**西南交通大学**

**嵌入式系统实验日志**

****

**学号：**杨志成

**姓名：**2023211042

**组号：**3组

**队号：**6队

**实验序号：**2

**实验名称：**数码管控制

**使用语言：**Python

1. **实验目的**

1、 了解 4 位 LED 显示模块的结构和控制方法

2、 掌握利用 Python 语言，通过树莓派的 GPIO 口控制 4 位 LED 模块的显示

3、 学习 I2C 接口规范

4、 学习 PCF8591 AD/DA 转换模块的使用方法，并利用树莓派的 I 2 C 接口读取 PCF8591 的 数据，并显示在 LED 显示模块中

5、 利用 TK 编写一个窗口界面，在界面中显示 PCF8591 模块的数据

1. **实验内容**

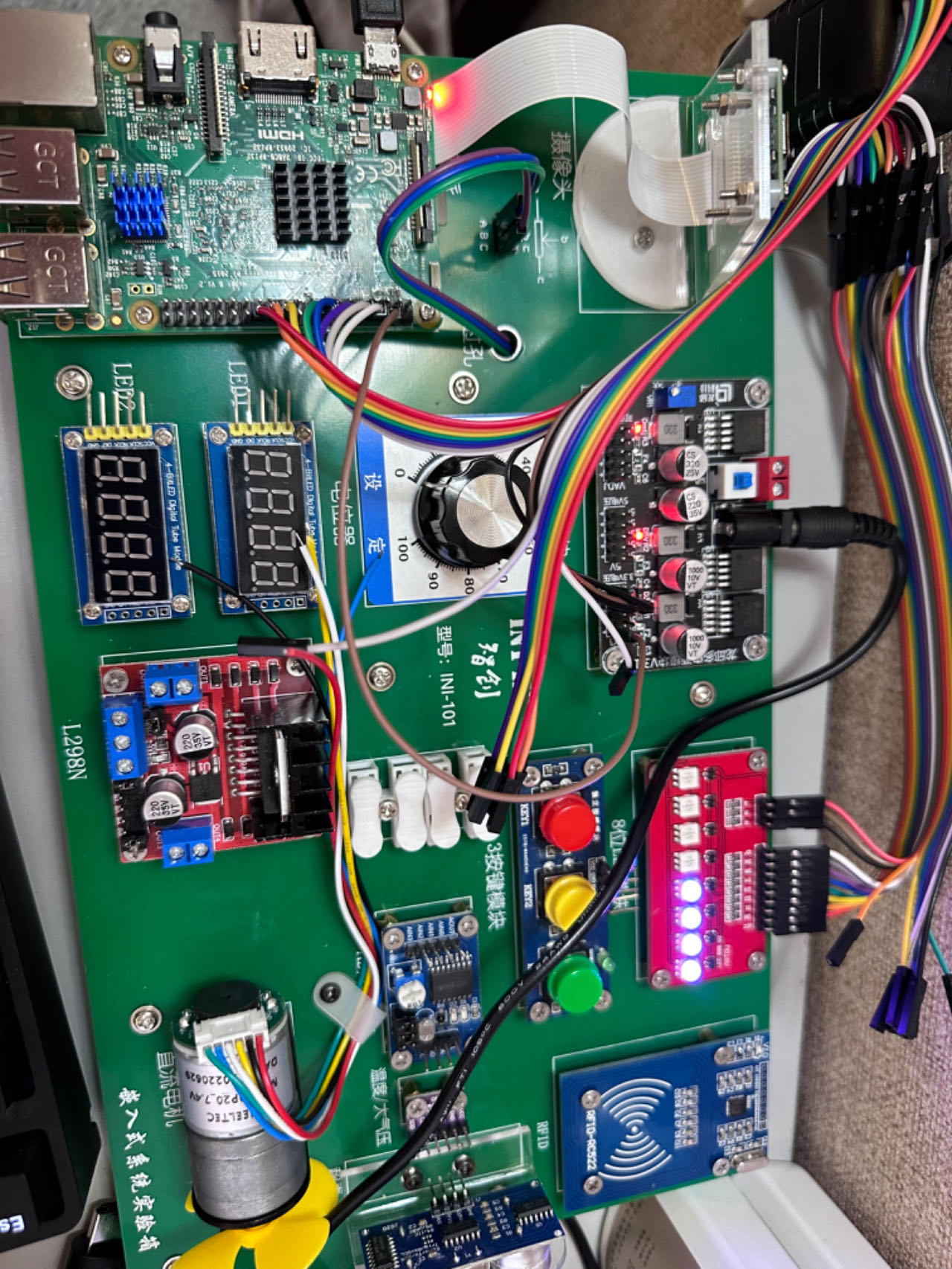
本实验以树莓派作为控制器，以 Python 作为开发工具，利用树莓派的 GPIO，控制 2 个 4 位 LED 显示模块，让其显示指定数据，利用 AD/DA 转换模块，读取其板载的热敏电 阻值、光敏电阻值、可变电阻的分压值，并在 LED 模块中显示；利用 TK 编写一个窗口界 面，将 PCF8591 模块的三个数据实时显示出来。

1. **实验步骤**

1. 连接：用杜邦线，将树莓派的 GPIO 与 8 个 LED 控制信号连接起来，由电源模块给 LED 模块供电；

2. 编写程序控制 8 个 GPIO 闪烁，从左到右依次点亮、再依次熄灭，从右到左依次点亮、再依次熄灭，再从两边向中间依次点亮，再依次熄灭，从中间向两边点亮，再依次熄灭。

1. **硬件连接图,软件流程图**

****



1. **关键代码分析**

代码使用RPi.GPIO库来控制树莓派上的GPIO引脚，实现LED灯的控制。

GPIO.setmode(GPIO.BCM) 设置GPIO的编号模式为BCM模式。

LED\_List 是包含8个GPIO引脚的列表，用于连接LED灯。

GPIO.setup(LED\_List, GPIO.OUT, initial=GPIO.HIGH) 配置LED引脚为输出模式，并初始化状态为高电平（LED熄灭）。

tt 是时间间隔，设置LED点亮和熄灭之间的时间间隔为0.1秒。

接下来，代码进入一个无限循环，循环中的关键部分如下：

使用for循环和GPIO.output逐个点亮和熄灭LED，从左到右和从右到左，以及两边向中间点亮和熄灭LED。这些操作创建了不同的LED闪烁效果。

异常处理部分捕获异常（例如按下Ctrl+C停止程序），并打印 "except"。

最后，通过GPIO.cleanup()清理GPIO资源，然后打印 "END of blinking" 表示闪烁结束。

1. **实验结果**

以跑马灯的形式展示了led灯的点亮效果:

1.左到右依次点亮 LED，然后熄灭。

2.从右到左依次点亮 LED，然后熄灭。

3.两侧向中间点亮 LED，然后熄灭。

4.中间向两边点亮 LED，然后熄灭。

1. **实验遇到的问题及处理方法**
2. 安装系统后环境出现问题，无法连接网络和运行程序

解决方法：重新烧录树莓派系统后解决问题。

2.连接好电路之后，运行程序没有反应

解决方法：查看手册和排查电路后，发现没有将树莓派和8位LED模块共接地，共接地后就可以正常运行并显示跑马灯流水效果。

1. **实验中，自己的创意部分**

实验要求是要准备3种不同的闪烁方案，我实现了4种闪烁方案的跑马灯效果。

1. **实验中的收获**

学习基本的硬件控制：这个实验使我学会如何使用树莓派的GPIO引脚来控制外部硬件，即LED灯。这是嵌入式系统和物联网项目中的基础技能。

编程技能提升：编写控制LED流水灯的代码，使我在Python编程方面有所提高。学会如何配置GPIO引脚、使用循环、条件语句和异常处理等编程概念。

电子原理理解：通过连接LED到树莓派，我理解LED灯的工作原理以及如何选择适当的电阻来限制电流。

问题解决能力：当遇到问题时，例如LED不亮或者闪烁不正常，我调试代码和硬件连接。培养我解决问题的能力。