12/16/2021

Density Based Traffic Signal

NET-455: Final Project

Name:	Aymen Zekeria	M80007954
	Tomas Petros	M80007961

Group #: 7

Instructor: Dr. Safa Otoum

Table of Contents

1. Introduction

Imagine being in a hurry to an important meeting with your boss where you want to show him/her an impressive idea which maybe your greatest leap to success, or in a hurry to save a life with your car and being stuck in traffic light now and then. It's a quite frustrating scenario when you are in an utmost hurry. According to the National Association of City Transportation Officials, the average time spent waiting at a red light is about 20% of total driving time (Levinson, 2020). Also, this problem seems to never go away as the number of cars is increasing year after year and the traffic signal system sees no improvement. If the current pace of increase continues, with the total number of cars doubling every 20 years, we may expect to see 2.8 billion automobiles on the road by 2036 (Chesterton, 2018). This implies that, given the current infrastructure and traffic signal architecture, the time spent waiting at a red light would jump up more than 20%.

The problem, as we identified, with most of the currently implemented traffic signals is the hard-coded time delay for the light bulbs for each side of the intersection. This could be easier to manage or create, from the developers' point of view, but it is not efficient in the practical world. Drivers must wait equal time delays from all sides of the intersection. This implies to every scenario, whether there are no cars or heavy congestion on the road. So, a busy side of an intersection will be allocated with the same green delay time (the green to be ON for a certain amount of time) as the other side of the intersection with or without congestion.

Our group decided to come up with a strategy to leverage the WSN concept to propose a technique to improve the efficiency of traffic signal control. Managing traffic flow through a junction, on the other hand, demands careful study and evaluation of a variety of criteria, including approach speed, space, cycle time, cost, traffic kinds and volumes, sight distance, and human factors (habits, expectations, and reaction times). As a result, we decided to narrow the scope of our study and concentrate on how to improve the traffic light algorithm. The traffic light is examined in this project report paper, and a new system based on WSN is presented. The following is how the paper is structured: Section 2 is an introduction to the project concept. Design and implementations in Section 3. Section 4 examines the design's outcome. Challenges, recommendations, and lessons learned are discussed in Section 5. The conclusion and future work of this research are presented in the last part.

2. Project Idea/Scenario

In this project, as stated earlier we are trying to propose a traffic light system that is aware of the density of the traffic and opens the roads according to the traffic. In the traditional traffic light system, always there is a fixed number of seconds/minutes given to all the roads and if one road has more congestion while there are no cars on the other road, there is nothing to be done to help release the congestion. However, in this project the system will have a look at the number of cars that are found in the road before turning the green light on and will base the number of seconds given to the green light will depend on the number of cars that are found in the road. The system will use IR sensors to count the number of cars that have entered the road and will be placed at the begging of the road and will count the total number of cars that pass by it. The system will then use this reading to decide how many seconds a green light on the given road will be turned on. In this project, we have tried to classify the road conditions into three scenarios/categories:

Scenario	***Congestion***	***Number of cars***
Scenario I	***Normal***	***< 10 cars***
Scenario II	***Moderate***	***> 10 cars and < 30***
Scenario III	***High***	***> 30 cars***

* N.B The Number of cars here is chosen on a sense that we think makes a road congested or normal and are not based on anything else.

3. Design and Implementation

3.1 Components and Pieces of Equipment Required

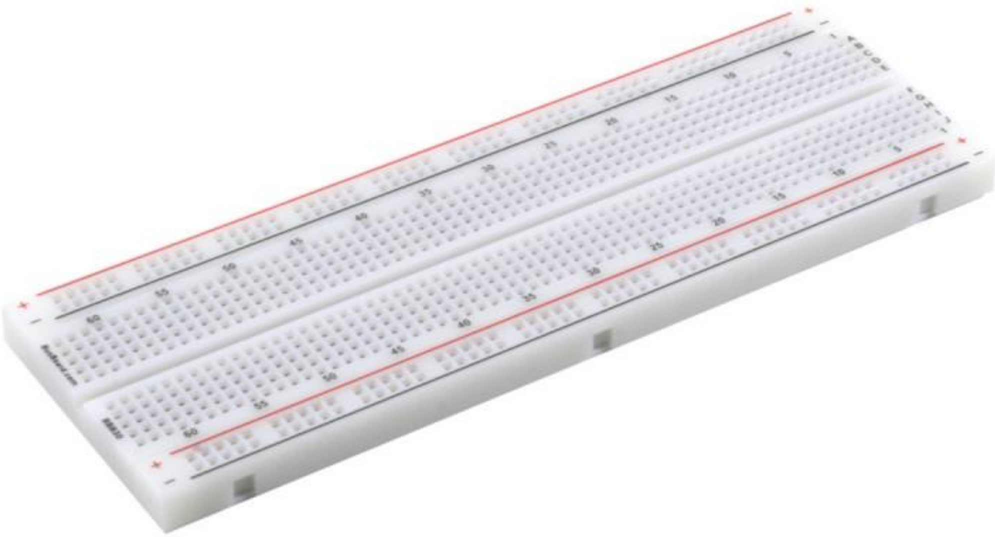
To perform this project, we need several components of electronic materials. In total, we used 5 pieces of equipment. These pieces of equipment are an Arduino Uno, a breadboard, several jumper wires, 8 LED lights (4 red, 4 green), and 4 infrared obstacle detection sensors. Below, we will discuss each component in brief. All the components were provided by the university in the Arduino uno starter kit, however, we bought the IR sensors as the starter kit that we received didn't include them.

Arduino Uno

The Arduino Uno is a programmable microcontroller board that may be used to design and build interactive circuits. 14 digital input/output pins, 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power connector, an ICSP header, and a reset button are all included on the Arduino Uno Board. It contains all the elements required to support the microcontroller. To get started, we can simply connect it to a computer by USB or power it with an AC-to-DC converter or battery. The Arduino Uno is a hardware device that may also be used with the software. You may use the Arduino software IDE to work with your Arduino Uno and program it. The software (the IDE) provides a simple platform for uploading programs to the Arduino board, and it may be used by simply connecting the board to a computer through a USB link.

Breadboard

A breadboard is a rectangular plastic board with a handful of small holes. Its primary function is to join and construct circuits. The holes in the beadboard are not all linked, but the holes in one row are. These holes make it simple to insert electrical components and design and test electronic circuits of any type. To connect gadgets or other circuits on the breadboard, you can use jumper wires. A breadboard is used to build and test circuits quickly before finalizing any circuit design. In this project, we used the breadboard to connect the components to the microcontroller which is the Arduino Uno and to power them with electricity.

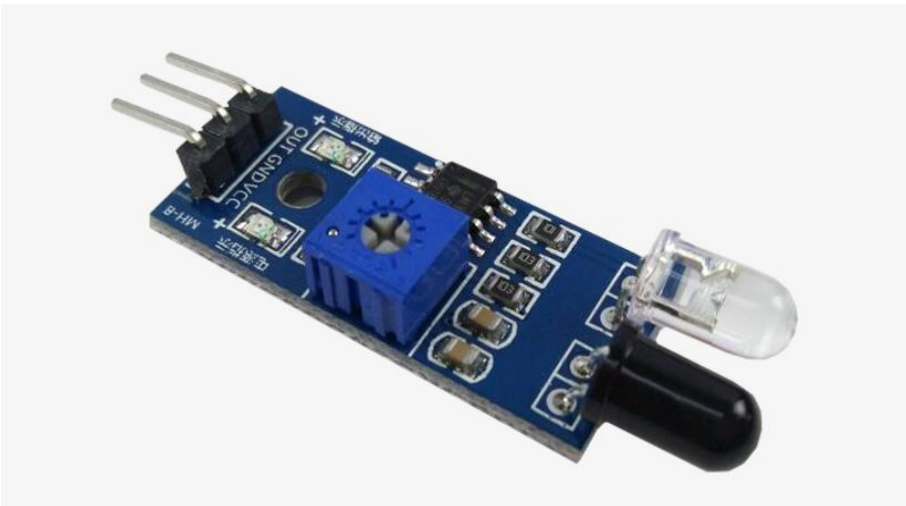


LED Lights

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. It has a wide range of applications because of its small size, low energy consumption, long lifespan, and versatility in terms of application. In this project, LEDs are used as traffic light signals. We will use 4 red LEDs and 4 green LEDs but not a yellow LED. This is due to the limited number of digital pins in the Arduino Uno microcontroller.

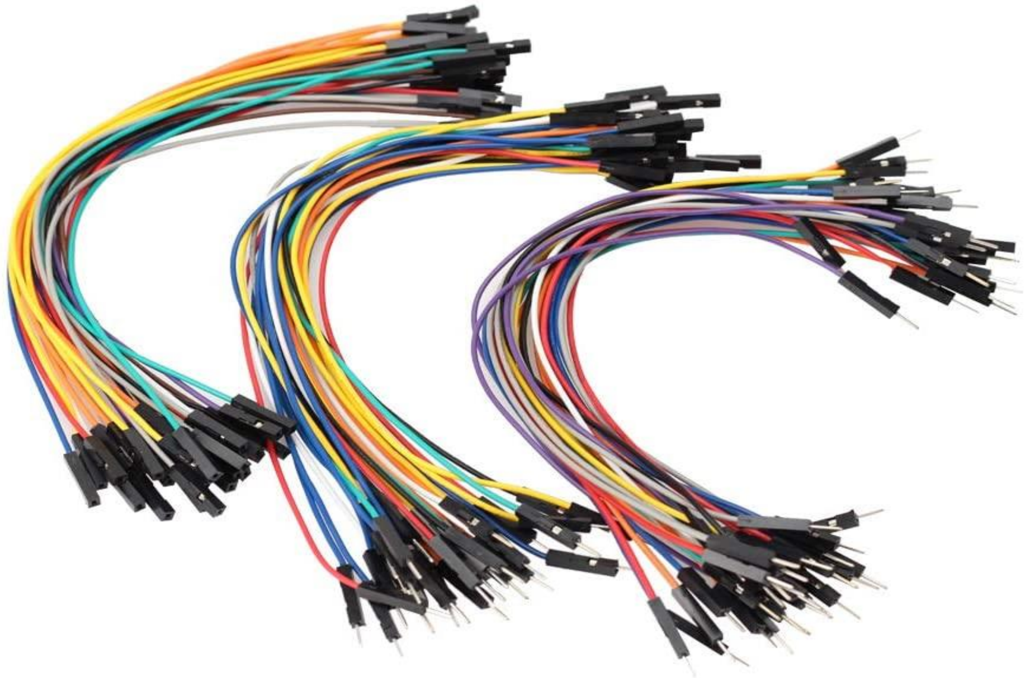
Infrared Obstacle Detection Sensors

Obstacle Avoidance IR Sensor board is an inexpensive solution to collision avoidance detection for robotics, smart car, and other electronic uses. This type of IR sensor has a built-in IR transmitter (the Black diode) and an IR receiver (the white diode). IR sensors work by measuring or reading infrared radiation. Infrared radiation is invisible to the human eye so one cannot see what is being emitted by the naked eye. However, the IR receiver can read the presence of infrared radiation and consider it as an obstacle is met. In this project, we will use infrared obstacle detection sensors to count the number of cars that are about to enter a junction with a traffic signal. This count will be used later to determine the status of the road.



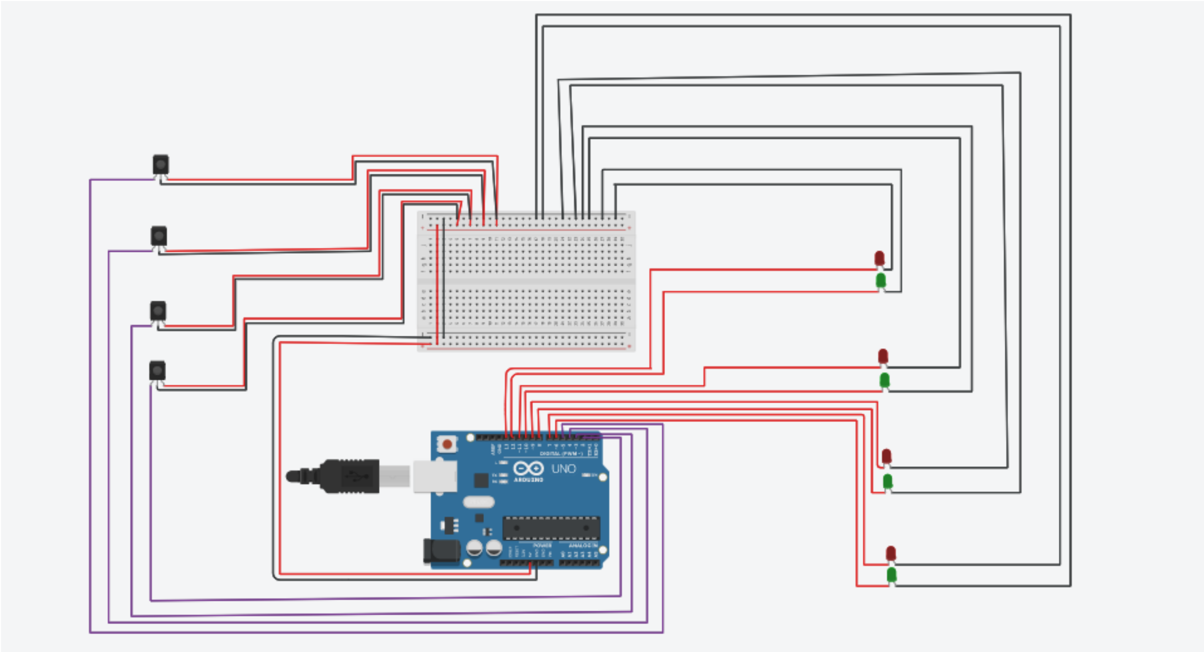
Jumper Wires

Jumper wires are electrical cables that are used to link various components and devices in a circuit. They're built of copper, and the fact that they're wrapped in different colours of plastic makes them easy to recognize and utilize without confusion. Jumper wires are easier to use with breadboards, and circuits.



3.2 Logical Design with tinkercad.com

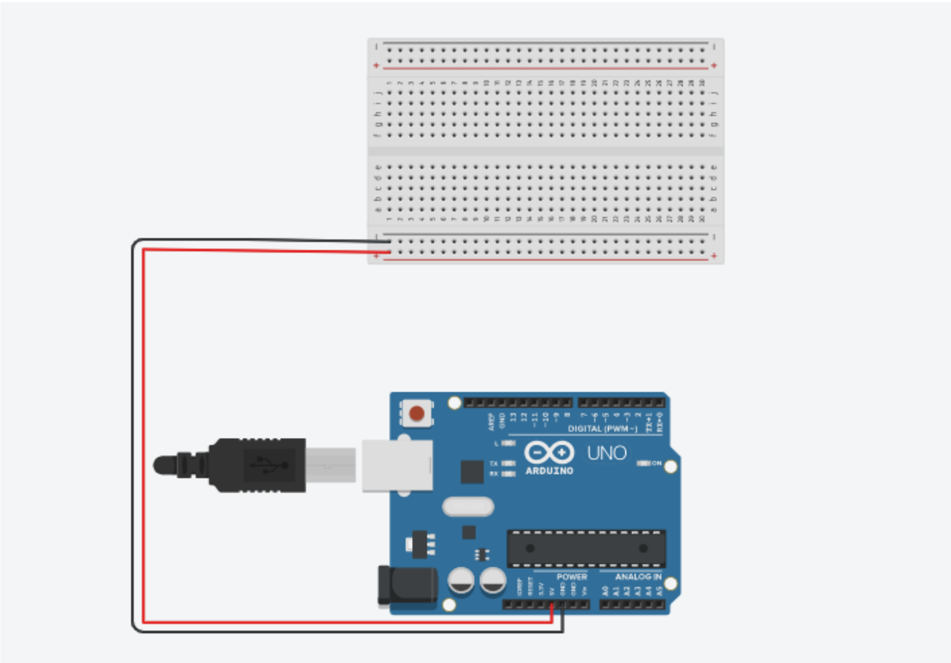
In this project, we used an Arduino simulation tool called tinkercad.com offered by autocad.com for free on the web to outline the logical design of the task. This tool is used for simulation purposes also. However, we used it for the sole purpose of logical design which was supposed to be done by software such as proteus which doesn't offer a free version. This design was done after we figured out which components and equipment are we going to use to build the logical as well as the physical model.

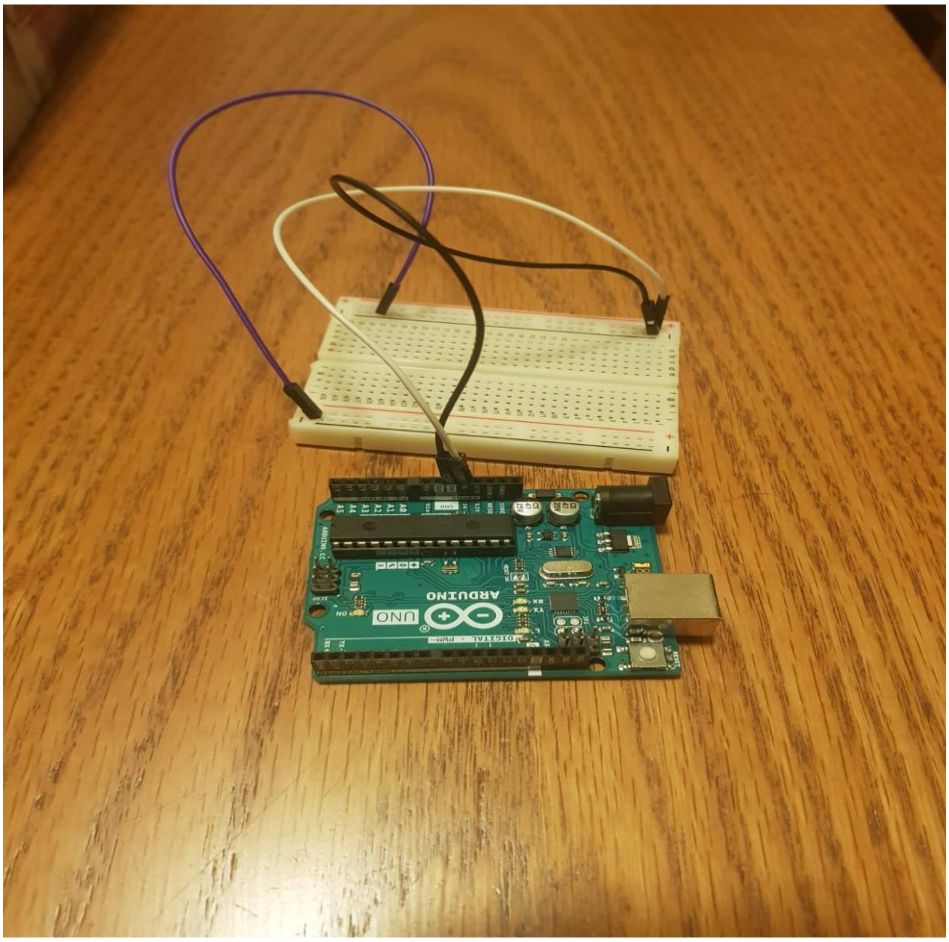


3.3 Implementing the Design

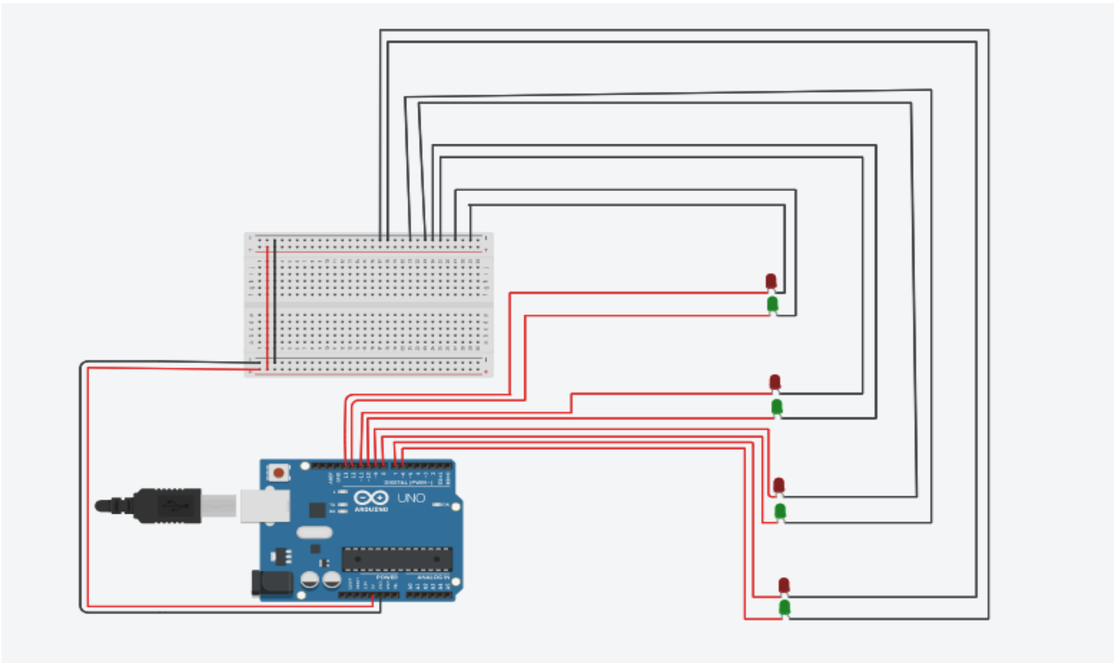
After completing the design in tinker cad, we started implementing our design and putting every component at its place. In order to make the process smoother we have tried to group similar tasks together and complete tasks by groups. The reason for the grouping is because our project includes task repetition. For example, all the traffic lights do the same thing and connecting the lights to the board is more or less the similar. Having this in mind, the below is the implementation of the design.

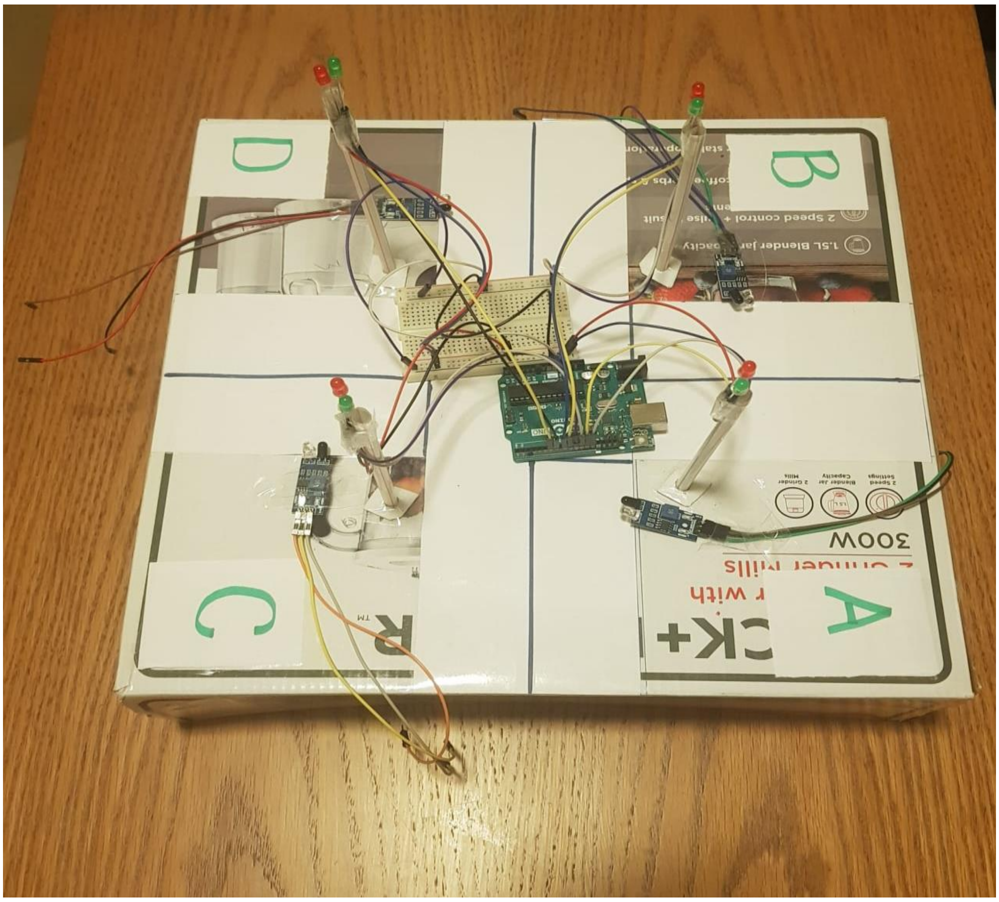
Task 1: Connecting the Breadboard to the Arduino.



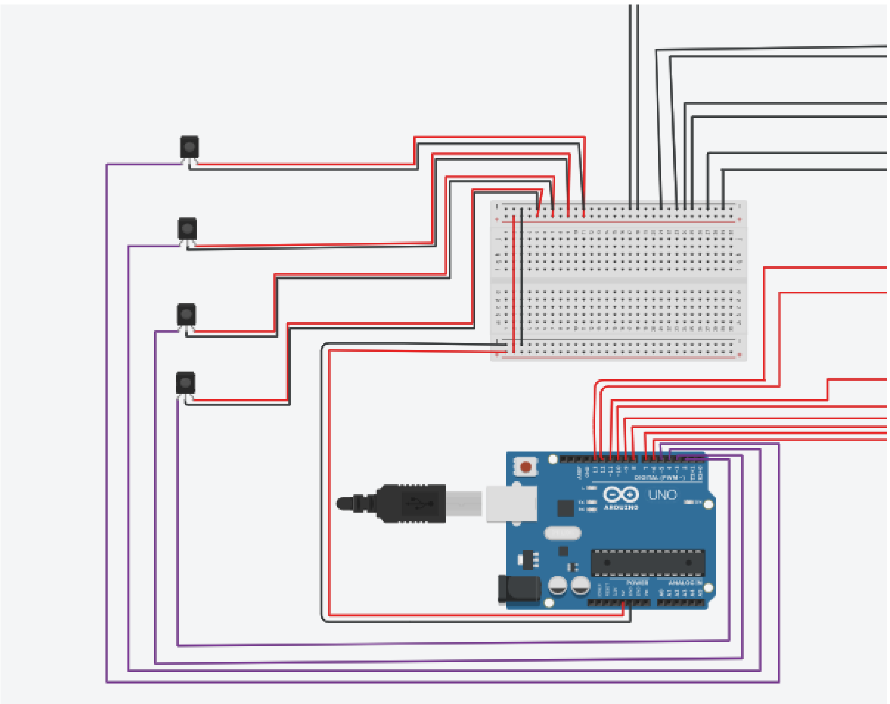


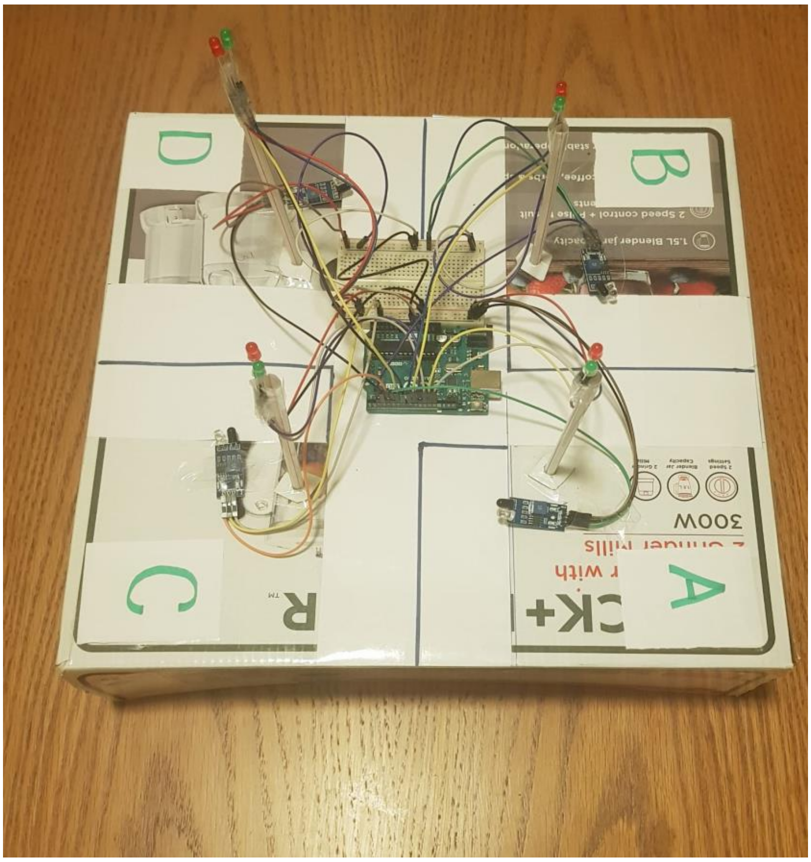
Task 2: Connecting the Traffic Lights to the Arduino and Breadboard.





Task 3: Connecting the Sensors to the Arduino and Breadboard.





4. Results of Experiment

Scenario #1: *If there are less than 10 cars on the road:*
Here the number of cars that are on a given road is less than 10 cars, and we have considered this to be the default case. So, when there are less than 10 cars on a given road, the green light will be turned on for 5 seconds.

Scenario #2: *If there are more than 10 but less than 30 cars on the road:*
Here the number of cars that are on a given road are more than 10 but less than 30. We have considered this scenario to be moderate congestion. So, when there are more than 10 cars but less than 30, the green light will be turned on for 8 seconds.

Scenario #3*: If there are more than 30 cars on the road:*
Here the number of cars that are on a given road is more than 30. We have considered this scenario to be high congestion. So, when there are more than 30 cars on a given road, the green light will be turned on for 15 seconds.

5. Discussion

5.1 Challenges

This project was full of challenges hence, creating a lot of learning experience and challenging our critical thinking skills. From the start to end, we were finding and stumbling across major problems but nevertheless, we managed to complete the project with a working demo. The first problem we came across was a limited number of pins that are built into the Arduino Uno microcontroller. Thus we were unable to connect yellow or amber LED to our project to make a complete signal system. This problem became almost unsolvable to accommodate all the LEDs that are used in this traffic signal system. So, we had to redesign all the projects to operate only on green and red lights. Therefore, we managed to solve the problem of limited digital pins by completely redesigning our working scheme to a completely new one.

Another major challenge we faced in this project is the lack of deep knowledge of the C++ programming language and designing an efficient working algorithm. As we tried to build the demo from scratch ourselves due to the lack of similar working logic on the Internet, we have to have a solid understanding of C++ language and efficient algorithm designing. So, we had to learn the syntax of the C++ language to code the program that operates the system. This took us not a considerable amount of time because we had a good understanding of programming languages such as python. For the algorithm part, we tried to maximize the efficiency as much as possible with a lot of try and error. In the end, we were able to draft a working algorithm and code it with the C++ language.

The most exhausting and prominent challenge we encountered during the implementation of this system was the inability of Arduino Uno to perform multiple tasks at a time or the inability to multithread a program. As we saw the Arduino is performing two tasks concurrently, one task being able to sense the environment and the other task blinking the LED lights accordingly. However, due to the behaviour of Arduino Uno not to multitask, we couldn’t sense the environment while simultaneously regulating the blinking of the LED lights. We tried to solve this problem by using two Arduino Uno boards where one performs one task and the other perform another and talk to each other using the Master-Slave scheme. However, this also was very complicated, and hard to manage the code between the two boards. In the end, we solved the problem by avoiding any delay functions, which was the main problem, in our code and replacing it with a function called *millis()* that performs a similar task to delay.

5.2 Recommendations

After completing the project and gaining a lot of experience, we recommend that one should start by having a basic understanding of electronics such as connecting electronic equipment. For example, a basic understanding of anode and cathode legs or pins could save a lot of time and energy while implementing. Also, a good understanding of C++ or C or any programming language that is may be used to manipulate or operate an Arduino Uno board is required. Moreover, a good logical design of the circuit helps a lot in saving time and energy. This is due to the fact that a logical design is used to visualize the connections and gives a better understanding of the system whilst the implementation will be full of jumper wires crossing each other which may make one dizzy.

5.3 Lessons Learned

- Learned a lot of C++ coding.
- Learned more about how sensors operate.
- How to work with two Arduinos using the Master-Slave scheme.
- Pseudo-multitasking using a Milli() function instead of delay.

6. Conclusion and future work

In the end, we want to say that we learned a lot from the project. We gained more knowledge on the practical aspect of the WSN applications. It was a fun experience to work with and control hardware with a program that we never have done before. This project helped us to work and collaborate as in a group and enhance our communication skills. Also, due to the major problems and challenges we were facing, it helped us to test our critical and analytical thinking skills. And for the future, we want to work on it and enhance the algorithm. Also, we want to dig deeper on the logic we used to control the congestion since there seems no previous work that uses the number of cars to determine if there is congestion on a road or not. We want to continue on the work and provide something useful to society and governments. Finally, we can proudly say that we are now able to work with almost any WSN equipment as we gained a considerable knowledge and experience from the project.

7. References

Chesterton, A. (2018, September 20). *How many cars are there in the world?* CarsGuide. Retrieved December 16, 2021, from https://www.carsguide.com.au/car-advice/how-many-cars-are-there-in-the-world-70629

Levinson, | D. (2020, January 2). *How much time is spent at traffic signals?* Transportist by David Levinson. Retrieved December 16, 2021, from https://transportist.org/2018/03/06/how-much-time-is-spent-at-traffic-signals/