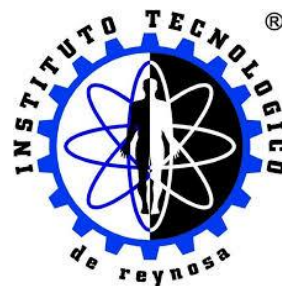




TECNOLÓGICO
NACIONAL DE MÉXICO



Programación básica

Detector de obstáculos y medidor de distancia

Ingeniería mecatrónica

Equipo:

- Elson Jair Montalvo González
- Greisy Margarita Lima Silverio
- Elí Emmanuel Flores Blanco
- Mauricio Gael Buenrostro Rosales
- Héctor Alejandro Pérez López
- María Fernanda Hernández Borjas

12 de mayo de 2025

Medidor de distancia y detector de obstáculos

Autores

- Elson Jair Montalvo González
- Greisy Margarita Lima Silverio
- Elí Emmanuel Flores Blanco
- Mauricio Gael Buenrostro Rosales
- Héctor Alejandro Pérez López
- María Fernanda Hernández Borjas

Resumen

Este proyecto combina dos aplicaciones del sensor ultrasónico HC-SR04: un sistema de detección de obstáculos que activa un buzzer y un LED, y un medidor de distancia que muestra lecturas en una pantalla LCD con interfaz I2C. Se utilizó una placa Arduino Uno y el circuito fue montado en una protoboard. El principal desafío fue la falta de conocimiento sobre algunos componentes, lo cual se resolvió mediante consulta de documentación técnica y pruebas prácticas. El sistema respondió correctamente, detectando obstáculos y mostrando las distancias en tiempo real.

Introducción

En el aprendizaje de la electrónica y la programación, Arduino representa una plataforma ideal debido a su accesibilidad y flexibilidad. Esta práctica permitió integrar dos funciones distintas usando el mismo sensor ultrasónico: detección de obstáculos con alerta visual y sonora, y medición de distancia con salida en pantalla. Arduino permite la creación de prototipos interactivos que facilitan el entendimiento de conceptos clave en sistemas embebidos y automatización.

Objetivo general: Integrar y programar un sistema que combine la detección de obstáculos y la medición de distancia con Arduino, utilizando un solo sensor ultrasónico y una pantalla I2C.

"Arduino es una plataforma abierta basada en hardware y software libre que facilita el aprendizaje de la programación y la electrónica" (Banzi, 2011).

Materiales y Métodos

Materiales utilizados

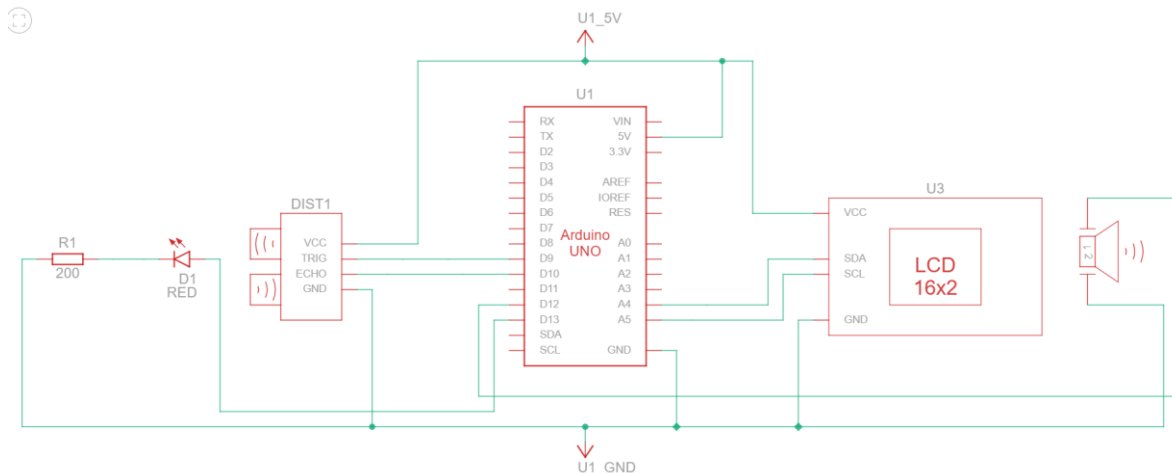
- Arduino Uno
- Sensor ultrasónico HC-SR04

- Pantalla LCD 16x2 con módulo I2C
- Protoboard
- Buzzer activo
- LED (color rojo)
- Resistencias (220 ohm)
- Jumpers (macho-macho y macho-hembra)
- Cable USB para programación

Métodos

Se montó un circuito en protoboard conectando el sensor HC-SR04 a la placa Arduino (pines trig y echo). El buzzer y el LED se conectaron a salidas digitales, mientras que la pantalla I2C se comunicó vía pines A4 (SDA) y A5 (SCL). El código se estructuró en dos partes: la primera detecta objetos a menos de 15 cm y activa el buzzer y el LED; la segunda muestra continuamente la distancia en la pantalla LCD.

Diagrama del circuito



Medición de distancia

```
void loop() {

    long duracion;

    int distancia;

    // pulso

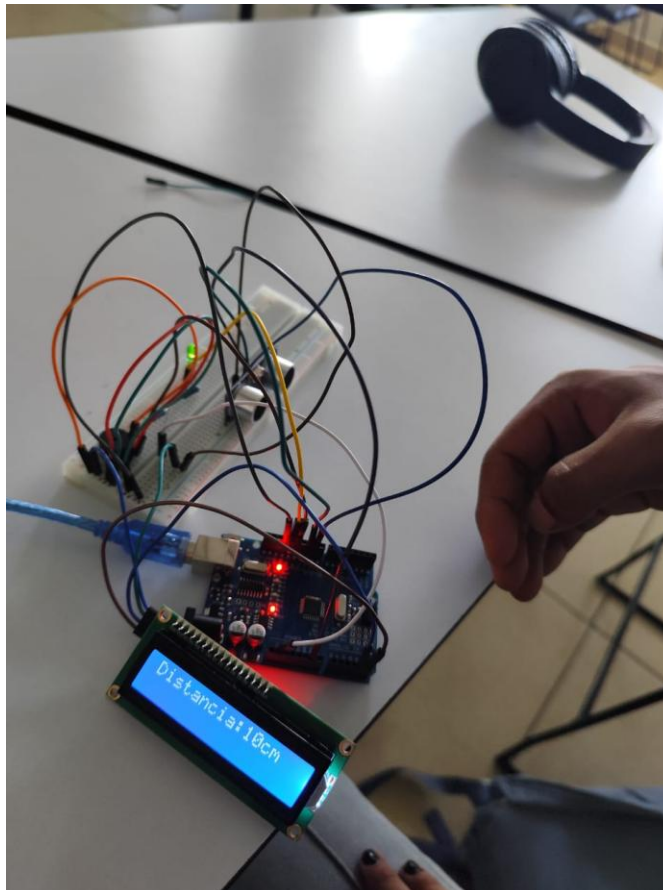
    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);  
  
duracion = pulseIn(echoPin, HIGH);  
  
// Conversión  
distancia = duracion * 0.034 / 2;  
  
// print  
lcd.setCursor(0, 0);  
lcd.print("Distancia: "); // clear  
lcd.setCursor(10, 0);  
lcd.print(distancia);  
lcd.print("cm");
```

Resultados

El sistema logró medir distancias de manera precisa desde 2 cm hasta más de 100 cm, visualizándolas correctamente en la pantalla LCD. Cuando un objeto se acercaba a menos de 15 cm, el buzzer y el LED se activaban, generando una alerta efectiva. Se comprobó el correcto funcionamiento de la lógica condicional implementada en el código.



Discusión

Esta práctica demostró la versatilidad del sensor ultrasónico al ser utilizado simultáneamente para dos funciones. Aunque se presentaron dificultades iniciales al trabajar con la pantalla I2C y el sensor, la consulta de documentación técnica y foros permitió resolver los errores. El uso de protoboard facilitó la experimentación y modificaciones rápidas durante las pruebas.

Conclusiones

- Se logró integrar exitosamente dos funcionalidades con el mismo sensor ultrasónico.
- El buzzer y el LED respondieron adecuadamente a la presencia de obstáculos cercanos.

- La pantalla I2C mostró las lecturas de distancia en tiempo real con buena precisión.
- Esta práctica fortaleció habilidades en el diseño de circuitos, programación estructurada y resolución de problemas técnicos.

Referencias

- Banzi, M. (2011). *Getting Started with Arduino*. O'Reilly Media.

Anexos

Código completo:

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

// SENSOR

const int trigPin = 9;
const int echoPin = 10;

// ledbuzzer

const int ledPin = 13;
const int buzzerPin = 12;

// COSO

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.backlight();

  // pres

  lcd.setCursor(0, 0);
  lcd.print("ENCENDIDO :D");
```

```
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(ledPin, OUTPUT);
pinMode(buzzerPin, OUTPUT);

delay(2000);

lcd.clear();
}

void loop() {
    long duracion;
    int distancia;

    // pulso
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duracion = pulseIn(echoPin, HIGH);

    // Conversión
    distancia = duracion * 0.034 / 2;

    // print
    lcd.setCursor(0, 0);
    lcd.print("Distancia: "); // clear
```

```

lcd.setCursor(10, 0);

lcd.print(distancia);

lcd.print("cm");

// IF
if (distancia < 15) {
    digitalWrite(ledPin, HIGH);
    digitalWrite(buzzerPin, HIGH);
} else {
    digitalWrite(ledPin, LOW);
    digitalWrite(buzzerPin, LOW);
}

delay(500);
}

```

Simulación del circuito en tinkercad

