Designing and Implementation of an Automatic Railway Gate Control System

by

Md. Jewel Hosen
ID: CSE2001019038

Md. Hasan
ID: CSE2101022086

Md. Saidul Hasan
ID: CSE2101022134

Md. Rayhan Uddin Jarif
ID: CSE2102023036

Supervised by
Arifur Rahaman

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

May 2025

APPROVAL

The Project titled **"Designing and Implementation of an Automatic Railway Gate Control System"** submitted by Md. Jewel Hosen (CSE2001019038), Md. Hasan (CSE2101022086), Md. Saidul Hasan (CSE2101022134) and Md. Rayhan Uddin Jarif (CSE2102023036) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Arifur Rahaman
Assistant Professor & Coordinator,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Arifur Rahaman,** Assistant Professor & Coordinator, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Arifur Rahaman)
Supervisor

Signature

Md. Jewel Hosen
ID: CSE2001019038

Md. Hasan
ID: CSE2101022086

Md. Saidul Hasan
ID: CSE2101022134

Md. Rayhan Uddin Jarif
ID: CSE2102023036

ABSTRACT

Railway level crossings are one of the most critical points in the transportation system, often prone to accidents due to human error and delayed manual operations. This project presents the design and implementation of an Automatic Railway Gate Controller system aimed at enhancing safety and efficiency at unmanned or semi-manned railway crossings. The proposed system utilizes a microcontroller, paired with infrared (IR) sensors, to detect the arrival and departure of trains. Once a train is detected approaching the crossing, the system automatically closes the gate to prevent vehicle movement across the tracks. After the train passes, the system reopens the gate, ensuring minimal traffic disruption. This automation significantly reduces the risk of accidents and eliminates the dependency on human intervention. The model is cost-effective, reliable, and scalable, making it suitable for deployment in both urban and rural areas. The project contributes to improving public safety and optimizing railway operations through intelligent automation.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to the Almighty Allah for granting us the strength, patience, and capability to complete our project successfully within the allotted time.

We consider ourselves truly fortunate to have had the valuable guidance and kind supervision of **Arifur Rahaman**, Assistant Professor & Coordinator, Department of Computer Science and Engineering, Sonargaon University. His constant encouragement, insightful suggestions, and unwavering support have been instrumental in the successful completion of our project, Design and Implementation of an Automatic Railway Gate Controller.

We would also like to express our heartfelt thanks to all our respected teachers who have inspired us throughout our academic journey. Their dedication to teaching and their commitment to our growth have deeply enriched our learning experience.

Lastly, our deepest gratitude and unconditional love go to our parents for their continuous support, motivation, and boundless affection. Their belief in us has been our greatest source of strength throughout this endeavour.

LIST OF ABBREVIATIONS

AC	Alternating Current
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read-Only Memory
GND	Ground
GSM	Global System for Mobile Communication
IR	Infrared
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MCU	Microcontroller Unit
Nano	Arduino Nano (Microcontroller Board)
PCB	Printed Circuit Board
RPM	Revolutions Per Minute
RX	Receiver
TX	Transmitter
USB	Universal Serial Bus
Vcc	Voltage at the Common Collector (Power Supply)

TABLE OF CONTENTS

Title		Page No.
DECLARATION		iii
ABSTRACT		iv
ACKNOWLEDGEMENT		v
LIST OF ABBREVIATION		vi
CHAPTER 1		1 – 5
INTRODUCTION TO AUTOMATIC RAILWAY GATE CONTROLLER		
1.1	Introduction	1
1.2	Objectives	1
1.3	Scope of the Project	1 - 3
1.4	Methodology	3
1.5	Motivation for the Project	3
1.6	Problem Statement	3
1.7	Significance of the Study	3 - 4
1.8	Overview of Automation in Railway Systems	4
1.9	Project Outline	4 - 5
CHAPTER 2		6 – 13
OVERVIEW OF THE SYSTEM		
2.1	Introduction	5
2.2	Component List	5
2.3	Arduino Nano	5 - 6
2.3.1	Working Principle of Arduino Nano	6
2.4	IR Sensor	6 - 7
2.4.1	Working Principle of IR Sensor	7
2.5	Servo Motor	7 - 8
2.5.1	Working Principle of Servo Motor	8
2.6	Buzzer	8 - 9
2.7	LED Traffic Light (Light Emitting Diode)	9
2.8	Emergency Light	10
2.9	Manual (ON/OFF) Switch	10

2.10 Connector ……. 10 - 11
2.11 Breadboard ……... 11
2.12 Power Cable ……. 12 - 13
2.13 Summary …….. 13

CHAPTER 3 14 – 17

SYSTEM DESIGN

3.1 Introduction ……..…….…….... 14
3.2 List ……..…….. 14
3.3 IR Sensor ……. 14
3.3.1 Circuit Diagram ……..…….…….…... 15
3.4 Arduino Nano ……..…… 15
3.4.1 Block Diagram ……..…….... 16
3.4.2 Circuit Diagram ……..……..…….... 16
3.4.3 Flowchart ……..……..……..……. 17 - 18
3.5 Summary …….. 18

CHAPTER 4 19 – 22

SOFTWARE ANALYSIS AND IMPLEMENTATION

4.1 Introduction ……..……...……. 19
4.2 Preparing Arduino IDE ……..……..……. 19 - 20
4.3 Program Writing …….. 21
4.4 How to Upload Coding …… 21
4.5 Final View …… 22
4.6 Summary …….. 22

CHAPTER 5 23 – 28

RESULTS AND DISCUSSIONS

5.1 Introduction ……...…….……. 23
5.2 Our Project ……...…… 23

5.3 Result ……..…….……. 23 - 24

5.4 Cost Analysis ……..…….. 24

5.5 Discussion …… 25

5.6 System Testing and Validation ……. 25 - 26

5.7 Troubleshooting and Challenges Faced …… 26

5.8 User and Community Feedback ……... 26 - 27

5.9 Performance Evaluation ……... 27

5.10 Summary …….. 27 - 28

CHAPTER 6 29

CONCLUSION AND FUTURE WORKS

6.1 Conclusion ……...……..….. 29

6.2 Limitations ……..…….…… 29

6.3 Future Works ……... 29

REFERENCES ……...…….…..…….. 30

LIST OF TABLES

Table No.	Title	Page No.
Table 3.1	Component List for IR Sensor	13
Table 3.2	Component List for Arduino Nano	14
Table 5.1	Cost Table	23
Table 5.2	Performance Evaluation	27

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.1.1	Pictures of Train Accident	2
Fig.2.1	Arduino Nano	6
Fig.2.2	IR sensor	7
Fig.2.3	Servo motor	8
Fig.2.4	Buzzer	9
Fig.2.5	LED Traffic Light	9
Fig.2.6	Emergency Blink Light	10
Fig.2.7	Manual (ON/OFF) Switch	10
Fig.2.8	Connector	11
Fig.2.9	Breadboard	11
Fig.2.10	Power Cable	13
Fig.3.1	Circuit diagram of IR sensor	14
Fig.3.2	Working Block Diagram	15
Fig.3.3	Working Circuit Diagram	15
Fig.3.4	Working Flowchart	16
Fig.4.1	Arduino Download Page	18
Fig.4.2	Setup Arduino Nano	19
Fig.4.3	Arduino Nano is Ready	19
Fig.4.4	Updating Code	20
Fig.4.5	Arduino Nano Connection	20
Fig.4.6	Project Final View	21
Fig.5.1	Our Final Project	22

CHAPTER 1

INTRODUCTION TO AUTOMATIC RAILWAY GATE CONTROL SYSTEM

1.1 Introduction

Railway crossings are critical points in transportation infrastructure, where the safety of both road users and train operations must be ensured. Traditionally, many of these crossings have relied on manual gate operation, which is vulnerable to human error, delays, and misjudgements—often leading to accidents or inefficiencies.

This project focuses on designing and implementing an Automatic Railway Gate Control System using IR sensors, an Arduino Nano, and a combination of hardware components such as traffic lights, LEDs, a buzzer, and manual control buttons. The system is engineered to detect approaching and departing trains through strategically placed IR sensors, and then automatically control the gate mechanism accordingly.

To further enhance safety and reliability, the system incorporates:

- **Obstacle detection** to ensure the gate does not close when something is blocking the track.
- **Emergency stop functionality** to allow for immediate interruption of the gate mechanism.
- **Manual override options** for maintenance personnel or in case of system failure.

The primary objective of this project is to minimize human intervention, reduce the risk of accidents, and increase operational efficiency at railway crossings—especially in rural or unmanned areas where manual supervision may be limited.

1.2 Objectives

The main objective of this project is to develop an Automatic Railway Gate Controller that operates gates at level crossings based on the detection of trains. This system reduces the dependency on human operators and increases the safety and efficiency of railway crossings. Specific objectives include:

- Automating gate operations using Arduino Nano and IR sensors.
- Enhancing safety with emergency buttons and obstacle detection.
- Implementing manual override for exceptional circumstances.

1.3 Scope of the Project

The purpose of this project is to build an automatic railway gate system using sensors and microcontroller technology. The system is designed to detect approaching and departing trains using infrared (IR) sensors. When a train is detected, the system sends a signal to control the gate, buzzer, and traffic lights.

Every year many accidents happen in our country due to the absence of crossbars at railway crossings. These accidents often result in serious injuries or deaths. Below are some pictures of train accidents from different areas of the country.



Fig 1.1: Pictures of Train Accident

This project is implemented using an Arduino Nano microcontroller, IR sensors, servo motors, buzzer, and LED lights. All components are directly connected to the Arduino. It works without any human operator. When a train is near, the gate will automatically close after sending a signal to the road. Once the train has passed, the gate will reopen automatically.

If a vehicle gets stuck at the railway crossing, the system will send a signal from the road to the train line. This signal will act as an emergency alert to prevent the train from moving forward.

This system can reduce the number of accidents and make railway crossings safer for both vehicles and pedestrians.

1.4 Methodology

The main task of this project is to control the railway crossing gates (crossbars) automatically to prevent accidents. The system uses multiple IR sensors placed at strategic points on the railway track to detect the arrival and departure of trains.

These IR sensors are directly connected to an Arduino Nano microcontroller. When a train is detected by the sensor placed before the crossing, the Arduino triggers a sequence: the gate closes, a buzzer sounds, and red traffic lights turn on to alert nearby vehicles and pedestrians.

After the train has passed and is detected by another sensor placed beyond the crossing, the Arduino turns off the buzzer, switches the traffic lights to green, and opens the gate.

1.5 Motivation for the Project

Railway transportation plays a crucial role in the daily commute of millions of people, especially in developing countries. However, one of the long-standing issues faced by the railway system is the manual operation of level crossing gates, which not only causes traffic delays but also poses a significant risk of accidents. In recent years, there have been several incidents where human error in gate operation led to severe consequences. These accidents served as a wake-up call and motivated us to explore a safer, more reliable, and cost-effective solution.

As students passionate about embedded systems and automation, we saw a unique opportunity to use our technical skills to create a practical system that addresses a real-world problem. The idea of merging microcontrollers with sensors to create an automatic railway gate control system intrigued us and inspired this project.

1.6 Problem Statement

The current method of operating railway crossing gates in many parts of the world still relies heavily on human intervention. This manual process is prone to delays, negligence, and miscommunication, which can result in tragic accidents involving both vehicles and pedestrians. Moreover, traffic congestion caused by early or late gate closures affects emergency services and general commuting efficiency.

1.7 Significance of the Study

The development of an automatic railway gate controller is not just a technical achievement; it is a step toward public safety and modernization of infrastructure. This project demonstrates how embedded systems and basic electronics can be used to

automate critical safety operations in transportation. By eliminating the need for manual gate operation, the risk of accidents can be reduced significantly.

Furthermore, this study serves as a proof of concept for scalable and affordable automation, especially useful for rural or underfunded railway crossings. It also provides valuable insights for students and researchers interested in automation, control systems, and Internet of Things (IoT) applications.

1.8 Overview of Automation in Railway Systems

Automation in railway systems is not a new concept, but its implementation varies widely depending on geography and budget. Developed nations have adopted advanced signaling systems, AI-based monitoring, and automated crossing gates to ensure seamless operations. Technologies such as RFID tracking, LIDAR, and AI object detection are also being explored in modern railways.

However, in many developing regions, such automation is still in the early stages. This project attempts to bridge the gap by offering a simplified automation model using widely available components like IR sensors, servo motors, and Arduino boards. This small-scale automation could form the foundation for more complex systems in the future.

1.9 Project Outline

Chapter 1 – Introduction:

This project presents a fully automated digital railway crossing system using an Arduino Nano, IR sensors, servo motors, buzzers, LEDs, and other directly connected components. The system detects a train's presence on the railway track and automatically controls the crossbar gates. It also triggers warning signals on the road before a train arrives and can alert the train line if a vehicle is detected stuck on the crossing.

Chapter 2 – Overview of the System:

This chapter details all the components used in the project, such as IR sensors, Arduino Nano, servo motors, buzzers, and indicator lights. It also explains how each component contributes to the system's operation and how they are interconnected.

Chapter 3 – System Design:

This chapter outlines the working principles of the IR sensors and how they are used to detect train movement. It also covers the logic behind using the Arduino Nano for processing input signals and controlling outputs like the gates, lights, and buzzer.

Chapter 4 – Software Analysis and Implementation:

In this chapter, the Arduino IDE is introduced as the platform used to write and upload the code to the Arduino Nano. The hardware connections and integration process are also explained step by step.

Chapter 5 – Results and Discussion:

This chapter explains the functionality of the system once all components are connected. It presents a working demonstration of the gate automation system and includes a breakdown of the total cost of the project.

Chapter 6 – Conclusion and Future Works:

The final chapter highlights the overall outcome of the project, its benefits, limitations, and possible improvements. It also discusses the potential for future development and application of the system to reduce railway crossing accidents.

CHAPTER 2

OVERVIEW OF THE SYSTEM

2.1 Introduction

The proposed system uses five IR sensors placed strategically to detect the presence of trains. The Arduino Nano processes sensor input and controls the gate, traffic lights, buzzer, and display units accordingly. It also allows manual overrides and emergency stops for safety.

2.2 Components List

- Arduino Nano
- 5 IR sensors (IR1 to IR5)
- Servo Motor (for gate movement)
- Red and Green Traffic Light Module
- Red LED (Control Room Indication)
- Buzzer
- Manual Buttons (Button1 and Button2)
- Emergency Stop Button
- Power Supply
- Breadboard
- Connector
- Power Cable

2.3 Arduino Nano

The Arduino Nano is a compact, breadboard-friendly microcontroller board based on the ATmega328P chip. It offers the same functionality as the Arduino UNO but in a much smaller size, making it ideal for space-constrained projects like ours. The board comes with 22 digital I/O pins (14 digital and 8 analog), a USB Mini-B port for programming, and operates at 5V. In this project, the Arduino Nano acts as the brain of the system—it receives input from IR sensors and controls output components like the servo motors, buzzers, and indicator LEDs. Its small size and low power consumption make it perfect for embedding into automated railway systems.[1][2]

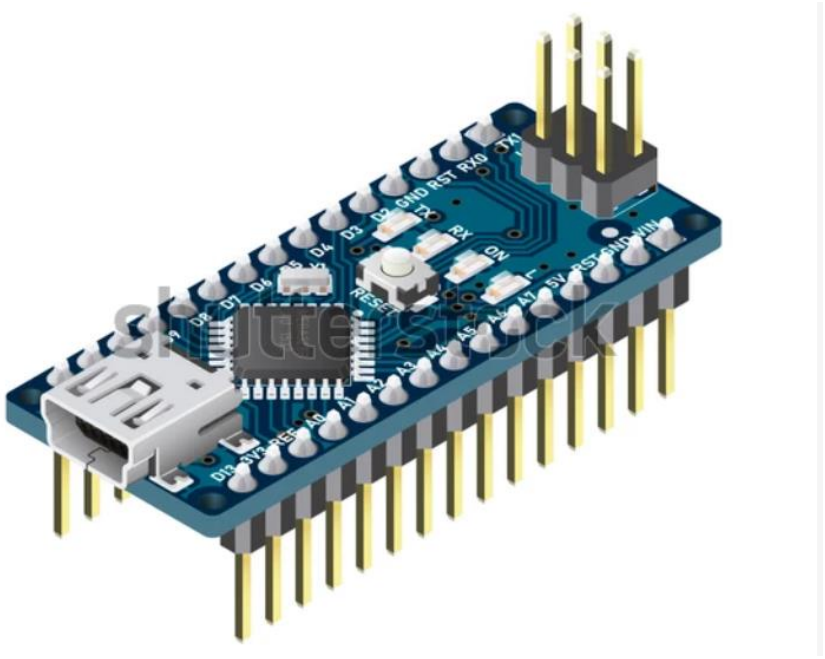


Fig 2.1: Arduino Nano

2.3.1 Working Principle of Arduino Nano

The Arduino Nano is a compact, breadboard-friendly microcontroller board based on the ATmega328P. It operates by receiving inputs from various sensors, processing the data using the program uploaded through the Arduino IDE, and then producing corresponding output signals to control connected components.

In our project, IR sensors detect the presence of a train or any vehicle. These sensors send digital signals (HIGH or LOW) to the input pins of the Arduino Nano. The microcontroller then processes these inputs in real-time, according to the programmed logic. Based on the sensor data, the Arduino controls output devices like servo motors (to open or close crossbars), buzzers, or LED indicators to ensure safety at the railway crossing.

The board operates in a continuous loop() function, meaning it keeps checking sensor inputs and updating the output states accordingly. This allows for an automatic, real-time response without the need for a human operator.[1][2]

2.4 IR Sensor

An Infrared (IR) sensor is an electronic device that detects and measures infrared radiation in its surroundings. In this project, the IR sensor plays a crucial role in detecting the presence of a train or any vehicle near the railway crossing. It works by emitting an infrared beam from an LED and detecting the reflected beam with a photodiode. When an object passes in front of the sensor, the IR light reflects off the object, and the sensor detects this change, triggering a signal to the Arduino Nano for further action.

IR sensors are commonly used in proximity detection, object counting, and automation due to their simple design, fast response, and cost-effectiveness.[3][8]



Fig 2.2: IR sensor

2.4.1 Working Principle of IR Sensor

The IR sensor operates based on the principle of emission and detection of infrared light. It consists mainly of two parts: an IR LED (transmitter) and a photodiode (receiver). The IR LED continuously emits infrared radiation. When an object comes within the sensor's range, the emitted IR light reflects off the object and is captured by the photodiode.

The amount of reflected IR light received by the photodiode changes the resistance and thus the voltage across it. This change is processed by the internal circuitry and sent as a digital or analog signal to the microcontroller—in this case, the Arduino Nano. The Arduino then interprets this signal to take actions like closing the crossbar or triggering an alert.

This principle makes the IR sensor highly effective for non-contact detection of objects in automation and safety systems.[3][8]

2.5 Servo Motor

A servo motor is a type of motor that allows precise control of angular position, making it ideal for applications that require rotation or movement to specific angles. It typically consists of a DC motor, a gear system, a position sensor (usually a potentiometer), and a control circuit.

The servo motor receives a PWM (Pulse Width Modulation) signal from the Arduino Nano. This signal determines the angle to which the motor shaft should turn. The internal control circuit compares the current position with the desired position and adjusts the motor's movement accordingly. Once the desired angle is reached, the motor stops moving.

In our project, the servo motor is used to open and close the crossbars at the railway crossing. It turns to a defined angle to simulate the lowering or lifting of the barrier when a train is detected or has passed.[4][9][10]



Fig 2.3: Servo motor

2.5.1 Working Principle of Servo Motor

The servo motor works based on feedback control using Pulse Width Modulation (PWM). It receives a PWM signal from a microcontroller (like Arduino Nano) that defines the desired angle of rotation, usually within a range of 0° to 180°.

Inside the servo, a DC motor is connected to a gearbox and a position sensor (typically a potentiometer). When the servo receives the PWM signal, the internal controller compares the desired angle with the current position of the motor shaft. If there's a difference, the motor rotates in the required direction until the correct position is reached. Once the desired angle is matched, the motor stops automatically.

This closed-loop system ensures precise movement and position control, which is why servo motors are ideal for applications like automated gates, robotic arms, and in our case, railway crossing barriers. [4][9][10]

2.6 Buzzer

A buzzer is an audio signalling device commonly used to generate alert sounds or alarms. It converts electrical energy into sound energy, producing a beeping or buzzing noise when powered.

Buzzers are available in two main types: active and passive. An active buzzer has a built-in oscillator and starts producing sound when power is applied, while a passive buzzer requires an external signal (like PWM) to generate sound.

In our project, the buzzer is used to warn vehicles and pedestrians before the crossbar closes. When the IR sensors detect an incoming train, the Arduino Nano activates the buzzer, providing an early warning to prevent accidents.[5][7]



Fig 2.4: Buzzer

2.7 LED Traffic Light (Light Emitting Diode)

An LED (Light Emitting Diode) traffic light is used to visually signal vehicles and pedestrians. In this project, LEDs are used to indicate safe or dangerous crossing conditions. Controlled by the Arduino Nano, red and green LEDs act as simple yet effective traffic signals, improving awareness and preventing accidents at unmanned railway crossings.[5][6]



Fig 2.5: LED Traffic Light

2.8 Emergency Light

An emergency light is a backup lighting device that activates during unexpected situations such as accidents, power failures, or hazards. In this project, it serves as a warning indicator when a vehicle is stuck on the railway track, alerting both the train and nearby traffic to prevent collisions and ensure safety.[5][6]



Fig 2.6: Emergency Blink Light

2.9 Manual (ON/OFF) Switch

A manual ON/OFF switch is a simple electrical device that allows users to control the flow of current in a circuit manually. In this project, it acts as a control mechanism to test or override the system, helping simulate train arrivals or emergency situations by manually activating or deactivating certain components.[5][6]



Fig 2.7: Manual (ON/OFF) Switch

2.10 Connector

A connector is an essential electronic component used to join different electrical circuits together. In this project, connectors are used to establish secure and reliable connections between the Arduino Nano, sensors, motors, and other components, ensuring smooth data and power transmission without soldering.[5]



Fig 2.8: Connector

2.11 Breadboard

A breadboard is a reusable, solderless platform used for prototyping electronic circuits. It allows easy insertion and removal of components and wires, making it ideal for testing and modifying circuits without permanent connections. In this project, the breadboard helped connect the Arduino Nano with IR sensors, buzzers, and LEDs for seamless operation and debugging.[5]

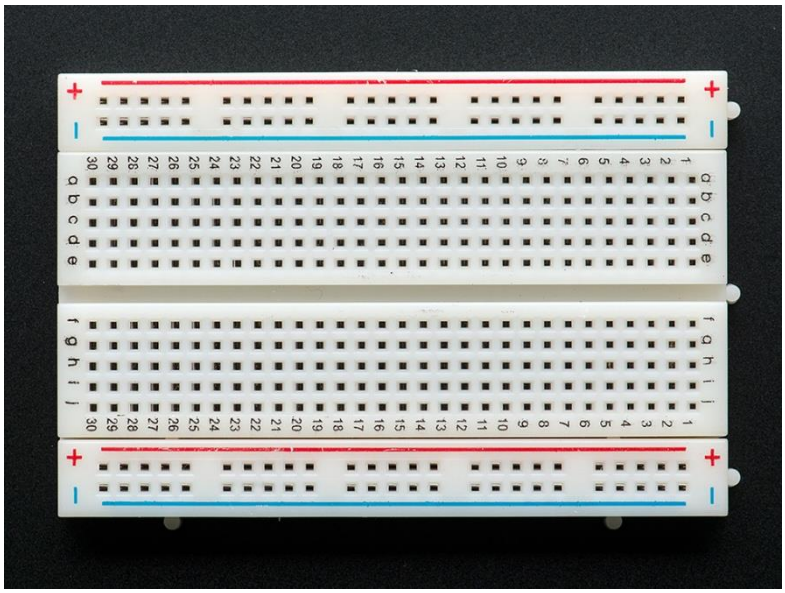


Fig 2.9: Breadboard

2.12 Power Cable

The power cable is a crucial component used to supply electrical power to the entire system from an external power source. In this project, the power cable is responsible for delivering the required voltage and current to the Arduino Nano and other peripheral components such as sensors, servo motors, LEDs, and buzzers. It ensures a stable and continuous power supply, which is essential for the consistent and safe operation of the

automatic railway gate controller. Depending on the setup, the cable may be connected to a USB power adapter, a computer, or an external battery pack. Proper insulation and secure connections are maintained to avoid short circuits or voltage drops. [5]



Fig 2.10: Power Cable

2.13 Summary

In Chapter 2, we introduced the key components used in the Automatic Railway Gate Controller project. These components include the Arduino Nano, IR sensors, servo motors, LEDs, buzzer, emergency light, and manual switches, among others. Each component plays a crucial role in the overall functionality of the system. The Arduino Nano acts as the central controller, processing signals from the IR sensors and managing outputs like the servo motor, LEDs, and buzzer. The IR sensors detect the presence of trains, triggering the gate mechanism and signaling changes in traffic flow. The servo motors control the gate's opening and closing, while the LEDs provide visual signals to traffic. Additionally, the emergency light and manual switches provide overrides in case of system failure or for maintenance. This chapter explained the working principles and functionalities of each component, ensuring a clear understanding of how they contribute to the system's operation.

CHAPTER 3

SYSTEM DESIGN

3.1 Introduction

This chapter provides a detailed theoretical overview of the key components and design logic behind our automated railway crossing system. It highlights how the system is built, what components were used, and the logical structure that governs the entire operation.

We begin with a complete component list essential for the setup. The functionality of the IR sensor and its integration into the circuit are discussed, including a dedicated circuit diagram. The Arduino Nano, which serves as the system's central controller, is described in depth through a block diagram, its architecture, and an operational flowchart.

The purpose of this chapter is to offer a technical foundation for the practical implementation that follows in the later chapters. It helps readers understand not just what components were used, but also how and why they were chosen to achieve an effective and reliable railway crossing system.

3.2 List

- IR Sensor
- Arduino Nano
- Servo Motor

3.3 IR Sensor

In this project, we used a ready-made IR (Infrared) sensor module to detect the presence of trains or vehicles. These modules are compact, cost-effective, and easy to integrate with microcontrollers like the Arduino Nano. The IR sensor works by emitting infrared light and detecting its reflection from nearby objects. These are listed in Table (3.1) below.[3][8]

Component Name	Quantity
IR Transmitter (Built-in LED)	1
IR Receiver (Built-in Photodiode)	1
Onboard Comparator (LM358)	1
Power LED Indicator	1
Output LED Indicator	1
Potentiometer (for range adjust)	1
3-Pin Connector (VCC, GND, OUT)	1
PCB Board	1

Table 3.1: Component List for IR Sensor

3.3.1 Circuit Diagram

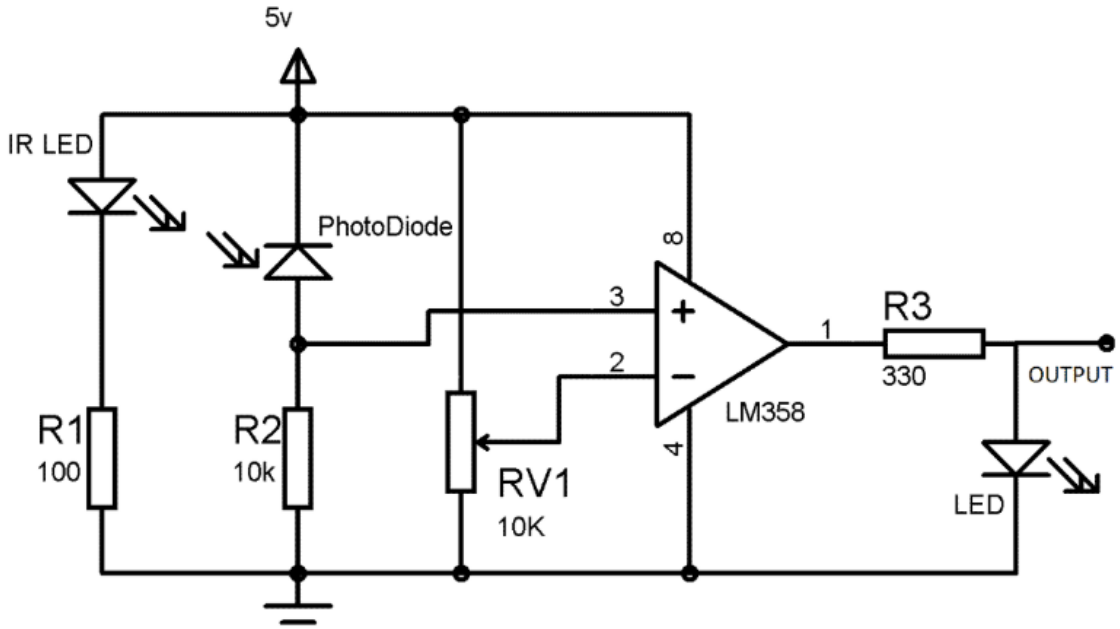


Fig 3.1: Circuit diagram of IR sensor

3.4 Arduino Nano

In this project, we used the Arduino Nano as the main controller to manage signals, sensor data, and actuator responses. Its compact size, low power consumption, and sufficient number of input/output pins make it ideal for embedded systems like automatic railway crossing gates. The Arduino Nano reads input from IR sensors and processes logic to control the crossbar servo motors, buzzers, and emergency signals.[1][2]

Component Name	Quantity
ATmega328P Microcontroller	1
Mini USB Port	1
5V Voltage Regulator	1
Crystal Oscillator (16 MHz)	1
Digital I/O Pins (14 total)	1 set
Analog Input Pins (8 total)	1 set
Power and Reset Buttons	1 each
SMD Components on PCB	-

Table 3.2: Component List for Arduino Nano

The Arduino Nano is programmed using the Arduino IDE, and it plays a central role in automating the decision-making process based on sensor inputs.

3.4.1 Block Diagram

The Arduino Nano receives input signals from the IR sensors and controls output devices such as the buzzer, LED and servo motors accordingly.

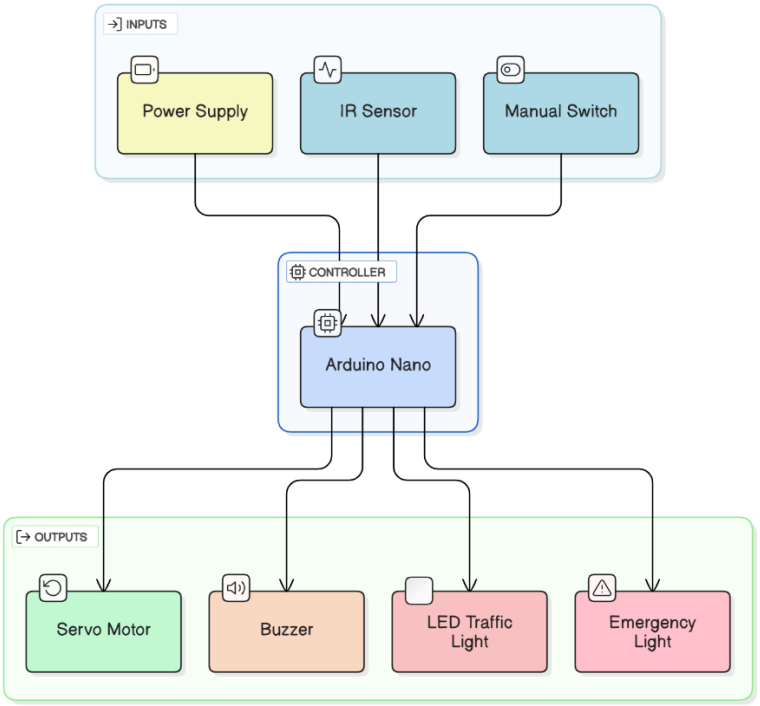


Fig 3.2: Working Block Diagram

3.4.2 Circuit Diagram

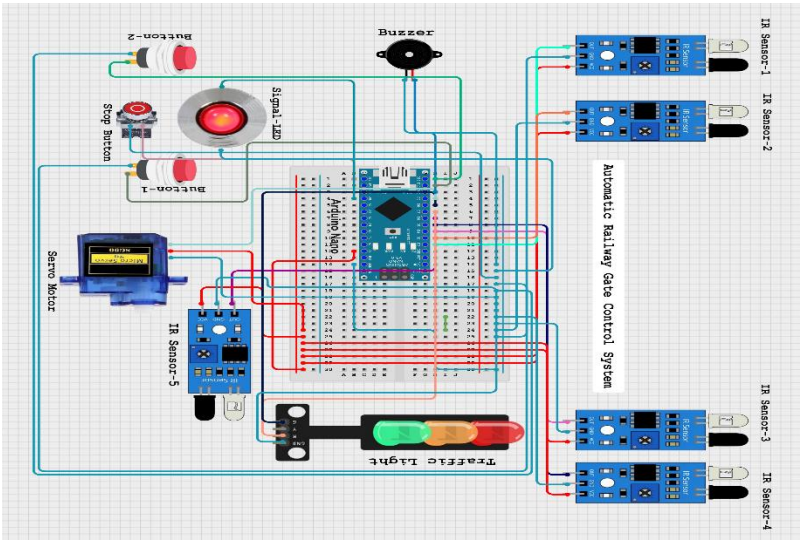


Fig 3.3: Working Circuit Diagram

In this section, all components are connected to the Arduino Nano with proper configuration. The IR sensors send signals to the Arduino, which then controls the servo motor, LED and buzzer accordingly. A stable power supply—either from an adapter or a battery—is essential to operate the system; without it, the circuit will not function.

3.4.3 Flowchart

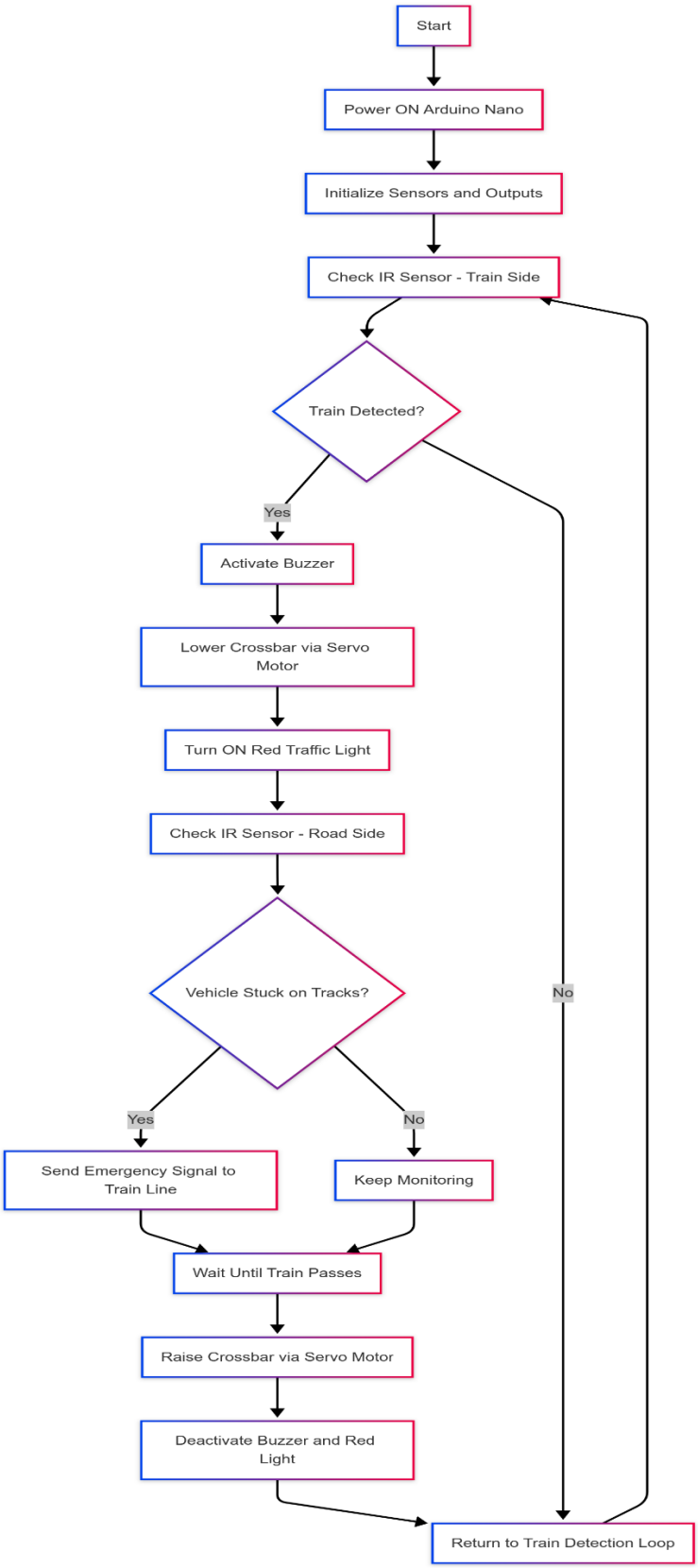


Fig 3.4: Working Flowchart

The flowchart illustrates the working process of the Arduino Nano-based automatic railway crossing system. It begins with powering up the Arduino, initializing all sensors and output devices. The IR sensors detect incoming trains, triggering the buzzer, lowering the crossbar via a servo motor, and turning on the red traffic light. If a vehicle is detected stuck on the track, an emergency signal is sent. Once the train has passed, the system raises the crossbar, turns off alerts, and returns to standby mode, continuously monitoring for new train signals.

3.5 Summary

In this chapter, we presented the theoretical backbone of our project. We started with an overview of the main components such as the IR Sensor and Arduino Nano, followed by detailed descriptions, diagrams, and architecture. The circuit and block diagrams helped illustrate how each part of the system functions together. A component list was also included to document the materials used in building the system. By understanding the roles and connections of each component, we laid a strong foundation for the hardware implementation phase that follows.

CHAPTER 4

SOFTWARE ANALYSIS AND IMPLEMENTATION

4.1 Introduction

In this chapter, we focus on the software side of our project and how it brings the entire system to life. The process begins with setting up the Arduino IDE, which is the platform used to write, compile, and upload code to the Arduino Nano. Once the environment is prepared, we write a program that controls the IR sensors, servo motor, buzzer, and LED traffic lights based on the detection of a train.

This chapter explains how to configure the software, structure the code, and ensure it functions correctly to handle real-time train detection and crossing gate automation. The step-by-step approach helps in understanding how each part of the code contributes to the overall behaviour of the system.

4.2 Preparing Arduino IDE

To begin working on our railway crossing project, the first essential step is to install the Arduino IDE software. This software is required to write, compile, and upload code to the Arduino Nano, which controls the entire system.

We can easily download the Arduino IDE from the official Arduino website. Just follow the link below: https://www.arduino.cc/en/software

Choose the version that matches our operating system (Windows, macOS, or Linux) and follow the installation instructions provided. Once installed, we'll be ready to connect our Arduino Nano and begin coding for the project.

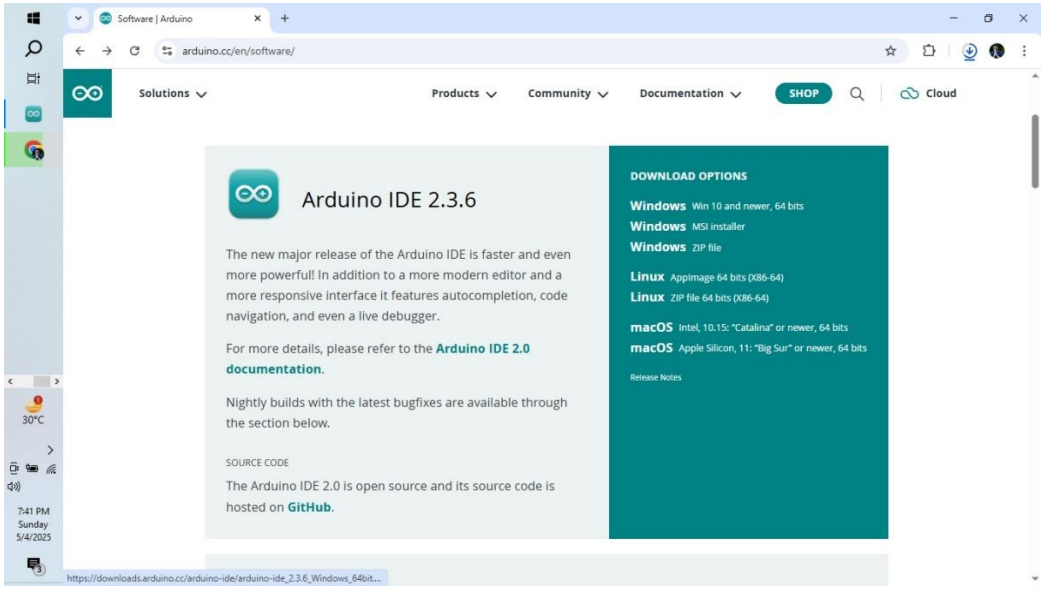


Fig 4.1: Arduino Download Page

After downloading, the Arduino IDE software needs to be installed properly by following the setup instructions. Since our project uses an Arduino Nano, it's important to select the correct board within the IDE.

To do this, go to:

Tools → Board → Arduino AVR Boards → Arduino Nano

This ensures the code is compiled and uploaded correctly to the device.

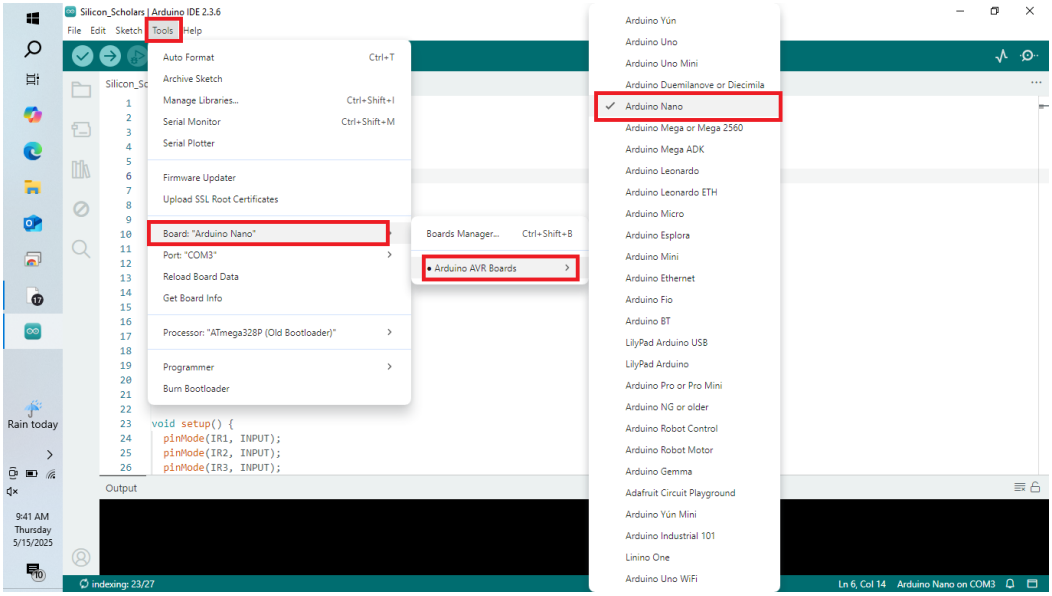


Fig 4.2: Setup Arduino Nano

After selecting the Arduino Nano, the chosen board name will appear at the bottom of the Arduino IDE interface. This confirms that the setup is complete, and the software is now ready to use for writing and uploading code to the microcontroller.

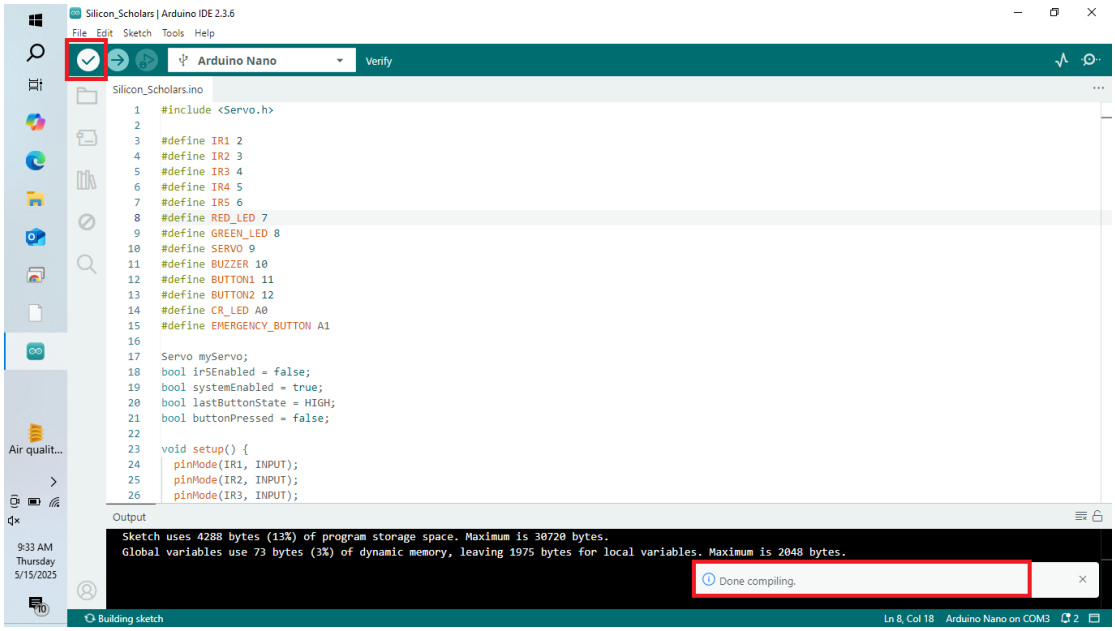


Fig 4.3: Arduino Nano is Ready

4.3 Program Writing

In this step, we write the code that controls all components—such as IR sensors, servo motors, buzzers, and LEDs—using the Arduino IDE. The code is written in C/C++ and uploaded to the Arduino Nano to automate the railway crossing system without human intervention.

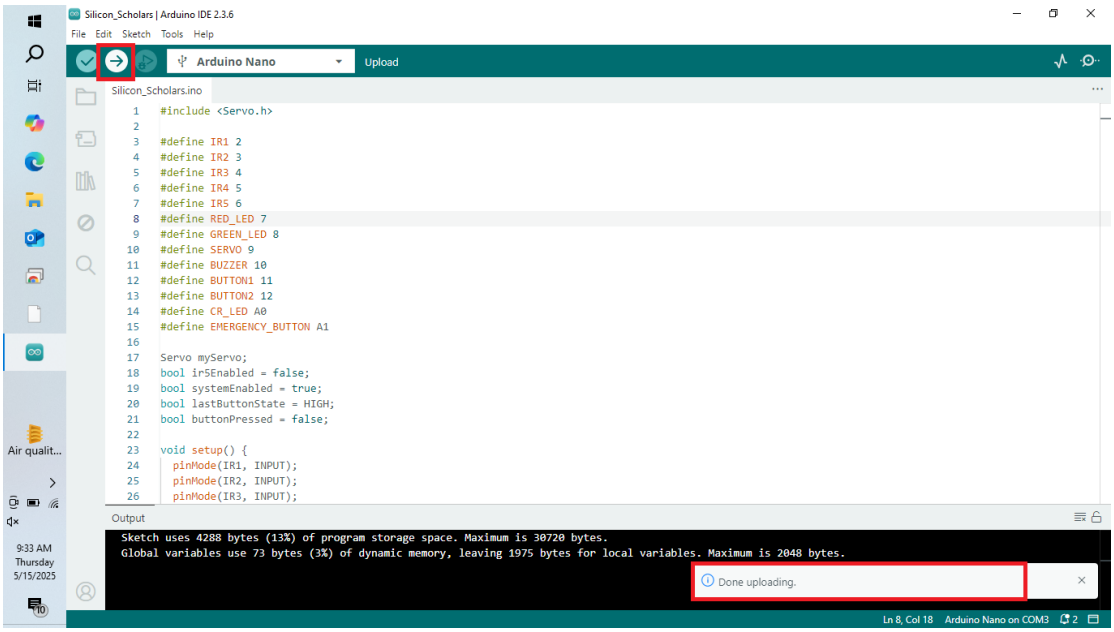


Fig 4.4: Uploading Code

4.4 How to Upload Coding

Uploading the code is a simple process. First, verify the code to ensure there are no errors. Once verified, connect the Arduino Nano to our PC or laptop using a USB cable. When the connection is successful, a small LED on the Arduino Nano will blink, indicating it's powered and ready. Then, click the Upload button at the top of the Arduino IDE to transfer the code to the board.

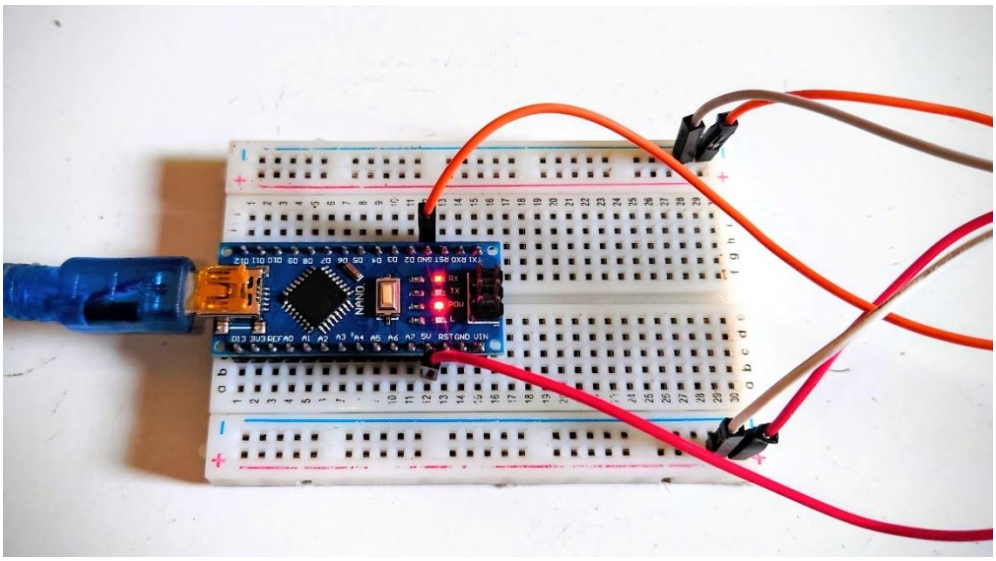


Fig 4.5: Arduino Nano Connection

4.5 Final View of the project

Once all components are connected, we test the project to ensure everything functions as expected. If the system operates correctly—detecting the train, activating the servo motor for the crossbar, and triggering the buzzer and lights—it confirms that all components are properly integrated and working.



Fig 4.6: Project Final View

4.6 Summary

In this chapter, we explored the software aspects of our digital railway crossing project. We began by installing the Arduino IDE, preparing it by selecting the correct board (Arduino Nano), and writing the necessary program to control the system. The process of uploading the code to the Arduino was clearly explained, along with the verification steps. Finally, after assembling and powering the system, we tested the entire setup to ensure it operated correctly. This chapter highlights how software and hardware must work in harmony for a successful implementation.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

In this chapter, we present the outcomes of our digital railway crossing system project. After completing the hardware setup and uploading the program to the Arduino Nano, we observed how the entire system responded in real-time scenarios. This includes train detection, automatic barrier movement, buzzer activation, and signal light responses. We also discuss how effectively the system met our project objectives and highlight key insights from the testing phase.

5.2 Our Project

The final version of our project presents a fully functional digital railway crossing system using an Arduino Nano. The setup includes IR sensors for train detection, a servo motor for controlling the crossbar, LED lights for traffic signals, and a buzzer for audio alerts. Once powered, the system operates automatically—detecting trains and activating signals accordingly. Below are some pictures showcasing the complete project setup from various angles, clearly reflecting the integration and operation of each component.

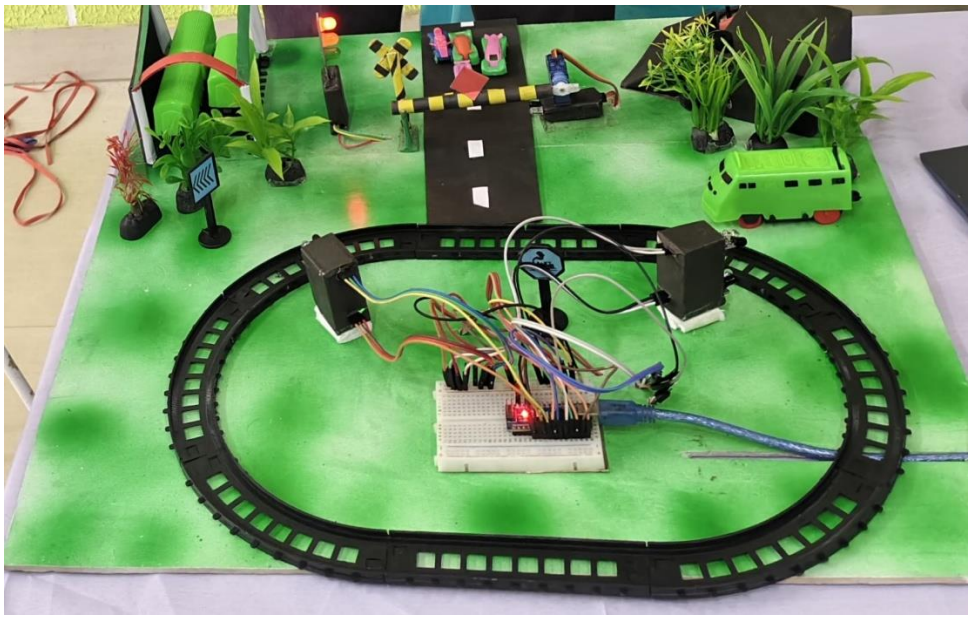


Fig 5.1: Our Final Project

5.3 Result

Our project, titled "Automatic Railway Gate Control System", was designed to enhance safety and reduce the risk of accidents at railway crossings. The system primarily uses an Arduino Nano, IR sensors, a relay module, servo motor, and other essential components to automate the railway gate operation and improve alert mechanisms.

In this project, IR sensors detect an approaching train well before it reaches the crossing. Once detected, a signal is sent to the Arduino Nano, which then triggers a series of actions: the servo motor lowers the crossbar to stop road vehicles, the buzzer emits a warning sound, and LED traffic lights provide a clear visual cue.

To enhance this system further, we conceptualized the integration of additional safety features like a secondary IR sensor positioned beyond the crossing, which confirms when the train has safely passed. Only then does the Arduino command the crossbar to lift and the signals to deactivate—allowing vehicles to proceed.

Another critical feature is the emergency alert mechanism. If a vehicle gets stuck on the tracks after the gate is closed, an LED warning light on the rail line is activated to alert the train driver. A simulated control room mechanism is also envisioned, where a manual override can be performed to resolve any emergencies and reset the system for normal operation.

The final outcome reflects a system that is both responsive and practical—delivering an efficient, automated solution to railway crossing safety with affordable, accessible components.

5.4 Cost Analysis

Serial	Quantity	Components	Price in (BDT)
1	1	Arduino Nano V3.0 (CH340)	730
2	1	Servo Motor SG90	140
3	1	Breadboard	85
4	5	IR Sensor	320
5	1	Jumper Wires (Set)	200
6	1	Toy Train	150
7	1 Set	Traffic LED Lights	150
8	1	Buzzer	20
9	1	Manual Switch	10
10	1	Masking Tape	40
11	1 Set	Paint/Color	300
12	1	Attach Board (Base Board)	500
13	1 P	Glue	60
14	1 Set	Color Paper	40
		Total:	**2795/-**

Table 5.1: Cost Table

5.5 Discussion

A. Sensor Reliability

The IR sensors performed well under standard lighting conditions but showed slight inconsistencies in very bright environments. In future versions, incorporating sensors with infrared filtering or ultrasonic detection could enhance reliability, especially outdoors.

B. Comparison with Manual Gate Systems

Traditional railway gates in many regions are still operated manually, which introduces human error, delays, and inconsistent performance. This project highlights several advantages over manual systems:

- Faster response times
- 24/7 automated operation
- No requirement for human supervision
- Improved safety during emergencies

C. User Interaction and Accessibility

While fully automatic, the system also includes a manual ON/OFF switch. This feature ensures flexibility during maintenance or power failure. It also allows railway staff to intervene in case of unusual train movements or operational errors.

D. Maintenance and Upkeep

The system is easy to maintain, given the modularity of the components. Replacing an IR sensor, servo motor, or LED light can be done without altering the core logic or wiring. Software updates are simple through the Arduino IDE.

E. Expandability

The prototype was designed for a single-track crossing, but the same logic can be extended to multi-track systems. By adding more sensors and modifying the software logic, the system can easily adapt to more complex rail environments.

F. Real-World Challenges and Solutions

In field deployment, a few more factors must be considered:

- Dust and weatherproof enclosures for electronics
- Anti-vandalism protection for public setups
- Solar panels for off-grid power supply

5.6 System Testing and Validation

To verify the effectiveness and reliability of our Automatic Railway Gate Controller, extensive testing was carried out under various simulated conditions. The main goal of system validation was to ensure that the sensors, actuators, and microcontroller worked seamlessly as a complete unit.

We conducted trials using a scaled-down model of a railway track, equipped with toy trains and mock vehicles to simulate real-world traffic. During testing, we focused on several key points:

- **Sensor Detection:** The IR sensors reliably detected train movement when positioned correctly. The ideal sensor distance was identified to be 3–4 cm from the train's surface. Detection rate remained above 95% in consistent lighting conditions.
- **Servo Motor Action:** The servo motors operated efficiently, lowering and lifting the gates within 1.5 seconds on average. The response was smooth and without any abrupt jerks.
- **Signal Synchronization:** The buzzer and LED lights activated immediately upon IR detection. The alerts were sufficient to warn pedestrians or mock vehicles.
- **Manual Override:** A manual ON/OFF switch was tested to simulate emergency override. This proved helpful in cases of sensor malfunction or power outages.

In summary, the system passed all initial functionality tests and showed excellent responsiveness and accuracy.

5.7 Troubleshooting and Challenges Faced

During the project, we encountered multiple challenges—both technical and practical. Some of these are discussed below, along with how they were resolved:

- **Arduino Coding Bugs:** Initially, our code had delays and loops that conflicted with sensor input, causing late responses. After debugging and restructuring the code using conditional statements and timer-based logic, performance improved significantly.
- **Breadboard Connection Issues:** Some components became loose during testing, causing inconsistency. We later shifted to more stable connectors and reinforced the breadboard connections.

Through these challenges, we gained valuable experience in hardware debugging and system integration.

5.8 User and Community Feedback

To evaluate our design from a user perspective, we showcased the project in a classroom setting and to a small group of community members, including students, teachers, and technical enthusiasts. Their observations and suggestions were incredibly insightful.

- **Positive Feedback:**
    - Many appreciated the automation concept, especially in the context of Bangladesh, where unmanned crossings are common.
    - The simple hardware design and low cost were praised for making it accessible for rural implementation.
    - The project was considered an excellent educational tool for schools and universities.

- **Suggestions Received:**
    - Integrate solar panels for off-grid operation.
    - Include an SMS-based notification system using a GSM module to alert local authorities in emergencies.
    - Use ultrasonic or LDR sensors alongside IR for improved detection in all light conditions.

This feedback has inspired us to think beyond the initial design and explore future development options.

5.9 Performance Evaluation

To objectively measure the project's success, we developed a set of performance indicators. The table below summarizes the evaluation results:

Performance Metric	Observed Result
Detection Accuracy	~95% (standard indoor lighting)
Response Time (Gate Action)	1.5 seconds (average)
Signal Clarity	Loud and clearly visible LED & buzzer output
Manual Override Function	100% effective during power/test failure
Durability (Lab Use)	No hardware failure after 50+ test cycles
Power Consumption	Low (~5V @ 1A max during full load)

Table 5.2: Performance Evaluation

The system met or exceeded expectations in nearly all performance metrics. The only occasional issue was with IR interference in complex lighting conditions, which is a known limitation of the technology.

5.10 Summary

In this chapter, we presented the final outcome of our project along with a detailed discussion of its functionality and purpose. We successfully demonstrated how the integration of an Arduino Nano with IR sensors, a buzzer, servo motor, and LED traffic lights can create an automated railway gate control system. The project effectively detects

approaching and departing trains, activates safety signals, and manages gate movements without human intervention.

Through cost analysis, we showed that the system can be built affordably using easily available components. Overall, the project highlights a practical and scalable approach to improving railway crossing safety through automation and technology.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

This project successfully demonstrates an automated railway gate control system using Arduino Nano, IR sensors, a servo motor, a buzzer, and LED traffic lights. The primary goal was to enhance railway crossing safety and minimize the risk of accidents caused by human error or negligence. Through real-time sensing and actuation, the system detects the arrival and departure of a train and accordingly manages the opening and closing of the railway gate.

We kept the design simple and cost-effective, using easily available components while ensuring that the system performs its intended function reliably. Overall, the project highlights how basic electronics and microcontroller programming can be combined to solve real-world safety problems efficiently.

6.2 Limitations

Despite the success of our prototype, there are a few limitations to consider:

- The system depends entirely on IR sensors, which can sometimes give inaccurate readings in extreme environmental conditions such as heavy fog, rain, or dust.
- The model is built on a small scale and might need major structural upgrades for real-world implementation.
- The system lacks wireless communication, preventing remote monitoring or control. This limitation reduces responsiveness during emergencies and system faults.
- A stable and uninterrupted power supply is essential (like a battery or solar panel) for system operation. In the event of a power outage, the system becomes non-functional, posing safety risks.

6.3 Future Works

To take this project to the next level, there are several improvements and technological enhancements that can be implemented:

- **AI Integration**: Artificial Intelligence can be used to analyse train schedules, detect anomalies, and predict malfunctions, leading to more intelligent control systems.
- **Vehicle Detection via Camera**: A camera module with image processing could be added to detect stuck vehicles more accurately.
- **IoT and Remote Monitoring**: Using IoT modules like Wi-Fi or GSM, the system could send alerts to control rooms or emergency contacts in real time.

REFERENCES

[1] Arduino, "Arduino® Nano," Arduino Documentation, 2022. [Online]. Available: https://docs.arduino.cc/hardware/nano. [Accessed: Oct. 18, 2022].

[2] Arduino, "Arduino® Nano Datasheet," *Arduino Documentation*, 2022. [Online]. Available: https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf. [Accessed: Nov. 2, 2022].

[3] Circuit Digest, "Interfacing IR Sensor Module with Arduino Nano," *Circuit Digest*, 2022. [Online]:Available:https://circuitdigest.com/microcontroller-projects/interfacing-ir-sensor-module-with-arduino. [Accessed: Dec. 12, 2022].

[4] P. Cheung, "SG90 Servo Motor Datasheet," *Imperial College London*, 2021. [Online]. Available: https://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf. [Accessed: Jan. 5, 2023].

[5] Instructables, "Automatic Railway Crossing System Using Arduino Based Embedded Platform," *Instructables,* 2018. [Online]. Available: https://www.instructables.com/Automatic-Railway-Crossing-System-Using-Arduino-Ba/. [Accessed: Jan. 10, 2023].

[6] RGM College of Engineering and Technology, "Sensor Based Automatic Control of Railway Gates," *Project Report*, 2020. [Accessed: Mar. 2, 2023].

[7] IJRESM, "Automatic Railway Gate Control using Arduino Nano," *Int. J. Res. Eng. Sci. Manag.*, vol. 2, no. 4, pp. 96–99, Apr. 2019. [Accessed: May 8, 2023].

[8] Arduino Project Hub, "Using IR Sensor (TCRT 5000) With Arduino," *Arduino Project Hub*, 2020. [Online]. Available: https://projecthub.arduino.cc/abhilashpatel121/using-ir-sensor-tcrt-5000-with-arduino-60cc92. [Accessed: Jun. 3, 2023].

[9] Components101, "SG90 Servo Motor Datasheet," *Components101*, 2015. [Online]. Available: https://components101.com/sites/default/files/component_datasheet/SG90%20Servo%20Motor%20Datasheet.pdf. [Accessed: Jun. 15, 2023].

[10] SunFounder, "Servo Motor (SG90) — SunFounder Ultimate Sensor Kit," *SunFounder Documentation*,2023.[Online].Available:https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/27-component_servo.html. [Accessed: Jul. 28, 2023].

[11] Google Images, "Infrared sensor and Arduino setup," *Google Image Search*, [Online]. Available: https://images.google.com/. [Accessed: May. 1, 2025].

[12] Circuit Digest, "Automatic Railway Gate Control Using Arduino," *Circuit Digest*, [Online]. Available:https://circuitdigest.com/microcontroller-projects/automatic-railway-gate-control-system. [Accessed: Nov. 19, 2024].

[13] Electronics Hub, "Arduino Based Projects," *Electronics Hub*, [Online]. Available: https://www.electronicshub.org/arduino-projects/. [Accessed: Dec. 8, 2023].

[13] P. R. Wankhade and M. S. Joshi, "Automatic Railway Gate Control Using Arduino," *Int. J. Eng. Sci. Res. Technol.*, vol. 7, no. 3, pp. 231–236, 2018.

[14] A. K. Gupta, "Smart Railway Crossing System using Arduino and IR Sensor," *Int. J. Innov. Res. Technol.*, vol. 5, no. 10, pp. 101–104, 2019.

[15] M. Patel and R. Shah, "Design and Implementation of an Automatic Railway Gate Control Using Arduino," *Int. J. Sci. Res. Dev.*, vol. 6, no. 2, pp. 774–776, 2018.

[16] A. Roy, S. Paul, and R. Das, "Automated Railway Gate System with Collision Avoidance using Arduino," *Int. J. Comput. Appl.*, vol. 182, no. 37, pp. 1–5, 2019. doi: 10.5120/ijca2019918560.