



# **IOT BASED SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY**

A PROJECT REPORT

*Submitted by*

<b>ARULKUMAR.R</b>	<b>(712819106709)</b>
<b>SOUNDERRAJAN.R</b>	<b>(712819106707)</b>
<b>AKSHAY.K</b>	<b>(712819106708)</b>
<b>PRAVEEN.M</b>	<b>(712819106302)</b>
<b>VASANTHARAJ.V</b>	<b>(712819106303)</b>

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**RVS COLLEGE OF ENGINEERING AND TECHNOLOGY,**

**COIMBATORE**

**ANNA UNIVERSITY:CHENNAI 600 025**

**NOVEMBER 2022**

## 1. INTRODUCTION

### 1.1 Project Overview

The goal of this project is to replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones.

### 1.2 Purpose

- To replace the static signboards, smart connected sign boards are used.
- These smart connected sign boards get the speed limitations from a web app using weather API and update automatically.
- Based on the weather changes the speed may increase or decrease.
- Traffic diversion signs are remotely controlled using APIs.
- **"DO NOT HONK"** message displayed at School and Hospital Zones which can we set using buttons.

## **2. Literature survey**

### **2.1 Existing problem**

- Rain makes brakes inefficient and leads to accidents
- Fog reduces visibility and increases the probability of accidents
- Traffic diversion requires human intervention

### **2.2 References**

# **Wireless Digital Traffic Signs of the Future**

**Authors :**

**Chai k.toh pietro**

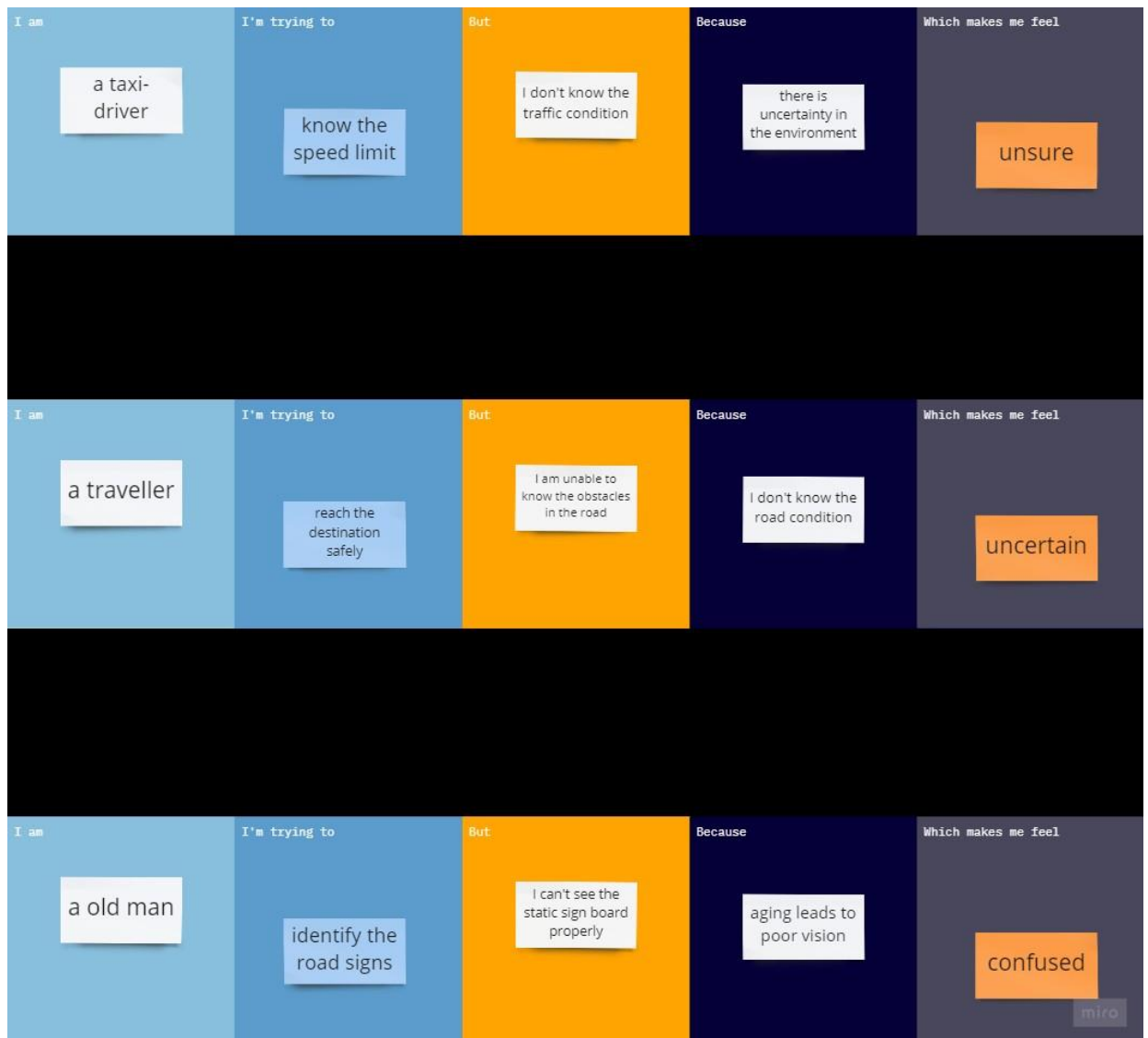
**Manzoni**

**Carlos tavares calafate**

Traffic signs have come a long way since the first automobile was invented. They have long served the purpose of warning and guiding drivers and also enforcing the traffic laws governing speed, parking, turns, and stopping. In this study, the authors discuss the issues and challenges facing current traffic signs, and how it will evolve into a next-generation traffic sign architecture using advanced wireless communications technologies. With technological advances in the areas of wireless communications and embedded electronics and software, we foresee that, in the future, digital traffic sign posts will be capable of transmitting the traffic sign information wirelessly to road users, and this will transform our roads into intelligent roads, where signs will appear promptly and automatically on in-vehicle displays to alert the driver. There is no longer the need to watch out for traffic signs since the detection will be automatic and performed wirelessly. This transformation will lessen the burden on the drivers, so that they can then focus more on the traffic ahead while driving. Also, this evolution into wireless digital sign posts will fit well with the vision of future smart cities, where smart transportation technologies will be present to transform how we drive and commute, yielding greater safety, ease, and assistance to drivers.

## 2.3 Problem statement definition

To replace the static signboards with smart connected sign boards to get the speed limitations from a web app using weather API and update it automatically based on the weather conditions, set diversions through API and warn drivers for school zones and hospital zones.




### 3. Ideation and proposed solution

#### 3.1 Empathy map canvas







### 3.2 Ideation & brainstorming



## Brainstorm & idea prioritization


Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

 10 minutes to prepare  
 1 hour to collaborate  
 2-8 people recommended



#### Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

 10 minutes

---

A

#### Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

#### Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

C

#### Learn how to use the facilitation tools


Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

1

#### Define your problem statement


What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

 5 minutes

---


PROBLEM


How might we [your problem statement]?





#### Key rules of brainstorming


To run an smooth and productive session


 Stay in topic.

 Encourage wild ideas.

 Defer judgment.

 Listen to others.

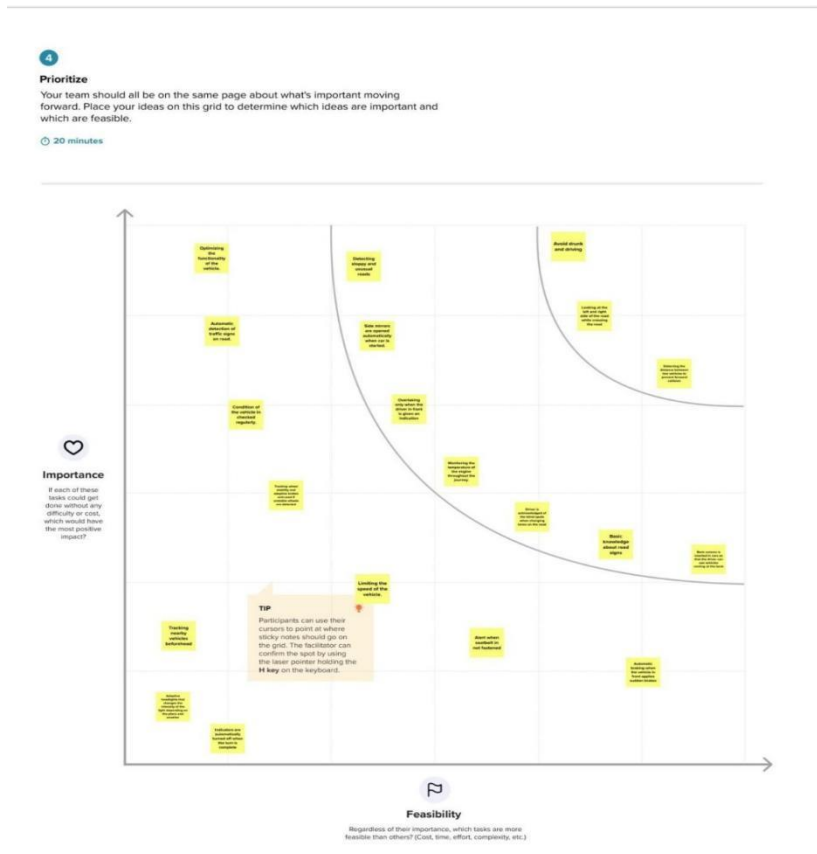
 Go for volume.

 If possible, be visual.

## SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

[illegible]

## Idea Prioritization



### 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> <li>To prevent the road accidents from happening using IOT.</li> </ul>
2.	Idea / Solution description	<ul style="list-style-type: none"> <li>By Preparing smart signs using IOT instead of regular signs hung on the road.</li> <li>Smart signs are built with IOT and LED are used.</li> </ul>

### SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY



3.	Novelty / Uniqueness	<ul style="list-style-type: none"><li>• Since LED'S are used which is visible from after.</li><li>• The smart signs consists of temperature, humidity, wind speed.</li><li>• These information are received from weather monitoring app.</li><li>• It also gives information about nearby places such as hospitals,schools,etc....,so that the users can decide their speeding according to that information.</li></ul>
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"><li>• These create a noticable impact on the road safety department.</li><li>• By deciding a speed limit for the user,there is significant chance in reducing the accidents.</li></ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"><li>• By executing these for commoners by the government, it is great initiative in creating a awareness among the people.</li><li>• A separate budget can be allotted for this By the government, which paves a way for a safer environment.</li></ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"><li>• It has greater chance in reducing the risk for the people as it is more visible than the normal signs, which saves a lot of lives at stake.</li></ul>

### 3.4 Problem Solution Fit

- The display replaces the static signs
- Processing requirement of microcontroller is reduced since all the processing is done in the cloud servers.
- Direction can be remotely set by the concerned authorities without needing to personally attend the site.

**Problem-Solution fit canvas 2.0**

Signs with Smart Connectivity for Better Road Safety

TEAM ID - PNT2022TMID42985

<b>1. CUSTOMER SEGMENT(S)</b> Who is your customer? <input type="checkbox"/> highway division	<b>6. CUSTOMER CONSTRAINTS</b> What constraints prevent your customers from taking action or limit their choices of solutions? The impact of the network on the tests is a significant and unexpected element. Given the quantity of sensors, this IoT-based system is successful in simulating a large-scale smart agricultural setting.	<b>5. AVAILABLE SOLUTIONS</b> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pre-d. can do the customer have? Along roadways, static signs with clear directions are put as potential fixes which gives clear solution.
<b>3. JOB TO-BE-DONE / PROBLEMS</b> Which jobs-to-be-done (or problems) do you address for your customers? There may different duties, the Smartboard Connectivity is in charge of taking correct temperature sensor readings and should inform the board of the speed of the customer's vehicle.	<b>8. PROBLEM ROOT CAUSE</b> What is the real reason that this <u>problem exists</u> ? <u>Why?</u> If there was no internet connection, no sensor readings from the weather would enter the speed restriction. Unnecessary pressing of the accident indicator button by anyone could lead to problems.	<b>7. BEHAVIOUR</b> What does your customer do to address the problem and get the job done? As a teacher, the IoT cloud updates the smart board on the condition of the roads on a regular basis so that the customer would address the problem and get the job done.
<b>2. TRIGGER</b> What triggers customers to act? i.e. seeing their neighbor installing. Weather will be bad most of the time. The car bought to be travelling at its threshold speed. To alert the customer, the sensor value should be shown on the smart board.	<b>10. YOUR SOLUTION</b> We employ smart limited sign boards as an alternative to static signboards. With the help of a web app and weather API, these intelligent limited sign boards automatically connect with the current speed limits. The speed may rise or fall in response to variations in the weather. The display of diversion signs is determined by traffic and potentially fatal situations. As appropriate, there are also signs that read "Guide (Schools), Warning, and Service" (Hospitals, Restaurants). Using buttons, it's possible to choose from safety of opening modes.	<b>8. CHANNELS of BEHAVIOUR</b> What kind of actions do customers take online? The departments can receive direct emails or messages from customers. (Office on nearby side). <b>3.2 What kind of actions do customers take offline?</b> Following directions is one of the major tasks for the traveler, but they can utilize the smartboard signs to check the state of the road from wherever they are standing.
<b>4. EMOTIONS: BEFORE / AFTER</b> How do customers feel when they face a problem or a job and afterwards? Clients will feel better after selecting an operation mode with the use of smartboard connectivity, and they will then follow the instructions on the smartboard.		

## 4. Requirement Analysis 4.1

### Functional

#### Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User visibility	<ul style="list-style-type: none"><li>• Informations can be written in short form in the sign boards so that it can be very easily captured by drivers.</li><li>• Place sign boards on popular places.</li><li>• Symbols can be used so that drivers can save some amount of time in reading.</li><li>• Static signs can be replaced by smart signs to reduce accidents.</li></ul>
FR-2	User convenience	<ul style="list-style-type: none"><li>• Display should be larger which can be visible from far distance.</li></ul>
FR-3	User need	<ul style="list-style-type: none"><li>• Awareness programmes should be conducted to bring awareness among the users about road safety.</li><li>• Road safety education is essential for users.</li></ul>

#### 4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
--------	----------------------------	-------------

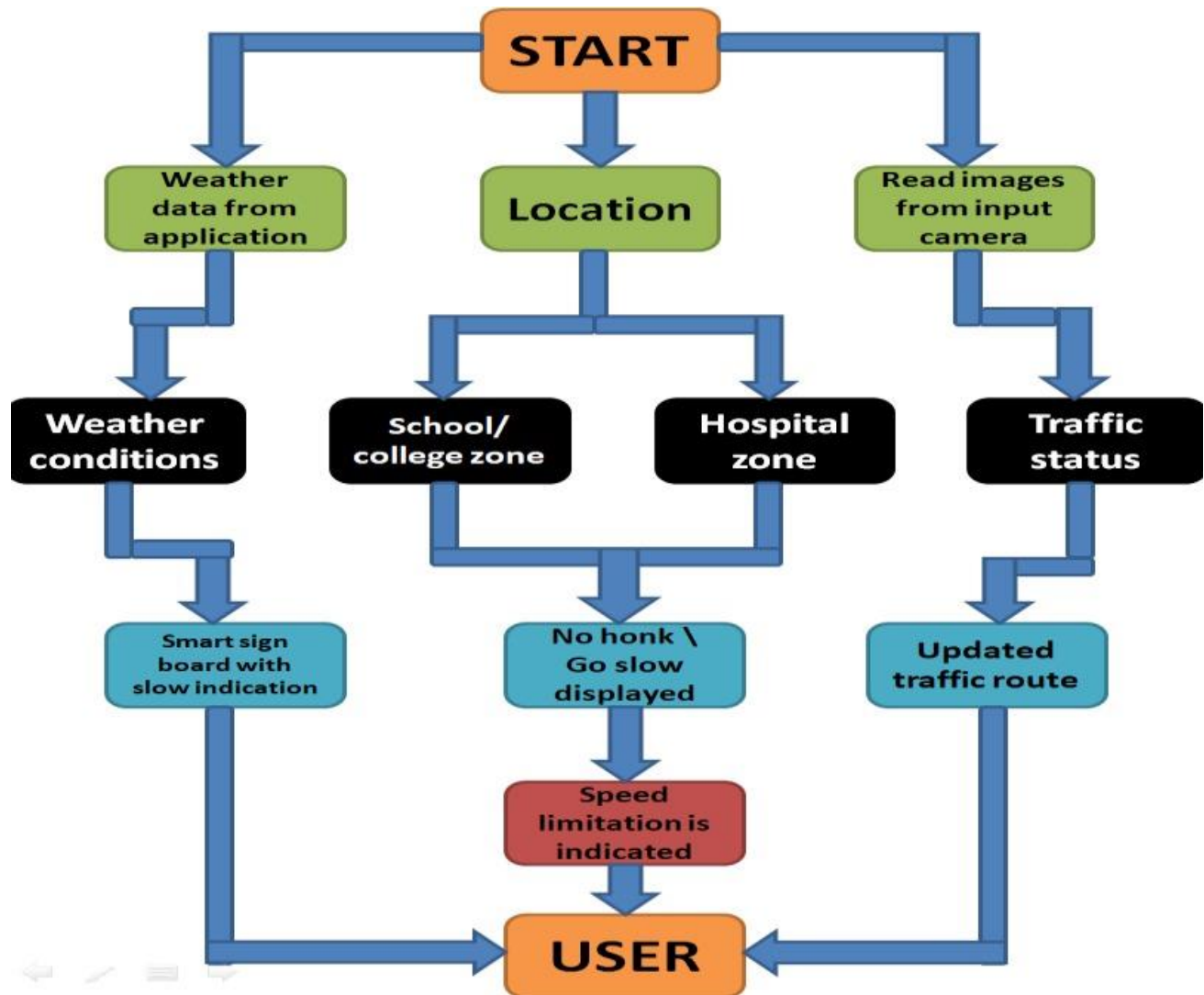
NFR-1	<b>Usability</b>	<ul style="list-style-type: none"><li>• When crowd on accident occur it guides thetravellers to choose best path.</li><li>• Intimates the speed range depending uponroadway condition.</li><li>• Ensure the vehicles are redirected to right path without causing much trouble for otherdrivers.</li><li>• Easy to follow instructions based on givendata on the digital board.</li></ul>
-------	------------------	---

NFR-2	<b>Security</b>	<ul style="list-style-type: none"><li>• Prediction of data gives them a fair and better road understanding about their upcoming of toad events.</li></ul>
NFR-3	<b>Reliability</b>	<ul style="list-style-type: none"><li>• Helps to travellers behaviour towards awareness of travel.</li></ul>

NFR-4	<b>Performance</b>	<ul style="list-style-type: none"><li>• Pre-functional record of voice record alongwith LED display provide in waiting time at traffic signs.</li><li>• There is a rain drop sensor which checks whether there is a rain, to transmit data over IOT helps to display on LED along with wifi connection of internet changing data dynamically with current reporting of event sensing flow of data</li></ul>
NFR-5	<b>Availability</b>	<ul style="list-style-type: none"><li>• Monitors the road events even in low light on poor weather conditions.</li><li>• Record traffic offenses</li></ul>
NFR-6	<b>Scalability</b>	<ul style="list-style-type: none"><li>• It is user friendly interface.</li><li>• Data accessibility is easy from source.</li></ul>

## 5. Project Design

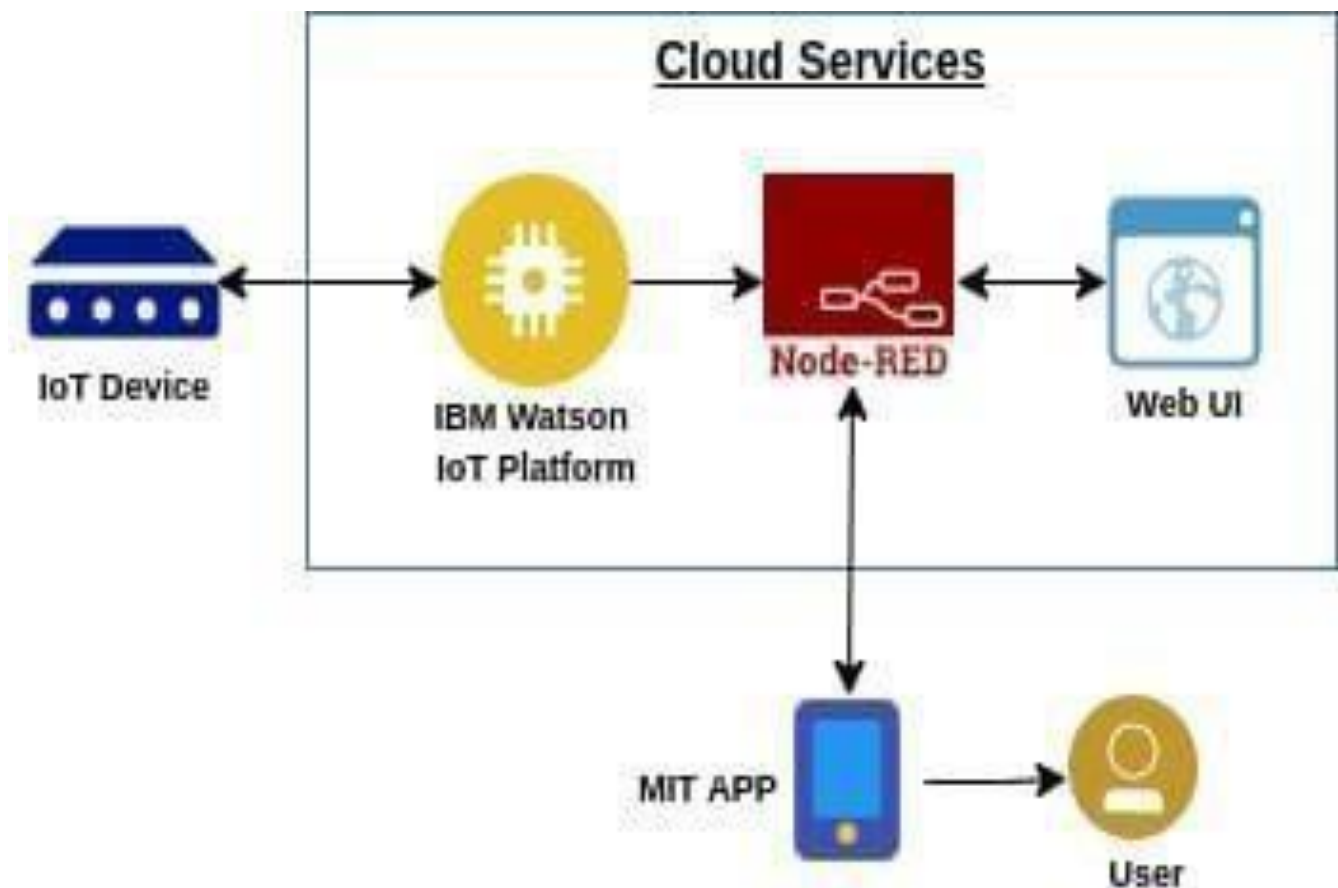
### 5.1 DATA FLOW DIAGRAMS



## 5.2 Solution & Technical Architecture

Solution architecture and Technical architecture is a complex process – with many sub- processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



## 5.3 User Stories

SCENARIO Browsing, booking, tending, and rating a local city tour	Entice How does someone initially become aware of this process?	Enter What do people experience as they begin the process?	Engage In the core moments in the process, what happens?	Exit What do people typically experience as the process finishes?	Extend What happens after the experience is over?
<b>Steps</b> What does the person (or group) typically experience?	Customer sees available traffic signs and data Customer must have GPS activate about driving	They were get info to follow a new the public direction towards safety Get pre-idea of road infrastructure	Real digital activation of road while users can see map of surrounding area Provide a collection of data about traffic, drive Speed limits also be submitted	People have information with regard to published instruction Less than power and automatic updates are displayed	Record traffic, offense & provide warning data
<b>Interactions</b> What interactions do they have at each step along the way? People: Who do they see or talk to? Places: Where are they? Things: What digital touchpoints or physical objects would they use?	Interaction with digital sign board Sign boards placed mostly on demand areas	Features include access about the traffic People will get updated with the traffic situation	Provide pre-functional record of specific road mode Collecting information displayed on LED display	Dynamic sign board under road - also provide planning via the computer Updates are done with the map given & not updates through internet	Maintaining road status & displayed even in sign & low light area Provides flexible services along long distances
<b>Goals &amp; motivations</b> At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")	Helps to know road condition is safe or not Helps to know the speed limit and other option if any	Make a communication feel as possible It helps the customer to reach as low to knowing money condition	Customer will be able to know their journey with more updates on traveling	To provide a communication strategy in the smart city manner Helps to reach the destination safely on that time	To provide a comparison from traffic management
<b>Positive moments</b> What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	Get new experience of smart city technology Visual effect makes highly interaction	Make a travel powerful	Prediction of data given from a user's understanding map based on location & time of speed	Flow of updates is quick and easy	Getting more information Make good attention into environment
<b>Negative moments</b> What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	It is difficult to understand, who are concerned of signs	Accessing of information takes some time	Lack of signs concerning the navigation, which confusing mode	Maintenance risky There is chance of error in the transfer of data	Is it truly worthy?
<b>Areas of opportunity</b> How might we make each step better? What ideas do we have? What have others suggested?	Helps to transfer behavior towards the awareness of travel	Making the travel easy through the idea notification	Improve the road safety measures	Does it useful to understanding the road user behaviour & flow of traffic	To alert the drivers to reduce delay & improve while travelling time

## 6. Project Planning And Scheduling Phase

### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story/Task	Story Points	Priority
Sprint-1	Intializing the Resources	Create an account in Open Weather API	1	LOW



Sprint-1	Code in Software is written	Write a python script  using the inputs given from OpenWeather  API	2	MEDIUM
Sprint-2	Sending the  software to cloud	The python code from  sprint 1 should be sent to cloud so that it is easily accessible	1	MEDIUM
Sprint-3	Initialising the connection between hardware and cloud	The hardware should be intergrated for the  Easy access of the cloud functions	2	HIGH
Sprint-4	User input-output optimisation and error identification and rectification	Rectify all the shortcomings/errors and initiate the optimisation for better	3	HIGH

## 6.2 Sprint Delivery Schedule

TITLE	DESCRIPTION	STATUS
<b>Literature Survey &amp; Information gathering</b>	A literature review is a comprehensive summary of previous researches on the topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research.	<b>Completed</b>
<b>Prepare Empathy Map</b>	An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. It helps us to understand the customer's difficulties from their point of view.	<b>Completed</b>
<b>Ideation- Brainstorming</b>	Brainstorming is a group problem-solving method that helped us to gather and organize various ideas and thoughts from team members.	<b>Completed</b>

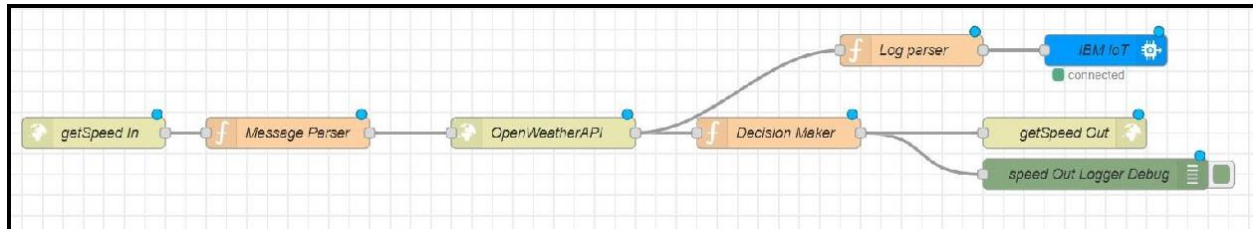
<b>Define Problem statement</b>	<b>The problem statement helps us to focus on what matters to create experiences the people. This allowed us to find the ideal solution for the challenges.</b>	<b>Completed</b>
<b>Problem Solution Fit</b>	<b>It helped us understand and analyze all the thoughts of our customers, their choice of options, problems, behavior and emotions.</b>	<b>Completed</b>
<b>Proposed solution</b>	<b>It helped us analyze and examine our solution more in the grounds of uniqueness, social impact, business model, scalability etc.</b>	<b>Completed</b>
<b>Solution Architecture</b>	<b>Solution architecture is a complex process with many sub-processes that bridges the gap between business problems and technology solutions. It helped us understand the features and components used to complete the project.</b>	<b>Completed</b>
<b>Customer journey map</b>	<b>It helped to analyze the various steps, interactions, goals and motivation, positives, negatives and opportunities.</b>	<b>Completed</b>

<b>Solution requirements</b>	<b>It briefs about functional and non-functional requirements. It involves the various steps in the entire process. It also specifies features usability, security, reliability, performance, availability and scalability.</b>	<b>Completed</b>
<b>Technology stack</b>	<b>A tech stack is the combination of technologies a company uses to build and run an application or project. It helps us analyze and understand various technologies that needs to be implemented in the project.</b>	<b>Completed</b>
<b>Data Flow</b>	<b>A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enter and leave the system, what changes their formation, and where data is stored.</b>	<b>Completed</b>

<b>Sprint Delivery plan</b>	<b>Sprint Planning is an event in scrum that defines what can be delivered in the upcoming sprint and how that work will be achieved. It helps us to organize and complete the work effectively and efficiently.</b>	<b>Completed</b>
<b>Prepare milestone and activity list</b>	<b>Helps us understand and evaluate our progress and accuracy so far.</b>	<b>Completed</b>
<b>Project Development - Delivery of Sprint-1,2,3,4</b>	<b>Develop and submit the developed code</b>	<b>Completed</b>

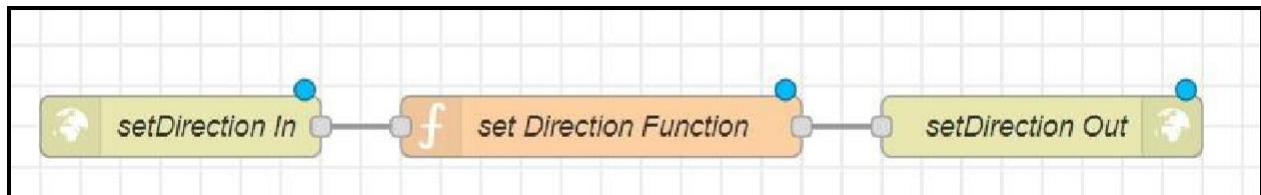
## 7. Coding & Solutioning

### 7.1 Feature 1 - Get Speed For Given Location & Climate



This part of Node RED flow accepts an http GET end point at **"/getSpeed"** from which the location, uid, hospital/school zone info are passed. Message parser sets the required APIKEY for OpenWeatherAPI for the next block. This data is then passed onto Decision Maker which makes all the decisions regarding the message to be output at the display and sends it as a http response. This data is displayed at the microcontroller. Thus a lot of battery is saved due to lesser processing time.

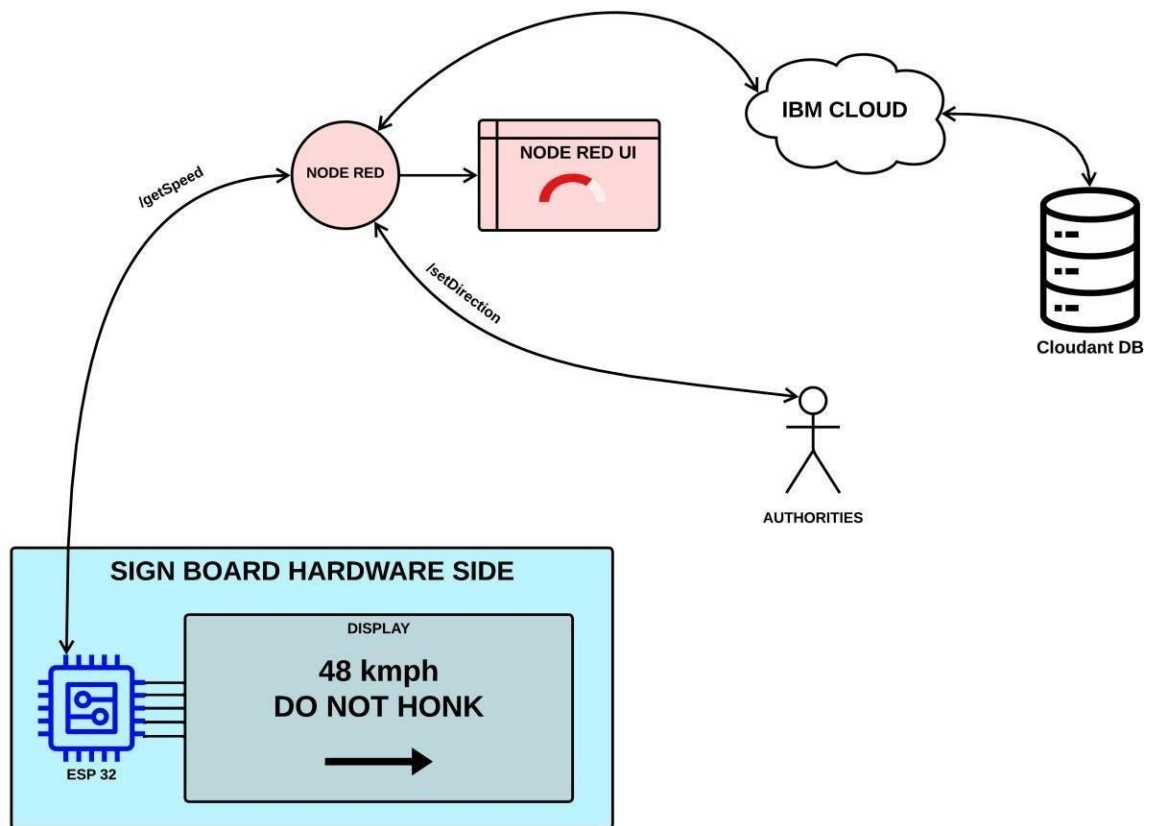
### 7.2 Feature 2 - Set Direction Remotely For A Given Sign Board



This part of Node RED flow accepts an http GET end point at **"/setDirection"** from which the uid and direction information are passed by the respective authorities. Set Direction Function block adds the direction information to the database and returns the same as an http response. This data is sent to the microcontroller along with the **"/getSpeed"** path and the microcontroller displays it.

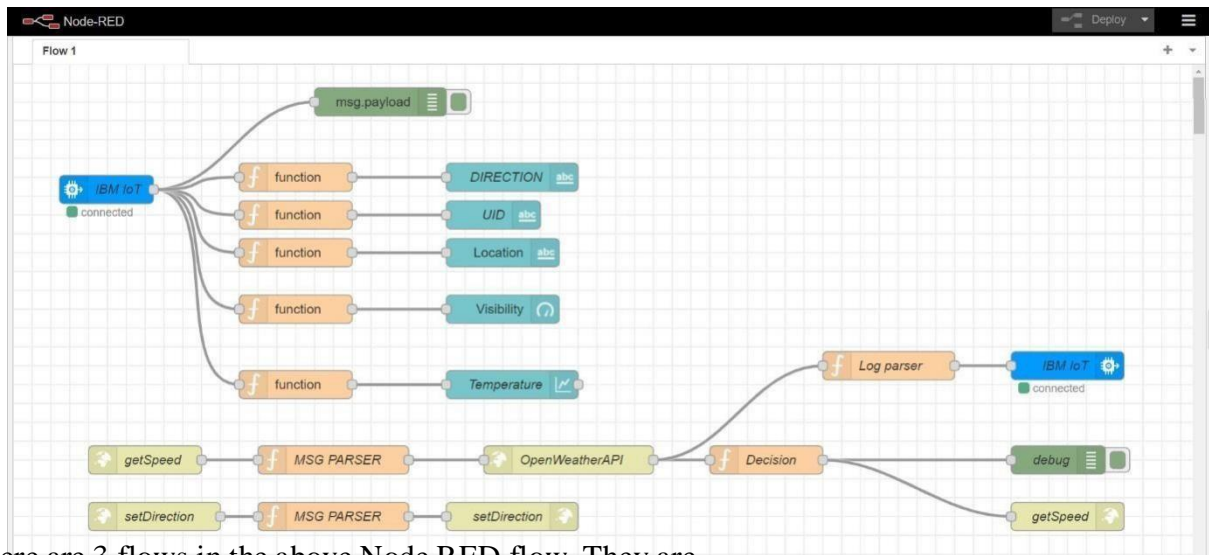
A detailed documentation of all the workflows is available at the following :

**Flow :**



## Node Red :

### Node RED flow :

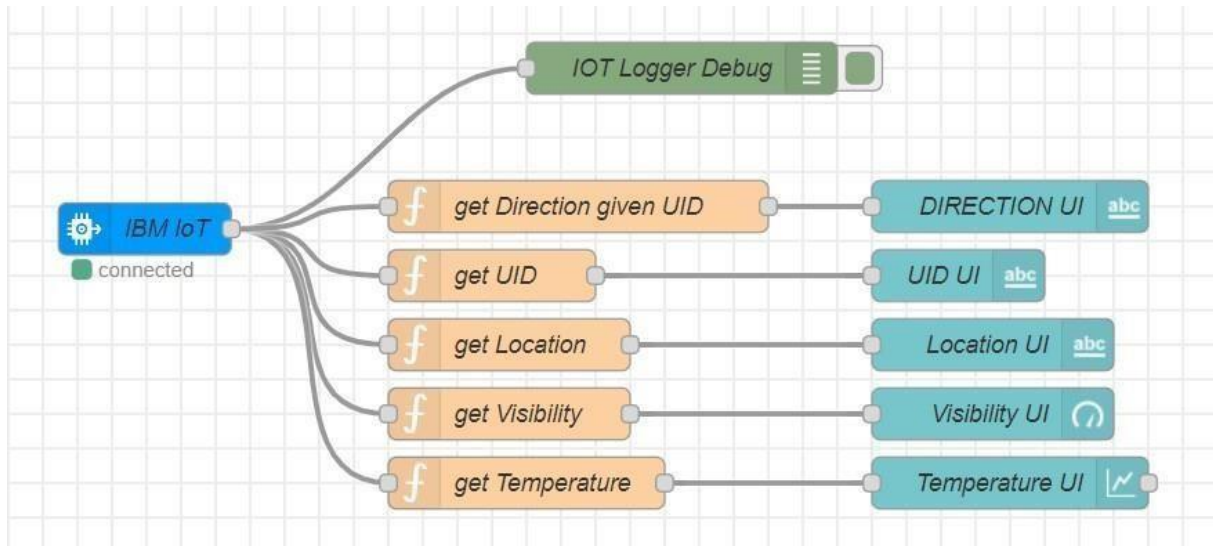


There are 3 flows in the above Node RED flow. They are

1. Node RED UI flow
2. /getSpeed API flow
3. /setDirection API flow 1.

### Node Red UI flow :





1. "IBM IOT" node connects the backend to Node RED UI

2. The function nodes such as "**get Direction given UID**", "**get UID**", "**get Location**", "**get Visibility**" & "**get Temperature**" extract the respective data out and provides them to the UI nodes "**Direction UI**", "**UID UI**", "**Location UI**", "**Visibility UI**" & "**Temperature UI**".

```
// get Direction given UID

msg.payload = global.get(String(msg.payload.uid)); return
msg;

// get UID

msg.payload = msg.payload.uid; return
msg;

// get Location

msg.payload = msg.payload.location; return
msg;

// get Visibility

msg.payload = msg.payload.visibility;
returnmsg;

// get Temperature

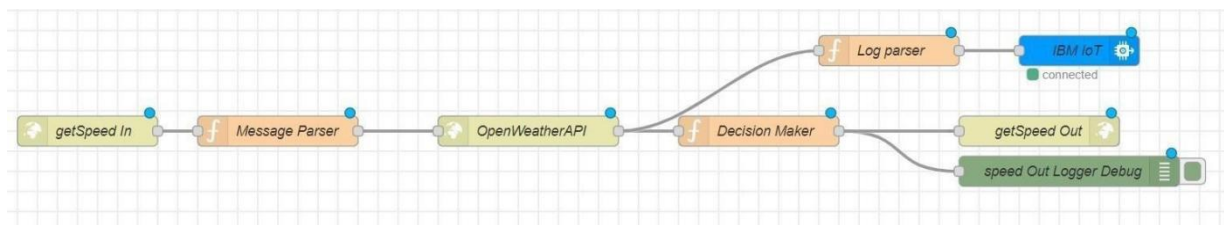
msg.payload = msg.payload.temperature; return msg;
```

3. "**IOT Logger Debug**" node logs the data at debugger.

---

**2.**

**/getSpeed API flow :**



1. **"getSpeed In"** node is an http end point. It accepts parameters like microcontroller UID, location, school & hospital zones info.
2. **"Message Parser"** node parses the data and passes on only required information to the next node

```
global.set("data",msg.payload);
```

```
msg.payload.q =
```

```
msg.payload.location;
```

```
msg.payload.appid = "bf4a8d480ee05c00952bf65b78ae826b"; returnmsg;
```

3. **"OpenWeatherAPI"** node is a http request node which calls the OpenWeatherAPI and send the data to the next node.
4. **"Log Parser"** node extracts specific parameters from the weather data and andsends it to the next node.

```
weatherObj = JSON.parse(JSON.stringify(msg.payload)); localityObj
```

```
= global.get("data");
```

```
var suggestedSpeedPercentage = 100;
```

```
var preciseObject = {
```

```
temperature          :
```

```
weatherObj.main.temp
```

```
- 273.15,    location :
```

```
localityObj.location,
```

```
visibility
```

```
: weatherObj.visibility/100,   uid :
  localityObj.uid,direction :
  global.get("direction")
};
```

```
msg.payload = preciseObject; return
msg;
```

5. **"IBM IoT"** node here (IBM IoT OUT) connects the **"IBM IoT"** node (IBM IoT IN) mentioned in the **Node RED UI flow** which enables UI updation and logging.

6. **"Decision Maker"** node processes the weather data and other information from the micro controller to form the string that is to be displayed at the SignBoard

```
weatherObj = JSON.parse(JSON.stringify(msg.payload)); localityObj
```

```
= global.get("data");
```

```
var suggestedSpeedPercentage = 100;
```

```
var preciseObject = {
```

```
  temperature : weatherObj.main.temp - 273.15,
```

```
  weather : weatherObj.weather.map(x=>x.id).filter(code => code<700),
```

```
  visibility : weatherObj.visibility/100
```

```
};
```

```
  if(preciseObject.visibility<=4
```

```
0)suggestedSpeedPercentage -=30
```

```
  switch(String(preciseObject.weather)[-1]) // https://openweathermap.org/weatherconditions
  refer weather codes meaning here
```

```
{
```

```
  case "0" : suggestedSpeedPercentage -=10;break;
```

```

case "1" : suggestedSpeedPercentage -=20;break;    case
"2" : suggestedSpeedPercentage -=30;break;
} msg.payload = preciseObject;
var doNotHonk = 0;
if(localityObj.hospitalZone=="1"||localityObj.schoolZone=="1")
doNotHonk = 1;
var responseObject = {

    suggestedSpeed : localityObj.usualSpeedLimit*(suggestedSpeedPercentage/100),
doNotHonk : doNotHonk
} msg.payload = String(responseObject.suggestedSpeed) + " kmph \n\n" +
(responseObject.doNotHonk==1?"Do Not Honk":"" ) + "$" +
global.get(String(localityObj.uid));return msg;

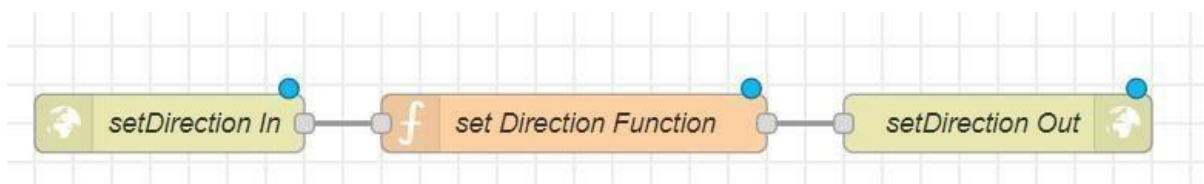
```

7. **"getSpeed Out"** node returns a http response for the request at node **"getSpeed In"**.
8. **"speed Out Logger Debug"** logs the data for debugging.

---

**3.**

### **/setDirection API flow :**



1. **"setDirection In"** node is an http end point. It accepts parameters like microcontroller UID & direction.

2. "set Direction Function" node sets the direction for the given UID.

```
global.set(String(msg.payload.uid),msg.payload.dir); returnmsg;
```

3. "setDirection Out" node returns a http response for the request at node "setDirection In".

[Click on this link to change direction to Straight](#)

[Click on this link to change direction to Left](#)

[Click on this link to change direction to Right](#)

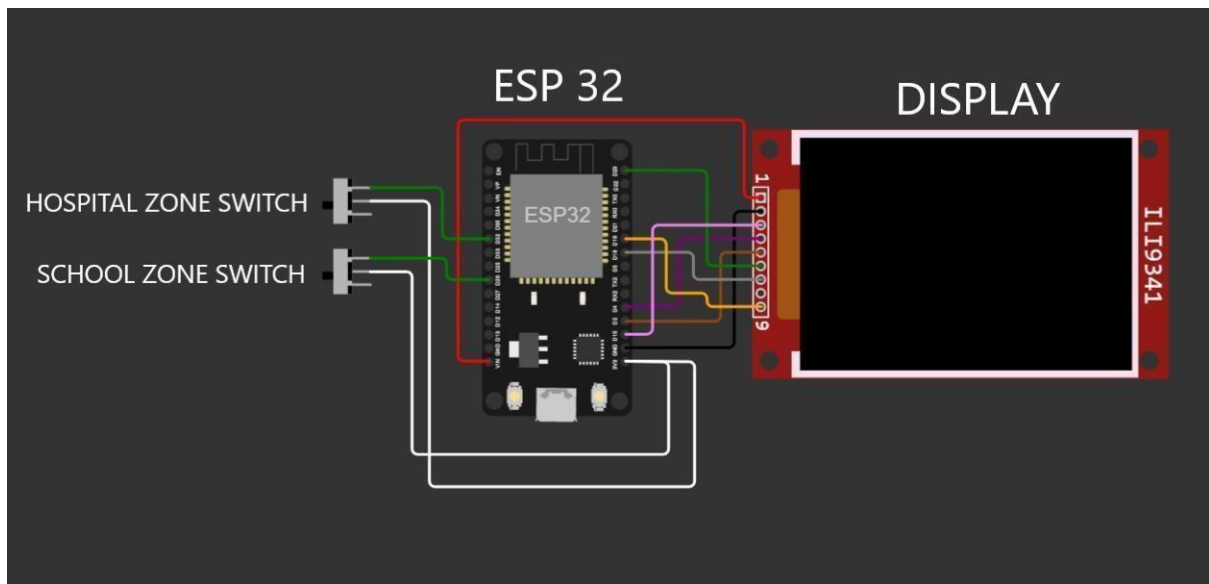
---

## Wokwi Circuit :

[Wokwi Code](#)

[Wokwi Link](#)

## Circuit Diagram :



## ESP 32 CODE :

```
#include <WiFi.h>

#include <HTTPClient.h>

#include

<Adafruit_GFX.h>

#include <Adafruit_ILI9341.h>

#include <string.h>      const
char* ssid = "Wokwi-

GUEST"; const char* password

="";

#define TFT_DC 2

#define TFT_CS 15

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

String myLocation = "Chennai,IN"; String
usualSpeedLimit = "70"; // kmph      int
schoolZone = 32;

int hospitalZone = 26;

int uid = 2504; // ID Unique to this Micro Contoller

String getString(char x)

{
    String s(1, x);  return s;
}

String stringSplitter1(String fullString,char delimiter='$')

{
```

```
String returnString = "";      for(int i = 0;
i<fullString.length();i++) {
char c = fullString[i];    if(delimiter==c)    break;
    returnString+=String(c);
}

return(returnString);

}
```

```
String stringSplitter2(String fullString,char delimiter='$')
```

```
{

String returnString = "";  bool flag =
false;for(int i = 0;
i<fullString.length();i++) {
char c = fullString[i];
                if(flag
                )
                returnString+=String(c);
if(delimiter==c)    flag = true;
                }
return(returnString);
}
```

```
void rightArrow()
```

```
{
```

```
int refX = 50;
```



```
int refY = tft.setCursorY() + 40;  
    tft.fillRect(refX,refY,100,20,ILI9341_RED);  
tft.fillTriangle(refX+100,refY-  
30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
```

```

}

void leftArrow()

{

    int refX = 50;

    int refY = tft.setCursorY() + 40;

    tft.fillRect(refX+40,refY,100,20,ILI9341_RED);

    tft.fillTriangle(refX+40,refY-30,refX+40,refY+50,refX,refY+10,ILI9341_RED);

}

void

upArrow()

{

    int refX = 125;

    int refY = tft.setCursorY() + 30;

    tft.fillTriangle(refX-

40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED); tft.fillRect(refX-

15,refY+40,30,20,ILI9341_RED);

}

String APICall()

{ HTTPClient

http;

String url = "https://node-red-grseb-2022-11-05test.eugb.mybluemix.net/getSpeed?"; url +=

"location="+myLocation+"&"; url +=

"schoolZone="+((String)digitalRead(schoolZone)+(String))+"&"; url +=

"hospitalZone="+((String)digitalRead(hospitalZone)+(String))+"&"; url +=

"usualSpeedLimit="+((String)usualSpeedLimit+(String))+"&"; url +=

```

```
"uid="+ (String)uid; http.begin(url.c_str()); int httpStatusCode =
http.GET();if (httpStatusCode>0) {

    String    payload    =    http.getString();
http.end();    return(payload);

} else {

    Serial.print("Error code: ");

    Serial.println(httpStatusCode);

} http.end();

} void myPrint(String contents)

{

tft.fillScreen(ILI9341_BLACK); tft.setCursor(0,
20); tft.setTextSize(4);

tft.setTextColor(ILI9341_RED);

//tft.println(contents);

tft.println(stringSplitter1(contents)); String c2
= stringSplitter2(contents);    if(c2=="s") //
represents Straight

{

    upArrow();

}

if(c2=="l") // represents left

{

    leftArrow();
```

```
}

if(c2=="r") // represents right

{

    rightArrow();

} }

void setup() {

    WiFi.begin(ssid, password, 6);  tft.begin();

    tft.setRotation(1);

    tft.setTextColor(ILI9341_WHITE);
    tft.setTextSize(2);

    tft.print("Connecting to WiFi");

    while (WiFi.status() !=
WL_CONNECTED) { delay(100);

        tft.print(".");

    }

    tft.print("\nOK! IP=");

    tft.println(WiFi.localIP());

}

void loop()

{

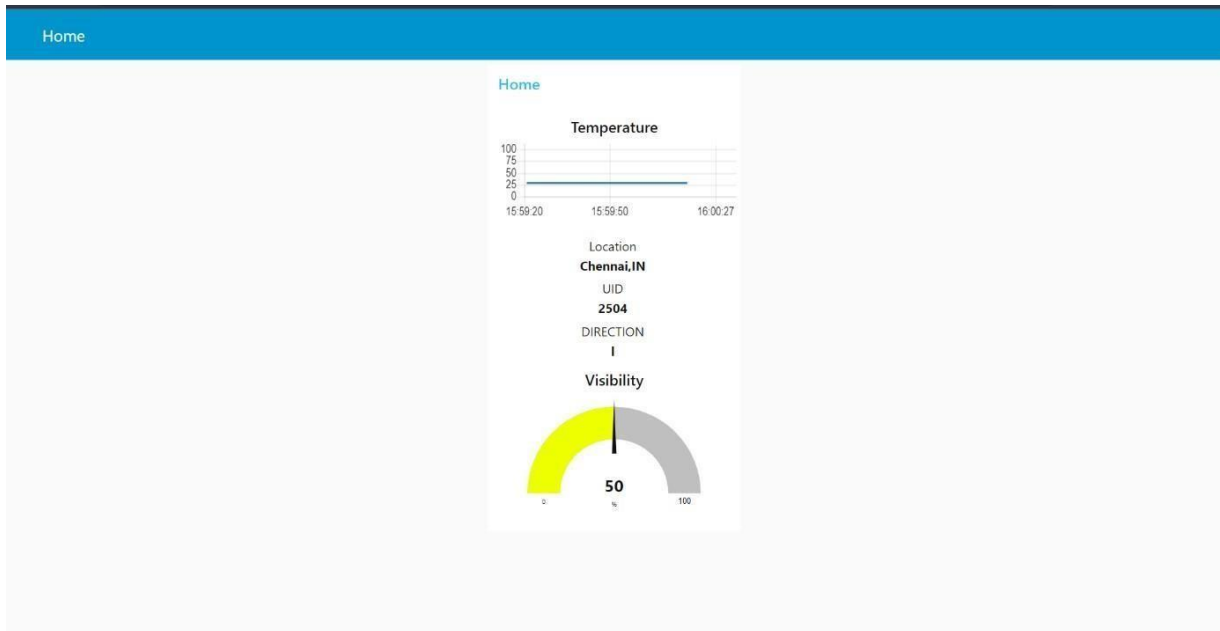
    myPrint(APICall());

    delay(100); }
```

---

## Output :

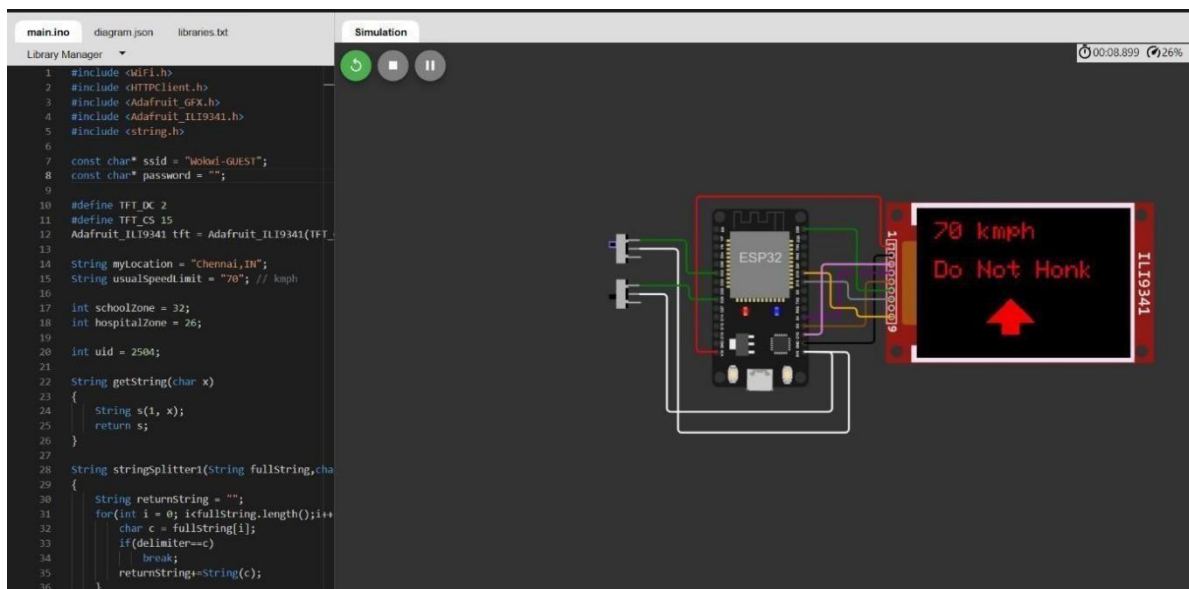
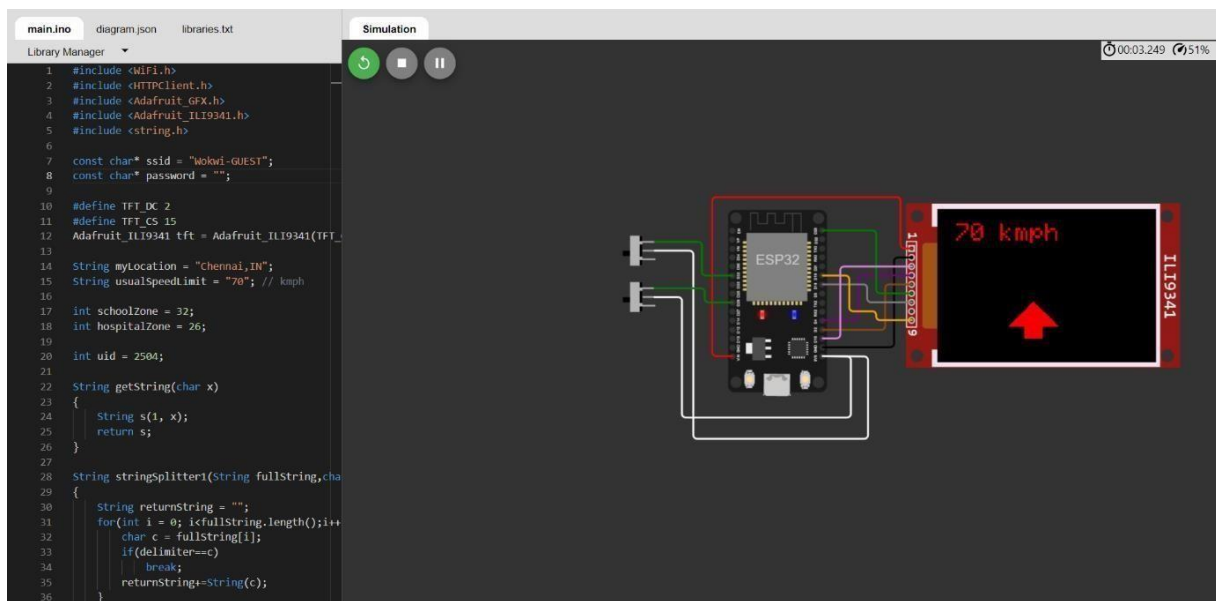
### Node Red Dashboard :

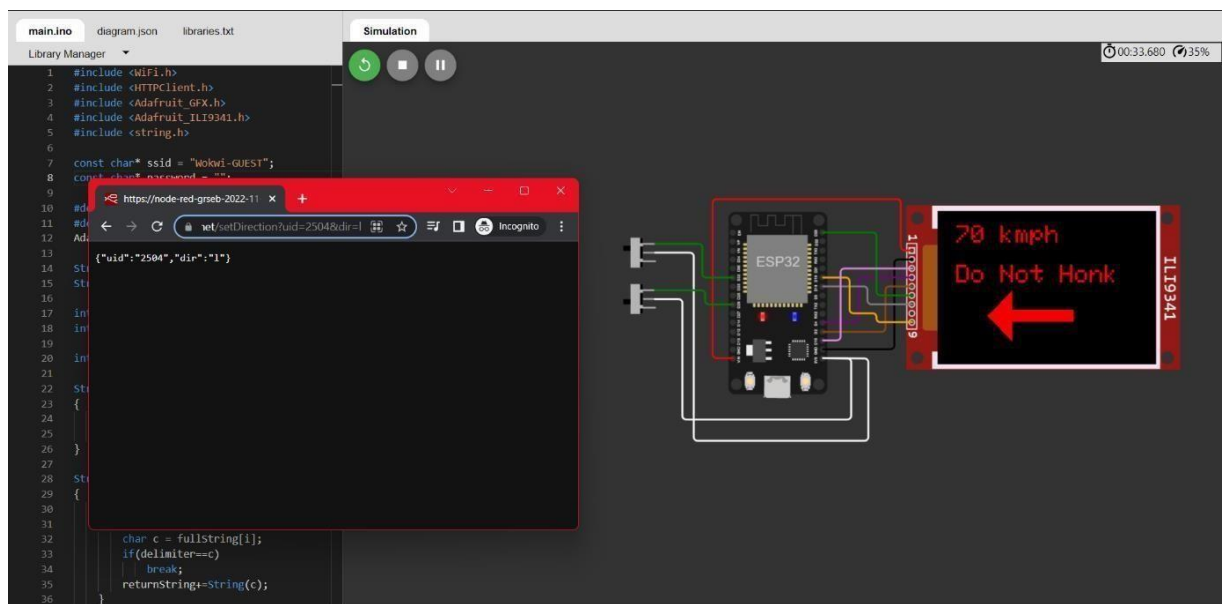
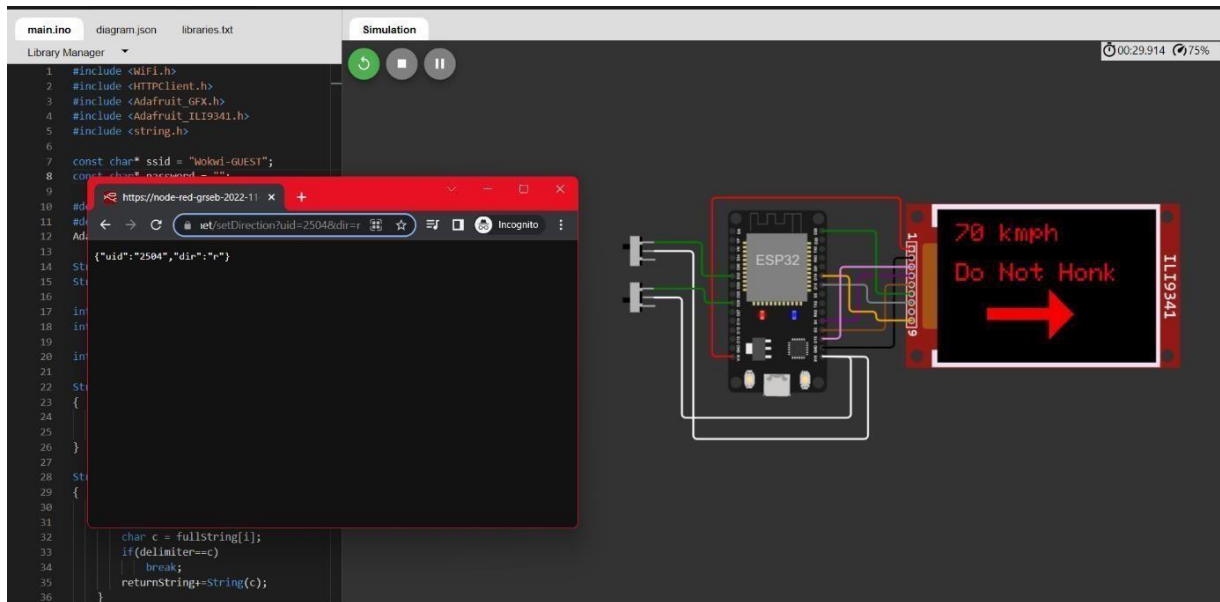


### Wokwi Output :

```
main.ino  diagram.json  libraries.txt  Simulation
Library Manager
1 #include <WiFi.h>
2 #include <HTTPClient.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_ILI9341.h>
5 #include <string.h>
6
7 const char* ssid = "Wokwi-GUEST";
8 const char* password = "";
9
10 #define TFT_DC 2
11 #define TFT_CS 15
12 Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
13
14 String myLocation = "Chennai,IN";
15 String usualSpeedLimit = "70"; // kmph
16
17 int schoolZone = 32;
18 int hospitalZone = 26;
19
20 int uid = 2504;
21
22 String getString(char x)
23 {
24   String s(1, x);
25   return s;
26 }
27
28 String stringsplitter1(String fullString, char delimiter)
29 {
30   String returnString = "";
31   for(int i = 0; i < fullString.length(); i++)
32   {
33     char c = fullString[i];
34     if(delimiter == c)
35       break;
36     returnString += String(c);
37   }
38 }
```

The simulation shows an ESP32 microcontroller connected to an ILI9341 display. The display shows the text: "Connecting to WiFi OK! IP=10.10.0.2".





### Change Directions Page

UID : 2504

left

straight

right

## 8. Testing

### 8.1 Test Cases

- **Test Case 1**

Clear weather - Usual Speed Limit.

- **Test Case 2**

Foggy Weather - Reduced Speed Limit.

- **Test Case 3**

Rainy Weather - Further Reduced Speed Limit.

- **Test Case 4**

School/Hospital Zone - Do not Honk sign is displayed.

### 8.2 User Acceptance Testing

Dynamic speed & diversion variations based on the weather and traffic helps user to avoid traffic and have a safe journey home. The users would welcome this idea to be implemented everywhere.

## 9. Results

### 9.1 Performance Metrics

Based on the IBM pack we chose, the performance of the website varies. Built upon NodeJS, a light and high performance engine, NodeRED is capable of handling upto 10,000 requests per second. Moreover, since the system is horizontally scalable, a even higher demand of customers can be served.

## 10. Advantages & Disadvantages

- **Advantages**



- Lower battery consumption since processing is done mostly by Node RED servers in the cloud.
- Cheaper and low requirement micro controllers can be used since processing requirements are reduced
- Longer lasting systems.
- Dynamic Sign updation.
- School/Hospital Zone alerts
- Costs are reduced.

- **Disadvantages**

- The size of the display determines the requirement of the micro controller
- Dependent on OpenWeatherAPI and hence the speed reduction is same for a large area in the scale of cities.

## **11. Conclusion**

Our project is capable of serving as a replacement for static signs for a comparatively lower cost and can be implemented in the very near future. This will help reduce a lot of accidents and maintain a more peaceful traffic atmosphere in the country.

## **12. Future Scope**

Introduction of intelligent road sign groups in real life scenarios could have great impact on increasing the driving safety by providing the end-user (car driver) with the most accurate information regarding the current road and traffic conditions. Even displaying the information of a suggested driving speed and road surface condition (temperature, icy, wet or dry surface) could result in smoother traffic flows and, what is more important, in increasing a driver's awareness of the road situation.

- Source Code - ESP 32

```
1 #include <WiFi.h>
2 #include <HTTPClient.h> 3 #include <Adafruit_GFX.h> 4 #include <Adafruit_ILI9341.h> 5
   #include <string.h>
6
7 const char* ssid = "Wokwi-GUEST";
8 const char* password = "";
9
10#define TFT_DC 2
11#define TFT_CS 15
12Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
13
14String myLocation = "Chennai,IN";
```

String

"70" // kmph

```
16
17int schoolZone = 32;
18int hospitalZone = 26;
19
20int uid = 2504; // ID Unique to this Micro Contoller 21
22String getString(char x)
23{
24    String s(1, x);
25    return s;
26}
27
28String stringSplitter1(String fullString,char delimiter='$') 29{
30    String returnString = "";
31    for(int i = 0; i<fullString.length();i++) {
32        char c = fullString[i];
33        if(delimiter==c)
34            break;
35        returnString+=String(c);
36    }
37    return(returnString);
38}
39
```

```
40String stringSplitter2(String fullString,char delimiter='$') 41{  
42  String returnString = "";
```

```
bool    false
for int  0
```

```
45     char c = fullString[i];
46     if(flag)
47         returnString+=String(c);
48     if(delimiter==c) 49         flag = true;
50     }
51     return(returnString);
52}
53
54void rightArrow()
55{
56     int refX = 50;
57     int refY = tft.setCursorY() + 40;
58
59                                     tft.fillRect(refX,refY,100,20,ILI9341_RED);
60                                     tft.fillTriangle(refX+100,refY-
61                                     30,refX+100,refY+50,refX+40+100,refY+10,ILI9341_RED);
62
63void leftArrow() 64{
65     int refX = 50;
66     int refY = tft.setCursorY() + 40;
67
```



```
68         tft.fillRect(refX+40,refY,100,20,ILI9341_RED);  
69         tft.fillTriangle(refX+40,refY-  
30,refX+40,refY+50,refX,refY+10,ILI9341_RED);
```

```
72 void upArrow()
73 {
74     int refX = 125;
75     int refY = tft.getCursorY() + 30;
76
77     tft.fillTriangle(refX-
40,refY+40,refX+40,refY+40,refX,refY,ILI9341_RED);
78     tft.fillRect(refX-
15,refY+40,30,20,ILI9341_RED);
79 }
80
81 String APICall() { 82 HTTPClient
http;
83
84     String url = "https://node-red-grseb-
2022-11-05-
test.eugb.mybluemix.net/getSpeed?";
85     url += "location="+myLocation+"&";
86     url +=
"schoolZone="+ (String)digitalRead(schoolZone)+(String) "&";
87     url +=
"hospitalZone="+ (String)digitalRead(hospitalZone)+(String) "& ";
```

```
88                                     url +=
    "usualSpeedLimit="+(String)usualSpeedLimit+(String)"&";
89                                     url += "uid="+(String)uid;
90                                     http.begin(url.c_str());
91                                     int httpResponseCode = http.GET();
92
```

```
if (responseCode == 0) {
    String payload = http.requestBody();
    http.end();
    return(payload);
} else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
http.end();
}

void myPrint(String contents) {
    tft.fillScreen(ILI9341_BLACK);
    tft.setCursor(0, 20);
    tft.setTextSize(4);
    tft.setTextColor(ILI9341_RED);
    //tft.println(contents);
    tft.println(stringSplitter1(contents));
    String c2 = stringSplitter2(contents);
    if(c2=="s") // represents Straight
    {
        upArrow();
    }
}
```

```
117     }  
118     if(c2=="l") // represents left  
119     {  
120         leftArrow();  
121     }  
     if     "r" // represents right
```

```
123  {
124    rightArrow();
125  }
126 }
127
128 void setup() {
129   WiFi.begin(ssid, password, 6);
130
131   tft.begin();
132   tft.setRotation(1);
133
134   tft.setTextColor(ILI9341_WHITE);
135   tft.setTextSize(2);
136   tft.print("Connecting to WiFi");
137
138   while (WiFi.status() != WL_CONNECTED) {
139     delay(100);
140     tft.print(".");
141   }
142
143   tft.print("\nOK! IP=");
144   tft.println(WiFi.localIP());
145 }
146
147 void loop() {
```

```
148   myPrint(APICall());  
149   delay(100);  
150 }
```

- Github And Project Link

<https://github.com/IBM-EPBL/IBM-Project-47665-1660801062>