

# Chapter 5 小车行驶

## 实验目的

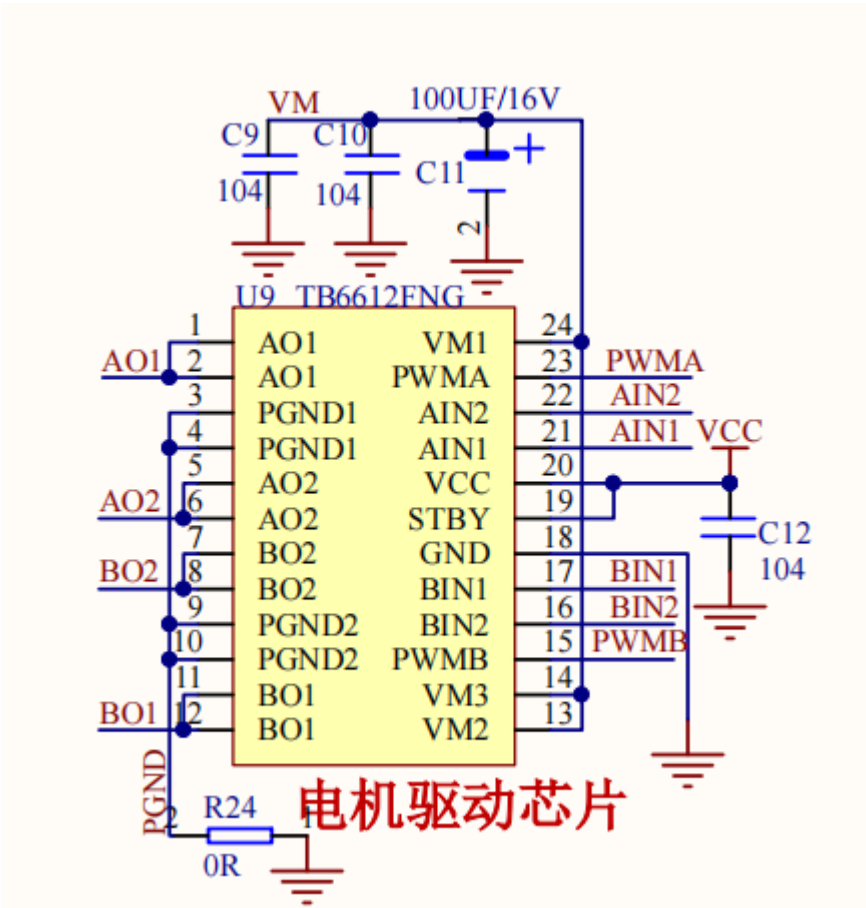
远程登录树莓派之后，运行程序，小车按照自定义的模式行驶。

## 实验原理

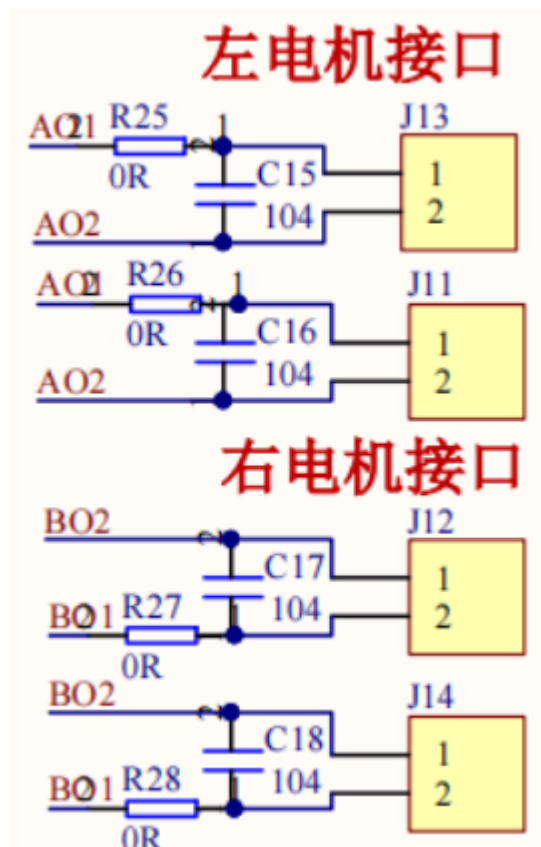
对于4路直流减速电机的控制我们采用的是TB6612FNG驱动芯片来驱动电机。

TB6612FNG 是东芝半导体公司生产的一款直流电机驱动器件，它具有大电流MOSFET-H 桥结构，双通道电路输出，可同时驱动 2路电机。也许大家更熟悉 L298N，其实这两者的使用基本一致的。而且，相比 L298N 的热耗性和外围二极管续流电路，它无需外加散热片，外围电路简单，只需外接电源滤波电容就可以直接驱动电机，利于减小系统尺寸。对于PWM信号输入频率范围，高达100KHz的频率更足以满足我们大部分的要求。

## 硬件原理



图一：树莓派主控板电路图



图二：电机模块接口

TB6612FNG能够独立双向控制2路直流电机，由AO控左侧两个电机，BO控制右侧两个电机。

## 引脚功能

	引脚	定义
	VM	驱动电压输入端（4.5–10V）
	VCC	逻辑电平输入端（2.7–5.5V）
	GND	电源地端
	STBY	正常工作/待机状态控制输入端
1路电机	PWMA	PWM信号输入端
	AIN1	电机控制模式输入端
	AIN2	
	A01	电机驱动输出端
	A02	
2路电机	PWMB	PWM信号输入端
	BIN1	电机控制模式输入端
	BIN2	
	B01	电机驱动输出端
	B02	

图三：引脚定义

通过控制驱动芯片的AIN1,AIN2,BIN1,BIN2,PWMA,PWMB的电平高低来控制电机的正转，反转，停止。通过PWMA和PWMB引脚改变占空比来控制小车的速度。

引脚电平与电机转动状态关系如下：

IN1	IN2	PWM	输出状态
H	H	H/L	制动
L	H	H	反转
L	H	L	制动
H	L	H	正转
H	L	L	制动
L	L	H	停止
-	-	-	待机

PWM不是调节电流的。PWM的意思是脉宽调节，也就是调节方波高电平和低电平的时间比，一个20%占空比波形，会有20%的高电平时间和80%的低电平时间，而一个60%占空比的波形则具有60%的高电

平时间和40%的低电平时间，占空比越大，高电平时间越长，则输出的脉冲幅度越高，即电压越高。如果占空比为0%，那么高电平时间为0，则没有电压输出。如果占空比为100%，那么输出全部电压。

## 代码编写

```
#-*- coding:UTF-8 -*-
import RPi.GPIO as GPIO
import time

#小车电机引脚定义，A控制左侧，B控制右侧。
AIN1 = 21
AIN2 = 20

BIN1 = 26
BIN2 = 19

PWMA = 16
PWMB = 13

#设置GPIO口为BCM编码方式
GPIO.setmode(GPIO.BCM)

#忽略警告信息
GPIO.setwarnings(False)

#电机引脚初始化操作
def motor_init():
    global pwm_PWMA
    global pwm_PWMB
    global delaytime

    #初始化引脚
    GPIO.setup(PWMA,GPIO.OUT,initial=GPIO.HIGH)
    GPIO.setup(AIN2,GPIO.OUT,initial=GPIO.LOW)
    GPIO.setup(AIN1,GPIO.OUT,initial=GPIO.LOW)

    GPIO.setup(PWMB,GPIO.OUT,initial=GPIO.HIGH)
    GPIO.setup(BIN2,GPIO.OUT,initial=GPIO.LOW)
    GPIO.setup(BIN1,GPIO.OUT,initial=GPIO.LOW)

    #设置pwm引脚和频率为2000hz
    pwm_PWMA = GPIO.PWM(PWMA, 2000)
    pwm_PWMB = GPIO.PWM(PWMB, 2000)
    pwm_PWMA.start(0)
    pwm_PWMB.start(0)

#小车前进
def run(delaytime):
    GPIO.output(AIN2, GPIO.HIGH)
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(BIN2, GPIO.HIGH)
    GPIO.output(BIN1, GPIO.LOW)
```

```
pwm_PWMA.ChangeDutyCycle(5)
pwm_PWMB.ChangeDutyCycle(5)
time.sleep(delaytime)
```

#小车后退

```
def back(delaytime):
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(AIN1, GPIO.HIGH)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.HIGH)
    pwm_PWMA.ChangeDutyCycle(5)
    pwm_PWMB.ChangeDutyCycle(5)
    time.sleep(delaytime)
```

#小车左转

```
def left(delaytime):
    #左侧轮子制动，右侧轮子正转，小车左转
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(BIN2, GPIO.HIGH)
    GPIO.output(BIN1, GPIO.LOW)
    pwm_PWMA.ChangeDutyCycle(0)
    pwm_PWMB.ChangeDutyCycle(5)
    time.sleep(delaytime)
```

#小车右转

```
def right(delaytime):
    #右侧轮子制动，左侧轮子正转，小车右转
    GPIO.output(AIN2, GPIO.HIGH)
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.LOW)
    pwm_PWMA.ChangeDutyCycle(5)
    pwm_PWMB.ChangeDutyCycle(0)
    time.sleep(delaytime)
```

#小车原地左转

```
def spin_left(delaytime):
    #左侧轮子反转，右侧轮子正转，小车原地左转
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(AIN1, GPIO.HIGH)
    GPIO.output(BIN2, GPIO.HIGH)
    GPIO.output(BIN1, GPIO.LOW)
    pwm_PWMA.ChangeDutyCycle(5)
    pwm_PWMB.ChangeDutyCycle(5)
    time.sleep(delaytime)
```

#小车原地右转

```
def spin_right(delaytime):
    #右侧轮子反转，左侧轮子正转，小车原地左右转
    GPIO.output(AIN2, GPIO.HIGH)
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.HIGH)
```

```
pwm_PWMA.ChangeDutyCycle(5)
pwm_PWMB.ChangeDutyCycle(5)
time.sleep(delaytime)
```

#小车停止

```
def brake(delaytime):
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.LOW)
    pwm_PWMA.ChangeDutyCycle(0)
    pwm_PWMB.ChangeDutyCycle(0)
    time.sleep(delaytime)
```

#延时2s

```
time.sleep(2)
```

#try/except语句用来检测try语句块中的错误，  
#从而让except语句捕获异常信息并处理。

```
try:
    motor_init()
    #while True:
    run(0.5)
    back(1)
    left(2)
    right(2)
    spin_left(3)
    spin_right(3)
    brake(1)
except KeyboardInterrupt:
    print("小车电机驱动出现异常！！")
pwm_PWMA.stop()
pwm_PWMB.stop()
GPIO.cleanup()
```