UNIDAD ACADÉMICA MULTIDISCIPLINARIA MANTE

REPORTE EXAMEN PRÁCTICO

6° J

PROGRAMACIÓN DE INTERFACES Y PUERTOS

LOPEZ PIÑA DANIEL

HERNANDEZ RAMOS JOSÉ UBALDO

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

Ejercicio 13

Para esta práctica, se trabajó con un sensor de temperatura analógico conectado a un Arduino UNO. La idea principal era poder leer la temperatura del ambiente y mostrarla en el monitor serie del IDE de Arduino.

Se comenzó colocando el sensor de temperatura en la protoboard, asegurándonos de que estuviera bien sujeto y con suficiente espacio para conectar los cables.

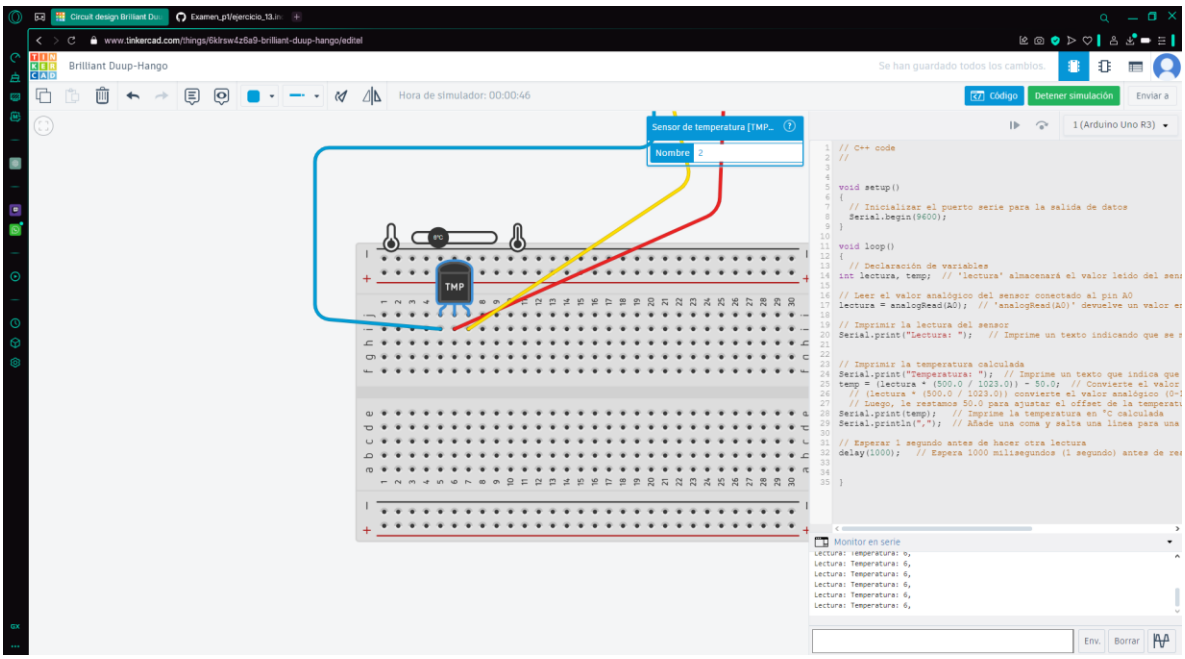
Luego, se realizaron las conexiones necesarias. El pin izquierdo del sensor se conectó a 5V en el Arduino para su alimentación. El pin central del sensor se conectó al puerto analógico A0 del Arduino, ya que este es el que enviará los datos de temperatura. El pin derecho se conectó a GND para completar el circuito.

Con las conexiones listas, se procedió a programar el Arduino. Se escribió un código para leer los valores del sensor, convertirlos a temperatura en grados Celsius y mostrarlos en el monitor serie. El código fue el siguiente:

```
void setup()
{
  // Se inicia la comunicación serie para poder ver los valores en el monitor serie
  Serial.begin(9600);
}

void loop()
{
  // Se declaran variables para almacenar los datos
  int lectura, temp;

  // Se obtiene la lectura del sensor desde el pin A0
  lectura = analogRead(A0); // Devuelve un valor entre 0 y 1023
```

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

  // Se convierte la lectura a 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 espera 1 segundo antes de la siguiente lectura para evitar valores muy seguidos
  delay(1000);
}
```

Finalmente, se cargó el código en la placa Arduino y se abrió el monitor serie del IDE para visualizar los valores de temperatura en tiempo real. Se pudo observar que el sensor analógico devuelve un valor numérico que, mediante una fórmula matemática, se convierte en temperatura en grados Celsius. Además, se comprendió la importancia de la comunicación serie para visualizar los datos y la necesidad de realizar conversiones adecuadas para interpretar correctamente las lecturas del sensor.



Ejercicio 14

Para comenzar con la práctica, primero se utilizó una protoboard como base para conectar los componentes de manera ordenada. Se tomó un Arduino UNO R3 y se conectó el puerto GND a una de las líneas negativas de la protoboard. Esto es importante porque todos los componentes que requieren conexión a tierra podrán usar esta misma línea, evitando cables innecesarios.

Después, se colocó una LED en las posiciones A14 y A15 de la protoboard. Para que la LED funcione correctamente y no se queme, se añadió una resistencia de 220Ω en las coordenadas E15 y G15. Esta resistencia está conectada en serie con la LED, permitiendo que la corriente se limite y la LED no se sobrecargue. Posteriormente, se conectó el extremo positivo de la LED (J15) al pin 13 del Arduino, y su extremo negativo se dirigió a la línea de tierra de la protoboard.

En el siguiente paso, se añadió un buzzer en las coordenadas A30 y A35. El buzzer es un pequeño altavoz que emite sonido cuando se le envía corriente. Se conectó su lado positivo al pin 7 del Arduino, mientras que su lado negativo se dirigió a la línea de tierra de la protoboard.

Luego, se incorporó un sensor de inclinación en las coordenadas A45 y A50. La posición del sensor fue elegida de manera que quedara alineado con la LED y el buzzer, facilitando su conexión y organización dentro del circuito. Este sensor tiene dos pines: uno positivo y otro negativo. El positivo se conectó al pin 2 del Arduino, mientras que el negativo se llevó a la línea de tierra de la protoboard.

El código que se cargó en el Arduino es el siguiente:

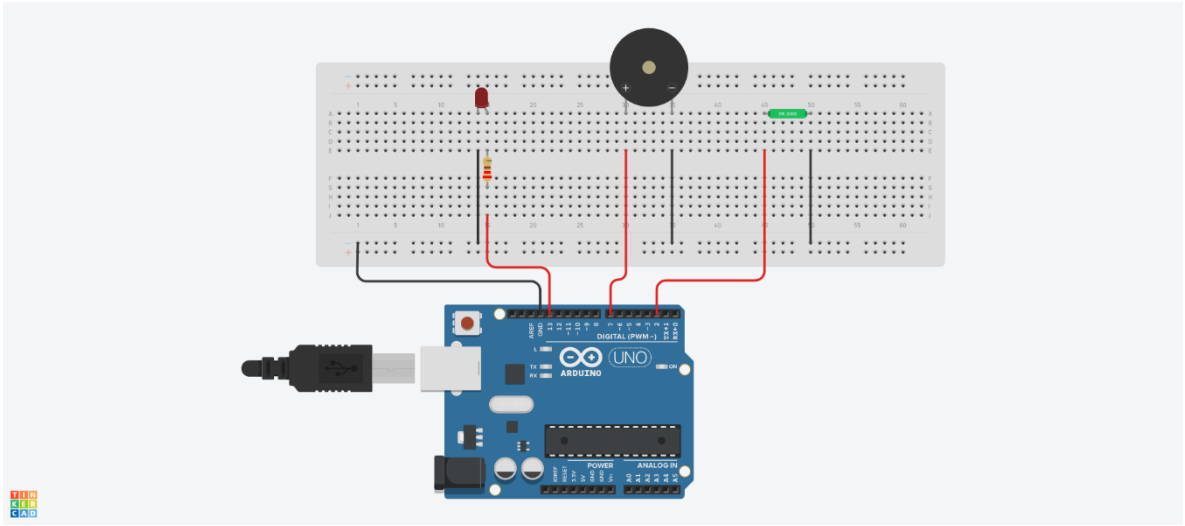
```
void setup() {
  // Configuramos el pin 2 como entrada con resistencia pull-up.
  // Esto significa que normalmente el sensor tendrá un estado HIGH (alto)
  // y solo cambiará a LOW (bajo) cuando se active.
  pinMode(2, INPUT_PULLUP);

  // Configuramos el pin 7 como salida para controlar el buzzer.
  pinMode(7, OUTPUT);
```

```
  // Configuramos el pin 13 como salida para controlar la LED.
  pinMode(13, OUTPUT);
}

void loop() {
  // Se lee el estado del sensor de inclinación.
  if (digitalRead(2) == HIGH) {
    // Si el sensor detecta que está en su posición normal, la LED permanece encendida.
    digitalWrite(13, HIGH);
    // El buzzer permanece apagado.
    digitalWrite(7, LOW);
  } else {
    // Si el sensor detecta inclinación, la LED se apaga.
    digitalWrite(13, LOW);
    // El buzzer comienza a emitir un sonido de alerta.
    digitalWrite(7, HIGH);
  }
}
```

Cuando el circuito comienza a operar, la LED está encendida de manera predeterminada, indicando que el sistema se encuentra en una posición estable.

Sin embargo, si el sensor de inclinación detecta que ha cambiado de posición (es decir, si se inclina demasiado), el sistema responde apagando la LED y encendiendo el buzzer. El sonido del buzzer servirá como una alerta, avisando que el sensor ha sido inclinado.

El buzzer seguirá sonando hasta que el sensor regrese a su posición inicial, momento en el cual la LED volverá a encenderse y el buzzer se apagará.


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Ejercicio 15

En esta práctica, se diseñó un sistema utilizando un sensor de ultrasonido HC-SR04 y un Arduino UNO para medir distancias. Cuando un objeto se encuentra a menos de 10 cm, un LED se enciende como indicador visual. Para lograrlo, se programó el Arduino para realizar la medición de la distancia y activar el LED según la condición establecida.

Para comenzar, se conectó el sensor de ultrasonido a la protoboard y se realizaron las conexiones con el Arduino. El HC-SR04 cuenta con cuatro pines:

VCC: Se conectó al pin de 5V del Arduino para su alimentación.

GND: Se conectó al pin GND del Arduino.

Trigger: Se conectó al pin digital 10 del Arduino.

Echo: Se conectó al pin digital 11 del Arduino.

Luego, se colocó un LED en la protoboard con una resistencia de 220 ohmios en serie para evitar daños. El ánodo del LED se conectó al pin digital 13 del Arduino y el cátodo a GND.

Una vez realizadas las conexiones, se cargó el siguiente código en el Arduino:

```
const int Trigger = 10;   // Definimos el pin 10 como la salida para el Trigger del sensor

const int Echo = 11;      // Definimos el pin 11 como la entrada para el Echo del sensor

const int Led = 13;       // Definimos el pin 13 para el LED indicador


void setup() {

  Serial.begin(9600);      // Inicia la comunicación con el monitor serie

  pinMode(Trigger, OUTPUT);// Configura el Trigger como salida
```

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

void loop() {
  long tiempo; // Variable para almacenar el tiempo del eco
  long distancia; // Variable para almacenar la distancia en cm

  // Generamos un pulso corto para activar la medición
  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trigger, LOW);

  // Medimos el tiempo que tarda en regresar la señal
  tiempo = pulseIn(Echo, HIGH);
  distancia = tiempo / 59;  // Convertimos el tiempo medido a distancia en centímetros

  // Mostramos la distancia en el monitor serie
  Serial.print("Distancia: ");
```

```
  Serial.print(distancia);
  Serial.println(" cm");

  // Si el objeto está a 10 cm o menos, encendemos el LED
  if (distancia <= 10) {
    digitalWrite(Led, HIGH);
  } else {
    digitalWrite(Led, LOW);   // Si está más lejos, apagamos el LED
  }

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

Finalmente, se abrió el Monitor Serie del IDE de Arduino para observar las mediciones en tiempo real. Se colocaron objetos frente al sensor para probar su funcionamiento. Al acercar un objeto a menos de 10 cm, el LED se encendió correctamente.