**一、概述**

与选手PC机上的LabVIEW程序交互的有**两个设备**：连接摄像头的PC机（上位机，以下简称上位机）和STM32单片机。

上位机的主要功能是：识别平台范围，识别当前小球位置，并将当前小球位置的坐标发送给选手PC机的LabVIEW

STM32的主要功能是：根据选手PC机的LabVIEW发送过来的PWM的占空比数据，分别设置控制两个方向舵机的PWM的值，使之转动到目标位置。

因此，整个信息的传递链路如下：

上位机——>选手PC的LabVIEW——>STM32单片机

为了保证选手的代码编写方便，我们采用串口通信方式（异步串行通信），**波特率115200，8位数据位，无校验位，1位停止位，无流控。**

**二、通信协议**

**（一）通信协议设置的目的和意义**

根据前期培训内容，按照字符逐个数字地发送会**占用较长的时间**，如：发送数据350时，按照ASCII码发送需要发送’3’，’0’，’0’三位数据，其优点是比较直观（在串口通信助手可以直接看到），缺点是**数据难以处理（**从字符拼回数据，如果数据长度不一致处理比较困难）、**占用较长发送时间**（这可能导致控制的不及时）。

通信协议的出现可以很好地解决这些问题，因为只需要发送**数值**，可以充分利用8位数据位（十进制数值范围0~255十六进制数值范围0x00~0xFF），对于超过8位的数据（大于255的数），可以将之拆分成高位七位+低位七位（**原因后续说明**），这样可以表示从0~（214-1）的十进制数据。对于本次比赛的收发范围，可以完全满足。

**（二）上位机->LabVIEW的通信协议**

上位机每40ms向LabVIEW发送一次位置信息，一个帧头，一组拆分的坐标信息。

|  |  |
| --- | --- |
| Byte0帧头 | 0xFA |
| Byte1 | (x高七位)+1 |
| Byte2 | (x低七位)+1 |
| Byte3 | (y高七位)+1 |
| Byte4 | (y低七位)+1 |

X的范围是0~400；Y的范围是0~400。当平板上没有识别到小球时，系统会发送X、Y的坐标均为500。

举个栗子：当小球的坐标为（352(0b101100000)，257(0b100000001)）【红色为高七位，绿色为低七位】时，上位机的发送内容为：0xFA 0x03 0x61 0x03 0x02（“0x”表示16进制）

详细计算过程：

Byte0 = 0xFA

Byte1 = ⌊352/128⌋ + 1 =3 =0x03

Byte2 = 352%128 + 1 =97 =0x61

Byte3 = ⌊257/128⌋ + 1 =3 =0x03

Byte4 = 257%128 + 1 =2 =0x02

**（三）LabVIEW->STM32的通信协议**

LabVIEW每接收一次来自上位机的数据，都会根据控制算法计算出PWM波占空比，其控制的延时绝大部分与计算的时间有关。

LabVIEW发送需要遵循以下协议：

|  |  |
| --- | --- |
| Byte0帧头 | 0xFA |
| Byte1 | PWM\_X高七位 |
| Byte2 | PWM\_X低七位 |
| Byte3 | PWM\_Y高七位 |
| Byte4 | PWM\_Y低七位 |
| Byte5帧尾 | 0xFF |

协议主体与“上位机->LabVIEW”类似，PWM\_X和PWM\_Y分别表示x方向、y方向的电机控制PWM占空比，其范围是500-2500，形成对舵机的转角的映射。其计算过程与上一章类似，注意不需要对结果+1（原因后续说明）

**三、推荐的连接方式**

**方法一：**选手可以使用同一台PC机充当上位机和LabVIEW控制端，连接方式为

PC(上位机)—(虚拟串口)—PC(LabVIEW接收端)——LabVIEW—（USB转TTL）—STM32

**方法二：**选手也可以通过和组员的电脑连接，PC\_0充当上位机，PC\_1充当控制端，连接方式为：

PC\_0(上位机)—(USB线/两个USB-TTL对插)—PC\_1(LabVIEW接收端) ——LabVIEW—（USB转TTL）—STM32

**注：**

USB-hub为选手提供了更多的USB口

注意TXD-RXD之间的连接，如两个USB-TTL对插时，需要将USB-TTL\_1的**TXD**连接到USB-TTL\_2的**RXD**连接端，USB-TTL\_1的**RXD**连接到USB-TTL\_2的**TXD**连接端；需要注意，LabVIEW与STM32端连接时，由于STM32端的电路板丝印印刷错误，请将USB-TTL的TXD和电路板的TXD连接、RXD与RXD连接。

注意，如果是跨设备的连接时，请保证信息链路中只有一个设备在供电，同时保证各设备共地。如，USB-TTL对插时，USB-TTL\_1的**GND**连接到USB-TTL\_2的**GND**连接端。

**四、编程指北**

编程流程：

1.**配置串口初始化**（注意创建一个选择串口号的控件！）

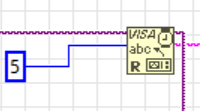
注意波特率、数据位、奇偶校验、停止位和终止符（请设置成F，禁用终止符）

注意，此配置只需要运行一次，切记不可放到while里重复运行



**2.读取串口**

可以**按照心情**读取**5位及以上**的数据（5位就够，读多了似乎也不会有事情…大概），为了防止来不及处理下一次的数据造成堆积，建议使用串口清空控件清空一下缓存区**（先读取，后清空）**

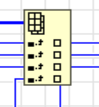


因为需要**连续读取**，所以请将之**放在while循环中！**（注意**不要把配置串口部分**也放进while循环中）

3.读取到字符串后，需要先**转化为字节数组**



4.**索引**每个数组，**拖拽**下方蓝色方框可以索引多个元素

5.**解码**

每个对应的数据取数组索引后数据类型是u8（无符号8位整型），只能表示0~255的数，**执行移位或乘法等操作时务必先转为u16（0~65536）类型**，**否则会溢出**。同样地，**算出PWM值准备发送时，也需要先将其转换为u16类型。**

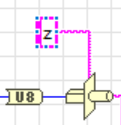


这是**移位**，右移n位就是整除2^n，左移n位就是乘上2^n，比乘法快。当然也可以用地板除。

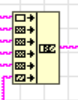


6.(针对向STM32发送)数值->字符串转换

需要使用到如下控件的组合（字符型为8位无符号整型，PWM拆分后保持原来的数据格式，为16位，故需要对按协议拆分后的结果进行长度转换，并将之转化成字符型），‘z’为字符常量，LabVIEW会根据输入的目标类型进行强制转换。



7. (针对向STM32发送)进行发送协议编码后，需要拼接字符串，用到如下控件，与索引数组的控件（4）相同，下拉可以获得更多的拼接。最上面的字符会成为字符串的头。



8. (针对向STM32发送)将拼接好的字符串发送出去，注意，发送之前，也需要在运行VI时**配置一次**发送串口的通信设置（波特率、奇偶校验、停止位等）。同读取一样，为了连续写入，**请将之放在while循环中**（**不要把串口配置部分放进while中**，这是第**n**次提醒了）



**五、Q&A**

Q1：为什么上位机->LabVIEW的通信协议需要对拆分的坐标数据+1？

A1：LabVIEW的读取控件受限，由于是按照字符串形式读取，LabVIEW会在读到0x00时认为已经读完了（因为0x00，即‘\0’，是字符串的结束标识）。因此如果数据中出现0x00的数据时（如，X=256，低七位为0x00），会提前终止读取，导致Y坐标信息丢失，程序出现意想不到的情况，因此编码时+1，避开LabVIEW的奇葩特性。

Q2：为什么采用7位的拆分方式，而不是8位的拆分方式？

A2：书接上文，Q1中提到，需要在编码时+1，为了避免+1导致的可能的溢出，采用7位更加稳妥。同时，高七位+低七位的方式够用了，够用就行。值得注意的是，高/低7位不代表发来的数据只有7位，数据仍然是8位的（这是串口配置的数据帧长度），只不过最高的一位通常是0，如0b0111 0111。(只有当低七位的数据是0b0111 1111时，才可能收到+1编码后的0b1000 0000)

Q3：为什么我读不到上位机的消息？

A3：首先，**检查硬件连接问题**：如果上位机是在队友电脑上运行的，请检查连接是否牢固、USB-TTL对插时，**上位机端的TXD是否插到了LabVIEW端的RXD上**，并且**两个USB-TTL的GND端是否连接（共地）**；

其次，如果选择不到上位机的串口,**请确认是否安装了VISA**。如果安装过了，**请先退出当前运行的可能占用该串口的程序（如串口调试助手），再重启LabVIEW。**

之后，如果已经可以选择串口了，请检查串口**通信配置**。

Q4：找不到控件怎么办？觉得自己太菜任务太难怎么办？

A4：

（1）不会有人真的找不到控件吧，不会吧不会吧！

（2）初赛任务**真的很简单**，如果你们小队完成了作业并顺利通过的话，初赛的任务实际上只是组装+通信模块的编写，PID的子VI基本可以直接使用，有任何问题欢迎及时反馈，学长学姐们也会积极热情地回答大家的问题。此外，**坚持就是胜利！**

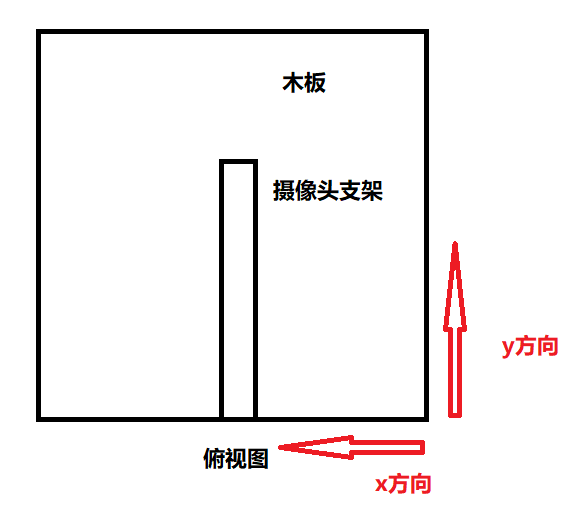
**六、使用TIPS**

**（一）器件组装及电路板连接：**

器件组装过程中，请务必按照教程中的顺序进行，这是**学长们摸了好几个晚上摸出来的较为方便的安装流程**。此外，舵臂安装时，请使用舵机测试仪（如果不会使用请询问学长/学姐），调平零点后，再安装舵臂。

请注意舵机的各条线，橙色为信号线，红色为+，棕黑色为GND

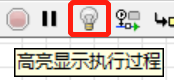
请注意安装方向。沿着摄像头架的方向看去，横向为X轴，纵向为Y轴，**X轴舵机连接PA0、Y轴舵机连接PA6**。



另：所有的装配体都会有空程（即感觉不太牢固），这是无法避免的事情，缓解方法是在空程较大的相应零件的连接处点胶/增加垫片。

**（二）LabVIEW端使用：**

**善用调试**，关于LabVIEW的详细调试方式请上网搜索。比较常用的方法是“小灯泡“+内存探针的方式，高亮显示执行过程中，vi会**以极慢的速度**运行，显示当前数据流图，点击对应的连线，可以查看到其内容，便于检查是否为运算错误、逻辑错误等等。



**分模块**进行编写，并**配以注释**，注释的结构可以在**结构——修饰——自由标签**处找到。

**（三）虚拟串口的使用**

善用百度/CSDN等。

https://blog.csdn.net/qq\_17351161/article/details/89607458