**CSE360 - Computer Interfacing**

**Project Report**

# Title of the Project: Advance Automated Rail Crossing System

BRAC
UNIVERSITY

Inspiring Excellence

**Submitted by:**

| | |
|---|---|
| Umme Jannat Taposhi | 20101135 |

**Submitted to:**

Fairooz Nawar Khan

Lecturer, Brac University

**Date of submission:** 12.08.2023

# Advance Automated Rail Crossing System

## Introduction

In the modern world, automated rail crossing systems are crucial for increasing efficiency and safety. These technologies stop accidents and collisions with automobiles and pedestrians by automatically turning on warning signals and barriers as a train approaches. They make transportation easier and more dependable by lowering human error and traffic congestion. Automated rail crossing systems are crucial for guaranteeing secure and effective rail operations as our cities expand and our transportation needs rise. However, the fact that our country has not yet embraced the automatic rail crossing system raises significant concerns. To address this issue, we have designed and developed a model for an advanced rail crossing system using Arduino.

Our system uses two types of sensor. First one is an IR sensor to sense the distance of the train when it is at a safe distance from the crossing point and when the rail crossing is needed to block the vehicles of the road. When a train is coming, the IR sensor that is situated close to the railway lines can generate light waves in its direction. The sensor can calculate how far away the train is from the crossing by timing how long it takes for the reflected light to return. As a result, the rail crossing bar could slide and obstruct the road. The bar can rise once again to allow the vehicle to move after the train departs and travels to a safe distance.

Then we have used another sensor which is called ultrasonic sensors to detect humans or cars in front of the crossing bar. If there is a train coming and there are any humans or cars in front of the bar, the sensors will sense the object and show a signal on the nearby signal post with an alarm.

Through this project, we have demonstrated the effectiveness of using Arduino and sensors to create a simple and efficient advanced rail crossing system. By implementing this system in real-world scenarios, we hope to contribute to the optimization and improvement of the overall crossing experience as well as help to reduce the number of accidents.

## Project Details:

### <u>Sensors and Materials used:</u>

In our project we have used the Arduino board and basic electronic components to build the project. As this is a simple railway crossing gate system, we have used simple tools and technology which are:

1. **The Arduino board:** It is the microcontroller of the project.
2. **Infrared obstacle avoidance sensor**: It measures the distance between the train and the sensor. The train is detected as an obstacle by the IR sensor and the bar is lifted or closed accordingly.
3. **Ultrasonic Sensors**: When there are humans or cars in front of the bar, it senses the objects and warns about the incoming train.
4. **LED lights**: They display the outputs of the signal of the sensors. When there are humans or animals on the tracks, one of the LED lights up.
5. **Buzzer:** it works as an alarm when there is any human or animal on the railway track.

6. **Breadboard**: It is used to make connections between the components.
7. **Jumper wires:** They were used to connect the components together. Male to female and male to male jumper wires were used for the connections.
8. **Servo Motor:** It controls the movement of the barrier through which the train enters or leaves  based on the control signal sent from the arduino.

The technology that has been used is the Arduino IDE which is the Arduino Integrated Development Environment with which the code has been written and compiled and then uploaded to the Arduino board. Arduino IDE is an open source software which has a simple code editor which supports multiple libraries to program the Arduino boards.

## **Project working mechanism:**

We will have 2 IR sensors across the train track. The first IR sensor will detect whether the train is coming or not. This sensor will be a little ahead of the actual crossing. This sensor will detect the upcoming train, turn on the warning signaling light(yellow) and will start the action of the ultrasonic sensors which will be placed around the actual location of the crossing bar. These sensors will detect whether there is anyone, person or car, on the train tracks. If there is someone or something the sensors will detect it and immediately turn on the buzzer to warn about the incoming train. The first sensor will be a little further ahead than the crossing just to give some time to the people to move away from the track which the ultrasonic sensors will help by warning and turning the buzzer on. Then the second IR, which will be placed closer to the crossing than the first one, will detect the incoming train and will promptly shut down the gate of the crossing. Then the train will pass and when the train crosses the second IR sensor completely, it will signal that the train has crossed the crossing area and now the area is safe to open. So, according to the output of the second sensor, the gate will be lifted after some time has passed as the sensor is placed a little further away from the crossing. Hence, we can completely control the railway crossing without any human intervention and fully automatically.

## **Circuit diagram:**

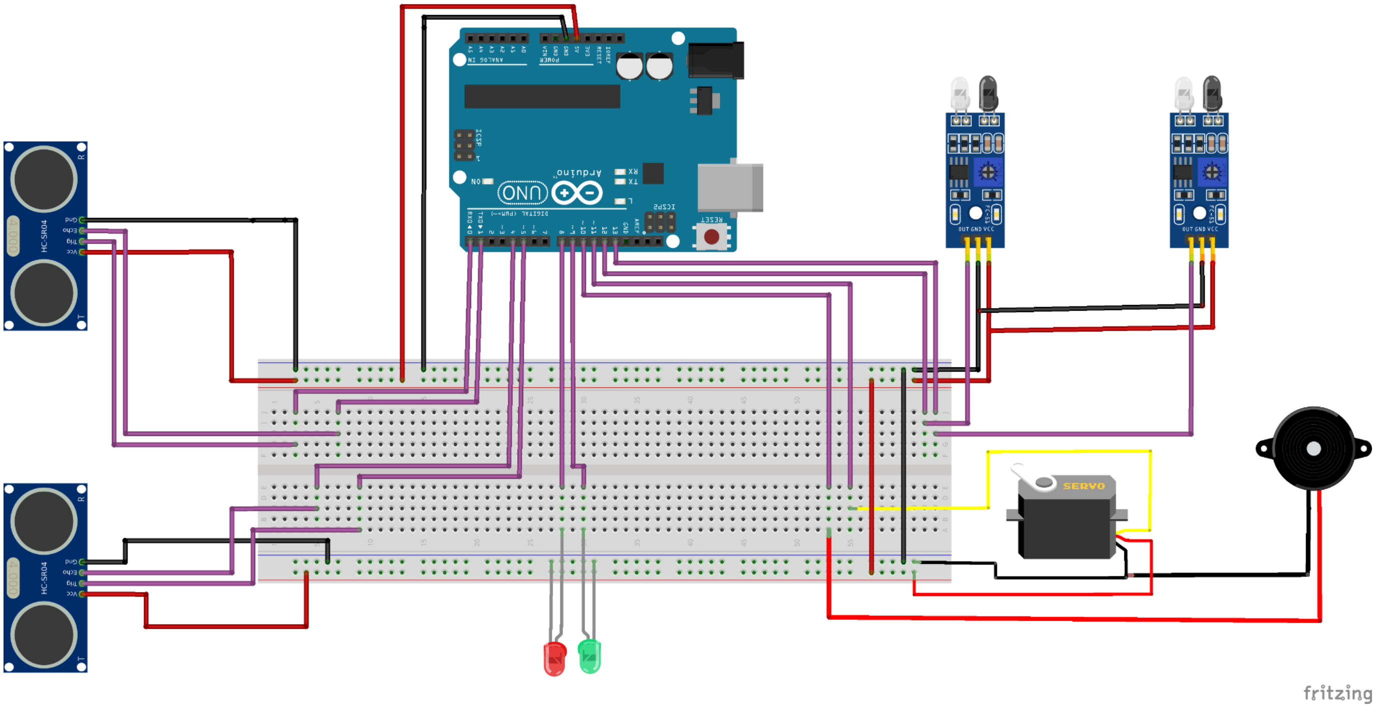

**Figure 1: Circuit diagram of the proposed system**

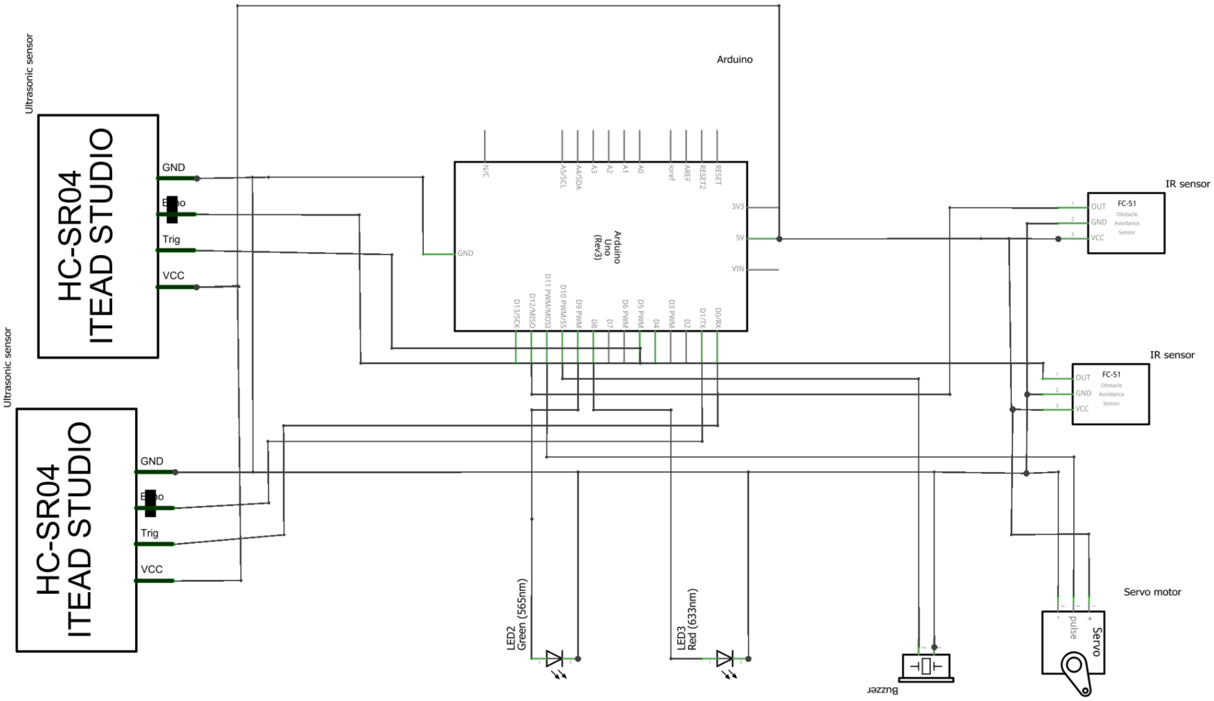

**Figure 2: Schematic circuit diagram of the proposed system**

## Project Cost analysis

| Component name | Unit Price (bdt) | Total Price (bdt) |
|---|---|---|
| Arduino Uno R3 | 1100 | 1100 |
| IR Infrared Obstacle Avoidance Sensor Module | 60 | 120 |
| Ultrasonic sensor | 93 | 186 |
| LED Light Module | 5 | 10 |
| Passive Buzzer | 15 | 15 |
| Full Metal Servo Motor | 435 | 435 |
| Train set | 1000 | 1000 |
| Others (jumper wire, breadboard, batteries, cardboard etc) | - | 500 |
|  |  | ∑ 3366 |

## Future Work

1) Incorporating with Train Control Systems: Future automated railway crossing systems are able to be easily connected with train control systems. This connection will improve train schedule optimisation and decrease wait times for automobiles and pedestrians by better coordinating approaching trains with the crossing devices.
2) Advanced Sensing Technologies: Ongoing research and development in sensing technologies may result in more advanced and precise sensors for spotting trains and other nearby objects. To improve detection skills, this may involve the deployment of cutting-edge lidar, radar, or AI-powered cameras.
3) Standardization and Regulations: To ensure interoperability, safety, and uniform application across many locations and countries, it will be essential to create standardized protocols and rules for automated railway crossing systems.

## Conclusion

In conclusion, the Arduino-based automated railway crossing project has demonstrated to be a practical and efficient solution to the rising issue of accidents in both urban and rural locations. The project's successful application of Arduino technology resulted in the development of a sophisticated railway crossing system that not only aids in the timely safety of the rail crossing spaces but also promotes the effective use of different factors that all the sensors rely on. The Arduino-based railway crossing system has proven accurate and reliable in detecting the presence of trains and human or animal object presence through a user-friendly interface by integrating a variety of sensors and components.

Overall, the initiative to build an advanced train crossing system using Arduino has offered a practical and affordable answer to the problem at hand. It has shown how using technology may help solve common problems and raise living standards for people in both urban and rural areas.

## Reference

1. theoryCIRCUIT. (2022, January 13). *Arduino IR Sensor Obstacle Detection*. theoryCIRCUIT - Do It Yourself Electronics Projects. https://theorycircuit.com/arduino-ir-sensor-obstacle-detection/
2. Dejan. (2022, February 18). *Ultrasonic Sensor HC-SR04 and Arduino – complete guide*. How to Mechatronics. https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/
3. Arduino. (n.d.). *Servo Motor Basics with Arduino | Arduino Documentation*. Arduino Docs. https://docs.arduino.cc/learn/electronics/servo-motors

CODE:

```
#include <Servo.h>

Servo myservo;

int trigPin1 = 10;  // Ultrasonic sensor trigger pin

int echoPin1 = 12;  // Ultrasonic sensor echo pin
// int trigPin2 = 2;  // Ultrasonic sensor trigger pin

// int echoPin2 = 3;

int IRSensor = 8; // warning

int IRSensor1 = 9; // gate

int LED_R = 2; // warning

int LED_G = 5; // green

int buz=6 ;// warning

int servoPin=7; // servo pin

void setup(){

  Serial.begin(115200); // Init Serial at 115200 Baud Rate.

  Serial.println("Serial Working"); // Test to check if serial is working or not

  pinMode(trigPin1, OUTPUT);

  pinMode(echoPin1, INPUT);
  // pinMode(trigPin2, OUTPUT);

  // pinMode(echoPin2, INPUT);

  pinMode(IRSensor, INPUT);

  pinMode(IRSensor1, INPUT);

  pinMode(LED_R, OUTPUT);

  pinMode(LED_G, OUTPUT);

  pinMode(buz,OUTPUT);

  myservo.attach(servoPin);

}


void loop(){
```

```
//IR1 warning
  digitalWrite(LED_G, HIGH);
  int sense=digitalRead(IRSensor);

  if (sense == 1) {

    digitalWrite(buz,LOW);
    digitalWrite(LED_R, LOW);




  }

  else{

     // Turn on the LED

     digitalWrite(LED_R, HIGH);

     digitalWrite(LED_G, LOW);


     // Clear the trigger pin

     digitalWrite(trigPin1, LOW);

     delayMicroseconds(2);


  // Send a 10us pulse to the trigger pin

    digitalWrite(trigPin1, HIGH);

    delayMicroseconds(10);

    digitalWrite(trigPin1, LOW);


  // Measure the duration of the pulse on the echo pin
    unsigned long duration1 = pulseIn(echoPin1, HIGH);

  // Calculate the distance in centimeters

     float distance1 = (duration1 * 0.0343) / 2;


     Serial.print("Distance: ");

     Serial.print(distance1);

     Serial.println(" cm");

  // Set a distance threshold for object detection

  float threshold = 20; // Adjust this value as needed

  // Check if an object is within the threshold distance

  if (distance1 <= threshold) {

    digitalWrite(buz,HIGH);

    tone(buz, 1000, 1000);

  } else {

     digitalWrite(buz,LOW);

  }

  delay(100); // Delay for stability

   }
```

```
//lathi uthabe namabe

  int sensorStatus = digitalRead(IRSensor1); // Set the GPIO as Input

  if (sensorStatus == 1) {


    myservo.write(180);

    Serial.println(" No Motion Detected!");

    delay(15);// print Motion Detected!



  }

  else  {



    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_G, LOW);
    myservo.write(90);
   delay(5000);
    Serial.println("left ir Motion Detected!");




    }


  delay(200);

}
```