

控制延遲特性

版本:1.0 | 日期:2026-08-20 | 韌體:6.0.65 | 量測機:OPTA-GW-001(8320 / WiFi)

01

為什麼要有這份文件

業主觀察到「燈號會停滯 1~3 秒」。追查後確認**這不是顯示問題,是控制迴圈的延遲**,所以有必要把它量化成規格,而不是用「應該很快」帶過。

02

架構事實(先釐清一個常見誤解)

這個韌體跑在單一核心(M7)、單一主迴圈,不是「控制一顆核心、網頁另一顆」。每輪迴圈依序跑完:

```
led → net → io → mqtt → modbus → mqpub → scan → mbtcp  
→ expupd → expio → rules → save → web → async → cs → ota
```

- LED 在 **led** 段更新,一輪只更新一次
- 規則評估、DO 輸出也在同一輪

所以任何一段卡住,燈與控制**同時**被延後。燈號停滯是徵兆,不是獨立問題。

(硬體上確實有第二顆 M4 核心,第二個 MB 的 flash 也保留給它,但本韌體未使用;改用它屬重新設計等級,且會動到 flash layout 使既有設備的 OTA 失效。)

實測數據

序列只記錄**超過 200ms** 的迴圈,低於此值不列入(那是正常情況)。

主迴圈週期(= 控制延遲的上界)

指標	值
超過 200ms 的迴圈	約每分鐘 4~5 次
p50(超標者)	668 ms
p90	2088 ms
p99	2879 ms
實測極值	3711 ms

結論:DI 觸發到 DO 動作,典型 < 200 ms,最差 2~3 秒(極值 3.7 秒)。這不是罕見事件 —— 每分鐘都在發生。

卡頓發生在哪一段

段	說明
net	最大宗。WiFi 驅動 + TLS 加密收送
web	服務網頁請求(有人看畫面時才有)
modbus	RS485 輪詢(每 2 秒一次,佔比小)

儀表板的代價(WI-189 前後對照)

同一台設備,期間完全不碰:

	有儀表板	無儀表板	變化
卡頓次數/分鐘	7.0	3.0	-57%
<code>web</code> 卡頓/分鐘	2.31 s	0.61 s	-73%
<code>net</code> 卡頓/分鐘	3.07 s	1.87 s	-39%
最久單次	2879 ms	2092 ms	

WI-189 已讓頁面切到背景時停止 telemetry(韌體本就是 keepalive 制,不續命就自停),所以現在只有「真的在看畫面」時才付這個代價。

04

根本原因

主迴圈裡有阻塞式網路操作,而 TLS 沒有有效的時間上界: `setSocketTimeout` 只管單次 socket 收送,一次握手是多個來回,總時長無界 (WI-183 的黑洞實驗證實 `connect()` 可卡 5 分鐘不返回)。

分執行緒解不了 —— `csThread` 已是獨立執行緒,WI-175 事故正是它在做 TLS 時 把主迴圈一起卡住,因為兩者搶同一個 WiFi 驅動鎖。

05

給業主的使用建議

1. 不看畫面時把頁面關掉或切到背景 —— 卡頓次數減半(WI-189 自動處理)
2. 儀表板更新頻率:畫面上的「快速(2s)/標準(3s)/省電(5s)/慢速(10s)/暫停」直接影響設備負載
3. 應用適配:
 - 狀態指示、三色燈、非即時回報 → 目前特性可接受
 - 計數、安全連鎖、需要確定性時序 → 需先確認 2~3 秒的最差延遲是否可接受

後續改善方向(交付後)

依數據,改善順序應為:

- 1. 讓網路操作有真正的時間上界(非阻塞 socket + 狀態機)
- 2. 控制路徑與網路路徑不共用驅動鎖 —— 這才是分執行緒有意義的前提
- 3. (大改)啟用 M4 核心做控制/網路分離

不建議把 LED 搬到獨立執行緒當解法:板上 L1-L4 是鏡射 DO 狀態, DO 本身被延遲時 LED 只會忠實顯示延遲後的狀態,真正變順的只有心跳燈 —— 那是把溫度計拆掉,不是退燒。

文件修訂紀錄

日期	版本	變更
2026-08-20	1.0	首版◦POC 實測 + WI-189 前後對照