# vofa+上位机使用教程

## 一．数据上传

### vofa+简介

VOFA+是一款直观、灵活、强大且具有高自由度的上位机软件，支持串口、UDP、TCP通信接口，个人感觉其突出的优点就是支持浮点数高频传输、具有功能完整的数据接收、发送、及可视化的插件。其有**FireWater、RawData、JustFloat** 3种数据协议。

**FireWater**是字符串流，MCU等下位机编程用printf即可实现发送，非常简单，但该协议比较耗计算资源，适用于通道数量不多，发送频率不高的场合。

**RawData**协议下就是串口助手的功能，收到字节流就显示字节流，收到字符串流就显示字符串流，不能解析数据。

**JustFloat**是我主要用到的协议，支持的是小端浮点数组形式的字节流，简单说就是在下位机将要发送到上位机的n个float数据（不是float类型的转换为float类型）按顺序转化为小端字节序的字节数组，并以0x00, 0x00, 0x80, 0x7f结尾，这样vofa+接收到后就可以自动解析为n个浮点数。比如，要发送1.0,2.0,3.0到上位机，则是0x00 00 80 3f 00 00 00 04 00 00 40 40 00 00 80 7f（注意小端字节序且以00 00 80 7f！）。JustFloat协议适合发送频率高、通道数量多的场合，可以满足大部分数据采集及可视化的需求。

### 使用教程

硬件环境：STM32G4开发板、USB转232线、PC

软件环境：Keil、vofa+1.3.10

实验内容：STM32以1kHz频率发送3个相位间隔120°的正弦波

波特率计算：发送3个浮点数+4个字节的帧尾，对应的bit数为 （3\*4+4）\*8=128bit，所以1kHz的发送频率对应的波特率为128\*1000=128000bps，根据vofa+的可选波特率，选择波特率为230400bps

### Keil代码

float num[3]={0};//要发送的3个浮点数

Uint16 txBuf[8]={0};//发送给上位机的字节数组，3个浮点数+4个字节的帧尾为16个字节,需要长度为8的Uint16数组

float Ts=0.001;//发送周期，单位s

float ts=0.01;//正弦波周期

float amp=16.0;//正弦波幅值

float t=0;

txBuf[6]=0x0000;//将帧尾写好

txBuf[7]=0x7f80;

while(1)

{

num[0]=amp\*sinf(2\*M\_PI/ts\*t);

num[1]=amp\*sinf(2\*M\_PI/ts\*t-120\*M\_PI/180);

num[2]=amp\*sinf(2\*M\_PI/ts\*t+120\*M\_PI/180);

memcpy(txBuf,(Uint16 \*)num,3\*sizeof(float));//将num数组以小端字节序复制到数组txBuf中

HAL\_UART\_Transmit(&huart1,txBuf,4\*sizeof(float),1000);

t+=Ts;

}

### vofa+设置

按下面的步骤设置即可

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

### 效果

上位机下位机弄好，就可以看效果啦：

图片包含 表格

描述已自动生成

可以将鼠标挪动到x轴和y轴滑动鼠标滚轮随意改变缩放度。

### 问题排查

可能出现的问题及排查：

**1、数据接收不到**

建议先在RawData协议下运行，看正下方接收栏有没有数据接收到，如果没有，排查串口参数，如波特率是否和下位机配置相同。

**2、接收到数据了，但JustFloat协议下无法解析为浮点数**

看下JustFloat协议下，正下方数据接收栏有没有数据，如果RawData协议有数据接收到而JustFloat协议没有，就是没有帧尾0x 00 00 80 7f，或帧尾发送顺序有误！如果每个发送周期都有该帧尾，JustFloat协议下数据是能接收到数据的

**3、JustFloat接收到数据了但数据解析出来不是发送的数据**

这就是下位机代码有问题了，可以看下接收到的字节是不是小端字节序。浮点数对应字节我一般用MATLAB的num2hex()函数，如num2hex(single(1.0)) 得到‘3f800000’, 这个是大端的，对应小端就是0x00 00 80 3f

## 二．参数下发

### 新建命令

图形用户界面, 应用程序, 表格

描述已自动生成

### 设置命令

图形用户界面, 文本, 应用程序

描述已自动生成

其中第3步发送内容中，前面"%%"就是控件的参数（下面会讲），A0是我自己定义的协议，当帧尾是A0时，代表参数P，A1时代表I，A2时代表D

### 绑定命令

绑定命令前先选择控件，方法如下：

图形用户界面, 应用程序

描述已自动生成

当然这里可以任意选择合适的控件，大家可以自己去尝试，我这边选择的是slider，第2步中命令发送内容中的"%%"就是控件的数值，同数据接收一样，为float类型，字节流发送。可以在"name"处更改命名。可以右键控件设置相关参数。

然后绑定命令，步骤如下

图形用户界面, 应用程序

描述已自动生成

这样一个P参数命令就设置完成了，另外两个参数也一样，最后的效果如下图所示图形用户界面, 应用程序

描述已自动生成

### MCU中设置串口接收中断或任务

上位机发送已经具备条件了，还需要MCU端设置接收的中断或任务，不同的MCU设置不同，这里不做赘述。

### 效果演示

我用的STM32F103的单片机进行演示，功能是Vofa发送PID参数，单片机通过串口接收中断接收到PID值后赋值对应的对象，然后发送PID三个值到上位机，效果如下图形用户界面

描述已自动生成