

HI221/HI221GW 用户手册

HI221 无线传输模块及接收器, Rev 1.0



HI221/HI221GW 用户手册

简介

特性

板载传感器

通讯接口及供电

硬件及尺寸(节点)

硬件参数

坐标系定义

性能指标

姿态角输出精度

陀螺仪

加速度计

磁传感器参数

模块数据接口参数(UART)

模块数据接口参数(2.4G RF)

传感器校准

地磁校准

磁干扰分类

6轴和9轴模式区别

校准方法

串口通讯协议

数据包

数据包总览

产品支持数据包列表

0X91(IMUSOL)

0x62(GWSOL)

出厂默认数据包

数据帧结构示例

数据帧配置为 0x91 数据包

数据帧配置为 0x62 数据包

AT指令

AT+ID

AT+INFO

AT+ODR

AT+BAUD

AT+EQUT

AT+RST

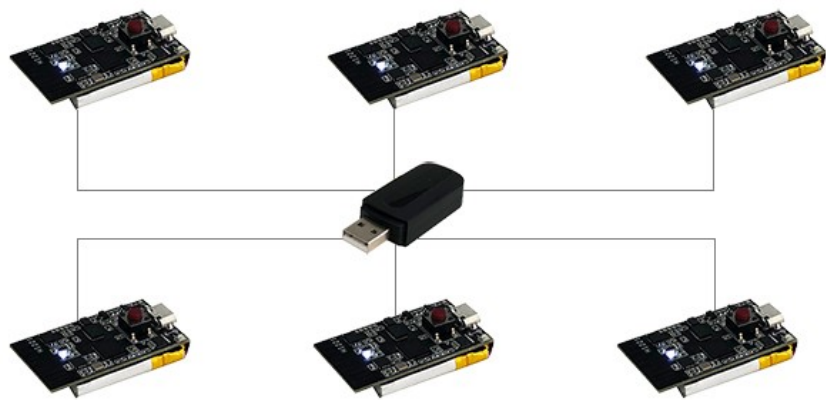
AT+GWID

AT+MODE

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16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: $\pm 2000^{\circ}/s$ 输出速率 2000Hz
- 三轴加速度计, 最大量程: $\pm 8g$ 输出速率 125Hz
- 三轴地磁场传感器, 最大量程: 800mG 内部采样率 100Hz

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)

硬件及尺寸(节点)

硬件参数

参数	值
输出数据接口	UART(TTL 1.8V - 3.3V) 或者 2.4RF Radio
工作电压	3.3V ($\pm 100mV$)
工作电流	30mA
待机电流	20uA
充电后平均工作时长	8h
温度范围	-20°C - 85 °C
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

坐标系定义

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

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

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

性能指标

姿态角输出精度

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

陀螺仪

参数	值
测量范围	±2000°/s
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	30mG
噪声密度	250 $\mu G\sqrt{Hz}$
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
串口输出波特率	115200/460800/921600可选
帧输出速率	1/25/50/100Hz 可选

模块数据接口参数(2.4G RF)

参数	值
空中波特率	2Mbps
帧速率(每个节点均可达到)	100Hz
接收器最大连接节点数	16

传感器校准

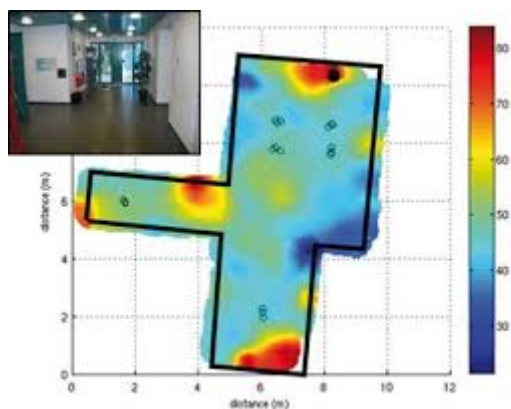
1. 每一个姿态传感器都单独进行过全测量范围内的校准和测试。陀螺和加速度计的非正交和刻度因子误差参数都会保存在模块内部的Flash中。
2. 陀螺仪自动校准需要在上电后静止模块5s 左右以获得最好的校准效果。如上电5s内模块处于非静止状态或者缓慢转动状态则陀螺性能严重下降。

地磁校准

磁干扰分类

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

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



注意

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

6轴和9轴模式区别

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

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

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

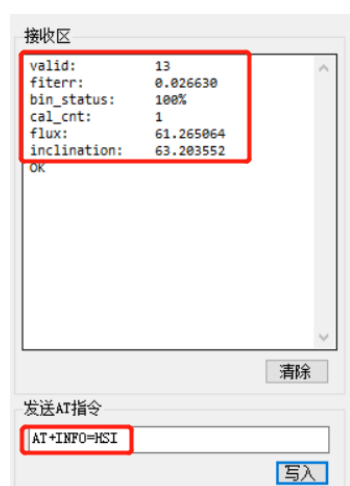
校准方法

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

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

地磁校准状态可以使用AT指令来查看：

发送AT+INFO=HSI 指令，模块会打印当前地磁校准系统状态：



接收区

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

清除

发送AT指令

AT+INFO=HSI

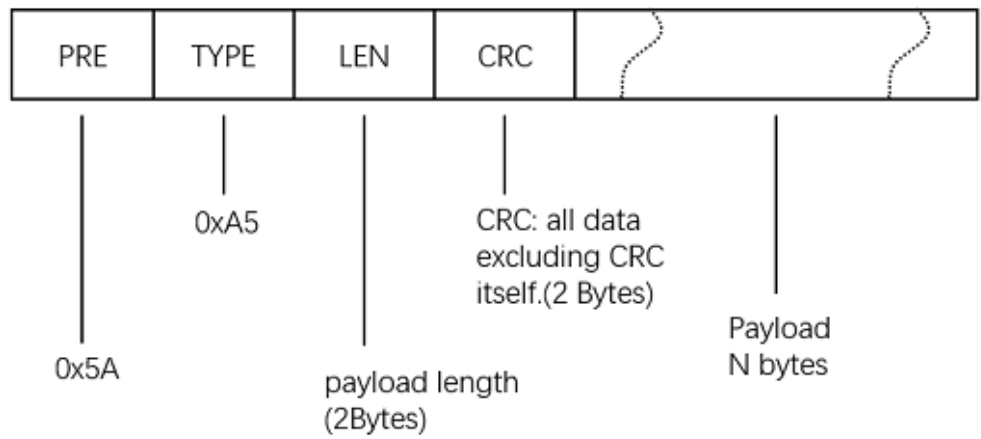
写入

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

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

串口通讯协议

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



其中:

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

CRC实现函数：

```
1  /*
2      currentCrc: previous crc value, set 0 if it's first section
3      src: source stream data
4      lengthInBytes: length
5  */
6  static void crc16_update(uint16_t *currentCrc, const uint8_t *src,
7                          uint32_t lengthInBytes)
8  {
9      uint32_t crc = *currentCrc;
10     uint32_t j;
11     for (j=0; j < lengthInBytes; ++j)
12     {
13         uint32_t i;
14         uint32_t byte = src[j];
15         crc ^= byte << 8;
16         for (i = 0; i < 8; ++i)
17         {
18             uint32_t temp = crc << 1;
19             if (crc & 0x8000)
20             {
21                 temp ^= 0x1021;
22             }
23             crc = temp;
24         }
25     }
26     *currentCrc = crc;
27 }
```

数据包

数据包总览

数据包标签 (DATA_ID)	数据包长度(包含标签 1字节)	名称	备注
0x91	76	IMUSOL(IMU数据集 合)	HI221输出数据 包
0x62	变长	GWSOL(无线节点数 据集合)	HI221GW输出 数据包

产品支持数据包列表

下表列出所有产品支持的数据包,* 表示支持 -表示不支持

型号	91	62
HI221	*	-
HI221GW	*	*

0X91(IMUSOL)

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

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

0x62(GWSOL)

新版本无线接收机支持此数据包。数据包前8个字节为接收机信息。后面分为N个数据块。每个数据块描述一个节点的姿态数据(最大支持16个节点)。每个数据块大小为76字节，数据结构同0x91。

此协议包数据量较大，建议将波特率调整至921600以获得最高的帧率输出。格式如下：

字节偏移	大小	类型	单位	说明
0	1	uint8_t	-	数据包标签:0x62
1	1	uint8_t	-	GWID,接收机网络ID
2	1	uint8_t	-	N,此帧包含节点数据块个数
3	5	-	-	保留
---- 节点数据块开始----	-	-	-	数据结构同0x91
8+76*N(N=0-15)	1	uint8_t	-	数据包标签:0x91
9+76*N(N=0-15)	1	uint8_t	-	节点N的ID
10+76*N	10	-	-	保留
20+76*N	12	float	-	节点N三轴加速度
32+76*N	12	float	-	节点N三轴角速度
44+76*N	12	float	-	节点N轴磁场强度
56+76*N	12	float	-	节点N欧拉角
68+76*N	16	float	-	节点N四元数
---- 节点数据块结束----	-	-	-	-----

出厂默认数据包

出厂默认一帧中携带数据包数据定义如下：

产品	默认输出数据包
HI221	91
HI221GW	62

数据帧结构示例

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

得到CRC值为0x516C. 帧CRC校验通过。

- 第三步：接收数据

从0x91开始为数据包的数据域。在C语言中可以定义结构体来方便的读取数据：

定义0x91数据包结构体如下：

```
1  __packed typedef struct
2  {
3      uint8_t      tag;                /* data packet tag */
4      uint8_t      id;
5      uint8_t      rev[6];            /* reserved */
6      uint32_t      ts;                /* timestamp */
7      float         acc[3];
8      float         gyr[3];
9      float         mag[3];
10     float         eul[3];            /* euler angles: Roll,Pitch,Yaw */
11     float         quat[4];           /* quaternion */
12 }id0x91_t;
```

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

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

最后得到dat数据结果:

1	id	:	0				
2	timestamp	:	310205				
3	acc	:	0.224	0.770	0.691		
4	gyr	:	-54.708	-20.077	-119.070		
5	mag	:	19.183	-26.208	-34.542		
6	eul(R/P/Y)	:	48.720	-21.014	-45.512		
7	quat	:	0.855	0.310	-0.310	-0.277	

数据帧配置为 0x62 数据包

使用串口助手采样一帧数据,共166字节,前6字节为帧头,长度和CRC校验值。剩余160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 A0 00 B5 DC 62 00 02 00 00 00 00 00 91 01 00 00 00 00 00 00 00 00 00 00 00 00 00 EC 51 B8
BD 12 83 40 3E 64 3B 8F 3F 33 33 E7 C2 CD CC 3C C1 33 33 0B C1 9A 99 99 41 9A 99 A9
40 CD CC 94 C1 80 D3 28 41 E8 CD 9B 40 B1 45 2C C0 78 97 7E 3F 83 20 BE 3D 14 5C 24
3D 32 69 DF BC 91 04 00 00 00 00 00 00 00 00 00 00 00 00 CD CC CC 3E 6F 12 83 3C 56 0E 6D
3F 9A 99 C7 C2 33 33 63 41 9A 99 35 42 00 00 94 41 33 33 63 41 00 00 D0 C1 F6 E6 A1 40
C0 93 A6 C1 FB 73 F5 C0 00 1D 7B 3F C8 E7 FF 3C DA 6C 3B BE 5E 99 6C BD

- 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:A0 00: (0x00<<8) + 0xA0 = 160

帧CRC校验值:B5 DC:(0xDC<<8) + 0xB5 = 0xDCB5

- 第二步: 校验CRC,过程同0x91数据包

计算得到CRC值为0xDCB5. 帧CRC校验通过。

- 第三步: 获得0x62帧网段, 节点数等信息

```
1  uint8_t gwsol_tag = buf[6];
2  uint8_t gwid = buf[7]; /* GWID网段 */
3  uint8_t node_cnt = buf[8]; /* 接收到的节点数 */
```

GWSOL标签: 0x62 = 0x62

接收机网段: 0x00 = 0

本帧节点数: 0x02 = 2

- 第四步: 接收每个节点数据

0x62数据包实际上是多个0x91数据包拼在一起, 头部再加接收机消息组成。在C语言中可以定义0x91结构体来方便的读取数据:

定义0x91数据包结构体如下: (以Keil为例)

```

1  __packed typedef struct
2  {
3      uint8_t    tag;                /* data packet tag */
4      uint8_t    id;
5      uint8_t    rev[6];            /* reserved */
6      uint32_t    ts;                /* timestamp */
7      float      acc[3];
8      float      gyr[3];
9      float      mag[3];
10     float      eul[3];            /* eular angles:R/P/Y */
11     float      quat[4];           /* quaternion */
12 }id0x91_t;
13

```

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

```

1  /* 接收数据并使用0x91数据包结构定义来解释数据 */
2  __align(4) id0x91_t id0x91;    /* struct must be 4 byte aligned */
3
4  int i;
5  for(i=0; i<node_cnt; i++)
6  {
7      memcpy(&id0x91, &buf[8+6] + i*sizeof(id0x91_t),
8      sizeof(id0x91_t)); /* 8+6: 6为帧头信息长度, 8为0x62数据包中的接收机包头长度 */
9
10     /*处理该包数据
11     ...
12     printf("node_id:%d\r\n", id0x91->id);
13     */
14 }

```

最后得到数据结果: 这帧数据包包含2个节点，ID分别为1和4，具体信息如下：

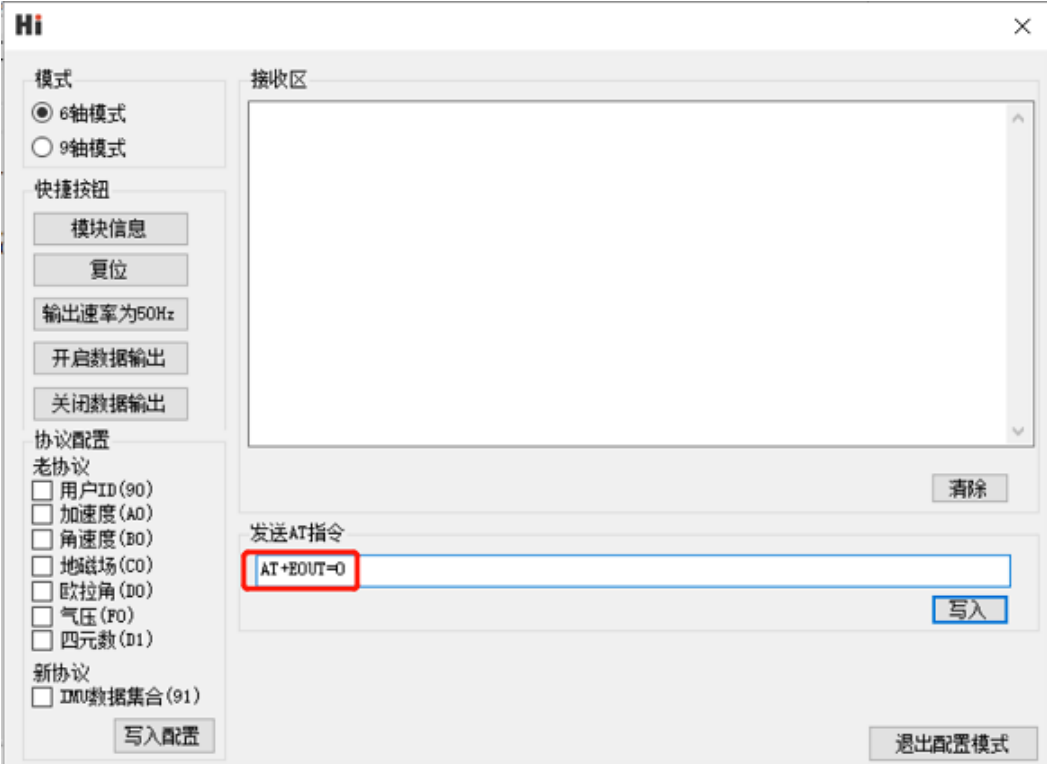
```

1  id          : 1
2  timestamp   : 0
3  acc         :  -0.090    0.188    1.119
4  gyr         : -115.600  -11.800   -8.700
5  mag         :   19.200    5.300  -18.600
6  eul (R/P/Y) :   10.552    4.869   -2.692
7  quat        :    0.994    0.093    0.040   -0.027
8
9  id          : 4
10 timestamp    : 0
11 acc          :    0.400    0.016    0.926
12 gyr          :  -99.800   14.200   45.400
13 mag          :   18.500   14.200  -26.000
14 eul (R/P/Y)  :    5.059  -20.822   -7.670
15 quat         :    0.981    0.031   -0.183   -0.058

```

AT指令

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



通用模块 AT指令如下

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

AT+ID

设置模块用户ID

例 AT+ID=1

AT+INFO

打印模块信息，包括产品型号，版本，固件发布日期等。

AT+ODR

设置模块串口输出速率。掉电保存，复位模块生效

例 设置串口输出速率为100Hz: AT+ODR=100

AT+BAUD

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

例 AT+BAUD=115200

注意

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

AT+EOUT

串口输出开关

例 打开串口输出 AT+EOUT=1 关闭串口输出 AT+EOUT=0

AT+RST

复位模块

例 AT+RST

AT+GWID

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，只有节点的GWID和接收器的GWID相同时模块和接收器才能通讯。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 三个HI221节点和一个HI221GW接收机。将HI221GW网段设置为GWID=0，将3个HI221节点的自身ID分别设置为0,1,2并连接到HI221GW上：

接收机配置：

AT+GWID=0

节点0配置：

AT+GWID=0

AT+ID=0

节点1配置：

AT+GWID=0

AT+ID=1

节点2配置：

AT+GWID=0

AT+ID=2

最后所有接收机节点复位/重新上电生效。

AT+MODE

设置模块工作模式

例

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

附录C - 固件升级与恢复出厂设置

本产品支持升级固件。固件升级步骤:

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

