实验一

要求：连接发光二极管基础电路，创建一个SOS信号灯，

目的：学习数字信号输出OUTPUT和digitalWrite()函数、

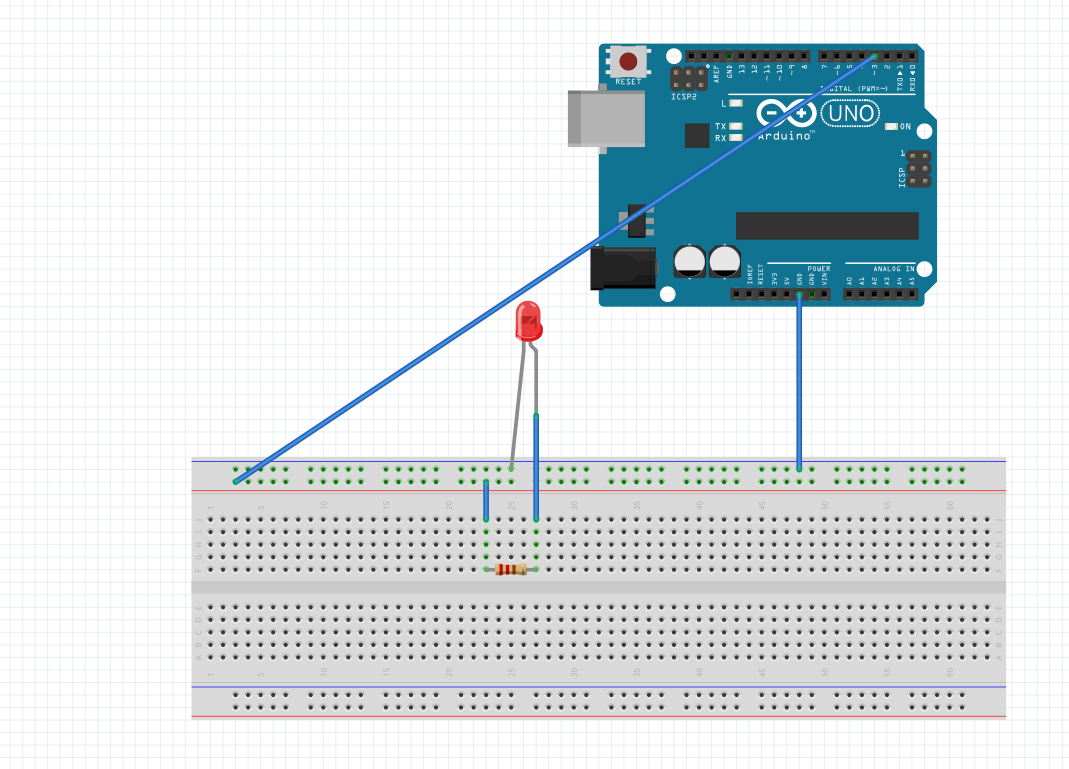
实验思路：

只需使某个引脚提供高电压执行较短时间后熄灭重复三次，而后再提供高电压较长时间后熄灭重复再次即可形成一次SOS信号，使用loop()函数即可实现周期性重复。

经计算为使得LED在工作电压需要提供220Ω电阻。

接线：

Pin3连接LED长引脚及220Ω电阻，短引脚接地



int delaytime = 500;//创建熄灭时间变量

int longSunTime = 1000;//创建长闪光变量

int ShortSunTime = 500;//创建短闪光变量

void setup() {

  pinMode(3,OUTPUT);//初始化pin3状态

}

void loop() {

for(int i=1;i<=3;i++){//短闪三次光

 digitalWrite(LED\_BUILTIN, HIGH);

 delay(ShortSunTime);//短光

 digitalWrite(LED\_BUILTIN, LOW);

 delay(delaytime);//熄灭

}

for(int i=1;i<=3;i++){//长闪三次光

 digitalWrite(LED\_BUILTIN, HIGH);

 delay(longSunTime);//长光

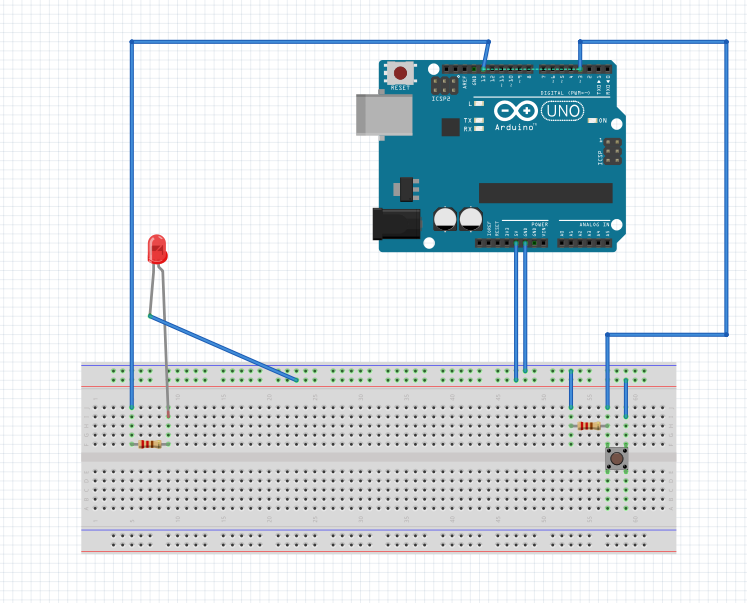
 digitalWrite(LED\_BUILTIN, LOW);

 delay(delaytime);//熄灭

}

}

**实验二**

要求：设计电路编写程序，按下开关点亮LED，松开开关LED熄灭

目的：学习数字量输入digital INPUT和digitalRead()函数

实验思路：

设置引脚为输入模式执行digitalRead可获取引脚电平状态，结合pushbotton及10kΩ电阻可使得arduino获取到pushbotton是否被摁下，从而进行逻辑运算改变给LED供电的引脚的状态，实现开关控制LED。

经计算为使得LED在工作电压需要提供220Ω电阻。

接线：

Pin13连接LED长引脚及220Ω电阻，短引脚接地

Pin2连接按键开关一侧，与10kΩ电阻并联接地

const int buttonPin = 2;//设置buttonPin变量

const int ledPin = 13;//设置ledPin变量

void setup() {//初始化引脚状态以及串口波特率

  // put your setup code here, to run once:

  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT);

  Serial.begin(9600);

}

void loop() {

  // put your main code here, to run repeatedly:

 int buttonState = digitalRead(buttonPin);//读取buttonPin引脚电平状态

 if (buttonState == HIGH) {

   digitalWrite(ledPin,HIGH);//高电平下对LED输出高电平

 }

 else {

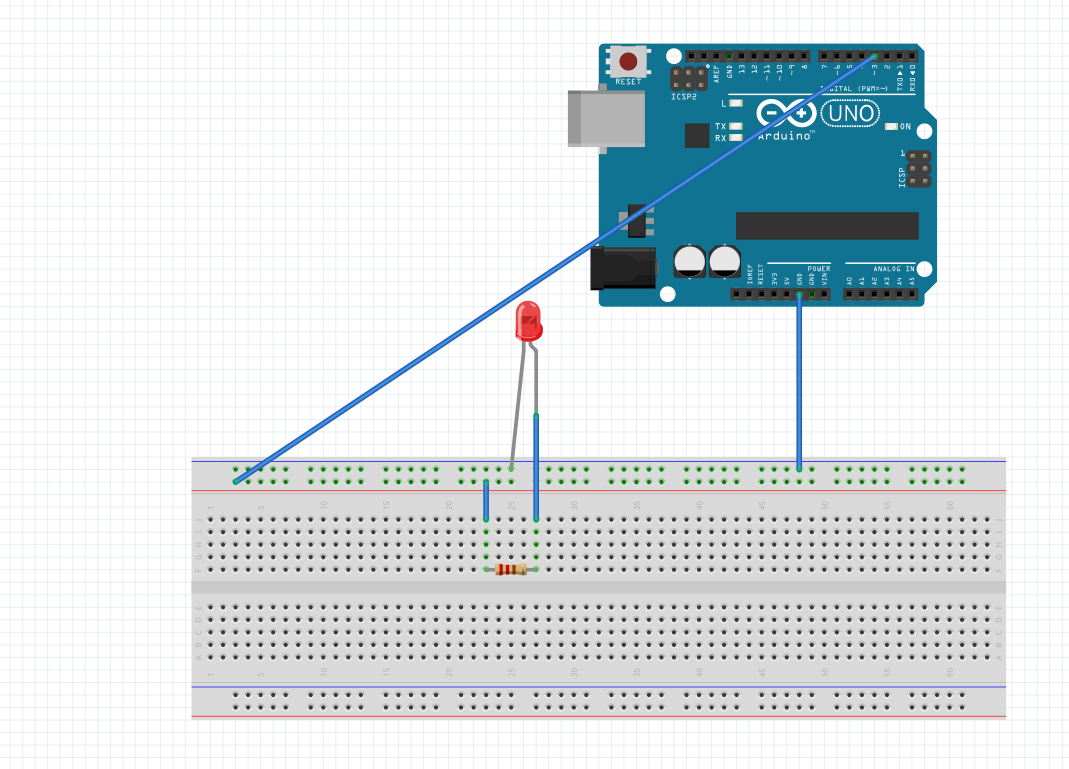
 digitalWrite(ledPin, LOW);

 }

Serial.println(buttonState);//低电平下对LED输出低电平

}

**实验三**

要求：利用程序调节LED的光强，由暗到亮，再从亮到暗

目的：学习analogWrite()函数，模拟量（PWM）输出

实验思路：

利用PWM使得LED逐步增亮及逐步变暗

经计算为使得LED在工作电压需要提供220Ω电阻。

接线：

Pin5连接LED长引脚及220Ω电阻，短引脚接地

int ledPin = 3;//创建ledPin变量

void setup() {

  // put your setup code here, to run once:

  pinMode(ledPin,OUTPUT);//初始化引脚状态

}

void loop() {

  // put your main code here, to run repeatedly:

  for(int i = 0;i<=250;i+=5){//通过循环逐步增加PWM输出电压

    analogWrite(ledPin,i);

    delay(50);

  }

  for(int i = 255;i>5;i-=5){//通过循环逐步降低PWM输出电压

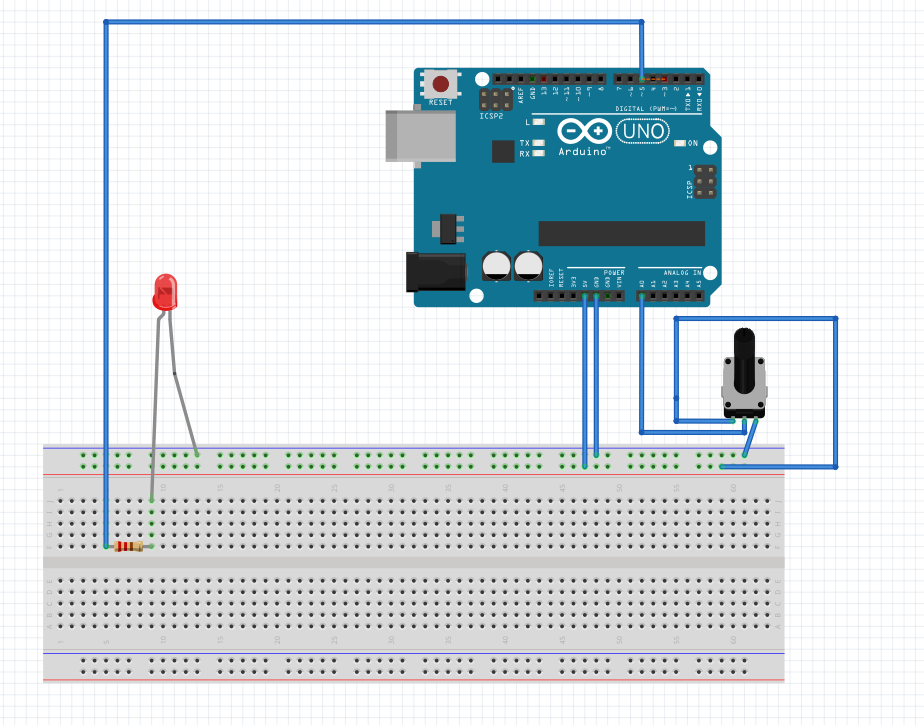
    analogWrite(ledPin,i);

    delay(50);

  }

}

**实验四**

要求：以读取电位器的分压值来调节LED亮度

目的：学习analogRead()函数

实验思路：

利用analogread函数结合电位器读取引脚电压变化，从而结合LED改变LED亮度。

经计算为使得LED在工作电压需要提供220Ω电阻。

接线：

Pin3连接LED长引脚及220Ω电阻，短引脚接地

#define ledPin 5//控制LED

#define potPin A0//读取电位器引起的电压变化

void setup() {

  // put your setup code here, to run once:

  pinMode(potPin, INPUT);// 改变A0引脚为输入模式

  pinMode(ledPin, OUTPUT);//改变第五引脚为输出模式

  Serial.begin(9600);//初始化波特率

}

void loop() {

  // put your main code here, to run repeatedly:

  int potValue = analogRead(A0);//创建状态变量

  analogWrite(5,potValue/4 );

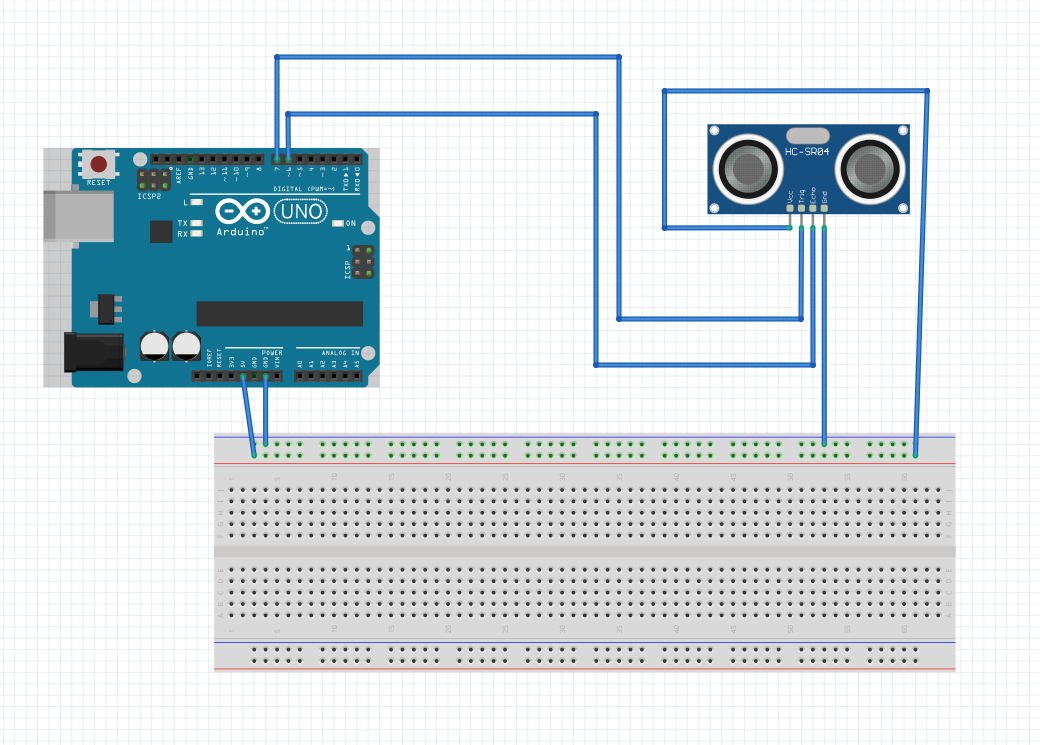
  Serial.print(potValue); //打印到串口监视器

  Serial.print(" ");

  Serial.println(potValue/4);// 将输出值打印

}

**实验六**

要求：学习利用pulseIn()函数和超声波传感器来完成测距

目的：学习pulseIn()和SR04超声测距模块

实验思路：

使用Arduino数字引脚给SR04模块的Trig引脚发送至少10m的高电平信号，触发SR04的发射探头发出8个40kHz的超声波脉冲，若有信号返回，SR04的Echo引脚会输出高电平，高电平持续时间与检测距离成比例。读取持续时间通过公式即可得到距离。

接线：

Pin7接trig，pin6接echo，VCC接5V，gnd接地

int trigPin = 7;    //Trig

int echoPin = 6;    //Echo

long duration, cm;

void setup() {

  //初始化串口

  Serial.begin (9600);

  //更改引脚模式

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

}

void loop()

{

  digitalWrite(trigPin, LOW);

  delayMicroseconds(5);//稳定电位

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);//给10ms脉冲

  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH,50000);//读回响脉冲时间

  cm = duration/58;//计算距离，单位cm

  Serial.print(cm);//打印计算结果至串口

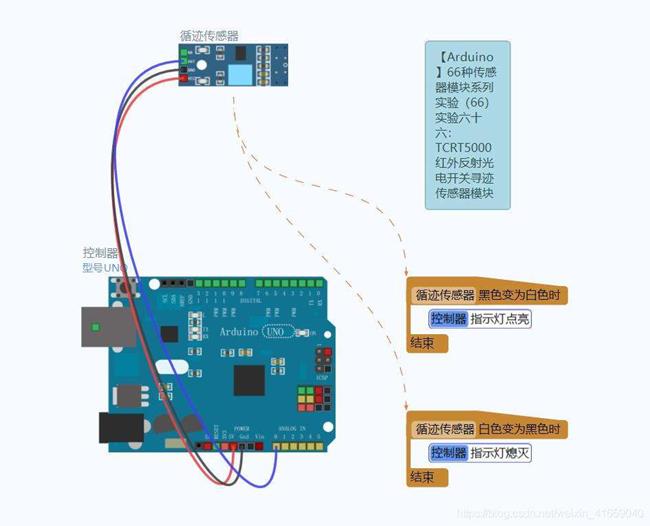
  Serial.print("cm");

  Serial.println();

  delay(100);//延迟100ms

}

**实验七**



——引用自PPT

要求：利用红外传感器检测黑白线

目的：学习TCRT5000模块

实验思路：

利用analogread()函数读取TCRT5000返回的电压值

接线：

D0接A0,GND接地，VCC接5V

#define readPin A0//定义输入引脚

void setup() {

  // put your setup code here, to run once:

pinMode(readPin, INPUT);//初始化引脚

Serial.begin(9600);//初始化波特率

}

void loop() {

  // put your main code here, to run repeatedly:

int readValue = analogRead(readPin);//定义读取电压值变量

if (readValue>=1000) {//逻辑判断黑白线

Serial.println('black');

}

else {

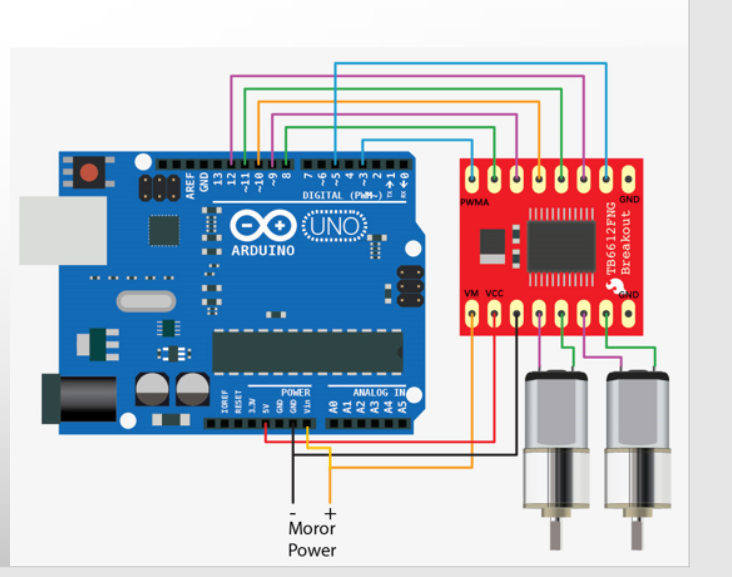
Serial.println('white');

}

}

}

**实验八**



——引用自PPT

要求：利用电机驱动模块TB6612FNG驱动车轮旋转

目的：学习电机驱动模块TB6612FNG

实验思路：

根据逻辑真值表对电机驱动行为进行封装，进而对车辆运动行为进行封装

接线：

Pin3 PWMA

Pin8 AIN2

Pin9 AIN1

Pin10 STBY

Pin11 BIN1

Pin12 BIN2

Pin5 PWMB

5V VCC

Vin VM

#define PWMA 3

#define AIN2 8

#define AIN1 9

#define STBY 10

#define BIN1 11

#define BIN2 12

#define PWMB 5

void motor\_full\_move(int IN1,int IN2){

  //define a move function

  digitalWrite(STBY, HIGH);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW);

  if(IN1 == AIN1){

  analogWrite(PWMA,80);

  }

  if(IN1 == BIN1){

  analogWrite(PWMB,80);

  }

}

void motor\_backmove(int IN1,int IN2){

  //define a backmove function

  digitalWrite(STBY, HIGH);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, HIGH);

  if(IN1 == AIN1){

    analogWrite(PWMA,156);

  }

  if(IN1 == BIN1){

    analogWrite(PWMB,156);

  }

}

void motor\_move(int IN1,int IN2){

  //define a backmove function

  digitalWrite(STBY, HIGH);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW);

  if(IN1 == AIN1){

  analogWrite(PWMA,255);

  }

  if(IN1 == BIN1){

  analogWrite(PWMB,255);

  }

}

void motor\_stop(int IN1,int IN2){

  //define a stop function

  digitalWrite(STBY, HIGH);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW);

  if(IN1 == AIN1){

  digitalWrite(PWMA,LOW);

  }

  if(IN1 == BIN1){

  digitalWrite(PWMB,LOW);

  }

}

void car\_right(){

  //define a move to left function

  //左电机快转，右电机慢转

  motor\_backmove(BIN1,BIN2);

  motor\_full\_move(AIN1,AIN2);

}

void car\_left(){

  //define a move to right function

  //左电机慢转，右电机快转

  motor\_backmove(AIN1,AIN2);

  motor\_full\_move(BIN1,BIN2);

}

void car\_stop(){

  //define a stop function

  motor\_stop(AIN1,AIN2);

  motor\_stop(BIN1,BIN2);

}

void car\_full\_move(){

  //define a function let car full-speed move

  motor\_full\_move(AIN1, AIN2);

  motor\_full\_move(BIN1, BIN2);

}

void car\_wait(){

  digitalWrite(STBY, LOW);

}