Team Cipher

Automated Railway Crossing

Team Members:

S Gagan Kumar

Spoorthy Madhusudan

Tejaas Mukunda Reddy

Thanush M

Viganesha H R

Vinay N B

Yaser Ahmed N

Acknowledgement

We would like to express our profound gratefulness to Tequedlabs for providing us an opportunity to do this project on the topic Automated Railway Crossing, which also helped us in doing a lot of Research and I came to know many new things, I am really thankful to them.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents and the staff of the Tequedlabs, who have directly or indirectlysupported us during the period of our internship project work.

Regards,

Team Cipher

Table of Contents

SL. No Title Page no.

Chapter-1: About Tequed Labs

TEQUED LABS (OPC) PRIVATE LIMITED is a Private(One Person Company) Company limited by Shares. It is registered with Registrar of Companies, Bangalore on Jan 22, 2018.

Current Status of Tequed Labs (Opc) Private Limited is Active.

It is a Non-govt company with an Authorized Capital of ₹ 1,00,000 (One Lakh Indian Rupees) and Paid Up Capital of ₹ 1,00,000 (One Lakh Indian Rupees).

There are 2 Directors associated with Tequed Labs (Opc) Private Limited. They are: Aditya Shivasharanappa and Supreeth Yerriswamy.

Its Registered Address and Contact Email are 'C/O M N Geetha No 10, BSK 3rd Stage, Bangalore Bangalore KA 560085 IN' and adityask007@gmail.com respectively.

They conduct workshops on various topics such as Internet Of Things(IOT), Machine Learning(ML), Android application development etc.

Chapter-2 : About the Topic

2.1 :Introduction

Automatic Railway Gate Control System is a simple but very useful project, which helpsin automatically opening and closing the railway gate upon detecting arrival or departure of the train.

In general, Railway gates are opened or closed manually by a gate keeper. The information about arrival of train for opening or closing of door is received from nearby station. But some railway crossings are totally unmanned and many railway accidents occur at these unmanned level crossings.

To avoid the human intervention at level crossings completely, we need to automate the process of railway gate control.

The aim of this project is to save lives of people who are crossing unmanned railway crossings; by providing an automatic railway gate solution. It also deals with the reduction of time for which the gate is being kept closed.

By employing the automatic railway gate control at the level crossing the arrival of the train is detected by the sensors placed near to the gate. Hence, the time for which it is closed is less compared to the manually operated gates.

The operation is automatic; error due to manual operation is prevented. Automatic railway gate control is highly microcontroller based arrangements, designed for use in almost all the unmanned level crossing in the train.

A major implementation in our project is the use of LCD displays, not only to provide information about the train arrival and departure, but also providing a provision of

displaying advertisements of growing companys. This will help government in its regular revenue.

2.2 :Applications & Benefits

If installed in unmanned railway crossings, it prevents accidents and saves lives.

- Saving lives of people
- Eco friendly solution
- Low cost
- Can be easily installed

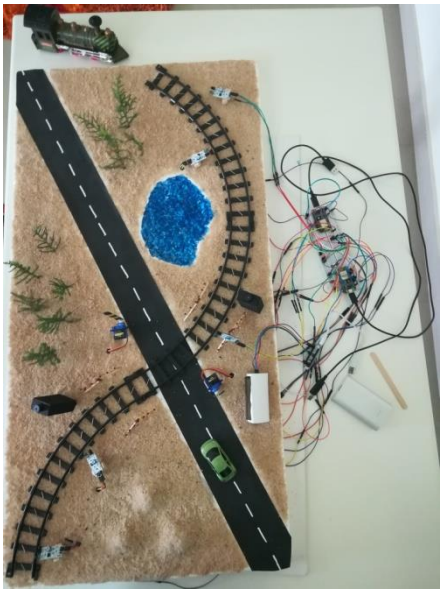
2.3 :Materials Used

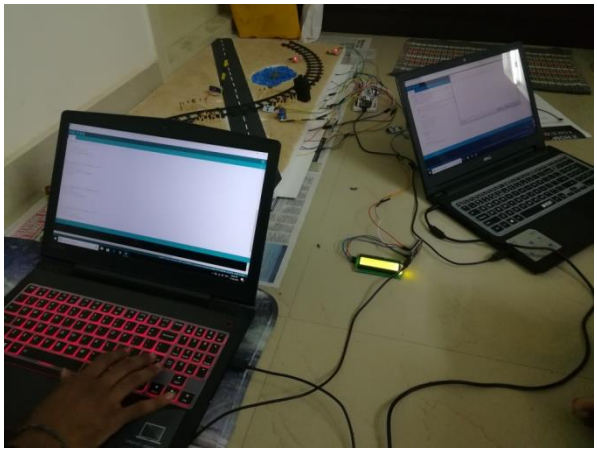
1. Toy train
2. Wood
3. Thermocol

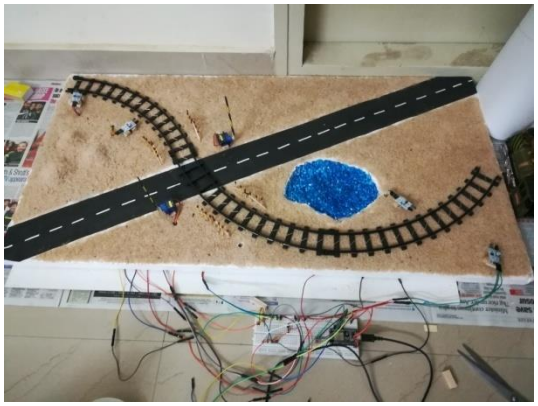
2.4 :Electrical Components

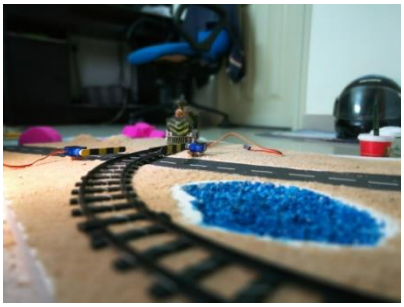
1. Node MCU (ESP8266)
2. IR Sensors
3. Servo Motors
4. LCD Display
5. Buzzer
6. LED lights
7. Bread Board, Wires and Power Supply

Chapter-3 : Tasks performed during the Internship









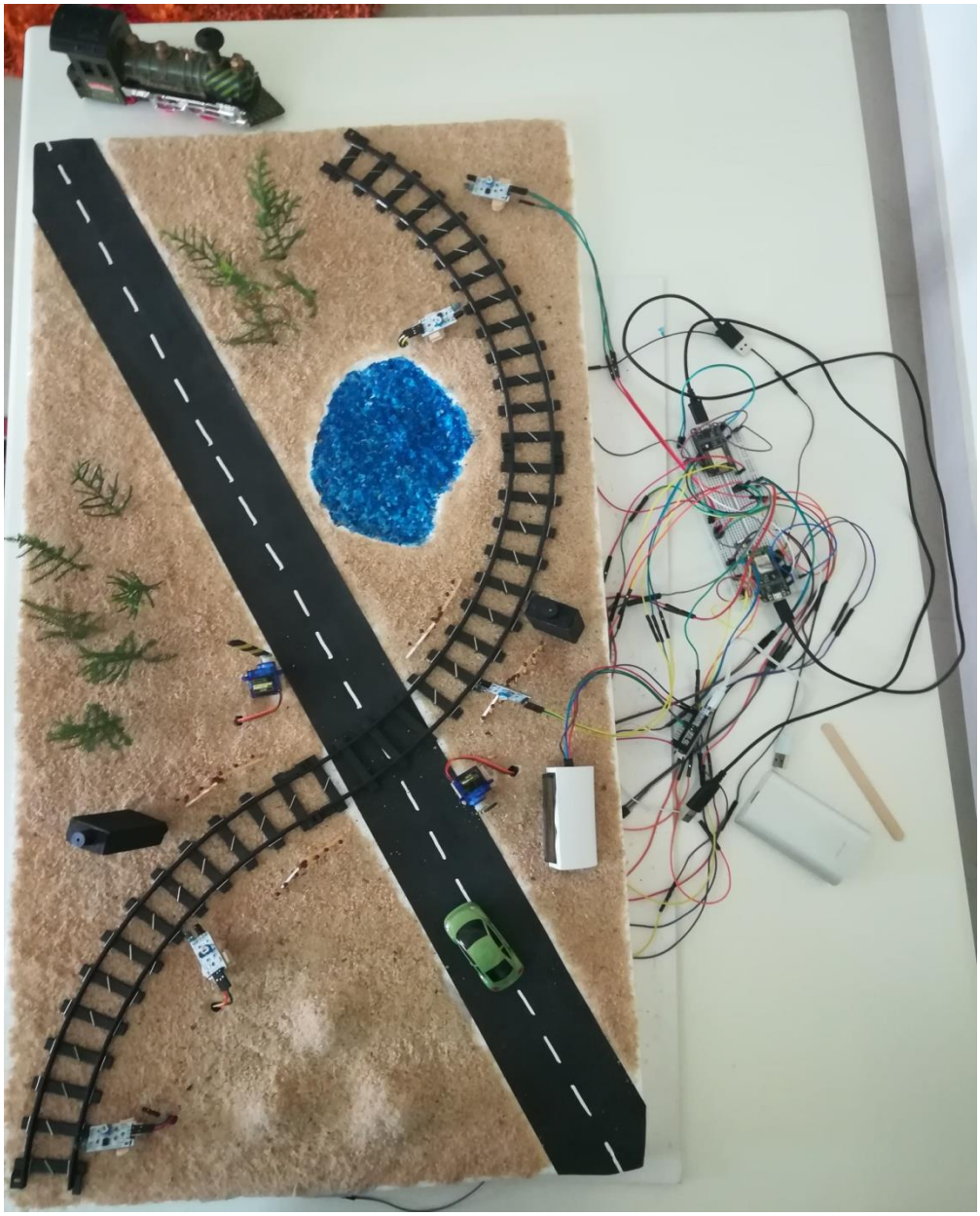
We had four weeks given for the completion of the project.

First week was about planning the roles of people and the resources needed for the project.

Second week each member completed their roles and reported by the end of the week.

Third week was about the integration and implementation of the code.

Fourth week was all about testing the code and debugging the errors and setting up a working prototype.



Chapter-4 : Implementation of the Topic

4.1 :Principle of Operation

The principle of operation behind the working of this project lies in the functioning of IR Sensor. A Reflective type IR Sensor is used in this project.

In Reflective Type IR Sensor, the IR transmitter and receiver are placed side by side. When there is no obstacle in front of the sensor, the IR rays transmitted by the IR Transmitter will travel undetected as there are no rays falling on the IR Receiver.

If there is an obstacle in front of the IR Transmitter and Receiver pair, the IR Rays gets reflected off from the surface of the obstacle and are incident on the IR Receiver.

This setup can be configured to detect an object like a Train and in turn can be used to switch ON or OFF the loads like motors with the help of microcontroller like a NodeMCU.

The NodeMCU powers the servo motors to function in order to open or close the gate whenever IR detects a train coming towards the railway crossing and when it leaves.

A buzzer is incorporated to function along with the servo acting as gates to alert the people of closing of gate.

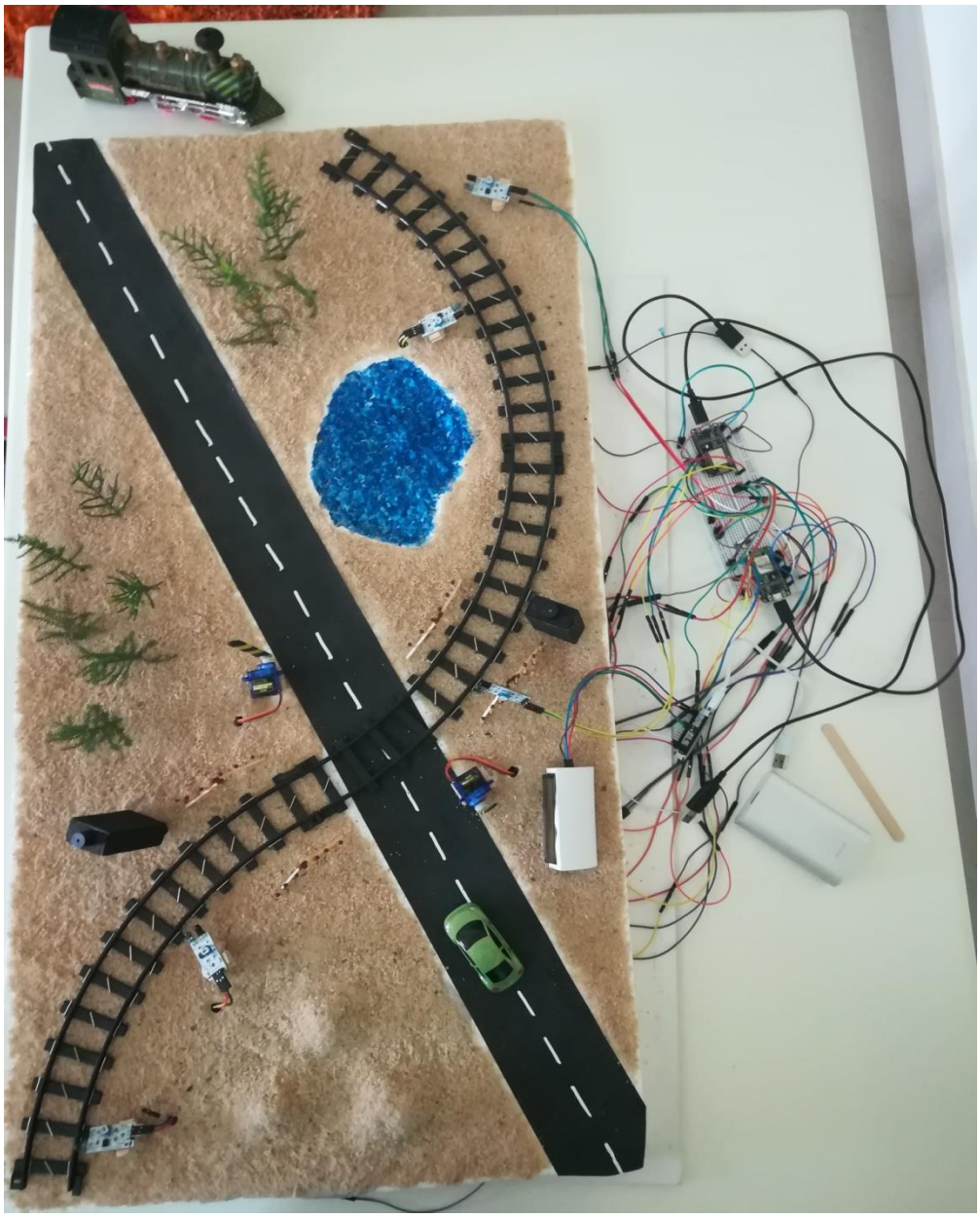
The LCD display meassages regularly about the status of the crossing. LEDs are used to inform the trian driver of any distortion present between the crossing path.

Another IR sensor is placed in the scope of the railway gates to detect any obstacles like vehicles which are jammed in between the gate and not able to move quick enough before the train could arrive.

In such a situation, an LED placed at approx. 500 mts from the gate would glow red indicating the train driver of some problem ahead and needs to slow down or stop the train. If this IR is clear then a green LED glows indicating a Safe to move signal.

Real time notifications about the Railway Crossing is sent to Railway Administrator Office in order to keep track of the functioning of railway crossing.

4.2 :The Prototype



4.3 : The Code

4.3.1 : Code for train detection and gate operation.

The following code use's 4 IR's which are used to detect the real time position of train before reaching the crossing and after leaving the crossing. Along with that, it has code for the operations of the servo motors for the gates.

```
//this is the program for the first IR(the 4 gates exept the car one)
#include<Servo.h>
#include <ESP8266WiFi.h>
        //D0 free
int ir1 = D1;
int ir2 = D2;
        //D6 free for LCD
int ir4 = D3;
        //D7 is servo motor
int ir5 = D4;
int t1 = D6;
int t2 = D7;
        //D7 free for LCD
int op  = D8; //this sends the digital value 1 to the second mcu
        //buzzer A0 free (this can be used as a regular digital pin also)for buzzer
int pos=0;
String apiKey = "GVKY13RMI3LSHAK6";     //  Enter your Write API key from ThingSpeak
const char *ssid =  "home";     // replace with your wifi ssid and wpa2 key
const char *pass =  "12345678";
const char* server = "api.thingspeak.com";
int timeCounter=0;
int valIp;
Servo servo;
WiFiClient client;
void setup()
{
  pinMode(ir1, INPUT);
  pinMode(ir2, INPUT);
  pinMode(ir4, INPUT);
  pinMode(ir5, INPUT);
  pinMode(t1, OUTPUT);
  pinMode(t2, OUTPUT);
  pinMode(op, OUTPUT);
  servo.attach(D5);
  Serial.begin(9600);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
     {
            delay(500);
            Serial.print(".");
     }
   Serial.println("");
   Serial.println("WiFi connected");
```

```
}

void loop()
{

  Serial.println(valIp);
  delay(1000);
  //condition1 00

  digitalWrite(t1,LOW);
digitalWrite(t2,LOW);
Serial.println(LOW);
Serial.println(HIGH);

Serial.println("waiting for a train to come ");
int ReadIr1 = digitalRead(ir1);
int ReadIr2, ReadIr4, ReadIr5;
ReadIr5 = digitalRead(ir5);


if (ReadIr1 == 1 )
{

      servo.write(0);
      ReadIr2 = digitalRead(ir2);
      Serial.println("IR1 is activated :train is coming");
      digitalWrite(t1,LOW);
      digitalWrite(t2,HIGH);
      Serial.println(t1);
      Serial.println(t2);

      //lcd_write("train is at 1km")
      while (ReadIr2 == 0)                          //wait for train to come from first ir to the 2nd ir
      {
          digitalWrite(op,HIGH); //car mcu starts working whenever the train is approaching. The red light
          ReadIr2 = digitalRead(ir2);
          Serial.println("waiting for train to reach ir2: train is at around 1 km from the crossing");
          delay(500);
          if (ReadIr2 == 1)
          {
              goto A;
          }
          ReadIr2 = digitalRead(ir2);
          timeCounter+=1;
      }



  A: if ( ReadIr2 == 1 )
      {
        for(pos=0;pos<=90;pos+=1)
        {
            servo.write(pos);
            //make buzzer high here with appropriate delays
            delay(25);
        }
        Serial.println("train has reached ir2: the train has arrived at 500 mts");
        digitalWrite(op,LOW); //car mcu stops working whenever the train is approaching. The red light
        ReadIr4 = digitalRead(ir4);

        while (ReadIr4 == 0)
          {
              //lcd_write("waiting for train to reach ir4: train is less than 500 mts from the gate")
              Serial.println("train is less than 500 mts from the gate");
```

```
    digitalWrite(t1,HIGH);
    digitalWrite(t2,LOW);
    Serial.println(t1);
    Serial.println(t2);

            delay(500);
            if (ReadIr4 == 1)
            {
               goto B;
            }
            ReadIr4 = digitalRead(ir4);
            timeCounter+=1;
        }
    }



B: if ( ReadIr4 == 1 )
    {
      //lcd_write(" calm....train has almost left")
      Serial.println("train reached ir 4: the train is leaving");
      ReadIr5 = digitalRead(ir5);
      while (ReadIr5 == 0)
      {
          //lcd_write("more calm...train has almost left")
          Serial.println("waiting for train to reach ir5: train has almost left");
   digitalWrite(t1,HIGH);
   digitalWrite(t2,HIGH);
   Serial.println(t1);
   Serial.println(t2);

         ReadIr5 = digitalRead(ir5);
         if (ReadIr5 == 1)
         {
             goto C;
         }
         timeCounter+=1;
         delay(500);
     }
C:   while (ReadIr5 == 1)
     {
         Serial.println("waiting for train to complete pass the IR5 : train is leaving(crossing the 5th gate)");
         ReadIr5 = digitalRead(ir5);
                 //condition1  0 0

   digitalWrite(t1,LOW);
   digitalWrite(t2,LOW);
   Serial.println(t1);
   Serial.println(t2);

         if (ReadIr5 == 0)
         {
             goto D;
         }
         timeCounter+=1;
```

```
            delay(500);
        }
  D: if (ReadIr5 == 0)
        {
            Serial.println("track now clear: Train has left the craoosing. Crossing is now open again");
            for(pos=90;pos>=0;pos-=1)
          {
              servo.write(pos);
              //make buzzer high here with appropriate delays
              delay(25);
          }

        }
      }

        Serial.print("time taken is : ");
        Serial.print(timeCounter);
        if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
        {

           String postStr = apiKey;
           postStr +="&field1=";
           postStr += String(timeCounter);
           postStr += "\r\n\r\n";
           client.print("POST /update HTTP/1.1\n");
           client.print("Host: api.thingspeak.com\n");
           client.print("Connection: close\n");
           client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
           client.print("Content-Type: application/x-www-form-urlencoded\n");
           client.print("Content-Length: ");
           client.print(postStr.length());
           client.print("\n\n");
           client.print(postStr);
       }
       timeCounter=0;


}



 else if (ReadIr5 == 1 )
 {

   servo.write(0);
   Serial.println("IR5 is activated :train is coming");
   //lcd_write("train is at 1km")
   ReadIr4 = digitalRead(ir4);
   while (ReadIr4 == 0)                          //wait for train to come from first ir to the 2nd
   {
     digitalWrite(op,HIGH); //car mcu starts working whenever the train is approaching. The red light
     ReadIr4 = digitalRead(ir4);
     Serial.println("waiting for train to reach ir2: train is at around 1 km from the crossing");
```

```
        //condition1  0 1

          digitalWrite(t1,LOW);
          digitalWrite(t2,HIGH);
          Serial.println(0);
          Serial.println(1);

        delay(500);
        if (ReadIr4 == 1)
        {
          goto x;
        }
        timeCounter+=1;
        ReadIr4 = digitalRead(ir4);
      }



x: if ( ReadIr4 == 1 )
      {
        for(pos=0;pos<=90;pos+=1)
        {
          servo.write(pos);
          delay(25);
        }
        Serial.println("train has reached ir4: the train has arrived at 500 mts");
        digitalWrite(op,HIGH); //car mcu starts working whenever the train is approaching.
        ReadIr2 = digitalRead(ir2);
        while (ReadIr2 == 0)
          {
            //lcd_write("waiting for train to reach ir2: train is less than 500 mts from the gate")
            Serial.println("train is less than 500 mts from the gate");

                    //condition1  1 0

          digitalWrite(t1,HIGH);
          digitalWrite(t2,LOW);
          Serial.println(t1);
          Serial.println(t2);


            delay(500);
            if (ReadIr2 == 1)
            {
              goto y;
            }
            ReadIr2 = digitalRead(ir2);
            timeCounter+=1;
          }
      }



y : if ( ReadIr2 == 1 )
```

```
    {
        //lcd_write(" calm....train has almost left")
        Serial.println("train reached ir 2: the train is leaving");
        ReadIr1 = digitalRead(ir1);
        while (ReadIr1 == 0)
        {
            //lcd_write("more calm...train has almost left")
            Serial.println("waiting for train to reach ir1: train has almost left");

        //condition1  1 1

        digitalWrite(t1,HIGH);
        digitalWrite(t2,HIGH);
Serial.println(t1);
        Serial.println(t2);


            ReadIr1 = digitalRead(ir1);
            if (ReadIr1 == 1)
            {
               goto z;
            }
            delay(500);
            timeCounter+=1;
        }
  z:    while (ReadIr1 == 1)
        {

       Serial.println("waiting for train to complete pass the IR1 : train is leaving(crossing the 5th gate)");
     //condition1  0 0

     digitalWrite(t1,LOW);
     digitalWrite(t2,LOW);
     Serial.println(t1);
     Serial.println(t2);

        ReadIr1 = digitalRead(ir1);
        if (ReadIr1 == 0)
        {
           goto v;
        }
        delay(500);
        timeCounter+=1;
     }
v: if (ReadIr1 == 0)
     {
        Serial.println("track now clear: Train has left the craoosing. Crossing is now open again");
        for(pos=90;pos>=0;pos-=1)
       {
        servo.write(pos);
        delay(25);
       }

     }
   }
```

```
 Serial.print("time taken is : ");
 Serial.print(timeCounter);

 if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
 {

    String postStr = apiKey;
    postStr +="&field1=";
    postStr += String(timeCounter);
    postStr += "\r\n\r\n";
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);
}
timeCounter=0;

 }


 delay(500);
}
```

4.3.2 : Code for Car detection and alerting train driver.

The following code is used for the fifth IR, used for car detection when it gets stuck in between the gates(when gates are closed). When the IR detects the cars, it immediately alerts the train driver by changing the signal light to red from green, thus giving him ample amount of time to stop the train. At the same time the buzzer at the crossing, starts to buzz thus alerting the people at the crossing. The buzzer is connected in parallel with the red led.

```
int ip = D1;
int red_led = D2;
int green_led = D3;
int IR3= D4;
int val =0;

void setup()
{
  pinMode(ip,INPUT);
  pinMode(IR3,INPUT);
  Serial.begin(9600);
  pinMode(red_led,OUTPUT);
  pinMode(green_led,OUTPUT);
}

void loop()
{

  val=digitalRead(ip);
  val=1;
  if(val == 1)
  {
      Serial.println("val high");
      while(digitalRead(IR3)==1) //on red led while there is a car stuck
      {   Serial.println("ir3 high :car detected");
          digitalWrite(green_led,LOW);
          digitalWrite(red_led,HIGH);

      }


  }

  digitalWrite(green_led,HIGH);
  digitalWrite(red_led,LOW);


  delay(500);
  // put your main code here, to run repeatedly:

}
```

4.3.3 : Code for LCD Display.

Depending on the train's position from the railway crossing different IR's will get activated and respective messages will be displayed on the LCD.

```
#include <ESP8266WiFi.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int ir1 = D1;
int ir2=D2;

void setup() {
  // put your setup code here, to run once:

   Wire.begin(D6,D7);

   lcd.init();

   lcd.begin(16,2);
   lcd.backlight();

    pinMode(ir1, INPUT);
       pinMode(ir2, INPUT);
    Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  lcd.clear();
int ReadIr1 = digitalRead(ir1);
int ReadIr2 = digitalRead(ir2);

if(ReadIr1==LOW && ReadIr2==LOW)
{ lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("THANK YOU");
  lcd.setCursor(0,1);
   lcd.print("TEQUED LABS");
    delay(5000);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DRIVE SAFE");
  lcd.setCursor(0,1);
   lcd.print("HAVE A GOOD DAY");
    delay(5000);
```

```
}
else if(ReadIr1==LOW && ReadIr2==HIGH)
{lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TRAIN IS COMING");
  lcd.setCursor(0,1);
  lcd.print("PLEASE STOP");

  delay(500);


}
  else if(ReadIr1==HIGH && ReadIr2==LOW)
{lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("GATE CLOSING");
  lcd.setCursor(0,1);
  lcd.print("PLEASE WAIT");
    delay(500);


}

  else if(ReadIr1==HIGH && ReadIr2==HIGH)
{ lcd.clear();
 lcd.setCursor(0,0);
   lcd.print("TRAIN LEAVING");
   lcd.setCursor(0,1);
  lcd.print("PLEASE WAIT");
      delay(500);
}
}
```

5. Executive Summary

The major aim of the project is to provide a safe automated railway crossing by using IR sensors.

LCD screens have been used to display real time messages about the arrival and departure of the train to the people present near the crossing platform.

A means of revenue for the government has been provisioned by the use of a LCD screen at railway crossing which can be used as a source of endorsement for many companies.

Real-time notifications is also sent to Railway Admistration Centre for checking status of the railway crossing.

6. References

www.roboindia.com

www.github.com

www.w3schools.com

www.wikipedia.com

www.nodemcu.com