树莓派小车循迹实验报告

智能系统中的嵌入式应用　　19180099　邱梓航　　2023 年 6 月 18 日

1　实验环境

操作系统：Windows11 的 Windows Subsystem for Linux、树莓派系统为 Linux version 6.2.0-v8+
开发工具：在 VSCode 中完成代码编写，再使用 scp 传输到树莓派
硬件环境：微雪电子 AlphaBot2-Pi、Raspberry Pi3

2　实验内容

实现 Python 程序控制树莓派小车在开放环境中进行简易循迹。

3　准备工作

3.1　硬件组装

小车的主要部分由一个基板 AlphaBot2-Base 和一个适配板 AlphaBot2-Pi 组成。根据适配板的安装方向，有两种组装方法，分别是树莓派连接口朝上和朝下两种情况。当树莓派连接口朝下时，这也是官网手册的安装方法，小车便于在适配板上安装摄像头进行相关开发，同时五向遥感、适配板上用于连接树莓派的串口等器件和接口也将暴露在小车上方，便于观察和操作；当树莓派连接口朝上时，树莓派被安装在小车上方，给小车中部腾出了空间，方便在基板的超声波模块接口处安装超声波模块，但摄像头的安装位置被遮挡。红外避障遇到透明物体或近似黑体等情况下效果没有超声波传感器好，因此可以根据实际实验需求选择合适的安装方法。出于避障和循迹原理类似，本次实验主要开展了小车循迹的过程研究。安装过程主要参考了该视频和微雪电子官网的组装图。

3.2　第三方库安装

实验涉及的 RPi.GPIO、time 库已经预装，在实现控制小车底部的彩灯时需要额外安装 neopixel 库。按照教材指引，通过 git clone 下载好第三方库 rpi_ws281x 后，传输到树莓派进行安装时会出现 ZIP does not support timestamps before 1980 的报错，使用命令 date 可知，这是因为提供的树莓派系统每次启动后的时间都被重置为 1970 年 1 月 1 日导致。可以使用类似于命令 date -s "2306171736" 修改系统日期到 2023 年 6 月 17 日 17 时 36 分，注意该方法是临时生效的，重启系统后失效，但足以用于解决安装问题。修改日期后使用命令 find . -type f -exec touch {} \ ; 在当前目录及其子目录中找到所有文件并更新它们的访问时间和修改时间为当前时间。完成修改后进行模块的安装即可正常运行，安装后在 Python 程序中导入的模块名为 neopixel。

3.3　程序开机自启

实现程序开机自启需要在/etc/init.d/下放入修改过执行权限且文件开头有释伴指明解释器的脚本，并在/etc/rc.d/rc.conf 配置文件中的 cfg_services 项中 Desktop 前添加该脚本名称，开机后稍等片刻便会执行该脚本。

4 小车循迹实现

根据官方手册和课程讲义，结合实际操作可知：

- GPIO12、13 控制小车左轮的前进方向，当 GPIO12 为高电平，GPIO13 为低电平时，左轮倒退运转，反之左轮前进运转，二者均为低电平（即无电压差）时停止运转；同时 GPIO6 使用 PWM 方式控制左轮的转速。
- 同理，GPIO20、21、26 分别控制小车右轮方向和转速。
- GPIO5、25、24、23 分别控制红外传感器连接的模数转换器的 CS、CLOCK、ADDRESS、DATAOUT，分别负责使能与禁止、时钟脉冲、通道编码和数据转换值。
- GPIO18 控制小车底部的四盏彩灯，并可以指定一个引脚控制彩灯的 DMA 通道，教材上使用了 GPIO10。
- GPIO7-11 控制五向摇杆，其中 GPIO7 控制中心按钮，可以用于小车脱离远程控制时进行简单操作。

实验中编写的程序主要参考了教材和微雪电子 WIKI官网的示例程序。实验实现的循迹程序主要由一个运行程序（Line_Follow.py）两个模块程序（AlphaBot2.py、TRSensors.py）构成。

模块 AlphaBot2.py 根据小车操作电机的端口，通过 RPi.GPIO 中 PWM、ChangeDutyCycle 等函数控制电机两轮转速，通过 output 函数调节电平高低控制车轮转向，从而可以组合实现小车前进、后退、停止、左转、右转等基本行进功能，提供小车行进功能的调用接口。

模块 TRSensors.py 根据小车红外传感器的接口，将 CS、Clock 和 Address 对应的 GPIO 接口设置为输出，将 DataOut 和五向摇杆按钮对应的接口设置为输入。每次读取数据时，根据 TLC1543 的时序，将 CS 输出为低电平进入读取状态，在 Clock 处于低电平时向 Address 输入一位通道编码，同时从 DataOut 读取一位转换值，再经过一次 Clock 的上升沿和下降沿完成 Address 和 DataOut 各一位的读写。数据通道共有 10 位，其中第一个通道数据无效，故选用后面位数的数据作为数据的转换值。

在模块 TRSensors.py 完成数据读写的基础之上，为了减少传感器自身以及环境的影响，对五个传感器输出数据进行了归一化处理，并为了数据处理方便将数据放大 1000 倍。归一化的方法是在程序启动时先进行不断的多方向移动与旋转，从中多次采集传感器的值，收集最值并保存，用于后续进行循迹时的数据归一化。

根据微雪官方教程，可以通过加权平均的方式将五个传感器的值转变成一个数值用于确定路线位置:

$$y = \frac{0 * value_0 + 1000 * value_1 + 2000 * value_2 + 3000 * value_3 + 4000 * value_4}{value_0 + value_1 + value_2 + value_3 + value_4}$$

经过处理后的数值范围为 0-4000，代表黑线的位置。由于基板底部的五个红外传感器为横向依次排列，故 2000 表示黑线在模块的正中间，0 表示在黑线在模块的最左侧，4000 表示黑线在最右侧。为了使模块探测的精度更高，根据微雪官方教程，黑线应该等于或略小于探测器的距离（16mm）。高度为黑线在两个传感器正中间时两个传感器都能够刚好探测到为宜，因此在自制路线时应该根据器件情况谨慎设计。

在运行程序 Line_Follow.py 中，设计了按钮启动循迹程序，这样可以在程序开机自启脱离远程控制时仍能控制电机的运行与停止。同时程序使用 neopixel 对象控制基板底部的全彩 LED 灯，使其运行时闪烁彩灯。此外程序调用 TRSensors.py 中的函数，查看当前传感器的黑线位置数值和各传感器数值，判断当前小车与黑线的相对位置。如果各传感器数值均较高，说明小车位于白色环境中未发现黑线，小车静止；否则小车前进，同时为了使小车一直沿着黑线走，黑线位置数值应该为 2000。为了使小车沿黑线走且减少小车左右摇摆，采用位置式 PID 控制小车前进。PID 是指通过比例 (P，当前的位置减去目标位置，即 2000)，积分 (I，每次误差的总和)，微分 (D，当前误差和上次误差的差值) 对误差进行反馈调节。可以对这三个数值进行加权求和将所得结果修正后作为小车左右轮转速差，从而控制小车转向幅度，并通过多次实验调节权重数值获得最佳性能。

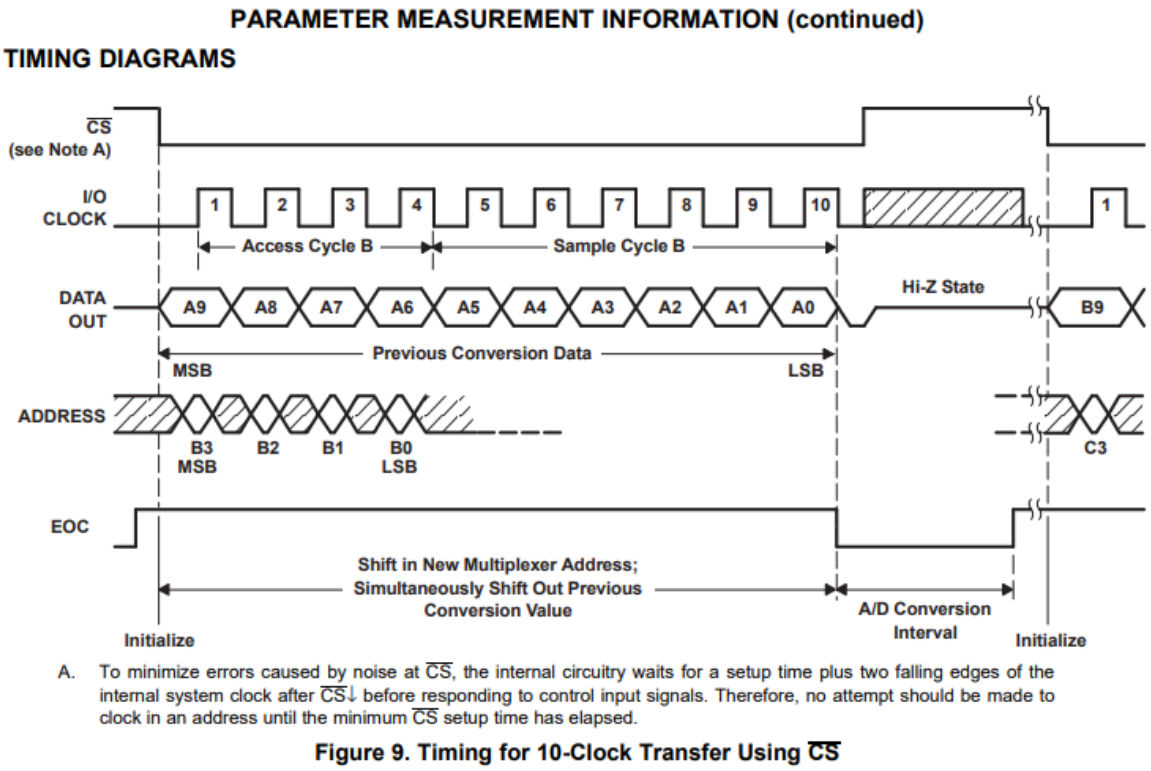

Figure 1: TLC1543 时序

5 实验结果展示

效果展示见bilibili 视频。代码见GitHub 仓库。

6 总结与展望

这次实验参考课程教材和微雪电子官方教程，结合课上所学，完成了树莓派控制小车组件进行简易的红外循迹的操作。树莓派和相关的智能电子组件为我们提供了一个低成本、灵活且强大的平台，推动了科技教育、创客文化、物联网应用和嵌入式系统开发的发展。它鼓励我们动手实践，尝试创造自己的电子设备和项目，培养了我们的创新思维、合作精神和问题解决能力，激发了我们对科学、技术、工程和数学（STEM）领域的兴趣。这学期的课程和实验为我们未来的学习和职业发展打下了一定的基础，希望今后有机会能进一步感受智能系统与嵌入式开发的魅力。