

Sistema de Monitoreo de Temperatura y Humedad con Indicadores LED

Juan Pablo Restrepo Uribe
Electrónica digital II

Índice

1. Identificación del proyecto	2
2. Resumen	2
3. Descripción del hardware	2
4. Descripción del software	3
5. Estructura del código	3
6. Explicación de funciones principales	3
7. Comunicación (si aplica)	3
8. Interfaz de usuario (si aplica)	4
9. Procedimiento de prueba	4
10. Manejo de errores y seguridad	4

1. Identificación del proyecto

- **Nombre del proyecto:** Sistema de Monitoreo Térmico con ESP32 y MicroPython
- **Integrantes del equipo:** Juan Pablo Restrepo Uribe

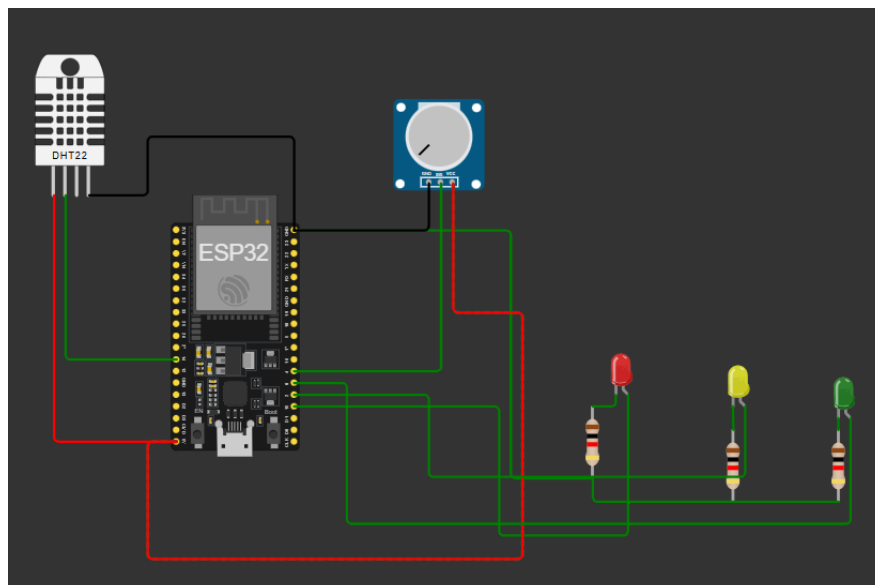
2. Resumen

Este proyecto implementa un sistema de medición de temperatura y humedad mediante un sensor DHT22 y un potenciómetro conectado a un ESP32 programado en MicroPython. Dependiendo del valor de temperatura, se activan LEDs para indicar si esta se encuentra dentro de rangos críticos o seguros.

3. Descripción del hardware

- **ESP32** (modelo con ADC y pines GPIO)
- **Sensor DHT22** conectado al pin 14
- **Potenciómetro** conectado al pin 4
- **LED rojo:** Pin 15
- **LED amarillo:** Pin 2
- **LED verde:** Pin 0
- **Fuente de alimentación:** 5V (USB o regulada)

Diagrama de conexión:



4. Descripción del software

- **Lenguaje usado:** MicroPython
- **Librerías:** `machine`, `time`, `dht`
- **IDE:** Thonny

Flujo general del programa:

1. Inicialización de sensores y LEDs
2. Lectura de potenciómetro y sensor DHT22
3. Conversión del valor analógico a voltaje
4. Evaluación de rangos de temperatura
5. Activación de LEDs según condiciones

5. Estructura del código

- Archivo principal: `main.py`
- No se usan módulos adicionales.
- Código lineal dentro de un ciclo infinito con lectura y decisión en tiempo real.

6. Explicación de funciones principales

- `Pot.read()`: Lee valor del potenciómetro (0 a 4095)
- `d.measure()`: Dispara la lectura del DHT22
- `d.temperature()` / `d.humidity()`: Devuelven temperatura y humedad
- `led.on()` / `led.off()`: Controlan estado de los LEDs
- Lógica condicional: determina qué LEDs se activan según la temperatura

7. Comunicación (si aplica)

Este proyecto no utiliza comunicación inalámbrica ni serial, pero puede extenderse fácilmente con MQTT o HTTP para enviar datos a un servidor.

8. Interfaz de usuario (si aplica)

No se incluye interfaz HTML o gráfica. El sistema muestra los datos por consola y usa LEDs como interfaz física.

9. Procedimiento de prueba

1. Conectar el ESP32 al computador.
2. Cargar el código en el ESP32.
3. Abrir consola para visualizar datos de temperatura, humedad y voltaje.
4. Modificar temperatura (con calor o frío) y observar comportamiento de LEDs.

10. Manejo de errores y seguridad

- Se asume que el sensor DHT22 puede fallar en la lectura; se recomienda agregar `try-except` en implementaciones reales.
- El voltaje del potenciómetro no debe superar el nivel tolerado por el ADC.
- No se usa protección por watchdog o reconexión automática en caso de reinicio.