Energy Conservation through smart management of street lights

Avinash Prabhu - 2018102027
Karan Mirakhor - 2018102034
Dipanwita G - 2018112004
Rahul Kashyap - 2018102037
Nihar Potturu - 2018102039
Sampath Kumar - 2018102026

Introduction

- Throughout the course of this project we hope to develop smart street lights which contribute to more eco-friendly and smart cities.

- We also hope to develop a solution which does not compromise on safety. We hope that the solution we come up with will not only reduce energy consumption but also help in keeping track of malfunctioning street lights so that timely repairs can be scheduled.

Objective

- **Energy Conservation :** To reduce unnecessary energy consumption by street lights by keeping them on only when there is movement in their vicinity.

- **Timely repairs** :To keep track of the traffic density and malfunctioning of streetlights by pushing all the data from the motion detection sensors to a local server.

- **Practical Utility :** To make the working model, as close as possible to an actual deployable project.

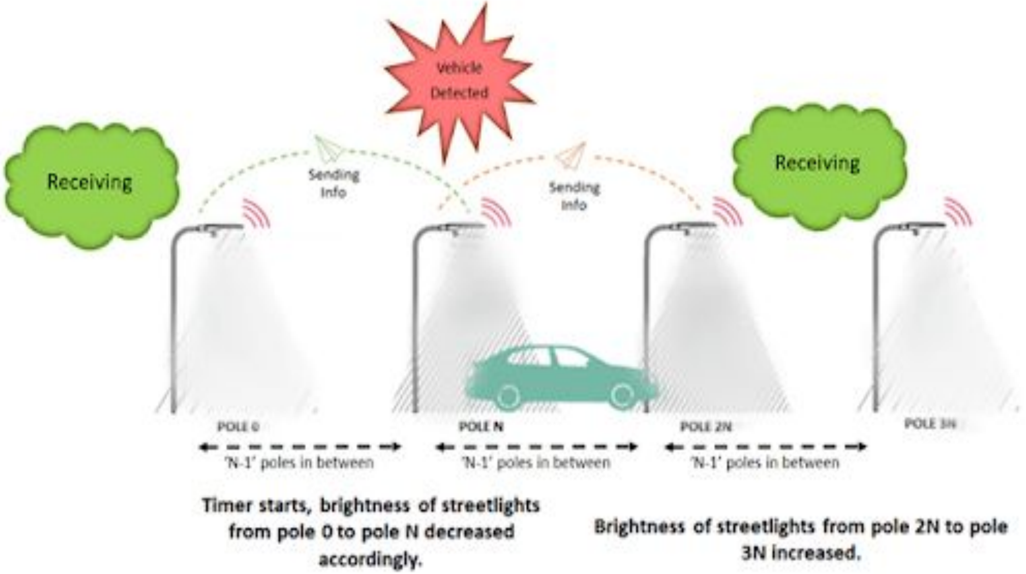
Main Idea

- We will divide the streetlights into sets of say ‘n’ streetlights each. The n is dependent on the road layout and the requirements such as locality, traffic, etc.
- At the beginning of each set, we will place our Master module.
- The operation of streetlights in any set will depend on the set’s master module. A master module can communicate to masters of it’s previous and next sets.
- Streets lights are controlled by master by observing data (presence of object) in current ,next and previous sets.

The above is a short video on the basic overview of the project and the components used.(It may take a few seconds to load)

Main Idea

- To know the presence of vehicles we use proximity sensor (**LIDAR/ultrasonic**)are placed at master nodes of each set to keep track of the count of the vehicles in the set .
- Street lights are controlled using the count of vehicles in the sets.
- Once the master streetlight registers a change in count, the count data is communicated to the neighboring streetlight via **LoRa transceiver.**



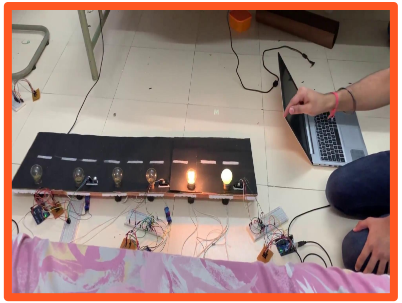
Timer starts, brightness of streetlights from pole 0 to pole N decreased accordingly.

Brightness of streetlights from pole 2N to pole 3N increased.

Main Logic

- We use vehicle count and timers to control the street lights
- The count takes care of the ideal case of control in which we are able to count each and every vehicle entering a section.
- There maybe a few cases in which detection or communication may not be perfect; in such a scenario we use timers that countdown to a particular value, with some presumptions, which are then used along with some particular conditions to toggle the lights.

1st Test case.



2nd Test case.

The above 2 videos are short videos explaining the logic of the solution we implemented a few test cases we solved. (It may take a few seconds to load)

Components we used

HPD13A SX1276 LoRa module

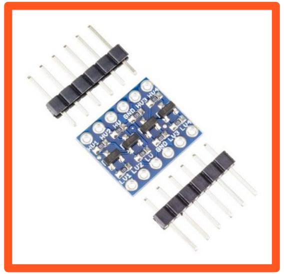
ESP8266 Board

HC-SR04 Ultrasonic Sensor

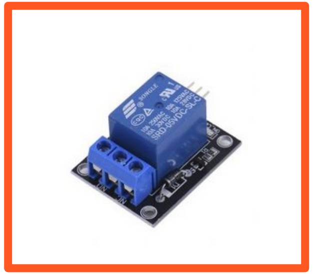
TFMini LIDAR Sensor

Arduino UNO

Logic Level Converter

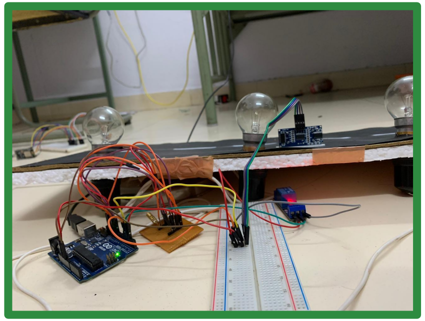


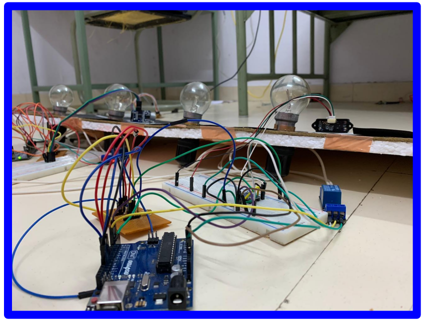
3.3V 10A relay

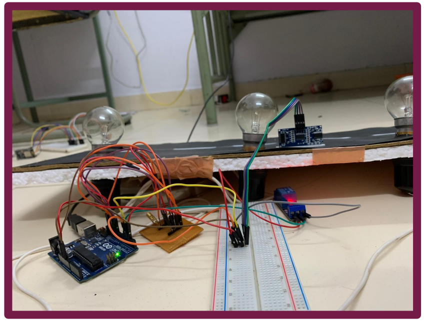


Modules Built - Master Modules

- As shown in the video, we have **3** sets of street lights with each set having a master module which communicate with each other and the gateway.
- Master module A - Consists of a LIDAR sensor, a LOGIC LEVEL CONVERTER, a LORA module and an ARDUINO board.
- Master module B - Consists of a LIDAR sensor, a LOGIC LEVEL CONVERTER, a LORA module, an ARDUINO board and a RELAY.
- Master module C- Consists of an ULTRASONIC sensor, a LOGIC LEVEL CONVERTER, a LORA module, an ARDUINO board and a RELAY.

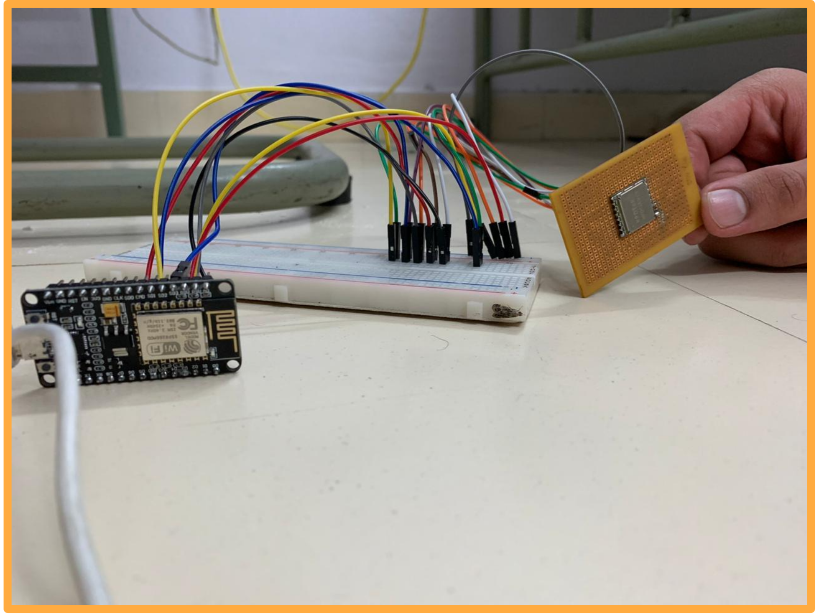


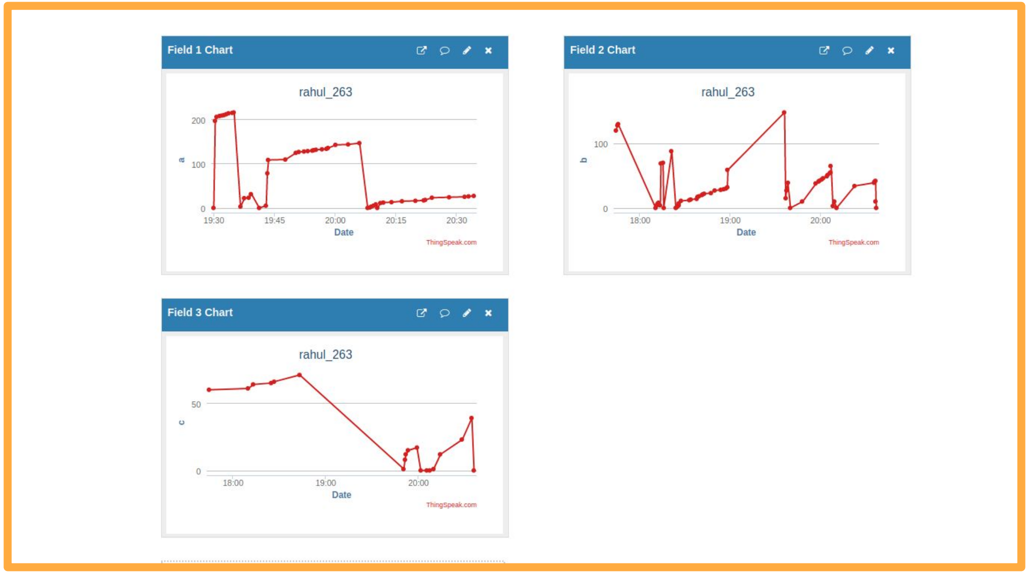




Modules Built - Gateway Module

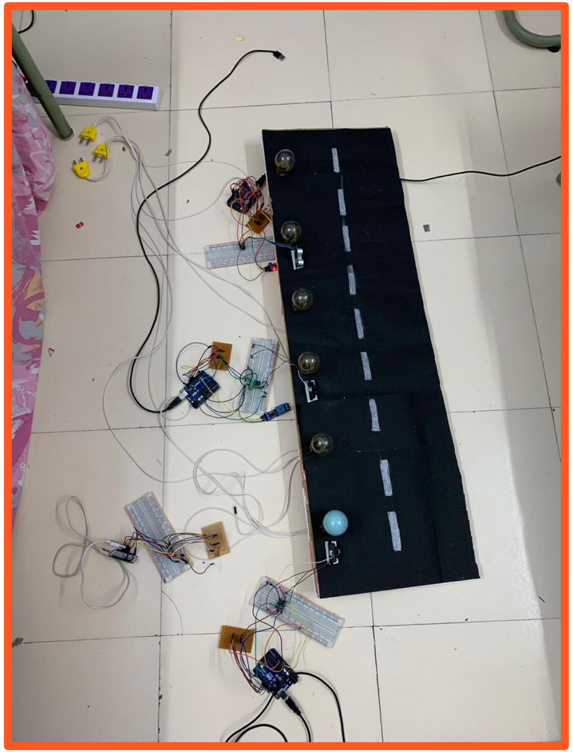
- As shown in the video, we have **1** gateway module which receivers data from modules A, B and C.
- The GATEWAY MODULE consists of a LORA module, an ESP8266 board and a THINGSPEAK server.





Modules Built - The track

- We have a basic model of a road with 6 streetlights divided into 3 sets.
- This was done to simplify the process of explaining our idea.



Results

- The masters of each set were successfully able to control the nodes of the next set as shown in the video.

- The logic we came up with to solve important cases were also implemented successfully as shown in the video

- The data was also successfully uploaded to the server as shown in the video

- The link for **Data in the server** can be seen here. (Note: The data in the server has a lot of spikes because the data was obtained during testing of the modules.)

- The count from the modules are pushed into thingspeak server every 15 seconds.The count is used to monitor traffic and to observe the state of street lights.This helps in keeping track of traffic and street lights.

- The link for the codes can be seen here.

Challenges encountered

- **Challenges with sensors:**
  - LIDAR can measure high speeds but the sensor(TFMini LIDAR) is very expensive, so we replaced the last sensor with an ultrasonic sensor.
  - We haven’t extended our implementation for pedestrians(Bi-directional traffic).
  - Two vehicles moving parallely at the same time will be counted as a single vehicle passing.
- **Challenges with LoRa communication:**
  - LoRa communication is based on Aloha protocol and hence it’s very unreliable, so every set needs to keep track of count from neighboring sets.
  - **Coding the LoRa modules was extremely challenging.**
  - We could not find any LoRa modules which were pre-soldered to a PCB. It took us a lot of time to solder them on to a board and make sure that they were working.

References

- Setting up LoRa (part 1)
- Setting up LoRa (part 2)
- Setting up gateway with esp8266 board
- To get started with LoRa coding, we took the help of sample codes in arduino.
- Setting up thingspeak server
- Reference for TFMINI
- HC-SR04
- logic-converter