

Automated Level Crossing with Alert System and GSM

K. Naresh Kumar^a, Jilani Basha M^b, Naveen Kumar S^c, Nithish Kumar S^d, Nizzar Ahmed Z^e,
Jabamani T^f

^{a,b,c,d,e,f}Department of EEE, R.M.K. Engineering College, Thiruvallur, Tamilnadu.
Senior Engineer, Vestas Wind Technology India Private Ltd

Abstract

The aim of our project is to automate the gate at a railway level crossing and to alert the vehicles near the gate and the officials in unavoidable situation. This idea reduces the accidents caused at level crossing due to human error and also it is time efficient. In the existing system, railway gates are operated by a gate keeper either completely manual or semi manual with switch control. Information about train position (arrival to the station) is gathered by the gate keeper. Station in-charge of previous station, informs the gate keeper of next station to be ready. But, automation eases this process flow. We have used Arduino as microcontroller. Microcontroller allows dynamic and faster control. Our idea is also to sense those vehicles, which are in between the gates, so as to provide safety even in unmanned situation. For the case of prototype, we will use motion detection sensors, as of now. But, in real-time situation, we will use the existing relay network as sensing element. Thus, making the system safe, reliable and efficient

Keywords: automate; Arduino; safety; sensors

1. Introduction

Indian Railway is the largest mode of transportation used and it is also the largest job providing organisation. There are 57.5% of manned gates and 42.5% of unmanned gates. Normally, these gates are operated manually, which leads to some percentage of errors and causing a lot of revenue loss for Indian Railways and thus to the Indian Government indirectly, due to delay of trains. To avoid this, we are introducing our system.

Our system eliminates complete manpower, which leads to zero possibility of errors and less management, which can save lots of time and money for the Indian Railway and also ensures safety for everyone.

As an additional safety feature, we have included GSM, as mentioned in the title, to alert nearby officials, in case of abnormal conditions, which occur rarely. We have included some tolerance time for both Alert system and GSM separately.

2. Principle Of Working

The main components used in this project are Arduino UNO, Micro Servo Motor, IR sensor, Buzzer, 9V Battery and GSM Module. The train which is arriving is detected by the IR sensor (for prototype only) placed on one side of the gate. The sensor placed on the other side senses the train leaving the gate. Generally, the sensor detecting the arrival of train is placed at a longer distance from the gate than that of the sensor sensing the departure for safety purpose.

Once train is sensed, signal is received by the Arduino and it checks for the presence of vehicle in the range of gate motion with help of IR sensor. If, none is sensed, the motor rotates for 90 degree and the gate closes and green signal is given for train. If, there is vehicle present between gates, alert signal is given to those vehicles

Automated Level Crossing with Alert System and GSM

after some threshold. Even after the alert, if the state continues, a SMS is sent to the nearby official with help of GSM module.

When, departure sensor senses the train, the gate opens.

We use IR Sensors for train detection only for the purpose of prototype. But, in real-time, we use the existing relay network to sense the vehicle rather than using sensors.

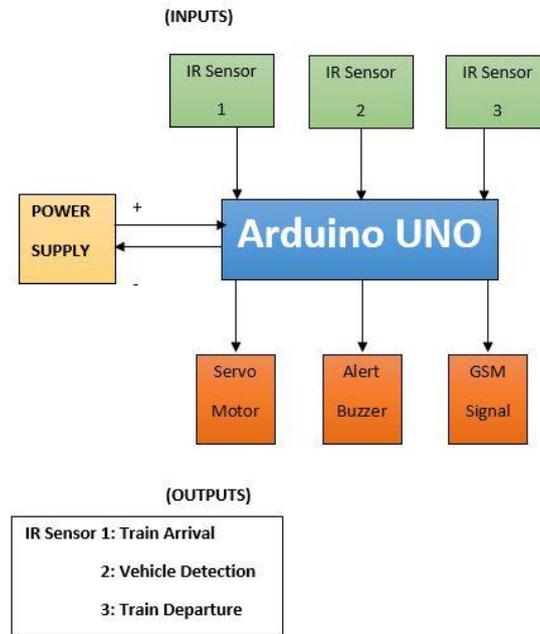


Fig 1: Block Diagram

3. Software

Tinkercad:

Tinkercad is an online based design platform developed by Autodesk, for designing 3D models, circuits and codes for simulation at a free of cost.

We used Tinkercad for simulation of our project. We will share images of simulation in later part of the report.

Arduino IDE:

We used this application to write code and upload that to an Arduino board. It also displays the error in the code, if any present and has serial monitor option, which helps in numerical visualisation of data.

4. Images Of The Used Components



Fig 2: Arduino UNO



Fig 3: Servomotor



Fig 4: IR Sensor front and back



Fig 5: Buzzer



Fig 6: GSM Module

5. Programming

Code 1:

```
#include <Servo.h>
const int pingPin = 7;
const int ledPin1 = 13;
const int ledPin2 = 12;
const int ledPin3 = 11;
const int pingPin1 = 6 ;
const int pingPin2 = 5 ;
int pos=0;
Servo myservo;
int val=0;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
```

```
pinMode(pingPin, INPUT);
pinMode(pingPin1, INPUT);
pinMode(pingPin2, INPUT);
myservo.attach(10);
}
void loop() {
  int cm1,cm2,cm3;
  cm1=digitalRead(pingPin);
  cm2=digitalRead(pingPin1);
  cm3=digitalRead(pingPin2);
  Serial.print(cm1);
  Serial.print(" ");
  Serial.print(cm2);
  Serial.print(" ");
  Serial.print(cm3);
  //Train sensed, Vehicle not sensed
  if(cm1 == 0 && cm2 == 1) {
    pos=1;
    val=0;
  }
  //Both sensed
  if(cm1 == 0 && cm2 == 0){
    val++;
  }
  if(val>0 && cm2 ==0){
    val++;
  }
  //departure sensed
  if(cm3 == 0){
    pos=0;
  }

  if(pos == 1){
    val=0;
  }

  if(cm2 == 1)
  {
    val=0;
  }
}
```

Automated Level Crossing with Alert System and GSM

```
}

if(val>=500 && val<1000)
{

    tone( ledPin2, 5000);
    //digitalWrite(ledPin2, HIGH);
}

else if(val>=1000)
{
    digitalWrite(ledPin3, HIGH);
}

else
{
    digitalWrite(ledPin3, LOW);
    noTone( ledPin2);
}

if(pos==1)
{
    myservo.write(90);
    val=0;
}
else
{
    myservo.write(0);
}

Serial.println();
Serial.print(val);
Serial.println();
}
```

Code Explanation:

As we work with servo motor, we included “Servo.h” header file. The signal pins of arrival sensor, vehicle detection sensor and departure sensor are given to pin 7, 6 and 5 respectively. These values are digitally read (‘0’ or ‘1’) into 3 int variables. Based on the values of these variables, we have written series of if and else conditions to obtain required output. In case of the sensor we use, the variables hold ‘0’ if object is sensed and holds ‘1’ if not sensed.

Code 2:

```
#include <SoftwareSerial.h>
SoftwareSerial serial(9, 10);
int state = 0;
const int pin = 2;
void setup()
{
  mySerial.begin(9600);
  Serial.begin(9600);
}
void loop(){
  int a=digitalRead(pin);
  if ( a==0 && state == 0) {
    if (Serial.available(>0)){
      if(Serial.read()=='s'){
        SendMessage();
      }
      if (serial.available(>0)
Serial.write(serial.read());
    }
  }
  if (a == 1 && state == 1) {
    state = 0;
  }
}
void SendMessage()
{
  serial.println("AT+CMGF=1");
  delay(1000);
  serial.println("AT+CMGS=\"+91xxxxxxxxxx\"");
  delay(1000);
  serial.println("I am SMS from GSM Module");
  delay(100);
  serial.println((char)26);
  delay(1000);
}
```

Code Explanation:

“Software Serial.h” header file is used, as we work on sharing data between two hardware components. Here, we used pin 9 and 10 of Arduino as RX and TX pins respectively instead of pin 0 and 1 to avoid some

Automated Level Crossing with Alert System and GSM

issues which may occur. An object of above mentioned header file is created and this is used to write and share content to the GSM module. SMS content to be sent is written within println function of the created object (for example: serial). Here X is replaced by mobile number and 91 is the Indian code.

6. Simulation

We have done our simulation through Tinkercad online platform. We designed a rough circuit based on our idea and then tinkered it step by step to get the desired output. Since, there is no IR sensor in Tinkercad, we had to use ultrasonic sensor and code it and make it work as a detection sensor. We have used 3 sensors. One each for detecting train arrival, train departure and vehicle detection. As, GSM Module can't be simulated, we have used an LED to indicate that signal to GSM is turned on at required test case.

Tinkercad platform also has coding section and serial monitor to view printed output, which was very useful for us to visualize numerical data output, so as to make changes accordingly to get our expected performance from the system.

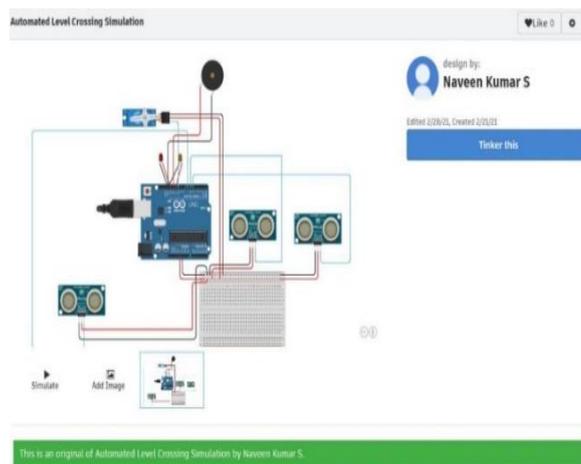


Fig 7: Tinkercad Simulation 1

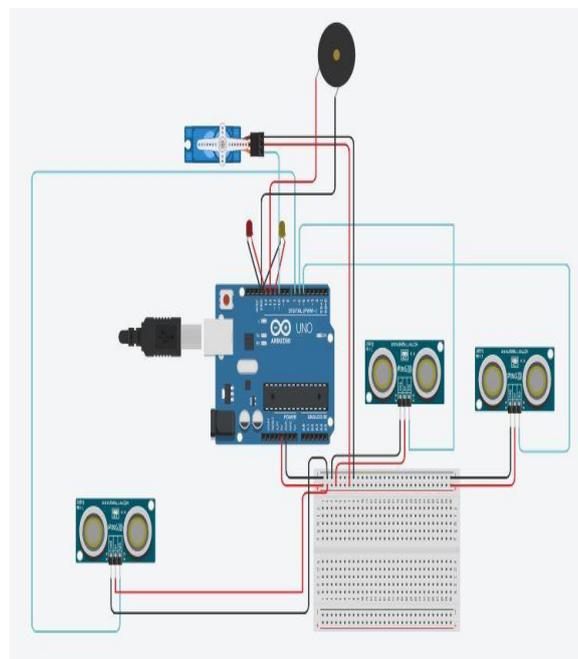


Fig 8: Tinkercad Simulation 2

7. Initial Prototype

On 14th March 2021, we built the initial hardware prototype and tested it. We got our expected outputs. We didn't use GSM Module in this prototype. But, we tested GSM Module for our test case on 16th March and it worked fine. So, now we are working on developing the model with decoration as the final prototype.

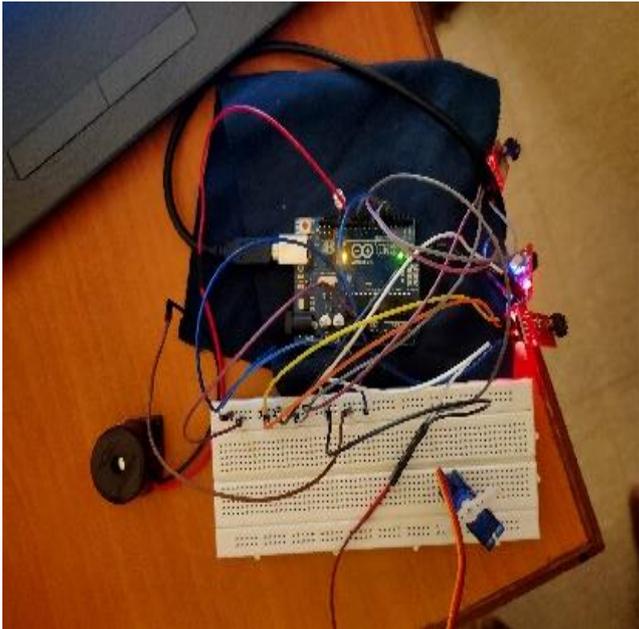


Fig 9i: Initial Prototype

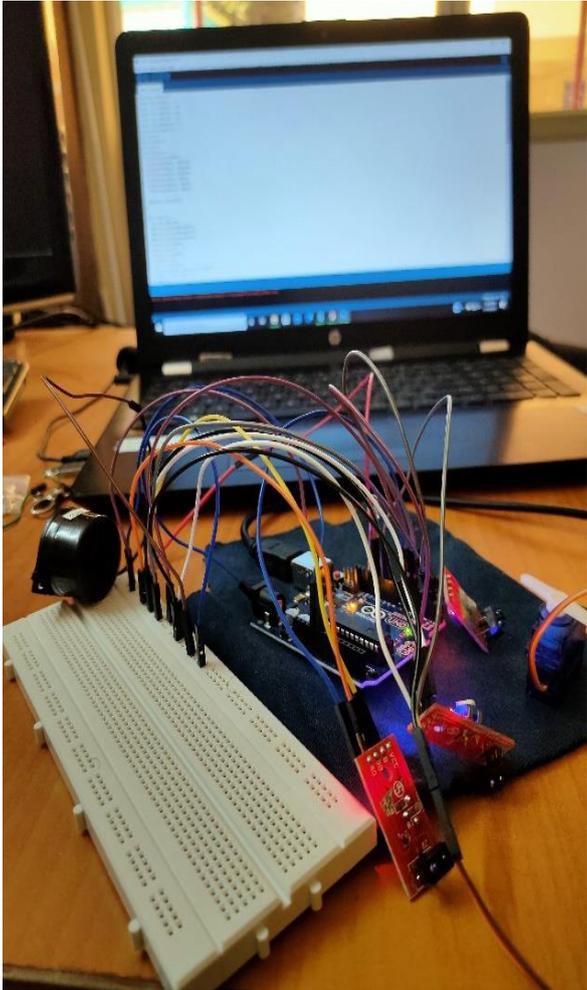


Fig 9ii: Initial Prototype

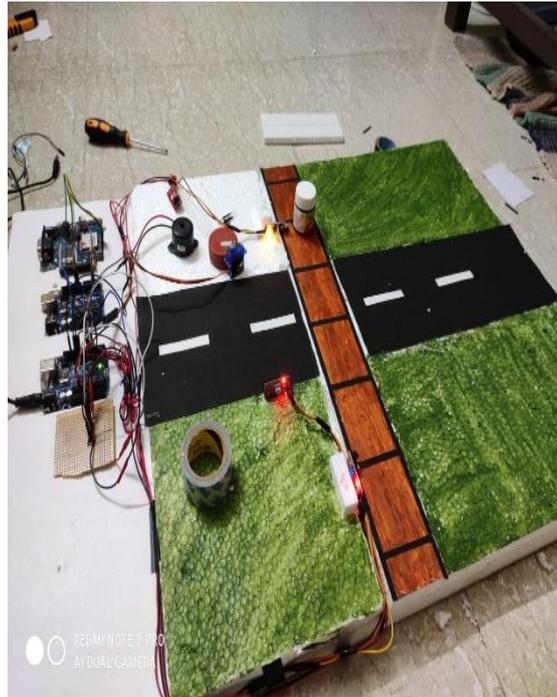


Fig 10i: Main Prototype Building

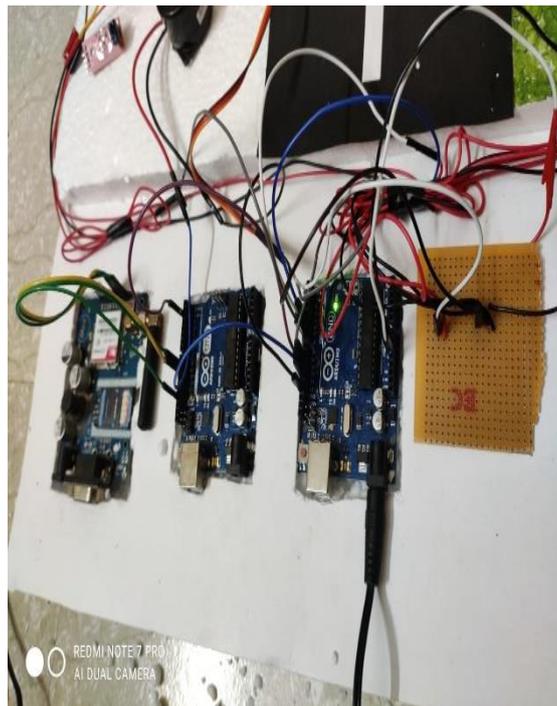


Fig 10ii: Main Prototype Building

8. Results And Discussion

Automating the gate will improve safety measures and will be time and manpower saving in a long run. First, research has been done to check the feasibility of the project. Discussions have been done with the officials of South Central Railway, to verify our idea. Components needed for the project was finalized and prototype has been built with our code for desired output.

9. Conclusion

Automatic level crossing with alert system and GSM is primarily concerned on reducing the human error in railway gate control, which is the main cause for accidents, which may also result in death. Hence, automating

the gate will improve safety measures and will be time and manpower saving in a long run. Human beings are liable to make errors. So automation will reduce the chances of accidents. As a future scope, we will also implement the same system for double and multiple tracks by interlinking those systems through communication between them

References

- [1] M Babusab, Senior Tech, South Central Railway, for idea verification.
- [2] R Selvakumar, A Grade Electrical Contractor, for electrical guidance.
- [3] https://en.wikipedia.org/wiki/Passive_infrared_sensor
- [4] www.fierceelectronics.com/sensors/what-ir-sensor
- [5] https://en.wikipedia.org/wiki/Level_crossing
- [6] http://www.brainkart.com/article/Classification-of-Railway-Level-Crossings_4321/
- [7] https://en.wikipedia.org/wiki/Motion_detector
- [8] <https://en.wikipedia.org/wiki/Servomotor>
- [9] https://en.wikipedia.org/wiki/Arduino_Uno
- [10] <https://www.electronicsforu.com/resources/gsm-module>
- [11] <https://www.circuitstoday.com/interface-gsm-module-with-arduino>
- [12] <https://www.arduino.cc/reference/en/>
- [13] <https://en.wikipedia.org/wiki/Buzzer>
- [14] [http://www.sphinxnsai.com/2018/ch_vol11_no4/1/\(6370\)V11N04CT.pdf](http://www.sphinxnsai.com/2018/ch_vol11_no4/1/(6370)V11N04CT.pdf)
- [15] <https://www.ijitee.org/wp-content/uploads/papers/v8i4s2/D1S0048028419.pdf>
- [16] <https://predictabledesigns.com/how-to-build-a-gsm-cellular-panic-alarm-using-arduino/>
- [17] <https://create.arduino.cc/projecthub/ales-svoboda/arduino-sms-automation-part-2-3-592add>
- [18] <https://www.arduino.cc/reference/en/libraries/servo/>
- [19] <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>
- [20] <https://www.youtube.com/watch?v=9UEcT5GxdBk&t=601s>
- [21] <https://create.arduino.cc/projecthub/muhammad-aqib/density-based-traffic-light-controller-using-arduino-8636ad>
- [22] <https://www.tinkercad.com/things/dRQ70AnOAYE-automatic-railway-gate-control>
- [23] <https://www.allaboutcircuits.com/projects/servo-motor-control-with-an-arduino/>
- [24] <https://create.arduino.cc/projecthub/Raushancpr/arduino-with-ir-sensor-1579b6>
- [25] https://indianrailways.gov.in/railwayboard/uploads/codesmanual/IRPWM/PermanentWayManualCh9_data.htm