**Smart Connected Sign Boards For Improved Road Safety**

****

Done by: G. Hima Vamsi

T. Likhitha

P. Sharanya

M. Revanth Kumar

**CONTENTS**

TOPIC PAGE

ABSTRACT 1

PROBLEM STATEMENT 2

1. INTRODUCTION 1

1.1 Overview 1

1.2 Aim of the Project 3

1.3 Existing System 3

2. REQUIREMENTS 6

2.1 Software Requirements 7

2.2 Hardware Requirements 7

3. IMPLEMENTATION 9

4. SOURCE CODE 10

5. RESULTS 13

6. CONCLUSION 16

**LIST OF FIGURES**

**Figure Name Page No**

Fig 3(a) Web app 7

Fig 3(b) Functionality of Web app 7

Fig 3(c) IBM Cloud 7

Fig 3(d) IBM Cloud 7

Fig 4(a) Three OLED displays 7

**ABSTRACT**:

The lack of traffic signal striking is often cited as a contributing factor by drivers who are involved in accidents at intersections. As such, increasing the striking of traffic signals should lead to improved safety performance. This document describes a project to determine the road safety effectiveness associated with improved signals with smart connected sign boards. A smart road using IOT devices is a special idea which makes the drivers to drive safer than before.

The first motive of smart connected sign boards is to provide safety, avoid accidents and reduce traffic. This can be implemented by using Nodemcu, Push button, Weather API. Traffic is a growing problem in India causing fuel wastage, time wastage & pollution. In Indian road-traffic, the problems like crowded roads, unpredictable time to travel from one place to another are a serious problems which is also polluted and noisy. Now, researchers have started to introduce smart connected signals technology which is difficult to implement on roads. In this project, we present a low cost innovative technology for smart connected sign boards for improved road safety. Different technologies have been introduced to increase road safety.

**PROBLEM STATEMENT:**

In India we see traffic junction, we can see traffic signals with red, yellow and green lights along with the timer display. To move the traffic has to wait for a fixed interval of time. For example, people have to wait for a small interval of time which is fixed for every signal even though the traffic is more in that particular lane. This leads to a huge “TRAFFIC JAM” which is a major problem now a days the society is facing. Due to this heavy traffic, the number of road accidents are increased which is a major issue. Our project helps to decrease the number of road accidents using smart connected sign boards using Internet Of things (IOT).

**1. INTRODUCTION:**

Nowadays traffic has become a major problem for the people in India. Due to which it causes wastage of precious time, fuel and electricity. The Internet of things (IOT) is the network of electrical appliances, vehicles, physical devices and other items embedded with electronics, actuators, sensors, software, and connectivity which enables all these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

**1.1 Overview:**

In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy traffic or due to accidents then we can change the road signs accordingly if they are digitalized. This project proposes a system which has digital sign boards on which the signs can be changed dynamically.

By using the Weather API we can get the weather reports based on which we can set the speed limit to particular area. If there is rainfall then the roads will be slippery and the speed limit would be decreased. There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly. There are three switches through which you can switch the display to different modes.

**1.2 Aim of the Project:**

Our aim is to develop smart connected sign boards for displaying speed limit, diversions and alerts of accident prone areas. Also to develop sign boards which display information.

**1.3 Existing System:**

The individual traffic signals are connected with traffic control system to perform network wide traffic operation .These control systems contain a central computer, a communication network, and intersection traffic signals. Coordination of control system can be implemented through different techniques like time-base, hardwired interconnection method. Coordination between traffic signals and agencies requires the development of data sharing and traffic signal control agreements. A traffic-signal system has only one purpose i.e. to deliver signal timings to the driver. The system provides features that improve the traffic engineer’s ability to achieve this goal. These are primarily access features. They provide access to the intersection signal controller for maintenance and operations. The more complete and convenient the access, the more efficient the operator will be and the more effective the system. In addition to control the traffic signals, modern technology also provide surveillance capabilities, including different kinds of video surveillance and traffic detection.

**2. REQUIREMENTS:**

The hardware and software requirements of the project are:

**2.1 Software Requirements:**

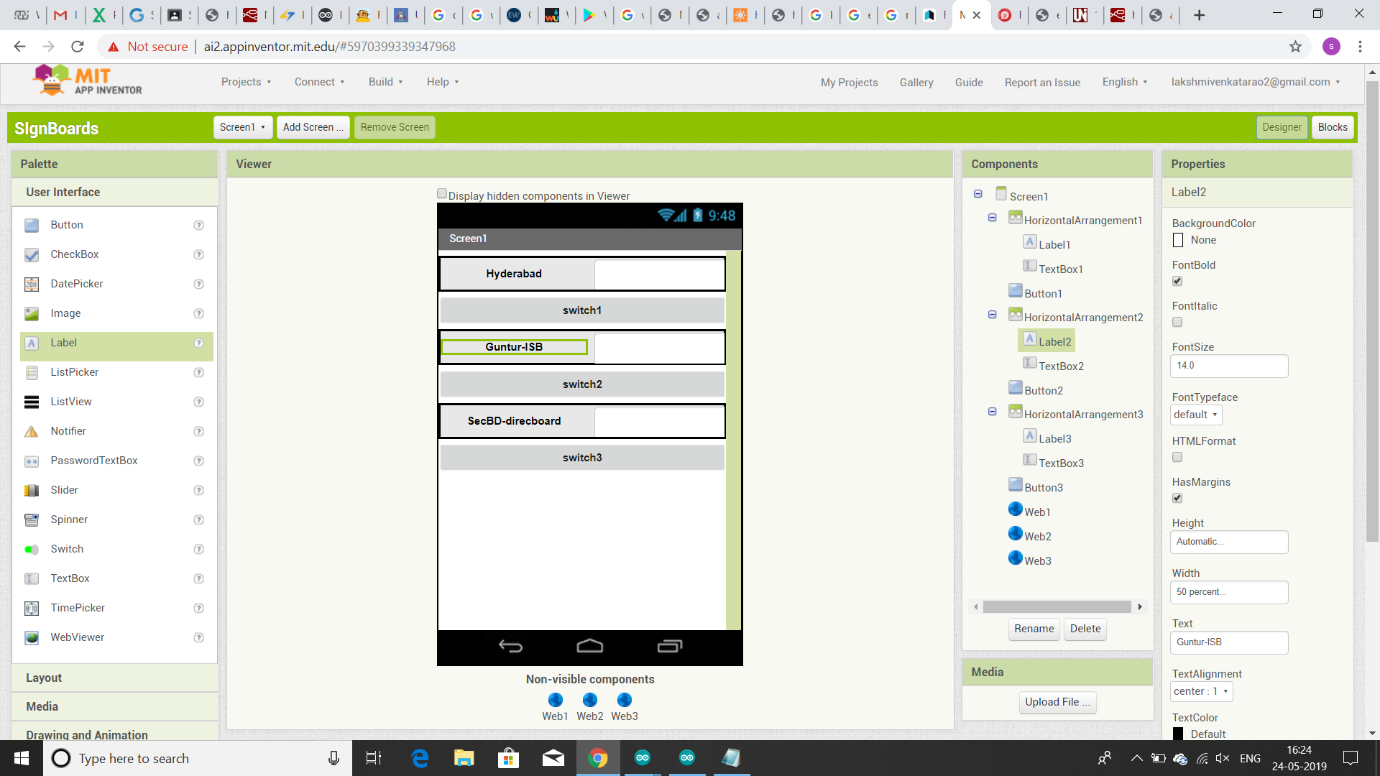
* Arduino IDE
* Weather API
* IBM Cloud Platform

**2.2 Hardware Requirements:**

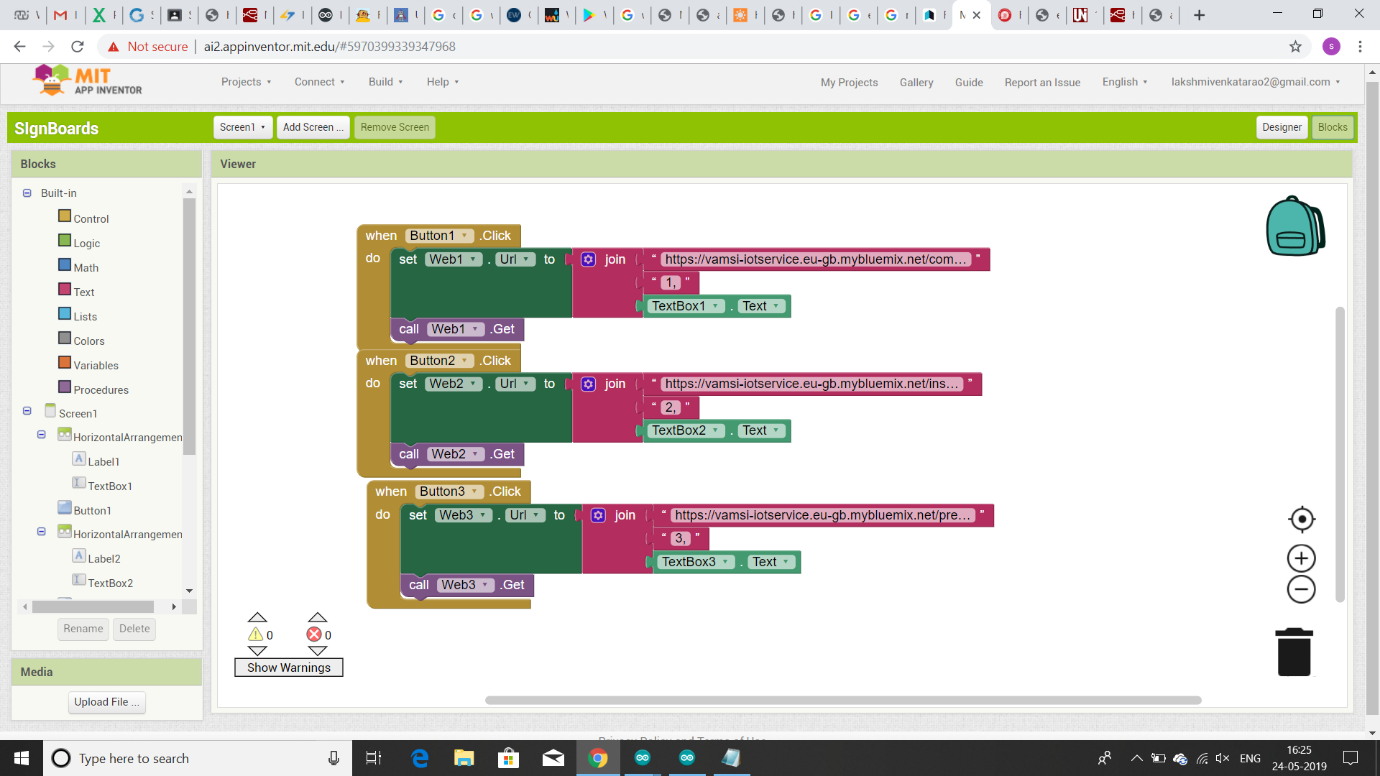
* NODEMCU (12-E)
* PUSH BUTTON
* OLED DISPLAY

**3. IMPLEMENTATION**

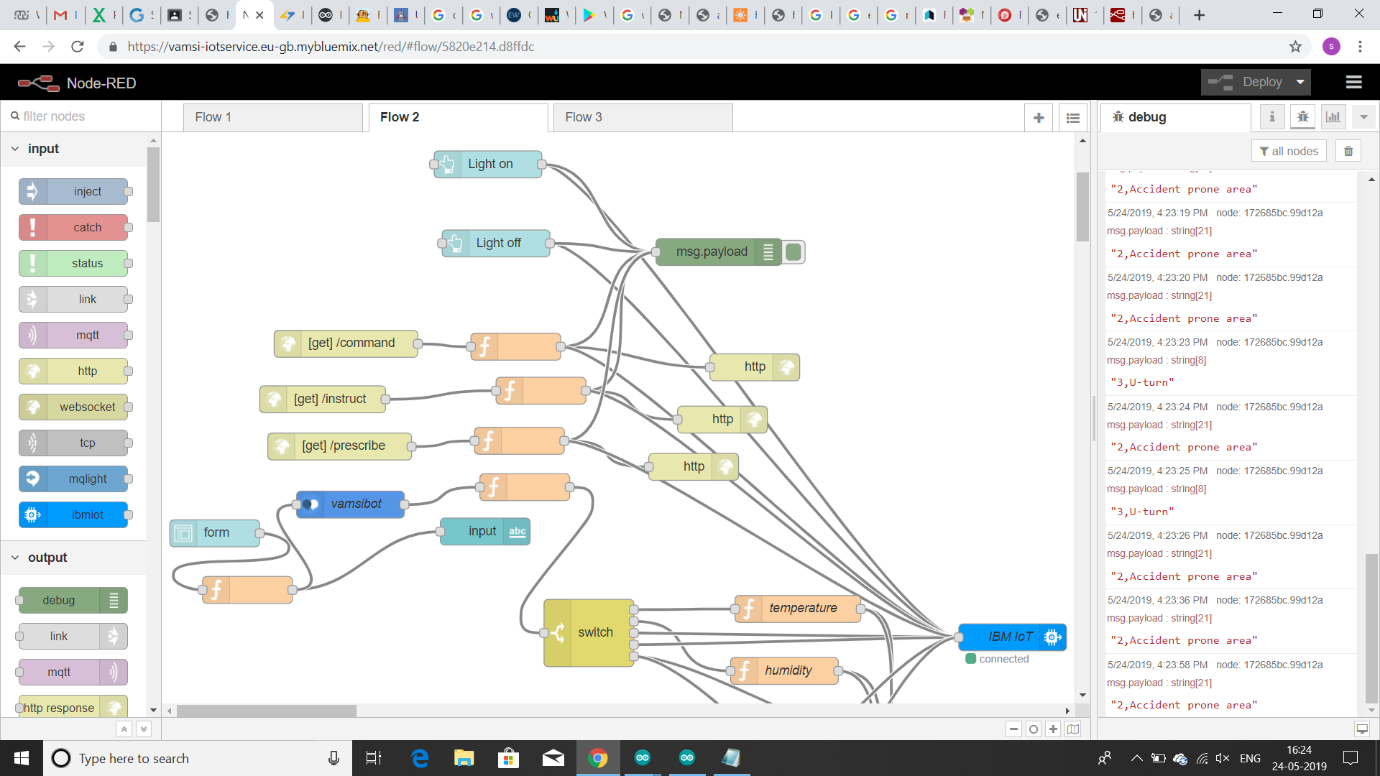
* OLED pins 3V3, GND, SCL, SDA are connected to VCC, GND, D1, D2 of Nodemcu pins respectively.
* We run the programs using Arduino.
* Those programs which are executed using Arduino are uploaded into the Nodemcu using USB cable.
* Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.
* Using MIT app inventor we create a web app which uses a drag and drop procedure .Here we can insert labels, buttons, check boxes, text boxes, etc. for creating an app.
* According to our app we create three text boxes in which we can enter desirable data.
* After entering the data in the text box, there a button present which is to be tapped.
* On click function of the button, the page will be directed to the URL which is of NodeRed (IBM Cloud).
* The data is sent to device (Nodemcu) from cloud.
* The output is displayed on OLED display after receiving data from Nodemcu.

****

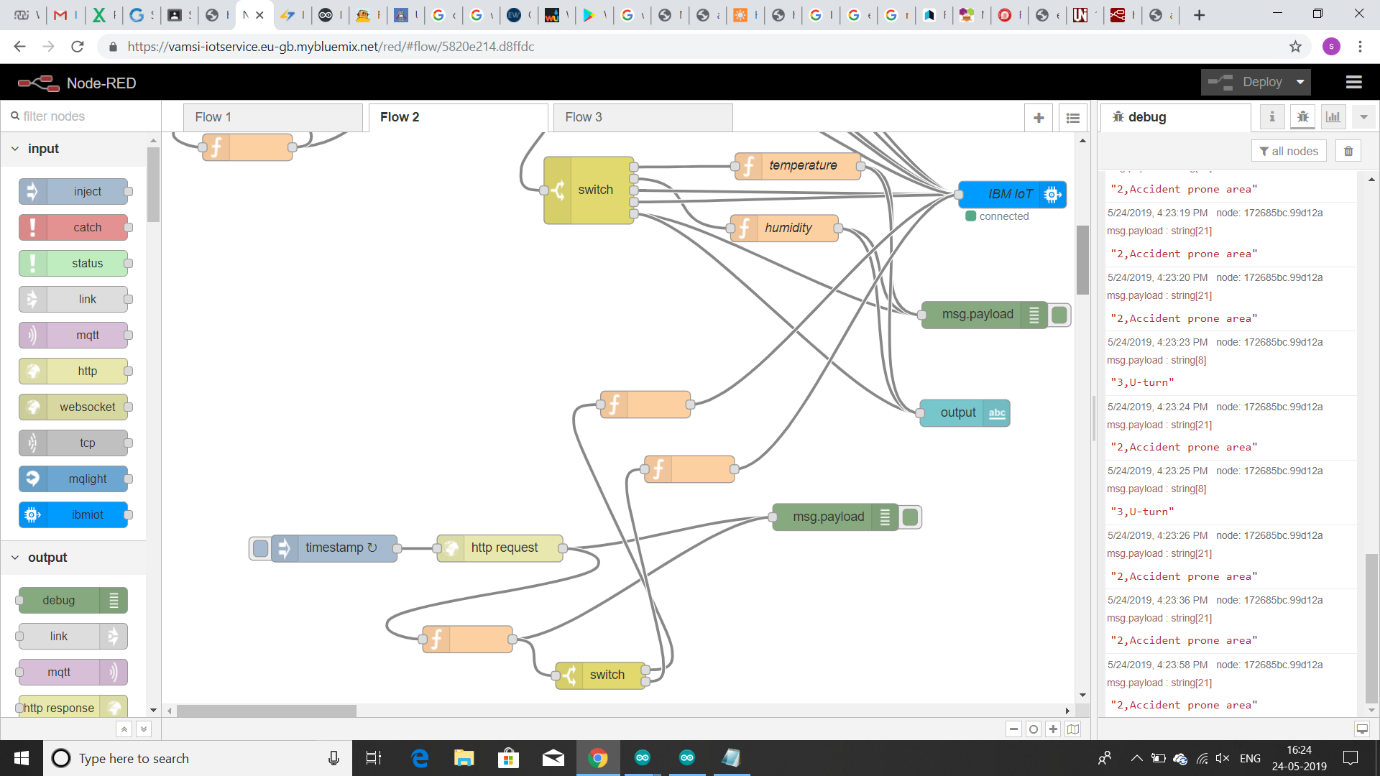
**Fig 3(a)** This figure is about the smart connected sign board web app along with three text boxes, three labels and three buttons for entering the data.

****

**Fig 3(b)** This figure shows the functionality of the on click function of the buttons.

****

**Fig 3(c)** This figure shows the IBM cloud where the URLs are directed.

****

**Fig 3(d)** This figure shows the IBM cloud where the URLs are directed.

**4. PROGRAM CODE:**

**PROGRAM 1:**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "SB-IOT1";

const char\* password = "sb@iot11";

String command1,command2;

#define ORG "bhip5y"

#define DEVICE\_TYPE "Vamsi"

#define DEVICE\_ID "8500"

#define TOKEN "8500913778"

String command;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

//Serial.println(clientID);

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_GFX.h>

#define SSD1306\_LCDHEIGHT 64

// OLED display TWI address

#define OLED\_ADDR 0x3C

Adafruit\_SSD1306 display(-1);

#if (SSD1306\_LCDHEIGHT != 64)

#error("Height incorrect, please fix Adafruit\_SSD1306.h!");

#endif

void callback(char\* topic, byte\* payload, unsigned int payloadLength);

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback, wifiClient);

void setup() {

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

Serial.begin(115200);

Serial.println();

pinMode(D1,OUTPUT);

wifiConnect();

mqttConnect();

}

void loop() {

if (!client.loop()) {

mqttConnect();

}

delay(100);

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) {

Serial.print("callback invoked for topic: "); Serial.println(topic);

for (int i = 0; i < payloadLength; i++) {

//Serial.println((char)payload[i]);

command += (char)payload[i];

}

Serial.println(command);

command1=getValue(command,',',0);

command2=getValue(command,',',1);

if(command1=="1"){

display.clearDisplay();

// display a line of text

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0,10);

display.print(command);

// update display with all of the above graphics

display.display();

}

command ="";

command1 ="";

command2="";

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {

if (data.charAt(i) == separator || i == maxIndex) {

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

**PROGRAM 2:**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "SB-IOT1";

const char\* password = "sb@iot11";

String command1,command2;

#define ORG "bhip5y"

#define DEVICE\_TYPE "Vamsi"

#define DEVICE\_ID "8500"

#define TOKEN "8500913778"

String command;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

//Serial.println(clientID);

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_GFX.h>

#define SSD1306\_LCDHEIGHT 64

// OLED display TWI address

#define OLED\_ADDR 0x3C

Adafruit\_SSD1306 display(-1);

#if (SSD1306\_LCDHEIGHT != 64)

#error("Height incorrect, please fix Adafruit\_SSD1306.h!");

#endif

void callback(char\* topic, byte\* payload, unsigned int payloadLength);

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback, wifiClient);

void setup() {

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

Serial.begin(115200);

Serial.println();

pinMode(D1,OUTPUT);

wifiConnect();

mqttConnect();

}

void loop() {

if (!client.loop()) {

mqttConnect();

}

delay(100);

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) {

Serial.print("callback invoked for topic: "); Serial.println(topic);

for (int i = 0; i < payloadLength; i++) {

//Serial.println((char)payload[i]);

command += (char)payload[i];

}

Serial.println(command);

command1=getValue(command,',',0);

command2=getValue(command,',',1);

if(command1=="2"){

display.clearDisplay();

// display a line of text

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0,10);

display.print(command2);

// update display with all of the above graphics

display.display();

}

command ="";

command1 ="";

command2="";

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {

if (data.charAt(i) == separator || i == maxIndex) {

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

**PROGRAM 3:**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "SB-IOT1";

const char\* password = "sb@iot11";

String command1,command2;

#define ORG "bhip5y"

#define DEVICE\_TYPE "Vamsi"

#define DEVICE\_ID "8500"

#define TOKEN "8500913778"

String command;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

//Serial.println(clientID);

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_GFX.h>

#define SSD1306\_LCDHEIGHT 64

// OLED display TWI address

#define OLED\_ADDR 0x3C

Adafruit\_SSD1306 display(-1);

#if (SSD1306\_LCDHEIGHT != 64)

#error("Height incorrect, please fix Adafruit\_SSD1306.h!");

#endif

void callback(char\* topic, byte\* payload, unsigned int payloadLength);

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback, wifiClient);

void setup() {

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

Serial.begin(115200);

Serial.println();

pinMode(D1,OUTPUT);

wifiConnect();

mqttConnect();

}

void loop() {

if (!client.loop()) {

mqttConnect();

}

delay(100);

}

void wifiConnect() {

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server);

while (!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice() {

if (client.subscribe(topic)) {

Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength) {

Serial.print("callback invoked for topic: "); Serial.println(topic);

for (int i = 0; i < payloadLength; i++) {

//Serial.println((char)payload[i]);

command += (char)payload[i];

}

Serial.println(command);

command1=getValue(command,',',0);

command2=getValue(command,',',1);

if(command1=="3"){

display.clearDisplay();

// display a line of text

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0,10);

display.print(command2);

// update display with all of the above graphics

display.display();

}

command ="";

command1 ="";

command2="";

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {

if (data.charAt(i) == separator || i == maxIndex) {

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

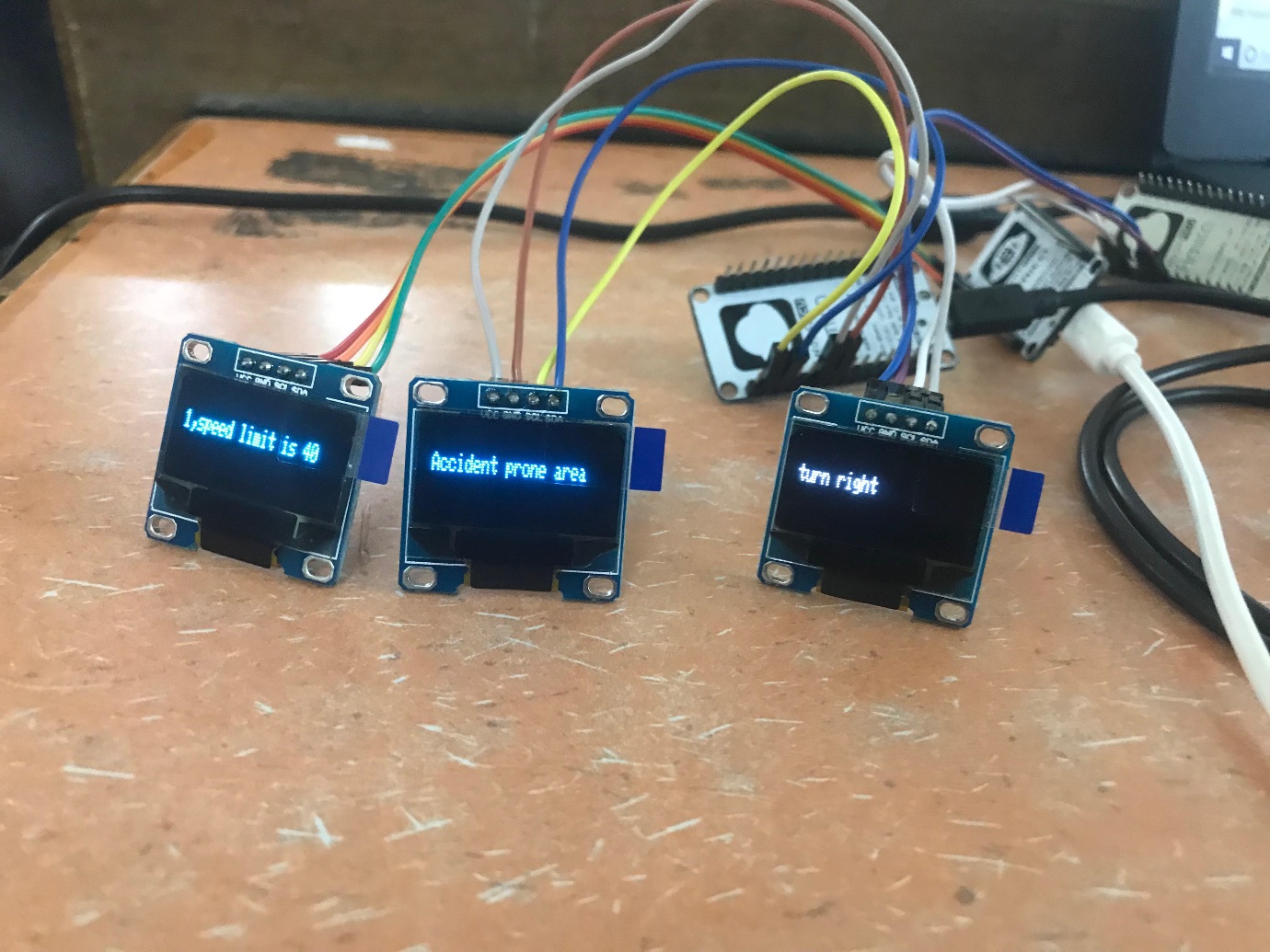
**4. RESULTS:**

The result shows three switches through which you can switch the display to different modes.

Mode1: Displaying Speed Limit

Mode2: Display of Diversions, Alerts of Accident prone area

Mode3: Information sign boards

****

**Fig 4(a)** This figure shows three OLEDs that we are displaying.

(1) The first OLED display shows speed limit using weather API.

(2)The second OLED display shows about the alerts of the accident prone area.

(3)The third OLED display shows the information sign boards

**5. CONCLUSION:**

Before now, roads were merely functional in nature. But now highways are designed that are safe, durable and accessible. The idea that a highway would become a vector for IOT networks or any other communication system was inconceivable and beyond practicality. But recent developments, such as the placement of digital sign boards along roadsides, have created a gateway that makes it possible for roads to act as data conveyors. Sign boards are now displaying data such as road conditions and traffic patterns. Wireless networks can springboard off sensor technology to provide more in-depth communications at higher levels. State and local departments of transportation could use IOT systems to pinpoint road maintenance needs, traffic usage, weather conditions and accident statistics.