04 開發者 / 整合

觸發式定期回報

📅 最後更新：2026-06-16 | 📌 負責人：KC

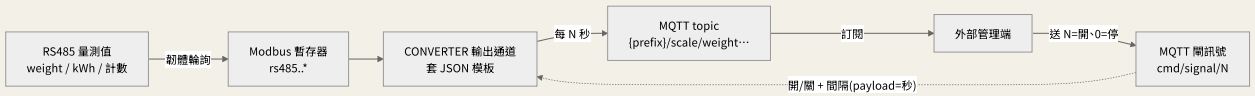
本文是「磅秤發重量 / 用電量發佈 / SF965 計米回報」這類需求背後**共用的通用機制**的權威說明。各裝置的可重現 recipe 見 §6 的實例連結;本文講「**精神、最小骨架、怎麼在 Web 介面做、怎麼變成工作流程頁一鍵卡片**」,看完任何 RS485 量測值都能自己兜出「外部下命令 → 設備每 N 秒回報」。

01

1. 精神：不為每種設備寫特製韌體

過去「磅秤發重量」一度想做成磅秤專屬韌體(WI-127),後來**全面廢除特製化**(WI-130),改成 **重用現有的通用機制**。核心只有三塊既有積木:

1. **RS485 來源** — 設備已在輪詢的 Modbus 暫存器(重量 / 用電量 / 計數值…)。
2. **CONVERTER 輸出通道** — 把暫存器值套進 JSON 模板,週期發到 MQTT topic。
3. **MQTT 閘訊號** — 外部發 `{prefix}/cmd/signal/{N}` 控制「發 / 停」與「多久發一次」。



✅ **沒有任何設備專屬韌體**。換一台新設備 = 換 Modbus 暫存器 map + 改 JSON 模板,流程骨架不變。

02

2. 最小骨架（純 level 閘）：1 個訊號 + 1 個通道

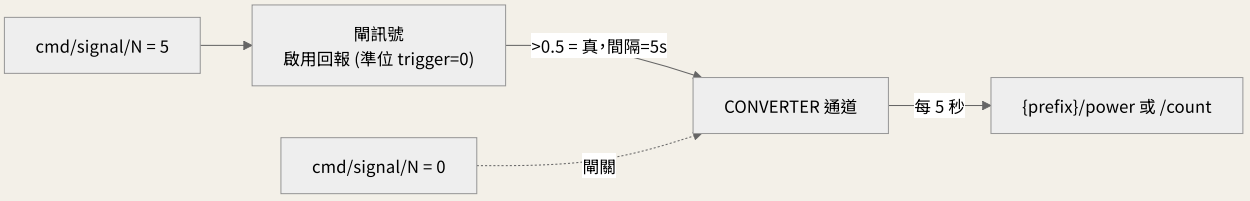
最單純的「按命令週期上報」只需兩個物件(用電量 / 計米器都走這個):

層	物件	設定
訊號	啟用回報(閘)	來源 `type=6(CH_MQTT) index=N`, `op=4 threshold=0.5`, `trigger=0`(`TRIG_ALWAYS`／**準位**)。讀 `{prefix}/cmd/signal/N`,>0.5 為真。
通道	週期回報 (CONVERTER)	`mode=2 direction=1`, `converterGateSignalId=<上面的訊號>`, `converterIntervalFromGate=true`(WI-143),JSON 模板取暫存器值。

> ⚠️ **閘訊號務必用準位** `trigger=0`(`TRIG_ALWAYS`),不要用 `CHANGE`(3)/ `RISING`(1)。準位才代表「持續開關」語意,也才會在工作流程頁自動呈現成 on/off 卡片(見 §5;實測:`CHANGE` 觸發的閘不會合成卡片)。

操作(payload = 間隔秒數,WI-143):

```
{prefix}/cmd/signal/N = 5   → 閘開，每 5 秒發一次
{prefix}/cmd/signal/N = 2   → 即時改成每 2 秒
{prefix}/cmd/signal/N = 0   → 閘關，停止
```



`converterIntervalFromGate=true` 時,**發佈間隔直接取閘訊號的 MQTT payload**(秒);設 `false` 則用固定的 `converterIntervalSec`。**送 `0` 一律停**。⚠️ 這些 `cmd/signal/*` **千萬別用 retain(`-r`)** —— retained 會在設備重連時反覆灌回、誤觸發。

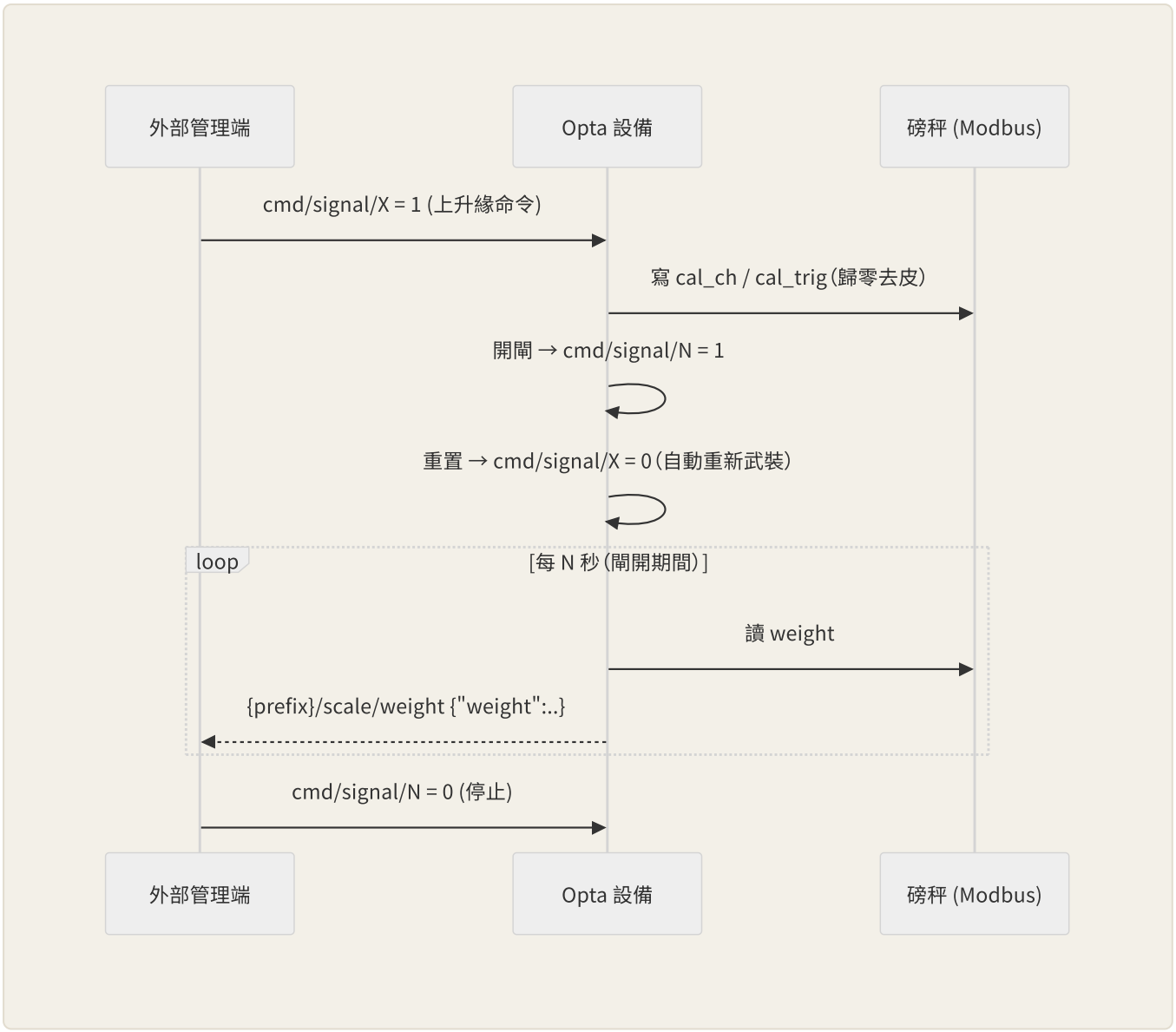
📌 **發佈「時機」三種模式**(並存,擇一):

模式	設定	行為
固定間隔	`converterIntervalSec`(網頁對話框預設)	每 N 秒固定發,例每 5 秒
payload=秒數(本文示範)	`converterIntervalFromGate=true`(WI-143)	送到閘的數字=幾秒發一次、送 `0` 停
變更時才發	`converterOnChange=true`(v5.9.263)	值有變才發、沒變不重複;間隔填 0=純變更

03

3. 進階骨架（命令 + 校正）：磅秤那種

當「上報前要先做一件事」(例如磅秤要先歸零去皮),就在最小骨架前面加一段 **上升緣命令 → 規則 → 動作**:



差別只是「閘」之前多了 edge-trigger 命令 + rules/actions + Modbus 寫入。**純上報需求不需要這段。**命令訊號的值會「黏著」,所以動作要回送 `0` 重新武裝 —— 因此送 `1` 可重複、不必手動先歸 0。

04

4. 在 Web 介面怎麼做（免命令列）

不想打 API 也行,整套都能在 Web Dashboard 點出來。兩步:

4.1 建「閘訊號」(規則頁)

到 **規則** 頁新增一個訊號:來源型別選 **MQTT 訊號**、index = `N`、條件 `> 0.5`、觸發模式 **準位 (ALWAYS／`trigger=0`)**。這就是「發/停」的閘 —— 用準位才會在工作流程頁變成 on/off 卡片(§5)。

規則頁 — 建立訊號 / 動作 / 規則

4.2 建「CONVERTER 輸出通道」(配置頁 → TCP I/O 通道)

到 **配置 → TCP I/O 通道** 新增一個通道:

- **方向 = 輸出、模式 = 轉換(CONVERTER)**。
- **MQTT 主題** = 你要回報的 topic(如 `…/scale/weight`、`…/power`、`…/count`)。
- **閘訊號** = 選 4.1 建的那個訊號。
- 勾 **「間隔取自閘 payload」**(`converterIntervalFromGate`)→ 送進閘的數字就是發佈秒數。
- **轉換模板** 填 JSON,用 `${rs485.<SlaveID>.<欄位>}` 取暫存器值,例 `{"weight":${rs485.<SlaveID>.weight}}`。

下圖為實機 OPTA-KC 的「電流監視」CONVERTER 通道:推送間隔 5 秒、門控信號=「啟用用電量發佈」、JSON 模板取 Finder 9 欄電力值:

配置頁 — CONVERTER 通道編輯(間隔 / 門控信號 / JSON 模板)

> 建好後不必重開機,送 `{prefix}/cmd/signal/N = 秒數` 即開始週期上報。

05

5. 工作流程頁：把這套流程變成「一鍵白話卡片」

建好「準位(`trigger=0`)閘訊號」後,**工作流程**頁(WI-128/129)會**自動**把它合成一張 **開關型卡片**(WI-130:無需規則,只要閘是準位觸發)—— 現場操作員不必懂訊號/規則/動作、也不必打任何 MQTT 命令,**直接撥 on/off 開關**就能開始 / 停止週期回報。

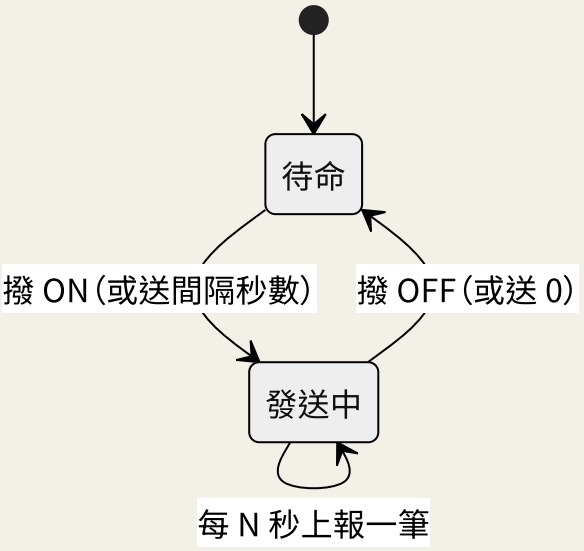
下圖為實機:一顆準位閘訊號即自動成為一張 on/off 卡片(「待命 / 已停止」):

工作流程頁 — 觸發式回報的開關型卡片

> ⚠️ **只有準位(`TRIG_ALWAYS`/0)的閘**會合成 on/off 卡片;若閘建成 `CHANGE`(3)/`RISING`(1),工作流程頁不會出現卡片(實測 confirmed)。要呈現成卡片,務必把閘訊號的觸發模式設為**準位**。

新流程也能直接從工作流程頁用**「＋ 新增工作流程」四步精靈**(命名 → 何時啟動 → 要做什麼 → 確認)建立:

工作流程頁 — 新增工作流程四步精靈



卡片角色	對應本文機制	操作
開關型(待命 / 發送中)	§2 的閘訊號 + CONVERTER 通道	撥 ON = 開始每 N 秒發、撥 OFF = 停
觸發型(▶ 執行,可填參數)	§3 的上升緣命令(如磅秤「需校正」)	按一下送一次命令

- 切 **ON** → 徽章「🟢 發送中」,下游每 N 秒送一筆;切 **OFF** → 「⚪ 待命」停止。
- 狀態由設備**即時回報**,多人 / 多裝置看到的開關狀態一致(不是只反映你按的那下)。
- 這取代了舊版「要記 topic、用 MQTT 工具手動發 `1` / `0` 」的做法。

> 📖 工作流程頁四種角色卡(觸發 / 開關 / 武裝監控 / 純被動)的完整操作見 **操作手冊 §10.5 工作流程頁操作**。

06

6. 裝置實例（可重現 recipe）

裝置	量測值	骨架	Recipe
TDA-08B 磅秤	weight / stable	進階(§3,含校正/去皮、上升緣命令)	見本文 §3 進階骨架
Finder 6M.TB 電力分析儀	9 欄電力 JSON	最小(§2,純 level 閘)	見本文 §2 最小骨架
SF965 計米器	當前計數值(米數)	最小(§2,純 level 閘)	套電力分析儀那套,換暫存器 map(見下)

SF965 計米器（套最小骨架）

SF965 沒有專屬流程,**完全比照 Finder 電力的 §2 最小骨架**,只換暫存器與模板:

- RS485 來源:slave `3`、`9600/8N1`、machineID `78`(`identifyDevice(78)` → SF965)。
- 暫存器:**當前計數值** `0x0000~0x0001`(40001~40002) `INT32` 唯讀。
  - ⚠️ 32 位元跨兩暫存器,讀出不對就換 Endianness:Modbus 同一 INT32 在不同設備可能用不同 byte / word 排列,需嘗試 byte swap 或 word swap 直到讀值正確。SF965 計數 raw 值另需乘 scale `0.00390625`(÷256)。
- 通道 JSON 模板(假設 count 對應 `rs485.<SlaveID>.count`):

```
{"count":${rs485.<SlaveID>.count}}
```

- 操作同 §2:`{prefix}/cmd/signal/N = 5` → 每 5 秒發 `{prefix}/count`;送 `0` 停。

07

7. 不必先知道 UID：whoami / info 探索（WI-131）

管理端要對全廠設備下命令、又不想預先記每台 UID:發**免前綴廣播** `mes/gateway/whoami` (或對單台發 `{prefix}/cmd/info`)→ 設備回 `{prefix}/info` (uid/name/fw/ip/online/counts)

- 每通道 `{prefix}/info/io/<id>` (含該通道的控制 / 資料 topic)。

```bash
H=mosquitto.smms.com.tw; P=8883; U=smms; PW=2ojujiru
mosquitto_sub --insecure -h $H -p $P -u $U -P $PW -t "mes/gateway/+/info" -t "mes
mosquitto_pub  --insecure -h $H -p $P -u $U -P $PW -t "mes/gateway/whoami" -m 1
```

> 對外整合合約(給第三方現場管理端)見 `../specs/spec-mqtt-integration.md` 。

08

8. 共同的坑（建流程必看）

1. **POST signal / action 一定要帶** `"enabled":true` —— 否則 API 預設 `false` ，訊號不評估、動作不執行，整條流程靜默不動(tcpio / rules 的 POST 本來就有帶)。
2. **compact JSON**:POST body 不能有空格(`separators=(',',':')`)，韌體用 `indexOf` 比對 `"key":value` 。
3. `cmd/signal/*` **絕不要 retain**;不慎發了用 `-r -n` 清除。
4. **寫入留 5s + 重試**:Opta 單執行緒，flash 存檔(WiFi-safe unmount/mount)會阻塞 6-12s，連續快速 POST 會 ConnectionReset。
5. **CH_TCP(output type 7)的 index 存 channelId**(WI-129-0a)，不是陣列位置;規則 GET 回讀欄位是單數 `signalId` / `actionId` 。
6. **payload = 間隔秒數**(WI-143， `converterIntervalFromGate=true`):送 N 每 N 秒、送 0 停。topic 用設備 **UID 前綴**(非 topicPrefix)。