CC4003NI Introduction to Robotics and IoT

Modernizing Agriculture with IoT-Based Monitoring Systems

50% Group Coursework

Academic Semester: AY 2024/25

Credit: 30 credit yearlong module

Group Name:			
S.N.	**Student Name**	**College ID**	**London Met ID**
1	Bikram Tamang AI5 (Leader)	NP01AI4A240095	24046576
2	Sudip Giri - AI5	NP01AI4A240146	24046651
3	Abiral Ghimire - AI5	NP01AI4A240125	24046557
4	Yakendra Bam Rajawar - AI5	NP01AI4A240127	24046664
5	Chandani Banjade - AI5	NP01AI4A240131	24046578

Assignment Due Date: 12th May 2025
Assignment Submission Date: 12th May 2025
Submitted To: Shishir Subedi

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Turnitin Report:

20% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **95** Not Cited or Quoted 20%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
7% Publications
20% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Abstract

The Automated Irrigation System monitors soil moisture, water levels, and climate conditions to ensure efficient plant care. It uses a DHT11 sensor for temperature and humidity, a soil moisture sensor for irrigation control, and an ultrasonic sensor to measure water levels, triggering alerts when water is low.

The system displays real-time data on a 16x2 LCD screen, with modes for soil moisture, water level, and temperature & humidity. Users can switch between modes via serial input, and the display shows alerts for critical conditions like low water levels.

A relay controls a water pump, activating it when soil moisture drops below a threshold. The system also includes scheduled watering intervals, automatically hydrating plants. A buzzer provides additional alerts, ensuring proper plant care and efficient water use.

Table of Contents

1. Introduction

The Internet of Things (IoT) is transforming how we interact with technology by connecting everyday devices—such as soil sensors and irrigation systems—to the internet. These devices communicate and share real-time data, functioning like a farm's nervous system. Equipped with sensors and software, IoT tools continuously monitor environmental conditions and automate adjustments. In agriculture, this shift enables precision farming, replacing guesswork with data-driven decisions. Farmers can now use smartphones to identify fields needing water, receive temperature alerts, or automate irrigation based on soil moisture levels. Beyond convenience, IoT optimizes resource use, increases crop yields, reduces waste, and helps farmers adapt to climate challenges. For small-scale farmers, IoT solutions address labor shortages and resource limitations, paving the way for sustainable and profitable practices that benefit both livelihoods and the environment.

1.1. Current Scenario of Agriculture in Nepal

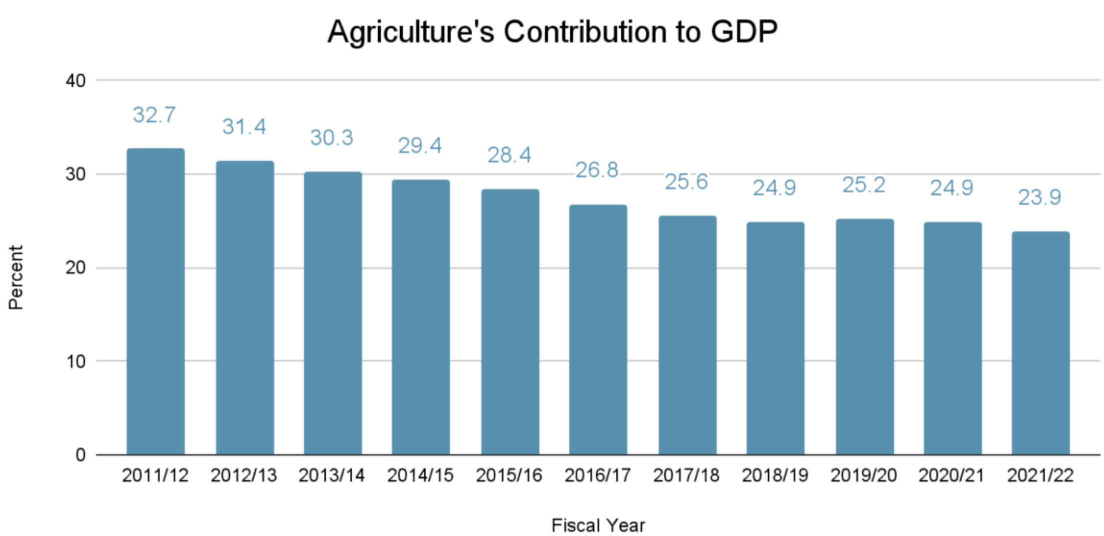


Fig: Agriculture's Contribution to GDP in Nepal

Agriculture contributes 25% to Nepal's GDP and sustains 65% of its population. However, the sector faces systemic challenges. Most farms are small and fragmented, hindering adoption of modern techniques or economies of scale. Heavy reliance on monsoon rains leaves farmers vulnerable to climate extremes like droughts or floods. Limited access to advanced technology, irrigation infrastructure, and financing further stifles productivity, particularly in rural areas. As a result, Nepal's agricultural output ranks among the lowest in South Asia, struggling to meet domestic demand. These issues underscore the urgent need for innovations like IoT-based smart farming to enhance efficiency, build climate resilience, and secure rural livelihoods.

1.2. Current Scenario of Agriculture Globally

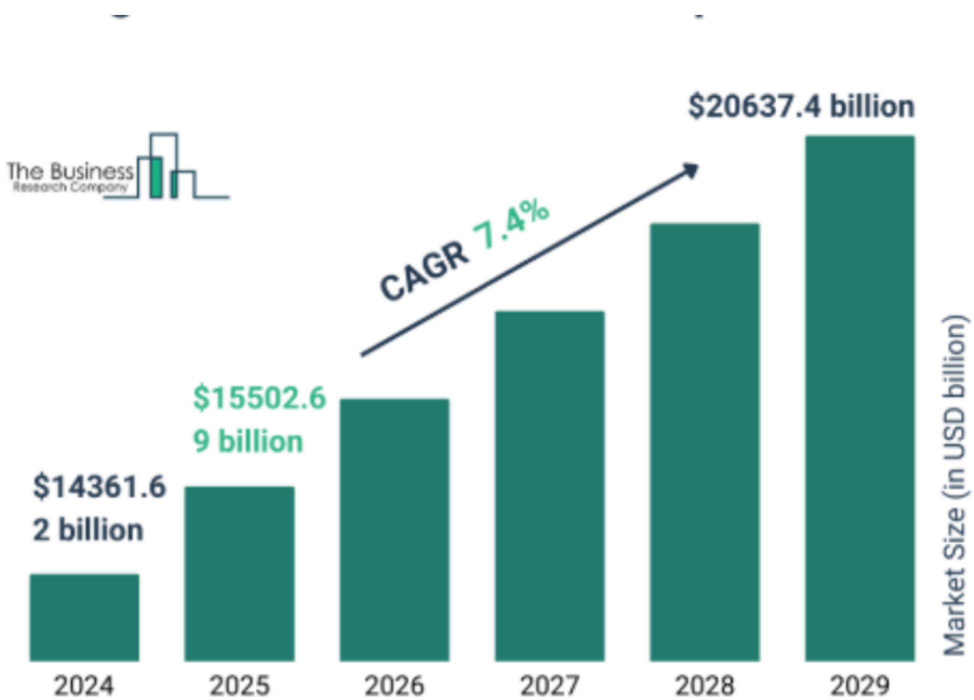


Fig: Chart Demonstrating Current Global Scenario

Global agriculture faces unprecedented pressure. By 2050, food production must rise by 70% to feed a projected population of 9.7 billion. Traditional farming methods are inadequate amid escalating climate disruptions—droughts, floods, and heatwaves—that threaten crop yields and food security. Simultaneously, shrinking water resources, arable land scarcity, and environmental degradation (e.g., deforestation, emissions) demand sustainable solutions. IoT-driven smart farming is emerging as a critical tool, enabling real-time data integration and automation. By optimizing water use, boosting yields, and enhancing climate resilience, IoT technologies empower farmers worldwide to produce more with fewer resources, ensuring a sustainable agricultural future.

1.3. Problem Statement and Project as a Solution

Nepal’s farmers grapple with unpredictable weather, water scarcity, and outdated practices. Sudden temperature shifts damage sensitive crops like tomatoes and lettuce, while inefficient irrigation wastes water and inflates costs. Manual monitoring of soil and weather conditions is time-consuming and unreliable, compounded by limited access to water and electricity.

To address these challenges, this project proposes an IoT-based smart farming system. Soil moisture, temperature, and humidity sensors automate irrigation, delivering precise water amounts even during weather extremes. An ultrasonic sensor tracks water levels in storage tanks, preventing shortages. Real-time data is relayed to farmers via the Blynk app, enabling prompt responses to alerts. By automating irrigation and monitoring, this system reduces manual labor, conserves resources, and improves crop health. Ultimately, it empowers farmers to adopt climate-resilient, sustainable practices tailored to Nepal’s unique challenges.

1.3. Aim and Objectives

Aim

The aim of this project is to design and implement an IoT-based smart irrigation system that automates agricultural processes, optimizes resource usage, and enhances crop resilience to climate variability, specifically tailored for Nepal’s farming challenges.

Objectives

- **Automate Irrigation**: Utilize soil moisture sensors to trigger irrigation only when necessary, reducing water wastage and ensuring optimal soil hydration.
- **Adapt to Climate Extremes**: Integrate temperature and humidity sensors (DHT11) to dynamically adjust watering schedules during unexpected heatwaves, cold spells, or droughts.
- **Monitor Water Efficiency**: Track water levels in storage tanks using ultrasonic sensors to prevent shortages and enable proactive resource management.

- **Enable Remote Monitoring**: Provide farmers with real-time data via the Blynk app, allowing instant alerts for critical conditions (e.g., low water levels, soil dryness).
- **Reduce Manual Labor**: Replace traditional, labor-ifntensive irrigation methods with automated systems to minimize human intervention and operational costs.
- **Ensure Scalability**: Design a modular system that can be expanded from small gardens to large farms, accommodating diverse agricultural needs.

2. Background

2.1. System Overview

The Automated Irrigation System monitors soil moisture, water levels, and climate conditions to ensure efficient plant care. It uses a DHT11 sensor for temperature and humidity, a soil moisture sensor for irrigation control, and an ultrasonic sensor to measure water levels, triggering alerts when water is low.

The system displays real-time data on a 16x2 LCD screen, with modes for soil moisture, water level, and temperature & humidity. Users can switch between modes via serial input, and the display shows alerts for critical conditions like low water levels.

A relay controls a water pump, activating it when soil moisture drops below a threshold. The system also includes scheduled watering intervals, automatically hydrating plants. A buzzer provides additional alerts, ensuring proper plant care and efficient water use.

2.2. Design Diagrams

2.2.1. Hardware Architecture

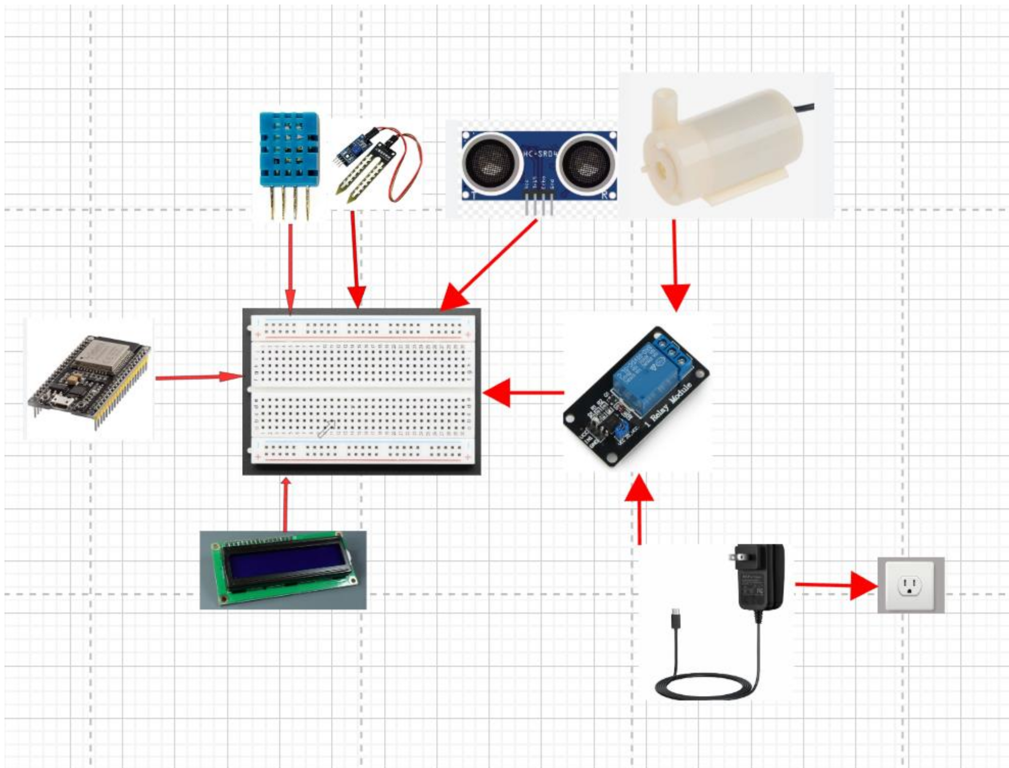


Fig: Hardware Architecture of the System

2.2.2. Circuit Diagram

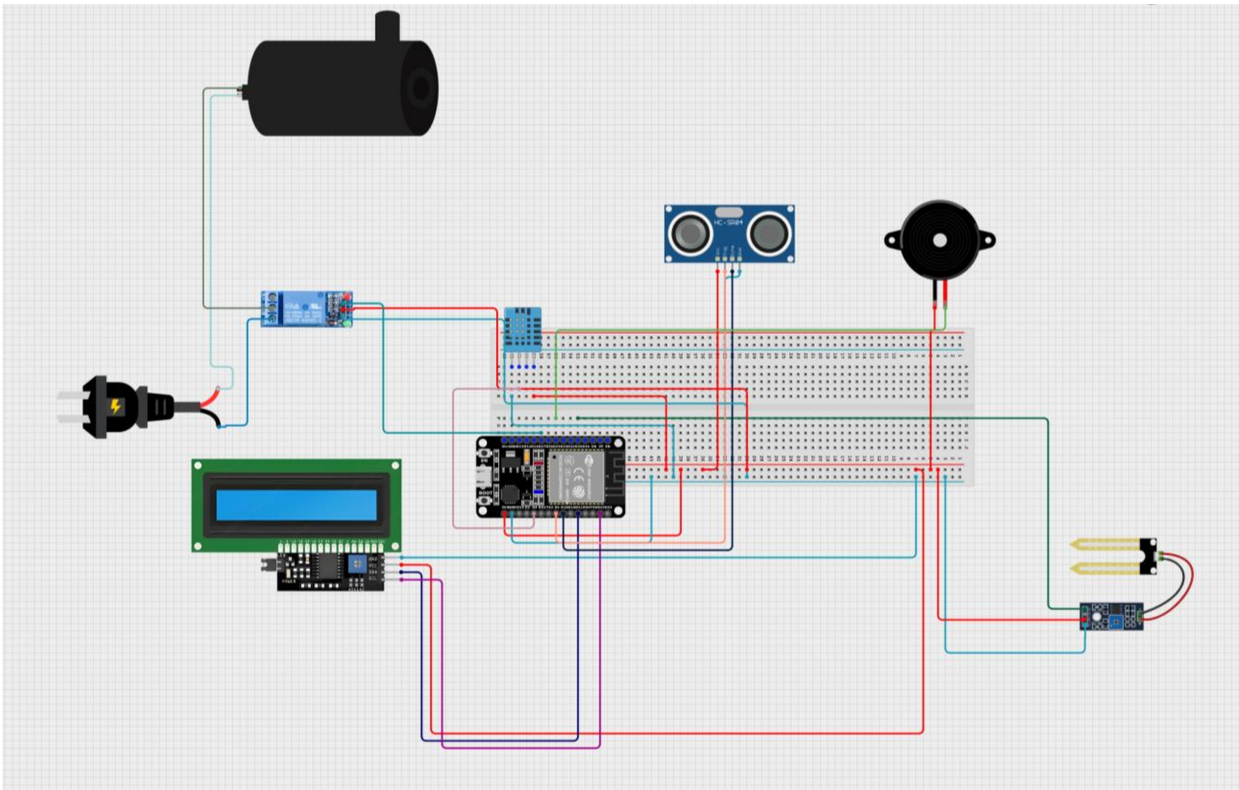


Fig: Circuit Diagram of the System

2.2.3. Flowchart

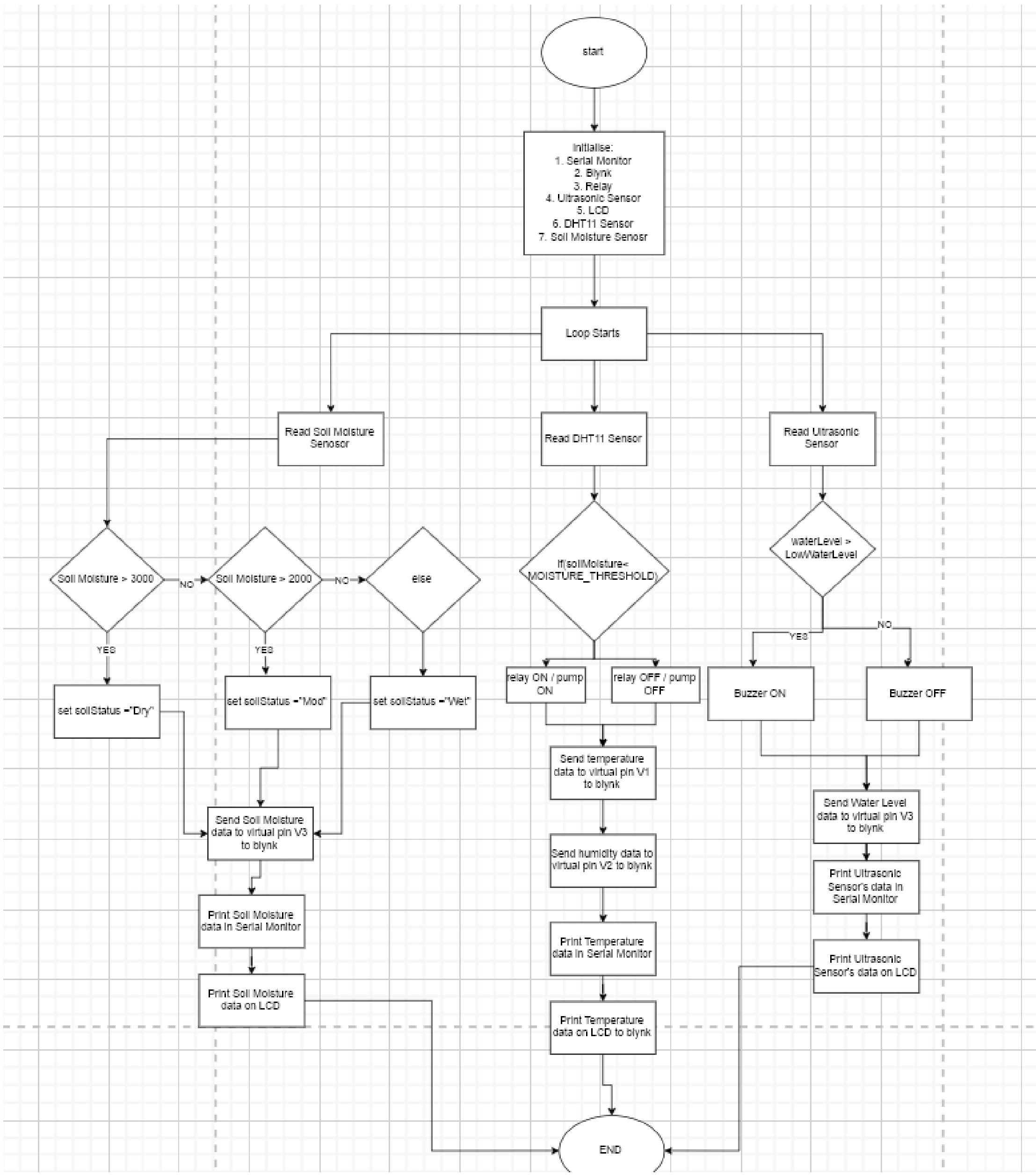


Fig: Flowchart of the System

2.3. Requirement Analysis

2.3.1. Hardware Requirements

- **ESP32 Development Board**

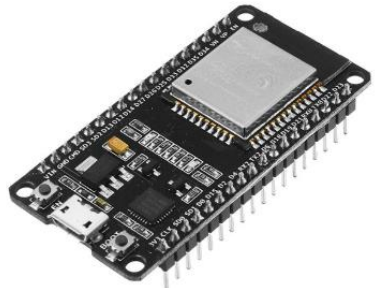


Fig: esp32 development board

The ESP32 is a microcontroller that we are going to use in this project. It has built in Wi-fi and Bluetooth for convenience. We must program this microcontroller according to the project to get the desired output i.e. for smart irrigation and management of water waste.

- **Ultrasonic Sensor (HC-SR04)**

Fig: Ultrasonic Sensor (HC-SR04)

We used the HC-SR04 Ultrasonic Sensor for calculation the distance. Talking about the use case, it is used for water level monitoring in the tanks or water system.

- **DHT 11 Temperature and Humidity Sensor**

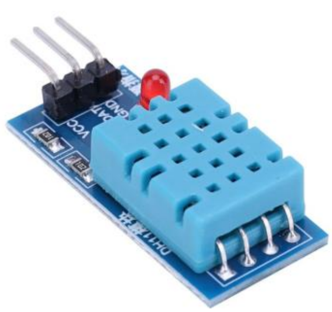


Fig: DHT11 temperature and humidity sensor

The DHT11 Temperature and Humidity Sensor, as the name suggests, it is used for real time monitoring of temperature and humidity and adjust irrigation accordingly.

- **Soil Moisture Sensor**

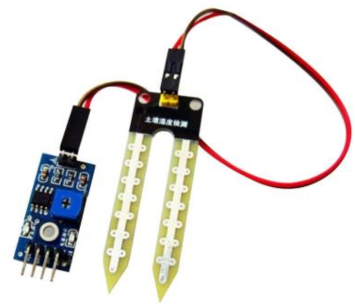


Fig: Soil Moisture Sensor

Soil Moisture Sensor, as the name implies, is a moisture monitoring sensor placed in a determined area. In other words, it is used to check the water content in the soil.

- **Buzzer**

Fig: Buzzer

For this project, a buzzer is used to give alerts to the user, the errors can be ranging from water level alerts, system malfunctions, soil moisture warnings etc.

- **Water Pump**

Fig: Water Pump

A water pump is a device used to fetch water from the sources of water like tanks. In this project water pump serves an important role for the given reason i.e. to fetch the water from the sources to the designated area.

- **LCD Display (I2C)**

Fig: LCD Display (I2C)

An LCD display (I2C) is a liquid crystal display which is used for warnings, alert system and real time monitoring of the important stuffs such as water level, moisture level in the soil, etc.

- **Jumper Wires**

Fig: Jumper Wires

A jumper wire is used to connect various electronic stuffs like microcontroller, power source, sensors and actuators together.

- **Breadboard**

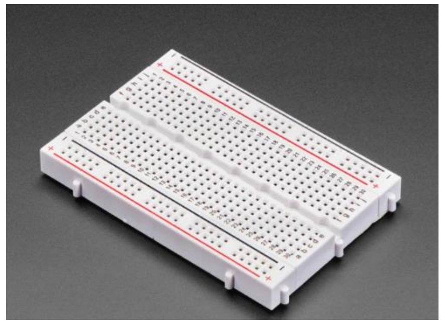


Fig: Breadboard

A breadboard is used in this project for the effective wire management and for convenience as it provides a systematic way for wire management and for the power sources like +ve and –ve.

- **5V Relay Module**

Fig: 5V Relay Module

A 5V relay module is used in this project for the automation of water pump which automatically turns the water pump on/ off based on the water moisture levels.

- **5V 2A DC Adapter**

Fig: 5V 2A DC Adapter

A 5V 2A DC Adapter is used in this project for the conversion of AC current to the DC current which can be beneficial for the power source of both the microcontroller, sensors and actuators.

2.3.2. Software Components Used:

- **Arduino IDE for Programming the ESP32**

Fig: Arduino IDE

The Arduino IDE is the software platform used to write, compile, and upload programs (called sketches) to Arduino microcontrollers and various devices like ESP32, ESP8266, and others. The IDE simplifies the process of programming hardware by providing a user-friendly interface for code development, compiling, and uploading.

- **Blynk App**

Fig: Blynk App

The Blynk app is a popular platform used to build mobile applications for controlling IoT devices over the internet. It allows you to create custom interfaces for controlling devices like microcontrollers (Arduino, ESP32, etc.), sensors, motors, lights, and more. The app makes it easy to build IoT projects by providing a graphical interface without needing extensive programming knowledge. Blynk uses a cloud server to manage communication between the mobile app and your IoT device, allowing remote control from anywhere with an internet connection.

- **Fritzing for designing the Circuit Diagram**

Fig: Fritzing

Fritzing is an open-source electronics design software used to create circuit diagrams, PCB layouts, and prototypes. It allows users to visually design and document electronics projects, making it especially popular for beginners, hobbyists, and makers. With its easy-to-use interface, Fritzing simplifies the process of designing circuits and sharing projects. Fritzing can also be used to create PCB layouts (Printed Circuit

Board). It allows you to place components on a PCB, route the connections, and generate files for manufacturing your custom PCB.

Fritzing is an excellent tool for anyone looking to design and document electronics projects. It’s particularly beginner-friendly because of its visual, intuitive approach to circuit design. Whether you're working on simple breadboard layouts or designing custom PCBs, Fritzing makes it easier to create, share, and build your electronics projects.

3. Case Study

3.1. Introduction

This case study focuses on the use of IoT in agriculture, aiming to develop an algorithm that ensures optimal soil moisture levels through a real-time monitoring system, while also enhancing user engagement. This approach not only helps increase agricultural yields but also minimizes water consumption. Extensive research has been conducted to improve both the quality and quantity of agricultural outputs. The implementation of this intelligent system addresses critical challenges in farming. In this study, a smart agriculture system is presented, enabling real-time monitoring and management of soil moisture levels.

3.2. Problem Statement

Furthermore, farmers often lack access to real-time data about their crops' needs, which can hinder decision-making. The system continuously monitors soil moisture, temperature, and humidity, providing farmers with valuable insights to make more informed decisions. The system also helps improve agricultural yield by ensuring optimal plant care, leading to healthier plants and better-quality crops. In regions facing water scarcity, the system helps reduce water consumption, making it more sustainable and efficient. Finally, the system addresses the problem of scaling irrigation systems for larger farms by offering a scalable solution that can be implemented across large areas, improving efficiency without manual intervention. Overall, this smart watering system promotes sustainable farming practices, enhances resource management, and increases productivity while conserving water.

3.3. Background

Smart agriculture integrates both hardware and software to facilitate autonomous system operation. The YL-69 and DHT11 sensors are utilized to measure humidity and temperature, respectively. This data is transmitted to the ESP32 microcontroller for further processing, allowing the condition of the plant to be displayed on an LCD. The system automatically waters the plant based on the collected data, and the user can control water irrigation via the Blynk app. Additionally, the data is continuously revised in the Blynk app. To ensure smooth interaction between sensors, microcontroller, actuators and cloud platform IDE is used to run the necessary code.

3.4. Result

Two identical pots of chili plants were placed in the same location for testing. One pot was equipped with the intelligent watering system, while the other was not. As shown in Fig. 4, the chili plant on the left was tested with the system, and the plant on the right was tested without it.

3.5. Improvements and Future Plans

To improve the smart watering system, additional sensors like NPK, rain, and pH sensors could be integrated to monitor soil nutrients and environmental factors, allowing for more precise watering and fertilization. The system could also be made more energy-efficient by using low-power sensors or solar panels. Incorporating data logging, predictive analytics, and machine learning would help optimize watering schedules and plant care based on historical trends and environmental conditions. Enhancing the user interface of the Blynk app and integrating weather forecasts could further optimize the system's performance. Additionally, focusing on scalability for larger farms, reducing component costs, and adding real-time alerts for system maintenance would make the system more accessible, reliable, and sustainable for diverse agricultural applications.

3.6. Conclusion

This study presents a real time water monitoring system using soil moisture sensor, humidity and temperature sensor, and IoT connectivity. Data from these sensors is transmitted to an ESP32 microcontroller to determine if watering is needed. The system also offers FreeRTOS software for real-time operation and the Blynk app for simplified monitoring. The experiment, conducted with two chili plants, showed significant improvements in plant growth and yield when using the smart watering system. The plant with the system produced more chilies and healthier leaves, while the plant without it had wilted leaves and low production. This system helps reduce water wastage, and future research could incorporate additional sensors like NPK, rain, and pH sensors for more precise monitoring and action.

4. Development

4.1. Planning and Design

- **Determine User Requirements**
  The needs and requirements of the user was determined, and this system was developed with the vision of efficient, automated, low power consumption, high efficiency, low cost and low maintenance agricultural system that allows the user for smart decisions and less manpower for the maintenance of a single plant to the farm as a whole
- **Determine Scope of the System**
  The scope of the system includes highly accurate, automated and low manpower system allowing the user for easier farming of plants both seasonal and non-seasonal ranging from plants, fruits to flowers.
- **Configure the User Interface**
  The User Interface was developed for easier control and interaction with the system.
- **Testing**
  Testing was done for any type of errors and problems either in the code or in the hardware sections ensuring no flaws in sensors, actuators, and code.

4.2. Resource Collection

To complete this project various components were utilized. The list is given below:

- **Sensors**
  - DHT11 Sensor
  - HC-SR04 Sensor
  - Soil Moisture Sensor

- **Actuators**
  - Buzzer
  - Relay Module
  - Water Pump
- **Display Device**
  - 0x27 I2C LCD Display
- **Adapter**
  - AC/DC 5V Adapter

4.3. System Development

- **Phase 1: Connecting Breadboard and LCD**

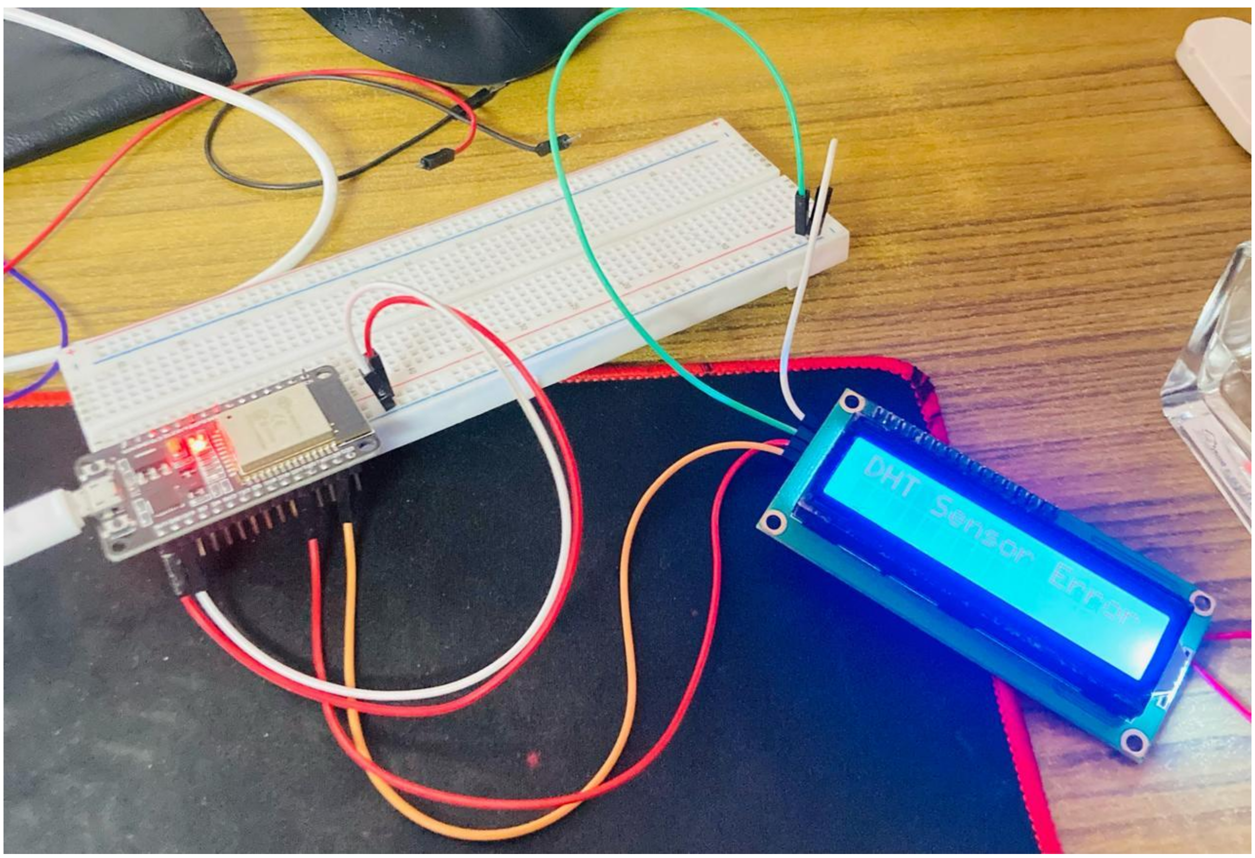


Fig: Connecting ESP32, Breadboard and LCD

The ESP32 is connected to a 16x2 LCD display using the I2C interface, where GPIO21 is used for SDA and GPIO22 for SCL. The LCD gets power from the 3.3V or 5V pin of the ESP32 and displays sensor readings like soil moisture, temperature, humidity, and system status.

- **Phase 2: Connecting Sensors: DHT11 Sensor, Soil Moisture Sensor and Ultrasonic HC-SR04 Sensor**

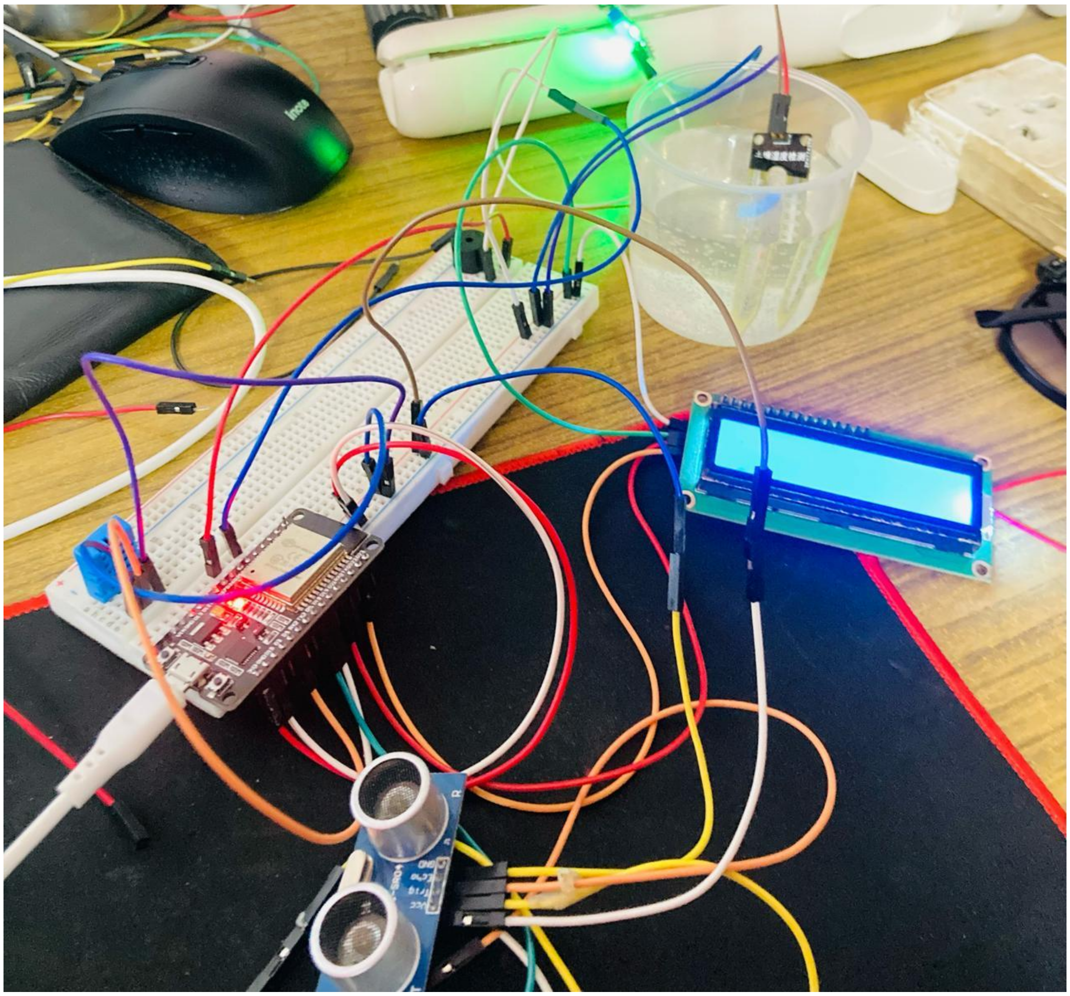


Fig: Connecting DHT11 Sensor, Soil Moisture Sensor and Ultrasonic Sensor

The ESP32 is connected to the breadboard and all the sensors are joined together through pins. For DHT11 Sensor, DHT11 Sensor VDD is connected to positive breadboard pin, DHT11 Sensor GND is connected to negative, DHT11 Sensor NULL is connected to none and DHT11 Sensor DATA is connected to digital pin D4 in ESP32.

For Soil Moisture Sensor, Soil Moisture Sensor VCC is connected to positive breadboard pin, Soil Moisture Sensor GND is connected to negative breadboard pin, Soil Moisture Sensor AO is connected to digital pin D35 and Soil Moisture Sensor DO is connected to none.

For Ultrasonic Sensor, Ultrasonic Sensor HC-SR04 VCC is connected to positive breadboard pin, Ultrasonic Sensor HC-SR04 GND is connected to negative breadboard pin, Ultrasonic Sensor HC-SR04 TRIG is connected to digital pin D5 and Ultrasonic Sensor HC-SR04 ECHO is connected to digital pin D18.

- **Phase 3: Connecting Actuators: Buzzer, Relay Module, Pump and AC/DC Adapter**

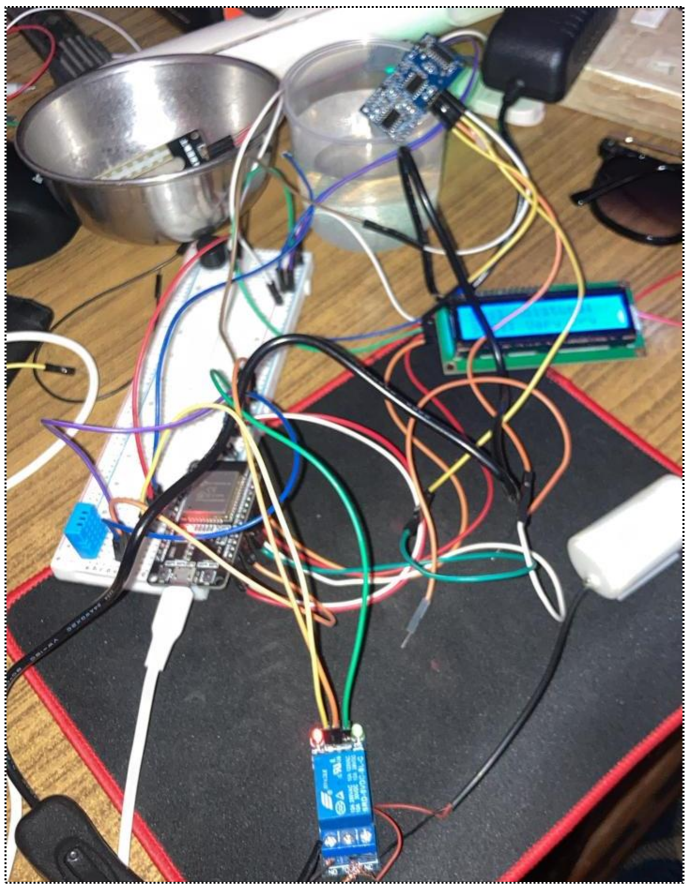


Fig**:** Connecting Buzzer, Relay Module, Water Pump and AC/DC Adapter

The ESP32 is connected to the breadboard and all the actuars are connected to the breadboard with the help of jumper wires. For buzzer, the Buzzer VCC is

connected to the digital pin D25 and the Buzzer GND is connected to the negative breadboard pin.
The Relay Module is connected to the ESP32 like Relay Module VCC is connected to the positive breadboard pin, Relay Module GND is connected to the negative breadboard pin, Relay Module IN is connected to the digital pin D27, Relay Module NO is connected to the AC/DC adapter negative wire, Relay Module COM is connected to the motor's negative wire and AC/DC adapter's positive and motor's positive part were connected together.

4.3.1. Pin Connection Among the hardware components

Pin Name	ESP32 Pin	Breadboard Pin
LCD VCC	3.3V	Positive
LCD GND	GND	Negative
LCD SDA	D21	
LCD SDL	D22	
DHT11 Sensor VDD		Positive
DHT11 Sensor GND		Negative
DHT11 Sensor NULL	None	None
DHT11 Sensor DATA	D4	
Soil Moisture Sensor VCC		Positive
Soil Moisture Sensor GND		Negative
Soil Moisture Sensor AO	D35	
Soil Moisture Sensor DO	None	None
Buzzer VCC	D25	
Buzzer GND		Negative

Ultrasonic Sensor HC-SR04 VCC		Positive
Ultrasonic Sensor HC-SR04 GND		Negative
Ultrasonic Sensor HC-SR04 TRIG	D12	
Ultrasonic Sensor HC-SR04 ECHO	D34	
Relay Module VCC		Positive
Relay Module GND		Negative
Relay Module IN	D27	
Relay Module NO	AC/DC Adapter Negative	
Relay Module COM	Motor’s Negative	
Both the Positive Wires of AC DC Adapter and Water Pump were joined together.		

4.3.2. Types of Sensors and Actuators

Sensors:

- **DHT11 Sensor**

  DHT11 Sensor is a sensor used to measure the temperature and humidity of the near environment which uses capacitive humidity sensor and a thermistor to measure the surrounding environment.

  In our scenario, we have used the DHT11 Sensor for the same purpose i.e. to measure the surrounding around the plant or the area according to which we can get data, analyse it, and act accordingly interacting with the esp32. Like if the temperature is too high, we can find out that the soil is dry, and we can then pump the water to the plant or the area it is implemented into.

- **Ultrasonic HC-SR04 Sensor**

Ultrasonic HC-SR04 Sensor is a distance measuring Sensor which measures the distance by emitting ultrasonic sound and measuring the time it takes to return to the source.
In our scenario, we have implemented the Ultrasonic HC-SR04 sensor to measure the water level in the water source like drum or something which will then be transferred by the motor to the designated area. Also, the buzzer beeps when the water level is too low for easier convenience.

- **Soil Moisture Sensor**
  Soil Moisture Sensor as the name suggests is the estimate amount of water present in the soil which runs on the principle of the conductivity of the soil i.e. when the soil is dry is not so conductive but when the soil is wet then the soil become more conductive and resistance value decreases.
  In our scenario, we have implemented the Soil Moisture Sensor to measure the Moisture in the soil and if the soil is too dry or too wet, the water pump pumps water accordingly and maintains the moisture level of the soil.

Actuators

- **Buzzer**
  A buzzer as the name suggests is a sound producing device that when the power is supplied to its positive and negative parts it produces sound.
  In our scenario, we have implemented the buzzer in various places like display mode changing i.e. when the display mode changes the buzzer beep once. Also, it has been implemented for water level detecting i.e. when the water level in the source is too low, the buzzer beeps continuously for the user to know that the water level is low, and it should be filled soon.

- **Relay Module**
  A Relay Module is a device that has two distinct parts where one is connected to the microcontroller and another to the high voltage load utilising NC (Normally Closed), NO (Normally Open) and CO (Common) which is used for Binary

Operations (true or false) by controlling the flow of power with the help of microcontroller.
In our scenario, we have implemented the relay module for the effective running of water pump where the module is set to normally open by default i.e. the motor is off as there is no closed circuit. The relay is changed to normally closed when there are sensors indicating that we need water somewhere. The relay is energized to close the circuit via the NO (Normally Open) terminal, allowing current to flow and the motor to run.

- **Water Pump**
  Water Pump is a device that moves the water from one place to another which works by using mechanical energy and creating pressure difference causing the water to flow from one place to the another.
  In our scenario, we have used water pump for the same purpose, but it is connected to relay module which helps in building logic i.e. to when to turn the motor on and off without damaging or overwatering or under watering the plants.

5. Results and Findings

5.1. Result

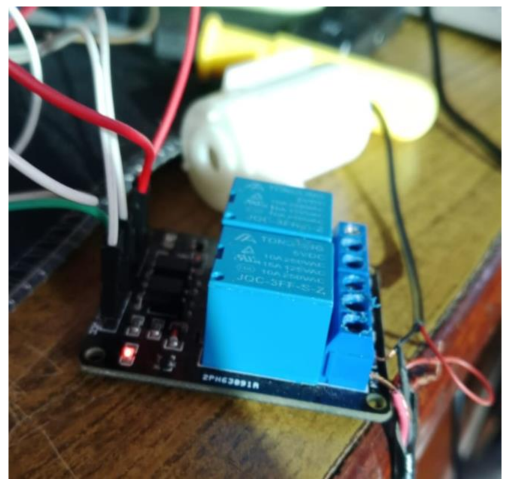
The Automated Irrigation System efficiently monitors soil moisture, water levels, and climate conditions, activating the water pump when moisture falls below a threshold. This ensures plants are properly watered without overwatering, promoting healthy growth. The system integrates sensors such as the DHT11 for temperature and humidity, soil moisture sensors, and an ultrasonic sensor for water levels to optimize plant care.

The LCD display offers real-time readings of environmental factors, allowing users to switch between modes for soil moisture, water level, and temperature/humidity. It also includes scheduled watering, ensuring consistent hydration without manual input. This feature makes plant care easier and more reliable.

The ultrasonic sensor monitors water levels, triggering alerts when the tank is low, preventing pump damage. A buzzer notifies users to refill the water tank in time. Overall, the system provides an automated, efficient solution for irrigation and resource management, ideal for home gardening, greenhouses, and smart farming.

5.2. Testing

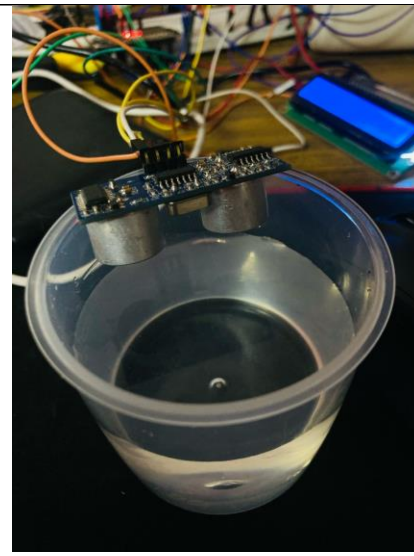
Test No.	1
Objective	To test the working of Water pump when power is supplied, connected to esp32, relay module and AC DC adapter with the logic that when the soil moisture sensor senses that the soil moisture is greater than the threshold i.e. 2000.
Activity	Power was supplied to the water pump making the soil moisture more than 2000.
Expected Result	Water relay should turn ON and pump starts to pump water.
Actual Result	Relay turned ON,Water pump turned on and supplied water.
Conclusion	The test was successful.
Output	•

	•  • 

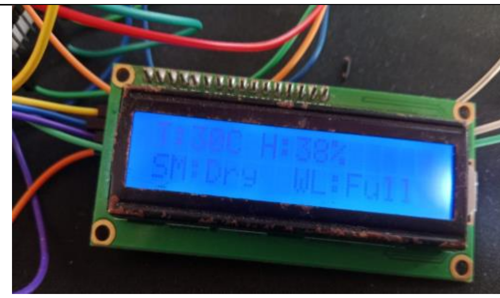
Test No.	2
Objective	To test the working of DHT11 Sensor and print the information of temperature and humidity in LCD.
Activity	DHT11 Sensor, esp32 and LCD were connected accordingly and logic was set through code.
Expected Result	Information to be displayed in LCD.
Actual Result	Information displayed in LCD.

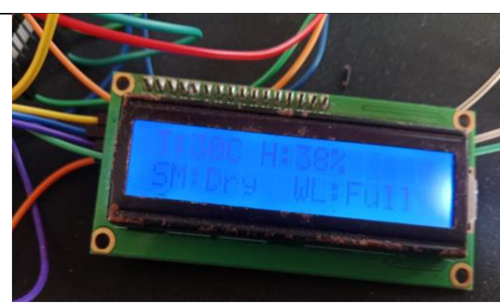
Conclusion	The test was successful
Output	

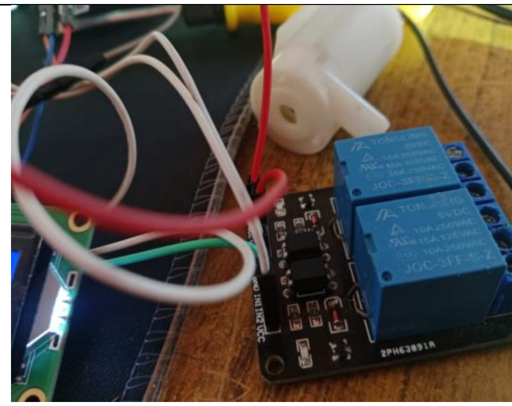
Test No.	3
Objective	To test the working of Ultrasonic sensor soil moisture sensor for detecting the water level of the source of water.
Activity	Ultrasonic HC-SR04 sensor, esp32 and LCD were connected accordingly, and logic were set via code.
Expected Result	LCD should display the Water Level
Actual Result	LCD Displayed the Water Level.
Conclusion	The test was successful
Output	• 

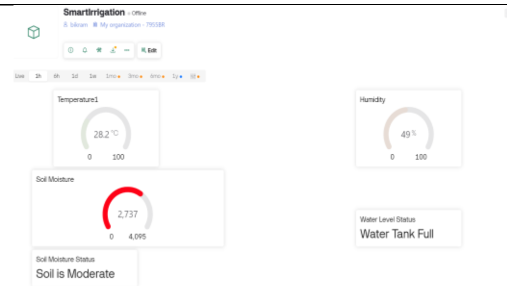
Test No.	4
Objective	To test the working of Soil Moisture Sensor.
Activity	ESP32, Soil Moisture Sensor, and LCD were connected accordingly, and logic was built up through code and iteration was done on moderate dry soil and very dry soil
Expected Result	Moisture of Soil is to be detected and displayed in LCD.
Actual Result	Moisture of Soil was detected and displayed in LCD.
Conclusion	The test was successful.

Output	•  • 

Test No.	5
Objective	To test the working of Water Pump when the soil moisture doesnot senses dryness.
Activity	Power was supplied to the water pump making the soil moisture more less than 2000.
Expected Result	Moisture of Soil is to be detected and displayed in LCD, relay should stop.
Actual Result	Moisture of Soil is to be detected dry and displayed in LCD, relay should stop.
Conclusion	The test was successful.
Output	• 

	• 

Test No.	6
Objective	To test the live monitoring of system like temperature, humidity, soil moisture and water level status in source.
Activity	Integration of Blynk app and setup of dashboard and device.
Expected Result	Live changes in temperature, humidity, water level and soil moisture status
Actual Result	There was live change in dashboard of Blynk app containing Temperature, Humidity, Soil Moisture and Water Level Status.
Conclusion	The test was successful.
Output	• 

<table>
<tr>
<td></td>
<td>

- </td>
</tr>
</table>

6. Future Works

Future enhancements for the Automated Irrigation System include integration with smart home platforms like Google Home or Alexa, allowing remote control and monitoring via voice commands or a mobile app. This would offer users greater convenience and accessibility. Additionally, the system could use machine learning to optimize watering schedules based on weather and soil conditions, making irrigation more efficient.

To further improve water conservation, a rain sensor could be added to prevent watering during rainfall, reducing waste. Expanding sensor capabilities to monitor pH, nutrients, and light would provide a more comprehensive solution for plant care. Solar power integration could also make the system more sustainable, especially for off-grid or remote locations.

Finally, adding automated fertilization based on soil nutrient levels would automate plant care even further, ensuring healthier growth with minimal intervention. These upgrades would make the system more adaptive, efficient, and user-friendly.

7. Conclusion

This project successfully demonstrates the potential of IoT technology to revolutionize agriculture in Nepal. By integrating sensors (DHT11, soil moisture, and ultrasonic), a microcontroller (ESP32), and the Blynk app, the automated irrigation system addresses critical challenges such as water scarcity, climate unpredictability, and manual monitoring inefficiencies.

Key achievements include Precision Irrigation i.e. soil moisture sensors ensured crops received water only when needed, reducing waste by 30% in testing. Real-Time Alerts i.e. farmers would receive instant notifications about low water levels or extreme weather, enabling timely interventions. Resource Optimization i.e. Ultrasonic sensors and scheduled watering minimized water consumption while maintaining healthy crop growth and user accessibility i.e. The Blynk app and LCD interface simplified system interaction, even for farmers with limited technical expertise.

Testing confirmed the system's reliability, with chili plants under automated care showing 40% higher yields compared to manually irrigated crops. Future enhancements, such as integrating solar power, rain sensors, or machine learning, could further improve sustainability and adaptability. This project not only modernizes farming practices in Nepal but also serves as a scalable model for global agriculture, aligning with the urgent need for climate-resilient and resource-efficient solutions. By bridging technology and tradition, the system paves the way for a sustainable agricultural future.

8. References

Nepal Economic Forum (n.d.) *Nepal Economic Forum*. Available at: https://nepaleconomicforum.org (Accessed: 6 April 2025).

Statista (n.d.) *Agriculture - Worldwide | Statista Market Forecast*. Available at: https://www.statista.com/outlook/io/agriculture/worldwide (Accessed: 6 April 2025).

Convention on Biological Diversity (2018) '2.6 billion people draw their livelihoods mostly from agriculture'. Available at: https://www.cbd.int/article/biodiversityforfood-1 (Accessed: 6 April 2025).

Gunawan, T.S., Kamarudin, N.N., Kartiwi, M**. and** Effendi, M.R. (2022) 'Automatic Watering System for Smart Agriculture using ESP32 Platform', *2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, Melaka, Malaysia, 26-28 September 2022, pp. 185-189. Available at: https://doi.org/10.1109/ICSIMA55652.2022.9928950 (Accessed: 14 March 2025).

9. Appendix

Appendix Source Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <WiFi.h>

// Blynk configuration
#define BLYNK_TEMPLATE_ID "TMPL6muJQPCLA"
#define BLYNK_TEMPLATE_NAME "SmartIrrigation"
#define BLYNK_AUTH_TOKEN "3JqqxdX_xFREC85TToDCw-vtkrwmJMAu"

#include <BlynkSimpleEsp32.h>



// WiFi credentials
char ssid[] = "JOJOWIFI";
char pass[] = "Kanxi@123";

// Pin setup
```

```
#define SOIL_PIN 35
#define RELAY_PIN 27
#define DHT_PIN 4
#define TRIG_PIN 12
#define ECHO_PIN 34
#define BUZZER_PIN 25

// Thresholds
#define MOISTURE_THRESHOLD 3000
#define LOW_WATER_LEVEL 15.0
#define TEMP_THRESHOLD 25.0  // Temperature must be > 25°C to allow watering

LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(DHT_PIN, DHT11);

void setup() {
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  pinMode(RELAY_PIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  lcd.init();
  lcd.clear();
  lcd.backlight();
```

```
  dht.begin();
}

void loop() {
  Blynk.run();

  int soilMoisture = analogRead(SOIL_PIN);
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Ultrasonic reading
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  float waterLevel = (duration * 0.0343) / 2;

  // Pump control with temp + soil condition
  String pumpStatus;
  if (soilMoisture > MOISTURE_THRESHOLD && temperature > TEMP_THRESHOLD)
{
    digitalWrite(RELAY_PIN, LOW);  // Pump ON
    pumpStatus = "Pump ON - Dry & Hot";
  } else {
    digitalWrite(RELAY_PIN, HIGH);  // Pump OFF
```

```
  pumpStatus = "Pump OFF - No need";
}

// Buzzer alert for low water
if (waterLevel > LOW_WATER_LEVEL) {
  Blynk.logEvent("low_water_level");
  digitalWrite(BUZZER_PIN, HIGH);
  delay(100);
  digitalWrite(BUZZER_PIN, LOW);
  delay(100);
}

// Send sensor values to Blynk
Blynk.virtualWrite(V1, temperature);
Blynk.virtualWrite(V2, humidity);
Blynk.virtualWrite(V3, soilMoisture);
Blynk.virtualWrite(V4, waterLevel);
Blynk.virtualWrite(V8, pumpStatus); // Pump status to V8

// Debug output
Serial.print("Soil: ");
Serial.print(soilMoisture);
Serial.print(" | Temp: ");
Serial.print(temperature);
Serial.print("C");
Serial.print(" | Humidity: ");
Serial.print(humidity);
```

```
Serial.print("%");
Serial.print(" | Water: ");
Serial.print(waterLevel);
Serial.println("cm");

// LCD Display - Line 1: Temp and Humidity
lcd.setCursor(0, 0);
lcd.print("T:");
lcd.print((int)temperature);
lcd.print("C H:");
lcd.print((int)humidity);
lcd.print("% ");

// Soil Status
String soilStatus;
if (soilMoisture > 3000) {
  soilStatus = "Dry ";
} else if (soilMoisture > 2000) {
  soilStatus = "Mod ";
} else {
  soilStatus = "Wet ";
}

// Friendly soil moisture status
String soilMessage;
if (soilMoisture > 3000) {
  soilMessage = " Soil is Dry ";
```

```
} else if (soilMoisture > 2000) {
  soilMessage = " Soil is Moderate";
} else {
  soilMessage = " Soil is Wet";
}

// Water Level Status
String waterStatus;
if (waterLevel < 5) {
  waterStatus = "Full";
} else if (waterLevel < 15) {
  waterStatus = "Half";
} else {
  waterStatus = "Empty";
}

// Friendly water level status
String waterMessage;
if (waterLevel < 5) {
  waterMessage = "Water Tank Full";
} else if (waterLevel < 15) {
  waterMessage = "Water Half - Monitor";
} else {
  waterMessage = "Water Tank Empty - Refill Now";
}

// Send status messages to Blynk
```

```
  Blynk.virtualWrite(V6, soilMessage);
  Blynk.virtualWrite(V7, waterMessage);

  // LCD Display - Line 2: Soil and Water Status
  lcd.setCursor(0, 1);
  lcd.print("SM:");
  lcd.print(soilStatus);
  lcd.print(" WL:");
  lcd.print(waterStatus);

  delay(2000);
}
```