# CHAPTER 1
# INTRODUCTION

The project is aimed at developing and evaluating the performance of a smart water management system using embedded technology and IoT principles. Before delving into its implementation, an introduction is needed to the components and technologies involved in the project. The entire report is centered around the field of Embedded Systems and the Internet of Things (IoT), specifically focusing on how microcontrollers, sensors, and wireless communication are used to automate water usage monitoring in a home environment. Aquaxpert is designed to integrate various hardware modules such as flow sensors, solenoid valves, and microcontrollers like the ESP32, with real-time monitoring and control enabled through cloud platforms and mobile applications. This system demonstrates how embedded hardware combined with software logic can be applied to solve real-world problems like water wastage and inefficient consumption. As such, the project report includes an overview of embedded systems, their role in automation, and how IoT platforms like ThingSpeak or Blynk can be leveraged for remote monitoring and control. The implementation of AquaXpert showcases how these technologies come together to create an efficient, user-friendly, and intelligent water management solution.

### 1.1 Internet of things (IOT)

The Internet of Things (IoT) is a technological paradigm that enables physical devices to connect to the internet and communicate with each other, often without direct human intervention. The concept of IoT can be traced back to the early 1980s, when a Coca-Cola vending machine at Carnegie Mellon University became the first internet-connected appliance, capable of reporting inventory and temperature status. Over the years, this concept evolved with advancements in embedded systems, wireless communication, and cloud computing. IoT began gaining significant attention in the early 2000s, particularly with the development of low-cost microcontrollers, sensors, and network modules. Today, IoT has found widespread application in domains such as smart homes, healthcare, agriculture, manufacturing, and environmental monitoring. In a typical IoT system, devices collect data from their surroundings through sensors, process it locally or via cloud servers, and perform intelligent actions based on that data.

Today they are widely used to serve various purposes like:

- Smart homes, including automated lighting, smart thermostats, water management systems, and security systems.
- Healthcare applications, such as remote patient monitoring, wearable health trackers, and smart medical devices.
- Industrial automation, including predictive maintenance, energy monitoring, and smart manufacturing.
- Agriculture, where IoT enables smart irrigation, soil monitoring, and livestock tracking.
- Smart cities, including waste management systems, traffic control, and environmental monitoring.

### 1.1.1 Types of Setup

AquaXpert, being an IoT-based embedded system, follows a setup that involves both development and deployment environments. The typical setup consists of a host system (usually a PC or laptop) used for programming and debugging, and a target system (microcontroller or embedded board) that executes the water management functions.

Standalone Iot setup:

In AquaXpert, the target (such as an ESP32 or Arduino board) is programmed from the host using a USB or serial connection. Once deployed, the target functions independently and connects to the cloud or local network via Wi-Fi. The setup does not require constant connection to the host after deployment.

This setup enables real-time water monitoring and smart control without needing a constant wired connection to the development environment, making it ideal for home automation.

**Removable Storage Setup:**

In the removable setup for AquaXpert, the development environment resides on a host computer (typically a PC with Arduino IDE or PlatformIO), where the firmware is compiled and uploaded to **a** removable microcontroller module, such as an ESP32. This microcontroller acts as the target device and is later detached from the host and integrated into the actual hardware setup (e.g., water flow sensor, valve, Wi-Fi module).

- The host system is used to write the firmware.
- The compiled code is uploaded via USB to the controller, which is then moved to the smart water management system.
- No continuous link is required between the host and the AquaXpert system after programming.
- Suitable for production-level deployment, where firmware updates are rare or handled via OTA (Over-The-Air).

Standalone Setup:

In the standalone setup, the AquaXpert system operates as a self-contained unit, with all essential components (microcontroller, sensors, actuator drivers, and communication modules) installed and configured locally**.**

- The ESP32 or equivalent board runs the firmware directly and continuously monitors water usage, controls valves, and sends updates to the cloud or mobile app.
- No host system is required during operation.
- Ideal for real-time operation, remote control, and continuous data logging.

- **1.2 Operating Systems**

In embedded systems such as AquaXpert, the complexity of the application determines the need for an operating system. If the system only needs to perform a single task, such as turning a valve on or off based on a sensor reading, then a simple binary file can be written to the microcontroller and executed. However, AquaXpert involves multiple tasks, such as continuously monitoring water flow, sending data to the cloud, receiving commands from a mobile application, and performing emergency shutoffs during leak detection. To manage these tasks efficiently and simultaneously, the use of an embedded operating system becomes necessary.

Embedded operating systems are typically classified into two categories: Real-Time Operating

Systems (RTOS) and Non-Real-Time Operating Systems. Real-Time Operating Systems are designed to guarantee a response to each event within a predefined time frame. This is critical for applications like AquaXpert, where immediate reaction is required in the event of abnormal water flow or leakage. For instance, a delayed response in closing a valve could lead to significant water waste or damage. RTOS ensures that tasks like sensor reading, alert generation, and valve control are executed within strict timing constraints. Some widely used RTOS platforms in embedded systems include FreeRTOS, Zephyr, Mbed OS, and VxWorks. AquaXpert can benefit significantly from FreeRTOS, which is lightweight and supports multitasking on platforms like ESP32. With FreeRTOS, separate tasks can be created to handle different functions such as monitoring, cloud updates, and emergency controls, all running concurrently without interfering with one another.

On the other hand, Non-Real-Time Operating Systems do not provide strict timing guarantees. These systems are generally used in scenarios where timing is not critical, but more complex application support is needed. Operating systems like Embedded Linux and Windows IoT Core fall into this category. They are often used in smart home systems that require user interfaces, cloud integration, and multimedia support. If AquaXpert were to be expanded into a more advanced system with features like voice control, touchscreen dashboards, or advanced analytics, then a non-real-time OS running on hardware like Raspberry Pi might be considered.

# CHAPTER 2
# LITERATURE SURVEY

**Chemical Ghosh et al. (2017)** explored the deployment of an IoT-enabled water management system for efficient water usage in urban areas. Their study combined multiple sensors for flow, pressure, and quality monitoring with real-time data transmission via MQTT protocol. The authors highlighted the role of sensor calibration and data accuracy in ensuring system reliability, laying foundational insights for Aquaxpert's architecture in managing water distribution and quality dynamically.

**Journal of Cleaner Production In this 2019 article, Singh et al.** developed a smart irrigation and water management system using IoT-based wireless sensor networks to optimize agricultural water use. They integrated soil moisture, temperature, and humidity sensors with automated water valves controlled through a mobile application. Their findings on real-time adjustments based on environmental data provide a model for Aquaxpert's decision-making capabilities in diverse applications, including urban and agricultural water management.

**IEEE Access Yadav et al. (2020**) presented a robust IoT-based water management framework that utilizes edge computing for local data processing and real-time alerts. Their work addresses the challenges of latency and data security by incorporating edge-based processing nodes before cloud transmission. This approach has direct implications for Aquaxpert, which can benefit from reduced reliance on cloud resources and quicker responsiveness to changes in water demand or quality.

**Environmental Monitoring and Assessment Patel and Patel (2021)** introduced a smart water management system that monitors leakage and consumption patterns in residential complexes. They used an array of ultrasonic and flow sensors to detect abnormal consumption, leading to significant water savings. Their emphasis on anomaly detection algorithms and predictive maintenance directly informs Aquaxpert's strategies to minimize wastage and ensure sustainable resource use.

**Informatics and Systems Bhargava and Gupta (2022)** explored an AI-enhanced IoT water management system that predicts demand patterns and optimizes supply. Their model integrates machine learning algorithms with sensor data to dynamically adjust water distribution. This work aligns closely with Aquaxpert's goals of proactive management and efficient resource allocation, highlighting the critical role of predictive analytics in smart water systems.

**IEEE Communications Surveys & Tutorials Al-Fuqaha et al. (2015)** provided a comprehensive review of IoT architectures and communication protocols relevant for smart water management systems. They detailed how IoT frameworks can address challenges in scalability, interoperability, and security—essential considerations for systems like Aquaxpert.

**Environmental Earth Sciences Bhata and Srivastava (2018)** examined the integration of IoT-based sensors and GIS mapping for comprehensive water resource management. Their approach involved spatial data analytics and real-time water quality sensing, enabling dynamic water resource allocation. These insights directly inform Aquaxpert's spatially aware management of water resources.

**IEEE Internet of Things Journal Nguyen and Kim (2018)** developed a low-power, long-range water monitoring system leveraging LoRaWAN technology for data transmission. Their emphasis on energy efficiency and reliable connectivity for remote areas can enhance Aquaxpert's deployment flexibility and reliability in large-scale water networks.

**Journal of Network and Computer Applications Farsi et al. (2019)** proposed an intelligent water monitoring system based on cloud computing and big data analytics. Their architecture included real-time analytics to detect leakage and optimize water distribution, which parallels Aquaxpert's data-driven, cloud-integrated framework for sustainable water management.

**Sustainable Cities and Society Panda and Kumar (2020)** introduced a smart water grid using IoT sensors and predictive models to manage urban water supply and demand. Their findings on integrating user consumption patterns and real-time adjustments offer critical insights for Aquaxpert's urban water management features.

**Environmental Technology & Innovation** In their **2020** study, **Kumar et al.** explored a modular IoT-based system for continuous water quality assessment in rural areas. Their work emphasizes affordability and ease of deployment, key aspects for Aquaxpert's scalable and cost-effective implementation in resource-constrained environments.

**Computers, Environment and Urban Systems Ali et al. (2021)** developed a cloud-based water management platform integrated with machine learning for real-time decision-making. They demonstrated significant improvements in leakage control and equitable water distribution, aligning with Aquaxpert's objectives of reducing waste and improving sustainability.

**Internet of Things and Cloud Computing Sundaram and Rajasekar (2021)** showcased an IoT-enabled automated water distribution system based on user consumption data and remote control of valves. Their findings highlight the importance of automation and user-centric data insights, key pillars for Aquaxpert's responsive water supply management.

## 2.1 EXISTING SYSTEM

In current domestic and residential water supply systems, water consumption is largely unmanaged, leading to unintentional overuse and wastage. The traditional setups involve a direct connection from the municipal water supply or storage tanks to household plumbing, without any intermediate system to monitor or control usage. These systems typically lack intelligence, and any form of management—such as turning off valves or identifying leaks is manual and reactive rather than proactive. Users are often unaware of how much water is being used daily, and there are no alerts to indicate abnormal usage or leakage.

Some existing solutions attempt to automate water management using basic sensors or timers. For example, float sensors are commonly used in water tanks to automatically stop motors when tanks are full. However, these setups are limited to tank level management and do not monitor real-time consumption. Others use manual meters for water billing, but these are not connected to any digital system and cannot generate timely alerts or allow remote control. Recent technological advancements have led to the development of smart meters and IoT-based water monitoring systems in urban infrastructure and high-end smart homes. These systems incorporate flow sensors, Wi-Fi modules, and microcontrollers to measure water usage and send data to centralized dashboards. However, such systems are often expensive, complex to install, and not user-configurable, which limits their accessibility for the general population.

## 2.2 PROPOSED METHOD

The AquaXpert system is proposed as a smart, integrated, and affordable solution to efficiently manage household water usage, prevent wastage, and ensure real-time monitoring and control. It addresses the limitations of existing systems by incorporating modern embedded and IoT technologies to create a comprehensive home water management system. The proposed system combines hardware components like water flow sensors and solenoid valves with wireless communication modules and cloud-based applications to provide users with complete visibility and control over their water consumption. At the core of AquaXpert is a microcontroller, such as the ESP32, which acts as the brain of the system. It reads real-time data from a flow sensor (such as the YF-S201), processes the information, and determines whether the water usage is within normal limits. If abnormal usage is detected—such as continuous flow indicating a leak or excessive consumption—the microcontroller automatically activates a solenoid valve to shut off the water supply. This action helps prevent water loss and potential damage due to leaks.

The system also connects to an IoT platform, such as Blynk, to enable remote monitoring and control through a smartphone application. The app provides users with live usage data, historical consumption trends, and instant notifications about anomalies or emergency shutoffs. Users can also manually control the water flow from anywhere using the app, giving them full authority over their water system even when they are not at home. The proposed system operates autonomously, requires minimal maintenance, and ensures data integrity through cloud synchronization. With its combination of automation, real-time alerts, remote accessibility, and low-cost implementation, AquaXpert provides a powerful and practical solution to modern water management challenges. It empowers users to actively participate in water conservation efforts while enjoying the convenience of a smart and connected home environment.

# CHAPTER 3
# DESIGN AND IMPLEMENTATION

## 3.1 BLOCK DIAGRAM

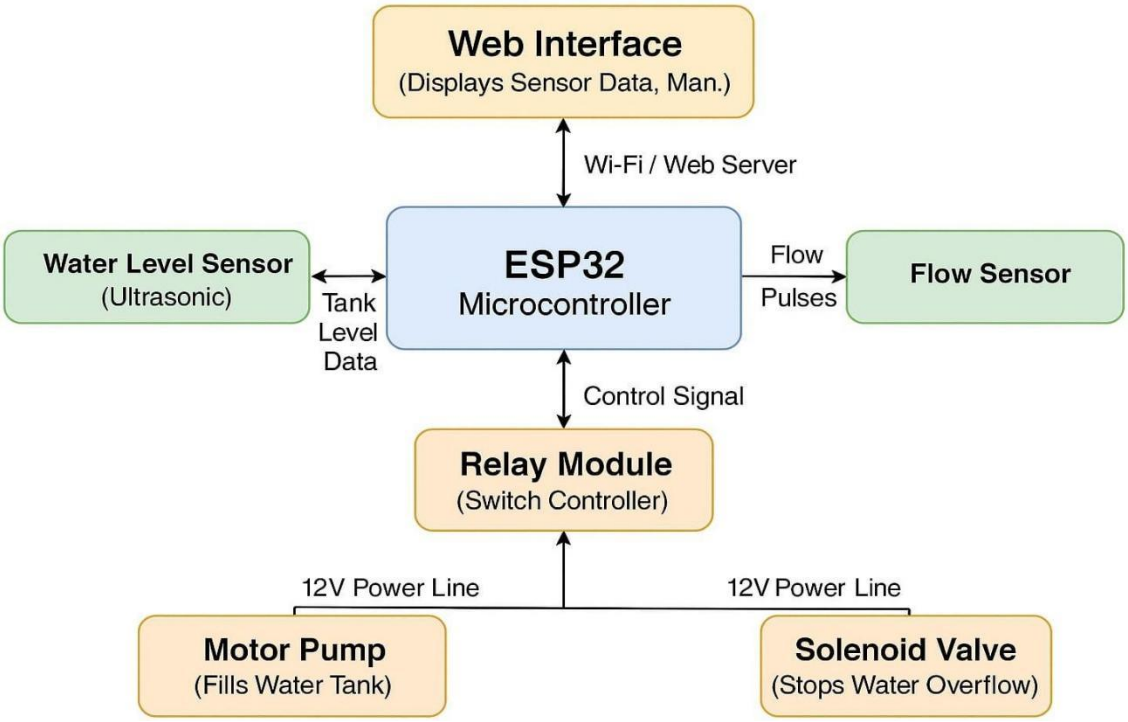



Fig. 3.1 Block diagram

The given block diagram illustrates an AquaXpert Smart Water Management System is an automated IoT-based solution designed to efficiently manage water tank filling and prevent overflow. The system uses an ESP32 microcontroller as its central unit, which receives water level data from an ultrasonic sensor and flow rate data from a flow sensor. Based on this data, the ESP32 intelligently controls a relay module that switches the motor pump and solenoid valve on or off. The motor pump is used to fill the tank when the water level is low, while the solenoid valve stops water flow when the tank is full to avoid overflow. Additionally, the ESP32 hosts a web server, allowing users to remotely monitor real-time sensor data and control the system through a web interface over Wi-Fi. By automating tank filling and providing remote access, AquaXpert enhances water usage efficiency, reduces manual intervention, and supports smart home water management.

## 3.2 ESP32

The ESP32 is a low-cost, low-power microcontroller with integrated Wi-Fi and Bluetooth capabilities. It is a powerful and versatile platform widely used in IoT applications. The ESP32 is a powerful and low-cost microcontroller developed by Espressif Systems, widely used in IoT and embedded system projects. It features a dual-core Tensilica LX6 processor, integrated Wi-Fi, and Bluetooth capabilities, making it ideal for wireless communication and smart device applications. The ESP32 offers multiple GPIO pins, ADCs, DACs, PWM outputs, SPI, I2C, UART interfaces, and other peripherals, allowing easy connection with sensors, actuators, and external modules.

With built-in support for Wi-Fi and Bluetooth, the ESP32 can connect to networks, create web servers, or communicate with smartphones and cloud services. It is programmable using the Arduino IDE or ESP-IDF, and its compact size, energy efficiency, and rich features make it highly suitable for applications like home automation, smart agriculture, industrial control, and real-time monitoring systems.


Figure 3.2 ESP32

### 3.2.1 Core Functions

- Data processing: Continuous sensor data analysis and interpretation
- Decision making: Executing control algorithms faster than human response
- Communication hub: Managing all system interconnections and protocols
- Web server hosting: Serving dynamic interface without external infrastructure
- Manual override handling: Accepting and processing user commands

The ESP32 is an advanced and cost-effective microcontroller developed by Espressif Systems, designed for use in embedded systems and Internet of Things (IoT) applications. It is a successor to the ESP8266 and offers enhanced features including integrated Wi-Fi and Bluetooth capabilities, dual-core processing, and a wide range of I/O interfaces. The ESP32 is widely used in smart devices, wireless sensor networks, and automation systems due to its powerful performance, energy efficiency, and ease of programming. It supports development using popular platforms such as the Arduino IDE and ESP-IDF, making it accessible for both beginners and professionals.

### 3.2.2 ESP32 Hardware Description

The ESP32 microcontroller integrates a powerful dual-core Tensilica LX6 processor capable of running at up to 240 MHz, providing excellent computational performance. It includes built-in Wi-Fi and bluetooth (Classic and BLE**)** modules for seamless wireless connectivity. The chip is equipped with a wide variety of GPIO pins that can be used for digital input/output operations, along with support for analog input (ADC) and analog output (DAC)**.** Additionally, the ESP32 features SPI, I2C, UART**,** and PWM interfaces to connect with external devices and peripherals. Onboard components also include touch sensors, a hall sensor, and temperature sensor, enhancing its capability for real-world interaction. Most ESP32 development boards come with a USB port**,** voltage regulators, and Flash memory, making them suitable for direct programming and deployment.

ESP32 microcontroller is a feature-rich hardware platform designed for embedded and IoT applications. At its core, it houses a dual-core Tensilica LX6 processor capable of operating up to 240 MHz, delivering high processing power suitable for multitasking and real-time operations. One of its most notable features is the integrated Wi-Fi and Bluetooth (Classic and BLE) modules, allowing seamless wireless communication without external components. The ESP32 includes a wide range of GPIO (General Purpose Input/Output) pins, many of which are multiplexed with functions such as ADC (Analog-to-Digital Converter), DAC (Digital-to-Analog Converter), PWM (Pulse Width Modulation), and serial communication interfaces like SPI, I2C, and UART. It also features onboard touch sensors, a hall effect sensor, and a temperature sensor, expanding its capability for physical sensing. Most ESP32 development boards include additional components such as flash memory, USB-to-serial converters, voltage regulators, and antennae, making them ready for immediate prototyping and deployment. This rich hardware configuration makes the ESP32 a highly versatile and powerful choice for modern electronic and IoT-based systems.

In the Smart Water Management System, the ESP32 is chosen as the core microcontroller because of its built-in Wi-Fi capability, which allows seamless wireless communication between the system and a web-based dashboard or mobile application. Its dual-core processor ensures fast data processing and responsiveness, which is essential for real-time control and automation tasks. Additionally, the ESP32 is energy-efficient, cost-effective, and programmable through the Arduino IDE, making it an ideal choice for developing scalable and reliable IoT-based smart water management solutions.

### 3.2.3 ESP32 Specifications

- **Processor**: Dual-core Tensilica LX6 CPU, up to 240 MHz
- **Wireless Connectivity**: 802.11 b/g/n Wi-Fi, Bluetooth v4.2 (Classic and BLE)
- **SRAM**: 520 KB
- **Flash Memory**: Typically 4 MB (external)
- **GPIO Pins**: Up to 34, with various alternate functions
- **Analog Features**: 12-bit ADCs, 8-bit DACs
- **Communication Interfaces**: SPI, I2C, UART, CAN, I2S
- **Timers**: Multiple general-purpose and watchdog timers
- **Power Modes**: Active, Modem Sleep, Light Sleep, Deep Sleep
- **Operating Voltage**: 3.0 – 3.3V
- **Development Support**: Arduino IDE, ESP-IDF, MicroPython, PlatformIO

- In this system, the ESP32 acts as the central controller for processing data from sensors (water level, pH, temperature, etc.) and controlling the water motor and solenoid valve. It also hosts a web-based dashboard for remote monitoring and control.

## 3.3 Ultrasonic Trigger (HC-SR04):

The HC-SR04 is a popular ultrasonic distance measurement sensor that uses sound waves to measure distance. It consists of two parts: a transmitter (trigger) and a receiver (echo). The HC-SR04 is a popular and low-cost ultrasonic distance sensor used to measure the distance between the sensor and an object using sound waves. It operates by emitting an ultrasonic pulse through its transmitter and measuring the time it takes for the echo to return to its receiver. Based on the time delay, the distance is calculated using the speed of sound. The sensor typically operates at 5V and has four pins: VCC, GND, TRIG (trigger), and ECHO. The TRIG pin is used to send a pulse, while the ECHO pin receives the reflected signal. The HC-SR04 can measure distances ranging from 2 cm to 400 cm with good accuracy, making it ideal for applications like water level monitoring, obstacle detection, and automation projects. Its ease of use and compatibility with microcontrollers like the ESP32 and Arduino make it a preferred choice for many embedded systems.

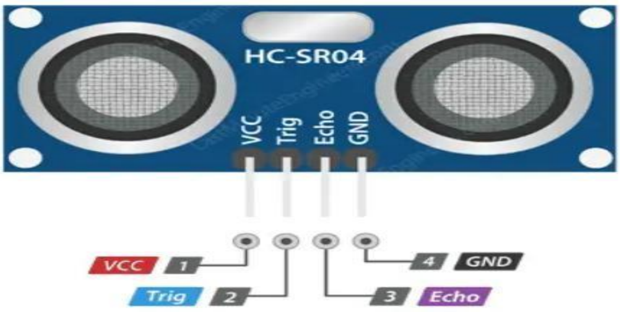



Figure 3.3 Ultrasonic Trigger (HC-SR04)

## 3.4 Water Flow Sensor (YF-S201):

Figure 3.4 Water Flow Sensor (YF-S201

The YF-S201 is a commonly used water flow sensor designed to measure the rate of water flow in a pipe. It consists of a plastic body, a rotor, and a Hall effect sensor. When water flows through the sensor, it spins the rotor, which in turn generates electrical pulses through the Hall sensor. These pulses are directly proportional to the flow rate. By counting the number of pulses per unit time, the microcontroller (like ESP32 or Arduino) can calculate the flow rate in liters per minute (L/min) and the total volume of water used.

The sensor typically operates at 5V, with three pins: VCC, GND, and Signal (Output). It is easy to interface with most microcontrollers and is widely used in smart irrigation systems, water usage monitoring, and automated water control projects. Its compact design, low cost, and accuracy make the YF-S201 an excellent choice for real-time water flow measurement in smart water management systems. In this system, the YF-S201 tracks the rate of water flow, providing data to ensure that the water flow is regulated properly and that the tank is not overfilled or wasting water.

## 3.5 Motor Pump

Figure 3.5 Motor Pump

A motor pump is an electromechanical device that combines an electric motor and a pump to move fluids (such as water, oil, or chemicals) from one location to another. The motor provides the rotational energy needed to drive the pump mechanism. When the motor is powered, it rotates the pump impeller or piston, creating a pressure difference that forces the fluid through the system. Motor pumps are widely used in various applications, including water supply systems, irrigation, industrial processes, and household appliances. They come in different types, such as centrifugal pumps, positive displacement pumps, and submersible pumps, depending on the specific requirements of flow rate and pressure. The motor pump is used to automatically fill the tank with water when the water level is low, based on signals from the ultrasonic sensor. It is controlled by the relay module.

## 3.6 Solenoid Valve

Figure 3.6 Solenoid Valve

A solenoid valve is an electromechanically operated valve used to control the flow of liquids or gases. It consists of two main parts: an electromagnetic coil (solenoid) and a valve mechanism. When electrical current passes through the coil, it generates a magnetic field that moves a plunger or piston, opening or closing the valve. When the current is stopped, a spring usually returns the valve to its original position.

Solenoid valves are widely used in automated fluid control systems due to their fast response, reliability, and ease of integration with electronic controllers. In smart water management systems, solenoid valves play a crucial role in automatically controlling the water flow based on sensor data or predefined settings. They help conserve water, prevent leaks, and enable remote operation of the system.

## 3.7 Relay (Relay Module)


Figure 3.7 Relay (Relay Module)

A relay is an electrically operated switch that allows a low-power control signal to switch on or off a higher-power circuit. It works by using an electromagnetic coil to mechanically operate a switch. When current flows through the coil, it creates a magnetic field that pulls a metal contact, changing the state of the circuit (either opening or closing it). A relay module is a ready-made circuit board that includes one or more relays along with supporting components such as transistors, diodes, and terminal blocks. It provides an easy interface between low-power microcontrollers (like Arduino or ESP32) and high-power devices (like motors, lights, and pumps). In smart water management systems, relay modules are used to control devices such as motor pumps and solenoid valves. The controller sends a low-voltage signal to the relay module, which then switches the high-voltage devices on or off as needed. This enables automated and safe control of electrical equipment in the system.

## 3.8 Battery

12V battery: It is a type of rechargeable or non-rechargeable battery that provides a nominal voltage of 12 volts. It is commonly used in a wide variety of applications, including automotive, solar energy systems, backup power supplies, and various electronics. These batteries are available in both lead-acid and lithium-ion chemistries, each with distinct characteristics and uses.

## 3.9 System Setup and Configurations

The AquaXpert project relies on a combination of embedded development tools, IoT platforms, and optional mobile integration to enable efficient smart water management. The software setup plays a critical role in enabling the system’s core functionalities such as real-time monitoring, control automation, and data visualization. To begin with, the primary development environment used is the **Arduino IDE**, a lightweight and open-source platform that supports coding and firmware uploading for microcontrollers like the ESP32. In order to program the ESP32 board, the corresponding board manager must be added via the Arduino Board Manager. Additionally, USB-to-Serial drivers such as CP210x or CH340 (depending on the ESP32 module version) must be installed on the computer to enable communication between the microcontroller and the development system.

### 3.9.1 Arduino IDE

**Arduino IDE (Integrated Development Environment)** is a software platform used to write, compile, and upload code to Arduino boards. It is simple, open-source, and widely used in electronics and embedded system projects. It provides a simple and user-friendly interface that allows users to create programs using a simplified version of C and C++. The IDE includes features such as a code editor, message area, text console, toolbar with buttons for common functions, and a series of menus for easy navigation.

One of the key functionalities of the Arduino IDE is its ability to compile and upload sketches (Arduino programs) directly to the hardware through a USB connection. The Serial Monitor feature within the IDE allows for real-time communication between the user and the Arduino board, which is especially useful for debugging and monitoring sensor data. Additionally, the IDE supports a wide range of libraries and board definitions, enabling users to easily integrate various sensors, actuators, displays, and communication modules into their projects.

The Arduino IDE is cross-platform and available for Windows, macOS, and Linux operating systems. Due to its simplicity and extensive community support, it is widely used in educational, hobbyist, and professional embedded system projects. Whether building a basic LED blinking circuit or a complex IoT-based system, the Arduino IDE serves as a powerful tool that makes embedded programming more accessible and efficient.

The programming for this is executed within the Arduino IDE

The IDE environment is mainly distributed into three sections:

- Menu Bar
- Text Editor
- Output Pane

As you download and open the IDE software, it will appear like an image below arduino ide the bar appearing on the top is called Menu Bar that comes with five different options as follow

• File – You can open a new window for writing the code or open an existing one. The following table shows the number of further subdivisions the file option is categorized into.


Figure 3.8Introduction to Arduino IDE

The checkmark appearing in the circular button is used to verify the code. Click this once you have written your code.

• The arrow key will upload and transfer the required code to the Arduino board.

• The dotted paper is used for creating a new file.

• The upward arrow is reserved for opening an existing Arduino project.

• The downward arrow is used to save the current running.

The button appearing on the top right corner is a Serial Monitor – A separate pop-up window that acts as an independent terminal and plays a vital role in sending and receiving the Serial Data.

You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+ Shift+ M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor. You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.

Figure 3.9 Serial Monitor

## 3.10 Arduino libraries:

- Contain reusable code (functions, classes, etc.)
- Make it easier to work with hardware modules or perform complex tasks
- Are stored in the libraries folder in your Arduino sketchbook

**Install it**

- Use **Arduino IDE**: Go to **Sketch** > **Include Library** > **Manage Libraries**
- Download manually (ZIP) from GitHub or elsewhere, then **Sketch** > **Include Library** > **Add .ZIP Library…**

This menu lets you:

- Manage libraries
- Add libraries (ZIP file or built-in)

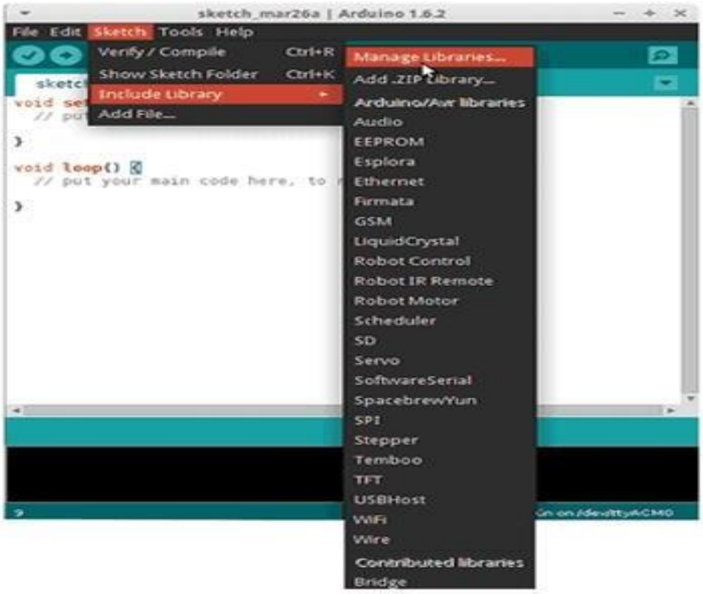



Figure 3.10 Arduino libraries

• As you click the Include Library and Add the respective library it will be on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.H>

• Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from external sources.

• Making Pins Input or Output.

• The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively.

• These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W” Writing it down with Digitalwrite

## 3.11 How to select Board:

• In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system.

• As you click the Tools on the menu, it will open like the figure below:

• Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the port's section of the Windows Device Manager.

- The following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with the COM4 port at the right bottom corner of the screen.
- After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six-button section or you can go to the Sketch section and press verify/compile and then upload.
- The sketch is written in the text editor and is then saved with the file extension .ino.
- It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, the older versions may require the physical reset on the board.
- Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

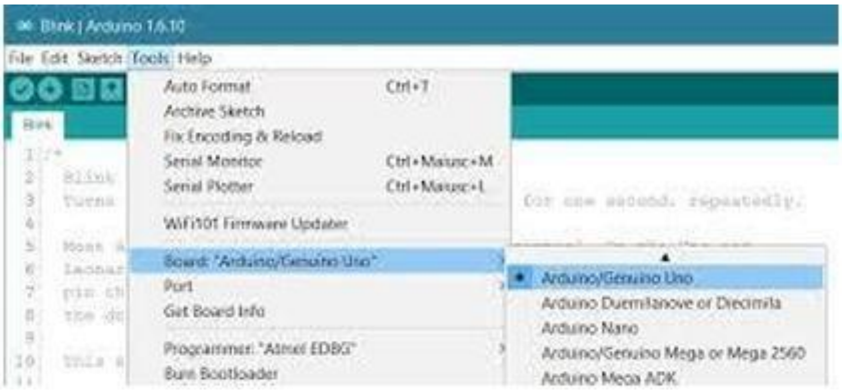



Figure 3.11 Selecting Tools

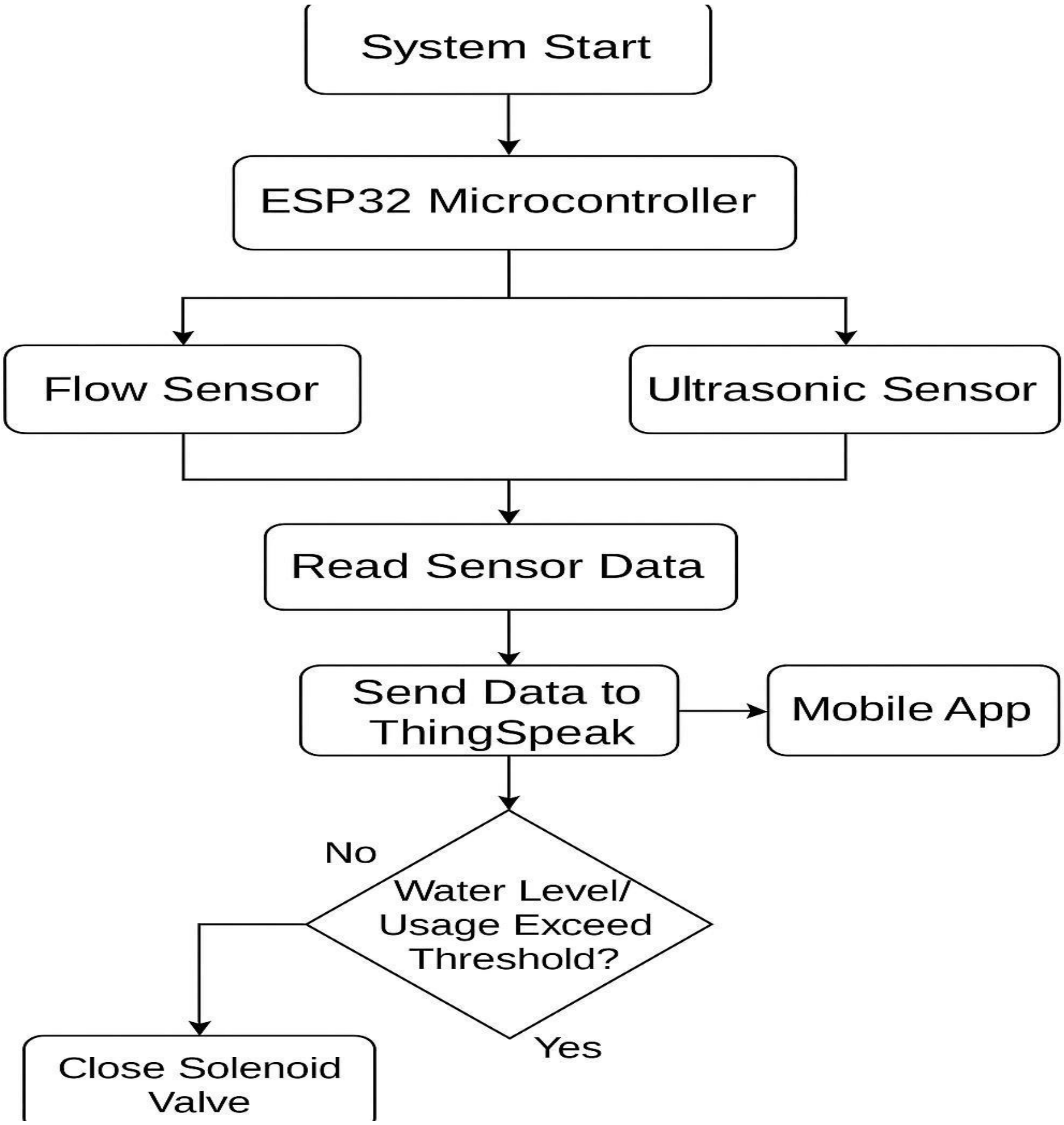
## 3.12 HTML Structure

HTML, which stands for HyperText Markup Language, is the fundamental building block used to design and structure content for web-based user interfaces. In Arduino and IoT projects, HTML is commonly used to create simple yet effective web pages that allow users to interact with devices through a browser. The basic structure of an HTML document consists of a declaration (<!DOCTYPE html>), followed by <html> as the root element, which contains two main sections: <head> and <body>. The <head> section holds metadata, the page title, and embedded styles or scripts, while the <body> section contains the visible content such as headings, text, buttons, and forms. This structure helps in organizing and displaying information on a webpage in a readable and interactive manner. In Arduino projects, especially those involving ESP32 or ESP8266 boards, HTML is used to build web dashboards that control actuators (like LEDs or motors) and monitor sensors. This allows users to communicate with the hardware wirelessly, making HTML a vital technology for user interface design in embedded systems.

# CHAPTER 4

# FUNCTIONAL DESCRIPTION

## 4.1 FLOW CHART



The given flowchart describes an IoT-based smart water management system utilizing an ESP32 microcontroller. The system begins by activating the ESP32, which controls and coordinates the operation of two sensors: a flow sensor to measure the water usage and an ultrasonic sensor to determine the water level. The ESP32 continuously reads the data from these sensors and transmits it to the ThingSpeak cloud platform for real-time monitoring and analysis. Additionally, the data is accessible through a mobile application, enabling users to track water usage and level remotely. The system then evaluates whether the collected data exceeds preset threshold values for water level or usage. If the thresholds are exceeded, the ESP32 triggers the closure of a solenoid valve to stop water flow, preventing wastage and ensuring efficient water management. This flowchart captures the essential workflow of Aquaxpert's smart water management, highlighting its automated and intelligent decision-making capabilities.

## 4.2 WORKING

The **AquaXpert** project is a IoT-based smart water management system designed to efficiently monitor, analyze, and control household water usage in real time. The working of the system begins with the deployment of water flow sensors at key water outlets such as taps, showers, sinks, kitchen pipelines, and overhead water tanks. These sensors detect the rate of water flow and transmit real-time data to a central microcontroller unit like ESP32, which is programmed to process the incoming data. The microcontroller constantly monitors water consumption, calculates the total volume used over specific periods, and identifies any anomalies such as continuous water flow that could indicate leakage or unintentional wastage.

**1. System Initialization**

- The ESP32 microcontroller is powered on and initializes the system, including Wi-Fi setup and sensor connections.

**2. Sensor Activation**

- The ESP32 activates two sensors:
  - Flow Sensor: Measures the water flow rate in the pipe.
  - Ultrasonic Sensor: Measures the water level in the storage tank or container.

**3. Data Acquisition**

- The sensors send real-time data to the ESP32.
- The flow sensor provides readings in liters/minute or gallons/minute.

- The ultrasonic sensor calculates the distance to the water surface to determine the water level.

**4.Data Processing**

- The ESP32 processes the incoming sensor data.
- It calculates parameters like total water usage or current water level.

**5.Data Transmission**

- The processed data is transmitted via Wi-Fi to the ThingSpeak cloud platform.
- ThingSpeak stores and visualizes the data in graphs, dashboards, or alerts.
- Simultaneously, the data is sent to a mobile app for real-time updates and monitoring by the user**.**

**6.Decision-Making (Threshold Checking)**

- The ESP32 continuously compares the current readings against preset threshold values (like maximum allowed water usage or tank level).
- If the water flow or water level is below the threshold, the system continues monitoring without any action.

**7Automatic Control**

- **If the threshold is exceeded (like over-usage or risk of overflow), the ESP32:**
  - **Triggers the solenoid valve to close, stopping the water flow automatically.**
  - **This prevents water wastage or overflow.**

# 4.3 SCHEMATIC DIAGRAM

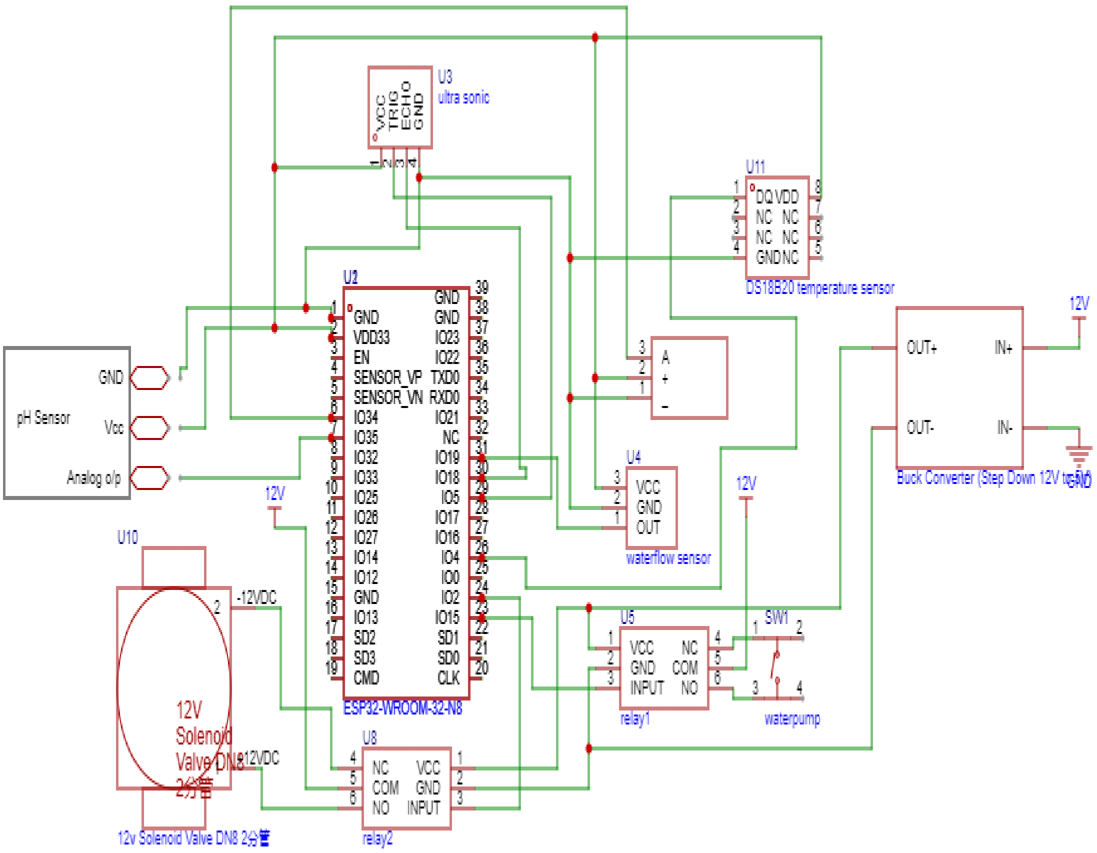



Figure 4.3 Circuit Diagram

# CHAPTER 5
# RESULTS

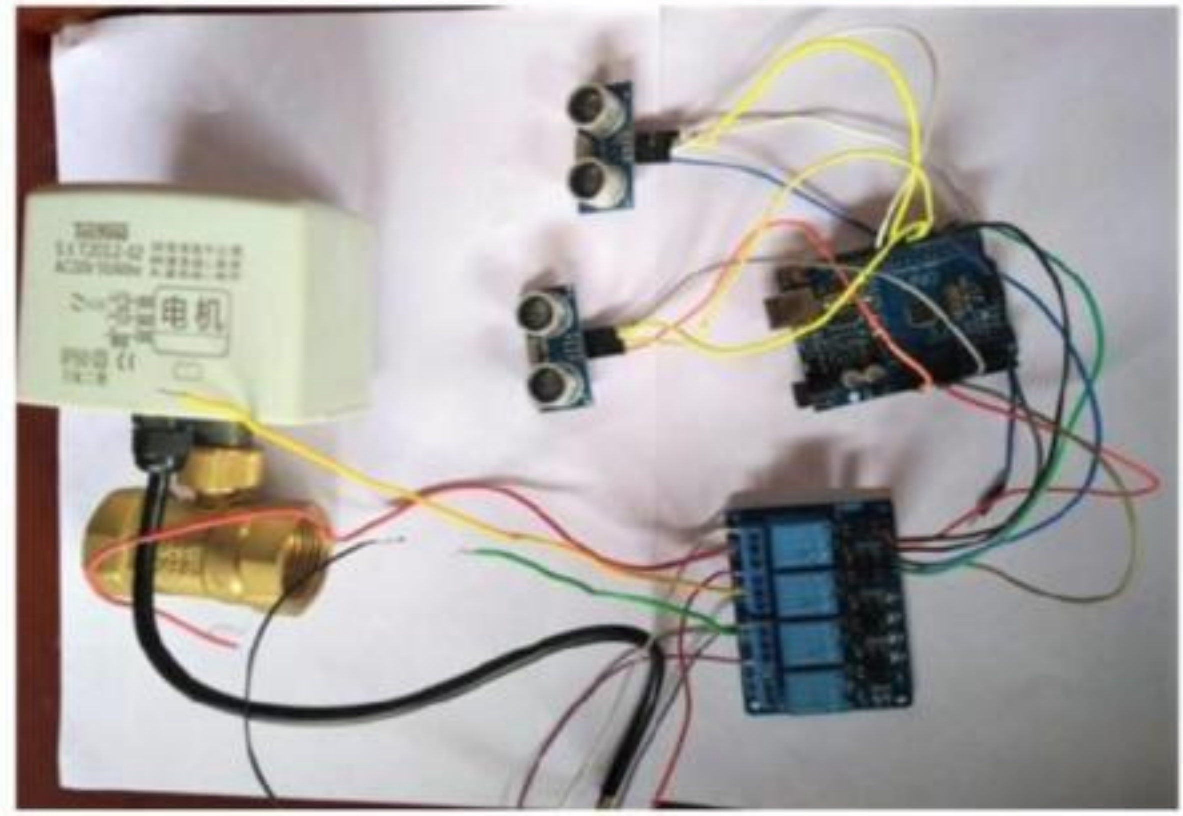


Figure 5.1 output

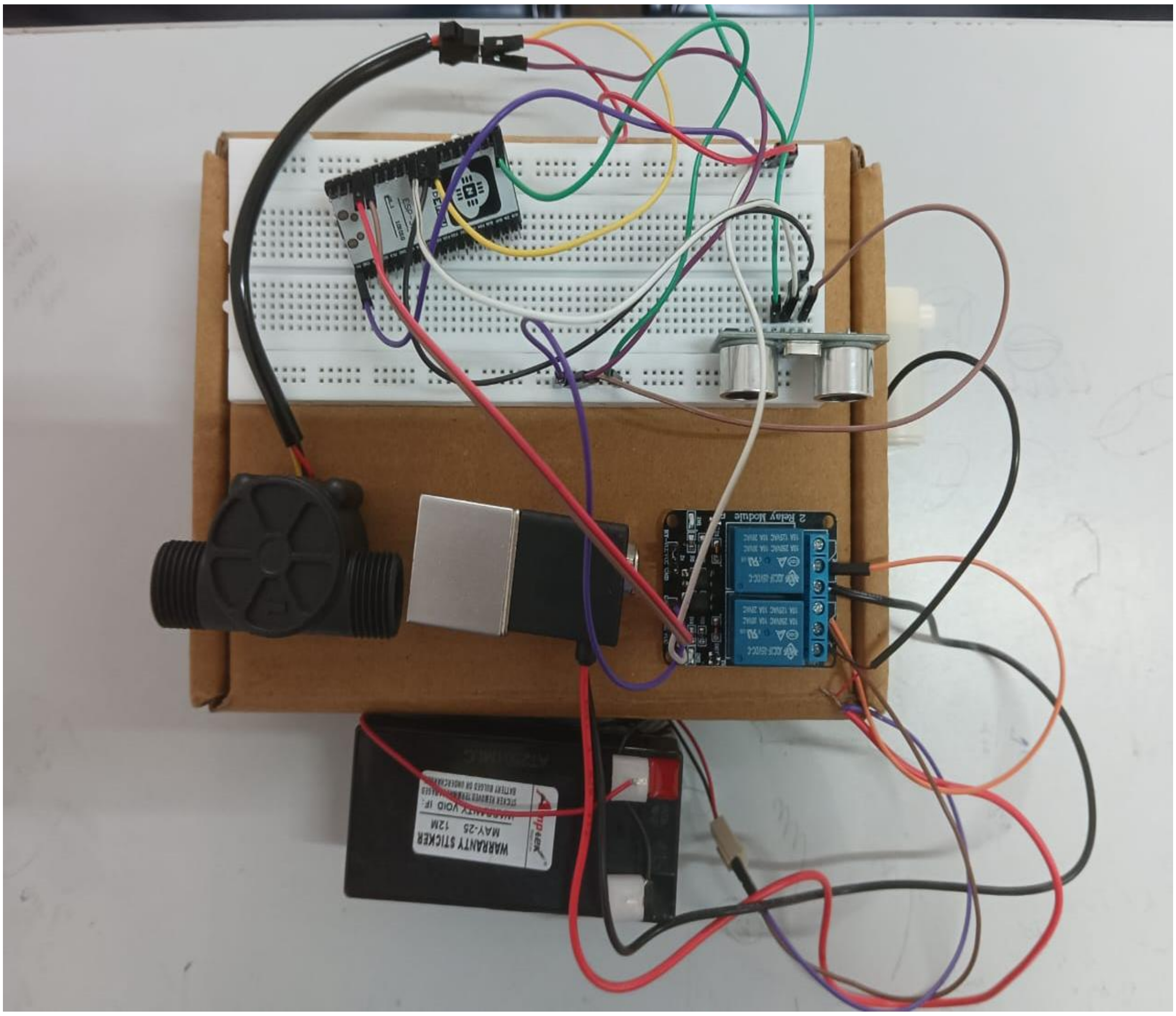


Figure 5.2 Circuit Diagram

The AquaXpert prototype was successfully implemented and tested. It automatically controlled the motor based on tank water levels and activated the solenoid valve upon detecting water flow. All operations, including sensor readings and control actions, were displayed in real-time on the serial monitor. The system functioned reliably without requiring internet access, validating its effectiveness as a low-cost, standalone smart water management solution suitable for both domestic and small-scale applications.

# CHAPTER 6

# ADVANTAGES

## ADVANTAGES

- **Automatic Motor Control Based on Water Level**
- **Basic Flow Detection for Leak or Usage Monitoring**
- **Solenoid Valve Activation on Flow Detection**
- **Real-Time Status Display on Serial Monitor**
- **Low Hardware Cost and Easy to Build**
- **Minimal Wiring and Simple Circuit Design**
- **No Internet Required for Operation**
- **Reliable Local Automation using ESP32**
- **Easy Debugging through Serial Output**
- **Suitable for Initial Testing and Functional Validation**

# CHAPTER 7

# APPLICATIONS

- **Residential Homes**

  Monitors and controls daily water usage, helps prevent leaks, overflows, and reduces water bills.

- **Apartment Complexes & Gated Communities**

  Centralized monitoring of water tanks and pipelines across multiple units; enables fair water usage tracking and billing.

- **Educational Institutions**

  Helps schools and colleges manage water usage in hostels, canteens, and washrooms, promoting water conservation.

- **Hospitals & Clinics**

  Ensures uninterrupted and efficient water supply for critical services like sanitation, sterilization, and patient care.

- **Smart Irrigation Systems**

  Automatically controls garden or farm irrigation based on soil moisture or time schedules, avoiding over-watering.

- **Industrial Use**

  Tracks and regulates water usage in factories or plants, especially in water-intensive processes.

# CHAPTER 8
# CONCLUSION

The AquaXpert project successfully demonstrates an efficient, smart water management system designed to promote water conservation, minimize wastage, and ensure safe and controlled usage of water in domestic and commercial environments. The integration of sensors, controllers, and automation allows for real-time monitoring and control of water flow, tank levels, and leak detection. By providing users with detailed usage data and alerts, AquaXpert empowers individuals to adopt more sustainable water practices while also reducing operational and maintenance costs. This system addresses one of the major challenges in urban and rural areas uncontrolled water usage and unnoticed leakage by offering a modern, tech-driven solution. AquaXpert thus contributes meaningfully toward the goal of environmental sustainability and smart living.

The AquaXpert Smart Water Management System provides an effective and intelligent solution for automating the monitoring and control of household or industrial water tanks. By integrating sensors, a microcontroller, and a web-based interface, the system ensures optimal water usage, prevents wastage through overflow control, and enables remote monitoring and management. The use of IoT technology enhances user convenience and system efficiency, reducing the need for manual supervision and promoting sustainable water conservation practices. AquaXpert thus represents a modern, reliable, and user-friendly approach to smart water management.

# CHAPTER 9

# FUTURE SCOPE

The AquaXpert system, while highly functional in its current version, offers significant potential for enhancements and broader applications. The following future developments can be implemented to make the system more advanced, scalable, and efficient:

**Mobile App Integration**

- Develop a dedicated Android/iOS app for real-time monitoring, remote control of valves, and receiving alerts.
- Include dashboard analytics for user-friendly visualization of water usage.

**Cloud-Based Data Storage & AI Analytics**

- Integrate cloud platforms (like Firebase or AWS) to store historical data securely.
- Use AI and machine learning to analyze usage patterns, detect anomalies, and predict future consumption.

**Voice Assistant Integration**

- Integrate with smart assistants like Alexa, Google Assistant, or Siri for voice-based status checks and control.

# APPENDIX

# SOURCE CODING

```
digitalWrite(MOTOR_RELAY, HIGH);  // Relay OFF
digitalWrite(VALVE_RELAY, HIGH);  // Relay OFF


pinMode(FLOW_SENSOR, INPUT_PULLUP);

attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR), flowISR, FALLING);



WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi...");
while (WiFi.status() != WL_CONNECTED) {
  delay(500); Serial.print(".");
}

Serial.println("\nConnected!");

Serial.print("IP Address: "); Serial.println(WiFi.localIP());



// Setup routes
server.on("/", handleRoot);
server.on("/motorOn", motorOn);
server.on("/motorOff", motorOff);
server.on("/valveOpen", valveOpen);
server.on("/valveClose", valveClose);
server.begin();


Serial.println("Web Server Running.");
```

```
}


void loop() {
  server.handleClient();

  // Flow rate calculation every 2 seconds
  if (millis() - lastFlowCheck >= 2000) {
    flowRate = (flowPulseCount / 7.5);
    Serial.print("[FLOW] Rate: "); Serial.print(flowRate); Serial.println(" L/min");
    flowPulseCount = 0;
    lastFlowCheck = millis();
  }


  // Tank level logic
  int distance = sonar.ping_cm();
  if (distance > 0) {
    Serial.print("[LEVEL] Tank Distance: "); Serial.print(distance); Serial.println(" cm");


    if (distance >= EMPTY_LEVEL) {
      digitalWrite(MOTOR_RELAY, LOW);
      Serial.println("[AUTO] Tank Low. Motor ON.");
    } else if (distance <= FULL_LEVEL) {
      digitalWrite(MOTOR_RELAY, HIGH);
      Serial.println("[AUTO] Tank Full. Motor OFF.");
    }
  }
}
```

# REFERENCES


## Books

1. **"Internet of Things: Architecture and Applications"** – Rajkumar Buyya, Amir Vahid Dastjerdi
   *Covers IoT system design and smart home applications..*
2. **"Smart Homes and Health Telematics"** – Springer Lecture Notes
   *Discusses smart home systems with real-world case studies.*

## Research Papers / Journals

1. **Smart Water Management in Smart Cities using IoT**
   *International Journal of Scientific Research in Engineering (IJSRE), 2020*
   [Available on Google Scholar]
2. **Design and Implementation of IoT Based Water Monitoring System**
   *International Research Journal of Engineering and Technology (IRJET), 2019*
3. **IoT-Based Smart Water Management System for Home Automation**
   *Published in IEEE Xplore or Scopus Indexed Journals.*

## Web References

1. **Arduino Official Documentation** – https://www.arduino.cc/reference/en/
   *For syntax, functions, and libraries.*
2. **Raspberry Pi Documentation** – https://www.raspberrypi.com/documentation/
   *For hardware and GPIO programming.*
3. **Blynk IoT Platform** – https://blynk.io/
   *For creating remote monitoring dashboards.*
4. **ThingSpeak Cloud Platform** – https://thingspeak.com/
   *Cloud analytics and data visualization for IoT projects.*
5. **MIT App Inventor** – https://appinventor.mit.edu/
   *If you create a mobile app for AquaXpert.*