

# CH110 使用說明書

IMU/VRU姿態測量模組, Rev 1.0



CH110 使用說明書

簡介

特性

板載傳感器

數據處理

通訊接口及供電

其他

硬體及尺寸

硬體參數

尺寸

接口定義

性能指標

姿態角輸出精度

陀螺儀

加速度計

模組數據接口參數

融合及校準算法

陀螺儀校準

參考系定義

串口通訊協議

數據包

數據包總覽

產品支持數據包列表

0x90(用戶ID)

0xA0(加速度)

0xB0(角速度)

0xC0(磁場強度)

0xD0(歐拉角)

0XD1(四元數)

0XF0(氣壓)

0X91( IMUSOL)

出廠默認數據包

數據幀結構示例

數據幀配置為 0x90,0xA0,0xB0,0xC0,0xD0,0xF0 數據包

數據幀配置為 0x91 數據包

CAN通訊協議

CANopen 默認設置

CANopen PTO傳輸細節

修改CAN接口配置

使能異步觸發數據輸出

修改CAN波特率

修改輸出速率

AT指令

產品支持數據包列表

AT+ID

AT+URFR

AT+INFO

AT+ODR

AT+BAUD

AT+EOUT

AT+RST

AT+TRG

AT+SETPTL

AT+MODE

AT+GWID

AT+GWCFG

附錄C - 韌體升級與恢復出廠設置

# 簡介

CH110是超核電子推出的一款超低成本、高性能、小體積、低延時的慣性測量單元(IMU)，本產品集成了三軸加速度計、三軸陀螺儀和一款微控制器。可輸出經過傳感器融合算法計算得到的基於當地地理坐標的三維方位數據，包含無絕對參考的相對航向角，俯仰角和橫滾角。同時也可以輸出校準過的原始的傳感器數據。

典型應用：

- 機器人航向追蹤/無人駕駛等

# 特性

## 板載傳感器

- 三軸陀螺儀, 最大量程:  $\pm 2000^{\circ}/s$
- 三軸加速度計, 最大量程: $\pm 8G$

## 數據處理

- 加速度和陀螺儀出廠前經過三軸非正交和標度因子校準
- 數據融合算法計算並輸出地理坐標系下的旋轉四元數及歐拉角等姿態資訊

## 通訊接口及供電

- RS232串列接口/CAN2.0總線
- 供電電壓：5-24V

## 其他

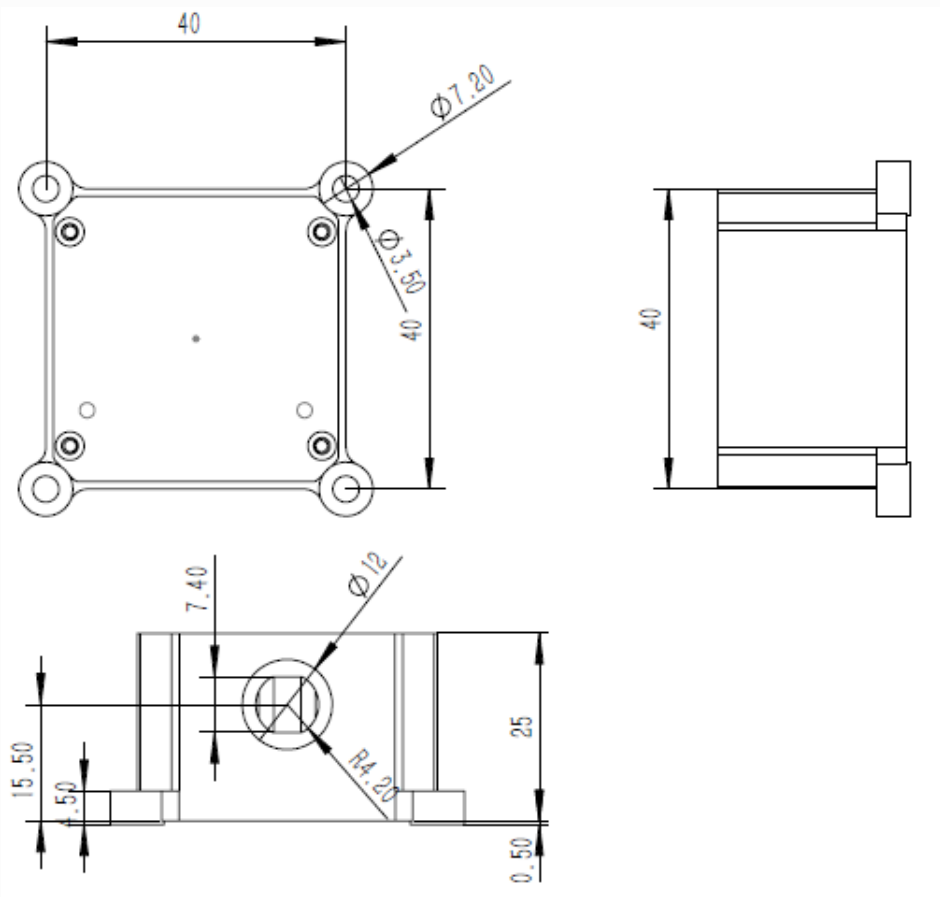
- PC端上位機程式，提供實時數據顯示，波形，校準及excel 數據記錄功能
- 多項模組參數用戶可配置

# 硬體及尺寸

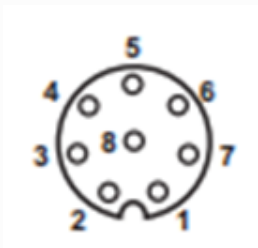
## 硬體參數

參數	描述
輸出數據接口	RS232串行接口
工作電壓	5-24V
溫度範圍	-20℃ - 85 ℃
最大線性加速度	0 - $115\ m/s^2$
尺寸	40 x 40 x 25mm (W x L x H)
板載傳感器	三軸加速度計 三軸陀螺儀

尺寸



接口定義



引腳號	序號	功能(RS232+CA)	功能(RS485+CAN)
紅	1	Vin	Vin
黑	2	GND	GND
藍	3	RS232 TX	485 A
灰	4	RS232 RX	485 B
白	5	同步輸出	同步輸出
棕	6	同步輸入	同步輸入
綠	7	CAN_H	CAN_H
黃	8	CAN_L	CAN_L

性能指標

## 姿態角輸出精度

姿態角	典型值
橫滾角\俯仰角 - 靜態誤差	0.8°
橫滾角\俯仰角 - 動態誤差	2.5°
零偏穩定性	10°/h

## 陀螺儀

參數	值
測量範圍	±2000°/s
非線性度	±0.1% (25°最佳)
噪聲密度	0.08°/s/ $\sqrt{Hz}$
採樣率	500Hz

## 加速度計

參數	值
測量範圍	±8G (1G = 1x 重力加速度 )
非線性度	±0.5% (25°最佳)
最大零點偏移	30mG
噪聲密度	250 $\mu G \sqrt{Hz}$
採樣率	500Hz

## 模組數據接口參數

參數	值
串口輸出波特率	9600/115200可選
幀輸出速率	10/50/100Hz 可選

## 融合及校準算法

### 陀螺儀校準

每一個姿態傳感器都單獨進行過全測量範圍內的校準和測試。陀螺和加速度計的非正交和刻度因子誤差參數都會保存在模組內部的Flash中。陀螺儀自動校準需要在上電後靜止模組3s 左右以獲得最好的校準效果。如果上電靜置短於規定時間，則模組陀螺儀零偏校準效果會下降。

姿態傳感器內建陀螺零速檢測機制，當檢測到長時間內三軸陀螺速度均小於1°/s時，模組認為當前為靜止狀態，陀螺輸出為零偏，此次模組會將此時的陀螺讀數記錄下來作為零偏補償。所以 **本產品不能用於旋轉速度<1°/s的運動場景。** (既旋轉速度低於秒針平均轉速的1/6)

## 參考系定義

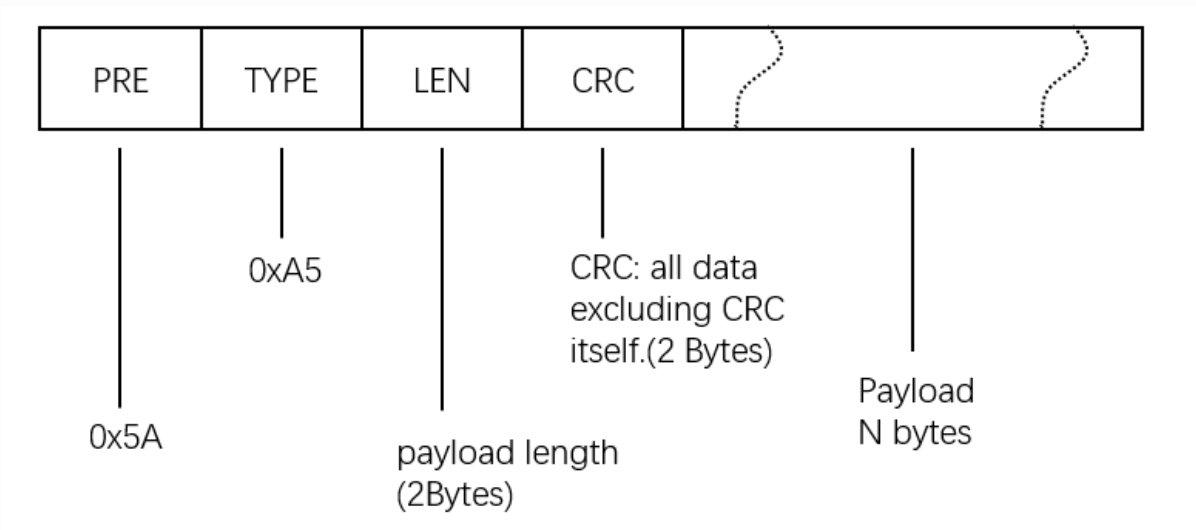
載體系使用 前-左-上(FLU)右手坐標系， 地理坐標系使用 北-西-天(NWU)坐標系。其中歐拉角旋轉順序為ZYX(先轉Z軸，再轉Y軸，最後轉X軸)旋轉順序。具體定義如下：

- 繞 Z 軸方向旋轉: 航向角\Yaw\phi( $\psi$ ) 範圍: -180° - 180°
- 繞 Y 軸方向旋轉: 俯仰角\Pitch\theta( $\theta$ ) 範圍: -90°-90°
- 繞 X 軸方向旋轉:橫滾角\Roll\psi( $\phi$ )範圍: -180°-180°

如果將模組視為飛行器的話。X 軸應視為機頭方向。當傳感器系與慣性系重合時，歐拉角的理想輸出為:Pitch = 0°, Roll = 0°, Yaw = 0°

## 串口通訊協議

模組上電後，模組默認按100Hz (出廠默認輸出速率) 輸出幀數據，幀格式如下：



其中：

域	值	長度 (字節)	說明
PRE	0x5A	1	固定為0x5A
TYPE	0xA5	1	固定為0xA5
LEN	1-512	2	幀中數據域的長度。LSB(低字節在前)，長度表示數據域的長度，不包含 PRE , TYPE , LEN , CRC 字段。
CRC	-	2	除CRC 本身外其餘所有幀數據的16 位CRC 校驗和。LSB(低字節在前)
PAYLOAD	-	1-512	一幀攜帶的數據。PAYLOAD 由若干個子數據包組成。每個數據包 包含：數據包標籤(DATA_ID)和數據(DATA) 兩部分。DATA_ID決定了數據的類型及長度，DATA 為數據包內容。

CRC實現函數：

```
1  /*
2      currentCrc: previous crc value, set 0 if it's first section
3      src: source stream data
4      lengthInBytes: length
5  */
6  static void crc16_update(uint16_t *currentCrc, const uint8_t *src, uint32_t
lengthInBytes)
7  {
8      uint32_t crc = *currentCrc;
9      uint32_t j;
10     for (j=0; j < lengthInBytes; ++j)
11     {
12         uint32_t i;
13         uint32_t byte = src[j];
14         crc ^= byte << 8;
15         for (i = 0; i < 8; ++i)
16         {
17             uint32_t temp = crc << 1;
18             if (crc & 0x8000)
19             {
20                 temp ^= 0x1021;
21             }
22             crc = temp;
23         }
24     }
25     *currentCrc = crc;
26 }
```

# 數據包

## 數據包總覽



數據包標籤(DATA_ID)	數據包長度(包含標籤1字節)	名稱	備註
0x90	2	用戶ID	
0xA0	7	加速度	
0xB0	7	角速度	
0xC0	7	磁場強度	
0xD0	7	歐拉角	
0xD1	17	四元數	
0xF0	5	氣壓	輸出0
0x91	76	IMUSOL(IMU數據集合)	推薦使用

## 產品支持數據包列表

下表列出所有產品支持的數據包, \* 表示支持 -表示不支持

產品	90	A0	B0	C0	D0	D1	F0	91
HI226	*	*	*	*	*	*	-	*
HI229	*	*	*	*	*	*	-	*
CH110	-	-	-	-	-	-	-	*

## 0x90(用戶ID)

共2字節，用戶設置的ID。

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0x90
1	uint8_t	1	-	用戶ID

## 0xA0(加速度)

共7 個字節，LSB。輸出傳感器的原始加速度

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0xA0
1	int16_t	2	0.001G(1G = 1重力加速度)	X軸加速度
3	int16_t	2	0.001G	Y軸加速度
5	int16_t	2	0.001G	Z軸加速度

## 0xB0(角速度)

共7字節，LSB。輸出傳感器的原始角速度

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤：0xB0
1	int16_t	2	0.1°/s	X軸角速度
3	int16_t	2	0.1°/s	Y軸角速度
5	int16_t	2	0.1°/s	Z軸角速度

0xC0(磁場強度)

共7字節，LSB。輸出傳感器的原始磁場強度

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0xC0
1	int16_t	2	0.001Gauss	X軸磁場強度
3	int16_t	2	0.001Gauss	Y軸磁場強度
5	int16_t	2	0.001Gauss	Z軸磁場強度

0xD0(歐拉角)

共7字節，LSB。格式為int16，共三個軸，每個軸占2 個字節，順序為Pitch/Roll/Yaw。接收到Roll, Pitch 為物理值乘以100 後得到的數值，Yaw 為乘以10 得到的數值。

例：當接收到的Yaw = 100 時，表示航向角為10°

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0xD0
1	int16_t	2	0.01°	Pitch(俯仰角)
3	int16_t	2	0.01°	Roll(橫滾角)
5	int16_t	2	0.1°	Yaw(航向角)

0xD1(四元數)

共17字節，格式為float，共4個值，順序為:W X Y Z.。每個值占4 字節(float)，整個四元數為4個float，LSB。

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0xD1
1	float	4	-	W
5	float	4	-	X
9	float	4	-	Y
13	float	4	-	Z

### 0XF0(氣壓)

共5字節，格式為float。(只針對有氣壓傳感器的產品)

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0xF0
1	float	4	Pa	大氣壓

### 0X91( IMUSOL)

共76字節，新加入的數據包，用於替代A0,B0,C0,D0,D1等數據包。集成了IMU的傳感器原始輸出和姿態解算數據。

字節偏移	類型	大小	單位	說明
0	uint8_t	1	-	數據包標籤:0x91
1	uint8_t	1	-	ID
2	-	6	-	保留
8	uint32_t	4	ms	時間戳資訊，從系統開機開始累加，每毫秒增加1
12	float	12	1G(1G = 1重力加速度)	X,Y,Z軸的加速度，注意單位和0xA0不同
24	float	12	deg/s	X,Y,Z軸的角速度，注意單位和0xB0不同
36	float	12	uT	X,Y,Z軸的磁場強度(HI229支持,注意單位和0xC0不同)
48	float	12	deg	節點歐拉角集合，順序為：橫滾角(Roll)，俯仰角(Pitch)，航向角(Yaw)(注意順序和單位與0xD0數據包不同)
60	float	16	-	節點四元數集合,順序為WXYZ

### 出廠默認數據包

出廠默認一幀中攜帶數據包數據定義如下：

產品	默認輸出數據包
HI226	90,A0,B0,C0,D0,F0
HI229	90,A0,B0,C0,D0,F0
CH110	90,A0,B0,C0,D0,F0

## 數據幀結構示例

### 數據幀配置為 0x90,0xA0,0xB0,0xC0,0xD0,0xF0 數據包

使用串口助手採樣一幀數據,共41字節,前6字節為幀頭,長度和CRC校驗值。剩餘35字節為數據域。假設數據接收到C語言數組 `buf` 中。如下所示:

5A A5 23 00 FD 61 **90** 00 **A0** 55 02 3D 01 E2 02 **B0** FE FF 17 00 44 00 **C0** 80 FF 60 FF 32 FF **D0** 64 F2 6C 0E BB 01 **F0** 00 00 00 00

- 第一步: 判斷幀頭, 得到數據域長度和幀CRC:

幀頭: 5A A5

幀數據域長度: 23 00 :  $(0x00 \ll 8) + 0x23 = 35$

幀CRC校驗值: FD 61 :  $(0x61 \ll 8) + 0xFD = 0x61FD$

- 第二步: 校驗CRC

```
1    uint16_t payload_len;  
2    uint16_t crc;  
3  
4    crc = 0;  
5    payload_len = buf[2] + (buf[3] << 8);  
6  
7    /* calculate 5A A5 and LEN filed crc */  
8    crc16_update(&crc, buf, 4);  
9  
10   /* calculate payload crc */  
11   crc16_update(&crc, buf + 6, payload_len);
```

得到CRC值為0x61FD, 與幀攜帶的CRC值相同, 幀CRC校驗通過。

- 第三步: 接收數據

90 00 : ID 數據包, 0x90為數據包標籤, ID = 0x00.

A0 55 02 3D 01 E2 02 :加速度數據包,0xA0為數據包標籤, 三軸加速度為:

X軸加速度 =  $(\text{int16\_t})((0x02 \ll 8) + 0x55) = 597$ (單位為mG)

Y軸加速度 =  $(\text{int16\_t})((0x01 \ll 8) + 0x3D) = 317$

Z軸加速度 =  $(\text{int16\_t})((0x02 \ll 8) + 0xE2) = 738$

B0 FE FF 17 00 44 00 :角速度數據包,0xB0為數據包標籤, 三軸角速度為:

X軸角速度 =  $(\text{int16\_t})((0xFF \ll 8) + 0xFE) = -2$ (單位為0.1°/s)

Y軸角速度 =  $(\text{int16\_t})((0x00 \ll 8) + 0x17) = 23$

Z軸角速度 =  $(\text{int16\_t})((0x00 \ll 8) + 0x44) = 68$

C0 80 FF 60 FF 32 FF :磁場數據包,0xC0為數據包標籤, 三軸磁場為:

X軸角速度=  $(\text{int16\_t})((0xFF \ll 8) + 0x80) = -128$  (單位為0.001Gauss)

Y軸角速度 =  $(\text{int16\_t})((0xFF \ll 8) + 0x60) = -160$

Z軸角速度=  $(\text{int16\_t})((0xFF \ll 8) + 0x32) = -206$

**D0 64 F2 6C 0E BB 01** 歐拉角數據包, 0xD0為數據包標籤

Pitch=  $(\text{int16\_t})((0xF2 \ll 8) + 0x64) / 100 = -3484 / 100 = -34.84^\circ$

Roll=  $(\text{int16\_t})((0x0E \ll 8) + 0x6C) / 100 = 3692 / 100 = 36.92^\circ$

Yaw =  $(\text{int16\_t})((0x01 \ll 8) + 0xBB) / 10 = 443 / 10 = 44.3^\circ$

**F0 00 00 00 00** 氣壓數據包, 0xF0為數據包標籤

```
1 float prs;
2 prs = memcpy(&prs, &buf[37], 4);
```

最後得到結果：

1	id	:	0		
2	acc(G)	:	0.597	0.317	0.738
3	gyr(deg/s)	:	-0.200	2.300	6.800
4	mag(uT)	:	-12.800	-16.000	-20.600
5	eul(R/P/Y)	:	36.920	-34.840	44.300

## 數據幀配置為 **0x91** 數據包

使用串口助手採樣一幀數據,共82字節, 前6字節為幀頭, 長度和CRC校驗值。剩餘76字節為數據域。假設數據接收到C語言數組 **buf** 中。如下所示:

5A A5 4C 00 6C 51 **91** 00 A0 3B 01 A8 02 97 BD BB 04 00 9C A0 65 3E A2 26 45 3F 5C E7 30 3F E2 D4 5A C2 E5 9D A0 C1 EB 23 EE C2 78 77 99 41 AB AA D1 C1 AB 2A 0A C2 8D E1 42 42 8F 1D A8 C1 1E 0C 36 C2 E6 E5 5A 3F C1 94 9E 3E B8 C0 9E BE BE DF 8D BE

- 第一步：判斷幀頭，得到數據域長度和幀CRC：

幀頭: **5A A5**

幀數據域長度: **4C 00** :  $(0x00 \ll 8) + 0x4C = 76$

幀CRC校驗值: **6C 51** :  $(0x51 \ll 8) + 0x6C = 0x516C$

- 第二步：校驗CRC

```
1 uint16_t payload_len;
2 uint16_t crc;
3
4 crc = 0;
5 payload_len = buf[2] + (buf[3] << 8);
6
7 /* calculate 5A A5 and LEN filed crc */
8 crc16_update(&crc, buf, 4);
9
10 /* calculate payload crc */
11 crc16_update(&crc, buf + 6, payload_len);
```

得到CRC值為0x516C. 幀CRC校驗通過。

- 第三步：接收數據

從 **0x91** 開始為數據包的數據域。在C語言中可以定義結構體來方便的讀取數據：

定義0x91數據包結構體如下：

```
1  __packed typedef struct
2  {
3      uint8_t    tag;           /* data packet tag */
4      uint8_t    id;
5      uint8_t    rev[6];       /* reserved */
6      uint32_t   ts;           /* timestamp */
7      float      acc[3];
8      float      gyr[3];
9      float      mag[3];
10     float      eul[3];       /* euler angles: Roll,Pitch,Yaw */
11     float      quat[4];      /* quaternion */
12 }id0x91_t;
```

**\_\_packed** 為編譯器關鍵字(Keil下)，表示結構體按字節緊對齊，結構體每一個元素一一對應0x91數據包的結構定義。接收數據時將接收到的數組直接memcpy到結構體即可：(注意定義結構體時必須4字節對齊), 其中 **buf** 指向幀頭, **buf[6]** 指向幀中數據域。

```
1  /* 接收數據並使用0x91數據包結構定義來解釋數據 */
2  __align(4) id0x91_t dat;    /* struct must be 4 byte aligned */
3  memcpy(&dat, &buf[6], sizeof(id0x91_t));
```

最後得到dat數據結果：

1	id	:	0			
2	timestamp	:	310205			
3	acc	:	0.224	0.770	0.691	
4	gyr	:	-54.708	-20.077	-119.070	
5	mag	:	19.183	-26.208	-34.542	
6	eul(R/P/Y)	:	48.720	-21.014	-45.512	
7	quat	:	0.855	0.310	-0.310	-0.277

## CAN通訊協議

模組CAN接口遵循以下標準：

- CAN接口符合CANopen協議，所有通訊均使用標準數據幀
- 只使用PTO1-4 傳輸姿態數據，所有傳輸均採用標準數據幀，不接收遠程幀和拓展數據幀。
- PTO採用異步定時觸發模式, 默認輸出速率為20Hz
- 當模組上電時，按照CANopen協議，模組會主動發送一條(一次)節點上線報文。節點上電處於預操作狀態(pre-operational) 需要主機發送NMT協議將節點設置為operation狀態才會開始發送數據

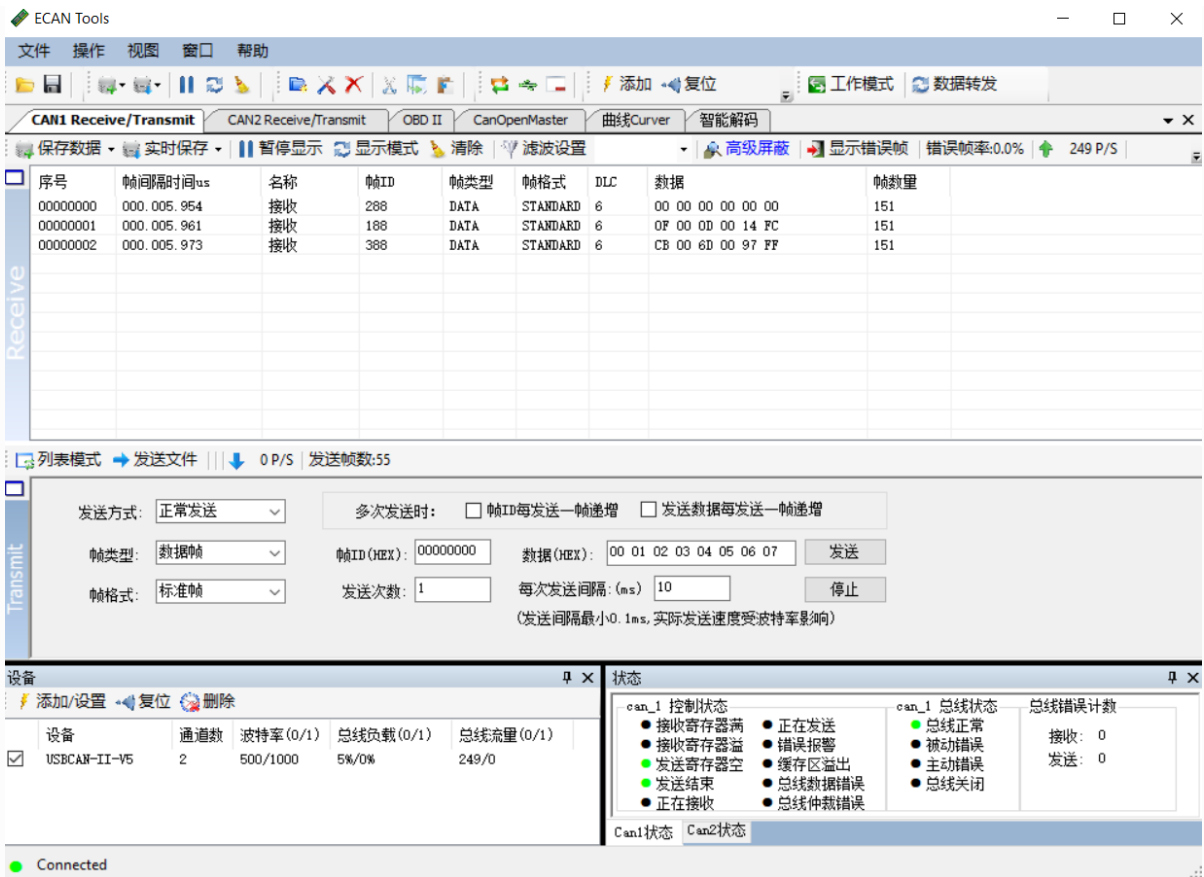
## CANopen 默認設置

CANopen 默認配置	值
CAN 波特率	500KHz
CANopen 節點ID	8
初始化狀態	預操作(Pre-operational),需要發送Start Remote Node命令模組才能開始輸出數據
心跳包	無

## CANopen PTO 傳輸細節

PTO 通道	PTO 幀 ID	長度	PTO 傳輸方式	發送數據	說明
TPDO1	0x180+ID	6	循環同步 (0x01)	加速度	每軸數據類型為(INT16,低字節在前), 分別為X,Y,Z軸加速度, 單位為mG(0.001重力加速度)
TPDO2	0x280+ID	6	循環同步 (0x01)	角速度	每軸數據類型為(INT16,低字節在前), 分別為X,Y,Z軸角速度, 單位為0.1DPS(°/s)
TPDO3	0x380+ID	6	異步定時 (0xFE)	歐拉角	每軸數據類型為(INT16,低字節在前), 順序分別為橫滾角(Roll,繞X軸旋轉),俯仰角(Pitch,繞Y軸旋轉),航向角(Yaw繞Z軸旋轉)。歐拉角單位為0.01°
TPDO4	0x380+ID	8	循環同步 (0x01))	四元數	每軸數據類型為(INT16,低字節在前), 分別為 $q_w$ $q_x$ $q_y$ $q_z$ 。單位四元數擴大10000倍後結果。如四元數為1,0,0,0 時, 輸出10000,0,0,0.

使用USB-CAN工具抓取默認CAN輸出包截圖如下：



其中 歐拉角(PTO3) CAN帧ID =0x380 + 8(默认ID) = 0x388， 數據為:

- X軸:  $(0x00 \ll 8) + 0x8F = 0x008F = 1.43^\circ$
- Y軸:  $(0xFF \ll 8) + 0xCF = 0xFFCF = -0.49^\circ$
- Z軸:  $(0x0F \ll 8) + 0xB9 = 0x0FB9 = 40.25^\circ$

## 修改CAN接口配置

數據字典以下位置存放廠商參數配置數據,可通過CANopen主站修改,掉電保存,重啟生效

數據字典位置	名稱	值類型	默認值	說明
0x2100	CAN_BAUD	INTEGER32	500000	CAN總線波特率
0x2101	NodeID	INTEGER32	8	節點ID

## 使能異步觸發數據輸出

發送標準CANopen協議幀,使用NMT: Start Remote Node命令:

ID=0x000, DLC=2, DATA=0x01, 0x08

其中 0x01為Start Remote Node指令, 0x08為節點ID

## 修改CAN波特率

發送標準CANopen協議幀,使用標準快速SDO指令

如將CAN波特率修改為125K, 則發送:

ID=0x608, DLC=8, DATA=0x23, 0x00, 0x21, 0x00, 0x48, 0xE8, 0x01, 0x00 (ID=0x608, 長度為8的標準數據幀)



其中  $0x01, 0xE8, 0x48 = (0x01 \ll 16) + (0xE8 \ll 8) + 0x48 = 125000$ , 注意發送ID為 $(0x600 + ID)$ , 寫SDO命令幀)

如將CAN波特率修改為250K, 則發送:

ID=0x608 , DLC=8, DATA=0x23, 0x00, 0x21, 0x00, 0x90, 0xD0, 0x03, 0x00

其中  $0x03, 0xD0, 0x90 = (0x03 \ll 16) + (0xD0 \ll 8) + 0x90 = 250000$

## 修改輸出速率

發送標準CANopen協議幀, 使用標準快速SDO指令:

如修改TPDO3(歐拉角)輸出速率為100Hz:(10ms event觸發)

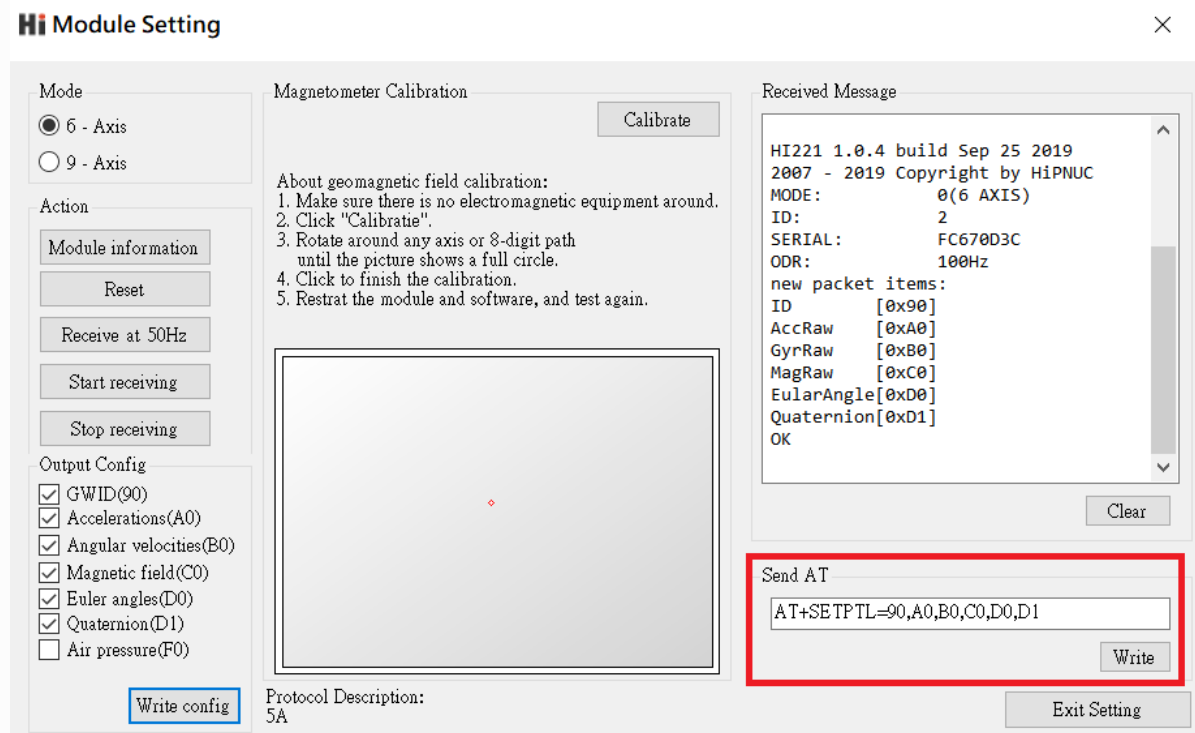
ID=0x608 , DLC=8, DATA=0x2B, 0x02, 0x18, 0x05, 0x0A, 0x00, 0x00, 0x00

其中  $0x00, 0x0A = (0x00 \ll 8) + 0x0A = 10$ (10ms觸發=100Hz)

## AT指令

當使用串口與模組通訊時, 模組支持AT 指令集配置/查看模組參數。AT 指令總以ASCII 碼 **AT** 開頭, 後面跟控制字符, 最

後以回車換行 `\r\n` 結束。可使用串口調試助手進行測試:



通用模組 AT指令如下

指令	功能	掉電保存(Y)	立即生效(Y),復位生效(R)
AT+ID	設置模組用戶ID	Y	R
AT+URFR	旋轉模組傳感器坐標系	Y	R
AT+INFO	打印模組資訊	N	Y
AT+ODR	設置模組串口輸出幀頻率	Y	R
AT+BAUD	設置串口波特率	Y	R
AT+EOUT	數據輸出開關	N	Y
AT+RST	復位模組	N	Y
AT+TRG	單次輸出觸發	N	Y
AT+SETPTL	設置輸出數據包	Y	Y
AT+MODE	設置模組工作模式	Y	R
AT+GWID	設置無線網關ID	Y	R
AT+GWCFG	設置接收機無線網絡屬性	Y	R

## 產品支持數據包列表

下表列出所有產品支持的數據包, \* 表示支持 -表示不支持

產品	ID	URFR	INFO	ODR	BAUD	EOUT	RST	TRG	SETPTL	MODE	GWID	GWCFG
HI226	*	*	*	*	*	*	*	*	*	-	-	-
HI229	*	*	*	*	*	*	*	*	*	*	-	-
CH110	*	*	*	*	*	*	*	-	*	-	-	-
HI221	*	*	*	*	*	*	*	-	-	-	*	-
HI221GW	*	*	*	*	*	*	*	-	-	-	*	*

### AT+ID

設置模組用戶ID

例 AT+ID=1

### AT+URFR

某些情況下傳感器需要傾斜垂直安裝，這時候需要旋轉傳感器坐標系，這條指令提供了旋轉傳感器坐標系的接口：

AT+URFR=C00,C01,C02,C10,C11,C12,C20,C21,C22

其中  $C_{nn}$  支持浮點數

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_U = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B$$

其中  $\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_U$  為旋轉後的 傳感器坐標系下 傳感器數據， $\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B$  為旋轉前 傳感器坐標系下 傳感器數據

下面是幾種常用旋轉舉例：

- 新傳感器坐標系為 繞原坐標系X軸 旋轉 90°，輸入命令：`AT+URFR=1,0,0,0,0,1,0,-1,0`
- 新傳感器坐標系為 繞原坐標系X軸 旋轉-90°，輸入命令：`AT+URFR=1,0,0,0,0,-1,0,1,0`
- 新傳感器坐標系為 繞原坐標系X軸 旋轉180°，輸入命令：`AT+URFR=1,0,0,0,-1,0,0,0,-1`
- 新傳感器坐標系為 繞原坐標系Y軸 旋轉 90°，輸入命令：`AT+URFR= 0,0,-1,0,1,0,1,0,0`
- 新傳感器坐標系為 繞原坐標系Y軸 旋轉-90°，輸入命令：`AT+URFR= 0,0,1,0,1,0,-1,0,0`
- 新傳感器坐標系為 繞原坐標系Y軸 旋轉180°，輸入命令：`AT+URFR= -1,0,0,0,1,0,0,0,-1`
- 恢復默認值：`AT+URFR=1,0,0,0,1,0,0,0,1`

## AT+INFO

打印模組資訊，包括產品型號，版本，韌體發佈日期等。AT+INFO可以拓展二級指令實現更多資訊的查詢

INFO二級拓展指令	功能	示例
CAL	顯示模組內部校準參數	AT+INFO=CAL
RF	顯示無線設備參數	AT+INFO=RF
VER	顯示詳細版本資訊	AT+INFO=VER

## AT+ODR

設置模組串口輸出速率。掉電保存，復位模組生效

例 設置串口輸出速率為100Hz: `AT+ODR=100`

## AT+BAUD

設置串口波特率，可選值：4800/9600/115200/256000/460800`

例 `AT+BAUD=115200`

!!! note "注意"

- 使用此指令需要特別注意，輸入錯誤波特率後可能會導致無法和模組通訊
- 波特率參數設置好後掉電保存，復位模組生效。上位機的波特率也要做相應修改。
- 升級韌體時，需要切換回115200 波特率。

## AT+EOUT

串口輸出開關

例 打開串口輸出 `AT+EOUT=1` 關閉串口輸出 `AT+EOUT=0`

## AT+RST

復位模組

例 AT+RST

## AT+TRG

觸發模組輸出一幀數據，可以配合AT+ODR=0來實現單次觸發輸出。

例 AT+TRG

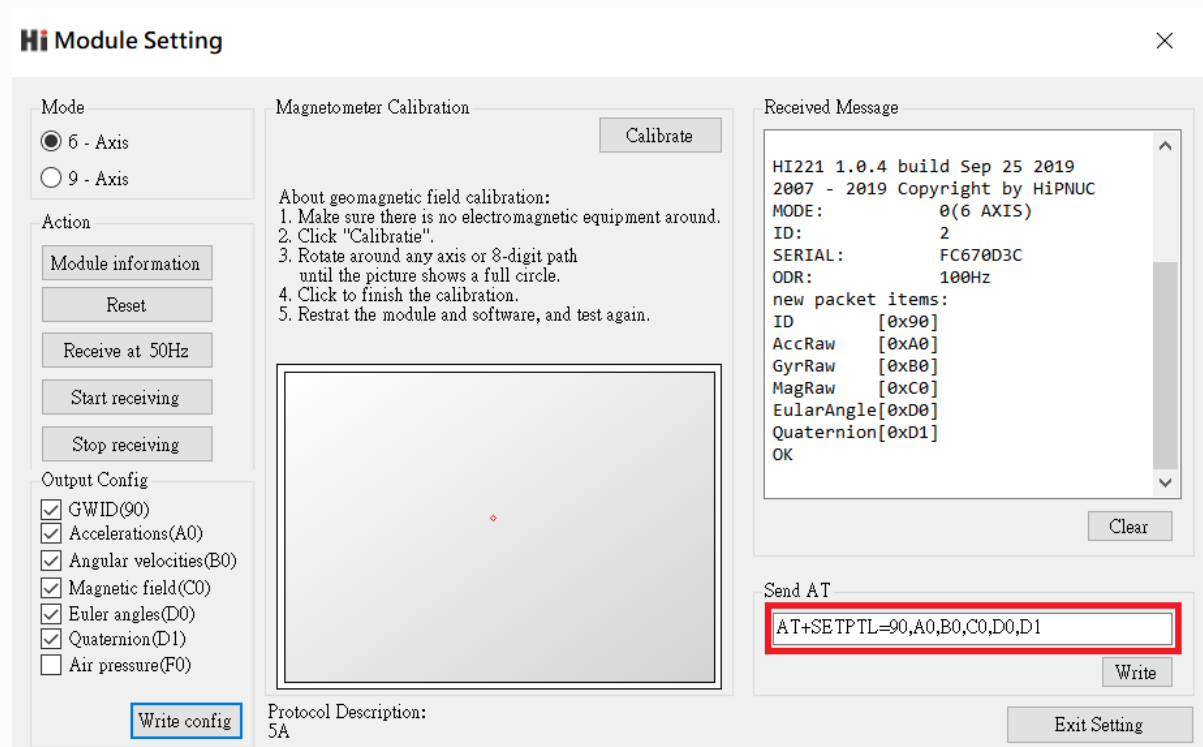
## AT+SETPTL

設置輸出協議:

模組數據幀中的數據包組成可使用AT指令配置，格式為 `AT+SETPTL=<ITEM_ID>,<ITEM_ID>...`

一幀輸出可包含最多8個數據包。

例 配置模組輸出加速度，角速度，整形格式歐拉角和四元數的指令為： `AT+SETPTL=A0,B1,D0,D1`



## AT+MODE

設置模組工作模式

例

- 設置模組工作在6軸模式(無磁校準) `AT+MODE=0`
- 設置模組工作在9軸模式(地磁場傳感器參與航向角校正) `AT+MODE=1`

## AT+GWID

可通過AT+GWID指令配置，GWID屬性決定了接收器和節點的RF頻率，只有節點的GWID 和接收器的GWID相同時，模組和接收器直接才能通訊。GWID相當於無線網段，當在同一地點使用多個接收機組成多個星形網絡時，必須保證每個接收器的GWID(網段)不同。

例 將一個接收器設置為GWID=3， 並將3個模組的自身ID設置為 0,1,2 並連接到這個接收器上：

接收機配置：

AT+GWID=3

節點0配置：

AT+GWID=3

AT+ID=0

節點1配置：

AT+GWID=3

AT+ID=1

節點2配置：

AT+GWID=3

AT+ID=2

最後所有接收機節點復位/重新上電生效。

## AT+GWCFG

配置接收機支持的節點數和無線通訊頻率。接收機默認支持8個節點，每個節點100Hz通訊頻率。(通訊頻率乘以節點)的乘積受到RF帶寬的限制，錯誤的配置將會導致無法輸出數據。推薦的配置如下：

配置選項	通訊頻率(Hz,每個節點)	支持的節點數
1(出廠默認)	100	8
2	200	4
3	30	16

例 配置接收機支持的最大節點數為16，每個節點接收頻率為30Hz，依次輸入：

AT+GWCFG=FRQ,30

AT+GWCFG=CNT,16

## 附錄C - 韌體升級與恢復出廠設置

本產品支持升級韌體。

韌體升級步驟：

- 連接模組，打開上位機，將模組和上位機波特率都設置為115200. 打開韌體升級窗口

- 點擊連接按鈕，如出現模組連接資訊。則說明升級系統準備就緒，點擊文件選擇器(...)選擇拓展名為.hex 的韌體，然後點擊開始編程。下載完成後會提示編程完成，此時關閉串口，重新給模組上電，模組升級完成。

## Hi Firmware Updater



File

h100\mdk\ch100\ch100.hex

..

Connect

Action

Write

Reset

Message

Protocol version: P1.2.0  
Program Version:K1.4.0  
Maximum packet length:256  
Flash size: 480KB  
DeviceID: 0x52832  
Flash sector: 32768  
connect successfully