POTENCIOMETRO\_LEDS

Montaremos un circuito con 5 leds y sus respectivas resistencias, y dependiendo del potenciómetro encenderemos 0,1,2,..5 leds.

Diagrama

Descripción generada automáticamente

int Led = 0 ;

void setup()

{

  Serial.begin(9600);

  pinMode(2, OUTPUT);

  pinMode(3, OUTPUT);

  pinMode(4, OUTPUT);

  pinMode(5, OUTPUT);

  pinMode(6, OUTPUT);

  digitalWrite(2, HIGH);

  digitalWrite(3, HIGH);

  digitalWrite(4, HIGH);

  digitalWrite(5, HIGH);

  digitalWrite(6, HIGH);

}

void loop()

{

  int lectura = analogRead(A0) ;

  Serial.println( lectura);

  delay(200) ;

  Led = map(lectura, 1005, 1023, 0, 5);

  Serial.print("Valor del Potenciometro = ");

  Serial.print(lectura);

  Serial.print(" Numero de Leds = ");

  Serial.println(Led);

  if ( Led == 0){

    digitalWrite(2, HIGH);

    digitalWrite(3, HIGH);

    digitalWrite(4, HIGH);

    digitalWrite(5, HIGH);

    digitalWrite(6, HIGH);

  }

  if ( Led == 1){

    digitalWrite(2, LOW);

    digitalWrite(3, HIGH);

    digitalWrite(4, HIGH);

    digitalWrite(5, HIGH);

    digitalWrite(6, HIGH);

  }

  if ( Led == 2){

    digitalWrite(2, LOW);

    digitalWrite(3, LOW);

    digitalWrite(4, HIGH);

    digitalWrite(5, HIGH);

    digitalWrite(6, HIGH);

  }

  if ( Led == 3){

    digitalWrite(2, LOW);

    digitalWrite(3, LOW);

    digitalWrite(4, LOW);

    digitalWrite(5, HIGH);

    digitalWrite(6, HIGH);

  }

  if ( Led == 4){

    digitalWrite(2, LOW);

    digitalWrite(3, LOW);

    digitalWrite(4, LOW);

    digitalWrite(5, LOW);

    digitalWrite(6, HIGH);

  }

  if ( Led == 5){

    digitalWrite(2, LOW);

    digitalWrite(3, LOW);

    digitalWrite(4, LOW);

    digitalWrite(5, LOW);

    digitalWrite(6, LOW);

  }

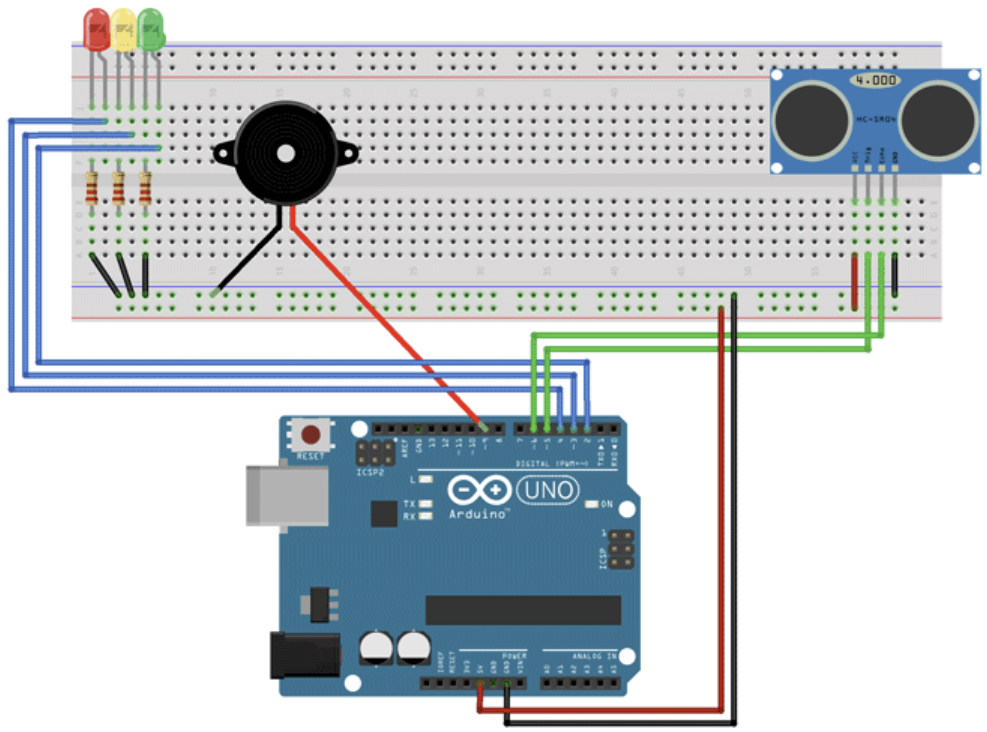
    delay(1000);

}

ALERTA

Implementar un sistema de alerta visual con el sensor ultrasónico Arduino.

Nos permitirá visualizar si estamos cerca o lejos de un obstáculo. Con 3 LEDs (verde, amarillo y rojo) conseguimos determinar si estamos lejos, cerca o en zona de peligro. También incorporaremos un buzzer para avisar cuando estemos en la zona de peligro.



#define trigPin 5

  #define echoPin 6

  #define ledrojo 4

  #define ledamarillo 3

  #define ledverde 2

  #define buzzer 9

  void setup()

  {

    Serial.begin (9600);

    pinMode(trigPin, OUTPUT);

    pinMode(echoPin, INPUT);

    pinMode(ledrojo, OUTPUT);

    pinMode(ledamarillo, OUTPUT);

    pinMode(ledverde, OUTPUT);

    pinMode(buzzer, OUTPUT);

    digitalWrite ( ledrojo , LOW) ;

    digitalWrite ( ledamarillo , LOW) ;

    digitalWrite ( ledverde , LOW) ;

    digitalWrite ( buzzer, LOW) ;

  }

  void loop()

  {

    long duracion, distancia;

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(trigPin, LOW);

    duracion = pulseIn(echoPin, HIGH) ;

    distancia = duracion / 2 / 29.1 ;

    Serial.println(String(distancia) + " cm.") ;

    int cerca = 10 ;

    int lejos = 20 ;

    if ( distancia < cerca)

    {

      digitalWrite ( ledrojo , HIGH) ;

      digitalWrite ( ledamarillo , LOW) ;

      digitalWrite ( ledverde , LOW) ;

      digitalWrite ( buzzer, HIGH) ;

    }

    else

    {

      if ( distancia > lejos){

        digitalWrite( ledrojo , LOW) ;

        digitalWrite ( ledamarillo , LOW) ;

        digitalWrite ( ledverde , HIGH) ;

        digitalWrite ( buzzer, LOW) ;

      }

      else

      {

        digitalWrite( ledrojo , LOW) ;

        digitalWrite ( ledamarillo , HIGH) ;

        digitalWrite ( ledverde , LOW) ;

        digitalWrite ( buzzer, LOW) ;

      }

    }

    delay (500) ;

  }

ALERTA TEMPERATURA

Vamos a modificar el ejercicio anterior (aún no tenemos motores o relés para accionarlos) para avisar de golpes de calor.

Por debajo de una determinada temperatura el led verde estará encendido, cuando este entre un intervalo (os lo dejo a vuestra elección) se encenderá el amarillo, y cuando supere un umbral encenderemos el rojo y el buzzer emitirá pitidos.

#define sensor 0

Imagen que contiene competencia de atletismo, baloncesto

Descripción generada automáticamente  #define ledrojo 4

  #define ledamarillo 3

  #define ledverde 2

  #define buzzer 9

  void setup()

  {

    Serial.begin (9600);

    pinMode(ledrojo, OUTPUT);

    pinMode(ledamarillo, OUTPUT);

    pinMode(ledverde, OUTPUT);

    pinMode(buzzer, OUTPUT);

    digitalWrite ( ledrojo , LOW) ;

    digitalWrite ( ledamarillo , LOW) ;

    digitalWrite ( ledverde , LOW) ;

    digitalWrite ( buzzer, LOW) ;

  }

  void loop()

  {

    int lectura = analogRead(sensor);

    float voltaje = 5.0 /1024 \* lectura ; // Atencion aqui

    // Si usais un LM35DZ vuestra formula será float temp = voltaje \* 100 ;

    // Si usais un TMP36 vuestra formula será float temp = voltaje \* 100 - 50 ;

    float temp = voltaje \* 100;

    Serial.println(temp) ;

    delay(1000);

    int frio = 15 ;

    int caliente = 30 ;

    if ( temp < frio)

    {

      digitalWrite ( ledrojo , LOW) ;

      digitalWrite ( ledamarillo , LOW) ;

      digitalWrite ( ledverde , HIGH) ;

      digitalWrite ( buzzer, LOW) ;

    }

    else

    {

      if ( temp > caliente){

        digitalWrite( ledrojo , HIGH) ;

        digitalWrite ( ledamarillo , LOW) ;

        digitalWrite ( ledverde , LOW) ;

        digitalWrite ( buzzer, HIGH) ;

      }

      else

      {

        digitalWrite( ledrojo , LOW) ;

        digitalWrite ( ledamarillo , HIGH) ;

        digitalWrite ( ledverde , LOW) ;

        digitalWrite ( buzzer, LOW) ;

      }

    }

    delay (500) ;                  // Para limitar el número de mediciones

  }

OSCURIDAD

De nuevo conectamos 3 LEDS, uno verde, uno amarillo y uno rojo. Cuando hay suficiente luz ambiente, todos los LEDS estarán apagados, y conforme baja el nivel de luz, se encienden las diferentes “alertas”, siendo la roja la que indica mayor oscuridad.

#define ledrojo 4

  #define ledamarillo 3

  #define ledverde 2

  #define Entrada 0

Un circuito electrónico

Descripción generada automáticamente con confianza baja

  void setup()

  {

    Serial.begin (9600);

    pinMode(ledrojo, OUTPUT);

    pinMode(ledamarillo, OUTPUT);

    pinMode(ledverde, OUTPUT);

    digitalWrite ( ledrojo , LOW) ;

    digitalWrite ( ledamarillo , LOW) ;

    digitalWrite ( ledverde , LOW) ;

  }

  void loop()

  {

    int p = analogRead(Entrada) ;

    int n = map (p, 500,1000, 0, 3) ;

    Serial.println("Valor de A0 " + String(p)) ;

    Serial.println("Numero de LEDs " + String(n)) ;

    if ( n == 0)

    {

      digitalWrite ( ledrojo , LOW) ;

      digitalWrite ( ledamarillo , LOW) ;

      digitalWrite ( ledverde , LOW) ;

    }

    if ( n == 1)

    {

      digitalWrite ( ledrojo , LOW) ;

      digitalWrite ( ledamarillo , LOW) ;

      digitalWrite ( ledverde , HIGH) ;

    }

    if ( n == 2)

    {

      digitalWrite ( ledrojo , LOW) ;

      digitalWrite ( ledamarillo , HIGH) ;

      digitalWrite ( ledverde , LOW) ;

    }

    if ( n == 3)

    {

      digitalWrite ( ledrojo , HIGH) ;

      digitalWrite ( ledamarillo , LOW) ;

      digitalWrite ( ledverde , LOW) ;

    }

    delay (500) ;

  }

TAREA BLOQUE 3

Vamos a simular un avisador de aparcamiento, de forma que un led y un buzzer nos avisen de la distancia al obstáculo. De forma que cuanto más cerca este más rápido debe parpadear y emitir señales acústicas el buzzer. Cuando la distancia sea menor a un umbral se queden fijas las señales (led y buzzer).

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Si ves que es complicado cambiar el tiempo de los pitidos, puedes hacer una segunda opción.

Que comience a pitar y a parpadear el led cuando está más cerca de una distancia y que el pitido sea continuo cuando este muy cerca del obstáculo, las distancias los determináis vosotros.

**Debes subir una fotografía de tu circuito, el programa del ejercicio y un vídeo (puedes hacer la primera solución o la segunda).**

Posibles modificaciones:

Utilizar un RGB para mostrar un color diferente dependiendo de la distancia.

Utilizar distintos tonos de sonido dependiendo de la distancia.

#define trigPin 3

#define echoPin 2

#define led 9

#define buzzer 5

void setup()

{

    Serial.begin (9600);

    pinMode(trigPin, OUTPUT);

    pinMode(echoPin, INPUT);

    pinMode(led, OUTPUT);

    pinMode(buzzer, OUTPUT);

    digitalWrite ( led , LOW) ;

    digitalWrite ( buzzer, LOW) ;

}

void loop()

{

    long duracion, distancia;

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(trigPin, LOW);

    duracion = pulseIn(echoPin, HIGH) ;

    distancia = duracion / 2 / 29.1 ;

    Serial.println(String(distancia) + " cm.") ;

    int cerca = 10 ;

    int lejos = 40 ;

    if ( distancia < cerca)

    {

      digitalWrite ( led , HIGH) ;

      digitalWrite ( buzzer, HIGH) ;

    }

    if ( distancia >= cerca && distancia <= lejos)

    {

        digitalWrite( led, HIGH) ;

        digitalWrite ( buzzer, HIGH) ;

        int espera = map (distancia, cerca, lejos, 1, 500) ;

        delay (espera);

        digitalWrite( led, LOW) ;

        digitalWrite ( buzzer, LOW) ;

    }

    if ( distancia > lejos)

    {

        digitalWrite( led , LOW) ;

        digitalWrite ( buzzer, LOW) ;

    }

    delay (300) ;

}