

**Project Proposal**

**TCET 4182 – E333**

**“Smart Traffic Light”**

Participants:

Manuel Mane Penton

Nayib Ega

Telecommunication Technology Engineering

School: New York City College of Technology

Revised version: 02/18/2019

Contents

[Abstract 1](#_Toc1570612)

[Objectives 2](#_Toc1570613)

[Introduction 3](#_Toc1570614)

[Methodology and Design 3](#_Toc1570615)

[Timeline 7](#_Toc1570616)

[Budget 8](#_Toc1570617)

[Team members 9](#_Toc1570618)

[References 9](#_Toc1570619)

# Abstract

Many cities like New York suffer the effect of traffic congestions every day. Intersections are one of the main reasons for these traffic congestions. Traffic lights play a major role since their timing is fixed and do not adjust to traffic changing conditions. How many times a driver have been waiting in the red light while there is not traffic coming on the other side of the intersection? How often we have seen an emergency vehicle stuck in traffic because there is a line of cars ahead waiting on the red light?

The intended system tries to alleviate these congestions problems by making traffic lights adaptable to real-time traffic conditions. This will allow traffic to flow continuously in one direction or other based on the need. This can be achieved using existing technology such as, infrared sensors, Wi-Fi, GPS, cameras, and system controllers. Example, infrared sensors can detect traffic on the road and send a command to the traffic light via Wi-Fi to change the duration of the light. In addition, the system can interact with the cities first-responders units such as the NYFD whom could take control of the system and override the lights sequence.

Objectives

To build a traffic light system that is capable to adapt to traffic conditions and to adjust light patterns. Additionally, first responders will be able to communicate with the system and overwrite light sequence, if necessary, to clear their way.

Introduction

The purpose of this project is to develop a traffic light system that could adapt to real-time road conditions. The system will be able to detect traffic congestion and adjust the timing of the traffic lights. This will allow cars (drivers) to move in one direction or the other preventing possible traffic congestion. First responders will be able to communicate with the system wirelessly to overwrite the light sequence, if necessary. This capability will allow emergency personal to take control of the light signals of the upcoming intersection and to clear the traffic upon their pass.

The existing infrastructure is limited, it is fixed with no real capabilities. These systems worked very well in the past, but certainly it is not a match for today’s traffic pattern and the amount of cars on the streets. As it is known, traffic congestion is a major problem mostly in big cities like New York. According to a study released by the [Texas A&M Transportation Institute](http://tti.tamu.edu/) (TTI) and [INRIX](http://inrix.com/); in 2014, New Yorkers wasted an average of 74 hours in the year on the road. Congestion issues, especially at intersections, are not just time and money related but one of public safety. The number of fatal accidents involving emergency vehicles is increasing in The United States. There are 6500 accident involving ambulance, as per The National Highway Traffic Safety Administration (NHTSA) of which, 35% result in in injuries or fatality.

Traffic light controlled by microcontrollers, and using IR are becoming common in many cities because they can easily adjust for different time sequences. However, they work according an estimation made of the traffic control data it has collected in its system. So, in certain parts of the days or the week the traffic flow could change and the data collected by the traffic light system would bother a proper traffic flow. So, this idea will help to alleviate the amount of traffic jam by diminishing the amount of time that drivers spend waiting for the light at traffic lights, sometimes unnecessary due the change of the light when no cars are not coming by the other side of the intersection.

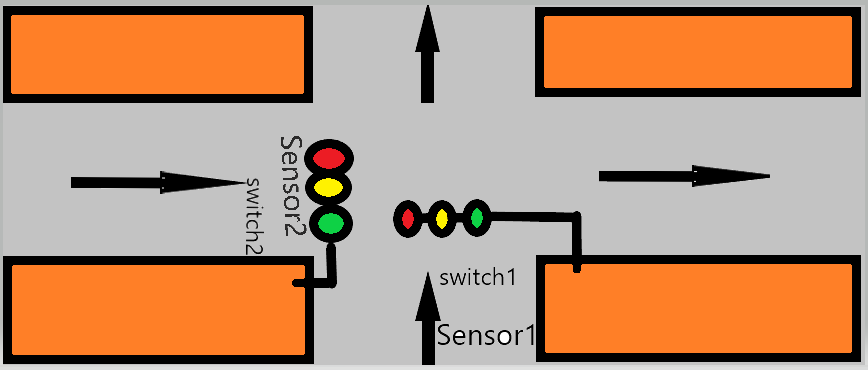
Methodology and Design

The goal is to design and develop a working prototype of a smart traffic light system. The system will be developed using an Arduino Mega 2560 platform, Arduino wireless module, IR sensors, and other electronic elements such as LEDs, resistors and LCD screen. Sensors are critical for the design. IR sensor will determine the presence of vehicles on the road. The 1 or 0 condition will determine the operation of the controller.

In addition to the hardware, the programming of the smart units will be achieved by using Arduino IDE 1.8.8. The code (sketch) and behavior of the traffic light system will consist of four different modules:

1. **Screen Module:** In this module presents the behavior of the screen LCD using an Arduino/Genuino MKR1000.
2. **Traffic Light Module:** Using the Arduino Mega 2500, the behavior is the following: When cars approximate any side of the street, and therefore will be sensed by the IR sensors, the traffic light will proceed to a state of working, and it will change its light (green) to give pass to the coming vehicles: This module is divided in three situations:
   1. **Situation A:** Vehicles are coming by one of the sides of the intersection: traffic light will proceed to access the pass (green light) to this side meanwhile there are not vehicles coming by the other side. If cars approach the other side of the street as well, the traffic lights will proceed to situation B; if vehicles are only coming by one side of the street, the traffic light will keep the green on until the situation changes. On the other side, if cars stop coming from both sides of the intersection, the traffic lights condition will go back to Null Module.
   2. **Situation B:** Vehicles coming from both sides of the intersection: In this case, the traffic lights will proceed to access pass to both sides of the intersection (normal condition) in a timely-equal way. If cars stop coming by one side of the intersection, the traffic lights will proceed to situation A and they will only give pass to one side of the intersection until the situation changes. It vehicles stop coming from both sides of the intersection, the traffic lights will go to Null Module.
   3. **Situation C:** When there are not vehicles coming by any side of the street, the traffic lights will acquire a null state, where both traffic lights will be blinking the yellow light with a two seconds timer.
3. **Emergency Module:** Through the Arduino Wireless Module (Arduino/Genuino MKR1000), emergency vehicles will send a signal to the traffic light system and the switch (Switch1 and Switch2) will activate. In this case, the emergency mode will activate, and will overwrite the traffic light systems. The traffic light system will proceed to give access to the side of the intersection the emergency vehicle is coming to clean its way through.
4. **WiFi Module:** Using the Arduino/Genuino MKR1000, with a WiFi program, the emergency vehicles will be able to send a signal when they are approaching the intersection. Once the signal is sent, this board will communicate with the Arduino Mega 2500 and the Emergency Module (emergency state) will activate overwriting the rest of the traffic light system.

Figure1 shows a scheme of the intersection.



*Figure 1: Traffic Light scheme.*

Flow Chart (figure2) summarizes the proceeding of the smart traffic light.

The following table: Table1 “Truth Table of the smart traffic light” describes the states which the system goes through.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| States | Inputs | | | | Outputs | | | | | | Description |
|  | Switch1 | Switch2 | Sensor1 | Sensor2 | Road 1 | | | Road 2 | | |  |
| R | Y | G | R | Y | G |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Null Mode. Working Module:  Situation C. |
| 2 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | Working Module: Situation A: Vehicles coming from road 2. |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | Working Module: Situation A: Vehicles coming from road 1. |
| 4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | Working Module: Situation B  (normal condition) |
| 5 | 0 | 1 | x | x | 1 | 0 | 0 | 0 | 0 | 1 | Emergency Module:  Emergency vehicle coming  from road 2. |
| 6 | 1 | 0 | x | x | 0 | 0 | 1 | 1 | 0 | 0 | Emergency Module:  Emergency vehicle coming  From road 1. |
| 7 | 1 | 1 | x | x | 0 | 0 | 1 | 1 | 0 | 0 | Emergency vehicle coming  From both roads. In this case  The system will give access  First to road 1, and later to  Road 2. (Similar to normal  Condition) |

*Table 1: “Truth Table of Smart Traffic Light”*

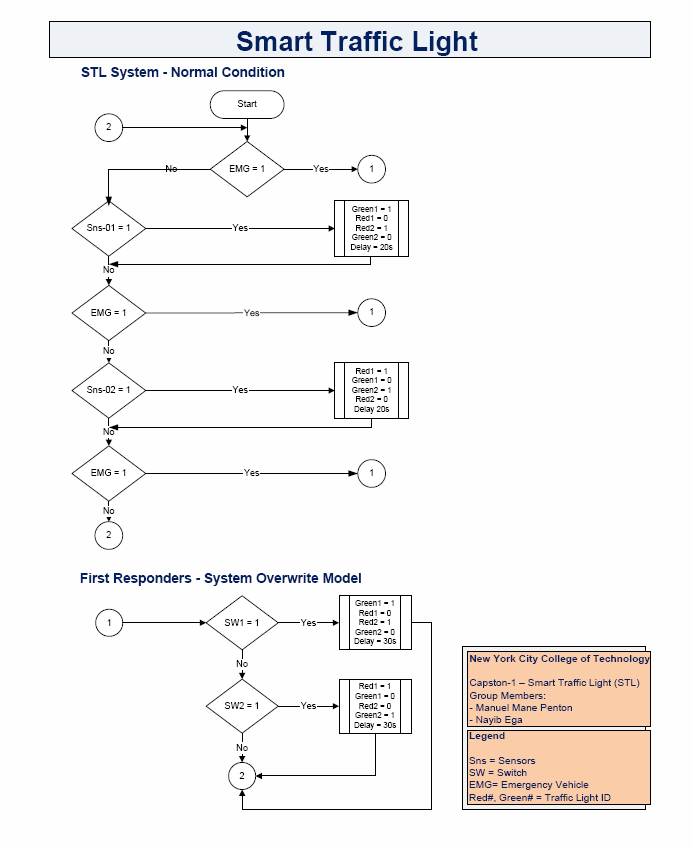
Flowchart

Figure2: “Flow Chart”

Program

**Library:**

#include <LiquidCrystal.h>

**Variables and function prototypes code:**

//Variables

int redled1 = 3;

int yellowled1 = 4;

int greenled1 = 5;

int redled2 = 7;

int yellowled2 = 8;

int greenled2 = 9;

int emergency1 = 50;

int emergency2 = 52;

//Funtion prototype

void trafficlight1();

void trafficlight2();

void normalcondition1();

void normalcondition2();

void notraffic();

void emergency();

void LCD();

**Setup code:**

void setup()

{

// put your setup code here, to run once:

pinMode (3, OUTPUT);

pinMode (4, OUTPUT);

pinMode (5, OUTPUT);

pinMode (7, OUTPUT);

pinMode (8, OUTPUT);

pinMode (9, OUTPUT);

pinMode (A0, INPUT);

pinMode (A3, INPUT);

lcd.begin(16,2); //dimension of the LCD

Serial.begin(9600); }

**Main function code:**

void loop()

{ Serial.println(analogRead(A0)); //print voltage of output pin (resolution 1-1024

Serial.println(analogRead(A3)); //print voltage of output pin (resolution 1-1024

//delay (1000); //wait a second

emergency();

if (analogRead(A3) < 800 && analogRead(A0) > 800)

{

//function for side 1 ON and side 2 OFF

trafficlight1();

}

else if ((analogRead(A3) > 800 && analogRead(A0) < 800))

{

trafficlight2();

}

else if ((analogRead(A3) < 800 && analogRead(A0) < 800))

{

normalcondition1();

normalcondition2();

}

else

{

notraffic();

}

}

**Functions**

**Emergency function:** It describes the behavior of the traffic light depending of which side of the intersection the emergency vehicle is coming from (Emergency Module).This function includes the LCD function later explained.

void emergency()

{

emergency1 = digitalRead(30);

emergency2 = digitalRead(34);

Serial.print("Emergency mode on\n");

if (emergency1 == HIGH)

{

LCD();

Serial.print("Emergency 1 mode on\n");

digitalWrite(greenled1, 0);

digitalWrite(greenled2, 0);

digitalWrite(redled1, 0);

digitalWrite(redled2, 0);

digitalWrite(yellowled1, 1);

digitalWrite(yellowled2, 1);

delay(1000);

digitalWrite(yellowled1, 0);

digitalWrite(yellowled2, 0);

delay(1000);

digitalWrite(yellowled1, 1);

digitalWrite(yellowled2, 1);

delay(1000);

digitalWrite(yellowled1, 0);

digitalWrite(yellowled2, 0);

delay(1000);

digitalWrite(yellowled1, 1);

digitalWrite(yellowled2, 1);

delay(1000);

digitalWrite(yellowled1, 0);

digitalWrite(yellowled2, 0);

delay(1000);

digitalWrite(greenled1, 1);

digitalWrite(redled2, 1);

delay(10000);

digitalWrite(greenled1, 0);

delay(500);

digitalWrite(greenled1, 1);

delay(500);

digitalWrite(greenled1, 0);

delay(500);

digitalWrite(yellowled1, 1);

delay(1000);

digitalWrite(yellowled1, 0);

delay(1000);

digitalWrite(redled1, 1);

delay(2000);

digitalWrite(redled1, 0);

digitalWrite(redled2, 0);

}

else if (emergency2 == HIGH)

{ LCD();

Serial.print("Emergency mode 2 on\n");

digitalWrite(greenled1, 0);

digitalWrite(greenled2, 0);

digitalWrite(redled1, 0);

digitalWrite(redled2, 0);

digitalWrite(yellowled2, 1);

digitalWrite(yellowled1, 1);

delay(1000);

digitalWrite(yellowled2, 0);

digitalWrite(yellowled1, 0);

delay(1000);

digitalWrite(yellowled2, 1);

digitalWrite(yellowled1, 1);

delay(1000);

digitalWrite(yellowled2, 0);

digitalWrite(yellowled1, 0);

delay(1000);

digitalWrite(yellowled2, 1);

digitalWrite(yellowled1, 1);

delay(1000);

digitalWrite(yellowled2, 0);

digitalWrite(yellowled1, 0);

delay(1000);

digitalWrite(greenled2, 1);

digitalWrite(redled1, 1);

delay(10000);

digitalWrite(greenled2, 0);

delay(500);

digitalWrite(greenled2, 1);

delay(500);

digitalWrite(greenled2, 0);

delay(500);

digitalWrite(yellowled2, 1);

delay(1000);

digitalWrite(yellowled2, 0);

delay(1000);

digitalWrite(redled2, 1);

delay(2000);

digitalWrite(redled2, 0);

digitalWrite(redled1, 0);

}

if(emergency1 == HIGH || emergency2 == HIGH)

emergency(); }

**Function traffic light1 and traffic light2:** This functions describes the behavior of the traffic light system when vehicles are approaching side of the intersection number 1 and number 2 depending on what the sensors read (Working Module, Situation A). It includes the emergency function as well.

void trafficlight1()

{

digitalWrite (redled2, 1); //Initial condition

digitalWrite (redled1, 0); //Light traffic 1

digitalWrite (greenled1, 1);

do //read the sensors to set a condition

{

emergency1 = digitalRead(30);

emergency2 = digitalRead(34);

Serial.println(analogRead(A0));

Serial.println(analogRead(A3));

digitalWrite (greenled1, 1);

}while (analogRead(A3) < 800 && analogRead(A0) > 800 &&

emergency1==LOW && emergency2 == LOW);

emergency();

delay (6000);

digitalWrite (redled1, 0);

digitalWrite (greenled1, 1);

digitalWrite (greenled1, 0);

delay (500);

digitalWrite (greenled1, 1);

delay (500);

digitalWrite (greenled1, 0);

delay (500);

digitalWrite (greenled1, 1);

delay (500);

digitalWrite (greenled1, 0);

delay (500);

digitalWrite (yellowled1, 1);

delay (3000);

digitalWrite (yellowled1, 0);

delay (200);

emergency();

digitalWrite (redled2, 1);

digitalWrite (redled1, 1);

delay (1000);

}

void trafficlight2()

{

digitalWrite (redled1, 1);

digitalWrite (redled2, 0); //Normal condition of traffic light 2

do //read the sensors to set a condition

{

emergency1 = digitalRead(30);

emergency2 = digitalRead(34);

Serial.println(analogRead(A0));

Serial.println(analogRead(A3));

digitalWrite (greenled2, 1);

}while (analogRead(A3) > 800 && analogRead(A0) < 800

&& emergency1==LOW && emergency2 == LOW);

emergency();

digitalWrite (redled1, 1);

digitalWrite (greenled2, 1);

delay (6000);

digitalWrite (greenled2, 0);

delay (500);

digitalWrite (greenled2, 1);

delay (500);

digitalWrite (greenled2, 0);

delay (500);

digitalWrite (greenled2, 1);

delay (500);

digitalWrite (greenled2, 0);

delay (500);

digitalWrite (yellowled2, 1);

delay (3000);

digitalWrite (yellowled2, 0);

delay (200);

emergency();

digitalWrite (redled1, 1);

digitalWrite (redled2, 1);

delay (1000);

}

**Function normalcondition:** They describe the behavior of the traffic light system when vehicles are coming by both sides of the intersection at the same time (Working Module, Situation B). It includes the emergency function as well.

void normalcondition1()

{

digitalWrite (redled2, 1); //Initial condition

digitalWrite (redled1, 0); //Light traffic 1

digitalWrite (greenled1, 1);

emergency();

digitalWrite (greenled1, 1);

digitalWrite (redled2, 1);

delay (6000);

digitalWrite (greenled1, 0);

delay (500);

digitalWrite (greenled1, 1);

delay (500);

digitalWrite (greenled1, 0);

delay (500);

digitalWrite (greenled1, 1);

delay (500);

digitalWrite (greenled1, 0);

delay (500);

digitalWrite (yellowled1, 1);

delay (3000);

digitalWrite (yellowled1, 0);

delay (200);

digitalWrite (redled1, 1);

emergency();

delay (1000);

}

void normalcondition2()

{

digitalWrite (redled1, 1);

digitalWrite (redled2, 0); //Normal condition of traffic light 2

digitalWrite (greenled2, 1);

delay (6000);

emergency();

digitalWrite (greenled2, 0);

delay (500);

digitalWrite (greenled2, 1);

delay (500);

digitalWrite (greenled2, 0);

delay (500);

digitalWrite (greenled2, 1);

delay (500);

digitalWrite (greenled2, 0);

delay (500);

digitalWrite (yellowled2, 1);

delay (3000);

digitalWrite (yellowled2, 0);

delay (200);

digitalWrite (redled2, 1);

delay (1000);

}

**Notraffic function:** It describes the behavior of the traffic light system when there are not vehicles coming by any side of the intersection (Working Module, Situation C). It includes the emergency function as well.

void notraffic()

{

digitalWrite(greenled1, 0);

digitalWrite(greenled2, 0);

digitalWrite (redled2, 0);

digitalWrite (redled1, 0);

digitalWrite (yellowled1, 1);

digitalWrite (yellowled2, 1);

emergency();

delay (2000);

digitalWrite (yellowled2, 0);

digitalWrite (yellowled1, 0);

emergency();

delay (2000);

}

**LCD Function:** It describes the setting of the LCD screen when emergency vehicles are coming into the intersection (Screen Module).

void LCD()

{

// put your main code here, to run repeatedly:

lcd.setCursor(0,0);

lcd.print("EMG Vehicle");

lcd.setCursor(0,1);

lcd.print("Approaching");

delay (1000);

lcd.setCursor(0,0);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print(" ");

delay(1000);

lcd.setCursor(0,0);

lcd.print("PLEASE");

lcd.setCursor(0,1);

lcd.print("CLEAR THE VIA");

delay(1000);

lcd.setCursor(0,0);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print(" ");

delay (1000);

LCD();

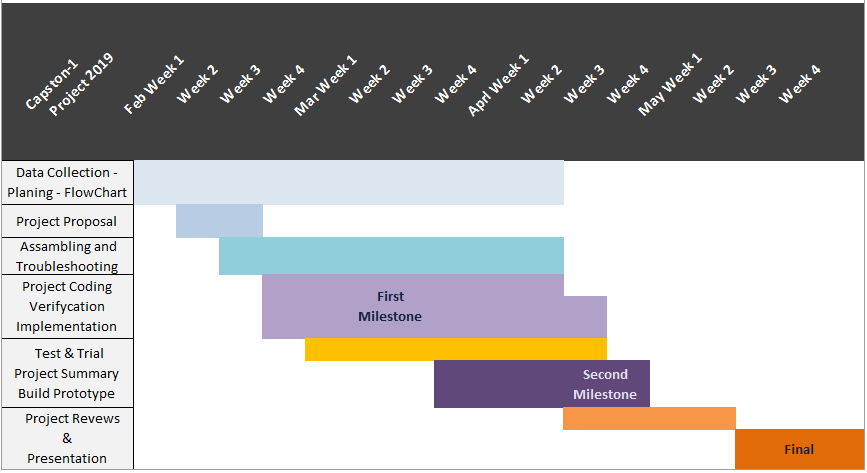
LCD();

LCD();

LCD();

}

Timeline

****

**Figure 3. Timeline**

Budget



Team members

* Nayib Ega: Basic knowledge C++
* Manuel Mane: Basic skills in Java and C++ procreation.

References

# Sain Nazir, Syed. “Traffic Signal Control Project using Arduino.” https://www.theengineeringprojects.com/2015/09/traffic-signal-control-project-using-arduino.html

* Bhalerao, Ashish, 2015. “Adaptive Traffic Light Control System.” http://www.iosrjournals.org/iosr-jece/papers/Vol.%2010%20Issue%205/Version-1/J010515361.pdf
* “Statistics on Emergency Vehicle Accidents in the United States.” (02/15/2018) https://www.arnolditkin.com/personal-injury-blog/2018/february/statistics-on-emergency-vehicle-accidents-in-the/