Task 4: IoT Automation Logic

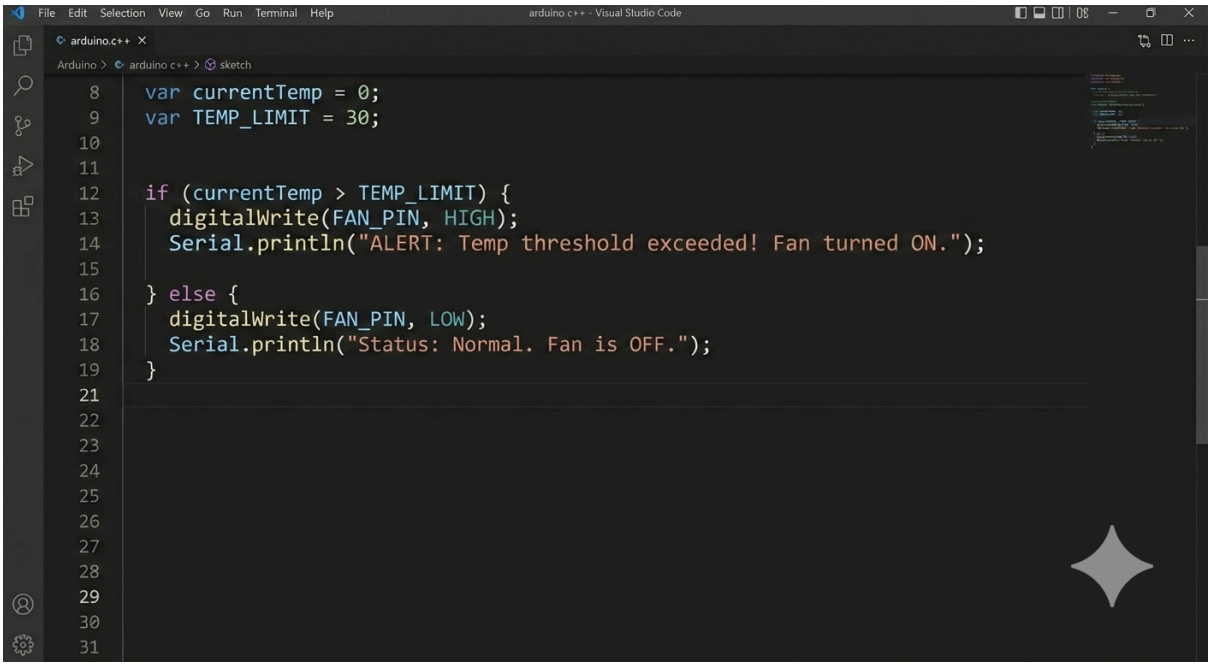
Automation logic is what elevates an IoT system from a simple data-logging tool to a "smart" system capable of independent decision-making. By applying conditional logic at the edge (on microcontrollers like an ESP8266 or ESP32) or in the cloud (using platforms like Blynk or Google Firebase), devices can respond to their environment in real-time.

1. **Defining Conditions (Thresholds)**

The foundation of automation is the conditional statement, usually structured as if-else blocks in code. You define specific thresholds or boundaries for your sensor data.

Simple Conditions: Evaluating a single variable against a static limit (e.g., if (temperature > 30.0)).

Compound Conditions: Evaluating multiple variables simultaneously to create more precise rules (e.g., if (temperature > 30.0 && humidity > 60.0)). This prevents false triggers and ensures actions are only taken when specific environmental profiles are met.

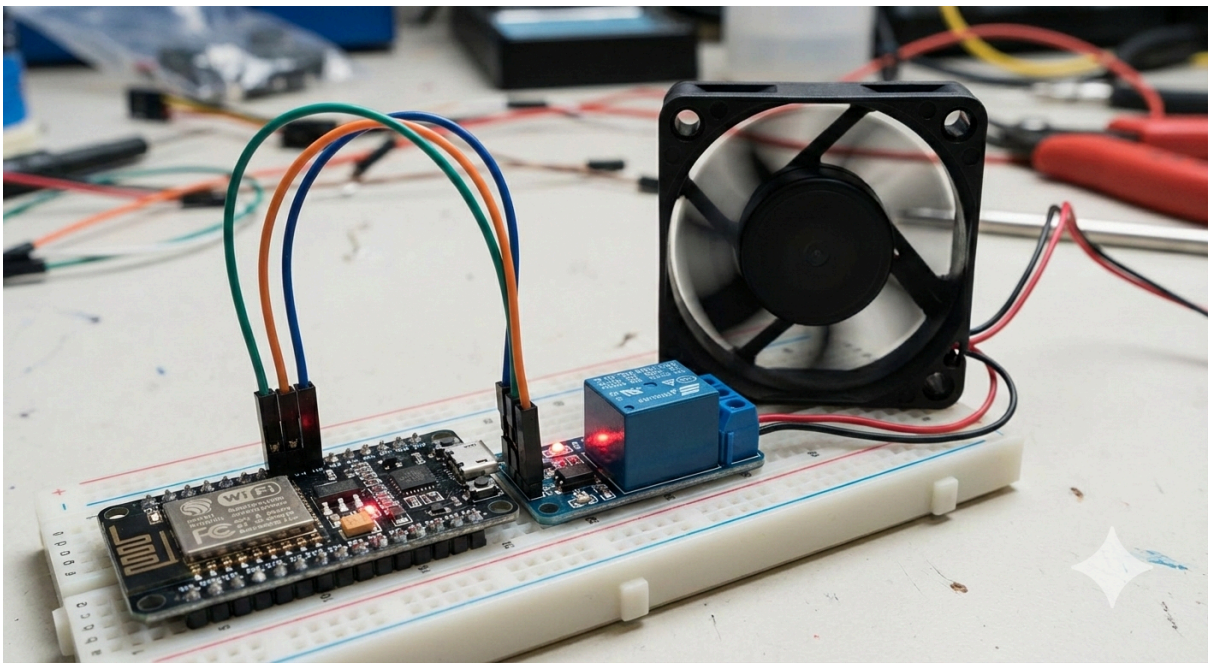


2. **Triggering Alerts or Actions**

Once a condition is met, the system must execute a predefined trigger. These triggers generally fall into two categories:

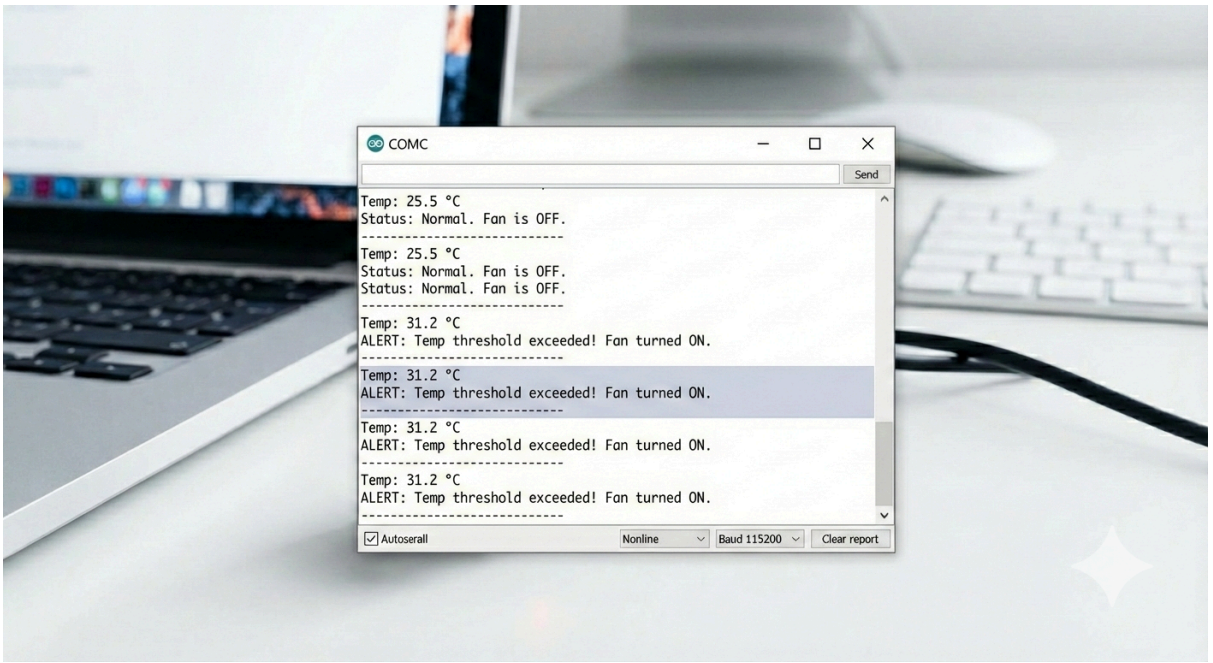
Local Actions (Actuation): The microcontroller sends a high/low signal to a digital pin to control a physical hardware component. For example, if the temperature limit is exceeded, a relay module might be activated to turn on a cooling fan or a water pump.

Cloud Alerts (Notifications): The edge device sends a specific payload flag to the cloud platform, which then triggers a software event, such as sending a push notification to a user's mobile app or firing off an automated warning email.



3. **Displaying Automation Results**

It is crucial to log not just the sensor data, but also the system's responses to that data. When an automation rule fires, the change in state must be reflected locally (via serial output or LEDs) and updated on the digital dashboard (e.g., changing a status indicator from "Normal" to "Warning").



Example implementation (Arduino C++)

Building on the sensor simulation from Task 2, here is how you can implement basic automation logic:

```
// Define our limits
const float TEMP_LIMIT = 30.0;
```

```
const int FAN_PIN = 5; // Example digital pin connected to a relay/fan

void setup() {
  Serial.begin(115200);
  pinMode(FAN_PIN, OUTPUT);
  digitalWrite(FAN_PIN, LOW); // Ensure fan is off initially
}

void loop() {
  // 1. Simulate reading (from previous task)
  float currentTemp = random(200, 351) / 10.0;

  Serial.print("Temp: ");
  Serial.print(currentTemp);
  Serial.println(" °C");

  // 2. Define Conditions & Triggers
  if (currentTemp > TEMP_LIMIT) {
    // Condition met: Trigger Action
    digitalWrite(FAN_PIN, HIGH);

    // 3. Display Results
    Serial.println("ALERT: Temp threshold exceeded! Fan turned ON.");

    // In a real system, you would also push an alert flag to the cloud here
    // e.g., pushToCloud("status", "warning");

  } else {
    // Condition normal: Ensure action is reversed/off
    digitalWrite(FAN_PIN, LOW);
    Serial.println("Status: Normal. Fan is OFF.");
  }

  Serial.println("-------------------------");
```

```
  delay(5000);
}
```