Name: Jeya Suriya M

Regno: 950321104022

Traffic Management IoT Project Documentation

Overview

In the fourth phase of our Traffic Management System project, our primary focus was on enhancing vehicle detection and congestion analysis. This project is aimed at addressing the challenges of urban traffic management by leveraging Internet of Things (IoT) technology. We made substantial progress by implementing Ultrasonic sensors, Arduino boards, and simulated traffic lights to improve the functionality and efficiency of our system. This documentation provides a comprehensive overview of our project, including objectives, IoT device setup, platform development, and code implementation.

Project Objectives

Our project aims to develop an efficient Traffic Management System to tackle urban traffic congestion. Key objectives include:

- Accurate vehicle detection and counting.
- Real-time congestion analysis.
- User-friendly access to congestion data.
- Cost-effective and accessible IoT infrastructure.

Vehicle Detection

In our Traffic Management System, we employ Ultrasonic sensors to detect the presence of vehicles on the road. The underlying principle is straightforward: when a vehicle passes by the sensor, the distance reading decreases, and when no vehicles are nearby, the distance reading remains at its maximum. This simple but effective method allows us to determine the degree of congestion on the road.

Pivots

During the project's evolution, we made several pivotal decisions to optimize our development process:

Utilizing Wokwi Platform

We decided to pivot and utilize the Wokwi platform for several reasons. Our team had previous experience with Wokwi, making it a familiar choice. This transition significantly simplified the development process and made it more accessible to all team members. Wokwi offered a robust environment for integrating our components and managing data flow efficiently.

Transitioning to Arduino Boards

In a move towards simplicity and cost-effectiveness, we shifted from our initial idea of using Raspberry Pi to utilizing Arduino boards. This change allowed us to maintain a cost-effective approach without compromising functionality. Arduino boards provided the necessary processing power and connectivity for our IoT components. Additionally, they allowed us to introduce LED simulations of traffic lights, which enhanced our system's capabilities. We chose to control the traffic lights in proximity to the Arduino board rather than connecting to a remote cloud-based system for controlling the lights, ensuring efficiency and reducing latency.

Congestion Density

It's important to note that our system does not provide a precise count of nearby vehicles. Instead, it offers an abstract score of congestion density, which serves as an indicator of traffic conditions in the area. This congestion score is valuable for traffic management and route optimization.

Three Key Components

Our project comprises three main components, each playing a vital role in achieving our objectives:

1. IoT Component

This component includes Arduino boards, Ultrasonic sensors, and simulated traffic lights. It is responsible for vehicle detection, counting, and local traffic light control. The Arduino boards collect data from Ultrasonic sensors and provide control signals for the simulated traffic lights.

2. Cloud Integration

We integrated the Wokwi platform as our cloud solution to facilitate data transmission and storage. The cloud component receives data on congestion levels from the IoT component, processes this information, and stores it for further analysis. This data is then visualized, allowing us to gain insights into congestion patterns and trends.

3. Client-Facing Application

A Dart-based client application has been developed to provide end-users with access to real-time congestion data. This application retrieves congestion data from the Wokwi platform based on the latest updates from our IoT component. Users can make informed decisions about their routes and travel plans based on this up-to-date congestion data. The application offers a user-friendly interface, making it easy for the public to access traffic information.

Data Flow

The data flow within our Traffic Management System is crucial for achieving our objectives:

1. The IoT component detects and records vehicle numbers, which are then transmitted to the Wokwi platform using standard IoT communication protocols.

2. The cloud component in Wokwi receives the data, processes it, and stores the information in a database for historical analysis.

3. Simultaneously, the client-facing application queries the Wokwi platform for the latest congestion data, enabling users to make well-informed decisions based on real-time information. This data flow ensures that the public can access up-to-date traffic information at their fingertips.

Code Implementation

The code implementation is a critical aspect of our project, encompassing the following steps:

Arduino Code

- Develop code for the Arduino boards to collect data from Ultrasonic sensors and control the simulated traffic lights.

- Implement data transmission protocols to send information to the Wokwi platform.

Cloud Integration Code

- Set up data reception and processing in the Wokwi platform to handle incoming data from the IoT component.

- Implement data storage and visualization, allowing us to monitor congestion patterns and trends over time.

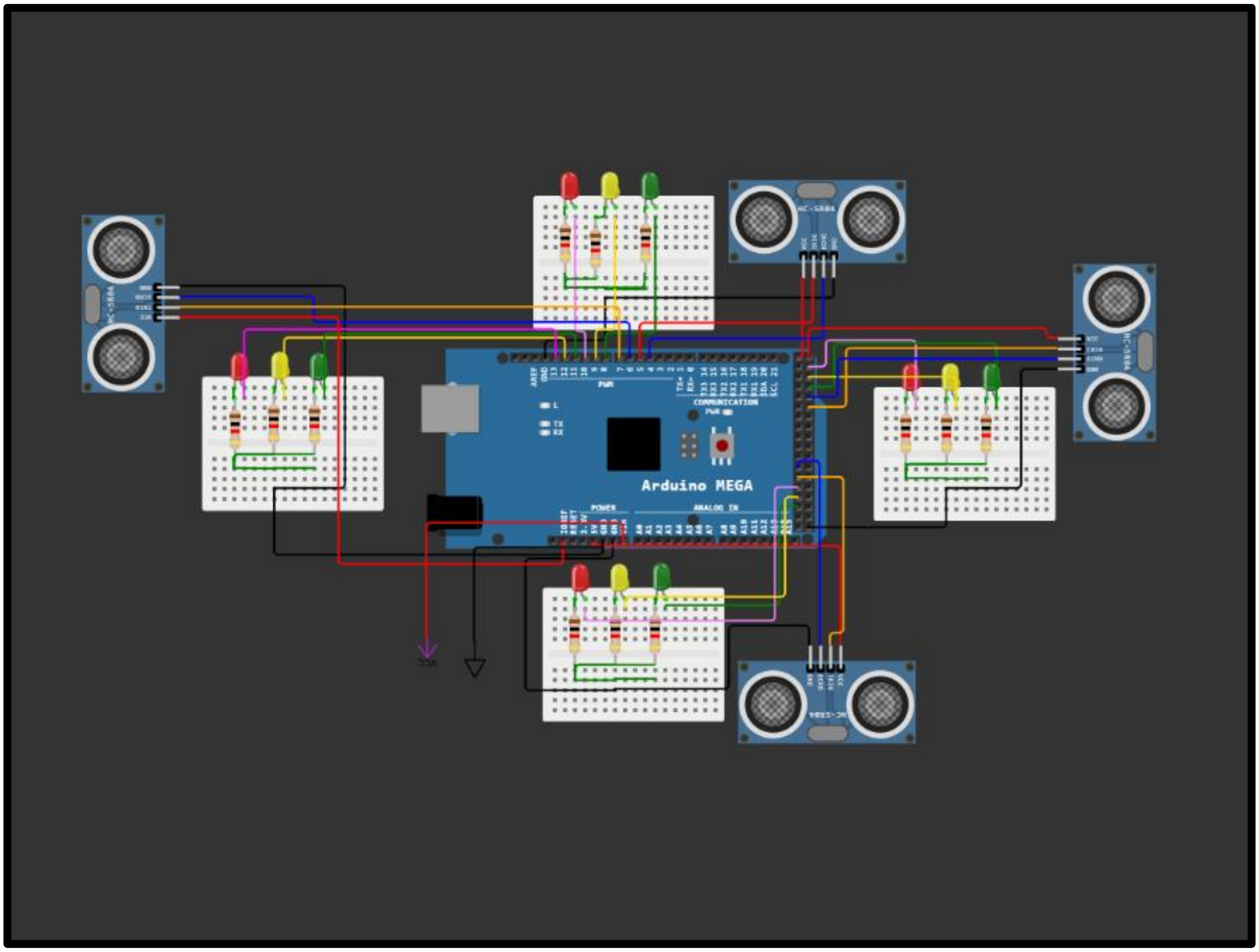
Client-Facing Application

- Develop a Dart-based application that communicates with the Wokwi platform to retrieve real-time congestion data.

- Create a user-friendly interface for end-users to access and visualize traffic information, enabling them to make informed decisions regarding their travel plans.

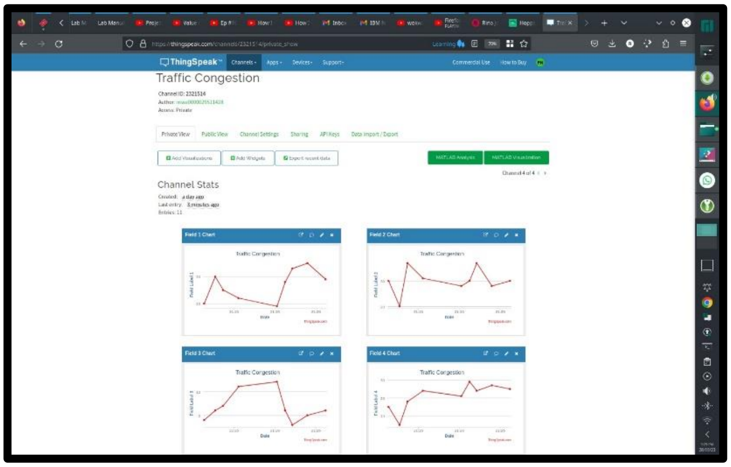
Diagrams, Schematics, and Screenshots

To provide a clear understanding of our project's architecture and components, we have included diagrams, schematics, and screenshots in our project documentation. These visuals illustrate the hardware and software components, as well as the data flow between them.



Example Outputs

To showcase the functionality of our Traffic Management System, we have included example outputs of IoT device data transmission and the user interface of the client-facing application. These examples demonstrate how our system effectively provides real-time congestion information to end-users.

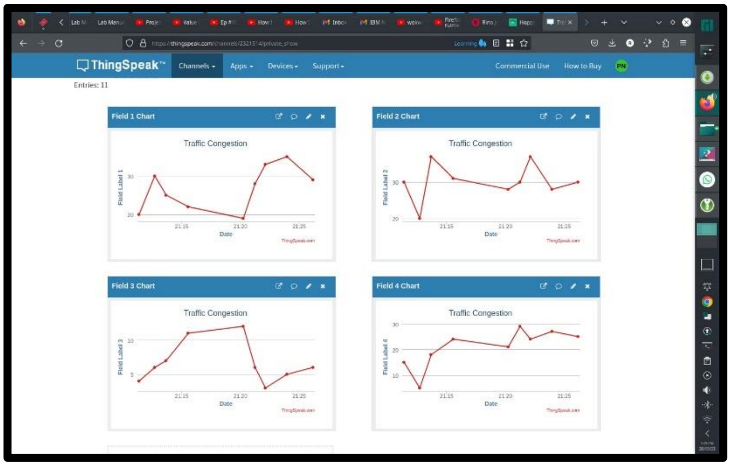


Traffic Congestion

Channel ID: 2321534
Author: mwa0000029511428
Access: Private

Channel Stats

Created: a day ago
Last entry: 5 minutes ago
Entries: 11



Entries: 11

GitHub Repository

Our project code and related files are available on GitHub. You can access the repository at the following link:

Replicating the Project

To replicate our project, follow these instructions:

IoT Device Setup

1. Acquire Arduino boards, Ultrasonic sensors, and simulated traffic lights.

2. Connect Ultrasonic sensors to the Arduino boards as per the manufacturer's instructions.

3. Write the Arduino code for data collection and simulation of traffic lights.

4. Ensure that the Arduino boards are connected to the simulated traffic lights and can control them effectively.

Platform Development

1. Sign up for a Wokwi account and create a project.

2. Follow the Wokwi documentation to set up data transmission between the Arduino boards and the Wokwi platform.

3. Create a dashboard in Wokwi to visualize congestion patterns and store data.

Client-Facing Application

1. Develop a Dart-based client application that communicates with the Wokwi platform to retrieve real-time congestion data.

2. Design and implement a user-friendly interface for end-users to access and visualize traffic information.

By following these instructions and referring to the code and documentation in our GitHub repository, you should be able to replicate the project, set up IoT devices, develop the data-sharing platform, and integrate them using Python or the appropriate programming language.

We hope that this documentation provides a comprehensive and in-depth understanding of our Traffic Management System project, its objectives, components, and functionality. If you have any questions or require further assistance, please do not hesitate to reach out to our team via the provided contact information in the repository.