

Benemérita Universidad Autónoma de Puebla

TALLER STEAM MAKER

MTRO. ALFREDO GARCÍA SUAREZ

INTEGRANTES:

KARYME SUSANA GÓMEZ CHÁVEZ

RICARDO ARTURO MARRUFO POLANCO

TANIA SEVILLA JIMÉNEZ

PRÁCTICA: CONTROL DE TEMPERATURA
CON SENSOR

Índice

Introducción	3
Actividades	4
□ Conexión de circuito	4
□ Código	4
Conclusión.....	6

Introducción

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida: activar un motor, encender un LED, publicar algo en línea. Puede decirle a su placa qué hacer enviando un conjunto de instrucciones al microcontrolador en la placa. Para ello se utiliza el lenguaje de programación Arduino (basado en Wiring), y el Software Arduino (IDE) , basado en Processing .

El software Arduino es fácil de usar para principiantes, pero lo suficientemente flexible para usuarios avanzados. Se ejecuta en Mac, Windows y Linux. Los profesores y estudiantes lo utilizan para construir instrumentos científicos de bajo costo, para probar los principios de la química y la física, o para iniciarse en la programación y la robótica.

Actividades

Se implementa un circuito para monitorear y controlar la temperatura, esta práctica consta de los siguientes materiales:

- ✓ Placa de Arduino UNO
 - ✓ Sensor de temperatura LM35
 - ✓ Pantalla LCD
 - ✓ Resistencias
 - ✓ Cable macho-macho
 - ✓ Ventilador
- Conexión de circuito

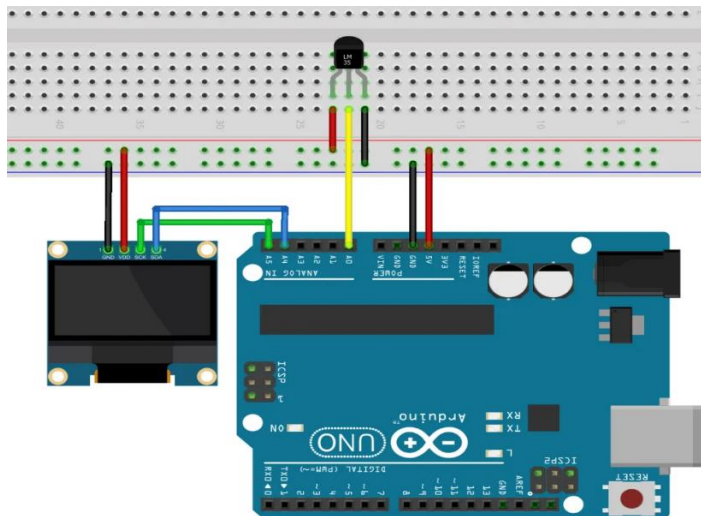


Figura 1. Conexión de circuito.

En la pantalla OLED se encontrarán cuatro pines, que son:

- ✓ GND: pin de tierra.
- ✓ VCC: es el pin de alimentación. Se puede alimentar la pantalla entre 5V.
- ✓ SCL: es el pin de la señal de reloj de la interfaz I2C. En este caso se encuentra conectado al pin A5.
- ✓ SDA: es el pin de la señal de datos de la interfaz I2C. A este le corresponde el pin A4.

El sensor LM35 cuenta con tres pines para su instalación, como lo son:

- ✓ GND: pin de tierra.
- ✓ VCC: es el pin de alimentación. Se puede alimentar la pantalla entre 5V.
- ✓ VSalida: es el pin de la señal de reloj de la interfaz I2C.

- Código

La siguiente figura engloba el código y a su vez, la explicación del funcionamiento en los comentarios.

```

#include <OneWire.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_I2CDevice.h>
Adafruit_SSD1306 display(-1); // <-- Ningun pin reiniciara la pantalla
#define ONE_WIRE_BUS 4 // <-- Se establece el pin 4 como bus OneWire
OneWire oneWire(ONE_WIRE_BUS); // <-- Protocolo de comunicacion por 1 cable
char buffer_temps[128];

const int lm35Pin = A0; // Pin analógico al que está conectado el sensor LM35
const float temperaturaMaxima = 23.0; // Temperatura máxima permitida para el foco

void setup() {
    // No se necesitan declarar los pines analogicos
    //ya que se realiza automáticamente
    Serial.begin(9600);
    delay(100);
    Serial.println("Iniciando pantalla OLED");
    // Iniciar pantalla OLED en la dirección 0x3C
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
#ifdef __DEBUG__
        Serial.println("No se encuentra la pantalla OLED");
#endif
        while (true);
    }
}

void loop() {
    //Se limpia la pantalla después de inicializarse
    display.clearDisplay();

    int rawValue = analogRead(lm35Pin); // Leer el valor analógico del sensor LM35
    float voltage = (rawValue / 1023.0) * 5.0; // Convertir el valor a voltaje (5V de referencia)
    float temperature = (voltage - 0.5) * 100.0; // Calcular la temperatura en grados Celsius

    display.setTextColor(WHITE, BLACK); // <-- Pasa parametro de colores
    display.setTextSize(2); // <-- Tamaño de fuente
    display.setCursor(0, 0); // <--Cordenadas para el cursor

    if (temperature > temperaturaMaxima) {
        display.println("Temp: " + String(temperature)); // <-- Mostrar mensaje en pantalla
        display.println("Estado: Critico");
        display.display(); // <-- Transferencia de lo que se mostrara
        //Se limpiará la pantalla cada vez que cambie la numeración
        display.clearDisplay();
    } else {
        display.println("Temp: " + String(temperature)); // <-- Mostrar mensaje en pantalla
        display.println("Estado: Bueno");
        display.display(); // <-- Transferencia de lo que se mostrara
        //Se limpiará la pantalla cada vez que cambie la numeración
        display.clearDisplay();
    }

    delay(1000); // Esperar 1 segundo antes de la siguiente lectura
}

```

Figura 2. Código de circuito.

Conclusión

En resumen, la práctica de sensor de temperatura y pantalla LCD fue exitosa. Se implementó un sensor de temperatura para medir y capturar datos de temperatura ambiente. Los datos se visualizaron en una pantalla LCD, proporcionando una interfaz clara y legible para el usuario. El sensor y la pantalla se interconectaron correctamente, permitiendo una comunicación fluida y precisa entre ambos componentes. Esta práctica demostró la capacidad de utilizar un sensor de temperatura junto con una pantalla LCD para crear un sistema funcional y práctico de medición y visualización de temperatura.