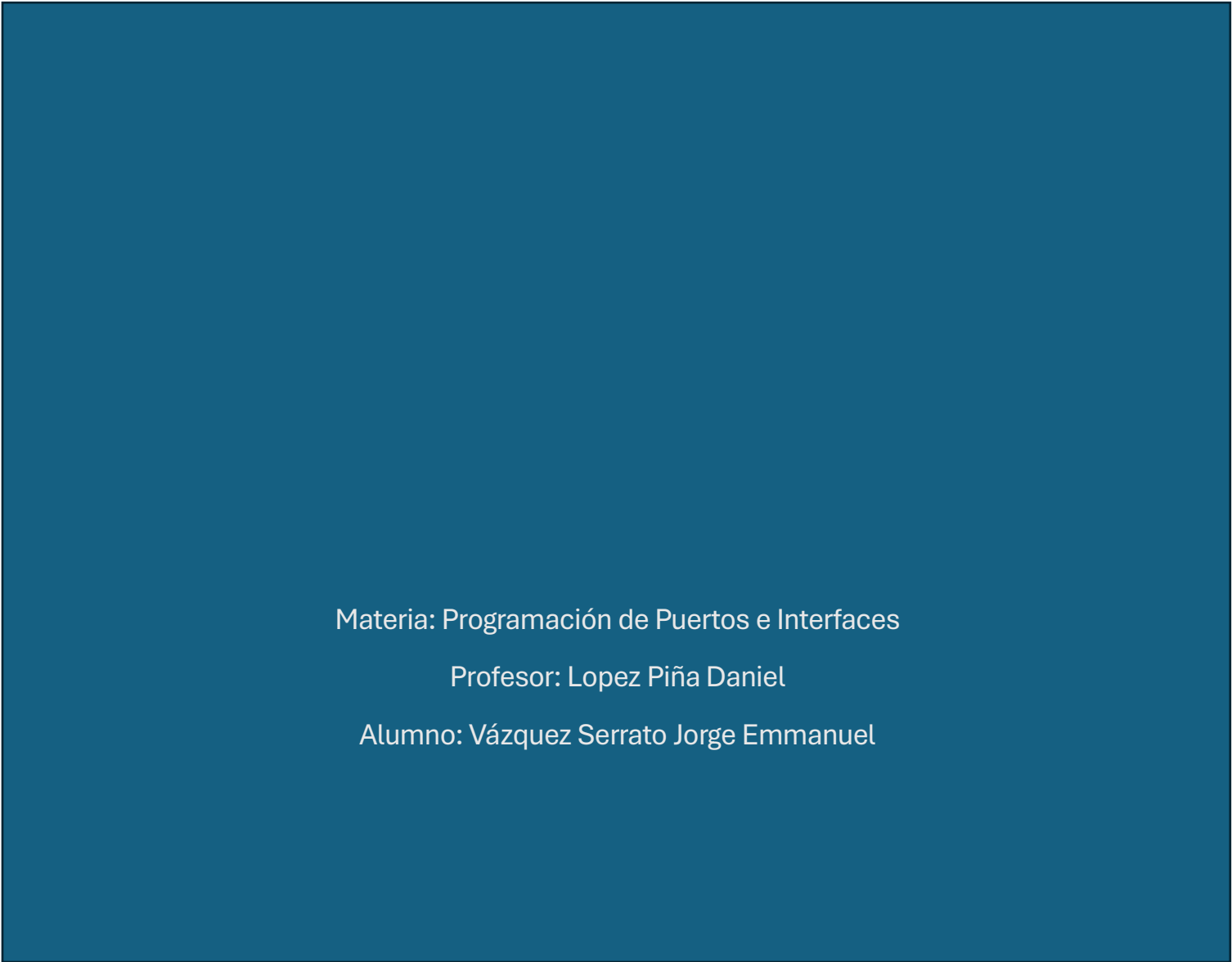




## **Reporte de las practicas del examen de primer parcial**



Materia: Programación de Puertos e Interfaces

Profesor: Lopez Piña Daniel

Alumno: Vázquez Serrato Jorge Emmanuel

## 1. Introducción

La realización de estas prácticas tiene como objetivo poner a prueba los conocimientos prácticos obtenidos durante el primer parcial. Con las pruebas de:

- Conecta un sensor de temperatura LM35 a un Arduino y escribe el código necesario para leer la temperatura en grados Celsius. El sensor está conectado al pin A0. (sensor alternativo TMP36).
- Diseña un sistema con Arduino que utilice un sensor de inclinación (tilt sensor) y un buzzer. El buzzer debe emitir un sonido continuo cuando el sensor detecte una inclinación (es decir, cuando el sensor este en posición horizontal). Incluye el código y el diagrama de conexión.
- Diseña un sistema con Arduino que utilice un sensor de ultrasonido (HC-SR04) para medir distancia y active un LED cuando un objeto este a menos de 10cm. Incluye el código y el diagrama de conexión

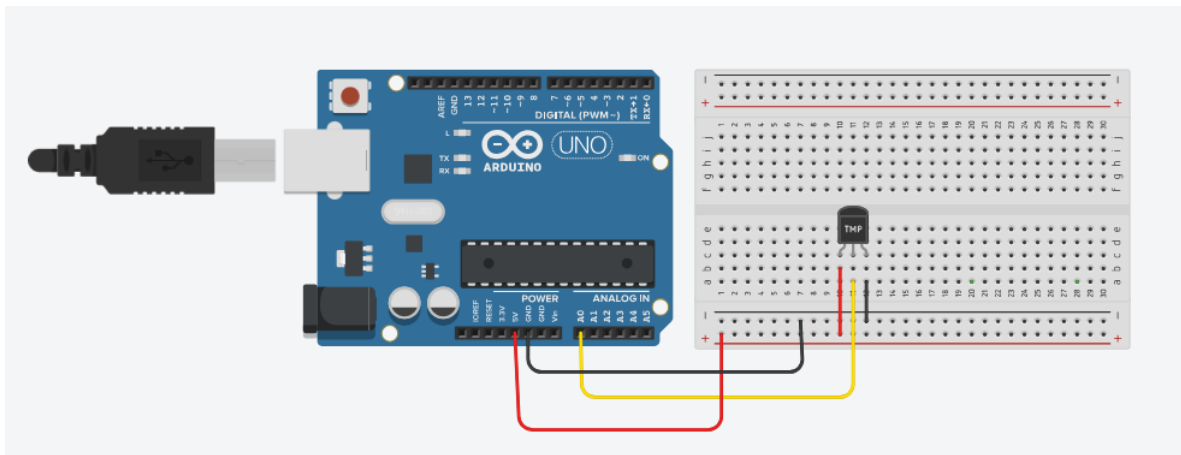
## 2. Desarrollo

### Practica 1: Sensor de temperatura

Para esta práctica se hizo uso de los siguientes componentes:

- Placa Arduino uno
- Protoboard
- Sensor de temperatura TMP36
- Cables

Para la realización de esta practica se comenzó por conectar los pines de 5v y negativo del Arduino a la protoboard y posteriormente un sensor TMP36 a esta última, enseguida se identificó cada una de las terminales de sensor y se conectó las terminales de tierra al negativo del protoboard, potencia al positivo y Vout al pin indicado en las instrucciones (A0).



Tras esto. Pasamos a la parte de programación, donde comenzamos abriendo la comunicación serial entre el Arduino y la pc, después declaramos las variables lectura y temp(temperatura), para posteriormente realizar una lectura del pin A0, el valor obtenido de este no es una medida de temperatura que podamos comprender, para esto es necesario realizar un pequeño calculo para obtener finalmente la temperatura en grados Celsius, la cual almacenamos en la variable temp para final mente imprimirla en el monito serie

Codigo:

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{

  int lectura, temp;

  //Obtengo la medicion del sensor
  lectura = analogRead(A0);

  Serial.print("Lectura: ");

  Serial.print(lectura);

  //Imprimo la temperatura
  Serial.print(", temperatura: ");

  temp = (lectura * (500.0 / 1023.0))-50.0;//Convierto la mediciondel sensor a
temperatura en C°

  Serial.print(temp);

  Serial.println("");

}
```

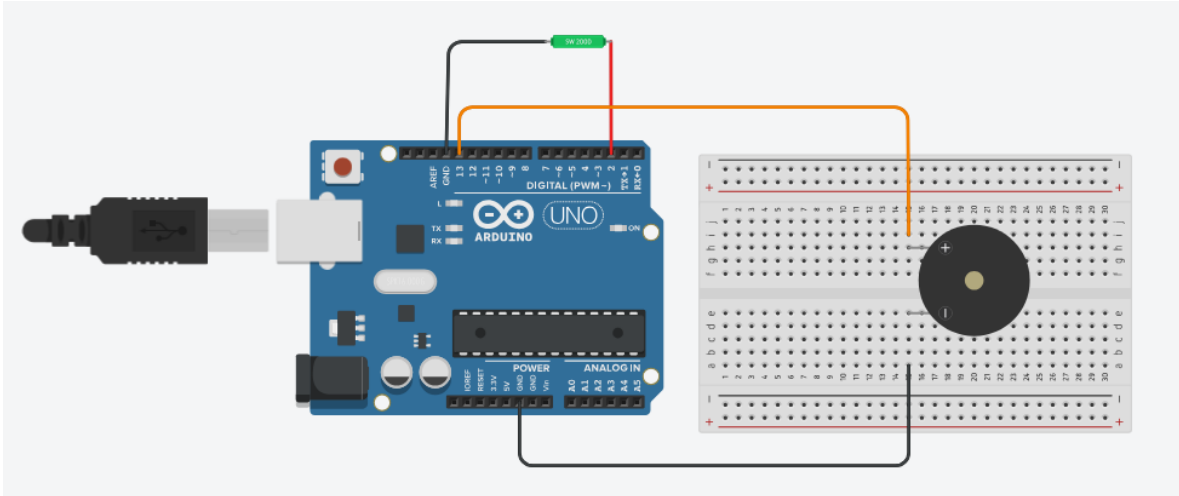
---

## Practica 2: Sensor de inclinación

En esta hicimos uso de los siguientes materiales.

- Placa Arduino
- Protoboard
- Sensor de inclinación (tilt sensor)
- Buzzer
- Cables

Para esta practica comenzamos conectando un terminal del sensor de inclinación al negativo de la placa Arduino y el otro terminal al pin 2, en cuando al buzzer, en este se conectó su terminal positivo al pin 13 y el terminal negativo al pin negativo del Arduino.



Para la parte de la programación, primero inicializar dos variables como “sensorPin” asignándole el pin 2 y “buzzerPin” asignándole el pin 13, después de esto, integramos una compuerta “if” con la que comprobamos si el sensor esta inclinado, y si es así, entonces activamos el buzzer.

Codigo:

```
int sensorPin = 2; // Pin donde está conectado el sensor
int buzzerPin = 13; // Pin donde está conectado el BUZZER
```

```
void setup() {
  pinMode(sensorPin, INPUT_PULLUP); // Activa la resistencia pull-up interna
  pinMode(buzzerPin, OUTPUT);      // Configura el pin del BUZZER como salida
}
```

```
void loop() {
  if (digitalRead(sensorPin) == LOW) {
```

```

    tone(buzzerPin, 1000); // Emite un tono de 1000 Hz

  } else {

    noTone(buzzerPin); // Detiene el buzzer

  }

}

```

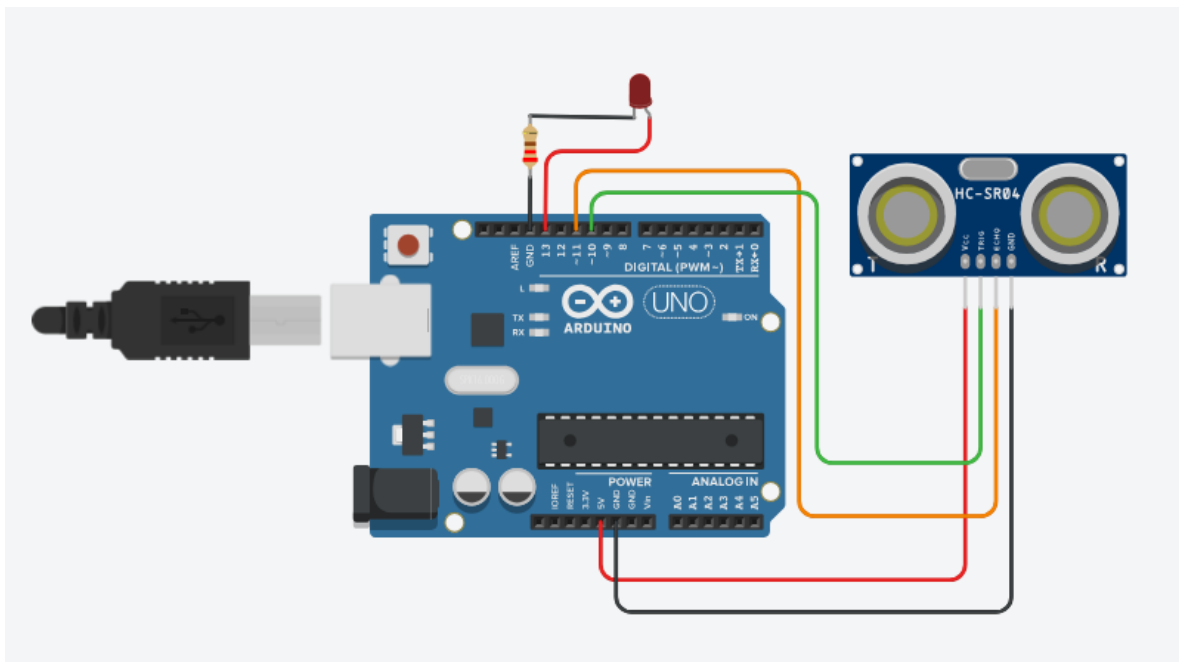
---

### Prueba 3: Sensor de ultrasonido

Con esta prueba hicimos uso de los componentes:

- Arduino
- Sensor de ultrasonido (HC-SR04)
- Led
- Resistencia (220)
- cables

En esta practica comenzamos por conectar el negativo del sensor al GND del arduino y su positivo al pin de 5v, el terminal TRIG fue conectado al pin 10 y el terminal ECHO al pin 11 del Arduino, en cuando al led, conectamos el ánodo al pin 13 y el cátodo a una resistencia de 220 ohm y esta al GND del Arduino.



Por la parte de la programación, primero inicializamos los contantes Trigger al pin 10, Echo al pin 11 y Led al pin 13, luego usamos pinMode para configurar Triger y led como salida y Echo como salida, posteriormente inicializamos las variables “t” y ”d”, para después usar digitalWrite para enviar un pulso y después usamos echo para obtener el ancho de pulso y lo

convertimos el tiempo a distancia, para enseguida imprimir la distancia y usa un “If” para comprobar si la distancia obtenida es menor a 10 y si es así encenderíamos el led.

Codigo:

```
const int Trigger = 10; // Pin digital 10 para el Trigger del sensor
const int Echo = 11;    // Pin digital 11 para el Echo del sensor
const int Led = 13;     // Pin digital 13 para el LED
```

```
void setup() {
  Serial.begin(9600); // Iniciamos la comunicación serie
  pinMode(Trigger, OUTPUT); // Pin Trigger como salida
  pinMode(Echo, INPUT);    // Pin Echo como entrada
  pinMode(Led, OUTPUT);    // Pin LED como salida
  digitalWrite(Trigger, LOW); // Inicializamos el Trigger en LOW
}
```

```
void loop() {
  long t; // Tiempo que demora en llegar el eco
  long d; // Distancia en centímetros

  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10); // Enviamos un pulso de 10us
  digitalWrite(Trigger, LOW);

  t = pulseIn(Echo, HIGH); // Obtenemos el ancho del pulso
  d = t / 59;              // Convertimos el tiempo a distancia en cm

  Serial.print("Distancia: ");
  Serial.print(d);
  Serial.println(" cm");
}
```

```
if (d <= 10) {  
    digitalWrite(Led, HIGH); // Enciende el LED si la distancia es menor o igual a 10 cm  
} else {  
    digitalWrite(Led, LOW); // Apaga el LED si la distancia es mayor a 10 cm  
}  
  
delay(100); // Pequeña pausa de 100ms  
}
```

### 3. Conclusión:

Las evaluaciones prácticas realizadas en este primer parcial permitieron reforzar los conocimientos adquiridos en el uso de sensores con Arduino. Cada una de las prácticas nos demostró la importancia de la correcta conexión de los componentes y la implementación de código eficiente para el correcto funcionamiento de los sistemas. La práctica con el sensor de temperatura permitió aprender cómo convertir señales analógicas en valores significativos de temperatura, mientras que la prueba con el sensor de inclinación mostró cómo detectar cambios de posición y la activación de un buzzer. Finalmente, la práctica con el sensor de ultrasonido evidenció la utilidad de estos dispositivos en la medición de distancia y la activación de elementos de salida.

A través de estas actividades, se fortalecieron habilidades en electrónica y programación, demostrando la versatilidad y aplicabilidad de Arduino en diversos proyectos. Además, se identificaron posibles mejoras en la implementación de los circuitos y el código para optimizar el rendimiento y la precisión de los sistemas desarrollados.