UNIDAD ACADÉMICA MULTIDISCIPLINARIA MANTE

Examen 1° Parcial

6° J

Álvarez Rodríguez José Eduardo

El Mante, Tamaulipas, México. **Febrero 2025.**

Examen 1° Parcial

Ejercicio 13

En esta práctica, se utilizó un sensor de temperatura analógico conectado a un Arduino UNO con el fin de medir la temperatura ambiente y visualizar los valores en el monitor serie del IDE de Arduino.

Para comenzar, el sensor de temperatura fue colocado en la protoboard, asegurando que estuviera correctamente fijado y con suficiente espacio para las conexiones.

Las conexiones se realizaron de la siguiente manera:

- El pin izquierdo del sensor se conectó al pin de 5V del Arduino para suministrar energía.

- El pin central se conectó a la entrada analógica A0 del Arduino, ya que este es el encargado de enviar los datos de temperatura.

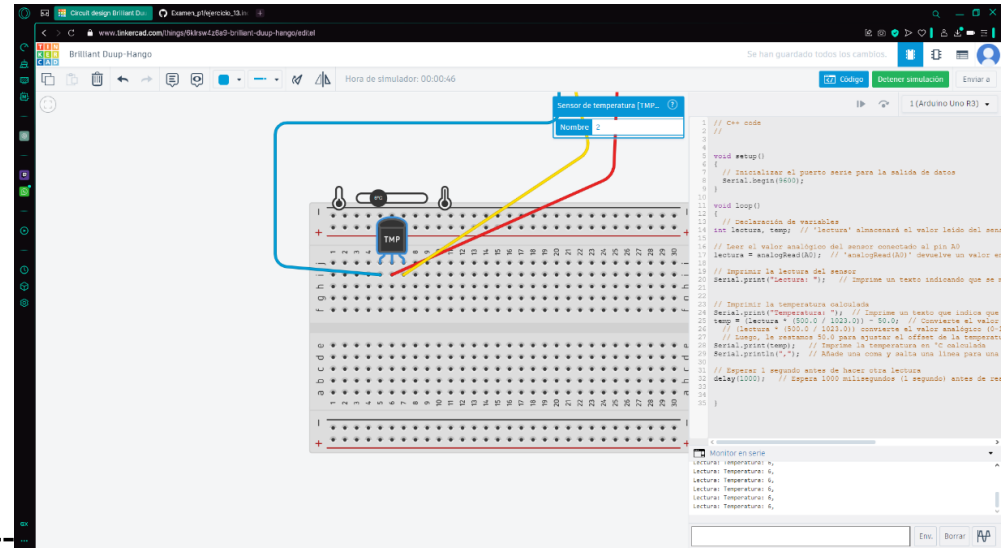
- El pin derecho se conectó a GND para completar el circuito.

Después de realizar las conexiones, se procedió a escribir el código en el IDE de Arduino. Este código se encargó de leer los valores proporcionados por el sensor, transformarlos en grados Celsius y mostrarlos en el monitor serie.

```
void setup()
{
  // Se inicia la comunicación serie para mostrar datos en el monitor
  Serial.begin(9600);
```

```
}

void loop()
{
  // Variables para almacenar la lectura del sensor y la temperatura calculada
  int lectura, temp;

  // Se obtiene el valor analógico del sensor conectado en A0
  lectura = analogRead(A0); // Devuelve un número entre 0 y 1023

  // Se muestra la lectura del sensor en el monitor serie
  Serial.print("Lectura del sensor: ");
  Serial.print(lectura);

  // Se convierte el valor leído en temperatura en grados Celsius
  temp = (lectura * (500.0 / 1023.0)) - 50.0;
  Serial.print("  Temperatura: ");
  Serial.print(temp);
  Serial.println(" °C");  // Se imprime la temperatura con su unidad

  // Se establece un intervalo de 1 segundo entre mediciones
  delay(1000);
}
```

Después de cargar el código en el Arduino, se abrió el monitor serie para visualizar en tiempo real las mediciones de temperatura. Se verificó que el sensor generaba valores numéricos que se transformaban en grados Celsius mediante una fórmula matemática, destacando la importancia de la comunicación serie para la correcta interpretación de los datos.



Ejercicio 14

En esta actividad, se ensambló un circuito utilizando una protoboard y un Arduino UNO R3, con el propósito de controlar un LED y un buzzer en función de un sensor de inclinación.

Las conexiones se realizaron de la siguiente manera:

- Se conectó la línea negativa de la protoboard al GND del Arduino.

- Se instaló un LED en la protoboard y se conectó en serie con una resistencia de 220Ω. El ánodo del LED se conectó al pin 13 del Arduino y el cátodo a GND.

- Se añadió un buzzer y su terminal positivo se conectó al pin 7 del Arduino, mientras que el negativo fue a GND.

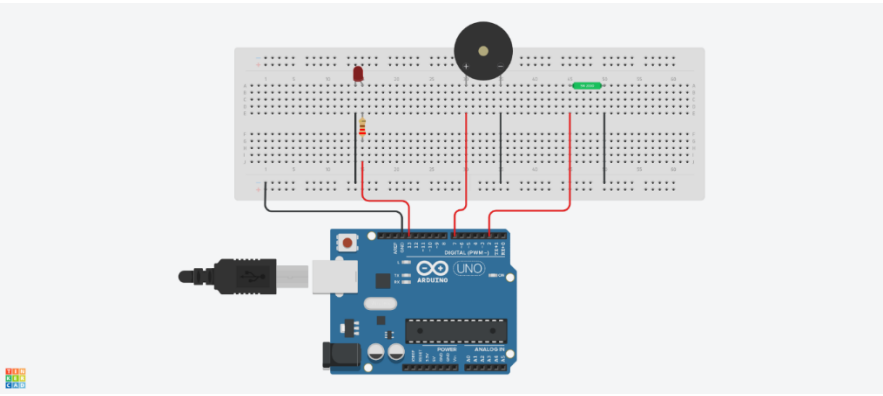
- Se incorporó un sensor de inclinación, cuyo terminal positivo se conectó al pin 2 del Arduino y el terminal negativo a GND.

El siguiente código fue cargado en el Arduino para controlar el comportamiento de los componentes:

```cpp
void setup() {
  // Se configura el pin 2 como entrada con resistencia pull-up interna
  pinMode(2, INPUT_PULLUP);

  // Se establecen los pines 7 y 13 como salidas para controlar LED y buzzer
  pinMode(7, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop () {
  // Se lee el estado del sensor de inclinación
  if (digitalRead(2) == HIGH) {
    // Si el sensor está en su posición normal, el LED se enciende
```

```
    digitalWrite(13, HIGH);
    // El buzzer permanece apagado
    digitalWrite(7, LOW);
  } else {
    // Si el sensor cambia de posición, el LED se apaga
    digitalWrite(13, LOW);
    // El buzzer emite un sonido
    digitalWrite(7, HIGH);
  }
}
```

Después de cargar el código y ensamblar los componentes, se realizó la simulación del circuito. Inicialmente, el LED permaneció encendido, pero al inclinar el sensor, este se apagó y el buzzer comenzó a sonar, indicando correctamente el cambio de estado.



Ejercicio 15

En esta práctica, se implementó un sistema de medición de distancias utilizando un sensor ultrasónico HC-SR04 y un Arduino UNO. El objetivo era encender un LED cuando un objeto estuviera a menos de 10 cm del sensor.

Las conexiones fueron las siguientes:

- El pin VCC del sensor se conectó al pin de 5V del Arduino.
- El pin GND del sensor se conectó al GND del Arduino.
- El pin Trigger se conectó al pin digital 10 del Arduino.
- El pin Echo se conectó al pin digital 11 del Arduino.
- Se añadió un LED con una resistencia de 220Ω en serie, conectando el ánodo al pin digital 13 y el cátodo a GND.

El código programado en el Arduino fue el siguiente:

```
const int Trigger = 10;   // Pin para generar el pulso ultrasónico
const int Echo = 11;      // Pin para recibir el eco del sensor
const int Led = 13;       // Pin para el LED indicador

void setup() {
  Serial.begin(9600);     // Se inicia la comunicación serie
```

```
  pinMode(Trigger, OUTPUT);// Se configura el Trigger como salida
  pinMode(Echo, INPUT);    // Se configura el Echo como entrada
  pinMode(Led, OUTPUT);    // Se configura el LED como salida
  digitalWrite(Trigger, LOW); // Se asegura que el Trigger inicie en
estado bajo
}

void loop() {
  long t; // Variable para almacenar el tiempo del eco
  long d; // Variable para calcular la distancia en centímetros

  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10);   // Se envía un pulso de 10 microsegundos
  digitalWrite(Trigger, LOW);

  t = pulseIn(Echo, HIGH); // Se mide el tiempo que tarda en regresar la
señal
  d = t / 59;              // Se convierte el tiempo en distancia en cm

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

  if (d <= 10) {
```

```
    digitalWrite(Led, HIGH);  // Si la distancia es menor a 10 cm, el LED se enciende
  } else {
    digitalWrite(Led, LOW);   // Si la distancia es mayor a 10 cm, el LED se apaga
  }

  delay(100);  // Pequeña pausa antes de la siguiente medición
}
```

Tras cargar el código en el Arduino y acceder al monitor serie, se probaron distintas distancias colocando objetos frente al sensor. Se verificó que, cuando un objeto se encontraba a menos de 10 cm, el LED se encendía correctamente, validando el funcionamiento del sistema.