Talos**嵌入式寒假考核**

**任务一：单片机程序报告**

目录

[一：引言 2](#_Toc16240)

[二：GPIO引脚输出 2](#_Toc32567)

[2.1 LED不同输出模式 2](#_Toc24861)

[2.2 蜂鸣器不同频率发声 3](#_Toc8355)

[三：Mobaxterm串口通信 3](#_Toc6043)

[3.1 Mobaxterm模拟终端 3](#_Toc8839)

[3.2 终端命令响应 5](#_Toc16633)

[四：测试和结果分析 5](#_Toc15350)

[4.1 多次测试分析结果 5](#_Toc19508)

[五：未来优化方案 6](#_Toc19271)

[5.1 未实现的功能 6](#_Toc25686)

[5.2 待优化代码 6](#_Toc9539)

[六：总结 6](#_Toc4443)

[6.1 任务总结 6](#_Toc1767)

# 一：引言

本次寒假考核旨在提升嵌入式成员在嵌入式领域的实践能力，任务一聚焦于单片机程序编写，涉及 IO 口控制、定时器运用、串口通信以及多种功能实现，并且加入了终端的知识，增加同学们对Linxu操作系统和对上位机概念的理解。通过完成此任务，能够深入理解嵌入式系统中硬件与软件的协同工作，掌握单片机基本外设的操作方法，熟练STM32各种常见指令的应用，为后续在嵌入式领域的学习和实践奠定坚实基础。

任务内容涉及LED和蜂鸣器的控制，以及串口通信，还有STM32定时器的PWM输出。最主要的逻辑构成还是在Mobaxterm中通过串口与上位机交互，难点也是在串口通信中对各种输入进行接收和判断。

# 二：GPIO引脚输出

## 2.1 LED不同输出模式

此次任务中，LED共有四种输出模式，常开，常闭，闪烁和呼吸。为实现LED呼吸灯功能，需要把LED对应的GPIO引脚设置为定时器的PWM输出模式。其中PWM输出模式可以在CubeMX中更改预分频系数，时钟总线频率和自动重装载值来修改输出的占空比，在程序中可以通过特定函数修改自动重装载值来修改占空比，控制引脚输出功率，从而控制LED的亮度，实现呼吸灯的效果。

LED的常亮模式只需要把占空比设置为某一固定值即可实现，LED的常灭模式则需要把占空比设置为0，使得对应GPIO引脚的输出功率降为0。LED闪烁而是让LED 在常亮和常灭两种模式下循环，但由于单片机运行速度太快，肉眼难以观察到闪烁的效果，所以需要在每次循环中添加延时函数，方便肉眼观察。LED呼吸是让对应GPIO 引脚的输出功率递增到最大值然后递减到零，并以此循环，实现LED从暗变亮再变暗的过程，模拟呼吸的效果。如果要让LED多次呼吸，可以在循环中嵌套循环，外层循环负责循环次数的控制，内层循环负责输出功率的递增和递减实现。同时设置一个循环标志位，以便实现打断闪烁或者呼吸的功能。



图1 定时器2预分频系数及自动重装载值设置

通道1为LED输出，通道3为无源蜂鸣器输出

# 2.2 蜂鸣器不同频率发声

蜂鸣器可以分为有源蜂鸣器和无源蜂鸣器，其中无源蜂鸣器可以实现不同频率的发声。实现方式也是通过定时器PWM输出。无源蜂鸣器没有内部振荡源，发声的基本原理是电磁感应，其内部的基本结构是一匝匝线圈以及一个微型扬声器。由于没有振荡源，且内部的主要结构是线圈，所以其工作时输入的电信号必须是交流，而且频率不能太低。所以可以通过设置预分频系数和自动重装载值使蜂鸣器发出不同频率的声音。与音调相关知识相结合，根据发声频率设置合适的预分频系数和自动重装载值。

# 三：Mobaxterm串口通信

## 3.1 Mobaxterm模拟终端

在Mobaxterm中，有多种通信协议和连接方式，常用的有SSH，shell和serial串口等通信协议，本次任务选择串口通信。串口连接时需要设置对应的波特率，校验位和停止位等信息。Mobaxterm串口通信与普通的串口调试助手不太相同，Mobaxterm发送的信息在按下键盘时就会发送出去，并不需要按下回车键，所以发送数据都是单字节发送到单片机，所以接收的时候可以使用单字节逐帧接收，接收完成后再判断，并且以回车键作为发送完成标志。并且需要在初始化过程中，使能串口接收和发送中断，以便及时处理串口数据。

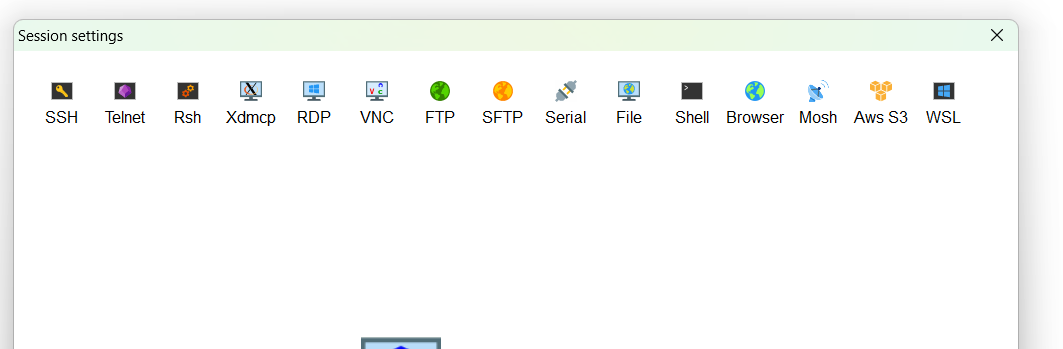


图2 Mobaxterm常用通信协议

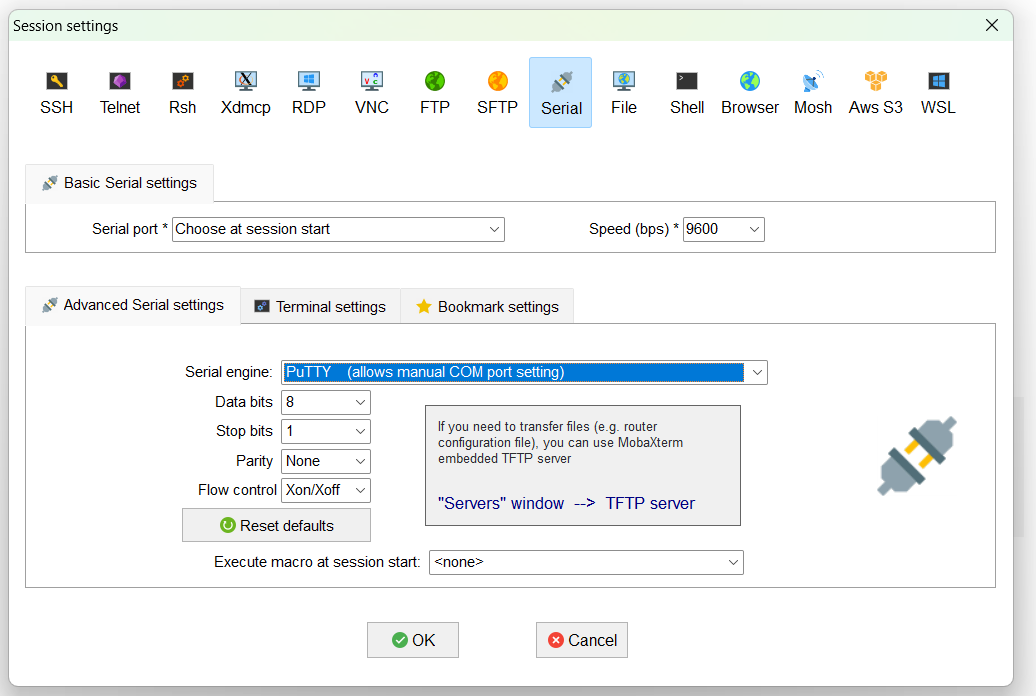


图3 串口参数设置

模拟终端需要有基本的符合认知习惯的输入方式。例如退格删除和清屏，以及回车等基本效果。退格清除和换行等操作可以通过ANSI转义序列实现，例如实现换行操作可以通过向终端发送“\r\n”。每个按键都有其对应的ASCII 码，方便在程序内判断按下的按键，判断用户输入的内容。判断用户输入的部分可以在串口中断回调函数里实现，判断不同的用户输入，检验是否与指定的输入相同，相同则跳转到指定模式下运行。

因为回车键代表输入完成，所以在每次按下回车键后清空当前缓存区的数据，开始新的接收。实现多次接收数据互不干扰。从单片机发送到电脑的信息会直接显示在终端上，为使得键盘输入在终端上显示，可以在向单片机发送信息的同时，让单片机向终端发送相同信息，模拟输入效果。

## 3.2 终端命令响应

模拟终端类似于一个单片机的外壳，内部时单片机的硬件，外部是用户，所以模拟终端相当于用户与单片机交互的媒介，用户通过模拟终端输入指令，单片机做出相应的动作。对于多种命令，可以分类为不同的模式和该模式下的子模式，使得代码更加简洁。在串口中断回调中判断输入的命令是否符合某一模式的要求，若符合就在主循环中调用该模式，在LED有限次闪烁或呼吸时，注意闪烁或呼吸完成后用break跳出循环。

用户输入clear命令时，清空终端显示，可以使用 ANSI 转义序列中的清除屏幕指令来清除终端显示内容，进而间接清除 ANSI 转义序列的显示效果常见的ANSI转义序列命令是"\033[2J\033[H"。ANSI 转义序列可以用于控制终端的显示和操作，其中 \033[D 可以将光标向左移动一个位置，配合 \033[K 可以清除从光标位置到行尾的所有内容，组合使用这两个序列可以模拟退格删除的效果。

# 四：测试和结果分析

## 4.1 多次测试分析结果

在基本功能测试中，多次开机测试 LED 的闪烁和蜂鸣器的发声情况，检查 LED 是否准确闪亮 5 次后熄灭，蜂鸣器是否发出 440Hz 声音 0.5 秒。通过串口输入各种字符、退格键以及 Clear 命令，验证终端的显示和处理功能是否正常。命令响应测试中，针对每个 LED 控制命令（ledOFF、ledON、ledblink、ledbreath、ledctl），分别测试不同参数设置下的功能实现情况。例如，对于 ledctl 命令，测试不同的闪烁模式、闪烁次数和间隔时间组合；

经过测试，各项功能均能正常实现。在基本功能测试中，LED 和蜂鸣器的控制准确无误，终端对输入字符、退格键和 Clear 命令的处理符合预期，未出现异常情况。多次测试不同模式下转换是否能够有正确的响应，测试过程中，不同组合的输入均符合预期。代码测试的结果较为符合设计预期。

# 五：未来优化方案

## 5.1 未实现的功能

本次任务中，未能实现IMU相关功能，包括摇晃检测和方向检测。摇晃检测可以通过判断板子的加速度反馈变换实现，当检测到板子加速度变化较大时即代表板子被快速摇晃。并与无源蜂鸣器和串口通信配合，摇晃发生时串口发送相关数据，蜂鸣器发声。

Flash存储功能也未能实现，由于 Flash 写入数据时只能将 1 变为 0，若要将 0 变为 1，必须先擦除相应扇区。擦除操作以扇区为单位，不能单独擦除一个字节或字。Flash存储数据先擦除要存储的扇区，然后检验擦除是否成功，若成功则存入数据，再读取数据，检验数据存储是否正确。

## 5.2 待优化代码

程序中的串口中断回调函数有些冗长，可以把判断数据是否符合某模式的部分写到函数中，使得代码更加简洁易懂。还可以新建LED子文件，用来存储LED闪烁和呼吸相关的函数，减少主函数的长度，增加可读性和条理性。同时，可以进一步完善与上位机的交互功能，开发图形化界面，方便用户操作和数据可视化展示。

# 六：总结

## 6.1 任务总结

本次寒假考核任务一的完成，使我在嵌入式开发领域取得了长足的进步。通过深入学习和实践，我熟练掌握了单片机 IO 口控制、定时器操作、串口通信和上位机交互等关键技术，能够运用这些技术实现复杂的功能需求。

在软件设计过程中，采用模块化编程思想，提高了代码的可读性、可维护性和可扩展性。通过精心设计和优化代码，确保了程序的稳定性和可靠性，能够在各种复杂环境下正常运行。同时，通过完成本次任务，我也培养了自己解决实际问题的能力，在遇到软件错误时，能够运用所学知识进行分析和解决。