

備援與韌性

17

最後更新：2026-05-26 |  對應韌體：v5.9.130+ |  負責人：KC

本手冊整理 Opta (MES Gateway) 的四項備援機制、各自的驗證結果，以及 MQTT subscribe 失效的外部 watchdog 解法與部署方式。

01

1. 備援機制總覽

#	機制	規則	狀態 (2026-05-26 實測)
A	NVS 設定持久化	LocalConfig save/load，跨重開不丟	✅ PASS
B	Config Server 雙層 fallback	單一心跳，地端優先 → 失敗 fallback 官方 cloud	✅ PASS
C	MQTT 斷線重連	broker 斷線無 auto-disable，持續 retry	⚠️ 連線復原但 subscribe 失效 → 外部 watchdog 解
D	MQTT broker 空 fallback	broker 設定空 → 連官方 <code>mosquitto.smms.com.tw</code>	✅ 韌體已實作

02

2. A — NVS 設定持久化

所有設定 (signals / actions / rules / tcpio / mqtt / configserver) 存於 QSPI Flash，由 `LocalConfig` 序列化。

驗證：軟重開 (`POST /api/system/reboot`) 後 uptime 657s → 35s，所有設定逐項比對完全一致。重開機、斷電重插皆保留。

03

3. B — Config Server 雙層 Fallback

層級	URL	職責
官方 Server	韌體寫死 <code>https://opta.smms.com.tw</code>	授權 + OTA + 備份還原
地端 Server	使用者設定 (可空)	備份還原

規則：維護**單一心跳**，目的地優先地端，地端不存在/失敗時 fallback 官方。授權與 OTA **只走官方**。

驗證：地端 URL 設假 (`192.168.99.99:9999`) + 官方留 `opta.smms.com.tw` → 心跳 60s 後 `cloud result=OK`，local 不嘗試 → fallback 正確。

設定 API： `GET/POST /api/configserver` (見 `../api/api-config-server.md`)。

04

4. C — MQTT Subscribe 失效 (Known Issue + Watchdog 解法)

4.1 問題

broker 重啟 / 網路抖動後：

- `mqttClient.connected()` 回 `true`，gateway **publish 正常** (遙測還在發)
- 但 **subscribe 失效**: 收不到任何 inbound (連自己 publish 給自己訂的 topic 的 echo 也收不到, **對稱性全死**)
- 結果: MQTT 命令 / wizard chain 全失靈, **直到設備 reboot 才復原**

4.2 已嘗試的 in-device 修法 (皆失敗)

嘗試	結果
<code>connected()</code> false→true 邊緣偵測 → resubscribe	✗
每 30s 無條件 refresh subscribe	✗
healthcheck → disconnect 強制重連	✗
被動 <code>lastInboundMs > 120s</code> → reboot	✗ 沒觸發
主動 self-ping > 2 次無 echo → reboot	✗ 沒觸發

根因疑為 MQTT lib (mbed/pubsubclient) 連線狀態機在 broker 重啟後的內部狀態, 無 serial debug 無法定位。in-device 自我診斷已證實不可靠。**結論: 設備無法可靠自我偵測此狀態, 需外部探測。**

4.3 解法: 外部 Watchdog (✅ 已驗證)

從設備**外部**定期測 round-trip, 連續失敗就打 reboot。腳本: `scripts/opta_watchdog.py` (部署位置見 § 4.5)。

原理:

1. publish nonce 到 `.../cmd/signal/5` (未使用的 MQTT 輸入 index)
2. 輪詢 `GET /api/signals`, 看 `wdprobe` signal 的 `currentValue` 是否反映
3. 連續 N 次 (預設 3) 沒反映 = subscribe 死 → `POST /api/system/reboot`
4. 等設備 + MQTT 復原, 重置計數

為什麼外部有效：它測的是**真正壞掉的 external→device 路徑**，不依賴設備自我診斷。HTTP 不通時不計入失敗（避開設備重開中誤判）。

前置：設備上需有 `wdprobe` signal (隨 NVS 持久化)：

```
POST /api/signals
{"index":4,"enabled":true,"name":"wdprobe","trigger":0,"debounceMs":0,
  "sources":[{"type":6,"index":5,"op":4,"threshold":0.5,"and":false,"expIndex":0,"
```

(slot4、CH_MQTT idx5、TRIG_ALWAYS、op≥0.5；不綁任何 rule，純供探測讀取。)

4.4 驗證紀錄 (2026-05-26)

```
18:28:44  探測失敗 1/3    ← 停 broker 25s + 重啟後 subscribe 死
18:28:57  探測失敗 2/3
18:29:10  探測失敗 3/3
18:29:10  → POST /api/system/reboot
18:30:07  設備已復原 (~50s)
之後：adv1 → rules/2/trigger 校正標零 → wizard chain 自動回來 ✓
```

4.5 部署

腳本已收錄於 `scripts/opta_watchdog.py`。

```
# 1. 確認 wdprobe signal 已建在設備上 (隨 NVS 持久化, 一次即可, 見 §4.3)

# 2. 常駐 (擇一):
#   a) 前景測試:
python3 scripts/opta_watchdog.py --interval 30 --fails 3
#   b) 部署到 always-on 主機 + macOS launchd / Linux systemd 常駐 (建議)
#       例: cp scripts/opta_watchdog.py /opt/opta/ 後設 systemd unit
```

參數建議(prod): `--interval 30 --fails 3` → 約 90s+ 持續失敗才 reboot, 濾掉瞬斷。

腳本內需設定: `IP`、`PRE` (UID 前綴)、`BROKER` (host/port/user/pass)。

注意: watchdog 應跑在**獨立 always-on 主機** (非設備本身), 且能同時連到 broker 與設備 HTTP。

05

5. D — MQTT Broker 空 Fallback

韌體已實作 (`main.cpp` MQTT init + `ApiHandlers_Core.cpp` config POST):

```
broker 設定非空 → 連地端 broker  
broker 設定空   → 連官方 mosquitto.smms.com.tw (port 依 useTls 8883/1883)
```

官方 broker reachable (DNS 210.241.233.133, 1883/8883 皆通)。實測 broker 清空後連不上, 原因為**官方 broker 需要對應 credentials (user/pass)** —— 這是部署端設定問題, 非韌體 bug。若要真正啟用官方 fallback, 需設備帶官方 broker 的帳密 (可由 license 服務派發)。

06

6. 故障排除速查

症狀	可能原因	處理
MQTT 命令沒反應、wizard chain 不動，但設備 web 正常	broker 重啟後 subscribe 失效 (C)	<code>POST /api/system/reboot</code> ；長期靠 watchdog 自動處理
設定重開後消失	(不應發生，A 已驗證)	檢查 QSPI 掛載 <code>qspiMounted</code>
心跳一直失敗	地端 URL 不通且官方需 credentials	檢查 configserver URL / token
「授權檢查失敗」modal 偶現	設備忙時 license API 逾時	v5.9.130+ 已放寬 timeout + auto-retry，通常自動消失

07

7. 相關文件

- 工作流引擎 / API: `../specs/spec-mqtt-chain-workflow.md`
- Config Server: `../api/api-config-server.md`
- MQTTS / broker: `spec-connectivity.md`、`mqtt-broker` SOP
- 單執行緒 HTTP 競爭 (C 根因背景): Opta 韌體 HTTP 處理為單執行緒，當輪詢與 inbound 請求並發時會互相競爭，這是情境 C (MQTT 重連失效) 的根因背景。