

CH110 用户手册

IMU/VRU姿态测量模块, Rev 1.0



CH110 用户手册

简介

特性

板载传感器

数据处理

通讯接口及供电

其他

硬件及尺寸

硬件参数

尺寸

接口定义

坐标系定义

性能指标

姿态角输出精度

陀螺仪

加速度计

磁传感器参数

模块数据接口参数

传感器校准

地磁校准

磁干扰分类

6轴和9轴模式区别

校准方法

串口通讯协议

数据包

数据包总览

产品支持数据包列表

0x90(用户ID)

0xA0(加速度)

0xB0(角速度)

0xC0(磁场强度)

0xD0(欧拉角)

0xD1(四元数)

0xF0(气压)

0x91(IMUSOL)

出厂默认数据包

数据帧结构示例

数据帧配置为 0x90, 0xA0, 0xB0, 0xC0, 0xD0, 0xF0 数据包

数据帧配置为 0x91 数据包

CAN通讯协议

CANopen 默认设置

CANopen PTO传输细节

修改CAN接口配置

1. 使能数据输出(开启异步触发)

2. 修改CAN波特率

3. 修改输出速率

AT指令

AT+ID

AT+INFO

AT+ODR

AT+BAUD

AT+EOUT

AT+RST

AT+SETPTL

AT+MODE

附录C - 固件升级与恢复出厂设置

简介

CH110是超核电子推出的一款超低成本、高性能、小体积、低延时的惯性测量单元(IMU)，本产品集成了三轴加速度计、三轴陀螺仪和一款微控制器。可输出经过传感器融合算法计算得到的基于当地地理坐标的三维方位数据，包含无绝对参考的相对航向角，俯仰角和横滚角。同时也可以输出校准过的原始的传感器数据。

典型应用:

- 机器人航向跟踪/无人驾驶等

特性

板载传感器

- 三轴陀螺仪, 最大量程: $\pm 2000^\circ/\text{s}$
- 三轴加速度计, 最大量程: $\pm 8\text{G}$

数据处理

- 加速度和陀螺仪出厂前经过三轴非正交和标度因子校准
- 数据融合算法计算并输出地理坐标系下的旋转四元数及欧拉角等姿态信息

通讯接口及供电

- RS232串行接口/CAN2.0总线
- 供电电压: 5-24V

其他

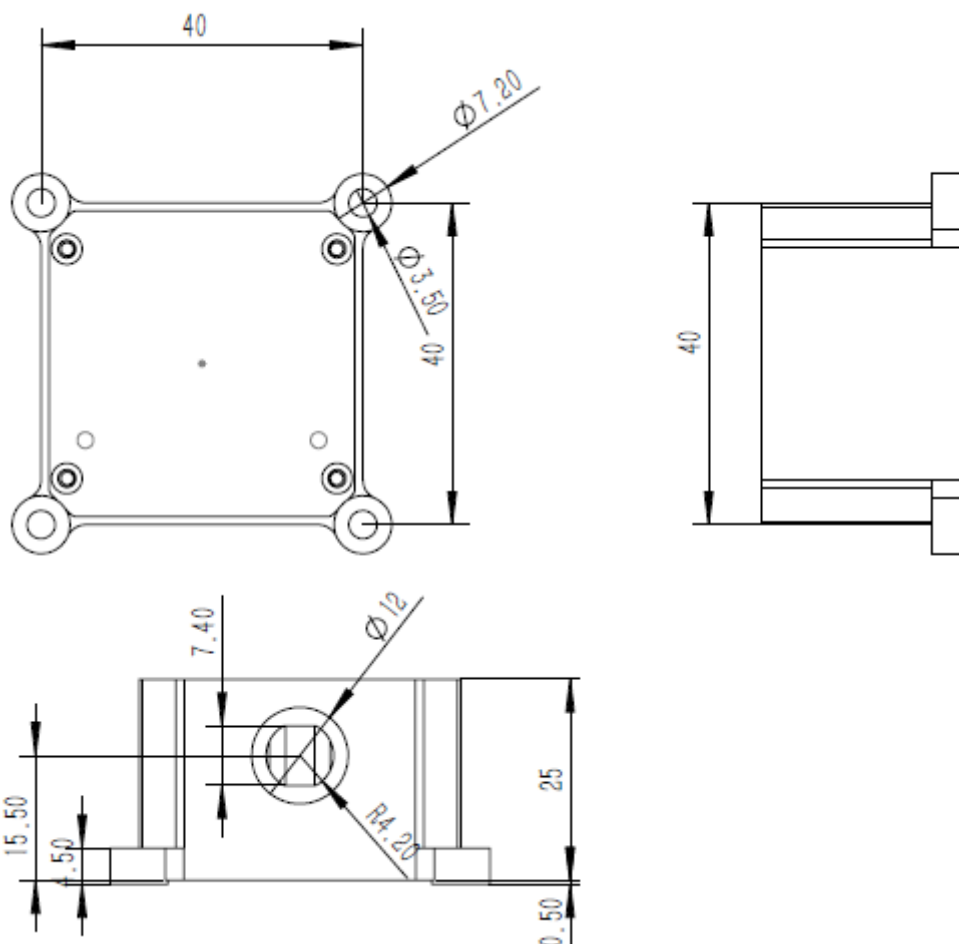
- PC端上位机程序，提供实时数据显示，波形，校准及excel 数据记录功能
- 多项模块参数用户可配置

硬件及尺寸

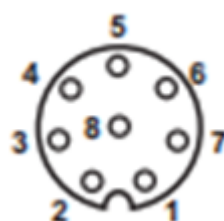
硬件参数

参数	描述
输出数据接口	RS232串行接口
工作电压	5-24V
温度范围	-20°C - 85 °C
最大线性加速度	0 - 115 m/s^2
尺寸	40 x 40 x 25mm (W x L x H)
板载传感器	三轴加速度计 三轴陀螺仪 三轴磁传感器

尺寸



接口定义



序号	引脚号	功能
1	红	Vin
2	黑	GND
3	黄	RS232 TX
4	绿	RS232 RX
5	白	保留
6	棕	保留
7	蓝	CAN_H
8	灰	CAN_L

坐标系定义

载体体系使用 前-左-上(FLU)右手坐标系，地理坐标系使用 北-西-天(NWU)坐标系。其中欧拉角旋转顺序为 ZYX(先转Z轴，再转Y轴，最后转X轴)旋转顺序。具体定义如下：

- 绕 Z 轴方向旋转: 航向角\Yaw\phi(ψ) 范围: -180° - 180°
- 绕 Y 轴方向旋转: 俯仰角\Pitch\theta(θ) 范围: -90° - 90°
- 绕 X 轴方向旋转:横滚角\Roll\psi(ϕ)范围: -180° - 180°

如果将模块视为飞行器的话。X 轴应视为机头方向。当传感器系与惯性系重合时，欧拉角的理想输出为:Pitch = 0°, Roll = 0°, Yaw = 0°

性能指标

姿态角输出精度

姿态角	典型值
横滚角\俯仰角 - 静态误差	0.8°
横滚角\俯仰角 - 动态误差	2.5°
运动中航向角精度(9轴模式下,无磁干扰,校准后)	3°

陀螺仪

参数	值
测量范围	±2000°/s
零偏稳定性	10°/h
刻度非线性度	±0.1%(满量程时)
噪声密度	0.08°/s/ $\sqrt{\text{Hz}}$
加速度敏感性	0.001°/s/g

加速度计

参数	值
测量范围	±8G (1G = 1x 重力加速度)
零偏稳定性	30mG
非线性度	±0.5% (满量程时)
噪声密度	120 $\mu\text{G}\sqrt{\text{Hz}}$

磁传感器参数

参数	值
测量范围	±8G(Gauss)
非线性度	±0.1%
分辨率	0.25mG

模块数据接口参数

参数	值
串口输出波特率	9600/115200/460800/921600可选
帧输出速率	1/25/50/100Hz 可选

传感器校准

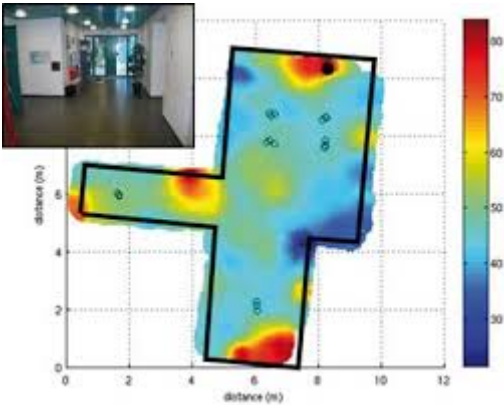
- 每一个姿态传感器都单独进行过全测量范围内的校准和测试。陀螺和加速度计的非正交和刻度因子误差参数都会保存在模块内部的Flash中。
- 陀螺仪自动校准需要在上电后静止模块5s 左右以获得最好的校准效果。如上电**5s**内模块处于非静止状态或者缓慢转动状态则陀螺性能严重下降。

地磁校准

磁干扰分类

种类	定义	典型干扰源	影响	措施
空间磁场干扰	干扰不随传感器运动而运动，而处于世界坐标系下	各种固定的磁干扰源，家具，家用电器，电缆，房屋内的钢筋结构等。一切不随磁传感器运动而运动的干扰源	无论磁场传感器是否校准的好，这些空间磁场的干扰(或者说环境磁场不均匀)都会使得空间地磁场发生畸变。地磁补偿会错误并且无法获得正确的航向角。他们是造成室内地磁融合难以使用的主要元凶。这种干扰不能被校准,会严重影响地磁性能。空间磁场干扰在室内尤其严重。	模块内置的匀质磁场检测及屏蔽非匀质磁场
传感器坐标系下的干扰	干扰源随传感器运动而运动	模块PCB，与模块固定在一起的板子，仪器设备，产品等。他们和磁传感器视为同一个刚体，随磁传感器运动而运动	对传感器造成硬磁/软磁干扰。这些干扰可以通过地磁校准算法加以很好的消除。	地磁校准

下图是一个典型的室内磁场分布图。可以看到：一般室内环境的空间磁场畸变是比较严重的。



注意

在室内环境下，空间磁场干扰尤其严重，而且空间磁干扰并不能通过校准来消除。在室内环境下，尽管模块内置均质磁场检测及屏蔽机制，但9轴模式航向角的准确度很大程度上取决于室内磁场畸变程度，如果室内磁场环境很差(如电脑机房旁，电磁实验室，车间，地下车库等等)，即使校准后，9轴的航向角精度可能还不如6轴甚至会出现大角度误差。

6轴和9轴模式区别

正因为地磁场非常容易受到空间干扰，所以使用9轴模式时应非常注意。下表列举了不同的使用场合和工况下的使用建议

模式	适用环境	典型应用	优点	缺点	注意事项
6轴模式	各种环境	云台等低动态姿态检测，室内机器人	1. 姿态角输出稳定性好 2. 完全不受磁场干扰	航向角随时间缓慢漂移	航向角会随时间缓慢漂移且无法补偿
9轴模式	无磁干扰环境	1. 指南针，寻北系统 2. 空旷且磁干扰较少的室内，模块基本不会大范围在室内移动(典型的如摄影棚内动作捕捉，且被测者不会做大范围走动)	1. 航向角不会随时间漂移 2. 一旦检测到地磁场可快速修正航向角指北	任何磁干扰都会出现航向角准确度下降。室内干扰严重情况下航向角无法指向正确方向。另外，移动机器人的金属结构和电机运行时会产生非常强的磁干扰，所以移动机器人平台不适用于9轴模式。	首次使用前需要校准地磁传感器

模块的自动地磁校准系统只能处理和模块安装在一起的，固定的磁场干扰。安装环境如果有磁场干扰，这种干扰必须是固定的，并且这个干扰磁场与模块安装之后不会再发生距离变化(例：模块安装在一个铁材料之上，因为铁会有磁场干扰，这时就需要把铁与模块一起旋转校准，并且这个铁在使用当中是不会和罗盘再分开的(发生相对位移)，一旦分开是需要再重新校准。如果这个铁大小是不固定的，或与罗盘的距离变化也不是固定的，这种干扰是无法校准，即使校准成功，也会精度非常差，只能避而远之安装。安全距离控制在40CM以上)。

校准方法

本模块集成了主动无干预地磁校准算法,模块会自动收集地磁场信息并进行硬磁软磁校准参数估计，校准成功后校准信息会保存在模块Flash上。用户无须任何操作/指令即可实现地磁校准。首次使用时，模块会自动采集周围地磁场，并尝试计算地磁传感器校准参数。当首次使用模块并且需要使用9轴模式时，应进行如下校准操作：

在尽量小范围内，缓慢的让模块运动和旋转，或者进行8字运动 或者分别绕每个轴360度，让模块经历尽量多的姿态。一般情况下，如果地磁干扰在可接受的范围内，即可完成校准。如果后面在同样地磁环境下(同地点)，则无需再次校准。如果始终没能成功校准模块，说明周围地磁场干扰比较大。

地磁校准状态可以使用AT指令来查看：

发送AT+INFO=HSI 指令，模块会打印当前地磁校准系统状态：

接收区

valid:13
fiterr:0.026630
bin_status:100%
cal_cnt:1
flux:61.265064
inclination:63.203552

OK

清除

发送AT指令

AT+INFO=HSI

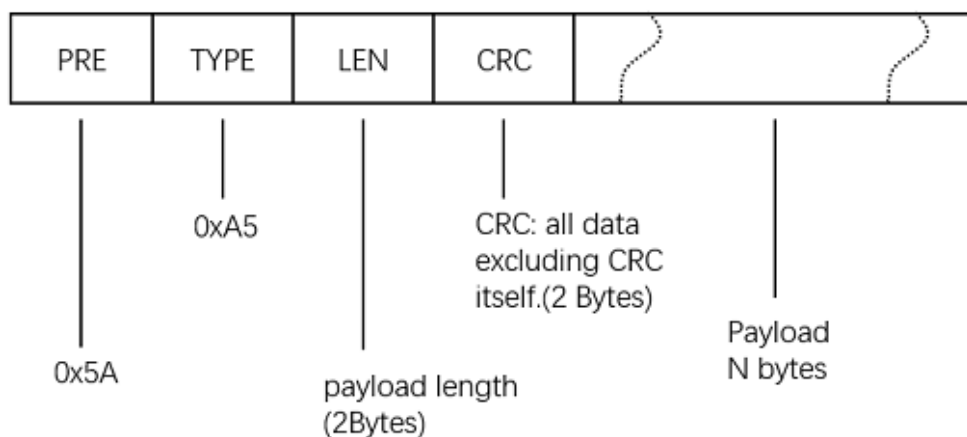
写入

参数显示	意义	说明
valid	有效标志	0: 不存在有效校准参数(没有校准或者从来没有校准成功过)。非0: 地磁校准完成
fiterr	拟合残差	残差越小，说明参数拟合效果越好，通常在0.03以下说明校准结果已经足够好。如果拟合结果始终>0.1，说明地磁干扰很大，最好再次校准以期得到更好的校准结果。拟合残差会随着时间缓慢增长。
flux	当地磁场	最近一次拟合器估计出的地磁场强，单位为uT
inclination	当地磁倾角	最近一次拟合器估计出的磁倾角，单位为°

- 虽然地磁参数估计可以在线自动采集数据，自动的动态拟合地磁校准参数。但是如果周围地磁环境改变(比如需要到另外房间或者室内室外切换)，最好还需重复手工校准操作。

串口通讯协议

模块上电后，模块默认按100Hz (出厂默认输出速率) 输出帧数据，帧格式如下：



其中:

域	值	长度 (字节)	说明
PRE	0x5A	1	固定为0x5A
TYPE	0xA5	1	固定为0xA5
LEN	1-512	2	帧中数据域的长度。LSB(低字节在前), 长度表示数据域的长度, 不包含PRE,TYPE,LEN,CRC 字段。
CRC	-	2	除CRC 本身外其余所有帧数据的16 位CRC 校验和。LSB(低字节在前)
PAYLOAD	-	1-512	一帧携带的数据。PAYLOAD 由若干个子数据包组成。每个数据包 包含: 数据包标签(DATA_ID)和数据(DATA) 两部分。DATA_ID决定了数据的类型及长度, DATA 为数据包内容。

CRC实现函数:

```

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,
7                          uint32_t lengthInBytes)
8  {
9      uint32_t crc = *currentCrc;
10     uint32_t j;
11     for (j=0; j < lengthInBytes; ++j)
12     {
13         uint32_t i;
14         uint32_t byte = src[j];
15         crc ^= byte << 8;
16         for (i = 0; i < 8; ++i)
17         {
18             uint32_t temp = crc << 1;
19             if (crc & 0x8000)
20             {
21                 temp ^= 0x1021;
22             }
23             crc = temp;
24         }
25     }
26 }

```

```
24     }
25     *currentCrc = crc;
26 }
```

数据包

数据包总览

数据包标签 (DATA_ID)	数据包长度(包含标签1字节)	名称	备注
0x90	2	用户ID	
0xA0	7	加速度	
0xB0	7	角速度	
0xC0	7	磁场强度	
0xD0	7	欧拉角	
0xD1	17	四元数	
0xF0	5	气压	输出0
0x91	76	IMUSOL(IMU数据集 合)	推荐使用

产品支持数据包列表

下表列出所有产品支持的数据包,*表示支持 -表示不支持

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

0x90(用户ID)

共2字节，用户设置的ID。

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0x90
1	uint8_t	1	-	用户ID

0xA0(加速度)

共7个字节，LSB。输出传感器的原始加速度

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0xA0
1	int16_t	2	0.001G(1G = 1重力加速度)	X轴加速度
3	int16_t	2	0.001G	Y轴加速度
5	int16_t	2	0.001G	Z轴加速度

0xB0(角速度)

共7字节，LSB。输出传感器的原始角速度

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签: 0xB0
1	int16_t	2	0.1°/s	X轴角速度
3	int16_t	2	0.1°/s	Y轴角速度
5	int16_t	2	0.1°/s	Z轴角速度

0xC0(磁场强度)

共7字节，LSB。输出传感器的原始磁场强度

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0xC0
1	int16_t	2	0.001Gauss	X轴磁场强度
3	int16_t	2	0.001Gauss	Y轴磁场强度
5	int16_t	2	0.001Gauss	Z轴磁场强度

0xD0(欧拉角)

共7字节，LSB。格式为int16，共三个轴，每个轴占2个字节，顺序为Pitch/Roll/Yaw。接收到Roll, Pitch 为物理值乘以100后得到的数值，Yaw 为乘以10得到的数值。

例：当接收到的Yaw = 100时，表示航向角为10°

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0xD0
1	int16_t	2	0.01°	Pitch(俯仰角)
3	int16_t	2	0.01°	Roll(横滚角)
5	int16_t	2	0.1°	Yaw(航向角)

0xD1(四元数)

共17字节，格式为float，共4个值，顺序为:W X Y Z。每个值占4字节(float)，整个四元数为4个float，LSB。

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0xD1
1	float	4	-	W
5	float	4	-	X
9	float	4	-	Y
13	float	4	-	Z

0xF0(气压)

共5字节，格式为float。(只针对有气压传感器的产品)

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0xF0
1	float	4	Pa	大气压

0X91(IMUSOL)

共76字节，新加入的数据包，用于替代A0,B0,C0,D0,D1等数据包。集成了IMU的传感器原始输出和姿态解算数据。

字节偏移	类型	大小	单位	说明
0	uint8_t	1	-	数据包标签:0x91
1	uint8_t	1	-	ID
2	-	6	-	保留
8	uint32_t	4	ms	时间戳信息，从系统开机开始累加，每毫秒增加1
12	float	12	1G(1G = 1重力加速度)	X,Y,Z轴的加速度，注意单位和0xA0不同
24	float	12	deg/s	X,Y,Z轴的角速度，注意单位和0xB0不同
36	float	12	uT	X,Y,Z轴的磁场强度(HI229支持,注意单位和0xC0不同)
48	float	12	deg	节点欧拉角集合,顺序为：横滚角(Roll)，俯仰角(Pitch)，航向角(Yaw)(注意顺序和单位与0xD0数据包不同)
60	float	16	-	节点四元数集合,顺序为WXYZ

出厂默认数据包

出厂默认一帧中携带数据包数据定义如下：

产品	默认输出数据包
HI226	90,A0,B0,C0,D0,F0
HI229	90,A0,B0,C0,D0,F0
CH110	90,A0,B0,C0,D0,F0

数据帧结构示例

数据帧配置为 **0x90, 0xA0, 0xB0, 0xC0, 0xD0, 0xF0** 数据包

使用串口助手采样一帧数据,共41字节,前6字节为帧头,长度和CRC校验值。剩余35字节为数据域。假设数据接收到C语言数组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：判断帧头，得到数据域长度和帧CRC：

帧头:5A A5

帧数据域长度:23 00: (0x00<<8) + 0x23 = 35

帧CRC校验值:FD 61:(0x61<<8) + 0xFD = 0x61FD

- 第二步：校验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值为0x61FD, 与帧携带的CRC值相同, 帧CRC校验通过。

- 第三步: 接收数据

90 00: ID 数据包, 0x90为数据包标签, ID = 0x00.

A0 55 02 3D 01 E2 02:加速度数据包, 0xA0为数据包标签, 三轴加速度为:

X轴加速度 = $(\text{int16_t})((0x02 \ll 8) + 0x55) = 597$ (单位为mG)

Y轴加速度 = $(\text{int16_t})((0x01 \ll 8) + 0x3D) = 317$

Z轴加速度 = $(\text{int16_t})((0x02 \ll 8) + 0xE2) = 738$

B0 FE FF 17 00 44 00:角速度数据包, 0xB0为数据包标签, 三轴角速度为:

X轴角速度 = $(\text{int16_t})((0xFF \ll 8) + 0xFE) = -2$ (单位为0.1°/s)

Y轴角速度 = $(\text{int16_t})((0x00 \ll 8) + 0x17) = 23$

Z轴角速度 = $(\text{int16_t})((0x00 \ll 8) + 0x44) = 68$

C0 80 FF 60 FF 32 FF:磁场数据包, 0xC0为数据包标签, 三轴磁场为:

X轴角速度 = $(\text{int16_t})((0xFF \ll 8) + 0x80) = -128$ (单位为0.001Gauss)

Y轴角速度 = $(\text{int16_t})((0xFF \ll 8) + 0x60) = -160$

Z轴角速度 = $(\text{int16_t})((0xFF \ll 8) + 0x32) = -206$

D0 64 F2 6C 0E BB 01 欧拉角数据包, 0xD0为数据包标签

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 气压数据包, 0xF0为数据包标签

```

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

```

最后得到结果:

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

数据帧配置为 0x91 数据包

使用串口助手采样一帧数据, 共82字节, 前6字节为帧头, 长度和CRC校验值。剩余76字节为数据域。假设数据接收到C语言数组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：判断帧头，得到数据域长度和帧CRC：

帧头:5A A5

帧数据域长度:4C 00: $(0x00 \ll 8) + 0x4C = 76$

帧CRC校验值:6C 51: $(0x51 \ll 8) + 0x6C = 0x516C$

- 第二步：校验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值为0x516C. 帧CRC校验通过。

- 第三步：接收数据

从0x91开始为数据包的数据域。在C语言中可以定义结构体来方便的读取数据：

定义0x91数据包结构体如下：

```

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];            /* eular angles: Roll,Pitch,Yaw */
11     float         quat[4];           /* quaternion */
12 }id0x91_t;

```

__packed为编译器关键字(Keil下)，表示结构体按字节紧对齐，结构体每一个元素一一对应0x91数据包的结构定义。接收数据时将接收到的数组直接memcpy到结构体即可：(注意定义结构体时必须4字节对齐),其中buf指向帧头,buf[6]指向帧中数据域。

```

1      /* 接收数据并使用0x91数据包结构定义来解释数据 */
2      __align(4) id0x91_t dat;        /* struct must be 4 byte aligned */
3      memcpy(&dat, &buf[6], sizeof(id0x91_t));

```

最后得到dat数据结果：

```

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通讯协议

本产品CAN接口遵循以下标准：

- CAN接口符合CANopen协议，所有通讯均使用标准数据帧
- 只使用PTO1-4 传输姿态数据，所有传输均采用标准数据帧，不接收远程帧和拓展数据帧
- PTO采用异步定时触发模式,默认输出速率为20Hz,最高可配置输出速率为100Hz
- 当模块上电时，按照CANopen协议，模块会主动发送一条(一次)节点上线报文。节点上电处于预操作状态(pre-operational) 需要主机发送NMT协议将节点设置为operation状态才会开始发送数据

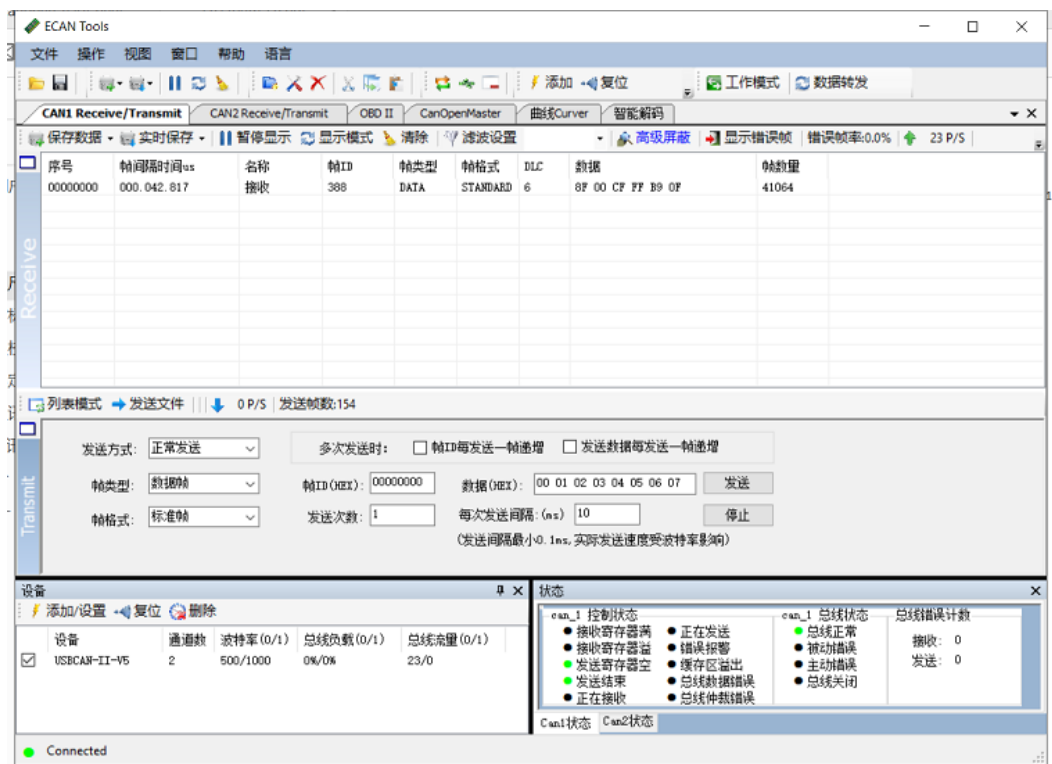
CANopen 默认设置

CANopen默认配置	值
CAN 波特率	500KHz
CANopen节点ID	8
初始化状态	预操作(Pre-operational),需要发送Start Remote Node命令模块才能开始输出数据
心跳包	无

CANopen PTO传输细节

PTO通道	PTO 帧ID	长度	PTO 传输方式	发送数据	说明
TPDO1	0x180+ID	6	循环同步 (0x01)	加速度	每轴数据类型为(INT16,低字节在前)，分别为X,Y,Z轴加速度，单位为mG(0.001重力加速度)
TPDO2	0x280+ID	6	循环同步 (0x01)	角速度	每轴数据类型为(INT16,低字节在前)，分别为X,Y,Z轴角速度，单位为0.1DPS(°/s)
TPDO3	0x380+ID	6	异步定时 (0xFE)	欧拉角	每轴数据类型为(INT16,低字节在前)，顺序分别为横滚角(Roll,绕X轴旋转),俯仰角(Pitch,绕Y轴旋转),航向角(Yaw绕Z轴旋转)。欧拉角单位为0.01°
TPDO4	0x480+ID	8	循环同步 (0x01))	四元数	每轴数据类型为(INT16,低字节在前)，分别为\$ q_{w} \backslash q_{x} \backslash q_{y} \backslash q_{z} \$。单位四元数扩大10000倍后结果。如四元数为1,0,0,0 时,输出10000,0,0,0.

使用USB-CAN工具抓取默认CAN输出包截图如下：



其中 欧拉角(PTO3) CAN帧ID =0x380 + 8(默认ID) = 0x388， 数据为:

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

修改CAN接口配置

数据字典以下位置存放厂商参数配置数据,可通过CANopen主站修改,掉电保存,重启生效

数据字典位置	名称	值类型	默认值	说明
0x2100	CAN_BAUD	INTEGER32	500000	CAN总线波特率
0x2101	NodeID	INTEGER32	8	节点ID

1. 使能数据输出(开启异步触发)

发送标准CANopen协议帧, 使用NMT: Start Remote Node命令:

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

其中 0x01为Start Remote Node指令, 0x08为节点ID

2. 修改CAN波特率

发送标准CANopen协议帧, 使用标准快速SDO指令

如将CAN波特率修改为125K, 则发送:

ID=0x608

, DLC=8, DATA=0x23, 0x00, 0x21, 0x00, 0x48, 0xE8, 0x01, 0x00(ID=0x608, 长度为8的标准数据帧)

其中 $0x01, 0xE8, 0x48 = (0x01 \ll 16) + (0xE8 \ll 8) + 0x48 = 125000$, 注意发送ID为(0x600+ID, 写SDO命令帧)

如将CAN波特率修改为250K, 则发送:

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

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

3. 修改输出速率

发送标准CANopen协议帧，使用标准快速SDO指令：

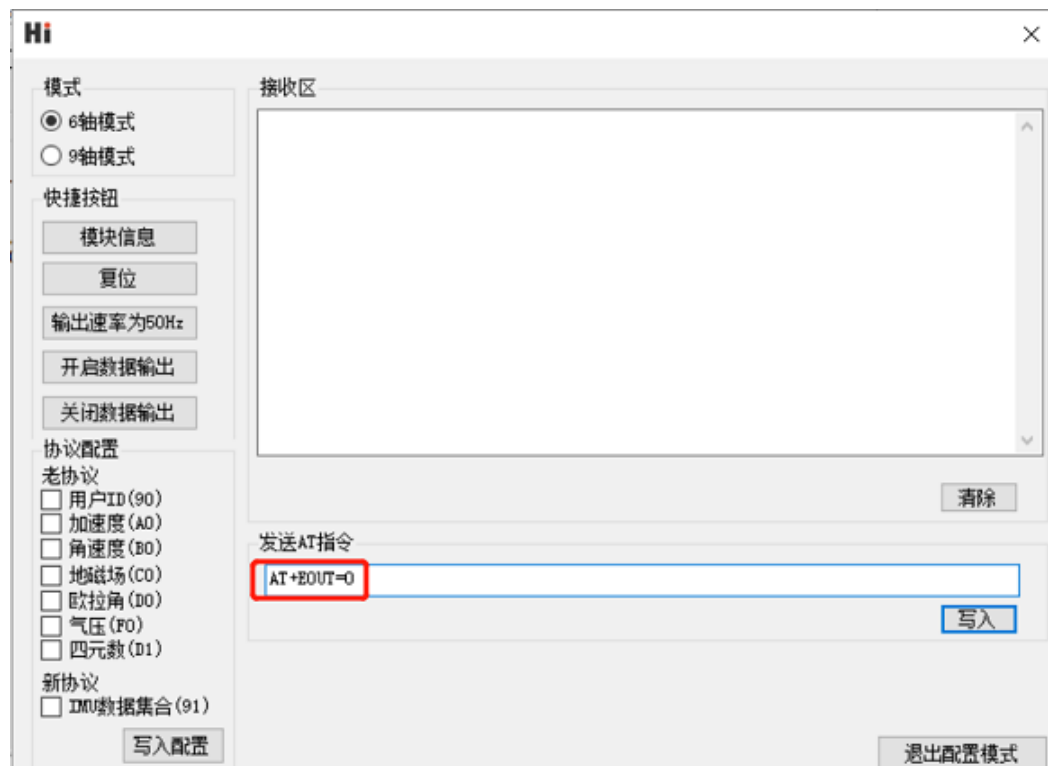
如修改TPDO3(欧拉角)输出速率为100Hz:(10ms event触发)

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

其中 0x00, 0x0A = (0x00<<8) + 0x0A = 10(10ms触发=100Hz)

AT指令

当使用串口与模块通讯时, 模块支持AT指令集配置/查看模块参数。AT指令总以ASCII码AT开头，后面跟控制字符，最后以回车换行\r\n结束。可使用串口调试助手进行测试：



通用模块 AT指令如下

指令	功能	掉电保存(Y)	立即生效(Y),复位生效(R)	备注
AT+ID	设置模块用户ID	Y	R	
AT+INFO	打印模块信息	N	Y	
AT+ODR	设置模块串口输出帧频率	Y	R	
AT+BAUD	设置串口波特率	Y	R	
AT+EOUT	数据输出开关	N	Y	
AT+RST	复位模块	N	Y	
AT+TRG	单次输出触发	N	Y	部分型号支持
AT+SETPTL	设置输出数据包	Y	Y	部分型号支持
AT+MODE	设置模块工作模式	Y	R	部分型号支持
AT+GWID	设置无线网关ID	Y	R	部分型号支持

AT+ID

设置模块用户ID

例 AT+ID=1

AT+INFO

打印模块信息，包括产品型号，版本，固件发布日期等。

AT+ODR

设置模块串口输出速率。掉电保存，复位模块生效

例 设置串口输出速率为100Hz: AT+ODR=100

AT+BAUD

设置串口波特率，可选值：9600/115200/460800/921600`

例 AT+BAUD=115200

注意

- 使用此指令需要特别注意，输入错误波特率后会导致无法和模块通讯
- 波特率参数设置好后掉电保存，复位模块生效。上位机的波特率也要做相应修改。
- 升级固件时，需要切换回115200波特率。

AT+EOUT

串口输出开关

例 打开串口输出 AT+EOUT=1 关闭串口输出 AT+EOUT=0

AT+RST

复位模块

例 AT+RST

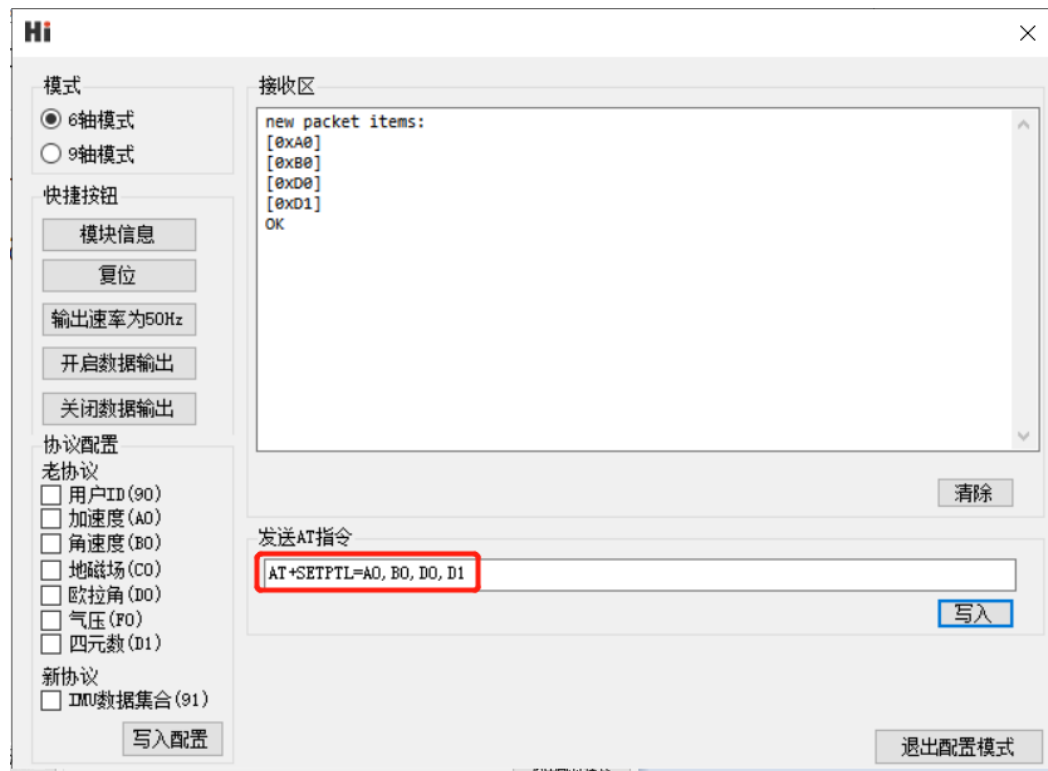
AT+SETPTL

设置输出协议:

模块数据帧中的数据包组成可使用AT指令配置, 格式为AT+SETPTL=<ITEM_ID>, <ITEM_ID>...

例 配置模块输出加速度, 角速度, 整形格式欧拉角和四元数的指令为:

AT+SETPTL=A0,B0,D0,D1



AT+MODE

设置模块工作模式

例

- 设置模块工作在6轴模式(无磁校准) AT+MODE=0
- 设置模块工作在9轴模式(地磁场传感器参与航向角校正) AT+MODE=1

附录C - 固件升级与恢复出厂设置

本产品支持升级固件。固件升级步骤:

- 连接模块, 打开上位机, 将模块和上位机波特率都设置为115200. 打开固件升级窗口
- 点击连接按钮, 如出现模块连接信息。则说明升级系统准备就绪, 点击文件选择器(...)选择拓展名为.hex的固件, 然后点击开始编程。下载完成后会提示编程完成, 此时关闭串口, 重新给模块上电, 模块升级完成。

