FLOOD MONITORING SYSTEM

DESCRIPTION:

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem. Please think on a design and present in form of a document. The project involves using IoT devices and data analytics to monitor traffic flow and congestion in real-time, providing commuters with access to this information through a public platform or mobile apps. The objective is to help commuters make informed decisions about their routes and alleviate traffic congestion. This project includes defining objectives, designing the IoT traffic monitoring system, developing the traffic information platform, and integrating them using IoT technology and Python.

Design Thinking:

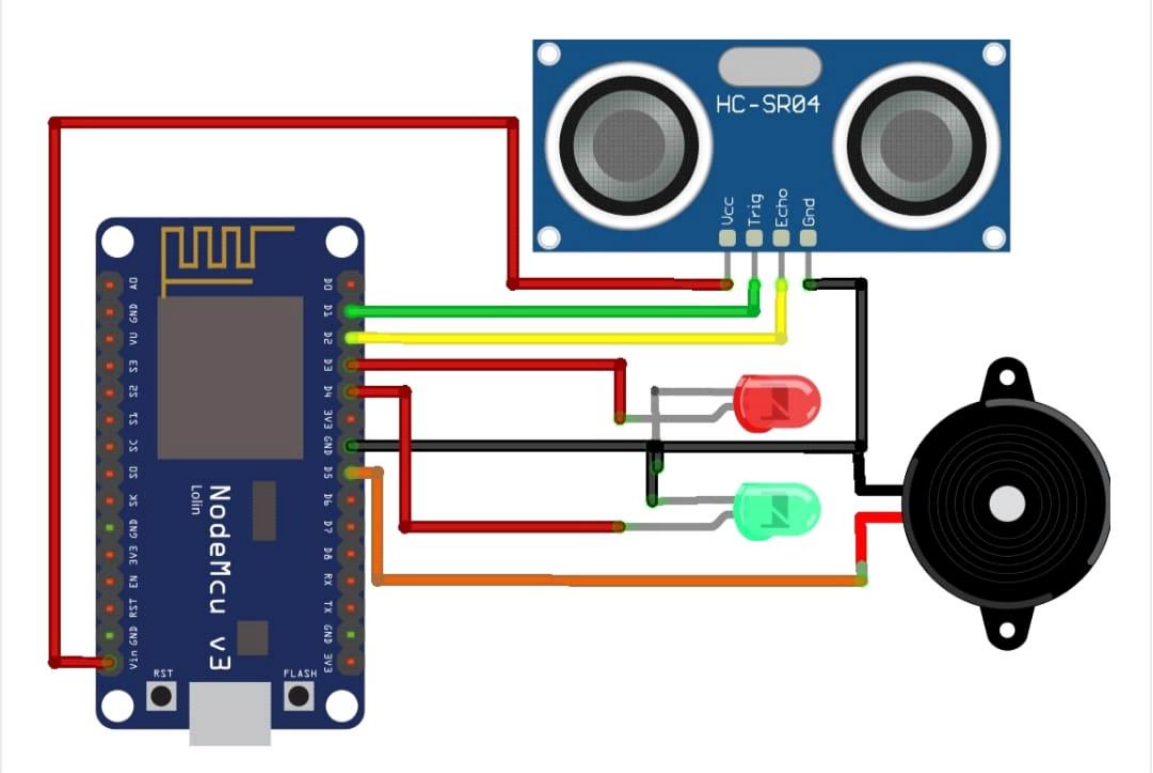
1. Define objectives such as real-time traffic monitoring, congestion detection, route optimization, and improved commuting experience..

2. IoT Sensor Design: Plan the deployment of IoT devices (sensors) to monitor traffic flow and congestion.

3.Real-Time Transit Information Platform: Design a web-based platform and mobile apps to display real-time traffic information to the public.

4. Integration Approach: Design a web-based platform and mobile apps to display real-time traffic information to the public.

ARCHITECTURE DESIGN FLOW:



DESCRIPTION OF COMPONENTS:

1.Node MCU ESP8266

2.Ultrasonic HC-SR04 sensor

3.Buzzer

4.LED -2 nos (Red and Green)

HC-SR04 ULTRASONIC SENSOR FUNCTION:

Connect Green and Red LEDs Negative Terminal to GND and Positive terminal to the D4 and D3 pin respectively. lastly, connect Buzzer GND to GND and Positive terminal to D5 pin of NodeMCU.

INNOVATIVE IDEA USE:

1.The rapid response water gate

2.Aquabex flood guard

3.The Thames Barrier

4.Water Inflated property protector

COMPONENTS PROGRAMMING:

HR-SR04 ULTRASONIC SENSOR PROGRAM:

```
#include
LiquidCrystal lcd(2,3,4,5,6,7);
float t = 0;
float dist = 0;
void setup()
{
 lcd.begin(16,2);
 pinMode(18,OUTPUT); //trigger pin
 pinMode(19,INPUT);  //echo pin
 pinMode(20,OUTPUT); //buzzer
 lcd.setCursor(0,1);
 lcd.print(" Water Level Detector");
 delay(2000);
}
void loop()
{
 lcd.clear();
 digitalWrite(20,LOW);
 digitalWrite(18,LOW);
 delayMicroseconds(2);
 digitalWrite(18,HIGH);
 delayMicroseconds(10);
```

```
 digitalWrite(18,LOW);
 delayMicroseconds(2);

 t=pulseIn(19,HIGH);
 dist=t*340/20000;

 lcd.clear();
 lcd.setCursor(0,1);
 lcd.print("Distance : ");
 lcd.print(dist/100);
 lcd.print(" m");
 delay(1000);
if(dist<40)
{
 digitalWrite(20,HIGH);
 lcd.clear();
 lcd.setCursor(0,1);
 lcd.print("Water level is rising. Kindly evacuate");
 delay(2000);
}
else
{
 digitalWrite(20,LOW);
 delay(2000);
```

```
}
}
```

NODE MCU ESP8266 PROGRAM:

```
#include <ESP8266WiFi.h>
const int trigPin1 = D1;
const int echoPin1 = D2;
#define redled D3
#define grnled D4
#define BUZZER D5 //buzzer pin
unsigned long ch_no = 1053193;//Replace with number
const char * write_api = "1WGTOHK9622G57JI";//Replace with write API
char auth[] = "mwa0000018384149";
char ssid[] = "Alsan Air WiFi 4";
char pass[] = "11122235122@kap1";
unsigned long startMillis;
unsigned long currentMillis;
const unsigned long period = 10000;
WiFiClient  client;
long duration1;
int distance1;
void setup()
{
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
```

```
  pinMode(redled, OUTPUT);
  pinMode(grnled, OUTPUT);
  digitalWrite(redled, LOW);
  digitalWrite(grnled, LOW);
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());
  ThingSpeak.begin(client);
  startMillis = millis();  //initial start time
}
void loop()
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
```

```
distance1 = duration1 * 0.034 / 2;
Serial.println(distance1);
if (distance1 <= 4)
{
  digitalWrite(D3, HIGH);
  tone(BUZZER, 300);
  digitalWrite(D4, LOW);
  delay(1500);
  noTone(BUZZER);
}
else
{
  digitalWrite(D4, HIGH);
  digitalWrite(D3, LOW);
}
currentMillis = millis();
if (currentMillis - startMillis >= period)
{
```