VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590 018

A REPORT ON MINI-PROJECT WORK

IoT BASED WATER FILLING MACHINE USING ARDUINO

Submitted in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS & COMMUNICATION ENGINEERING

Mini Project Associates

ANKITHA L	**4BD22EC010**
ANUSHA P R	**4BD22EC012**
ASHWATH B S	**4BD22EC014**
BHAVANA T E	**4BD22EC016**

Mrs. Vanishree H V
M Tech
Mini Project Guide

Dr. G. S. SUNITHA
MTech. (DEAC), Ph.D., MISTE, FIETE, FIE.
Program Coordinator

Dr. H.B. ARAVIND
B.E.,M.E., Ph.D., MISTE, FIEE.
Principal

Bapuji Educational Association®

Bapuji Institute of Engineering and Technology

Davangere-577 004

Department of Electronics & Communication Engineering

2024-2025

Bapuji Educational Association®

BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY DAVANGERE, KARNATAKA – 577004

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Certificate

Certified that the Mini-Project work entitled **“IoT based Water filling machine using Arduino”** carried out by, Ms. ANKITHA L. USN 4BD22EC010, Ms. ANUSHA P R. USN 4BD22EC012, Mr. ASHWATH B S. USN 4BD22EC014, Ms. BHAVANA T E. USN 4BD22EC016, bonafide students of this institution submitted in partial fulfillment for the award of degree of **Bachelor of Engineering in Electronics & Communication** by **VisvesvarayaTechnological University**, Belagavi during the academic year **2024-25**. It is certified that all corrections / suggestions indicated for continuous internal evaluation have been incorporated in the report deposited in the department library. The report has been approved as it satisfies the academic requirements in respect of mini-Project Work prescribed for the said degree.

Mrs. VANISHREE H V
M.Tech
Mini-Project Guide

Dr. G.S. Sunitha
MTech. (DEAC), Ph.D., MISTE, FIETE., FIE
Program Coordinator

Dr. H.B. Aravind
M.E., (Env. Eng.) Ph.D., FIE., MISTE., FIEE.
Principal

Internal Evaluation	Name	Signature with Date
1. Project Guide	Mrs. Vanishree H V	………………………..
2. Faculty Member		………………………..
3. Program Coordinator	Dr. G. S. Sunitha	………………………....

VISION OF THE INSTITUTE

To be a center of excellence recognized nationally and internationally, in distinctive areas of engineering education and research, based on a culture of innovation and excellence.

MISSION OF THE INSTITUTE

BIET contributes to the growth and development of its students by imparting a broad-based engineering education and empowering them to be successful in their chosen field by inculcating in them positive approach, leadership qualities and ethical values.

VISION OF THE DEPARTMENT

To be in the forefront in providing quality technical education and research in Electronics & Communication Engineering to produce skilled professionals to cater to the challenges of the society.

MISSION OF THE DEPARTMENT

M1. To facilitate the students with profound technical knowledge through effective teaching learning process for a successful career.

M2. To impart quality education to strengthen students to meet the industry standards and face confidently the challenges in the program.

M3. To develop the essence of innovation and research among students and faculty by providing infrastructure and a conducive environment.

M4. To inculcate the student community with ethical values, communication skills, leadership qualities, entrepreneurial skills and lifelong learning to meet the societal needs.

MINI PROJECT WORK (BEC586)

Course Learning Objectives:

This course will enable us to:

- Understand and analyze an engineering problem.
- Acquire technical knowledge and collect the information.
- Enhance communication, technical presentation and report preparation skills.
- Provide an opportunity to exercise the creative and innovative ideas in group.

ABSTRACT

This project introduces a cutting-edge solution for efficient and intelligent water management, integrating advanced automation and IoT technologies. Designed to optimize resource usage and enhance user convenience, The system offers a seamless blend of precision, sustainability, and connectivity. By leveraging real-time monitoring, remote control capabilities, and dynamic data visualization, the system ensures uninterrupted water supply while minimizing waste and promoting sustainable practices. Adaptable to a wide range of applications, from domestic to industrial and agricultural settings, this project represents a step forward in modern liquid management. It prioritizes reliability, accuracy, and user-centric design, providing a robust platform for continuous supply and smart water management. The integration of smart monitoring tools and IoT connectivity underscores the potential for innovation in tracking global challenges related to resource efficiency and sustainability. This project represents a significant contribution to the future of intelligent water management systems.

ACKNOWLEDGEMENTS

We would like to acknowledge the help and encouragement given by various people during the course of this Mini project.

We are deeply indebted and very grateful to the invaluable guidance given by project guide **Mrs.Vanishree H V,** Assistant professor during this Mini-Project work.

We would like to thank Mini-Project coordinator **Mrs. Suma K G**, Assistant Professor for the kind co-ordination during the course of the work.

We would like to express our sincere gratitude to **Dr. G S Sunitha**, Professor and Head, Department of Electronics & Communication Engineering, BIET, Davangere, for her kind support, guidance and encouragement throughout the course of this work.

We are thankful to our beloved principal **Dr. H B Aravind** for providing excellent academic climate.

We would like to thank all the teaching and non-teaching staff of Dept. of E&CE for their kind co-operation during the course of the work. The support provided by the college and departmental library is greatly acknowledged.

Finally, we are thankful to our parents and friends, who helped us in one way or the other throughout our project work.

Project Associates

Ankitha L

Anusha P R

Ashwath B S

Bhavana T E

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Efficient resource management is essential in addressing the challenges of modern living, and water, as one of the most precious resources, demands innovative solutions. This project presents a forward-thinking approach to water management by combining advanced automation with the power of IoT. Designed for precision and adaptability, it integrates smart technologies to offer a seamless solution for monitoring, control, and efficient usage of water resources across diverse applications.

The project emphasizes sustainability by reducing waste and ensuring continuous availability through intelligent resource management. By leveraging real-time data and remote access capabilities, it transforms traditional water systems into dynamic and responsive solutions. This shift not only enhances convenience but also empowers users to take control of their water usage, paving the way for smarter living and better resource conservation.

At its core, this system represents a fusion of technological innovation and practicality. Its ability to adapt to various settings, whether in households, industries, or agriculture, showcases its versatility and relevance in today's world. The inclusion of IoT-based monitoring and control further highlights its modern approach, aligning with global trends of integrating smart technologies into everyday life.

It's a vision for a future where water is managed responsibly, efficiently, and intelligently. By addressing key challenges in water management, With this solution, we take a step closer to creating a sustainable, connected, and innovative world.

CHAPTER 2
LITERATURE SURVEY

PAPERS	TITLE	AUTHOR	METHODOLOGY	COMPONENTS
PAPER 1	Automated water bottle filling system	Abdul Faimeed, Ambedkar Gajanan, Adhav Atish	System starts by moving the conveyor belt, which stops when IR sensor detects a bottle at the filling point, triggering the pump to fill bottle for a user-set time. Once filling, it resumes moving to the next bottle, and it stops until bottle is removed.	LCD, DC gear, DC submersible Pump, IR sensors, Relay module.
PAPER 2	Autonomous water tank filling system using IOT	S.Nalini durga, M.Ramakrishna, Dayanandam.	Arduino Uno connected sensors to monitor the water level and control motor for filling the tank automatically.The system continuously checks for waterflow, power availability sending alerts.	Arduino Uno, Flow Sensor, DC Motor, Ultrasonic sensor, LCD, SolenoidValve, Cloud, GSM shield
PAPER 3	IOT Based bottle filling system using PLC	R.Sureshkumar, R. Bharath, Karthikeyan, S.Suji Prasad.	PLC with IoT to automate bottle filling process, where Sensors detect bottle's position on a conveyor belt, and the PLC controls the filling based on user-defined volumes. This system is monitored and managed remotely.	PLC-57120, PC, UBIDOTS, IOT2040gateway, Bottlefilling system.

CHAPTER 3
OBJECTIVES

- To develop an automated water filling system that ensures accurate liquid dispensing.
- To create a smart system that stops the pump automatically once the desired quantity is fulfilled, preventing wastage.
- To integrate continuous water level monitoring using an ultrasonic sensor for effective tank management.
- To enable remote control and refill activation through the Blynk app for convenient operation.
- To display real-time water level data and dispensing information on ThingSpeak for ongoing monitoring.

CHAPTER 4
METHODOLOGY

4.1 Block Diagram

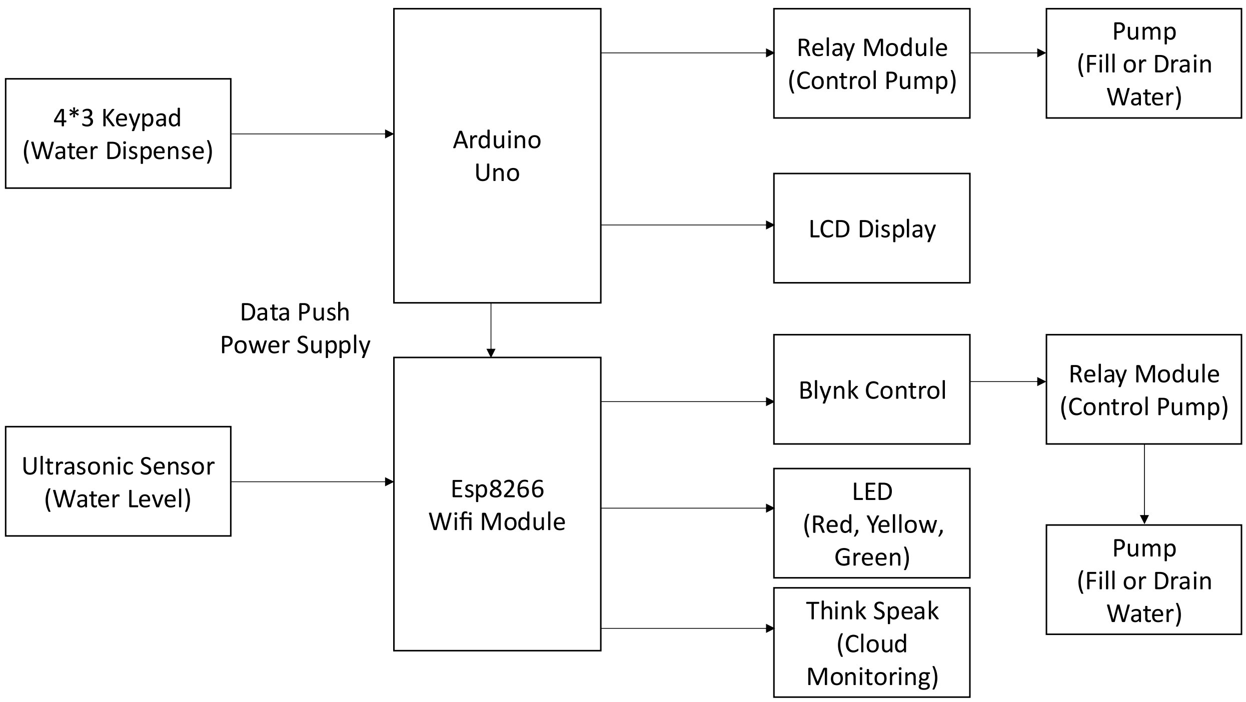


Figure 4.1 Block Diagram

Methodolody:

- User Input via Keypad: The user enters the desired amount of water using a keypad connected to the Arduino.
- Water Dispensing Control: The Arduino processes the input and starts the pump to dispense the water.
- Flow Sensor for Accuracy: A flow sensor measures the amount of water dispensed and sends data to the Arduino.
- Pump Shutoff: Once the flow sensor detects the target amount of water, the Arduino stops the pump automatically.
- Ultrasonic Sensor for Tank Monitoring: An ultrasonic sensor measures the water level in the tank to check if it's low.
- Remote Control via Blynk App: If the tank level is low, the Blynk app, allowing remote refilling.
- Data Monitoring on ThingSpeak: Real-time water levels are uploaded to ThingSpeak for remote monitoring and analysis.

CHAPTER 5
HARDWARE DESCRIPTION

5.1 Hardware Description
5.1.1 Arduino uno

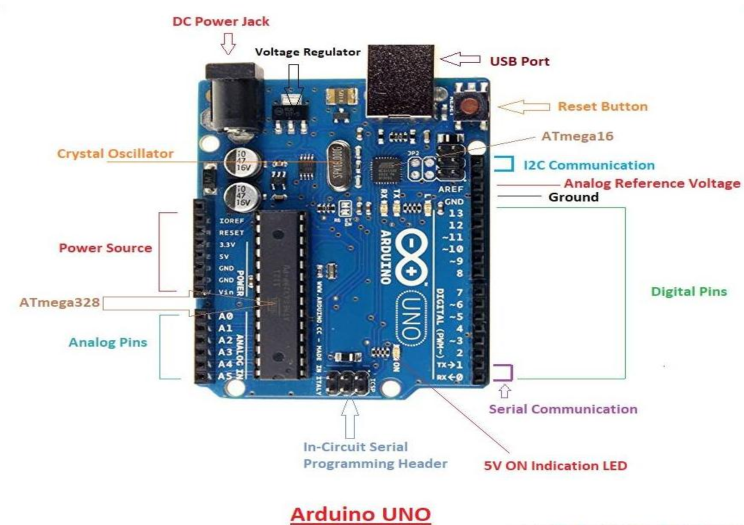


Figure 5.1.1 Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the ATmega328P microchip. It is one of the most widely used boards in Arduino-based projects due to its simplicity, versatility, and availability of extensive documentation and libraries.

Key Features:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Digital I/O Pins: 14 (of which 6 can be used as PWM outputs)
- Analog Input Pins: 6
- Flash Memory: 32 KB (of which 0.5 KB is used by the bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB

Functionalities:

- Acts as the central controller for the entire system.
- Processes input from the keypad, flow sensor, and ultrasonic sensor.
- Controls the pump via the relay module based on user input and water level.
- Handles communication with the ESP8266 for remote control via Blynk app and ThingSpeak.
- Displays relevant information on the I2C LCD module.

5.1.2 Ultrasonic Sensor

Figure 5.1.2(a) Ultrasonic sensor

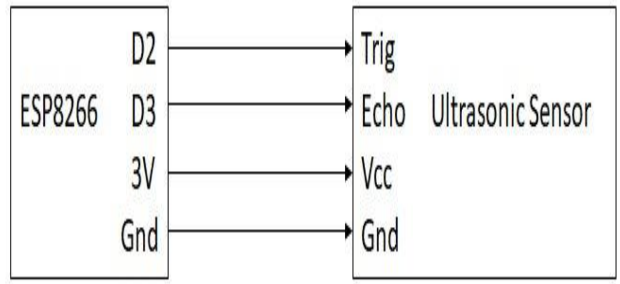


Figure 5.1.2(b) Interfacing of ESP8266 with Ultrasonic sensor

The HC-SR04 ultrasonic sensor is an affordable and widely used distance-measuring device. It operates by emitting sound waves at a frequency of 40 kHz and measures the time it takes for the sound to bounce back after hitting an object. The time of flight is used to calculate the distance between the sensor and the object (in this case, the water surface).

Key Features:

- Operating Voltage: 5V
- Measuring Range: 2 cm to 400 cm
- Accuracy: ±3mm
- Working Frequency: 40 kHz
- Output Type: Digital (Trigger and Echo Pins)
- Power Consumption: 15mA (standby), 25mA (active)

Functionalities:

- Continuously measures the distance from the sensor to the water surface in the tank.
- Determines the water level in the tank by calculating the time taken for the ultrasonic waves to return.
- Triggers the refill process when the water level falls below a certain threshold.

5.1.3 I2C LCD

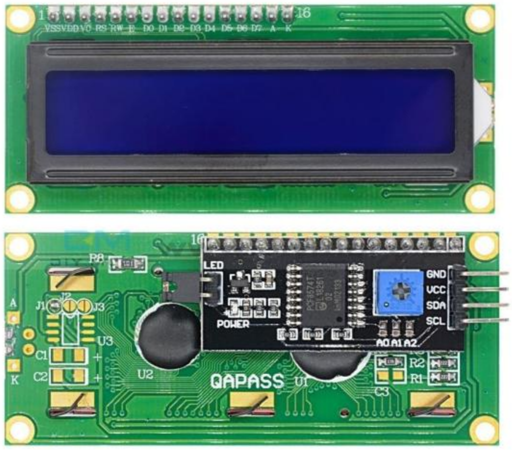


Figure 5.1.3(a) I2C LCD

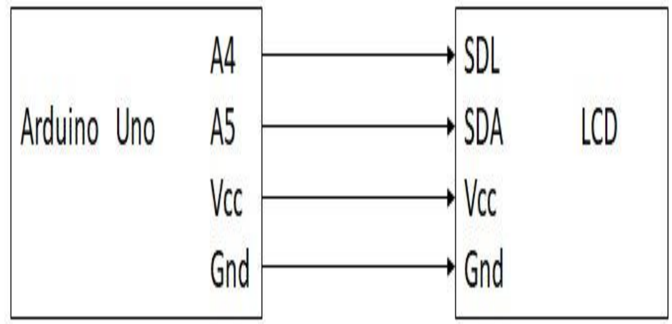


Figure 5.1.3(b) Interfacing of Arduino Uno with LCD

The I2C LCD is a compact display screen that utilizes the I2C communication protocol to connect with microcontrollers like Arduino. This protocol allows for efficient data transfer between the device and the Arduino, using just two data lines, minimizing the need for multiple wiring connections. This is particularly useful in projects with limited input/output (I/O) pins.

Key Features:

- Display Type: Typically available as 16x2 or 20x4 LCD screens for displaying text or numbers.
- Communication Protocol: Uses I2C (Inter-Integrated Circuit) for communication, reducing wiring complexity.
- Operating Voltage: 5V for compatibility with Arduino.

Functionalities:

- Displays the user's input for the desired amount of water to be dispensed.
- Shows the amount of water dispensed in real time.
- Continuously updates and displays the current water level in the tank.
- Uses I2C communication protocol, requiring only two wires (SCL and SDA).
- Reduces the need for multiple wiring connections, saving Arduino I/O pins.

Figure 5.1.3(c) LCD Display

5.1.4 Relay module

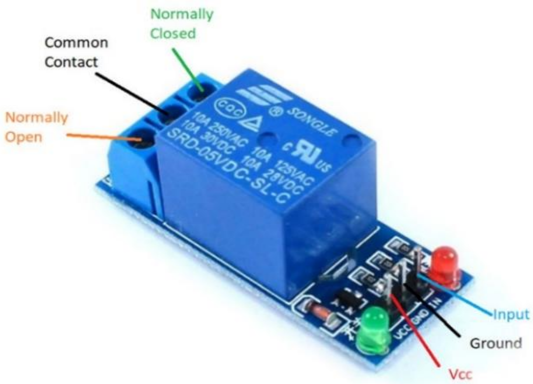


Figure 5.1.4(a) Relay module

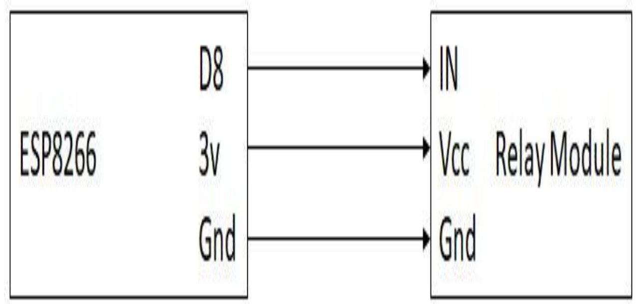


Figure 5.1.4(b) Interfacing of ESP8266 with Relay module

A relay module is a versatile electronic component used to control high-voltage or high-current devices, such as pumps, motors, through a low-voltage signal from a microcontroller. The relay operates as an electromechanical switch, using an internal coil to open or close a circuit.

Available in different configurations like SPDT (Single Pole Double Throw) or DPDT (Double Pole Double Throw), relays are suitable for various applications, offering flexibility in controlling single or multiple devices simultaneously.

Key Features:

- Operating Voltage: Typically 5V for microcontroller compatibility.
- Current Handling: Supports currents up to 10A for high-power devices
- Electrical Isolation: Provides complete isolation between the control circuit and high-power device
- Compact Design: Designed for easy integration into various projects.

Functionalities:

- Controls the high-power water pump using a low-power signal from the Arduino.
- Acts as an interface between the Arduino and the water pump, ensuring electrical isolation.
- Receives control signals from the Arduino based on water level readings.

5.1.5 4x3 Keypad

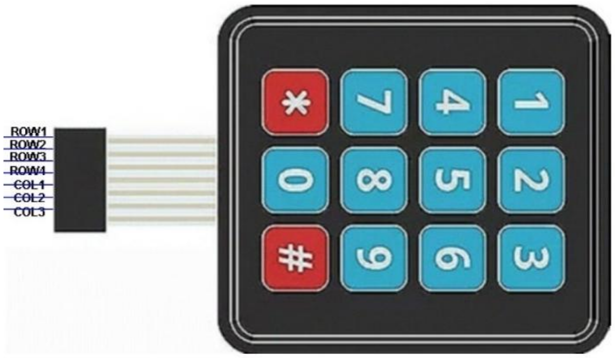


Figure 5.1.5(a) 4x3 Keypad

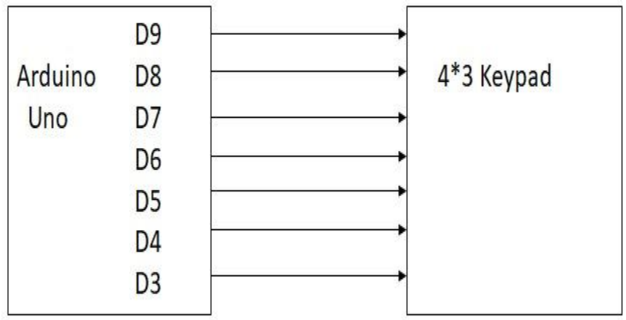


Figure 5.1.5(b) Interfacing of Arduino Uno with 4x3 Keypad

The 4x3 matrix keypad is a widely used input device designed to provide an intuitive and efficient interface for user interaction, offering a compact and reliable solution for entering numeric data. The keys are typically made of durable material with a tactile response, ensuring smooth operation and long-lasting performance. The keypad works by completing an electrical circuit whenever a button is pressed. This circuit generates a signal that is read and processed by a microcontroller, such as the Arduino, making it an essential component in embedded systems and automation projects.

Key Features:

- Compact Design: Offers 12 buttons in a 4-row by 3-column layout.
- Efficient Pin Usage: Requires only seven pins for operation.
- Flexible Integration: Compatible with most microcontrollers, including Arduino.
- Operating Voltage: Standard 5V, ensuring compatibility with low-power devices.

Functionalities:

- Allows users to input the desired quantity of liquid to be dispensed.
- Sends a signal to the Arduino when a button is pressed for further processing.
- Minimizes I/O pin usage by operating through a 4x3 matrix configuration.
- Ensures reliable and tactile button response for accurate data input.

5.1.6 Node MCU ESP 8266

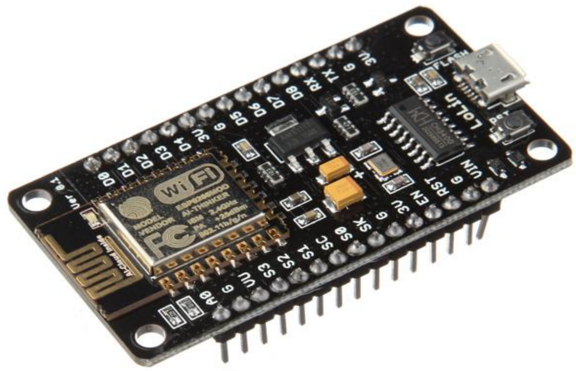


Figure 5.1.6 Node MCU ESP8266

The NodeMCU ESP8266 V3 is a powerful and versatile Wi-Fi module that integrates a microcontroller and a Wi-Fi chip, enabling seamless wireless connectivity.

The NodeMCU ESP8266 V3 acts as the bridge between the system and the internet, enabling remote control and real-time monitoring of water levels. By connecting to a Wi-Fi network, it transmits data from the sensors to cloud platforms like ThingSpeak, where users can view water level updates. It also facilitates remote pump control through the Blynk app, enhancing convenience and user interaction.

Key Features:

- Processor: ESP8266 chip with 80 MHz clock speed and Tensilica L106 architecture.
- Connectivity: Built-in Wi-Fi for wireless communication.
- Operating Voltage: 3.3V (with 5V input supported via onboard voltage regulator).
- Flash Memory: 4MB, ideal for storing program code and data.
- GPIO Pins: Multiple pins for peripheral connectivity.
- Programming: Compatible with Arduino IDE for easy coding and debugging.
- Compact Design: Fits easily into IoT projects requiring limited space.

Functionalities:

- Connects the system to Wi-Fi for remote control and monitoring.
- Sends real-time water level data to the ThingSpeak platform for visualization and analysis.
- Enables remote activation of the pump through the Blynk app when the tank water level is low.
- Acts as the communication hub, linking the Arduino with cloud services.
- Ensures efficient data transfer and control operations, enhancing system automation.
- Provides a seamless user experience by integrating IoT capabilities into the water filling machine.

5.1.7 DC Submersible Pump

Figure 5.1.7 DC Submersible Pump

The 3V to 9V mini DC submersible pump is a compact and efficient water pump designed for small-scale liquid transfer applications. It operates on a low DC voltage, making it ideal for battery-powered or microcontroller-based systems. This pump is lightweight, energy-efficient, and easy to integrate into automated systems. Its ability to work while submerged in water adds to its versatility, ensuring reliable operation in water-related projects.

In the Water Filling Machine project, the mini DC submersible pump is responsible for transferring water from the tank to the dispensing outlet. Controlled by the relay module and monitored by the flow sensor, the pump ensures precise liquid dispensing according to user input. Additionally, the pump plays a crucial role in maintaining a continuous water supply by refilling the tank.

Key Features:

- Operating Voltage: 3V to 9V DC, compatible with microcontroller systems like Arduino.
- Current Rating: Typically 0.1A to 0.3A, depending on voltage.
- Flow Rate: Up to 200 liters per hour (varies with voltage).
- Material: Durable plastic housing suitable for submersible use.
- Compact Size: Easy to install in space-constrained systems.
- Noise Level: Low noise operation, suitable for indoor use.
- Application: Ideal for small-scale water transfer, aquarium circulation, or irrigation systems.

Functionalities:

- Pumps water for accurate dispensing based on user-specified quantity.
- Works in coordination with the flow sensor and Arduino to control the output volume.
- Responds to commands from the relay module to start or stop operation.
- Refills the water tank when the ultrasonic sensor detects low water levels.
- Operates reliably in submerged conditions, ensuring safe and efficient performance.
- Ensures a steady water flow, contributing to the overall automation and precision of the system.

5.1.8 LED

Figure 5.1.8(a) LED

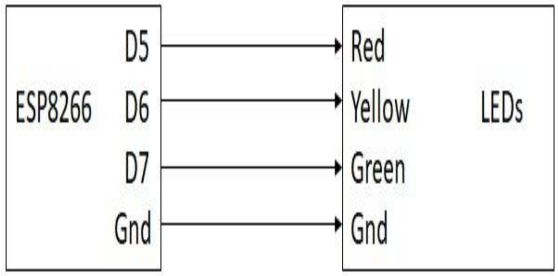


Figure 5.1.8(b) Interfacing ESP8266 with LED's

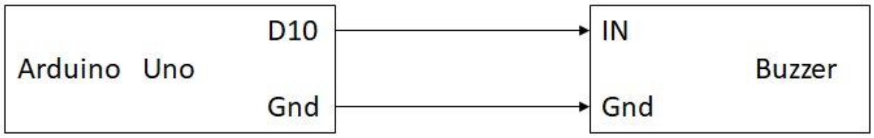
The LED is a simple yet effective visual indicator used in electronic systems to communicate status or alerts.

Functionalities:

- Indicates system power status or readiness for operation.
- Signals active water dispensing when the pump is running.
- Alerts users visually when the water level in the tank is low, Medium or High.

5.1.9 Piezo Buzzer

Figure 5.1.9 Piezo Buzzer



The piezo buzzer is an audio signaling device that generates sound when an electrical signal is applied to its piezoelectric element. It is widely used in systems requiring auditory feedback or alerts.

Functionalities:

- Issues a warning sound if the tank's water level is critically low.
- Enhances user interaction by alerting users to system updates or critical conditions.
- Works in tandem with the LED to ensure both visual and audio feedback.

5.2 Circuit Diagram

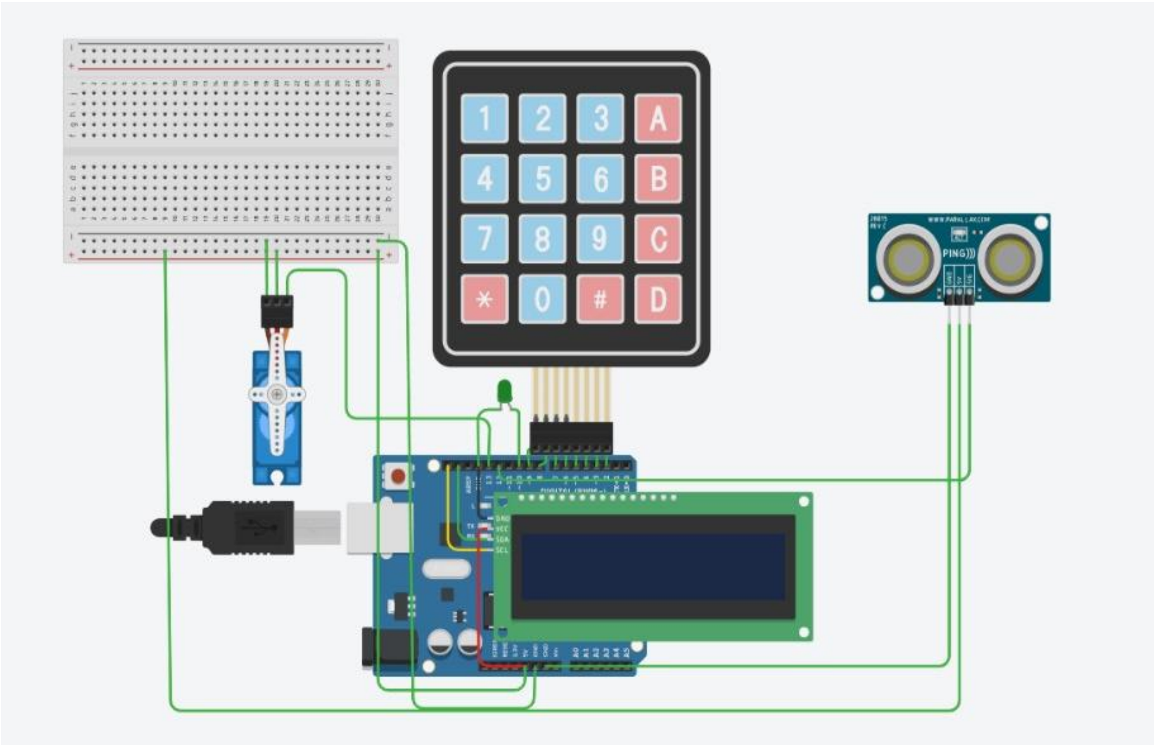


Figure 5.2 Circuit Diagram

The above figure 5.2 illustrates the circuit diagram for an IoT-based an IoT-based water filling machine using an Arduino. Here's a breakdown of the circuit components:

1. Arduino Board: The central microcontroller that processes inputs and controls outputs.
2. Keypad: A 4x4 keypad module connected to the Arduino to input user commands.
3. Ultrasonic Sensor: Measures the water level in the tank.
4. LCD Display: Displays the water level and status of the machine.
5. Piezo Buzzer: Provides audio alerts when the water level is critically low.
6. Breadboard: Used for connecting components without soldering.
7. LED: Works with the piezo buzzer to provide visual feedback.

The circuit uses connections between the Arduino's digital and analog pins to interact with these components, enabling automated water-level monitoring and control.

CHAPTER 6
SOFTWARE DESCRIPTION

6.1 Software Description

6.1.1 Arduino IDE

Figure 6.1.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is more than just a software tool; it is a playground for innovation, creativity, and technological exploration. Specifically designed to program Arduino boards, the IDE empowers developers, hobbyists, and innovators to bring their ideas to life with ease. Its simplicity, combined with robust capabilities, makes it a universal platform for building dynamic and interactive projects.

The IDE offers an intuitive, minimalist interface where users can write, compile, and upload code effortlessly. It supports a streamlined version of C/C++, ensuring an easy learning curve for beginners while offering advanced features for experts. With a single click, you can transform your written logic into actionable commands for Arduino hardware. Its open-source nature encourages collaboration, fostering a thriving community of developers who share libraries, projects, and expertise.

Key Features:

- Code Compilation & Uploading: A seamless process to compile code and upload it directly to Arduino boards via USB.
- Extensive Library Support: Prebuilt libraries for sensors, actuators, and communication modules simplify complex programming tasks.
- Cross-Platform Compatibility: Runs effortlessly on Windows, macOS, and Linux, providing flexibility for developers.
- Serial Monitor: Real-time communication between the Arduino and your computer for debugging and monitoring.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor. Arduino IDE is used to control all the input and output devices that are connected to the Arduino uno with programming.
For code, the following libraries are installed:

<ESP8266WiFi.h>
<BlynkSimpleEsp8266.h>
<ThingSpeak.h>
<ESP8266HTTPClient.h>

- ESP8266WiFi.h: Enables Wi-Fi connectivity for the ESP8266.
- BlynkSimpleEsp8266.h: Connects the ESP8266 to the Blynk app for IoT control.
- ThingSpeak.h: Sends data to ThingSpeak for cloud storage and analysis.
- ESP8266HTTPClient.h: Allows the ESP8266 to send HTTP requests to web services.

6.1.2 BLYNK CLOUD

Figure 6.1.2 Blynk Icon

1. Blynk Cloud is a cloud-based platform designed to simplify the development and management of Internet of Things (IoT) projects. It offers a suite of tools for connecting hardware devices to mobile applications and the internet, allowing for remote control, data visualization, and project management.

   Here's a breakdown of its key features:

2. **Device Management:**
   a. Blynk Cloud facilitates the connection and management of various IoT hardware devices, including popular boards like Arduino, Raspberry Pi, and ESP8266.
   b. It provides a user-friendly interface for configuring devices and associating them with specific projects.
3. **Mobile App Development:**
   a. Blynk offers a visual programming environment for building user interfaces on smartphones and tablets. This drag-and-drop interface allows users to create custom dashboards featuring various widgets like buttons, sliders, graphs, and displays.
   b. These widgets can be linked to specific hardware functions, enabling users to control devices remotely from their mobile apps.
4. **Data Visualization and Storage:**
   a. Blynk Cloud enables the visualization of sensor data collected from connected devices. Users can create graphs and charts to monitor sensor readings in real-time or analyse historical data.
   b. The platform might offer limited data storage capabilities, allowing users to retain sensor data for later analysis.
5. **Real-time Communication:**
   a. Blynk Cloud facilitates real-time communication between devices and mobile apps. Users can trigger actions on devices based on app interactions or send commands based on sensor readings.

6. **Security:**
    a. Blynk Cloud implements security measures to protect user data and device communication. This might include features like user authentication and secure communication protocols.

Benefits of using Blynk Cloud:

- **Simplified Development:** Blynk Cloud's visual programming approach simplifies the creation of mobile apps for IoT projects, even for users with limited coding experience.
- **Rapid Prototyping:** The platform allows for quick prototyping of IoT projects without requiring extensive coding efforts.
- **Remote Control and Monitoring:** Users can remotely control and monitor their IoT devices from anywhere with an internet connection.
- **Scalability:** Blynk Cloud can support a range of projects, from simple hobbyist applications to more complex industrial deployments (depending on the chosen plan).

Limitations:

- **Limited Features:** The free tier of Blynk Cloud might have limitations on data storage, connected devices, or project complexity.
- **Cloud Dependence:** Projects rely on Blynk's cloud infrastructure for communication, which could introduce a single point of failure if the service becomes unavailable.
- **Security Considerations:** While Blynk implements security measures, users should be aware of potential security risks associated with cloud-based platforms.

Overall, Blynk Cloud offers a valuable tool for developers and hobbyists to create and manage IoT projects. Its user-friendly interface and suite of features make it a popular choice for rapid prototyping and remote-control applications.

6.1.3 ThingSpeak Cloud

Figure 6.1.3 ThingSpeak Icon

ThingSpeak is a cloud-based Internet of Things (IoT) analytics platform that provides a seamless environment for collecting, analyzing, and visualizing sensor data. Widely used for IoT projects, ThingSpeak allows devices to communicate and share data effortlessly. Its integration with MATLAB further empowers users to apply advanced data analytics and machine learning to their projects. The platform's user-friendly interface and real-time visualization capabilities make it a top choice for developers, researchers, and innovators aiming to harness the potential of IoT.

Key Features of ThingSpeak

- **Real-Time Data Collection**
  ThingSpeak allows IoT devices to send real-time data to the cloud via RESTful APIs. This enables continuous monitoring of parameters such as temperature, humidity, and, in the case of your project, water levels.
- **Data Visualization**
  One of the standout features of ThingSpeak is its ability to create interactive charts and graphs. Users can visualize trends, analyze patterns, and track real-time data on an intuitive dashboard.
- **Integration with MATLAB**
  ThingSpeak provides a direct interface with MATLAB, enabling users to perform complex calculations, predictive analytics, and machine learning on their IoT data. This powerful feature is especially useful for applications requiring data optimization and forecasting.
- **Custom Alerts and Notifications**
  The platform supports event-based triggers, allowing users to set up alerts when data crosses predefined thresholds. For instance, in your project, you can configure alerts for low water levels or excessive water usage.

- **Open-Source and Extensible**
  ThingSpeak is open-source, meaning users can customize its features to suit their specific needs. It also supports third-party integrations, making it a highly adaptable tool for IoT ecosystems.
- **Device Compatibility**
  The platform is compatible with a wide range of IoT devices and microcontrollers, including the ESP8266, used in your project. Its robust API ensures seamless communication between hardware and the cloud.

Functionalities :

ThingSpeak plays a crucial role in monitoring and managing water levels. Here's how:

- **Real-Time Monitoring**
  ThingSpeak receives water level data from the ultrasonic sensor via the ESP8266 module. This data is visualized on interactive graphs, providing an at-a-glance overview of the system's performance.
- **Data Logging**
  The platform logs historical data, allowing you to track trends in water consumption and tank refilling patterns. This data can be invaluable for optimizing system efficiency.
- **Alerts for Critical Levels**
  ThingSpeak can be configured to send notifications when water levels drop below or exceed a certain threshold, ensuring timely action and preventing operational interruptions.
- **Integration with IoT Ecosystem**
  As part of a larger IoT framework, ThingSpeak seamlessly integrates with the Blynk app and Arduino IDE, enabling a unified, automated solution for water management.

6.2 Flow Chart

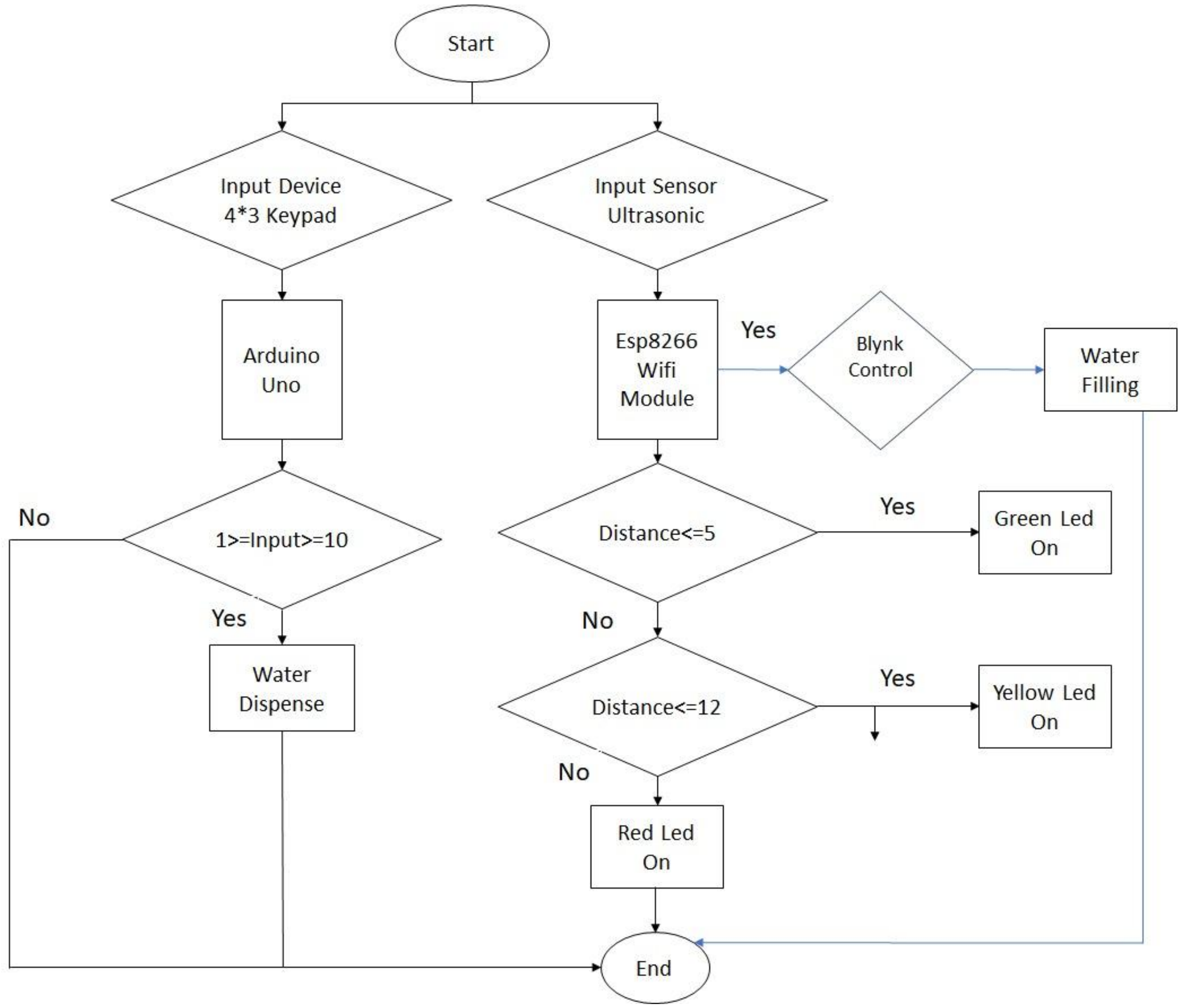


Figure 6.2 Flow chart

- **Start:** The process begins.
- **Input Sensor:** The system checks input from sensors, leading to branches in the flow.
- **Keypad (4x3):** The keypad interacts with the Arduino Uno.
- **Ultrasonic Sensor:** The ultrasonic sensor interfaces with the ESP8266 microcontroller.
- **ESP8266 Microcontroller:** Handles Blynk (manual control), connects to Thingspeak (cloud integration), and updates LED status.
- **End:** Marks the conclusion of the process.

CHAPTER 7
RESULT AND DISCUSSION

The IoT-based water-filling machine successfully integrates automation with IoT technology to provide a reliable and efficient solution for water level management.

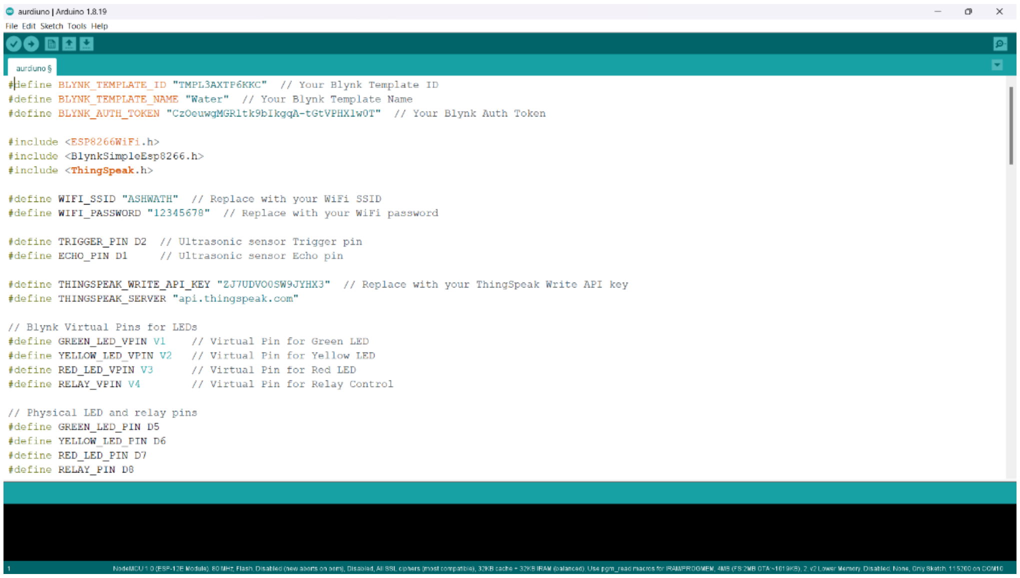


Figure 7.1.1 Arduino IDE software with Embedded C code and Initialization

- Automation: The water-filling mechanism is fully automated, reducing the need for manual intervention. The system leverages sensors to detect water levels and automatically controls the water flow, ensuring precision and reducing human errors. The automated process enhances efficiency and ensures the required water level is achieved without human intervention.

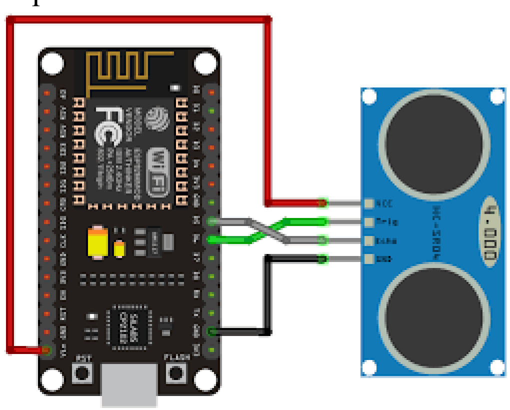


Figure 7.1.2 Interfacing of ESP8266 with Ultrasonic Sensor

- Real-Time Monitoring: Continuous monitoring of water levels ensures timely and accurate filling operations. Sensors transmit real-time data to the microcontroller, which analyzes the information and initiates necessary actions. This reduces water wastage and prevents overflow or underfill scenarios. Real-time monitoring also allows for prompt detection of issues, ensuring the system remains reliable and consistent.

- IoT Connectivity: Users are notified of system status and water levels via mobile applications or LCD displays, enhancing usability. IoT platforms like Blynk or MQTT enable remote monitoring and control, providing flexibility and convenience to users. Notifications about low water levels, system malfunctions, or completed tasks keep users informed at all times. The seamless integration of IoT ensures that users remain in control, even from remote locations.

- Energy Efficiency: The system only activates components when necessary, conserving energy. By optimizing the usage of pumps and valves, the system ensures minimal energy consumption, making it an eco-friendly solution. The energy-efficient design not only reduces operational costs but also contributes to environmental sustainability.

- Scalability: The system is highly scalable and can be adapted for various applications, including residential, industrial, and agricultural uses. This flexibility ensures that the machine can cater to a wide range of needs and scenarios.

- Low Maintenance Requirements: The design of the machine incorporates durable and reliable components, minimizing maintenance needs. Automated alerts for faults or service requirements ensure the system remains operational with minimal downtime.

- Cost-Effectiveness: By reducing water wastage, energy consumption, and manual labor, the IoT-based water-filling machine proves to be a cost-effective solution. The use of affordable IoT platforms further lowers the overall implementation cost.

- Environmental Impact: The machine promotes sustainable water management by preventing wastage and conserving energy. Its eco-friendly design makes it a valuable addition to efforts aimed at reducing environmental degradation.

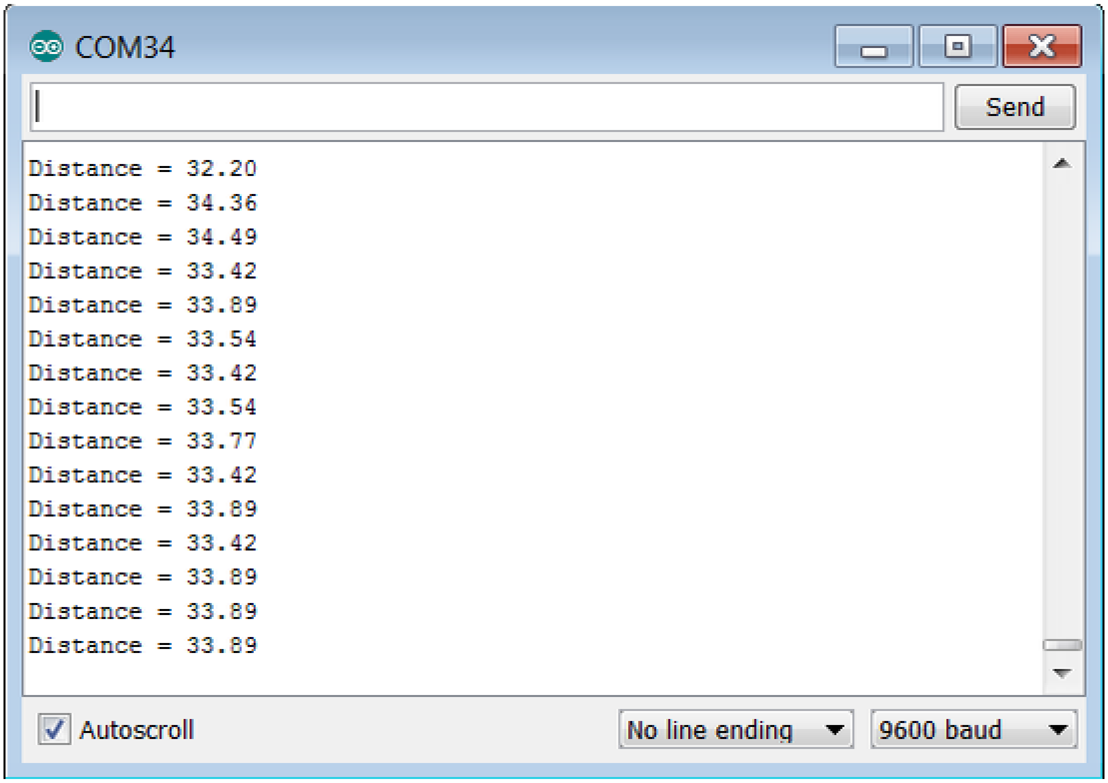


Figure 7.1.3 Serial Monitor Display

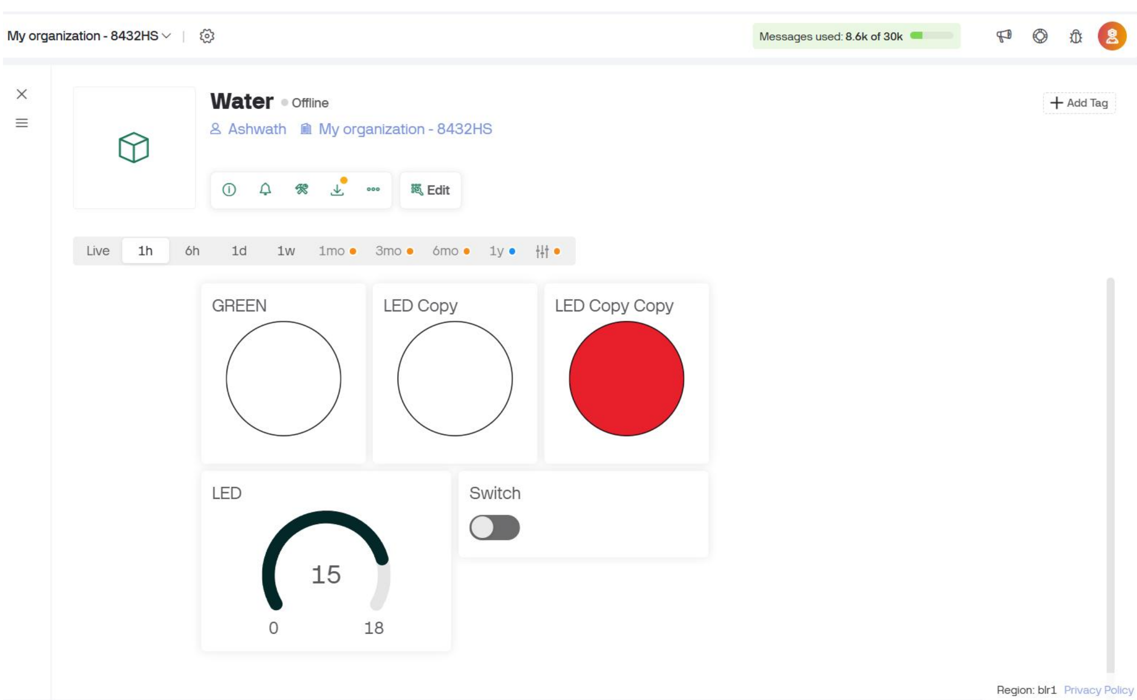


Figure 7.1.4 Blynk dashboard

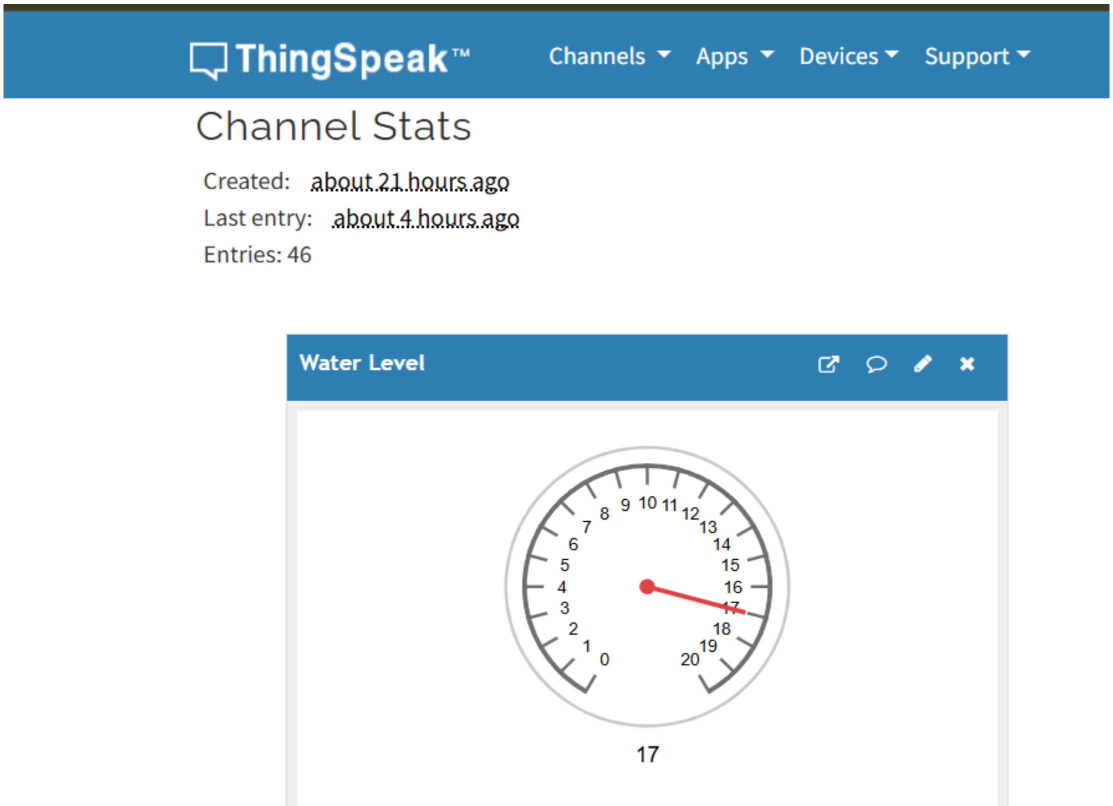


Figure 7.1.5 Thing Speak Channel

This project demonstrates the potential of IoT in improving everyday tasks by combining advanced technology with practical applications. The system's ability to provide real-time data and remote control addresses the inefficiencies of traditional water management methods. The automated, energy-efficient, and IoT-enabled design makes this water-filling machine a reliable and sustainable solution.

The IoT-based water-filling machine has applications across a variety of sectors, including domestic water management, industrial processes, and agricultural irrigation. Its scalability and flexibility allow it to be tailored to specific needs, making it a versatile solution for diverse water management challenges. By bridging the gap between traditional practices and modern technology, this system lays the groundwork for future advancements in resource management, setting a benchmark for smart, automated systems.

CHAPTER 8
APPLICATIONS

- **Industrial Fluid Dispensing**
  Used in industries for precise dispensing of liquids such as chemicals, oils, and other fluids to improve efficiency and reduce wastage.
- **Smart Home Automation**
  Ideal for automated water dispensing in household applications, including kitchen appliances, aquariums, and gardening systems.
- **Agriculture and Irrigation**
  Enables controlled irrigation systems by dispensing accurate amounts of water, promoting water conservation and better crop management.
- **Water Bottling Plants**
  Can be implemented in bottling plants for precise filling of water bottles, ensuring consistency and accuracy in packaging.
- **Laboratory and Research Applications**
  Useful for dispensing precise volumes of liquids in laboratories and research setups for experiments requiring accurate measurements.
- **Hotel and Hospitality Industry**
  Facilitates automated water dispensing in hotel kitchens or restaurants, streamlining operations and minimizing human intervention.
- **Public Water Dispensing Systems**
  Can be deployed in public water dispensing units to provide a measured and automated system for water distribution.
- **IoT-Based Water Resource Management**
  Integrates with IoT platforms like ThingSpeak to enable efficient management and monitoring of water resources in remote or urban areas.
- **Environmental Conservation Projects**
  Supports initiatives focused on conserving water by preventing overuse and ensuring optimal resource utilization.

ADVANTAGES

Main advantages are:

- **Precision in Liquid Dispensing:**
  The use of a flow sensor ensures accurate measurement and dispensing of liquid, eliminating wastage and enhancing efficiency.
- **Automation for Efficiency:**
  Automated pump operation stops dispensing when the desired quantity is fulfilled, reducing the need for constant supervision and manual intervention.
- **Continuous Monitoring:**
  An ultrasonic sensor provides real-time tank water level monitoring, enabling effective management of water resources.
- **Remote Accessibility:**
  Integration with the Blynk app allows users to control the system remotely, providing convenience and flexibility for tank refills.
- **IoT Integration:**
  ThingSpeak displays real-time data and logs historical trends, enabling informed decision-making and easy system diagnostics.
- **Resource Optimization:**
  Preventing overfilling or underfilling ensures optimal use of water resources, contributing to sustainability.
- **Scalability and Adaptability:**
  The system can be expanded to include multiple tanks or larger dispensing units, making it suitable for diverse applications such as agriculture, industries, and households.
- **User-Friendly Interface:**
  The keypad and LCD module offer a simple and intuitive way for users to interact with the system, reducing learning curves.
- **Cost-Effective Design:**
  Utilizing commonly available components like Arduino, ultrasonic sensors, and flow sensors makes the system affordable to build and maintain.
- **Improved Operational Safety:**
  The automated features reduce risks associated with manual handling, ensuring a safer and more reliable water management process.

LIMITATIONS

The limitations of IoT-based Water filling machine using Arduino:

- **Dependence on Accurate Flow Sensor:**
  The system relies on the accuracy of the flow sensor, and any calibration errors can affect the dispensed water quantity.
- **Limited Water Dispensing Capacity:**
  The pump and system may be limited by the size of the tank and the flow rate of the pump.
- **Power Supply Dependency:**
  The system requires a stable power supply for the Arduino and sensors, which could be disrupted in case of power failure.
- **Keypad Input Limitation:**
  Users must manually input water quantities, which could be impractical for larger quantities or multiple users.
- **Limited Sensor Range:**
  The ultrasonic sensor may have limitations in detecting water levels accurately if there is interference, such as tank shape or water surface turbulence.
- **Internet Dependency for Blynk and ThingSpeak:**
  Remote monitoring through Blynk and ThingSpeak requires internet connectivity, limiting use in areas without reliable internet access.
- **Maintenance of Components:**
  Sensors and the pump may require periodic maintenance or recalibration to maintain performance.
- **Complexity in Handling Large Volumes:**
  The system may face challenges when handling large volumes of water, as pumps and sensors might not be optimized for high-flow rates.
- **Limited to Water Only:**
  The system is designed for water, and using other liquids could affect the accuracy of the flow sensor and other components.
- **Reliance on Code Stability:**
  The performance of the system depends on the stability of the code; any bugs or errors in the programming could disrupt the functionality.

CONCLUSION

In conclusion, the Water Filling Machine marks a significant advancement in optimizing water usage through automation and smart technology. By integrating precise water dispensing with real-time monitoring, the system ensures efficient water usage and minimizes waste. The ability to remotely activate the pump via the Blynk app to refill the tank when the water level is low, along with the convenience of remote tracking, empowers users to manage their water resources effortlessly and effectively. It addresses the need for accurate water dispensing while also demonstrating the power of IoT in creating smarter, more sustainable systems.

As the world increasingly embraces automation, this project stands as a model for innovation in water management. It opens up endless possibilities for enhancing resource efficiency across homes, businesses, and industries. With future advancements like AI-driven predictions and seamless integration with smart homes, this system has the potential to play a pivotal role in shaping the future of sustainable water usage and conservation, ensuring that one of the world's most precious resources is managed more intelligently and responsibly.

FUTURE SCOPE

- **Multiple Liquid Support:** The system can be expanded to handle various liquids, such as juices and oils, by adjusting the flow sensor and pump for different viscosities and properties.
- **Smartphone Control:** Users will be able to control and monitor the water filling system remotely through a dedicated mobile app, making the system more user-friendly.
- **Voice Activation:** The system can be integrated with voice control technologies like Alexa or Google Assistant to allow hands-free operation for greater convenience.
- **Solar Power:** The system can be made more energy-efficient by incorporating a solar power option, making it suitable for off-grid locations and reducing dependency on conventional electricity.
- **Water Purification:** Advanced water purification methods, such as filtration or UV treatment, can be added to ensure that the dispensed water is clean and safe for drinking.
- **AI Water Prediction:** The use of AI-based algorithms will allow the system to predict water usage patterns, optimizing the refilling process and reducing wastage.
- **Multi-Tank Management:** The system can be expanded to support multiple water tanks, enabling automatic switching between tanks and ensuring continuous water supply.
- **Touchscreen Interface:** A touchscreen interface can be implemented for a more intuitive and modern user experience, providing easier input and control options.
- **Leak Detection:** A leak detection system can be added to alert users if a leak is detected, preventing unnecessary water wastage and damage to the system.

REFERENCES

1. R. Sureshkumar, S. S. Prasad, M. Thangatamilan, R. B. Karthikeyan, K. S. Jagadeesh and E.L. Kumar, "IoT based Bottle Filling System using PLC," 2023 International Conference on Energy, Materials and Communication Engineering (ICEMCE), Madurai, India, 2023, pp. 1-5

2. S. Nalini Durga, M. Ramakrishna, G. Dayanandam, "Autonomous Water tank Filling System using IoT," September 2018 International Journal of Computer Sciences and Engineering 6(9):1-4.

3. Abdul Faimeed, Ambekar Gajanan, Adhav Atish, Bedre Somnath, Bhure Vishal, "Automated water bottle filling system, "International Research Journal of Engineering and Technology (IRJET),".

4. Ms. Rucha R, Shastrakar, Vaishnavi Sakinala, "IoT based water level monitoring & controlling system using arduino uno esp8266 module," International Research Journal of Modernization in Engineering Technology and Science. Issue:05/May-2024

APPENDIX A

DATASHEET

DATA SHEET OF Node MCU ESP8266 V3

Feature	**Specification**
Microcontroller	ESP8266EX
Operating Voltage	3.3V
Input Voltage	4.5V – 9V (via VIN pin)
Digital I/O Pins	11 (GPIO pins, configurable as input or output)
Analog Input Pin	1 (ADC0, 10-bit resolution)
Wi-Fi Standards	IEEE 802.11 b/g/n
Flash Memory	4MB
RAM	64KB instruction RAM + 96KB data RAM
Clock Speed	80MHz (can be overclocked to 160MHz)
Communication Protocols	UART, SPI, I2C, PWM, GPIO
Wi-Fi Modes	Station, SoftAP, SoftAP + Station
Programming Language	Lua, C/C++ (via Arduino IDE other compatible software)
USB Interface	Micro-USB for power supply and programming
Dimensions	Approx. 58mm x 31mm x 13mm
Weight	~8 grams
Operating Temperature	-40°C to 125°C
Applications	IoT devices, smart home automation, remote sensing, wireless data communication.

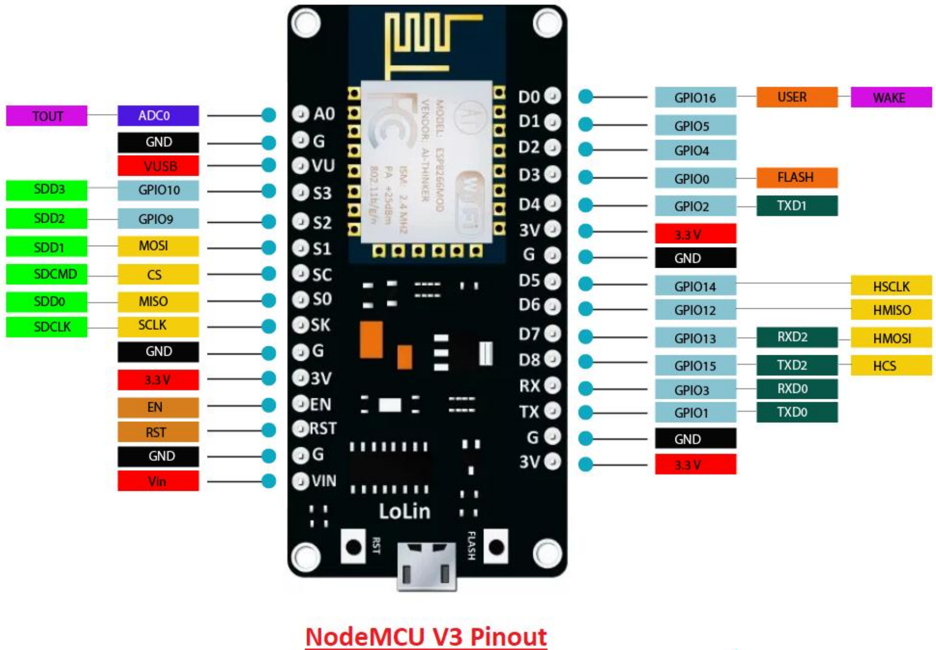


Figure:A1 ESP8266 V3 Pin Configuration

The ESP8266 V3 (often used in NodeMCU modules) is a Wi-Fi-enabled microcontroller based on the ESP8266EX chip, widely utilized for IoT applications. Key features include:

1. **CPU & Memory**: The ESP8266EX integrates a 32-bit Tensilica L106 RISC processor with a clock speed of up to 160 MHz. It includes 160KB of SRAM and integrates up to 4MB of flash memory.
2. **Connectivity**: It supports Wi-Fi 802.11 b/g/n standards, capable of transmitting data at 54 Mbps using OFDM modulation. It also has various communication protocols like UART, SPI, and I2C
3. **Power**: The ESP8266 operates within a 3.0V to 3.6V range and features low power consumption modes, including deep sleep mode. Power consumption during Wi-Fi transmissions can vary, with 170 mA for 802.11b at 11 Mbps
4. **GPIO & Peripherals**: It offers multiple General Purpose I/O pins (GPIO), many of which are multifunctional (e.g., PWM, SPI, I2C), allowing integration with sensors, motors, and other peripherals

DATA SHEET OF ULTRASONIC SENSOR

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

(1) Using IO trigger for at least 10us high level signal
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
(4) Test distance = (high level time×velocity of sound (340M/S) / 2

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

DATA SHEET OF WATER FLOW SENSOR

Water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, rotor rolls. Its speed changes with different rate of flow. The hall-effect sensor outputs the corresponding pulse signal. This one is suitable to detect flow in water dispenser or coffee machine. We have a comprehensive line of water flow sensors in different diameters. Check them out to find the one that meets your need most.

Features	**Specifications**
Working Voltage:	DC 4.5V~24V
Normal Voltage:	DC 5V~18V
Max. Working Current	Load capacity: ≤ 10 mA (DC 5V)
Flow Rate Range	1~30L/min
Load Capacity:	≤10mA (DC 5V)
Operating Temperature:	≤80℃
Liquid Temperature	≤120℃
Allowing Pressure	: ≤1.75MPa
Storage Temperature:	-25～+ 80℃
Insulation resistance	≥ 100MΩ
Outer diameter	20mm
Intake diameter	9mm
Outlet diameter	12mm

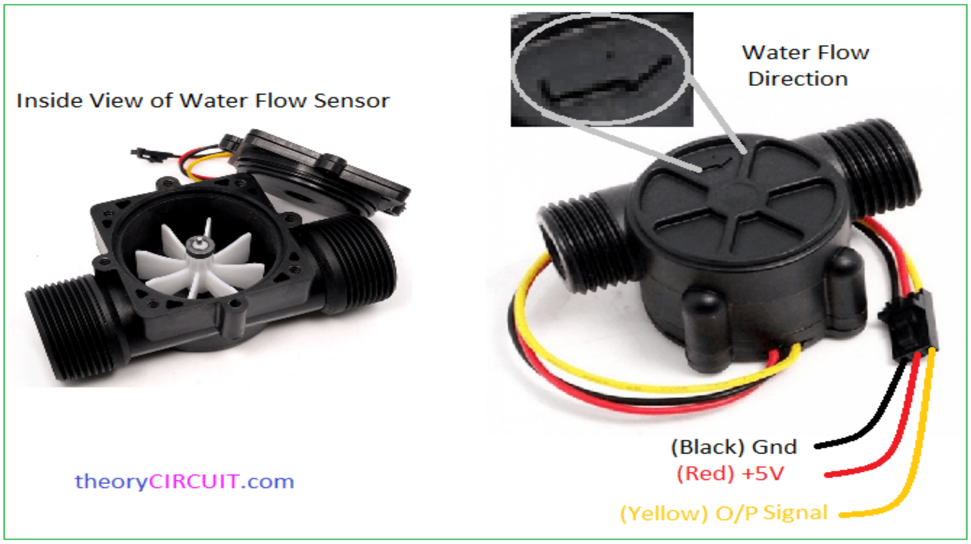


Figure A2: Inside view of water flow sensor

APPENDIX-B: CODE

ARDUINO SOURCE CODE

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2); // Set LCD address to 0x27 for a 16x2 display

// Keypad setup
const byte ROWS = 4; // Four rows
const byte COLS = 3; // Three columns
char keys[ROWS][COLS] = {
 {'1', '2', '3'},
 {'4', '5', '6'},
 {'7', '8', '9'},
 {'*', '0', '#'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; // Connect to the row pins of the keypad
byte colPins[COLS] = {5, 4, 3}; // Connect to the column pins of the keypad
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// Relay pin
const int relayPin = 2;
// Buzzer pin
const int buzzerPin = 3; // Define the buzzer pin

// Variables
float waterAmount = 0; // Water amount in liters
bool dispensing = false;
const int MAX_LITERS = 10; // Maximum allowed liters

void setup() {
  pinMode(relayPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT); // Set buzzer pin as OUTPUT
  digitalWrite(relayPin, HIGH); // Ensure relay is off initially
  digitalWrite(buzzerPin, LOW); // Ensure buzzer is off initially
  lcd.init();
  lcd.backlight();

  // Display project name
  lcd.setCursor(0, 0);
  lcd.print("IOT BASED WATER");
  lcd.setCursor(0, 1);
  lcd.print("FILLING MACHINE");
  delay(3000);
  lcd.clear();
```

```
}

void loop() {
  if (!dispensing) {
    lcd.setCursor(0, 0);
    lcd.print("Enter amount:");
    lcd.setCursor(0, 1);
    lcd.print(waterAmount, 1); // Display water amount with one decimal point
    lcd.print(" L ");

    char key = keypad.getKey();
    if (key) {
      if (key >= '0' && key <= '9') {
        waterAmount = waterAmount * 10 + (key - '0') * 0.1;
        if (waterAmount > MAX_LITERS) { // Limit to max liters
          waterAmount = MAX_LITERS;
        }
      } else if (key == '#') {
        if (waterAmount > 0) {
          startDispensing();
        }
      } else if (key == '*') {
        waterAmount = 0; // Reset amount
      }
    }
  }
}

void startDispensing() {
  dispensing = true;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Dispensing...");
  lcd.setCursor(0, 1);
  lcd.print("Amount: ");
  lcd.print(waterAmount, 1);
  lcd.print(" L");

  // Activate the buzzer at the start of dispensing
  digitalWrite(buzzerPin, HIGH); // Turn on the buzzer
  delay(500); // Buzzer sounds for 500 milliseconds
  digitalWrite(buzzerPin, LOW); // Turn off the buzzer

  digitalWrite(relayPin, LOW); // Turn on the relay (start motor)
  delay(2000); // Simulate dispensing for 2 seconds (fixed time)

  digitalWrite(relayPin, HIGH); // Turn off the relay (stop motor)
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
  lcd.print("Dispensing Done!");
  delay(2000);

  waterAmount = 0;
  dispensing = false;
  lcd.clear();
}
```

ESP8266 SOURCE CODE

```
#define BLYNK_TEMPLATE_ID "TMPL3AXTP6KKC"
#define BLYNK_TEMPLATE_NAME "Water"
#define BLYNK_AUTH_TOKEN "CzOeuwgMGRltk9bIkgqA-tGtVPHX1w0T"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <ThingSpeak.h>

#define WIFI_SSID "ASHWATH"
#define WIFI_PASSWORD "12345678"

#define TRIGGER_PIN D2
#define ECHO_PIN D1

#define THINGSPEAK_WRITE_API_KEY "ZJ7UDVO0SW9JYHX3"
#define THINGSPEAK_SERVER "api.thingspeak.com"

#define GREEN_LED_VPIN V1
#define YELLOW_LED_VPIN V2
#define RED_LED_VPIN V3
#define RELAY_VPIN V4

#define GREEN_LED_PIN D5
#define YELLOW_LED_PIN D6
#define RED_LED_PIN D7
#define RELAY_PIN D8

long duration;
int distance;
WiFiClient client;

unsigned long lastUpdateTime = 0;
const unsigned long updateInterval = 15000;
void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

```
}
Serial.println(" Connected to WiFi!");
Blynk.begin(BLYNK_AUTH_TOKEN, WIFI_SSID, WIFI_PASSWORD);
ThingSpeak.begin(client);
pinMode(TRIGGER_PIN, OUTPUT);




pinMode(ECHO_PIN, INPUT);
pinMode(GREEN_LED_PIN, OUTPUT);
pinMode(YELLOW_LED_PIN, OUTPUT);
pinMode(RED_LED_PIN, OUTPUT);
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, LOW);
}

void loop() {
digitalWrite(TRIGGER_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIGGER_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);
distance = duration * 0.0344 / 2;
Blynk.virtualWrite(V0, distance);
Serial.print("Distance: ");
Serial.println(distance);

if (distance < 6) {
  Blynk.virtualWrite(GREEN_LED_VPIN, HIGH);
  Blynk.virtualWrite(YELLOW_LED_VPIN, LOW);
  Blynk.virtualWrite(RED_LED_VPIN, LOW);
  digitalWrite(GREEN_LED_PIN, HIGH);
  digitalWrite(YELLOW_LED_PIN, LOW);
  digitalWrite(RED_LED_PIN, LOW);
} else if (distance >= 6 && distance <= 12) {
  Blynk.virtualWrite(GREEN_LED_VPIN, LOW);
  Blynk.virtualWrite(YELLOW_LED_VPIN, HIGH);
  Blynk.virtualWrite(RED_LED_VPIN, LOW);
  digitalWrite(GREEN_LED_PIN, LOW);
  digitalWrite(YELLOW_LED_PIN, HIGH);
  digitalWrite(RED_LED_PIN, LOW);
} else if (distance > 12 && distance <= 18) {
  Blynk.virtualWrite(GREEN_LED_VPIN, LOW);
  Blynk.virtualWrite(YELLOW_LED_VPIN, LOW);
  Blynk.virtualWrite(RED_LED_VPIN, HIGH);
  digitalWrite(GREEN_LED_PIN, LOW);
  digitalWrite(YELLOW_LED_PIN, LOW);
```

```
    digitalWrite(RED_LED_PIN, HIGH);
  }
   unsigned long currentTime = millis();
   if (currentTime - lastUpdateTime >= updateInterval) {
    ThingSpeak.setField(1, distance);




 int response = ThingSpeak.writeFields(2791748, THINGSPEAK_WRITE_API_KEY);
    if (response == 200) {
      Serial.println("Data successfully sent to ThingSpeak!");
    } else {
      Serial.print("Failed to send data. HTTP error code: ");
      Serial.println(response);
    }
    lastUpdateTime = currentTime;
  }

  Blynk.run();
}

BLYNK_WRITE(RELAY_VPIN) {
  int relayState = param.asInt();
  if (relayState == HIGH && distance > 0) {
    digitalWrite(RELAY_PIN, HIGH);
  } else {
    digitalWrite(RELAY_PIN, LOW);
  }
}
```

COURSE OUTCOMES

On completion of this course, we are able to

- Solve the identified problems.
- Analyze the available resources and their utilization.
- Present the work carried out and prepare the report.
- Work in a team to find the solutions for societal and technical problems.

CONTACT DETAILS

1.Name: ANKITHA L

USN No: 4BD22EC010

Email id: ankithaanku489@gmail.com

Mobile No:6364529820

Address: Bhoomika Nagar 2nd main 7th cross SS hospital road

2.Name: ANUSHA P R

USN No:4BD22EC012

Email id: anushapr39@gmail.com

Mobile No:8861194017

Address: Sangahalli, chennagiri taluk, Davangere district

3.Name: ASHWATH B S

USN No:4BD22EC014

Email id: ashwathsankannavar2004@gmil.com

Mobile No:6362126708

Address: Nayanagar, Taluk Bailhongal, District Belagavi

4.Name: BHAVANA T E

USN No:4BD22EC016

Email id: bhavanatlk@gmail.com

Mobile No:9606539733

Address: 16th cross Anjaneya badavane, Dvg