**Robo for IOT-Based Irrigation System**

*Project report submitted in partial fulfilment of the requirements for the course- Embedded System and Internet of Things (23IC002) of*

## Bachelor of Engineering

in

**Computer Science and Engineering**

*Submitted by*

**Parth Chandel**

**(2310991090)**

**Parth Thakur**

**(2310991091)**

**Prashant Sainbhi**

**(2310991098)**

**Sania Khanijo**

**(2310991122)**

**Satvik Lakhanpal**

**(2310991126)**

Under the Supervision of

**Dr. Manish Kumar Singla**

DESIGNATION (Professor/Associate Professor/Assistant Professor)

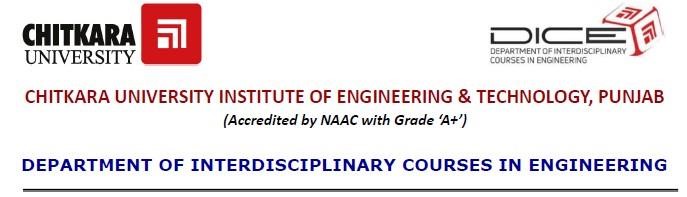


**Chitkara University Institute of Engineering and Technology**

**Department of Interdisciplinary Courses in Engineering (DICE)**

**CHITKARA UNIVERSITY ,PUNJAB**

**NOVEMBER, 2024**



# CERTIFICATE

This is to certify that the project titled “ Robo for IOT-Based Irrigation System”

submitted to the **Chitkara University Institute of Engineering and Technology (CUIET)** by **Parth Chandel (2310991090) , Parth Thakur(2310991091) , Prashant Sainbhi (2310991098) , Sania Khanijo (2310991122) , Satvik Lakhanpal (2310991126)** is a bonafide record of the work done by the students towards partialfulfilment of requirements for the course- Embedded System and Internet of Things (23IC002) of **Bachelor of Engineering in** **Computer Science and Engineering.**

|  |  |
| --- | --- |
|  | **Supervisor/Project Guide** |
| **Signature** |  |
| **Name** |  |
| **Designation** |  |
| **Department** | **DICE** |

Place: Chitkara University, Rajpura, Punjab

Date: 6 November, 2024



# Candidates’ Declaration

We **Parth Chandel (2310991090) , Parth Thakur(2310991091) , Prashant Sainbhi (2310991098) , Sania Khanijo (2310991122) , Satvik Lakhanpal (2310991126)** of **Group- 14A**, B.E. -2023 batch of Chitkara University, Punjab hereby declare that the Embedded System and Internet of Things (ES&IoT) project entitled **“Robo for IOT-Based Irrigation System”** is an original work and data provided in the project report is authentic and to the best of our knowledge. This project has not been submitted by us to any other institute for the award of any other course.

***#Paste here color picture of your project along with students involved***  ***in the project (standing or sitting behind the project)***

**Contribution Details:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Student Name** | **Roll No.** | **Contact Number** | **Contribution in Project** | **Signature** |
| 1. | Parth Chandel | 2310991090 | 8091385199 |  |  |
| 2. | Parth Thakur | 2310991091 | 7018550680 |  |  |
| 3. | Prashant Sainbhi | 2310991098 | 8580709994 |  |  |
| 4. | Sania Khanijo | 2310991122 | 9468090956 |  |  |
| 5. | Satvik Lakhanpal | 2310991126 | 7807493617 |  |  |

# ACKNOWLEDGEMENT

Completion of this project would not have been possible without the help of many people, to whom we are thankful. First, we would like to convey our sincere thanks to **Dr. Rajneesh Talwar**, **Professor and Dean**, DICE-CUIET, Chitkara University, Punjab, for his constant support, motivation and encouragement towards projects and proving healthy environment for working on real time problems.

We express our sincere gratitude to **Dr. Vikas Solanki, Dean (I-Beta),** CSE, for his constant support and cooperation.

We would like to express our sincere gratitude to our supervisor**, Dr. Manish Kumar Singla.** His constant motivation, guidance and support helped us a great deal to achieve this feat.

We would like to thank **Dr. Danvir Mandal, Professor,** Course Coordinator ES&IOT, DICE, for guiding and inspiring us in many ways. We are also thankful to other faculty members and staff of the Department of Interdisciplinary Courses in Engineering and Computer Science and Engineering department for their support.

We also thank our staff **Ms. Navneet Kaur** and **Mr. Chirag Mongia,** who helped us in lab experiments throughout the semester.

We wish a deep sense of gratitude and heartfelt thanks to the management for providing excellent lab facilities and tools.

# PROBLEM STATEMENT

The IoT-Based Irrigation System is designed to revolutionize water management in agriculture, addressing the inefficiencies of traditional irrigation methods that often lead to water wastage and increased operational costs. By integrating a network of environmental sensors, the system continuously monitors essential parameters such as soil moisture, temperature, humidity, and rainfall. This real-time data enables precise assessment of crop water needs, allowing for timely and efficient irrigation.

At the heart of the system is an automated irrigation control mechanism that activates irrigation based on sensor readings, ensuring that crops receive the optimal amount of water without requiring manual intervention. This automation not only saves time for farmers but also significantly reduces water wastage, promoting sustainable agricultural practices.

To enhance user experience, the system includes a user-friendly mobile or web application that allows farmers to monitor and control their irrigation systems remotely. Through this application, farmers can access real-time data, receive notifications about irrigation schedules, and view historical analytics to make informed decisions regarding water usage. Alerts for system malfunctions or extreme weather conditions further empower farmers to respond promptly to any issues.

Data analytics plays a crucial role in optimizing irrigation strategies by analyzing trends in water usage and crop health. By incorporating weather forecasts, the system can proactively adjust irrigation schedules, enhancing water efficiency.

Ultimately, this IoT-based approach not only improves crop yields and quality but also contributes to food security and the conservation of vital water resources, addressing the challenges of water scarcity in modern agriculture.

# 

# ABSTRACT

Rising demands for higher agricultural productivity and the urgent need for water conservation have driven significant advancements in modern irrigation systems. This paper discusses an IoT-based solution that includes a robotic vehicle, a centralized water reservoir, and a flexible conduit network equipped with precision fountain fittings. By using real-time data analytics and autonomous operation, the system aims to improve water distribution efficiency and maximize crop yields in various agricultural settings.

This solution is particularly applicable to:

1. Large-Scale Agriculture: Enhancing efficiency and reducing water waste in extensive farming.

2. Sustainable Agriculture: Supporting environmentally friendly practices for long-term productivity.

3. Greenhouses and Controlled Environments: Ensuring optimal, precise irrigation for controlled growth conditions.

4. Water-Scarce Regions: Using water resources more effectively to meet the challenges of limited supply.

Key development goals include:

1. Enhanced Efficiency: Optimizing water usage to ensure each drop contributes to plant growth.

2. Reliability and Durability: Building robust systems capable of operating in diverse conditions.

3. User-Friendly Interface: Simplifying system management for ease of use.

4. Resource Conservation: Reducing water and energy consumption.

5. Cost-Effectiveness: Maintaining affordability with components like Arduino IDE, C++ programming, and ThinkSpeak server for data monitoring and analysis.

# TABLE OF CONTENTS

**Page No**

**Certificate i**

**Candidates’ Declaration ii**

**Acknowledgement iii**

**Problem statement iv**

**Abstract v**

**List of Figures vi**

**Table of Contents vii**

1. **INTRODUCTION 1**
2. **RELATED WORK 3**
3. **HARDWARE & SOFTWARE 5**
4. **PROJECT DETAILS/METHODOLOGY 9**
5. **RESULTS 15**

**References**

**Appendix A**

**Source Code**

**CHAPTER 1**

### INTRODUCTION

In the modern era, agriculture faces numerous challenges, including water scarcity, inefficient resource management, and the need for sustainable practices. Traditional irrigation methods often fall short in addressing these issues, leading to overuse or underuse of water resources. To tackle these challenges, we introduce the Robo-Based IoT Irrigation System—a groundbreaking solution designed to revolutionize how we manage agricultural irrigation .Our system leverages the power of the Internet of Things (IoT) combined with advanced robotics to provide a smart, efficient, and automated approach to irrigation. At its core, this system integrates sensors, data analytics, and robotic automation to optimize water usage, reduce waste.

#### Problem Identification

An IoT-based irrigation system with robotic integration faces several potential issues that need addressing for effective operation. Key challenges include connectivity problems due to unstable wireless communication, which can disrupt data flow. Limited battery life and power management can impact prolonged functionality. Environmental factors like extreme weather may hinder the robot's mobility, requiring weather-resistant designs. Sensor inaccuracies or malfunctions could lead to incorrect irrigation, emphasizing the need for regular calibration. Navigation across rough terrain poses mobility challenges that need sophisticated path-planning and obstacle-avoidance technology. Data management, security, and integration complexities may affect efficiency, demanding robust computing and encryption. Maintenance issues due to exposure to moisture and chemicals can increase downtime, while high initial costs could hinder scalability. To ensure user adaptability, simplified interfaces and training programs are essential. Addressing these concerns is vital for a reliable, efficient, and scalable irrigation solution.

**Project Solution**

To develop an IoT-based irrigation robot car connected via a Bluetooth module, ensure stable, long-range Bluetooth connectivity for consistent communication. Address power management by integrating rechargeable batteries and energy-efficient components. Design the robot with weather-resistant materials to withstand outdoor conditions. Implement precise water sprinkling using reliable, self-calibrating moisture sensors for optimal irrigation. Navigation challenges on rough terrain can be solved using robust wheels and ultrasonic sensors for obstacle detection. A user-friendly mobile app interface will allow easy control and monitoring. Implement basic data encryption for secure Bluetooth connections. Regular maintenance alerts can be programmed for proactive upkeep, ensuring consistent performance. This approach will create an efficient, user-friendly, and reliable irrigation solution for field watering.

#### Project Objectives

The objective of an IoT-based irrigation system project is to develop an automated, efficient, and reliable solution that optimizes water usage in agricultural fields. This system aims to enhance farming practices by integrating a robotic car capable of sprinkling water, controlled via a Bluetooth module connected to a user’s device. The primary goal is to ensure precise and timely irrigation based on real-time data from soil moisture sensors, reducing water waste and labor costs. The project seeks to provide farmers with an easy-to-use, cost-effective tool that boosts crop productivity while conserving resources. Additionally, it strives to incorporate durable, weather-resistant technology and maintain seamless connectivity for effective remote operation and monitoring.

**CHAPTER 2**

### RELATED WORK

#### Introduction

In the modern era, agriculture faces numerous challenges, including water scarcity, inefficient resource management, and the need for sustainable practices. Traditional irrigation methods often fall short in addressing these issues, leading to overuse or underuse of water resources. To tackle these challenges, we introduce the Robo-Based IoT Irrigation System—a groundbreaking solution designed to revolutionize how we manage agricultural irrigation .Our system leverages the power of the Internet of Things (IoT) combined with advanced robotics to provide a smart, efficient, and automated approach to irrigation. At its core, this system integrates sensors, data analytics, and robotic automation to optimize water usage, reduce waste.

**IoT-Based Pest Detection System**

An IoT-based pest detection system is an innovative solution designed to enhance agricultural productivity by identifying and managing pest issues in real-time. Traditional pest control methods often rely on visual inspections and reactive measures, which can be time-consuming and lead to significant crop damage if pests are detected late. By leveraging IoT technology, this system integrates smart sensors, cameras, and data analytics to continuously monitor crops for signs of pest activity and potential infestations. These devices can detect changes in the environment or pest patterns and send instant alerts to farmers, enabling timely interventions

**Advantages:**

An IoT-based pest detection system offers numerous advantages:

* **Early Detection and Prevention**: Continuous monitoring allows for the early identification of pests, enabling timely intervention before infestations spread and cause significant damage.
* **Reduced Pesticide Use:** Targeted pest control minimizes the need for blanket pesticide applications, leading to more sustainable farming and lower chemical exposure to the environment and consumers.
* **Cost Savings:** Early intervention and precise pest management reduce crop loss and lower overall pest control costs, improving the farmer's profitability.
* **Real-Time Alerts and Monitoring:** Farmers receive immediate notifications via mobile apps or other devices, allowing for quick responses and better management decisions.

**Disadvantages:**

* **High Initial Costs:** The implementation of IoT technology, including sensors, cameras, and communication infrastructure, can involve significant upfront investment, which may be a barrier for some farmers.
* **Connectivity Issues:** The effectiveness of an IoT-based system relies on stable internet or network connectivity. In rural or remote areas with poor connectivity, this can limit functionality.
* **Data Privacy and Security:** The collection and transmission of data raise concerns about privacy and cybersecurity risks, making systems vulnerable to hacking and data breaches.

**Remote Crop Monitoring System**

A Remote Crop Monitoring System is a cutting-edge solution that utilizes Internet of Things (IoT) technology to enhance agricultural efficiency and productivity. As the demand for food continues to rise globally, farmers face the challenge of optimizing crop yields while managing resources effectively. This system integrates various sensors and devices to collect real-time data on environmental factors such as soil moisture, temperature, humidity, and crop health. By transmitting this data to a centralized platform, farmers can monitor their fields remotely via smartphones or computers, allowing for informed decision-making and timely interventions. The system not only helps in detecting issues early, such as water stress or pest infestations, but also facilitates precision farming practices that conserve resources and maximize output

**Advantages:**

* **Real-Time Monitoring:** Farmers can track crop health and environmental conditions in real time, allowing for immediate responses to any issues that arise, such as water stress or disease.
* **Informed Decision-Making:** Access to accurate and timely data empowers farmers to make better decisions regarding irrigation, fertilization, and pest control, leading to optimized crop management.
* **Resource Efficiency:** By monitoring soil moisture and weather conditions, the system helps optimize water and fertilizer use, reducing waste and lowering costs.
* **Increased Yields:** Continuous monitoring allows for early detection of problems, improving crop health and ultimately increasing yields.

**Disadvantages:**

* **High Initial Costs:** Implementing IoT technology can require significant upfront investment in sensors, devices, and communication infrastructure, which may be a barrier for small-scale farmers.
* **Technical Complexity**: The setup, configuration, and maintenance of IoT systems can be complex, requiring technical expertise that some farmers may lack.
* **Connectivity Issues:** Reliable internet connectivity is crucial for the effective functioning of the system. In rural areas with poor network coverage, data transmission may be unreliable.

**Automated Fertilizer Dispenser**

An Automated Fertilizer Dispenser is an innovative agricultural solution designed to optimize the application of fertilizers in farming operations. By leveraging advanced technology and IoT connectivity, this system allows for precise and timely delivery of fertilizers based on real-time soil nutrient levels and crop requirements. Traditional fertilizer application methods often lead to overuse or underuse, resulting in wasted resources, increased costs, and potential harm to the environment. The Automated Fertilizer Dispenser addresses these challenges by utilizing sensors to monitor soil conditions and automated dispensing mechanisms to ensure that crops receive the right amount of nutrients at the right time

**Advantages:**

* **Precise Nutrient Delivery:** The system can accurately measure soil nutrient levels and dispense the exact amount of fertilizer needed, reducing waste and ensuring optimal plant growth.
* **Cost Efficiency:** By minimizing over-application of fertilizers, farmers can significantly reduce input costs while maximizing crop yield and quality.
* **Time Savings:** Automation streamlines the fertilizer application process, allowing farmers to save time and reduce labor requirements, enabling them to focus on other critical farming tasks.

**Disadvantages:**

* **High Initial Costs:** The setup and installation of IoT technology, including sensors and dispensing mechanisms, can involve significant upfront investment, which may be a barrier for some farmers, particularly those operating on a small scale.
* **Technical Complexity:** The system requires technical knowledge for setup, maintenance, and troubleshooting, which may be challenging for farmers without a background in technology.
* **Dependence on Connectivity:** Reliable internet access is crucial for the operation of an IoT-based system. In areas with poor connectivity, the system may face disruptions that could impact its effectiveness.
* **Data Security and Privacy Risks:** The collection and transmission of sensitive agricultural data raise concerns about cybersecurity threats and unauthorized access to the system.

**CHAPTER 3**

### HARDWARE & SOFTWARE

#### Introduction

In this chapter, we will discuss the hardware and software components that make up the

Robo for IOT based irrigation system. The system’s core components include the Water Pump, Soil Moisture Sensor ,Jumper Wires ,Water Container Battery Pack ,BO Motor ,NodeMCUESP32 ,Wheel and Chasis ,L298N and Motor Driver Module . Each component serves a unique role in making the IOT Based ROBO car. This chapter will detail these components' features, functions, and pin configurations and cover the software tools necessary for programming and operation.

**Hardware Components**

#### 1. Soil Moisture Sensor

* **Features**: Soil Moisture Detection: Accurately measures water content in soil to assess moisture levels.
* **Operation**: Sensor Calibration: Calibrates to ensure accurate moisture readings.
* **Pin Diagram**: VCC: Supplies power to the sensor (3.3V or 5V)

D0 (Digital Output): Triggers irrigation systems based on moisture levels.

A0 (Analog Output): Outputs an analog voltage proportional to the soil moisture content.

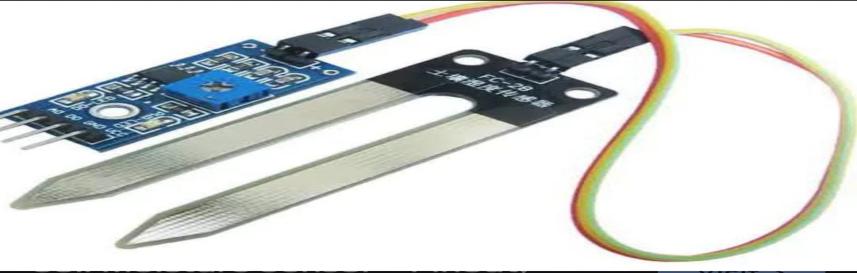


Fig 3.1 Soil Moisture Sensor

#### 2. NodeMCU ESP32

**Features**: Wi-Fi and Bluetooth Connectivity: Enables easy integration with IoT platforms for remote monitoring and control of the irrigation system.

**Operation**: Data Processing and Control: Collects data from connected sensors, processes it, and activates irrigation systems based on moisture levels.

* **Pin Diagram**:
* Power and Ground: VCC provides power (3.3V) while GND connects to ground for circuit completion.
* Digital pin D5: Typically used for controlling the irrigation pump or relay, allowing automated watering based on sensor input.



Fig 3.2 NodeMCU ESP32

#### 3. BO motor

**Features**: Variable Speed Control: Allows precise control over the motor's speed, enabling adjustable water flow rates for irrigation.

* **Operation**: Automatic Activation: The motor is triggered by a microcontroller based on soil moisture sensor readings to provide water when needed.
* **Pin Diagram**:
  + Power Connections: The motor typically connects to a power supply (VCC) and ground (GND) for operation.
  + Digital pin D4: Commonly used to send a digital signal to control the motor driver or relay, activating the motor for irrigation tasks.



Fig 3.3 BO Motor

#### 4 Wheels and chassis

• **Operation**: Maneuverability: The wheels enable the robot car to navigate various terrains and change direction based on motor controls.

**Pin Diagram:** Mounting Points: The chassis includes designated mounting points for securing motors, wheels, and electronic components, ensuring a stable assembly



. Fig 3.4 Wheels and chasis

#### 5. Jumper Wires

Jumper wires play a crucial role in connecting the different components of the system.

**Features:** Flexible Connectivity: Connecting wires allow for versatile connections between components, enabling easy assembly and reconfiguration.

**Operation:** Signal Transmission: Wires carry power and data signals between the microcontroller, sensors, motors, and other electronic components.



Fig 3.5 Jumper Wires

**6. Battery pack**

**Feature:**

Battery Management System (BMS): Monitors and controls the battery’s charge, discharge, and health to ensure safe and efficient operation.

**Operation:**

Charging: The battery goes through two phases—constant current (CC), where the current is steady, and constant voltage (CV), where the voltage is maintained while the current decreases as the battery reaches full charge.



Fig 3.6 Battery pack

**7.L298N Motor Driver Module**

The L298N is a dual H-bridge motor driver IC used to control the direction and speed of DC motors and stepper motors.

**Feature:**

Dual H-Bridge Configuration: Allows independent control of two motors, enabling both forward and reverse motion, as well as speed control through Pulse Width Modulation (PWM).

**Operation**:

Motor Control: The L298N drives motors by switching the polarity of the motor's voltage using the H-bridge. By applying logic signals to its input pins, the direction of the motor can be controlled, and by using PWM on the enable pins, the motor's speed can be adjusted.



Fig 3.7 L298N Motor Driver Module

**8.WATER PUMP**

**Feature:**

Flow Rate Control: Many water pumps have the ability to control the flow rate, allowing for adjustable water output depending on the application, such as in irrigation systems, aquariums, or cooling systems.

**Operation:**

Water Movement: The pump works by using a motor to drive an impeller or diaphragm that moves water from one location to another. It creates pressure that forces water through pipes or hoses, either pushing it to a higher elevation or through a filtration system, depending on the design of the pump.

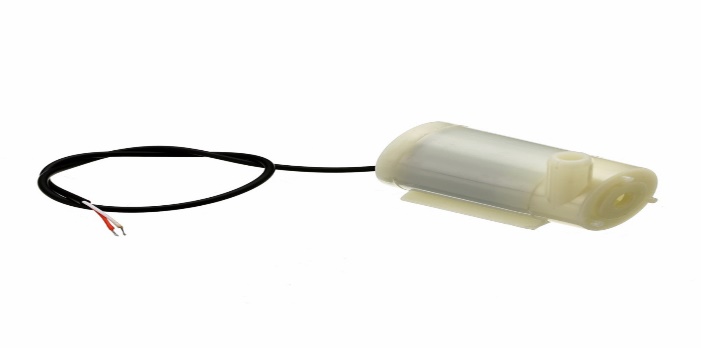


Fig 3.8 Water Pump

**Software Components**

#### WiFi Library (<WiFi.h>)

The WiFi.h library is a fundamental component of the ESP32 platform, enabling seamless wireless connectivity for Internet of Things (IoT) applications. It provides essential functions to connect the ESP32 to WiFi networks, manage network settings, and facilitate data communication. With support for both station and access point modes, the library allows for flexible networking configurations, making it ideal for a wide range of projects that require reliable internet access and connectivity.

**Feature :** The WiFi.h library allows the ESP32 to operate in both station (client) and access point modes. In station mode, the ESP32 connects to an existing WiFi network, while in access point mode, it can create its own network for other devices to join. This versatility is useful for various IoT applications, such as remote monitoring or device control.

**Operation :** The primary operation of the WiFi.h library is to connect the ESP32 to a WiFi network using the WiFi.begin(ssid, password); function. This initiates the connection process, enabling the ESP32 to communicate over the internet, send data to servers, and receive information. This capability is essential for many connected device applications.

#### BluetoothSerial Library (<BluetoothSerial.h>)

The BluetoothSerial.h library is a key component for enabling Bluetooth communication on the ESP32 platform. This library simplifies the process of establishing a Bluetooth Serial connection, allowing the ESP32 to communicate wirelessly with other Bluetooth-enabled devices. It provides essential functions to initialize Bluetooth, send and receive data, and manage connections

**Feature:** Enables easy wireless communication between the ESP32 and other Bluetooth-enabled devices using the Serial protocol.

**Operation:** Initializes Bluetooth communication with serialBT.begin("DeviceName"); and facilitates data transmission through serialBT.print() and serialBT.read() functions.

#### Integration of Hardware and Software

The integration of hardware and software in a robotic car IoT-based water integration system involves a harmonious combination of various components and technologies to create an efficient and automated system for monitoring and managing water resources. Here’s an overview of the key elements involved in this integration:

Hardware Components:

ESP32 Microcontroller: Serves as the central processing unit, handling both WiFi and Bluetooth communication, and controlling other components.

Motors and Motor Drivers: Provide movement capabilities to the robotic car, enabling it to navigate through different terrains.

Soil Moisture Sensor: Monitors soil moisture levels and provides real-time data to determine irrigation needs.

**CHAPTER 4**

### PROJECT DETAILS

#### Introduction

An IoT-based Robo Car in irrigation system is an autonomous vehicle designed to navigate and perform tasks with minimal human intervention, powered by the Internet of Things (IoT) technology. It integrates sensors, actuators, to collect and process real-time data, enabling the car to make decisions based on its environment. By using various IoT components such as GPS, cameras, ultrasonic sensors, and motor controllers, the robo car can navigate roads, avoid obstacles, and interact with external systems. The car communicates with a central server or mobile device for monitoring, remote control, or data collection, offering a glimpse into the future of smart transportation. This technology is not only applicable in autonomous vehicles but can also be used for applications in robotics, delivery systems, and smart city infrastructure, driving the evolution of intelligent, connected devices in everyday life.

#### Circuit Diagram

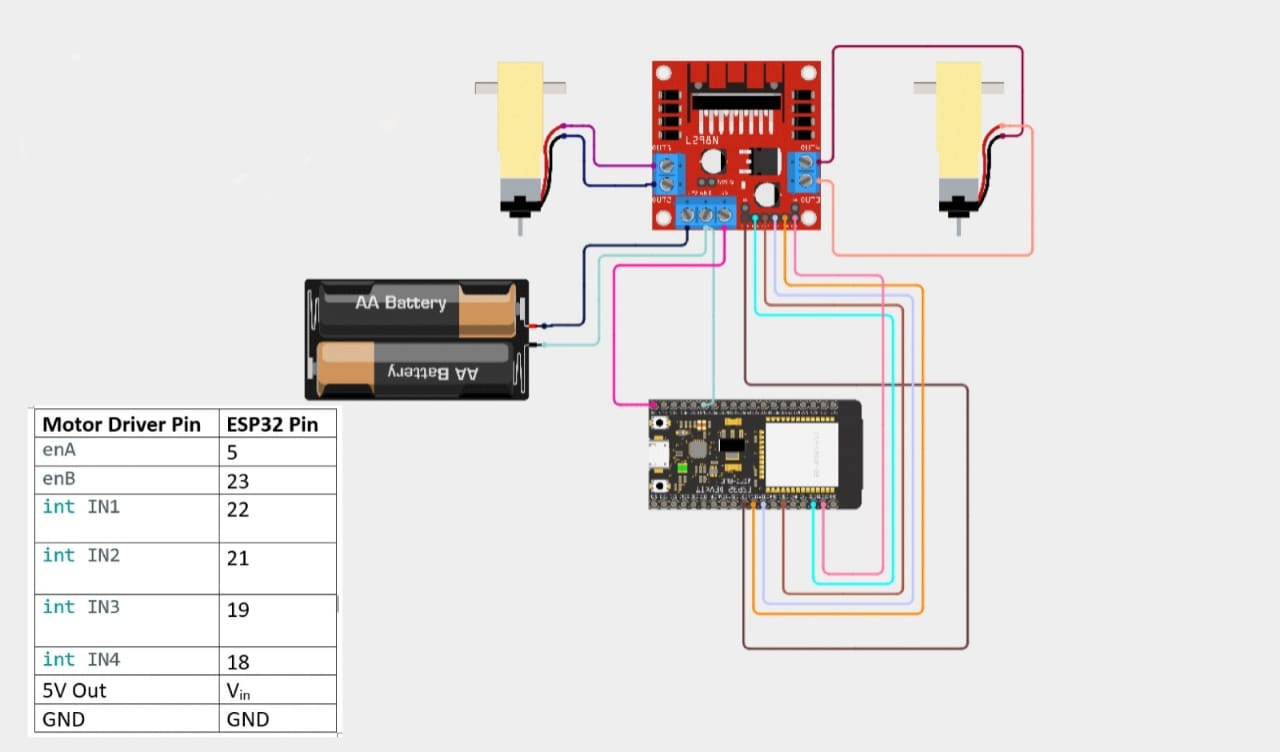


Fig 4.1 Circuit Diagram

**Full Working of the Project**

1. Assemble Hardware: Begin by mounting the DC motors, motor driver module, water pump, soil moisture sensor, ultrasonic sensors, and water tank securely onto the robotic car chassis. Ensure the water pump nozzle is positioned to effectively spray water onto the ground as the robot moves. Connect all components to the microcontroller, securing them in place for stable operation.
2. Connect Circuitry: Wire the motor driver module to the microcontroller’s digital pins for motor control. Connect the relay module, which will control the water pump, and wire the soil moisture sensor to an analog input pin for data collection. Attach the ultrasonic sensors to digital pins for obstacle detection. Verify that the battery pack meets the voltage requirements for all connected devices.
3. Write and Upload Code: Develop and upload code to the microcontroller that manages motor movement, reads data from the soil moisture sensor, activates the water pump when soil moisture falls below a set threshold, and uses ultrasonic sensors to avoid obstacles. Include logic for basic navigation, forward and backward movement, and turning.
4. Integrate IoT Connectivity: Set up Wi-Fi or Bluetooth modules for connecting the robotic car to a mobile app or cloud platform. Implement HTTP or MQTT protocols for real-time data communication, enabling remote monitoring and manual control of the irrigation system from a smartphone or computer.
5. Test and Calibrate: Conduct thorough testing to ensure the robot reads soil moisture levels accurately and waters the plants when needed. Check navigation to ensure it avoids obstacles smoothly and moves efficiently. Fine-tune sensor thresholds and adjust code to improve the robot’s performance for reliable and effective plant irrigation.Software Used

**SOFTWARE USED**

#### WiFi Library (<WiFi.h>)

The \*WiFi library\* in an IoT-based robo car enables wireless communication, allowing remote control and data transfer via WiFi modules like \*ESP8266\* or \*ESP32\*. It facilitates real-time monitoring, command execution, and sensor data retrieval, making the car smarter and easier to control from any connected device.

#### 2.BluetoothSerial Library (<BluetoothSerial.h>)

The \*Bluetooth Serial Library\* allows an IoT-based robo car to communicate wirelessly with devices via Bluetooth, using modules like \*HC-05\* or \*HC-06\*. It enables real-time control, data exchange, and monitoring through a Bluetooth connection, making it ideal for short-range wireless communication with smartphones or computers.

**Methodology Followed to Implement Project**

Implementing an IoT-based Robo Car Water Irrigation System involves integrating a mobile robot with a water pump and sensor system, all connected through IoT technology for real-time monitoring and automation. Here's a step-by-step methodology to implement the project:

1**. Project Planning and Design**

Define Objectives: Determine the purpose of the project, such as automating irrigation based on soil moisture levels, and designing a robo car that can move and water plants.

Component Selection: Choose necessary components like:

- Robo car platform(DC motors, wheels, chassis)

- Water pump(for irrigation)

- Soil moisture sensors

- Microcontroller(e.g., Arduino, ESP32)

- WiFi or Bluetooth module (for remote control and monitoring)

- Power source (battery pack)

2. **Hardware Integration**

- Assemble the Robo Car: Set up the robot chassis with motors, wheels, and a microcontroller to drive the car's movement.

- Water Pump Integration: Connect a water pump to the robo car, which will be activated based on input from soil moisture sensors.

- Sensors Setup: Integrate soil moisture sensors to measure the soil's water content and trigger the pump when the moisture level drops below a certain threshold.

- Power Supply: Ensure adequate power to run the motors, sensors, pump, and the microcontroller. Consider using rechargeable batteries or solar panels for sustainability.

3**. Software Development**

- Control Logic:

- Program the microcontroller (e.g., Arduino or ESP32) to control the car's movement (forward, backward, turns) and water pump based on sensor readings.

5**. Testing and Calibration**

- Test Sensor Accuracy: Calibrate the soil moisture sensors to ensure accurate readings.

- Car Movement and Watering Tests: Test the robot’s ability to move autonomously and water plants based on moisture levels.

- Remote Control Validation: Ensure the IoT communication works properly, with real-time control and monitoring available through the mobile app or web interface.

6. **Optimization and Finalization**

- Power Efficiency: Optimize power consumption to extend battery life by refining movement and watering logic.

- Automation: Automate the watering process based on real-time soil moisture readings, with an option for manual control when needed.

7**. Deployment**

- Deploy the IoT-based robo car in a real-world environment, such as a garden or farm, for automated irrigation.

- Monitor and evaluate its performance, making any adjustments or improvements based on real-time data.

**CHAPTER 5**

### RESULTS

#### Results

An IoT-based robotic car water irrigation system automates the process of watering plants, enhancing efficiency and reducing labor. The system moves autonomously or via remote control, spraying water through an integrated pump while equipped with sensors that monitor soil moisture levels. When connected to a mobile app, users can control and receive real-time updates, ensuring precise irrigation only when needed, which conserves water and prevents overwatering. The system's data collection enables analysis for future optimization. With reliable navigation, consistent water distribution, and effective IoT connectivity, this solution supports sustainable farming, making it valuable for home gardens and small farms alike.

#### 

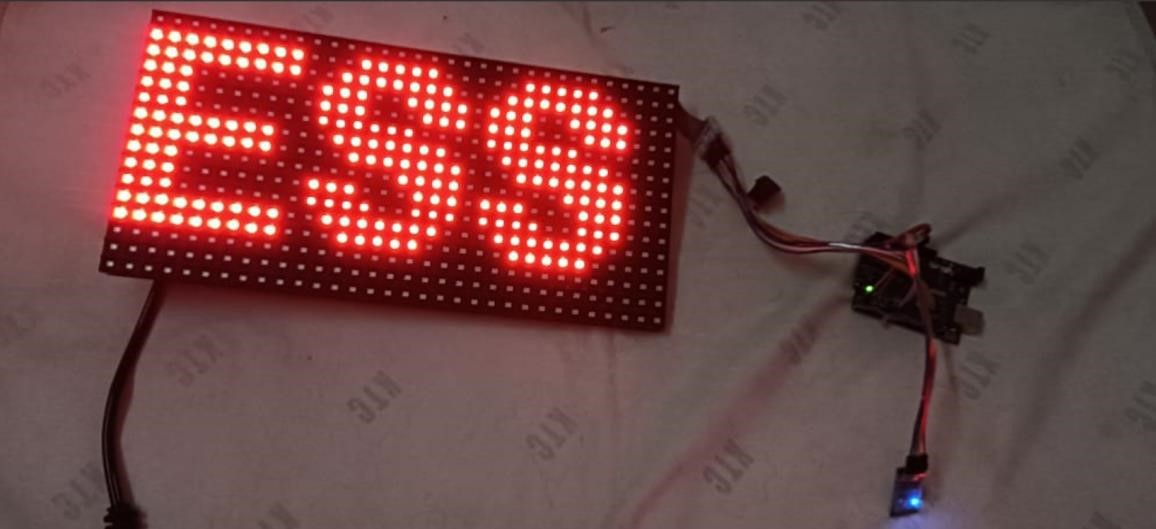


Fig 5.1 Result 1



Fig 5.2 Result 2



Fig 5.2 Result 3

#### Application Areas

An IoT-based water irrigation robotic car can be effectively used in the following application areas:

**Large-Scale Agriculture:** Automates irrigation in extensive fields, enhancing coverage and water distribution, reducing labor, and improving overall efficiency.

**Sustainable Agriculture**: Supports eco-friendly practices by minimizing water waste, using real-time data to optimize water use and reduce environmental impact.

**Greenhouses and Controlled Environments**: Provides precise watering to specific areas, maintaining ideal moisture levels crucial for plant health in controlled settings.

. **Water-Scarce Regions:** Maximizes the use of limited water resources by applying water precisely where needed, conserving water while maintaining crop health.

**Urban and Vertical Farming:** Assists in efficient water management within space- constrained urban farms, enabling smart irrigation in stacked or indoor growing setups.

#### Justification of Results with Respect to Chapter 2 (Related Work)

Your IoT-based water irrigation robotic car stands out from IoT-based pest control and crop monitoring systems due to its direct impact on essential resources and immediate benefits. While pest control prevents damage and crop monitoring provides data insights, your system actively conserves water through precise, autonomous delivery, supporting sustainability and addressing water scarcity. This immediate and tangible benefit boosts crop growth and yield, offering a compelling advantage. To improve, consider enhancing the system with AI-driven decision-making for optimal watering schedules based on weather forecasts and real-time soil data. Integrating multi-functional capabilities, like nutrient sensors or a pest detection module, could make it a comprehensive field management tool. Adding solar power for energy efficiency and a user-friendly mobile app for remote control and monitoring would elevate its appeal. These improvements position your project as an advanced, multifunctional, and eco-friendly solution for modern agriculture.

**Advantages**

**Precise Nutrient Delivery:** The system can accurately measure soil nutrient levels and dispense the exact amount of fertilizer needed, reducing waste and ensuring optimal plant growth.

**Cost Efficiency:** By minimizing over-application of fertilizers, farmers can significantly reduce input costs while maximizing crop yield and quality.

**Time Savings:** Automation streamlines the fertilizer application process, allowing farmers to save time and reduce labor requirements, enabling them to focus on other critical farming tasks.

**Disadvantages**

**High Initial Costs:** The setup and installation of IoT technology, including sensors and dispensing mechanisms, can involve significant upfront investment, which may be a barrier for some farmers, particularly those operating on a small scale

**Dependence on Connectivity:** Reliable internet access is crucial for the operation of an IoT-based system. In areas with poor connectivity, the system may face disruptions that could impact its effectiveness.

#### REFERENCES

1. C.-C. Cheng and C.-C. Lu, “Robotic Arm Control Based on Internet of Things,” 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)., Aug 2017.
2. <https://circuitdigest.com/home-automation-projects>
3. [https://www.electronicsforu.com/electronics-projects/15-awesomeautomationprojects](https://www.electronicsforu.com/electronics-projects/15-awesome-automation-projects)
4. https://iotdesignpro.com/projects

**APPENDIX A**

**Source Code of the Project :**

#include <WiFi.h>

#include "ThingSpeak.h"

#include "BluetoothSerial.h"

#include <ESP32Servo.h>

BluetoothSerial serialBT;

Servo sensorServo;

const char\* ssid = "your\_SSID";

const char\* password = "your\_PASSWORD";

unsigned long myChannelNumber = 2728424;

const char \* myWriteAPIKey = "YOUR\_API\_KEY";

WiFiClient client;

#define IN1 22

#define IN2 21

#define IN3 19

#define IN4 18

#define SOIL\_SENSOR\_PIN 34

#define PUMP\_PIN 25

int moistureThreshold = 400;

unsigned long previousMillis = 0;

const long interval = 15000; // 15 seconds

void setup() {

Serial.begin(115200);

serialBT.begin("Parth's RC Car");

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

}

ThingSpeak.begin(client);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(PUMP\_PIN, OUTPUT);

digitalWrite(PUMP\_PIN, LOW);

sensorServo.attach(26);

}

void loop() {

if (serialBT.available()) {

moveCar(serialBT.read());

}

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

int soilMoistureValue = readSoilMoisture();

if (WiFi.status() == WL\_CONNECTED) {

ThingSpeak.setField(1, soilMoistureValue);

ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

}

if (soilMoistureValue < moistureThreshold) {

digitalWrite(PUMP\_PIN, HIGH);

} else {

digitalWrite(PUMP\_PIN, LOW);

}

}

}

void moveCar(char direction) {

switch (direction) {

case 'F':

digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);

break;

case 'B':

digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);

break;

case 'S':

digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);

break;

}

}

int readSoilMoisture() {

sensorServo.write(90);

delay(1000);

int soilMoistureValue = analogRead(SOIL\_SENSOR\_PIN);

sensorServo.write(0);

return soilMoistureValue;

}