04 開發者 / 整合

MQTT 原始量測時間

適用版本：6.0.122 起；126 交付候選已重新驗證真實 Finder→MQTT 路徑。這份指南說明資料裡的 `timestamp` 是什麼，以及怎麼確認它有沒有填對。

假設設備在 **08:20 量到 3 公斤**，但網路慢了，業主在 **08:25 才收到**。這筆的量測時間仍應記 **08:20**，不能變成 08:25。`timestamp` 就是用來保存這個時間的欄位。

126 線上校正時，同一重量 payload 另附 `calibration` 狀態與 ACK；量測時間仍使用原 `timestamp`，不另訂閱必要狀態 topic。操作見 線上多段校正。

01

先看懂一筆資料

磅秤會送出類似下面的資料：

```json
{"weight":3.00,"stable":1,"timestamp":"2026-10-01T08:20:00Z"}
```

欄位	白話意思	你要確認什麼
`weight`	這次讀到的重量	與磅秤及設定的小數位一致
`stable`	磅秤回報的穩定狀態	沿用原來的欄位；它不保證與重量同時讀取
`timestamp`	OPTA 成功讀到這次重量的時間	是那次量測的時間，不是發送或收到的時間

時間格式是 UTC、ISO 8601，最後的 `Z` 表示 UTC。台灣時間要加 8 小時；例如 `08:20:00Z` 是台灣下午 4:20。接收程式先依 UTC 解析，再決定畫面顯示哪個時區。

如果設備還不知道正確時間，會送：

```JSON
{"weight":3.00,"stable":1,"timestamp":null}
```

`null` 的意思是「這筆的量測時間未知」，不是零點，也不是請接收端改填現在時間。

02

現場以網頁核對時間戳

1. 在 OPTA 左側點「配置」→「TCP 通道」，編輯本輪重量發送通道（標準名稱「磅秤發重量-直接」）。確認協定、topic 與設備正確；先記下原內容。
2. 查看「Converter 設定」的資料範本，找到重量及對應 timestamp。站號與變數名須沿用現場值，例如：

   ```TEXT
   {"weight":${rs485.3.實時重量},"stable":${rs485.3.穩定狀態},"timestamp":${rs485.3.實時重量.timestamp}}
   ```

3. 按「👁 預覽填充結果」。應看到已填入的 JSON：重量與當次讀值一致；已對時時 timestamp 為 UTC ISO8601 字串，未對時為 JSON `null`。變數無法解析、仍顯示 `${...}` 或紅字時，先檢查站號與名稱。
4. **只查看時不必按儲存**，關閉編輯視窗即可。既有範本缺 timestamp 時先由工程人員核對並備份，依下一節補欄位；不要整份貼上範例覆蓋自訂欄位。
5. 在業主 Edge 的接收畫面核對真正收到的 payload，再核對資料庫量測時間與接收時間分開保存。若 Edge 畫面不顯示原始時間，請業主提供資料詳情或入庫紀錄；不能只看 OPTA 預覽就簽接收端通過。

這個預覽可檢查模板與目前讀值，不能證明 MQTT 已送達、校正 ACK 已完成或延遲補送正確。不要求未在預覽中出現的 calibration 狀態必須在此顯示；校正狀態以原重量 topic 真正收到的 payload 為準。正常 on-change 也不應因時間更新就多發；校正連續採樣另依 OC.4。

03

第一步：先確認你是新設定，還是舊設定

你的情況	要做什麼	做完應該看到什麼
在 6.0.122 網頁新建一鍵磅秤／電表流程	使用一鍵流程，查看產生的資料範本	已經有 `timestamp`
原本就有磅秤／電表流程，只升級韌體	用工具先預覽，再補上時間欄位	原欄位與精度保留，只增加對應的時間
自己寫的資料格式或 topic	請工程人員依下方「自訂格式」核對	不會被工具偷偷改成另一種格式

升級韌體不會自動改寫你原來的範本。**不要為了增加一個欄位，就重新套用整套流程覆蓋現場設定。**

舊設定要用的工具：下載 migrate_measurement_timestamps.py。

請打開 現場驗收的逐步命令，依自己的作業系統做下面四個動作。每一步都有 macOS／Linux 與 Windows PowerShell 指令：

1. 下載工具，執行 `--help` 確認可以啟動。
2. 執行「預覽既有標準範本遷移」。它只列出準備改的地方，不會寫入設備。
3. 核對無誤，再執行「套用已核對的遷移」。工具先把完整備份放進你指定的 `--backup-dir` 私有目錄，再修改。
4. 再執行一次預覽。應顯示零個變更，表示沒有重複增加欄位。

不符合時：如果工具報錯、時間來源不同或出現不認識的格式，先停下來，保存結果檔交給工程人員。不要反覆套用。備份可能含登入資料，請留在本機私有目錄。

04

第二步：收到新資料，確認時間跟著數值

1. 依驗收文件的「訂閱新磅秤量測」指令接收新訊息；電表用 `/power` 主題。
2. 讓設備取得新的讀值，保存收到的完整訊息與接收時間。
3. 查看 `timestamp`：有正確時間時是以 `Z` 結尾的字串，尚未校時則為 `null`。

4. 交給業主確認：資料庫的「量測時間」使用 `timestamp`，「接收時間」另外保存。

通過：量測與接收時間分開，沒有因為資料晚到就改寫量測時間。

不符合時：若始終為 `null`，請先檢查設備是否已取得正確時間，再取一筆新讀值。若設備送出的時間正確、資料庫卻存成接收時間，需修正接收程式，不能只改設備畫面。

05

第三步：確認晚收到的資料沒有被換時間

此項由工程人員依 驗收流程 TS.4 操作；需要能保存訂閱的 MQTT 接收端與隔離測試設備。一般現場不要直接停掉正在生產的量測。

1. 先確認訂閱成功，再保存一筆資料。
2. 用測試設備暫停量測來源，讓後續發送仍引用同一筆讀值。
3. 暫時中斷接收端，稍後用同一個訂閱連線身分重新接上。
4. 比對補收到的資料：數值與 `timestamp` 應保持原樣，接收時間則較晚。
5. 恢復量測、清掉測試訂閱，確認新的讀值會帶上新的時間。

通過：晚收到不會變成晚量到。

不符合時：若沒有補收到資料，先確認訂閱是否真的保存、訊息是否排隊；不能把「沒有資料」算成時間正確。若沒有這套測試條件，填「待工程測試」，不要勾通過。

06

給工程人員：自訂格式與資料規則

自訂範本要讓數值與時間來自同一個欄位。例如：

```text
{"weight":${rs485.1.weight},"timestamp":${rs485.1.weight.timestamp}}
```

`1` 要換成實際的磅秤通訊位址（Slave ID），`weight` 要換成設定中的暫存器名稱。數值若有 `:N` 小數位設定，時間欄位不加 `:N`。**時間變數外不要再加雙引號**，因為它本身會產生完整字串

或 `null`。

- 磅秤的時間屬於 `weight` 的成功讀取，不代表 `stable` 同時取得。
- Finder 電表保留 `v/i/p/q/s/pf/hz/thd/kwh` 九個欄位，另加同一次批次讀取的 `timestamp`。
- 自訂暫存器各自保存時間。一個欄位讀取成功，不能替另一個讀取失敗的欄位更新時間。
- 讀取失敗時保留原快取值與時間；來源判定離線後清除快取，並沿用既有停止發送規則。恢復成功讀取後才有新樣本時間。
- 尚未校時取得的舊樣本仍是 `null`，不能在後來校時時替舊資料補上現在時間。開機保存的 `lastKnownTime` 不等於本次已校時。
- 6.0.128 起採毫秒解析度（例如 `2026-10-06T04:03:06.237Z`），不保證與標準時間誤差小於 1 毫秒；這是 OPTA 讀到資料的時間，不是磅秤內部感測器的時間。
- 勾選「值變更才發送」時，只有時間改變不會多發一筆；數值變更才觸發。定期或規則觸發發送仍保留該筆原來的時間。
- 一筆資料若有不同來源，請分別保存各來源的時間。不要拿其中一個時間代表全部欄位。
- 加入欄位後，輸出超過 511 bytes 就不發送，以免送出不完整的 JSON。

07

補送的限制也要說清楚

這次增加的是「原始量測時間」，不是「斷電後每筆資料都能補回」的功能。OPTA 的 Converter 目前直接發送；一般 MQTT 排隊空間有限，也可能合併同 topic 的資料，沒有保證跨重開保存全部未送資料。

補送服務應保存並原樣重送整筆資料，不可重新產生 `timestamp`。接收端可另存 `received_at`，但不能把它當量測時間。舊版到秒，6.0.128 起到毫秒；同一毫秒仍可能有多筆，不能只拿 timestamp 當唯一識別碼。

完整驗收還包含未知時間、讀取失敗、時間更正與「值變更才發送」。這些由工程測試項 TS.3～6 留下各自結果；正常收到三筆資料，不能代替那些故障測試。

08

通用訊息時間（6.0.127 測試候選）

只需要整筆訊息的產生時間時，在 Converter 資料範本使用：

```JSON
{"value":${rs485.1.voltage},"timestamp":${sys.timestamp}}
```

`${sys.timestamp}` 在 Converter 組合這筆訊息時取一次 UTC 時間；同一筆中重複使用會相同。尚未對時為 JSON `null`，外面不要再加引號。預覽會取預覽當下的時間，不表示已發送；它不是 MQTT 接收時間，也不是控制器原始量測時間。

需要同時追溯量測時間時，另外保留來源欄位：

```JSON
{"value":${rs485.1.voltage},"timestamp":${sys.timestamp},"measured_at":${rs485.1.voltage.timestamp}}
```

變數參考將通用訊息時間列為常用項目，逐欄位量測時間收在「進階：量測時間」。既有範本與量測時間語意不自動改寫。「值變更才發送」不因時間單獨變化而重送；新一輪主動組合／發送會產生新訊息時間。已產生的 payload 若由 broker／接收端保存後補送，應原樣保留時間。本功能不新增裝置端持久補送佇列，也不保證斷線期間資料不遺失。

毫秒格式（6.0.128 測試候選）

上述兩種變數語法不變，升級後都輸出三位小數：

```JSON
{"value":114.11,"timestamp":"2026-10-06T04:03:06.237Z","measured_at":"2026-10-06T04:03:05.918Z"}
```

這是格式示例，不是實秤紀錄。Edge 應使用支援 ISO 8601 小數秒的日期解析器，不要固定截取 20 個字元。既有時間欄位會由秒格式變為毫秒格式；尚未填時間變數的範本仍不會自動新增欄位。

NTP 保留回包的小數秒；網頁對時送出秒與 0～999 的 `milliseconds`。只送秒的舊客戶端仍接受，以該整秒作為時間基準。對時後以 64-bit 單調毫秒時鐘推進，避免 32-bit `millis()` 約

49.7 天回繞。重新對時只影響之後的新訊息／新樣本，舊樣本及已組好的 payload 不重寫。

這是毫秒解析度，不是毫秒級 UTC 準確度保證：目前對時未補償網路延遲，也未量測長時漂移。量測時間仍是 OPTA 成功讀取資料的時間，非感測器內部取樣時間。`null`、值變更才發送、補送保存原時間的規則不變。