

Alarm System

Assignment-1

By

P.Sneka

952319106033

Alarm System

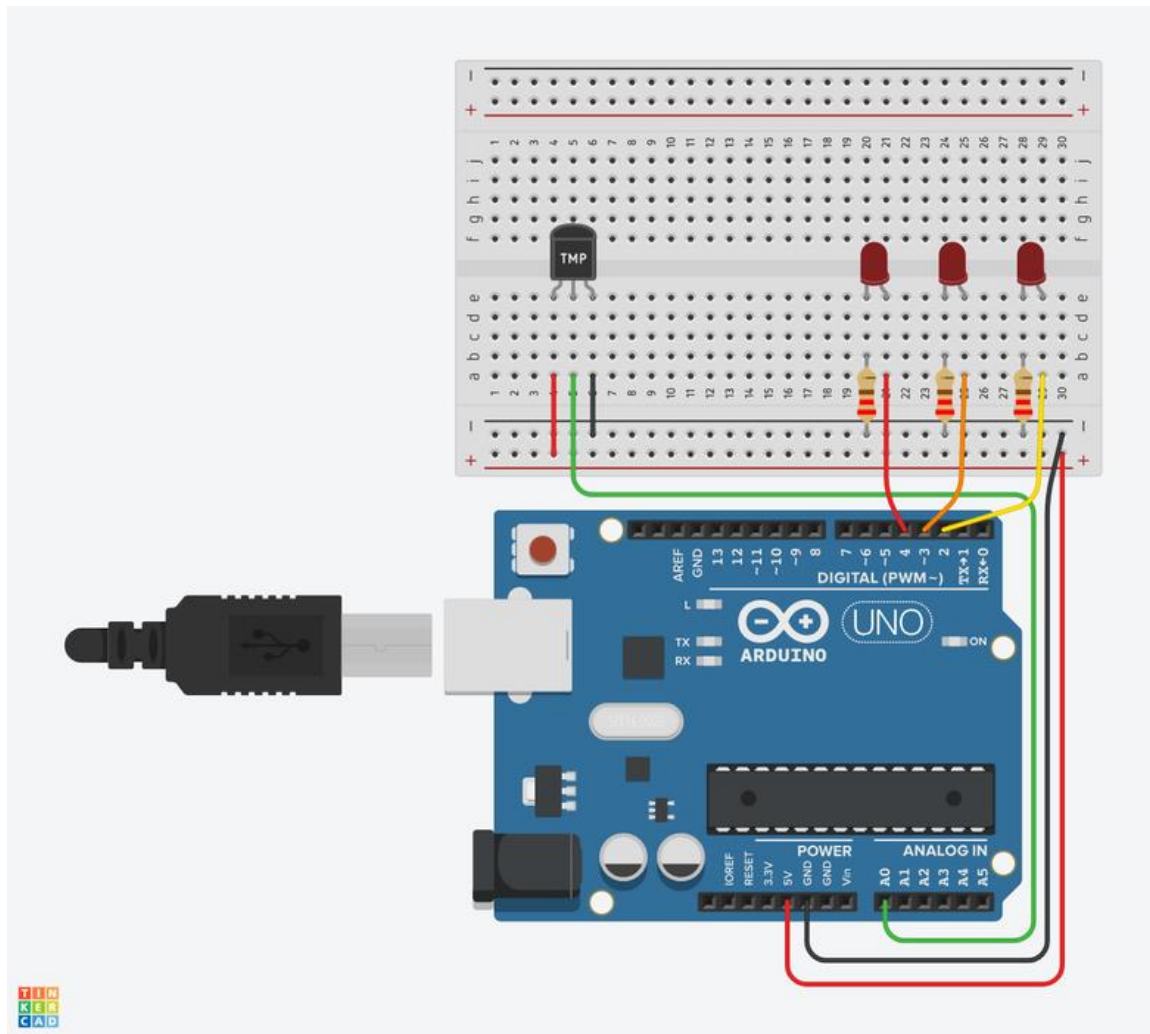
IOT or Internet Of Things is an upcoming technology that allows us to control hardware devices through the internet

In this project, you will turn the Arduino into a thermometer! Use a temperature sensor to measure your skin temperature, and register the output with three LEDs. Even though the Arduino is a digital tool, it can interpret signals from an analog input, like the TMP36 temperature sensor, using the built in Analog-to-Digital (ADC) converter, accessed through the analog pins A0-A5, which you may have learned about in a previous lesson about analog input.



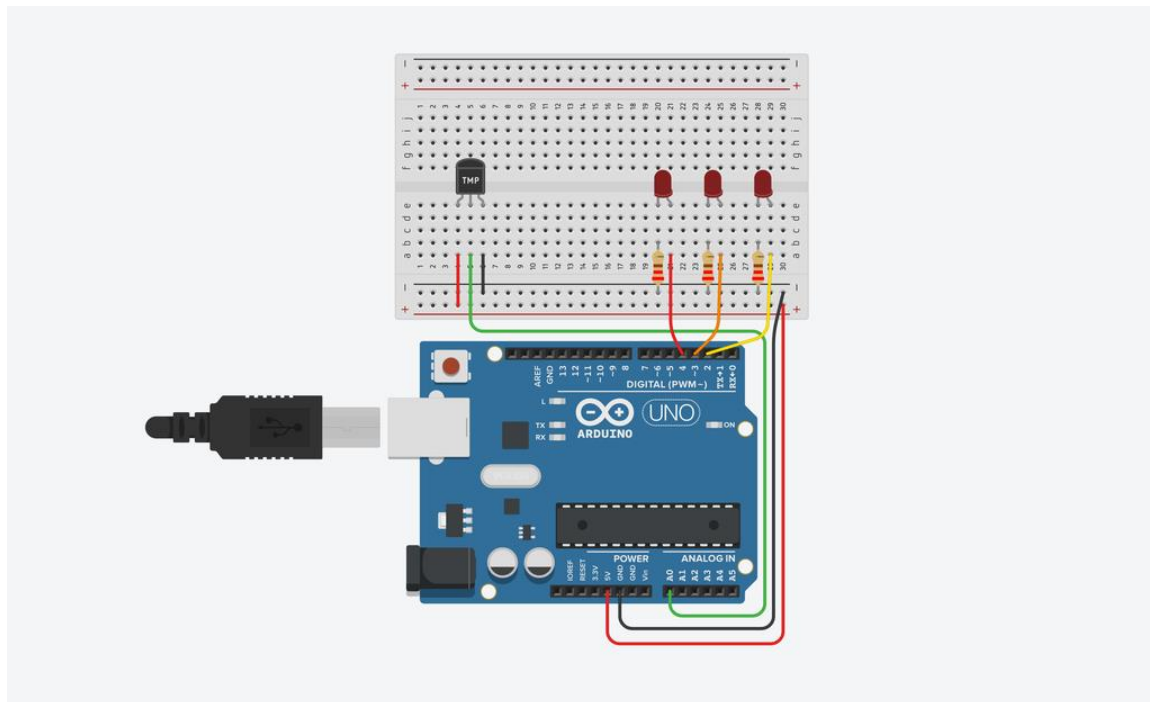
In this lesson, you'll build this simulated circuit yourself along side the sample. To optionally build the physical circuit, gather up your Arduino Uno board, USB cable, solderless breadboard, three LEDs, three alike resistors (any value from 100-1K, 220

ohms preferred), a TMP36 temperature sensor, and breadboard wires.



Just as youve learned from the introductory lessons, start by wiring up your Arduino and breadboard with power and ground next to the example circuit, then

add the the three red LEDs to the breadboard, as shown. These will be the indicator or "bar graph" lights for the project.



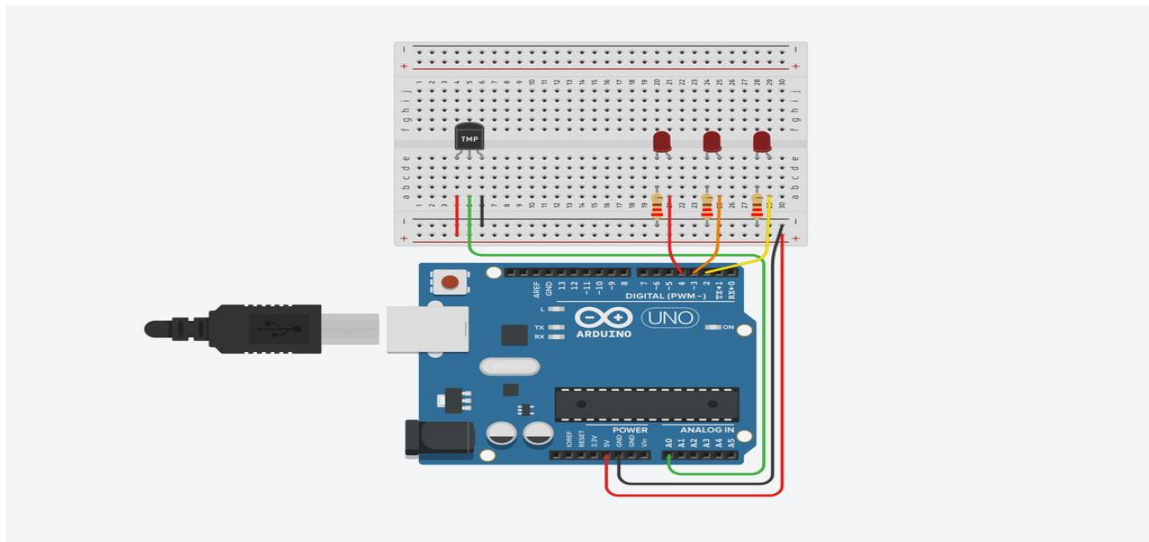
A temperature sensor creates a changing voltage signal depending on the temperature it senses. It has three pins: one that connects to ground, another that connects to 5 volts, and a third that outputs

a variable voltage to your Arduino, similar to the analog signal from a potentiometer

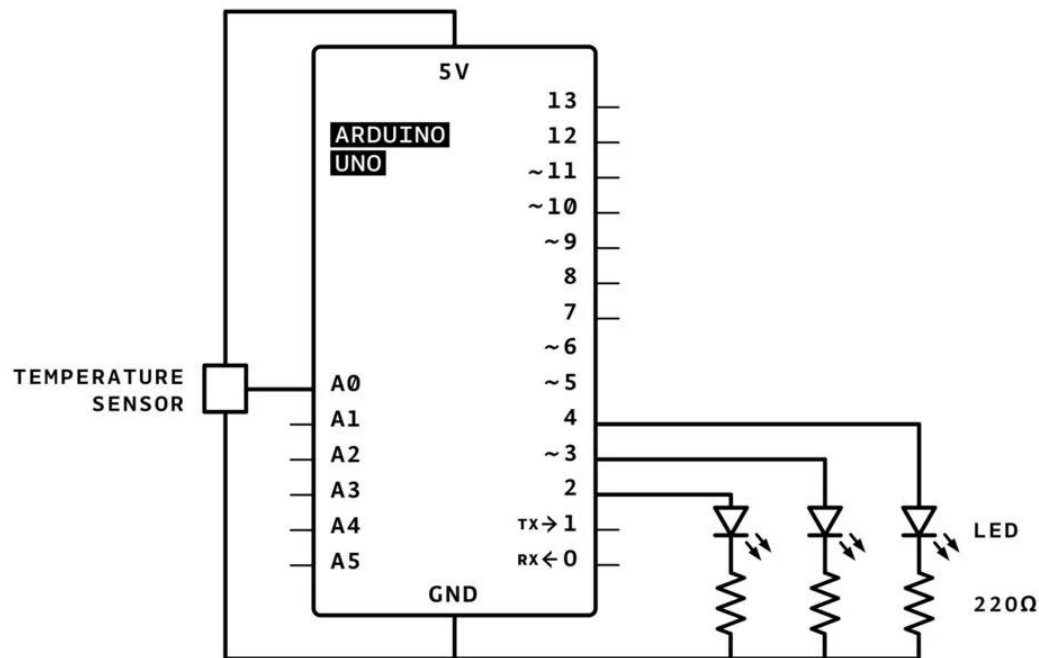
There are several different models of temperature sensor. This model, the TMP36, is convenient because its output voltage is directly proportional to temperature in degrees of celsius.

Place the temperature sensor (TMP36) on the breadboard with the rounded part facing away from the Arduino, as shown in the figure (this is the default orientation).

Step 2: Add Temperature Sensor



Analog Input Observation



In the circuit schematic, you can see that the temperature sensor is connected to power (5 volts) and ground (0 volts) and the analog pin A0. As temperature rises, the pin connected to A0 increases its voltage. You can also see that three LEDs are each connected to their own digital pin.

In the circuit schematic, you can see that the temperature sensor is connected to power (5 volts) and ground (0 volts) and the analog pin A0. As temperature rises, the pin connected to A0 increases its voltage. You can also see that three LEDs are each connected to their own digital pin. In the circuit schematic, you can see that the temperature sensor is connected to power (5 volts) and ground (0 volts) and the analog pin A0. As temperature rises, the pin connected to A0 increases its voltage. You can also see that three LEDs are each connected to their own digital pin.

```
int baselineTemp = 0;
```

```
int celsius = 0;
```

```
int fahrenheit = 0;

void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
}void loop()
{
    // set threshold temperature to activate
    LEDs

    baselineTemp = 40;

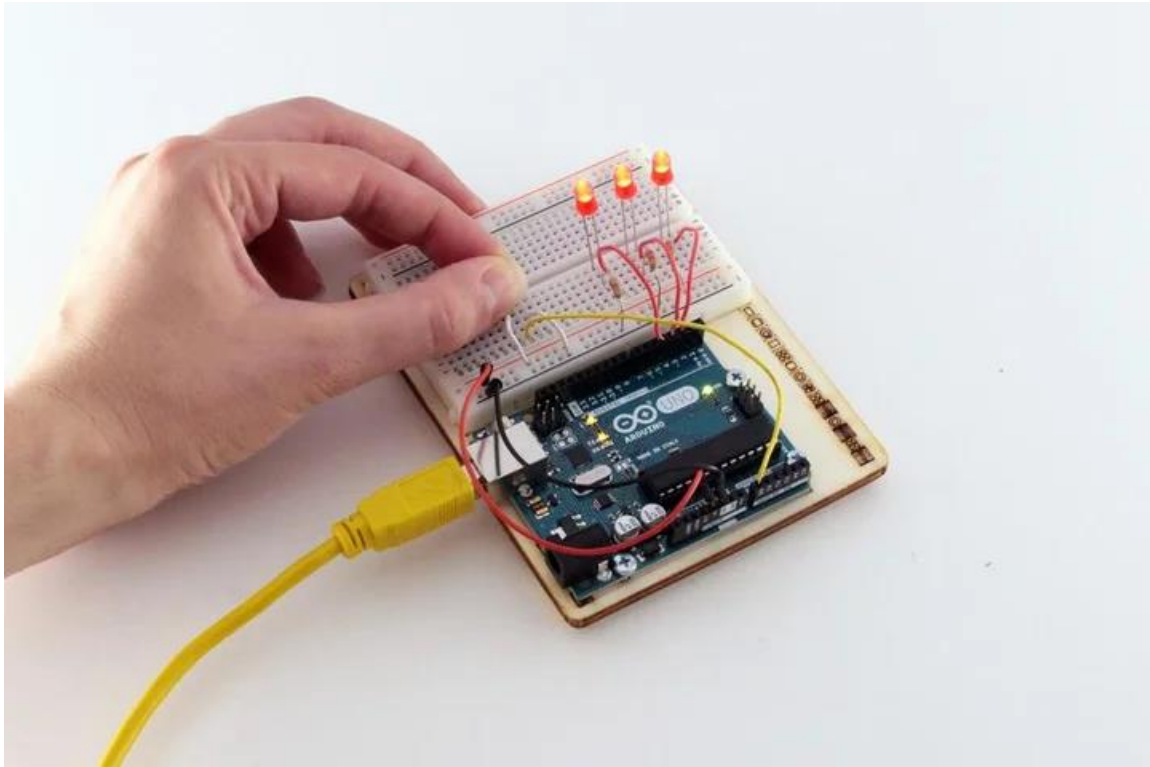
    // measure temperature in Celsius
```

```
celsius = map(((analogRead(A0) - 20) *  
3.04), 0 // convert to Fahrenheit  
  
fahrenheit = ((celsius * 9) / 5 + 32);  
  
Serial.print(celsius);  
  
Serial.print(" C, ");  
  
Serial.print(fahrenheit);  
  
Serialif (celsius < baselineTemp) {  
    digitalWrite(2, LOW);  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
}  
  
if (celsius >= baselineTemp && celsius <  
baselineTemp + 10) {  
    digitalWrite(2, HIGH);
```

```
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
}  
  
if (celsius >= baselineTemp + 10 && celsius  
< baselineTemp + 20) {  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, LOW);  
}  
  
if (celsius >= baselineTemp + 20 && celsius  
< baselineTemp + 30) {  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, HIGH);
```

```
}  
if (celsius >= baselineTemp + 30) {  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, HIGH);  
}l.println(" F");, 1023, -40, 125);
```

The loop's six if statements evaluate for different segments of a certain temperature range between 40 and 46 degrees C, lighting up more LEDs the warmer the temperature.



If you built a physical version of this circuit, you can try it out with the Arduino software's serial monitor (magnifying glass button in the upper right of the sketch window), activating the sensor with your fingers. The project might not do what you want it to if the room temperature is really cold or really warm, or if your fingers are cold!

If using a physical board, observe the room temperature using the serial monitor, and set `baselineTemp` to that value.

Adjust your different temperature threshold "buckets" to a smaller range (2, 4, 6, instead of 10, 20, 30).

Upload your code again, and try holding the sensor in your fingers. As the temperature rises, you should see the LEDs turn on one by one.

