

# INNOVATION:

Creating an ESP32 IoT project to monitor temperature is a popular and practical application. In this project, we'll use the ESP32 microcontroller, a temperature sensor (like the DHT22 or DS18B20), and an IoT platform like Adafruit IO or ThingSpeak to display and log temperature data. Here's a step-by-step guide:

## Components you'll need:

1. ESP32 development board
2. Temperature sensor (DHT22, DS18B20, etc.)
3. Breadboard and jumper wires
4. USB cable for programming

## Software requirements:

1. Arduino IDE (with ESP32 support)
2. Required libraries for your chosen sensor (e.g., Adafruit DHT sensor library or OneWire library for DS18B20)

## Step 1: Set Up Arduino IDE

1. Install the Arduino IDE

## Step 2: Connect the Hardware

1. Connect your ESP32 to your computer using the USB cable.
2. Connect the temperature sensor to DHT22
3. the ESP32. Ensure you have the datasheet or pinout for your specific sensor.
  - For DHT22, connect VCC to 3.3V, GND to GND, and Data to a GPIO pin (e.g., GPIO4).
  - For DS18B20, connect VCC to 3.3V, GND to GND, and Data to a GPIO pin (e.g., GPIO4).

**Step 3: Write the Code** Here's a basic example of ESP32 code to read temperature data from a DHT22 sensor.

### Code:

```
#include "DHTesp.h"
```

```

DHTesp dht;

void setup()
{
  Serial.begin(115200);

  dht.setup(27);
}

void loop()
{

  float temperature = dht.getTemperature();

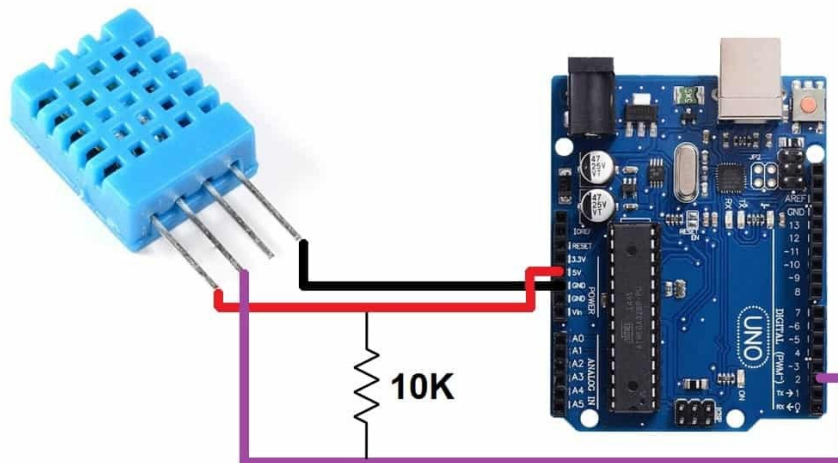
  Serial.print("Temperature: ");
  Serial.println(temperature);

  dely(10000);

}

```

- Select your ESP32 board and COM port in the Arduino IDE.
- Click the "Upload" button to upload the code to your ESP32.



## Step 5: Monitor Temperature Data

- Open the Serial Monitor (Tools > Serial Monitor) to view the temperature readings.
- Once you confirm the sensor is working, you can proceed to send data to an IoT platform.

### **Step 6: Send Data to IoT Platform**

- Sign up for an IoT platform like Adafruit IO or ThingSpeak.
- Modify your code to send temperature data to the platform using the appropriate library or API provided by the platform.
- Follow the platform's documentation for sending data.

With these steps, you'll have a basic ESP32 IoT project to monitor temperature. You can enhance it by adding features like data logging, remote control, or notifications based on temperature thresholds.

### **TEAM MEMBERS:**

1. J.Dharani
  2. S.Nandhini
  3. S.Nivetha
  4. N.Sevventhi
  5. E.Sumithra
-