

一、底层发送数据到上位机的通讯协议

底层发送 22 个数据到上位机，发送频率约 200HZ，所以上位机获取频率最好大于或等于 200HZ，数据才不会延迟或者堵塞

发送 22 个数据到上位机

| | | | | | |
|------|------|------|---------|--------------|-----|
| 0XAA | 0XAA | 0XF1 | 88 | (0X_) * 88 | sum |
| 帧头 | 帧头 | 功能字 | 数据长度 88 | 88 字节/22 个数据 | 和校验 |

底层发送数据到上位机通讯代码

1.先把要发送的数据存到一个 float 型数组

```
float Velocity_line_A, Velocity_line_B, Velocity_line_C, Velocity_line_D;
//500线光电4倍频=2000 电机51减速比 编码器读取频率200HZ 轮子周长 0.628m
Velocity_line_A = ( (Velocity_current_A/2000)/51)*200 *0.628;// 单位m/s
Velocity_line_B = ( (Velocity_current_B/2000)/51)*200 *0.628;// 单位m/s
Velocity_line_C = ( (Velocity_current_C/2000)/51)*200 *0.628;// 单位m/s
Velocity_line_D = ( (Velocity_current_D/2000)/51)*200 *0.628;// 单位m/s

/*<01>*/data_u[0] = Start_Flag;//电机启动开关，1启动 0停止
/*<02>*/data_u[1] = Encoder_Position_A/Enco_circle*360.f ;
/*<03>*/data_u[2] = Encoder_Position_B/Enco_circle*360.f ;
/*<04>*/data_u[3] = Encoder_Position_C/Enco_circle*360.f ;
/*<05>*/data_u[4] = Encoder_Position_D/Enco_circle*360.f ;//四轮当前转角
/*<06>*/data_u[5] = Velocity_line_A ;
/*<07>*/data_u[6] = Velocity_line_B ;
/*<08>*/data_u[7] = Velocity_line_C ;
/*<09>*/data_u[8] = Velocity_line_D ; //ABCD四轮的当前线速度 m/s
/*<10>*/data_u[9] = gyro_Roll ;
/*<11>*/data_u[10] = gyro_Pitch ;
/*<12>*/data_u[11] = gyro_Yaw ;//XYZ三轴角速度原始数值
/*<13>*/data_u[12] = accel_x ;
/*<14>*/data_u[13] = accel_y ;
/*<15>*/data_u[14] = accel_z ;//XYZ三轴加速度原始数值
/*<16>*/data_u[15] = Roll ;
/*<17>*/data_u[16] = Pitch ;
/*<18>*/data_u[17] = Yaw ; //XYZ三轴角度
/*<19>*/data_u[18] = Voltage ; //电池电压
/*<20>*/data_u[19] = 0 ;//预留位
/*<21>*/data_u[20] = 0 ;//预留位
/*<22>*/data_u[21] = 0 ;//预留位
```

2. 将 float 型分为 4 个单字节

3. 存放到另一数组中, 然后使用单片机的 DMA 发送数据到串口

```
//*****发送22个数据*****//
float data_u[22];
u8 len=22;
void Send_data_ROS(void)
{
    u8 tbuf[88];
    unsigned char *p;

    for(u8 i=0;i<len;i++){
        p=(unsigned char *)&data_u[i];
        tbuf[4*i+0]=(unsigned char) (* (p+3));
        tbuf[4*i+1]=(unsigned char) (* (p+2));
        tbuf[4*i+2]=(unsigned char) (* (p+1));
        tbuf[4*i+3]=(unsigned char) (* (p+0));
    }

    usart1_sent(0XF1,tbuf,len*4);//自定义帧,功能码0XF1
}
//fun:功能字. 0XA0~0XAF
//data:数据缓存区,最多48字节!!
//len:data区有效数据个数

u8 send_buf1[93];
void usart1_sent(u8 fun,u8*data,u8 len)
{
    u8 i;
    send_buf1[len+4]=0; //校验数置零
    send_buf1[0]=0XAA; //帧头
    send_buf1[1]=0XAA; //帧头
    send_buf1[2]=fun; //功能字
    send_buf1[3]=len; //数据长度
    for(i=0;i<len;i++)send_buf1[4+i]=data[i]; //复制数据
    for(i=0;i<len+4;i++)send_buf1[len+4]+=send_buf1[i]; //计算校验和
}

if(DMA1->ISR&(1<<13))//通道4传输完成标志((通道号-1)*4+1)
{
    Dma_Disable(DMA1_Channel4);//DMA搬运除能,通道2

    Send_data_ROS();//发送22个数据

    //发送数据
    DMA1->IFCR|=1<<13;//清空通道4传输完成标志
    Dma_Enable(DMA1_Channel4,len*4+5);//DMA搬运使能,通道1
}
```

二、上位机发送数据到底层的通讯协议

底层是使用串口中断的形式接收数据,然后在 5 毫秒中断函数里面解算得到数据,所以上位机发送数据的间隔应该在 200HZ 左右,底层处理数据的频率是不变的,上位机发送数据的频率太大不仅无效可能还会因为频繁进入串口中断而影响底层代码的运行

ROS 上位机发送数据到底层的通讯协议

发送 12 个数据到底层

| | | | | | |
|------|------|------|---------|--------------|-----|
| 0XAA | 0XAA | 0XF1 | 48 | (0X_) *48 | sum |
| 帧头 | 帧头 | 功能字 | 数据长度 48 | 48 字节/12 个数据 | 和校验 |

ROS 上位机发送数据到底层的通讯代码

```
/*<01>*/Data_US[0] = Flag_start; //电机启动开关, 1启动 0停止
/*<02>*/Data_US[1] = angle_A;
/*<03>*/Data_US[2] = angle_B ;
/*<04>*/Data_US[3] = angle_C ;
/*<05>*/Data_US[4] = angle_D ; //ABCD四轮的当前转角 deg
/*<06>*/Data_US[5] = speed_A ;
/*<07>*/Data_US[6] = speed_B ;
/*<08>*/Data_US[7] = speed_C ;
/*<09>*/Data_US[8] = speed_D ; //ABCD四轮的当前线速度 m/s
/*<10>*/Data_US[9] = 0 ; //预留位
/*<11>*/Data_US[10] = 0 ; //预留位
/*<12>*/Data_US[11] = 0 ; //预留位

send_data(); //发送指令控制电机运行
```

```

//*****发送12个数据*****//

void send_data(void)
{
    uint8_t len=12;
    uint8_t tbuf[53];

    unsigned char *p;

    for(uint8_t i=0;i<len;i++){
        p=(unsigned char *)&Data_US[i];
        tbuf[4*i+4]=(unsigned char)*(p+3);
        tbuf[4*i+5]=(unsigned char)*(p+2);
        tbuf[4*i+6]=(unsigned char)*(p+1);
        tbuf[4*i+7]=(unsigned char)*(p+0);
    }
    //fun:功能字 0XA0~0XAF
    //data:数据缓存区, 48字节
    //len:data区有效数据个数

    tbuf[len*4+4]=0; //校验位置零
    tbuf[0]=0XAA; //帧头
    tbuf[1]=0XAA; //帧头
    tbuf[2]=0XF1; //功能字
    tbuf[3]=len*4; //数据长度
    for(uint8_t i=0;i<(len*4+4);i++)tbuf[len*4+4]+=tbuf[i]; //计算和校验
    try
    {
        sp.write(tbuf, len*4+5); //发送数据下位机(数组, 字节数)
    }
}

```