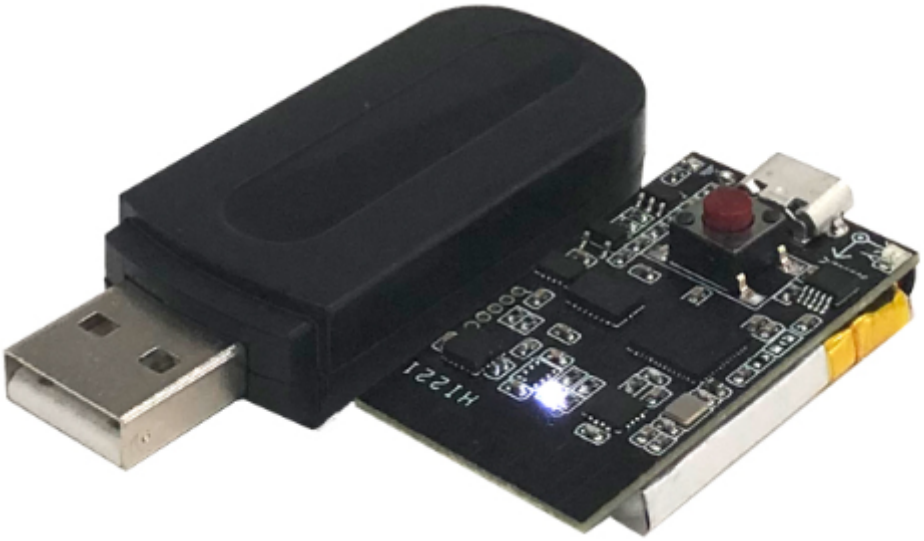
HI221/HI221GW 用户手册

HI221 无线传输模块及接收器, Rev 0.2

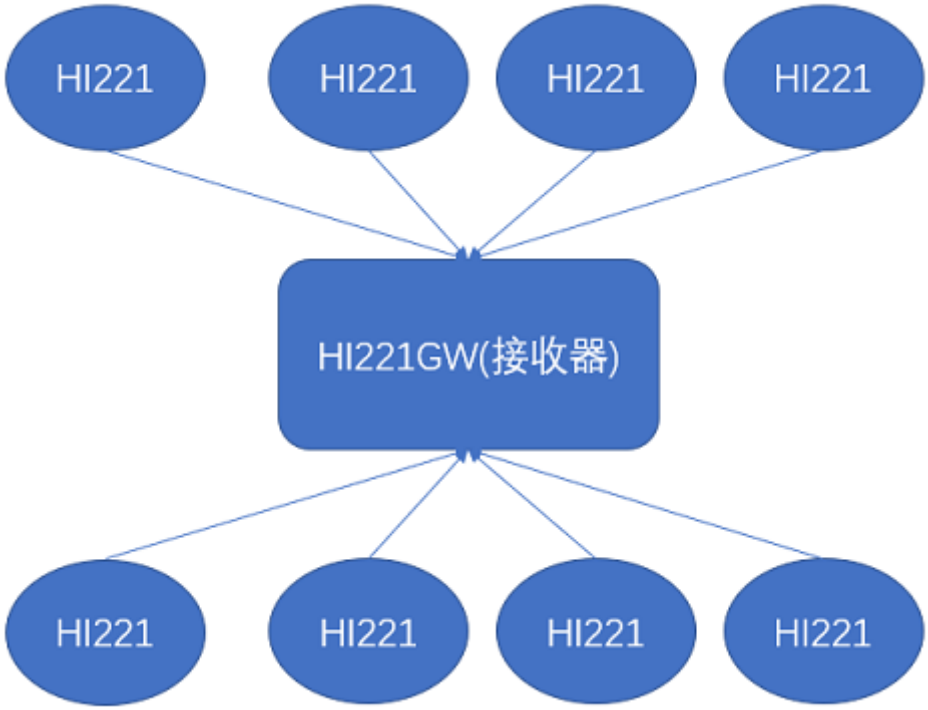


HI221/HI221GW 用户手册

- 简介
- 特性
  - 板载传感器
  - 数据处理
  - 通讯接口及供电
  - 其他
- 硬件及尺寸(节点)
  - 硬件参数
  - 安装建议
- 性能指标
  - 姿态角输出精度
  - 陀螺仪
  - 加速度计
  - 磁传感器参数
  - 模块数据接口参数(UART)
  - 模块数据接口参数(2.4G RF)
- 参考系定义
- 串口通讯协议
  - 数据包格式
  - 出厂默认数据包
  - 数据帧结构示例
- 通用AT指令
  - AT+ID
  - AT+GWID (HI221/HI221GW)
  - AT+URFR
  - AT+INFO
  - AT+ODR
  - AT+BAUD
  - AT+EOUT
  - AT+RST
  - AT+TRG
  - AT+SETPEL
  - AT+MODE
- 附录B - 四元数-欧拉角转换
  - 四元数基础
  - 四元数与旋转矩阵，欧拉角转换
    - 四元数->旋转矩阵
    - 四元数->欧拉角
    - 欧拉角->四元数
    - 欧拉角->旋转矩阵(n->b)
    - 旋转矩阵(n->b) 到欧拉角
  - 四元数基本操作C语言
  - 获得惯性坐标系(绝对坐标系)下线性加速度
- 附录C - 固件升级与恢复出厂设置

简介

H221/HI221GW是超核电子推出的一款低成本、高性能、小体积、低延时的惯性测量单元（IMU），本产品集成了三轴加速度计、三轴陀螺仪以及一款低功耗微处理器。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，包含横滚角、俯仰角以及以相对的航向角。同时也可以输出原始的传感器数据。HI221 由**HI221GW(接收机)** 和 **HI221(姿态模块)**组成。一个HI221GW和最多8个HI221模块组成星形网络结构。每个HI221可输出最高达100Hz的实时姿态数据。



特性

板载传感器

- 三轴陀螺仪, 最大量程: ±2000°/s 输出速率 2000Hz
- 三轴加速度计, 最大量程:±8g 输出速率 125Hz
- 三轴地磁场传感器，最大量程: 800mG 内部采样率 100Hz

数据处理

- 加速度出厂前经过校准
- 数据融合算法计算并输出地理坐标系下的旋转四元数及欧拉角

通讯接口及供电

- 串口(兼容TTL 可直接与5V 或3.3V 串口设备连接)
- 供电电压：3.3 (+/- 100 mV)
- 最大峰值功耗：120mA(RF Tx发射)

其他

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

硬件及尺寸(节点)

硬件参数

参数	描述

参数	描述
输出数据接口	UART(TTL 1.8V - 3.3V) 或者 2.4RF Radio
工作电压	3.3V (± 100mV)
功耗	396mW @3.3V
温度范围	-20℃ - 85 ℃
最大线性加速度	0 - 115 m/s^2
尺寸	20 x 38 x 8.5mm (W x L x H)
板载传感器	三轴加速度计 三轴陀螺仪 三轴地磁场传感器

安装建议

由于传感器制造工艺的原因，XY 和 Z 轴性能有略微差异，安装时建议将模块 Z 轴与重力方向保持平行(既水平安装)。尽量让距离模块10cm 内不能有铁质外壳，小功率电机等磁性器件。

性能指标

姿态角输出精度

姿态角	典型值	最大值
横滚角\俯仰角 - 静态	0.2°	0.4°
横滚角\俯仰角 - 动态	0.5°	2.0°
航向角	-	-

陀螺仪

参数	值
测量范围	±2000°/s
非线性度	±0.1% (25°最佳)
噪声密度	0.08°/s/$\sqrt{Hz}$
采样率	2000Hz

加速度计

参数	值
测量范围	±8G
非线性度	±0.5% (25°最佳)
最大零点偏移	10mG(校准后)
噪声密度	250 $uG\sqrt{Hz}$
采样率	125Hz

磁传感器参数

参数	值
测量范围	±8Gauss
非线性度	±0.1%
采样率	100Hz

模块数据接口参数(UART)

参数	值

参数	值
串口输出波特率	4800/9600/115200/460800可选
帧输出速率	1 - 400Hz

模块数据接口参数(2.4G RF)

参数	值
空中波特率	1Mbps
帧输出速率	100Hz
接收器最大连接模块数	8

参考系定义

本产品采用右手(RH, Right-Hand)坐标系。输出的四元数及欧拉角为 传感器坐标系 到 惯性坐标系(世界坐标系) 的旋转。其中欧拉角旋转顺序为 ZYX(也称 321)旋转顺序，欧拉角具体定义如下：

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

本产品使用北西天(North-West-Up NWU) 坐标系统，即视为模块的地理坐标系(世界坐标系)定义如下：

- X 轴正方向指向北
- Y 轴正方向指向西
- Z 轴正方向指向天

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

串口通讯协议

数据包格式

模块资料包中提供了C 和C#的数据解析函数以供参考。模块上电后，模块默认按100Hz (出厂默认输出速率) 输出数据包，数据包格式如下：

域	同步帧头	帧类型	帧长度	CRC16	帧携带数据
名称	PRE	TYPE	LEN	CRC	ID(N) + DATA(N)
大小(byte)	0	1	2	2	可变(1-64)
偏移(byte)	0	1	2	4	6
值(hex)	0x5A	0xA5	帧长度	CRC校验码	具体意义参看下节
类型	uint8_t	uint8_t	uint16_t	uint16_t	-

- PRE 固定为0x5A
- TYPE 固定为0xA5 代表数据帧
- LEN 帧中数据域的长度。一帧最大为256 字节LSB(低字节在前)，长度只是值真正数据的长度，不包含`PRE`,`TYPE`,`LEN`,`CRC` 字段。
- CRC 除**CRC 本身外**其余所有帧数据的16 位CRC 校验和LSB[^LSB]。CRC实现函数：

```
/*
    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;
}
```

- ID和DATA 一帧数据可由多个数据包组成，每个数据包包含ID 和DATA 两部分。ID 标识该数据包的类型及长 度，DATA 为数据包数据内容。模块支持的数据包如下:

数据包ID	数据域长度(字节)	名称	单位
0x90	1	用户ID	无
0xA0	6	加速度	0.001G [1]
0xA5	6	线性加速度	0.001G
0xB0	6	角速度	0.1°/s
0xC0	6	磁场强度	0.001Gauss
0xD0	6	欧拉角 (整形输出)	度
0xD9	12	欧拉角(浮点输出)	度
0xD1	16	四元数	N/A
0xF0	4	气压	Pa
0x71	128	无线节点四元数集合	无
0x72	48	无线节点欧拉角集合	同0xD0
0x75	48	无线节点加速度集合	同0xA0
0x78	48	无线节点角速度集合	同0xB0
0x61	3	数据帧拓展信息	N/A

- 0x90 用户ID
- 0xA0 加速度，格式为int16，共三个轴，每个轴占2 个字节，X、Y、Z 三轴共6 个字节，LSB。传感器输出的原始加速度
- 0xA5 性加速度，格式为int16，共三个轴，每个轴占2 个字节，X、Y、Z 三轴共6 个字节，LSB。地理坐标系下去除重力分量的加速度值
- 0xB0 角速度，格式为int16，共三个轴，每个轴占2 个字节，X、Y、Z 三轴共6 个字节，LSB。传感器输出的角速度

- 0xC0 磁场强度，格式为int16，共三个轴，每个轴占2 个字节，X、Y、Z 三轴共6 个字节，LSB。传感器输出的磁场强度
- 0xD0 欧拉角整形格式，格式为int16，共三个轴，每个轴占2 个字节，顺序为Pitch/Roll/Yaw。LSB。接收到Roll, Pitch 为物理值乘以100 后得到的数值，Yaw 为乘以10 得到的数值举例：当接收到的Yaw = 100 时，表示航向角为10°
- 0xD9 浮点格式输出的欧拉角。格式为float，共3 个值(Pitch/Roll/Yaw)，每个值占4 字节(float 型单精度 浮点数)，LSB。
- 0XD1 四元数，格式为float，共4 个值(W X Y Z)，每个四元数占4 字节(单精度浮点数)，LSB。
- 0XF0 气压。(只针对有气压传感器的产品)
- 0x71 (HI221GW接收机专用)节点四元数集合. 所有节点的四元数， 每个节点16字节，从0到最后一个节点顺序排列。每个节点 4个浮点数，分别为W X Y Z, 每个数用float 型表示，每个float 4字节。float为LSB
- 0x72 (HI221GW接收机专用)节点欧拉集合. 所有节点的欧拉角， 每个节点6字节，从0到最后一个节点顺序排列。每个节点欧拉为角整形格式，格式为int16，共三个轴，每个轴占2 个字节，顺序为Pitch/Roll/Yaw。LSB。接收到Roll, Pitch 为物理值乘以100 后得到的数值，Yaw 为乘以10 得到的数值举例：当接收到的Yaw = 100 时，表示航向角为10°
- 0x75 (HI221GW接收机专用)节点加速度集合. 每个节点6字节，从0到最后一个节点顺序排列。每个节点3个int16_t 型数据。分别为X Y Z的加速度。每个int16_t 占2字节， LSB
- 0x78 (HI221GW接收机专用)节点角速度集合. 每个节点6字节，从0到最后一个节点顺序排列。每个节点3个int16_t 型数据。分别为X Y Z的角速度。每个int16_t 占2字节， LSB
- 0x61

  (HI221GW接收机专用)数据帧拓展信息，共3个字节:

数据帧拓展信息字节偏移	值	说明
0	0x00	保留
1	GWID	接收机GWID
2	CNT	此帧包含无线节点数: 1-16

出厂默认数据包

出厂默认一帧中携带数据包定义如下：

HI219/HI22x:

顺序	数据包	说明
1	0x90	用户ID
2	0xA0	加速度
3	0xB0	角速度
4	0xC0	磁场强度
5	0xD0	欧拉角(整形输出)
6	0xF0	气压

HI216：

顺序	数据包	说明
1	0xA0	加速度
2	0xB0	角速度
3	0xD0	欧拉角(整形输出)

HI221GW(无线节点接收机):

顺序	数據包	说明
1	0x71	四元数
2	0x75	加速度

数据帧结构示例

假设输出的数据帧带有 A0,B0,D0 数据包，使用串口助手采样一帧数据如下所示:

5A A5 15 00 A9 8B A0 EA FF D0 03 45 FF B0 00 00 00 00 00 00 D0 87 00 6F 27 F5 FF

其中：

5A A5帧头

15 00帧数据域长度：(0x00<<8) + 0x15 = 21

A9 8B帧CRC校验值: (0x8B<<8) + 0xA9 = 0x8BA9

A0 EA FF D0 03 45 FF 加速度数据包, A0为加速度数据包ID， 三轴加速度为：

AccX = (int16_t)((0xFF<<8)+ 0xEA) = -22 AccY = (int16_t)((0x03<<8)+ 0xD0) = 976 AccZ = (int16_t)((0xFF<<8)+ 0x45) = -187

B0 00 00 00 00 00 00 角速度数据包, B0为数据包ID， 三轴角速度全为0

D0 87 00 6F 27 F5 FF 欧拉角数据包, D0为数据包ID

Pitch= (int16_t)((0x00<<8)+ 0x87) / 100 = 1.35°

Roll= (int16_t)((0x27<<8)+ 0x6F) / 100 = 100.95°

Yaw = (int16_t)((0xFF<<8)+ 0xF5) / 10 = -1.1°

计算CRC校验值：

记上面接收到的一帧数据存为C语言uint8_t 数组 buf:

```
    uint16_t payload_len;
    uint16_t crc;

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

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

    /* calulate payload crc */
    crc16_update(&crc, buf + 6, payload_len);
```

最后计算得 CRC值为 0x8BA9, 与帧携带CRC值相同，帧校验正确。

通用AT指令

模块采用AT 指令集配置/查看模块参数。AT 指令总以ASCII 码AT 开头，后面跟控制字符，最 后以回车换行\r\n结束。可使用串口调试助手进行测试：

通用模块 AT指令如下

指令	功能	是否掉电保存
AT+ID	设置模块用户ID	Y
AT+GWID	设置无线网关ID(针对于无线产品)	Y
AT+URFR	旋转模块传感器坐标系	Y
AT+INFO	打印模块信息	N
AT+ODR	设置模块串口输出帧频率	Y
AT+BAUD	设置串口波特率	Y
AT+EOUT	数据输出开关	N
AT+RST	复位模块	N
AT+TRG	单次输出触发	N
AT+SETPEL	设置输出数据包	Y
AT+MODE	设置模块工作模式	Y

AT+ID

设置模块用户ID

例 `AT+ID=1`

AT+GWID (HI221/HI221GW)

HI221GW(接收机) 和 HI221(节点) 拥有GWID属性，可通过AT+GWID指令配置，GWID属性决定了接收器和节点的RF频率，只有HI221模块的GWID 和 接收器的 GWID属性相同时，模块和接收器直接才能通讯。GWID相当于无线网段，当在同一地点使用多个接收机组成多个星形网络时，必须保证每个接收器的GWID(网段)不同。

例 将一个接收器设置为GWID=3， 并将3个模块的自身ID设置为 0,1,2 并连接到这个接收器上：

接收机配置：`AT+GWID=3`

节点0配置： `AT+GWID=3` `AT+ID=0`

节点1配置： `AT+GWID=3` `AT+ID=1`

节点2配置： `AT+GWID=3` `AT+ID=2`

AT+URFR

某些情况下传感器需要倾斜垂直安装，这时候需要旋转传感器坐标系，这条指令提供了旋转传感器坐标系的接口：

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

其中 $\mathbb{C}_{nn}$ 支持浮点数

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

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， $\cdot \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B$ 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°， 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°， 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Y轴 旋转 90°， 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°， 输入命令：`AT+URFR= 0,0,1,0,1,0,-1,0,0`
- 恢复默认值：`AT+URFR=1,0,0,0,1,0,0,0,1`

AT+INFO

打印模块信息，包括产品型号，版本，固件发布日期等。AT+INFO可以拓展二级指令实现更多信息的查询

INFO二级拓展指令	功能	示例
CAL	显示模块内部校准参数	AT+INFO=CAL
RF	显示无线设备参数	AT+INFO=RF
VER	显示详细版本信息	AT+INFO=VER

AT+ODR

设置模块串口输出速率。 掉电保存，复位模块生效

例 设置串口输出速率为100Hz: `AT+ODR=100`

AT+BAUD

设置串口波特率，可选值：4800/9600/115200/256000/460800`

例 `AT+BAUD=115200`

!!! note "注意"

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

AT+EOUT

串口输出开关

例 打开串口输出 `AT+EOUT=1` 关闭串口输出 `AT+EOUT=0`

AT+RST

复位模块

例 `AT+RST`

AT+TRG

触发模块输出一帧数据，可以配合AT+ODR=0来实现单次触发输出。

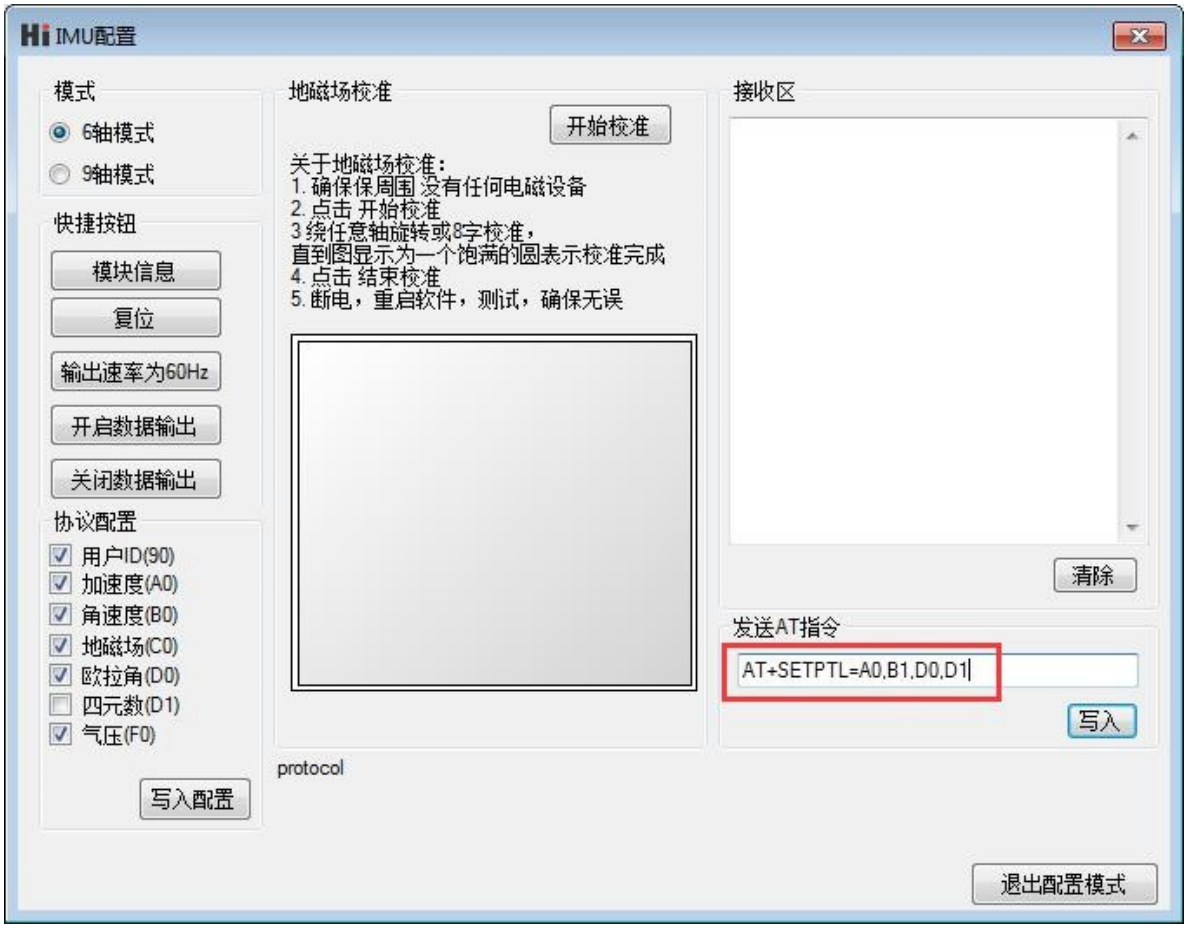
例 `AT+TRG`

AT+SETPEL

设置输出协议:

模块数据帧中的数据包组成可使用AT指令配置，格式为`AT+SETPTL=<ITEM_ID>,<ITEM_ID>...`一帧输出可包含最多8个数据包。

例 配置模块输出加速度，角速度,整形格式欧拉角和四元数的指令为：`AT+SETPTL=A0,B1,D0,D1`



AT+MODE

设置模块工作模式

例

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

附录B - 四元数-欧拉角转换

四元数基础

四元数是一个四维空间上的一点，使用一个实数和三个虚数来代表：$q \in \mathbb{R}^4 = \mathbb{H}$

四元数有如下几种常用的表示方法：

复数表示	向量表示	四元数表示法1	四元数表示法2
$q = q_0 + \mathrm{i}q_1 + \mathrm{j}q_2 + \mathrm{k}q_3$	$q = [q_0, \mathbf{q}] = \left[q_0, \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}\right]$	$q = [q_0, q_1, q_2, q_3]$	$q = [q_w, q_x, q_y, q_z]$

其中：

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1$$

$$\mathrm{ij} = \mathrm{k} = -\mathrm{ji}, \quad \mathrm{jk} = \mathrm{i} = -\mathrm{kj}, \quad \mathrm{ki} = \mathrm{j} = -\mathrm{ik}$$

四元数乘法：

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix}$$

一个单位四元数总是可以表示为这种这种形式：$q_R(\alpha, \mathbf{u}) = \left[\cos\frac{\alpha}{2}, \sin\frac{\alpha}{2} \cdot \mathbf{u}\right]$

其中α是旋转角度，$\mathbf{u} \in \mathbb{R}^3$ 为旋转轴，且$\|\mathbf{u}\| = 1$.

四元数与旋转矩阵，欧拉角转换

四元数->旋转矩阵

(其中四元数是 b->n, R为 n->b)

$$R_n^b = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

四元数->欧拉角

旋转矩阵，四元数和欧拉角是表示旋转的三种常用方式，其中另外两种表示形式转换为欧拉角时，必须先指定欧拉角旋转顺序。本产品使用"ZYX"旋转顺序,即先旋转航向角，然后俯仰角，最后横滚角：

转换公式为：

$$\begin{bmatrix} \phi(\text{横滚}) \\ \theta(\text{俯仰}) \\ \psi(\text{航向}) \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(2q_2 q_3 + 2q_0 q_1, q_3^2 - q_2^2 - q_1^2 + q_0^2\right) \\ -\operatorname{asin}(2q_1 q_3 - 2q_0 q_2) \\ \operatorname{atan2}\left(2q_1 q_2 + 2q_0 q_3), q_1^2 + q_0^2 - q_3^2 - q_2^2\right) \end{bmatrix}$$

欧拉角->四元数

记$s_\phi = \sin\frac{\phi}{2}, c_\phi = \cos\frac{\phi}{2}$,以此类推：

$$\mathbf{q} = \begin{bmatrix} c_{\phi/2} c_{\theta/2} c_{\psi/2} + s_{\phi/2} s_{\theta/2} s_{\psi/2} \\ -c_{\phi/2} s_{\theta/2} s_{\psi/2} + c_{\theta/2} c_{\psi/2} s_{\phi/2} \\ c_{\phi/2} c_{\psi/2} s_{\theta/2} + s_{\phi/2} c_{\theta/2} s_{\psi/2} \\ c_{\phi/2} c_{\theta/2} s_{\psi/2} - s_{\phi/2} c_{\psi/2} s_{\theta/2} \end{bmatrix}$$

欧拉角->旋转矩阵(n->b)

$$R_n^b = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\theta s_\phi \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\theta c_\phi \end{bmatrix}$$

旋转矩阵(n->b) 到欧拉角

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(r_{23}, r_{33}\right) \\ -\operatorname{asin}(r_{13}) \\ \operatorname{atan2}\left(r_{12}, r_{11}\right) \end{bmatrix}$$

四元数转换欧拉角的C函数如下：

```
1
2   /* eular angle: e[0]:Roll e[1]:Pitch e[2]:Yaw */
3   void quat2eul(float32_t *q, float32_t *e, const char *seq)
4   {
5       float xz = q[1] * q[3];
6       float wy = q[0] * q[2];
7       float yz = q[2] * q[3];
8       float wx = q[0] * q[1];
9       float wz = q[0] * q[3];
10      float xy = q[1] * q[2];
11      float ww = q[0] * q[0];
12      float xx = q[1] * q[1];
13      float yy = q[2] * q[2];
14      float zz = q[3] * q[3];
15
16      if(!strcmp(seq, "ZYX") || !strcmp(seq, "321"))
17      {
18          e[0] = (float)atan2f(2.0F*(yz + wx), ww - xx - yy + zz);
19          e[1] = (float)asinf(2*(wy -xz));
20          e[2] = (float)atan2f(2.0F*(wz + xy), ww + xx  - yy - zz);
21      }
22  }
23
```

调用

```
1   quat2eul(q, eul, "321");
```

四元数基本操作C语言

```
1  /* 平方 */
2  #define PF(a)   ((a) * (a))
3
4  /* 四元数取共轭(逆) b = conj(a) */
5  void qconj(float32_t *a, float32_t *b)
6  {
7      b[0] =  a[0];
8      b[1] = -a[1];
9      b[2] = -a[2];
10     b[3] = -a[3];
11 }
12
13 /* 四元数旋转向量 input: a, output b, q: w,x,y,z */
14 void qrotate(float32_t *q, float32_t *a, float32_t *b)
15 {
16     b[0] = a[0]*(1 - 2*PF(q[2]) - 2*PF(q[3])) + a[1]*2*(q[1]*q[2] + q[0]*q[3])
   + a[2]*2*(q[1]*q[3] - q[0]*q[2]);
```

```
    b[1] = a[0]*2*(q[1]*q[2] - q[0]*q[3])     + a[1]*(1 - 2*PF(q[1]) - 2*PF(q[3])) + a[2]*2*(q[2]*q[3] + q[0]*q[1]);
    b[2] = a[0]*2*(q[1]*q[3] + q[0]*q[2])     + a[1]*2*(q[2]*q[3] - q[0]*q[1])   + a[2]*(1 - 2*PF(q[1]) - 2*PF(q[2]));
}

/* 向量减法运算： A <- A - B */
void vsub2(float32_t *a, float32_t *b, uint32_t len)
{
    int i;
    for(i=0; i<len; i++)
    {
        a[i] = a[i] - b[i];
    }
}
```

获得惯性坐标系(绝对坐标系)下线性加速度

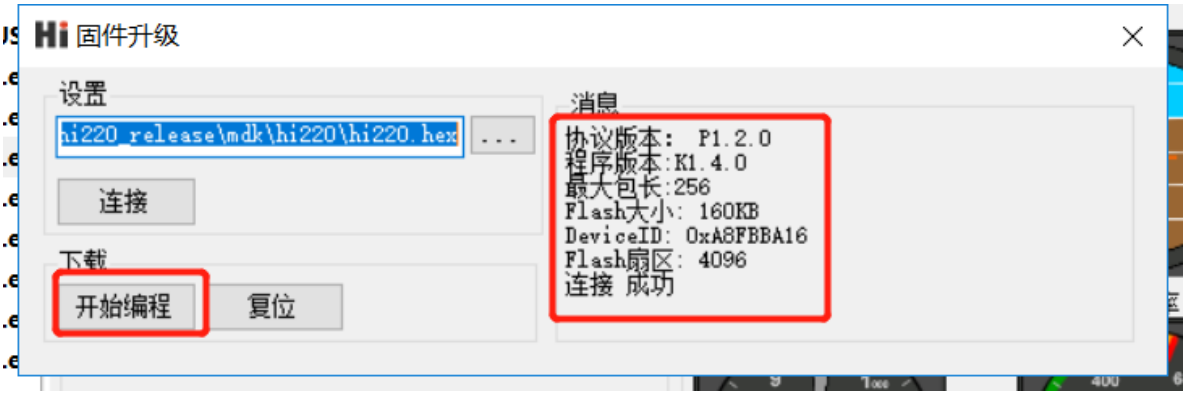
基本思路：使用姿态四元数将 传感器坐标系下的加速度 转换到惯性系下，再减去惯性系下重力加速度(0,0,1)即可:

```
/*
    获得世界系下线性加速度： q:姿态四元数,
    input:acc_body:传感器下加速度(加计读值)
    output: lin_acceleration 世界系下线性加速度
*/
void get_linear_acceleration(float32_t *q, float32_t *acc_body, float32_t *lin_acceleration)
{
    /* gravity in N frame */
    float gN[3];
    float qBN[4];
    gN[0] = 0;
    gN[1] = 0;
    gN[2] = 1;

    /* q must be N->B */
    qconj(q, qBN);

    qrotate(qBN, acc_body, lin_acceleration);

    vsub2(lin_acceleration, gN, 3);
}
```

附录C - 固件升级与恢复出厂设置

本产品支持在线升级固件，请关注超核电子官网www.hipnuc.com 来获取最新固件版本 固件升级步骤:

- 获取最新的固件程序。拓展名为.hex
- 连接模块，打开上位机，将模块和上位机波特率都设置为115200.切换到固件升级窗口
- 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xxx.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， 重新上电，模块升级完成。



---

1. 1G = 1x当地重力加速度↩