

**Capstone-2 TCET 4282 – E299**

**Prof: Hossain**

**Project Proposal**

**“Interactive Traffic System”**

Participants:

Manuel Mane Penton

Nayib Ega

Balewa Glen

Telecommunication Technology Engineering

School: New York City College of Technology

Due date: 10/07/2019

Contents

[Abstract 2](#_Toc21362034)

[Objectives 2](#_Toc21362035)

[Introduction 3](#_Toc21362036)

[Methodology and Design 3](#_Toc21362037)

[Flow Chart 5](#_Toc21362038)

[Road Representation – PoC 7](#_Toc21362039)

[Timeline 8](#_Toc21362040)

[Budget 9](#_Toc21362041)

[References 10](#_Toc21362042)

# Abstract

Traffic congestion is a common issue in cities such as New York, which is the third most congested city in the world and the second worst traffic-jammed city in United States. New York drivers spend 13 percent of their time sitting in congestion as average, with 11 percent of that being attributed to daytime traffic. It is expected that traffic congestion will cost the city $100 billion over the next five years. Not to mention the problems that traffic congestions causes to emergency vehicles. For that reason, governments has been taking different measures to alight this situation such as the congestion pricing plan and the modernization of highways, roads, and streets as well as mass transportation systems.

Information and Communication Technology (ICT) is taking an important role assisting city planners collecting, analyzing data, designing and implementing new systems and protocols aimed to resolve congestion issues. Technologies such as Adaptive Traffic Signals, Real-Time Traffic Feedback and Pedestrian Tracking System are already in use in many large cities.

Interactively changing the street rails according to the traffic demand is a resource that some cities are already implementing to improve their traffic flow. Our project consists in building a traffic system that allows streets, roads, and rails intersections signaling to adjust to real-time traffic conditions. An embedded system with Wi-Fi interface can be used to build a system capable of reading the traffic needs, adapting the rails lane accordingly. In addition, this system interacts with the city first responder units such as the NYFD who would have the capabilities to manipulate traffic signals and streets lines. These will allow them to create the condition for a better circulation and to improve arrival time.

A train crossroad system will be proposed as well to provide an emergency system for vehicles stuck in the line when trains approach.

# Objectives

* To develop a traffic and street rail system that will be able to read real-time traffic conditions and makes the necessary adjustments to alleviate congestions and improve the traffic flow.
* To develop an interface for first-responders that will allow them to communicate with the system and to override street-way directions and traffic signaling conditions.
* To develop a train-crossroad system that will includes an emergency module. This safety mechanism will trigger in the event a vehicle or object blocks the crossroad as a train approaches.

# Introduction

Understanding the volume of vehicles in the road will be possible by the utilization of sensors located in both sides of the street (e.g. North and South). The sensors reading will generate a count. This data will be periodically analyzed and compared. If a threshold is cross, the system will decide if a lane needs to be taken from the side of the road with the least traffic-volume and to re-assign it to the busy side of the road. The extra lane will help to ease traffic flow and reduce congestion.

A web server/client interface will be developed between the lane sensors and the rail controls to display their performances. Emergency vehicles will have access to this control interface to modify the lanes as they need

A second interface will be developed for the crossroad system. It will provide a safety mechanism in the event a car or any other object stalls in the crossroad as the train approaches. The system will communicate with the crossroad-sensor to ensure there are not vehicles or objects stalled in the way. If the cross road is blocked, an alert will be sent to the train engineer which will pop on the web interface. This will allow the conductor to stop the train in time to avoid a collision. The system will proceed to return to service once the emergency and the road is cleared.

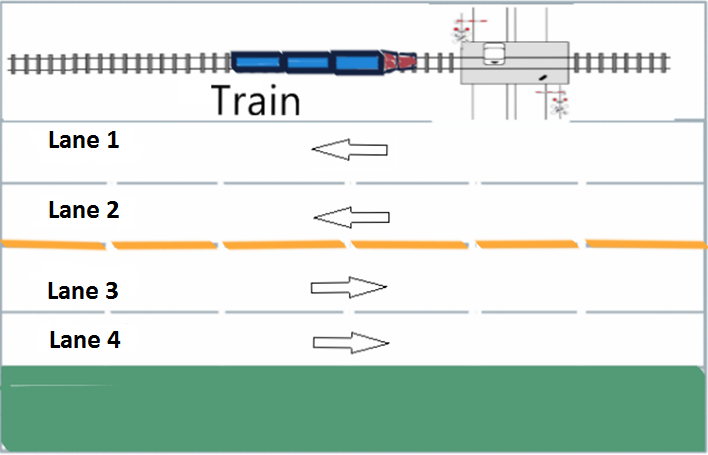
A major challenge in the system design is the development of a bi-directional notification interface between the system, the drivers on the road, and other public agencies. Additional considerations will be taking into account when implement changes in street rails and other traffic signs. These changes have to proceed smoothly to avoid confusions that could provoke accidents. When it comes to the web interface, this needs to be user-friendly and functional.

# Methodology and Design

The project consists on two modules:

* Street Rail Module
* Cross Road Module

Figure 1 shows the representation of both modules and their basic design:



**Figure 1. Road system representation.**

## Street Rails Module

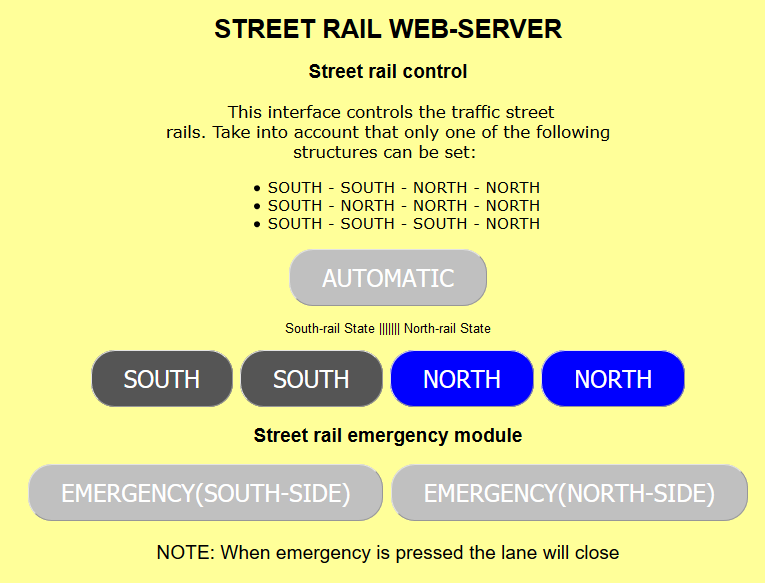
The Street Rails Module consists of four rails (Figure 1). The directions of the rail direction are identified as North or South. Their configuration, depending of the automatic traffic reading, and the manual web interface (Figure 2) can take the following structures:

1. SOUTH – SOUTH – NORTH – NORTH
2. SOUTH – NORTH – NORTH – NORTH
3. SOUTH – SOUTH – SOUTH – NORTH
4. SOUTH – CLOSED - NORTH – NORTH
5. SOUTH – SOUTH – CLOSED – NORTH
6. SOUTH – CLOSED – CLOSED – NORTH

The equipment used for the design of this module consists of:

* Arduino platforms such as MKR1000 and ESP32
* IR sensors
* LCD screens
* LEDs

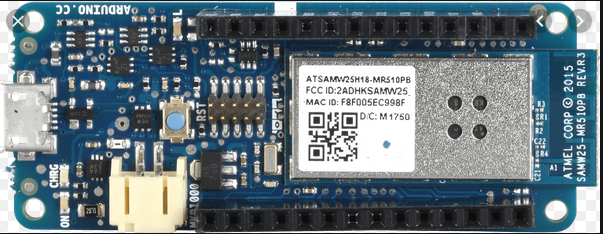
The web server interface (Figure 2) provides the possibility of manually control the rail’s directions. Besides, it will a “close for emergency” feature that will allow the shut of rails because of emergency vehicles approaching.



**Figure 2. WEB-SERVER interface for street rail module**

### **MKR 1000**

#### Features

 **Figure 1. MKR 1000 board**

The Arduino MKR1000 (Figure 1) is a powerful board that combines the functionality of an Arduino Zero and a handy WiFi shield. This thin board offers a practical and cost-effective solutions for the designing of IoT projects. Its design includes a LiPo (lithium polymer) charging circuit that allows the Arduino to run on battery power or external 5V, charging the LiPo battery while running on external power. Switching from one source to the other is done automatically. A good 32-bit computational power similar to the Arduino Zero board, the usual rich set of I/O interfaces, low-power WiFi with a Cryptochip for secure communication, and the ease of use of the Arduino Integrated Development Environment (IDE) for code development and programming. All of these features make this board the preferred choice for the emerging IoT battery-powered projects in a compact form factor. The USB port can be used to supply power (5V) to the board. The Arduino MKR1000 is able to run with or without the LiPo battery connected and has limited power consumption [5].

There are totally 15 digital Pins and 7 Analog pins on the MKR1000 board. These entire pins are only 3.3V compatible. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like pinMode() and digitalWrite() can be used to control their operation. The operating voltage is 0V and 3.3VV for digital pins. The analog pins can measure analog voltage from 0V to 3.3V using any of the 7 Analog pins using a simple function liken analogRead().

These pins apart from serving their purpose can also be used for special purposes which are:

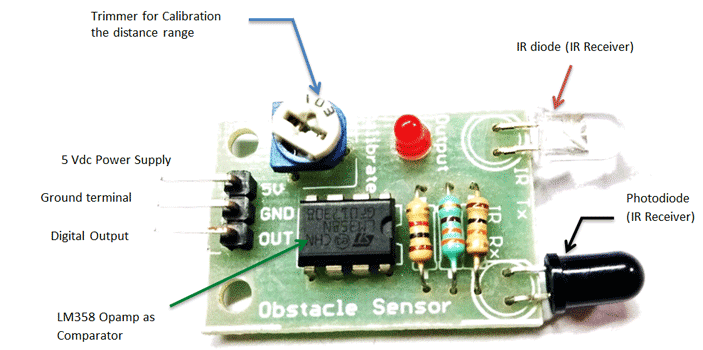
* **Serial Pins 13 (Rx) and 14 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data.
* **External Interrupt Pins:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* **PWM Pins:** These pins provide an 8-bit PWM output by using analogWrite() function.
* **SPI Pins:** These pins are used for SPI communication.
* **In-built LED Pin D6:** This pin is connected with a built-in LED, when pin 6 is HIGH – LED is on and when pin 6 is LOW, its off.
* **I2C Pins:** Used for IIC communication using Wire library.
* **AREF:** Used to provide reference voltage for analog inputs with analogReference() function.
* **Reset Pin:** Making this pin LOW, resets the microcontroller.

#### Use in the project

In our “Interactive Traffic System”, “Street Rail Module”, the MKR 1000 will be use to manage the IR sensors as well as LCDs and LEDs that will take care of the traffic leading interface. Four IR sensors are connected to four analog inputs reading the approximation of objects. In normal conditions (no objects), IR sensors reads values around 1000 that are translated to the inputs of the MKR1000. When they sense an object then the reads will go down around 300-500 depending of the distance of the approximation and the setting of the IR sensors. The MKR 1000 (analog inputs) are programed to react to those values, meaning the sensing of a vehicle. Using this procedure, they will build an internal variable to save the account of vehicles of every rail.

### **IR sensors**

##### Features



**Figure 3. IR sensor**

* 5VDC Operating voltage
* I/O pins are 5V and 3.3V compliant
* Range: Up to 20cm
* Adjustable Sensing range
* Built-in Ambient Light Sensor
* 20mA supply current
* Mounting hole

The IR sensor module consists mainly of the IR Transmitter and Receiver, Op-Amp, Variable Resistor (Trimmer pot), and an output LED [6].

**IR LED Transmitter**

[IR LED](https://components101.com/ir-led-pinout-datasheet) emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. few centimeters to several feet, it depends upon the type of IR transmitter and the manufacturer. Some transmitters have the range in kilometers [6].

**Photodiode Receiver**

Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection. Photodiode looks like a LED, with a black color coating on its outer side, the black color absorbs the highest amount of light [6].

**LM358 Op-Amp**

[LM358](https://components101.com/ic-lm358-pinout-details-datasheet) is an Operational Amplifier (Op-Amp) is used as voltage comparator in the IR sensor (Figure 3). The comparator will compare the threshold voltage set using the preset and the photodiode’s series resistor voltage [6].

Photodiode’s series resistor voltage drop > Threshold voltage = Op-Amp output is High

Photodiode’s series resistor voltage drop < Threshold voltage = Op-Amp output is Low

When Op-Amp's output is **high** the LED at the Op-Amp output terminal **turns ON** (Indicating the detection of Object) [6].

**Variable Resistor**

The variable resistor used here is a preset. It is used to calibrate the distance range at which object should be detected [6].

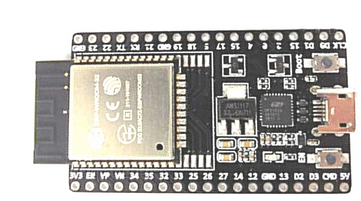
##### Use in the project

The Infrared sensors are mostly used on the MKR1000 platforms to count the number of the vehicles transiting the road over a set period of time. Once, the time is complied, the account will be wireless sent, by using MQTT protocol, to an ESP32 board which, together with the MKR1000 board, will proceed to manage street rails and signaling (LCD and LEDs) according to the traffic needs.

#### **ESP 32**

##### Features

While ESP8266 is still one of the most used WiFi development board in handmade IoT projects, the new ESP32 (Figure 4) is getting allot attention. Although it is not so new, the ESP32 popularity has begun to grow in the last year mostly because of its Bluetooth capabilities, but not only. The ESP32 comes now with almost 32 I/O pins and a powerful 32bits dual core CPU ready to process more complex jobs.



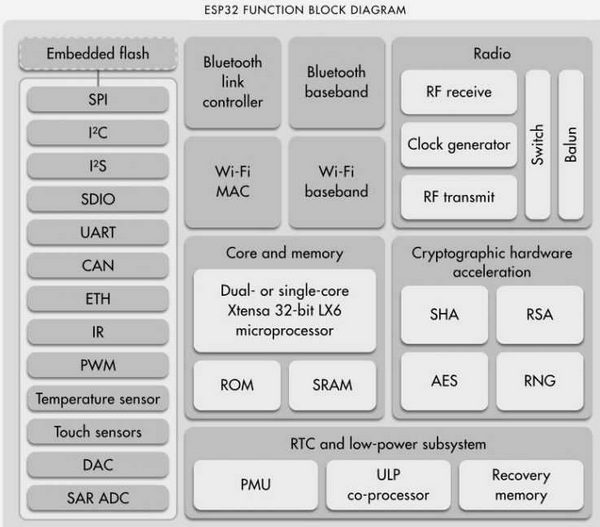
**Figure 4. ESP32 Board**

ESP32 technical specifications

Same as ESP8266 and older ESP modules, ESP32 can be found in various models with different hardware features. Below are some of the most common versions:

* **Processors:**
  + **Main:** Tensilica Xtensa 32-bit LX6 microprocessor
    - **Cores:** 2 or 1 (depending on variation) – All chips in the ESP32 series are dual-core except for ESP32-S0WD, which is single-core.
    - **Clock frequency:** up to 240 MHz
    - **Performance:** up to 600 DMIPS
  + [Secondary (ULP):](http://esp-idf.readthedocs.io/en/latest/api-guides/ulp.html) dedicated for ADC conversions, computation, and level thresholds while in deep sleep.
* **Wireless connectivity:**
  + **Wi-Fi:** 802.11 b/g/n/e/i (802.11n @ 2.4 GHz up to 150 Mbit/s)
  + **Bluetooth:** v4.2 BR/EDR and Bluetooth Low Energy (BLE)
* **Memory:**
  + **Internal memory:**
    - **ROM:** 448 KiB – For booting and core functions.
    - **SRAM:** 520 KiB – For data and instruction.
    - **RTC slow SRAM:** 8 KiB – For co-processor accessing during deep-sleep mode.
    - **RTC fast SRAM:** 8 KiB – For data storage and main CPU during RTC Boot from the deep-sleep mode.
    - **eFuse:** 1 Kbit – Of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID.
    - **Embedded flash: –** Flash connected internally via IO16, IO17, SD\_CMD, SD\_CLK, SD\_DATA\_0 and SD\_DATA\_1 on ESP32-D2WD and ESP32-PICO-D4.
      * 0 MB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips)
      * 2 MB (ESP32-D2WD chip)
      * 4 MB (ESP32-PICO-D4 SIP module)
  + **External flash & SRAM:** ESP32 without embedded flash supports up to 4 × 16 MB of external QSPI flash and 8 MB SRAM with hardware encryption based on AES to protect developer’s programs and data. ESP32 chips with embedded flash do not support the address mapping between external flash and peripherals.
* **Peripheral input/output:** Rich peripheral interface with DMA that includes capacitive touch, ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I²C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I²S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.
* **Security:**
  + IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI
  + Secure boot
  + Flash encryption
  + 1024-bit OTP, up to 768-bit for customers
  + Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

The ESP32 diagram block (Figure 5) gives a better overview of this board:



**Figure 5. Block Diagram of ESP32**

##### Use in the Project

The ESP32 is designed to receive the counter from the traffic sensors and manage part of the LCDs and LEDs to lead the drivers. Through the WiFi feature of this board, the ESP32 will use MQTT protocol to receive and transmit the messages wirelessly to and from the MKR1000. This board carries the configuration of the Web Server in the different configurations and structures of the manual interfaces

A web interface reflects the count coming from the sensors and will provide manual control of the street rails for emergency services such as fire fighters, police and ambulance to use. LCDs and LEDs will be used to communicate with drivers on the road

**Cross-road Module**

The second module of this project consists of a train crossroad with a signal controller which is triggered when trains approach. In addition, a sensor will be set on the crossroad to detect objects or vehicles. Working together, an emergency system will be implemented, which will send an alert to the train management interface when a train approach to the crossroad and a vehicles is stalled on the way. This will avoid possible accidents.

IR sensors will be used to detect the approaching train and the vehicles passing the crossroad. An ESP32 platform will be used for this configuration and will be responsible of wirelessly transmitting the result information to a MKR1000/ESP32 platform. Besides, the ESP 32 platform will interact with the management interface of the train for the emergency system. The receiving platform will control the cross-road arms and others components (LEDs, LCDs) responsible of interacting with drivers and pedestrians.

# Flow Chart

**Street Rails Module**



**Figure 1. Street Rail Module Flow Chart**

**Crossroad Module**



**Figure 2. Crossroad Module Flow Chart**

# Road Representation – PoC

# Timeline



**Figure 4. Project Timeline**

# Budget

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Material and Equipment | Quantity | Price |
| 1. Project Design & Development | Arduino Mega 2560 | 1 | 30.50 |
| Arduino MKR1000 | 2 | 76.20 |
| ESP 8266 | 2 | 22.00 |
| ESP 32 | 2 | 26.00 |
| Connectors and Electronic Components (Servo Motor, LCDs, LEDs, Relays, LED string, resistors) | 1 | 100.00 |
| 1. Buildup | Plywood, Paint, other construction materials | 1 | 150.00 |
| 1. Deliverable | Documentation & Project Closeout | 1 | 100.00 |
| **Total 504.70** | | | |

**Figure 5. Project Budget Worksheet**

# 

# References

1. <https://ny.curbed.com/2018/2/6/16979696/new-york-city-traffic-congestion-second-worst>
2. [www.instructables.com](http://www.instructables.com)
3. [www.github.com](http://www.github.com)
4. [www.arduino.cc](http://www.arduino.cc)
5. <https://www.sparkfun.com/products/14394>
6. <https://components101.com/sensors/ir-sensor-module>