

Estructura de Computadores: Práctica 5

IDE Arduino: Sistema de seguridad ultrasónico

Pablo Olivares Martínez

En esta práctica trataré de implementar un sistema de seguridad que se activa por ultrasonidos en **Arduino** y el montaje de su circuito. El programa es muy sencillo y consiste en lo siguiente. El sensor de ultrasonido **HC-SR04** va a ir midiendo la distancia del dispositivo de forma periódica. En todo momento irá imprimiendo la distancia por pantalla. Habrán tres umbrales: **100 cm**, que activará el **led verde**; **50 cm**, que activará el **led amarillo**; y **25 cm**, que activarán el **led rojo**, un **mensaje de aviso** y el **zumbador** comenzará a sonar al timbre establecido. Los materiales necesarios son:

- Arduino UNO x1
- Breadboard x1
- Sensor ultrasónico HC-SR04 x1
- Zumbador x1
- LEDs x3
- Resistencia 220 ohm x4
- Cables

He aquí el código:

```
#define trigPin 7
#define echoPin 6
#define LEDlampRed 9
#define LEDlampYellow 10
#define LEDlampGreen 11
#define soundbuzzer 3
int sound = 500;

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDlampRed, OUTPUT);
  pinMode(LEDlampYellow, OUTPUT);
  pinMode(LEDlampGreen, OUTPUT);
  pinMode(soundbuzzer, OUTPUT);
}

void loop() {
  long durationindigit, distanceincm;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  durationindigit = pulseIn(echoPin, HIGH);
  distanceincm = durationindigit / 58.2;

  if (distanceincm < 100) {
    digitalWrite(LEDlampGreen, HIGH);
```

```

    }
    else {
        digitalWrite(LEDlampGreen, LOW);
    }

    if (distanceincm < 50) {
        digitalWrite(LEDlampYellow, HIGH);
    }
    else {
        digitalWrite(LEDlampYellow, LOW);
    }
    if (distanceincm < 25) {
        digitalWrite(LEDlampRed, HIGH);
        sound = 1000;
    }
    else {
        digitalWrite(LEDlampRed, LOW);
    }

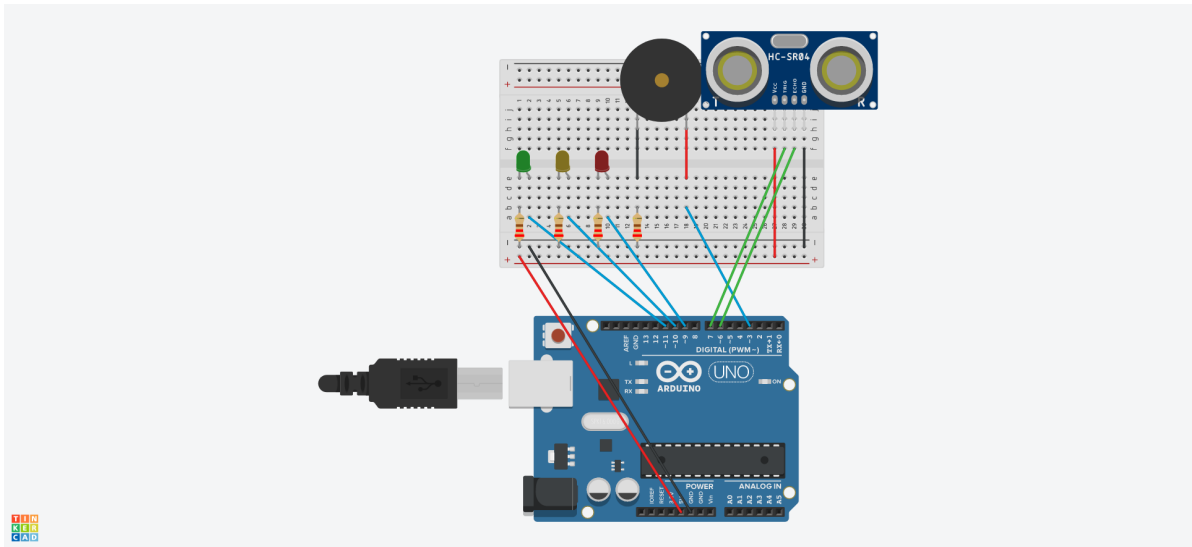
    if (distanceincm > 25 || distanceincm <= 0) {
        Serial.print(distanceincm);
        Serial.println(" cm");
        noTone(soundbuzzer);
    }
    else {
        Serial.println("Cuidado, esta demasiado cerca.");
        Serial.print(distanceincm);
        Serial.println(" cm");
        tone(soundbuzzer, sound);
    }

    delay(300);
}

```

Como se ve, es un código muy sencillo. Definimos los puertos para los leds, el sensor y el zumbador y modificamos su salida según nos convenga. Si van superando los umbrales, la señal de los dispositivos a encender se cambiarán a **HIGH** y los que tengamos que apagar a **LOW** con **digitalWrite()** (en el caso del zumbador se apaga con **noTone()** y se enciende con **tone(pinSalida, timbre)**). Las salidas por terminal las hacemos con **Serial.print()**, y su variante con salto de línea, **Serial.println()**. También usaremos la función **delay()** para marcar los tiempos de cada ciclo del loop. Por último, cabe mencionar que el sensor no mide directamente en cm, por lo que debemos hacerle una conversión dividiendo entre 58.2.

Finalmente, así debería quedar el proyecto montado:



ENLACE AL PROYECTO EN TINKERCAD: <https://www.tinkercad.com/things/gFjLuYiNDqh-neat-elzing-blorr/editel?sharecode=I3L2arbkyWUm-ToUUxRfh9qvv8YvYeYq1W7X3vCUxdk>

NOTA: EL ENLACE SOLAMENTE ESTARÁ DISPONIBLE DURANTE 336 HORAS A PARTIR DEL 23/12/2020 POR COMO ES EL SISTEMA DE COMPARTICIÓN DE TINKERCAD.