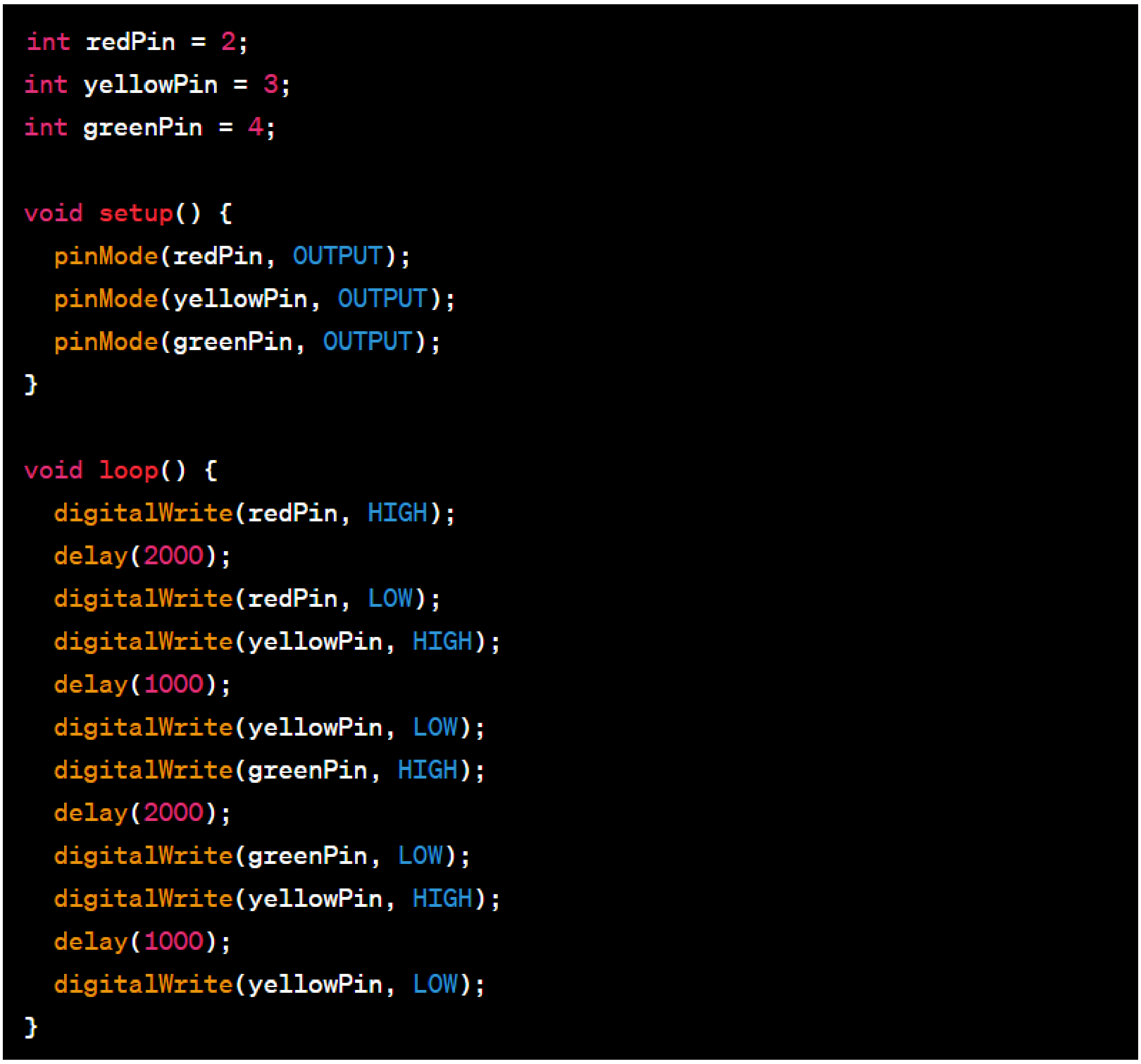
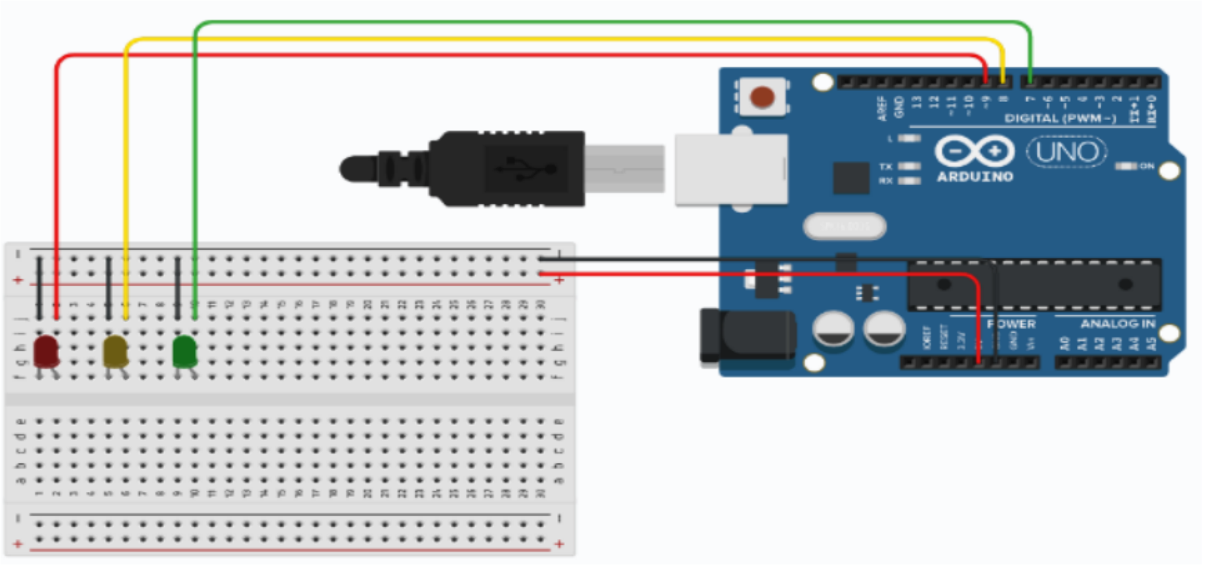
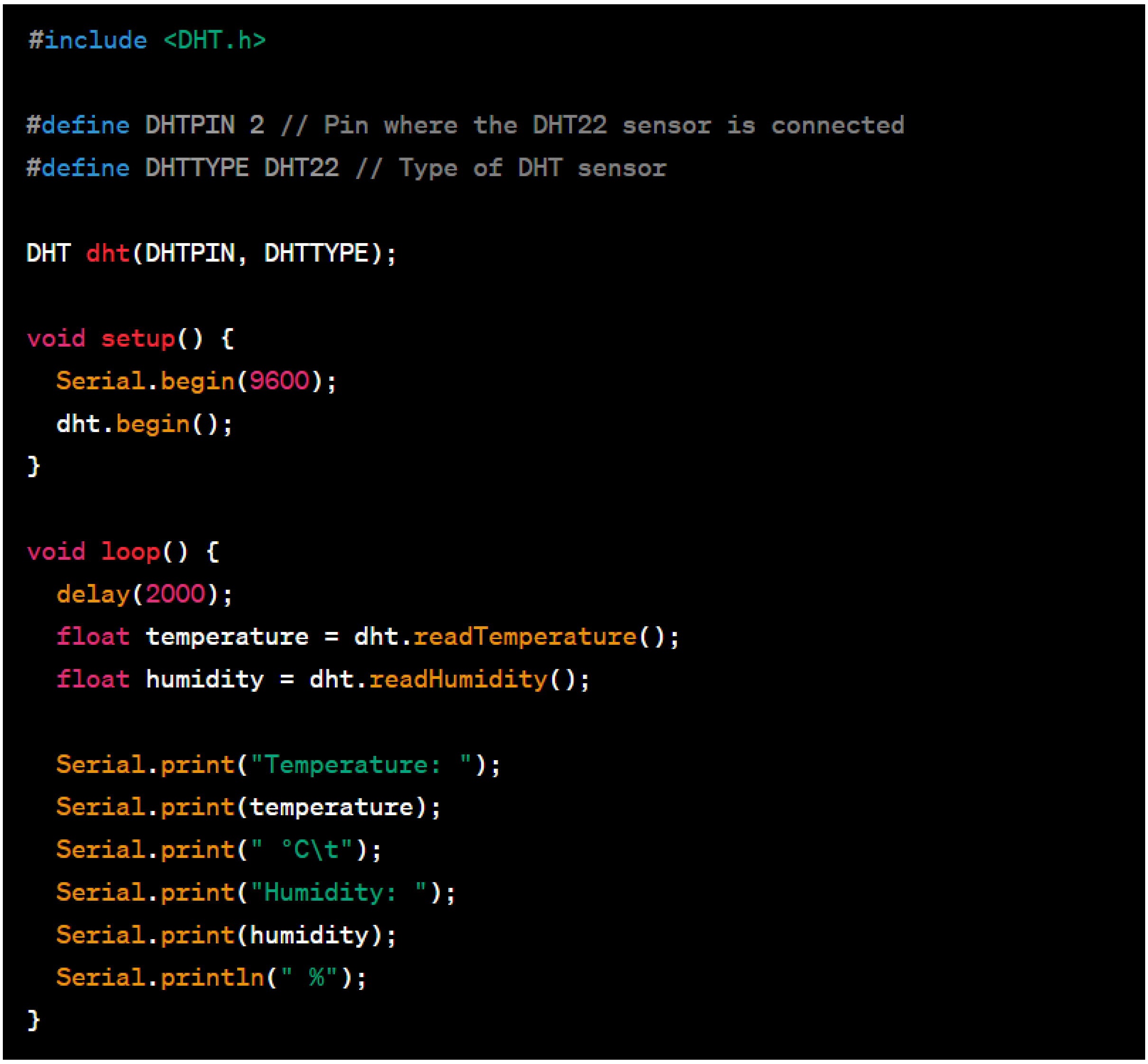
1. Design of Traffic light using Arduino.



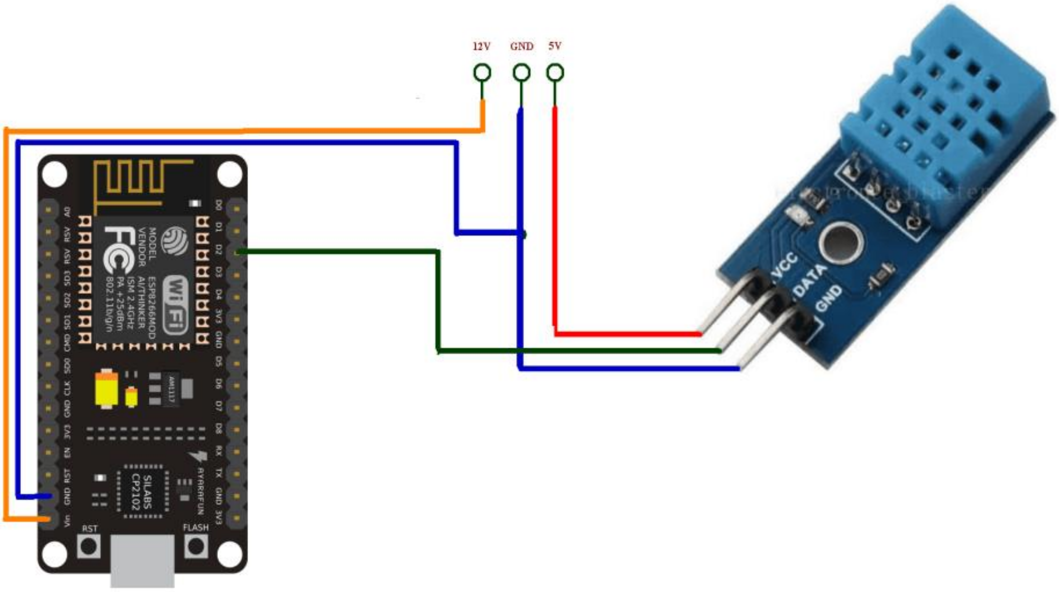
```
int redPin = 2;
int yellowPin = 3;
int greenPin = 4;

void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(yellowPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
}

void loop() {
  digitalWrite(redPin, HIGH);
  delay(2000);
  digitalWrite(redPin, LOW);
  digitalWrite(yellowPin, HIGH);
  delay(1000);
  digitalWrite(yellowPin, LOW);
  digitalWrite(greenPin, HIGH);
  delay(2000);
  digitalWrite(greenPin, LOW);
  digitalWrite(yellowPin, HIGH);
  delay(1000);
  digitalWrite(yellowPin, LOW);
}
```



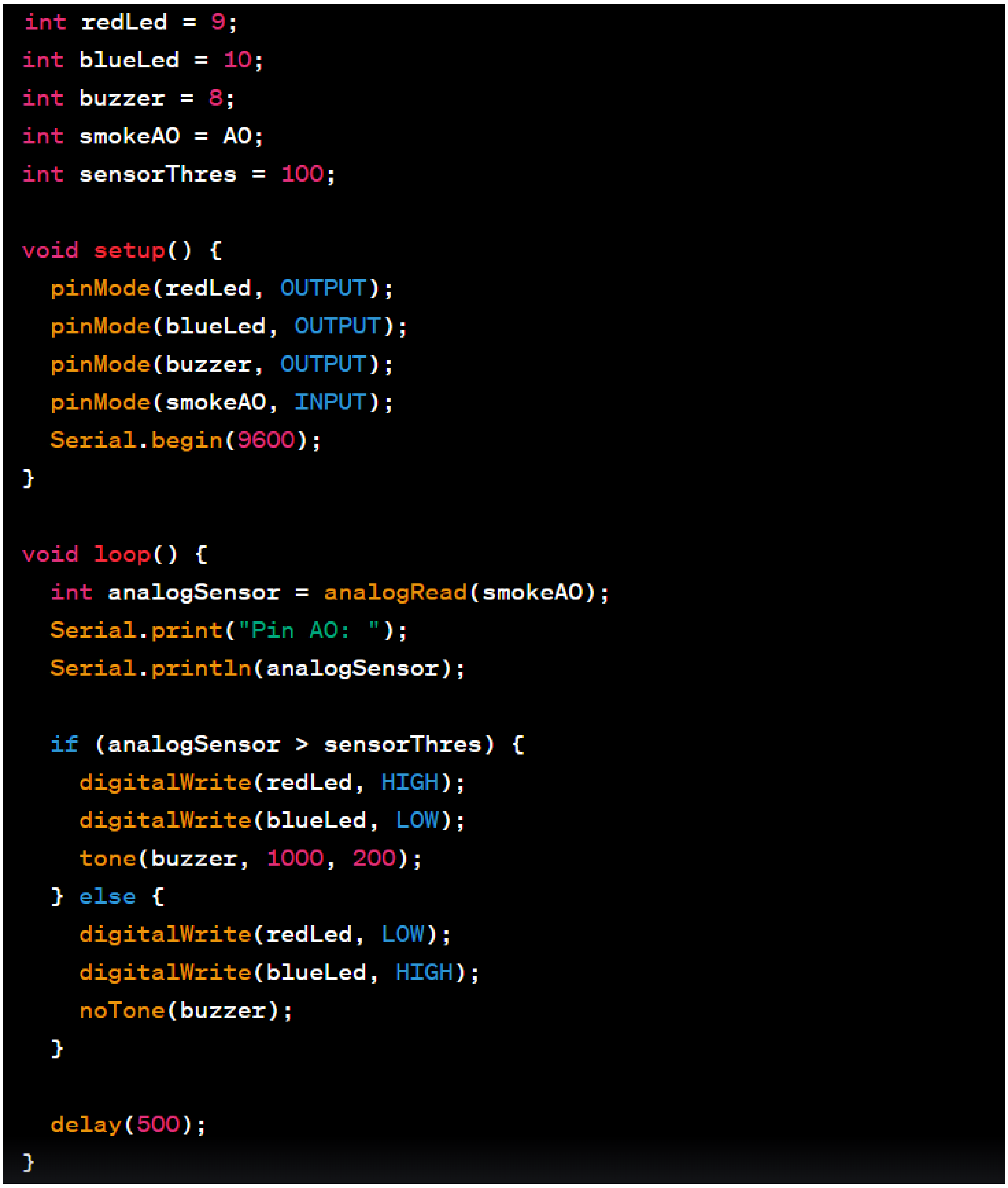
2. **To Print Temperature and Humidity in serial monitor using appropriate sensor.**



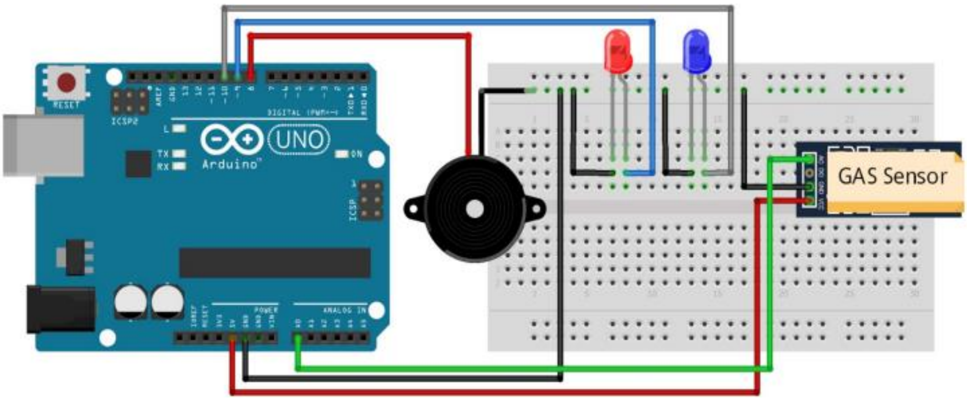
```
#include <DHT.h>

#define DHTPIN 2 // Pin where the DHT22 sensor is connected
#define DHTTYPE DHT22 // Type of DHT sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000);
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C\t");
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");
}
```



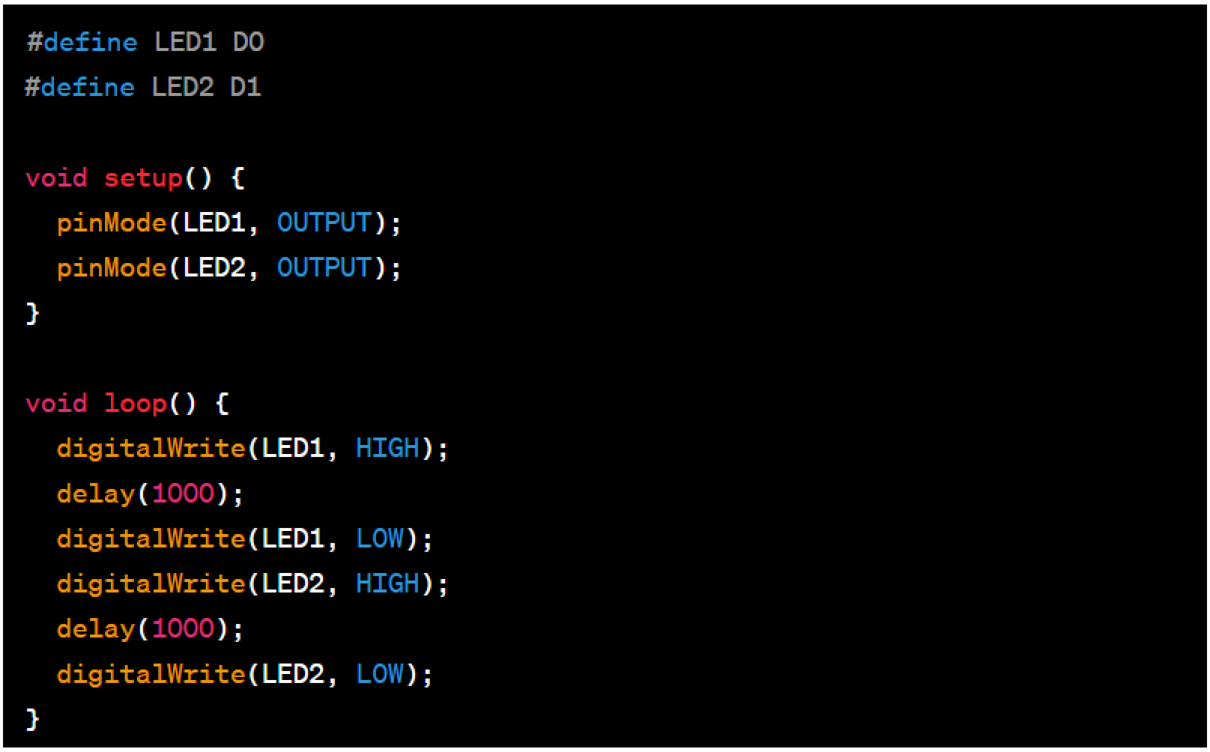
3. **To Detect Smoke and turn on the buzzer by using an appropriate sensor.**
4. **To Detect Alcohol and turn on the buzzer by using an appropriate sensor.**



```
int redLed = 9;
int blueLed = 10;
int buzzer = 8;
int smokeA0 = A0;
int sensorThres = 100;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(blueLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int analogSensor = analogRead(smokeA0);
  Serial.print("Pin A0: ");
  Serial.println(analogSensor);

  if (analogSensor > sensorThres) {
    digitalWrite(redLed, HIGH);
    digitalWrite(blueLed, LOW);
    tone(buzzer, 1000, 200);
  } else {
    digitalWrite(redLed, LOW);
    digitalWrite(blueLed, HIGH);
    noTone(buzzer);
  }

  delay(500);
}
```

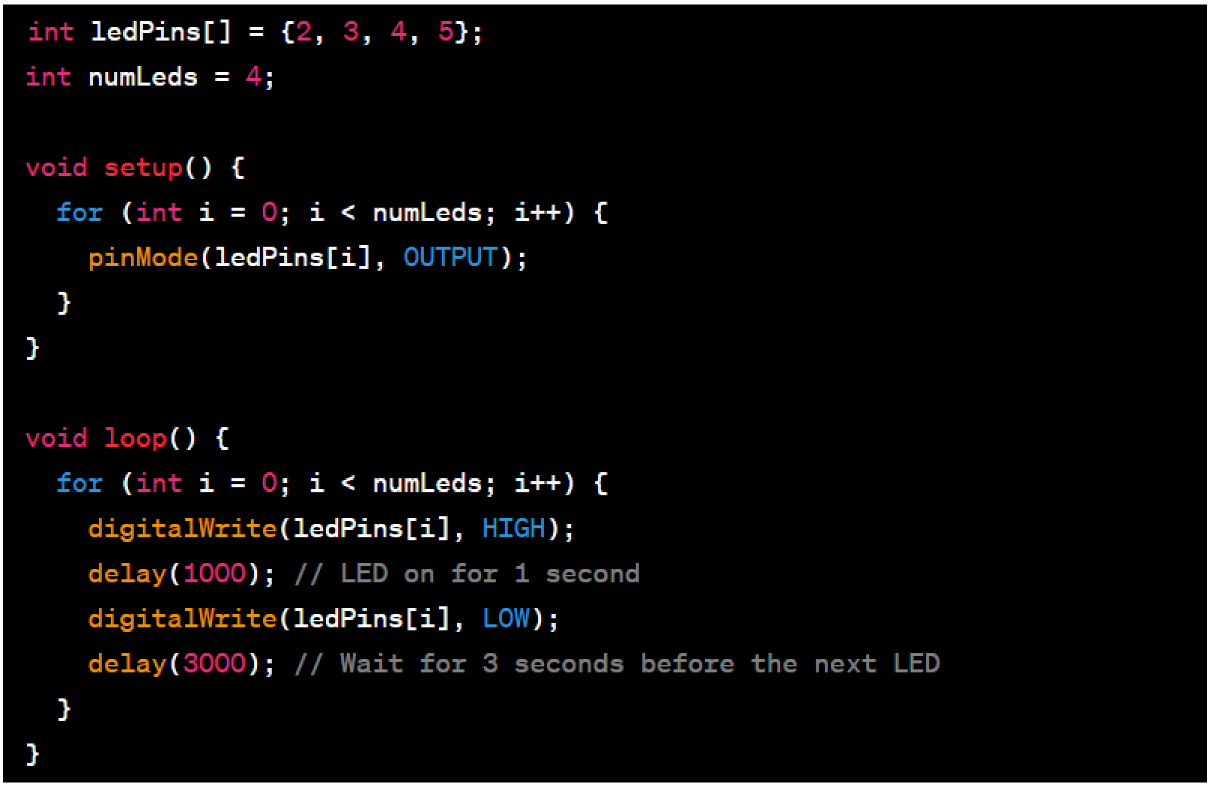


5. **To glow two LEDs serially using NODE MCU.**



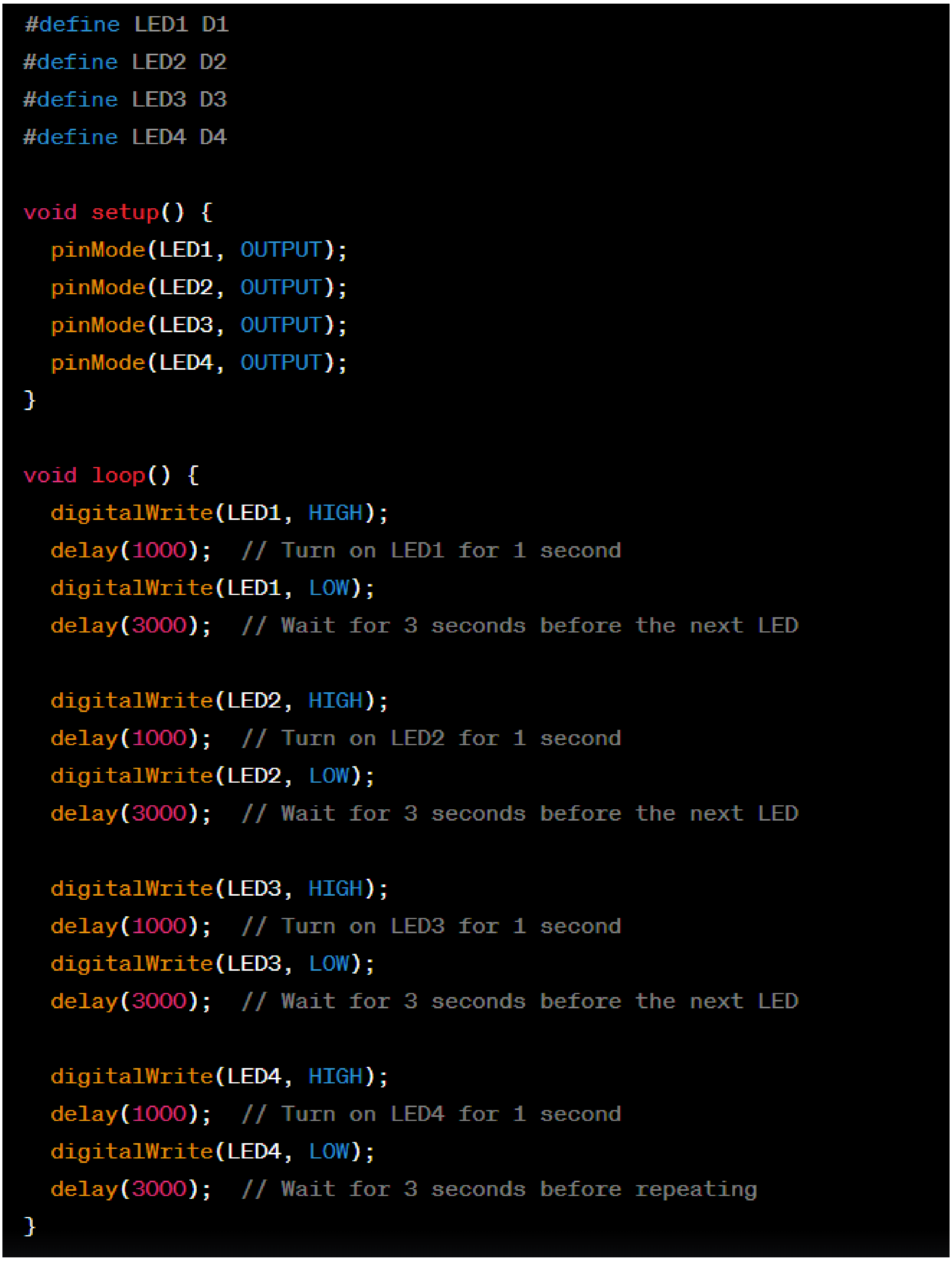
```
#define LED1 D0
#define LED2 D1

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
}

void loop() {
  digitalWrite(LED1, HIGH);
  delay(1000);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
  delay(1000);
  digitalWrite(LED2, LOW);
}
```

6. **To Blink 4 LEDs one by one with 3sec delay using Arduino.**

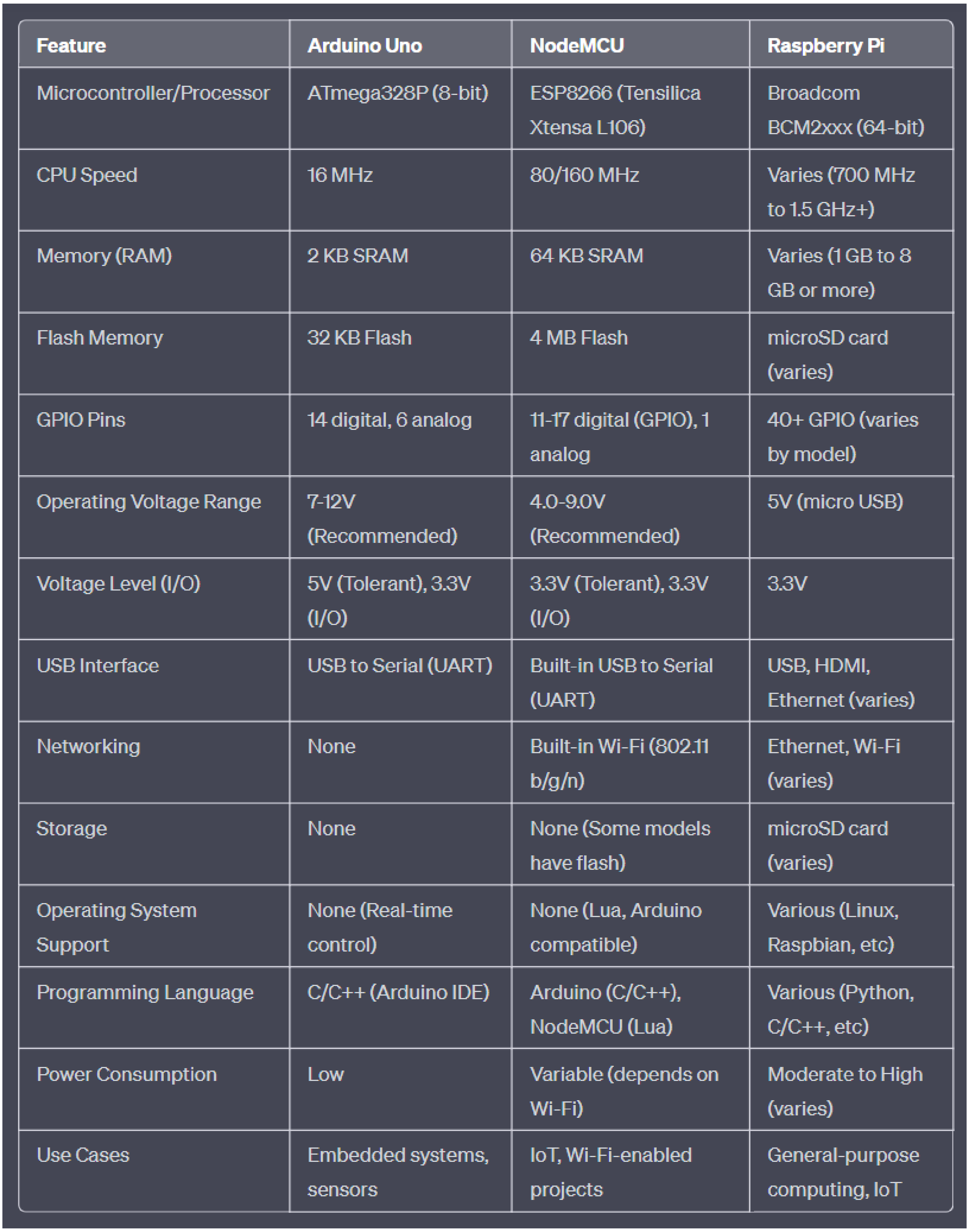


```
int ledPins[] = {2, 3, 4, 5};
int numLeds = 4;

void setup() {
  for (int i = 0; i < numLeds; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop() {
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(1000); // LED on for 1 second
    digitalWrite(ledPins[i], LOW);
    delay(3000); // Wait for 3 seconds before the next LED
  }
}
```

7. To Blink 4 LEDs one by one with 3sec delay using NodeMCU.



```
#define LED1 D1
#define LED2 D2
#define LED3 D3
#define LED4 D4

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
}

void loop() {
  digitalWrite(LED1, HIGH);
  delay(1000);  // Turn on LED1 for 1 second
  digitalWrite(LED1, LOW);
  delay(3000);  // Wait for 3 seconds before the next LED

  digitalWrite(LED2, HIGH);
  delay(1000);  // Turn on LED2 for 1 second
  digitalWrite(LED2, LOW);
  delay(3000);  // Wait for 3 seconds before the next LED

  digitalWrite(LED3, HIGH);
  delay(1000);  // Turn on LED3 for 1 second
  digitalWrite(LED3, LOW);
  delay(3000);  // Wait for 3 seconds before the next LED

  digitalWrite(LED4, HIGH);
  delay(1000);  // Turn on LED4 for 1 second
  digitalWrite(LED4, LOW);
  delay(3000);  // Wait for 3 seconds before repeating
}
```



Feature	Arduino Uno	NodeMCU	Raspberry Pi
Microcontroller/Processor	ATmega328P (8-bit)	ESP8266 (Tensilica Xtensa L106)	Broadcom BCM2xxx (64-bit)
CPU Speed	16 MHz	80/160 MHz	Varies (700 MHz to 1.5 GHz+)
Memory (RAM)	2 KB SRAM	64 KB SRAM	Varies (1 GB to 8 GB or more)
Flash Memory	32 KB Flash	4 MB Flash	microSD card (varies)
GPIO Pins	14 digital, 6 analog	11-17 digital (GPIO), 1 analog	40+ GPIO (varies by model)
Operating Voltage Range	7-12V (Recommended)	4.0-9.0V (Recommended)	5V (micro USB)
Voltage Level (I/O)	5V (Tolerant), 3.3V (I/O)	3.3V (Tolerant), 3.3V (I/O)	3.3V
USB Interface	USB to Serial (UART)	Built-in USB to Serial (UART)	USB, HDMI, Ethernet (varies)
Networking	None	Built-in Wi-Fi (802.11 b/g/n)	Ethernet, Wi-Fi (varies)
Storage	None	None (Some models have flash)	microSD card (varies)
Operating System Support	None (Real-time control)	None (Lua, Arduino compatible)	Various (Linux, Raspbian, etc)
Programming Language	C/C++ (Arduino IDE)	Arduino (C/C++), NodeMCU (Lua)	Various (Python, C/C++, etc)
Power Consumption	Low	Variable (depends on Wi-Fi)	Moderate to High (varies)
Use Cases	Embedded systems, sensors	IoT, Wi-Fi-enabled projects	General-purpose computing, IoT