



**UANL**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



**FIME**

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**

**Laboratorio de Controladores y Microcontroladores  
Programables**

*Termostato Bluetooth – Actividad 8*

**Docente: Ing. Héctor Hugo Flores Moreno**

**Grupo:408**

**Equipo 1**

<b>Matricula</b>	<b>Alumno</b>	<b>Carrera</b>
1499106	Efraín Treviño Garza	ITS
1724347	José Francisco Jasso González	ITS
1817537	Carlos Alejandro Vázquez López	ITS

Fecha de entrega: jueves 30 de octubre de 2025

Ciudad Universitaria, San Nicolás de los Garza N.L.

## Introducción

El termostato Bluetooth es un sistema electrónico diseñado para controlar la temperatura ambiente de forma automatizada y remota. Utiliza un microcontrolador Arduino UNO, sensores, actuadores y un módulo de comunicación inalámbrica para simular el funcionamiento de un sistema térmico inteligente. El presente proyecto se representa mediante un diagrama esquemático elaborado en Tinkercad, donde se muestran los componentes, conexiones y simbología electrónica que permiten interpretar y construir el circuito.

Este tipo de diseño es fundamental en el estudio de controladores programables, ya que permite aplicar principios de electrónica, lógica condicional y comunicación serial en un entorno académico. Además, el sistema está contextualizado para operar dentro de los rangos térmicos típicos de Monterrey, Nuevo León, lo que refuerza su aplicabilidad práctica.

## Desarrollo

El diagrama esquemático del Termostato Bluetooth muestra cómo se interconectan los componentes electrónicos para crear un sistema funcional de control térmico. A continuación se describen los elementos principales:

### 1. Arduino UNO (U1)

Representado como un rectángulo con múltiples pines, el Arduino UNO actúa como el cerebro del sistema. Recibe señales analógicas del sensor TMP36 y del potenciómetro, y señales digitales del pulsador. Procesa esta información mediante lógica condicional y activa los actuadores (motor, LEDs) según el umbral térmico definido. También gestiona la comunicación serial con el módulo Bluetooth.

### 2. Sensor de temperatura TMP36 (U2)

Este sensor analógico proporciona un voltaje proporcional a la temperatura ambiente. Está conectado a un pin analógico del Arduino, que interpreta el valor y lo compara con el umbral definido por el usuario. Su precisión y rango lo hacen adecuado para simular condiciones reales en Monterrey.

### **3. Potenciómetro (RPOT1)**

Simbolizado como una resistencia variable, permite ajustar manualmente el setpoint térmico. El voltaje generado se lee en un pin analógico del Arduino y se utiliza como referencia para activar o desactivar los actuadores. Es útil para pruebas locales sin necesidad de la app móvil.

### **4. Pulsador (S1)**

Este componente digital permite al usuario activar funciones específicas como reinicio, cambio de modo o activación del Bluetooth. Su señal es leída por el Arduino como una entrada digital.

### **5. Módulo Bluetooth HC-05**

Este módulo permite la comunicación inalámbrica entre el sistema y una app móvil. Recibe el umbral de temperatura definido por el usuario y puede enviar retroalimentación del estado del sistema. Aunque no está simulado en Tinkercad, se considera en el diseño físico.

### **6. Módulo Wi-Fi ESP8266 (U3)**

Permite la comunicación inalámbrica con una app móvil o servidor web. Recibe el umbral de temperatura definido por el usuario y puede enviar retroalimentación del sistema. Requiere resistencias divisoras de voltaje para adaptar niveles lógicos.

### **7. Motor de CC (M1)**

Representa el ventilador del sistema. Se activa cuando la temperatura supera el umbral definido, simulando la expulsión de aire caliente. Está conectado a una salida digital del Arduino.

### **8. LED rojo (D2)**

Indica visualmente que el sistema ha activado el modo calefactor. Se enciende cuando la temperatura está por debajo del umbral.

## **9. LED RGB (D1)**

Este componente muestra el estado térmico general mediante colores: azul para frío, verde para templado y rojo para caliente. Está controlado por salidas PWM del Arduino.

## **10. Resistencias (R3, R4, R5, R8, R9)**

Limitan la corriente en los LEDs y el motor, protegiendo los componentes de sobrecargas. Su valor de 220  $\Omega$  es estándar para simulaciones educativas.

## **11. Resistencias 20 k $\Omega$ (R1) y 10 k $\Omega$ (R2)**

Forman un divisor de voltaje para adaptar la señal TX del Arduino al nivel lógico de entrada del ESP8266 (3.3V), evitando daños por sobrevoltaje.

## **12. Fuente de alimentación (5V y GND)**

El sistema se alimenta desde el pin de 5V del Arduino, con retorno a tierra (GND). Esta conexión asegura el funcionamiento estable de todos los componentes.

## **13. Líneas de conexión**

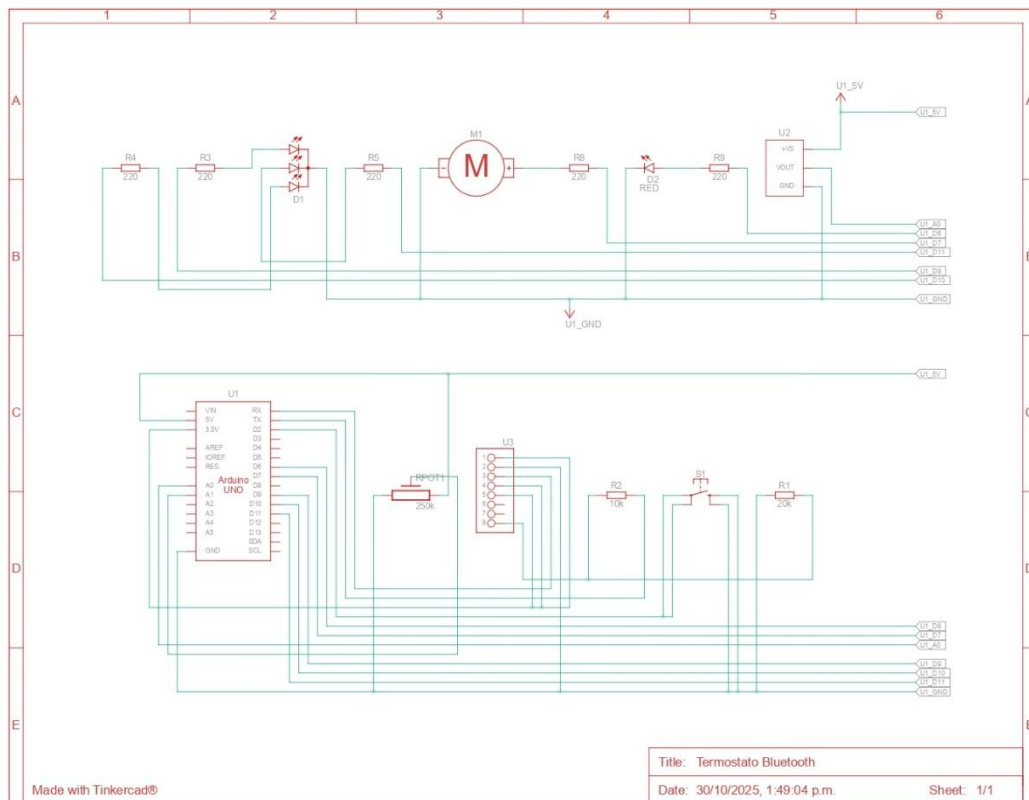
Las líneas del diagrama representan los cables eléctricos. Las intersecciones con puntos indican conexiones reales; sin punto, solo cruce visual. Esta simbología asegura claridad en el diseño.

### **Interacción entre componentes**

- El sistema opera mediante un flujo continuo de información:
- El sensor TMP36 mide la temperatura ambiente.
- El potenciómetro o la app móvil definen el umbral deseado.
- El Arduino compara ambos valores mediante lógica condicional.
- Si la temperatura es mayor al umbral, se activa el motor (ventilador).
- Si es menor, se enciende el LED rojo (calefactor).
- El LED RGB muestra el estado térmico general.
- El pulsador permite control manual adicional.
- El módulo Bluetooth gestiona la comunicación remota.

Este ciclo refleja el principio de control automático, integrando entrada, procesamiento y salida en tiempo real.

## Diagrama Esquemático



## Conclusión

El diagrama esquemático del proyecto “Termostato Bluetooth” representa de forma clara y funcional la arquitectura de un sistema de control térmico automatizado. La correcta interpretación de la simbología electrónica y la integración de componentes como sensores, actuadores y comunicación serial permiten simular un sistema real en un entorno académico.

Este diseño no solo facilita la simulación en plataformas como Tinkercad, sino que también promueve el aprendizaje práctico de electrónica, programación y lógica de control. Además, su adaptación a condiciones climáticas locales refuerza su aplicabilidad y relevancia técnica.