

Automatic Railway Tracking and Switching

By

Soumik Dutta Gupta (2140309)

N. Kirutheeka (2140317) Anson Thomas (2140319)

Under the guidance of

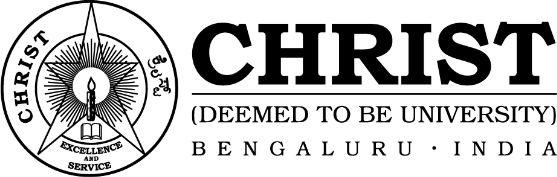
Prof. Benny Sebastian

Department of Physics and Electronics

A project report submitted in partial fulfillment for the award of degree of B.Sc. (Electronics) of CHRIST (Deemed to be University), Bengaluru.

[www.christuniversity.in](http://www.christuniversity.in/)

**April 2024**



C

This is to certify that, the project titled **“Automatic Railway Tracking and Switching”** is a bonafide record of the work done by **“Soumik Dutta Gupta, N. Kirutheeka, Anson Thomas”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Science (Electronics) of CHRIST (Deemed to be University), Bengaluru during the year 2023 - 24.

Head (Electronics) Project Guide

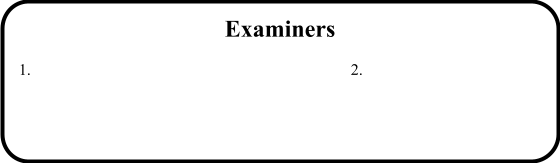
(Prof. Benny Sebastian) (Prof. Benny Sebastian)

Examination Date: 3rd April, 2024 Examination Center: CHRIST University

**Name of Student:** Soumik Dutta Gupta, N. Kirutheeka, Anson Thomas

**Class:** VI Semester, PME

**Register Number:** 2140309, 2140317, 2140319



**Acknowledgement**

We would like to thank professors Neethu Joseph and Benny Sebastian for encouraging us to take on this project and their invaluable guidance throughout the course of this undertaking. Additionally, we would like to thank the rest of the department of Physics and Electronics for their guidance and support whenever needed.

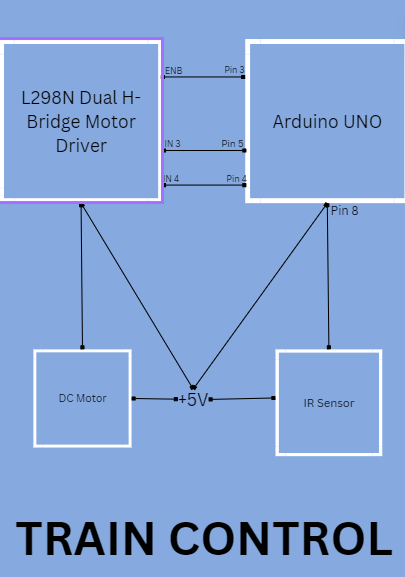
**Index Page**

1. Introduction …………………………………………………………………………………………………………… 3
2. Project Description ………………………………………………………………………………………………… 4
3. Details of the circuit ………………………………………………………………………………………………. 6
4. Working and Operating Procedure ………………………………………………………………………. 10
5. Software tools and programming used ………………………………………………………………… 12
6. Conclusion and Future Scope
7. References

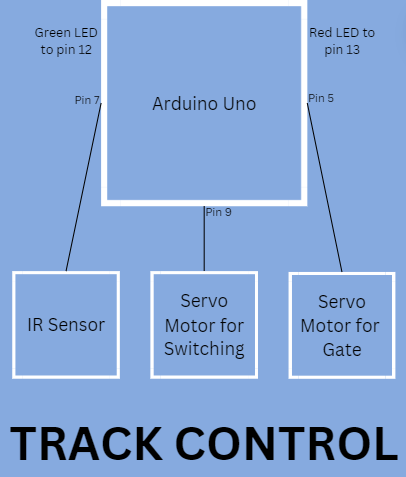
**INTRODUCTION**

The Indian Railway system, a very important aspect of the nation's transportation infrastructure, offers great convenience and affordability. However, yearly losses persist due to management issues and outdated technology, leading to delays, disputes, and accidents. Our solution, an Automatic Railway Signalling and Switching System, addresses these challenges by leveraging rudimentary technology to enhance safety and efficiency. Using components such as Arduino Uno and L298N motor driver, our system detects obstacles and controls track switches with precision. With custom-made tracks and just basic sensors, we ensure seamless operations and minimize accidents. By embracing innovation, we aim to develop a cost effective solution with basic technologies.

**PROJECT DESCRIPTION**

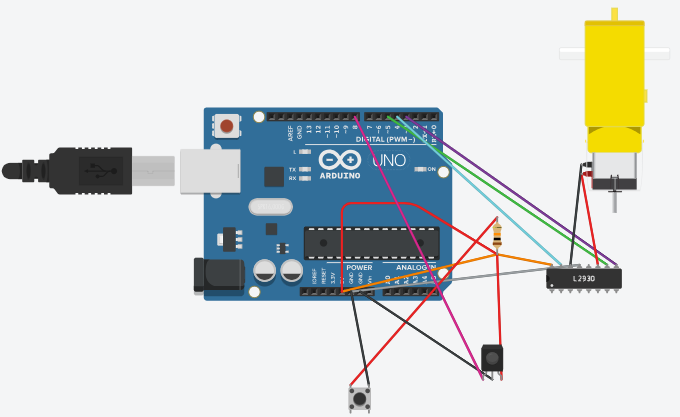


This block diagram shows a simplified version of the electronics and circuitry inside the train model that we have made. The main focus of this is the L298N dual H-bridge motor driver which drives a singular 60 rpm DC motor, connected to two wheels which act as the “driving wheels”. The power supply is provided by a 9V battery connected to the Arduino Uno board through the adapter.



This block diagram is a representation of the track control part of our project. The main components used here is two servo motors for controlling the barrier and the moving part of the track. The LEDs are used for signalling purposes like in real-life traffic signals. The IR Sensor prompts the user through the Arduino-based program for track switching.

**DETAILS OF THE CIRCUIT**



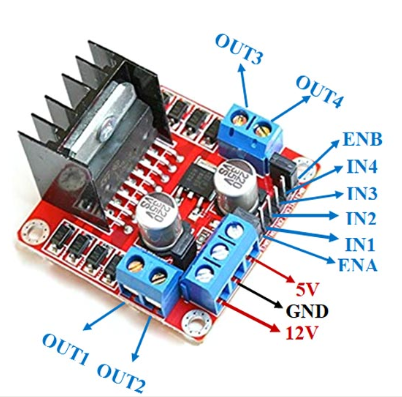
The train control is mainly done by programming the L298N Dual H-bridge motor driver. We require only one motor; hence, we are connecting only one of the power pins of the L298N motor driver to the motor. We choose the input side of the L298N driver such that the motor will operate and thus pull the train in the desired direction. A button is used to indicate whether or not the train will start running or not. When the button is pressed, the motor will turn on and the train starts moving. The infrared sensor helps for obstacle detection and the motors immediately stop when an obstacle falls in the suitable range of the IR sensor.

The major components used in this part are:

* DC Motor: The electrical motor operated by dc that converts dc electrical energy into mechanical energy is called DC motor. DC motor works on principle that when magnetic and electrical fields interact mechanical force is generated which is known as motoring action.

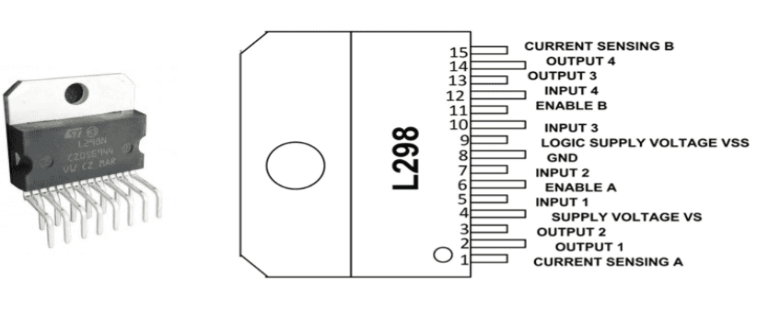


* L298N Dual H-Bridge Motor Driver: This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

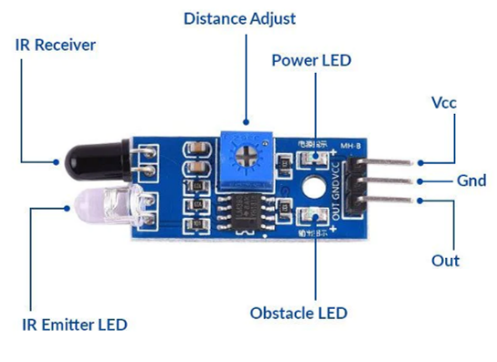


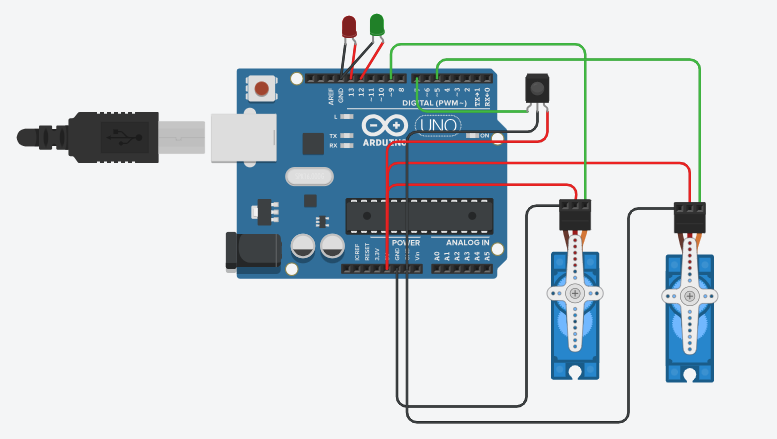
L298 IC is a 15-pin IC. The pin configuration and real image of L298 IC are shown in the figure below:

1. Pin1 (Current Sensing A): This pin is used to control the flow of load current
2. Pin2 (Output 1): This pin is the output pin of the H- Bridge A where the current flows through the load which is monitored at pin 1.
3. Pin3 (Output 2): This pin is the output pin of the H- Bridge A where the current flows through the load which is monitored at pin 1.
4. Pin4 (VS): This is a voltage supply pin and is connected to a +5V supply.
5. Pin5 (Input 1): This pin Control the Input of Bridge A and is compatible with TTL.
6. Pin6 (Enable A): This pin is TTL Compatible Enable Input. The LOW state for the disabled.
7. Pin7 (Input 2): This pin Control the Input of Bridge A and is compatible with TTL.
8. Pin8 (GND): This is a Ground pin.
9. Pin9 (Logic Supply Voltage Vss): This pin provides supply voltage for the logic blocks.
10. Pin10 (Inputs3): This pin control inputs of bridge-B and is compatible with TTL
11. Pin11 (Enable B): This pin is TTL Compatible Enable Input. The LOW state for the disabled.
12. Pin12 (Inputs4): This pin control inputs of bridge-B and is compatible with TTL
13. Pin 13 (Output 3): This pin is the output pin of the H- bridge B where the current flow through the load is monitored at pin15.
14. Pin 14 (Output 4): This pin is the output pin of the H- bridge B where the current flow through the load is monitored at pin15.
15. Pin15 (Current Sensing B): This pin is used to control the flow of load current



* IR Sensor: IR sensors use IR LEDs as emitters and IR photodiodes as detectors to measure heat and detect motion by sensing invisible thermal radiation in the infrared spectrum. The system comprises five elements: an IR source (LED or laser), a trans mission medium, optical components, IR detectors, and signal processing. The photodiode’s resistance and output voltage change proportionally to the received IR light.





Gate Servo

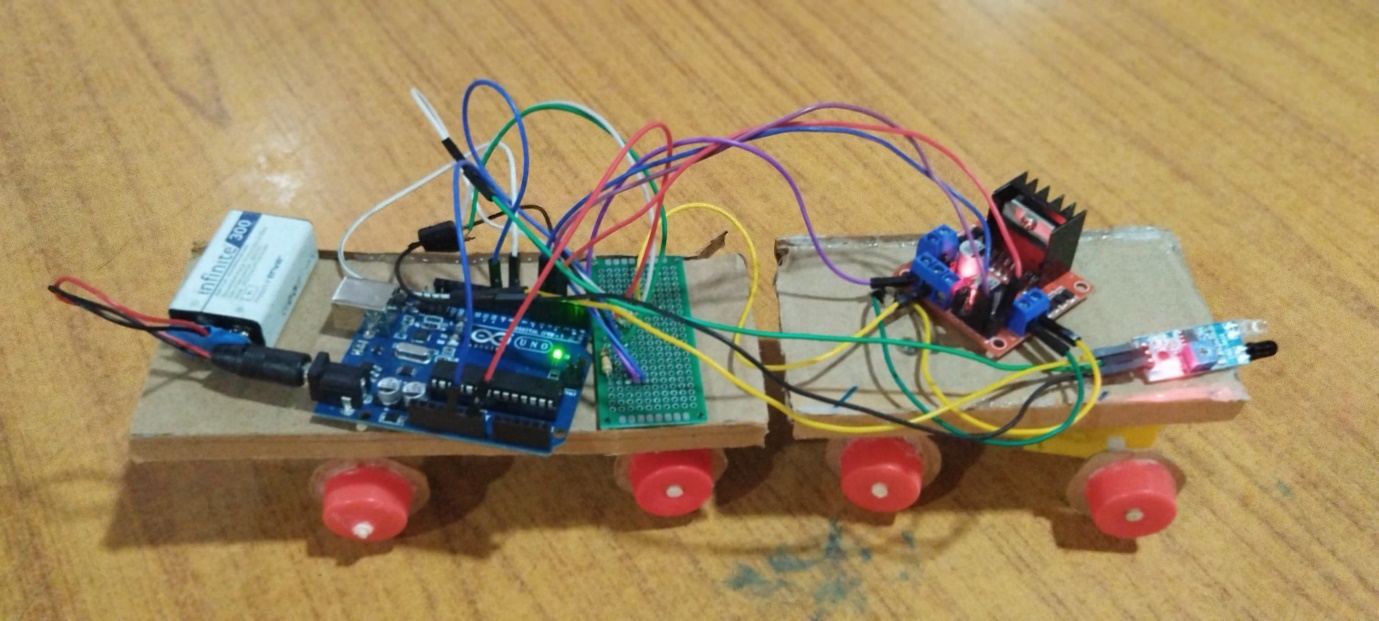
Switching Servo

This circuit is the part which controls the track switching and signalling. When the train is detected by the infrared sensor placed at an appropriate position, we send a prompt to the Arduino code which asks us to which position the track will switch to and the switching servo motor will switch the tracks accordingly. The gate servo is used to control the barrier. The leds act us traffic signals.

The main major component is the Servo Motor. The SG90 Micro Servo Motor is a small, high-performance servo motor commonly used in robotics, model making, and other hobbyist projects. It has a compact form factor and is relatively low-cost, making it an attractive choice for many applications. The SG90 has a 9-gram weight and a size of 22.8 x 11.8 x 22.7 mm, making it small enough to be used in compact and lightweight robotic designs. It has a torque rating of 1.8 kg/cm, which is sufficient for most hobbyist applications and small robotic projects. The servo motor also features a dead-band width of only 1 µs, which provides precise control and positioning of the servo’s output shaft.



**WORKING AND OPERATING PROCEDURE**



The main power supply is provided by a 9V rechargeable battery that is connected to the Arduino via the in-built adapter; and the Arduino automatically regulates the voltage to the desired voltage. We use a PCB to save space and avoid the use of a bread-board and also to make the whole thing a lot less messy. The 5V connections are made via a 10 kΩ resistor and the ground connections are made as usual. A button is also connected to the Arduino which is just used for starting and stopping the train. The 60 rpm DC motor is connected to the IN3 and IN4 pins’ side and the direction is controlled according to our desire. An infrared sensor is also integrated at the front and connected to the Arduino. For the tracks, we have used cardboard to make them and used ice-cream sticks to join them in place as shown below:



First, we press the button to switch on the train. The train starts moving forward, as the motor starts working, along the tracks made as shown. When it reaches near the junction, the IR Sensor detects a barrier and the motors immediately stop, thus stopping the entire train. An infrared sensor placed on the track will detect the train and this will prompt the “Track Switching Code” to select which part the moving track will curve towards, the curved tracks or remain along the same straight track. Accordingly, the track will switch and the train will move, after the barrier has been raised and the train is free to go.

**SOFTWARE TOOLS AND PROGRAMMING USED**

We have used the Arduino IDE, which is basically C++ based, to code our programs and upload them to make our components run properly. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

The codes are given as follows:

**1)** Train Control:

const int buttonPin = 10;    // Pin number for the pushbutton

const int ledPin = 13;      // Pin number for the LED

const int irSensorPin = 8;  // Pin number for the IR sensor (adjust as needed)

int in3 = 5;

int in4 = 4;

int enB = 3;

int buttonState = 0;        // Variable to store the button state

int lastButtonState = 0;    // Variable to store the previous button state

int ledState = LOW;         // Variable to store the LED state

int irSensorState = HIGH;   // Variable to store the IR sensor state

int prevLedState = LOW;     // Variable to store the previous LED state

void setup() {

  pinMode(buttonPin, INPUT);     // Set the pushbutton pin as input

  pinMode(ledPin, OUTPUT);       // Set the LED pin as output

  pinMode(irSensorPin, INPUT);   // Set the IR sensor pin as input

  pinMode (enB, OUTPUT);

  pinMode (in3, OUTPUT);

  pinMode (in4, OUTPUT);

  digitalWrite(in3, LOW);

  digitalWrite(in4, LOW);

  digitalWrite(enB, LOW);

}

void loop() {

  buttonState = digitalRead(buttonPin);        // Read the state of the pushbutton

  irSensorState = digitalRead(irSensorPin);    // Read the state of the IR sensor

  // Check if the button state has changed

  if (buttonState != lastButtonState) {

    // If the button is pressed (LOW), change the LED state

    if (buttonState == LOW) {

      ledState = !ledState;          // Toggle the LED state

      digitalWrite(ledPin, ledState);  // Update the LED

      digitalWrite(in3, ledState);

      digitalWrite(enB, ledState);

    }

    delay(50);  // Debounce delay to avoid multiple readings for a single press

  }

  // Check if motion is detected by the IR sensor

  if (ledState == HIGH && irSensorState == LOW) {

    // Store the previous LED state before turning it off

    prevLedState = ledState;

    ledState = LOW;                 // Turn off the LED if motion is detected

    digitalWrite(ledPin, ledState);

    digitalWrite(in3, ledState);

    digitalWrite(enB, ledState);

  }

  // If motion stops and LED was previously on, restore the LED state

  if (ledState == LOW && prevLedState == HIGH && irSensorState == HIGH) {

    ledState = HIGH;

    digitalWrite(ledPin, ledState);

    digitalWrite(in3, ledState);

    digitalWrite(enB, ledState);

    prevLedState = LOW;  // Reset prevLedState

  }

  // Update the last button state

  lastButtonState = buttonState;

}

2) Track Control:

#include <Servo.h>

Servo servo;

Servo gateServo;

const int redPin = 13;

const int grnPin = 12;

const int infraPin = 7;

int pos = 0;

int irState = HIGH;

void setup() {

  servo.attach(9);

  gateServo.attach(5);

  pinMode(infraPin, INPUT);

  pinMode(redPin, OUTPUT);

  pinMode(grnPin, OUTPUT);

  servo.write(pos);

  gateServo.write(pos);

  Serial.begin(9600);

}

void loop() {

  irState = digitalRead(infraPin);

  digitalWrite(redPin, HIGH);

  digitalWrite(grnPin, LOW);

  //Serial.println(irState);

  if (irState == LOW) {

    // Read direction: left/right

    Serial.println("Enter Direction 'a' for Hogwarts and 'b' for Wonderland:");

    while (Serial.available() == 0) {}

    String dirn = Serial.readString();

    dirn.trim();

    if (dirn == "a") {

      if (pos <= 0) {

        gate();

      } else {

        for (pos = 90; pos >= 1; pos -= 1) {

          servo.write(pos);

          delay(15);

        }

        gate();

      }

    } else if (dirn == "b") {

      if (pos <= 0) {

        for (pos = 0; pos <= 90; pos += 1) {

          servo.write(pos);

          delay(15);

        }

        gate();

      } else {

        gate();

      }

    } else {

      Serial.println("Wrong Direction!");

    }

  }

}

void gate() {

  Serial.println("Hello");

  for (pos = 0; pos <= 90; pos += 1) {

    gateServo.write(pos);

    delay(15);

  }

  digitalWrite(redPin, LOW);

  digitalWrite(grnPin, HIGH);

  delay(1000);

  digitalWrite(redPin, HIGH);

  digitalWrite(grnPin, LOW);

  for (pos = 90; pos >= 1; pos -= 1) {

    gateServo.write(pos);

    delay(15);

    }

}

**CONCLUSION AND FUTURE SCOPE**

The "Automatic Railway Tracking and Switching" project has demonstrated the feasibility and effectiveness of automated systems for improving railway safety and efficiency. Through rigorous testing, it has shown promising results in minimizing errors and optimizing routing. Future development can focus on advancing sensor technologies, integrating predictive analytics, expanding system coverage, addressing cybersecurity concerns, and fostering collaborations with industry stakeholders. These efforts aim to further enhance safety, efficiency, and reliability in railway transportation globally.

**REFERENCES**

Building Train Model:

1. <https://www.youtube.com/watch?v=EmiLMTgpgSI>
2. <https://www.youtube.com/watch?v=WZzEw-er5UI>
3. <https://www.instructables.com/How-to-Make-Electric-Remote-Train-From-a-Cardboard/>

Track Switching using Servo Motor:

1. <https://www.youtube.com/watch?v=dxNanN-tw_0>
2. <https://www.youtube.com/watch?v=_M6vhDvmtrI>

Rechargeable battery

1. <https://www.flipkart.com/hubx-combo-powerbank-module-battery-case-5-watts-psu/p/itme3bd95d6da3eb?pid=PSUGB3XCZNYU65BF&lid=LSTPSUGB3XCZNYU65BFPKKHEH&marketplace=FLIPKART&cmpid=content_psu_8965229628_gmc>

L289N Driver

1. <https://www.youtube.com/watch?v=E2sTbpFsvXI>
2. <https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/>