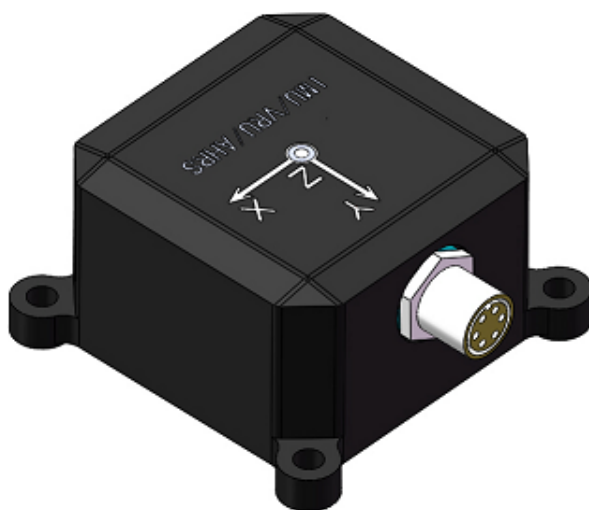


CH10X系列用户手册

IMU/VRU/AHRS姿态测量模块, Rev 1.0

适用于 CH102(M)/CH104(M)/CH108M



文件: CH10X_UM_CN

技术支持: support@hipnuc.com

属性: 公开

网站: www.hipnuc.com

HIPNUC[®]
超核电子

© 2016-2022, 北京超核电子科技有限公司版权所有, 本文档所包含的信息在没有通知的情况下可能会发生更改。

文档变更历史

| 版本 | 日期 | 作者 | 变更内容 |
|------|------------|--------|------|
| V1.0 | 2022年6月18日 | HiPNUC | 初始版本 |

CH10X系列用户手册

1 产品简介

1.1 算法架构

1.2 主要应用场景:

1.3 产品特点

1.3.1 出厂标定

1.3.2 可输出多种数据类型

1.3.3 数据接口

1.3.4 其他

1.4 辅助开发套件

1.4.1 屏蔽双绞线

1.4.2 上位机软件CHCenter

1.5 订购信息

2 机械与电气特性

2.1 CH10X尺寸图

2.2 CH10X系列航插引脚定义

2.3 电气与机械参数

2.4 坐标系定义

3 性能指标

3.1 姿态角输出精度

3.2 陀螺仪

3.3 加速度计

3.4 磁传感器参数

3.5 关键电气参数

4 标定

4.1 出厂标定

4.2 启动标定

4.3 磁力计校准

4.3.1 磁干扰分类

4.3.1.1 空间磁场干扰(干扰不随传感器位姿改变而改变)

4.3.1.2 传感器坐标系下的干扰(干扰随传感器位姿改变而改变)

4.3.2 地磁使用注意事项

4.3.3 案例分析

5 关于模式说明

5.1 型号的模式区别

5.2 不同模式的区别

6 232串口通讯协议

6.1 串口数据包

6.2 产品支持数据包列表

6.2.1 0X91(IMUSOL)

6.3 数据帧结构示例

6.3.1 数据帧配置为 0x91 数据包

7 AT指令

7.1 AT指令列表

7.2 指令详解

7.2.1 AT+ID

7.2.2 AT+INFO

7.2.3 AT+ODR

7.2.4 AT+BAUD

7.2.5 AT+EOUT

7.2.6 AT+RST

7.2.7 AT+SETYAW

7.2.8 AT+MODE

7.2.9 AT+URFR

7.2.10 AT+RSTORT

8 CAN通讯协议

8.1 CANopen协议

8.1.1 CANopen 默认设置

8.1.2 CANopen TPDO

8.1.3 CANopen接口常用命令

8.1.3.1 使能数据输出(开启异步触发)

8.1.3.2 修改CAN波特率

8.1.3.3 修改节点ID

8.1.3.4 修改数据输出速率

8.1.3.5 关闭数据输出

8.1.3.6 开启/关闭站点

8.1.3.7 配置TPDO为同步模式

8.2 SAE J1939协议

8.2.1 默认PGN配置

8.2.2 PGN65280(FF00)

8.2.3 PGN65281(FF01)

8.2.4 默认帧输出示例

8.2.5 写入配置参数

8.2.6 读取配置参数

9 附录A - 固件升级与恢复出厂设置

10 附录B-技术支持

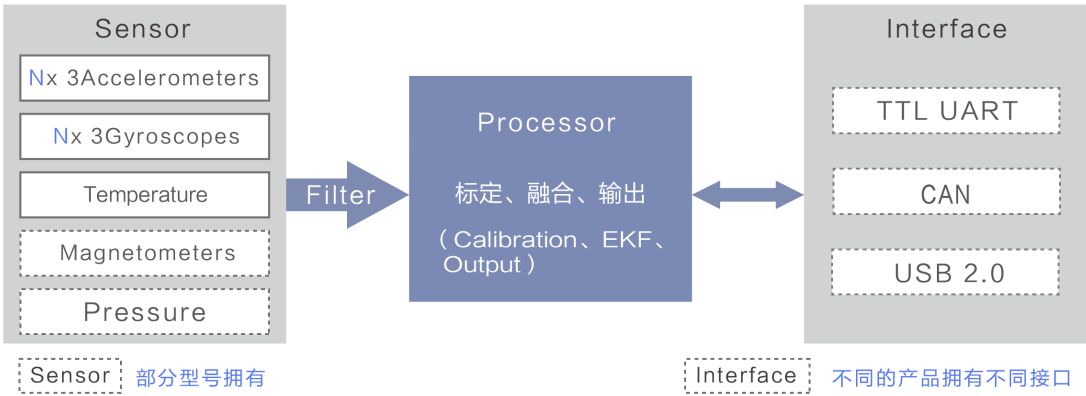
1. 产品简介

CH10X系列是利用高性能、小体积MEMS惯性器件感知物体姿态信息的姿态感知系统，它集成了惯性测量单元（IMU）、磁力计和一款搭载扩展卡尔曼融合算法（EKF）的微控制器。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，包含有绝对参考的航向角，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。IP68级防水的外壳式的封装可以很方便的集成到用户的系统，CH10X系列主要包含CH102(M)，CH104(M)，CH108M三个系列产品：

| 型号 | CH102 | CH102M | CH104 | CH104M | CH108 | CH108M |
|-------|-------|--------|-------|--------|-------|--------|
| 级别 | 工业级 | | 工业级 | | 战术级 | |
| IMU阵列 | 2 | | 4 | | 8 | |
| 地磁 | - | ● | - | ● | - | ● |

1.1 算法架构

CH10X采用超核最新的融合算法引擎，该算法引擎采用了自主研发的扩展卡尔曼滤波和IMU噪声动态分析技术，可以满足高动态下姿态角的精度，并且减小航向角的漂移。算法架构如图



1.2 主要应用场景：

CH10X系列可以精确地感知移动设备的俯仰（Pitch）、横滚（Roll）、航向（Yaw）等姿态信息，比如AR/VR，自动导引小车（AGV），巡检机器人、无人机等应用领域。它可与激光雷达（Lidar）、视觉（Camera）等导航方案形成优势互补，增强设备的导航精度，并减小对外界参考物体的依赖。典型的应用市场如下：

- 自动导引小车（AGV/AMR）
- 服务机器人
- 巡检机器人
- 组合导航（INS）
- 工程机械
- 智慧农机
- 倾角检测

1.3 产品特点

1.3.1 出厂标定

MEMS惯性传感器由于制造工艺、材质等问题，有一些共性的误差源，比如零偏（Bias）、比例因子（Scale Factor）、跨轴（Cross Axis），这些误差会在出厂之前会经过严格的标定程序，减少这些误差的影响，标定参数会保存在每一个产品中。

1.3.2 可输出多种数据类型

CH10X拥有数据预处理和基于四元数的扩展卡尔曼（EKF）数据融合引擎，因此可以为用户提供原始数据，姿态数据、四元数、温度等数据信息。详细介绍如下

- 加速度（Accleration） 含有重力并标定后的加速度
- 角速度（Angular Rates） 标定后的角速度
- 姿态角（Attitude） 俯仰（Pitch）、航滚（Roll）、航向（Yaw）
- 四元数（Quaternion） 姿态四元数
- 温度（Temprature） 传感器内部温度

1.3.3 数据接口

为更好地满足CH10X系列在各种场景下的应用，我们为CH10X系列配备了USB、232、CAN、485等数据接口，其中USB最高帧率可以达到400Hz,232最高速率400Hz,CAN2.0最高帧率200Hz。并且提供完善的Linux、ROS1/ROS2、Win、MCU驱动。

- CH10X系列CAN遵循CANopen协议和SAE J1939协议
- 如果使用232接口并且需要400Hz传输频率，那么用户的232芯片的波特率需要满足921600的波特率

1.3.4 其他

- 制造优势： 选用高性能惯性传感器，并且采用自动化批量标定与测试，保证产品一致性。
- 软件优势： 丰富的行业经验，融合先进的算法，可以在静止检测、慢速检测、快速启动，实时估算零偏、地磁自动校准、抗磁场干扰等方面，具有优势表现。

1.4 辅助开发套件

1.4.1 屏蔽双绞线



默认1.5m、0.5m如需其他长度可以与我们联系定做。

1.4.2 上位机软件CHCenter

CHCenter是我们为了用户快速评估产品而开发的一款PC端软件，可以自由的运行在WIN/Linux之上。
CHCenter具有如下特点：

- 数据显示
- 数据记录
- 数据分析
- 产品参数配置
- 固件升级



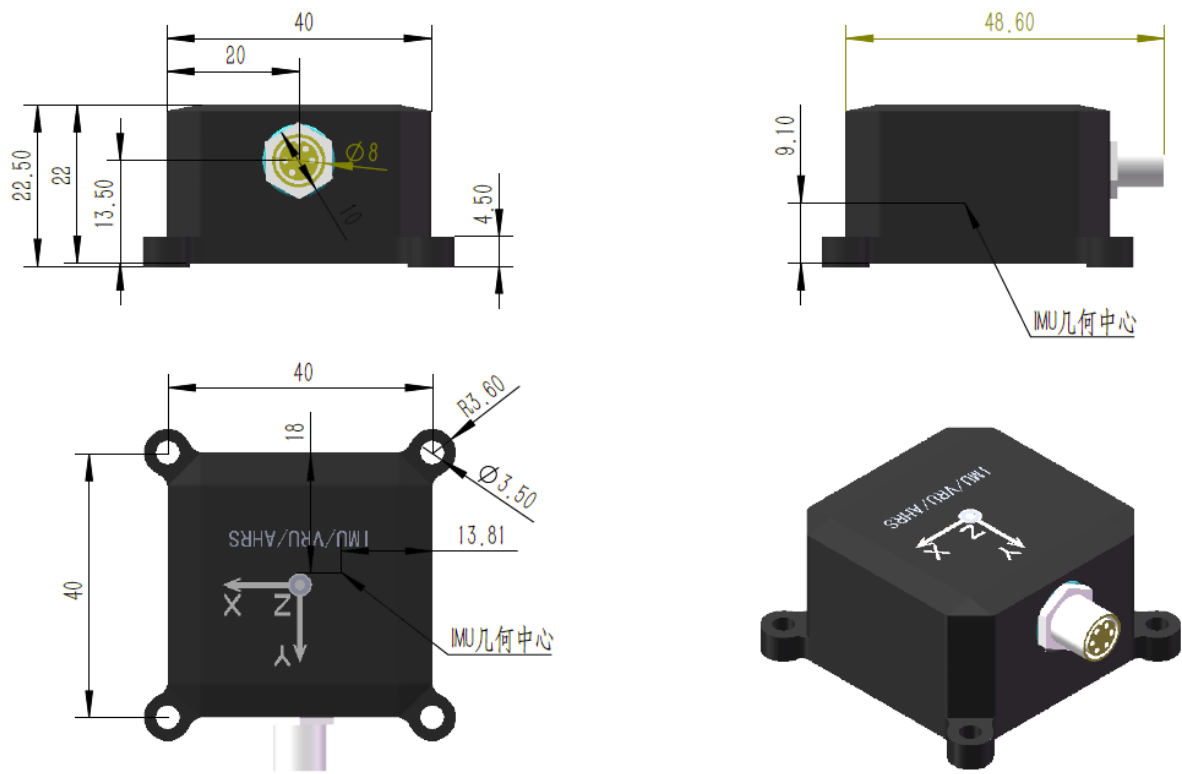
1.5 订购信息

| 型号 | P/N | 规格描述 | 级别 |
|----------------|------------|---------------------------------------|-----|
| CH102_USB | AM07-17001 | 6轴阵列轴姿态传感器 IMU VRU 外壳封装 USB输出 | 工业级 |
| CH102M_USB | AM07-17002 | 9轴阵列姿态传感器 IMU VRU AHRS外壳封装 USB输出 | 工业级 |
| CH102_CAN_232 | AM07-17003 | 6轴阵列姿态传感器 IMU VRU 外壳封装 CAN/232输出 | 工业级 |
| CH102M_CAN_232 | AM07-17004 | 9轴阵列轴姿态传感器 IMU VRU AHRS外壳封装 CAN/232输出 | 工业级 |
| CH102_485 | AM07-17005 | 6轴阵列姿态传感器 IMU VRU 外壳封装 485输出 | 工业级 |
| CH102M_485 | AM07-17006 | 9轴阵列姿态传感器 IMU VRU AHRS外壳封装 485输出 | 工业级 |
| CH104_USB | AM07-17007 | 6轴阵列轴姿态传感器 IMU VRU 外壳封装 USB输出 | 工业级 |
| CH104M_USB | AM07-17008 | 9轴阵列姿态传感器 IMU VRU AHRS外壳封装 USB输出 | 工业级 |
| CH104_CAN_232 | AM07-17009 | 6轴阵列轴姿态传感器 IMU VRU 外壳封装 CAN/232输出 | 工业级 |
| CH104M_CAN_232 | AM07-17010 | 9轴阵列轴姿态传感器 IMU VRU AHRS外壳封装 CAN/232输出 | 工业级 |
| CH104_485 | AM07-17011 | 6轴阵列姿态传感器 IMU VRU 外壳封装 485输出 | 工业级 |
| CH104M_485 | AM07-17012 | 9轴阵列姿态传感器 IMU VRU AHRS外壳封装 485输出 | 工业级 |
| CH108M_USB | AM07-17013 | 9轴阵列姿态传感器 IMU VRU AHRS外壳封装 USB输出 | 战术级 |
| CH108M_CAN_232 | AM07-17014 | 9轴阵列轴姿态传感器 IMU VRU AHRS外壳封装 CAN/232输出 | 战术级 |
| CH108M_485 | AM07-17015 | 9轴阵列姿态传感器 IMU VRU AHRS外壳封装 485输出 | 战术级 |

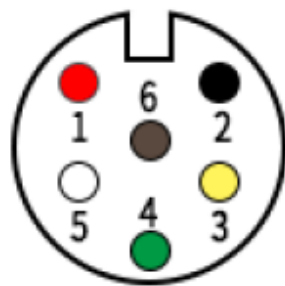
2. 机械与电气特性

2.1 CH10X尺寸图

所有标注单位均为mm



2.2 CH10X系列航插引脚定义



| 序号 | 颜色 | CH10X_USB(虚拟串口) | CH10X_CAN_232 | CH10X_485 |
|----|----|-----------------|---------------|------------|
| 1 | 红 | VCC_USB(5V) | VCC(5-36V) | VCC(5-36V) |
| 2 | 黑 | GND | GND | GND |
| 3 | 黄 | DN | H | - |
| 4 | 绿 | DP | L | - |
| 5 | 白 | - | TXD(RS232) | A(RS485) |
| 6 | 棕 | - | RXD(RS232) | B(RS485) |

CH10X_USB系列姿态传感器配专用的USB转接线，可以即插即用,VCC_USB供电电压绝对最大值5.2V

2.3 电气与机械参数

| 类型 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------|---------------|-------|-------|------|----|
| 电源输入VCC_USB | USB接口 | 4.9 | 5 | 5.2 | V |
| 电源输入VCC | CAN/232/485接口 | 5 | - | 36 | V |
| DN、DP | USB接口 | -0.3 | - | 3.6 | V |
| RS-232 | | -15 | - | 15 | V |
| CAN | | -36 | - | 36 | V |
| CAN差分电压 | | 1.5 | - | 3.0 | V |
| RS-485 | | -12 | - | 12 | V |
| 功耗P | CH102(M) | - | - | 206 | mW |
| 功耗P | CH104(M) | - | - | 245 | mW |
| 功耗P | CH108M | - | - | 410 | mW |
| 工作温度 | | -40 | - | 85 | °C |
| 振动冲击 | | -2000 | - | 2000 | g |
| MTBF | | | 50000 | | h |

EMC防护标准

USB接口: IEC61000-4-2 ESD 15KV Air, 8KV contact compliance

CAN/232/485: IEC61000-4-2 ESD 30KV Air, 30KV contact compliance

EFT, IEC 61000-4-4,50A (5/50ns)

Lightning, IEC 61000-4-5 2nd edition, 10A

2.4 坐标系定义

载体系使用 右-前-上(RFU)坐标系，地理坐标系使用 东-北-天(ENU)坐标系。其中欧拉角旋转顺序为东-北-天-312(先转Z轴，再转X轴，最后转Y轴)旋转顺序。具体定义如下：

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

如果将模块视为飞行器的话。Y轴正方向应视为机头方向。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°

3. 性能指标

3.1 姿态角输出精度

为了提升模块的精度与稳定性，我们将CH10X系列传感器系统科学地排布在PCB上，并且搭载新一代EKF算法，使得模块性能得以充分发挥。

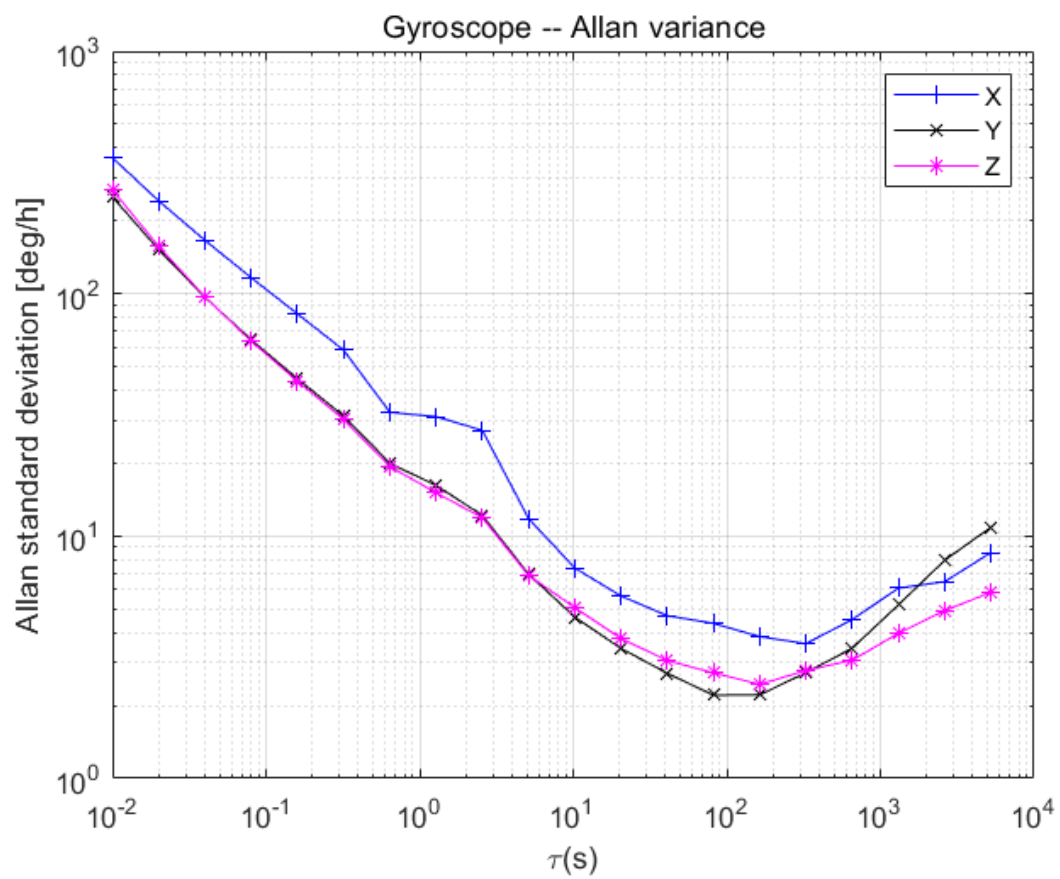
| 型号 | CH102 | CH102M | CH104 | CH104M | CH108M |
|--------------------------------|-------|--------|-------|--------|--------|
| Pitch/Roll(static) | 0.2° | 0.2° | 0.2° | 0.2° | 0.2° |
| Pitch/Roll(dynamic) | 0.2° | 0.2° | 0.2° | 0.2° | 0.2° |
| Yaw(VRU)(static) ^① | 0.3° | 0.2° | 0.2° | 0.2° | 0.2° |
| Yaw(VRU)(dynamic) ^② | 5° | 5° | 3° | 3° | 2° |
| Yaw(magnetic) ^③ | - | 2° | - | 2° | 2° |
| Yaw(VRU) ^④ | 0.3° | 0.3° | 0.2° | 0.2° | 0.1° |
| 气压计（高度） | - | - | - | - | - |

- ①模块在25℃绝对静止1h测得
- ②模块在机器人上运动1h测得，室温25℃，1 σ
- ③地磁校准之后，周边无磁场干扰情况下室温25℃测得，CH010M、CH020M、CH040MP需要配置为9轴模式,配置方法参考AT指令章节
- ④转台转10圈取每圈平均值，室温25℃

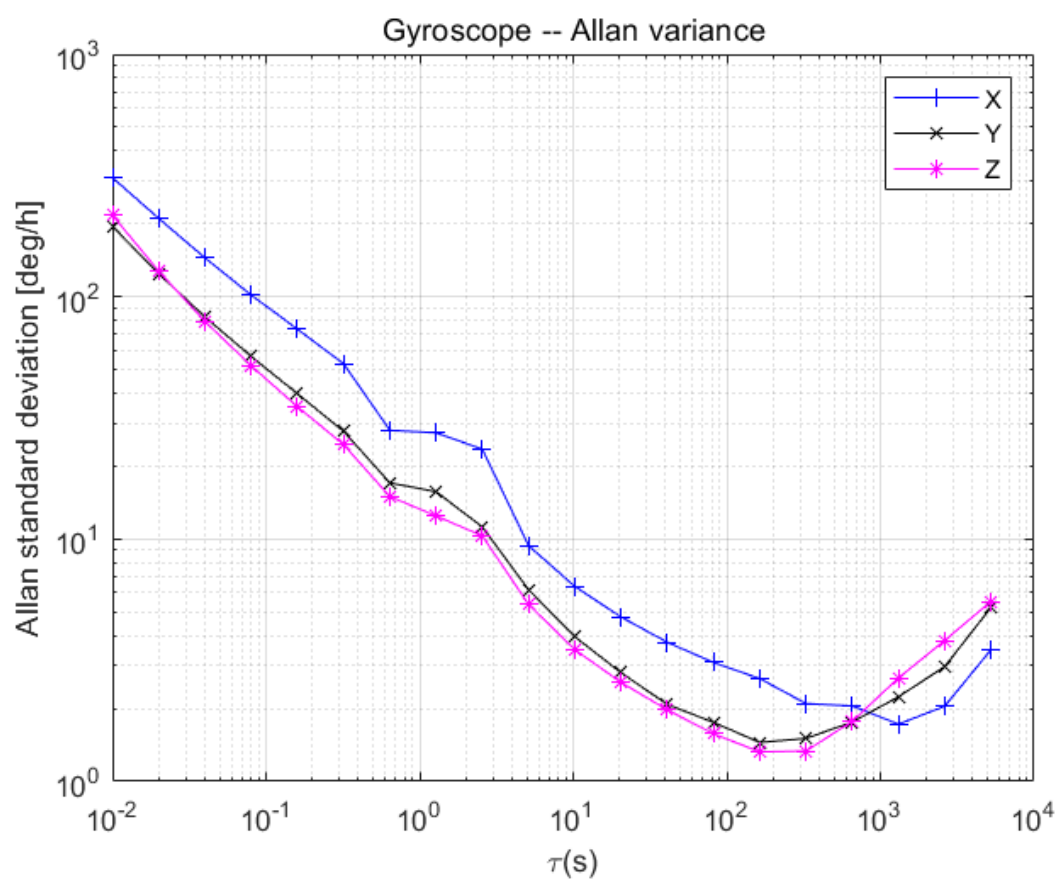
3.2 陀螺仪

| 参数 | CH102(M) | CH104(M) | CH108M | 备注 |
|-----------|--------------------|-------------------|--------------------|-----------------|
| 测量范围 | ±2000°/s | ±2000°/s | ±2000°/s | 可配置 |
| 分频率 | 0.01°/s | 0.01°/s | 0.01°/s | |
| 零偏稳定性 | 3.5°/hr | 2.5°/hr | 1.76°/hr | @25℃,1 σ |
| 零偏重复性 | 0.07°/s | 0.05°/s | 0.03°/s | @25℃,1 σ |
| 非正交误差 | ±0.1% | ±0.1% | ±0.1% | @25℃,1 σ |
| 随机游走 | 0.42°/ \sqrt{hr} | 0.3°/ \sqrt{hr} | 0.21°/ \sqrt{hr} | @25℃,1 σ |
| 刻度非线性度 | ±0.1% | ±0.1% | ±0.1% | 满量程时(最大) |
| 刻度系数误差 | 800ppm | 550ppm | 300ppm | 出厂前校准后 |
| 加速度敏感性 | 0.1°/s/g | 0.1°/s/g | 0.1°/s/g | |
| Z轴全温变化最大值 | 0.003°/s/℃ | 0.003°/s/℃ | 0.003°/s/℃ | -40℃ - 85℃ |

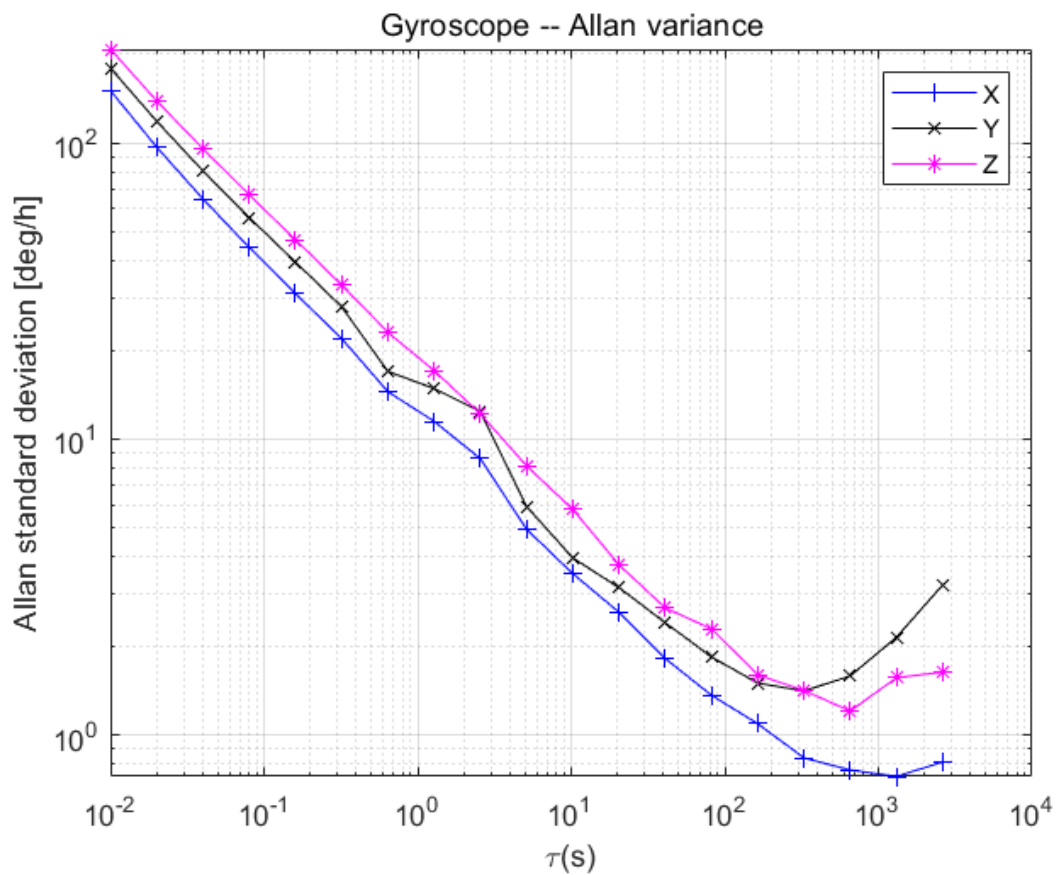
陀螺Allan方差曲线(CH102)



陀螺Allan方差曲线(CH104)



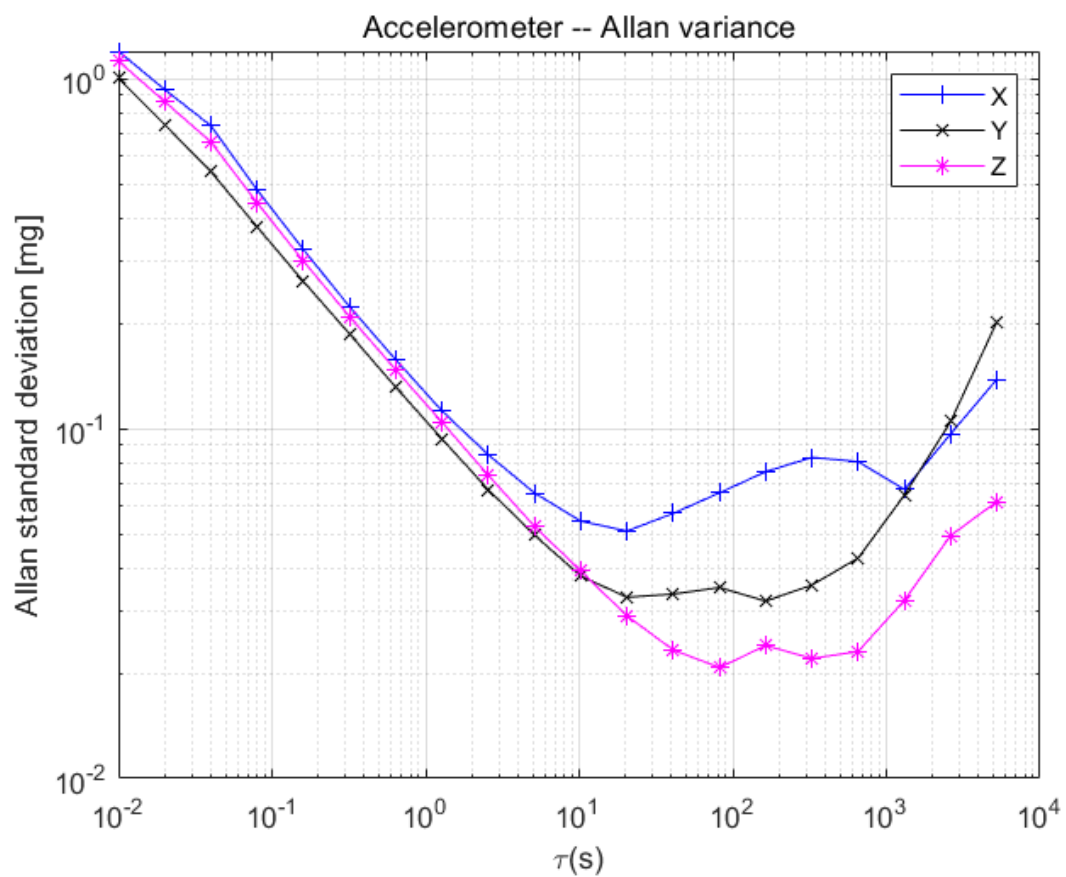
陀螺Allan方差曲线(CH108M)



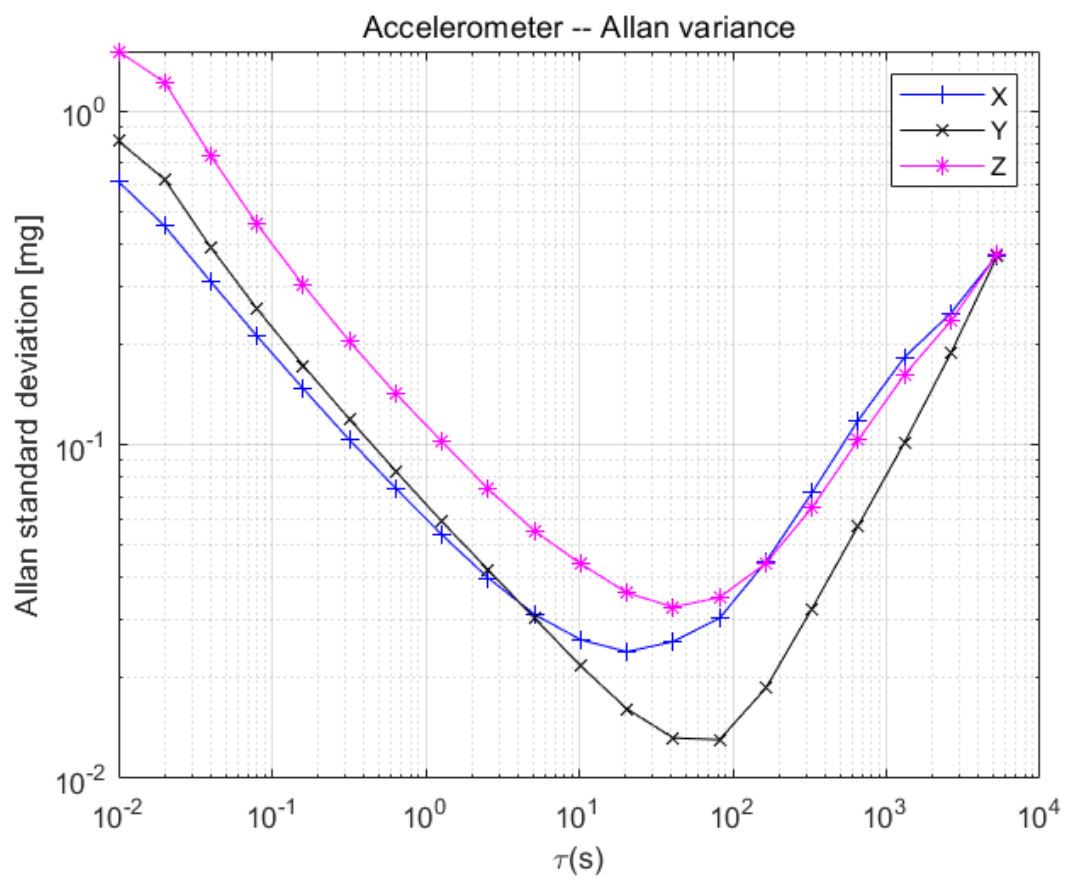
3.3 加速度计

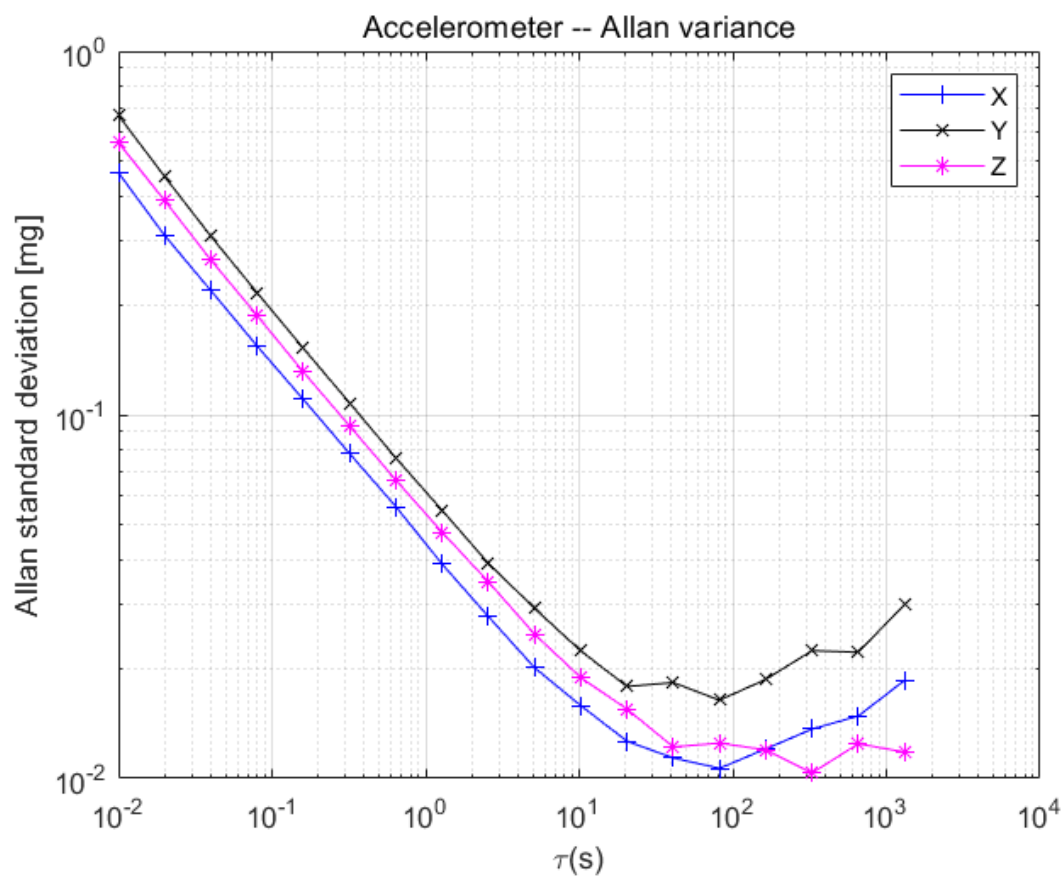
| 参数 | CH102 | CH104 | CH108 | 备注 |
|-------------|---------------------|---------------------|----------------------|--------------|
| 测量范围 | ±8G (1G = 1x 重力加速度) | ±8G (1G = 1x 重力加速度) | ±24G (1G = 1x 重力加速度) | |
| 分辨率 | 1uG | 1uG | 1uG | |
| 零偏稳定性 | 42uG | 30uG | 21uG | @25°C,1σ |
| 零偏重复性 | 2.52mG | 1.8mG | 0.6mG | @25°C,1σ |
| 非正交误差 | ±0.1% | ±0.1% | ±0.1% | ±0.1% |
| 随机游走 | $0.056m/s\sqrt{h}$ | $0.04m/s\sqrt{h}$ | $0.028m/s\sqrt{h}$ | @25°C,1σ |
| 刻度系数误差 | ±0.3% (满量程时) | ±0.3% (满量程时) | ±0.2% (满量程时) | 出厂前校准后 |
| 全温范围温度变化最大值 | ±0.45mg/°C | ±0.45mg/°C | ±0.45mg/°C | -40°C - 85°C |

加速度Allan方差曲线(CH102)



加速度Allan方差曲线(CH104)





3.4 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 |

3.5 关键电气参数

| 指标 | CH102系列 | CH104系列 | CH108M |
|--------------|---|----------|-------------|
| 机械与电气 | | | |
| 尺寸 | 40x40x24.5mm(螺丝孔间距) | | |
| 重量 | 73g | 74g | 76g |
| 电压 | USB接口：5V CAN/232/485接口5-36V | | |
| 功耗 | 206mW | 245mW | 410mW |
| 启动时间与方式 | 静止启动时间<1s | | |
| 接口-USB | | | |
| USB(虚拟串口波特率) | 9600/115200(默认)/230400/460800/921600 | | |
| 模块输出帧率 | 1/50/100(默认)/200/400Hz | | |
| 传感器最大输出帧率 | 400Hz原始数据(加速度,角速度) | 400Hz姿态角 | 100Hz磁场原始数据 |
| 接口-串口RS-232 | | | |
| 串口波特率 | 9600/115200(默认)/230400/460800/921600 | | |
| 模块输出帧率 | 1/50/100(默认)/200/400Hz | | |
| 传感器最大输出帧率 | 400Hz原始数据(加速度,角速度) | 400Hz姿态角 | 100Hz磁场原始数据 |
| 接口-CAN2.0 | | | |
| 波特率 | 125K/250K/500K(默认)/1000K | | |
| 协议 | CANopen/SAEJ1939 | | |
| 最大数据速率 | 200Hz原始数据（加速度，角速度） | 200Hz姿态角 | 100Hz磁场数据 |
| 接口-485 | | | |
| 协议 | Modbus/私有 | | |
| 最大数据速率 | 100Hz原始数据(加速度，角速度) | 100Hz姿态角 | 100Hz磁场数据 |
| 环境 | | | |
| ESD | USB接口:IEC61000-4-2 ESD 15KV Air, 8KV contact compliance CAN/232/485:IEC61000-4-2 ESD 30KV Air, 30KV contact compliance EFT, IEC 61000-4-4,50A (5/50ns) Lightning, IEC 61000-4-5 2nd edition, 10A | | |
| 工作/存储温度 | -40℃-85℃ | | |
| 振动 | 1mm (0-100Hz) &<18g(100Hz-2kHz) | | |
| 冲击 | 2000g | | |
| MTBF | 50000h | | |

如果用232接口进行400hz传输数据，用户的232接口芯片需要支持921600的波特率

4. 标定

4.1 出厂标定

加速度计和陀螺仪出厂前会经过标定，这些被标定的误差包括比例因子、零偏、非正交误差以及温度。校准参数会被写在模块内部。

4.2 启动标定

为了模组获得更好的性能，我们还为用户提供了上电自动标定的功能，这需要用户在上电的过程中保持水平静止1s，这样可以获得更准确的零偏参数。我们独有的启动算法可以模组很好的避免上电零偏计算错误，充分发挥模块性能。

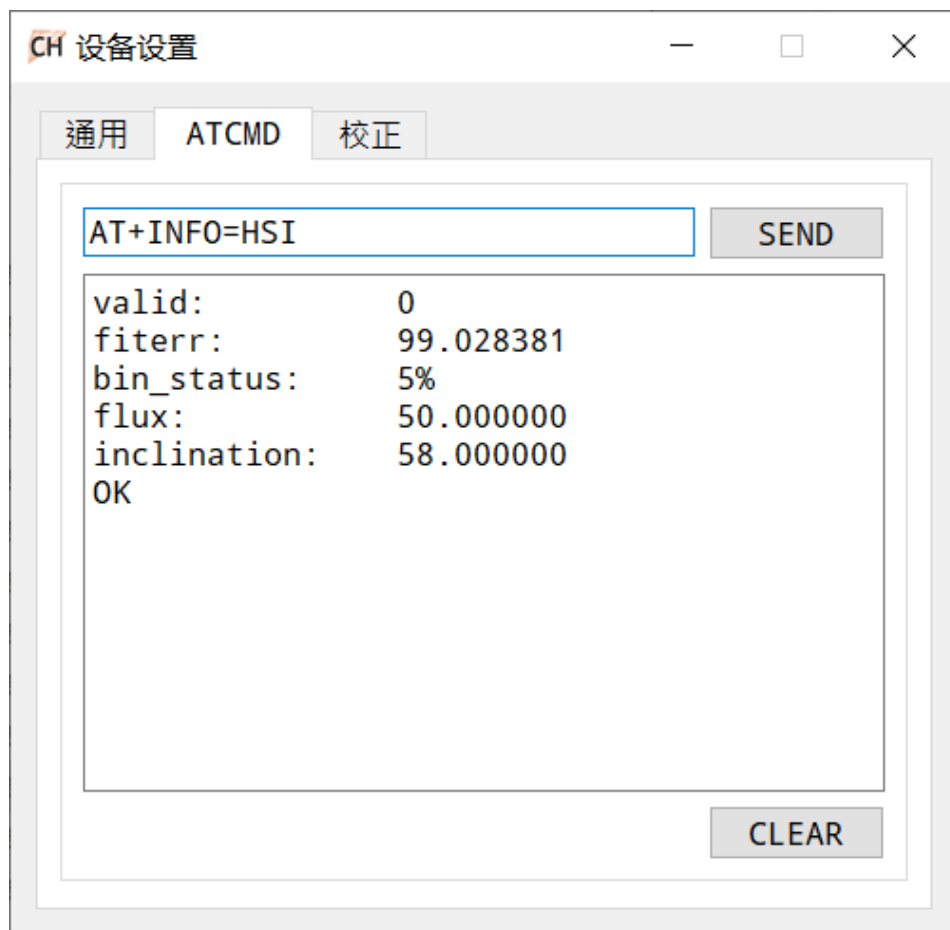
4.3 磁力计校准

地磁传感器出厂前经过椭球校准，但磁传感器很容易受到外界环境磁场干扰，一般都需要客户在拿到产品后重新校准。

模块内部自带主动地磁校准系统，该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，并且维持一定时间，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9轴模式时，应进行如下校准操作：

1. 检查周围是否存在磁场干扰：室内桌子旁，大型铁质框架结构附近，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(>0.5m)实验室桌子、电脑、电机、手机等容易产生地磁干扰的物体。
2. 在尽量小范围内(位置不动，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)。一般情况下即可完成校准。如果始终没能成功校准模块，说明周围地磁场干扰比较大。
3. 校准的成功与否可用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，如图4-1所示：这里只需关心fiterr项即可：0.03以下说明校准结果已经足够好。如果fiterr始终>0.1，说明地磁干扰很大，需要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。
4. 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，或者是模块被安装/焊接到了新的环境中)，还需重复执行1-3步。
5. 虽然地磁校准无需手动开启或停止，模块自动在后台自动运行该系统，但是用户仍然可以手动控制地磁校准系统的开启与否，使用 `AT+MCALCTL=0` 来关闭地磁校准系统，使用 `AT+MCALCTL=1` 来开启地磁校准系统，该命令立即生效，掉电保存。用户可以开启地磁校准后使 `AT+INFO=HSI` 来判断校准质量，一旦校准成功完成，使用 `AT+MCALCTL=0` 来关闭校准系统并锁定校准值。一般情况下，只要在没有磁区域校准成功过一次，后面是无需再次校准的。
6. 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。



4.3.1 磁干扰分类

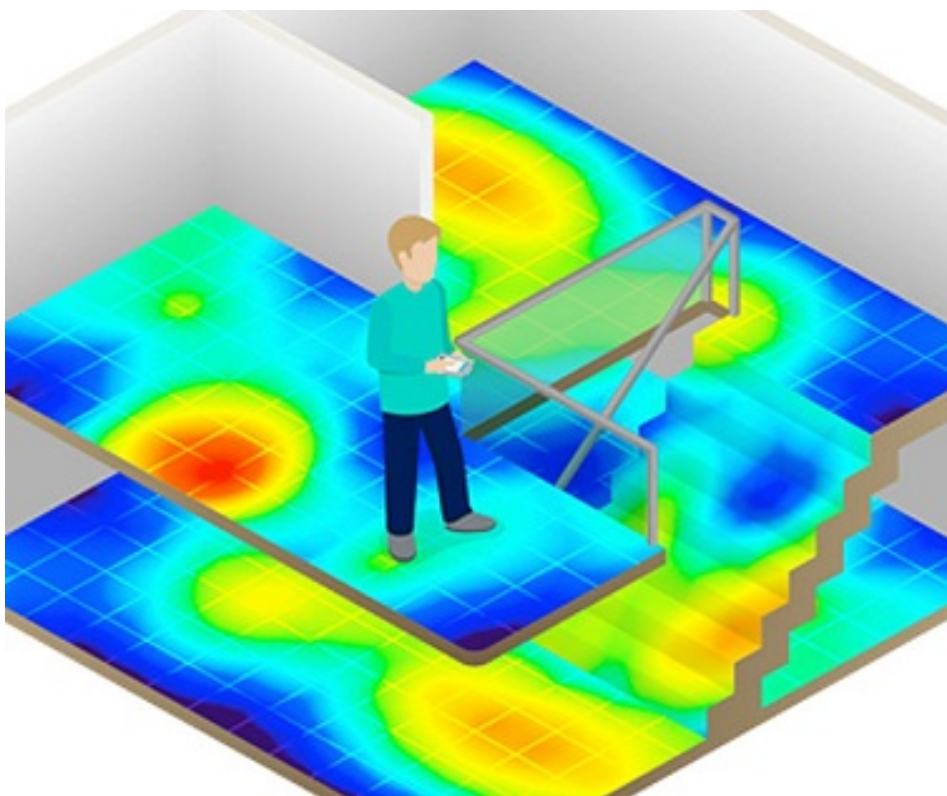
地磁干扰可分为空间磁场干扰与传感器坐标系下的磁场干扰，如下图所示

| 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. |

4.3.1.1 空间磁场干扰(干扰不随传感器位姿改变而改变)

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

4.3.1.2 传感器坐标系下的干扰(干扰随传感器位姿改变而改变)

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，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。

4.3.2 地磁使用注意事项

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)，即使校准后9轴的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的，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 50cm以上)。

4.3.3 案例分析

假设客户想在移动机器人上使用9轴模式获得准确的不漂移的航向角，模块安装在机器人上(看做一个刚体)，由于机器人本身的金属结构(部件，电路)会有一个很大的硬磁干扰，相当于上文提到的"传感器坐标系下的干扰"。这部分干扰可以被校准。由于机器人的电机启动停止，以及机器人在室内经过各种房间的磁干扰导致空间磁场改变，会产生上文中提到的"空间磁场干扰"。这部分干扰不能被校准。两种干扰同时存在的可能性很大，给9轴模式造成了很大挑战。这时候建议客户用6轴模式，如果9轴模式必须使用则要做到如下几点：

1. 校准：必须带着机器人一起校准(机器人足够小)，单独把模块拿下来校准好再安装上去是不正确的做法。必须将机器人和模块视做一个刚体去校准才能得到正确的校准结果，具体校准环节请参考上文，校准成功后，重新上电(复位)生效。
2. 由于室内磁环境复杂，即使校准正确完成可能还是会出现较大的航线误差，尤其是电机启停，功率改变时，对磁场影响巨大。

如果应用场合不需要地磁传感器，可以忽略地磁传感器，产品默认处于6轴模式，即地磁场不参与解算，不建议机器人行业使用9轴姿态传感器

5. 关于模式说明

5.1 型号的模式区别

CH10X每个系列支持的模式如下表：

| 型号 | CH102 | CH102M | CH104 | CH104M | CH108M |
|------------|-------|--------|-------|--------|--------|
| 加速度计 | ● | ● | ● | ● | ● |
| 陀螺仪 | ● | ● | ● | ● | ● |
| 地磁 | - | ● | - | ● | ● |
| 气压 | - | - | - | - | ● |
| 6轴(VRU)模式 | ● | ● | ● | ● | ● |
| 9轴(AHRS)模式 | - | ● | - | ● | ● |

5.2 不同模式的区别

| 模式 | 使用环境 | 典型应用 | 优点 | 缺点 |
|----|----------|--------------|--|---|
| 6轴 | 不用考虑磁场影响 | 无人驾驶 倾角检测 | 姿态角输出稳定性 好完全不受磁场干扰 | 在没有外界参考条件下航向角随时间缓慢漂移 |
| 9轴 | 需要考虑磁场影响 | 指南针 寻北系统 | 航向角不会随时间 漂移一旦检测到地 磁场可快速修正航 向角指北 | 任何磁干扰都会使航向角准确度下降。干扰严重的情况下航向角无法指向正确方向。使用前需要校准传感器 |

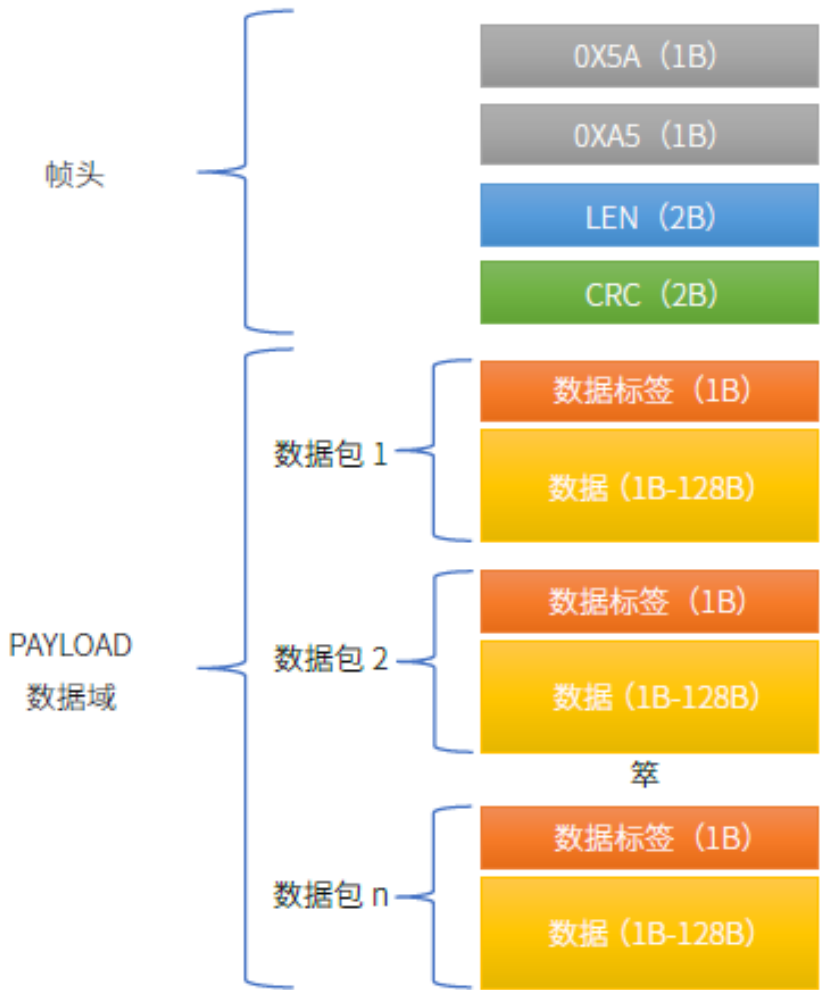
支持地磁的产品，无论在哪种模式下都会输出地磁信息，6轴模式地磁不会参与解算，9轴模式地磁会参与解算

6. 232串口通讯协议

模块上电后，默认按出厂帧率(通常为100Hz)输出帧数据，帧格式如下：

- 1 串口数据帧结构：
- <帧头(0x5A)><帧类型(0xA5)><长度><CRC校验><数据域>

| 域名称 | 值 | 长度(字节) | 说明 |
|-------|-------|--------|---|
| 帧头 | 0x5A | 1 | 固定为0x5A |
| 帧类型 | 0xA5 | 1 | 固定为0xA5 |
| 长度 | 1-512 | 2 | 帧中数据域的长度，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 本身外其余所有字段(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。每个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。 |



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


6.1 串口数据包

| 数据包标签 | 数据包长度(包含标签1字节) | 名称 | 备注 |
|-------|----------------|--------|---------|
| 0x91 | 76 | IMUSOL | IMU数据集合 |

6.2 产品支持数据包列表

6.2.1 0X91(IMUSOL)

共76字节。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 | uint8_t | 1 | - | 保留 |
| 3 | int8_t | 1 | °C | 模块平均温度 |
| 4 | float | 4 | Pa | 气压(部分型号支持) |
| 8 | uint32_t | 4 | ms | 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 | float | 12 | 1G(1G = 1重力加速度) | 经过出厂校准后的加速度,顺序为：XYZ |
| 24 | float | 12 | deg/s | 经过出厂校准后的角速度,顺序为：XYZ |
| 36 | float | 12 | uT | 磁强度,顺序为：XYZ |
| 48 | float | 12 | deg | 节点欧拉角 顺序为：横滚角(Roll, -180°~180°)，俯仰角(Pitch, -90°~90°)，航向角(Yaw, -180°~180°) |
| 60 | float | 16 | - | 节点四元数集合,顺序为WXYZ |

6.3 数据帧结构示例

6.3.1 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` : $(0x00<<8) + 0x4C = 76$

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

- 第二步：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, 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, 定义数据结构体和常用转换宏:

```
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     uint32_t id; /* user define ID */
15     float acc[3]; /* acceleration */
16     float gyr[3]; /* angular velocity */
17     float mag[3]; /* magnetic field */
18     float eul[3]; /* attitude: euler angle */
19     float quat[4]; /* attitude: quaternion */
20     float pressure; /* air pressure */
21     uint32_t timestamp;
22 }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 strat at buf[6] */
3     i0x91.tag =          U1(buf+offset+0);
4     i0x91.id =          U1(buf+offset+1);
5     i0x91.pressure =     R4(buf+offset+4);
6     i0x91.timestamp =   U4(buf+offset+8);
7     i0x91.acc[0] =       R4(buf+offset+12);
8     i0x91.acc[1] =       R4(buf+offset+16);
9     i0x91.acc[2] =       R4(buf+offset+20);
10    i0x91.gyr[0] =        R4(buf+offset+24);
11    i0x91.gyr[1] =        R4(buf+offset+28);
12    i0x91.gyr[2] =        R4(buf+offset+32);
13    i0x91.mag[0] =        R4(buf+offset+36);
14    i0x91.mag[1] =        R4(buf+offset+40);
15    i0x91.mag[2] =        R4(buf+offset+44);
16    i0x91.eul[0] =        R4(buf+offset+48);
17    i0x91.eul[1] =        R4(buf+offset+52);
18    i0x91.eul[2] =        R4(buf+offset+56);
19    i0x91.quat[0] =       R4(buf+offset+60);
20    i0x91.quat[1] =       R4(buf+offset+64);
21    i0x91.quat[2] =       R4(buf+offset+68);
22    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%d\r\n",            "id:",            i0x91.id);
3     printf("%-16s%8.4f %8.4f %8.4f\r\n", "acc(G):",        i0x91.acc[0],
i0x91.acc[1], i0x91.acc[2]);
4     printf("%-16s%8.3f %8.3f %8.3f\r\n", "gyr(deg/s):",    i0x91.gyr[0],
i0x91.gyr[1], i0x91.gyr[2]);
5     printf("%-16s%8.3f %8.3f %8.3f\r\n", "mag(uT):",      i0x91.mag[0],
i0x91.mag[1], i0x91.mag[2]);
6     printf("%-16s%8.3f %8.3f %8.3f\r\n", "eul(deg):",      i0x91.eul[0],
i0x91.eul[1], i0x91.eul[2]);
7     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]);
8     printf("%-16s%8.3f\r\n",            "pressure(pa):",
i0x91.pressure);
9     printf("%-16s%d\r\n",              "timestamp(ms):",
i0x91.timestamp);
```

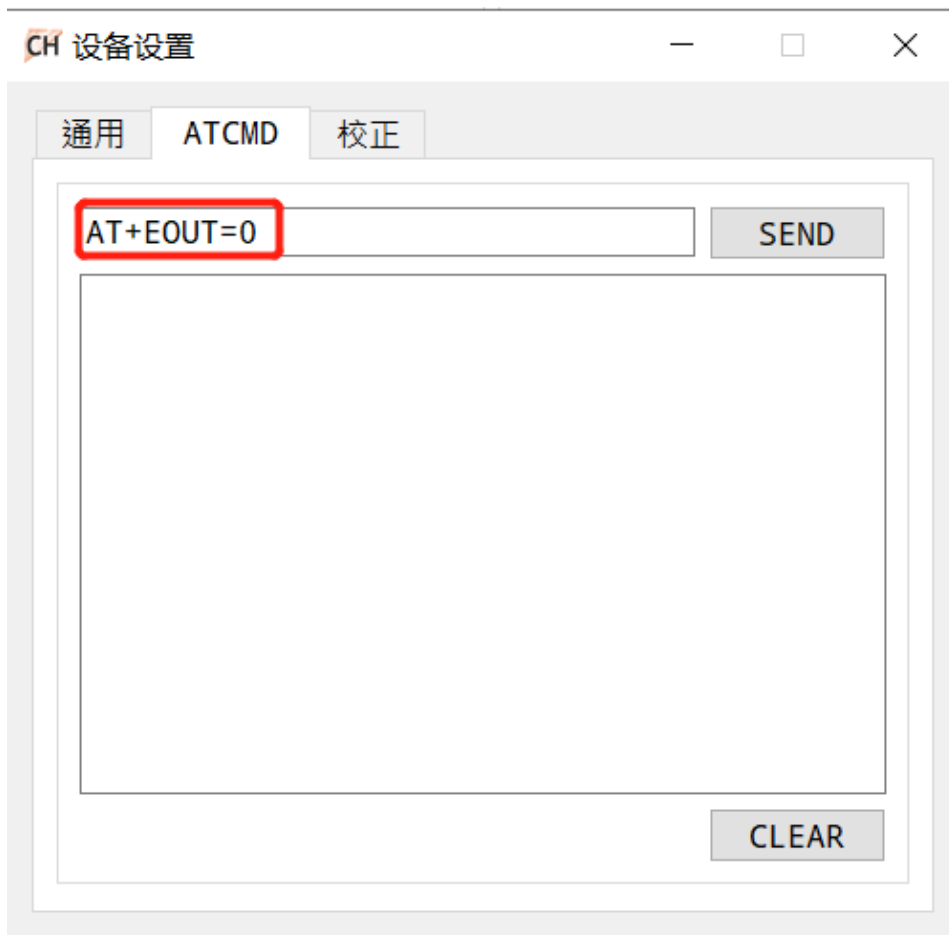
打印出的解析结果:

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

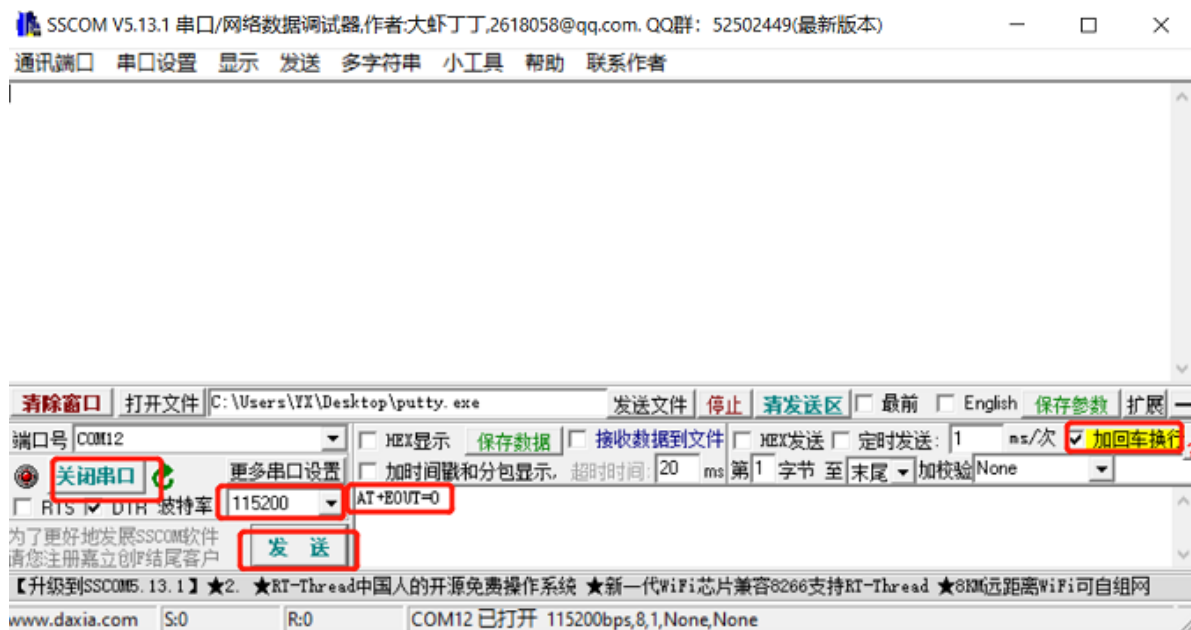
7. AT指令

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。

使用上位机输入AT指令：



使用串口调试助手进行测试:



7.1 AT指令列表

| 指令 | 功能 | 掉电保存(Y) /掉电不保存(N) | 立即生效(Y)/复位生效(N) | 备注 |
|-----------|-------------|-------------------|-----------------|--------|
| AT+ID | 设置模块用户ID | Y | N | |
| AT+INFO | 打印模块信息 | N | Y | |
| AT+ODR | 设置模块串口输出帧频率 | Y | N | |
| AT+BAUD | 设置串口波特率 | Y | N | |
| AT+EOUT | 数据输出开关 | N | Y | |
| AT+RST | 复位模块 | N | Y | |
| AT+URFR | 安装角度设置 | Y | N | |
| AT+SETYAW | 手动设置航向角 | N | Y | |
| AT+RSTORT | 安装对齐设置 | Y | 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 | N | 部分型号支持 |

7.2 指令详解

7.2.1 AT+ID

设置模块用户ID

例 AT+ID=1

7.2.2 AT+INFO

打印模块信息，包括产品型号，版本，固件发布日期等。

7.2.3 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

当ODR设置为比较高时(如200),默认的115200波特率可能不满足输出带宽要求，此时需要将模块波特率设高(如921600)后，模块才能按设置的ODR输出数据帧。输出帧率可以为1,2,5,10,20,50,100,200,400Hz。

7.2.4 AT+BAUD

设置串口波特率，可选值：9600/115200/460800/921600

例 AT+BAUD=115200

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。

7.2.5 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`

7.2.6 AT+RST

复位模块

例 `AT+RST`

7.2.7 AT+SETYAW

设置航向角，格式为 `AT+SETYAW=<MODE>,<VAL>`

- MODE=0 绝对模式：将航向角直接设置为VAL的值。如 `AT+SETYAW=0,90` 将航向角直接设置为90°
- MODE=1 相对模式：将原航向角递增VAL值。如 `AT+SETYAW=1,-10.5` 将航向角递增-10.5°，如原来为30°，发送命令后航向角变为19.5°。

7.2.8 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`

该指令仅支持CH010M、CH020M、CH040MP，不支持CH010、CH020、CH040

7.2.9 AT+URFR

这条指令提供了旋转传感器XYZ轴的接口，可用于任意角度的垂直安装。

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

其中 C_{nm} 支持浮点数

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_U = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B \quad (1)$$

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°(适用于垂直安装:Y轴正方向朝下)， 输入命令：

`AT+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°(适用于垂直安装:Y轴正方向朝上)， 输入命令：

`AT+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°， 输入命令： `AT+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°(适用于垂直安装:X轴正方向朝上)， 输入命令：

`AT+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°(适用于垂直安装:X轴正方向朝下)， 输入命令：

`AT+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°， 输入命令： `AT+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°， 输入命令： `AT+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°， 输入命令： `AT+URFR=0,1,0,-1,0,0,0,0,1`

- 恢复出厂默认值： `AT+URFR=1,0,0,0,1,0,0,0,1`

7.2.10 AT+RSTORT

一些情况下，模块安装时需要设置水平对齐(Pitch=Roll=0) 或者全部对齐(Pitch=Roll=Yaw=0)，这是需要给模块的一个初始姿态偏移。使用此命令可以完成对齐配置。

- `AT+RSTORT=0` : Object Reset: Pitch = Roll = Yaw = 0
- `AT+RSTORT=1` : Heading Reset: Pitch = Roll = 0, Yaw保持不变
- `AT+RSTORT=2` : Alignment Reset: Pitch, Roll 保持不变, Yaw=0
- `AT+RSTORT=3` : Reset Offset: 清除所有对齐偏移设置

8. CAN通讯协议

CAN通讯协议可支持 CANopen协议和SAE J1939协议。模块默认支持CANopen协议，如需J1939协议版本，请联系技术支持。

8.1 CANopen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PT01-4 传输数据。不接收/发送远程帧和拓展数据帧
- 所有PT0采用异步定时触发模式。

8.1.1 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 | Operational |
| 心跳包 | 无 |
| TPDO输出速率 | 10Hz - 200Hz(每个TPDO) |

8.1.2 CANopen TPDO

| PTO通道 | PTO 帧ID | 长度 (DLC) | PTO 传输方式 | 异步输出频率 (Hz) | 发送数据 | 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(0.001重力加速度) |
| 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. |
| TPDO5 | 0x680+ID | 4 | 异步定时 (0xFE) | 20 | 气压 | 类型:int32 共4字节。单位Pa |

例:

收到加速度CAN帧: ID=0x188,DLC=6,DATA = 4A 00 1F 00 C8 03

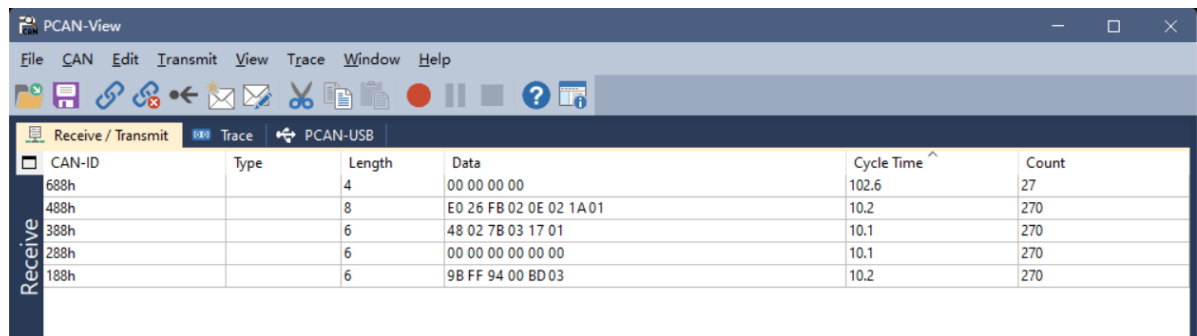
- ID=0x188: ID为8的设备发送的加速度数据帧
- 加速度X轴 = 0x004A = 74 = 74mG
- 加速度Y轴 = 0x001F = 731 = 31mG
- 加速度Z轴 = 0x03C8 = 968 = 968mG

收到角速度CAN帧: ID=0x288,DLC=6,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 = 0x0113 = 275 = 27.5dps
- 角速度Z轴 = 0x0023 = 35 = 3.5dps

8.2 使用上位机连接CAN设备

使用PCAN-View工具，配合PCAN-USB，可以在接收框(Rx Message)中会显示收到的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' button is highlighted. The 'PCAN-USB' interface is selected. A table displays received messages with columns for CAN-ID, Type, Length, Data, Cycle Time, and Count. The messages are filtered by ID 188h.

| 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 |

8.2.3 CANopen接口常用命令

8.2.3.1 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

数据字典以下位置存放厂商参数配置数据, 可通过CANopen 发送快速SDO指令修改，**掉电保存，重新上电生效。**

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

以上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:

| PTO通道 | PTO 帧ID | TPDO参数索引地址(CANopen协议默认定义) |
|-------|----------|---------------------------|
| TPDO1 | 0x180+ID | 0x1800 |
| TPDO2 | 0x280+ID | 0x1801 |
| TPDO3 | 0x380+ID | 0x1802 |
| TPDO4 | 0x480+ID | 0x1803 |
| TPDO5 | 0x680+ID | 0x1804 |

8.2.3.2 修改CAN波特率

修改波特率:(ID=0x608, 长度为8的标准数据帧, 重新上电生效)

- CAN波特率修改为125K: ID=0x608 , DLC=8, DATA=23,00,21,00,48,E8,01,00
- CAN波特率修改为250K: ID=0x608 , DLC=8, DATA=23,00,21,00,90,D0,03,00
- CAN波特率修改为500K: ID=0x608 , DLC=8, DATA=23,00,21,00,20,A1,07,00
- CAN波特率修改为1M: ID=0x608 , DLC=8, DATA=23,00,21,00,40,42,0F,00

以设置波特率为125KHz命令为例:

- ID=0x608为快速写SDO地址,其中8为默认节点ID, 修改节点ID后要做相应修改,如CANopenID改为9后,ID=0x609.
- 0x23为SDO写四个字节指令
- 0x00, 0x21为写0x2100索引
- 0x00 子索引位置, 默认0
- $(4-7)0x00, 0x01, 0xE8, 0x48 = (0x00 \ll 24) + (0x01 \ll 16) + (0xE8 \ll 8) + 0x48 = 125000$

8.2.3.3 修改节点ID

如将设备CANopen节点ID改为9, 重新上电生效

ID=0x608 , DLC=8, DATA=23,01,21,00,09,00,00,00

- 0x23为SDO写四个字节指令
- 0x01, 0x21为写0x2101索引
- $0x09\ 0x00, 0x00, 0x00 = (0x00 \ll 24) + (0x00 \ll 16) + (0x00 \ll 8) + 0x09 = 9$

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

8.2.3.4 修改数据输出速率

此项配置立即生效, 掉电保存

修改TPDO3(欧拉角)输出速率为20Hz(每50ms输出一次):

ID=0x608 , DLC=8, DATA=2B,02,18,05,32,00,00,00

其中

- 0x2B为SD0写两个字节指令
- 0x02, 0x18为写0x1802索引,
- 0x05为子索引
- 0x00, 0x32= (0x00<<8) + 0x32 = 50(单位为ms), 后面不足补0.

将TPDO1(加速度)输出速率修改为10Hz(每100ms输出一次): 2B 00 18 05 64 00 00 00

将TPDO2(角速度)输出速率修改为5Hz(每200ms输出一次): 2B 01 18 05 C8 00 00 00

8.2.3.5 关闭数据输出

同修改数据输出速率，只不过将数据输出速率设置每0ms输出一次(每0ms输出一次代表关闭, 重新上电生效):

将TPDO2(角速度)定时输出为0: 2B,01,18,05,00,00,00,00

将TPDO1(加速度)输出速率为0: 2B 00 18 05 00 00 00 00

将TPDO3(欧拉角)输出速率为0: 2B 02 18 05 00 00 00 00

将TPDO4(四元数)输出速率为0: 2B 03 18 05 00 00 00 00

8.2.3.6 开启/关闭站点

使用 NMT命令StartRemoteNode和 StopRemoteNode来开启关闭节点:

- 开启节点: ID:0,DLC:2,DATA:01 08 其中01为开启节点命令, 08为节点ID(出厂默认为8)
- 关闭节点: ID:0,DLC:2,DATA:02 08 其中02为关闭节点命令, 08为节点ID(出厂默认为8)

8.2.3.7 配置TPDO为同步模式

先关闭所有TPDO(设置TPDO输出速率为0), 然后发送CANopen同步帧即可:

CANopen 同步帧: ID:80 DLC:0, DATA:空

8.3 SAE J1939协议

8.3.1 默认PGN配置

| PGN 52810,52811 | 描述 |
|------------------|-------------------------|
| 通讯模式 | 广播通信 |
| 传输时间间隔 | 10ms |
| 数据长度 | 每个PGN8字节(两个PGN在同一个时刻输出) |
| PF(PDU format) | 0xFF |
| PS(PDU specific) | 0x00 |
| 优先级 | 6 |
| 地址 | 0x08 |

8.3.2 PGN65280(FF00)

双轴倾角仪角度信息及加速度XY(dual axis: Angle information + Acceleration x, y)

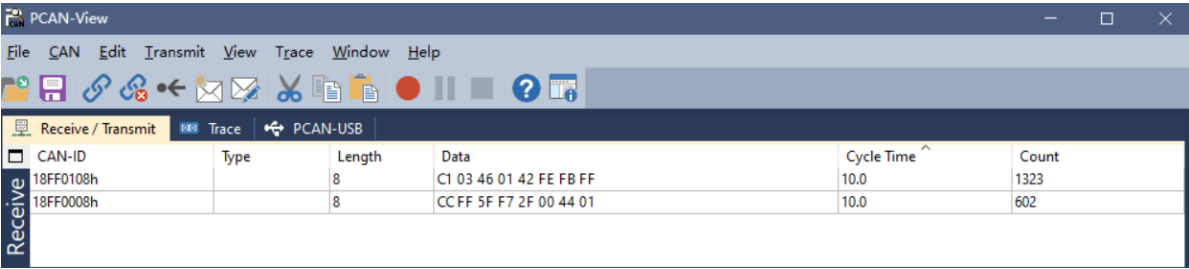
| SPN 名称 | SPN 位置(bit) | SPN 大小(bit) | 分辨率 | 偏移 | 范围 | 描述 |
|----------------|-------------|-------------|-------|----|----------------|---------------------------------|
| Pitch/X-Angle | 0-15 | 16 | 0.01° | - | -90 - 90° | Pitch Angle Data (±90° range) |
| Roll/Y-Angle | 16-31 | 16 | 0.01° | - | -180 - 180° | Roll Angle Data (±180° range) |
| Acceleration X | 32-47 | 16 | 1mg | - | -4000 - 4000mg | X-axis acceleration (±4g range) |
| Acceleration Y | 48-63 | 16 | 1mg | - | -4000 - 4000mg | Y-axis acceleration (±4g range) |

8.3.3 PGN65281(FF01)

加速度Z, 角速度 X Y Z(Acceleration z + Gyro x, y, z)

| SPN 名称 | SPN 位置 (bit) | SPN 大小 (bit) | 分辨率 | 偏移 | 范围 | 描述 |
|----------------|-----------------|-----------------|------------|----|-----------------------|---|
| Acceleration Z | 0-15 | 16 | 1mg | - | -4000 - 4000 mg | Z-axis acceleration (±4g range) |
| Gyro X | 16-31 | 16 | 0.1 °/s | - | -500,0 - 500,0 °/s | X-axis rate of rotation (±500°/s range) |
| Gyro Y | 32-47 | 16 | 0.1 °/s | - | -500,0 - 500,0 °/s | Y-axis rate of rotation (±500°/s range) |
| Gyro Z | 48-63 | 16 | 0.1 °/s | - | -500,0 - 500,0 °/s | Z-axis rate of rotation (±500°/s range) |

8.3.4 默认帧输出示例



| CAN-ID | Type | Length | Data | Cycle Time ^ | Count |
|-----------|------|--------|-------------------------|--------------|-------|
| 18FF0108h | | 8 | C1 03 46 01 42 FE FB FF | 10.0 | 1323 |
| 18FF0008h | | 8 | CC FF 5F F7 2F 00 44 01 | 10.0 | 602 |

模块默认发送PGN52810, PGN52811两条信息, 均为8字节远程拓展帧:

PGN52810: 0x18FF0008 CC FF 5F F7 2F 00 44 01

- Pitch = (int16_t)0xFFCC = -52 = -0.52°
- Roll = (int16_t)0xF75F = -2209 = -22.09°
- AccX = (int16_t)0x002F = 47mG
- AccY = (int16_t)0x0144 = 324mG

PGN52811: 0x18FF0108 C1 03 46 01 42 FE FB FF

- AccZ = (int16_t)0x03C1 = 961mG
- GyrX = (int16_t)0x0146 = 326 = 32.6°/s
- GyrY = (int16_t)0xFE42 = -446 = -44.6°/s
- GyrZ = (int16_t)0xFFFFB = -5 = -0.5°/s

可使用ZLG CAN_PRO 加载例程中提供的 .dbc 协议文件后直接查看DBC解析后的结果，如下：

| | | | | | | | | | | | | |
|--|--------|------|------------|-------------|----------|--------|-----|------|-----|-----|-------|------------------|
| FrameAnalyzer-C:\Users\14533\Desktop\gitResp\products_imu\examples\CAN\1939_PGNs.dbc | | | | | | | | | | | | |
| 加载协议 暂停 分类显示 自动滚屏 清空列表 导出 发送 协议类型: J1939 | | | | | | | | | | | | |
| 序号 | CAN... | 传输方向 | 时间 | 消息名 | ID(HEX) | PGN(H) | 源地址 | 目的地址 | 帧类型 | 帧格式 | 数据... | 帧数据(HEX) |
| 1 | 0 | 接收 | 159118.012 | TxPGN_65280 | 18FF0008 | 00FF00 | 08 | -- | 扩展帧 | 数据帧 | 08 | 4801DBFE3300YA00 |
| 信号名 实际值 描述 注释 原始值 起始位 位宽 变换比例 变换偏移 | | | | | | | | | | | | |
| Angle_X 3.28deg -- 148 0 16 0.01 0 | | | | | | | | | | | | |
| Angle_Y -2.93deg -- FEDB 16 16 0.01 0 | | | | | | | | | | | | |
| Accelerat 0.051G -- 33 32 16 0.001 0 | | | | | | | | | | | | |
| Accelerat 0.058G -- 3A 48 16 0.001 0 | | | | | | | | | | | | |
| 2 | 0 | 接收 | 159118.042 | TxPGN_65281 | 18FF0108 | 00FF01 | 08 | -- | 扩展帧 | 数据帧 | 08 | C503000000000000 |
| 信号名 实际值 描述 注释 原始值 起始位 位宽 变换比例 变换偏移 | | | | | | | | | | | | |
| Accelerat 0.965G -- 3C5 0 16 0.001 0 | | | | | | | | | | | | |
| Gyro_X 0deg/s -- 0 16 16 0.1 0 | | | | | | | | | | | | |
| Gyro_Y 0deg/s -- 0 32 16 0.1 0 | | | | | | | | | | | | |
| Gyro_Z 0deg/s -- 0 48 16 0.1 0 | | | | | | | | | | | | |

8.3.5 写入配置参数

| 29'b CAN拓展帧地址 | CAN 数据 | 描述 |
|---------------|--------------------------------------|---|
| 0x18EF08xx | 01 <u>00</u> xx xx xx xx xx xx | Index:0x01: Pitch 角度正负配置: 0:SAE J670(默认) 1:反方向 |
| 0x18EF08xx | 02 <u>00</u> xx xx xx xx xx xx | Index:0x02: Roll角度正负配置: 0:SAE J670(默认) 1:反方向 |
| 0x18EF08xx | 05 <u>0A</u> xx xx xx xx xx xx xx | Index:0x05: PGN52810 定时发送时间(单位为 ms, 范围10-200,默认:10) |
| 0x18EF08xx | 06 <u>0A</u> xx xx xx xx xx xx xx | Index:0x06: PGN52811定时发送时间(单位为 ms, 范围10-200,默认:10) |
| 0x18EF08xx | 07 <u>08</u> xx xx xx xx xx xx | Index:0x07: 地址(范围0-253, 默认:0x08) |
| 0x18EF08xx | 08 <u>05</u> xx xx xx xx xx xx | Index:0x08: 波特率, 0x03: 125Khz, 0x04:250Khz, 0x05:500Khz(默认) |
| 0x18EF08xx | 20 <u>00</u> xx xx xx xx xx xx | Index:0x20: 0:默认模式, 1:TILTIX AKS-180-F, 2:TILTIX AKS-180-E |
| 0x18EF08xx | FA 73 61 76 65 xx xx xx | Index:0xFA: 保存所有参数到Flash并重启 |
| 0x18EF08xx | FC 6C 6F 61 64 xx xx xx | Index:0xFA: 恢复所有参数到出厂配置并重启 |

8.3.6 读取配置参数

| 29'b CAN拓展帧地址 | CAN 数据 | 描述 |
|--------------------|-----------------------------------|---|
| 主机发送 0x18EA08xx | 01 EF 00 | 读取参数请求, Index:0x01 |
| 模块返回 0x18EFxx08 | 01 <u>00</u> FF FF FF FF FF FF | Index:0x01: Pitch 角度正负配置: 0:SAE J670(默认) 1:反方向 |
| 主机发送 0x18EA08xx | 02 EF 00 | 读取参数请求, Index:0x02 |
| 模块返回 0x18EFxx08 | 02 <u>00</u> FF FF FF FF FF FF | Index:0x02: Roll角度正负配置: 0:SAE J670(默认) 1:反方向 |
| 主机发送 0x18EA08xx | 05 EF 00 | 读取参数请求, Index:0x05 |
| 模块返回 0x18EFxx08 | 05 <u>0A</u> FF FF FF FF FF FF | Index:0x05: PGN52810 定时发送时间(单位为 ms, 范围10-200,默认:10) |
| 主机发送 0x18EA08xx | 06 EF 00 | 读取参数请求, Index:0x06 |
| 模块返回 0x18EFxx08 | 06 <u>0A</u> FF FF FF FF FF FF | Index:0x06: PGN52811定时发送时间(单位为 ms, 范围10-200,默认:10) |
| 主机发送 0x18EA08xx | 07 EF 00 | 读取参数请求, Index:0x07 |
| 模块返回 0x18EFxx08 | 07 <u>08</u> FF FF FF FF FF FF | Index:0x07: 地址(范围0-253, 默认:0x08) |
| 主机发送 0x18EA08xx | 08 EF 00 | 读取参数请求, Index:0x08 |
| 模块返回 0x18EFxx08 | 08 <u>05</u> FF FF FF FF FF FF | Index:0x08: 波特率, 0x03: 125Khz, 0x04:250Khz, 0x05:500Khz(默认) |
| 主机发送 0x18EA08xx | 20 EF 00 | 读取参数请求, Index:0x20 |
| 模块返回 0x18EFxx08 | 20 <u>00</u> FF FF FF FF FF FF | Index:0x20: 0:默认模式, 1:TILTIX AKS-180-F, 2:TILTIX AKS-180-E |

地址域中xx: J1939协议中的源地址, 可为任意字节。

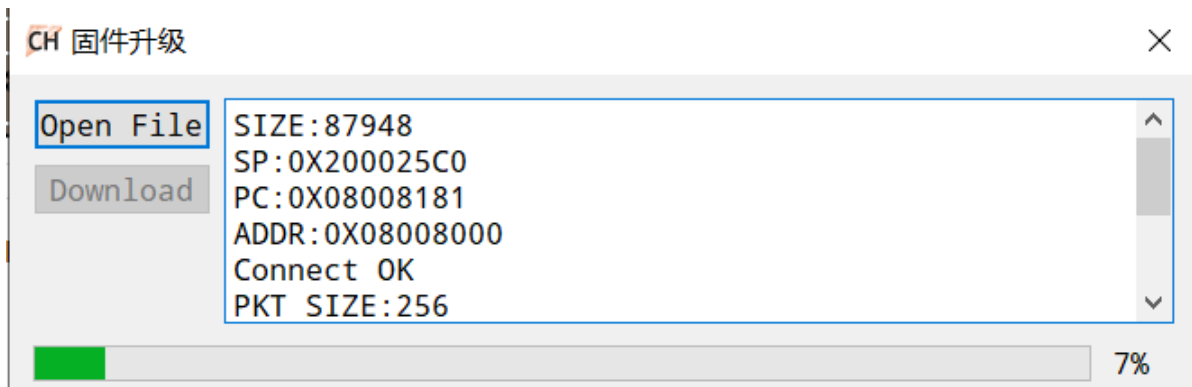
数据域中xx: 任意字节

9. 附录A - 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。



10. 附录B-技术支持

新产品信息及技术支持，请关注超核电子公众号和官方网站

