

赞同 5
分享

电机PID控制补充篇-野火上位机串口协议介绍

码农爱学习
码农爱学习，专注嵌入式~

已关注

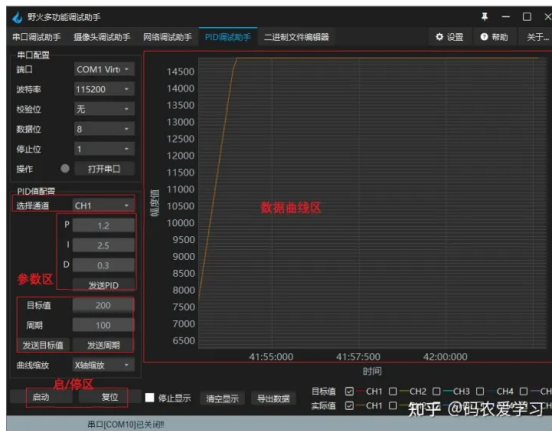
5 人赞同了该文章

0 引言

之前介绍的电机PID控制的系列文章，得到不少电子爱好者的关注，不过也收到一些关于串口通信的疑问反馈。之前的一系列文章，确实没有着重介绍串口软件的使用细节以及通信协议的具体格式。本篇就来补充一下野火PID调试助手的串口协议。

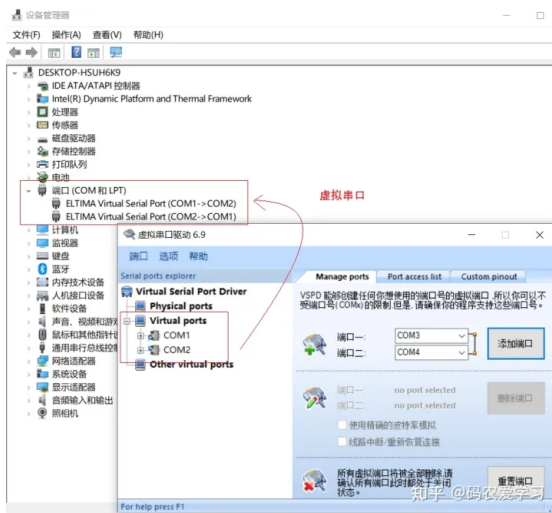
下野火PID调试助手的使用界面如下，与串口通信协议相关的，主要分为三个部分：

- 参数区：包括数据通道选择、PID参数设置与展示、目标值、周期值的设置与展示
- 启/停区：控制电机的启动、停止以及程序的复位
- 数据曲线区：接收板子上发上来的数据，进行速度曲线或位置曲线等的绘制



在介绍串口协议之前，推荐安装一个虚拟串口驱动软件来在自己的电脑上虚拟出两个串口，方便分析串口数据。

比如我们在电脑上产生两个虚拟串口，这两个串口可以看作是通过线连接了起来，比如我们使用野火PID助手和另一个串口助手软件分别连接这两个虚拟串口，当野火PID助手发送数据时（比如按下启动按钮），另一端的串口助手软件就可以收到并显示野火PID助手发来的数据，这样我们就能分析野火PID助手下发数据的数据格式了。



1 PID调试助手串口协议介绍

先来看一下野火PID调试助手的串口协议数据格式。

串口协议的定义参考野火论坛的介绍

1.1 指令包格式

字节数	说明	十六进制
1bytes	数据通道	0xXX
4bytes	包长度	0xXXXXXXX
1bytes	指令	0xXX
1bytes	参数1	0xXX
...
1bytes	参数n	0xXX
1bytes	校验和	0xXX

说明:

1. 所有多字节的**低字节**在前 (关于高低字节或数据大小端的介绍, 可参考:

码农爱学习: C语言打印数据的二进制格式-原理解析与编程实现
11 赞同 · 4 评论 文章



)

1. 包头固定为四字节的**0x59485A53**
2. 通道地址1到5对应软件上的CH1到CH5, **CH1为0x01**, CH2为0x02
3. 包长度为从包头到校验的**所有数据长度**
4. 指令为相应的**功能码**
5. 参数为指令**需要参数时加入**
6. 校验为校验和方式——8位

1.2 指令分类

上述数据格式中, 需要重点关注的是“指令”这一字段, 它表明了这一包数据的具体含义。

另外, 串口数据包括**下发数据**和**上传数据**, 下发数据就是野火PID助手按照协议包格式向板子发送串口数据, 上传数据就是板子按照协议包格式向野火PID助手送串口数据。

1.2.1 下发数据

野火PID助手**下发**设定的数据或指令到板子中

功能	指令	参数
PID	0x10	3个float
目标值	0x11	1个int
启动	0x12	无
停止	0x13	无
复位	0x14	无
周期	0x15	1个uint

1.2.2 上传数据

板子**上传**数据或指令到野火PID助手

功能	指令	参数
目标值	0x01	1个int
实际值	0x02	1个int
PID	0x03	3个float
启动	0x04	无
停止	0x05	无
周期	0x06	1个uint

2 数据实测与分析

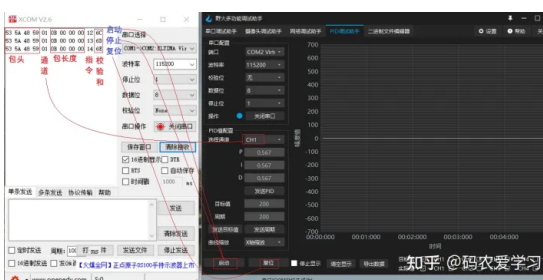
2.1 实测下发的数据

测试数据的下发, 可以只使用一个电脑进行测试, 通过虚拟串口, 分别连接野火PID助手和另一个串口助手软件, **通过野火PID助手向另一个串口助手发送数据**, 以Hex方式显示接收的数据, 并观察数据的格式。

2.1.1 下发启动(0x12) 停止(0x13) 复位(0x14)

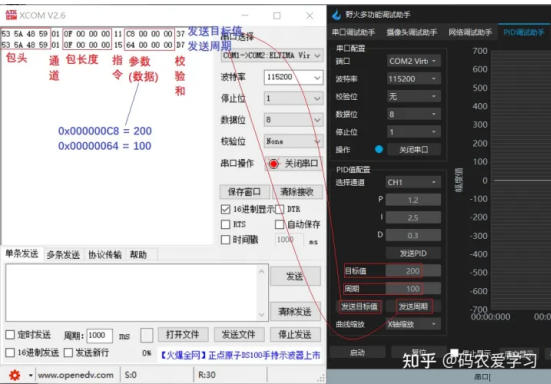
这3个是指令, 没有数据参数, 数据长度为0x0B, 也就是11个byte

	包头	通道	包长度	指令	校验
启动	53 5A 48 59	01	0B 00 00 00	12	6C
停止	53 5A 48 59	01	0B 00 00 00	13	6D
复位	53 5A 48 59	01	0B 00 00 00	14	6E



这2个指令，带有1byte数据，数据长度为0x0F，也就是15个byte

	包头	通道	包长度	指令	参数 (数据)	校验
目标值	53 5A 48 59	01	0F 00 00 00	11	C8 00 00 00	37
周期	53 5A 48 59	01	0F 00 00 00	15	64 00 00 00	D7



2.1.3 下发送PID(0x10)

这个指令，带有12byte数据(3个float)，数据长度为0x17，也就是23个byte

包头	通道	包长度	指令	参数 (数据)	校验
53 5A 48 59	01	17 00 00 00	10	9A 99 99 3F 00 00 20 40 9A 99 99 3E	EB



注意：这里PID的数据是float型的，在发送时是需要拆分成4个字节的Hex格式发送的，关于float类型数据转为Hex格式的介绍，可参考：

码农爱学习：C语言将float拆分为4个hex
传输与重组
6 赞同 · 7 评论 文章

2.2 实测上传数据

测试数据的上传，需要将程序下载到板子中，板子通过软件连接电脑上的任意串口调试软件，以Hex方式显示接收的数据，分析数据的格式。

2.2.1 上传启动(0x04)/停止(0x05)状态

	包头	通道	包长度	指令	校验
启动	53 5A 48 59	01	0B 00 00 00	04	5E
停止	53 5A 48 59	01	0B 00 00 00	05	5F

2.2.2 上传目标值(0x01)/周期值(0x06)

	包头	通道	包长度	指令	参数 (数据)	校验
目标值	53 5A 48 59	01	0F 00 00 00	01	70 3A 00 00	09
周期	53 5A 48 59	01	0F 00 00 00	06	XX XX XX XX	XX

2.2.3 上传PID参数(0x03)

包头	通道	包长度	指令	参数 (数据)	校验
53 5A 48 59	01	17 00 00 00	03	CD CC 4C 3D 00 00 00 00 00 00 00 00	8B



2.2.4 上传实际值(0x02)

这里就是将电机的转速和位置值上传到野火PID助手中，用于显示位置曲线或速度曲线。

包头	通道	包长度	指令	参数 (数据)	校验
53 5A 48 59	01	0F 00 00 00	02	2D 00 00 00	8D

3 串口收发程序

再来看一下与串口数据发送与接收相关的主要代码：

3.1 发送

按格式组包数据:

```

/**
 * @brief 设置上位机的值
 * @param cmd: 命令
 * @param ch: 曲线通道
 * @param data: 参数指针
 * @param num: 参数个数
 * @retval 无
 */
void set_computer_value(uint8_t cmd, uint8_t ch, void *data, uint8_t num)
{
    static packet_head_t set_packet;

    uint8_t sum = 0;    // 校验和
    num *= 4;           // 一个参数 4 个字节

    set_packet.head = FRAME_HEADER;    // 包头 0x5948A5A3
    set_packet.ch = ch;                // 设置通道
    set_packet.len = 0x08 + num;       // 包长
    set_packet.cmd = cmd;              // 设置命令

    sum = check_sum(0, (uint8_t *)&set_packet, sizeof(set_packet));    // 计算包头校验和
    sum = check_sum(sum, (uint8_t *)data, num);                        // 计算参数校验和

    usart1_send((uint8_t *)&set_packet, sizeof(set_packet));    // 发送数据头
    usart1_send((uint8_t *)data, num);                          // 发送参数
    usart1_send((uint8_t *)∑, sizeof(sum));                  // 发送校验和
}

```

将一包数据一个个发送给上位机。

```
void uart1_send(u8*data, u8 len)
{
    u8 i;
    for(i=0;i<len;i++)
    {
        while(USART_GetFlagStatus(USART1, USART_FLAG_TC)==RESET);
        USART_SendData(USART1,data[i]);
    }
}
```

3.2 接收

使用中断的方式进行串口数据的接收:

```
//=====
uint8_t Recv1[128]={0};//串口接收缓存
u8 rx_cnt=0;//接收数据个数计数变量
int sizecopy=128;

void USART1_IRQHandler(void)
{
    uint8_t data;//接收数据暂存变量
    uint8_t bufcopy[128];//最多只取前64个数据

    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET) //接收中断
    {
        data = USART_ReceiveData(USART1);
        Recv1[rx_cnt++]=data;//接收的数据存入接收数组
        USART_ClearITPendingBit(USART1,USART_IT_RXNE);
    }

    if(USART_GetITStatus(USART1, USART_IT_IDLE) != RESET)//空闲中断
    {
        data = USART1->SR;//串口空闲中断的中断标志只能通过先读SR寄存器，再读DR寄存器清除！
        data = USART1->DR;

        //清空本地接收数组
        memset(bufcopy,0,sizecopy);
        memcpy(bufcopy,Recv1,rx_cnt);//有几个复制几个
        protocol_data_recv(bufcopy, rx_cnt);
        memset(Recv1,0,sizecopy);
        rx_cnt=0;
    }
}
```

接收之后的数据保存在串口接收缓存Recv1数组中，然后再根据协议格式解析数据即可。

4 总结

本篇介绍的野火PID助手的串口协议格式，包括下发的数据格式和上传的数据格式，并通过实际获取串口数据的Hex格式数据，与协议的定义进行对比分析，使得能够更加的理解串口数据的格式。

如果遇到野火PID助手下发指令板子没反应，或板子上传数据PID数据无法显示曲线，这时就要先**排查一下串口数据的格式是否正确**，若不正确，就要看下自己程序中的串口收发函数编写的是否正确，只有串口数据符合了规定的协议格式，才能正确的进行数据通信。

如果串口数据格式正常，电机还不转，就要**排查硬件接线是否正常**以及是否要**根据自己的电机编码器参数来修改程序中的参数**。

编辑于 2022-03-21 23:29

pid控制 串口通信 协议



评论千万条，友善第一条

1 条评论

默认 最新



杨印
兄弟，代码可以发一下吗
2022-11-05

回复 赞

文章被以下专栏收录



码农爱学习
学习分享

推荐阅读



单片机也能 OO? —— 串口命令解析器的实现

邵国际

单片机 | 一种串口高效收发数据的实现方法

摘要：本文在探讨传统数据收发不足之后，介绍如何使用带FIFO的串口来减少接收中断次数，通过一种自定义通讯协议格式，给出帧打包方法；之后介绍一种特殊的串口数据发送方法，可避免使用串...

AI电堂

单片机 | 一种串口高效收发数据的实现方法

摘要：本文在探讨传统数据收发不足之后，介绍如何使用带FIFO的串口来减少接收中断次数，通过一种自定义通讯协议格式，给出帧打包方法；之后介绍一种特殊的串口数据发送方法，可避免使用串...

AI电堂

暴揍单片机 9 -- 串口实操纪实

前面两个章节全部讲解了单片机串口的相关理论基础，本章节突出讲解一下单片机串口的实际操作。这篇文章才发现，写的有点多!!! 但是这里面的东西确实值得大家学习，有个思想特别重...

turke...

发表于暴揍单片机