

SparkNavi

向成電子有限公司 XIANG CHENG Electronic Co., Ltd.



InertiaX core

IMU600-V2

使用手冊

User Manual

三顆工業級 6 軸 IMU 加氣壓計的慣性量測模組

Triple-IMU Inertial Measurement Module with Barometer

對應韌體	Firmware: 0.11.2
設定工具	SparkNavi IMU600 GUI v1.6.1
發行日期	2026 年 10 月
版 號	1.1

XIANG CHENG Electronic Co., Ltd. (向成電子有限公司)

No. 41, Wenxian 1st Rd., North Dist., Tainan City 704, Taiwan (R.O.C.)

Tel +886-6-3586207 Fax +886-6-3581363

ethan@robotlab.com.tw · sparknavi.com

© 2026 XIANG CHENG Electronic Co., Ltd. All rights reserved.

目錄

1. 產品概述	5
2. 接頭腳位 (J1)	8
開發板接頭定義	12
3. IMU 座標軸 (機體 FRD)	13
4. 快速開始	14
5. GUI 介面介紹 (SparkNavi IMU600 GUI v1.6.1)	15
5.1 連線列	15
5.2 儀表板	15
5.3 參數設定	16
5.4 終端機	16
5.5 檔案選單	16
6. ASCII 協定與指令	17
6.1 行格式	17
6.2 \$SPNVIMURAW — 單顆 IMU (每格 3 行)	18
6.3 \$SPNVIMU — 三顆合併 (每格 1 行)	19
6.4 其他資料行	19
6.5 IMU status 位元	20
6.6 指令	21
7. Binary 協定 (D6 44)	25
7.1 幀格式	25
7.2 訊息表	26

7.3 欄位配置	27
7.4 頻寬參考 (binary).....	27
8. UART2、GPS PPS 對齊、外部觸發、SPI 飛控輸出	28
8.1 UART2 第二輸出埠	28
8.2 GPS PPS 對齊	29
GPS_PPS_IN 輸入要求 (精度與容許範圍)	29
8.3 外部觸發 (EXT_TRIGGER)	31
8.4 SPI 飛控輸出與 ArduPilot (fw 0.10.0 起).....	32
介面規格.....	32
封包格式 (128 bytes · little-endian · SPI, ArduPilot).....	32
ArduPilot 設定步驟	33
注意事項.....	34
9. MCLR 外部重置參考電路	35
10. 韌體更新與救援	37
11. 疑難排解	38

修訂紀錄

[illegible]

1. 產品概述

IMU600-V2 是三顆工業級 6 軸 IMU 加一顆氣壓計的慣性量測模組，透過 UART 或 SPI 輸出三顆各自的資料、合併後的資料、姿態 (AHRS) 與氣壓，並可作為 ArduPilot 飛控的外接 IMU (§8.4)。本手冊對應韌體 **fw 0.11.2** 與設定工具 **SparkNavi IMU600 GUI v1.6.1**。

項目	規格
感測器	3 × 工業級 6 軸 IMU (陀螺 $\pm 300^\circ/\text{s}$ 、加速度 $\pm 156.9 \text{ m/s}^2$) ; 1 × 氣壓計
內部取樣率	每顆約 1475 Hz
輸出率	5 / 10 / 25 / 50 / 100 / 200 / 400 / 500 / 1000 Hz (出廠 50 Hz) ; 或外部觸發，每個 EXT_TRIGGER 上升緣一筆 (1 ~ 1000 Hz)。 接 ArduPilot 請用 1000 Hz (§8.4)
氣壓輸出率	10 / 25 / 50 / 100 Hz (出廠 50 Hz)
座標系	機體 FRD (+X 前、+Y 右、+Z 下)，三顆已轉到同一座標
主介面 UART1	3.3 V TTL，8N1，出廠 460800 bps ASCII ; 可改 460800 / 921600 / 1500000 / 2000000、binary
第二輸出 UART2	可輸出 binary 或 ASCII 格式、三顆 IMU 原始數據、IMU 合併融合數據、氣壓計數據等
SPI 飛控介面	SPI3 從機，可設定 ArduPilot IMU 或是一般 SPI 應用：電壓準位 3.3 V，SPI 時脈模式：Mode 3 / Mode 0，建議 SCK 2 MHz ; 128 byte 封包 (CRC-16) ; 出廠預設關閉，見 §8.4

項目	規格
飛控軟體支援	ArduPilot (以 Copter 4.7.1 於 SparkNavi Blue2 飛控驗證) · IMU 輸出率須設 1000 Hz ; 驅動由 SparkNavi 提供
時間同步	GPS_PPS_IN (PPS 上升緣對齊輸出時間格) 、EXT_TRIGGER (外部觸發取樣)
供電	輸入電壓 : +5.0 V $\pm$ 5% 建議工作電壓+5V , 可讓飛控直接供電。
韌體更新	加密簽章 .spfw 檔 , 用 GUI 經 UART1 更新

輸出數據一覽

資料	ASCII 行	Binary id	內容
單顆 IMU 原始	\$SPNVIMURAW	0x01	每顆每格一筆：平均角速度、比力、溫度、 $\Delta\theta$ 、 Δv 、狀態
三顆合併	\$SPNVIMU	0x02	剔除異常顆後進行演算法計算
氣壓	\$SPNVBARO	0x03	氣壓 hPa、板溫、標準大氣高度
姿態	\$SPNVAHRS	0x04	roll / pitch / yaw、四元數
GPS PPS	\$SPNVPPS	0x05	PPS 鎖定狀態、晶振誤差
統計 (1 Hz)	\$SPNVSTAT / \$SPNVFUS / \$SPNVSYS	0x10 / 0x11 / 0x12	錯誤計數、頻寬、系統狀態

2. 接頭腳位 (J1)

所有對外訊號都在 J1 (Samtec MW-12-03-G-D-130-110 · $2 \times 12 = 24$ pin · 奇數腳一排、偶數腳一排)。訊號皆為 3.3 V 邏輯，只有 pin 3 / 4 接 +5.0 V 供電。

※ 圖 1、注意 PIN 1 的位置。

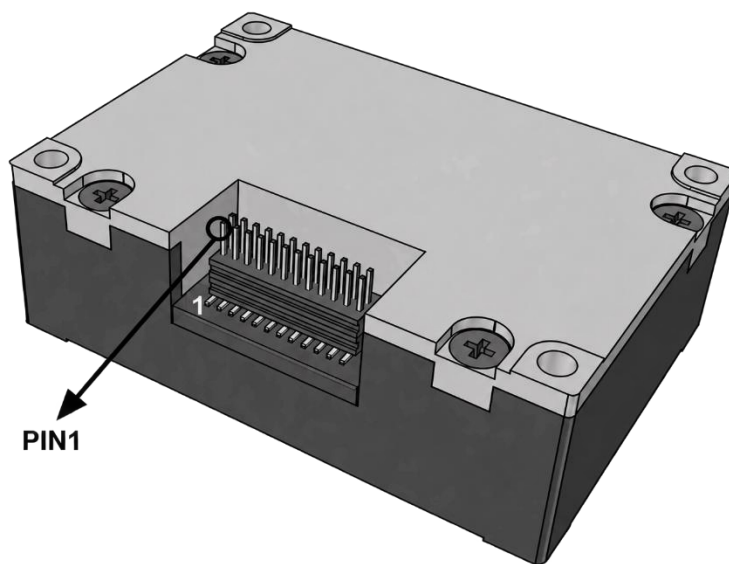


圖 1、IMU600 反面圖

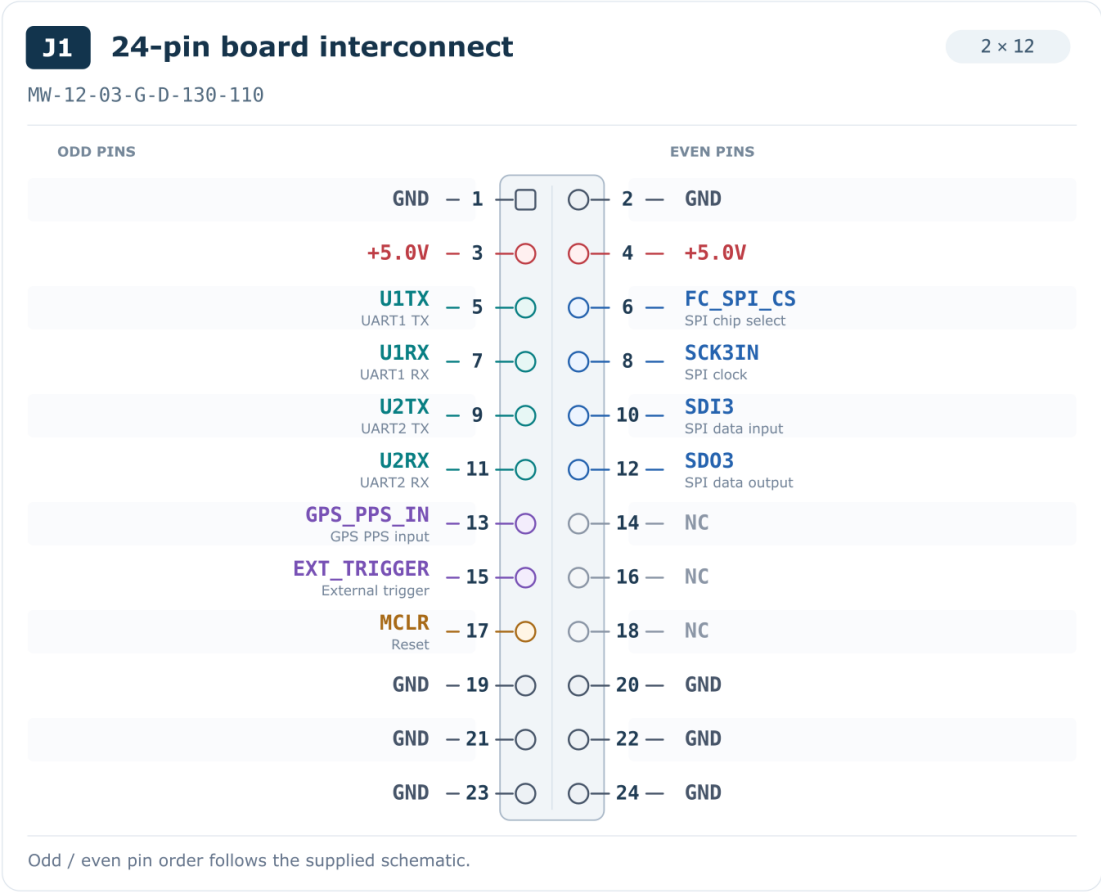


圖 2、IMU600-V2 接腳定義圖

Pin	訊號	方向 (對模組)	說明	Pin	訊號	方向	說明
1	GND	—	地	2	GND	—	地
3	+5.0V	電源輸入	模組供電	4	+5.0V	電源輸入	模組供電
5	U1TX	輸出	UART1 資料 → 主機 RX	6	FC_SPI_CS	輸入	SPI 片選 (\$8.4)

Pin	訊號	方向 (對模組)	說明	Pin	訊號	方向	說明
7	U1RX	輸入	UART1 指令 ← 主機 TX	8	SCK3IN	輸入	SPI 時脈 (§8.4)
9	U2TX	輸出	UART2 第二輸出 埠	10	SDI3	輸入	SPI MOSI (§8.4)
11	U2RX	輸入	目前不使 用，可空 接	12	SDO3	輸出	SPI MISO (§8.4)
13	GPS_PPS_IN	輸入	GPS PPS，上 升緣有效 (§8)	14	—	—	不接
15	EXT_TRIGGER	輸入	外部觸 發，上升 緣 (§8)	16	—	—	不接
17	MCLR	輸入	外部重 置，低電 位有效 (§9)	18	—	—	不接
19–22	—	—	不接				
23	GND	—	地	24	GND	—	地

電氣注意事項

- 供電接 pin 3 / 4 (+5.0 V) , 地(GND)至少接 pin 1 / 2 。板上穩壓產生 3.3 V 給 MCU 與感測器。
- UART、PPS、EXT_TRIGGER、MCLR 都是 3.3 V CMOS 。輸入不要超過 3.3 V ; 主機是 5 V 邏輯時請加電位轉換晶片。
- UART 接法：模組 U1TX → 主機 RX、模組 U1RX ← 主機 TX , 共地。USB 轉 UART 請選 3.3 V 版、支援 ≥ 921600 bps (如 FT232H / CP2102N)。
- 用不到的輸入 (U2RX、PPS、EXT_TRIGGER、MCLR) 可以空接(浮接) ; MCLR 板上已有 10 k Ω 上拉。
- SPI (pin 6 / 8 / 10 / 12) 是 SPI 從機介面 (§8.4) , 出廠關閉 (MISO 高阻抗) ; 不接飛控時請空接 , CS 板上已有內部上拉。

開發板接頭定義

J3	SPI	7 PIN
PIN	SIGNAL	
1	VDD_5V_IN	
2	SCK3IN	
3	SD03	
4	SDI3	
5	FC_SPI_CS	
6	MCLR	
7	GND	

J4	UART	10 PIN
PIN	SIGNAL	
1	VDD_5V_IN	
2	U1RX	
3	U1TX	
4	U2RX	
5	U2TX	
6	MCLR	
7	EXT_TRIGGER	
8	GPS_PPS_IN	
9	NC	
10	GND	

圖 3、開發底板 J3 與 J4 接頭定義圖

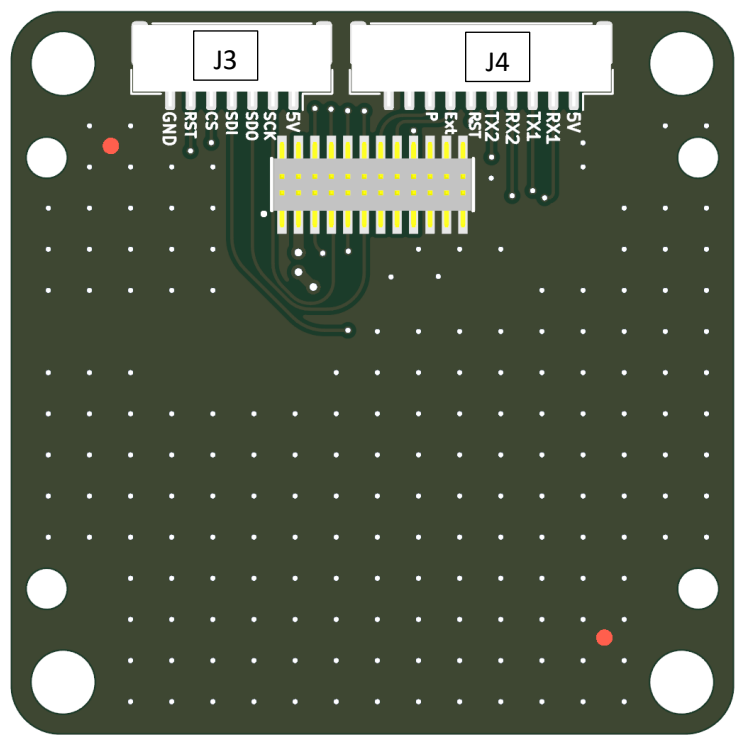


圖 4、開發底板正面圖

3. IMU 座標軸 (機體 FRD)

所有輸出 (三顆原始、合併、AHRS) 都在同一個機體座標 **FRD** : +X 朝前 (Forward)、+Y 朝右 (Right)、+Z 朝下 (Down) , 右手座標系。

量	正方向	靜置水平時的讀數
ax / ay / az	比力 (加速度計量到的力, 含重力反作用) 沿 +X / +Y / +Z	$ax \approx 0$ 、 $ay \approx 0$ 、 $az \approx -9.79 \text{ m/s}^2$
gx (roll rate)	繞 +X 右手旋轉: 右側往下	≈ 0
gy (pitch rate)	繞 +Y 右手旋轉: 前端抬起	≈ 0
gz (yaw rate)	繞 +Z 右手旋轉: 由上往下看順時針	≈ 0 (只有零偏與地球自轉分量)
roll / pitch / yaw (AHRS)	ZYX 歐拉角, 機體 FRD → NED	右滾、抬頭、順時針為正

快速確認方向: 板子元件面朝上平放, az 應為 -9.79 ; 把前端抬起, ax 變正、pitch 變正; 右側往下傾, ay 變負、roll 變正; 由上往下看順時針轉, gz 為正。GUI 的 3D 模型以紅色箭頭標示 +X (FWD)。

安裝建議

- 模組 +X 對齊載具前方、元件面朝上, 此時輸出即載具 FRD。其他安裝方向請在主機端做旋轉。
- 剛性固定; 引擎 / 馬達 / 螺旋槳震動大的載具請加減震, 避免加速度瞬間峰值超過量程 (狀態 bit4 飽和, 見 §6)。

機體 FRD：+X 前、+Y 右、+Z 朝下穿過板子

俯視，元件面朝上

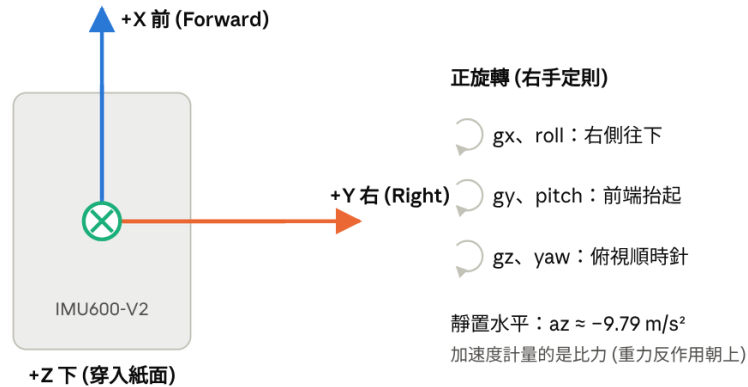


圖 5、IMU600-V2 機體座標 FRD

4. 快速開始

接上電源與 UART1、打開 GUI 按「連線」，就能看到資料；出廠設定為 460800 bps、ASCII、50 Hz、三顆原始 + 合併、AHRS 與氣壓全開。

1. 接線：J1 pin 3/4 接 +5 V、pin 1/2 接地；pin 5 (U1TX) 接 USB-UART 的 RX、pin 7 (U1RX) 接 TX。
2. 安裝 GUI：解壓 SparkNavi-IMU600-GUI-v1.6.1.zip，執行 SparkNavi_IMU600.exe。
3. 連線：上方選 COM port、鮑率 460800 → 按「連線」。GUI 自動讀回設定 (CFG) 與裝置資訊 (INFO)，狀態列顯示韌體版本與序號。
4. 確認資料：儀表板的 IMU 1/2/3、合併、AHRS、氣壓計卡片持續更新；靜置水平時 az ≈ -9.79、roll / pitch ≈ 0。
5. 改設定：參數設定頁改「設定值」→「套用變更」。套用只改 RAM，要下次上電仍生效請按「寫入裝置 (SAVE)」。
6. 錯的鮑率 / 模式連不上：先按「自動偵測鮑率」；仍不行用「開機救援」(\$5.3) 回到出廠值。

不用 GUI 也可以：任何串口終端機 (460800 8N1) 都能看到 ASCII 行，指令格式見 \$6。

5. GUI 介面介紹 (SparkNavi IMU600 GUI v1.6.1)

GUI 有三個頁籤：儀表板看資料、參數設定改設定與校正、終端機看原始收發與手打指令；上方是連線列，「檔案」選單有錄製與韌體更新，「工具」選單有陀螺尋北精靈 (請依照視窗畫面步驟進行操作)。需要韌體 fw 0.4.2 以上；UART2 / PPS 設定需要 fw 0.8.0 以上；輸出率 200 / 400 / 500 Hz 需要 fw 0.10.3 以上；SPI3 輸出設定需要 fw 0.10.0 以上。

※ 出廠預設韌體版本為: fw 0.11.2，此版本支援全部功能。

5.1 連線列

元件	功能
連接埠 / 鮑率	選 COM port 與 UART1 鮑率 (出廠 460800)
連線 / 中斷	連線後自動送 CFG,1、INFO,1
自動偵測鮑率	依序以 460800 / 921600 / 1500000 / 2000000 送 PING
狀態列	韌體版本、序號 (SN)、接收速率
● REC	錄製中閃爍 (檔案 → 開始錄製原始資料...，存成 .bin)

5.2 儀表板

- 左側：AHRS 姿態儀 (HUD) 與 3D 模型。HUD 左框 = IMU 溫度、右框 = 氣壓相對高度；「AHRS 重置」需靜置 1 秒；「設定航向...」手動給 heading。3D 模型標題列可旋轉 / 翻轉模型使前方對齊紅色 X。
- 右上方版面畫面 1-IMUx：IMU 1 / 2 / 3、合併、AHRS、氣壓計、系統、裝置。狀態綠色 = 正常；顏色只看錯誤位元，PPS 標記不算錯誤。
- 右上方版面畫面 2-系統資訊：UART 丟棄 (括號內為 UART2)、CPU、輸出率、GPS PPS 狀態 (NO_SIGNAL / ACQUIRING / LOCKED + 晶振誤差 ppm / HOLDOVER)。系統卡與錯誤計數需 STAT 開啟。

- **即時圖**：角速度 / 加速度圖上方勾選來源 [IMU 1][IMU 2][IMU 3][合併] × 軸向 [X][Y][Z]；下方讀數條顯示最新值、平均 μ 、標準差 σ (靜置時 σ 即雜訊)。氣壓圖可切相對高度 / hPa / 溫度，「高度歸零」只影響顯示。

5.3 參數設定

- **設定表**：左欄「裝置目前值」(由 CFG 讀回)，右欄「設定值」可編輯；改過的項目以 ● 標橘色。分組：輸出 (RATE / OUT / BARO / BRATE / AHRS / STAT)、介面 (MODE / BAUD)、合併門檻 (GTH / ATH)、AHRS (AKP / AKI)、**UART2** (輸出內容複選 / 格式 / 鮑率)、**GPS PPS 對齊**、**SPI3 輸出** (用途 / 內容 / 格式 / 時脈模式，§8.4)。
- **套用變更**：只送有變更的項目；改 BAUD 時 GUI 收到 ACK 自動跟著切換本機鮑率。
- **寫入裝置 (SAVE)**：存入 flash，下次上電即為這組設定。**恢復出廠 (FACTORY)**：清除使用者設定 (校正不清)。
- **開機救援**：不知道存了什麼鮑率 / 模式而連不上時使用。按下後 20 秒內把模組斷電再上電，GUI 在開機 0.5 秒救援窗送 FACTORY,1。
- **裝置資訊**：韌體版本、建置旗標、序號、初始化結果。
- **UART 頻寬估計**：UART1 與 UART2 各一條， $\geq 90\%$ 標紅 (會丟資料)。
- **校正**：加速度六面校正精靈 (板子六面依序朝上各靜置 4 秒，自動計算並寫入)、合併權重、寫入校正 (CALSAVE)、清除校正 (CALCLR)。
- **IMU 輸出模式 (fw 0.9.0)**：「外部觸發 EXT_TRIGGER」分組選「連續輸出」或「外部觸發」，套用後按 SAVE 保留。外部觸發時儀表板系統卡「輸出模式」顯示已收到的觸發數，詳見 §8.3。

5.4 終端機

- 所有收發解碼成文字 (binary 幀也轉成文字)，可依類別過濾；高頻資料行預設不顯示 (勾「資料行」)。
- 輸入 RATE,100 或 rate 100 都會送成 \$SPNVCMD,RATE,100*CS；以 \$ 開頭的整行原樣送出。↑ / ↓ 叫出歷史指令。
- 快捷鍵：INFO / CFG / DBG / PING / STAT now / **BOOT (重新啟動)**。

5.5 檔案選單

- **開始錄製原始資料...**：把收到的 bytes 原封不動存成 .bin，之後可用任何解析器處理。
- **韌體更新...**：見 §10。

- 說明 → 語言：繁體中文 / English (重開生效)；淺色 / 深色主題。

6. ASCII 協定與指令

出廠輸出為 NMEA 風格的 ASCII 行；指令不論輸出模式一律以 ASCII \$SPNVCMDB 送到 UART1。

6.1 行格式

項目	規格
行格式	\$<body>*<XX>\r\n，body 第一欄為訊息名稱 (SPNV 開頭)，欄位以, 分隔
校驗 XX	\$ 與 * 之間所有字元逐位元組 XOR，2 位大寫十六進位 (同 NMEA 0183)
數值	十進位；浮點為固定小數位數
單位	°/s、m/s ² 、°C、rad、m/s、hPa、μs、s
ts_us	64-bit，開機歸零的微秒，為該筆輸出視窗的結束時刻；同一時間格的三顆 RAW 與合併行 ts_us 完全相同
解析建議	以 \$ 同步、校驗錯整行丟；用第一欄分派；新版韌體可能在尾端加欄位，不要假設固定欄數
頻寬不足	整行丟棄 (不會送半行)，累計在 \$SPNVSYS 第 1 欄

```
def spnv_checksum(body: str) -> str:
```

```
    cs = 0
```

```
    for ch in body.encode('ascii'):
```

```
        cs ^= ch
```

```
    return f"{cs:02X}"
```

```
cmd = "SPNVCMDB,RATE,100"
```

```
ser.write(f"${cmd}*{spnv_checksum(cmd)}\r\n".encode())
```

6.2 \$SPNVIMURAW — 單顆 IMU (每格 3 行)

\$SPNVIMURAW,<id>,<ts_us>,<gx>,<gy>,<gz>,<ax>,<ay>,<az>,<temp>,<n>,<status>,<dthx>,<dthy>,<dthz>,<dvx>,<dvy>,<dvz>,<dt>*XX

#	欄位	單位	說明
1	id	—	IMU 1 / 2 / 3
2	ts_us	μs	視窗結束時刻
3–5	gx gy gz	°/s	視窗平均角速度 (4 位小數)
6–8	ax ay az	m/s ²	視窗平均比力・含重力 (4 位)
9	temp	°C	感測器溫度
10	n	—	這筆包含的原始樣本數 (50 Hz ≈ 29~30)
11	status	bit	0 = 全部正常・見 6.5
12–14	dthx dthy dthz	rad	視窗內旋轉向量 Δθ (含 coning 補償・7 位)
15–17	dvx dvz dvz	m/s	視窗內速度增量 Δv (含 sculling 補償・6 位)
18	dt	s	實際積分區間 ≈ 1 / 輸出率

範例 (50 Hz 靜置) : \$SPNVIMURAW,1,12345680,0.0123,-0.0456,0.0089,0.0210,-0.0150,-9.7912,31.25,30,0,0.0000043,-0.0000159,0.0000031,0.000420,-0.000300,-0.195824,0.020000*0F

6.3 \$SPNVIMU — 三顆合併 (每格 1 行)

\$SPNVIMU,<ts_us>,<gx>,<gy>,<gz>,<ax>,<ay>,<az>,<temp>,<dthx>,<dthy>,<dthz>,<dvx>,<dvy>,<dvz>,<dt>,<n_used>,<flags>*XX

欄位與 RAW 相同 (沒有 id、n、status) · 另加 n_used (實際採用顆數 1~3) 與 flags：

位元	值	名稱	意義
bit0-2	1 / 2 / 4	IMUn_REJECTED	IMU 1 / 2 / 3 這格被排除
bit3	8	SPREAD_WARN	某顆偏離其他顆 · 提醒注意
bit4	16	DEGRADED	可用顆數 < 2
bit5	32	INCOMPLETE	這格沒收齊所有顆
bit6	64	SATURATED	至少一顆量程飽和 · 數值被截頂
bit7	128	PPS	不是錯誤：這格 ts_us 正好是 GPS PPS 上升緣

6.4 其他資料行

訊息	格式	說明
\$SPNVBARO	<ts_us>,<press_hpa>,<temp_c>,<alt_m>,<status>	氣壓 hPa (4 位)、板溫、ISA 標準大氣高度 (只供相對高度參考)、status 0 = 正常。 ts_us 對齊 IMU 時間格
\$SPNVAHRS	<ts_us>,<roll>,<pitch>,<yaw>,<q0>,<q1>,<q2>,<q3>	姿態 deg、四元數 body → NED (q0 純量)、對準狀態、旗標、陀螺零偏估計 °/s、靜

訊息	格式	說明
	,<align>,<flags>,<bgx>,<bggy>,<bgz>,<zb_s>	止零偏校正進度 zb_s (s · 0 ~ 10)
\$SPNVPPS	<ts_us>,<edges>,<captures>,<state>,<err_ppm>,<age_ms>,<align>	GPS PPS 狀態 · 見 §8.2
\$SPNVSTAT	<id>,<dry>,<out>,<crc_err>,<frame_err>,<spi_timeout>,<dropped>,...	每顆 1 Hz 統計 ; crc_err / spi_timeout / dropped 正常為 0
\$SPNVFUS	<outputs>,<rej1>,<rej2>,<rej3>,<incomplete>,<degraded>,<gth>,<ath>	合併統計
\$SPNVSYS	<uart_drop>,...,<rate_hz>,<out_mode>,...,<proto_mode>,<baud>,<pps_state>,<u2_drop>	系統統計 (18 欄) ; uart_drop / u2_drop 增加 = 頻寬不足 ; 最後兩欄 trig_cnt / trig_rej = 外部觸發數 / 被擋的毛刺

AHRS align : 0 = 初始對準中 (開機或 reset 後 1 秒 , 請靜置) ; 1 = roll / pitch 有效、yaw 為相對角會漂 ; 2 = 已用 HDG 手動給航向。AHRS flags : bit0 動態中、bit1 零偏估計暫停、bit2 航向來自 HDG、bit3 本筆不用加速度計、bit4 對準時板子有動 (建議靜置後 AHRS,reset)、bit5 陀螺零偏已由 ≥ 10 秒的靜止時段估計過 (狀態 , 不是警告)、bit6 最近 10 秒判定為靜止 (狀態)。

6.5 IMU status 位元

status 是該輸出視窗內所有原始樣本旗標的 OR , 意思是「這段時間內有沒有發生過」。判斷資料好壞請用 status & 0x7F。

位元	值	名稱	意義	資料可信度
bit0	1	SPI_CRC	內部通訊錯誤	不可信
bit1	2	CE	感測器收到無效請求	仍有效

位元	值	名稱	意義	資料可信度
bit2	4	IDS	感測器內部資料 狀態異常	可疑
bit3	8	S_ERROR	感測器自我診斷 錯誤	不可信
bit4	16	S_SAT	量程飽和 (> ±300 °/s 或 > ±156.9 m/s²)	數值被截頂
bit5	32	S_INIT	感測器初始化中 (上電後約 250 ms)	不可信
bit7	128	PPS	不是錯誤：這筆 ts_us 是 PPS 上 升緣	可信

常見組合：0 正常；16 飽和 (撞擊、劇烈震動)；12 感測器錯誤；32 / 44 開機初始化中。合併時 `status & 0x7F ≠ 0` 的顆該格自動排除。

6.6 指令

格式 \$SPNVCMD,<key>,<value>[*XX]\r\n。校驗可省略 (方便手打)，有帶就會驗；不分大小寫。

每個指令回 \$SPNVACK,<key>,<實際套用值> 或 \$SPNVNAK,<key>,<原因>。指令只改 RAM，要保留請 SAVE,1。

key	value	作用	出廠值
RATE	5 / 10 / 25 / 50 / 100 / 200 / 400 / 500 / 1000	IMU 輸出率 Hz (200 / 400 / 500 為 fw 0.10.3 起)；接 ArduPilot 用 1000	50

key	value	作用	出廠值
OUT	raw / fused / both / none	UART1 送哪些 IMU 行	both
BARO	0 / 1	氣壓行關 / 開	1
BRATE	10 / 25 / 50 / 100	氣壓輸出率 Hz	50
AHRS	0 / 1 / reset	姿態行關 / 開 ; reset = 重新對準 (靜置 1 秒)	1
HDG	-360 ~ 360	手動設定航向 deg (align 需 ≥ 1)	—
AKP / AKI	0 ~ 50	AHRS 增益 Kp / Ki	2.0 / 0.05
GTH / ATH	$0 < v \leq 100$	合併剔除門檻 °/s / m/s ²	0.5 / 0.5
STAT	0 / 1 / now	1 Hz 統計關 / 開 ; now = 立刻送一組	1
HB	0 / 1	診斷心跳 \$SPNVHB	0
MODE	ascii / binary	UART1 輸出協定 (ACK 以舊模式送出)	ascii
BAUD	460800 / 921600 / 1500000 / 2000000	UART1 鮑率 (ACK 以舊鮑率送出，主機收到後再切)	460800
U2OUT	raw / fused / ahrs / baro / stat 用 + 組合 ; none ; all	UART2 輸出內容	raw
U2MODE	ascii / binary	UART2 協定	binary

key	value	作用	出廠值
U2BAUD	115200 / 230400 / 460800 / 921600 / 1500000 / 2000000	UART2 鮑率	460800
PPS	0 / 1	GPS PPS 對齊輸出時 間格	1
TRIG	0 / 1	IMU 輸出模式：0 = 連續輸出、1 = 外部 觸發 (§8.3)	0
AZB	0 / 1	AHRS 靜止零偏估計 (含 Z 軸)：靜止 10 秒 自動校正陀螺零偏 (fw 0.9.1 起)	1
SPI	off / ardupilot / stream	SPI3 從機輸出用途： off = 關閉 (MISO 高 阻抗)、ardupilot = 給 ArduPilot 的 IMU 封包、stream = 與 UART2 相同內容 (§8.4)	off
SPIOUT	同 U2OUT	stream 模式送哪些資 料	none
SPIMODE	ascii / binary	stream 模式的格式	binary
SPICK	3 / 0	SPI 時脈模式 (CPOL / CPHA)：3 = Mode 3、0 = Mode 0	3

key	value	作用	出廠值
SAVE	1	目前全部設定寫入 flash (期間 IMU 短暫停止)	—
FACTORY	1	恢復出廠設定 (校正不清) ; ACK 後主機改回 460800 ASCII	—
CFG	1	印目前設定 \$SPNVCFG + \$SPNVU2 + \$SPNVSPI	—
INFO	1	印版本、重置原因、序號、初始化結果	—
REINIT	1	重新初始化三顆 IMU	—
PING	任意字串	連線測試，原樣回傳	—
BOOT	1	重新啟動 (經 bootloader，載入已存設定)	—
CAL	1	印目前校正 \$SPNVCAL	—
ACAL	<imu>,<bx>,<by>,<bz>,<kx>,<ky>,<kz>	設定加速度校正： $a = (a_{\text{raw}} - b) \cdot k$; $ b \leq 2$ 、 $0.95 \leq k \leq 1.05$	無校正
FWGT	g a,<w1>,<w2>,<w3>	合併權重 (陀螺 / 加速度) · 相對值 $0 < w \leq 100$	全 1
CALSAVE	1	校正與權重寫入 flash	—

key	value	作用	出廠值
CALCLR	1	清除校正	—

讀回設定範例：

```
$SPNVCMD,CFG,1
$SPNVACK,CFG,1*XX
$SPNVCFG,baud=460800,mode=ascii,rate=50,out=3,stat=1,hb=0,baro=1,brate=50,ahrs=1,gth=0.5
00,ath=0.500,akp=2.000,aki=0.0500,saved=1,fw=0.11.2,trig=0,azb=1*XX
$SPNVU2,out=raw,mode=binary,baud=460800,drop=0,pps=1,pps_state=1,pps_err_ppm=0.000*XX
$SPNVSPI,mode=off,out=none,fmt=binary,ck=3,txn=0,ok=0,short=0,delivered=0,drop=0,...*XX
```

開機訊息：出貨韌體上電不印任何文字，第一行就是資料；用 INFO,1 索取 \$SPNVBOOT (版本) 與 \$SPNVRESET (上次重置原因：POR 上電、EXTR(MCLR) 外部重置、SWR 軟體重置、WDTO 看門狗)。

7. Binary 協定 (D6 44)

\$SPNVCMD,MODE,binary 後 UART1 輸出改為二進位幀，頻寬約為 ASCII 的一半；指令仍以 ASCII 送，回覆 (ACK / NAK / INFO) 以 0x7F TEXT 幀送出。單位與位元定義與 ASCII 完全相同。UART2 出廠就是 binary。

7.1 幀格式

D6 44 id (1) len (1) seq (1) payload (len) crc16 (2)				
欄位		說明		
sync		0xD6 0x44		
id		訊息類別 (見 7.2)		
len		payload 長度 0–255		
seq		每個 id 各自 +1 循環，用來偵測掉幀； UART1 與 UART2 各自計數		
payload		little-endian；f32 = IEEE-754 單精度； ts_us = u64		

欄位	說明
crc16	CRC-16/CCITT-FALSE (poly 0x1021、init 0xFFFF、不反射、無 xorout) · LE · 範圍為 id、len、seq、payload；"123456789" → 0x29B1

```
def crc16_ccitt(data: bytes) -> int:
```

```
    crc = 0xFFFF
```

```
    for b in data:
```

```
        crc ^= b << 8
```

```
        for _ in range(8):
```

```
            crc = ((crc << 1) ^ 0x1021) & 0xFFFF if crc & 0x8000 else (crc << 1) & 0xFFFF
```

```
    return crc
```

```
# 驗證：crc16_ccitt(frame[2:5+len]) == int.from_bytes(frame[5+len:7+len], 'little')
```

7.2 訊息表

id	名稱	對應 ASCII	payload	Python struct
0x01	IMU_RAW	\$SPNVIMURAW	68	<BBHQ3f3ff3f3ff
0x02	IMU (合併)	\$SPNVIMU	68	<Q3f3ff3f3ffBBB B
0x03	BARO	\$SPNVBARO	24	<QfffB3x
0x04	AHRS	\$SPNVAHRS	52	<Q4f3f3fBBH
0x05	PPS	\$SPNVPPS	24	<QIiBBH
0x10	STAT	\$SPNVSTAT	44	<BBBB9IHH
0x11	FUS	\$SPNVFUS	32	<6lff
0x12	SYS	\$SPNVSYS	48 (fw 0.8.0 : 40 ; 更舊 : 36)	<7IHBBBBBBIII
0x7F	TEXT	ACK / NAK / INFO 等文字	可變	ASCII 字串 (不含 \$、*XX)

7.3 欄位配置

0x01 IMU_RAW : imu_id u8、status u8、n u16、ts_us u64、gyro[3] f32 °/s、acc[3] f32 m/s²、temp f32 °C、dth[3] f32 rad、dv[3] f32 m/s、dt f32 s。

0x02 IMU : ts_us u64、gyro[3]、acc[3]、temp、dth[3]、dv[3]、dt (f32)、n_used u8、flags u8、used_mask u8 (bit n = IMU n+1 有採用)、reserved u8。

0x03 BARO : ts_us u64、press_hpa f32、temp_c f32、alt_m f32、status u8、reserved[3]。

0x04 AHRS : ts_us u64、q[4] f32 (q0 q1 q2 q3、body → NED)、roll / pitch / yaw f32 deg、gyro_bias[3] f32 °/s、align u8、flags u8、zb_ds u16 (靜止零偏校正進度，單位 0.1 s)。(注意：順序為先四元數、再歐拉角，與 ASCII 不同。)

0x05 PPS : ts_us u64 (最近合格 PPS 邊緣)、edges u32、captures u32、err_ppb i32 (本地晶振相對 PPS，+ = 本地快)、state u8、align u8、age_ms u16。

0x10 STAT : imu_id、dry_timeout、dry_pin、isr_cpu_pct (u8)、dry、out、crc_err、frame_err、spi_timeout、dropped、dcnt_skip、dry_kicks、resync (u32)、isr_us_max、isr_us_last (u16)。

0x11 FUS : outputs、rej[3]、incomplete、degraded (u32)、gyro_thresh、acc_thresh (f32)。

0x12 SYS : uart_drop、loops_per_s、uart_resets、pps_cnt、baro_samples、baro_i2c_err、baro_timeouts (u32)、rate_hz u16、isr_cpu_total_pct u8、out_mode u8 (0 none / 1 raw / 2 fused / 3 both)、baro_stale u8、baro_hz u8、proto_mode u8 (0 ascii / 1 binary)、pps_state u8、u2_drop u32、trig_cnt u32、trig_rej u32。解析時依 len 判斷 36 / 40 / 48。

7.4 頻寬參考 (binary)

設定	流量	460800	921600
50 Hz both + AHRS + BARO	≈ 19 kB/s	OK	OK
100 Hz both + AHRS + BARO	≈ 37 kB/s	OK	OK

設定	流量	460800	921600
200 Hz fused + AHRS	≈ 27 kB/s	OK	OK
400 Hz fused + AHRS	≈ 54 kB/s	超過	OK
500 Hz fused + AHRS	≈ 67 kB/s	超過	OK
1000 Hz 單顆 raw	≈ 75 kB/s	超過	OK
1000 Hz fused + AHRS	≈ 134 kB/s	超過	超過 → 用 1500000 / 2000000

460800 bps 實際約 46 kB/s。GUI 參數頁的頻寬估計會在 ≥ 90% 時標紅。

8. UART2、GPS PPS 對齊、外部觸發、SPI 飛控輸出

8.1 UART2 第二輸出埠

UART2 (J1 pin 9 U2TX) 是只輸出的第二個資料埠，適合同時接飛控 / 導航電腦與 GUI。出廠為 binary、三顆 IMU 原始資料 (raw)、460800 bps，上電即輸出。

- 設定一律經 UART1：GUI 參數頁「UART2」分組，或指令 U2OUT / U2MODE / U2BAUD，再 SAVE,1。U2RX 目前不收指令。
- 輸出內容可複選：raw 三顆 IMU、fused 合併、ahrs 姿態、baro 氣壓、stat 1 Hz 統計 + PPS 狀態。例：\$SPNVCMU,U2OUT,raw+baro。
- 格式與 UART1 完全相同 (ASCII 行或 D6 44 幀)；binary seq 自己計數。ACK / NAK / INFO 等文字只走 UART1。
- 與 UART1 的 OUT / BARO / AHRS / STAT 互不影響，例如 UART1 只送合併給 GUI、UART2 送三顆 raw 給飛控。
- 頻寬：binary raw 三顆 = 225 byte / 輸出，100 Hz ≈ 22.5 kB/s (460800 OK)；1000 Hz ≈ 225 kB/s 超過 2000000 bps，請改送 fused 或降頻。緩衝滿整筆丟棄，累計在 \$SPNVSYS 第 16 欄與 GUI 系統卡「UART 丟棄 (U2 n)」。

8.2 GPS PPS 對齊

GPS (或任何 1 Hz 參考) 的 PPS 接到 GPS_PPS_IN (pin 13) 後，IMU 輸出時間格的 0 點會對齊 PPS 上升緣。模組不接收 GPS 時間，只用邊緣；主機把帶 PPS 標記的那一筆對到 GPS 整秒即可。出廠已開啟 (PPS,1)，沒接 PPS 時自動用自由時間格。

- 鎖定後每個 PPS 秒之間剛好輸出 RATE 筆，其中一筆的 ts_us 等於 PPS 上升緣時間戳；該筆 \$SPNVIMURAW status bit7 = 1、\$SPNVIMU flags bit7 = 1。
- 每秒以量到的 PPS 間隔修正本地晶振誤差 (格點間隔 = 量到的 1 s / RATE)，晶振的數十 ppm 偏差不會累積。err_ppm 即此偏差。
- ts_us 仍是模組自己的開機時鐘；PPS 只決定「在哪個時間點切輸出視窗」。
- 鎖定那一刻時間格會跳一次相位 (最多一個輸出週期)，該筆 dt 較短或較長；之後連續。
- 電氣：3.3 V CMOS，上升緣有效，內部下拉，不要超過 3.3 V。脈衝寬度 $\geq 1 \mu\text{s}$ (GPS 一般 100 ms)。

GPS_PPS_IN 輸入要求 (精度與容許範圍)

項目	要求
頻率 / 週期	1 Hz，相鄰兩個上升緣的間隔必須在 $1 \text{ s} \pm 1000 \text{ ppm}$ 內 (0.999 ~ 1.001 Hz，即 999 ~ 1001 ms)。GPS 模組的 PPS 沒有問題；MCU 軟體延時、一般信號產生器、PC 定時產生的 1 Hz 常常不夠準
抖動	每個間隔都必須在上述範圍內。任何一次超出就重新取得，連續 2 個合格間隔才會鎖定
鎖定	連續 2 個合格 1 s 間隔 $\rightarrow$ state 由 2 (ACQUIRING) 變 3 (LOCKED)
漏脈衝	漏掉 1 ~ 9 個脈衝仍可沿用相位 (間隔為 $k \text{ 秒} \pm k \times 1000 \text{ ppm}$ ， $k = 2 \sim 10$)；失去 PPS 超過 1.5 s 進入 HOLDOVER
雜訊	間隔短於 1 s - 1000 ppm 的邊緣視為毛刺丟棄，不影響狀態；captures - edges 為被擋掉的次數
量測解析度	模組 50 MHz 時脈擷取邊緣，解析度 20 ns；err_ppm 為本地晶振相對 PPS 的偏差 (一般數十 ppm 內)

提示：\$SPNVPPS 只在「合格邊緣」或「狀態改變」時送出 (UART1 需 STAT,1)。\$SPNVSYS 第 7 欄 pps_cnt 每秒增加、但第 15 欄 pps_state 一直是 2，代表邊緣有進來但間隔不合格，請用示波器量 PPS 的實際頻率 (範例：產生器標示 1 Hz 但實測 993.83 mHz，約 6200 ppm 偏差，不會鎖定)。

state	名稱	說明
0	OFF	PPS,0：自由時間格
1	NO_SIGNAL	沒有合格 PPS (沒接)，用自由時間格
2	ACQUIRING	有邊緣，尚未連續 2 次落在 $1\text{ s} \pm 1000\text{ ppm}$
3	LOCKED	對齊中
4	HOLDOVER	曾鎖定，> 1.5 s 沒有 PPS：照最後的間隔外推；PPS 回來立即恢復 LOCKED

\$SPNVPPS,<ts_us>,<edges>,<captures>,<state>,<err_ppm>,<age_ms>,<align> 在每個合格邊緣與狀態改變時送出 (UART1 需 STAT,1；UART2 需 U2OUT 含 stat)。例：

\$SPNVPPS,1234000000,58,58,3,12.345,0,1*XX。

鎖定後每個 PPS 秒剛好 RATE 筆，其中一筆時間戳等於 PPS 邊緣

以 RATE = 5 Hz 為例 (示意，脈衝寬度未依比例)

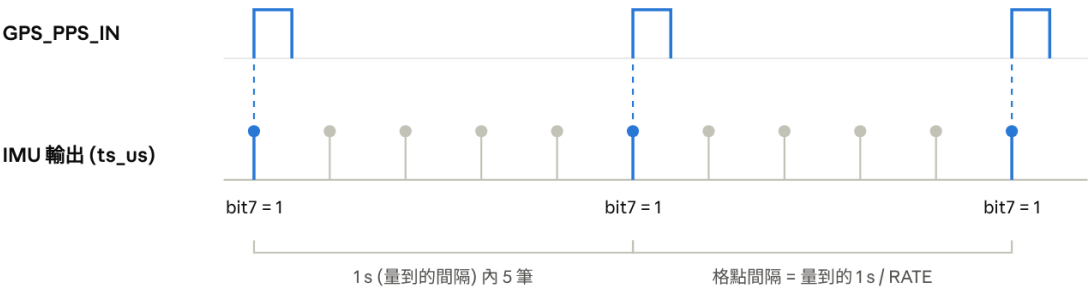


圖 6、PPS 對齊時序 · RATE 5 Hz

帶 bit7 的那一筆 ts_us 就是 PPS 上升緣，主機把它對到 GPS 整秒，其餘各筆依序往後推 1 / RATE。

8.3 外部觸發 (EXT_TRIGGER)

IMU 有兩種輸出模式，用 \$SPNVCMD,TRIG,0|1 或 GUI 參數頁「IMU 輸出模式」切換，SAVE,1 後上電即為該模式（fw 0.9.0 起）。

模式	TRIG	上電後	每筆輸出	EXT_TRIGGER
連續輸出 (Continuous， 出廠)	0	自動依 RATE 輸出	一個 RATE 時間 格	忽略
外部觸發 (Trigger)	1	不輸出，等觸發	兩次觸發之間	每個上升緣輸出一筆

外部觸發模式的行為：

- 第一個上升緣只是起點，不輸出；之後**每個上升緣輸出一筆**（三顆 \$SPNVIMURAW + \$SPNVIMU + \$SPNVAHRS，仍依 OUT / AHRS / U2OUT 開關）。外部送 100 Hz 就輸出 100 筆。
- 該筆 ts_us = **觸發上升緣的時間戳**； $\Delta\theta / \Delta v$ = 上一個觸發到這個觸發之間的完整積分，dt = 兩次觸發的間隔。逐筆相加不重複、不漏，觸發頻率不必是 RATE 的倍數。
- 輸出延遲：觸發後約 0.7 ms（一個原始樣本）加上處理與傳輸時間。
- 建議頻率 1 ~ 1000 Hz。間隔小於 0.5 ms 的邊緣當毛刺丟棄；超過約 10 秒沒有觸發，下一個觸發重新當起點；間隔超過 1 秒時 AHRS 不更新。
- RATE 在此模式不影響 IMU 輸出；氣壓計照 BRATE 自己輸出；PPS 對齊不作用。頻寬請以觸發頻率估算。
- 電氣：3.3 V CMOS，上升緣有效，內部下拉（沒接線不會誤觸發）。觸發源請用推挽 3.3 V 輸出、脈衝寬度 $\geq 1\ \mu\text{s}$ ，**不要超過 3.3 V**。
- 確認：\$SPNVCFG 的 trig=；\$SPNVSYS 第 17 / 18 欄 trig_cnt（已採用的觸發數）/ trig_rej（被丟掉的毛刺）；GUI 系統卡「輸出模式」。

8.4 SPI 飛控輸出與 ArduPilot (fw 0.10.0 起)

J1 pin 6 / 8 / 10 / 12 是 **SPI 從機** 介面：IMU600 為從機，飛控 (或其他主機) 為主機，以固定週期讀取封包。封包含合併後的 $\Delta\theta / \Delta v$ 與三顆 IMU 各自的 $\Delta\theta / \Delta v$ ，可直接作為 **ArduPilot 飛控的外接 IMU**。出廠為關閉 (MISO 高阻抗)，用 SPI 指令或 GUI 參數頁「SPI3 輸出」開啟，並 **SAVE,1** 保存。

ArduPilot 請把 IMU 輸出率設為 1000 Hz。ArduPilot 的起飛前檢查要求陀螺後端速率不低於主迴圈速率的 1.8 倍 (預設 400 Hz 主迴圈需 720 Hz 以上)，200 / 400 / 500 Hz 會出現 **PreArm: Gyro n rate 400Hz < loop rate*1.8 720Hz** 而無法解鎖。200 / 400 / 500 Hz 適用於 UART 輸出或其他主機。

介面規格

項目	規格
電氣	3.3 V；CS 低態有效 (模組內部上拉，飛控沒上電時不會浮接)；MOSI 未使用，主機送 0；SPI 關閉時 MISO 高阻抗
時序	Mode 3 (出廠，SPICK,3) 或 Mode 0 (SPICK,0)；MSB first；建議 SCK 2 MHz (已用 ArduPilot 驗證)
讀取	每次 CS 低態讀 128 bytes ；讀滿才算讀走，讀不滿下一次 CS 重給同一包；讀超過 128 bytes 補 0。沒有 DRDY 腳，主機以固定週期 (建議輸出週期的 0.9 倍) 讀取， pending > 0 時可立即再讀
佇列	16 包 (1000 Hz 時 16 ms)；主機沒讀走時丟最舊的

封包格式 (128 bytes，little-endian，SPI, ArduPilot)

offset	型別	內容
0	u8 × 2	0x53 0x4E ("SN")
2	u8	版本 = 1
3	u8	類型：0 = 空 (沒有新資料)、1 = IMU、2 = 串流

offset	型別	內容
4	u16	seq : IMU 包為輸出時間格序號 (每格 +1) ; 主機以 seq 偵測漏包 , 重複 seq 應忽略
6	u8	pending : 這包之後還排著幾包
7	u8	used_mask : bit n = IMU n+1 參與合併
8	u64	ts_us : 時間格結束時刻 (μs , 與 UART 輸出同一時間軸 ; PPS 對齊、外部觸發規則相同)
16	f32	dt (s)
20	f32 × 3 + f32 × 3	合併 $\Delta\theta$ (rad) / Δv (m/s) , 機體 FRD
44 / 68 / 92	各 f32 × 6	IMU 1 / 2 / 3 各自的 $\Delta\theta[3]$ 、 $\Delta v[3]$; 該顆這格缺席時全 0 且 status = 0xFF
116	i16 × 3	IMU 1 ~ 3 溫度 (0.01 °C)
122	u8 × 3	IMU 1 ~ 3 status (同 §6.5 , bit7 = PPS 標記)
125	u8	合併 flags (同 §6.3)
126	u16	CRC-16/CCITT-FALSE , 範圍 byte 0 ~ 125 (與 §7.1 同一個 CRC)

stream 模式 (類型 2) : byte 7 為有效長度 (0 ~ 118) , byte 8 起依序接起來就是與 UART2 相同的資料流 (SPIOUT 選內容、SPIMODE 選 ascii / binary) , 用 §6 / §7 的解析方式處理即可。

ArduPilot 設定步驟

步驟	設定
1. IMU600	GUI 參數設定頁「SPI3 輸出」用途選 ArduPilot、時脈模式選 3、輸出率選 1000 Hz、按「套用變更」後「寫入裝置 (SAVE)」 ; 或指令 SPI, ArduPilot、SPICK,3、RATE,1000、SAVE,1

步驟	設定
2. UART1	同時接 GUI 時，1000 Hz 的 ASCII 三顆資料遠超過 UART1 頻寬 (uart_drop 會增加、儀表板速率顯示不準，但不影響 SPI)；接飛控使用時建議 OUT,none、AHRS,0、U2OUT,none，需要看資料時再暫時打開
3. 飛控韌體	使用 SparkNavi 提供的 ArduPilot 驅動 (ArduPilot Copter 4.7.1，板型 SparkNavi-Blue2-IMU600)。SPI 時脈 2 MHz、Mode 3、獨立片選；IMU600 是飛控的第三顆 IMU
4. 飛控參數	EK3_IMU_MASK = 7：三顆 IMU 都給 EKF3 使用 (位元 0 / 1 / 2 = IMU 1 / 2 / 3，每顆各跑一個 EKF 核心)。ArduPilot 預設是 3，只用前兩顆，IMU600 (IMU 3) 不會進 EKF。SparkNavi 板型的預設參數已設為 7
5. 設為優先 IMU (選用)	EK3_PRIMARY = 2：EKF 核心與 IMU 的編號從 0 開始，Blue2 內建的兩顆是 0、1，IMU600 是 2，設 2 就讓 IMU600 成為第一優先 (預設 0 = 內建第一顆)。EKF3 以它為主核心，姿態、速度、位置與控制迴圈使用的陀螺都跟著主核心。需要 EK3_IMU_MASK 含 IMU600 (= 7) 且 INS_USE3 = 1。未解鎖時會固定用你指定的核心；飛行中若主核心的狀態明顯比其他核心差，EKF3 仍會自動切換到較好的核心，這是安全機制，請不要用 EK3_OPTIONS 關閉。若飛行中發生切換，地面站訊息會出現 EKF3 lane switch n。
6. 校正	在 Mission Planner 做加速度計六面校正 (三顆 IMU 一起校)。IMU600 內建的六面校正 (§5.3) 與 ArduPilot 的校正只做其中一邊，之後若改用 GUI 校正，ArduPilot 的要重做
7. 確認	Messages 出現 IMU600 found, 1000 Hz；參數出現 INS_ACC3_ID / INS_GYR3_ID；靜置水平時 Accel3 Z 約 -1000 mG、X / Y 約 0，轉動方向與 IMU 1 / 2 一致；EKF3 對準後 HUD 顯示 Ready to Arm

注意事項

- ArduPilot 驅動由 SparkNavi 提供，尚未併入 ArduPilot 主線。
- 偶發的 SPI 讀取錯誤 (CRC 錯) 由驅動自動處理 (模組重送該包)，不會在地面站顯示；只有短時間內大量錯誤才會顯示 IMU600: SPI link errors。

- IMU600 若由飛控的 5 V 供電，飛控重新開機時 5 V 可能短暫中斷；模組開機會先等電源穩定，且 SPI 錯誤持續時自動重新初始化 IMU (fw 0.10.1 起，\$SPNVHEAL 訊息)。
- 檢查 SPI 狀態：CFG,1 的 \$SPNVSPI 中 txn (CS 次數) 與 delivered (被讀走的資料包) 應持續增加，飛控開始讀取之後 drop 不應再增加。

9. MCLR 外部重置參考電路

把 J1 pin 17 (MCLR) 拉低再放開，模組就硬體重置。板上已有 10 k Ω (R4) 上拉到 3.3 V，所以客戶端只需要「拉低」的元件，**不要主動驅動高電位**。

建議做法 (擇一)：

方式	接法	適用
A. 按鈕	MCLR $\rightarrow$ R1 (470 Ω) $\rightarrow$ 常開按鈕 $\rightarrow$ GND	手動重置、測試治具
B. N-MOSFET 開汲極 (建議)	MCLR $\rightarrow$ R1 (100–470 Ω) $\rightarrow$ Q1 汲極 (2N7002 或同級)，源極接 GND；主機 GPIO $\rightarrow$ R2 (100 Ω) $\rightarrow$ 閘極，閘極加 R3 (100 k Ω) 下拉	主機 MCU / 飛控程式控制重置；主機是 3.3 V 或 5 V 都可用
C. 主機開汲極 GPIO	MCLR $\rightarrow$ R1 (470 Ω) $\rightarrow$ 主機設為 open-drain 的 GPIO	主機為 3.3 V 且 GPIO 支援 open-drain

MCLR 只需拉低：板上已上拉，客戶端串 R1 後接按鈕或 MOSFET

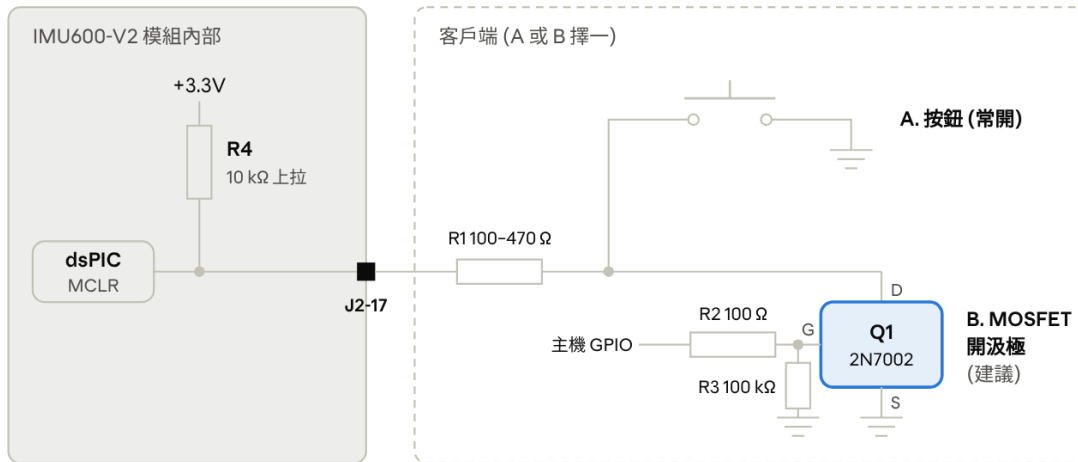


圖 7、MCLR 接法參考圖

MCLR 外部重置參考電路·按鈕或 MOSFET

板上 R4 負責上拉；客戶端只經 R1 把 pin 17 拉到地。B 案中主機 GPIO 輸出高電位 = 重置，低電位 = 正常執行。

設計要點

- **串聯電阻 R1 (100–470 Ω)：**限流並隔離外部連線的 ESD / 雜訊，必要。
- **不要用 5 V 驅動 MCLR，**也不要 push-pull 輸出驅動高電位；放開時交由板上 10 kΩ 上拉。
- **R3 閘極下拉：**主機上電中 / GPIO 浮接時保持 Q1 關閉，避免模組被一直重置。
- **脈衝寬度：**拉低 $\geq 10 \mu\text{s}$ 即可 (建議 1–10 ms)；放開後模組重新開機。**不要長時間保持低電位，**否則模組一直停在重置。
- **電容 (選用)：**連線長或雜訊大時，可在客戶端 R1 外側對地加 1–10 nF 濾波；不要直接在 MCLR 上加電容。
- 連線超過約 30 cm 或會接觸外界時，可在客戶端加 3.3 V TVS。
- 不使用外部重置時 pin 17 空接即可。

重置後的行為

- 放開後 bootloader 先等約 120 ms，再啟動主程式、初始化 IMU (約 250 ms)，接著恢復輸出。剛開機的幾筆 status 可能帶 32 (初始化中)。
- 重置後載入 flash 中已 SAVE 的設定；只用指令改、未 SAVE 的設定會回到已存值。
- ts_us 重新從 0 開始；AHRS 重新對準 (請靜置 1 秒)。
- \$SPNVCMDB,INFO,1 的 \$SPNVRESET 會顯示 EXTR(MCLR)，可確認是外部重置。
- 不需要硬體重置時，也可用指令 \$SPNVCMDB,BOOT,1 (或 GUI 終端機「BOOT (重新啟動)」) 軟體重開。

10. 韌體更新與救援

韌體以 SparkNavi 提供的加密簽章 .spfw 檔發佈，用 GUI 經 UART1 更新；參數設定與六面校正存在另一個 flash 區，更新不會清掉。

一般更新

1. GUI 連線到模組 (任何鮑率 / 模式皆可)。
2. 檔案 → 韌體更新... → 選 .spfw。視窗顯示檔案版本、適用硬體、打包時間與裝置目前版本。
3. 按「開始更新」。模組重開進入 bootloader，GUI 以 460800 傳送 (進度條)，驗證通過後自動回到主程式並重新連線。
4. 確認狀態列 / 裝置資訊的韌體版本為新版。

救援更新 (更新中斷、斷電，模組停在 bootloader、沒有任何輸出)

1. GUI 保持「中斷連線」狀態，選好 COM port。
2. 檔案 → 韌體更新... → 選 .spfw → 「救援更新」。
3. 30 秒內把模組斷電再上電 (或用 MCLR 重置)。GUI 在開機窗內接手並完成更新。

錯誤訊息	原因	處理
金鑰不符	檔案不是給這批模組的	向 SparkNavi 索取正確的 .spfw
硬體不符	檔案是其他產品的	同上

錯誤訊息	原因	處理
驗證失敗	檔案損壞或被修改	重新下載檔案，用救援更新再做一次 (模組會停在 bootloader 等待)
只能用 PICKit 燒錄	模組是開發版韌體	聯絡 SparkNavi

更新過程中請不要拔線、斷電；若發生了也不會變磚，用救援更新即可。

11. 疑難排解

現象	可能原因	處理
連不上、沒有任何資料	TX / RX 接反、沒共地、鮑率不對	檢查接線 (\$2)；GUI 「自動偵測鮑率」
收到亂碼	模組存成 binary 或其他鮑率	GUI 可直接解 binary；終端機軟體請先 MODE,ascii
改了設定重開後又回去	只套用沒存	按「寫入裝置 (SAVE)」或送 SAVE,1
完全不知道存了什麼設定	—	參數頁「開機救援」→ 20 秒內斷電再上電，回到出廠值
\$SPNVSYS uart_drop 一直增加	UART1 頻寬不足	降 RATE、改 OUT,fused、改 binary 或提高 BAUD
UART2 丟棄 (U2 n) 增加	UART2 頻寬不足	減少 U2OUT 內容、提高 U2BAUD 或降 RATE
status = 16 或合併 SATURATED	量程飽和	撞擊 / 甩動屬正常；運轉中持續出現請加減震
status = 32 / 44	感測器初始化中	開機後幾筆屬正常；運轉中出現請檢查供電是否掉壓

現象	可能原因	處理
status = 8 / 12	感測器自我診斷錯誤	REINIT,1；仍持續請聯絡 SparkNavi
status bit7 (128)	PPS 標記，不是錯誤	判斷資料好壞用 status & 0x7F
AHRS yaw 慢慢漂	沒有磁力計 / GPS 航向參考	正常；用 HDG 手動給航向，或主機端融合外部航向
AHRS flags bit4	開機對準時板子在動	靜置後 AHRS,reset
PPS 一直 NO_SIGNAL	沒接、電位不對、下降緣有效的 PPS	確認 3.3 V、上升緣有效、每秒一個
PPS 一直 ACQUIRING	PPS 間隔不穩 (不在 $1\text{ s} \pm 1000\text{ ppm}$)	確認 GPS 已定位、PPS 來源穩定；檢查接地與雜訊
模組一直沒反應、沒輸出	MCLR 被拉低或韌體更新中斷	檢查 pin 17 外部電路；用「救援更新」(\$10)
\$SPNVHALT	模組認證失敗	請聯絡 SparkNavi，提供 INFO,1 的輸出
外部觸發模式下沒有 IMU 輸出	沒有觸發訊號、只來了一個觸發 (第一個是起點)、電位不到 3.3 V	看系統卡「輸出模式」觸發數有沒有增加；確認 pin 15 是 3.3 V 上升緣；不用觸發時改回 TRIG,0
觸發數比輸出多、trig_rej 增加	觸發線上有彈跳或毛刺 (間隔 $< 0.5\text{ ms}$)	縮短連線、共地，觸發源用推挽輸出
ArduPilot : PreArm: Gyro n rate 400Hz < loop rate*1.8 720Hz	IMU600 輸出率不是 1000 Hz	RATE,1000 後 SAVE,1 (\$8.4)
ArduPilot 找不到 IMU600 (Messages 沒有 IMU600	SPI 沒開、接線錯、時脈模式不符	SPI, ArduPilot、SPICK,3、SAVE,1；檢查 pin 6 / 8 / 10 /

現象	可能原因	處理
found 或出現 IMU600 probe: ...)		12 與共地；看 \$SPNVSPI 的 txn 有沒有增加
ArduPilot：IMU600: SPI is in stream mode	SPI 設成 stream	改 SPI, ArduPilot
GUI 在 400 / 1000 Hz 時 uart_drop 增加、輸出率顯示 不準	UART1 頻寬不足 (OUT both / AHRS 全開)	不影響 SPI。接飛控時 OUT,none、AHRS,0；要看資 料請用 binary、1500000 以 上鮑率 (\$7.4)
pps_cnt 每秒增加，但 pps_state 一直是 2 (ACQUIRING)、沒有 \$SPNVPPS 或 bit7	PPS 來源頻率不準：相鄰邊緣 間隔超出 $1\text{ s} \pm 1000\text{ ppm}$ (例 如輸入的 PPS 實際：993.83 mHz，IMU 不接受這種精度)	用示波器量實際頻率並調到 1.000 Hz，或改用 GPS PPS (\$8.2)；STAT,1 後看 \$SPNVPPS

向 SparkNavi 回報問題時，請附上 \$SPNVCMDB,INFO,1 與 \$SPNVCMDB,CFG,1 的回覆，以及用 GUI
「開始錄製原始資料」錄的 .bin 檔。