DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi - 590018)
Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout
Bengaluru-560111

Mini Project Report on

"EV Charging Station for Warehouse Bots"

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Electrical and Electronics Engineering

Submitted by:

1. THEJAS DR	1DS22EE096
2. SACHIN GOWDA P	1DS23EE420
3. SUNIL	1DS23EE426
4. THEJAS GOWDA MR	1DS23EE427

Under the Guidance of

Mr. SATISH BA
Assistant Professor
Dept. of E&E Engg.
DSCE, Bengaluru

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI-590018
2024-25

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi – 590018, Approved by AICTE & ISO 9001:2015 Certified)
Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout
Bengaluru-560111
2024-2025

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

CERTIFICATE

Certified that the mini project report entitled **"EV Charging Station for Warehouse Bots**" carried out by **THEJAS DR(1DS22EE096), SACHIN GOWDA P(1DS23EE420), SUNIL(1DS23EE426), THEJAS GOWDA MR(1DS23EE427)** are bonafide students of **DAYANANDA SAGAR COLLEGE OF ENGINEERING**, an autonomous institution affiliated to VTU, Belagavi in partial fulfillment for the award of Degree of **Bachelor of Engineering in Electrical and Electronics Engineering** during the year **2024-2025**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of work prescribed for the said Degree.

Signature of the Guide
Mr. Satish BA
Assistant Professor
Dept. of E&E Engg.
DSCE, Bengaluru

Signature of the HOD
Dr. Premkumar M.
Professor & HOD
Dept. of E&E Engg.
DSCE, Bengaluru

Name of the Examiners	**Signature with date**
1.	**..**
2.	**..**

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi – 590018, Approved by AICTE & ISO 9001:2015 Certified)
Accredited by NBA, National Assessment & Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout
Bengaluru-560111
2024-2025

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

DECLARATION

We, THEJAS DR(1DS22EE096), SACHIN GOWDA P(1DS23EE420), SUNIL(1DS23EE426) and THEJAS GOWDA MR(1DS23EE427) respectively, hereby declare that the mini project work entitled "EV Charging Station for Warehouse Bots" has been independently done by us under the guidance of 'SATISH BA', Assistant Professor, EEE department and submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Electrical & Electronics Engineering, at Dayananda Sagar College of Engineering, an autonomous institution affiliated to VTU, Belagavi during the academic year 2024-2025.

We further declare that we have not submitted this report either in part or in full to any other university for the award of any degree.

THEJAS DR	**1DS22EE096**
SACHIN GOWDA P	**1DS23EE420**
SUNIL	**1DS23EE426**
THEJAS GOWDA MR	**1DS23EE427**

PLACE: BENGALURU

DATE: 29-5-2025

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible and under whose constant guidance and encouragement the task was completed. We express our sincere regards to the **Management of Dayananda Sagar College of Engineering, Bengaluru.**

We express our sincere regards and thanks to **Dr. B G Prasad, Principal, Dayananda Sagar College of Engineering, Bengaluru.** His incessant encouragement, guidance and valuable support have been an immense help in realizing this mini project.

We express our sincere regards and thanks to **Dr. Premkumar M, Professor & HOD, Department of Electrical and Electronics Engineering, Dayananda Sagar College of Engineering, Bengaluru.** His incessant encouragement, guidance and valuable technical support have been of immense help in realizing this mini project. His guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedicated and hard work.

We would like to express profound gratitude to our guide **Mr. Satish BA, Assistant Professor, Department of Electrical and Electronics Engineering, Dayananda Sagar College of Engineering, Bengaluru** who has encouraged us throughout the mini project. **His** moral support enabled us to complete our work successfully.

We express our sincere thanks to Mini project Coordinator **Dr. Suchana Mishra**, **Assistant Professor, Department of Electrical and Electronics Engineering** for her continued support and guidance. We thank all teaching and non-teaching staff of the Electrical and Electronics Engineering department for their kind and constant support throughout the academic Journey.

THEJAS DR	**1DS22EE096**
SACHIN GOWDA P	**1DS23EE420**
SUNIL	**1DS23EE426**
THEJAS GOWDA MR	**1DS23EE427**

CONTENTS

List of Figures

Department of Electrical and Electronics Engineering

(Accredited by NBA Tier 1: 2022-2025)

CO-PO Mapping

COURSE OUTCOMES

CO1: Identify a research problem by doing literature survey with supervisor's advice.

CO2: Apply the fundamental knowledge of mathematics, science and engineering principles in design of solutions of system components.

CO3: Select and apply a suitable engineering/IT tool in modeling/data interpretation /analytical studies, conduct experiments leading to a logical solution.

CO4: Design a system/ system component/process, build it and test its functioning as a solution to a complex engineering problem with sustainable practice.

CO5: Communicate effectively to a diverse audience about the outcome and societal impact; develop technical reports and publication.

CO-PO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1		2											2	2	1
CO2	1												2	2	1
CO3	2				2								2	2	1
CO4		3	3	3	3	3	3						2	2	1
CO5						3		3		3		3	2	2	1

	PO1	PO2	PO3	PO4	PO5	PO6	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1		2									2	2	1
CO2	1										2	2	1
CO3	2				2						2	2	1
CO4		3	3	3	3	3					2	2	1
CO5						3	3		3	3	2	2	1

ABSTRACT

In modern industrial and logistics environments, the adoption of autonomous warehouse robots—such as Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs)—has significantly enhanced productivity, accuracy, and operational efficiency. However, the effectiveness of these robotic systems largely depends on reliable, uninterrupted power supply and efficient energy management. This project introduces a wireless EV charging station specifically designed for warehouse robots, addressing the limitations of traditional plug-in or contact-based charging methods through the application of inductive power transfer technology.

The wireless charging system operates through electromagnetic induction, allowing energy to be transferred from a stationary charging pad to a receiver coil embedded within the robot. This setup eliminates the need for physical connectors or manual charging processes, which can be prone to wear, misalignment, or downtime. Charging pads can be embedded into the warehouse floor at predefined locations allowing for opportunity charging while the robot is temporarily stationary during routine tasks or scheduled breaks.

The system is integrated with a smart charging management platform that continuously monitors the battery status of robot, manages charging cycles, and prioritizes power distribution based on operational needs. This minimizes the risk of battery depletion and ensures that robots are charged optimally without interrupting workflow.

In addition to improving energy efficiency, the wireless charging station enhances safety by reducing exposed electrical contacts and eliminating the need for human intervention. It also contributes to sustainability goals by optimizing energy use and supporting battery health.

Overall, the proposed wireless charging solution represents a future-ready infrastructure for autonomous warehouses, offering a clean, safe, and intelligent method for maintaining continuous robot operation in high-demand environments.

Chapter-1

Introduction

1.1. Background

Conventional EV charging systems in warehouses typically rely on conductive charging, where robots must dock at a station and establish a physical connection for power transfer. While effective, this method presents several challenges, including connector wear, alignment issues, and operational delays due to charging schedules. As the number of robots in warehouse environments increases, the demand for more efficient, scalable, and maintenance-free charging solutions grows. Wireless power transfer (WPT) specifically inductive charging emerges as a promising alternative. This technology enables power transmission through magnetic fields without direct contact, allowing robots to charge opportunistically while performing their routine tasks. By embedding charging pads into strategic warehouse locations, robots can maintain charge without interrupting workflow.

1.2. Motivation

The primary motivation behind this project is to overcome the limitations of contact-based charging systems and to support the next generation of smart, autonomous warehouses. Downtime for charging reduces the productivity of robotic fleets and increases the need for complex scheduling. Additionally, physical connectors are prone to damage and require regular maintenance. Wireless charging offers a cleaner, safer, and more efficient solution, especially when integrated with intelligent fleet management software that can optimize charging schedules and battery usage. The goal is to create a system that not only reduces operational costs and maintenance needs but also contributes to a safer and more flexible warehouse environment.

1.3. Objectives

1. To design and implement a wireless EV charging system tailored for autonomous warehouse robots using inductive power transfer.
2. To integrate smart charging management software capable of real-time monitoring, scheduling, and energy optimization.
3. To enhance robot uptime and reduce operational interruptions by enabling opportunity charging during idle times.
4. To improve safety and reduce maintenance costs by eliminating physical connectors and manual intervention.
5. To create a scalable, modular charging infrastructure that can grow with increasing fleet sizes and warehouse demands.

Chapter-2

Literature Review

2.1. Introduction

Wireless charging has gained significant attention in recent years as a viable solution for charging electric vehicles (EVs), including mobile service robots and industrial AGVs. The fundamental principle relies on magnetic resonance coupling or inductive coupling between a transmitter (charging pad) and a receiver (embedded in the device). Researchers such as Kurs et al. (2007) and Covic & Boys (2013) demonstrated the feasibility of wireless power transfer over short distances with high efficiency, laying the foundation for its application in industrial settings.

2.2. Review of existing/similar projects

1. Kiwibot's Wireless Charging Stations

Kiwibot, an autonomous delivery robot company, developed wireless charging stations to enhance the operational efficiency of their robots. The project commenced with successful testing at New Mexico State University, demonstrating significant improvements in charging speed and efficiency compared to traditional methods. Prototypes were developed and tested in controlled environments, addressing challenges such as water infiltration and alignment issues. The wireless charging stations were later deployed at both New Mexico State University and Jacksonville State University campuses for real-world testing. kiwibot.com

2. TU Delft's Autonomous Wireless Charging System for Robot Swarms

Researchers at TU Delft focused on building a wireless charging station for Lunar Zebro robots, aiming to match the interfacing voltage and current requirements for the wireless power transceiver. The project involved designing a charging station compatible with the robots' specifications to ensure efficient and autonomous charging capabilities. repository.tudelft.nl

3. Autonomous Wireless Charging System for Multi-Rotor UAVs

A study explored the development of an autonomous wireless charging system for multi-rotor unmanned aerial vehicles (UAVs). The system utilized an open-source AR.Drone 2.0 with onboard wireless power transfer (WPT) electronics. The charging station was equipped with transmitting coils and WPT electronics to power the UAVs during landing. The system demonstrated the feasibility of autonomous charging for UAVs, contributing to extended operational times without manual intervention. MDPI

4. Autonomous Trash Collector Robot with Wireless Charging

An autonomous trash collector robot was developed for campus environments, featuring wireless charging capabilities powered by solar energy. The robot utilized GSM and GPS modules for localization and status updates. The wireless charging system allowed the robot to autonomously recharge, ensuring continuous operation without manual intervention. The project demonstrated the integration of renewable energy sources with autonomous charging systems for sustainable operations. repository.biust.ac.bw

5. Development of Wireless Charging Robot for Indoor Environments

A project aimed at developing a robotic system capable of navigating indoor environments to charge multiple electrical devices. The system employed algorithms such as the Traveling Salesman Problem, Probabilistic Roadmap, and Fuzzy C-means Clustering for optimal path planning. The robot was designed to optimize performance for charging various devices, with computation tasks offloaded to a remote server to reduce the computational load on the robot. The project highlighted the potential of autonomous robots in managing wireless charging tasks in indoor settings

2.3. Identified gaps

- **Robots Have to Park Perfectly to Charge**

  Many systems only work if the robot stops in just the right spot. If it’s even a little off, it might not charge properly.

- **Different Systems Don’t Work Well Together**

  Each project uses its own custom hardware and software. That means if you have different types of robots, they might not all be able to use the same charging station.

- **Takes Too Long to Charge**

  In some cases, charging still takes a long time — which means the robot has to sit and wait instead of doing its job.

- **Expensive to Set Up**

  Putting in wireless charging systems can be costly. It might mean installing special pads in the floor, upgrading the robots, and reprogramming how everything works.

- **Robots Don’t Always Know When to Charge**

  Some systems don’t give robots a clear way to decide when they need a charge. So, they might wait too long and shut down — or go charge too early and waste time.

- **Doesn't Use Clean Energy Smartly**

  Most projects don't think about where the power comes from. Even if solar panels or clean energy are available, the system may not use them efficiently.

- **No Way to Monitor or Fix Problems Easily**

  If something goes wrong like a robot isn't charging properly many systems don't have a good way to alert workers or fix the issue automatically.

- **Robots Still Need Human Help Sometimes**

  Even though it's called "autonomous" charging, in some setups, humans still need to adjust or position the robot if it misses the mark.

- **Doesn't Consider Wear and Tear on Batteries**

  Charging too often or at the wrong times can slowly damage batteries, reducing their life and increasing replacement costs.

- **Energy Waste Is Still a Problem**

  Wireless systems often lose some energy in the transfer process — especially if alignment isn't perfect. That adds up over time.

2.4. Solutions

- Use better charging pads with larger or multiple coils, or systems that automatically guide the robot into the right spot.
- Improve charging speed with faster wireless tech and make sure robots charge during natural breaks like while waiting for the next taskinstead of running completely out first.
- Design smaller, modular charging stations that can be installed one at a time.
- Use AI or rules-based software that teaches robots to charge when they have free time
- Connect the chargers to solar panels or energy management systems.
- Add simple sensors and notifications. If something goes wrong, the system can send a message to the team.
- Add more small charging stations in quiet spots around the warehouse so robots don't all head to one place.
- Use line follower and sensors that help robots find and line up with the charger automatically like a self-parking car.
- Use SOC indication and sensors that tracks battery health and charges in ways that extend life like slow-charging at night and avoiding overcharging.

Chapter-3

System Design and Analysis

3.1. Block Diagram

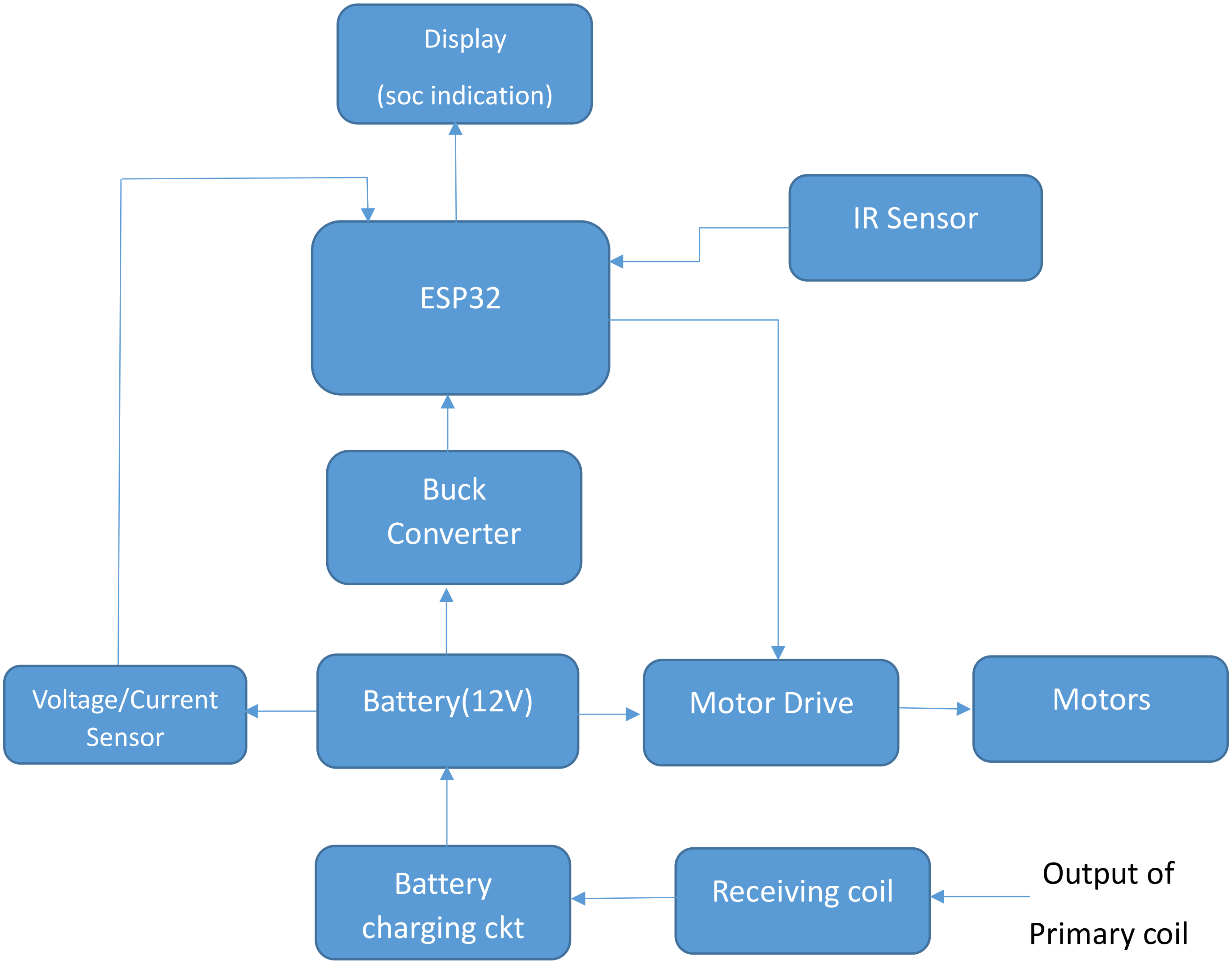

Fig.3.1 Block Diagram of Robot (Bot)

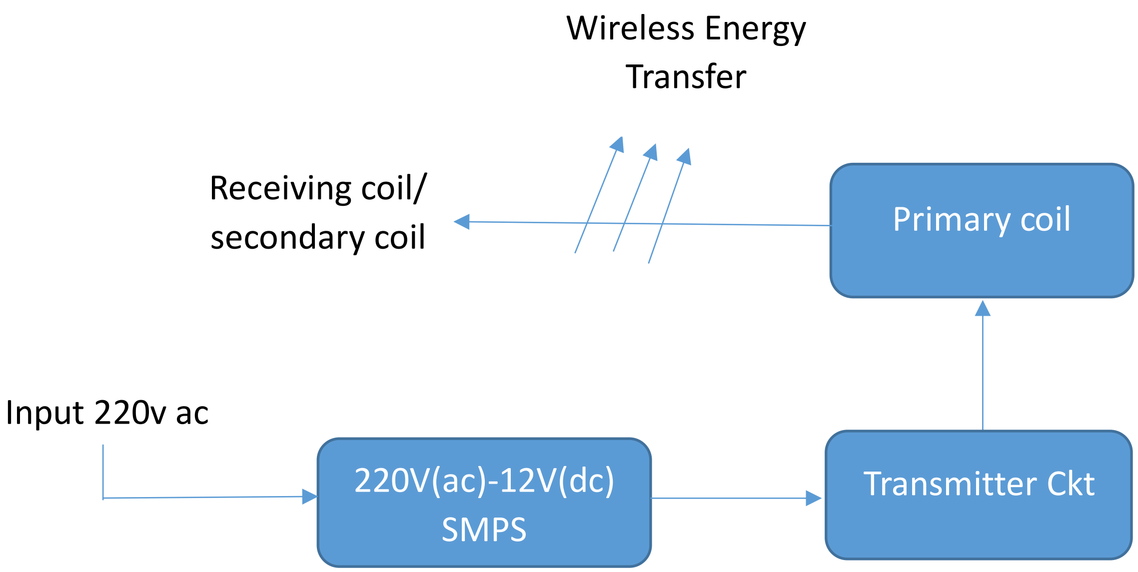

Fig.3.2 Block Diagram of Charging Station

3.2. Components/Tools used

3.2.1. ROBOT

- **ESP32**

  The ESP32 acts like the brain of our wireless charging system. It watches what's happening, makes decisions, and talks to other devices.

Fig.3.2.1.1 ESP32

- **Motors**

  Motors are the parts that actually make the robot move. They turn electrical energy into mechanical movement.

Fig.3.2.1.2 Motors

- **Motor Drive**

  A motor drive is like the robot’s muscle controller. It takes commands from the brain (ESP32) and controls the power sent to the motor.

Fig.3.2.1.3 Motor Dive

- **Battery(3.7*3V)**

  A battery stores electrical energy that powers the robot’s motors, sensors, and the ESP32.

Fig.3.2.1.4 Li-ion Battery

- **Voltage and Current Sensor**

  A voltage sensor measures the out put voltage of the battery.

  In a robot or charging station, it tells you how much voltage the battery or charger is providing**.**

A current sensor measures how much electricity (current) is available in the battery.

It tells you how much power the motor or charger is using at any moment.

Fig.3.2.1.5 Voltage and Current sensor

- **IR Sensor**

  Here we used 3 sensors, these detects the path for the bot. Acts like eyes of the bot.

Fig.3.2.1.6 IR sensor

- **Charging Circuit**

  It uses Diods and Capacitors.

  The capacitor connected near the battery acts like a cushion, reducing voltage fluctuations.

- The diode protects battery and circuit from damage by preventing reverse current.

Fig.3.2.1.7 Capacitor and Diode

- **Display Module**

Basically it is a LCD display, which shows the indication of the SOC of the bot.\

Fig.3.2.1.8 LCD Display

3.2.2. Charging Station

- **Inductive Coil**

  It’s a wire loop (or several loops) made of copper wire. Acts like a wireless power transmitter or receiver**.**

Fig.3.2.2.1 Inductive coil

- **MOSFET**

  It acts like a smart switch that can turn power ON or OFF very quickly and efficiently.

  Controlled by a small signal (like from the ESP32), it can switch large currents flowing to your coils or motors.

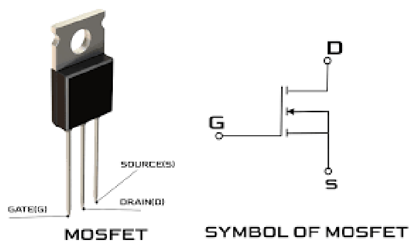

Fig.3.2.2.3 MOSFET

- **Resistor**

  A resistor is an electronic component that resists the flow of electric current.

  Purpose of Resistor, it protects sensitive parts (like the ESP32 or sensors) by controlling how much current flows through them.

  To divide voltage in circuits so components get the right amount.

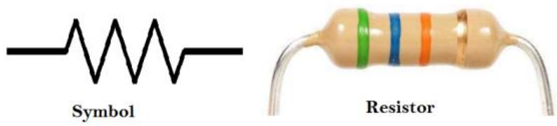

Fig.3.2.2.4 Resistor

- **Zener Diode**

  A Zener diode is a special type of diode that allows current to flow normally in one direction like a regular diode but also allows current to flow backward when the voltage reaches a certain set value**.** When voltage rises above the Zener voltage, the Zener diode starts conducting in reverse and prevents the voltage from going any higher**.**

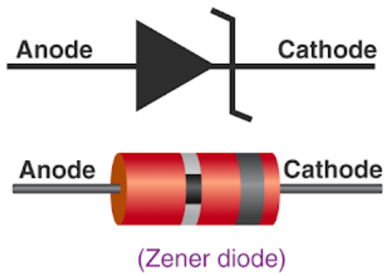

Fig.3.2.2.4 Zener Diode

- **Capacitor Coil**

  The coil and capacitor together create a resonant circuit, which means they "ring" at a specific frequency.

  This resonance makes the wireless energy transfer much more efficient like tuning a radio to the right station.

3.3. Circuit Diagram

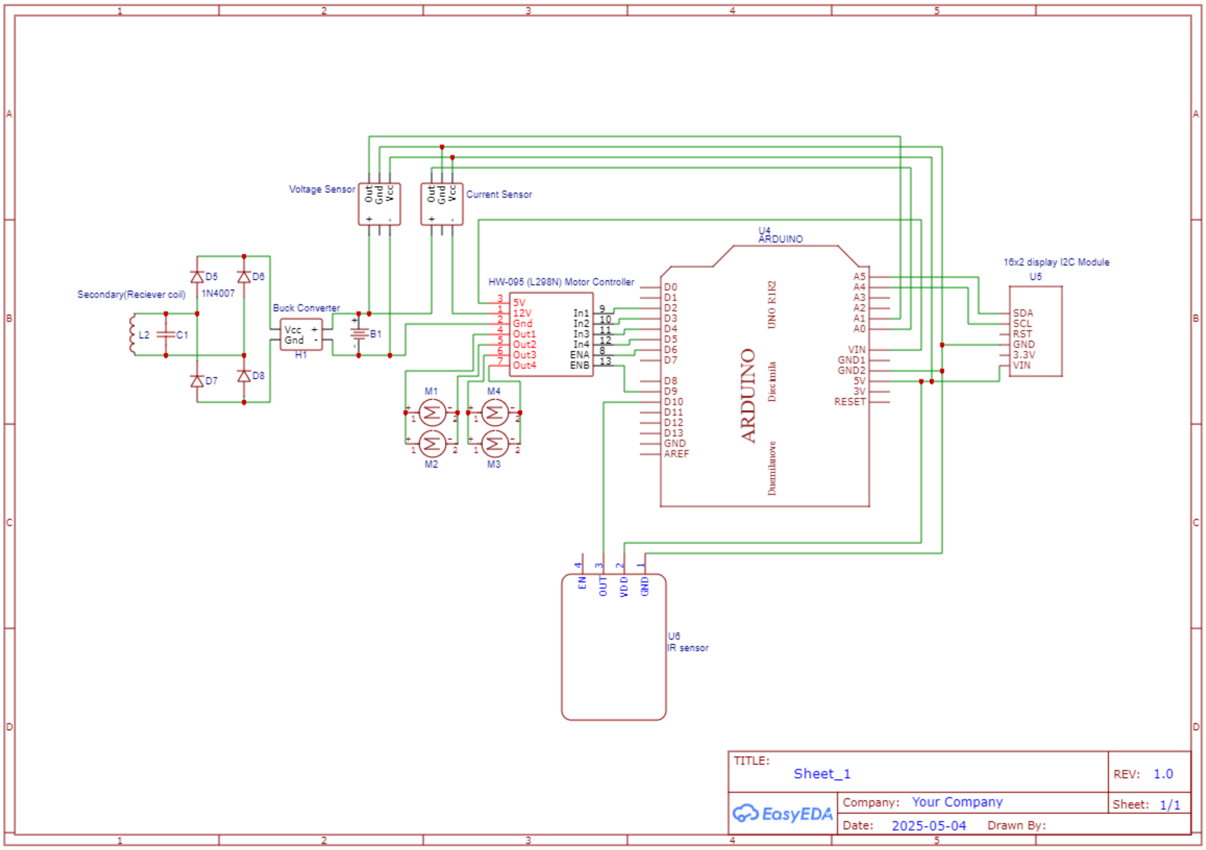

Fig.3.3.1 Circuit Diagram of Bot

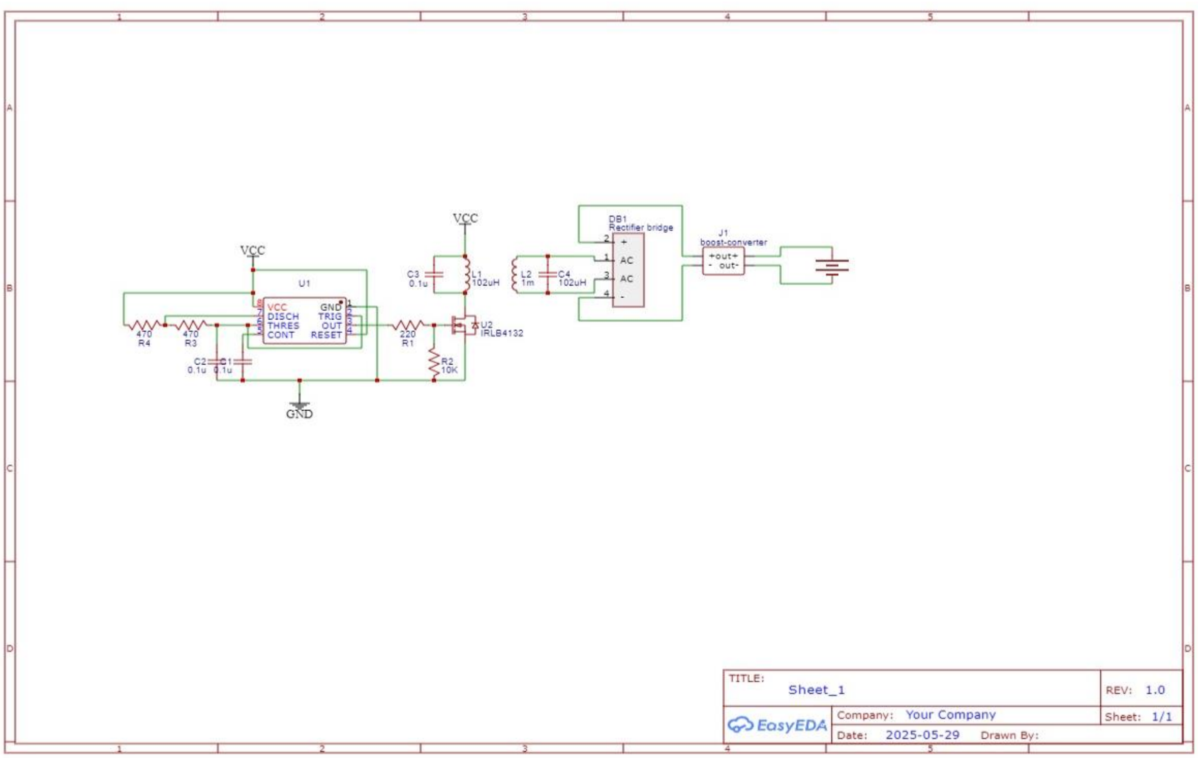

Fig.3.3.2 Circuit Diagram of Charing Station

Chapter-4

System Implementation

4.1. Embedded C Program

4.1.1. Code for Bot

```
#include <Arduino.h>
bool leftdone=false;
bool rightdone=false;
bool stopped=true;
bool juncdetected=false;
bool enddetected=false;

int leftState;
int middleState;
int rightState;

int leftIR = 9;
int middleIR = 10;
int rightIR = 11;
int M1F = 4;
int M1R = 5;
int M1Speed = 3;
int M2F = 8;
int M2R = 7;
int M2Speed = 6;

String lastCommand = "stop";
String recent = "start";
int juncs=0;

#include "BluetoothSerial.h"
#include <Arduino.h>
BluetoothSerial serialBT;

char btSignal;

#define R 0
#define L 1
//initial Speed
int Speed = 90;

//declare channel for pwm Output
#define R 0
#define L 1

//PWM Pin for Controlling the speed
int enA = 26;
int enB = 25;
```

```
//motor controlling pin

int IN4 = 13;
int IN3 = 12;
int IN2 = 14;
int IN1 = 27;
*/
void setup() {

  serialBT.begin("Cool Car");

  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);

  ledcSetup(R, 5000, 8);  // Channel 0 for Motor A, 5 kHz frequency, 8-bit resolution
  ledcAttachPin(enA, R);
  ledcSetup(L, 5000, 8);  // Channel 0 for Motor A, 5 kHz frequency, 8-bit resolution
  ledcAttachPin(enB, L);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  stop();

  pinMode(M1F, OUTPUT);
  pinMode(M1R, OUTPUT);
  pinMode(M2F, OUTPUT);
  pinMode(M2R, OUTPUT);

  pinMode(leftIR, INPUT);
  pinMode(middleIR, INPUT);
  pinMode(rightIR, INPUT);

  pinMode(M1Speed, OUTPUT);
  pinMode(M2Speed, OUTPUT);
  analogWrite(M1Speed, 90);
  analogWrite(M2Speed, 90);
}

void loop() {

  while (serialBT.available()) {
    btSignal = tolower((char)serialBT.read());
    if (btSignal=="a"){
      if (recent!="start"){
        navigate(recent,"start");
      }
    }
    else if (btSignal=="b"){
```

```
      if (recent!="L1"){
        navigate(recent,"L1");
      }
    }
    else if (btSignal=="c"){
      if (recent!="R1"){
        navigate(recent,"R1");
      }
    }
    else if (btSignal=="d"){
      if (recent!="L2"){
        navigate(recent,"L2");
      }
    }
    else if (btSignal=="e"){
      if (recent!="R1"){
        navigate(recent,"R1");
      }
    }
    else if (btSignal=="f"){
      if (recent!="end"){
        navigate(recent,"end");
      }
    }
  }

}

void IRstates(){
  leftState = digitalRead(leftIR);
  middleState = digitalRead(middleIR);
  rightState = digitalRead(rightIR);
}

void lineFollow(){
  IRstates();
  if (middleState == HIGH) {
    forward();
    lastCommand = "forward";
  }
  else if (leftState == HIGH) {
    left();
    lastCommand = "left";
  }
  else if (rightState == HIGH) {
    right();
    lastCommand = "right";
  }
  else {
    if (lastCommand == "forward") {
      forward();
```

```
    }
    else if (lastCommand == "left") {
      left();
    }
    else if (lastCommand == "right") {
      right();
    }
    else {
      stop();
    }
  }
}

void leftjunc(){
  while (juncdetected == true){
    left();
    IRstates();
    if (leftState==HIGH){
      while (juncdetected == true){
        left();
        IRstates();
        if (middleState==HIGH){
          juncdetected = false;
        }
      }
    }
  }
  lastCommand = "left";
}

void rightjunc(){
  while (juncdetected == true){
    right();
    IRstates();
    if (rightState==HIGH){
      while (juncdetected == true){
        right();
        IRstates();
        if (middleState==HIGH){
          juncdetected = false;
        }
      }
    }
  }
  lastCommand = "right";
}

void juncturner(String direction){
  juncs=0;
  while(juncs==0){
    IRstates();
```

```
    if (leftState == HIGH && middleState == HIGH && rightState == HIGH){
      juncdetected=true;
      stop();
      juncs++;
      delay(300);
    }
    if (juncdetected==true){
      if (direction=="right"){
        rightjunc();
      }
      else if (direction=="left"){
        leftjunc();
      }
      else if (direction=="straight"){
        stop();
        lastCommand = "forward";
      }
    }
    else{
      lineFollow();
    }
  }
}

void uturn(){
  while(juncdetected==true){
    backward();
    IRstates();
    if (
      (leftState == LOW && middleState == LOW && rightState == LOW)  ||

      (leftState == HIGH && middleState == LOW && rightState == LOW) ||
      (leftState == LOW && middleState == LOW && rightState == HIGH) ||


      (leftState == HIGH && middleState == LOW && rightState == HIGH)
    )
    {
      juncdetected = false;
      delay(500);
      stop();
    }
  }
  while(juncdetected==false){
    right();
    IRstates();
    if (rightState==HIGH){
      while (juncdetected == false){
        right();
        IRstates();
        if (middleState==HIGH){
```

```
      juncdetected = true;
      stop();
     }
    }
   }
  }
  juncdetected = false;
  lastCommand = "right";
}

void uturner(){
  juncs=0;
  while(juncs==0){
    IRstates();
    if (leftState == HIGH && middleState == HIGH && rightState == HIGH){
      juncdetected=true;
      stop();
      juncs++;
      delay(300);
    }
    if(juncdetected==true){
      uturn();
    }
    else{
      lineFollow();
    }
  }
}
// --- Motor control functions ---
void left() {
  digitalWrite(M1F, LOW);
  digitalWrite(M1R, HIGH);
  digitalWrite(M2F, HIGH);
  digitalWrite(M2R, LOW);
}

void right() {
  digitalWrite(M1F, HIGH);
  digitalWrite(M1R, LOW);
  digitalWrite(M2F, LOW);
  digitalWrite(M2R, HIGH);
}

void forward() {
  digitalWrite(M1F, HIGH);
  digitalWrite(M1R, LOW);
  digitalWrite(M2F, HIGH);
  digitalWrite(M2R, LOW);
}

void backward() {
```

```
  digitalWrite(M1F, LOW);
  digitalWrite(M1R, HIGH);
  digitalWrite(M2F, LOW);
  digitalWrite(M2R, HIGH);
}

void stop() {
  digitalWrite(M1F, LOW);
  digitalWrite(M1R, LOW);
  digitalWrite(M2F, LOW);
  digitalWrite(M2R, LOW);
}

void navigate(String current, String destination){

  if (current=="start"){
    if(destination=="R1"){
      juncturner("right");
      uturner();
      recent="R1";
    }
    else if(destination=="L1"){
      juncturner("left");
      uturner();
      recent="L1";
    }
    else if(destination=="R2"){
      juncturner("straight");
      juncturner("right");
      uturner();
      recent="R2";
    }
    else if(destination=="L2"){
      juncturner("straight");
      juncturner("left");
      uturner();
      recent="L2";
    }
    else if(destination=="end"){
      juncturner("straight");
      juncturner("straight");
      uturner();
      recent="end";
    }
  }
  else if (current=="L1"){
    if(destination=="start"){
      juncturner("right");
      uturner();
      recent="start";
    }
```

```
  else if(destination=="R1"){
    juncturner("straight");
    uturner();
    recent="R1";
  }
  else if(destination=="L2"){
    juncturner("left");
    juncturner("left");
    uturner();
    recent="L2";
  }
  else if(destination=="R2"){
    juncturner("left");
    juncturner("right");
    uturner();
    recent="R2";
  }
  else if(destination=="end"){
    juncturner("left");
    juncturner("straight");
    uturner();
    recent="end";
  }
}

else if (current=="R1"){
  if(destination=="start"){
    juncturner("left");
    uturner();
    recent="start";
  }
  else if(destination=="L1"){
    juncturner("straight");
    uturner();
    recent="L1";
  }
  else if(destination=="L2"){
    juncturner("right");
    juncturner("left");
    uturner();
    recent="L2";
  }
  else if(destination=="R2"){
    juncturner("right");
    juncturner("right");
    uturner();
    recent="R2";
  }
  else if(destination=="end"){
    juncturner("right");
    juncturner("straight");
```

```
    uturner();
    recent="end";
  }
}

else if (current=="L2"){
  if(destination=="start"){
    juncturner("right");
    juncturner("straight");
    uturner();
    recent="L2";
  }
  else if(destination=="R1"){
    juncturner("right");
    juncturner("left");
    uturner();
    recent="R1";
  }
  else if(destination=="R2"){
    juncturner("straight");
    uturner();
    recent="R2";
  }
  else if(destination=="L1"){
    juncturner("right");
    juncturner("right");
    uturner();
    recent="L1";
  }
  else if(destination=="end"){
    juncturner("left");
    uturner();
    recent="end";
  }
}

else if (current=="R2"){
  if(destination=="start"){
    juncturner("left");
    juncturner("straight");
    uturner();
    recent="start";
  }
  else if(destination=="R1"){
    juncturner("left");
    juncturner("left");
    uturner();
    recent="R1";
  }
  else if(destination=="L2"){
    juncturner("straight");
```

```
      uturner();
      recent="L2";
    }
    else if(destination=="L1"){
      juncturner("left");
      juncturner("right");
      uturner();
      recent="L1";
    }
    else if(destination=="end"){
      juncturner("right");
      uturner();
      recent="end";
    }
  }
  else if (current=="end"){
    if(destination=="R1"){
      juncturner("straight");
      juncturner("left");
      uturner();
      recent="R1";
    }
    else if(destination=="L1"){
      juncturner("straight");
      juncturner("right");
      uturner();
      recent="L1";
    }
    else if(destination=="R2"){
      juncturner("left");
      uturner();
      recent="R2";
    }
    else if(destination=="L2"){
      juncturner("right");
      uturner();
      recent="L2";
    }
    else if(destination=="start"){
      juncturner("straight");
      juncturner("straight");
      uturner();
      recent="start";
    }
  }
}
```

4.1.2. Code for SOC

```
const float BATTERY_CAPACITY_Ah = 1.2;    // Example: 2200mAh = 2.2Ah

const float NOMINAL_VOLTAGE = 12.5;         // Nominal voltage
```

```
const float FULLY_CHARGED_VOLTAGE = 12.86;   // Voltage at 100% SOC
const float CUTOFF_VOLTAGE = 11.5;          // Voltage at 0% SOC

// --- Voltage sensor setup ---
const int voltageSensorPin = A2;           // Voltage sensor pin
const float factor = 5.128;                // Voltage divider reduction factor
const float vCC = 12.8;                    // Arduino reference voltage (usually 5V)

float voltageSensorVal;
float vIn;
float vOut;
// --- SOC variables ---
float SOC = 100.0;         // Start SOC
unsigned long lastTime = 0; // For timing
// --- Variables for measured values ---
float voltageM = 0.0;
float currentM = 0.0;
float voltage() {
  voltageSensorVal = analogRead(voltageSensorPin);
  vOut = (voltageSensorVal / 1024.0) * vCC;
  vIn = vOut * factor;
  return vIn;
}
// --- Current reading function ---
float current() {
  float sum = 0.0;
  for (int i = 0; i < 100; i++) {
    sum += (0.0264 * analogRead(A1) - 13.51) / 1000.0;
  }
  float averageCurrent = sum / 100.0
  return averageCurrent;
```

```
}
// --- SOC Update function ---
void updateSOC(float currentMeasured, float voltageMeasured) {
  unsigned long currentTime = millis();
  float deltaTimeHours = (currentTime - lastTime) / 3600000.0;  // ms to hours
  lastTime = currentTime;
  // Coulomb Counting
  SOC -= (currentMeasured * deltaTimeHours) / BATTERY_CAPACITY_Ah * 100.0;
  // Voltage Correction if resting
  if (abs(currentMeasured) < 0.05) { // Near idle
    float estimatedSOC = mapVoltageToSOC(voltageMeasured);
    SOC = 0.9 * SOC + 0.1 * estimatedSOC; // Blend
  }
  // Clamp SOC
  SOC = constrain(SOC, 0.0, 100.0);
}
// --- Map voltage to SOC ---
float mapVoltageToSOC(float voltageMeasured) {
  if (voltageMeasured >= FULLY_CHARGED_VOLTAGE) return 100.0;
  if (voltageMeasured <= CUTOFF_VOLTAGE) return 0.0;
  return (voltageMeasured - CUTOFF_VOLTAGE) / (FULLY_CHARGED_VOLTAGE - 
CUTOFF_VOLTAGE) * 100.0;
}
```

4.2. Simulation Setup

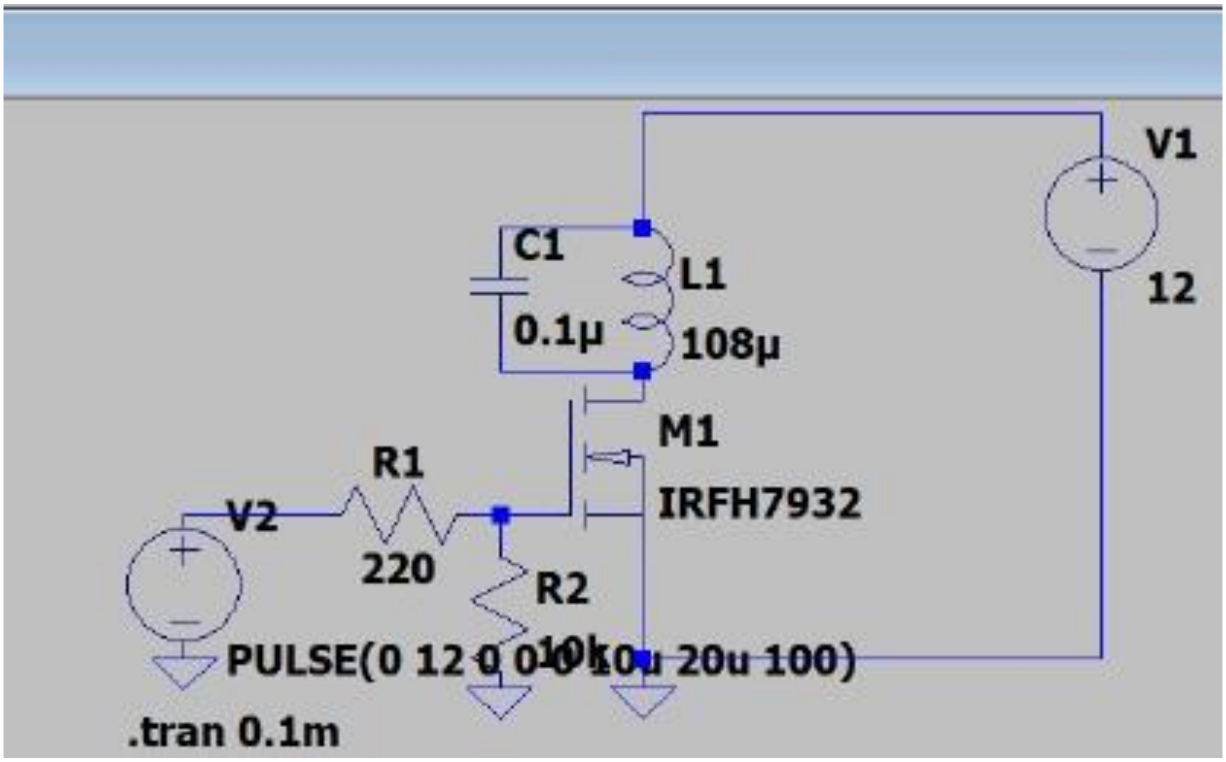

Fig.4.2.1 Simulation setup of Charging Station

4.3. Working

1. **The robot realizes it's running low on battery**

   The robot has sensors and software that constantly check its battery level. When the battery charge drops below a certain safe level (say 20%), the robot knows it needs to recharge before it runs out of power completely.

2. **The robot finds its way to the charging station**

   Once it knows it needs charging, the robot doesn't wait for help. It starts moving on its own toward a wireless charging spot. It uses a **line-following system** to move through the warehouse.

   These wireless chargers are placed in convenient spots near workstations or in resting areas. The robot finds the closest one and heads there.

3. **It stops and aligns with the charger**

   When the robot reaches the charging area, it doesn't just park randomly. It has to stop at a very specific position so the **charging coil inside the robot** lines up properly with the **charging pad on the floor**.

It might use cameras, magnetic sensors, or special markings on the floor to adjust its position. This alignment ensures the power can transfer smoothly from the floor pad to the robot kind of like lining up your phone just right on a wireless charger.

4. **Charging begins automatically — no wires, no plugs**

   As soon as the robot is correctly positioned, the wireless charging system kicks in automatically. No one has to press a button or plug anything in.

- The charging pad on the floor creates a **magnetic field**.
- This field passes through a small air gap and reaches the **receiver coil inside the robot**.
- The robot's coil picks up this energy and turns it into electric power.
- That power is used to start charging the robot's battery.

5. **Smart system controls the charging safely**

   The charging isn't just a dumb "on/off" system. It's smart.

   There are **current sensors**, **voltage sensors**, and **temperature sensors** that constantly monitor the charging process. These sensors check:

- How fast the battery is charging
- If the battery is getting too hot
- How close the battery is to being full

If anything looks wrong (like overheating or too much current), the system slows down or stops the charging to protect the battery. It also avoids overcharging by reducing the current when the battery is almost full.

This smart control makes the battery last longer and keeps the robot safe.

6. **Robot finishes charging and disconnects automatically**

   Once the battery reaches a full charge usually 100%, or a preset safe level like 95% the system automatically stops sending power.

7. **Robot returns to work**

   After charging, the robot smoothly drives away from the charging pad and heads back to work.

   It might go back to picking up packages, moving inventory, delivering parts to a station, or whatever job it was doing before. The whole process is automatic and doesn't interrupt warehouse operations.

In Summary:

The robot behaves just like a smart worker:

- It knows when it's low on energy
- It lines itself up for wireless power
- It charges safely with sensors keeping watch
- It goes back to work once it's full

4.4. Challenges faced during project implementation

- Robot needs to stop in the right spot.
- Wireless charging is usually slower than plug-in charging, so it may take longer to charge.
- Nearby machines or metal parts can affect the wireless signal and reduce charging efficiency.
- The charging system must work smoothly with the robot control system, which can be tricky.
- Wireless charging equipment is more expensive to set up than normal plug-in chargers.
- Charging pads on the floor can get dirty, which may reduce performance over time.

Chapter-5

Results

5.1. Simulation Result

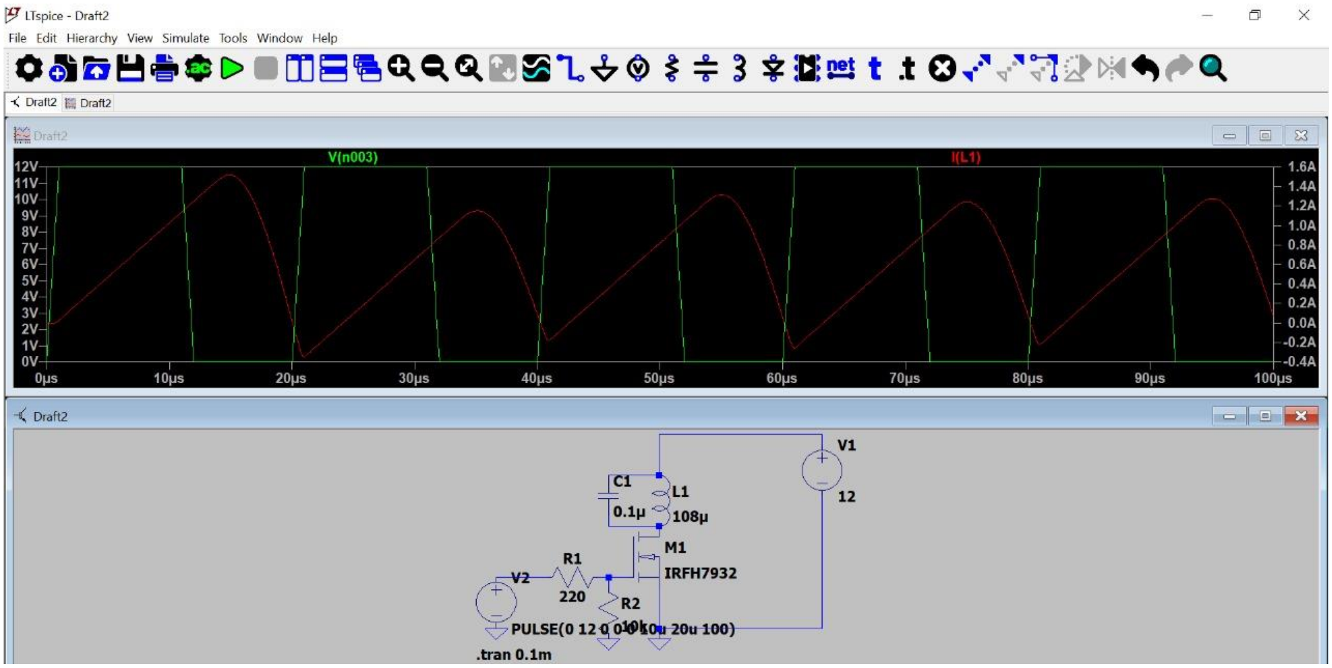

Fig.5.1.1 Simulation result of Charging Station

5.2 System Results

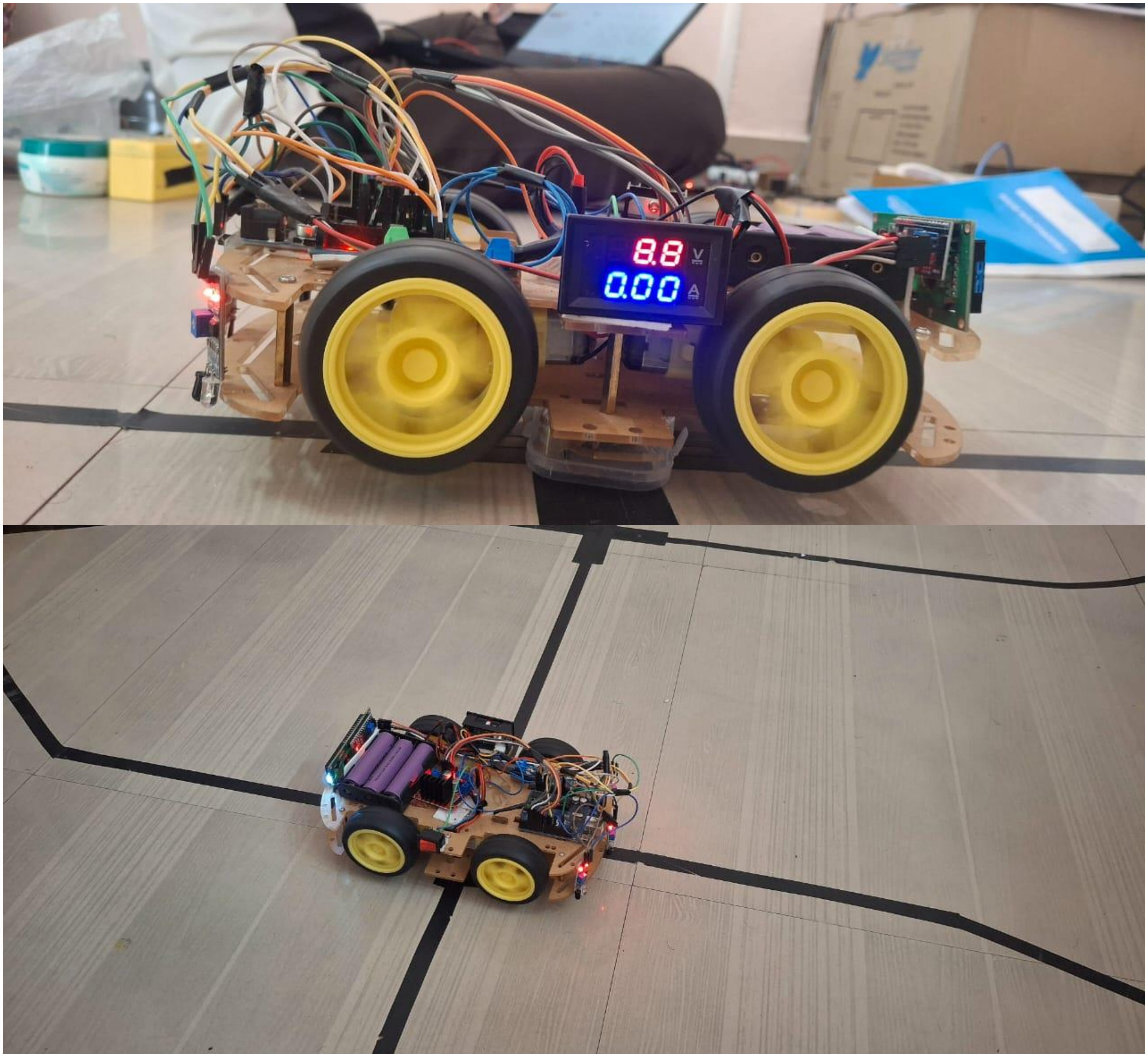

Fig.5.2.1 Hardware Test Result

Chapter-6

Applications and Advantages

6.1. Real world applications

- Using in Amazon Warehouses to move tools and packages
- Robots are used to carry goods around big storage areas.
- Hospitals using small delivery robots to carry medicines and supplies.
- In car or electronics factories, robots carry parts between workstations.
- Wireless charging helps keep robots working in clean, dust-free environments where plugs might spread germs.
- Airports and Large Malls for Cleaning Robots.
- Some airports use small autonomous carts to move luggage.
- Home Appliance Factories use robots to carry TVs, refrigerators, and small parts.
- Some smart hotels use robots to deliver towels, snacks, or room service.
- Military Equipment Warehouses to manage tools, weapons, or supply stock in secure areas.

6.2. Advantages

- No need to plug in.
- Saves time.
- Works automatically.
- Less wear and tear.
- Cleaner and safer.
- Better for 24/7 use.
- Reduces human work.
- Improves robot lifespan.
- Flexible charging spots.
- No broken plugs or sockets.
- Useful in cold or wet environments.
- Supports nonstop operations.

6.3. Relevant SDG(Sustainable Development Goals)

- **SDG 9 – Industry, Innovation, and Infrastructure:**
  Supports smart warehouse systems and advanced tech like automated charging.

Chapter-7

Conclusion

Wireless charging stations for warehouse bots are a smart and efficient upgrade for modern warehouses. Instead of using cables or plugs, these systems let robots charge just by parking over a charging pad. This means the bots can charge themselves automatically without needing human help. It also reduces wear and tear on cables and makes the whole system safer and easier to manage.

One of the biggest benefits is that robots can charge during short breaks or while waiting for their next task—this is called *opportunity charging*. It helps keep the robots running longer without needing to stop for long charging sessions. Over time, this adds up to better productivity and less downtime.

Overall, using wireless charging in warehouses makes daily operations smoother, reduces maintenance, and supports fully automated systems. As more companies rely on robots to handle goods and materials, wireless charging is quickly becoming an important part of keeping everything running efficiently and reliably.

References

- **INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)**
  An International Open Access, Peer-reviewed, Refereed Journal

➢ **Automatic Wireless Power Hub For Electric Vehicles(EVs)**

www.ijcrt.org

- **© MAY 2023 | IRE Journals | Volume 6 Issue 11 | ISSN: 2456-8880**
  - ➢ **Wireless Electric Vehicle Charging System**

- **https://warehouselogistiek.eu/en/general/wireless-charging-transforms-robotics/?utm_source=perplexity**

- **https://www.iiot-world.com/video/elevating-industrial-robotics-how-wireless-charging-is-overcoming-traditional-limitations/?utm_source=perplexity**