华南理工大学

South China University of Technology

《机器人编程基础》
实验报告

实验题目：基于 RaspberryPi 的机器人编程

姓名、学号：成子谦，201730681303

学院、班级：软件学院 2017 级 1 班

华南理工大学软件学院

二〇一九年十一月

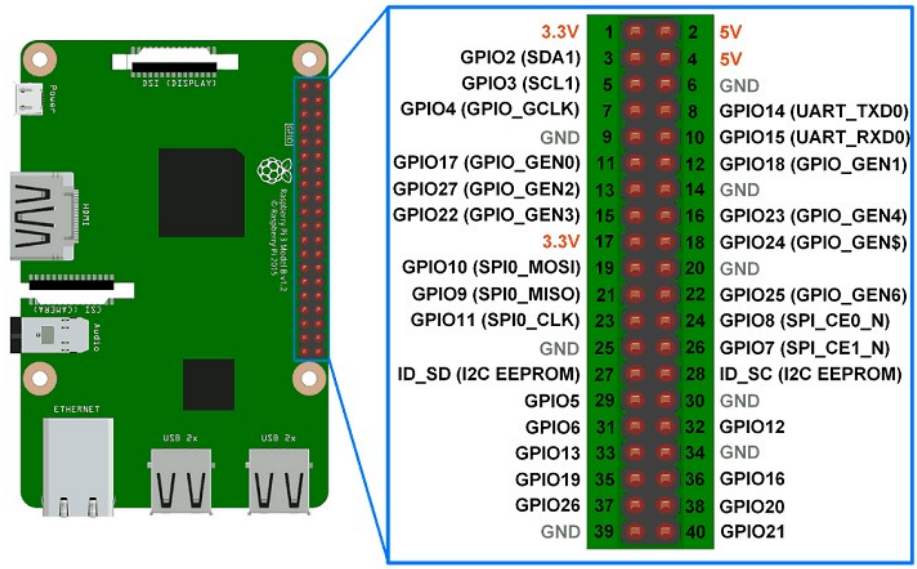
目　录

1. 实验目的

1. 掌握 RaspberryPi 的基本用法
2. 掌握基于脚本的机器人编程

2. 实验材料

2.1. RaspberryPi 3B 嵌入式开发板



注意管脚编号在板上的物理位置，其中 39 和 40 是靠近 USB 口的。

2.2. 机器人车体及传感器

与《基于 Arduino 的机器人编程》实验中的设备一样。

3. 实验内容

3.1. 安装 Raspbian 系统

安装 Raspbian 系统到 SD 卡

过程可参考： https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up

下载地址：https://www.raspberrypi.org/downloads/

3.2. 配置 Wi-Fi 热点

安装 RaspAP，配置树莓派为一个 wifi 热点，修改热点名称和密码。

操作过程见 https://github.com/billz/raspap-webgui 的 readme.md

成功后，PC 可以远程 ssh 登录到树莓派，而不需要网线和显示器。

3.3. 安装 WringPi-Python 库

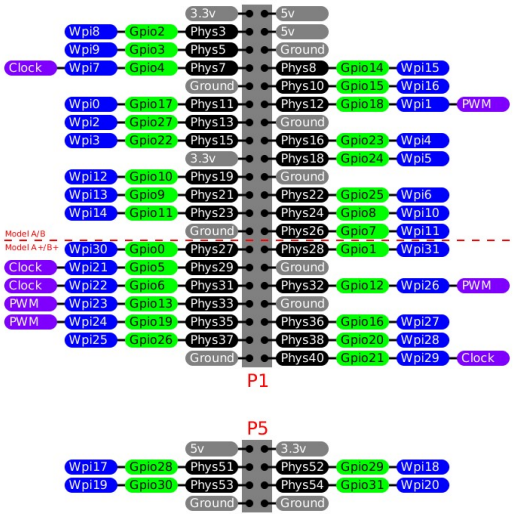
打开终端，运行：

```
pip3 install wiringpi
```

参考例子见源代码：

https://github.com/WiringPi/WiringPi-Python

wiringPi 的管脚编号：



3.4. 组装机器人

将 RaspberryPi 板替换《基于 Arduino 的机器人编程》中的 Arduino Uno 板。

连线 RaspberryPi 与外围设备。

注意事项：

1. 连线前，请断开电源开关。
2. 确认 P1 排针的管脚编号 1 的位置，管脚 39 和 40 是靠近 USB 口。

3.5. 软件模块

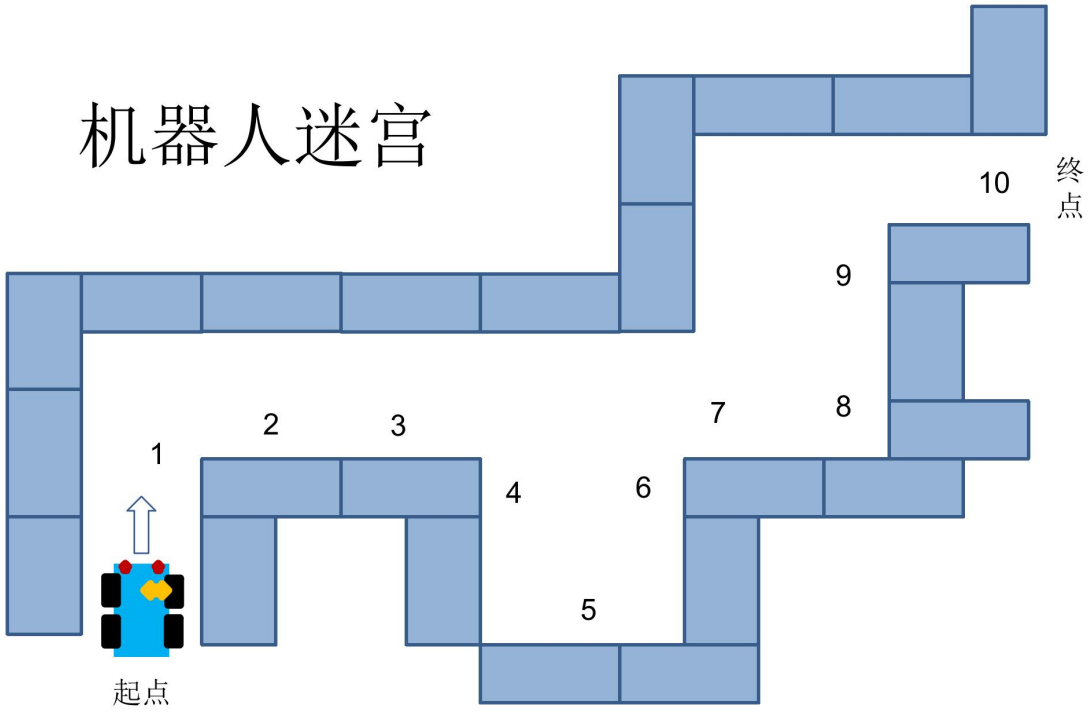
编写各个模块的测试程序，包括：

1. 红外障碍物检测。
2. 超声波测距。
3. 电机驱动。

3.6. 机器人走迷宫

编写走迷宫机器人程序。

提示：机器人控制策略是一直靠右墙走。



附录

附录 A：机器人走迷宫源代码

```
import wiringpi
import time


# infrared pin
INFR_LEFT = 0
INFR_RIGHT = 1
# ultra sound
ECHO = 2
TRIG = 3
# wheel pin
WHEEL_1 = 6
WHEEL_2 = 10
WHEEL_3 = 11
WHEEL_4 = 31
# wheel speed
FAST = 90
SLOW = 50
# epsilon
EPS = 0.000001
LIM = 50
# infr initial state
left_infr = 0
right_infr = 0


# pin init
def init():
```

```
    # pin init
    wiringpi.wiringPiSetup()
    # wheel pin init
    wiringpi.pinMode(WHEEL_1,1)
    wiringpi.pinMode(WHEEL_2,1)
    wiringpi.pinMode(WHEEL_3,1)
    wiringpi.pinMode(WHEEL_4,1)
    wiringpi.softPwmCreate(WHEEL_1,0,200)
    wiringpi.softPwmCreate(WHEEL_2,0,200)
    wiringpi.softPwmCreate(WHEEL_3,0,200)
    wiringpi.softPwmCreate(WHEEL_4,0,200)
    # infrared init
    wiringpi.pinMode(INFR_LEFT,0)
    wiringpi.pinMode(INFR_RIGHT,0)
    # ultra sound init
    wiringpi.pinMode(TRIG,1)
    wiringpi.pinMode(ECHO,0)
    wiringpi.digitalWrite(TRIG,0)

# car stop
def stop():
    wiringpi.softPwmWrite(WHEEL_1,0)
    wiringpi.softPwmWrite(WHEEL_2,0)
    wiringpi.softPwmWrite(WHEEL_3,0)
    wiringpi.softPwmWrite(WHEEL_4,0)

# car go forward
def goForward():
    wiringpi.softPwmWrite(WHEEL_1,FAST)
    wiringpi.softPwmWrite(WHEEL_2,0)
```

```
    wiringpi.softPwmWrite(WHEEL_3,0)
    wiringpi.softPwmWrite(WHEEL_4,FAST)

# car go backward
def goBackward():
    wiringpi.softPwmWrite(WHEEL_1,0)
    wiringpi.softPwmWrite(WHEEL_2,FAST)
    wiringpi.softPwmWrite(WHEEL_3,FAST)
    wiringpi.softPwmWrite(WHEEL_4,0)

# car turn right
def turnRight():
    stop()
    wiringpi.softPwmWrite(WHEEL_1,0)
    wiringpi.softPwmWrite(WHEEL_2,FAST)
    wiringpi.softPwmWrite(WHEEL_3,0)
    wiringpi.softPwmWrite(WHEEL_4,SLOW)
    wiringpi.delay(200)
    stop()

# car turn left
def turnLeft():
    stop()
    wiringpi.softPwmWrite(WHEEL_1,FAST)
    wiringpi.softPwmWrite(WHEEL_2,0)
    wiringpi.softPwmWrite(WHEEL_3,SLOW)
    wiringpi.softPwmWrite(WHEEL_4,0)
    wiringpi.delay(200)
    stop()
```

```
# car infr check
def infrCheck():
    left_infr = wiringpi.digitalRead(INFR_LEFT)
    right_infr = wiringpi.digitalRead(INFR_RIGHT)
    if left_infr == 0 and right_infr == 0:
        return 0
    elif left_infr == 1 and right_infr == 0:
        return 1
    elif left_infr == 0 and right_infr == 1:
        return 2
    else:
        return 3


def ultraCheck():
    wiringpi.digitalWrite(TRIG,1)
    time.sleep(1e-5)
    wiringpi.digitalWrite(TRIG,0)
    while wiringpi.digitalRead(ECHO) == 0:
        pass
    t1 = time.time()
    while wiringpi.digitalRead(ECHO) == 1:
        pass
    t2 = time.time()
    deltaTime = (t2-t1) * 1000000
    distance = deltaTime * 17 / 1000.0
    print('Current distance =',distance,'cm')
    if distance - LIM < EPS:
        return 0
    else:
        return 1
```

```
def main():
    stop()
    frontState = infrCheck()
    if frontState == 0:
        print('Current left == 0, right == 0')
        rightState = ultraCheck()
        if rightState == 1:
            turnRight()
        else:
            turnLeft()
    elif frontState == 1:
        print('Current left == 1, right == 0')
        turnLeft()
    elif frontState == 2:
        print('Current left == 0, right == 1')
        turnRight()
    else:
        print('Current left == 1, right == 1')
        goForward()
        wiringpi.delay(200)

if __name__=='__main__':
    init()
    while (True):
        main()
```

提交文件说明

|- report.doc

|- report.pdf

|- image

    |- RaspberryPi 小车.jpg

|- src

    |- main.py

|- video

    |- RaspberryPi 小车.mp4