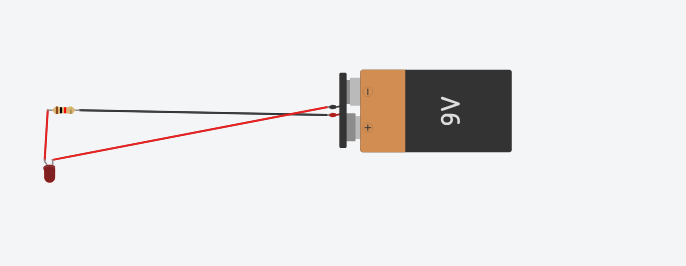
1. **Basic LED Circuit**



In this basic LED circuit:

1. A 9V battery provides power.

2. A resistor limits the current to protect the LED.

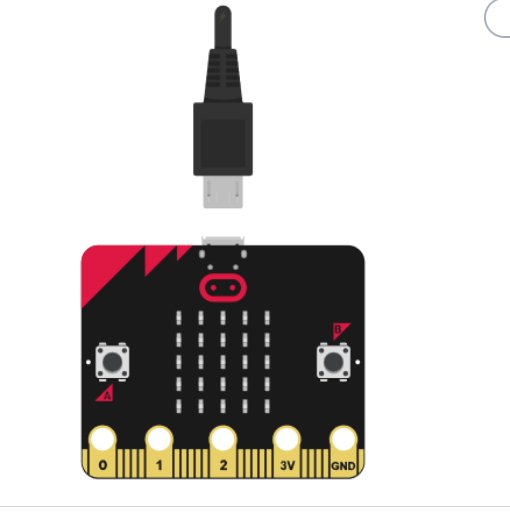
3. The positive terminal of the battery is connected to the resistor.

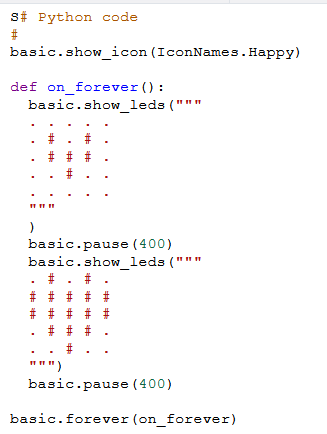
4. The other end of the resistor connects to the LED's anode (longer leg).

5. The LED's cathode (shorter leg) connects to the negative terminal of the battery.

6. When the circuit is complete, the LED lights up.

1. **Micro bit**





The micro:bit displays two alternating LED patterns in a loop. The script shows:

1. A happy face icon initially.

2. In the `on\_forever` function:

- One pattern of LEDs.

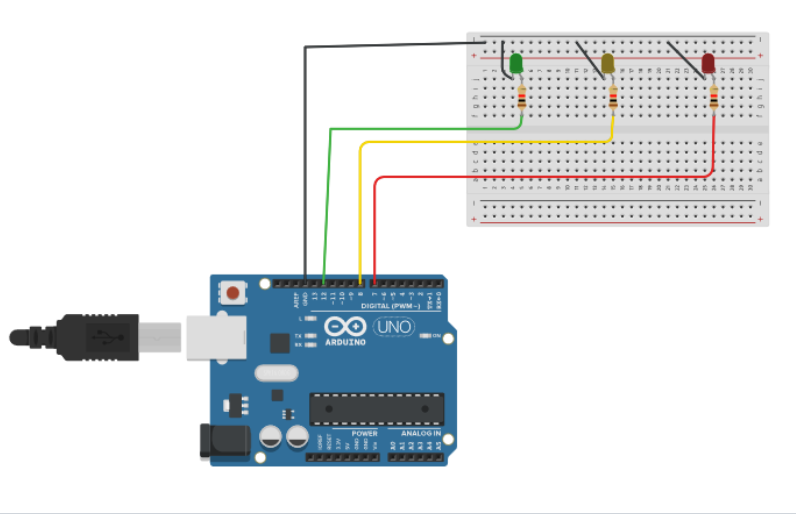
- Pauses for 400 milliseconds.

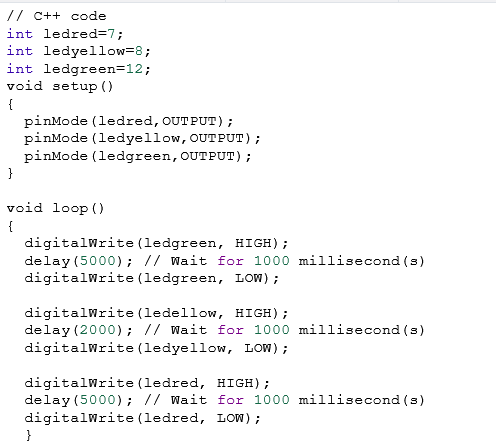
- Another pattern of LEDs.

- Pauses for 400 milliseconds.

3. The `on\_forever` function runs continuously, making the LED patterns blink.

1. **Traffic Light Control System**

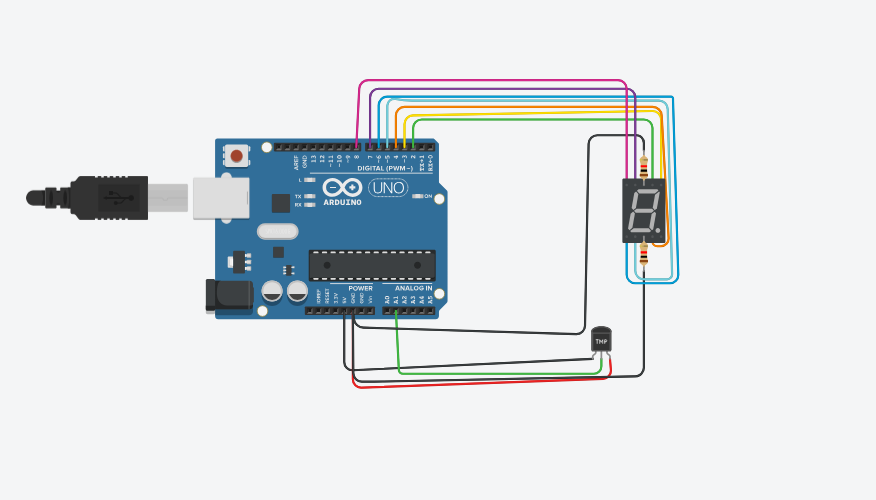




* **Green LED**: ON for 5 seconds.
* **Yellow LED**: ON for 2 seconds.
* **Red LED**: ON for 5 seconds.

This code creates a basic traffic light system that cycles through green, yellow, and red lights, with delays representing the duration each light stays on.

1. **Temperature Sensor and Display**

****

int SensorPin = A1;

int pinA = 2;

int pinB = 3;

int pinC = 4;

int pinD = 5;

int pinE = 6;

int pinF = 7;

int pinG = 8;

float ReadValue, mVout, tempC;

void setup() {

pinMode(SensorPin, INPUT);

pinMode(pinA, OUTPUT);

pinMode(pinB, OUTPUT);

pinMode(pinC, OUTPUT);

pinMode(pinD, OUTPUT);

pinMode(pinE, OUTPUT);

pinMode(pinF, OUTPUT);

pinMode(pinG, OUTPUT);

Serial.begin(9600);

}

void loop() {

// Read the temperature from the sensor

ReadValue = analogRead(SensorPin);

mVout = (5000.0 / 1023.0) \* ReadValue;

tempC = mVout / 10;

Serial.print("Output Voltage in mV : ");

Serial.println(mVout);

Serial.print("Temperature in C : ");

Serial.println(tempC);

// Display H if temperature is 25°C or above

if (tempC >= 25) {

displayH();

}

// Display L if temperature is 20°C or below

else if (tempC <= 20) {

displayL();

}

// Turn off all segments if temperature is between 20°C and 25°C

else {

turnOffAllSegments();

}

delay(1000);

}

void displayH() {

digitalWrite(pinA, LOW);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, LOW);

digitalWrite(pinE, HIGH);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, HIGH);

}

void displayL() {

digitalWrite(pinA, LOW);

digitalWrite(pinB, LOW);

digitalWrite(pinC, LOW);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, HIGH);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, LOW);

}

void turnOffAllSegments() {

digitalWrite(pinA, LOW);

digitalWrite(pinB, LOW);

digitalWrite(pinC, LOW);

digitalWrite(pinD, LOW);

digitalWrite(pinE, LOW);

digitalWrite(pinF, LOW);

digitalWrite(pinG, LOW);

}

**Brief Explanation**

This project demonstrates how to use an Arduino to read the temperature from an analog temperature sensor and display the result on a 7-segment display. Depending on the temperature reading, the display will show the letter "H" for high temperature (25°C or above) or "L" for low temperature (20°C or below).