EX-1	Design Smart Home Automation System

AIM:

To Create a system to control LED lights and fans remotely.

Components Required:

Arduino, LEDs, push buttons, and a Temperature sensor.

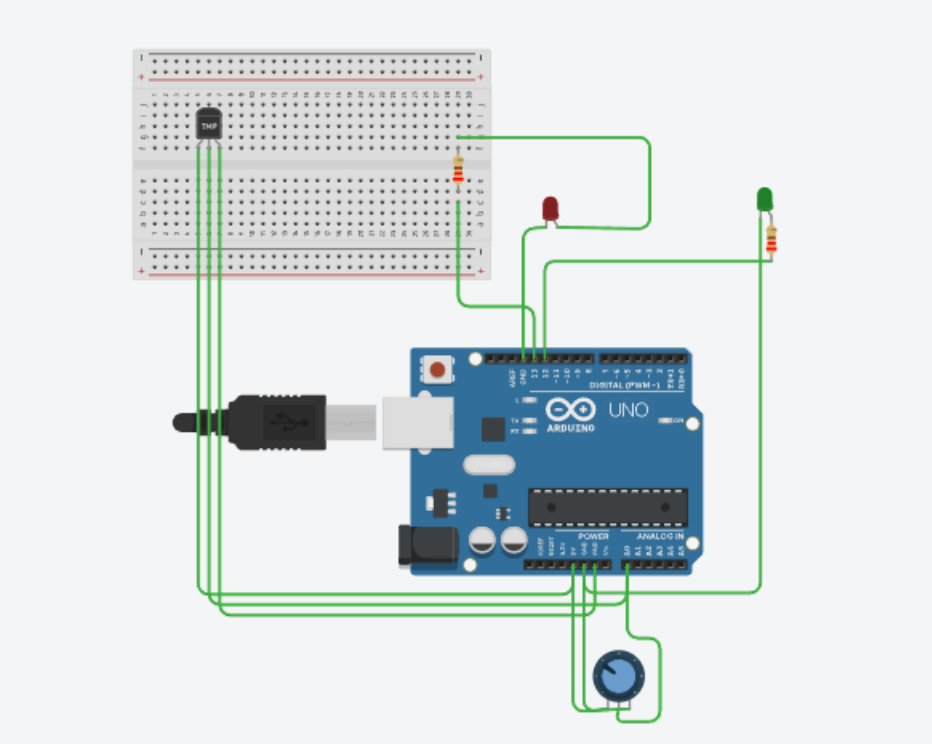
PROCEDURE:

Here's a detailed 10-point procedure to design and simulate the Smart Home Automation System:

Procedure:

- Set Up the Tinkercad Environment
- Add Components to the Workspace
- Connect LEDs and Resistors
- Add Push Buttons and Pull-down Resistors
- Integrate the DHT11 Sensor
- Write the Arduino Sketch
- Include Required Libraries
- Upload and Simulate the Code
- Test the LED Control
- Monitor Temperature and Automate LED Control

Sketch:



Arduino Code:

```
 #define TMP36_PIN A0    // TMP36 sensor connected to analog pin A0

#define LED_PIN1 13     // LED for controlling light 1

#define LED_PIN2 12     // LED for controlling light 2


void setup() {

  pinMode(LED_PIN1, OUTPUT);  // Set LED pins as output

  pinMode(LED_PIN2, OUTPUT);


  // Start serial communication for debugging

  Serial.begin(9600);
```

```
}

void loop() {
  // Read the temperature from TMP36
  int sensorValue = analogRead(TMP36_PIN);

  // Convert the analog reading (0 to 1023) to a voltage (0 to 5V)
  float voltage = sensorValue * (5.0 / 1023.0);

  // Convert the voltage to temperature (TMP36 has 500 mV at 25°C and +20 mV/°C)
  float temperature = (voltage * 1000 - 500) / 20;

  // Print the temperature to Serial Monitor for debugging
  Serial.print("Temp: ");
  Serial.print(temperature);
  Serial.println("°C");

  // If temperature exceeds 10°C, blink LEDs
  if (temperature > 30) {
    digitalWrite(LED_PIN1, HIGH);  // Turn on LED1
    digitalWrite(LED_PIN2, HIGH);
    Serial.println("ON");
                    // Turn on LED2
```

```
  } else {
    digitalWrite(LED_PIN1, LOW);   // Turn off LED1
    digitalWrite(LED_PIN2, LOW);
    Serial.println("OFF");// Turn off LED2
  }

  delay(2000);  // Delay between readings
}
```

Result:

Created a system to control LED lights and fans remotely.