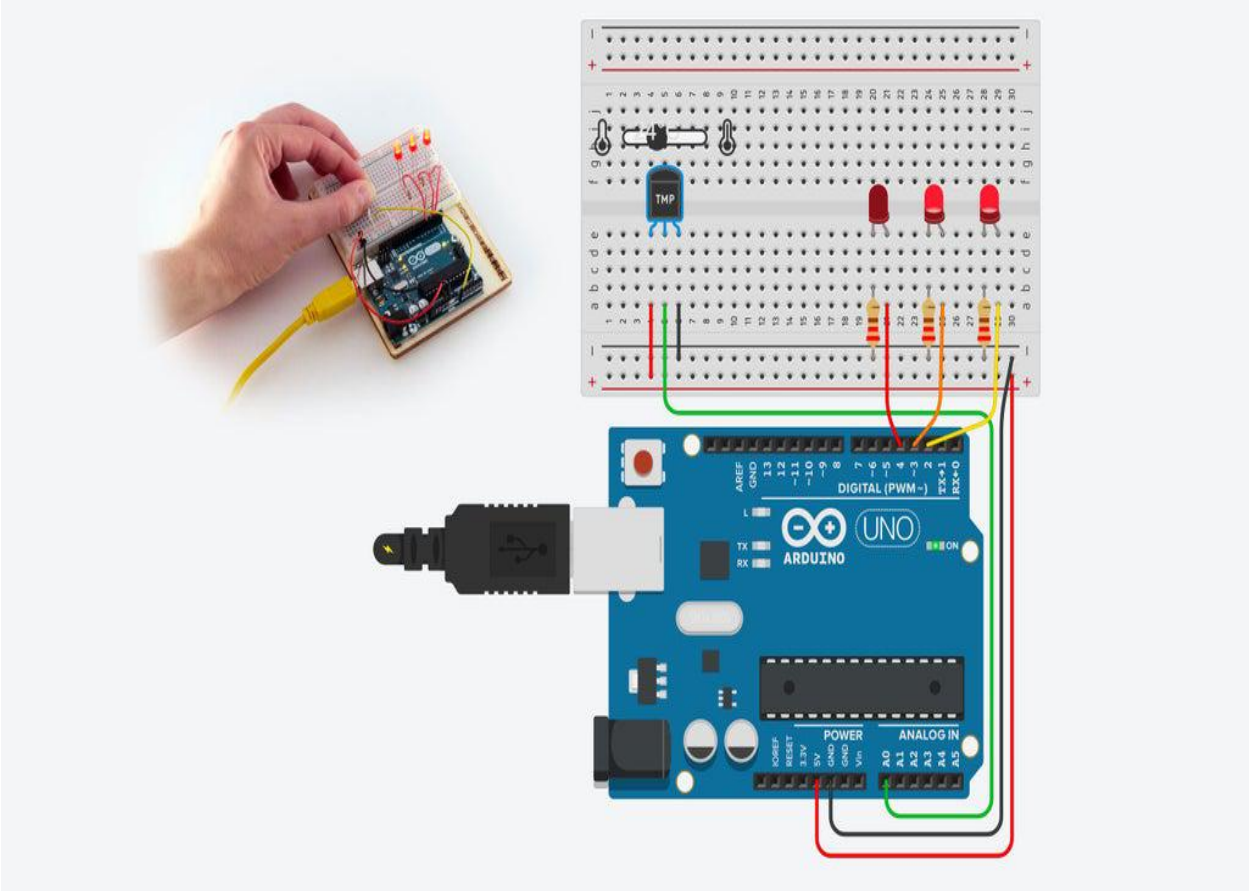
ASSIGNMENT FOR IOT

A Alarm should give one sound when there is amotion near PIR Sensor

Submitted by

M.Mageshwaran

TMP 36 TEMPERATURE SENSOR WITH ARDUINO USING PIR SENSOR

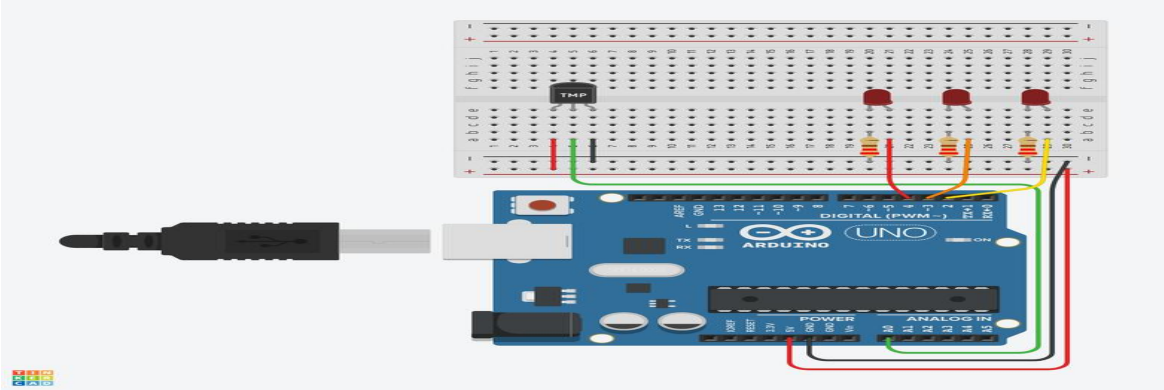


USING COMPONENTS:

- PIR SENSOR
- ARDUINO
- PIR SENSOR
- LED CIRCUIT
- BLOCKS
- USB CABLES
- WIRES

PROCEDURES :

Step 1: Build the LED Circuit



Connect the 5 volt and ground pins on the Arduino to the power (+) and ground (-) rails on the breadboard with wires. You can change the wire colors if you want to! Either use the inspector dropdown or the number keys on your keyboard.

Drag three LEDs on the breadboard in row E, spaced 2 breadboard sockets apart. You can change the LED color using the inspector that pops up when you click on each one.

Step 2: Add Temperature Sensor

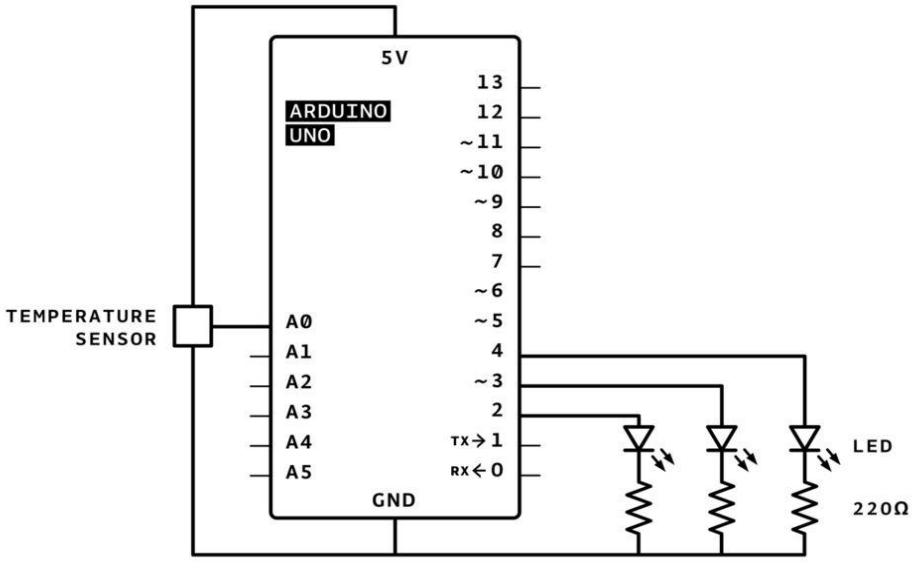
A temperature sensor creates a changing voltage signal depending on the temperature it senses. It has three pins: one that connects to ground, another that connects to 5

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 Celsius.

Step 3: 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.

Step 4: Blocks Code

Click the "Code" button to open the code editor. The grey Notation blocks are comments for making note of what you intend for your code to do, but this text isn't required or executed as part of the program.

Click on the Variables category in the code editor. Create a new variable called baselineTemp and use a "set" block to set it to 40 (degrees C).

To store the sensor value, create a variable named "celsius".

Drag out a "set" block and adjust the dropdown to our new variable celsius.

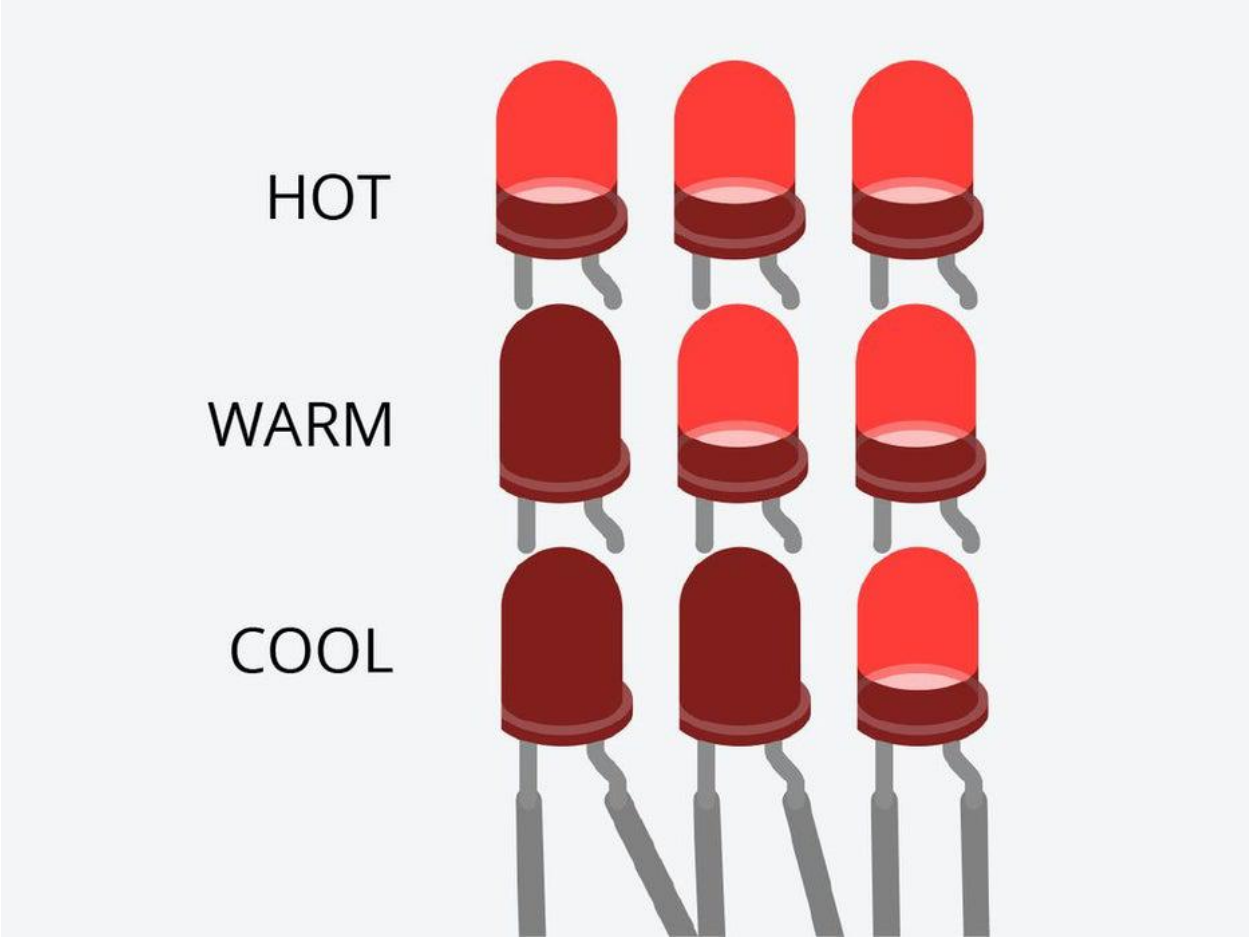
In the Math category, drag out a "map" block, and nest two arithmetic blocks ("1 + 1") within its first field.

Adjust the range from -40 to 125.

Click on the Input category and drag out an "analog read pin" block, and place it into the first arithmetic field inside the "map" block.

Adjust the arithmetic blocks to "(read analog pin A0 - 20) x 3.04".

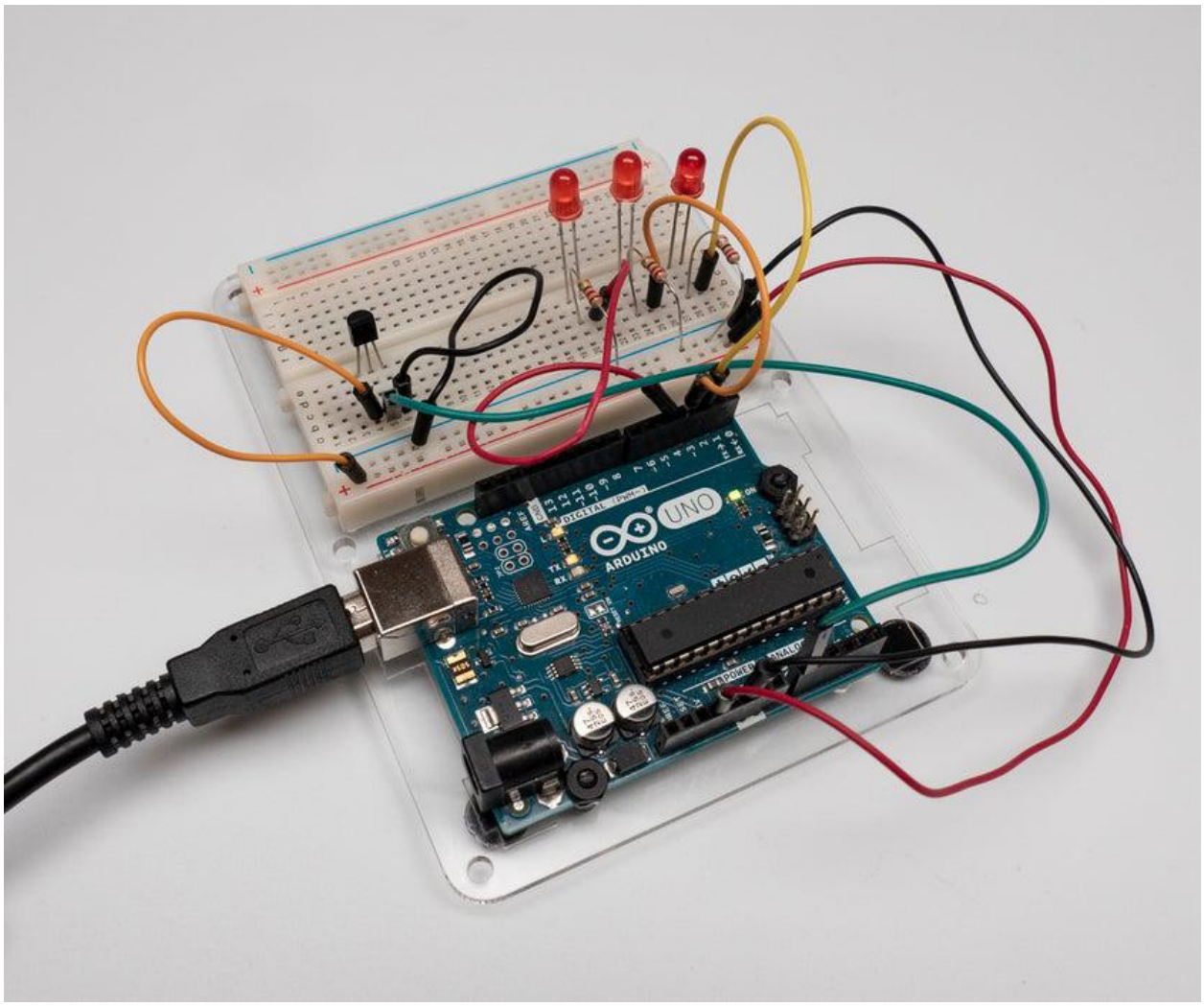
Step 5: Arduino Code Explained




```
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);
}
if (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);
  }
  delay(1000); // Wait for 1000 millisecond(s)
}
```


ARDUINO
UNO