

# 指令与编程手册

Rev 1.2

适用于HI02/HI04/HI14/HI15/HI9/HI7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 模块配置简述

- 1.1 地磁辅助场景(AHRS/9轴模式)
- 1.2 同步输入与同步输出(时间同步)
  - 1.2.1 数据同步触发(SIN)
  - 1.2.2 GPS时间同步(PPS)
  - 1.2.3 同步输出(SOUT)
- 1.3 船舶升沉位(Heave)移输出
  - 1.3.1 使用限制

### 2 产品特性支持表

### 3 模块配置命令(ASCII)

- 3.1 配置命令总览
- 3.2 指令详解
  - 3.2.1 REBOOT
  - 3.2.2 SAVECONFIG
  - 3.2.3 SERIALCONFIG
  - 3.2.4 CONFIG
    - 3.2.4.1 工作模式模式配置: 6轴 或者9轴(地磁辅助)模式
    - 3.2.4.2 水平校准
    - 3.2.4.3 坐标系旋转(改变安装方式)
    - 3.2.4.4 多功能IO复用功能配置
    - 3.2.4.5 SOUT同步输出分频设置
    - 3.2.4.6 用户级陀螺校准
  - 3.2.5 LOG
    - 3.2.5.1 ENABLE/DISABLE: 全局打开/关闭数据输出
    - 3.2.5.2 VERSION: 显示模块版本信息
    - 3.2.5.3 COMCONFIG: 显示串口配置信息
    - 3.2.5.4 设置数据帧输出类型及频率
  - 3.2.6 FRESET

### 4 RS-232/TTL/USB 数据协议(二进制)

- 4.1 数据帧格式
- 4.2 出厂默认输出
- 4.3 数据域内容
  - 4.3.1 浮点型IMU数据帧(HI91)
  - 4.3.2 整型IMU数据帧(HI92)
  - 4.3.3 STATUS 状态字说明
- 4.4 CRC
- 4.5 数据帧结构示例(以HI91为例)
- 4.6 C语言解析代码示例(以HI91为例)
- 4.7 最大传输速率

### 5 RS-485输出协议(Modbus)

- 5.1 数据帧格式
  - 5.1.1 读寄存器(0x03)
  - 5.1.2 写寄存器(0x06)

- 5.1.3 CRC校验
- 5.2 寄存器列表
- 5.3 常用配置
  - 5.3.1 控制寄存器说明(0x00)
  - 5.3.2 配置波特率(0x04)
  - 5.3.3 配置节点ID(0x05)
  - 5.3.4 设置安装方式 (0xA6)
  - 5.3.5 设置水平校准(0xA5)
  - 5.3.6 设置6轴或9轴模式(0x06)
- 5.4 读取模块版本信息(0x70-0x82)
- 5.5 读取传感器数据(0x34-0x4B)
- 6 CAN数据协议(CANopen)
  - 6.1 CANopen 默认设置
  - 6.2 CANopen TPDO
  - 6.3 使用上位机连接CAN设备
  - 6.4 配置指令(SDO协议)
    - 6.4.1 SDO(Service Data Object)协议
      - 6.4.1.1 修改节点ID (0x20A0)
      - 6.4.1.2 保存配置到Flash(0x2000)
      - 6.4.1.3 复位(0x2000)
      - 6.4.1.4 恢复出厂设置(0x2000)
      - 6.4.1.5 修改CAN波特率(0x209A)
      - 6.4.1.6 修改/关闭/开启数据输出速率(0x1800-0x1805)
      - 6.4.1.7 设置倾角仪输出正负号(0x209E)
      - 6.4.1.8 设置倾角仪零点(0x20A5)
    - 6.4.2 同步协议
      - 6.4.2.1 配置TPDO为同步模式
      - 6.4.2.2 发送CANopen同步帧
      - 6.4.2.3 设置心跳包
- 7 CAN数据协议(J1939)
  - 7.1 PGN消息列表
    - 7.1.1 PGN65327(FF2F) 时间信息
    - 7.1.2 PGN65332(FF34) 加速度
    - 7.1.3 PGN65335(FF37) 角速度
    - 7.1.4 PGN65341(FF3D) 俯仰横滚角
    - 7.1.5 PGN65345(FF41) 航向角
    - 7.1.6 PGN65354(FF4A) 倾角仪输出
  - 7.2 配置指令
    - 7.2.1 配置格式
    - 7.2.2 配置模块
- 8 地磁校准
  - 8.1 地磁校准步骤
    - 8.1.1 用户地磁校准
    - 8.1.2 空间磁场干扰

### 8.1.3 关于模块地磁校准算法

#### 8.2 再次强调

#### 9 附录1 四元数/欧拉角/旋转矩阵互转

##### 9.1 四元数转旋转矩阵

9.2 四元数转欧拉角 - 东北天(ENU)-312(先转Z,然后X轴,最后Y轴)旋转顺序下的欧拉角

9.3 四元数转欧拉角 - 北东地(NED)-321(先转Z轴,然后Y轴,最后X轴)旋转顺序下的欧拉角

#### 10 附录2 固件升级

#### 11 附录3 技术支持



# 1. 模块配置简述

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)

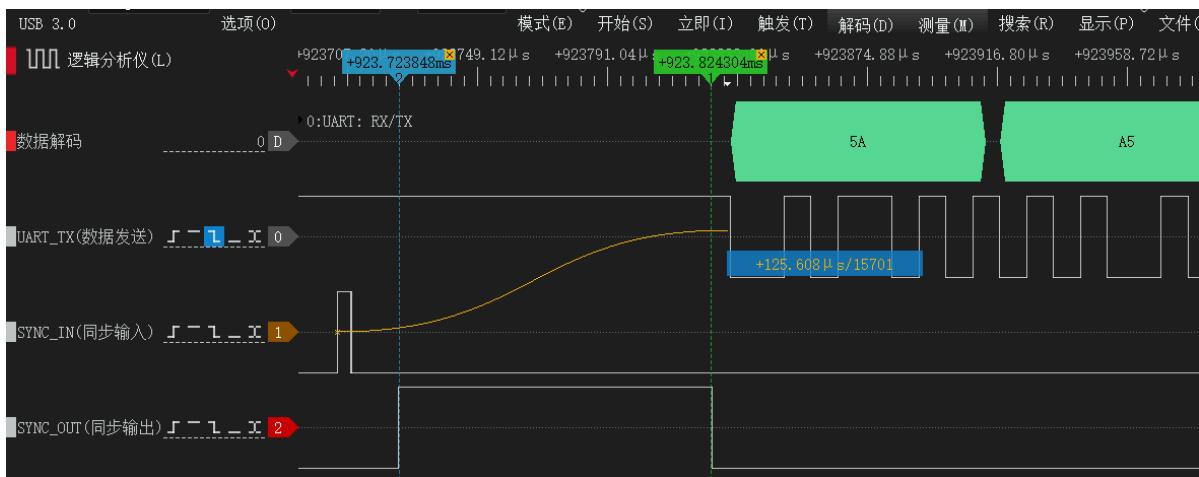
在绝对大多数情况下，机器人及室内环境，AHRS(9轴)模式很容易受到干扰导致航向角产生误差。在少数空旷且无磁场干扰的环境下，可以尝试使用地磁辅助模式，比如无人机，在使用前需要先将模块配置为地磁辅助模式并且进行地磁校准。详见地磁校准章节。

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(时间同步)

### 1.2.1 数据同步触发(SIN)

部分产品有同步输入引脚(SIN/PPS)，用于IMU时间同步，不使用时可悬空。当某个输出帧被配置为同步触发时(ONMARK触发，见LOG指令)，每当SIN引脚检测到上升沿，都会输出一帧该数据。该功能主要用于接收主控产生的高精度方波脉冲，以触发高频同步IMU数据。从SIN引脚检测到上升沿到数据帧发出的延迟为125us(见下图)。



### 1.2.2 GPS时间同步(PPS)

此功能同样使用(SIN/PPS)引脚，此引脚可接入GPS的PPS秒脉冲信号，配合串口的GPRMC消息为IMU提供UTC时间同步。一旦时间同步成功，所有数据帧中的本地时间戳将立即转换为UTC时间(以毫秒为单位)。此功能要求：

1. SIN/PPS必须接入有效的GNSS PPS秒脉冲，要求：
  - 周期：1秒，触发方式：上升沿，脉宽：1-100 ms
  - 对齐：上升沿对齐UTC时间整秒，电平：高电平不超过5V
2. 串口需要输入标准GPRMC消息：
  - 频率：1-10Hz，波特率：必须与IMU配置的波特率相同
  - 其他：仅支持RMC或GGA消息，串口数据线上不能存在GSV、GSA等消息

时间戳格式:时间同步后，数据帧中的时间戳(uint32\_t)表示从当日UTC 00:00:00开始的毫秒计数:

- 范围: 0-86,399,999 ms (对应00:00:00.000至23:59:59.999)
- 精度: 1毫秒

转换示例:

| 时间戳 (ms) | UTC时间        |
|----------|--------------|
| 0        | 00:00:00.000 |
| 3661000  | 01:01:01.000 |
| 43200000 | 12:00:00.000 |
| 86399999 | 23:59:59.999 |

代码示例:

```
//小时 = 时间戳 / 3600000
//分钟 = (时间戳 % 3600000) / 60000
//秒 = (时间戳 % 60000) / 1000
//毫秒 = 时间戳 % 1000

void ms_to_utc_time(uint32_t total_ms, char *utc_time, int buf_size)
{
    // Calculate hours, minutes, seconds and milliseconds
    uint32_t total_seconds = total_ms / 1000;
    uint32_t ms_part = total_ms % 1000;

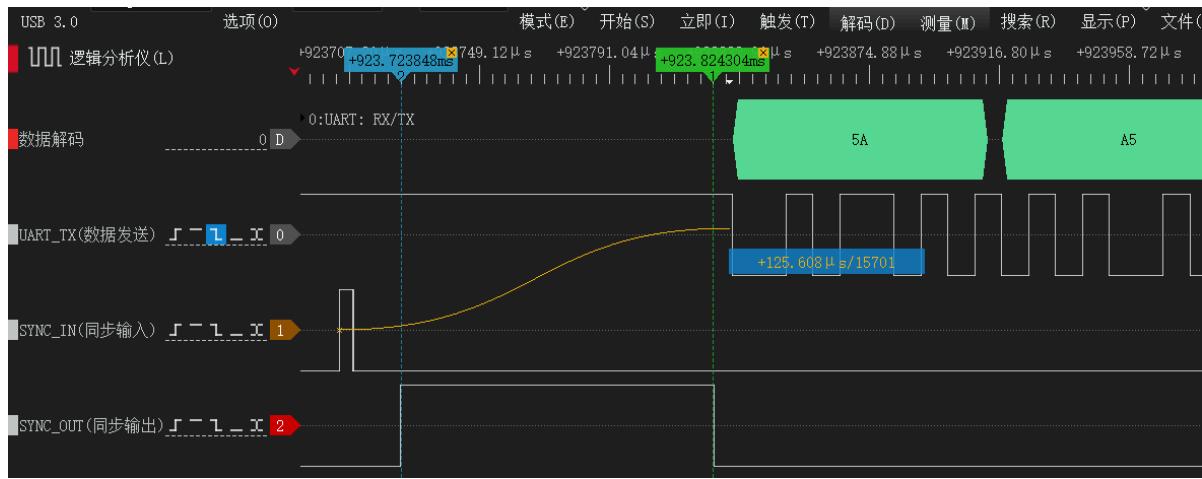
    uint8_t hours = (total_seconds / 3600) % 24;
    uint8_t minutes = (total_seconds % 3600) / 60;
    uint8_t seconds = total_seconds % 60;

    // Format as hh:mm:ss.sss
    snprintf(utc_time, buf_size, "%02d:%02d:%02d.%03d",
             hours, minutes, seconds, ms_part);
}
```

注意事项: 时间戳不包含日期信息，仅表示当日内的UTC时间, 不包含时区信息，如需本地时间需应用时区偏移.

### 1.2.3 同步输出(SOUT)

- 数据同步输出(SYNC\_OUT): 输出引脚, 无数据输出时为低电平(空闲), 一帧数据开始发送前会发送一个高脉冲, 高脉冲结束(下降沿)后紧接数据输出, 见下图。



另外,同步输出脉冲信号还可以进行分频输出(如每2帧,4帧,8帧数据输出一次脉冲),用于触发相机等低速率设备,进行严格时钟同步.

- 在使用同步输入功能时, 先关闭所有异步定时输出消息, 然后再使用 LOG <MSG> ONMARK 1 打开同步触发输入功能, 详情见请参见LOG命令

### 1.3 船舶升沉位(Heave)移输出

本产品支持测量海浪引起的周期性船舶升沉(Heave), 可为提供高精度的实时的海浪垂直运动信息. 可输出由海浪引起的垂直位移, 垂直速度, 海浪周期.

应用场景包括:

- 船舶动态监测
- 海洋工程作业
- 波浪特性研究
- 海上平台稳定性分析

#### 1.3.1 使用限制

- 本产品无法测量周期很长的线性运动或阶跃运动, 因为它假设垂荡位置的平均值为0, heave输出仅可用于海浪等周期往复运动的模型, 无法测量长期的静态位移或阶跃式位移, 只适用于周期性的波浪运动测量.
- 升沉输出需要稳定的5-20个海浪周期的初始稳定时间才能获得正确的结果.
- 不适用于测量:
  - 极慢速水位变化(周期 > 30s )
  - 装卸载导致的吃水变化

仅海洋类产品支持此功能.

## 2. 产品特性支持表

| 型号/指令        | HI02 | HI05 | CH1XX/CH0XX | HI04 | HI14 | HI7X/HI9X |
|--------------|------|------|-------------|------|------|-----------|
| 地磁辅助/地磁校准    | X    | ●    | ●           | ●    | ●    | ●         |
| 用户水平校准       | X    | ●    | ●           | ●    | ●    | ●         |
| 设置安装方式       | ●    | ●    | ●           | ●    | ●    | ●         |
| IO复用动态映射     | X    | ●    | X           | X    | X    | X         |
| CAN通讯        | X    | ●    | ●           | ●    | ●    | ●         |
| Modbus 485协议 | X    | ●    | ●           | ●    | ●    | ●         |
| 支持用户陀螺校准     | X    | ●    | ●           | ●    | ●    | ●         |
| 船舶升沉信息输出     | X    | X    | X           | X    | X    | ●         |

其他未列特性: 所有产品均支持

## 3. 模块配置命令(ASCII)

模块配置采用串口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。

### 3.1 配置命令总览

| 指令           | 功能            | 备注     |
|--------------|---------------|--------|
| REBOOT       | 复位模块          | 和重上电等效 |
| 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   |
| 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

### 3.2 指令详解

#### 3.2.1 REBOOT

复位模块，立即生效，和重新上电效果相同。

#### 3.2.2 SAVECONFIG

保存所有用户配置到Flash。

#### 3.2.3 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。
- 配置波特率指令立即生效

### 3.2.4 CONFIG

用于配置模块工作参数，且需要 **SAVECONFIG** 才能掉电保存

#### 3.2.4.1 工作模式模式配置: 6轴 或者9轴(地磁辅助)模式

- **CONFIG ATT MODE 0** 配置模块为6DOF(6轴)模式
- **CONFIG ATT MODE 1** 配置模块为AHRS(9轴)模式

#### 3.2.4.2 水平校准

- **CONFIG ATT RST 2** 设置相对零点: 将当前Pitch/Roll角度设置为零。
- **CONFIG ATT RST 5** 取消水平校平: 清除相对俯仰横滚角度。
- **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°。 其中, “接近”定义为 Pitch Roll 均小于5°

- 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
- **ATT RST 2** 和 **ATT RST 3** 的区别: **ATT RST 3** 在执行校平时会检测当前是否只是轻微倾斜, 如果倾斜度很大, 则无法执行成功, 而**RST 2**不会做任何执行前检查, 强制执行.

#### 3.2.4.3 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 =

|   |   |    |                  |
|---|---|----|------------------|
| 1 | 0 | 0  | (转换后的X = 转换前的X)  |
| 0 | 0 | -1 | (转换后的Y = 转换前的-Z) |
| 0 | 1 | 0  | (转换后的Z = 转换前的Y)  |

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:

|   |    |   |
|---|----|---|
| 1 | 0  | 0 |
| 0 | 0  | 1 |
| 0 | -1 | 0 |

#### 3.2.4.4 多功能IO复用功能配置

模块具有多个多功能引脚: IO1-IO9这些引脚可用作不同的功能, 不同功能之间通过配置命令切换。

CONFIG <PMUX> <IO>

- PMUX: 复用功能: PMUX1 - PMUX5
- IO: 引脚号: IO1-IO9

例:

- 将IO2分配为LED(PMUX3)功能: CONFIG PMUX3 IO2

| 复用功能<br>编号 | 复用名<br>称 | 方<br>向 | 说明   | 默认分<br>配IO |
|------------|----------|--------|--|------------|
| PMUX1      | SIN      | I      | 同步脉冲输入(SIN/PPS): 输入引脚。见同步输入与同步输出章节                     | IO1        |
| PMUX2      | SOUT     | O      | 同步输出: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,用于同步数据 | IO2        |
| PMUX3      | LED      | O      | 输出,运行状态指示灯   | IO5        |

- 并不是所有IO引脚在本产品上都引出,具体参见产品用户手册硬件部分。

### 3.2.4.5 SOUT同步输出分频设置

SOUT 同步输出可以进行分频,用于触发相机等低速率传感器融合设备.格式为:

**CONFIG PMUX2 DIV <N>**

- N为分频系数,范围: 1 - 100. 默认为1(不分频)
- 例: `CONFIG PMUX2 DIV 5`, 设置SOUT分频系数为5, 若数据输出速率为100Hz, 则SOUT脉冲输出频率为20Hz

### 3.2.4.6 用户级陀螺校准

陀螺仪测量精度会受到时间老化(性能缓慢衰减)、安装应力(PCB焊接变形)、温度变化和机械应力(振动冲击)等因素影响。本产品支持X、Y、Z三轴独立校准, 可提升旋转角度测量精度(校准后可保证比例因子误差在0.1%以内), 校准参数将永久保存在Flash中, 并具备实时质量监测功能确保校准有效性。

#### 校准方式说明:

- Z轴校准: 可直接放置在水平面上进行, 无需特殊设备, 适用于AGV, 大型机器这种无法特定安装工装的情况, 但是请不要使用手或手动转台去校准, 此种方法无法保证校准精度.
- X/Y轴校准: 需要使用可以精确转动的装置(如专业转台), 确保转动轴与待校准轴对齐, 如果没有专业转台设备, 不建议对X/Y轴进行校准.
- 三轴校准互相独立, 可只校准任意一个轴, 校准时无需指定哪个轴旋转, 程序会自动判断, 校准成功后立即生效, 掉电保存.

#### 校准步骤:

##### 1. 准备工作

- Z轴校准: 将模块放置在水平, 平整地面上.
- X/Y轴校准: 将模块固定在转动装置上, 确保转动轴与待校准轴对齐.

- 确保环境温度稳定，避免强烈振动干扰。

### 2. 发送开始命令: `CONFIG USRCAL START <ANGLE>`

- <ANGLE>: 校准角度(720-1800度, 即2-5圈, 圈数越多精度越高)
- 示例: `CONFIG USRCAL START 720` 开始2圈校准(可以是X,Y,Z中任意一个轴)

### 3. 执行校准动作

- 速度要求: 20-100 deg/s (建议50 deg/s, 大约5-6秒转一圈, 旋转方向任意)
- 校准: 通过转台或装置匀速转动
- 整个过程需保持速度均匀, 避免中途停顿, 突然加减等

### 4. 发送结束命令: `CONFIG USRCAL STOP`

- 返回"OK": 校准成功, 新参数自动保存生效
- 返回"ERR": 校准失败, 继续使用原参数

## 常见校准失败原因:

- 角度偏差过大: 实际转动角度与设定值相差超过5%
- 旋转操作不规范: Z轴: 确保模块放置平稳, 转动时保持水平, X/Y轴: 确保转动轴与待校准轴准确对齐。
- 操作不当: 转动太快或太慢, 中途停顿。
- 环境干扰: 避免在有振动的设备附近操作, 机器人电机振动过大等。
- 校准后误差依然存在甚至更大: 最有可能的原因是校准操作不规范(见1,2,3,4). 校准是一个精密的过程, 假设标定时规定旋转720°, 实际确旋转了725°, 则比例因子则会错误的标定为 $5/720 = 0.6\%$ , 这已经远超出厂精度。请确保设备规范, 校准过程精确, 否则不建议执行用户校准。

## 3.2.5 LOG

### 3.2.5.1 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

### 3.2.5.2 VERSION: 显示模块版本信息

`LOG VERSION` 打印固件版本信息

### 3.2.5.3 COMCONFIG: 显示串口配置信息

`LOG COMCONFIG` 打印串口及输出协议配置信息

### 3.2.5.4 设置数据帧输出类型及频率

`LOG <MSG> <TYPE> <VALUE>`

## 定时输出

- MSG:** HI91, HI92
- TYPE:** 固定为: ONTIME
- VALUE:** 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

示例:

- `LOG HI91 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)
- `LOG HI91 ONTIME 0` 关闭 91 数据包输出

## 同步(外部触发)输出

- **MSG:** HI91, HI92
- **TYPE:** 固定为: ONMARK
- **VALUE:** 固定为 1

示例:

- `LOG HI91 ONMARK 1` 将当前串口的 91 数据包配置为同步输出模式, 当 SIN/PPS 引脚有脉冲输入时, 触发一帧
- `LOG HI91 ONMARK ONCE` 指令手动触发一次数据输出, 和 SIN 脉冲效果相同

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。

### 3.2.6 FRESET

恢复出厂设置, 恢复所有可配置参数至出厂默认状态.

## 4. RS-232/TTL/USB 数据协议(二进制)

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)

### 4.1 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    | 5A<br>A5 | 2      | 帧头  |
| 数据域长度 | 1-512    | 2      | 帧中数据域的长度，LSB(低字节在前)<br>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。<br>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部分。<br>标签决定了数据的类型及长度。 |

### 4.2 出厂默认输出

出厂默认输出: 浮点型IMU数据帧(HI91)

### 4.3 数据域内容

#### 4.3.1 浮点型IMU数据帧(HI91)

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 HI91 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:0x91  |
| 1    | status       | uint16_t | 2        | -          | -    | 状态字,见status状态字说明                                    |
| 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    | 当GPS时间未同步成功时,该数据为本地时间戳:即从系统开机开始累加,每毫秒增加1.           |
|      |              |          |          |            |      | 当GPS时间同步成功时,该数据为UTC时间.                              |
| 12   | acc_b        | float    | 4*3      | G          | 1    | 经过出厂校准后的加速度,顺序为:XYZ轴.<br>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   |

| 字节偏移 | 名称    | 数据类型  | 大小<br>(Byte) | 单位  | 比例因子 | 说明              |
|------|-------|-------|--------------|-----|------|-----------------|
| 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 |

#### 4.3.2 整型IMU数据帧(HI92)

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<br>(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,见status状态字说明                |
| 3    | temperature  | int8_t    | 1            | °C      | 1         | 系统平均温度                          |
| 4    | reserved     | uint16_t  | 2            | -       | -         | 保留                              |
| 6    | air_pressure | int16_t   | 2            | Pa      | 1         | 大气压+100000Pa : 如2000表示 102000Pa |
| 8    | heave        | int16_t   | 2            | m       | 0.01      | 船舶升沉(垂向位移), 船舶垂向升沉位移(仅海洋测量类产品)  |
| 10   | gyr_b        | int16_t*3 | 6            | rad/s   | 0.001     | IMU角速度: X,Y,Z轴(出厂校准后)           |
| 16   | acc_b        | int16_t*3 | 6            | m/s^(2) | 0.0048828 | IMU加速度: 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                 |

#### 4.3.3 STATUS 状态字说明

| 位   | 名称            | 描述  |
|-----|---------------|---|
| 0-4 |               | 保留  |
|     |               | 量程报警:   |
| 5   | RANGE_WARN    | 0: 加速度计或陀螺未超过最大量程,<br>1: 加速度计或陀螺超过最大量程, 姿态结果误差极大, 此位一旦置1,只有重启才能清除 |
| 6   | ATT_WARN      | 保留  |
| 7   | BIAS_WARN     | 保留  |
|     |               | 地磁环境指示:   |
| 8   | MAG_DIST_STAT | 0:当前地磁环境良好,或系统处于6轴模式<br>1: 当前地磁收到干扰,航向不可靠                         |
|     |               | 地磁融合标志:   |
| 9   | MAG_AIDING    | 0: 磁传感器不参与航向计算(6轴模式)<br>1:磁传感器正在参与融合航向(9轴模式)                      |
|     |               | 位置误差指示  |
| 10  | POS_WARN      | 0: 位置精度正常,或产品不支持位置输出<br>1: 位置精度误差大                                |

| 位     | 名称              | 描述   |
|-------|-----------------|--|
| 12    | SOUT_PULSE_FLAG | SOUT 脉冲输出标志, 用于和低速传感器(摄像头,Lidar)等时间同步<br>0: 当前数据帧没有SOUT脉冲输出<br>1: 当前数据帧帧存在SOUT脉冲输出 |
| 12-15 |                 | 保留   |

STATUS状态字只支持产品固件版本 >= 1.6.9

## 4.4 CRC

16-bit CRC实现例程:

```
/*
    currectCrc: previous crc value, set 0 if it's first section
    src: source stream data
    lengthInBytes: length

*/
static void crc16_update(uint16_t *currectCrc, const uint8_t *src, uint32_t lengthInBytes)
{
    uint32_t crc = *currectCrc;
    uint32_t j;
    for (j=0; j < lengthInBytes; ++j)
    {
        uint32_t i;
        uint32_t byte = src[j];
        crc ^= byte << 8;
        for (i = 0; i < 8; ++i)
        {
            uint32_t temp = crc << 1;
            if (crc & 0x8000)
            {
                temp ^= 0x1021;
            }
            crc = temp;
        }
    }
    *currectCrc = crc;
}
```

## 4.5 数据帧结构示例(以HI91为例)

使用串口助手采样一帧数据(HI91数据帧),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 14 BB 91 08 15 23 09 A2 C4 47 08 15 1C 00 CC E8 61 BE 9A 35 56 3E 65 EA 72 3F 31 D0 7C BD  
75 DD C5 BB 6B D7 24 BC 89 88 FC 40 01 00 6A 41 AB 2A 70 C2 96 D4 50 41 ED 03 43 41 41 F4 F4 C2 CC CA  
F8 BE 73 6A 19 BE F0 00 1C 3D 8D 37 5C 3F
```

| 字段名称         | 类型       | 原始值         | 解析值                  | 描述                         |
|--------------|----------|-------------|----------------------|----------------------------|
| 帧头           | /        | 5A A5       | -                    | 帧头                         |
| 数据域长度        | /        | 4C 00       | 76                   | 数据域长度=76字节                 |
| CRC          | /        | 14 BB       | BB14                 | CRC校验值                     |
| tag          | /        | 91          | 91                   | 0x91数据包(从次字段开始为payload数据域) |
| 状态字,保留       | uint16_t | 08 15       | 5384                 | 保留                         |
| temperature  | int8_t   | 23          | 35                   | 温度: °C                     |
| air_pressure | float    | 09 A2 C4 47 | 100676               | 气压,Pa                      |
| system_time  | uint32_t | 08 15 1C 00 | 0x001C1508 = 1840392 | 时间戳, ms                    |
| acc_b_x      | float    | CC E8 61 BE | -0.220615            | 加速度X轴,G                    |
| acc_b_y      | float    | 9A 35 56 3E | 0.209189             | 加速度Y轴,G                    |
| acc_b_z      | float    | 65 EA 72 3F | 0.948889             | 加速度Z轴,G                    |
| gyr_b_x      | float    | 31 D0 7C BD | -0.061722            | 角速度X轴, dps                 |
| gyr_b_y      | float    | 75 DD C5 BB | -0.00603836          | 角速度Y轴, dps                 |
| gyr_b_z      | float    | 6B D7 24 BC | -0.0100611           | 角速度Z轴, dps                 |
| mag_b_x      | float    | 89 88 FC 40 | 7.89167              | 磁场X轴, uT                   |
| mag_b_y      | float    | 01 00 6A 41 | 14.625               | 磁场Y轴, uT                   |
| mag_b_z      | float    | AB 2A 70 C2 | -60.0417             | 磁场Z轴, uT                   |
| roll         | float    | 96 D4 50 41 | 13.0519              | 横滚角, deg                   |
| pitch        | float    | ED 03 43 41 | 12.1885              | 俯仰角, deg                   |
| yaw          | float    | 41 F4 F4 C2 | -122.477             | 航向角, deg                   |
| q_w          | float    | CC CA F8 BE | -0.485922            | 四元数W                       |
| q_x          | float    | 73 6A 19 BE | -0.14982             | 四元数X                       |
| q_y          | float    | F0 00 1C 3D | 0.0380868            | 四元数Y                       |
| q_z          | float    | 8D 37 5C 3F | 0.860223             | 四元数Z                       |

## 4.6 C语言解析代码示例(以HI91为例)

### 1.校验CRC

```

    uint16_t payload_len;
    uint16_t crc;
    crc = 0;
    payload_len = buf[2] + (buf[3] << 8);

    /* calculate 5A A5 and LEN filed crc */
    crc16_update(&crc, buf, 4);

    /* calculate payload crc */
    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。

## 2. 定义数据接受结构

从 0x91 开始为数据包的数据域, 定义数据结构体和常用转换宏:

```

#include "stdio.h"
#include "string.h"

/* common type conversion */

#define U1(p) (*((uint8_t *) (p)))
#define I1(p) (*((int8_t *) (p)))
#define I2(p) (*((int16_t *) (p)))

static uint16_t U2(uint8_t *p) {uint16_t u; memcpy(&u, p, 2); return u;}
static uint32_t U4(uint8_t *p) {uint32_t u; memcpy(&u, p, 4); return u;}
static int32_t I4(uint8_t *p) {int32_t u; memcpy(&u, p, 4); return u;}
static float R4(uint8_t *p) {float r; memcpy(&r, p, 4); return r;}

typedef struct
{
    uint8_t tag;           /* item tag: 0x91      */
    float acc[3];          /* acceleration      */
    float gyr[3];          /* angular velocity */
    float mag[3];          /* magnetic field   */
    float eul[3];          /* attitude: eular angle */
    float quat[4];         /* attitude: quaternion */
    float pressure;        /* air pressure     */
    uint32_t timestamp;
} imu_data_t;

```

## 3. 接收数据, 从buf[6]=0x91开始为payload部分:

```

imu_data_t i0x91 = {0};
int offset = 6; /* payload strat at buf[6] */
i0x91.tag = U1(buf+offset+0);

```

```

i0x91.pressure = R4(buf+offset+4);
i0x91.timestamp = U4(buf+offset+8);
i0x91.acc[0] = R4(buf+offset+12);
i0x91.acc[1] = R4(buf+offset+16);
i0x91.acc[2] = R4(buf+offset+20);
i0x91.gyr[0] = R4(buf+offset+24);
i0x91.gyr[1] = R4(buf+offset+28);
i0x91.gyr[2] = R4(buf+offset+32);
i0x91.mag[0] = R4(buf+offset+36);
i0x91.mag[1] = R4(buf+offset+40);
i0x91.mag[2] = R4(buf+offset+44);
i0x91.eul[0] = R4(buf+offset+48);
i0x91.eul[1] = R4(buf+offset+52);
i0x91.eul[2] = R4(buf+offset+56);
i0x91.quat[0] = R4(buf+offset+60);
i0x91.quat[1] = R4(buf+offset+64);
i0x91.quat[2] = R4(buf+offset+68);
i0x91.quat[3] = R4(buf+offset+72);

```

#### 4. 打印接收到的数据:

```

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

```

## 4.7 最大传输速率

| 协议 | 字节数 | 9600bps | 115200bps | 230400bps | 256000bps | 460800bps | 921600bps |
|----|-----|---------|-----------|-----------|-----------|-----------|-----------|
| 91 | 76  | 10Hz    | 100Hz     | 250Hz     | 250Hz     | 500Hz     | 1000Hz    |
| 92 | 48  | 10Hz    | 200Hz     | 250Hz     | 250Hz     | 500Hz     | 1000Hz    |

## 5. RS-485输出协议(Modbus)

- 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)

### 5.1 数据帧格式

#### 5.1.1 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                                       |

### 5.1.2 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     |

### 5.1.3 CRC校验

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

- C代码:

```
static const uint16_t modbus_crc_table[256] = {  
    0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,  
    0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,  
    0xcc01, 0x0cc0, 0xd80, 0xcd41, 0xf00, 0xfc1, 0xe81, 0xe40,  
    0xa00, 0xcac1, 0xcb81, 0xb40, 0xc901, 0x9c0, 0x880, 0xc841,  
    0xd801, 0x18c0, 0x1980, 0xd941, 0xb00, 0dbc1, 0xda81, 0xa40,  
    0x1e00, 0xdec1, 0xdf81, 0xf40, 0xdd01, 0xdc0, 0xc80, 0xdc41,  
    0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,  
    0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,  
    0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,  
    0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,  
    0x3c00, 0xfc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,  
    0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,  
    0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,  
    0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,  
    0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,  
};
```

```

0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
0x6c00, 0xac1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbd1, 0xbc81, 0x7c40,
0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,
0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040
};

uint16_t modbus_crc_calc(uint8_t *buf, uint16_t len)
{
    uint16_t crc = 0xFFFFU;
    uint8_t nTemp;

    while (len--)
    {
        nTemp = *buf++ ^ crc;
        crc >>= 8;
        crc ^= modbus_crc_table[(nTemp & 0xFFU)];
    }

    return(crc);
}

```

## 5.2 寄存器列表

| 地址<br>(Hex) | 地址<br>(Dec) | 名称           | 类<br>型 | 功<br>能       | R/W | 说<br>明   |
|-------------|-------------|--------------|--------|--------------|-----|--|
| 0x00        | 0           | CTRL         | u16    | 控制           | W   | 参见Modbus 设置模块章节                                |
| 0x04        | 4           | UART1_BAUD   | u16    | 波特率          | R/W | 串口波特率  |
| 0x05        | 5           | MD_ID        | u16    | Modbus<br>ID | R/W | Modbus ID: 有效范围: 1-128                         |
| 0x06        | 6           | HEADING_MODE | u16    | 航向角模<br>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<br>磁融合, 绝对航向) |
| 0x34        | 52          | ACCX         | i16    | 加速度X         | R   | 单位G(1G=1重力加速度), 比例因子:0.00048828                |

| 地址<br>(Hex) | 地址<br>(Dec) | 名称      | 类<br>型 | 功能          | R/W | 说明  |
|-------------|-------------|---------|--------|-------------|-----|---|
| 0x35        | 53          | ACCY    | i16    | 加速度Y        | R   | 单位G(1G=1重力加速度), 比例因子:0.00048828   |
| 0x36        | 54          | ACCZ    | i16    | 加速度Z        | R   | 单位G(1G=1重力加速度), 比例因子:0.00048828   |
| 0x37        | 55          | GYRX    | i16    | 角速度X        | R   | 单位deg/s, 比例因子:0.061035  |
| 0x38        | 56          | GYRY    | i16    | 角速度Y        | R   | 单位deg/s, 比例因子:0.061035  |
| 0x39        | 57          | GYRZ    | i16    | 角速度Z        | R   | 单位deg/s, 比例因子:0.061035  |
| 0x3A        | 58          | MAGX    | i16    | 磁强度X        | R   | 单位uT, 比例因子: 0.030517  |
| 0x3B        | 59          | MAGY    | i16    | 磁强度Y        | R   | 单位uT, 比例因子: 0.030517  |
| 0x3C        | 60          | MAGZ    | i16    | 磁强度Z        | R   | 单位uT, 比例因子: 0.030517  |
| 0x3D        | 61          | R_H     | i32    | 横滚角高<br>16位 | R   | 单位deg, 比例因子:0.001   |
| 0x3E        | 62          | R_L     | -      | 横滚角低<br>16位 | R   | 单位deg, 比例因子:0.001   |
| 0x3F        | 63          | P_H     | i32    | 俯仰角高<br>16位 | R   | 单位deg, 比例因子:0.001   |
| 0x40        | 64          | P_L     | -      | 俯仰角低<br>16位 | R   | 单位deg, 比例因子:0.001   |
| 0x41        | 65          | Y_H     | i32    | 航向角高<br>16位 | R   | 单位deg, 比例因子:0.001   |
| 0x42        | 66          | Y_L     | -      | 航向角低<br>16位 | R   | 单位deg, 比例因子:0.001   |
| 0x43        | 67          | TEMP    | i16    | 温度          | R   | 单位°C, 比例因子:0.01   |
| 0x44        | 68          | PRS_H   | i32    | 气压高16<br>位  | R   | 单位Pa, 比例因子:0.01   |
| 0x45        | 69          | PRS_L   | -      | 气压低16<br>位  | R   | 单位Pa, 比例因子:0.01   |
| 0x46        | 70          | Q0      | u16    | 四元数<br>QW   | R   | 四元数, 比例因子: 0.0001   |
| 0x47        | 71          | Q1      | u16    | 四元数<br>QX   | R   | 四元数, 比例因子: 0.0001   |
| 0x48        | 72          | Q2      | u16    | 四元数<br>QY   | R   | 四元数, 比例因子: 0.0001   |
| 0x49        | 73          | Q3      | u16    | 四元数<br>QZ   | R   | 四元数, 比例因子: 0.0001   |
| 0x4A        | 74          | INCLI_X | i16    | 倾角仪X<br>轴角度 | R   | 双轴倾角仪产品:X角度, ±180, 单位deg, 比例因子:<br>0.011<br>单轴倾角仪产品:X角度, 0-360, 单位deg, 比例因子:<br>0.011 |
| 0x4B        | 75          | INCLI_Y | i16    | 倾角仪Y<br>轴角度 | R   | 双轴倾角仪Y角度: ±90, 单位deg, 比例因子: 0.011<br>单轴倾角仪: 此寄存器保留                                    |
| 0x4E        | 78          | HEVAE   | i16    | 船舶垂荡        | R   | 船舶垂荡位移输出, 单位 m, 比例因子 0.01   |

| 地址<br>(Hex)   | 地址<br>(Dec) | 名称           | 类<br>型 | 功能      | R/W | 说明   |
|---------------|-------------|--------------|--------|---------|-----|--|
| 0x51          | 81          | HEAVE_PERIOD | i16    | 船舶垂荡周期  | R   | 船舶垂荡位移周期, 单位s, 比例因子 0.001  |
| 0x70-<br>0x77 | 112-<br>119 | PNAME        | u16    | 设备名     | R   | 设备名字符串, ASCII码, 共占8个寄存器  |
| 0x78          | 120         | SW_VERSION   | u16    | 软件版本    | R   | 软件版本   |
| 0x79          | 121         | BL_VERSION   | u16    | BL版本    | R   | BL版本   |
| 0x7F-<br>0x82 | 127-<br>130 | SN           | u16    | 产品唯一序列号 | R   | 产品唯一序列号, 占4寄存器   |
| 0xA5          | 165         | SET_LV       | u16    | 自动校平    | W   | <p>3: 执行一次自动调平: 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. 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°,180° 适用于机器人安装环境。其中, “接近”定义为Pitch Roll均小于15°</p> <p>5: 取消自动调平, 恢复绝对测量角度</p> <p>其他值: 无效</p> |
| 0xA6          | 166         | URFR         | u16    | 安装设置    | W   | <p>0: 设置为水平安装(默认模式)</p> <p>1: 垂直安装:Y轴正方向朝下</p> <p>2: 垂直安装:Y轴正方向朝上</p> <p>3: 垂直安装:X轴正方向朝上</p> <p>4: 垂直安装:X轴正方向朝下</p>  |

## 5.3 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。

### 5.3.1 控制寄存器说明(0x00)

| 命令             | CTL寄存器写入值 | 命令(Hex) ID=0X50(出厂默认)   |
|----------------|-----------|-------------------------|
| 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             | 0x0OFF    | 50 06 00 00 FF C4 0B    |

### 5.3.2 配置波特率(0x04)

| 配置目标波特率 | 指令(Hex) ID=0X50(出厂默认)   |
|---------|-------------------------|
| 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 |

### 5.3.3 配置节点ID(0x05)

[CUR\_ID] 06 00 05 00 [NEW\_ID] CRC(2字节)

- CUR\_ID: 当前设备Modbus ID.
- NEW\_ID: 新(要设置的)ID号

示例(当前ID=0x50):

- 设置NEW\_ID为0x50: 50 06 00 05 00 50 94 76
- 设置NEW\_ID为0x51: 50 06 00 05 00 51 55 B6
- 设置NEW\_ID为0x52: 50 06 00 05 00 52 15 B7
- 设置NEW\_ID为0x53: 50 06 00 05 00 53 D4 77

注意: 修改成功后Modbus地址会立即生效, 需要修改Modbus的发送CUR\_ID字段, 如果没有Modbus协议基础, 建议使用上位机操作。

### 5.3.4 设置安装方式 (0xA6)

| 配置目标安装方式         | 指令(Hex) ID=0X50(出厂默认)   |
|------------------|-------------------------|
| 0: 设置为水平安装(默认模式) | 50 06 00 A6 00 00 64 68 |
| 1: 垂直安装:Y轴正方向朝下  | 50 06 00 A6 00 01 A5 A8 |
| 2: 垂直安装:Y轴正方向朝上  | 50 06 00 A6 00 02 E5 A9 |
| 3: 垂直安装:X轴正方向朝上  | 50 06 00 A6 00 03 24 69 |
| 4: 垂直安装:X轴正方向朝下  | 50 06 00 A6 00 04 65 AB |

### 5.3.5 设置水平校准(0xA5)

- 启动自动水平校平: 50 06 00 A5 00 02 15 A9
- 取消自动水平校平: 50 06 00 A5 00 05 54 6B

### 5.3.6 设置6轴或9轴模式(0x06)

- 设置为6轴模式: 50 06 00 06 00 00 64 4A
- 设置为9轴模式: 50 06 00 06 00 01 A5 8A

## 5.4 读取模块版本信息(0x70-0x82)

读取模块产品名, 软件版本及SN号:

请求帧 50 03 00 70 00 14 49 9F

| 字段    | 值      | 说明       |
|-------|--------|----------|
| 设备地址  | 0x50   | 模块地址     |
| 功能码   | 0x03   | 读保持寄存器   |
| 起始地址  | 0x0070 | 产品信息起始地址 |
| 读取长度  | 0x0014 | 读取20个寄存器 |
| CRC校验 | 0x9F49 | -        |

响应帧: 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 00 00 04 7D 95 5F 8D 2A 17 08 00 00 4D 0C

| 字段   | 数据               | 说明       |
|------|------------------|----------|
| 产品名  | 48 49...30 30    | CH10x(M) |
| 软件版本 | 0x98             | V1.52    |
| 引导版本 | 0x6B             | V1.07    |
| 序列号  | 047D955F8D2A1708 | SN码      |

## 5.5 读取传感器数据(0x34-0x4B)

请求帧 50 03 00 34 00 18 09 8F

| 字段   | 值      | 说明        |
|------|--------|-----------|
| 设备地址 | 0x50   | 模块地址      |
| 功能码  | 0x03   | 读保持寄存器    |
| 起始地址 | 0x0034 | 传感器数据起始地址 |

| 字段    | 值      | 说明       |
|-------|--------|----------|
| 读取长度  | 0x0018 | 读取24个寄存器 |
| CRC校验 | 0x8F09 | -        |

响应帧: 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

加速度(单位:G, 可取9.8m/s^(2)):

| 轴向 | 寄存器值(HEX) | 原始值(DEC) | 比例因子       | 计算得出物理值 |
|----|-----------|----------|------------|---------|
| X  | FF 01     | -255     | 0.00048828 | -0.1245 |
| Y  | 03 B0     | 944      | 0.00048828 | 0.4609  |
| Z  | 06 50     | 1616     | 0.00048828 | 0.7891  |

角速度(单位: deg/s)

| 轴向 | 寄存器值(HEX) | 原始值(DEC) | 比例因子     | 计算得出物理值  |
|----|-----------|----------|----------|----------|
| X  | FC C9     | -823     | 0.061035 | -50.2318 |
| Y  | FF 7C     | -132     | 0.061035 | -8.0566  |
| Z  | 00 91     | 145      | 0.061035 | 8.8501   |

磁场(单位: uT)

| 轴向 | 寄存器值(HEX) | 原始值(DEC) | 比例因子     | 计算得出物理值  |
|----|-----------|----------|----------|----------|
| X  | 01 D5     | 469      | 0.030517 | 14.3125  |
| Y  | FD DB     | -549     | 0.030517 | -16.7538 |
| Z  | FD 27     | -729     | 0.030517 | -22.2469 |

欧拉角(单位: deg)

| 轴向         | 寄存器值(HEX)   | 原始值(DEC) | 比例因子  | 计算得出物理值  |
|------------|-------------|----------|-------|----------|
| 横滚(Roll)   | 00 00 21 FF | 8703     | 0.001 | 8.703    |
| 俯仰角(Pitch) | 00 00 7F F6 | 32758    | 0.001 | 32.758   |
| 航向角(Yaw)   | FF FD 73 E7 | -166937  | 0.001 | -166.937 |

## 6. CAN数据协议(CANopen)

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。

### 6.1 CANopen 默认设置

| 默认配置     | 值                   |
|----------|---------------------|
| CAN 波特率  | 500 kbit/s          |
| 节点ID     | 8                   |
| 初始化状态    | Operational         |
| TPDO输出速率 | 1Hz - 200Hz(每个TPDO) |

### 6.2 CANopen TPDO

| 通道    | 帧ID      | 数据长度<br>(DLC) | 传输方式           | 输出频率<br>(Hz) | 数据    | 说明   |
|-------|----------|---------------|----------------|--------------|-------|--|
| TPDO1 | 0x180+ID | 6             | 异步定时<br>(0xFE) | 100          | 加速度   | 类型:int16 低字节在前，每个轴2字节，共6字节<br>分别为X,Y,Z轴加速度，单位为mG(0.001G)   |
| TPDO2 | 0x280+ID | 6             | 异步定时<br>(0xFE) | 100          | 角速度   | 类型:int16 低字节在前，每个轴2字节，共6字节<br>分别为X,Y,Z轴角速度，单位为0.1dps(°/s)  |
| TPDO3 | 0x380+ID | 6             | 异步定时<br>(0xFE) | 100          | 欧拉角   | 类型:int16 低字节在前，每个轴2字节，共6字节<br>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<br>(0xFE) | 100          | 四元数   | 类型:int16 低字节在前，每个元素2字节，共8字节<br>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<br>(0xFE) | 20           | 气压    | 类型:int32 共4字节。单位Pa   |
| TPDO7 | 0x780+ID | 8             | 异步定时<br>(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<br>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 = 731 = 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

## 6.3 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:

| 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   |

## 6.4 配置指令(SDO协议)

所有配置指令均采用快速SDO配置。所有配置更改后需要发送保存配置指令才能保存到Flash

### 6.4.1 SDO(Service Data Object)协议

快速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     | 保留     |

#### 6.4.1.1 修改节点ID (0x20A0)

ID=0x608, DATA=23,A0,20,00,[ID],00,00,00, ID修改范围: 1-127, 修改后需要保存配置到Flash且复位(或重新上电生效)。

#### 6.4.1.2 保存配置到Flash(0x2000)

ID=0x608, DATA=23,00,20,00,00,00,00,00

#### 6.4.1.3 复位(0x2000)

ID=0x608, DATA=23,00,20,00,FF,00,00,00

#### 6.4.1.4 恢复出厂设置(0x2000)

ID=0x608, DATA=23 00 20 00 01 00 00 00 恢复出厂设置，将所有参数包括波特率，节点ID等恢复至出厂值，重新上电生效，谨慎使用。

#### 6.4.1.5 修改CAN波特率(0x209A)

ID=0x608, DATA=23,9A,20,00,[ID]，修改后需要保存配置到Flash且复位(或重新上电)生效。

- CAN波特率修改为1000 kbit/s: ID=0x608, DATA=23,9A,20,00,00,00,00,00
- CAN波特率修改为500kbit/s: ID=0x608, DATA=23,9A,20,00,02,00,00,00
- CAN波特率修改为250kbit/s: ID=0x608, DATA=23,9A,20,00,03,00,00,00
- CAN波特率修改为125kbit/s: ID=0x608, DATA=23,9A,20,00,04,00,00,00

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 |

#### 6.4.1.6 修改/关闭/开启数据输出速率(0x1800-0x1805)

此项配置立即生效

- ID=0x608, DATA=2B,00,18,05,00,00,00,00 关闭加速度输出(1800.5=0)
- ID=0x608, DATA=2B,00,18,05,05,00,00,00 加速度200Hz输出(1800.5=5)
- ID=0x608, DATA=2B,00,18,05,0A,00,00,00 加速度100Hz输出(1800.5=10)
- ID=0x608, DATA=2B,00,18,05,14,00,00,00 加速度50Hz输出(1800.5=20)
- ID=0x608, DATA=2B,00,18,05,32,00,00,00 加速度20Hz输出(1800.5=50)
- ID=0x608, DATA=2B,00,18,05,64,00,00,00 加速度10Hz输出(1800.5=100)
- ID=0x608, DATA=2B,01,18,05,00,00,00,00 关闭角速度输出(1801.5=0)
- ID=0x608, DATA=2B,01,18,05,05,00,00,00 角速度200Hz输出(1801.5=5)
- ID=0x608, DATA=2B,01,18,05,0A,00,00,00 角速度100Hz输出(1801.5=10)
- ID=0x608, DATA=2B,01,18,05,14,00,00,00 角速度50Hz输出(1801.5=20)
- ID=0x608, DATA=2B,01,18,05,32,00,00,00 角速度20Hz输出(1801.5=50)
- ID=0x608, DATA=2B,01,18,05,64,00,00,00 角速度10Hz输出(1801.5=100)
- ID=0x608, DATA=2B,02,18,05,00,00,00,00 关闭欧拉角输出(1802.5=0)
- ID=0x608, DATA=2B,02,18,05,05,00,00,00 欧拉角200Hz输出(1802.5=5)

- ID=0x608, DATA=2B,02,18,05,0A,00,00,00 欧拉角100Hz输出(1802.5=10)
- ID=0x608, DATA=2B,02,18,05,14,00,00,00 欧拉角50Hz输出(1802.5=20)
- ID=0x608, DATA=2B,02,18,05,32,00,00,00 欧拉角20Hz输出(1802.5=50)
- ID=0x608, DATA=2B,02,18,05,64,00,00,00 欧拉角10Hz输出(1802.5=100)
- ID=0x608, DATA=2B,03,18,05,00,00,00,00 关闭四元数输出(1803.5=0)
- ID=0x608, DATA=2B,03,18,05,05,00,00,00 四元数200Hz输出(1803.5=5)
- ID=0x608, DATA=2B,03,18,05,0A,00,00,00 四元数100Hz输出(1803.5=10)
- ID=0x608, DATA=2B,03,18,05,14,00,00,00 四元数50Hz输出(1803.5=20)
- ID=0x608, DATA=2B,03,18,05,32,00,00,00 四元数20Hz输出(1803.5=50)
- ID=0x608, DATA=2B,03,18,05,64,00,00,00 四元数10Hz输出(1803.5=100)
- ID=0x608, DATA=2B,04,18,05,00,00,00,00 关闭气压输出(1804.5=0)
- ID=0x608, DATA=2B,04,18,05,05,00,00,00 气压200Hz输出(1804.5=5)
- ID=0x608, DATA=2B,04,18,05,0A,00,00,00 气压100Hz输出(1804.5=10)
- ID=0x608, DATA=2B,04,18,05,14,00,00,00 气压50Hz输出(1804.5=20)
- ID=0x608, DATA=2B,04,18,05,32,00,00,00 气压20Hz输出(1804.5=50)
- ID=0x608, DATA=2B,04,18,05,64,00,00,00 气压10Hz输出(1804.5=100)

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.

#### 6.4.1.7 设置倾角仪输出正负号(0x209E)

- 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

#### 6.4.1.8 设置倾角仪零点(0x20A5)

- 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)

### 6.4.2 同步协议

遵循CANopen协议, 模块可以将各个TPDO设置为同步模式, 即停止异步定时发送, 而等待CANopen同步帧, 当同步帧到来时, 发送一帧TPDO数据。

#### 6.4.2.1 配置TPDO为同步模式

将想设置为同步模式的TPDO配置为同步模式, 通过TPDO通讯参数字典[0x180x.2] (Transmission type)设置为0x01同步模式即可。具体意义请查看CANopen协议。以TPDO1(加速度信息)为例:

ID=0x608, DATA=2F,00,18,02,01,00,00,00 写入[0x1800.2(US8类型)]=1, 设置TPDO1 通讯模式为同步模式

ID=0x608, DATA=2F,00,18,02,FF,00,00,00 写入[0x1800.2(US8类型)]=0xFF, 设置TPDO1 通讯模式为异步通讯模式(出厂默认)

#### 6.4.2.2 发送CANopen同步帧

发送CANopen同步帧, CANopen 同步帧: ID:80, DATA:空, 模块接收到同步帧后, 所有配置为同步模式的 PDO会发送一帧数据, 实现同步.

#### 6.4.2.3 设置心跳包

心跳协议通过写入[0x1017.0(US16)]设置,有效范围0-65535,单位ms. 0表示禁用心跳包.

ID=0x608, DATA=2B,17,10,00,64,00,00,00 设置心跳包周期为100ms.

## 7. CAN数据协议(J1939)

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    |

### 7.1 PGN消息列表

#### 7.1.1 PGN65327(FF2F) 时间信息

CANID=0xCFF2F08

| SPN 名称 | SPN 位置(byte) | 说明                                    |
|--------|--------------|---------------------------------------|
| UTC年   | 0            | 0-99, 20代表2020,以此类推, 如无法获得UTC时间则此位为20 |
| UTC月   | 1            | 0-12, 如无法获得UTC月份则此位为0                 |
| UTC日   | 2            | 0-31, 如无法获得UTC日期则此位为0                 |
| UTC时   | 3            | 0-23                                  |
| UTC分   | 4            | 0-59                                  |
| UTC秒   | 5            | 0-59                                  |
| UTC毫秒  | 6-7          | 0-999, 单位:ms, 比例因子:1                  |

#### 7.1.2 PGN65332(FF34) 加速度

CANID=0xCFF3408

| 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      | -                               |

#### 7.1.3 PGN65335(FF37) 角速度

CANID=0xCFF3708

| 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 |

| 名称 | 位置(byte) | 说明 |
|----|----------|----|
| 保留 | 6-7      |    |

#### 7.1.4 PGN65341(FF3D) 倾角仪输出

CANID=0x0CFF3D08

| SPN 名称     | SPN 位置(byte) | 说明              |
|------------|--------------|-----------------|
| 横滚角(Roll)  | 0-3          | 单位°, 比例因子:0.001 |
| 俯仰角(Pitch) | 4-7          | 单位°,比例因子:0.001  |

#### 7.1.5 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          |                              |

#### 7.1.6 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 ±90,单位:deg, 比例因子:0.001  |

## 7.2 配置指令

### 7.2.1 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        | 写入:写入的值, 读取:保留   |

### 7.2.2 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 | 3D 01 06 00 [VAL] | VAL: 4字节 | PGN:FF3D(俯仰横滚) 发送间隔, 单位ms, 范围:5-1000 |

| 29'b 拓展帧地址 | 数据                         | 描述           | 说明  |
|------------|----------------------------|--------------|---|
| 0x0CEF08xx | 41 01 06 00 [VAL]<br>节     | VAL: 4字<br>节 | PGN:FF41(航向角) 发送间隔, 单位ms, 范围:5 -1000                          |
| 0x0CEF08xx | 4A 01 60 00 [VAL]<br>节     | VAL: 4字<br>节 | PGN:FF4A(倾角仪输出) 发送间隔, 单位ms, 范围:5 -1000                        |
| 0x0CEF08xx | 9D 00 06 00 01 00<br>00 00 | -            | 全局使能节点数据输出  |
| 0x0CEF08xx | 9D 00 06 00 00 00<br>00 00 | -            | 全局关闭节点数据输出(默认)  |
| 0x0CEF08xx | 00 00 06 00 00 00<br>00 00 | -            | 保存所有配置参数到Flash  |
| 0x0CEF08xx | 00 00 06 00 01 00<br>00 00 | -            | 恢复出厂设置  |
| 0x0CEF08xx | 00 00 06 00 FF 00<br>00 00 | -            | 复位  |
| 0x0CEF08xx | 9A 00 06 00 [VAL]<br>节     | VAL: 4字<br>节 | 配置波特率(保存设置, 复位生效): 0:1000K, 1:800K, 2:500K,<br>3:250K, 4:125K |
| 0x0CEF08xx | 9C 00 06 00 [VAL]<br>节     | VAL: 4字<br>节 | 设置J1939 节点ID: 1-128   |
| 0x0CEF08xx | A5 00 06 00 [VAL]<br>节     | VAL: 4字<br>节 | 设置倾角仪零位, 0x02:设置当前位置为零位, 0x05: 取消零位设<br>置, 输出绝对物理角度           |
| 0x0CEF08xx | 9E 00 06 00 [VAL]<br>节     | VAL: 4字<br>节 | 设置倾角仪X轴正负方向, 0:默认 1:反向  |
| 0x0CEF08xx | 9F 00 06 00 [VAL]<br>节     | VAL: 4字<br>节 | 设置倾角仪Y轴正负方向, 0:默认 1:反向  |

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)

## 8. 地磁校准

### 8.1 地磁校准步骤

9轴模式(磁辅助绝对航向角模式)使用的前提是:

1. 首次使用9轴模式必须**至少进行一次用户地磁校准**
2. 使用过程中没有**空间磁场干扰**(室内等复杂磁环境下很难做到)

只有同时达到这两个要求，9轴模式下航向角才能达到手册标称精度。

#### 8.1.1 用户地磁校准

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。
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(30CM以上)。
3. 在尽量小范围内(**位置尽量不动，只旋转模块**)，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. 如果模块安装位置改变(比如第一次校准是拿着模块单独去校准的，但最后需要安装在您的PCB或者外壳中使用)。则需要带着PCB或者外壳一起重新校准，如果模块是放到机器人/无人机上，则需要带着机器人/无人机一起校准。总之，**必须将模块和被安装体(PCB/外壳/机器人/无人机等)视作一个刚性整体去校准**。
5. 查看地磁校准结果: 输入 LOG MAGCONFIG, 返回结果如下:

```
...
MB=14.14,-2.08,-8.75
OK
```

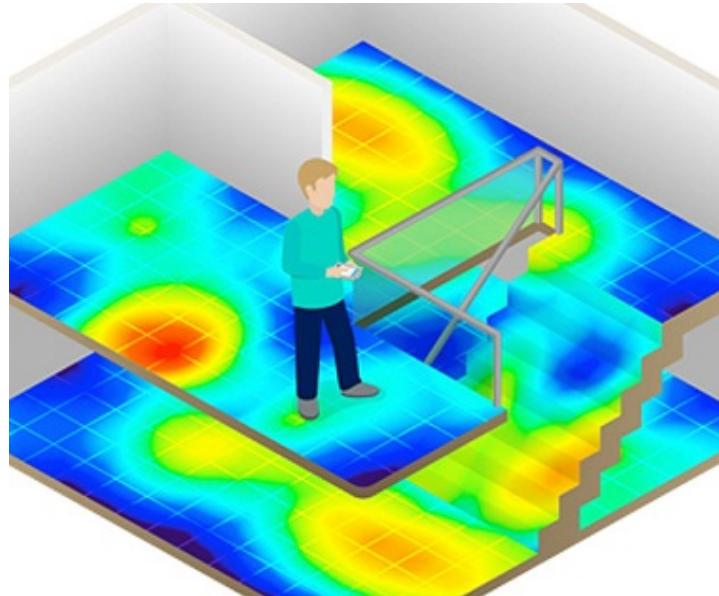
如果MAG\_BIAS显示三个值不是0,0,0。代表校准成功。

#### 8.1.2 空间磁场干扰

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"><li>• Calibration errors</li><li>• Hard iron effects</li><li>• Soft iron effects</li><li>• Etc.</li></ul>                             | <ul style="list-style-type: none"><li>• Spatial distortions</li><li>• Temporal distortions</li><li>• Etc.</li></ul>   |

|         |   |   |
|---------|---|---|
| 磁干扰类型   | 随传感器坐标系的磁场干扰, 也称硬磁干扰/软磁干扰<br>(Distortion that move with the sensor) | 空间磁场干扰(Distortion that do not move with the sensor)   |
| 特点      | 干扰源随传感器运动而运动  | 干扰源不随传感器运动而运动   |
| 典型      | 与模块固定在一起的PCB,   | 家具, 家用电器, 线缆, 房屋内的钢筋结构等, 不能随着传感器移动而移动。  |
| 干扰源     | 金属外壳, 将模块固定后的无人机等等  | 还有人形机器人:本身作为一次磁性刚体磁场分布不停变化(走动)  |
| 是否可能被校准 | 可以  | 不可能   |
| 抗干扰措施   | 可以通过用户磁校准过程消除   | <p>无论如何校准, 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, 理论上就无法获得正确的航向角。空间磁场干扰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, 会严重影响航向角误差。</p> <p>空间磁场干扰在室内尤其是靠近桌椅家电等地方尤其严重。见下图为一个典型的室内外空间磁场干扰: 蓝色为弱干扰区, 红色为强干扰区</p> |



### 8.1.3 关于模块地磁校准算法

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, 校准成功后, 会自动将校准参数保存到Flash。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。**自动校准的前提是首次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。

## 8.2 再次强调

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。

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/三脚架/PCB板等)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。

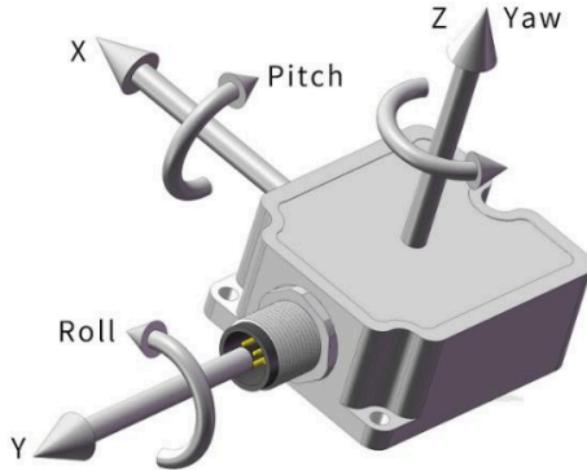
## 9. 附录1 四元数/欧拉角/旋转矩阵互转

### 9.1 四元数转旋转矩阵

给定四元数  $Q_{b2n} = [q_0, q_1, q_2, q_3]^T$ , 方向余弦矩阵为:

$$C_{b2n} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

### 9.2 四元数转欧拉角 - 东北天(ENU)-312(先转Z,然后X轴,最后Y轴)旋转顺序下的欧拉角



给定四元数  $Q_{b2n} = [q_0, q_1, q_2, q_3]^T$ , 其中  $q_0$  为标量部分,  $[q_1, q_2, q_3]$  为矢量部分。 $Q_{b2n}$  代表b系到n系的坐标旋转四元数: 其中:

- $\text{pitch}(\theta)$ : 绕X轴的旋转角度, 范围  $[-\frac{\pi}{2}, \frac{\pi}{2}]$
- $\text{roll}(\varphi)$ : 绕Y轴的旋转角度, 范围  $[-\pi, \pi]$
- $\text{yaw}(\psi)$ : 绕Z轴的旋转角度, 范围  $[-\pi, \pi]$

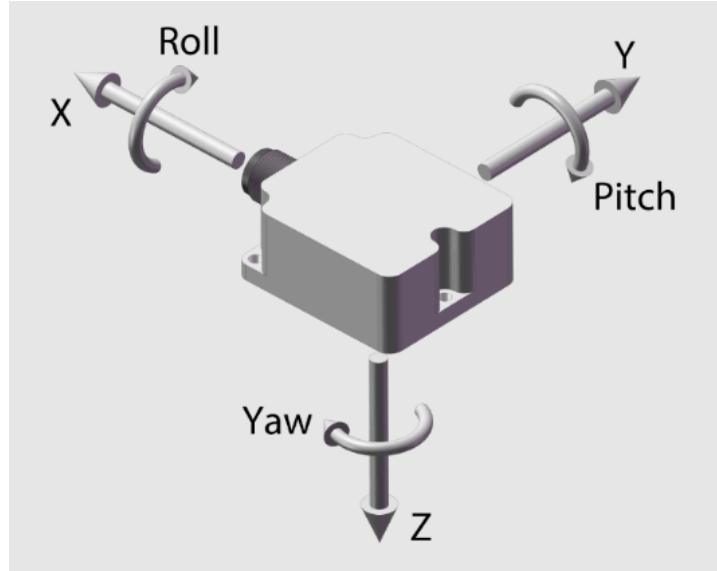
四元数转欧拉角:

$$\begin{aligned} \text{pitch} &= \arcsin(2(q_0q_1 + q_2q_3)) \\ \text{roll} &= -\arctan 2(2(q_1q_3 - q_0q_2), q_0^2 - q_1^2 - q_2^2 + q_3^2) \\ \text{yaw} &= -\arctan 2(2(q_1q_2 - q_0q_3), q_0^2 - q_1^2 + q_2^2 - q_3^2) \end{aligned}$$

欧拉角转四元数:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) - \sin\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) \\ \cos\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) \sin\left(\frac{\text{pitch}}{2}\right) - \cos\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) \\ \cos\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) \sin\left(\frac{\text{roll}}{2}\right) + \cos\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) \\ \cos\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) + \sin\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) \end{bmatrix}$$

### 9.3 四元数转欧拉角 - 北东地(NED)-321(先转Z轴,然后Y轴,最后X轴)旋转顺序下的欧拉角



给定四元数  $Q_{b2n} = [q_0, q_1, q_2, q_3]^T$ , 其中  $q_0$  为标量部分,  $[q_1, q_2, q_3]$  为矢量部分。 $Q_{b2n}$  代表 b 系到 n 系的坐标旋转四元数: 其中:

- pitch( $\theta$ ): 绕Y轴的旋转角度, 范围  $[-\frac{\pi}{2}, \frac{\pi}{2}]$
- roll( $\phi$ ): 绕X轴的旋转角度, 范围  $[-\pi, \pi]$
- yaw( $\psi$ ): 绕Z轴的旋转角度, 范围  $[-\pi, \pi]$

四元数转欧拉角:

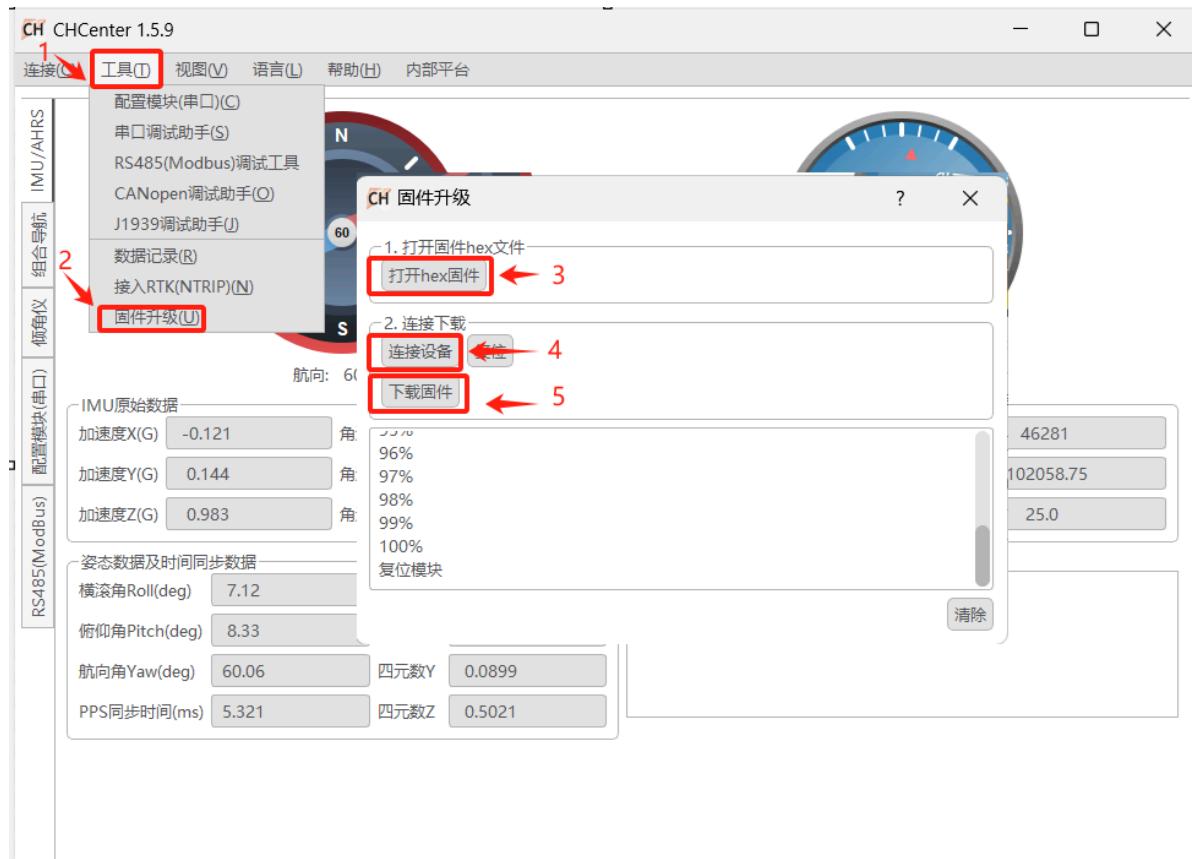
$$\begin{aligned} roll &= \arctan 2(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ pitch &= \arcsin(2(q_0q_2 - q_1q_3)) \\ yaw &= \arctan 2(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{aligned}$$

欧拉角转四元数:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) + \sin\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) \\ \sin\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) - \cos\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) \\ \cos\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) + \sin\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) \\ \cos\left(\frac{\text{roll}}{2}\right) \cos\left(\frac{\text{pitch}}{2}\right) \sin\left(\frac{\text{yaw}}{2}\right) - \sin\left(\frac{\text{roll}}{2}\right) \sin\left(\frac{\text{pitch}}{2}\right) \cos\left(\frac{\text{yaw}}{2}\right) \end{bmatrix}$$

## 10. 附录2 固件升级

本产品支持固件升级，请使用CHCenter上位机软件按下图步骤进行固件升级，固件升级文件(.hex)请向我司技术支持人员索取。



## 11. 附录3 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

微信:



Telegram:

