# CHAPTER ONE

# INTRODUCTION

## 1.1 General

In almost all trains and metros a driver is needed to control and move that train to the desired location, and in recent years this driver was replaced by using micro-controlled machines to do the same work with higher efficiency. And these machines are now implemented in the field of automation widely.

## 1.2 Problem Statement

There are multiple of problems in the current train and carriage transfer systems. Some of these problems are:

- High expenditure for staff (staff swallows a significant part of the costs of running a transport system) and limit of working Time.
- Normal trains cannot be shortened and run frequently.
- Service frequency cannot be easily adjusted to meet sudden unexpected demands.
- Human's detection ability is ineffective and cannot be depended upon.
- Long turnover time preventing fast response and resuming ability.

## 1.3 Research Objectives

The main objectives of this research are to:

- Design of control circuit for simple automatic railway train.
- Simulate of automatic railway control circuit.
- Implement and test of proposed control circuit.

## 1.4 Methodology

- The system is modeled using Arduino mega 2560
- The system simulation has been carried out by using Arduino C as programming language.
- Proteus software is used as a simulator.
- Arduino open-source (IDE) software is used as a compiler.

## 1.5 Project Layout

This project consists of five Chapters including Chapter One. Chapter Two consists of an introduction to the project, automatic control and a historical background. Chapter Three concerns with system hardware components and the simulation of the system. Chapter Four concerns with the practical side of the project that deals with the implementation of the system. Chapter Five presents a brief conclusion and recommendation for future studies.

# CHAPTER TWO

# GENRAL OVERVIEW

## 2.1 Introduction

In a modern control system, electronic intelligence controls some physical process. Control systems are the “automatic” in such things as automatic pilot and automatic washer. Because the machine itself is making the routine decisions, the human operator is freed to do other things. In many cases, machine intelligence is better than direct human control because it can react faster or slower (keep track of long-term slow changes), respond more precisely, and maintain an accurate log of the system’s performance [1].

In This project we will be designing to demonstrate the technology used in Metro/Train traffic which is used in most of the developed countries in its simplest form. This simple train is equipped with a controller that enables the automatic movement from one station to another.

This proposed system is an autonomous and it eliminates the need of any driver. Thus, any human error is ruled out. Whenever this train arrives at the station it stops automatically. Then after prescribed time the door opens so that the passengers may go inside. The door closes after another prescribed time then the train will return to its original set spot.

## 2.2 Automatic Control

A control system is a collection of components working together under the direction of some machine intelligence. Automatic control is the application of control theory for regulation of processes without direct human intervention.

In a digital control system, the controller uses a digital circuit. In most cases, this circuit is actually a computer, usually microprocessor- or microcontroller-based. The computer executes a program that repeats over-and-over (each repetition is called an iteration or scan). The program instructs the computer to read the set point and sensor data and then use these numbers to calculate the controller output (which is sent to the actuator). The program then loops back to the beginning and starts over again. The total time for one pass through the program may be less than 1 millisecond. The digital system only "looks" at the inputs at a certain time in the scan and gives the updated output later. If an input changes just after the computer looked at it, that change will remain undetected until the next time through the scan. This is fundamentally different than the analog system, which is continuous and responds immediately to any changes. However, for most digital control systems, the scan time is so short compared with the response time of the process being controlled that, for all practical purposes, the controller response is instantaneous [1].

## 2.3 A Brief history of automatic control

Automatic control, particularly the application of feedback, has been fundamental to the development of automation. Its origins lie in the level control, water clocks, and pneumatics/hydraulics of the ancient world. From the 17th century onwards, systems were designed for temperature control, the mechanical control of mills, and the regulation of steam engines. During the 19th century it became increasingly clear that feedback systems were prone to instability. A stability criterion was derived independently towards the end of the century by Routh in England and Hurwitz in Switzerland. The 19th century, too, saw the development of servomechanisms, first for ship steering and later for stabilization and autopilots. The invention of aircraft added (literally) a new dimension to the problem. Minorsky's theoretical analysis of ship control in the 1920s clarified the nature of three-term control, also being used for process applications by the 1930s. Based on servo and communications engineering developments of the 1930s, and driven by the need for high-performance gun control systems, the coherent body of theory known as classical control emerged during and just after WWII in the US, UK and elsewhere, as did cybernetics ideas. Meanwhile, an alternative approach to dynamic modelling had been developed in the USSR based on the approaches of Poincaré and Lyapunov [2].

## 2.4 Arduino Controller

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for Internet of Things applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide. Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50. Also The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. As well as, the Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. In addition, the Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, we can add AVR-C code directly into your Arduino programs if you want to. Also the plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

## 2.5 Railway System

Rail transportation, in the widest sense of the expression, is becoming more and more significant both economically and ecologically. The growing number of megacities needs mass transportation systems both within their ever expanding city limits, but also between the urban centers. In addition, more isolated areas need cost efficient, safe, and reliable transportation systems for people and goods into the urban centers. Politicians all over the world are beginning to realize the importance of a well-developed and stable mass transport infrastructure as an alternative to the ever expanding and accident prone individual transportation on congested roads, as is witnessed by the numerous strategic programs for railway and metro systems [3].

### 2.4.1 History of evolution in railway

The history of integrating the traction system (the machine-ensemble of rolling stock and permanent way) with the apparatus of control. A high degree of Integration was found on steam railways in the 19th century using both electrical and mechanical equipment, but the most important feature of the system was the caliber of the personnel who worked it. In the 19th century, nations like Britain, Germany, France and the USA drew on the skills of a plentiful workforce which was low paid and not effectively organized to demand better conditions until well into the 20th century [4].

### 2.4.2 Railway electrification system

A railway electrification system supplies electrical energy to railway locomotives and multiple units so that they can operate without having an on-board prime mover. There are several different electrification systems in use throughout the world. Railway electrification has many advantages but requires significant capital expenditure for installation. The main advantage of electric traction is a higher power-to-weight ratio than forms of traction such as diesel or steam that generate power on board. Electricity enables faster acceleration and higher tractive effort on steep gradients. On locomotives equipped with regenerative brakes, descending gradients require very little use of air brakes as the locomotive's traction motors become generators sending current back into the supply system and/or on-board resistors, which convert the excess energy to heat. Other advantages include the lack of exhaust fumes at point of use, less noise and lower maintenance requirements of the traction units. Given sufficient traffic density, electric trains produce fewer carbon emissions than diesel trains, especially in countries where electricity comes primarily from non-fossil sources. A fully electrified railway has no need to switch between methods of traction thereby making operations more efficient. Two countries that approach this ideal are Switzerland and Hong Kong, but both use more than one system, so unless multi-system locomotives or other rolling stock is used, a switch of traction method may still be required. The main disadvantages are the capital cost of the electrification equipment, most significantly for long distance lines which do not generate heavy traffic. Suburban railways with closely-spaced stations and high traffic density are the most likely to be electrified and main lines carrying heavy and frequent traffic are also electrified in many countries. Also, if the overhead wiring breaks down in some way, all trains can be brought to a standstill [3].

### 2.4.3 Railway intelligent transportation systems

The main purpose of railway transportation which is supported by information flow is to realize safety, efficient, convenient transportation of passengers and freight, on condition that mobile and fixed infrastructures are defined. Whether or not modern railway transportation, which is broad in scale, wide in area, comprehensive in business and social needs, could implement the optimum utilization of resources, as well as safety, efficiency and convenience in transportation process, depends on unimpeded and sharing degree of information flow. Consequently, modern railway transportation system is in fact a freight and passenger flow transportation system which is based on the foundation of information flow. Currently, society brought forward a claim “higher, faster and farther” to railway transportation, which brought unprecedented chances for railway transportation. Plentiful achievements in related area of modern science and technology including IT, communication technology, sensing technology, intelligent control and decision-making technique were integrated organically, which provided the possibility for establishing railway transportation system of the new era. This railway transportation system which possessed epitomizing characteristics was the Railway Intelligent Transportation Systems (RITS) which will bring revolutionary changes to the entire railway transportation. RITS, vehicle and foundational infrastructure constitutes three technical equipment systems of railway transportation. The most basic condition on improving transportation efficiency is to raise the technical equipment's levels. Consequently, strength and perfection of RITS need to be focused upon for technological innovation and rapidly implement action of new vehicles to improve technical merit of infrastructure including lines, power supply and communication signals, to realize high efficiency, safety and high-quality service for new railway transportation patterns [5].

# CHAPTER THREE

# SOFTWERE AND HARDWERE

## 3.1 System Description

In this study the train control system shown in Figure 3.1 is equipped with a controller that enables the automatic running of the train from one station to another.

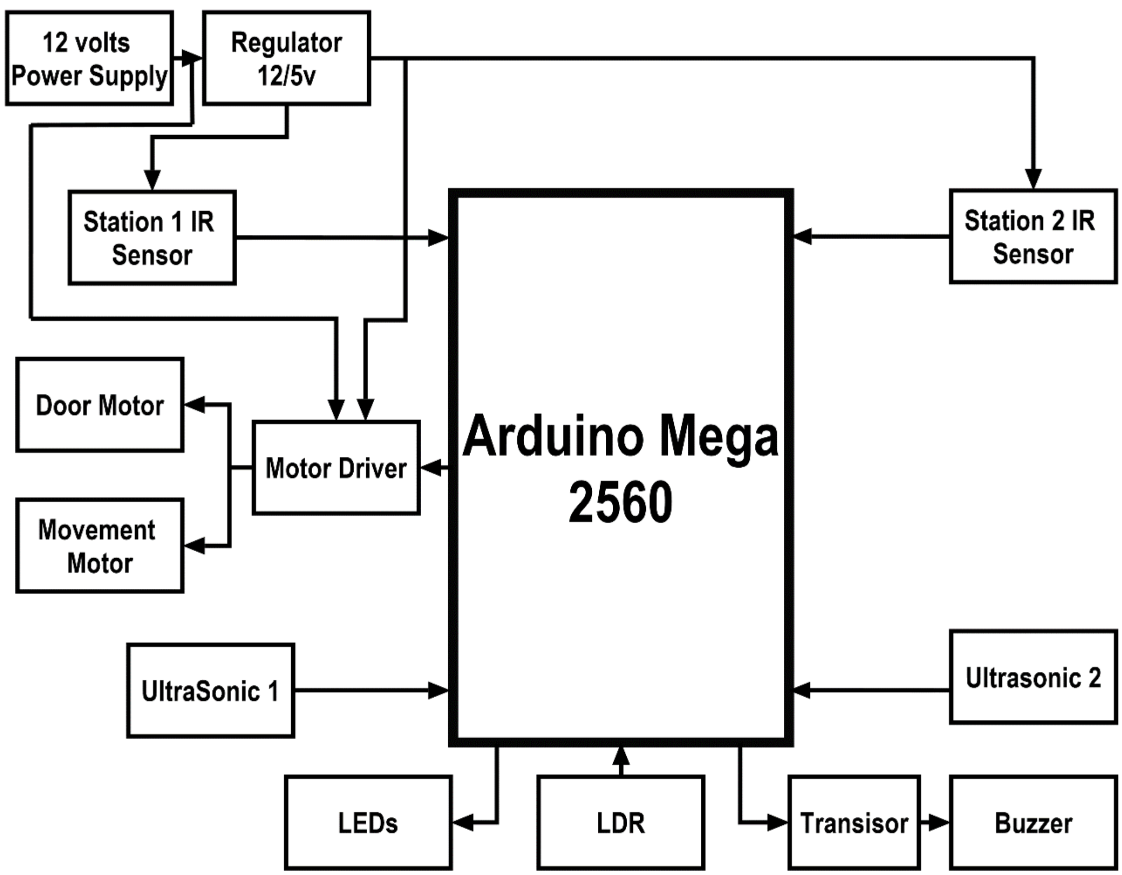


Figure 3.1: Automatic control of a railway system

This proposed system is an autonomous train that eliminates the need of any driver. Thus, any human error is ruled out. In this project Arduino has been used as CPU and Arduino C as its programming language. Whenever the train arrives at the station it stops automatically, as sensed by an IR sensor, then the door opens automatically so that the passengers can go inside. The door then closes after a specified number of passengers set by the programmer. The door closes when the metro reaches its maximum capacity. The movement of the train is controlled by a motor driver L293d. The train incorporates a buzzer to alert the passengers before the door open/close or in case any obstacle appears on the way. As the train reaches the destination the process repeats thus achieving the desired for safety purpose the system has 2 ultrasonic to avoid any type of accidents. Each one is working for one direction forward and backward. In case any obstacle appears on the way the ultrasonic will send signal to the Arduino to stop the system, after the obstacle has been removed the metro will continue to its preset destination.

## 3.2 System Hardware

The hardware used in this study, consists of:

- Main control (Arduino).
- Light sensors.
- Ultrasonic sensor.
- Infrared sensor.
- Motor Driver.

### 3.2.1 Arduino Mega 2560

The controller used in this study is Arduino Mega 2560. It is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator. Figure 3.2 shows Arduino Mega 2560.

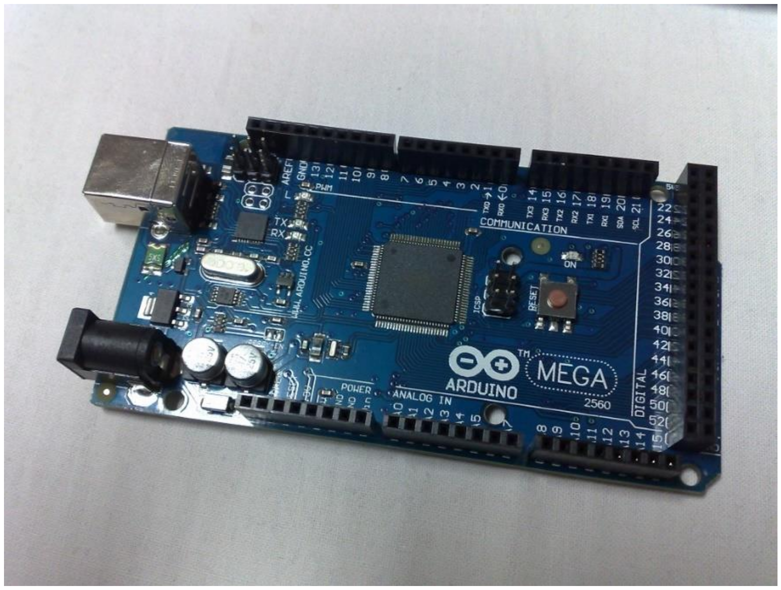

Figure 3.2: Arduino Mega 2560

A USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega 2560 is an update to the Arduino Mega, which it replaces. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on the Processing project,

Which includes support for the **C** and C++ programming languages. The Mega 2560 is designed for more complex projects.

## 3.2.2 Ultrasonic

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400cm or 1 feet to 13 feet. As shown in Figure 3.3

Figure 3.3: HC-SR04 ultrasonic

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

The ultrasonic transmitter emits a short burst of sound in a particular direction. The pulse bounces off a target and returns to the receiver after a time interval. The receiver records the length of this time interval, and calculates the distance travelled based on the speed of sound **c** as showing in Figure 3.4

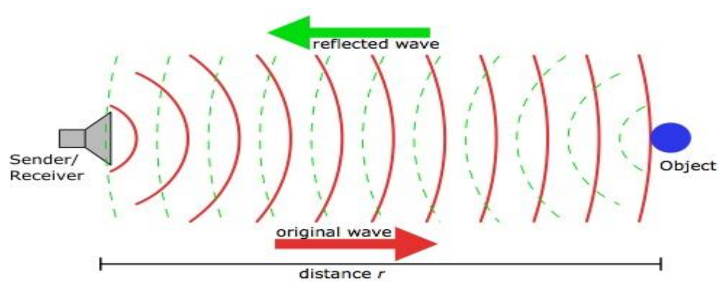


Figure 3.4: Operation of Ultrasonic

### 3.2.3 IR sensor module

Sensors are very important part of electronics, especially in Robotics and automations. There are many kinds of sensors like fire sensor, humidity sensor, motion sensor, temperature sensor and IR sensor which is shown in Figure 3.5

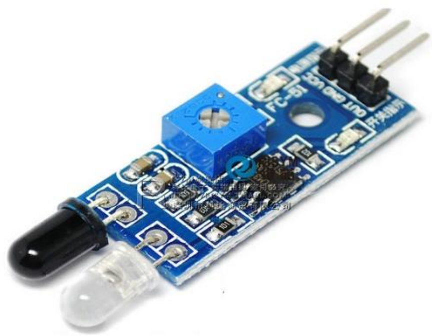

Figure 3.5: IR Sensor module

IR sensor basically consist an IR LED and a Photodiode, this pair is generally called IR pair or Photo coupler. Photodiode resistance changes according to the amount of IR radiation falling on it, hence the voltage drop across it also changes and by using the voltage comparator (like LM358) we can sense the voltage change and generate the output accordingly. The placing of IR LED and Photodiode which can be done in two ways: Direct and indirect as shown in Figure 3.6

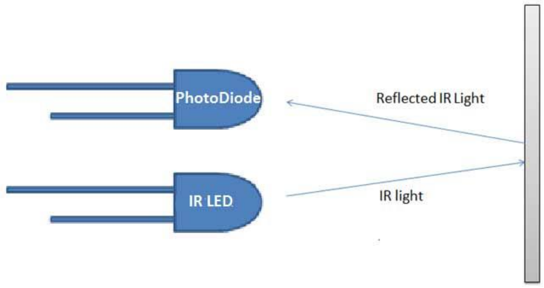


Figure 3.6: IR indirect mode

### 3.2.4 L239D Motor driver IC

L239D as shown in Figure 3.7 is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L239D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motors with a single L239D IC.

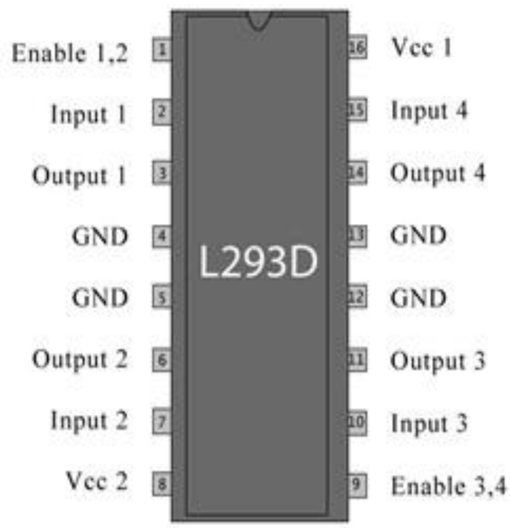


Figure 3.7: L239D Motor driver IC pins

The l293D can drive small and quite big motors as well. It works on the concept of H-bridge. In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors.

### 3.2.5 LDR

Photo resistors are light sensitive whose resistance decrease as the intensity of light they are exposed to increase. Photo resistors, also known as light dependent resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity. The LDR is shown in Figure 3.8

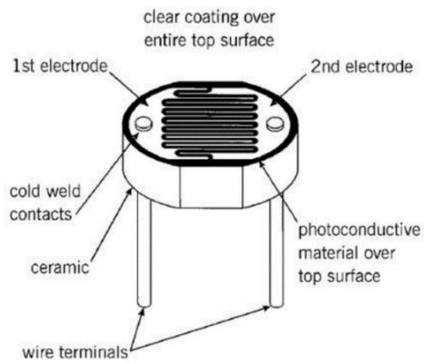


Figure 3.8: LDR Internal structure

In the dark, their resistance is very high, sometimes up to 1MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity.

### 3.2.6 5v co2 laser red pointer

The 5v co2 Laser Red Pointer shown in Figure 3.9 used in many applications like: bending machine, riveting machine button sewing machine and other mechanical equipment positioning lights.

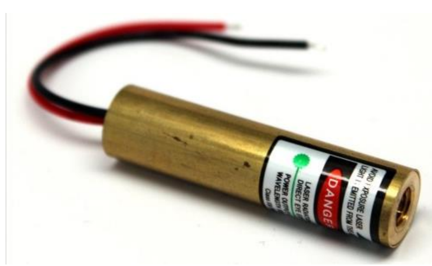

Figure 3.9: 5v co2 Laser Red Pointer

This laser is driven by a DC power supply low-power semiconductor laser emits a laser beam through an optical system convergence, making it well-collimated laser can be fine-tuned in a horizontal position and vertical directions, to play the role of quasi-directed.

### 3.2.7 Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. The buzzer is shown in Figure 3.10 below.

Figure 3.10: Buzzer

Buzzer has 2 pins its positive is connected to the supply and negative pin is connected to the Arduino. Buzzer will sound when it receives 0 from the Arduino and stop when it receives 1.

## 3.3 System Software

The algorithmic flow chart of the proposed system presents a brief explanation of the system. Figure 3.11 shows the flow chart of the proposed system.

### 3.3.1 System Code

The Arduino code was written in C language and compiled with the Arduino application. The code is shown in the appendix.

## 3.3.2 System Simulation

The simulation methodology chosen is Proteus software simulation. The Proteus schematic capture module lies at the heart of the system. It combines the design environment with the ability to define most aspects of the drawing appearance. Proteus software simulation is shown in Figure 3.12.

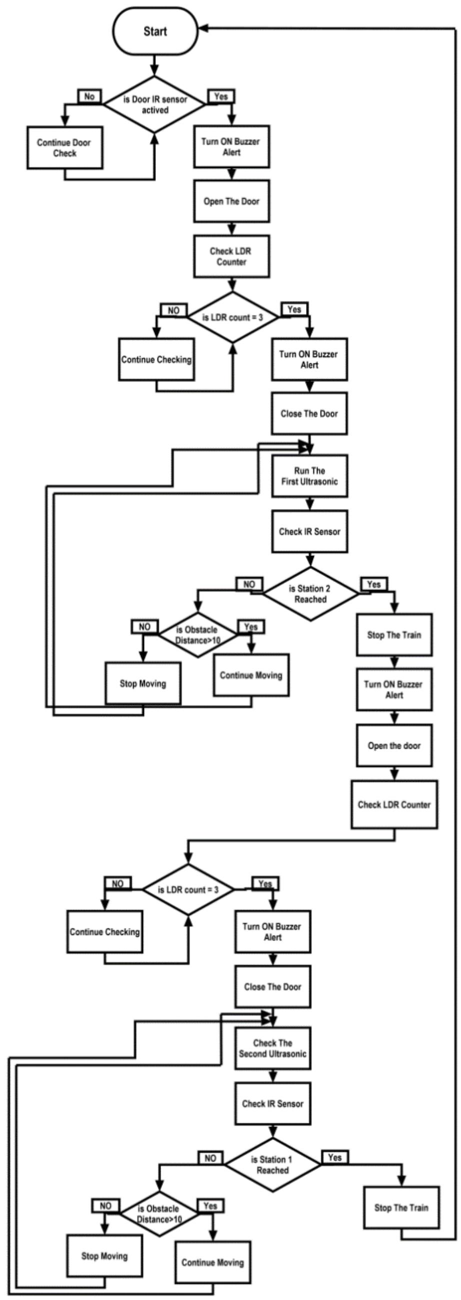

Figure 3.11: flow chart of proposed system

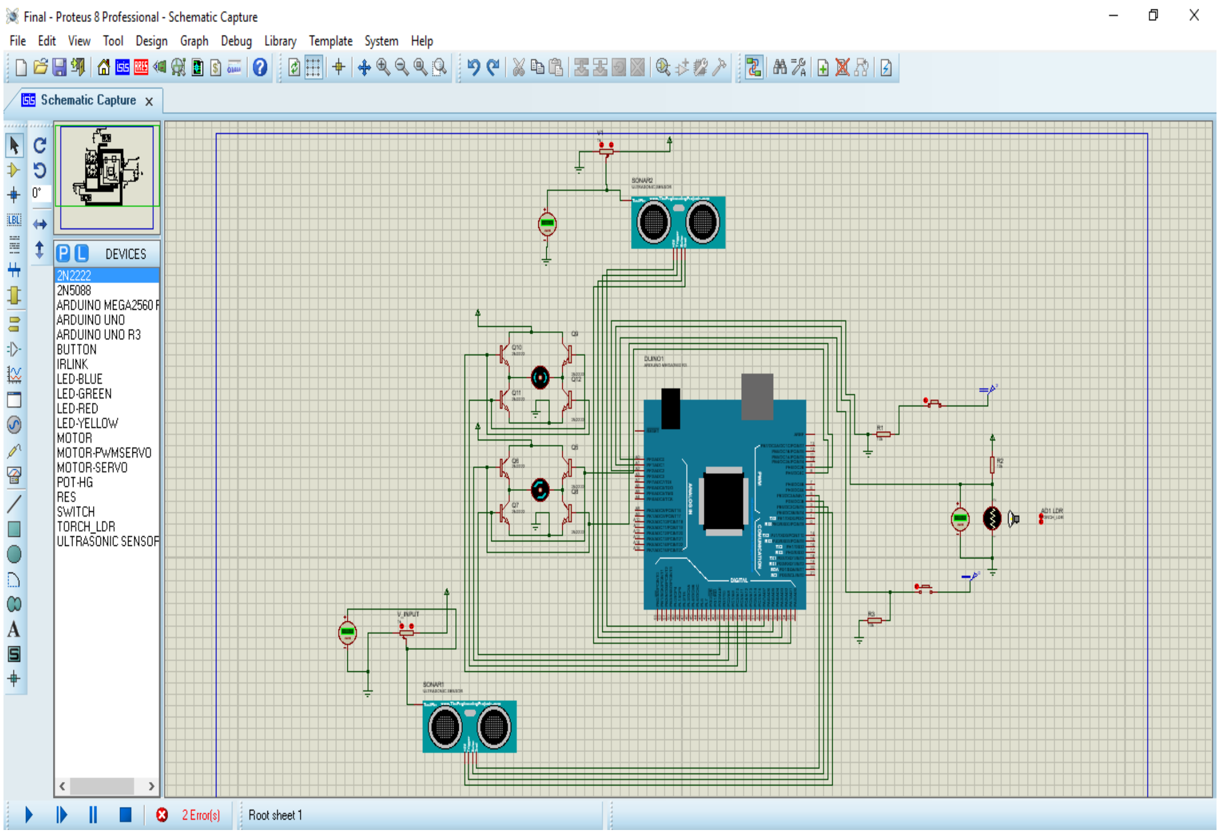


Figure 3.12: System Simulation

# CHAPTER FOUR

# SYSTEM IMPLEMENTATION

The main components required are: Arduino controller, DC motor, IR sensors, ultrasonic sensor, and a buzzer. The motors are controlled using L293D (for bi-directional movement for the vehicle and the door). Motor's shaft is connected to move the vehicle, hence as soon as the motor receives power, the motor on the shaft starts moving and hence results in the movement of the vehicle also. Since the motor driver is connected to the Arduino, which controls the whole system and also contains the instructions for the desired control, it tells the shaft when to start and when to stop. Arduino Mega 2560 is the central component which controls all the activities like reading data from serial port, writing and reading data to/from EEPROM. In order to do all the activities a program (sequence of instructions) must be written for the Arduino. This program is called firmware. In order to execute the program, Arduino requires basic configuration like 5V regulated power supply, clock, and reset circuit. It provides easy user interface. It needs to initialize before is starts processing data. This initialization is done by the Arduino. Buzzer is used for audio indication for ongoing activities. Microcontrollers Arduino and IC's require 5V regulated power supply, which is obtained from 230v AC to 5v DC by using step down transformer. Receiver is based on 125 kHz trans receiver module. L293D motor driver is used to drive the two motors. M1 is for driving the train, M2 is to open and close the door. Arduino mega 2560 has been joined with the breadboard as shown in Figure 4.1.

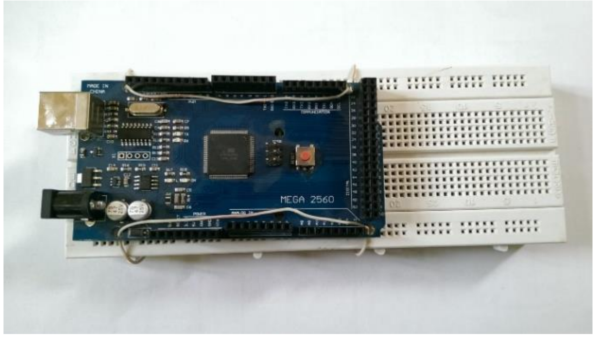

Figure 4.1: The Arduino joined with the breadboard

The driver circuit has been plugged in the breadboard as shown in Figure 4.2

Figure 4.2: Plugging in the motor driver to the breadboard

The voltage regulator is then plugged to the board to step down the voltage from the external power supply as shown in Figure 4.3.

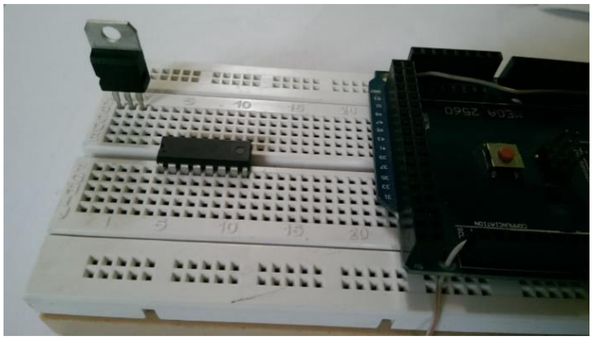

Figure 4.3: Plugging in the regulator to the breadboard

Figure 4.4 shows the connections between the motor driver and the voltage regulator to the Arduino.

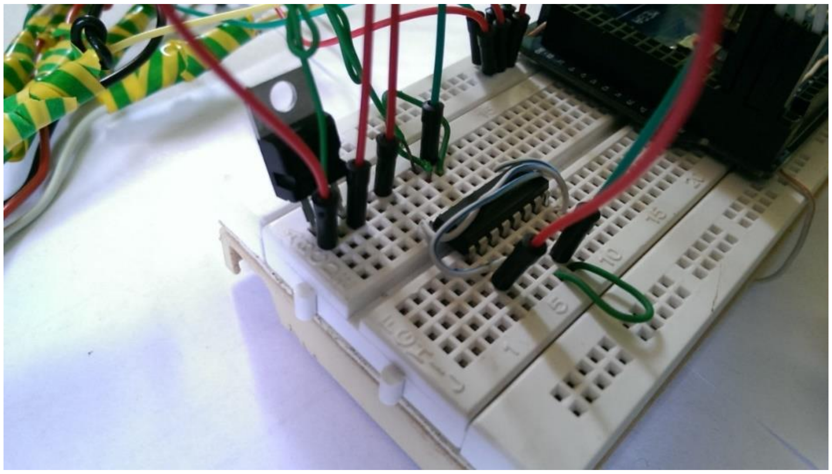

Figure 4.4: connections between the motor driver and the voltage regulator

For safety purposes the system has 2 ultrasonic sensors connected to the Arduino. In case any obstacle appears on the way the ultrasonic will stop the system, after the obstacle has been moved the metro will continue to its preset destination. Ultrasonic is connected to the Arduino as shown in Figure 4.5.

Ultrasonic 1 (forward movement)

- Vcc attached to pin 29
- Trig attached to pin 27
- Echo attached to pin 25
- GND attached to pin 23

Ultrasonic 2 (backward movement)

- Vcc attached to pin 2
- Trig attached to pin 3
- Echo attached to pin 4
- GND attached to pin 5

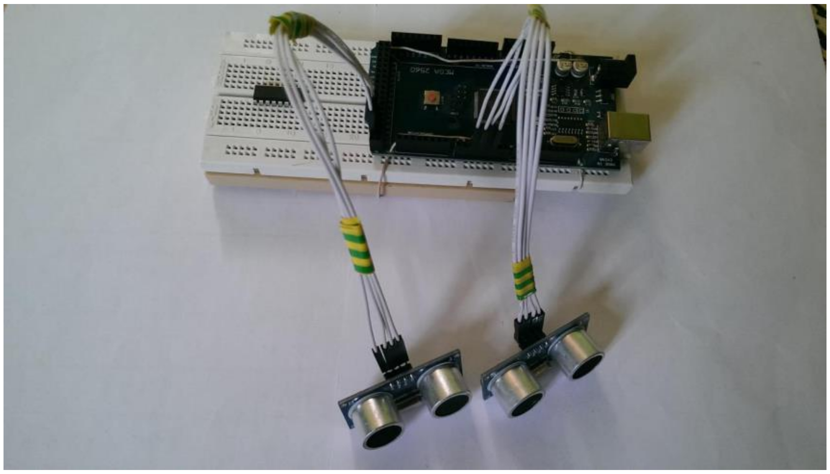

Figure 4.5: the two ultrasonic Connected to the Arduino Mega

The system contains two IR sensors (the door sensor and the destination sensor) connections to the Arduino is shown in Figure 4.6.

IR sensor (door sensor)

- Vcc attached to 5v pin
- GND attached to the GND pin
- OUT attached to A2 pin

IR sensor (destination sensor)

- Vcc attached to 5v pin
- GND attached to the GND pin
- OUT attach to A0 pin

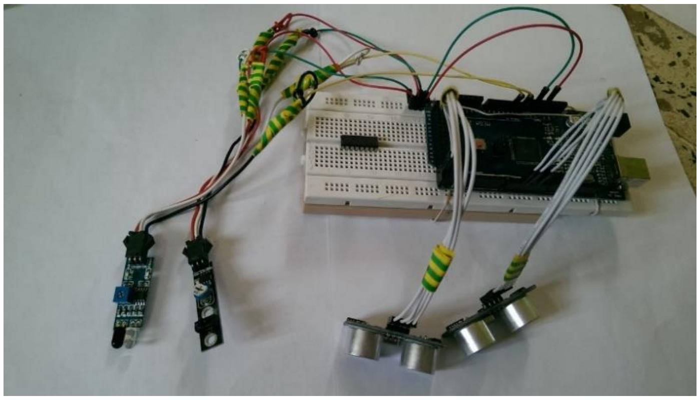

Figure 4.6: IR sensors connected to the Arduino

The red laser pointer is part of the counting section of the system. The laser is fabricated to the body as shown in Figure 4.7.

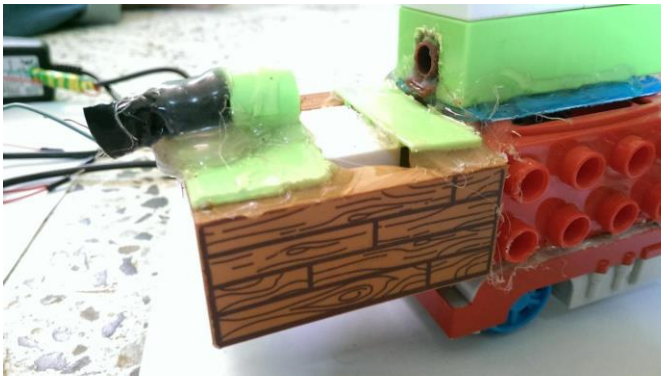

Figure 4.7: Red laser pointer fabricated to the body

For the train door to open and close, a dc motor is incorporated with a wheel to convey the movement of the door as shown in Figure 4.8 below.

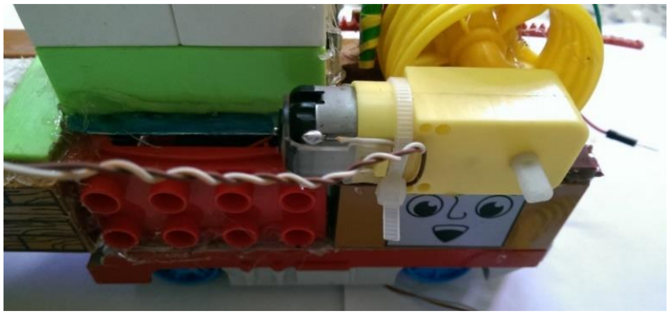

Figure 4.8: Installation of dc motor

Wires are connected between the door motor and Arduino board for interfacing purpose as shown in Figure 4.9.

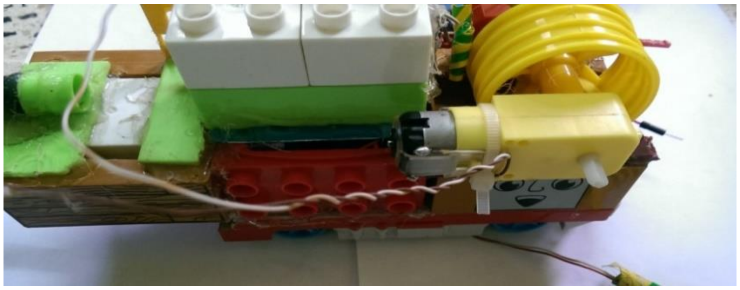

Figure 4.9: Wires connection of dc motor

The buzzer is plugged in the board and connected to the Arduino through a transistor as shown in Figure 4.10.

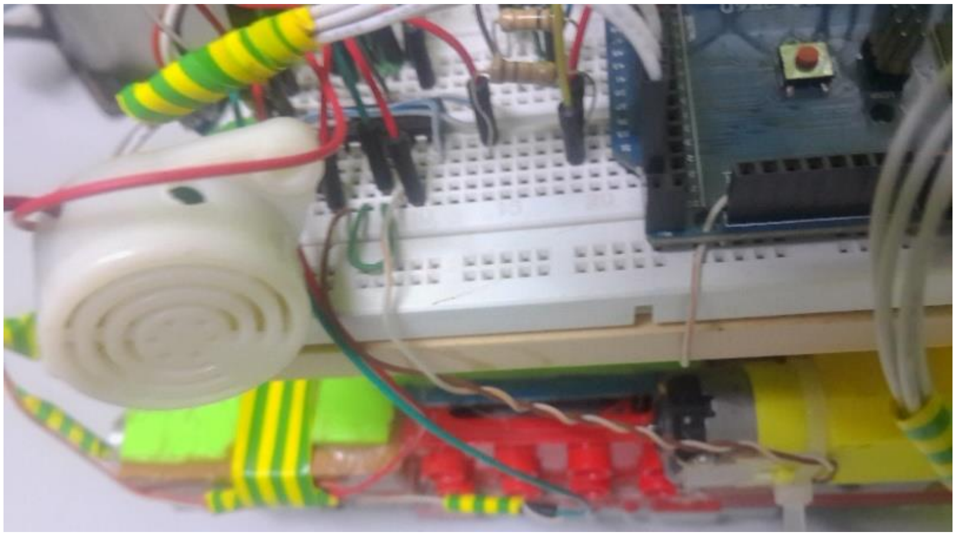

Figure 4.10: Buzzer plugged in the board and connected the Arduino

The assembled hardware system is illustrated in Figure 4.11 below

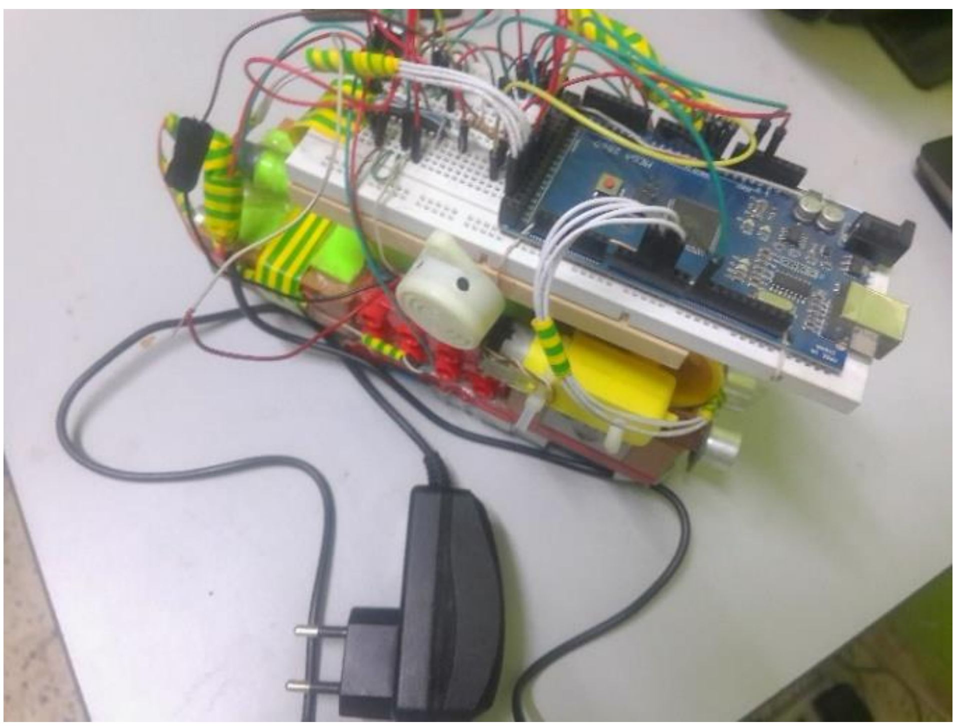

Figure 4.11: The assembled hardware system

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

In this project the main control circuit for the system is designed, simulated, implemented and tested successfully. The sensors are working as they are required to, the mechanical part is working as it should and the system has been tested multiple times and has been found successful. This project's usefulness comes from is its very low cost that can be helpful in operating the metro train system without driver. One can operate the train from remote distance. It will also be beneficial for the developing countries decreasing the cost of labors. Also the station information and emergency will also be displayed to the passengers. It will go to increase the technology trends.

## 5.2 Recommendations

To improve the performance of this prototype it is recommended that you:

- Implement use of intelligent programming.
- Implement use of monitoring system.
- Increase the safety conditions and efficiency.
- Include multiple stations instead of two.

# Appendix

```
int DoorPin_1=33;
int DoorPin_2=35;
int MovmentPin_1=37;
int MovmentPin_2=39;
int CounterCheck = 0;
int CounterCheck2 = 0;
int Destination = A0;
int LDR = A1;
int IRDoor = A2;
int Current_Stage = 0;
int LDR_Value = 0;
int DestCheck=0;
long duration, inches, cm;
int vcc = 29; //attach pin 2 to vcc
int trig = 27; // attach pin 3 to Trig
int echo = 25; //attach pin 4 to Echo
int gnd = 23; //attach pin 5 to GND
long duration2, inches2, cm2;
int vcc2 = 2; //attach pin 2 to vcc
int trig2 = 3; // attach pin 3 to Trig
int echo2 = 4; //attach pin 4 to Echo
int gnd2 = 5; //attach pin 5 to GND
int Buzzer=8;
int currentState=0;
int previousState=0;
```

```
int getout=0;
long microsecondsToInches(long microseconds)
{  return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{  return microseconds / 29 / 2;
}
long microsecondsToInches2(long microseconds2)
{  return microseconds2 / 74 / 2;
}
long microsecondsToCentimeters2(long microseconds2)
{  return microseconds2 / 29 / 2;
}
void UltraSonic()
{ digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm,");
  Serial.println();
delay(100);
```

```
}
void UltraSonic2()
{ digitalWrite(trig2, LOW);
  delayMicroseconds(2);
  digitalWrite(trig2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig2, LOW);
  duration2 = pulseIn(echo2, HIGH);
  inches2 = microsecondsToInches2(duration2);
  cm2 = microsecondsToCentimeters2(duration2);
  Serial.print(cm2);
  Serial.print("cm,");
  Serial.println();
delay(100);
}
void setup() {
  pinMode(MovmentPin_1, OUTPUT);
  pinMode(MovmentPin_2, OUTPUT);
  pinMode(DoorPin_1, OUTPUT);
  pinMode(DoorPin_2, OUTPUT);
  pinMode(A0, INPUT);   //Destination
  pinMode(A1, INPUT);   //LDR
  pinMode(A2, INPUT);   //IRDoor
  pinMode (vcc, OUTPUT);
  pinMode (gnd, OUTPUT);
  pinMode (vcc2, OUTPUT);
```

```
  pinMode (gnd2, OUTPUT);
  pinMode(trig, OUTPUT);
  pinMode(trig2, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(echo2, INPUT);
  pinMode(Buzzer,OUTPUT);
  digitalWrite(vcc, HIGH);
  digitalWrite(gnd, LOW);
  digitalWrite(vcc2, HIGH);
  digitalWrite(gnd2, LOW);
  Serial.begin(9600);
}
void loop() {
Current_Stage=0;
First_State:
   if ((analogRead(IRDoor) <100)&&(Current_Stage==0))
  {
Serial.println(IRDoor);
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
      delay(300);
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
      delay(300);
```

```
    digitalWrite(Buzzer,HIGH);
    delay(300);
    digitalWrite(Buzzer,LOW);
    delay(300);
    digitalWrite(DoorPin_1,HIGH);
  digitalWrite(DoorPin_2,LOW);
  delay(200);
  digitalWrite(DoorPin_1,LOW);
  digitalWrite(DoorPin_2,LOW);
LoopCount:
  if((analogRead(LDR) > 700)&&(Current_Stage==0))
  { delay(250);
    currentState=1;
  }  else{ currentState=0;
}
if((currentState!=previousState)&&(Current_Stage==0))
{ if(currentState==1)
{  CounterCheck=CounterCheck+1;
}}
previousState=currentState;
delay(250);
if(CounterCheck<3)
goto LoopCount;
else{  digitalWrite(Buzzer,HIGH);
    delay(300);
    digitalWrite(Buzzer,LOW);
```

```
    delay(300);
    digitalWrite(Buzzer,HIGH);
    delay(300);
    digitalWrite(Buzzer,LOW);
    delay(300);
    digitalWrite(Buzzer,HIGH);
    delay(300);
    digitalWrite(Buzzer,LOW);
    delay(300);
  digitalWrite(DoorPin_1,LOW);
  digitalWrite(DoorPin_2,HIGH);
     delay(200);
  digitalWrite(DoorPin_1,LOW);
  digitalWrite(DoorPin_2,LOW);
digitalWrite(MovmentPin_2,HIGH);
digitalWrite(MovmentPin_1,LOW);
delay(500);
      Case1:
     UltraSonic();
     UltraSonic();
     UltraSonic();
     UltraSonic();
if((cm>10)&&(analogRead(Destination)>200)&&(Current_Stage==0)) {
digitalWrite(MovmentPin_2,HIGH);
digitalWrite(MovmentPin_1,LOW);
}
```

```
  else{ digitalWrite(MovmentPin_2,LOW);
        digitalWrite(MovmentPin_1,LOW);
        digitalWrite(Buzzer,HIGH);
       delay(150);
       digitalWrite(Buzzer,LOW);
       delay(150);
  }
if((analogRead(Destination)<90)&&(Current_Stage==0)){
  digitalWrite(MovmentPin_2,LOW);
  digitalWrite(MovmentPin_1,LOW);
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
      delay(300);
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
      delay(300);
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
      delay(300);
  digitalWrite(DoorPin_1,HIGH);
  digitalWrite(DoorPin_2,LOW);
  delay(200);
  digitalWrite(DoorPin_1,LOW);
```

```
  digitalWrite(DoorPin_2,LOW);
CounterCheck=0;
Current_Stage=1;
LoopCount2:
  if(analogRead(LDR) > 700)
  {    currentState=1;
  }    else{   currentState=0;
}
if((currentState!=previousState))
{  if(currentState==1)
{    CounterCheck=CounterCheck+1;
 }
}
previousState=currentState;
delay(250);

if(CounterCheck<3)
goto LoopCount2;
else {  Current_Stage=1;
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
      delay(300);
      digitalWrite(Buzzer,HIGH);
      delay(300);
      digitalWrite(Buzzer,LOW);
```

```
    delay(300);
    digitalWrite(Buzzer,HIGH);
    delay(300);
    digitalWrite(Buzzer,LOW);
    delay(300);
  digitalWrite(DoorPin_1,LOW);
  digitalWrite(DoorPin_2,HIGH);
  delay(200);
  digitalWrite(DoorPin_1,LOW);
  digitalWrite(DoorPin_2,LOW);
  digitalWrite(MovmentPin_2,LOW);
  digitalWrite(MovmentPin_1,HIGH);
  delay(500);
Case4:
  UltraSonic2();
  UltraSonic2();
  UltraSonic2();
  UltraSonic2();
  if((cm2>10)&&(analogRead(Destination)>200)&&(Current_Stage==1)) {
  digitalWrite(MovmentPin_2,LOW);
  digitalWrite(MovmentPin_1,HIGH);
  }
  else{
  digitalWrite(MovmentPin_2,LOW);
  digitalWrite(MovmentPin_1,LOW);
    digitalWrite(Buzzer,HIGH);
```

```
      delay(150);
      digitalWrite(Buzzer,LOW);
      delay(150);
  }
if((analogRead(Destination)<90)&&(Current_Stage==1)){
  digitalWrite(MovmentPin_2,LOW);
  digitalWrite(MovmentPin_1,LOW);
  goto Case5;
}else{ goto Case4;
}}}
else{  delay(250);
goto Case1;
}
Case2:
Case5:
CounterCheck=0;
currentState=0;
previousState=0;
}}
else{
delay(250);
goto First_State;
}
delay(5000);
}
```

# References

[1] Christopher T. Kilian, “Modern Control Technology: Components and Systems”, Delmar Cengage, 2005.

[2] Christopher Charles Bissell, “handbook of automation”, Shimon Y, 2009.

[3] Sheilah Frey, “Railway Electrification Systems”, White Word Publications, Delhi, 2012.

[4] Michael C. Duffy, “Electric railways 1880–1990”, Institution of Engineering and Technology, Stevenage, 2003.

[5] Wang Zhuo, Jia Li-min, “The theory and methods of design and optimization for railway intelligent and transportation system”, Beijing, 2011.