PES UNIVERSITY

MPCA Mini Project

Project Title :Density based traffic control system with emergency vehicle assistance

Section : B

Students Name :Ajitesh Nair,Ishan Padhy,G Ghana Gokul

SRN :PES2201800681,PES2201800158,PES2201800077

Problem Statement

As the population grows, the number of vehicles on the roads are also exponentially increasing, which results in increase in road accidents and traffic congestion. Specifically, when an emergency vehicle such as Ambulance or Fire engine gets stuck in traffic jam, saving the human life becomes difficult.

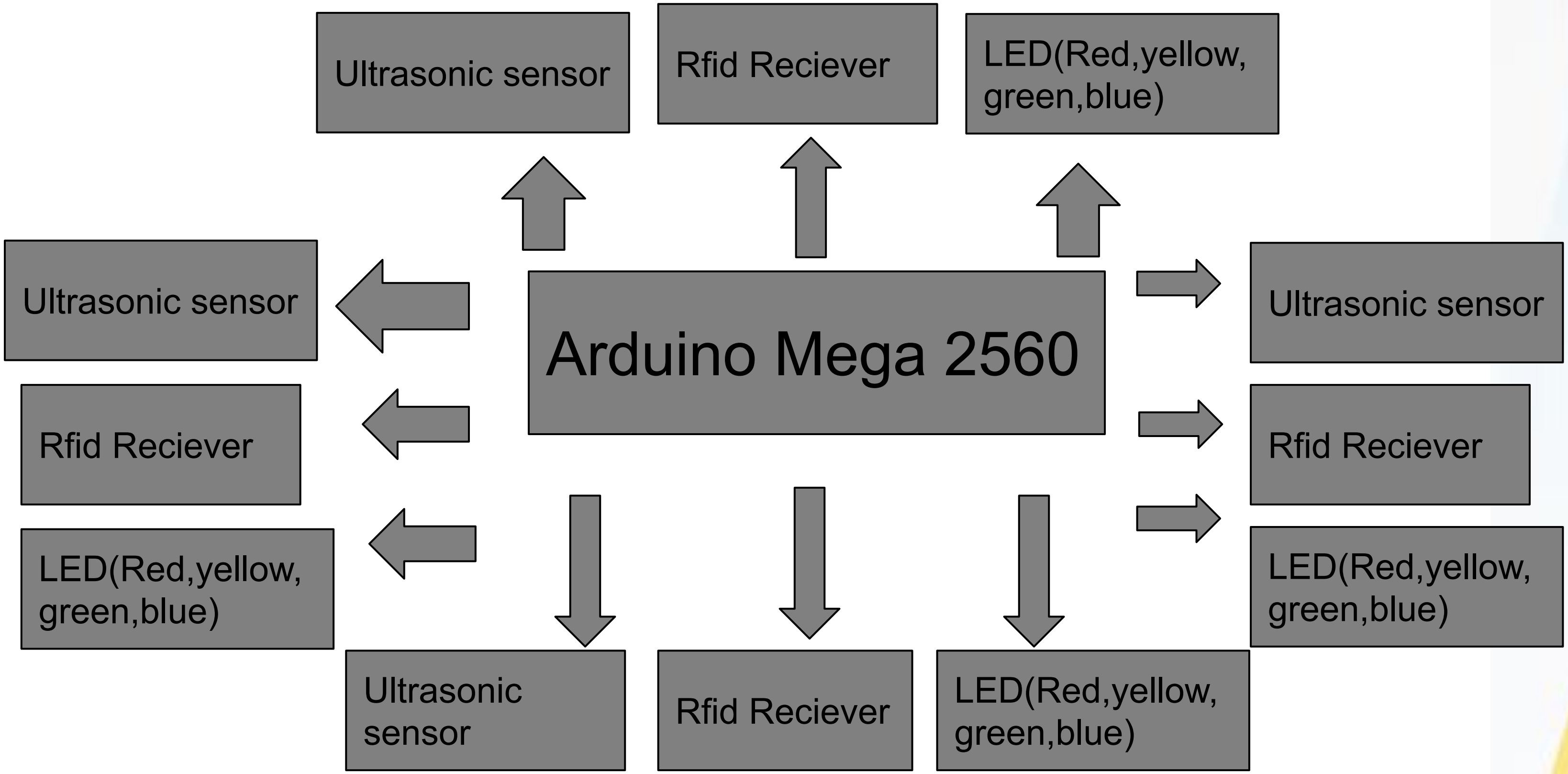
Under such circumstances, a promising system which can clear the traffic congestions especially in peak hours and thereby providing a safe path for emergency vehicles is very much essential. In the existing literature, less focus is given towards the problem of providing a clear path for emergency vehicles during traffic congestions.

Introduction

To solve these issues, an Ultrasonic sensor and RFID-based system is proposed, which manages and regulates the traffic signals at junctions when the emergency vehicle arrives, by allowing the easy passage out of the traffic congestions. The proposed frame work is modelled by means of an experimental setup using ARDUINO and LEDs which simulates a real time traffic scenario. Ultrasonic sensors are installed on the roads to manage the traffic efficiently.RFID chips are installed in emergency vehicles.

The ultrasonic sensor which is placed at a threshold distance from the junction calculates the vehicles density. This density is used by ARDUINO to regulate the traffic. The RFID receiver is also placed at a threshold distance from the junction. The RFID receiver informs the ARDUINO about the arrival of the emergency vehicle. The ARDUINO then takes the required measures to allow a safety passage for the emergency vehicle.

Block Diagram/Circuit Diagram



Ultrasonic sensor

Rfid Reciever

LED(Red,yellow, green,blue)

Ultrasonic sensor

Arduino Mega 2560

Ultrasonic sensor

Rfid Reciever

Rfid Reciever

LED(Red,yellow, green,blue)

LED(Red,yellow, green,blue)

Ultrasonic sensor

Rfid Reciever

LED(Red,yellow, green,blue)

Required Components

- Arduino Mega 2560
- 4 Ultrasonic sensors HC-SR04
- LEDS(4 Red,4 Yellow,4 Green,4 Blue)
- 4 RFID transmitter and receiver MFRC 522
- Jumper Wires
- Resistors

Project Explanation

The working of the project is divided into four steps

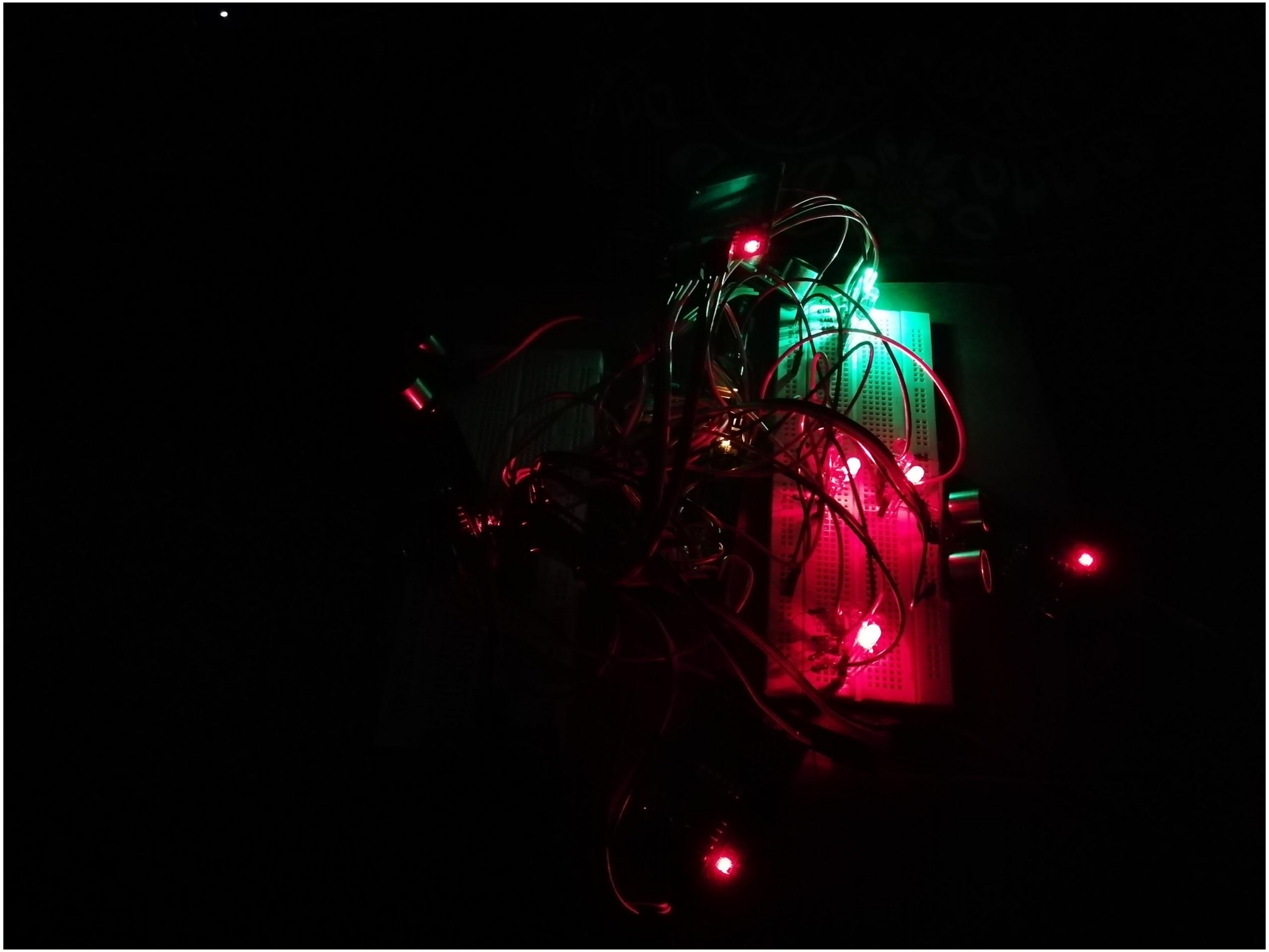
- If there is traffic at all the signals, then the system will work normally by controlling the signals one by one.
- If there is no traffic near a signal, then the system will skip this signal and will move on to the next one. For example, if there is no vehicle at signal 2, 3 and currently the system is allowing vehicles at signal 1 to pass. Then after signal 1, the system will move on to signal 4 skipping signal 2 and 3.
- If there is no traffic at all the 4 signals, system will stop at the current signal and will only move on the next signal if there will be traffic at any other signal.

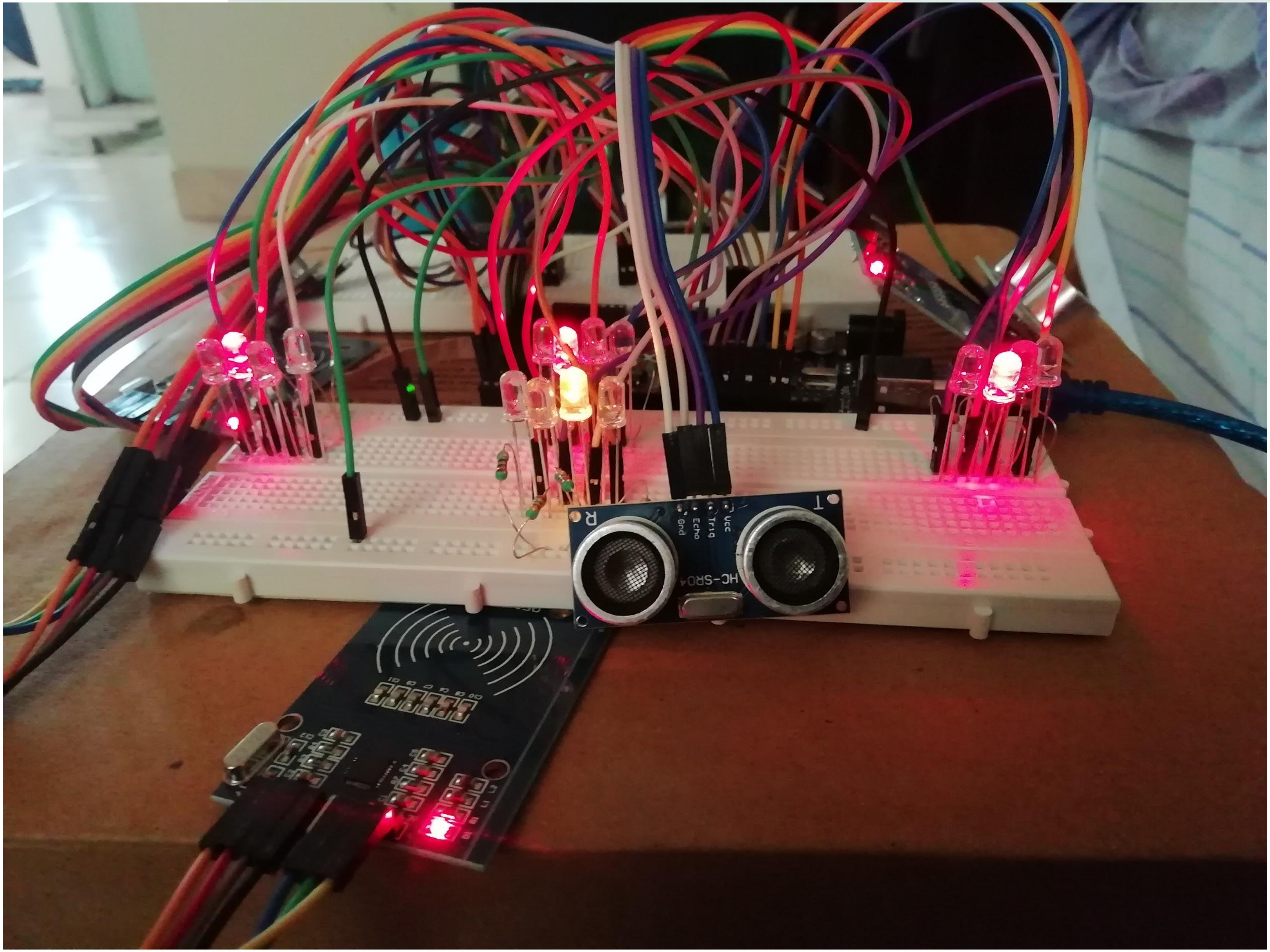
- In case an emergency vehicle is detected by the rfid receiver that signal is immediately given green signal along with a blue flashing light to indicate the presence of emergency vehicle so that other vehicles can make way.

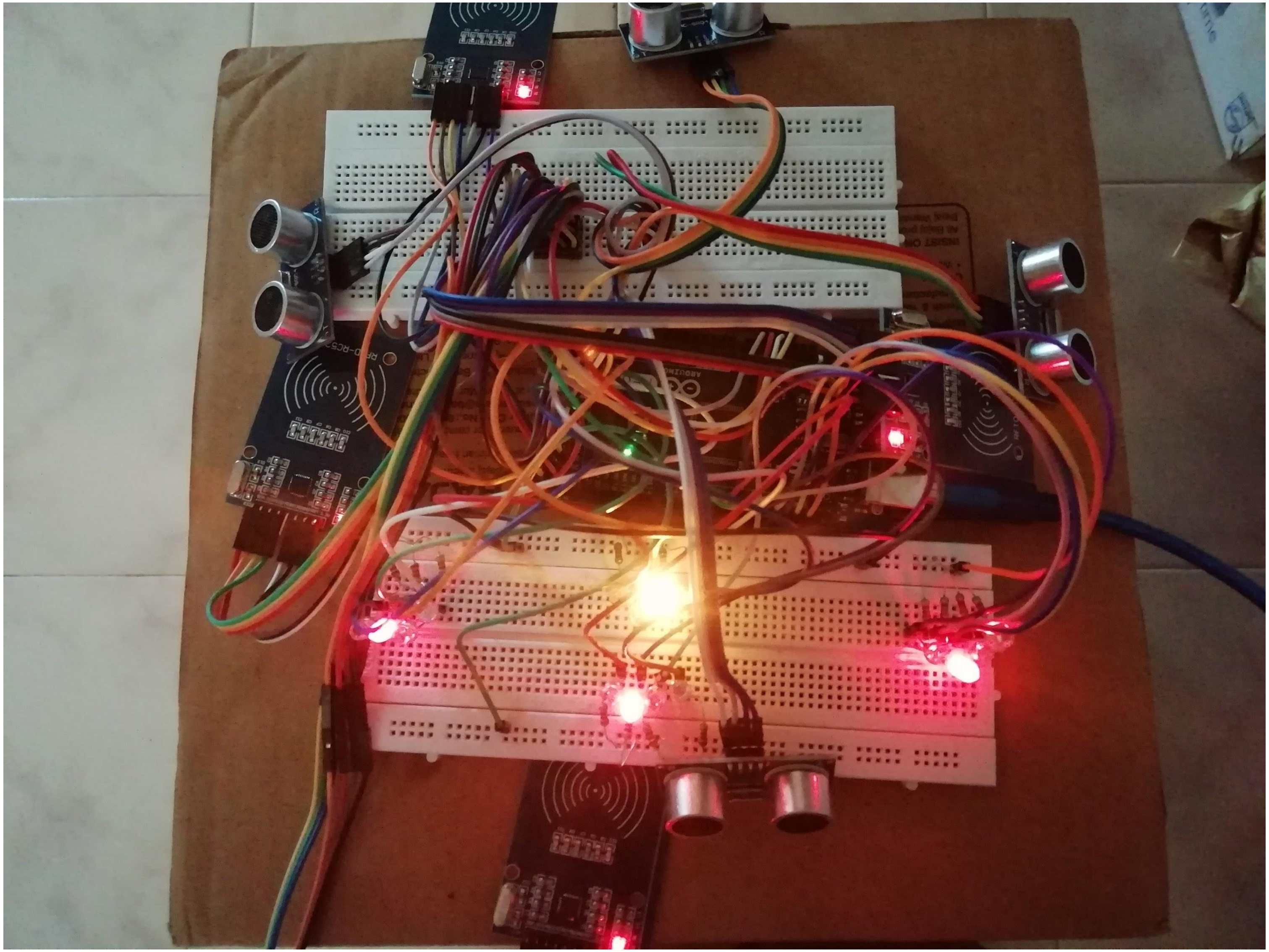
Application

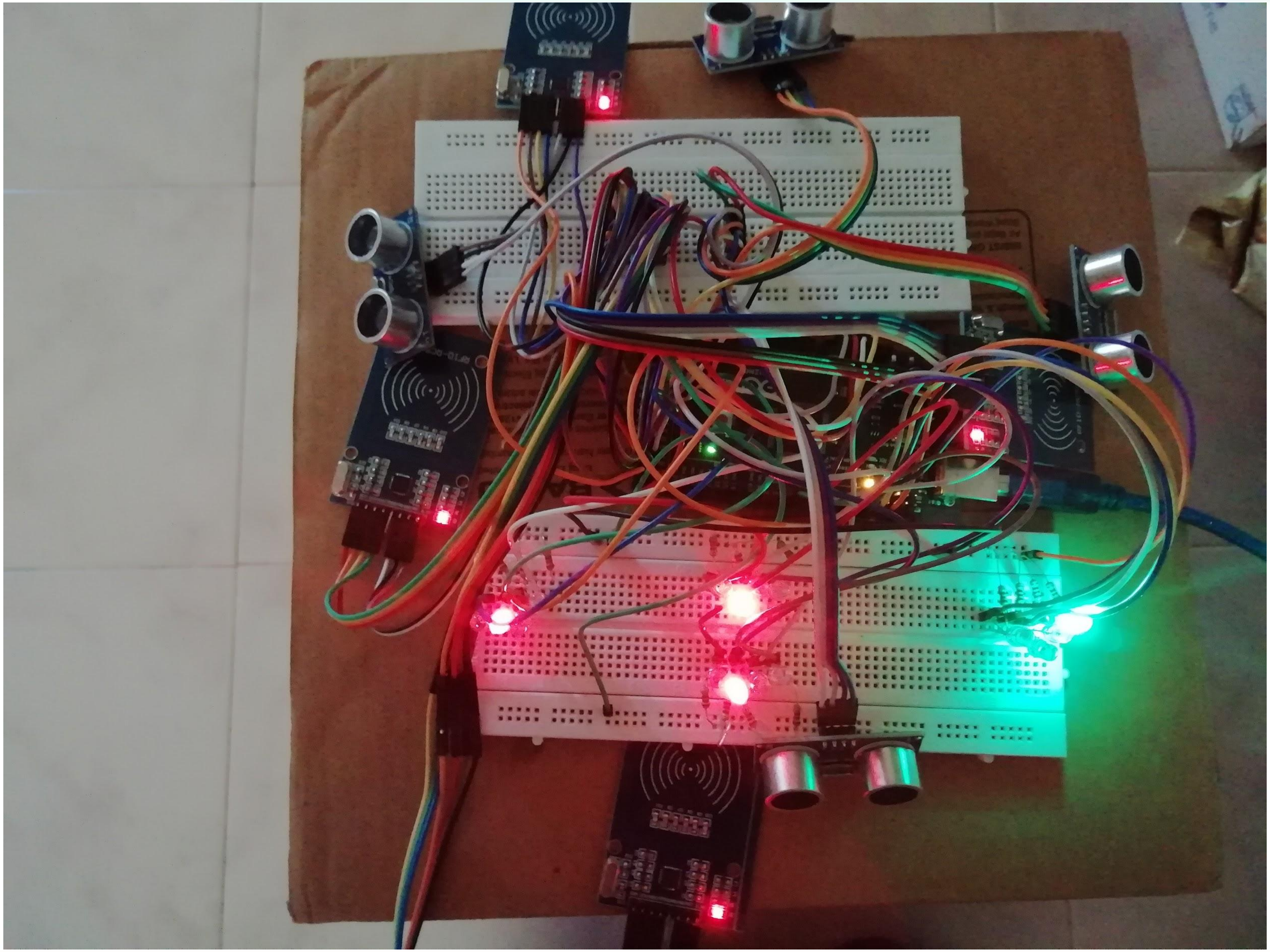
- Can be used in Smart Cities to manage traffic efficiently.
- Green Corridor can be used for fast movement of emergency vehicles and save many lives.

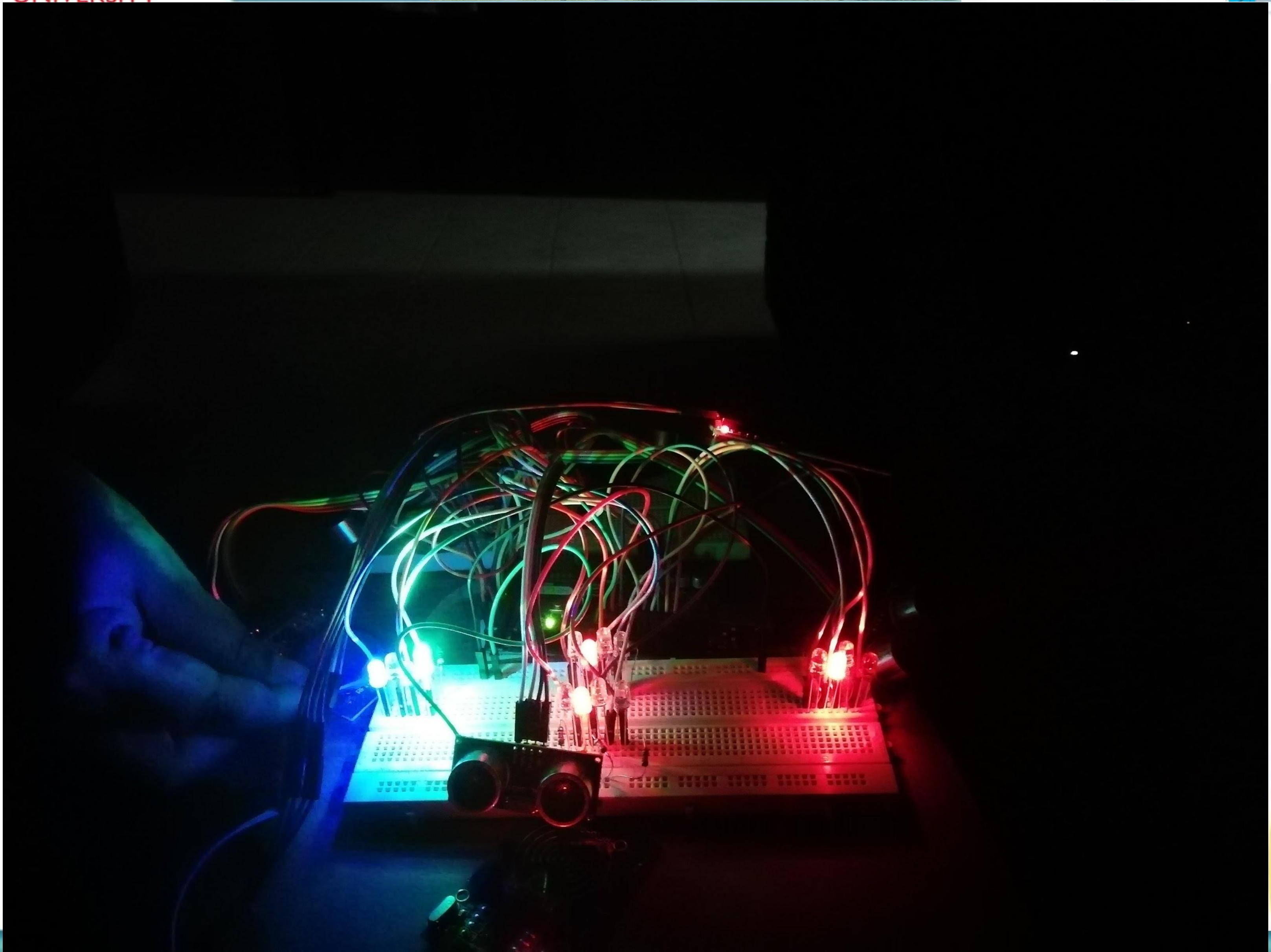
The project has been completed and code along with photos of the project has been attached in the following slides.

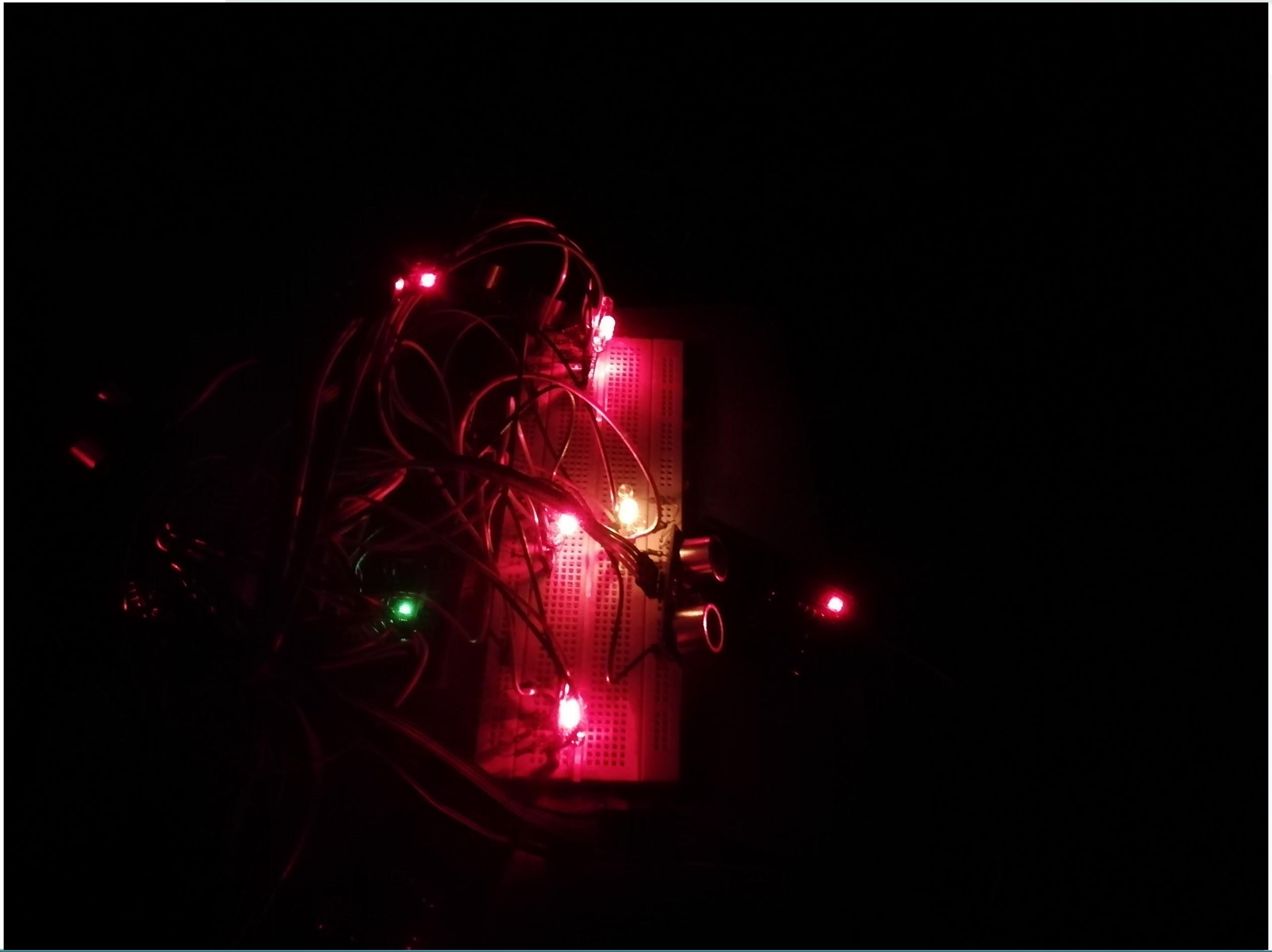



R
T
Vcc
Trig
Echo
Gnd
HC-SR04









```
#include<TimerOne.h>
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         5          // Configurable, see typical pin layout above
#define SS_1_PIN        22         // Configurable, take a unused pin, only HIGH/LOW
required, must be diffrent to SS 2
#define SS_2_PIN        23          // Configurable, take a unused pin, only HIGH/LOW
required, must be diffrent to SS 1
#define SS_3_PIN        24
#define SS_4_PIN        25
#define NR_OF_READERS   4

byte ssPins[] = {SS_1_PIN, SS_2_PIN,SS_3_PIN,SS_4_PIN};
MFRC522 mfrc522[NR_OF_READERS];   // Create MFRC522 instance.
int signal1[] = {46,43,42,47};// Red,yellow,green,blue
int signal2[] = {40,39,38,41};
int signal3[] = {11,12,13,10};
int signal4[] = {7,8,9,6};
```

```
int i;
int redDelay = 5000;
int blueDelay = 4500;
int yellowDelay = 3000;
uint8_t reader=60;
volatile int triggerpin1 = 29;
volatile int echopin1 = 28;
volatile int triggerpin2 = 31;
volatile int echopin2 = 30;
volatile int triggerpin3 = 33;
volatile int echopin3 = 32;
volatile int triggerpin4 = 35;
volatile int echopin4 = 34;
volatile long time;                    // Variable for storing the time traveled
volatile int S1, S2, S3, S4;            // Variables for storing the distance covered

int t = 7;  // distance under which it will look for vehicles.

void setup(){
  Serial.begin(9600);
  SPI.begin();
  Timer1.initialize(700000);  //Begin using the timer. This function must be called first.
"microseconds" is the period of time the timer takes.
```

```
Timer1.attachInterrupt(softInterr); //Run a function each time the timer period finishes.

 // Declaring LED pins as output
 for(int i=0; i<4; i++){
   pinMode(signal1[i], OUTPUT);
   pinMode(signal2[i], OUTPUT);
   pinMode(signal3[i], OUTPUT);
   pinMode(signal4[i], OUTPUT);
 }

 // Declaring ultrasonic sensor pins as output
 pinMode(triggerpin1, OUTPUT);
 pinMode(echopin1, INPUT);
 pinMode(triggerpin2, OUTPUT);
 pinMode(echopin2, INPUT);
 pinMode(triggerpin3, OUTPUT);
 pinMode(echopin3, INPUT);
 pinMode(triggerpin4, OUTPUT);
 pinMode(echopin4, INPUT);
  for (uint8_t reader = 0; reader < NR_OF_READERS; reader++) {
   mfrc522[reader].PCD_Init(ssPins[reader], RST_PIN);} // Init each MFRC522 card
}
```

```
void loop()
{


  // If there are vehicles at signal 1
  if(S1<t)
  {
    signal1Function();
  }

  // If there are vehicles at signal 2
  if(S2<t)
  {
    signal2Function();
  }

  // If there are vehicles at signal 3
  if(S3<t)
  {
    signal3Function();
  }
```

```
// If there are vehicles at signal 4
  if(S4<t)
  {
    signal4Function();
  }

}

// This is interrupt function and it will run each time the timer period finishes. The timer
period is set at 100 milli seconds.
void softInterr()
{
  for (reader = 0; reader < NR_OF_READERS; reader++) {
    // Look for new cards

    if (mfrc522[reader].PICC_IsNewCardPresent() &&
mfrc522[reader].PICC_ReadCardSerial()) {
      Serial.print(F("Rfid reader "));
      Serial.print(reader);
      Serial.print(F(" detected.Calling  "));
      if(reader==0){signal1Function();blue1();}
      if(reader==1){signal2Function();blue2();}
      if(reader==2){signal3Function();blue3();}
      if(reader==3){signal4Function();blue4();}
```

```
// Halt PICC
     mfrc522[reader].PICC_HaltA();
     // Stop encryption on PCD
     mfrc522[reader].PCD_StopCrypto1();

   } //if (mfrc522[reader].PICC_IsNewC
 } //for(uint8_t reader

 // Reading from first ultrasonic sensor
 digitalWrite(triggerpin1, LOW);
 delayMicroseconds(2);
 digitalWrite(triggerpin1, HIGH);
 delayMicroseconds(10);
 digitalWrite(triggerpin1, LOW);
 time = pulseIn(echopin1, HIGH);
 S1= time*0.034/2;

 // Reading from second ultrasonic sensor
 digitalWrite(triggerpin2, LOW);
 delayMicroseconds(2);
 digitalWrite(triggerpin2, HIGH);
 delayMicroseconds(10);
 digitalWrite(triggerpin2, LOW);
 time = pulseIn(echopin2, HIGH);
 S2= time*0.034/2;

 /
```

```
/ Reading from third ultrasonic sensor
  digitalWrite(triggerpin3, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin3, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin3, LOW);
  time = pulseIn(echopin3, HIGH);
  S3= time*0.034/2;

  // Reading from fourth ultrasonic sensor
  digitalWrite(triggerpin4, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerpin4, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerpin4, LOW);
  time = pulseIn(echopin4, HIGH);
  S4= time*0.034/2;


}

void signal1Function()
{
  Serial.println("signal 1");
  low();
```

```
// Make RED LED LOW and make Green HIGH for 5 seconds
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal1[2], HIGH);
  delay(redDelay);
  // if there are vehicels at other signals
  if(S2<t || S3<t || S4<t)
  {
    // Make Green LED LOW and make yellow LED HIGH for 2 seconds
  digitalWrite(signal1[2], LOW);
  digitalWrite(signal1[1], HIGH);
  delay(yellowDelay);
  }
}

void signal2Function()
{
  Serial.println("signal 2");
  low();
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal2[2], HIGH);
  delay(redDelay);
  if(S1<t || S3<t || S4<t)
  {
  digitalWrite(signal2[2], LOW);
  digitalWrite(signal2[1], HIGH);
  delay(yellowDelay);
  }
}
```

```
void signal3Function()
{
  Serial.println("signal 3");
  low();
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal3[2], HIGH);
  delay(redDelay);
  if(S1<t || S2<t || S4<t)
  {
  digitalWrite(signal3[2], LOW);
  digitalWrite(signal3[1], HIGH);
  delay(yellowDelay);
  }
}

void signal4Function()
{
  Serial.println("signal 4");
  low();
  digitalWrite(signal4[0], LOW);
  digitalWrite(signal4[2], HIGH);
  delay(redDelay);
  if(S1<t || S2<t || S3<t)
  {
  digitalWrite(signal4[2], LOW);
  digitalWrite(signal4[1], HIGH);
  delay(yellowDelay);
  }
}
```

```
// Function to make all LED's LOW except RED one's.
void low()
{
  for(int i=1; i<3; i++)
  {
    digitalWrite(signal1[i], LOW);
    digitalWrite(signal2[i], LOW);
    digitalWrite(signal3[i], LOW);
    digitalWrite(signal4[i], LOW);
  }
  for(int i=0; i<1; i++)
  {
    digitalWrite(signal1[i], HIGH);
    digitalWrite(signal2[i], HIGH);
    digitalWrite(signal3[i], HIGH);
    digitalWrite(signal4[i], HIGH);
  }
}
void blue1()
{
    for(i=0;i<20;i++){
      digitalWrite(signal1[3], LOW);
      delay(blueDelay);
      digitalWrite(signal1[3], HIGH);
      delay(blueDelay);
      digitalWrite(signal1[3], LOW);
    }
    Serial.print(F("blue1 "));
}

void blue2()
{
        for(i=0;i<20;i++){
      digitalWrite(signal2[3], LOW);
      delay(blueDelay);
      digitalWrite(signal2[3], HIGH);
      delay(blueDelay);
      digitalWrite(signal2[3], LOW);
    }
    Serial.print(F("blue2 "));
}
void blue3()
{
    for(i=0;i<20;i++){
      digitalWrite(signal3[3], LOW);
      delay(blueDelay);
      digitalWrite(signal3[3], HIGH);
      delay(blueDelay);
      digitalWrite(signal3[3], LOW);
    }
    Serial.print(F("blue3 "));
}
```

```
void blue4()
{
    for(i=0;i<20;i++){
       digitalWrite(signal4[3], LOW);
     delay(blueDelay);
    digitalWrite(signal4[3], HIGH);
    delay(blueDelay);
    digitalWrite(signal4[3], LOW);
    }
    Serial.print(F("blue4 "));

}
```

Thank You