

PUBLIC TRANSPORTATION OPTIMIZATION

INTERNET OF THINGS

PHASE 4: DEVELOPMENT PART 2



TEAM LEADER 1. NITHIYA KUMAR G H
TEAM MEMBER 2. VIGNESH S
TEAM MEMBER 3. NITHISH KUMAR.M
TEAM MEMBER 4. NAVEEN G
TEAM MEMBER 5. SANJAY A

INTRODUCTION:

Building a project to develop a real-time transit information platform involves creating a system that provides up-to-the-minute information about public transportation services. This typically includes data on the current locations of vehicles, estimated arrival times, route information, and possibly other relevant details. The goal is to offer passengers accurate and timely information to improve their commuting experience and make informed travel decisions. The platform may involve various technologies like GPS tracking, data analysis, and user interfaces for both web and mobile applications.

IOT DEVICE:

Sensors: IoT sensors will be installed in public transport vehicles, including buses, [Vehicle](#) trams, and trains. These sensors will collect data such as vehicle location, speed, fuel consumption, passenger count, and engine health.

Passenger Counters: IoT cameras or sensors will be placed at vehicle entry and exit points to track passenger counts and occupancy in real-time.

Infrastructure Sensors: IoT sensors will be deployed at bus stops, train stations, and other passenger hubs to collect data on passenger foot traffic, environmental conditions (e.g., temperature, humidity), and vehicle arrival times.

PLATFORM DEVELOPMENT:

The IoT data collected from the devices will be transmitted to a centralized platform for analysis and decision-making. The platform will consist of the following components:

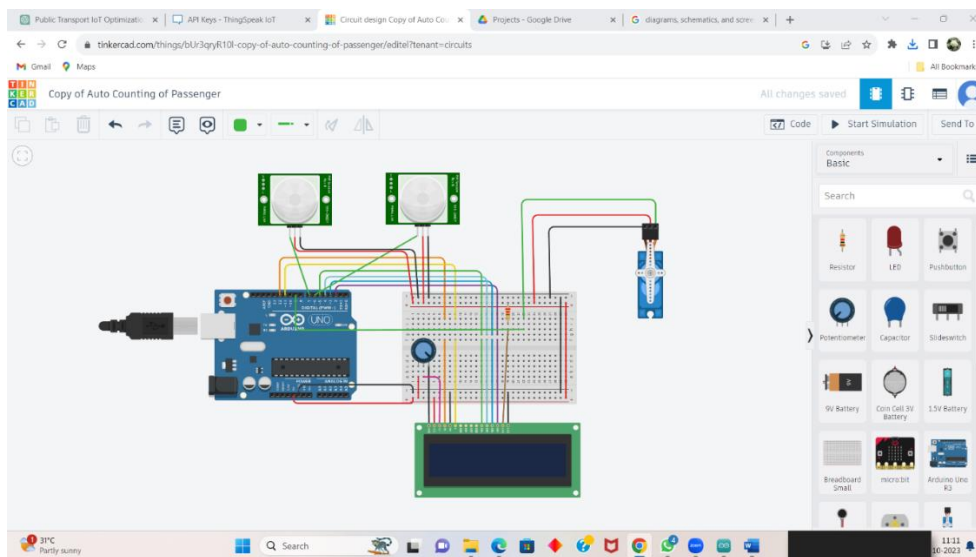
IoT Gateway: Data from the IoT devices will be sent to a central gateway for aggregation and forwarding. This gateway will be responsible for data preprocessing and transmission.

Cloud Data Storage: The preprocessed data will be stored in a cloud-based database for real-time and historical analysis.

Data Analysis and Prediction: Machine learning models will be developed to analyze the data and make predictions about vehicle schedules, maintenance needs, and passenger demand.

Dashboard: A user-friendly dashboard will be created for transportation authorities and commuters to access real-time information about vehicle locations, passenger counts, and estimated arrival times.

CIRCUIT DESIGN IN PUBLIC TRANSPORT:



PYTHON CODE FOR RUNNING ABOVE CIRCUIT:

```
#include <LiquidCrystal.h>

#include <Servo.h>

#include <ThingSpeak.h> // Include the ThingSpeak library

// Define your ThingSpeak channel details

char thingSpeakAddress[] = "api.thingspeak.com";

unsigned long channelID = 2303456; // Replace with your channel ID

const char * writeAPIKey = " 6EKT0ALDBXGG60Q1"; // Replace with your Write API Key

// Rest of your code...

void setup() {
```

```

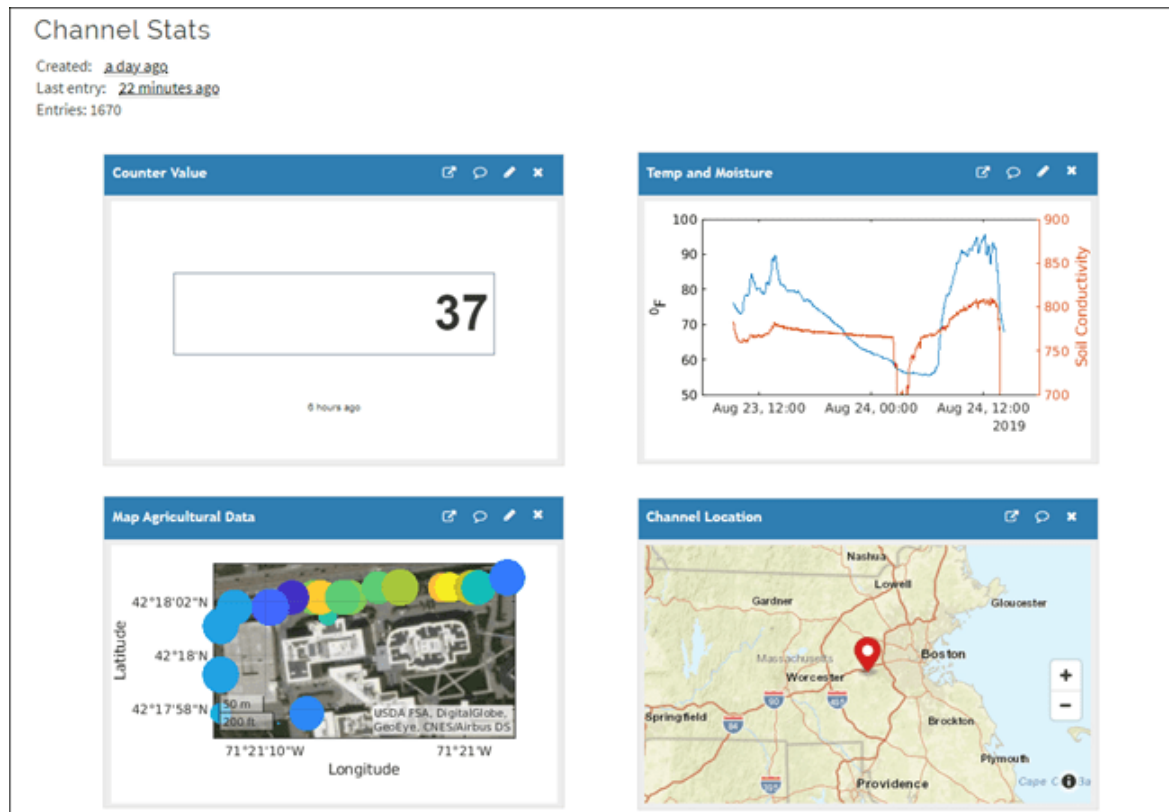
// Initialize ThingSpeak
ThingSpeak.begin(client); // Initialize the ThingSpeak library
// Rest of your setup code...
}

// Update ThingSpeak with passenger count
void UpdateThingSpeak(int count) {
    ThingSpeak.setField(1, count); // Field 1 is for passenger count
    int status = ThingSpeak.writeFields(channelID, writeAPIKey);
    if (status == 200) {
        Serial.println("ThingSpeak update successful");
    } else {
        Serial.println("Error updating ThingSpeak");
    }
}

// Update the passenger count and ThingSpeak when a passenger enters or exits
void UpdatePassengerCounter(int x) {
    Passenger = Passenger + x;
    lastRIPdetected = 0;
    if (Passenger >= 0) {
        UpdateThingSpeak(Passenger); // Update ThingSpeak with the new passenger count
    }
}

```

THINGSPEAK CHANNEL FOR REALTIME OUTPUT:



PLATFORM UI CODE FOR PUBLIC TRANSPORT OPTIMIZATION:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Bus Passenger Counter</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Bus Passenger Counter</h1>
```

```
  <p>Passenger Count: <span id="passengerCount">Loading...</span></p>
```

```
  <script>
```

```
    // Function to update passenger count from ThingSpeak
```

```
    function updatePassengerCount() {
```

```
      // Make an AJAX request to fetch the passenger count from ThingSpeak
```

```
      var xhr = new XMLHttpRequest();
```

```
      xhr.open("GET",
```

```
      "https://api.thingspeak.com/update?api_key=OL6MICDSS2G0VN7J&field1=0", true);
```

```
      xhr.onreadystatechange = function () {
```

```
    if (xhr.readyState == 4 && xhr.status == 200) {  
        var count = xhr.responseText;  
        document.getElementById("passengerCount").textContent = count;  
    }  
};  
xhr.send();  
}  
// Periodically update passenger count (e.g., every 10 seconds)  
setInterval(updatePassengerCount, 10000);  
// Initial update  
updatePassengerCount();  
</script>  
</body>  
</html>
```

OUTPUT FOR ABOVE PROGRAM:

Bus Passenger Counter

Passenger Count: 12