

适用于HI01/HI02/HI04/HI13/HI14/HI50系列



文件: imu_cum_cn

技术支持: support@hipnuc.com

属性: 公开

网站: www.hipnuc.com

HiPNUC

指令与编程手册

1 模块配置概述

1.1 地磁辅助场景(AHRS/9轴模式)

1.2 同步输入与同步输出

1.2.1 同步输入(SIN)

1.2.2 同步输出(SOUT)

2 模块配置命令

2.1 配置命令总览

2.2 指令详解

2.2.1 REBOOT

2.2.2 SAVECONFIG

2.2.3 SERIALCONFIG

2.2.4 CONFIG

2.2.4.1 模式配置(6轴, 9轴)

2.2.4.2 水平校准

2.2.4.3 坐标系旋转(改变安装方式)

2.2.4.4 设置报警触发参数(仅倾角仪产品)

2.2.5 LOG

2.2.5.1 ENABLE/DISABLE: 全局打开/关闭数据输出

2.2.5.2 VERSION: 显示模块版本信息

2.2.5.3 USRCONFIG: 显示用户配置信息

2.2.5.4 COMCONFIG: 显示串口配置信息

2.2.5.5 MSG: 配置数据帧输出及设置输出频率

2.2.6 UNLOGALL

2.2.7 FRESET

2.3 更多高级配置

2.3.1 配置加速度和陀螺仪带宽

3 数据输出协议(自定义二进制协议)

3.1 数据帧格式

3.2 出厂默认输出

3.3 数据域内容

3.3.1 浮点型IMU数据帧(0x91)

3.3.2 整型IMU数据帧(0x92)

3.4 CRC

3.5 数据帧结构示例

4 RS-485数据协议与指令(Modbus协议)

4.1 数据帧格式

4.1.1 读寄存器(0x03)

4.1.2 写寄存器(0x06)

4.1.3 CRC校验

4.2 寄存器列表

- 4.3 常用配置
 - 4.3.1 控制寄存器说明(地址:0x0000)
 - 4.3.2 配置波特率
- 4.4 读取模块版本信息
- 4.5 读取传感器数据
- 4.6 Modbus ID自动分配
- 5 CAN数据协议与指令(CANopen协议)
 - 5.1 CANopen 默认设置
 - 5.2 CANopen TPDO
 - 5.3 使用上位机连接CAN设备
 - 5.4 配置指令
 - 5.4.1 LLS(Layer Setting Services)协议
 - 5.4.1.1 修改波特率(重新上电生效)
 - 5.4.1.2 修改节点ID
 - 5.4.1.3 保存配置
 - 5.4.2 NMT(Network Management)协议
 - 5.4.2.1 全局使能/关闭数据输出(开启异步触发)
 - 5.4.2.2 复位
 - 5.4.3 SDO(Service Data Object)协议
 - 5.4.3.1 修改/关闭/开启数据输出速率
 - 5.4.3.2 设置倾角仪输出正负号
 - 5.4.3.3 设置倾角仪零点(相对位置输出)
 - 5.4.3.4 设置倾角输出分辨率(定制功能)
 - 5.4.4 同步协议
 - 5.4.4.1 配置TPDO为同步模式
- 6 CAN数据协议与指令(SAE-J1939协议)
 - 6.1 PGN消息列表
 - 6.1.1 PGN65332(FF34) 加速度
 - 6.1.2 PGN65335(FF37) 角速度
 - 6.1.3 PGN65345(FF41) 航向角
 - 6.1.4 PGN65354(FF4A) 倾角仪输出
 - 6.2 配置指令
 - 6.2.1 配置格式
 - 6.2.2 配置模块
- 7 CAN数据协议与指令(NMEA2000协议)
- 8 地磁校准
 - 8.1 地磁校准步骤
 - 8.2 磁干扰的知识
 - 8.2.1 空间磁场干扰
 - 8.2.2 传感器坐标系下的干扰
 - 8.3 地磁使用注意事项

1. 模块配置概述

产品的默认配置已经可以满足大多数用户需求，因此在使用产品前需要仔细阅读此章节，结合自身的使用需求，判断是否需要进行用户配置。

1.1 地磁辅助场景(AHRS/9轴模式)

在绝大多数情况下，机器人及室内环境，AHRS(9轴)模式很容易受到干扰导致航向角产生误差。在少数空旷且无磁场干扰的环境下，可以尝试使用地磁辅助模式，比如无人机，在使用前需要先将模块配置为地磁辅助模式并且进行地磁校准。详见地磁校准章节。

串行接口配置请参考 CONFIG-模式配置

1.2 同步输入与同步输出

1.2.1 同步输入(SIN)

同步脉冲输入(PPS_SYNC_PIN/SIN): 输入引脚, 空闲状态为高电平。当模块检测到下降沿时, 会输出一帧数据(如果为ONMARK触发), 同时, 一些协议可以输出PPS同步时间戳。PPS同步时间戳指的是: 当模块检测到最近一次下降沿信号时开始, 到当前一帧数据采样时所经过的时间。该引脚不使用可悬空。

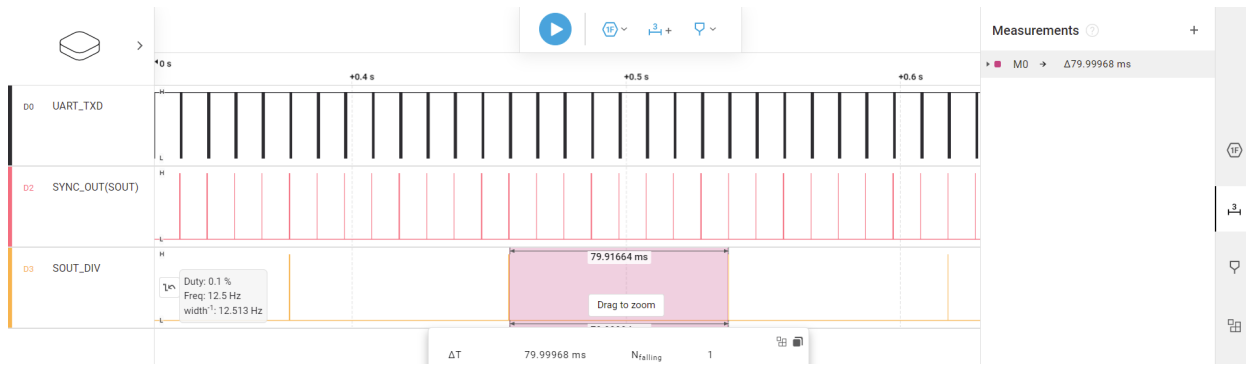
1.2.2 同步输出(SOUT)

- 数据同步输出(SOUT): 输出引脚, 无数据输出时为低电平 (空闲), 一帧数据开始发送时会发送一个高脉冲, 脉冲宽度位80us. 例: 默认状态下, 模块输出100Hz数据, 该引脚输出100Hz 脉冲信号。不使用时需悬空。
- 同步输出波形示例(100Hz输出速率):



在使用同步输入功能时，先要通过串口输入指令UNLOGALL来取消所有定时输出

- 部分定制型号还拥有SOUT_DIV引脚，用于同步输出引脚分频输出，例，数据输出速率为100Hz，则SOUT输出频率为100Hz，如SOUT_DIV分频系数为8(默认)，则SOUT_DIV输出频率为12.5Hz(周期80ms)，如下图:



2. 模块配置命令

模块配置采用ASCII字符串命令，每条命令必须以回车换行 `\r\n` 结束(类似AT指令)，才能被系统识别。

2.1 配置命令总览

| 指令 | 功能 | 备注 |
|--------------|---------------|---------|
| REBOOT | 复位模块 | 和重新上电等效 |
| SAVECONFIG | 保存所有配置参数 | 立即生效 |
| SERIALCONFIG | 波特率设置 | 立即生效 |
| CONFIG | 设置用户参数及模式 | 立即生效 |
| LOG | 打印模块信息或配置输出数据 | 立即生效 |
| UNLOGALL | 关闭所有定时输出消息 | 立即生效 |
| FRESET | 恢复出厂设置 | 立即生效 |

所有配置指令需要 复位或重新上电后才能生效

2.2 指令详解

2.2.1 REBOOT

复位模块，立即生效，和重新上电效果相同。

2.2.2 SAVECONFIG

保存所有用户配置到Flash。

2.2.3 SERIALCONFIG

设置串口波特率，可选值：9600/115200/256000/460800/921600

配置串口波特率为115200

SERIALCONFIG 115200

SAVECONFIG

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。

2.2.4 CONFIG

用于配置模块工作参数，所有配置后都需要 SAVECONFIG 后且复位生效

2.2.4.1 模式配置(6轴, 9轴)

CONFIG ATT MODE 0 配置模块为6DOF模式

CONFIG ATT MODE 1 配置模块为AHRS(9轴)模式

2.2.4.2 水平校准

CONFIG ATT RST 3 自动校平: 如果当前俯仰角/横滚角接近0°,0°(水平正面放置), 则自动校准到0,0。如果当前俯仰角/横滚角接近0° 或 180°(水平倒放), 则自动校准到 0°,180°。适用于机器人安装环境。其中, “接近” 定义为 Pitch Roll均小于15°

CONFIG ATT RST 5 取消水平校平: 清除当前俯仰横滚角校平设置(恢复默认)

执行 CONFIG ATT RST命令时, 模块需要保持静止, 如果模块在运动中执行该命令, 则有可能造成较大校平误差

2.2.4.3 坐标系旋转(改变安装方式)

CONFIG IMU URFR C00,C01,C02,C10,C11,C12,C20,C21,C22

其中 C_{nn} 支持浮点数

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

其中 $\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_U$ 为旋转后的 传感器坐标系下 传感器数据, $\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B$ 为旋转前 传感器坐标系下 传感器数据

下面是几种常用旋转举例:

- 新传感器坐标系为 绕原坐标系X轴 旋转 -90°(Y轴正方向朝下的垂直安装), 配置命令:

CONFIG IMU URFR 1,0,0,0,0,1,0,-1,0

- 新传感器坐标系为 绕原坐标系X轴 旋转 90°(Y轴正方向朝上的垂直安装), 配置命令:

CONFIG IMU URFR 1,0,0,0,0,-1,0,1,0

- 新传感器坐标系为 绕原坐标系X轴 旋转180°, 配置命令:

CONFIG IMU URFR 1,0,0,0,-1,0,0,0,-1

- 新传感器坐标系为 绕原坐标系Y轴 旋转 90°(X轴正方向朝上的垂直安装), 配置命令

CONFIG IMU URFR 0,0,-1,0,1,0,1,0,0

- 新传感器坐标系为 绕原坐标系Y轴 旋转-90°(X轴正方向朝下垂直安装), 配置命令:

CONFIG IMU URFR 0,0,1,0,1,0,-1,0,0

- 新传感器坐标系为 绕原坐标系Y轴 旋转180°， 配置命令：

CONFIG IMU URFR -1,0,0,0,1,0,0,0,-1

- 新传感器坐标系为 绕原坐标系Z轴 旋转90°， 配置命令：

CONFIG IMU URFR 0,-1,0,1,0,0,0,0,1

- 新传感器坐标系为 绕原坐标系Z轴 旋转-90°， 配置命令：

CONFIG IMU URFR 0,1,0,-1,0,0,0,0,1

- 水平，Z轴朝上安装 (默认值)：

CONFIG IMU URFR 1,0,0,0,1,0,0,0,1

- 设置URFR后需要软件复位或重新上电才能生效，不需要每次上电都发送该指令。
- 如何确定URFR参数:(以绕原坐标系X轴 旋转 -90°(Y轴正方向朝下的垂直安装) 为例)。可写出转换后坐标与转换前坐标的关系：

- $X_U = X_B$

- $Y_U = -Z_B$

- $Z_U = Y_B$

从而可以写出转换矩阵 =

$$\begin{matrix} 1 & 0 & 0 \text{ (转换后的X = 转换前的X)} \\ 0 & 0 & -1 \text{ (转换后的Y = 转换前的-Z)} \\ 0 & 1 & 0 \text{ (转换后的Z = 转换前的Y)} \end{matrix}$$

按照上述URFR定义公式，URFR参数需要的实际上是上述矩阵的转置，即：

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{matrix}$$

2.2.4.4 设置报警触发参数(仅倾角仪产品)

此功能用于输出报警信号，可以设置当X或Y角度超过一定范围后触发报警。SOUT(同步输出引脚)用于报警触发功能。当报警时，SOUT引脚输出高电平，否则输出低电平。

- CONFIG ALARM AIXS,MIN,MAX,THR,ENABLE 设置角度报警参数
 - AIXS: 轴选择: 'X' : X轴, 'Y' : Y轴
 - MIN, MAX: 安全区域最大最小值, 如 MAX= -5, MAX=5 则 -5° ~ 5°内不报警，超出此范围则报警。
 - THR: 报警软件滤波时间: 当角度发生报警时，报警信号不会立马输出，需要角度报警持续一定时间，才能输出报警信号。该值用于滤除误动作导致的报警。如 THR设置为500， 则滤波时间为500ms
 - ENABLE: 0:关闭,1:使能

例: CONFIG ALARM Y,-5,5,50,1: 使能Y轴报警功能, 当角度低于-5°或者高于5°且超出持续大于50ms时, 触发报警。

2.2.5 LOG

2.2.5.1 ENABLE/DISABLE: 全局打开/关闭数据输出

LOG ENABLE 全局使能数据帧输出(默认)

LOG DISABLE 全局禁止数据帧输出

2.2.5.2 VERSION: 显示模块版本信息

LOG VERSION 打印固件版本信息

2.2.5.3 USRCONFIG: 显示用户配置信息

LOG USRCONFIG 打印用户配置信息, 用于查看配置是否被写入成功。

```
1 ATT_MODE: 0 /* 工作模式: 0:6轴, 1:9轴 */
2 ...
```

2.2.5.4 COMCONFIG: 显示串口配置信息

LOG COMCONFIG 打印串口及输出协议配置信息

2.2.5.5 MSG: 配置数据帧输出及设置输出频率

LOG <MSG> <TYPE> <PERIOD>

- MSG: IMU91, HI91(同IMU91), HI92
- TYPE: ONTIME: 定时输出, ONMARK: 外部触发同步输出
- PERIOD: 输出帧周期, 单位为s, 取值范围: 1(1Hz), 0.5(2Hz), 0.1(10Hz), 0.02(50Hz), 0.01(100Hz), 0.005(200Hz), 0.002(500Hz),以此类推

例(定时100Hz输出):

- **LOG IMU91 ONTIME 0.01** 将当前串口的 91数据包输出周期设置为0.01s(100Hz)
- **LOG HI92 ONTIME 0.05** 将当前串口的 92数据包输出周期设置为0.05s(20Hz)

例(关闭输出):

- **LOG IMU91 ONTIME 0** 关闭91数据包输出

例(同步触发输出):

- **UNLOGALL** 取消所有数据输出。

- LOG IMU91 ONMARK 1 将当前COM口的 91数据包设置为同步触发模式。

当输出帧率设置为比较高时(比如500Hz),默认的115200波特率不满足输出带宽要求,此时需要将模块波特率设高(比如921600)后,模块才能正确的输出数据。

波特率参数设置好后掉电保存,复位模块生效。上位机或者其他主机的波特率也要做相应修改。

2.2.6 UNLOGALL

将所有定时输出消息输出频率设为0(不输出)

2.2.7 FRESET

恢复出厂设置

2.3 更多高级配置

除了以上常用配置外，模块还可以配置一些高级性能参数，这些参数一般情况不需要修改，出厂前已经调节到最优。

2.3.1 配置加速度和陀螺仪带宽

CONFIG IMU ABW <VAL> 调整加速度计带宽 VAL = 2(20Hz), 3(40Hz), 4(80Hz), 5(125Hz), 6(230Hz,默认)

CONFIG IMU GBW <VAL> 调整陀螺仪带宽 VAL = 0 (12Hz), 3(47Hz), 4(80Hz), 5(116Hz,默认), 6(230Hz)

例: 将陀螺仪滤波器3db截至带宽调整为47Hz: CONFIG IMU GBW 3

- 不要配置为手册上未标注值, 否则可能造成输出数据异常, 改变加速度计和陀螺的带宽并不能带来精度的提升, 如为提高精度, 不建议修改。
- 为了减小静态噪声或者振动带来的影响可以去适当降低加速度计带宽。
- 调低带宽的代价是输出数据相位延迟更大, 在一些要求实时高动态的应用中不应将带宽调的过低。

3. 数据输出协议(自定义二进制协议)

该协议是超核定义的二进制协议，可以输出全部传感器信息，支持该协议的接口:RS-232/TTL/USB(虚拟串口)。默认串口格式为N-8-N-1(8位数据位，1位停止位，0位校验位)

3.1 数据帧格式

模块上电后，按照默认帧率(100Hz)输出帧数据，帧格式如下：

| 域名称 | 值 | 长度(字节) | 说明 |
|-------|-------|--------|---|
| 帧头 | 0x5A | 1 | 0x5A |
| 帧类型 | 0xA5 | 1 | 0xA5 |
| 数据域长度 | 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(不包含帧头，帧类型，长度，CRC) |
| CRC校验 | - | 2 | 除CRC 字节外其余所有字段(帧头,帧类型,长度, 数据域)的16 位CRC 校验和。 LSB(低字节在前) |
| 数据域 | - | 1-512 | 一帧携带的数据，由若干个子数据包组成，数据包包含数据包标签和数据两部分。 标签决定了数据的类型及长度。 |

3.2 出厂默认输出

出厂默认输出: 浮点型IMU数据帧(0x91)

3.3 数据域内容

3.3.1 浮点型IMU数据帧(0x91)

数据域共76字节。包含模块ID、温度、IMU的原始数据、地磁、气压、融合后的姿态数据等。

开启数据帧示例: LOG IMU91 ONTIME 1, 或 LOG HI91 ONTIME 1 (需固件版本>=1.5.0支持), 具体详见配置指令章节

| 字节偏移 | 名称 | 数据类型 | 大小 (Byte) | 单位 | 比例因子 | 说明 |
|------|----------------|----------|-----------|------------|------|---|
| 0 | tag | uint8_t | 1 | - | - | 数据包标签:0x91 |
| 1 | pps_sync_stamp | uint16_t | 2 | ms | 1 | PPS脉冲同步时间戳，用于精确时间同步，该值定义为：最近一次检测到PPS同步引脚脉冲沿到本帧数据采样时刻所经过的时间，范围 0-8192,当超过最大值时会自动回滚到0，例：如果用户在PPS引脚上输入的是标准秒脉冲，则该值会在0-1000之间。 |
| 3 | temperature | int8_t | 1 | °C | 1 | 模块平均温度 |
| 4 | air_pressure | float | 4 | Pa | 1 | 气压 |
| 8 | system_time | uint32_t | 4 | ms | 1 | 节点本地时间戳信息，从系统开机开始累加，每毫秒增加1 |
| 12 | acc_b | float | 4*3 | G | 1 | 经过出厂校准后的加速度,顺序为：XYZ轴. 1G=1x当地重力加速度, 可近似为9.8 m/s^2 |
| 24 | gyr_b | float | 4*3 | deg/s(dps) | 1 | 经过出厂校准后的角速度,顺序为：XYZ轴 |
| 36 | mag_b | float | 4*3 | uT | 1 | 磁强度,顺序为：XYZ轴 |
| 48 | roll | float | 4 | deg | 1 | 横滚角 |
| 52 | pitch | float | 4 | deg | 1 | 俯仰角 |
| 56 | yaw | float | 4 | deg | 1 | 航向角 |
| 60 | quat | float | 4*4 | - | - | 节点四元数集合,顺序为WXYZ |

3.3.2 整型IMU数据帧(0x92)

共48字节，比浮点型数据帧数据量更小。开启数据帧示例: LOG HI92 ONTIME 1, 具体详见配置指令章节

| 字节偏移 | 名称 | 数据类型 | 大小 (Byte) | 单位 | 比例因子 | 说明 |
|------|----------------|-----------|--------------|---------|-----------|--|
| 0 | tag | uint8_t | 1 | - | - | 数据包标签:0x92 |
| 1 | status | uint16_t | 2 | - | - | 状态字, 保留 |
| 3 | temperature | int8_t | 1 | °C | 1 | 系统平均温度 |
| 4 | pps_sync_stamp | uint16_t | 2 | ms | 1 | PPS脉冲同步时间戳, 用于精确时间同步, 该值定义为: 最近一次检测到PPS同步引脚脉冲沿到本帧数据采样时刻所经过的时间, 范围 0-8192,当超过最大值时会自动回滚到0, 例: 如果用户在PPS引脚上输入的是标准秒脉冲, 则该值会在0-1000之间。 |
| 6 | air_pressure | int16_t | 2 | Pa | 1 | 大气压+100000Pa : 如2000表示 102000Pa |
| 8 | reserved | - | 2 | - | - | 保留 |
| 10 | acc_b | int16_t*3 | 6 | m/s^(2) | 0.0048828 | IMU加速度: X,Y,Z轴(出厂校准后) |
| 16 | gyr_b | int16_t*3 | 6 | rad/s | 0.001 | IMU角速度: X,Y,Z轴(出厂校准后) |
| 22 | mag_b | int16_t*3 | 6 | uT | 0.030517 | IMU磁强度: X,Y,Z轴(出厂校准后) |
| 28 | roll | int32_t | 4 | deg | 0.001 | 横滚角 |
| 32 | pitch | int32_t | 4 | deg | 0.001 | 俯仰角 |
| 36 | yaw | int32_t | 4 | deg | 0.001 | 航向角 |
| 40 | quat | int16_t*4 | 8 | - | 0.0001 | 节点四元数集合,顺序为WXYZ |

3.4 CRC

16-bit CRC实现例程:

```
1  /*
2   correctCrc: 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 *correctCrc, const uint8_t *src, uint32_t lengthInBytes)
7  {
8      uint32_t crc = *correctCrc;
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 }
```

3.5 数据帧结构示例

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

- 1.判断帧头, 得到数据域长度和CRC校验值:

帧头:5A A5

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

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

- 2.校验CRC

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

得到CRC值为0x516C, 与帧中携带CRC值相同, CRC校验通过。

- 3.接收数据

从 0x91 开始为数据包的数据域，定义数据结构和常用转换宏：

```
1  #include "stdio.h"
2  #include "string.h"
3  /* common type conversion */
4  #define U1(p) (*((uint8_t *) (p)))
5  #define I1(p) (*((int8_t *) (p)))
6  #define I2(p) (*((int16_t *) (p)))
7  static uint16_t U2(uint8_t *p) {uint16_t u; memcpy(&u,p,2); return u;}
8  static uint32_t U4(uint8_t *p) {uint32_t u; memcpy(&u,p,4); return u;}
9  static int32_t I4(uint8_t *p) {int32_t u; memcpy(&u,p,4); return u;}
10 static float R4(uint8_t *p) {float r; memcpy(&r,p,4); return r;}
11 typedef struct
12 {
13     uint8_t tag; /* item tag: 0x91 */
14     float acc[3]; /* acceleration */
15     float gyr[3]; /* angular velocity */
16     float mag[3]; /* magnetic field */
17     float eul[3]; /* attitude: euler angle */
18     float quat[4]; /* attitude: quaternion */
19     float pressure; /* air pressure */
20     uint32_t timestamp;
21 }imu_data_t;
```

接收数据，从buf[6]=0x91开始为payload部分：

```
1  imu_data_t i0x91 = {0};
2  int offset = 6; /* payload start at buf[6] */
3  i0x91.tag = U1(buf+offset+0);
4  i0x91.pressure = R4(buf+offset+4);
5  i0x91.timestamp = U4(buf+offset+8);
6  i0x91.acc[0] = R4(buf+offset+12);
7  i0x91.acc[1] = R4(buf+offset+16);
8  i0x91.acc[2] = R4(buf+offset+20);
9  i0x91.gyr[0] = R4(buf+offset+24);
10 i0x91.gyr[1] = R4(buf+offset+28);
11 i0x91.gyr[2] = R4(buf+offset+32);
```

```

12  i0x91.mag[0] =    R4(buf+offset+36);
13  i0x91.mag[1] =    R4(buf+offset+40);
14  i0x91.mag[2] =    R4(buf+offset+44);
15  i0x91.eul[0] =    R4(buf+offset+48);
16  i0x91.eul[1] =    R4(buf+offset+52);
17  i0x91.eul[2] =    R4(buf+offset+56);
18  i0x91.quat[0] =    R4(buf+offset+60);
19  i0x91.quat[1] =    R4(buf+offset+64);
20  i0x91.quat[2] =    R4(buf+offset+68);
21  i0x91.quat[3] =    R4(buf+offset+72);

```

打印接收到的数据:

```

1  printf("%-16s0x%X\r\n",      "tag:",      i0x91.tag);
2  printf("%-16s%8.4f %8.4f %8.4f\r\n",  "acc(G):",    i0x91.acc[0], i0x91.acc[1], i0x91.acc[2]);
3  printf("%-16s%8.3f %8.3f %8.3f\r\n",  "gyr(deg/s):", i0x91.gyr[0], i0x91.gyr[1], i0x91.gyr[2]);
4  printf("%-16s%8.3f %8.3f %8.3f\r\n",  "mag(uT):",    i0x91.mag[0], i0x91.mag[1], i0x91.mag[2]);
5  printf("%-16s%8.3f %8.3f %8.3f\r\n",  "eul(deg):",   i0x91.eul[0], i0x91.eul[1], i0x91.eul[2]);
6  printf("%-16s%8.3f %8.3f %8.3f %8.3f\r\n", "quat:",      i0x91.quat[0], i0x91.quat[1], i0x91.quat[2],
    i0x91.quat[3]);
7  printf("%-16s%8.3f\r\n",            "pressure(pa):", i0x91.pressure);
8  printf("%-16s%d\r\n",                "timestamp(ms):", i0x91.timestamp);

```

打印出的解析结果:

```

1  tag:      0x91
2  acc(G):    0.2242  0.7701  0.6910
3  gyr(deg/s): -54.708 -20.077 -119.070
4  mag(uT):    19.183 -26.208 -34.542
5  eul(deg):   48.720 -21.014 -45.512
6  quat:      0.855  0.310 -0.310 -0.277
7  pressure(pa): -0.000
8  timestamp(ms): 310205

```

4. RS-485数据协议与指令(Modbus协议)

- 支持该协议的接口: RS485
- 默认串口配置: 115200 - N8N1
- Modbus 指令：
 - RS485通讯协议遵循Modbus RTU协议规范，数据以寄存器为单位进行发送和接收，每个寄存器占用2个字节，采用大端模式(高字节在前)
 - 写入: 0x06 (Write Single Register) :写单个寄存器(每个Modbus寄存器为2个字节)
 - 读取: 0x03 (Read Holding Registers): 读取单个或多个寄存器数据
 - 自定义功能码: 0x50, 用于 Modbus ID自动分配，方便量产部署，固件升级等
- Modbus设备地址可修改, 出厂默认: 80 (0x50)

4.1 数据帧格式

4.1.1 读寄存器(0x03)

主机发送:

| 域名称 | 值 | 说明 |
|----------|--------|------------------------|
| ID | 1-0xFF | Modbus设备地址 |
| FUN_CODE | 0x03 | 命令码 |
| ADDR_H | - | 要读取的寄存器地址高8位 |
| ADDR_L | - | 要读取的寄存器地址低8位 |
| LEN_H | - | 要读取寄存器长度高8位(以寄存器个数为单位) |
| LEN_L | - | 要读取寄存器长度低8位(以寄存器个数为单位) |
| CRC_L | - | CRC低8位 |
| CRC_H | - | CRC高8位 |

从机(模块)返回:

| 域名称 | 值 | 说明 |
|----------|--------|--|
| ID | 1-0xFF | Modbus设备地址 |
| FUN_CODE | 0x03 | 命令码 |
| LEN | - | 返回寄存器数据的长度(不算ID, FUN_CODE,LEN,CRC字段)以字节为单位 |
| DATAH | - | 返回数据高8位 |
| DATAL | - | 返回数据低8位 |
| ---- | - | 返回数据高8位 |
| ---- | - | 返回数据低8位 |
| CRC_L | - | CRC低8位 |
| CRC_H | - | CRC高8位 |

4.1.2 写寄存器(0x06)

| 域名称 | 值 | 说明 |
|----------|--------|------------|
| ID | 1-0xFF | Modbus设备地址 |
| FUN_CODE | 0x06 | 命令码 |
| ADDR_H | - | 寄存器地址高8位 |
| ADDR_L | - | 寄存器地址低8位 |
| DATA_H | - | 写入数据高8位 |
| DATA_L | - | 写入数据低8位 |
| CRC_L | - | CRC低8位 |
| CRC_H | - | CRC高8位 |

从机返回:

| 域名称 | 值 | 说明 |
|----------|--------|------------|
| ID | 1-0xFF | Modbus设备地址 |
| FUN_CODE | 0x06 | 命令码 |
| ADDR_H | - | 寄存器地址高8位 |
| ADDR_L | - | 寄存器地址低8位 |
| DATA_H | - | 写入数据高8位 |
| DATA_L | - | 写入数据低8位 |
| CRC_L | - | CRC低8位 |
| CRC_H | - | CRC高8位 |

4.1.3 CRC校验

- 在线计算CRC: <https://www.23bei.com/tool/59.html>
- C代码:

```
1 static const uint16_t modbus_crc_table[256] = {
2     0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,
3     0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,
4     0xcc01, 0x0cc0, 0x0d80, 0xcd41, 0x0f00, 0xcfc1, 0xce81, 0x0e40,
5     0x0a00, 0xcac1, 0xcb81, 0x0b40, 0xc901, 0x09c0, 0x0880, 0xc841,
6     0xd801, 0x18c0, 0x1980, 0xd941, 0x1b00, 0xdbc1, 0xda81, 0x1a40,
7     0x1e00, 0xdec1, 0xdf81, 0x1f40, 0xdd01, 0x1dc0, 0x1c80, 0xdc41,
8     0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,
9     0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,
10    0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,
11    0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,
12    0x3c00, 0xfcc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,
13    0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,
```

```

14 0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,
15 0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,
16 0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,
17 0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
18 0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
19 0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
20 0x6c00, 0xacc1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
21 0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
22 0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
23 0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbdc1, 0xbc81, 0x7c40,
24 0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
25 0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,
26 0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
27 0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
28 0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
29 0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
30 0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
31 0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
32 0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
33 0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040
34 };
35
36 uint16_t modbus_crc_calc(uint8_t *buf, uint16_t len)
37 {
38     uint16_t crc = 0xFFFFU;
39     uint8_t nTemp;
40
41     while (len--)
42     {
43         nTemp = *buf++ ^ crc;
44         crc >>= 8;
45         crc ^= modbus_crc_table[(nTemp & 0xFFU)];
46     }
47
48     return(crc);
49 }

```

4.2 寄存器列表

| 地址 (Hex) | 地址 (Dec) | 名称 | 功能 | R/W | 说明 |
|-------------|-------------|--------------|--------------|-----|---|
| 0x00 | 0 | CTRL | 控制 | W | 参见Modbus 设置模块章节 |
| 0x04 | 4 | UART1_BAUD | 波特率 | R/W | 串口波特率 |
| 0x05 | 5 | MD_ID | Modbus ID | R/W | Modbus ID: 有效范围: 1-128 |
| 0x06 | 6 | HEADING_MODE | 航向角模 式 | R/W | 0: 6轴模式(相对航向, 航向角上电为0). 1: 9轴模式(地磁融合, 绝对航向) |
| 0x34 | 52 | ACCX | 加速度X | R | 单位G(1G=1重力加速度), 比例因子: 0.00048828 |
| 0x35 | 53 | ACCY | 加速度Y | R | 单位G(1G=1重力加速度), 比例因子: 0.00048828 |
| 0x36 | 54 | ACCZ | 加速度Z | R | 单位G(1G=1重力加速度), 比例因子: 0.00048828 |
| 0x37 | 55 | GYRX | 角速度X | R | 单位deg/s, 比例因子: 0.061035 |
| 0x38 | 56 | GYRY | 角速度Y | R | 单位deg/s, 比例因子: 0.061035 |
| 0x39 | 57 | GYRZ | 角速度Z | R | 单位deg/s, 比例因子: 0.061035 |
| 0x3A | 58 | MAGX | 磁强度X | R | 单位uT, 比例因子: 0.030517 |
| 0x3B | 59 | MAGY | 磁强度Y | R | 单位uT, 比例因子: 0.030517 |
| 0x3C | 60 | MAGZ | 磁强度Z | R | 单位uT, 比例因子: 0.030517 |
| 0x3D | 61 | R_H | 横滚角高 16位 | R | 单位deg, 比例因子: 0.001 |
| 0x3E | 62 | R_L | 横滚角低 16位 | R | 单位deg, 比例因子: 0.001 |
| 0x3F | 63 | P_H | 俯仰角高 16位 | R | 单位deg, 比例因子: 0.001 |
| 0x40 | 64 | P_L | 俯仰角低 16位 | R | 单位deg, 比例因子: 0.001 |
| 0x41 | 65 | Y_H | 航向角高 16位 | R | 单位deg, 比例因子: 0.001 |
| 0x42 | 66 | Y_L | 航向角低 16位 | R | 单位deg, 比例因子: 0.001 |
| 0x43 | 67 | TEMP | 温度 | R | 单位°C, 比例因子: 0.01 |
| 0x44 | 68 | PRS_H | 气压高 16位 | R | 单位Pa, 比例因子: 0.01 |
| 0x45 | 69 | PRS_L | 气压低 16位 | R | 单位Pa, 比例因子: 0.01 |
| 0x46 | 70 | Q0 | 四元数 QW | R | 四元数, 比例因子: 0.0001 |
| 0x47 | 71 | Q1 | 四元数 QX | R | 四元数, 比例因子: 0.0001 |
| 0x48 | 72 | Q2 | 四元数 QY | R | 四元数, 比例因子: 0.0001 |

| 地址 (Hex) | 地址 (Dec) | 名称 | 功能 | R/W | 说明 |
|---------------|-------------|------------|-------------|-----|---|
| 0x49 | 73 | Q3 | 四元数 QZ | R | 四元数，比例因子: 0.0001 |
| 0x4A | 74 | SA_X | 倾角仪X 轴角度 | R | 双轴倾角仪产品:X角度, ± 180 , 单位deg, 比例因子: 0.005493 单轴倾角仪产品:X角度, 0-360, 单位deg, 比例因子: 0.005493 |
| 0x4B | 75 | SA_Y | 倾角仪Y 轴角度 | R | 双轴倾角仪Y角度, ± 90 , 单位deg,比例因子: 0.005493 |
| 0x70- 0x77 | 112- 119 | PNAME | 设备名 | R | 设备名字符串, ASCII码, 共占8个寄存器 |
| 0x78 | 120 | SW_VERSION | 软件版本 | R | 软件版本 |
| 0x79 | 121 | BL_VERSION | BL版本 | R | BL版本 |
| 0x7F- 0x82 | 127- 130 | SN | 产品唯一 序列号 | R | 产品唯一序列号, 占4寄存器 |
| 0xA5 | 165 | SET_LV | 自动校平 | W | 3: 执行一次自动调平: 如果当前俯仰角/横滚角接近 $0^{\circ}, 0^{\circ}$ (水平正 面放置), 则自动校准到0,0. 前如果当前俯仰角/横滚角接近 0° 或 180° (水平倒放), 则自动校准到 $0^{\circ}, 180^{\circ}$ 适用于机器人安装环 境。其中, “接近” 定义为 Pitch Roll均小于 15° 5: 取消自动调平, 恢复绝对测量角度 其他值: 无效 |
| 0xA6 | 166 | URFR | 安装设置 | W | 0: 设置为水平安装(默认模式) 1: 垂直安装:Y轴正方向朝下 2: 垂直安装:Y轴正方向朝上 3: 垂直安装:X轴正方向朝上 4: 垂直安装:X轴正方向朝下 |

4.3 常用配置

以下所有配置示例默认Modbus地址为0x50(出厂默认), 如果Modbus ID已经被用户修改, 则ID字段和CRC字段需要更改。

4.3.1 控制寄存器说明(地址:0x0000)

| 命令 | CTL寄存器写入值 | 命令(Hex) ID=0X50 |
|----------------|-----------|-------------------------|
| 保存所有配置参数到Flash | 0x0000 | 50 06 00 00 00 00 84 4B |
| 恢复出厂设置 | 0x0001 | 50 06 00 00 00 01 45 8B |
| 复位 | 0x00FF | 50 06 00 00 00 FF C4 0B |

4.3.2 配置波特率

| 配置目标波特率 | 指令 |
|---------|-------------------------|
| 4800 | 50 06 00 04 00 00 C5 8A |
| 9600 | 50 06 00 04 00 01 04 4A |
| 19200 | 50 06 00 04 00 02 44 4B |
| 38400 | 50 06 00 04 00 03 85 8B |
| 57600 | 50 06 00 04 00 04 C4 49 |
| 115200 | 50 06 00 04 00 05 05 89 |
| 230400 | 50 06 00 04 00 06 45 88 |
| 460800 | 50 06 00 04 00 07 84 48 |
| 921600 | 50 06 00 04 00 08 C4 4C |

4.4 读取模块版本信息

1.读取模块产品名, 软件版本及SN号

TX(主机发送): 50 03 00 70 00 14 49 9F

ID=0x50, CMD=0x03, 读取起始地址0x70, 读取长度:0x14, CRC: 0x499F

RX(从机响应): 50 03 28 48 49 31 34 52 32 4E 2D 34 38 35 2D 30 30 30 00 00 98 00 6B 00 00 00 00 00 00 00 00 04 7D 95 5F 8D 2A 17 08 00 00 4D 0C

50 03 28: 从机地址0x50, 命令码:0x03, 数据部分共0x28= 40字节, 48 49 31 34 52 32 4E 2D 34 38 35 2D 30 30 30 00 00 98 00 6B 00 00 00 00 00 00 00 00 00 00 04 7D 95 5F 8D 2A 17 08 00 00 ,数据段: 产品名: CH10x(M), 软件版本:0x98(152), BL版本:0x6B(107), SN:047D955F8D2A1708, 4D 0C: CRC校验

4.5 读取传感器数据

TX(主机发送): 50 03 00 34 00 18 09 8F

ID=0x50, CMD=0x03, 读取起始地址0x34, 读取长度:0x18, CRC: 0x098F

RX(从机响应): 50 03 30 FF 01 03 B0 06 50 FC C9 FF 7C 00 91 01 D5 FD DB FD 27 00 00 21 FF 00 00 7F F6
FF FD 73 E7 00 00 00 00 00 10 A6 0D 59 DD 4E 86 A8 06 30 17 82 1E CE

0x30 = 从机返回48字节, 第一个寄存器值为0xFF01=-255, 第二寄存器值为0x03B0=944 ...以此类推

- 加速度: X=-255, Y=944, Z=1616 -> 乘比例因子后结果: X= -0.1245, Y=0.4609, Z=0.7891G (1G=1重力加速度, 可取 9.8m/s^2)
- 角速度: X=-823, Y=-132, Z=145 -> 乘比例因子后结果: X=-50.2318, Y=-8.0566, Z=8.8501deg/s
- 磁场: X=469, Y=-549, Z=-729 -> 乘比例因子后结果: X=14.3125, Y=-16.7538, Z=-22.2469uT
- 欧拉角: 横滚(Roll) =8703, 俯仰角(Pitch)=32758, 航向角(Yaw) =-166937 -> 乘比例因子后结果: Roll= 8.703deg, Pitch=32.758deg, Yaw=-166.937deg

4.6 Modbus ID自动分配

ID地址自动分配机制用于量产部署时多个模块挂接在同一个485总线的情况, 模块可配合上位机完成ID自动地址分配, 该功能只开放给量产客户, 具体资料请与我们联系。

0x50 自定义指令格式为: [ADDR] 0x50 [SUB_CMD] [DATA_LEN] [DATA]

目前支持的自定义指令列表:

1. 通过SN号设定ID地址(0x0031): 格式: 00 50 00 31 00 0A [SN] [NEW_ADDR] CRC

SN: 设备唯一序列号,8字节

新的ID地址: 1-255, 2字节

2. ID地址随机化生成(0x0030): 该指令会强制总线上的所有模块抛弃原来ID且生成新的介于MIN_ADDR和MAX_ADDR之间的新ID 格式: 00 50 00 30 00 06 [MIN] [MAX] FF FF CRC

MIN: 生成随机ID的最小值, 2字节。

MAX: 生成随机ID的最大值 2字节。

5. CAN数据协议与指令(CANopen协议)

CAN接口符合CANopen协议，所有通讯均使用标准数据帧，使用TPDO1-7 传输数据。不接收/发送远程帧和拓展数据帧，所有TPDO采用异步定时触发模式。

5.1 CANopen 默认设置

| 默认配置 | 值 |
|-------------|---------------------|
| CAN 波特率 | 500KHz |
| CANopen节点ID | 8 |
| 初始化状态 | Operational |
| 心跳包 | 无 |
| TPDO输出速率 | 1Hz - 200Hz(每个TPDO) |

5.2 CANopen TPDO

| 通道 | 帧ID | 数据长度 (DLC) | 传输方 式 | 输出频 率 (Hz) | 数据 | 说明 |
|-------|----------|---------------|--------------------|------------------|---------------|---|
| TPDO1 | 0x180+ID | 6 | 异步定 时 (0xFE) | 100 | 加速 度 | 类型:int16 低字节在前，每个轴2字节，共6字节 分别为X,Y,Z轴加速度，单位为mG(0.001G) |
| TPDO2 | 0x280+ID | 6 | 异步定 时 (0xFE) | 100 | 角速 度 | 类型:int16 低字节在前，每个轴2字节，共6字节 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) | 100 | 欧拉 角 | 类型:int16 低字节在前，每个轴2字节，共6字节 顺序分别为 横滚角:Roll, 俯仰角:Pitch, 航向角:Yaw。单 位为0.01° |
| TPDO4 | 0x480+ID | 8 | 异步定 时 (0xFE) | 100 | 四元 数 | 类型:int16 低字节在前,每个元素2字节，共8字节 分别为 q_w q_x q_y 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. |
| TPDO6 | 0x680+ID | 4 | 异步定 时 (0xFE) | 20 | 气压 | 类型:int32 共4字节。单位Pa |
| TPDO7 | 0x780+ID | 8 | 异步定 时 (0xFE) | 100 | 倾角 仪角 度 | 类型:int32 低字节在前，每个轴4字节，共8字节 顺序分别为 X轴, Y轴。单位为0.01° |

以加速度和角速度为例解析数据

加速度CAN帧：ID=0x188，DATA = 4A 00 1F 00 C8 03

- ID=0x188: ID为8的设备发送的加速度数据帧

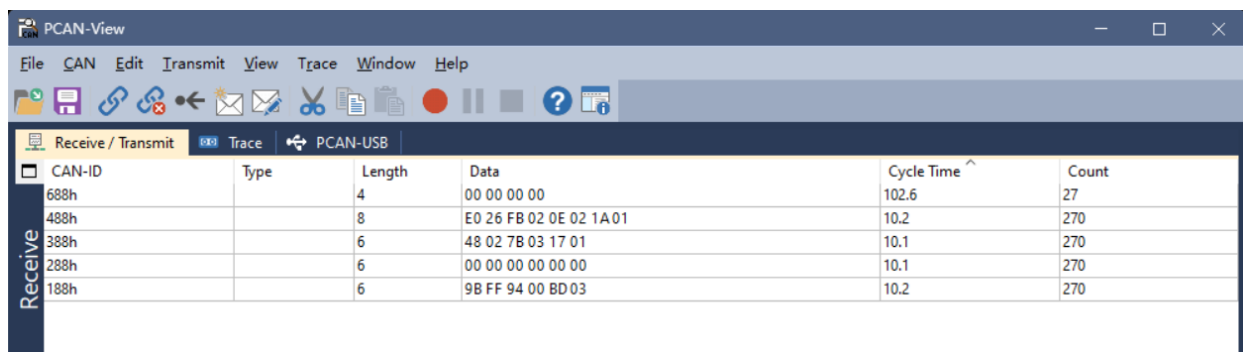
- 加速度X轴 = 0x004A = 74 = 74mG
- 加速度Y轴 = 0x001F = 31 = 31mG
- 加速度Z轴 = 0x03C8 = 968 = 968mG

角速度CAN帧: ID=0x288, DATA = 15 00 14 01 34 00

- ID=0x288: ID为8的设备发送的角速度数据帧
- 角速度X轴 = 0x0015 = 21 = 2.1dps
- 角速度Y轴 = 0x0114 = 276 = 27.6dps
- 角速度Z轴 = 0x0034 = 52 = 5.2dps

5.3 使用上位机连接CAN设备

使用PCAN-View工具, 配合PCAN, 可以在接收框(Rx Message)中显示收到的CAN消息及帧率, 如下图所示:



The screenshot shows the PCAN-View application window. The 'Receive / Transmit' tab is active, and the 'Trace' window displays a list of received messages. The messages are filtered by ID (688h, 488h, 388h, 288h, 188h). The table shows the CAN-ID, Type, Length, Data, Cycle Time, and Count for each message.

| CAN-ID | Type | Length | Data | Cycle Time ^ | Count |
|--------|------|--------|-------------------------|--------------|-------|
| 688h | | 4 | 00 00 00 00 | 102.6 | 27 |
| 488h | | 8 | E0 26 FB 02 0E 02 1A 01 | 10.2 | 270 |
| 388h | | 6 | 48 02 7B 03 17 01 | 10.1 | 270 |
| 288h | | 6 | 00 00 00 00 00 00 | 10.1 | 270 |
| 188h | | 6 | 9B FF 94 00 BD 03 | 10.2 | 270 |

5.4 配置指令

所有配置更改后需要发送保存配置指令才能保存到Flash

5.4.1 LLS(Layer Setting Services)协议

- 主机发送LLS命令到从机: ID = 0x07E5, DATA = CS,00,00,00,00,00,00,00 其中CS为命令字段, DATA为数
据字段
- 从机回复LLS命令到主机: ID = 0x07E4, DATA = CS, ECODE, 00, 00, 00, 00, 00, 00, 00 其中ECODE为错误
码, 0表示无错误。

5.4.1.1 修改波特率(重新上电生效)

- CAN波特率修改为1000 kbit/s: ID=0x07E5, DATA=13,00,00,00,00,00,00,00
- CAN波特率修改为800 kbit/s: ID=0x07E5, DATA=13,00,01,00,00,00,00,00
- CAN波特率修改为500 kbit/s: ID=0x07E5, DATA=13,00,02,00,00,00,00,00
- CAN波特率修改为250 kbit/s: ID=0x07E5, DATA=13,00,03,00,00,00,00,00
- CAN波特率修改为125 kbit/s: ID=0x07E5, DATA=13,00,04,00,00,00,00,00
- CAN波特率修改为100 kbit/s: ID=0x07E5, DATA=13,00,05,00,00,00,00,00

5.4.1.2 修改节点ID

ID=0x07E5, DATA=11,ID,00,00,00,00,00,00

ID修改范围：1-64，生效后发送启动节点命令(比如节点启动命令数据变为01 09)和SDO指令(发送CAN帧ID变为0x609时注意为新的地址)

5.4.1.3 保存配置

ID=0x07E5, DATA=17,00,00,00,00,00,00,00

所有配置更改后需要发送保存配置指令才能保存到Flash

5.4.2 NMT(Network Management)协议

5.4.2.1 全局使能/关闭数据输出(开启异步触发)

使用CANopen NMT协议帧:

- 使能全局数据输出 ID=0x00, DATA=01 08
- 关闭全局数据输出 ID=0x00, DATA=02 08

5.4.2.2 复位

- 复位节点 ID=0x00, DATA=81 08

5.4.3 SDO(Service Data Object)协议

SDO格式:

- 主机发送SDO命令到从机:

| CAN_ID | CS命令符(1B) | 数据字典索引(2B) | 子索引(1B) | 数据(4B) |
|----------|-----------|------------|---------|----------|
| 0x600+ID | 0x23(写4B) | 低位在前 | 子索引 | 数据, 低位在前 |

- 从机回复SDO命令到主机:

| CAN_ID | SDO命令(1B) | 数据字典索引(2B) | 子索引(1B) | 数据(4B) |
|----------|-------------|------------|---------|----------|
| 0x580+ID | 0x60(写成功应答) | 低位在前 | 子索引 | 数据, 低位在前 |

以下配置操作均使用快速SDO来写数据字典, 其中TPDO通道与其对应的参数索引为:

| 通道 | 帧ID | 参数索引地址 | 说明 |
|-------|----------|--------|-------|
| TPDO1 | 0x180+ID | 0x1800 | 加速度 |
| TPDO2 | 0x280+ID | 0x1801 | 角速度 |
| TPDO3 | 0x380+ID | 0x1802 | 欧拉角 |
| TPDO4 | 0x480+ID | 0x1803 | 四元数 |
| TPDO6 | 0x680+ID | 0x1804 | 气压 |
| TPDO7 | 0x780+ID | 0x1805 | 倾角仪输出 |

5.4.3.1 修改/关闭/开启数据输出速率

此项配置立即生效

- ID=0x608, DATA=23,00,18,05,00,00,00,00 关闭加速度输出
- ID=0x608, DATA=23,00,18,05,05,00,00,00 加速度200Hz输出
- ID=0x608, DATA=23,00,18,05,0A,00,00,00 加速度100Hz输出
- ID=0x608, DATA=23,00,18,05,14,00,00,00 加速度50Hz输出
- ID=0x608, DATA=23,00,18,05,32,00,00,00 加速度20Hz输出
- ID=0x608, DATA=23,00,18,05,64,00,00,00 加速度10Hz输出(最低10Hz)
- ID=0x608, DATA=23,01,18,05,00,00,00,00 关闭角速度输出
- ID=0x608, DATA=23,01,18,05,05,00,00,00 角速度200Hz输出
- ID=0x608, DATA=23,01,18,05,0A,00,00,00 角速度100Hz输出
- ID=0x608, DATA=23,01,18,05,14,00,00,00 角速度50Hz输出
- ID=0x608, DATA=23,01,18,05,32,00,00,00 角速度20Hz输出
- ID=0x608, DATA=23,01,18,05,64,00,00,00 角速度10Hz输出(最低10Hz)
- ID=0x608, DATA=23,02,18,05,00,00,00,00 关闭欧拉角输出
- ID=0x608, DATA=23,02,18,05,05,00,00,00 欧拉角200Hz输出
- ID=0x608, DATA=23,02,18,05,0A,00,00,00 欧拉角100Hz输出
- ID=0x608, DATA=23,02,18,05,14,00,00,00 欧拉角5Hz输出
- ID=0x608, DATA=23,02,18,05,32,00,00,00 欧拉角20Hz输出
- ID=0x608, DATA=23,02,18,05,64,00,00,00 欧拉角10Hz输出(最低10Hz)
- ID=0x608, DATA=23,03,18,05,00,00,00,00 关闭四元数输出
- ID=0x608, DATA=23,03,18,05,05,00,00,00 四元数200Hz输出
- ID=0x608, DATA=23,03,18,05,0A,00,00,00 四元数100Hz输出
- ID=0x608, DATA=23,03,18,05,14,00,00,00 四元数50Hz输出
- ID=0x608, DATA=23,03,18,05,32,00,00,00 四元数20Hz输出
- ID=0x608, DATA=23,03,18,05,64,00,00,00 四元数10Hz输出(最低10Hz)

- ID=0x608, DATA=23,04,18,05,00,00,00,00 关闭气压输出
- ID=0x608, DATA=23,04,18,05,05,00,00,00 气压200Hz输出
- ID=0x608, DATA=23,04,18,05,0A,00,00,00 气压100Hz输出
- ID=0x608, DATA=23,04,18,05,14,00,00,00 气压50Hz输出
- ID=0x608, DATA=23,04,18,05,32,00,00,00 气压20Hz输出
- ID=0x608, DATA=23,04,18,05,64,00,00,00 气压10Hz输出(最低10Hz)

以 TPDO1(加速度)输出速率为100Hz(每10ms输出一次)为例: 0x23为SDO写四个字节指令。0x00, 0x18为写0x1800索引。0x05为子索引。0x00, 0x0A= (0x00<<8) + 0x0A = 10(单位为ms), 后面不足补0。

5.4.3.2 设置倾角仪输出正负号

- ID=0x608, DATA=23,9E,20,00,00,00,00,00 X轴正负号为出厂默认方向
- ID=0x608, DATA=23,9E,20,00,01,00,00,00 X轴正负号反向
- ID=0x608, DATA=23,9F,20,00,00,00,00,00 Y轴正负号为出厂默认方向
- ID=0x608, DATA=23,9F,20,00,01,00,00,00 Y轴正负号反向

5.4.3.3 设置倾角仪零点(相对位置输出)

- ID=0x608, DATA=23,A5,20,00,02,00,00,00 写入后即设置当前位置为输出零点(X=0,Y=0)
- ID=0x608, DATA=23,A5,20,00,05,00,00,00 写入后取消零点配置, 输出真实的X, Y角度(相当于X,Y offset=0)

5.4.3.4 设置倾角输出分辨率(定制功能)

- ID=0x608, DATA=23,A2,20,00,00,00,00,00 倾角数据输出分辨率为 0.01°(默认)
- ID=0x608, DATA=23,A2,20,00,01,00,00,00 倾角数据输出分辨率为 0.02°
- ID=0x608, DATA=23,A2,20,00,02,00,00,00 倾角数据输出分辨率为 0.1°

5.4.4 同步协议

5.4.4.1 配置TPDO为同步模式

- 关闭所有TPDO(设置TPDO输出速率为0),
- 发送CANopen同步帧, CANopen 同步帧: ID:80, DATA:空

6. CAN数据协议与指令(SAE-J1939协议)

模块默认输出协议为CANOpen, 如需SAE J1939协议, 请联系我司。

| PGN | 描述 |
|------------------|-------------------------------------|
| 通讯模式 | 广播通信 |
| 默认传输时间间隔 | 100ms |
| 数据长度 | 每个PGN8 字节 |
| PF(PDU format) | 0xFF |
| PS(PDU specific) | PF > 0xF0时为拓展PGN地址(GE), 否则为目的地址(DA) |
| 优先级 | 3 |
| 默认J1939地址 | 0x08 |
| 数据格式 | 所有帧中数据格式采用LSB(低位在前),无特殊说明均为有符号整型 |

6.1 PGN消息列表

6.1.1 PGN65332(FF34) 加速度

CANID=0x0CFF3408

| 名称 | 位置(byte) | 说明 |
|------|----------|---------------------------------|
| 加速度X | 0-1 | 单位G(1G=1重力加速度), 比例因子:0.00048828 |
| 加速度Y | 2-3 | 单位G(1G=1重力加速度), 比例因子:0.00048828 |
| 加速度Z | 4-5 | 单位G(1G=1重力加速度), 比例因子:0.00048828 |
| 保留 | 6-7 | - |

6.1.2 PGN65335(FF37) 角速度

CANID=0x0CFF3708

| 名称 | 位置(byte) | 说明 |
|------|----------|------------------------|
| 角速度X | 0-1 | 单位deg/s, 比例因子:0.061035 |
| 角速度Y | 2-3 | 单位deg/s, 比例因子:0.061035 |
| 角速度Z | 4-5 | 单位deg/s, 比例因子:0.061035 |
| 保留 | 6-7 | |

6.1.3 PGN65345(FF41) 航向角

CANID=0x0CFF4108

| SPN 名称 | SPN 位置(byte) | 说明 |
|----------|--------------|------------------------------|
| 航向角(Yaw) | 0-3 | 0-360, 单位°, 比例因子:0.001,顺时针为正 |
| 保留 | 4-7 | |

6.1.4 PGN65354(FF4A) 倾角仪输出

CANID=0x0CFF4A08 (只适用于输出J1939协议的倾角仪产品)

| 名称 | 位置(byte) | 说明 |
|-------|----------|----------------------------------|
| X倾角角度 | 0-3 | 范围0-360或 ±180,单位:deg, 比例因子:0.001 |
| Y倾角角度 | 4-7 | 范围0-360或 ±180,单位:deg, 比例因子:0.001 |

6.2 配置指令

6.2.1 配置格式

主机发送: ADDR+ CMD + STATUS + VAL , 从机响应: ADDR+ CMD + STATUS + VAL

| 字段 | 大小(Byte) | 说明 |
|--------|----------|----------------------|
| ADDR | 2 | 寄存器地址 |
| CMD | 1 | 0x06:写入, 0x03:读取 |
| STATUS | 1 | 保留 |
| VAL | 4 | 当写入时: 写入的值, 当读取时: 保留 |

6.2.2 配置模块

| 29'b 拓展帧地址 | 数据 | 描述 | 说明 |
|------------|----------------------------|----------|--|
| 0x0CEF08xx | 34 01 06 00 [VAL] | VAL: 4字节 | PGN:FF34(加速度) 发送间隔, 单位ms, 范围:5 -1000 |
| 0x0CEF08xx | 37 01 06 00 [VAL] | VAL: 4字节 | PGN:FF37(角速度) 发送间隔, 单位ms, 范围:5 -1000 |
| 0x0CEF08xx | 41 01 06 00 [VAL] | VAL: 4字节 | PGN:FF41(航向角) 发送间隔, 单位ms, 范围:5 -1000 |
| 0x0CEF08xx | 4A 01 60 00 [VAL] | VAL: 4字节 | PGN:FF4A(倾角仪输出) 发送间隔, 单位ms, 范围:5 -1000 |
| 0x0CEF08xx | 00 00 06 00 00 00 00 00 | - | 保存所有配置参数到Flash |
| 0x0CEF08xx | 00 00 06 00 01 00 00 00 | - | 恢复出厂设置 |
| 0x0CEF08xx | 00 00 06 00 FF 00 00 00 | - | 复位 |
| 0x0CEF08xx | 9A 00 06 00 [VAL] | VAL: 4字节 | 配置波特率(保存设置, 复位生效): 0:1000K, 1:800K, 2:500K, 3:250K, 4:125K |
| 0x0CEF08xx | 9C 00 06 00 [VAL] | VAL: 4字节 | 设置J1939 节点ID: 1-128 |
| 0x0CEF08xx | A5 00 06 00 [VAL] | VAL: 4字节 | 设置零位, 0x02:设置当前位置为零位, 0x05: 取消零位设置, 输出绝对物理角度 |
| 0x0CEF08xx | 9E 00 06 00 [VAL] | VAL: 4字节 | 设置X轴正负方向, 0:默认 1:反向 |
| 0x0CEF08xx | 9F 00 06 00 [VAL] | VAL: 4字节 | 设置Y轴正负方向, 0:默认 1:反向 |
| 0x0CEF08xx | A3 00 06 00 [VAL] | VAL: 4字节 | (定制功能)设置XY倾角输出分辨率0: 0.001°(默认), 1: 0.01°, 2: 0.1° |

地址域中xx: J1939协议中的源地址, 可为任意字节。

数据域中xx: 任意字节

例: ID=0x0CEF0855, DATA = 37 01 06 00 64 00 00 00 : 将PGN:FF37设置为100ms周期(10Hz)

7. CAN数据协议与指令(NMEA2000协议)

定制协议, 如需NMEA2000协议, 请联系我司。

8. 地磁校准

8.1 地磁校准步骤

- 首先确认校准前提: 当前测试下航向角精度达不到要求。
- 模块安装环境有磁场干扰, 这种干扰是固定的, 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模块再分开的, 一旦分开是需要再重新校准。如果这个铁大小是不固定的, 或与模块的距离变化也不是固定的, 这种干扰是无法校准, 只能避而远之安装, 安全距离控制在 40CM 以上)

模块内部自带主动地磁校准系统, 不需要用户发送任何指令, 该系统在后台自动采集一段时间内地磁场数据, 并做分析比较, 剔除异常数据, 一旦数据足够, 就会尝试地磁校准。所以, 当使用地磁辅助(9轴)模式时, 不需要用户任何干预即可完成地磁校准。但是模块仍然提供接口来让用户检查当前校准状态。**自动校准的前提是需要模块有充分的姿态变化(缓慢的让模块经历尽量多的姿态变化)**, 内部校准系统才能搜集不同姿态下的地磁场信息, 从而完成校准, 静止状态下是无法进行地磁校准的。

当首次使用模块并且需要使用AHRS (9轴) 模式时, 应进行如下校准操作:



1. 首先需要将模块切换到9轴模式, 才能开启地磁校准。
2. 检查周围是否存在磁场干扰: 实验室铁质或者含有铁质的桌子、电脑、电机、手机等旁边都属于常见的干扰区域。建议将模块拿到室外空旷处, 即使没有条件拿到室外, 尽量将模块远离干扰源。
3. 在尽量小范围内(位置尽量不动, 只是旋转), 缓慢的让模块旋转, 让模块经历尽量多的姿态位置(每个轴至少都旋转360°, 持续约1分钟), 即可完成校准, 如果始终没能成功校准模块, 说明周围地磁场干扰比较大。
4. 如果客户安装位置改变(比如上一次校准是拿着模块单独去校准的, 使用的时候却是安装在目标设备上)。则需要带着目标设备进行重新校准。
5. 如何查看地磁校准结果: 输入 LOG MAGCONFIG, 返回结果如下:

```
1  MAG_MIS=
2    1.000  0.000  0.000
3    0.000  1.000  0.000
4    0.000  0.000  1.000
5  MAG_BIS=
6    15.723
7    -0.320
8    -14.014
9  OK
```

如果MAG_BIAS显示三个值不是0,0,0。代表校准成功。

8.2 磁干扰的知识

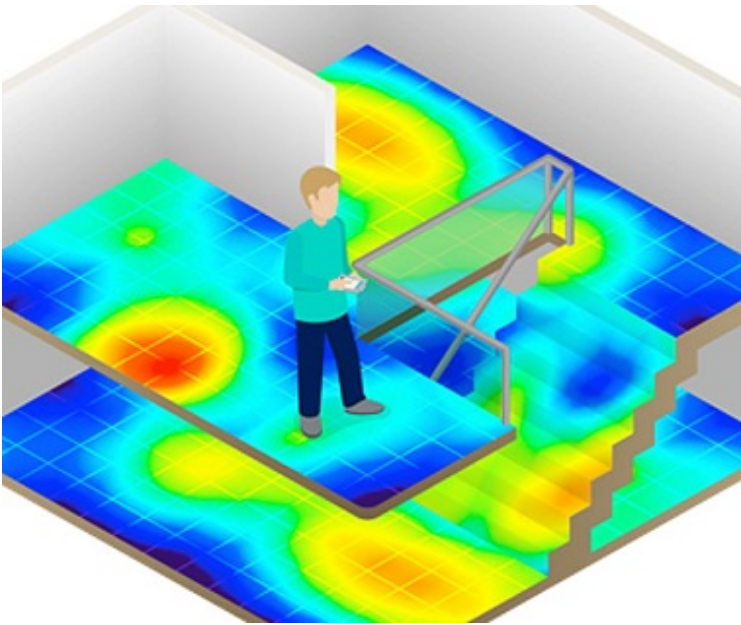
地磁干扰可分为空间磁场干扰与传感器坐标系下的磁场干扰，如下图所示

| Distortions that move with the sensor | Distortions that do not move with the sensor |
|---|---|
|  |  |
| <ul style="list-style-type: none">• Calibration errors• Hard iron effects• Soft iron effects• Etc. | <ul style="list-style-type: none">• Spatial distortions• Temporal distortions• Etc. |

8.2.1 空间磁场干扰

定义：磁场干扰不随传感器运动而运动，处于世界坐标系下

典型干扰源：各种固定的地磁干扰源，家具，家用电器，线缆，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，下图是典型的室内磁场分布图。



对模块的影响：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，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原因。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。

应对措施：只能尽量避免这种干扰源

8.2.2 传感器坐标系下的干扰

定义：地磁场干扰源随传感器运动而运动

典型干扰源：与模块固定在一起的PCB板子，仪器设备等。他们和磁传感器视为同一个刚体，随磁传感器运动而运动。

对模块的影响：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。

应对措施：对模块进行地磁校准。

8.3 地磁使用注意事项

- 1 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，但地磁辅助(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，实验室，车间，地下车库等)，即使校准后航向角精度可能还不如6轴模式甚至会出现大角度误差。
- 2
- 3 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：模块安装在一个导磁刚体 (机器人/机械设备/车辆/船舶等)之上，以机器人为例： 因为机器人金属材料会有磁场干扰，这时就需要把机器人与模块一起旋转校准，并且模块在使用当中是不会和机器人再分开的(发生相对位移)，一旦分开是需要再重新校准。

9. 技术支持

新产品与资料信息可以通过网站以及公众号获得

