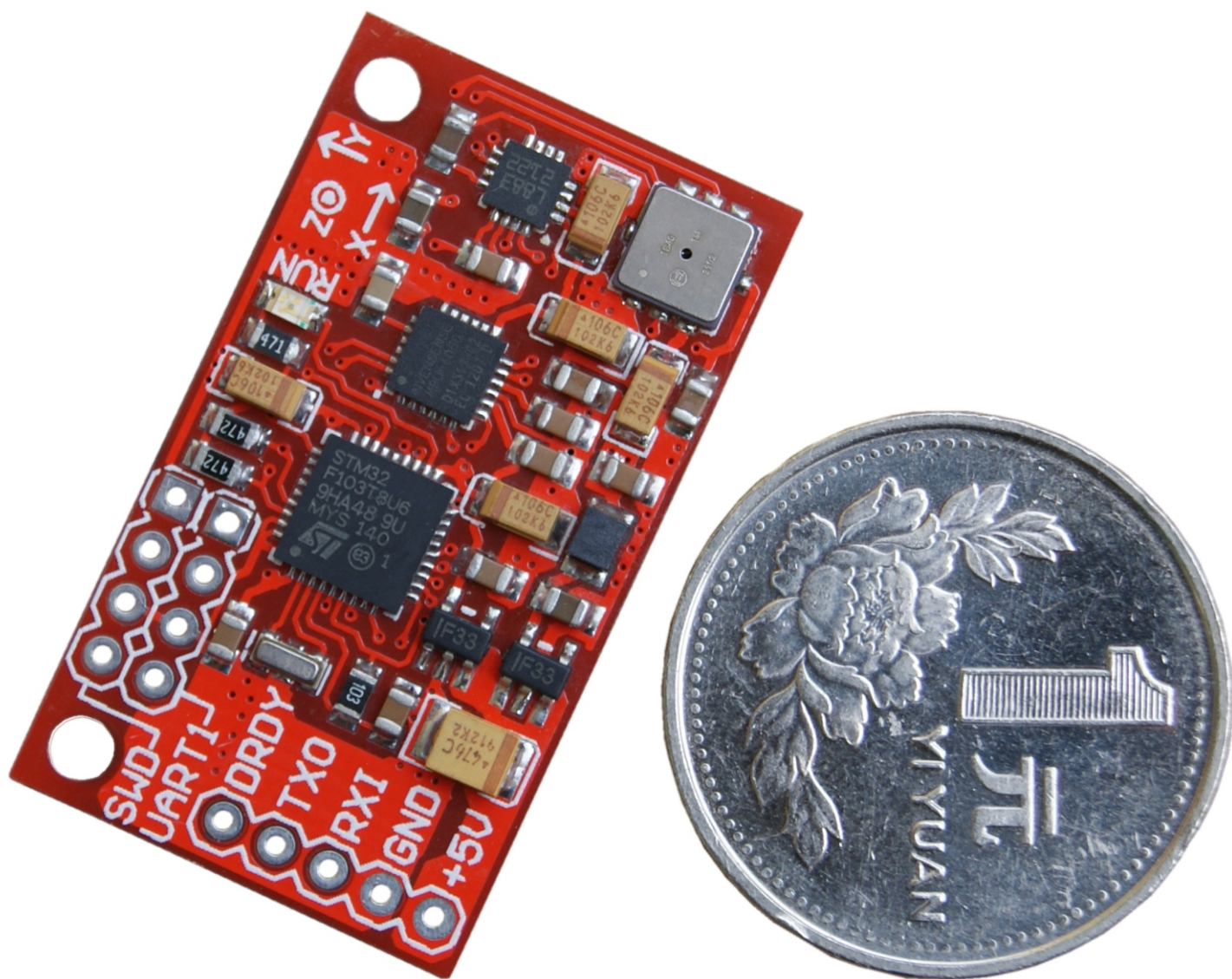


Mini IMU AHRS 数据输出

本章旨在说明如何接收Mini AHRS 发送的数据包括：

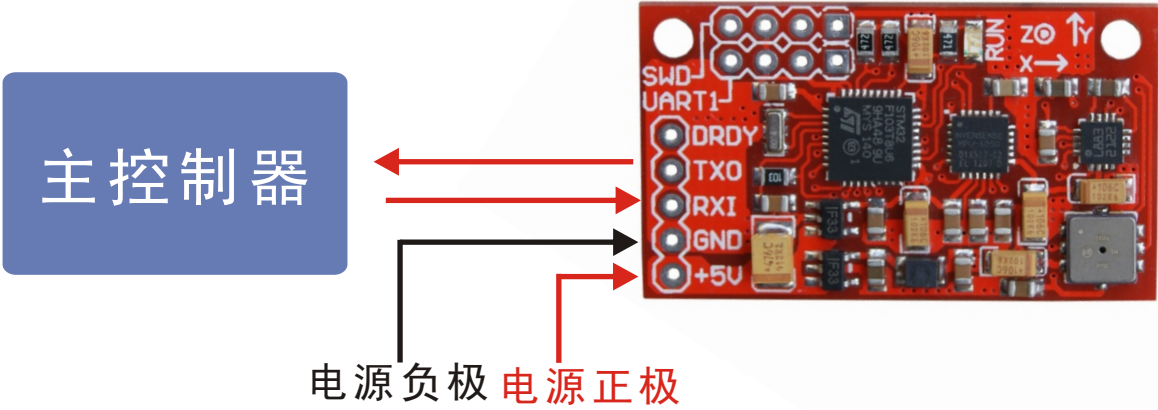
1. 使用串口**USART**接口器接收数据流
2. 将接收到的数据进行解帧得到真正的数据



Mini IMU与您的系统连接

Mini IMU 专注于姿态解算，需要将解算完成后的数据通过板子上的UART传送给使用这些数据的系统。我们称之为控制器。那么，Mini IMU要与你的控制器连接需要满足以下条件：

- 1. 一致的波特率，默认的波特率为115200bps
- 2. 一致的帧格式，Mini IMU 使用8位数据，一个起始位+一个停止位
- 3. 一致的电平，Mini IMU 的UART接口支持 TTL 电平，可以与5V的TTL电平相连接。



特别提示

TX0 为模块的数据发送接口，信号方向为输出
RXI 是模块的数据接收端口，信号方向为输入

除了连接以上的通信装置外，你还需要给模块供电才能使它工作。Mini AHRs内部有稳压芯片，支持4-7V的直流电源输入

数据帧分析



起始字节 A5 5A



除了起始字节外，所有的字节数

帧功能 A1 表示这是IMU报告

结束字节 AA

解算后的数据帧格式

0xA5	0x5A	0x12	0xA1	Yaw	Pitch	Roll	高度	温度	气压	解算速度	校验	0xAA
------	------	------	------	-----	-------	------	----	----	----	------	----	------

0xA5 0x5A 为帧的起始字节
0x12 代表，后面带有18[十六进制12]个字节的数据。
0xA1 帧标识字节，用来指明此帧带的是解算后的角度数据
Yaw 经过解算后的航向角度。单位为0.1度 0 -> 3600 对应 0 -> 360.0度
Pitch 解算得到的俯仰角度，单位 0.1度。-900 - 900 对应 -90.0 -> 90.0 度
Roll 解算后得到的横滚角度，单位0.1度。 -1800 -> 1800 对应 -180.0 -> 180.0度
高度 单位0.1米。 范围一个整型变量
温度 单位0.1摄氏度 范围：直到你的电路板不能正常工作
气压压力 单位10Pa
解算速率 姿态运算1MUp^{er}sec每秒。
校验 以上数据，除了起始字节A5 5A 外累加合取低字节做校验
0xAA 结束字节

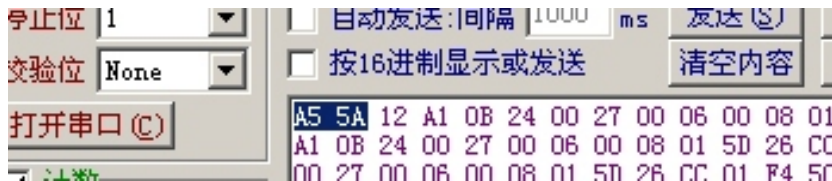
传感器的输出值

0xA5	0x5A	0x16	0xA2	ACCx	ACCy	ACCz	GYROx	GYROy	GYROz	Mx	My	Mz	校验	0xAA
------	------	------	------	------	------	------	-------	-------	-------	----	----	----	----	------

0xA5 0x5A 为帧的起始字节
0x16 代表，后面带有22[十六进制16]个字节的数据。
0xA2 帧标识字节，用来指明此帧带的是传感器输出值
ACCx 加速度 X轴ADC输出 范围：一个有符号整型
ACCy 加速度 Y轴ADC输出 范围：一个有符号整型
ACCz 加速度 Z轴ADC输出 范围：一个有符号整型
GYROx 陀螺仪 X轴ADC输出 范围：一个有符号整型
GYROy 陀螺仪 Y轴ADC输出 范围：一个有符号整型
GYROz 陀螺仪 Z轴ADC输出 范围：一个有符号整型
Mx 磁罗盘 X轴ADC输出 范围：一个有符号整型
My 磁罗盘 Y轴ADC输出 范围：一个有符号整型
Mz 磁罗盘 Z轴ADC输出 范围：一个有符号整型
校验 以上数据，除了起始字节A5 5A 外累加合取低字节做校验
0xAA 结束字节

串行接口的软件框图

从接收的数据流中找到帧起始字节



从将起始字节后的字节总数拿出来



清除接收指针，将字节总数存放在缓冲区的第一个字节。开始接收帧数据

将后面的数据存入缓冲区并计数接收到的字节数是否等于帧总字节数 判断完成收帧



检查帧校验是否通过

从接收到的帧数据中提取数据

数据接收的程序编写

特别提示

为了使你的系统软件更快地响应数据字节的接收，建议使用中断方式接收数据。而不是使用查询模式接收字节。

中断程序越精简越好，下面给出一个数据帧接收的范例程序。基于STM32F单片机。移植需要开启串口中断，并在相应的中断程序中编写以下代码

串行接口中断程序编写

```
//uart reicer flag
#define b_uart_head 0x80 //收到A5 头 标志位
#define b_rx_over 0x40 //收到完整的帧标志
// USART Receiver buffer
#define RX_BUFFER_SIZE 100 //接收缓冲区字节数

volatile unsigned char rx_buffer[RX_BUFFER_SIZE]; //接收数据缓冲区
volatile unsigned char rx_wr_index; //缓冲写指针
volatile unsigned char RC_Flag; //接收状态标志字节

//以下中断子程序是基于STM32的，如果你使用的不是STM32 请做相应的修改
//-----
void USART2_IRQHandler(void) //接收中断子程序
{
    unsigned char data;
    if(USART_GetITStatus(USART2, USART_IT_RXNE) != RESET) //是否收到数据?
    {
        data=USART_ReceiveData(USART2); //读取接收到的字节
        if(data==0xa5) { RC_Flag|=b_uart_head; //如果接收到A5 置位帧头标专位
            rx_buffer[rx_wr_index++]=data; //保存这个字节.
        }
        else if(data==0x5a)
        {
            if(RC_Flag&b_uart_head) //如果上一个字节是A5 那么认定 这个是帧起始字节
            { rx_wr_index=0; //重置 缓冲区指针
                RC_Flag&=~b_rx_over; //这个帧才刚刚开始收
            }
            else //上一个字节不是A5
            { rx_buffer[rx_wr_index++]=data;
                RC_Flag&=~b_uart_head; //清帧头标志
            }
        }
        else
        { rx_buffer[rx_wr_index++]=data;
            RC_Flag&=~b_uart_head;
            if(rx_wr_index==rx_buffer[0]) //收够了字节数.
            {
                RC_Flag|=b_rx_over; //置位 接收完整的一帧数据
            }
        }
    }

    if(rx_wr_index==RX_BUFFER_SIZE) //防止缓冲区溢出
    rx_wr_index--;
    /* Clear the USART2 Recive interrupt */
    USART_ClearITPendingBit(USART2, USART_IT_RXNE); //清中断标志.
}
```

数据的校验

Mini AHRS发送的数据都带有校验，在接收完一帧数据后，第一件事就是校验数据，以确定数据的完整性。当前的数据帧使用的是累加合校验。就是除了起始字节 检验字节本身还有结束字节外，所有的字节都累加。最后得到的将是校验字节。

```
//--校验当前接收到的一帧数据是否 与帧校验字节一致
unsigned char Sum_check(void)
{   unsigned char i;
    unsigned int checksum=0;
    for(i=0;i<rx_buffer[0]-2;i++)
        checksum+=rx_buffer[i];
    if((checksum%256)==rx_buffer[rx_buffer[0]-2])
        return(0x01); //Checksum successful
    else
        return(0x00); //Checksum error 校验错误，此帧数据不可用
}
```

更详细的数据解帧 请参考光盘中提供的C语言解帧范例。

Thanks !

感谢你关注和选择mini IMU

MiniIMU AHRS姿态板销售网址：
[Http://chiplab7.taobao.com/](http://chiplab7.taobao.com/)