電池外殼箭頭標記 AOI 檢測系統 — 操作手冊

版本：1.0
平台：Raspberry Pi 5，Ubuntu 24.04
程式名稱：DetectArrowMark

目錄

1. 系統簡介
2. 系統架構
3. 啟動程式
4. 主畫面介面說明
    - 4.1 左側 IO 訊號面板
    - 4.2 中央影像預覽區
    - 4.3 右側控制面板
5. 作業模式說明
    - 5.1 即時影像模式
    - 5.2 手動測試模式
    - 5.3 生產模式
6. 相機操作說明
7. 影像操作說明
8. IO 輸出設定說明
9. 檢測結果顯示說明
10. 參數設定說明
11. 檢測演算法原理
12. 設定檔說明
13. 常見問題與注意事項

---

1. 系統簡介

本系統為自動光學檢測（AOI, Automated Optical Inspection）應用程式，專門用於檢測電池外殼底部是否存在箭頭標記。

主要功能

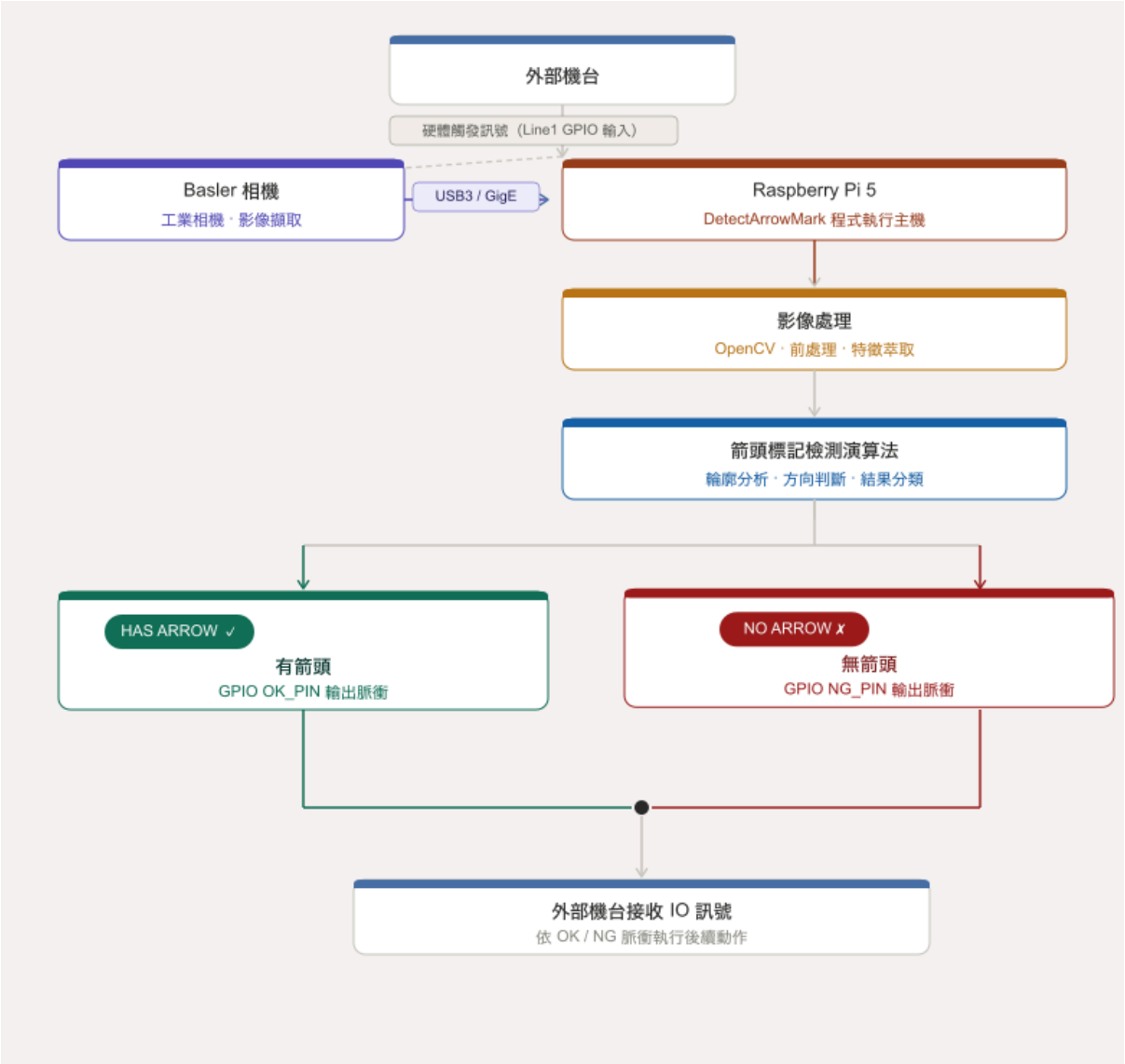
- 透過 Basler 工業相機擷取電池外殼影像
- 自動定位電池外殼位置，並在指定的感興趣區域（ROI）內搜尋箭頭標記
- 依據檢測結果（有箭頭 / 無箭頭）透過 GPIO IO 訊號通知機台
- 提供三種作業模式：即時影像串流、手動軟體觸發、生產用硬體觸發
- 即時顯示 GPIO 腳位狀態，方便確認 IO 訊號輸出狀況

適用場景

場景	說明
生產線品質管制	硬體觸發模式，由機台外部訊號驅動，自動檢測並輸出 OK/NG 訊號
參數調試	即時影像或手動測試模式，方便工程師調整 ROI 及閾值參數
離線影像驗證	載入靜態影像進行參數驗證，無需連接相機

---

2. 系統架構



---

3. 啟動程式

方法一：從桌面選單啟動

在 Linux 應用程式選單中找到「**Detect Arrow Mark**」圖示，點擊啟動。

方法二：從終端機啟動

```
detect-arrow-mark
```

方法三：直接執行腳本

```
cd /home/aoi/AOI20260411
bash detect-arrow-mark.sh
```

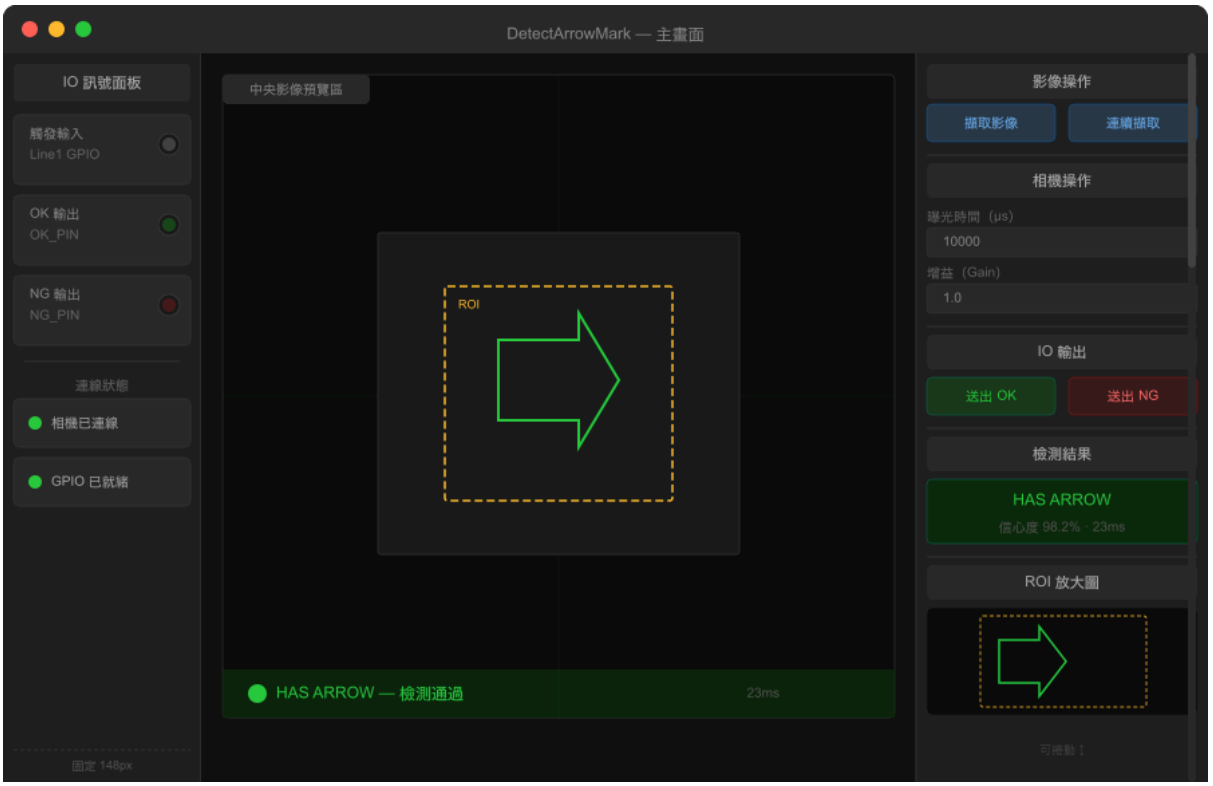
啟動後的初始化

程式啟動時會自動：1. 讀取上次儲存的參數設定（`~/.config/detect-arrow-mark/app_params.json`）2. 讀取 GPIO 腳位設定（`~/.config/detect-arrow-mark/gpio_settings.json`）3. 初始化 GPIO 管理器（若系統不支援，會以無 GPIO 模式運行）4. 顯示主畫面，等待使用者操作

---

4. 主畫面介面說明

主畫面分為三個區域：



4.1 左側 IO 訊號面板（IO 訊號）

標題：IO 訊號
寬度：固定 130px

此面板即時顯示 Raspberry Pi 所有 GPIO 腳位（BCM 編號 2～27）的電壓狀態，讓操作人員隨時確認 IO 訊號是否正常輸出。

圖示說明

圖示顏色	意義
綠色 ●	HIGH（高電位，約 3.3V）
灰色 ●	LOW（低電位，約 0V）
淺灰色 ●	讀取失敗（無法取得狀態）

特別標示

- **OK** 腳位（設定為輸出 OK 訊號的 GPIO）：腳位編號下方有底線標示
- **NG** 腳位（設定為輸出 NG 訊號的 GPIO）：腳位編號下方有底線標示

更新頻率

每 **400** 毫秒自動刷新一次，顯示最新腳位狀態。

4.2 中央影像預覽區

顯示相機擷取或載入的影像，並在檢測完成後疊加以下標示：

標示	說明
綠色矩形框（有箭頭）	標示感興趣區域（ROI）位置
紅色矩形框（無箭頭）	標示感興趣區域（ROI）位置
橙色圓點	標示電池外殼頂點（定位點）
文字標籤	顯示「HAS ARROW」或「NO ARROW」

影像會自動縮放以符合預覽區大小，維持原始長寬比。

4.3 右側控制面板

右側面板包含六個子區塊，由上至下排列，可上下捲動：

(A) 影像操作

控制元件	類型	功能說明
開啟影像	按鈕	開啟檔案對話框，載入靜態影像（PNG / BMP / JPG），載入後可進行離線檢測

控制元件	類型	功能說明
執行檢測	按鈕	對目前顯示的影像執行箭頭標記檢測，並顯示結果（需先有影像）
儲存影像	按鈕	將目前顯示的影像（含標示框）儲存至磁碟，支援 PNG / BMP / JPG / JPEG 格式

(B) 相機操作

控制元件	類型	功能說明
作業模式	單選按鈕群組	選擇相機作業模式（詳見第 5 節）
開始	按鈕	連接相機並依選定模式開始擷取影像
停止	按鈕	停止擷取影像並斷開相機連線
測試	按鈕	僅在「手動測試」模式下可用，每按一次擷取一幀影像並執行檢測
曝光時間 **(µs)**	數值輸入框 + 滑桿	設定相機曝光時間，範圍 100～1,000,000 微秒，需在相機連接前設定或重新連接後生效
影像格式	下拉選單	選擇相機輸出的像素格式（連接相機後才可選擇，選項由相機硬體決定）
相機狀態	文字標籤	顯示目前相機狀態（開啟中 / 串流中 / 觸發中 / 已關閉 / 錯誤訊息）

影像格式選項說明：

格式名稱	說明
Mono8	8-bit 灰階，單通道，推薦用於本檢測任務
BayerRG / BayerGB / BayerGR / BayerBG	Bayer 原始格式，程式自動去馬賽克轉為彩色
RGB8	24-bit 彩色，3 通道 RGB

注意：檢測演算法使用灰階影像，選用 Mono8 效能最佳。

(C) IO 輸出

控制元件	類型	功能說明
IO 狀態標籤	文字標籤	顯示目前 IO 設定摘要（啟用狀態、OK/NG 腳位編號、脈衝時間）
IO 設定	按鈕	開啟 GPIO 設定對話框，進行詳細的 IO 腳位設定

(D) 檢測結果

顯示元件	說明
結果標籤	大字體顯示檢測結論：• **HAS ARROW**（綠色）：偵測到箭頭標記• **NO ARROW**（紅色）：未偵測到箭頭標記
亮點像素數	顯示 ROI 內亮像素計數及最低門檻值（格式：**亮點像素**：XXX / **門檻**：XXX）
外殼定位座標	顯示偵測到的電池外殼頂點位置（格式：**外殼位置**：**(列，行)**）

(E) ROI 放大圖

固定 **280×280** 像素的放大預覽視窗，顯示感興趣區域（ROI）的放大影像：

顯示內容	說明
深灰色背景	ROI 的一般暗色像素
藍色高亮像素	超過亮度門檻的像素，即箭頭標記的潛在位置

此放大圖讓操作人員清楚看到 ROI 內的亮點分佈，方便調整參數。

(F) 參數設定

共 7 個參數，每個參數以「標籤 + 數值輸入框 + 滑桿」的組合呈現，輸入框與滑桿同步連動。

參數名稱	說明	調整範圍
深色門檻 (dark_thresh)	定義「深色像素」的灰階上限，用於定位電池外殼邊緣。灰階值低於此門檻	50 ～ 255

參數名稱	說明	調整範圍
	的像素視為深色(外殼區域)	
ROI 列偏移 (roi_row_offset)	ROI 起點相對於外殼頂點的垂直偏移量(像素)	0 ～ 800
ROI 行偏移 (roi_col_offset)	ROI 起點相對於外殼頂點的水平偏移量(像素)	0 ～ 800
ROI 高度 (roi_height)	感興趣區域的高度(像素)	10 ～ 800
ROI 寬度 (roi_width)	感興趣區域的寬度(像素)	10 ～ 800
高亮門檻 (bright_thresh)	定義「亮點像素」的灰階下限，灰階值高於此門檻的像素視為亮點(箭頭標記)	50 ～ 255
最低高亮像素 (bright_count_min)	判斷為「有箭頭」所需的最少亮點像素數。若 ROI 內亮點數量 ≥ 此值，判定為 HAS ARROW	0 ～ 5000

控制元件	類型	功能說明
重設預設值	按鈕	將所有參數重設為出廠預設值

出廠預設值:

參數	預設值
深色門檻	124
ROI 列偏移	129
ROI 行偏移	201
ROI 高度	275
ROI 寬度	289
高亮門檻	145
最低高亮像素	400

5. 作業模式說明

在「相機操作」區塊中，透過單選按鈕選擇以下三種作業模式之一：

5.1 即時影像模式（即時影像）

適用時機：參數調試、初始設定、觀察相機影像品質

運作流程：

點擊「開始」
→ **相機連線並開始連續擷取影像（5 FPS）**
→ **每一幀自動執行箭頭標記檢測**
→ **持續更新預覽畫面、檢測結果、GPIO 輸出**
→ **點擊「停止」**
→ **相機停止串流並斷線**

特性：- 相機以「自由運行（Free Run）」模式持續擷取影像 - 幀率限制為每秒 5 幀（5 FPS），避免系統過載 - 每幀均自動執行完整檢測流程並輸出 IO 訊號

按鈕狀態：

按鈕	狀態
開始	啟動後禁用
停止	啟動後啟用
測試	禁用（此模式不需要手動觸發）

5.2 手動測試模式（手動測試）

適用時機：逐幀測試、確認單次檢測結果、參數微調

運作流程：

點擊「開始」
→ **相機連線，進入軟體觸發待命狀態**
→ **點擊「測試」**
→ **相機擷取單幀影像（ExecuteSoftwareTrigger）**
→ **執行箭頭標記檢測，更新畫面與 IO 訊號**
→ **可再次點擊「測試」重複擷取**
→ **點擊「停止」**
→ **相機解除觸發模式並斷線**

特性：- 相機進入「軟體觸發（Software Trigger）」模式 - 僅在使用者按下「測試」按鈕時才擷取一幀 - 適合精確控制每次擷取時機，方便逐步驗證

按鈕狀態：

按鈕	狀態
開始	啟動後禁用
停止	啟動後啟用
測試	啟動後啟用（每按一次擷取一幀）

5.3 生產模式(生產模式)

適用時機: 正式生產線自動化檢測

運作流程:

點擊「開始」
→ 相機連線，進入硬體觸發待命狀態
→ 主視窗背景變為淡綠色(#c8f0c8)，提示已進入生產模式
→ 等待外部機台透過 Line1 GPIO 腳位送出觸發訊號
→ 收到觸發訊號 → 擷取一幀影像
→ 執行箭頭標記檢測
→ 輸出 OK 或 NG GPIO 脈衝訊號通知機台
→ 繼續等待下次觸發訊號
→ 點擊「停止」
→ 解除觸發模式，視窗背景恢復正常

特性: - 相機進入「硬體觸發(Hardware Trigger，TriggerSource=Line1)」模式 - 由外部機台控制擷取時機，程式被動等待觸發 - 具備 **0.2** 秒去抖動機制，防止連續觸發誤動作 - 視窗背景顯示淡綠色，直觀提示目前處於生產作業狀態

按鈕狀態:

按鈕	狀態
開始	啟動後禁用
停止	啟動後啟用
測試	禁用(生產模式由硬體觸發，不支援手動觸發)

視覺提示對照:

狀態	視窗背景色
非生產模式	系統預設色
生產模式運行中	淡綠色(#c8f0c8)

---

6. 相機操作說明

6.1 相機連線

1. 確認 Basler 相機已透過 USB3 或 GigE 連接至 Raspberry Pi
2. 在右側面板選擇「作業模式」
3. 設定所需的「曝光時間」
4. 點擊「開始」按鈕
5. 觀察「相機狀態」標籤確認連線成功

相機狀態說明:

狀態文字	意義
已關閉	相機未連接
開啟中	相機已連接，正在初始化

狀態文字	意義
串流中	相機連續串流中（即時影像模式）
觸發中	相機等待觸發訊號（手動測試或生產模式）
錯誤：…	發生錯誤，錯誤訊息顯示於括號內

6.2 調整曝光時間

1. 在「曝光時間 (µs)」欄位輸入數值，或拖動滑桿調整
2. 範圍：**100** ～ **1,000,000** 微秒（0.1ms ～ 1s）
3. 典型生產環境建議值：**10,000 µs**（**10** 毫秒）

> 注意：曝光時間需在相機未啟動時設定，或停止後修改再重新啟動，變更才會生效。

6.3 選擇影像格式

1. 點擊「開始」連接相機後，「影像格式」下拉選單會列出該相機支援的格式
2. 選擇所需格式（建議選擇 **Mono8** 以獲得最佳效能）
3. 變更格式後需重新啟動相機才會生效

6.4 相機斷線

點擊「停止」按鈕，程式會：1. 停止影像擷取 2. 解除觸發模式（若為觸發模式） 3. 斷開相機連線 4. 相機狀態更新為「已關閉」

---

7. 影像操作說明

7.1 載入靜態影像

無需連接相機，可載入圖片進行離線參數調試：

1. 點擊「開啟影像」按鈕
2. 在檔案對話框中選擇圖片（支援 PNG、BMP、JPG）
3. 影像載入後顯示於中央預覽區
4. 點擊「執行檢測」進行箭頭標記檢測

7.2 執行手動檢測

對目前顯示的影像進行檢測：

1. 確認影像已載入（從檔案或相機擷取）
2. 點擊「執行檢測」按鈕
3. 程式執行完整檢測流程
4. 結果顯示於「檢測結果」區塊，預覽圖疊加標示框

7.3 儲存影像

將帶有檢測標示的影像儲存：

1. 確認影像已載入且已執行檢測
2. 點擊「儲存影像」按鈕

3. 在對話框中選擇儲存位置與格式（PNG / BMP / JPG / JPEG）
4. 點擊儲存

---

8. IO 輸出設定說明

8.1 開啟 GPIO 設定對話框

點擊「IO 設定」按鈕，開啟 GPIO 設定視窗。

8.2 GPIO 設定對話框介面

左側設定區

控制元件	說明
啟用 **GPIO** 輸出 核取方塊	勾選後啟用 GPIO 輸出功能；取消勾選則停用（僅顯示，不輸出訊號）
OK 腳位 數值輸入框	設定輸出 OK 訊號的 BCM GPIO 編號（預設：17）
NG 腳位 數值輸入框	設定輸出 NG 訊號的 BCM GPIO 編號（預設：18）
脈衝時間 **(ms)** 數值輸入框	設定 IO 輸出脈衝的持續時間，單位毫秒（預設：500ms）
測試脈衝（**OK**）按鈕	立即輸出一次 OK 腳位脈衝，用於測試 IO 硬體接線
測試脈衝（**NG**）按鈕	立即輸出一次 NG 腳位脈衝，用於測試 IO 硬體接線
儲存 按鈕	儲存設定並關閉對話框
取消 按鈕	放棄變更並關閉對話框

右側 *GPIO* 狀態顯示區

即時顯示 BCM 2～27 所有 GPIO 腳位的目前電壓狀態，每 400ms 更新一次。

指示燈顏色	意義
綠色 ●	HIGH（高電位）
灰色 ●	LOW（低電位）
淺灰色 ●	讀取失敗

8.3 IO 訊號輸出邏輯

每次檢測完成後，依據結果自動輸出一次脈衝訊號：

偵測到箭頭（HAS ARROW）→ OK_PIN 輸出 HIGH 脈衝，持續 pulse_ms 毫秒後回到 LOW
未偵測到箭頭（NO ARROW）→ NG_PIN 輸出 HIGH 脈衝，持續 pulse_ms 毫秒後回到 LOW

8.4 硬體接線參考

GPIO 腳位	功能	預設 BCM 編號
OK 輸出	有箭頭標記時輸出高電位脈衝	BCM 17
NG 輸出	無箭頭標記時輸出高電位脈衝	BCM 18
硬體觸發輸入	接收機台觸發訊號（生產模式）	Line1（相機 I/O）

注意：Raspberry Pi GPIO 電壓為 **3.3V**，請確認接收端設備相容，必要時加裝電位轉換模組。

---

9. 檢測結果顯示說明

9.1 結果標籤

顯示內容	意義	顏色
HAS ARROW	偵測到箭頭標記，判定合格	綠色
NO ARROW	未偵測到箭頭標記，判定不合格	紅色

9.2 亮點像素資訊

顯示格式：**亮點像素：[計數] / 門檻：[最低高亮像素]**

- 計數：ROI 區域內灰階值超過「高亮門檻」的像素總數
- 門檻：目前設定的「最低高亮像素」參數值
- 當計數 ≥ 門檻，判定為 HAS ARROW

9.3 外殼定位座標

顯示格式：**外殼位置：(列，行)**

- 程式自動掃描影像，找出電池外殼的左上角頂點位置
- 此座標為 ROI 偏移計算的基準點
- 若掃描失敗（找不到外殼），結果欄位顯示錯誤訊息

---

10. 參數設定說明

10.1 參數調整原則

良好的參數設定需確保：1. 外殼定位穩定：調整「深色門檻」使程式能準確找到外殼邊緣 2. **ROI** 準確覆蓋箭頭：調整 ROI 偏移及尺寸，使 ROI 範圍完整涵蓋箭頭標記位置 3. 亮點識別正確：調整「高亮門檻」區分箭頭亮點與背景雜訊 4. 計數門檻合適：「最低高亮像素」設定得宜，避免誤判

10.2 參數調試步驟

步驟一：確認外殼定位 1. 選擇「即時影像」模式並點擊「開始」（或載入靜態影像） 2. 觀察中央預覽區，確認橙色圓點位於電池外殼頂點 3. 若圓點位置不正確，調整「深色門檻」

步驟二：調整 **ROI** 位置 1. 觀察 ROI 矩形框（綠色或紅色框）是否覆蓋箭頭標記位置 2. 調整「ROI 列偏移」和「ROI 行偏移」移動 ROI 起點 3. 調整「ROI 高度」和「ROI 寬度」改變 ROI 大小

步驟三：調整亮點識別 1. 觀察右側「ROI 放大圖」，確認藍色高亮像素分佈 2. 若藍色像素太少（箭頭未顯示），降低「高亮門檻」 3. 若藍色像素太多（雜訊過多），提高「高亮門檻」

步驟四：設定計數門檻 1. 確認有箭頭時的亮點計數（觀察「亮點像素」數值） 2. 確認無箭頭時的亮點計數 3. 將「最低高亮像素」設定在有箭頭與無箭頭計數之間的中間值

10.3 參數存取

- 自動儲存：程式關閉時自動儲存所有參數至設定檔
- 自動載入：程式啟動時自動讀取上次儲存的參數
- 重設預設值：點擊「重設預設值」按鈕可恢復出廠設定

---

11. 檢測演算法原理

11.1 演算法流程



11.2 結果輸出格式

```
{
  "has_arrow": true / false,
  "bright_count": 亮點像素數,
```

```
  "shell_topleft": (列, 行),   ← 外殼頂點座標
  "roi_topleft": (列, 行),     ← ROI 起點座標
  "roi": 灰階影像陣列            ← ROI 區域原始資料
}
```

---

12. 設定檔說明

設定檔儲存於：`~/.config/detect-arrow-mark/`

12.1 app_params.json（應用程式參數）

```json
{
  "dark_thresh": 124,
  "roi_row_offset": 129,
  "roi_col_offset": 201,
  "roi_height": 275,
  "roi_width": 289,
  "bright_thresh": 145,
  "bright_count_min": 400,
  "exposure_time": 10000,
  "operation_mode": "realtime",
  "pixel_format": "Mono8"
}
```

`operation_mode` 可為：`realtime`、`software`、`hardware`

12.2 gpio_settings.json（GPIO 設定）

```json
{
  "enabled": true,
  "ok_pin": 17,
  "ng_pin": 18,
  "pulse_ms": 500
}
```

---

13. 常見問題與注意事項

問題一：點擊「開始」後相機無法連線

可能原因及處理方式：- 確認 USB3 或 GigE 連線是否牢固 - 確認 Basler 相機驅動程式已正確安裝（pypylon）- 確認相機電源指示燈正常 - 嘗試重新插拔 USB 連接線後再點擊「開始」

問題二：手動測試模式下點擊「測試」沒有反應

可能原因及處理方式：- 確認已先點擊「開始」完成相機連線 - 確認相機狀態標籤顯示「觸發中」- 若狀態異常，點擊「停止」後重新點擊「開始」

問題三：找不到外殼（無法定位電池外殼）

可能原因及處理方式：- 調整「深色門檻」：若影像整體偏亮，可降低門檻值；若整體偏暗，可提高門檻值 - 確認電池外殼在影像視野內且清晰可見 - 調整相機曝光時間，避免過曝或曝光不足

問題四：IO 訊號未正常輸出

可能原因及處理方式：- 確認「IO 設定」對話框中已勾選「啟用 GPIO 輸出」 - 使用「測試脈衝」按鈕確認 GPIO 腳位是否正常動作 - 確認 OK/NG 腳位編號設定與實際硬體接線相符 - 確認系統具備 lgpio 程式庫（Raspberry Pi 5 使用 gpiochip4）

問題五：生產模式下機台觸發訊號無法觸發相機

可能原因及處理方式：- 確認觸發訊號接線至相機的 Line1 輸入腳位 - 確認觸發訊號電壓與相機規格相符 - 確認相機狀態標籤顯示「觸發中」（表示程式已就緒） - 注意去抖動機制：兩次觸發間隔至少需 0.2 秒以上

注意事項

1. 程式關閉時自動儲存所有參數，下次啟動時自動恢復
2. 生產模式下視窗背景顯示淡綠色，確認此視覺提示再開始生產
3. **Raspberry Pi GPIO** 電壓為 **3.3V**，請確認外部設備的輸入電壓相容性
4. 即時影像模式幀率上限為 5 FPS，以確保系統穩定運行
5. 修改參數後，在即時影像模式下可立即看到效果，無需重新啟動

本手冊適用於 *DetectArrowMark v1.0*，平台 *Raspberry Pi 5*，*Ubuntu 24.04*