

东莞市微宏智能科技有限公司

网址: www.wheeltec.net

STM32 驱动激光雷达例程

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
V1.0	2020/07/08	第一次发布

序言

本文档主要讲解如何用 STM32F103C8T6 驱动激光雷达开始扫描并接收数据，经过简单处理之后，在串口调试助手进行打印。

目录

序言.....	2
1. STM32F103C8T6 驱动激光雷达.....	4
1.1 硬件接线说明.....	4
1.2 激光雷达驱动程序说明.....	5

1. STM32F103C8T6 驱动激光雷达

STM32 驱动激光雷达可以通过单击用户按键来控制雷达的启停，在雷达开始转动之后，可以通过电脑端的串口调试助手给激光雷达发送指令来控制激光雷达，通过更改 PWM 设置也可以改变激光雷达的转速。

1.1 硬件接线说明

激光雷达一共有 6 个引脚，其中 2 个 GND，2 个 5V，一对 RX 和 TX 引脚，按照下表 1-1 的接线要求接在 STM32 单片机上。其中 2 个 5V 因为电流要求，可以接在 TTL 转串口的 5V 引脚或者 ST Link 的 5V 引脚上。激光雷达的立贴座端子不能直接与 STM32 连接，需将引脚用排线接出。

开发套装中 RPLIDAR A1 模块底部电机接口和核心接口分别采用 PH1.25-3P 和 PH1.25-4P 的立贴座。用户可以使用带有 1.25 间距 3pin 和 1.25 间距 4pin 端子的线缆与其连接。其信号定义如下：

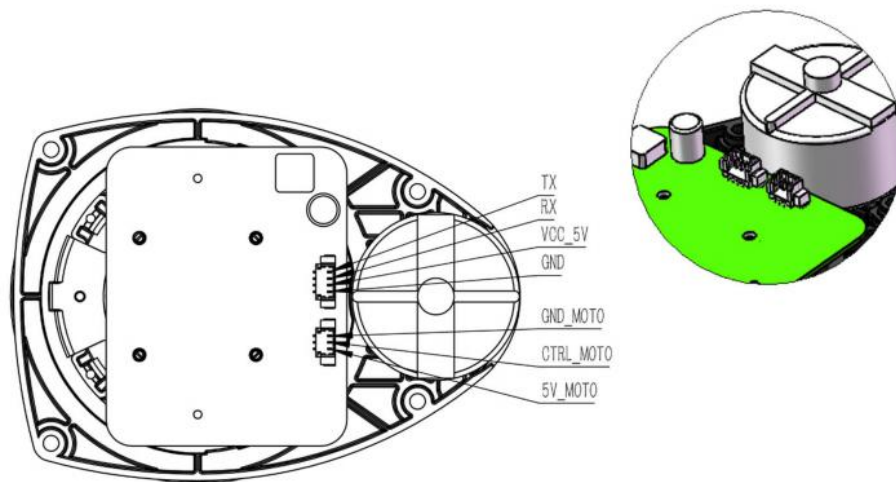


图 1-1-1 激光雷达引脚图

STM32 除了需要和激光雷达连接之外，还需要将 STM32 的 Micro USB 串口接在电脑上，方便程序下载和调试。注意：需要将 ST Link 与 STM32 的 USB 串口接在同一设备或者 HUB。

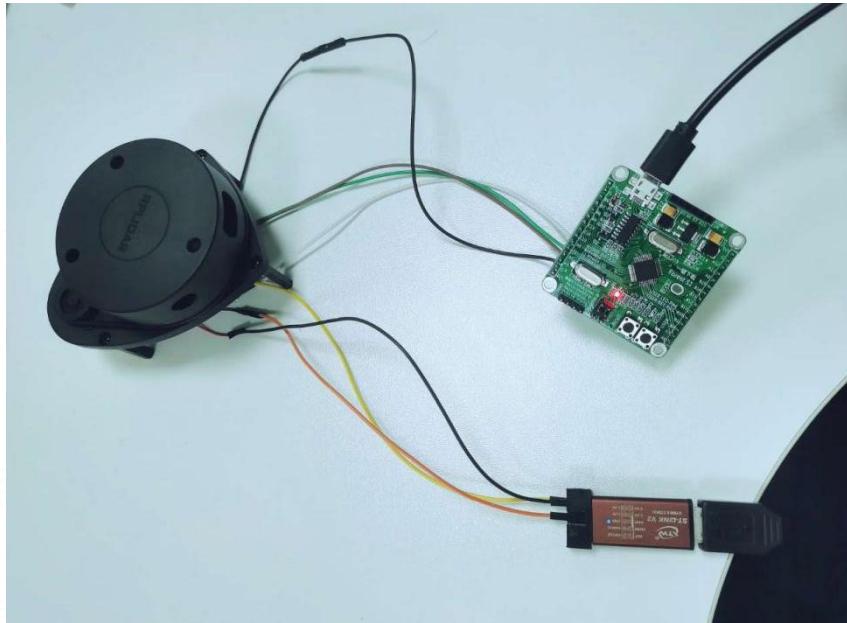


图 1-1-2 硬件接线图

接线要求如表 1-1 所示。

表 1-1 激光雷达接线说明

雷达引脚	STM32
TX	A3
RX	A2
VCC_5V	外接 5V(雷达供电)
GND	GND
GND_MOTO（电机地线）	外接 GND
CTL_MOTO	A0
5V_MOTO	外接 5V（电机供电 5-10V）

1. 2激光雷达驱动程序说明

① 串口通信实现

想要实现激光雷达数据的接收和发送，就需要将雷达接在串口上，因为串口 1 需要用来与电脑通信，所以这里选用串口 2，通过查找芯片手册可以找到串口 2 的 TX 引脚为 A2，RX 引脚为 A3。串口 1 和串口 2 的配置方法请参考实验串口通信例程。

注意：这里的串口 2 并不需要使用接收中断，因为雷达数据传输速度非常快，如果使用中断的话很容易卡死在中断里。

在串口 1 的接收中断里，我们把接收到的数据通过串口 2 发送出去。

```
void USART1_IRQHandler(void)           //串口 1 中断服务程序
{
    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)
        //接收中断(接收到的数据必须是 0x0d 0x0a 结尾)
        {
            Res =USART_ReceiveData(USART1); //读取接收到的数据
            USART_SendData(USART2, Res);    //发送到串口 2
            while(USART_GetFlagStatus(USART2, USART_FLAG_TC) !=SET);
            //等待发送完成
        }
}
```

在串口调试助手与 STM32 单片机的通信中，串口通信的波特率 115200、停止位 1、数据位 8、校验位无，接收使用 16 进制。通过串口调试助手发送指令来控制激光雷达时，需要勾选 16 进制发送，取消发送新行。

开始扫描采样(SCAN)命令请求与回应数据格式

注意：RPLIDAR A2 以及其他支持 4khz 采样频率的设备在该命令下将降低自身采样频率，请使用 EXPRESS_SCAN 获得最佳性能。



图 1-2-1 激光雷达开始扫描指令

停止扫描 (STOP) 命令请求

请求报文: A5 25

在外部系统发送了请求命令字段为停止扫描(STOP, 0x25)的请求报文后，RPLIDAR 将退出正在进行的扫描采样状态，关闭测距系统和激光器，进入空闲模式。如果 RPLIDAR 先前已经工作在空闲状态或者保护停机状态下，则该命令则会被忽略。

RPLIDAR 不会为该请求发送回应报文。建议外部系统需要在发送该请求命令后，延迟 1ms 以上时间后发送下一次请求。

图 1-2-2 激光雷达停止扫描指令

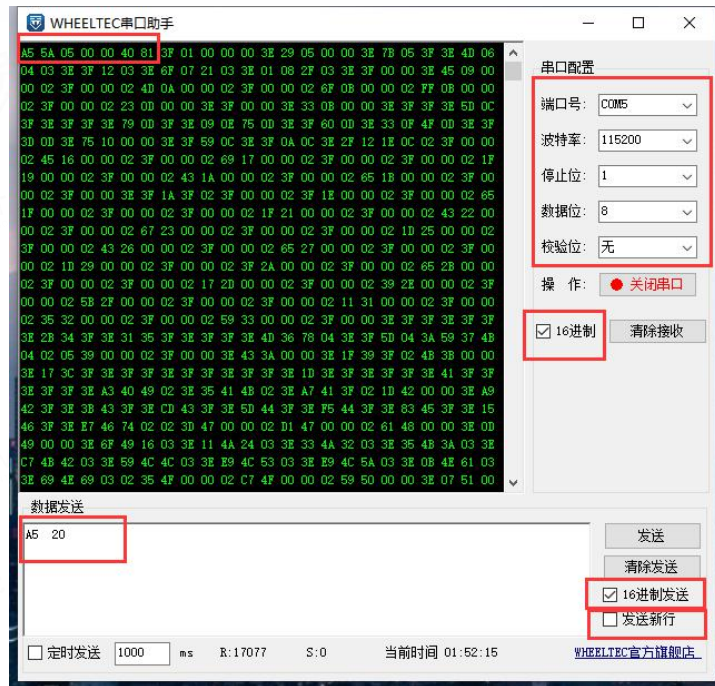


图 1-2-3 串口调试助手接收原始数据

② 激光雷达数据处理

因为不用在串口 2 接收中断里接收数据，所以我们只需要判断 USART_SR 状态寄存器的 RXNE（读数据寄存器非空）位就可以了。接收到数据后，我们先判断有效性，判断激光雷达发送数据第一位的质量是否满足要求（直接过滤掉质量低的点），即判断质量是否为 0x3E，如果满足要求再将之后的 4 位数据接收进行处理。

25.6 USART寄存器描述

有关寄存器描述里所使用的缩写，请参考第1.1节。

可以用半字(16位)或字(32位)的方式操作这些外设寄存器。

25.6.1 状态寄存器(USART_SR)

地址偏移: 0x00

复位值: 0x00C0



图 1-2-4 USART_SR 寄存器

```
if (USART_GetFlagStatus(USART2, USART_FLAG_RXNE)) // 串口寄存器接收到数据
{
    Res =USART_ReceiveData(USART2); //读取接收到的数据
    if(flag)
```

```

{
    i++;
    if(i>4)
    {
        flag=0;
        i=0;
    }
    if(i==1) if(!(Res&0x01)) flag=0, i=0; //第二位数据 C 的奇偶校验
    if(i==2) printf("%d:", Res<<1); //打印角度数据 (°)
    if(i==3) distan=Res>>2;
    if(i==4)
    {
        result=Res;
        result=result<<6|distan; //整合距离数据
        printf("%d\r\n", result); // 打印距离数据 (mm)
    }
}

if(Res==0x3E) flag=1; //有效性数据判断
}

```

如果只是单纯想要接收数据的话可以将数据处理部分改为将接收到的数据发送到串口 1。

处理数据的方式主要是通过移位操作来完成对数据的整合和解算，再将数据通过 `printf` 的方式打印到串口 1 进行输出。因为激光雷达数据的校验位很少，通过判断质量的方式来解算数据非常简单有效。

数据应答报文格式

RPLIDAR 使用如下的数据应答报文结构：



图表 4-4 RESET 请求的通讯时序

图 1-2-5 激光雷达数据结构

下图为串口调试助手接收 STM32 处理之后的激光雷达数据，这里要取消 16 进制的接收模式，数据格式为：角度 (°)：距离 (mm)。

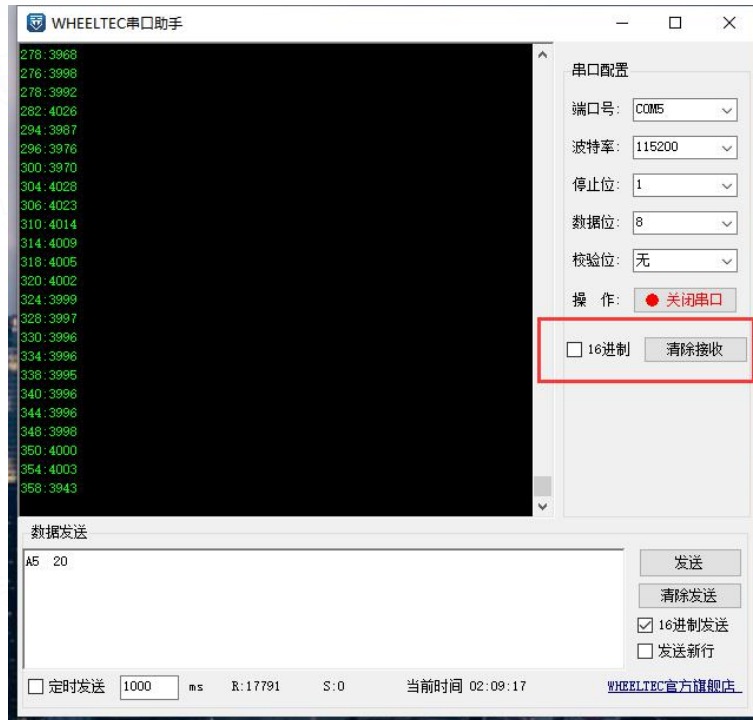


图 1-2-6 激光雷达数据处理输出

③ PWM 控制激光雷达启停

PWM 配置部分请参考 PWM 输出配置例程，为了让激光雷达在按键按下之后开始转动，我们在初始化完成后将 PWM 的输出调整到低电平，不让电机转动，直到按键按下再设置 PWM 为所需要的输出，同时通过串口 2 向激光雷达发送开始扫描的指令，按键结束也是同样的方法控制电机停止转动，并发送停止扫描的指令。

初始化 PWM:

```
TIM2_PWM_Init(1000, 71); //pwm 初始化
TIM_SetCompare1(TIM2, 0); //pwm 设置输出低电平
```

按键按下 PWM 设置和指令发送:

```
TIM_SetCompare1(TIM2, 1000); //电机开始转动, Compare1 最大值
for (i=0; i<2; i++) //发送开始扫描指令
{
    USART_SendData(USART2, scan[i]);
    while(USART_GetFlagStatus(USART2, USART_FLAG_TC) != SET);
}
```

结束后 PWM 设置和指令发送:

```
TIM_SetCompare1(TIM2, 0); //设置 pwm, 电机停止转动
for (i=0; i<2; i++) //发送停止命令
{
```

```
USART_SendData(USART2, stop[i]);  
while(USART_GetFlagStatus(USART2, USART_FLAG_TC) != SET);  
}
```