

南京大学电子科学与工程学院

实验报告

学号	181098040
姓名	丛进
课程名称	智能系统中的嵌入式应用
实验名称	避障循迹遥控小车
理论教师	方元
实验教师	方元

目录

一、 实验目的	3
二、 实验器材	3
2.1 树莓派	3
2.2 Alfabot2 小车	3
三、 实验原理	4
3.1 传感器	4
3.1.1 超声波传感器——避障原理	4
3.1.2 蜂鸣器	4
3.1.3 前端红外传感器——避障原理	5
3.1.4 底部红外传感器——循迹原理	5
3.2 彩灯	6
3.3 遥控器	6
3.4 直流电机车轮——避障原理	6
3.4.1 前进	6
3.4.2 转向	7
3.5 交互控制方式	7
四、 实验内容	7
4.1 模块化编程	7
4.2 超声波探测	8
4.3 前端红外探测	8
4.4 底部红外探测	8
4.5 移动	8
4.6 LED 变色灯光	8
4.7 蜂鸣器	8
4.8 命令行操作	8
4.9 多线程	9
4.10 遥控器	9

五、 实验结果	9
六、 实验展望	11
七、 实验感悟	11

南 京 大 学

实 验 报 告

学生姓名：丛进 学号：181098040 指导教师：方元

实验地点：仙林校区基础实验楼嵌入式教室

实验时间：第 15-17 周周四 18：30-20：30,1 月 5 日，1 月 6 日

实验学时：2

实验名称：避障循迹遥控小车

一、实验目的

- 1) 拼装小车
- 2) 自动避障
- 3) 循迹
- 4) 遥控器遥控

二、实验器材

2.1 树莓派

树莓派这类计算机结构简单、体积小、耗电低,却拥有与普通计算机几乎相同的功能和性能,可以很方便地植入各种应用系统中。这类单板计算机也是典型的嵌入式系统的基础。本次实验用到的树莓派 3B/3B+, 带有一个以太网接口、4 个 USB host、1 个无线网接口和蓝牙接口。片内大量 I/O 接口通过一组 2*20 引脚引出, 作为扩展设备控制接口。引脚功能见图1。

2.2 Alphabot2 小车

拼装采用底部超声波, 顶部安放树莓派的方式。

最终拼装成果见实验结果中的图8图9图10

Item	GPIO	Item	GPIO
Buzzer	GPIO4	Joystick	GPIO7-11
IR sensors (Bottom) (TLC1543)	GPIO5(CS)	Wheel A	GPIO12-13
	GPIO25(CLK)		GPIO6 (CTRL)
	GPIO24(ADDR)	Wheel B	GPIO20-21
IR remoter recv.	GPIO23(DOUT)		GPIO26(CTRL)
	GPIO17	Servo (I2C)	GPIO2(SDA)
Color LEDs (Bottom)	GPIO18	(PCA9685)	GPIO3(SCL)
UltraSonic	GPIO22(Trig)	IR (Front-R)	GPIO19
	GPIO27(Echo)	IR (Front-L)	GPIO16

图 1: Alphasbot2 资源分配表

三、实验原理

3.1 传感器

3.1.1 超声波传感器——避障原理

超声波硬件见图3，测距模块原理见图2。Trig 端口产生不短于 10ms 的正脉冲，模块会自动发出 8 个周期的超声脉冲信号 (40kHz), 并在 Echo 端输出高电平。当检测到回波时将 Echo 回置到低电平。通过测量 Echo 高电平维持周期, 再根据声波速度, 就可以算出发射器到障碍物之间的距离。

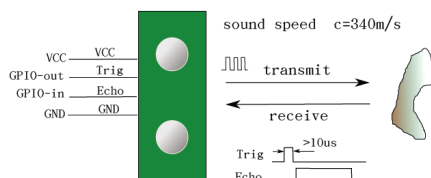


图 2: 超声波测距原理



图 3: 超声波硬件

3.1.2 蜂鸣器

蜂鸣器是一种一体化结构的电子发声设备，主要有压电式和电磁式两种类型。多谐振荡器通电源后输出 1.5 2.5kHz 的音频信号，推动压电蜂鸣片发声。

蜂鸣器的 GPIO 口输入为高电平后，开始发声音。修改为低电平后，停止发声。

3.1.3 前端红外传感器——避障原理

硬件见图4，小车前部左右各一个。红外传感器由一个红外发射管和一个红外接收管组成。发射管和接收管之间有隔离板。遇到障碍物时，发射管发出的红外光，被物体反射后由接收管接收，有障碍时输出低电平，没有时输出高电平。

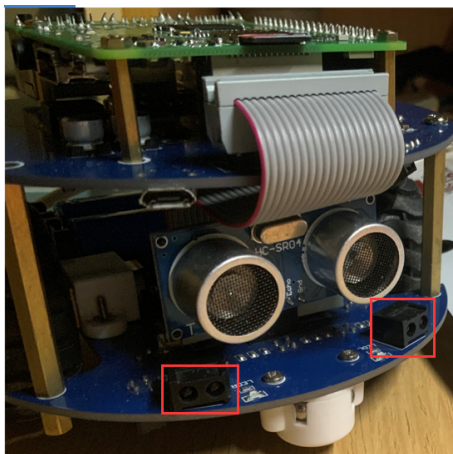


图 4: 2 个光照传感器

3.1.4 底部红外传感器——循迹原理

硬件见图5，小车底部有 5 个。与前端的红外传感器只能输出 0-1 不同，底部传感器经过模数转换器得到量化的红外强度值，即输出 0-999 的具体强度值。接收强度与传播距离和反射面对红外光的吸收性质有关。



图 5: 底部 5 个光照传感器

3.2 彩灯

每一个彩灯灯珠是一个三色 LED, 通过红、绿、蓝三个发光二极管产生的不同亮度形成不同的颜色。每个发光二极管由一个 8 位的数字信号控制, 实现 256 级亮度。数字信号以串行方式输入, 多个灯珠串接, 形成一个灯带。rpi_ws281x 模块提供了 Python 接口。

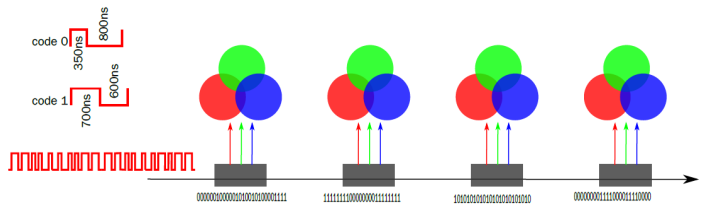


图 6: 彩灯控制原理

3.3 遥控器

红外遥控器发射的信号使用 38kHz 左右的载波对基带进行调制。接收端对信号监测、放大、滤波、解调等一系列处理, 然后输出基带信号。收发方采用约定的协议进行通信。图7是 NEC 红外通信协议发送信号的波形。发送端首先发送一个 9ms 低电平接 4.5ms 高电平的引导码, 接收方检测到引导码后开始识别后面的数据: 0.56ms 低 +0.56ms 高表示“0”, 0.56ms 低 +1.69ms. 高表示“1”。一组 0、1 序列构成一个按键特征字。接收方根据收到的特征字产生一定的动作。

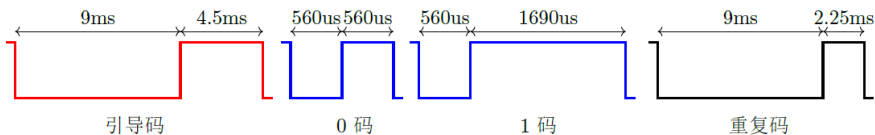


图 7: NEC 红外遥控器协议

3.4 直流电机车轮——避障原理

小车拥有两个独立的直流电机, 分别控制左侧和右侧的两个车轮。可以分别设置顺时针或者逆时针转动, 从而可以实现前进, 后退, 转弯。

3.4.1 前进

没有障碍物, 一直直线前进。速度可通过命令行在运行中调节。

3.4.2 转向

1. 采用原地转圈方案。每次转向 30° 左右。如果转向 1 次之后，还是有障碍物，则继续转向，直到前方和侧方没有障碍物。使用小角度原地多次调整的原因是：无法保证小车周围是否有足够空间进行后退转向，或者非对称轴偏移转向。在且低速状态下，一个轮子不转，另一个轮子有可能转不动。

2. 超声波和红外共同检测障碍物。仅使用红外会因为材质原因撞上部分物体，比如黑色的门。仅使用超声波，会因为只能探测直线，而导致侧着撞到障碍物。两者结合可以保证几乎 100% 不撞到障碍物。

当红外探测到左边有障碍物 → 右转 (左侧轮子前转，右侧轮子后转)。

当红外探测到右边有障碍物 → 右转 (左侧轮子后转，右侧轮子前转)。

当红外同时探测到障碍物，或者只有超声波探测到障碍物，随机左右转。

随机的转圈是为了进入几乎环形的死角有机会出来。

3.5 交互控制方式

树莓派的有静态 Ip 地址 192.168.208.132，电脑与树莓派之间通过 ssh 链接控制。通过命令 `ssh root@192.168.208.132` 进行链接，链接完成后进入树莓派命令行页面

传输代码文件使用 `scp` 命令。在电脑上用 `pycharm` 进行代码编写，完成后将文件夹上传至树莓派：`scp -r car root@192.168.208.132:。` `-r` 表示递归文件夹其中 `car` 时本项目控制小车所有代码文件所在的文件夹。

四、实验内容

4.1 模块化编程

将超声波探测、红外探测、彩灯、蜂鸣器等功能分离成独立的、可相互改变的“模块”，使得每个模块都包含着执行预期功能的唯一方面。

由于 `python` 没有指针概念，所以没法实现跨文件变量共享。这给基于模块化的多线程操作全局变量带来麻烦。于是在 `global_var.py` 维护了一个字典 `_global_dict`。其他文件在使用时 `import global_var`。对 `_global_dict` 的值进行 `get` 和 `set` 操作。

具体代码实现见 [global_var.py](#) 文件

4.2 超声波探测

用超声波硬件探测直线精确距离。本次实验中将危险距离设置为 15cm，即当正前方 15cm 厘米内有物体，小车就会转向。

具体代码实现见 [sonic.py](#) 文件

4.3 前端红外探测

主要用于探测小车左右有无障碍物。通过改变底部的灵敏度按钮，用手掌进行测试，当 10cm 左右有障碍物时，改变电平。

具体代码实现见 [light.py](#) 文件

4.4 底部红外探测

5 个探测器分别探测底部的具体光强。在低速模式下前进。高频检测 5 个探测器的光强，如果有一侧光强和其他差距太多，则进行超微小角度调整，修正前进方向。

具体代码实现见 [infrared.py](#) 文件

4.5 移动

速度可以在小车移动时通过命令行修改。小车可以前进、后退、左右转向。

移动策略见实验原理直流电机部分。具体代码实现见 [move.py](#)

4.6 LED 变色灯光

彩灯可以显示全白，也可以五颜六色。具体代码比较长，限于文章篇幅，可在仓库中查看源代码 [LED.py](#)。展示效果可以看视频。

4.7 蜂鸣器

遇到障碍物时，设置警报，小车发声。检测频率和距离探测的频率一致。代码实现见：[buzzer.py](#)

4.8 命令行操作

可以初始速度设置，决定是否打开蜂鸣器功能、彩灯功能
参数解释

- 1) -s -speed, 设置小车速度。并检验是否是 0-100 的合法值。运行时可以改变速度。
- 2) -b -buzzer 控制蜂鸣器开关。1 为开, 0 为关
- 3) -l -led 控制彩灯开关。1 为开, 0 为关

代码实现: [parse.py](#)

4.9 多线程

由于距离探测等功能是 `while True` 一直运行的, 如果集中在一个文件中, `while` block 下方的代码可能会永远无法执行。所以采用多线程技术, 以实现距离探测、蜂鸣器警报、LED 灯效、前进并行运算。代码见 [run.py](#)

4.10 遥控器

由于拿到的盒子里并没有遥控器, 在第 17 周借用同学遥控器实现了简单的代码: [remotor.py](#)

五、实验结果

- 1) 完整小车

见图8图9图10

- 2) 避障功能

本次实验实现了基于红外检测的自动避障小车应用, 小车可以利用红外传感组件实时探测前、左、右三个方向。障碍物, 并根据检测结果实时操纵直流电机来调整行进的速度与方向, 实现避障功能。

避障效果展示视频见附件。

全部代码地址: <https://github.com/Ironstarboy/RP3b/tree/master/car>

视频对应控制台输出结果见图11. 解释: `led stated=1` `buzzer state=1` 表示蜂鸣器和彩灯都开着。`turnLeft,turnRight` 记录了每次转向的方向。出现的数字 0, 10, 20 表示在运行中输入的速度数据。

- 3) 遥控功能

本次实验完成了遥控器代码的撰写。

- 4) 循迹功能

本次实验在理论上完成了循迹的代码。并用手掌成功测试且可以输出数据。



图 8: 小车顶部

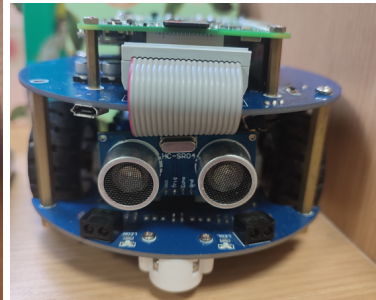


图 9: 小车前部



图 10: 小车后部

```
RaspberryPI:~/car # python3 parse.py --speed 10 -b 1 -l 1
led state = 1
buzzer state = 1
current speed = 10
buzzer on!
led on!
turnLeft!
turnLeft!
turnLeft!
turnLeft!
turnRight!
turnLeft!
20turnRight!
turnRight!
turnLeft!
0
turnRight!
turnLeft!
10
20
```

图 11: 命令行运行输出与控制

六、实验展望

本次实验实现了小车避障的所有功能，但仍然存在些许不足的地方，可以作为后续实验文进的研究方向。

- 1) 由于超声波和红外探测其距离地盘比较高。所以小车无法探测低于探测平面的物体，比如地上的绳子
- 2) 红外探测器灵敏度比较难调，很难调到左右一致
- 3) 由于宿舍无黑色胶布，没有条件完成循迹的实操

七、实验感悟

作为一位来自商学院的本科生，本专业内几乎接触不到代码和硬件，但我从小一直对电子科学有着极大的兴趣。这门智能系统嵌入式应用课程，成功让我体验到电子科学的无比魅力！当自己一个个实现功能的时候，很有成就感。我想哪些奋斗在科研或者工业一线的电子人才，也一定很享受亲手改变世界，造出让生活更美好、更便利的电子产品的过程！

非常感谢老师一学期以来的指导！