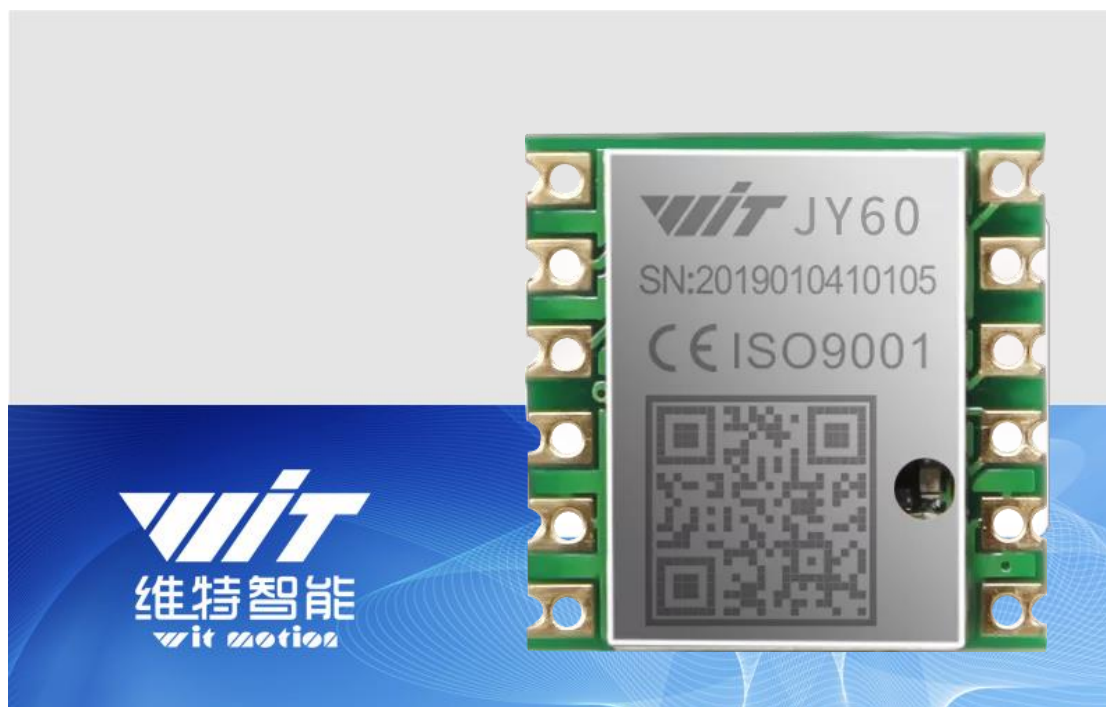


JY60 姿态角度传感器说明书



产品规格书 : SPECIFICATION

型 号: JY60

描 述: 六轴姿态角度传感器

生产执行标准参考

企业质量体系标准: ISO9001:2016 标准

倾角仪生产标准: GB/T191SJ 20873-2016

产品试验检测标准: GB/T191SJ 20873-2016

修 订 日 期: 2019.10.24

版本号	版本更新内容	更改人	日期
V1.0	发布	胡名林	20191024

目录

1 产品概述.....	- 4 -
2 性能参数.....	- 5 -
3 引脚说明.....	- 5 -
4 轴向说明.....	- 6 -
5 硬件连接方法.....	- 7 -
5.1 串口（TTL）连接.....	- 7 -
5.2 连单片机.....	- 8 -
6 上位机使用方法.....	- 8 -
6.1 使用方法.....	- 9 -
6.2 模块校准.....	- 10 -
6.2.1 Z 轴归 0.....	- 11 -
6.2.2 加计校准.....	- 12 -
6.3 设置通信波特率.....	- 13 -
6.4 记录数据.....	- 13 -
6.5 安装方向.....	- 14 -
6.6 休眠及解休眠.....	- 15 -
6.7 静止阈值及测量带宽.....	- 16 -
7 通信协议.....	- 17 -
7.1 上位机至模块.....	- 17 -
7.2 模块至上位机:.....	- 18 -
7.2.1 加速度输出:.....	- 18 -
7.2.2 角速度输出:.....	- 19 -
7.2.3 角度输出:.....	- 19 -
7.3 数据解析示例代码:.....	- 20 -
7.4 嵌入式环境下解析数据实例.....	- 21 -
8 应用领域.....	- 23 -

1 产品概述

✧ 模块内部自带电压稳定电路，可以兼容 3.3V/5V 的嵌入式系统，连接方便。

✧ 采用先进的数字滤波技术，能有效降低测量噪声，提高测量精度。

带屏蔽罩护,能提供更好的电路保护。

✧ 带温度补偿，硬件级的温补。降低由温度变化带来的干扰。（温度变化比较快、幅度较大的情况下需用 HWT605 模块才能解决）

✧ 模块内部集成了姿态解算器，配合动态卡尔曼滤波算法，能够在动态环境下准确输出模块的当前姿态，姿态测量精度 0.05 度，稳定性极高，性能甚至优于某些专业的倾角仪！

✧ 采用邮票孔镀金工艺，品质保证，可嵌入用户的 PCB 板中。

✧ 去除了较为复杂的 IIC 传输模式。

✧ 使用 icm42605 芯片，可测的角速度和加速度。

注：本模块不含磁场计，没有磁场的观测量对偏航角进行滤波，所以偏航角度是通过纯积分计算出来的，不可避免地会有漂移现象，只能实现短时间内的旋转角度测量。而 X，Y 轴角度可以通过重力场进行滤波修正，不会出现漂移现象。



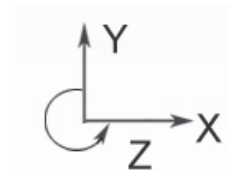
尺寸	15.24mmX15.24mm X 4mm
----	-----------------------

重量	0.6g
----	------

2 性能参数

- 1、电压：3.3V~5V
- 2、正常工作电流：<10mA 休眠工作电流：<10uA
- 3、体积：15.24mm X 15.24mm X 2mm
- 4、焊盘间距：上下 100mil(2.54mm)，左右 600mil(15.24mm)
- 5、测量维度：加速度：3 维，角速度：3 维，角度：3 维
- 6、量程：加速度： $\pm 16g$ ，角速度： $\pm 2000\text{deg/s}$ ，角度 X Z 轴 $\pm 180^\circ$ Y 轴 $\pm 90^\circ$ 。
- 7、分辨率：加速度：0.0005g，角速度： $0.61^\circ /s$ 。
- 8、测量精度：静态 0.05° ，动态 0.1° 。
- 9、数据输出内容：加速度、角速度、角度。
- 10、数据输出频率：20HZ。
- 11、**波特率：9600kps（固定不可更改）**
- 12、数据接口：串口（TTL 电平）

3 引脚说明





名称	功能
D0	NC 预留端口
VCC	模块电源, 3.3V 或 5V 输入
RX	串行数据输入, TTL 电平
TX	串行数据输出, TTL 电平
GND	地线
SWIM	SWIM 数据接口
D2	NC 预留端口
VCC	模块电源, 3.3V 或 5V 输入
SCL	NC 预留端口
SDA	NC 预留端口
GND	地线
D3	NC 预留端口

4 轴向说明

如上图所示，模块的轴向在上图右上方标示出来，向右为 X 轴，向上为 Y 轴，垂直与模块向外为 Z 轴。旋转的方向按右手法则定义，即右手大拇指指向轴向，四指弯曲的方向即为绕该轴旋转的方向。X 轴角度即为绕 X 轴旋转方向的角度，Y 轴角度即为绕 Y 轴旋转方向的角度，Z 轴角度即为绕 Z 轴旋转方向的角度。

5 硬件连接方法

5.1 串口（TTL）连接

与计算机连接，需要 USB 转 TTL 电平的串口模块。推荐以下两款 USB 转串口模块：



1. USB-TTL 串口模块：把模块和 USB-TTL 连接好，在插到电脑上。模块和 USB-TTL 连接方法是：模块的 VCC TX RX GND 分别于 USB 串口模块的+5V/3V3 RX TX GND 对应相接，**注意 TX 和 RX 需要交叉**，即 TX 接 RX，RX 接 TX。
2. 六合一模块：模块拨码开关 1 拨至 ON，拨码开关 2 拨至 2，开关 S1 拨至 other（丝印）。模块的 VCC TX RX GND 分别于六合一模块的+5V/3V3 RX TX GND 对应相接，**注意 TX 和 RX 需要交叉**，即 TX 接 RX，RX 接 TX。



5.2 连单片机



5.3

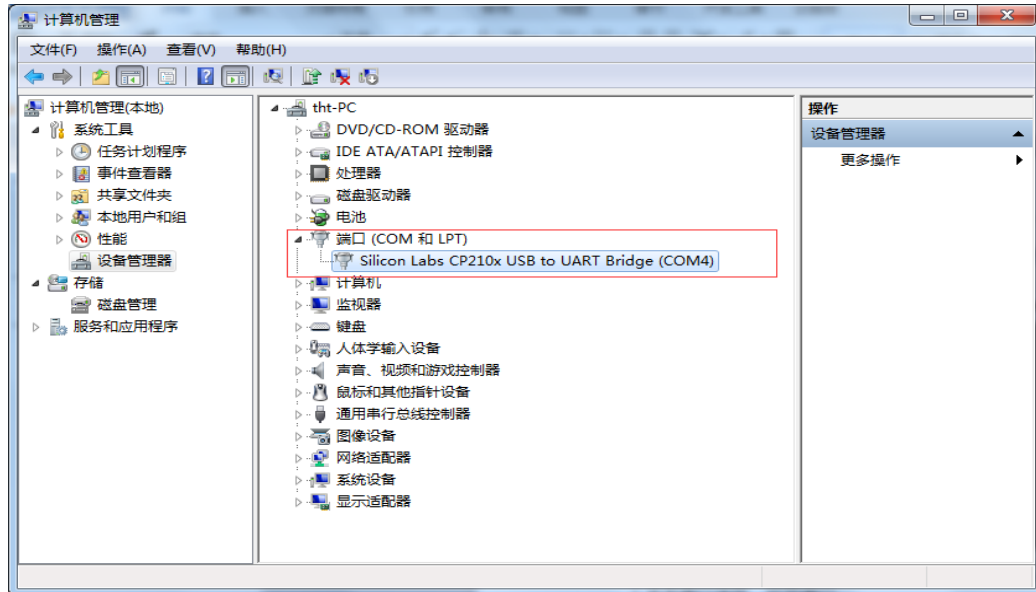
6 上位机使用方法

6.1 使用方法

注意，上位机无法运行的用户请下载安装.net framework4.0:

<http://www.microsoft.com/zh-cn/download/details.aspx?id=17718>

通过 USB-TTL 模块连接上电脑打开上位机，安装好串口模块对应的驱动 CP210X 以后，可以再设备管理器中查询到对应的端口号，如图所示：



驱动程序为 CP210X，如下：

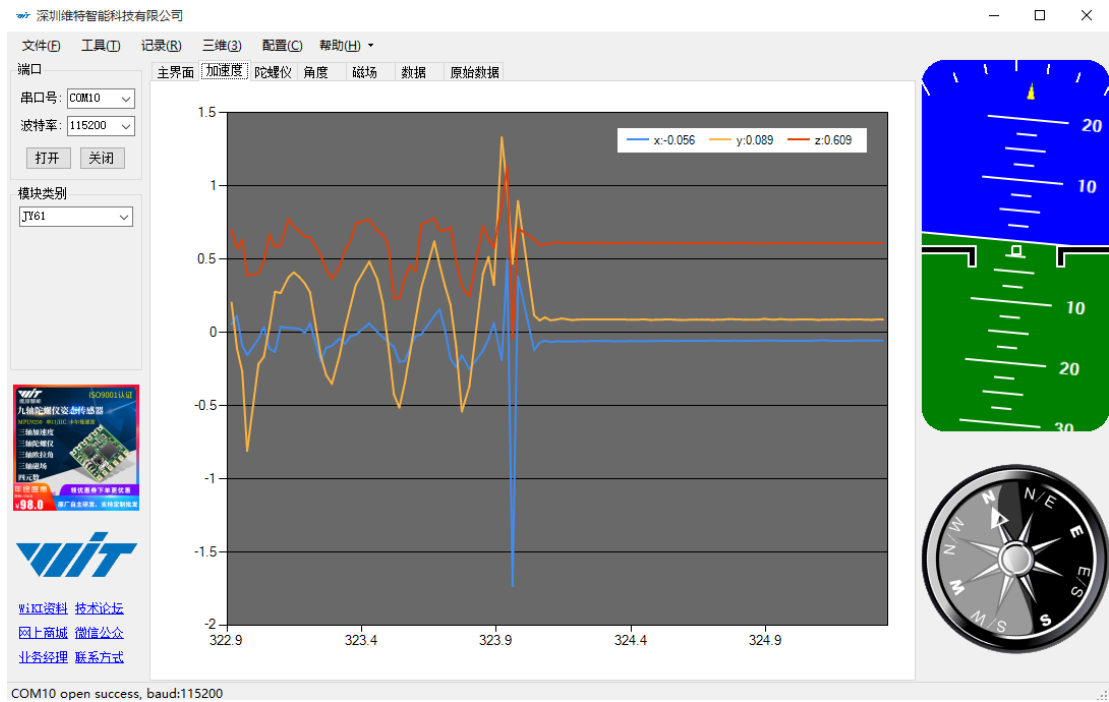
<http://pan.baidu.com/s/1o6Rleae?frm=fujian>

打开 MiniIMU.exe 软件，在【资料包/上位机】中，点击串口选择菜单，选择刚才设备管理器里面看到的 COM 号。

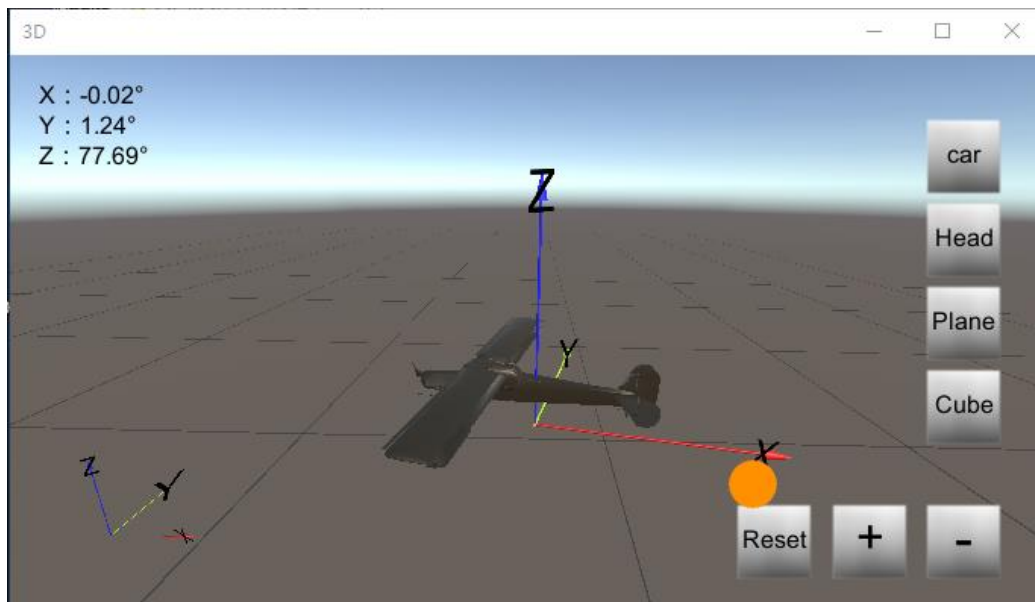


在上位机软件上点击型号菜单选择型号为 WT60。

在上位机软件上点击波特率菜单选择波特率 9600（固定不可更改），选择完成后，上位机软件上即可出现数据。



点击三维按钮，可以调出三维显示界面，显示模块的三维姿态。



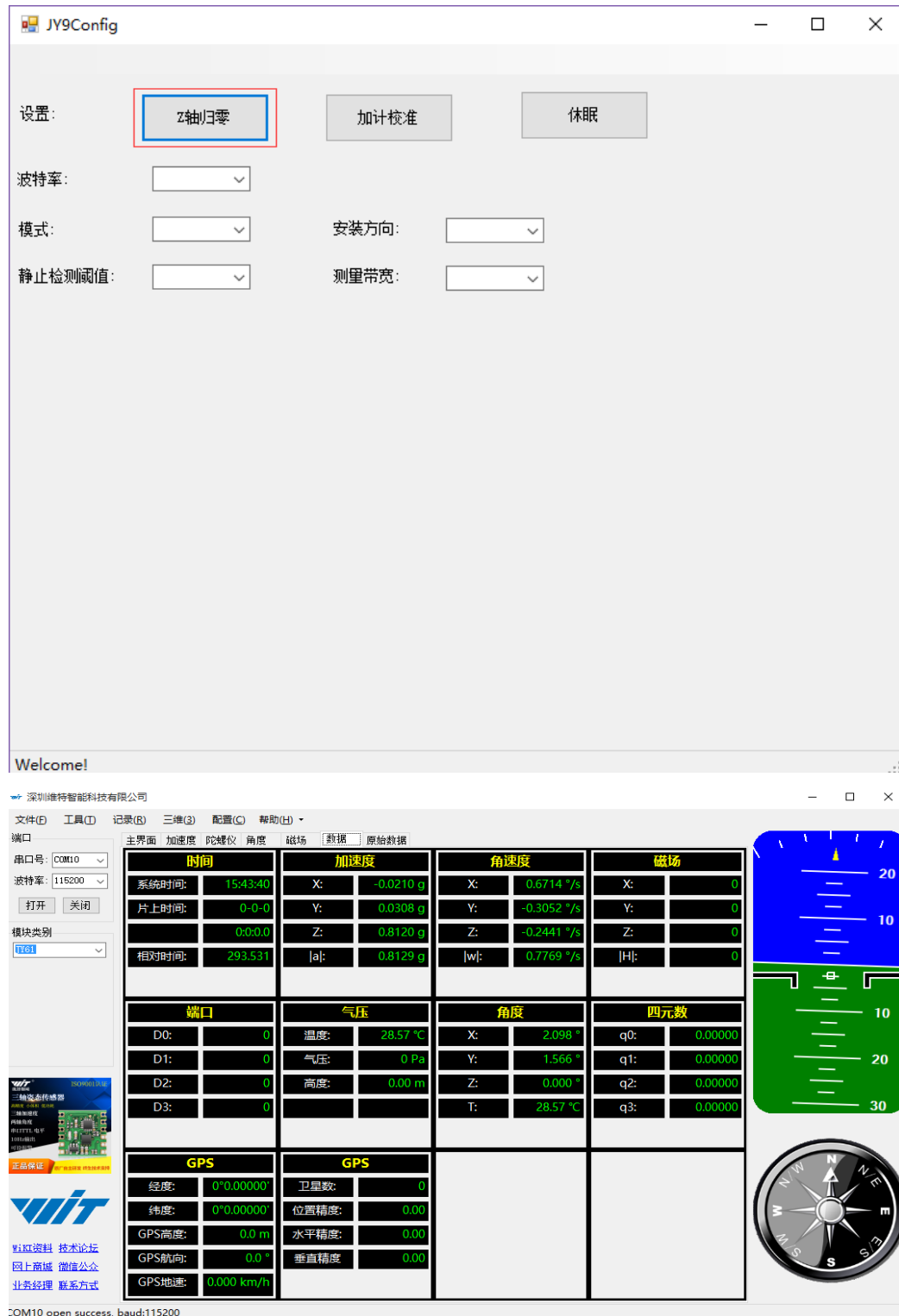
6.2 模块校准

模块使用前，需要对模块进行校准。模块的校准包括 Z 轴归 0、加计校准。

6.2.1 Z 轴归 0

Z 轴归 0 是使模块 Z 轴角度初始状态为相对 0 度角，模块使用前和 Z 轴漂移较大的情况下可以进行 Z 轴归 0 校准，模块上电时 Z 轴会自动归 0。

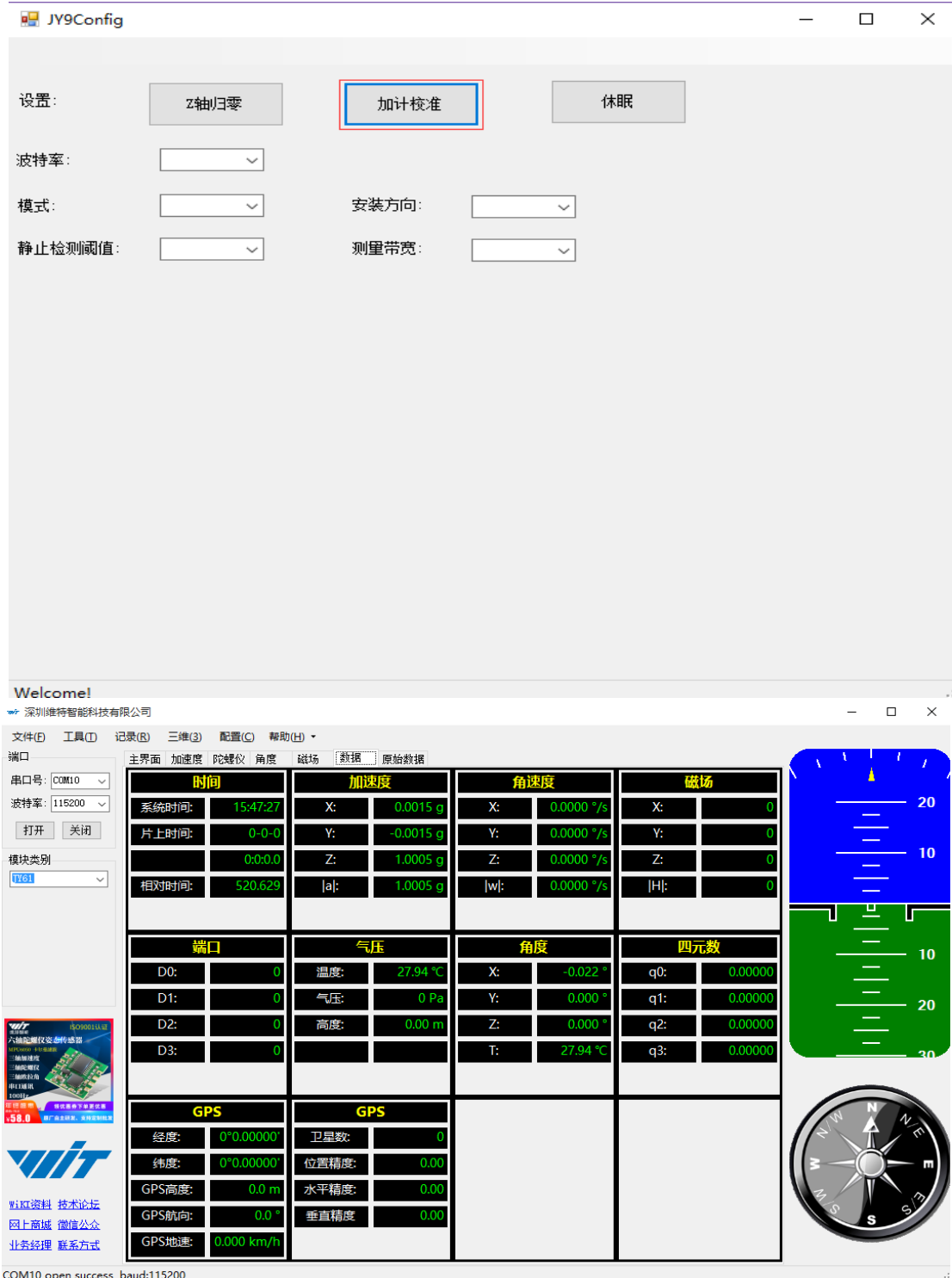
上位机 Z 轴归 0 方法如下：首先模块静止放置，点击配置打开配置栏，在配置栏里面的“Z 轴归零”选项，模块数据栏里面可以看到 Z 轴角度回到 0°。



6.2.2 加计校准

加计校准用于去除加速度计的零偏。传感器在出厂时都会有不同程度的零偏误差，需要手动进行校准后，测量才会准确。

加计校准方法如下：首先使模块保持水平静止，点击加计校准，1~2 秒后模块加速度三个轴向的值会在 0 0 1 左右，X 和 Y 轴角度在 0° 左右。校准后 X Y 轴角度就更精确了。



注意：Z 轴水平静止的时候是有 1 个 G 的重力加速度的。

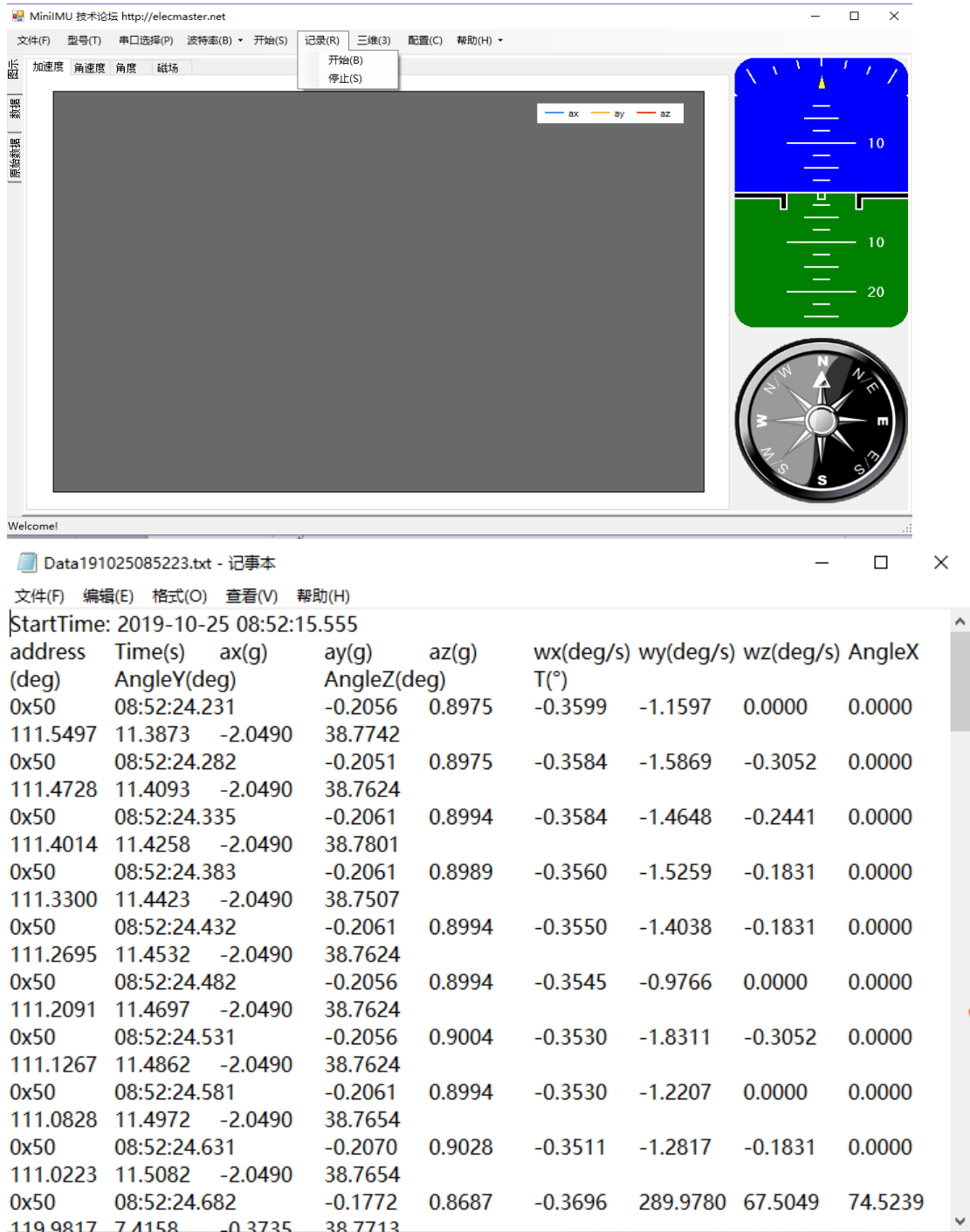
6.3 设置通信波特率

模块默认波特率为 9600，固定且不可更改，波特率为 9600 时模块回传速率为 20HZ。

6.4 记录数据

传感器模块内部不带存储芯片，数据可以通过上位机来记录保存。

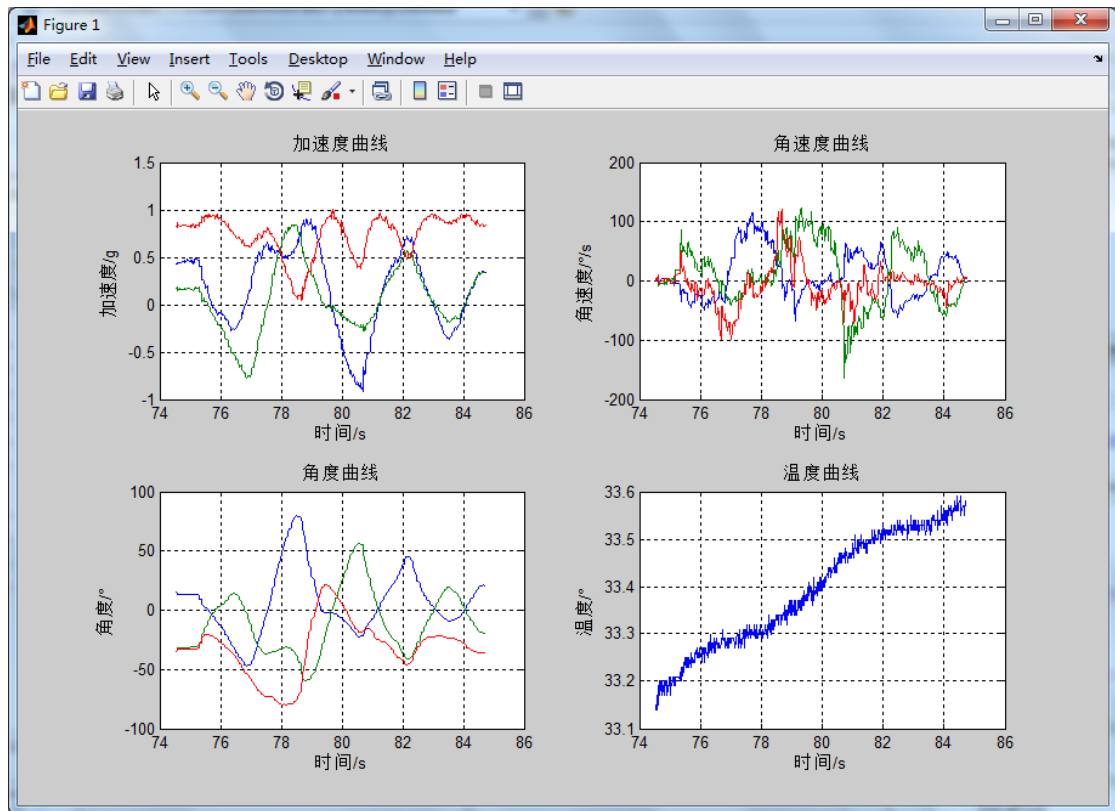
使用方法：点记录按钮可以将数据保存为文件



保存的文件在上位机程序的目录下 Data.txt:

文件开头有标明数据对应的值, Time 代表时间, ax ay az 分别表示 x y z 三个轴向上的加速度, wx wy wz 分别表示 x y z 三个轴向上的角速度, Anglex Angley Anglez 分别表示 x y z 三个轴向上的角度, T 代表时间。

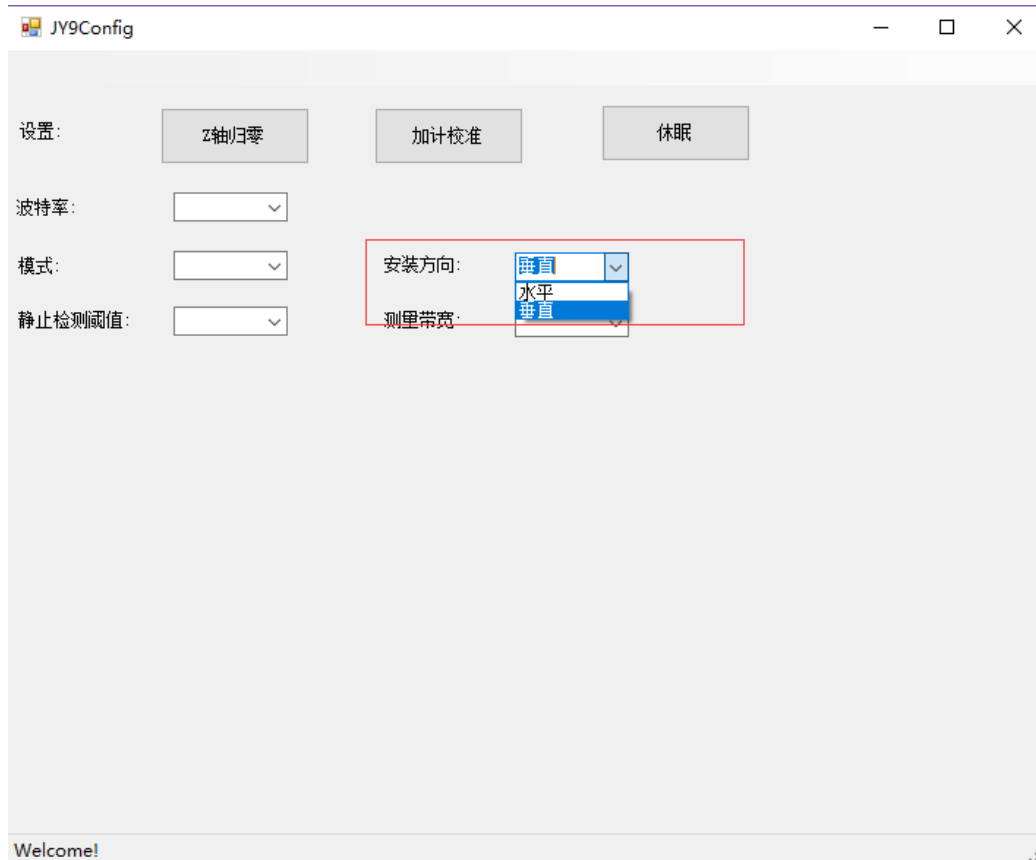
数据可以导入到 Excel 或者 Matlab 中进行分析。在 Matlab 环境下运行上位机根目录下的 “Matlab 绘图.m” 文件, 可以绘制数据曲线图。



6.5 安装方向

模块默认安装方向为水平安装, 当模块需要垂直放置时, 可以用垂直安装设置。

垂直安装方法: 垂直安装时, 把模块绕 X 轴旋转 90° 垂直放置, 在上位机配置栏里面 “安装方向” 选项中选择 “垂直”。设置完成后要进行校准才能使用。



6.6 休眠及解休眠

休眠：模块暂停工作，进入待机状态。休眠后可以降低功耗。

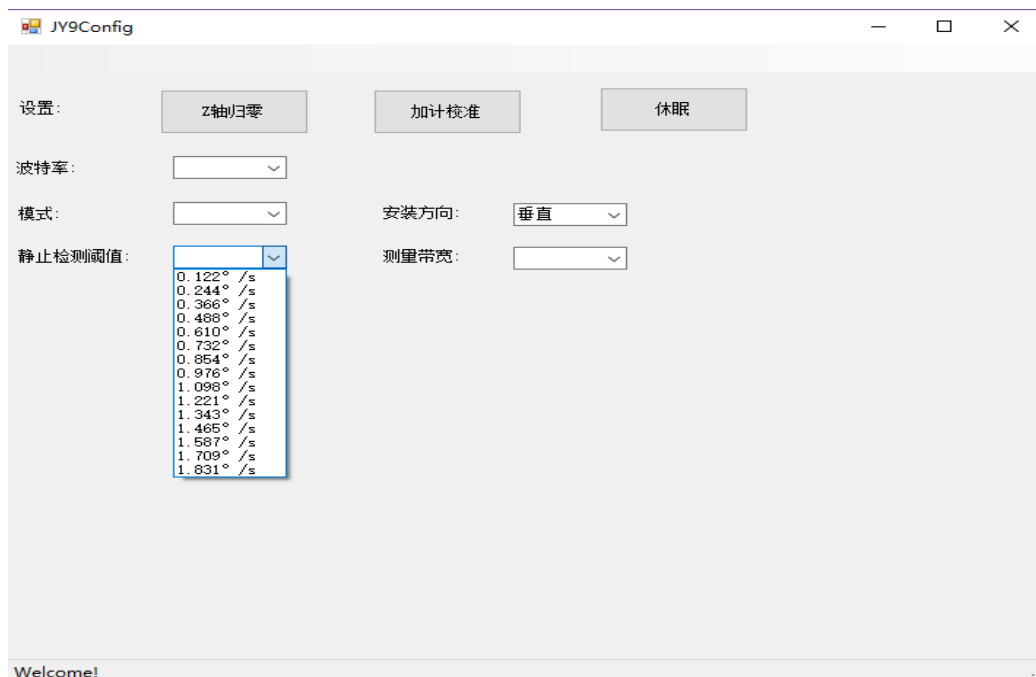
解休眠：模块从待机状态进入工作状态。

使用方法：模块默认为工作状态，在上位机配置栏里面点击“休眠”选项，进入休眠状态，再点击“休眠”选项，模块解除休眠。

6.7 静止阈值及测量带宽

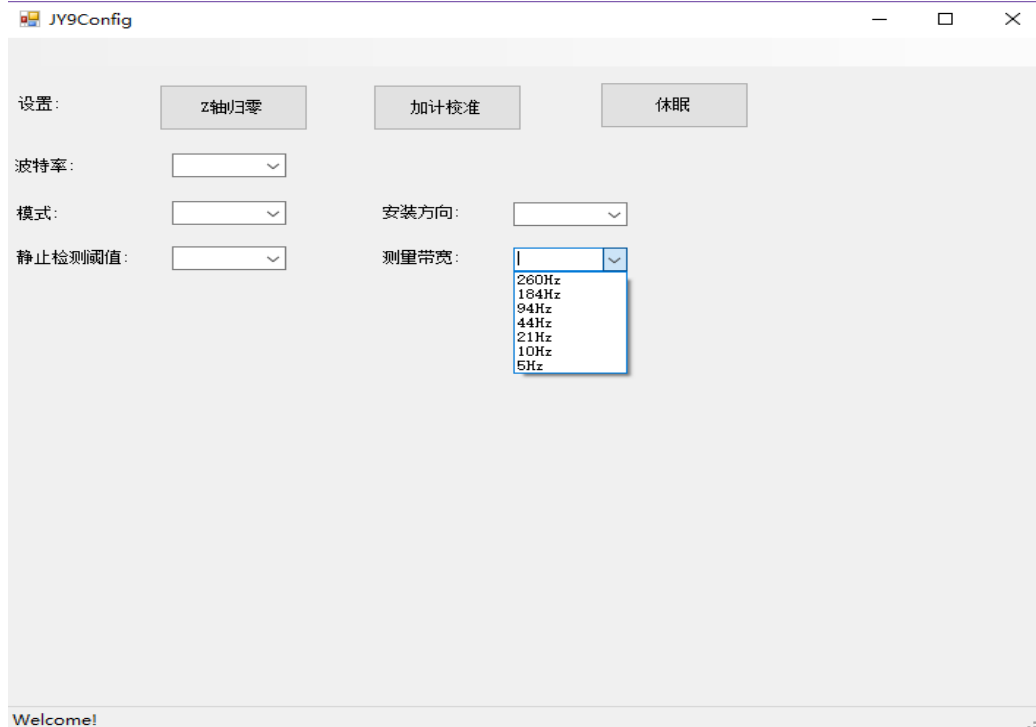
静止阈值：模块静止时，陀螺仪芯片测量的角速度是有微小变化的。静止阈值的作用是当角速度小于阈值时，模块输出角速度为 0。（**注意角速度在匀速转动的情况下，输出有问题，推荐在匀速转动下用 61P**）

使用方法：在上位机配置栏里面点击“静止阈值”选项，即可设置阈值。模块默认为 $0.122^{\circ}/s$ 。



测量带宽：模块只输出测量带宽以内的数据，大于带宽的数据会自动滤除。

使用方法：在上位机配置栏里面点击“测量带宽”选项，即可设置。默认为 10HZ。



注：静止阈值和测量带宽一般为默认的了就可以了，不需要去设置。

7 通信协议

电平：TTL 电平（非 RS232 电平，若将模块错接到 RS232 电平可能造成模块损坏）

波特率：9600，停止位 1，校验位 0。

7.1 上位机至模块

指令内容	功能	备注
0xFF 0xAA 0x52	角度初始化	使 Z 轴角度归零
0xFF 0xAA 0x67	加速度计校准	校准加速度零偏
0xFF 0xAA 0x60	休眠及解休眠	待机模式和工作模式
0xFF 0xAA 0x65	水平安装	模块水平放置
0xFF 0xAA 0x66	垂直安装	模块垂直放置

说明：

1.模块上电以后需先保持静止，模块内部的 MCU 会在模块静止的时候进行自动校准（消除陀螺零漂），校准以后 Z 轴的角度会重新初始化为 0，Z 轴角度输出为 0 时，可视为自动校准完成的信号。

2.配置可通过上位机软件配置，因为所有配置都是掉电保存的，所以只需配置一次就行。

7.2 模块至上位机:

模块发送至上位机每帧数据分为 3 个数据包，分别为加速度包，角速度包和角度包，3 个数据包顺序输出。波特率 9600 时每隔 50ms 输出 1 帧数据。

7.2.1 加速度输出:

数据编号	数据内容	含义
0	0x55	包头
1	0x51	标识这个包是加速度包
2	AxL	X 轴加速度低字节
3	AxH	X 轴加速度高字节
4	AyL	Y 轴加速度低字节
5	AyH	Y 轴加速度高字节
6	AzL	Z 轴加速度低字节
7	AzH	Z 轴加速度高字节
8	TL	温度低字节
9	TH	温度高字节
10	Sum	校验和

加速度计算公式:

$$a_x = ((AxH \ll 8) | AxL) / 32768 * 16g \text{ (g 为重力加速度, 可取 } 9.8m/s^2 \text{)}$$

$$a_y = ((AyH \ll 8) | AyL) / 32768 * 16g \text{ (g 为重力加速度, 可取 } 9.8m/s^2 \text{)}$$

$$a_z = ((AzH \ll 8) | AzL) / 32768 * 16g \text{ (g 为重力加速度, 可取 } 9.8m/s^2 \text{)}$$

温度计算公式:

$$T = ((TH \ll 8) | TL) / 340 + 36.53 \text{ } ^\circ C$$

校验和:

$$Sum = 0x55 + 0x51 + AxH + AxL + AyH + AyL + AzH + AzL + TH + TL$$

说明:

- 1、数据是按照 16 进制方式发送的，不是 ASCII 码。
- 2、每个数据分低字节和高字节依次传送，二者组合成一个有符号的 short 类型的数据。例如 X 轴加速度数据 Ax，其中 AxL 为低字节，AxH 为高字节。转换方法如下:

假设 Data 为实际的数据，DataH 为其高字节部分，DataL 为其低字节部分，那么： $Data = ((short)DataH \ll 8) | DataL$ 。这里一定要注意 DataH 需要先强制转换为一个有符号的 short 类型的数据以后再移位，并且 Data 的数据类型也是有符号的 short 类型，这样才能表示出负数。

详细解算示例:

<http://www.openedv.com/forum.php?mod=viewthread&tid=79352&page=1&extra=#pid450195>

7.2.2 角速度输出：

数据编号	数据内容	含义
0	0x55	包头
1	0x52	标识这个包是角速度包
2	wxL	X 轴角速度低字节
3	wxH	X 轴加速度高字节
4	wyL	Y 轴加速度低字节
5	wyH	Y 轴加速度高字节
6	wzL	Z 轴加速度低字节
7	wzH	Z 轴加速度高字节
8	TL	温度低字节
9	TH	温度高字节
10	Sum	校验和

角速度计算公式：

$$w_x = ((wxH < 8) | wxL) / 32768 * 2000 (^\circ/s)$$

$$w_y = ((wyH < 8) | wyL) / 32768 * 2000 (^\circ/s)$$

$$w_z = ((wzH < 8) | wzL) / 32768 * 2000 (^\circ/s)$$

温度计算公式：

$$T = ((TH < 8) | TL) / 340 + 36.53 \text{ } ^\circ\text{C}$$

校验和：

$$\text{Sum} = 0x55 + 0x52 + wxH + wxL + wyH + wyL + wzH + wzL + TH + TL$$

7.2.3 角度输出：

数据编号	数据内容	含义
0	0x55	包头
1	0x53	标识这个包是角度包
2	RollL	X 轴角度低字节
3	RollH	X 轴角度高字节
4	PitchL	Y 轴角度低字节
5	PitchH	Y 轴角度高字节
6	YawL	Z 轴角度低字节
7	YawH	Z 轴角度高字节
8	TL	温度低字节
9	TH	温度高字节
10	Sum	校验和

角速度计算公式：

$$\text{滚转角 (x 轴) Roll} = ((RollH < 8) | RollL) / 32768 * 180 (^\circ)$$

$$\text{俯仰角 (y 轴) Pitch} = ((PitchH < 8) | PitchL) / 32768 * 180 (^\circ)$$

$$\text{偏航角 (z 轴) Yaw} = ((YawH < 8) | YawL) / 32768 * 180 (^\circ)$$

温度计算公式:

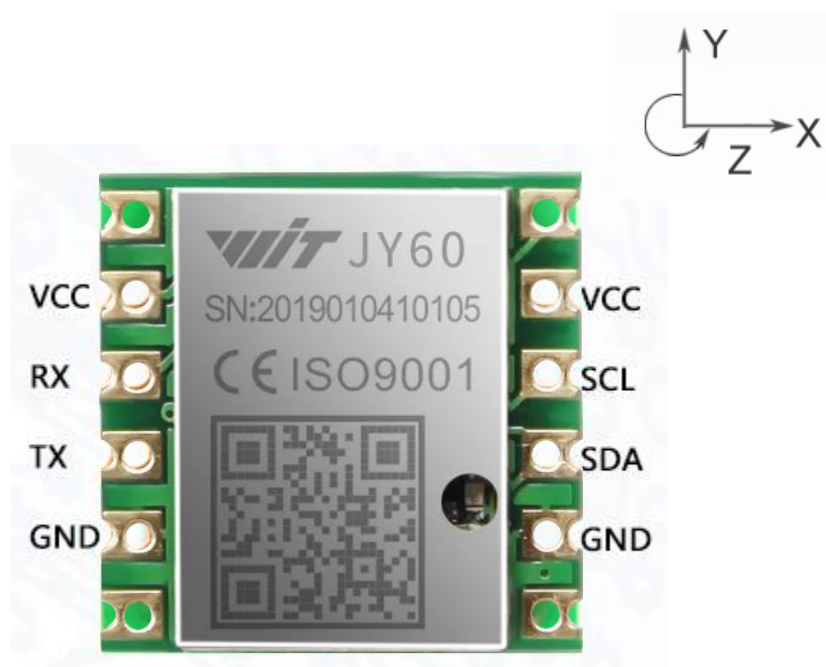
$$T = ((TH < 8) | TL) / 340 + 36.53 \text{ } ^\circ\text{C}$$

校验和:

$$\text{Sum} = 0x55 + 0x53 + \text{RollH} + \text{RollL} + \text{PitchH} + \text{PitchL} + \text{YawH} + \text{YawL} + TH + TL$$

注:

1. 姿态角解算时所使用的坐标系为东北天坐标系, 正方向放置模块, 如下图所示向右为 X 轴, 向上为 Y 轴, 垂直模块向外为 Z 轴。欧拉角表示姿态时的坐标系旋转顺序定义为 Z-Y-X, 即先绕 Z 轴转, 再绕 Y 轴转, 再绕 X 轴转。
2. 滚转角的范围虽然是 ± 180 度, 但实际上由于坐标旋转顺序是 Z-Y-X, 在表示姿态的时候, 俯仰角 (Y 轴) 的范围只有 ± 90 度, 超过 90 度后会变换到小于 90 度, 同时让 X 轴的角度大于 180 度。详细原理请自行百度欧拉角及姿态表示的相关信息。
3. 由于三轴是耦合的, 只有在小角度的时候会表现出独立变化, 在大角度的时候姿态角度会耦合变化, 比如当 Y 轴接近 90 度时, 即使姿态只绕 Y 轴转动, X 轴的角度也会跟着发生较大变化, 这是欧拉角表示姿态的固有问题。



7.3 数据解析示例代码:

```
double a[3], w[3], Angle[3], T;
void DecodeIMUData(unsigned char chrTemp[])
{
    switch(chrTemp[1])
    {
        case 0x51:
            a[0] = (short(chrTemp[3]<<8|chrTemp[2]))/32768.0*16;
            a[1] = (short(chrTemp[5]<<8|chrTemp[4]))/32768.0*16;
            a[2] = (short(chrTemp[7]<<8|chrTemp[6]))/32768.0*16;
            T = (short(chrTemp[9]<<8|chrTemp[8]))/340.0+36.25;
            break;
    }
}
```

```
case 0x53:
    Angle[0] = (short(chrTemp[3]<<8|chrTemp[2]))/32768.0*180;
    Angle[1] = (short(chrTemp[5]<<8|chrTemp[4]))/32768.0*180;
    T = (short(chrTemp[9]<<8|chrTemp[8]))/340.0+36.25;
    printf("a = %4.3f\t%4.3f\t%4.3f\t\r\n",a[0],a[1],a[2]);
    printf("Angle = %4.2f\t%4.2f\tT=%4.2f\r\n",Angle[0],Angle[1],T);
    break;
}
}
```

7.4 嵌入式环境下解析数据实例

分成两个部分，一个是中断接收，找到数据的头，然后把数据包放入数组中。另一个是数据解析，放在主程序中。

中断部分(一下为 AVR 单片机代码，不同单片机读取寄存器略有差异，需根据实际情况调整):

```
unsigned char Re_buf[11],counter=0;
unsigned char sign;
interrupt [USART_RXC] void usart_rx_isr(void) //USART 串行接收中断
{
    Re_buf[counter]=UDR;//不同单片机略有差异
    if(counter==0&&Re_buf[0]!=0x55) return;    //第 0 号数据不是帧
    头, 跳过
    counter++;
    if(counter==11) //接收到 11 个数据
    {
        counter=0; //重新赋值, 准备下一帧数据的接收
        sign=1;
    }
}
```

主程序部分:

```
float a[3],w[3],angle[3],T;
extern unsigned char Re_buf[11],counter;
extern unsigned char sign;
while(1)
{
    if(sign)
    {
        sign=0;
        if(Re_buf[0]==0x55)    //检查帧头
        {
            switch(Re_buf [1])
            {
                case 0x51:
                    a[0] = (short(Re_buf [3]<<8| Re_buf [2]))/32768.0*16;
                    a[1] = (short(Re_buf [5]<<8| Re_buf [4]))/32768.0*16;
                    a[2] = (short(Re_buf [7]<<8| Re_buf [6]))/32768.0*16;
                    T = (short(Re_buf [9]<<8| Re_buf [8]))/340.0+36.25;
                    break;
            }
        }
    }
}
```

```
case 0x52:
    w[0] = (short(Re_buf [3]<<8| Re_buf [2]))/32768.0*2000;
    w[1] = (short(Re_buf [5]<<8| Re_buf [4]))/32768.0*2000;
    w[2] = (short(Re_buf [7]<<8| Re_buf [6]))/32768.0*2000;
    T = (short(Re_buf [9]<<8| Re_buf [8]))/340.0+36.25;
    break;
case 0x53:
    angle[0] = (short(Re_buf [3]<<8| Re_buf [2]))/32768.0*180;
    angle[1] = (short(Re_buf [5]<<8| Re_buf [4]))/32768.0*180;
    angle[2] = (short(Re_buf [7]<<8| Re_buf [6]))/32768.0*180;
    T = (short(Re_buf [9]<<8| Re_buf [8]))/340.0+36.25;
    break;
}
}
}
```

具体程序代码可以参考资料里的示例代码。

8 应用领域

农业机械



物联网



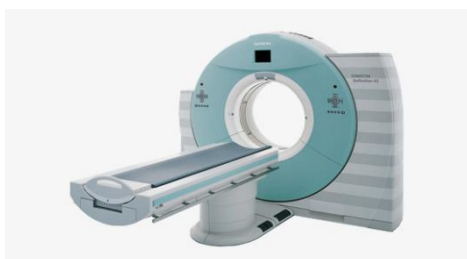
太阳能



电力监控



医疗器械



工程机械



地质监测





深圳维特智能科技有限公司

WitMotion ShenZhen Co., Ltd

JY60 姿态角度传感器

电话: 0755-33185882

邮箱: wit@wit-motion.com

网站: www.wit-motion.com

店铺: <https://robotcontrol.taobao.com>

地址: 广东省深圳市宝安区松岗镇星际家园宏海大厦