Lab 6

ICS 423: Internet Of Things

Jayant Kolapkar - 2021BCS0132

Task 1: Design an alarm system that triggers warning if a package gets damaged. Use force sensor and a buzzer alarm.

Algorithm

1. Initialize Components

- Define force sensor pin (Analog A0).
- Define buzzer pin (Digital 9).
- Define LED pin (Digital 7).
- Set the threshold value for the force sensor.

2. Setup Function (Runs Once)

- · Start the Serial Monitor for debugging.
- Set LED and Buzzer as OUTPUT.

3. Loop Function (Repeats Continuously)

- Read the force sensor value from A0.
- Print the sensor value to the Serial Monitor.
- Compare the sensor value with the threshold:
    - If the value is greater than the threshold:
        - Turn ON the LED.
        - Turn ON the Buzzer.
    - Else:
        - Turn OFF the LED.
        - Turn OFF the Buzzer.

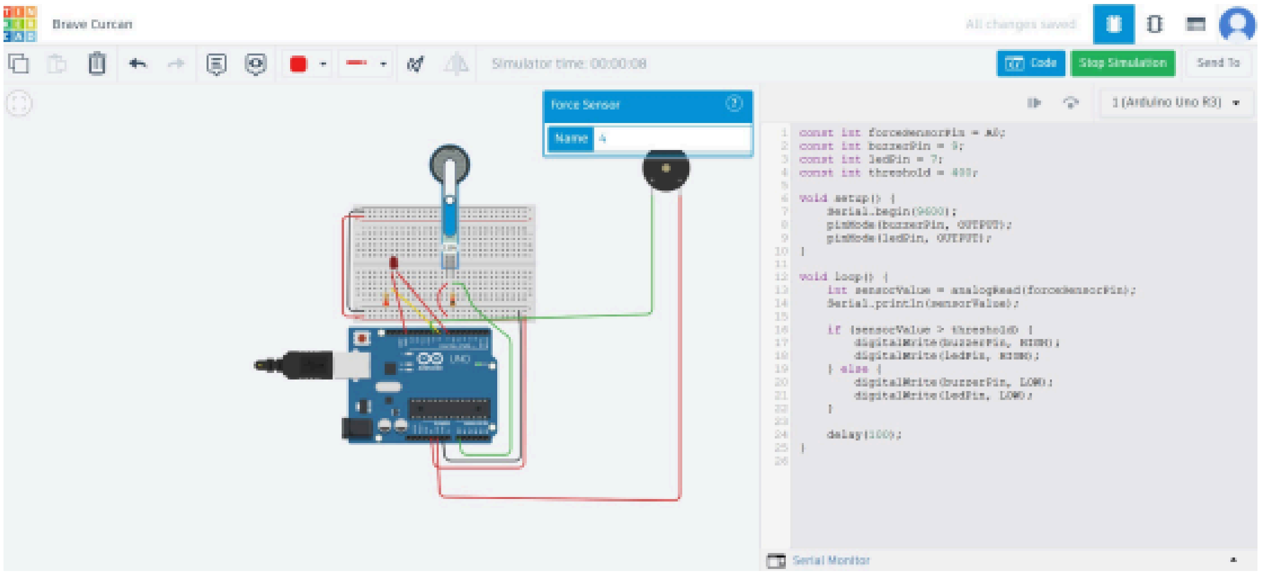
- Add a small delay for stability.

Code

```
const int forceSensorPin = A0;
const int buzzerPin = 9;
const int ledPin = 7;
const int threshold = 400;

void setup() {
        Serial.begin(9600);
        pinMode(buzzerPin, OUTPUT);
        pinMode(ledPin, OUTPUT);
}

void loop() {
        int sensorValue = analogRead(forceSensorPin);
        Serial.println(sensorValue);

        if (sensorValue > threshold) {
        digitalWrite(buzzerPin, HIGH);
        digitalWrite(ledPin, HIGH);

        } else {

        digitalWrite(buzzerPin, LOW);

        digitalWrite(ledPin, LOW);

        }



        delay(100);

}
```

Output

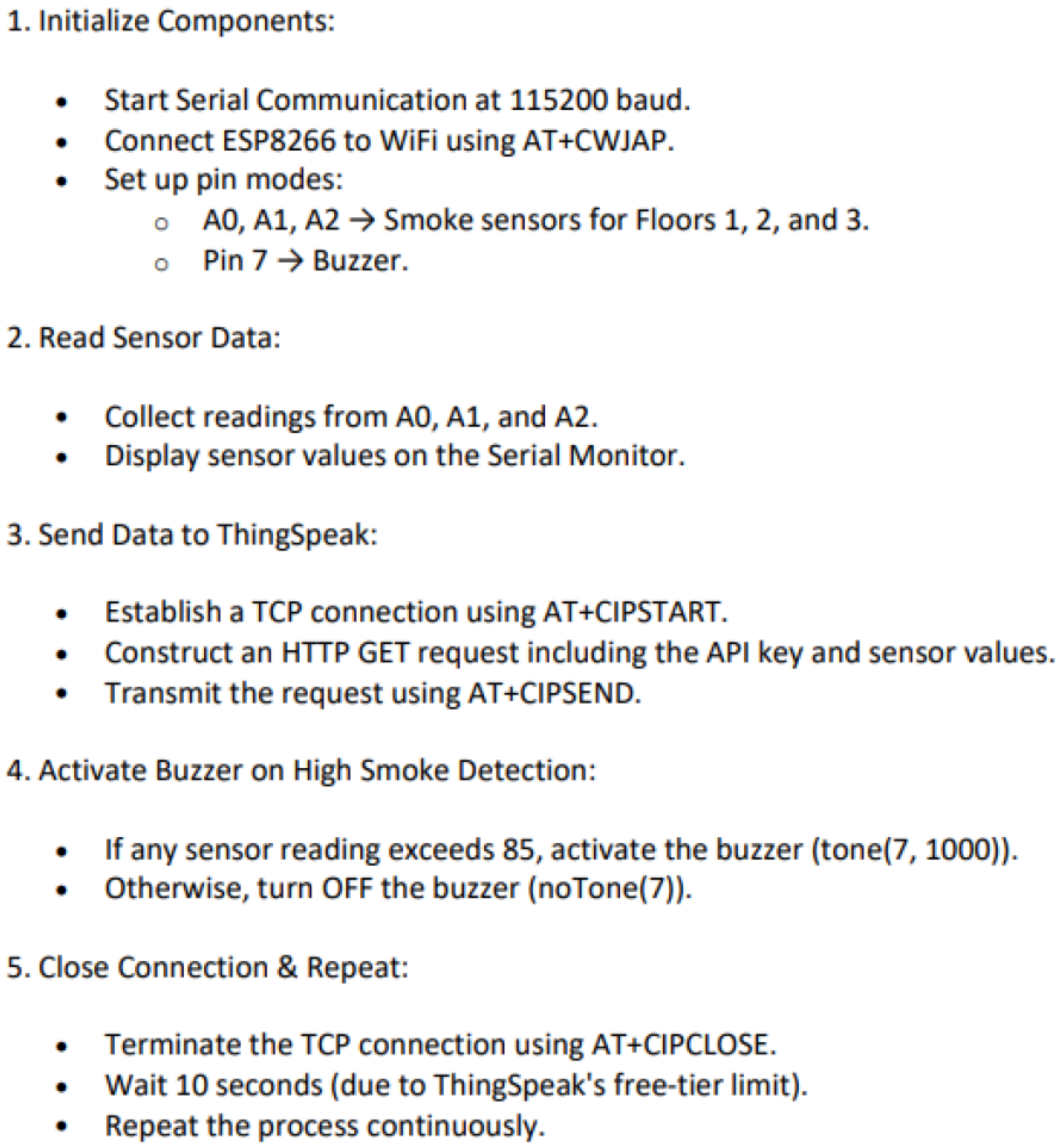
The alarm and the LED bulb are off when the force applied is below the threshold.



```
const int forceSensorPin = A0;
const int buzzerPin = 9;
const int ledPin = 7;
const int threshold = 400;

void setup() {
    Serial.begin(9600);
    pinMode(buzzerPin, OUTPUT);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    int sensorValue = analogRead(forceSensorPin);
    Serial.println(sensorValue);

    if (sensorValue > threshold) {
        digitalWrite(buzzerPin, HIGH);
        digitalWrite(ledPin, HIGH);
    } else {
        digitalWrite(buzzerPin, LOW);
        digitalWrite(ledPin, LOW);
    }

    delay(100);
}
```

The alarm rings and the LED bulb glows when the force crosses the threshold limit.

Task 2: Develop a smoke detecting system for a building consisting of three floors. Notify the real-time value of the floors to thingspeak IoT cloud platform.



1. Initialize Components:

- Start Serial Communication at 115200 baud.
- Connect ESP8266 to WiFi using AT+CWJAP.
- Set up pin modes:
  - A0, A1, A2 → Smoke sensors for Floors 1, 2, and 3.
  - Pin 7 → Buzzer.

2. Read Sensor Data:

- Collect readings from A0, A1, and A2.
- Display sensor values on the Serial Monitor.

3. Send Data to ThingSpeak:

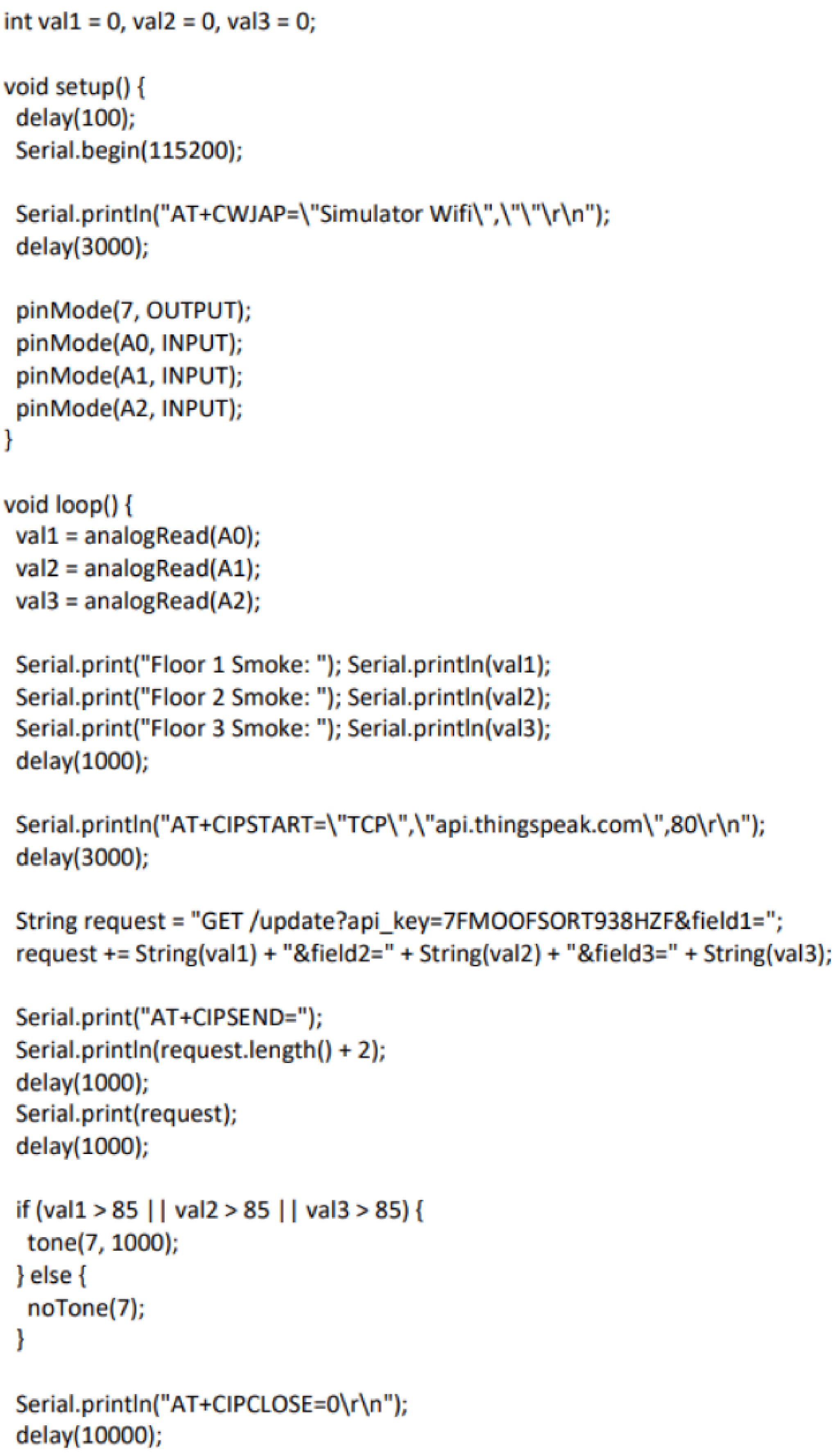
- Establish a TCP connection using AT+CIPSTART.
- Construct an HTTP GET request including the API key and sensor values.
- Transmit the request using AT+CIPSEND.

4. Activate Buzzer on High Smoke Detection:

- If any sensor reading exceeds 85, activate the buzzer (tone(7, 1000)).
- Otherwise, turn OFF the buzzer (noTone(7)).

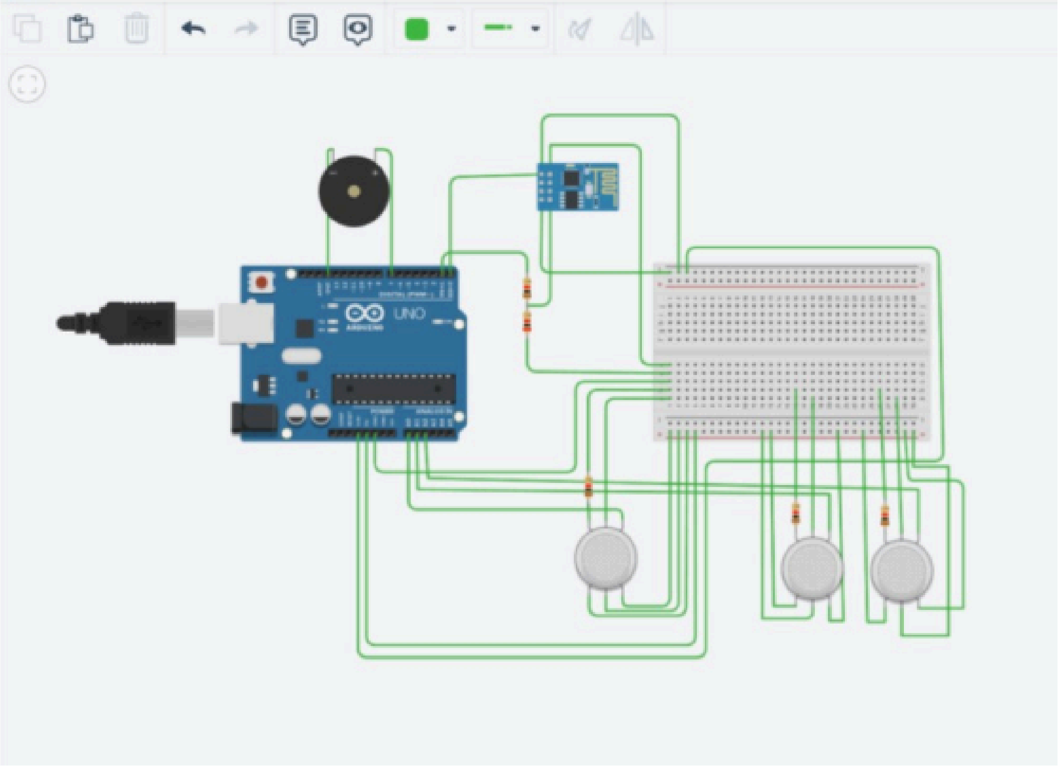
5. Close Connection & Repeat:

- Terminate the TCP connection using AT+CIPCLOSE.
- Wait 10 seconds (due to ThingSpeak's free-tier limit).
- Repeat the process continuously.

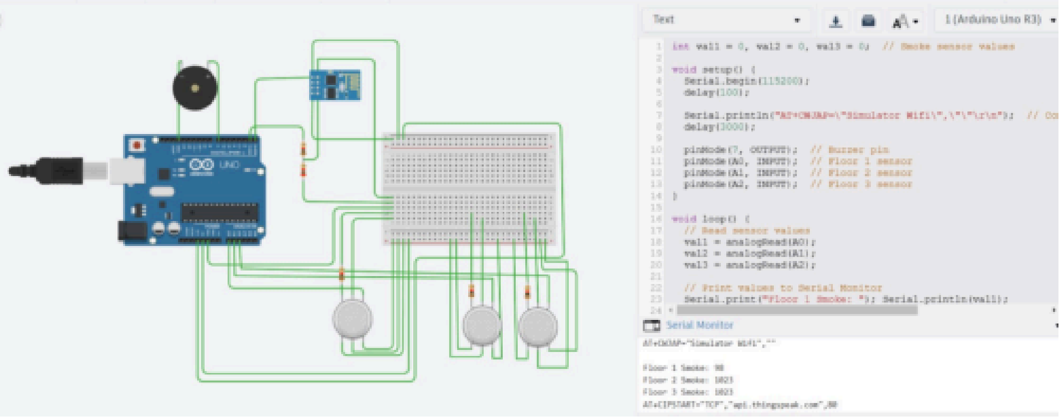


```
int val1 = 0, val2 = 0, val3 = 0;

void setup() {
 delay(100);
 Serial.begin(115200);

 Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"\r\n");
 delay(3000);

 pinMode(7, OUTPUT);
 pinMode(A0, INPUT);
 pinMode(A1, INPUT);
 pinMode(A2, INPUT);
}

void loop() {
 val1 = analogRead(A0);
 val2 = analogRead(A1);
 val3 = analogRead(A2);

 Serial.print("Floor 1 Smoke: "); Serial.println(val1);
 Serial.print("Floor 2 Smoke: "); Serial.println(val2);
 Serial.print("Floor 3 Smoke: "); Serial.println(val3);
 delay(1000);

 Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n");
 delay(3000);

 String request = "GET /update?api_key=7FMOOFSORT938HZF&field1=";
 request += String(val1) + "&field2=" + String(val2) + "&field3=" + String(val3);

 Serial.print("AT+CIPSEND=");
 Serial.println(request.length() + 2);
 delay(1000);
 Serial.print(request);
 delay(1000);

 if (val1 > 85 || val2 > 85 || val3 > 85) {
  tone(7, 1000);
 } else {
  noTone(7);
 }

 Serial.println("AT+CIPCLOSE=0\r\n");
 delay(10000);
```

Output



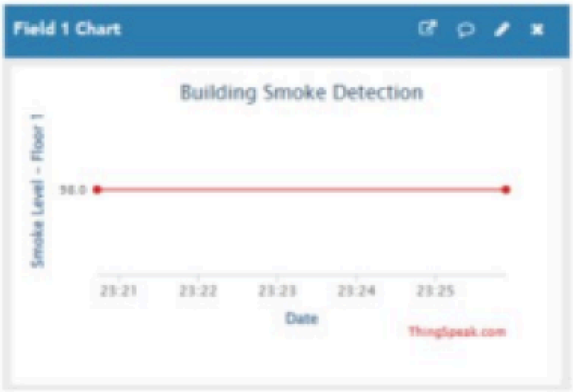
Simulation

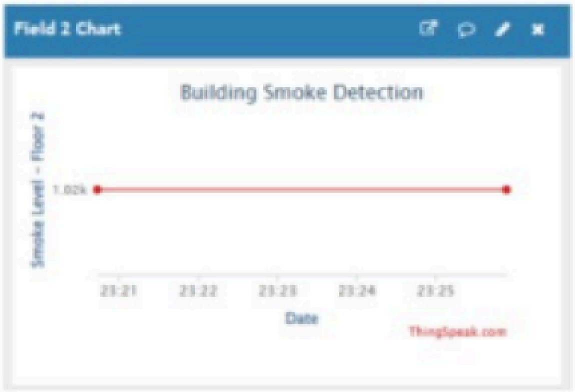


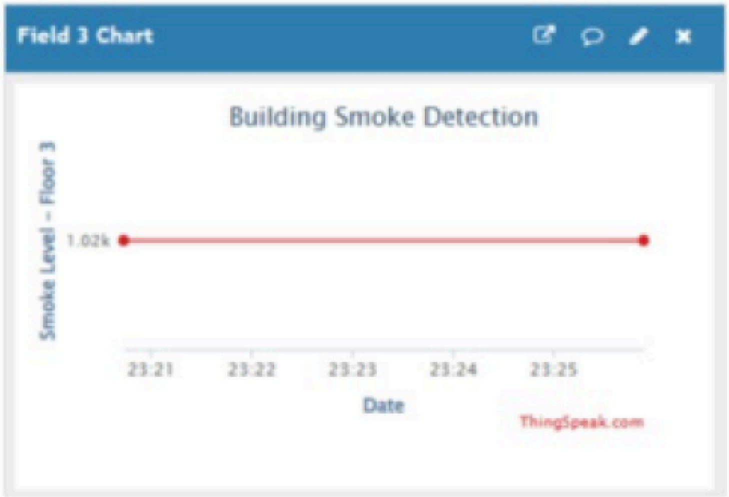
```
int val1 = 0, val2 = 0, val3 = 0;  // Smoke sensor values

void setup() {
  Serial.begin(115200);
  delay(100);

  Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"\r\n");  // Co
  delay(3000);

  pinMode(7, OUTPUT);  // Buzzer pin
  pinMode(A0, INPUT);  // Floor 1 sensor
  pinMode(A1, INPUT);  // Floor 2 sensor
  pinMode(A2, INPUT);  // Floor 3 sensor
}

void loop() {
  // Read sensor values
  val1 = analogRead(A0);
  val2 = analogRead(A1);
  val3 = analogRead(A2);

  // Print values to Serial Monitor
  Serial.print("Floor 1 Smoke: "); Serial.println(val1);
```

```
AT+CWJAP="Simulator Wifi",""

Floor 1 Smoke: 98
Floor 2 Smoke: 1023
Floor 3 Smoke: 1023
AT+CIPSTART="TCP","api.thingspeak.com",80
```

Graph visualization on ThingSpeak


Field 1 Chart
Building Smoke Detection
Smoke Level - Floor 1
98.0
23:21
23:22
23:23
23:24
23:25
Date
ThingSpeak.com


Field 2 Chart
Building Smoke Detection
Smoke Level - Floor 2
1.02k
23:21
23:22
23:23
23:24
23:25
Date
ThingSpeak.com


Field 3 Chart
Building Smoke Detection
Smoke Level - Floor 3
1.02k
23:21
23:22
23:23
23:24
23:25
Date
ThingSpeak.com