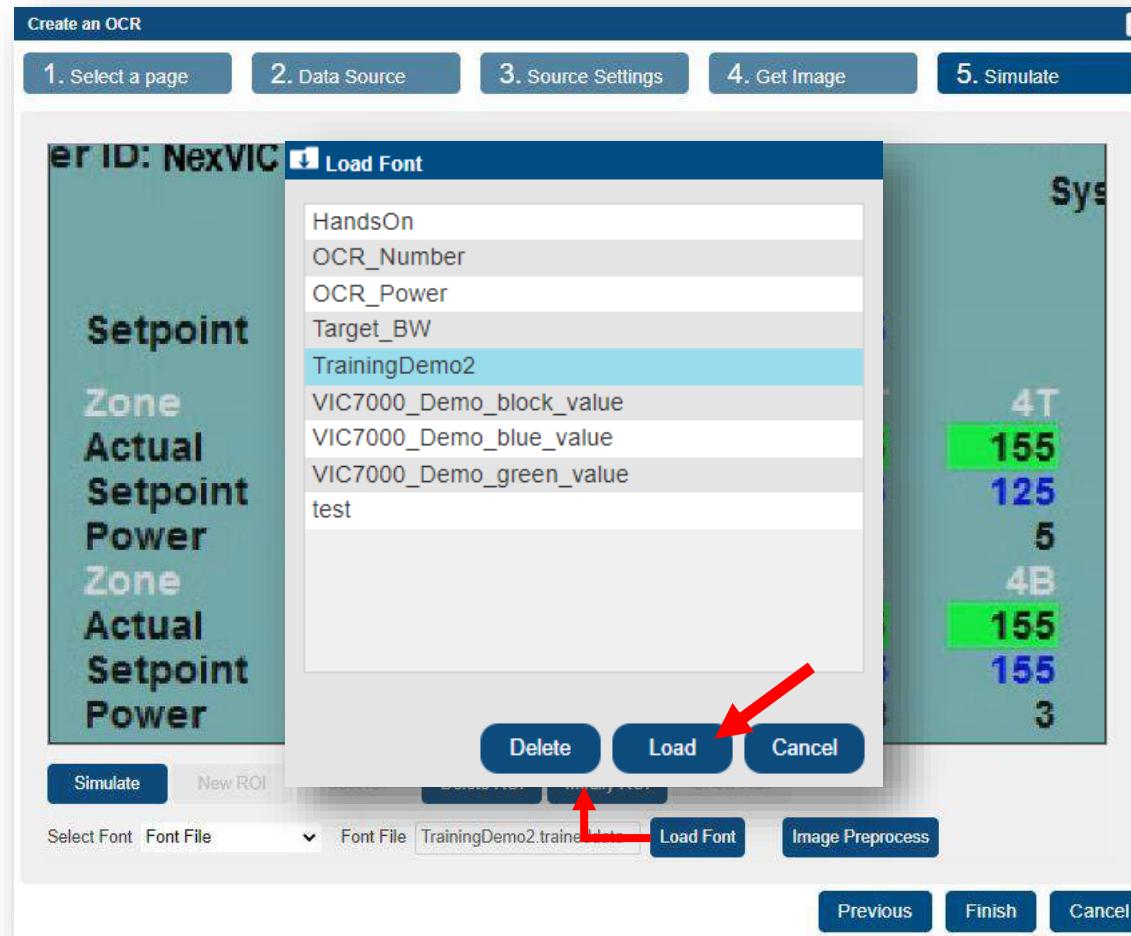
右鍵新增ROI or 點擊 新增ROI → 調整大小和位置 → 右鍵設置ROI or 點擊設置ROI



- 精靈：新增 OCR

## 5. 模擬

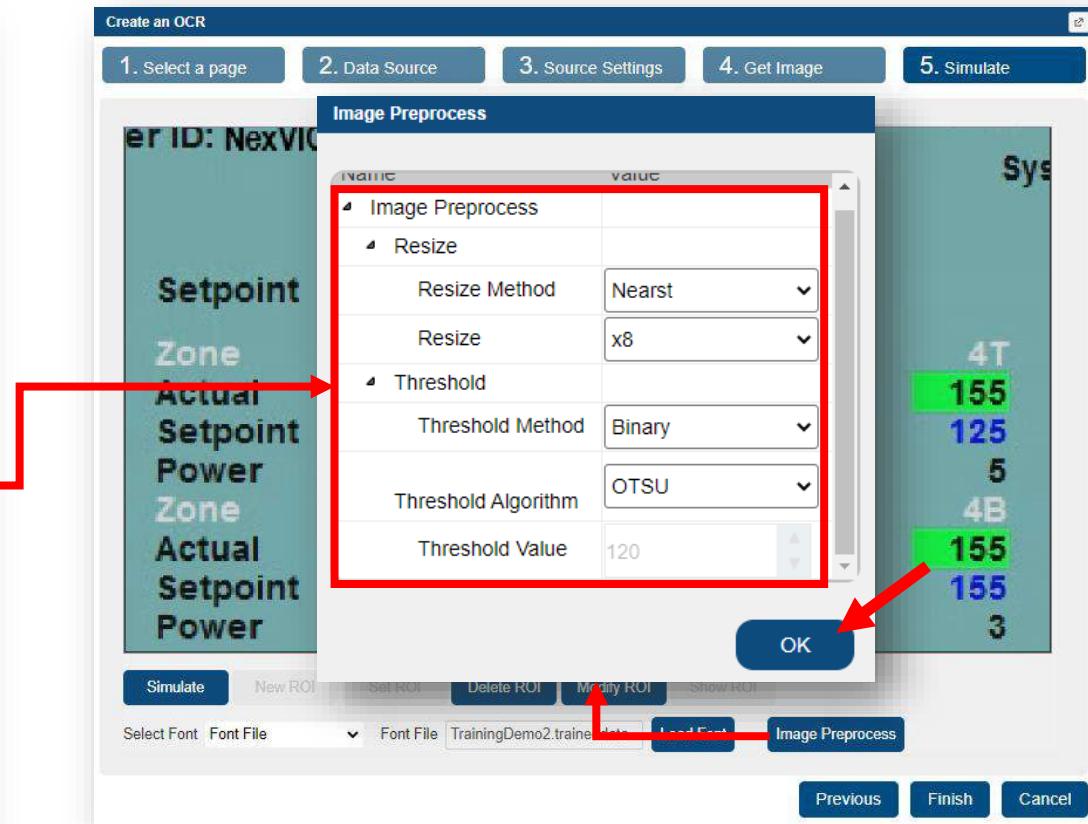
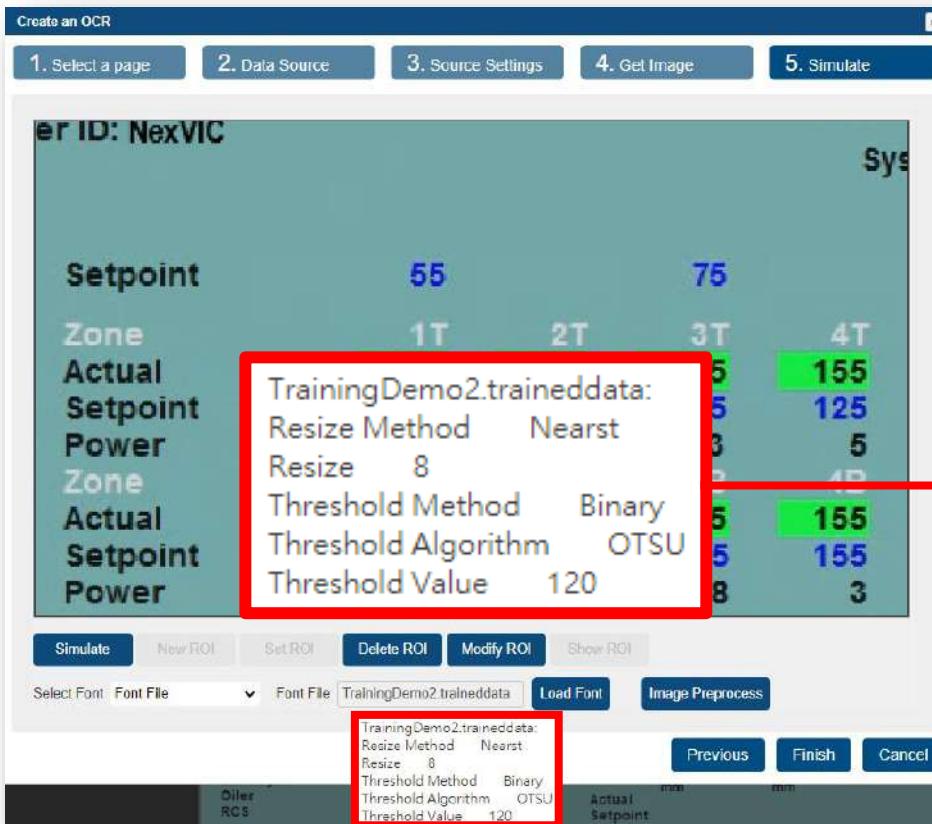
載入字型 → 選擇要使用的字型檔 → 點擊載入



- 精靈：新增 OCR

## 5. 模擬

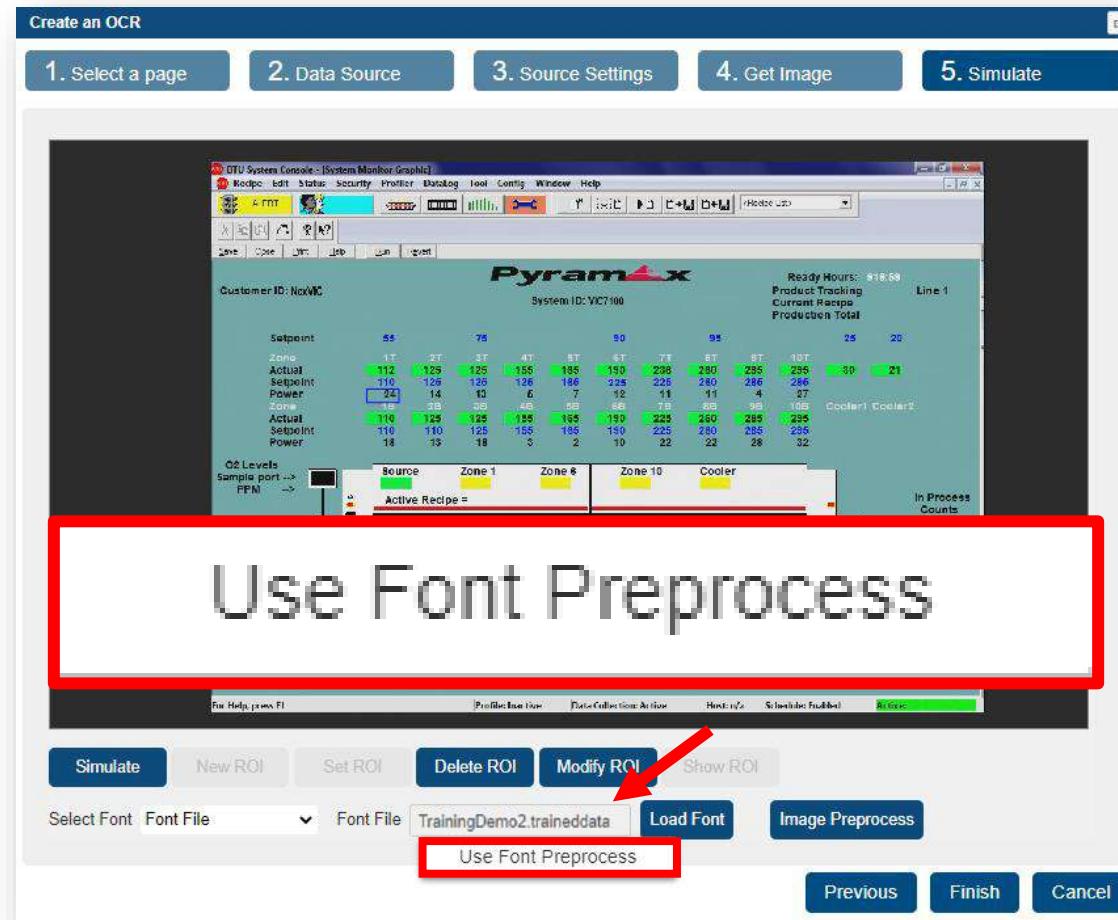
影像前處理 → 設定影像前處理參數，要和字型專案內的前處理參數相同 → 點擊確定  
將游標移到字型檔上，會顯示該字型檔使用的前處理參數



- 精靈：新增 OCR

## 5. 模擬

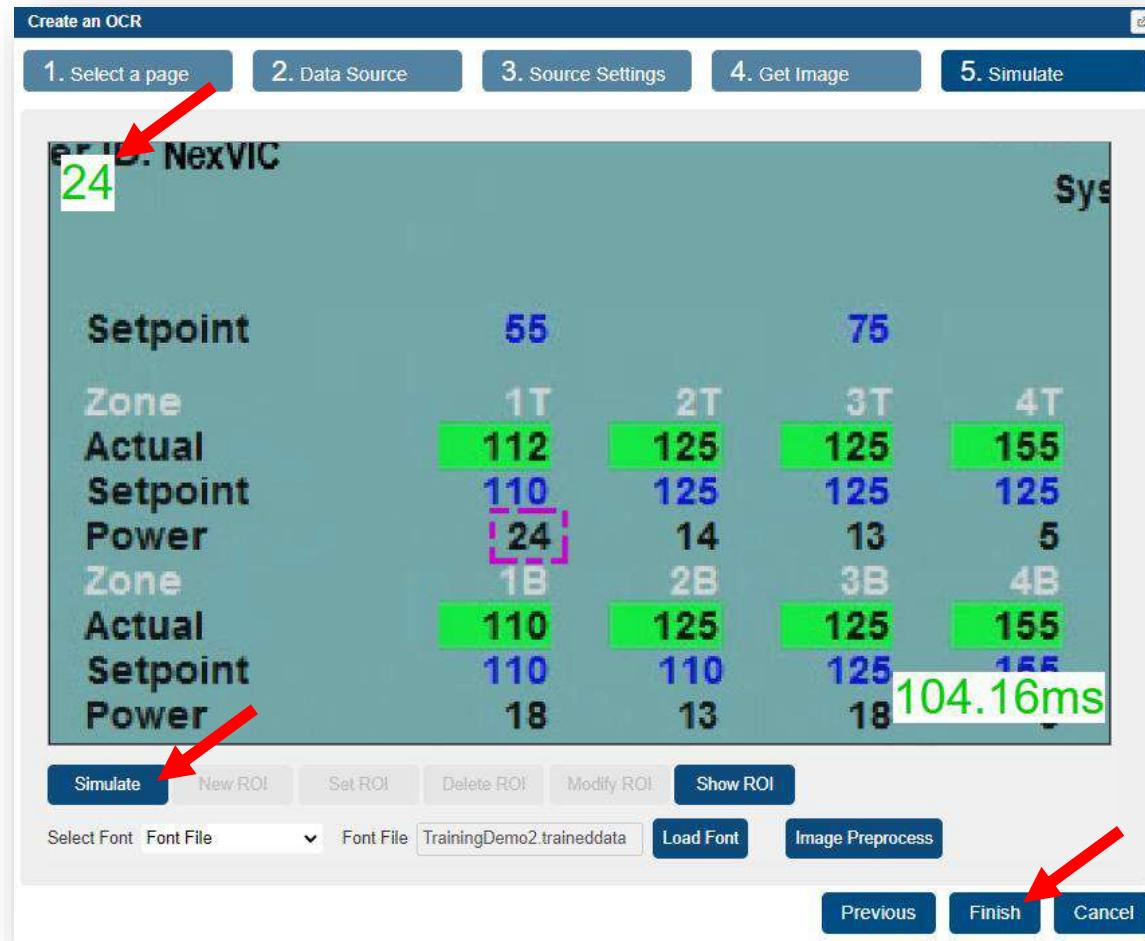
在字型檔欄位點擊右鍵，選擇使用字型的前處理值，可以快速選用前處理參數



- 精靈：新增 OCR

## 5. 模擬

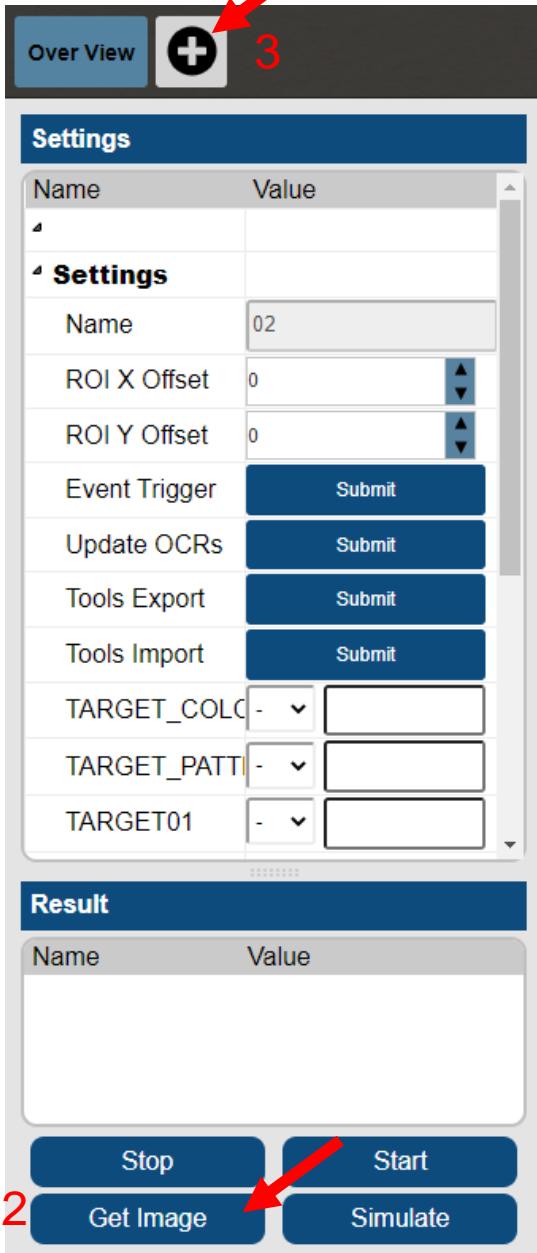
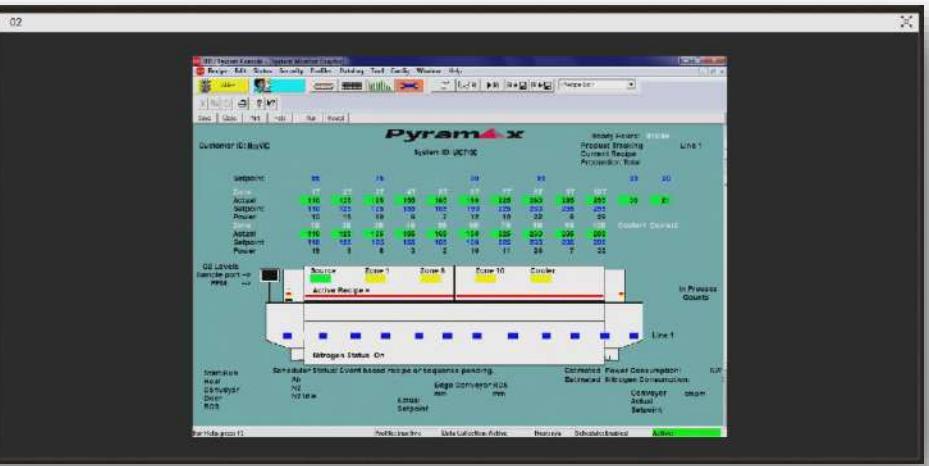
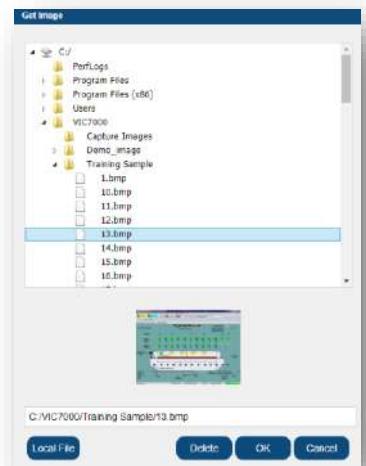
點擊模擬 → 確認辨識值



# 辨識工具

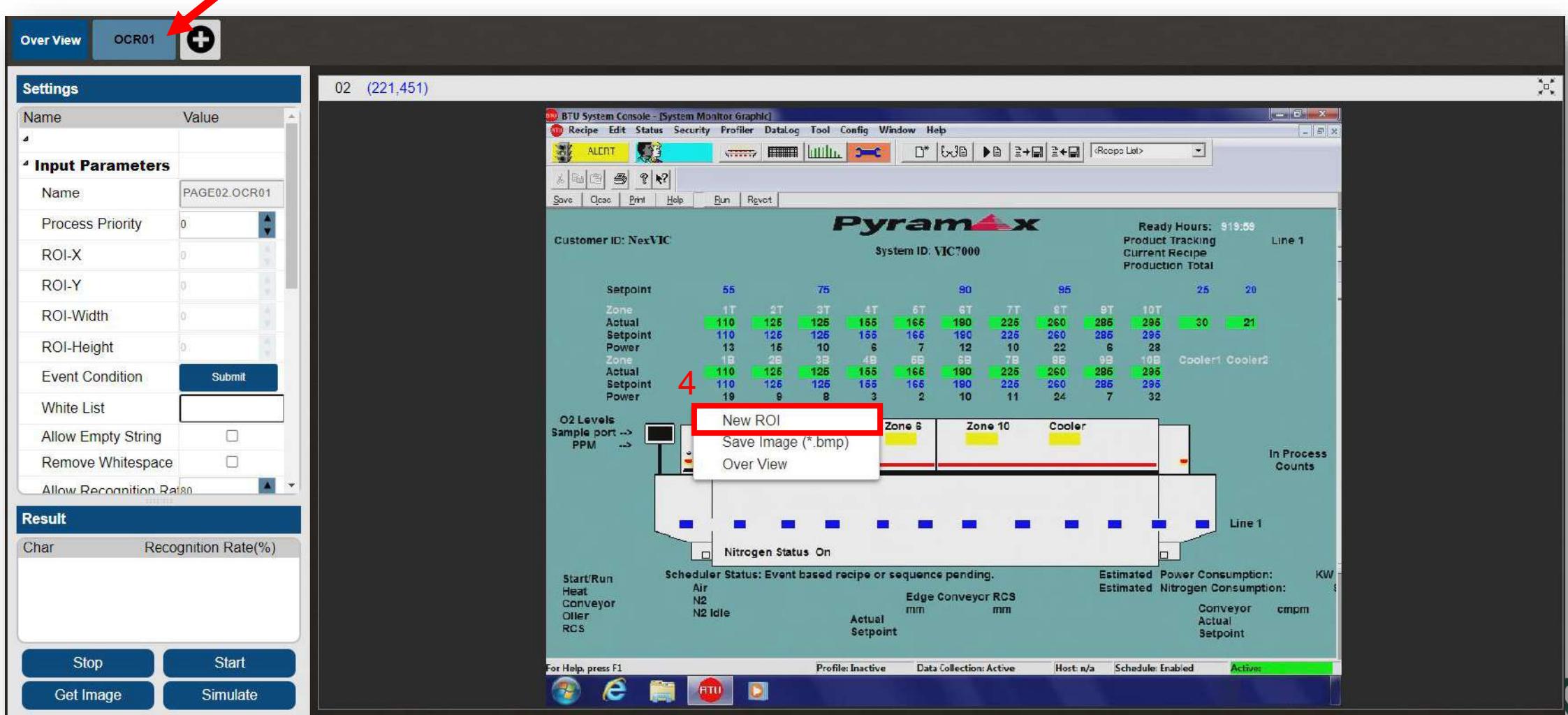
## • 一般：新增 OCR

1. 選擇頁面：進入要新增 OCR 的頁面
2. 取得圖像
  - a. 來自檔案：讀取本地或遠端的圖像
  - b. 來自擷取卡、RSTP 影像、瀏覽器  
、本機畫面擷取：從取像來源取得一張圖像
3. 新增 OCR：點擊新增按鈕



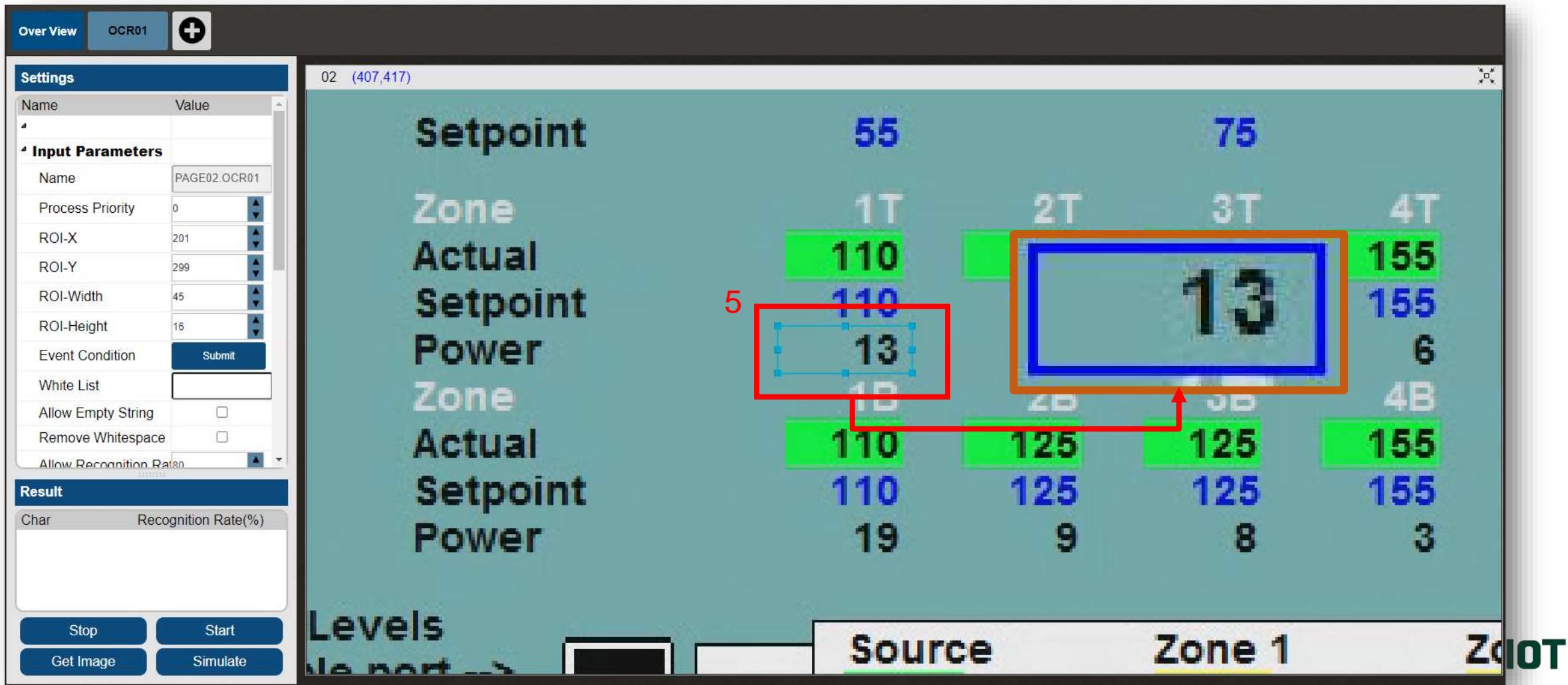
- 一般：新增 OCR

- 4. 新增ROI：在圖像上點擊右鍵 → 選擇新增ROI



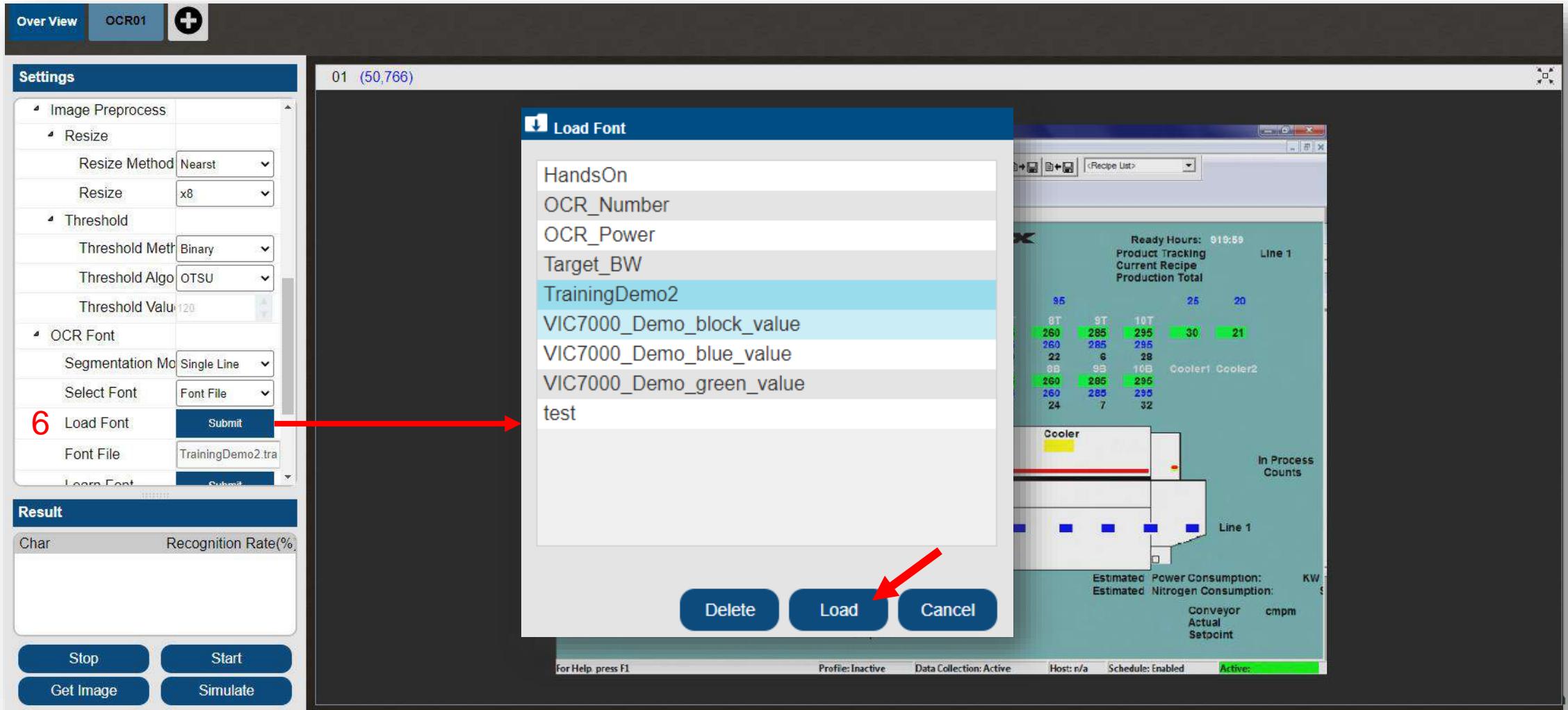
- 一般：新增 OCR

5. 設置ROI：調整ROI大小和位置後 → 在圖像任意處點擊左鍵



- 一般：新增 OCR

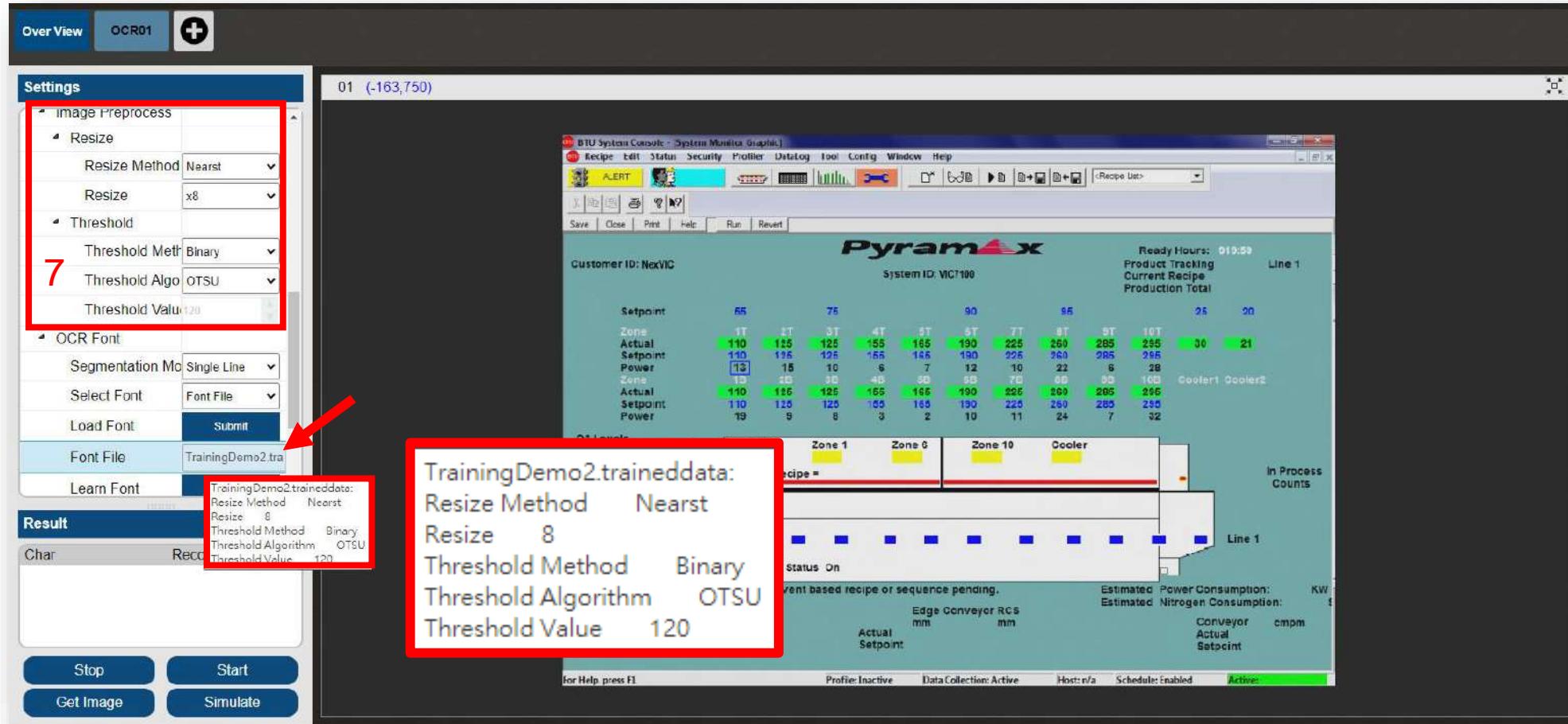
- 選擇字型檔：載入字型 → 選擇要使用的字型檔 → 點擊載入



- 一般：新增 OCR

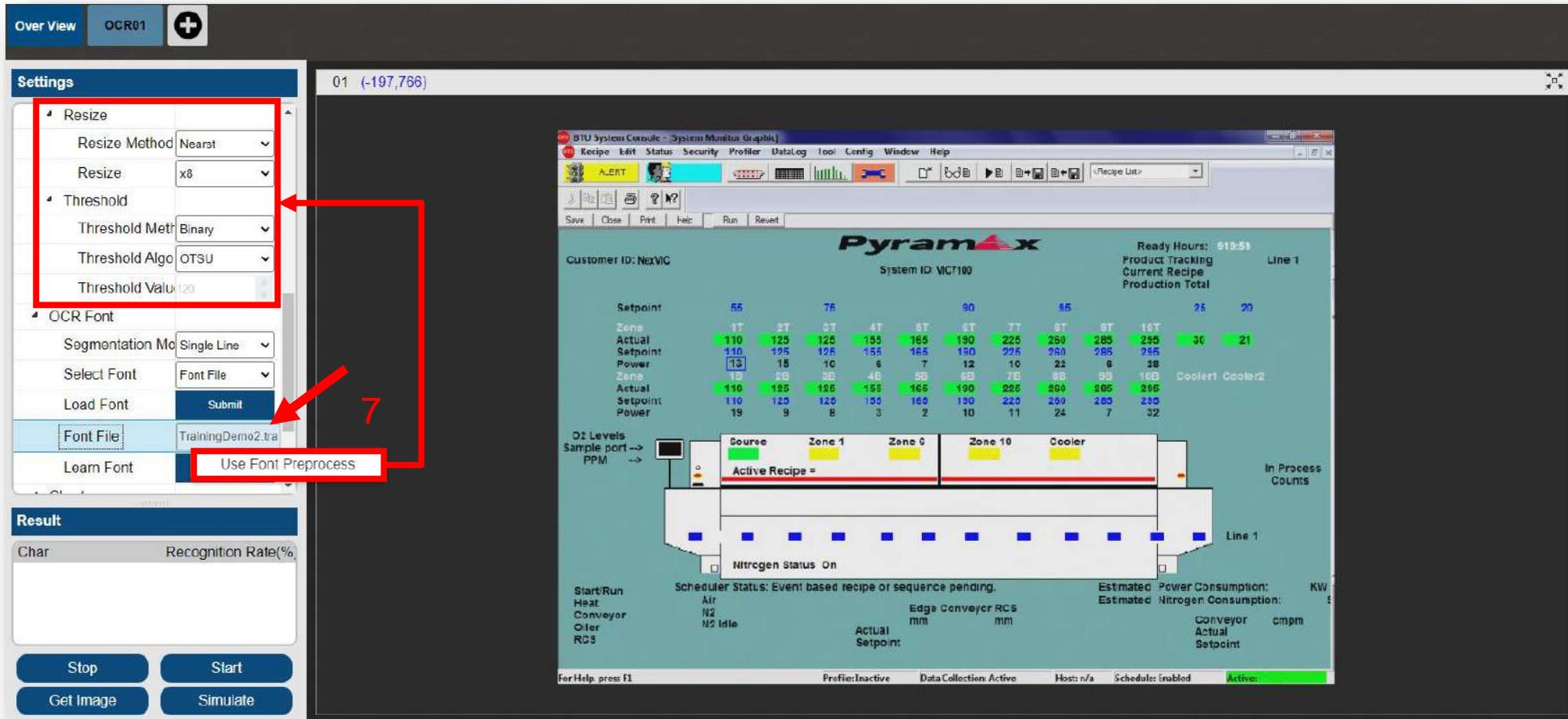
7. 設定前處理參數：設定影像前處理參數，要和字型專案內的前處理參數相同

將游標移到字型檔上，會顯示該字型檔使用的前處理參數



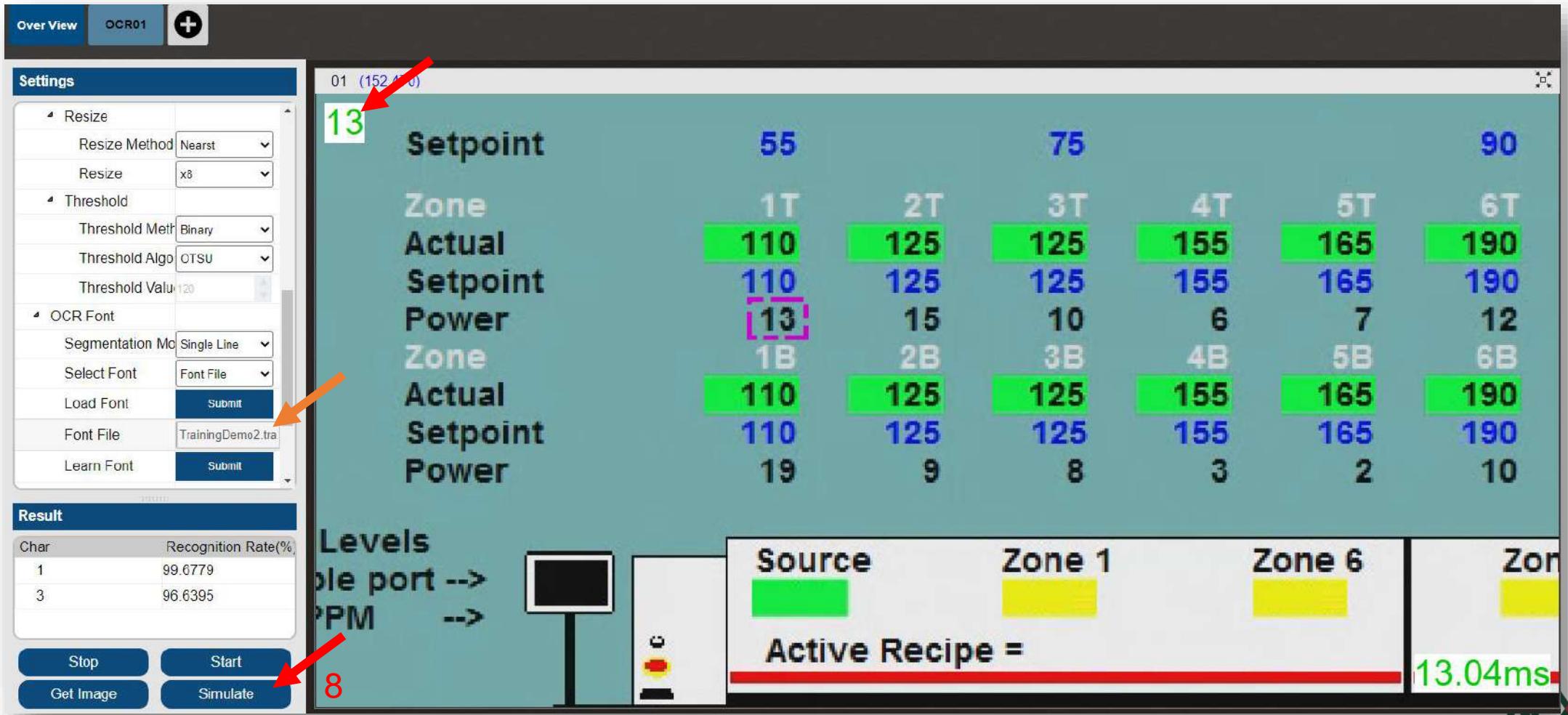
- 一般：新增 OCR

7. 設定前處理參數：在字型檔欄位點擊右鍵，選擇使用字型的前處理值，可以快速選用前處理參數



- 一般：新增 OCR

8. 模擬確認：點擊模擬 → 確認辨識正確 此 OCR 內的資料就是要擷取的資料



- **OCR 設定**

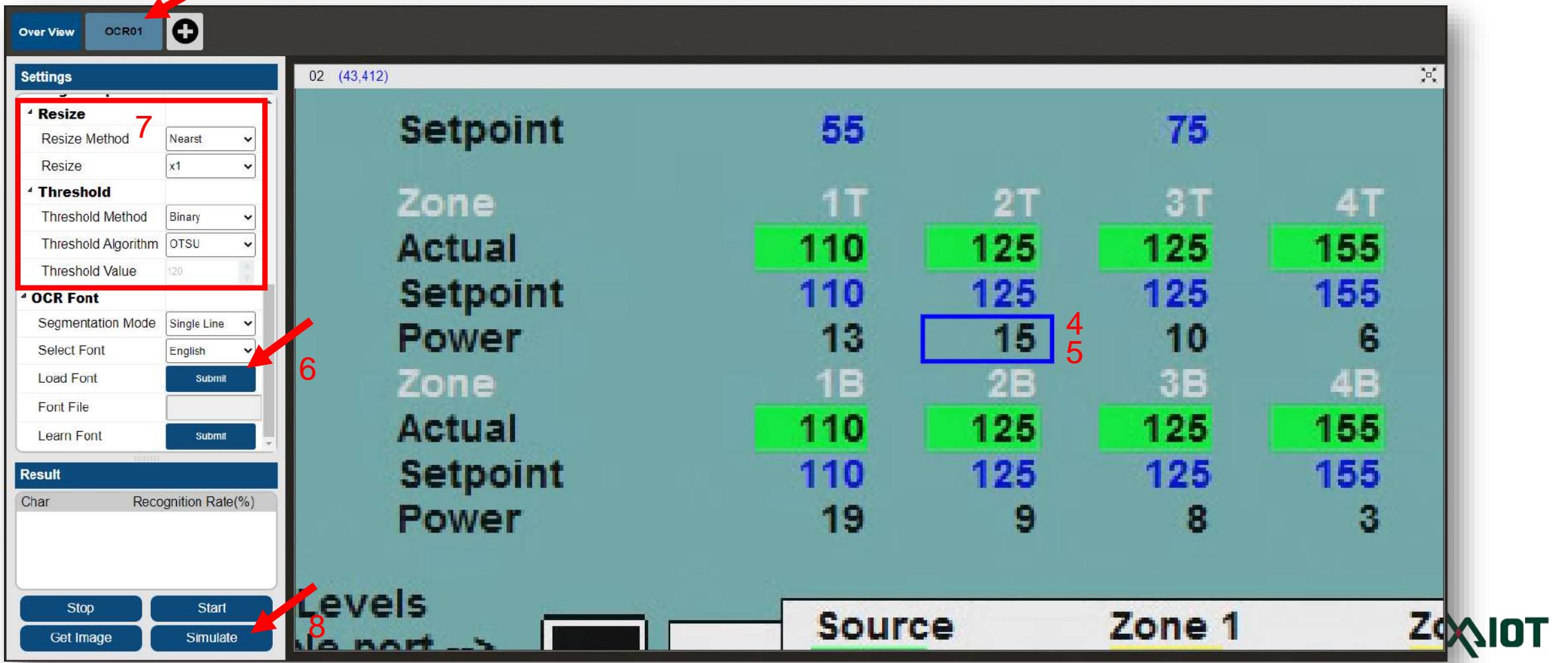
- **相關參數**

- **ROI - X** : 設置 ROI 的 X 座標
    - **ROI - Y** : 設置 ROI 的 Y 座標
    - **ROI - 寬度** : 設置 ROI 的寬度
    - **ROI - 高度** : 設置 ROI 的高度
    - **白名單** : 若該有欄位有字元，系統僅會辨識出欄位內的字元
    - **允許空字串** : 若啟用，辨識到空字串時，不會顯示 NG
    - **移除空白字元** : 移除辨識結果的空白字元
    - **允許辨識率** : 若辨識率低於此數值，就會顯示 NG
    - **頁面位移偵測無效** : 若勾選，則頁面位移偵測功能會略過此辨識工具

- 一般：新增OCR

- **多個 OCR**：在同一頻道和頁面上點擊新增按鈕 → 依照前述步驟可新增更多 OCR

3



- 精靈：新增顏色

1. 選擇頻道

選擇要新增顏色的頁面

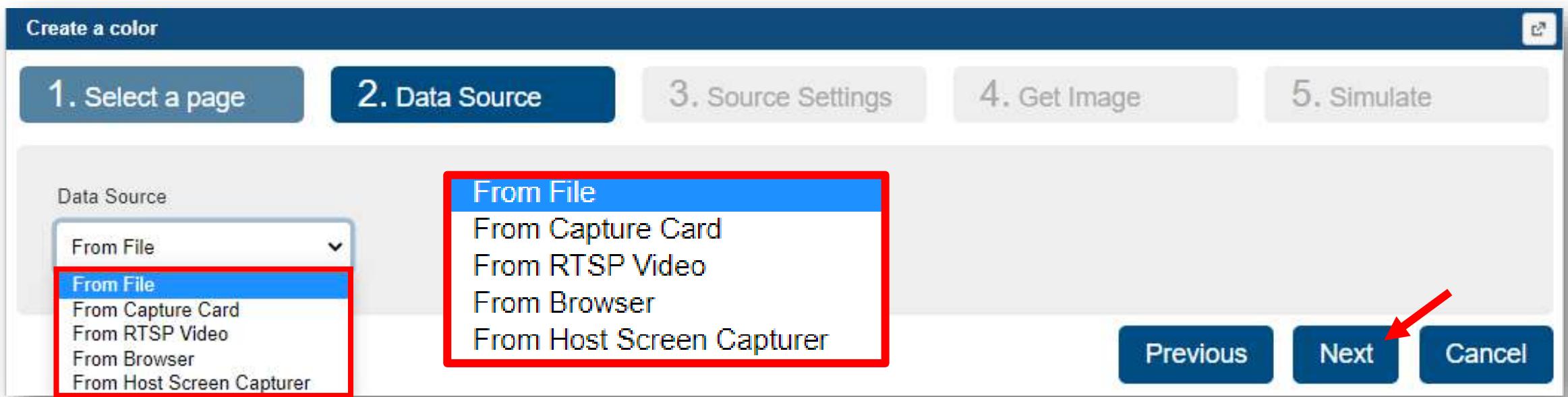


- 精靈：新增顏色

## 2. 影像來源

來自檔案、來自擷取卡、來自 RTSP 影像

來自瀏覽器、來自本機畫面擷取

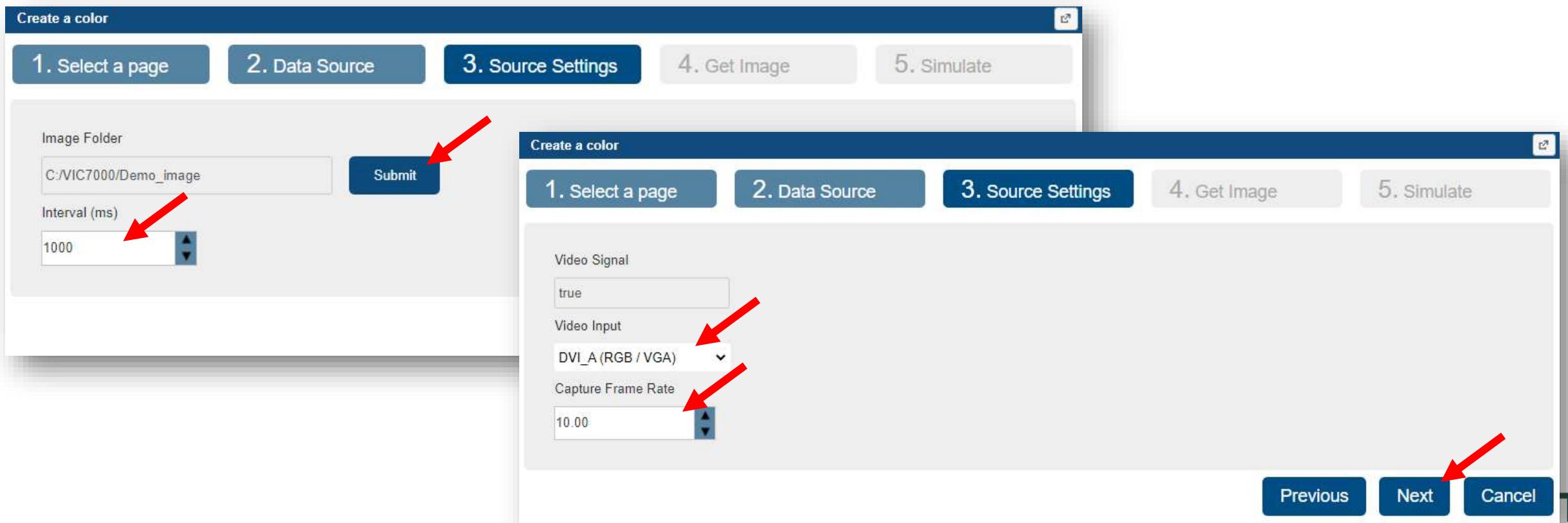


- 精靈：新增顏色

## 3. 影像來源設定

來自檔案：圖檔路徑、讀取時間間隔

來自擷取卡：輸入訊號類型、擷取頻率(fps)



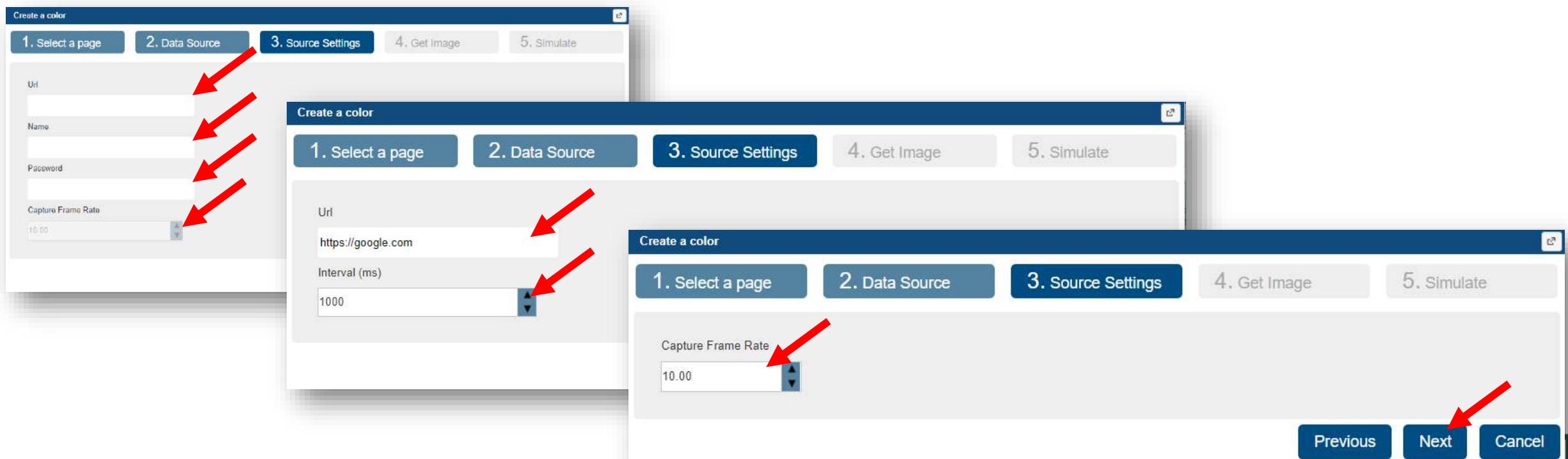
- 精靈：新增顏色

## 3. 影像來源設定

來自 RTSP 影像：RTSP 來源網址、名稱、密碼、擷取頻率

來自瀏覽器：網址、讀取時間間隔

來自本機畫面擷取：擷取頻率(fps)



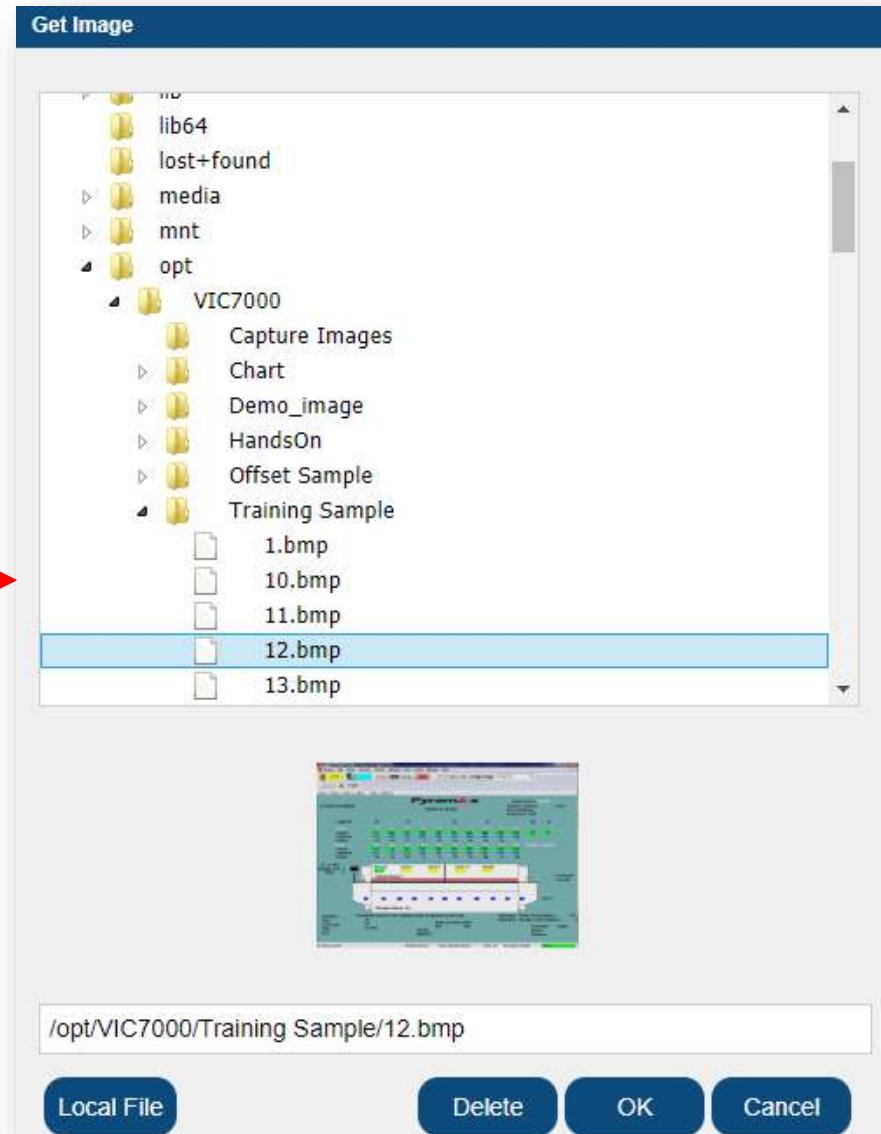
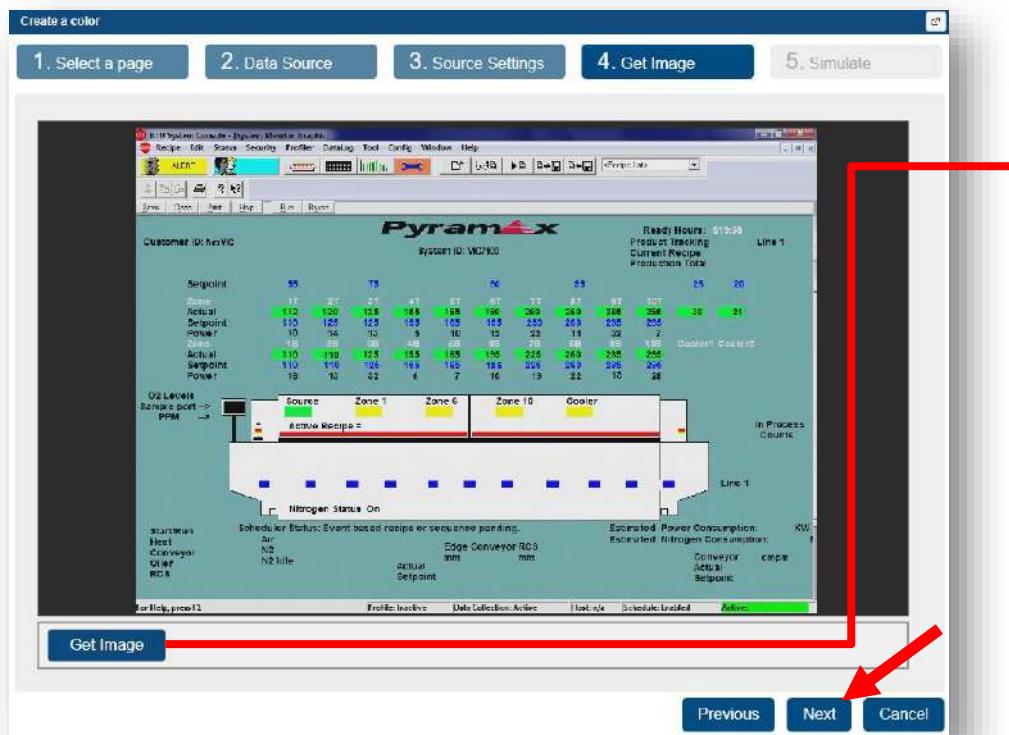
- 精靈：新增顏色

## 4. 取得圖像

來自檔案：讀取本機或遠端的圖像

來自擷取卡、來自 RTSP 影像、來自瀏覽器、

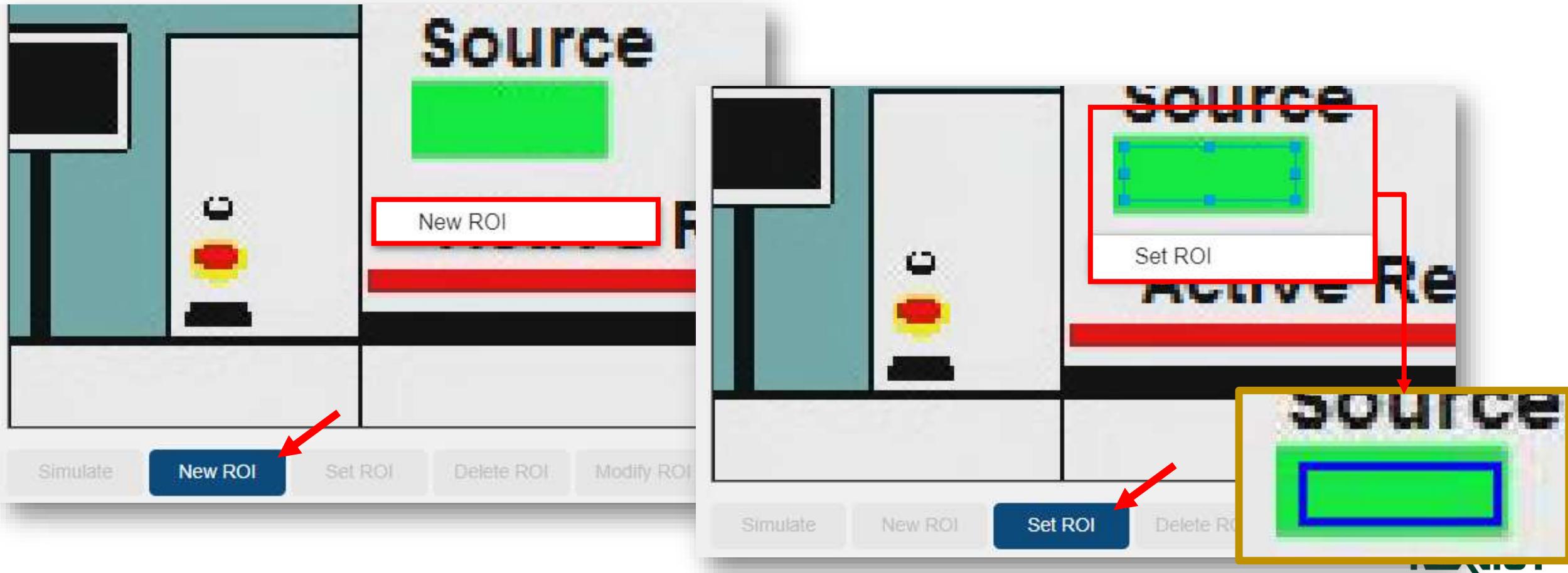
來自本機畫面擷取：從取像來源取得一張圖像



- 精靈：新增顏色

## 5. 模擬

右鍵新增ROI or 點擊 新增ROI → 調整大小和位置 → 右鍵設置ROI or 點擊設置ROI



- 精靈：新增顏色

## 5. 模擬

設置ROI時會自動取得ROI內的RGB平均值，並將其設為顏色比對條件

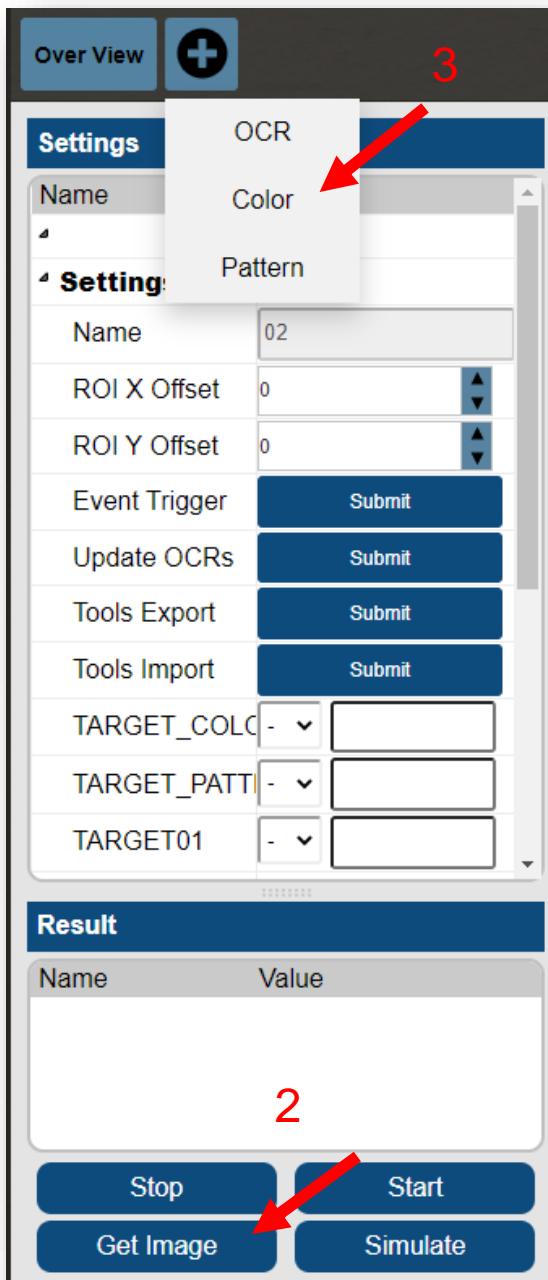
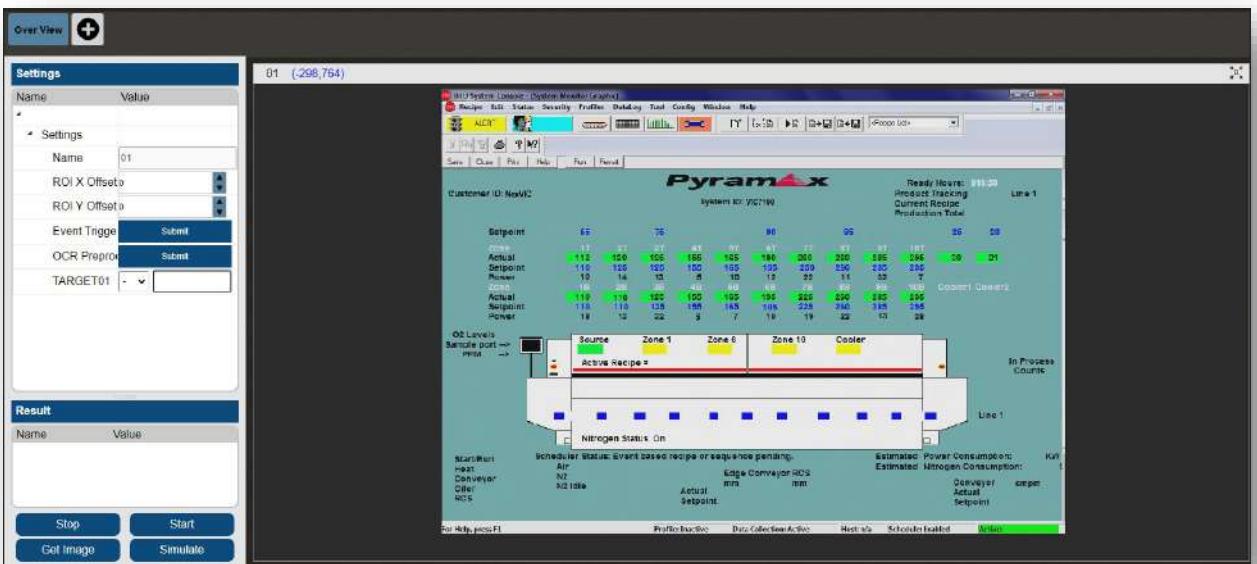
點擊模擬 → 確認辨識值



# 辨識工具

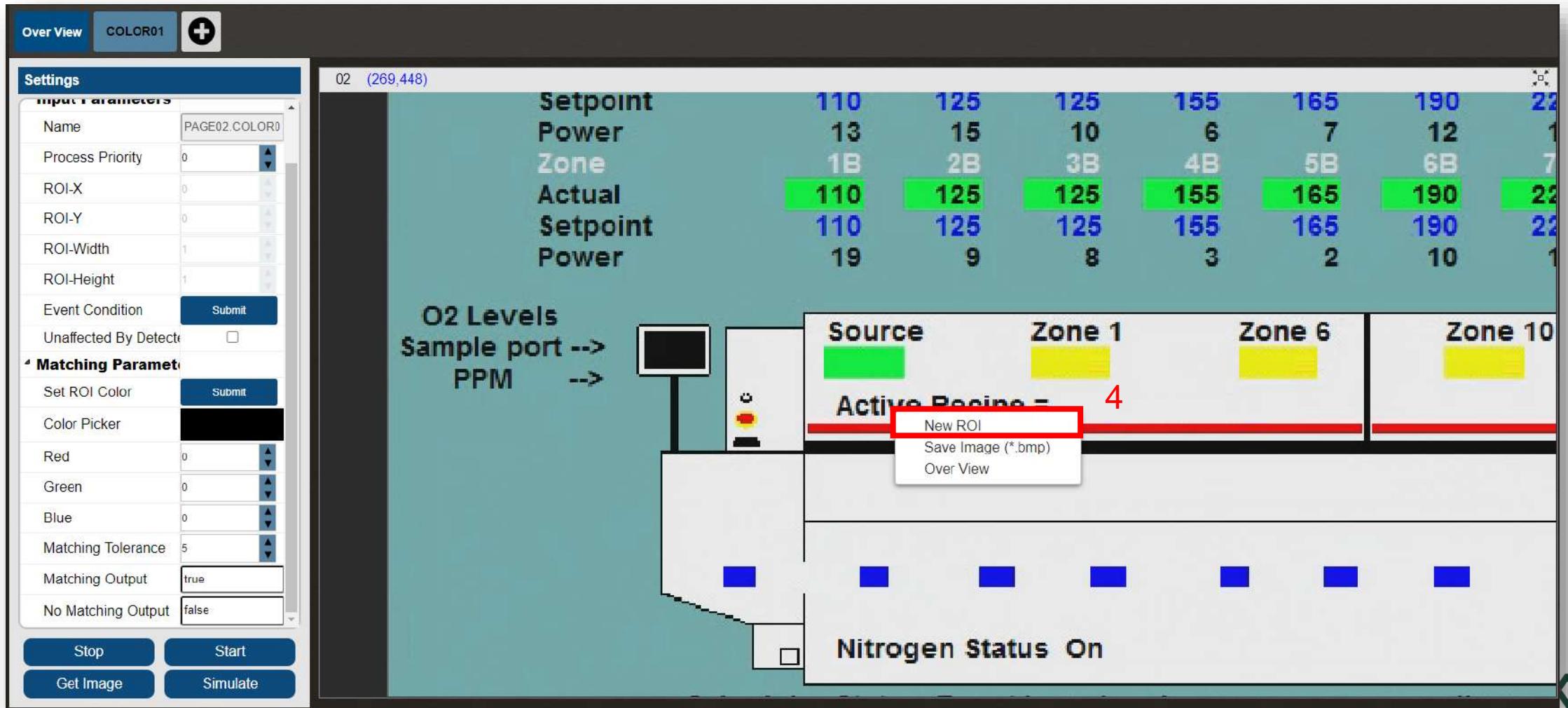
## • 一般：新增顏色

1. 選擇頁面：選擇要新增顏色的頁面
2. 取得圖像
  - a. 來自檔案：讀取本地或遠端的圖像
  - b. 來自擷取卡、RTSP 影像、瀏覽器、  
本機畫面擷取：從來源取得一張圖像
3. 新增顏色：滑鼠移到新增按鈕 → 點擊 Color



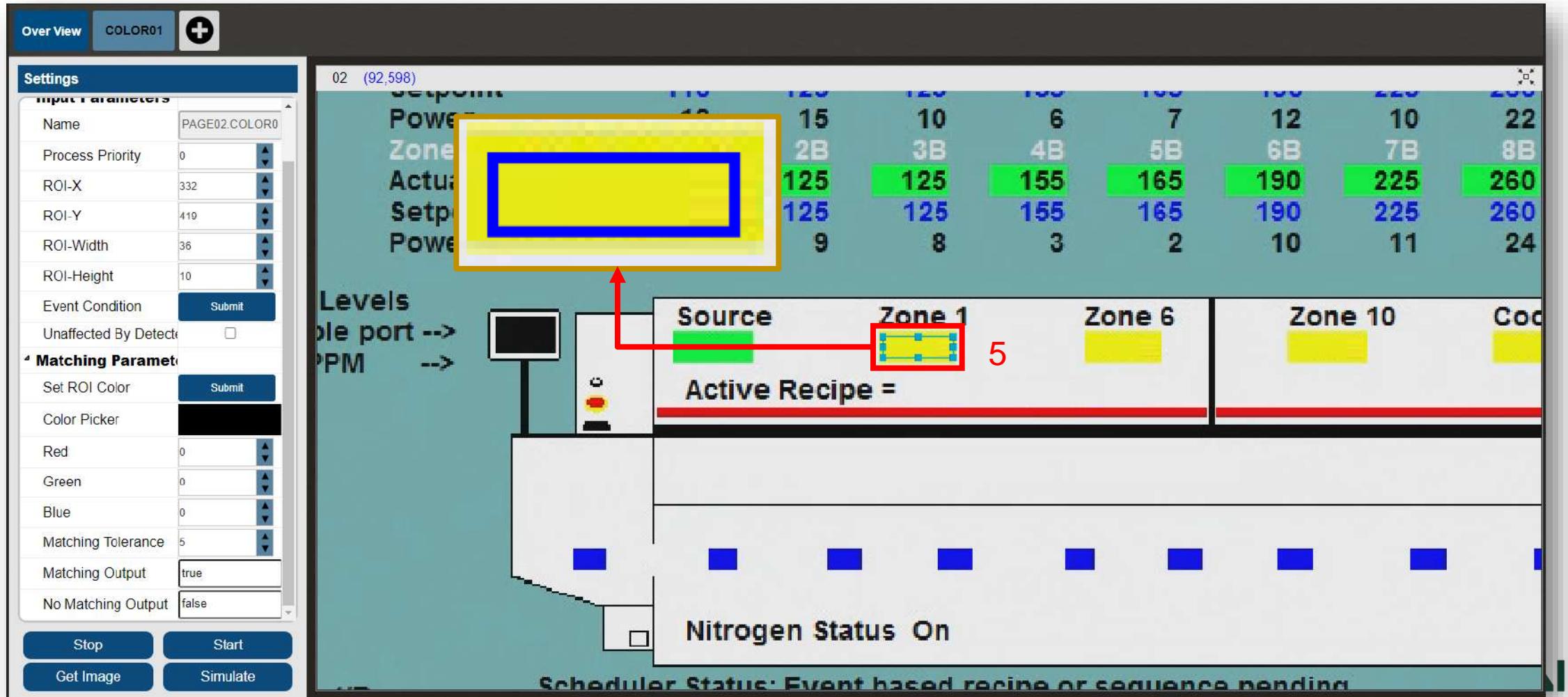
- 一般：新增顏色

4. 新增ROI：在圖像上點擊右鍵 → 選擇新增ROI



- 一般：新增顏色

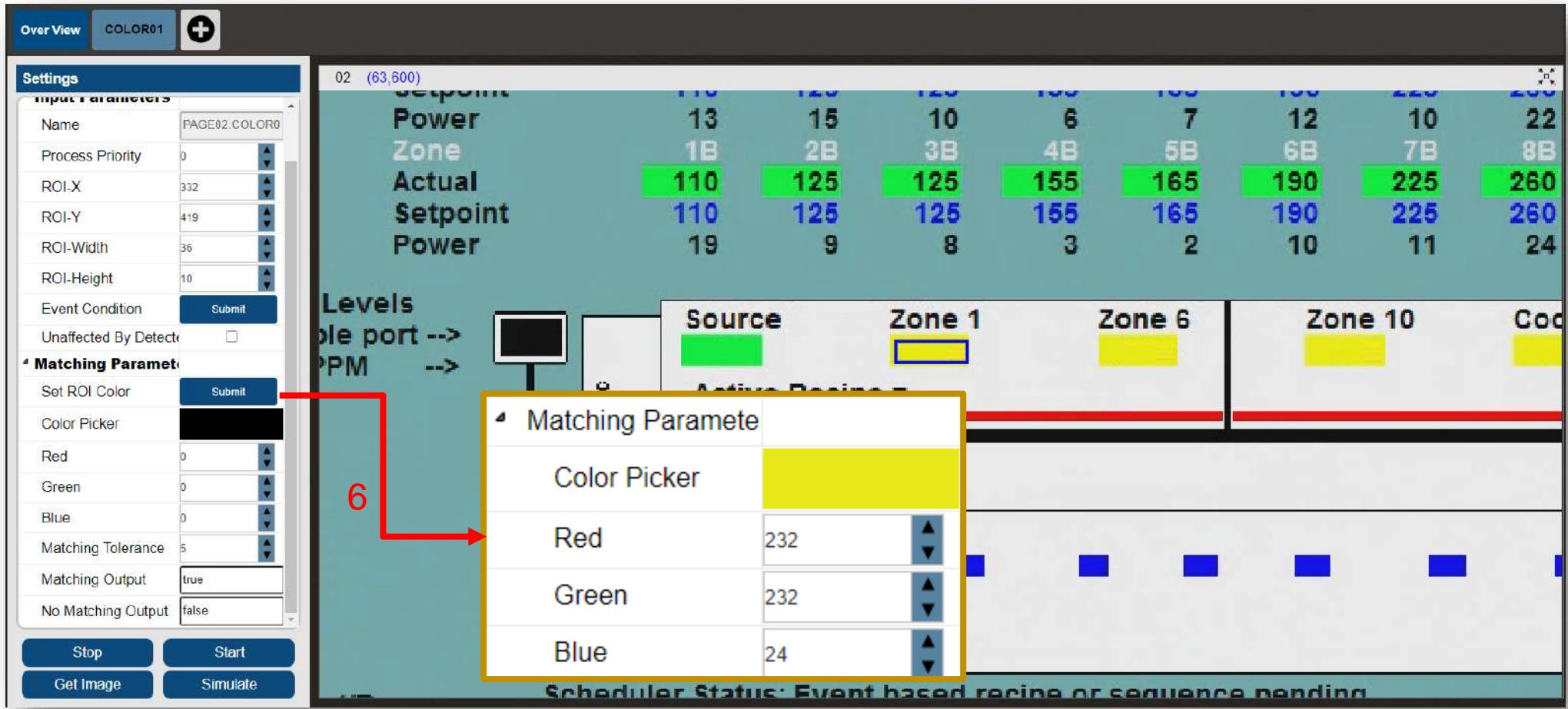
5. 設置ROI：調整ROI大小和位置 → 在圖像任意處點擊左鍵



- 一般：新增顏色

6. 設置比對顏色：點擊設置ROI顏色

系統會自動取得ROI內的 RGB 平均值，並將其設為顏色比對條件

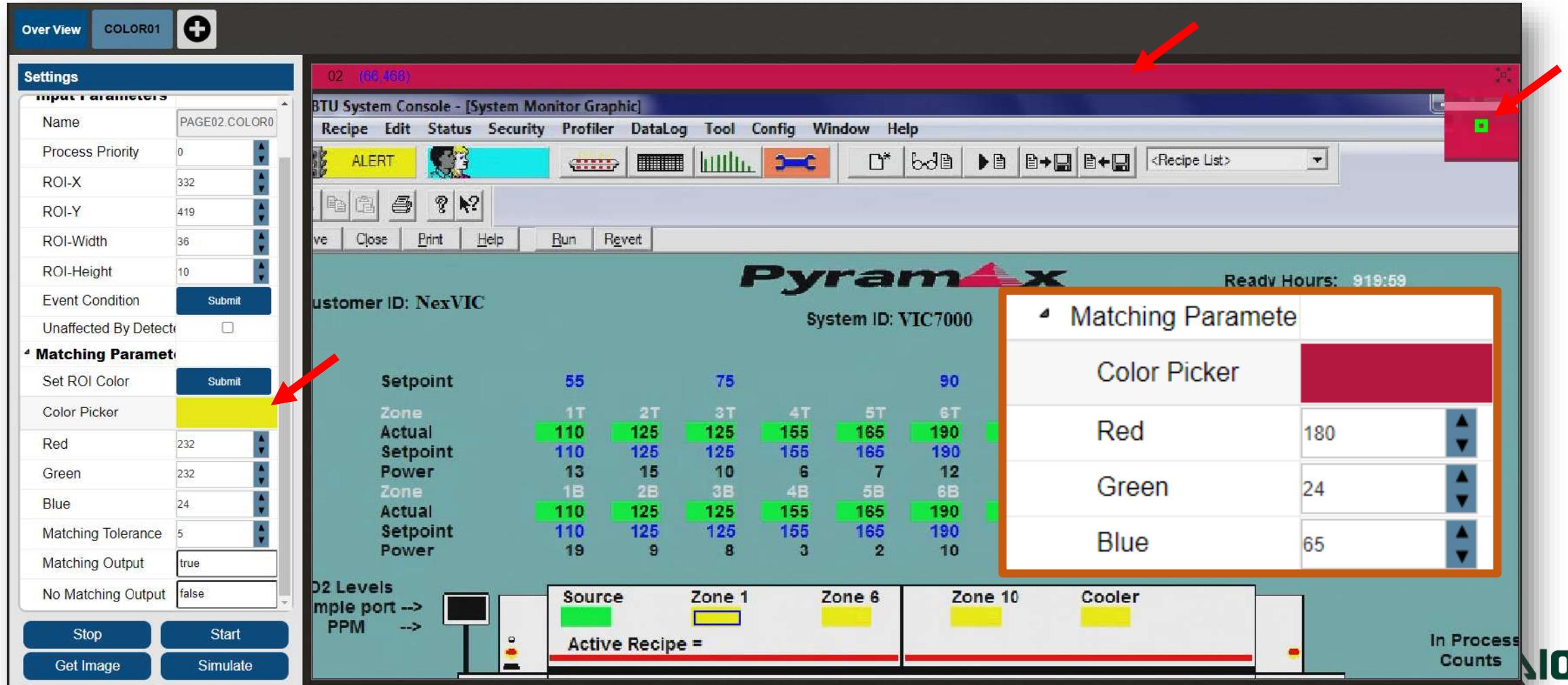


# 辨識工具

nexVIC

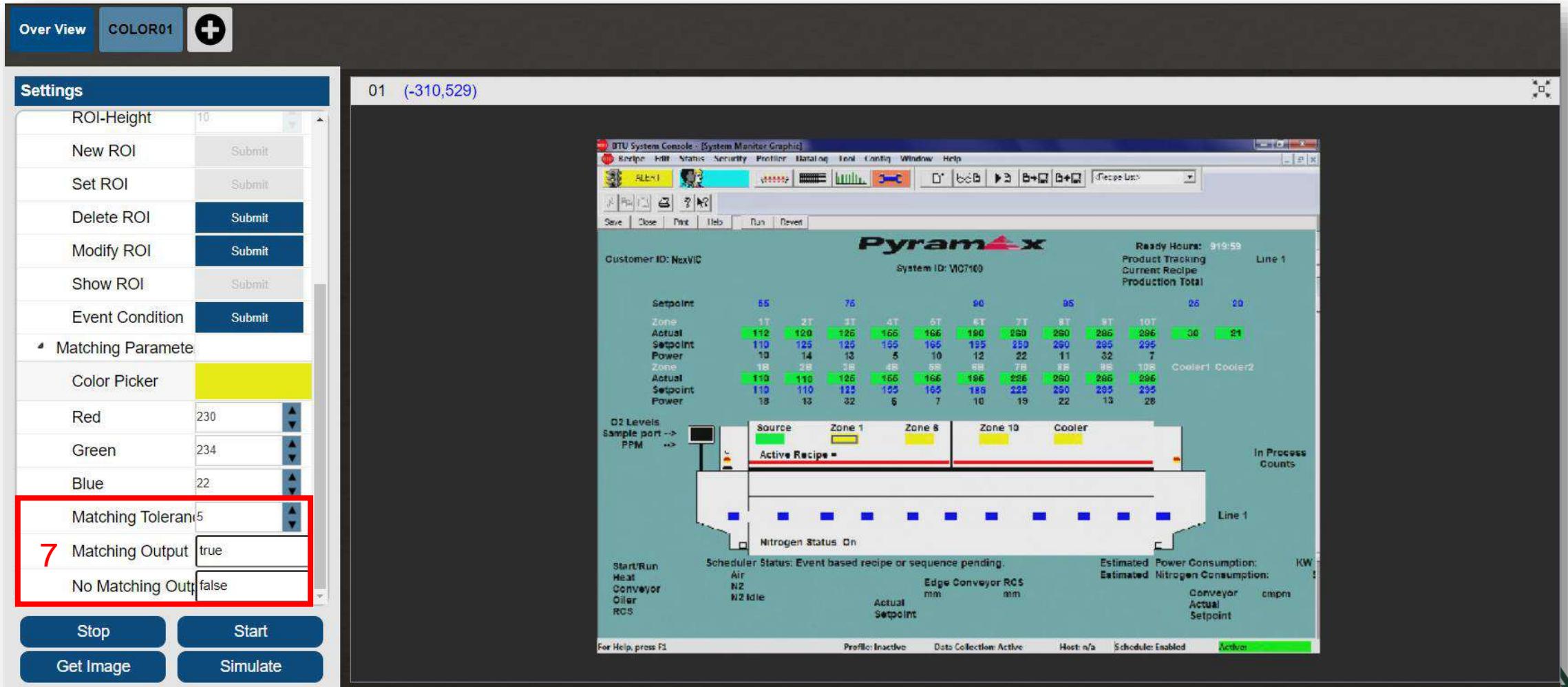
- 一般：新增顏色

- 自選顏色：點擊顏色選擇 → 點擊圖像中設為比對條件的顏色 完成顏色比對條件設定

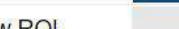


## • 一般：新增顏色

### 7. 設定比對容錯度和比對結果輸出：設定比對容錯度(RGB±) → 輸入比對輸出結果



The screenshot displays the nexVIC software interface. At the top, there are tabs for 'Over View', 'COLOR01' (which is selected and highlighted in blue), and a '+' icon. The 'COLOR01' tab contains a 'Settings' section with the following parameters:

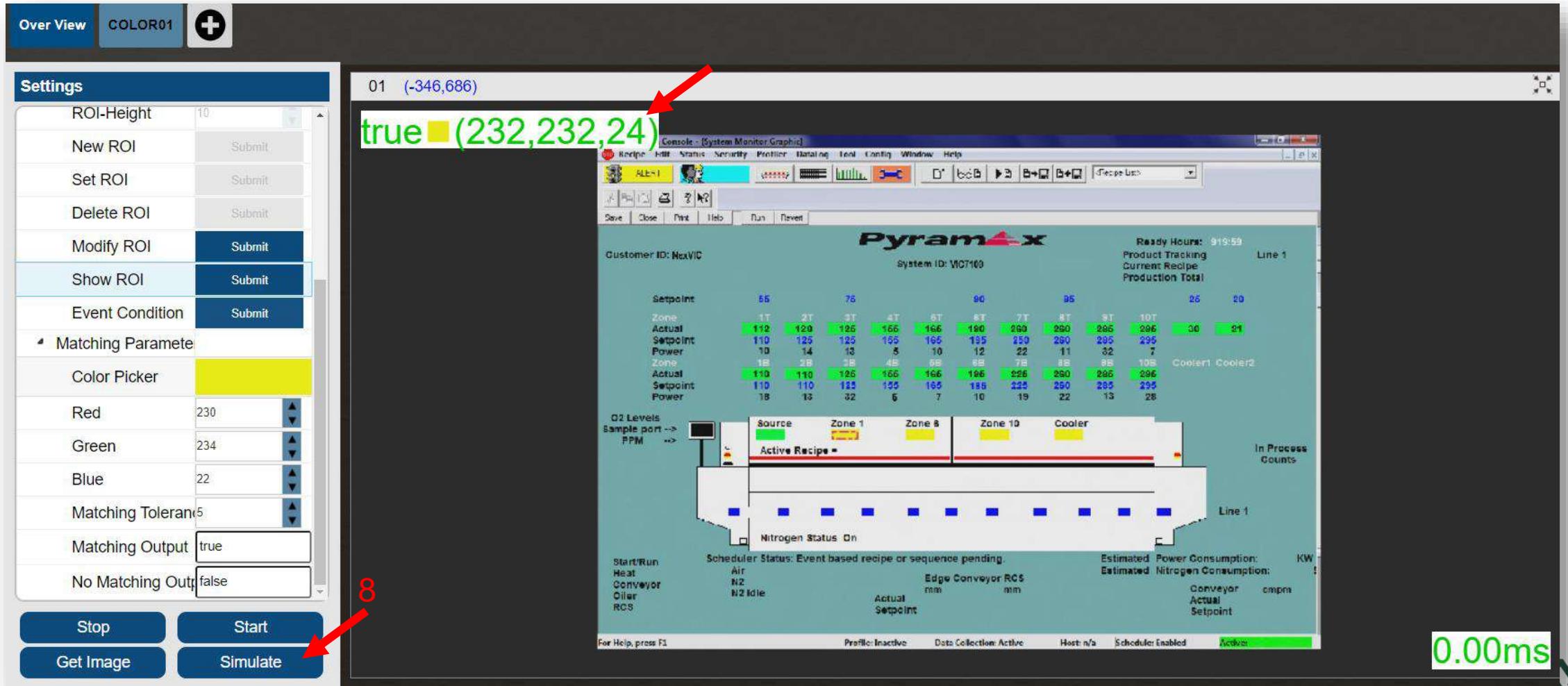
ROI-Height	10
New ROI	Submit
Set ROI	Submit
Delete ROI	Submit
Modify ROI	Submit
Show ROI	Submit
Event Condition	Submit
Matching Parameters	
Color Picker	
Red	230
Green	234
Blue	22
Matching Tolerance	5
Matching Output	<input checked="" type="checkbox"/> true
No Matching Output	<input type="checkbox"/> false

Below the settings are buttons for 'Stop', 'Start', 'Get Image', and 'Simulate'.

On the right side of the interface, a 'Pyramax' system monitor graphic is displayed. The graphic shows various operational data for a production line, including setpoints and actual values for different zones and coolers, as well as nitrogen status and conveyor information. The 'Customer ID: NexVIC' and 'System ID: V107100' are visible at the top of the monitor.

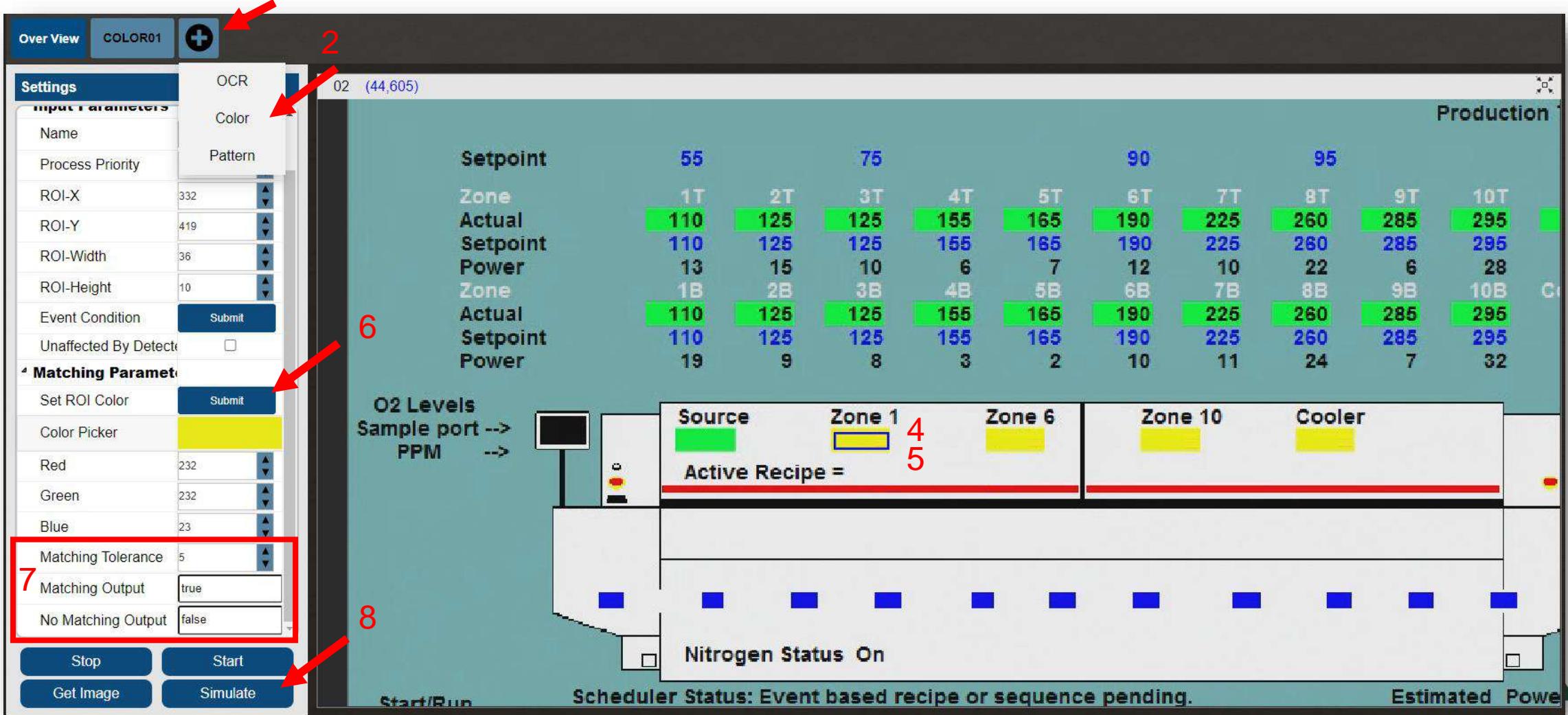
- 一般：新增顏色

- 模擬確認：點擊模擬 → 確認辨識結果



## • 一般：新增顏色

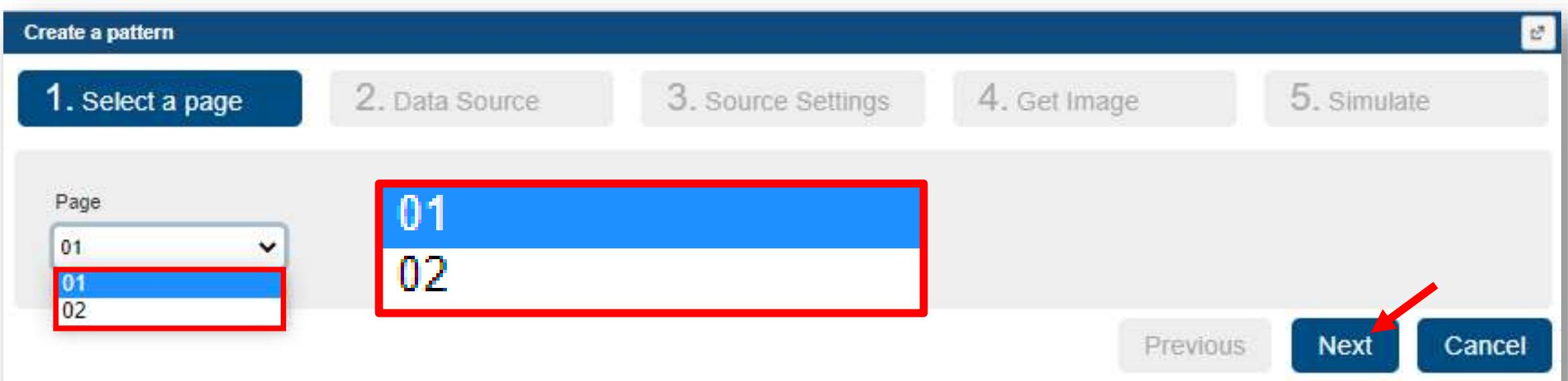
- **多個顏色**：在同一頻道和頁面上使用新增按鈕 → 點擊 Color → 依照前述步驟可新增更多顏色



- 精靈：新增樣式

1. 選擇頁面

選擇要新增樣式的頁面

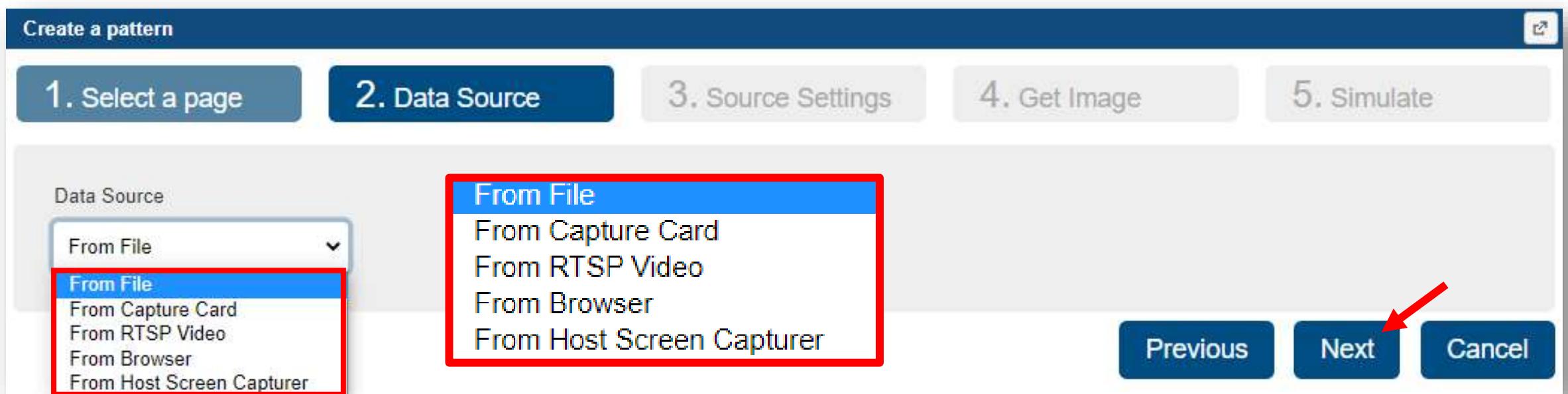


- 精靈：新增樣式

2. 影像來源

來自檔案、來自擷取卡、來自 RTSP 影像

來自瀏覽器、來自本機畫面擷取

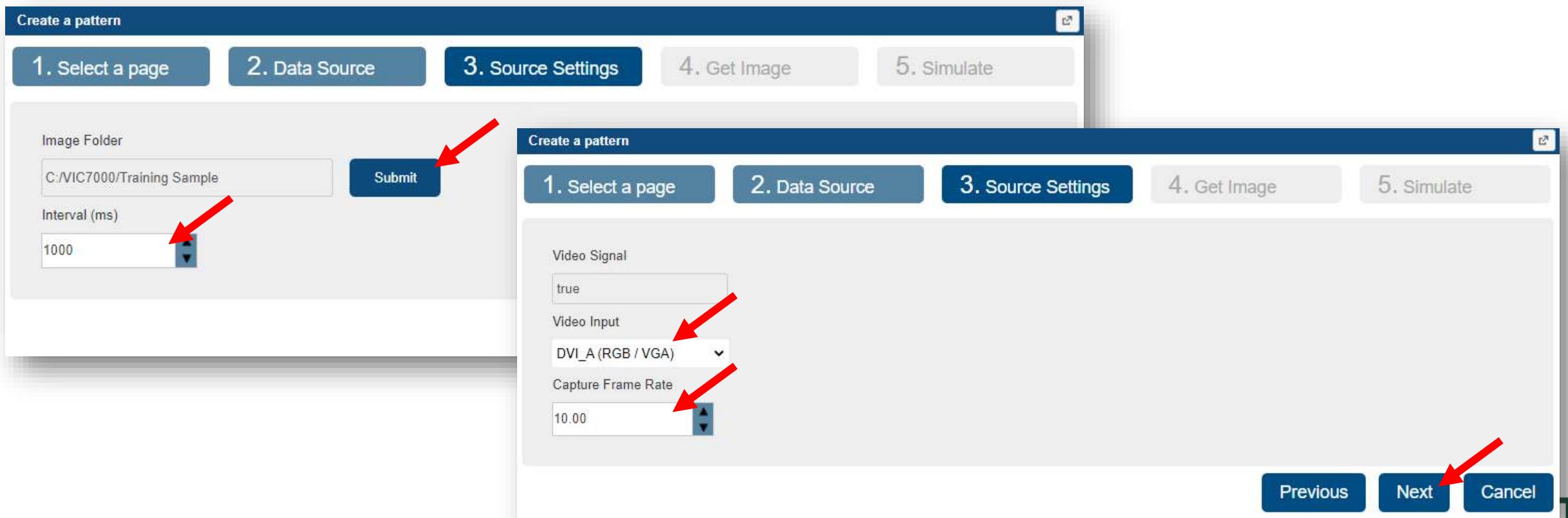


- 精靈：新增樣式

## 3. 影像來源設定

來自檔案：圖檔路徑、讀取時間間隔

來自擷取卡：輸入訊號類型、擷取頻率(fps)



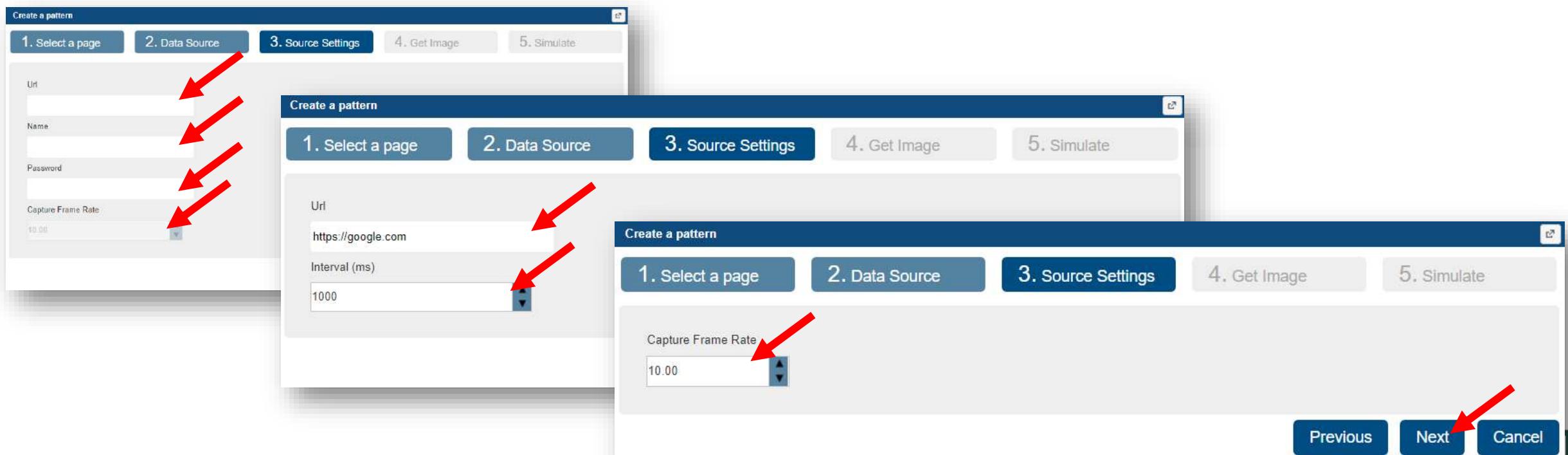
- 精靈：新增樣式

## 3. 影像來源設定

來自 RTSP 影像：RTSP 來源網址、名稱、密碼、擷取頻率

來自瀏覽器：網址、讀取時間間隔

來自本機畫面擷取：擷取頻率(fps)



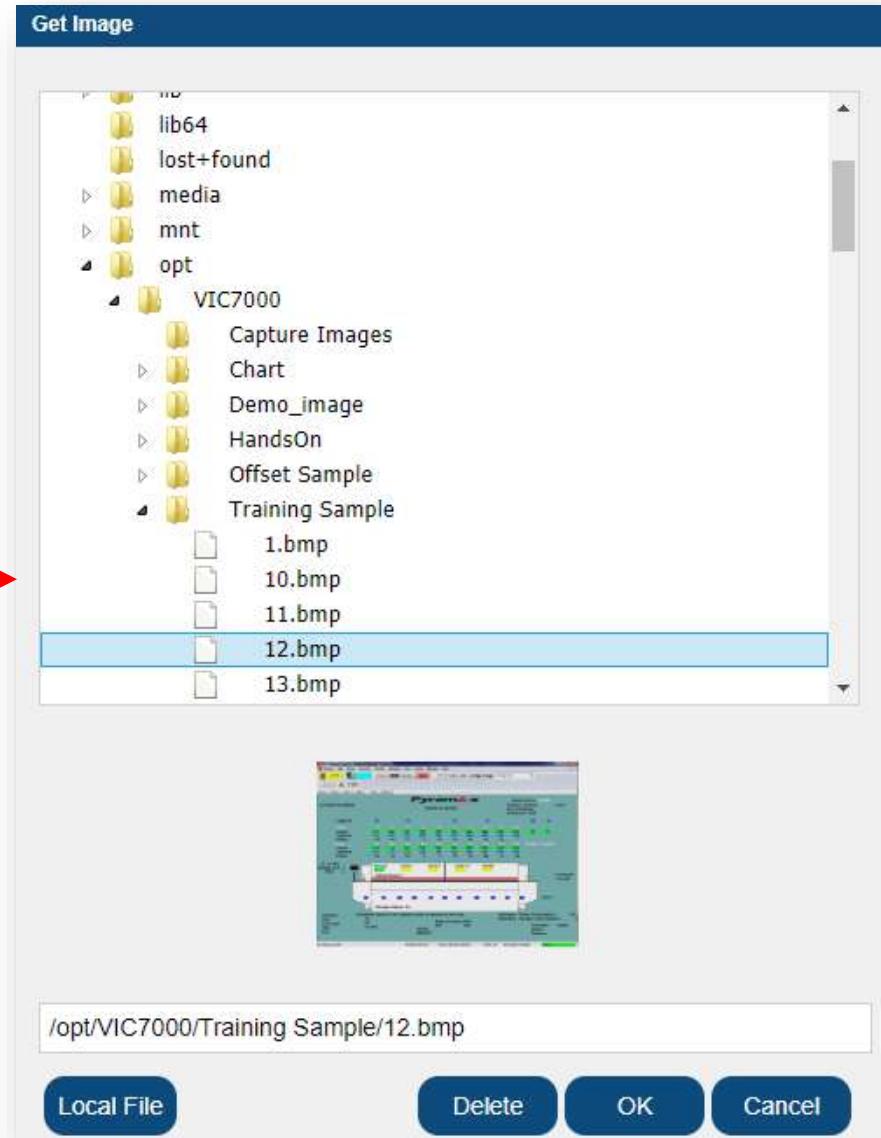
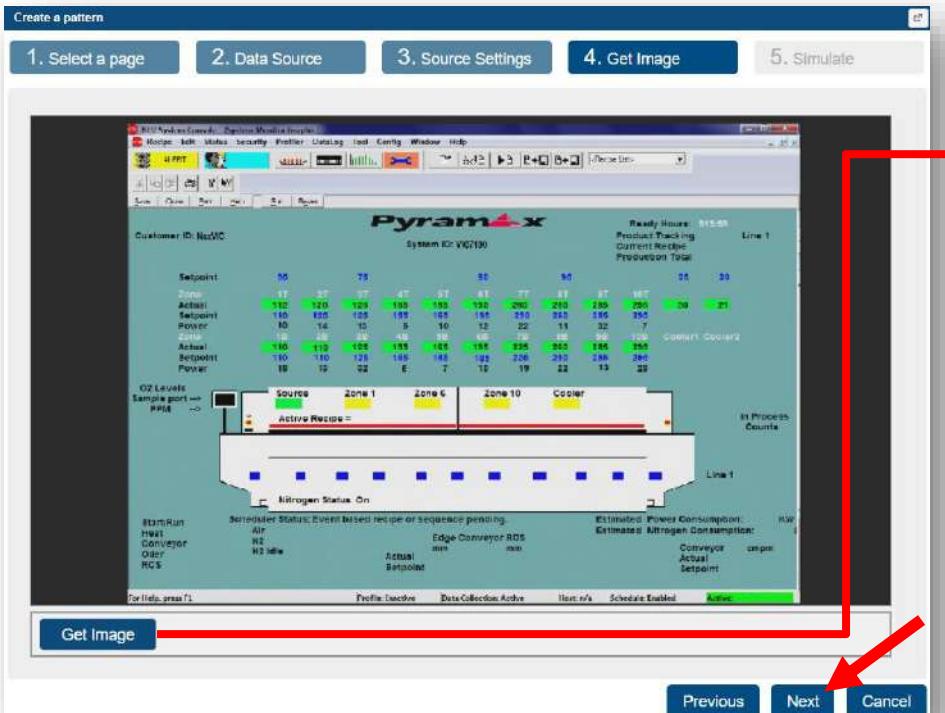
- 精靈：新增樣式

## 4. 取得圖像

來自檔案：讀取本機或遠端的圖像

來自擷取卡、來自 RTSP 影像、來自瀏覽器、

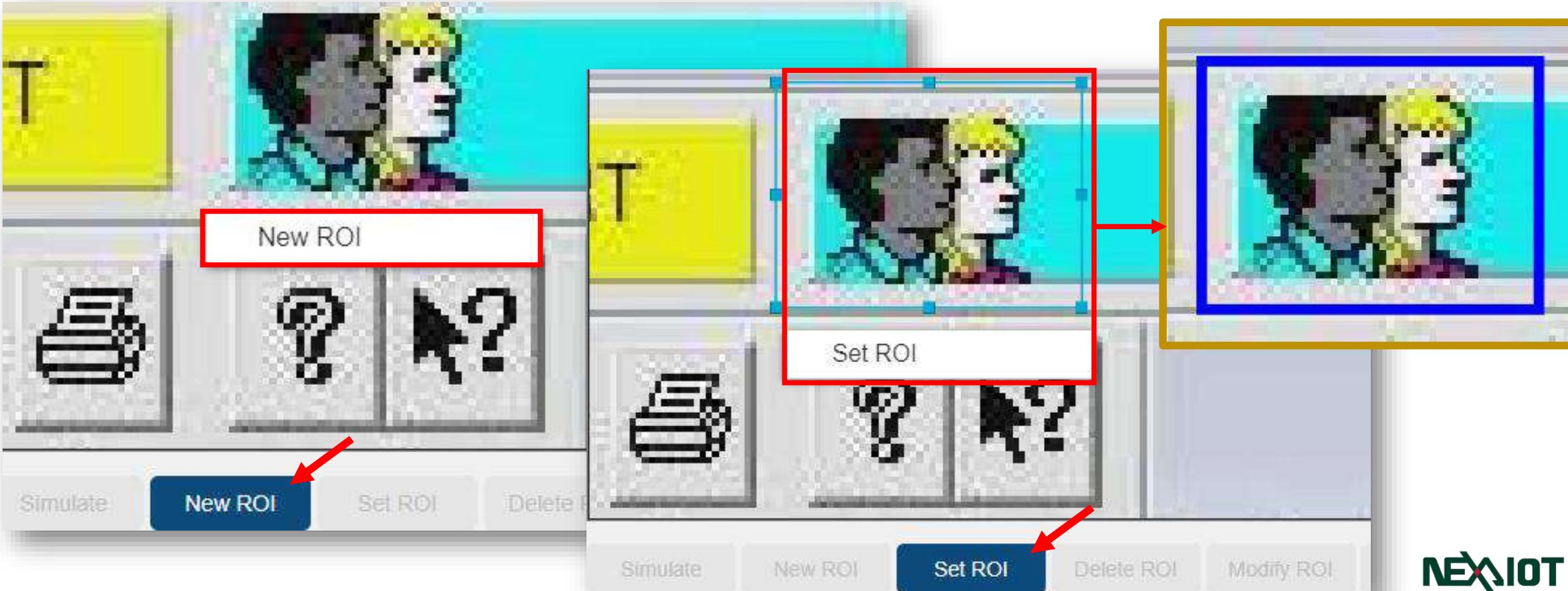
來自本機畫面擷取：從取像來源取得一張圖像



- 精靈：新增樣式

## 5. 模擬

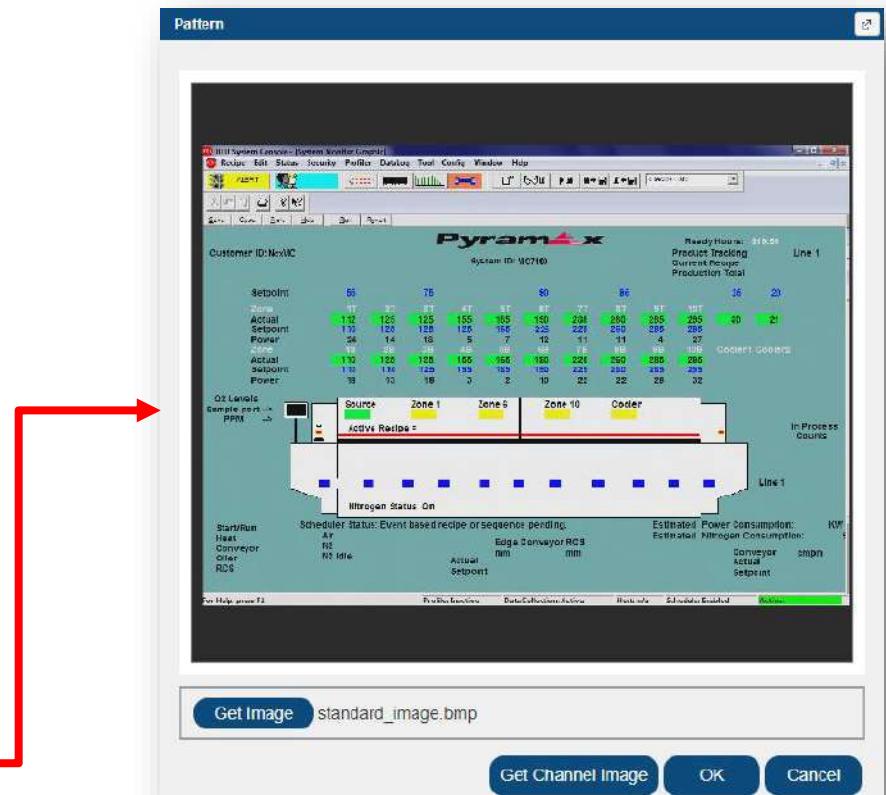
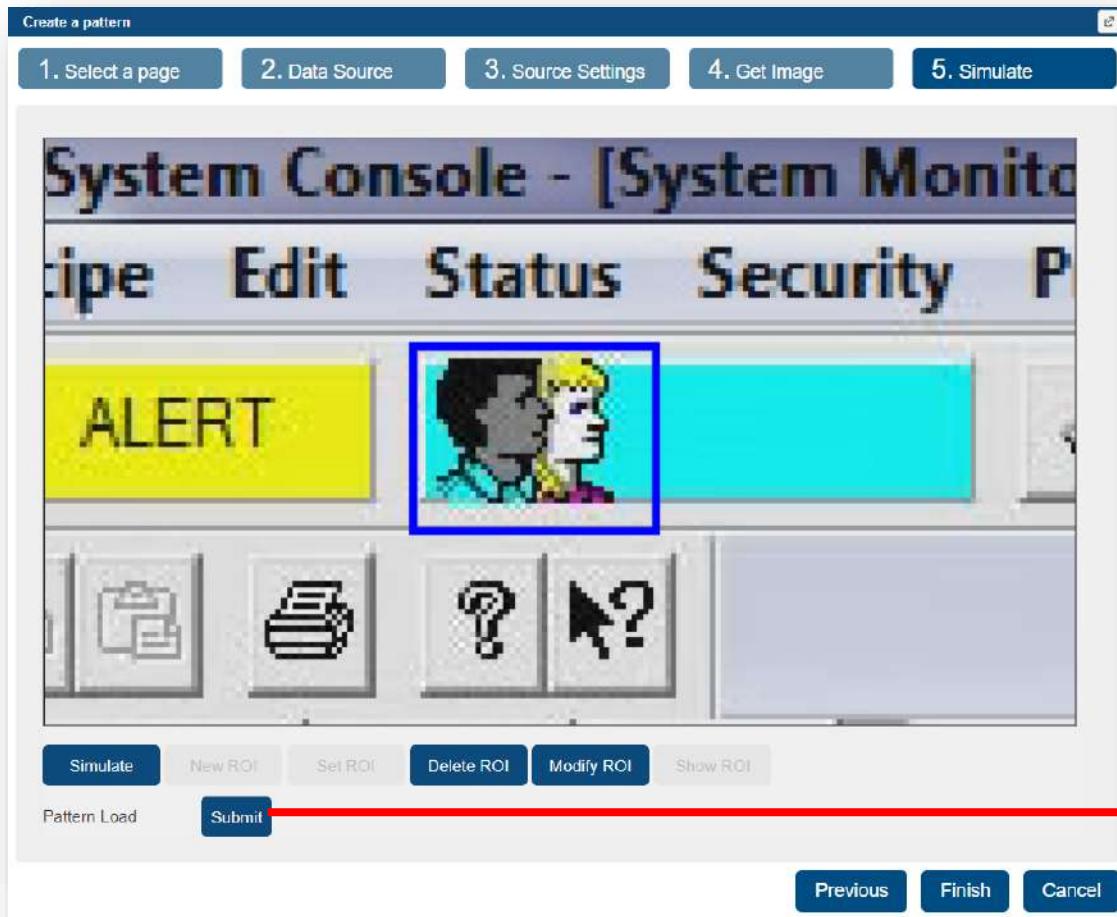
右鍵新增ROI or 點擊 新增ROI → 調整大小和位置 → 右鍵設置ROI or 點擊設置ROI



- 精靈：新增樣式

## 5. 模擬

點擊樣式載入的執行，開啟樣式視窗

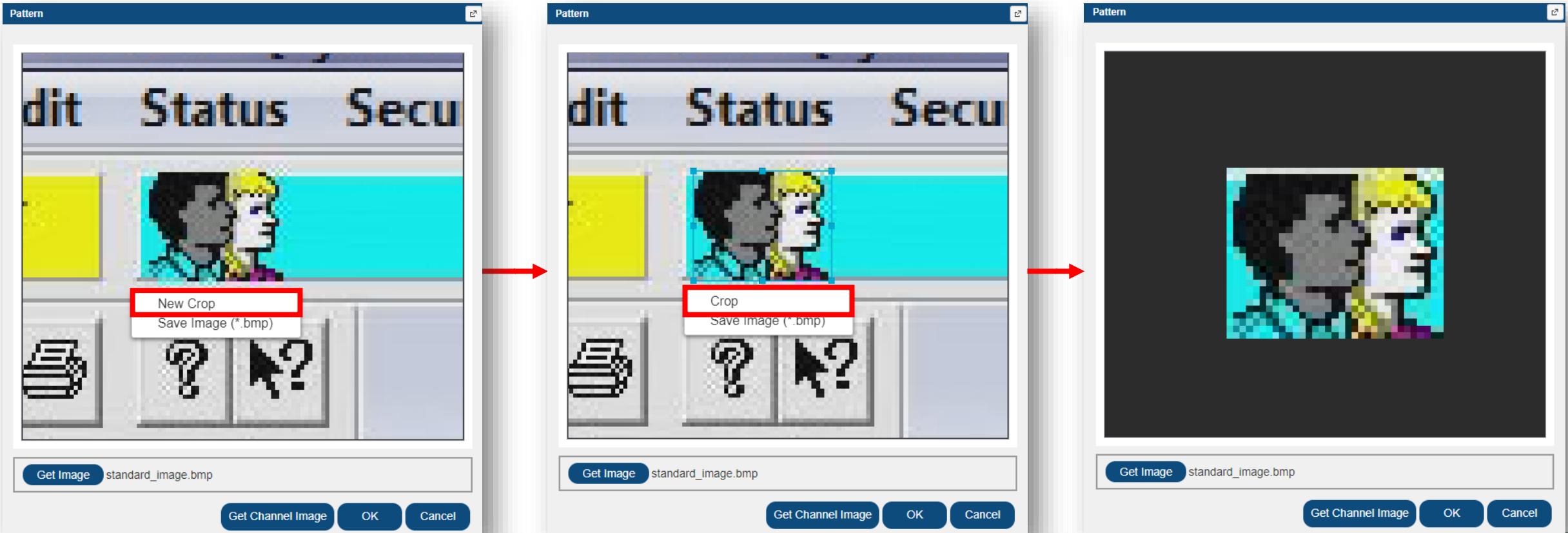


# 辨識工具

- 精靈：新增樣式

## 5. 模擬

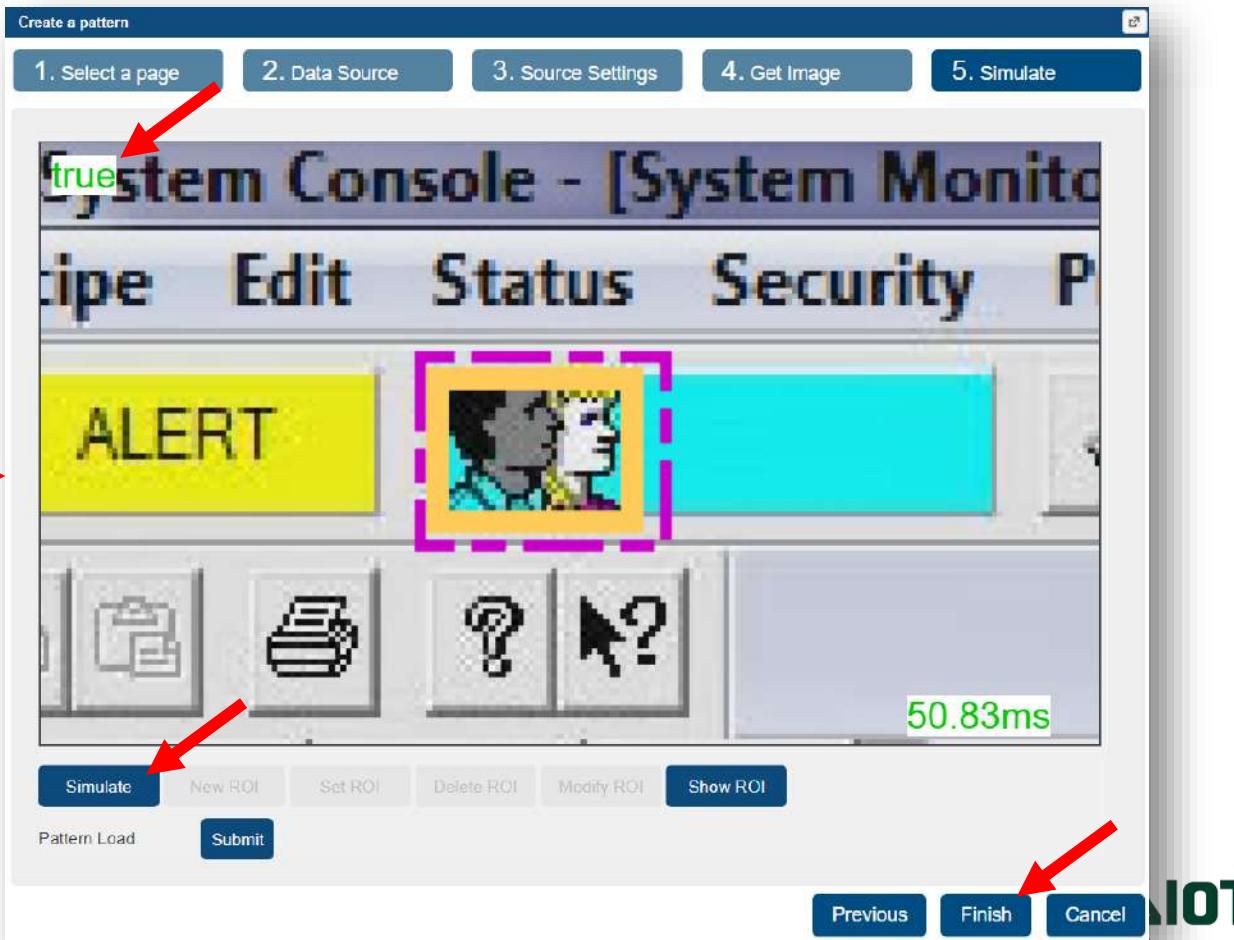
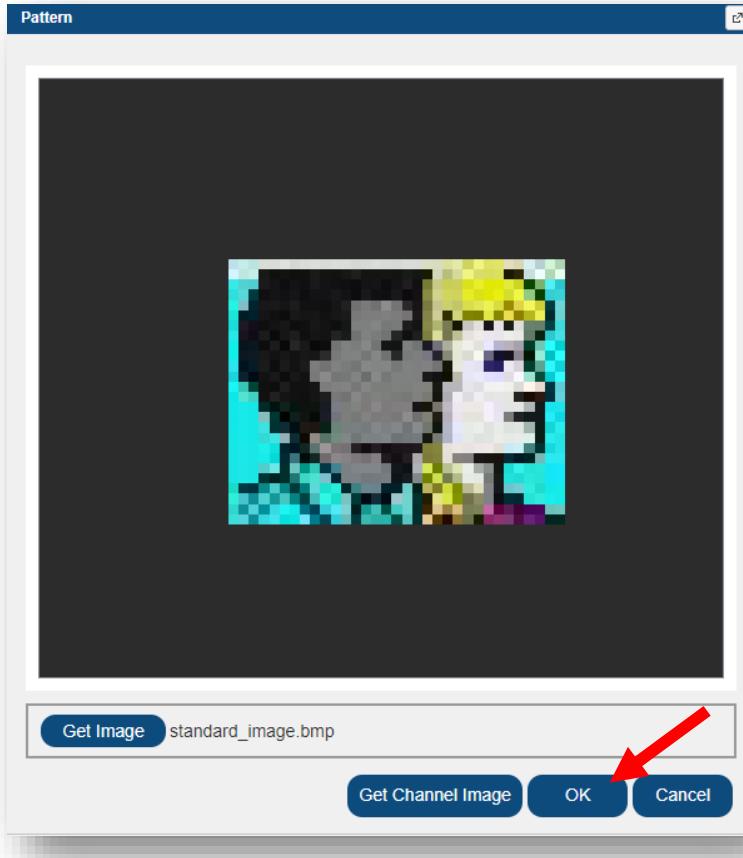
右鍵選擇建立裁切 → 調整ROI大小和位置 → 右鍵選擇裁切 選擇的樣式就會被裁切出來



- 精靈：新增樣式

## 5. 模擬

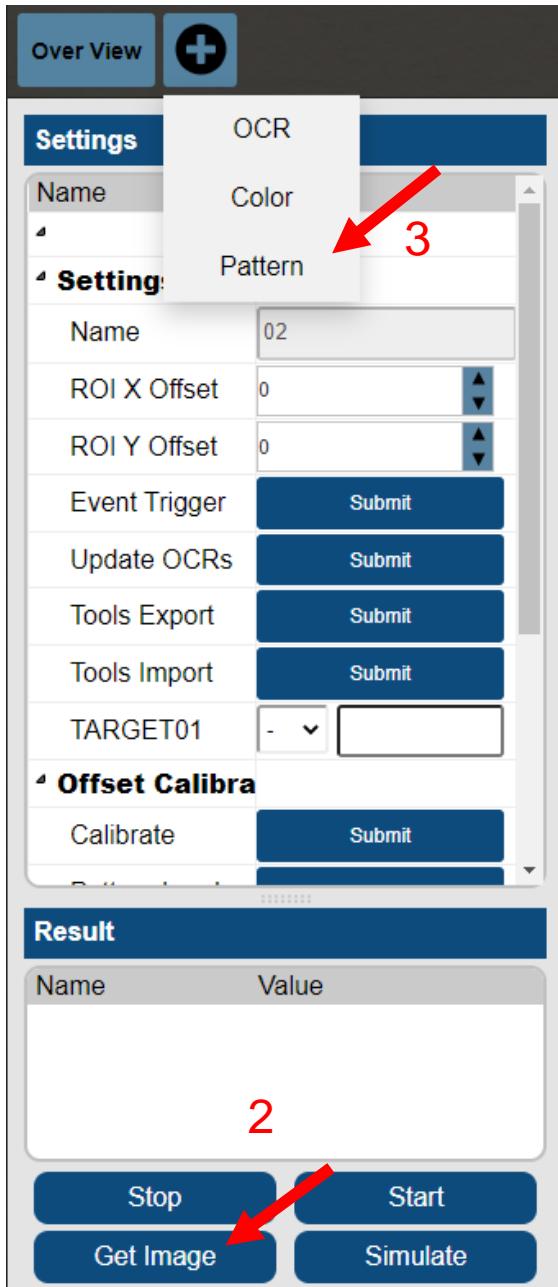
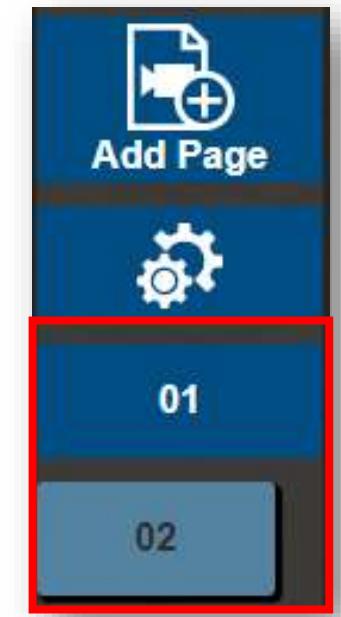
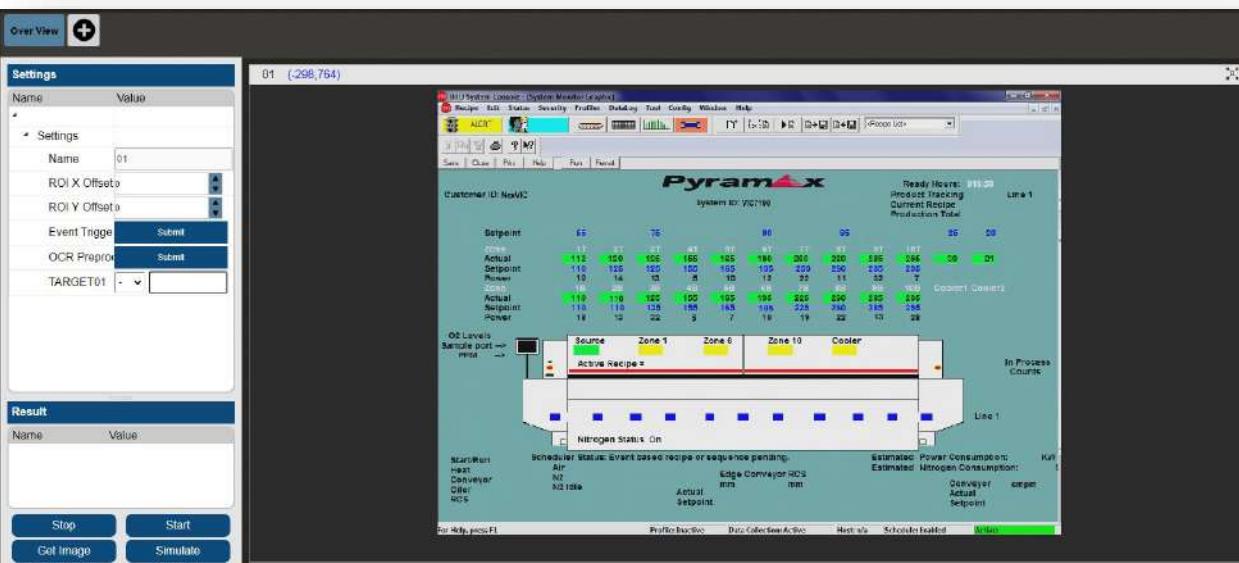
點擊確定關閉樣式視窗，點擊模擬 → 確認辨識值



# 辨識工具

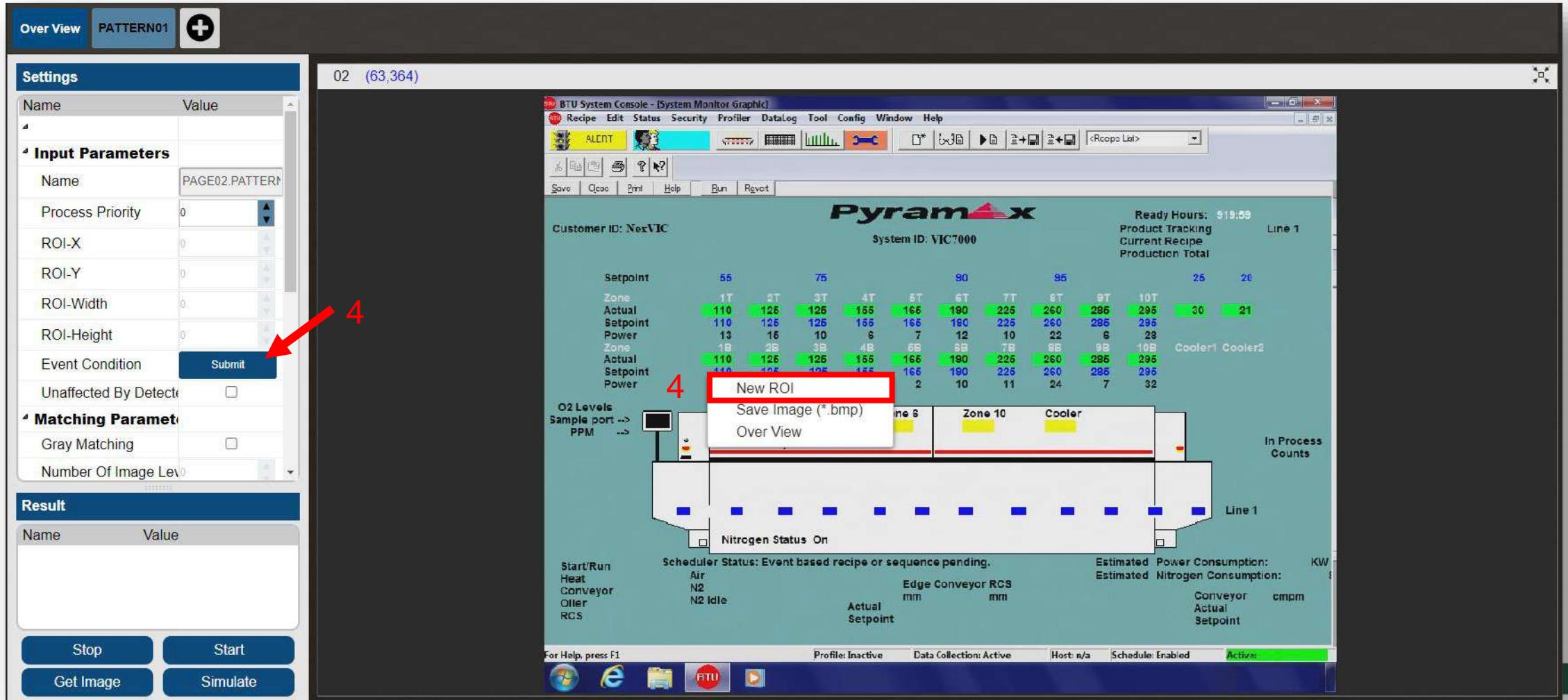
## • 一般：新增樣式

1. 選擇頁面：進入要新增樣式的頁面
2. 取得圖像
  - a. 來自檔案：讀取本地或遠端的圖像
  - b. 來自擷取卡、RTSP 影像、瀏覽器  
、本機畫面擷取：從來源取得一張圖像
3. 新增樣式：滑鼠移到新增按鈕 → 點擊 Pattern



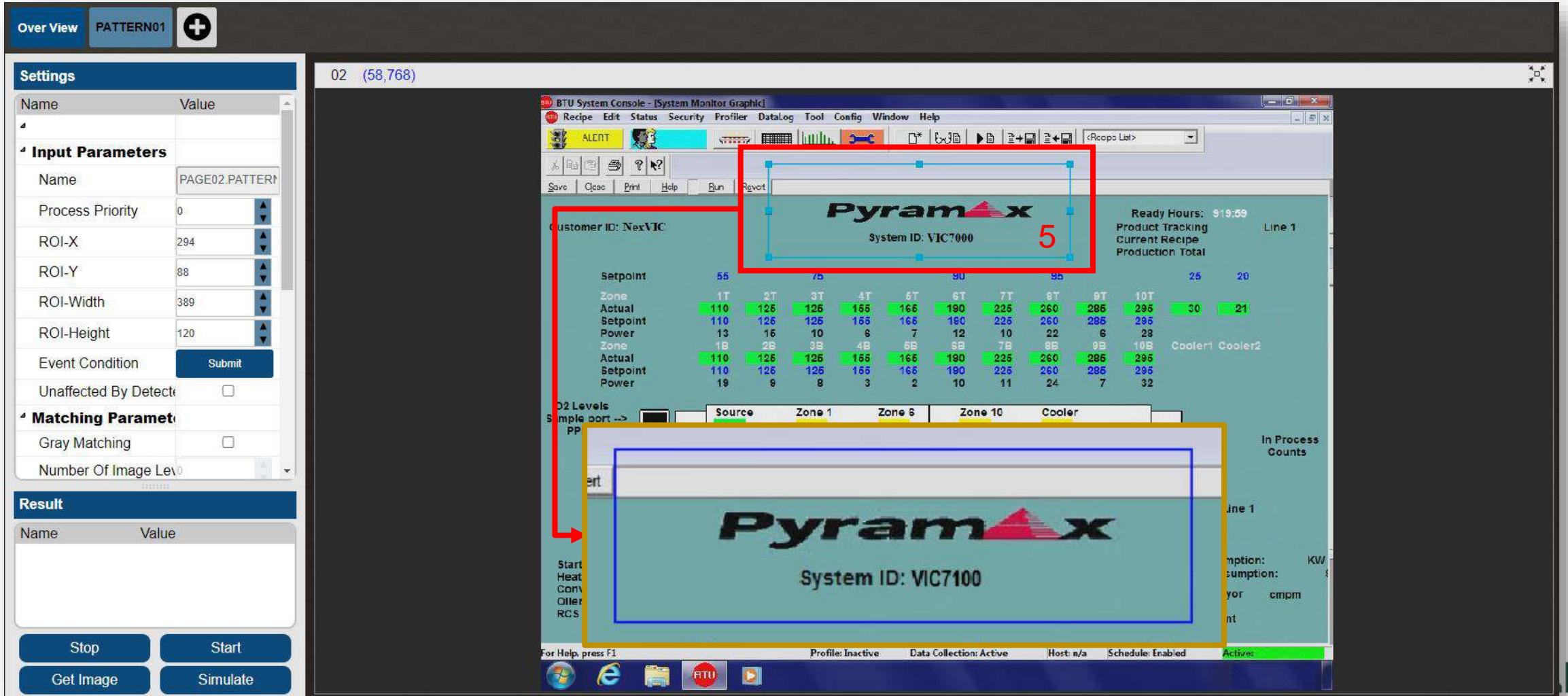
- 一般：新增樣式

- 新增ROI：在圖像上點擊右鍵 → 選擇新增ROI



- 一般：新增樣式

5. 設置ROI：調整ROI大小和位置 → 在圖像任意處點擊左鍵

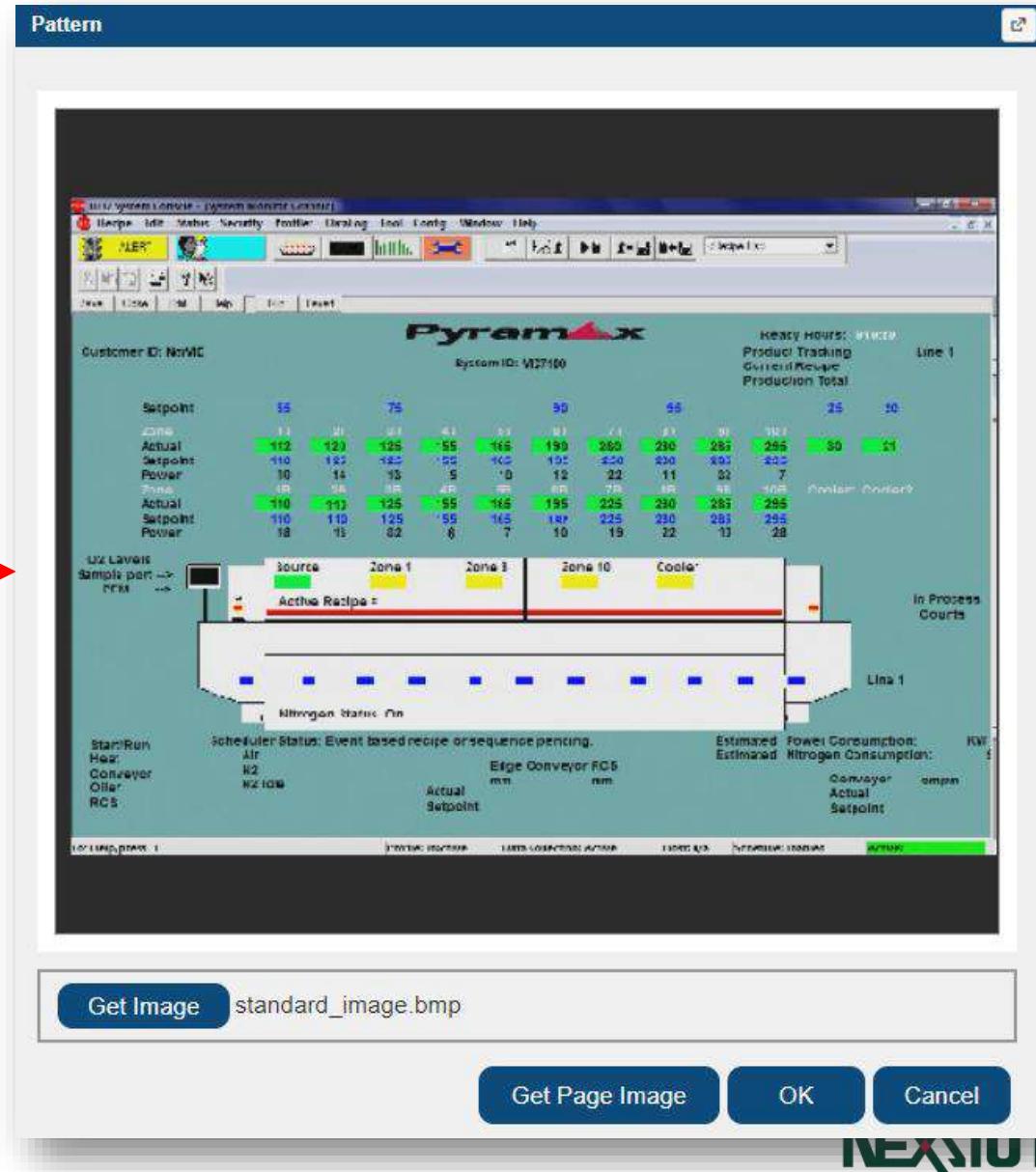


- 一般：新增樣式

- ## 6. 設置比對樣式：點擊樣式載入，進入裁切視窗

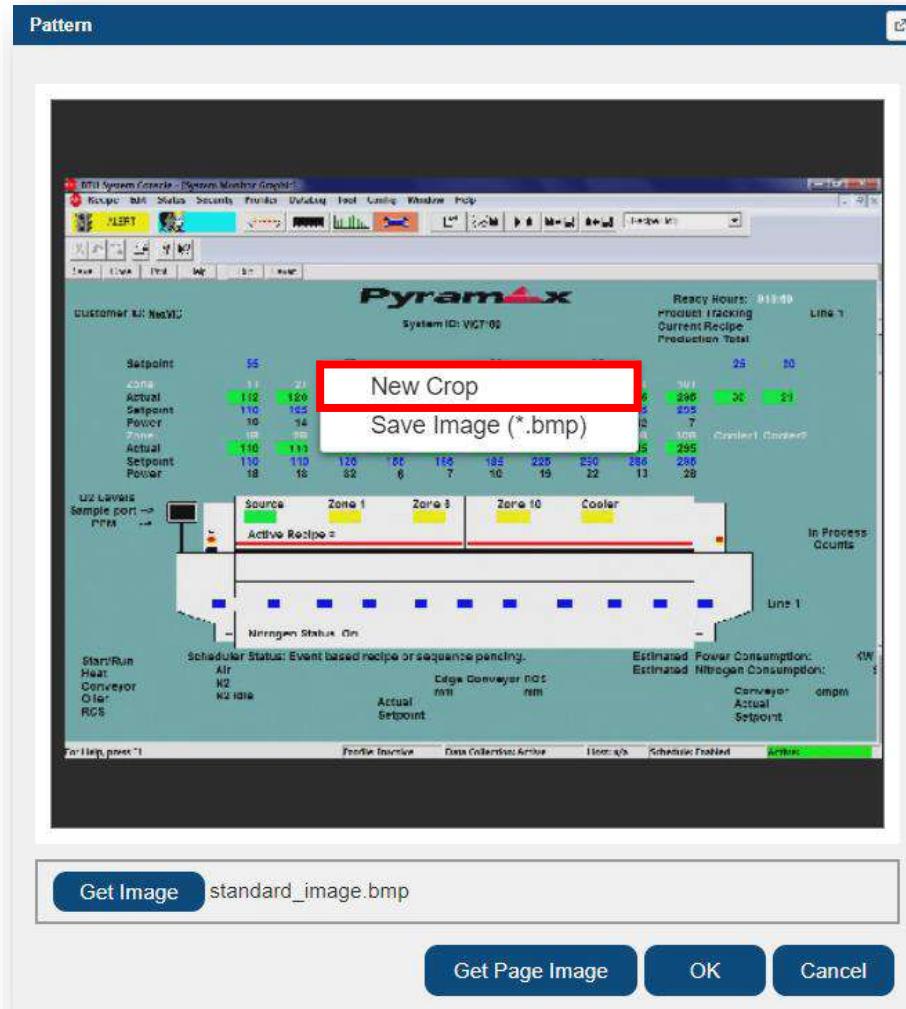
圖像會被自動載入

Over View	PATTERN01	
<h2>Settings</h2>		
No Matching Output <input checked="checked" type="checkbox"/>		
<h3>Pattern</h3>		
Pattern Load	<input type="button" value="Submit"/>	
Pattern Name		
Pattern X	0	
Pattern Y	0	
Pattern Width	0	
Pattern Height	0	



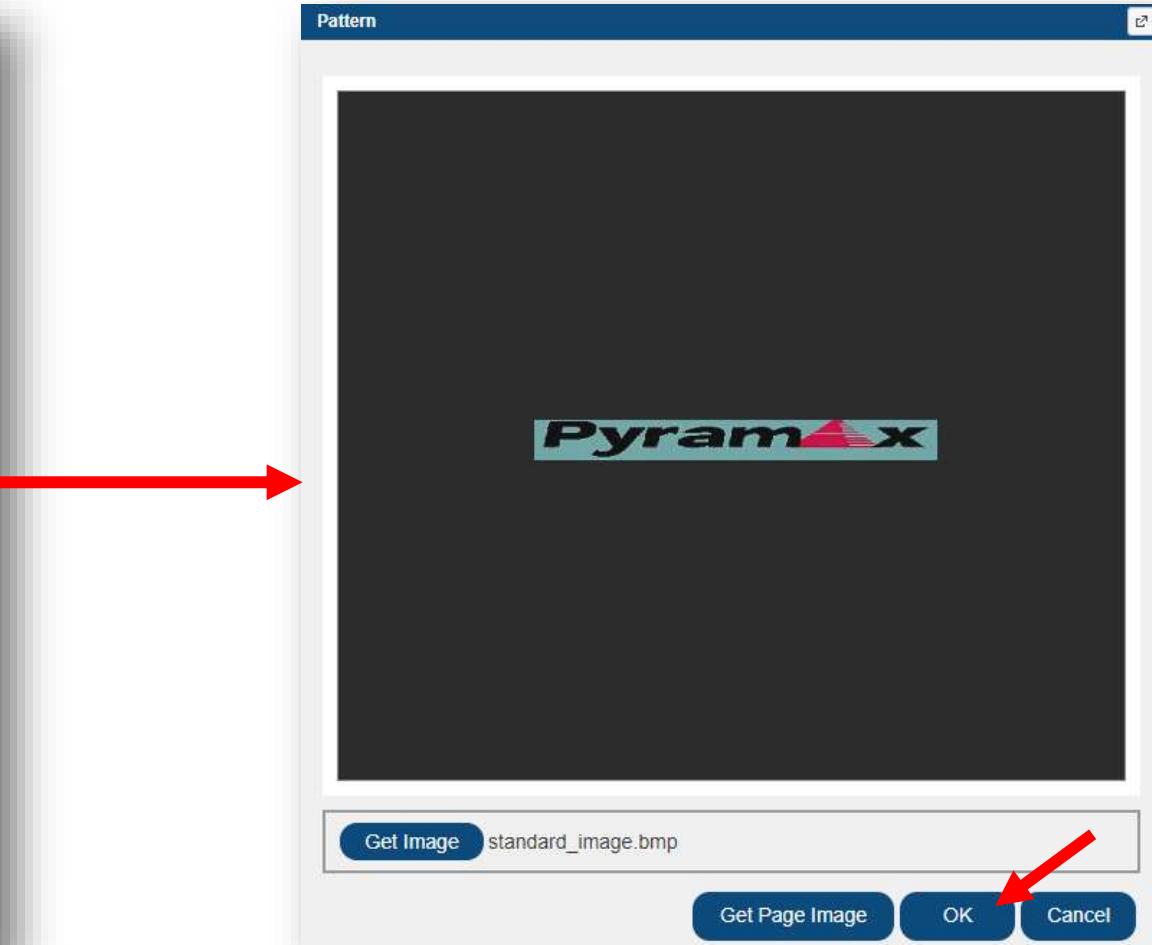
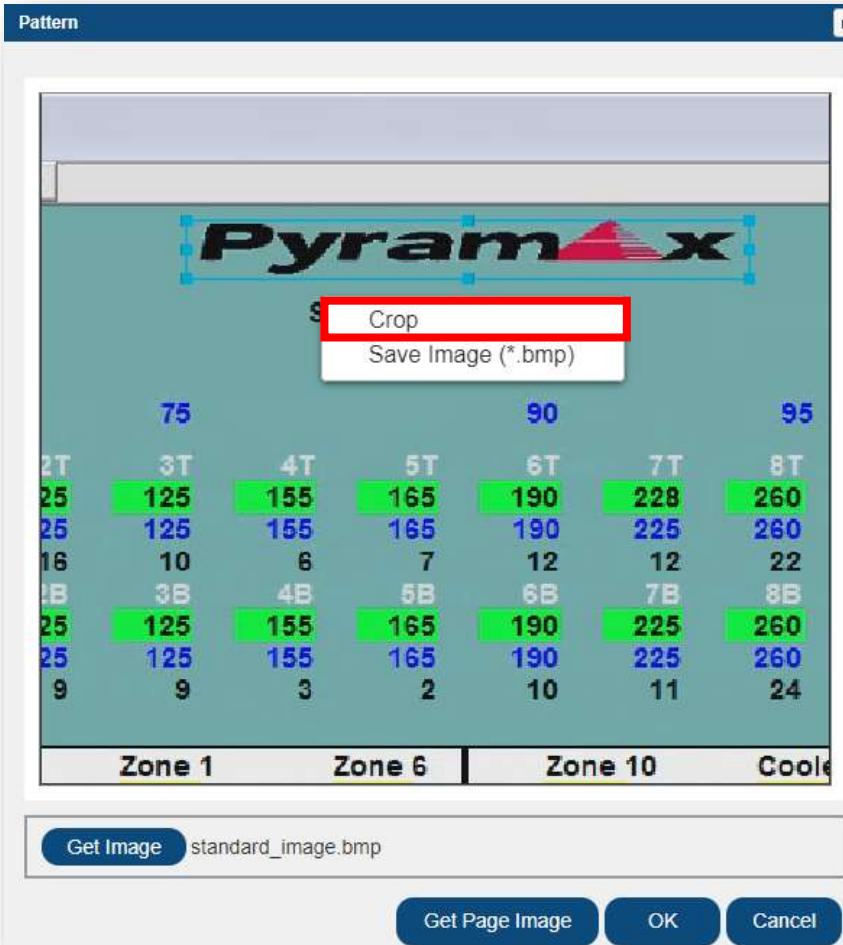
- 一般：新增樣式

6. 設置比對樣式：在圖像上點擊右鍵 → 選擇建立裁切



- 一般：新增樣式

- 設置比對樣式：選擇要比對的樣式 → 在圖像上點擊右鍵 → 選擇裁切 → 點擊確定



- 一般：新增樣式

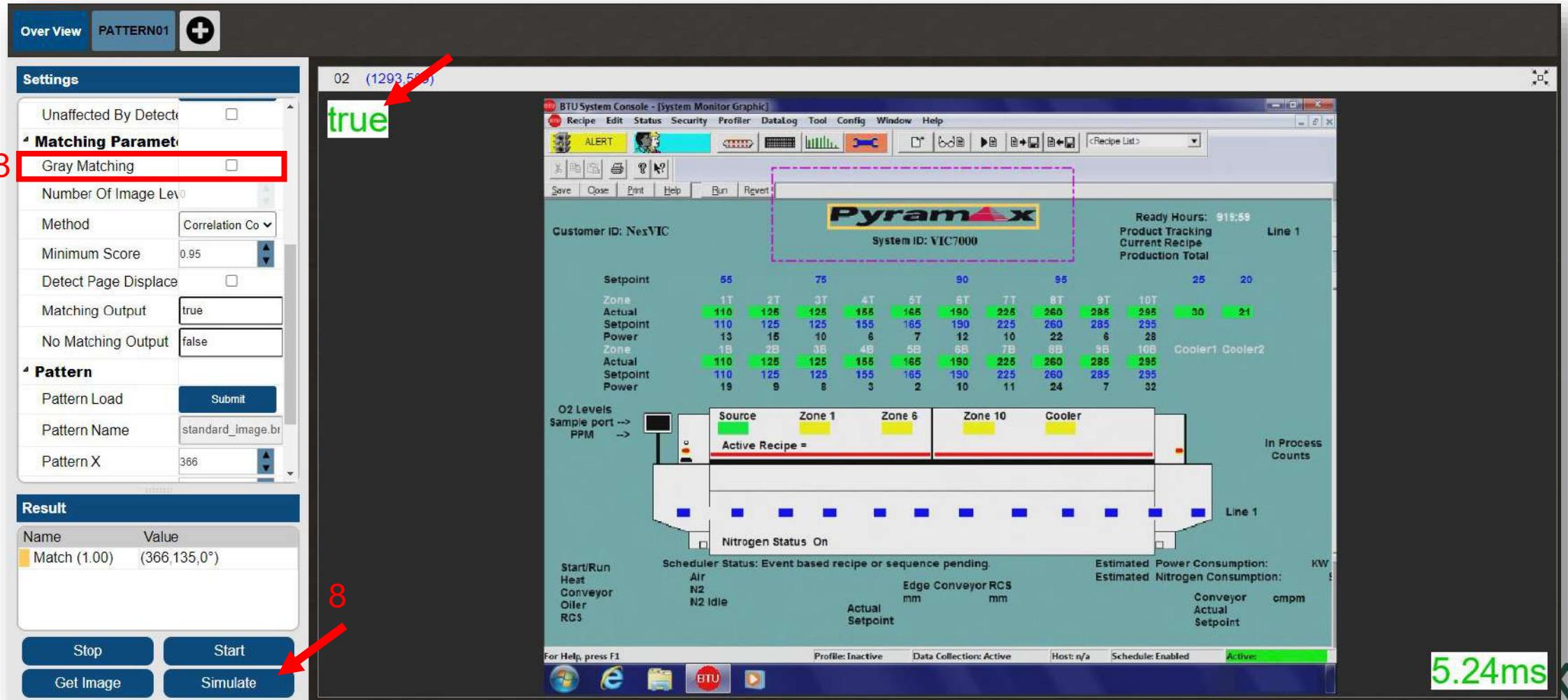
## 7. 設定比對結果輸出：輸入比對結果輸出

The screenshot displays a process control software interface with the following components:

- Top Bar:** Over View, PATTERN01, and a plus sign icon.
- Left Panel (Settings):**
  - Unaffected By Detect:
  - Matching Parameters:**
    - Gray Matching:
    - Number Of Image Levels:
    - Method: Correlation Co.
    - Minimum Score: 0.95
    - Detect Page Displace:
    - Matching Output:** true (highlighted with a red box)
    - No Matching Output: false
  - Pattern Load:
  - Pattern Name: standard\_image.br
  - Pattern X: 366- Result Panel:** A table with columns Name and Value, currently empty.
- Bottom Buttons:** Stop, Start, Get Image, Simulate.
- Central Monitor Graphic:** BTU System Console - System Monitor Graphic for Pyramax. The graphic shows:
  - Customer ID: NEXVIC, System ID: VIC7000
  - Ready Hours: 919:59
  - Product Tracking: Current Recipe, Production Total
  - Line 1: Production data for zones 55, 75, 90, 95, 25, 20. The table includes columns for Setpoint, Actual, Power, and Zone.
  - CO<sub>2</sub> Levels: Sample port --> PPM --> Source, Zone 1, Zone 6, Zone 10, Cooler.
  - Active Recipe =
  - Nitrogen Status: On
  - Scheduler Status: Event based recipe or sequence pending.
  - Estimated Power Consumption: KW
  - Estimated Nitrogen Consumption: cmppm
  - Conveyor Actual Setpoint
  - Start/Run, Heat, Conveyor, Oiler, RCS buttons.
  - Edge Conveyor RCS: Actual mm, Setpoint mm.
  - For Help, press F1, Profile: Inactive, Data Collection: Active, Host: n/a, Schedule: Enabled, Active.

## • 一般：新增樣式

8. 模擬確認：建議開啟灰階比對，加快辨識速度 → 點擊模擬 → 確認辨識結果



5.24ms

IOT

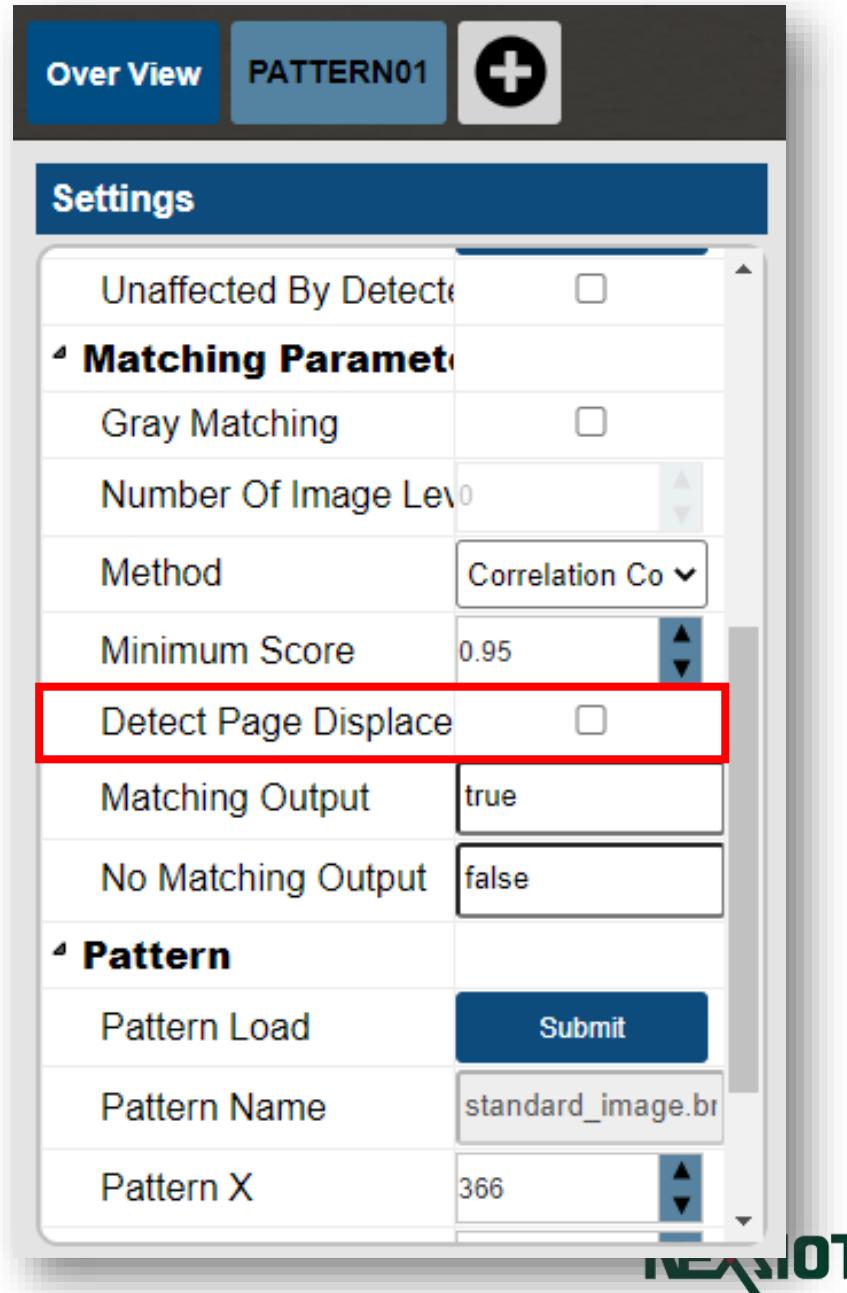
# 辨識工具

- 一般：新增樣式

- **設定是否偵測位移**：以設定樣式的圖像上的樣式位置為基準

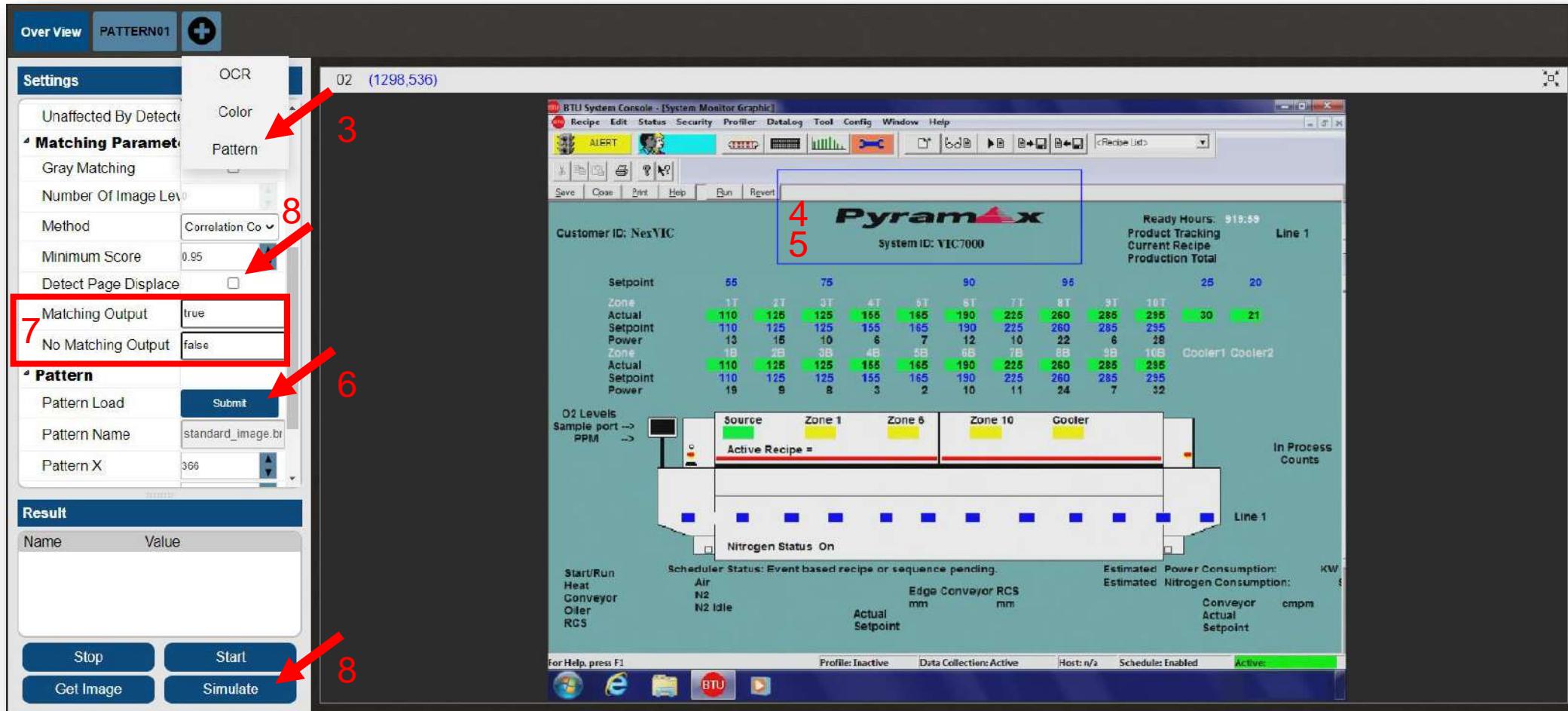
**偵測頁面位移**：讓同頁面的OCR和顏色的ROI隨著樣式所在位置不同而移動

**Detect Page Displacement**



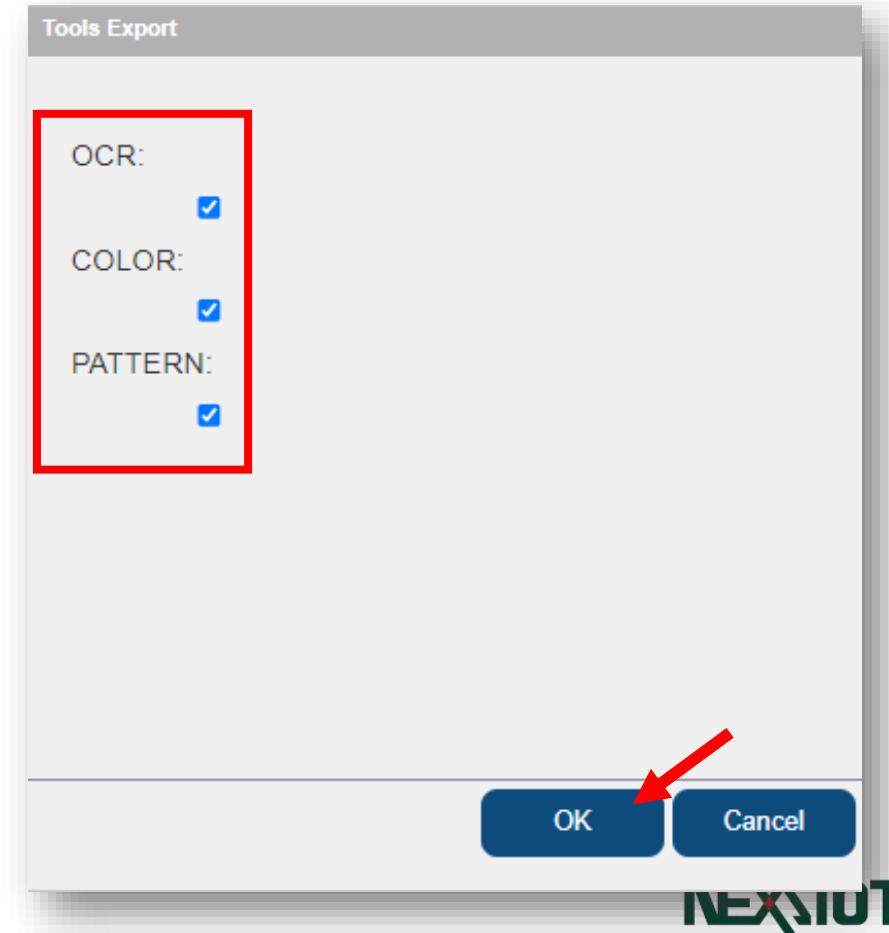
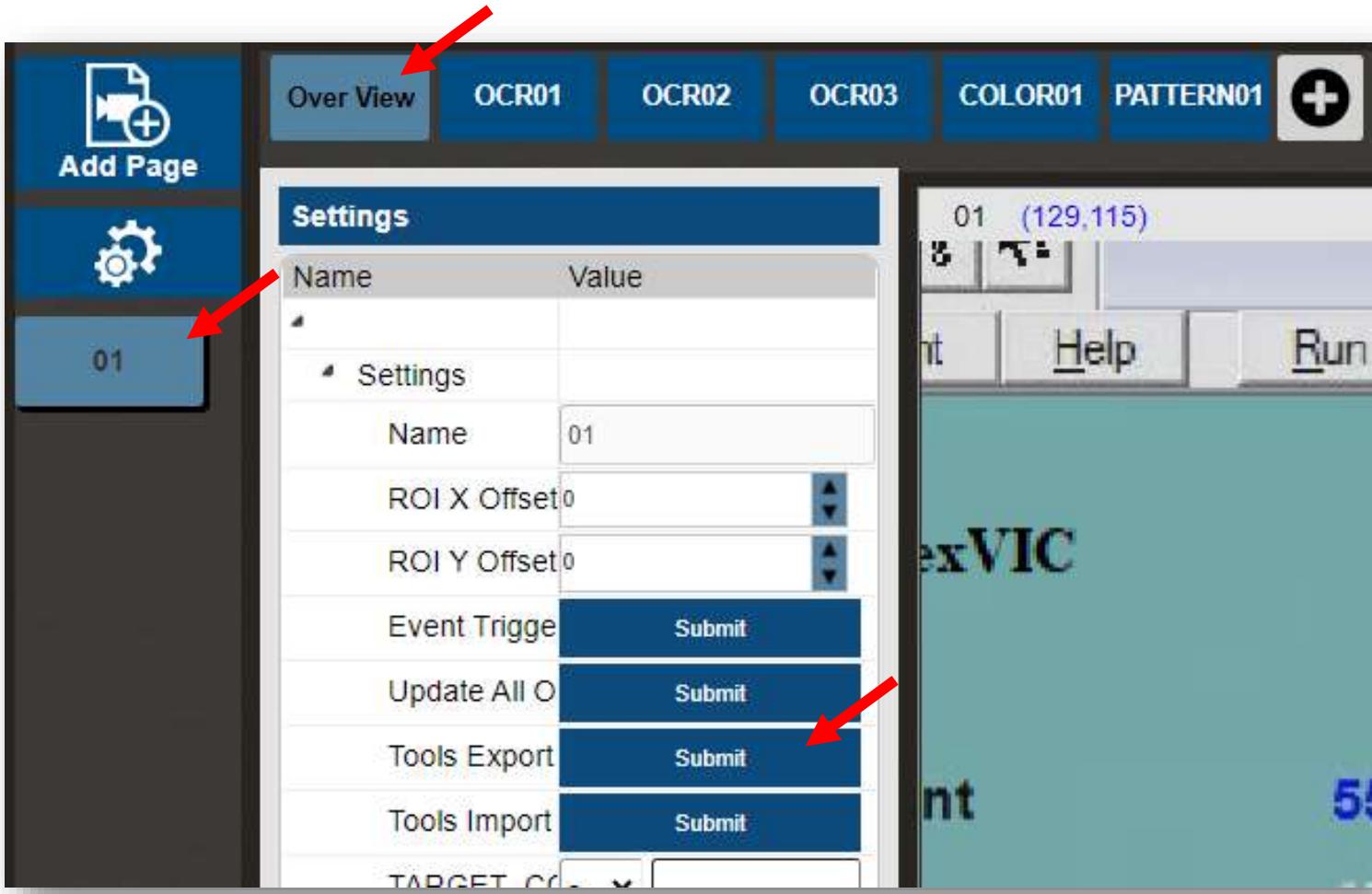
## • 一般：新增樣式

- **多個樣式**：在同一頻道和頁面上使用新增按鈕 → 點擊 Pattern → 依照前述步驟可新增更多樣式



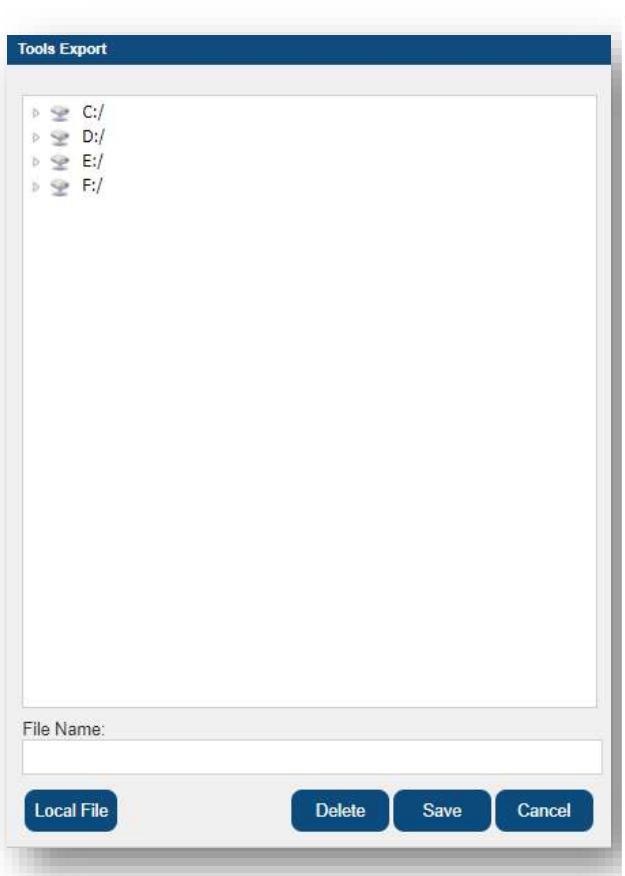
- 工具匯出：匯出辨識工具設定

點擊工具匯出按鈕 → 選擇儲存類型 → 點擊確定



## • 工具匯出：匯出辨識工具設定

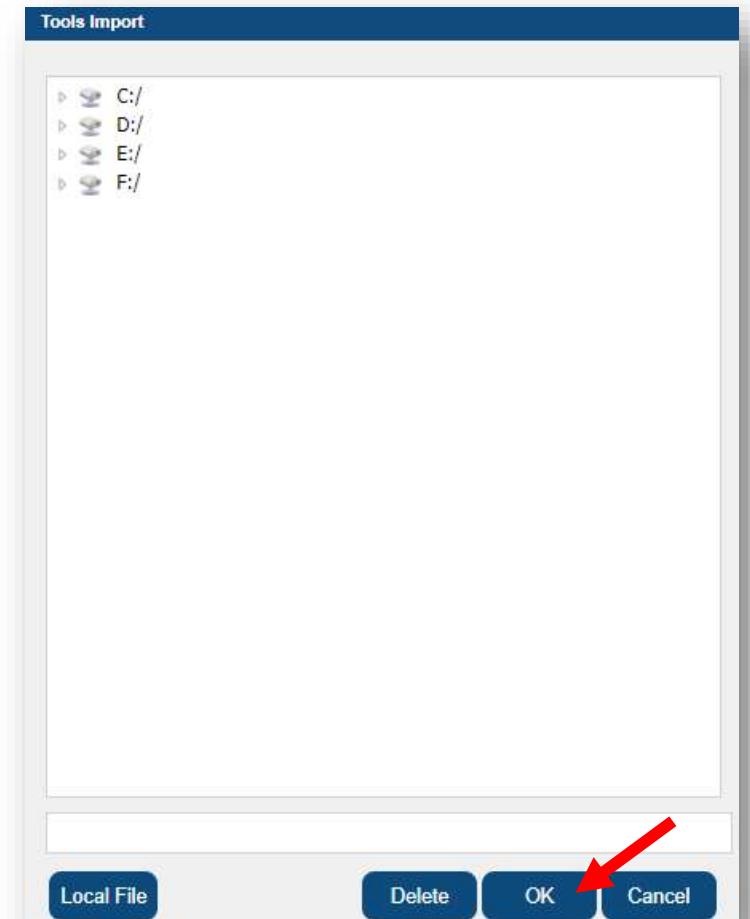
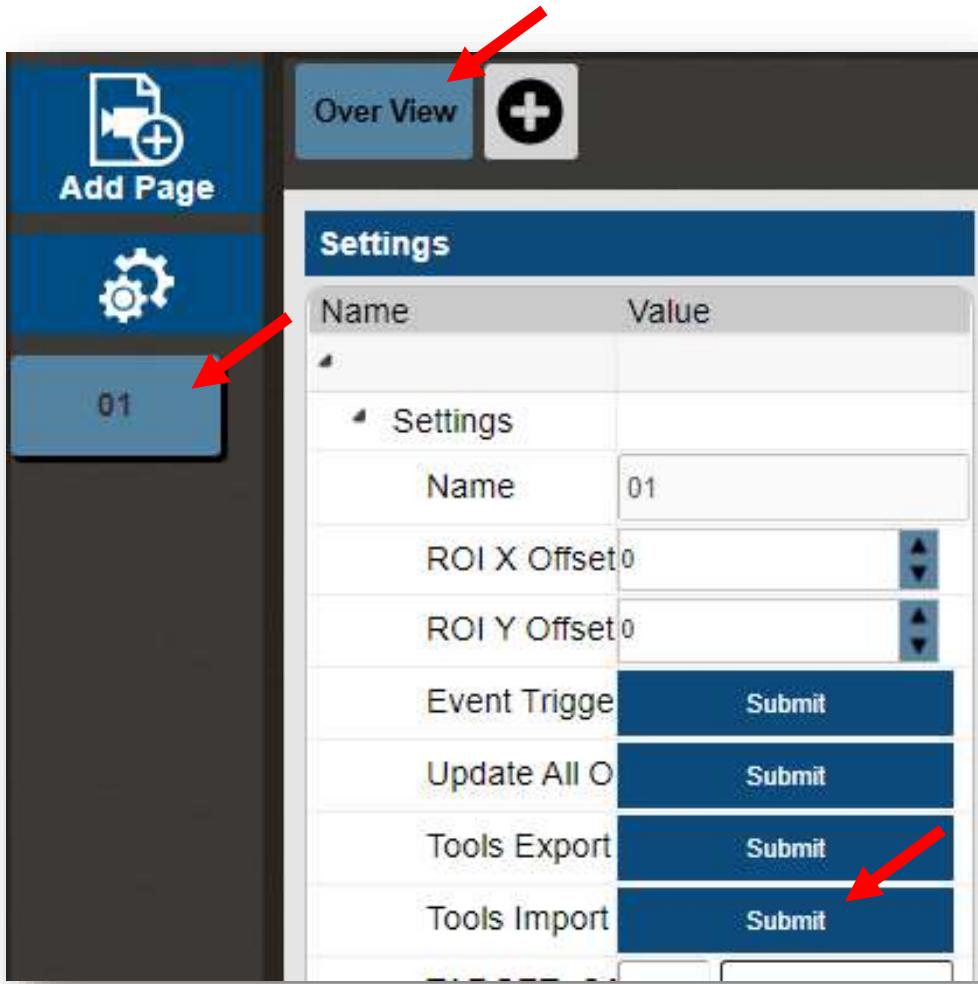
選擇儲存位址，匯出檔案的副檔名是 tools，內容為 json 字串



```
{
  "tools": [
    {
      "roi": true,
      "roi_x": 228,
      "roi_y": 419,
      "roi_width": 31,
      "roi_height": 8,
      "type": "COLOR",
      "id": "COLOR01",
      "red": 20,
      "green": 232,
      "blue": 64,
      "tolerance": 5,
      "matching_output": "true",
      "no_matching_output": "false"
    },
    {
      "roi": true,
      "roi_x": 203,
      "roi_y": 265,
      "roi_width": 41,
      "roi_height": 18,
      "type": "OCR",
      "id": "OCR01",
      "segmentation": 7,
      "preprocess_resize": 2,
      "preprocess_resize_method": 0,
      "preprocess_threshold_method": 1,
      "preprocess_threshold_algorithm": 0,
      "preprocess_threshold_value": 100,
      "allow_recognition_rate": 80,
      "allow_empty_string": 0,
      "allow_remove_whitespace": 0,
      "font_select": 0,
      "font_file": "VIC7000_Demo_blue_value.traineddata"
    },
    {
      "roi": true,
      "roi_x": 202,
      "roi_y": 283,
      "roi_width": 42,
      "roi_height": 14,
      "type": "OCR",
      "id": "OCR02",
      "segmentation": 7,
      "preprocess_resize": 2,
      "preprocess_resize_method": 0,
      "preprocess_threshold_method": 1,
      "preprocess_threshold_algorithm": 0,
      "preprocess_threshold_value": 95,
      "allow_recognition_rate": 80,
      "allow_empty_string": 0,
      "allow_remove_whitespace": 0,
      "font_select": 0,
      "font_file": "VIC7000_Demo_green_value.traineddata"
    },
    {
      "roi": true,
      "roi_x": 356,
      "roi_y": 98,
      "roi_width": 301,
      "roi_height": 124,
      "type": "PATTERN",
      "id": "PATTERN01",
      "gray_match": 0,
      "pattern_name": "",
      "pattern_x": 0,
      "pattern_y": 0,
      "pattern_width": 0,
      "pattern_height": 0,
      "min_score": 0.95,
      "detect_page_displacement": 0,
      "matching_output": "true",
      "no_matching_output": "false",
      "pattern_img": ""
    },
    {
      "roi": true,
      "roi_x": 202,
      "roi_y": 298,
      "roi_width": 42,
      "roi_height": 15,
      "type": "OCR",
      "id": "OCR03",
      "segmentation": 7,
      "preprocess_resize": 2,
      "preprocess_resize_method": 0,
      "preprocess_threshold_method": 1,
      "preprocess_threshold_algorithm": 0,
      "preprocess_threshold_value": 100,
      "allow_recognition_rate": 80,
      "allow_empty_string": 0,
      "allow_remove_whitespace": 0,
      "font_select": 0,
      "font_file": "VIC7000_Demo_black_value.traineddata"
    }
  ]
}
```

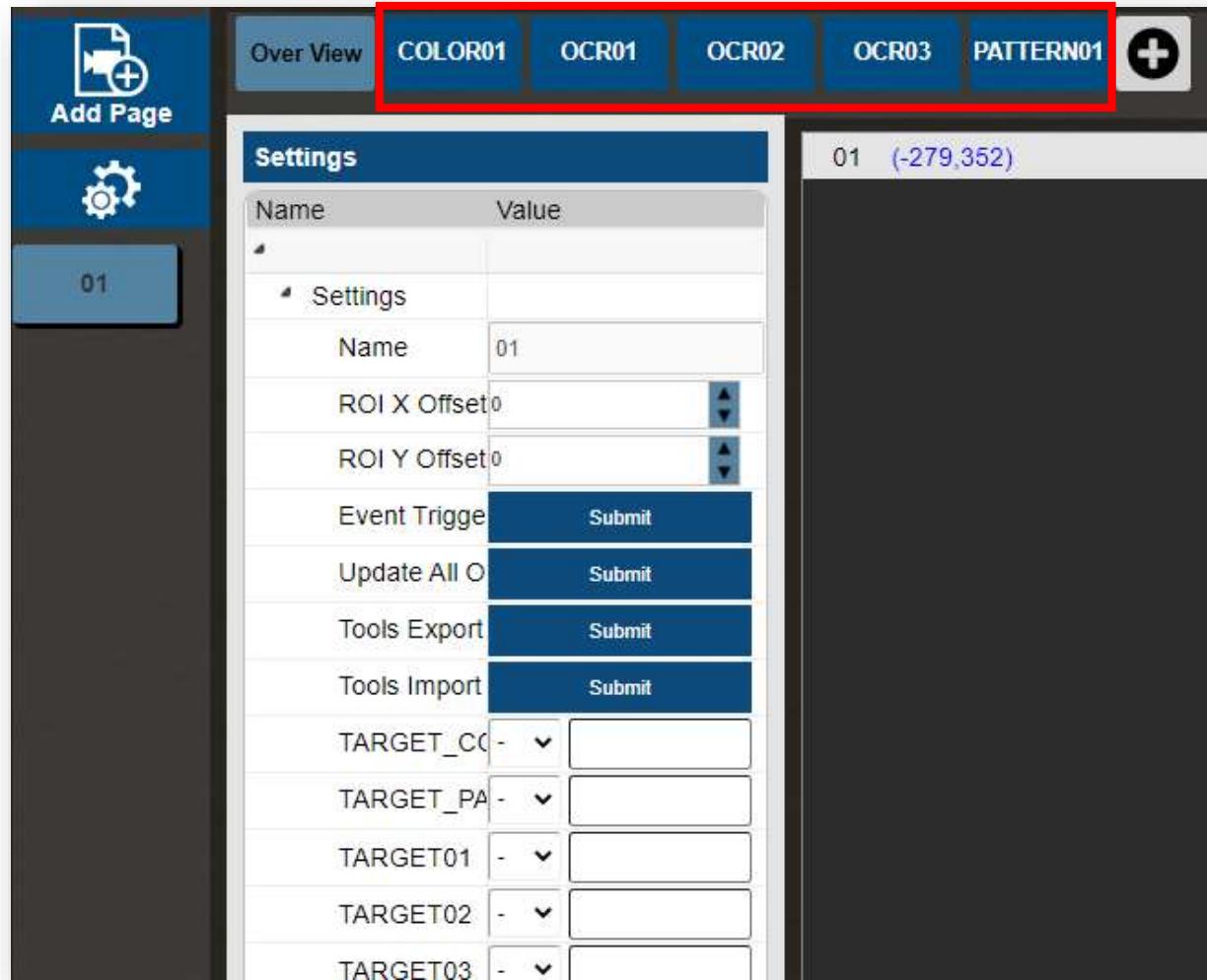
- 工具匯入：匯入辨識工具設定

點擊工具匯入按鈕 → 選擇編輯好的 tools 檔案 → 點擊確定



- 工具匯入：匯入辨識工具設定

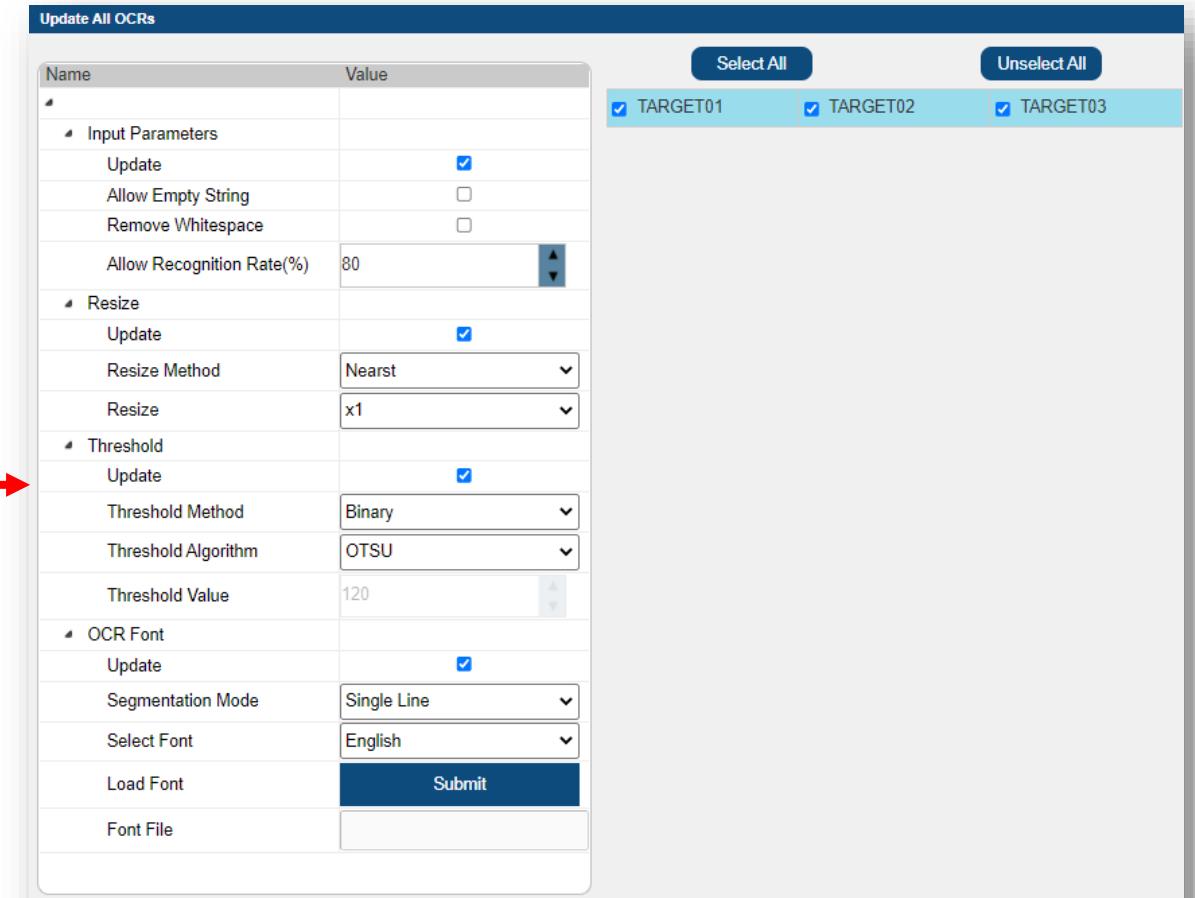
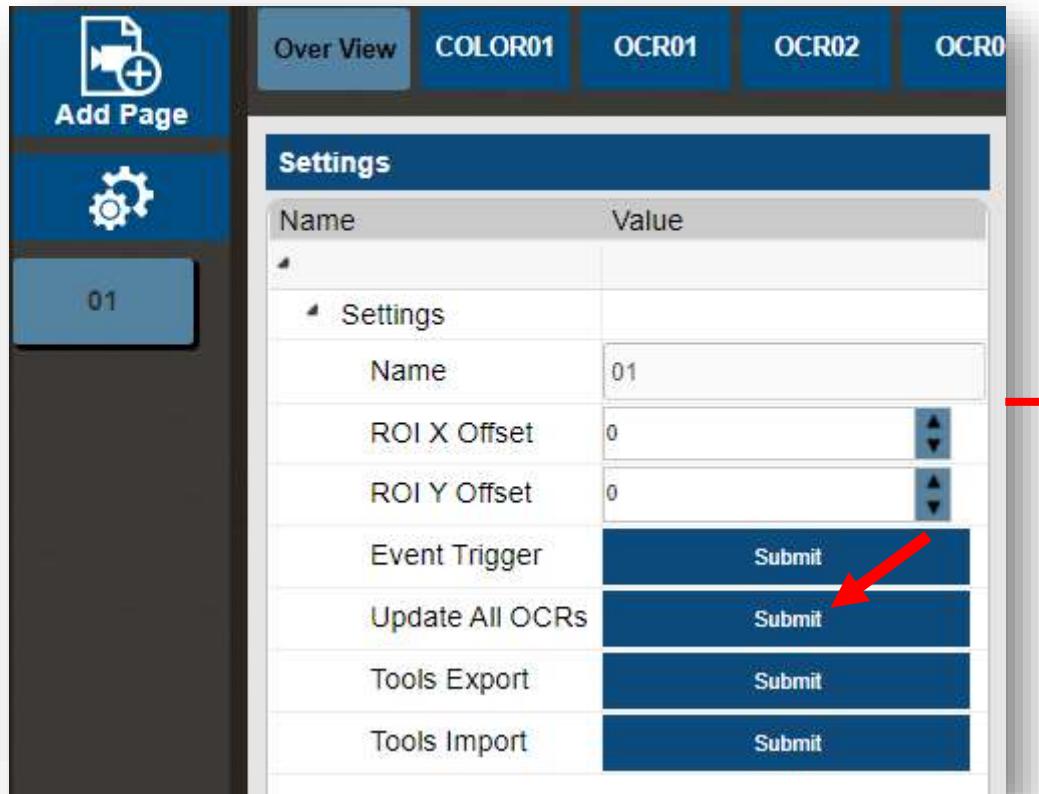
系統會依照匯入的 tools 檔案自動建立辨識工具



# 辨識工具

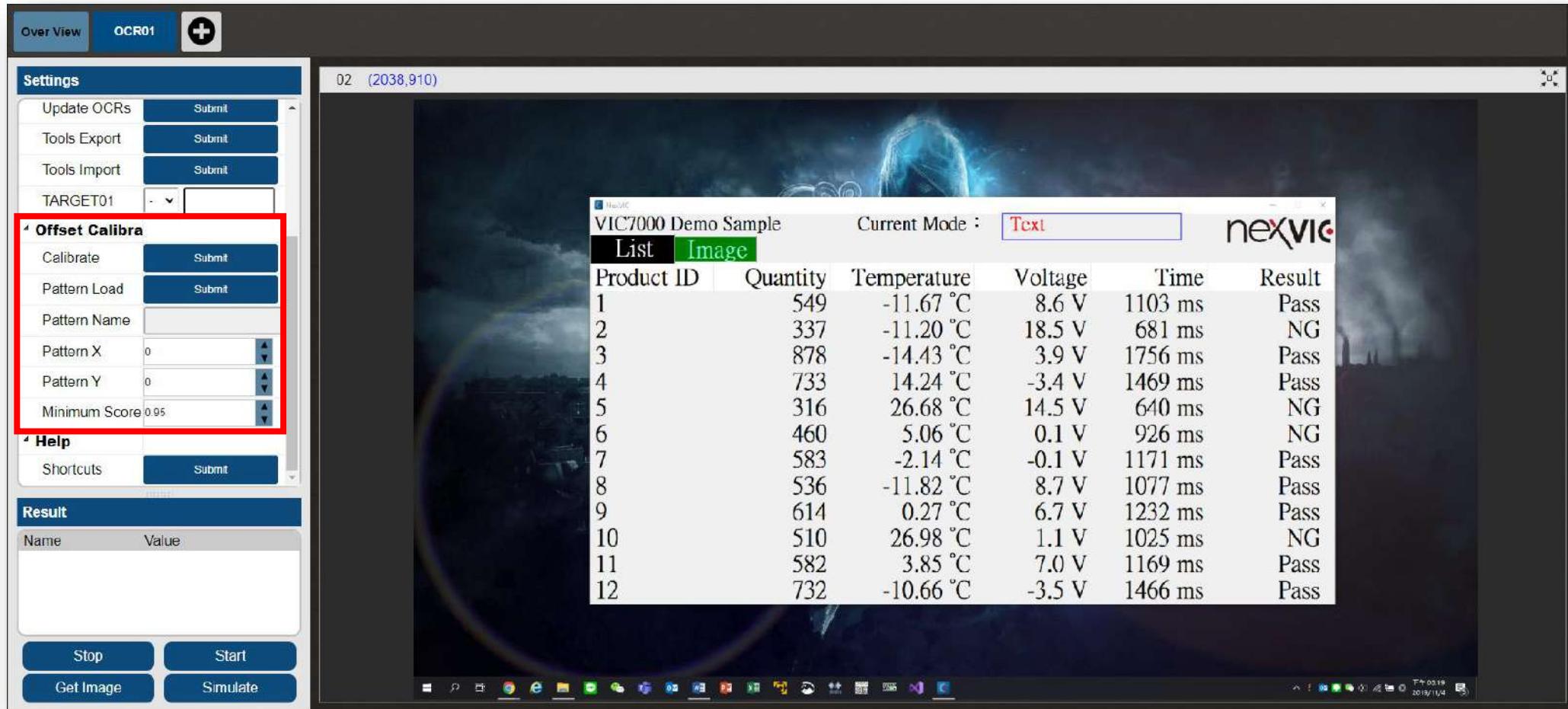
- 更新全部OCR

點擊更新全部OCR，可以一次更新全部的TARGET設定



## • 位移校正

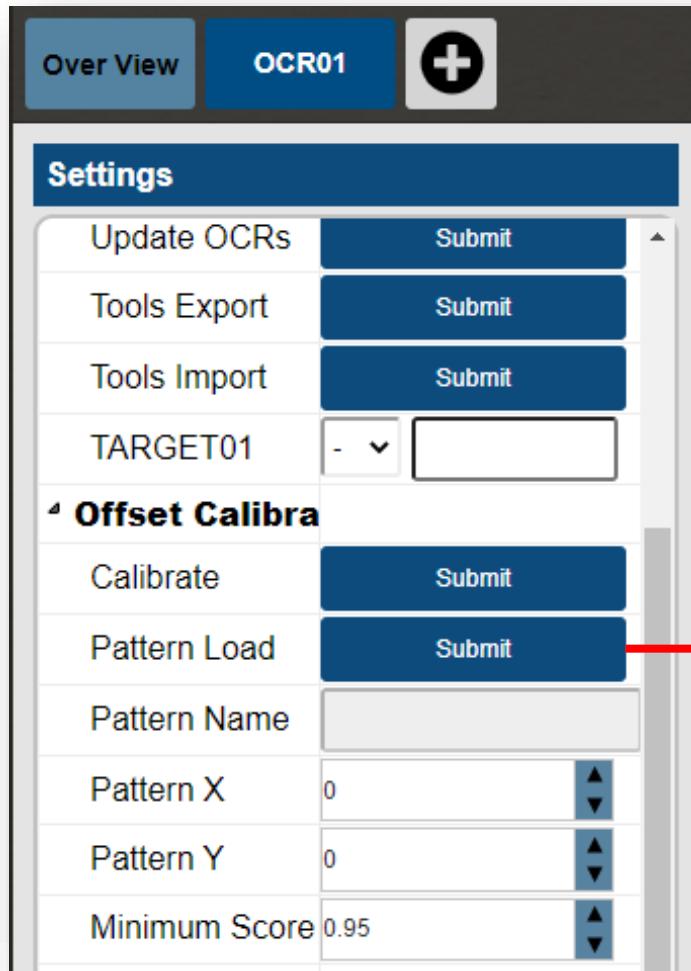
設置樣式比對標準，往後若載入辨識區域有位移的圖像，可使用位移校正來偵測ROI X位移和ROI Y位移，即可使用有位移的圖像進行辨識設定



# 辨識工具

## • 位移校正

點擊樣式載入，載入欲比對的樣式



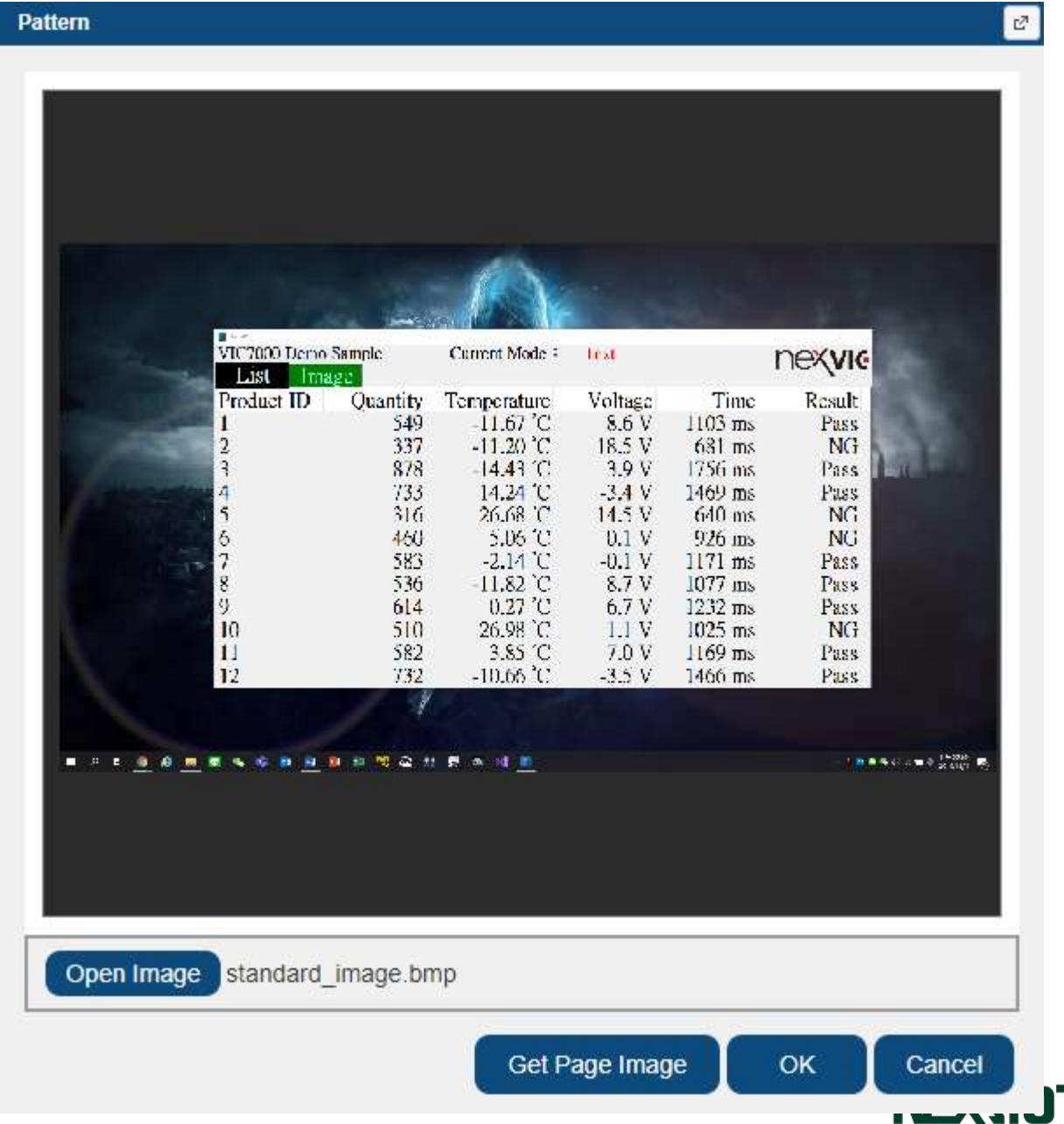
Over View    OCR01    

**Settings**

- Update OCRs    
- Tools Export    
- Tools Import    
- TARGET01    

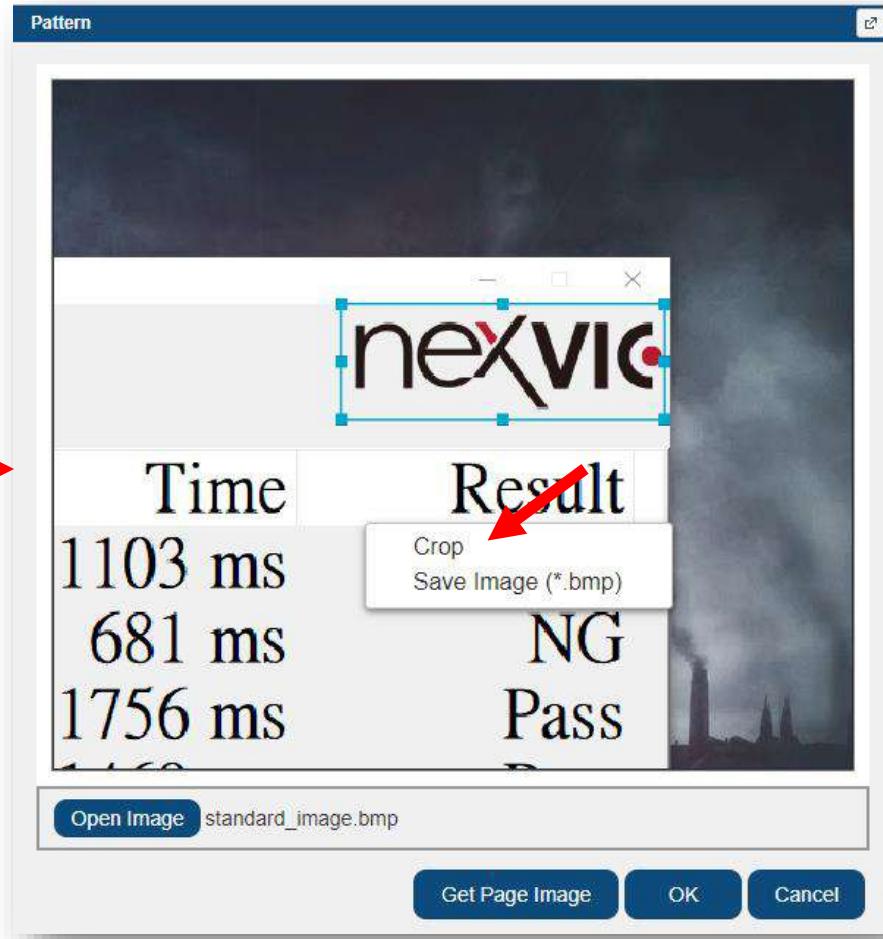
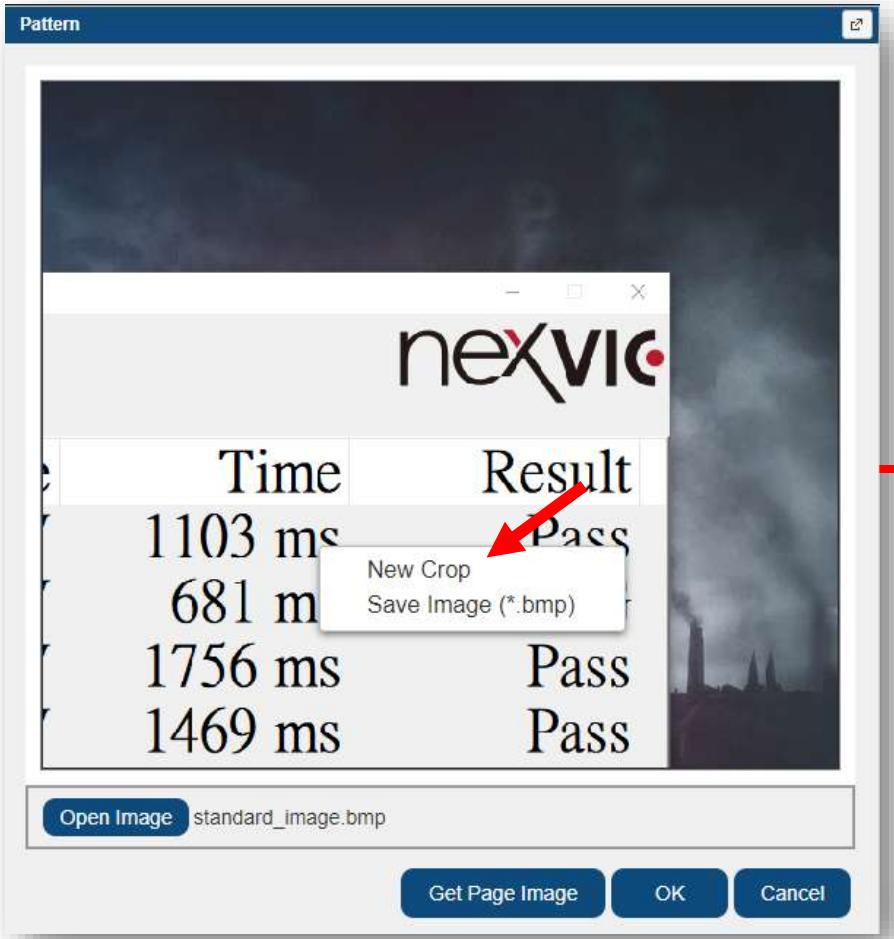
**Offset Calibration**

- Calibrate    
- Pattern Load    
- Pattern Name
- Pattern X
- Pattern Y
- Minimum Score



- 位移校正

點擊右鍵並選擇建立裁切，選擇要使用的樣式後，點擊右鍵並選擇裁切



- 位移校正

確認比對樣式後，點擊確定，系統同時會載入樣式 X 和樣式 Y，即為比對樣式的座標

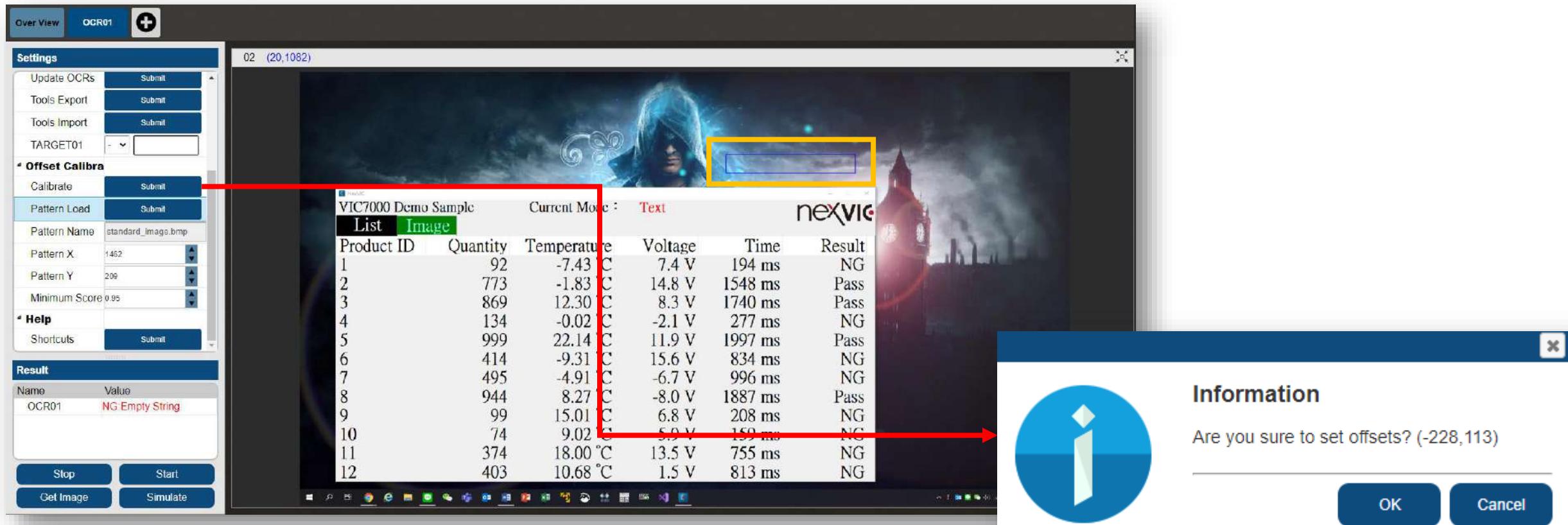


**Offset Calibra**

Calibrate	Submit
Pattern Load	Submit
Pattern Name	standard_image.bmp
Pattern X	1458
Pattern Y	204
Minimum Score	0.95

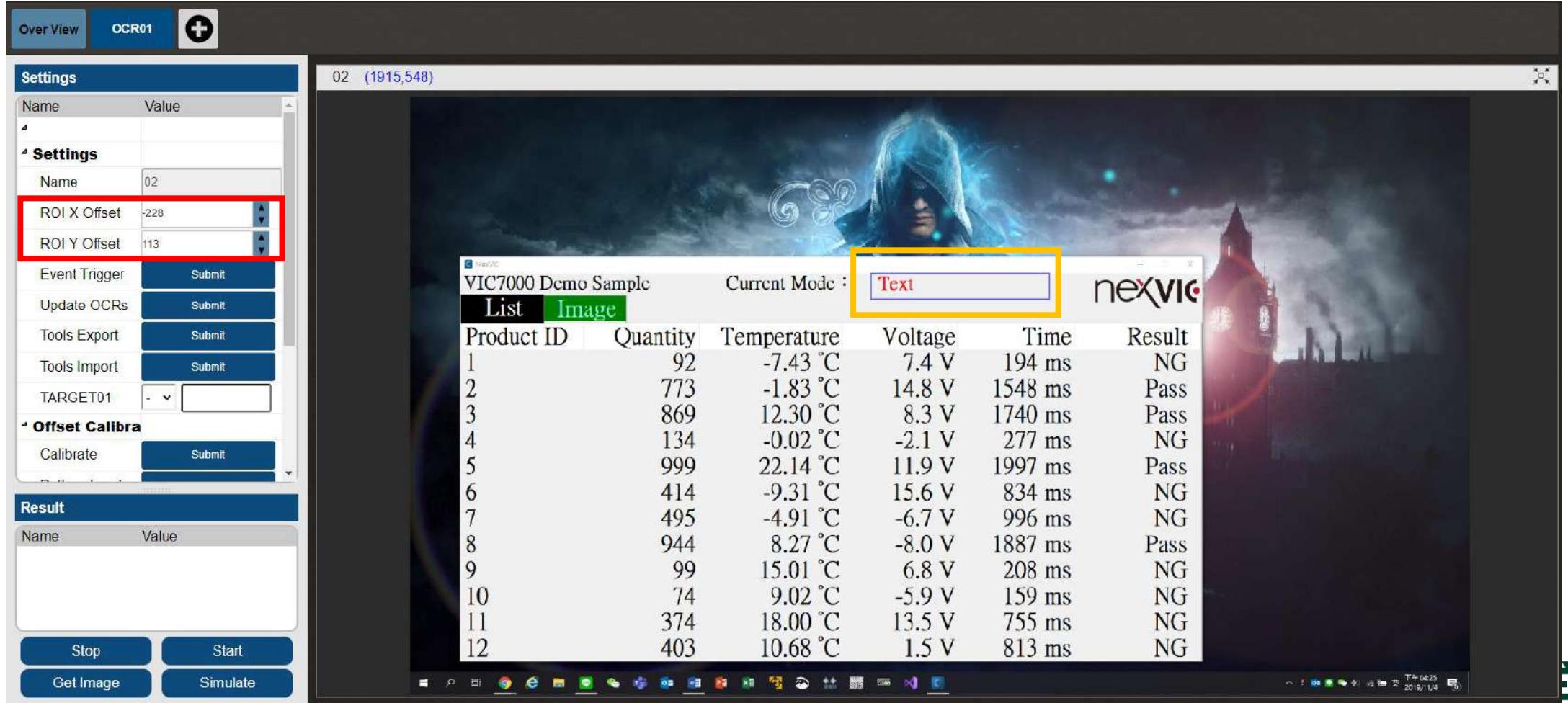
## • 位移校正

載入有辨識區域位移的圖像，點擊校正，系統會自動計算位移量，點擊確定



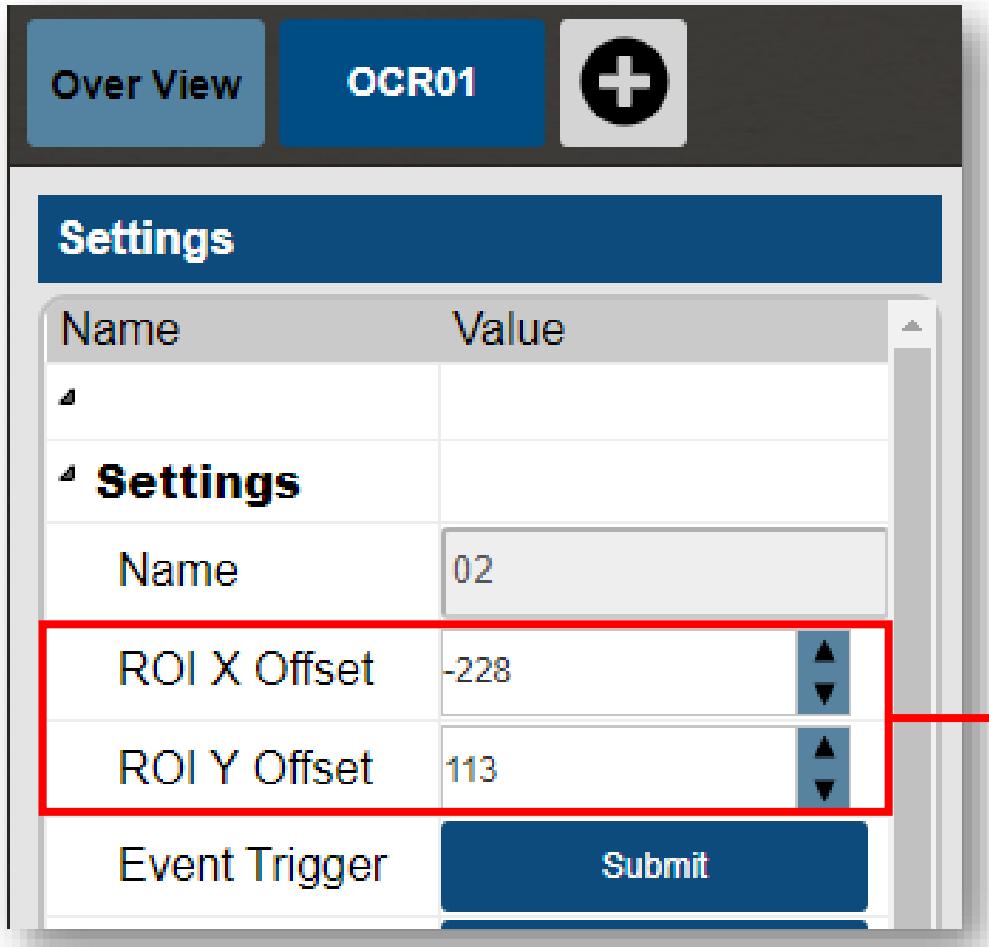
## • 位移校正

點擊確定後，系統會將位移量帶入ROI X位移和ROI Y位移，即可用有位移的圖像進行辨識設定



- 位移校正

辨識設定結束後，建議將ROI X位移和ROI Y位移復歸為0，避免實際辨識時錯誤

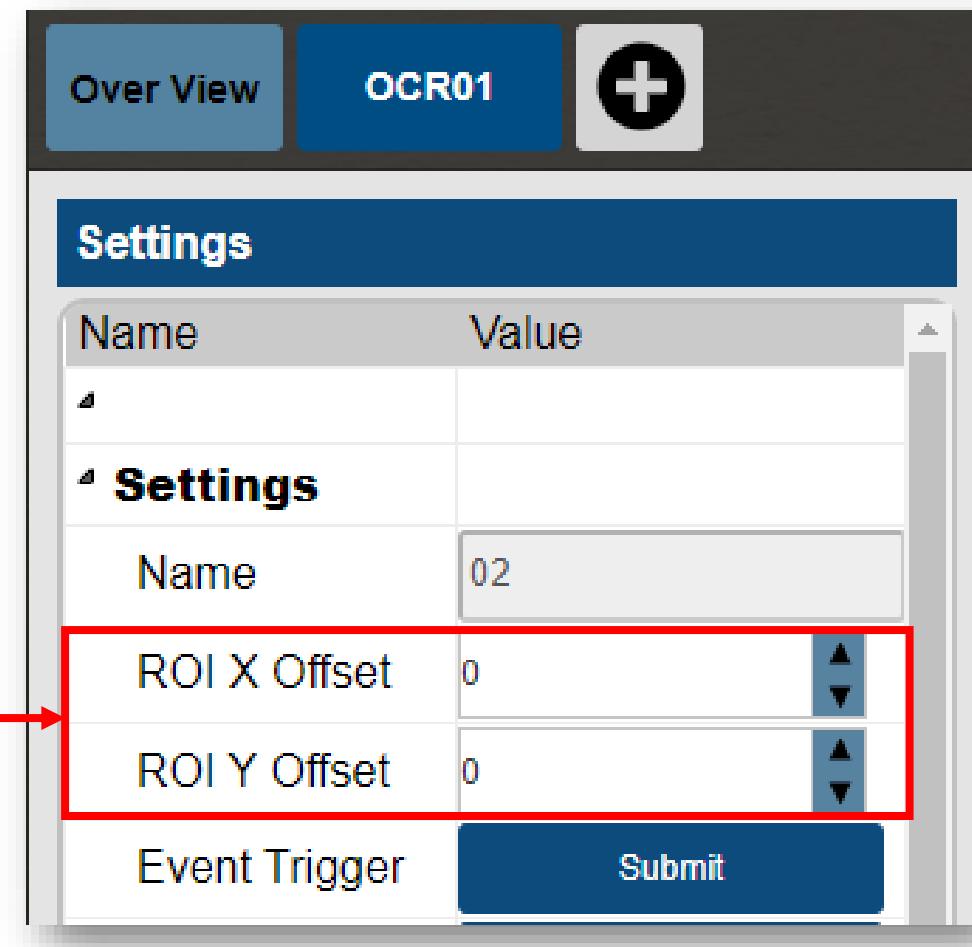


Over View    OCR01    +

### Settings

Name	Value
Settings	02
ROI X Offset	-228
ROI Y Offset	113

Event Trigger    Submit



Over View    OCR01    +

### Settings

Name	Value
Settings	02
ROI X Offset	0
ROI Y Offset	0

Event Trigger    Submit

# Outline

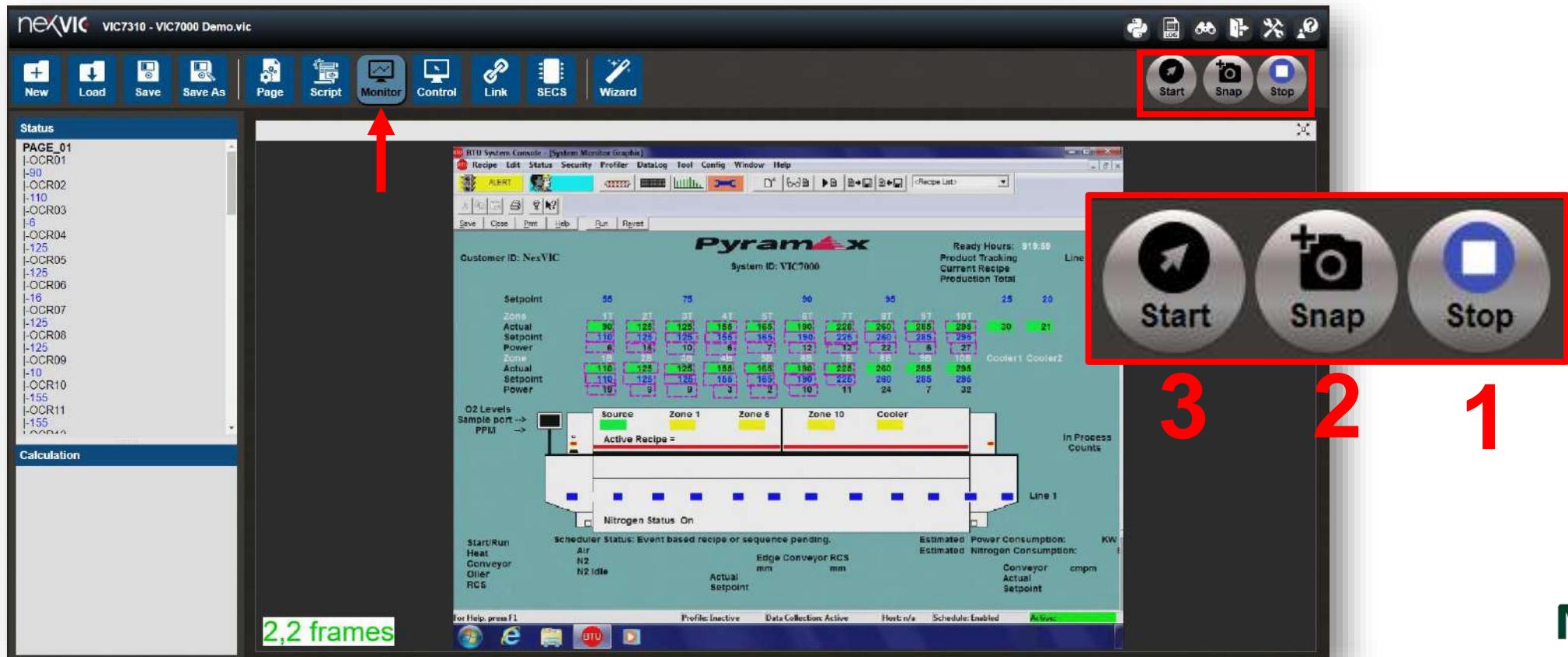
- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

## ● 監視頁面

1. 開始/停止專案：持續取得圖像並進行辨識
2. 觸發：取得一張圖像並進行辨識
3. 開始/停止控制：運行設定的控制檔

辨識結果**不寫入資料庫和外部連線**

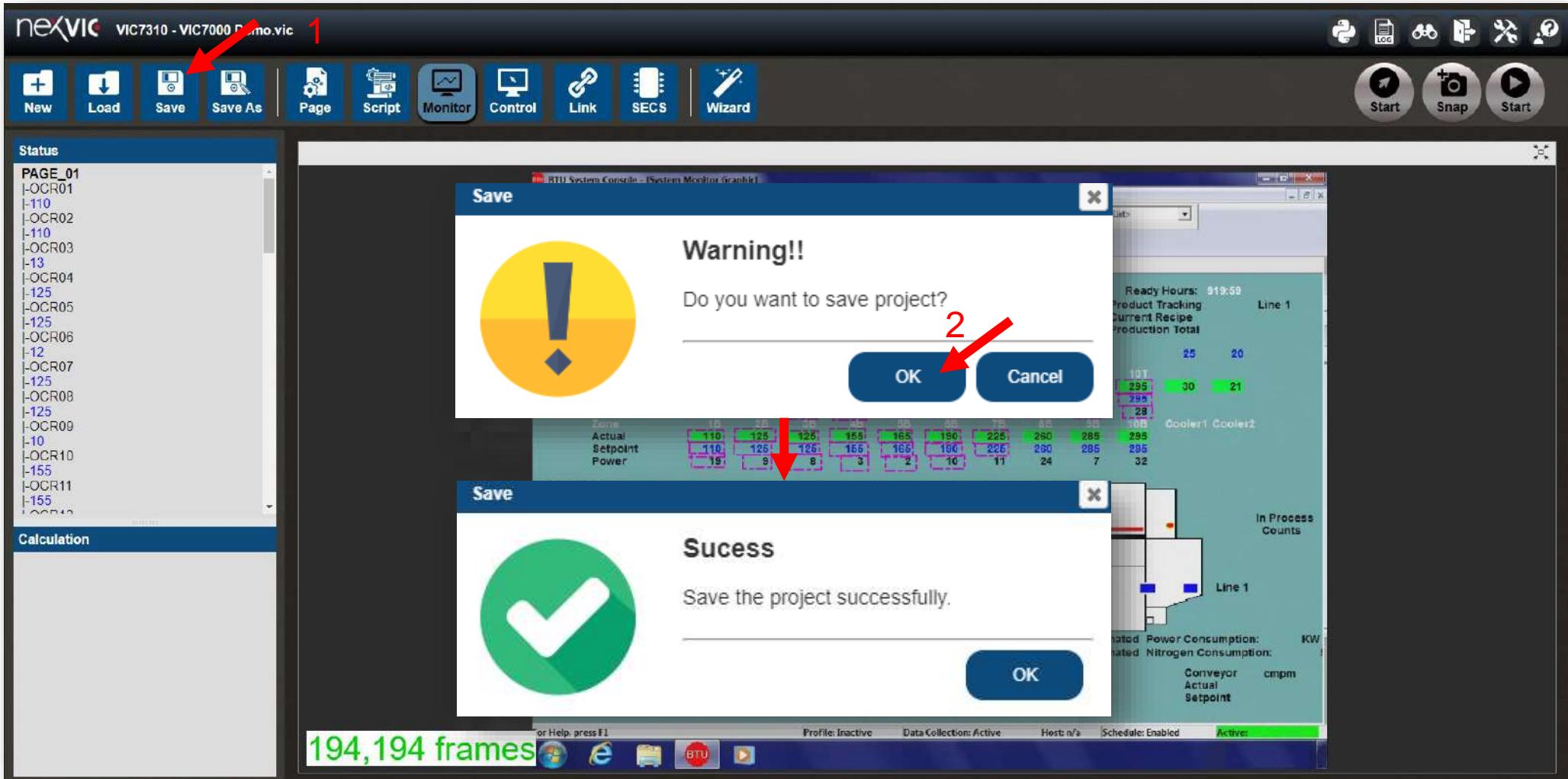
但**會執行事件觸發和腳本指令**



# 執行專案

- 儲存專案

1. 儲存專案：點擊儲存按鈕

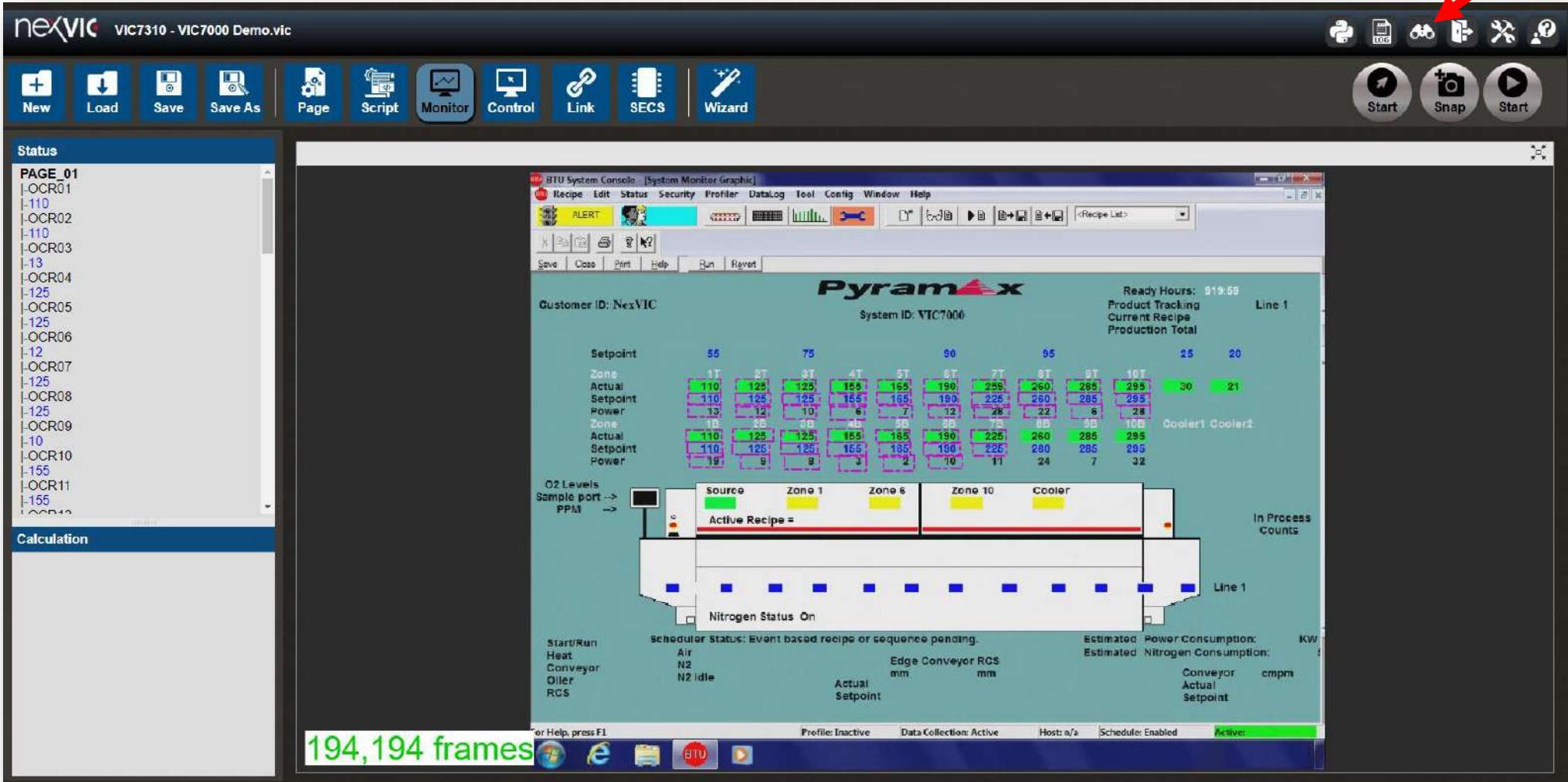


2. 點擊OK

# 執行專案

- 啟動專案

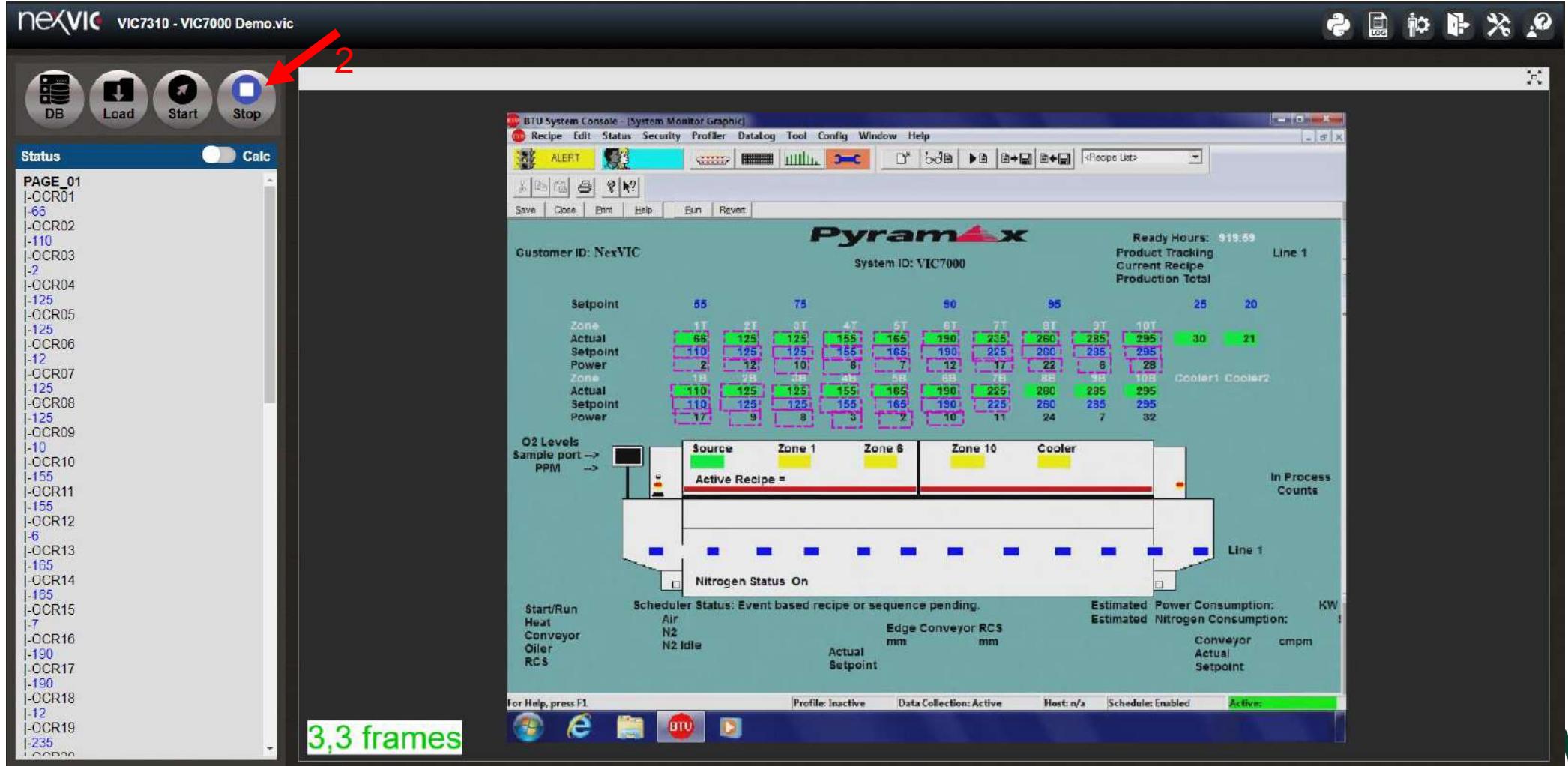
- 回到一般模式



# 執行專案

- 啟動專案

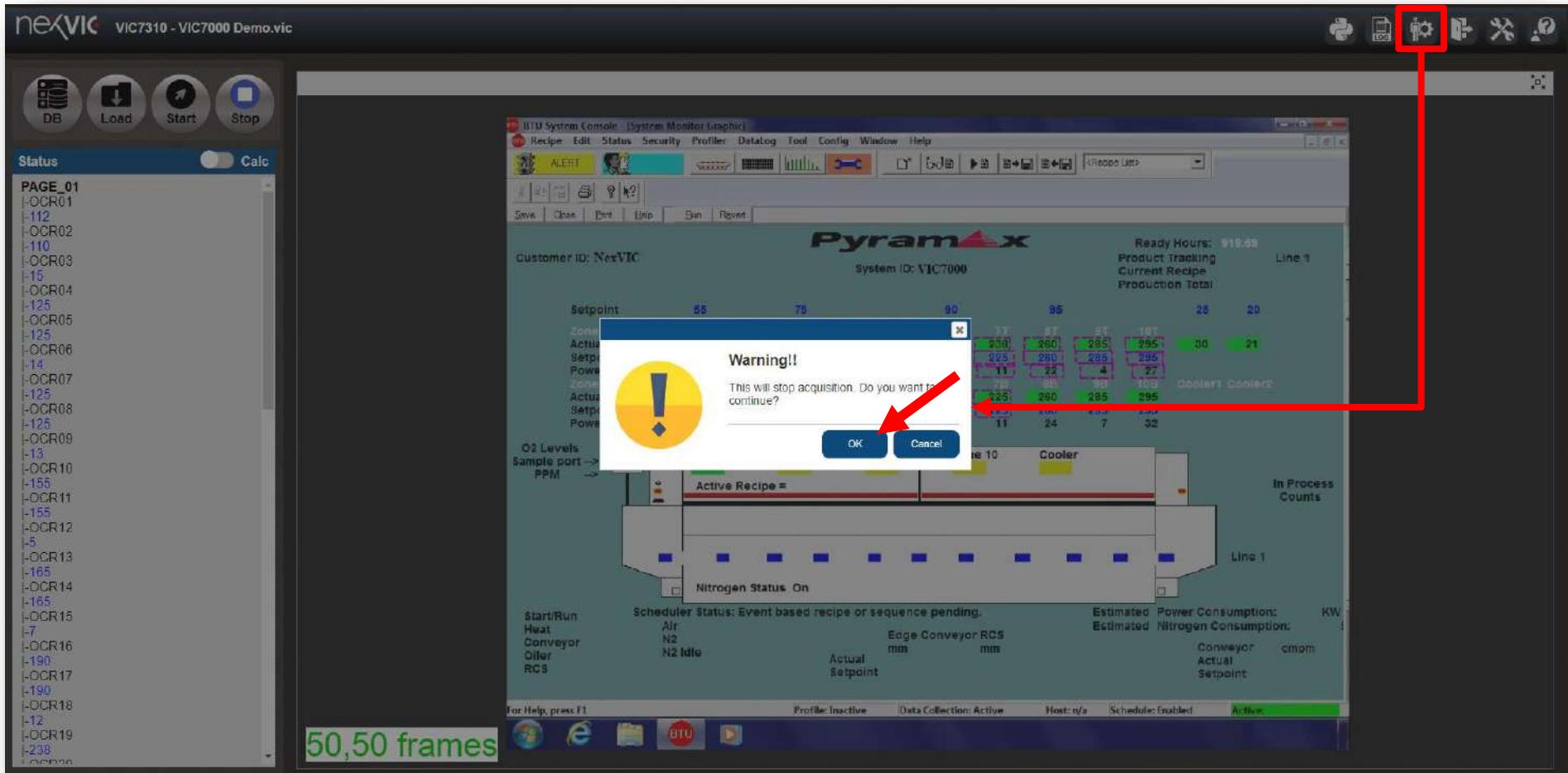
- 點擊開始專案



## • 停止提醒

在專案取像時，若要進入管理模式，需要停止專案。

點擊管理模式 → 停止專案提示視窗出現 → 點擊確認      即可進入管理模式



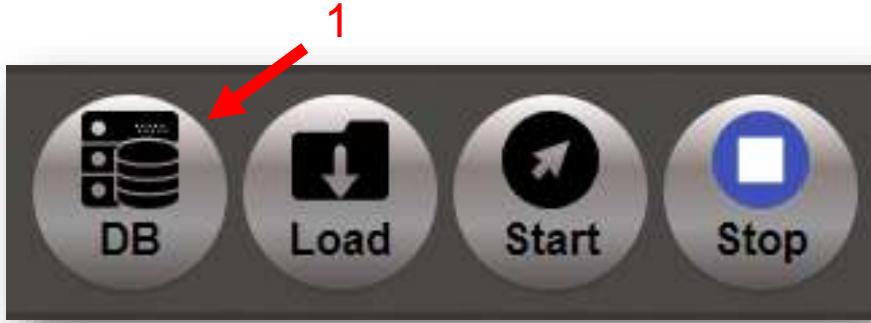
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# 資料庫

- 開啟資料庫

- 開啟資料庫：點擊資料庫按鈕



nexVIC VIC7310:Database

File CSV export XLS export

**Form**

**Target**  
Quantity: 5

**Target\_Color**  
Quantity: 5

**Target\_Pattern**  
Quantity: 5

**OCR**  
Quantity: 20

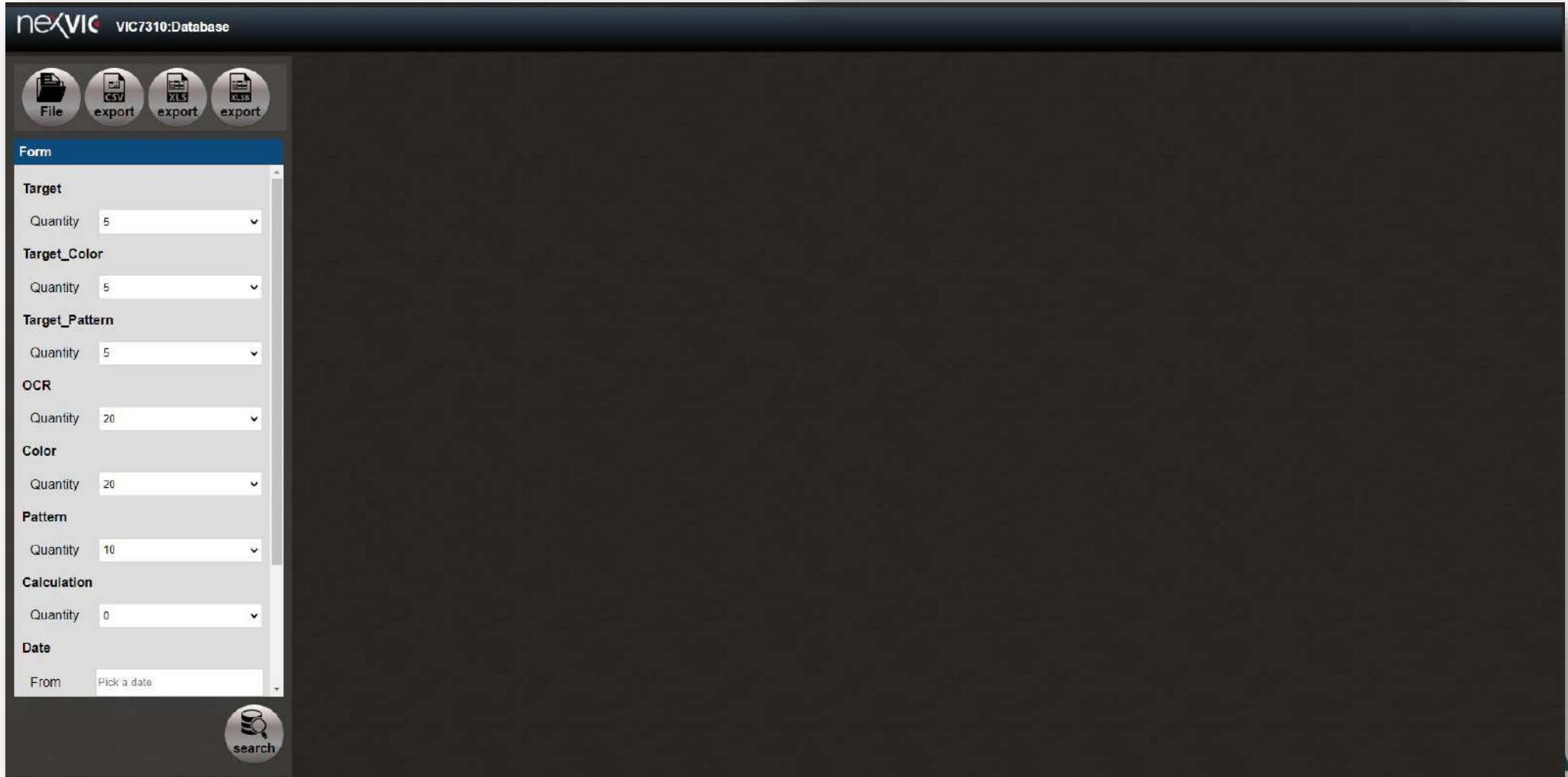
**Color**  
Quantity: 20

**Pattern**  
Quantity: 10

**Calculation**  
Quantity: 0

**Date**  
From: Pick a date

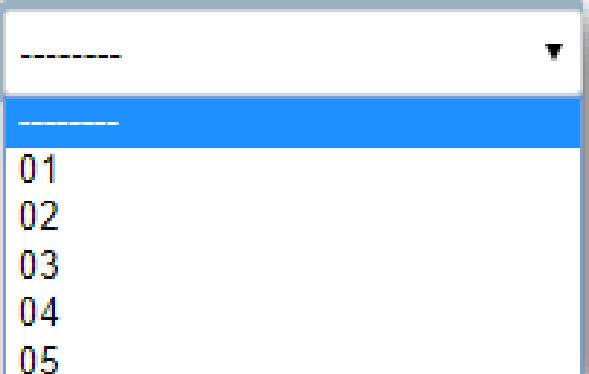
search



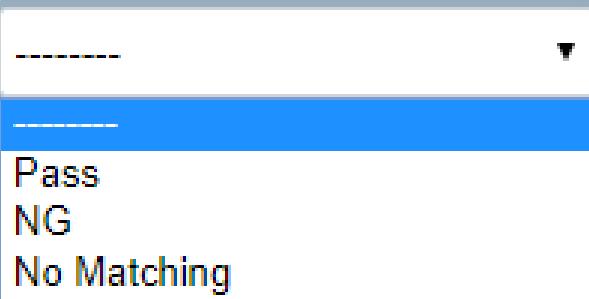
- 開啟資料庫

- 搜尋參數設定：要搜尋的目標值數量、顏色目標值數量、樣式比對目標值數量、OCR 數量、顏色數量、樣式比對數量、日期範圍、特定頻道、特定結果
- 開始搜尋：點擊搜尋按鈕

Page :



Result :



Date

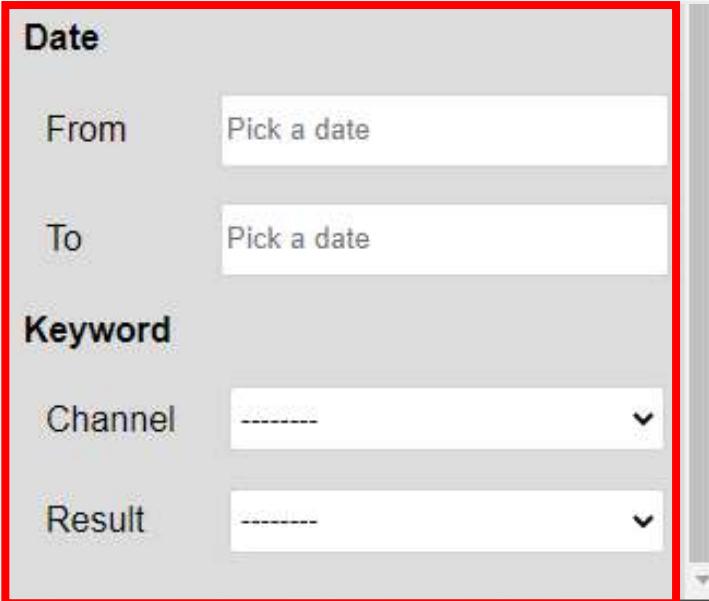
From

To

Keyword

Channel

Result



Form

Target

Quantity

Target\_Color

Quantity

Target\_Pattern

Quantity

OCR

Quantity

Color

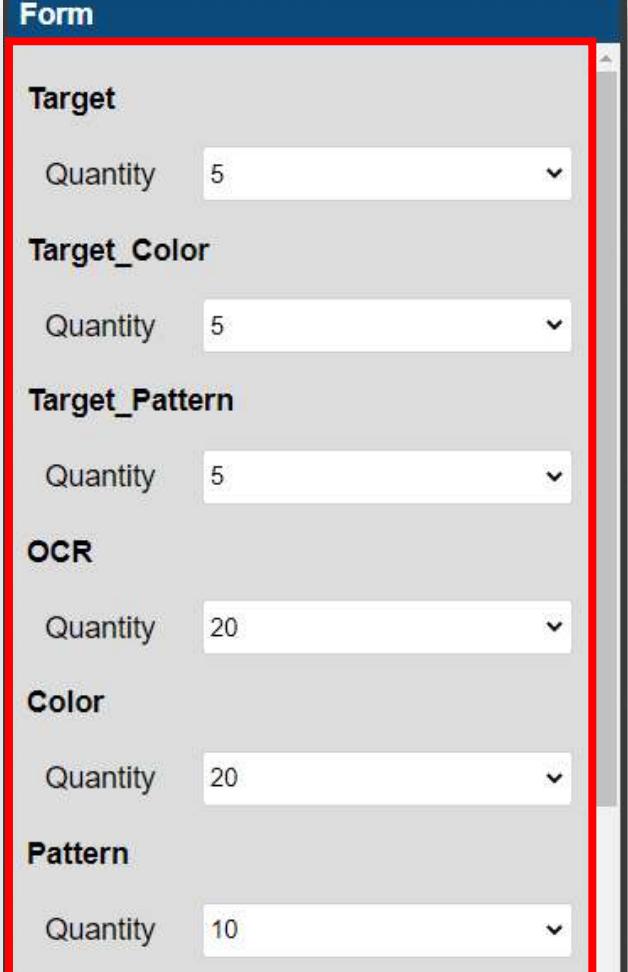
Quantity

Pattern

Quantity

Date

From



- 開啟資料庫

## 4. 搜尋結果：依照參數搜尋資料，一個小分頁有 20 筆資料，一個大分頁有 10 個小分頁

**nexVIC VIC7310.Database**

**Form**

**Target**  
Quantity: 0

**Target\_Color**  
Quantity: 0

**Target\_Pattern**  
Quantity: 0

**OCR**  
Quantity: 20

**Color**  
Quantity: 0

**Pattern**  
Quantity: 0

**Calculation**  
Quantity: 0

**Date**  
From: 2022-07-29 11:00

**File** **export** **export** **export**

**Page** **Result** **TimeStamp** **Image** **OCR\_01** **OCR\_02** **OCR\_03** **OCR\_04** **OCR\_05** **OCR\_06** **OCR\_07** **OCR\_08** **OC**

24850303	01	Pass	2022-07-29 11:40:40.329		110	110	13	125	125	15	125	125	10
24850304	01	Pass	2022-07-29 11:48:41.328		90	110	6	125	125	16	125	125	10
24850305	01	Pass	2022-07-29 11:48:42.312		66	110	2	125	125	12	125	125	10
24850306	01	Pass	2022-07-29 11:52:53.443		110	110	13	125	125	15	125	125	10
24850307	01	Pass	2022-07-29 11:52:54.440		90	110	6	125	125	16	125	125	10
24850308	01	Pass	2022-07-29 11:52:55.439		66	110	2	125	125	12	125	125	10
24850309	01	Pass	2022-07-29 11:52:56.441		110	110	13	125	125	12	125	125	10
24850310	01	Pass	2022-07-29 11:52:57.438		155	110	23	125	125	15	125	125	10
24850311	01	Pass	2022-07-29 11:52:58.439		195	110	26	125	125	15	125	125	10
24850312	01	Pass	2022-07-29 11:52:59.440		225	110	35	125	125	12	125	125	10
24850313	01	Pass	2022-07-29 11:53:00.439		185	110	32	125	125	14	125	125	10
24850314	01	Pass	2022-07-29 11:53:01.444		125	110	21	125	125	14	125	125	10
24850315	01	Pass	2022-07-29 11:53:02.438		112	110	15	125	125	14	125	125	13
24850316	01	Pass	2022-07-29 11:53:03.440		110	110	13	125	125	15	125	125	10
24850317	01	Pass	2022-07-29 11:53:04.447		90	110	6	125	125	16	125	125	10
24850318	01	Pass	2022-07-29 11:53:05.440		66	110	2	125	125	12	125	125	10
24850319	01	Pass	2022-07-29 11:53:06.441		110	110	13	125	125	12	125	125	10
24850320	01	Pass	2022-07-29 11:53:07.446		155	110	23	125	125	15	125	125	10
24850321	01	Pass	2022-07-29 11:53:08.438		195	110	26	125	125	15	125	125	10
24850322	01	Pass	2022-07-29 11:53:09.442		225	110	35	125	125	12	125	125	10

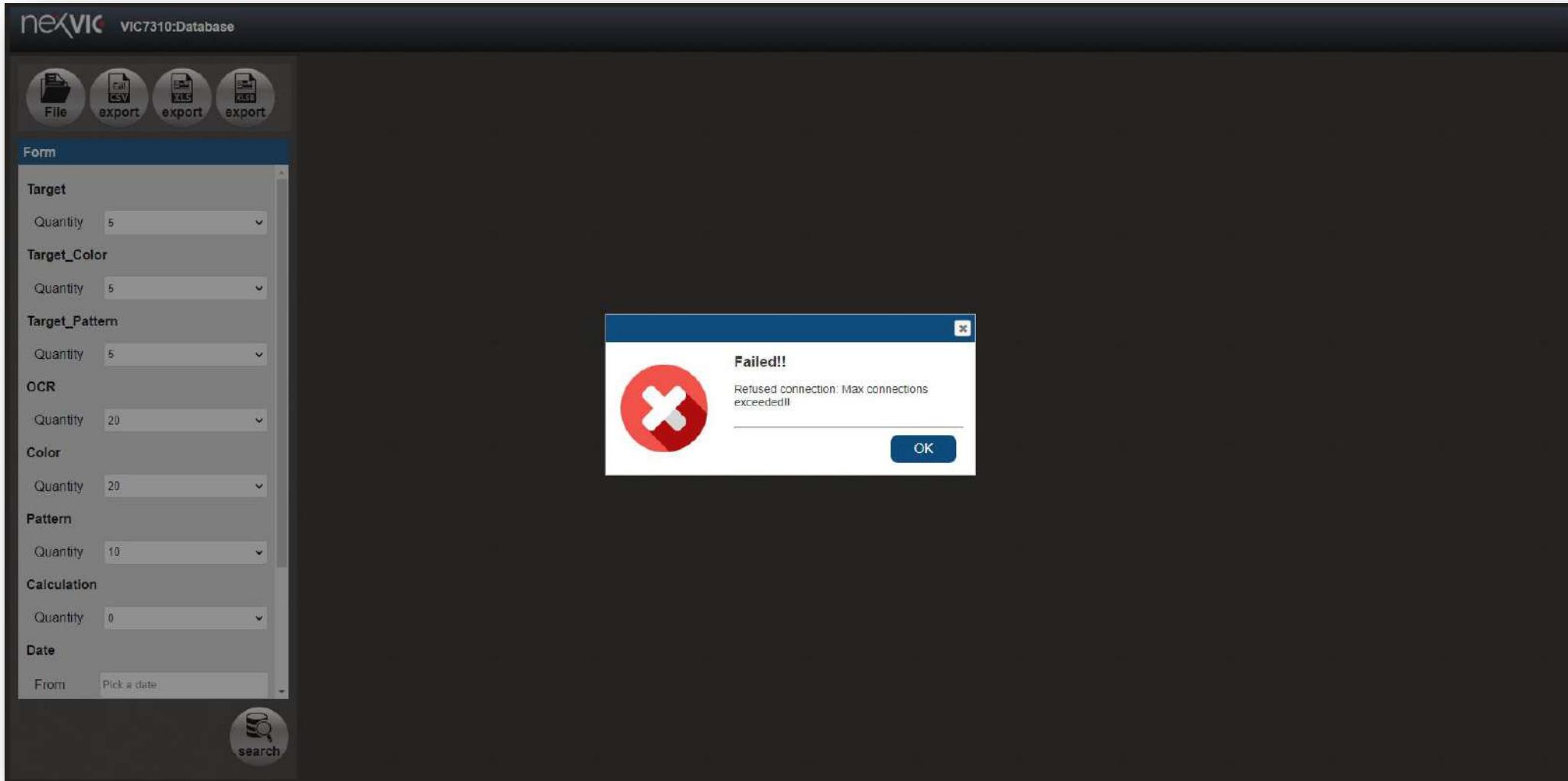
Page: **1** 2 3 4 5 6 7 8 9 10

page: first prev **1** 2 next last

**search**

- 人數限制

目前資料庫僅允許一人登入



## ● 儲存圖檔

Save Image

資料庫可儲存 NG 時的圖像，在設定視窗中啟用，儲存的圖像可用來再次學習

nexVIC VIC7310:Database

**File** **CSV export** **XLS export** **XLSB export**

**Form**

**Target**  
Quantity: 1

**Target\_Color**  
Quantity: 0

**Target\_Pattern**  
Quantity: 0

**OCR**  
Quantity: 20

**Color**  
Quantity: 20

**Pattern**  
Quantity: 10

**Calculation**  
Quantity: 0

**Date**  
From: 2021-09-16 16:00

page: first prev 1 next last

**ID** **Page** **Result** **TimeStamp** **Image** **OCR\_01** **OCR\_02** **OCR\_03** **OCR\_04** **OCR\_05** **OCR\_06** **OCR\_07** **OCR\_08** **OCR\_09**

30508	01	NG	2021-05-17 14:53:05.859		110	110				125	125	10	
30509	01	NG	2021-05-17 14:53:06.870		90	110				125	125	10	
30510	01	NG	2021-05-17 14:53:07.874		66	110				125	125	10	
30511	01	NG	2021-05-17 14:53:08.867		110	110	13	125	125	12	125	125	10
30512	01	NG	2021-05-17 14:53:09.867		155	110	23	125	125	15	125	125	10
30513	01	NG	2021-05-17 14:53:10.868		195	110	26	125	125	15	125	125	10
30514	01	NG	2021-05-17 14:53:24.196	image									10
30515	01	NG	2021-05-17 14:53:25.195	image									10
30516	01	NG	2021-05-17 14:53:26.192	image									10
30517	01	NG	2021-05-17 14:53:27.196	image									10
30518	01	NG	2021-05-17 14:53:28.208	image									10
30519	01	NG	2021-05-17 14:53:29.201	image									10
30520	01	NG	2021-05-17 14:53:30.196	image									10
30521	01	NG	2021-05-17 14:53:31.195	image									10
30522	01	NG	2021-05-17 14:53:32.195	image									10
30523	01	NG	2021-05-17 14:53:33.196	image									10
30524	01	NG	2021-05-17 14:53:34.194	image									10
30525	01	NG	2021-05-17 14:53:35.205	image									10
30526	01	NG	2021-05-17 14:53:36.192	image									10
30527	01	NG	2021-05-17 14:53:37.194	image									10

**image**

**RTU System Console - System Monitor Graphics**

**Pyramax**

Customer ID: NexVIC System ID: VIC7000 Ready Hours: 918:00 Line 1

Product Tracking Current Recipe Production Total

Setpoint: 69 79 90 36 29 20

Zone: 15 35 45 55 65 75 85 95 105 115

Actual: 110 125 125 155 165 180 225 260 265 265

Setpoint: 13 18 10 6 7 12 16 22 28 28

Power: 15 18 10 6 7 12 16 22 28 28

Actual: 110 125 125 155 165 180 225 260 265 265

Setpoint: 110 125 125 155 165 180 225 260 265 265

Power: 19 8 5 2 10 11 24 7 32

O2 Levels Sample port: PPM Active Recipe =

Source Zone 1 Zone 6 Zone 10 Cooler

Nitrogen Status: On

Scheduler Status: Event based recipe or sequence pending

Estimated Power Consumption: KW

Estimated Nitrogen Consumption: cmm

Start/Run Heat Conveyor Oiler RCS

Actual Setpoint Edge Conveyor RCS mm

Conveyor Actual Setpoint cmppm

For Help, press F1

Profile Inactive Data Collection: Active Host: n/a Schedule Enabled Active

## ● 儲存不重複資料

Skip Repeated Data

資料庫可忽略重複的資料，在設定視窗中啟用。在取像頻率快且出現重複資料時，可啟用此功能

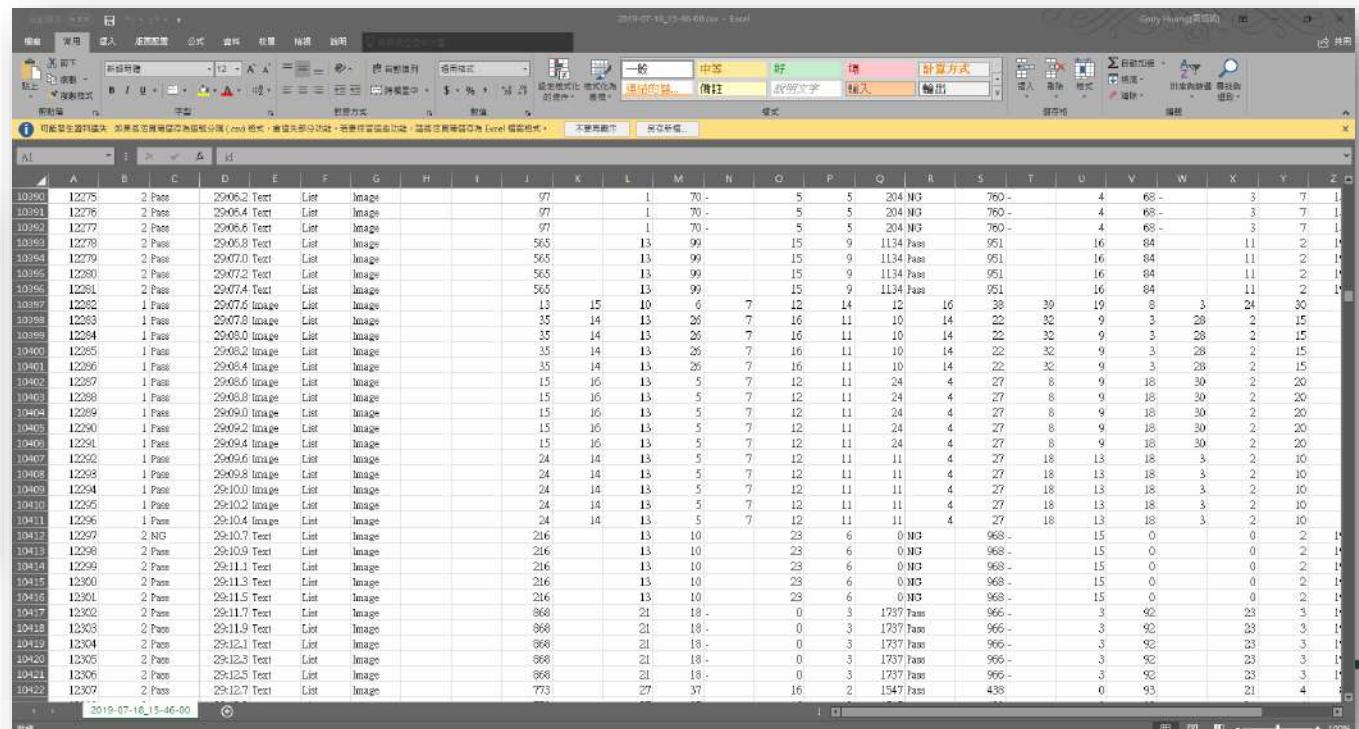
<b>id</b>	<b>page</b>	<b>result</b>	<b>timeStamp</b>	<b>image</b>	<b>OCR_01</b>	<b>OCR_02</b>	<b>OCR_03</b>	<b>OCR_04</b>
6378551	02	Pass	2020-03-06 15:42:14.651	0	0.078	-0.610	0.838	
6378552	02	Pass	2020-03-06 15:42:15.667	0	0.953	-0.627	-0.994	
6378553	02	Pass	2020-03-06 15:42:16.700	0	0.998	-0.444	-0.346	
6378554	02	Pass	2020-03-06 15:42:17.701	0	0.160	0.629	-0.534	
6378555	02	Pass	2020-03-06 15:42:18.717	0	-0.786	-0.969	-0.875	
6378556	02	Pass	2020-03-06 15:42:19.734	0	0.697	-0.633	0.416	
6378557	02	Pass	2020-03-06 15:42:20.751	0	-0.988	0.957	0.647	
6378558	02	Pass	2020-03-06 15:42:21.767	0	-0.143	-0.422	-0.758	
6378559	02	Pass	2020-03-06 15:42:22.784	0	0.844	-0.062	0.438	
6378560	02	Pass	2020-03-06 15:42:23.800	0	0.006	0.302	-0.515	
6378561	02	Pass	2020-03-06 15:42:24.818	0	0.081	-0.611	-0.305	
6378562	02	Pass	2020-03-06 15:42:25.833	0	-0.278	0.880	-0.560	
6378563	02	Pass	2020-03-06 15:42:26.650	0	0.350	-0.573	-0.502	
6378564	02	Pass	2020-03-06 15:42:27.650	0	-0.071	0.824	0.046	
6378565	02	Pass	2020-03-06 15:42:28.684	0	0.059	0.081	-0.616	
6378566	02	Pass	2020-03-06 15:42:29.700	0	-0.701	0.173	0.057	
6378567	02	Pass	2020-03-06 15:42:30.717	0	-0.773	0.112	-0.834	
6378568	02	Pass	2020-03-06 15:42:31.717	0	0.899	0.889	-0.259	
6378569	02	Pass	2020-03-06 15:42:32.733	0	-0.633	0.963	-0.386	
6378570	02	Pass	2020-03-06 15:42:33.750	0	-0.469	0.845	0.829	

## • 資料庫匯出和備份

VIC7000 內的資料庫可儲存七天內的資料

依照搜尋的結果，匯出 .csv 、 .xls 或 .xlsb ，匯出路徑為設定頁面中設定的路徑，匯出在 VIC7300 電腦上

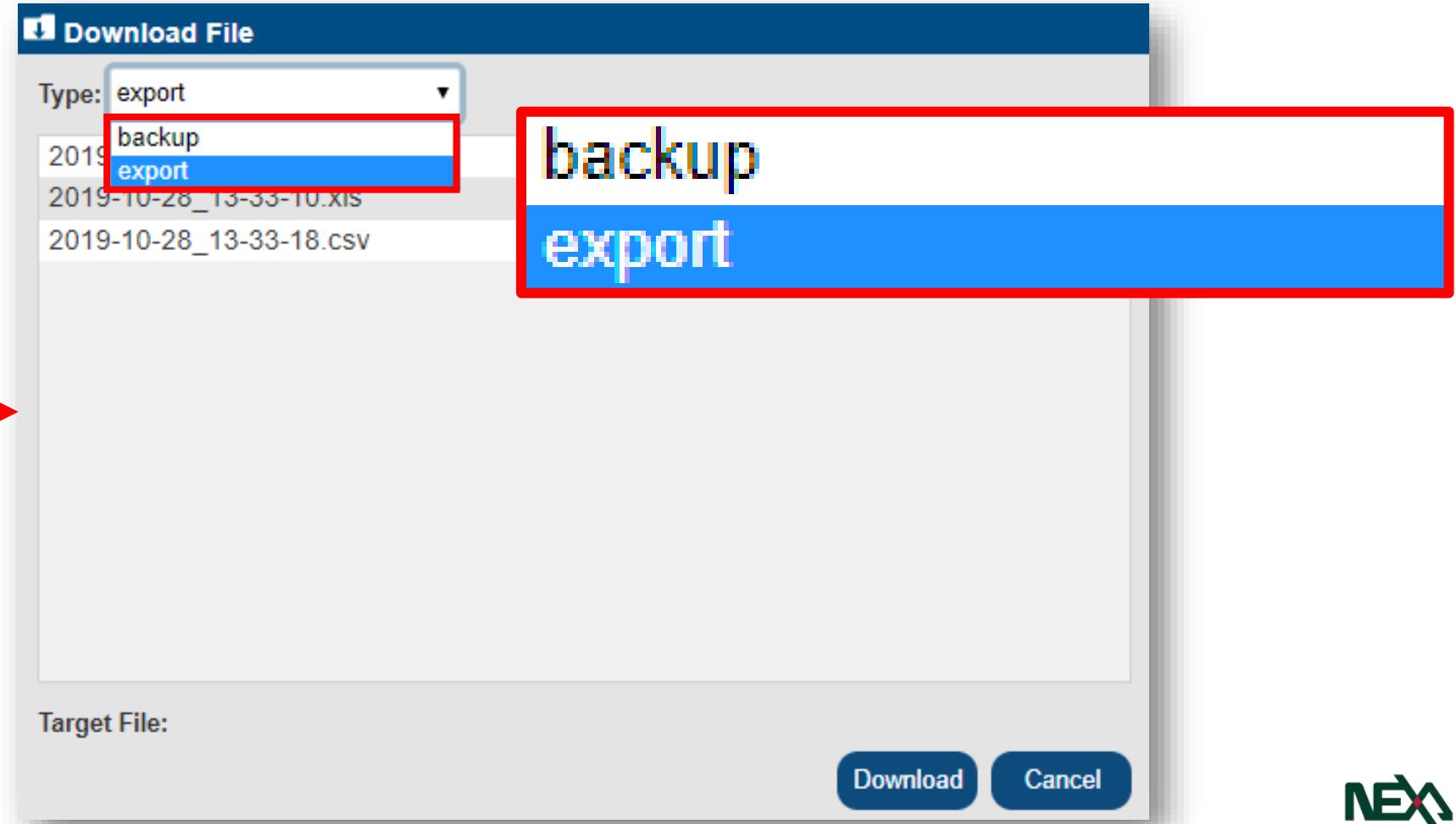
VIC7000 每天會自動備份七天以前的資料，備份的檔案為 .xlsb ，備份路徑為設定頁面中設定的路徑，備份在 VIC7000 電腦上



A screenshot of an Excel spreadsheet titled '2019-07-16\_15-46-00.xlsx'. The spreadsheet contains a table with columns labeled A through Z. The data consists of 104 rows of information, each row containing a unique ID (e.g., 10390, 10391, 10392, etc.) and various numerical and text values. The table includes columns for 'Pass' (containing values like '2 Pass', '1 Pass', '2 NG'), 'Text' (containing '29062 Text', '29066 Text', '29068 Text', etc.), 'List' (containing 'List'), and 'Image' (containing 'Image'). The 'Text' and 'Image' columns are visually represented by small icons of document and image files respectively. The 'Pass' column indicates the test status for each row.

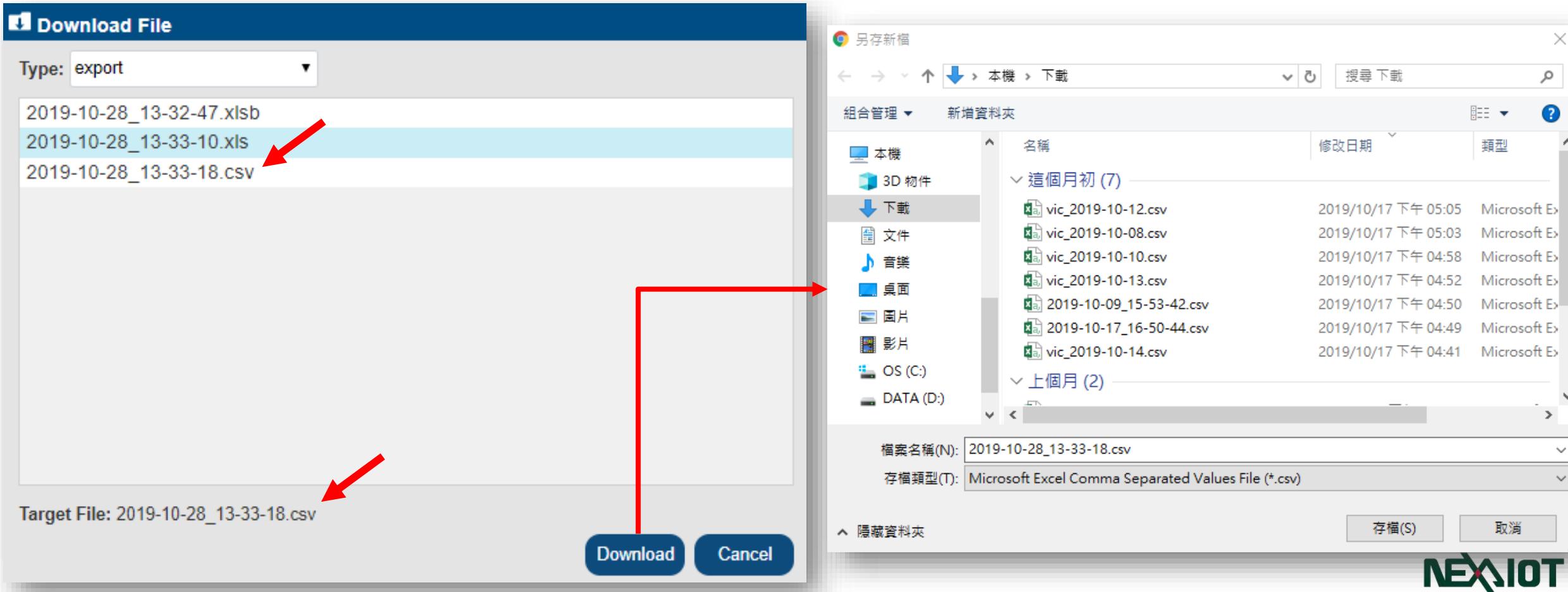
- 遠端下載檔案

點擊檔案按鈕，可由此下載存放在 VIC 內的匯出和備份資料



## • 遠端下載檔案

選定要下載的檔案後，點擊下載，即可遠端下載檔案，路徑為瀏覽器設定的下載路徑



## • 資料庫注意事項

1. 資料庫內可保留七天內的資料，七天前的資料會備份在 VIC7000 本機內
2. VIC7000 有**硬體容量偵測機制**，若可用容量**低於 20G**，會跳出警示，並記錄在日誌內，此時資料會**繼續儲存**；若可用容量**低於 10G**，會跳出警示，並記錄在日誌內，此時新的資料將**覆蓋**舊的資料。
3. VIC7000 有**自動關閉存圖機制**，每分鐘計算一次，並記錄在日誌中。若資料庫一分鐘內儲存的**辨識失敗數量**超過**閥值**，就會**關閉存圖**功能，但資料還是會**繼續儲存**，同時計算辨識失敗數量是否超過閥值。若**辨識失敗數量**小於**閥值**，則**恢復存圖**；反之，則**關閉存圖**。辨識失敗代表辨識結果為「NG」或「No Matching」。
4. 匯出資料最大值為一個檔案**一百萬筆資料**，若超出則會自動切割成多個檔案。
5. VIC7000 可以每天自動備份資料，備份時間約**半小時以上**，依資料筆數而定。進行備份時，應避免進行專案編輯。

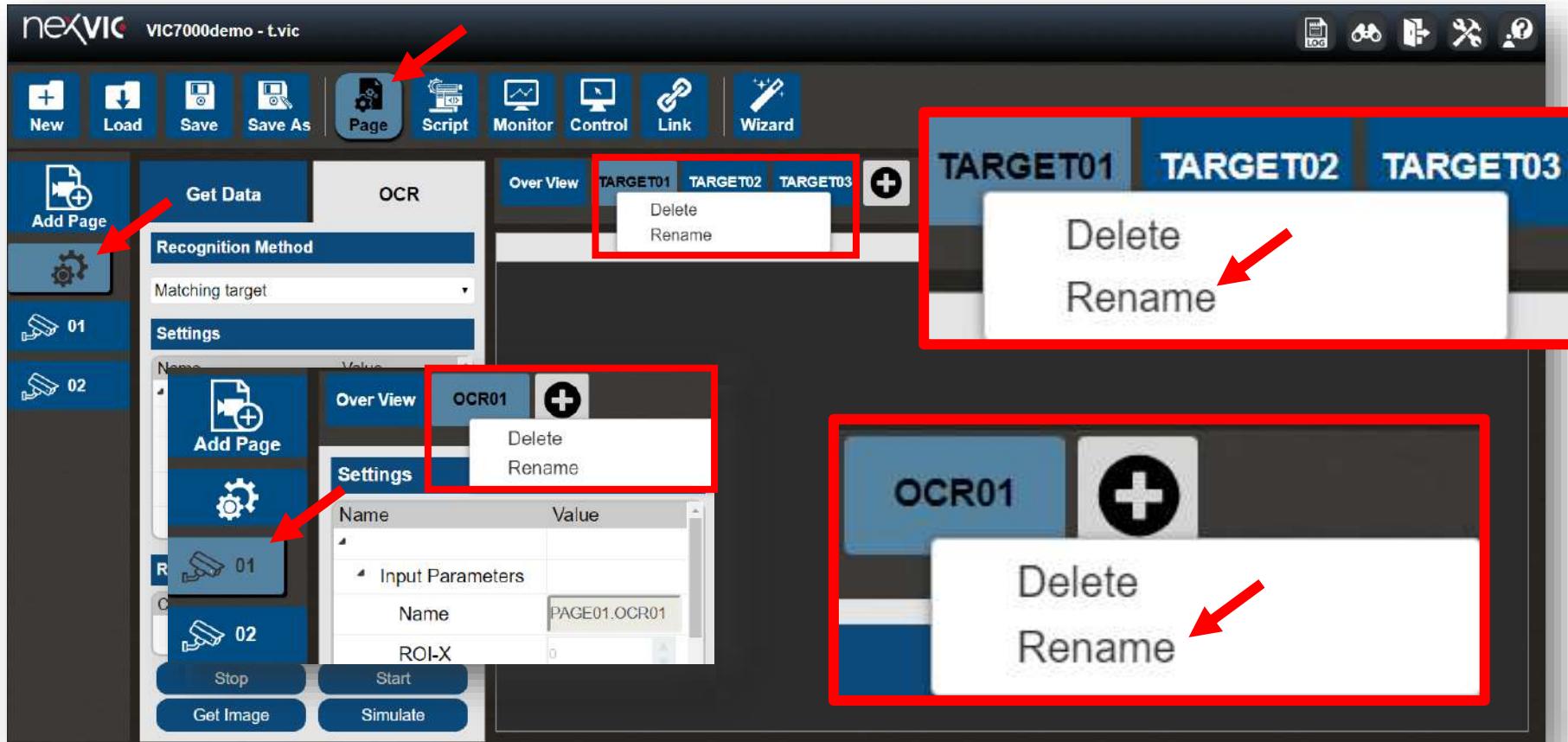
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# 重新命名

- 重新命名設定

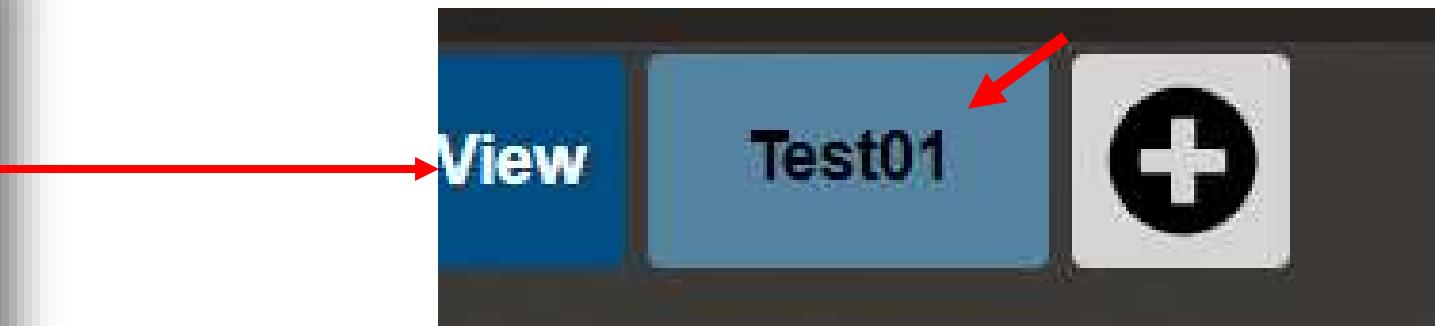
1. 重新命名：進入功能頁面 → 在 PAGE、TARGET、TARGET\_COLOR、TARGET\_PATTERN、OCR、COLOR、PATTERN 欄位上點擊右鍵 → 選擇重新命名  
這裡以 TARGET 和 OCR 為例



# 重新命名

- 重新命名設定

1. 重新命名：輸入新名稱(不可重覆) → 點擊確定，可看到 TARGET 和 OCR 已被重新命名



# 重新命名

- 重新命名結果

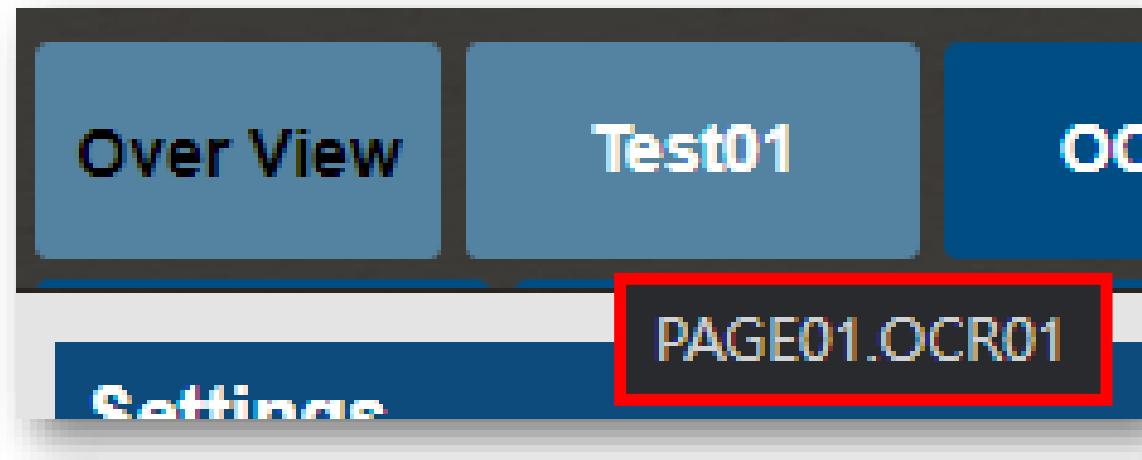
- 檢視重新命名：在一般模式啟動專案時，就可以看到重新命名的內容



# 重新命名

- 重新命名結果

- 檢視重新命名：把滑鼠移到重新命名的 TARGET 和 OCR 上，會出現原名稱



# 重新命名

- 重新命名結果

- 檢視重新命名：在資料庫頁面內，把滑鼠移到重新命名的 TARGET 和 OCR 上，會出現修改後的名稱

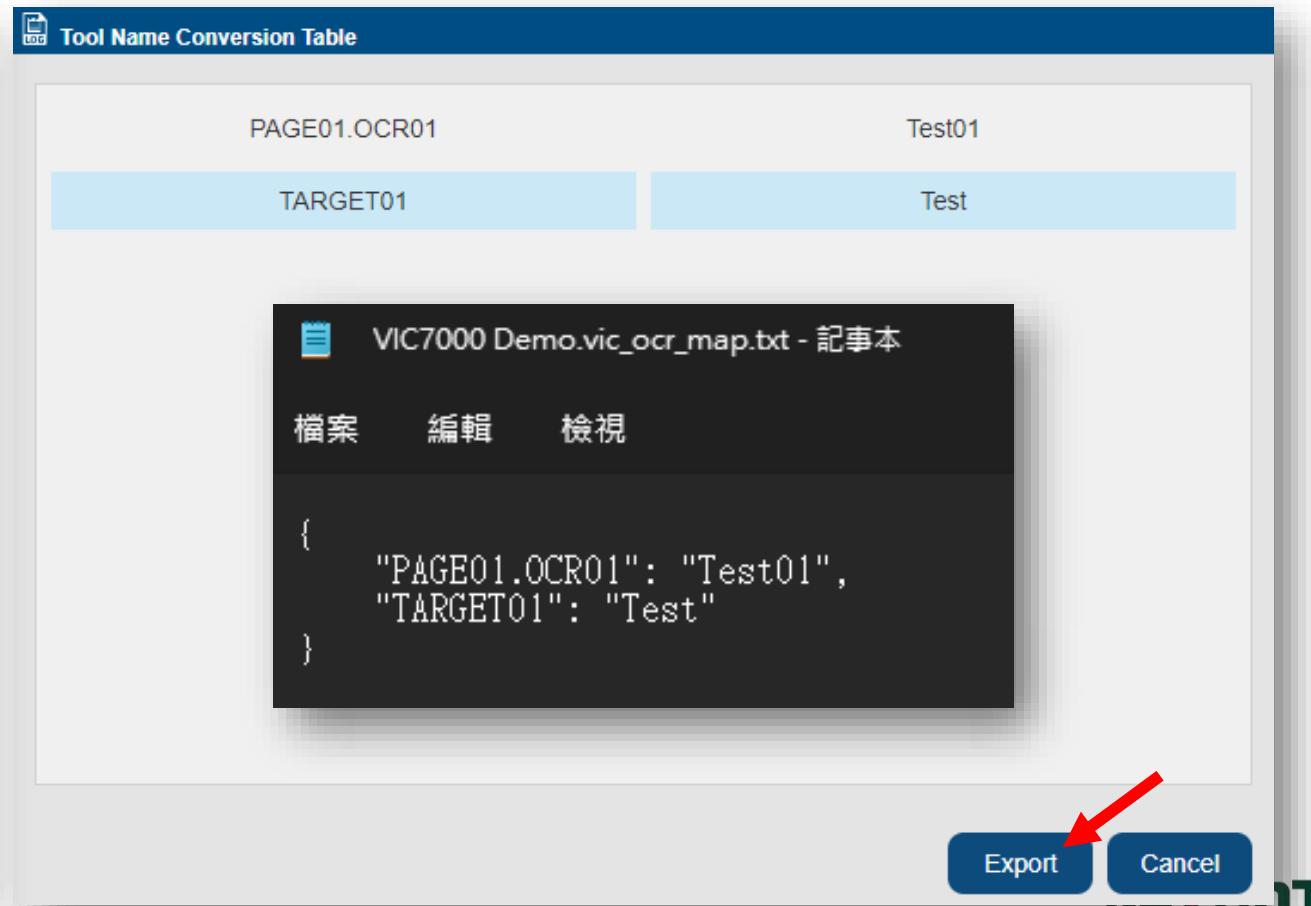
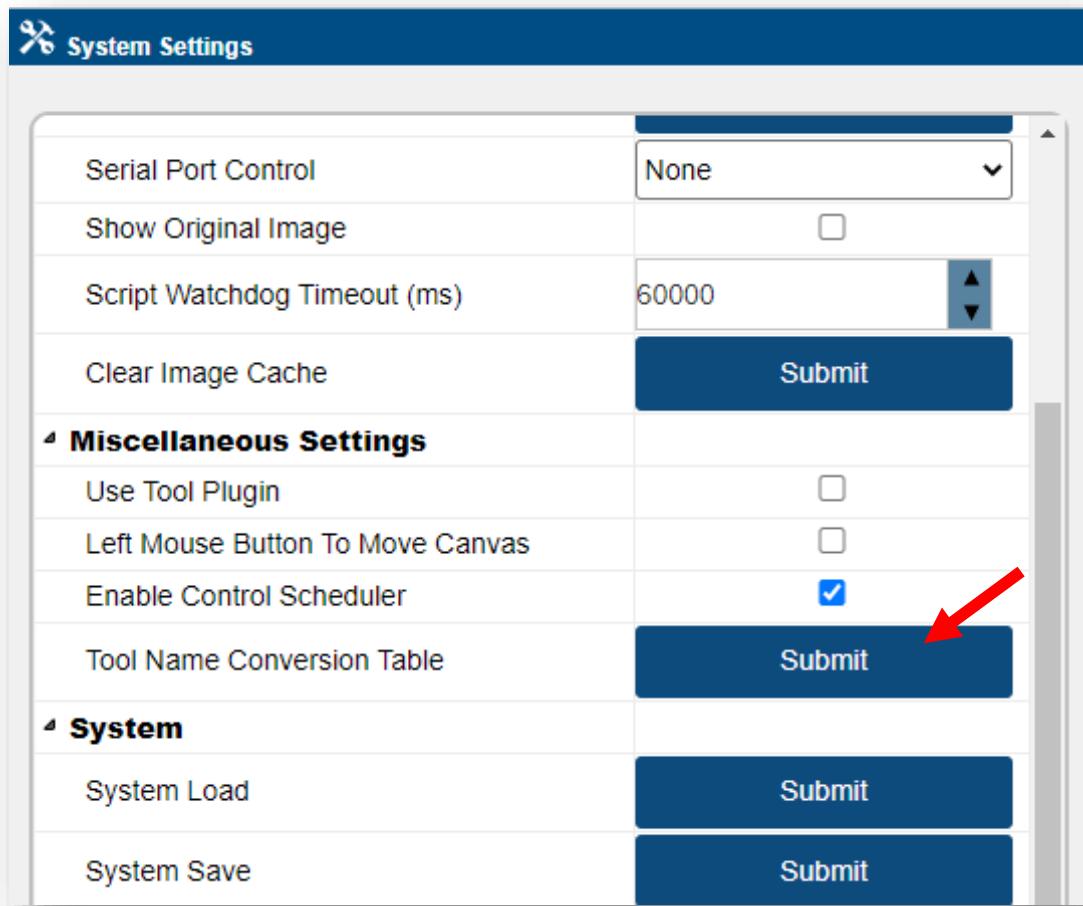
	TARGET_01	OCR_01
	55	Test
	55	125
	55	112

	TARGET_01	OCR_01	OCR_02
	55	185	110
	55	125	PAGE01.OCR01:Test01
	55	112	110

# 重新命名

- 重新命名結果

- 檢視重新命名：在設定視窗內，點擊工具命名對照圖的執行鍵，會出現重新命名的對照圖，亦可輸出成 .txt，為 json 格式文字



# 重新命名

- 重新命名結果

- 檢視重新命名：用 RESTful API 方式，在網址列輸入 **IP/restful/ocr\_map**，會出現重新命名的對照內容，為 json 格式文字

192.168.133.144/restful/ocr\_map

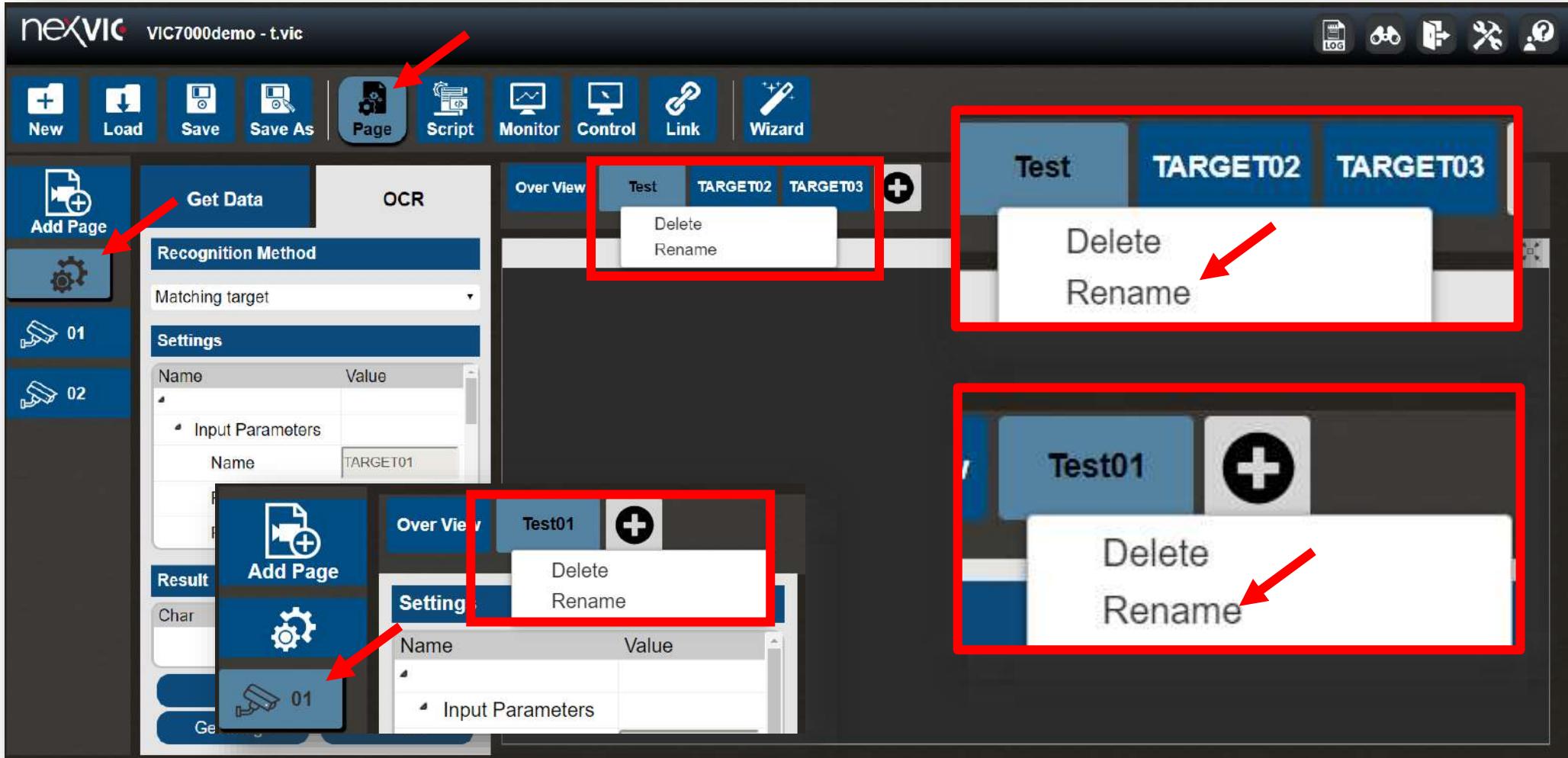


```
{"TARGET01": "Test", "PAGE01.OCR01": "Test01"}
```

# 重新命名

- 重新命名復原

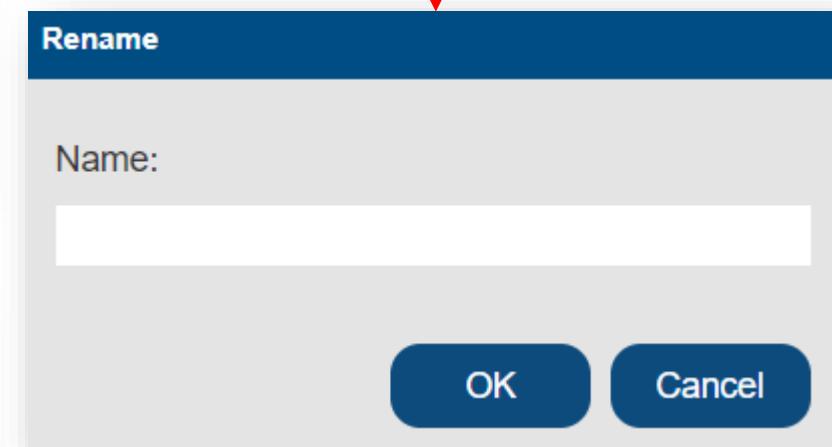
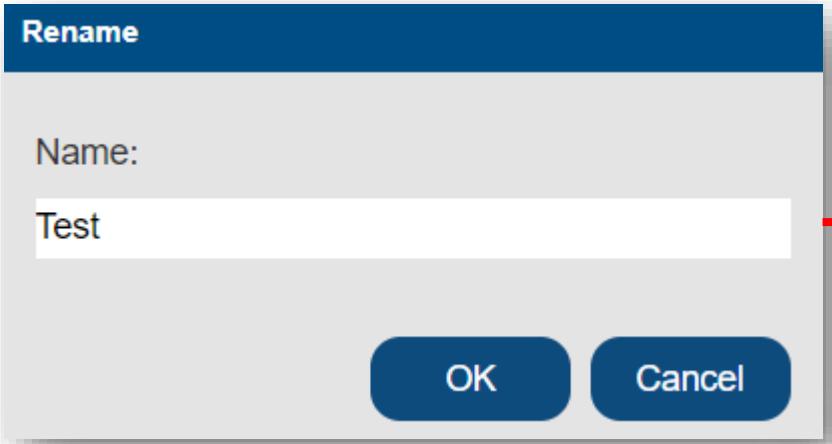
- 復原：進入功能頁面 → 在重新命名後的 TARGET 和 OCR 上點擊右鍵 → 選擇重新命名



# 重新命名

- 重新命名復原

- 復原：清空新名稱 → 點擊確定，TARGET 和 OCR 會變回預設名稱



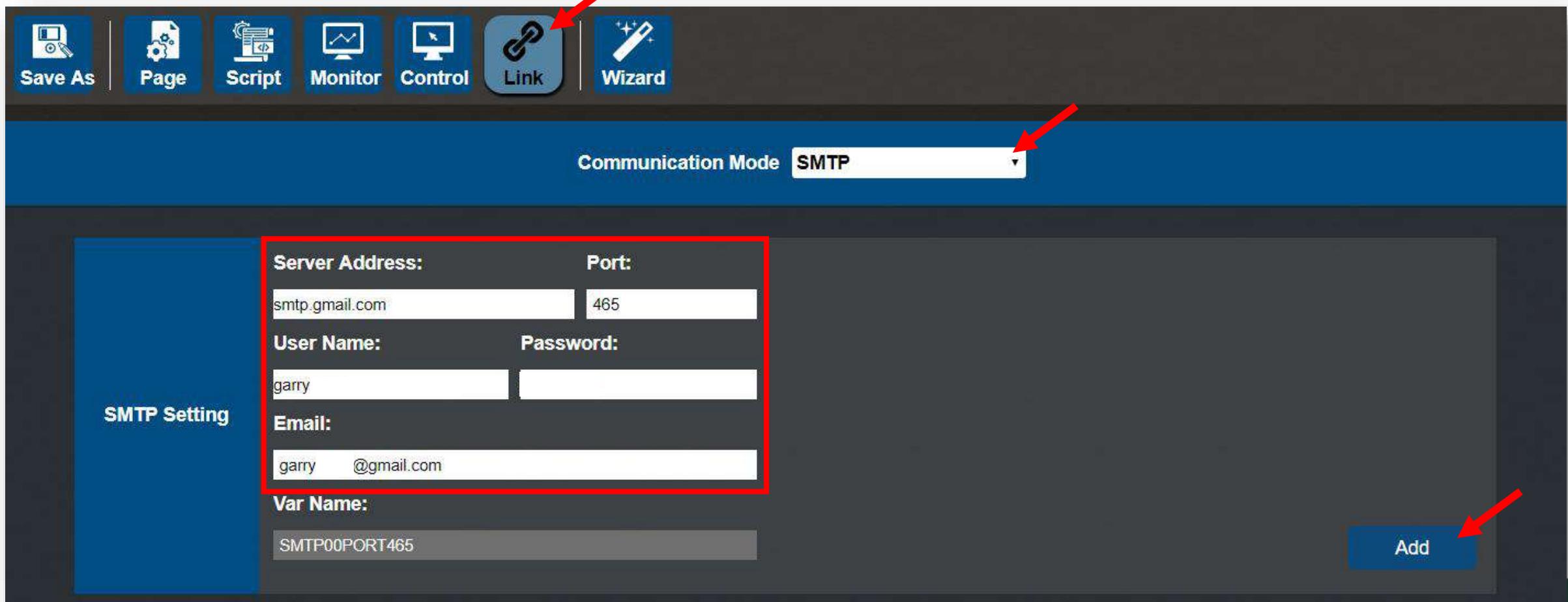
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# 事件觸發

- 連線設定

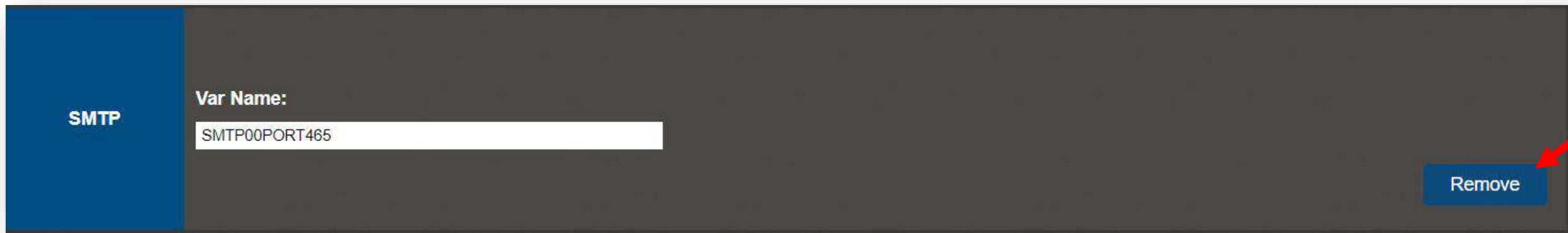
1. 新增連線 – Email：進入連結頁面 → 通訊模式選擇 SMTP → 輸入伺服器位址、通訊埠、使用者名稱、密碼、電子信箱位址 → 點擊新增



# 事件觸發

- 連線設定

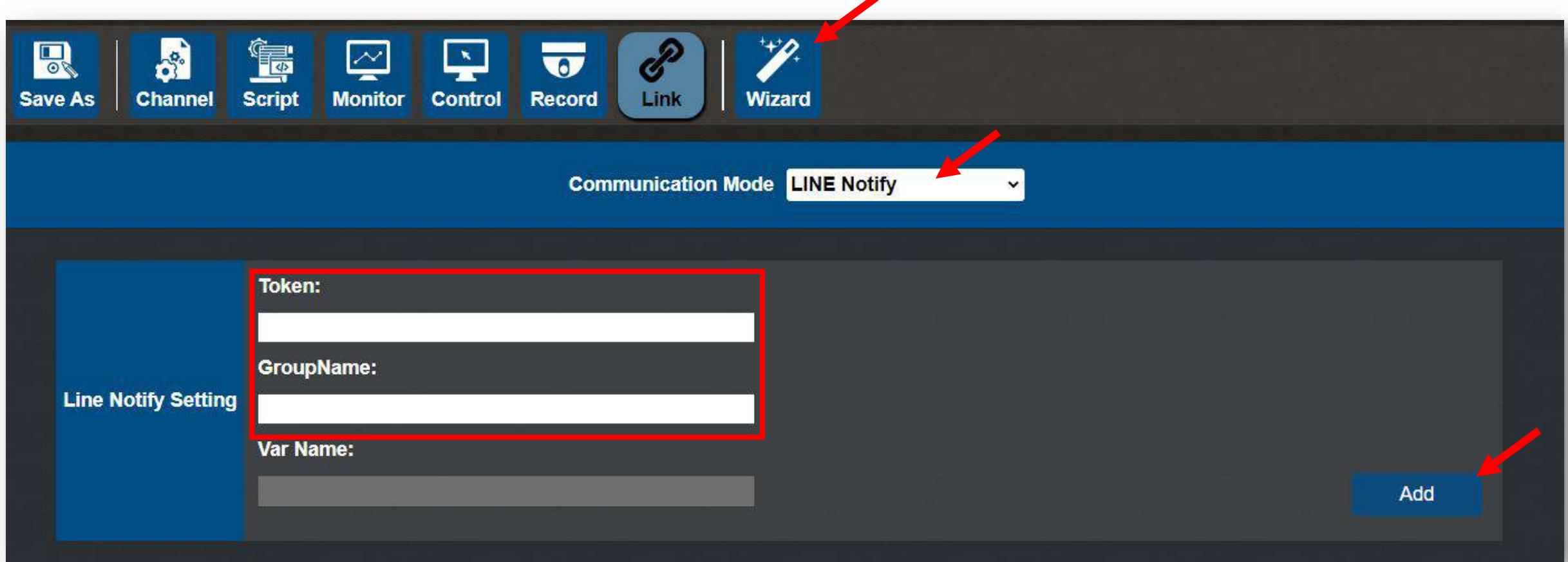
1. 新增連線 – Email：下方確認已新增 SMTP 連線，可在此移除連線



# 事件觸發

- 連線設定

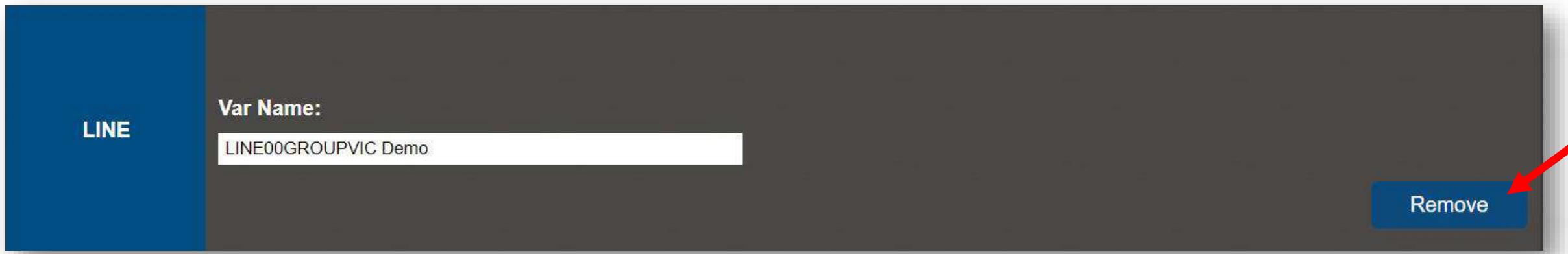
1. 新增連線 – LINE Notify：進入連結頁面 → 通訊模式選擇 LINE Notify、輸入權杖和自定義名稱 → 點擊新增



# 事件觸發

- 連線設定

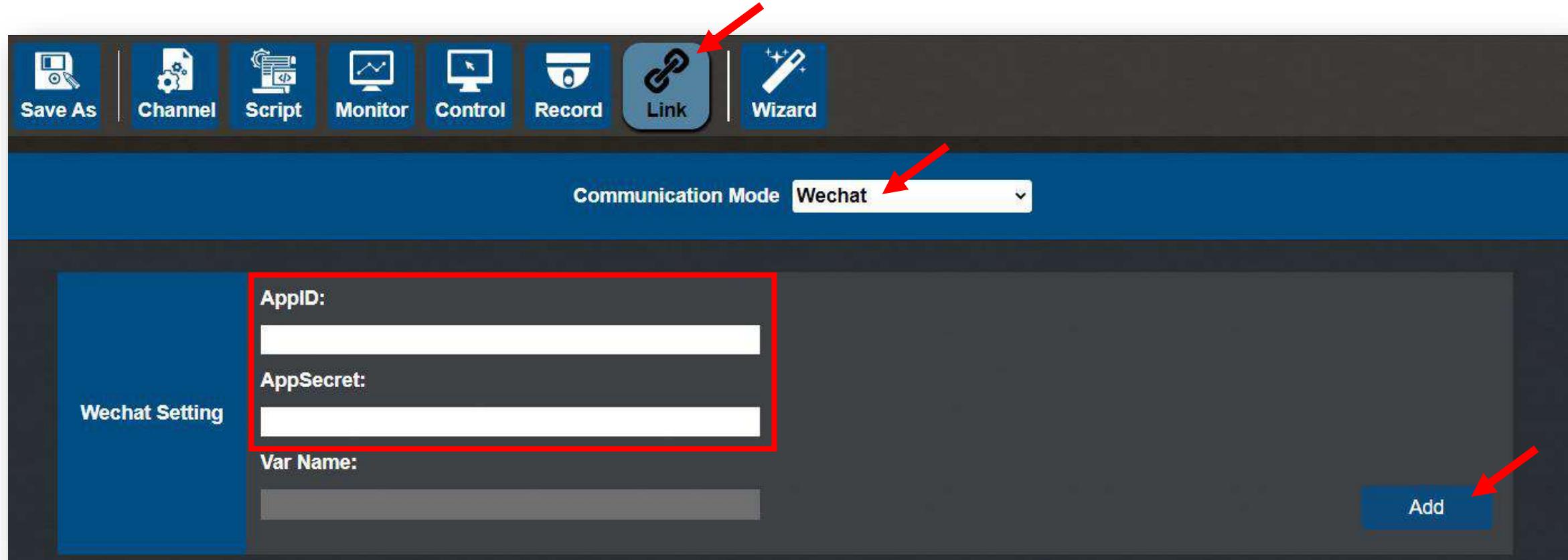
- 新增連線 – LINE Notify：下方確認已新增 LINE 連線，可在此移除連線



# 事件觸發

- 連線設定

1. 新增連線 – WeChat：進入連結頁面 → 通訊模式選擇 WeChat、輸入 appID、appsecret → 點擊新增

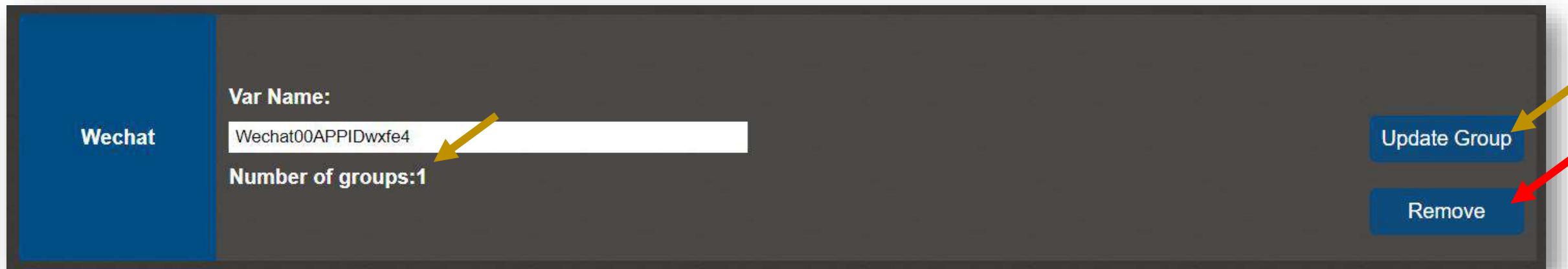


# 事件觸發

- 連線設定

1. 新增連線 – WeChat：下方確認已新增 WeChat 連線，可在此移除連線

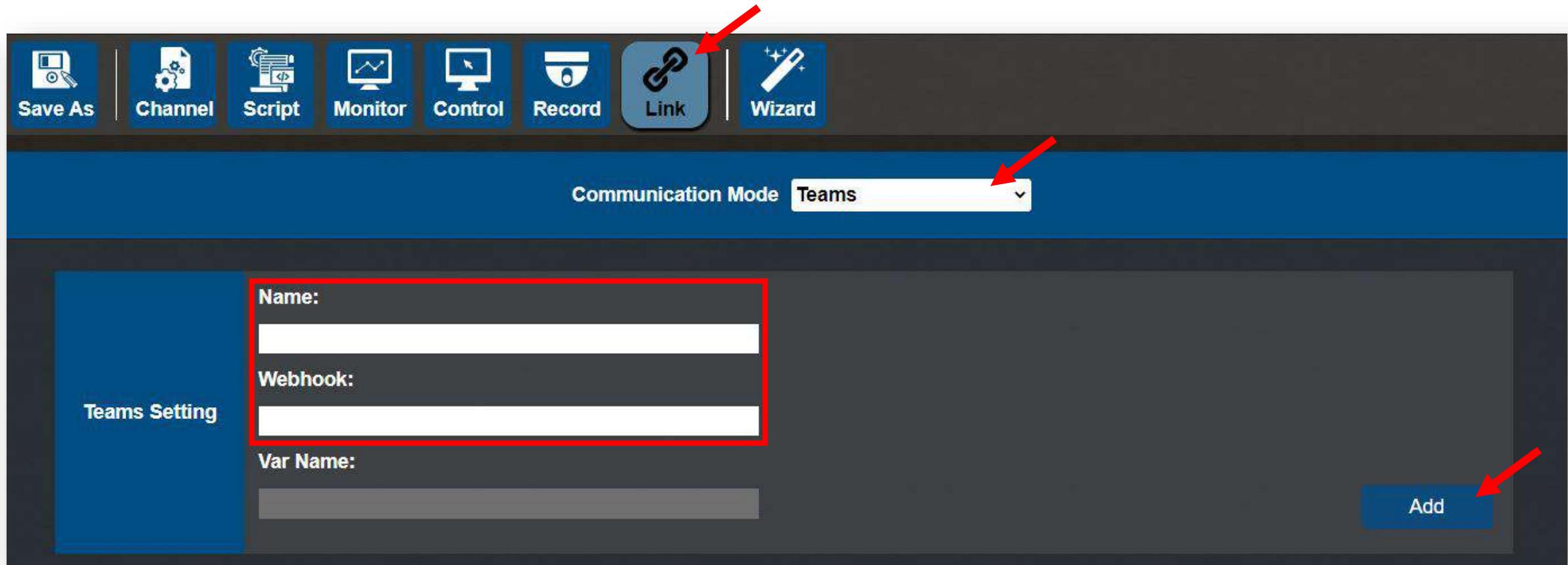
若有新的用戶關注公眾號，需要點擊更新人數更新公眾號人數



# 事件觸發

- 連線設定

1. 新增連線 – Teams : 進入連結頁面 → 通訊模式選擇 Teams 、輸入名稱、Webhook → 點擊新增



# 事件觸發

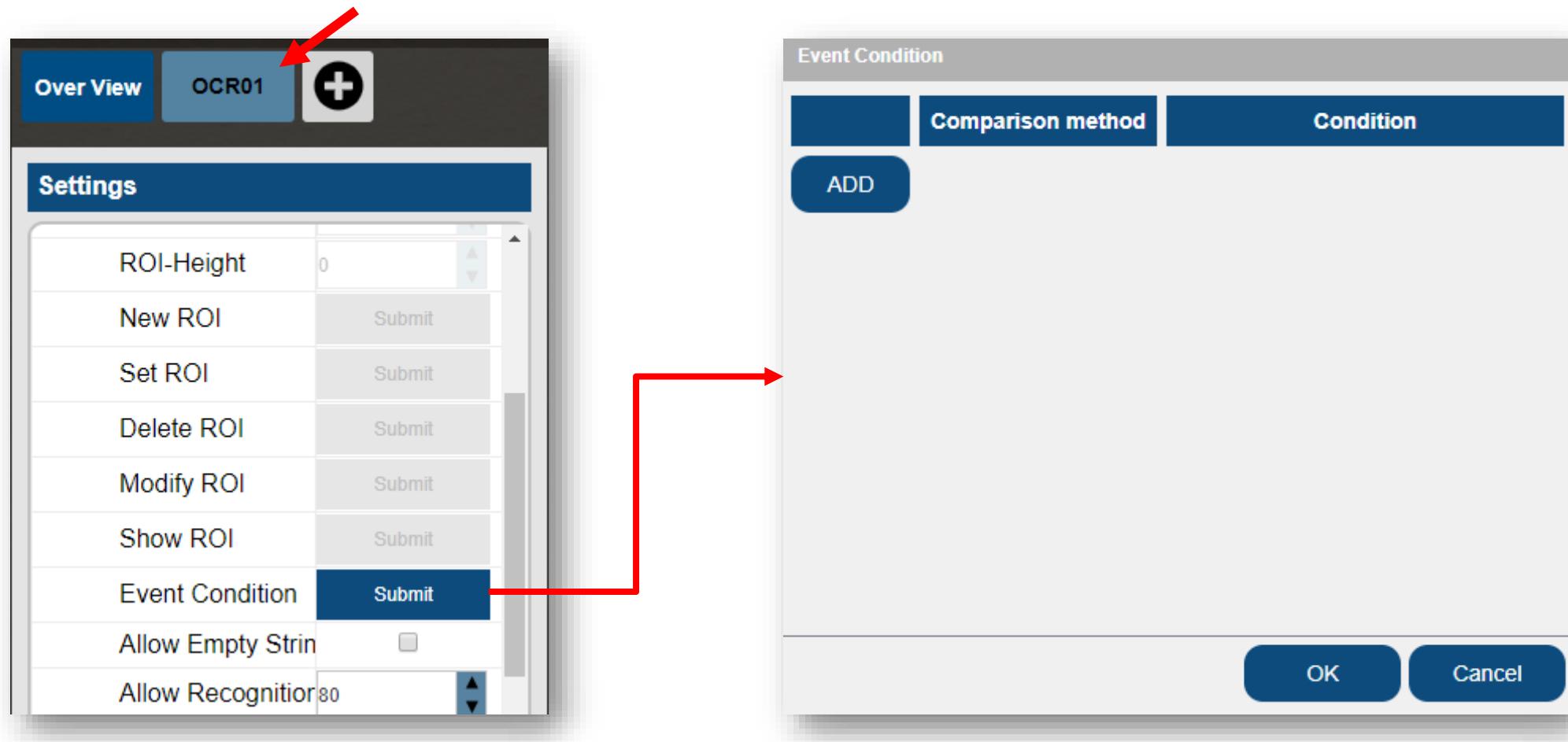
- 連線設定

1. 新增連線 – Teams : 下方確認已新增 Teams 連線，可在此移除連線



## • 事件條件

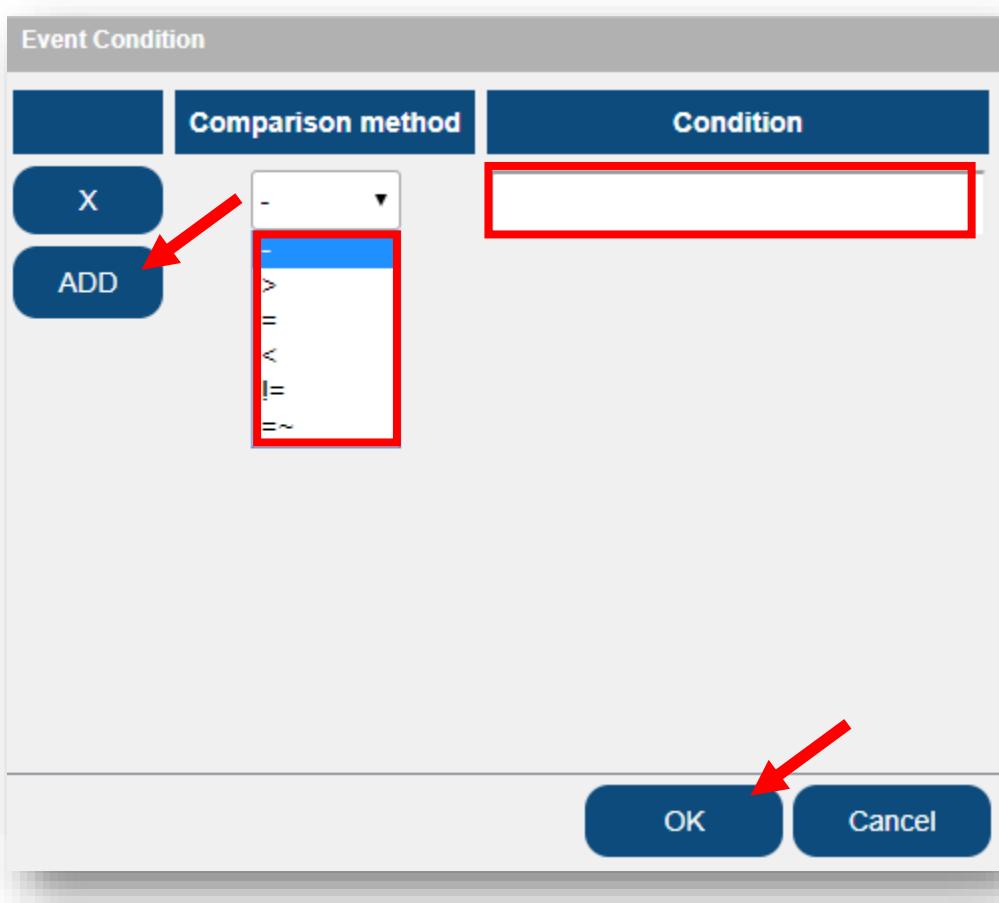
2. 設定事件條件 – **OCR**：點擊事件條件的執行鍵 → 開啟事件條件視窗



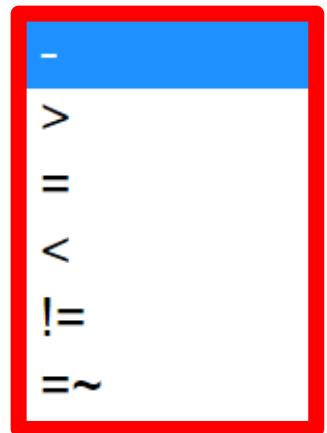
# 事件觸發

- 事件條件

2. 設定事件條件 – OCR : 點擊增加 → 設定比較方法 → 設定條件 → 點擊確定



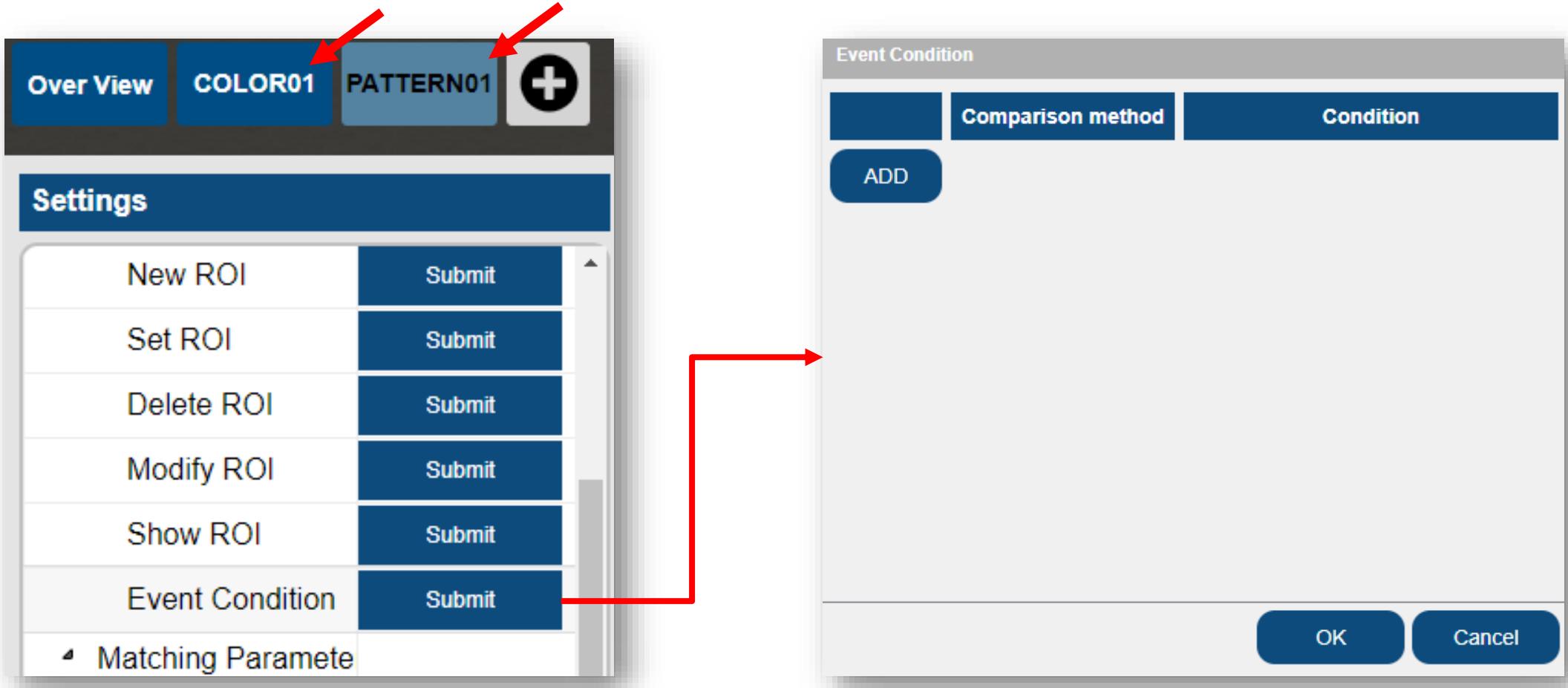
- : 不比較
- > : 大於
- = : 等於
- < : 小於
- != : 不等於
- =~ : 包含(僅限字串)



# 事件觸發

- 事件條件

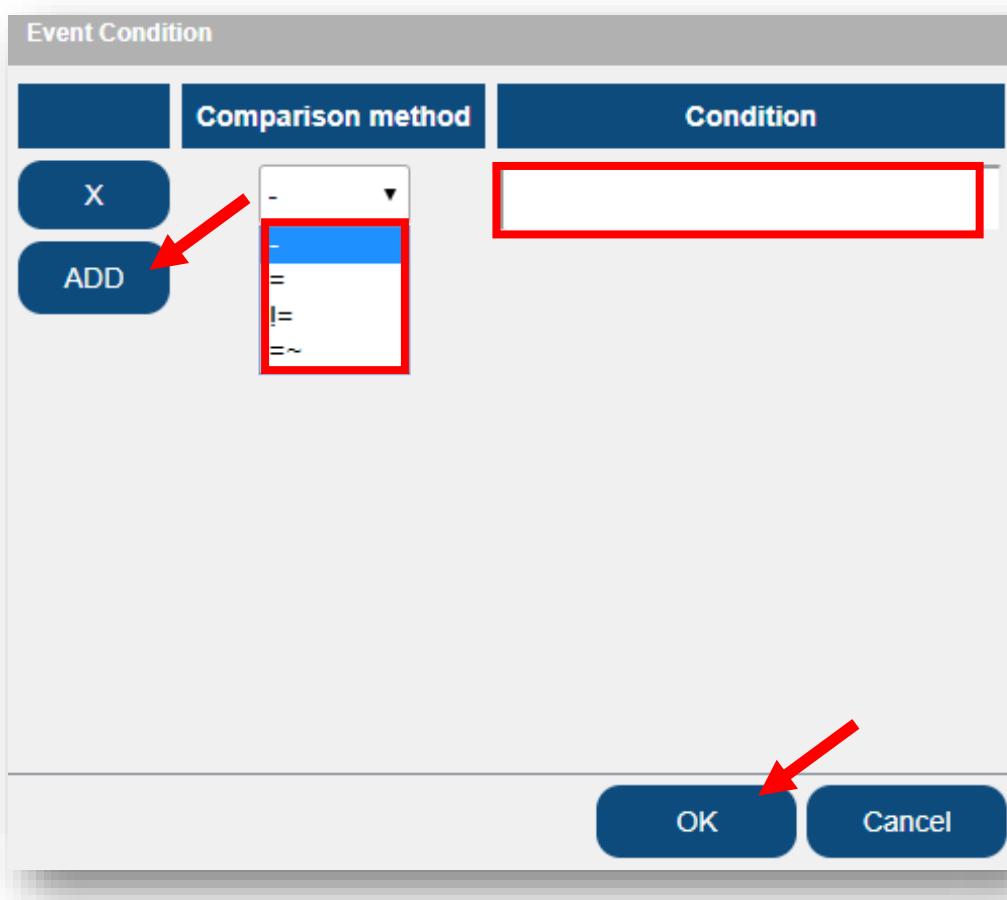
2. 設定事件條件 – 顏色、樣式比對：點擊事件條件的執行鍵 → 開啟事件條件視窗



# 事件觸發

- 事件條件

2. 設定事件條件 – 顏色、樣式比對 : 點擊增加 → 設定比較方法 → 設定條件 → 點擊確定

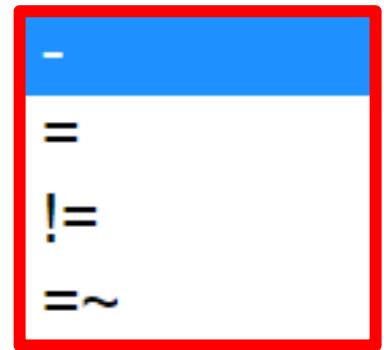


- : 不比較

= : 等於

!= : 不等於

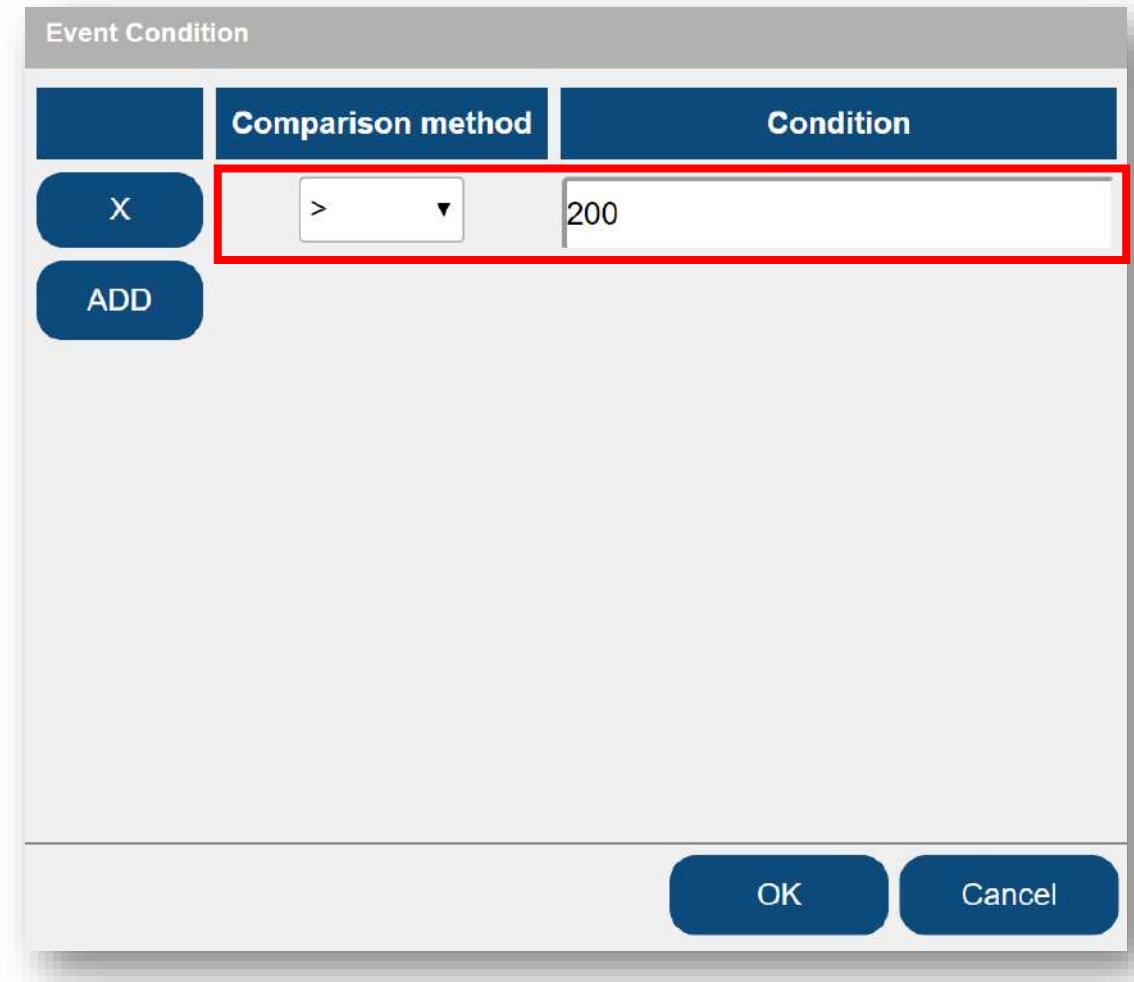
=~ : 包含(僅限字串)



# 事件觸發

- 事件條件

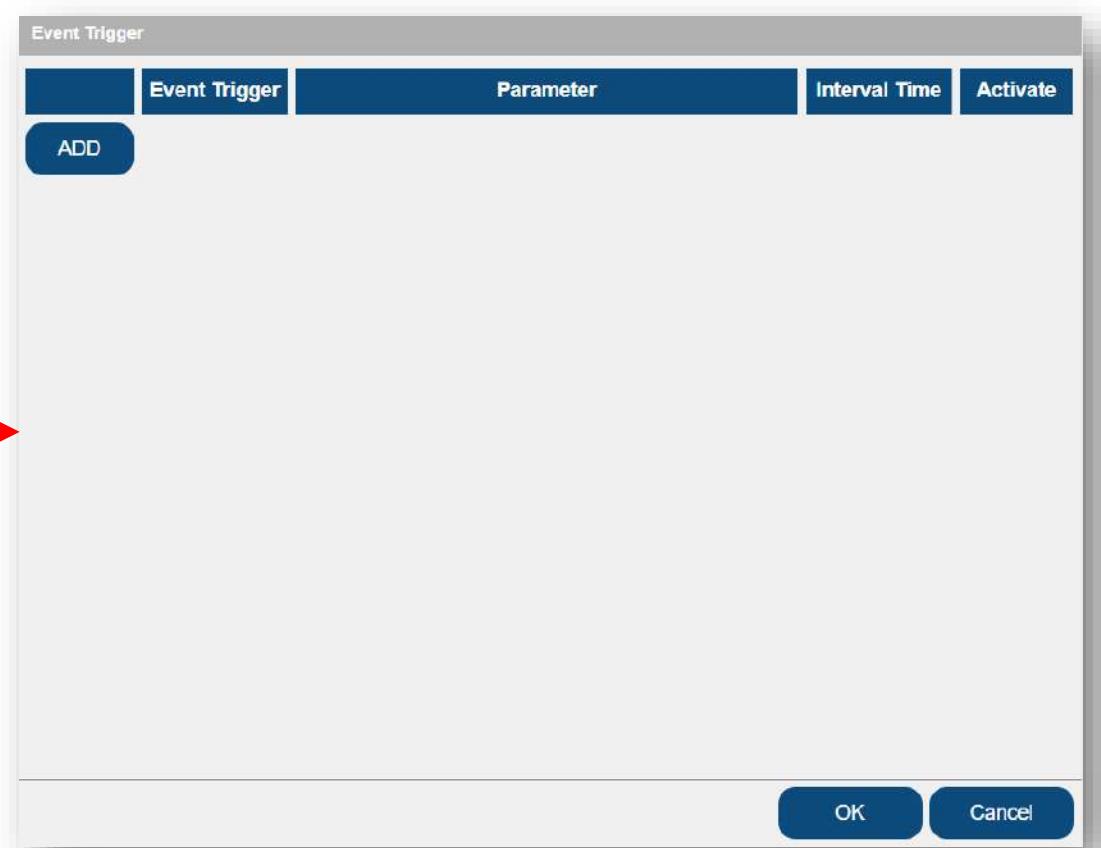
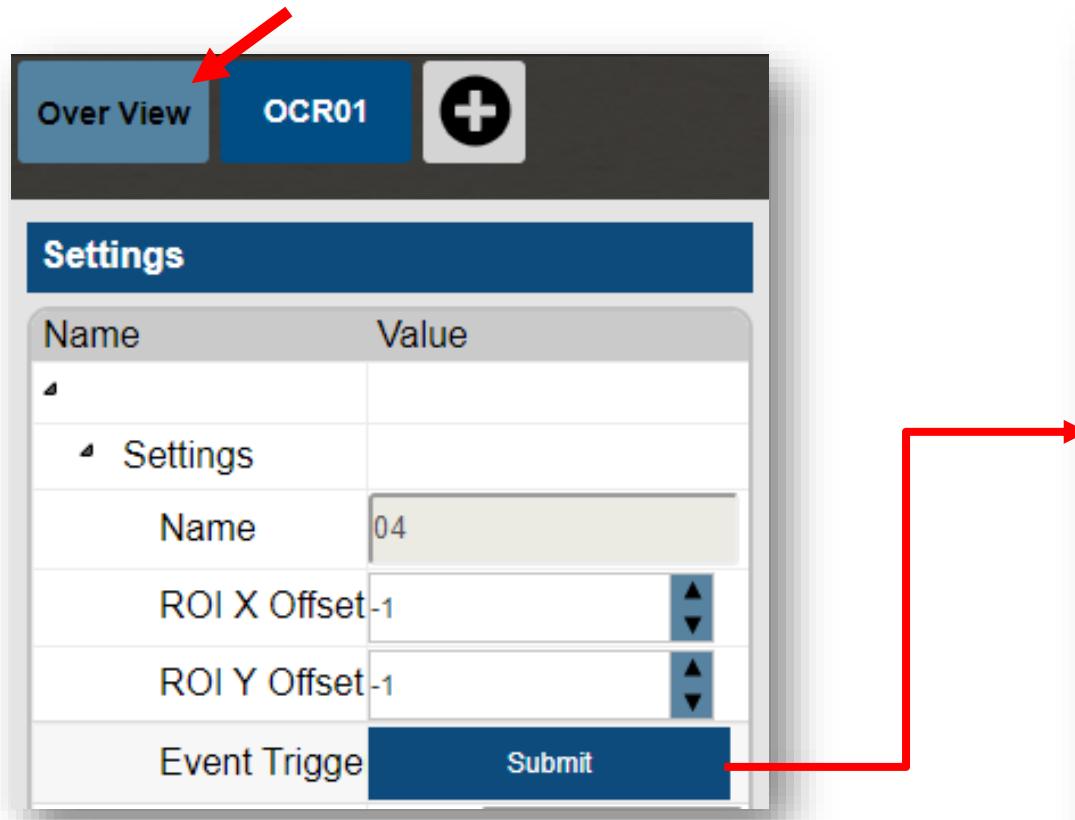
範例 – 設定事件條件 – OCR : 設定 OCR01 的事件條件為大於 200



# 事件觸發

- 事件觸發

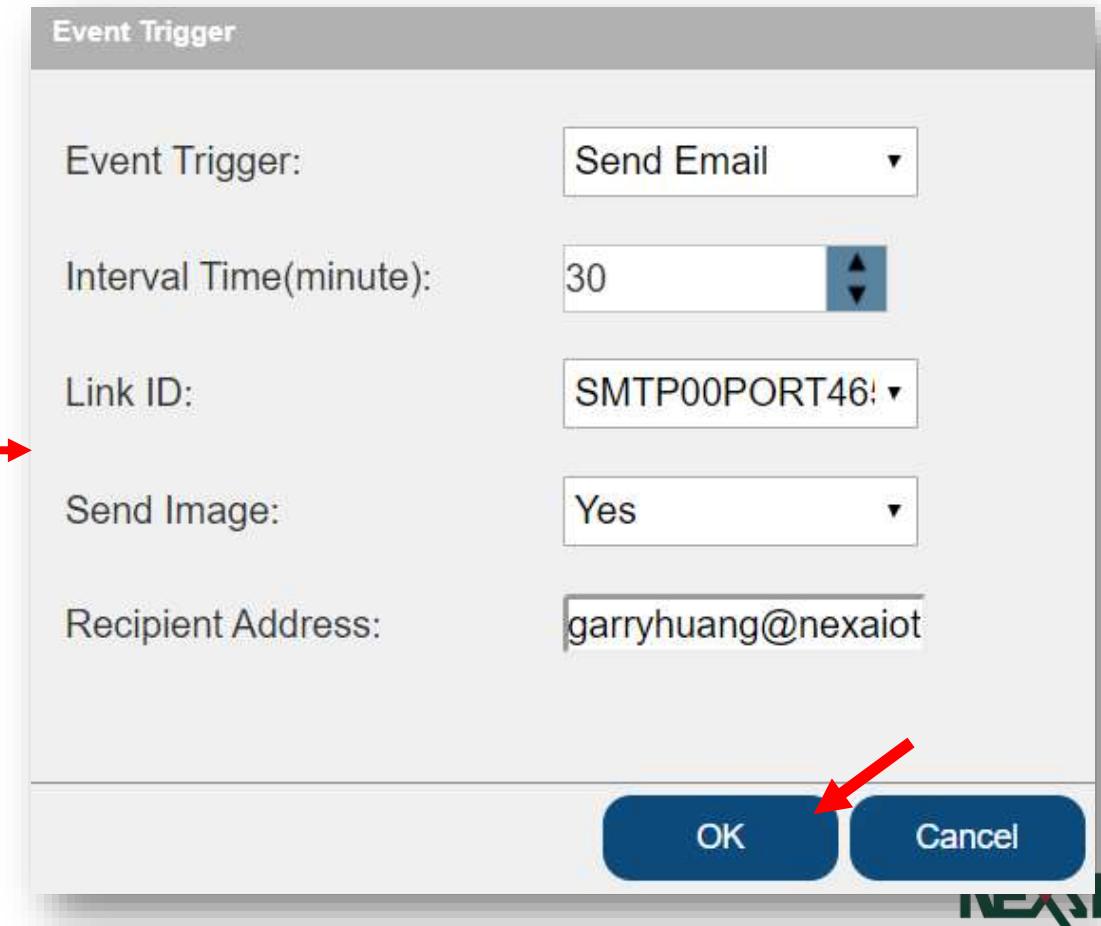
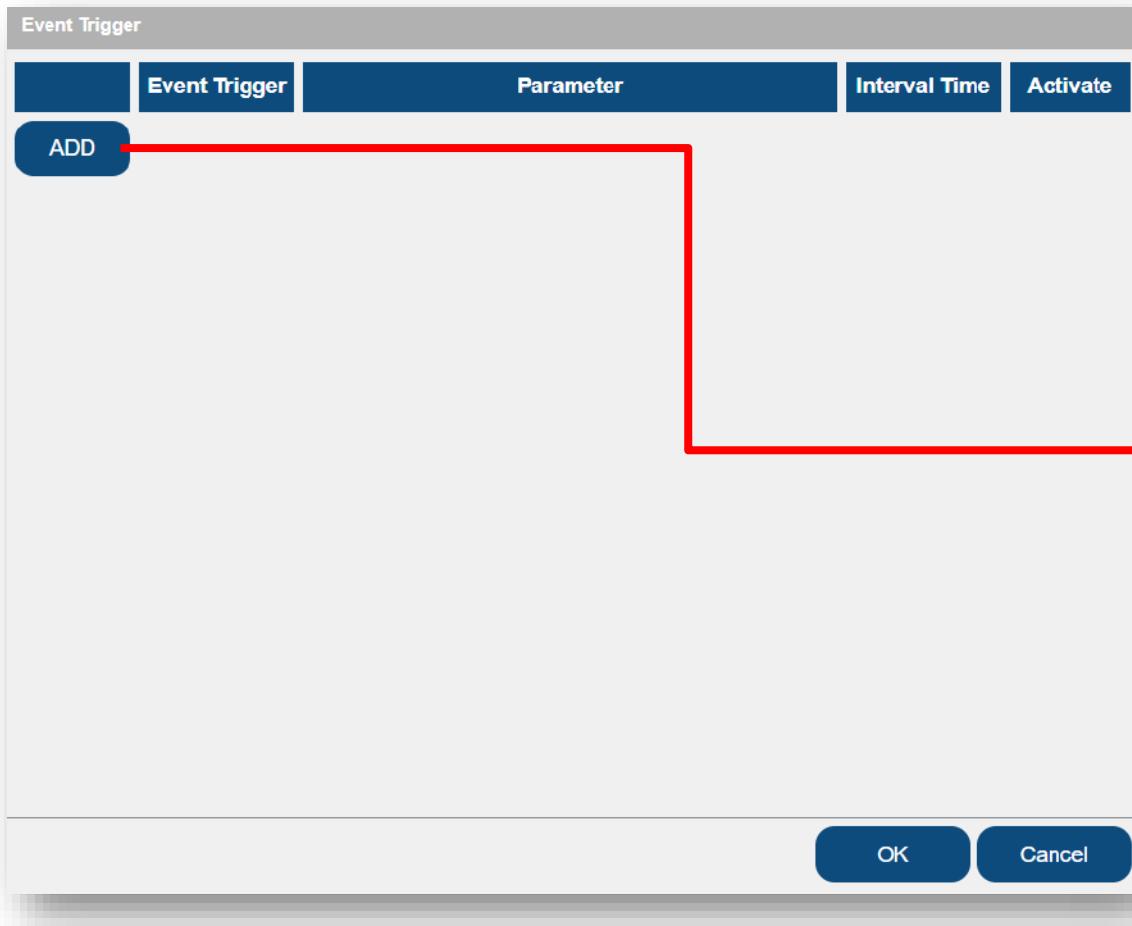
3. 設定事件觸發：點擊事件觸發的執行鍵 → 開啟事件觸發視窗



# 事件觸發

- 事件觸發

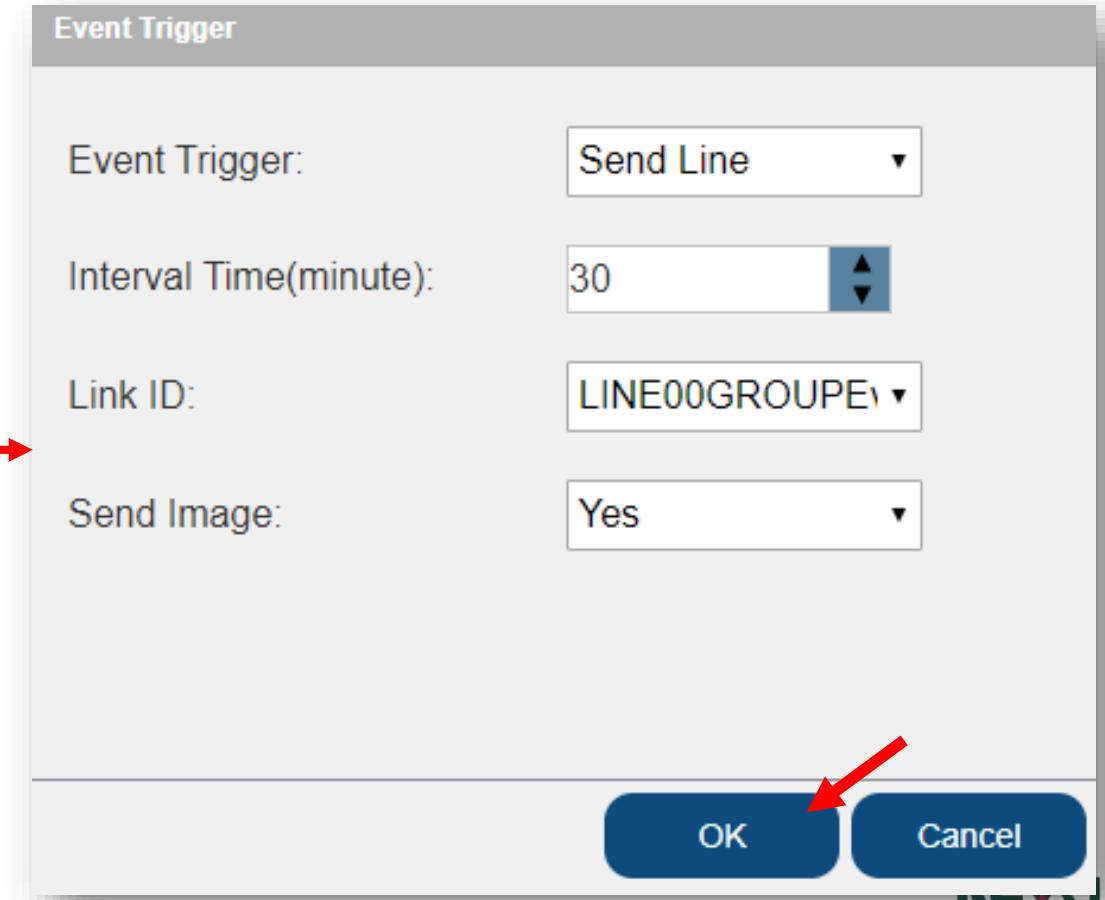
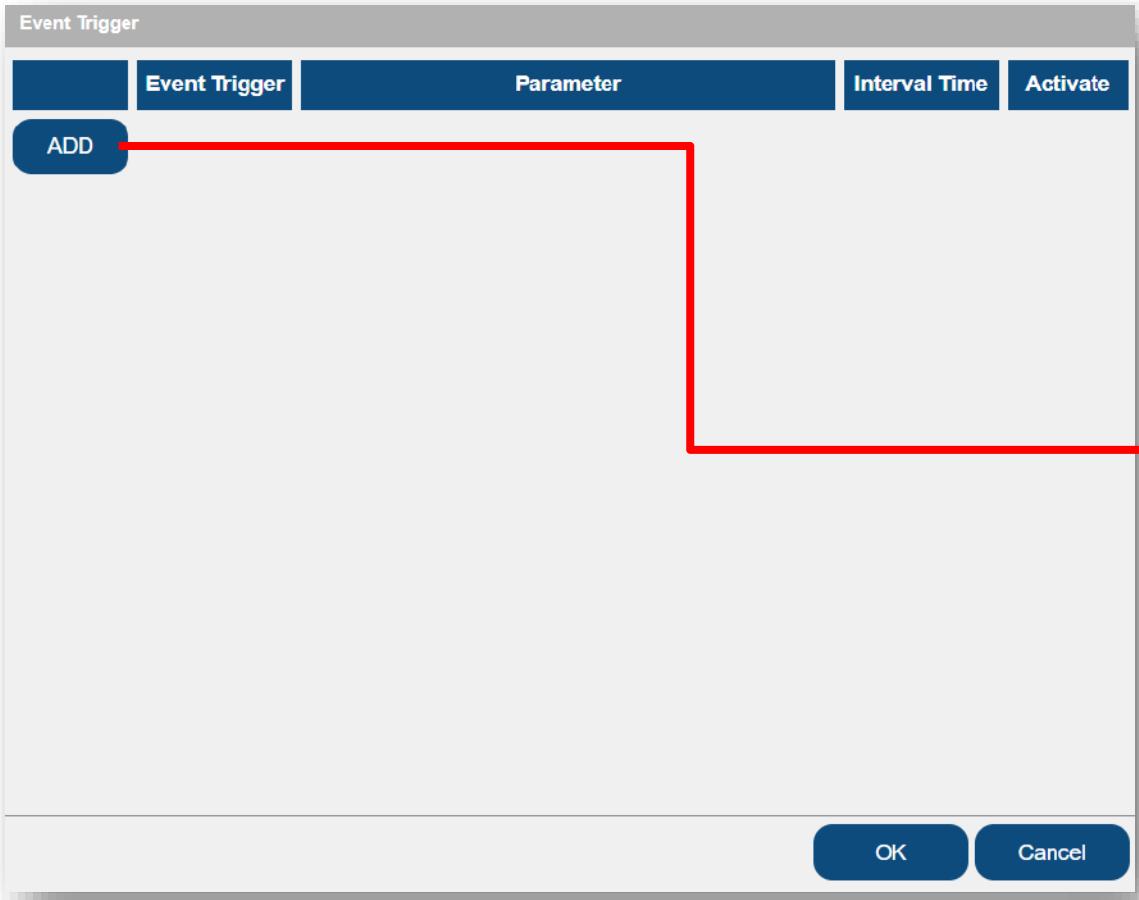
3. 設定事件觸發 – Email：點擊增加 → 事件觸發選擇傳送電子郵件、設定間隔時間、連結名稱選擇 SMTP 連線、選擇是否送圖、輸入收件人信箱 → 點擊確定



# 事件觸發

- 事件觸發

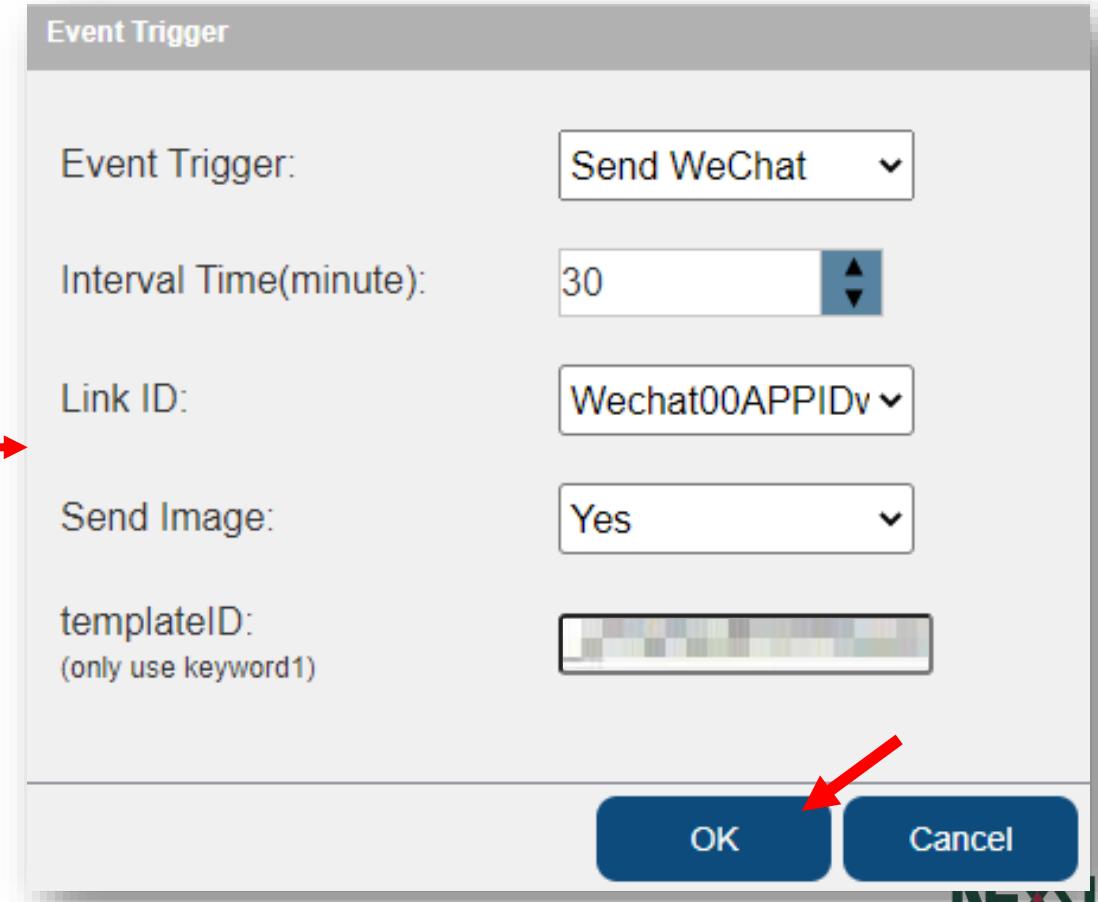
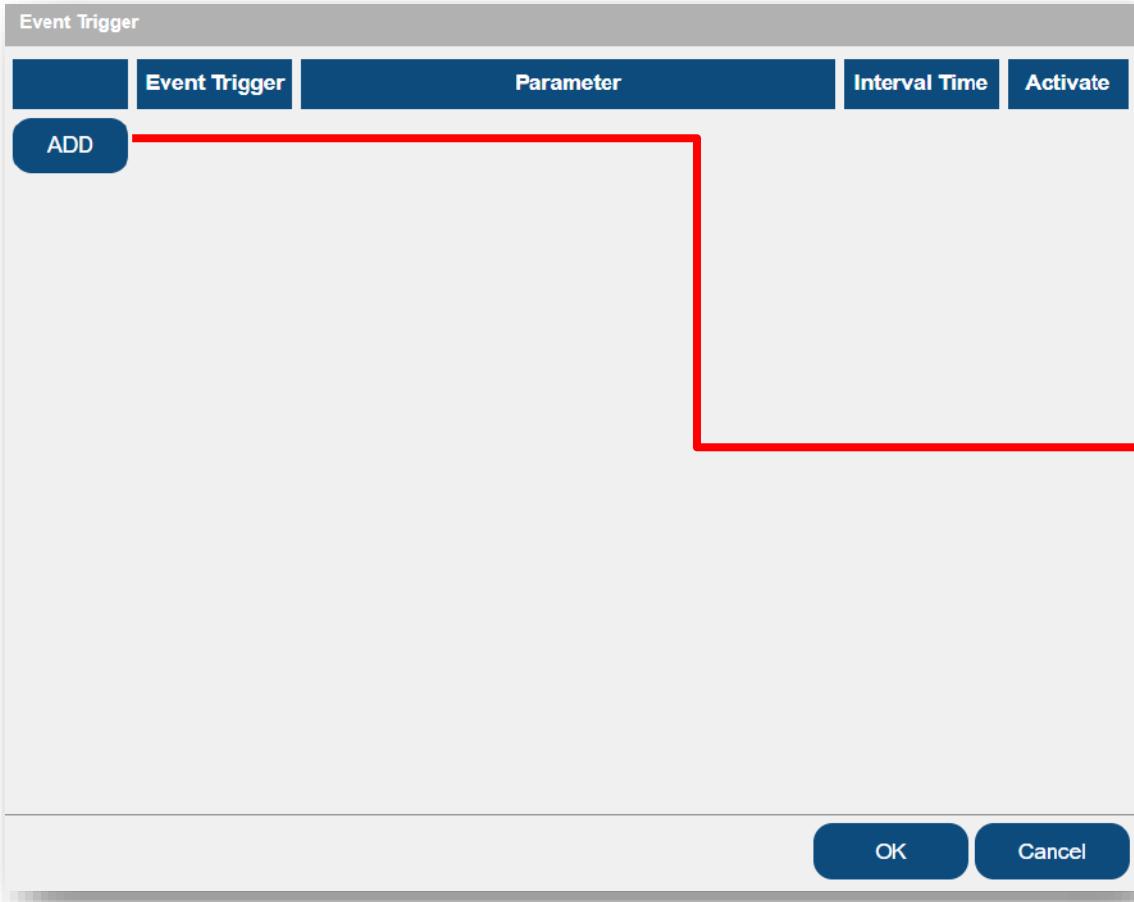
3. 設定事件觸發 – LINE Notify：點擊增加 → 事件觸發選擇傳送 Line、設定間隔時間、連結名稱選擇 LINE Notify 連線、選擇是否送圖 → 點擊確定



# 事件觸發

- 事件觸發

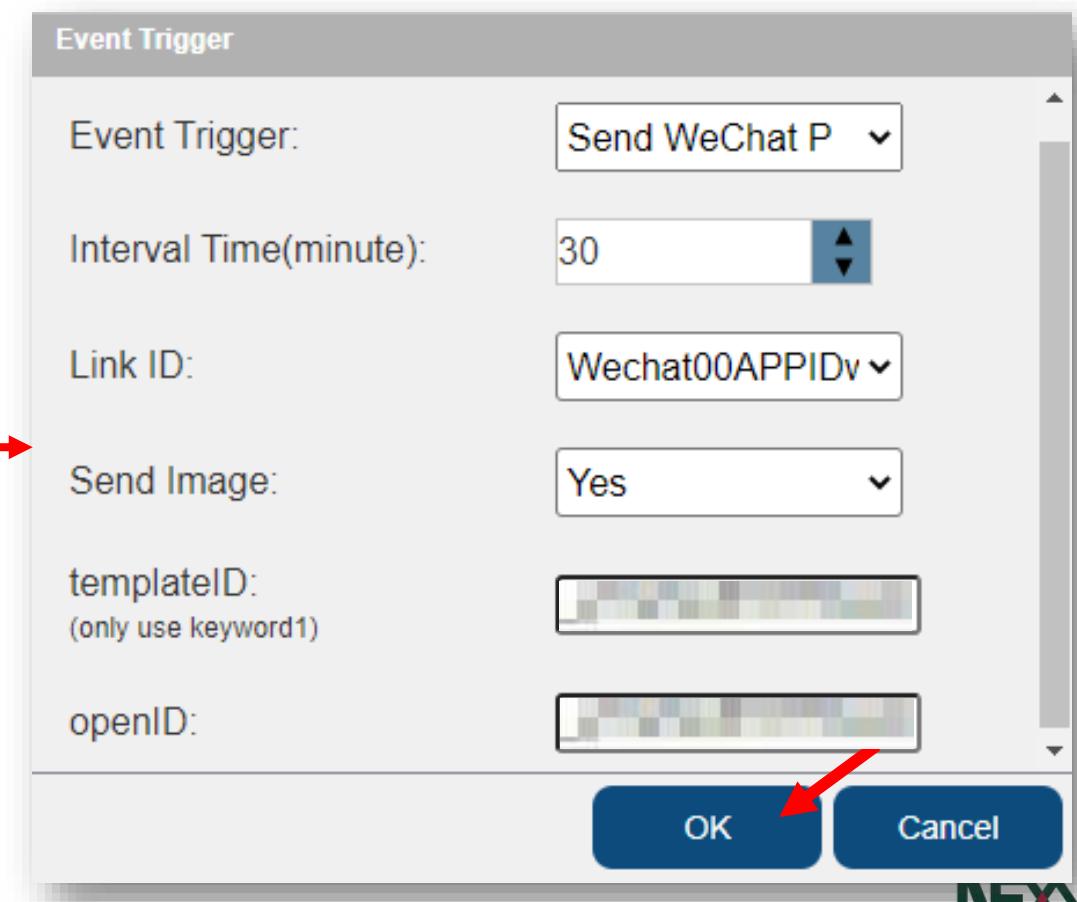
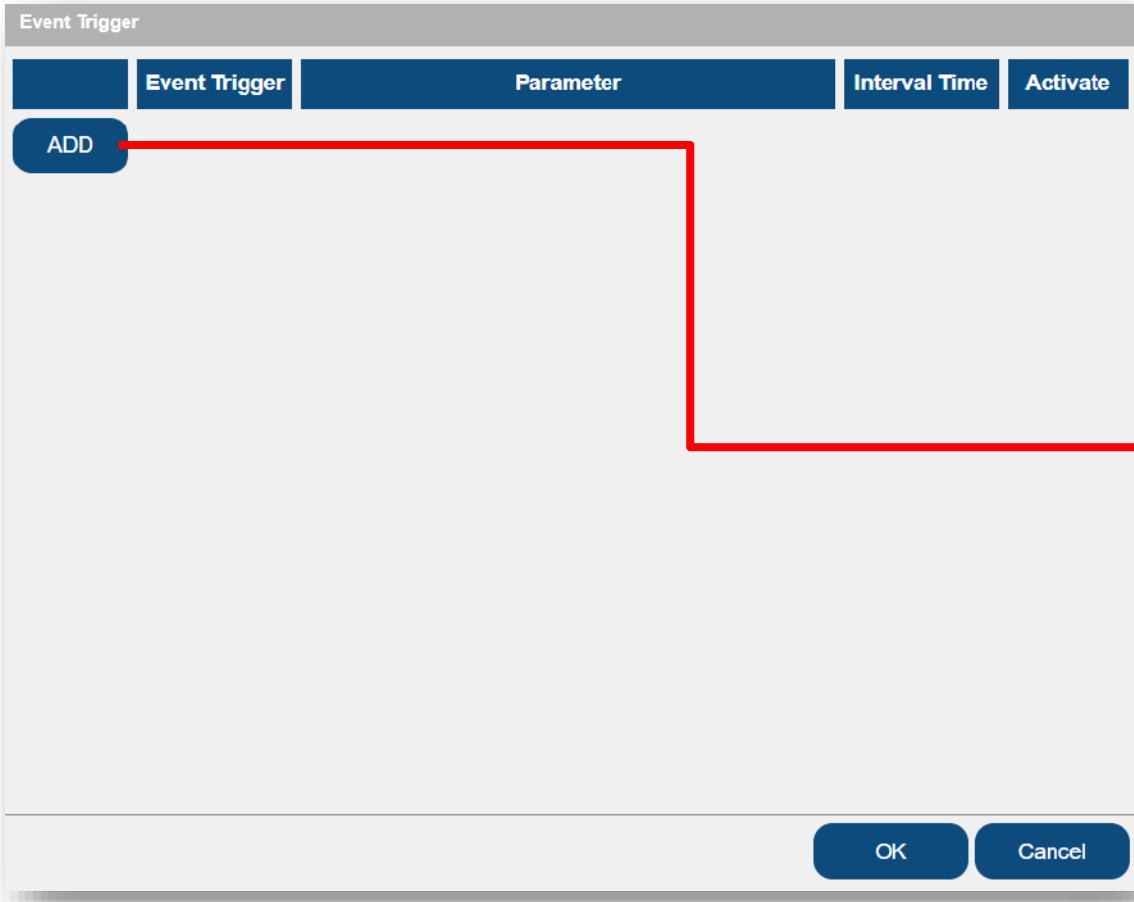
3. 設定事件觸發 – WeChat : 點擊增加 → 事件觸發選擇傳送微信、設定間隔時間、連結名稱選擇 WeChat 連線、選擇是否送圖、輸入模板ID → 點擊確定



# 事件觸發

- 事件觸發

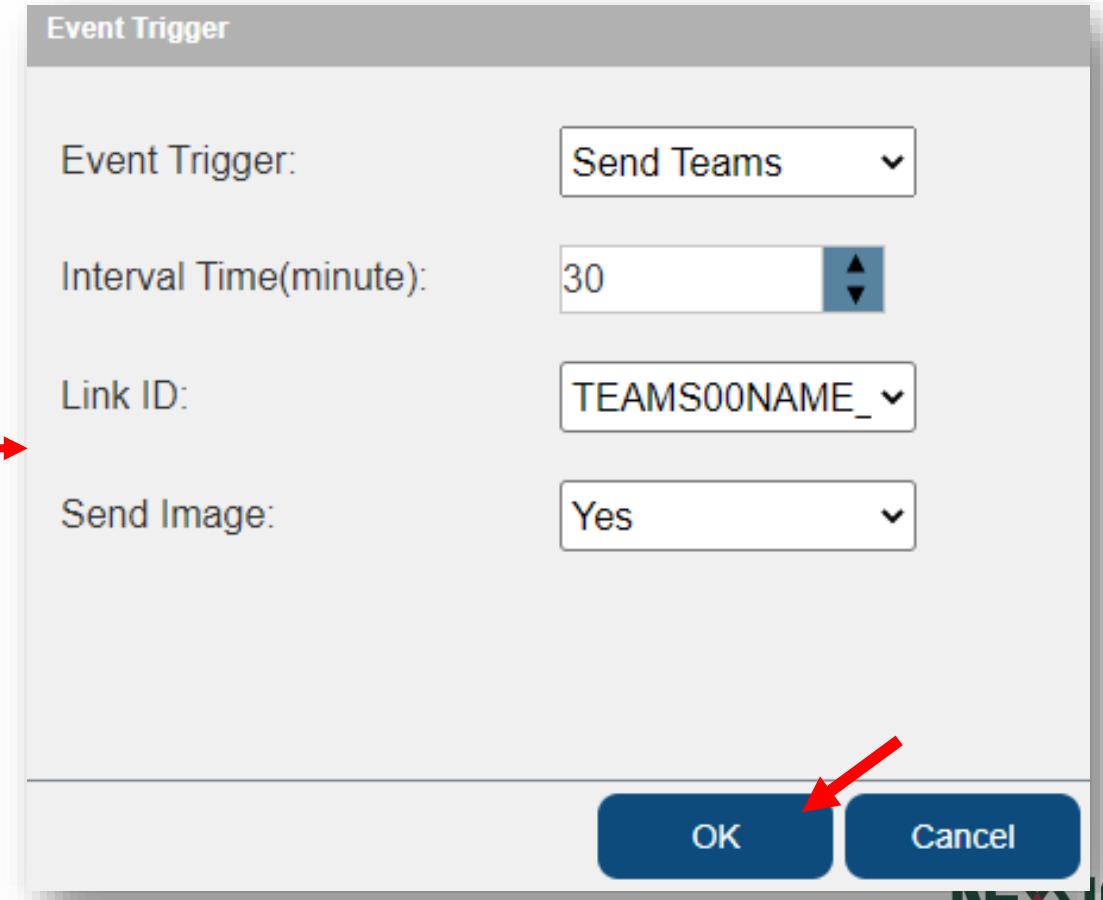
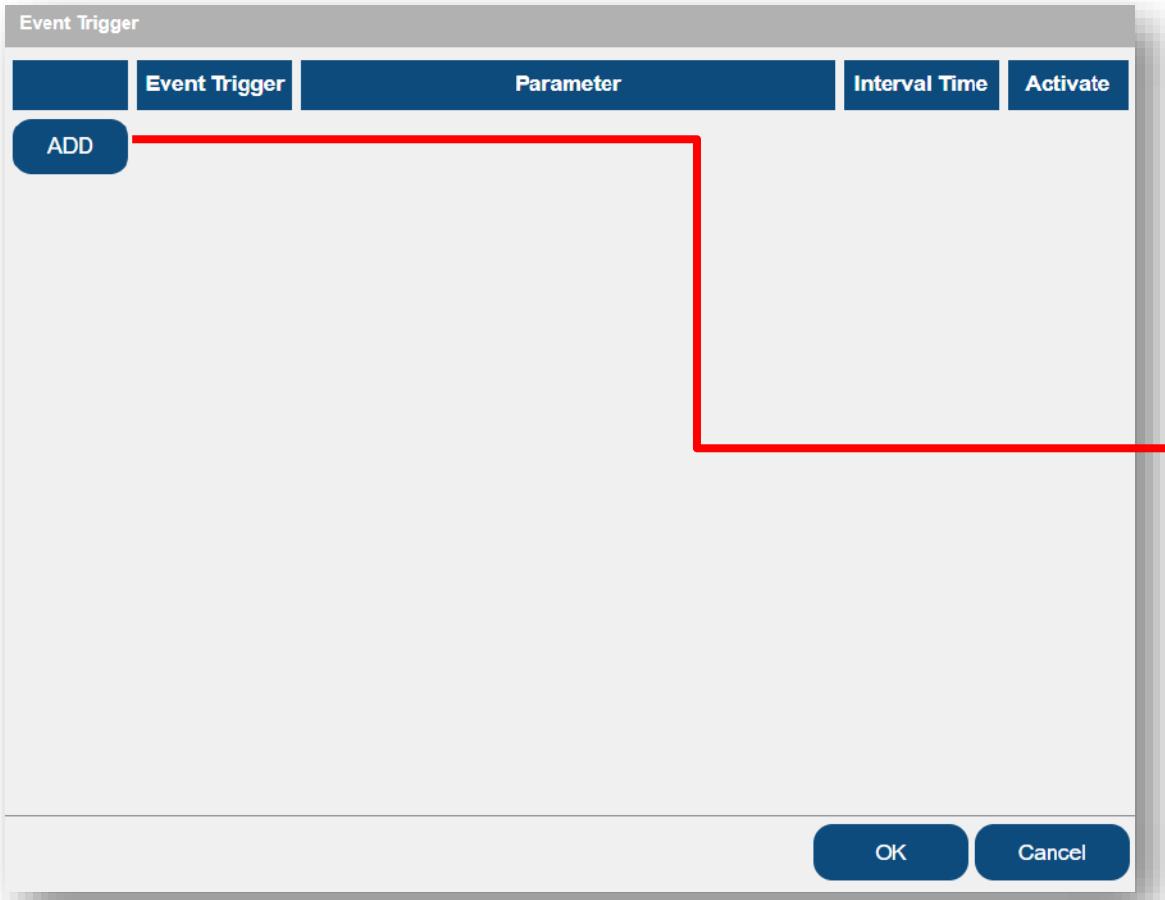
3. 設定事件觸發 – WeChat\_P : 點擊增加 → 事件觸發選擇傳送微信(個人)、設定間隔時間、連結名稱選擇 WeChat 連線、選擇是否送圖、輸入模板ID、輸入用戶微信號 → 點擊確定



# 事件觸發

- 事件觸發

3. 設定事件觸發 – Teams : 點擊增加 → 事件觸發選擇傳送Teams、設定間隔時間、連結名稱  
選擇 Teams 連線 → 點擊確定

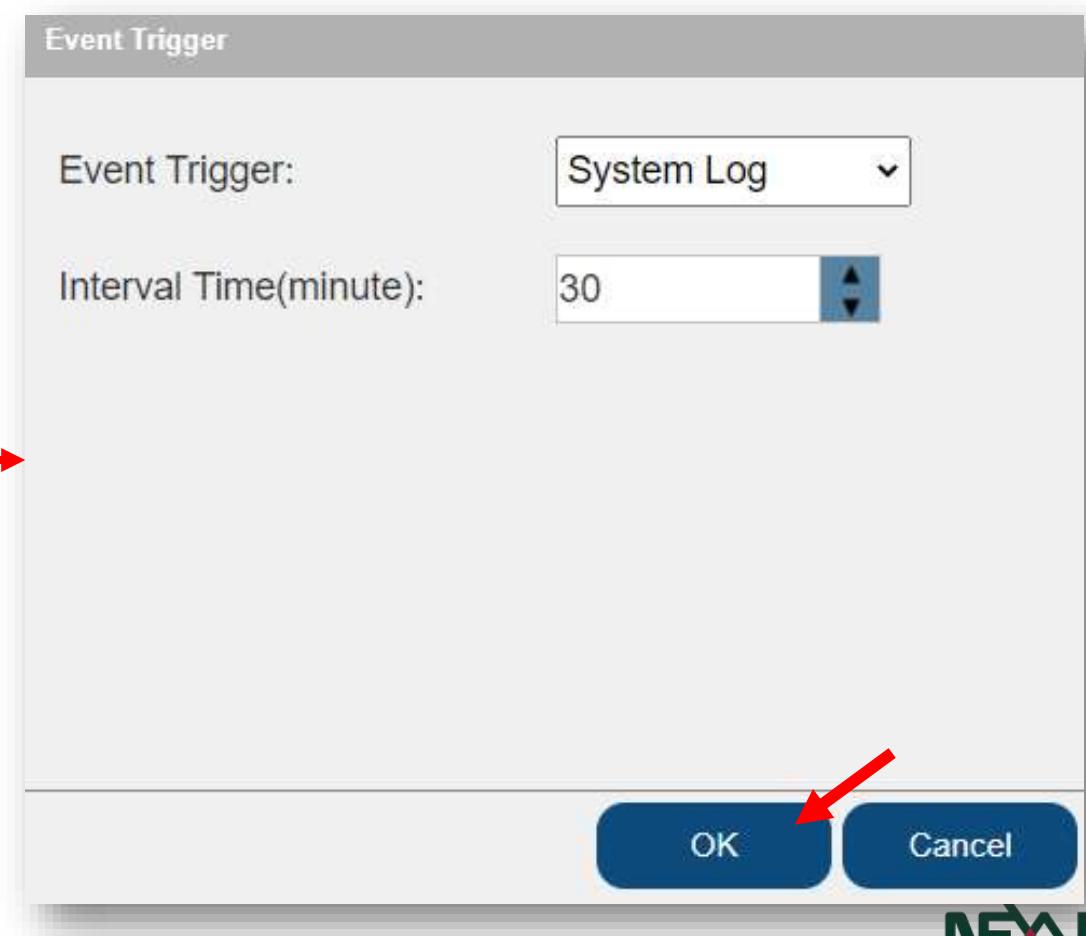
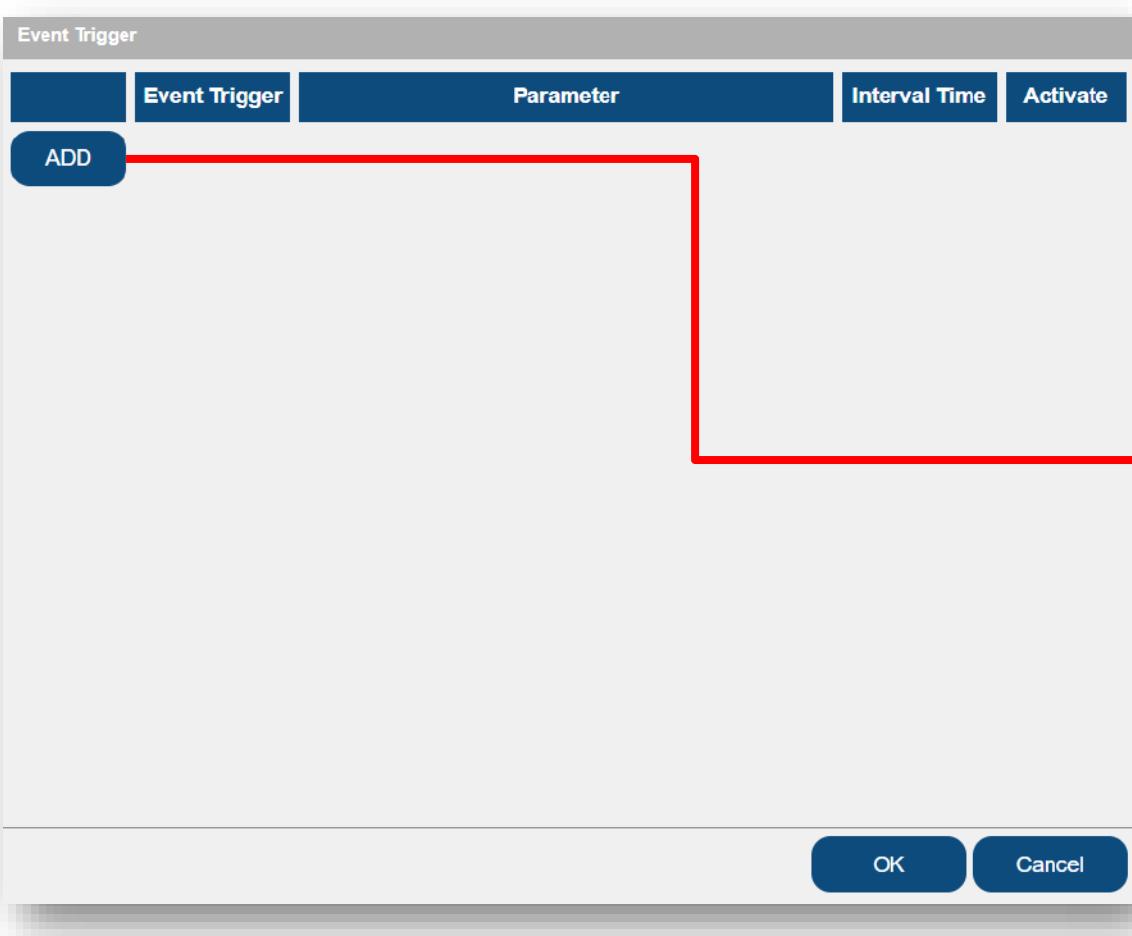


# 事件觸發

- 事件觸發

3. 設定事件觸發 – System Log : 點擊增加 → 事件觸發選擇系統日誌、設定間隔時間

→ 點擊確定

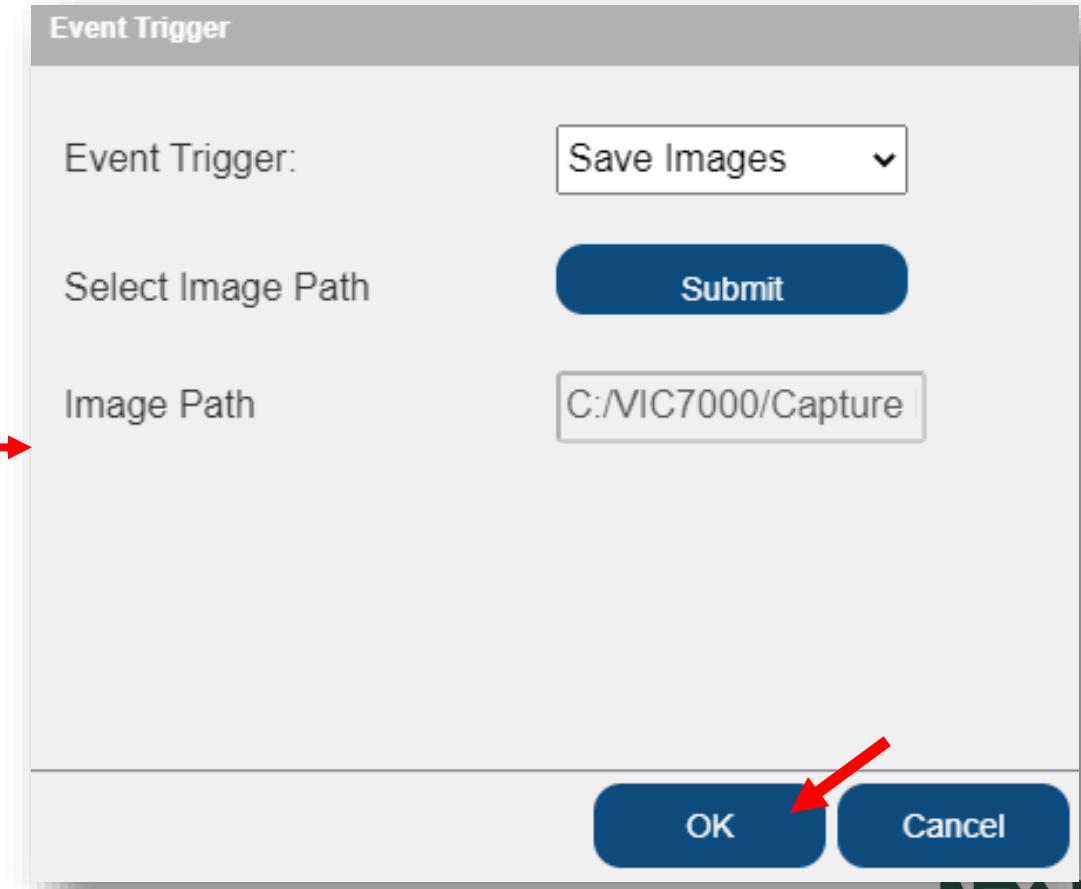
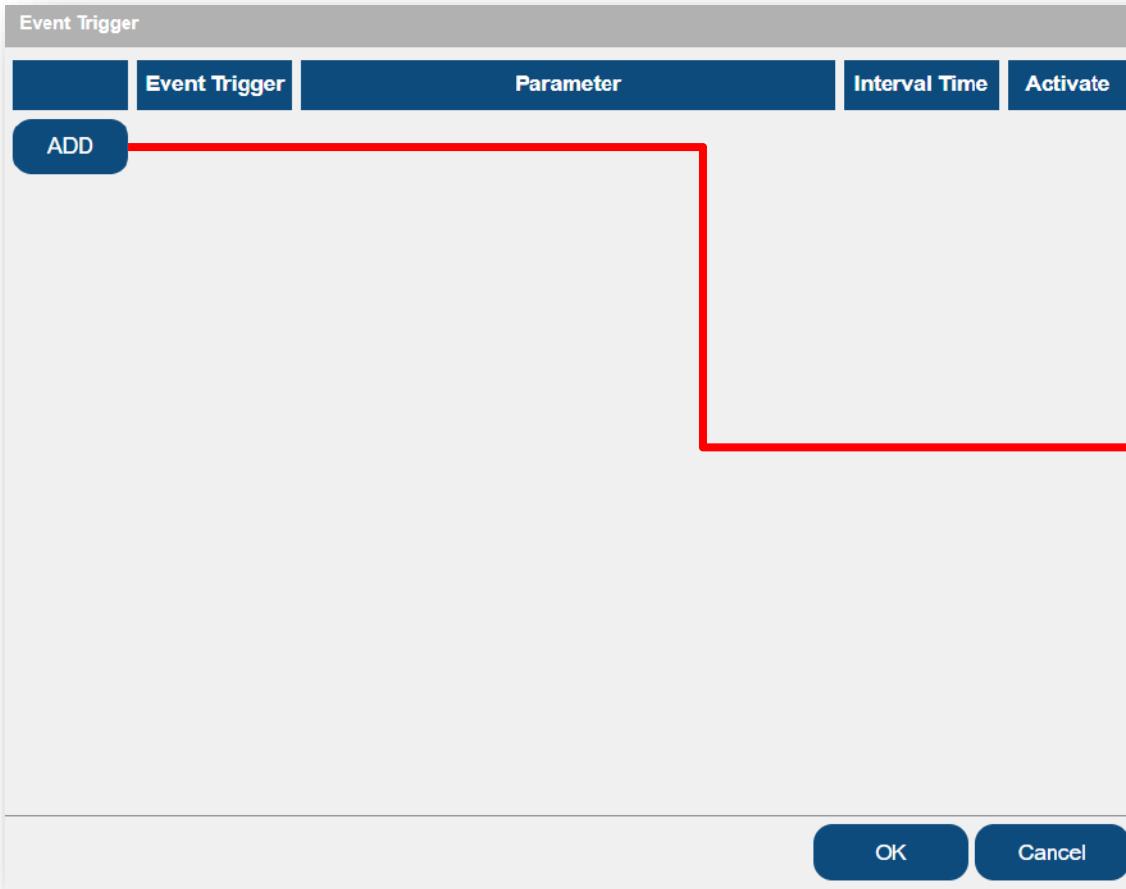


# 事件觸發

- 事件觸發

3. 設定事件觸發 – Save Images : 點擊增加 → 事件觸發選擇儲存影像圖片、設定圖檔路徑

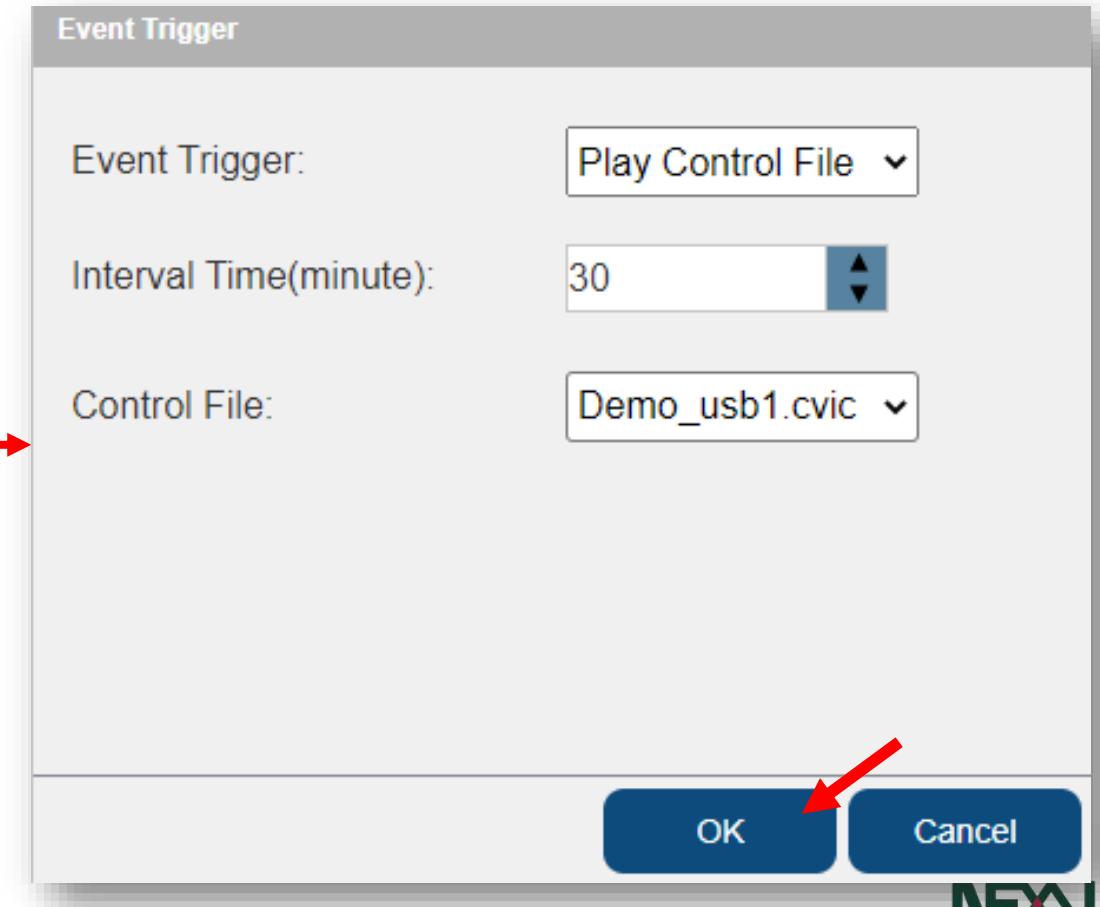
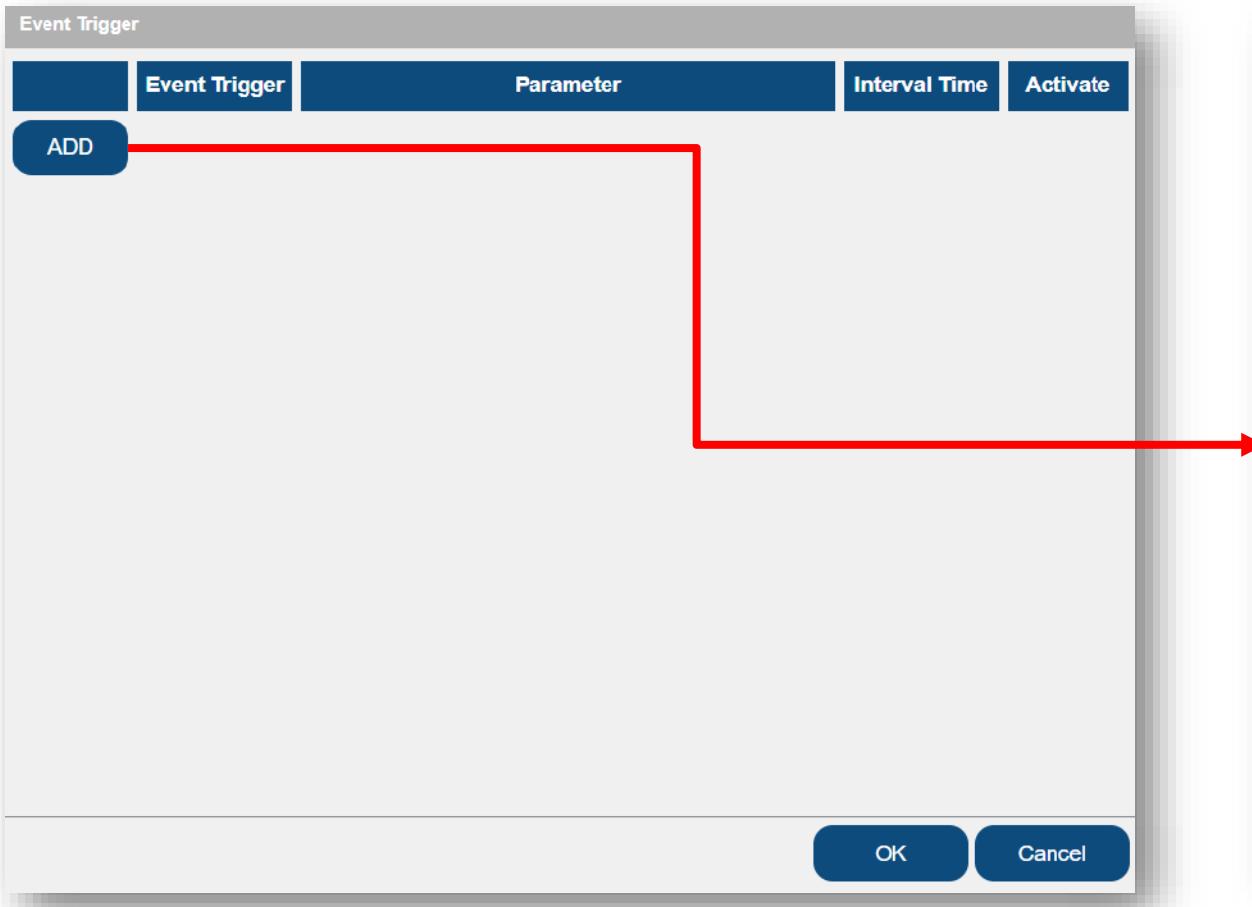
→ 點擊確定



# 事件觸發

- 事件觸發

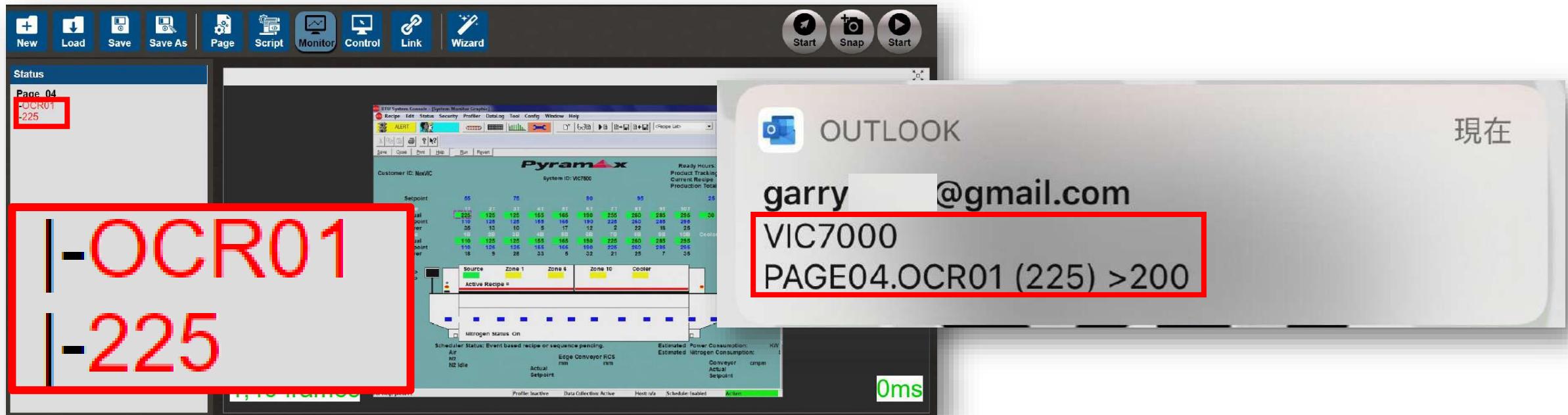
3. 設定事件觸發 – Play Control File : 點擊增加 → 事件觸發選擇播放控制檔、設定間隔時間、選擇要播放的控制檔 → 點擊確定



# 事件觸發

- 結果確認

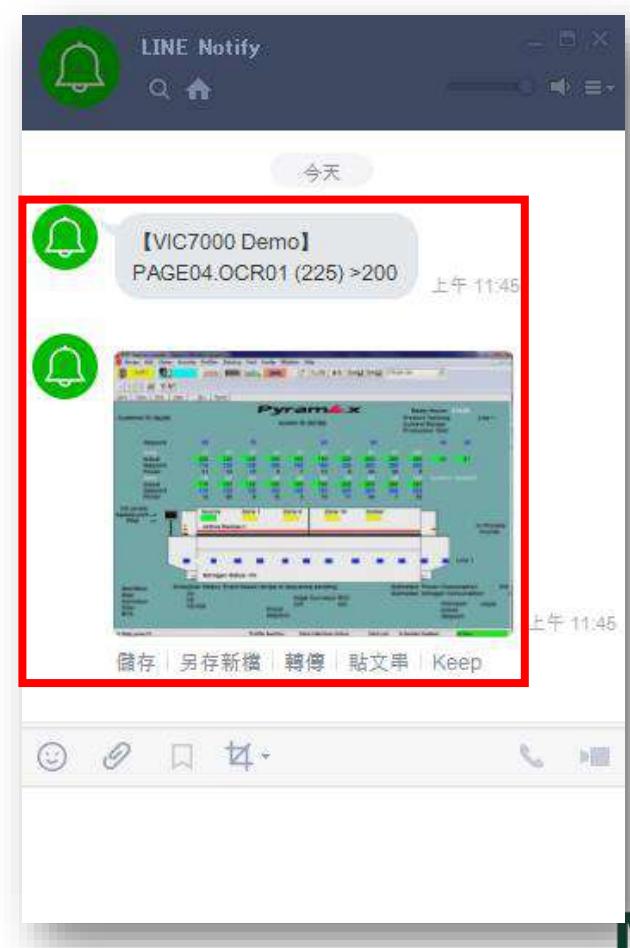
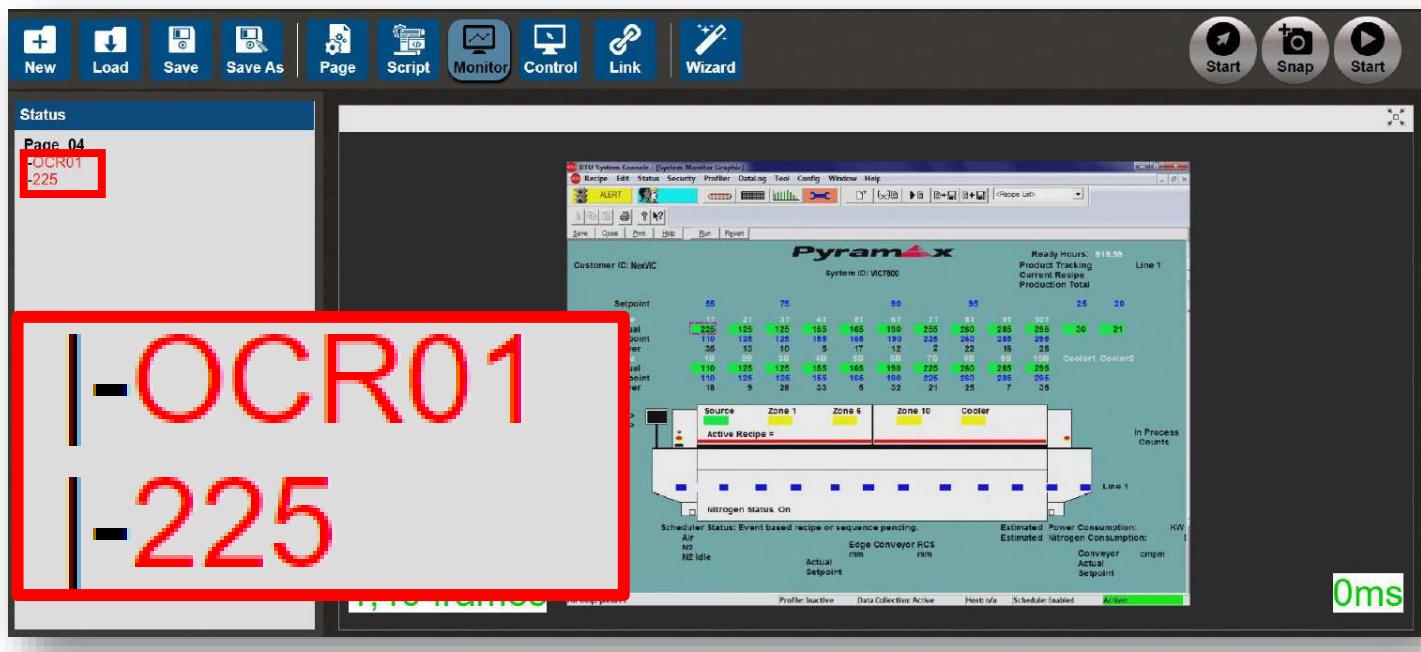
- 模擬確認 – Email：進入監視頁面 → 點擊觸發 → 條件成立時，檢查是否有收到通知  
條件成立時，該辨識工具會以紅字顯示



# 事件觸發

- 結果確認

- 模擬確認 – LINE Notify：進入監視頁面 → 點擊觸發 → 條件成立時，檢查是否有收到通知
- 條件成立時，該辨識工具會以紅字顯示



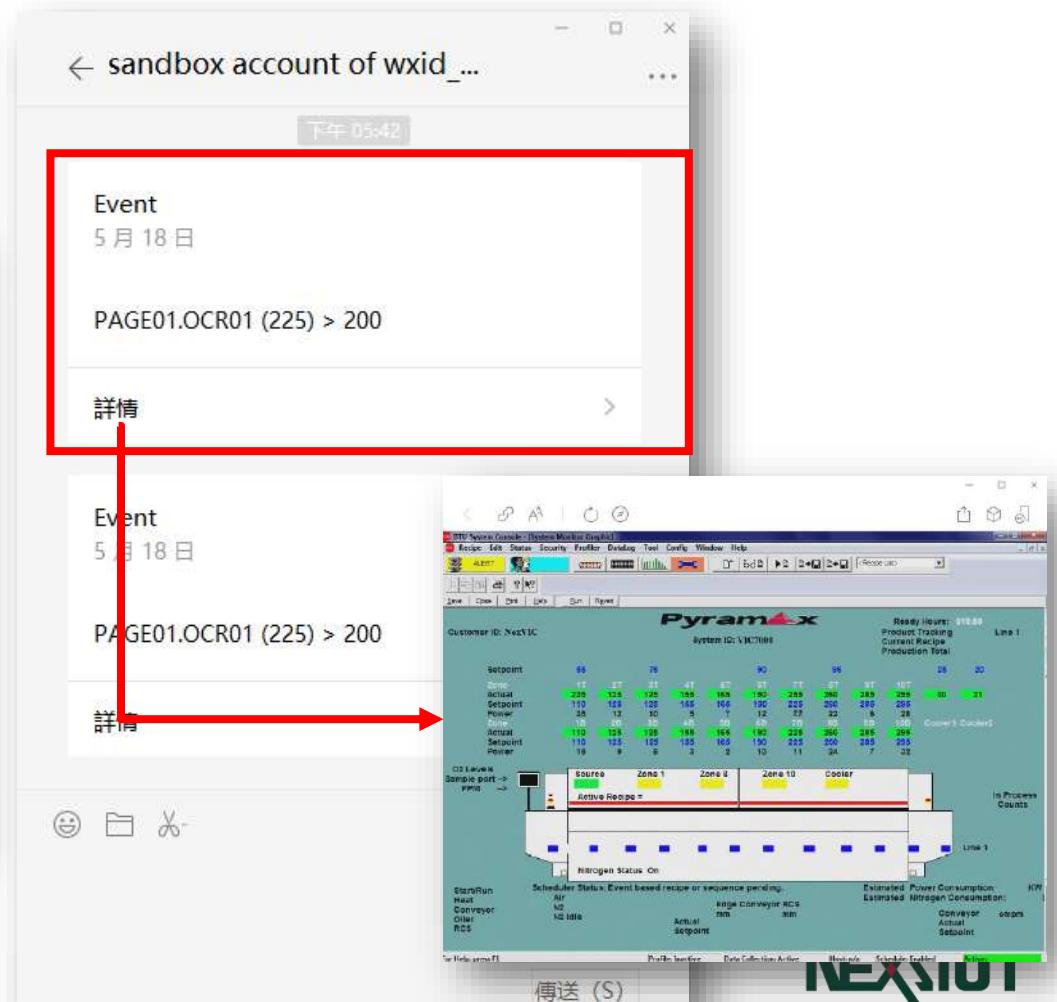
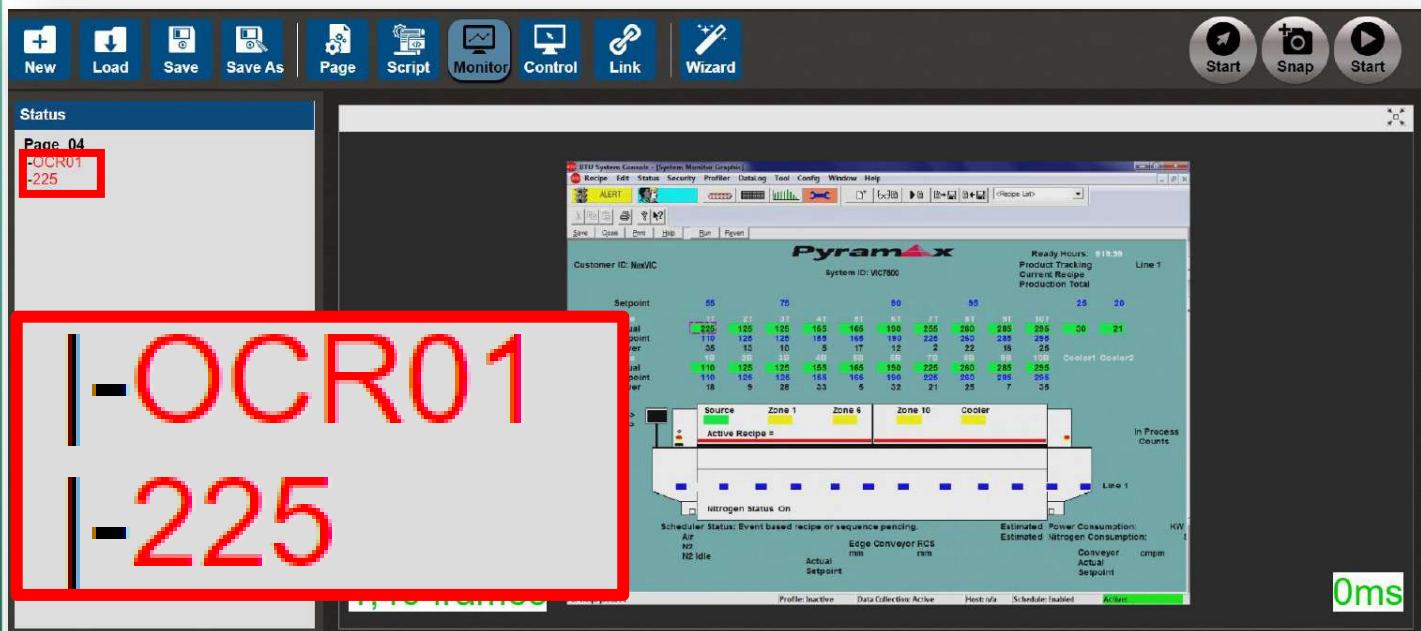
# 事件觸發

## • 結果確認

### 4. 模擬確認 – WeChat、WeChat\_P：進入監視頁面 → 點擊觸發 →

條件成立時，檢查是否有收到通知

條件成立時，該辨識工具會以紅字顯示



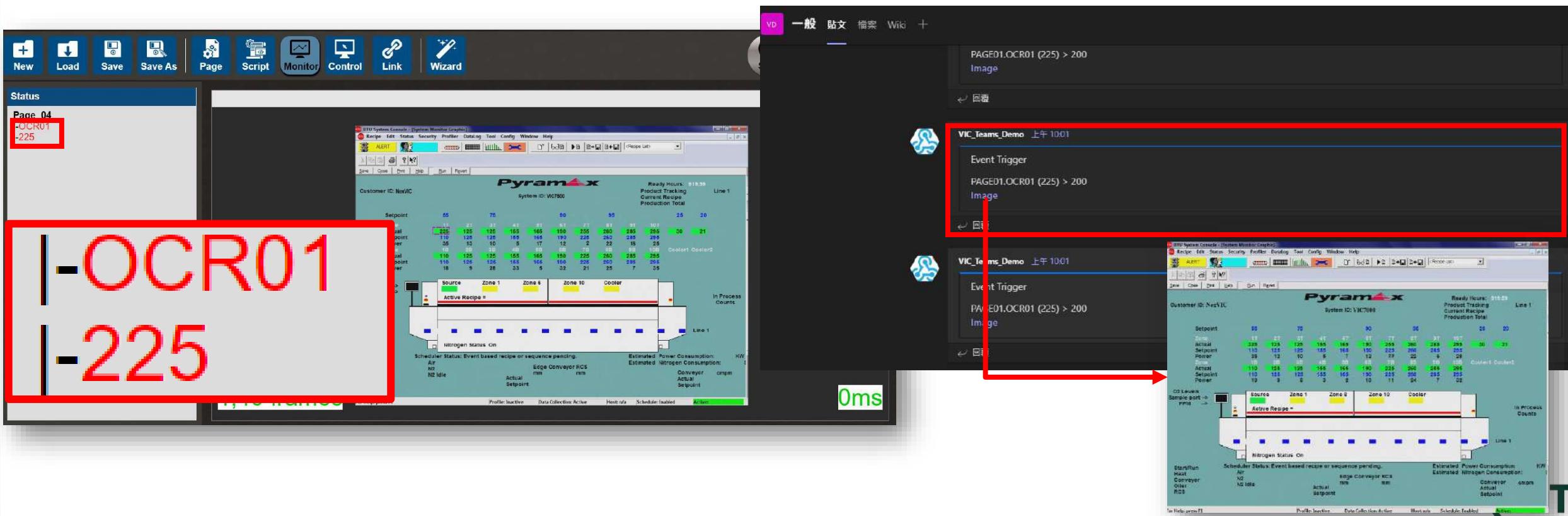
# 事件觸發

- 結果確認

4. 模擬確認 – Teams: 進入監視頁面 → 點擊觸發 → 條件成立時，檢

查是否有收到通知

條件成立時，該辨識工具會以紅字顯示



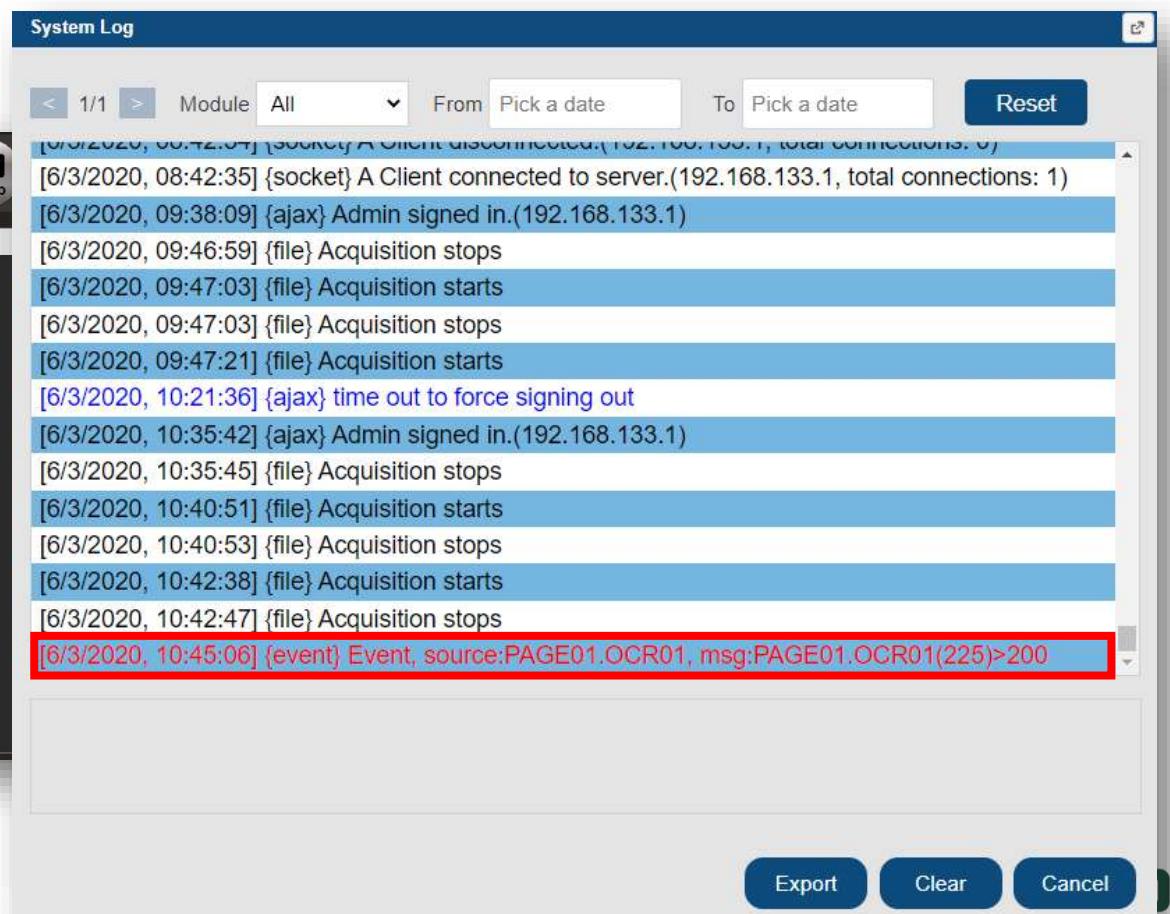
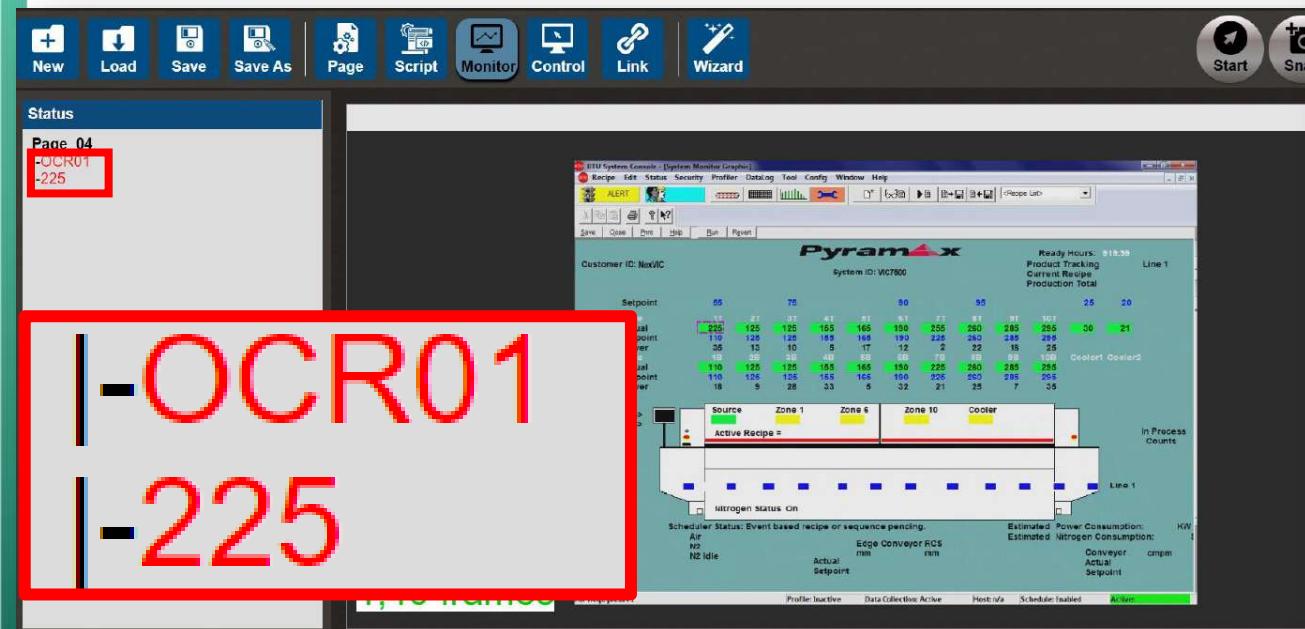
# 事件觸發

- 結果確認

## 4. 模擬確認 – System Log

條件成立時，檢查是否有記錄資訊在系統日誌中

條件成立時，該辨識工具會以紅字顯示



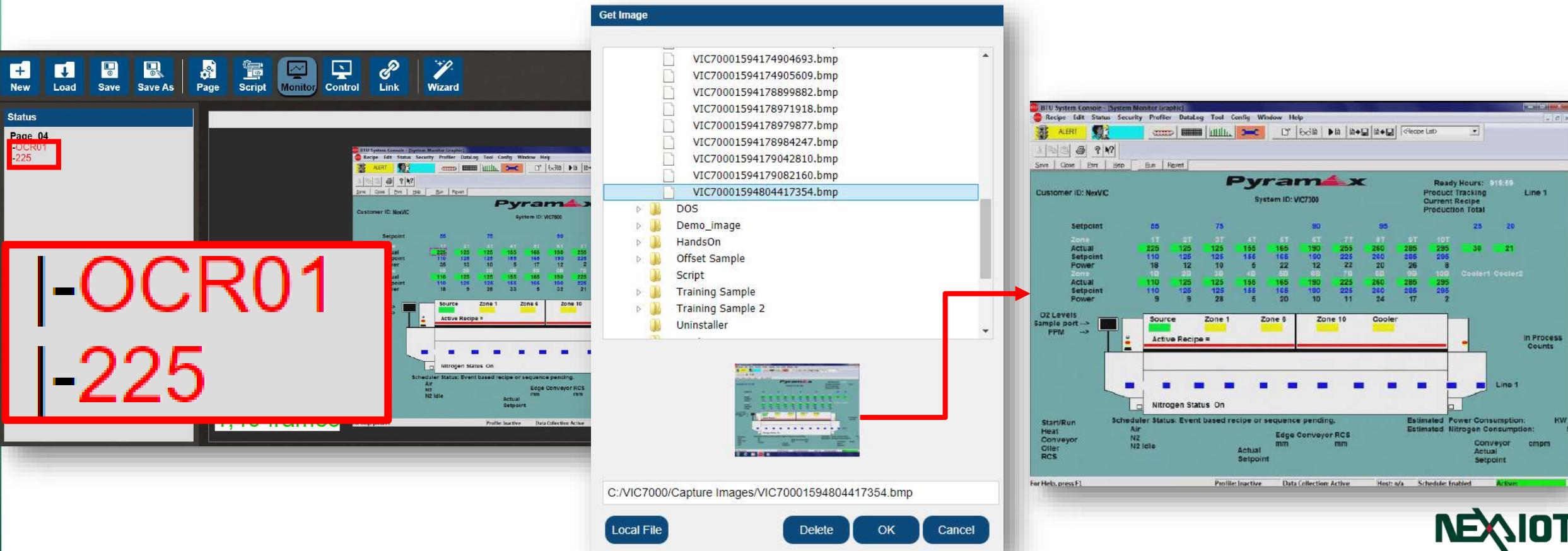
# 事件觸發

- 結果確認

4. 模擬確認 – Save Images : 進入監視頁面 → 點擊觸發 →

條件成立時，檢查是否有儲存下來的圖片

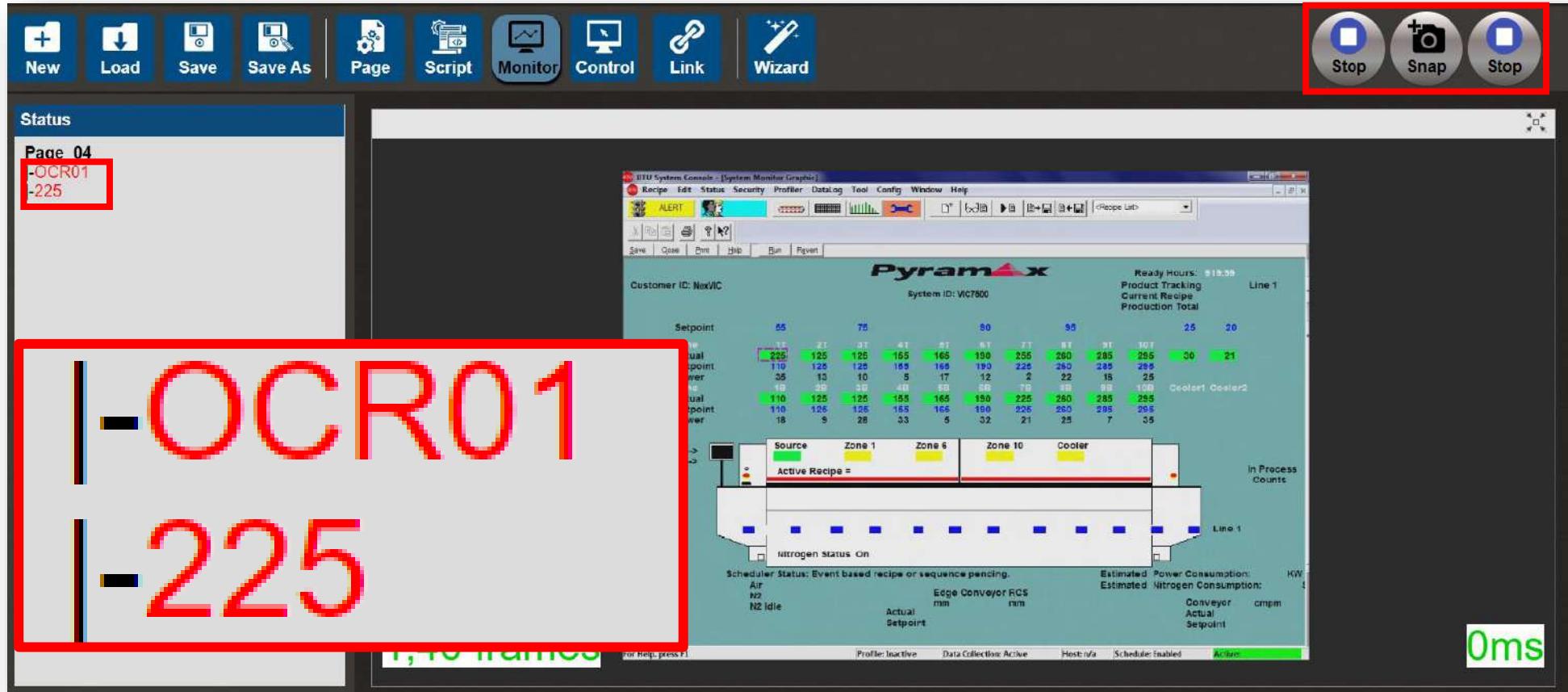
條件成立時，該辨識工具會以紅字顯示



# 事件觸發

- 結果確認

- 模擬確認 – Play Control File：進入監視頁面 → 點擊啟動專案 → 條件成立時，檢查是否有播放設定的控制檔  
條件成立時，該辨識工具會以紅字顯示



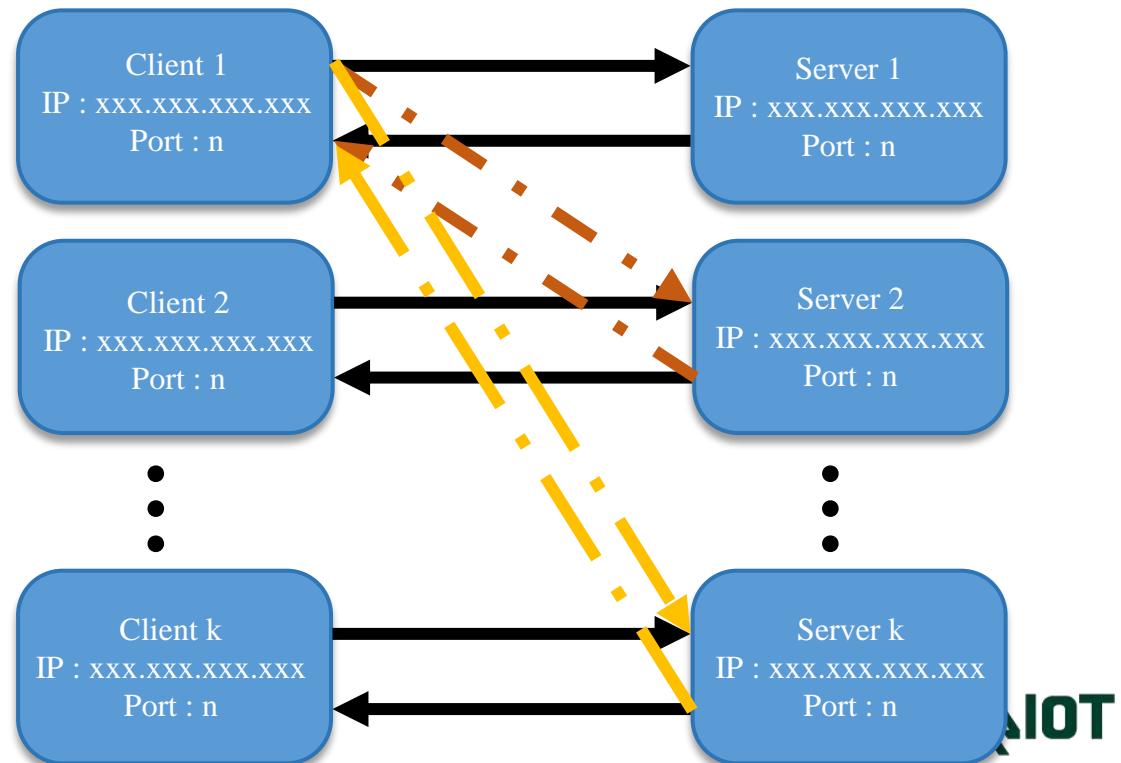
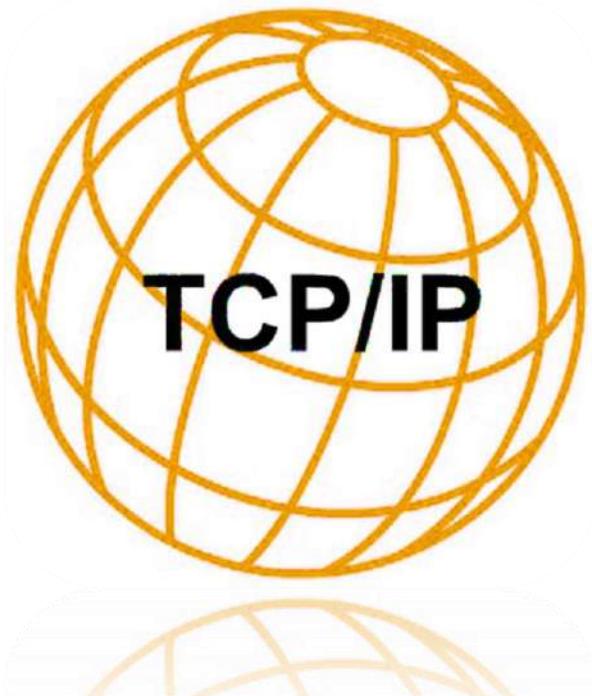
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

- **TCP/IP**

TCP/IP 是網際網路協議套組的簡稱，即使是不同電腦或作業環境，都可以透過此通訊協定互通訊息，包括兩個核心協議，**TCP(傳輸控制協定)**和**IP(網際網路協定)**。

TCP/IP 提供點對點的連結機制，分為 **Server** 和 **Client** 兩種角色，可有多個 Client 和 Server，透過IP位址和 TCP 的 **port** 來進行連線，VIC7000 為 **Server**。



- **TCP/IP：資料格式**

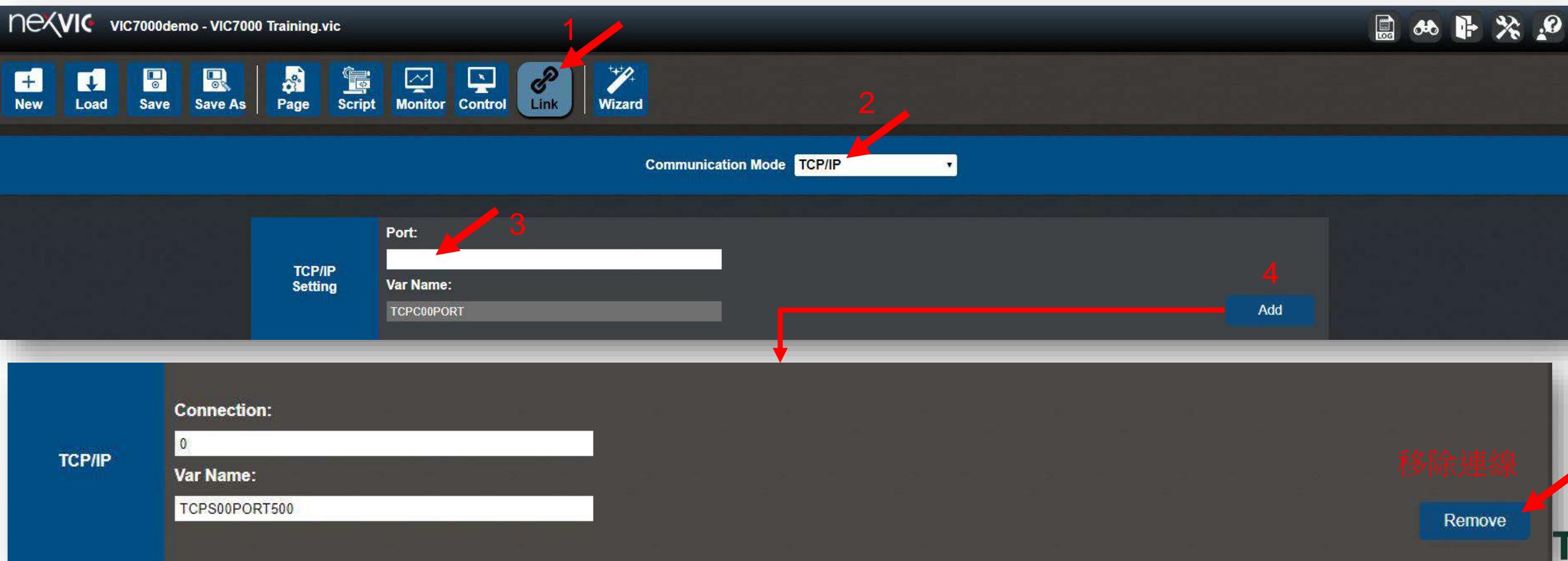
1. 標頭：v，長度為 1 byte
2. 資料長度：辨識資料長度，單位為 1 個 ASCII 碼 (1 byte)，長度為 4 bytes (Int32)
3. 辨識資料內容：辨識資料，長度為資料長度



# 外部連線

- **TCP/IP** : 連線設定

1. 進入連結頁面：從管理模式進入
2. 選擇 TCP/IP
3. 輸入通訊埠
4. 點擊新增

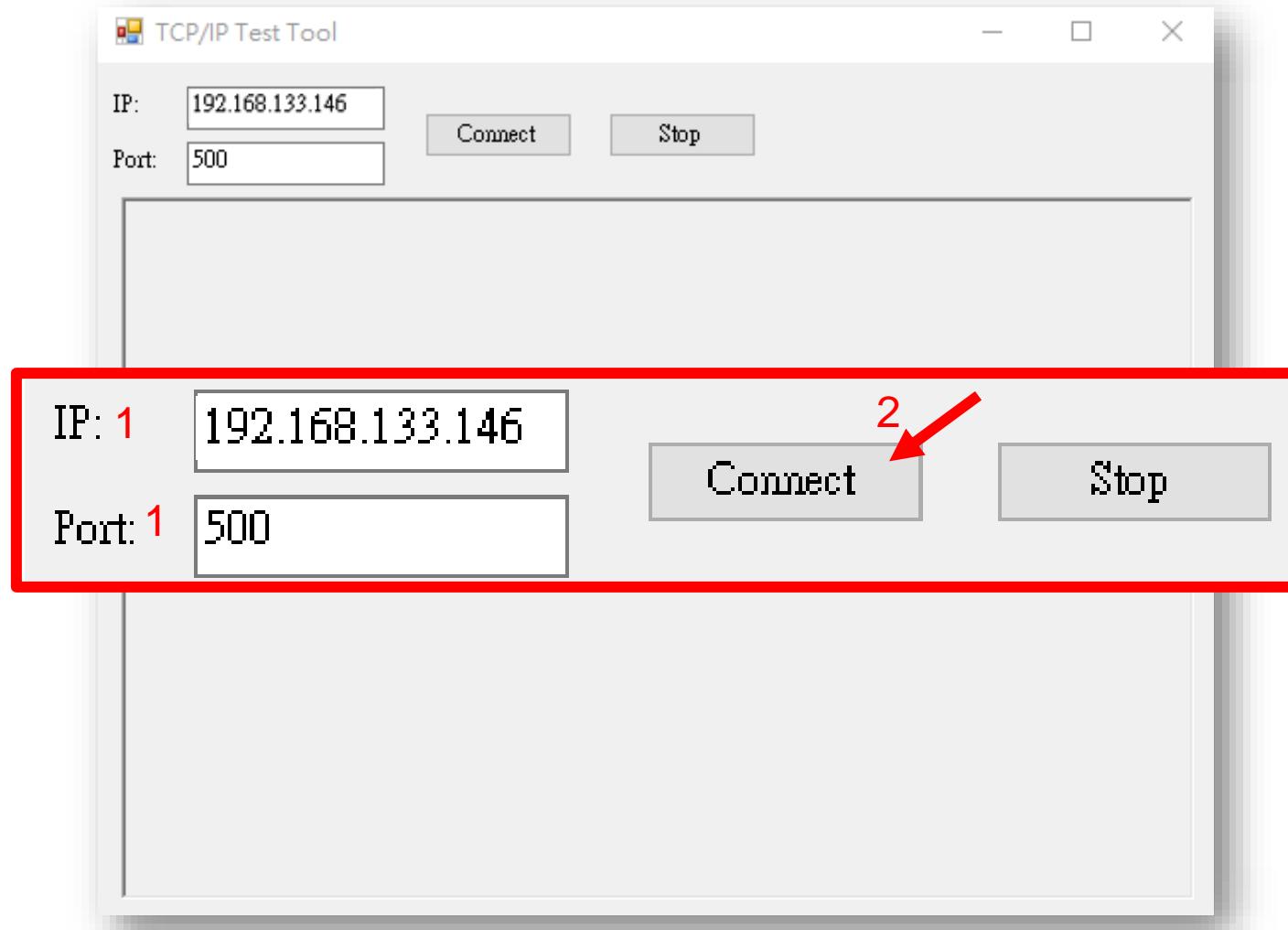


# 外部連線

- **TCP/IP** : 連線測試 : 使用 TCP 測試軟體

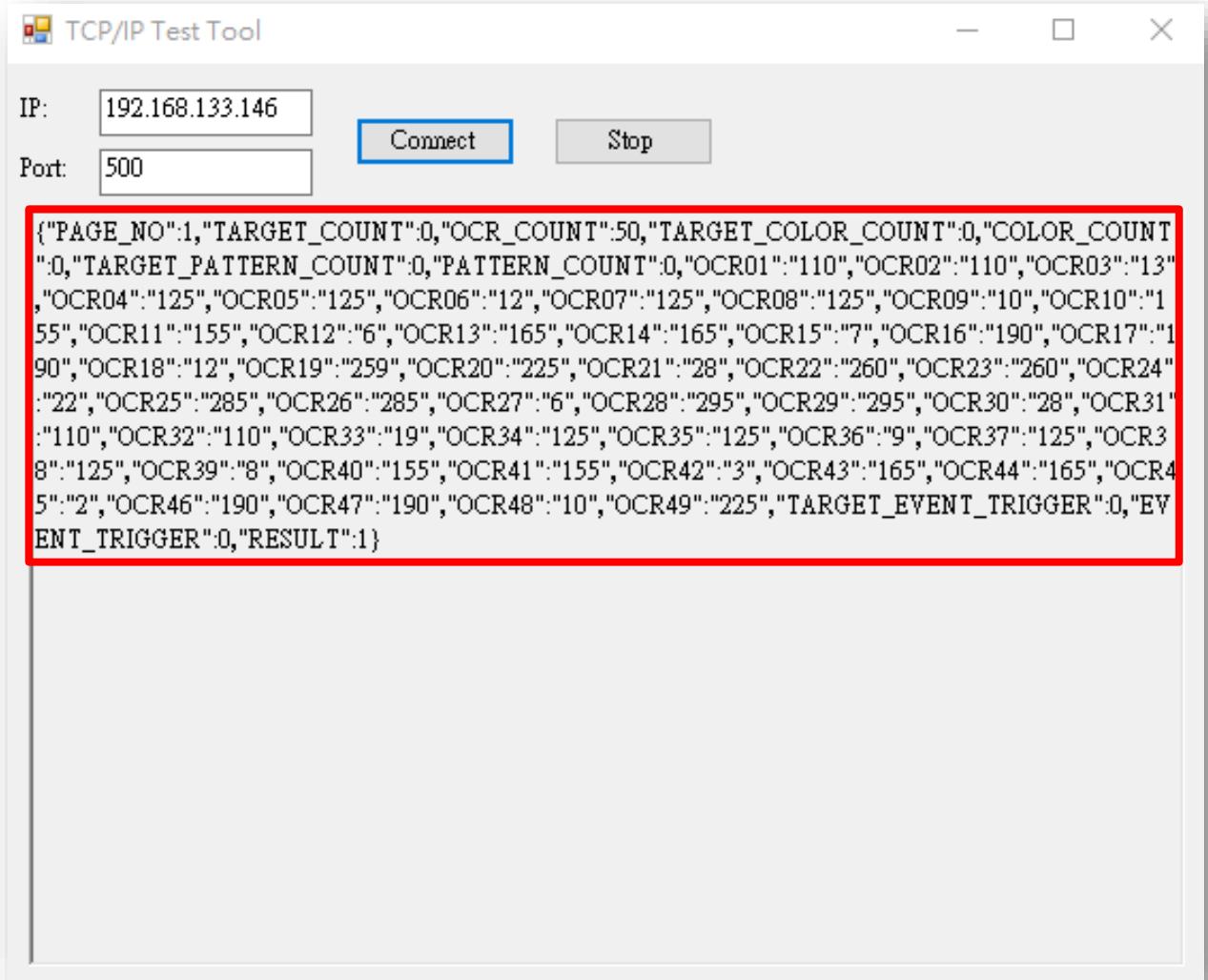
1. 輸入 IP 和 Port

2. 點擊 Connect



- **TCP/IP** : 連線測試

3. 檢查回傳值 : 回傳資料類型為 ASCII , 解碼後為 JSON 字串



- **Modbus TCP**

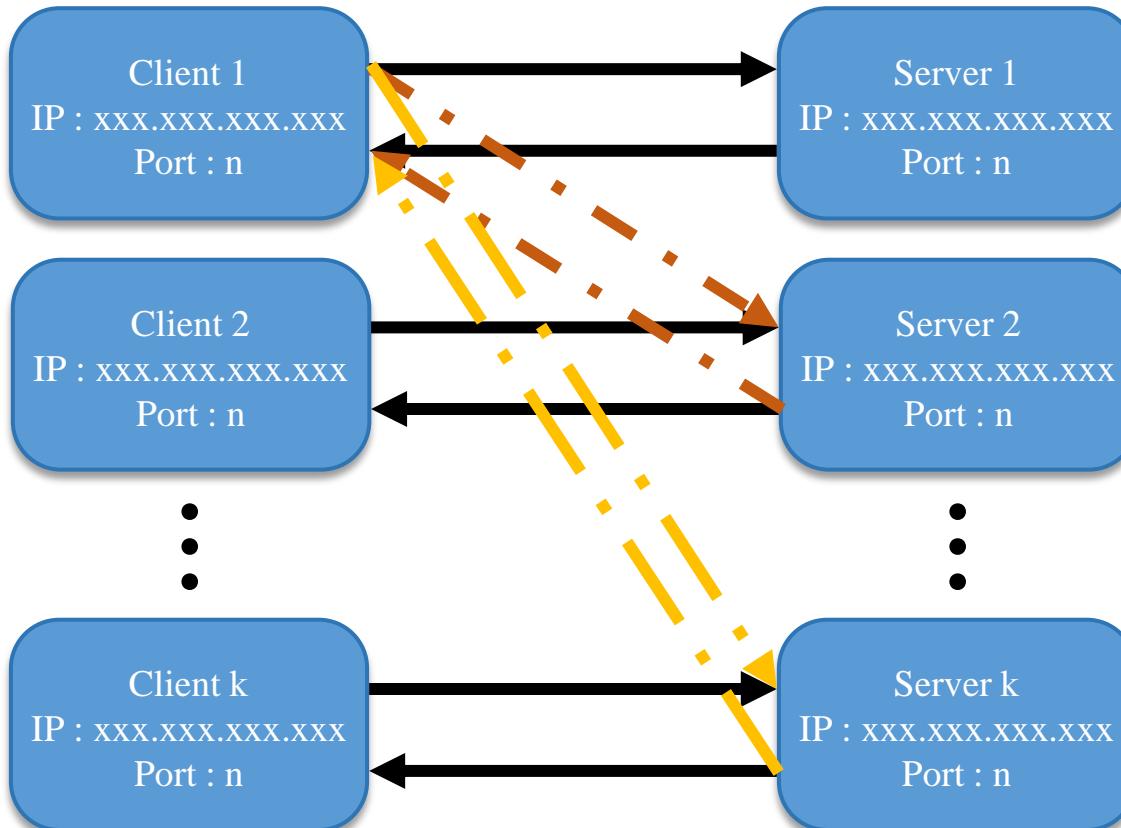
Modbus 是 Modicon 在1979年為了使用PLC而發表的協議，是全球第一個真正用於工業現場的匯流排協定。

Modbus 現在已成為工業領域通信協議的標準，也是工業電子設備之間最常用的連接方式，常用於和 PLC 及 SCADA 軟體的連線，經由串列網路、乙太網路等和各設備連線。



- **Modbus TCP : 資料欄位**

- 使用 Modbus TCP , VIC7000 為 Server
- 系統資訊和辨識資料存放在 Input Status 和 Input Register
- Coil Status 和 Holding Register 供外部寫入和控制 VIC



Address	Name	Data Type	Type
0X0001 ~ 0X9999	Coil Status	Bit	R/W
1X0001 ~ 1X9999	Input Status	Bit	R
3X0001 ~ 3X9999	Input Register	Word	R
4X0001 ~ 4X9999	Holding Register	Word	R/W

- **Modbus TCP** : 資料欄位

- Little – Endian

從低位元開始填入資料

OCR01 Result

A	B	C	\0
---	---	---	----

Modbus Input Register

42	41	0	43
3x12626		3x12627	

## • Address Mapping – Input Status : System Bit

存放系統資訊，輸出格式為 Boolean，站號為 1，單一長度為 1 個 Bit

Input Status (1X)	
Address	Content
10001	程式是否運行中， 1 = 運行中，0 = 停止
10002	是否有使用者登入， 1 = 有人登入，0 = 無人登入
10003	系統硬碟容量 5G 檢測， 1 = 低於 5G，0 = 高於 5G
10004	系統硬碟容量 10G 檢測， 1 = 低於 10G，0 = 高於 10G
10005	是否停止存圖， 1 = 停止，0 = 運行中
10006 (Bit 6)	OCR辨識結果 0 = 辨識成功，1 = NG，2 = 比對失敗
10007 (Bit 7)	
10009	序列埠控制是否被啟用， 1 = 取用，0 = 停用
10010	序列埠目前狀態， 1 = 連線，0 = 斷線

10017	資料硬碟容量 10G 檢測， 1 = 低於 5G，0 = 高於 5G
10018	資料硬碟容量 20G 檢測， 1 = 低於 10G，0 = 高於 10G
10019	錄製硬碟容量 1G 檢測， 1 = 低於 1G，0 = 高於 1G
10020	錄製硬碟容量 100G 檢測， 1 = 低於 100G，0 = 高於 100G
10021	額外儲存路徑容量 1G 檢測 1 = 低於 1G，0 = 高於 1G
10022	目前錄製狀態 1 = 錄製中，0 = 停止錄製

Bit 7	Bit 6	Sum	Result
0	0	0	辨識成功
0	1	1	辨識 NG
1	0	2	比對失敗

- **Address Mapping – Input Register (Int)** : PAGE\_NO、TARGET\_COUNT、OCR\_COUNT、TARGET\_COLOR\_COUNT、COLOR\_COUNT、TARGET\_PATTERN\_COUNT、PATTERN\_COUNT、RESULT、TARGET\_EVENT\_TRIGGER、EVENT\_TRIGGER  
存放當前頁面編號、目標值數量、辨識數量、辨識結果和是否事件觸發，站號為1  
輸出格式為 Int32，單一長度為 2 個 Word

RESULT :

0 = 辨識成功

1 = NG

2 = 比對失敗

Input Register (3X)	
Address Range	Content
312501 ~ 312502	PAGE_NO
312503 ~ 312504	TARGET_COUNT
312505 ~ 312506	OCR_COUNT
312507 ~ 312508	TARGET_COLOR_COUNT
312509 ~ 312510	COLOR_COUNT
312511 ~ 312512	TARGET_PATTERN_COUNT
312513 ~ 312514	PATTERN_COUNT
312515 ~ 312516	RESULT
312601 ~ 312602	TARGET_EVENT_TRIGGER
312603 ~ 312604	EVENT_TRIGGER

- **Address Mapping – Input Register (String) : TARGET、OCR**

存放目標值和 OCR 結果，站號為1，輸出格式為 ASCII，單一長度為 125 個 Word

Input Register (3X)	
Address Range	Content
30126 ~ 30250	Target01 (String)
30251 ~ 30375	Target02 (String)
30376 ~ 30500	Target03 (String)
30501 ~ 30625	Target04 (String)
30626 ~ 30750	Target05 (String)

Input Register (3X)	
Address Range	Content
312626 ~ 318875	OCR01 ~ 50 (String)

- **Address Mapping – Input Register (Int) : TARGET、OCR**

若目標值和 OCR 內辨識值為數字，存放輸出相對應整數，站號為1

數值格式為 Int32，單一長度為 2 個 Word

Input Register (3X)	
Address Range	Content
350003 ~ 350004	Target01 (Int)
350005 ~ 350006	Target02 (Int)
350007 ~ 350008	Target03 (Int)
350009 ~ 350010	Target04 (Int)
350010 ~ 350012	Target05 (Int)

Input Register (3X)	
Address Range	Content
350203 ~ 350302	OCR01 ~ 50 (Int)

- **Address Mapping – Input Register : TARGET\_COLOR、COLOR**

存放顏色目標值和顏色比對結果

站號為1，數值格式為 Int8

Input Register (3X)	
Address Range	Content
350803 ~ 350804	Target_color01
350805 ~ 350806	Target_color02
350807 ~ 350808	Target_color03
350809 ~ 350810	Target_color04
350811 ~ 350812	Target_color05

Input Register (3X)	
Address Range	Content
350803 ~ 350862	Target_color01 ~ 30
351003 ~ 351102	Color01 ~ 50

COLOR01 Result



Modbus Input Register



3x51003

3x51004

COLOR = [M, R, G, B]

M = 比較結果

(0 = 比對失敗 1 = 比對成功)

R = 紅色值 (0 ~ 255)

G = 綠色值 (0 ~ 255)

B = 藍色值 (0 ~ 255)

- **Address Mapping – Input Register : TARGET\_PATTERN 、 PATTERN**

存放樣式比對目標值和樣式比對結果，站號為1

數值格式為 Int16 和 Float64

PATTERN = [M, R, S, X, Y, W, H]

M = 比較結果

(0 = 比對失敗 1 = 比對成功)

R = 保留欄位

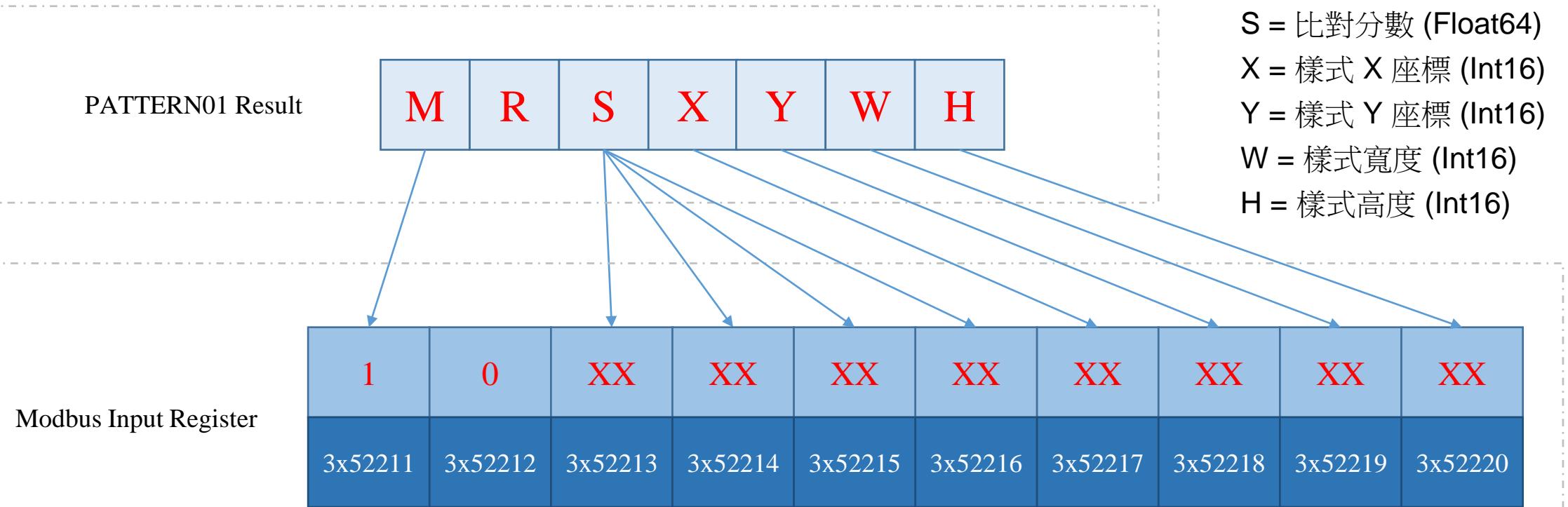
S = 比對分數 (Float64)

X = 樣式 X 座標 (Int16)

Y = 樣式 Y 座標 (Int16)

W = 樣式寬度 (Int16)

H = 樣式高度 (Int16)



- **Address Mapping – Input Register : TARGET\_PATTERN 、 PATTERN**

存放樣式比對目標值和樣式比對結果，站號為1

數值格式為 Int16 和 Float64

Input Register (3X)	
Address Range	Content
351211 ~ 351220	Target_pattern01
351221 ~ 351230	Target_pattern02
351231 ~ 351240	Target_pattern03
351241 ~ 351250	Target_pattern04
351251 ~ 351260	Target_pattern05

Input Register (3X)	
Address Range	Content
351211 ~ 351310	Target_pattern01 ~ 10
352211 ~ 352310	Pattern01 ~ 10

- **Address Mapping – Coil Status : System Control**

控制系統，站號為1，啟停專案、觸發截圖一張需要在沒有 admin 登入時才能使用

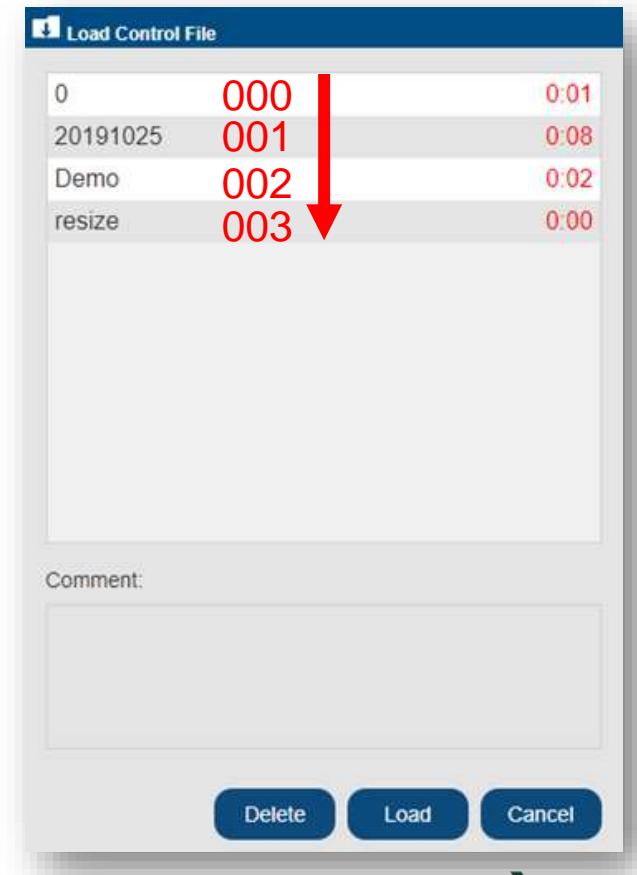
Coil Status (0X)	
Address	Content
00001	啟動專案：1 = 啟動，0 = 停止
00002	運行控制：1 = 啟動，0 = 停止
00003	辨識資料是否寫入資料庫：1 = 寫入，0 = 不寫入
00004	是否觸發錄製事件：1 = 觸發，0 = 不觸發
00005	是否使用記錄器：1 = 使用，0 = 不使用
00006	觸發截圖一張：1 = 截圖
00007	停用控制，1 = 停用，0 = 不停用

- **Address Mapping – Coil Status : System Control**

控制系統，指定當前頁面、執行腳本函示需要在沒有 **admin** 登入時才能使用  
站號為 1

Coil Status (0X)	
Address	Content
00101 ~ 00300	運行編號000 ~ 199 的控制檔：1 = 啟動，0 = 停止
00302 ~ 00331	指定當前頁面，1 = 指定該頁面；00302代表頁面01
00401 ~ 00500	執行腳本函式，傳入啟動的位址：1 = 啟動

OnTriggerScriptFunction(address)  
address = 400 ~ 499

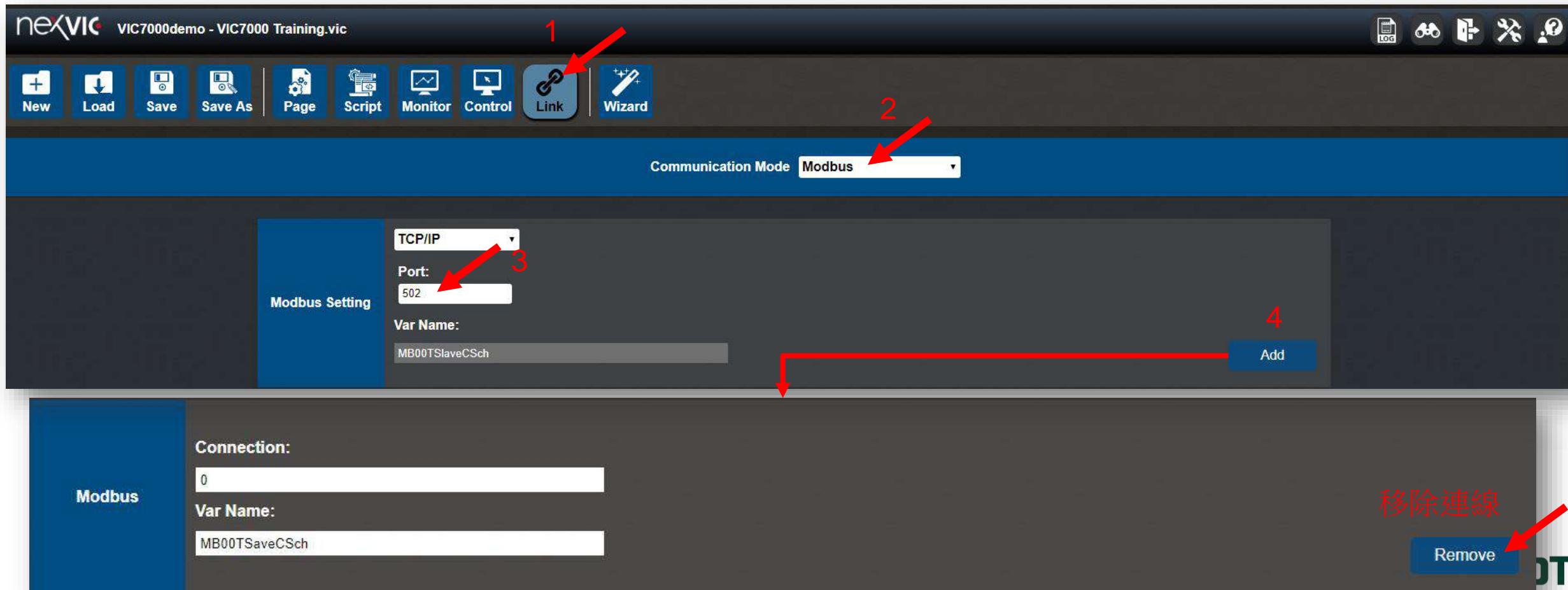


# 外部連線

- **Modbus TCP : 連線設定**

1. 進入連結頁面：從管理模式進入
2. 選擇 Modbus

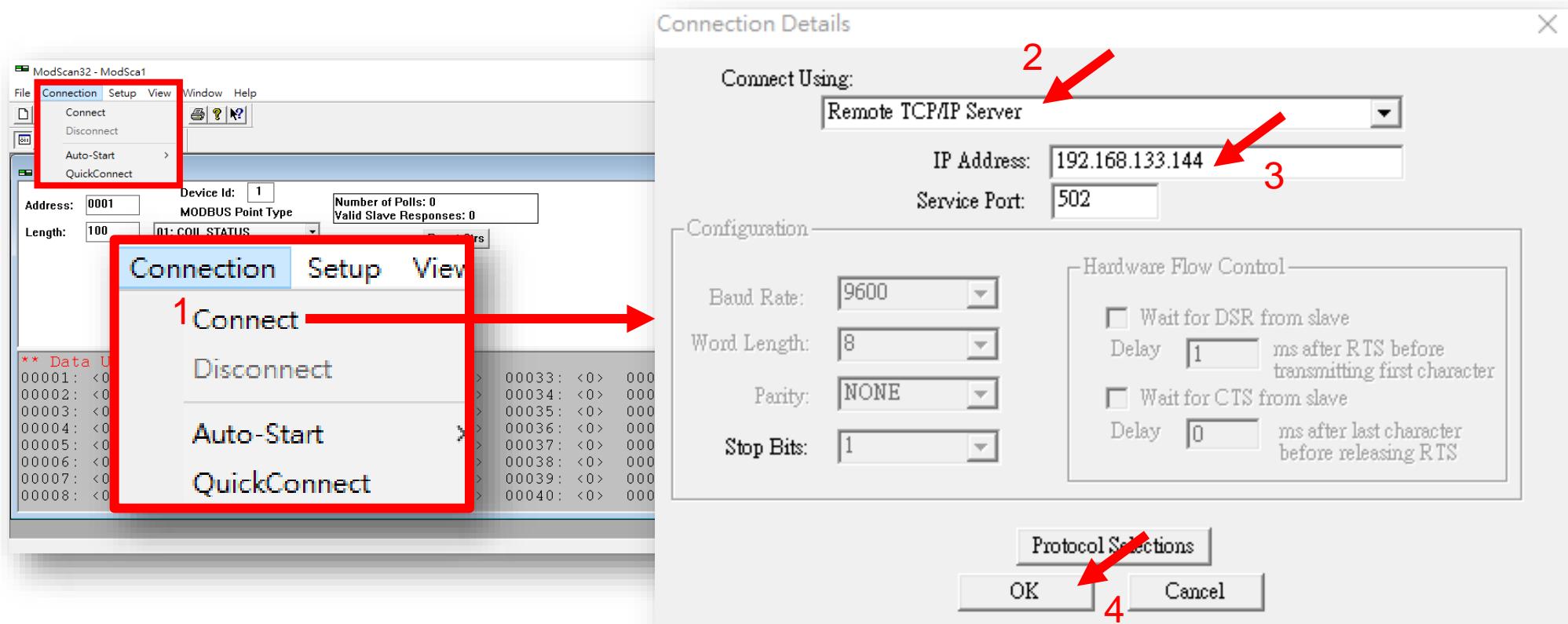
3. 輸入通訊埠：預設為 502
4. 點擊新增



- Modbus TCP : 連線測試

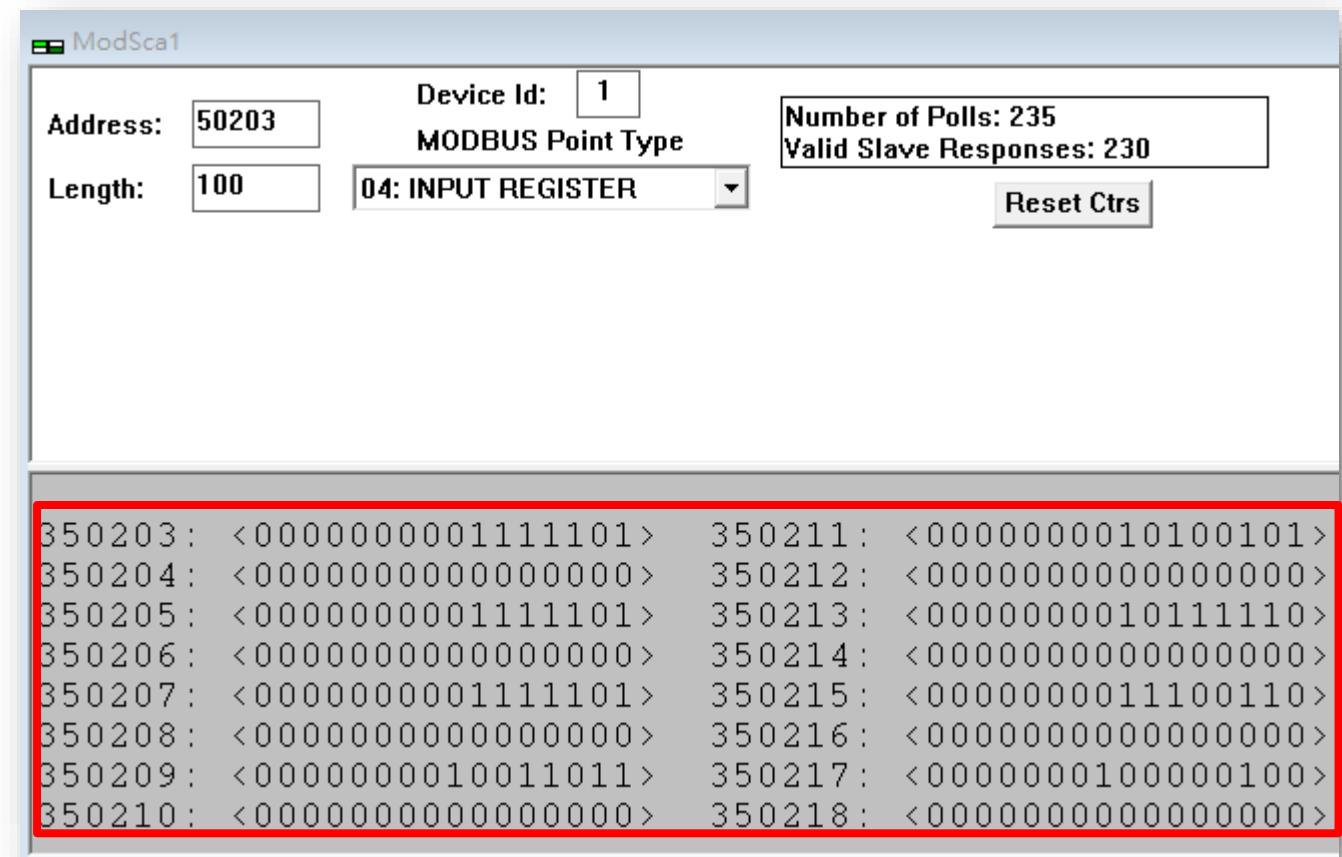
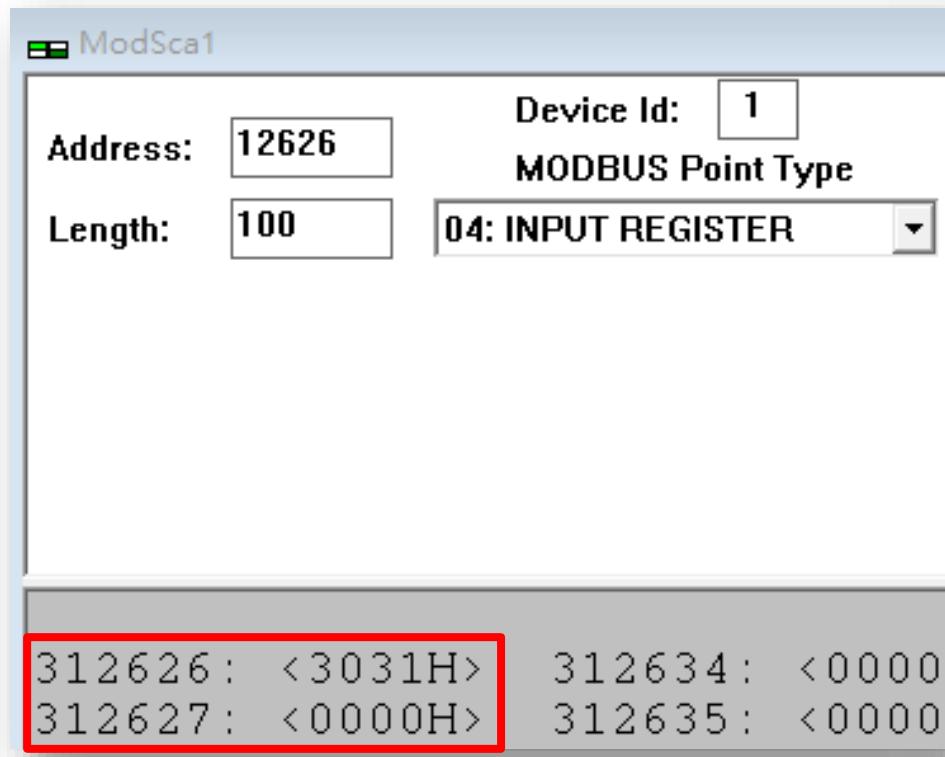
使用 Modscan

1. 進入 Connect 設定 : Connection → Connect
2. 選擇 Remote TCP/IP Server
3. 輸入 IP Address 和 Service Port
4. 點擊 OK



- Modbus TCP : 連線測試

5. 檢查回傳值：字串回傳資料類型為 ASCII，整數回傳資料類型為 Int32



- **RESTful API**

REST(Representation State Transfer) 為 2000 年 Dr. Roy Thomas Fielding 博士在論文中提出的一種全球資訊網軟體架構風格，目的是方便**不同軟體/程式在網路中互相傳遞資訊**。

REST 不是一種標準或協定，而是一種**軟體構建風格**，符合或相容這種架構風格的網路服務，稱為 RESTful，RESTful 允許用戶發出一串 **URL** 來存取和操作網路資源，常用於 IT 產業。RESTful API 回傳資源大多數為 **XML** 或 **JSON**，**VIC7000** 回傳資料為 **JSON**。

# RESTful API

GET PUT POST DELETE

- RESTful API : 取得資料

1. 呼叫 API : 開啟瀏覽器 → 在網址列輸入 **IP/restful/頻道編號**
2. 查看回傳值

1

192.168.133.144/restful

2

```
{"PAGE_NO":1,"TARGET_COUNT":0,"OCR_COUNT":50,"TARGET_COLOR_COUNT":0,"COLOR_COUNT":0,"TARGET_PATTERN_COUNT":0,"PATTERN_COUNT":0,"RESULT":0,"TARGET_EVENT_TRIGGER":0,"EVENT_TRIGGER":0,"TARGET01": "", "TARGET02": "", "TARGET03": "", "TARGET04": "", "TARGET05": "", "OCR01": "110", "OCR02": "110", "OCR03": "13", "OCR04": "125", "OCR05": "125", "OCR06": "18", "OCR07": "125", "OCR08": "125", "OCR09": "25", "OCR10": "155", "OCR11": "155", "OCR12": "6", "OCR13": "165", "OCR14": "165", "OCR15": "27", "OCR16": "190", "OCR17": "195", "OCR18": "12", "OCR19": "225", "OCR20": "215", "OCR21": "10", "OCR22": "260", "OCR23": "265", "OCR24": "22", "OCR25": "285", "OCR26": "285", "OCR27": "26", "OCR28": "295", "OCR29": "295", "OCR30": "28", "OCR31": "110", "OCR32": "110", "OCR33": "29", "OCR34": "125", "OCR35": "125", "OCR36": "9", "OCR37": "125", "OCR38": "125", "OCR39": "8", "OCR40": "155", "OCR41": "155", "OCR42": "3", "OCR43": "165", "OCR44": "165", "OCR45": "13", "OCR46": "190", "OCR47": "195", "OCR48": "18", "OCR49": "225", "OCR50": "225", "TARGET_COLOR01": [0,0,0,0], "TARGET_COLOR02": [0,0,0,0], "TARGET_COLOR03": [0,0,0,0], "TARGET_COLOR04": [0,0,0,0], "TARGET_COLOR05": [0,0,0,0], "COLOR01": [0,0,0,0], "COLOR02": [0,0,0,0], "COLOR03": [0,0,0,0], "COLOR04": [0,0,0,0], "COLOR05": [0,0,0,0], "COLOR06": [0,0,0,0], "COLOR07": [0,0,0,0], "COLOR08": [0,0,0,0], "COLOR09": [0,0,0,0], "COLOR10": [0,0,0,0], "COLOR11": [0,0,0,0], "COLOR12": [0,0,0,0], "COLOR13": [0,0,0,0], "COLOR14": [0,0,0,0], "COLOR15": [0,0,0,0], "COLOR16": [0,0,0,0], "COLOR17": [0,0,0,0], "COLOR18": [0,0,0,0], "COLOR19": [0,0,0,0], "COLOR20": [0,0,0,0], "COLOR21": [0,0,0,0], "COLOR22": [0,0,0,0], "COLOR23": [0,0,0,0], "COLOR24": [0,0,0,0], "COLOR25": [0,0,0,0], "COLOR26": [0,0,0,0], "COLOR27": [0,0,0,0], "COLOR28": [0,0,0,0], "COLOR29": [0,0,0,0], "COLOR30": [0,0,0,0], "COLOR31": [0,0,0,0], "COLOR32": [0,0,0,0], "COLOR33": [0,0,0,0], "COLOR34": [0,0,0,0], "COLOR35": [0,0,0,0], "COLOR36": [0,0,0,0], "COLOR37": [0,0,0,0], "COLOR38": [0,0,0,0], "COLOR39": [0,0,0,0], "COLOR40": [0,0,0,0], "COLOR41": [0,0,0,0], "COLOR42": [0,0,0,0], "COLOR43": [0,0,0,0], "COLOR44": [0,0,0,0], "COLOR45": [0,0,0,0], "COLOR46": [0,0,0,0], "COLOR47": [0,0,0,0], "COLOR48": [0,0,0,0], "COLOR49": [0,0,0,0], "COLOR50": [0,0,0,0], "TARGET_PATTERN01": [0,0,0,0,0,0,0], "TARGET_PATTERN02": [0,0,0,0,0,0,0], "TARGET_PATTERN03": [0,0,0,0,0,0,0], "TARGET_PATTERN04": [0,0,0,0,0,0,0], "TARGET_PATTERN05": [0,0,0,0,0,0,0], "PATTERN01": [0,0,0,0,0,0,0], "PATTERN02": [0,0,0,0,0,0,0], "PATTERN03": [0,0,0,0,0,0,0], "PATTERN04": [0,0,0,0,0,0,0], "PATTERN05": [0,0,0,0,0,0,0], "PATTERN06": [0,0,0,0,0,0,0], "PATTERN07": [0,0,0,0,0,0,0], "PATTERN08": [0,0,0,0,0,0,0], "PATTERN09": [0,0,0,0,0,0,0], "PATTERN10": [0,0,0,0,0,0,0]}
```

- RESTful API : 取得資料

3. 個別呼叫 : 在網址列輸入 **IP/restful/ocr/編號** , 會回傳單一OCR辨識值 , **target** 、 **target\_color** 、 **target\_pattern** 、 **color** 、 **pattern** 亦然

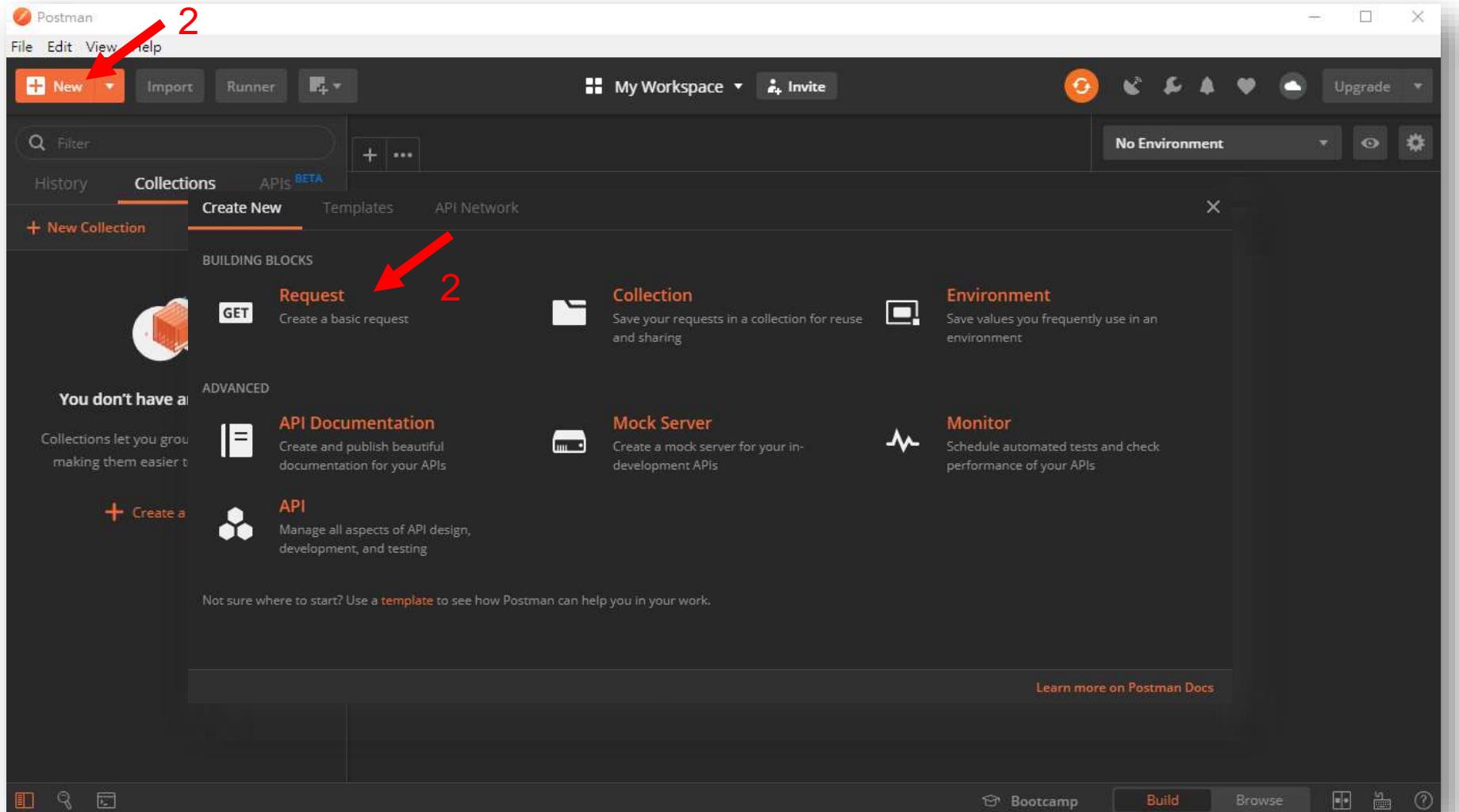
192.168.133.144/restful/ocr/01 → {"OCR01": "125"}

192.168.133.144/restful/target/01 → {"TARGET01": "NexVIC"}

- RESTful API : 控制 VIC 使用 Postman

1. 開啟 Postman

2. 新增 Request : 點擊 New → 選擇 Request

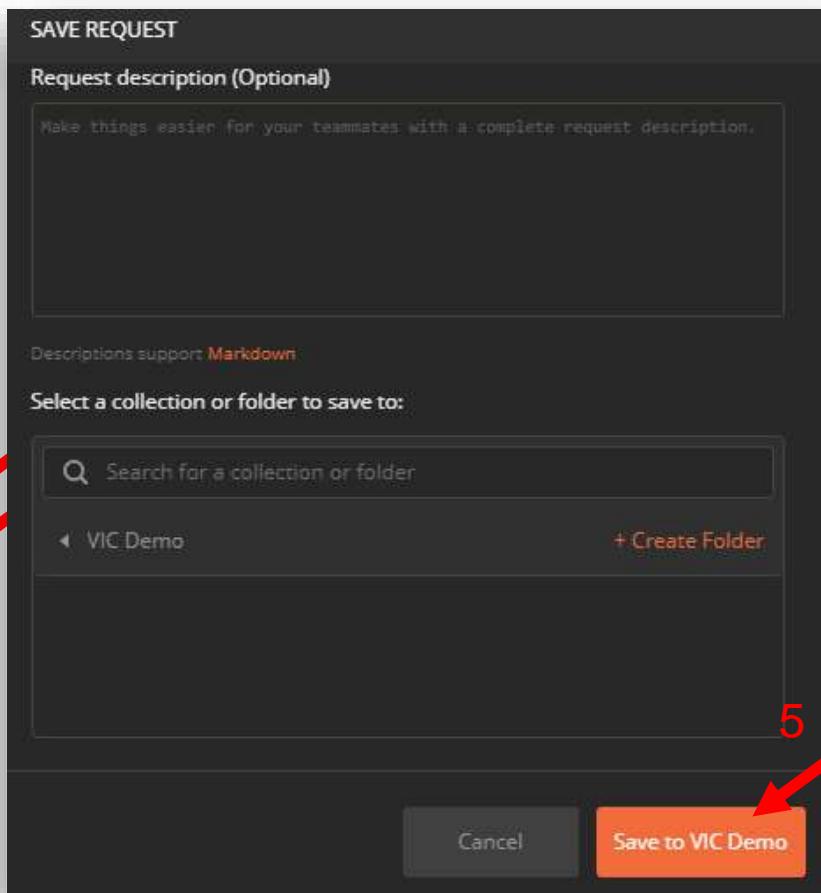
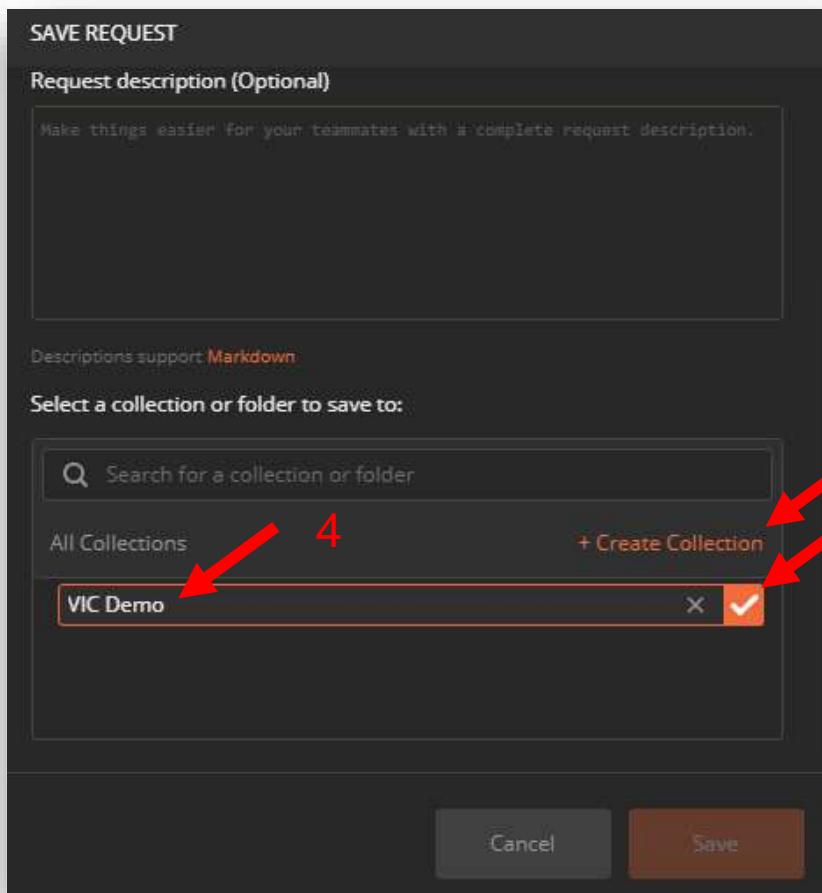
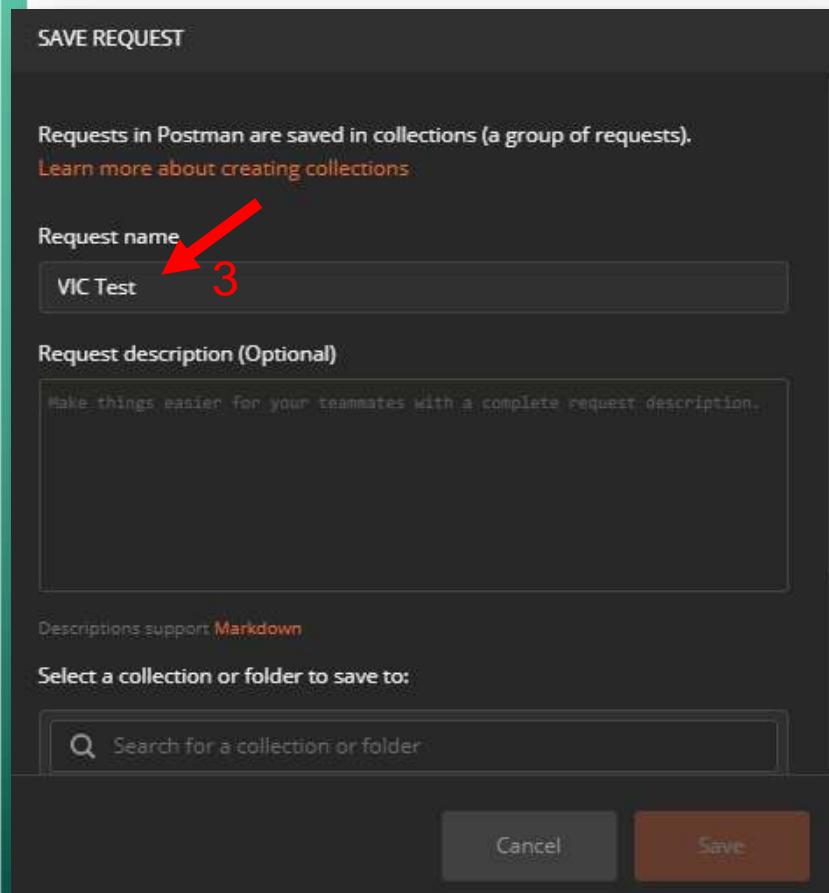


- RESTful API : 控制 VIC

3. 輸入 Request name

4. 選擇 Collection : 若無 Collection , 選擇 + Create Collection

5. Save

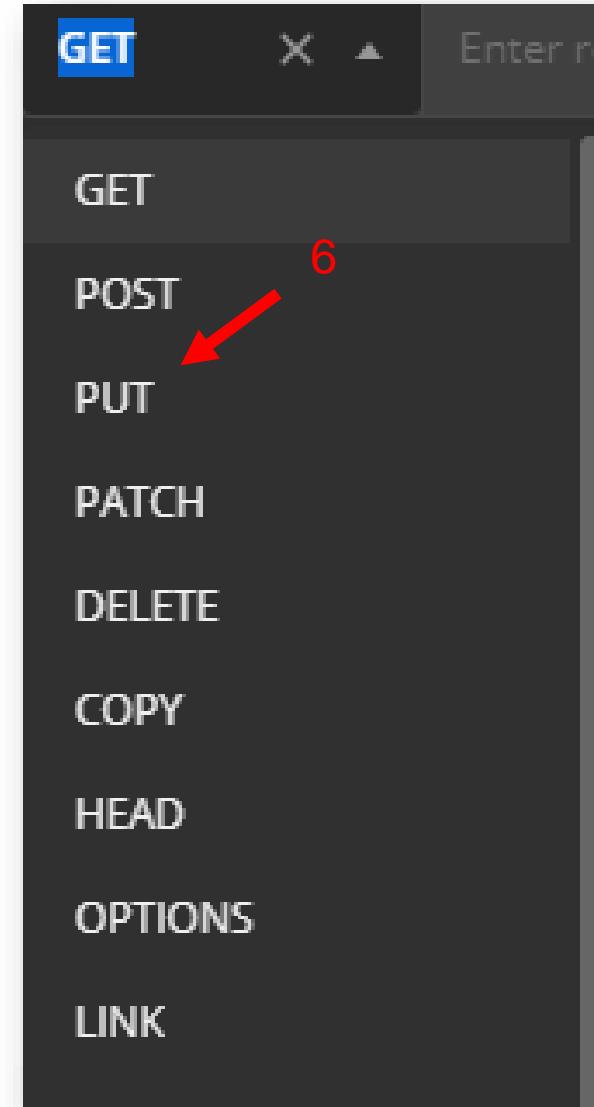
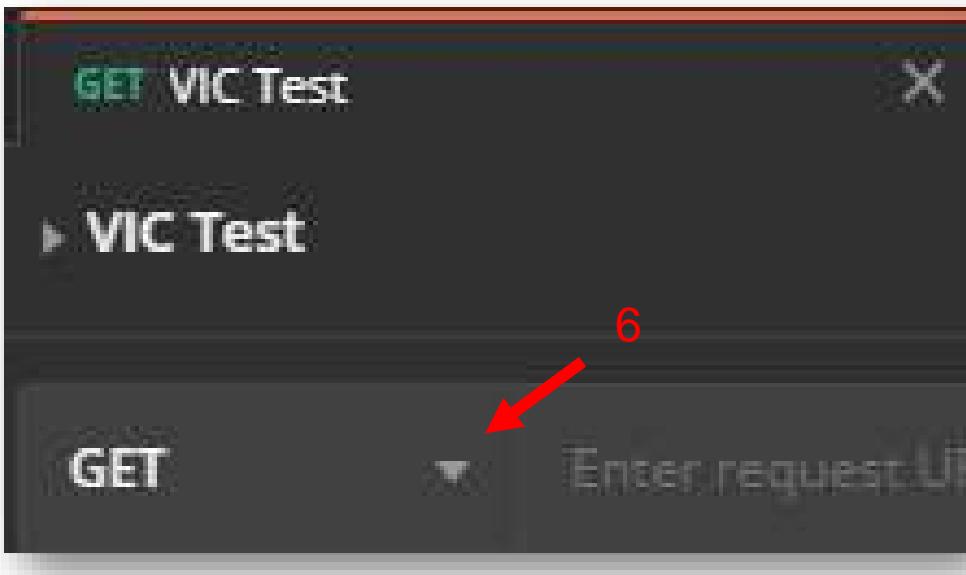


# 外部連線

- RESTful API : 控制 VIC

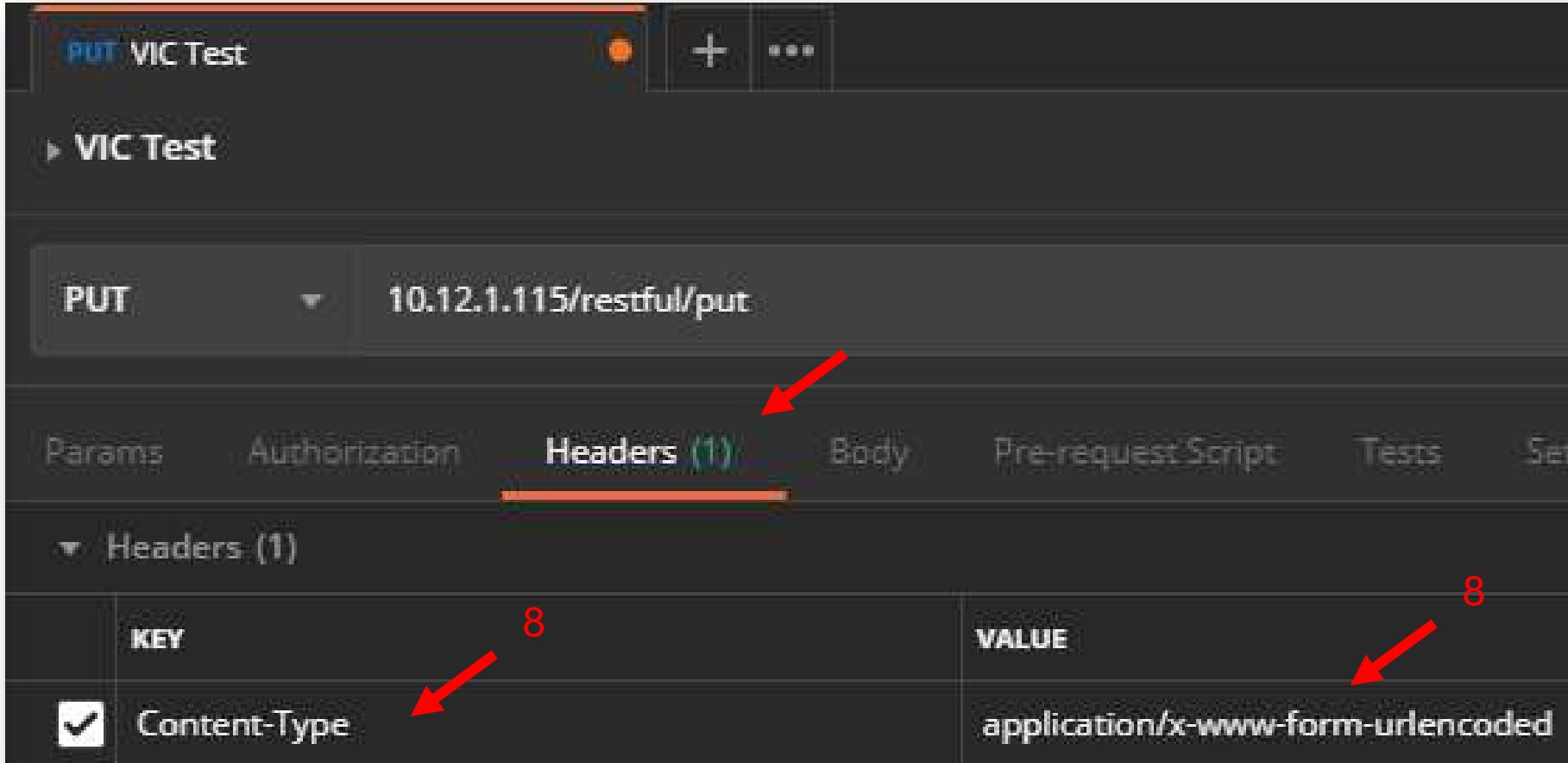
6. 選擇 PUT

7. 輸入 URL : IP/restful/put



- RESTful API : 控制 VIC

8. Headers : Content-Type : application/x-www-form-urlencoded



PUT VIC Test

▶ VIC Test

PUT 10.12.1.115/restful/put

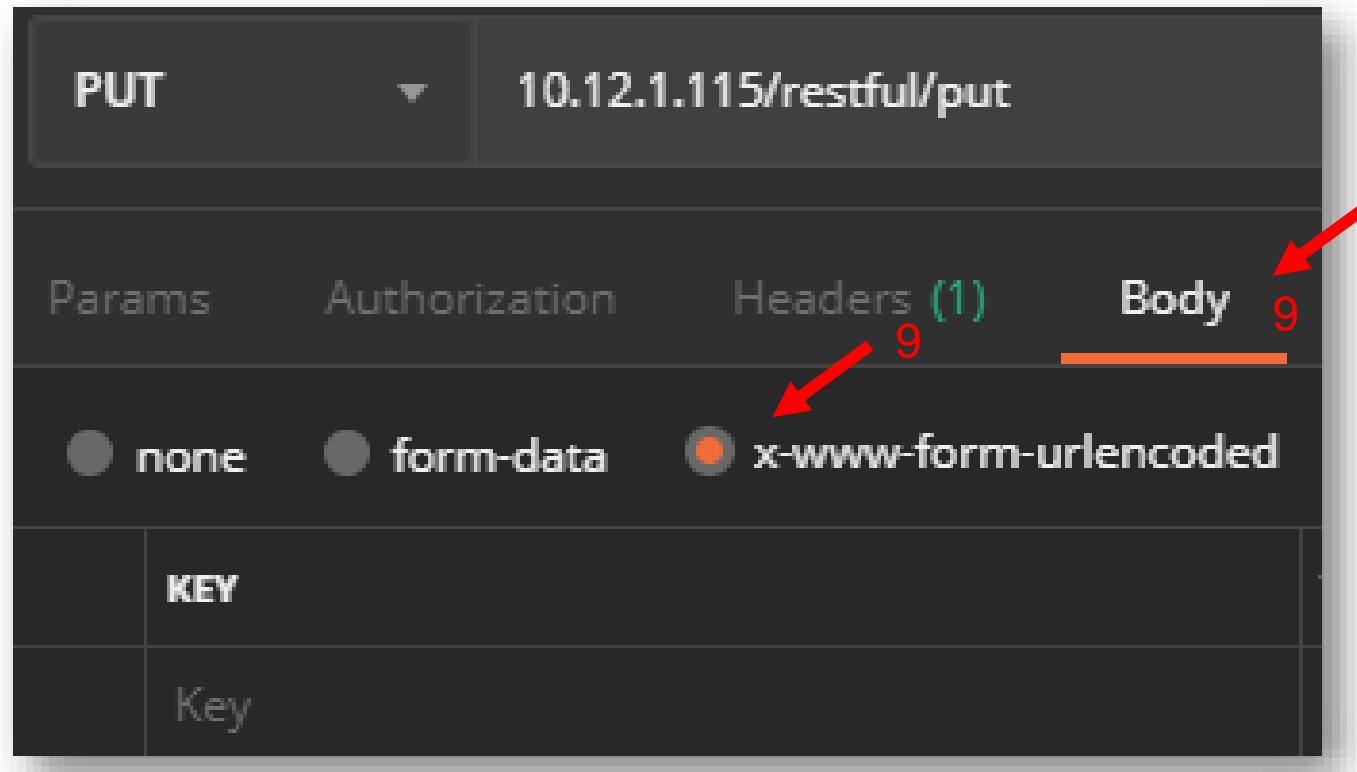
Params Authorization Headers (1) Body Pre-request Script Tests Set

▼ Headers (1)

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/x-www-form-urlencoded

- RESTful API : 控制 VIC

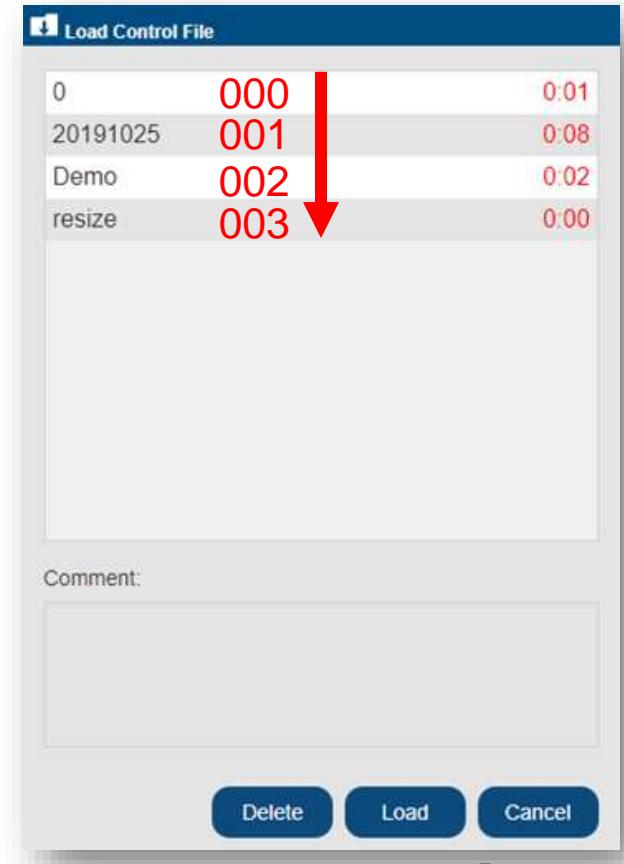
9. Body : x-www-form-urlencoded



- RESTful API : 控制 VIC

9. Body : 啟動程式需要在沒有 admin 登入時才能使用

	啟動程式	運行控制	運行編號 i 的控制檔
<b>username</b>	admin or user	admin or user	admin or user
<b>password</b>	123456 (default)	123456 (default)	123456 (default)
<b>cmd</b>	req_play	req_play_control	req_play_control_file
<b>value</b>	1 = 運行 , 0 = 停止	1 = 運行 , 0 = 停止	1 = 運行 , 0 = 停止
<b>index_value</b>	-	-	0 ~ 199



- RESTful API : 控制 VIC

9. Body : 觸發截圖一張需要在沒有 admin 登入時才能使用

	依名稱啟動控制檔	觸發截圖一張
<b>username</b>	admin or user	admin or user
<b>password</b>	123456 (default)	123456 (default)
<b>cmd</b>	req_play_control_ file_name	req_trigger_one_shot
<b>value</b>	1 = 運行 0 = 停止	-
<b>file_name</b>	要啟動的 控制檔名稱	-

- RESTful API : 控制 VIC

9. Body : 指定當前辨識使用的頁面需要在沒有 admin 登入時才能使用

	設定MODBUSHSTR的值	指定當前辨識使用的頁面	辨識資料是否寫入資料庫
<b>username</b>	admin or user	admin or user	admin or user
<b>password</b>	123456 (default)	123456 (default)	123456 (default)
<b>cmd</b>	req_write_modbush_string	req_trigger_page_assignment	req_trigger_database_write
<b>value</b>	-	-	1 = 寫入 · 0 = 不寫入
<b>index_value</b>	MODBUSHSTR編號 0 ~ 199	要指定的頁面編號 1 ~ 30	-
<b>content</b>	要寫入的內容	-	-

- RESTful API : 控制 VIC

9. Body : 執行腳本函式需要在沒有 admin 登入時才能使用

	執行腳本函式	停用控制
username	admin or user	admin or user
password	123456 (default)	123456 (default)
cmd	req_trigger_script_function	req_control_disable
value	-	1 = 停用 · 0 = 不停用
index_value	400~499	-

**OnTriggerScriptFunction(address)**  
index\_value = address

# 外部連線

- RESTful API : 控制 VIC

9. Body : 啟動程式

10. Send

PUT VIC Test

No Environment

Comments (0)

PUT 10.12.1.115/restful/put Send

Params Authorization Headers (1) **Body** Pre-request Script Tests Settings

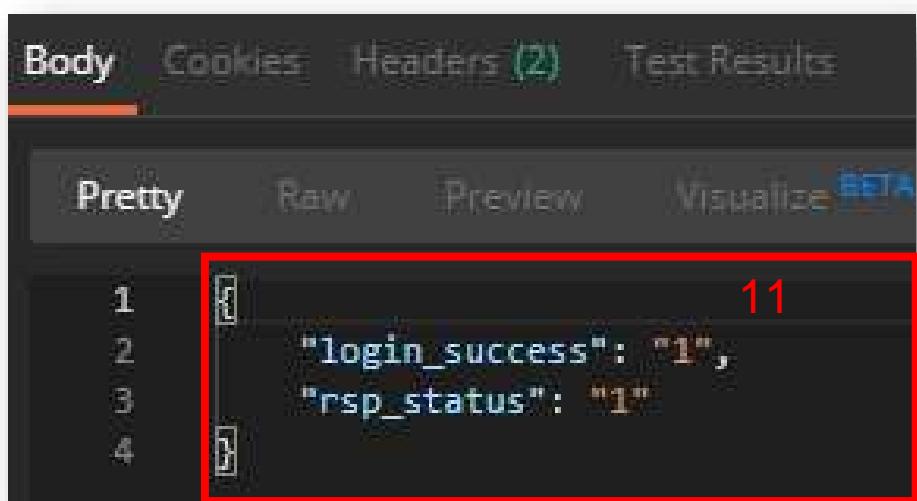
none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	9	DESCRIPTION
<input checked="" type="checkbox"/> username	admin		
<input checked="" type="checkbox"/> password	123456		
<input checked="" type="checkbox"/> channel_type	play		
<input checked="" type="checkbox"/> value	1		

Key Value Description

- RESTful API : 控制 VIC

- 11. 觀察 Response



```
1  {
2   "login_success": "1",
3   "rsp_status": "1"
4 }
```

<b>login_success</b>	是否登入成功 1 = 成功 · 0 = 失敗
<b>rsp_status</b>	是否成功執行指令 1 = 成功 · 0 = 失敗

- **SQL**

SQL，Structure Query Language，結構化查詢語言是一種特定目的的語法，用於管理資料庫系統。SQL 的範圍包括資料插入、查詢、更新和刪除、資料庫模式建立和修改、資料存取控制。

## 查詢

```
SELECT “欄位”  
FROM “表格”  
WHERE “條件”  
GROUP BY “條件”  
ORDER BY “[ASC, DESC]”  
LIMIT “限制數量”
```



- **MySQL**

MySQL 是一個開放原始碼的關聯式資料庫系統，現為 Oracle 公司旗下產品。由於效能高、成本低、可靠性好，MySQL 已成為最流行的開源資料庫

VIC7000 使用 MySQL 來儲存資料，使用者也可以由另一台電腦使用 MySQL 連到 VIC7000 電腦的資料庫

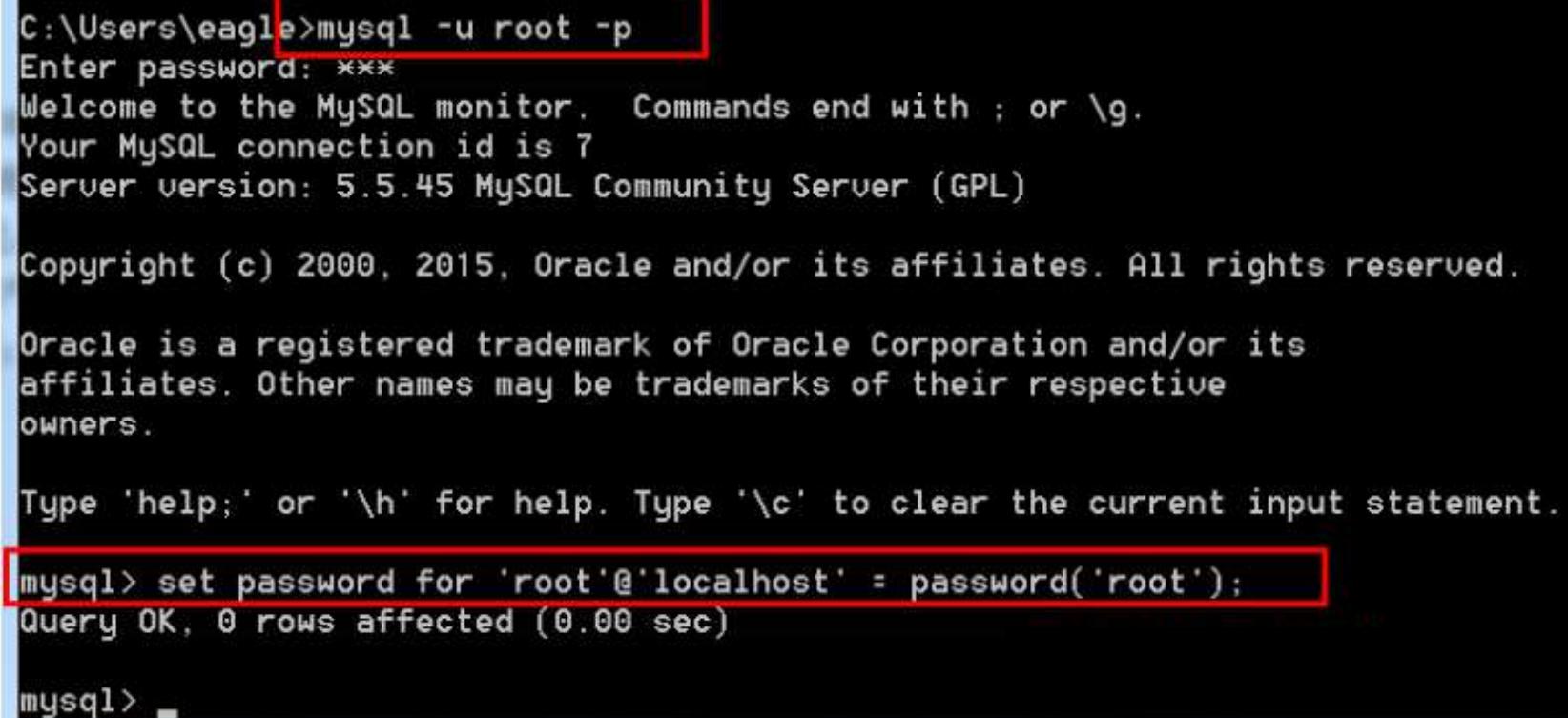


- MySQL

VIC7000 電腦內的資料庫名稱為 vic，使用者名稱為 vic\_7000，密碼為 vic\_7000，需使用 VIC7000 電腦才可取用資料

若要遠端取得資料，需用使用者 **root** 自行**新增使用者**，預設密碼為 123456，使用者可自行更換密碼

```
set password for 'root'@'localhost' = password('root');
```



```
C:\Users\user>mysql -u root -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.5.45 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> set password for 'root'@'localhost' = password('root');
Query OK, 0 rows affected (0.00 sec)

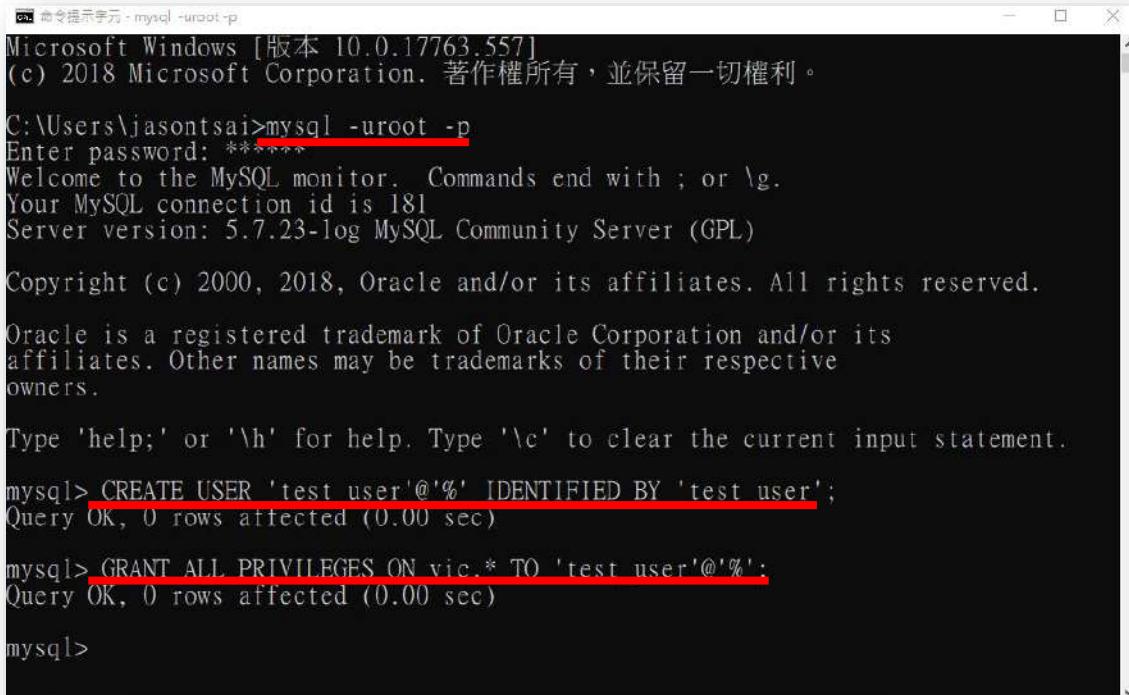
mysql> _
```

- MySQL

用使用者 **root** 自行新增使用者後，再給予新使用者取用 **vic** 資料庫的權限，  
就可以使用新增的使用者讀取 **vic** 資料庫內的資料

```
CREATE USER 'test_user'@'%' IDENTIFIED BY 'test_user';
```

```
GRANT ALL PRIVILEGES ON vic.* TO 'test_user'@'%';
```



```
命令提示字元 - mysql -uroot -p
Microsoft Windows [版本 10.0.17763.557]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\jasontsai>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 181
Server version: 5.7.23-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

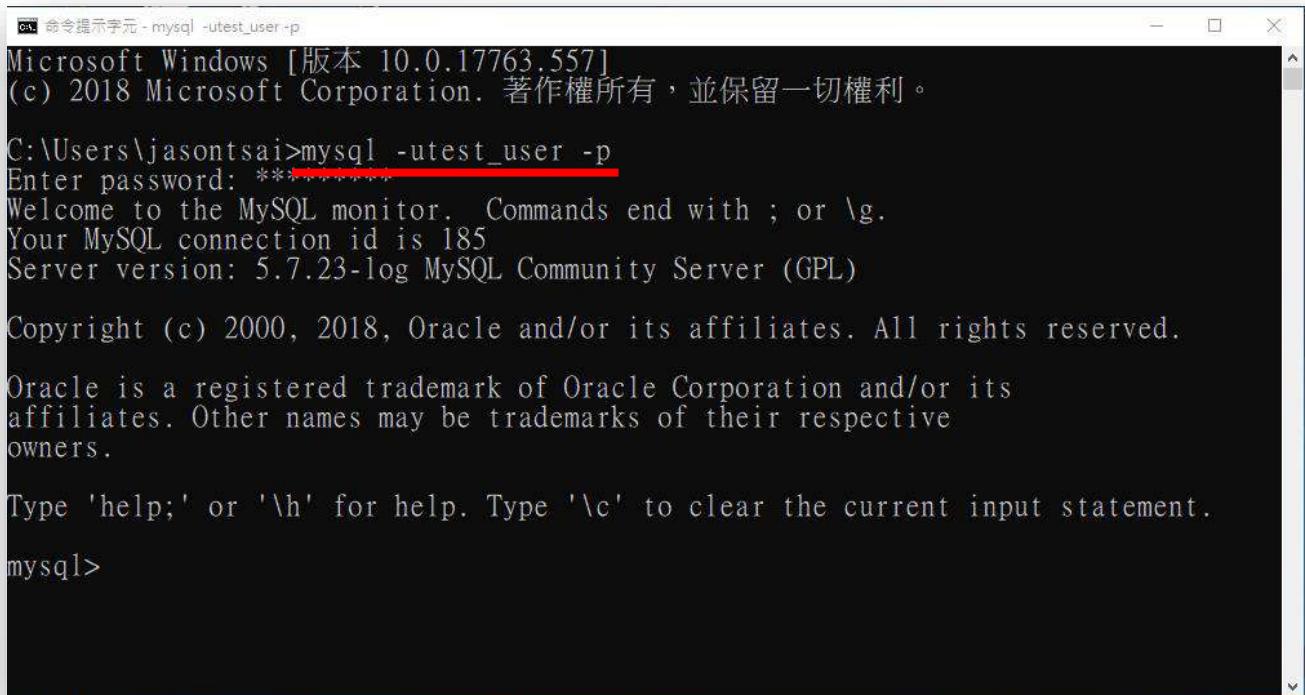
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'test_user'@'%' IDENTIFIED BY 'test_user';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON vic.* TO 'test_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql>
```



```
命令提示字元 - mysql -utest_user -p
Microsoft Windows [版本 10.0.17763.557]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\jasontsai>mysql -utest_user -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 185
Server version: 5.7.23-log MySQL Community Server (GPL)

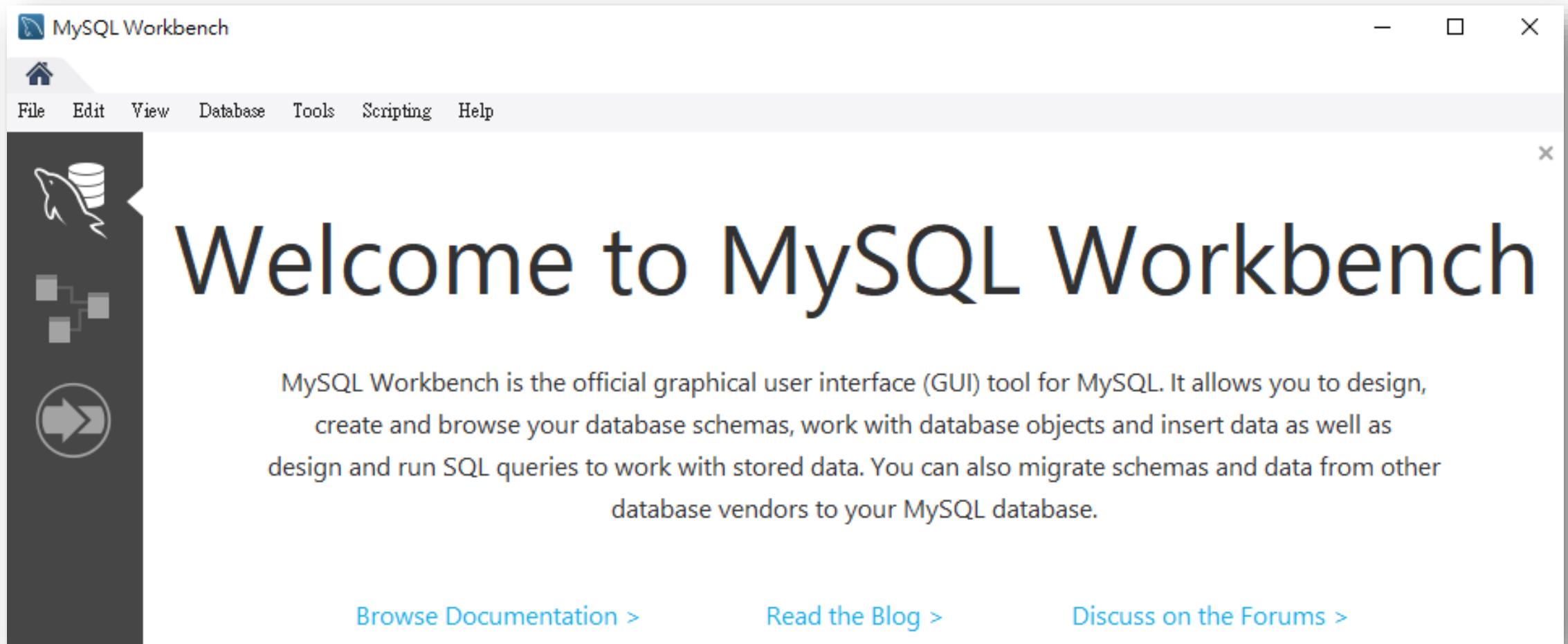
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

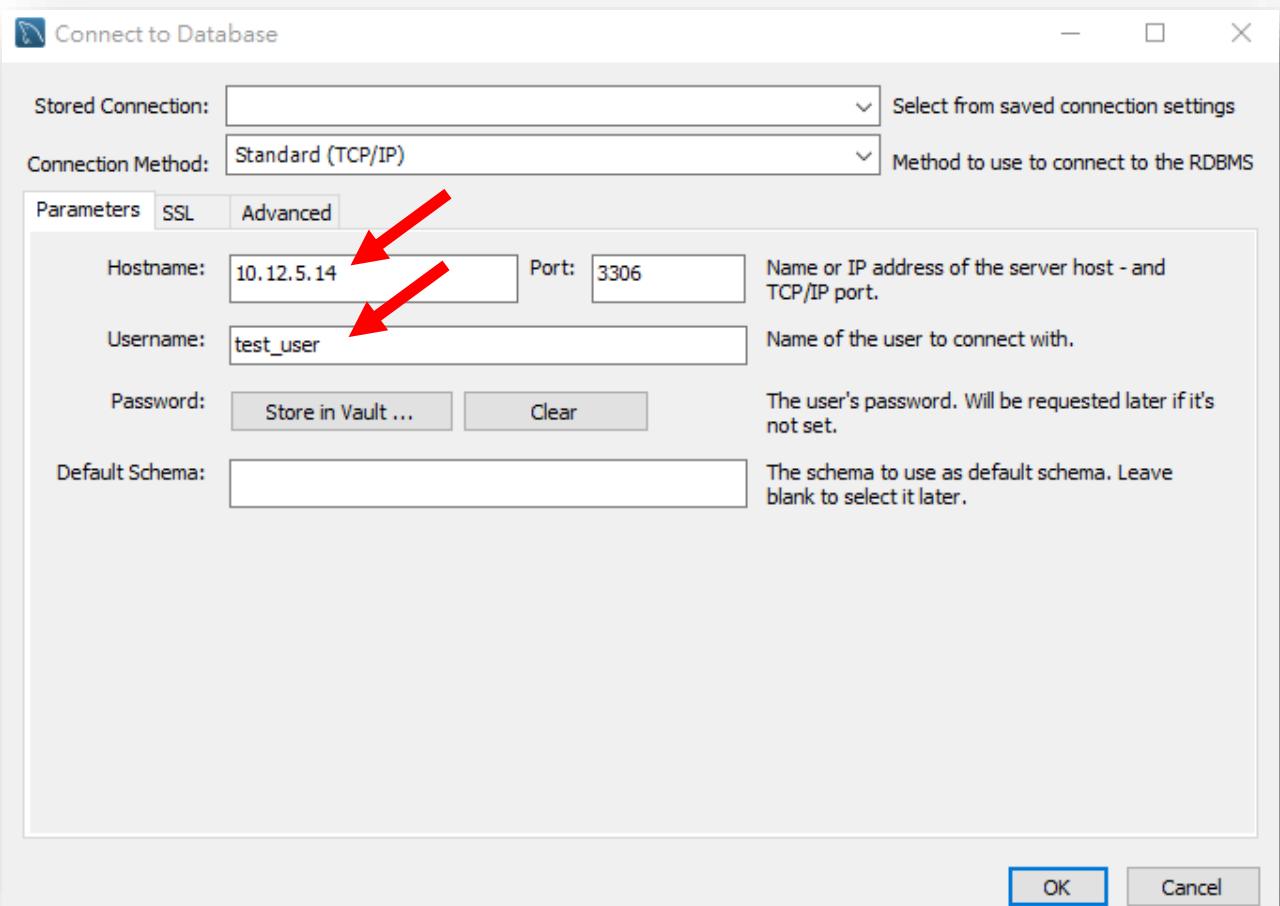
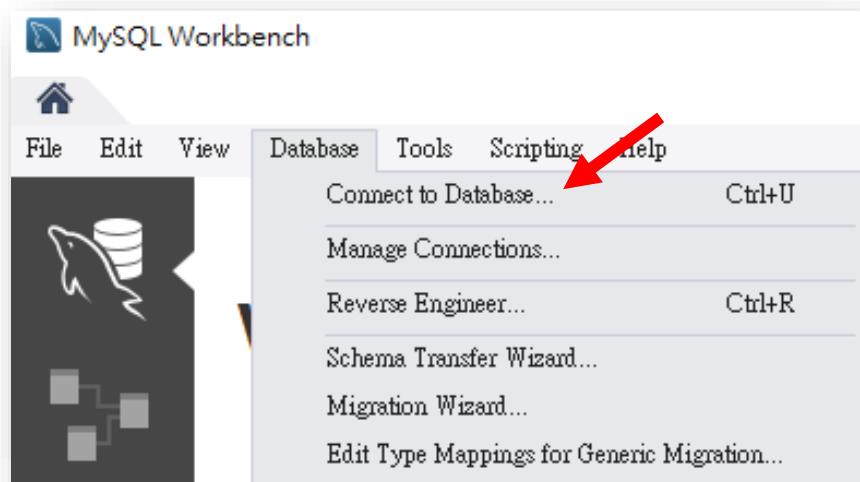
mysql>
```

- MySQL : 連線測試 : 使用 MySQL Workbench



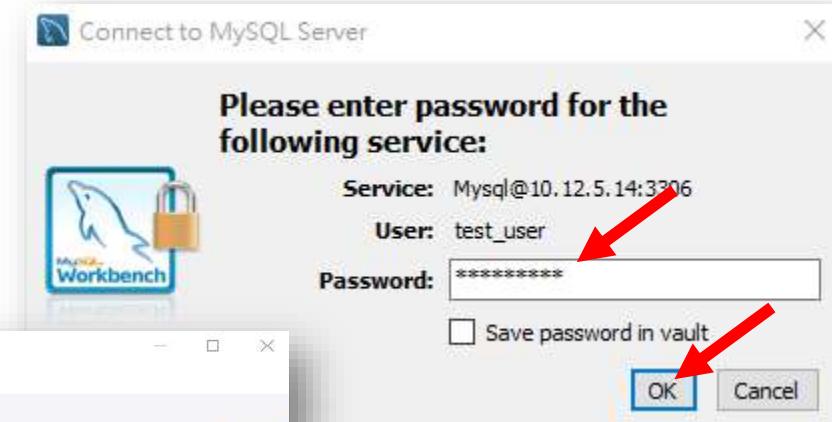
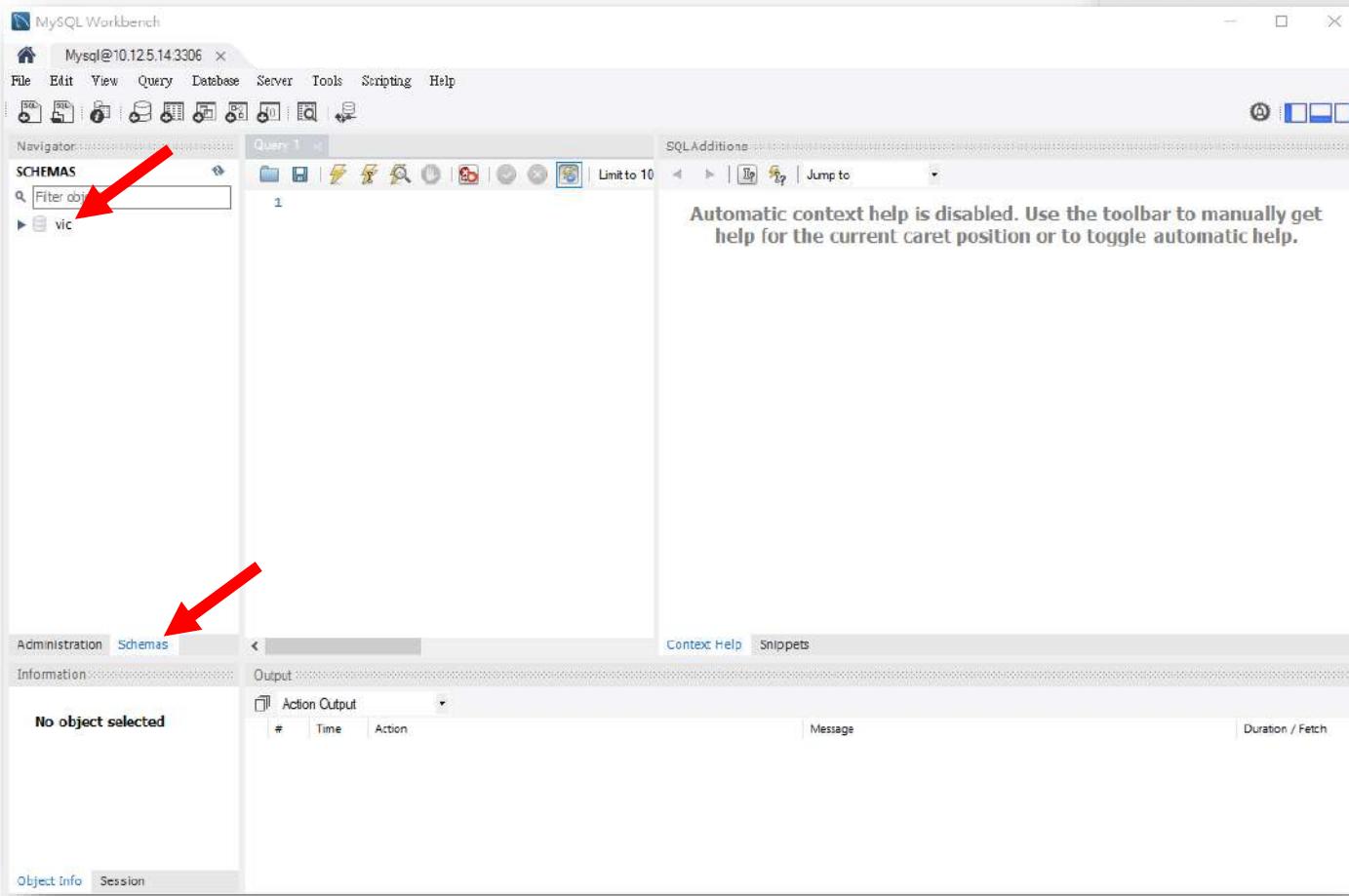
- MySQL : 連線測試

- 連線設定 : Database → Connect to Database... → Hostname : IP → Username : test\_user



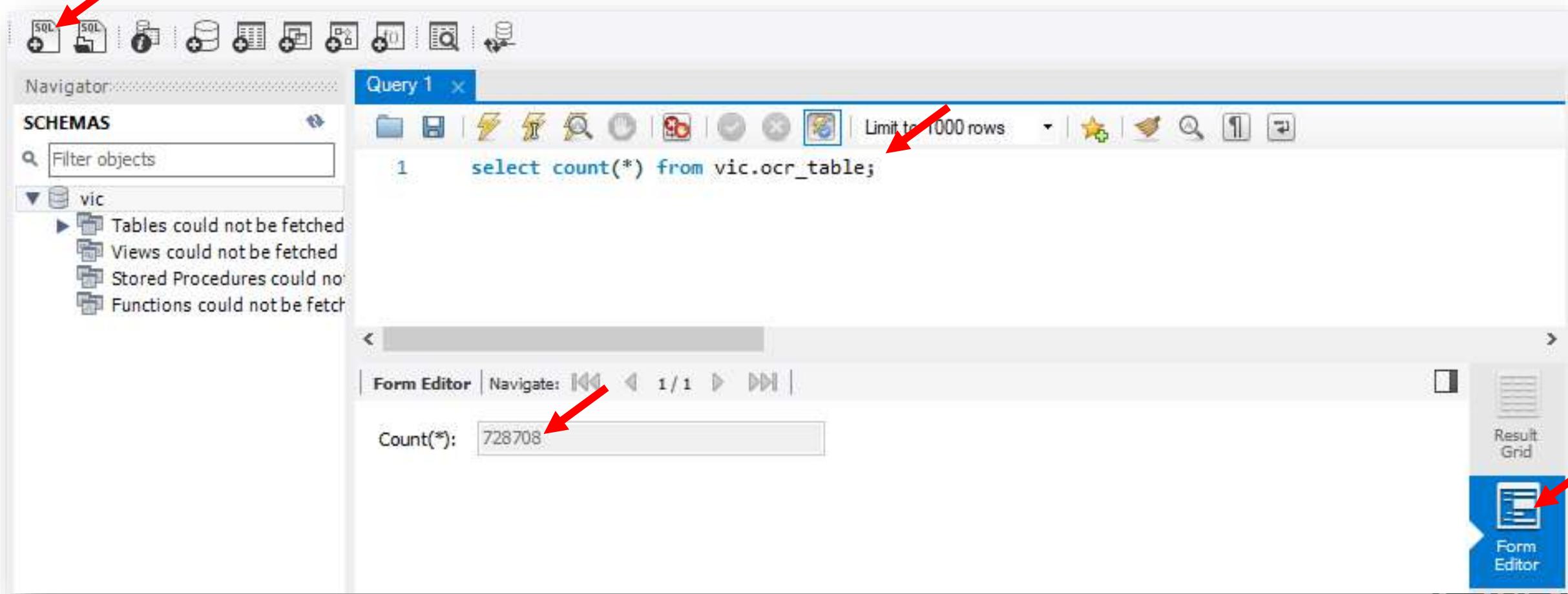
# 外部連線

- MySQL : 連線測試
- 3. 輸入密碼點擊 OK
- 4. 查看資料庫



- MySQL : 連線測試

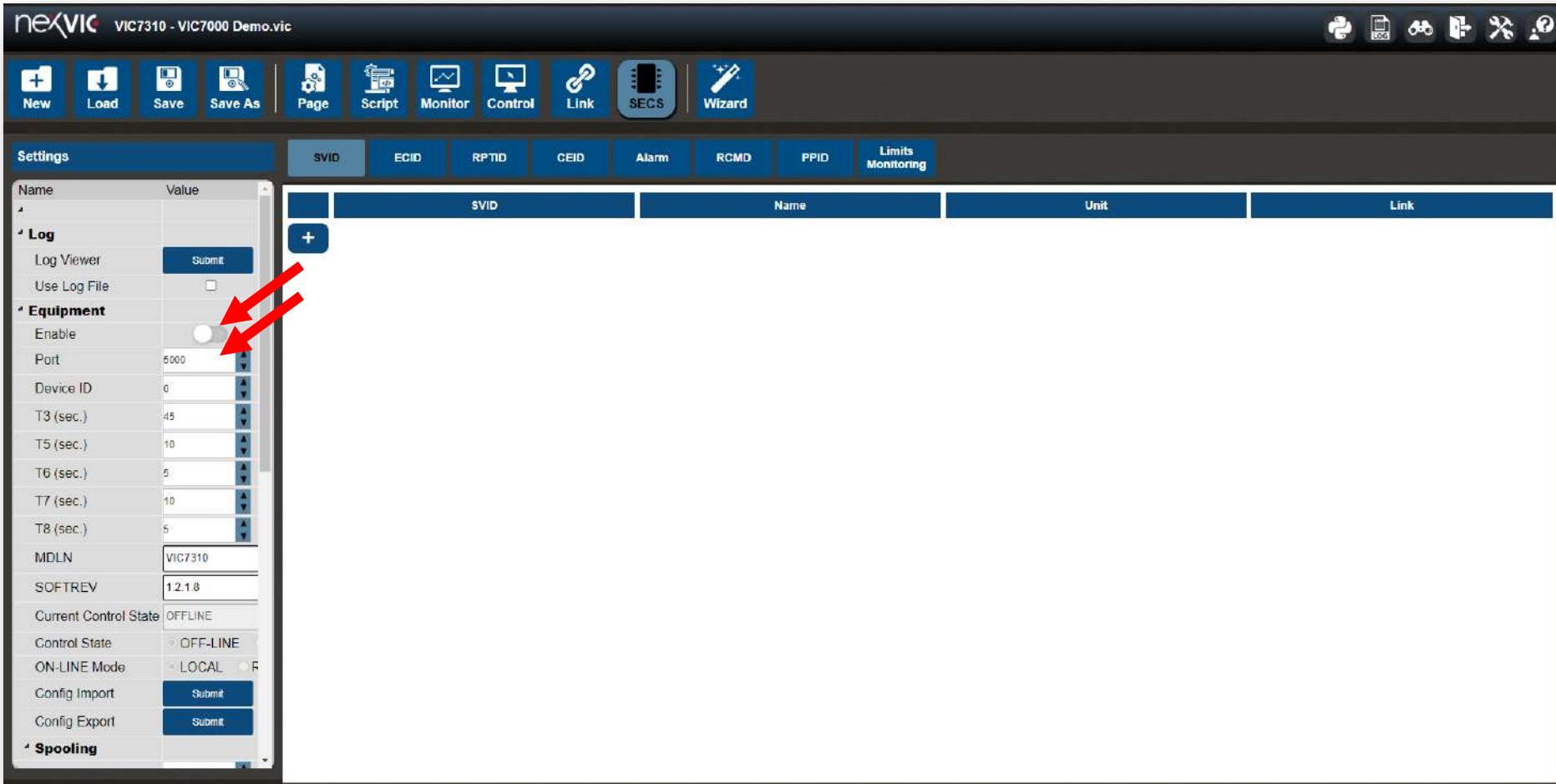
- 輸入和執行指令 : Create a new SQL tab for executing queries → 輸入 SQL 指令(此範例為計算 vic 內有幾筆資料) → Form Editor



## • SECS/GEM : Equipment

進入SECS頁面，在Equipment設定中切換Enable的狀態，即可開啟SECS/GEM連線

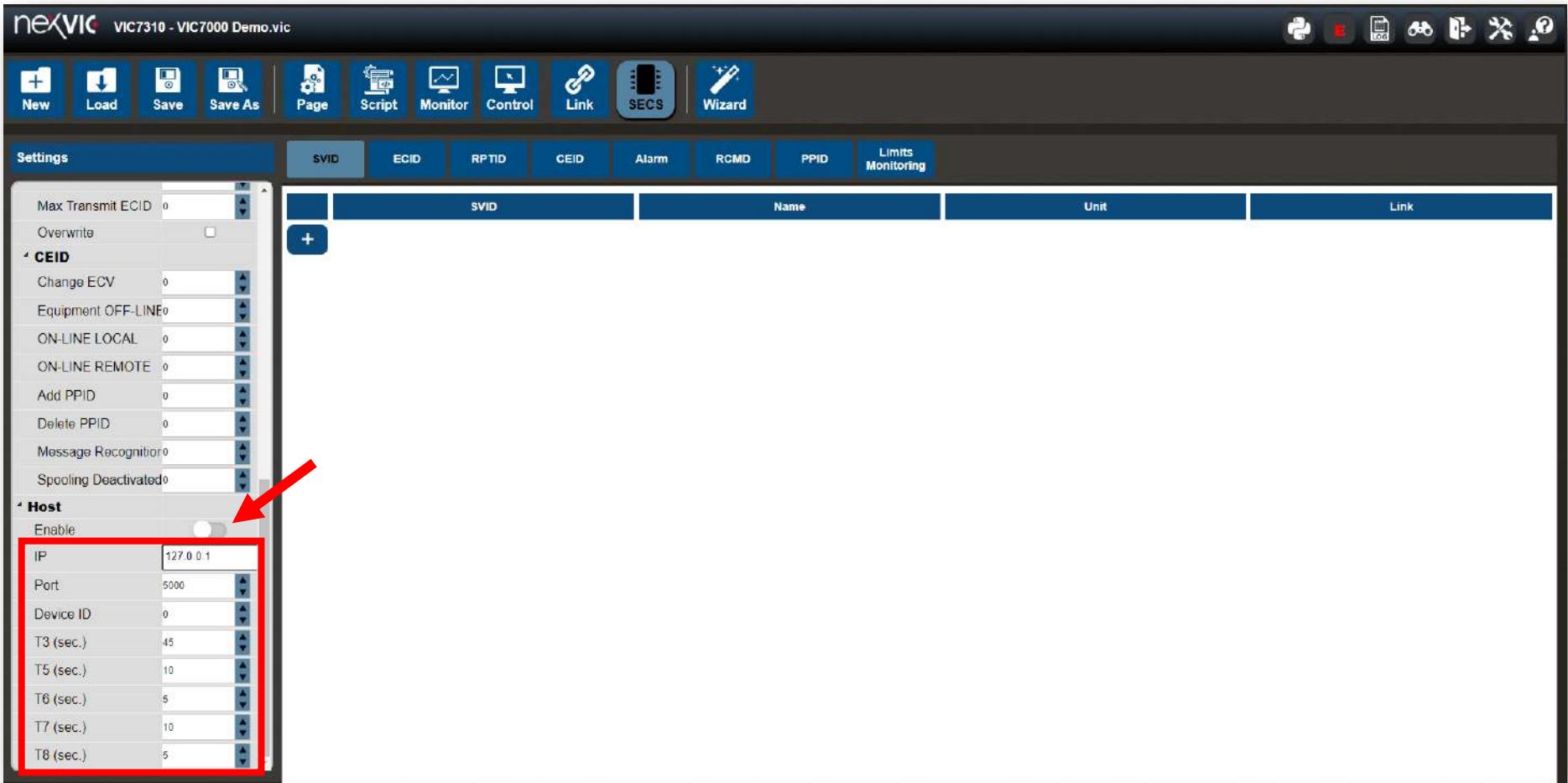
此時VIC為Equipment，開啟連線前，可由通訊埠欄位設置要使用的通訊埠，預設為5000



- **SECS/GEM : Host**

進入SECS頁面，在Host設定中切換Enable的狀態，即可開啟SECS/GEM連線

此時VIC為Host，連線前，需設置連接Equipment的IP位址、通訊埠、站號、通訊等待時間



# 外部連線

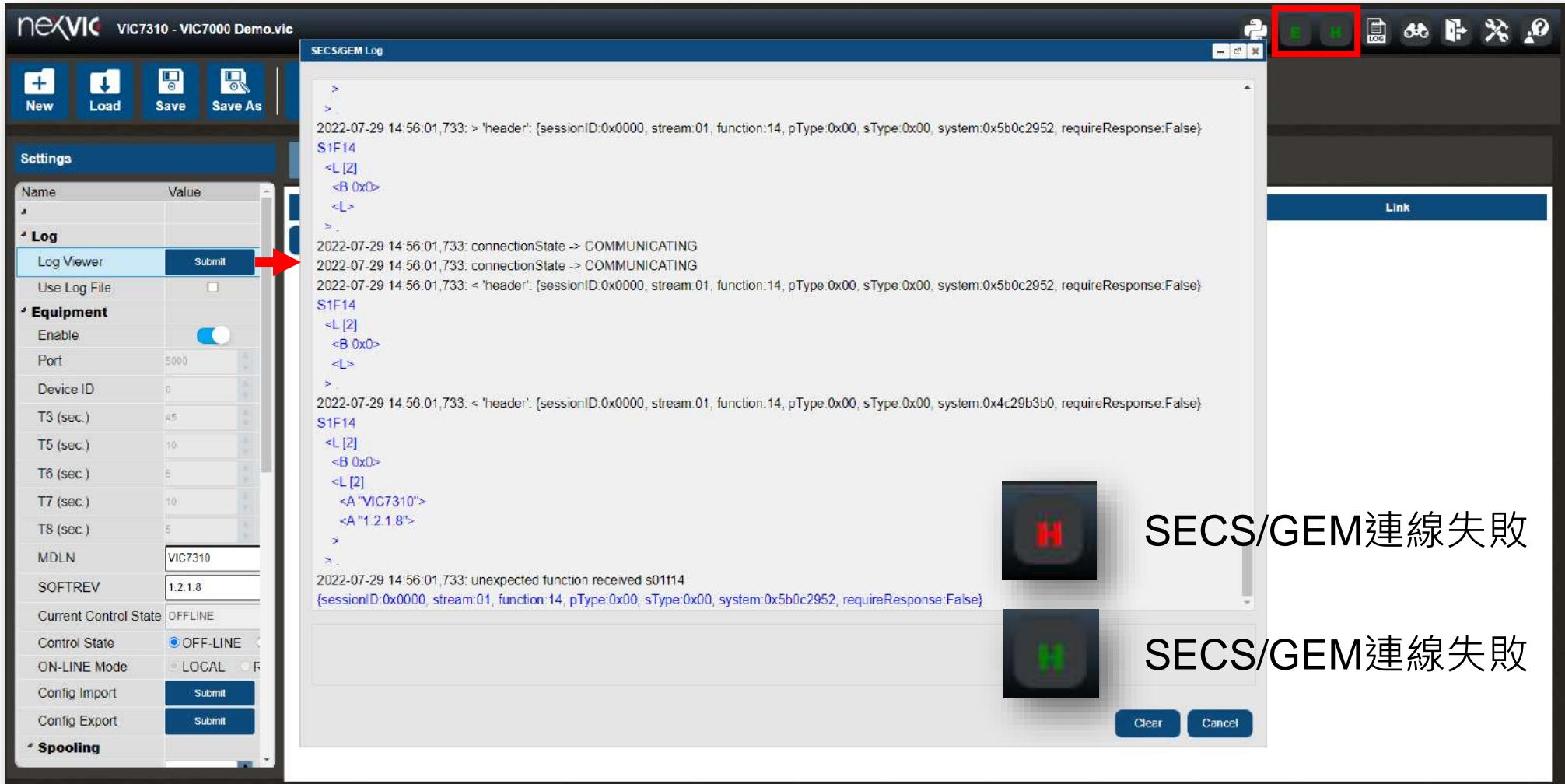
- SECS/GEM : Host

1. 進入連結頁面：從管理模式進入
2. 選擇 SECS/GEM
3. 輸入 Equipment IP、通訊埠、通訊等待時間
4. 點擊新增 即可新增其他SECS/GEM Host連線



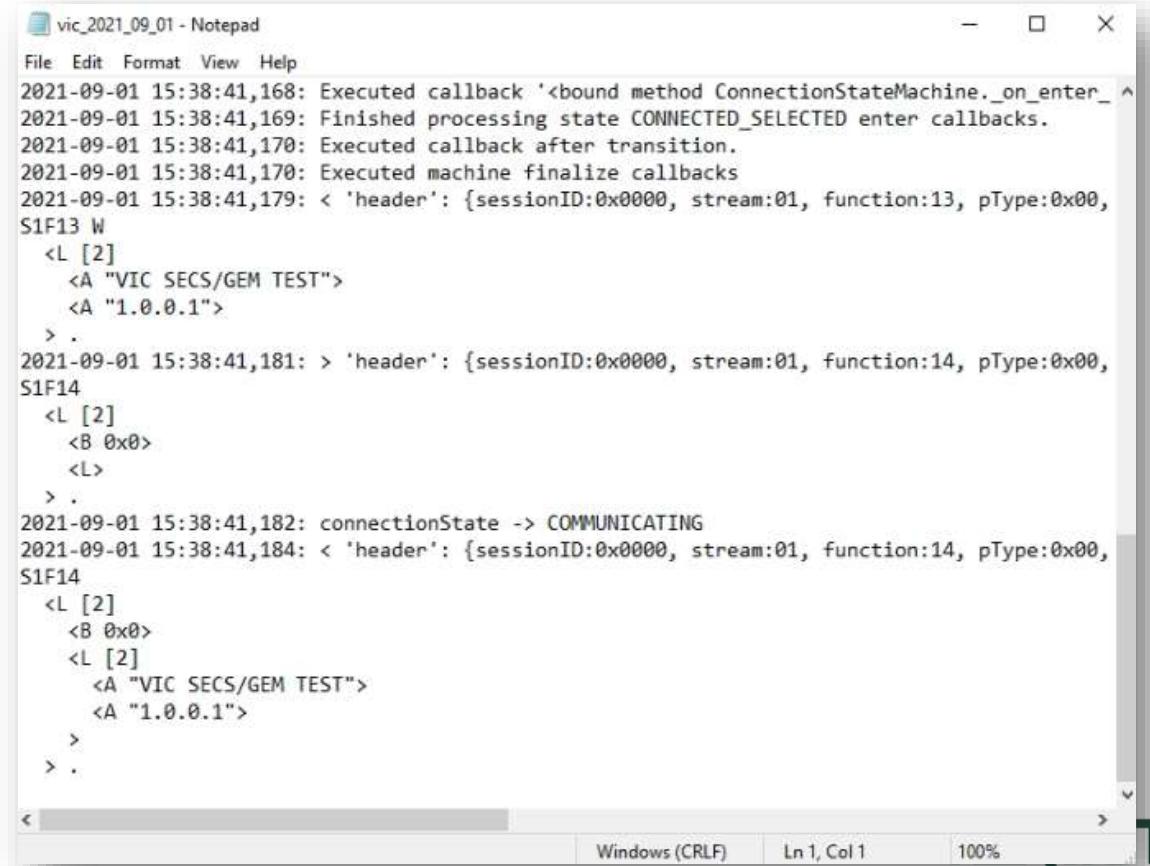
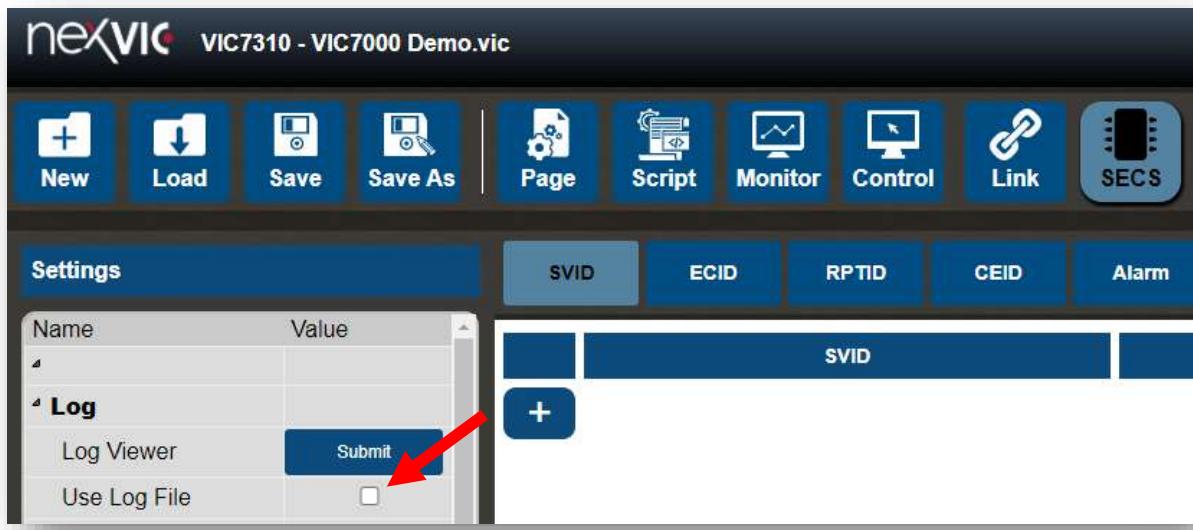
- SECS/GEM：日誌檢視

點擊日誌檢視或右上角的SECS/GEM連線圖示，即可看到SECS/GEM的通訊日誌



- **SECS/GEM** : 使用日誌檔

在SECS頁面，啟用使用日誌檔，系統會儲存 SECS/GEM 通訊日誌在 VIC7000 本機上  
儲存路徑為 C:\VIC7000\bin\log，可以看到儲存的 SECS/GEM 通訊日誌檔  
日誌檔格式為 txt

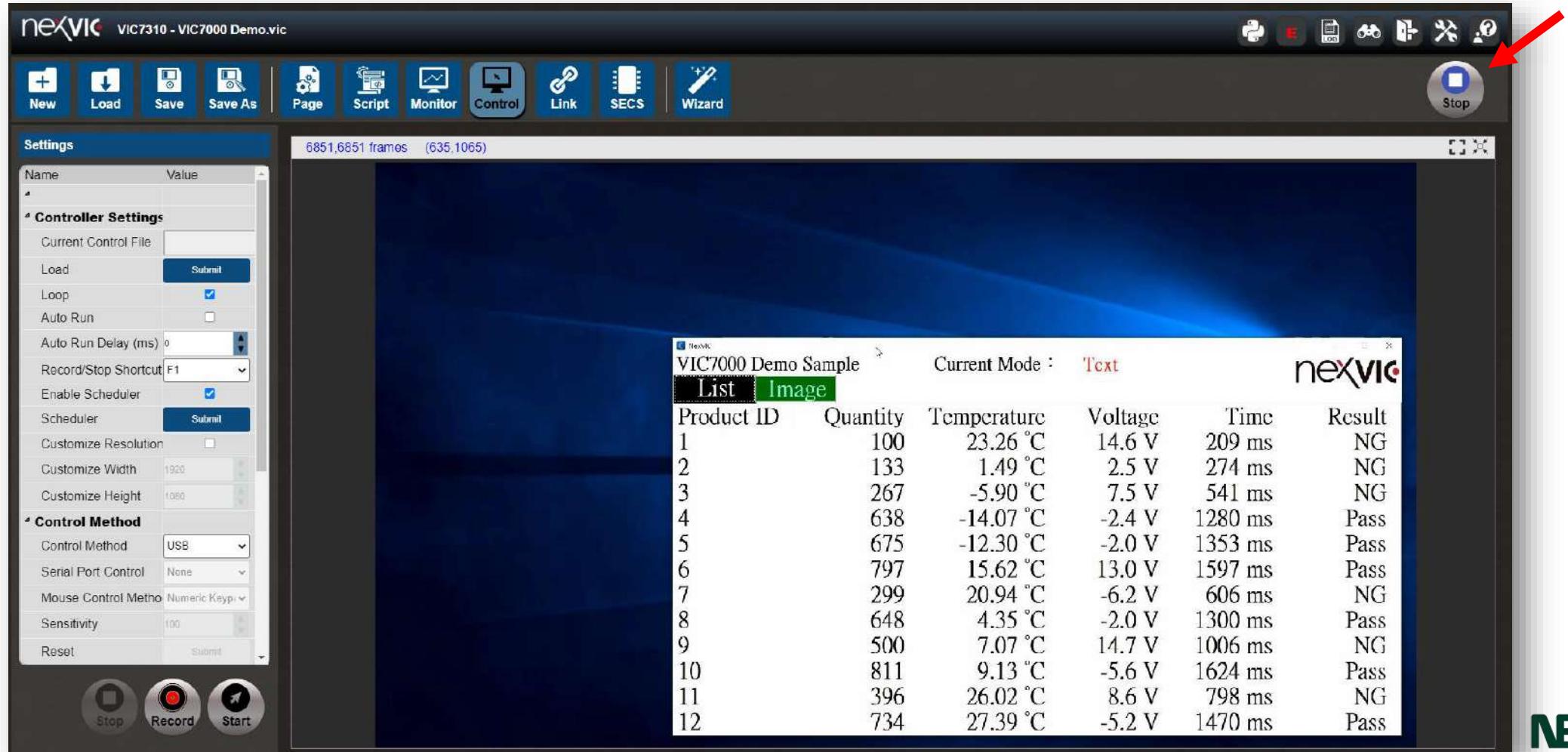


# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

- **控制**：錄製滑鼠和鍵盤動作

1. 進入控制頁面：需搭配擷取卡輸入使用，編輯時需要讓擷取卡持續取像



## • 控制設定

### • 控制參數

目前檔案路徑

載入：載入控制檔

循環：是否循環執行控制檔

自動啟動：開始取像時是否  
同時啟動控制

自動啟動延遲時間 (ms)

錄製/停止快捷鍵

是否啟動排程

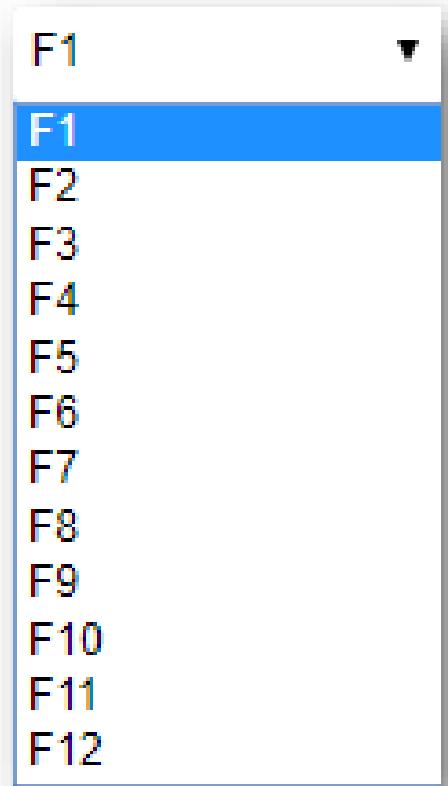
排程設定

自訂控制解析度

自訂寬度/高度

Settings

Name	Value
Controller Settings	
Current Control File	<input type="text"/>
Load	<input type="button" value="Submit"/>
Loop	<input checked="" type="checkbox"/>
Auto Run	<input type="checkbox"/>
Auto Run Delay (ms)	0 <input type="button" value="▼"/>
Record/Stop Shortcut	F1 <input type="button" value="▼"/>
Enable Scheduler	<input checked="" type="checkbox"/>
Scheduler	<input type="button" value="Submit"/>
Customize Resolution	<input type="checkbox"/>
Customize Width	1920 <input type="button" value="▼"/>
Customize Height	1080 <input type="button" value="▼"/>



# 控制

NOVIC

- **控制設定**

- **控制方法**

控制方法：USB、序列COM、本機、瀏覽器

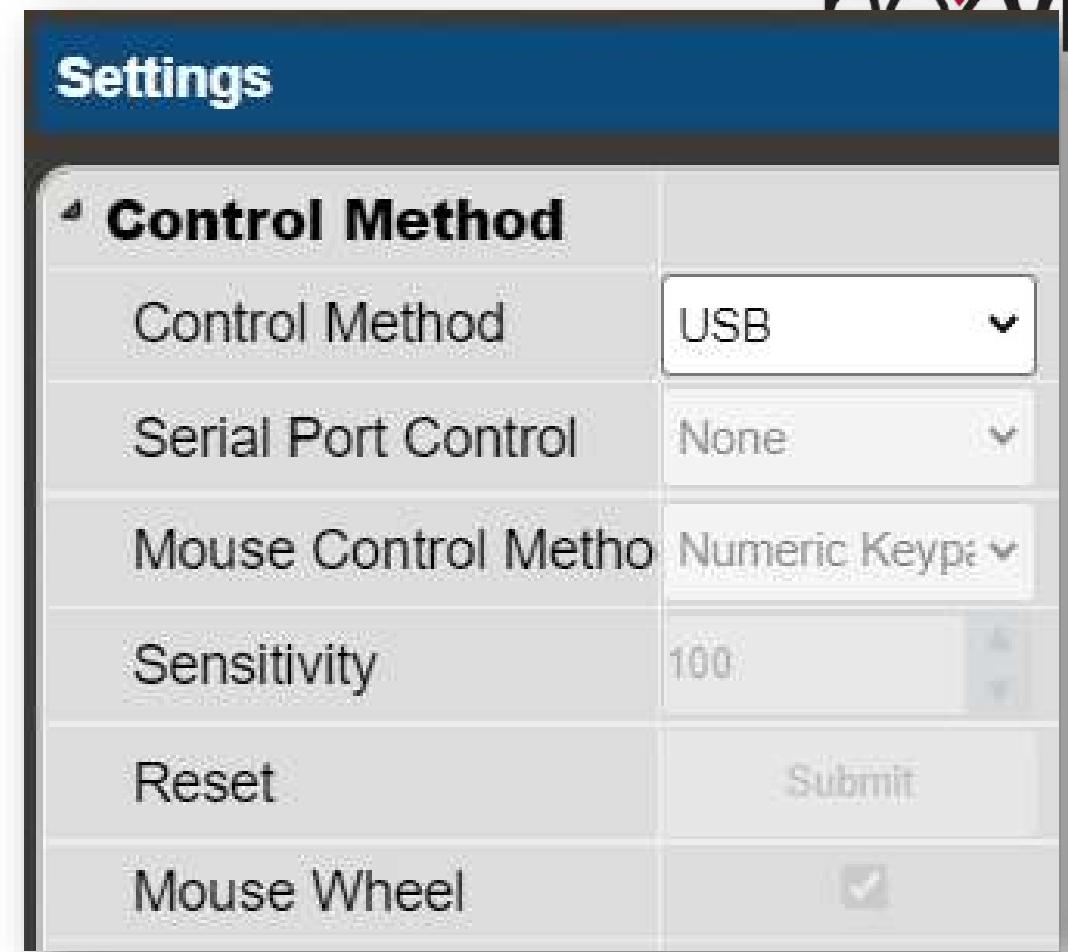
序列埠控制：連接 PS/2 控制盒的序列埠

滑鼠控制方法：PS/2 的滑鼠控制方法

靈敏度

重置

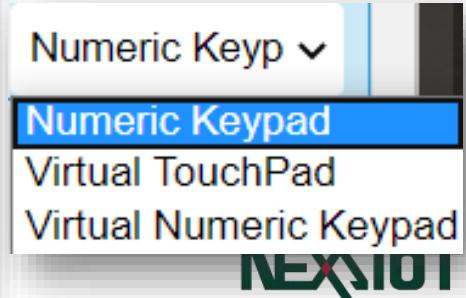
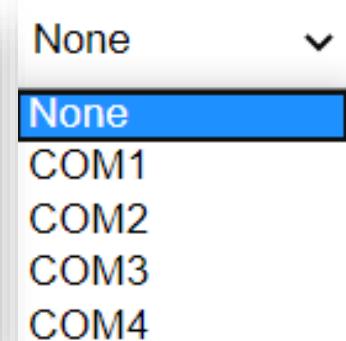
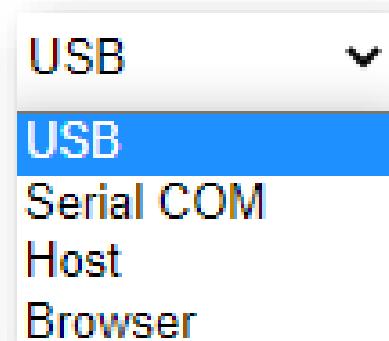
PS/2控制的滑鼠是否有滾輪



USB、序列COM 常用於控制影像輸入源

本機控制可控制VIC7000本機

瀏覽器可控制VIC本機開啟的瀏覽器



NEXIOT

- **控制設定**

- **影像輸出**

是否調整影像大小及品質：修改影像寬度和品質可改善**錄製控制檔時**的流暢度

寬度

品質

是否啟用灰階模式

- **影像資訊**

影像寬度

影像高度

影像頻率

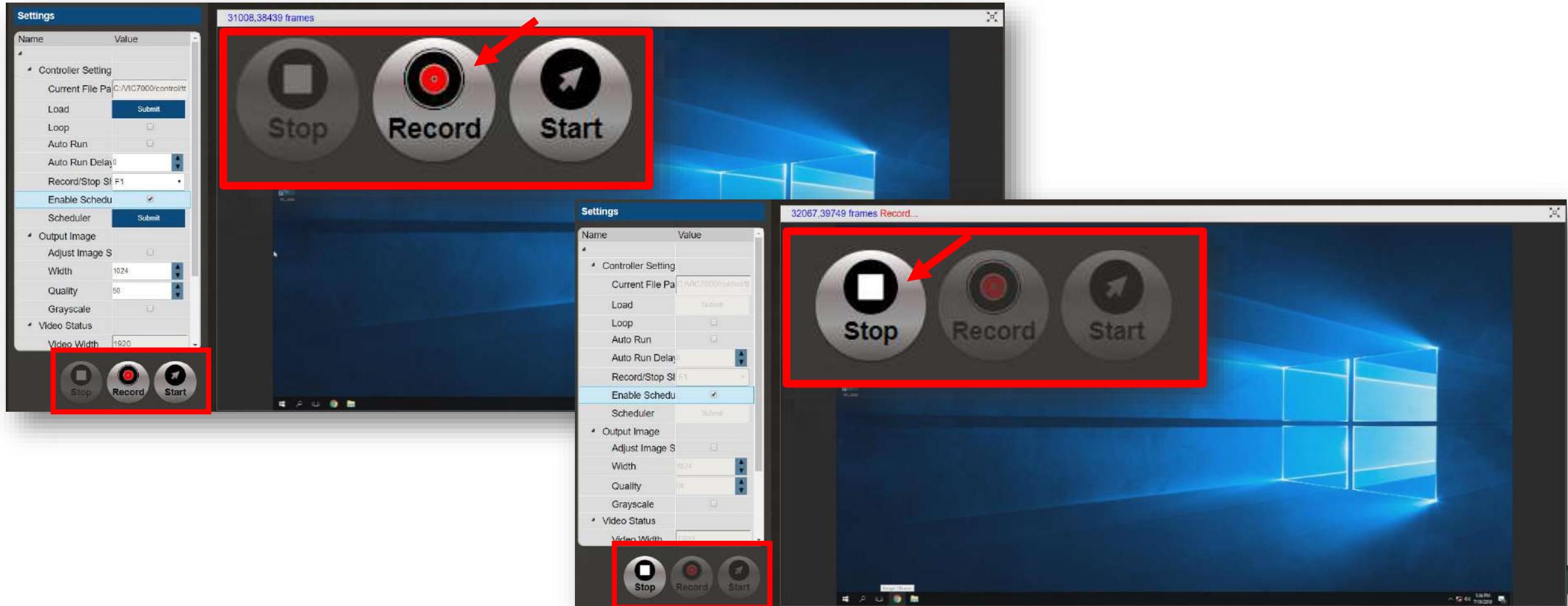
影像信號

影像輸入

<b>Output Image</b>	
Adjust Image Size An	<input type="checkbox"/>
Width	1024
Quality	50
Grayscale	<input type="checkbox"/>
<b>Video Status</b>	
Video Width	1920
Video Height	1080
Video Frame Rate	59.94
Video Signal	true
Video Input	DVI_D

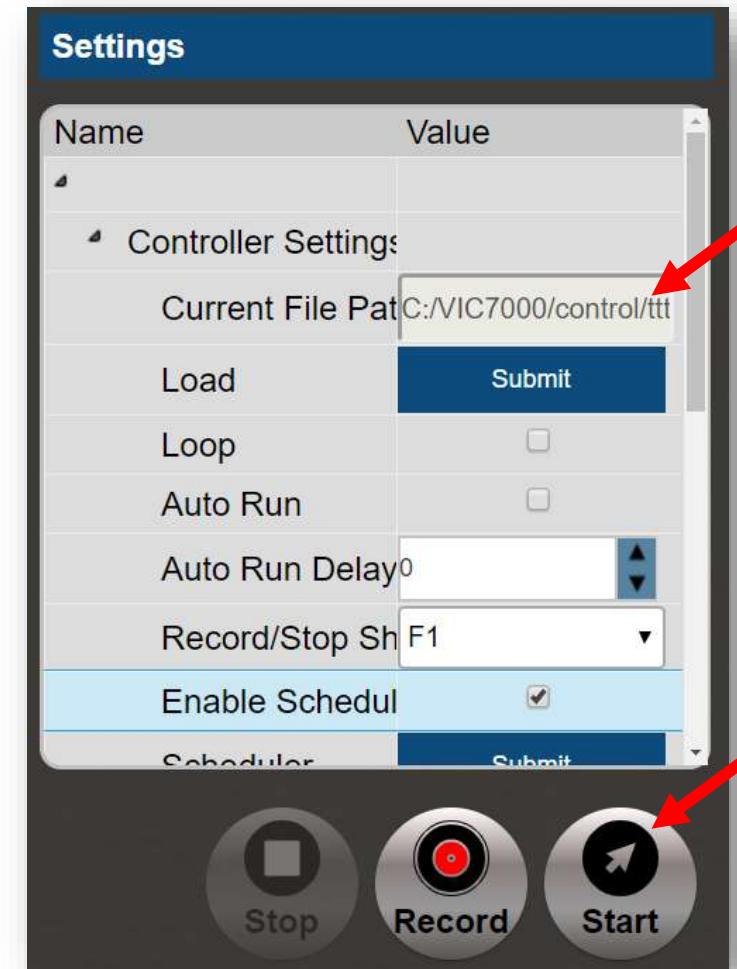
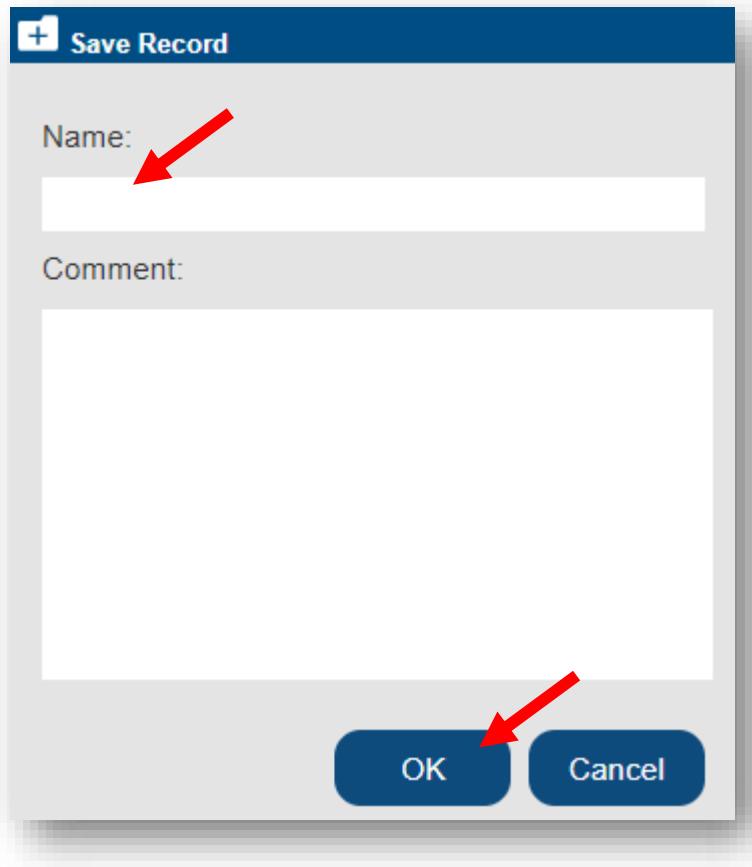
## • 控制錄製

2. 錄製動作：點擊錄製 or 按下錄製快捷鍵 → 完成滑鼠和鍵盤要錄製的操作 → 點擊停止 or 按下停止快捷鍵



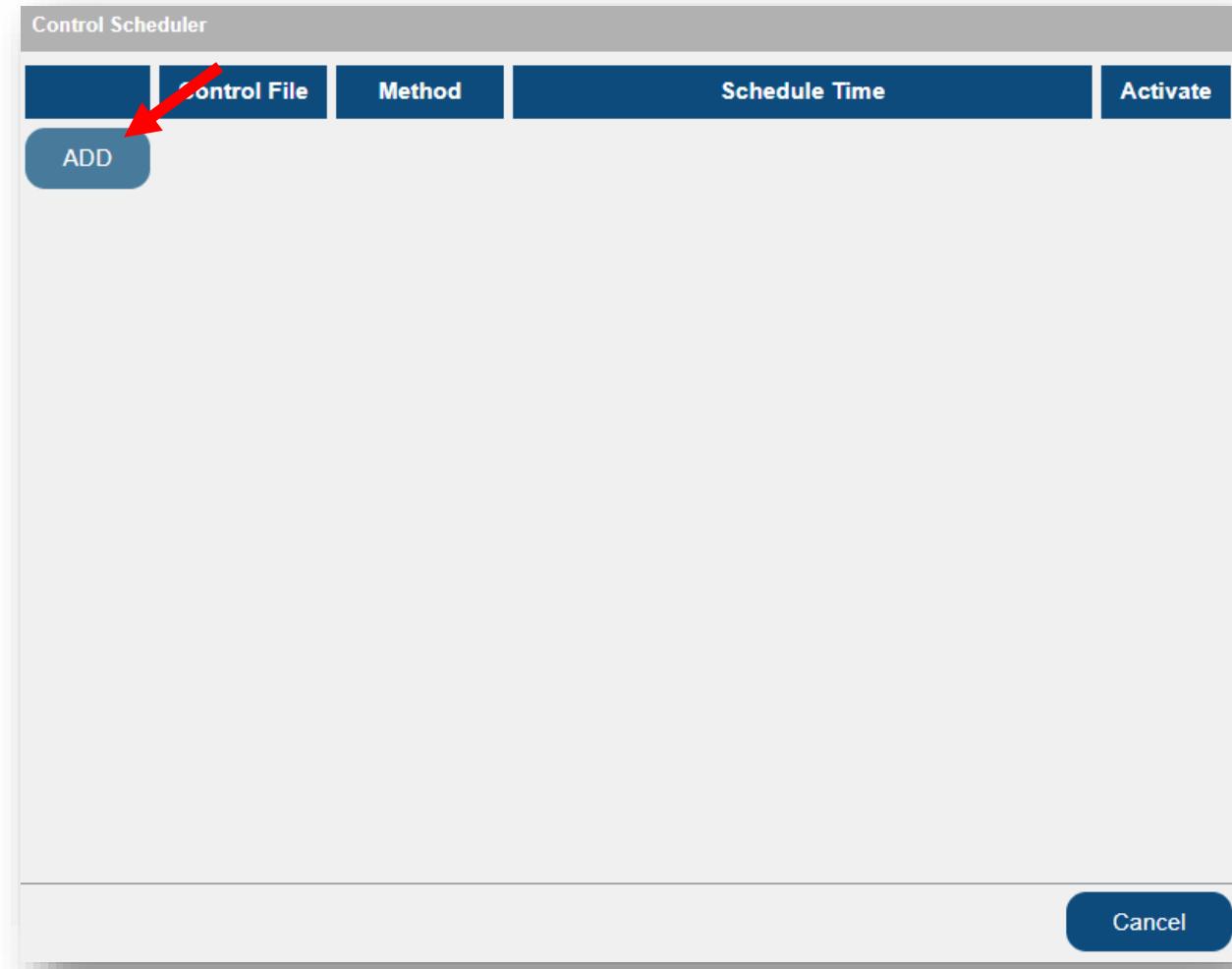
- 控制錄製

2. 錄製動作：輸入控制檔的名稱 → 點擊確定，此控制檔會被載入，點擊啟動進行模擬



- **控制設定**

- **排程設定**：點擊排程的執行鍵，可看到排程視窗 → 點擊增加



## • 控制設定

- **排程設定 – 間隔**：每隔多久時間執行一次

選擇控制檔

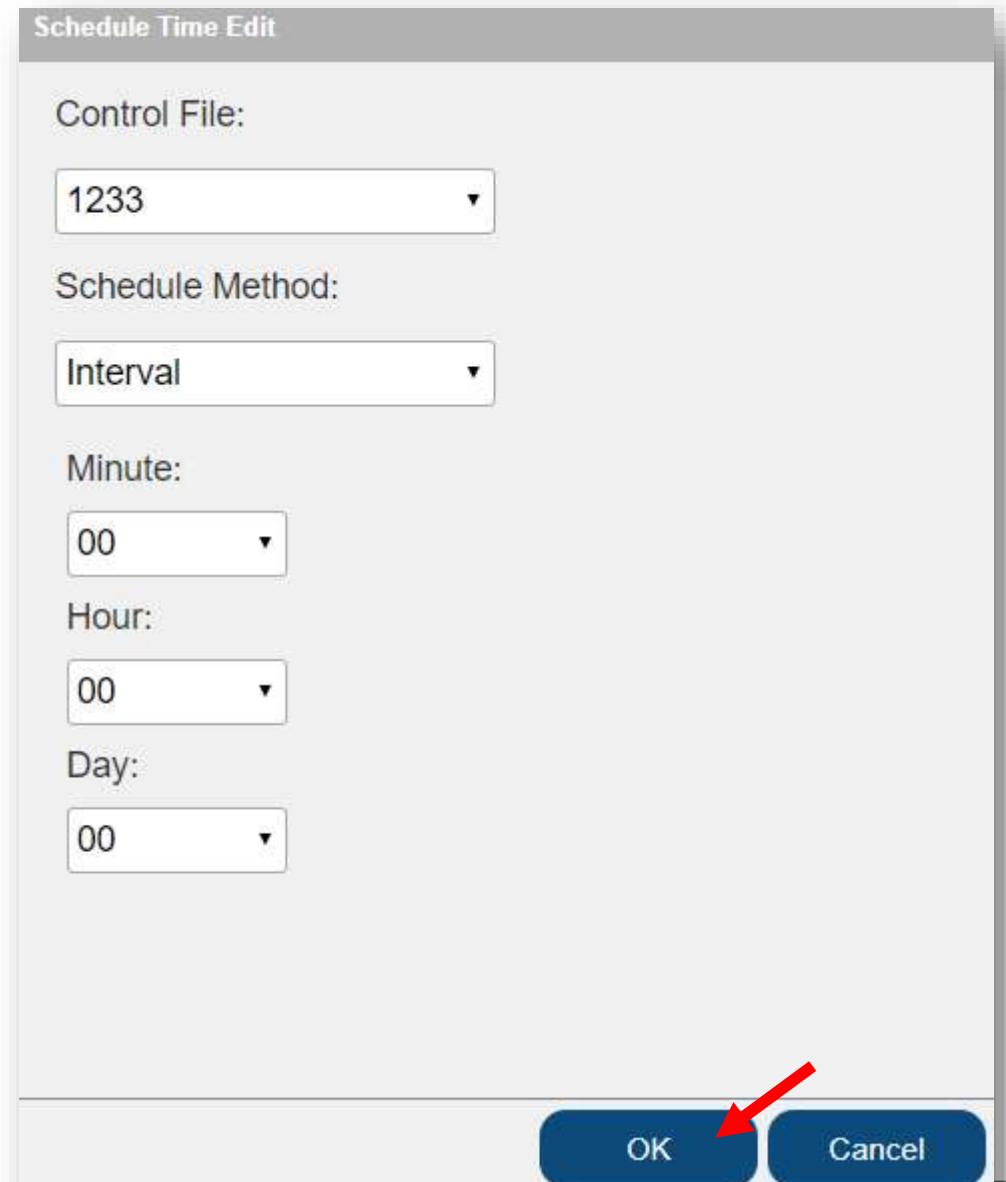
排程方法：間隔

分鐘

小時

日

設定結束後點擊確定



## • 控制設定

- **排程設定 – 時刻**：在指定時間執行一次

選擇控制檔

排程方法：時刻

分鐘

小時

日

若有設定日和星期，需 **同時滿足條件**，才會執行

Ex：設定日-5、星期-1，往後只有星期一

星期

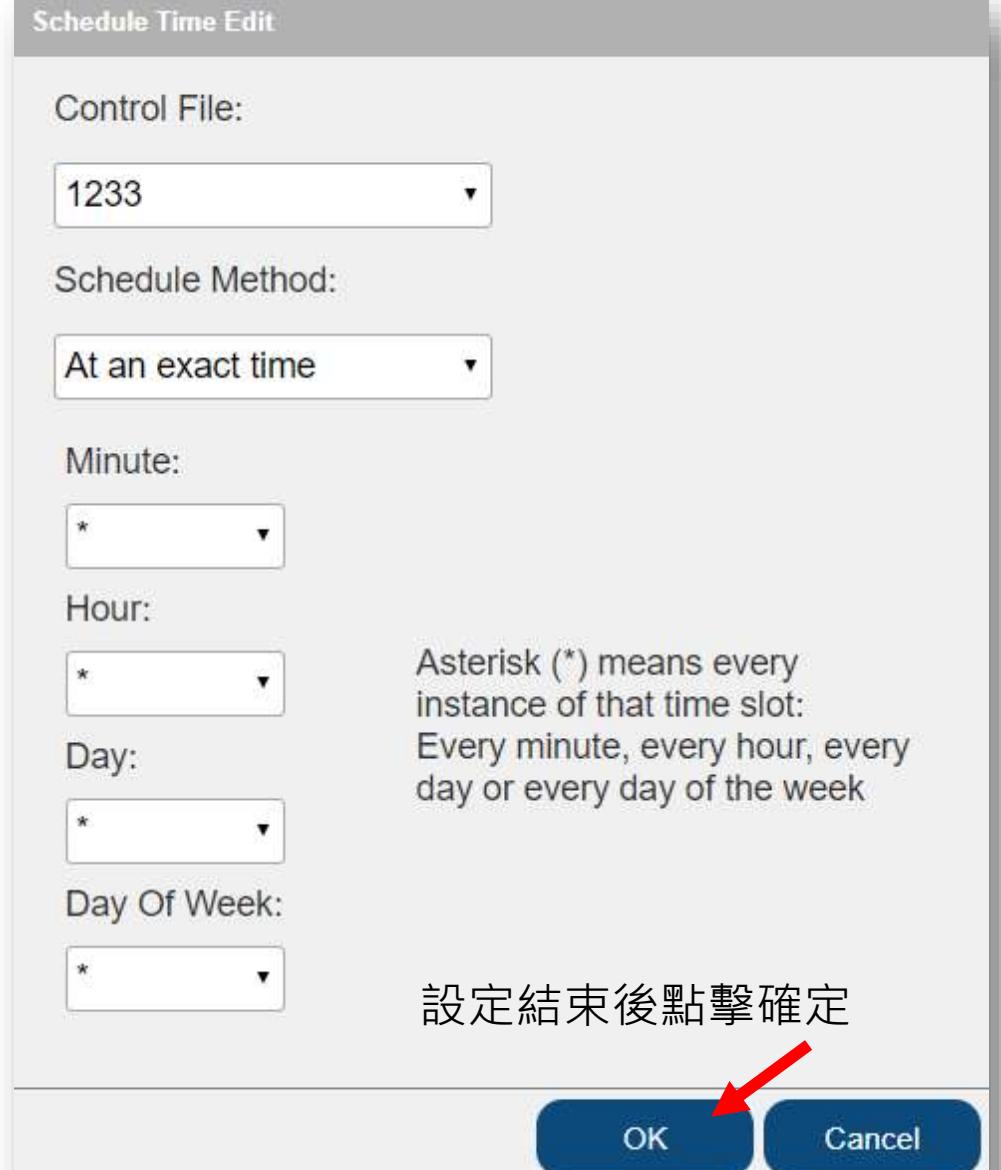
且剛好為當月5日時才會執行控制檔

\*：代表該欄位的每一個時間點

Ex：若設定為分鐘-\*、小時- 12、日- 25、星期- \*，

在每個月的 25 日的中午 12 點，每分鐘都會

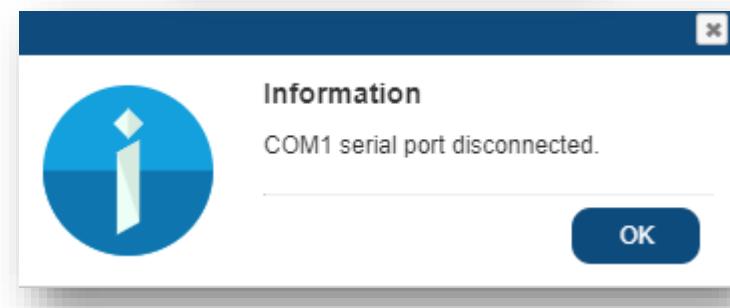
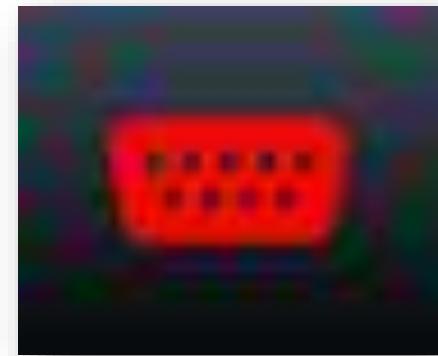
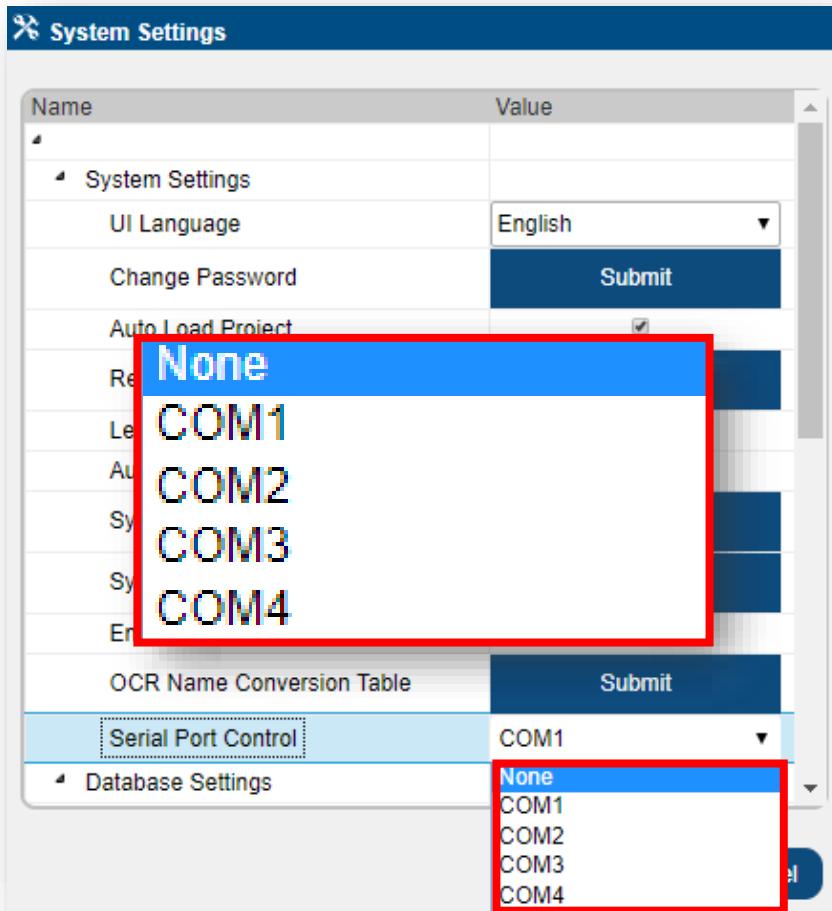
執行設定的控制檔





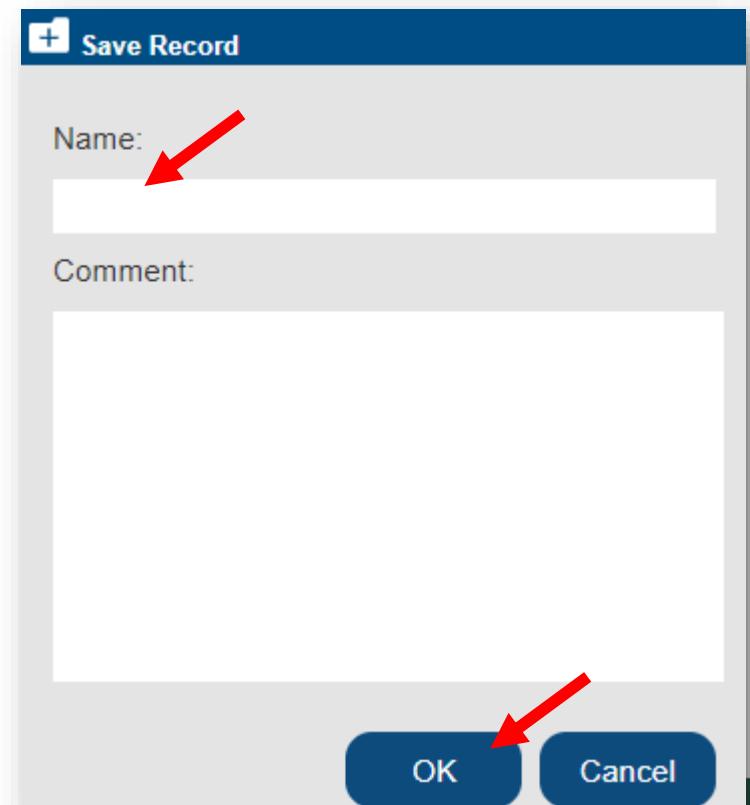
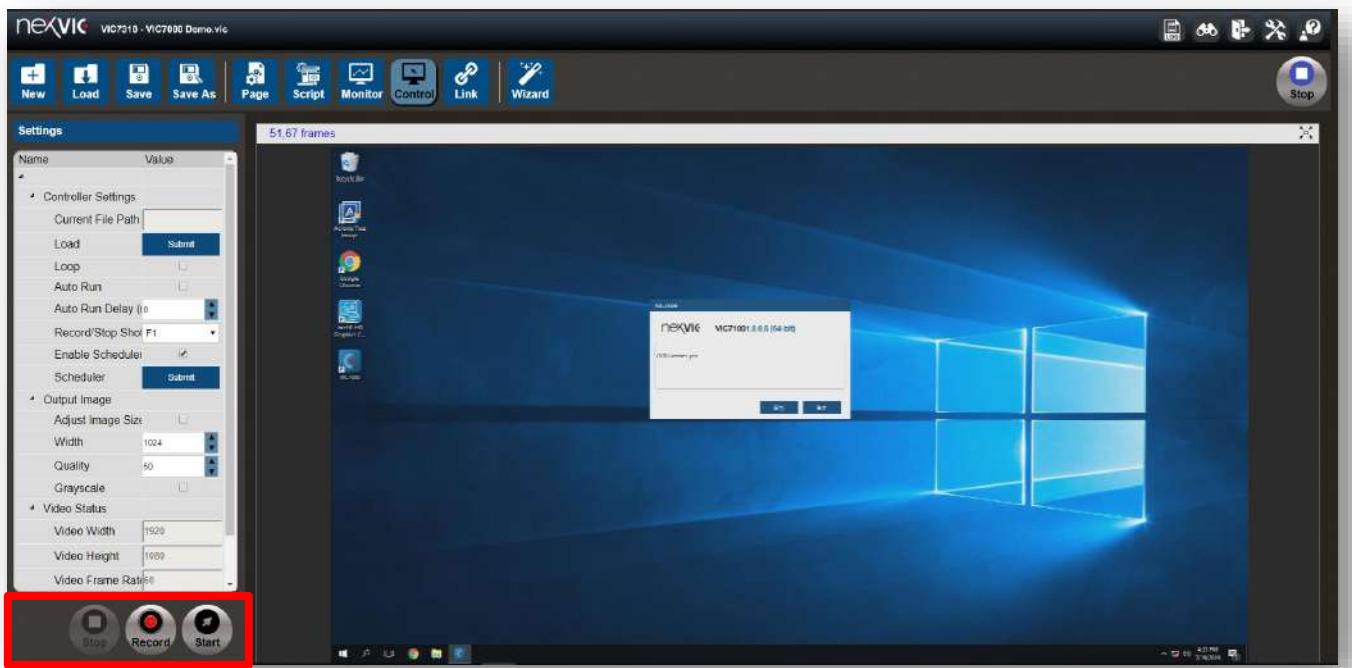
## • 序列埠控制

在設定視窗選擇要用的通訊埠控制端口，以隨附的 COM 線是否連接為準，顯示在右上工具列連接時，擷取資料和控制都會運作；斷開時，擷取資料和控制都會暫停



## • 進階控制 – 輸入

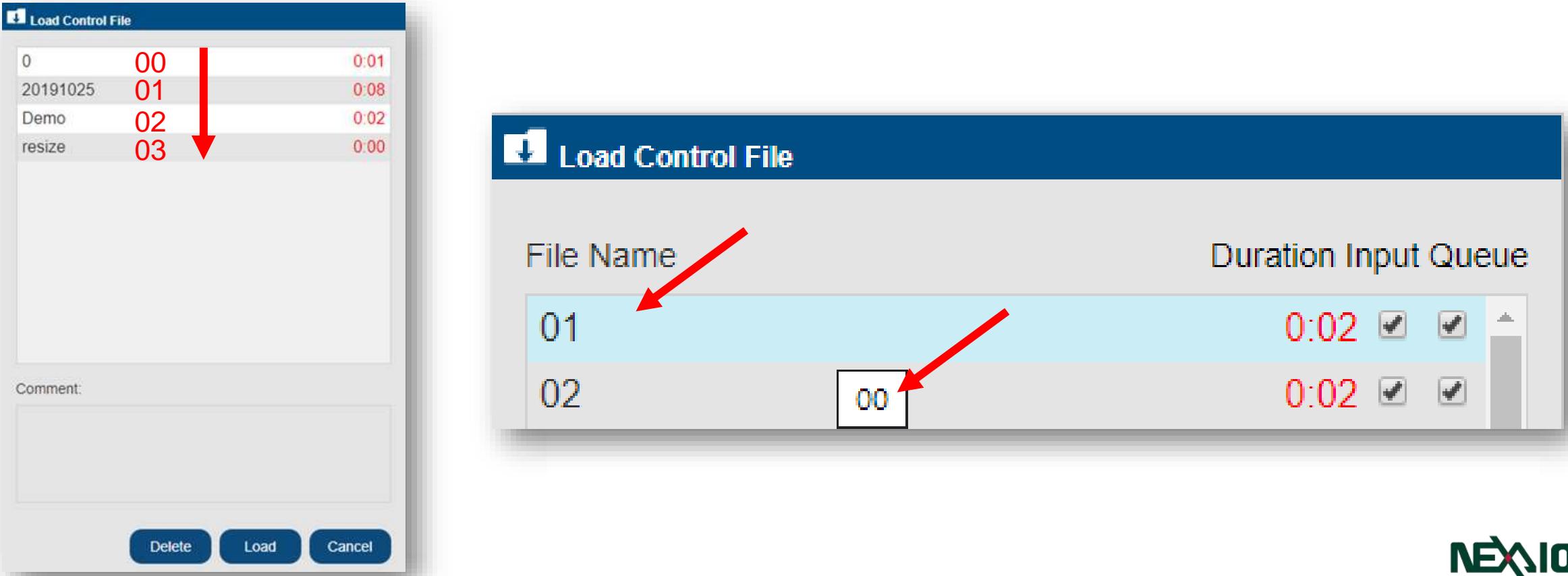
1. **錄製控制檔**：進入控制頁面 → 點擊錄製 or 按下錄製快捷鍵 → 完成要錄製的操作 → 點擊停止 or 按下停止快捷鍵 → 輸入控制檔的名稱 → 點擊確定  
控制檔結束時必須讓要輸入資料的欄位是可輸入的狀態



## • 進階控制 – 輸入

### 1. 錄製控制檔：進入載入控制檔視窗

控制檔順序如控制頁面內載入控制視窗列出的順序，從 00 開始由上往下編號；  
或是移動游標到控制檔的名稱上，會出現這個控制檔的編號。

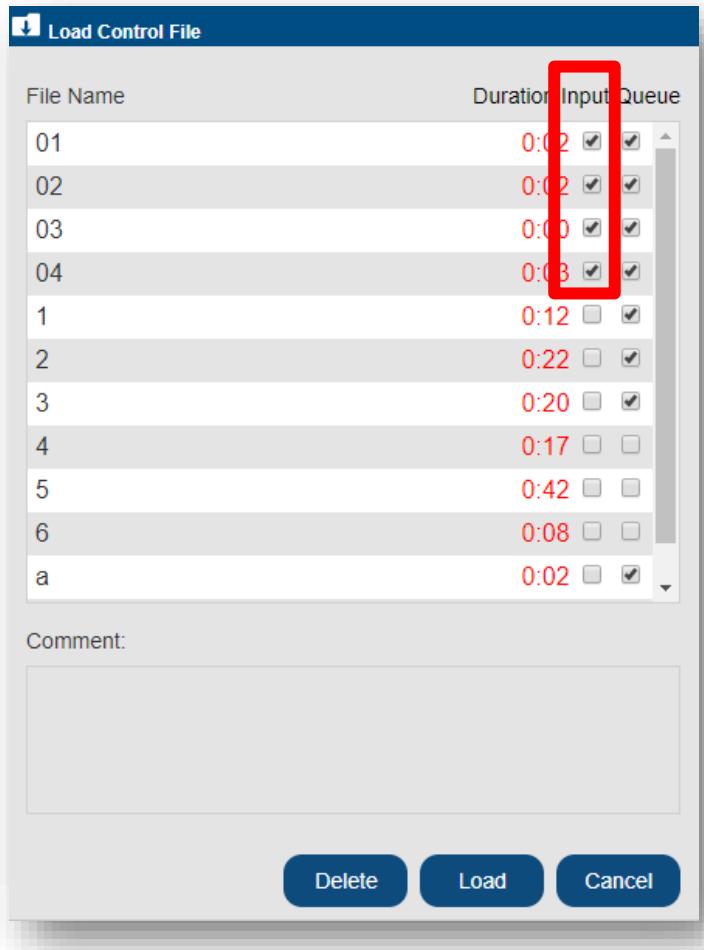


File Name	Duration	Input Queue
01	0:02	<input checked="" type="checkbox"/>
02	0:02	<input checked="" type="checkbox"/>

- 進階控制 – 輸入

2. 設定控制檔 – 輸入：進入載入控制檔視窗 → 勾選輸入

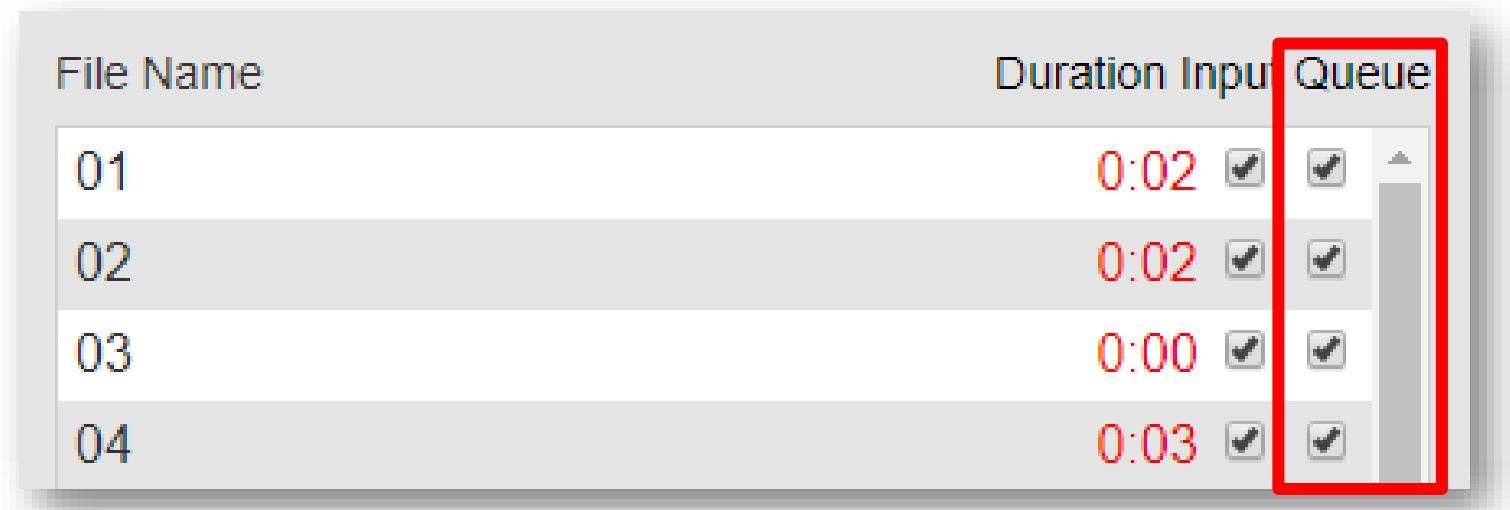
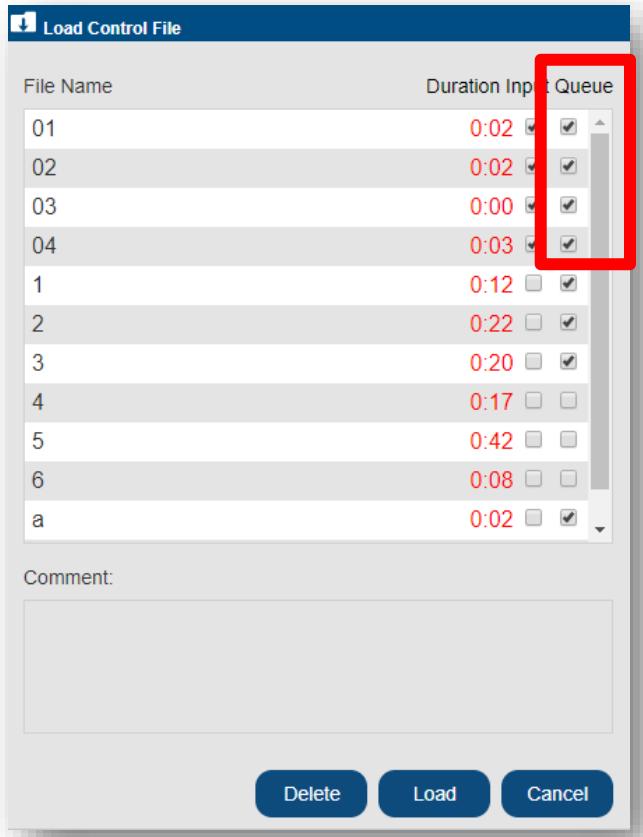
如果勾選輸入，VIC7000 會在控制檔執行結束後輸入儲存在對應欄位內的字元



- 進階控制 – 輸入

2. 設定控制檔 – 佇列：進入載入控制檔視窗 → 勾選佇列

預設狀況下，若前一個控制檔未執行結束就執行下一個控制檔，前一個控制檔會被直接**覆蓋掉**。若勾選佇列，該控制檔會在前一個控制檔執行結束後再執行

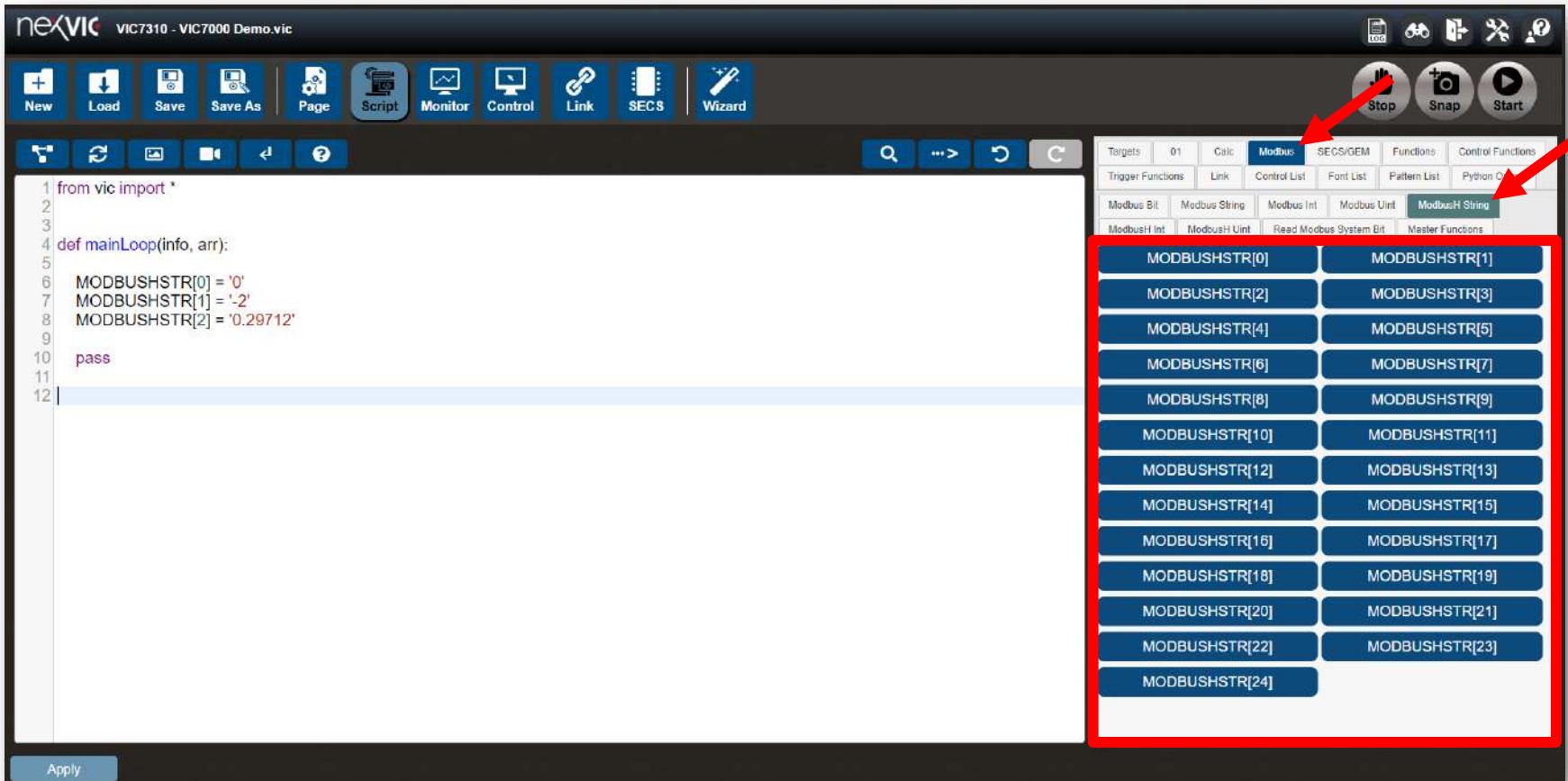


## • 進階控制 – 輸入

### 3. 設定輸入內容 – 腳本

：進入腳本頁面 → 設定 MODBUSHSTRn 的值

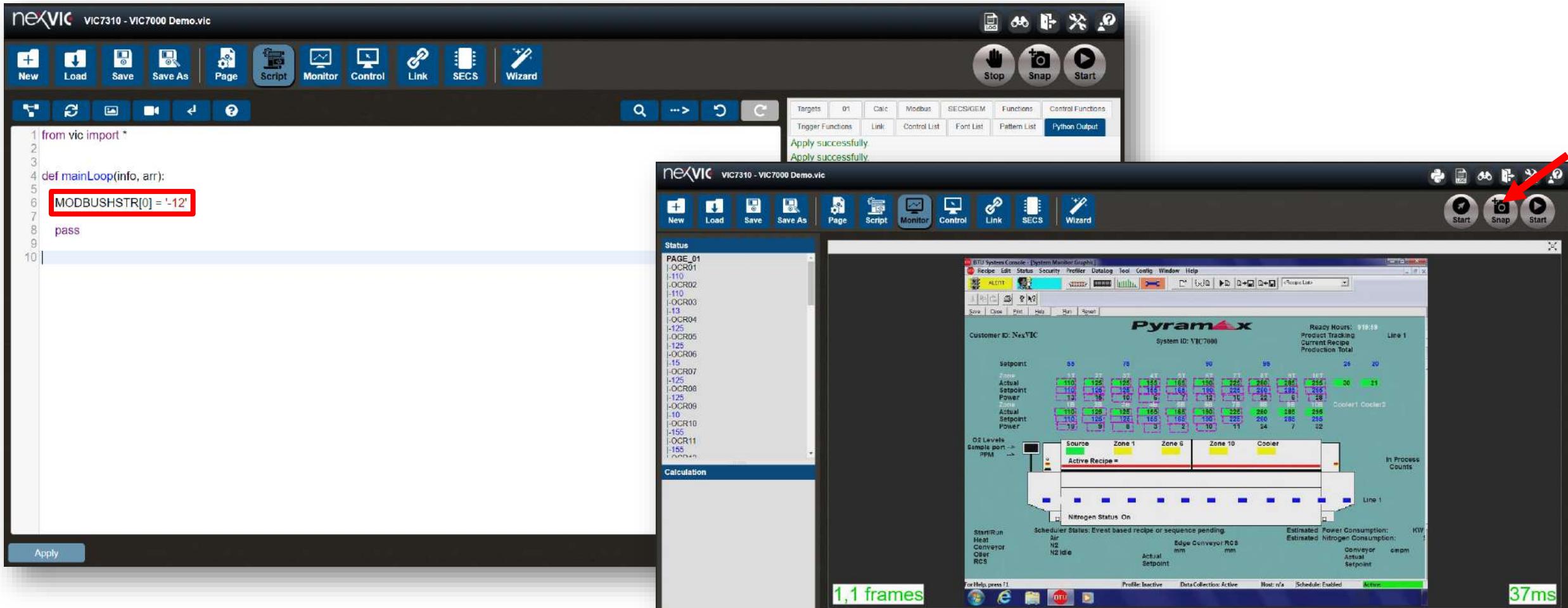
n 代表 MODBUSHSTR 的編號，MODBUSHSTRn 代表相同編號的控制檔執行結束輸入的內容。舉例來說，MODBUSHSTR00 就是編號 00 的控制檔執行結束後要輸入的內容



## • 進階控制 – 輸入

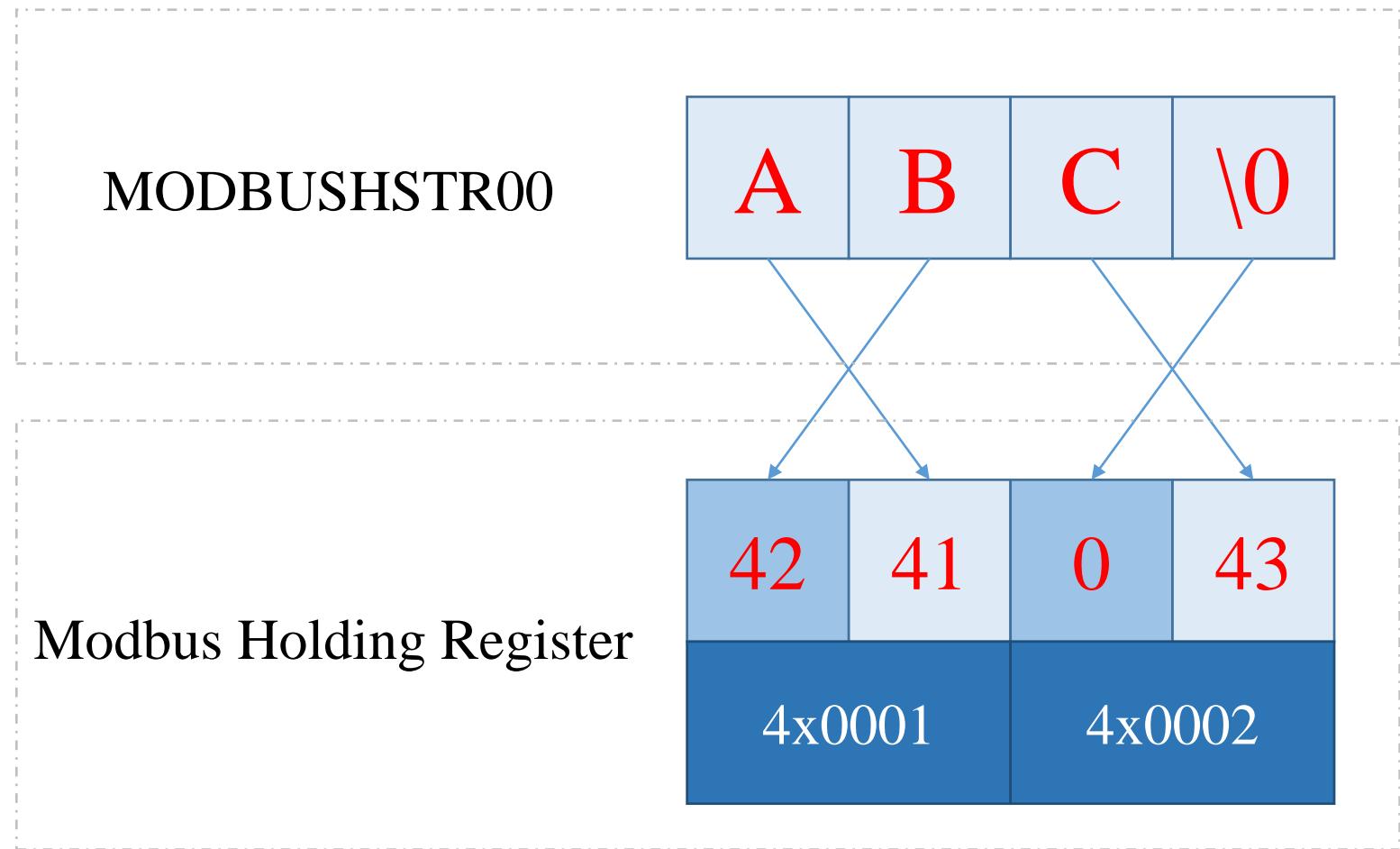
範例 – 設定輸入內容 – Script : 使用腳本來設定 MODBUSHSTR00 的值為 -12 ,

再到監視頁面點擊觸發，讓腳本被執行一次。



- 進階控制 – 輸入

3. 設定輸入內容 – Modbus TCP : VIC7000 是 little-endian 系統，也就是說資料要從低位元開始讀取，寫入亦要從低位元開始寫入



- 進階控制 – 輸入

3. 設定輸入內容 – Modbus TCP：設定 MODBUSHSTRn 的值，資料類型是 ASCII 碼(8 bits)。

一個寄存器(16 bits)可以儲存兩個ASCII碼，ASCII碼會自動被轉換成字元。n代表 MODBUSHSTR 的編號，MODBUSHSTRn 代表相同編號的控制檔執行結束輸入的內容。舉例來說，MODBUSHSTR00 的內容就是編號 00 的控制檔執行結束後要輸入的內容。

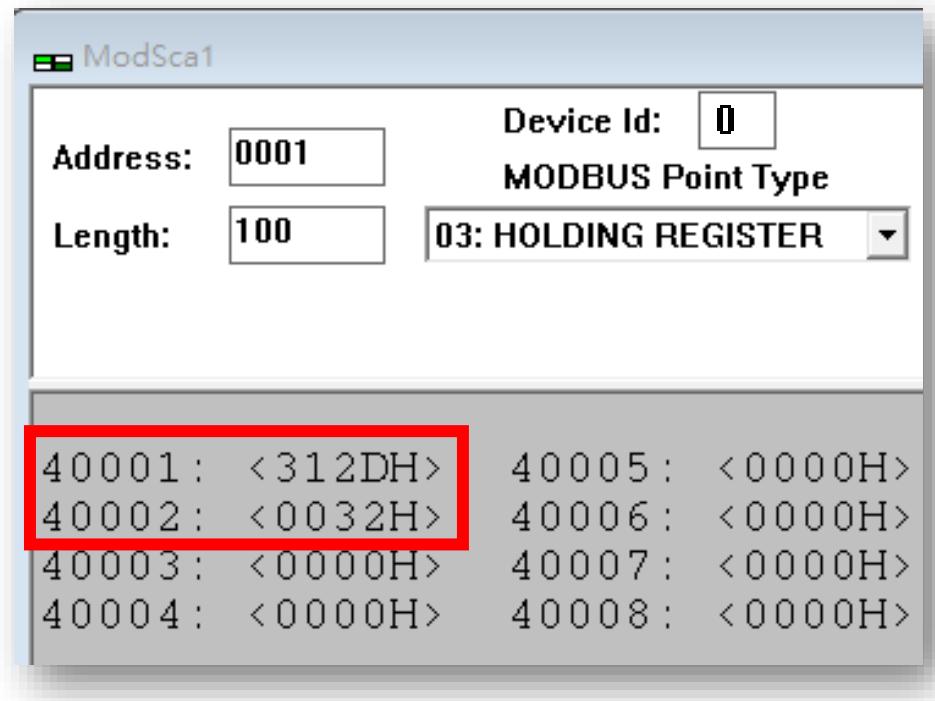
Holding Register (4X)	
Address Range	Content
40001 ~ 40051	MODBUSHSTR00
40052 ~ 40102	MODBUSHSTR01
40103 ~ 40153	MODBUSHSTR02
40154 ~ 40204	MODBUSHSTR03
40205 ~ 40255	MODBUSHSTR04

Holding Register (4X)	
Address Range	Content
40001 ~ 410200	MODBUSHSTR00 ~ 199

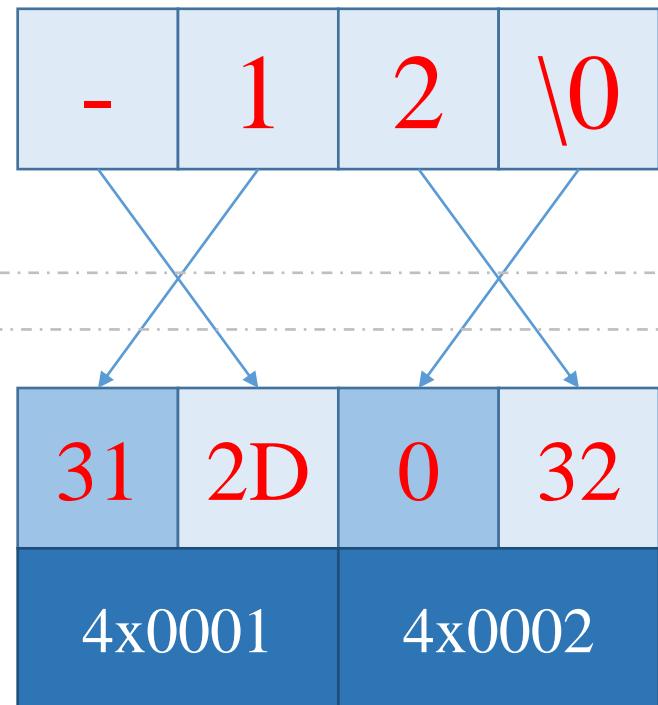
一個 MODBUSHSTR 可以儲存101個ASCII碼，也就是可以儲存101個字元。

- 進階控制 – 輸入

範例 – 設定輸入內容 – Modbus TCP : 站號為0，使用 Modscan (16進位模式)來設定 MODBUSHSTR00 的內容，設定的值為 -12



MODBUSHSTR00



Modbus Holding Register

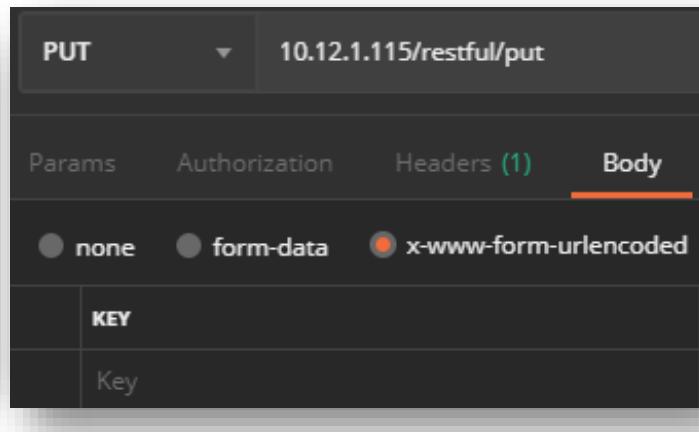
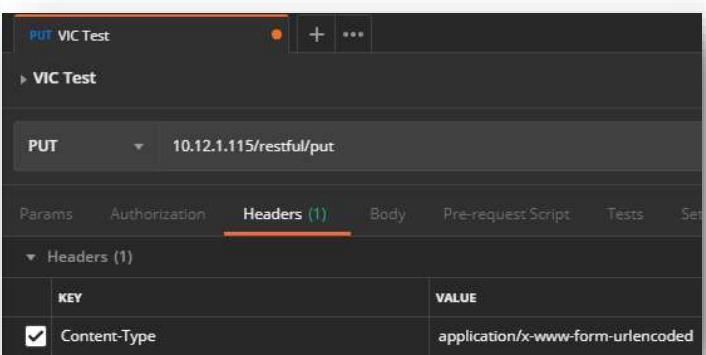
- 進階控制 – 輸入

3. 設定輸入內容 – RESTful API：使用 PUT 功能來設定要輸入的內容，index\_value 對應到控制檔的編號，其內容會在對應的控制檔執行結束後自動輸入。

舉例來說，index\_value 設定為 0 的 content 就是編號 00 的控制檔執行結束後要輸入的內容。  
這裡使用 Postman 作為呼叫工具

Headers : Content-Type : application/x-www-form-urlencoded

Body : x-www-form-urlencoded



	設定要輸入的內容
<b>username</b>	admin or user
<b>password</b>	123456 (default)
<b>cmd</b>	req_write_modbush_string
<b>index_value</b>	0 ~ 199
<b>content</b>	要輸入的內容

- 進階控制 – 輸入

範例 – 設定輸入內容 – RESTful API : 使用 Postman 來設定 index\_value 0 的 content 為 -12，執行後檢查回傳結果

PUT 192.168.0.100/restful/put

Params Authorization Headers (9) **Body** Pre-request Script Tests Setting

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
username	admin
password	123456
cmd	req_write_modbush_string
index_value	0
content	-12

Body Cookies Headers (2) Test Results

Pretty Raw Preview Visualize BETA

```
1
2
3
4
"login_success": "1",
"rsp_status": "1"
```

<b>login_success</b>	是否登入成功 1 = 成功，0 = 失敗
<b>rsp_status</b>	是否成功執行指令 1 = 成功，0 = 失敗

- 進階控制 – 輸入

3. 設定輸入內容 – SQL：要輸入的內容須設定在 vic 資料庫的 controller\_table 內，id 對應控制檔的編號，在對應的控制檔執行後會自動輸入該欄位的 value

輸入時，VIC7000 會優先輸入使用透過腳本、Modbus TCP 和 RESTful API 設定的內容  
舉例來說，id 0 欄位內的 value 就是編號 00 的控制檔執行結束後要輸入的內容

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the SCHEMAS section, with 'vic' selected. Under 'Tables could not be fetched', there are three entries: 'controller\_table', 'key\_table', and 'ocr\_table'. A red arrow points to the 'controller\_table' entry. The main workspace shows the 'controller\_table' tab selected, with the following SQL query in the SQL Editor:

```
1 •  SELECT * FROM vic.controller_table;
```

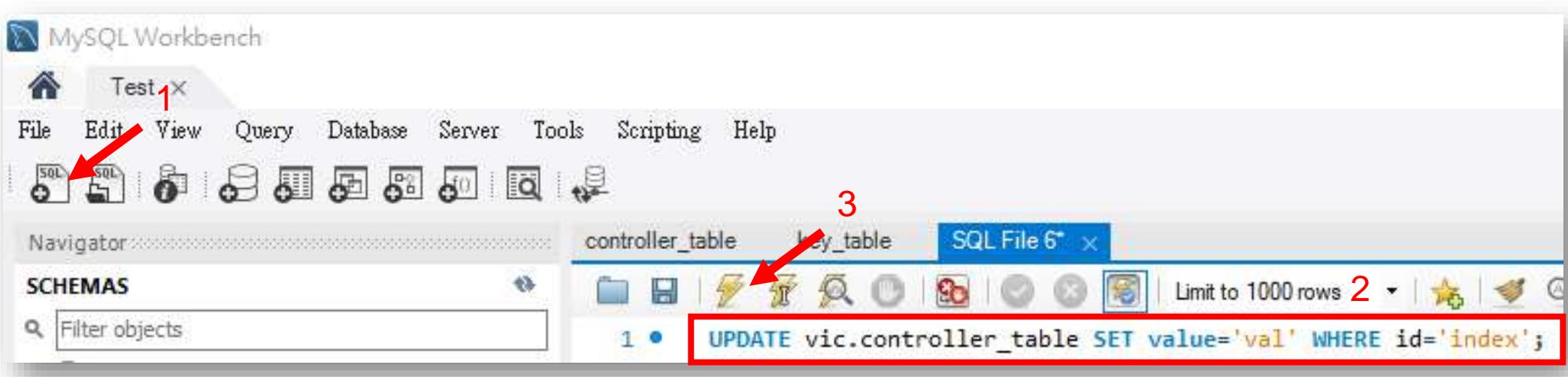
Below the SQL editor is a Result Grid showing the data from the controller\_table:

	id	value	extra
▶	0		NULL
	1	12	NULL
	2	-4	NULL

- 進階控制 – 輸入

3. 設定輸入內容 – SQL : 使用 SQL 語法的 UPDATE 來設定要輸入的內容，  
index 就是控制檔的編號，val 就是要輸入的內容  
新增指令頁 → 輸入指令 → 點擊執行

```
UPDATE vic.controller_table SET value='val' WHERE id='index';
```

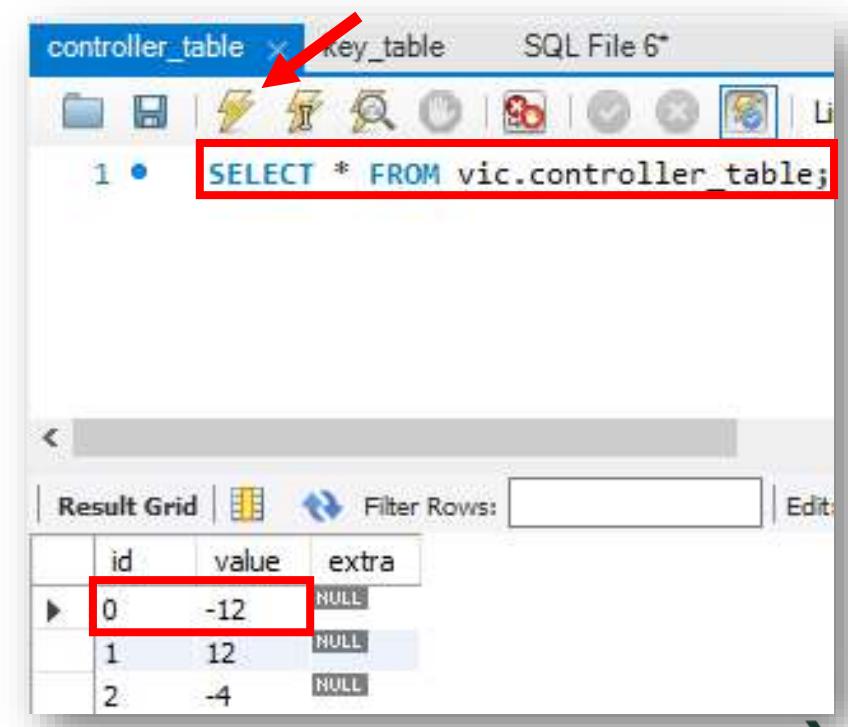


- 進階控制 – 輸入

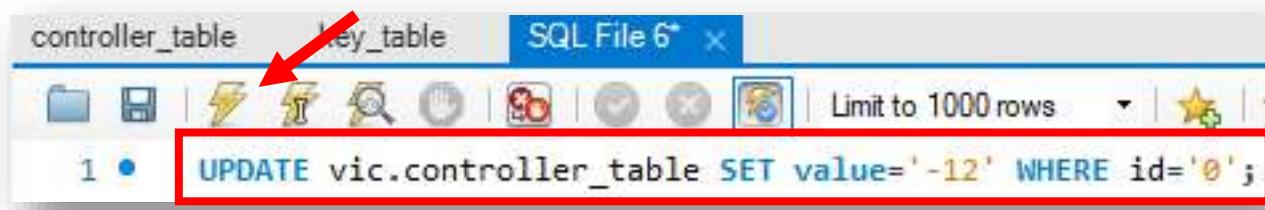
範例 – 設定輸入內容 – SQL : 使用 MySQL Workbench 來設定 id 0 的 value 為 -12

執行後可使用 SELECT 來檢查資料被正確寫入

```
SELECT * FROM vic.controller_table;
```



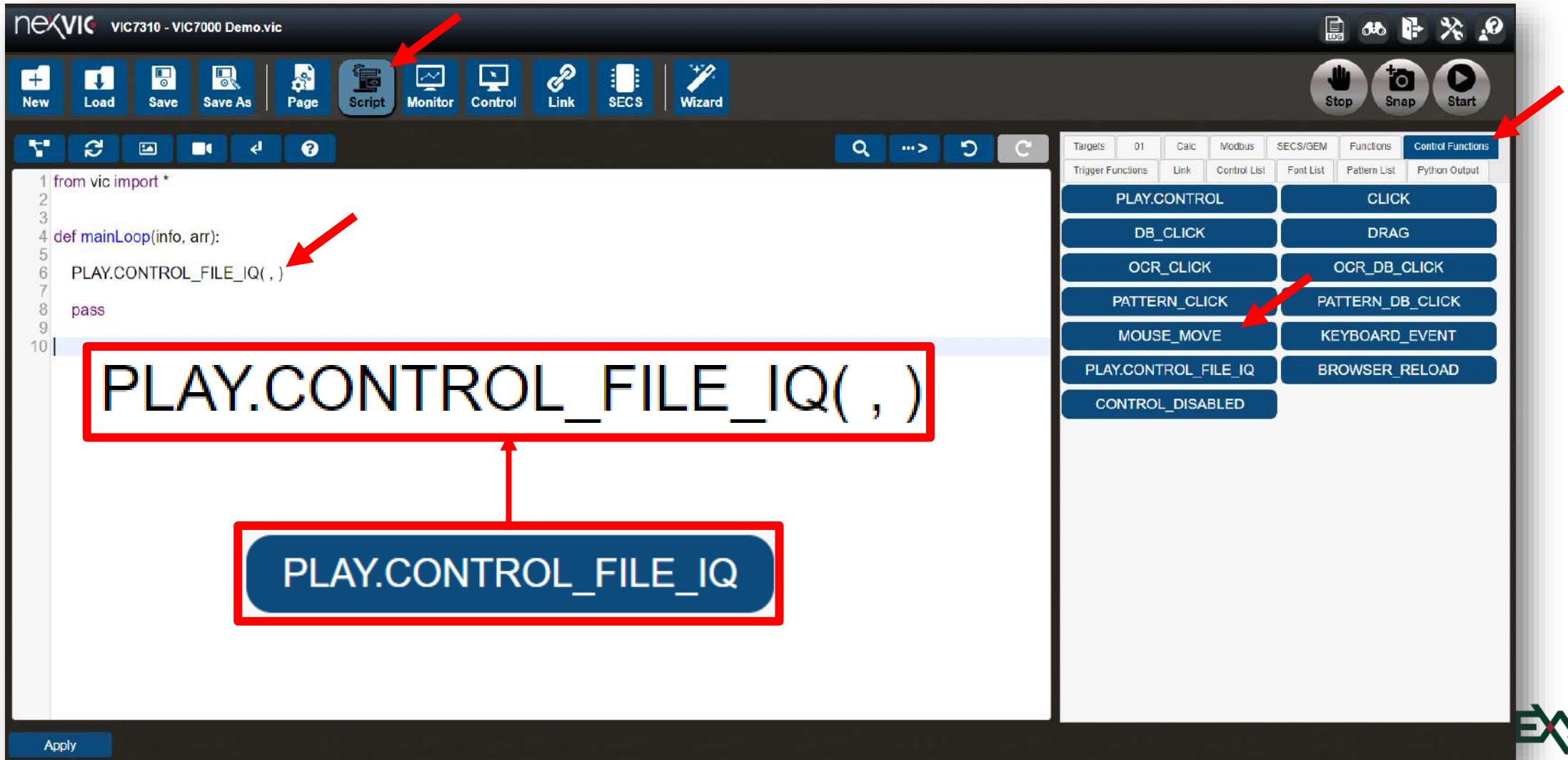
	id	value	extra
▶	0	-12	NULL
	1	12	NULL
	2	-4	NULL



	id	value	extra
▶	0	-12	NULL
	1	12	NULL
	2	-4	NULL

- 進階控制 – 輸入

4. 啟動控制檔 – 腳本：進入腳本頁面 → 在函式內選擇 PLAY.CONTROL\_FILE\_IQ



## • 進階控制 – 輸入

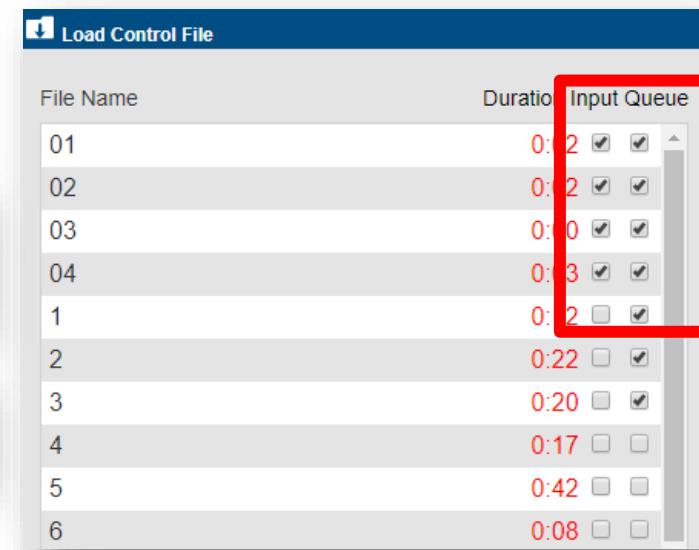
- **PLAY.CONTROL\_FILE\_IQ** : 使用輸入及佇列屬性執行特定控制檔

1. `control_file_name` : 設定要執行的控制檔，用控制檔列表選擇
2. `sec` : 多少秒內不再執行 (int)
  - `offset_x` : 控制 x 座標偏移，預設為 0
  - `offset_y` : 控制 y 座標偏移，預設為 0
  - `sync` : 是否為同步控制，預設為非同步控制

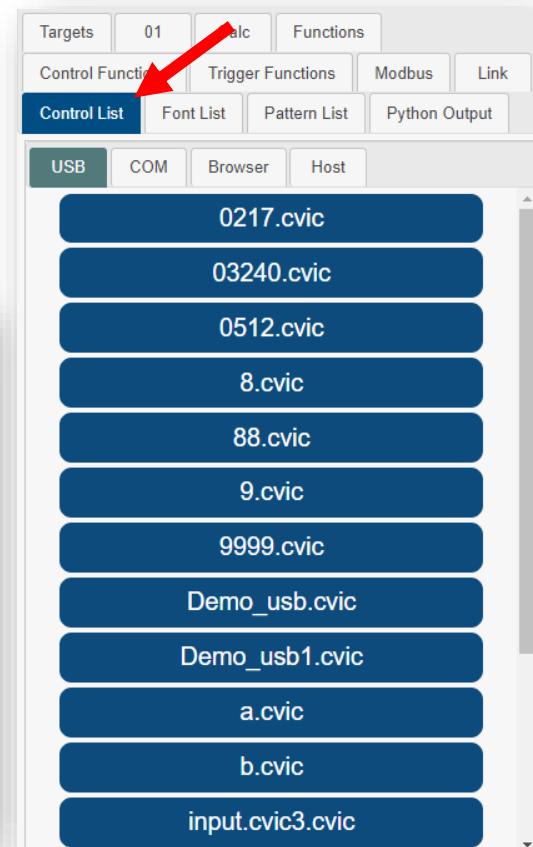
\*若要啟用輸入或佇列功能，需到載入控制檔視窗內開啟



**PLAY.CONTROL\_FILE\_IQ( , );**

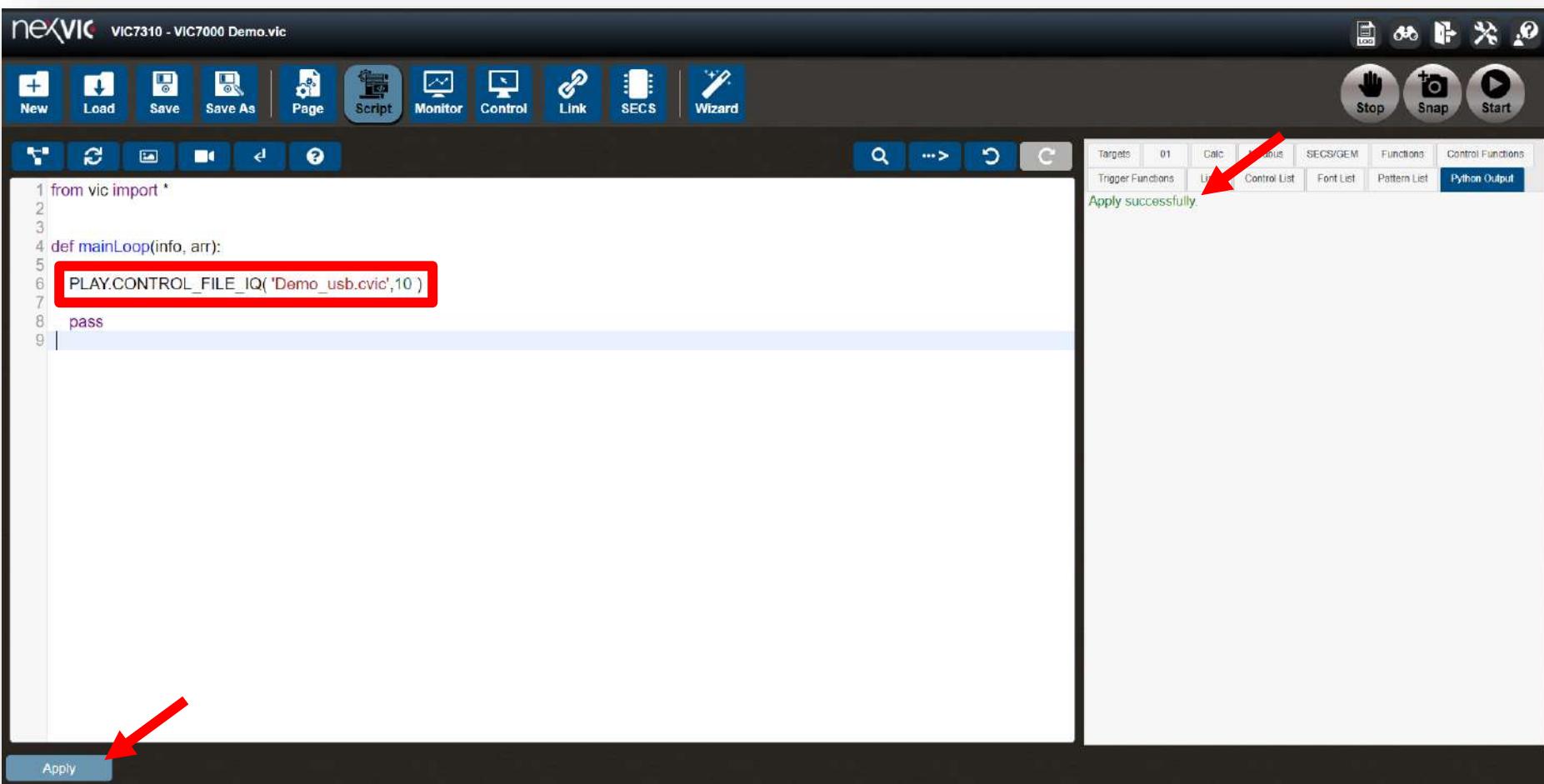


控制檔列表



## • 進階控制 – 輸入

範例 – 啟動控制檔 – 腳本 : 使用 PLAY.CONTROL\_FILE\_IQ 啟動名稱 00 的控制檔  
編寫結束後要儲存變更，腳本才會生效



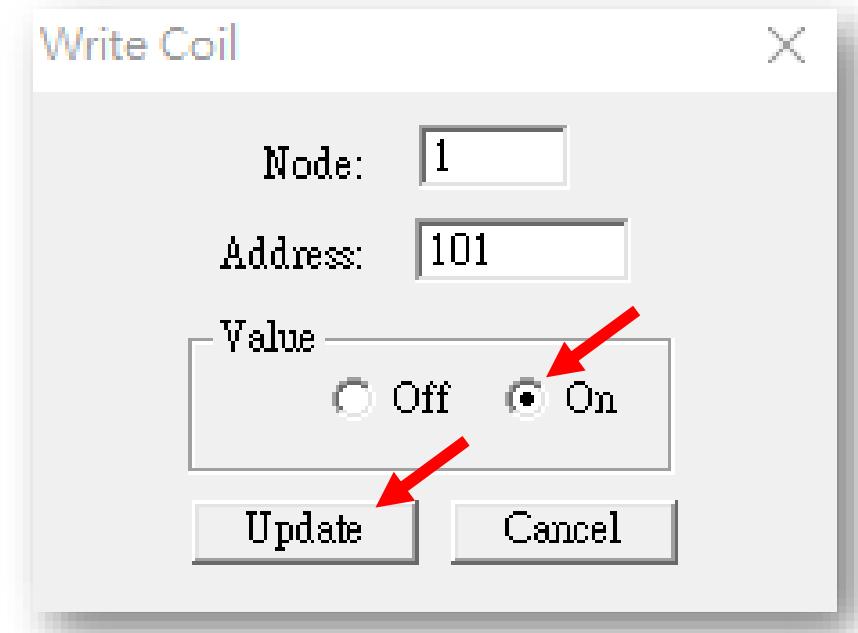
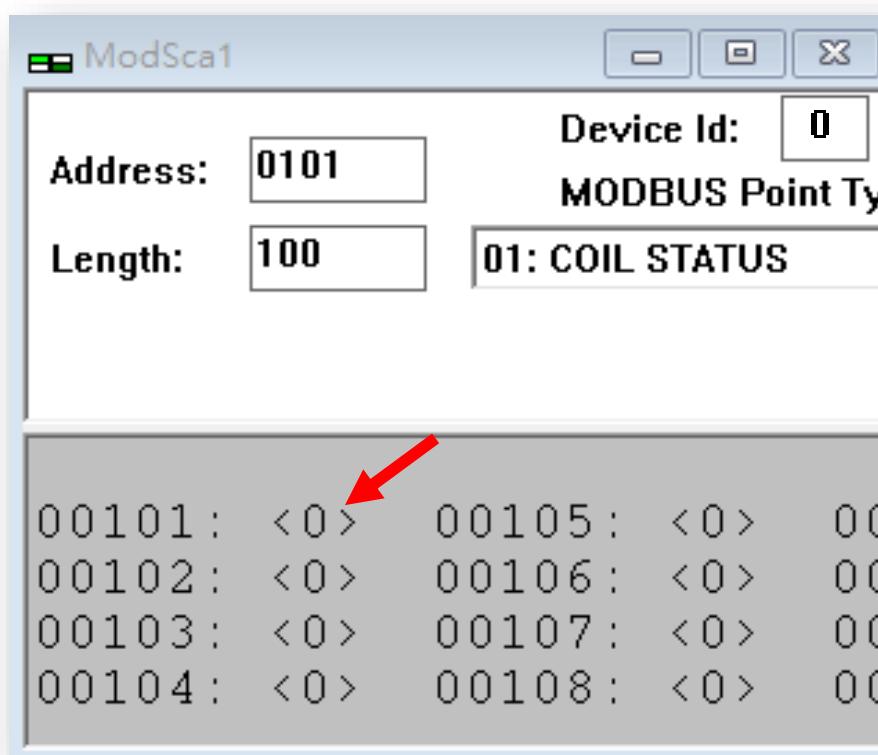
- 進階控制 – 輸入

4. 啟動控制檔 – Modbus TCP : 改變 Modbus 位址的值，可以啟動特定的控制檔

Coil Status (0X)	
Address	Content
00101 ~ 00300	運行編號00 ~ 199 的控制檔， 1 = 運行，0 = 停止

- 進階控制 – 輸入

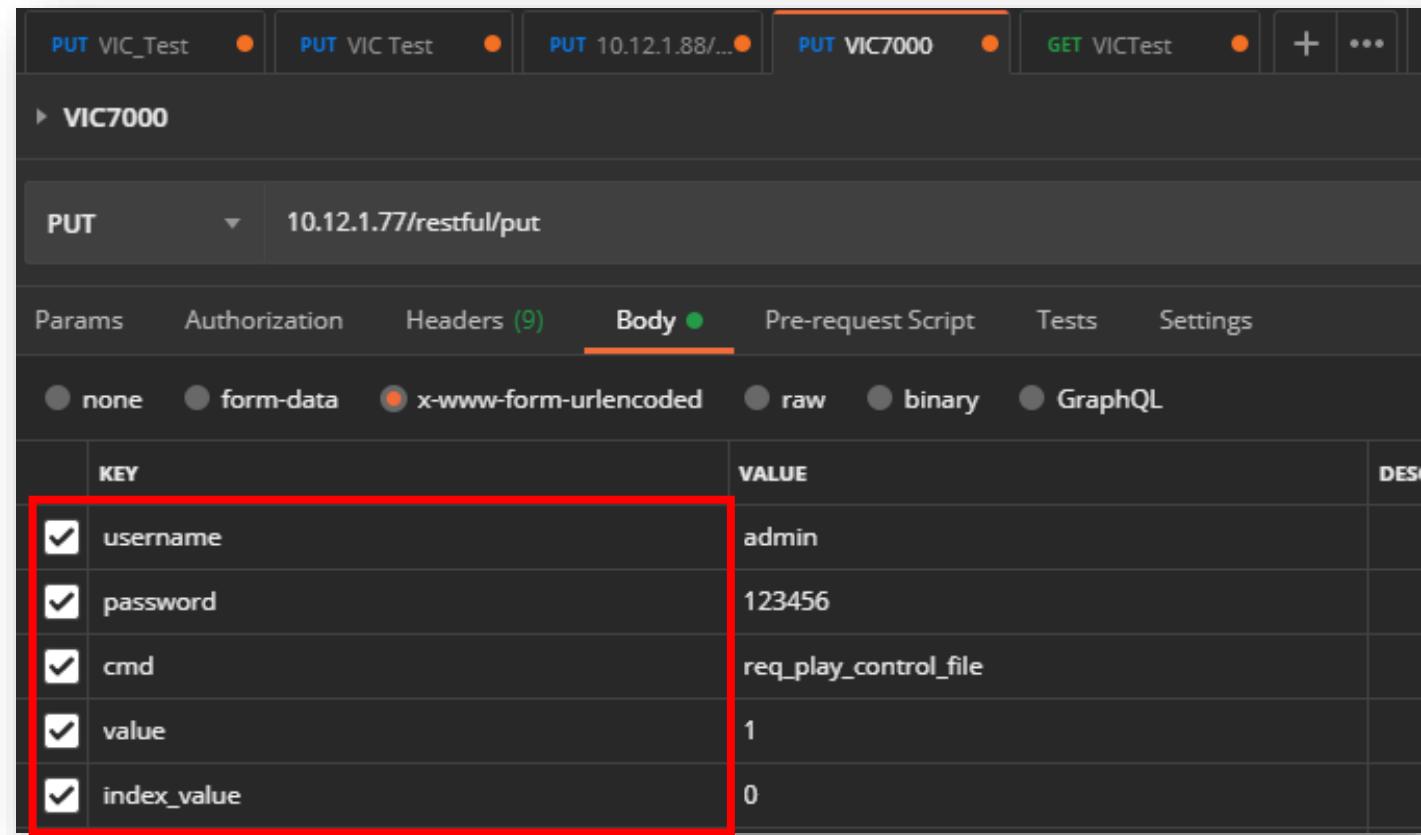
範例 – 啟動控制檔 – Modbus TCP：站號為0，連續點擊位址 00101 值的欄位兩下，改變該位址的值 (On)，即可啟動編號 00 的控制檔



- 進階控制 – 輸入

## 4. 啟動控制檔 – RESTful API : 用 PUT 功能來啟動控制檔

運行編號 i 控制檔	
<b>username</b>	admin or user
<b>password</b>	123456 (default)
<b>cmd</b>	req_play_control_file
<b>value</b>	1 = 運行 · 0 = 停止
<b>index_value</b>	0 ~ 199

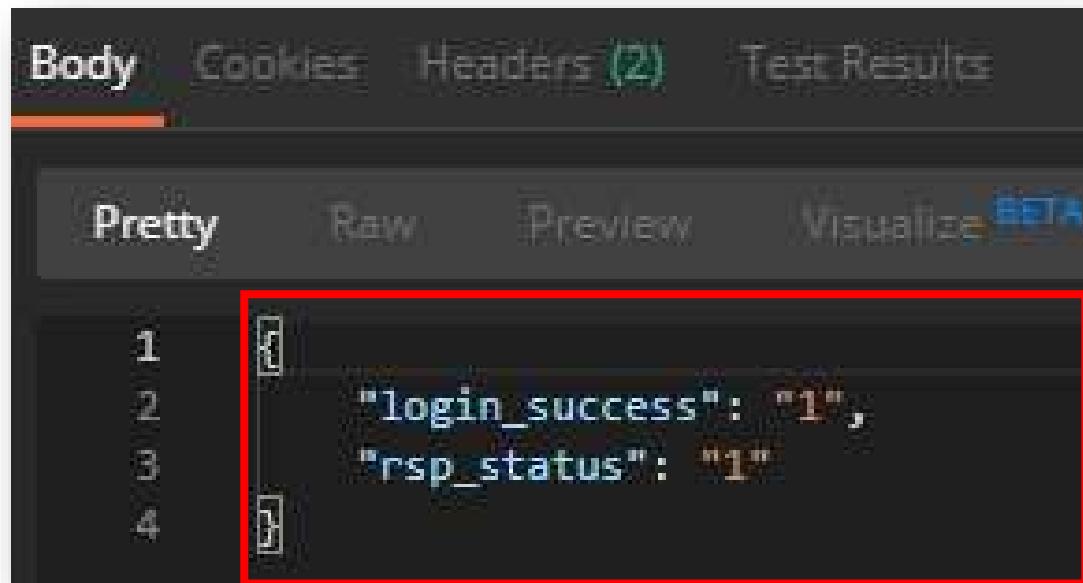


The screenshot shows the Postman application interface. A red box highlights the 'Body' tab, which is selected. The request URL is '10.12.1.77/restful/put'. The 'Body' tab is set to 'x-www-form-urlencoded'. The data table contains the following key-value pairs:

KEY	VALUE
username	admin
password	123456
cmd	req_play_control_file
value	1
index_value	0

- 進階控制 – 輸入

4. 啟動控制檔 – RESTful API : 檢查回傳結果

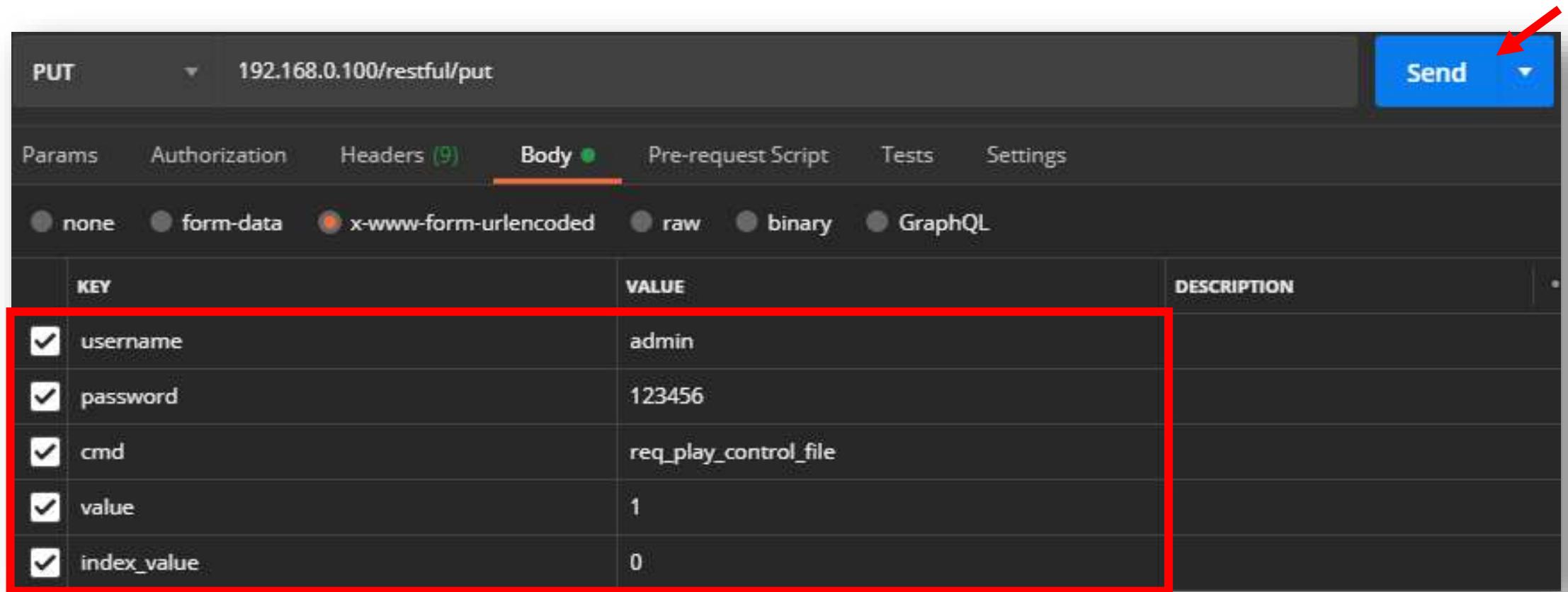


```
1
2
3
4
{
  "login_success": "1",
  "rsp_status": "1"
}
```

<b>login_success</b>	是否登入成功 1 = 成功 , 0 = 失敗
<b>rsp_status</b>	是否成功執行指令 1 = 成功 , 0 = 失敗

- 進階控制 – 輸入

範例 – 啟動控制檔 – RESTful API : 啟動編號為 00 的控制檔，設定好參數之後發送請求



PUT      192.168.0.100/restful/put

Send

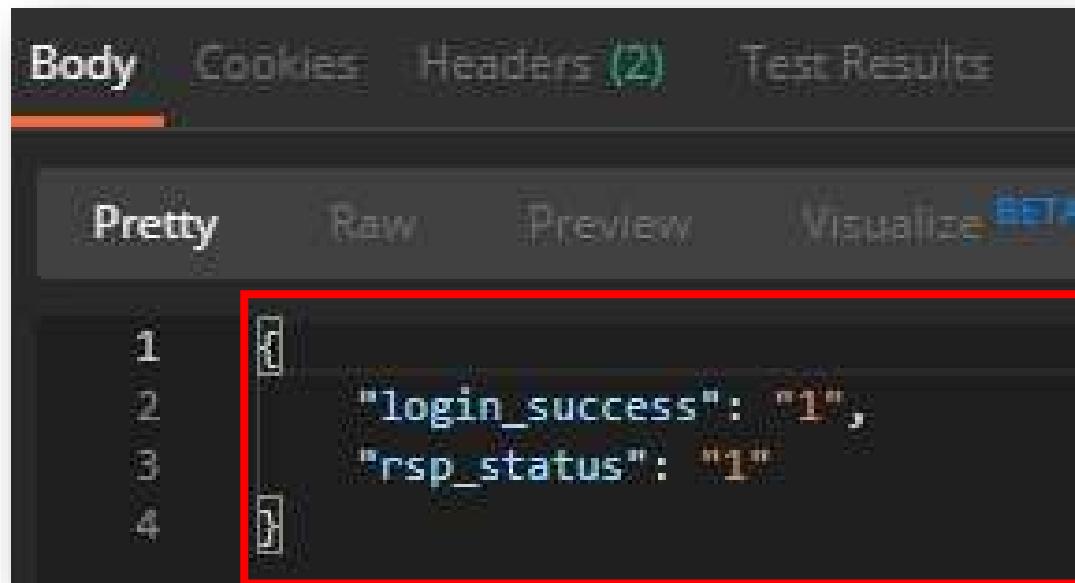
Params    Authorization    Headers (9)    Body    Pre-request Script    Tests    Settings

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	admin	
<input checked="" type="checkbox"/> password	123456	
<input checked="" type="checkbox"/> cmd	req_play_control_file	
<input checked="" type="checkbox"/> value	1	
<input checked="" type="checkbox"/> index_value	0	

- 進階控制 – 輸入

範例 – 啟動控制檔 – RESTful API : 發送後再檢查回傳結果



Body Cookies Headers (2) Test Results

Pretty Raw Preview Visualize

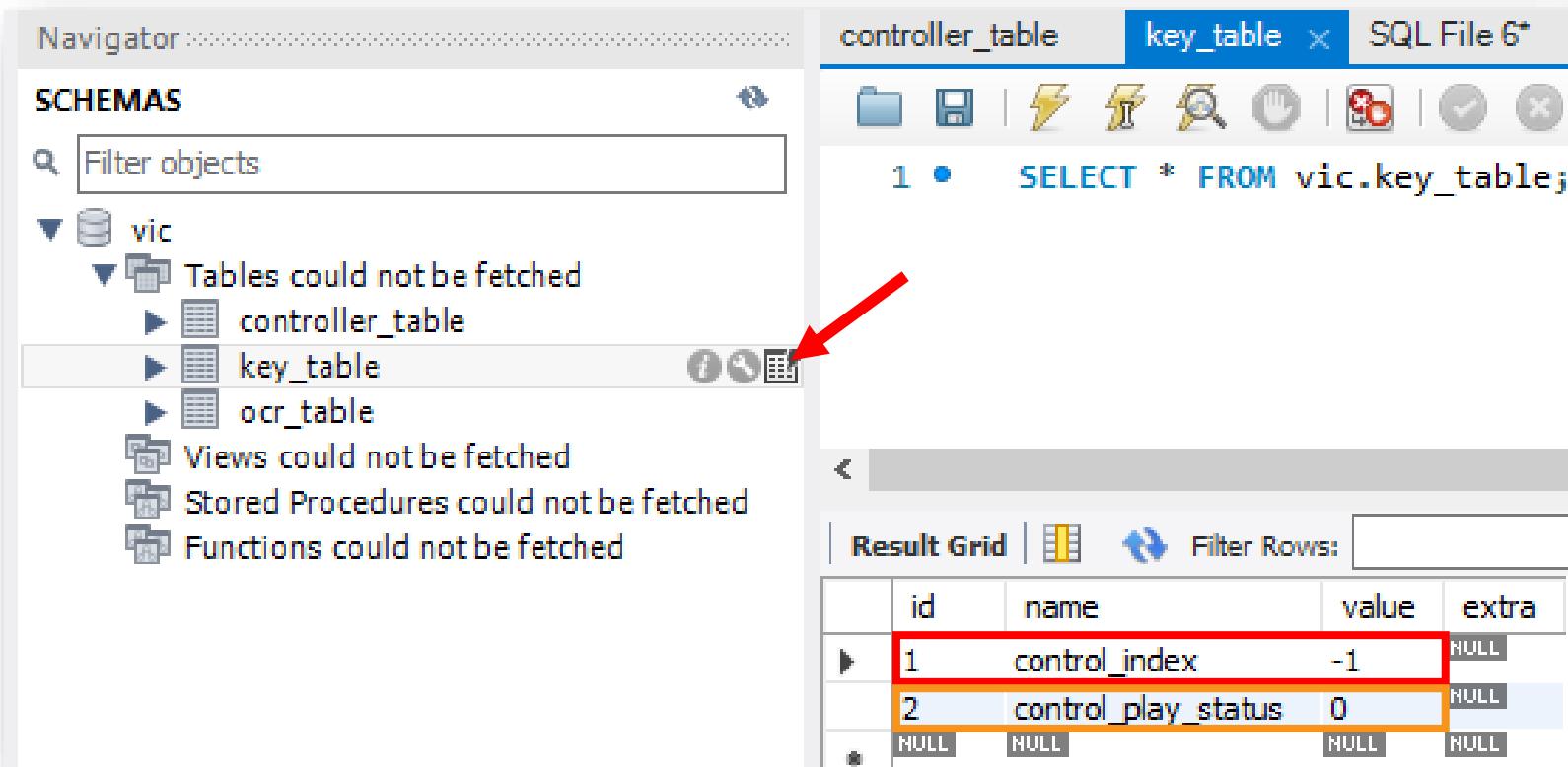
```
1
2   "login_success": "1",
3   "rsp_status": "1"
4
```

<b>login_success</b>	是否登入成功 1 = 成功 , 0 = 失敗
<b>rsp_status</b>	是否成功執行指令 1 = 成功 , 0 = 失敗

- 進階控制 – 輸入

4. 啟動控制檔 – SQL : 要啟動控制檔時須使用 vic 資料庫內的 key\_table , 使用 SQL 語法的 UPDATE 來啟動控制檔 , index 對應控制檔的編號
- 可使用 SELECT 來取得當前系統為控制運行狀態 , 0 = 停止 , 1 = 運作中

```
UPDATE vic.key_table SET value='index' WHERE id='1';    SELECT value FROM vic.key_table WHERE id='2';
```



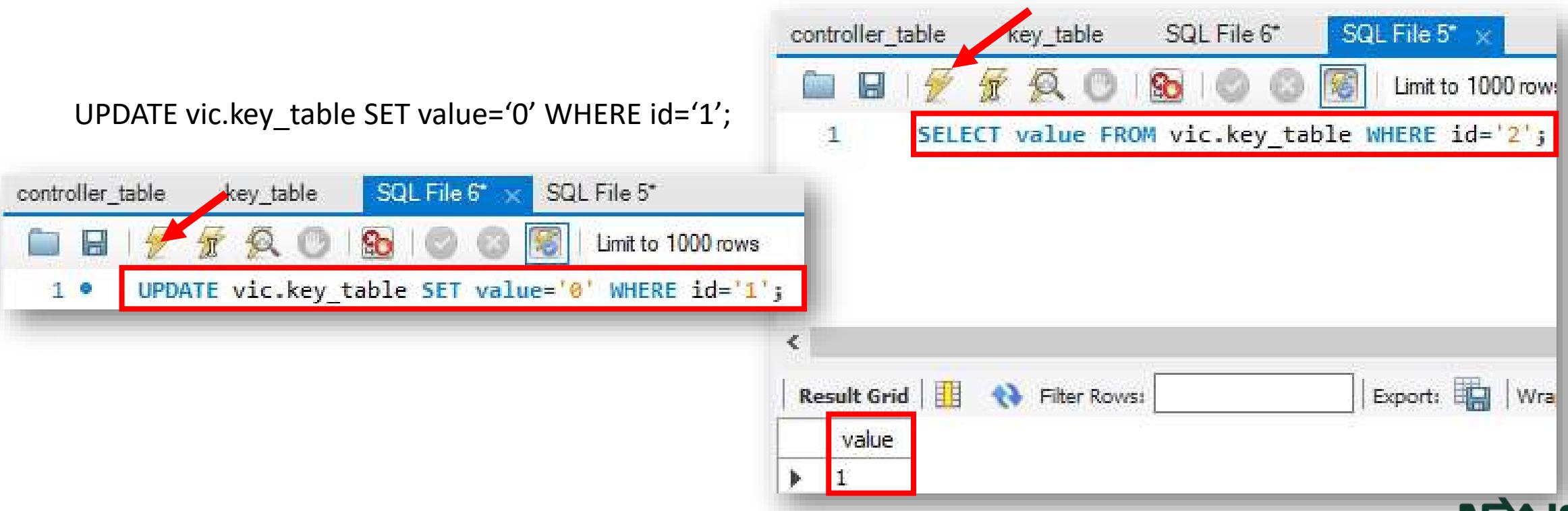
The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the SCHEMAS section, with 'vic' selected. Under 'Tables could not be fetched', there are three tables: 'controller\_table', 'key\_table', and 'ocr\_table'. A red arrow points to the 'key\_table' entry. On the right, the main workspace shows the 'key\_table' tab selected. A query window contains the SQL command: 'SELECT \* FROM vic.key\_table;'. Below it, a 'Result Grid' shows the following data:

	id	name	value	extra
▶	1	control_index	-1	NULL
▶	2	control_play_status	0	NULL
▶	NULL	NULL	NULL	NULL

- 進階控制 – 輸入

範例 – 啟動控制檔 – SQL : 使用 UPDATE 來啟動編號 00 的控制檔，並用 SELECT 取得系統的控制運行狀態

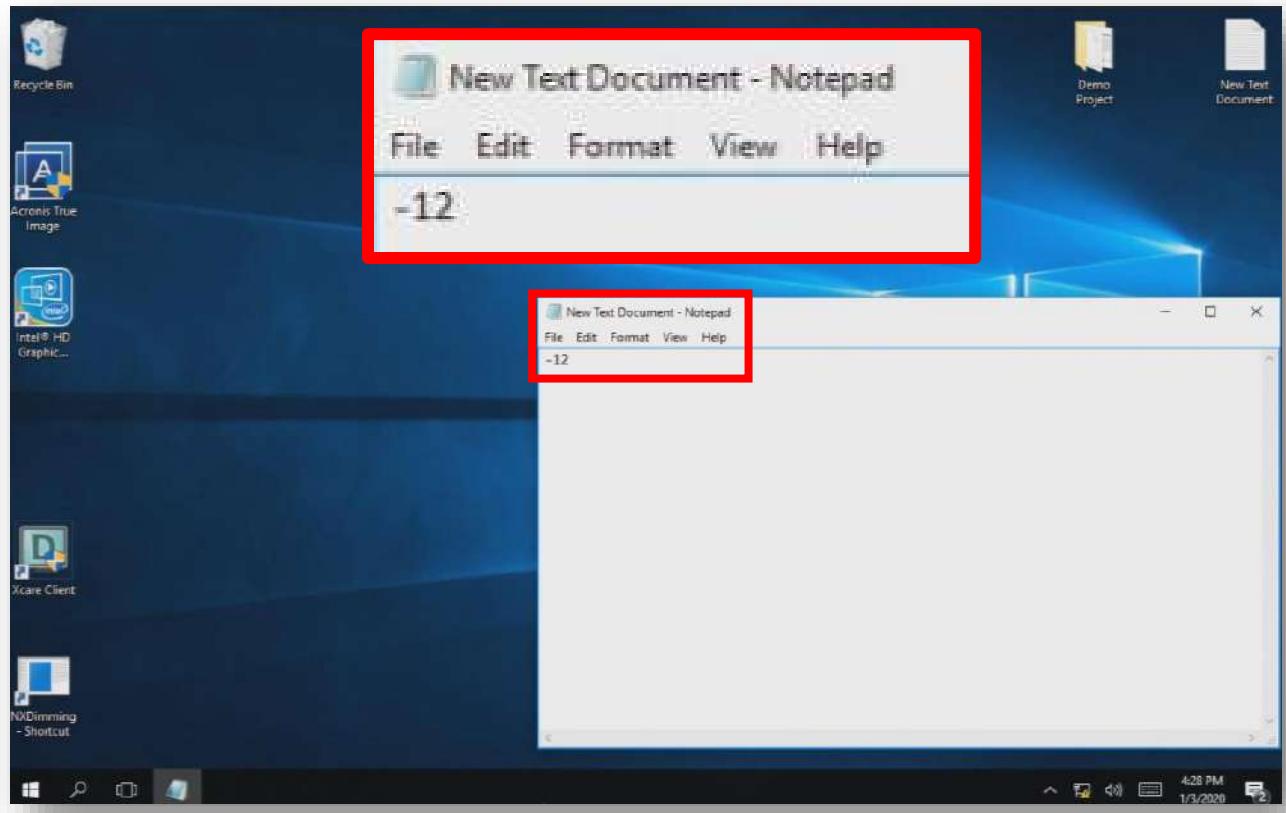
```
SELECT value FROM vic.key_table WHERE id='2';
```



## • 進階控制 – 輸入

5. 模擬確認：範例使用的控制檔為讓記事本變成可輸入的狀態，並在控制檔執行結束後輸入 -12 模擬時可確認兩點：

- ✓ 是否正確執行控制檔
- ✓ 輸入內容是否正確

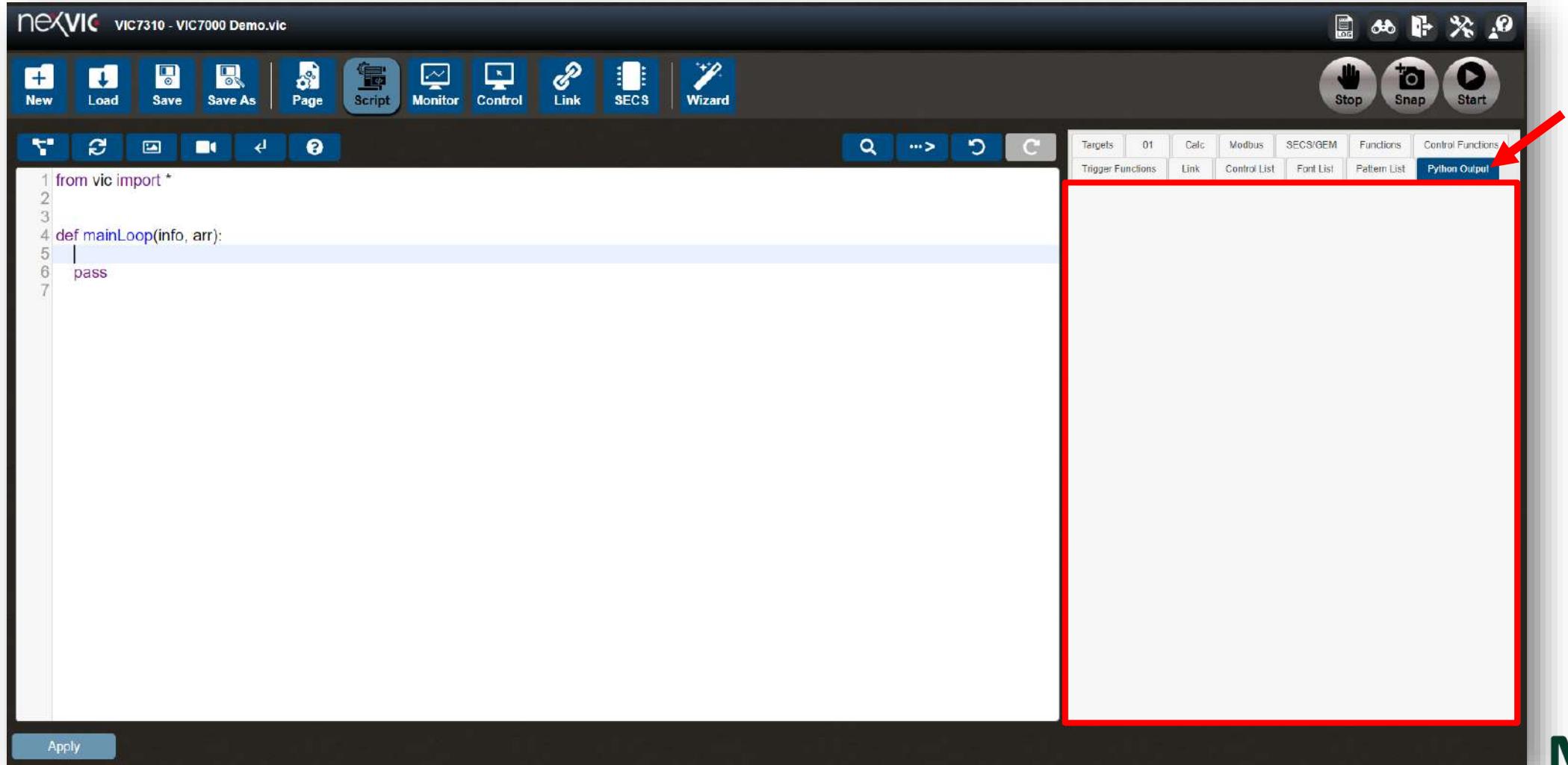


# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

- 開啟腳本

進入腳本頁面，在右邊的欄位選擇Python 輸出，可以看到 Python 腳本執行的輸出



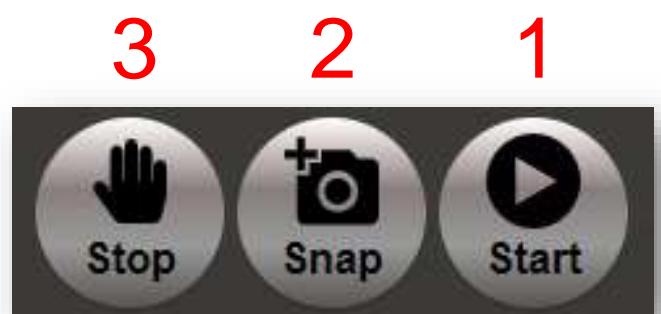
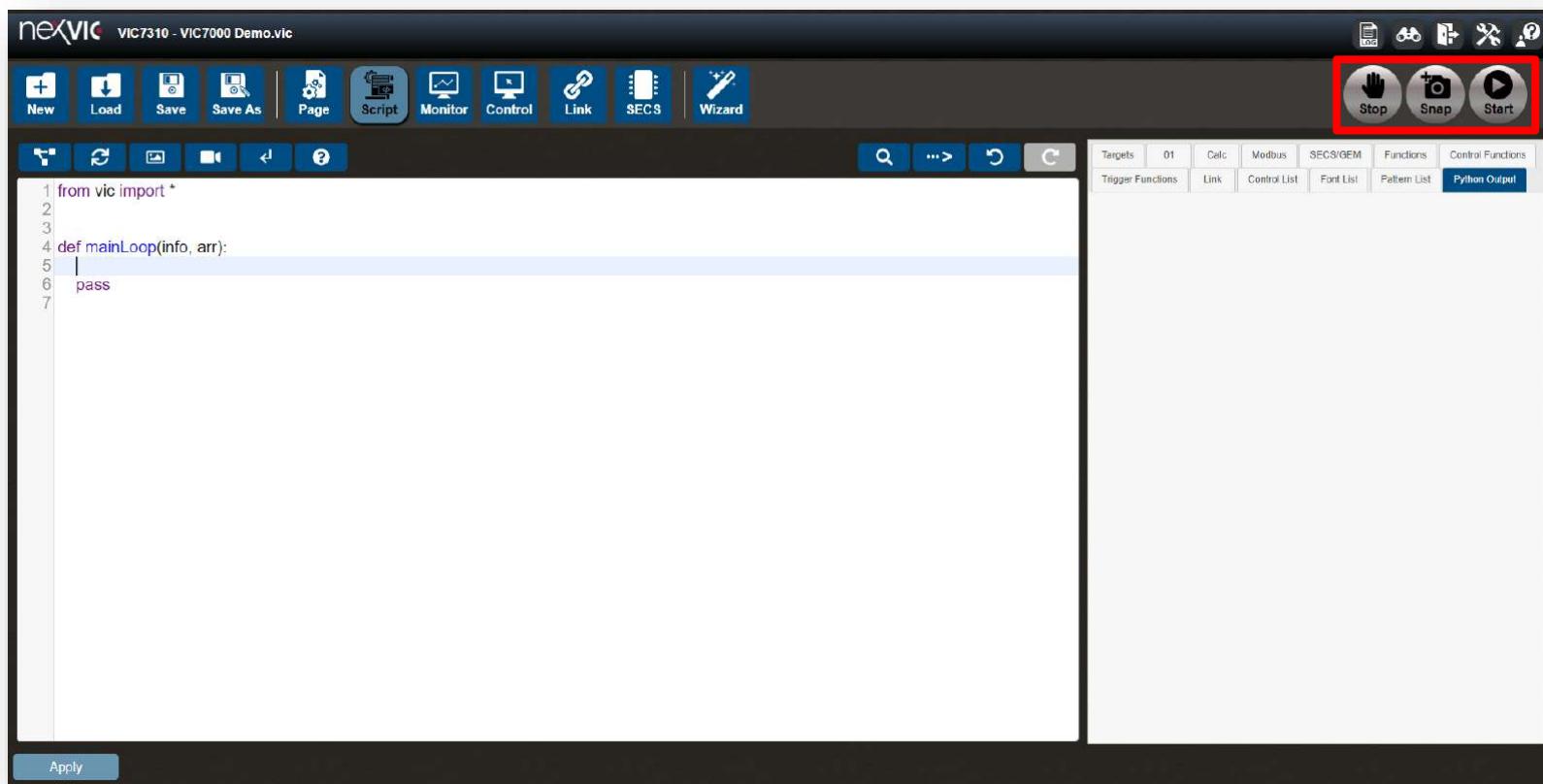
- **mainLoop**

每辨識一張圖，就會執行一次 mainLoop 函式

```
1 from vic import *
2
3
4 def mainLoop(info, arr):
5     pass
6
7
```

## • 功能按鍵

1. 啟動/停止：持續/停止從取像來源取得圖像進行辨識並執行腳本
2. 觸發：從取像來源擷取一張圖像進行辨識並執行腳本
3. 停止：停止所有腳本動作



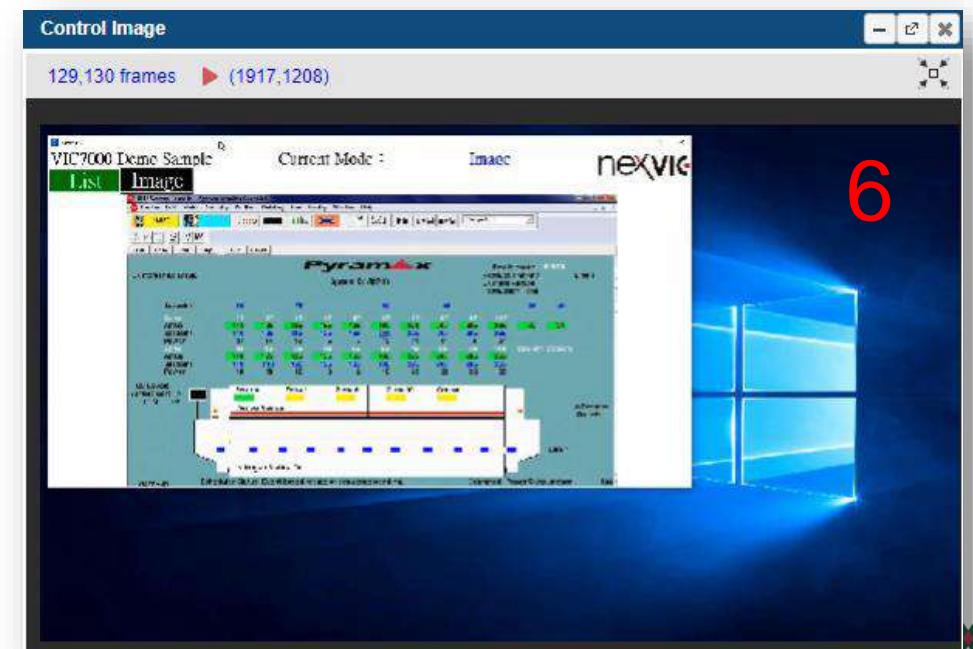
# 腳本

- 功能按鍵

3. 切換至流程圖模式
4. 重新載入Python插件模組
5. 腳本圖像檢視器
6. 控制圖像檢視器



3 4 5 6 7 8



# 腳本

- 功能按鍵

7. 顯示跳位字元
8. 鍵盤快捷鍵說明



```
1 from vic import *
2
3
4 def mainLoop(info, arr):
5     pass
6
7
8
```



3 4 5 6 7 8

Function Description	Keyboard Shortcuts
Snap	F1
Start	8 F2
Reload Python module	F9
Search	Ctrl + F
Go to line	Ctrl + G
Apply	Ctrl + S
Undo	Ctrl + Z
Redo	Ctrl + Y
Copy	Ctrl + C
Paste	Ctrl + V
Select all	Ctrl + A

OK

T

- 功能按鍵

9. 搜尋

10. 前往

11. 退回

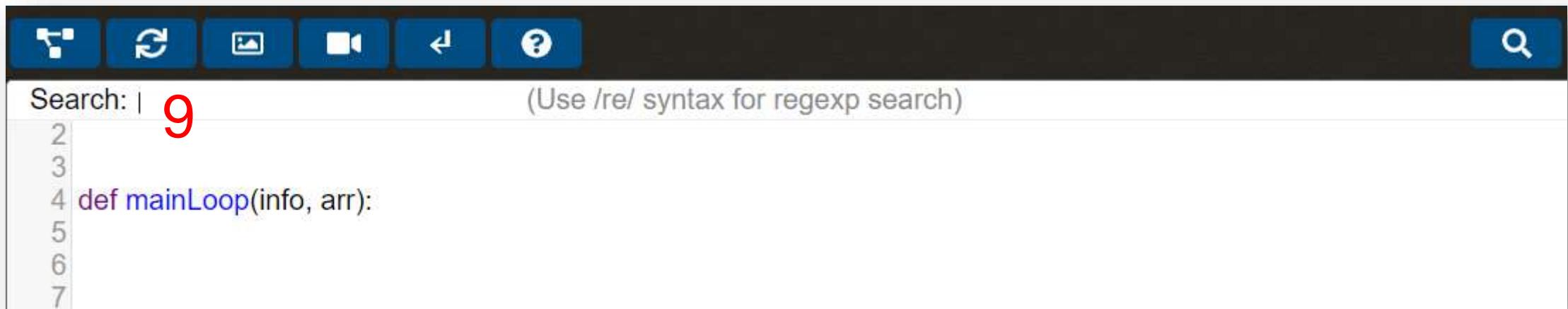
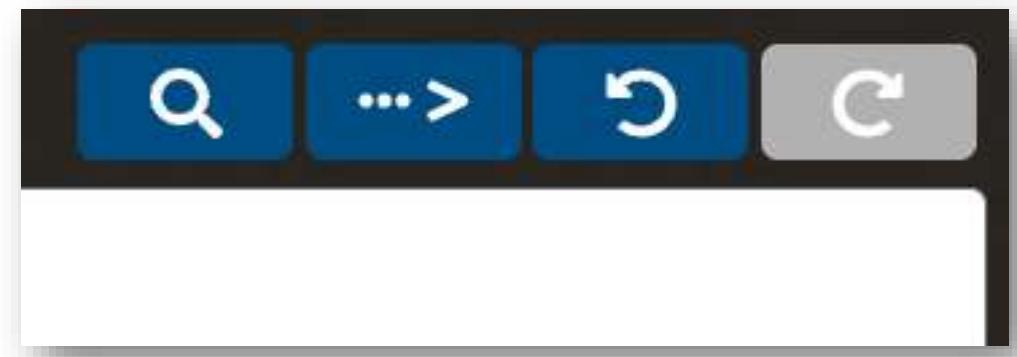
12. 重做

9

10

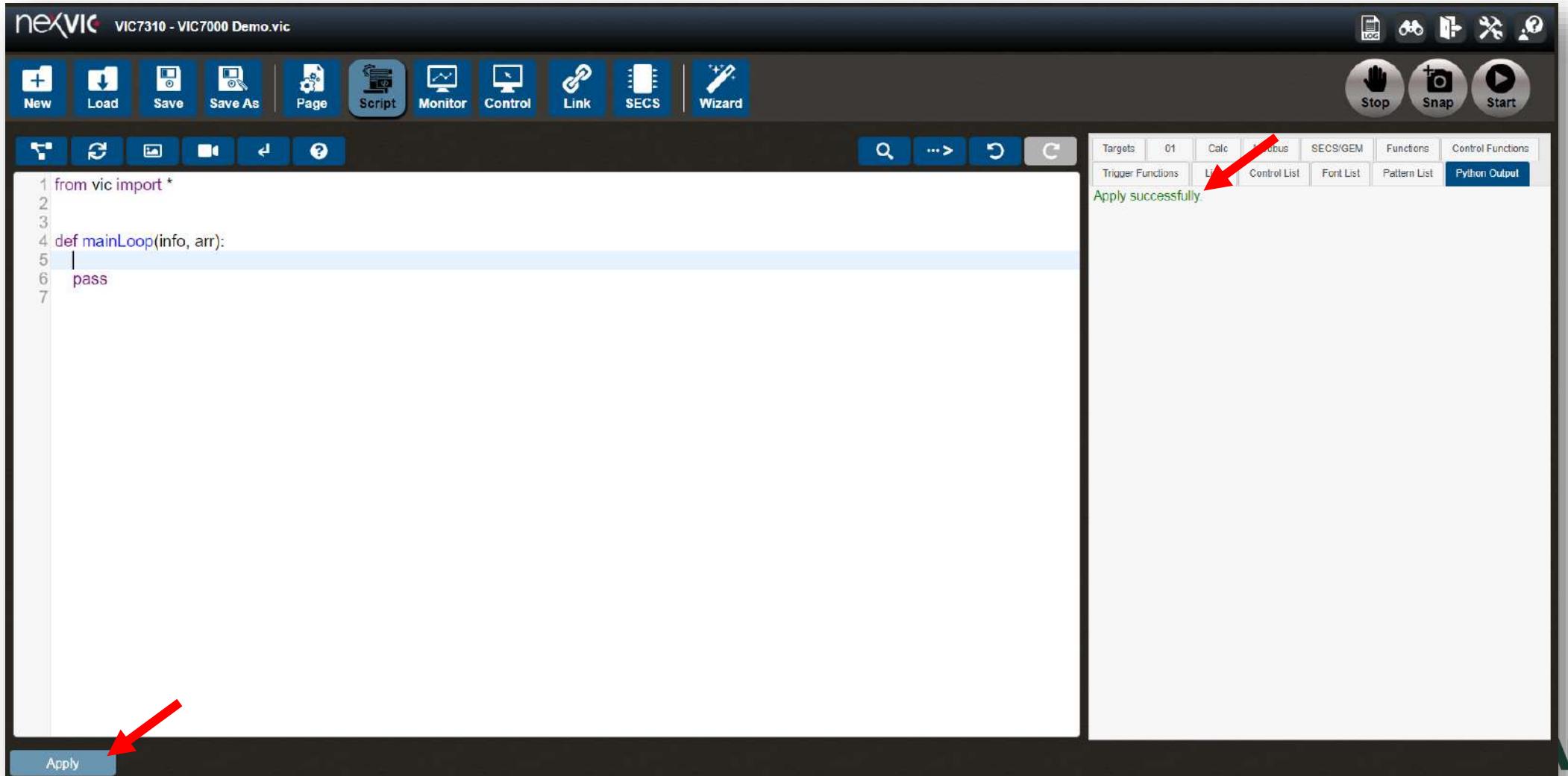
11

12



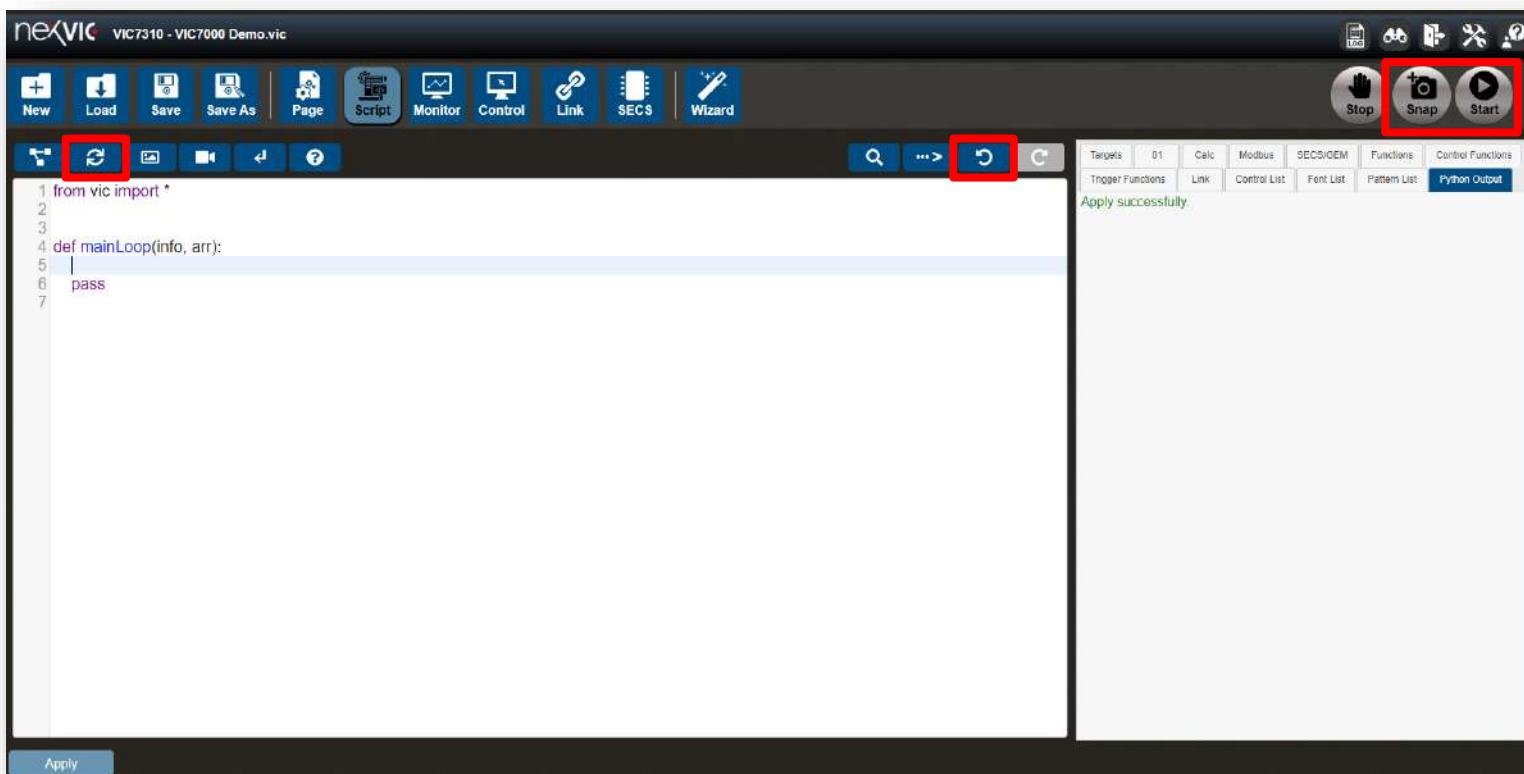
- 儲存變更

編輯完腳本後，要點擊左下方的儲存變更，腳本才會生效，快捷鍵為 **ctrl + s**



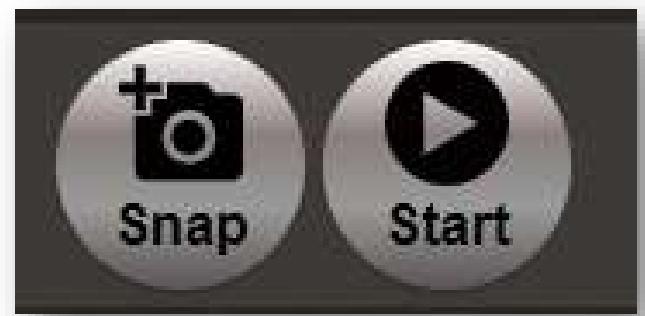
- 快捷按鍵

1. F1 = 觸發
2. F2 = 啟動停止顯示跳位字元
3. F9 = 重新載入Py模組
4. Ctrl + Z = 退回



F1

F2



F9



NEVIOT

- 快捷按鍵

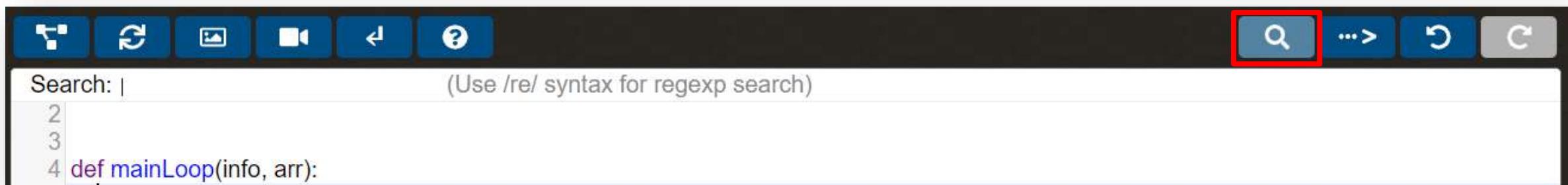
5. **Ctrl + F = 搜尋**

6. **F3 = 搜尋 (無搜尋目標時)**

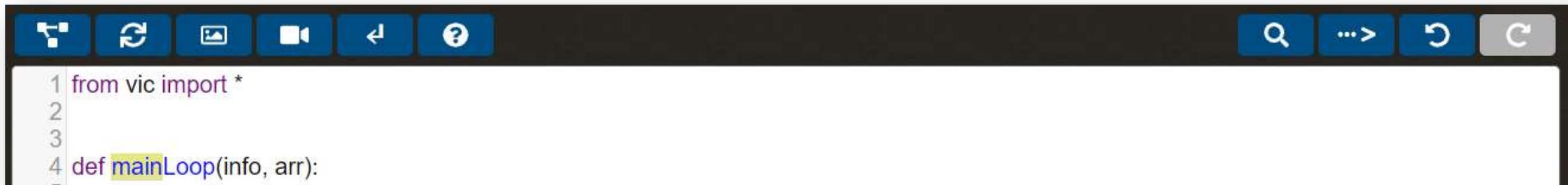
7. **Ctrl + Shift + F = 取代**

輸入搜尋目標後，按下 **Enter** 開始搜尋

有搜尋目標時，**F3 = 下一筆**，**Shift + F3 = 上一筆**



```
2
3
4 def mainLoop(info, arr):
```



```
1 from vic import *
2
3
4 def mainLoop(info, arr):
```

## • 引用辨識工具

在右邊的欄位中，選擇目標值和其他頁面編號，可以引用每一頁設定的辨識工具

這裡列出的辨識工具和專案內設定的辨識工具相同



- 辨識工具

- TARGET、OCR

此變數為當前圖片的文字辨識結果

變數形式為 TA.TARGET`xx` 和 PAGE`yy`.OCR`xx`

`xx` 為辨識工具編號

`yy` 為頁面編號

TA.TARGET01

PAGE01.OCR01

- 辨識工具

- TARGET\_COLOR、COLOR

此變數為當前圖片的顏色比對結果

變數形式為 TA.TARGET\_COLORxx

和 PAGEyy.COLORxx

xx 為辨識工具編號，yy 為頁面編號

還可細分為

VALUE：顏色比對結果輸出

MATCH：是否比對成功

RED：當前圖片上顏色比對區域的 RGB 色域紅色值

GREEN：當前圖片上顏色比對區域的RGB 色域綠色值

BLUE：當前圖片上顏色比對區域的RGB 色域藍色值

TA.TARGET\_COLOR01  
TA.TARGET\_COLOR01.VALUE  
TA.TARGET\_COLOR01.MATCH  
TA.TARGET\_COLOR01.RED  
TA.TARGET\_COLOR01.GREEN  
TA.TARGET\_COLOR01.BLUE  
PAGE01.COLOR01  
PAGE01.COLOR01.VALUE  
PAGE01.COLOR01.MATCH  
PAGE01.COLOR01.RED  
PAGE01.COLOR01.GREEN  
PAGE01.COLOR01.BLUE

- 辨識工具

- TARGET\_PATTERN、PATTERN

此變數為當前圖片的顏色比對結果

變數形式為 TA.TARGET\_PATTERN<sub>XX</sub>

和 PAGE<sub>yy</sub>.PATTERN<sub>XX</sub>

<sub>XX</sub> 為辨識工具編號，<sub>yy</sub> 為頁面編號，還可細分為

VALUE：樣式比對結果輸出

FOUND：是否有找到樣式

SCORE：樣式比對成功時的比對分數

X：當前圖片上找到樣式的 X 座標

Y：當前圖片上找到樣式的 Y 座標

WIDTH：當前圖片上找到樣式的寬度

HEIGHT：當前圖片上找到樣式的高度

BASE\_X：設定比對樣式的圖片上的樣式 X 座標

BASE\_Y：設定比對樣式的圖片上的樣式 Y 座標

TA.TARGET\_PATTERN01

TA.TARGET\_PATTERN01.VALUE

TA.TARGET\_PATTERN01.FOUND

TA.TARGET\_PATTERN01.SCORE

TA.TARGET\_PATTERN01.X

TA.TARGET\_PATTERN01.Y

TA.TARGET\_PATTERN01.WIDTH

TA.TARGET\_PATTERN01.HEIGHT

TA.TARGET\_PATTERN01.BASE\_X

TA.TARGET\_PATTERN01.BASE\_Y

PAGE01.PATTERN01

PAGE01.PATTERN01.VALUE

PAGE01.PATTERN01.FOUND

PAGE01.PATTERN01.SCORE

PAGE01.PATTERN01.X

PAGE01.PATTERN01.Y

PAGE01.PATTERN01.WIDTH

PAGE01.PATTERN01.HEIGHT

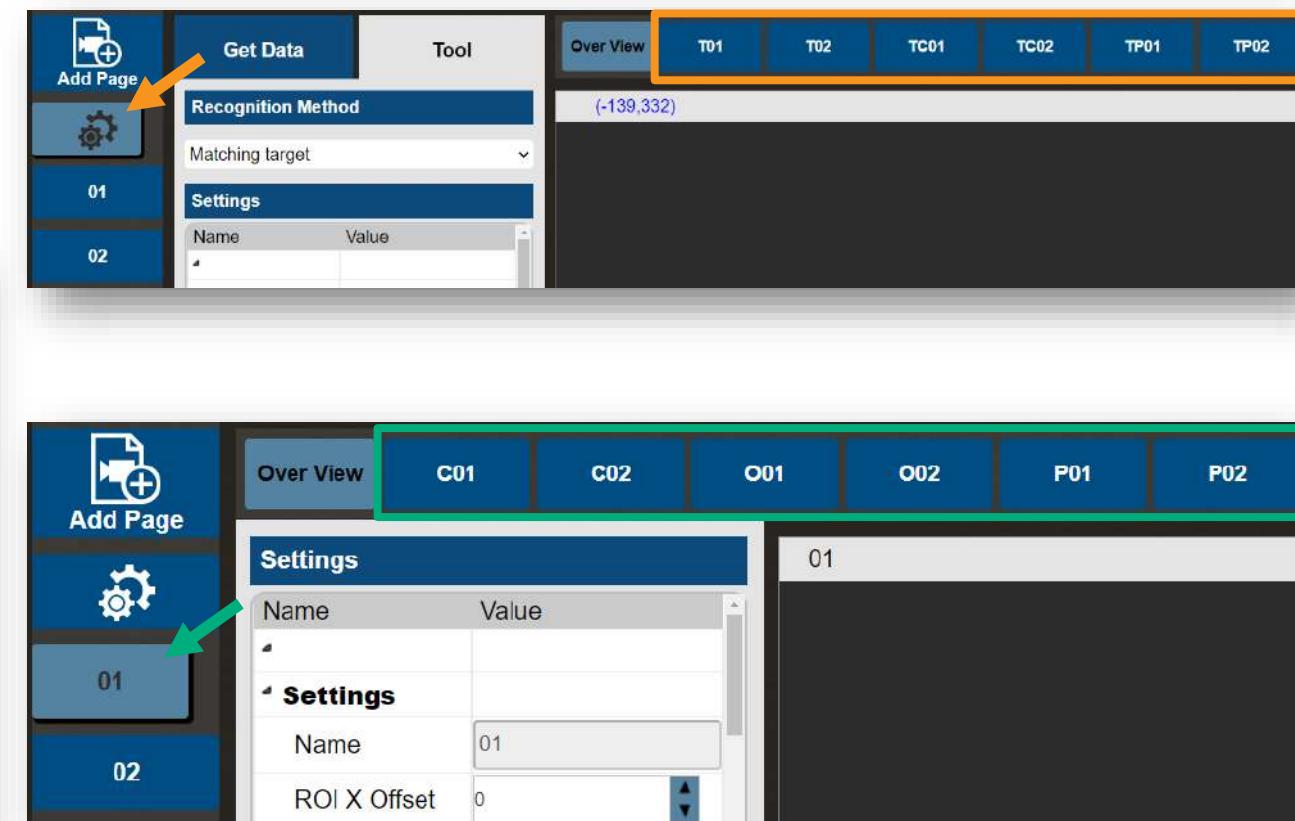
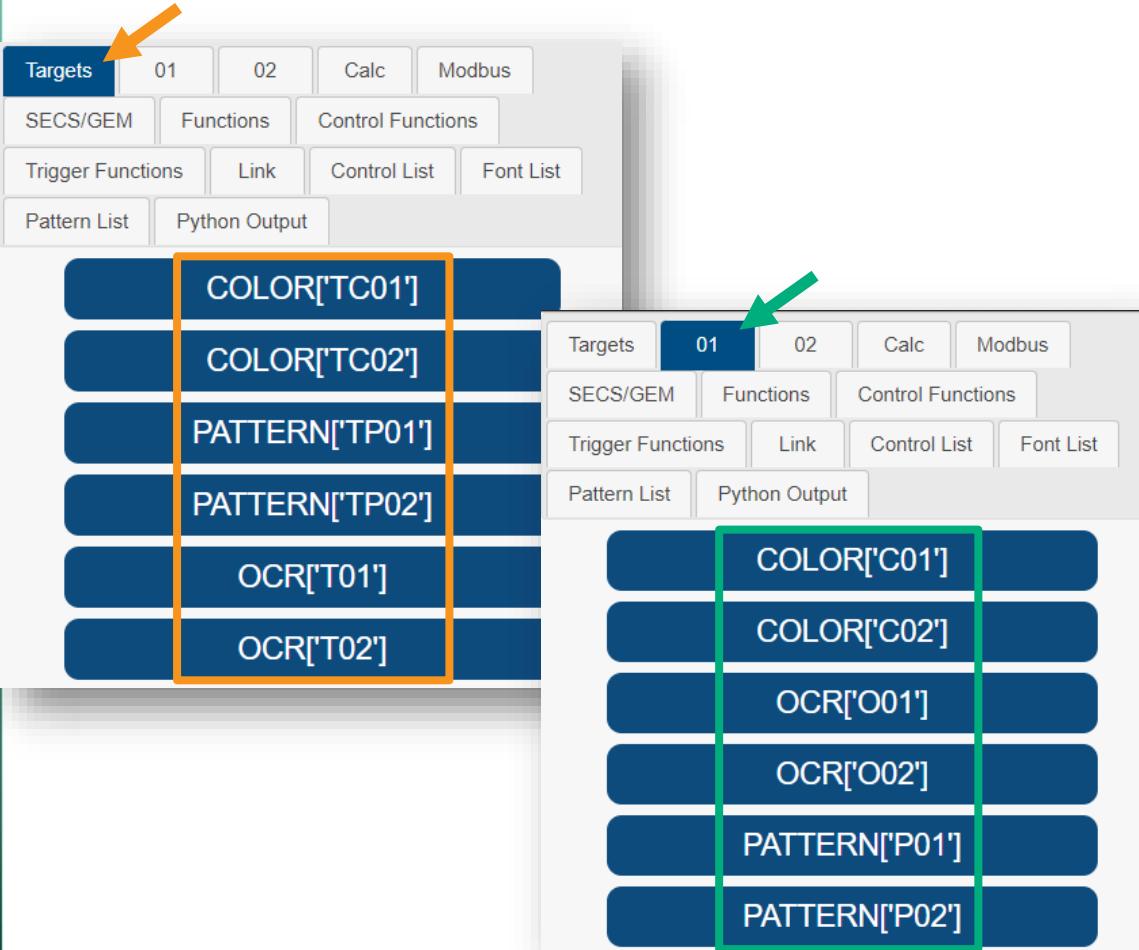
PAGE01.PATTERN01.BASE\_X

PAGE01.PATTERN01.BASE\_Y

## • 引用辨識工具

可以透過重新命名來設定辨識工具

這裡列出的辨識工具名稱和專案內重新命名的辨識工具相同



- 引用辨識工具

變數形式為

TA.OCR[AAAA]

TA.COLOR[AAAA]

TA.PATTERN[AAAA]

PAGEyy.OCR[AAAA]

PAGEyy.COLOR[AAAA]

PAGEyy.PATTERN[AAAA]

其中

yy 為頁面編號

AAAA 為重新命名後的辨識工具名稱

TA.OCR['T01']

TA.COLOR['TC01']

TA.PATTERN['TP01']

PAGE01.OCR['O01']

PAGE01.COLOR['C01']

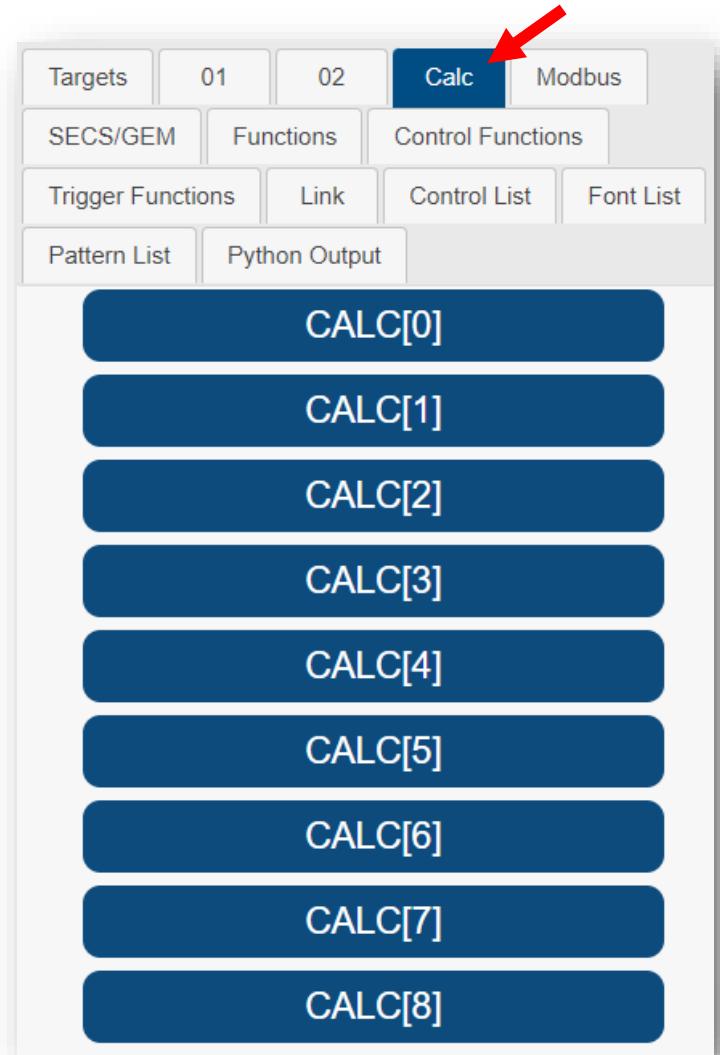
PAGE01.PATTERN['P01']

- **運算**

- **CALC**：為變數，可用於儲存腳本執行結果，運算值會被儲存進資料庫，且可透過外部連線傳輸

使用語法為CALC[x]，x 為 0 ~ 99

CALC[0]  
CALC[1]  
CALC[98]  
CALC[99]



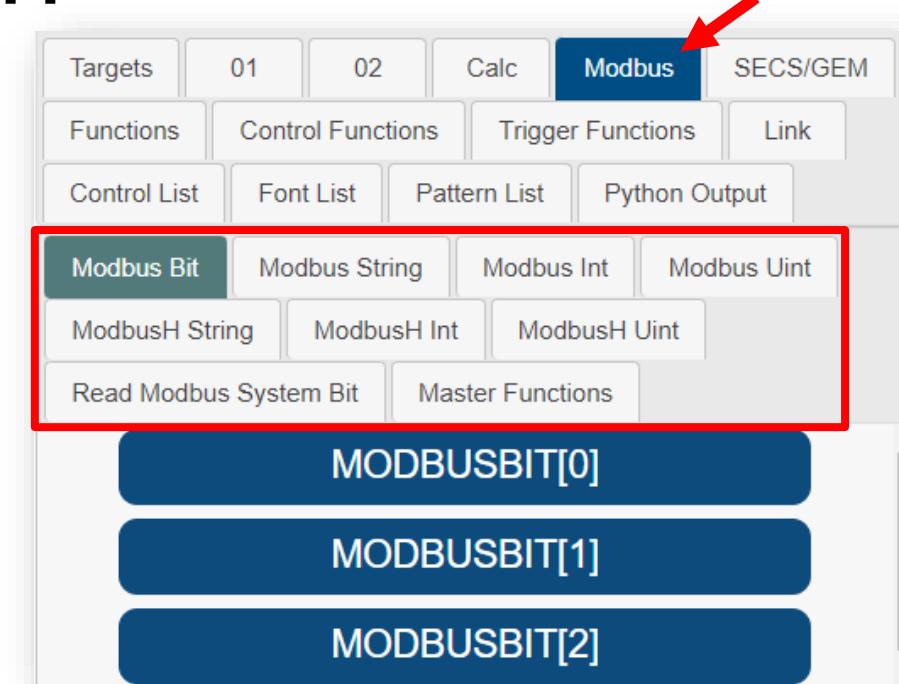
- **Modbus 資料欄位**

- MODBUSBIT、MODBUSSTR、MODBUSINT、MODBUSUINT、MODBUSHSTR、MODBUSHINT、MODBUSHUINT：為變數，可自定義要存放的位元、字串、有號整數、無號整數在 MODBUS 欄位中

使用語法為 MODBUSBIT[x]、MODBUSSTR[x]、MODBUSINT[x]、MODBUSUINT[x]、MODBUSHSTR[y]、MODBUSHINT[z]、MODBUSHUINT[z]

其中 x 為 0 ~ 99，y 為 0 ~ 199，z 為 0 ~ 499

MODBUSBIT[0]  
MODBUSSTR[0]  
MODBUSINT[0]  
MODBUSUINT[0]  
MODBUSHSTR[0]  
MODBUSHINT[0]  
MODBUSHUINT[0]



- **Modbus 系統資訊欄位**

- **MODBUSSYSBIT**：存放代表系統資訊的欄位，使用語法為 MODBUSSYSBIT[x]

$x = 0$ ：程式是否運行中，1 = 運行中，0 = 停止

$x = 1$ ：是否有使用者登入，1 = 有人登入，0 = 無人登入

$x = 2$ ：系統硬碟容量 5G 檢測，1 = 低於 5G，0 = 高於 5G

$x = 3$ ：系統硬碟容量 10G 檢測，1 = 低於 10G，0 = 高於 10G

$x = 4$ ：是否停止存圖，1 = 停止，0 = 運行中

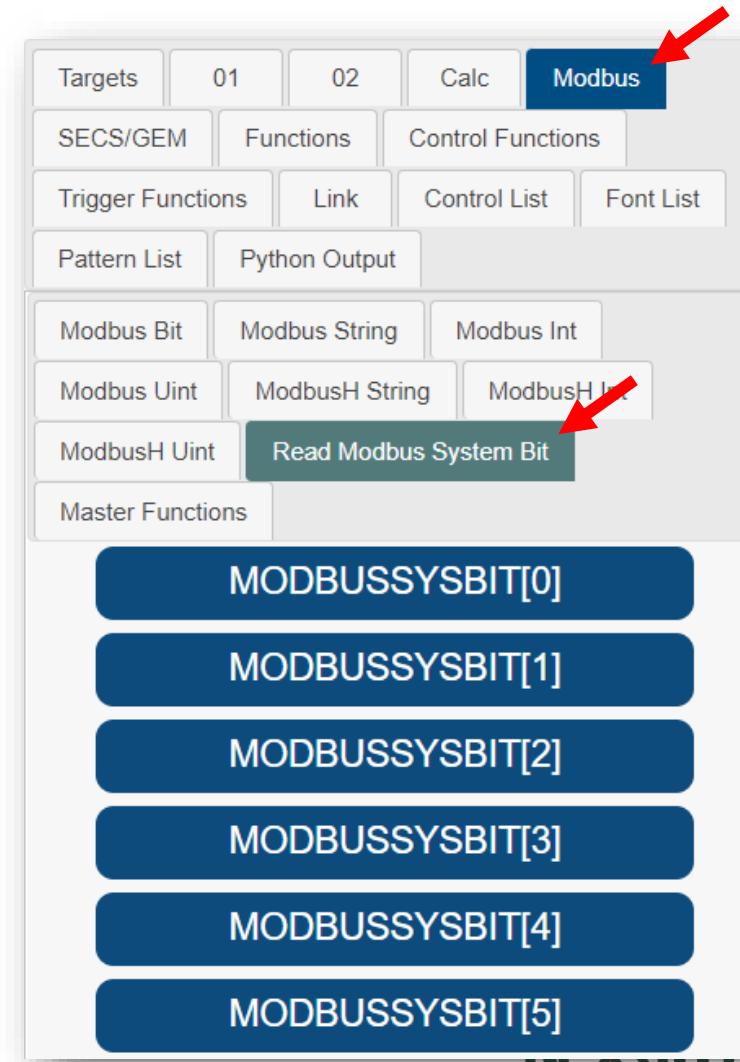
$x = 7$ ：控制是否運行中，1 = 運行中，0 = 停止

$x = 8$ ：序列埠控制是否被啟用，1 = 取用，0 = 停用

$x = 9$ ：序列埠目前狀態，1 = 連線，0 = 斷線

$x = 16$ ：資料硬碟容量 10G 檢測，1 = 低於 5G，0 = 高於 10G

$x = 17$ ：資料硬碟容量 20G 檢測，1 = 低於 10G，0 = 高於 20G



- **Modbus 系統資訊欄位**

- **MODBUSSYSBIT**：存放代表系統資訊的欄位，使用語法為 MODBUSSYSBIT[x]

$x = 5, 6$ ：OCR辨識結果，0 = 辨識成功，1 = NG，2 = 比對失敗

$x = 19$ ：是否有使用者登入，1 = 有人登入，0 = 無人登入

$x = 20$ ：額外儲存路徑容量 1G 檢測，1 = 低於 1G，0 = 高於 1G

<b><math>x = 6</math></b>	<b><math>x = 5</math></b>	<b>Sum</b>	<b>Result</b>
0	0	0	辨識成功
0	1	1	辨識 NG
1	0	2	比對失敗

## • 函式

在右邊的欄位選擇函式，這裡會列出 VIC 的功能函式及變數，可直接點擊要引用的函式或變數



PYTHON\_MAIN\_LOOP：在腳本區域輸入腳本基本內容

LOG：記錄訊息在系統日誌中

SEND.EMAIL：透過 SMTP 發送電子郵件

SEND.LINE：發送 LINE 通知

SEND.WECHAT：發送 WeChat 通知

SEND.WECHA\_P：發送 WeChat 通知

SEND.TEAMS：發送 Teams 通知

RT.PAGE\_NO：當前辨識的頁面編號

RT.PAGE\_NAME：當前辨識的頁面名稱

RT.RESULT：當前辨識結果

RT.CURRENT\_TIME：執行函式的當前時間

## • 函式

在右邊的欄位選擇函式，這裡會列出 VIC 的功能函式及變數，可直接點擊要引用的函式或變數

**ASSIGN\_PAGE** : 指定要使用的辨識頁面

**TOOL.OCR** : 在當前辨識的圖像上進行文字辨識

**TOOL.COLOR** : 在當前辨識的圖像上進行顏色比對

**TOOL.PATTERN** : 在當前辨識的圖像上進行樣式比對

**PAGE\_PROCESS** : 使用辨識頁面設定來進行辨識

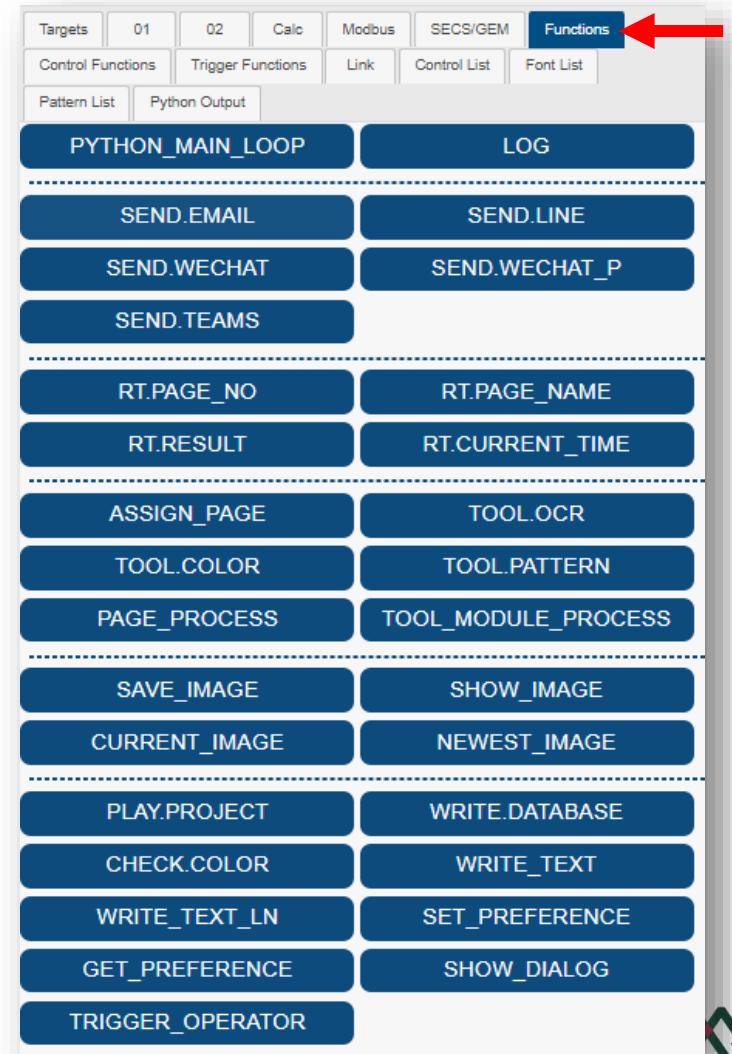
**TOOL\_MODULE\_PROCESS** : 使用工具插件來進行處理或辨識

**SAVE\_IMAGE** : 儲存當前辨識的圖片到本地

**SHOW\_IMAGE** : 在腳本圖像中顯示當前辨識的圖片

**CURRENT\_IMAGE** : 當前辨識的圖片

**NEWEST\_IMAGE**: 可取得最新擷取圖像



## • 函式

在右邊的欄位選擇函式，這裡會列出 VIC 的功能函式及變數，可直接點擊要引用的函式或變數

**PLAY.PROJECT**：設定專案啟動/停止

**WRITE.DATABASE**：設定辨識資料是否寫入資料庫

**CHECK.COLOR**：偵測顏色是否閃爍

**WRITE\_TEXT**：寫入文字到本地文字檔

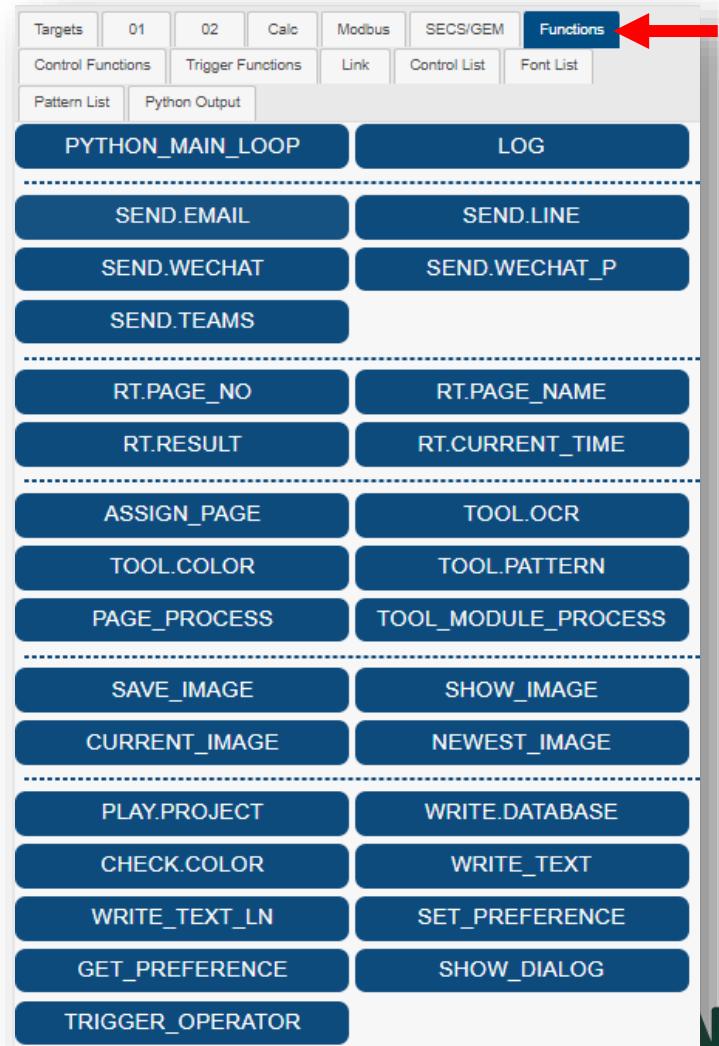
**WRITE\_TEXT\_LN**：寫入文字到本地文字檔，寫入後換行

**SET\_PREFERENCE**：設定系統變數的值

**GET\_PREFERENCE**：取得系統變數的值

**SHOW\_DIALOG**：跳出提示視窗

**TRIGGER\_OPERATOR**：觸發VIC flow內的觸發運算子



## • 控制函式

在右邊的欄位選擇控制函式，這裡會列出 VIC 控制函式，可直接點擊要引用的控制函式

**PLAY.CONTROL**：設定控制啟動/停止

**CLICK**：控制滑鼠游標移動到特定位置，並點擊一次

**DB\_CLICK**：控制滑鼠游標移動到特定位置，並連點兩次

**DRAG**：控制滑鼠游標從一點移動另一點，同時進行拖曳

**OCR\_CLICK**：控制滑鼠在符合比對條件的文字的位置點擊一次

**OCR\_DB\_CLICK**：控制滑鼠在符合比對條件的文字的位置連點兩次

**PATTERN\_CLICK**：控制滑鼠在符合比對條件的樣式的位置點擊一次

**PATTERN\_DB\_CLICK**：控制滑鼠在符合比對條件的樣式的位置連點兩次

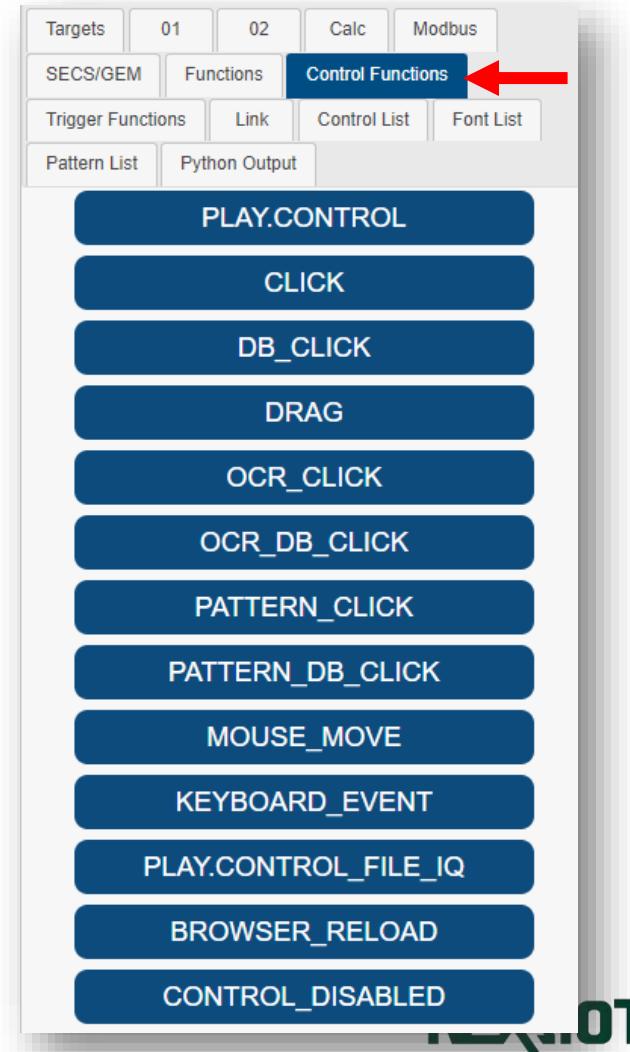
**MOUSE\_MOVE**：移動滑鼠游標到特定位置

**KEYBOARD\_EVENT**：控制鍵盤按下設定的字串內容

**PLAY.CONTROL\_FILE\_IQ**：套用佇列和輸入屬性，執行特定控制檔

**BROWSER\_RELOAD**：取像來源為來自瀏覽器時，重新載入瀏覽器

**CONTROL\_DISABLED**：停用控制功能



- **Modbus函式**

在右邊的欄位選擇Modbus，再選擇函式，這裡會列出 VIC Modbus函式，可直接引用Modbus函式

MODBUS\_MASTER\_READ\_BIT：從 Modbus slave/server 讀取位元資料

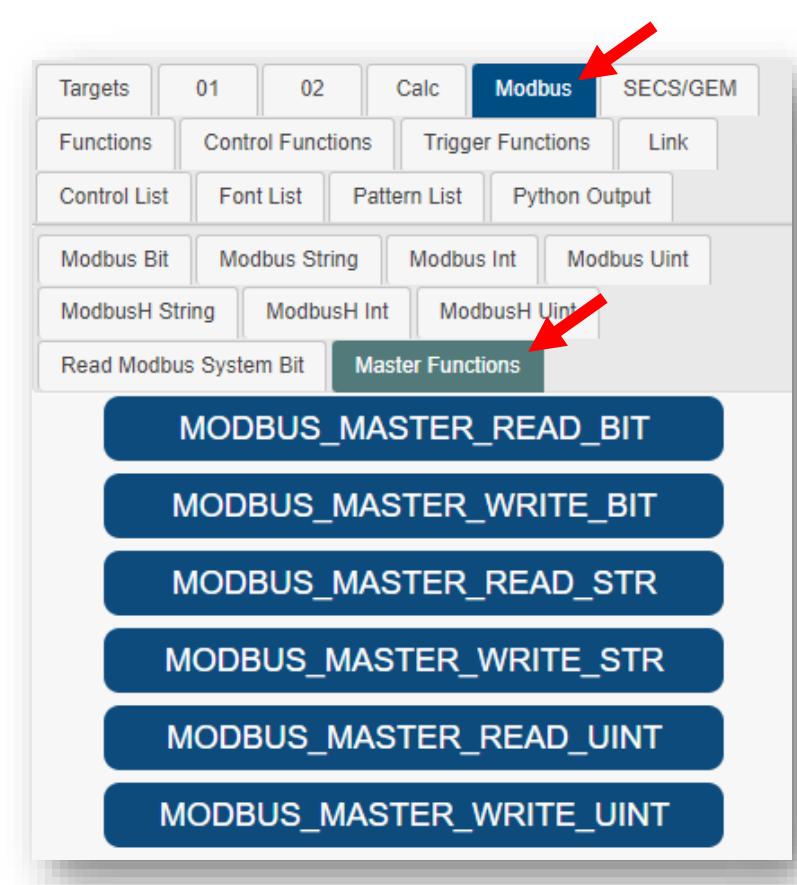
MODBUS\_MASTER\_WRITE\_BIT：寫入位元資料到 Modbus slave/server

MODBUS\_MASTER\_READ\_STR：從 Modbus slave/server 讀取字串資料

MODBUS\_MASTER\_WRITE\_STR：寫入字串資料到 Modbus slave/server

MODBUS\_MASTER\_READ\_UINT：從 Modbus slave/server 讀取整數資料

MODBUS\_MASTER\_WRITE\_UINT：寫入整數資料到 Modbus slave/server



- 觸發函式

在右邊的欄位選擇觸發函示，這裡會列出 VIC 觸發函式，可直接引用要使用的觸發函式

OnInitScript：腳本初始化時會觸發的函式

OnReloadFunction：重新載入腳本時會觸發的函式

OnTriggerScriptFunction：透過Modbus TCP/RESTful API觸發的函式

OnAcqStart：開始擷取影像時觸發的函式

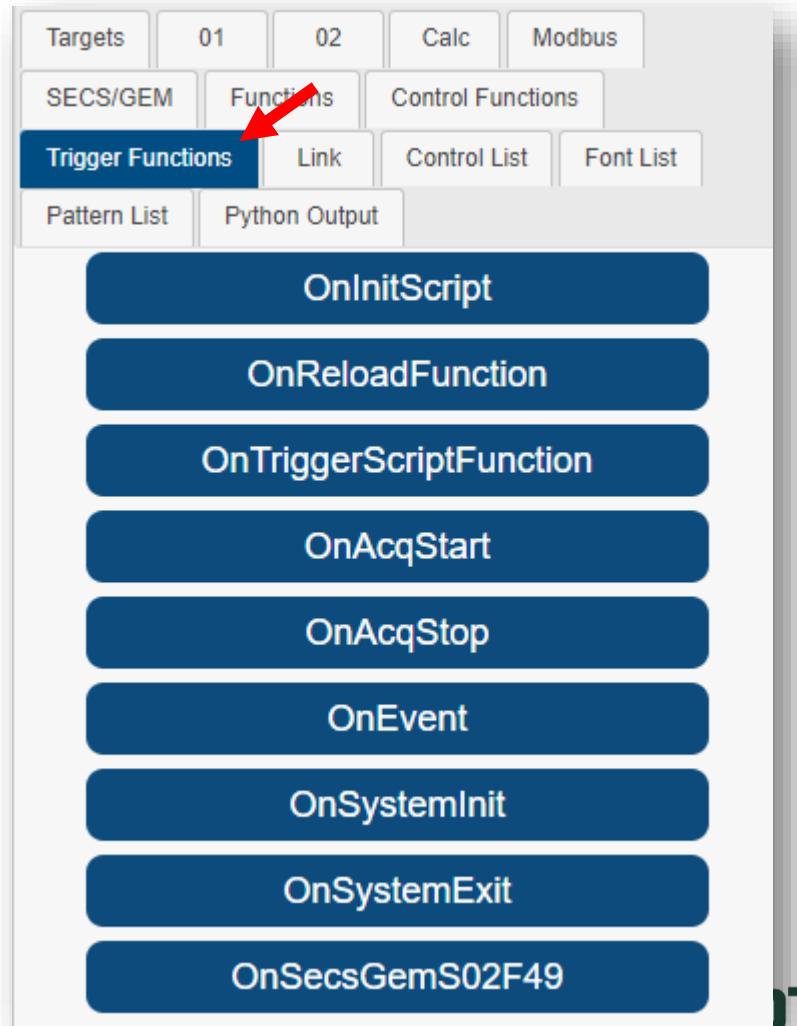
OnAcqStop：停止擷取影像時觸發的函式

OnEvent：擷取卡訊號改變時觸發的函式

OnSystemInit：系統啟動且初始化完成時觸發的函式

OnSystemExit：系統被關閉時執行的函式

OnSecsGemS02F49：系統透過SECS/GEM收到S02F49時觸發的函式

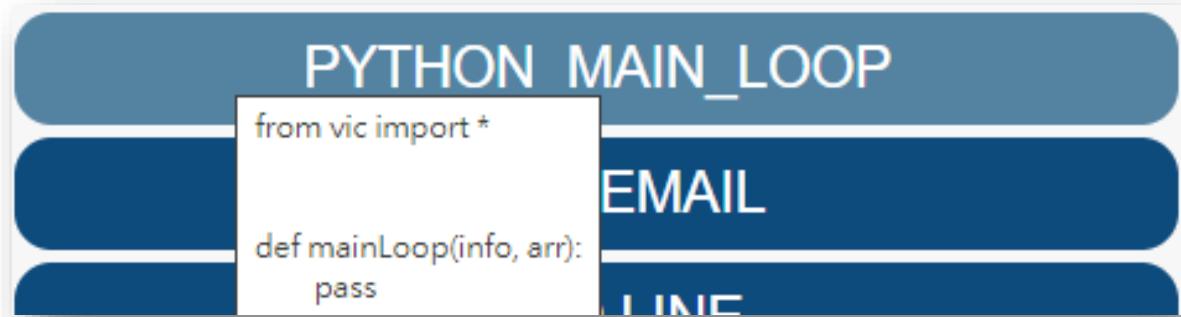


- 函式

- **PYTHON\_MAIN\_LOOP**：在腳本區域輸入 Python 腳本基本內容

輸入內容包括匯入 vic 函式庫和定義 mainLoop 函式

每執行一次辨識，系統就會執行一次 mainLoop 函式



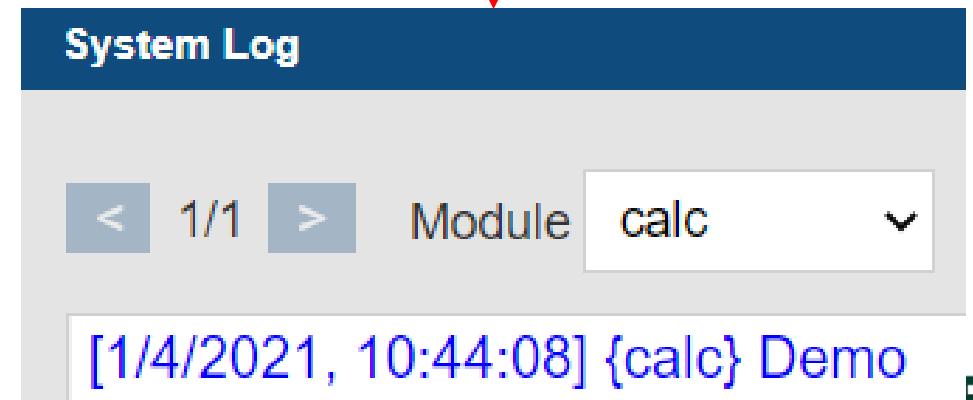
```
from vic import *
def mainLoop(info, arr):
    pass
```

- 函式

- **LOG**：記錄訊息在系統日誌中
  - 1. type：記錄在系統日誌中的訊息文字顏色，0 = 黑色，1 = 紅色，2 = 藍色
  - 2. message：要記錄的訊息內容
  - 3. sec：執行間隔時間



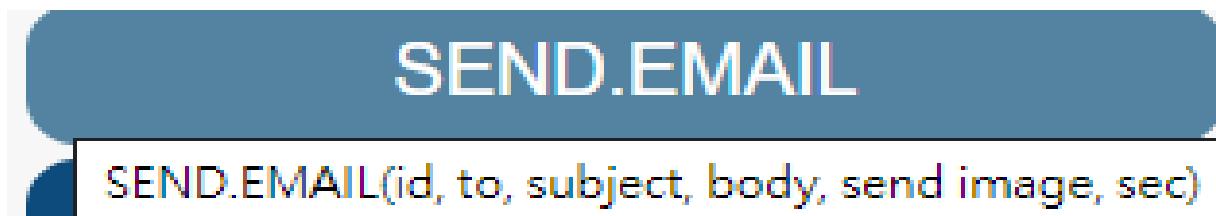
**LOG(2 , 'Demo', 10 )**



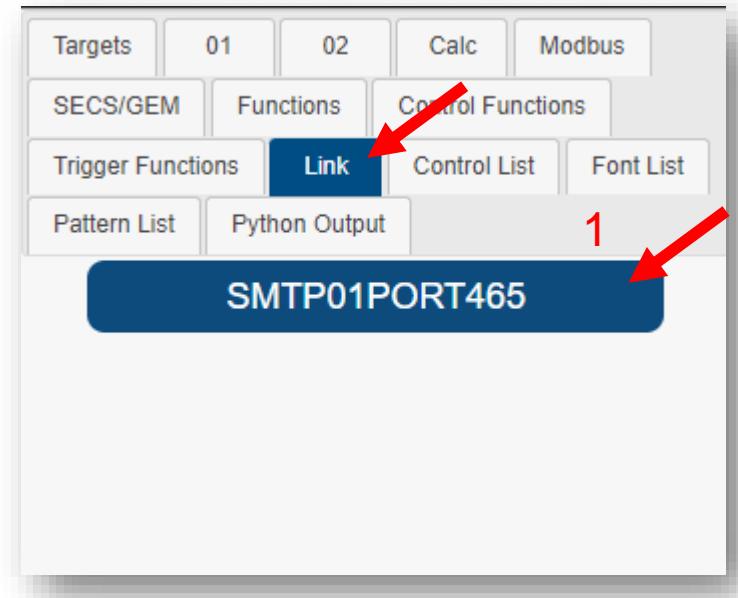
- 函式

- **SEND.EMAIL** : 透過 SMTP 發送電子郵件

1. id : 要使用的 SMTP 連線，要先在連結頁面內建立連線可從連結中選擇要使用的連線
2. to : 收件人信箱
3. subject : 信件主旨
4. body : 信件內容
5. send image : 是否傳送當下畫面(1 : 傳送，0 : 不傳送)
6. sec : 執行間隔時間



**SEND.EMAIL( , , , , , )**



- 函式

- **SEND.LINE** : 發送 LINE 通知

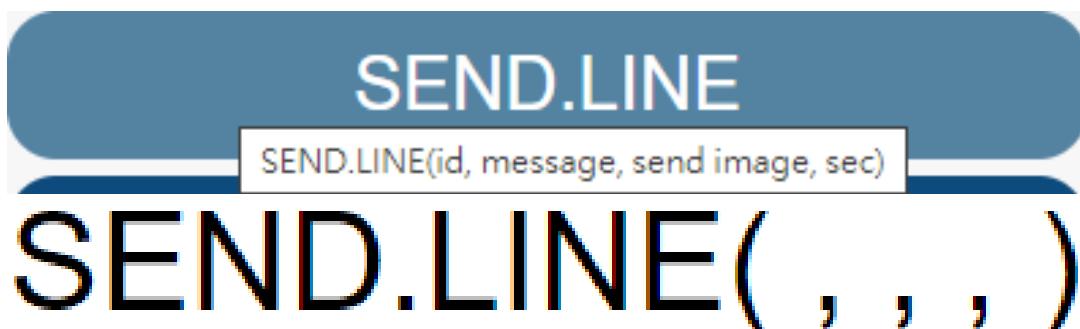
- 1. id : 要使用的 LINE Notify 連線，要先在連結頁面內建立連線

可從連結中選擇要使用的連線

- 2. message : 要傳送的訊息

- 3. send image : 是否傳送當下畫面(1 : 傳送，0 : 不傳送)

- 4. sec : 執行間隔時間



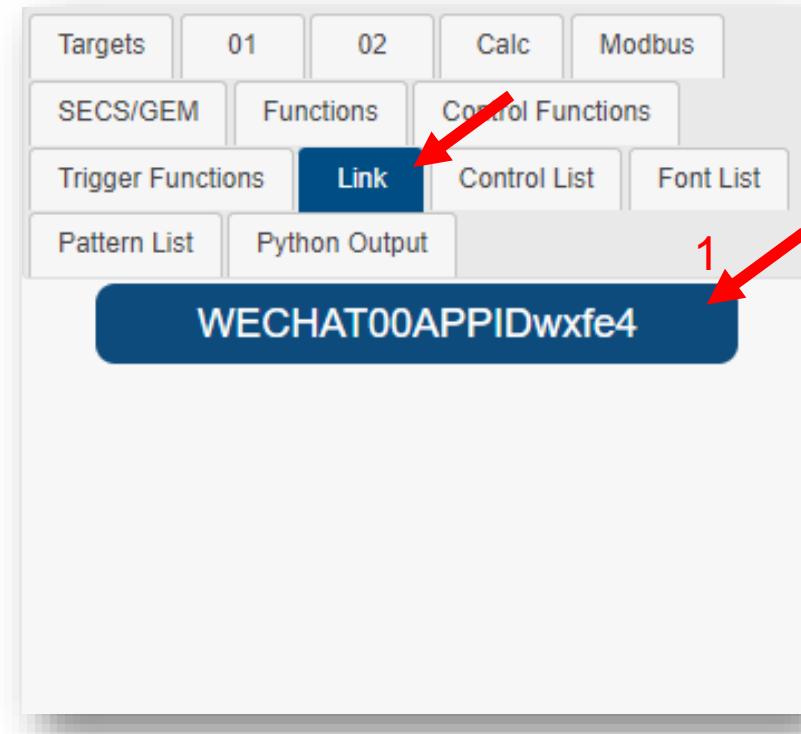
## • 函式

- **SEND.WECHAT**：發送 WeChat 通知給所有關注此公眾號的用戶

1. id：要使用的 WeChat 連線，要先在連結頁面內建立連線

可從連結中選擇要使用的連線

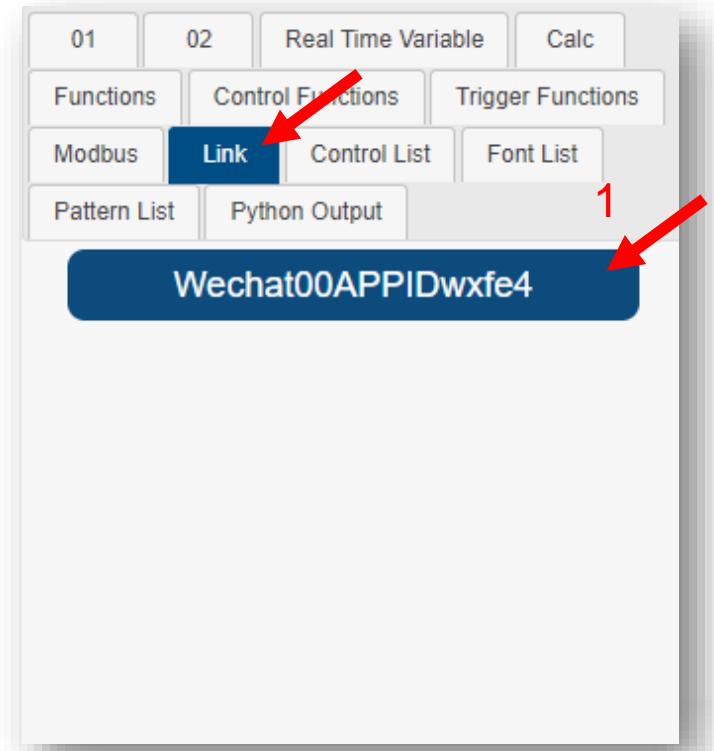
2. templateID：要傳送的模板的ID
3. keyword1：要傳送的模板內的{{keyword1.DATA}}
4. keyword2：要傳送的模板內的{{keyword2.DATA}}
5. keyword3：要傳送的模板內的{{keyword3.DATA}}
6. send image：是否傳送當下畫面(1：傳送，0：不傳送)
7. sec：執行間隔時間



## • 函式

- **SEND.WECHAT\_P**：發送 WeChat 通知給所有關注此公眾號的一個用戶

1. id：要使用的 WeChat 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
2. templateID：要傳送的模板的ID
3. openID：要接收通知的用戶微信號
4. keyword1：要傳送的模板內的{{keyword1.DATA}}
5. keyword2：要傳送的模板內的{{keyword2.DATA}}
6. keyword3：要傳送的模板內的{{keyword3.DATA}}
7. send image：是否傳送當下畫面(1：傳送，0：不傳送)
8. sec：執行間隔時間

**SEND.WECHAT\_P**

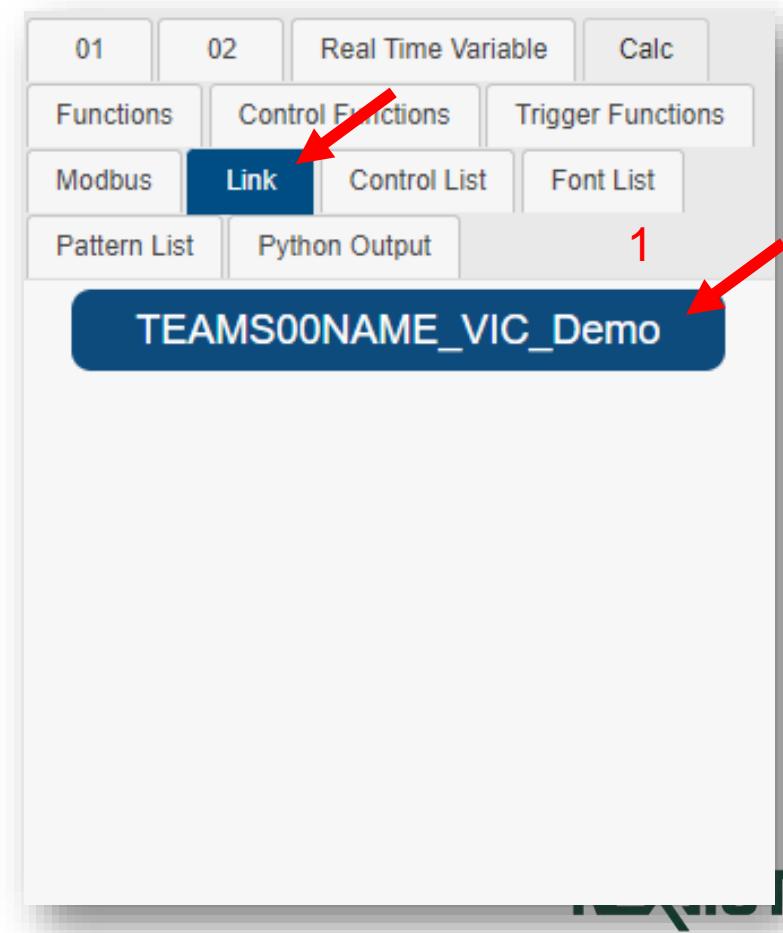
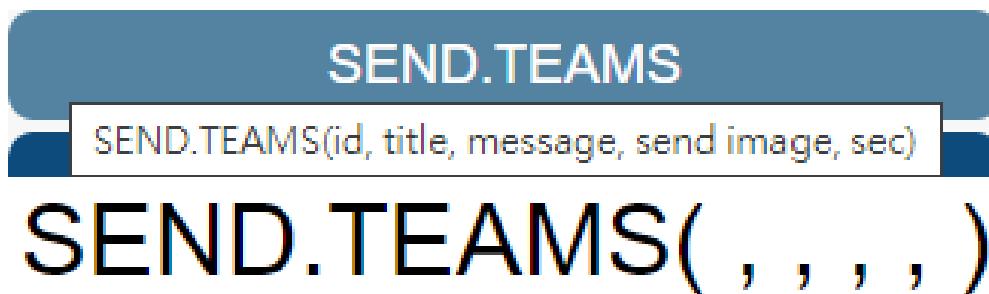
SEND.WECHAT\_P(id, templateID, openID, keyword1, keyword2, keyword3, send image, sec)

**SEND.WECHAT\_P( , , , , , , )**

- 函式

- **SEND.TEAMS**：發送 Teams 通知

1. id：要使用的 Teams 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
2. title：要發送的通知的標題
3. message：要發送的通知的訊息
4. send image：是否傳送當下畫面(1：傳送，0：不傳送)
5. sec：執行間隔時間



- 函式

- RT.PAGE\_NO : 當前辨識頁面的編號

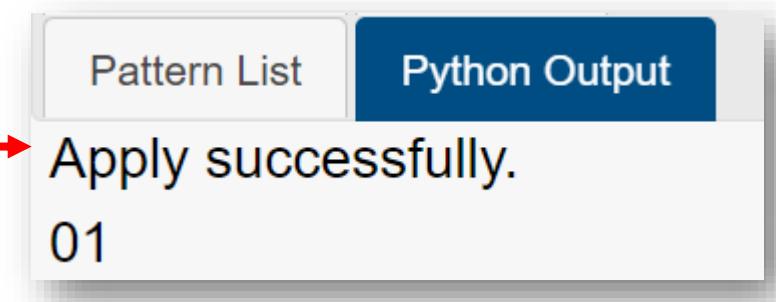


```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    print(RT.PAGE_NO)
```

```
    pass
```



- 函式

- RT.RESULT : 當前辨識結果

0 = Pass

1 = NG

2 = No Matching



RT.RESULT

```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    print(RT.RESULT)
```

```
    pass
```

Apply successfully.

0



- 函式

- RT.CURRENT\_TIME : 執行函式的當前時間，可再細分為

RT.CURRENT\_TIME.YEAR : 執行函式的當前時間的年份

RT.CURRENT\_TIME.MONTH : 執行函式的當前時間的月份

RT.CURRENT\_TIME.DAY : 執行函式的當前時間的日期

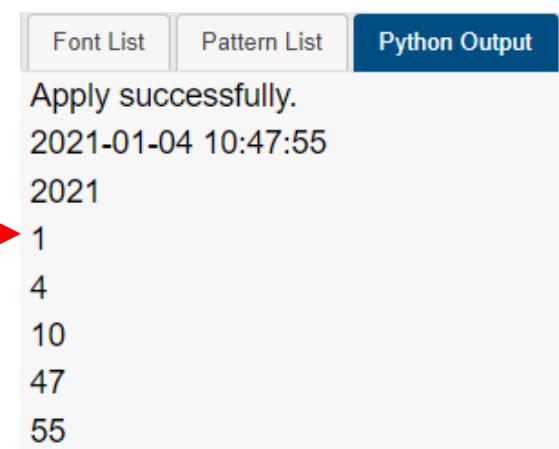
RT.CURRENT\_TIME.HOUR : 執行函式的當前時間的時

RT.CURRENT\_TIME.MINUTE : 執行函式的當前時間的分

RT.CURRENT\_TIME.SECOND : 執行函式的當前時間的秒

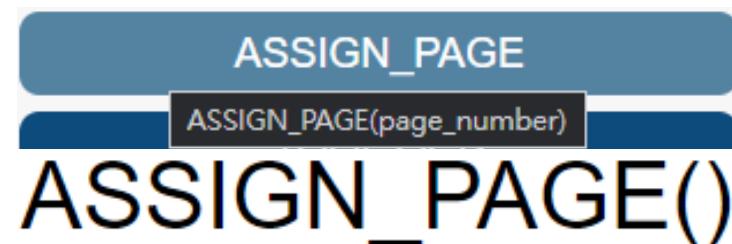
```
from vic import *
def mainLoop(info, arr):
    print(RT.CURRENT_TIME)
    print(RT.CURRENT_TIME.YEAR)
    print(RT.CURRENT_TIME.MONTH)
    print(RT.CURRENT_TIME.DAY)
    print(RT.CURRENT_TIME.HOUR)
    print(RT.CURRENT_TIME.MINUTE)
    print(RT.CURRENT_TIME.SECOND)
    pass
```

RT.CURRENT\_TIME  
RT.CURRENT\_TIME  
RT.CURRENT\_TIME.YEAR  
RT.CURRENT\_TIME.MONTH  
RT.CURRENT\_TIME.DAY  
RT.CURRENT\_TIME.HOUR  
RT.CURRENT\_TIME.MINUTE  
RT.CURRENT\_TIME.SECOND

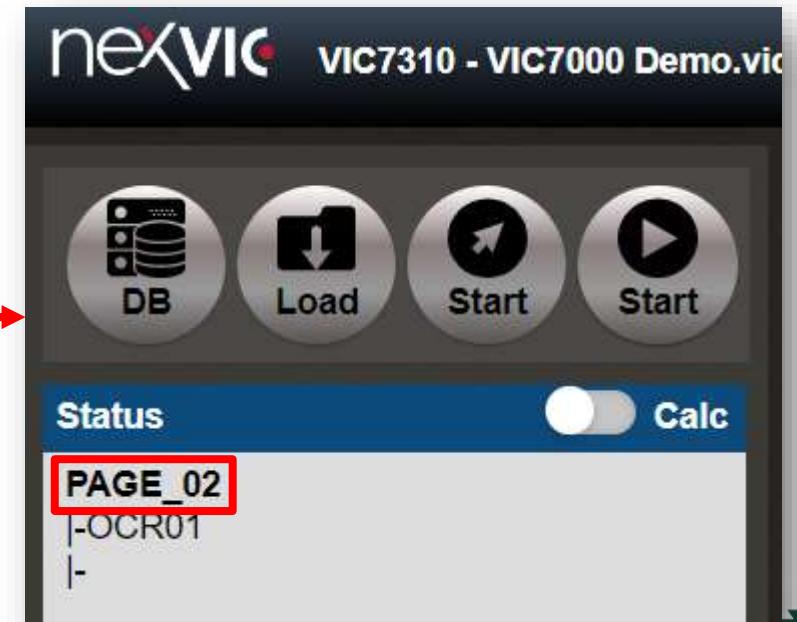


- 函式

- **ASSIGN\_PAGE** : 指定要使用的辨識頁面，此功能在有管理者登入時無效



```
def mainLoop(info, arr):  
    ASSIGN_PAGE('02')  
    pass
```



## • 函式

- **TOOL.OCR** : 在當前辨識的圖像上進行文字辨識

1. select : 選擇字型

2. fontName : 要使用的辨識字型檔名稱 ,

可從字型檔列表選擇

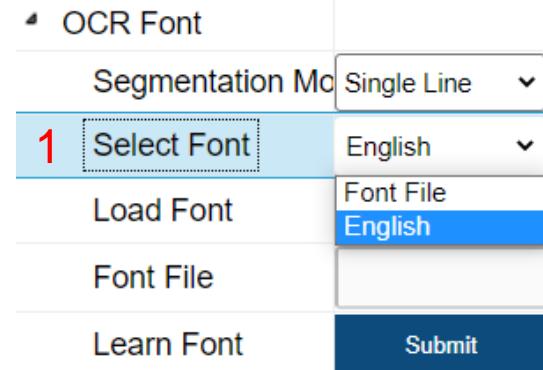
3. roi\_x : 辨識 ROI 的 x 座標

4. roi\_y : 辨識 ROI 的 y 座標

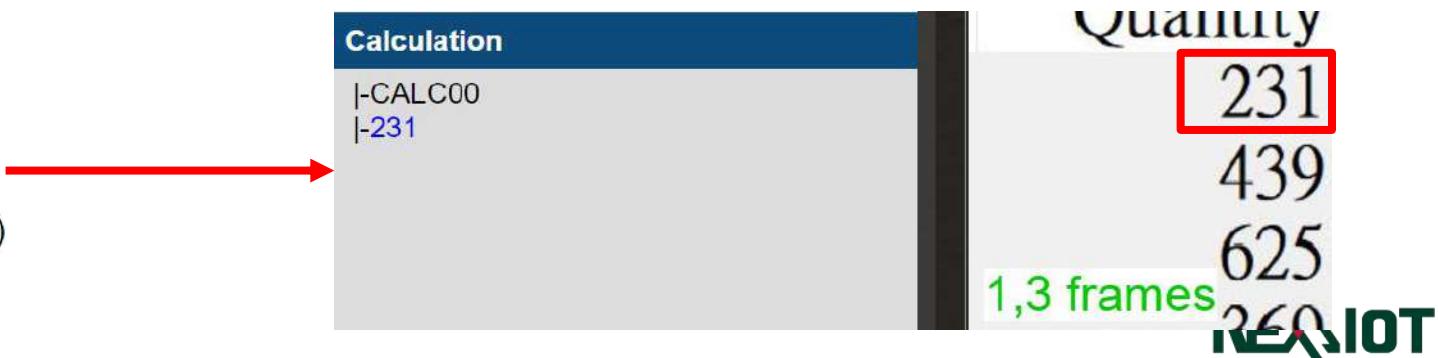
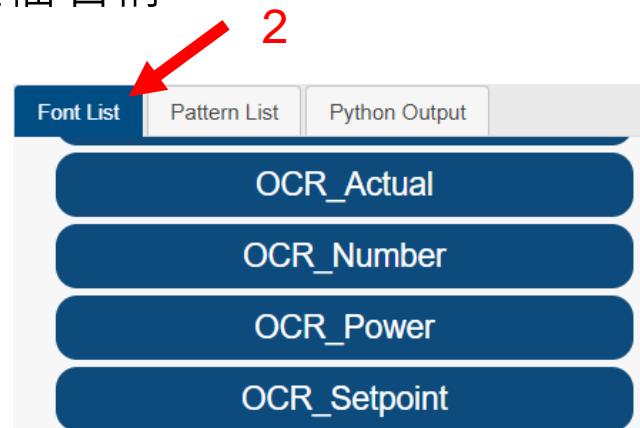
5. roi\_width : 辨識 ROI 的寬度

6. roi\_height : 辨識 ROI 的高度

```
from vic import *
def mainLoop(info, arr):
    CALC[0]=TOOL.OCR( 0,"OCR_Number" , 360,172,73,43 )
    pass
```



0: 使用字型檔  
1: 使用預設字型庫



231

439

625

1,3 frames

260 IOT

## • 函式

• **TOOL.OCR** : 進階應用

1. **select** : 選擇字型
2. **fontName** : 要使用的辨識字型檔名稱，可從字型檔列表選擇
3. **roi\_x** : 辨識 ROI 的 x 座標
4. **roi\_y** : 辨識 ROI 的 y 座標
5. **roi\_width** : 辨識 ROI 的寬度
6. **roi\_height** : 辨識 ROI 的高度
7. **img** : 要辨識的圖片
8. **params** : 前處理參數和OCR設置

```
CALC[0] = TOOL.OCR(0, 'OCR_Number', 360, 172, 73, 43, arr,  
{'segmentation':a, 'preprocess_resize':b,  
'preprocess_resize_method': c, 'preprocess_threshold_method':d,  
'preprocess_threshold_algorithm': e, 'preprocess_threshold_value': f,  
'allow_empty_string':g, 'allow_recognition_rate':h,  
'remove_whitespace':i, 'white_list':j})
```

```
params = {'segmentation':a, 'preprocess_resize':b, 'preprocess_resize_method': c, 'preprocess_threshold_method':d,  
'preprocess_threshold_algorithm': e, 'preprocess_threshold_value': f, 'allow_empty_string':g, 'allow_recognition_rate':h,  
'remove_whitespace':i, 'white_list':j}
```

## • 函式

- TOOL.OCR : 進階應用

‘segmentation’:a

a = 0 ~ 13

a	Mode
0	PSM OSD ONLY
1	PSM AUTO OSD
2	PSM AUTO ONLY
3	PSM AUTO
4	PSM SINGLE COLUMN
5	PSM SINGLE BLOCK VERT TEXT
6	PSM SINGLE BLOCK

a	Mode
7	PSM SINGLE LINE
8	PSM SINGLE WORD
9	PSM CIRCLE WORD
10	PSM SINGLE CHAR
11	PSM SPARSE TEXT
12	PSM SPARSE TEXT OSD
13	PSM RAW LINE

{'segmentation':a, 'preprocess\_resize':b,

‘preprocess\_resize’:b

b = 0 ~ 9

b	Resize
0	x1
1	x2
2	x3
3	x4
4	x5
5	x6
6	x7
7	x8
8	x9
9	x10

## • 函式

'preprocess\_resize\_method': c, 'preprocess\_threshold\_method': d,

- **TOOL.OCR** : 進階應用

'preprocess\_resize\_method':c

c = 0 ~ 4

'preprocess\_threshold\_method':d

d = 0 ~ 2

c	Method
0	INTER_NEARSET
1	INTER_LINEAR
2	INTER_CUBIC
3	INTER_AREA
4	INTER_LANCZOS4

d	Method
0	None
1	Binary
2	Binary Inverse

- 函式 'preprocess\_threshold\_algorithm': e, 'preprocess\_threshold\_value': f,

- **TOOL.OCR** : 進階應用

'preprocess\_threshold\_algorithm':e

e = 0 ~ 2

'preprocess\_threshold\_method':d

f = 0 ~ 255

Binary Threshold Value

e	Method
0	None
1	THRESH_OTSU
2	THRESH_TRIANGLE

## • 函式

'allow\_empty\_string':g, 'allow\_recognition\_rate':h,'remove\_whitespace':i, 'white\_list':j})

- **TOOL.OCR** : 進階應用

‘allow\_empty\_string’:g

g = True/False

‘allow\_recognition\_rate’:h

f = 0 ~ 100

‘remove\_whitespace’:i

i = True/False

‘white\_list’:j

j = string

g / i	Method
True	Enable
False	Disable

## • 函式

- **TOOL.COLOR**：在當前辨識的圖像上進行顏色比對

1. red : 要比對的 RGB 色域紅色值
2. green : 要比對的 RGB 色域綠色值
3. blue : 要比對的 RGB 色域藍色值
4. roi\_x : 辨識 ROI 的 x 座標
5. roi\_y : 辨識 ROI 的 y 座標
6. roi\_width : 辨識 ROI 的寬度
7. roi\_height : 辨識 ROI 的高度
8. tolerance : 比對容錯度

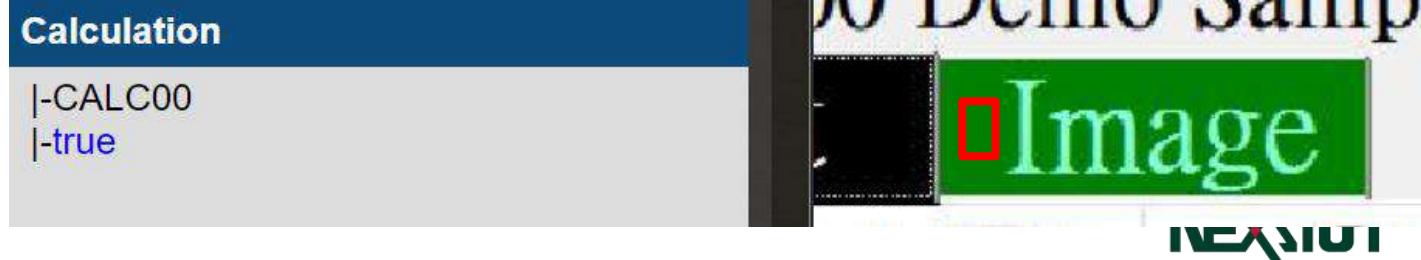
TOOL.COLOR  
TOOL.COLOR( red, green, blue, roi\_x, roi\_y, roi\_width, roi\_height, tolerance )

```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    CALC[0]=TOOL.COLOR(0,128,0,161,89,13,13,5)
```

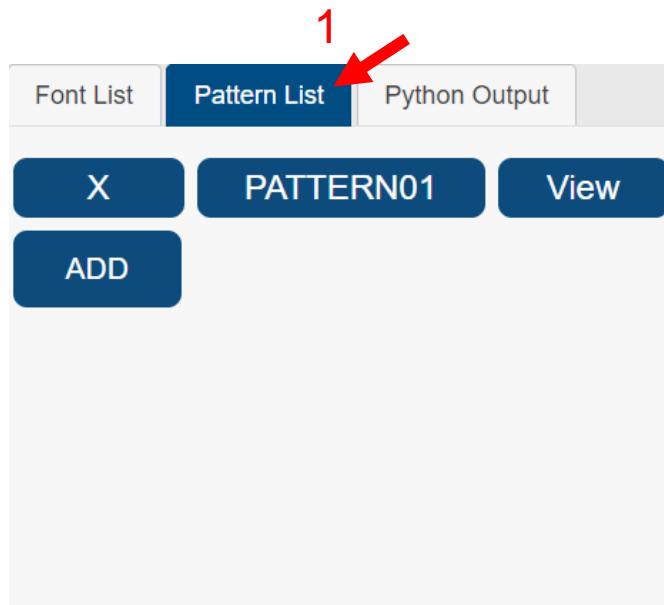
```
    pass
```



- 函式

- TOOL.PATTERN：在當前辨識的圖像上進行樣式比對

- pattern\_id：要比對的樣式，從樣式列表進行選擇和設定
- roi\_x：辨識 ROI 的 x 座標
- roi\_y：辨識 ROI 的 y 座標
- roi\_width：辨識 ROI 的寬度
- roi\_height：辨識 ROI 的高度



### TOOL.PATTERN

TOOL.PATTERN(pattern\_id, roi\_x, roi\_y, roi\_width, roi\_height)

TOOL.PATTERN( , , , , )

```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    CALC[0]=TOOL.PATTERN("PATTERN01",1142,35,205,63)
```

```
    pass
```

### Calculation

```
|CALC00
|true
```



## • 函式

## • TOOL.PATTERN : 進階應用

1. pattern\_id : 要比對的樣式，從樣式列表進行選擇和設定
2. roi\_x : 辨識 ROI 的 x 座標
3. roi\_y : 辨識 ROI 的 y 座標
4. roi\_width : 辨識 ROI 的寬度
5. roi\_height : 辨識 ROI 的高度
6. img : 要辨識的圖片
7. params : 樣式比對參數

```
params = {'gray_match':True/False, 'image_level':a, 'method': b, 'min_score':0~1.0}
```

gray\_match : 是否使用灰階比對

min\_score : 比對最小分數

```
CALC[0] = TOOL.PATTERN('PATTERN01',  
1142, 35, 205, 63, arr,  
{'gray_match':True/False, 'image_level':a,  
'method': b, 'min_score':0.95})
```

image\_level : 圖像層數，a = 0~5

method : 比對方法，b = 3 or 5

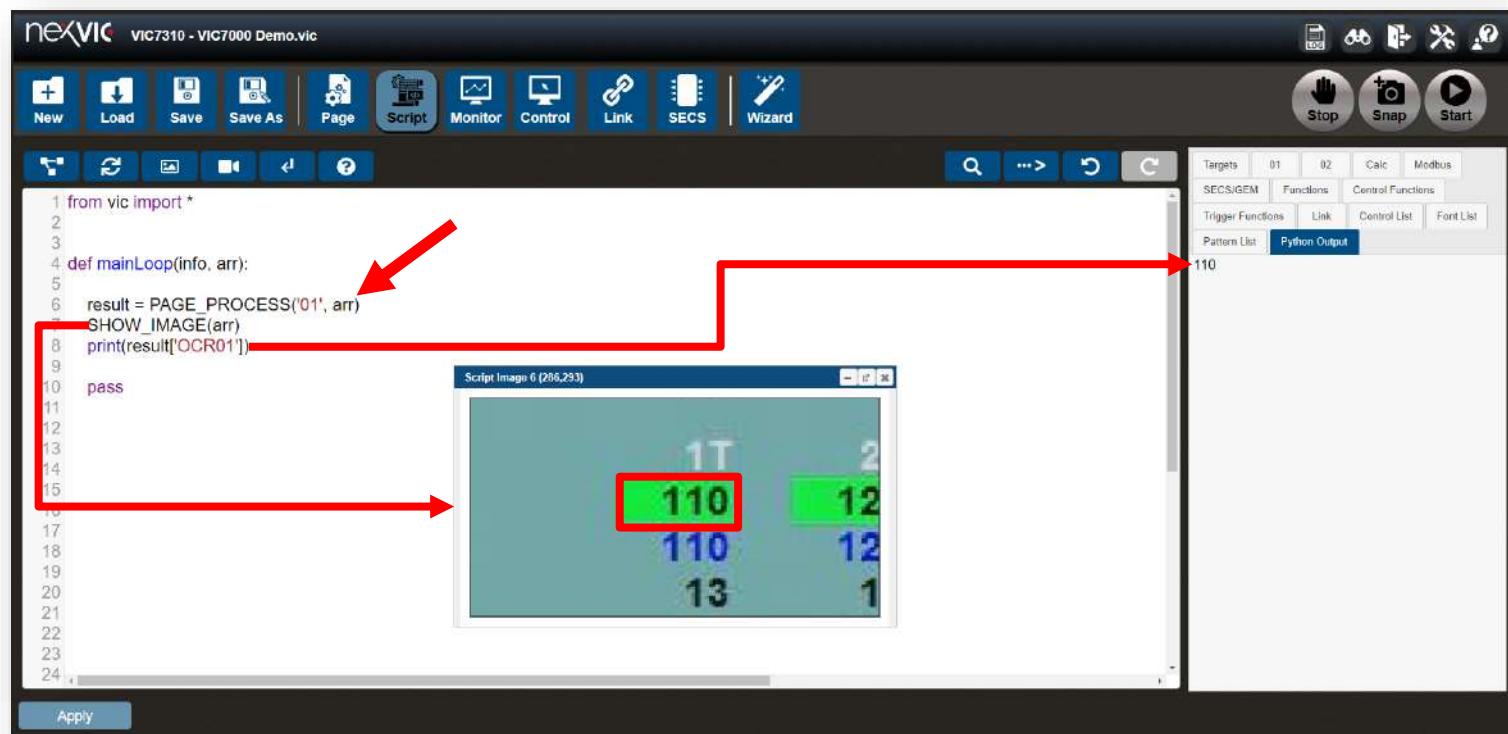
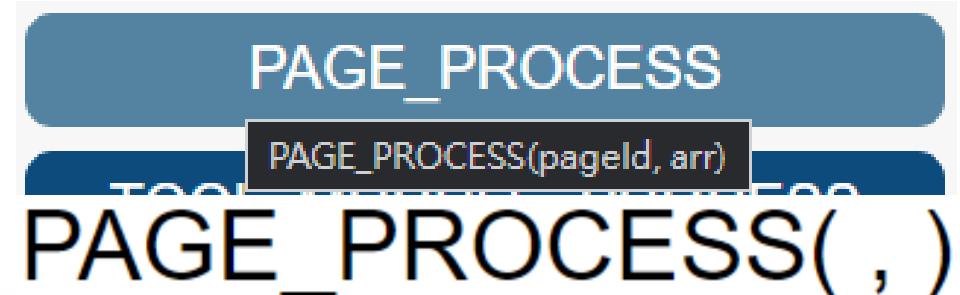
3 = 互相關，5 = 相關係數

- 函式

- **PAGE\_PROCESS** : 使用特定的辨識頁面設定來進行辨識

需要使用一個變數來儲存辨識結果

1. `pageIndex` : 要使用的辨識頁面編號
2. `arr` : 要辨識的圖片



- 函式

- **TOOL\_MODULE\_PROCESS**：使用工具插件來進行處理或辨識

1. name：要使用的工具插件名稱
2. properties：要使用工具插件參數，格式為 Dictionary
3. arr：要處理或辨識的圖片
4. roi：設置要處理或辨識的圖片的ROI，格式為 Dictionary

輸入內容為{'x':ROI的X座標, 'y':ROI的Y座標, 'width':ROI的寬度, 'height':ROI的高度}



TOOL\_MODULE\_PROCESS

... TOOL\_MODULE\_PROCESS(name, properties, arr, roi) ...

TOOL\_MODULE\_PROCESS( , , , )

## • 函式

- **SAVE\_IMAGE**：儲存當前辨識的圖片到本地

1. **filepath**：儲存圖片的資料夾路徑
2. **sec**：執行間隔時間

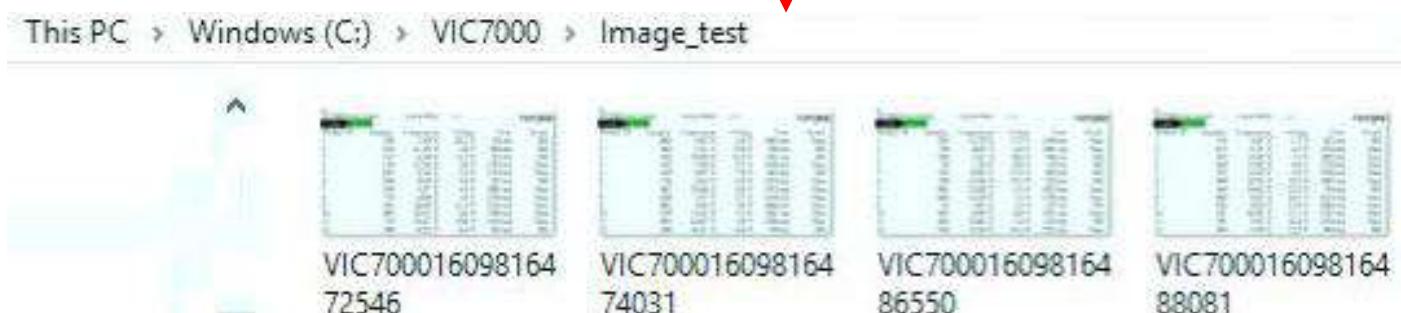


```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    SAVE_IMAGE('C:\\VIC7000\\Image_test', 1 )
```

```
    pass
```

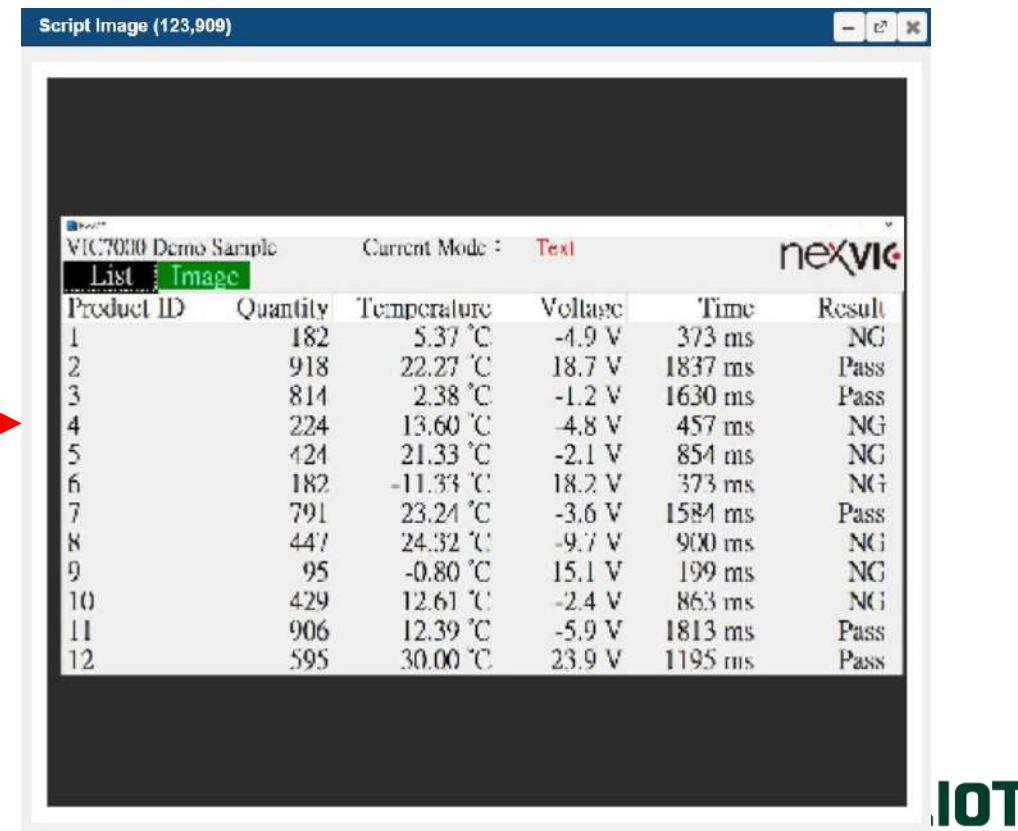


## • 函式

- **SHOW\_IMAGE**：在腳本圖像中顯示圖片

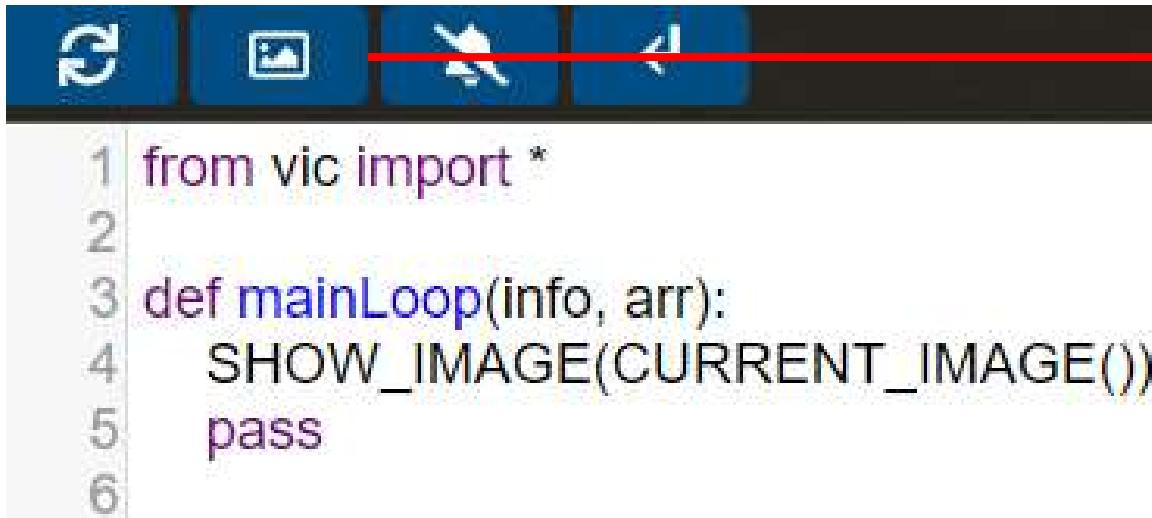
1. image：要顯示的圖片

mainLoop 內的 arr 代表專案啟動時辨識的圖片



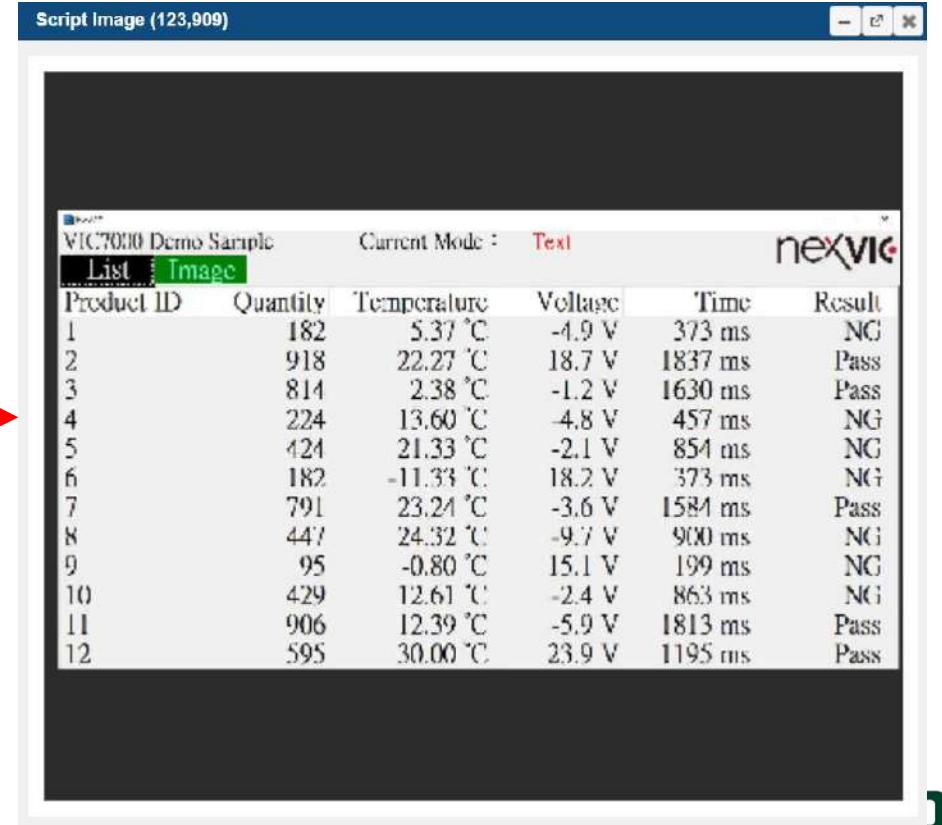
## • 函式

- CURRENT\_IMAGE : 當前辨識的圖片



```
1 from vic import *
2
3 def mainLoop(info, arr):
4     SHOW_IMAGE(CURRENT_IMAGE())
5     pass
6
```

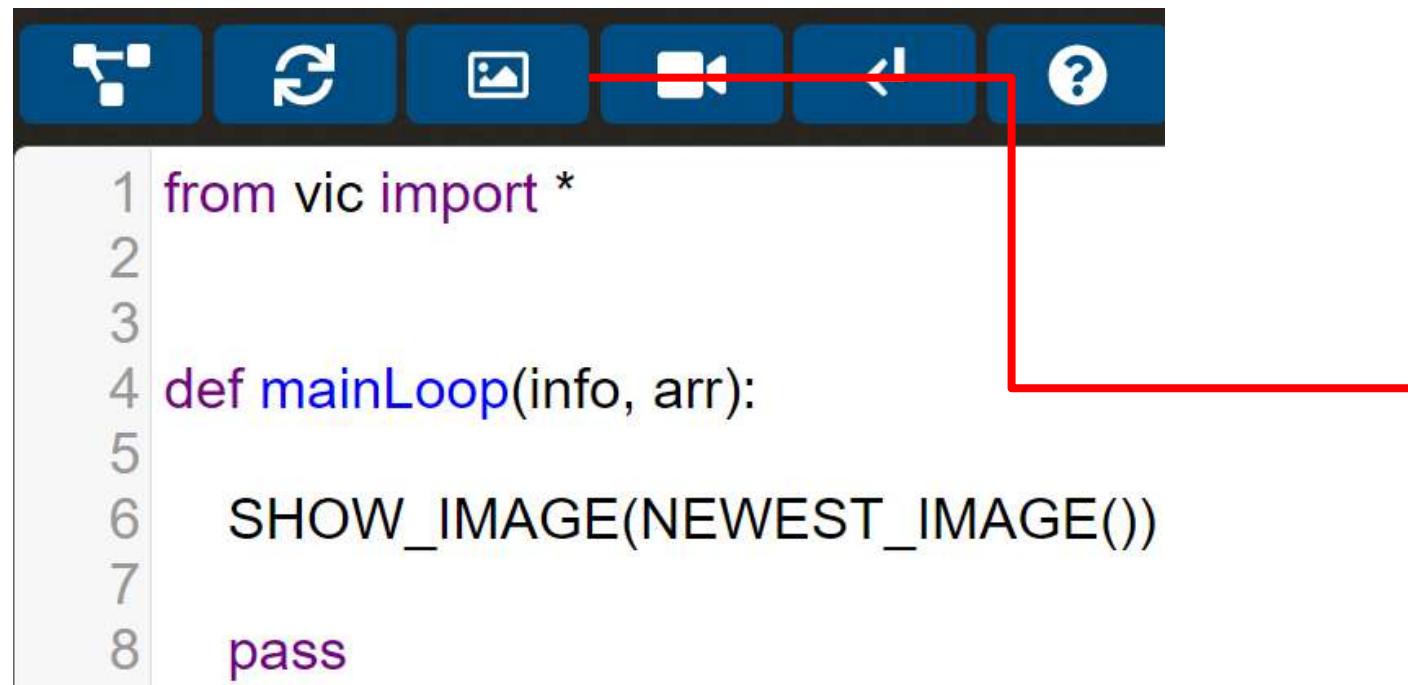
Script Image (123,909)



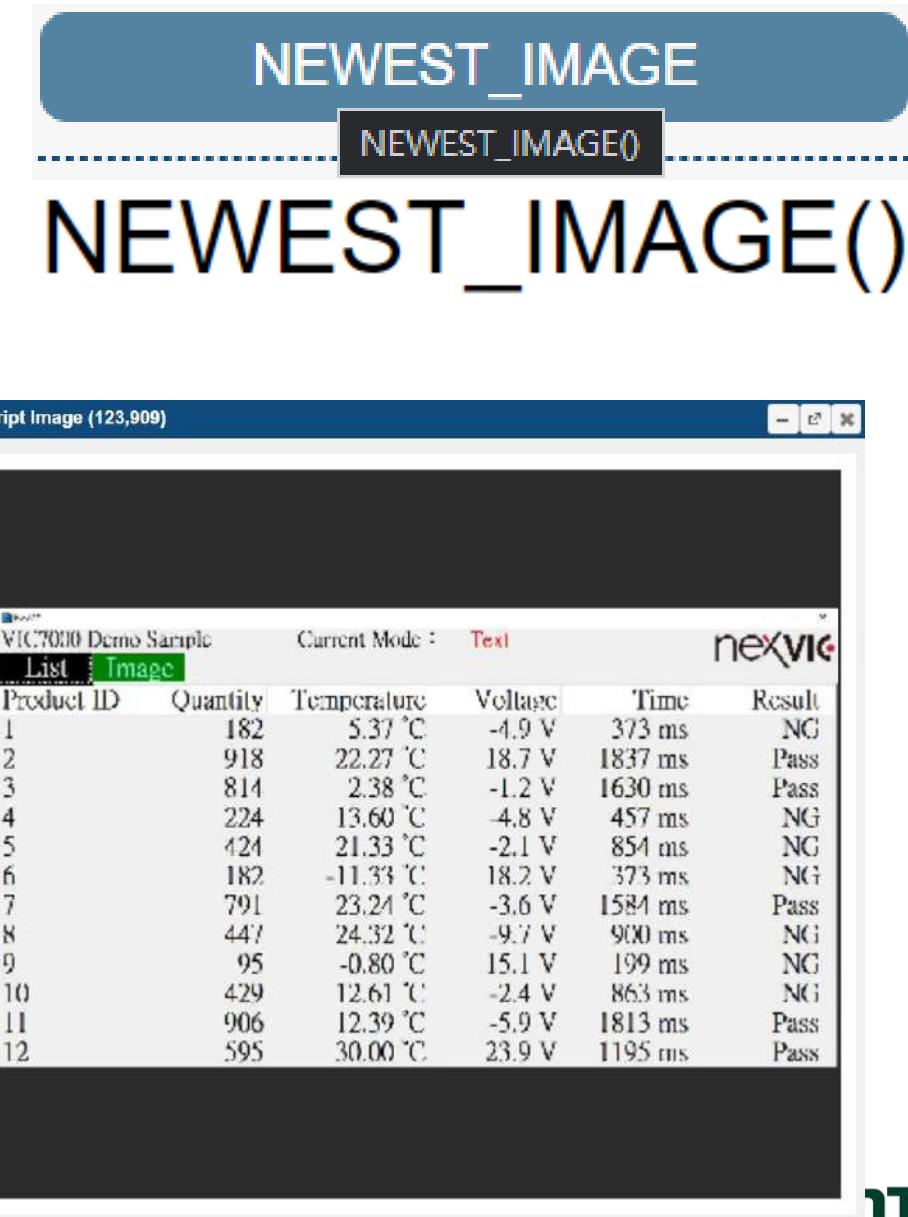
VIC7000 Demo Sample		Current Mode :	Text	nexVIC	
Product ID	Quantity	Temperature	Voltage	Time	Result
1	182	5.37 °C	-4.9 V	373 ms	NG
2	918	22.27 °C	18.7 V	1837 ms	Pass
3	814	2.38 °C	-1.2 V	1630 ms	Pass
4	224	13.60 °C	-4.8 V	457 ms	NG
5	424	21.33 °C	-2.1 V	854 ms	NG
6	182	-11.33 °C	18.2 V	373 ms	NG
7	791	23.24 °C	-3.6 V	1581 ms	Pass
8	447	24.32 °C	-9.7 V	900 ms	NG
9	95	-0.80 °C	15.1 V	199 ms	NG
10	429	12.61 °C	-2.4 V	863 ms	NG
11	906	12.39 °C	-5.9 V	1813 ms	Pass
12	595	30.00 °C	23.9 V	1195 ms	Pass

## ● 函式

- NEWEST\_IMAGE : 由取像來源取得一張最新的圖像



```
1 from vic import *
2
3
4 def mainLoop(info, arr):
5
6     SHOW_IMAGE(NEWEST_IMAGE())
7
8 pass
```



- 函式

- **PLAY.PROJECT** : 設定專案啟動/停止

1. status : 專案執行狀態

0 = 啟動專案

1 = 停止專案



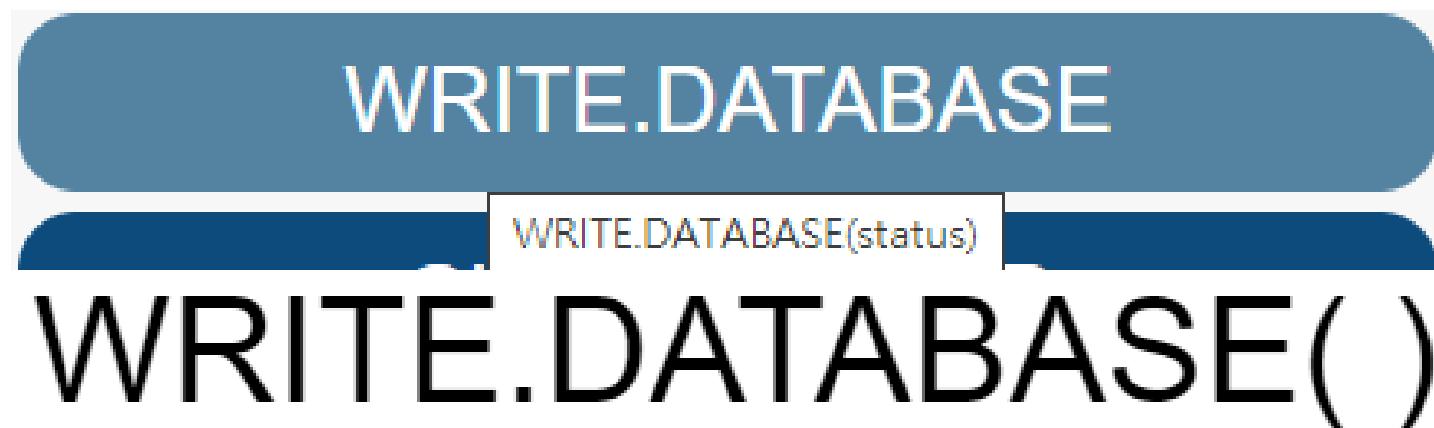
- 函式

- **WRITE.DATABASE**：設定辨識資料是否寫入資料庫

1. status：資料庫寫入狀態

0 = 不寫入資料庫

1 = 寫入資料庫



- 函式

- **CHECK.COLOR** : 偵測顏色是否閃爍

1. channel\_color : 要被偵測的顏色

2. calc : 存放偵測結果的運算值

若在偵測時間內顏色比對結果有改變則輸出為 1，反之則為 0

3. sec : 偵測時間



## • 函式

- **WRITE\_TEXT**：寫入文字到本地文字檔

1. **filepath**：儲存絕對路徑或檔案名稱

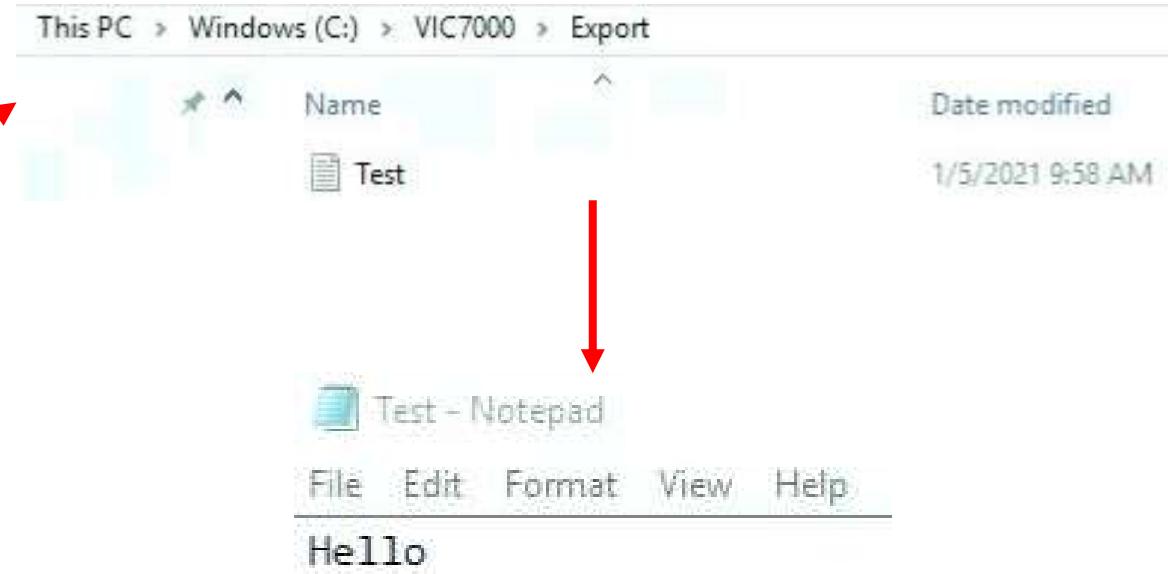
輸入儲存絕對路徑：文字檔會被儲存在該路徑

輸入檔案名稱：文字檔會被儲存在 VIC7000 資料夾內的 Export

2. **content**：要寫入文字檔的內容

3. **sec**：執行間隔時間

```
from vic import *\n\ndef mainLoop(info, arr):\n    WRITE_TEXT('Test','Hello', 1)\n    pass
```



- 函式

- **WRITE\_TEXT\_LN** : 寫入文字到本地文字檔，寫入後換行

1. **filepath** : 儲存絕對路徑或檔案名稱

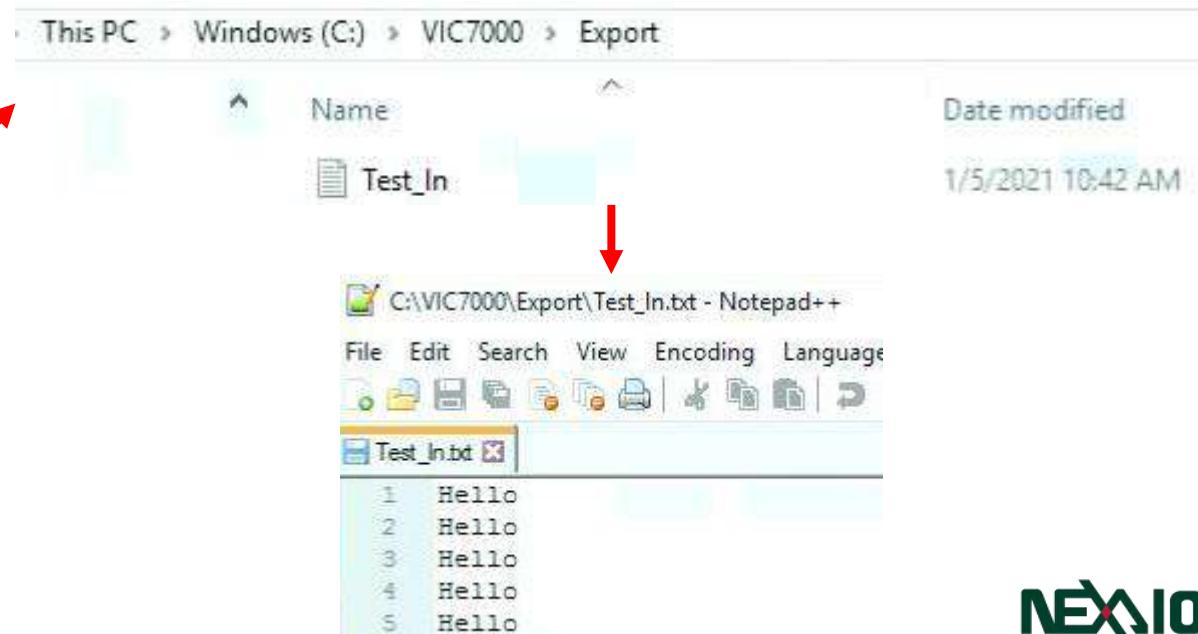
輸入儲存絕對路徑：文字檔會被儲存在該路徑

輸入檔案名稱：文字檔會被儲存在 VIC7000 資料夾內的 Export

2. **content** : 要寫入文字檔的內容

3. **sec** : 執行間隔時間

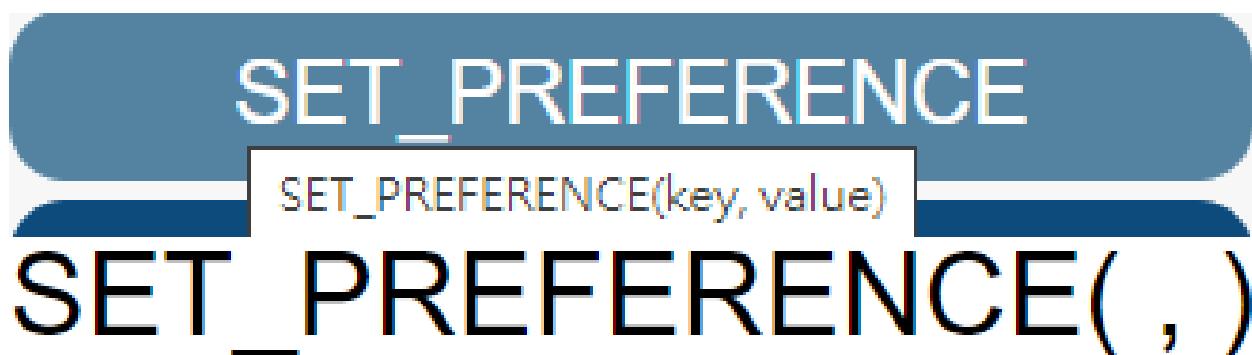
```
from vic import *
def mainLoop(info, arr):
    WRITE_TEXT_LN('Test_In','Hello', 1)
    pass
```



- 函式

- **SET\_PREFERENCE** : 設定系統變數的值

1. key : 變數名稱
2. value : 設定值



```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    SET_PREFERENCE('abc','123')
```

```
    pass
```

- 函式

- **GET\_PREFERENCE** : 取得系統變數的值

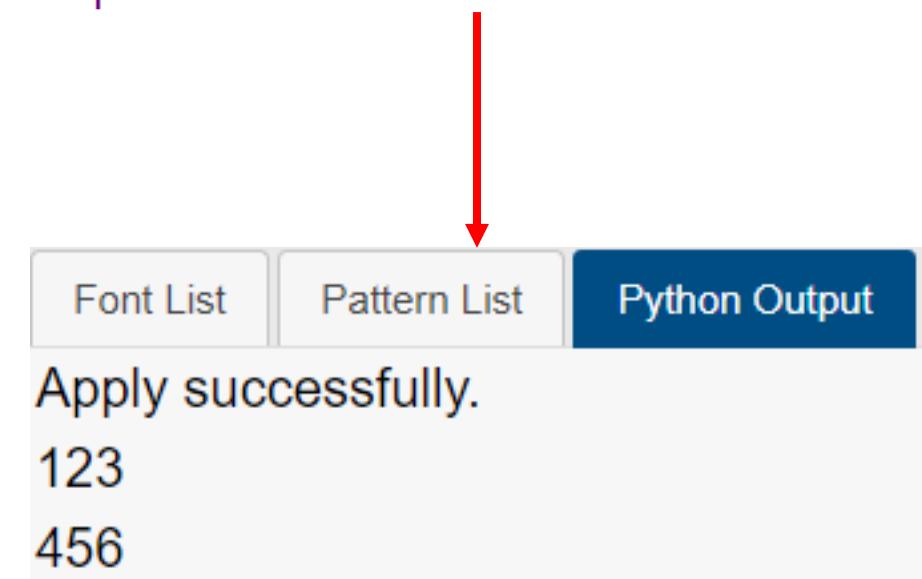
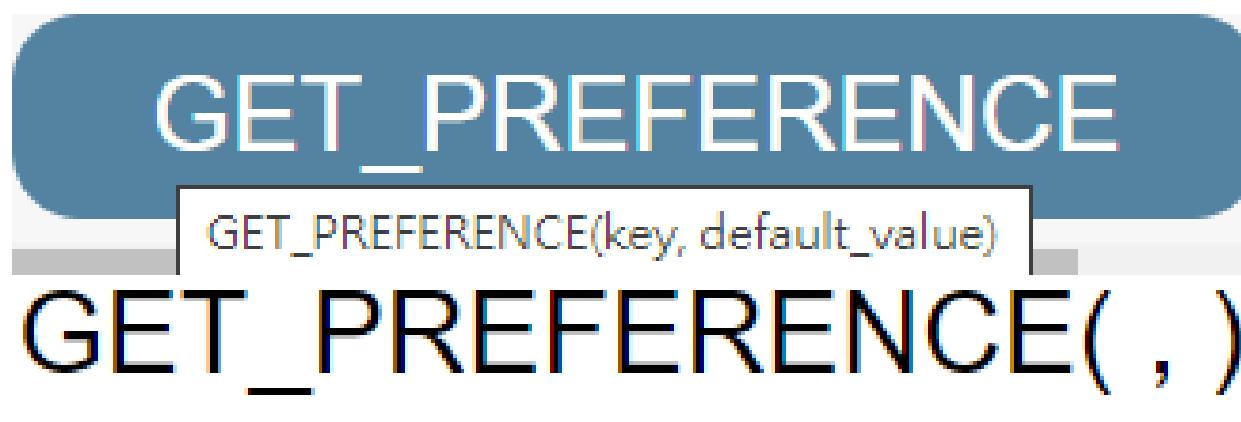
1. key : 變數名稱
2. default : 若未找到該變數的預設值

```
from vic import *
```

```
def mainLoop(info, arr):
```

```
    print(GET_PREFERENCE('abc','456'))  
    print(GET_PREFERENCE('def','456'))
```

```
    pass
```



- 函式

- **SHOW\_DIALOG**：彈出提示視窗

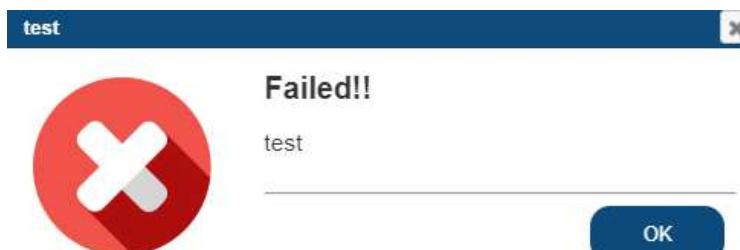
1. type：視窗類型

0 = 成功，1 = 失敗，2 = 資訊，3 = 警告

2. title：視窗標題

3. content：訊息內容

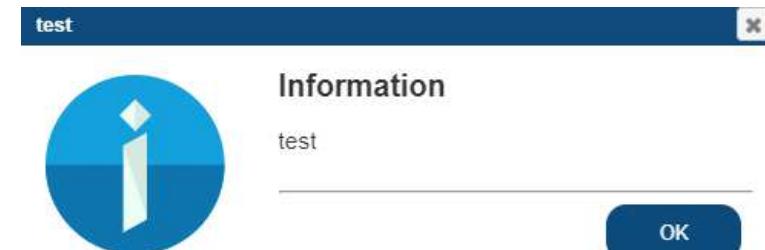
4. sec : 執行間隔時間



**SHOW\_DIALOG**

**SHOW\_DIALOG(type, title, content, sec)**

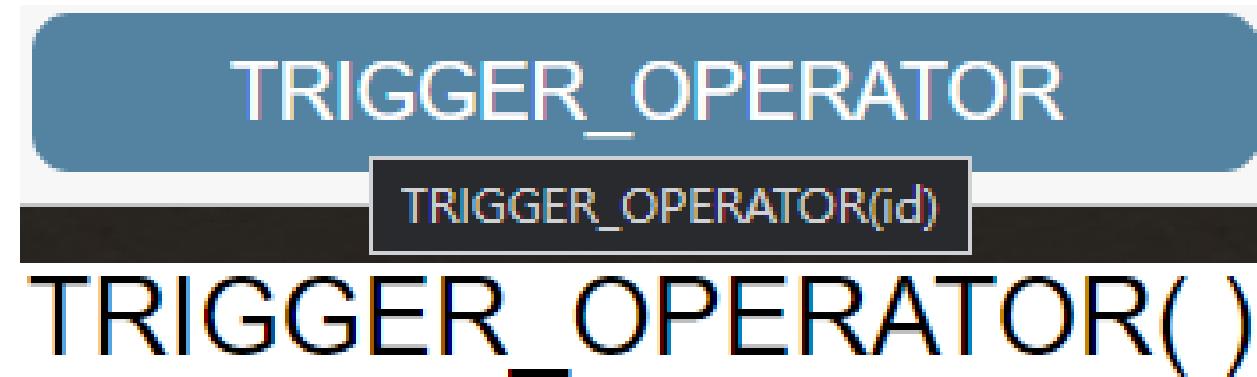
**SHOW\_DIALOG( , , , )**



- 函式

- **TRIGGER\_OPERATOR**：觸發VIC Flow內的觸發運算子

1. id：要觸發的觸發運算子的 id



- **控制函式**

- **PLAY.CONTROL** : 設定控制啟動停止

- 1. **status** : 控制執行狀態

- 0 = 啟動控制

- 1 = 停止控制



- **控制函式**

- **CLICK**：控制滑鼠游標移動到特定位置，並點擊一次

1. control\_type：要使用的控制方式

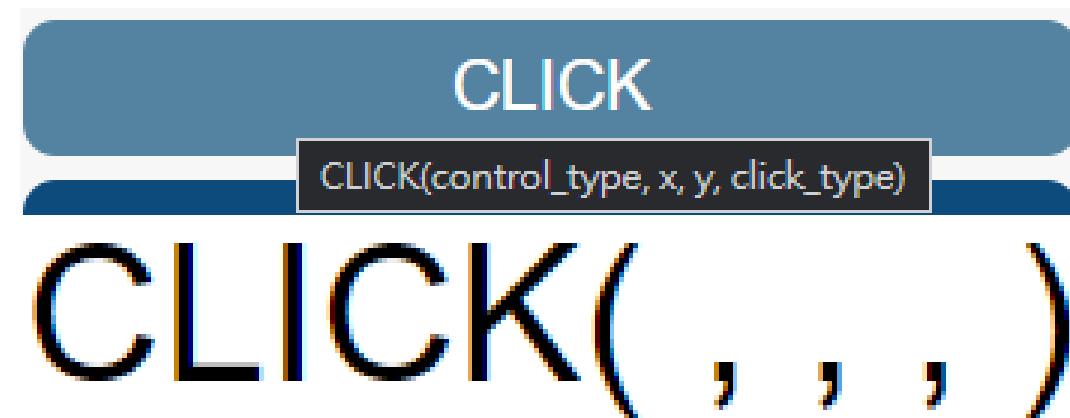
0 = USB · 1 = 本機

2. x：目標 x 座標

3. y：目標 y 座標

4. click\_type：點擊按鍵類型

1 = 左鍵 · 2 = 滑鼠中鍵 · 3 = 右鍵



- **控制函式**

- **DB\_CLICK**：控制滑鼠游標移動到特定位置，並連點兩次

1. `control_type`：要使用的控制方式

`0 = USB` , `1 = 本機`

2. `x`：目標 `x` 座標

3. `y`：目標 `y` 座標

4. `click_type`：點擊按鍵類型

`1 = 左鍵` , `2 = 滑鼠中鍵` , `3 = 右鍵`



- **控制函式**

- **DRAG**：控制滑鼠游標從一點移動另一點，同時進行拖曳

1. `control_type`：要使用的控制方式

0 = USB , 1 = 本機

2. `click_type`：點擊按鍵類型

1 = 左鍵 , 2 = 滑鼠中鍵 , 3 = 右鍵

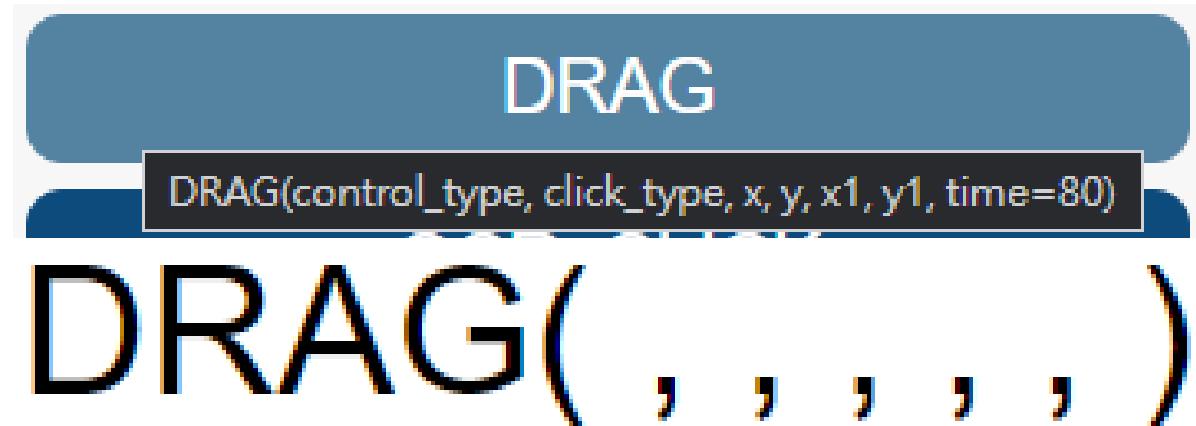
3. `x`：拖曳起點 x 座標

4. `y`：拖曳起點 y 座標

5. `x1`：拖曳終點 x 座標

6. `y1`：拖曳終點 y 座標

7. `time`：拖曳時間，單位為毫秒(ms)，預設為80



- **控制函式**

- **OCR\_CLICK**：控制滑鼠在符合比對條件的文字的位置點擊一次

1. `control_type`：要使用的控制方式

0 = USB , 1 = 本機

2. `click_type`：點擊按鍵類型

1 = 左鍵 , 2 = 滑鼠中鍵 , 3 = 右鍵

3. `text`：文字比對條件

4. `select`：選擇字型

5. `fontName`：要使用的辨識字型檔名稱，可從字型檔列表選擇

6. `roi_x`：辨識 ROI 的 x 座標

7. `roi_y`：辨識 ROI 的 y 座標

8. `roi_width`：辨識 ROI 的寬度

9. `roi_height`：辨識 ROI 的高度



- **控制函式**

- **OCR\_CLICK** : 進階使用

1. **control\_type** : 要使用的控制方式 , 0 = USB , 1 = 本機
2. **click\_type** : 點擊按鍵類型 , 1 = 左鍵 , 2 = 滑鼠中鍵 , 3 = 右鍵
3. **text** : 文字比對條件
4. **select** : 選擇字型
5. **fontName** : 要使用的辨識字型檔名稱 , 可從字型檔列表選擇
6. **roi\_x** : 辨識 ROI 的 x 座標
7. **roi\_y** : 辨識 ROI 的 y 座標
8. **roi\_width** : 辨識 ROI 的寬度
9. **roi\_height** : 辨識 ROI 的高度
10. **img** : 要辨識的圖像
11. **params** : 前處理參數和OCR設置

```
params = {'segmentation':a, 'preprocess_resize':b,  
'preprocess_resize_method': c, 'preprocess_threshold_method':d,  
'preprocess_threshold_algorithm': e,  
'preprocess_threshold_value': f, 'allow_empty_string':g,  
'allow_recognition_rate':h,  
'remove_whitespace':i, 'white_list':j}
```

- 控制函式

{'segmentation':a, 'preprocess\_resize':b,

- **OCR\_CLICK** : 進階使用

‘segmentation’:a

a = 0 ~ 13

a	Mode
0	PSM OSD ONLY
1	PSM AUTO OSD
2	PSM AUTO ONLY
3	PSM AUTO
4	PSM SINGLE COLUMN
5	PSM SINGLE BLOCK VERT _TEXT
6	PSM SINGLE BLOCK

a	Mode
7	PSM SINGLE LINE
8	PSM SINGLE WORD
9	PSM CIRCLE WORD
10	PSM SINGLE CHAR
11	PSM SPARSE TEXT
12	PSM SPARSE TEXT _OSD
13	PSM RAW LINE

‘preprocess\_resize’:b

b = 0 ~ 9

b	Resize
0	x1
1	x2
2	x3
3	x4
4	x5
5	x6
6	x7
7	x8
8	x9
9	x10

## • 控制函式

'preprocess\_resize\_method': c, 'preprocess\_threshold\_method': d,

- 
- OCR\_CLICK
- : 進階使用

'preprocess\_resize\_method':c

c = 0 ~ 4

'preprocess\_threshold\_method':d

d = 0 ~ 2

c	Method
0	INTER_NEARSET
1	INTER_LINEAR
2	INTER_CUBIC
3	INTER_AREA
4	INTER_LANCZOS4

d	Method
0	None
1	Binary
2	Binary Inverse

- **控制函式** 'preprocess\_threshold\_algorithm': e, 'preprocess\_threshold\_value': f,
  - **OCR\_CLICK** : 進階使用

‘preprocess\_threshold\_algorithm’:e

e = 0 ~ 2

‘preprocess\_threshold\_method’:d

f = 0 ~ 255

Binary Threshold Value

e	Method
0	None
1	THRESH_OTSU
2	THRESH_TRIANGLE

'allow\_empty\_string':g, 'allow\_recognition\_rate':h,'remove\_whitespace':i, 'white\_list':j})

- **控制函式**

- **OCR\_CLICK** : 進階使用

‘allow\_empty\_string’:g

g = True/False

‘allow\_recognition\_rate’:h

f = 0 ~ 100

‘remove\_whitespace’:i

i = True/False

‘white\_list’:j

j = string

g / i	Method
True	Enable
False	Disable

- **控制函式**

- **OCR\_DB\_CLICK** : 控制滑鼠在符合比對條件的文字的位置連點兩次，進階使用同 **OCR\_CLICK**

1. **control\_type** : 要使用的控制方式，0 = USB，1 = 本機

2. **click\_type** : 點擊按鍵類型

- 1 = 左鍵，2 = 滑鼠中鍵，3 = 右鍵

3. **text** : 文字比對條件

4. **select** : 選擇字型

5. **fontName** : 要使用的辨識字型檔名稱，可從字型檔列表選擇

6. **roi\_x** : 辨識 ROI 的 x 座標

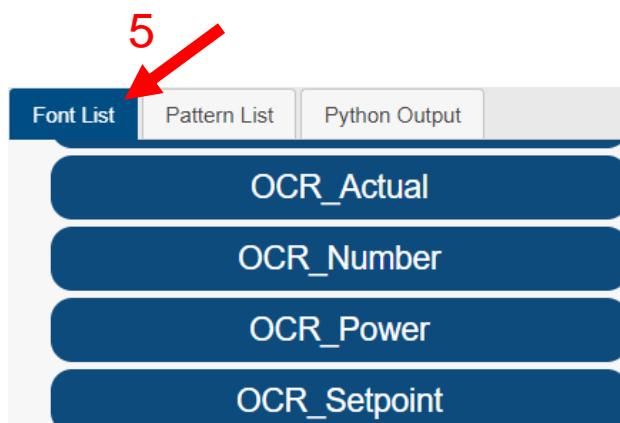
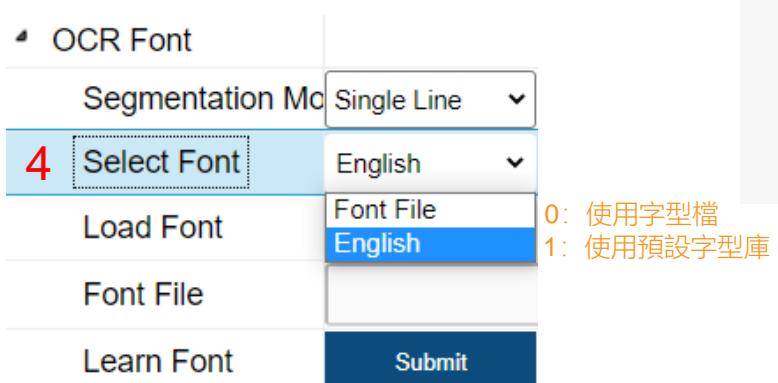
7. **roi\_y** : 辨識 ROI 的 y 座標

8. **roi\_width** : 辨識 ROI 的寬度

9. **roi\_height** : 辨識 ROI 的高度

**OCR\_DB\_CLICK**  
OCR\_DB\_CLICK(control\_type, click\_type, text, select, font\_name, roi\_x, roi\_y, roi\_width, roi\_height)

**OCR\_DB\_CLICK( , , , , , , , )**



- **控制函式**

- **PATTERN\_CLICK** : 控制滑鼠在符合比對條件的樣式的位置點擊一次

1. `control_type` : 要使用的控制方式

`0 = USB` , `1 = 本機`

2. `click_type` : 點擊按鍵類型

`1 = 左鍵` , `2 = 滑鼠中鍵` , `3 = 右鍵`

3. `pattern_id` : 要比對的樣式 , 從樣式列表進行選擇和設定

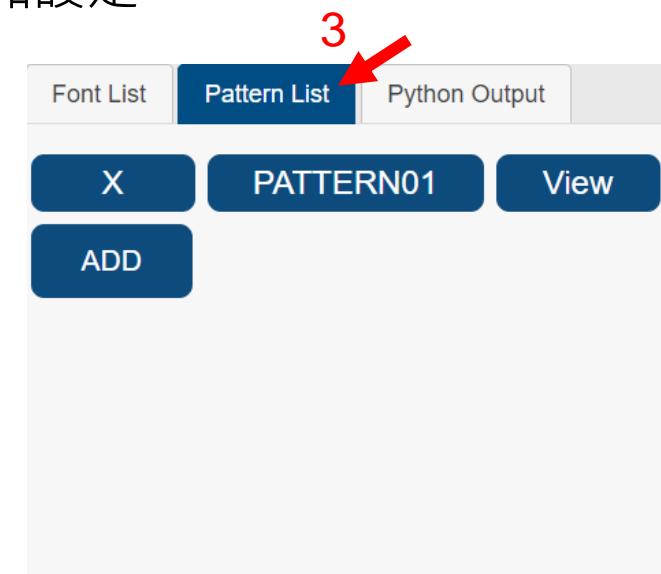
4. `roi_x` : 辨識 ROI 的 x 座標

5. `roi_y` : 辨識 ROI 的 y 座標

6. `roi_width` : 辨識 ROI 的寬度

7. `roi_height` : 辨識 ROI 的高度

8. `score` : 最小比對分數



- **控制函式**

- **PATTERN\_CLICK** : 進階使用

- 1. control\_type : 要使用的控制方式 , 0 = USB , 1 = 本機

- 2. click\_type : 點擊按鍵類型

- 1 = 左鍵 , 2 = 滑鼠中鍵 , 3 = 右鍵

- 3. pattern\_id : 要比對的樣式 , 從樣式列表進行選擇和設定

- 4. roi\_x : 辨識 ROI 的 x 座標

- 5. roi\_y : 辨識 ROI 的 y 座標

- 6. roi\_width : 辨識 ROI 的寬度

- 7. roi\_height : 辨識 ROI 的高度

- 8. score : 最小比對分數

- 9. img : 要辨識的圖片

- 10. params : 樣式比對參數

params = {'gray\_match':True/False,  
'image\_level':a, 'method': b, 'min\_score':0~1.0}

gray\_match : 是否使用灰階比對

image\_level : 圖像層數 , a = 0~5

min\_score : 比對最小分數

method : 比對方法 , b = 3 or 5

3 = 互相關 , 5 = 相關係數

- **控制函式**

- **PATTERN\_DB\_CLICK**：控制滑鼠在符合比對條件的樣式的位置連點兩次，進階使用同

### PATTERN\_CLICK

1. `control_type`：要使用的控制方式

0 = USB · 1 = 本機

2. `click_type`：點擊按鍵類型

1 = 左鍵 · 2 = 滑鼠中鍵 · 3 = 右鍵

3. `pattern_id`：要比對的樣式，從樣式列表進行選擇和設定

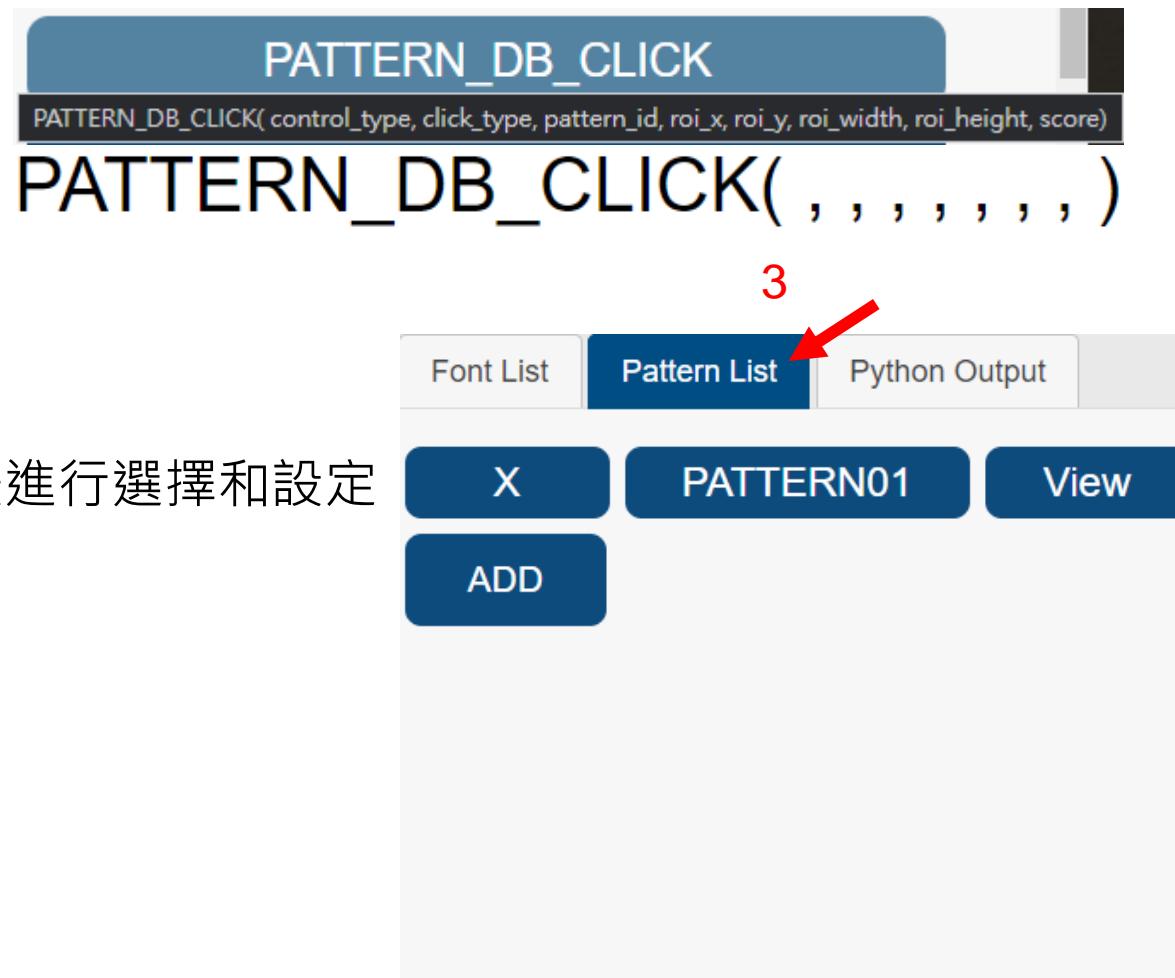
4. `roi_x`：辨識 ROI 的 x 座標

5. `roi_y`：辨識 ROI 的 y 座標

6. `roi_width`：辨識 ROI 的寬度

7. `roi_height`：辨識 ROI 的高度

8. `score`：最小比對分數



- **控制函式**

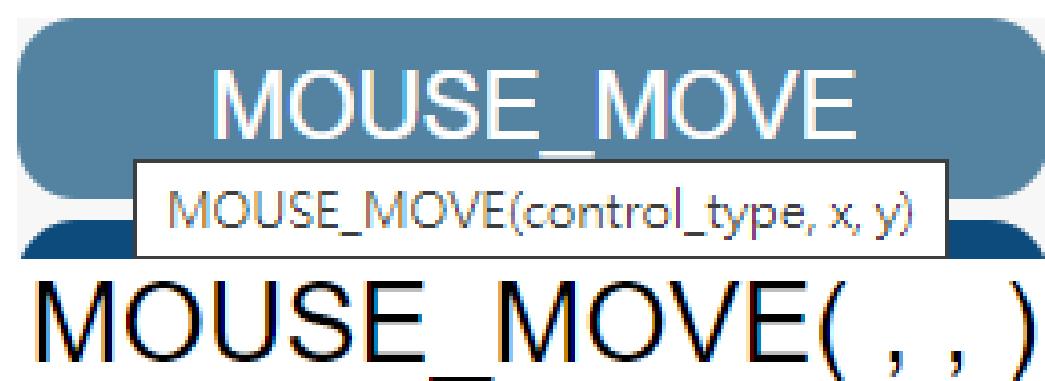
- **MOUSE\_MOVE** : 控制滑鼠游標移動到特定位置

1. `control_type` : 要使用的控制方式

0 = USB · 1 = 本機

2. `x` : 要移動到的 `x` 座標

3. `y` : 要移動到的 `y` 座標



- **控制函式**

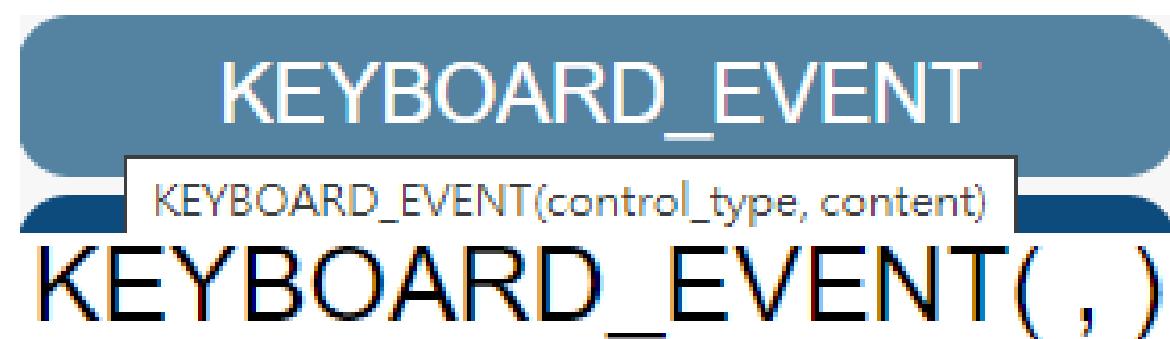
- **KEYBOARD\_EVENT** : 控制鍵盤按下設定的字串內容

1. **control\_type** : 要使用的控制方式

0 = USB · 1 = 本機 · 2 = 序列埠(PS/2) · 3 = 瀏覽器

2. **content** : 要按下的字串內容

實際輸入內容會以受控裝置的設定為主



- **控制函式**

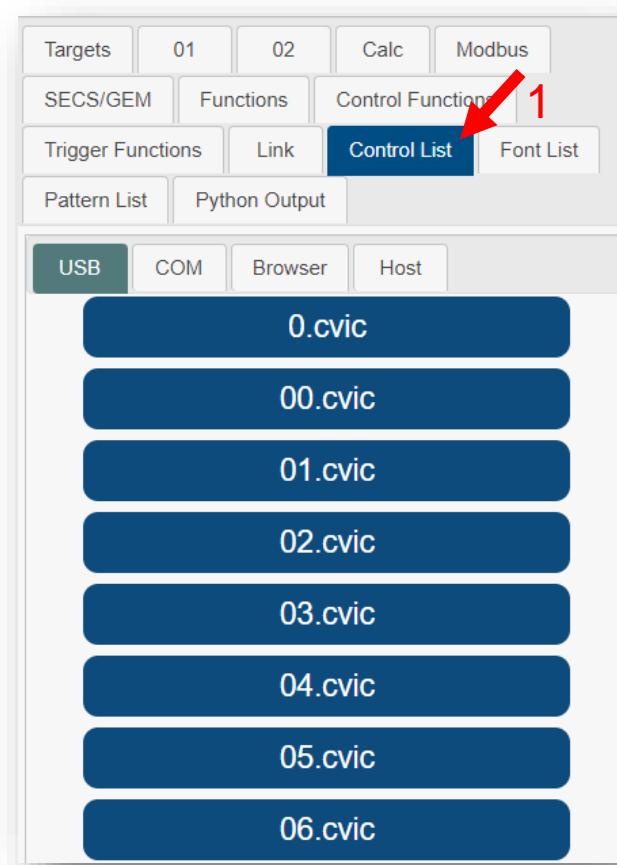
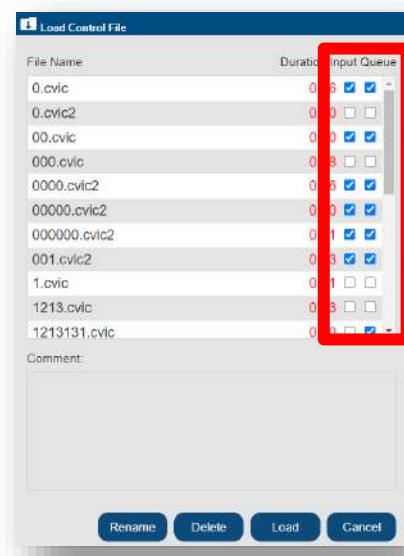
- **PLAY.CONTROL\_FILE\_IQ** : 套用佇列和輸入屬性，執行特定控制檔  
須先到載入控制檔視窗內設定，\* 代表可不設定，含進階設定

1. `control_file_name` : 設定要執行的控制檔，可從控制檔列表選擇
2. `sec` : 執行間隔時間
3. `*offset_x` : 控制檔執行時的 x 軸偏移量，預設為 0
4. `*offset_y` : 控制檔執行時的 y 軸偏移量，預設為 0
5. `*sync` : 是否同步執行，預設為 `False`

**PLAY.CONTROL\_FILE\_IQ**

`PLAY.CONTROL_FILE_IQ(control_file_name, sec, offset_x=0, offset_y=0, bSync=False)`

**PLAY.CONTROL\_FILE\_IQ( , )**



- **控制函式**

- **BROWSER\_RELOAD** : 取像來源為來自瀏覽器時，重新載入瀏覽器



- **控制函式**

- **CONTROL\_DISABLED** : 停用控制功能

- 1. **value** : 設定值

- 1 = 停用 · 0 = 不停用



CONTROL\_DISABLED

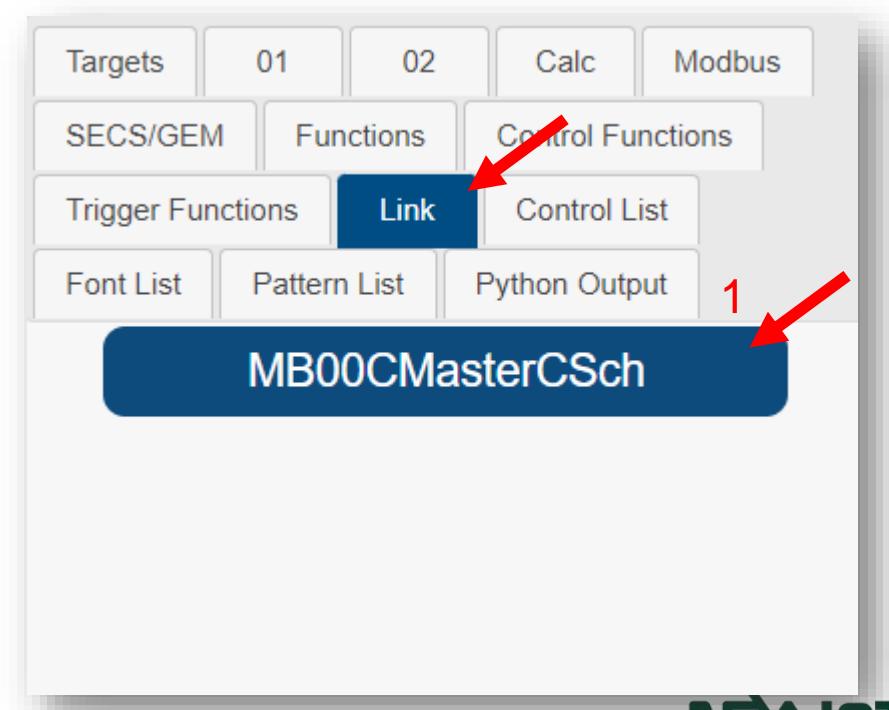
CONTROL\_DISABLED(value)

CONTROL\_DISABLED( )

- **Modbus函式**

- **MODBUS\_MASTER\_READ\_BIT** : 從 Modbus slave/server 讀取位元資料
  - 1. id : 要使用的 Modbus 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
  - 2. prefix : Modbus 表格代碼，0 = Coil Status，1 = Input Status
  - 3. regs : 讀取位址，1-based
  - 4. sec : 執行間隔時間

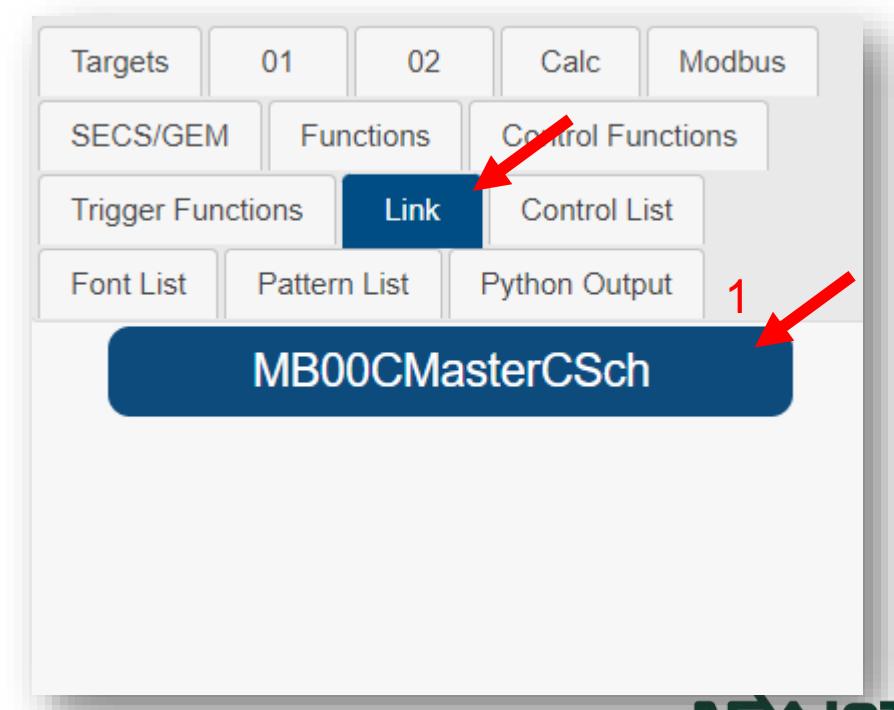
**MODBUS\_MASTER\_READ\_BIT**  
MODBUS\_MASTER\_READ\_BIT(id, prefix, regs, sec)  
MODBUS\_MASTER\_READ\_BIT(, , , )



- **Modbus函式**

- **MODBUS\_MASTER\_WRITE\_BIT** : 寫入位元資料到 Modbus slave/server
  - 1. id : 要使用的 Modbus 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
  - 2. prefix : Modbus 表格代碼，0 = Coil Status
  - 3. regs : 寫入位址，1-based
  - 4. value : 要寫入的位元資料
  - 5. sec : 執行間隔時間

**MODBUS\_MASTER\_WRITE\_BIT**  
MODBUS\_MASTER\_WRITE\_BIT(id, prefix, regs, value, sec)  
MODBUS\_MASTER\_WRITE\_BIT( , , , , )



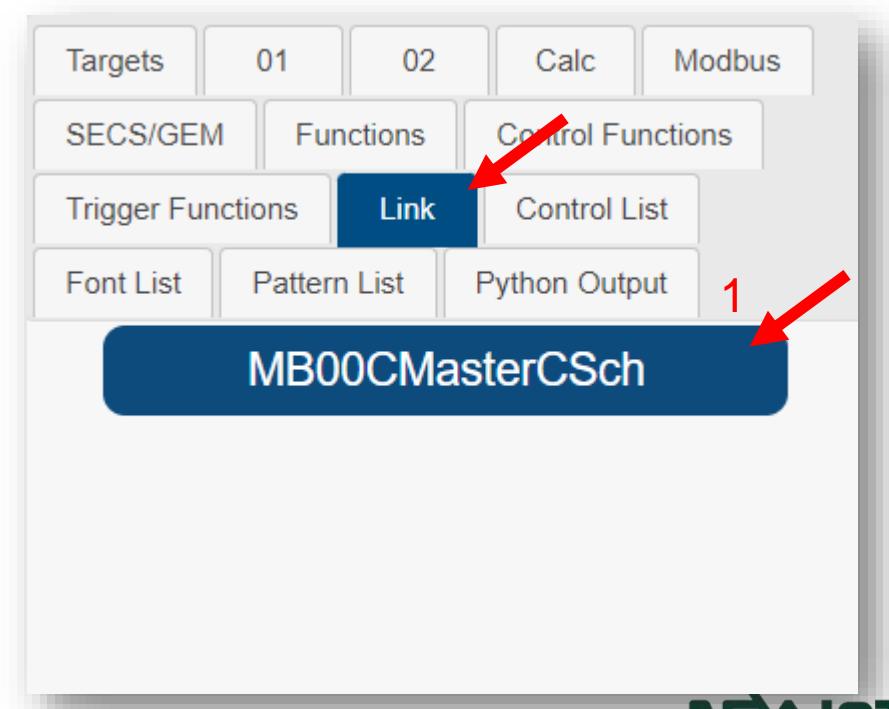
- **Modbus函式**

- **MODBUS\_MASTER\_READ\_STR**：從 Modbus slave/server 讀取字串資料
  - 1. id：要使用的 Modbus 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
  - 2. prefix：Modbus 表格代碼，3 = Input Register，4 = Holding Register
  - 3. regs：讀取位址，1-based
  - 4. count：讀取位址數量
  - 5. sec：執行間隔時間

**MODBUS\_MASTER\_READ\_STR**

```
MODBUS_MASTER_READ_STR(id, prefix, regs, regs_count, sec)
```

**MODBUS\_MASTER\_READ\_STR( , , , , )**

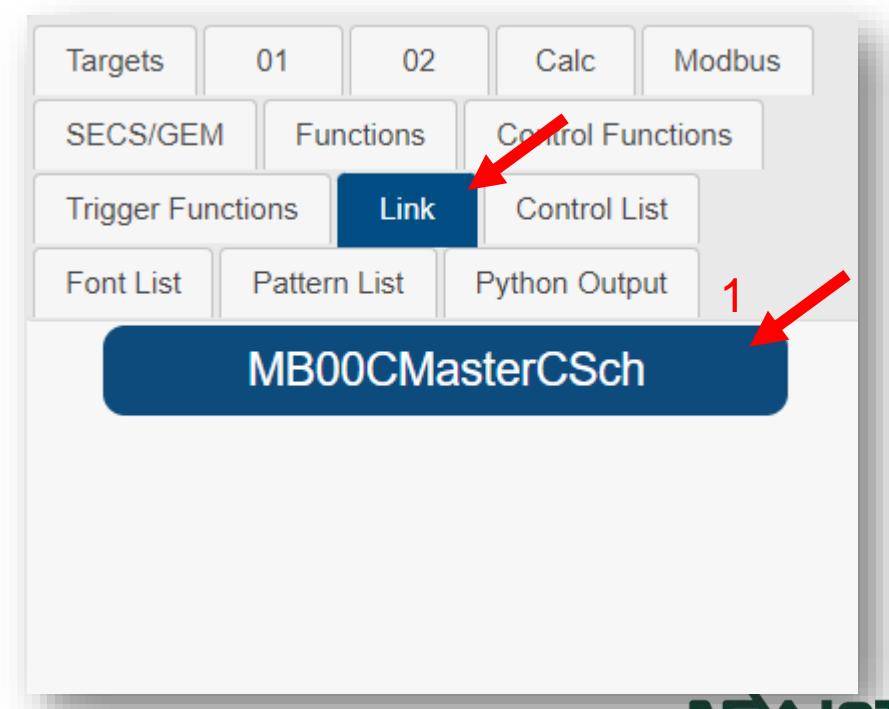


- **Modbus函式**

- **MODBUS\_MASTER\_WRITE\_STR**：寫入字串資料到 Modbus slave/server

1. id：要使用的 Modbus 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
2. prefix：Modbus 表格代碼，4 = Holding Register
3. regs：寫入位址，1-based
4. value：要寫入的字串資料
5. sec：執行間隔時間

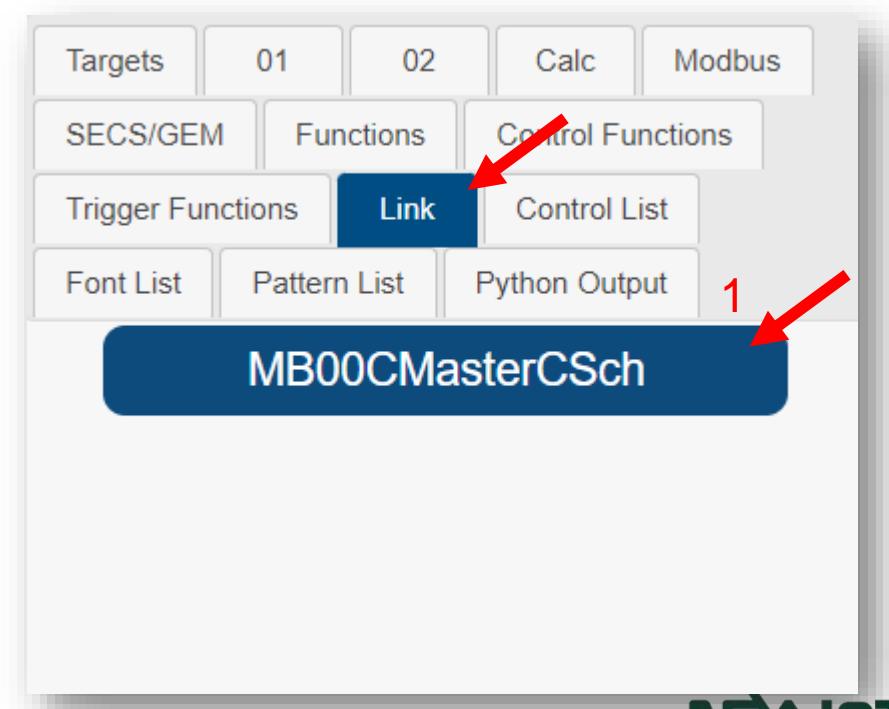
**MODBUS\_MASTER\_WRITE\_STR**  
MODBUS\_MASTER\_WRITE\_STR(id, prefix, regs, value, sec)  
MODBUS\_MASTER\_WRITE\_STR( , , , , )



- **Modbus函式**

- **MODBUS\_MASTER\_READ\_UINT**：從 Modbus slave/server 讀取整數資料
  - 1. id：要使用的 Modbus 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
  - 2. prefix：Modbus 表格代碼，3 = Input Register，4 = Holding Register
  - 3. regs：讀取位址，1-based
  - 4. count：讀取位址數量
  - 5. sec：執行間隔時間

**MODBUS\_MASTER\_READ\_UINT**  
MODBUS\_MASTER\_READ\_UINT(id, prefix, regs, regs\_count, sec)  
MODBUS\_MASTER\_READ\_UINT( , , , , )

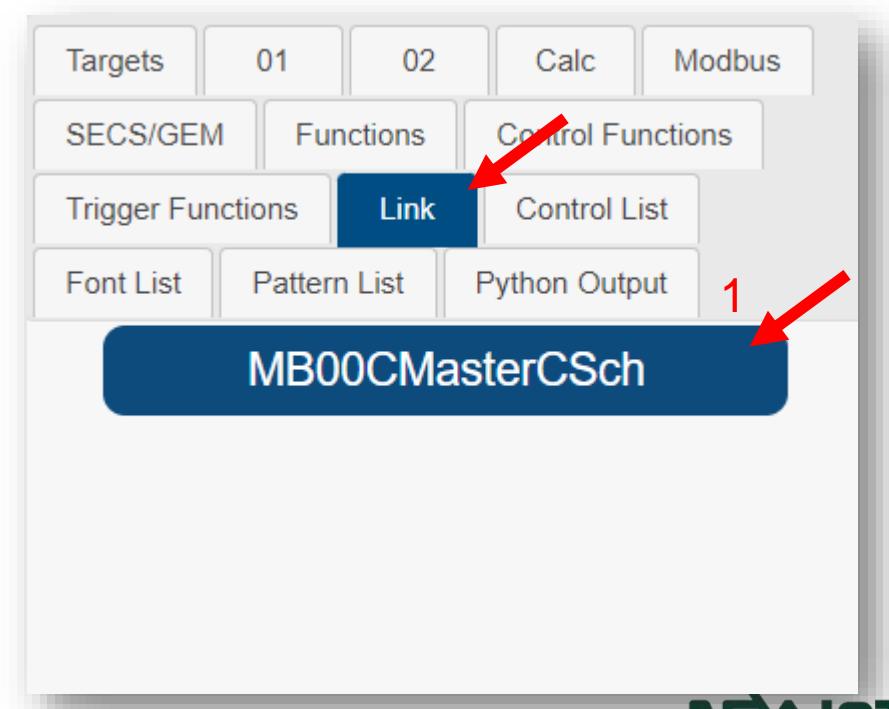


- **Modbus函式**

- **MODBUS\_MASTER\_WRITE\_UINT**：寫入整數資料到 Modbus slave/server

1. id：要使用的 Modbus 連線，要先在連結頁面內建立連線，可從連結中選擇要使用的連線
2. prefix：Modbus 表格代碼，4 = Holding Register
3. regs：寫入位址，1-based
4. value：要寫入的整數資料
5. sec：執行間隔時間

**MODBUS\_MASTER\_WRITE\_UINT**  
MODBUS\_MASTER\_WRITE\_UINT(id, prefix, regs, value, sec)  
MODBUS\_MASTER\_WRITE\_UINT( , , , , )

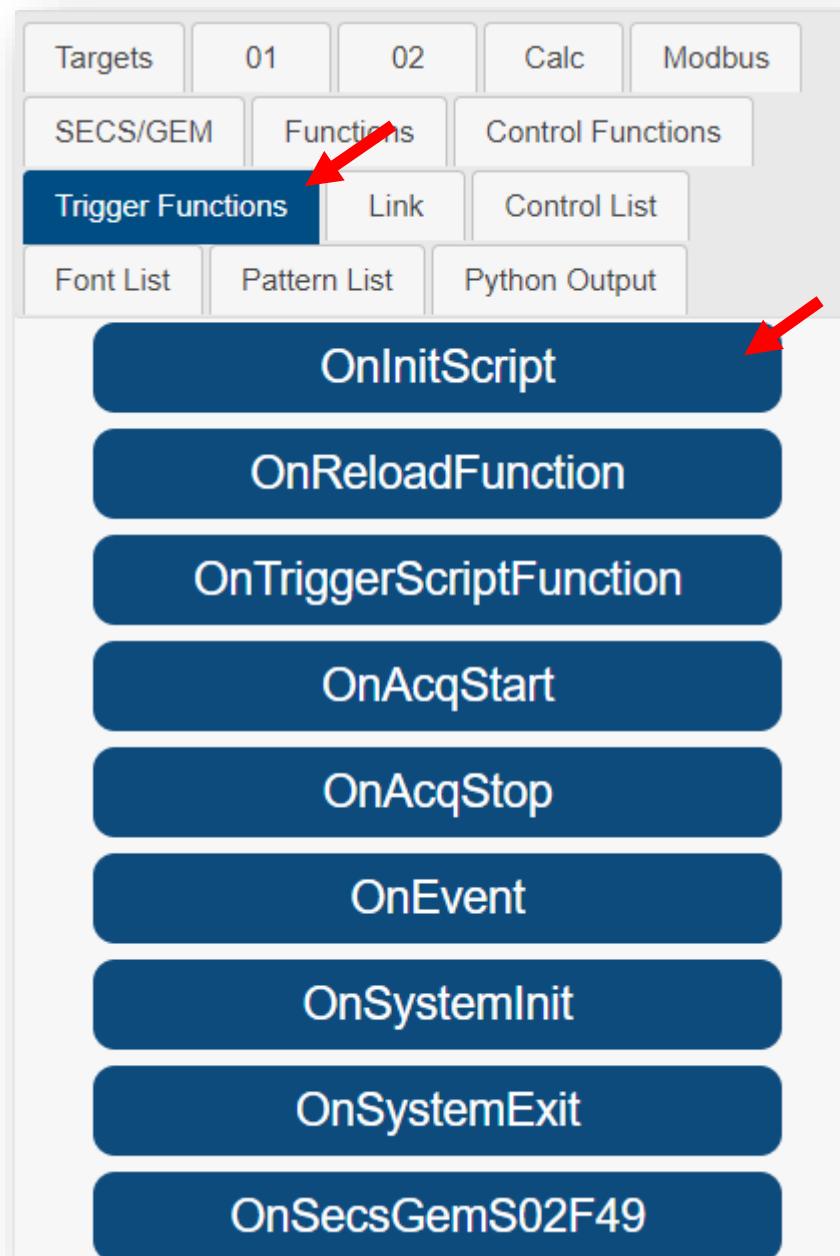


- 觸發函式

- **OnInitScript**：腳本初始化時會觸發的函式

載入專案時腳本會進行初始化

```
def OnInitScript():
    pass
```



- 觸發函式

- **OnReloadFunction**：重新載入工具插件時會觸發的函式

```
def OnReloadFunction():
    pass
```



- 觸發函式

- OnTriggerScriptFunction：透過Modbus TCP/RESTful API觸發的函式

address 代表觸發的Modbus位址和RESTful API的index\_value

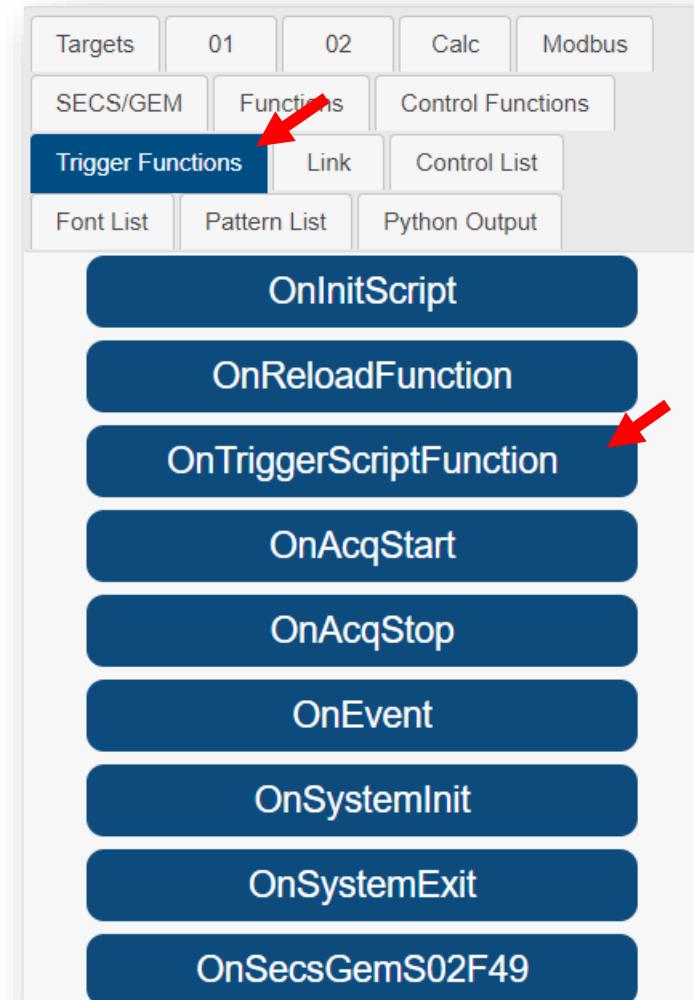
Modbus 觸發位址為 00401 ~ 00500

RESTful 發送的 index\_value 範圍為 400 ~ 499

觸發時 address 會變為函式中的變數

此函式在管理者登入時無效

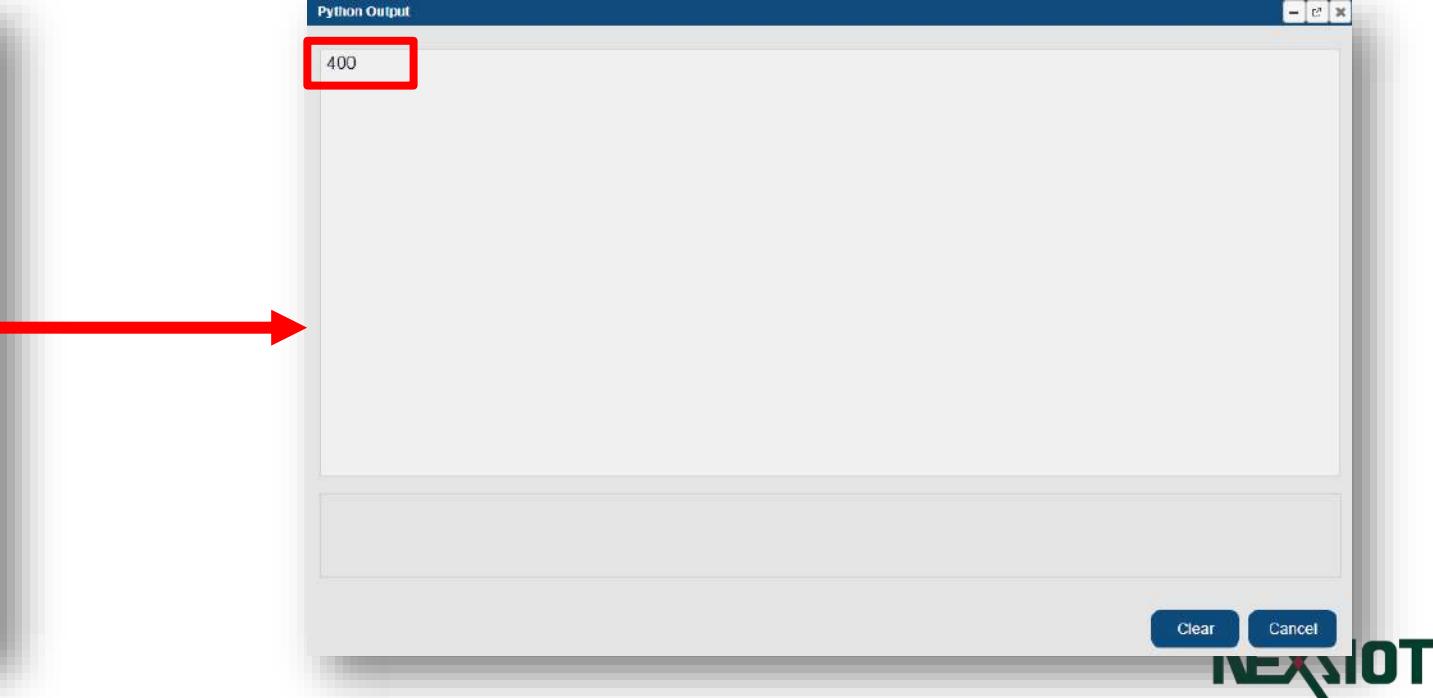
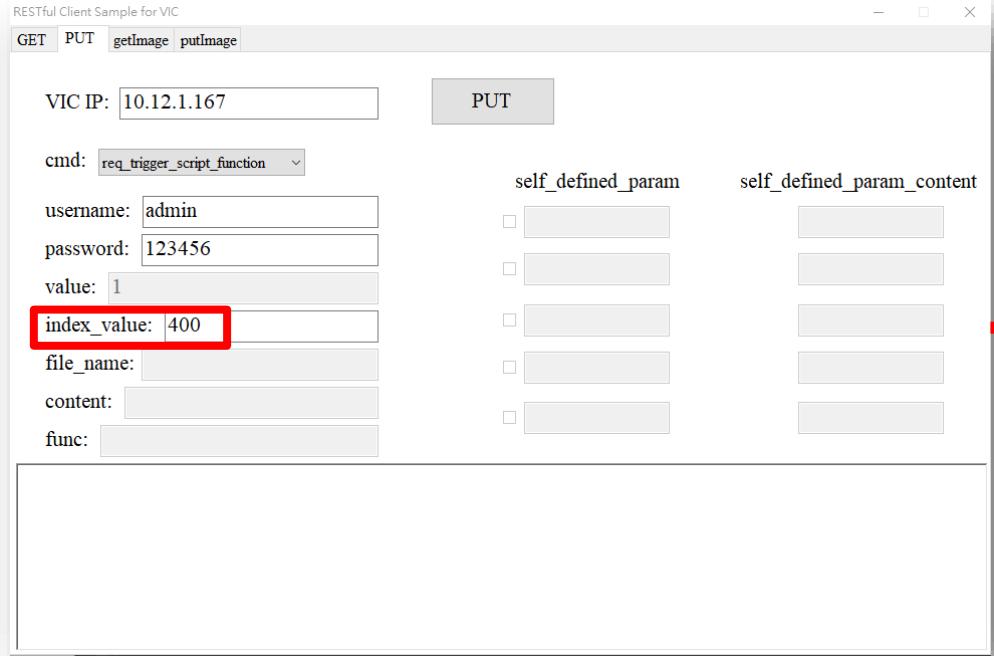
```
def OnTriggerScriptFunction(address):  
    pass
```



- 觸發函式

- OnTriggerScriptFunction

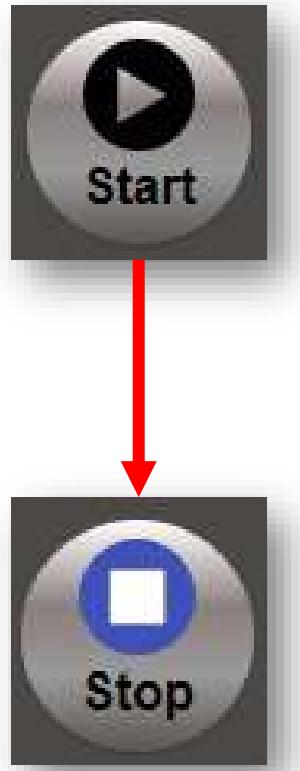
```
def OnTriggerScriptFunction(address):  
    print(str(address))  
    pass
```



- 觸發函式

- OnAcqStart : 開始擷取影像時觸發的函式

```
def OnAcqStart(cause):  
    pass
```

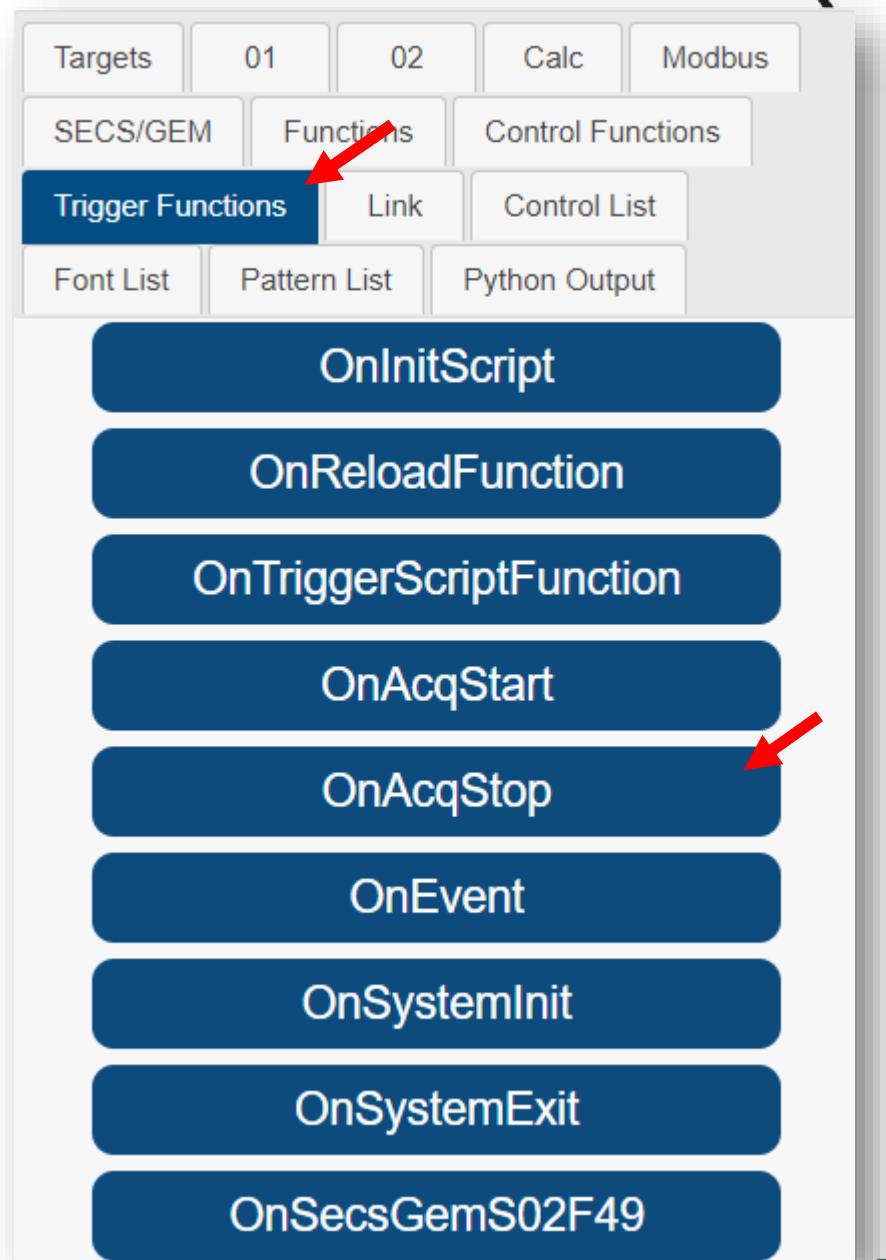
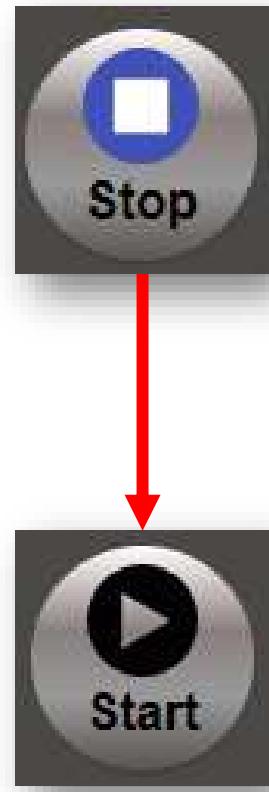


Targets	01	02	Calc	Modbus
SECS/GEM	Functions	Control Functions		
	Trigger Functions	Link	Control List	
Font List	Pattern List	Python Output		
<b>OnInitScript</b>				
<b>OnReloadFunction</b>				
<b>OnTriggerScriptFunction</b>				
<b>OnAcqStart</b>				
<b>OnAcqStop</b>				
<b>OnEvent</b>				
<b>OnSystemInit</b>				
<b>OnSystemExit</b>				
<b>OnSecsGemS02F49</b>				

- 觸發函式

- OnAcqStop : 停止擷取影像時觸發的函式

```
def OnAcqStop():
    pass
```



- 觸發函式

- OnEvent : 事件發生時觸發的函式

params 內有事件發生的敘述，由 params['eventType'] 和 params['signal'] 取得事件內容

params['eventType'] = vic\_event\_capture\_card\_removed : 拔除擷取卡

vic\_event\_capture\_card\_detected : 偵測到擷取卡

vic\_event\_signal\_removed : 無螢幕訊號輸入

vic\_event\_format\_changed : 訊號改變

params['signal'] = True/False : 有無影像訊號輸入

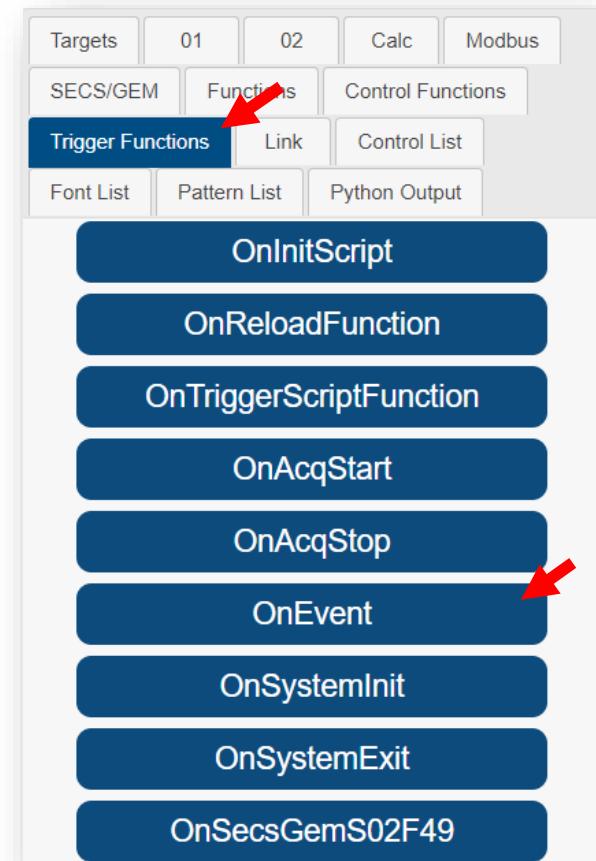
(僅在eventType為vic\_event\_format\_changed有效)

```
def OnEvent(params):
```

```
    print(params['eventType'])
    print(params['signal'])
```

```
    pass
```

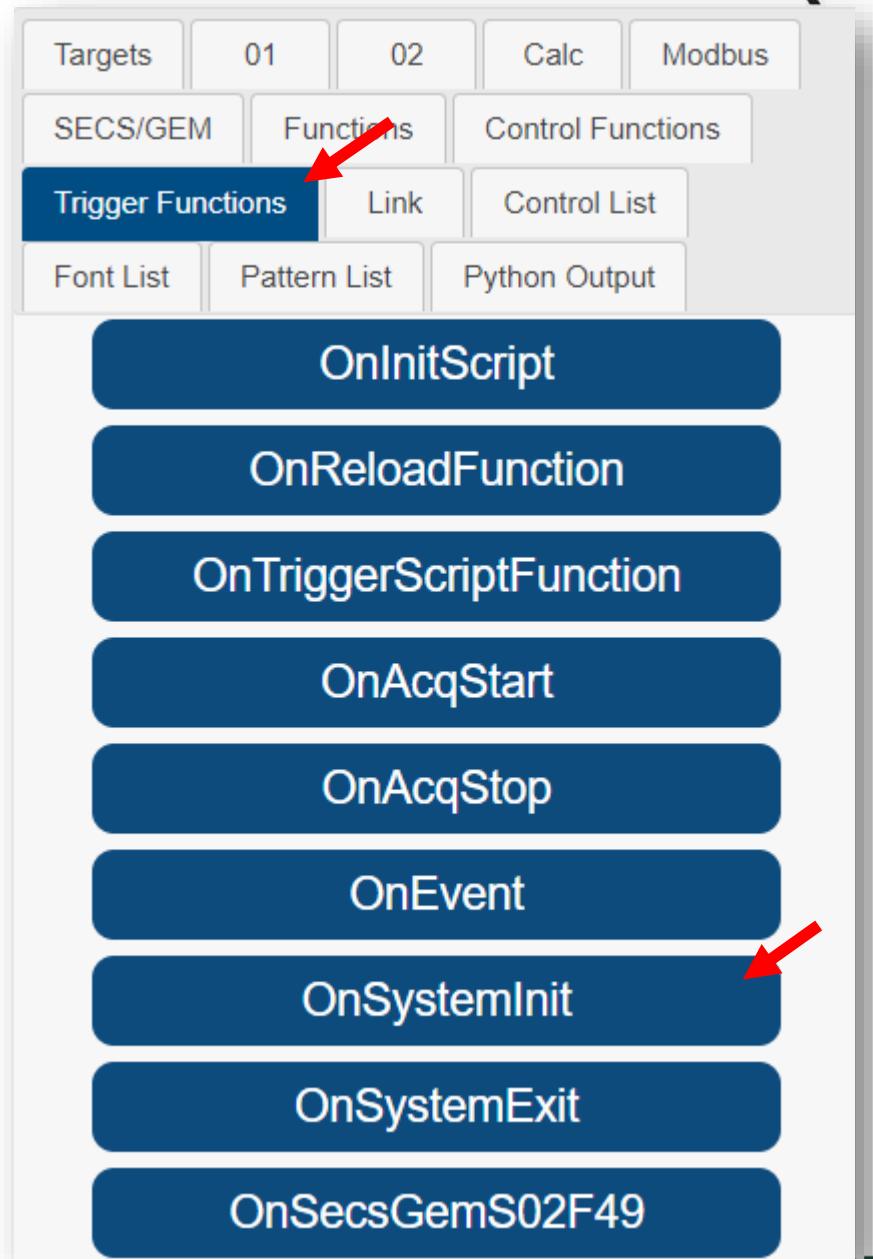
```
def OnEvent(params):
    pass
```



- 觸發函式

- **OnSystemInit** : 系統啟動且初始化完成時觸發的函式

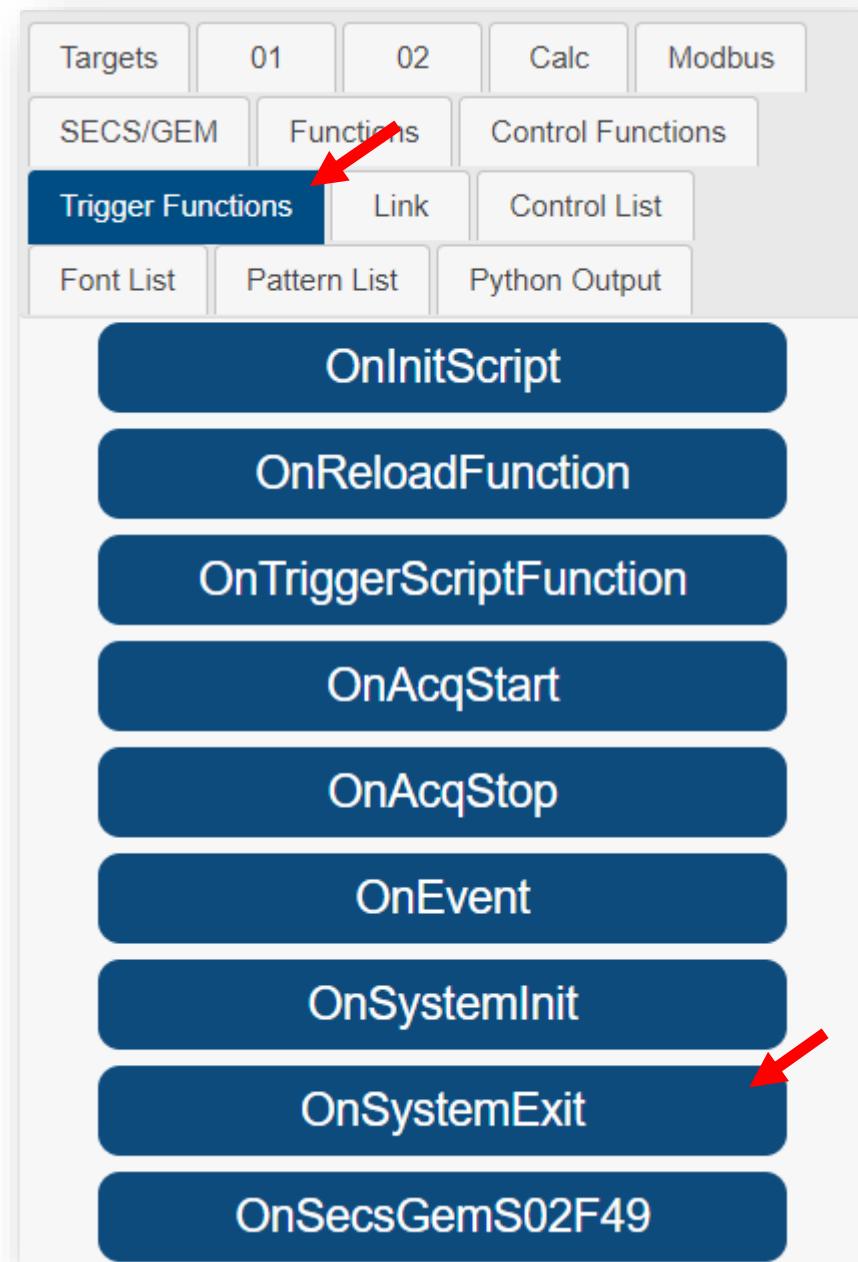
```
def OnSystemInit():
    pass
```



- 觸發函式

- **OnSystemExit**：系統被關閉時執行的函式

```
def OnSystemExit():
    pass
```



- 觸發函式

- OnSecsGemS02F49：系統被透過SECS/GEM接收到 S02F49 時會觸發的函式  
parms 代表接收到的 S02F49 傳送的參數內容

```
def OnSecsGemS02F49(**parms):  
    pass
```



- 呼叫自定義函式

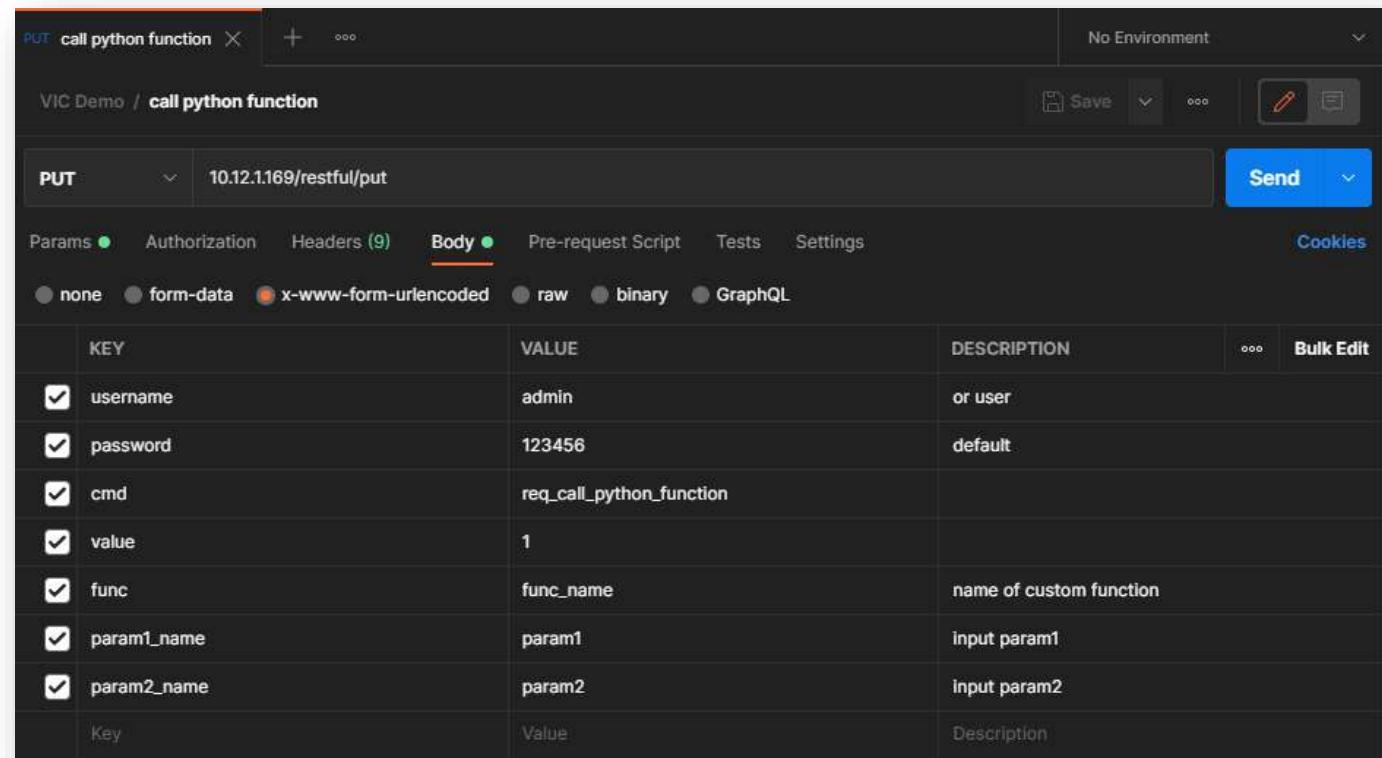
- 自定義函式設置

1. func\_name : 自定義函式名稱
2. kwargs : 輸入參數引數
3. param1\_name : 輸入參數1名稱
4. param2\_name : 輸入參數2名稱
5. return1\_name : 回傳參數1名稱
6. return2\_name : 回傳參數2名稱

```
def demo_call_python_function(**kwargs):  
    param1=kwargs['param1_name']  
    param2=kwargs['param2_name']  
      
      
    return1=int(param1)+int(param2)  
    return2=int(param1)*int(param2)  
      
      
    return{'return1_name':return1,'return2_name':return2}
```

- 呼叫自定義函式
  - 透過 RESTful API 呼叫 VIC 執行自定義函式

1. username : admin or user
2. password : 123456(default)
3. cmd : req\_call\_python\_function
4. value : 1
5. func : func\_name
6. param1\_name : param1
7. param2\_name : param1

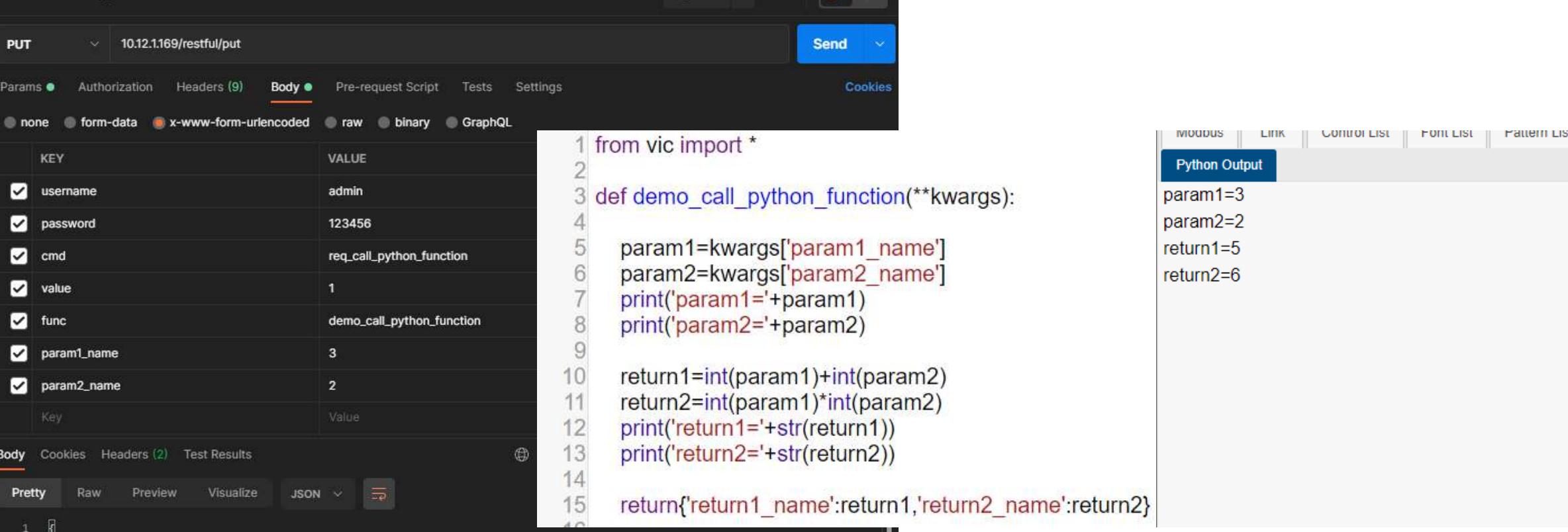


The screenshot shows the Postman interface for a 'PUT' request to 'call python function'. The 'Body' tab is selected, showing the following parameters:

KEY	VALUE	DESCRIPTION
username	admin	or user
password	123456	default
cmd	req_call_python_function	
value	1	
func	func_name	name of custom function
param1_name	param1	input param1
param2_name	param2	input param2

## • 呼叫自定義函式

範例：透過 RESTful API 發出兩個參數，透過 VIC 計算後回傳計算值



PUT call python function

VIC Demo / call python function

PUT 10.12.1.169/restful/put

Body

KEY	VALUE
username	admin
password	123456
cmd	req_call_python_function
value	1
func	demo_call_python_function
param1_name	3
param2_name	2

1 from vic import \*  
2  
3 def demo\_call\_python\_function(\*\*kwargs):  
4  
5 param1=kwargs['param1\_name']  
6 param2=kwargs['param2\_name']  
7 print('param1='+param1)  
8 print('param2='+param2)  
9  
10 return1=int(param1)+int(param2)  
11 return2=int(param1)\*int(param2)  
12 print('return1='+str(return1))  
13 print('return2='+str(return2))  
14  
15 return{'return1\_name':return1,'return2\_name':return2}

param1=3  
param2=2  
return1=5  
return2=6

- 呼叫自定義函式

- 透過 SECS/GEM 呼叫 VIC 執行自定義函式

1. RCMD : 要執行的函式名稱

- 2-1. CPNAME : 參數名稱

- 2-2. CPVAL : 參數值

```
s2f41: S2F41 W /* Host Command Send */
<L [2]
  <A 'RCMD'> /* <RCMD> Remote command code */
  <L [4]
    <L [2]
      <A 'CPNAME1'> /* <CPNAME1> Command Parameter Name */
      <A 'CPVAL1'> /* <CPVAL1> Command Parameter Value */
    >
    <L [2]
      <A 'CPNAME2'>
      <A 'CPVAL2'>
    >
    <L [2]
      <A 'CPNAME3'>
      <A 'CPVAL3'>
    >
    <L [2]
      <A 'CPNAME4'>
      <A 'CPVAL4'>
    >
  >
>.
```

- 呼叫自定義函式

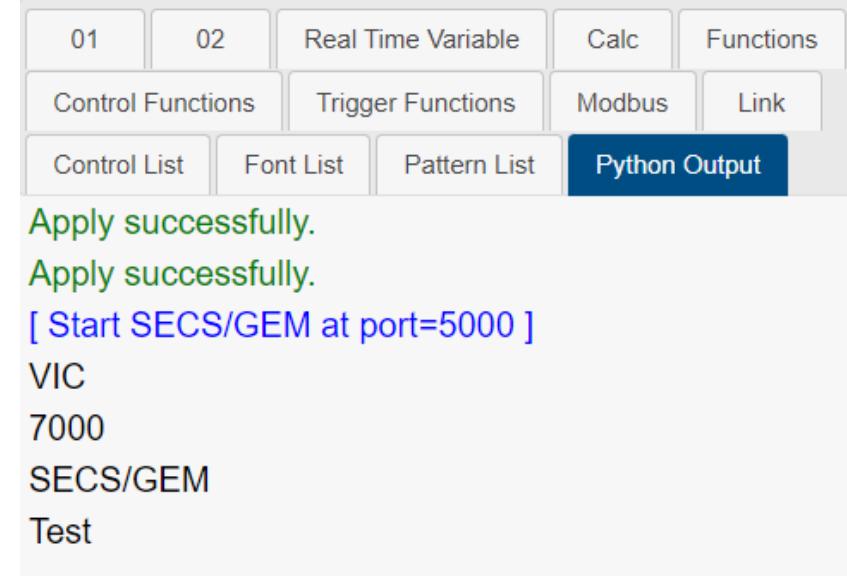
範例：透過 SECS/GEM 發出四個參數，在 VIC 上顯示出來

```
s2f41: S2F41 W /* Host Command Send */
<L [2]
  <A 'vic_secs_gem_demo'> /* <RCMD> Remote command code */
  <L [4]
    <L [2]
      <A 'CPNAME1'> /* <CPNAME1> Command Parameter Name */
      <A 'VIC'> /* <CPVAL1> Command Parameter Value */
    >
    <L [2]
      <A 'CPNAME2'>
      <A '7000'>
    >
    <L [2]
      <A 'CPNAME3'>
      <A 'SECS/GEM'>
    >
    <L [2]
      <A 'CPNAME4'>
      <A 'Test'>
    >
  >.
```

def vic\_secs\_gem\_demo(\*\*args):

```
param1 = args['CPNAME1']
param2 = args['CPNAME2']
param3 = args['CPNAME3']
param4 = args['CPNAME4']
print(param1)
print(param2)
print(param3)
print(param4)
```

```
14:32:40.319: Receive S2F42 (18742110).
S2F42
<L[2]
  <B[1] 0x00>
  <L[0]
>
>
.
```



- 呼叫自定義函式

範例：透過 SECS/GEM 發出四個參數，在 VIC 上顯示出來，並透過 S6F11 回傳資訊

```
s2f41: S2F41 W /* Host Command Send */
<L [2]
<A 'vic_secs_gem_demo'> /* <RCMD> Remote command code */
<L [4]
<L [2]
<A 'CPNAME1'> /* <CPNAME1> Command Parameter Name */
<A 'VIC'> /* <CPVAL1> Command Parameter Value */
>
<L [2]
<A 'CPNAME2'>
<A '7000'>
>
<L [2]
<A 'CPNAME3'>
<A 'SECS/GEM'>
>
<L [2]
<A 'CPNAME4'>
<A 'Test'>
>
>.
```

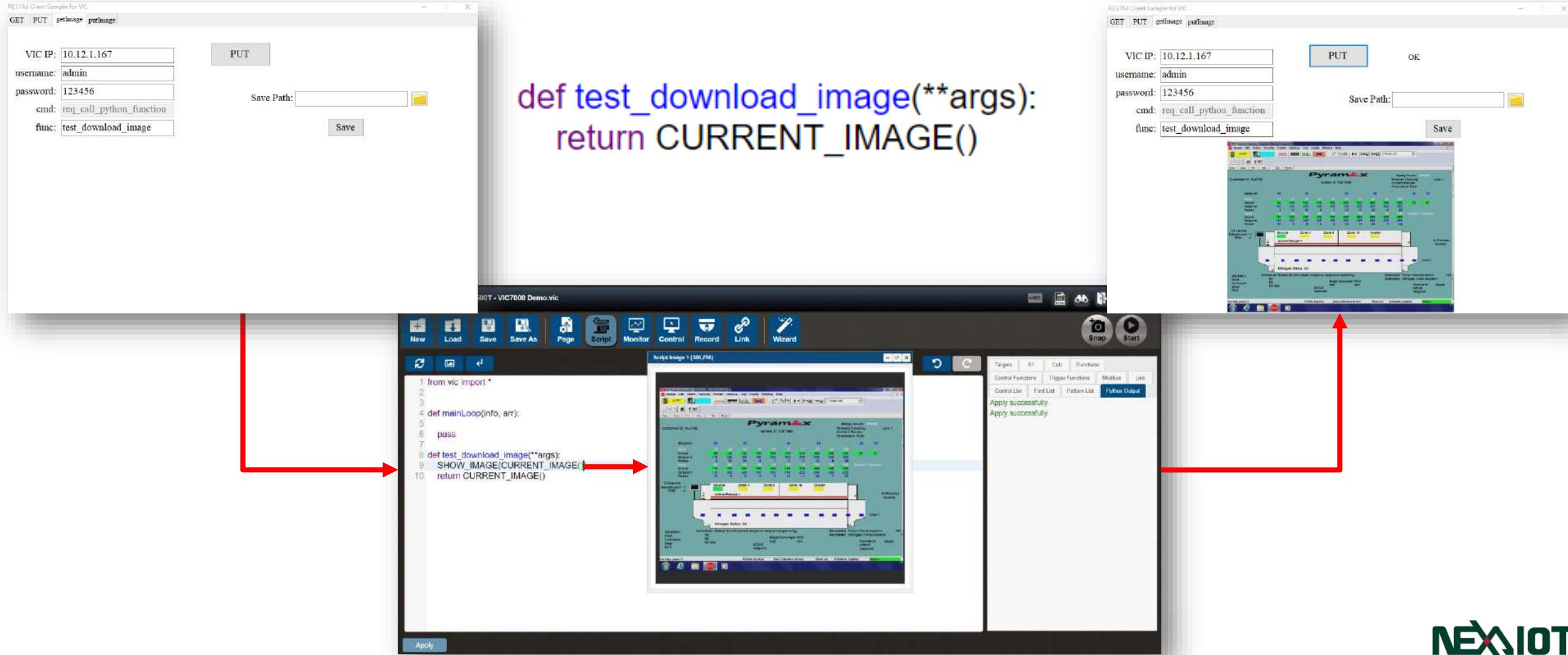
```
def vic_secs_gem_demo(**args):
    param1 = args['CPNAME1']
    param2 = args['CPNAME2']
    param3 = args['CPNAME3']
    param4 = args['CPNAME4']
    print(param1)
    print(param2)
    print(param3)
    print(param4)
```

```
return {'DATAID':123, 'CEID':456, 'RPTID':5, 'RPT':[{'ABC': 123, '45.6': 45.6, 'True': True}]} NEXIOT
```

```
14:39:20.067: Receive S6F11(1499936893).
S6F11 W
<L [3]
<U4[1] 123>
<U4[1] 456>
<L[1]
<L[2]
<U1[1] 5>
<L[4]
<A[3] "ABC">
<I4[1] 123>
<F8[1] 45.600000>
<Boolean[1] 0x01>
>
>
>
>
.
.
```

## • 傳送圖片

透過呼叫自定義函式，由該函式回傳圖片



# 腳本

nexVIC

## ● 收到圖片

收到透過RESTful API PUT傳送的圖片，再回傳資料

```
def test_upload_image(**args):  
    img = args['arg2']  
    SHOW_IMAGE(img)  
    return{'Return':'OK'}
```

The image shows the nexVIC software interface with several windows and a central data table.

- Top Left Window:** RESTful Client Sample for VIC. It shows a PUT request with parameters: self\_defined\_param (arg1: 456) and self\_defined\_param\_content. The file path is arg2.
- Top Right Window:** RESTful Client Sample for VIC. It shows a PUT request with parameters: self\_defined\_param (arg1: 456) and self\_defined\_param\_content. The file path is arg2. The response is a JSON object: {"Return": "OK", "login\_success": "1"}
- Bottom Left Window:** Script Image 2. It contains a Python script with a function test\_upload\_image that uses SHOW\_IMAGE to display an image and returns {"Return": "OK"}. A red arrow points from this window to the central data table.
- Bottom Right Window:** RESTful Client Sample for VIC. It shows a PUT request with parameters: self\_defined\_param (arg1: 456) and self\_defined\_param\_content. The file path is arg2. The response is a JSON object: {"Return": "OK", "login\_success": "1"}. A red arrow points from this window to the central data table.
- Central Data Table:** A table showing product data with columns: Product ID, Quantity, Temperature, Voltage, Time, and Result. The data is as follows:

Product ID	Quantity	Temperature	Voltage	Time	Result
1	968	-7.47 °C	19.2 V	1936 ms	Pass
2	791	16.43 °C	19.0 V	1584 ms	Pass
3	119	-13.20 °C	-0.5 V	247 ms	NG
4	670	-4.10 °C	14.8 V	1343 ms	Pass
5	539	11.12 °C	12.6 V	1082 ms	Pass
6	177	19.97 °C	7.0 V	362 ms	NG
7	701	13.33 °C	-7.5 V	1404 ms	Pass
8	479	-10.15 °C	8.1 V	963 ms	NG
9	875	16.48 °C	17.3 V	1750 ms	Pass
10	528	-1.33 °C	18.5 V	1062 ms	Pass
11	260	-1.20 °C	17.4 V	528 ms	NG
12	928	-2.48 °C	-9.2 V	1857 ms	Pass

**Bottom Navigation Bar:** Includes buttons for Save As, Page, Script, Monitor, Control, Record, Link, and Wizard.

NEXIOT

- 匯入 py：匯入自行編寫的 py 檔

將自行編寫的 py 檔放置於 VIC 的 py 資料夾中，路徑如下

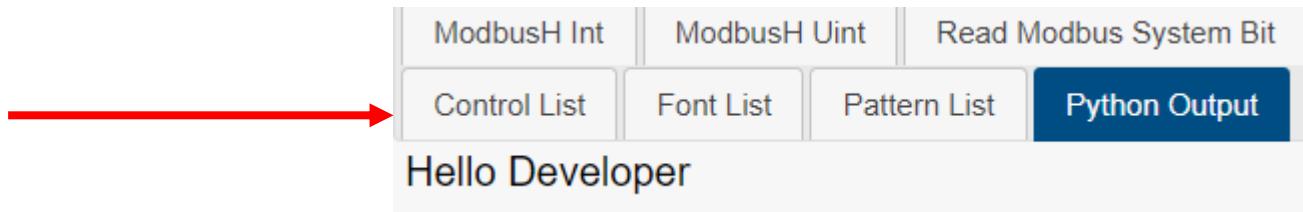
Windows : C:\VIC7000\bin\py

Linux : /opt/VIC7000/bin/py

就可以在 Python 腳本中匯入 py 檔並使用它，如下範例第 3 行

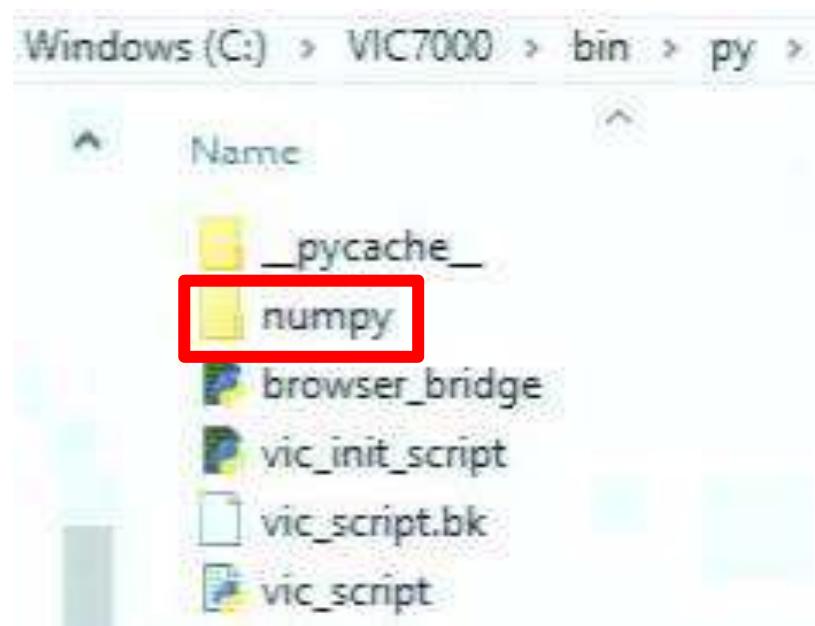
若該 py 檔會被更改，則需要 reload，如下範例第 2、4 行

```
1  def demo_develop():
2      print('Hello Developer')
3
4  from vic import *
5  from imp import reload
6  import Demo_Develop_Import
7  reload(Demo_Develop_Import)
8
9  def mainLoop(info, arr):
10     Demo_Develop_Import.demo_develop()
```



- **匯入套件PYPI**

- **線上安裝**：使用 Windows 的 Command Prompt (Admin) 和 Linux 的 terminal，輸入  
pip install XXXX，XXXX 為套件名稱，即可安裝套件，使用 Python 腳本時要再次 import 該套件
- **離線安裝**：使用另一台可連網並安裝 python 的電腦，使用上述 pip 安裝後，將套件放入  
VIC7000 電腦上的 C:\VIC7000\bin\py 內即可，使用 Python 腳本時要再次 import 該套件



```
from vic import *
import numpy
```

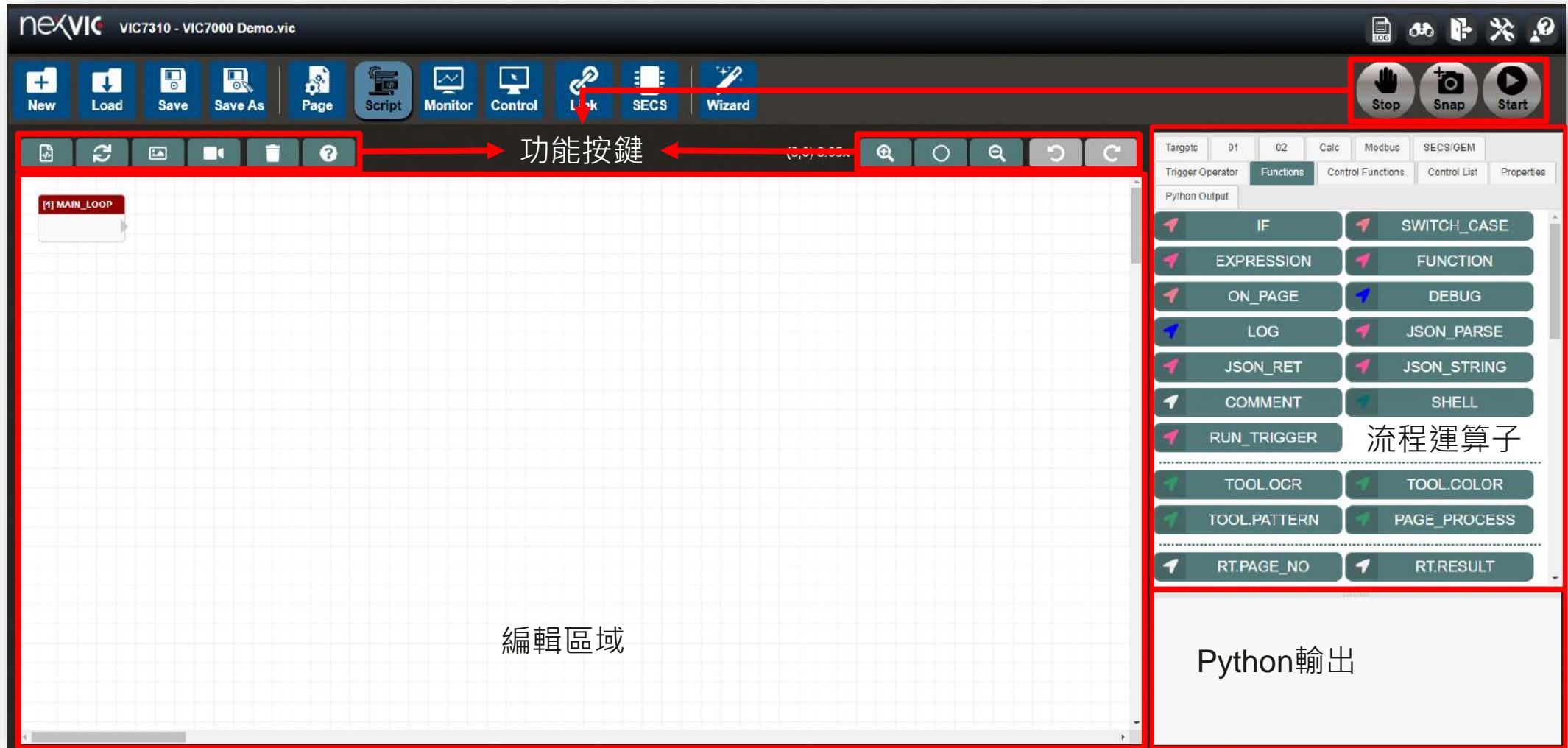
```
def mainLoop(info, arr):
    pass
```

# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- **VIC Flow 操作介面**
- **VIC Flow 表達式&運算子**
- **VIC Flow Modbus**
- **VIC Flow 觸發運算子**
- **VIC Flow 函式**
- **VIC Flow 控制函式**

# VIC Flow 操作介面

- 操作介面



# VIC Flow 操作介面

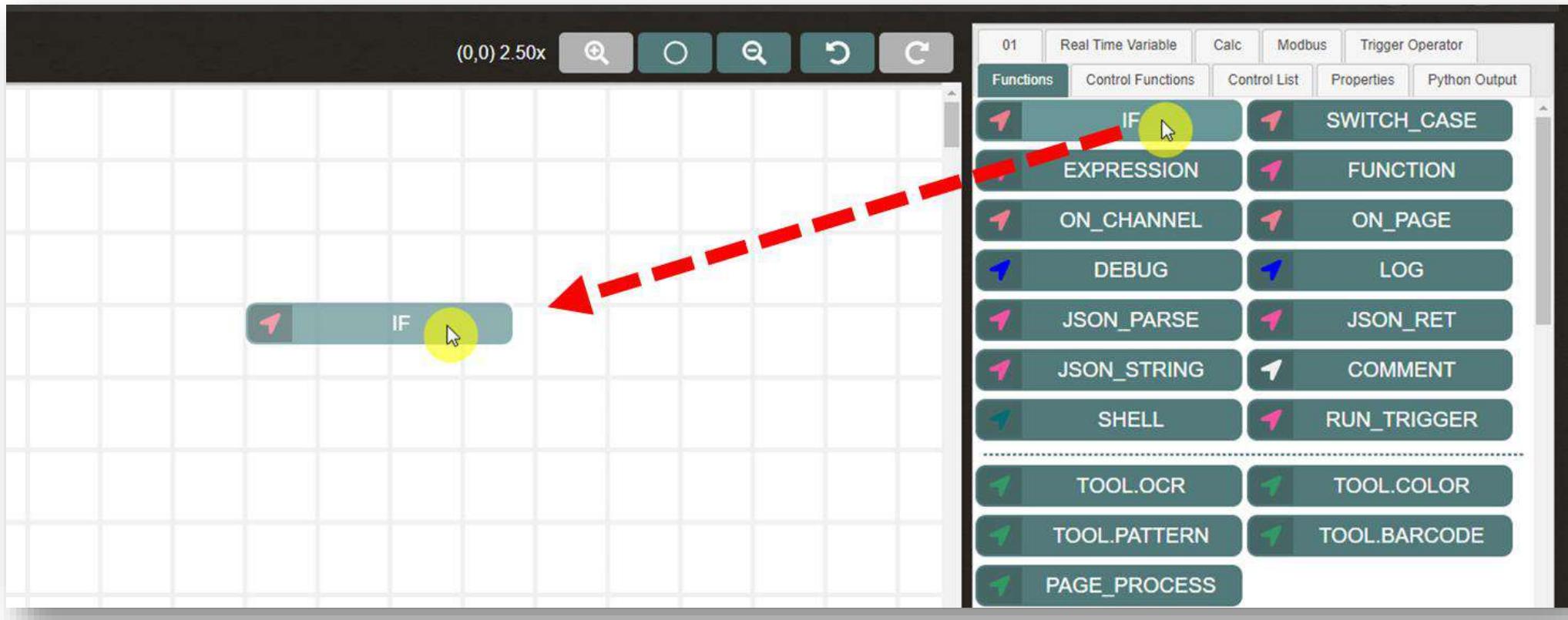
- 功能按鍵

圖示	行為	圖示	行為
	切換至文字模式		放大編輯區域的倍率 最大至 2.50x
	重新載入Python模組		重置編輯區域的放大/縮小倍率
	開啟腳本圖像視窗		縮小編輯區域的倍率 最小至 0.4x
	開啟控制影像視窗		退回 最多記錄 100 個動作
	重置VIC Flow內容		重做 最多記錄 100 個動作
	顯示快捷鍵資訊		編輯區域左上角的X, Y座標 與編輯區域的當前放大/縮小倍率

# VIC Flow 操作介面

- 使用運算子

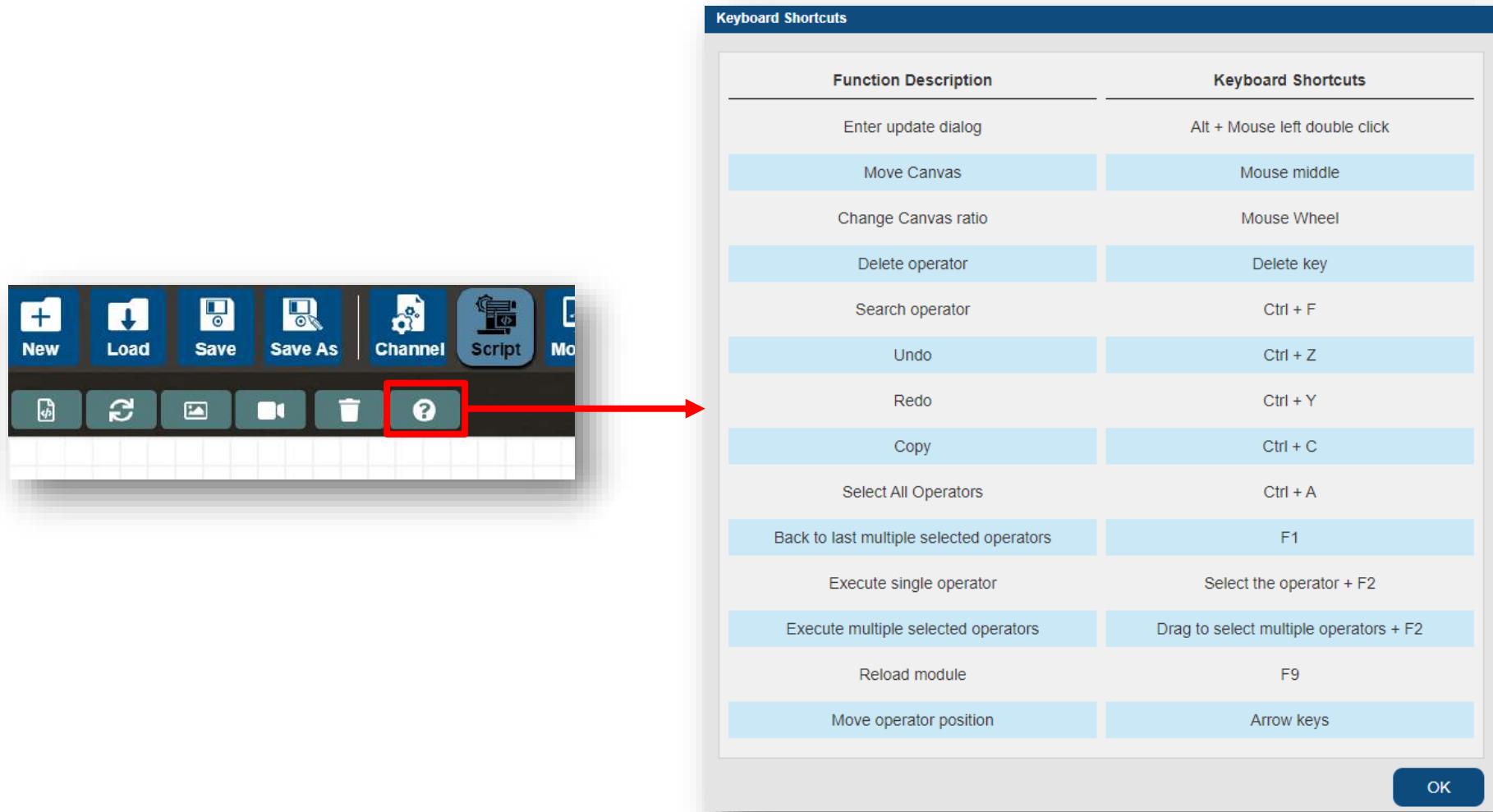
將運算子從列表拖拉至編輯區域中，即可將該運算子加入編輯區域



# VIC Flow 操作介面

## • 快捷鍵

點擊快捷鍵資訊按鈕，可以看到快捷鍵列表的說明



The screenshot shows the VIC Flow interface with a 'Keyboard Shortcuts' dialog box overlaid. The dialog box has a dark blue header with the text 'Keyboard Shortcuts'. It contains a table with two columns: 'Function Description' and 'Keyboard Shortcuts'. The table lists various functions and their corresponding keyboard shortcuts. The 'Function Description' column includes: 'Enter update dialog', 'Move Canvas', 'Change Canvas ratio', 'Delete operator', 'Search operator', 'Undo', 'Redo', 'Copy', 'Select All Operators', 'Back to last multiple selected operators', 'Execute single operator', 'Execute multiple selected operators', 'Reload module', and 'Move operator position'. The 'Keyboard Shortcuts' column includes: 'Alt + Mouse left double click', 'Mouse middle', 'Mouse Wheel', 'Delete key', 'Ctrl + F', 'Ctrl + Z', 'Ctrl + Y', 'Ctrl + C', 'Ctrl + A', 'F1', 'Select the operator + F2', 'Drag to select multiple operators + F2', 'F9', and 'Arrow keys'. The 'OK' button is located at the bottom right of the dialog box. A red arrow points from the question mark icon in the toolbar to the 'Keyboard Shortcuts' dialog box.

Function Description	Keyboard Shortcuts
Enter update dialog	Alt + Mouse left double click
Move Canvas	Mouse middle
Change Canvas ratio	Mouse Wheel
Delete operator	Delete key
Search operator	Ctrl + F
Undo	Ctrl + Z
Redo	Ctrl + Y
Copy	Ctrl + C
Select All Operators	Ctrl + A
Back to last multiple selected operators	F1
Execute single operator	Select the operator + F2
Execute multiple selected operators	Drag to select multiple operators + F2
Reload module	F9
Move operator position	Arrow keys

OK

# VIC Flow 操作介面

- 快捷鍵

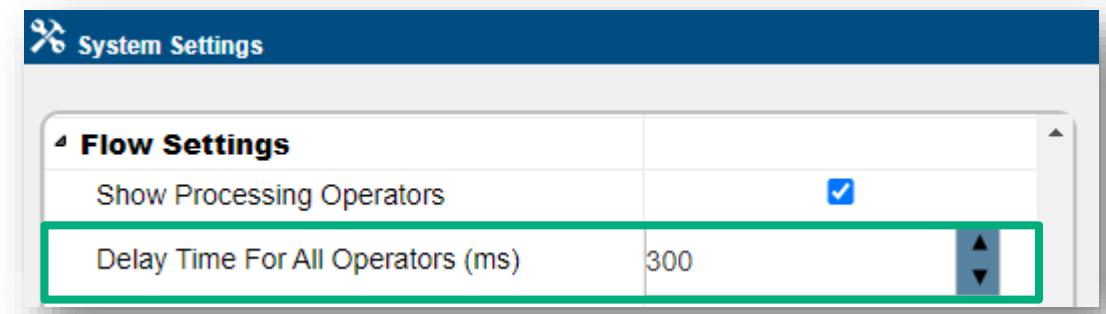
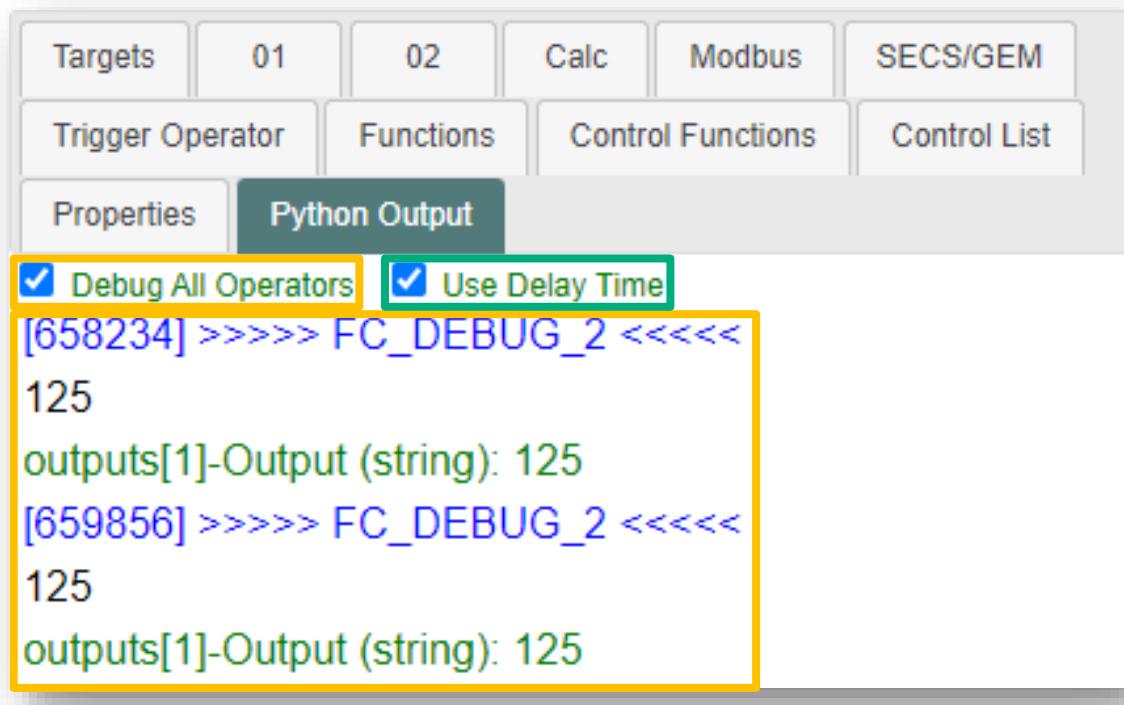
快捷鍵	行為
<b>F1</b>	返回至上一個多選的運算子
<b>F2</b>	選擇單一運算子，執行選擇的運算子
	圈選多個運算子，執行圈選的運算子
<b>F9</b>	重新載入Python模組
<b>Delete</b>	刪除運算子
方向鍵(上、下、左、右)	移動運算子
<b>Ctrl + C</b>	複製出該運算子
<b>Ctrl + F</b>	搜尋運算子
<b>Ctrl + Z</b>	退回
<b>Ctrl + Y</b>	重作
<b>Alt + 滑鼠左鍵雙次點擊運算子</b>	進入該運算子的更新視窗
滑鼠滾輪	改變放大倍率
滑鼠中鍵	移動畫布

# VIC Flow 操作介面

- **Python 輸出**

偵錯全部運算子：勾選後，在 Python 輸出可以看到所有運算子的執行內容

使用延遲時間：勾選後，每個運算子執行後會有延遲時間，延遲時間可在系統設定視窗中設定

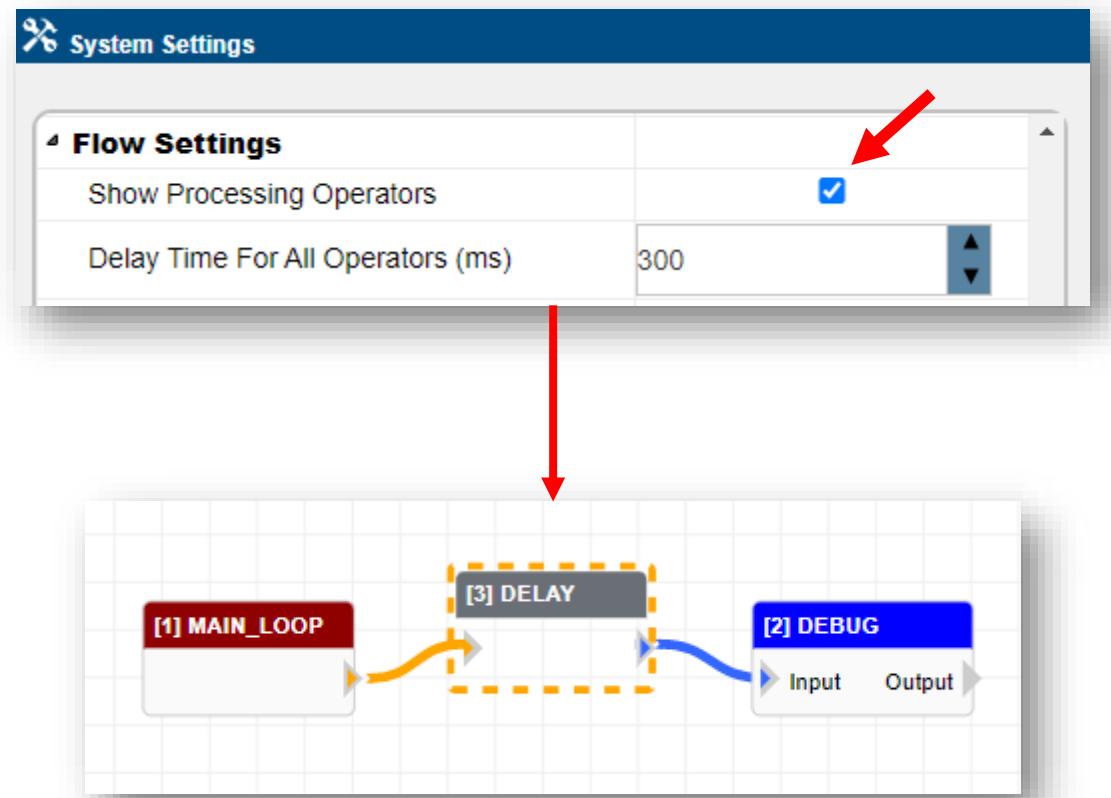


# VIC Flow 操作介面

- 顯示運作中的運算子

在系統設定視窗中可以找到顯示運作中的運算子

勾選後，在VIC Flow可以看到流程當前執行的進度



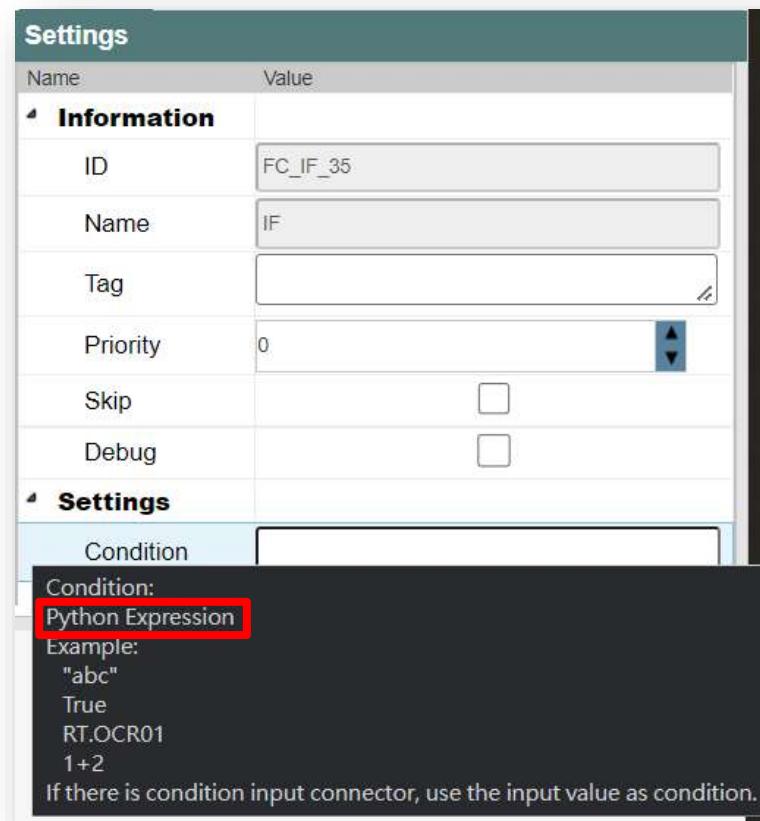
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

## • Python Expression

Python表達式，可以使用整數(Integer)、字串(String)、布林(Boolean)和VIC內部的辨識工具如TA.TARGET01、PAGE01.OCR01等等

舉例來說，IF運算子的條件欄位即為Python表達式

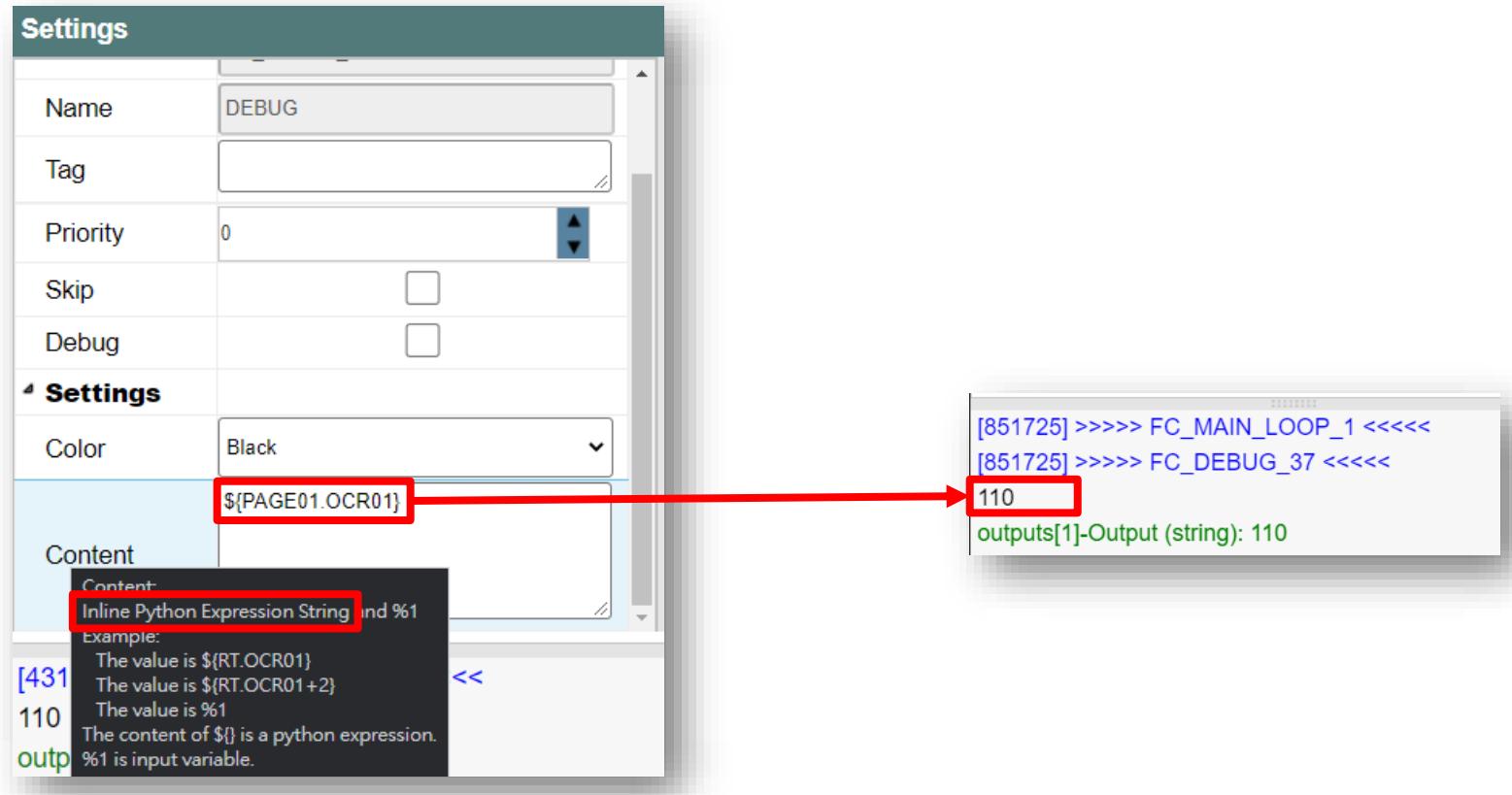


- **Inline Python Expression String**

Python表達式字串，將Python表達式以字串的類別來執行運算子

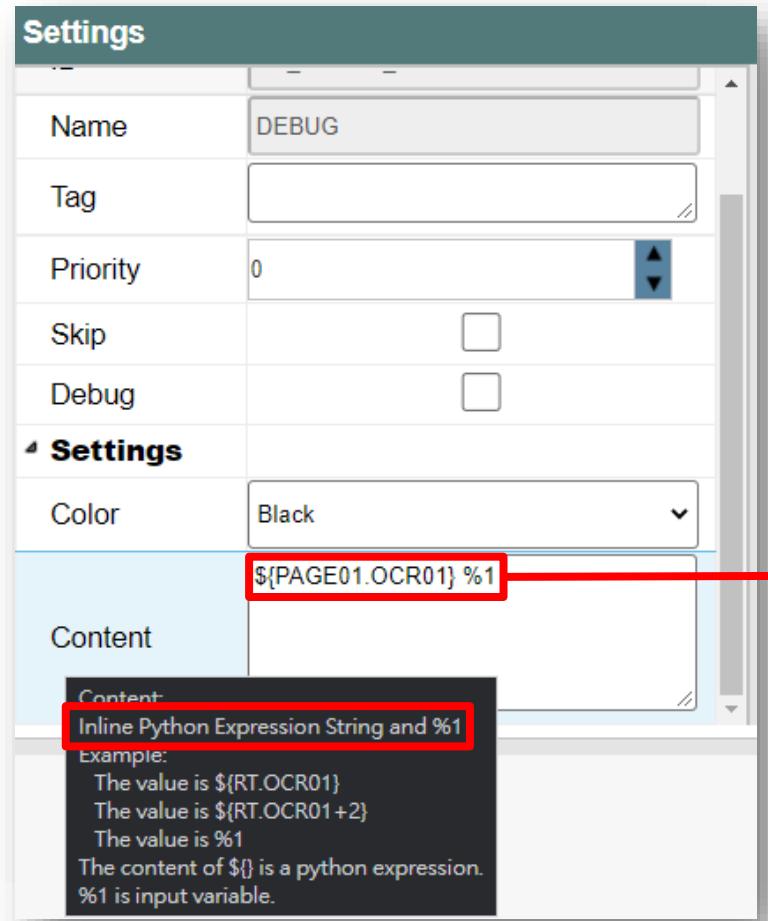
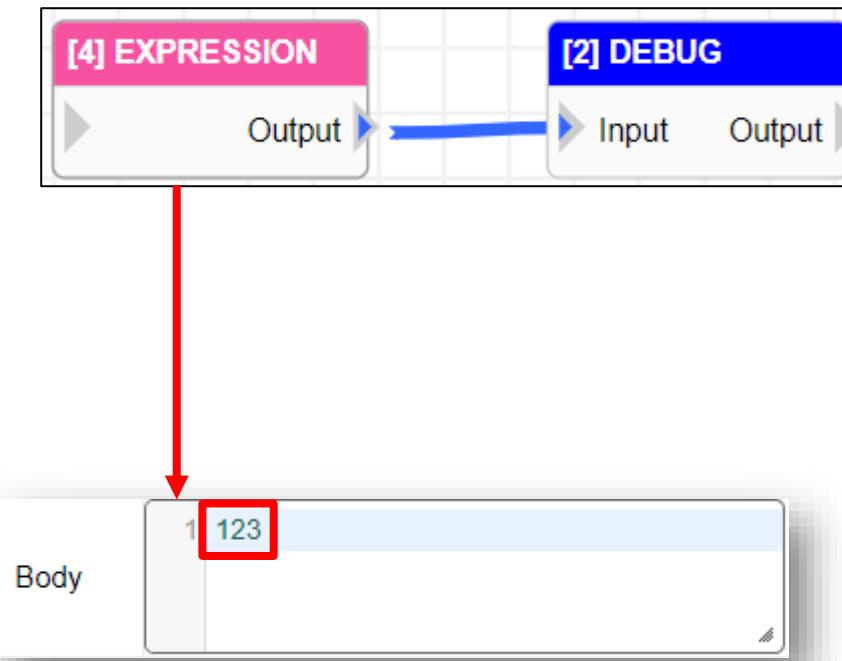
若要使用VIC的辨識工具，則必須使用 \${}

如\${RT.OCR01}、\${TA.TARGET01}、\${PAGE01.OCR01}等等



## • Inline Python Expression String and %1

同Python表達式字串，附加功能為透過「%1」，可用來呼叫該運算子的輸入端變數



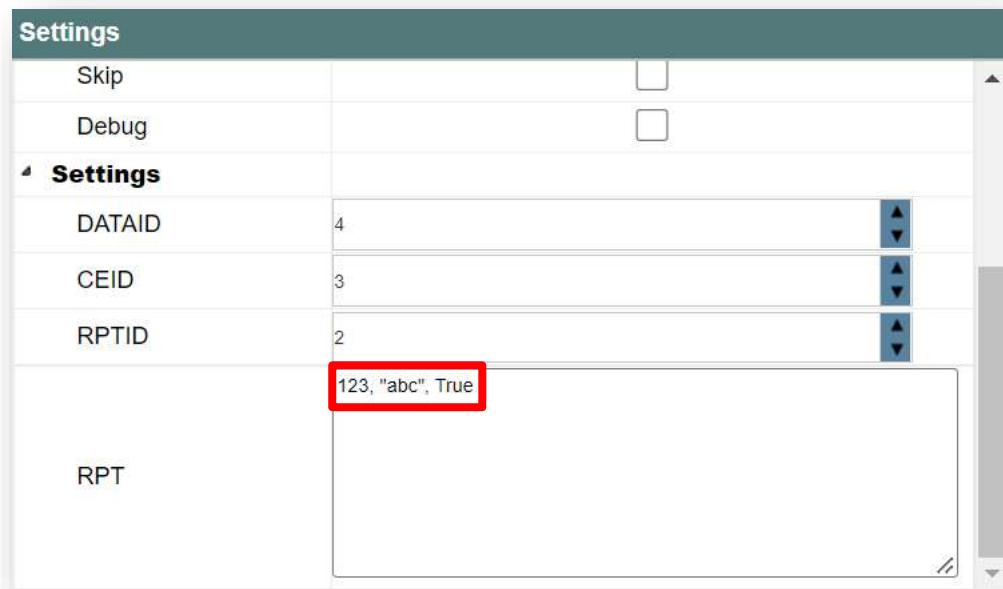
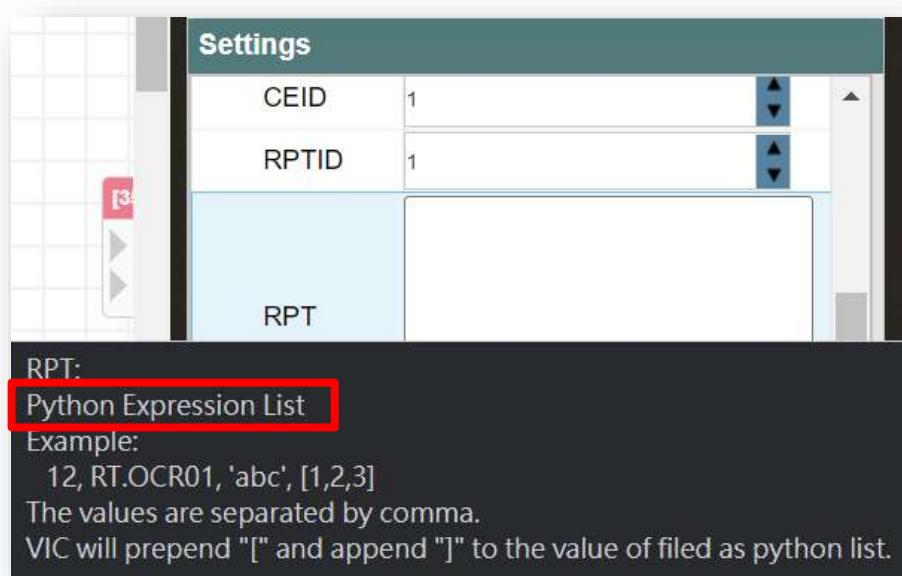
```
[475483] >>>> FC_EXPRESSION_4 <<<<
outputs[1]-Output (integer): 123
[475784] >>>> FC_DEBUG_2 <<<<
inputs[1]-Input (integer): 123
110 123
outputs[1]-Output (string): 110 123
```

# VIC Flow 表達式

- **Python Expression List**

同Python表達式，只是輸出為list

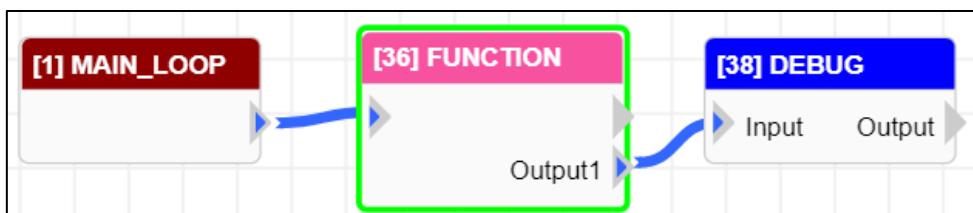
舉例來說，RT.OCR01, 123, 'abc'，輸出即為[RT.OCR01, 123, 'abc']



# VIC Flow 表達式

- **Python Code**

如同編寫Python程式語言



**Settings**

```
1 if int(RT.OCR01)>= int(RT.OCR02):  
2     #self.outputs[1] = int(RT.OCR01)+50  
3     print(RT.OCR01+' IS BIGGER')  
4 else:  
5     print(RT.OCR02+' is bigger')  
6
```

Body:  
Python Code

Example:

```
self.outputs[1]='abc'  
print('input1=',inputs[1]) # input1 must be connected  
self.skips[2] = True  
self.ret='123'  
[758271] >>>> FC_FUNC_36 <<<<  
110 IS BIGGER  
outputs[1]-Output1 (integer): 160  
[758271] >>>> FC_DEBUG_38 <<<<  
inputs[1]-Input (integer): 160  
110 160  
[758271] >>>> FC_DEBUG_38 <<<<
```

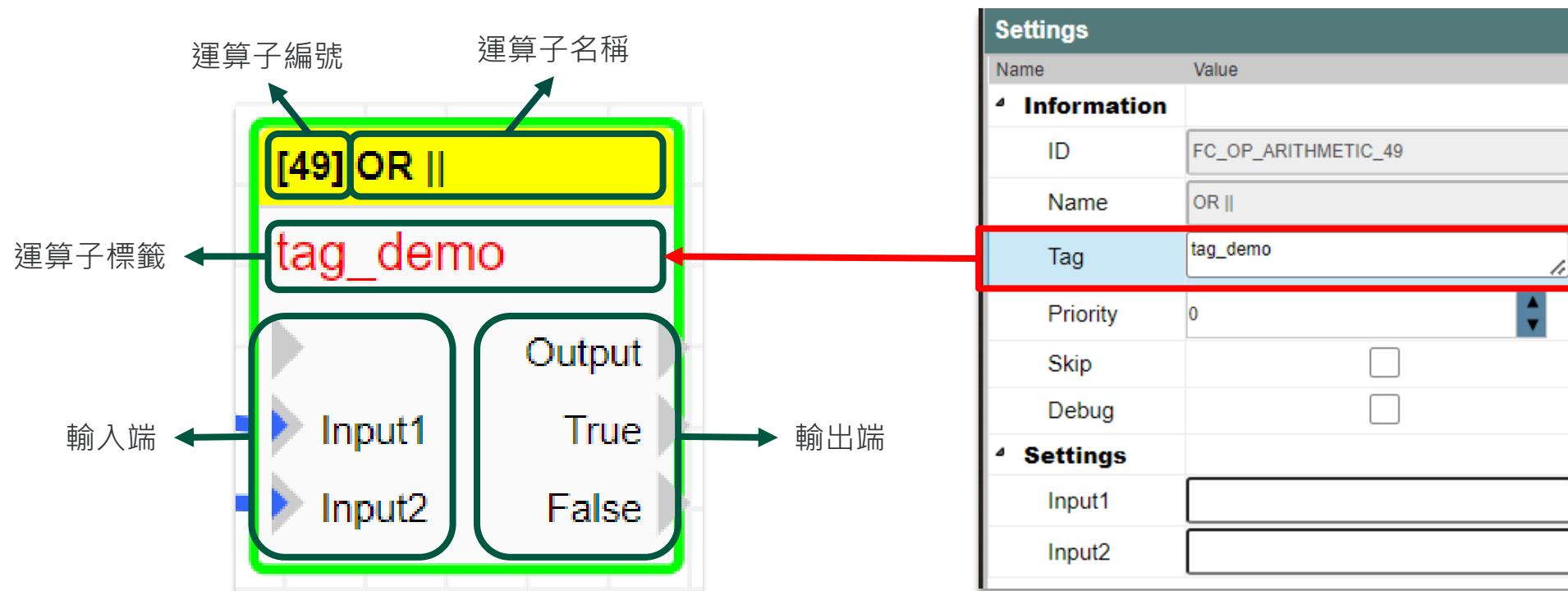
# VIC Flow 運算子

- 運算子

編輯區域內，運算子有固定的資訊和編輯欄位，以FUNCTION運算子為例

運算子的標籤欄位需要設置運算子內容的標籤才會顯示

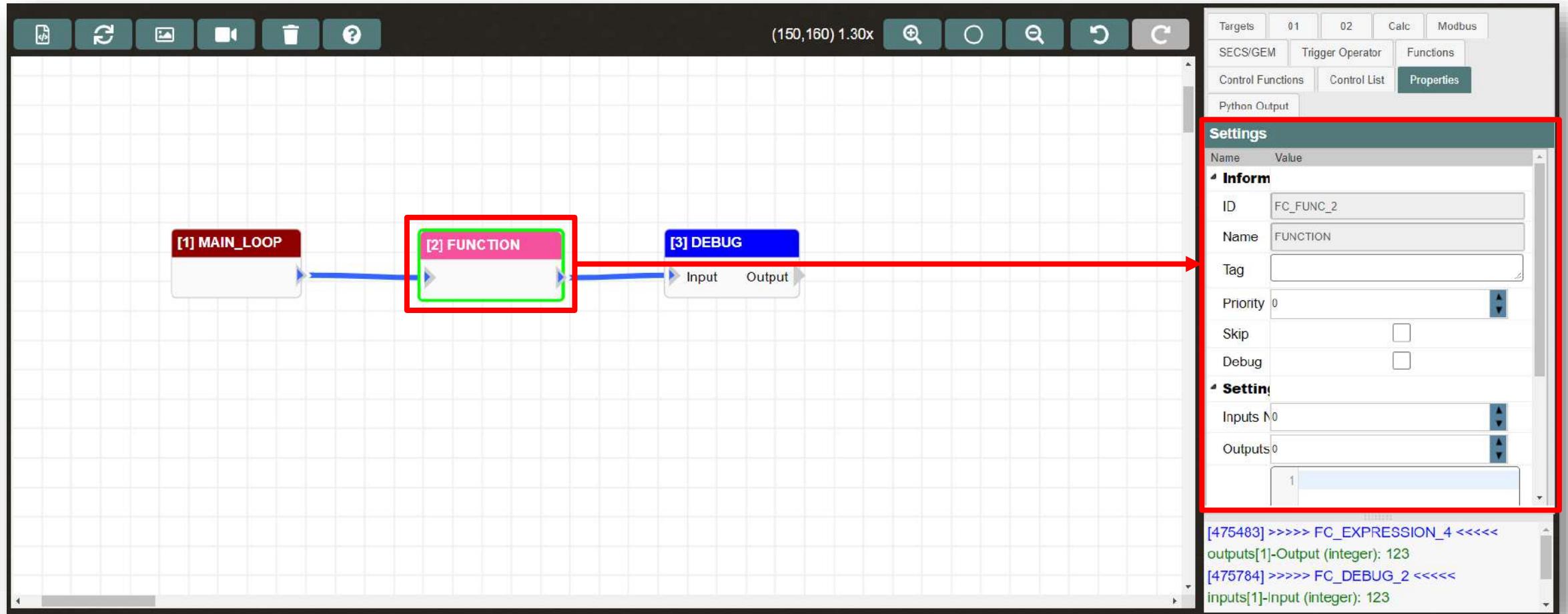
運算子編號是該運算子的創建順序，非執行順序



# VIC Flow 運算子

- 運算子屬性

在編輯區域內，雙擊運算子後，右邊的設置欄即會出現屬性調整欄位。



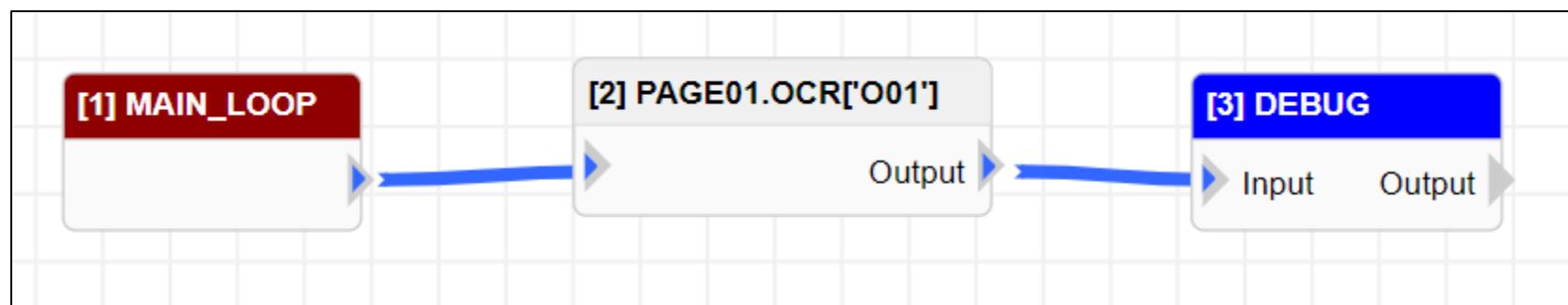
- **辨識結果**

取得辨識設定中，辨識頁面內的辨識工具的辨識結果

輸出端：輸出辨識工具的辨識結果

其他運算子可使用Python表達式(PAGE<sub>x</sub>.OCR<sub>y</sub>或\${PAGE<sub>x</sub>.OCR<sub>y</sub>})來取得辨識工具的辨識結果

x為頁面編號，y為辨識工具編號。



# VIC Flow 運算子

- 辨識結果

**Settings**

Name	Value
<b>Information</b>	
ID	FC_IMP_2
Name	PAGE01.OCR01
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

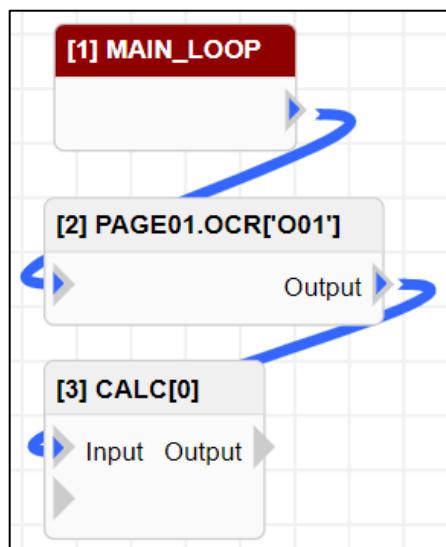
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

- 運算

運算是腳本內的變數，可用於存放執行結果，顯示執行結果在介面上，記錄執行結果並存進資料庫且可透過其他通訊協定進行傳輸，腳本內的運算變數共有100個

	UI名稱	行為
輸入端	Input	輸入欲寫入運算變數的值
輸出端	Output	輸出運算變數存放的值

寫進資料庫



```

[821733] >>>> FC_MAIN_LOOP_1 <<<<
[821734] >>>> FC_IMP_26 <<<<
outputs[1]-Output (string): 110
[821734] >>>> FC_CALC_25 <<<<
inputs[1]-Input (string): 110
outputs[1]-Output (string): 110
[821734] >>>> FC_DEBUG_27 <<<<
inputs[1]-Input (string): 110

```

	OCR_01	CALC_01	ts
110	110	2022-06-15 14:50:16.731	
110	110	2022-06-15 14:50:17.744	
110	110	2022-06-15 14:50:18.734	
110	110	2022-06-15 14:50:19.726	
110	110	2022-06-15 14:50:20.735	
110	110	2022-06-15 14:50:21.736	

# VIC Flow 運算子

- 運算

Settings

Name	Value
<b>Information</b>	
ID	FC_CALC_25
Name	CALC[0]
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Index	0

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 99	Integer

# Outline

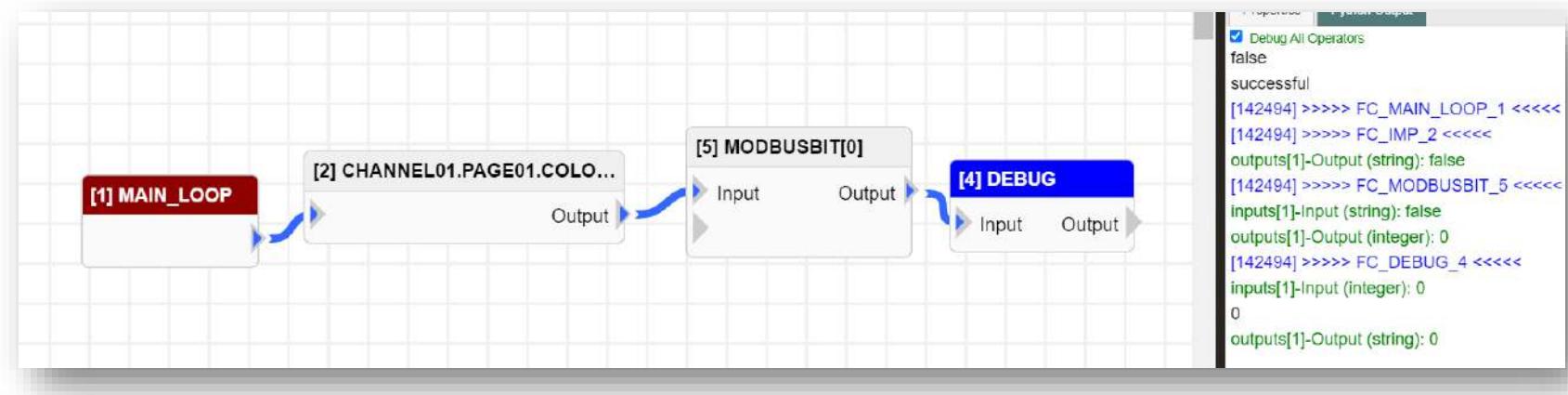
- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

## • Modbus Bit

自定義VIC的Modbus位元輸出，其通訊欄位為Input Status(1x)，可使用的Modbus位元共有100個

	UI名稱	行為
輸入端	Input	輸入欲寫入Modbus位元的值
輸出端	Output	輸出Modbus位元存放的值

地址範圍(1x)	說明
10101	MODBUSBIT[0]
10102	MODBUSBIT[1]
.	.
.	.
10200	MODBUSBIT[99]



- Modbus Bit

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUSBIT_24
Name	MODBUSBIT[0]
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Index	0

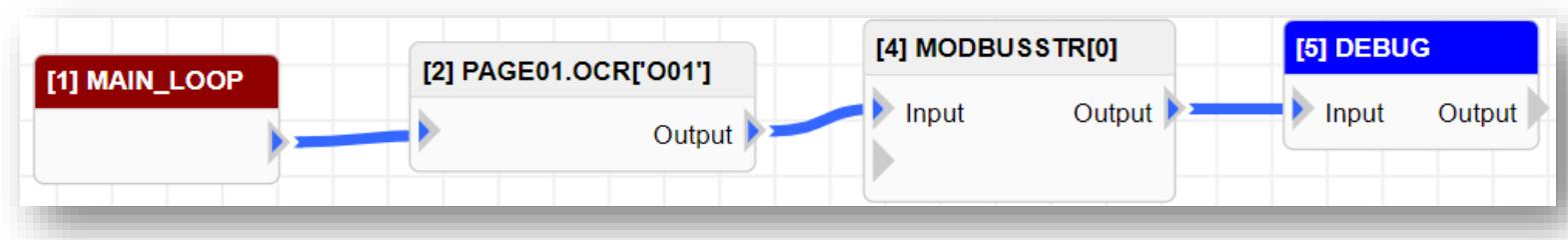
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 99	Integer

- **Modbus String**

自定義VIC的Modbus字串輸出，其通訊欄位為Input register(3x)，一個MODBUSSTR的長度為249bytes，等同於125個Word，可使用的Modbus字串共有100個

	UI名稱	行為
輸入端	Input	輸入欲寫入ModbusH字串的值
輸出端	Output	輸出ModbusH字串存放的值

地址範圍(3x)	說明
337501 ~ 337625	MODBUSSTR[0]
337626 ~ 337750	MODBUSSTR[1]
.	.
349876 ~ 350000	MODBUSSTR[99]



- Modbus String

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUSSTR_26
Name	MODBUSSTR[0]
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Index	0

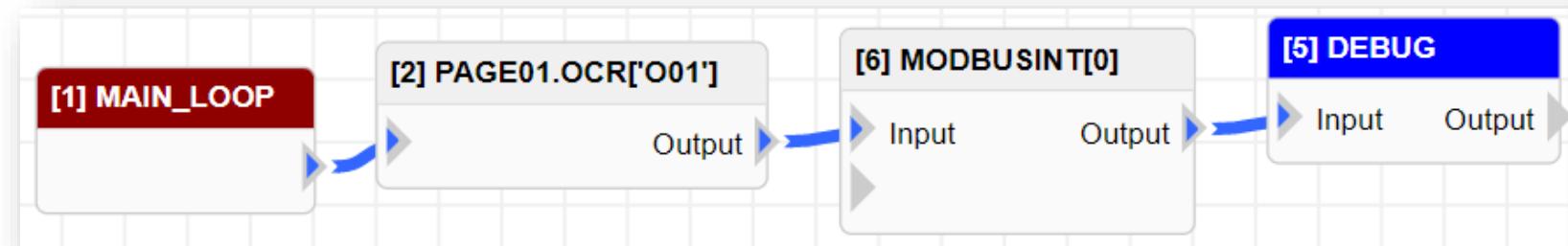
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 99	Integer

- **Modbus Int**

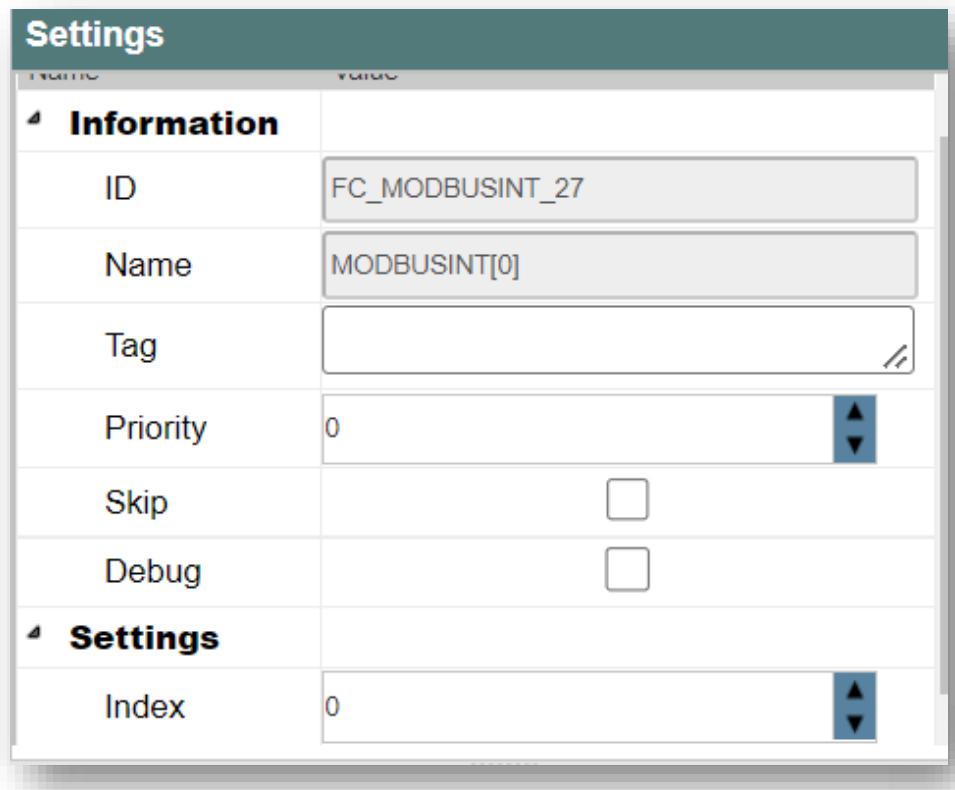
自定義VIC的Modbus Int輸出，其通訊欄位為Input Register(3x)，一個MODBUSINT的長度為4 bytes，等同於2個Word，一個Modbus Int為32位元並帶有正負號的整數整數範圍為 -2,147,483,648 ~ 2,147,483,647，可使用的Modbus Int共有100個

	UI名稱	行為
輸入端	Input	輸入欲寫入ModbusH Int的值
輸出端	Output	輸出ModbusH Int存放的值

地址範圍(3x)	說明
350401 ~ 350402	MODBUSINT[0]
350403 ~ 350404	MODBUSINT[1]
⋮	⋮
350599 ~ 350600	MODBUSINT[99]



- Modbus Int



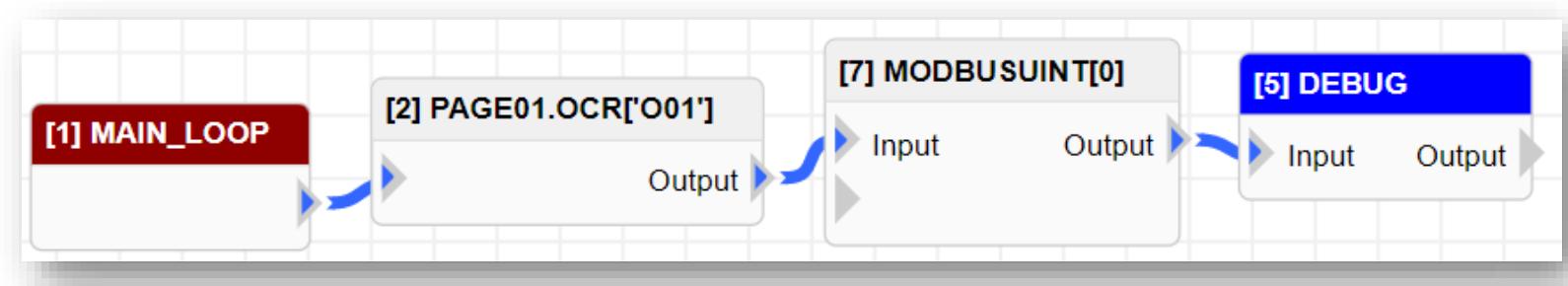
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 99	Integer

## • Modbus UInt

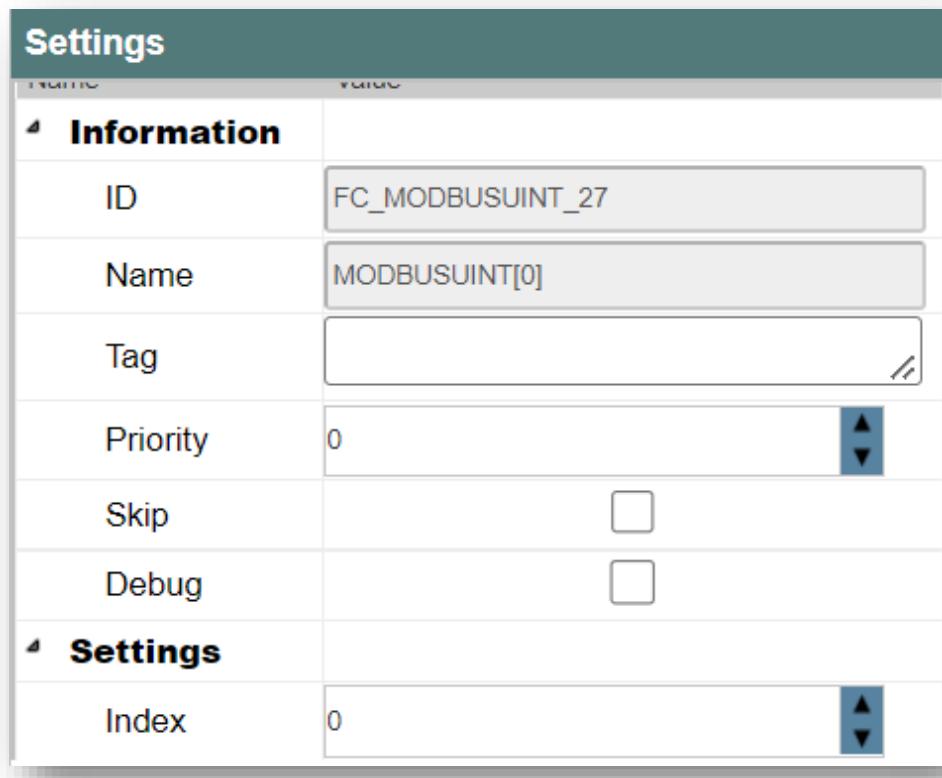
自定義VIC的Modbus UInt輸出，其通訊欄位為Input Register(3x)，一個MODBUSUINT的長度為4 bytes，等同於2個Word，一個Modbus UInt為32位元且不帶正負號的整數範圍為0 ~ 4,294,967,295，可使用的Modbus UInt共有100個

	UI名稱	行為
輸入端	Input	輸入欲寫入ModbusH UInt的值
輸出端	Output	輸出ModbusH UInt存放的值

地址範圍(3x)	說明
350601 ~ 350602	MODBUSUINT[0]
350603 ~ 350604	MODBUSUINT[1]
...	
350799 ~ 350800	MODBUSUINT[99]



- Modbus Uint



名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 99	Integer

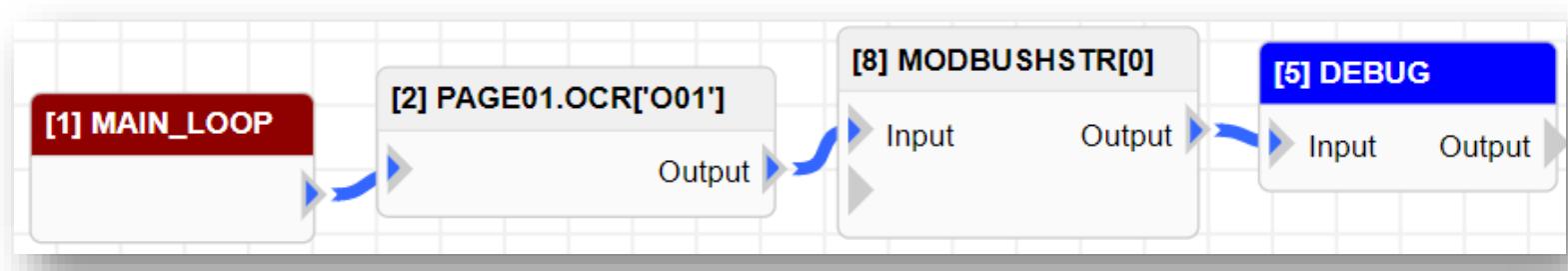
## • ModbusH String

可自定義VIC的Modbus字串輸出，其通訊欄位為Holding Register(4x)

一個MODBUSHSTR的長度為101bytes，等同於51個Word，可使用的ModbusH字串有200個

	UI名稱	行為
輸入端	Input	輸入欲寫入ModbusH字串的值
輸出端	Output	輸出ModbusH字串存放的值

地址範圍(4x)	說明
40001 ~ 40051	MODBUSHSTR[0]
40052 ~ 40102	MODBUSHSTR[1]
.	.
410150 ~ 410200	MODBUSHSTR[199]



- ModbusH String

Settings	
Name	Value
<b>Information</b>	
ID	FC_MODBUSHSTR_28
Name	MODBUSHSTR[0]
Tag	//
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Index	0

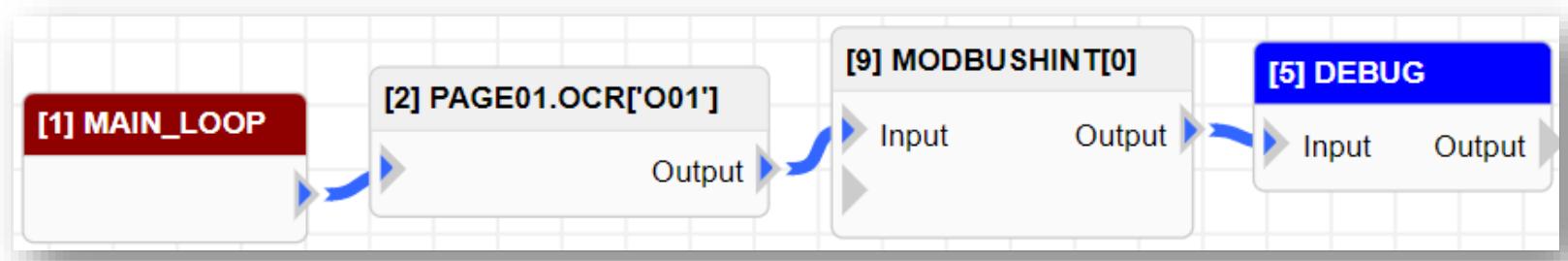
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 199	Integer

## • ModbusH Int

自定義VIC的Modbus Int輸出，其通訊欄位為Holding Register(4x)，一個MODBUSHINT的長度為4 bytes，等同於2個Word，一個ModbusH Int為32位元且不帶正負號的整數範圍為 -2,147,483,648 ~ 2,147,483,647，可使用的ModbusH Int有500個

	UI名稱	行為
輸入端	Input	輸入欲寫入ModbusH Int的值
輸出端	Output	輸出ModbusH Int存放的值

地址範圍(4x)	說明
410201 ~ 410202	MODBUSHINT[0]
410203 ~ 410204	MODBUSHINT[1]
⋮	⋮
411199 ~ 411200	MODBUSHINT[499]



- ModbusH Int

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUSHUINT_10
Name	MODBUSHUINT[0]
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Index	0

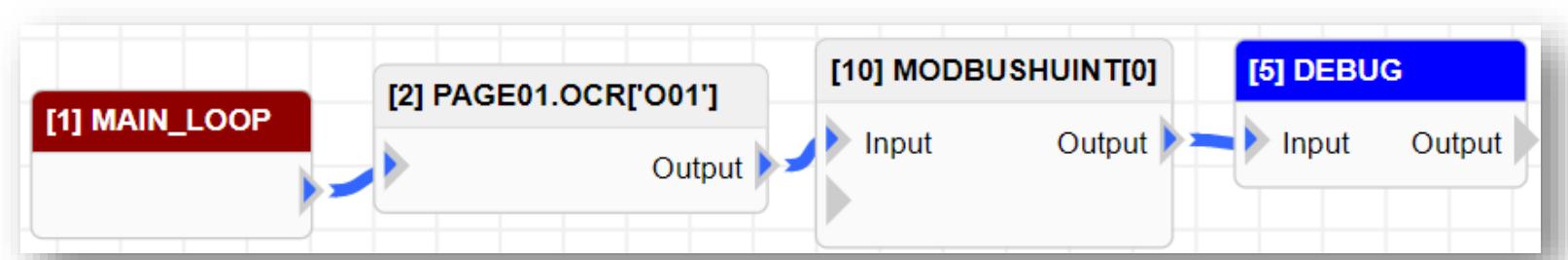
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 499	Integer

## • ModbusH UInt

自定義VIC的Modbus UInt輸出，其通訊欄位為Holding Register(4x)，一個MODBUSHUINT的長度為4 bytes，等同於2個Word，一個ModbusH UInt為32位元且不帶正負號的整數範圍為0 ~ 4,294,967,295，可使用的ModbusH UInt有500個

	UI名稱	行為
輸入端	Input	輸入欲寫入ModbusH UInt的值
輸出端	Output	輸出ModbusH UInt存放的值

地址範圍(4x)	說明
411201 ~ 411202	MODBUSHUINT[0]
411203 ~ 411204	MODBUSHUINT[1]
⋮	⋮
412199 ~ 412200	MODBUSHUINT[499]



# VIC Flow Modbus

- **ModbusH UInt**

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUSHUINT_29
Name	MODBUSHUINT[0]
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Index	0

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>索引 (Index)</b>	設定該運算子使用的運算變數的編號 範圍為 0 ~ 499	Integer

## • Read Modbus System Bit

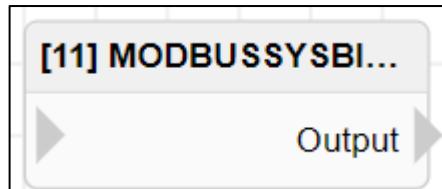
讀取VIC系統Modbus位元

可以用來取得系統當前狀況

目前規劃的系統Modbus位元共有

16個，其通訊欄位為Input Status(1x)

	UI名稱	行為
輸入端		
輸出端	Output	輸出系統Modbus位元的值



Bit 7	Bit 6	Sum	Result
0	0	0	辨識成功
0	1	1	辨識 NG
1	0	2	比對失敗

Input Status (1X)	
Address	Content
10001	程式是否運行中， 1 = 運行中，0 = 停止
10002	是否有使用者登入， 1 = 有人登入，0 = 無人登入
10003	系統硬碟容量 5G 檢測， 1 = 低於 5G，0 = 高於 5G
10004	系統硬碟容量 10G 檢測， 1 = 低於 10G，0 = 高於 10G
10005	是否停止存圖， 1 = 停止，0 = 運行中
10006 (Bit 6)	OCR辨識結果 0 = 辨識成功，1 = NG，2 = 比對失敗
10007 (Bit 7)	
10009	序列埠控制是否被啟用， 1 = 取用，0 = 停用
10010	序列埠目前狀態， 1 = 連線，0 = 斷線
10017	資料硬碟容量 10G 檢測， 1 = 低於 5G，0 = 高於 5G
10018	資料硬碟容量 20G 檢測， 1 = 低於 10G，0 = 高於 10G

- Read Modbus System Bit

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUS_SYS_BIT_28
Name	MODBUSSYSBIT[0]
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

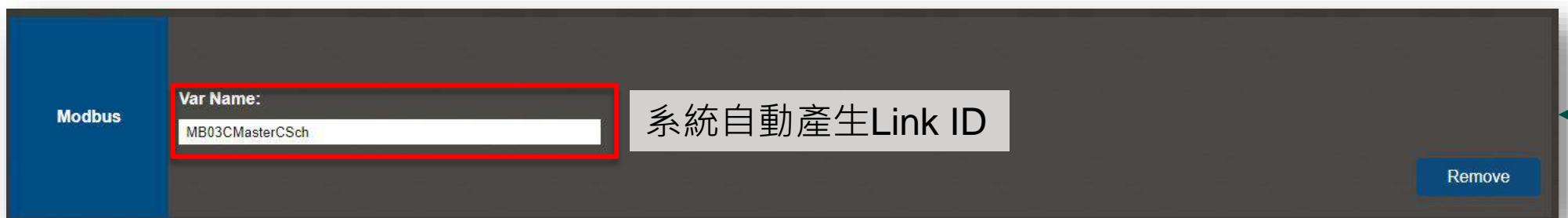
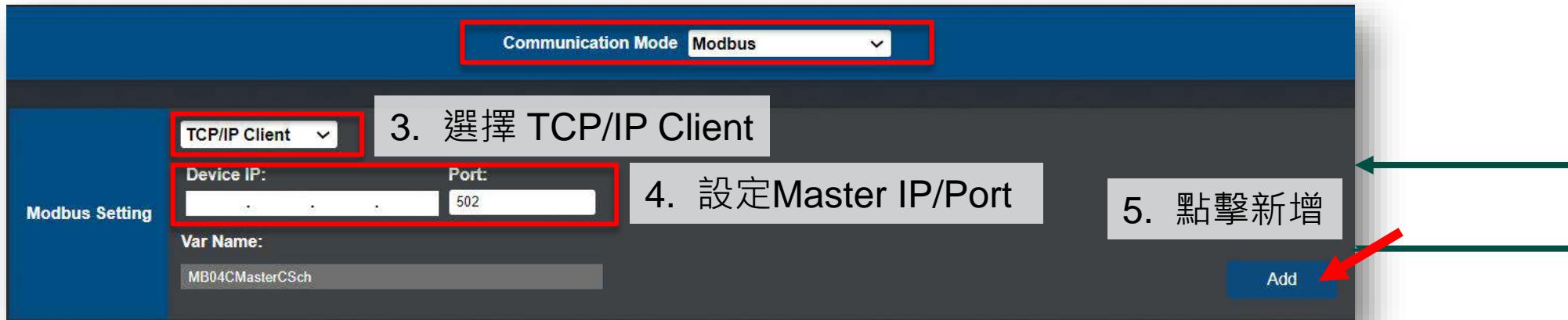
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

- Modbus TCP/IP Client 連線

1. 前往Link頁面

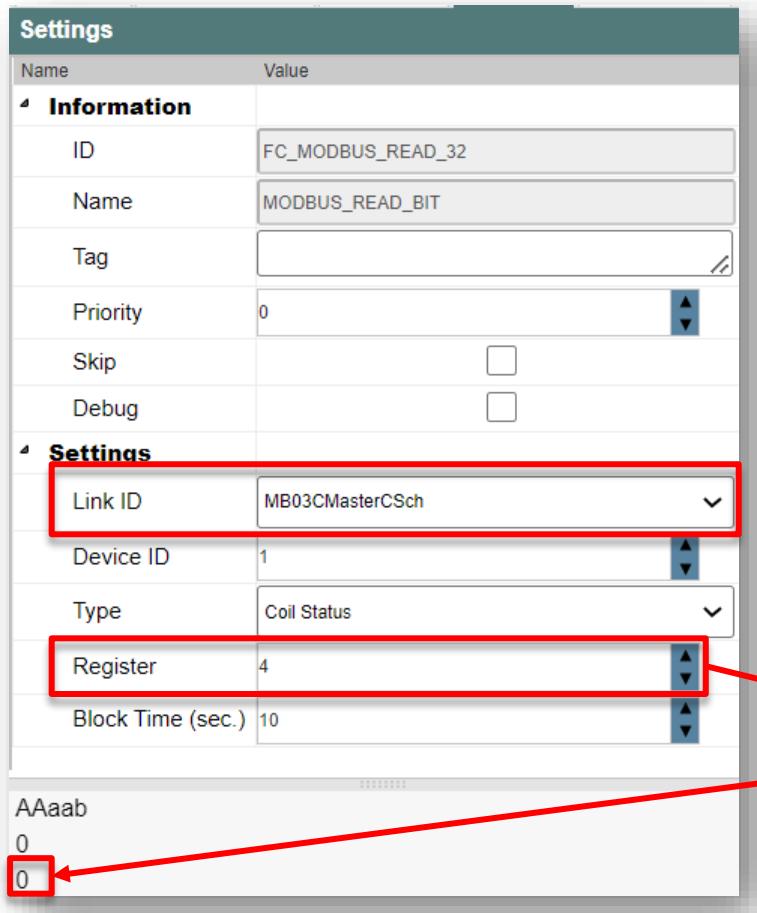


2. 通訊模式選擇選擇Modbus

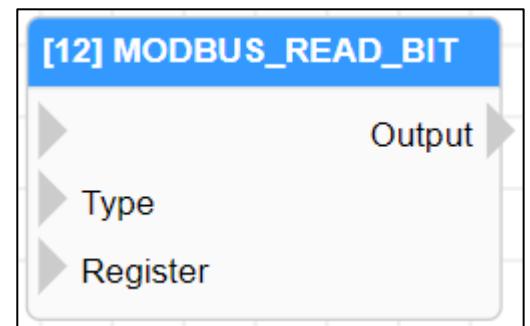
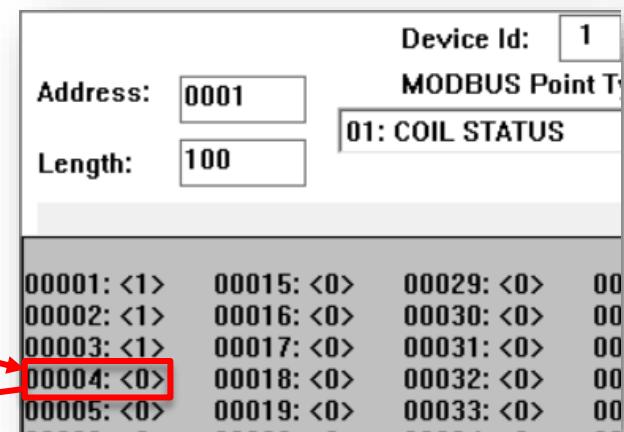


## • Modbus Read Bit

讀取Modbus TCP Server或Modbus RTU Slave的位元資料



	UI名稱	行為
輸入端	Type	輸入欲讀取的類型，也就是要讀取的位址 如Coil Status(0x)與Input Status(1x)
	Register	輸入欲讀取的暫存器位址
輸出端	Output	輸出讀取結果



## • Modbus Read Bit

若類型(Type)與暫存器(Register)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置欲使用的Modbus連線	
<b>站號 (Device ID)</b>	設置欲連接裝置的站號	Integer
<b>類型 (Type)</b>	設置欲讀取的Modbus位址代碼 (0 = Coil Status、1 = Input Status)	
<b>暫存器 (Register)</b>	設置欲讀取的暫存器位址	Integer
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUS_READ_12
Name	MODBUS_READ_BIT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	MB00CMasterCSch
Device ID	1
Type	Coil Status
Register	1
Block Time (sec.)	10

- **Modbus Write Bit**

寫入位元資料至Modbus TCP Server或Modbus RTU Slave的Coil Status位址。

Settings	
Name	Value
<b>Information</b>	
ID	FC_MODBUS_WRITE_31
Name	MODBUS_WRITE_BIT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	MB03CMasterCSch
Device ID	1
<b>Register</b>	1
<b>Value</b>	1
Block Time (sec.)	10

	UI名稱	行為
輸入端	Register	輸入欲寫入的暫存器位址
	Value	輸入欲寫入的值
輸出端		

Device Id: 1 MODBUS Point Table				
Address:	0001	Length:	100	01: COIL STATUS
00001: <1>	00015: <0>	00029: <0>	00033: <0>	00037: <0>
00002: <1>	00016: <0>	00030: <0>	00034: <0>	00038: <0>
00003: <1>	00017: <0>	00031: <0>	00035: <0>	00039: <0>
00004: <0>	00018: <0>	00032: <0>	00036: <0>	00040: <0>
00005: <0>	00019: <0>	00033: <0>	00037: <0>	00041: <0>
00006: <0>	00020: <0>	00034: <0>	00038: <0>	00042: <0>
00007: <0>	00021: <0>	00035: <0>	00039: <0>	00043: <0>
00008: <0>	00022: <0>	00036: <0>	00040: <0>	00044: <0>

[13] MODBUS\_WRITE\_BIT

- Register
- Value

## • Modbus Write Bit

若類型(Type)與暫存器(Register)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置欲使用的Modbus連線	
<b>站號 (Device ID)</b>	設置欲連接裝置的站號	Integer
<b>暫存器 (Register)</b>	設置欲寫入的暫存器位址	Integer
<b>值 (Value)</b>	設置欲寫入的數值	Integer
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUS_WRITE_13
Name	MODBUS_WRITE_BIT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	MB00CMasterCSch
Device ID	1
Register	1
Value	0
Block Time (sec.)	10

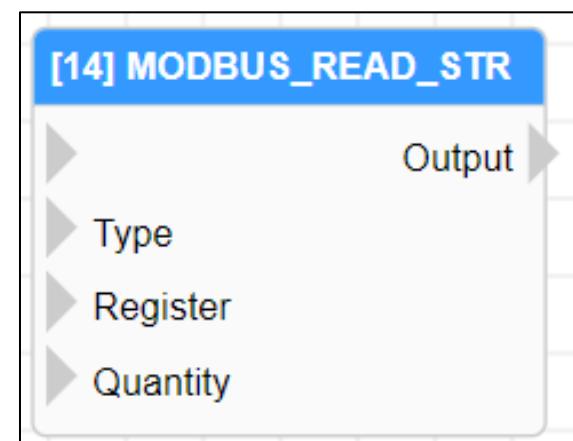
## • Modbus Read String

讀取Modbus TCP Server或Modbus RTU Slave的字串資料，將暫存器內的資料以ASCII標準轉換成字串，從低位元開始填入資料

Settings	
Information	
ID	FC_MODBUS_READ_29
Name	MODBUS_READ_STR
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Settings	
Link ID	MB03CMasterCSch
Device ID	1
Type	Holding Register
Register	1
Quantity	3
Block Time (sec.)	10

	UI名稱	行為
輸入端	Type	輸入欲讀取的類型，也就是要讀取的位址如Input Register(3)與Holding Register(4)
	Register	輸入欲讀取的暫存器位置
	Quantity	輸入欲讀取的暫存器數量
輸出端	Output	輸出讀取結果

Device Id:	1	
Address:	0001	
Length:	100	
MODBUS Point Type:	03: HOLDING REGISTER	
40001: <4141H>	40015: <0000H>	40
40002: <6161H>	40016: <0000H>	40
40003: <0062H>	40017: <0000H>	40
40004: <0000H>	40018: <0000H>	40
40005: <0000H>	40019: <0000H>	40
	40020: <0000H>	40
	40021: <0000H>	40
	40022: <0000H>	40
	40023: <0000H>	40
	40024: <0000H>	40



## • Modbus Read String

若類型(Type)與暫存器(Register)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置欲使用的Modbus連線	
<b>站號 (Device ID)</b>	設置欲連接裝置的站號	Integer
<b>類型 (Type)</b>	設置欲讀取的Modbus位址代碼 (3 = Input Register、4 = Holding Register)	
<b>暫存器 (Register)</b>	設置欲讀取的暫存器位址	Integer
<b>數量 (Quantity)</b>	設置欲讀取的暫存器數量	Integer
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

Settings	
Name	Value
<b>Information</b>	
<b>ID</b>	FC_MODBUS_READ_14
<b>Name</b>	MODBUS_READ_STR
<b>Tag</b>	
<b>Priority</b>	0
<b>Skip</b>	<input type="checkbox"/>
<b>Debug</b>	<input type="checkbox"/>
<b>Settings</b>	
<b>Link ID</b>	MB00CMasterC5ch
<b>Device ID</b>	1
<b>Type</b>	Holding Register
<b>Register</b>	1
<b>Quantity</b>	1
<b>Block Time (sec.)</b>	10

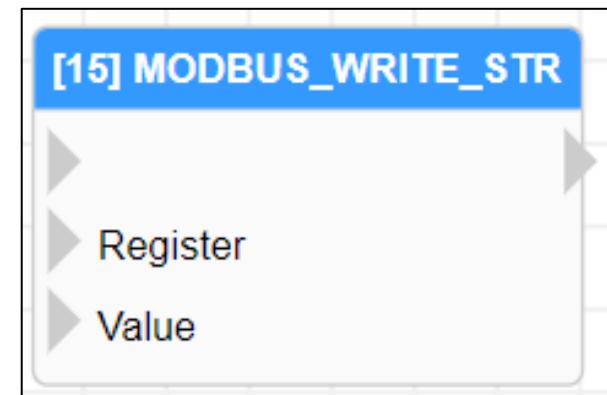
## • Modbus Write String

寫入字串資料至Modbus TCP Server或Modbus RTU Slave的Holding Register位址

Settings	
ID	FC_MODBUS_READ_29
Name	MODBUS_READ_STR
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	MB03CMasterCSch
Device ID	1
Type	Holding Register
Register	2
Quantity	1
Block Time (sec)	10
<pre>a a a</pre>	

	UI名稱	行為
輸入端	Register	輸入欲寫入的暫存器位址
	Value	輸入欲寫入的值
輸出端		

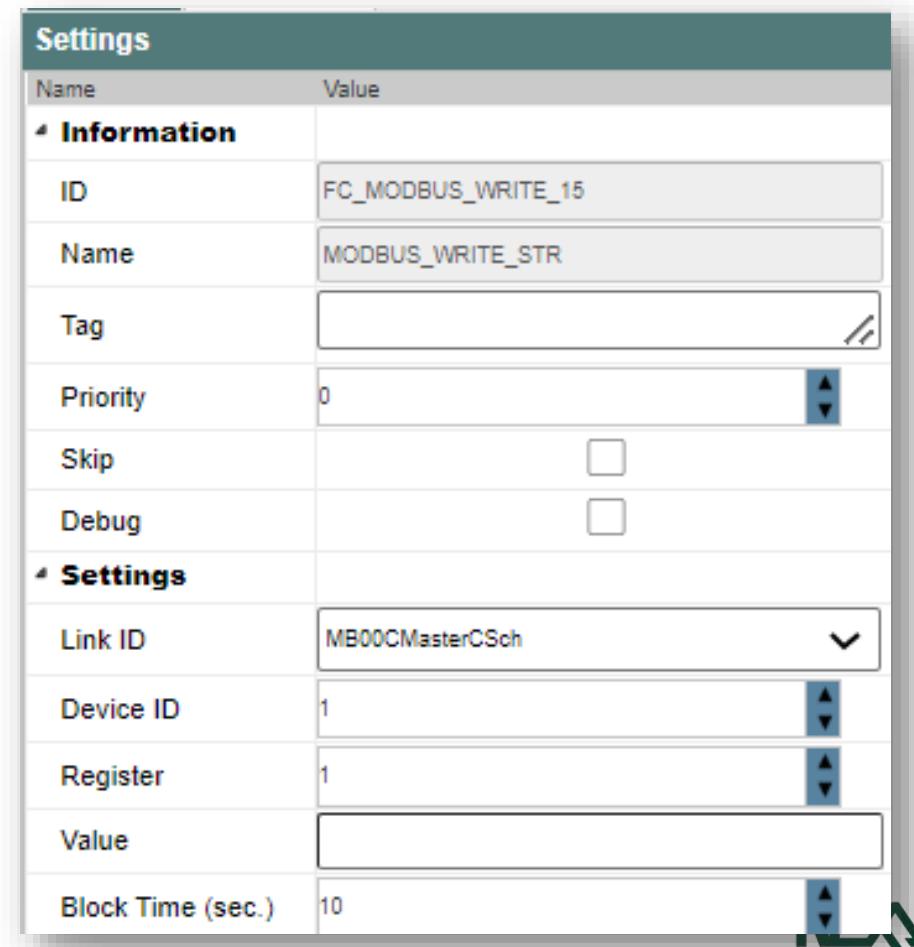
Device Id: 1	MODBUS Point T
Address: 0001	03: HOLDING REGISTERS
Length: 100	
40001: <00065>	40
40002: <00097>	40
40003: <00000>	40
40004: <00000>	40
40005: <00000>	40
40006: <00000>	40
40007: <00000>	40
40008: <00000>	40
40009: <00000>	40
40010: <00077>	40
40011: <00000>	40
40012: <00000>	40
40015: <00000>	40
40016: <00000>	40
40017: <00000>	40
40018: <00000>	40
40019: <00000>	40
40020: <00000>	40
40021: <00000>	40
40022: <00000>	40
40023: <00000>	40
40024: <00000>	40
40025: <00000>	40
40026: <00000>	40



## • Modbus Write String

若類型 (Type)與暫存器(Register)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值

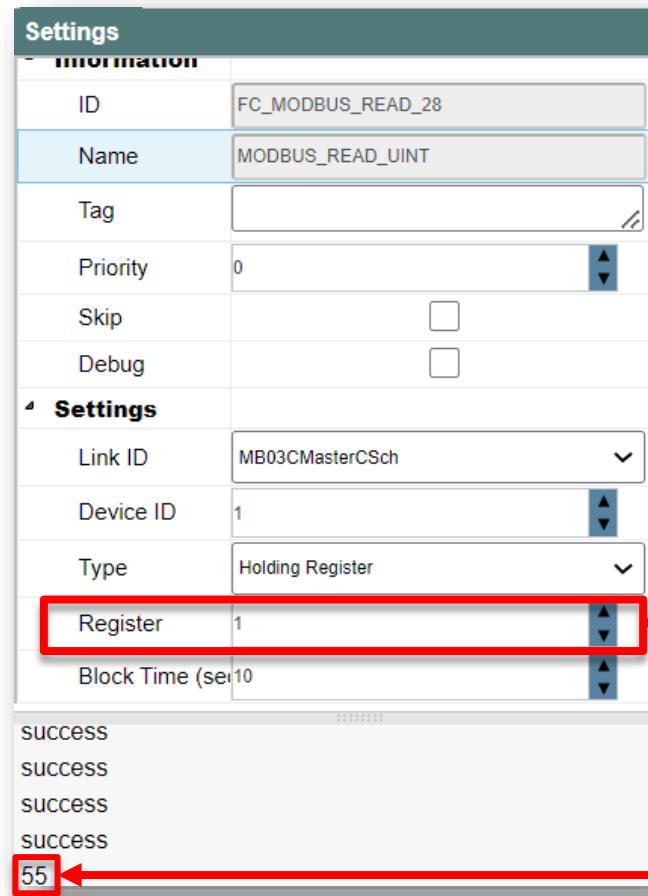
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置欲使用的Modbus連線	
<b>站號 (Device ID)</b>	設置欲連接裝置的站號	Integer
<b>暫存器 (Register)</b>	設置欲寫入的暫存器位址	Integer
<b>值 (Value)</b>	欲寫入的值	String
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer



## • Modbus Read Unit

讀取Modbus TCP Server或Modbus RTU Slave的整數資料，讀取單一暫存器內的整數資料

資料格式為16 bits無號整數



UI名稱		行為
輸入端	Type	輸入欲讀取的類型，也就是要讀取的位址 如Input Register(3)與Holding Register(4)
	Register	輸入欲讀取的暫存器位址
輸出端	Output	輸出讀取結果

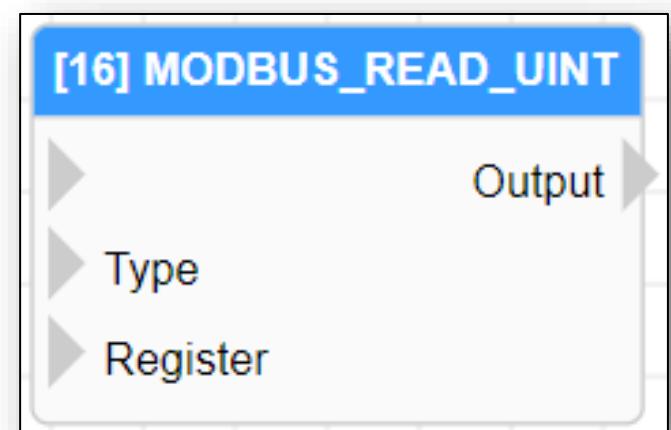
[16] MODBUS_READ_UINT		
Address: 0001	Device Id: 1	MODBUS Point Type: 03: HOLDING REGISTER
Length: 100		

The table shows the Modbus read parameters:

- Address: 0001
- Device Id: 1
- MODBUS Point Type: 03: HOLDING REGISTER
- Length: 100

The data table lists 16 registers, with the first one highlighted in red:

Register Address	Value	Length
40001: <000055>	40015: <000000>	40
40002: <000000>	40016: <000000>	40
40003: <000000>	40017: <000000>	40
40004: <000000>	40018: <000000>	40
40005: <000000>	40019: <000000>	40
40006: <000000>	40020: <000000>	40
40007: <000000>	40021: <000000>	40
40008: <000000>	40022: <000000>	40
40009: <000000>	40023: <000000>	40
40010: <000077>	40024: <000000>	40
40011: <000000>	40025: <000000>	40
40012: <000000>	40026: <000000>	40
40013: <000000>	40027: <000000>	40
40014: <06666>	40028: <000000>	40



## • Modbus Read Unit

若類型(Type)與暫存器(Register)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置欲使用的Modbus連線	
<b>站號 (Device ID)</b>	設置欲連接裝置的站號	Integer
<b>類型 (Type)</b>	設置欲讀取的Modbus位址代碼 (3 = Input Register, 4 = Holding Register)	
<b>暫存器 (Register)</b>	設置欲讀取的暫存器位址	Integer
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODBUS_READ_16
Name	MODBUS_READ_UINT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	MB00CMasterCSch
Device ID	1
Type	Holding Register
Register	1
Block Time (sec.)	10

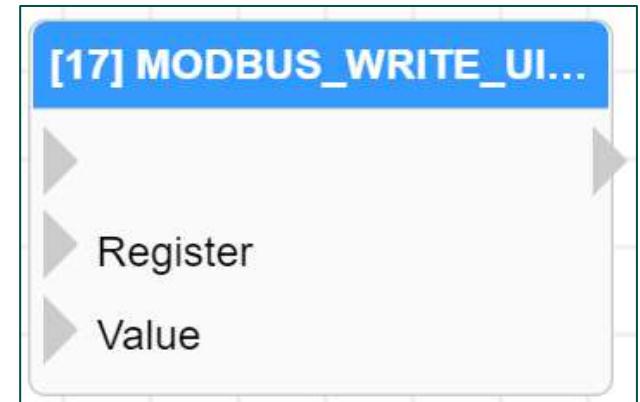
## • Modbus Write Unit

寫入整數資料至Modbus TCP Server或Modbus RTU Slave的Holding Register位址

Settings	
Name	Value
<b>Information</b>	
ID	FC_MODBUS_WRITE_25
Name	MODBUS_WRITE_UINT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	MB03CMasterCSch
Device ID	1
Register	10
Value	77
Block Time (sec)	10

	UI名稱	行為
輸入端	Register	輸入欲寫入的暫存器位置
	Value	輸入欲寫入的值
	輸出端	

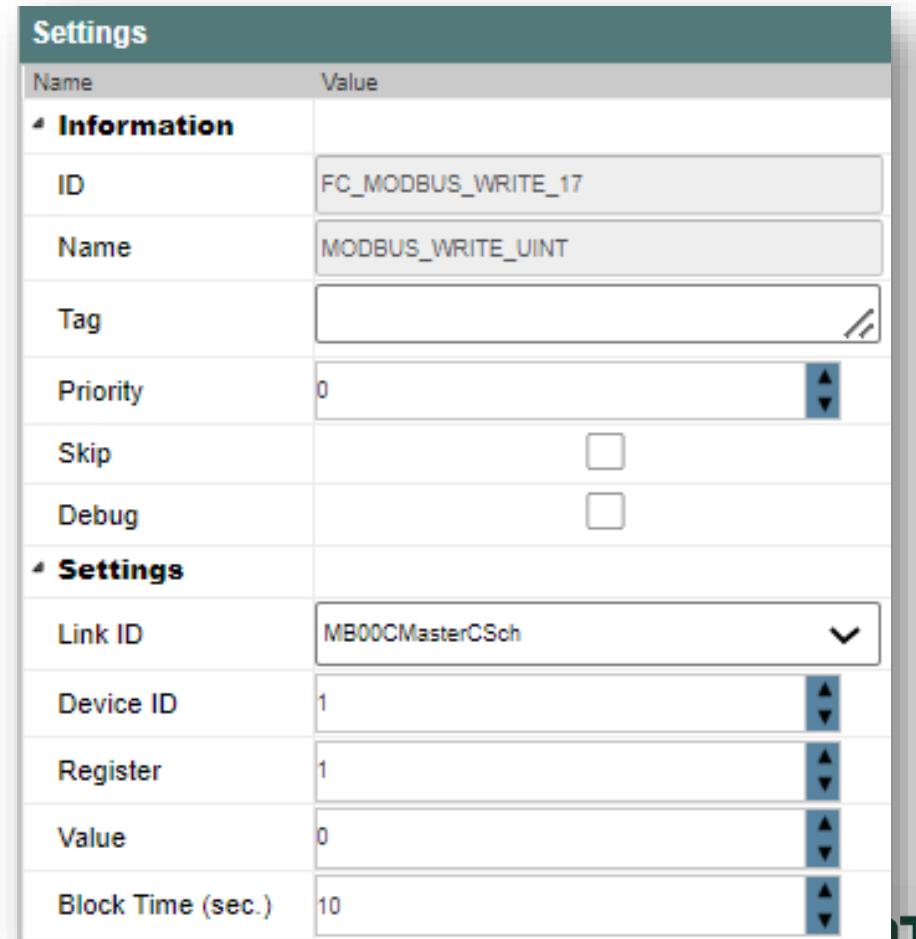
Device Id: 1	MODBUS Point Ty
Address: 0001	03: HOLDING REGIST
Length: 100	
40001: <000055>	40015: <000000> 40
40002: <000000>	40016: <000000> 40
40003: <000000>	40017: <000000> 40
40004: <000000>	40018: <000000> 40
40005: <000000>	40019: <000000> 40
40006: <000000>	40020: <000000> 40
40007: <000000>	40021: <000000> 40
40008: <000000>	40022: <000000> 40
40009: <000000>	40023: <000000> 40
40010: <000077>	40024: <000000> 40
40011: <000000>	40025: <000000> 40
40012: <000000>	40026: <000000> 40
40013: <000000>	40027: <000000> 40
40014: <066666>	40028: <000000> 40



## • Modbus Write Unit

若類型(Type)與暫存器(Register)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置欲使用的Modbus連線	
<b>站號 (Device ID)</b>	設置欲連接裝置的站號	Integer
<b>暫存器 (Register)</b>	設置欲寫入的暫存器位址	Integer
<b>值 (Value)</b>	欲寫入的數值	Integer
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer



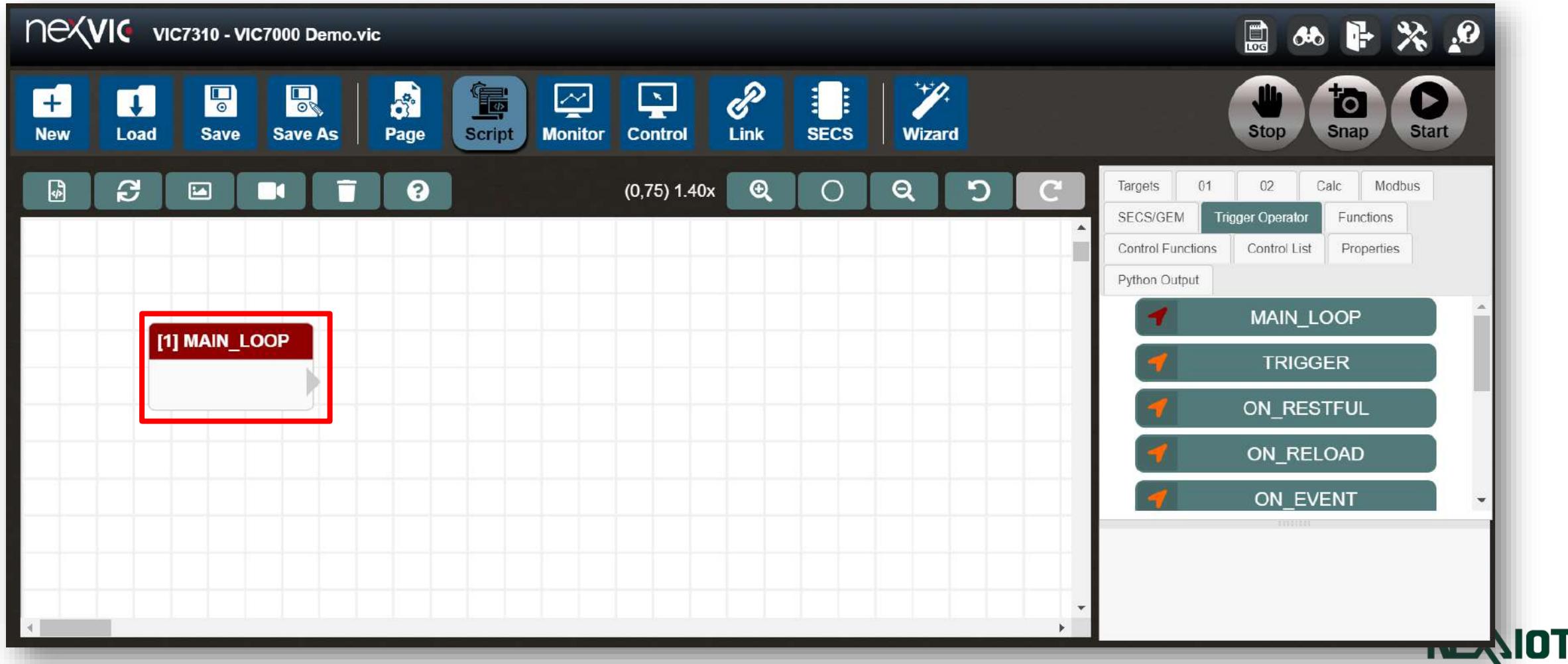
# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# VIC Flow 觸發運算子

## • MAIN\_LOOP

VIC主要迴圈，本軟體每次執行辨識時都會觸發此運算子



# VIC Flow 觸發運算子

- **MAIN\_LOOP**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

**Settings**

Name	Value
Information	
ID	FC_MAIN_LOOP_1
Name	MAIN_LOOP
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

# VIC Flow 觸發運算子

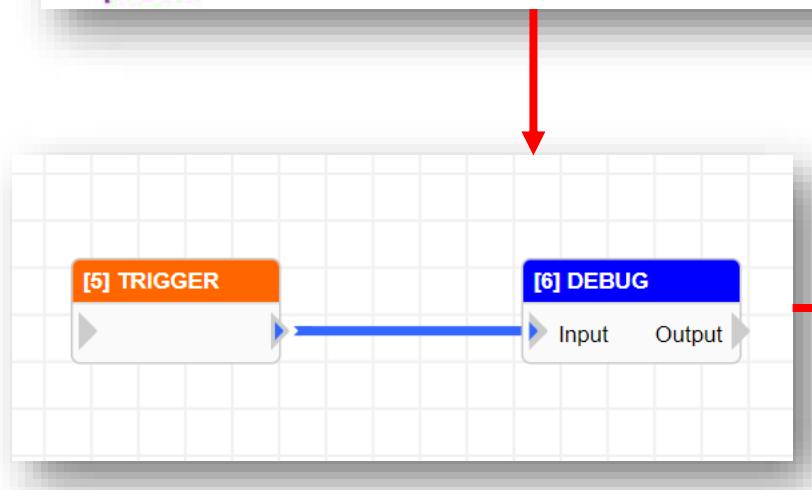
## • TRIGGER

可透過文字模式腳本的函式進行觸發的運算子，由TRIGGER\_OPERATOR(operator\_id)進行觸發  
若有設定回傳值，執行結束後會得到該回傳值

```
def mainLoop(info, arr):
```

```
    TRIGGER_OPERATOR('FC_TRIGGER_18')
```

```
    pass
```



### Settings

Name	Value
<b>Information</b>	
ID	FC_TRIGGER_18
Name	TRIGGER
Tag	
Priority	0

Apply successfully.

[134177] >>>> FC\_TRIGGER\_18 <<<<  
[134477] >>>> FC\_DEBUG\_19 <<<<

Trigger

# VIC Flow 觸發運算子

## • TRIGGER

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	顯示運算子的ID	
<b>名稱 (Name)</b>	顯示運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>別名 (Alias)</b>	設置該運算子的別名	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

Settings

Name	Value
<b>Information</b>	
ID	FC_TRIGGER_18
Name	TRIGGER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Alias	
<b>Fixed arr</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

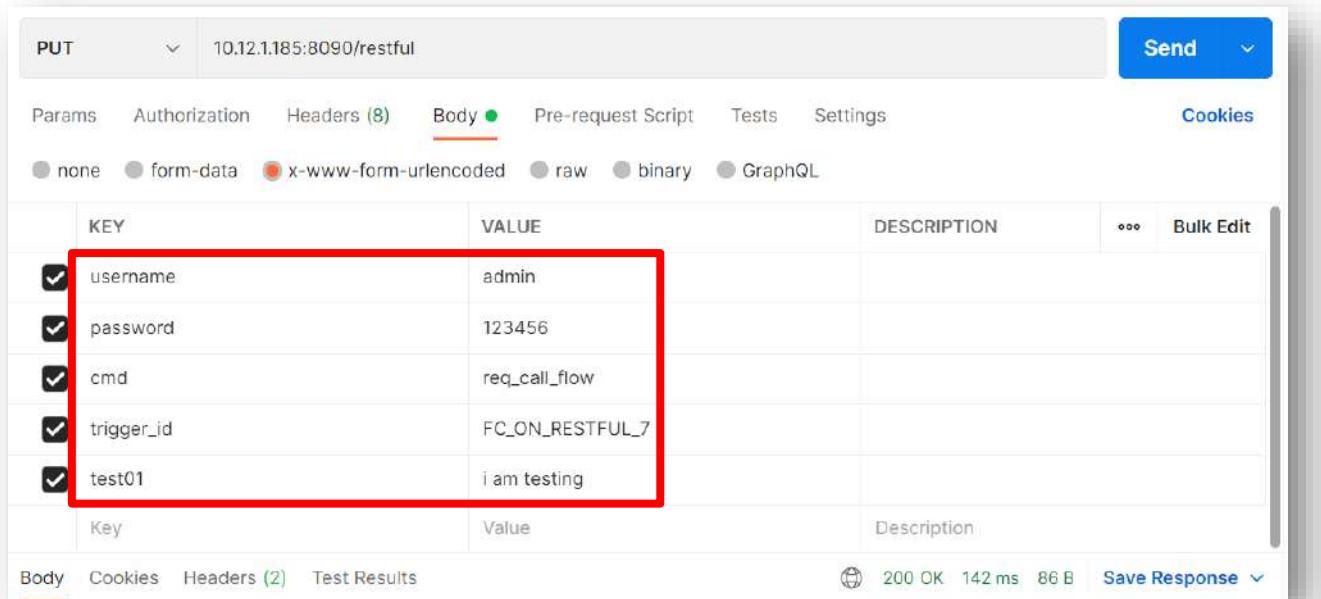
## • ON\_RESTFUL

可透過RESTful API觸發的運算子

若呼叫參數無trigger\_id，將會觸發全部的ON\_RESTFUL運算子

輸出端	UI名稱	行為
	json	輸出透過RESTful API傳送進VIC的參數 資料型態為JSON字串
	arr	輸出透過RESTful API傳送進VIC的圖像

參數名稱	參數內容
username	admin 或 user
password	123456(預設)
cmd	req_call_flow
trigger_id	運算子ID
param1_name (自定義參數名稱)	param1_val (自定義參數值)

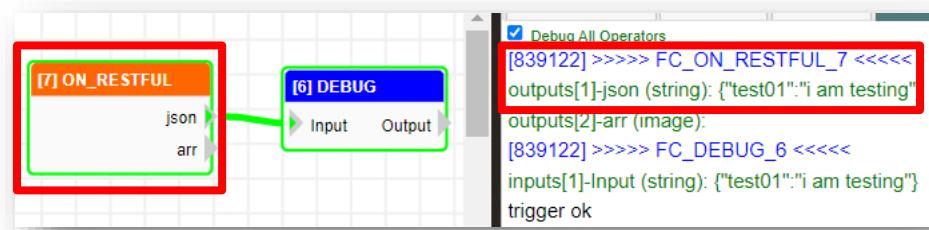


PUT 10.12.1.185:8090/restful

Params: Authorization: Headers (8) Body: Pre-request Script: Tests: Settings: Cookies: none: form-data: x-www-form-urlencoded: raw: binary: GraphQL: KEY VALUE DESCRIPTION Bulk Edit

KEY	VALUE	DESCRIPTION
username	admin	
password	123456	
cmd	req_call_flow	
trigger_id	FC_ON_RESTFUL_7	
test01	i am testing	

Body: Cookies: Headers (2) Test Results: 200 OK 142 ms 86 B Save Response



ON\_RESTFUL (7)

DEBUG (6)

Debug All Operators

[839122] >>>> FC\_ON\_RESTFUL\_7 <<<<  
outputs[1]-json (string): {"test01":"i am testing"}  
outputs[2]-arr (image):  
[839122] >>>> FC\_DEBUG\_6 <<<<  
inputs[1]-Input (string): {"test01":"i am testing"}  
trigger ok

# VIC Flow 觸發運算子

- ON\_RESTFUL

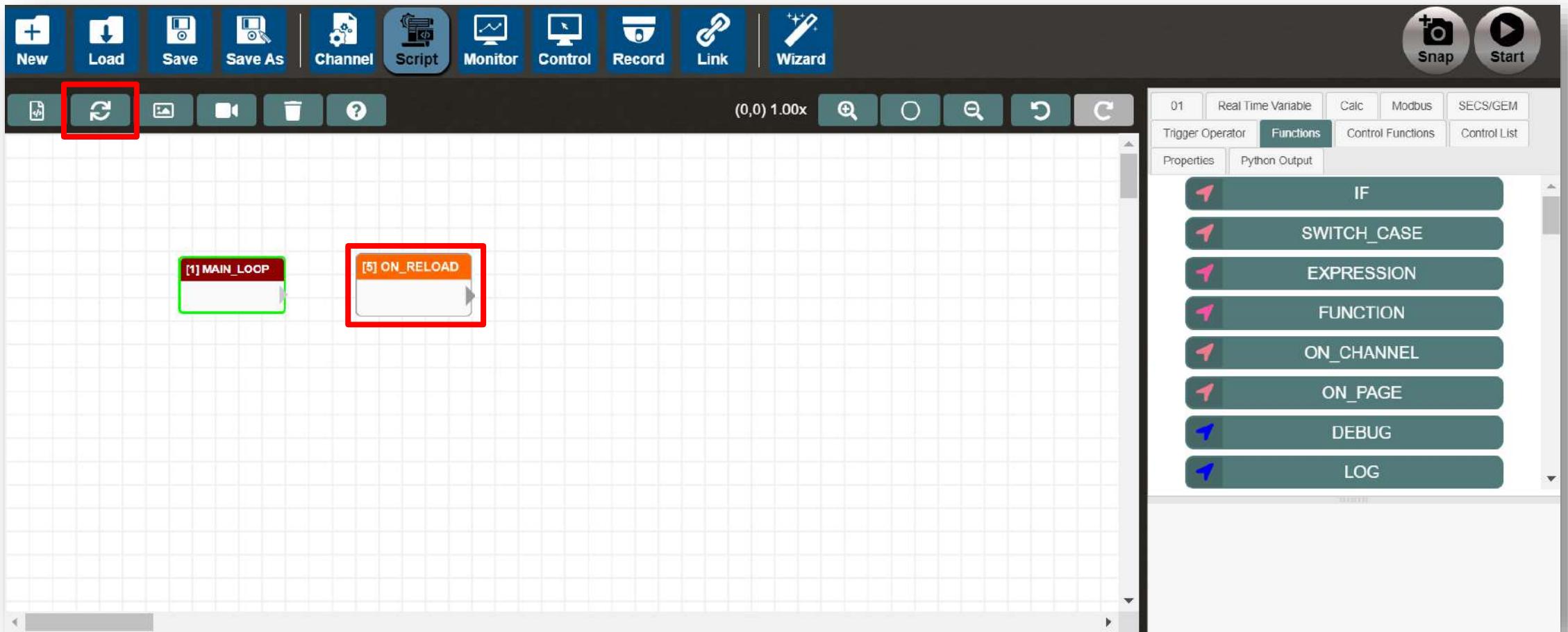
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

Settings	
Name	Value
<b>Information</b>	
ID	FC_ON_RESTFUL_32
Name	ON_RESTFUL
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

## • ON\_RELOAD

當點擊重新載入Python模組(工作插件)的按鈕或按下快捷鍵(F9)，即會觸發此運算子



# VIC Flow 觸發運算子

- **ON\_RELOAD**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

Settings	
Name	Value
<b>Information</b>	
ID	FC_ON_RELOAD_33
Name	ON_RELOAD
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

## • ON\_EVENT

當有事件發生時，即會觸發此運算子

觸發事件包含插拔擷取卡、偵測來源端訊號、來源端解析度改變...等等

	UI 名稱	行為	
輸出端		輸出觸發的事件	
	eventType	eventType	內容
		vic_event_capture_card_removed	拔除擷取卡
		vic_event_capture_card_detected	偵測到擷取卡
		vic_event_signal_removed	螢幕訊號被拔除
		vic_event_signal_not_detected	偵測不到螢幕訊號
		vic_event_format_changed	訊號改變
	info	輸出觸發該事件的資訊	



# VIC Flow 觸發運算子

## • ON\_EVENT

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	顯示運算子的ID	
<b>名稱 (Name)</b>	顯示運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

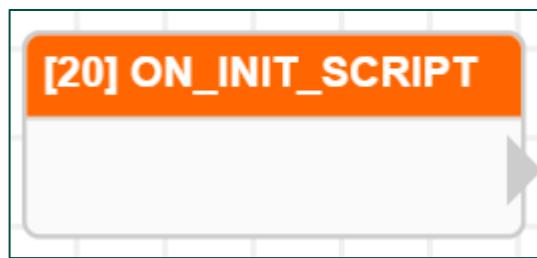
Settings

Name	Value
<b>Information</b>	
ID	FC_ON_EVENT_34
Name	ON_EVENT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

- **ON\_INIT\_SCRIPT**

當載入專案並初始化腳本完成時，即會觸發此運算子



# VIC Flow 觸發運算子

- **ON\_INIT\_SCRIPT**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

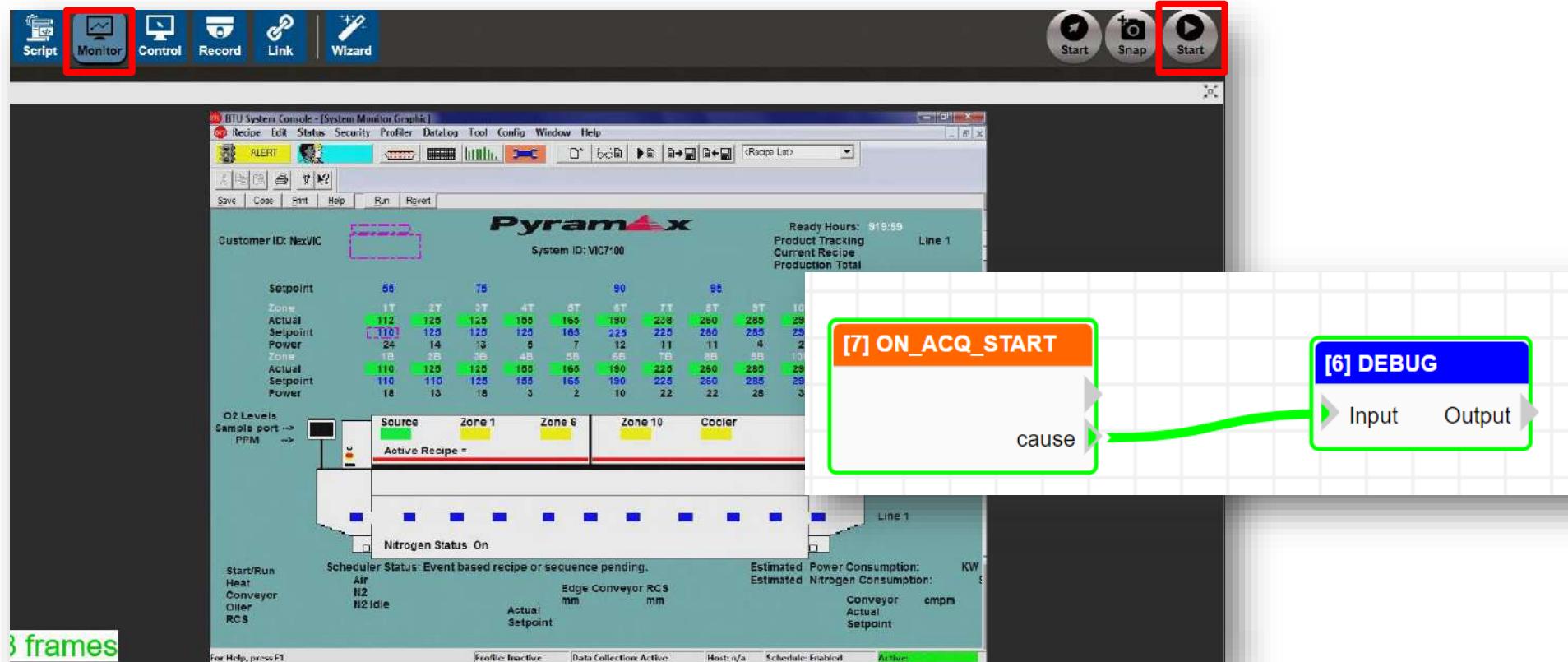
Settings	
Name	Value
<b>Information</b>	
ID	FC_ON_INIT_SCRIPT_35
Name	ON_INIT_SCRIPT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>

# VIC Flow 觸發運算子

## • ON\_ACQ\_START

除了控制頁面外，開始擷取圖像時，即會觸發此運算子

舉例來說，在監視頁面，開始擷取時，即會觸發此運算子



UI名稱	行為
輸出端	cause

# VIC Flow 觸發運算子

- ON\_ACQ\_START

名稱	意義	格式
<b>訊息 (Information)</b>		
ID	運算子的ID	
名稱 (Name)	運算子的名稱	
標籤 (Tag)	設定運算子的標籤	String
優先順序 (Priority)	設置同層級運算子的執行優先順序	Integer
略過 (Skip)	勾選後，即會略過執行此運算子	
偵錯 (Debug)	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
使用固定arr (Use Fixed arr)	勾選後，即使用設定的圖像(arr)	
固定arr (Fixed arr)	選擇要使用的圖像(arr)	

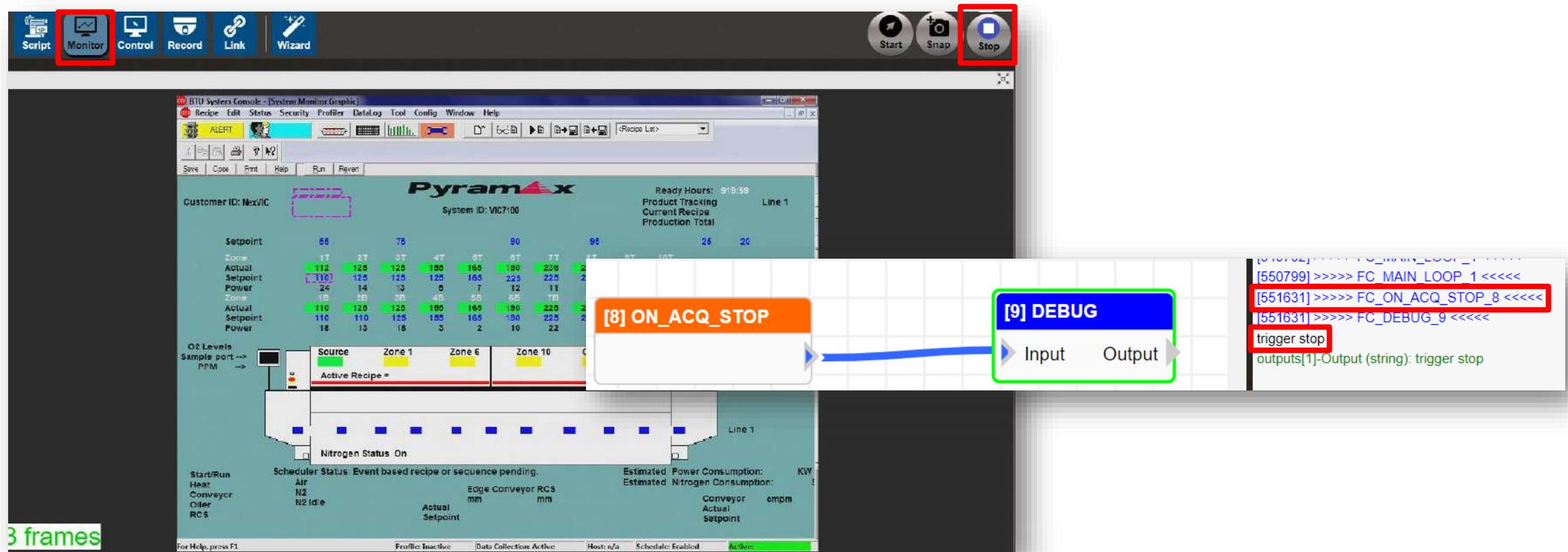
Settings	
Name	Value
<b>Information</b>	
ID	FC_ON_ACQ_START_31
Name	ON_ACQ_START
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

## • ON\_ACQ\_STOP

除了控制頁面外，停止擷取圖像時，即會觸發此運算子

舉例來說，在監視頁面，停止擷取時，即會觸發此運算子



# VIC Flow 觸發運算子

- ON\_ACQ\_STOP

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

Settings

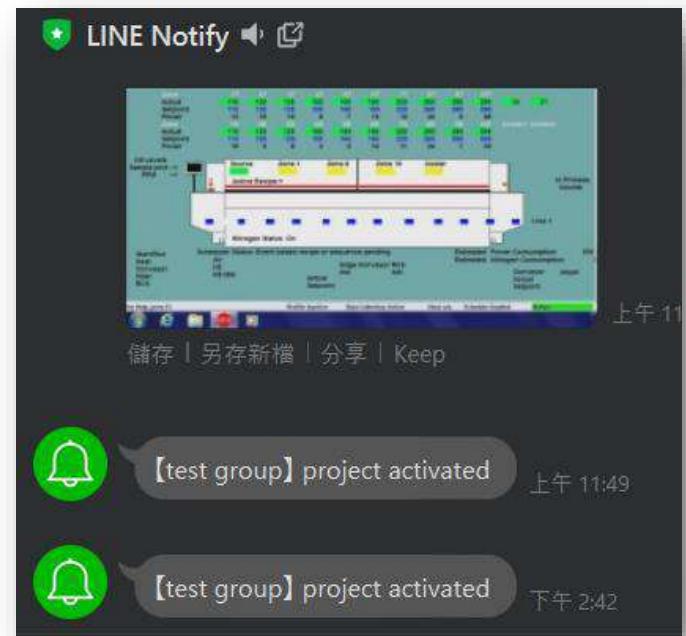
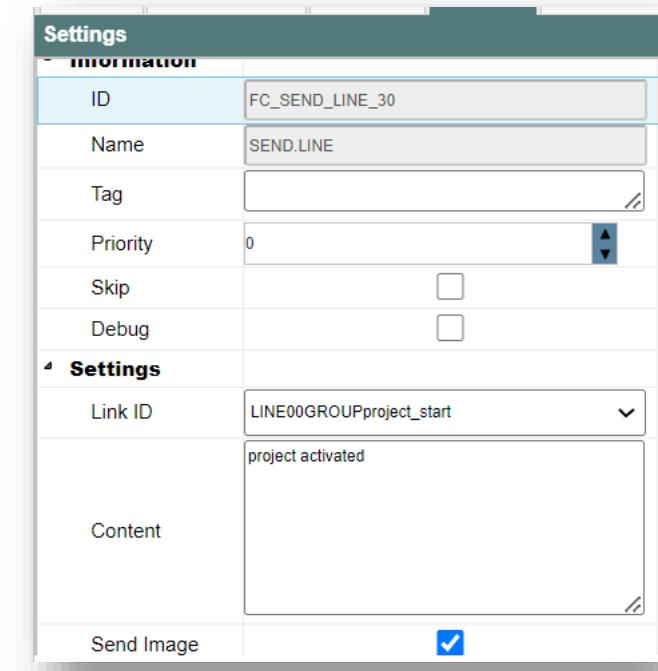
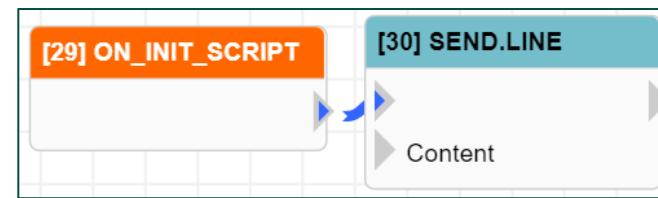
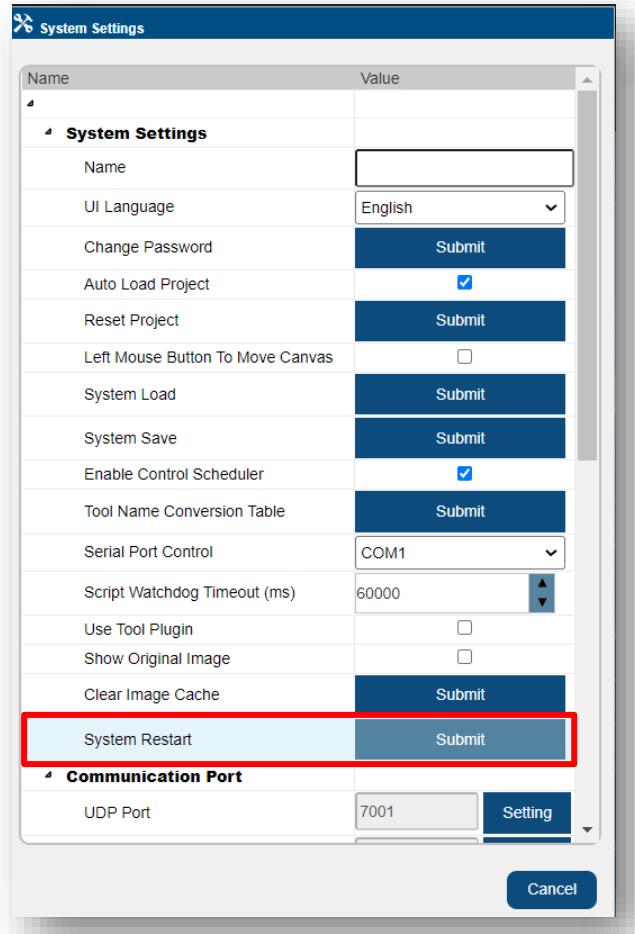
Name	Value
<b>Information</b>	
ID	FC_ON_ACQ_STOP_32
Name	ON_ACQ_STOP
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

## • ON\_INIT\_SYS

VIC主程式開啟、初始化完成、自動載入專案完成後，即會觸發此運算子

舉例來說，在系統設定視窗內點擊系統重啟，在VIC系統重啟後，即會觸發此運算子



# VIC Flow 觸發運算子

- **ON\_INIT\_SYS**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

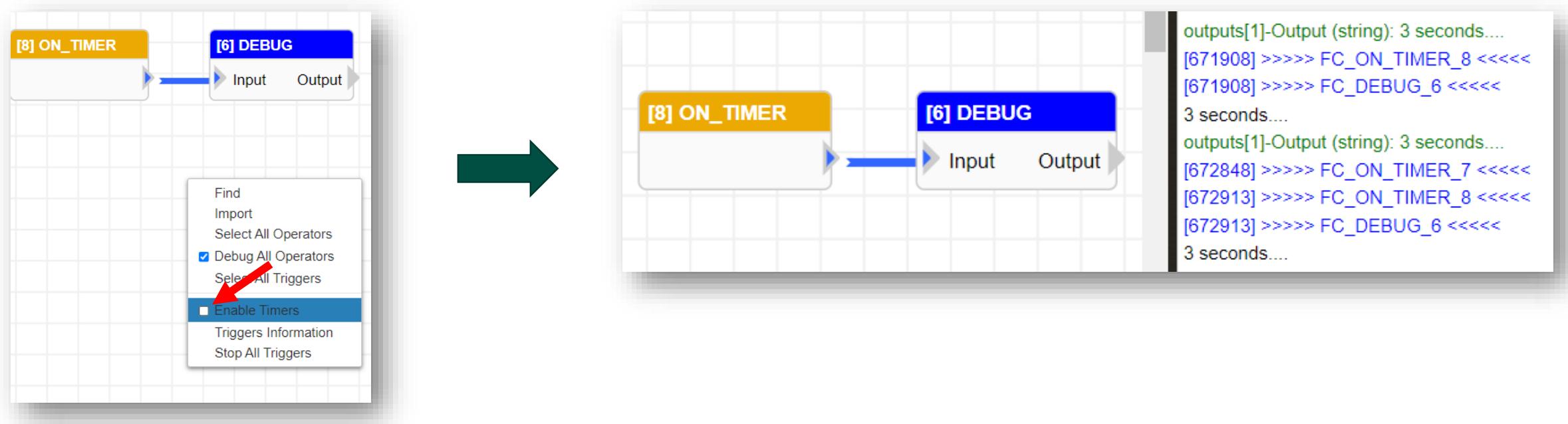
Settings	
Name	Value
<b>Information</b>	
ID	FC_ON_INIT_SYS_33
Name	ON_INIT_SYS
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

- **ON\_TIMER**

可以指定時間或是間隔時間進行觸發的運算子

在編輯區域中點擊右鍵，並選擇使用計時器，才會觸發此運算子



# VIC Flow 觸發運算子

## • ON\_TIMER

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>設置 (Settings)</b>		
<b>啟動 (Activate)</b>	勾選後，即會依據設置的時間進行觸發	
<b>更新 (Update)</b>	點擊後，即可進行觸發時間設置	
<b>方法 (Method)</b>	觸發時間的方法 分別有間隔(Interval)以及時刻(At an exact time)	
<b>秒 (Second)</b>	觸發的秒	
<b>分鐘 (Minute)</b>	觸發的分鐘	
<b>小時 (Hour)</b>	觸發的小時	
<b>日 (Day)</b>	觸發的日	
<b>星期 (Day Of Week)</b>	觸發的星期	
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

**Settings**

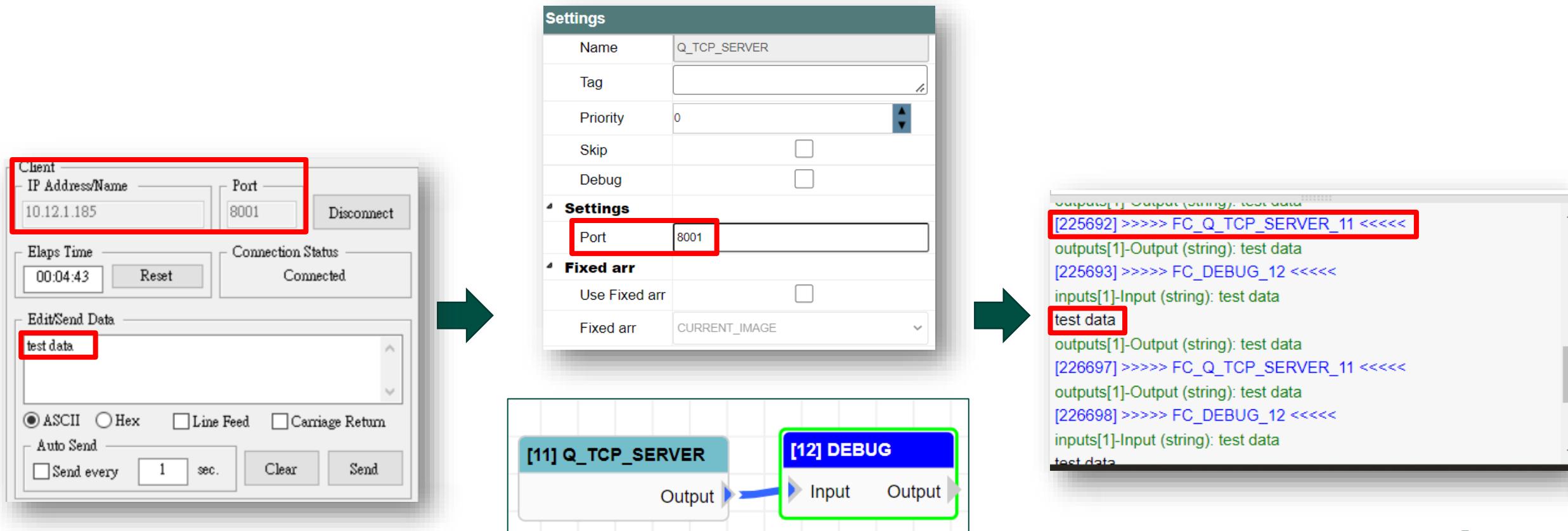
Name	Value
<b>Information</b>	
ID	FC_ON_TIMER_21
Name	ON_TIMER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Activate	<input type="checkbox"/>
Update	<b>Submit</b>
Method	Interval
Second	0
Minute	0
Hour	0
Day	0
Day Of Week	
<b>Fixed arr</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# VIC Flow 觸發運算子

## • Q\_TCP\_SERVER

建立一個TCP/IP Server，當接收到訊息時，即會觸發此運算子，並輸出接收到的訊息

舉例來說，使用TCP/IP測試工具的Client功能，即可傳送訊息到此運算子



# VIC Flow 觸發運算子

- Q\_TCP\_SERVER

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>通訊埠 (Port)</b>	設置TCP/IP Server的通訊埠 預設為8001	Integer
<b>固定arr (Fixed arr)</b>		
<b>使用固定arr (Use Fixed arr)</b>	勾選後，即使用設定的圖像(arr)	
<b>固定arr (Fixed arr)</b>	選擇要使用的圖像(arr)	

Settings

Name	Value
<b>Information</b>	
ID	FC_Q_TCP_SERVER_22
Name	Q_TCP_SERVER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Port	8001
<b>Fixed arr</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

# VIC Flow 函式

## • IF

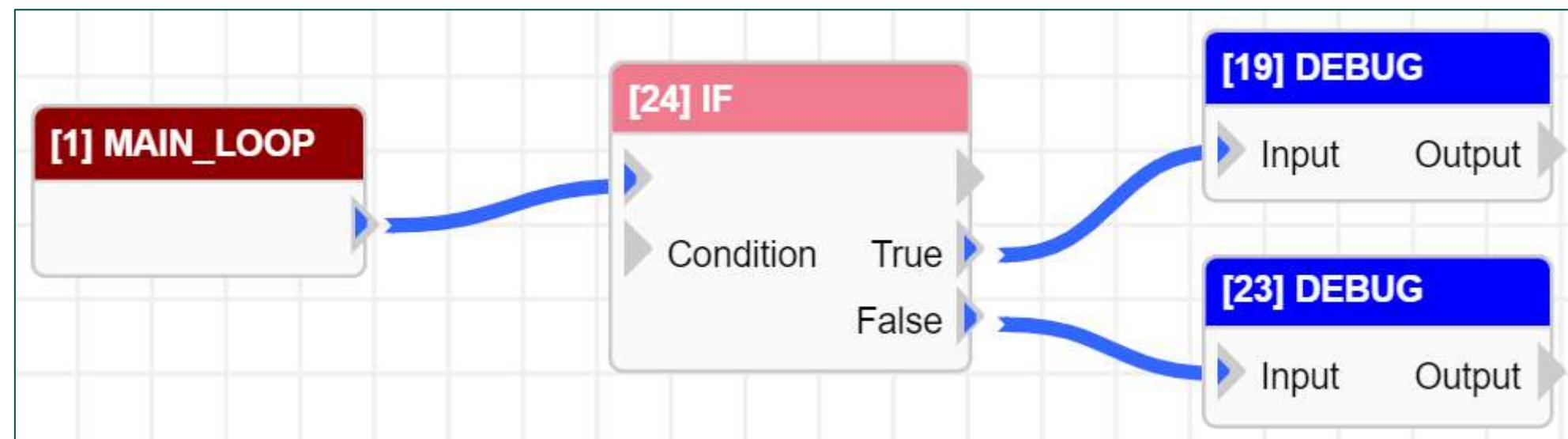
可判斷輸入的條件式，並將判斷結果進行對應輸出的運算子

若條件式為整數、字串，且不為空字串，則判斷結果皆為真(True)

輸出端為無值輸出

若條件(Condition)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值作為條件式

	UI名稱	行為
輸入端	Condition	輸入欲判斷的條件式
輸出端	True	判斷結果為真，則由此輸出
	False	判斷結果為否，則由此輸出



# VIC Flow 函式

- IF

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>條件 (Condition)</b>	設置欲判斷的條件式	Python Expression

Settings	
Name	Value
<b>Information</b>	
ID	FC_IF_36
Name	IF
Tag	/
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Condition	

## • SWITCH CASE

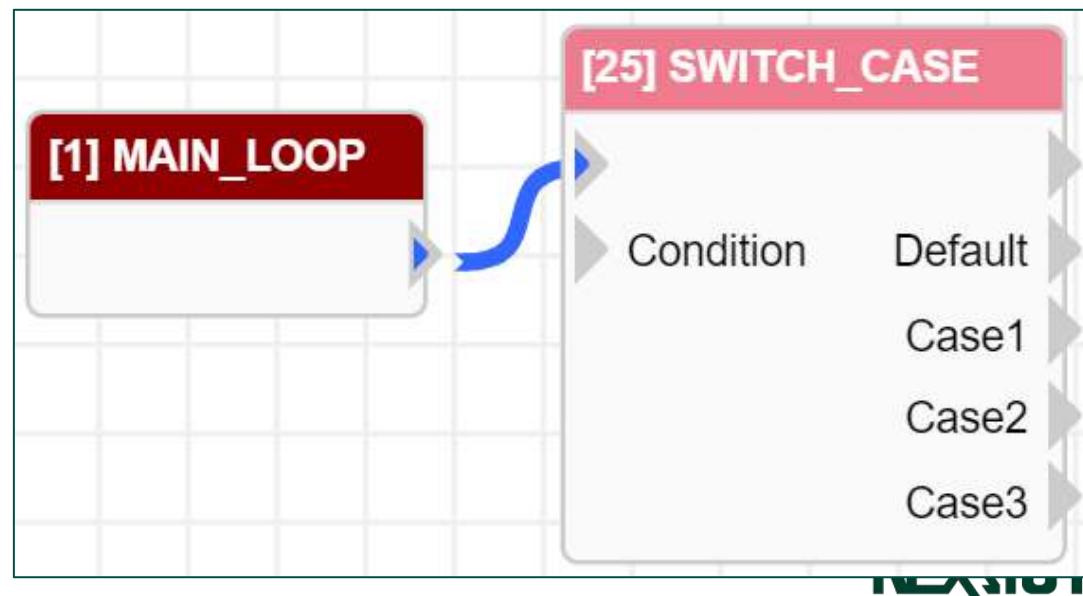
可將變數或運算式做為條件式並進行多重選擇的運算子，最多可設定10個Case

若條件式為整數、字串，且不為空字串，則判斷結果皆為真(True)

輸出端為無值輸出

若條件(Condition)的輸入端有值輸入，則運算子執行時會使用輸入端接收到的值作為條件式

	UI名稱	行為
輸入端	Condition	輸入欲判斷的條件
輸出端	Default	若無符合條件的Case，則由此輸出
	Case1	若符合Case1條件，則從此輸出
	Case2	若符合Case2條件，則從此輸出
	CaseX	若符合CaseX條件，則從此輸出



# VIC Flow 函式

## • SWITCH CASE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>條件 (Condition)</b>	設置欲判斷的條件式	Python Expression
<b>新增案例 (Add Case)</b>	點擊後，即會新增一個案例	
<b>案例1</b>	設置案例1的條件	String
<b>案例2</b>	設置案例2的條件	String
<b>案例X</b>	設置案例X的條件	String

**Settings**

Name	Value
<b>Information</b>	
ID	FC_SWITCH_CASE_25
Name	SWITCH_CASE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Condition	
Add Case	<b>Submit</b>
Case1	<input type="button" value="Remove"/>
Case2	<input type="button" value="Remove"/>
Case3	<input type="button" value="Remove"/>

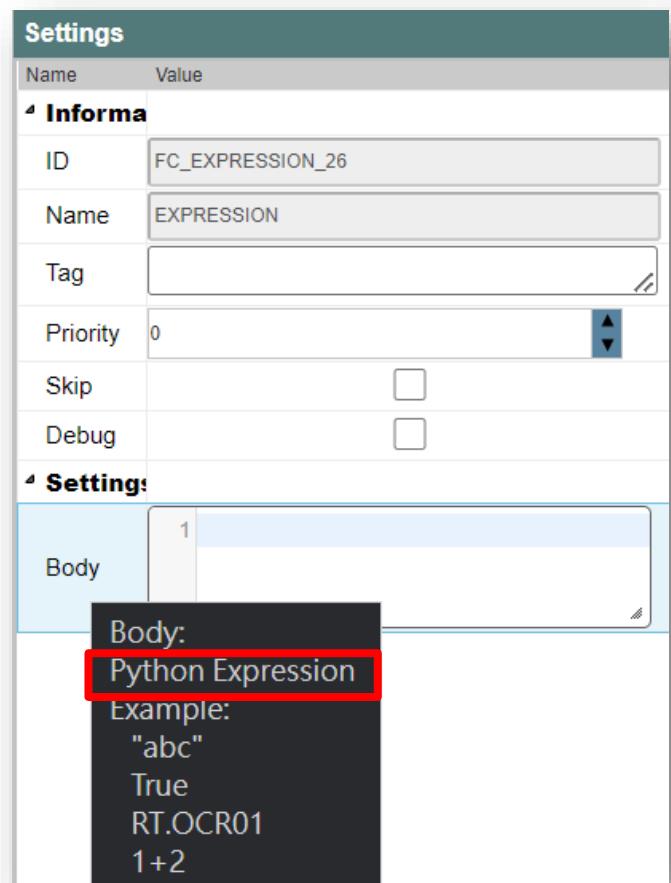
# VIC Flow 函式

## • EXPRESSION

可將程式內容(Body)解析成一組Python語句，並進行輸出的運算子

程式內容輸入格式為Python表達式

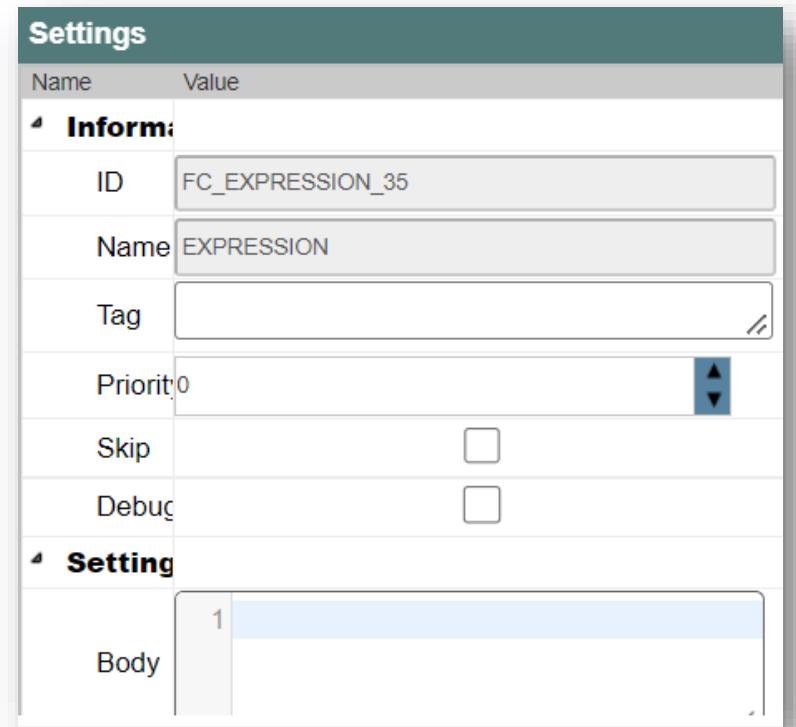
	UI名稱	行為
輸出端	Output	輸出執行結果



# VIC Flow 函式

## • EXPRESSION

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>程式內容 (Body)</b>	設置欲執行輸出的表達式內容	Python Expression



The screenshot shows the 'Settings' configuration for an Expression function. The configuration includes:

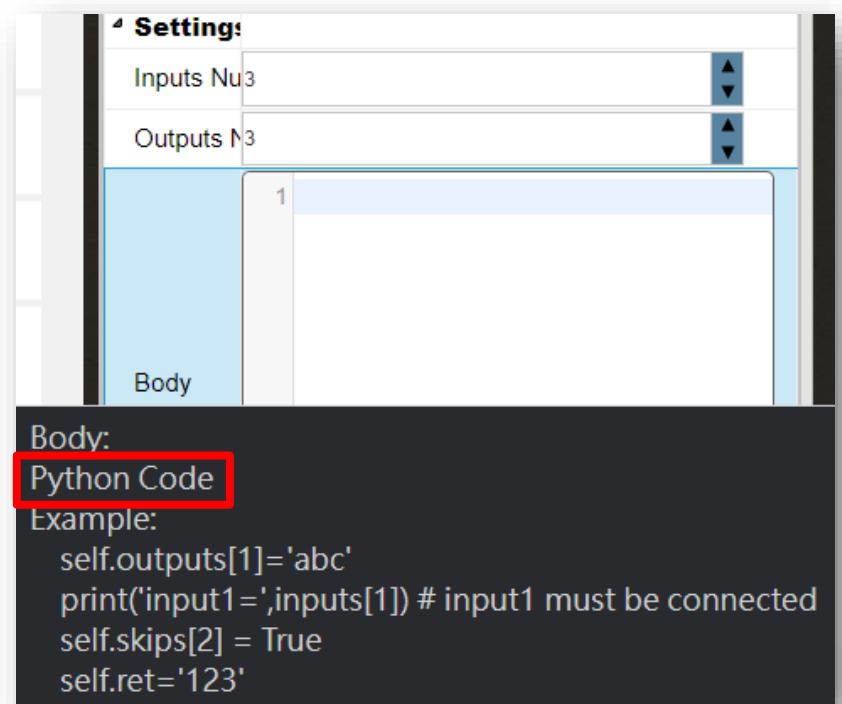
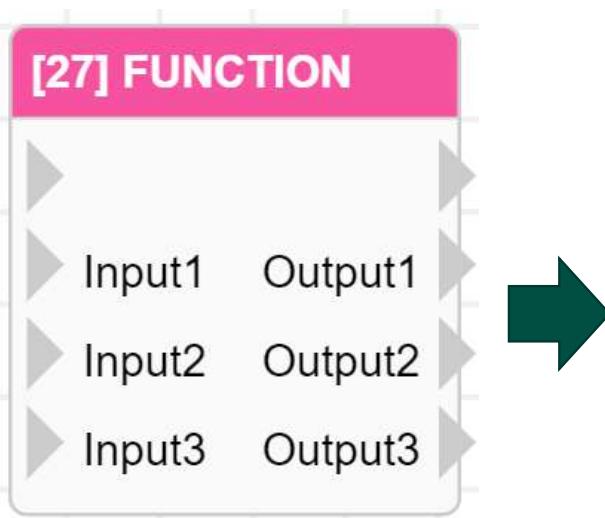
- ID:** FC\_EXPRESSION\_35
- Name:** EXPRESSION
- Tag:** (empty)
- Priority:** 0
- Skip:** (unchecked)
- Debug:** (unchecked)
- Body:** 1

## • FUNCTION

可自定義函式並執行的運算子，且可自行定義輸入端與輸出端數量的運算子

程式內容的輸入格式為Python程式

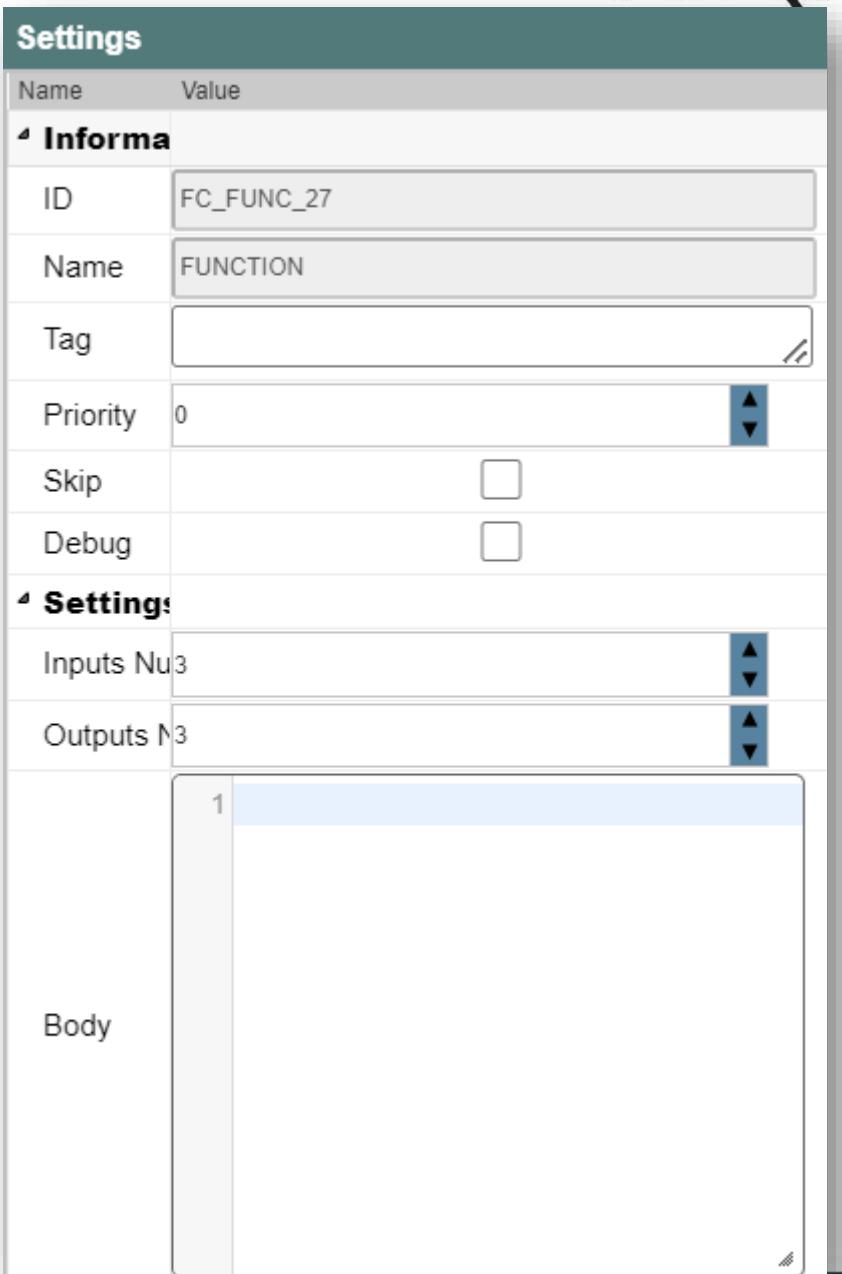
	UI名稱	行為
輸入端	Input1	輸入欲輸入到函式的輸入1
	Input2	輸入欲輸入到函式的輸入2
	InputX	輸入欲輸入到函式的輸入X
輸出端	Output1	由此輸出函式的輸出端1
	Output2	由此輸出函式的輸出端2
	OutputX	由此輸出函式的輸出端X



# VIC Flow 函式

## • FUNCTION

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入個數 (Inputs Number)</b>	設置輸入端的數量	Integer
<b>輸出個數 (Outputs Number)</b>	設置輸出端的數量	Integer
<b>程式內容 (Body)</b>	設置欲執行的程式內容	Python Code



## • FUNCTION

程式內容(Body)的編寫語法即為編寫Python程式碼，n為端點的編號，此編號為依序產生  
若在其他運算子使用\${ID}引用FUNCTION運算子，得到的值將會是該運算子設置的self.ret的值

使用參數	引用輸入端值	設置輸出端值	設置回傳值	設置略過單一輸出端
呼叫方法	語法為 input[n] Ex : input [3] 即可引用第3個輸入端的值	語法為 self.output[n] Ex : self.output[3] 即為設置第3輸出端的值	語法為 self.ret=value · value 即為要回傳的值 取得該運算子回傳的值的語法為 \${ID} Ex : \${運算子ID}	設置忽略輸出端的語法為 self.skip[n]=True · 預設為false n 為輸出端編號 Ex : self.skip[3]=True 即可略過第3個輸出端

```

1 print(inputs[2])
2 Input2_val = inputs[2]
3 self.skips[3] = True
4 self.outputs[1] = Input2_val + 10
5 self.ret = Input2_val + 20

```

Body

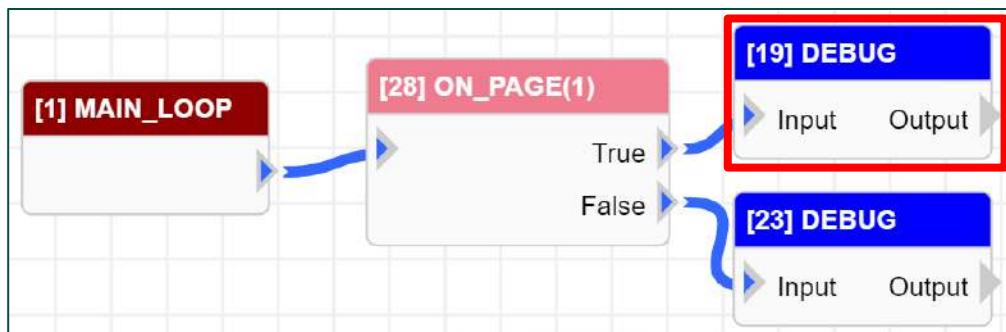
左圖程式範例內容：

- 打印第二個輸入端的輸入值
- 忽略第三個輸出端
- 將第二個輸入端的輸入值加上10，作為第一個輸出端數值
- 將運算子回傳值設置為第二個輸入端的參數值加上20

## • ON\_PAGE

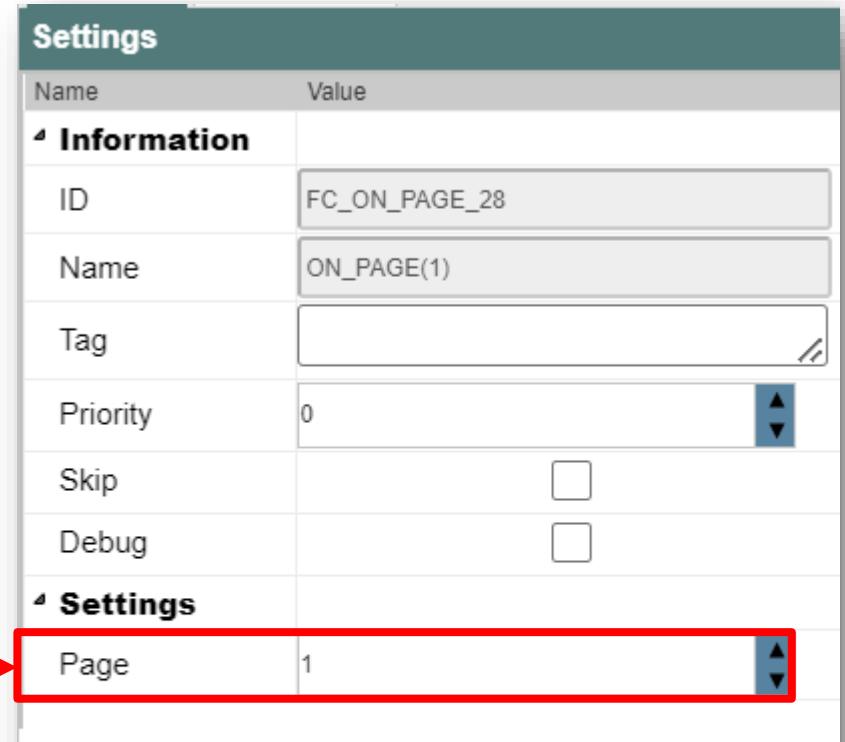
可設置指定頁面的運算子，若當下辨識的頁面為指定的頁面，則由真(True)輸出，反之則由否(False)輸出，輸出端無數值輸出

	UI名稱	行為
輸入端		
輸出端	True	若辨識當下的頁面為設置的頁面，則從由輸出
	False	若辨識當下的頁面不是設置的頁面，則由此輸出



Status

PAGE\_01



[206819] >>>> FC\_MAIN\_LOOP\_1 <<<<

[207119] >>>> FC\_ON\_PAGE\_28 <<<<

[207419] >>>> FC\_DEBUG\_19 <<<<

On Page1

outputs[1]-Output (string): On Page1

# VIC Flow 函式

- **ON\_PAGE**

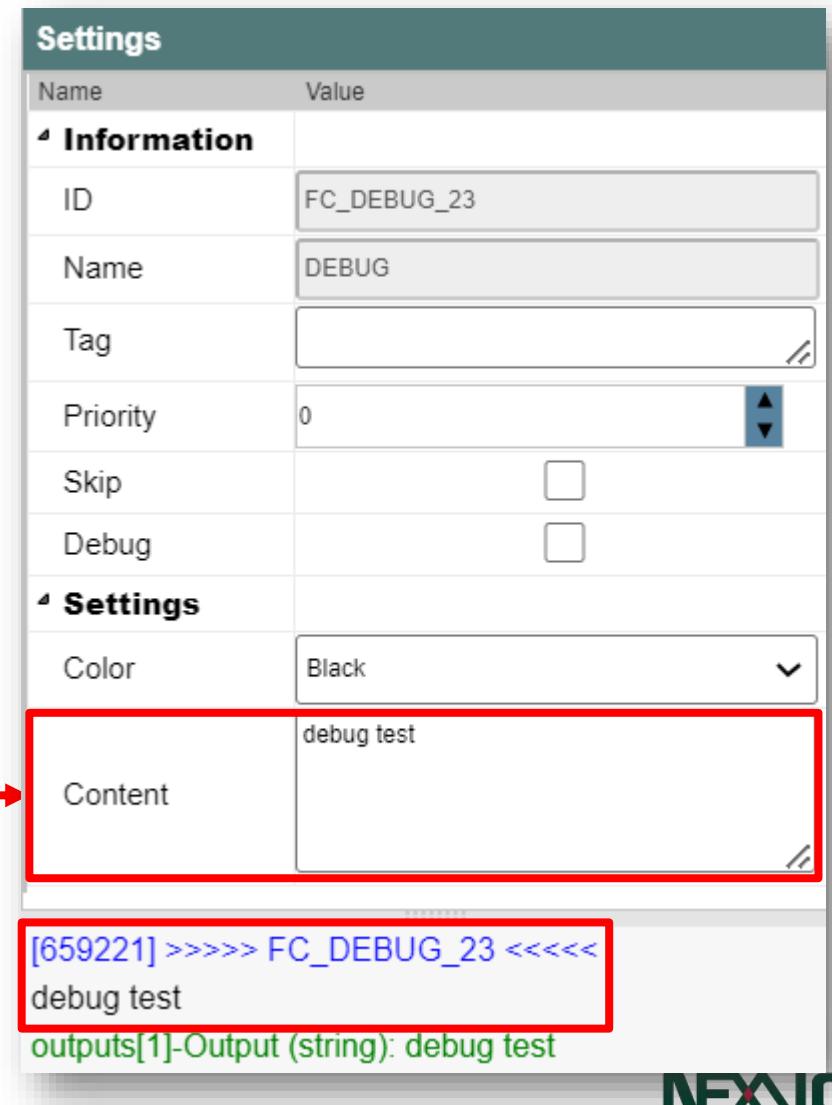
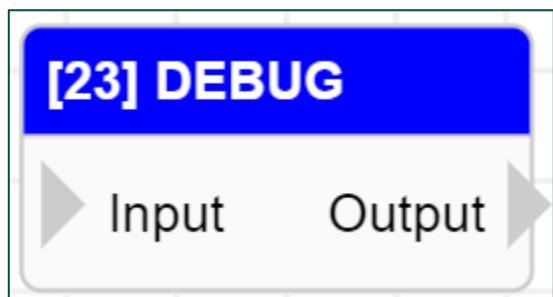
名稱	意義	格式
<b>訊息(Information )</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤(Tag)</b>	設定運算子的標籤	String
<b>優先順序(Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過(Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯(Debug )</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>頁面(Page)</b>	設置指定頁面	Integer

Settings	
Name	Value
<b>Information</b>	
ID	FC_ON_PAGE_28
Name	ON_PAGE(1)
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Page	1

## • DEBUG

可將輸入端或設置的內容打印在Python輸出欄位中的運算子

	UI名稱	行為
輸入端	Input	輸入欲打印的內容
輸出端	Output	輸出打印的內容



# VIC Flow 函式

- DEBUG

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>顏色 (Color)</b>	設置打印文字的顏色	
<b>內容 (Content)</b>	設置欲打印的內容 若此欄位不為空，則會套用欄位設定的值 若設置%1，代表打印該運算子的ID	Inline Python Expression String and %1

**Settings**

Name	Value
<b>Information</b>	
ID	FC_DEBUG_23
Name	DEBUG
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Color	Black
Content	

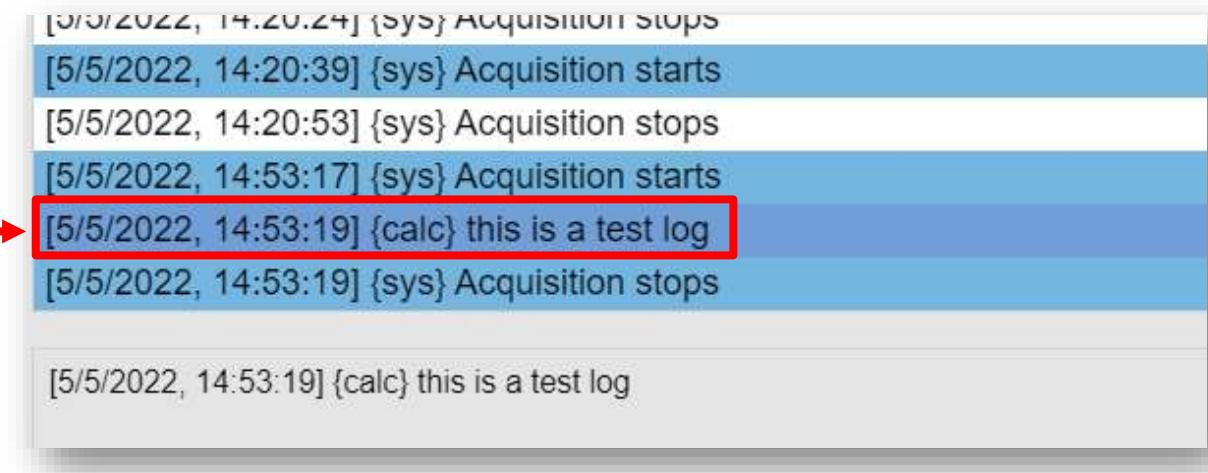
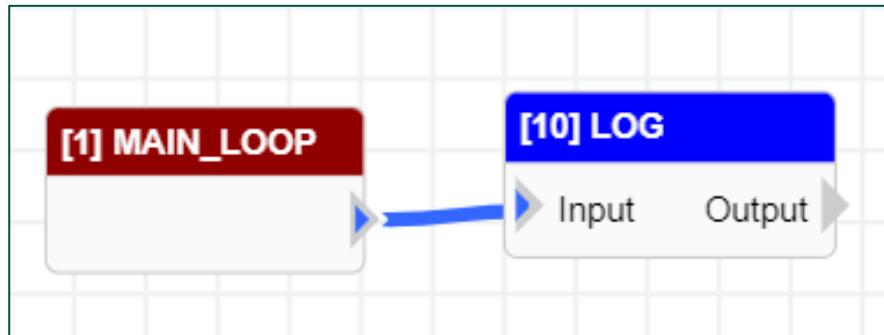
# VIC Flow 函式



- **LOG**

可將輸入端或設置的內容記載至系統日誌中的運算子，可在系統日誌內查詢到紀錄的訊息

	UI名稱	行為
輸入端	Input	輸入欲記載的內容
輸出端	Output	輸出記載的內容



## • LOG

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>顏色 (Color)</b>	設置記載內容的字元顏色	
<b>模組 (Module)</b>	欲記載的模組類別 若此欄位為空，則會套用預設值calc	String
<b>內容 (Content)</b>	設置欲記載的內容 若此欄位不為空，則會記載此欄位設定的值 若此欄位為空，則會記載輸入端的值	Inline Python Expression String and %1
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

Settings	
Name	Value
<b>Information</b>	
ID	FC_LOG_29
Name	LOG
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Color	Black
Module	
Content	
Block Time (sec.)	10

# VIC Flow 函式

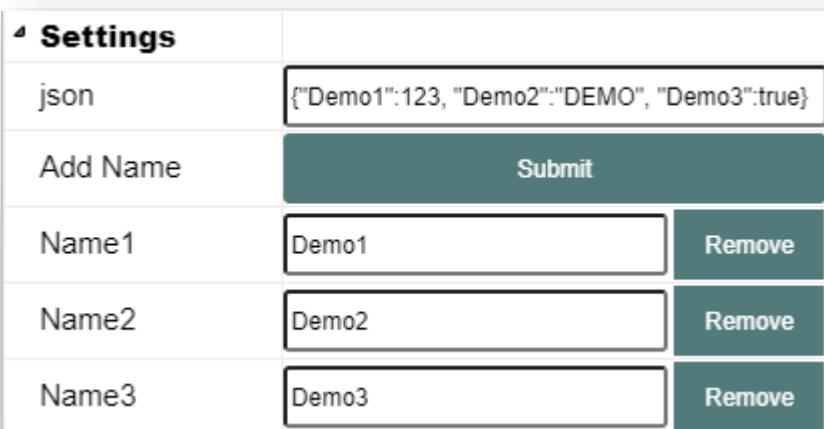
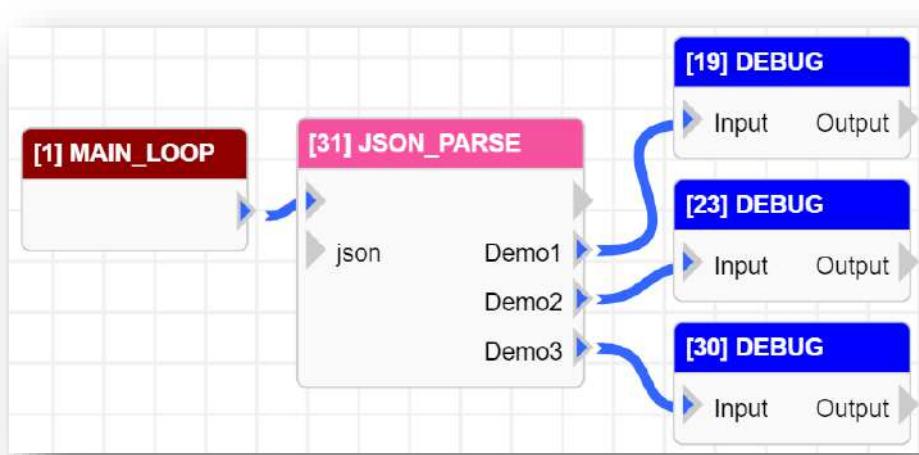
## • JSON\_PARSE

可將JSON字串進行轉換解析的運算子，最多可新增100個名稱(Name)欄位

假設傳入的JSON字串為 {"Demo1":123, "Demo2":"DEMO", "Demo3":true}

利用JSON\_PARSE運算子進行轉換，並利用DEBUG運算子其解析的值打印在Python輸出中

	UI名稱	行為
輸入端	json	輸入欲解析的JSON字串
輸出端		



```
[101133] >>>> FC_MAIN_LOOP_1 <<<<
[101433] >>>> FC_JSON_PARSE_31 <<<<
outputs[2]-Name1 (integer): 123
outputs[3]-Name2 (string): DEMO
outputs[4]-Name3 (bool): true
[101734] >>>> FC_DEBUG_30 <<<<
inputs[1]-Input (bool): true
true
outputs[1]-Output (string): true
[102034] >>>> FC_DEBUG_23 <<<<
inputs[1]-Input (string): DEMO
DEMO
outputs[1]-Output (string): DEMO
[102334] >>>> FC_DEBUG_19 <<<<
inputs[1]-Input (integer): 123
123
outputs[1]-Output (string): 123
```

# VIC Flow 函式

## • JSON\_PARSE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>json</b>	設置欲解析的JSON字串 若輸入端有值，則套用由輸入端輸入的值	String
<b>新增名稱 (Add Name)</b>	點擊後，即新增一個Name	
<b>名稱1 (Name1)</b>	名稱1	String
<b>名稱2 (Name2)</b>	名稱2	String
<b>名稱X (NameX)</b>	名稱X	String

**Settings**

Name	Value
<b>Information</b>	
ID	FC_JSON_PARSE_31
Name	JSON_PARSE
Tag	/
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
json	
Add Name	<b>Submit</b>

# VIC Flow 函式

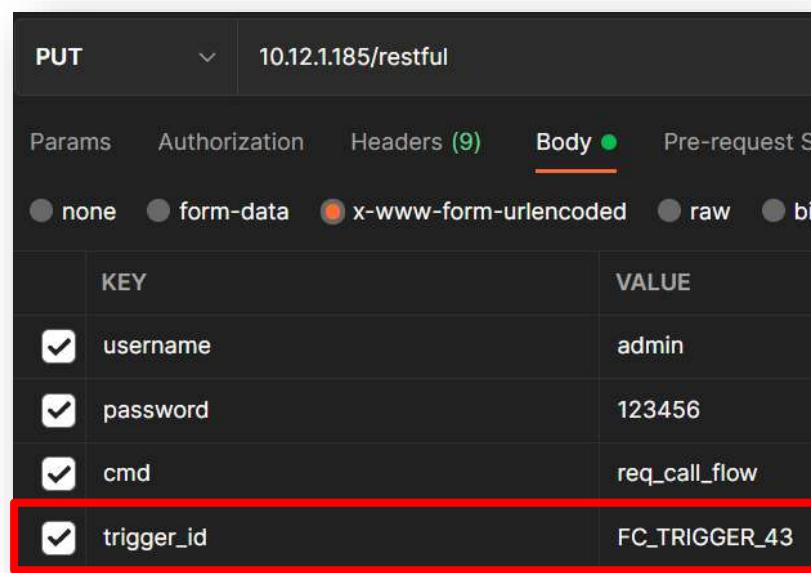
## • JSON\_RET

可設置回傳的JSON物件的運算子，最多可新增100個名稱(Name)欄位

放置於觸發運算子執行程序的最後

即為觸發運算子和子程序執行結束後的回傳內容

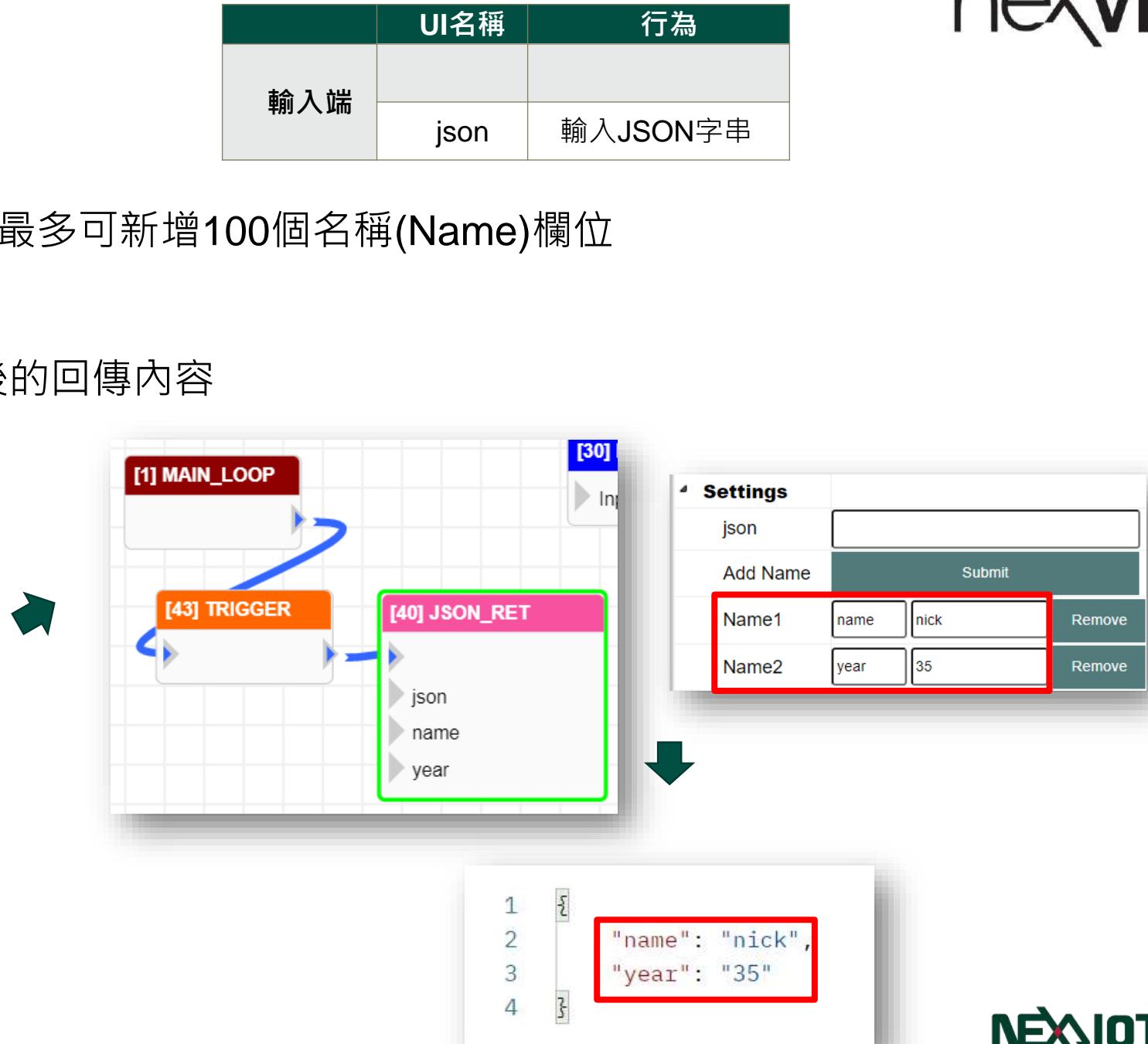
回傳格式為 json



PUT 10.12.1.185/restful

Params Authorization Headers (9) Body (x-www-form-urlencoded)

KEY	VALUE
username	admin
password	123456
cmd	req_call_flow
trigger_id	FC_TRIGGER_43



## • JSON\_RET

在運算子的設置中，可以設置回傳的JSON物件內容的名稱和數值

若數值欄位有值，系統會以欄位中的數值為主，忽略名稱輸入端的值

**Settings**

json			
Add Name	Submit		
Name1	name	nick	Remove
Name2	year	35	Remove

名稱 (string) 數值 (string)



**Settings**

json	{ "Demo1":123, "Demo2":"DEMO", "Demo3":true }		
Add Name	Submit		
Name1			Remove
Name2			Remove

# VIC Flow 函式

- **JSON\_RET**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級執行運算子的優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯(Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>Json</b>	直接設置JSON並字串進行回傳 若輸入端有輸入值，則會套用輸入端的輸入值 系統會忽略運算子中設定的名稱和值	String
<b>新增名稱 (Add Name)</b>	點擊後，即會新增一個名稱	
<b>名稱1 (Name1)</b>	名稱1	String
<b>名稱2 (Name2)</b>	名稱2	String
<b>名稱X (NameX)</b>	名稱X	String

**Settings**

Name	Value
<b>Information</b>	
ID	FC_JSON_RET_32
Name	JSON_RET
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
json	
Add Name	<b>Submit</b>

# VIC Flow 函式

## • JSON\_STRING

可將輸入端或設置的內容轉換為JSON字串的運算子

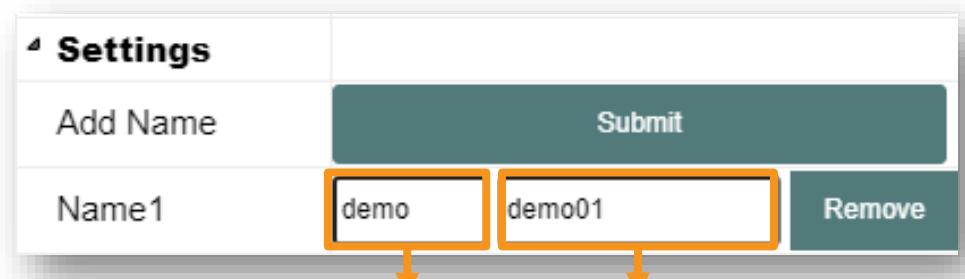
在運算子的設置中，可以設置欲回傳的JSON物件內容的名稱和數值

最多可新增100個名稱(Name)欄位

若數值欄位有值，系統會以欄位中的數值為主，忽略名稱輸入端的值

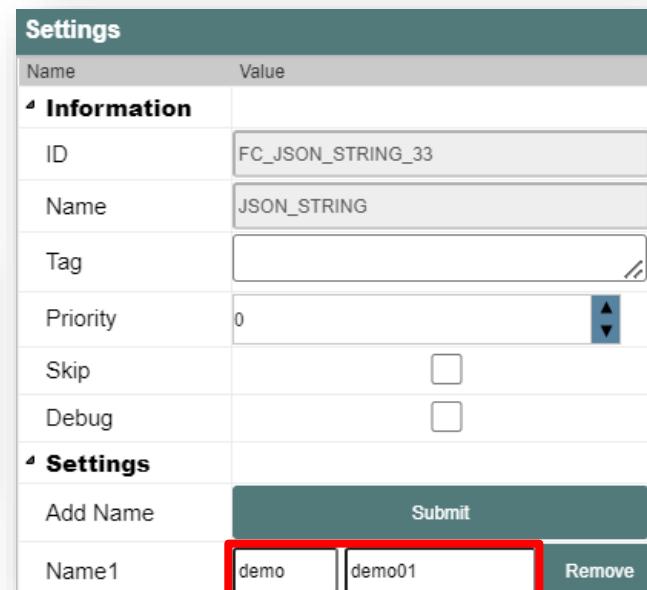


	UI名稱	行為
輸入端		
輸出端	json	輸出JSON字串



Add Name	Submit		
Name1	demo	demo01	Remove

名稱  
(string)      數值  
(string)



Add Name	Submit		
Name1	demo	demo01	Remove

```
[105891] >>>> FC_JSON_STRING_33 <<<<
outputs[2]-json (string): {"demo":"demo01"}
[106192] >>>> FC_DEBUG_34 <<<<
inputs[1]-Input (string): {"demo":"demo01"}
{"demo":"demo01"}
outputs[1]-Output (string): {"demo":"demo01"}
```

- **JSON\_STRING**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>新增名稱 (Add Name)</b>	點擊後，即會新增一個名稱(Name)	
<b>名稱1 (Name1)</b>	名稱1	String
<b>名稱2 (Name2)</b>	名稱2	String
<b>名稱X (NameX)</b>	名稱X	String

**Settings**

Name	Value
ID	FC_JSON_STRING_43
Name	JSON_STRING
Tag	/
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

**Settings**

Add Name	Submit
----------	--------

- **COMMENT**

可設置備註於編輯區域中的運算子

備註內容須設置於該運算子的標籤欄位中



Settings	
Name	Value
<b>Information</b>	
ID	FC_COMMENT_35
Name	COMMENT
Tag	Comment Test

# VIC Flow 函式

## • COMMENT

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>寬度 (Width)</b>	運算子於介面顯示的寬度	Integer
<b>顏色 (Color)</b>	顯示的字型顏色	
<b>字型大小 (Font Size)</b>	顯示的字型大小	Integer

Settings

Name	Value
<b>Information</b>	
ID	FC_COMMENT_35
Name	COMMENT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Width	200
Color	Red
Font Size	16

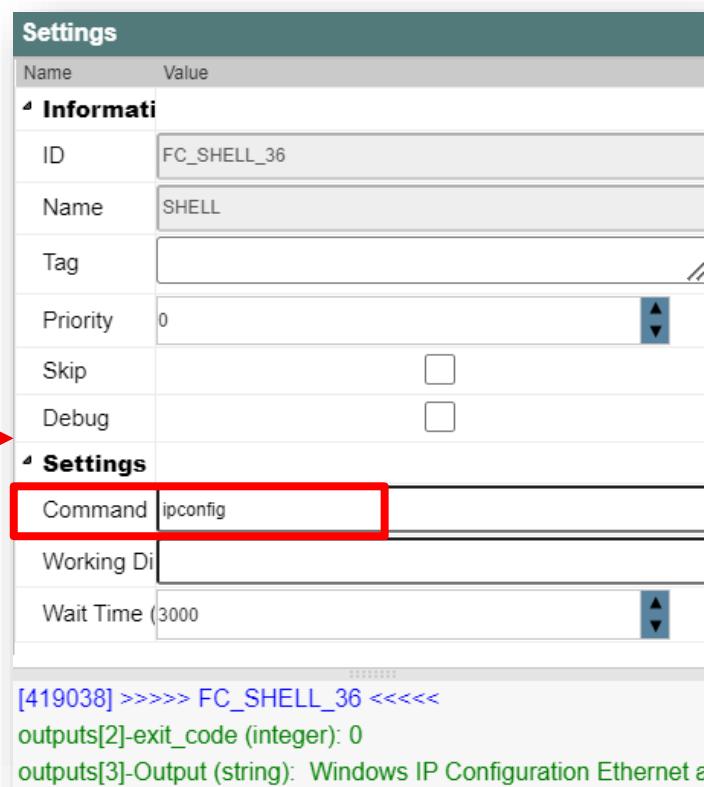
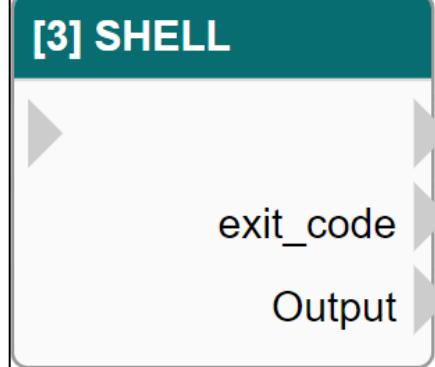
# VIC Flow 函式

## • SHELL

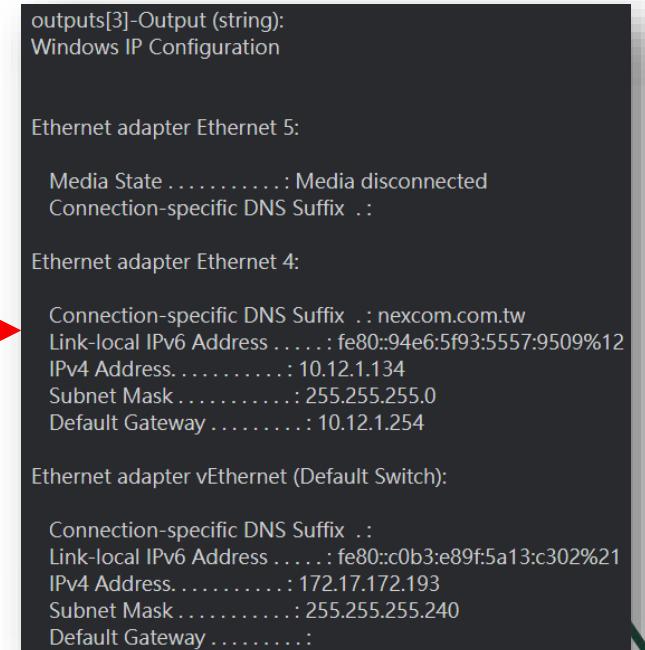
可透過該運算子在VIC本機上執行命令行(command line)的運算子

Command = 欲執行的指令

Working Dictionary = 執行命令行的工作目錄



	UI名稱	行為
輸入端		
輸出端	exit_code	輸出執行結束後的退出碼
	Output	輸出執行結果



The output text area displays the results of the 'ipconfig' command:

```
outputs[3]-Output (string): Windows IP Configuration Ethernet a.

Ethernet adapter Ethernet 5:
Media State ..... : Media disconnected
Connection-specific DNS Suffix .:

Ethernet adapter Ethernet 4:
Connection-specific DNS Suffix . : nexcom.com.tw
Link-local IPv6 Address . . . . . : fe80::94e6:5f93:557:9509%12
IPv4 Address. .... . . . . : 10.12.1.134
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.12.1.254

Ethernet adapter vEthernet (Default Switch):
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::c0b3:e89f:5a13:c302%21
IPv4 Address. .... . . . . : 172.17.172.193
Subnet Mask . . . . . : 255.255.255.240
Default Gateway . . . . . :
```

# VIC Flow 函式

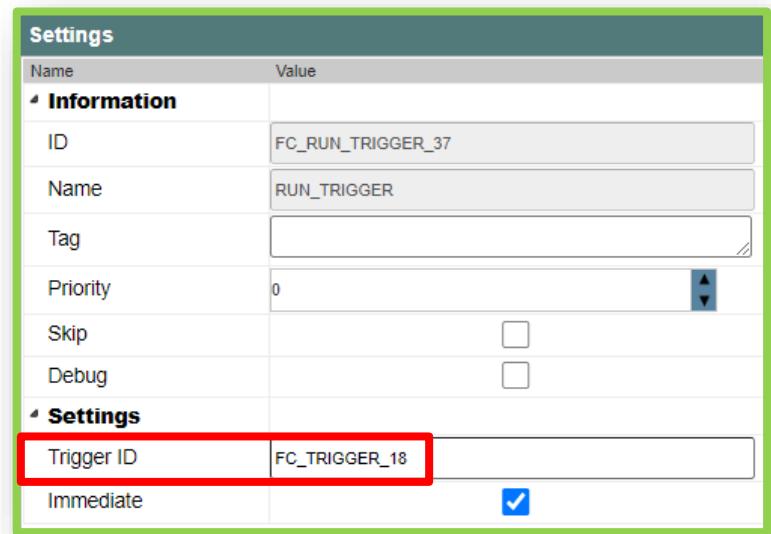
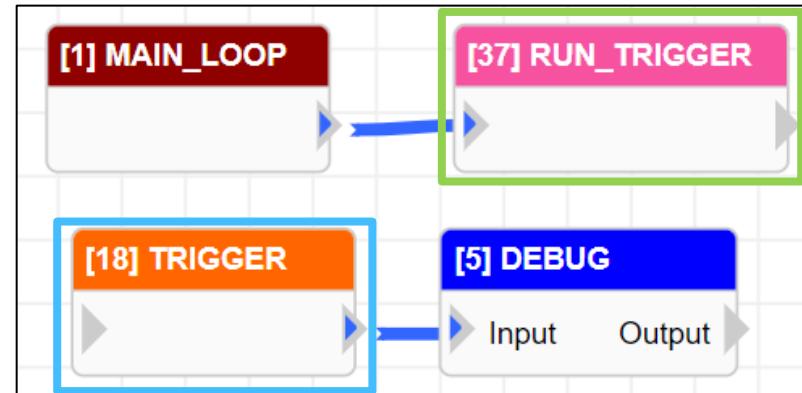
- SHELL

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>指令 (Command)</b>	欲執行的指令	String
<b>工作目錄 (Working Directory)</b>	欲執行指令的工作目錄	String
<b>等待時間(ms) (Wait Time (ms))</b>	下次執行的等待時間	Integer

Settings	
Name	Value
▲ <b>Information</b>	
ID	FC_SHELL_36
Name	SHELL
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
▲ <b>Settings</b>	
Command	
Working Di	
Wait Time (3000	

- **RUN\_TRIGGER**

可執行觸發運算子的運算子



```

[490783] >>>> FC_MAIN_LOOP_1 <<<<
[491084] >>>> FC_RUN_TRIGGER_37 <<<<
[491084] >>>> FC_TRIGGER_18 <<<<
[491384] >>>> FC_DEBUG_5 <<<<
Trigger
outputs[1]-Output (string): Trigger
  
```

# VIC Flow 函式

## • RUN\_TRIGGER

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>觸發ID (Trigger ID)</b>	設置要執行的觸發運算子的ID	String
<b>立即 (Immediate)</b>	若勾選，系統會立即執行設置的觸發運算子和它的子運算子	

Settings	
Name	Value
<b>Information</b>	
ID	FC_RUN_TRIGGER_37
Name	RUN_TRIGGER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Trigger ID	FC_TRIGGER_18
Immediate	<input checked="" type="checkbox"/>

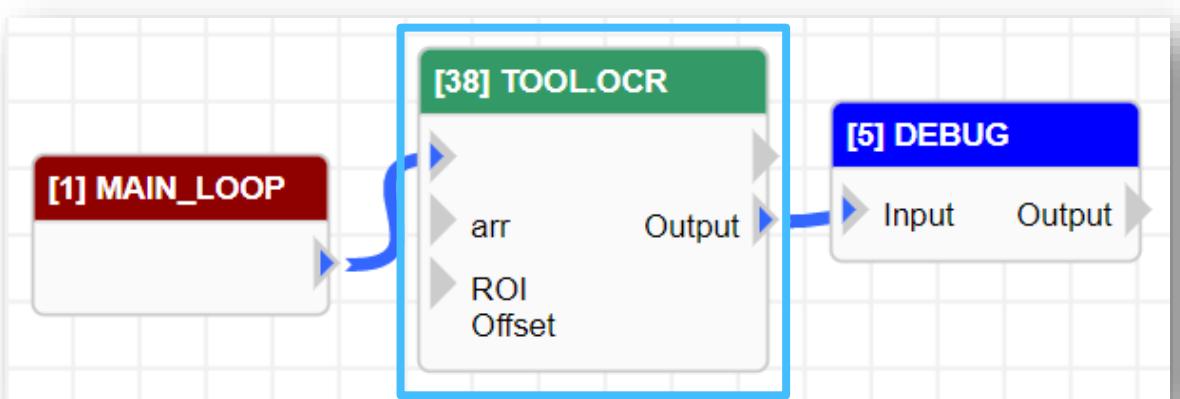
# VIC Flow 函式

## • TOOL.OCR

可自行設置辨識圖檔及和辨識參數並進行文字辨識的運算子

在運算子設置內可調整文字辨識的所有參數

	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用系統當下的辨識圖像進行辨識
	ROI Offset	輸入ROI需偏移的位移量 格式為offset_x, offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	Output	輸出辨識結果



Settings

Input Parameters

- ROI-X: 0
- ROI-Y: 0
- ROI-Width: 0
- ROI-Height: 0
- Update: Submit

ROI X Offset:

ROI Y Offset:

arr: CURRENT\_IMAGE

White List:

Allow Empty String:

Remove Whitespace:

Allow Recognition Rate(%): 80

Image Preprocess

Resize

- Resize Method: Nearest
- Resize: x1

Threshold

- Threshold Method: Binary
- Threshold Algorithm: OTSU
- Threshold Value: 120

OCR Font

- Segmentation Mode: Single Line
- Select Font: English
- Load Font: Submit
- Font File:

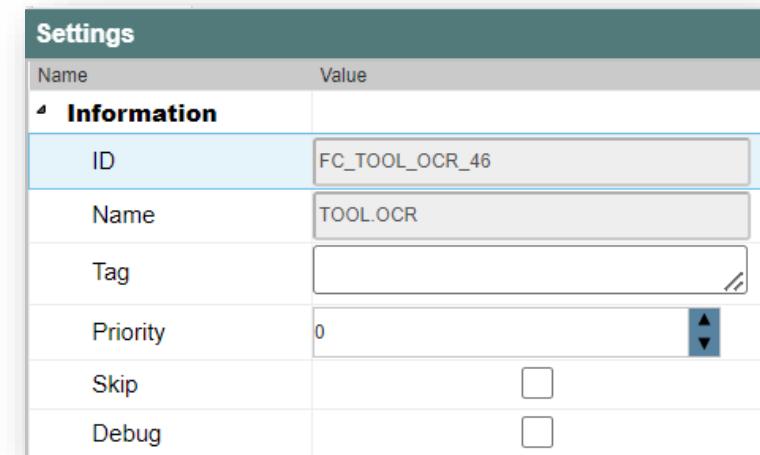


```
[238425] >>>> FC_MAIN_LOOP_1 <<<<  
[238725] >>>> FC_TOOL_OCR_38 <<<<  
outputs[2]-Output (string): 112  
[239039] >>>> FC_DEBUG_5 <<<<  
inputs[1]-Input (string): 112  
Trigger  
outputs[1]-Output (string): Trigger
```

# VIC Flow 函式

## • TOOL.OCR

名稱	意義	格式
<b>訊息(Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱(Name)</b>	運算子的名稱	
<b>標籤(Tag)</b>	設定運算子的標籤	String
<b>優先順序(Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過(Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯(Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	



# VIC Flow 函式

## • TOOL.OCR

名稱	意義	格式
<b>輸入參數(Input Parameters )</b>		
<b>ROI-X</b>	設置辨識範圍的X座標值	Python Expression
<b>ROI-Y</b>	設置辨識範圍的Y座標值	Python Expression
<b>ROI-寬度 (ROI-Width)</b>	設置辨識範圍的寬度	Python Expression
<b>ROI-高度 (ROI-Height)</b>	設置辨識範圍的高度	Python Expression
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>ROI X位移(ROI X Offset )</b>	設置辨識範圍的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>ROI Y位移 (ROI Y Offset )</b>	設置辨識範圍的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>白名單 (White List )</b>	僅辨識該設置值，若無設置則全辨識	String
<b>arr</b>	設置辨識使用的圖像 預設為CURRENT_IMAGE	
<b>允許空字串 (Allow Empty String)</b>	啟用後，當辨識到空字串時，不會顯示NG	
<b>移除空白字元 (Remove Whitespace)</b>	啟用後，當辨識結果有空白字元，將會自動移除	
<b>允許辨識率(%) (Allow Recognition Rate (%))</b>	設置允許最低的辨識率，若低於此值，將視為辨識失敗	

The screenshot shows the 'Input Parameters' configuration panel. It includes fields for ROI-X, ROI-Y, ROI-Width, ROI-Height, Update, ROI X Offset, ROI Y Offset, arr (set to CURRENT\_IMAGE), White List, Allow Empty String (unchecked), Remove Whitespace (unchecked), and Allow Recognition Rate (set to 80%).

- TOOL.OCR

名稱	意義	格式
<b>影像前處理 (Image Preprocess)</b>		
<b>影像前處理-調整大小 (Image Preprocess-Resize)</b>		
調整大小方式 (Resize Method)	設置調整大小的方式	
調整大小 (Resize)	設置放大倍率	
<b>影像前處理-二值化 (Image Preprocess-Threshold)</b>		
二值化方法 (Threshold Method)	設置二值化方法	
二值化演算法 (Threshold Algorithm)	設置二值化演算法	
閥值 (Threshold Value)	設置二值化閥值 僅在二值化演算法設為無時才可設置	Integer
<b>辨識字型 (OCR Font)</b>		
分割方法 (Segmentation Mode)	設置OCR的切割方法	
選擇字型 (Select Font)	設置使用的字型	
載入字型 (Load Font)	點擊後，即可選擇辨識字型檔	
字型檔 (Font File)	當前使用的字型檔名稱	

The screenshot shows the 'Image Preprocess' configuration interface. It includes sections for Resize (Method: Nearest, Factor: x1), Threshold (Method: Binary, Algorithm: OTSU, Value: 120), and OCR Font (Mode: Single Line, Font: English, Load Font button highlighted in green).

# VIC Flow 函式

## • TOOL.OCR

選擇字型檔  
&  
調整ROI大小及位置

設置調整大小的方式和倍數

調整二值化處理參數

**Input Parameters**

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	Submit

**OCR Font**

Segmentation Mode	Single Line
Select Font	English
Load Font	Submit
Font File	

**Resize**

Resize Method	Nearst
Resize	x1

**Threshold**

Threshold Method	Binary
Threshold Algorithm	OTSU
Threshold Value	120

**OCR Font**

Segmentation Mode	Single Line
Select Font	Font File
Load Font	Submit
Font File	VIC7000_Demo_green_value.traineddata

**Use Font Preprocess**

在字型檔欄位點擊右鍵  
選擇使用字型的前處理參數  
可快速設置字型檔使用的前處理參數

# VIC Flow 函式

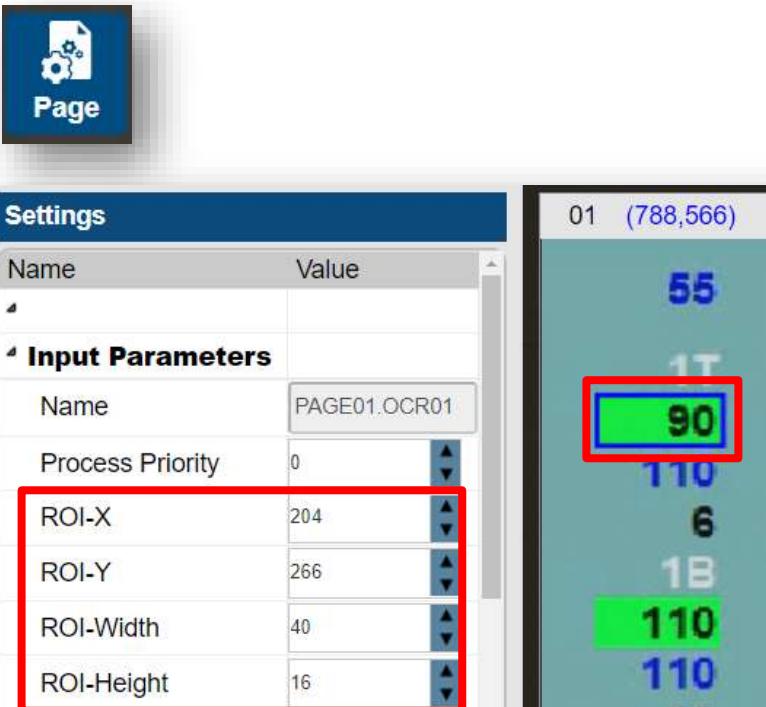
## • TOOL.OCR

選擇字型檔 & ROI大小及位置 & 前理處參數設定(方式1)：

在頁面設定中的OCR設置，找到要使用的**ROI資訊**

要使用的**字型檔**、要使用的**前處理參數**

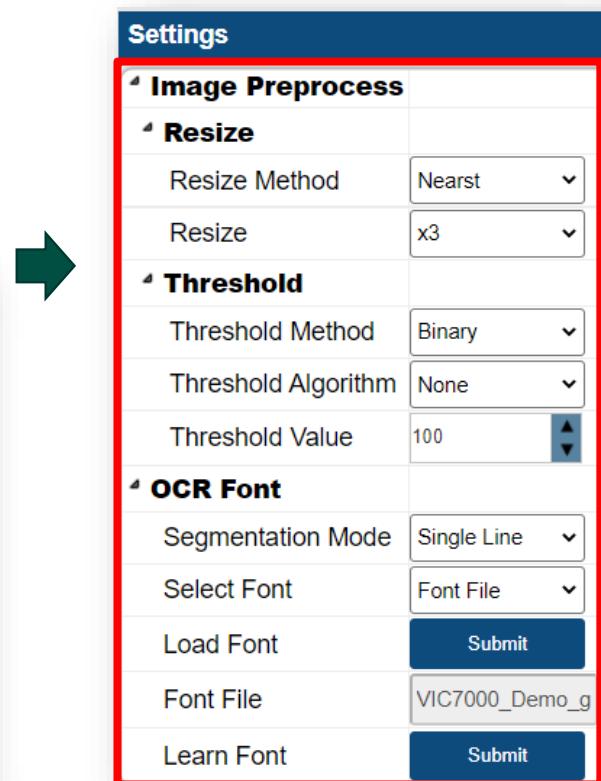
最後帶入TOOL.OCR運算子



Page

Settings

Name	Value
Name	PAGE01.OCR01
Process Priority	0
ROI-X	204
ROI-Y	266
ROI-Width	40
ROI-Height	16



Settings

**Image Preprocess**

**Resize**

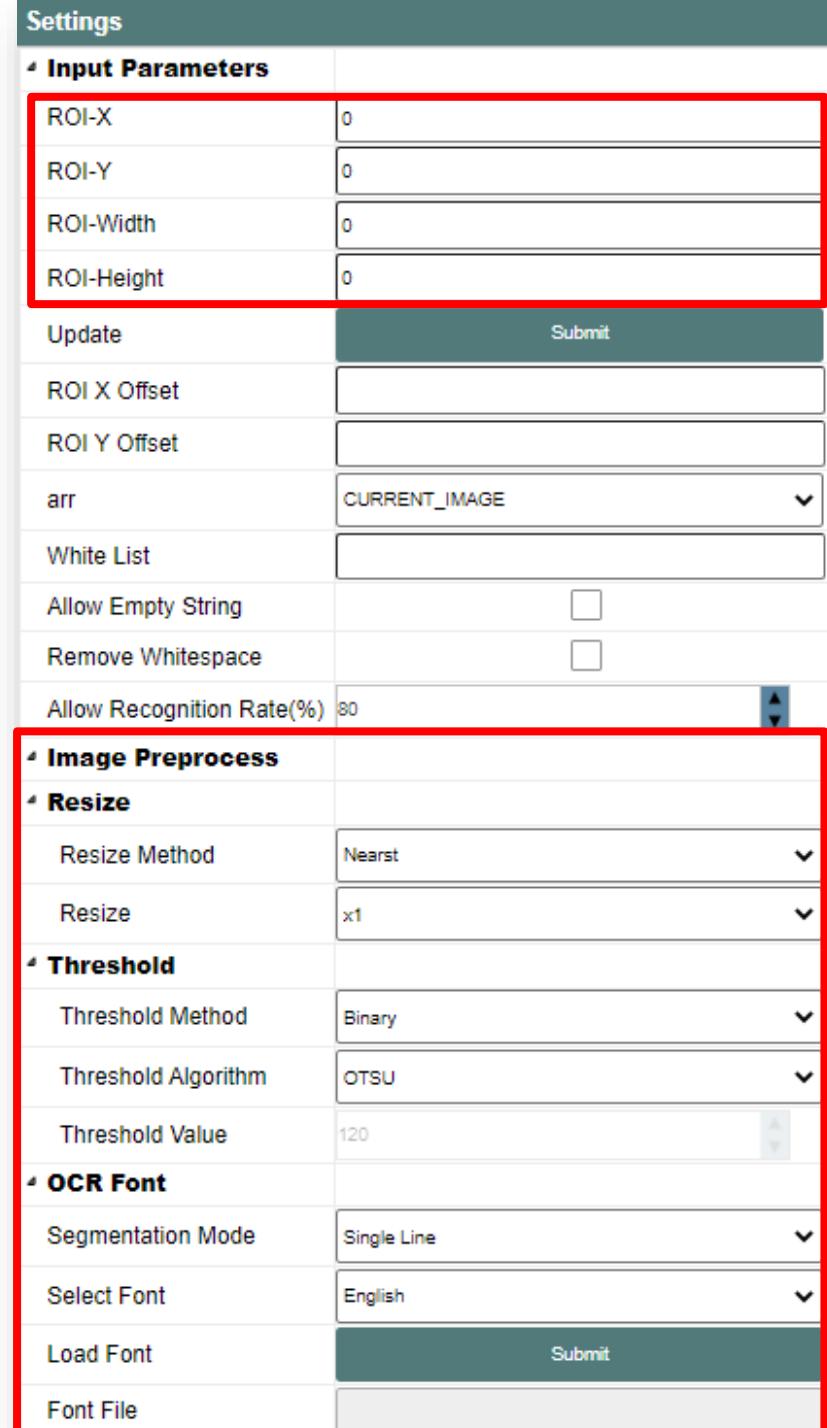
- Resize Method: Nearest
- Resize: x3

**Threshold**

- Threshold Method: Binary
- Threshold Algorithm: None
- Threshold Value: 100

**OCR Font**

- Segmentation Mode: Single Line
- Select Font: Font File
- Load Font: Submit
- Font File: VIC7000\_Demo\_g
- Learn Font: Submit



Settings

**Input Parameters**

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0

Update: Submit

ROI X Offset

ROI Y Offset

arr: CURRENT\_IMAGE

White List

Allow Empty String

Remove Whitespace

Allow Recognition Rate(%): 80

**Image Preprocess**

**Resize**

- Resize Method: Nearest
- Resize: x1

**Threshold**

- Threshold Method: Binary
- Threshold Algorithm: OTSU
- Threshold Value: 120

**OCR Font**

- Segmentation Mode: Single Line
- Select Font: English
- Load Font: Submit
- Font File:

# VIC Flow 函式

## • TOOL.OCR

選擇字型檔 & ROI大小及位置 & 前理處參數設定(方式2)：

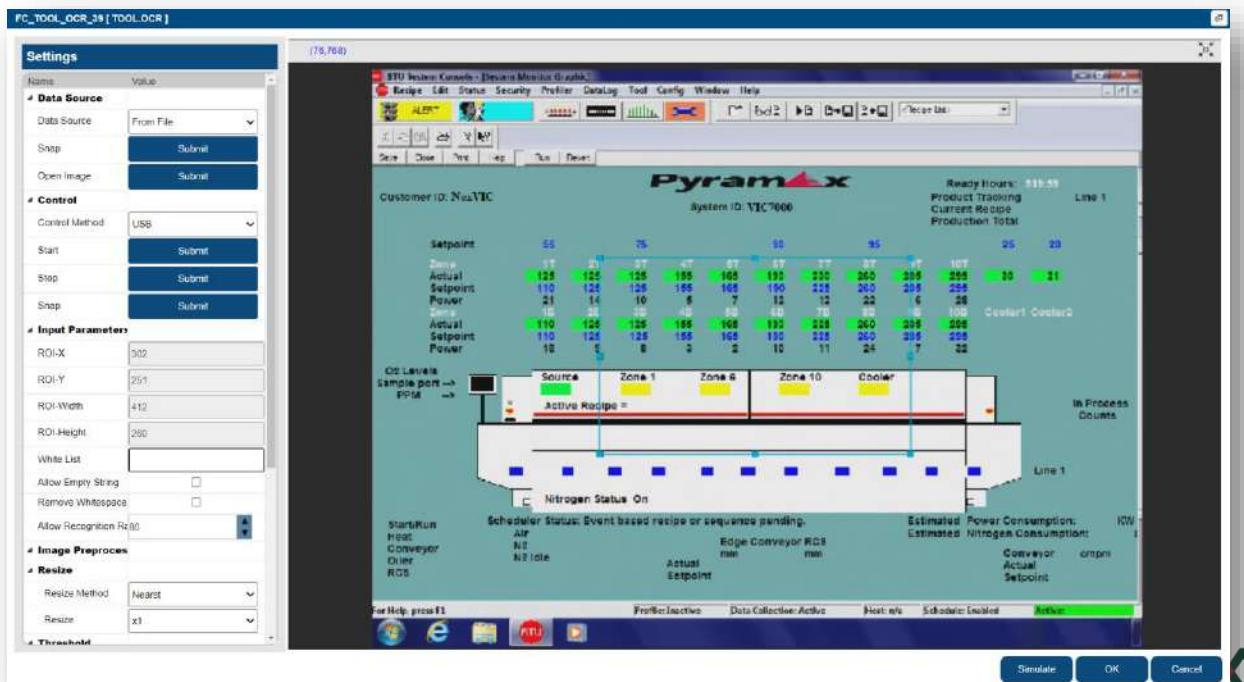
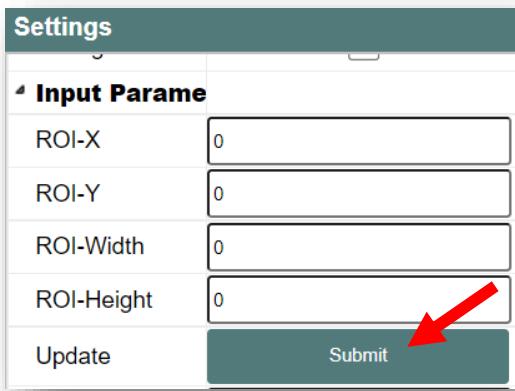
使用TOOL.OCR內的更新設窗

設置方式同頁面模式中進行OCR設置

視窗中設定的ROI資訊、使用的字型檔、使用的前處理參數

在設置完成並點擊確定後

會自動帶入TOOL.OCR的設定內



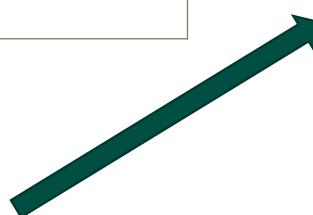
# VIC Flow 函式

## • TOOL.COLOR

可自行設置辨識圖檔及辨識參數並進行顏色辨識的運算子

輸出端(True與False)輸出的數值為布林值(Boolean)

	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用系統當下的圖像進行辨識
	ROI Offset	輸入ROI需偏移的位移量 格式為offset_x, offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	True	比對成功，則由此輸出
	False	比對失敗，則由此輸出

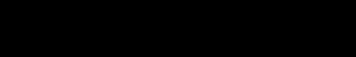


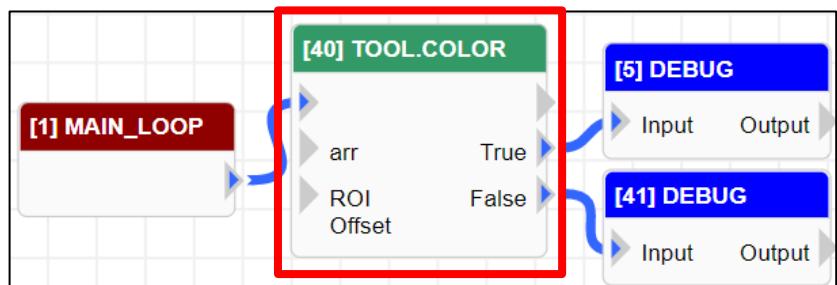
Settings

Input Parameters

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	Submit
ROI X Offset	
ROI Y Offset	
arr	CURRENT_IMAGE

Matching Parameters

Color Picker	
Red	0
Green	0
Blue	0
Matching Tolerance	5



[551126] >>>> FC\_MAIN\_LOOP\_1 <<<<  
[551427] >>>> FC\_TOOL\_COLOR\_40 <<<<  
outputs[1] (bool): true  
[551740] >>>> FC\_DEBUG\_5 <<<<  
outputs[1]-Output (string):

# VIC Flow 函式

## • TOOL.COLOR

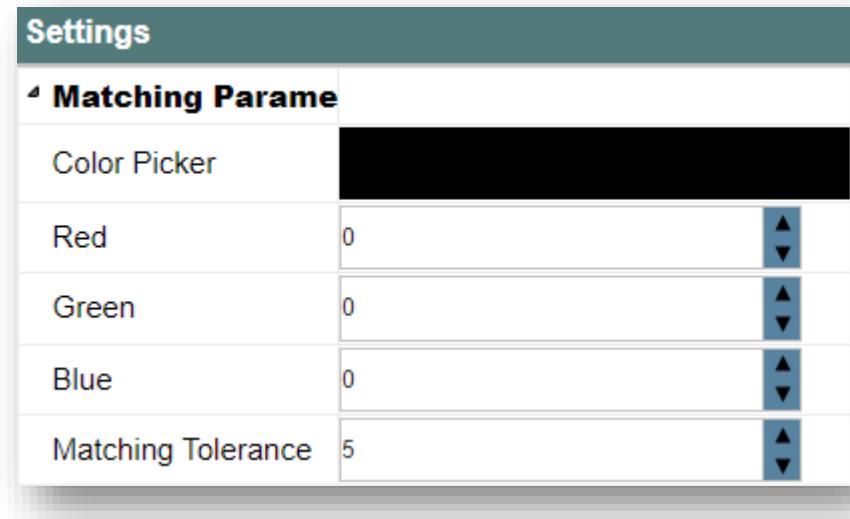
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>輸入參數 (Input Parameters)</b>		
<b>ROI-X</b>	設置辨識範圍的X座標值	Python Expression
<b>ROI-Y</b>	設置辨識範圍的Y座標值	Python Expression
<b>ROI-寬度 (ROI-Width)</b>	設置辨識範圍的寬度	Python Expression
<b>ROI-高度 (ROI-Height)</b>	設置辨識範圍的高度	Python Expression
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>ROI X 位移 (ROI X Offset)</b>	設置辨識範圍的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>ROI Y 位移 (ROI Y Offset)</b>	設置辨識範圍的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>arr</b>	設置辨識使用的圖像 預設為CURRENT_IMAGE	

Settings	
Name	Value
<b>Information</b>	
ID	FC_TOOL_COLOR_40
Name	TOOL.COLOR
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Input Parameters</b>	
ROI-X	332
ROI-Y	420
ROI-Width	31
ROI-Height	7
Update	Submit
ROI X Offset	
ROI Y Offset	
arr	CURRENT_IMAGE

# VIC Flow 函式

- **TOOL.COLOR**

名稱	意義	格式
<b>比對參數 (Matching Parameters)</b>		
顏色選擇 (Color Picker)	點擊後，再點擊欲比對顏色的位置，即可將比對顏色設置為點擊位置的顏色	
紅色 (Red)	設置比對RGB三原色中的紅色(Red)值	Integer
綠色 (Green)	設置比對RGB三原色中的綠色(Green)值	Integer
藍色 (Blue)	設置比對RGB三原色中的藍色(Blue)值	Integer
比對容錯度 (Matching Tolerance)	設置顏色比對容錯度	Integer



## • TOOL.COLOR

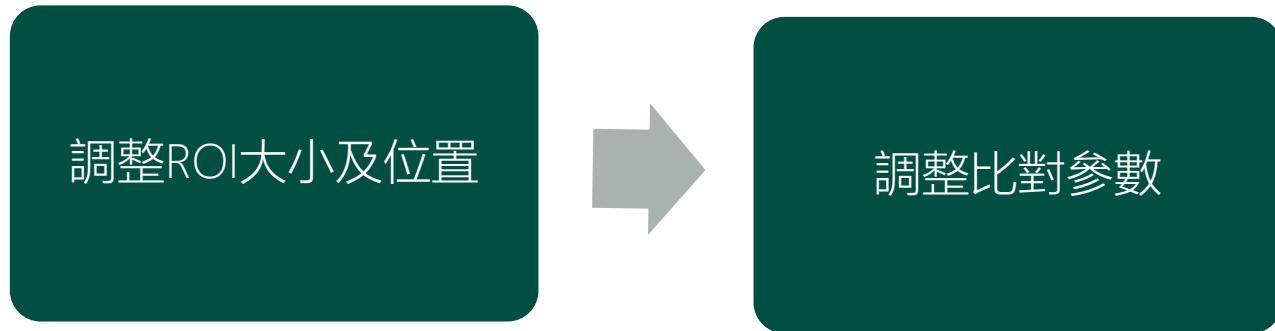
TOOL.COLOR回傳的資訊有RED、GREEN、BLUE、VALUE、MATCH

呼叫方式及代表意義如下

呼叫方式	意義
<code> \${TOOL_COLOR_ID.RED}</code>	比對RGB三原色中的紅色(Red)
<code> \${TOOL_COLOR_ID.GREEN}</code>	比對RGB三原色中的綠色(Green)
<code> \${TOOL_COLOR_ID.BLUE}</code>	比對RGB三原色中的藍色(Blue)
<code> \${TOOL_COLOR_ID.VALUE}</code>	比對結果值，成功為true，反之為false
<code> \${TOOL_COLOR_ID.MATCH}</code>	是否比對成功，成功為1，反之為0

# VIC Flow 函式

- **TOOL.COLOR**



Input Parameters	
ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0

Matching Parameters	
Color Picker	
Red	0
Green	0
Blue	0
Matching Tolerance	5

# VIC Flow 函式

## • TOOL.COLOR

ROI大小及位置 & 比對參數設定(方式1)：

在頁面設定中的COLOR設置，找到要使用的ROI資訊

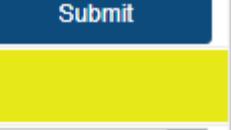
以及比對參數，最後帶入TOOL.COLOR運算子



Settings

Name	Value
<b>Input Parameters</b>	
Name	PAGE01.COLOR01
Process Priority	0
ROI-X	332
ROI-Y	420
ROI-Width	31
ROI-Height	7

Matching Parameters

Set ROI Color	Submit
Color Picker	
Red	231
Green	232
Blue	23
Matching Tolerance	5



Settings

**Input Parameters**

ROI-X	332
ROI-Y	420
ROI-Width	31
ROI-Height	7

**Matching Parameters**

Color Picker	
Red	231
Green	232
Blue	23
Matching Tolerance	5

# VIC Flow 函式

## • TOOL.COLOR

ROI大小及位置 & 比對參數設定(方式2)：

使用TOOL.COLOR內的更新設窗，設置方式同頁面模式中進行COLOR設置

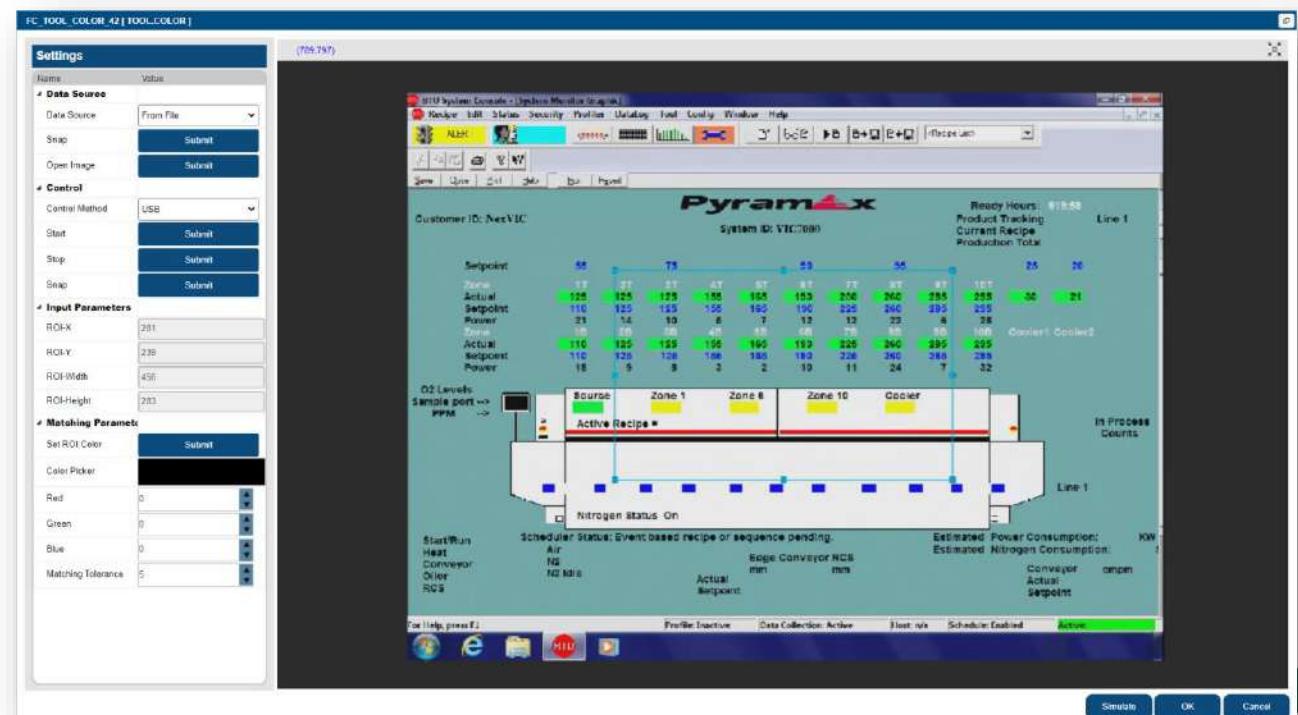
視窗中設定的**ROI資訊**和使用的**比對參數**

在設置完成並點擊確定後，會自動帶入TOOL.COLOR的設定內

**Settings**

**Input Parameters**

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	<b>Submit</b>



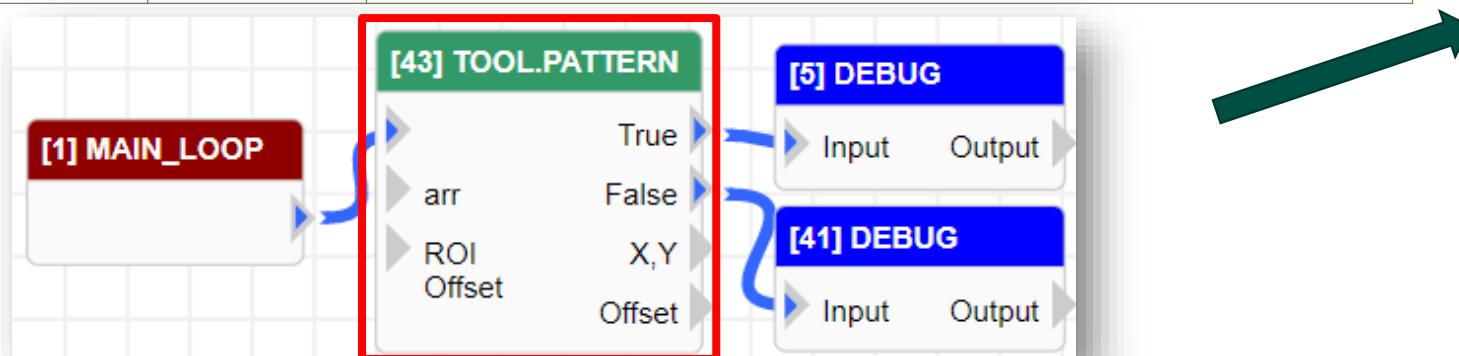
# VIC Flow 函式

## • TOOL.PATTERN

可自行設置辨識圖檔及辨識參數並進行樣式比對的運算子

輸出端(True與False) · 輸出的數值為布林值(Boolean)

	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用系統當下的圖像進行辨識
	ROI Offset	輸入ROI需偏移的位移量 格式為offset_x,offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	True	若比對成功，則由此輸出
	False	若比對失敗，則由此輸出
	X,Y	輸出比對成功樣式的中心座標位置
	Offset	輸出比對成功樣式的位置與設置比對樣式的位置相比的偏移量



**Settings**

**Input Parameters**

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	Submit
ROI X Offset	
ROI Y Offset	
arr	CURRENT_IMAGE

**Matching Parameters**

Pattern Name	
Pattern Load	Submit
Pattern X	0
Pattern Y	0
Gray Matching	<input type="checkbox"/>
Number Of Image Level	0
Method	Correlation Coefficient
Minimum Score	0.95
Wait Time (ms)	0
Wait Condition	Until True

[051677] >>>> FC\_MAIN\_LOOP\_1 <<<<  
[051977] >>>> FC\_TOOL\_PATTERN\_43 <<<<  
outputs[3]-X,Y (string): 501,149  
outputs[4]-Offset (string): 0,0  
[052313] >>>> FC\_DEBUG\_5 <<<<  
outputs[1]-Output (string):

# VIC Flow 函式

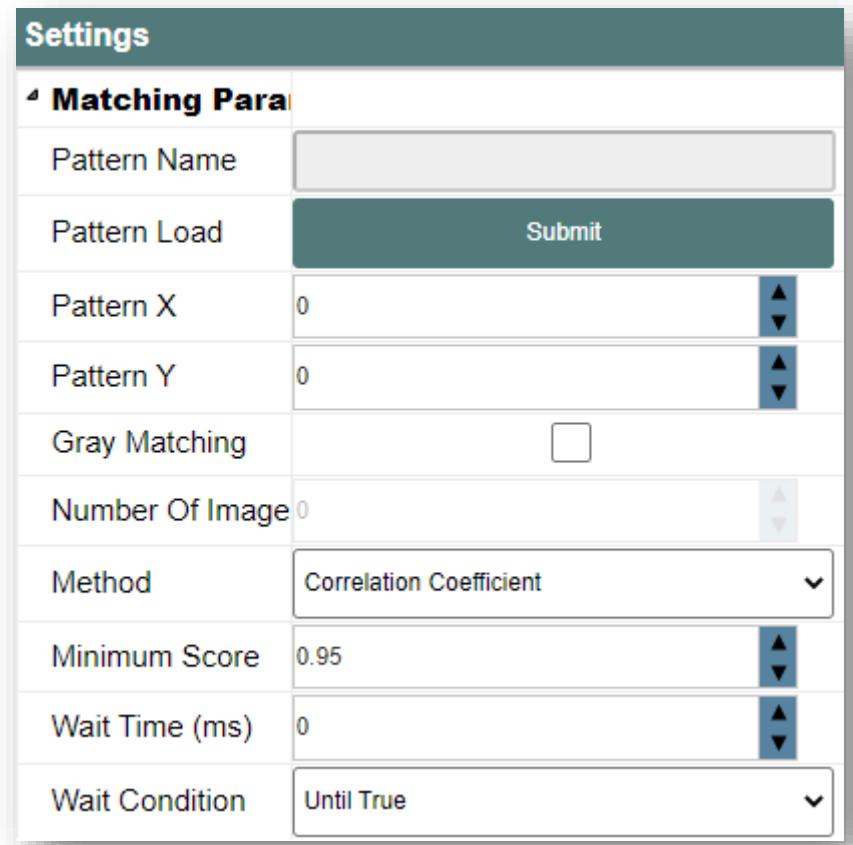
## • TOOL.PATTERN

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>輸入參數 (Input Parameters)</b>		
<b>ROI-X</b>	設置辨識範圍的X座標值	Python Expression
<b>ROI-Y</b>	設置辨識範圍的Y座標值	Python Expression
<b>ROI-寬度 (ROI-Width)</b>	設置辨識範圍的寬度	Python Expression
<b>ROI-高度 (ROI-Height)</b>	設置辨識範圍的高度	Python Expression
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>ROI X 位移 (ROI X Offset)</b>	設置辨識範圍的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>ROI Y 位移 (ROI Y Offset)</b>	設置辨識範圍的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>arr</b>	設置辨識使用的圖像 預設為CURRENT_IMAGE	

Settings	
Name	Value
<b>Information</b>	
ID	FC_TOOL_PATTERN_44
Name	TOOL.PATTERN
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Input Parameters</b>	
ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	Submit
ROI X Offset	
ROI Y Offset	
arr	CURRENT_IMAGE

## • TOOL.PATTERN

名稱	意義	格式
<b>比對參數 (Matching Parameters)</b>		
<b>樣式影像名稱 (Pattern Name)</b>	顯示比對樣式影像的檔案名稱	
<b>樣式載入 (Pattern Load)</b>	點擊後，即會顯示樣式載入視窗	
<b>樣式X (Pattern X)</b>	設置比對樣式的X值	Integer
<b>樣式Y (Pattern Y)</b>	設置比對樣式的Y值	Integer
<b>灰階比對 (Gray Matching)</b>	啟用後，會將圖像轉為灰階化再進行樣式比對 可加快樣式比對的速度	
<b>圖像層數 (Number Of Image Levels)</b>	設置比對圖像層數，在啟用灰階比對時才有效	
<b>方法(Method)</b>	設置比對方法，分為相關係數和互相關	
<b>最小分數 (Minimum Score)</b>	設置樣式比對成功的最小分數 預設為0.95，有效值範圍為0至1 1為完全吻合	Float
<b>等待時間(ms) (Wait Time (ms))</b>	設置每次比對的等待時間	Integer
<b>等待條件 (Wait Condition)</b>	設置等待的條件： 包含直到True(Until True)，代表直到比對成功以及直到False(Until False)，代表直到比對失敗	



## • TOOL.PATTERN

TOOL.PATTERN回傳的資訊有FOUND、VALUE、SCORE、X、Y、WIDTH、HEIGHT、BASE\_X、  
BASE\_Y、CENTER\_X、CENTER\_Y、OFFSET\_X、OFFSET\_Y

其呼叫方式和代表意義如下

呼叫方式	意義
<code> \${TOOL_PATTERN_ID.FOUND}</code>	樣式是否比對成功，若成功為1；反之為0
<code> \${TOOL_PATTERN_ID.VALUE}</code>	樣式比對成功輸出，若成功為true；反之為false
<code> \${TOOL_PATTERN_ID.SCORE}</code>	樣式的比對分數，範圍為0 ~ 1.0
<code> \${TOOL_PATTERN_ID.X}</code>	比對到的樣式X座標
<code> \${TOOL_PATTERN_ID.Y}</code>	比對到的樣式Y座標
<code> \${TOOL_PATTERN_ID.WIDTH}</code>	比對到的樣式寬度
<code> \${TOOL_PATTERN_ID.HEIGHT}</code>	比對到的樣式高度
<code> \${TOOL_PATTERN_ID.BASE_X}</code>	設定的樣式X座標(可用於計算偏移量)
<code> \${TOOL_PATTERN_ID.BASE_Y}</code>	設定的樣式Y座標(可用於計算偏移量)
<code> \${TOOL_PATTERN_ID.CENTER_X}</code>	比對到的樣式中心點X座標
<code> \${TOOL_PATTERN_ID.CENTER_Y}</code>	比對到的樣式中心點Y座標
<code> \${TOOL_PATTERN_ID.OFFSET_X}</code>	比對到的樣式與比對樣式的X座標偏移量
<code> \${TOOL_PATTERN_ID.OFFSET_Y}</code>	比對到的樣式與比對樣式的Y座標偏移量

# VIC Flow 函式

- **TOOL.PATTERN**

調整ROI大小及位置



設置比對樣式  
和比對參數

**Input Parameters**

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0

**Settings**

**Matching Parameters**

Pattern Name	<input type="text"/>
Pattern Load	<input type="button" value="Submit"/>
Pattern X	0
Pattern Y	0
Gray Matching	<input type="checkbox"/>
Number Of Image Levels	<input type="button"/>
Method	Correlation Coefficient
Minimum Score	0.95
Wait Time (ms)	0
Wait Condition	Until True

## • TOOL.PATTERN

ROI大小及位置 & 比對樣式和參數設定(方式1)：

在頁面設定中的PATTERN設置，找到要使用的ROI資訊

將它帶入TOOL.PATTERN運算子



Settings

Name	Value
<b>Input Parameters</b>	
Name	PAGE01.PATTERN
Process Priority	0
ROI-X	345
ROI-Y	94
ROI-Width	313
ROI-Height	119



Settings

**Input Parameters**

ROI-X	345
ROI-Y	94
ROI-Width	313
ROI-Height	119

Update      Submit

ROI X Offset

ROI Y Offset

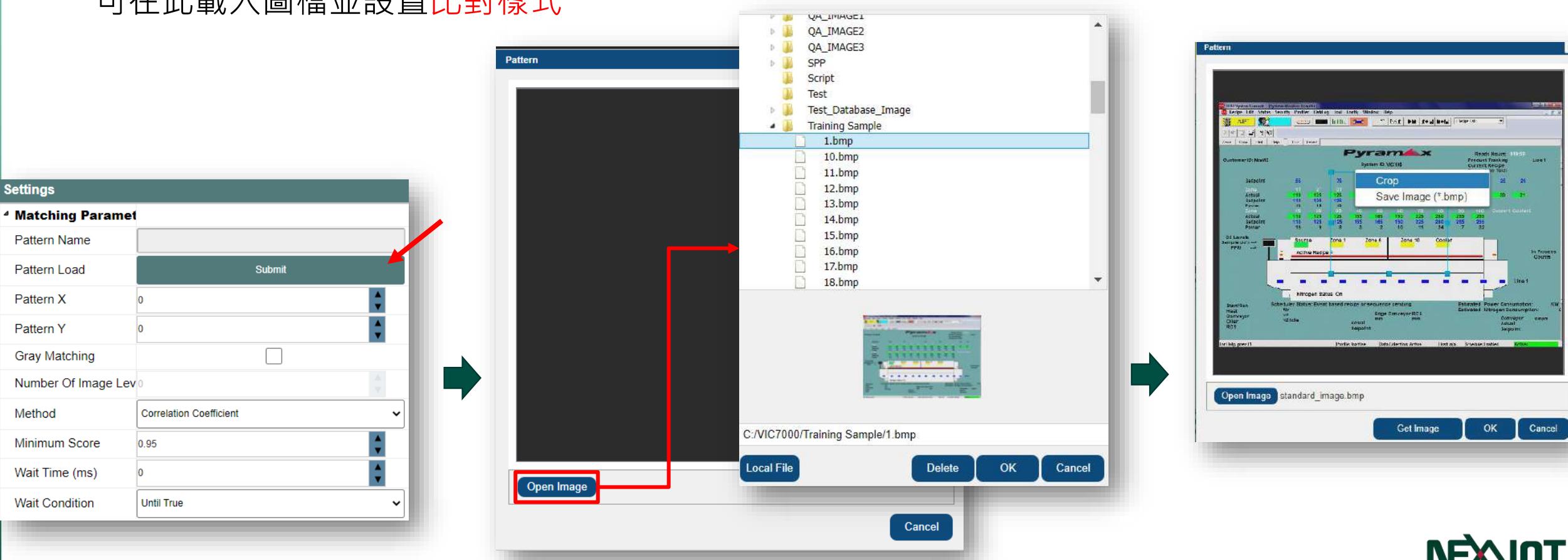
arr      CURRENT\_IMAGE

# VIC Flow 函式

## • TOOL.PATTERN

ROI大小及位置 & 比對樣式和參數設定(方式1)：

在TOOL.PATTERN的設置內，點擊樣式載入，即會出現樣式載入視窗  
可在此載入圖檔並設置比對樣式



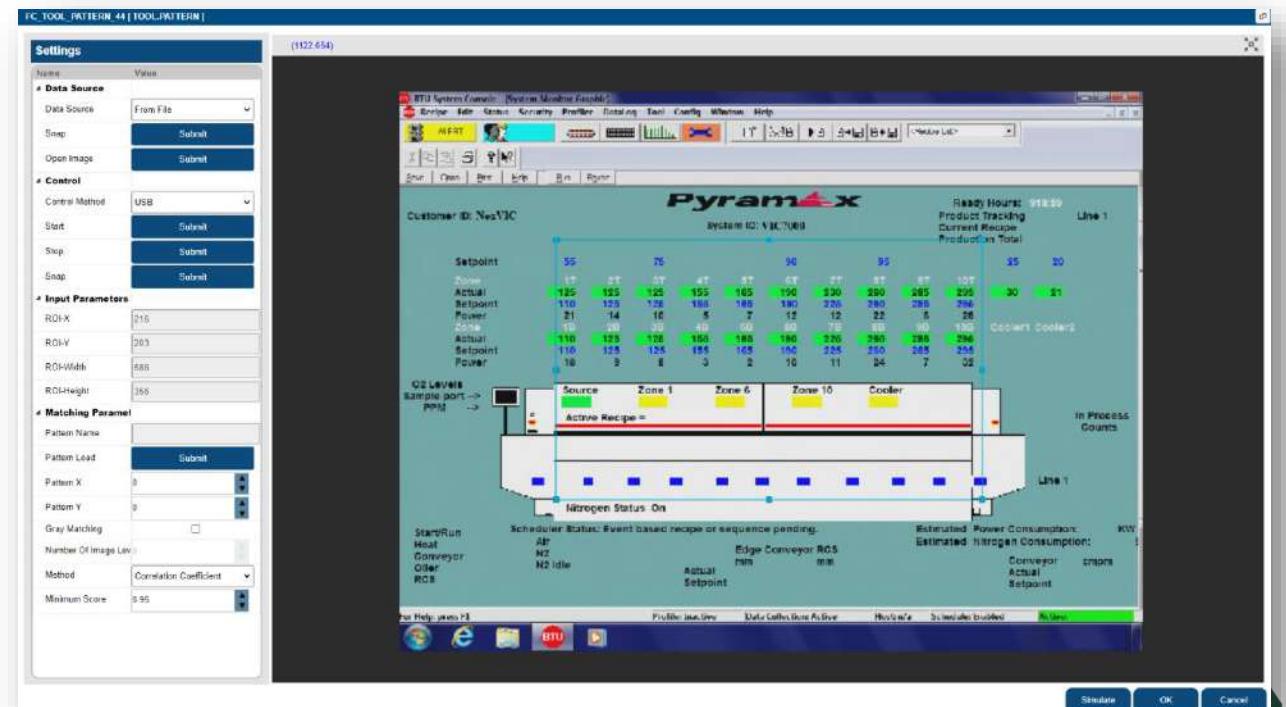
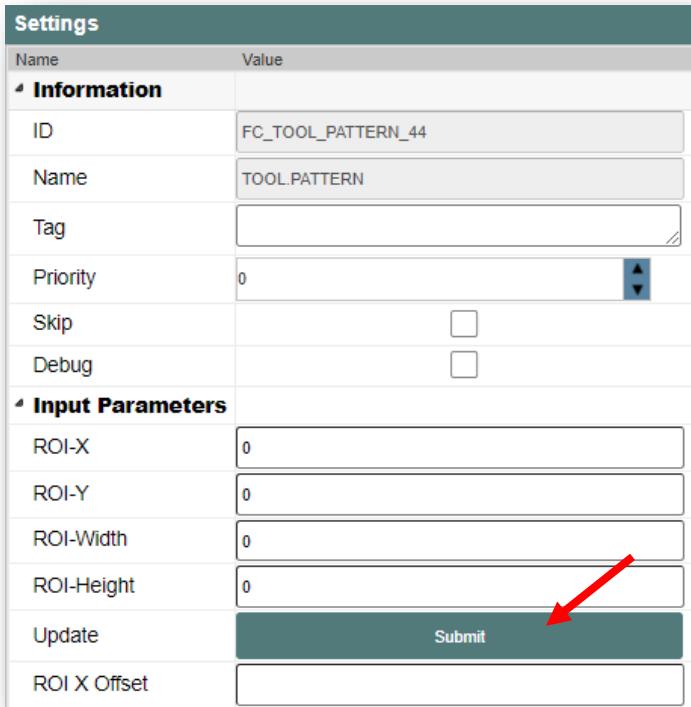
- **TOOL.PATTERN**

## ROI大小及位置 & 比對樣式及參數設定(方式2)：

使用TOOL.PATTERN內的更新設窗，設置方式同頁面模式中進行PATTTERN設置

視窗中設定的ROI資訊和使用的比對樣式和參數

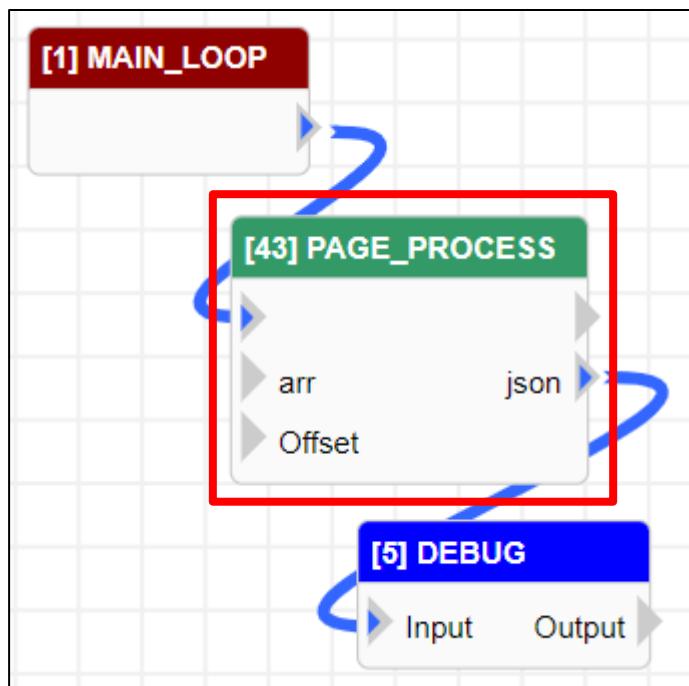
在設置完成並點擊確定後，會自動帶入TOOL.PATTERN的設定內容。



## • PAGE\_PROCESS

可使用指定頁面的辨識設定進行辨識圖像的運算子，輸出結果的格式為 JSON 字串

若文字辨識結果為NG，此結果將不會被加入輸出的 JSON 字串中



	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像
	位移	輸入辨識設定ROI的偏移量
輸出端	json	輸出辨識結果 輸出格式為JSON字串



```

[303359] >>>> FC_MAIN_LOOP_1 <<<<
[303660] >>>> FC_PAGE_PROCESS_43 <<<<
outputs[2]-json (string): {"O01":"110","OCR01":"110",...}
[303977] >>>> FC_DEBUG_5 <<<<
inputs[1]-Input (string): {"O01":"110","OCR01":"110",...}
{"O01":"110","OCR01":"110","O02": "", "OCR02": "", "C0...
outputs[1]-Output (string): {"O01":"110","OCR01":"11...

```

## • PAGE\_PROCESS

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>頁面 (Page)</b>	欲使用的頁面設定的頁面編號	Integer
<b>工具列表 (Tools List)</b>	欲使用的辨識工具，如OCR01、COLOR01、PATTERN01 若為空，代表使用該頁面中設定的全部辨識工具 可使用辨識工具的別名	String
<b>X位移 (X Offset)</b>	設置該頁面工具辨識範圍的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置該頁面工具辨識範圍的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Arr</b>	設置辨識使用的圖像 預設為CURRENT_IMAGE	
<b>完整json (Detailed json)</b>	設置是否輸出詳細的辨識結果 詳細辨識結果包含COLOR辨識結果中比對RGB三原色各別的數值 以及PATTERN辨識結果中比對分數、比對樣式座標等資訊	

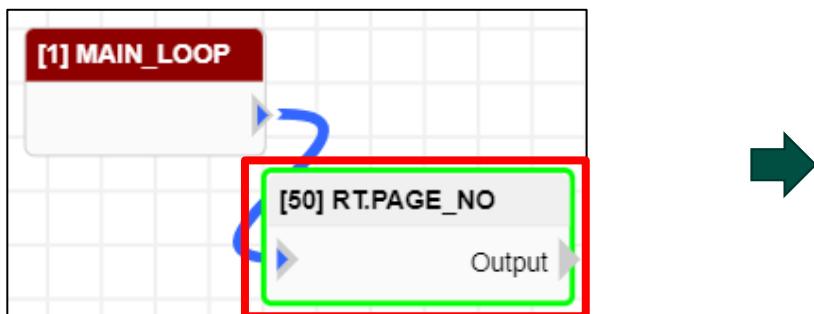
Settings	
Name	Value
<b>Information</b>	
ID	FC_PAGE_PROCESS_43
Name	PAGE_PROCESS
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Page Id	1
Tools List	
X Offset	
Y Offset	
arr	CURRENT_IMAGE
Detailed json	<input type="checkbox"/>

- **RT.PAGE\_NO**

可取得當前執行辨識的頁面編號的運算子，輸出格式為字串

舉例來說，若當前執行辨識的頁面為頁面三，其輸出的字串為03

	UI名稱	行為
輸入端		
輸出端	Output	輸出當前執行辨識的頁面的編號



```
[495427] >>>> FC_MAIN_LOOP_1 <<<<  
[495427] >>>> FC_RT_VAR_50 <<<<  
outputs[1]-Output (string): 01
```

# VIC Flow 函式

- RT.PAGE\_NO

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

Settings	
Name	Value
<b>Information</b>	
ID	FC_RT_VAR_51
Name	RT.PAGE_NO
Tag	/
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

## • RT.RESULT

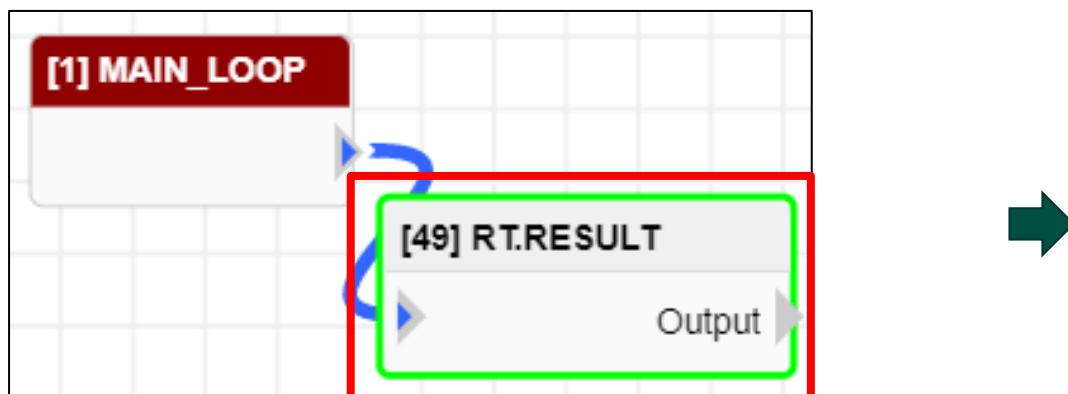
可取得當前辨識結果的運算子，輸出格式為整數，其代表含意如下表所示

辨識結果	輸出
辨識成功 (Pass)	0
辨識失敗 (NG)	1
比對失敗 (No Matching)	2

UI名稱	行為
輸入端	

輸出端	Output	輸出辨識結果
-----	--------	--------



```
[413323] >>>> FC_MAIN_LOOP_1 <<<<
[413324] >>>> FC_RT_VAR_15 <<<<
outputs[1]-Output (integer): 0
[493520] >>>> FC_MAIN_LOOP_1 <<<<
[493520] >>>> FC_RT_VAR_15 <<<<
outputs[1]-Output (integer): 2
```

# VIC Flow 函式

- RT.RESULT

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

Settings	
Name	Value
<b>Information</b>	
ID	FC_RT_VAR_52
Name	RT.RESULT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

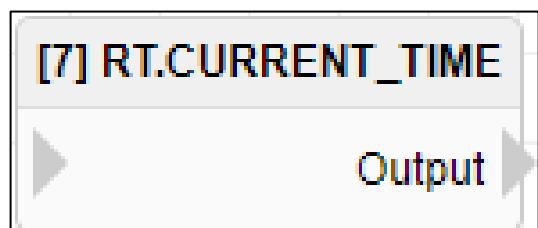
- **RT.CURRENT\_TIME**

可取得當前時間的運算子，輸出字串格式為YYYY-MM-DD HH:MM:SS

該運算子格式為物件，可以個別取得年、月、日、時、分、秒，取得方式如下表

	UI名稱	行為
輸入端		
輸出端	Output	輸出當前時間

方式	意義
<code> \${OPERATOR_ID.YEAR}</code>	取得當前時間的年份
<code> \${OPERATOR_ID.MONTH}</code>	取得當前時間的月份
<code> \${OPERATOR_ID.DAY}</code>	取得當前時間的日期
<code> \${OPERATOR_ID.HOUR}</code>	取得當前時間的小時
<code> \${OPERATOR_ID.MINUTE}</code>	取得當前時間的分鐘
<code> \${OPERATOR_ID.SECOND}</code>	取得當前時間的秒數



```

[410771] >>>>> FC_MAIN_LOOP_1 <<<<<
[410771] >>>>> FC_RT_VAR_16 <<<<<
outputs[1]-Output (string): 2022-05-06 15:33:30
  
```

# VIC Flow 函式

- RT.CURRENT\_TIME

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

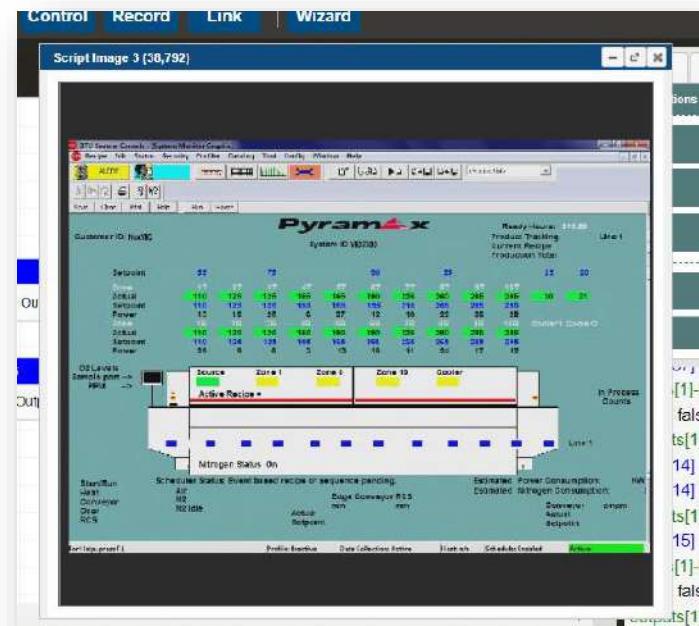
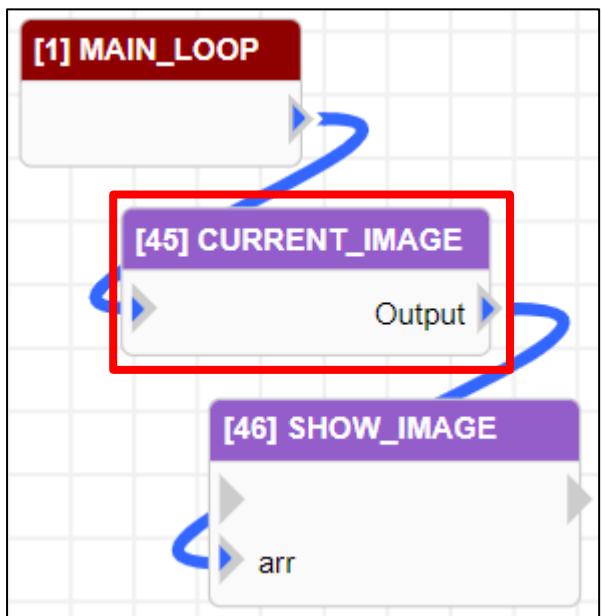
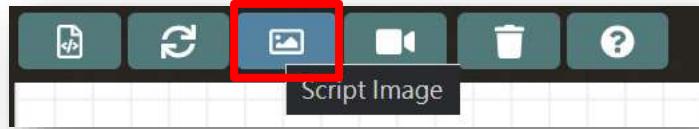
Settings	
Name	Value
▲ <b>Information</b>	
ID	FC_RT_VAR_53
Name	RT.CURRENT_TIME
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

# VIC Flow 函式

## • CURRENT\_IMAGE

可取得當前辨識圖像的運算子

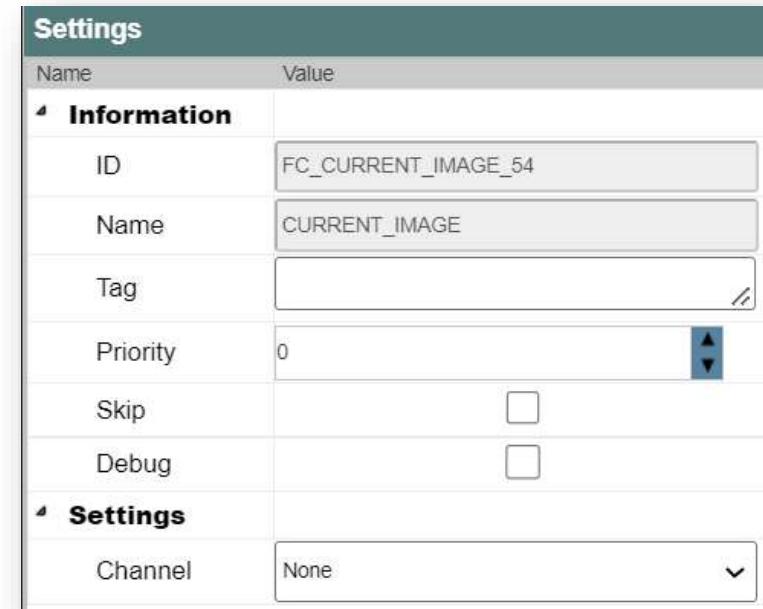
	UI名稱	行為
輸入端		
輸出端	Output	輸出當前辨識圖像



# VIC Flow 函式

## • CURRENT\_IMAGE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	



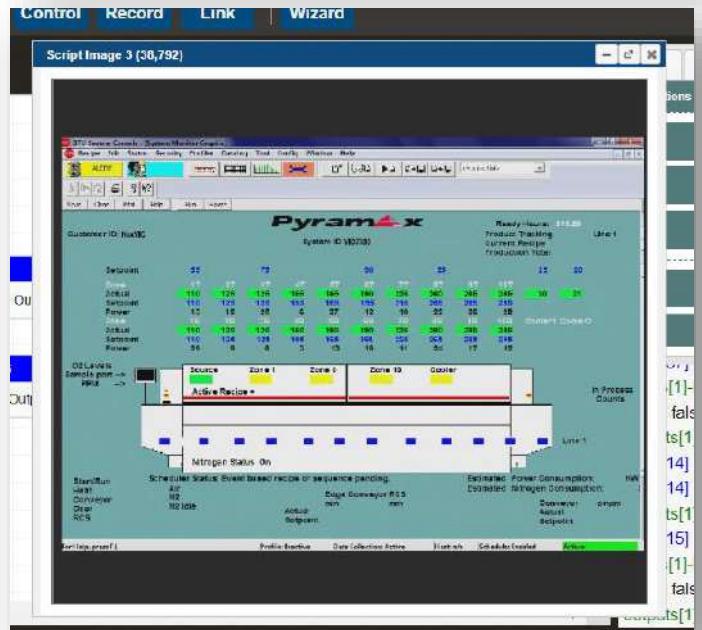
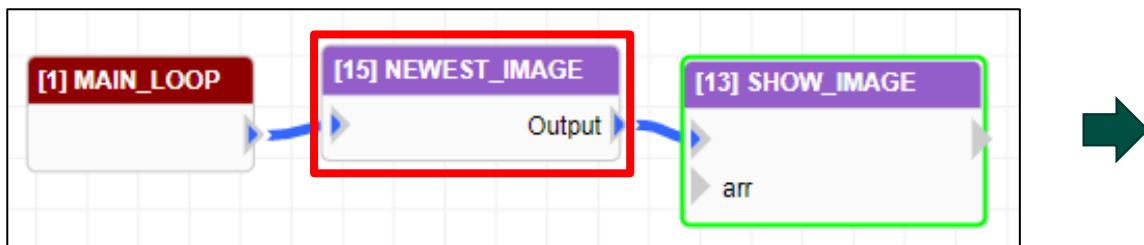
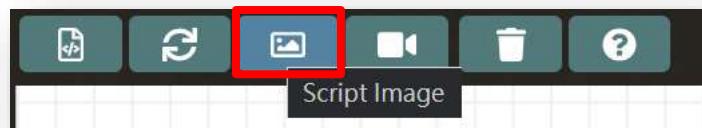
Name	Value
ID	FC_CURRENT_IMAGE_54
Name	CURRENT_IMAGE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

# VIC Flow 函式

## • NEWEST\_IMAGE

可由取像來源取得最新擷取圖檔的運算子

	UI名稱	行為
輸入端		
輸出端	Output	輸出最新擷取影像



# VIC Flow 函式

- NEWEST\_IMAGE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

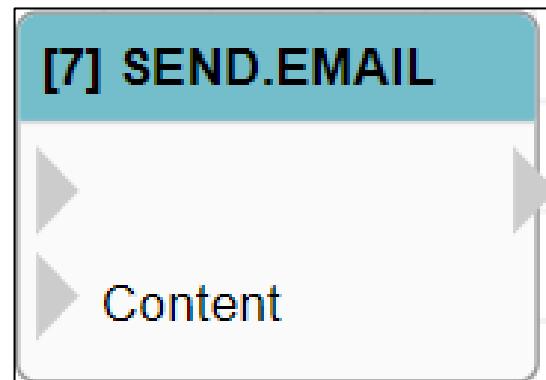
Settings

Name	Value
ID	FC_NEWEST_IMAGE_55
Name	NEWEST_IMAGE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Channel	None

- **SEND.EMAIL**

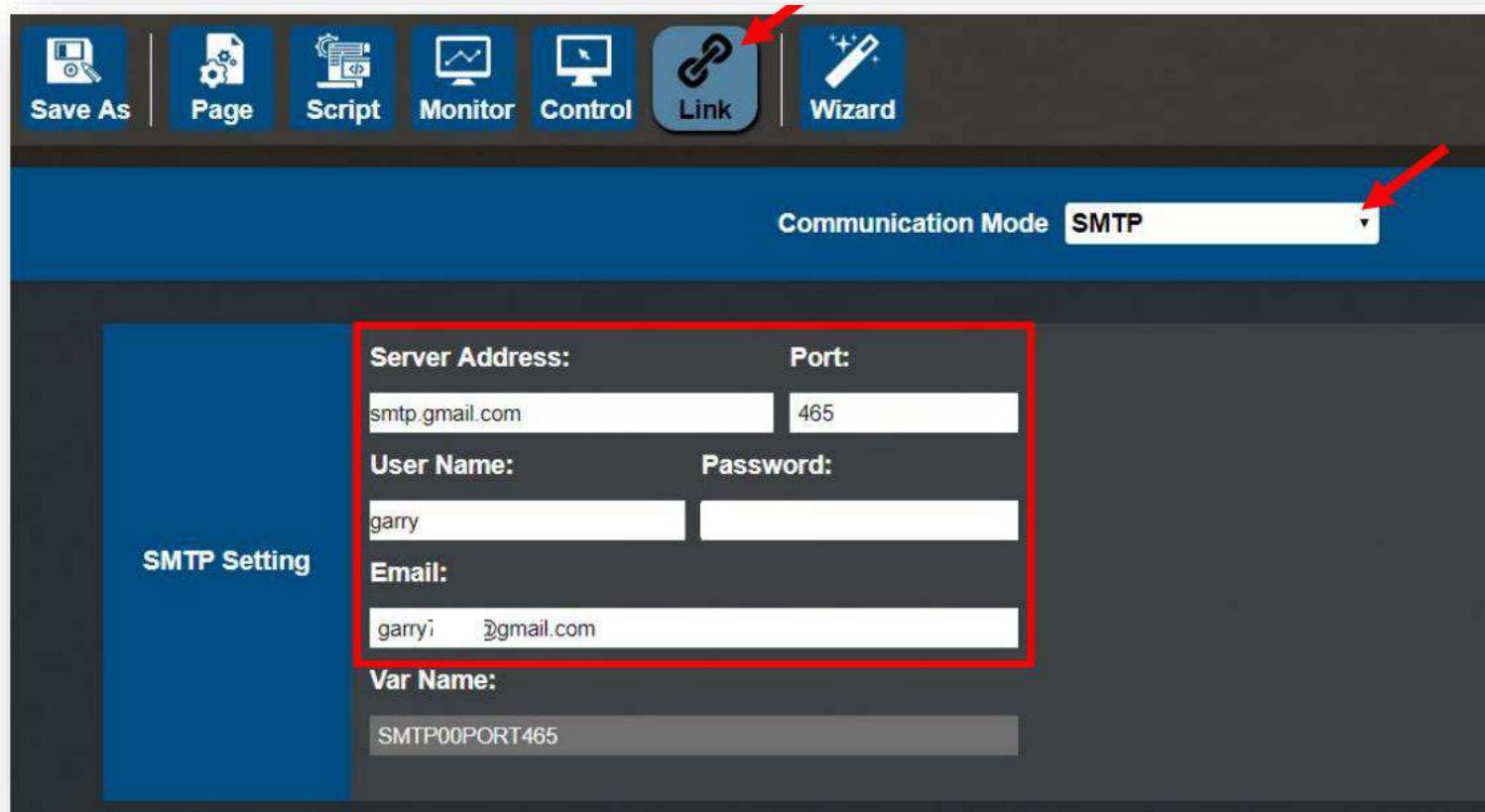
可將輸入端或設置的內容透過電子郵件發送通知的運算子

	UI名稱	行為
輸入端		
	Content	輸入欲發送的內容
輸出端		



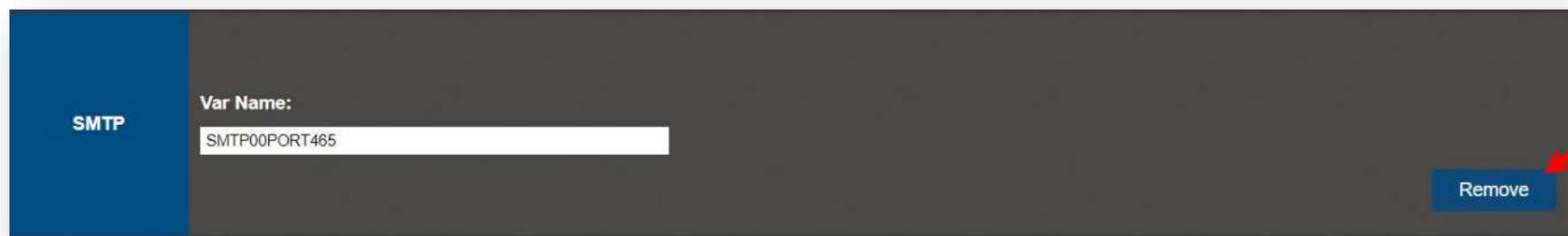
- **SEND.EMAIL**

新增連線 – Email：進入連結頁面，通訊模式選擇SMTP，輸入伺服器位址、通訊埠、使用者名稱、密碼、電子信箱位址，點擊新增



- **SEND.EMAIL**

新增連線 – Email：下方確認已新增SMTP連線，可在此移除連線

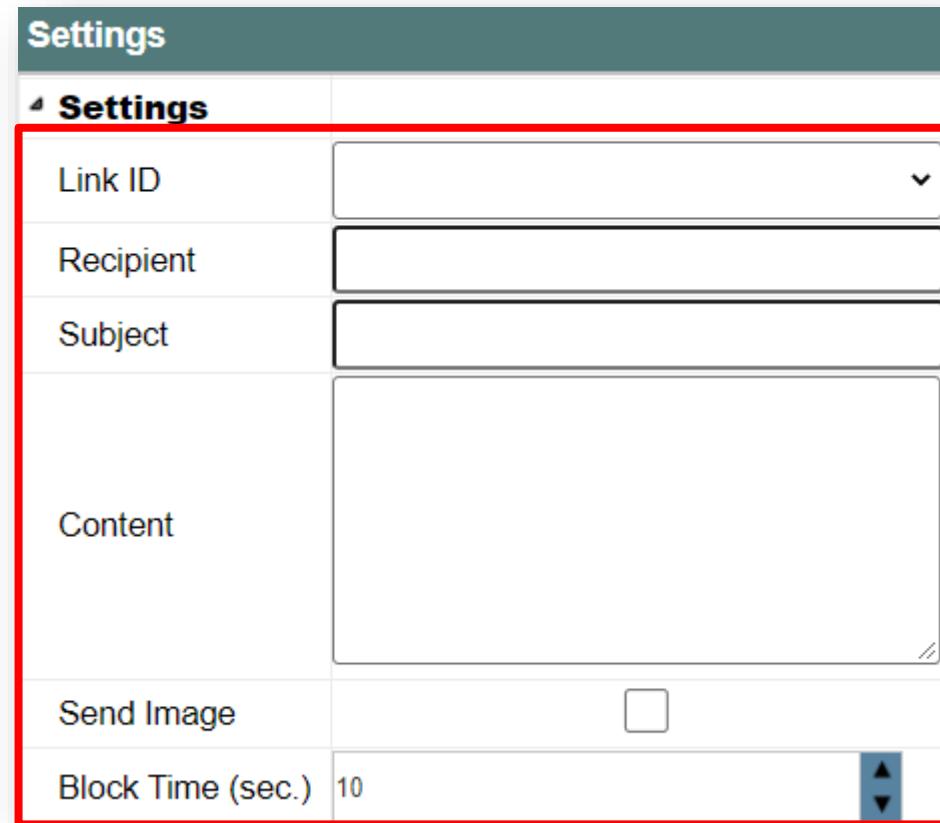


# VIC Flow 函式

- **SEND.EMAIL**

設置連結名稱、收件者、信件主旨、信件內容、是否送圖、阻隔時間

執行此運算子即可發送電子郵件



# VIC Flow 函式

## • SEND.EMAIL

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置要使用的SMTP連線	
<b>收件者 (Recipient)</b>	設置欲收件的信箱E-Mail位址	String
<b>主旨 (Subject)</b>	設置欲傳送的信件標題	String
<b>內容 (Content)</b>	設置欲發送的信件內容 若欄位不為空，則套用欄位設定的值	Inline Python Expression String and %1
<b>送圖 (Send Image)</b>	設置是否寄送當下辨識畫面 勾選即寄送	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

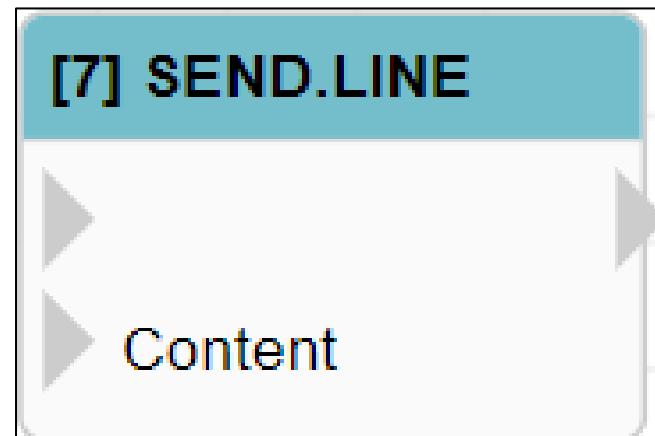
**Settings**

Name	Value
<b>Information</b>	
ID	FC_SEND_EMAIL_47
Name	SEND.EMAIL
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	
Recipient	
Subject	
Content	
Send Image	<input type="checkbox"/>
Block Time (sec.)	10

- **SEND.LINE**

可將輸入端或設置的內容透過Line Notify發送通知的運算子

	UI名稱	行為
輸入端		
	Content	輸入欲發送的內容
輸出端		



- **SEND.LINE**

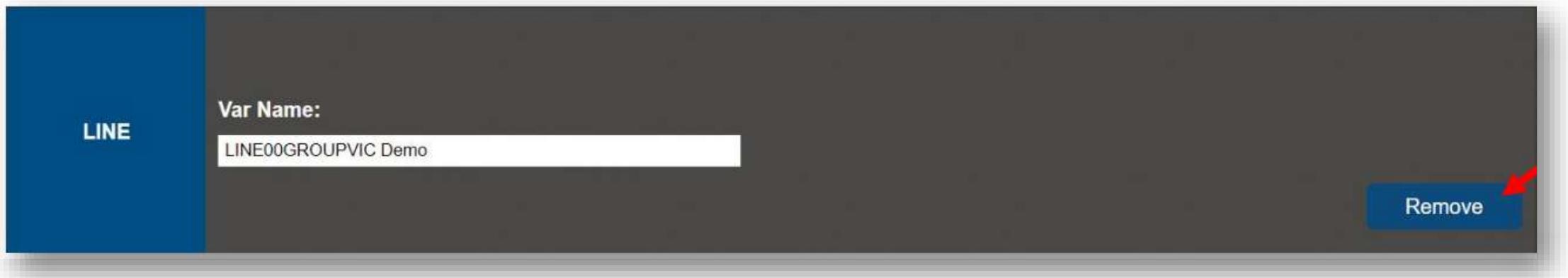
新增連線 – LINE Notify：進入連結頁面、通訊模式選擇LINE Notify、輸入權杖和自定義名稱、點擊新增



# VIC Flow 函式

- **SEND.LINE**

新增連線 – LINE Notify：下方確認已新增LINE連線，可在此移除連線

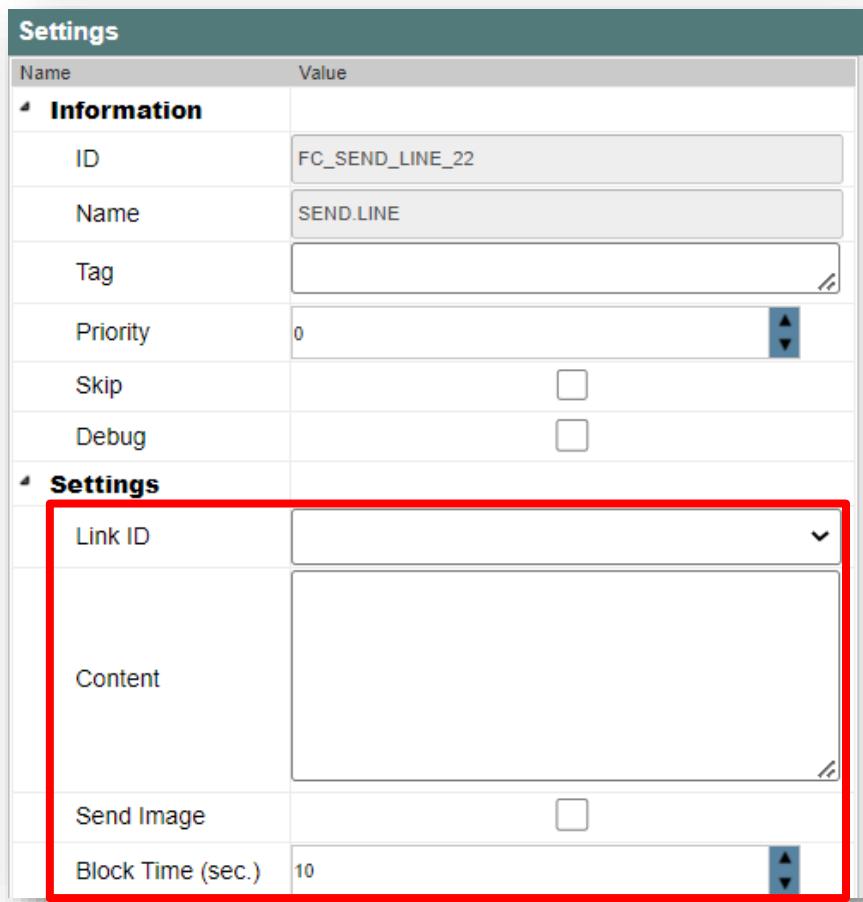


# VIC Flow 函式

- **SEND.LINE**

輸入連結名稱、訊息內容、是否送圖、阻隔時間

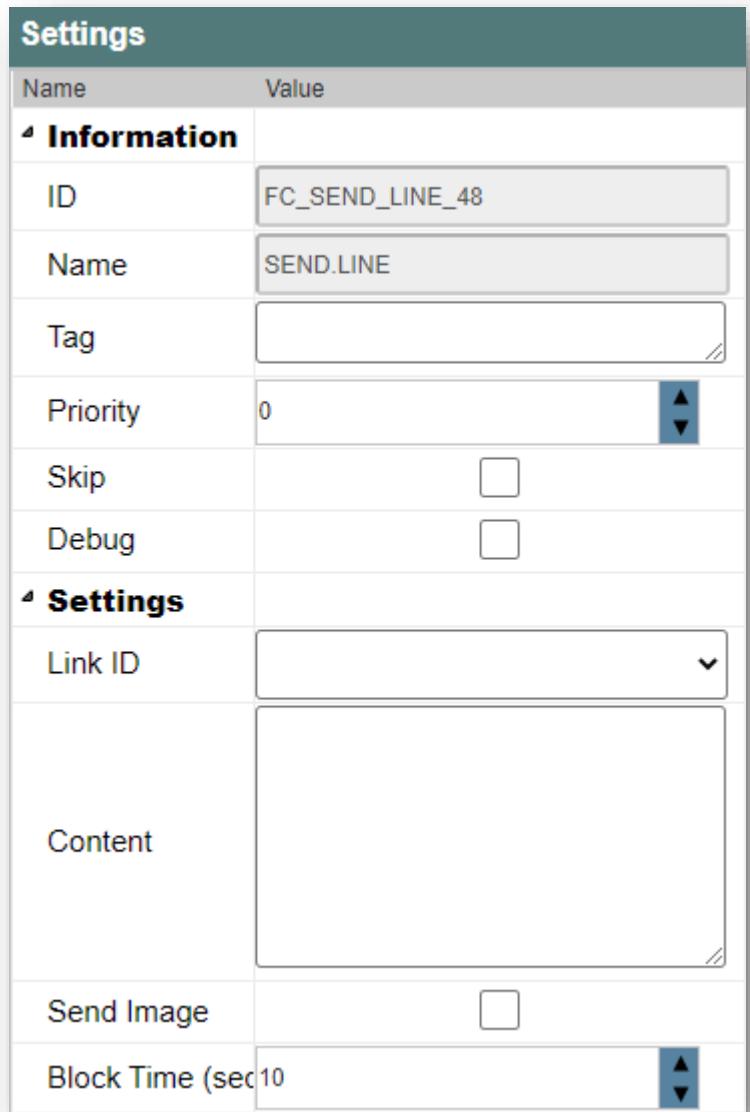
執行此運算子即可發送 LINE 訊息



# VIC Flow 函式

## • SEND.LINE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置要使用的Line Notify連線	
<b>內容 (Content)</b>	設置欲發送的內容 若欄位不為空，則套用欄位設定的值	Inline Python Expression String and %1
<b>送圖 (Send Image)</b>	設置是否寄送當下辨識畫面 勾選即寄送	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer



- **SEND.WECHAT**

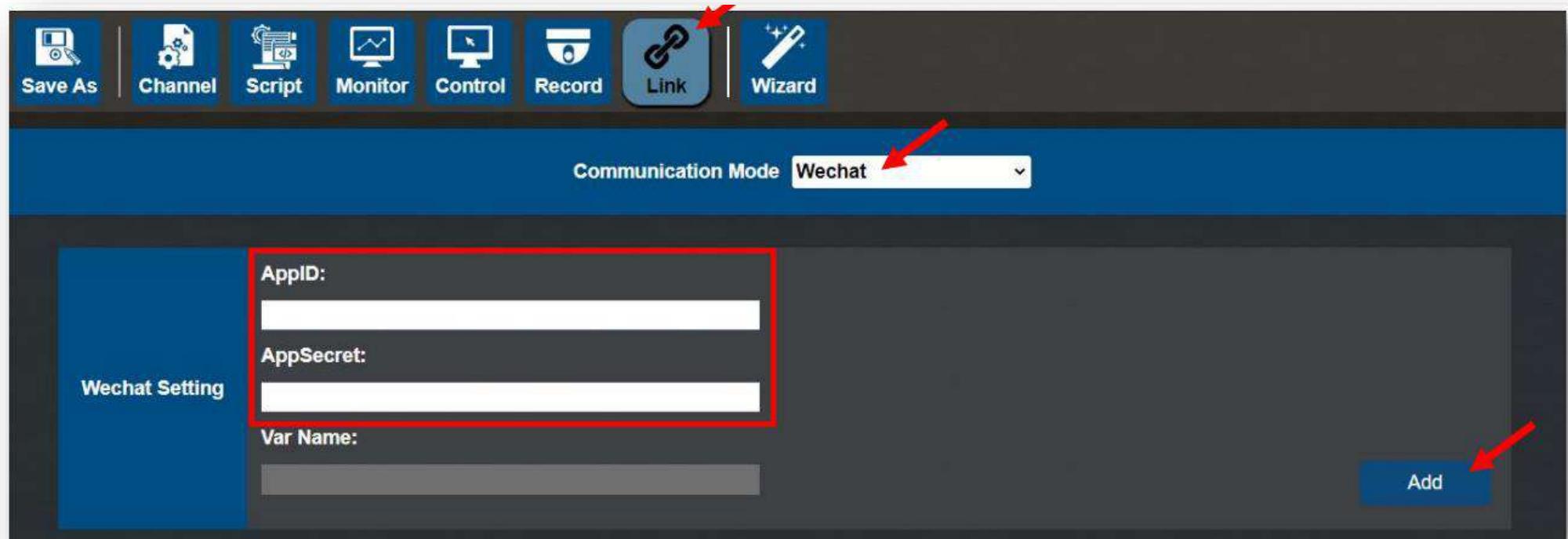
可將輸入端或設置的內容透過微信公眾號發送訊息給關注該公眾號的每位使用者的運算子

	UI名稱	行為
輸入端		
輸出端		



- **SEND.WECHAT**

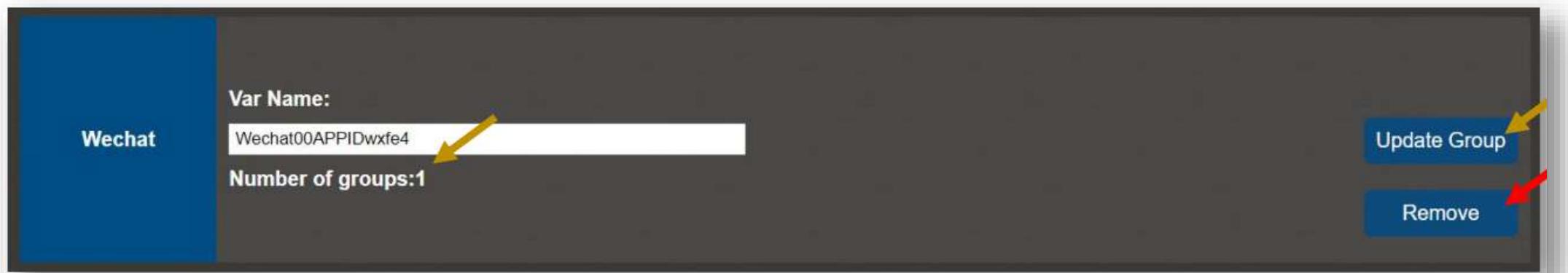
新增連線 – WeChat：進入連結頁面、通訊模式選擇WeChat、輸入appID、appsecret、點擊新增



- **SEND.WECHAT**

新增連線 – WeChat：下方確認已新增WeChat連線，可在此移除連線

若有新的用戶關注公眾號，需要點擊更新人數更新公眾號人數

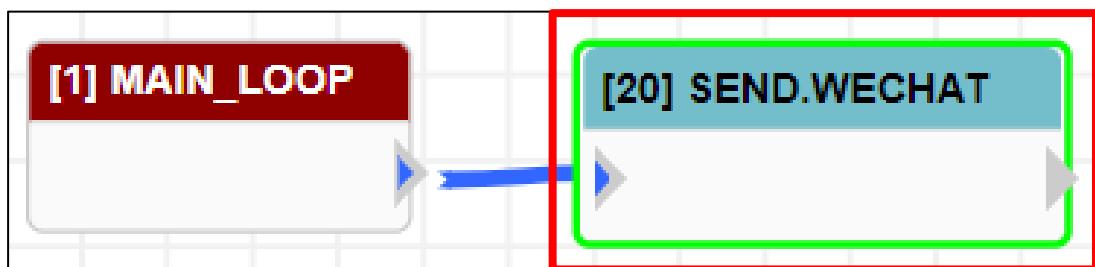


# VIC Flow 函式

## • SEND.WECHAT

輸入連結名稱、Template ID、keyword1、keyword2、keyword3、是否送圖、阻隔時間

執行此運算子即可發送 Wechat 訊息



Settings

Name	Value
Information	
ID	FC_SEND_WECHAT_20
Name	SEND.WECHAT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Settings	
Link ID	
Template ID	
keyword1	
keyword2	
keyword3	
Send Image	<input type="checkbox"/>
Block Time (sec.)	10

# VIC Flow 函式

## • SEND.WECHAT

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置要使用的Wechat連線	
<b>Template ID</b>	欲傳送的訊息模板ID	String
<b>keyword1</b>	設定欲傳送模板內容的keyword1資料	Inline Python Expression String
<b>keyword2</b>	設定欲傳送模板內容的keyword2資料	Inline Python Expression String
<b>keyword3</b>	設定欲傳送模板內容的keyword3資料	Inline Python Expression String
<b>送圖 (Send Image)</b>	設置是否寄送當下辨識畫面 勾選即寄送	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

Settings	
Name	Value
<b>Information</b>	
ID	FC_SEND_WECHAT_20
Name	SEND.WECHAT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	
Template ID	
Keyword1	
Keyword2	
Keyword3	
Send Image	<input type="checkbox"/>
Block Time (sec.)	10

- **SEND.WECHAT\_P**

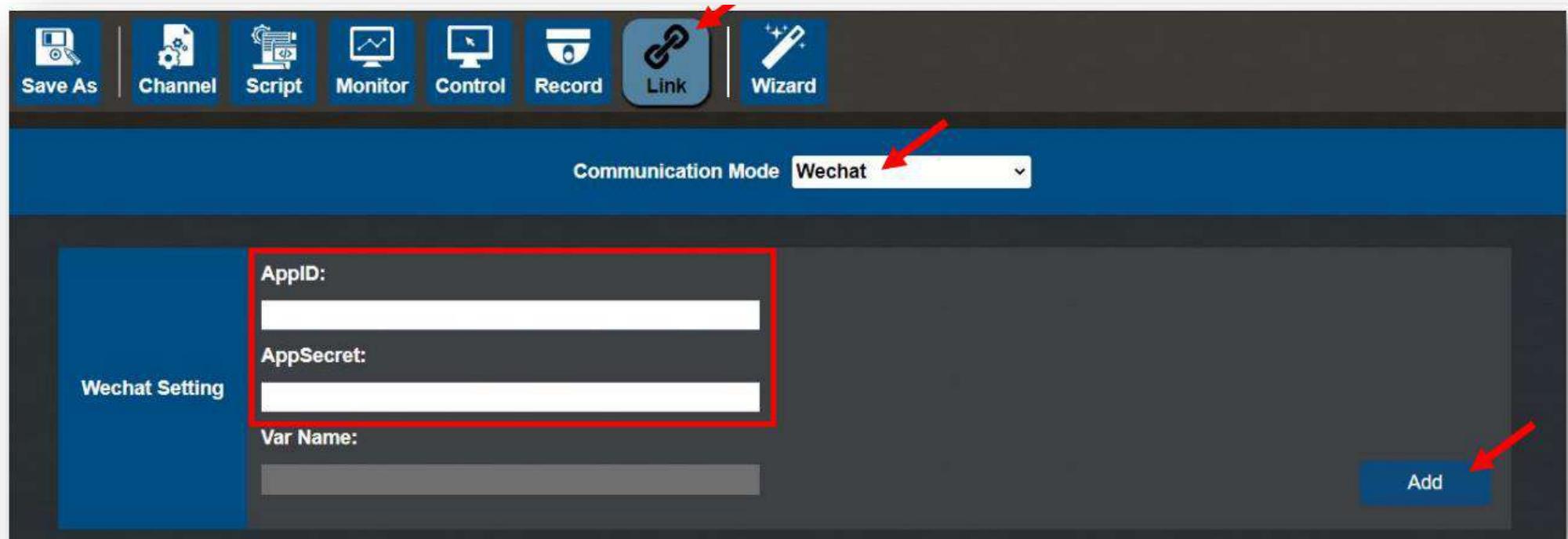
可將輸入端或設置的內容透過微信公眾號發送訊息給關注該公眾號的特定使用者的運算子

	UI名稱	行為
輸入端		
輸出端		



- **SEND.WECHAT**

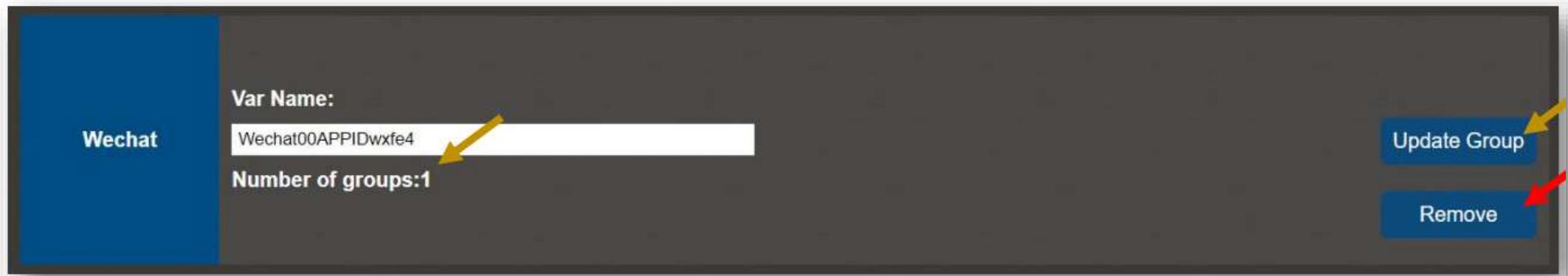
新增連線 – WeChat：進入連結頁面、通訊模式選擇WeChat、輸入appID、appsecret、點擊新增



- **SEND.WECHAT**

新增連線 – WeChat：下方確認已新增WeChat連線，可在此移除連線

若有新的用戶關注公眾號，需要點擊更新人數更新公眾號人數



- **SEND.WECHAT\_P**

輸入連結名稱、Template ID、Open ID、keyword1、keyword2、keyword3、是否送圖、阻隔時間  
執行此運算子即可發送 Wechat 訊息



Settings	
Name	Value
<b>Information</b>	
ID	FC_SEND_WECHAT_P_50
Name	SEND.WECHAT_P
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	
Template ID	
Open ID	
Keyword1	
Keyword2	
Keyword3	
Send Image	<input type="checkbox"/>
Block Time (sec)	10

# VIC Flow 函式

## • SEND.WECHAT\_P

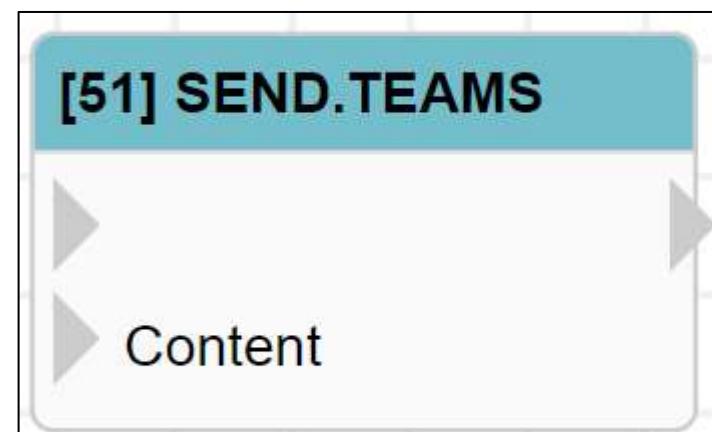
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置要使用的Wechat連線	
<b>Template ID</b>	欲傳送的訊息模板ID	String
<b>Open ID</b>	欲傳送關注公眾號中該用戶的微信號	String
<b>keyword1</b>	設定欲傳送模板內容的 <b>keyword1</b> 資料	Inline Python Expression String
<b>keyword2</b>	設定欲傳送模板內容的 <b>keyword2</b> 資料	Inline Python Expression String
<b>keyword3</b>	設定欲傳送模板內容的 <b>keyword3</b> 資料	Inline Python Expression String
<b>送圖 (Send Image)</b>	設置是否寄送當下辨識畫面 勾選即寄送	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

Settings	
Name	Value
<b>Information</b>	
ID	FC_SEND_WECHAT_P_50
Name	SEND.WECHAT_P
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	
Template ID	
Open ID	
Keyword1	
Keyword2	
Keyword3	
Send Image	<input type="checkbox"/>
Block Time (sec)	10

- **SEND.TEAMS**

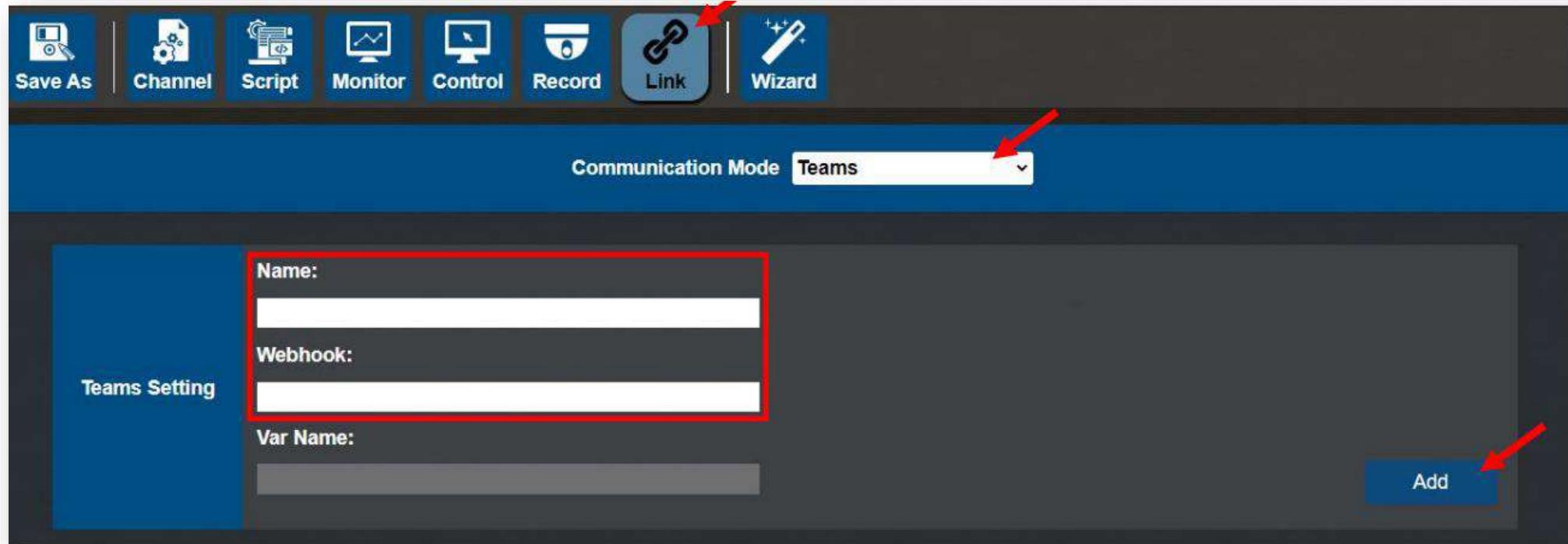
可將輸入端或設置的內容透過Microsoft Teams發送通知的運算子

	UI名稱	行為
輸入端	Content	輸入欲發送的內容
輸出端		



- **SEND.TEAMS**

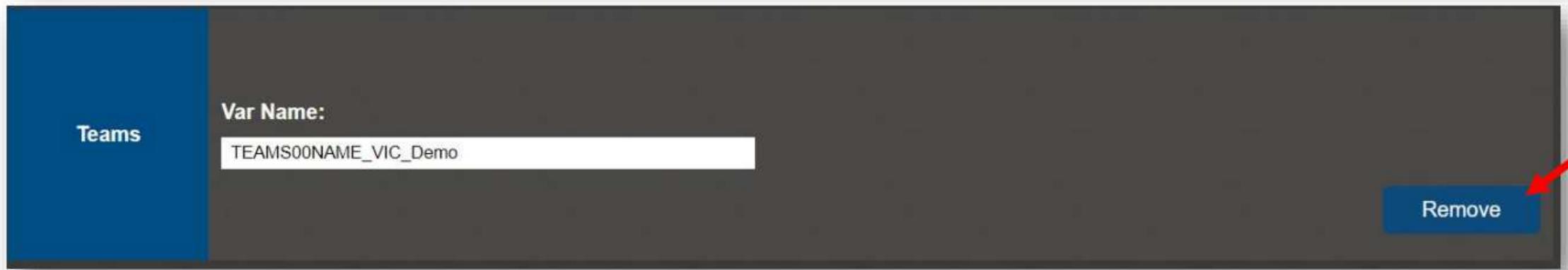
新增連線 – Teams：進入連結頁面、通訊模式選擇TEAMS、輸入名稱、Webhook、點擊新增



# VIC Flow 函式

- **SEND.TEAMS**

新增連線 – Teams : 下方確認已新增Teams連線，可在此移除連線

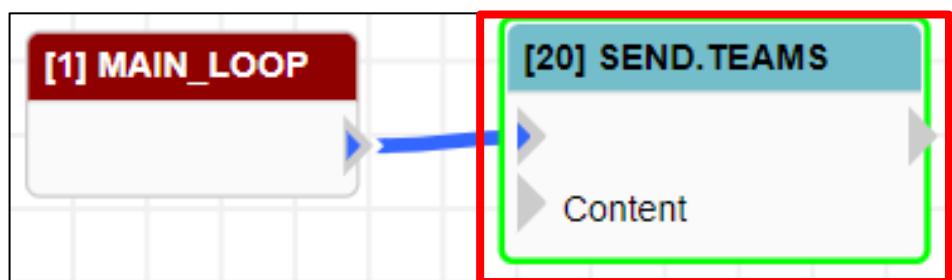


# VIC Flow 函式

## • SEND.TEAMS

輸入連結名稱、標題、內容、是否送圖、阻隔時間

執行此運算子即可發送 Teams 訊息



**Settings**

Name	Value
<b>Information</b>	
ID	FC_SEND_TEAMS_20
Name	SEND.TEAMS
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	
Title	
Content	
Send Image	<input type="checkbox"/>
Block Time (sec.)	10

# VIC Flow 函式

## • SEND.TEAMS

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>連結名稱 (Link ID)</b>	設置要使用的Teams	
<b>標題 (Title)</b>	欲傳送的訊息標題 此欄位不可為空	Inline Python Expression String
<b>內容 (Content)</b>	欲傳送的訊息內容 若欄位不為空，則套用欄位設定的值	Inline Python Expression String and %1
<b>送圖 (Send Image)</b>	設置是否寄送當下辨識畫面 勾選即寄送	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

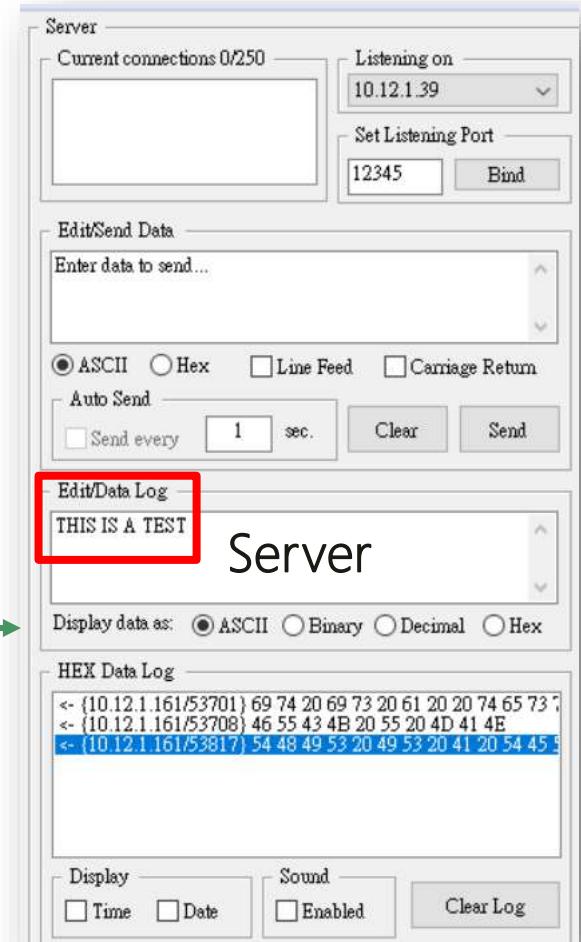
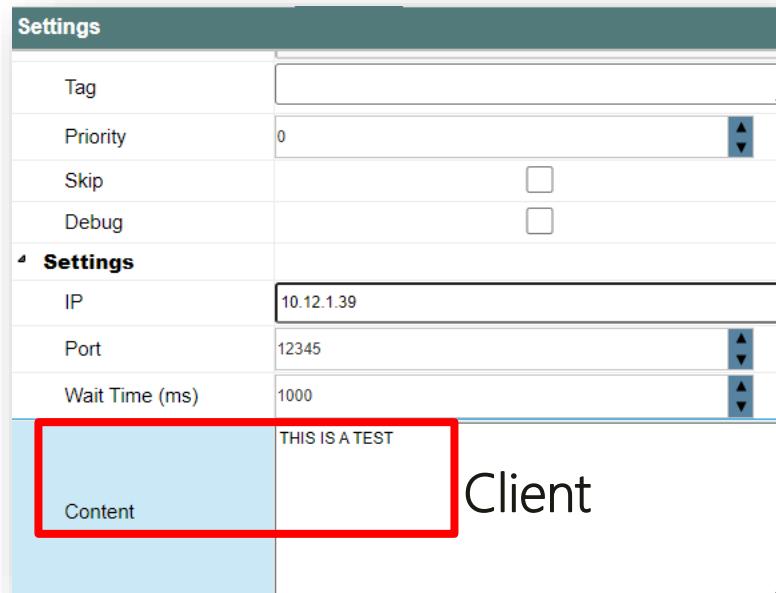
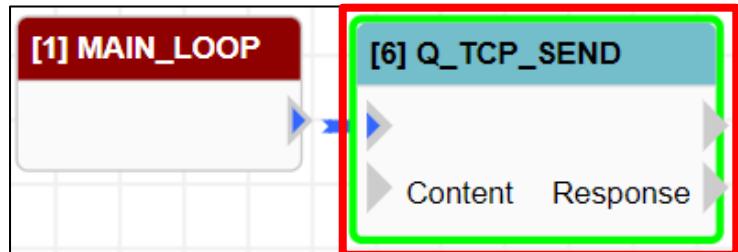
Settings	
Name	Value
<b>Information</b>	
ID	FC_SEND_TEAMS_20
Name	SEND.TEAMS
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Link ID	
Title	
Content	
Send Image	<input type="checkbox"/>
Block Time (sec.)	10

# VIC Flow 函式

## • Q\_TCP\_SEND

可將輸入端或設置的內容透過TCP/IP Client傳送的運算子

	UI名稱	行為
輸入端	Content	輸入欲發送的內容
	Response	輸出Host回傳的內容



# VIC Flow 函式

- Q\_TCP\_SEND

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>IP</b>	設置欲連接的Host IP位址	String
<b>通訊埠 (Port)</b>	設置欲連接的Host通訊埠	Integer
<b>等待時間(ms) (Wait Time (ms))</b>	設置下次執行須等待的時間	Integer
<b>內容 (Content)</b>	設置欲傳送的內容 若欄位不為空，則套用欄位設定的值	String

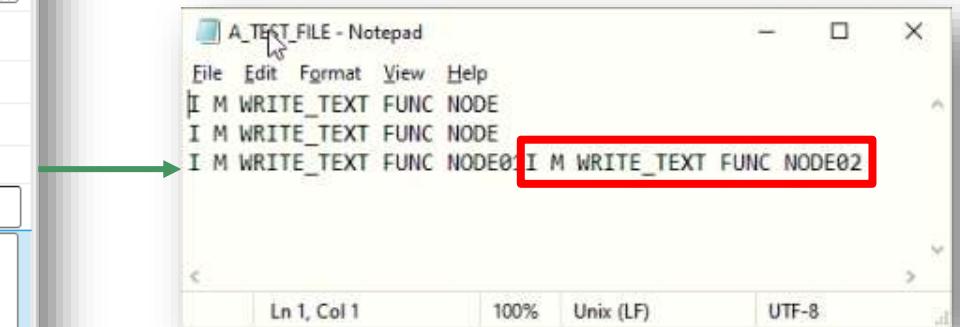
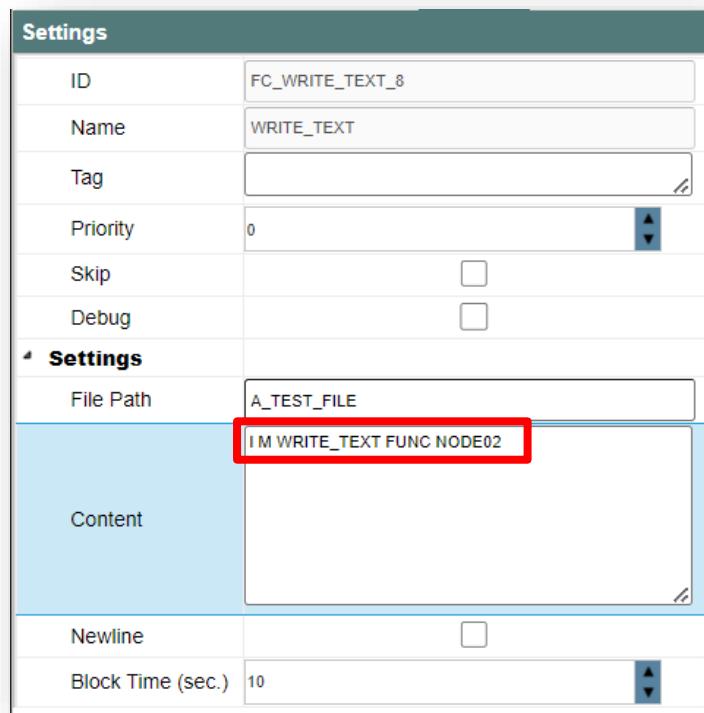
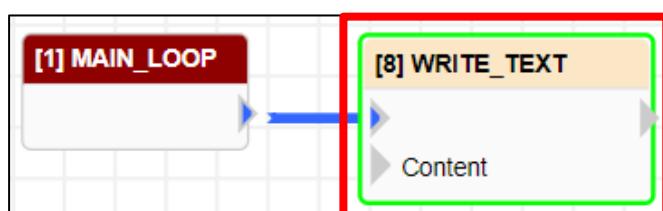
**Settings**

Name	Value
<b>Information</b>	
ID	FC_Q_TCP_SEND_53
Name	Q_TCP_SEND
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
IP	127.0.0.1
Port	8000
Wait Time (ms)	1000
Content	

## • WRITE.TEXT

可將輸入端或設置的內容儲存成文字檔的運算子

	UI名稱	行為
輸入端		
	Content	輸入欲儲存的文字內容
輸出端		



# VIC Flow 函式

## • WRITE.TEXT

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>檔案路徑 (File Path)</b>	設置文字檔的檔案路徑位置 若只設置檔案名稱，則檔案會建立在 C:\VIC7000\Export資料夾內	String
<b>內容 (Content)</b>	設置欲寫入文字檔的內容 若欄位不為空，則套用欄位設定的值	Inline Python Expression String and %1
<b>換行 (Newline)</b>	設置每次執行是否要換行 若勾選，執行寫入後會自動換行	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

Settings	
Name	Value
<b>Information</b>	
ID	FC_WRITE_TEXT_54
Name	WRITE_TEXT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
File Path	
Content	
Newline	<input checked="" type="checkbox"/>
Block Time (sec.)	10

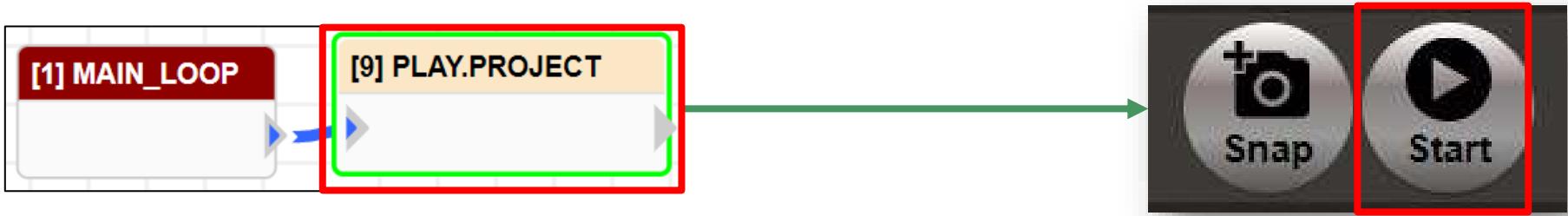
# VIC Flow 函式

- **PLAY.PROJECT**

可控制是否運行程式的運算子

PLAY.PROJECT 如同點擊專案啟動按鈕，即可啟動專案

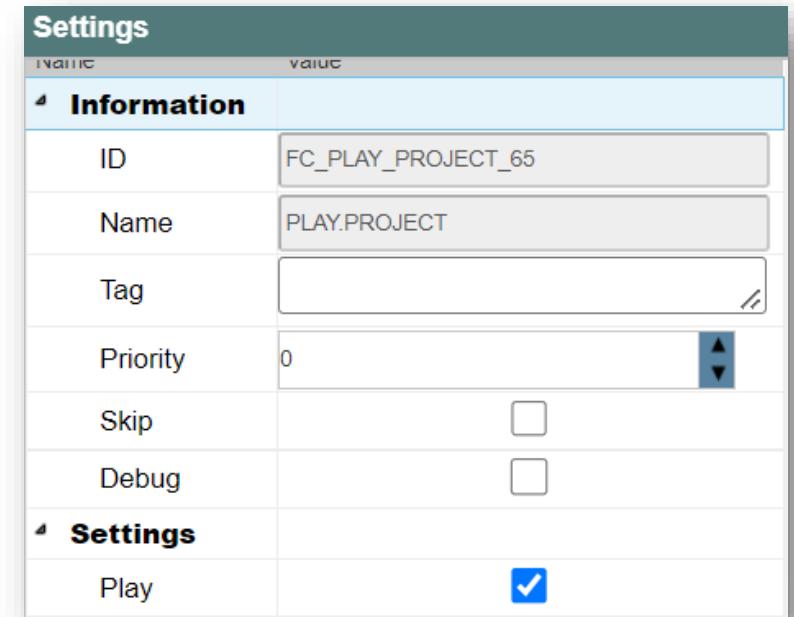
	UI名稱	行為
輸入端		
輸出端		



# VIC Flow 函式

- PLAY.PROJECT

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>播放 (Play)</b>	設置是否運行程式 勾選為運行	



# VIC Flow 函式

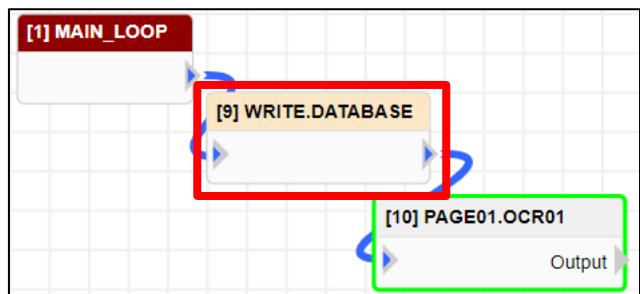
nexvia

## • WRITE.DATABASE

可控制是否將辨識資料寫入資料庫的運算子

若啟用寫入資料庫，在一般模式下執行辨識，此時資料將會寫入資料庫

	UI名稱	行為
輸入端		
輸出端		



Settings	
Information	
ID	FC_WRITE_DATABASE_9
Name	WRITE.DATABASE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Settings	
Write	<input checked="" type="checkbox"/>

# VIC Flow 函式

## • WRITE.DATABASE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>寫入 (Write)</b>	設置是否寫入資料庫 勾選為寫入	

Settings

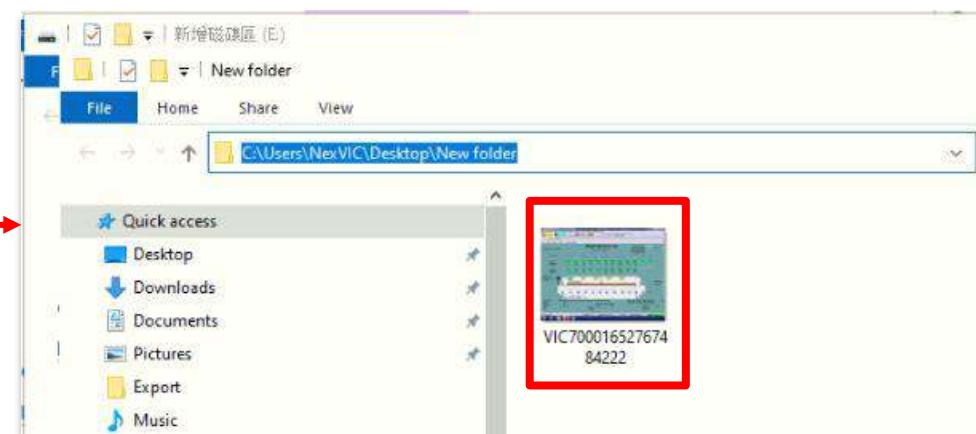
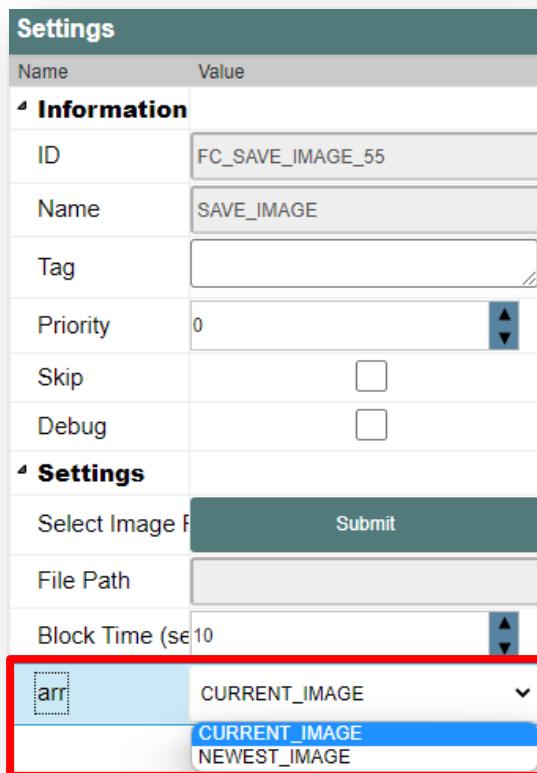
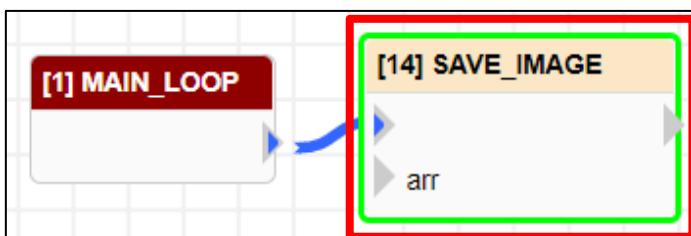
Name	Value
<b>Information</b>	
ID	FC_WRITE_DATABASE_66
Name	WRITE.DATABASE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Write	<input checked="" type="checkbox"/>

# VIC Flow 函式

## • SAVE\_IMAGE

可將辨識圖像儲存成圖檔的運算子

輸入端	UI名稱	行為
	arr	輸入欲儲存的圖像
輸出端		



# VIC Flow 函式

- **SAVE\_IMAGE**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>檔案路徑 (File Path)</b>	設置儲存的檔案路徑位置	String
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer
<b>arr</b>	若無輸入端，即套用此欄位的設置的圖片 可以設置CURRENT_IMAGE或是 NEWEST_IMAGE	

**Settings**

Name	Value
<b>Information</b>	
ID	FC_SAVE_IMAGE_55
Name	SAVE_IMAGE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Select Image File	<input type="button" value="Submit"/>
File Path	
Block Time (sec.)	10
arr	CURRENT_IMAGE

# VIC Flow 函式

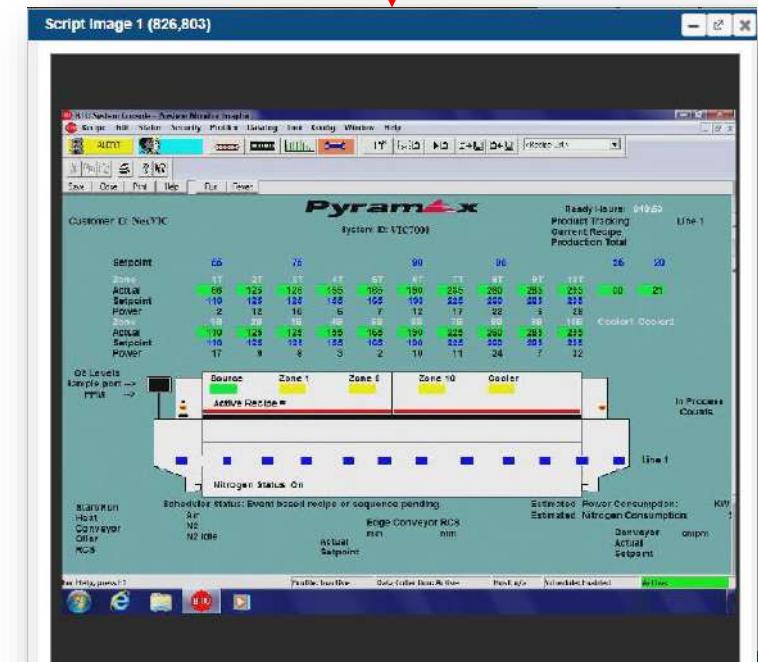
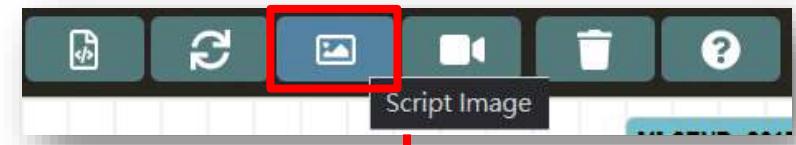
- **SHOW.IMAGE**

可將圖像顯示在腳本圖像顯示視窗中的運算子

	UI名稱	行為
輸入端	arr	輸入欲顯示的圖像
輸出端		



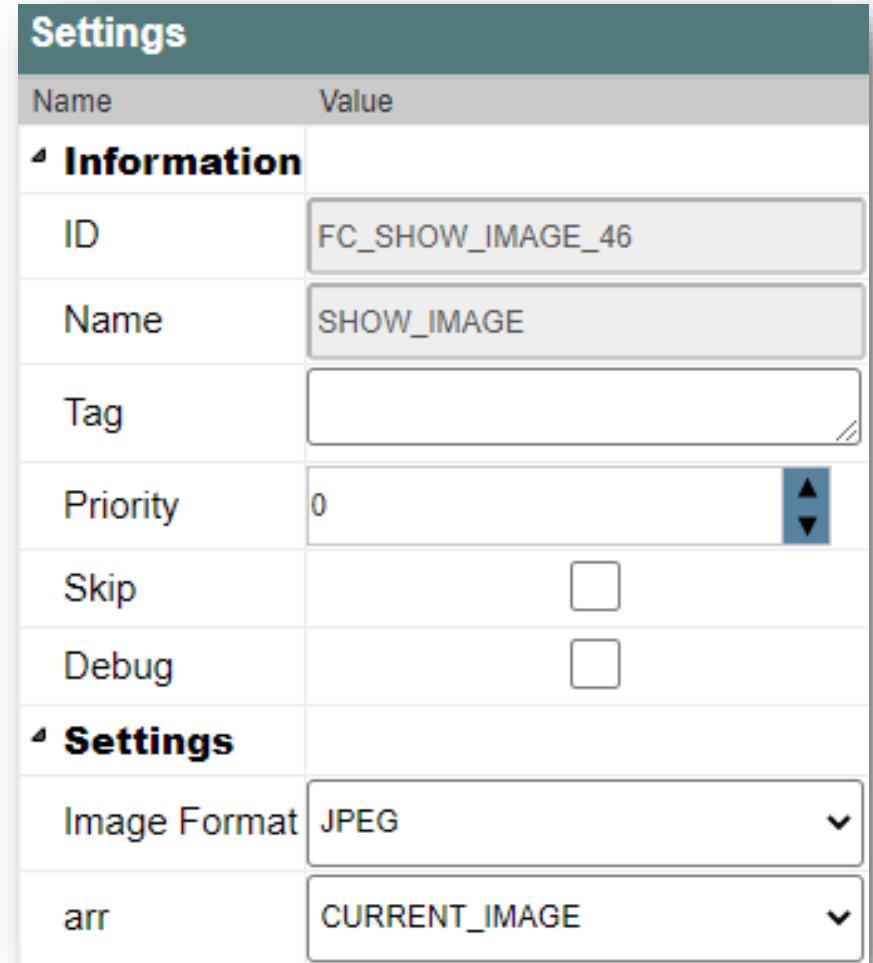
Name	Value
<b>Information</b>	
ID	FC_SHOW_IMAGE_46
Name	SHOW_IMAGE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Image Format	JPEG
arr	CURRENT_IMAGE
	CURRENT_IMAGE
	NEWEST_IMAGE



# VIC Flow 函式

- SHOW.IMAGE

名稱	意義	格式
<b>訊息 (Information)</b>		
ID	運算子的ID	
名稱 (Name)	運算子的名稱	
標籤 (Tag)	設定運算子的標籤	String
優先順序 (Priority)	設置同層級運算子的執行優先順序	Integer
略過 (Skip)	勾選後，即會略過執行此運算子	
偵錯 (Debug)	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
圖片格式 (Image Format)	設置顯示的圖片格式，包含JPEG與BMP	
arr	若無輸入端，即套用此欄位的設置的圖片 可以設置CURRENT_IMAGE或是 NEWEST_IMAGE	



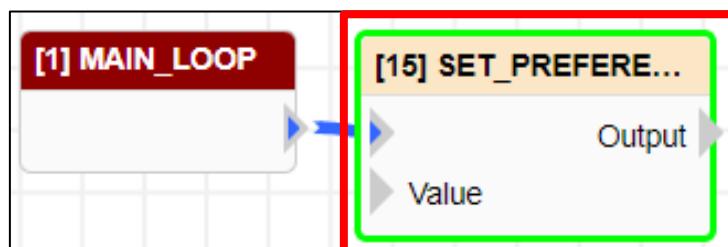
- **SET\_PREFERENCE**

可設置系統變數的運算子。即使VIC軟體重開後，系統變數依然存在，單一運算子設定一個系統變數

Key = 變數名稱

Value = 設定值

	UI名稱	行為
輸入端		
	Value	輸入欲設置的參數值
輸出端	Output	輸出設置的參數值



**Settings**

Name	Value
<b>Information</b>	
ID	FC_SET_PREFERENCE_56
Name	SET_PREFERENCE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Key	abc
Value	123

# VIC Flow 函式

## • SET\_PREFERENCE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>鍵 (Key)</b>	設置參數名稱	String
<b>值 (Value)</b>	設置參數值 若欄位不為空，則套用欄位設定的值	Inline Python Expression String and %1

**Settings**

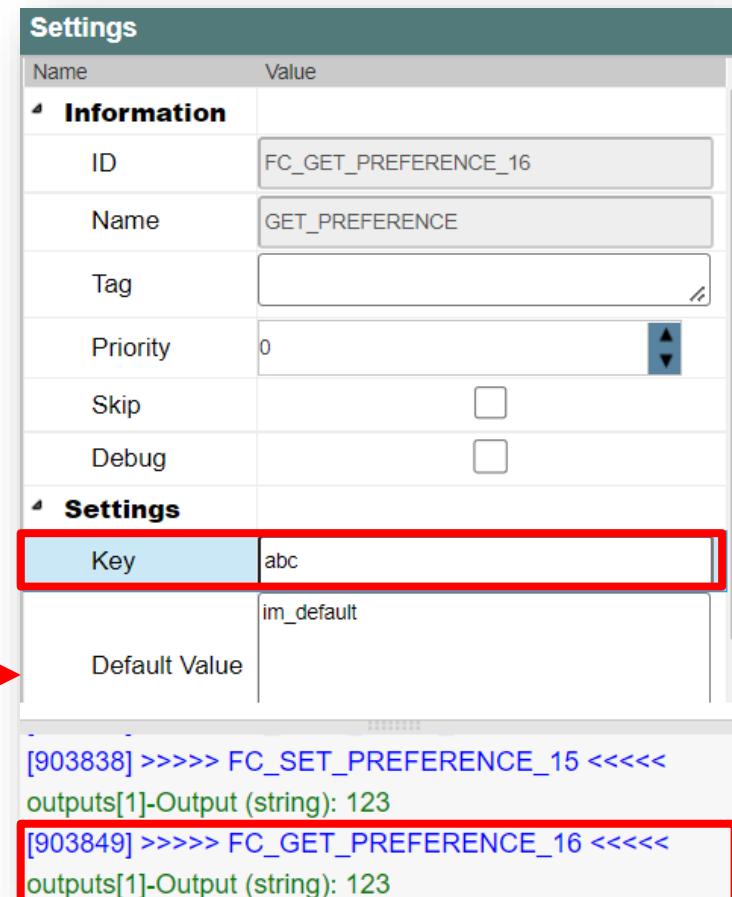
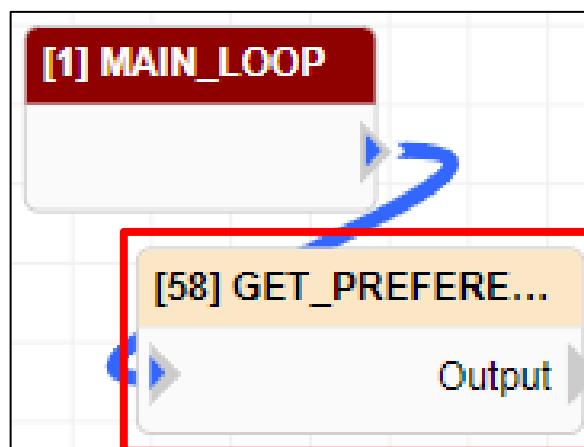
Name	Value
<b>Information</b>	
ID	FC_SET_PREFERENCE_57
Name	SET_PREFERENCE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Key	
Value	

## • GET\_PREFERENCE

可以取得系統變數值的運算子，若無該變數將會回傳預設值(Default Value)

單一運算子取得一個系統變數

	UI名稱	行為
輸入端		
輸出端	Output	若無該變數，則會輸出設置的預設值 反之若有該變數，則會輸出該變數儲存的數值



# VIC Flow 函式

## • GET\_PREFERENCE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>鍵 (Key)</b>	設置參數名稱	String
<b>預設值 (Default Value)</b>	設置該參數預設值	Inline Python Expression String

Settings

Name	Value
<b>Information</b>	
ID	FC_GET_PREFERENCE_58
Name	GET_PREFERENCE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Key	
Default Value	

- ADD +

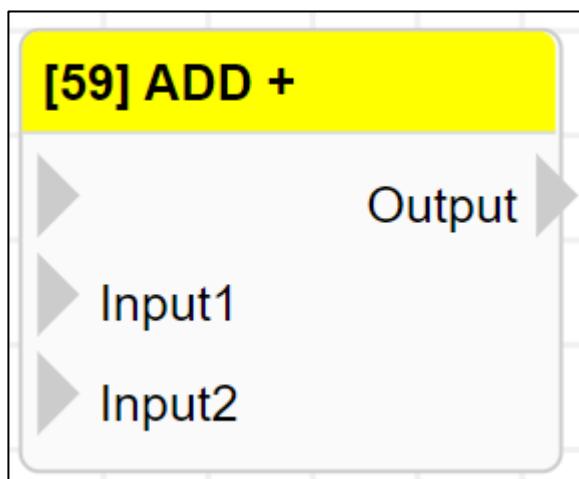
可將兩數值相加的運算子

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行相加的數值1
	Input2	輸入欲進行相加的數值2
輸出端	Output	輸出數值相加結果



```
[638637] >>> FC_OP_ARITHMETIC_15 <<<  
inputs[2]-Input1 (string): 123  
inputs[3]-Input2 (string): 123  
outputs[1]-Output (float): 246
```

# VIC Flow 函式

- ADD +

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲相加的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲相加的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_59
Name	ADD +
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- **SUB -**

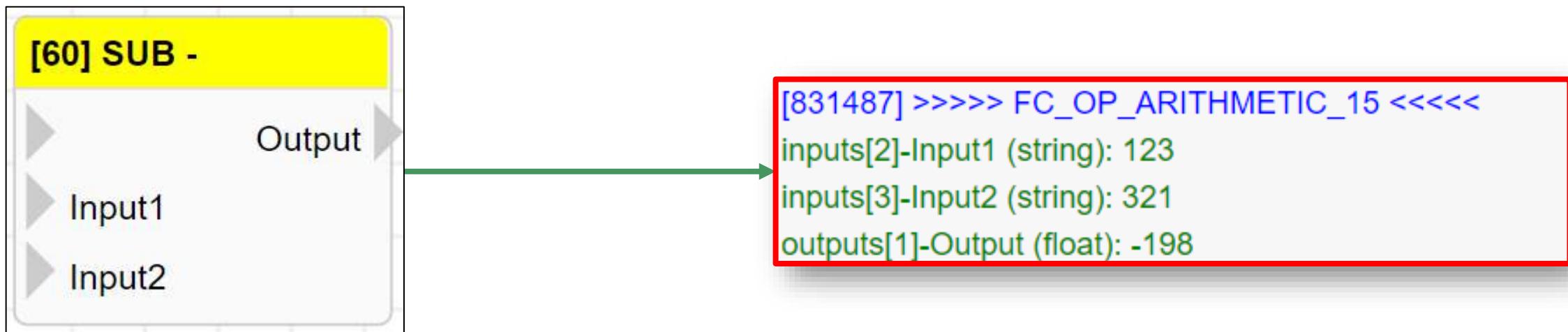
可將兩數值相減的運算子

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行相減的數值1
	Input2	輸入欲進行相減的數值2
輸出端	Output	輸出數值相減結果



# VIC Flow 函式

## • SUB -

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲相減的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲相減的數值2	Python Expression

**Settings**

Name	Value
<b>▪ Information</b>	
ID	FC_OP_ARITHMETIC_60
Name	SUB -
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>▪ Settings</b>	
Input1	
Input2	

- **MUL \***

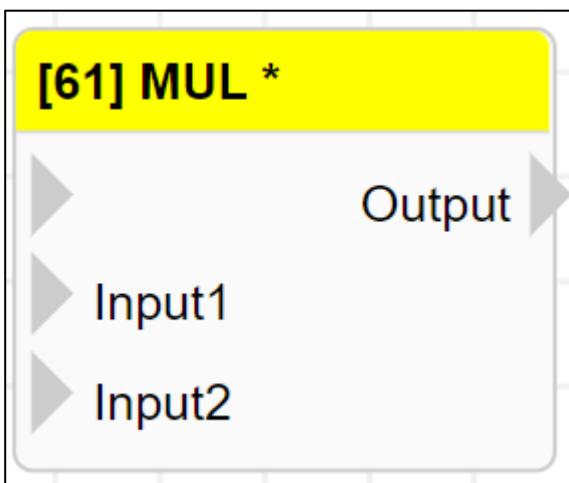
可將兩數值相乘的運算子

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行相乘的數值1
	Input2	輸入欲進行相乘的數值2
輸出端	Output	輸出數值相乘結果



```
[598355] >>>> FC_OP_ARITHMETIC_15 <<<<  
inputs[2]-Input1 (string): 123  
inputs[3]-Input2 (string): 321  
outputs[1]-Output (float): 39483
```

# VIC Flow 函式

- **MUL \***

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲相乘的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲相乘的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_61
Name	MUL *
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- DIV /

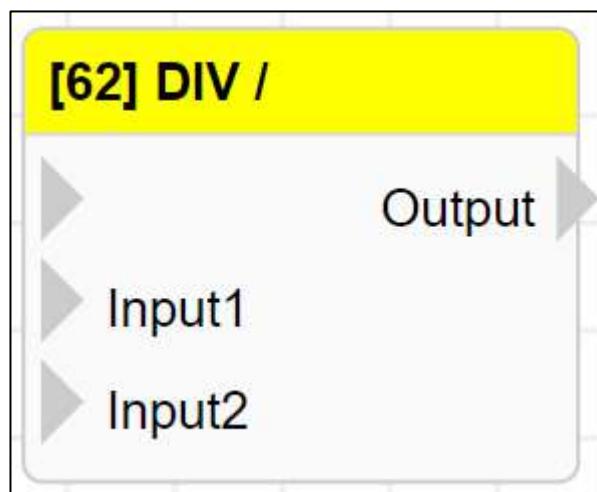
可將兩數值相除的值

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行相除的數值1
	Input2	輸入欲進行相除的數值2
輸出端	Output	輸出相除結果



```
[292909] >>>> FC_OP_ARITHMETIC_20 <<<<  
inputs[2]-Input1 (string): 123  
inputs[3]-Input2 (string): 321  
outputs[1]-Output (float): 0.38317757009345793
```

# VIC Flow 函式

- DIV /

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲相除的數值1。	Python Expression
<b>輸入2 (Input2)</b>	設置欲相除的數值2。	Python Expression

**Settings**

Name	Value
ID	FC_OP_ARITHMETIC_62
Name	DIV /
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Input1	
Input2	

- **EQU ==**

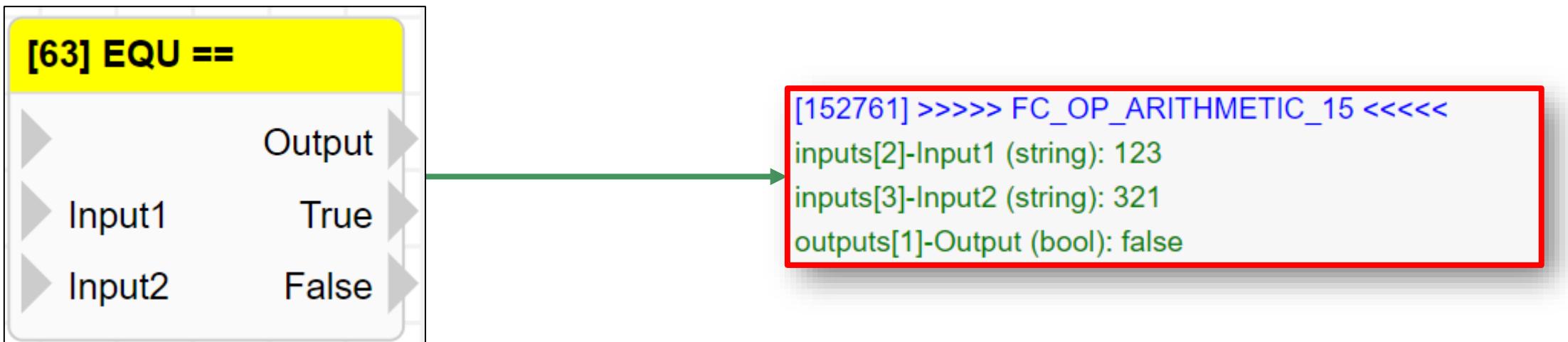
可將兩個值進行比較的運算子，若兩者相等則輸出true，反之兩者不相等則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的數值1
	Input2	輸入欲進行比較的數值2
輸出端	Output	輸出比較結果
	True	若兩數值比較相等，則由此輸出
	False	若兩數值比較不相等，則由此輸出



# VIC Flow 函式

- EQU ==

名稱	意義	格式
<b>訊息 (Information)</b>		
ID	運算子的ID	
名稱 (Name)	運算子的名稱	
標籤 (Tag)	設定運算子的標籤	String
優先順序 (Priority)	設置同層級運算子的執行優先順序	Integer
略過 (Skip)	勾選後，即會略過執行此運算子	
偵錯 (Debug)	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
輸入1 (Input1)	設置欲比較的數值1	Python Expression
輸入2 (Input2)	設置欲比較的數值2	Python Expression

**Settings**

Name	Value
ID	FC_OP_ARITHMETIC_63
Name	EQU ==
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Input1	
Input2	

- **NEQU !=**

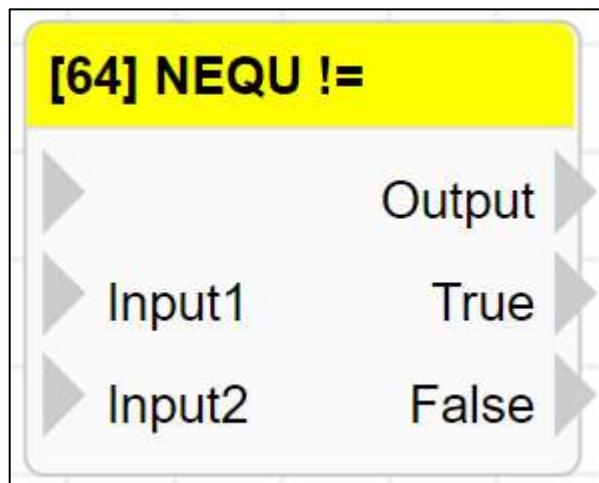
可將兩個值進行比較的運算子，若兩者不相等則輸出true；反之兩者相等則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的數值1
	Input2	輸入欲進行比較的數值2
輸出端	Output	輸出比較結果
	True	若兩數值比較不相等，則由此輸出
	False	若兩數值比較相等，則由此輸出



[300520] >>>> FC\_OP\_ARITHMETIC\_20 <<<<  
 inputs[2]-Input1 (string): 123  
 inputs[3]-Input2 (string): 321  
 outputs[1]-Output (bool): true

# VIC Flow 函式

- NEQU !=

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲比較的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲比較的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_64
Name	NEQU !=
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- LT <

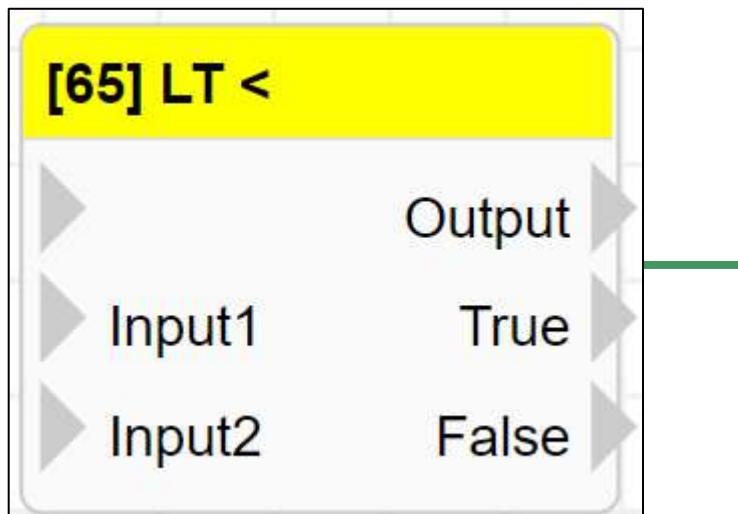
可將兩數值進行比較的運算子，若數值1<數值2則輸出true；反之則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的數值1
	Input2	輸入欲進行比較的數值2
輸出端	Output	輸出比較結果
	True	若數值1<數值2，則由此輸出
	False	若數值1≥數值2，則由此輸出



```
[300520] >>>> FC_OP_ARITHMETIC_20 <<<<
inputs[2]-Input1 (string): 123
inputs[3]-Input2 (string): 321
outputs[1]-Output (bool): true
```

# VIC Flow 函式

- LT <

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲比較的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲比較的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_65
Name	LT <
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- **LE <=**

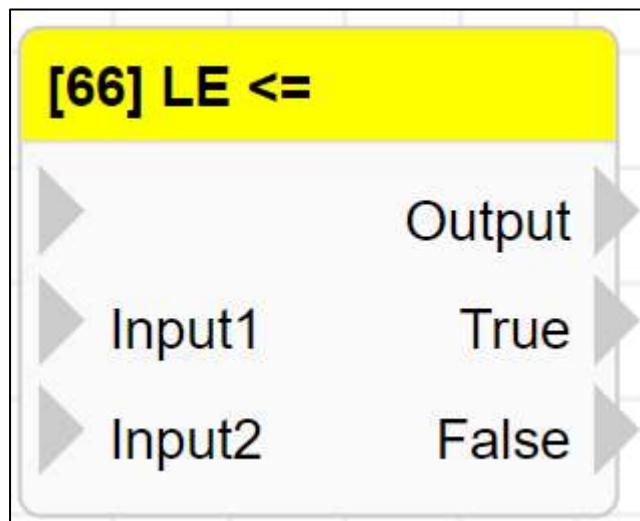
可將兩數值進行比較的運算子，若數值1≤數值2則輸出true；反之則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的數值1
	Input2	輸入欲進行比較的數值2
輸出端	Output	輸出比較結果
	True	若數值1≤數值2，則由此輸出
	False	若數值1>數值2，則由此輸出



[681161] >>>> FC\_OP\_ARITHMETIC\_20 <<<<  
inputs[2]-Input1 (string): 123  
inputs[3]-Input2 (string): 321  
outputs[1]-Output (bool): true

# VIC Flow 函式

- LE <=

名稱	意義	格式
<b>Information (訊息)</b>		
<b>ID</b>	運算子的ID	
<b>Name (名稱)</b>	運算子的名稱	
<b>Tag (標籤)</b>	設定運算子的標籤	String
<b>Priority (優先順序)</b>	設置同層級運算子的執行優先順序	Integer
<b>Skip (略過)</b>	勾選後，即會略過執行此運算子	
<b>Debug (偵錯)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>Settings (設置)</b>		
<b>Input1 (輸入1)</b>	設置欲比較的數值1	Python Expression
<b>Input2 (輸入2)</b>	設置欲比較的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_66
Name	LE <=
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- **GT >**

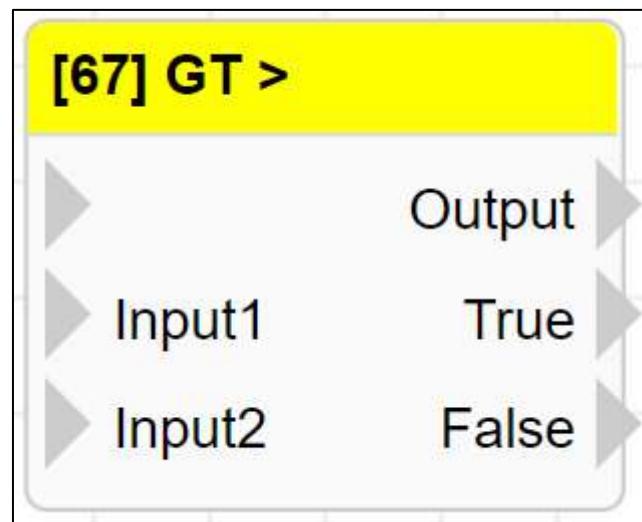
可將兩數值進行比較的運算子，若數值1>數值2則輸出true；反之則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的數值1
	Input2	輸入欲進行比較的數值2
輸出端	Output	輸出比較結果
	True	若數值1>數值2，則由此輸出
	False	若數值1≤數值2，則由此輸出



[865863] >>>> FC\_OP\_ARITHMETIC\_15 <<<<  
 inputs[2]-Input1 (string): 123  
 inputs[3]-Input2 (string): 321  
 outputs[1]-Output (bool): false

# VIC Flow 函式

- GT >

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲比較的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲比較的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_67
Name	GT >
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- **GE >=**

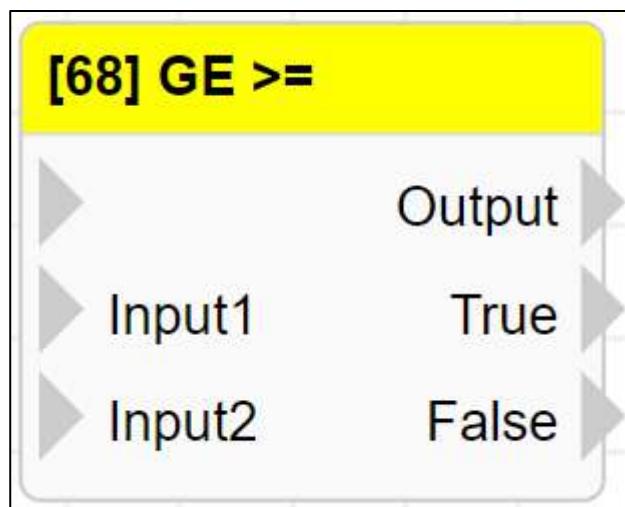
可將兩數值進行比較的運算子，若數值1 $\geq$ 數值2則輸出true；反之則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的數值1
	Input2	輸入欲進行比較的數值2
輸出端	Output	輸出比較結果
	True	若數值1 $\geq$ 數值2，則由此輸出
	False	若數值1 $<$ 數值2，則由此輸出



[080294] >>>> FC\_OP\_ARITHMETIC\_20 <<<<  
inputs[2]-Input1 (string): 123  
inputs[3]-Input2 (string): 321  
outputs[1]-Output (bool): false

# VIC Flow 函式

- GE >=

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲比較的數值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲比較的數值2	Python Expression

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_68
Name	GE >=
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

# VIC Flow 函式

- **CT =~**

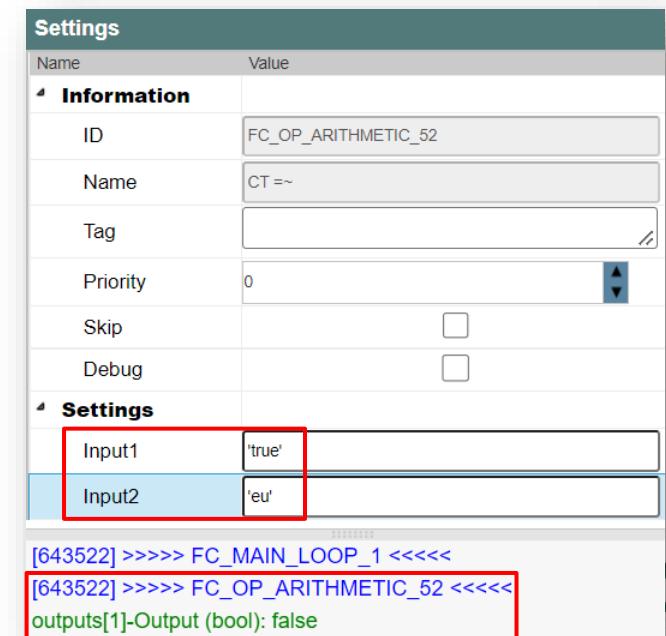
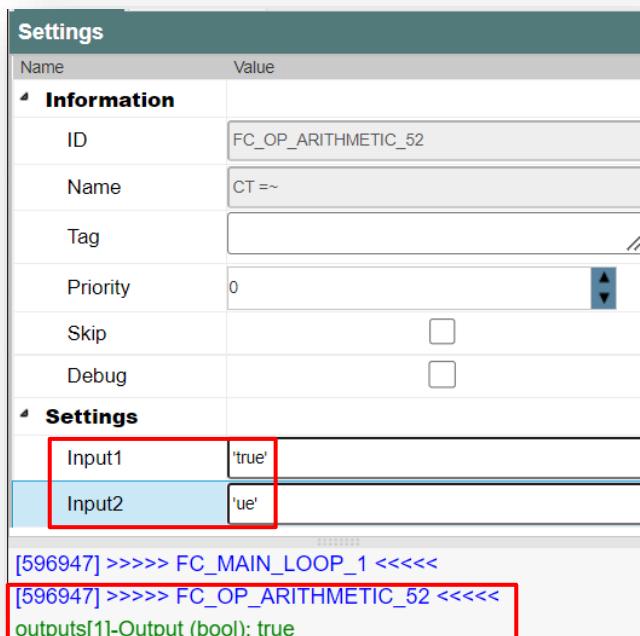
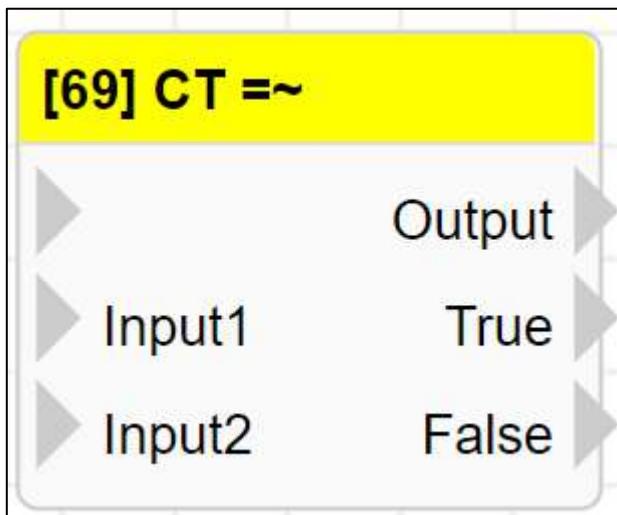
可將兩個值進行比較的運算子，若輸入值1包含輸入值2，則輸出true；反之則輸出false

若輸入端有值輸入，則套用輸入端的值

此運算子會優先將輸入端的值轉為數值進行運算

若無法轉為數值，將會以原型態進行運算

	UI名稱	行為
輸入端	Input1	輸入欲進行比較的值1
	Input2	輸入欲進行比較的值2
輸出端	Output	輸出比較結果
	True	若值1包含值2，則由此輸出
	False	若值1不包含值2，則由此輸出



# VIC Flow 函式

- CT =~

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲比較的值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲比較的值2	Python Expression

**Settings**

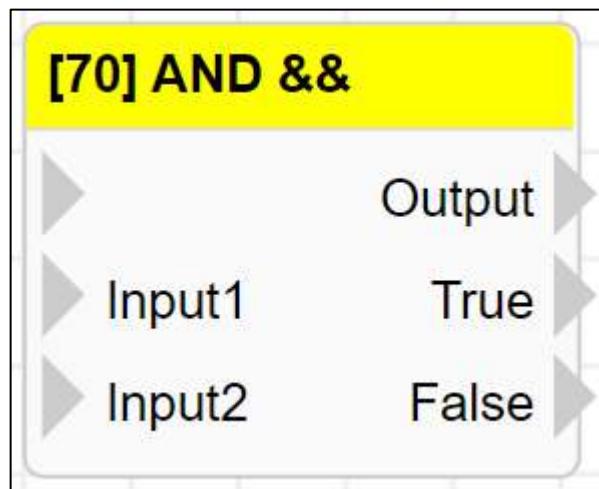
Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_69
Name	CT =~
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- **AND &&**

可將兩個值進行邏輯運算中的「且」運算的運算子，若運算結果為真(True)，則輸出true；反之為否(False)則輸出false

若輸入端有值輸入，則套用輸入端的值

	UI名稱	行為
輸入端	Input1	輸入欲進行運算的值1
	Input2	輸入欲進行運算的值2
輸出端	Output	輸出運算結果
	True	若運算結果為真，則由此輸出
	False	若運算結果為否，則由此輸出



```
[834503] >>>> FC_OP_ARITHMETIC_49 <<<<
inputs[2]-Input1 (bool): false
inputs[3]-Input2 (bool): true
outputs[1]-Output (bool): false
```

- AND &&

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>輸入1 (Input1)</b>	設置欲運算的值1	Python Expression
<b>輸入2 (Input2)</b>	設置欲運算的值2	Python Expression

**Settings**

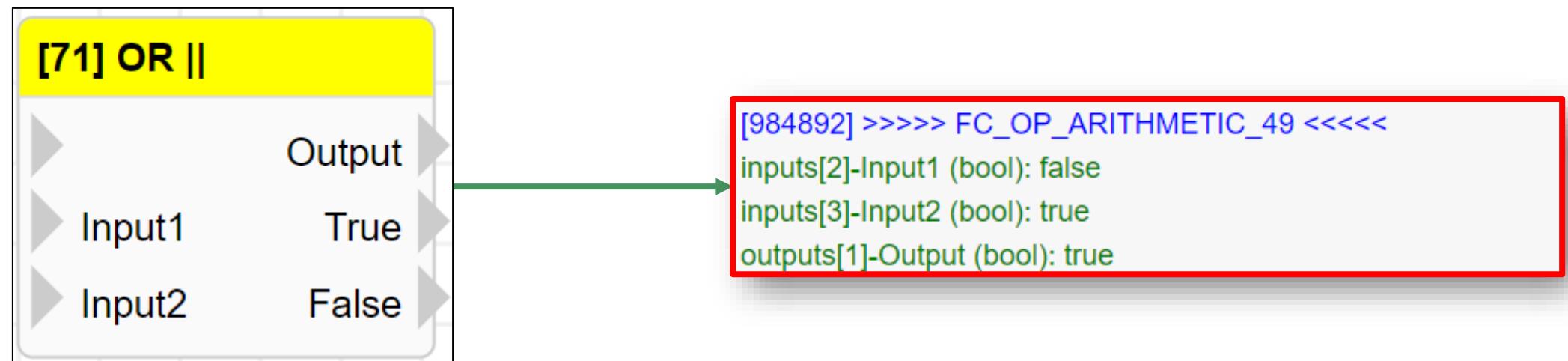
Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_70
Name	AND &&
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

- **OR ||**

可將兩個值進行邏輯運算中的「或」運算的運算子，若運算結果為真(True)，則輸出true；反之為否(False)則輸出false

若輸入端有值輸入，則套用輸入端的值

	UI名稱	行為
輸入端	Input1	輸入欲進行運算的數值1
	Input2	輸入欲進行運算的數值2
輸出端	Output	輸出運算結果
	True	若運算結果為真，則由此輸出
	False	若運算結果為否，則由此輸出



# VIC Flow 函式

- OR ||

名稱	意義	格式
<b>訊息 (Information)</b>		
ID	運算子的ID	
名稱 (Name)	運算子的名稱	
標籤 (Tag)	設定運算子的標籤	String
優先順序 (Priority)	設置同層級運算子的執行優先順序	Integer
略過 (Skip)	勾選後，即會略過執行此運算子	
偵錯 (Debug)	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
輸入1 (Input1)	設置欲運算的數值1	Python Expression
輸入2 (Input2)	設置欲運算的數值2	Python Expression

Settings

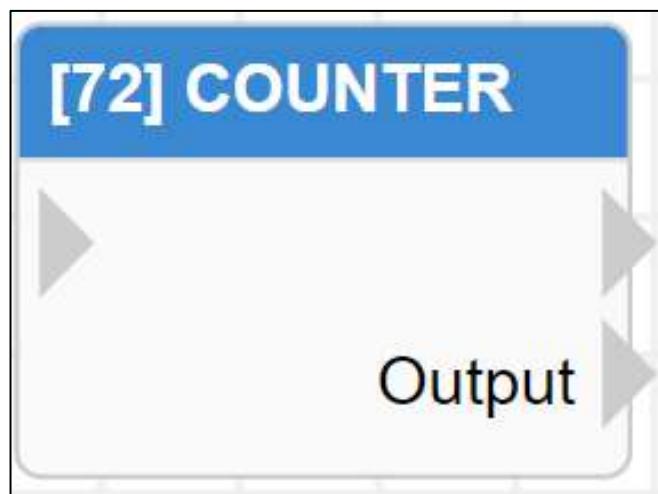
Name	Value
<b>Information</b>	
ID	FC_OP_ARITHMETIC_71
Name	OR
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Input1	
Input2	

# VIC Flow 函式

- **COUNTER**

可作為計數器的運算子

	UI名稱	行為
輸入端		
輸出端		
	Output	輸出當前計數值



```
[988867] >>>> FC_MODULE_20 [ COUNTER ] <<<<  
inputs[1] (string): 123  
outputs[2]-Output (integer): 18
```

- COUNTER

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置(Settings)</b>		
<b>重置 (Reset)</b>	重置該計數器	
<b>打印 (Print)</b>	打印該計數器當前計數值	

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODULE_72
Name	COUNTER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Reset	<b>Submit</b>
Print	<b>Submit</b>

- **RESET\_COUNTER**

可重置指定的計數器的運算子

	UI名稱	行為
輸入端		
輸出端		



# VIC Flow 函式

- **RESET\_COUNTER**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>重置 (Reset)</b>	重置指定的計數器	
<b>打印 (Print)</b>	打印指定計數器當前計數值	
<b>運算子ID (Operator ID)</b>	設置要重製的計數器的運算子ID	String

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODULE_73
Name	RESET_COUNTER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Reset	<b>Submit</b>
Print	<b>Submit</b>
Operator ID	

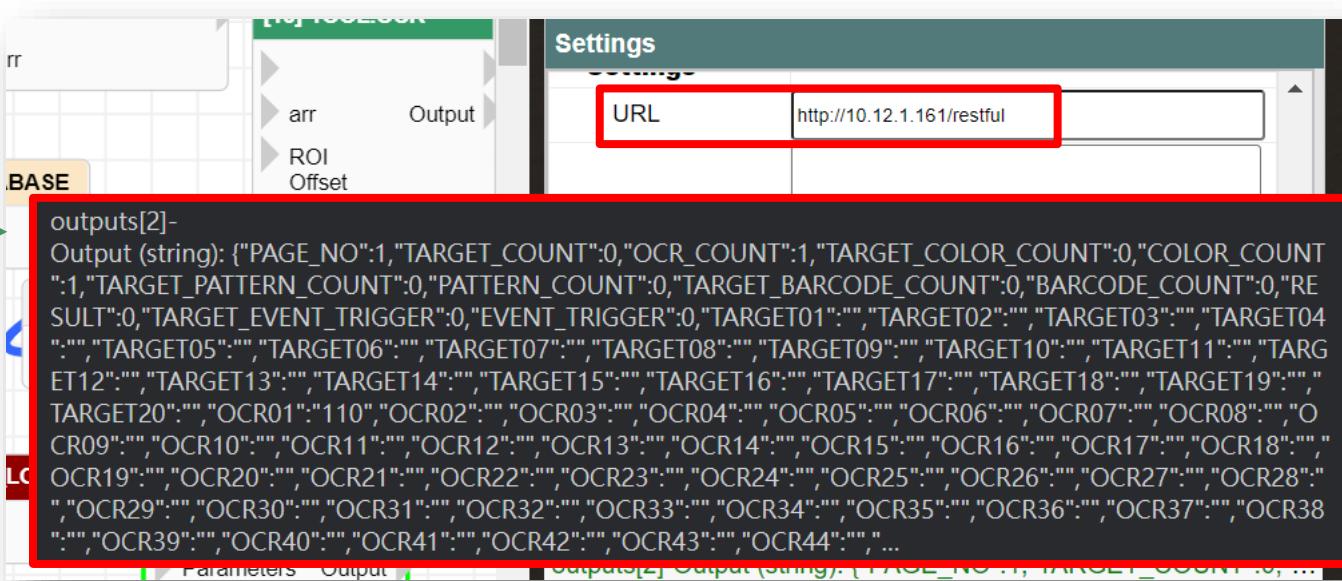
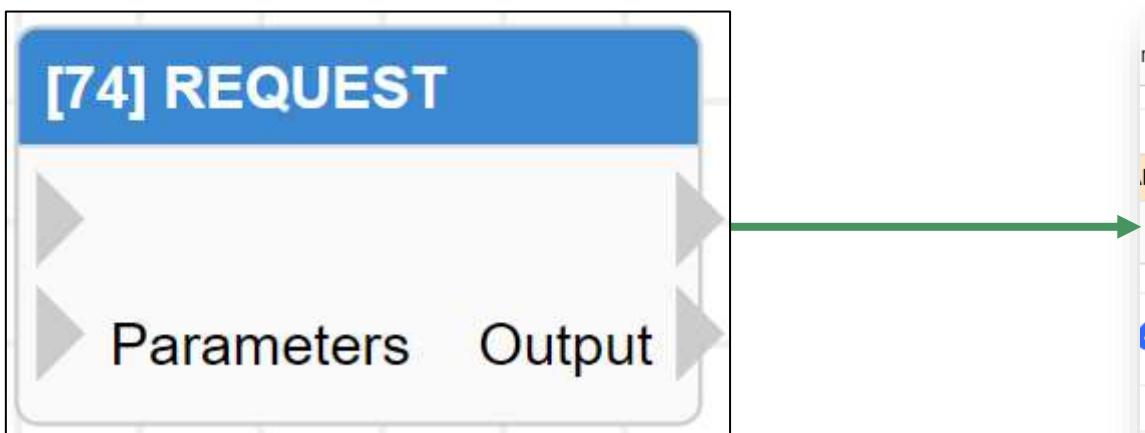
# VIC Flow 函式

## • REQUEST

可傳送RESTful要求的運算子

範例為透過RESTful取得VIC軟體的辨識結果

	UI名稱	行為
輸入端	Parameters	輸入欲傳送的參數
輸出端	Output	輸出回傳的結果



# VIC Flow 函式

## • REQUEST

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>URL</b>	設置欲傳送的URL	String
<b>設定參數 (Parameters)</b>	設定欲傳送的參數 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>方法 (Method)</b>	設定欲傳送的方法	
<b>時限(秒) (Timeout(s))</b>	設定傳送要求後等待回應的時限	Integer
<b>Authentication</b>	設定驗證方式	

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODULE_74
Name	REQUEST
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
URL	
Parameters	
Method	POST
Timeout (s)	3
Authentication	No Auth

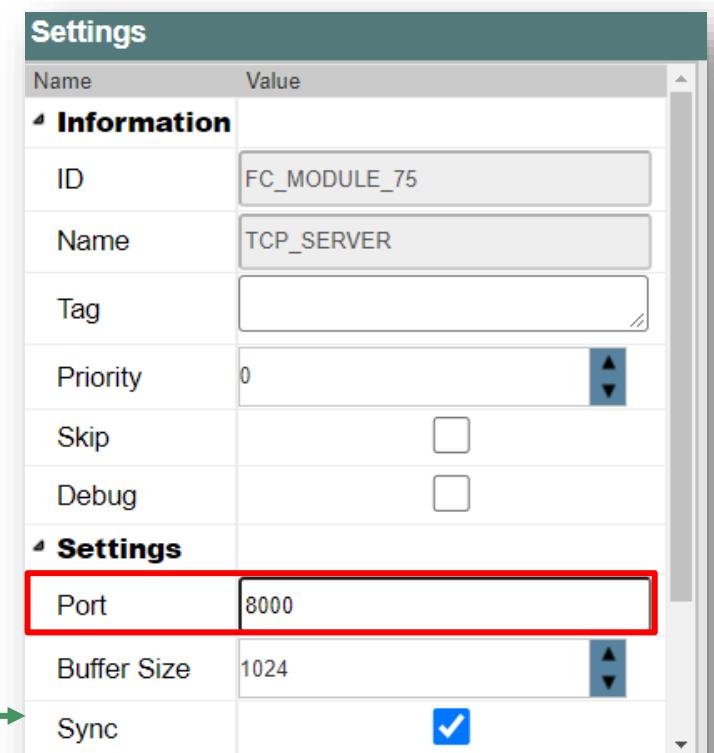
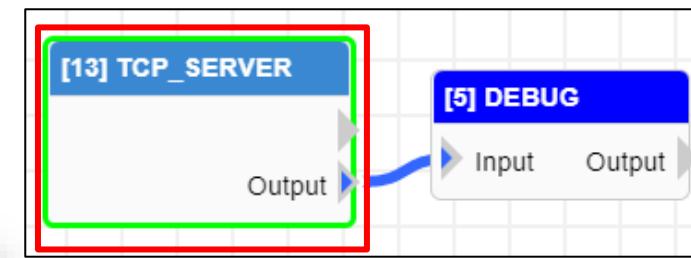
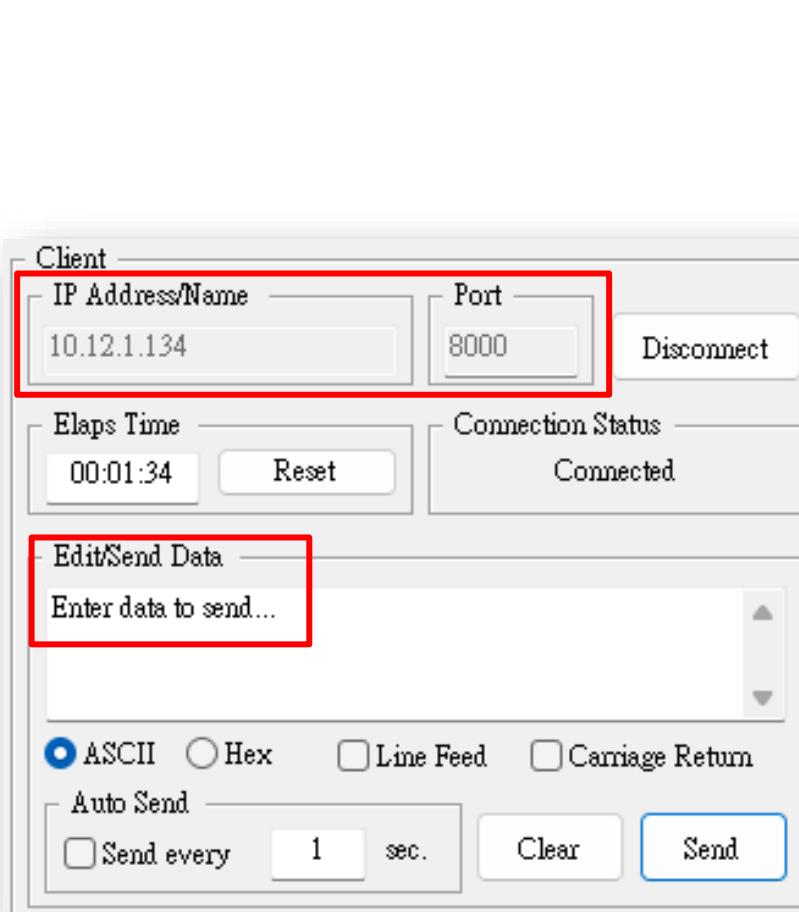
# VIC Flow 函式

## • TCP\_SERVER

可建立TCP Server並接收資料的運算子

若有設置回傳值，系統會將回傳值給傳送的TCP Client

	UI名稱	行為
輸出端	Output	輸出接收的資料



```
[293158] >>>> FC_MODULE_75 [ TCP_SERVER... outputs[2]-Output (string): Enter data to send... ] [293459] >>>> FC_DEBUG_5 <<<< inputs[1]-Input (string): Enter data to send... Enter data to send... outputs[1]-Output (string): Enter data to send...
```

- **TCP\_SERVER**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>通訊埠 (Port)</b>	設置TCP Server的通訊埠	Integer
<b>訊息大小 (Data Size)</b>	設定接收的訊息大小	Integer
<b>同步 (Sync)</b>	設定是否同步	

**Settings**

Name	Value
<b>Information</b>	
ID	FC_MODULE_75
Name	TCP_SERVER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Port	8000
Buffer Size	1024
Sync	<input checked="" type="checkbox"/>
<b>Fixed arr</b>	
Use Fixed arr	<input type="checkbox"/>
Fixed arr	CURRENT_IMAGE

# Outline

- 軟體設定
- IP 設定
- 登入
- 新增專案
- 取像
- 字型學習
- 目標值
- 頁面
- 辨識工具
- 執行專案
- 資料庫
- 重新命名
- 事件觸發
- 外部連線
- 控制
- 腳本
- VIC Flow 操作介面
- VIC Flow 表達式&運算子
- VIC Flow Modbus
- VIC Flow 觸發運算子
- VIC Flow 函式
- VIC Flow 控制函式

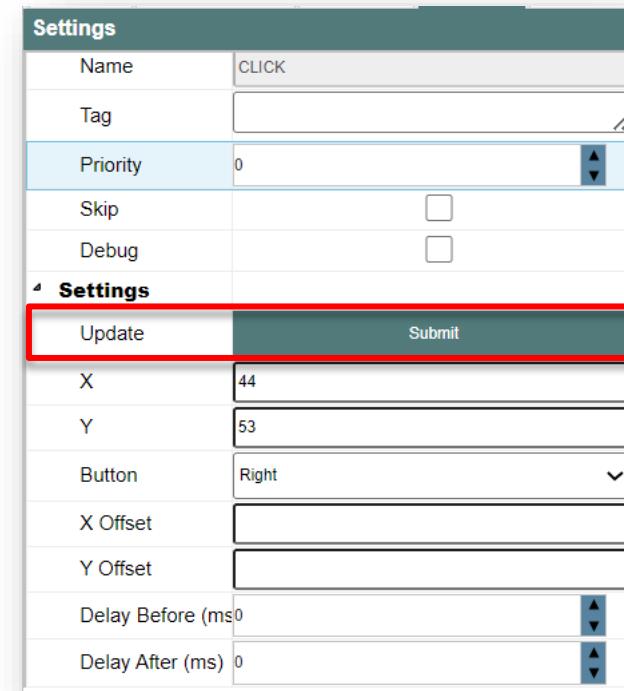
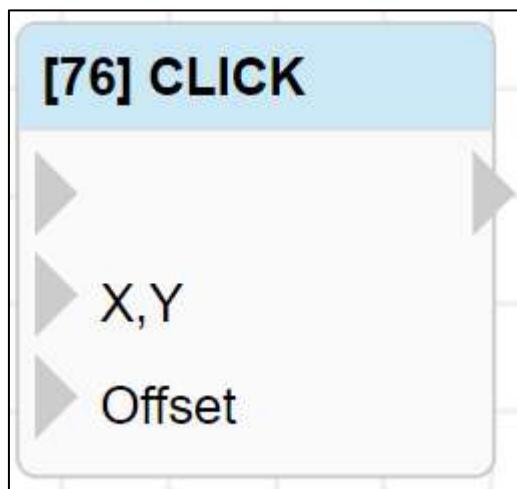
# VIC Flow 控制函式

## • CLICK

可執行滑鼠單次點擊的運算子

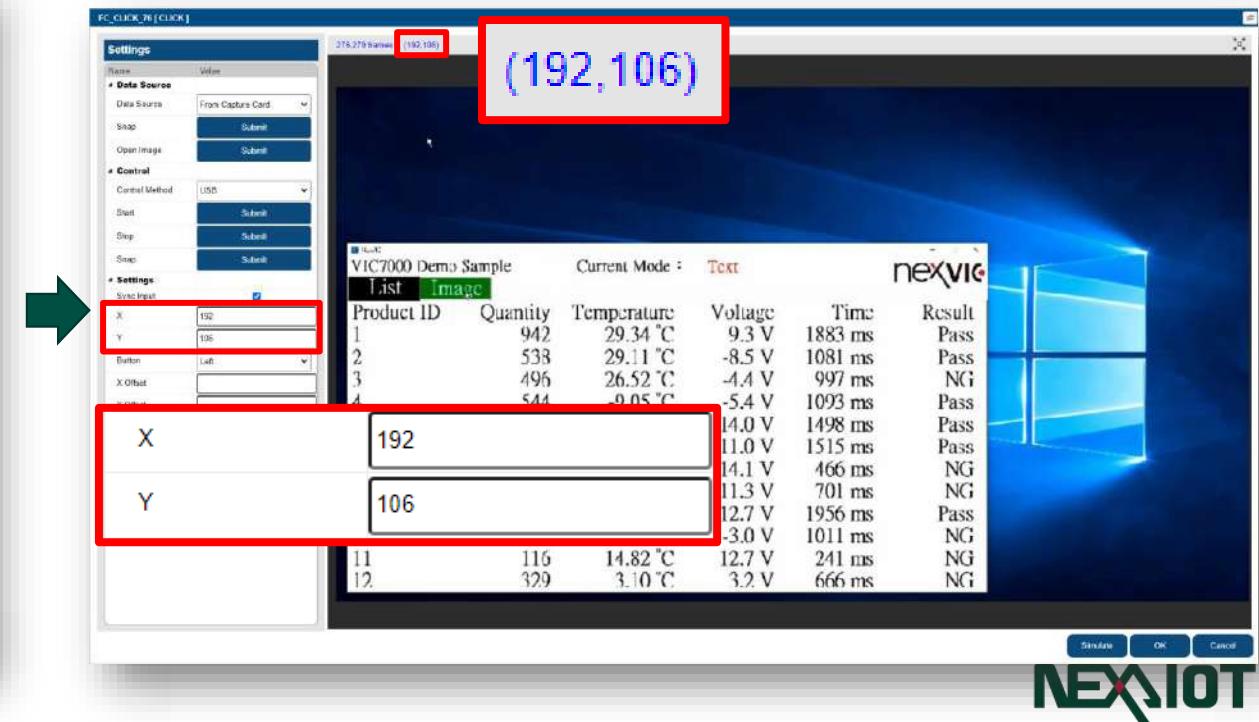
點擊CLICK運算子設置中的更新，可開啟更新視窗

在更新視窗內進行點擊，系統會記錄點擊位置，並在點擊確定後，將位置帶入CLICK運算子



	UI名稱	行為
輸入端	X,Y	輸入滑鼠點擊的X, Y座標
	Offset	輸入執行時欲偏移的量 輸入格式為x_offset, y_offset
輸出端		

Ex : 在座標(192,106)位置執行點擊



# VIC Flow 控制函式

## • CLICK

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>X</b>	設置滑鼠點擊的X座標	Python Expression
<b>Y</b>	設置滑鼠點擊的Y座標	Python Expression
<b>按鍵(Button)</b>	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
<b>X位移 (X Offset)</b>	設置執行的X座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置執行的Y座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>之前延遲(ms) (Delay Before(ms))</b>	設置執行此運算子之前的置延遲時間	Integer
<b>之後延遲(ms) (Delay After(ms))</b>	設置執行此運算子之後的延遲時間	Integer

**Settings**

Name	Value
<b>Information</b>	
ID	FC_CLICK_76
Name	CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Update	<b>Submit</b>
X	
Y	
Button	Left
X Offset	
Y Offset	
Delay Before (ms)	0
Delay After (ms)	0

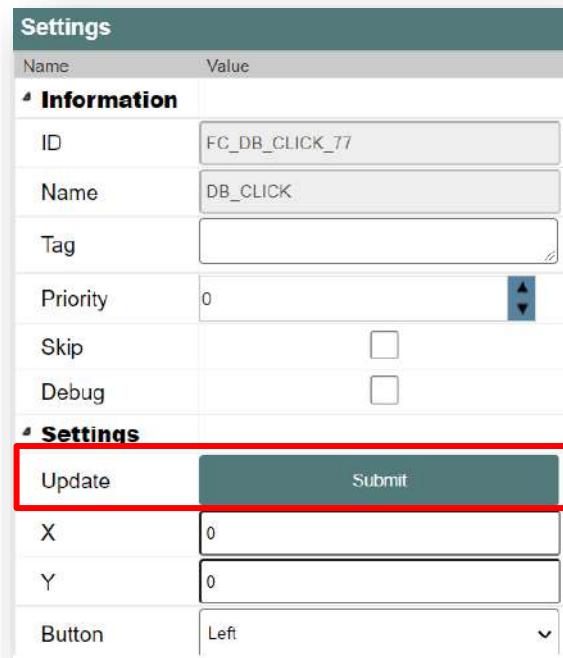
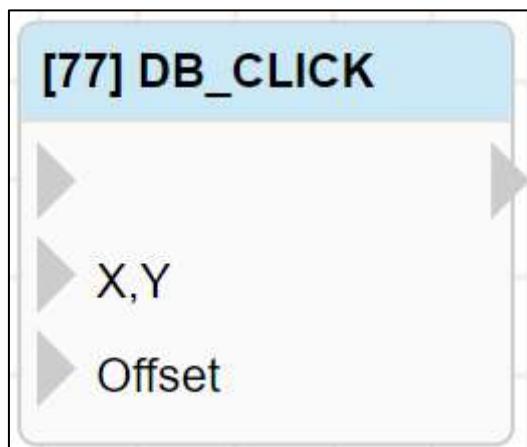
# VIC Flow 控制函式

## • DB\_CLICK

可執行滑鼠雙次點擊的運算子

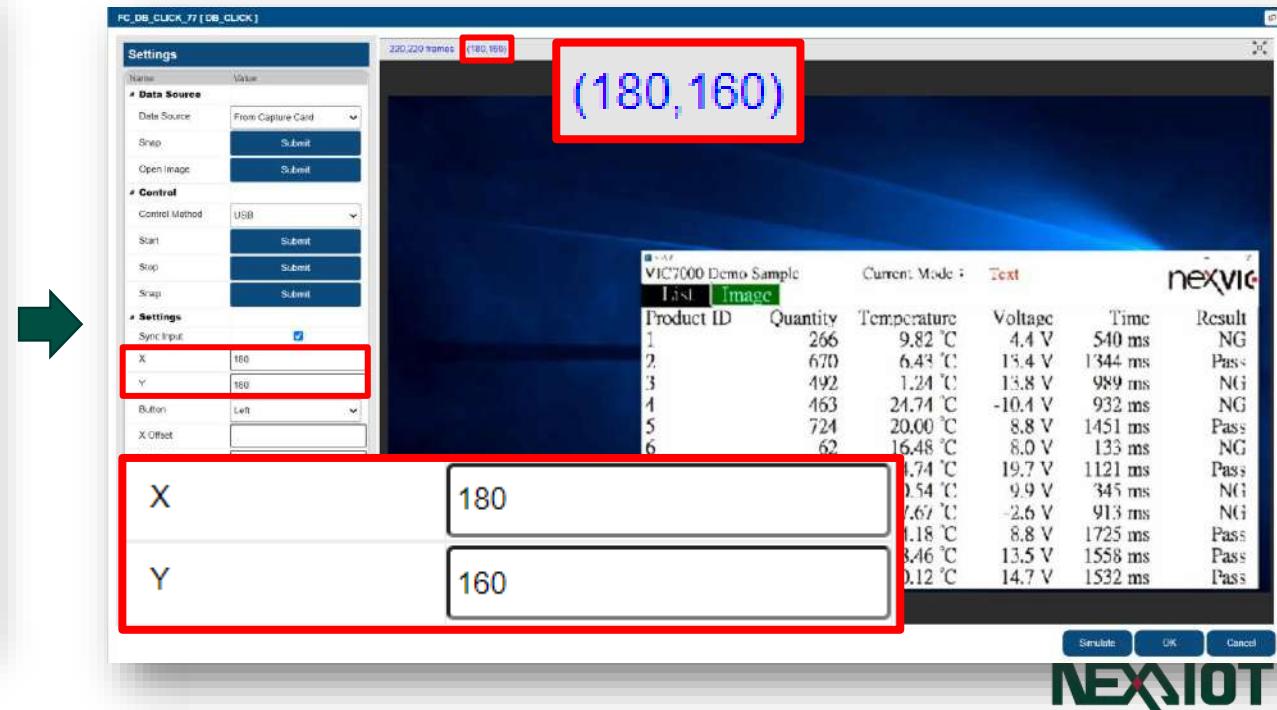
點擊DB\_CLICK運算子設置中的更新，可開啟更新視窗

在更新視窗內進行點擊，系統會記錄點擊位置，並在點擊確定後，將位置帶入DB\_CLICK運算子



	UI名稱	行為
輸入端	X,Y	輸入滑鼠點擊的X, Y座標
	Offset	輸入執行時欲偏移的量 輸入格式為x_offset, y_offset
輸出端		

Ex : 在座標(180, 160)位置執行點擊



# VIC Flow 控制函式

- DB\_CLICK

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>X</b>	設置滑鼠點擊的X座標	Python Expression
<b>Y</b>	設置滑鼠點擊的Y座標	Python Expression
<b>按鍵(Button)</b>	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
<b>X位移 (X Offset)</b>	設置執行的X座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置執行的Y座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>之前延遲(ms) (Delay Before(ms))</b>	設置執行此運算子之前的置延遲時間	Integer
<b>之後延遲(ms) (Delay After(ms))</b>	設置執行此運算子之後的延遲時間	Integer

Settings	
Name	Value
<b>Information</b>	
ID	FC_DB_CLICK_77
Name	DB_CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Update	<b>Submit</b>
X	0
Y	0
Button	Left
X Offset	
Y Offset	
Delay Before (ms)	0
Delay After (ms)	0

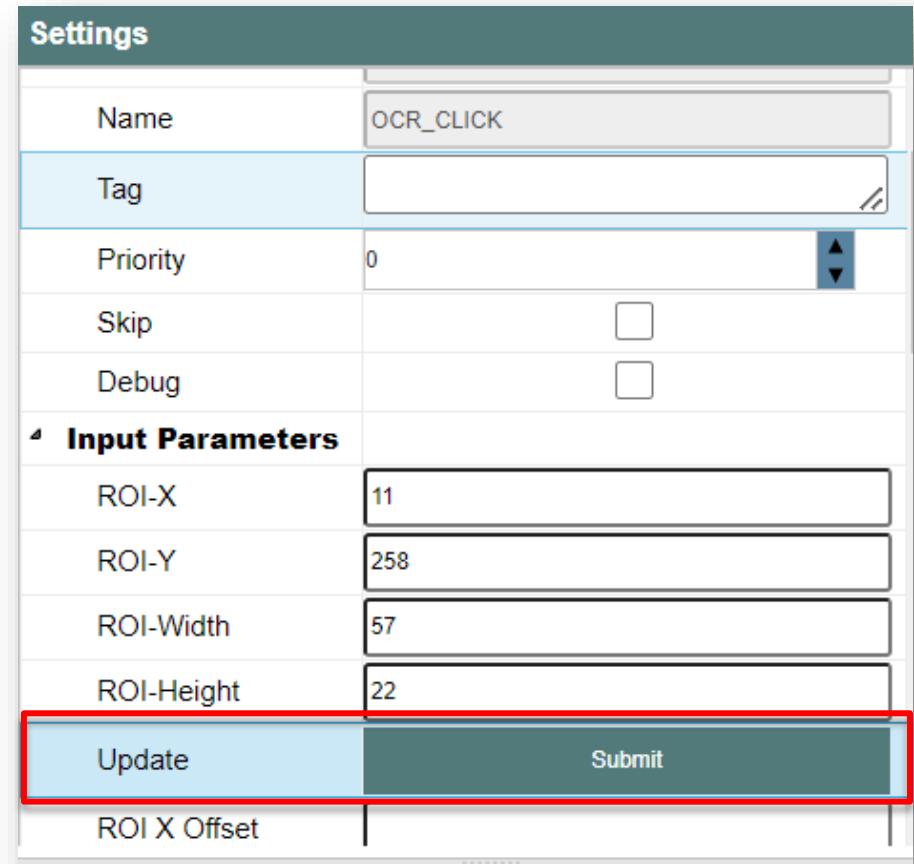
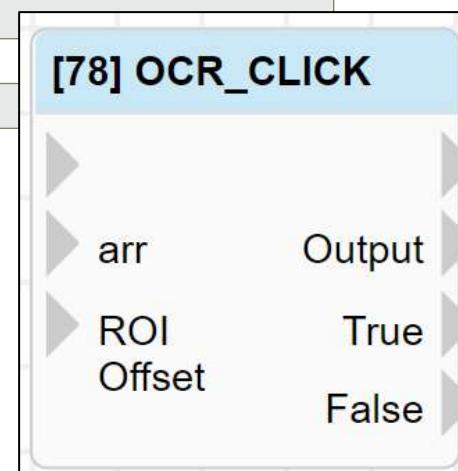
# VIC Flow 控制函式

## • OCR\_CLICK

可設置OCR辨識範圍，當辨識結果有包含設置的條件，滑鼠單次點擊符合條件的OCR位置中心點的運算子，True與False的輸出端無數值輸出

點擊OCR\_CLICK運算子設置中的更新，可開啟更新視窗

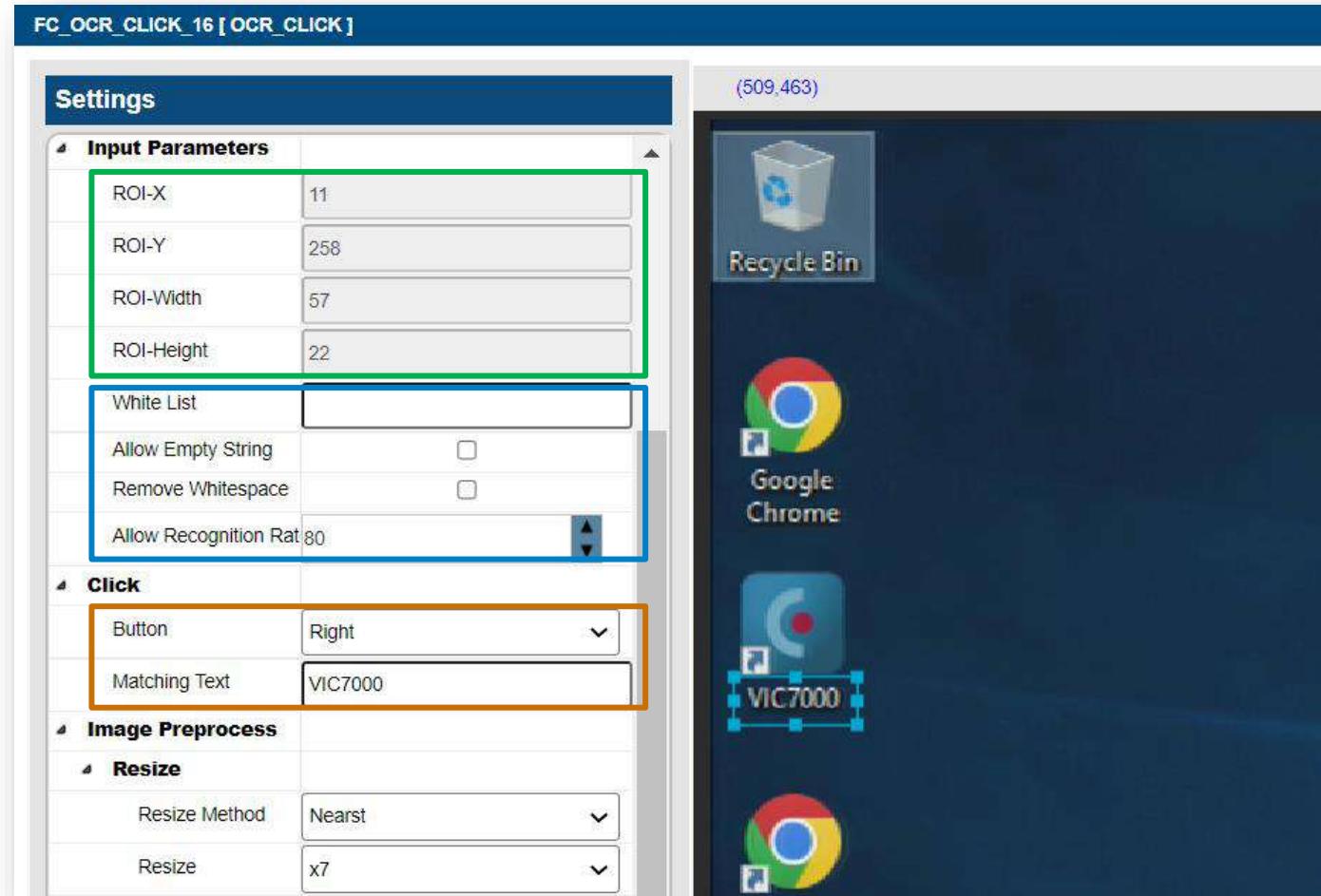
	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用當前擷取的圖像進行辨識
	ROI Offset	輸入辨識ROI偏移的位移量，格式為offset_x, offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	Output	輸出辨識結果
	True	若有符合的辨識結果，則由此輸出
	False	若無符合的辨識結果，則由此輸出



# VIC Flow 控制函式

## • OCR\_CLICK

在OCR\_CLICK的更新視窗中可直接進行OCR設置，包含設置ROI資訊以及OCR選項  
也可設置點擊按鍵和要比對的字串

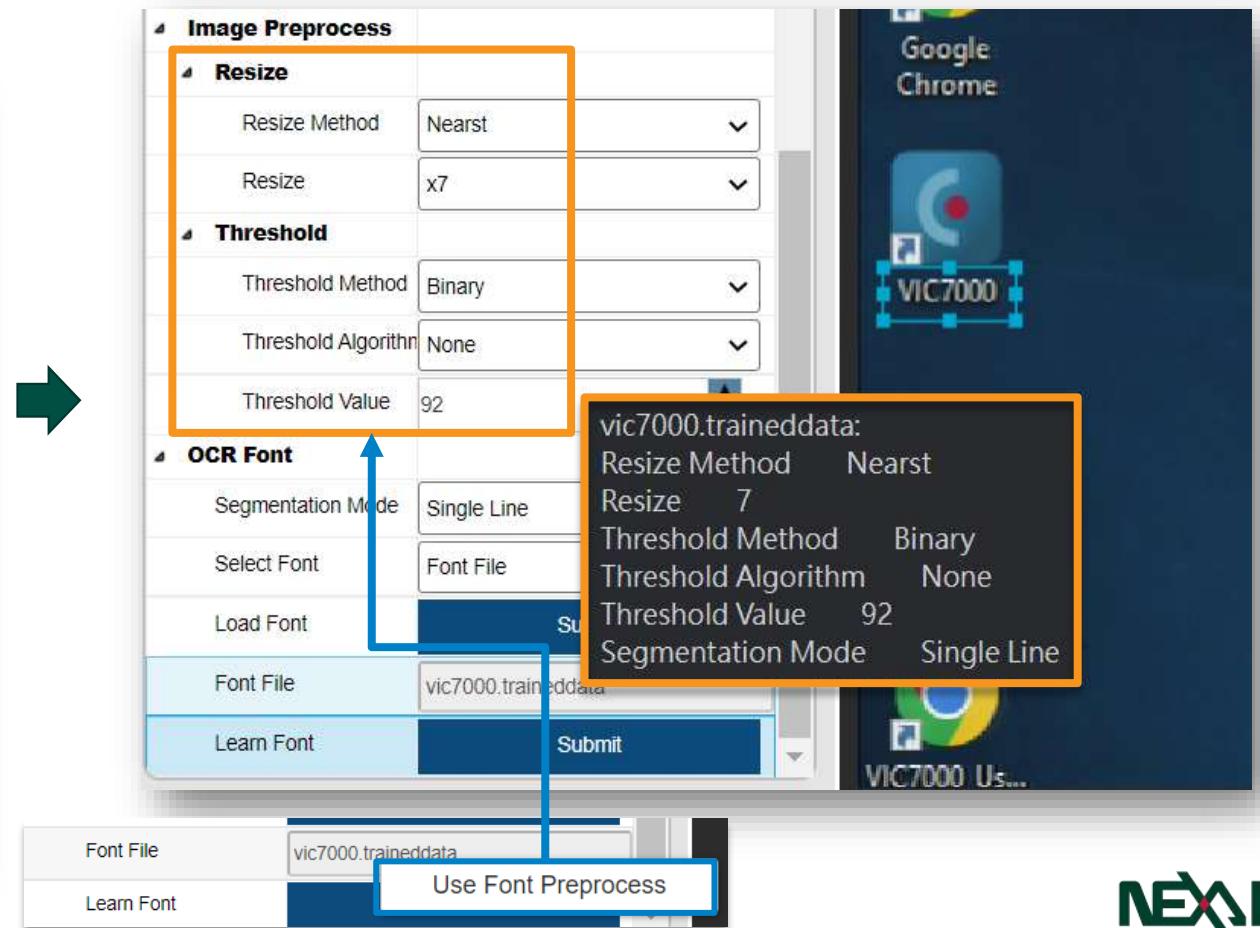
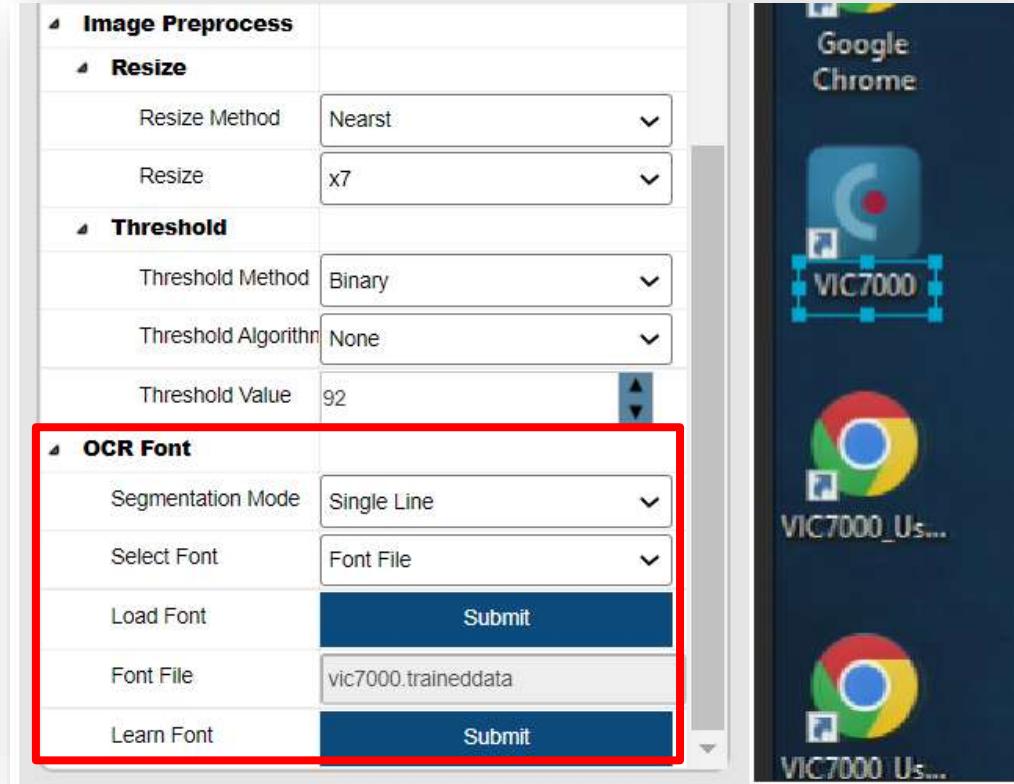


# VIC Flow 控制函式

## • OCR\_CLICK

在OCR\_CLICK的更新視窗中可直接進行OCR設置，包含載入**辨識字型檔**和**設置前處理參數**

在字型檔欄位上點擊右鍵，選擇**使用字型的前處理參數**，即可快速設置載入字型檔的前處理參數



# VIC Flow 控制函式

- OCR\_CLICK

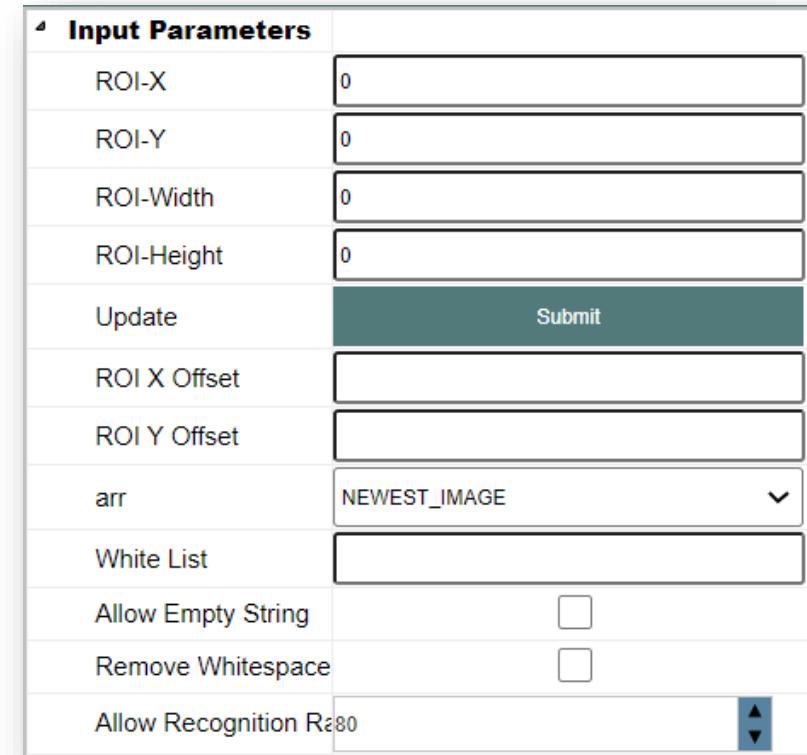
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

Settings	
Name	Value
Information	
ID	FC_OCR_CLICK_44
Name	OCR_CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

# VIC Flow 控制函式

## • OCR\_CLICK

名稱	意義	格式
<b>設置 (Input Parameters)</b>		
ROI-X	設置辨識範圍的X座標值	Python Expression
ROI-Y	設置辨識範圍的Y座標值	Python Expression
ROI-寬度 (ROI-Width)	設置辨識範圍的寬度	Python Expression
ROI-高度 (ROI-Height)	設置辨識範圍的高度	Python Expression
更新 (Update)	點擊後，將開啟進階設定視窗	
ROI X 位移 (ROI X Offset)	設置辨識範圍的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
ROI Y 位移 (ROI Y Offset)	設置辨識範圍的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
arr	設置欲辨識的圖像 預設為NEWEST_IMAGE	
白名單 (White List)	僅辨識該欄位設置的值 若無設置代表全辨識	String
移除空白字元 (Remove Whitespace)	啟用後，當辨識結果有空白字元，將會自動 移除空白字元	
允許辨識率(%) (Allow Recognition Rate(%))	設置允許最低的辨識率 若低於此值，將會視為辨識NG	

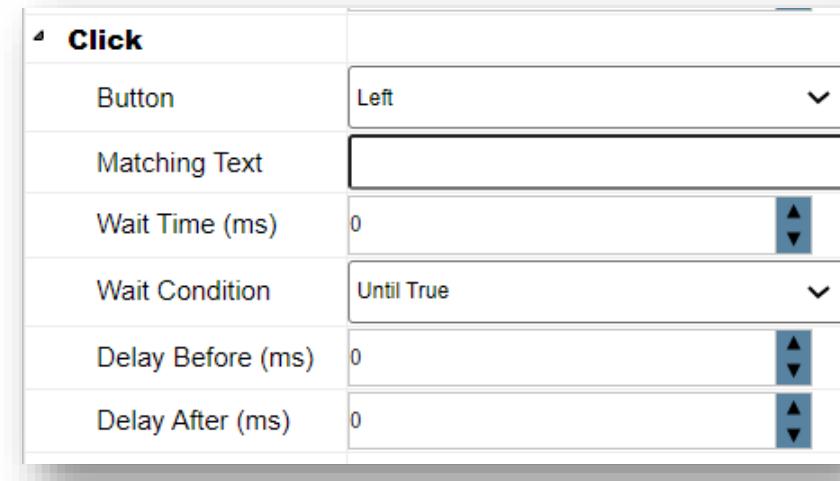


The screenshot shows a configuration interface for the 'Input Parameters' of the OCR\_CLICK function. The interface includes fields for ROI-X, ROI-Y, ROI-Width, ROI-Height, and an 'Update' button. It also includes fields for 'ROI X Offset' and 'ROI Y Offset', both currently empty. The 'arr' field is set to 'NEWEST\_IMAGE'. The 'White List' field is empty. There are checkboxes for 'Allow Empty String' (unchecked), 'Remove Whitespace' (unchecked), and 'Allow Recognition Ra80' (unchecked). The 'Submit' button is highlighted in green.

# VIC Flow 控制函式

## • OCR\_CLICK

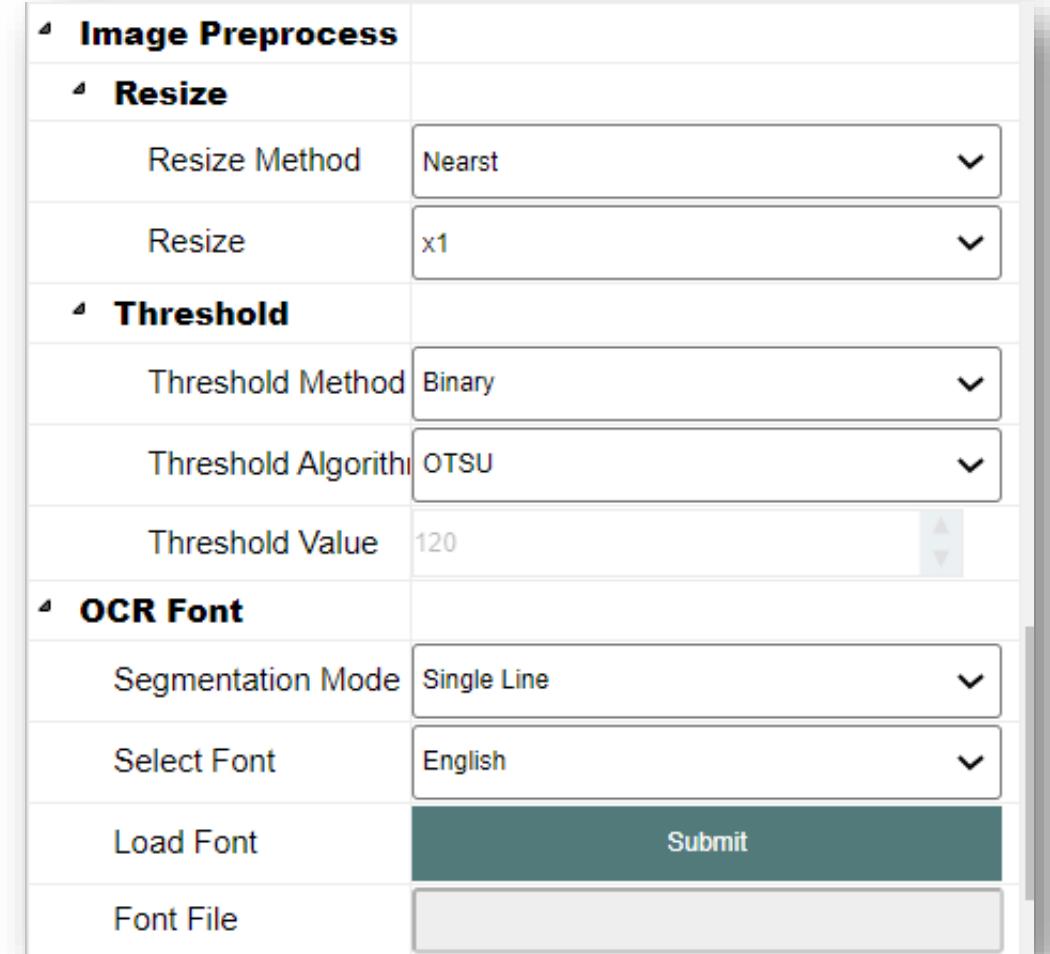
名稱	意義	格式
<b>點擊 (Click)</b>		
按鍵 (Button)	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
搜尋文字 (Matching Text)	欲搜尋的文字	Inline Python Expression String
等待時間(ms) (Wait Time (ms))	設置每次辨識的等待時間	Integer
等待條件 (Wait Condition)	設置等待的條件： 包含直到True(Until True) · 代表直到比對成功 以及直到False(Until False) · 代表直到比對失敗	
之前延遲(ms) (Delay Before(ms))	設置執行此運算子之前的置延遲時間	Integer
之後延遲(ms) (Delay After(ms))	設置執行此運算子之後的延遲時間	Integer



# VIC Flow 控制函式

## • OCR\_CLICK

名稱	意義	格式
<b>影像前處理 (Image Preprocess)</b>		
<b>影像前處理-調整大小 (Image Preprocess - Resize)</b>		
調整大小方式 (Resize Method)	設置調整大小的方式	
調整大小 (Resize)	設置放大倍率	
<b>影像前處理-二值化 (Image Preprocess - Threshold)</b>		
Threshold Method (二值化方法)	設置二值化方法	
二值化演算法 (Threshold Algorithm)	設置二值化演算法	
閾值 (Threshold Value)	設置二值化閾值 僅在二值化演算法設為無時才可設置	Integer
<b>辨識字型(OCR Font)</b>		
分割方法 (Segmentation Mo)	設置OCR的切割方法	
選擇字型 (Select Font)	設置使用的字型	
載入字型 (Load Font)	點擊後，即可選擇辨識字型檔	
字型檔 (Font File)	當前使用的字型檔名稱	



The screenshot shows the VIC Flow software interface with the following configuration settings:

- Image Preprocess**
  - Resize**
    - Resize Method: Nearst
    - Resize: x1
  - Threshold**
    - Threshold Method: Binary
    - Threshold Algorithm: OTSU
    - Threshold Value: 120
- OCR Font**
  - Segmentation Mode: Single Line
  - Select Font: English
  - Load Font: Submit
  - Font File: (empty)

# VIC Flow 控制函式

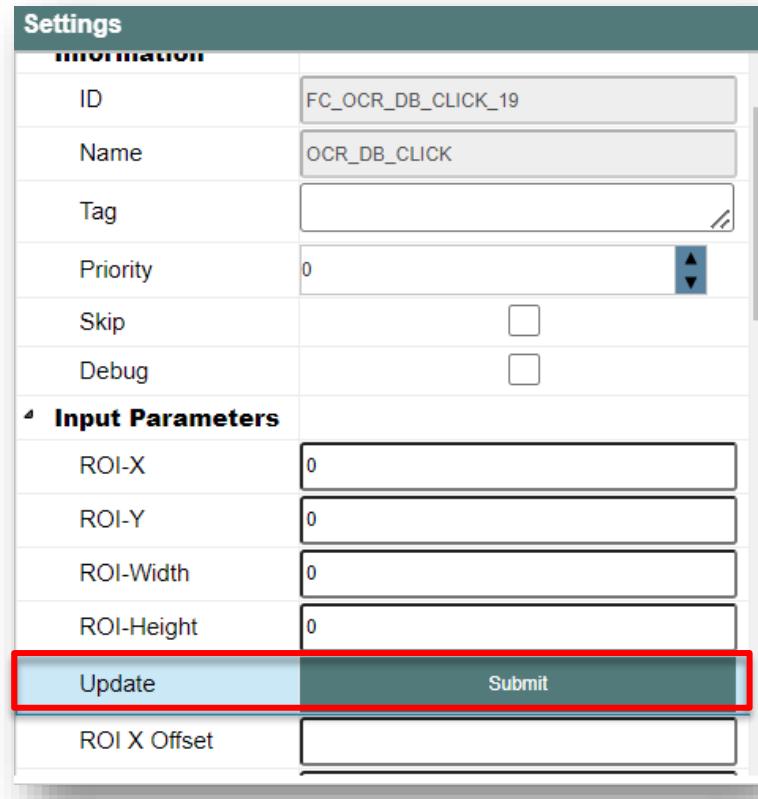
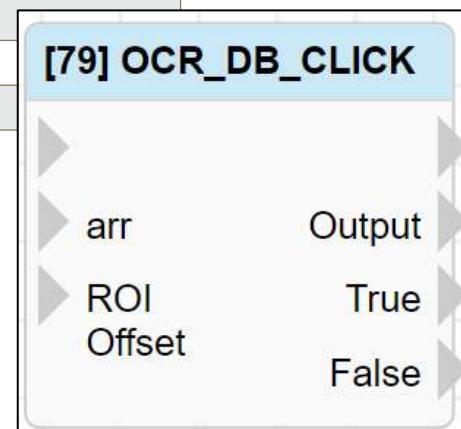
## • OCR\_DB\_CLICK

可設置OCR辨識範圍，當辨識結果有包含設置的條件，進行滑鼠雙次點擊符合條件的OCR位置中心點的運算子，True與False的輸出端無數值輸出

點擊OCR\_DB\_CLICK運算子設置中的更新，可開啟更新視窗

更新視窗的使用方法和OCR\_CLICK的更新視窗相同

	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用當前擷取的圖像進行辨識
	ROI Offset	輸入辨識ROI偏移的位移量，格式為offset_x, offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	Output	輸出辨識結果
	True	若有符合的辨識結果，則由此輸出
	False	若無符合的辨識結果，則由此輸出



# VIC Flow 控制函式

- OCR\_DB\_CLICK

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

**Settings**

Name	Value
<b>Information</b>	
ID	FC_OCR_DB_CLICK_7
Name	OCR_DB_CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

# VIC Flow 控制函式

## • OCR\_DB\_CLICK

名稱	意義	格式
<b>設置 (Input Parameters)</b>		
ROI-X	設置辨識範圍的X座標值	Python Expression
ROI-Y	設置辨識範圍的Y座標值	Python Expression
ROI-寬度 (ROI-Width)	設置辨識範圍的寬度	Python Expression
ROI-高度 (ROI-Height)	設置辨識範圍的高度	Python Expression
更新 (Update)	點擊後，將開啟進階設定視窗	
ROI X位移 (ROI X Offset)	設置辨識範圍的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
ROI Y位移 (ROI Y Offset)	設置辨識範圍的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
arr	設置欲辨識的圖像 預設為NEWEST_IMAGE	
白名單 (White List)	僅辨識該欄位設置的值 若無設置代表全辨識	String
移除空白字元 (Remove Whitespace)	啟用後，當辨識結果有空白字元，將會自動移除空白字元	
允許辨識率(%) (Allow Recognition Rate(%))	設置允許最低的辨識率，若低於此值，將會視為辨識NG	

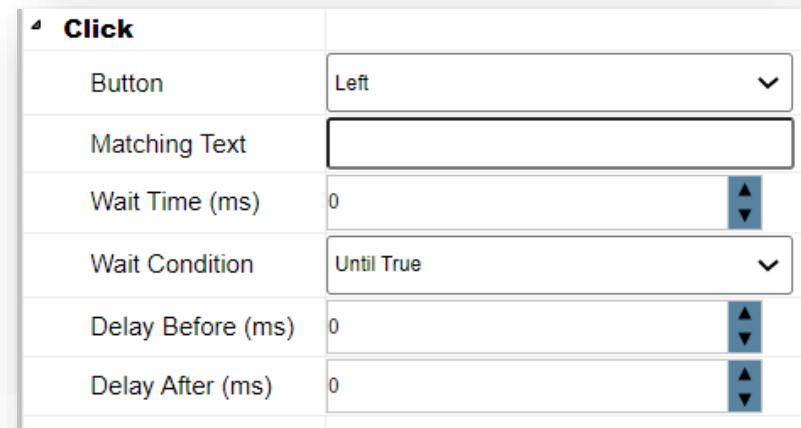
Input Parameters

ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	Submit
ROI X Offset	
ROI Y Offset	
arr	NEWEST_IMAGE
White List	
Allow Empty String	<input type="checkbox"/>
Remove Whitespace	<input type="checkbox"/>
Allow Recognition Rate	80

# VIC Flow 控制函式

## • OCR\_DB\_CLICK

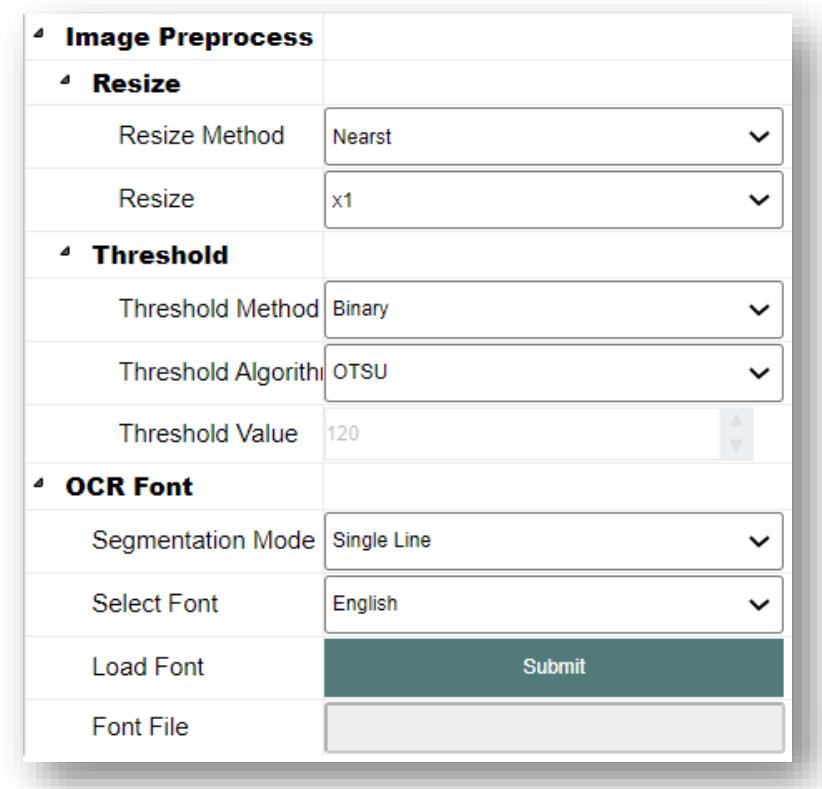
名稱	意義	格式
<b>點擊 (Click)</b>		
按鍵 (Button)	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
搜尋文字 (Matching Text)	欲搜尋的文字	Inline Python Expression String
等待時間(ms) (Wait Time (ms))	設置每次辨識的等待時間	Integer
等待條件 (Wait Condition)	設置等待的條件： 包含直到True(Until True) · 代表直到比對成功 以及直到False(Until False) · 代表直到比對失敗	
之前延遲(ms) (Delay Before(ms))	設置執行此運算子之前的置延遲時間	Integer
之後延遲(ms) (Delay After(ms))	設置執行此運算子之後的延遲時間	Integer



# VIC Flow 控制函式

## • OCR\_DB\_CLICK

名稱	意義	格式
<b>影像前處理 (Image Preprocess)</b>		
<b>影像前處理-調整大小 (Image Preprocess - Resize)</b>		
調整大小方式 (Resize Method)	設置調整大小的方式	
調整大小 (Resize)	設置放大倍率	
<b>影像前處理-二值化 (Image Preprocess - Threshold)</b>		
二值化方法 (Threshold Method)	設置二值化方法	
二值化演算法 (Threshold Algorithm)	設置二值化演算法	
閥值 (Threshold Value)	設置二值化閥值 僅在二值化演算法設為無時才可設置	Integer
<b>辨識字型 (OCR Font)</b>		
分割方法 (Segmentation Mode)	設置OCR的切割方法	
選擇字型 (Select Font)	設置使用的字型	
載入字型 (Load Font)	點擊後，即可選擇辨識字型檔	
字型檔 (Font File)	當前使用的字型檔名稱	



The screenshot shows the VIC Flow software interface for configuring the OCR\_DB\_CLICK function. The configuration is organized into sections:

- Image Preprocess** section:
  - Resize** sub-section: Contains dropdown menus for "Resize Method" (set to "Nearst") and "Resize" (set to "x1").
  - Threshold** sub-section: Contains dropdown menus for "Threshold Method" (set to "Binary") and "Threshold Algorithm" (set to "OTSU"). It also includes a "Threshold Value" input field set to "120".
- OCR Font** section:
  - Segmentation Mode** dropdown menu (set to "Single Line").
  - Select Font** dropdown menu (set to "English").
  - Load Font** button (highlighted in green).
  - Font File** input field.

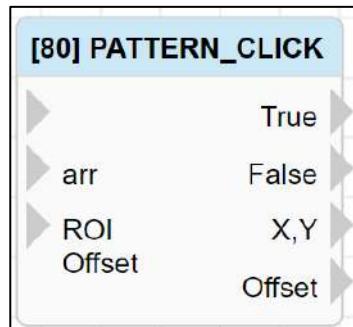
# VIC Flow 控制函式

## • PATTERN\_CLICK

可設置辨識範圍，進行樣式比對並進行滑鼠單次點擊比對到的樣式位置中心點的運算子  
True與False的輸出端無數值輸出

點擊PATTERN\_CLICK運算子設置中的更新，可開啟更新視窗

	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用當前擷取的圖像進行辨識
	ROI Offset	輸入辨識 ROI 偏移的位移量，格式為 offset_x, offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	True	若比對成功，則由此輸出
	False	若比對失敗，則由此輸出
	X,Y	輸出比對到的樣式中心座標位置
	Offset	輸出比對到的樣式，與設置比對樣式時的位置偏移量



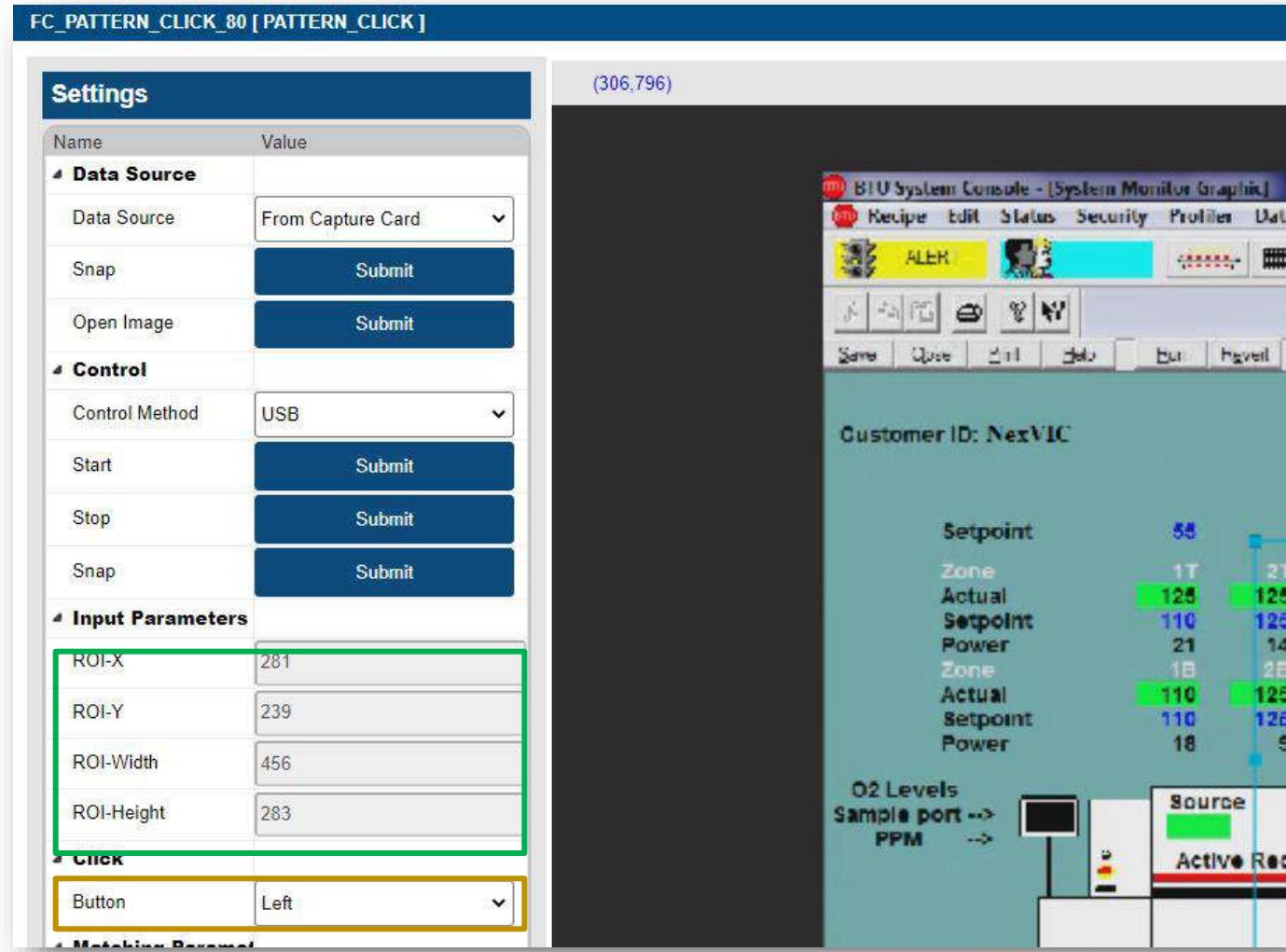
Settings

Name	Value
ID	FC_PATTERN_CLICK_16
Name	PATTERN_CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Input Parameters	
ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	Submit

# VIC Flow 控制函式

## • PATTERN\_CLICK

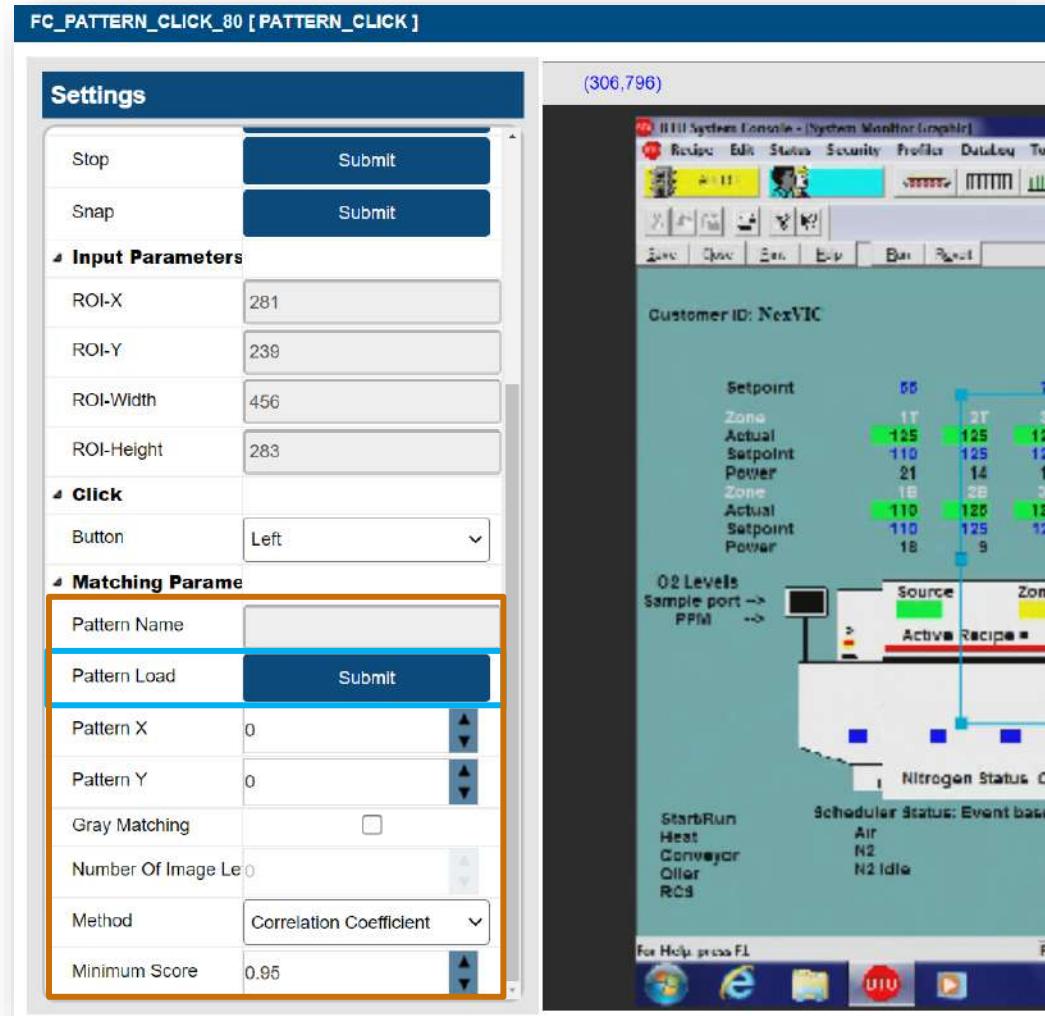
在PATTERN\_CLICK的更新視窗中可直接進行PATTERN設置，包含設置ROI資訊也可設置點擊按鍵



# VIC Flow 控制函式

## • PATTERN\_CLICK

在PATTERN\_CLICK的更新視窗中可直接進行PATTERN設置，包含比對樣式和PATTERN比對參數

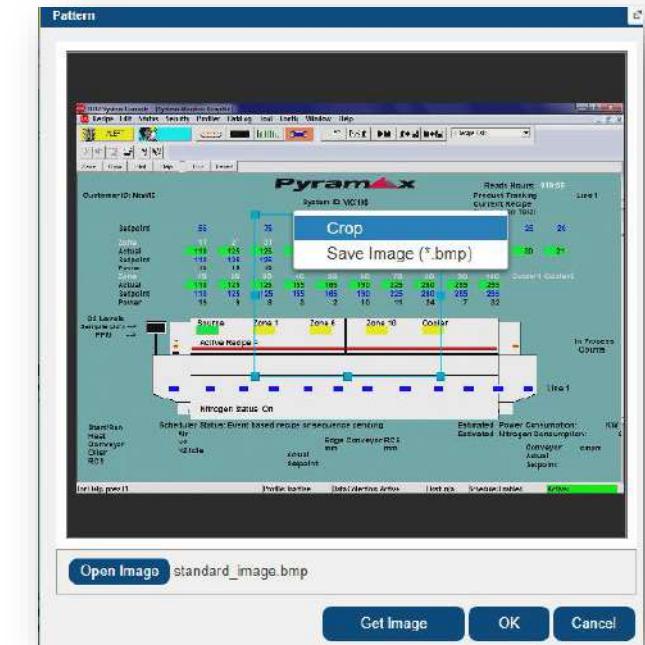
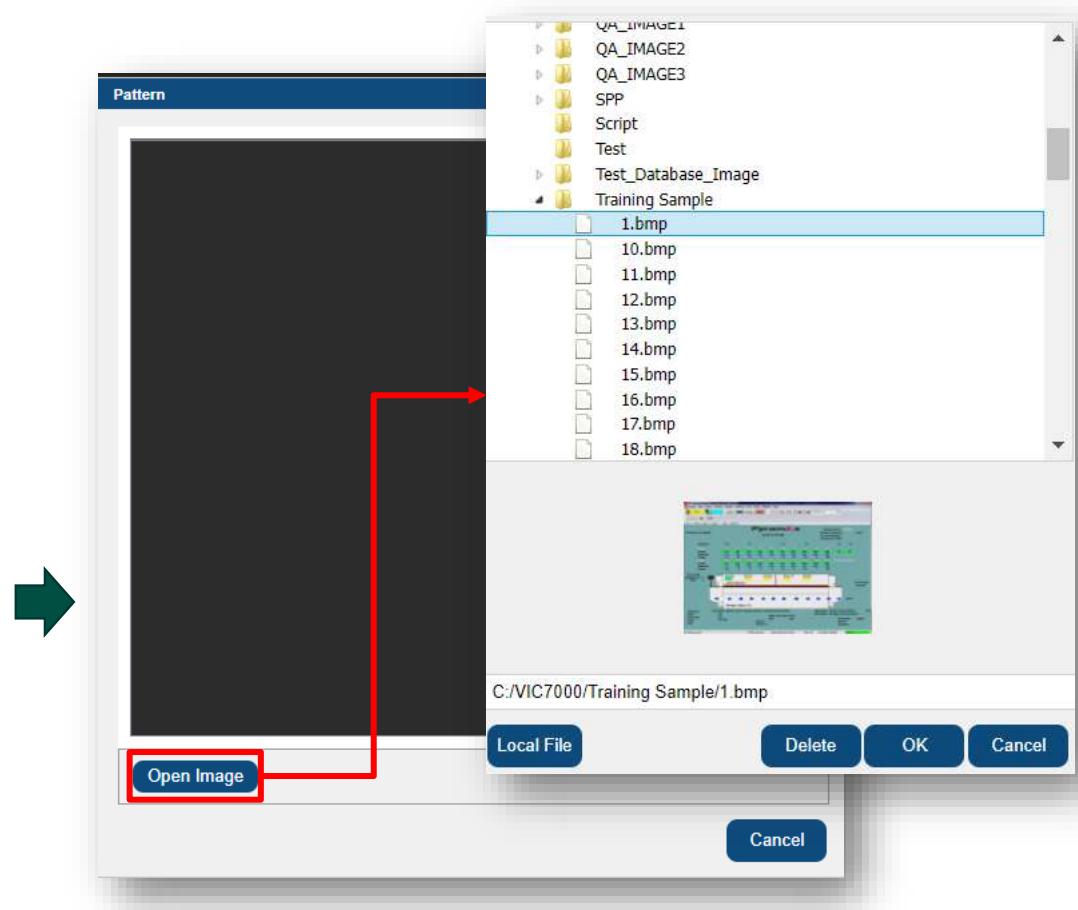
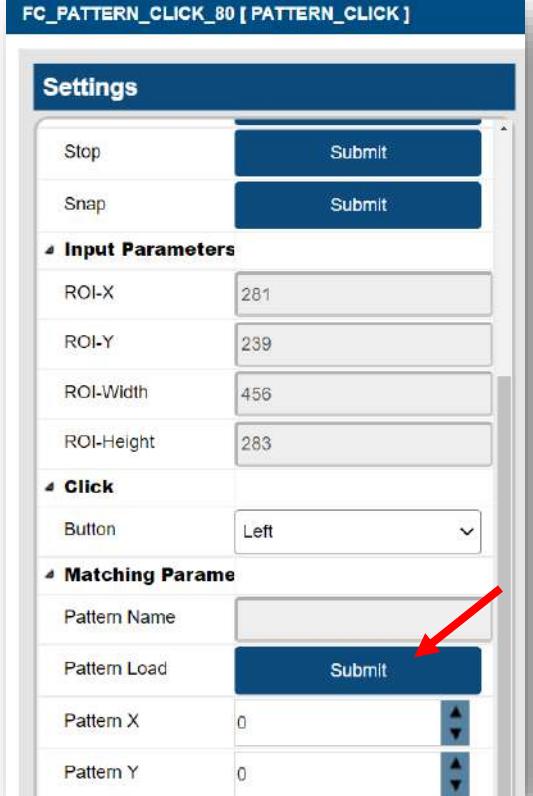


# VIC Flow 控制函式

## • PATTERN\_CLICK

在更新視窗中點擊樣式載入，即會出現樣式載入視窗

可在此載入圖檔和設置比對樣式



# VIC Flow 控制函式

## • PATTERN\_CLICK

名稱	意義	格式
<b>訊息(Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Input Parameters)</b>		
<b>ROI-X</b>	設置辨識範圍的X座標值	Python Expression
<b>ROI-Y</b>	設置辨識範圍的Y座標值	Python Expression
<b>ROI-寬度 (ROI-Width)</b>	設置辨識範圍的寬度	Python Expression
<b>ROI-高度 (ROI-Height)</b>	設置辨識範圍的高度	Python Expression
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>ROI X位移 (ROI X Offset)</b>	設置辨識範圍的X座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>ROI Y位移 (ROI Y Offset)</b>	設置辨識範圍的Y座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>arr</b>	設置辨識使用的圖像 預設為NEWEST_IMAGE	

**Settings**

Name	Value
<b>Information</b>	
ID	FC_PATTERN_CLICK_80
Name	PATTERN_CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Input Parameters</b>	
ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	<b>Submit</b>
ROI X Offset	
ROI Y Offset	
arr	NEWEST_IMAGE

# VIC Flow 控制函式

## • PATTERN\_CLICK

名稱	意義	格式
<b>點擊 (Click)</b>		
按鍵 (Button)	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
<b>比對參數 (Matching Parameters)</b>		
樣式影像名稱 (Pattern Name)	顯示比對樣式影像的檔案名稱	
樣式載入 (Pattern Load)	點擊後，即會顯示樣式載入視窗	
樣式X (Pattern X)	設置比對樣式的X值	Integer
樣式Y (Pattern Y)	設置比對樣式的Y值	Integer
灰階比對 (Gray Matching)	啟用後，會將圖像轉為灰階化進行比對，可加快樣式比對的時間	
最小分數 (Minimum Score)	設置樣式比對成功的最小分數 預設為0.95，有效值範圍為0至1 1代表完全吻合	Float
等待時間(ms) (Wait Time (ms))	設置每次比對的等待時間	Integer
等待條件 (Wait Condition)	設置等待的條件：包含直到True(Until True)，代表直到比對成功；以及直到False(Until False)，代表直到比對失敗	
之前延遲(ms) (Delay Before (ms))	設置執行動作前的延遲時間	Integer
之後延遲(ms) (Delay After (ms))	設置執行動作後的延遲時間	Integer

Settings

**Click**

Button: Left

**Matching Parameters**

Pattern Name:

Pattern Load:

Pattern X: 0

Pattern Y: 0

Gray Matching:

Number Of Image Load: 0

Method: Correlation Coefficient

Minimum Score: 0.95

Wait Time (ms): 0

Wait Condition: Until True

Delay Before (ms): 0

Delay After (ms): 0

# VIC Flow 控制函式

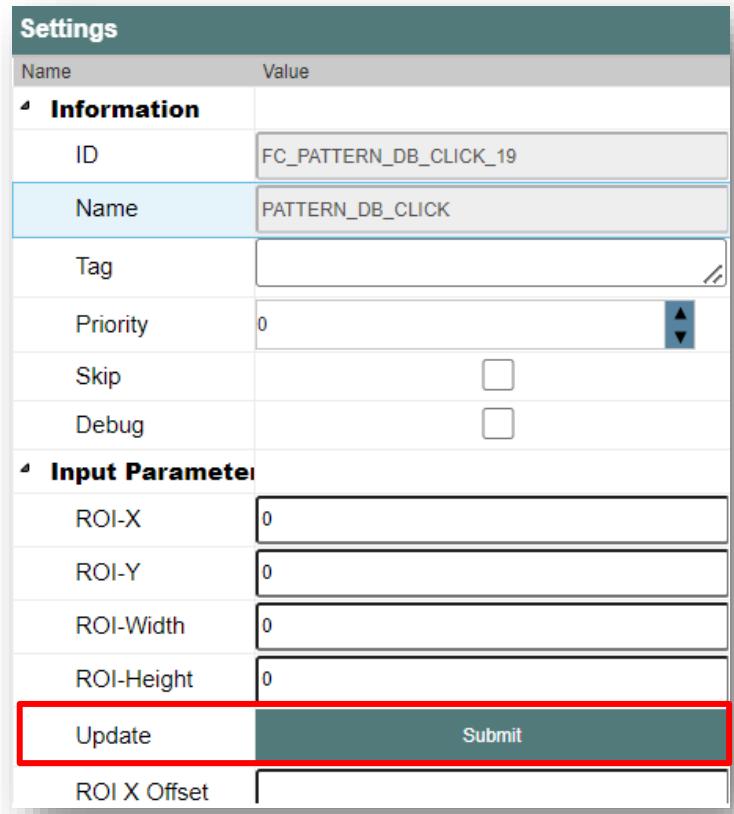
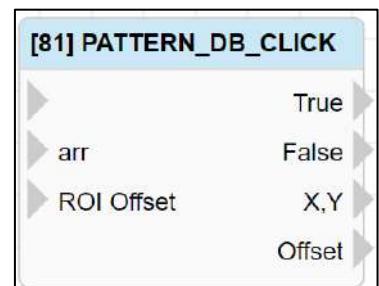
## • PATTERN\_DB\_CLICK

可設置辨識範圍進行樣式比對並進行滑鼠雙次點擊比對到的樣式位置中心點的運算子  
True與False的輸出端無數值輸出

點擊PATTERN\_DB\_CLICK運算子設置中的更新，可開啟更新視窗

更新視窗的使用方法和PATTERN\_CLICK的更新視窗相同

	UI名稱	行為
輸入端	arr	輸入欲辨識的圖像 若未輸入，則使用當前擷取的圖像進行辨識
	ROI Offset	輸入辨識ROI偏移的位移量，格式為offset_x, offset_y 假設X偏移10，Y偏移20，則輸入的格式為10, 20
輸出端	True	若比對成功，則由此輸出
	False	若比對失敗，則由此輸出
	X,Y	輸出比對到的樣式中心座標位置
	Offset	輸出比對到的樣式，與設置比對樣式時的位置偏移量



# VIC Flow 控制函式

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過Skip ()</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Input Parameters)</b>		
<b>ROI-X</b>	設置辨識範圍的X座標值	Python Expression
<b>ROI-Y</b>	設置辨識範圍的Y座標值	Python Expression
<b>ROI-寬度ROI-Width ()</b>	設置辨識範圍的寬度	Python Expression
<b>ROI-高度ROI-Height ()</b>	設置辨識範圍的高度	Python Expression
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>ROI X位移 (ROI X Offset)</b>	設置辨識範圍的X座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>ROI Y位移 (ROI Y Offset)</b>	設置辨識範圍的Y座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>arr</b>	設置辨識使用的圖像 預設為NEWEST_IMAGE	

Settings	
Name	Value
<b>Information</b>	
ID	FC_PATTERN_DB_CLICK_81
Name	PATTERN_DB_CLICK
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Input Parameters</b>	
ROI-X	0
ROI-Y	0
ROI-Width	0
ROI-Height	0
Update	<b>Submit</b>
ROI X Offset	
ROI Y Offset	
arr	NEWEST_IMAGE

# VIC Flow 控制函式

- **PATTERN\_DB\_CLICK**

名稱	意義	格式
<b>點擊 (Click)</b>		
按鍵 (Button)	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
<b>比對參數 (Matching Parameters)</b>		
樣式影像名稱 (Pattern Name)	顯示比對樣式影像的檔案名稱	
樣式載入 (Pattern Load)	點擊後，即會顯示樣式載入視窗	
樣式X (Pattern X)	設置比對樣式的X值	Integer
樣式Y (Pattern Y)	設置比對樣式的Y值	Integer
灰階比對 (Gray Matching)	啟用後，會將圖像轉為灰階化進行比對，可加快樣式比對的時間	
最小分數 (Minimum Score)	設置樣式比對成功的最小分數 預設為0.95，有效值範圍為0至1 1代表完全吻合	Float
等待時間(ms) (Wait Time (ms))	設置每次比對的等待時間	Integer
等待條件 (Wait Condition)	設置等待的條件：包含直到True(Until True)，代表直到比對成功；以及直到False(Until False)，代表直到比對失敗	
Delay Before (ms) (之前延遲(ms))	設置執行動作前的延遲時間	Integer
Delay After (ms) (之後延遲(ms))	設置執行動作後的延遲時間	Integer

The screenshot shows the 'Click' configuration dialog in the nexVIC software. The dialog is divided into sections: 'Click' (selected), 'Matching Parameters' (disabled), and other parameters like 'Pattern Name', 'Pattern Load' (set to 'Submit'), 'Pattern X', 'Pattern Y', 'Gray Matching', 'Number Of Image Lev', 'Minimum Score' (set to 0.95), 'Wait Time (ms)', 'Wait Condition' (set to 'Until True'), 'Delay Before (ms)', and 'Delay After (ms)'. The 'Pattern Load' field is highlighted in green, indicating it is the active or selected parameter.

# VIC Flow 控制函式

## • MOUSE\_MOVE

可移動滑鼠游標到指定位置的運算子

點擊MOUSE\_MOVE運算子設置中的更新，可開啟更新視窗

在更新視窗內進行點擊，系統會記錄點擊位置，並在點擊確定後，將位置帶入CLICK運算子

	UI名稱	行為
輸入端	X,Y	輸入滑鼠游標移動目標的X, Y座標
	Offset	輸入執行時欲偏移的量 輸入格式為x_offset, y_offset
輸出端		

[82] MOUSE\_MOVE

X,Y

Offset

Settings	
Name	Value
Information	
ID	FC_MOUSE_MOVE_82
Name	MOUSE_MOVE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Settings	
Update	<b>Submit</b>
X	
Y	
X Offset	



# VIC Flow 控制函式

## • MOUSE\_MOVE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>X</b>	設置滑鼠游標移動位置的X座標	Python Expression
<b>Y</b>	設置滑鼠游標移動位置的Y座標	Python Expression
<b>X位移 (X Offset)</b>	設置執行的X座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置執行的Y座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>之前延遲(ms) (Delay Before(ms))</b>	設置執行此運算子之前的置延遲時間	Integer
<b>之後延遲(ms) (Delay After(ms))</b>	設置執行此運算子之後的延遲時間	Integer

**Settings**

Name	Value
ID	FC_MOUSE_MOVE_82
Name	MOUSE_MOVE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

**Settings**

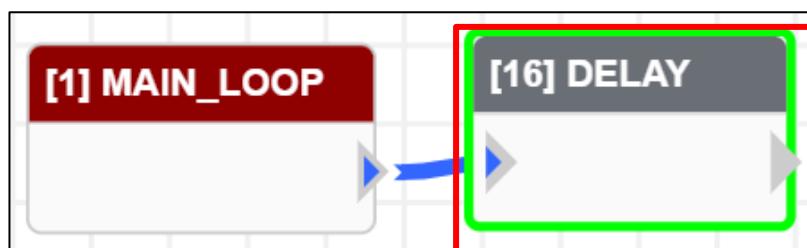
Update	<b>Submit</b>
X	
Y	
X Offset	
Y Offset	
Delay Before (ms)	0
Delay After (ms)	0

# VIC Flow 控制函式

- **DELAY**

可延遲一段時間再繼續執行後續流程的運算子

	UI名稱	行為
輸入端		
輸出端		



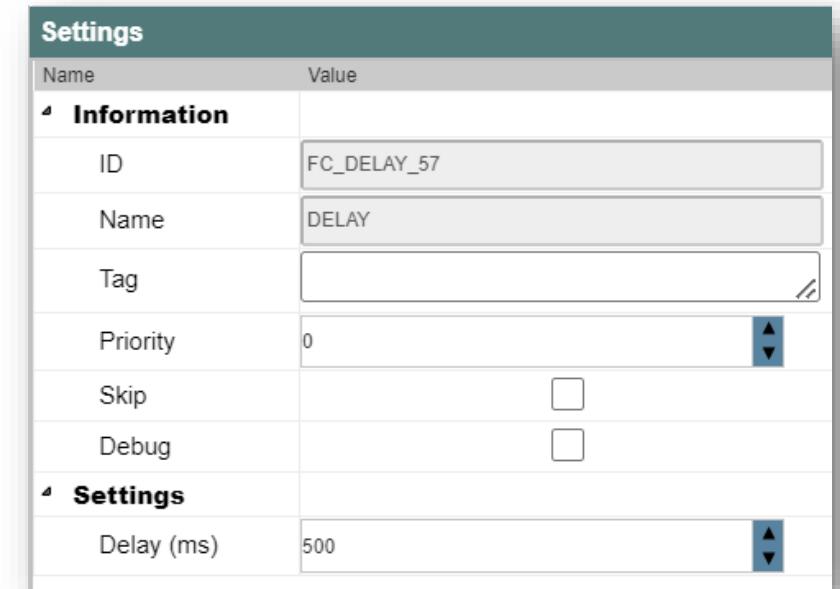
**Settings**

Name	Value
<b>Information</b>	
ID	FC_DELAY_16
Name	DELAY
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Delay (ms)	500

# VIC Flow 控制函式

- **DELAY**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>延遲時間 (ms) (Delay (ms))</b>	設置欲延遲的時間	Integer



Name	Value
Information	
ID	FC_DELAY_57
Name	DELAY
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Settings	
Delay (ms)	500

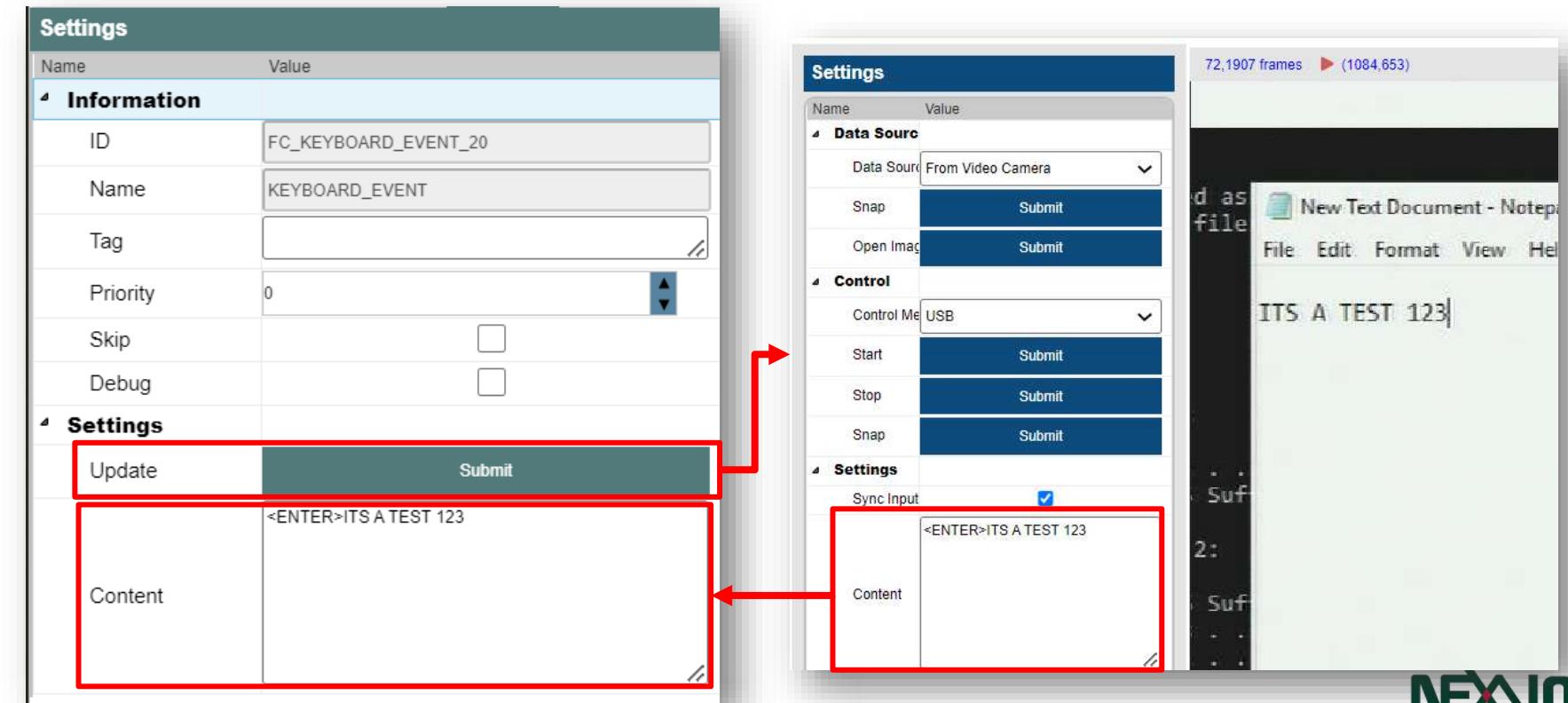
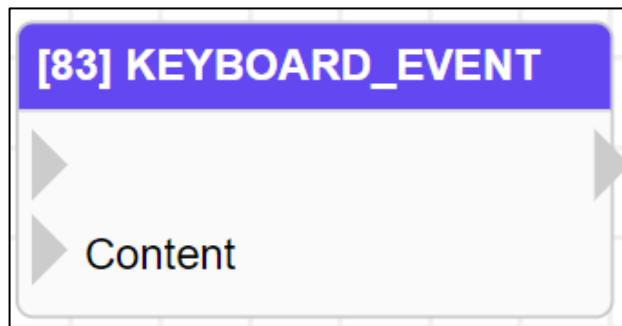
# VIC Flow 控制函式

## • KEYBOARD\_EVENT

可模擬鍵盤並輸入內容的運算子

點擊KEYBOARD\_EVENT運算子設置中的更新，可開啟更新視窗

在更新視窗內進行鍵盤輸入，系統會記錄輸入內容，並在點擊確定後，將此內容帶入運算子設置中



# VIC Flow 控制函式

## • KEYBOARD\_EVENT

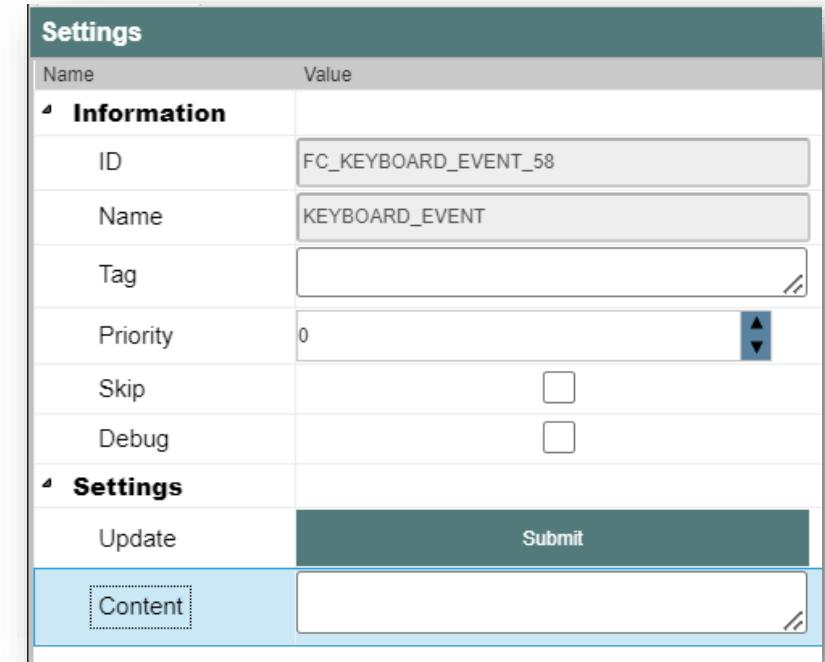
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>內容 (Content)</b>	設置欲輸入的內容 若此欄位不為空，則會套用欄位設定的值	Inline Python Expression String and %1

Settings

Name	Value
ID	FC_KEYBOARD_EVENT_58
Name	KEYBOARD_EVENT
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

Update **Submit**

Content



# VIC Flow 控制函式

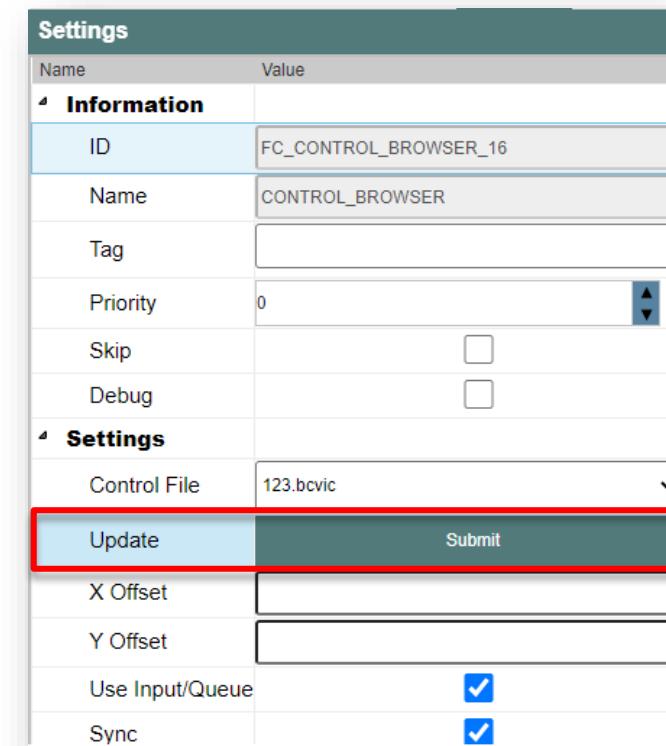
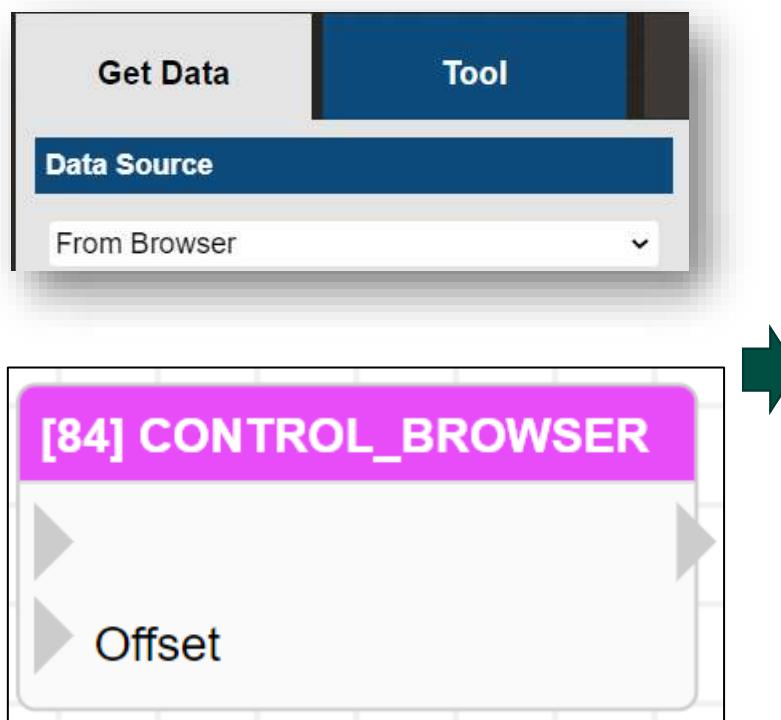
## • CONTROL\_BROWSER

取像來源為瀏覽器時，可執行瀏覽器控制檔的運算子

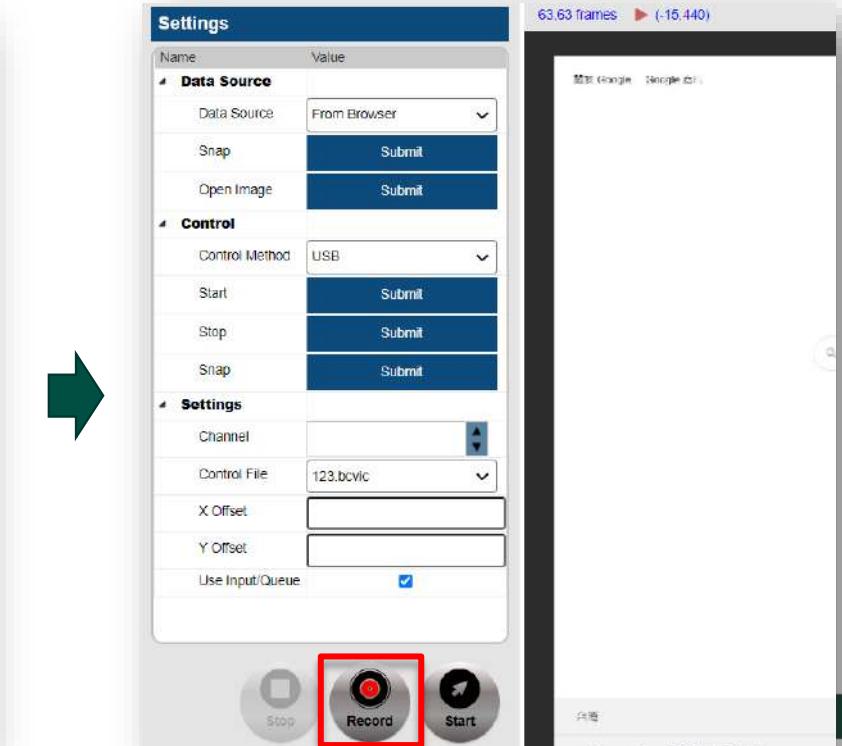
點擊CONTROL\_BROWSER運算子設置中的更新，可開啟更新視窗

在更新視窗內即可錄製控制檔，操作方式和在控制頁面內錄製控制檔的操作相同

若要使用控制檔的**併列**或**輸入**功能，須先到控制檔頁面內的載入控制檔視窗啟用控制檔的屬性



	UI名稱	行為
輸入端	Offset	輸入執行控制檔時欲偏移的量 輸入格式為x_offset, y_offset
輸出端		



# VIC Flow 控制函式

## • CONTROL\_BROWSER

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制。	
<b>設置 (Settings)</b>		
<b>頻道 (Channel)</b>	設置欲執行控制的頻道	Integer
<b>控制檔 (Control File)</b>	設置欲執行的控制檔	
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗。	
<b>X位移 (X Offset)</b>	設置執行的X座標偏移量。 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置執行的Y座標偏移量。 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>使用輸入/佇列 (Use Input/Queue)</b>	設置是否套用控制檔的輸入及佇列屬性 勾選表示使用	
<b>同步 (Sync)</b>	設置是否套用同步執行 勾選表示使用	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

**Settings**

Name	Value
<b>Information</b>	
ID	FC_CONTROL_BROWSER_84
Name	CONTROL_BROWSER
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Control File	
Update	<b>Submit</b>
X Offset	
Y Offset	
Use Input/Queue	<input type="checkbox"/>
Sync	<input type="checkbox"/>
Block Time (sec.)	10

# VIC Flow 控制函式

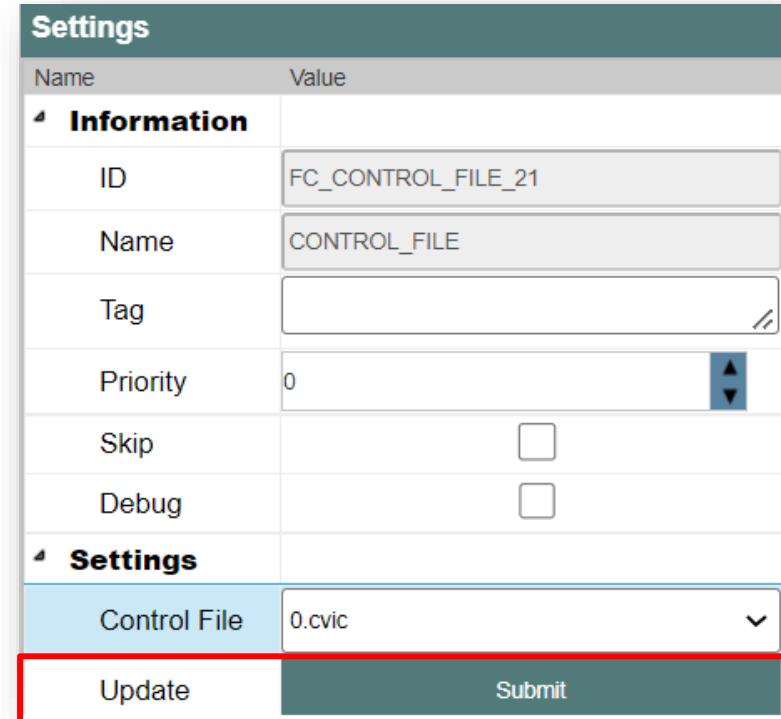
## • CONTROL\_FILE

可執行控制檔的運算子

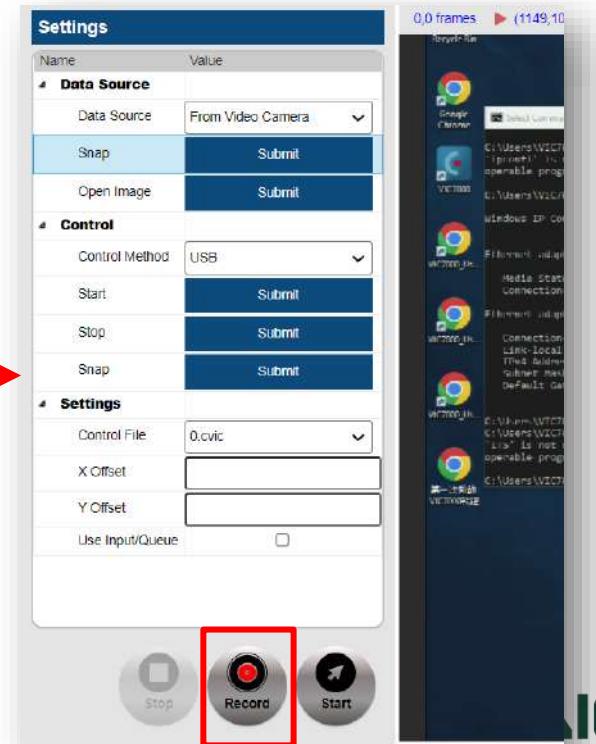
點擊CONTROL\_FILE運算子設置中的更新，可開啟更新視窗

在更新視窗內即可錄製控制檔，操作方式和在控制頁面內錄製控制檔的操作相同

若要使用控制檔的**併列**或**輸入**功能，須先到控制檔頁面內的載入控制檔視窗啟用控制檔的屬性



	UI名稱	行為
輸入端		
Offset		輸入執行控制檔時欲偏移的量 輸入格式為x_offset, y_offset



# VIC Flow 控制函式

## • CONTROL\_FILE

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>控制檔 (Control File)</b>	設置欲執行的控制檔	
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>X位移 (X Offset)</b>	設置執行的X偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置執行的Y偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>使用輸入/佇列 (Use Input/Queue)</b>	設置是否套用控制檔的輸入及佇列屬性 勾選表示使用	
<b>同步 (Sync)</b>	設置是否套用同步執行 勾選表示使用	
<b>阻隔時間(sec.) (Block Time (sec.))</b>	設置運算子的阻隔時間 執行後，在該時間內，不執行此運算子	Integer

**Settings**

Name	Value
ID	FC_CONTROL_FILE_85
Name	CONTROL_FILE
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Control File	0.cvic
Update	<b>Submit</b>
X Offset	
Y Offset	
Use Input/Queue	<input type="checkbox"/>
Sync	<input type="checkbox"/>
Block Time (sec.)	10

# VIC Flow 控制函式

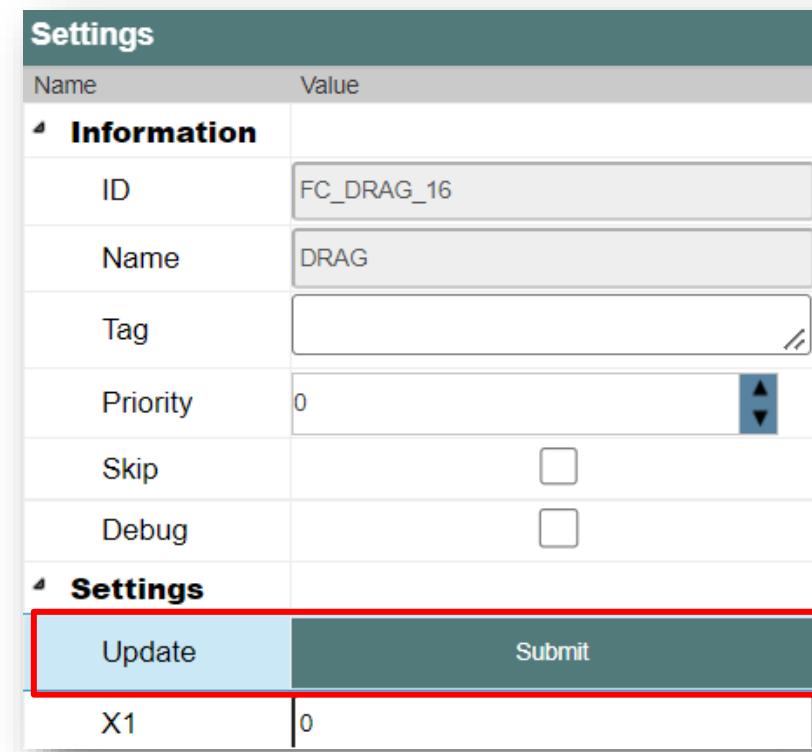
## • DRAG

可執行滑鼠拖移的運算子

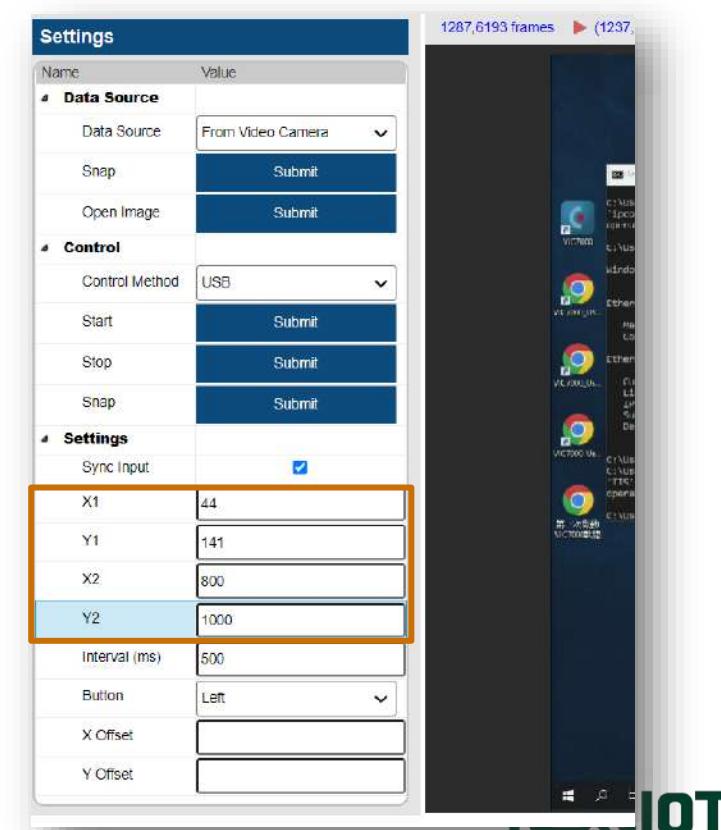
點擊DRAG運算子設置中的更新，可開啟更新視窗

在更新視窗內進行拖移，系統會記錄拖移動作的起點和終點的座標

在點擊確定後，將座標資訊帶入DRAG運算子



	UI名稱	行為
輸入端		
Offset		輸入執行控制檔時欲偏移的量 輸入格式為x_offset, y_offset
輸出端		



# VIC Flow 控制函式

- DRAG

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>更新 (Update)</b>	點擊後，將開啟進階設定視窗	
<b>X1</b>	設置起始位置的X座標值	Python Expression
<b>Y1</b>	設置起始位置的Y座標值	Python Expression
<b>X2</b>	設置終點位置的X座標值	Python Expression
<b>Y2</b>	設置終點位置的Y座標值	Python Expression
<b>時間間隔 (ms) (Interval (ms))</b>	設置兩座標位置的移動間隔時間	Integer
<b>按鍵 (Button)</b>	設置滑鼠點擊的按鍵，分別有左鍵(Left)、中鍵(Middle)以及右鍵(Right)	
<b>X位移 (X Offset)</b>	設置執行的X座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression
<b>Y位移 (Y Offset)</b>	設置執行的Y座標偏移量 若輸入端有值輸入，則套用輸入端的值	Python Expression

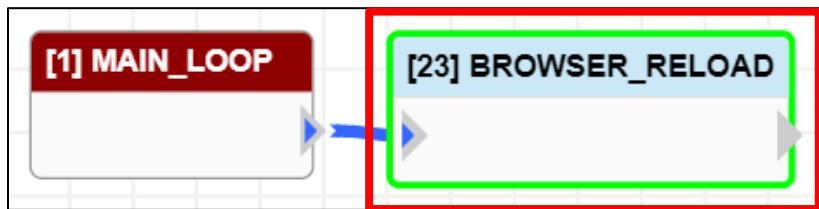
Settings	
Name	Value
<b>Information</b>	
ID	FC_DRAG_86
Name	DRAG
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Settings</b>	
Update	Submit
X1	0
Y1	0
X2	0
Y2	0
Interval (ms)	80
Button	Left
X Offset	
Y Offset	

# VIC Flow 控制函式

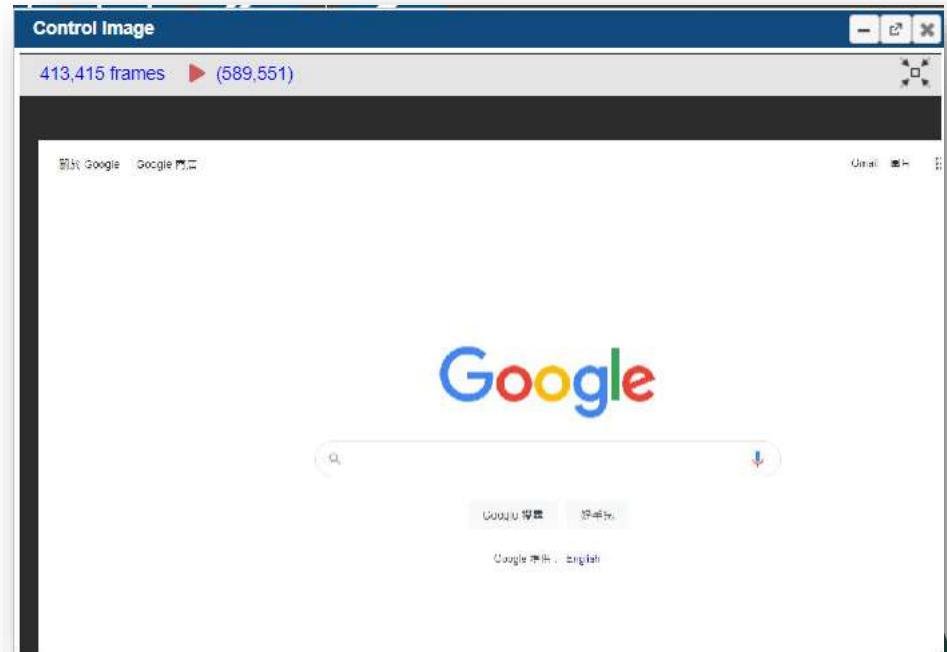
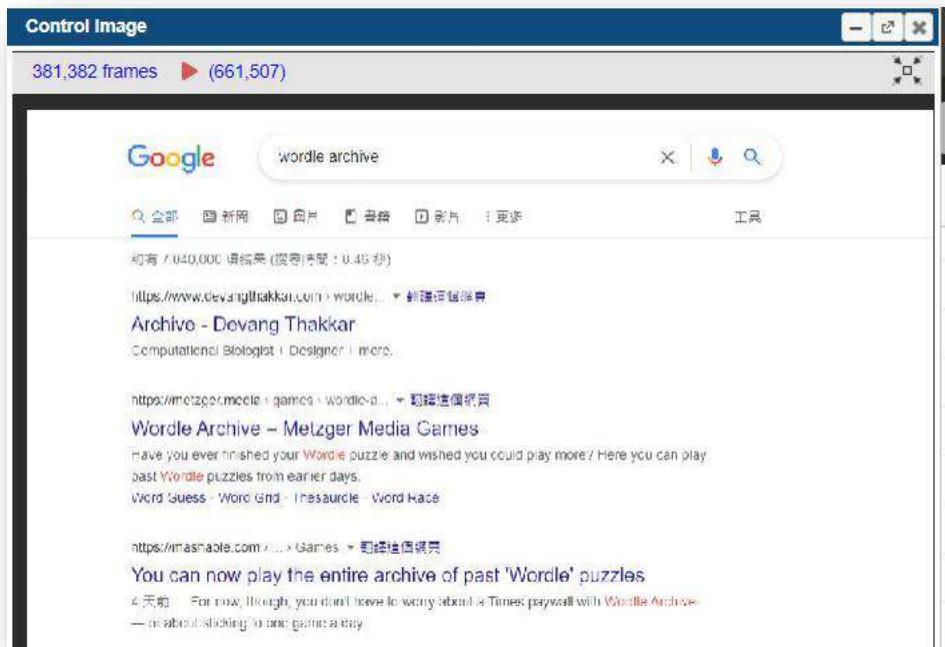
## • BROWSER\_RELOAD

取像來源為來自瀏覽器十，可重新載入設置的瀏覽器網址的運算子

執行運算子後，系統即會重新載入取像設定的網址



	UI名稱	行為
輸入端		
輸出端		



# VIC Flow 控制函式

- BROWSER\_RELOAD

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	

Settings	
Name	Value
^ <b>Information</b>	
ID	FC_BROWSER_RELOAD_88
Name	BROWSER_RELOAD
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>

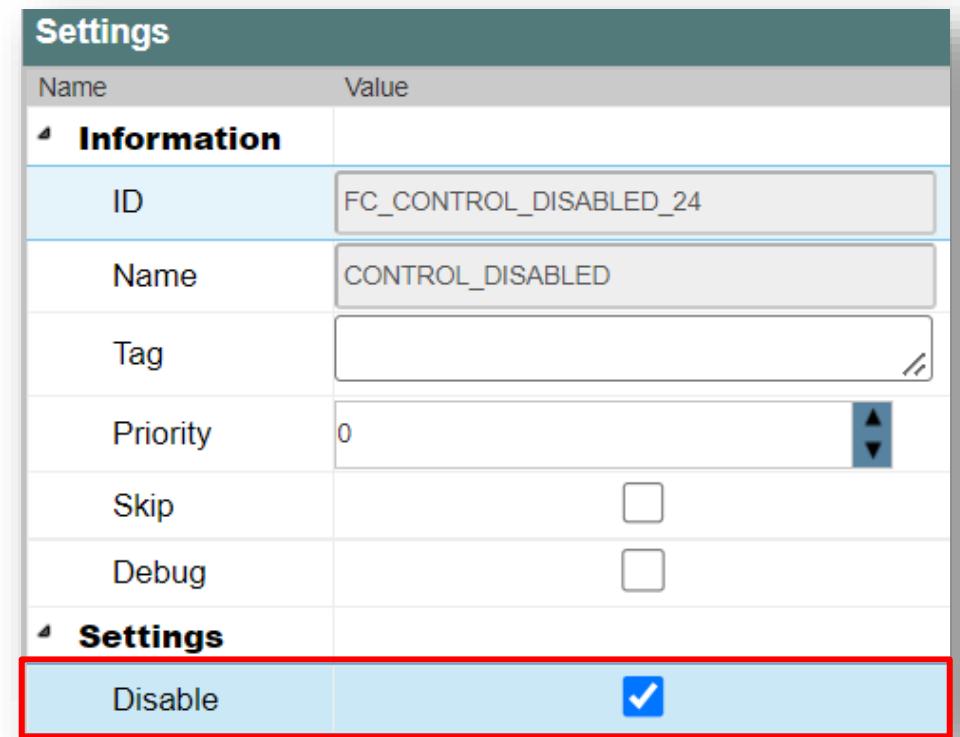
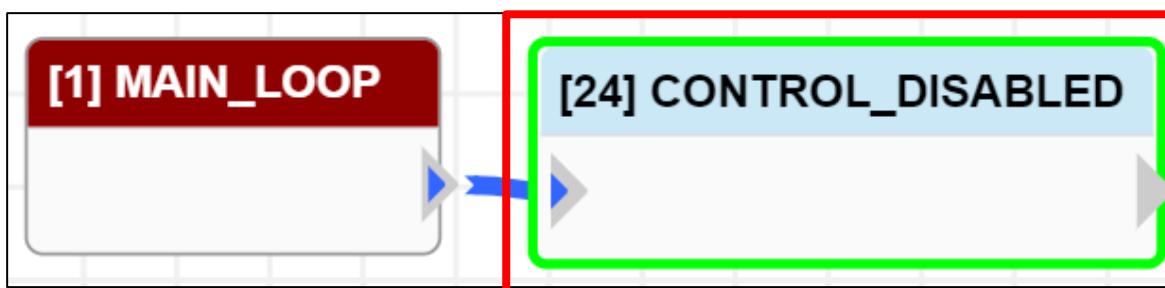
# VIC Flow 控制函式

- **CONTROL\_DISABLED**

可設置是否停用控制功能的運算子

勾選停止並執行該運算子後，控制功能就會停止，此時無法控制受控電腦的滑鼠鍵盤

	UI名稱	行為
輸入端		
輸出端		



# VIC Flow 控制函式

- **CONTROL\_DISABLED**

名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>停止 (Disable)</b>	設置是否要停止執行控制指令 勾選即停止	

Settings

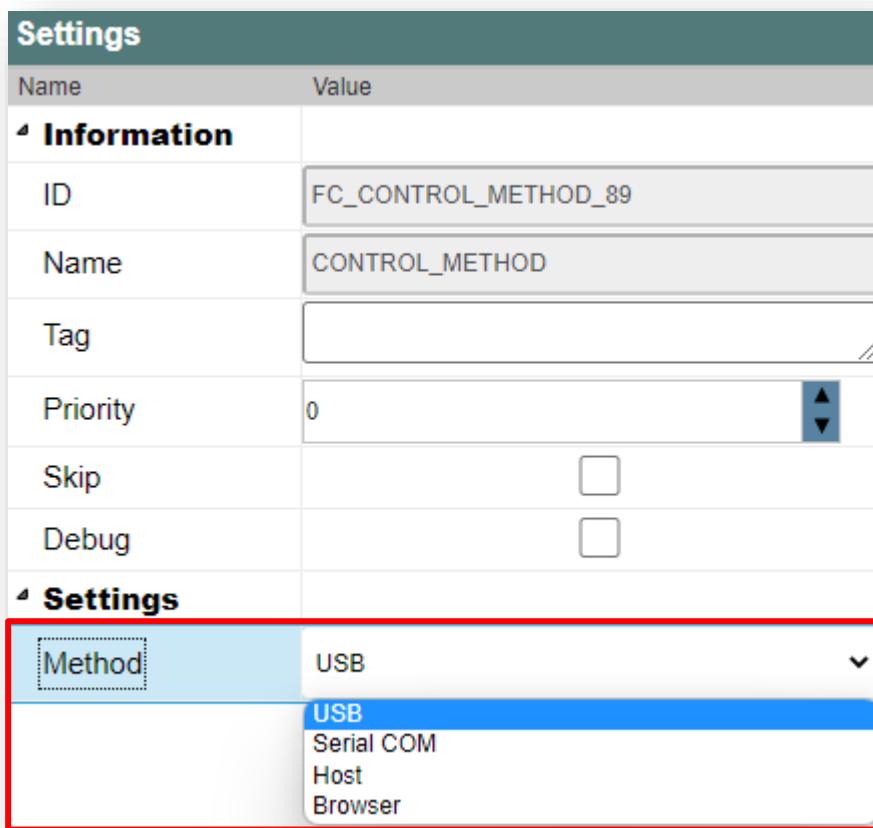
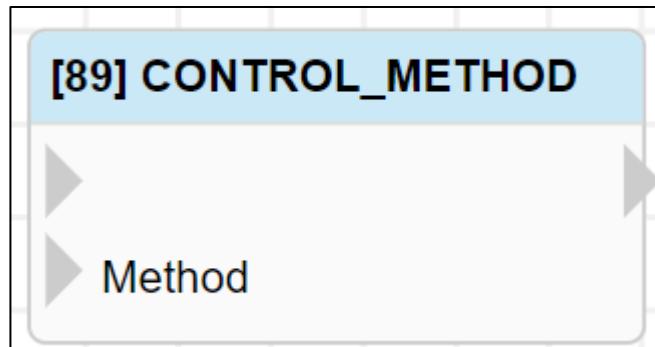
Name	Value
ID	FC_CONTROL_DISABLED_66
Name	CONTROL_DISABLED
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
Disable	<input checked="" type="checkbox"/>

# VIC Flow 控制函式

- **CONTROL\_METHOD**

可設置控制方法的運算子

	UI名稱	行為
輸入端		
	方法	輸入要設定的控制方法 0 = USB、1 = 序列COM 2 = 本機、3 = 瀏覽器
輸出端		



# VIC Flow 控制函式

- **CONTROL\_METHOD**

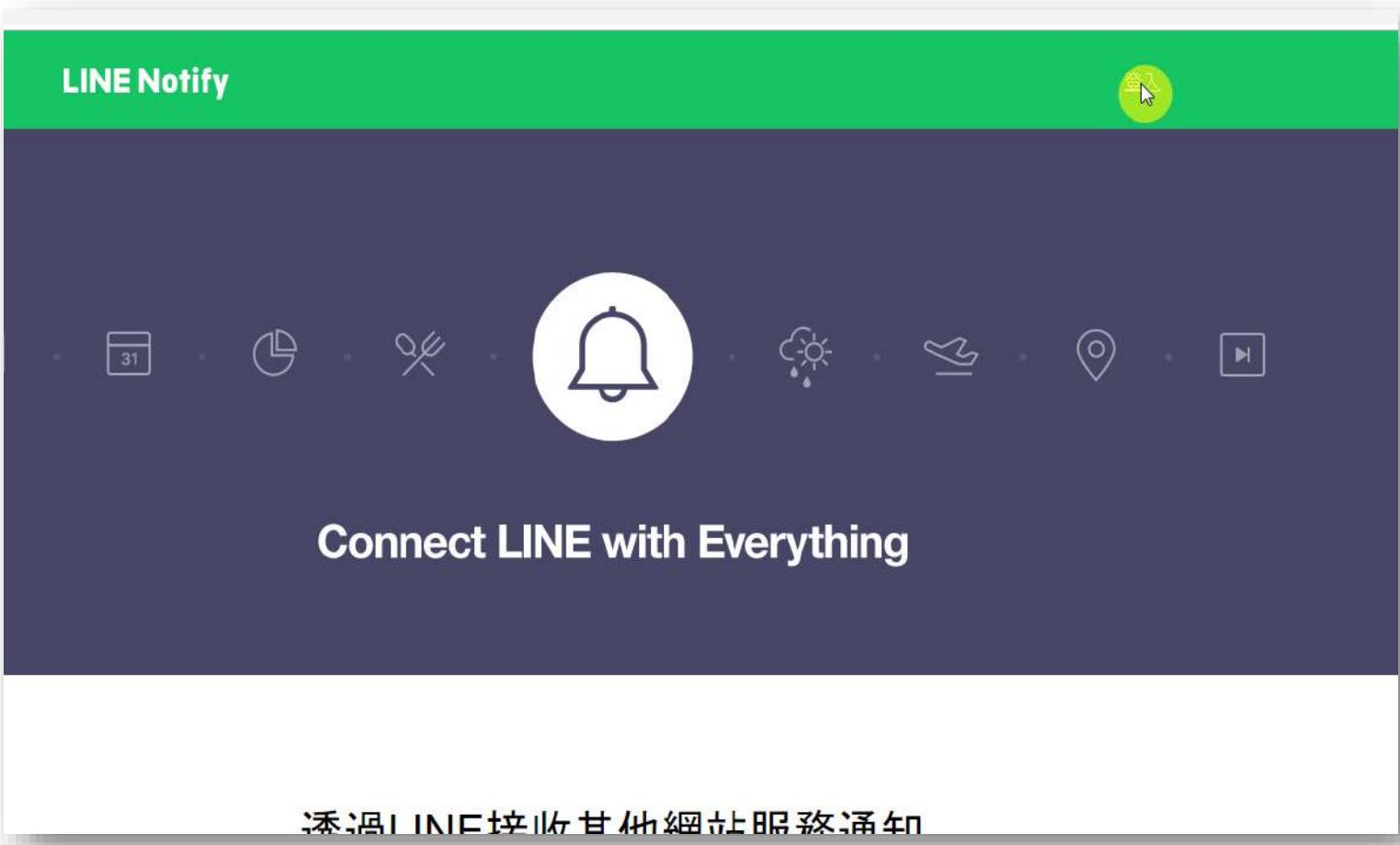
名稱	意義	格式
<b>訊息 (Information)</b>		
<b>ID</b>	運算子的ID	
<b>名稱 (Name)</b>	運算子的名稱	
<b>標籤 (Tag)</b>	設定運算子的標籤	String
<b>優先順序 (Priority)</b>	設置同層級運算子的執行優先順序	Integer
<b>略過 (Skip)</b>	勾選後，即會略過執行此運算子	
<b>偵錯 (Debug)</b>	勾選後，即會啟動該運算子的偵錯機制	
<b>設置 (Settings)</b>		
<b>方法 (Method)</b>	設置控制方法，包含USB、序列COM、本機、瀏覽器	

**Settings**

Name	Value
ID	FC_CONTROL_METHOD_89
Name	CONTROL_METHOD
Tag	
Priority	0
Skip	<input type="checkbox"/>
Debug	<input type="checkbox"/>
<b>Method</b>	
<input type="button" value="USB"/>	

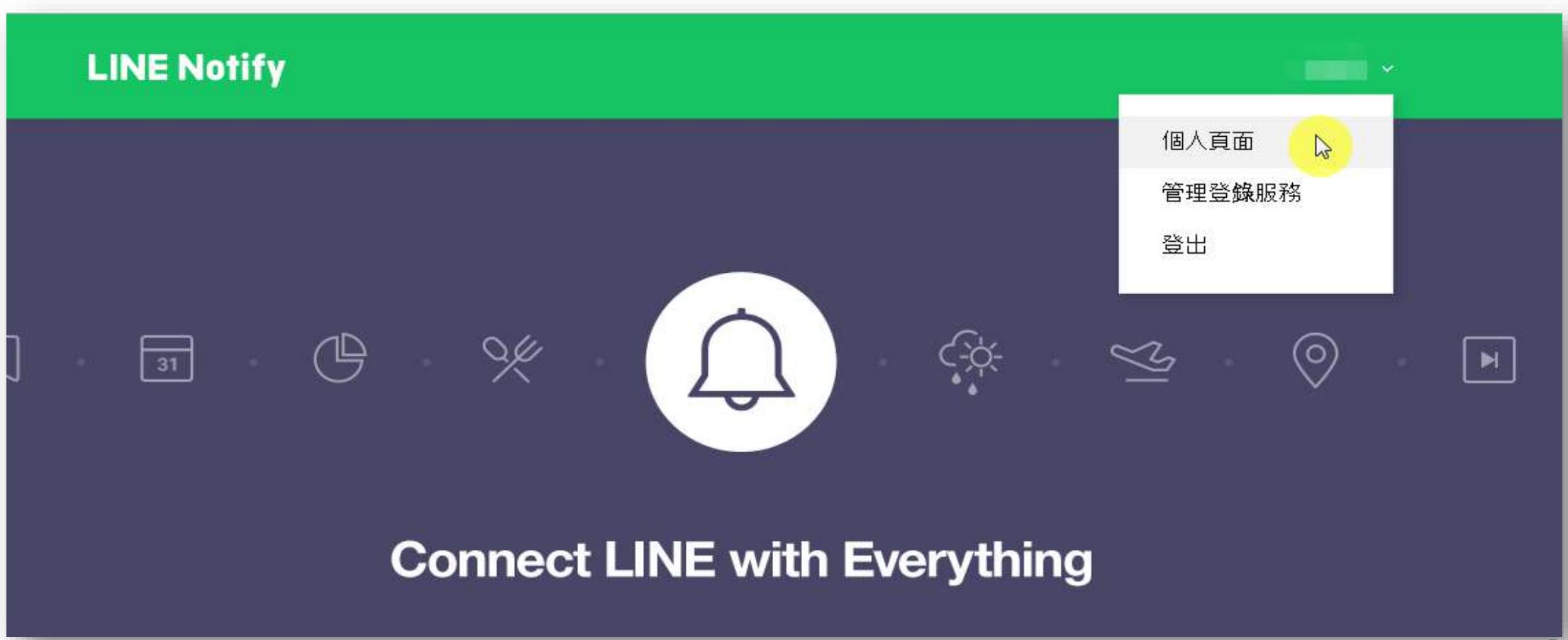
- **LINE Notify**

- 取得權杖：進入 LINE Notify 網站([https://notify-bot.line.me/zh\\_TW/](https://notify-bot.line.me/zh_TW/)) → 點擊登入 → 輸入帳號密碼



- **LINE Notify**

- 取得權杖：點擊右上角帳戶名稱 → 點擊個人頁面



## • LINE Notify

- 取得權杖：點擊發行權杖 → 輸入權杖名稱 → 選擇要接收訊息的聊天室 → 點擊發行



The screenshot shows the 'LINE Notify' API documentation page. In the '已連動的服務' (Connected Services) section, there are three icons: GitHub, IFTTT, and Mackerel. Below these, a button labeled '發行存取權杖(開發人員用)' (Release Access Token (for Developers)) is highlighted with a yellow circle. At the bottom, there is a link to 'LINE Notify API Document'.



The dialog box is titled '發行權杖' (Release Token). It has a text input field containing 'VIC7000' and a dropdown list titled '請選擇您要接收通知的聊天室' (Please select the chat room to receive notifications) with several options listed. At the bottom is a large green '發行' (Release) button.

- LINE Notify

- 取得權杖：記下發行權杖，離開此頁面後，無法再看到此權杖



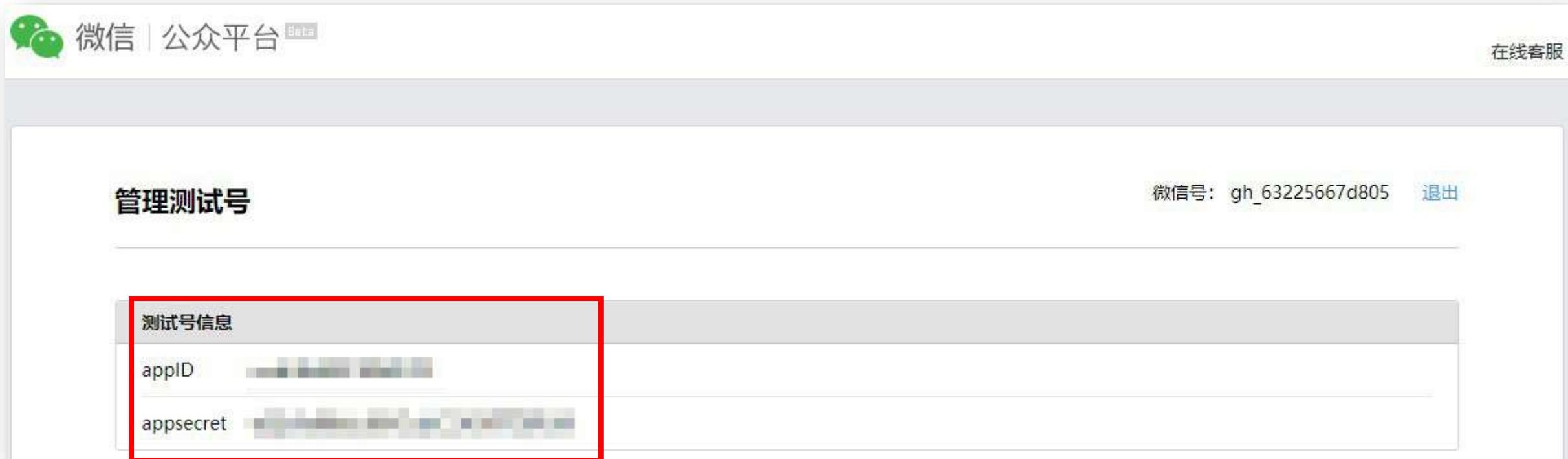
- WeChat

- 取得資訊：進入微信公眾平台網站(<https://mp.weixin.qq.com/debug/cgi-bin/sandbox?t=sandbox/login>)  
→ 點擊登入 → 手機掃描登入



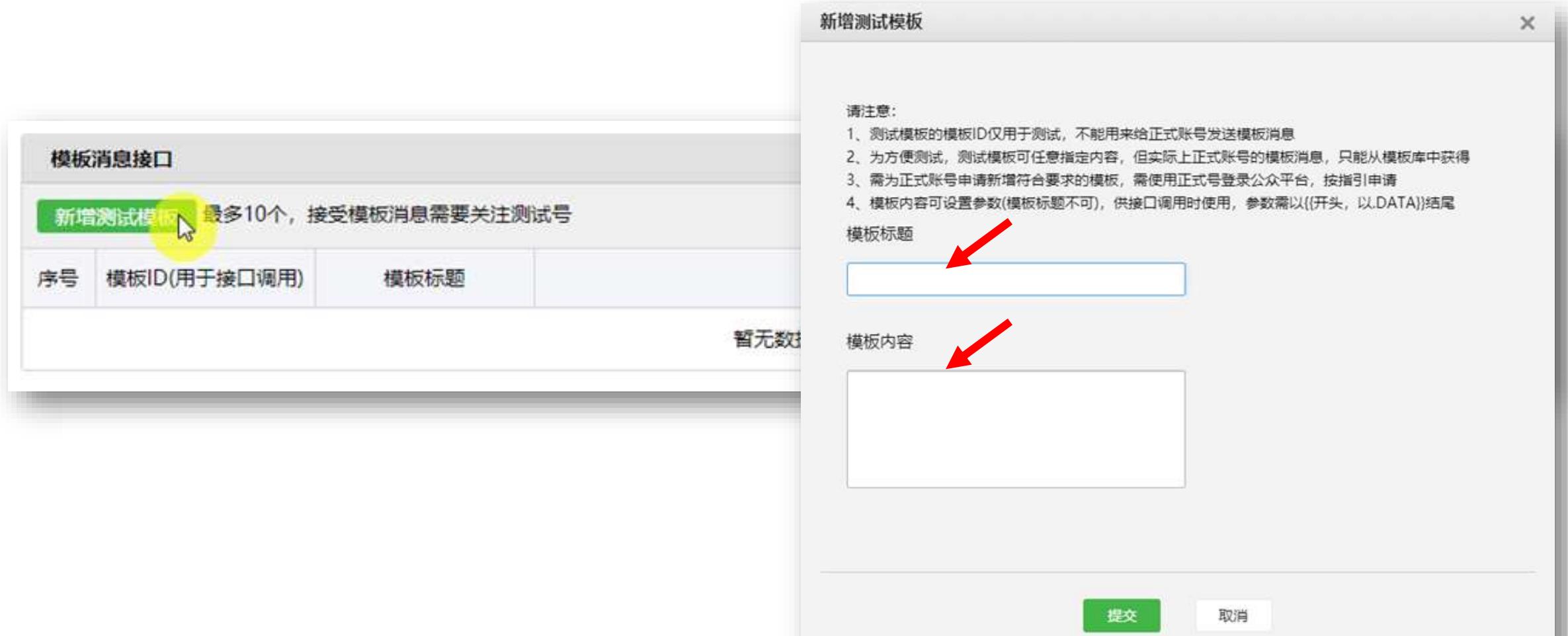
- **WeChat**

- **取得資訊**：登入後，記錄 appID 和 appsecret



## • WeChat

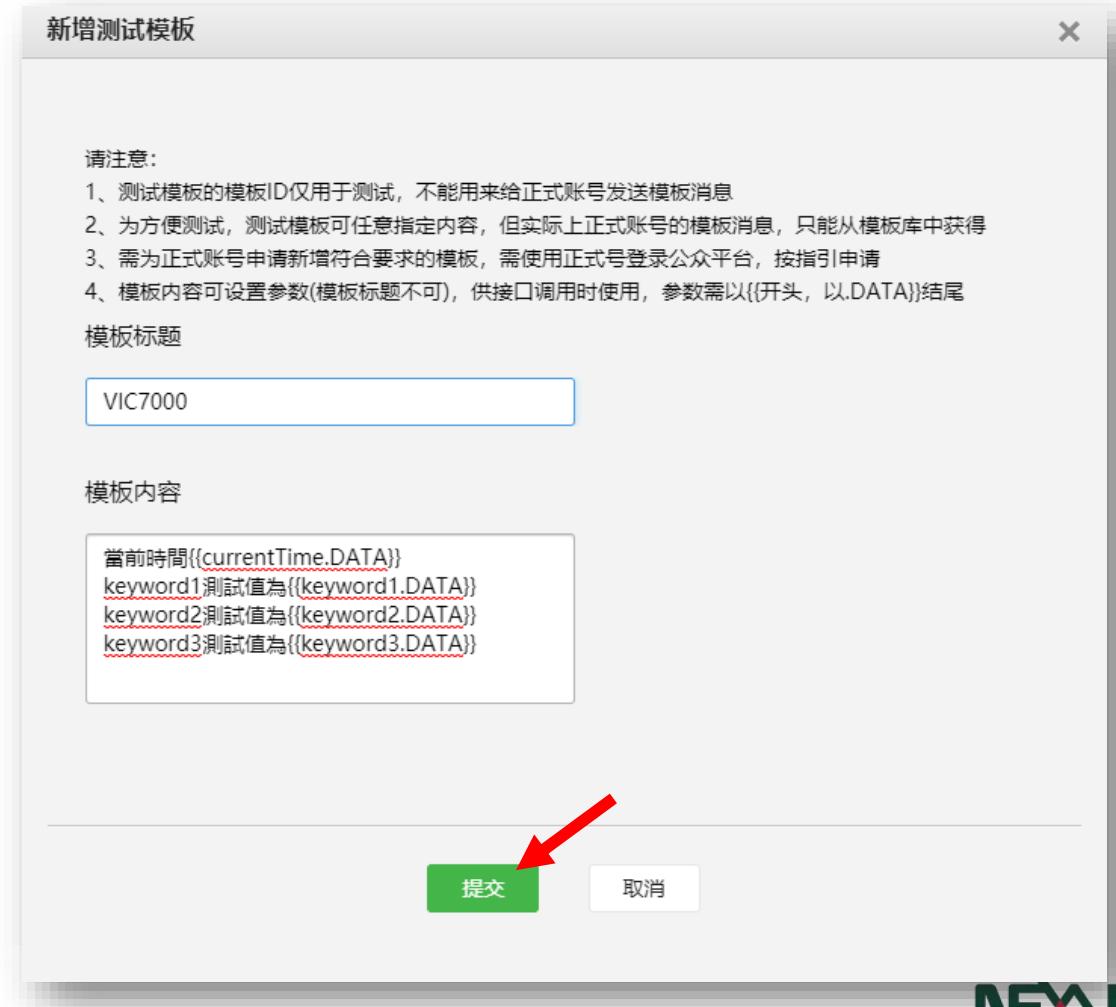
- **設定 templateID**：點擊新增測試模板 → 輸入模板標題和內容



## • WeChat

- 設定 templateID：輸入完成後，點擊提交

腳本參數	模板對應內容
keyword1	<code>{keyword1.DATA}</code>
keyword2	<code>{keyword2.DATA}</code>
keyword3	<code>{keyword3.DATA}</code>
非腳本參數	模板對應內容
當前時間	<code>{currentTime.DATA}</code>



- WeChat

- **設定 templateID**：產生的 templateID 將用於 SEND.WECHAT 和 SEND.WECHAT\_P

模板消息接口

新增测试模板 最多10个，接受模板消息需要关注测试号

序号	模板ID(用于接口调用)	模板标题	模板内容	操作
1	xxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxx	VIC7000	當前時間 {{currentTime.DATA}} keyword1 測試值為{{keyword1.DATA}} keyword2 測試值為{{keyw ord2.DATA}} keyword3 測試值為{{keyword3.DATA}}	<a href="#">刪除</a>

## • WeChat

- **設定 openID**：用手機微信掃描測試號二維碼，進行關注公眾號，須關注此公眾號才可接到通知



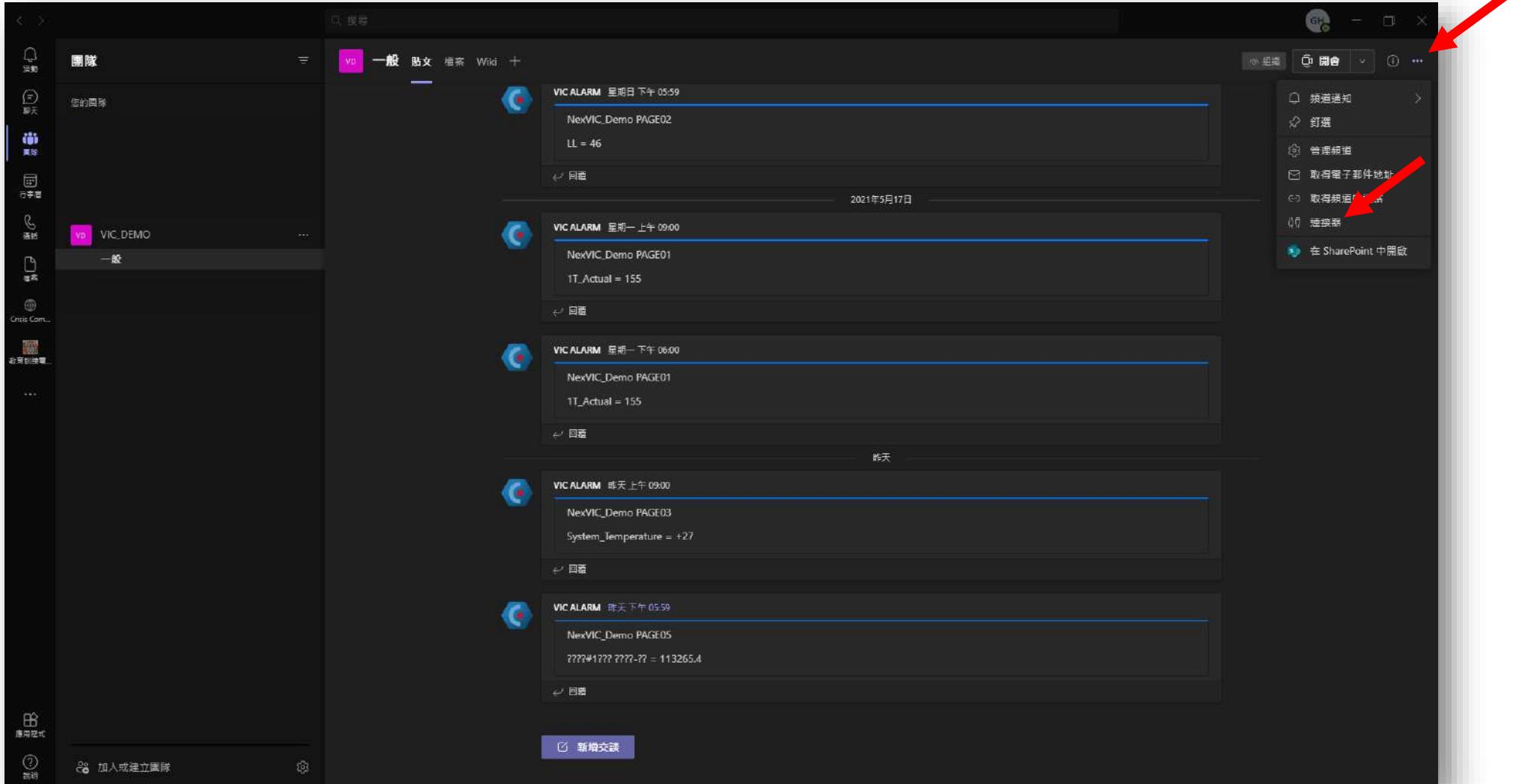
- **WeChat**

- **設定 openID**：有用戶關注後，會出現該用戶的微信號，即為 openID，用於 SEND.WECHAT\_P



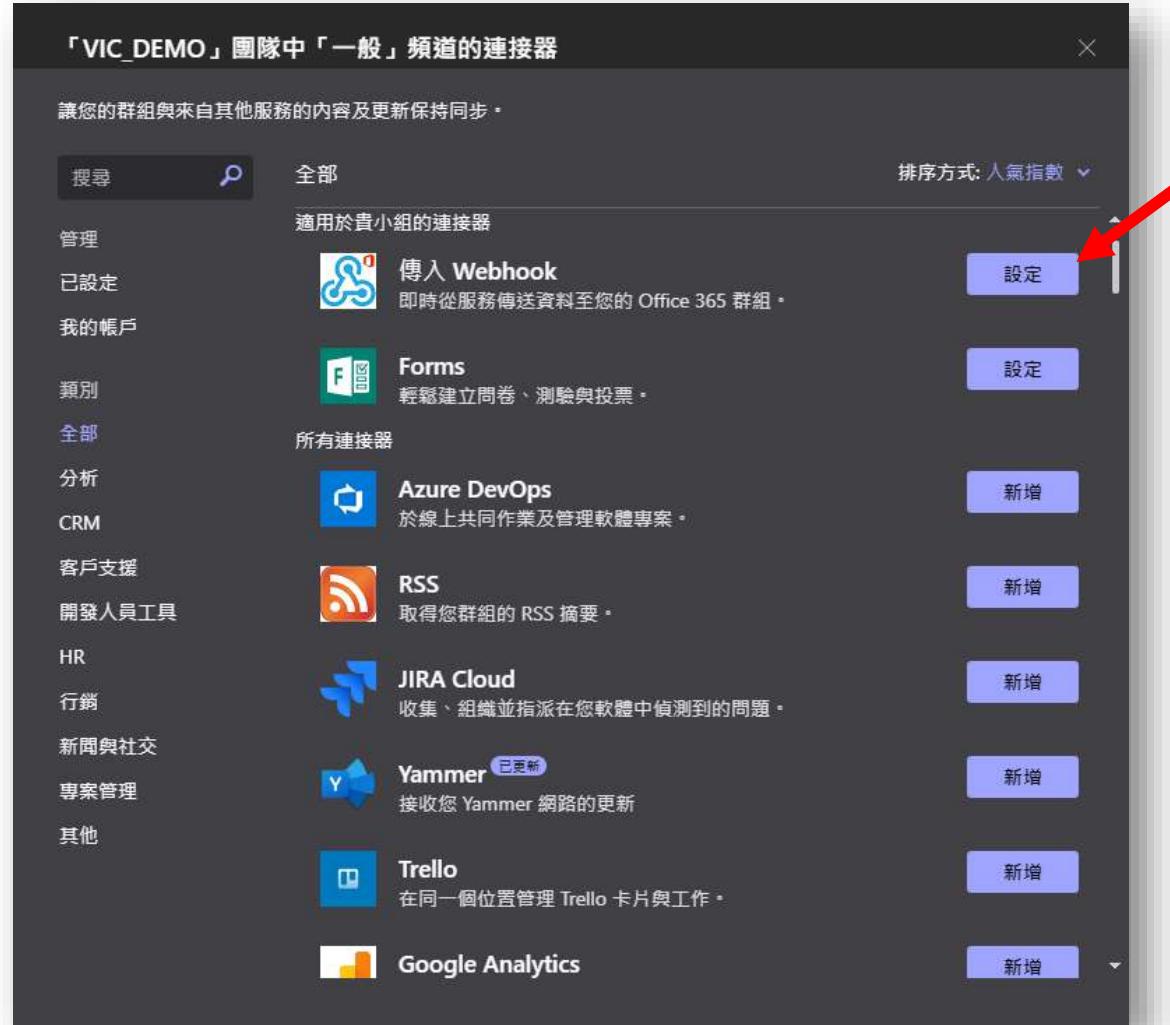
## • Teams

- 取得 Webhook：開啟Teams，選擇要接收訊息的團隊，點擊右上角的其他選項，選擇連接器



## • Teams

- 取得 Webhook：在連接器中找到傳入Webhook，進入它的設定



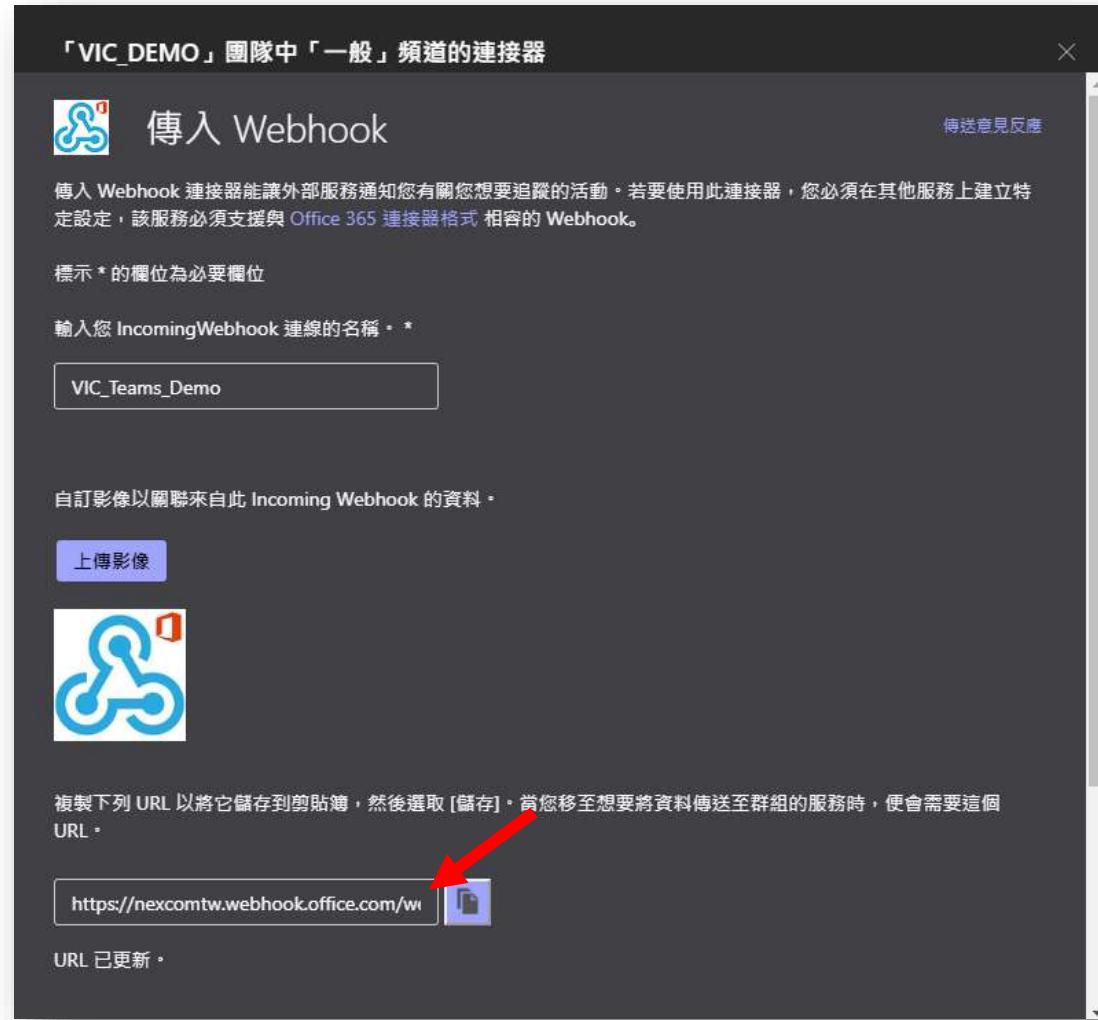
## • Teams

- 取得 Webhook：輸入Webhook的名稱，接著點擊建立



## • Teams

- 取得 Webhook：產生的URL即為Webhook





*Your Partner in Smart Manufacturing*



Q&A



[NexVIC產品教學影片](#)



NexVIC產品服務群組



FB Page:  
NexAlot in Action @