

EE351实验报告

微机原理与微系统

12212635 韩骐骏

EE351

目录

1. EE351实验报告	3
1.1 实验报告列表	3
2. 实验报告列表	4
2.1 树莓派开发环境搭建	4
2.2 双色灯实验	8
2.3 轻触开关实验	0
2.4 PCF8591模数转换器实验	0
2.5 模拟温度传感器实验	0
2.6 Lab6实验报告：超声波传感器测距实验	0
2.7 蜂鸣器实验	0
2.8 PS2操纵杆实验	0
2.9 红外遥控实验	0
2.10 中断实验	0

1. EE351实验报告

此处是南方科技大学EE351“微机原理与微系统”24Fall课程实验报告的主页，你可以在[这里](#)找到所有的实验报告。



本课程所有实验在树莓派4B上进行，使用的操作系统为RaspberryPi OS-64-bit-desktop。

本实验用到的硬件设备包括：

- 树莓派4 model B
- PCF8591模数转换器
- 传感器模块（如温度传感器、超声波传感器等）
- LED灯、蜂鸣器、电位器等
- PS2操纵杆、红外遥控器等
- 面包板、杜邦线等

本实验用到的软件工具包括：

- [Rpi.GPIO](#)库（Python）
- [wiringPi](#)库（C/C++）
- [python-smbus](#)库（I2C通信）

1.1 实验报告列表

- [实验一：树莓派开发环境搭建](#)
- [实验二：双色灯实验](#)
- [实验三：轻触开关实验](#)
- [实验四：PCF8591模数转换器实验](#)
- [实验五：模拟温湿度传感器实验](#)
- [实验六：超声波传感器实验](#)
- [实验七：蜂鸣器实验](#)
- [实验八：PS2操纵杆实验](#)
- [实验九：红外遥控实验](#)
- [实验十：中断实验](#)

作者: 12212635 韩骏骏



本文档使用[mkdocs](#)生成静态网页，使用[mkdocs-with-pdf](#)生成PDF文档。访问[网页版](#)，获得更好的阅读体验。

2. 实验报告列表

2.1 树莓派开发环境搭建

Lab1实验报告：Raspberry Pi开发环境搭建

一、实验介绍

本次实验将配置后续实验用到的软硬件环境，包括操作系统、网络、远程连接等。

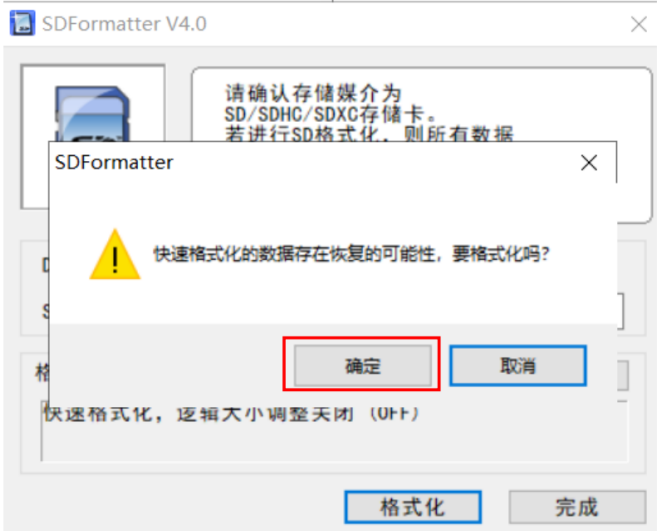
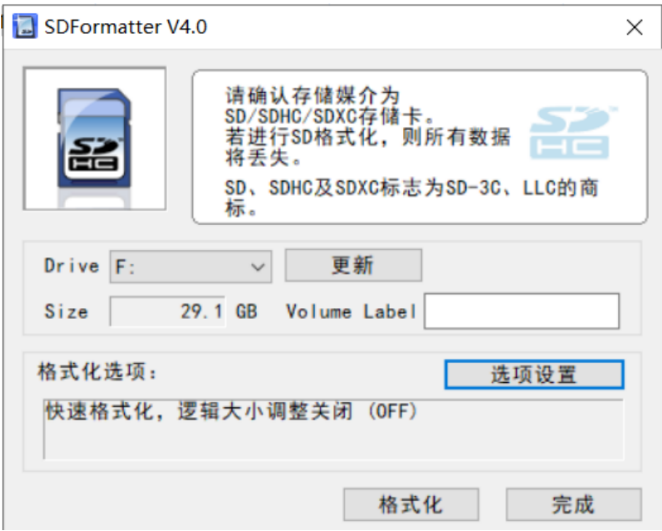
二、实验目标

- 1. 熟悉Raspberry Pi硬件组成及其引脚布局。
- 2. 完成Raspberry Pi OS镜像的下载与烧录。
- 3. 配置Wi-Fi，确保能够访问互联网。

三、实验步骤

(1) 硬件准备与检查

- 1. 确认所需材料：
- 2. Raspberry Pi 4 Model B
- 3. microSD卡
- 4. 电源适配器（最好使用官方提供的USB-C电源,否则可能出问题）
- 5. HDMI显示器及HDMI线缆
- 6. 键盘和鼠标
- 7. 检查硬件状态：
- 8. 插入microSD卡到Raspberry Pi的卡槽中。
- 如果是使用过的卡，先用SDFormatter工具格式化。



- 9. 连接显示器、键盘和鼠标（如果打算使用GUI）。
- 10. 将电源线插入Raspberry Pi，并确保另一端连接到合适的电源插座上。

(2) 操作系统安装

- 1. 下载Raspberry Pi Imager工具：

- 访问[Raspberry Pi官方网站](#)下载Imager工具。

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Ubuntu for x86](#)

[Download for Windows](#)

[Download for macOS](#)

To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`
 in a Terminal window.



- 选择并写入OS镜像：

- 打开Raspberry Pi Imager，点击“CHOOSE OS”按钮，选择推荐的Raspberry Pi OS (32-bit)版本。
- 点击“CHOOSE STORAGE”，挑选之前准备好的microSD卡作为存储介质。
- 确认无误后，点击“WRITE”开始烧录过程。请耐心等待，直到提示写入完成。
- 配置屏幕参数：
- 在microSD卡的根目录下，找到 `config.txt` 文件，编辑并添加以下内容：

```
1 max_usb_current=1
2 hdmi_force_hotplug=1
3 config_hdmi_boost=7
4 hdmi_group=2
5 hdmi_mode=1
6 hdmi_mode=87
7 hdmi_drive=1
8 display_rotate=0
9 hdmi_cvt 1024 600 60 6 0 0 0
```

Warning

在 `hdmi_cvt 1024 600 60 6 0 0 0 0` 这里填入实际显示屏的分辨率，不同显示器分辨率不同。

- 保存文件后，将microSD卡插回Raspberry Pi中。
- 初次启动与初始化设置：
- 将烧录好OS镜像的microSD卡重新插回Raspberry Pi后，给它通电。
- 第一次启动时，根据屏幕上的指示进行语言、地区、时区等基本信息的配置。

(3) 网络配置

- 连接Wi-Fi：

2. 在命令行中输入 `sudo raspi-config` 打开配置菜单。
3. 选择“Network Options”，然后按照提示输入您的Wi-Fi SSID和密码。
4. 验证网络连接：
5. 使用 `ping www.bing.com` 测试是否能成功访问外部网站。

(4) 配置开发环境

1. 更新软件包列表：
2. 执行 `sudo apt-get update` 刷新本地数据库以获取最新的软件包信息。
3. 升级已安装的软件包：
4. 使用 `sudo apt-get upgrade` 命令来更新所有现有的软件包至最新版本。
5. 安装额外的开发工具：
6. 安装Git用于版本控制：

```
1 sudo apt-get install git
```

7. 安装Vim编辑器： `bash`
`sudo apt-get install vim`

8. 安装远程连接工具：

9. 启动SSH服务：

```
1 sudo systemctl start ssh
```

10. 验证SSH服务是否正常运行：

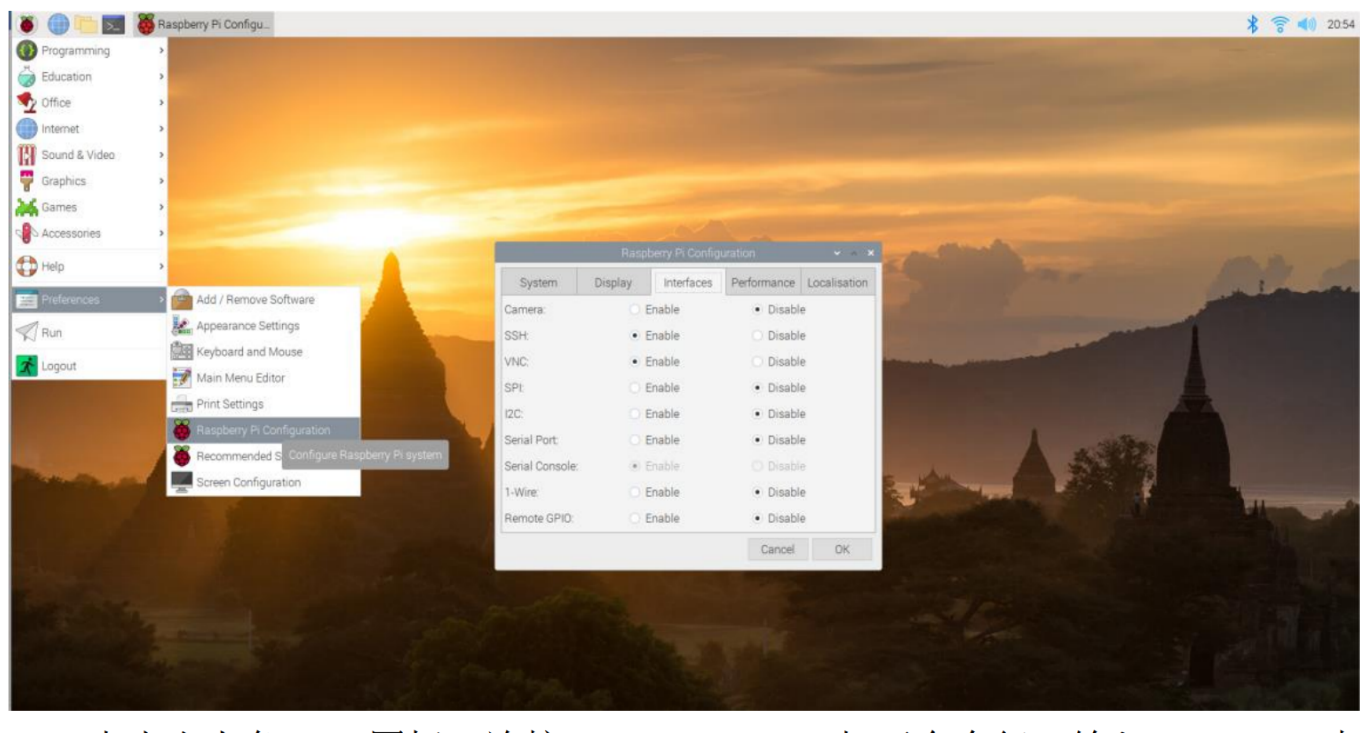
```
1 sudo systemctl status ssh
```

11. 设置SSH服务开机自启动： `bash`
`sudo systemctl enable ssh`

12. 在本地通过vscode连接远程Raspberry Pi：

13. 安装Remote - SSH插件。
14. 在树莓派终端运行 `ifconfig` 命令查看IP地址。
15. 使用 `Ctrl+Shift+P` 打开命令面板，输入 `Remote-SSH: Connect to Host`，然后输入Raspberry Pi的IP地址和用户名。
16. 输入密码后，即可通过VSCode连接到远程的Raspberry Pi。
17. 把 `ssh-rsa` 公钥添加到 `~/.ssh/authorized_keys` 文件中，实现无密码登录。
18. 远程可视化开发:
19. 在本地安装VNC Viewer。

20. 在树莓派上启用VNC Server：



21. 确保电脑和 Raspberry 在同一局域网。打开软件，在框内输入获取到的 Raspberry Pi ip 地址，回车。

22. 输入用户名和密码，即可远程连接到 Raspberry Pi 的桌面环境。

四、总结

本实验完成了Raspberry Pi的初步设置，包括操作系统的安装、网络的配置以及开发环境的搭建。

2.2 双色灯实验

Lab2实验报告：学习知识准备与双色LED实验

一、实验介绍

本实验旨在了解Raspberry Pi的IO接口及其引脚编号方式，并通过实际操作掌握使用wiringPi库和RPi.GPIO库控制硬件的方法，最终目标是实现一个简单的双色LED红绿交替闪烁效果。

二、实验原理

1. **Raspberry Pi IO口：**
2. Raspberry Pi拥有40个GPIO管脚，这些管脚可以通过不同的编号系统来引用，包括物理位置编号、wiringPi指定的编号以及BCM2837 SOC指定的编号。
3. 在本次实验（以及后续实验）中，我们使用BCM编码来连接和编程。
4. **wiringPi库：**
5. wiringPi是一个用于C/C++语言的GPIO控制库，它简化了对Raspberry Pi GPIO的操作。安装此库后，可以方便地在命令行或程序中控制GPIO引脚。
6. **RPi.GPIO库：**
7. RPi.GPIO是一个Python库，允许用户直接从Python代码中控制Raspberry Pi的GPIO。它是Raspbian操作系统的一部分，默认已安装，因此可以直接调用其API进行编程。
8. **Mu编辑器与Geany IDE：**
9. Mu是一款适合初学者使用的Python编辑器，提供了基本的IDE功能如语法检查、运行和调试等。
10. Geany则是一款轻量级的跨平台IDE，支持多种编程语言，对于C/C++项目来说非常适合。



后续经验表明，这两款IDE在树莓派下渲染效果不佳，推荐使用VSCode远程开发插件。

1. **双色LED模块：**
2. 双色LED通常指的是包含两个独立发光单元（红色和绿色）在一个封装内的LED。通过改变输入电压或电流的方向，可以使不同的颜色发光或者两者同时亮起形成黄色。

三、实验步骤

1. **硬件连线：**
2. 将双色LED的S引脚（绿色）、中间引脚（红色）分别连接到Raspberry Pi的GPIO接口上，GND引脚连接到Raspberry Pi的地线。记住使用的GPIO引脚编号（本次实验使用的是BCM编号下的GPIO19, GPIO20, GND）。
3. **编写并上传代码：**
4. 使用Mu编辑器创建一个新的Python脚本文件，编写一段代码来控制双色LED的红绿交替闪烁。代码应该设置好相应的GPIO模式（输入/输出），然后按照设定的时间间隔切换LED的状态。
5. 如果使用C/C++，则可以在Geany中新建一个源文件，同样需要配置GPIO模式，并且要记得在编译时链接wiringPi库。
6. **运行测试：**
7. 执行编写的Python脚本或编译后的C/C++程序，观察双色LED是否能够按照预期的顺序红绿交替闪烁。
8. 除了代码逻辑，**一定要检查硬件连接是否正确无误！！！！**（比如是否插紧T型板、是否接错了引脚等）
9. **清理工作：**
10. 实验结束后，记得关闭所有运行中的进程，否则LED灯可能会一直亮着。

四、编写代码

程序框图：

