BRAC UNIVERSITY

Inspiring Excellence

Department of Electrical and Electronic Engineering

Project Report Submission

Semester: Spring 24
Course Code: Embedded System Design Laboratory
Course Title: EEE373L

Project Title: AUTOMATIC RAILWAY GATE CONTROL

Group No: 04

Group members:

SL	***Name***	***ID***
1.	*Khairul Amin Sojib*	*21121003*
2.	*Sheikh Walid Hasan*	*21121022*
3.	*Mohammad Shibli Noman*	*21121009*
4.	*Mosammat Ramia Sauda*	*20221007*
5.	*Abdullah Al Fahad*	*20321050*
6.	*Ishrak Amin Joarder*	*21121034*

Date of Submission: 28-04-24

Abstract

Automating railway gates is important for making sure trains can move safely and efficiently.
In this report, we explain how we made a system to open and close train gates automatically using ATmega32 microcontroller and IR sensors. The system uses special sensors to see if a train is there or not and the gates close on their own when a train is coming. The system's design involves creating a circuit with a microcontroller and connecting it to an IR sensor,writing code for the ATmega32 microcontroller. The system was checked to make sure it worked well and could be depended on. The study found that using ATmega32 and IR sensors to automate railway gates works well, making it efficient and safer. The system was able to detect the presence of trains accurately and operate the gates automatically, reducing the risk of accidents and improving the flow of traffic across the crossing.

In summary, using ATmega32 and IR sensors to automate railway gates is a cheap and dependable way to make railway crossings safer and more efficient. The system can be changed to fit the needs of different railway crossings, making it a good way to make railways safer and more efficient.

Introduction

Railway crossings are very dangerous spots on the road where accidents between trains and vehicles can have really bad outcomes. In recent years, controlling railway gates with machines has become very important to make sure trains are safe and run smoothly. These systems use sensors and signs to gather information.The system can tell when trains and vehicles are nearby and close the gates automatically ensuring safety. In this report, we explain how we made a system to control railway gates automatically using ATmega32 microcontroller and IR sensors. The system uses special sensors to see if a train is coming. When it does, it opens and closes the gates by itself, making sure accidents are reduced.

Literature Review

Researchers have actively pursued the development of automatic railway gate control systems to bolster safety and efficiency at railway crossings. Drawing from the works of Hien and Thuy (2019), Prabhu et al. (2015), and Shalaby and El-Mohandes (2014), these endeavors have harnessed a variety of microcontroller and sensor technologies such as Arduino, ultrasonic, ATmega328, and IR. By integrating these technologies, these systems autonomously detect approaching trains and trigger gate closure mechanisms, thus minimizing reliance on human intervention and mitigating the risk of accidents. This automation not only reduces the potential for human error but also significantly decreases reaction times, leading to a smoother flow of traffic at railway crossings. Such advancements underscore the transformative potential of technology in enhancing railway safety and operational efficiency.

Scopes and Objectives

There are many aspects in a railway gate where an AVR microcontroller can be used to automate several aspects. For example;

- Train Detection
- Gate or Barrier Operation
- Safety features
- Communication
- Control and Monitoring
-

The objectives of this project are as follows:

- Use IR sensor to detect the presence of a train
- After detecting the train automatically operate the gates
- Reduce accidents and improve safety of railway crossing
- Make Railway operation more efficient

This project is very important because it could help make railway crossings safer and reduce the number of accidents that happen there, which is a big problem all over the world.According to the International Union of Railways (2021) in 2019, there were 2750 accidents at train crossings around the world. These accidents caused 755 deaths and 763 injuries. So, using machines to control railway gates can make railways safer.
Moreover, automating railway gate control systems will improve the efficiency of railway operations by reducing the time taken to manually operate the gates. This can result in reduced train delays, improved traffic flow, and increased productivity.

Apparatus and Software

Hardware components

1. ATMega32 Microcontroller with trainer board
2. Leds
3. DC motors
4. IR sensor

5. Breadboard
6. Servo Motors
7. Motor Driver
8. Jumper Wires

Softwares

1. CodeVision AVR
2. Proteus Design suite

Project Specification

Using an ATMega32 microcontroller and IR sensor automatic railway gate control system is designed to detect any train and operate the railway gates automatically. The system should have the following specifications:

1. Detection accuracy: The system should be able to accurately and reliably detect the presence of a train using IR sensors.
2. Gate operation: Without delaying the system should be able to control the gates after detecting the train
3. Safety: The system should be designed to reduce the risk of accidents at railway crossings.
4. Power consumption: The system should be designed to consume minimum power making the system most efficient possible
5. Cost-effectiveness: The system should be made in a way that doesn't cost too much and can be used at many different railway crossings.

Technical Requirements and Constraints:

To make sure the system works well, we need to consider certain technical requirements and constraints:

1. Microcontroller: For controlling the system At mega32 microcontroller should be used. e Using CodeVisionAVR, the program should be written in C language.
2. IR Sensors: it is used for detecting the presence of a train. The sensors should be placed at an appropriate distance (1 cm - 5 cm) from the railway track.
3. Gate Operation: The servo motors should be able to both open and close the gate automatically.
4. Power Supply: The system is designed to operate using a DC power supply.

5. Safety: The system needs to be made with safety as the top priority. If something goes wrong, the system should be able to switch to manual control to keep the railway crossing safe.
6. Cost: The system should be designed to be cost-effective, making it accessible to a wide range of railway crossings.

The automated railway gate system with ATmega32 microcontroller and IR sensors is made to make railway crossings safer and work better. The system needs to be designed to meet the above mentioned specific requirements and technical needs to work efficiently.

Non-technical constraints

Environmental Constraints:

- The system needs to be made so that it does not harm the environment, like making less noise and air pollution.
- The materials being used should be eco-friendly and environmentally friendly

Societal Constraints:

- impact on society should be considered that is safety of pedestrians and passengers
- The system should also cause as little trouble as possible for the people living in the area when it is being installed and used.

Cultural Constraints:

- System should respect the cultural values and beliefs of the communities living near the railway.
- The design should be sensitive to the cultural needs of the local population.

Political Constraints:

- The system must follow all the rules and laws set by the government related to railway safety and operation.
- The system should meet the standards set by the regulatory bodies responsible for railway safety.

Ethical Constraints:

- The system needs to be made so that accidents are stopped and everyone's safety on the railway is guaranteed.
- privacy of passengers should be maintained and personal data should not be used inappropriately.

Health and Safety Constraints:

- The system needs to focus on keeping workers safe and healthy while they install, maintain, and use it.
- The system should prevent accidents and injuries to people working and using the system

Sustainability Constraints:

- The system should use energy efficiently and minimize waste.
- The things used in the system design should be able to be recycled and used again to help the environment.

Methodology

Our projects consist of many subsystems. They are given below:

1. Detection
2. Processing
3. Response
4. Emergency Override

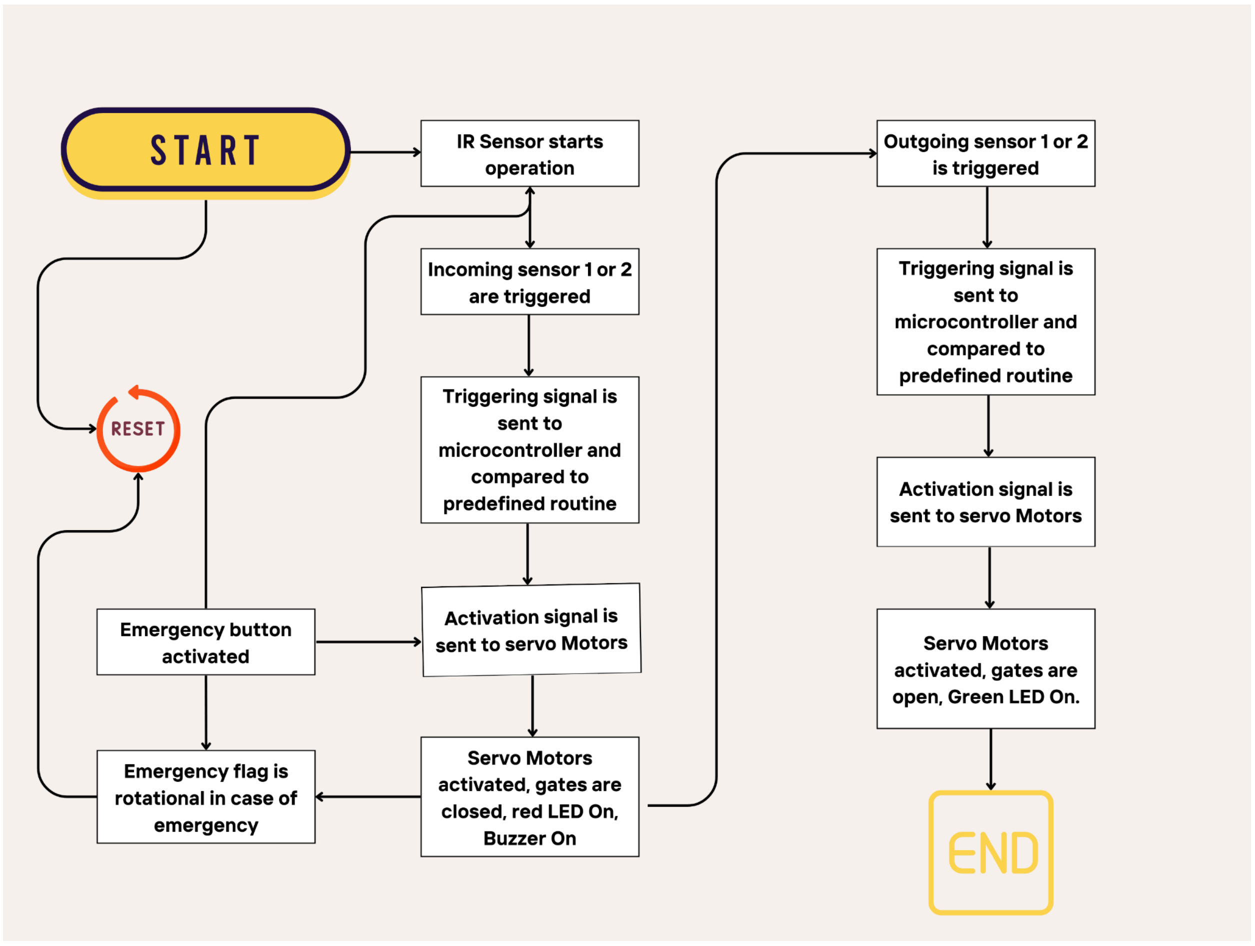
1. **Detection:** In this subsystem we use IR sensors for the detection of the train. These IR sensors are divided into two sections. When a train arrives the IR sensor of the incoming section detects it and when the train leaves the junction the IR sensor of the outgoing section detects it.

2. **Processing:** When the detection unit detects any train it triggers the certain PIN in the microcontroller. Then the microcontroller compares the triggered signal to the predefined condition and then sends commands to the response subsystem.

3. **Response:** The response unit consists of signal LED, a buzzer and two servo motors. According to the processing units order they activated or deactivated for the safety of the pedestrian and vehicle.

4. **Emergency Override:** There is an emergency override subsection which also has access to the signal LED, buzzer, servo motor and it also has an additional emergency flag. When an emergency occurs, by pressing only one button it can immediately stop the gate , turn on the red signal LED and buzzer. It also raises the emergency flag in time of emergency.

Flowchart of the project:


START
IR Sensor starts operation
Incoming sensor 1 or 2 are triggered
Triggering signal is sent to microcontroller and compared to predefined routine
Activation signal is sent to servo Motors
Servo Motors activated, gates are closed, red LED On, Buzzer On
RESET
Emergency button activated
Emergency flag is rotational in case of emergency
Outgoing sensor 1 or 2 is triggered
Triggering signal is sent to microcontroller and compared to predefined routine
Activation signal is sent to servo Motors
Servo Motors activated, gates are open, Green LED On.
END

Design Choices and Alternatives:

Design Approach 1- Unidirectional

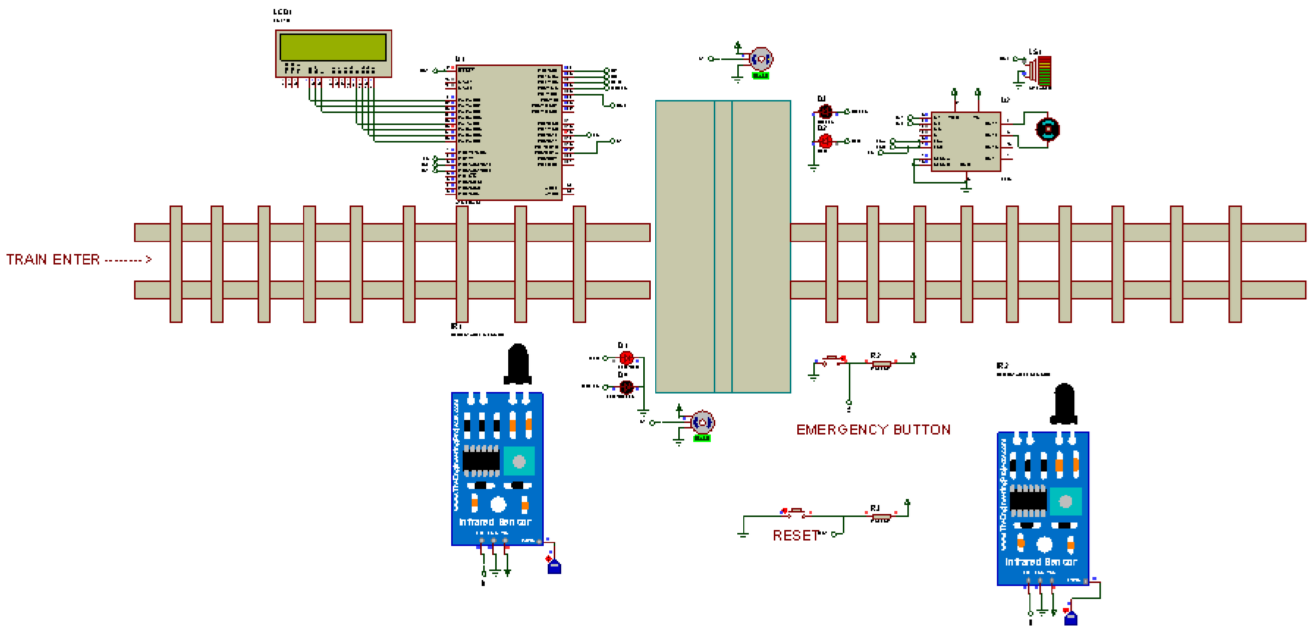


Figure: Design Approach 1

Our first design consists of two IR sensors for the detection purpose. One of these sensors detect the train before it arrives at the intersection. The signal of the sensor is sent to the microcontroller. The microcontroller senses the signal and compares it to the predefined code. Then, the microcontroller sends control signals to the servo motors and they rotate to close the gates. During this time, a buzzer and red signal LEDs remain operational. After the train passes the intersection, it is detected by the second IR sensor along the path. The sensor is triggered and sends a signal to the microcontroller. The microcontroller senses the signal and opens the gates by rotating the servos. An emergency switch is connected to the INT0 pin, which runs the gate closing operation upon pressing. A dc motor also rotates during this state which contains an emergency flag. The limitation with this design is that it only operates unidirectionally. Meaning, the system can work only when the train is coming from a certain direction.

Code:

```
#include <mega32.h>

#include <delay.h>

#include <alcd.h>



interrupt[EXT_INT0] void ext_int0_isr(void) {
  PORTB .1 = 1; //ENA
  PORTB .2 = 0;
  PORTC .2 = 1;
  PORTC .3 = 0;
  PORTC .4 = 1;
  lcd_clear();
  lcd_gotoxy(0, 0);
  lcd_putsf("EMERGENCY");
  OCR1A = 5200;
  delay_ms(10);
  while (1) {
    OCR0 = 35;
    delay_ms(5000);
  }
}
void main(void) {
  lcd_init(16);
  lcd_clear();
  DDRA = 0xFF;
  DDRB = 0xFF;
  DDRC = 0b11111100;

  DDRD .5 = 1; //SERVO
```

```
 GICR = (0 << INT1) | (1 << INT0) | (0 << INT2);
  MCUCR = (0 << ISC11) | (0 << ISC10) | (1 << ISC01) | (0 << ISC00);
  TCCR0 = (1 << WGM00) | (1 << COM01) | (0 << COM00) | (1 << WGM01) | (1 << CS02) |
(0 << CS01) |(1 << CS00);

 TCCR1A = (1 << COM1A1) | (0 << COM1A0) | (0 << COM1B1) | (0 << COM1B0) |
     (1 << WGM11) | (0 << WGM10);
 TCCR1B = (0 << ICNC1) | (0 << ICES1) | (1 << WGM13) | (1 << WGM12) | (0 << CS12) |
     (1 << CS11) | (0 << CS10);

  ICR1H = 0x9C;
  ICR1L = 0x40;
  OCR1A = 2000;

  PORTB .0 = 0;
  PORTB .3 = 0;
  PORTB .4 = 0;
  PORTC .3 = 1;
  PORTC .4 = 0;
  lcd_gotoxy(0, 0);
  lcd_putsf("GO");
  // Global enable interrupts
  #asm("sei")
  while (1) {
    if (PINC .0 == 1 && PINC .1 == 0) {
      PORTC .2 = 1;
      PORTC .3 = 0;
      PORTC .4 = 1;
      lcd_clear();
      lcd_gotoxy(0, 0);
      lcd_putsf("STOP");
      OCR1A = 5200;
      delay_ms(10);
    }
    if (PINC .0 == 1 && PINC .1 == 1) {
      PORTC .2 = 1;
      PORTC .3 = 0;
      PORTC .4 = 1;
      lcd_clear();
      lcd_gotoxy(0, 0);
      lcd_putsf("STOP");
      OCR1A = 5200;
      delay_ms(10);
    }
    if (PINC .0 == 0 && PINC .1 == 1) {
      PORTC .2 = 0;
      PORTC .3 = 1;
      PORTC .4 = 0;
      lcd_clear();
      lcd_gotoxy(0, 0);
      lcd_putsf("GO");
      OCR1A = 2000;
      delay_ms(10);
    }
    lcd_clear();
  }
}
```

Design Approach 2- Multidirectional

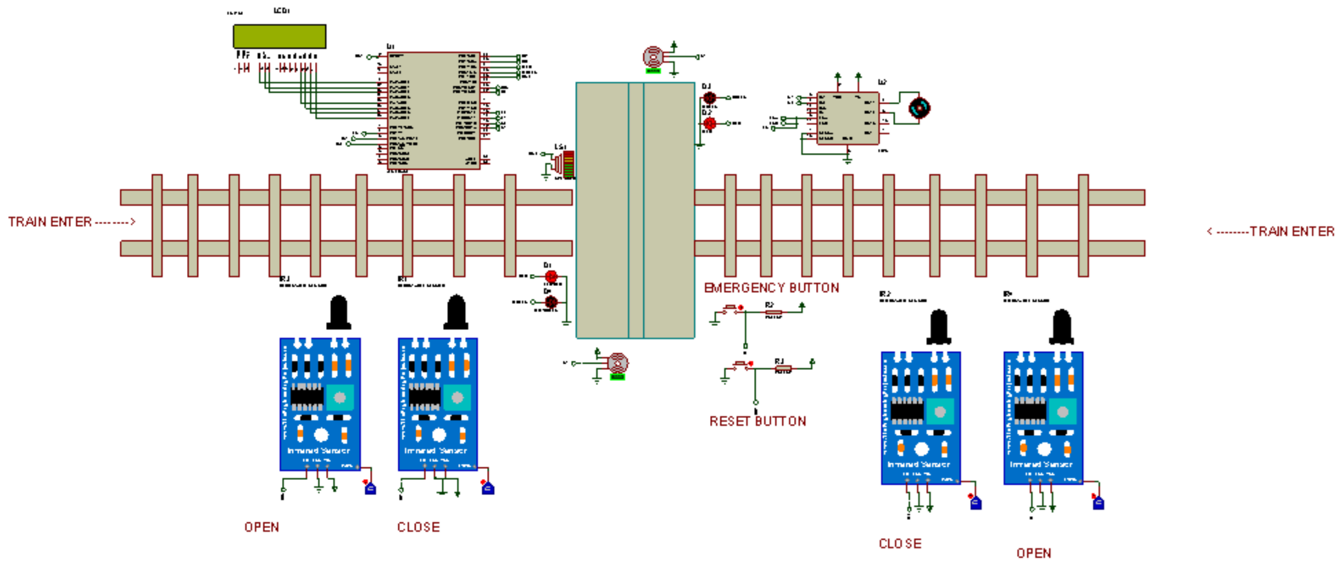


Figure: Design Approach 2

The second design is an attempt to overcome the limitations of the first design. In this design, we used four Infrared sensors, instead of two. These two are connected to two additional pins. While most of the functionalities from the previous design are restored in this one, there are also major alterations. The four IR sensors are set up in a way that two of them work to detect the train when it arrives and the other two work to detect it when it departs. Now, as the number of sensors have doubled, the complexity is increased in the source code. We had to configure and list all the states possible for four sensors and had to define the system's operation for those states. This resulted in a noticeable larger code. The system can now operate for a train appearing from any side. A larger code and two additional sensors might have solved the previous design's problem, but the code is still large. Which is why we moved on to the third iteration.

Code:

```
#include <mega32.h>

#include <delay.h>

#include <alcd.h>


interrupt[EXT_INT0] void ext_int0_isr(void) {

        PORTB .1 = 1; //ENA

        PORTB .2 = 0;

        PORTC .2 = 1;

        PORTC .3 = 0;

        PORTC .4 = 1;

        lcd_clear();

        lcd_gotoxy(0, 0);

        lcd_putsf("EMERGENCY");
        OCR1A = 5200;

        delay_ms(10);

        while (1) {

        OCR0 = 35;

        delay_ms(5000);

        }

}

void main(void) {

        lcd_init(16);

        lcd_clear();

        DDRA = 0xFF;

        DDRB = 0xFF;

        DDRC = 0b00111100; //INPUT PC0,1,6,7  PIN


        DDRD .5 = 1;  //SERVO



        GICR = (0 << INT1) | (1 << INT0) | (0 << INT2);

        MCUCR = (0 << ISC11) | (0 << ISC10) | (1 << ISC01) | (0 << ISC00);

        TCCR0 = (1 << WGM00) | (1 << COM01) | (0 << COM00) | (1 << WGM01) | (1 << CS02) | (0
<< CS01) |

        (1 << CS00);

        TCCR1A = (1 << COM1A1) | (0 << COM1A0) | (0 << COM1B1) | (0 << COM1B0) |

        (1 << WGM11) | (0 << WGM10);

        TCCR1B = (0 << ICNC1) | (0 << ICES1) | (1 << WGM13) | (1 << WGM12) | (0 << CS12) |

        (1 << CS11) | (0 << CS10);
```

```
ICR1H = 0x9C;

ICR1L = 0x40;

OCR1A = 2000;


#asm("sei")

while (1) {


if (PINC .0 == 1 && PINC .1 == 0&& PINC .6 == 1 && PINC .7 == 0) {

        PORTC .2 = 1;

        PORTC .3 = 0;

        PORTC .4 = 1;

       lcd_clear();

        lcd_gotoxy(0, 0);

    lcd_putsf("STOP");

        OCR1A = 5200;

        delay_ms(10);

}

if (PINC .0 == 1 && PINC .1 == 1&& PINC .6 == 0 && PINC .7 == 0) {

        PORTC .2 = 1;

        PORTC .3 = 0;

        PORTC .4 = 1;

        lcd_clear();

        lcd_gotoxy(0, 0);

    lcd_putsf("STOP");
```

```
        OCR1A = 5200;

        delay_ms(10);

}

        if (PINC .0 == 0 && PINC .1 == 1&& PINC .6 == 0 && PINC .7 == 0) {

        PORTC .2 = 0;

        PORTC .3 = 1;

        PORTC .4 = 0;

        lcd_clear();

        lcd_gotoxy(0, 0);

        lcd_putsf("GO");

        OCR1A = 2000;

        delay_ms(10);

}

if (PINC .0 == 0 && PINC .1 == 0 && PINC .6 == 0 && PINC .7 == 1) {

        PORTC .2 = 0;

        PORTC .3 = 1;

        PORTC .4 = 0;

        lcd_clear();

        lcd_gotoxy(0, 0);

        lcd_putsf("GO");

        OCR1A = 2000;

        delay_ms(10);


}
```

```
if (PINC .0 == 0 && PINC .1 == 1 && PINC .6 == 0 && PINC .7 == 1) {
            PORTC .2 = 1;
            PORTC .3 = 0;
   PORTC .4 = 1;
            lcd_clear();
            lcd_gotoxy(0, 0);
      lcd_putsf("STOP");
            OCR1A = 5200;
            delay_ms(10);
}
if (PINC .0 == 1 && PINC .1 == 1 && PINC .6 == 0 && PINC .7 == 0) {
            PORTC .2 = 1;
            PORTC .3 = 0;
            PORTC .4 = 1;
            lcd_clear();
            lcd_gotoxy(0, 0);
      lcd_putsf("STOP");
            OCR1A = 5200;
            delay_ms(10);
}

if (PINC .0 == 0 && PINC .1 == 0 && PINC .6 == 1 && PINC .7 == 0) {
            PORTC .2 = 0;
            PORTC .3 = 1;
```

```
        PORTC .4 = 0;

        lcd_clear();

        lcd_gotoxy(0, 0);

        lcd_putsf("GO");

        OCR1A = 2000;

        delay_ms(10);


}

if (PINC .0 == 0 && PINC .1 == 0 && PINC .6 == 0 && PINC .7 == 0) {

        PORTC .2 = 0;

        PORTC .3 = 1;

        PORTC .4 = 0;

        lcd_clear();

        lcd_gotoxy(0, 0);

        lcd_putsf("GO");

        OCR1A = 2000;

        delay_ms(10);


}


}

}
```

Design Approach 3- Multidirectional and efficient

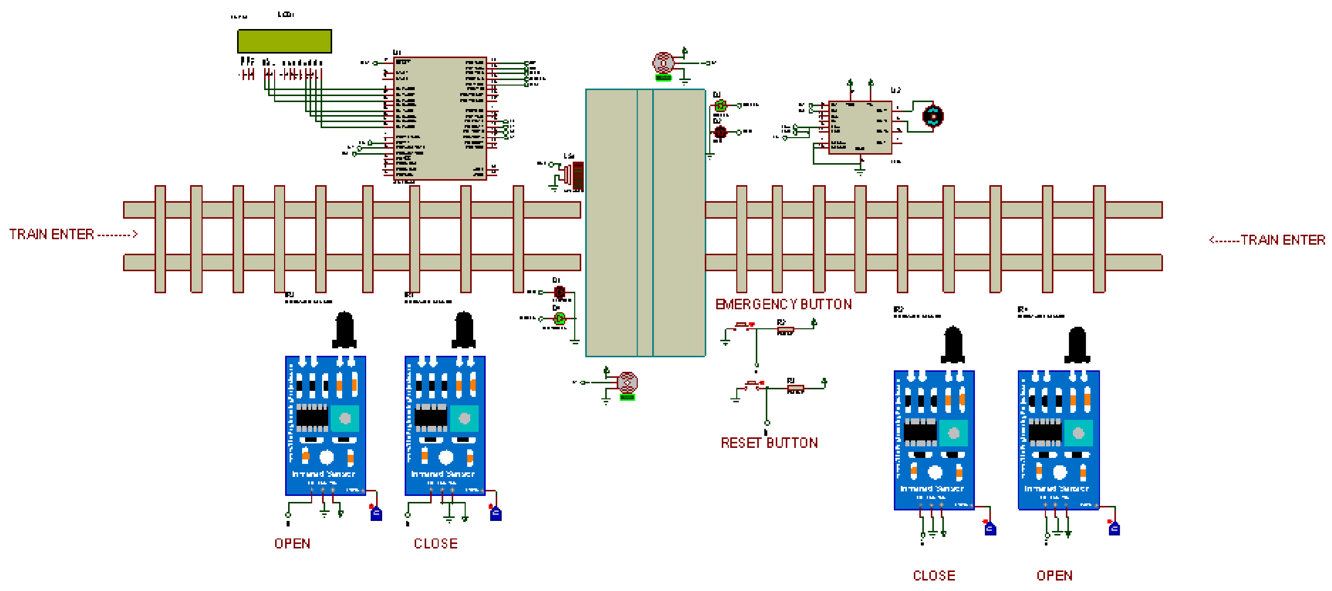


Figure: Design Approach 3

The third and final design is our attempt to simplify the system and make it more efficient than the previous design. In this design, the number of IR sensors is still four, but the source code used to operate the system is much shorter. This was achieved by bringing a fundamental change in our circuit design. In the previous design, the IR sensors were connected in different pins while in this design, we took two IR sensors, shorted them and connected them to the same pin. We did this for both sides. Now, our system is operational with the code from the first design, which significantly reduced the length of the source code.

So, to summarize, in this design, the system can detect a train coming from any side. Then it sends a signal to the microcontroller which then sends control signals to the servo motor and they close the gate. When the train passes the intersection, the IR sensor on the other side detects it and opens the gates. Both of these states are indicated to the vehicles and pedestrians by using red, green LEDs and a buzzer. In addition, during any emergency, pressing the emergency switch will instantly close the gates and initiate the emergency flag which is connected to the DC motor.

Code:

```
#include <mega32.h>

#include <delay.h>

#include <alcd.h>


interrupt[EXT_INT0] void ext_int0_isr(void) {

  PORTB .1 = 1; //ENA

  PORTB .2 = 0;

  PORTC .2 = 1;

  PORTC .3 = 0;

  PORTC .4 = 1;

  lcd_clear();

  lcd_gotoxy(0, 0);

  lcd_putsf("EMERGENCY");

  OCR1A = 5200;

  delay_ms(10);

  while (1) {

        OCR0 = 35;

        delay_ms(5000);

  }

}
```

```
void main(void) {

  lcd_init(16);

  lcd_clear();

  DDRA = 0xFF;

  DDRB = 0xFF;

  DDRC = 0b11111100;


  DDRD .5 = 1; //SERVO




 GICR = (0 << INT1) | (1 << INT0) | (0 << INT2);

 MCUCR = (0 << ISC11) | (0 << ISC10) | (1 << ISC01) | (0 << ISC00);

TCCR0 = (1 << WGM00) | (1 << COM01) | (0 << COM00) | (1 << WGM01) | (1 <<     CS02) |
(0 << CS01) |(1 << CS00);

 TCCR1A = (1 << COM1A1) | (0 << COM1A0) | (0 << COM1B1) | (0 << COM1B0) |

          (1 << WGM11) | (0 << WGM10);

 TCCR1B = (0 << ICNC1) | (0 << ICES1) | (1 << WGM13) | (1 << WGM12) | (0 << CS12) |

          (1 << CS11) | (0 << CS10);

  ICR1H = 0x9C;

  ICR1L = 0x40;

  OCR1A = 2000;

   PORTB .0 = 0;

  PORTB .3 = 0;

  PORTB .4 = 0;
```

```
PORTC .3 = 1;

PORTC .4 = 0;

lcd_gotoxy(0, 0);

lcd_putsf("GO");

// Global enable interrupts

#asm("sei")

while (1) {

      if (PINC .0 == 1 && PINC .1 == 0) {

      PORTC .2 = 1;

          PORTC .3 = 0;

      PORTC .4 = 1;

      lcd_clear();

      lcd_gotoxy(0, 0);

    lcd_putsf("STOP");

      OCR1A = 5200;

      delay_ms(10);

      }

      if (PINC .0 == 1 && PINC .1 == 1) {

      PORTC .2 = 1;

      PORTC .3 = 0;

      PORTC .4 = 1;

      lcd_clear();

      lcd_gotoxy(0, 0);
```

```
        lcd_putsf("STOP");
          OCR1A = 5200;
          delay_ms(10);
          }
          if (PINC .0 == 0 && PINC .1 == 1) {
          PORTC .2 = 0;
          PORTC .3 = 1;
          PORTC .4 = 0;
          lcd_clear();
          lcd_gotoxy(0, 0);
        lcd_putsf("GO");
          OCR1A = 2000;
          delay_ms(10);
          }
          lcd_clear();
  }
}
```

Justification of best approach:

After designing a couple of iterations of the project, we’ve come to conclude that the design approach 3 is the best approach.

The third design is simpler and more efficient than the previous designs. By shorting two IR sensors and connecting them to the same pin, the number of pins required for sensor connections is reduced. This reduces the complexity of the source code significantly, making it easier to manage and debug. Additionally, the simpler circuit design makes it easier to manufacture and assemble the system.This design is more flexible than the previous designs. It can detect a train coming from any side, which makes it suitable for various situations and scenarios. This design is also more cost-effective than the previous designs. By using fewer pins and a simpler circuit design, the cost of components and assembly is reduced. This makes it easier to manufacture and deploy the system in various locations.

Circuit Diagram:

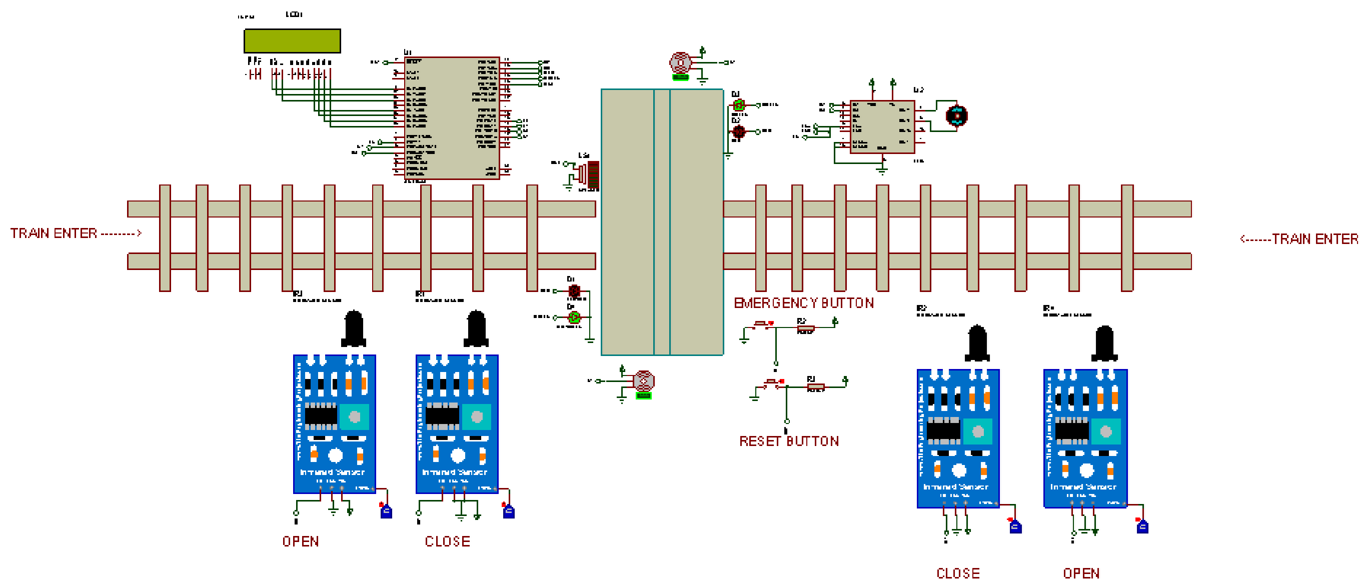


Figure: The final circuit diagram of the project

Results and Discussion

We have been installing an automatic railway gate system since the beginning of this project. The railway gate is supposed to shut on its own. Additionally, to alert surrounding people and automobiles as a train approaches, we plan to employ traffic lights, an instruction display, and a sound alarm system. Based on our Proteus and hardware connections, the first and third infrared sensors are positioned to detect a train moving from the right to the left, while the second and fourth sensors detect in various directions. We chose infrared sensors for this hardware arrangement because they continuously pulse when they detect an item. Despite our first attempts to employ PIR sensors, we found that IR sensors were more effective for our purposes due to the continuous nature they possess. To recreate genuine railway gates, we used servos for direction and programmed the angle for the appropriate spot. The atmosphere produced by the traffic lights, instruction display, and buzzer notifies oncoming automobiles and the general public when a train is ready to cross a double-sided road. We have designed the system such that it may be manually interrupted using switches. An interrupt switch is used as an emergency switch in case of an emergency. It immediately stops all automobiles and notifies an emergency display, alerting the public with an emergency sign motion. The reset switch can be used to reset the system when there are no train crossings.

Applications

The automated railway gate control system is employed in the railway industry for several purposes, including:

1. Improving railway safety: The technology lowers the likelihood of accidents caused by human error by automatically opening and closing the gates at railway crossings.

2. Improving traffic flow: The technology reduces traffic congestion at railway crossings by reducing the time required to open and close the gates.

3. Reducing operational costs: The technology lowers operating costs by doing away with the requirement for human gate operators at train crossings.

4. Increasing efficiency: The technology helps the railway industry become more efficient by reducing the amount of time it takes to manually operate gates at railroad crossings.

Stakeholders:

The following parties are involved in the automated railway gate control system:

1. Bangladesh Railway: As the nation's train operator and caretaker of railway crossings, the Bangladesh Railway is the major aim of the system.

2. Bangladesh Railway Authority: This organization oversees and establishes guidelines for railway operations throughout the nation and has to be consulted throughout system design and implementation.

3. Local communities: The system's increased safety and decreased noise pollution may be advantageous to the nearby communities that live near railroad crossings.

4. Ministry of Railways: The Ministry of Railways is in charge of monitoring railroad activities around the nation and would be curious about how well the system works to increase safety at railroad crossings.

5. Government organizations: Organizations in charge of transportation safety, including the Bangladesh Road Transport Authority (BRTA), can also be curious about the advantages of the system.

Effective installation of the automated railway gate control system in Bangladesh requires communication with these stakeholders. During system design and implementation, Bangladesh Railway and the Bangladesh Railway Authority should be consulted to ensure compliance with legislation and industry standards. The benefits of the system should be communicated to the surrounding communities, along with any potential installation-related problems. The Ministry of Railways and other governmental bodies should also be consulted in order to ensure collaboration and support for the project.

Future Scope

In the future, railway gate automation may be employed in several ways, such as:

Integration with other intelligent transportation systems: Two examples of other intelligent transportation systems that railway gate automation can be integrated with are intelligent transportation systems and traffic signal control systems. This integration could improve traffic flow and reduce congestion at railroad crossings.

Use modern sensors: Advanced sensors like LiDAR and radar can be used at railroad crossings to detect vehicles and people. These sensors' capacity to provide data that is more dependable and accurate than that of conventional sensors can improve the safety and efficacy of railway gate automation.

Application of machine learning and artificial intelligence: By evaluating data from numerous sensors and making choices in real time, these technologies can be applied to increase the effectiveness of railway gate automation.

Use of autonomous trains: To further increase efficiency and safety, autonomous trains can be combined with automation systems for railroad gates. In order to minimize the amount of time that cars and pedestrians must wait at a railway crossing, autonomous trains can interact with the automation system to ascertain the best moment to cross.

Google Drive Link: (The link contains all the code, proteus file and project demonstration videos)

https://drive.google.com/drive/folders/1qXpIWmiiGIE8iF9JwhV9YYNUXTMfWFxE?usp=sharing

Picture by shibli :

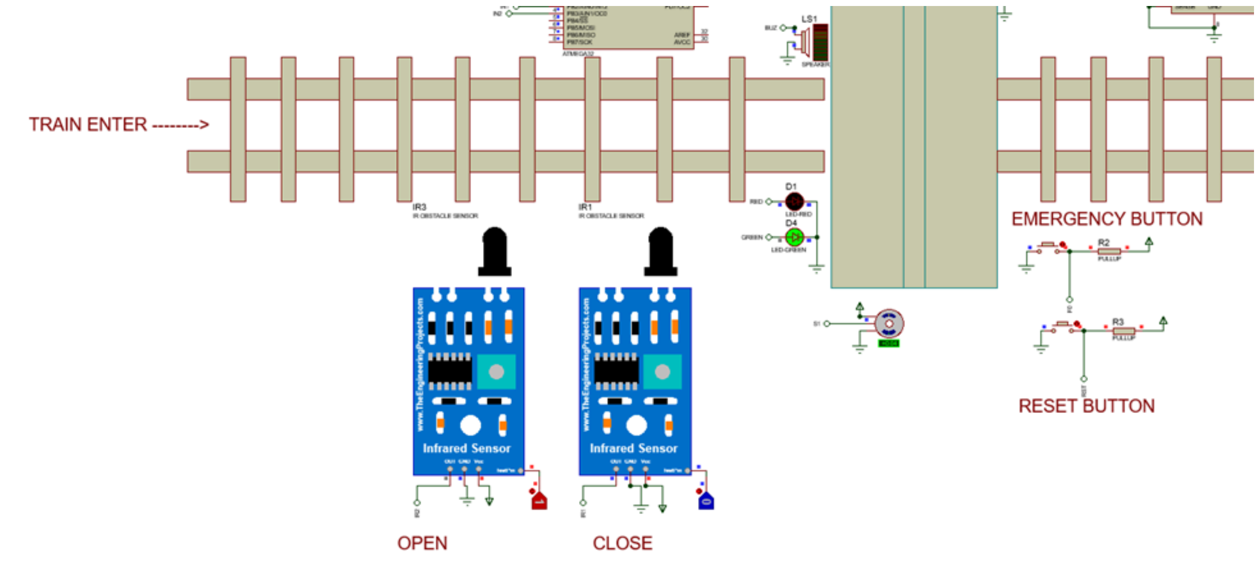


Figure: Go state of the System

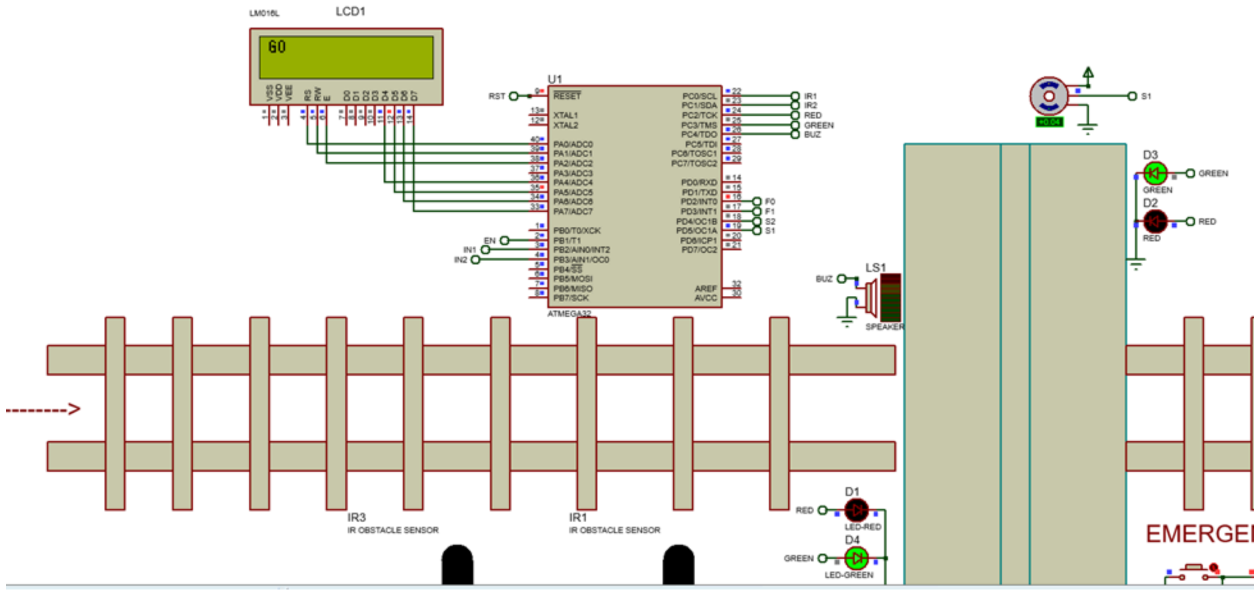


Figure: The writing of the seven segment display and the state of Buzzer when the system in “GO” state

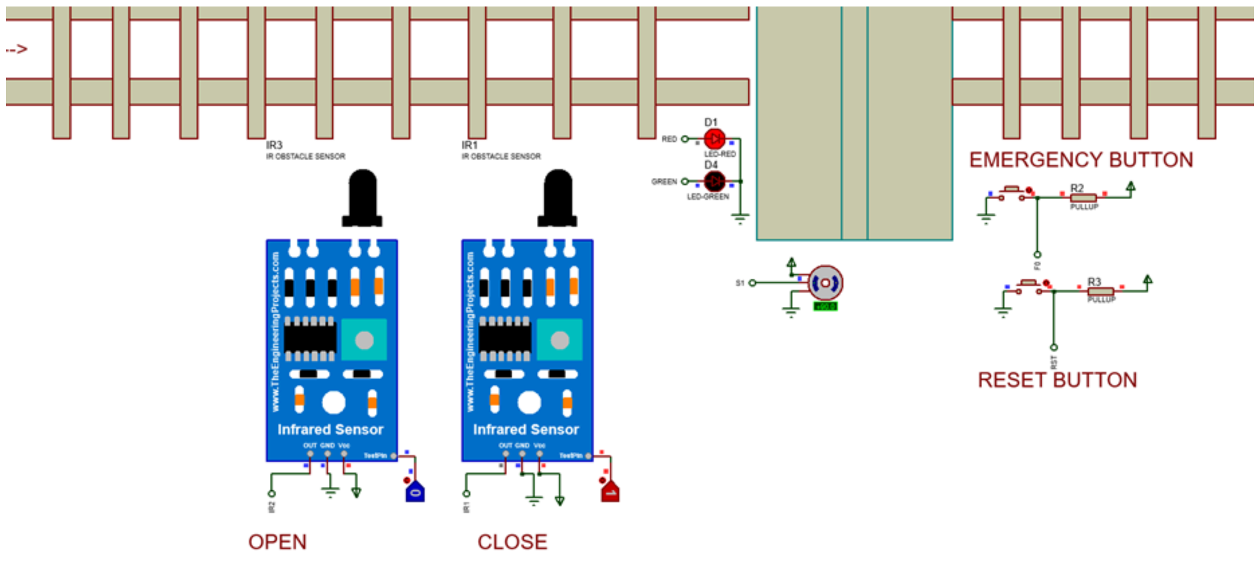


Figure: Stop state of the System{ there is a error in doc file)

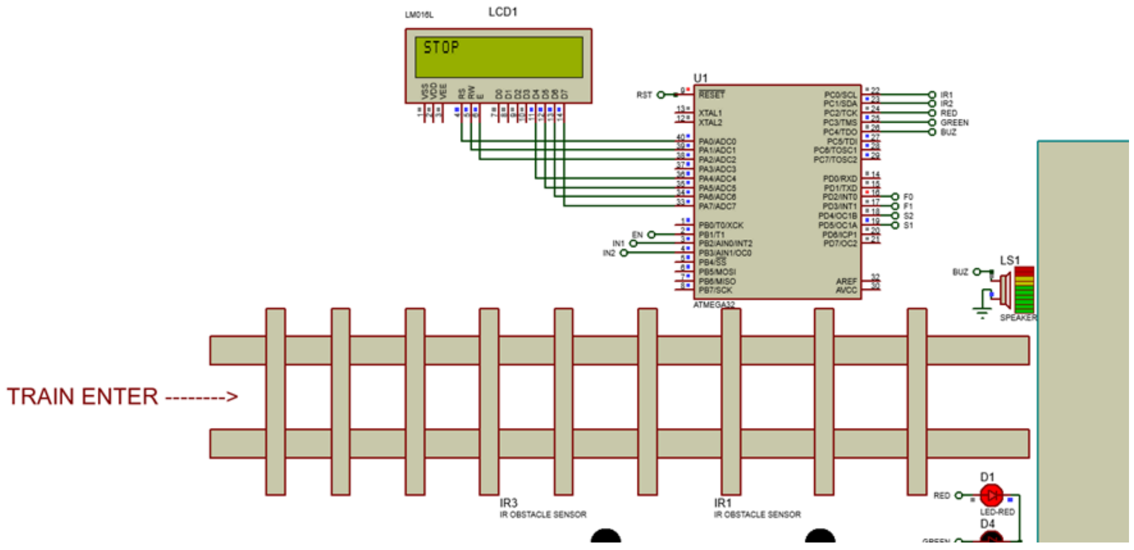


Figure: The writing of the seven segment display and the state of Buzzer when the system in "STOP" state

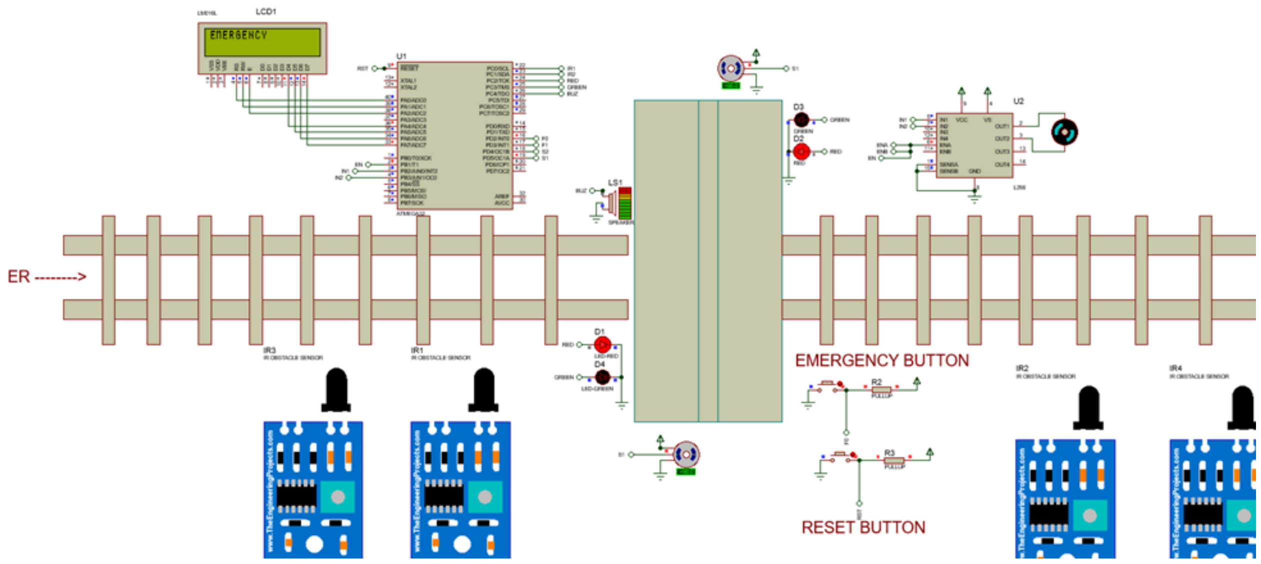


Figure: The emergency state

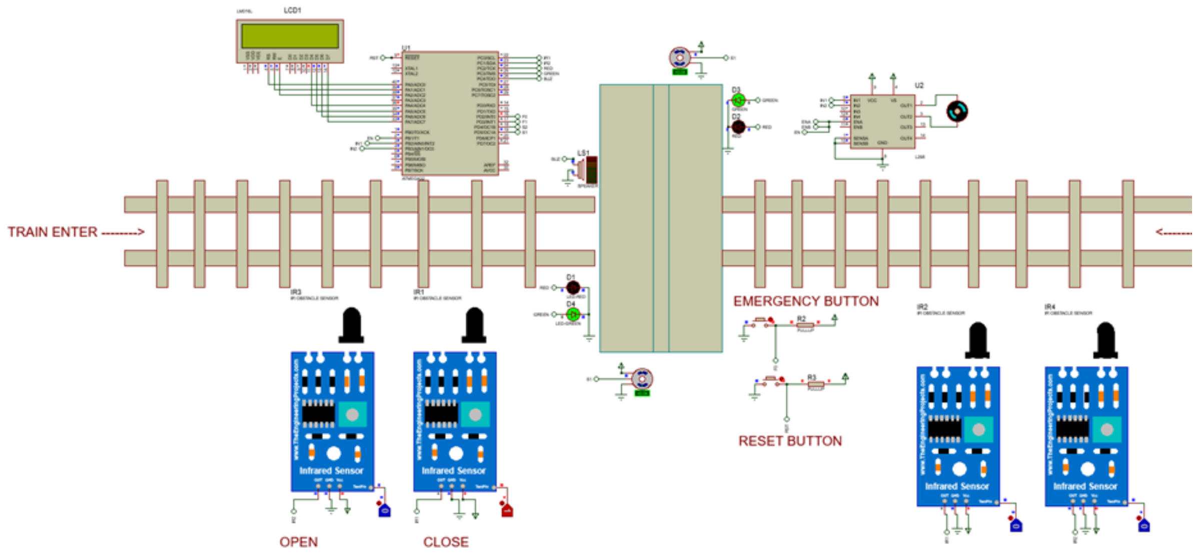


Figure: The Reset State

Test Case:

Approach 1:

STOP state:

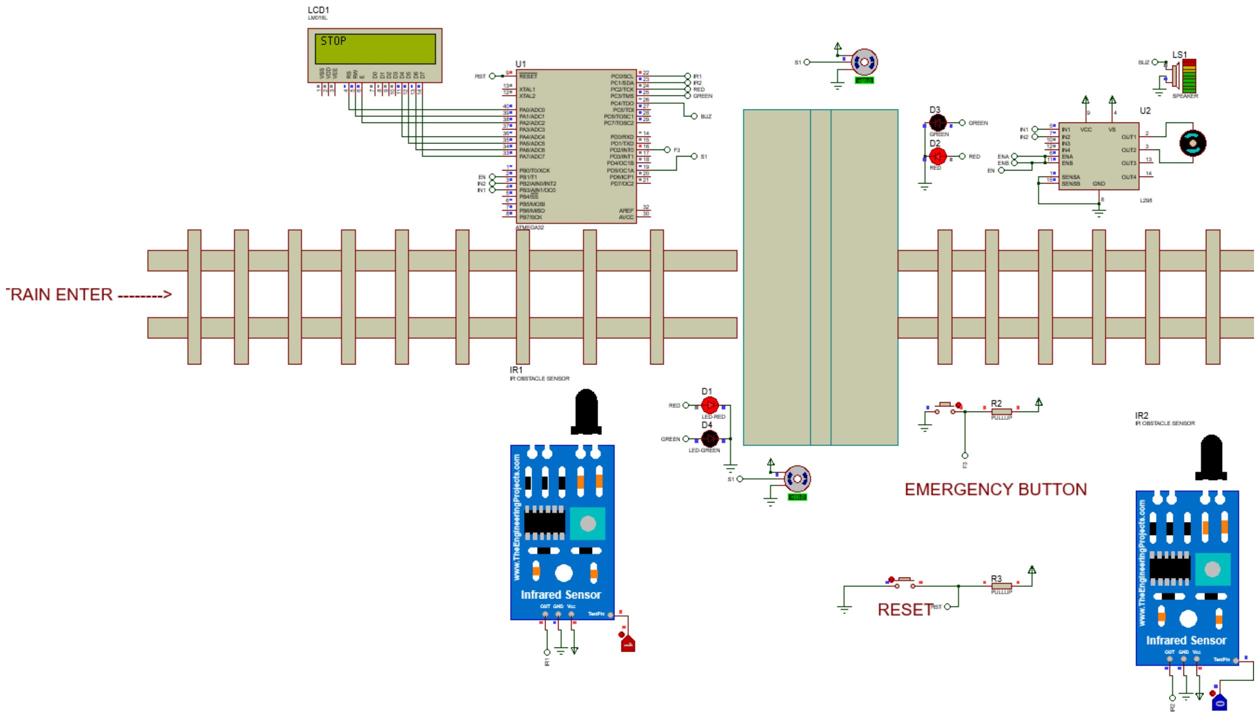


Figure: Stop state of the System

Please write the description walid

Go state:

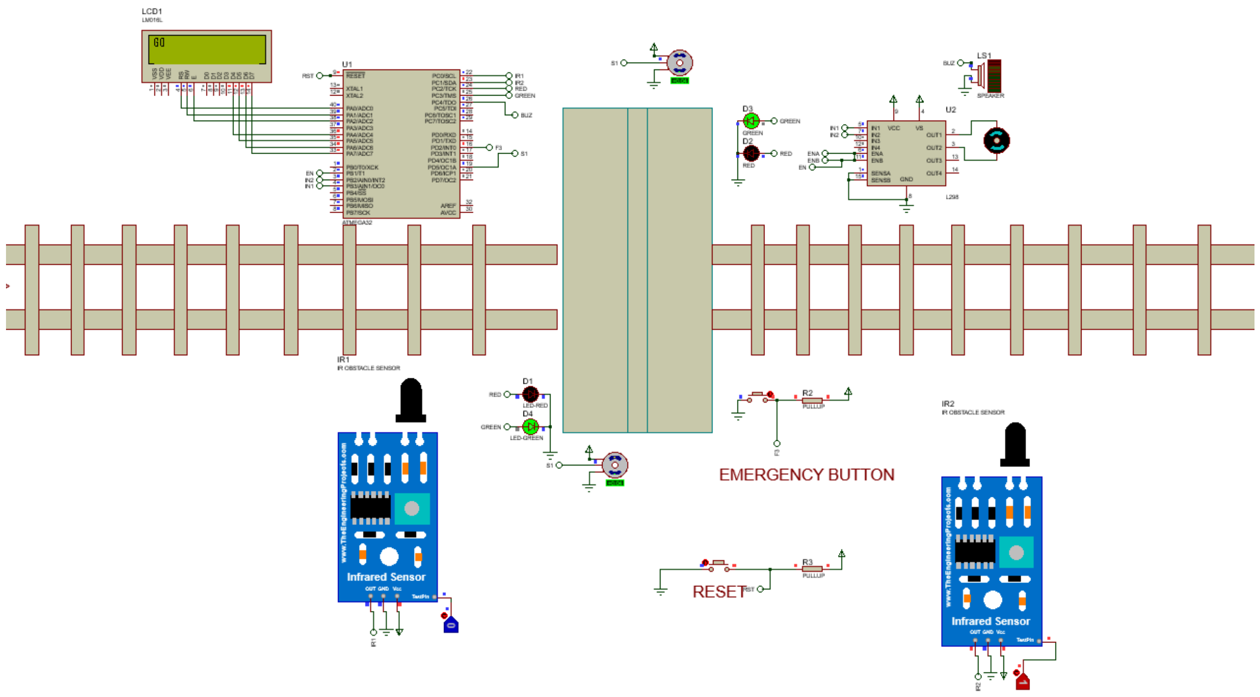


Figure: "GO" state of the System

Emergency State:

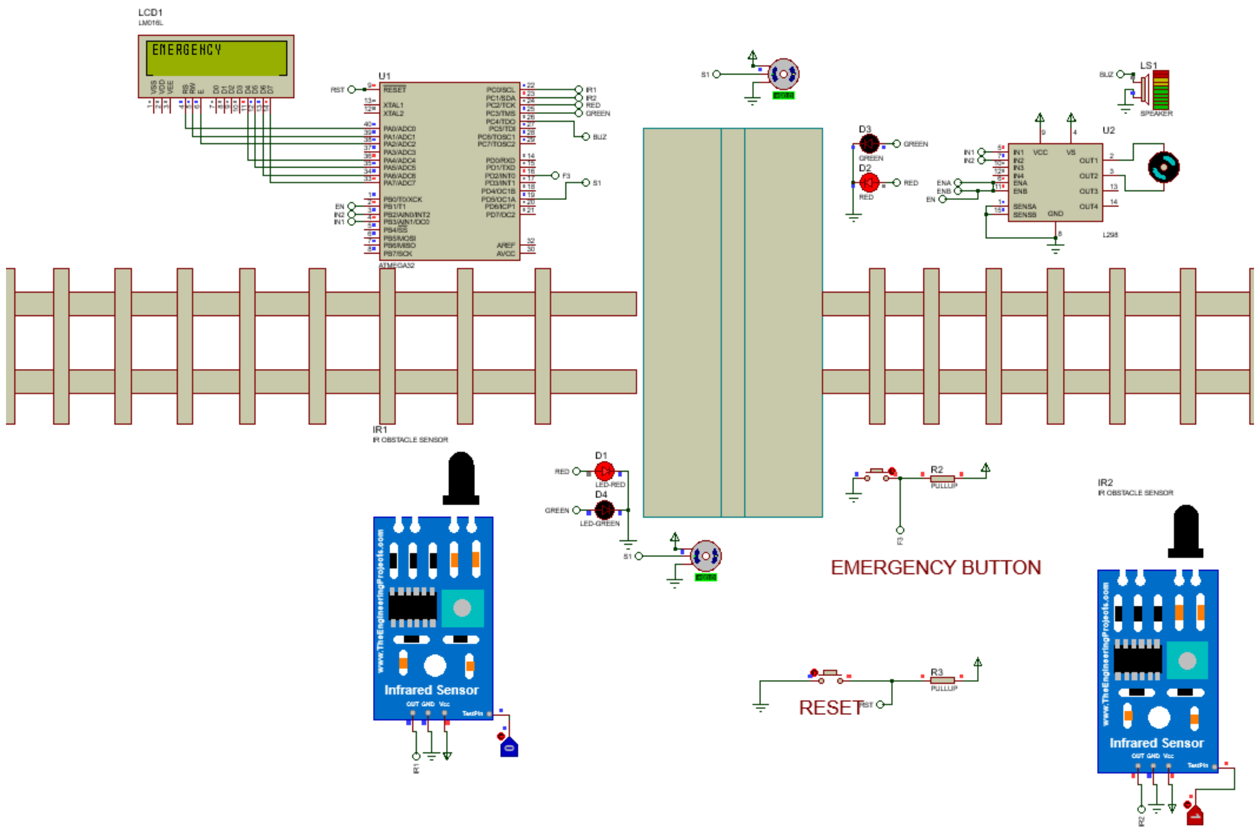


Figure: "Emergency " state of the System

Reset state:

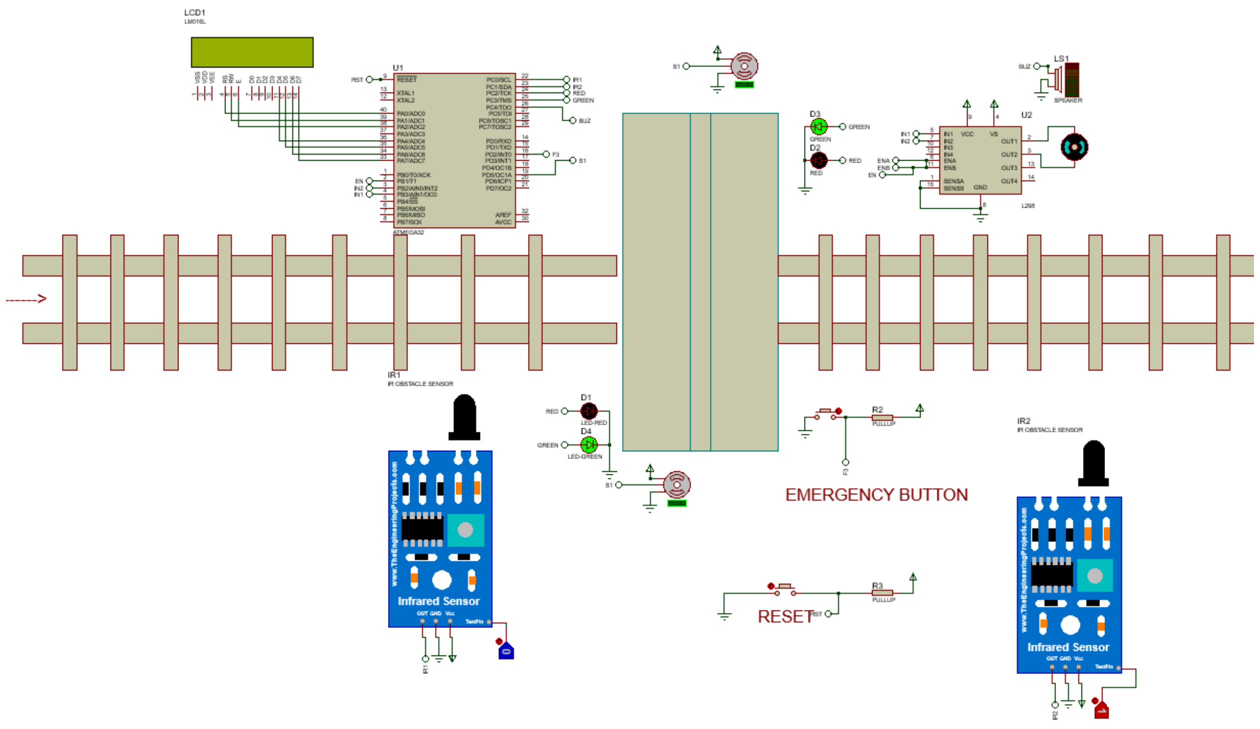


Figure: “Reset ” state of the System