

# How to build a Microcontroller Relay control with Temperature Sensor?

A Practical Tutorial for ESP32-C with BME680 Integration

Rick Rejeleene

February 19, 2026

## Table of contents

<b>1 Objective</b>	<b>2</b>
<b>2 Components required</b>	<b>2</b>
<b>3 Conceptual background</b>	<b>2</b>
3.1 Micro-controller . . . . .	2
3.2 Relay . . . . .	3
3.3 Breadboard . . . . .	4
3.4 Jumper Cables . . . . .	5
3.5 USB-C Cable . . . . .	6
3.6 AdaFruit BME680 . . . . .	6
3.7 Arduino IDE . . . . .	6
<b>4 Wiring the Circuit</b>	<b>7</b>
<b>5 Uploading the Test Code</b>	<b>7</b>
<b>6 Integrating the BME680 Sensor</b>	<b>9</b>
6.1 Wiring Diagram . . . . .	10
6.2 Pin map . . . . .	10
6.3 ASCII schematic . . . . .	10
6.4 Output from BME680 . . . . .	10

In this tutorial, I share how to build a Microcontroller Relay and Temperature Sensor. For Hardware and Software, you will need the pre-requisite components. Most of these can be bought online or local electronics store.

# 1 Objective

- a. To show how to control a relay
- b. How to read temperature using an ESP32 and BME680 sensor

# 2 Components required

1. ESP32-C Dev-Board
2. Breadboard
3. Jumper Cables
4. USB-C Cable
5. Single Relay 5V (Inland)
6. BME680
7. Arduino IDE (Software)
8. Install dependency libraries in IDE

In implementing this project, I am reminded of Professor Poovammal who taught Computer Architecture class for my undergrad-engineering. The class was most engaging and allowed me to understand Computer Architecture. Few years later, Code written by Charles Petzold book allowed me to understand from bottom-up method from zero to one to web-browsers on how computers work.

My recommendation is to always use technical textbook or original patent work for defining and understanding. There are many blogs which has easier ways to understand and describing. It's best to understand from original works. The reason is you will become comfortable in understanding technical domain. It's highly important as an Engineer or Scientist, you'd need to become familiar with newer domains.

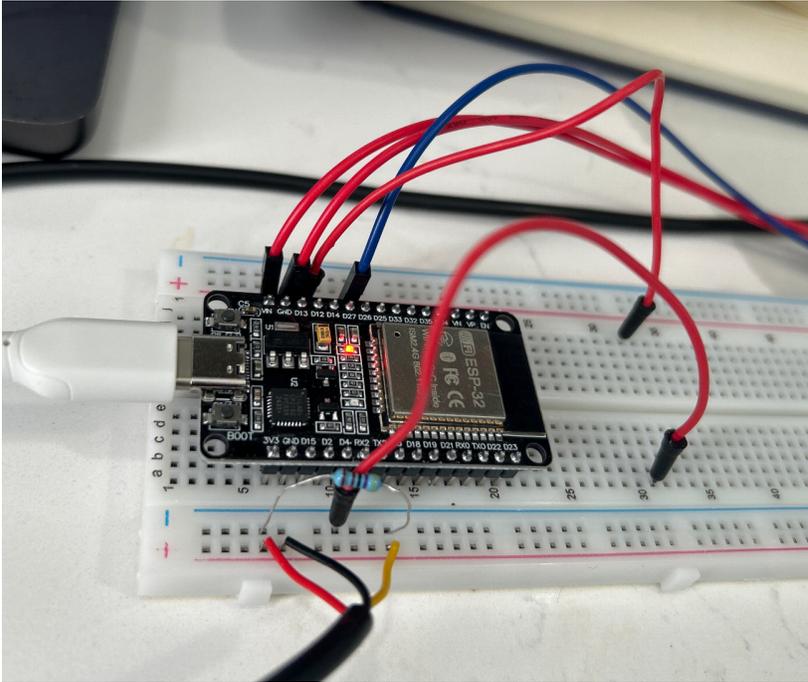
# 3 Conceptual background

## 3.1 Micro-controller

A microcontroller is a complete computer on a single chip.

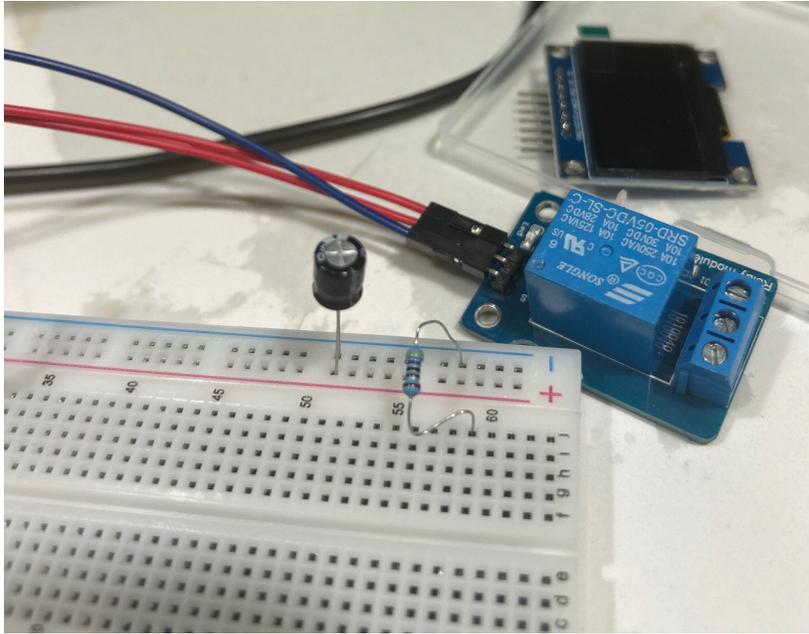
For definition, We refer Paul Scherz and Simon Monk's Practical Electronics for Inventors. In Chapter 13, Micro-controllers, the authors state, *“The microcontroller is essentially a computer on a chip. It contains a processing unit, ROM, RAM, serial communications ports, ADCs, and so on. In essence, a microcontroller is a computer, but without the monitor, keyboard, and mouse. These devices are called micro-controllers”*

For this Tutorial, We use ESP32, dev-kit with Type-C. ESP32 is an accessible micro-controller with Wifi, Bluetooth features.



### 3.2 Relay

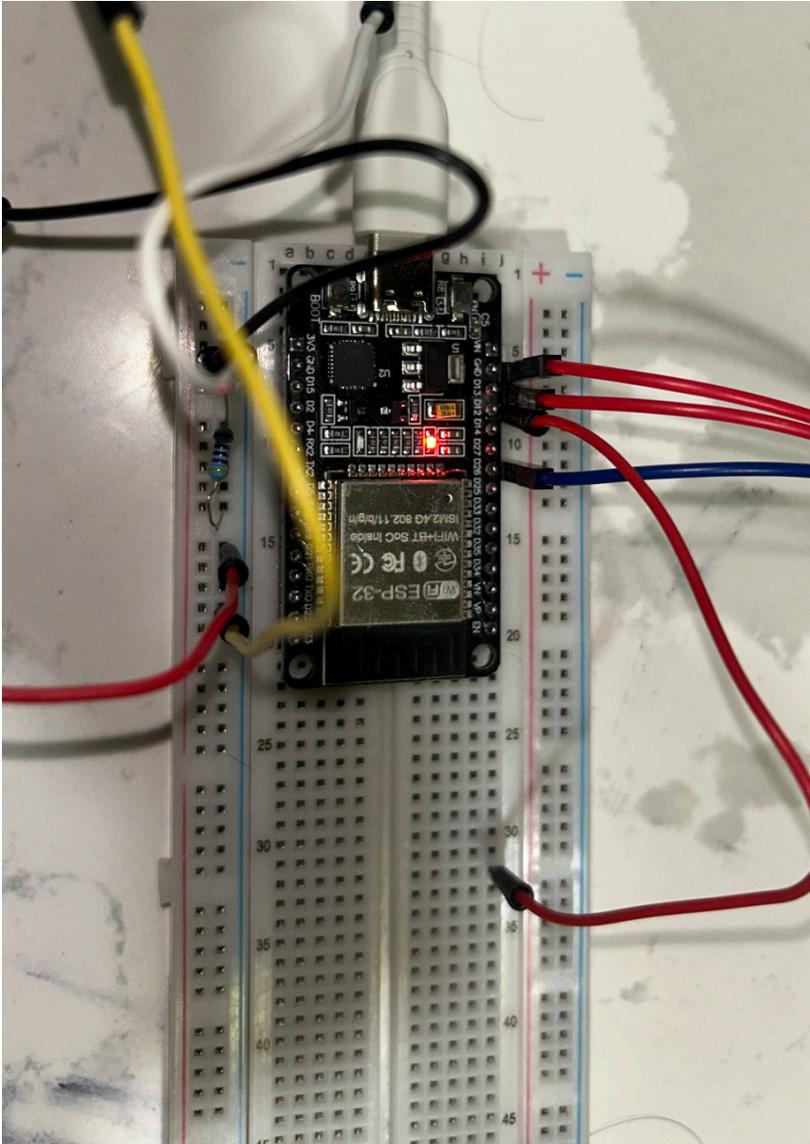
Relays are like switches, which we can use for remote control. We use Inland Single 5V Relay Module for Arduino available from local electronics store as Micro-Center. Our module uses SONGLE 5v high-quality relay. It can also be used to control lighting, electrical and other equipment. The modular design makes it easy to expand with the ESP32. The relay output is by a light-emitting diode.



### 3.3 Breadboard

US Patent D228136 (1971) by Ronald J. Portugal (E&L Instruments) The founding patent of the modern solderless breadboard. Definition: A reusable construction base for prototyping electronic circuits without soldering.

Our breadboard connected on a Type-C with ESP32 along with Dupont wires.



### 3.4 Jumper Cables

Definition (IEEE/Technical): An electrical wire with a connector or pin at each end used to interconnect components on a breadboard or prototype circuit, without soldering. Also called DuPont wires.

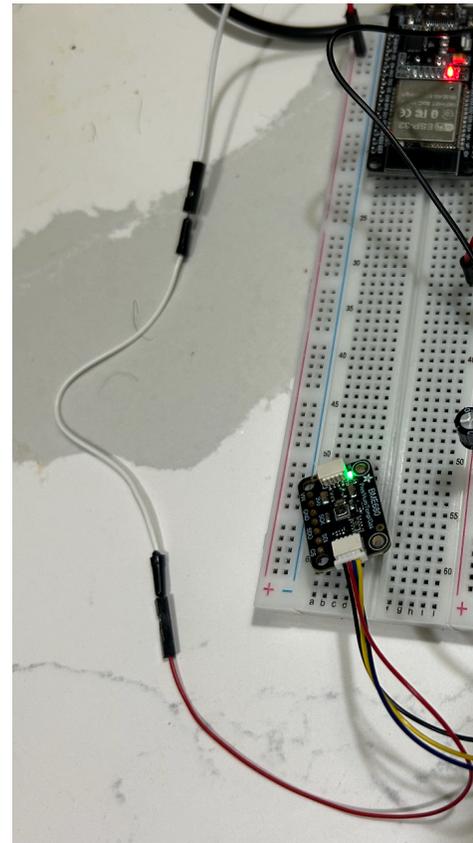
Variants: Male-to-Male, Male-to-Female, Female-to-Female — differentiated by solid (breadboard) or socket (header) tips.

### 3.5 USB-C Cable

Standard: USB-IF Type-C Specification 1.0 (August 2014), adopted as IEC 62680-1-3 in 2016.  
Definition: A 24-pin, fully reversible connector and cable assembly specification covering electromechanical performance, device detection, interface configuration, and USB Power Delivery (USB PD). USB-C is a connector standard, not a protocol.

### 3.6 AdaFruit BME680

Definition: A 4-in-1 MEMS environmental sensor measuring gas (VOCs), humidity, pressure, and temperature. Housed in a  $3.0 \times 3.0 \times 1.0$  mm LGA package.



Adafruit BME680 connected to ESP32 on a breadboard with Dupont wires.

### 3.7 Arduino IDE

Definition: A cross-platform Integrated Development Environment (Windows/macOS/Linux) built on the Processing IDE in Java. Features a code editor with syntax highlighting, one-click

compile/upload, and a software library derived from the Wiring project supporting C and C++. Programs are called sketches (.ino extension).

Source: Arduino Official Documentation

## 4 Wiring the Circuit

1. Setup the Breadboard
2. Fix the ESP32 dev board to communication lane of Breadboard
3. Connect the ESP32 USB-C to your laptop
4. Install Arduino IDE
5. Use Jumper pins to connect the Relay Module with Breadboard

Once you have this setup, this should be enough to test On/Off of Relay.

## 5 Uploading the Test Code

In the IDE setup this code to check if both works:

```
void setup() {
  Serial.begin(115200);
  delay(500);
  Serial.println("Hello ESP32");
}

void loop() {
  Serial.println("tick");
  delay(1000);
}
```

After this step, Execute this code in IDE

```
# define RELAY_PIN 25 // Change if using different GPIO

void setup() {
  Serial.begin(115200);
  while (!Serial);
  Serial.println("Relay Sound Test - Listen for clicks!");

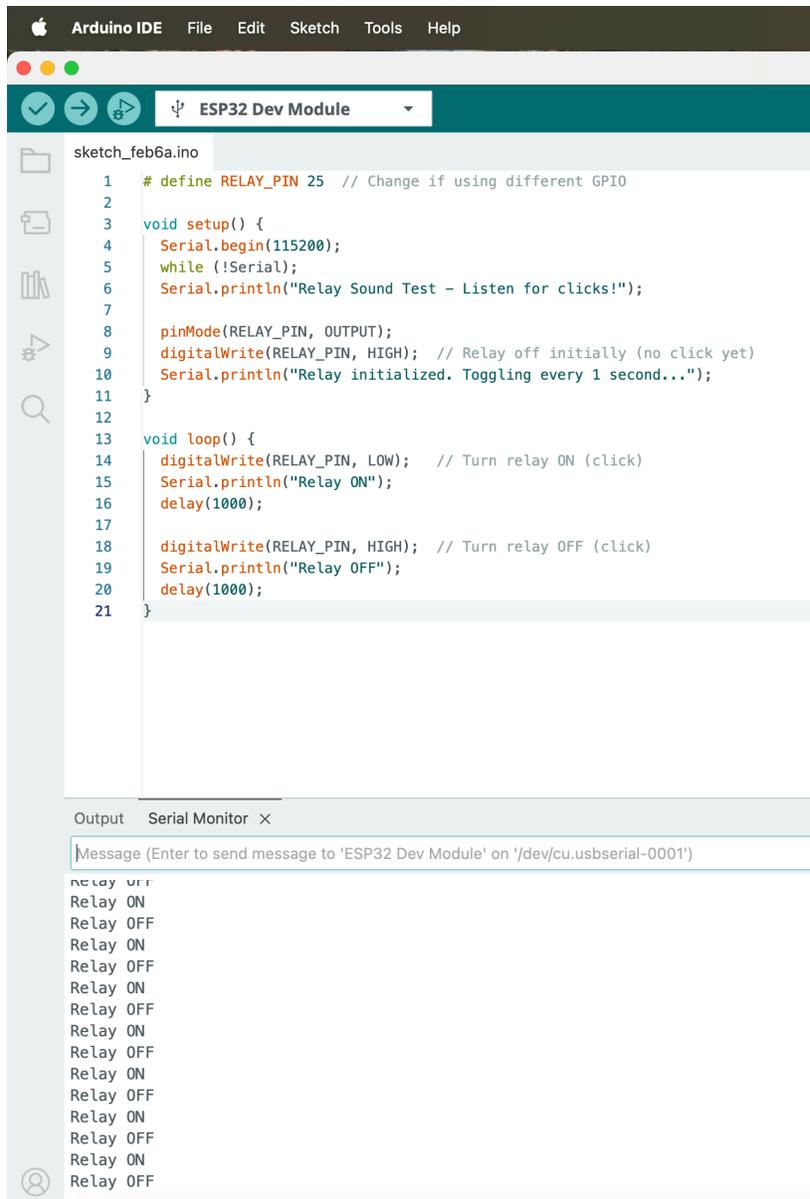
  pinMode(RELAY_PIN, OUTPUT);
}
```

```
digitalWrite(RELAY_PIN, HIGH); // Relay off initially (no click yet)
Serial.println("Relay initialized. Toggling every 1 second...");
}

void loop() {
  digitalWrite(RELAY_PIN, LOW); // Turn relay ON (click)
  Serial.println("Relay ON");
  delay(1000);

  digitalWrite(RELAY_PIN, HIGH); // Turn relay OFF (click)
  Serial.println("Relay OFF");
  delay(1000);
}
```

This will give output On/OFF and show if your relay works.



## 6 Integrating the BME680 Sensor

1. Install Dependent libraries in IDE
2. We use Adafruit\_BME680 and Adafruit\_Sensor.h
3. Connect the BME680 at appropriate pins of ESP32

## 6.1 Wiring Diagram

This sketch uses I<sup>2</sup>C on custom pins

- SDA\_PIN = 33'
- SCL\_PIN = 32'
- I<sup>2</sup>C speed: 100000 (100 kHz)
- BME680 I<sup>2</sup>C address used: 0x77

## 6.2 Pin map

```
-ESP32 pin      -> BME680 pin
-3V3            -> VIN (or VCC)
-GND           -> GND
-GPIO33 (SDA)  -> SDA
-GPIO32 (SCL)  -> SCL
```

## 6.3 ASCII schematic

```
ESP32 (Dev Board)                                BME680 Breakout (I2C)
-----
3V3 -----> VIN / VCC
GND -----> GND
GPIO33 (SDA) -----> SDA
GPIO32 (SCL) -----> SCL
```

## 6.4 Output from BME680

```
BME680 Sensor Readings
Sensor initialized successfully!
===== BME680 Readings =====
  Temperature : 23.45C
  Humidity    : 45.67%
Pressure: 1013.25 hPa
  Altitude    : 150.23 m
  Gas (VOC)   : 25.67 KOhm
=====
```