



PART THREE

3

代码及运行展示

基础功能

```
//定义马达的四根引脚
int leftMotor1 = 4;
int leftMotor2 = 5;
int rightMotor1 = 6;
int rightMotor2 = 7;
void setup() {
    //设置控制电机的引脚为输出状态
    pinMode(leftMotor1, OUTPUT);
    pinMode(leftMotor2, OUTPUT);
    pinMode(rightMotor1, OUTPUT);
    pinMode(rightMotor2, OUTPUT);
}
```

```
void loop() {
    int cmd;
    //依次执行向前、向后、向左、向右
    和停止五个运动状态
    for(cmd=0; cmd<5; cmd++)
    {
        motorRun(cmd);
        delay(2000);
        //每个命令执行2s
    }
}
```

基础功能

```
void motorRun(int cmd)
{
    switch(cmd){
        case FORWARD:{...}
        case BACKWARD:{...}
        case TURNLEFT:{...}
        case TURNRIGHT:{...}
        default:{...} //停止
    }
}
```

```
case FORWARD:
    digitalWrite(leftMotor1, LOW);
    digitalWrite(leftMotor2, HIGH);
    digitalWrite(rightMotor1, LOW);
    digitalWrite(rightMotor2, HIGH);
    break;

case TURNLEFT:
    digitalWrite(leftMotor1, HIGH);
    digitalWrite(leftMotor2, LOW);
    digitalWrite(rightMotor1, LOW);
    digitalWrite(rightMotor2, HIGH);
    break;
```

避障功能

```
Servo myservo1; //左侧马达
Servo myservo2; //右侧马达
void setup()
{
    myservo1.attach(7);
    //连接左马达到端口7
    myservo2.attach(9);
    //连接右马达到端口9
}
```

```
void loop()
{
    val = analogRead(GP2D12);
    //读取红外测距传感器数据
    distance_float=2547.8/
    ((float)val*0.49-10.41)-0.42;
    //转化为浮点型数值
    start();
    //开始执行主要函数
}
```

避障功能

```
void start()  
{  
    //如果距离小于20，则避让  
    if (distance_float<=20 &&  
distance_float>0)  
    {  
        avoid();  
    }  
    else //否则继续前进  
    {  
        advance();  
    }  
}
```

```
void avoid()  
{  
    stop(); //停止0.2s  
    delay(200);  
    back(); //后退0.8s  
    delay(800);  
    //小车随机向左或者向右转动  
    turn();  
}
```

避障功能

```
void advance() //前进
{
    angle1=180;
    angle2=0;
    myservo1.write(angle1);
    myservo2.write(angle2);
}

void back() {...} //颠倒参数即可
void left() {...} //同理
void right() {...} //同理
```

```
void stop()//小车停止
{
    angle1=93;
    angle2=93;
    myservo1.write(angle1);
    myservo2.write(angle2);
}

/* Servo.write(angle)
 * 0—full speed in one
 * direction; 180—full speed
 * in the other; ~90—no
 * movement
 */
```

寻线功能

```
int IR_left=2; //左侧寻线传感器
int IR_right=4; //右侧寻线传感器
void setup()
{
    myservo1.attach(7);
    myservo2.attach(9);
    pinMode(IR_right, INPUT);
    //左侧传感器设为输入
    pinMode(IR_left, INPUT);
    //右侧传感器设为输入
}
```

```
void loop()
{
    //读寻线传感器的信号
    Right=digitalRead(IR_right);
    Left=digitalRead(IR_left);
    react(Right, Left);
    //对两个信号做进一步处理
}
```

寻线功能

```
void react(int Right, int Left)
{
    if(Right==LOW && Left==LOW)           //如果右、左两个红外为低电平
        advance();                       //说明小车没有偏离黑色跑道
    else if(Right==HIGH && Left==LOW)       //如果右为高电平，说明小车向右偏离
        left();                          //如果左为高电平，说明小车向左偏离
    else if(Right==LOW && Left==HIGH)       //否则左右都偏离
        right();
    else
        stop();
}
```

//转向函数**advance**等同避障功能



PART FOUR

4

问题分析



问题分析

✔ 小车轮子动态与程序不符

uno板和驱动板引脚对应模糊、杜邦线接触不良、电压不够。考虑更换更稳健的硬件，配备电压较高的航模电池。

✔ 小车两轮速度不协调

硬件的物理原因造成力度不均，可以通过设置速度比率，调节两轮速度（电压）达到平衡



PART FIVE

5

未来展望

系统设计

轿车系统

- ❖ 蓝牙控制
- ❖ 语音控制
- ❖ 自动停车入位
- ❖ 识别红绿灯
- ❖ 识别限速标志

表演系统

- ❖ 双星造型
- ❖ 极限符号造型
- ❖ 分组造型
- ❖ 模仿造型

赛车系统

- ❖ 两辆车追逐
- ❖ 沿椭圆轨道
- ❖ 后车外道超车
- ❖ 超车后继续在轨道上行驶

2018 10.27

谢谢观看

