

智能小车车灯实验

目录

一、 实验概述.....	2
二、 实验器材.....	2
三、 知识要点.....	2
四、 实验原理.....	2
五、 代码编写.....	3
六、 硬件连接及运行效果.....	5
七、 思维发散及课后作业.....	6
官 网: www.xiao-r.com	6
论 坛: www.wifi-robots.com	6
官方商城: wifi-robots.taobao.com	6
微信公众号:	6

一、实验概述

本文主要介绍小车电机实验。包括前进调试、前后左右综合实验、花式动作。

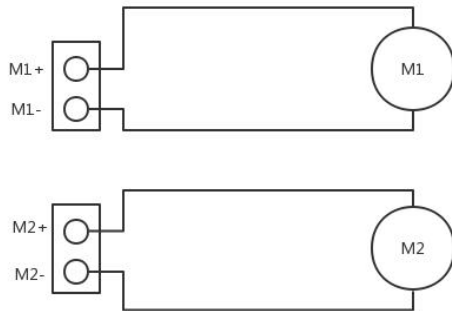
二、实验器材



三、知识要点

- 1、智能小车转向方式 PWM 调速
- 2、PWM 调速
- 3、Arduino 模拟输出

四、实验原理



Arduino 智能小车有四个电机马达，我们将左侧的两个电机并联起来就形成了一组电机，那么左侧的两个电机运动就是一致的，这样就可以把左侧的两个电机看作是上图中的 M1，同理右侧的两个也是一样，看作是 M2

运动状态分析

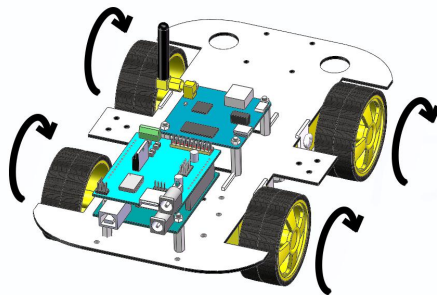
上节我们讲到主芯片通过 4 个数字输出 IO 引脚 IN1、IN2、IN3、IN4，2 个模拟输出 IO 引脚 ENA、ENB 连接 74HC244 隔离后输出对应信号 I1、I2、I3、I4，EA、EB 再连接

至 TB6612 电机驱动芯片并让电机驱动芯片输出对应的电机驱动电压 OUT1、OUT2、OUT3、OUT4，以及两组电机输出端的电压(即转速)。下面我们就分析各运动状态这些引脚对应的状态。

转向 状态	IN1	IN2	IN3	IN4	E NA	EN B	I1	I2	I3	I4	EA	E B	OU T 1	OU T2	OU T3	OU T4
前进	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0

注：1 代表高电平，0 代表低电平

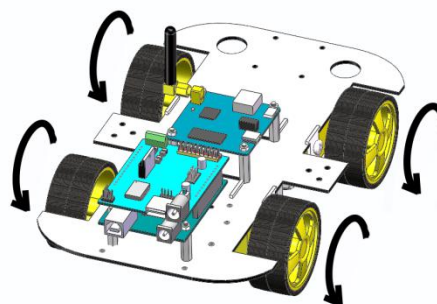
前进



转向 状态	IN1	IN2	IN3	IN4	E NA	EN B	I1	I2	I3	I4	EA	E B	OU T 1	OU T2	OU T3	OU T4
前进	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0	1

注：1 代表高电平，0 代表低电平

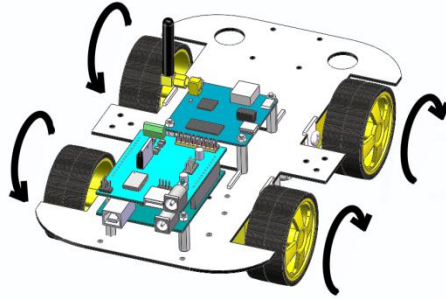
后退



转向 状态	IN1	IN2	IN3	IN4	E NA	EN B	I1	I2	I3	I4	EA	E B	OU T 1	OU T2	OU T3	OU T4
前进	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0

注：1 代表高电平，0 代表低电平

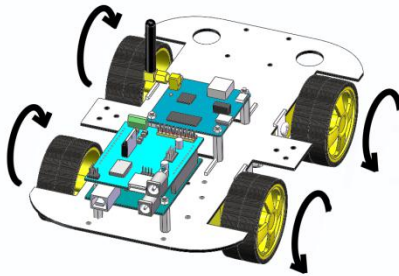
左转



转向 状态	IN1	IN2	IN3	IN4	E NA	EN B	I1	I2	I3	I4	EA	E B	OU T 1	OU T2	OU T3	OU T4
前进	1	0	0	1	1	1	1	0	0	1	1	1	1	0	0	1

注：1 代表高电平，0 代表低电平

右转



左转弯原理

很多人不理解，为什么小车都是固定方向的，也没有方向盘可以转动，那怎么可以让小车转向呢，其实这个可以参考坦克的转向方式，利用两边轮子的速度差，当一边轮子速度比另一边轮子速度小的时候，车子在前进过程中就会向速度小的一边偏转，因此我们通过控制两个使能引脚电压输出 ENA、ENB 来控制小车的转速及方向。

上图我们转向方式更为简单，我们固定了小车俩边的转速，只改变俩边的电机轮子的转动方向，当需要左转或右转的时候，只需要将左转或右转那边的轮子的转动方向设置为往后转即可，这样的话小车就可以原地转向。

五、代码编写

```
#coding:utf-8
```

```
import os
```

```
import time
```

```
import RPi.GPIO as GPIO
```

```
#####
```

```
#####信号引脚定义#####
```

```
#####
```

```
GPIO.setmode(GPIO.BCM)
```

```
#####电机驱动接口定义#####
```

```
ENA = 13    ##L298 使能 A
```

```
ENB = 20    ##L298 使能 B
```

```
IN1 = 19 ##电机接口 1
```

```
IN2 = 16 ##电机接口 2
```

```
IN3 = 21 ##电机接口 3
```

```
IN4 = 26 ##电机接口 4
```

```
#####电机初始化为 LOW#####
```

```
GPIO.setup(ENA,GPIO.OUT,initial=GPIO.LOW)
```

```
GPIO.setup(ENB,GPIO.OUT,initial=GPIO.LOW)
```

```
GPIO.setup(IN1,GPIO.OUT,initial=GPIO.LOW)
```

```
GPIO.setup(IN2,GPIO.OUT,initial=GPIO.LOW)
```

```
GPIO.setup(IN3,GPIO.OUT,initial=GPIO.LOW)
```

```
GPIO.setup(IN4,GPIO.OUT,initial=GPIO.LOW)
```

```
#####电机前进函数#####
```

```
def Motor_Forward():
```

```
print 'motor forward'
GPIO.output(ENA,True)
GPIO.output(ENB,True)
GPIO.output(IN1,True)
GPIO.output(IN2,False)
GPIO.output(IN3,True)
GPIO.output(IN4,False)
#####电机电机后退函数#####
def Motor_Backward():
    print 'motor_backward'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,False)
    GPIO.output(IN2,True)
    GPIO.output(IN3,False)
    GPIO.output(IN4,True)
#####电机电机左转函数#####
def Motor_TurnLeft():
    print 'motor_turnleft'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,True)
    GPIO.output(IN2,False)
    GPIO.output(IN3,False)
    GPIO.output(IN4,True)
#####电机电机右转函数#####
def Motor_TurnRight():
    print 'motor_turnright'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,False)
    GPIO.output(IN2,True)
    GPIO.output(IN3,True)
    GPIO.output(IN4,False)
```


#####电机电机停止函数#####

```
def Motor_Stop():
```

```
    print 'motor stop'
```

```
    GPIO.output(ENA,True)
```

```
    GPIO.output(ENB,True)
```

```
    GPIO.output(IN1,False)
```

```
    GPIO.output(IN2,False)
```

```
    GPIO.output(IN3,False)
```

```
    GPIO.output(IN4,False)
```

```
while True:
```

```
    Motor_Forward()        ##前进
```

```
    time.sleep(1)          ##等待 1 秒
```

```
    Motor_Backward()       ##后退
```

```
    time.sleep(1)          ##等待 1 秒
```

```
    Motor_TurnLeft()       ##左转
```

```
    time.sleep(1)          ##等待 1 秒
```

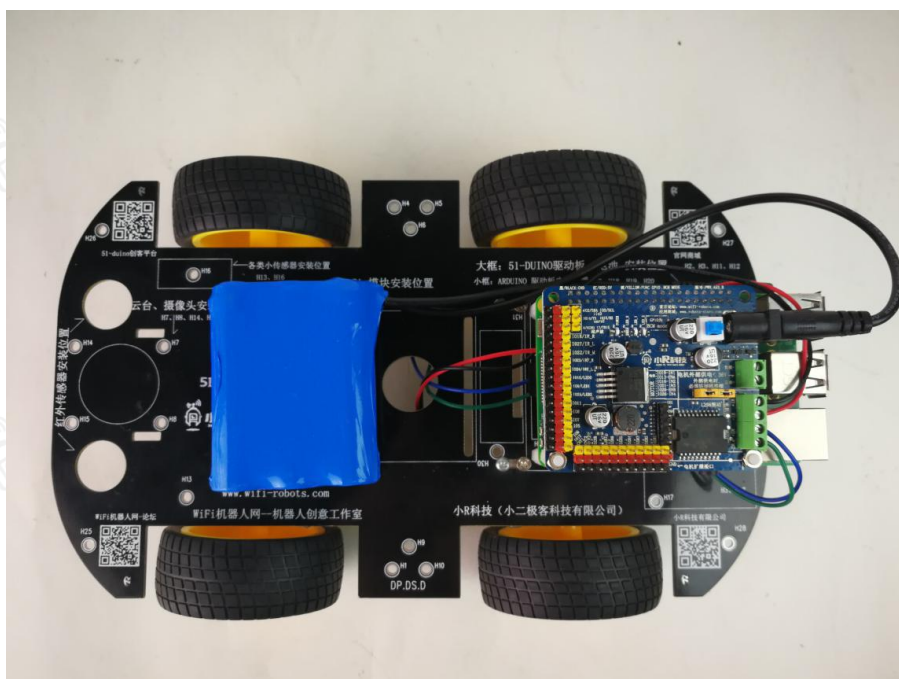
```
    Motor_TurnRight()      ##右转
```

```
    time.sleep(1)          ##等待 1 秒
```

```
    Motor_Stop()           ##停止
```

```
    time.sleep(1)          ##等待 1 秒
```

六、硬件连接及运行效果



七、思维发散及课后作业

整个实验中我们没有对小车进行调速处理，只是简单的给与电机最大速度运转或不转，我们可以使用 RPI.GPIO 模块的脉宽调制（PWM）功能对小车进行调速

创建一个 PWM 实例：

```
p = GPIO.PWM(channel, frequency)
```

启用 PWM：

```
p.start(dc) # dc 代表占空比（范围：0.0 <= dc <= 100.0）
```

更改频率：

```
p.ChangeFrequency(freq) # freq 为设置的新频率，单位为 Hz
```

更改占空比：

```
p.ChangeDutyCycle(dc) # 范围：0.0 <= dc <= 100.0
```

停止 PWM：

```
p.stop()
```

以下为使 LED 每两秒钟闪烁一次的示例：

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)
```

```
p = GPIO.PWM(12, 0.5)
```

```
p.start(1)
```

```
input('点击回车停止：') # 在 Python 2 中需要使用 raw_input
```

```
p.stop()
```

```
GPIO.cleanup()
```




官网: www.xiao-r.com

论坛: www.wifi-robots.com

官方商城: wifi-robots.taobao.com

微信公众号:

