

硬體安裝與接線

 最後更新: 2026-03-14 |  負責人: KC

本指南涵蓋 Arduino Opta MES Gateway 的實體特性、分割區規劃、安裝以及周邊整合。

01

1. 硬體版本與辨識

Arduino Opta (常以 **Finder Opta** 品牌出貨) 共有三種主要版本, 全部皆由 v5.0+ Unified Firmware 堆疊支援。

版本	主要硬體	FCC ID (WiFi lookup)	Part Number (Finder)
Opta Lite	Ethernet	N/A	8A.04.9.024.8300
Opta RS485	Ethernet + RS485	N/A	8A.04.9.024.8310
Opta WiFi	Ethernet + WiFi + BT	2A97G-3A001	8A.04.9.024.8320

[!TIP] **WiFi 辨識**: 查看 USB-C 連接埠附近的 FCC ID, 以確認工業機型是否具備無線功能。

1.1 韌體型號感知 (OTP, v5.9.244+ / WI-151)

韌體開機會讀 OTP `OptaBoardInfo._board_functionalities.wifi` bit 自動分辨機型 (讀 OTP, 不碰 WiFi 硬體, 可靠):

偵測結果	機型	對應差異
wifi=1	Opta WiFi (8320)	完整 WiFi/BLE/Ethernet, AP 設定模式可用, 心跳用藍燈
wifi=0	Opta RS485 (8310)	僅 Ethernet + RS485, 無 WiFi/BLE; 網頁過濾掉 WiFi 設定、心跳改用琥珀燈、進不了 AP 模式

- 偵測結果曝光於 `GET /api/system` 與 `GET /api/license` 的 `hasWifi` (true/false) 與 `boardModel` (`Opta WiFi (8320)` / `Opta RS485 (8310)`) 欄位, 供前端做型號感知過濾。
- 開機 serial log 印 `[BOARD] OTP wifi=.. rs485=.. eth=..`。讀不到 OTP 時保守假設 `hasWifi=true` (行為同舊版)。
- ⚠ 不可用 `WiFi.status()` 判型號: 8310 實測回 `0` (WL_IDLE) 而非 `255` (WL_NO_MODULE), 與閒置的 8320 無法區分。

02

2. 輸入／輸出特性

2.1 共用輸入腳位 (I1-I8 / A0-A7)

這 8 個端子 (I1-I8) 可同時作為數位與類比輸入使用。

- 數位邏輯:** 0-24V DC 邏輯準位。HIGH 為 >6.6V, LOW 為 <4.46V。
- 類比範圍:** 0-10V DC 量測 (ADC)。
- 輸入阻抗:** 約 9kΩ (工業 sinking)。
- ⚠ **重要實作注意事項:** 由於 STM32H747XI 的線路佈局, 在這些腳位上呼叫標準的 `digitalRead()` 永遠回傳 LOW。工業 DI 訊號必須透過 ADC 讀取: `analogRead(pin) > 512` (假設為 10-bit 解析度)。

2.2 數位輸出 (DO1-DO4)

- 4 路繼電器或固態輸出 (依機型而定)。
- 透過 `digitalWrite()` 或閘道器的 rule engine 控制。

3. QSPI Flash 與 LittleFS 儲存

Arduino Opta 配備 **16MB 外部 QSPI Flash**，用於存放設定、網頁資產 (SPA) 與生產統計資料。

3.1 分割區配置（標準對應）

Address	Partition	Size	用途
0x000000	P1	1 MB	WiFi Radio Firmware (Murata 1DX)
0x100000	P2	5 MB	OTA 更新緩衝區
0x600000	P3	1 MB	Mbed KVStore (系統設定)
0x700000	P4	7 MB	使用者資料 (LittleFS): /fs/config.json、/fs/stats.bin、/web/index.html

3.2 QSPI 空白 MBR 自我修復 (v5.9.241+ / 8310 安裝根治)

[!IMPORTANT] **全新 8310「安裝一直有問題」的真兇已根治 (v5.9.241)**。全新出廠 Opta 8310 (無 WiFi) 的 QSPI 是**空白 MBR** → `MBRBlockDevice::init()` 回 **-3101**

(**BD_ERROR_INVALID_MBR**)。舊韌體的自動建分割區救援**只判 -3102** (MBR 有效但缺 partition 4)，沒涵蓋 **-3101** → 檔案系統不掛載 → config/token/網路設定永遠存不了 (serial 狂洗 `[LocalConfig] Cannot save: filesystem not mounted`)。

修法 (src/LocalConfig.h)：救援條件改為 `err == -3102 || err == -3101`。任何空白 QSPI 設備第一次燒入本韌體就自我建分割區 + 格式化 (P1 WiFi 1MB / P2 OTA 6MB / P3 KVStore 1MB / P4 user 7MB，與官方 QSPIFormat 一致)。

相容 8320：救援只在 `userPartition->init()` 失敗時才進入；已 provision 的設備 `init()` 回 0、整段跳過、既有分割區/資料不動。

因此 **v5.9.241+ 起，全新 8310/8320 都可直接燒韌體，不再強制依賴下方 QSPIFormat 手續** (8320 仍需經官方 WiFiFirmwareUpdater 寫入 WiFi 韌體與憑證至 P1 才能用 WiFi)。

3.3 必要的初始化 (QSPIFormat)

部署前，必須使用 **QSPIFormat** sketch (取自 `STM32H747_System` 範例) 為 flash 建立分割區。

1. 透過 Arduino IDE (Core v4.5.0+) 上傳 **QSPIFormat**。
2. 以 115200 baud 開啟 Serial Monitor。
3. **確認建立分割區 (Y)**：建立 7MB 的使用者資料分割區。
4. **確認完整抹除 (Y)**：讓硬體準備好建立索引。
5. **還原 WiFi Firmware (Y)**：填入 Partition 1 的關鍵步驟。
6. **選擇 LittleFS (Y)**：將 Partition 4 格式化供閘道器使用。

04

4. 安裝與設定流程

4.1 初次部署

1. 完成 **QSPIFormat** (第 3.2 節)。

2. 若缺少 Radio firmware，請燒入 **WiFiFirmwareUpdater** (IDE WiFi 函式庫)。
3. 透過 PlatformIO 部署專案 firmware：`pio run -t upload`。
4. 透過 Python 部署 Web Dashboard：`python3 scripts/upload_web.py --ip <IP>`。

4.2 拓荒包 (Pioneer) USB 首燒流程 (WI-149，新設備推薦)

新設備 (或救磚) 建議 USB 首燒精簡的**拓荒包 (Bootstrap / Pioneer)** `.bin`，它自包單一網頁 (不依賴 QSPI 資產)，開機即可設定並一鍵升級成完整版：

1. USB 燒 `mes-gateway-bootstrap.bin` (DFU；指令見 `../guides/guide-cli-opta.md`)。
2. 開機 → 瀏覽器開設備 IP → **拓荒設定頁**：顯示 UID (可複製)、網路設定、大大的「上傳 .mesb」鈕。
3. 設好網路 (持久化生效) → 上傳官方 `.mesb` (韌體 + 6 個網頁資產一包)。
4. 設備自動燒錄並重開 → 變成**完整版** (韌體 + 完整 UI)。閉環 bootstrap → .mesb → 完整版已實機驗證。

型號感知過濾 (WI-151)：8310 (無 WiFi) 開拓荒頁時自動隱藏 WiFi SSID/密碼欄，並隱藏整個介面模式選單 (強制乙太網路，只留 Ethernet IP 的 DHCP/靜態設定)。8320 則顯示完整 WiFi 設定。

4.2.1 拓荒包是「無版號、固定產物」 (為什麼、怎麼編)

拓荒包**刻意不帶版號**，檔名永遠是 `mes-gateway-bootstrap.bin`、雲端下載連結永不變：

- 它只負責 onboarding (顯 UID / 設網路 / 上傳 .mesb)，**完整版的版本由 .mesb 自己攜帶**，拓荒包本身不顯版本。
- 因此**極少更新** —— 可能一兩百個完整版迭代都不用碰它。**只有「bootstrap 關鍵環節」出 bug 才需要重出**，例如：QSPI 空白 MBR provisioning (§ 3.2)、OTP 型號偵測、onboarding 流程或網路設定頁。一般功能 / 規則 / UI 改版，走 `.mesb` OTA 即可，與拓荒包無關。
- **編譯方式 (需要時才呼叫)**：

```
bash scripts/build_bootstrap.sh
```

```
# → 產出 release/mes-gateway-bootstrap.bin
```

腳本內部編 `env:opta_pioneer`，事後自動 `git checkout src/WebPage.h` 還原完整版 (拓荒首頁不入庫)。韌體內仍有一個診斷用內部版本號 (不對外顯示)。

4.3 配置設定 (AP Mode , 8320 only)

若不存在任何設定，或在開機時透過前面板的 **USER button** 強制進入：

- 1. 連線至 SSID: **MES-Gateway-Setup** (密碼: **12345678**)。
- 2. 瀏覽至 **http://192.168.3.1**。
- 3. 設定 WiFi/Ethernet 後重新開機。

[!WARNING] **User button 開機長按分段是「型號感知」的** (`g_hasWifi` ,v5.9.249+)：

機型	0-5s	5-10s	10s+
8320 (有 WiFi) 三段	正常開機	AP 設定模式	原廠重置 (清設定)
8310 (無 WiFi) 兩段	正常開機	原廠重置 (清設定)	同左 (5s+ 即重置，無 AP 段)

AP 模式需要 WiFi、只有 8320 有。 8310 無 WiFi 模組 (`WiFi.beginAP()` 不可用)，故跳過 AP 段、**5s+ 直接原廠重置**；8310 要重設網路請用拓荒頁或 Ethernet 設定。⚠ button 原廠重置**只清設定、不清授權憑證** (屬下節 § 4.4 的 L1；交機請改用 L2)。

4.4 把 OPTA 恢復到全新出廠狀態（交機 / 轉售 / 重新部署）

客戶最常問：「我想把這台 OPTA 變回全新的，怎麼做？」依「要清得多乾淨」分三級。 **關鍵觀念：** USB 燒韌體 (DFU) 只覆蓋**韌體**，**不會清 QSPI** (設定、授權憑證都在 QSPI)。要真的全新，必須**另外清 QSPI**。

級別	方法	是否 CLI	清除範圍	適用情境
L1 設定重置 (現場、免電腦)	開機長按 USER button (8320:10s+ / 8310:5s+, 燈快閃後放開)	✗ 實體 鈕	只清 <code>config.json</code> (網路/規則/IO 設定)。⚠️ 不清授權憑證	現場改設定重來、忘記 IP
L2 完整出廠 (網頁一鍵, 推薦交機用)	設定頁按「恢復原廠」, 或 <code>POST</code> <code>/api/system/factory-reset</code> (Bearer <code>smmsadmin</code>)	✗ 網頁/HTTP	<code>localConfig.format()</code> 格式化整個 LittleFS 設定分割區 (P4) → 設定 + 授權憑證 全清	交機、轉售、徹底乾淨
L3 連韌體一起重來 (救磚 / 換乾淨 onboarding)	DFU 燒 <code>mes-gateway-bootstrap.bin</code> (§ 4.2) + 接著做 L2	⚠️ USB DFU	韌體換成拓荒包 + QSPI 設定全清	韌體疑損壞、要回到「剛出廠 + onboarding」
L4 全 QSPI 重格 (最徹底)	官方 <code>QSPIFormat</code> sketch (§ 3.3) : Arduino IDE 上傳 → Serial 115200 → 依序 <code>Y</code> 確認 <code>partition / erase / WiFi firmware / LittleFS</code>	✅ CLI/IDE	整顆 QSPI 重格 (P1 WiFi 韌體 + P2/P3/P4 全部) → 出廠空白	換代、QSPI 疑損壞、要連 WiFi 韌體一起重建

業主問「有沒有 CLI 清 QSPI 的方法」→ 就是 L4 (QSPIFormat sketch, § 3.3)。但自 v5.9.241+ 的空白 MBR 自我修復 (§ 3.2) 後, 連整顆空白 QSPI 都會開機自我 provision, **L4 幾乎只在 QSPI 實體疑損壞時才需要**; 日常「恢復全新」用 L2 (網頁、免 CLI) 即可。

為什麼交機一定要用 L2 (不是按鈕 L1) : 按鈕原廠重置 (8320 10s+ / 8310 5s+) 只刪設定檔, 授權簽章憑證 (`/cfg/license.tok`) 會殘留 → 重開機後韌體會「從有效憑證還原 ACTIVE」, 等於授權沒清掉就交出去。L2 的 `factory-reset` 會 `format()` 整個分割區、連憑證一起抹掉, 才是真正的全新。

L2 操作 (指令版) :

```
# 對設備本機 IP 直接打 (需設備管理 token, 預設 smmsadmin)
curl -X POST http://<device-ip>/api/system/factory-reset -H "Authorization: Bearer
# 設備回 {"success":true,...} 後自動重開 → 開機進 onboarding (無設定、授權 UNKNOWN)
```

驗證已回到全新: 重開後 `GET /api/license` 應為 `UNKNOWN`、`GET /api/config` 無自訂設定、首頁進設定/拓荒流程即代表乾淨。

05

5. 部署疑難排解

5.1 LED 狀態指示燈

USER LED 心跳 (型號感知, WI-152):

- **8320 (有 WiFi):** 藍燈 (LED_USER = LEDB = PE_5) 慢閃 = 系統心跳。
- **8310 (無 WiFi):** 藍燈是 **BLE 模組燈**, 官方明文「僅 WiFi 版能亮」, 8310 點不亮。故 8310 心跳改用**琥珀色** (紅 LEDR + 綠 LEDG 同亮), 與純紅 (故障)、純綠 (DFU/reset) 明確區隔。
- 心跳閃速: MQTT 已啟用但未連上 broker → 100ms 快閃; 正常 → 500ms 慢閃。

其他:

- **恆亮綠燈:** Bootloader/DFU 模式啟用中 (連按兩下 Reset 進入)。
- **快速紅燈閃爍:** 致命當機 (Mbed OS Panic)。
- **緩慢紅燈閃爍:** Radio 模組初始化失敗 (缺少 Firmware 或匯流排競用)。

5.2 常見修正

- **「Failed to mount filesystem for WiFi firmware」:** 重跑第 3.2 節 (QSPIFormat), 並確認 WiFi firmware 還原達到 100%。
- **系統在 WiFi 啟動時卡住:** 確認在 `WiFi.beginAP()` 之前有呼叫 `localConfig.unmount()` (已於 v5.4.24+ 修正)。
- **網頁上傳期間出現 Broken Pipe:** 上傳程式現已改用 `curl`。請確認部署用的機器已安裝 `curl`。

5.3 USB CDC 阻塞陷阱 (Mbed OS)

Arduino Opta (Mbed OS) 使用原生的 USB CDC 實作來進行序列除錯。

- **問題:**若主機 (PC) 開著作用中的 Serial Monitor，而 USB 線被實體拔除，Mbed OS 的序列緩衝區可能會被阻塞。主迴圈內的 `Serial.print` 呼叫可能會停滯，直到逾時為止。
- **症狀:**I/O 規則無反應，或主迴圈看似「凍結」。
- **解法:**在拔除 USB 線之前，**務必先關閉 PC 上的 Serial Monitor**，以確保 firmware 在外部供電下能以確定性的時序持續運作。

06

6. 周邊整合範例

6.1 PATLITE NE-M1ATB-M（七色積層信號燈）

- **Opta 接線:**DO1 (Red)、DO2 (Green)、DO3 (Blue)、DO4 (Buzzer)。
- **運作:**24V DC 控制。
- **Setup:** 在 Rule Engine 手動建立 Action Groups (紅燈/綠燈/藍燈各控制 DO1-DO3、蜂鳴器用 DO4 pulse 模式 500ms)。v5.9.112 之後不再提供一鍵套用模版。

6.2 校正引導燈號（三色燈 / 任意兩路 DO）

- 號誌燈與磅秤是**各自獨立的設備**，由 Rule Engine 的 CH_DO 動作串起來；沒有燈也能完成校正。
- **燈號沒有固定顏色** —— 韌體只控制「第幾顆燈亮/滅」，顏色由現場接線決定。
- **DO 編碼:** `index = (expIdx << 4) | ch`，`expIdx` 0=本機、1=第一台擴充；`ch` 從 0 起算（面板標示 DO1 → `ch=0` → index 0）。第一台擴充的 ch2/ch3 → index 18 / 19。
- **燈號對照:** state=0 全滅、state=1 第一顆燈亮（放空籃）、state=2 第二顆燈亮（完成）。
- Action 用 CH_DO output (`outputMode` 0=ON / 4=OFF) 依步驟切換；**該滅的要明寫 OFF**，否則上一顆不會熄。
-  接線後請在 I/O 頁逐路實測，記錄哪一路點亮哪顆燈，再據以填 Action。

-  完整設定與操作: [../guides/guide-calibration-wizard.md](#) § 7 號誌燈搭配 (接線、索引、Action、驗收)。