UTTARA UNIVERSITY

Department of Electrical & Electronics Engineering

SMART IRRIGATION SYSTEM

DESIGN AND SIMULATE SOLAR POWERED WATER PUMPING SYSTEM FOR IRRIGATION

MD. ARAFATH HOSSEN (2213171025)
ASHIK HOSSAIN (2203171016)
MD. ABDUR RAHIM (2213171011)
MD. SOHEL RANA (2183161004)

In Partial Fulfillment of the Requirements for the Degree of

BSc. in Electrical & Electronics Engineering

December 2024

Declaration

We do hereby cordially declare that the work presented in the project report has been carried out buys and has not been previously submitted for any Degree in University/College /Organization or any professional qualification.

Signature of Supervisor Signature of Authors

Md. Tanvir Shahed	MD.ARAFATH HOSSEN (2213171025)
Assistant Professor	ASHIK HOSSAIN (2203171016)
Dept. of EEE, Uttara University	MD. ABDUR RAHIM (2213171011)
	MD SOHEL RANA (2183161004)

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

We highly indebted to Md. Tanvir Shahed for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards our member of Uttara University Dept. of EEE for their kind co-operation and encouragement which help me in completion of this project.

We would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

Thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Abstract

Bangladeshi agriculture is diverse ranging from impoverished farm villages to developed farms utilizing modern agricultural technologies. Facility agriculture area in China is expanding and is leading the world. However, its ecosystem control technology is still immature, with low level of intelligence. Promoting application of modem information technology in agriculture will solve a series of problems facing by farmers. Lack of exact information and communication leads to the loss in production. This system provides an intelligent monitoring platform framework and system structure for facility agriculture ecosystem based on IOT. The Internet of Things makes everything connected. Over 50 years since independence, Bangladesh has made immense progress towards food productivity. Modern agriculture practices have a great promise for the economic development of a nation. So we have brought-in an innovative project for the welfare of farmers and also for the farms. There are no day or night restrictions. This is helpful at any time.

Table of Contents

List of Table

LIST OF FIGURE

Introduction

Agriculture remains the backbone of many economies, but its sustainability is increasingly threatened by challenges such as water scarcity, inefficient resource utilization, and reliance on non-renewable energy sources. To address these issues, modern agricultural practices are turning toward innovative technologies that blend renewable energy and intelligent systems. Among these, solar-powered water pumping systems integrated with IoT-based smart irrigation solutions offer a promising approach to sustainable and efficient farming.

This project, titled "Design and Simulate Solar Powered Water Pumping Smart Irrigation System Using New Blynk App," explores the convergence of renewable energy and IoT technologies to create an intelligent irrigation system. The project leverages solar power as a clean, renewable energy source to operate water pumps, thereby reducing dependency on traditional electricity or diesel-powered systems. Simultaneously, it employs IoT sensors and controllers to monitor and optimize irrigation in real time, ensuring efficient water usage and improved crop yield.

The innovation lies in the integration of the Blynk app, a versatile IoT platform that allows seamless communication between hardware components and the user. This application facilitates remote monitoring and control of the irrigation system, enabling farmers to make data-driven decisions and reduce manual intervention. By combining renewable energy and IoT, this project addresses critical issues in agriculture while promoting environmental sustainability.

The objectives of this project are to:

1. Design a solar-powered water pumping system tailored to meet irrigation requirements.
2. Develop a smart irrigation system that uses IoT technology to monitor environmental conditions such as soil moisture and temperature.
3. Simulate the system's performance using advanced tools to optimize energy efficiency and water usage.
4. Integrate the Blynk app for real-time control and data visualization.
5. Analyze the system's effectiveness in addressing agricultural challenges and propose improvements based on simulation results.

Through this project, we aim to demonstrate the potential of renewable energy and IoT in transforming traditional agricultural practices into sustainable and technology-driven solutions. The outcomes of this research will contribute to the development of smarter, greener, and more resilient agricultural systems for a sustainable future.

Literature Review

The implementation of a solar-powered water pumping smart irrigation system is grounded in interdisciplinary research, combining solar energy technology, water pumping mechanisms, and IoT-based smart irrigation systems. This review highlights key contributions from existing literature, identifying the technological advancements and challenges in these domains.

Solar-Powered Irrigation Systems

Solar energy has emerged as a sustainable solution for agricultural applications, particularly in water-scarce regions. Duffie and Beckman (2013) provide a comprehensive overview of solar thermal processes, emphasizing the role of photovoltaic (PV) technology in generating clean energy. Chandel et al. (2015) discuss the technical and economic feasibility of solar-powered water pumping systems, highlighting their potential to replace conventional irrigation methods. Meah et al. (2008) further explore the application of solar PV systems in remote areas, demonstrating their reliability in off-grid settings. These studies underscore the importance of optimizing PV panel efficiency and integrating energy storage solutions to ensure uninterrupted operation.[1] *(Duffie, 2013)*

Water Pumping Technologies

Efficient water pumping mechanisms are critical for effective irrigation. Zou and Zhao (2018) analyze solar-powered irrigation systems designed for arid regions, emphasizing the need for tailored solutions based on local conditions. Al-Karaghouli and Kazmerski (2013) compare conventional and renewable energy-powered pumping systems, illustrating the cost-effectiveness and sustainability of solar-powered alternatives. Their findings reveal that factors such as head height, flow rate, and power requirements significantly influence the performance of water pumps in solar-powered systems.

IoT in Smart Irrigation

The integration of IoT technologies has revolutionized irrigation practices by enabling real-time monitoring and control. Kumar and Chaturvedi (2016) demonstrate the effectiveness of IoT-based systems in optimizing water usage and reducing manual intervention. Their research highlights the use of sensors to monitor soil moisture, temperature, and water levels, which informs automated irrigation decisions. The Blynk app, as documented in its official resources (Blynk Documentation, n.d.), offers a user-friendly platform for IoT integration, facilitating remote monitoring and control of irrigation systems. Arduino IDE's compatibility with IoT devices (Arduino, n.d.) further simplifies the development of customized solutions for smart irrigation.

Simulation and Design Tools

Simulation tools play a pivotal role in designing and evaluating solar-powered irrigation systems. PVsyst software (PVsyst, n.d.) provides detailed modeling of PV system performance under various environmental conditions, enabling designers to optimize energy generation. Proteus Design Suite (Proteus Design Suite, n.d.) supports circuit simulation for IoT-based systems, ensuring seamless integration of sensors and controllers. MATLAB/Simulink and ANSYS are also commonly employed to simulate water flow and system dynamics, as demonstrated in multiple case studies.

Challenges and Future Directions

Despite significant advancements, challenges remain in the widespread adoption of solar-powered smart irrigation systems. Issues such as high initial costs, variable solar energy availability, and the need for skilled maintenance personnel limit their scalability. Research continues to focus on improving energy storage technologies, enhancing sensor accuracy, and developing cost-effective solutions to address these barriers. The combination of renewable energy and IoT technologies holds promise for sustainable agriculture, as evidenced by the growing body of literature.

Methodology

The methodology for designing and simulating a solar-powered water pumping smart irrigation system using the new Blynk app involves a structured approach encompassing research, design, simulation, and analysis. This section outlines the key steps undertaken in the project..

System Design

Solar Power System

Calculated the power requirements for the water pump based on the irrigation area and water flow rate.

Selected appropriate solar panels, charge controllers, and batteries to ensure uninterrupted operation.

Water Pumping System

Identified a suitable water pump with optimal efficiency for solar-powered operation.

Designed a layout for the water distribution network tailored to the irrigation requirements.

IoT Integration

Selected IoT devices (e.g., ESP32/NodeMCU) for system control and data transmission.

Integrated sensors such as soil moisture, temperature, and water level sensors to gather environmental data.

Developed a circuit diagram integrating all components.

Simulation

Solar Power Simulation

Used simulation tools like MATLAB/Simulink or PVsyst to model solar panel performance under various weather conditions.

Analyzed power generation and storage efficiency.

Pumping System Simulation

Simulated water flow rates and pump operation using tools like Proteus or ANSYS.

IoT System Simulation

Configured the Blynk app to visualize sensor data and control the water pump in real-time.

Used platforms like Arduino IDE and Blynk's widget library to test the IoT setup.

Implementation

Hardware Assembly

Procured and assembled the solar panels, pump, IoT devices, and sensors.

Connected the components according to the designed circuit diagram.

Software Development

Programmed the IoT microcontroller using Arduino IDE to handle sensor data and communication with the

Blynk app.

Customized the Blynk app interface for real-time monitoring and control. (https://docs.blynk.io/)

Testing and Validation

Conducted field tests to evaluate the performance of the solar panels, pump, and sensors.

Monitored the system's efficiency under varying environmental conditions.

Validated the IoT integration by testing real-time data updates and pump control via the Blynk app.

Data Analysis

Analysed sensor data to assess irrigation effectiveness and water usage efficiency.

Evaluated the energy consumption of the system and compared it with the simulated values.

Optimization

Identified areas for improvement based on testing results, such as enhancing energy storage capacity or improving sensor calibration.

Iteratively refined the hardware and software components to achieve optimal performance.

Documented all design parameters, simulation results, and testing procedures.

Prepared visual aids such as flowcharts, diagrams, and graphs to illustrate system functionality.

Tools and Technologies Used

Hardware: Solar panels, DC/AC water pump, IoT sensors (soil moisture, temperature), microcontroller (ESP32/NodeMCU), batteries, and charge controllers.

Software: Arduino IDE, Blynk app, MATLAB/Simulink, Proteus, and PVsyst.

This methodology ensured a systematic approach to designing and simulating the solar-powered smart irrigation system, resulting in a robust and efficient solution for sustainable agriculture.

Theory

The design and simulation of a solar-powered water pumping smart irrigation system using the new Blynk app involve integrating solar energy and IoT technology to achieve sustainable and efficient irrigation. The theoretical framework encompasses three primary domains: solar energy utilization, water pumping mechanisms, and IoT-enabled smart systems.

Solar Energy Utilization

Solar energy is a renewable and abundant energy source that can power irrigation systems sustainably. Photovoltaic (PV) panels convert sunlight into electrical energy, which is either directly used to run the water pump or stored in batteries for later use. The efficiency of solar panels depends on factors such as sunlight intensity, angle of installation, and temperature. To optimize energy usage, charge controllers regulate the power flow to the pump and batteries, ensuring the system's reliability.

Water Pumping Mechanisms

Water pumping systems play a crucial role in irrigation by ensuring a consistent and adequate water supply to crops. Solar-powered pumps, typically DC or AC types, draw water from a source such as a well or reservoir and distribute it across the field. The pump's performance is influenced by parameters such as head height, flow rate, and power input. Efficient design ensures minimal energy loss and meets the irrigation requirements effectively. (Chandel, 2015)

IoT-Enabled Smart Systems

IoT technology integrates sensors, controllers, and communication modules to enable real-time monitoring and control of irrigation systems. Sensors such as soil moisture, temperature, and water level provide essential data about environmental conditions. Microcontrollers (e.g., ESP32/NodeMCU) process this data and communicate with the Blynk app, which serves as the user interface. The app allows users to monitor system performance, control the pump remotely, and receive alerts, thereby reducing manual intervention and optimizing water usage.

Integration of Components

The integration of solar energy, water pumping, and IoT technologies creates a smart irrigation system that is both energy-efficient and environmentally friendly. The solar panels provide sustainable power, the pumps ensure water delivery, and the IoT system enables intelligent decision-making based on real-time data. Together, these components form a cohesive system that addresses the challenges of water scarcity and energy consumption in agriculture.

This theoretical foundation supports the methodology described in this project, guiding the design, simulation, and implementation processes to achieve an innovative solution for smart irrigation.

Types of Sensors Used in Smart Irrigation

Smart irrigation systems rely heavily on sensors to optimize water usage based on environmental conditions. These sensors help monitor soil moisture, temperature, humidity, and other variables, ensuring that water is applied only when necessary. Some of the most commonly used sensors in smart irrigation systems include:

Soil Moisture Sensors: These are essential in determining the water content in the soil. They can be capacitive, resistive, or volumetric, and they help decide when irrigation is required. They help avoid both over-irrigation and under-irrigation, leading to efficient water use.

Rain Sensors: These sensors detect rainfall and help prevent irrigation systems from operating when it is raining, thereby saving water.

Temperature and Humidity Sensors: These sensors measure the environmental temperature and relative humidity. Higher temperatures and lower humidity typically increase the rate of evaporation, signaling a need for irrigation.

Flow Sensors: These measure the flow of water through irrigation pipes. They can help detect blockages, leaks, or other system malfunctions, ensuring efficient water distribution.

Solar Radiation Sensors: These measure the amount of solar energy reaching the ground. Higher solar radiation often corresponds to higher water evaporation, and thus, more irrigation may be required.

Communication Protocols in Smart Irrigation

To enable communication between various components in a smart irrigation system, different communication protocols are used. These protocols allow sensors, microcontrollers, and actuators to exchange data and commands, often in real-time.

Wi-Fi: Wi-Fi is widely used in smart irrigation systems because it provides a high-speed, reliable connection for transmitting data. Wi-Fi is often used in systems where there is easy access to a power source and the distance between devices is relatively short. It allows remote monitoring and control through apps or web interfaces.

RFID (Radio Frequency Identification): RFID is typically used in smart irrigation for asset tracking, such as identifying and managing irrigation equipment or sensors. In the context of irrigation, RFID tags can be attached to devices or sensors, allowing them to be easily tracked and managed via an RFID reader.

GSM (Global System for Mobile Communications): GSM communication is used for remote control and monitoring of irrigation systems, especially in areas where Wi-Fi may not be available. Through GSM modules, data from sensors can be sent to a mobile phone via SMS, or the system can receive commands from remote locations. This communication is especially useful for remote agricultural areas.

Types of Microcontrollers Used in Smart Irrigation

Microcontrollers form the brain of a smart irrigation system, processing data from sensors and controlling irrigation actuators like valves and pumps. Several types of microcontrollers are commonly used, each offering unique advantages based on the complexity of the system and the requirements of the project.

ESP32: The ESP32 is a powerful microcontroller known for its built-in Wi-Fi and Bluetooth capabilities. It's ideal for smart irrigation systems because it allows easy integration with IoT platforms and supports remote control through web or mobile applications. The ESP32 is energy-efficient, has a variety of analog and digital I/O pins, and is suitable for both small and large-scale systems.

Arduino Uno: Arduino Uno is one of the most popular microcontrollers used in smart irrigation systems due to its simplicity, open-source nature, and large community support. It is easy to program and integrate with a wide range of sensors, making it ideal for small to medium-sized projects. However, it lacks built-in wireless communication capabilities like Wi-Fi, so additional modules like Wi-Fi shields or GSM modules are required for remote communication.

Raspberry Pi: The Raspberry Pi is a single-board computer that offers higher processing power than typical microcontrollers. It is ideal for advanced smart irrigation systems that require more complex computations, data logging, or integration with a full-fledged user interface. The Raspberry Pi supports a wide range of communication protocols, including Wi-Fi, Ethernet, and Bluetooth, and can easily run Python or other high-level programming languages for sensor data processing and system control. (Ard)

Process and results

System Design Overview

The project utilizes the following components:

NodeMCU ESP8266: This microcontroller acts as the brain of the system. It will:

- Process the inputs from the **Soil Moisture Sensor** and **DHT22 Sensor**.
- Control the **Water Pump** through the **Relay Module**.
- Provide connectivity to the **Blynk app** for remote monitoring and control.

16x2 LCD Screen: The LCD screen displays real-time information like:

- Soil moisture levels.
- Temperature and humidity readings from the DHT22 sensor.
- System status (whether the pump is on or off).

DHT22 Sensor: This sensor is used to measure:

- **Temperature** and **Humidity** in the environment. This data is displayed on the LCD screen and can be useful for monitoring the climate conditions affecting soil moisture and irrigation needs.

Soil Moisture Sensor: Measures the moisture level in the soil. The microcontroller uses this data to trigger the water pump via the relay when the moisture level falls below a preset threshold.

Relay Module (2 modules): These are used to control the water pump. The **NodeMCU ESP8266** sends signals to the relay to activate or deactivate the pump based on the soil moisture sensor readings.

Water Pump: A small DC pump that extracts water from a reservoir or well to irrigate the crops when the system detects that the soil is dry.

System Workflow

Soil Moisture Monitoring:

- The **Soil Moisture Sensor** continuously checks the moisture content in the soil.
- If the soil becomes too dry (below the threshold), the **NodeMCU ESP8266** sends a signal to the **Relay Module** to activate the **Water Pump**.

Temperature and Humidity Monitoring:

- The **DHT22 Sensor** measures the temperature and humidity of the environment.
- The **NodeMCU ESP8266** reads the data and displays it on the **16x2 LCD screen** for local monitoring.

Water Pumping:

- When the soil moisture drops below a set threshold, the **NodeMCU** triggers the relay, turning on the **Water Pump** to irrigate the plants.
- The pump stays on until the moisture level is adequate or until a preset irrigation time limit is reached.

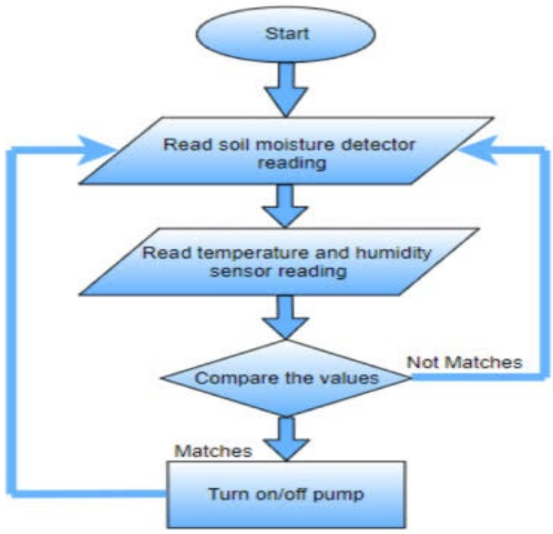
Remote Control and Monitoring (via Blynk):

- The **NodeMCU ESP8266** connects to Wi-Fi and sends sensor data (soil moisture, temperature, and humidity) to the **Blynk app**.
- The user can monitor the system remotely, check soil moisture levels, view temperature/humidity readings, and manually control the pump if necessary.

Local Display:

- The **16x2 LCD screen** provides real-time feedback on the system's status, displaying information like:
  - Current soil moisture level.
  - Temperature and humidity.
  - Whether the water pump is on or off.

Flow Chart:



Figur 1.1 Shows Flow Chart of Smart irregation

Explanation of the Flowchart:

Start (System On): The system initializes.

Read Soil Moisture: The **soil moisture sensor** reads the moisture level in the soil.

Moisture Level OK?: If the soil is dry (below the set threshold), it moves to activate the water pump.

Read Temperature & Humidity (DHT22): Simultaneously, the **DHT22 sensor** measures temperature and humidity.

Display Data on LCD: This information is displayed on the **16x2 LCD screen**.

Send Data to Blynk: The **NodeMCU ESP8266** sends the readings to the **Blynk app**.

Manual Control via Blynk: The user can manually control the water pump via the app if needed.

Turn On Water Pump: If moisture is low, the **relay module** activates the water pump.

Pump On for X Time: The pump runs for a preset time or until the soil moisture reaches the optimal level.

Check Moisture Level After Irrigation: After a set irrigation time, the system checks the moisture again.

Irrigation Complete: Once the soil moisture level is sufficient, the irrigation cycle is complete.

System Idle/Standby: The system goes idle until moisture levels drop again.

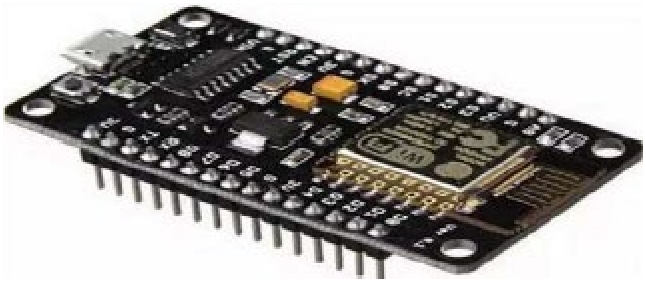
End (System Off): If the system is manually turned off or the battery is low, the system stops.

Basic Requirement

- NodeMCU ESP8266
- 16x2 LCD Screen
- DHT22 Sensor
- Soil Moisture Sensor
- 2 Really
- Module Water Pump

NodeMCU ESP8266

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from EspressifSystems, and hardware which is based on the ESP-12 module.



Figur 1.2 Shows NodeMCU ESP8266

Here We just Show the Pin Diagram Of The ESP8266 Board On the Right showing all Digital GPIO Pins available. In this ESP8266 Board Is Only One Analog Pin available And They are connected to the Soil Moisture Sensor.

NodeMCU ESP8266 Specifications & Features

• Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX l 06
• Operating Voltage: 3.3V
• Input Voltage: 7 12V
• Digital 1/0 Pins (DIO): 16
• Analog Input Pins (ADC): 1
• UARTs: I
• SPis: 1
• I2Cs: 1
• Flash Memory: 4 MB
• SRAM: 64 KB
• Clock Speed: 80 MHz
• USB-TTL based on CP2 l 02 is included on board, Enabling Plug n Play
• PCB Antenna
• Small Sized module to fit smartly inside your IoT projects

Brief About NodeMCU ESP8266

The NodeMCU ESP8266 development board comes with the ESP- l 2E module containing ESP8266 chip having TensilicaXtensa 32-bit LX I 06 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi *I* Bluetooth and Deep Sleep Operating features make it ideal for loT projects. NodeMCU can be powered using Micro USB jack and VIN pin (Extemal Supply Pin). It supports UART, SPI, and I2C interface.

Programming NodeMCU ESP8266 with Arduino IDE

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use.

Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself.

Uploading your first program

Once Arduino IDE is installed on the computer, connect the board with the
computer using the USB cable. Now open the Arduino IDE and choose the correct board by selecting Tools>Boards>NodeMCU 1.0 (ESP-I 2E Module), and choose the correct Port by selecting Tools>P011.
To get it started with the NodeMCU board and blink the built-in LED, load the
example code by selecting Files>Examples>Basics>Blink. Once the example code is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the built-in LED of the board blinking.

Applications of NodeMCU

• Prototyping of IoT devices
• Low power battery operated applications
• Network projects
• Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth Functionalities

16×2 LCD Display

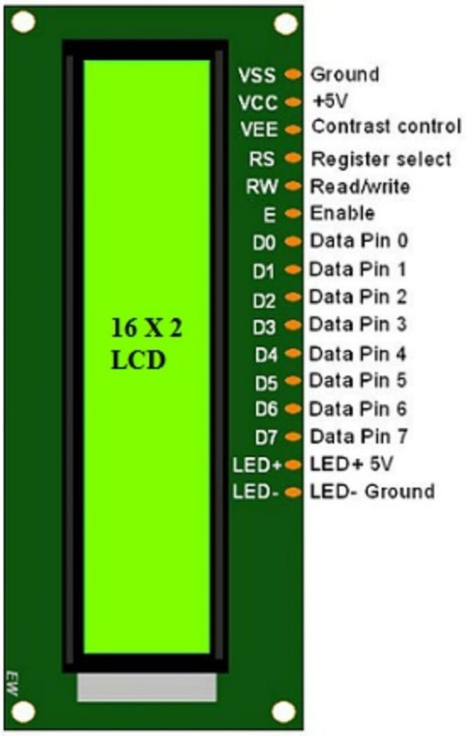
Here we used the 16×2 LCD Display and they will be connected to the I2C Protocol

VCC-5V

GND-GND

SDA-A4

SCL-A5



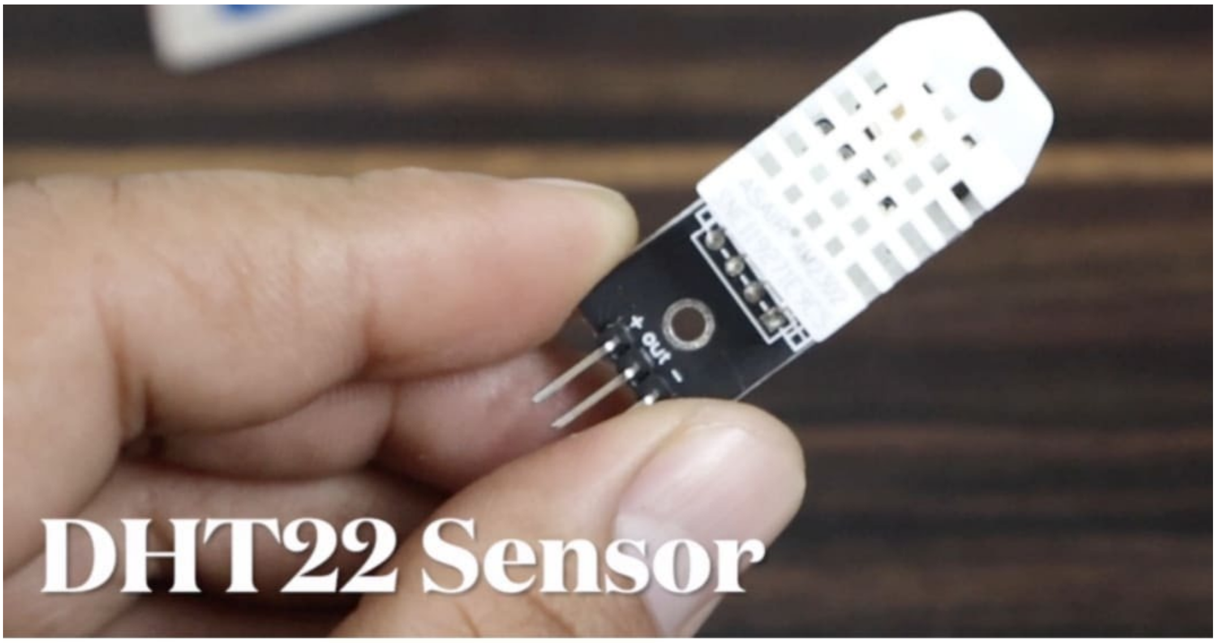
LCD-16x2-pin-diagram

Figur 1.3 Shows 16×2 LCD Display

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO's or calculators

DHT22 Sensor

It Measuring The Temperature And Humidity And They Will be Connected To The Pin Number D3
VCC-5V Digital Pin-D3 GND-GND



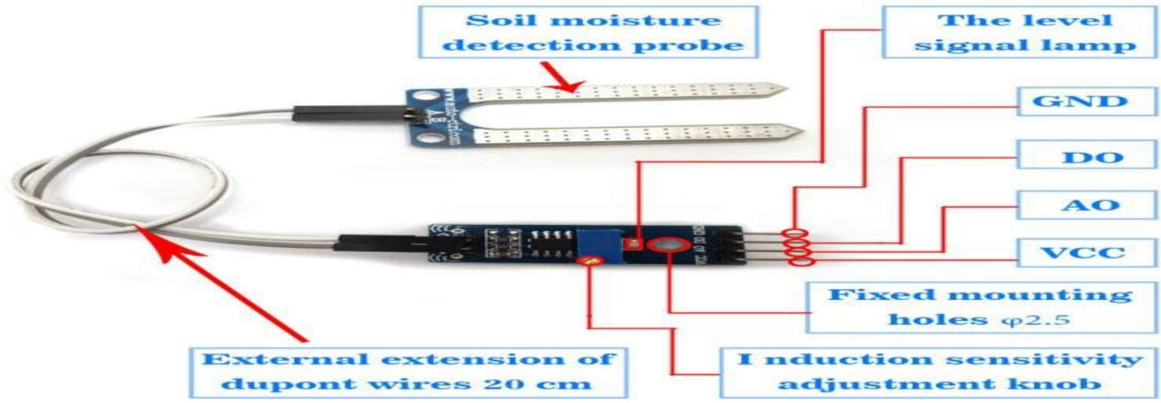
Figur 1.4 Shows DHT22 Sensor

The DHT22 sensor measures temperature and humidity using a thermistor and a capacitive humidity sensor. It outputs data digitally via a single pin using a 1-wire protocol. The sensor provides 16-bit data for both humidity and temperature, along with an 8-bit checksum for validation. It operates on 3.3V–6V, with a sampling rate of 0.5Hz (1 reading every 2 seconds). It's accurate, easy to use, and ideal for applications like weather monitoring and climate control.

Soil Moisture Sensor

Soil Moisture Sensor measures the soil water level in the earth and sends data to Blynk App The Will is connected to Pin Number A0.

VCC-5V Analog Pin -A0 GND- GND



Figur 1.5 Shows Soil Moisture Sensor

When there is more water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there will be less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower.

Features :

1. Dual output mode, analog output more accurate
2. A fixed bolt hole for easy installation
3. With power indicator (red) and digital switching output indicator (green)
4. Having LM393 comparator chip, stable.

The module includes a potentiometer for adjusting the sensitivity of the digital output (DO). you can use it to set a threshold, so that when the soil moisture level exceeds the threshold, the module outputs LOW otherwise HIGH. This setup is very useful for triggering an action when a certain threshold is reached. For example, if the moisture level in the soil exceeds a certain

threshold, you can activate a relay to start watering the plant.

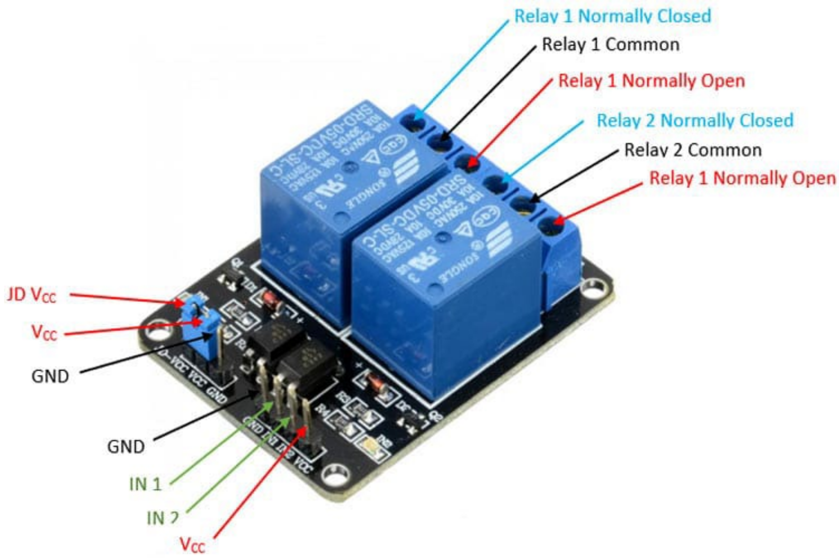
2 Relay Module

Here We Used The 5v relay module and they are well connected to PIN D5.

VCC – 5V

Relay Pin – D5

GND – GND

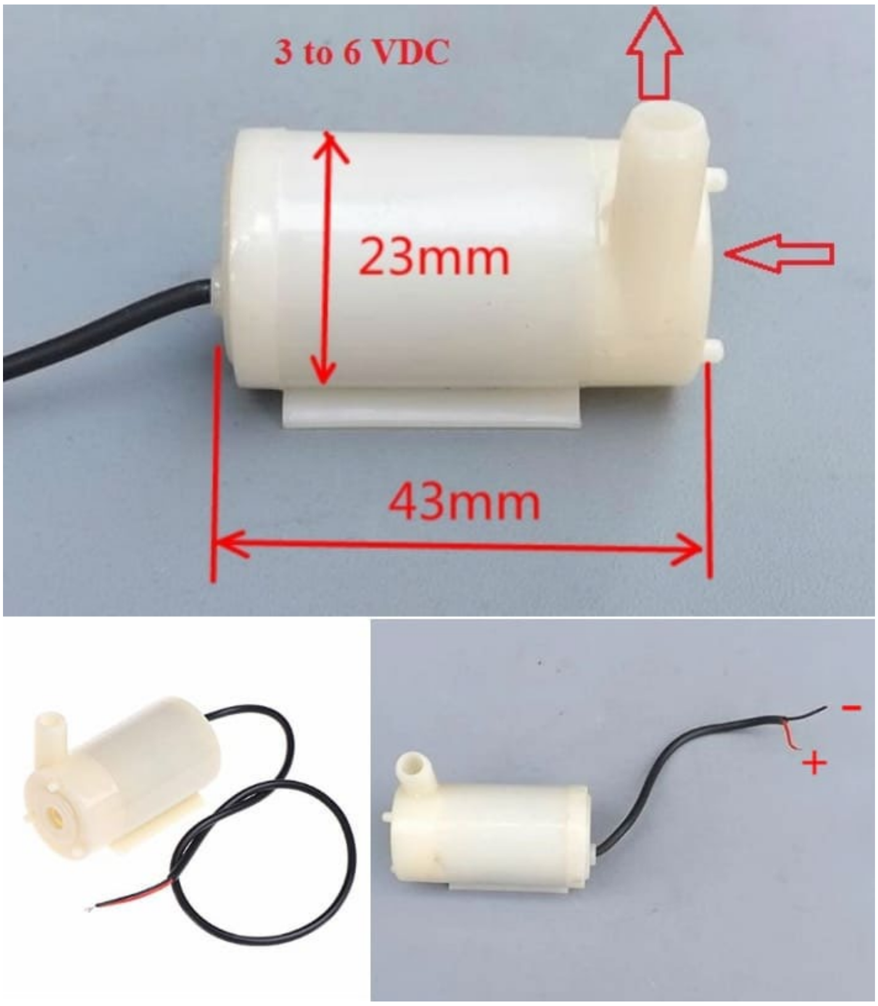


Figur 1.6 Shows 2 Relay Module

Relay is an electromagnetic device which is used to isolate two circuits electrically and connect them magnetically. They are very useful devices and allow one circuit to switch another one while they are completely separate. They are often used to interface an electronic circuit (working at a low voltage) to an electrical circuit which works at very high voltage. For example, a relay can make a SY DC battery circuit to switch a 230V AC mains circuit. Thus a small sensor circuit can drive, say, a fan or an electric bulb.

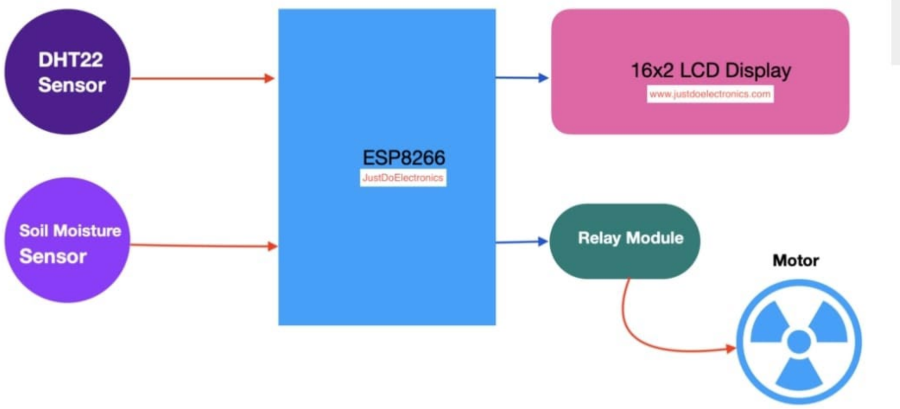
DC PUMP

A 9 volt DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC Pumps have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor. DC Pumps were the first type widely used, since they could be powered f rom existing direct cmTent lighting power distribution systems. A DC Pump's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances.



Figur 1.7 Shows DC PUMP

Block Diagram:

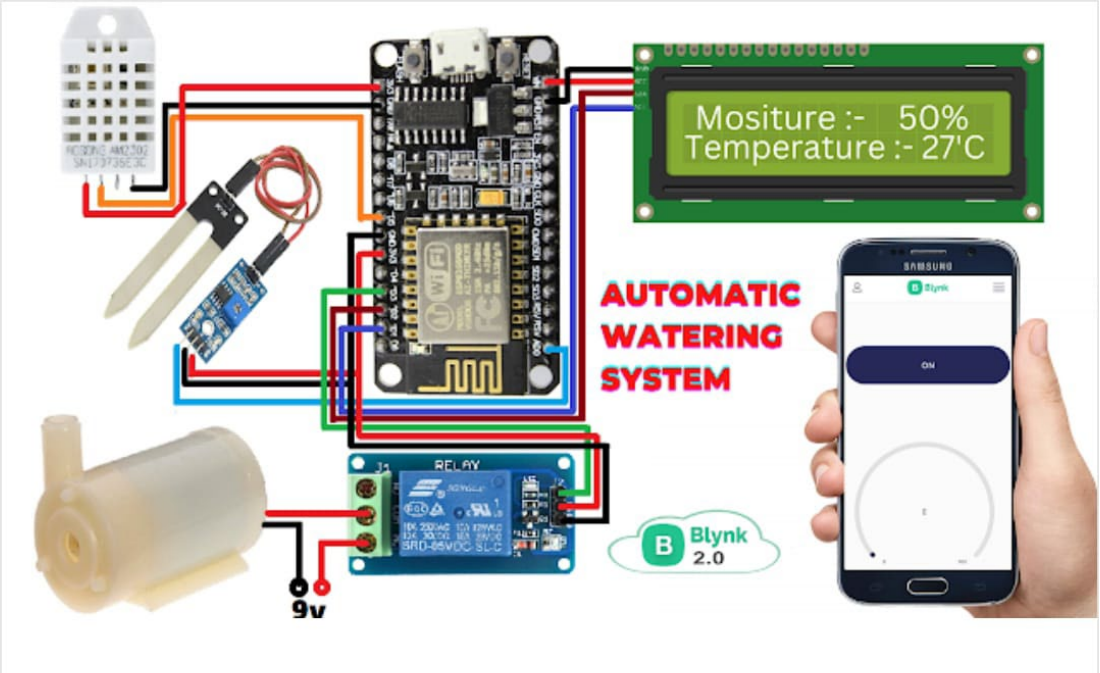


Figur 1.8 Shows System Block Diagram

Block Diagram

- Solar Panel (Optional):
- Powers the system and charges the battery for continuous operation. Battery:
- Stores energy for low power. NodeMCU ESP8266:
- Central controller that reads sensor data, controls relay and water pump. Sends data to Blynk app for remote monitoring and control. Soil Moisture Sensor:
- Measures soil moisture and tells NodeMCU to irrigate. DHT22 Sensor:
- Monitors temperature and humidity. Relay Module:
- Toggles water pump ON/OFF based on NodeMCU commands. Water Pump:
- Pumps water when relay is activated. 16x2 LCD Screen:
- Displays real-time data like soil moisture, temperature, humidity, pump status. Blynk App:
- Remote monitoring of sensor data and manual control of water pump. Working Flow: Sensors → NodeMCU → Relay Module → Water Pump. NodeMCU sends data to LCD for local display and Blynk app for remote monitoring and control. This is a compact view of how each block works together.

Circuit Diagram:



Figur 1.9 Shows Circuit Diagram

Circuit Diagram

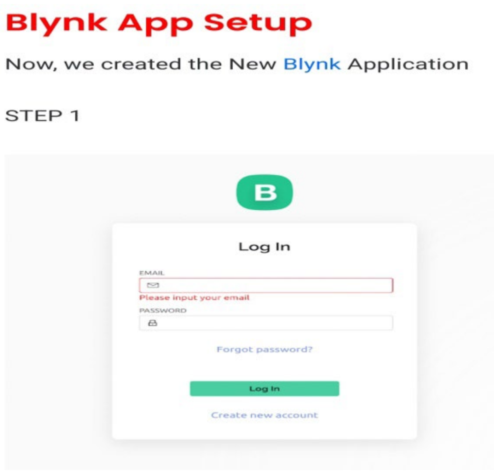
The circuit diagram of this project represents a seamless fusion of renewable energy and IoT technology to create an intelligent, self-sustaining irrigation system. At its core, solar panels harness sunlight to power the system, regulated by a charge controller and battery storage to ensure uninterrupted operation. A microcontroller (ESP32/NodeMCU) acts as the brain, processing real-time data from soil moisture, temperature, and water level sensors to optimize irrigation efficiency. The water pump is activated only when needed, reducing energy and water wastage. Meanwhile, the Blynk app provides a dynamic user interface, enabling remote monitoring and control, making farming smarter, more efficient, and eco-friendly.

SOFTWARE BLYNK

Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. After downloading the app, create an account and log in. (If possible than log in with your real mail id for better connectivity later.) You'll also need to install the Blynk Arduino Library, which helps generate the firmware running on your ESP8266. Download the latest release from https://github.com/blynkkk/blynk-library/releases , and follow along with the directions there to install the required libraries. Create a Blynk Project Click the "Create New Project" in the app to create a new Blynk app. Give it any name Blynk works with hundreds of hardware models and connection types. Select the Hardware type. After this, select connection type. In this project we have select Wi-Fi connectivity. The Source code is very important - you'll need to stick it into your ESP8266's firm ware. For now, copy it down or use the "E-mail" button to send it to yourself. (https://docs.blynk.io/)

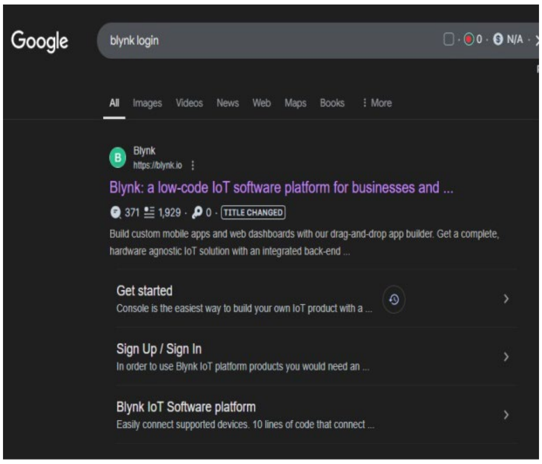
Add Widgets To The Project

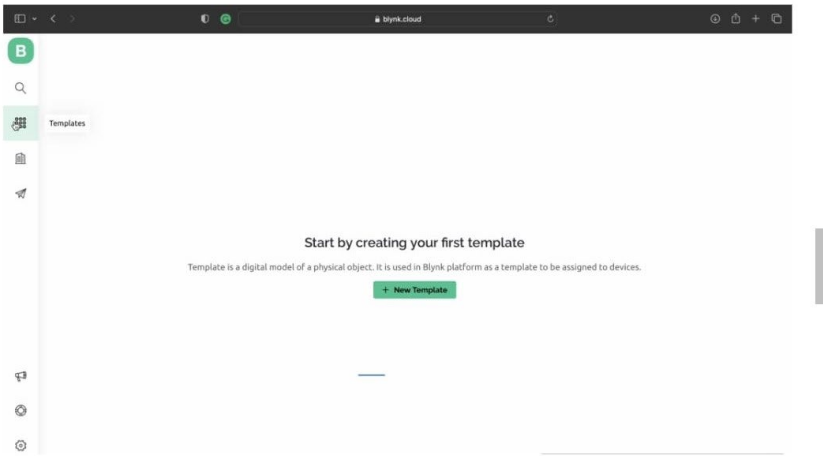
Then you'll be presented with a blank new project. To open the widget box, click in the project window to open.
We are selecting a button to control Led connected with NodeMCU. I .Click on Button. 2.Give name to Button say led. 3.Under OUTPUT tab- Click pin and select the pin to which led is connected to NodeMCU, here it is digital pin 2, hence select digital and under pin D2. And Click continue. Under MODE tab- Select whether you want this button as "push button" or "Switch".You have successfully created a GUI for Arduino.



Upload The Firmware

Now that your Blynk project is set-up, open Arduino and navigate to the ESP8266_Standalone example in the File > Examples >Blynk>Boards_ WiFi> ESP8266 Standalone menu.





Figur 1.10 Shows Blynk app Process

Step 1: Install and Set Up the Blynk App

- **Action**: Begin by downloading the latest version of the Blynk app from either the Google Play Store or Apple App Store or website of Blynk
- **Create Account/Log In**: After installation, either create a new Blynk account or log into an existing one.
- **Create New Project**:

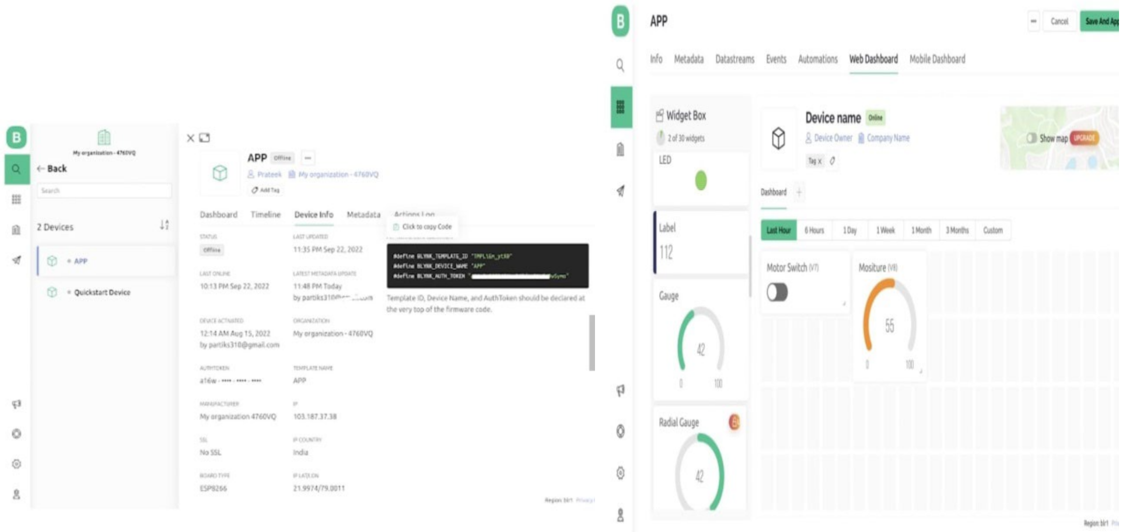
- Open the app and click on "Create New Project."
- Choose the hardware platform as **NodeMCU ESP8266** and set the connection type to **Wi-Fi**.

Blynk will generate an **Auth Token**, which is critical for linking the app with the hardware. Copy and save this token for future use

Step 2: Install Blynk Library in Arduino IDE

- **Action**: Open Arduino IDE and navigate to **Sketch > Include Library > Manage Libraries**.
- **Search for Blynk**: Type "Blynk" in the search bar and install the latest version of the Blynk library.

Link the Library: Include the necessary library in your code



Figur 1.11 Shows Uploading Codes in Blynk Firmwere

Step 3: Set Up the Hardware Components

- **Action**: Assemble the hardware components. The system will include:
  - **NodeMCU ESP8266** to serve as the microcontroller.
  - **DHT22 Sensor** to measure temperature and humidity.
  - **Soil Moisture Sensor** to monitor soil moisture levels.
  - **Relay Module** to control the water pump.
  - **16x2 LCD Display** to show sensor data.
  - **Water Pump** to irrigate the plants.

Connections: Wire each sensor and actuator to the corresponding pins on the NodeMCU based on the project schematic

Step 4: Write the Code for NodeMCU

- **Action**: Write the code to interface with the Blynk app and the sensors.
    - **Wi-Fi Setup**: Include the Wi-Fi credentials to connect the NodeMCU to the local Wi-Fi network.
    - **Blynk Setup**: Initialize the Blynk library using the Auth Token.
    - **LCD Initialization**: Set up the LCD to display data such as temperature, humidity, and soil moisture.
    - **Sensor Readings**: Write code to fetch sensor data from the DHT22 and soil moisture sensors.
    - **Relay Control**: Implement logic to control the water pump via the relay module based on soil moisture readings.
- **Code Example for Initialization**:

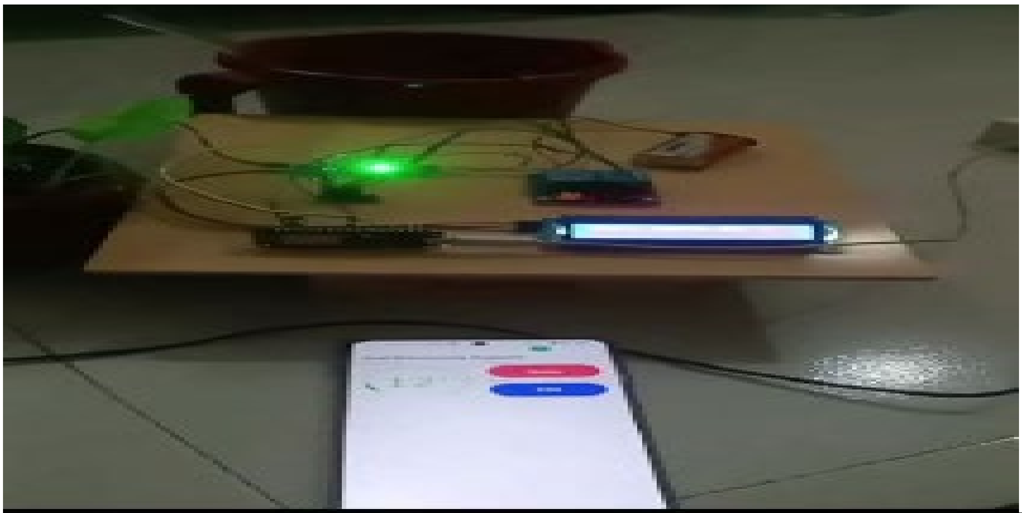
```
Blynk.begin(auth, wifi, ssid, pass);
dht.begin();  // Initialize DHT sensor
```

- **Sensor Data Display on LCD**: Implement a function to update the LCD with current readings.
- **Soil Moisture Control**: Add code to check the soil moisture level and trigger the water pump if moisture is below a threshold.

Step 5: Add Blynk Widgets for Monitoring and Control

- **Action**: In the Blynk app, set up virtual pins and widgets to allow real-time monitoring and control of the system.
    - **Virtual Pins**: Assign virtual pins in the Blynk app (e.g., V1 for soil moisture, V2 for pump control).
    - **Widgets**:
        - **Value Display**: Use for real-time updates of soil moisture, temperature, and humidity.
        - **Button**: Add a button to manually control the water pump (turn on/off).
        - **Slider**: Use a slider to set a custom soil moisture threshold for automatic irrigation.
- **Example Widget Setup**:
    - Virtual Pin **V1**: Display soil moisture reading.
    - Virtual Pin **V2**: Button to control the water pump.

Implementation & Screenshots:



Description:

A smart irrigation system optimizes water usage in agriculture by integrating sensors, communication protocols, and microcontrollers. This project ensures efficient water distribution, conserving resources while enhancing crop health.

Components

1. **NodeMCU ESP8266**: A microcontroller with built-in Wi-Fi, suitable for IoT applications.
   - **Connections**: Connects to sensors, the LCD screen, and the relay module for control and communication.
2. **16x2 LCD Screen**: Displays system status and sensor data.
   - **Connections**: Interfaces with NodeMCU through I2C or parallel communication.
3. **DHT22 Sensor**: Measures temperature and humidity.
   - **Connections**: Data pin connected to a GPIO pin on the NodeMCU with a pull-up resistor.
4. **Soil Moisture Sensor**: Determines water needs based on soil moisture levels.
   - **Connections**: Outputs analog or digital signals to a corresponding pin on the NodeMCU.
5. **2-Relay Module**: Controls actuators like water pumps and valves.
   - **Connections**: IN1 and IN2 connected to NodeMCU GPIO pins; relays control the water pump and other devices.
6. **Water Pump**: Distributes water for irrigation.
   - **Connections**: Powered via the relay module, which is controlled by the NodeMCU.

Functionality

Sensors collect field data, processed by the microcontroller to determine irrigation needs. If required, pumps and valves are activated. Communication protocols enable remote monitoring and control, ensuring automation and efficiency.

Benefits

- **Water Conservation**: Reduces waste by targeted irrigation.
- **Improved Crop Yield**: Maintains optimal soil moisture.
- **Automation**: Minimizes manual intervention.
- **Remote Access**: Enables system management from anywhere.

This sustainable solution addresses water scarcity challenges, providing scalable benefits for modern agriculture.

System Sample Data Table

Temperature	Soil Moisture Sensor Value	Irrigation Status(On/Off)
20%	LOW	ON
40%	LOW	ON
60%	MEDIUM	ON
80%	HIGH	OFF
95%	HIGH	OFF

Description:

The sample data table illustrates how the smart irrigation system responds to varying temperature and soil moisture levels. At lower temperatures and soil moisture values, the system identifies a need for irrigation and turns it ON. As the soil moisture increases to medium levels, irrigation continues to ensure optimal conditions. When the soil moisture reaches high levels, indicating sufficient water, the system automatically turns irrigation OFF to conserve water. This dynamic response ensures efficient water management based on real-time conditions.

Discussion

ADVANTAGES AND DISADVANTAGES ADVANTAGES

Prevents Disease and Weeds

Specialized drip irrigation systems direct water specifically to each plant's root ball, rather than sprinkling the entire garden like a typical rainstorm. As a result, surrounding weed seeds cannot germinate, so you'll have less weeding to do. Water at the roots also prevents leaf diseases caused by standing droplets on the foliage.
Because the water does not strike the leaves or flowers, blight diseases have no chance of proliferating.

Conserves Water and Time

Hand watering with a hose or watering can takes substantial time and early morning and evening watering rituals take away from family and work. Both drip and sprinkler irrigation systems have timers that can be preset for daily or weekly watering so you do not need to monitor the watering because the timer shuts the water off when it has finished. Your water bill should be lower if the irrigation system is effective.

Preserves Soil Structure and Nutrients

Watering with a wide open garden hose may allow too much water to seep into the soil. As a result, nutrients leach out with the water runoff, leaving the plants with fewer nutrients available. The soil may also become compacted when you water with a hose. Plants may show signs of withering or root disease with suffocating, compacted soil, Using either drip or sprinkler irrigation produces smaller droplets helping to preserve nutrients and reducing soil compaction.

Gardening Flexibility

If you have a busy schedule, you'll appreciate being able to work in the garden at the same time as the plants are being watered. While one garden section is being watered, you can plant and prune in another area.

DISADVANTAGES

1. Agriculture being a natural phenomenon relies mostly on nature, and man predict or control nature let it be rain drought sunlight availability. pests control etc. So ever implementation loT system agriculture.
2. The smart agriculture need availability on internet continuously. Rural part of the developing countries did not fulfil this requirements. Moreover internet is slower.
3. Fault sensor or data processing engines can cause faulty l decisions which may lead to over use of water, fertilizers and other wastage of resources.
4. The smart farming based equipment require farmer to understand and learn the use of technology. This is the major challenge in adopting smart agriculture farming at large scale across the continues.
5. It also has some issues which have to be tracked properly in order to attain the full benefit of it.
6. The current loT systems are not scalable or reliable and the initial costs are highwhich the farmers cannot afford.

Discuss

Limitations

Solar Energy Dependence:

- Low Sunlight: Performance can be affected on cloudy or rainy days, especially in regions with limited sunlight.
- Battery Capacity: Limited storage can lead to power shortages during extended cloudy periods, affecting system operation.

Connectivity Issues:

- Wi-Fi Reliance: The system depends on Wi-Fi for remote monitoring, which may be unreliable in rural or remote areas with poor internet coverage.
- Remote Areas: Areas with no internet access would require additional components, like a cellular data module, increasing cost and complexity.

Sensor Accuracy:

- Moisture Sensor Variability: Soil composition and sensor drift can cause inaccuracies in moisture readings, potentially leading to over or under-irrigation.
- Environmental Influence: Temperature and humidity readings from the DHT22 sensor can be affected by placement and extreme conditions, influencing irrigation decisions.

Limited Monitoring Scope:

- Basic Parameters: Only soil moisture, temperature, and humidity are monitored; additional parameters like pH and nutrients are not considered.
- Lack of Predictive Features: The system reacts to real-time data but doesn’t predict future needs based on weather forecasts or historical trends.

Scalability and Expansion:

- Single Area Focus: Designed for smaller areas, scaling up to larger farms requires additional components and complexity.
- Sensor Types: Limited to basic sensors; additional sensors for soil nutrients or light would improve system versatility but at an added cost.

Maintenance Needs:

- Component Wear: Regular maintenance is necessary to keep components like solar panels, pumps, and sensors in optimal condition.
- Downtime Risks: Failure of any part, such as the pump or relay, can cause system downtime, affecting performance.

Cost Considerations:

- Initial Investment: High upfront costs due to solar panels, sensors, and pumps may be prohibitive for small-scale farmers.
- Long-Term Maintenance: Ongoing maintenance and periodic upgrades add to the total cost of the system.

Future Improvements

Improved Power Management:

- o Integrating more efficient solar panels and larger batteries to ensure reliable operation during extended cloudy periods, and optimizing energy storage systems for better performance in low-light conditions.

Better Connectivity Solutions:

- o Use of cellular data modules or satellite connections to enhance connectivity in remote areas with limited or no Wi-Fi access, ensuring seamless monitoring and control.

Enhanced Sensor Accuracy:

- o Incorporating calibration algorithms to improve the precision of soil moisture sensors and environmental sensors like the DHT22, and adding more sensors (e.g., soil pH, nutrient levels) to provide more comprehensive environmental data.

Predictive Analytics and AI Integration:

- o Integrating weather forecasts and using machine learning models to predict irrigation needs based on weather patterns and soil conditions, enabling more proactive and precise irrigation scheduling.

Scalability Enhancements:

- o Developing modular systems that can scale easily, with the ability to connect multiple pumps, sensors, and controllers, making the system adaptable for both small-scale and large-scale agricultural operations.

Cost Reduction Strategies:

- o Exploring affordable components and optimizing the design for cost-effectiveness, potentially with subsidies or partnerships to make the system accessible to smallholder farmers. Automated Maintenance Monitoring:
- o Implementing self-diagnosing systems that alert users to potential component failures (e.g., malfunctioning pumps or sensors) to reduce downtime and ensure continuous operation.

Conclusions

The automatic irrigation control using NODE MCU has been experimentally proven to work satisfactorily and we could successfully set the timer and managed to control the motor over time. This process not only records values of temperature and humidity it also controls the motor accordingly. Analysing the weather condition motor will automatically maintain water supply making it possible to maintain greenery without human intervention. Using this system as framework, the system can be expanded to include variousother options which could include mobile application control of motor and Wi-Fi controlled monitoring. These will expand the working capability and efficiency of this prototype. It can be implemented not in agriculture but in gardens in any places using the sprinkler concept. It has a vast scope when it is mixed with IOT

References

1. Duffie, J. A., & Beckman, W. A. (2013). *Solar Engineering of Thermal Processes*. Wiley.
2. Al-Karaghouli, A., & Kazmerski, L. L. (2013). Energy consumption and water production cost of conventional and renewable-energy-powered desalination processes. *Renewable and Sustainable Energy Reviews, 24*, 343-356.
3. Blynk Documentation. (n.d.). Retrieved from https://docs.blynk.io/
4. Arduino. (n.d.). Arduino IDE User Guide. Retrieved from https://www.arduino.cc/
5. Chandel, S. S., Naik, M. N., & Chandel, R. (2015). Review of solar photovoltaic water pumping system technology for irrigation and community drinking water supplies. *Renewable and Sustainable Energy Reviews, 49*, 1084-1099.
6. PVsyst. (n.d.). PVsyst Software. Retrieved from https://www.pvsyst.com/
7. Proteus Design Suite. (n.d.). Labcenter Electronics. Retrieved from https://www.labcenter.com/
8. Meah, K., Ula, S., & Barrett, S. (2008). Solar photovoltaic water pumping for remote locations. *Renewable and Sustainable Energy Reviews, 12*(2), 472-487.
9. Zou, P. X., & Zhao, G. (2018). Solar-powered irrigation system for arid regions: Design and analysis. *Journal of Irrigation and Drainage Engineering, 144*(3), 1-10.
10. Kumar, A., & Chaturvedi, S. (2016). IoT-based remote monitoring and control of smart irrigation systems. *International Journal of Engineering Research and Technology, 5*(3), 1-8.

This list includes foundational literature and resources utilized in the design, simulation, and development of the project.

Appendix A

```
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>

#include <DHT.h>
#define BLYNK_TEMPLATE_ID "TMPL64HIhU_kd"
#define BLYNK_TEMPLATE_NAME "plant watering system"
#define BLYNK_AUTH_TOKEN "eYJeyR8FGL0U6LS5LZMCYaHnofN-lLpT"
#include <BlynkSimpleEsp8266.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);

char auth[] = "uBTcT8j6LZTNPvcxkRg9Y";
char ssid[] = "JDE";
char pass[] = "par12345";

BlynkTimer timer;
bool Relay = 0;

DHT dht(D3, DHT22);
#define sensor A0
#define waterPump D5

void setup() {
  Serial.begin(9600);
  pinMode(waterPump, OUTPUT);
  digitalWrite(waterPump, HIGH);
  lcd.begin(16,2);
  lcd.backlight();
  dht.begin();

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  lcd.setCursor(1, 0);
  lcd.print("System Loading");
  for (int a = 0; a <= 15; a++) {
    lcd.setCursor(a, 1);
    lcd.print(".");
    delay(500);
  }
  lcd.clear();

  timer.setInterval(100L, DHT22sensor);
  timer.setInterval(100L, soilMoistureSensor);
}

//Get the button value
BLYNK_WRITE(V7) {
  Relay = param.asInt();

  if (Relay == 1) {
    digitalWrite(waterPump, LOW);
```

```
  } else {
    digitalWrite(waterPump, HIGH);
  }
}

//Get the soil moisture values
void soilMoistureSensor() {
  int value = analogRead(sensor);
  value = map(value, 0, 1024, 0, 100);
  value = (value - 100) * -1;

  Blynk.virtualWrite(V8, value);
  lcd.setCursor(0, 0);
  lcd.print("Moisture :");
  lcd.print(value);
  lcd.print(" ");
}

void DHT22sensor() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  lcd.setCursor(0, 1);
  lcd.print("T:");
  lcd.print(t);

  lcd.setCursor(8, 1);
  lcd.print("H:");
  lcd.print(h);
}

void loop() {
  Blynk.run();  //Run the Blynk library
  timer.run();  //Run the Blynk timer
}
```