PRACTICAL 4 : Temperature Detector

Aim : To measure and display temperature using a temperature sensor with Arduino.

Overview :

This project uses a temperature sensor to measure environmental temperature and display it on a serial monitor or other output devices. It introduces sensor interfacing with Arduino and provides a fundamental understanding of analog-to-digital conversion in IoT applications.

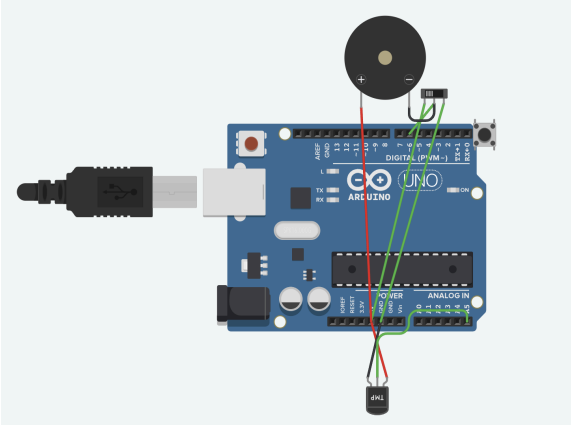
Materials Required :

- Arduino Uno R3
- Temperature Sensor (TMP36)
- Pushbutton
- Slide Switch
- Peizo
- Jumper Wires
- Arduino IDE (Installed on your Computer)

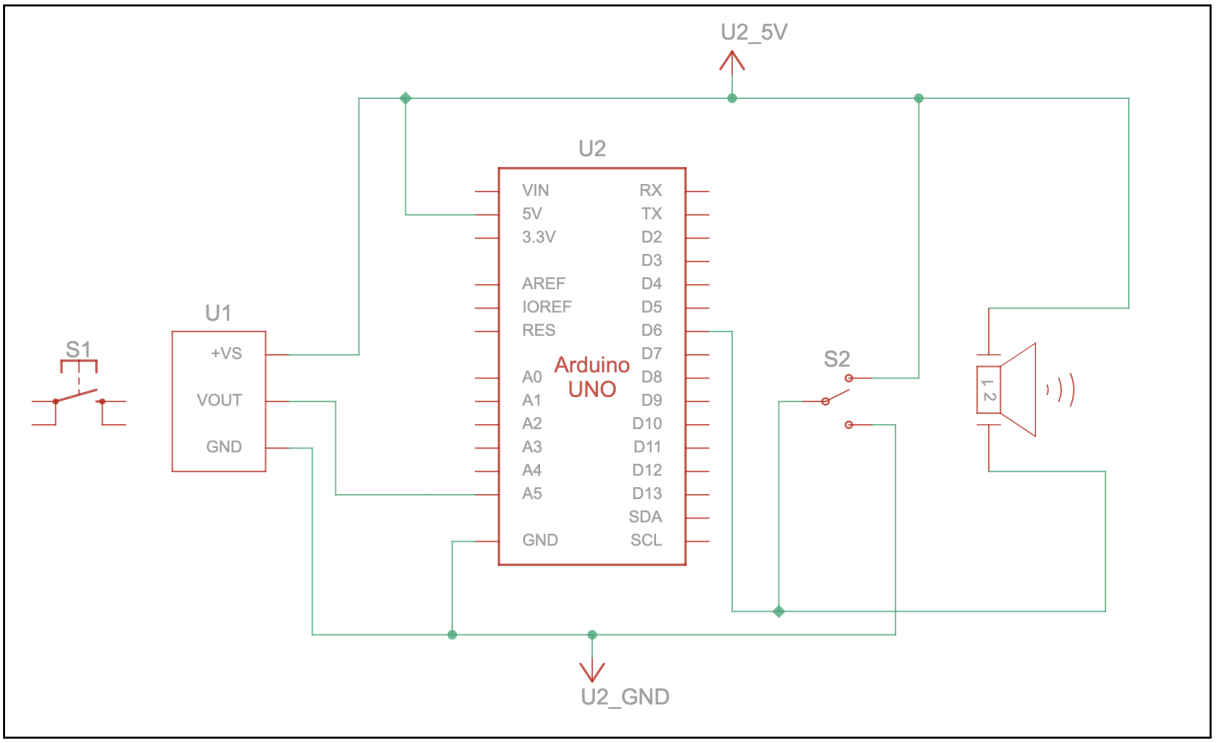
Circuit Connection and Steps :

1. **Connect the Temperature Sensor (TMP36) :**
   - **VCC** (left pin) → **5V**
   - **GND** (right pin) → **GND**
   - **VOUT** (middle pin) → **Analog Pin A5**
2. **Connect the Buzzer and Pushbutton:**
   - Buzzer: Positive (+) to Digital Pin 11, Negative (-) to GND
   - Pushbutton: One terminal to Digital Pin 6, the other to GND
3. **Set Up the Arduino Environment :**
   - Open Arduino IDE, select the correct board and port & upload the code.

Circuit Diagram :



Schematic Diagram :

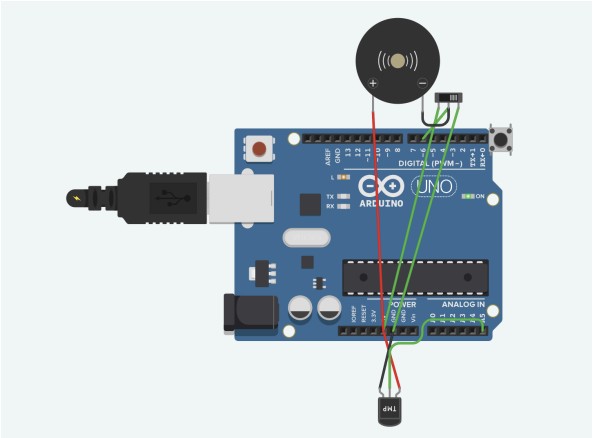


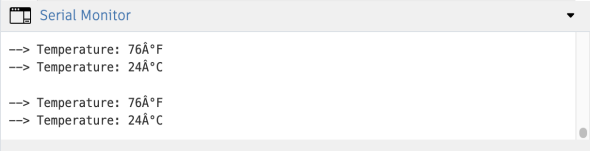
Code :

```
// C++
// Define the pin connections for the components

int tempSensorPin = A5;  // Pin connected to the temperature sensor
int buttonPin = 6;        // Button to activate temperature reading
int ledPin = LED_BUILTIN; // Built-in LED for indication

int TMP = 0;       // Variable for Fahrenheit temperature
int Celcius = 0;   // Variable for Celsius temperature

// Setup function runs once when the program starts
void setup() {
  pinMode(ledPin, OUTPUT);      // Set LED as OUTPUT
  pinMode(buttonPin, INPUT);    // Set button as INPUT
  pinMode(tempSensorPin, INPUT); // Set temperature sensor as INPUT
  Serial.begin(9600);           // Start serial communication
}

// Loop function runs repeatedly
void loop() {
  // Blink LED to show system is running
  digitalWrite(ledPin, HIGH);
  delay(100); // Wait for 100 milliseconds
  digitalWrite(ledPin, LOW);
```

```
  delay(100); // Wait for 100 milliseconds

  // Check if the button is pressed
  if (digitalRead(buttonPin) == LOW) {
    // Read temperature and convert
    TMP = (-40 + 0.878679 * (analogRead(tempSensorPin) - 20)); //
Fahrenheit
    Celcius = (-40 + 0.488155 * (analogRead(tempSensorPin) - 20)); //
Celsius

    // Print temperature readings to Serial Monitor
    Serial.print("⟶ Temperature: ");
    Serial.print(TMP);
    Serial.println("°F");

    Serial.print("⟶ Temperature: ");
    Serial.print(Celcius);
    Serial.println("°C\n");
  }
}
```

Results :





Conclusion :

The Temperature Detector project successfully measures and displays temperature readings using a sensor and Arduino. It introduces the fundamental concept of sensor interfacing and data acquisition. This project serves as a foundation for more advanced IoT applications in environmental monitoring and smart home systems.