Density-based Arduino's traffic light and barrier system

By

Foysal Hasan Rafi [201014058]
Zeenat Akter Mim [201014098]
Md Eunusur Rahaman Bhuiyan [181014117]

Abstract

Traffic problems have become one of the most challenging issues in Bangladesh, particularly at road junctions. The traffic police are assigned to most of the road junctions in an effort to mitigate these problems. This project "Density-based Arduino's traffic light and barrier system" proposes a very simple but very effective solution to mitigate use of human resources on such junctions. It provides automatic control of the light and the barrier for traffic. Our microcontroller: Arduino, is equipped with switches as sensors. Based on the input from those switches, the system gets activated. In addition to reducing traffic congestion, this project is extremely cost-effective and could automate every junction in our country.

Introduction

The Ministry of Road Transport and Bridges has been working with the aim of improving traffic systems and regulations continuously. As the number of vehicles increases, congestion is occurring in places where there was no traffic before. In our project we focus on the junction of T-shape. Sometimes, traffic jams at T-shape junctions are so dense that it is difficult for connected roads' cars to enter the main road. Here most traffic policemen's responsibility is to allow vehicles from linked roads (figure 01) to enter a main road by stopping the traffic on the main road. Our strategy to automate this process is to use a barrier and traffic light by analyzing the density of vehicles using switches as sensors to detect the density of vehicles on the road.

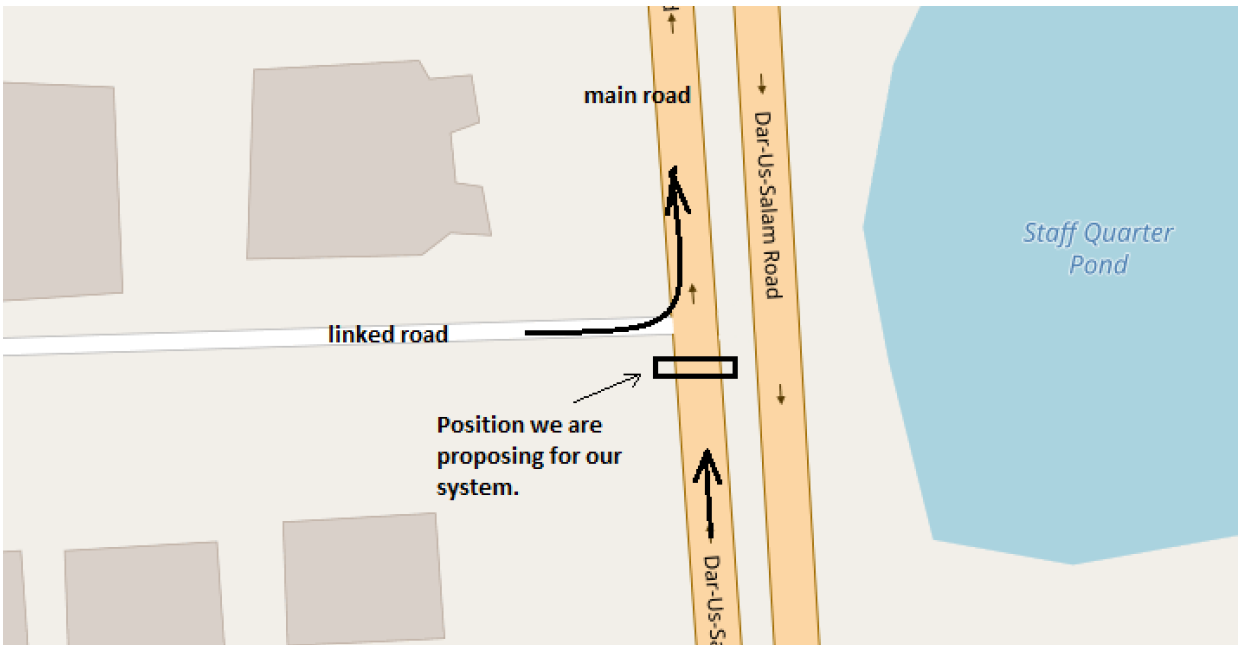


Figure 01: A T-shaped junction at Mirpur, Dhaka.

Theory

The system mainly depends on the number of push buttons implemented on a road. It is important to note that the more switches on the road, the more accurate is the traffic density. But we have used only 3 switches in our experiment. The theory of the system is, it always remains at case03. Even if there are few cars on the road, it still stays in case03. However, when the car density increases a little bit and a small traffic jam is created in front of the traffic light and barrier, the system will activate case02. Last but not least, case03, the system enables this option when there is a lot of traffic on the road.

Case 01:	it will go from yellow to green then red for 20 minute. Then again go yellow and then green.
Case 02:	it will go from yellow to green then red for 08 minute. Then again go yellow and then green.
Case 03:	in this case the green light will always be turned on.

		traffic light	t1	t2	t3	Details of the case
case01	switch1: on	red	off	off	on	Stays red for 20 second
	switch2: on	yellow	off	on	off	
	switch3: on	green	on	off	off	
case02	switch1: on	red	off	off	on	Stays red for 10 second
	switch2: on	yellow	off	on	off	
	switch3: off	green	on	off	off	
case03	Anything except case01 or case02	red	off	off	off	Always stays Green And barrier
		yellow	off	off	off	
		green	on	on	on	open

Implementation

The program is developed in tinkercad.com, it is an online Arduino simulator. These programs have the role of receiving data from the user or from the computer interfaces with the process, processing them and then displaying, storing or transmitting them remotely.

The Arduino contains the following components

1. Power Jack Input (5V to 13V input)
2. Power and Ground Pins (Vin, 5V, 3.3V, GND)
3. Digital Input Pins (pin 8– 10)
4. Digital Output (pin 4-7)
5. Arduino Microcontroller direct programming input
6. Digital Input and Output Ports (0 – 13)

7. Arduino Platform Reset button
8. USB input Jack for PC connection

The implementation was carried out with an Arduino Uno. Arduino is an open-source platform based on easy-to-use hardware and software. Open Source, meaning the user has the right to modify the software as he/she pleases. Due to its ease of use and low chances of connection errors, we decided to implement our project using Tinkercad.

As a first step, in the Arduino we connected the sensor (push button switch) to digital inputs 8,9,10. With these inputs, the sensor signals are transferred to the Arduino board, which then analyzes the density of cars based on those signals. Then it outputs via pin number 4, which is for the barrier, and pin numbers 5,6 and 7, which is for the traffic lights.

The entire process is accomplished by the following code:

```
/*
 these three represent three push button switch
 indicating number of vehicles on the road;
 SWITCH 01 is connected to pin number 10
 SWITCH 02 is connected to pin number 9
 SWITCH 03 is connected to pin number 8
*/
#include <Servo.h>
// these three represent traffic lights;
int GREEN = 5;
int YELLOW = 6;
int RED = 7;

Servo myservo; // Create servo object called "myservo"

void setup(){
  myservo.attach(4); //attach the servo at pin 4
  pinMode(GREEN, OUTPUT);
  pinMode(YELLOW, OUTPUT);
  pinMode(RED, OUTPUT);
  pinMode(8, INPUT);
  pinMode(9, INPUT);
  pinMode(10, INPUT);
}
```

```
void loop(){
  //read the inputs from three sensors
  int SWITCH03=digitalRead(8);
  int SWITCH02=digitalRead(9);
  int SWITCH01=digitalRead(10);

  //control barrier and traffic light for the 3 cases
  if(SWITCH03==1 && SWITCH02==1 && SWITCH01==1)
  {
    forSixCar();
  }
  else if(SWITCH03==1 && SWITCH02==1 && SWITCH01==0)
  {
    forFourCar();
  }
  else
  {
    green_light();
    myservo.write(0); //open the barrier | let car pass
  }
}

void forFourCar(){
  yellow_light();
  delay(2000);
  red_light();
  myservo.write(90); //close the barrier | don't let car pass
  delay(8000); // wait for 08 minute
  yellow_light();
  delay(2000);
  myservo.write(0); //open the barrier | let car pass
  green_light();
}

void forSixCar(){
  yellow_light();
  delay(2000);
  red_light();
  myservo.write(90); //close the barrier | don't let car pass
  delay(20000); // wait for 20 minute
  yellow_light();
  delay(2000);
  myservo.write(0); //open the barrier | let car pass
  green_light();
}
```

```
void green_light(){
  digitalWrite(GREEN, HIGH);
  digitalWrite(YELLOW, LOW);
  digitalWrite(RED, LOW);
}

void yellow_light(){
  digitalWrite(GREEN, LOW);
  digitalWrite(YELLOW, HIGH);
  digitalWrite(RED, LOW);
}

void red_light(){
  digitalWrite(GREEN, LOW);
  digitalWrite(YELLOW, LOW);
  digitalWrite(RED, HIGH);
}
```

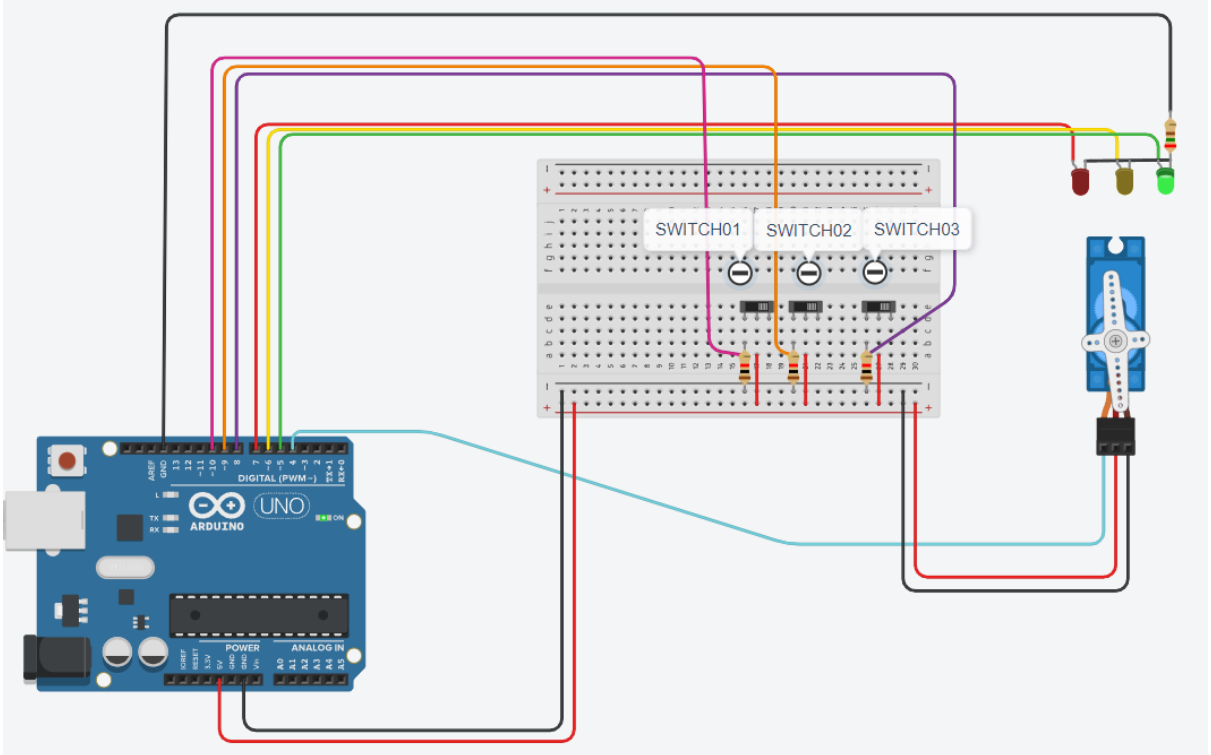
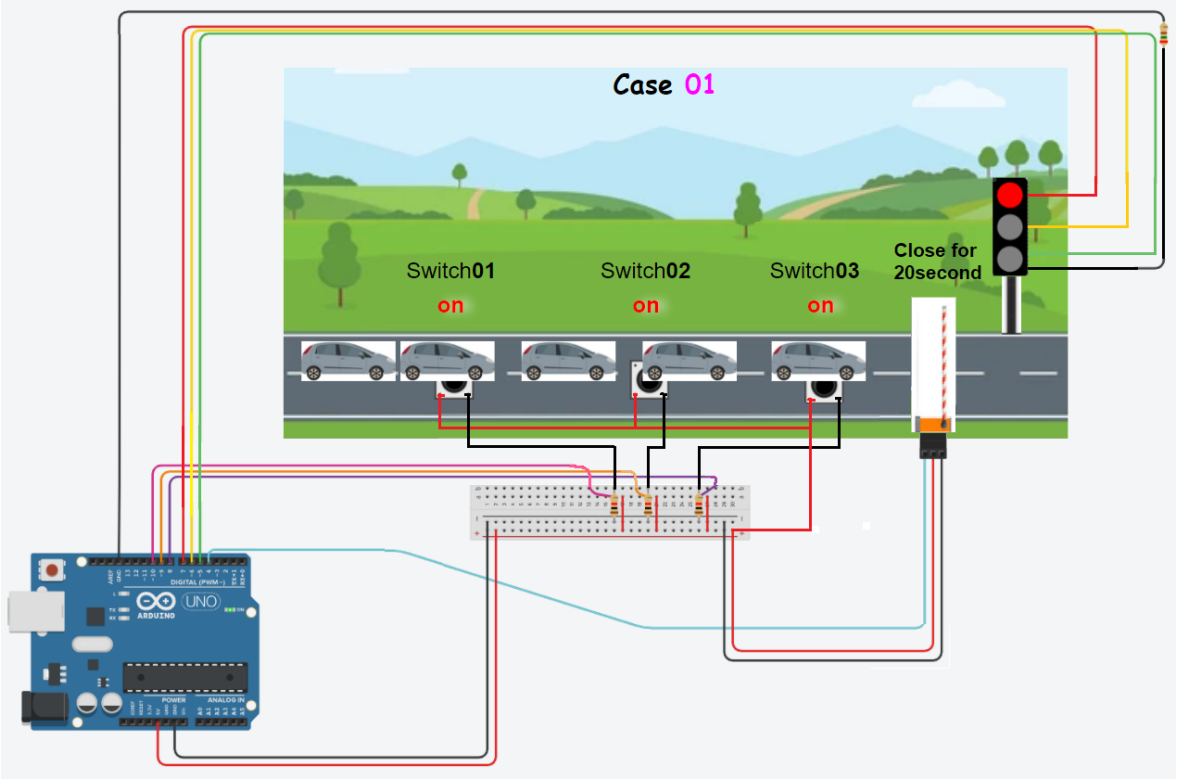


Figure 02: circuit design in tinkercad: case03 | barrier open and traffic light green

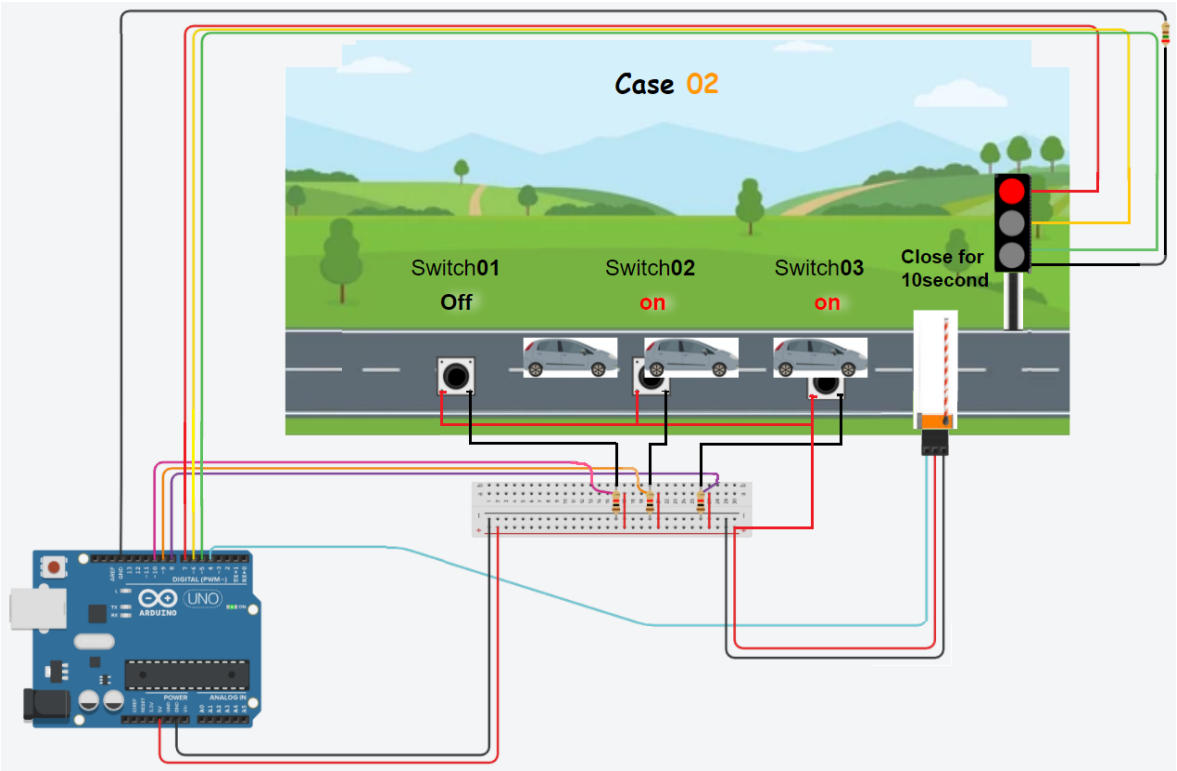
Experimental Results

The results of the proposed model can be categorized into three different cases based on four different scenarios.

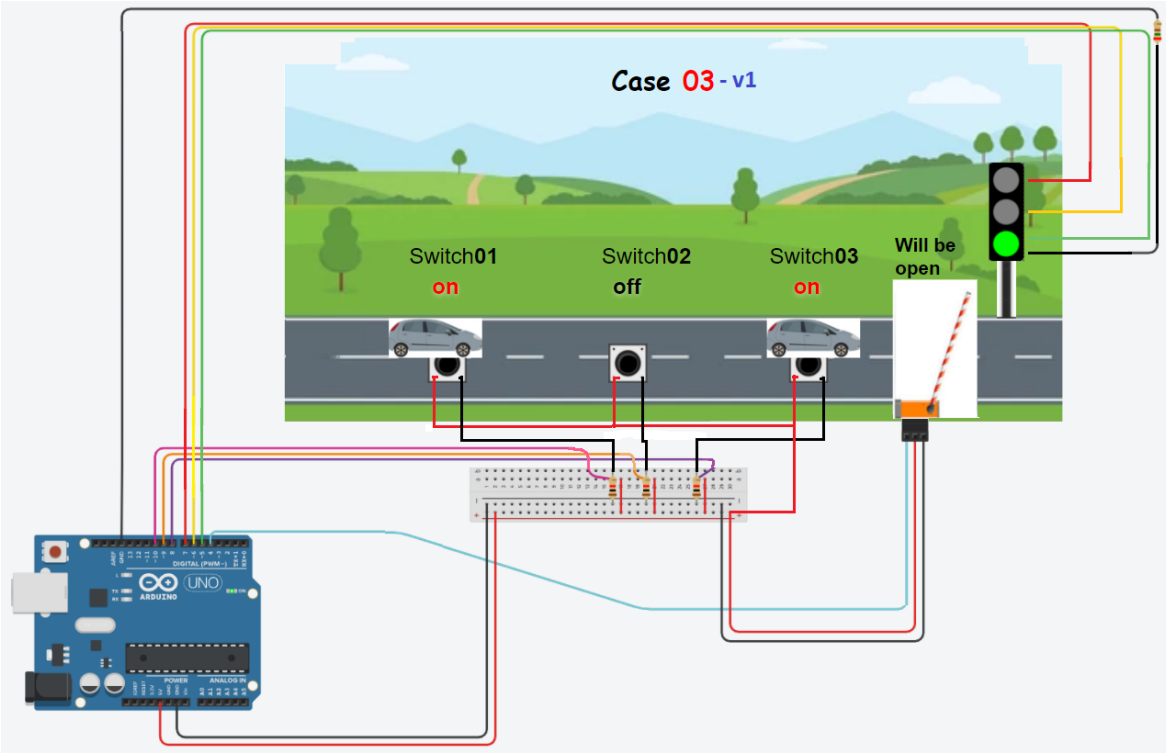
1. The signal and barrier system functions normally when there are heavy vehicles on the road. The barrier stays closed and the light turns from green to yellow, then red for almost 20 seconds.

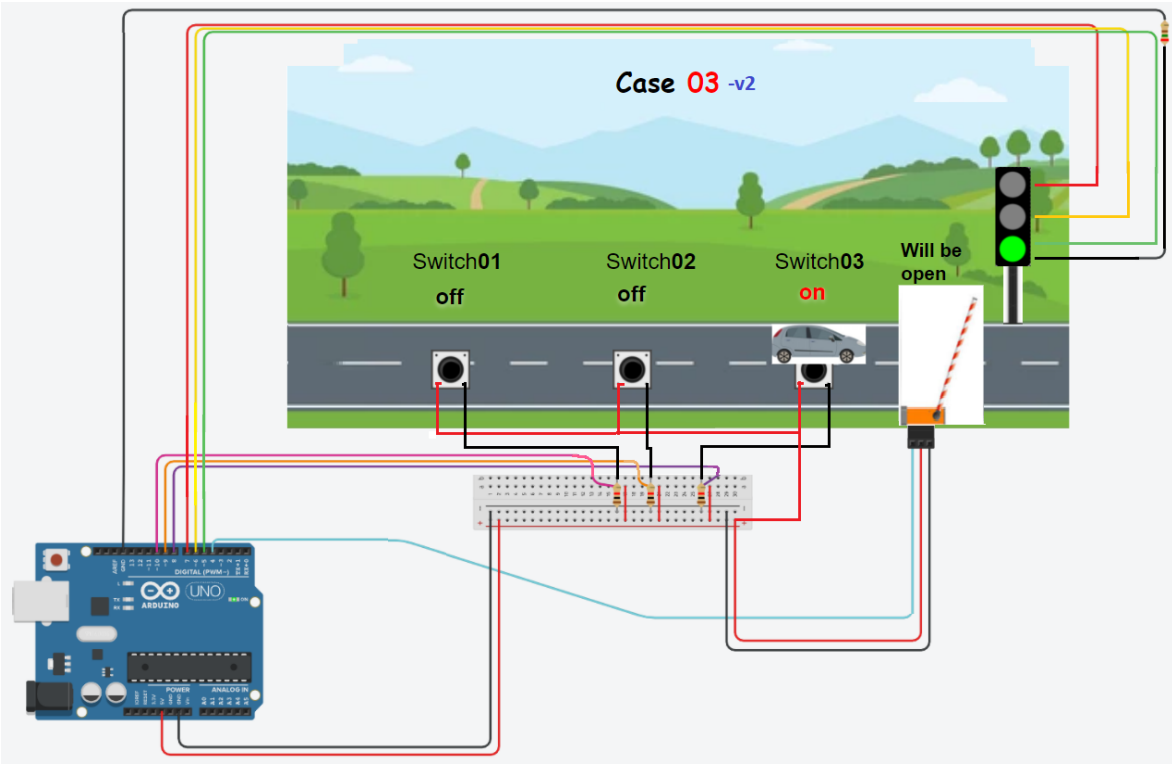


2. When there are moderate amounts of vehicles on the road, the signal and barrier system operates normally. The barrier stays closed and the light turns from green to yellow, then red for almost 10 seconds.



3. And for the last case, both scenarios imply that there is no traffic jam on the road, so the traffic light stays green and the barrier open.





Conclusion

The following traffic system is pretty much an upgrade to a basic traffic light system. With the various advancement of the world every little upgrade is significant for the well being of a society. The proposed system's aim is to allow vehicles from linked roads to access highways/main-Road without human intervention. Since this system is not that much intelligent it will not be able to let ambulance/emergency vehicles pass the barrier. However, with the help of A.I. For future development, we can set up cameras around traffic lights that can detect those vehicles and traffic around them, and control lanes, roads, or highways more accurately for a completely safe journey. For a better future.

References

- Traffic Congestion in Bangladesh- Causes and Solutions: A study of Chittagong Metropolitan City(repec.org)
- (PDF) DENSITY BASED TRAFFIC CONTROL SYSTEM (researchgate.net)