IOT-BASED WATER MANAGEMENT SYSTEMS: SURVEY AND FUTURE RESEARCH DIRECTION

A PROJECT REPORT

Submitted by

DURGA DEVI M	**[723721106011]**
INIYASREE B	**[723721106016]**
MOUNICA S	**[723721106030]**
VANATHI K V	**[723721106051]**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

V.S.B. COLLEGE OF ENGINEERING TECHNICAL CAMPUS
(An Autonomous Institution)
COIMBATORE-642109, TAMILNADU.

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

ANNAUNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**IOT BASED WATER MANAGEMENT SYSTEMS: SURVEY AND FUTURE RESEARCH DIRECTION**" is the bonafide work of the following students, **DURGADEVI M [723721106011], INIYASREE B [723721106016], MOUNICA S [723721106030], VANATHI K V [723721106051]** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Mrs. P. LATHA, M.E., (Ph.D.),	**Mrs. G. JEYALAKSHMI,** **M. Tech, (Ph.D.).,**
HEAD OF THE DEPARTMENT,	**SUPERVISOR,** Assistant Professor,
Department of Electronics and Communication Engineering,	Department of Electronics and Communication Engineering,
V.S.B College of Engineering Technical Campus,	V.S.B College of Engineering Technical Campus,
Coimbatore-642109	Coimbatore-642109

Submitted for the university examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First of all, we extend our heartfelt gratitude to the management of V.S.B College of Engineering Technical Campus institutions, Chairman **Mr. V. S. BALSAMY B.Sc., L.L.B.,** for providing us with all sort of support in the completion of this project.

We record our indebtedness to our principal **Dr. V. VELMURUGAN, M.E., Ph.D.,** for his guidance and sustained encouragement for the successful of the project.

We also record our indebtedness to our vice principal **Dr. P. RAJU, M.E., Ph.D.,** for his guidance and sustained encouragement for the successful of the project.

We are highly grateful to **Mrs. P. LATHA, M.E., (Ph.D.).,** Head of the department, Department of Electronics and Communication and Engineering for her valuable suggestions throughout the course of this project

We owe our gratitude to our project coordinator **Dr. N. RAMYA RANI, M.E., Ph.D., and Dr. A. S. NARMADHA, M.E., Ph.D.,** for her unlisted encouragement and more over for his timely support and guidance till the completion of the project work

We also extend our sincere thanks to **Mrs. G. JEYALAKSHMI, M. Tech, (Ph.D.).,** project guide his kind support and timely assistance rendered which lead us in successful completion of our project.

We also extend our heartfelt salutation to our beloved parents and friends who have always been an integral part in helping us through tough times and all teaching and non-teaching staff for providing their moral support making herculean success of our project

VISION AND MISSION STATEMENTS OF THE INSTITUTE

Vision:

We endeavor to impart futuristic technical education of the highest quality in the student community to inculcate discipline in them to face the world with self- confidence and thus we prepare them for life responsible citizens to uphold human values and to be of services at large. We strive to bring up the Institution as an Institution of Academic excellence of International standard.

Mission:

We transform persons into personalities by the state-of-the-art infrastructure, time consciousness, quick response and the best academic practices through assessment and advice.

VISION AND MISSION STATEMENTS OF THE DEPARTMENT

Name of the Department: Department of Electronics and Communication Engineering

Vision:

To become a forerunner in producing skilled graduates with strong foundation in Electronics and Communication Engineering who can contribute significantly to the society.

Mission:

- To provide excellent infrastructure and enriched curriculum to train and develop highly competent engineers with research aptitude
- To foster the skills of employ ability and entrepreneurship along with social responsibility among the students transforming them into intellectual professionals to support nation's growth.
- To motivate the students in gaining knowledge about the modern technologies to meet the dynamic industrial needs supporting lifelong learning.

PEO, PO and PSO statements

Program Educational Objectives (PEO):

PEO1-Graduates will have sand knowledge in the core as of Electronics, Communication and allied Engineering to analyze and propose solutions for day-to-day engineering problems facing the society.

PEO2-Graduates will have necessary skills to do research and work as an entrepreneur with self- confidence in interdisciplinary fields of Engineering.

PEO3-Graduates will have managerial and interpersonal skills to work as an individual and as a team with continuous learning to boost their organization's growth.

PEO4-Graduates will have self-discipline, ethical values and social responsibility setting a good role model for future generations.

Program Outcomes (POs):

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, emulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design the system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need.

8. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

9. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

10. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

11. Life Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

12. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

Program Specific Outcomes (PSOS):

PSO1-Students will be trained in the field of electronics systems, designs and develop solutions enhancing their employ-ability and entrepreneurship skills.

PSO2-Students will be inspired to develop analytical, domain and soft skills which will able them to crack national level competitive examinations for higher studies and employment.

RELEVANCE TO PO'S AND PSO'S:

DESCRIPTION	PO'S MAPPING	PO'S MAPPING
Real-time monitoring of water consumption using sensors and IoT technologies	PO1, PO2, PO5	PSO1, PSO2
RFID-based water tracking and automated billing system for individual accountability	PO3, PO9, PO10	PSO1, PSO2
Continuous pH-based water quality monitoring and cloud-based reporting via Blynk application	PO4, PO5	PSO1
Smart regulation of water supply using motorized systems based on usage trends	PO1, PO3, PO10	PSO2
Promoting sustainable consumption through technology-driven accountability mechanisms	PO6, PO7, PO8	PSO1, PSO2

ABSTRACT

The increasing global demand for efficient water management has driven the adoption of Internet of Things (IoT) technologies, enabling real-time monitoring, resource optimization, and quality assurance. This paper presents a comprehensive survey of IoT-based water management systems, emphasizing key parameters such as water consumption tracking, quality monitoring, and automated regulation. The proposed system integrates advanced IoT components, including an ESP32 microcontroller, pH sensor, RFID-based tracking, and a motorized water distribution mechanism, to enhance efficiency in urban water management. To ensure precise water consumption measurement, the system employs sensors that track usage in real time. An RFID reader and tag system is incorporated to monitor individual water consumption and facilitate payment for excess usage beyond a predefined threshold. This automated billing mechanism enhances accountability and encourages sustainable water consumption practices. Additionally, a pH sensor continuously evaluates water quality, transmitting real-time acidity data to the ESP32 microcontroller, which relays the information to a cloud-based platform or user interface via the Blynk application. This ensures continuous monitoring and alerts users of any deviations from safe water quality standards. The system also features a motor driver and water motor that regulate water supply based on demand and consumption trends, effectively minimizing manual intervention. By integrating IoT with smart water management strategies, this system not only improves resource allocation and quality control but also provides a scalable framework for future enhancements in urban and industrial water management.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF ABBEREVIATION

VSI	VOLTAGE SOURCE INVERTERS
ISI	IMPEDANCE SOURCE INVERTERS
SBI	SWITCHED BOOST INVERTER
CFSI	CURRENT FED SWITCHED INVERTER
VRM	VOLTAGE REGULATOR MODULES
PWM	PULSE WIDTH MODULATION
ICFSI	INTERLEAVED TOPOLOGY FOR CFSI
THD	TOTAL HARMONIC DISTORTION
MPPT	MAXIMUM POWER POINT TRACKING
SE	SOLAR ENERGY
AC	ALTERNATING CURRENT
DC	DIRECT CURRENT
SPWM	SINUSOIDAL PULSE WIDTH MODULATION
ZVS	ZERO VOLTAGE SWITCHING
ZCD	ZERO CROSSING DETECTOR
EEPROM	ELECTRICALLY ERASABLE PROGRAMMALE READ ONLY MEMORY

CHAPTER 1
INTRODUCTION

Water is one of the most essential natural resources for sustaining life and supporting a wide range of ecological, economic, and societal functions. Its significance extends across various domains, including agriculture, industry, domestic use, and environmental preservation. However, in recent decades, the world has been facing a severe water crisis driven by rapid population growth, unplanned urbanization, industrial pollution, and the effects of global climate change. These factors have significantly reduced the availability of clean and usable water, putting pressure on existing water management systems. In many regions, water scarcity has become a persistent challenge, leading to increased conflicts over its usage and raising concerns about long-term sustainability.

Traditional water management approaches, which often depend on manual monitoring, fixed schedules, and human intervention, are no longer adequate to address these modern-day challenges. These systems typically lack real-time feedback, automation, and accurate consumption tracking, which leads to issues such as water leakage, overuse, wastage, and delayed response to quality deterioration. As a result, there is a growing need to shift toward smart water management solutions that can deliver greater precision, automation, and intelligence.

The integration of the Internet of Things (IoT) into water management systems has emerged as a promising and innovative solution to modern water-related issues. IoT refers to a network of interconnected physical devices equipped with sensors and software that enable the collection, exchange, and analysis of real-time data. In the context of water management, IoT devices can be deployed to monitor various parameters such as flow rate, water level, quality indicators, and user consumption. These systems can operate autonomously, reducing the need for manual intervention while enhancing operational efficiency and transparency.

This project proposes an IoT-based smart water management system designed to monitor water consumption, assess water quality, and control water distribution intelligently. The system is built using an ESP32 microcontroller, which serves as the central controller for processing sensor inputs and communicating with a cloud platform. To measure water consumption accurately, the system employs digital flow sensors that collect real-time data on the amount of water used. An RFID-based tracking system is incorporated to associate water usage with specific users, allowing individual consumption monitoring and the implementation of a structured billing mechanism. This promotes user accountability and encourages the conservation of water by making users more aware of their consumption behaviour.

In addition to monitoring usage, the system features a pH sensor that continuously evaluates water quality by measuring its acidity or alkalinity. This parameter is critical in determining the potability and safety of water for domestic or industrial use. If the pH value deviates from the acceptable range, the system can generate alerts, enabling timely intervention to prevent health risks. All sensor data is transmitted to the cloud via Wi-Fi, and users can access real-time data and alerts through the Blynk mobile application. This ensures transparency and allows users to monitor their water usage and quality remotely at any time.

Another significant component of the system is the integration of a motorized water control mechanism. Using a motor driver and water motor, the system can automatically regulate the water supply based on consumption patterns and pre-set conditions. This feature eliminates the need for manual operation and ensures that water distribution is optimized according to actual demand. For instance, during low-demand periods, the system can reduce flow to conserve water and energy, while ensuring adequate supply during peak hours. This level of automation helps improve resource utilization, reduces operational costs, and ensures a more sustainable approach to water distribution.

Furthermore, the proposed system supports scalability and adaptability, making it suitable for various applications such as residential communities, apartment complexes, industrial facilities, and even rural water distribution schemes. Its modular design allows for future enhancements, such as the inclusion of additional water quality sensors (e.g., turbidity, total dissolved solids), integration with solar power systems for off-grid operation, and the application of artificial intelligence for predictive analytics and system optimization.

Overall, the implementation of IoT in water management offers significant advantages, including enhanced accuracy, improved efficiency, real-time monitoring, user engagement, and better control over distribution and quality. This project aims to demonstrate a working prototype of such a system, contributing toward smarter, more sustainable, and more accountable water usage practices. As urbanization and water demand continue to increase, solutions like the one proposed in this project will play a crucial role in building resilient water infrastructure for future generations.

.

CHAPTER 2

LITERATURE SURVEY

1. Surface Water Monitoring Systems—The Importance of Integrating Information Sources for Sustainable Watershed Management

Authors: Yuli Sudriani, Viktor Sebestyén, János Abonyi (2023)

Mechanism Used:

This study introduces a comprehensive framework that integrates multiple data sources, water monitoring techniques, and predictive models to support decision-making in watershed management. It combines remote sensing data, in-situ sampling, IoT sensors, and simulation models to assess environmental conditions in surface water systems.

Advantages:

- Provides a holistic view of the watershed by integrating different data types.
- Enhances the accuracy and reliability of environmental assessments.

Drawbacks:

- Requires high technical expertise and computational power to manage and interpret the large amount of integrated data.

2. IoT Based Water Consumption Monitoring System for Water Management

Authors: Alfred E. Clemente, Reinier Miguel G. Samaniego, Febus Reidj G. Cruz (2023)

Mechanism Used:

This system utilizes three flow sensors strategically installed at the sink, toilet, and bidet to monitor water consumption in real-time. It uses microcontrollers to gather data from sensors and sends the information to a central interface for analysis and feedback.

Advantages:

- User-specific consumption data helps in identifying water usage patterns.
- Promotes awareness and accountability in daily water usage.

Drawbacks:

- Limited to indoor applications (primarily lavatories).
- Does not include water quality monitoring or automation of water supply.

3. Smart Automatic Water Management System with RF Communication and Remote Monitoring

Authors: Imon Deb Nath, Md. Naeem Uddin Novel, Md. Fakwer Uddin Mazumder, Mohammed Abdul Kader (2022)

Mechanism Used:

This system monitors water levels in roof and reserve tanks using ultrasonic sensors and water level sensors**.** The collected data is sent via RF (Radio Frequency) communication, and the pump is automatically turned on or off based on real-time tank levels.

Advantages:

- Automation of pump control reduces the need for human supervision.
- Energy and water-saving through optimized water level management.

Drawbacks:

- Use of RF communication limits the system's range and scalability.
- Focused only on storage and distribution; lacks usage tracking and water quality monitoring.

4. Water Quality Management Using Hybrid Machine Learning and Data Mining Algorithms: An Indexing Approach

Authors: Bilal Aslam, Ahsen Maqsood, Ali Hassan Cheema, Fahim Ullah, Abdullah Alharbi, Muhammad Imran (2022)

Mechanism Used:

This study involves the development of Water Quality Index (WQI) prediction models using hybrid machine learning algorithms such as Random Tree, Random Forest, M5P, and REPT. Water samples from wells are analyzed, and machine learning is applied to predict water quality trends.

Advantages:

- Highly accurate predictive models for water quality evaluation.
- Supports data-driven decision-making and risk prevention in water resource management.

Drawbacks:

- Depends on large datasets and proper training, which may not be available in all areas.
- Implementation requires technical knowledge in data science and machine learning.

5. IoT Innovations in Sustainable Water and Wastewater Management and Water Quality Monitoring: A Comprehensive Review

Authors: Ahmad Alshami, Eslam Ali, Moustafa Elsayed, Abdelrahman E. E. Eltoukhy, Tarek Zayed (2022)

Mechanism Used:

This paper reviews a variety of IoT-based solutions applied in water and wastewater management, including sensor networks**,** edge devices**,** cloud platforms, and AI integration for monitoring and control. It covers applications in smart cities, industrial wastewater treatment, and environmental monitoring.

Advantages:

- Provides a broad perspective on IoT applications in water management.
- Emphasizes scalability, automation, and remote access in smart water systems.

Drawbacks:

- Being a review, it lacks implementation-specific details or hardware specifications.
- Generalized findings may not be directly applicable to a specific use case without adaptation.

CHAPTER 3

EXISTING SYSTEM

The current water level detection methods primarily focus on monitoring and preventing water overflow in household environments. These systems employ sensors such as flow sensors, ultrasonic sensors, presence sensors, and water level sensors, along with microcontrollers and audio-visual alarm systems, to detect and manage water usage. However, despite these implementations, existing systems still suffer from inefficiencies in control mechanisms, leading to frequent water overflow and wastage.

One of the primary methods used is a flow sensor, which detects the amount of water passing through a faucet or pipe. This sensor, combined with an ultrasonic sensor, is designed to measure water levels and detect possible overflow conditions. In case of excess water usage, an audio-visual alarm system is triggered to alert users about potential spillage. Although these alarms provide real-time alerts, they do not prevent overflow proactively, making manual intervention necessary.

Another approach integrates a presence sensor and a water level sensor to monitor water usage in bathtubs or similar household applications. The presence sensor detects human movement, ensuring that water flow is regulated when a person enters the bathtub. Simultaneously, the water level sensor measures the water level to prevent excessive filling. This method aims to improve water conservation, but its scope is limited to household applications and does not extend to broader water management scenarios such as irrigation, industrial water systems, or large-scale urban water distribution networks.

Additionally, existing water consumption monitoring systems mainly focus on tracking wastewater and water levels within a household. While these systems provide insights into water wastage, they lack automation for controlling or optimizing water usage. This means users are responsible for taking corrective actions based on the data provided. Without an automated intervention mechanism, water overflow remains a challenge.

In summary, while existing systems incorporate multiple sensors and microcontroller-based solutions for water monitoring, they fall short in achieving efficient water conservation.

They primarily rely on monitoring rather than active control, requiring user intervention for effective water management. An advanced IoT-based system with automated regulation, remote monitoring, and real-time data processing would be necessary to enhance water conservation efforts and prevent unnecessary wastage.

3.1 DISADVANTAGES

Lack of Automated Control

- The existing system primarily focuses on monitoring water levels and providing alerts when overflow is detected. However, it lacks an automated intervention mechanism to control water flow and prevent excessive usage. Users must manually turn off the faucet or adjust water levels, which may lead to delays in response and continued water wastage.

Limited Scope of Application

- The current system is designed mainly for household water monitoring, particularly for detecting spills from faucets and bathtubs. It does not effectively extend to larger-scale applications, such as urban water supply systems, agricultural irrigation, or industrial water management, where automated regulation is crucial.

Reactive Instead of Preventive Approach

- Most existing systems function reactively, alerting users after an overflow or excessive water usage has already occurred. This means that water wastage still takes place before corrective actions can be implemented. A more effective system would involve a preventive approach, where real-time adjustments to water flow prevent wastage in the first place.

Dependence on User Intervention

- Since the system relies on alarms and notifications, users must manually respond to these alerts to stop unnecessary water flow. If the user is unavailable or inattentive, water will continue to be wasted despite the warning signals. A fully automated system would minimize the need for human intervention.

CHAPTER 4
SYSTEM IMPLEMENTATION

4.1 PROPOSED SYSTEM

The proposed IoT-based water management system is designed to provide a smart, automated, and scalable solution for efficient water usage, quality monitoring, and regulated distribution. At the core of the system lies the ESP32 microcontroller, which functions as the central processing and communication unit. The ESP32 is a powerful, low-cost microcontroller with built-in Wi-Fi and Bluetooth capabilities, enabling seamless real-time monitoring, remote control, and cloud-based data transmission. This allows users and administrators to access water usage data and quality reports from anywhere using a mobile application, specifically through the Blynk platform.

To measure water consumption, IoT-enabled sensors are used to track real-time water flow to households or user-defined areas. These sensors continuously send usage data to the ESP32, which processes the information and uploads it to the cloud. The system also features an RFID (Radio Frequency Identification) module, which enables personalized monitoring and billing. Each user is provided with an RFID tag that logs their individual water usage. When a user’s water consumption exceeds a predefined threshold, the system automatically generates a charge for the excess usage. This charge is reflected in the user’s profile on the mobile app, through which payment can be made digitally, thus promoting transparency and accountability.

For water quality assurance, a pH sensor is integrated into the system to monitor the acidity or alkalinity of the water. The sensor is directly connected to the ESP32, which evaluates the incoming data and sends real-time updates to the user interface. If the pH level falls outside the safe range, the system generates an alert through the Blynk app, enabling quick action to prevent health risks.

In terms of water distribution, the system includes a motor driver and a water pump, which regulate the water supply based on real-time demand and consumption patterns. When water demand is detected or a user requests water through their RFID interaction, the ESP32 activates the motor through the motor driver circuit to deliver water. Once the required amount is delivered, the motor automatically turns off, ensuring energy and water are conserved. This automated regulation reduces human intervention and enhances system efficiency.

Overall, this proposed system integrates hardware and software components to create a complete water management solution that is capable of monitoring, controlling, and billing in an automated and intelligent manner. By enabling real-time visibility into consumption and quality, along with digital payment facilities and remote control, the system not only optimizes resource utilization but also promotes sustainable water usage in urban and industrial environments.

4.2 ADVANTAGES

- Real-Time Water Monitoring and Control
- Automated Water Distribution
- Smart Water Consumption Tracking and Billing
- Remote Monitoring and Accessibility
- Water Quality Assurance
- Efficient Water Conservation

4.3 WORKING MODEL

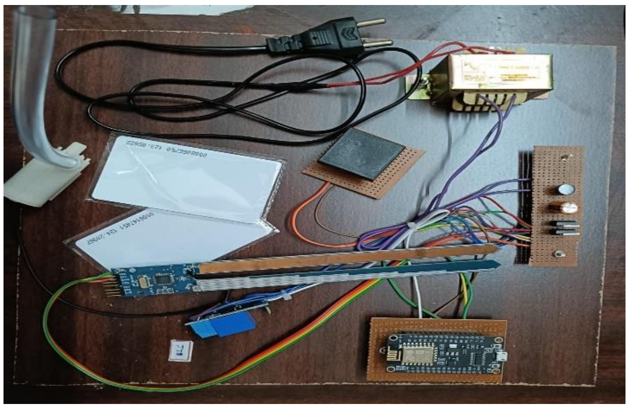


FIG 4.1 WORKING MODEL

The proposed system leverages IoT to enhance water management by interconnecting components for real-time monitoring, efficient allocation, and quality enforcement. An ESP32 microcontroller, which facilitates Wi-Fi connectivity, serves as the control center, directly connecting to some components while interfacing with others through intermediaries. The ESP32 enables remote monitoring and control of the entire water management system by collecting sensor data and transmitting it to a cloud platform or user interface, providing real-time access and alerts. Users can monitor water usage, quality, and consumption remotely. An RFID system tracks additional water consumption and manages payments for usage beyond a predefined threshold, assigning each user an

RFID tag to monitor personal water usage and generate bills when limits are exceeded. A pH sensor monitors water quality, sending readings to the ESP32 and subsequently to the Blynk app. Additionally, a motor driver and water motor control water flow to various locations, adjusting according to demand and consumption, thereby enabling automated water supply with minimal manual intervention. The system also allows users to track water consumption and make payments for additional usage via mobile phone.

4.4 BLOCK DIAGRAM

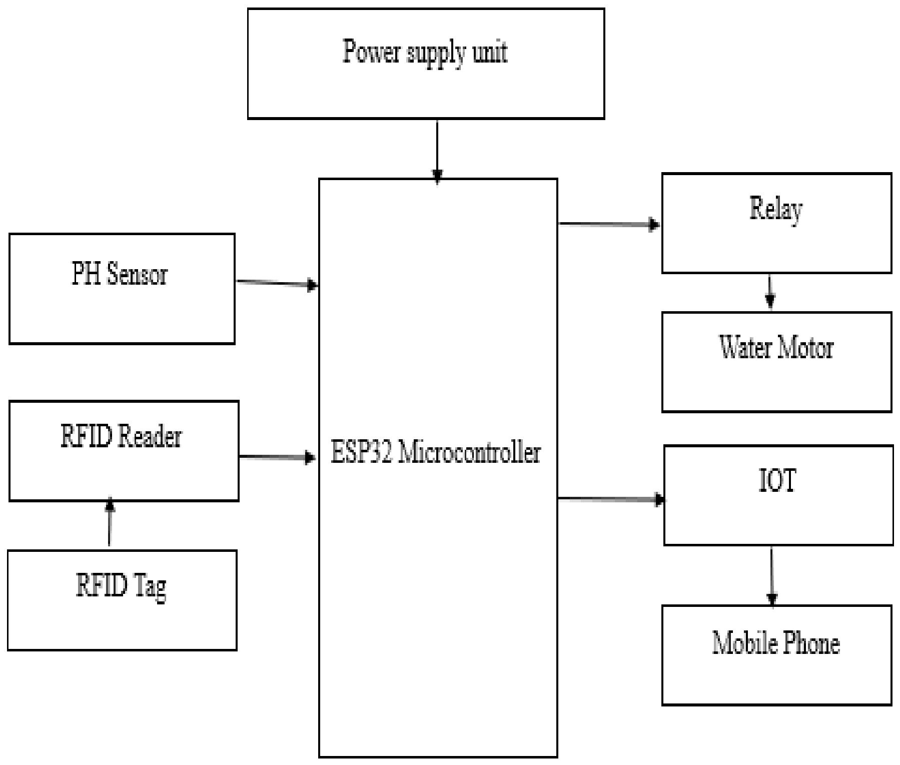


FIG 4.2 BLOCK DIAGRAM

4.5 WORKING PRINCIPLE

The working principle of the proposed IoT-based water management system revolves around real-time monitoring, automated control, and intelligent decision-making enabled by the ESP32 microcontroller. The system begins with a regulated power supply that converts AC current into a stable DC voltage using a transformer, bridge rectifier, capacitor, and voltage regulator. This stable voltage powers the ESP32 microcontroller and other connected components.

When a user wants to access water, they first scan their RFID tag on the RFID reader. The reader captures the tag's unique ID and sends it to the ESP32. The microcontroller verifies the user's identity and checks whether the user is within their allowed consumption limit. If the limit is exceeded, the system records the extra consumption and prepares an automated billing alert via the IoT module. This promotes responsible water usage and discourages wastage.

Simultaneously, a pH sensor is used to monitor the water quality. It measures the acidity or alkalinity of the water and sends the data to the ESP32. If the pH level falls outside of safe limits, the system triggers a warning through the Blynk mobile application, allowing immediate corrective action to ensure that users only receive safe, clean water.

Once the user is authenticated and the water quality is verified, the ESP32 sends a control signal to the relay module. The relay acts as a switch to turn on the water motor, which then supplies water to the user. After the required volume is delivered, or the user session ends, the ESP32 instructs the relay to turn off the motor, thereby stopping the water flow. This reduces water waste and minimizes the need for manual intervention.

All sensor data, including usage logs and water quality reports, are transmitted by the ESP32 to the IoT cloud platform, which is accessible through a mobile phone using the Blynk application. This interface allows users to monitor their consumption in real-time, get alerts, and pay for extra usage digitally.

In essence, the system works through a combination of hardware components (RFID, sensors, relay, motor) and software (ESP32 firmware, Blynk app, IoT cloud integration) to provide a complete, smart, and automated solution for water management.

CHAPTER 5

HARDWARE SPECIFICATION

5.1 Hardware Requirements

5.1.1 Power Supply Unit

Power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

Power supplies for electronic devices can be broadly divided into linear and switching power supplies. The linear supply is a relatively simple design that becomes increasingly bulky and heavy for high current devices; voltage regulation in a linear supply can result in low efficiency. A switched-mode supply of the same rating as a linear supply will be smaller, is usually more efficient, but will be more complex.

Linear Power Supply

An AC powered linear power supply usually uses a transformer to convert the voltage from the wall outlet (mains) to a different, usually a lower voltage. If it is used to produce DC, a rectifier is used. A capacitor is used to smooth the pulsating current from the rectifier. Some small periodic deviations from smooth direct current will remain, which is known as ripple. These pulsations occur at a frequency related to the AC power frequency (for example, a multiple of 50 or 60 Hz).

The voltage produced by an unregulated power supply will vary depending on the load and on variations in the AC supply voltage. For critical electronics applications a linear regulator will be used to stabilize and adjust the voltage. This regulator will also greatly reduce the ripple and noise in the output direct current. Linear regulators often provide current limiting, protecting the power supply and attached circuit from over current.

Adjustable linear power supplies are common laboratory and service shop test equipment, allowing the output voltage to be set over a wide range. For example, a bench power supply used

by circuit designers may be adjustable up to 30 volts and up to 5 amperes output. Some can be by circuit designers may be adjustable up to 30 volts and up to 5 amperes output. Some can be driven by an external signal, for example, for applications requiring a pulsed output.

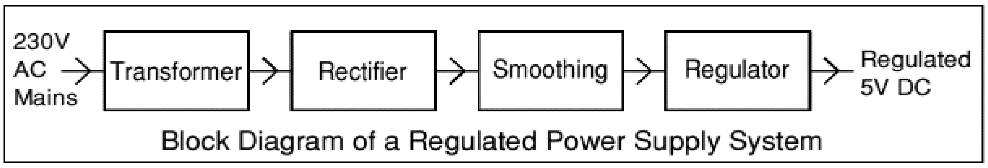


FIG 5.1 POWER SUPPLY UNIT

Transformers convert AC electricity from one voltage to another with little loss of power.

Transformers work only with AC and this is one of the reasons why mains electricity is AC.

Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in UK) to a safer low voltage.

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core.

Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up.

The ratio of the number of turns on each coil, called the turn's ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

Turns ratio=Vp/Vs=Nn/Ns and Power out=Power in

Vs*Is=Vp * Ip

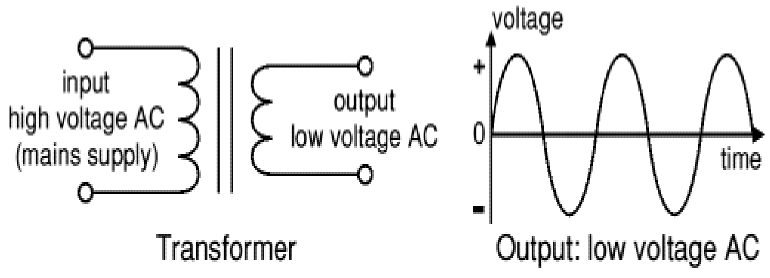


FIG 5.2 TRANSFORMER OUTPUT VOLTAGE

Rectifier:

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier

but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC.

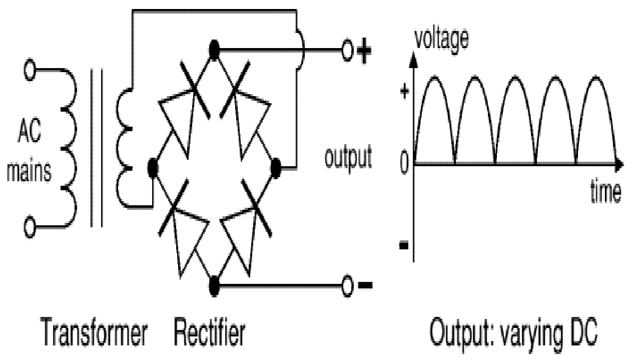


FIG 5.3 RECTIFIER CIRCUIT

The varying DC output is suitable for lamps, heaters and standard motors. It is not suitable for electronic circuits unless they include a smoothing capacitor.

Bridge Rectifier:

A bridge rectifier can be made using four individual diodes, but it is also available in special packages containing the four diodes required. It is called a full-wave rectifier because it uses the entire AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting, as shown in the diagram below. Bridge rectifiers are rated by the maximum current they can pass and the maximum reverse voltage they can withstand (this must be at least three times the supply RMS voltage so the rectifier can withstand the peak voltages).

Alternate pairs of diodes conduct, changing over the connections so the alternating directions of AC are converted to the one direction of DC.

Output: full-wave varying DC: (using the entire AC wave):

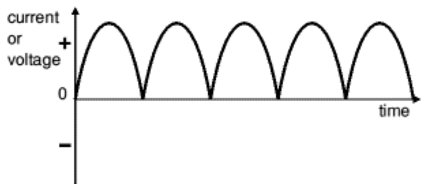


FIG 5.4 OUTPUT WAVE FORM

Single Diode Rectifier:

A single diode can be used as a rectifier but this produces half-wave varying DC which has gaps when the AC is negative. It is hard to smooth this sufficiently well to supply electronic circuits unless they require a very small current so the smoothing capacitor does not significantly discharge during the gaps.

5.1.2 ESP32 MICROCONTROLLER

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC- V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power- management modules. ESP32 is created and developed by Espressif Systems, a Shanghai- based Chinese company, and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 microcontroller.

FIG 5.5 ESP32 MICROCONTROLLER

Processors:

- CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
- Ultra low power (ULP) co-processor
- Memory: 320 KiB RAM, 448 KiB ROM
- Wireless connectivity:
- Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)
- Peripheral interfaces:
- 34 × programmable GPIOs
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DACs
- 10 × touch sensors (capacitive sensing GPIOs) × SPI
- 2 × I^{2}S interfaces
- 2 × I^{2}C interfaces
- 3 × UART
- SD/SDIO/CE-ATA/MMC/eMMC host controller
- SDIO/SPI slave controller
- Ethernet MAC interface with dedicated DMA and planned IEEE 1588 Precision Time Protocol support
- CAN bus 2.0
- Infrared remote controller (TX/RX, up to 8 channels)
- Motor PWM

- LED PWM (up to 16 channels)
- Hall effect sensor
- Ultra low power analog pre-amplifier

Security:

5.1.1.1.1 IEEE 802.11 standard security features all supported, including WPA, WPA2, WPA3 (depending on version) [5] and WLAN Authentication and Privacy Infrastructure (WAPI)

5.1.1.1.2 Secure boot

5.1.1.1.3 Flash encryption

5.1.1.1.4 1024-bit OTP, up to 768-bit for customers

5.1.1.1.5 Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

5.1.1.1.6 Power management:

5.1.1.1.7 Internal low-dropout regulator

5.1.3 PH SENSOR

This multi-purpose meter to help provide a healthy growing environment for all plants. It tests for soil alkalinity / acidity, soil moisture, and sunlight.

Features:

- 100% Brand New.
- Weight: 57g
- size:28.2 x 4.8 x 3.6cm
- Color: Green
- Probe length: 20cm
- The tool plant owners, gardeners can`t live without!
- 3 IN 1 moisture light &amp; PH meter Soil analyzer meter.
- For outdoor &amp; indoor plants, gardens &amp; grass lawn.
- Take the guess work out of your daily garden watering light and moisture.
- Save water, energy and keep your plants, lawn, flower in top condition.
- Measures moisture at root level.
- Prevents over and under watering
- Scientifically accurate.
- Easy to use, just insert and read.
- No battery required, simple and convenient to use
- Simply insert the meter into the soil, switch to the setting you want to measure and read the scale.

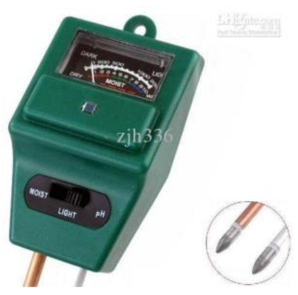


FIG 5.6 PH SENSOR

5.1.4 RFID READER

This module directly connects to any microcontroller UART or through a RS232 converter to PC. It gives UART/Wiegand26 output. This RFID Reader Module works with any 125 KHz RFID tags

SPECIFICATIONS

- 5VDC through USB (External 5V supply will boost range of the module)
- Current: <50mA
- Operating Frequency: 125Khz
- Read Distance: 10cm
- Size of RFID reader module: 32mm(length) * 32mm(width) * 8mm(height)

RS232 INTERFACE FORMAT:

10 ASCII DATA (card no.) + 2 ASCII DATA (XOR result)

E.g. Card number is 4500C5D1E9B8 read from reader then the card number on card will be as below. 45 - Preamble

00C5D1E9 value in Hex = 12964329. / B8 is XOR value for (45 XOR 00 XOR C5 XOR D1 XOR E9)

Hence number on the card is 0012964329.

1. Data baud rate: 9600 bps
2. Data bit 8 bits
3. Parity check: None
4. Stop bit

WORKING PRINCIPLE:

Many types of RFID exist, but at the highest level, we can divide RFID devices into two classes:

- Active

  Active tags require a power source they're either connected to a powered infrastructure or use energy stored in an integrated battery. In the latter case, a tag's lifetime is limited by the stored energy, balanced against the number of read operations the device must undergo. One example of an active tag is the transponder attached to an aircraft that identifies its national origin.

- Passive

  A passive tag consists of three parts:

  1. Antenna
  2. A semiconductor chip attached to the antenna
  3. Some form of encapsulation.

The tag reader is responsible for powering and communicating with a tag. The tag antenna captures energy and transfers the tag's ID (the tag's chip coordinates this process). The encapsulation maintains the tag's integrity and protects the antenna and chip from environmental conditions or reagents. The encapsulation could be a small glass vial or a laminar plastic substrate with adhesive on one side to enable easy attachment to goods.

Two fundamentally different RFID design approaches exist for transferring power from the reader to the tag: magnetic induction and electromagnetic (EM) wave capture. These two designs take advantage of the EM properties associated with an RF antenna—the near field and the far field. Both can transfer enough power to a remote tag to sustain its operation—typically between 10 W and 1 mW, depending on the tag type. (For comparison, the nominal power an Intel

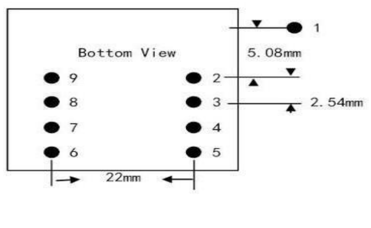


FIG 5.7 RFID READER

X Scale processor consumes is approximately 500 mW, and an Intel Pentium 4 consumes up to 50 W.) Through various modulation techniques, near- and far-field-based signals can also transmit and receive dat

5.1.5 RFID TAG

An RFID reader is a device that is used to interrogate an RFID tag. The reader has an antenna that emits radio waves; the tag responds by sending back its data.

An RFID tag is a microchip combined with an antenna in a compact package; the packaging is structured to allow the RFID tag to be attached to an object to be tracked. "RFID" stands for Radio Frequency Identification. The tag's antenna picks up signals from an RFID reader or scanner and then returns the signal, usually with some additional data (like a unique serial number or other customized information).

A passive tag is an RFID tag that does not contain a battery; the power is supplied by the reader. When radio waves from the reader are encountered by a passive rfid tag, the coiled antenna within the tag forms a magnetic field. The tag draws power from it, energizing the circuits in the tag. The tag then sends the information encoded in the tag's memory.

The RX and TX pins of RFID reader connected to Tx and Rx pins of 8051 Microcontroller respectively. Then the reader senses the data from the Tag and transmits the sensed data to microcontroller via serial port.

RFID can be used in a variety of applications, such as:

FIG 5.8 RFID TAG REALTIME

APPLICATIONS:

- Access management
- Tracking of goods
- Tracking of persons and animals
- Toll collection and contactless payment
- Machine readable travel documents
- Smart dust (for massively distributed sensor networks)
- Tracking sports memorabilia to verify authenticity
- Airport baggage tracking logistics[21]
- Timing sporting events

5.1.6 RELAY

- A relay is an electrically operated switch.
- Electric current through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts.
- The coil current can be on or off so relays have two switch positions and there are double- throw (changeover) switches.
- It consists of a coil of wire surrounding a soft iron core, an iron yoke, which provides a low reluctance path for magnetic flux, a movable iron armature, and a set, or sets, of contacts.
- In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open.

5.1.7 WATER MOTOR

A DC motor in simple words is a device that converts direct current (electrical energy) into mechanical energy. It's of vital importance for the industry today, and is equally important for engineers to look into the working principle of DC motor in details that has been discussed in this article. In order to understand the operating principle of dc motor we need to first look into its constructional feature.

The very basic construction of a dc motor contains a current carrying armature which is connected to the supply end through commutator segments and brushes and placed within the north south poles of a permanent or an electro-magnet as shown in the diagram below.

Now to go into the details of the operating principle of DC motor its important that we have a clear understanding of Fleming's left hand rule to determine the direction of force acting on the armature conductors of dc motor.

Fleming's left hand rule says that if we extend the index finger, middle finger and thumb of our left hand in such a way that the current carrying conductor is placed in a magnetic field (represented by the index finger) is perpendicular to the direction of current(represented by the middle finger), then the conductor experiences a force in the direction (represented by the thumb) mutually perpendicular to both the direction of field and the current in the conductor.

STEP1:

Initially considering armature is in its starting point or reference position where the angle α = 0.

$$\therefore\ \tau = BILw \times \cos 0^\circ = BILw$$

Since α = 0, the term cos α = 1, or the maximum value, hence torque at this position is maximum given by τ = BILw. This high starting torque helps in overcoming the initial inertia of rest of the armature and sets it into rotation.

STEP2:

Once the armature is set in motion, the angle α between the actual position of the armature and its reference initial position goes on increasing in the path of its rotation until it becomes 90° from its initial position. Consequently the term cosα decreases and also the value of torque. The torque in this case is given by τ = BILw cosα which is less than BIL w when α is greater than 0°.

STEP3:

In the path of the rotation of the armature a point is reached where the actual position of the rotor is exactly perpendicular to its initial position, i.e. α = 90°, and as a result the term cosα = 0. The torque acting on the conductor at this position is given by, i.e. virtually no rotating torque acts on the armature at this instance. But still the armature does

$$\therefore\ \tau = BILw \times \cos 90^\circ = 0$$

Since it is not come to a standstill, this is because of the fact that the operation of dc motor has been engineered in such a way that the inertia of motion at this point is just enough to overcome this point of null torque. Once the rotor crosses over this position the angle between the actual position of the armature and the initial plane again decreases and torque starts acting on it again.

FIG 5.9 WATER MOTOR

CHAPTER 6

SOFTWARE SPECIFICATION

6.1 SOFTWARE REQUIREMENTS

6.1.1 ARDUINO IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

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 right hand 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.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

✓ Verify

Checks your code for errors compiling it.

➜ Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbookmenu instead.

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

FILE

- New

  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open

  Allows to load a sketch file browsing through the computer drives and folders.
- Open Recent

  Provides a short list of the most recent sketches, ready to be opened.
- Sketchbook

  Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

- Examples

  Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

- Close

  Closes the instance of the Arduino Software from which it is clicked.

- Save

  Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

- Save as...

  Allows saving the current sketch with a different name.

- Page Setup

  It shows the Page Setup window for printing.

- Print

  Sends the current sketch to the printer according to the settings defined in Page Setup.

- Preferences

  Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

- Quit

  Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

EDIT

- Undo/Redo

  Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

- Cut

  Removes the selected text from the editor and places it into the clipboard.

- Copy

  Duplicates the selected text in the editor and places it into the clipboard.

- Copy for Forum

  Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

- Copy as HTML

  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

- Paste

  Puts the contents of the clipboard at the cursor position, in the editor.
- Select All

  Selects and highlights the whole content of the editor.
- Comment/Uncomment

  Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent

  Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find

  Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next

  Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous

  Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

SKETCH

- Verify/Compile

  Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload

  Compiles and loads the binary file onto the configured board through the configured Port.
- Upload Using Programmer

  This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- Export Compiled Binary

  Saves a .hex file that may be kept as archive or sent to the board using other tools.

- Show Sketch Folder

  Opens the current sketch folder.
- Include Library

  Adds a library to your sketch by inserting #include statements at the start of your code. For more details, seelibraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File...

  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

TOOLS

- Auto Format

  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- Archive Sketch

  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- Fix Encoding & Reload

  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial Monitor

  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board

  Select the board that you're using. See below for descriptions of the various boards.
- Port

  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer

  For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

- Burn Bootloader

  The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

HELP

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

- Find in Reference

  This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

SKETCHBOOK

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino. Tabs, Multiple Files, and Compilation . Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

UPLOADING

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or/dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for . USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx ,/dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the File menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

LIBRARIES

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #includestatements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library. To write your own library, see this tutorial.

THIRD-PARTY HARDWARE

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third- party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

SERIAL MONITOR

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor. You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

PREFERENCES

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

LANGUAGE SUPPORT

0

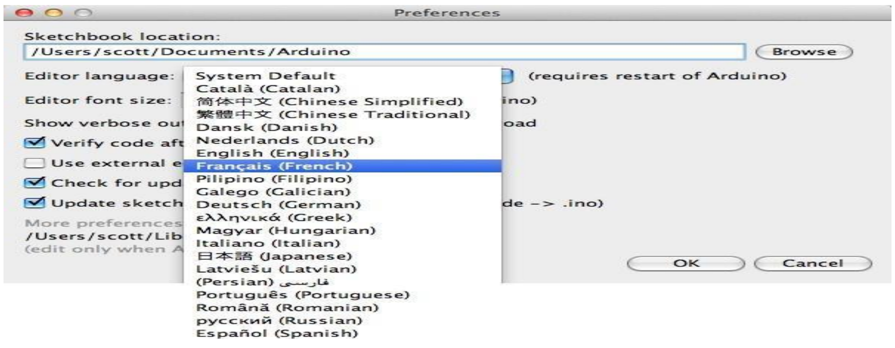


FIG: 6.1 SOFTWARE

Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selectingSystem Default from the Editor Language drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

BOARDS

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find a comparison table between the various boards here.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. TheBoards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

- Arduino Yùn

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino/Genuino Uno

  An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Diecimila or Duemilanove w/

ATmega168 An ATmega168 running at 16

MHz with auto-reset.

- Arduino Nano w/ ATmega328

  An ATmega328 running at 16 MHz with auto-reset. Has eight analog inputs.

- Arduino/Genuino Mega 2560

  An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino Mega

  An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino Mega ADK

  An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino Leonardo

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino Micro

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino Esplora

  An ATmega32u4 running at 16 MHz with auto-reset.

- Arduino Mini w/ ATmega328

  An ATmega328 running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.

- Arduino Ethernet

  Equivalent to Arduino UNO with an Ethernet shield: An ATmega328 running at 16

  MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

- Arduino Fio

  An ATmega328 running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini

  (3.3V, 8 MHz) w/ATmega328, 6 Analog In, 14 Digital I/O and 6 PWM.

- Arduino BT w/ ATmega328

  ATmega328 running at 16 MHz. The bootloader burned (4 KB) includes codes to initialize

  the on- board bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM..

- LilyPad Arduino USB

  An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.

- LilyPad Arduino

An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328

  An ATmega328 running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328; 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino NG or older w/ ATmega168

  An ATmega168 running at 16 MHz without auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Robot Control

  An ATmega328 running at 16 MHz with auto-reset.
- Arduino Robot Motor

  An ATmega328 running at 16 MHz with auto-reset.
- Arduino Gemma

  An ATtiny85 running at 8 MHz with auto-reset, 1 Analog In, 3 Digital I/O and 2 PWM.

6.1.2 EMBEDDED C

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

A Technical Report was published in 2004 and a second revision in 2006.

NECESSITY

During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check for correct execution of the program. Some 'very fortunate' developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well. As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

As assembly language programs are specific to a processor, assembly language didn't offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn't find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

ADVANTAGES

- It is small and simpler to learn, understand, program and debug.

1. Compared to assembly language, C code written is more reliable and scalable, more portable between different platforms.
2. C compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.

3. Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
4. As C combines functionality of assembly language and features of high level languages, C is treated as a ‘middle-level computer language’ or ‘high level assembly language’.

EMBEDDED SYSTEMS PROGRAMMING

Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:

1. Embedded devices have resource constraints(limited ROM, limited RAM, limited stack space, less processing power)
2. Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components.
3. Embedded systems are more tied to the hardware.

Two salient features of Embedded Programming are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

Use of C in embedded systems is driven by following advantages

- It is small and reasonably simpler to learn, understand, program and debug.
- C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
- As C combines functionality of assembly language and features of high level languages, C is treated as a 'middle-level computer language' or 'high level assembly language'
- It is fairly efficient
- It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high level languages.

Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation.

Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes). Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity. C is based on the philosophy 'programmers know what they are doing'; only the intentions are to be stated explicitly. It is easier to write good code in C & convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

Objected oriented language, C++ is not apt for developing efficient programs in resource constrained environments like embedded devices. Virtual functions & exception handling of C++ are some specific features that are not efficient in terms of space and speed in embedded systems. Sometimes C++ is used only with very few features, very much as C. And, also an object-oriented language, is different than C++. Originally designed by the U.S. DOD, it didn't gain popularity despite being accepted as an international standard twice (Ada83 and Ada95). However, Ada language has many features that would simplify embedded software development. Java is another language used for embedded systems programming. It primarily finds usage in high-end mobile phones as it offers portability across systems and is also useful for browsing applications. Java programs require Java Virtual Machine (JVM), which consume lot of resources. Hence it is not used for smaller embedded devices. Dynamic C and B# are some proprietary languages which are also being used in embedded applications.

Efficient embedded C programs must be kept small and efficient; they must be optimized for code speed and code size. Good understanding of processor architecture embedded C programming and debugging tools facilitate this.

DIFFERENCE BETWEEN C AND EMBEDDED C

Though **C and embedded C** appear different and are used in different contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications.

C is used for desktop computers, while **embedded C** is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.

Compilers for C (ANSI C) typically generate OS dependent executables. Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications. Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications.

Embedded systems often do not have a console, which is available in case of desktop applications.

So, what basically is different while programming with embedded C is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient, and satisfy real time constraints, if any. All this is done using the basic constructs, syntaxes, and function libraries of 'C'.

6.1.3 BLYNK IOT

Blynk is a platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. It's really simple to set everything up and you'll start tinkering in less than 5 mins. Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online and ready for the Internet Of Your Things.

FIG: 6.2 BLYNK APPLICATION

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things. There are three major components in the platform:

Blynk App: – It allows you to create amazing interfaces for your projects using various widgets which are provided.

Blynk Server: – It is responsible for all the communications between the smartphone and hardware. You can use the Blynk Cloud or run your private Blynk server locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

Blynk Libraries: – It enables communication, for all the popular hardware platforms, with the server and process all the incoming and outcoming commands. Now imagine, every time you press a Button in the Blynk app, the message travels to the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.

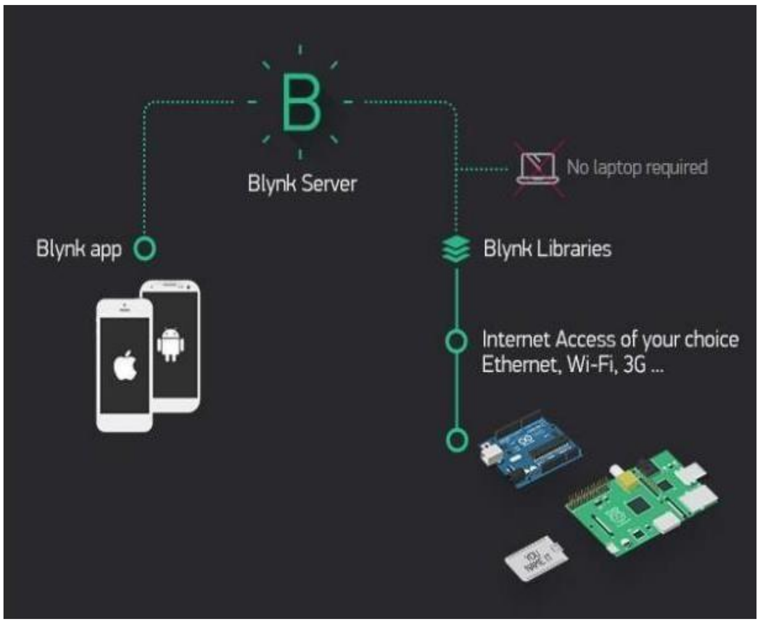


FIG 6.3 DATA PROCESSING

Characteristics of Blynk are:

- Similar API & UI for all supported hardware & devices
- Connection to the cloud can be done using Ethernet, Wi-Fi, Bluetooth, BLE and USB (Serial)
- Set of easy-to-use Widgets
- Direct pin manipulation with no code writing
- Easy to integrate new functionality using virtual pins
- History data monitoring via History Graph widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc.

CHAPTER 7
RESULT AND DISCUSSION

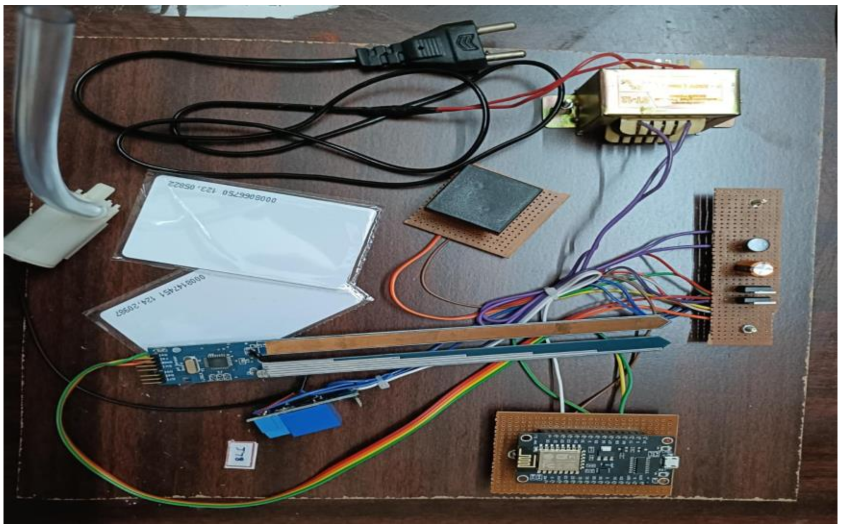


FIG 7.1 HARDWARE

The proposed system leverages IoT to enhance water management by interconnecting components for real-time monitoring, efficient allocation, and quality enforcement. An ESP32 microcontroller, which facilitates Wi-Fi connectivity, serves as the control center, directly connecting to some components while interfacing with others through intermediaries. The ESP32 enables remote monitoring and control of the entire water management system by collecting sensor data and transmitting it to a cloud platform or user interface, providing real-time access and alerts. Users can monitor water usage, quality, and consumption remotely. An RFID system tracks additional water consumption and manages payments for usage beyond a predefined threshold, assigning each user an RFID tag to monitor personal water usage and generate bills when limits are exceeded. A pH sensor monitors water quality, sending readings to the ESP32 and subsequently to the Blynk app. Additionally, a motor driver and water motor control water flow to various locations, adjusting according to demand and consumption, thereby enabling automated water supply with minimal manual intervention. The system also allows users to track water consumption and make payments for additional usage via mobile phone.

CHAPTER 8
CONCLUSION

The IoT-Based Water Management System presented in this project is a technologically advanced, efficient, and automated solution aimed at addressing water conservation, quality monitoring, and smart consumption tracking. By integrating IoT components such as the ESP32 microcontroller, pH sensor, RFID system, motor driver, and cloud-based remote monitoring, the proposed system offers a scalable and intelligent approach to managing water resources effectively.

One of the most significant advantages of this system is its real-time monitoring capability, which allows users to track water consumption, water quality, and distribution remotely. The use of an RFID-based tracking and billing mechanism ensures that water usage is monitored at an individual level, promoting responsible consumption. By automating water supply regulation based on demand, the system reduces wastage and optimizes resource allocation, minimizing manual intervention. The inclusion of a pH sensor provides continuous monitoring of water quality, ensuring that users receive clean and safe water at all times.

Unlike traditional water monitoring methods that rely heavily on manual intervention and reactive alert systems, the proposed system is proactive and automated. The ESP32 microcontroller, acting as the core processing unit, efficiently handles data collection from sensors and transmits it to a cloud platform through Wi-Fi. This enables data analytics, historical tracking, and instant notifications via a user-friendly mobile application such as Blynk. With this functionality, users can make data-driven decisions to optimize their water usage patterns.

Furthermore, the system is scalable and flexible, making it suitable not only for households but also for industrial, agricultural, and urban water management applications. The ability to remotely monitor, control, and regulate water supply enhances its usability in larger networks, making it an ideal solution for smart cities and sustainable water conservation initiatives.

In addition to its technical advantages, the system also promotes economic efficiency by implementing smart billing for excess water consumption. This ensures fair usage, encourages conservation, and supports sustainable water distribution models. The energy-efficient nature of the ESP32 microcontroller further contributes to the system’s overall cost-effectiveness and environmental sustainability.

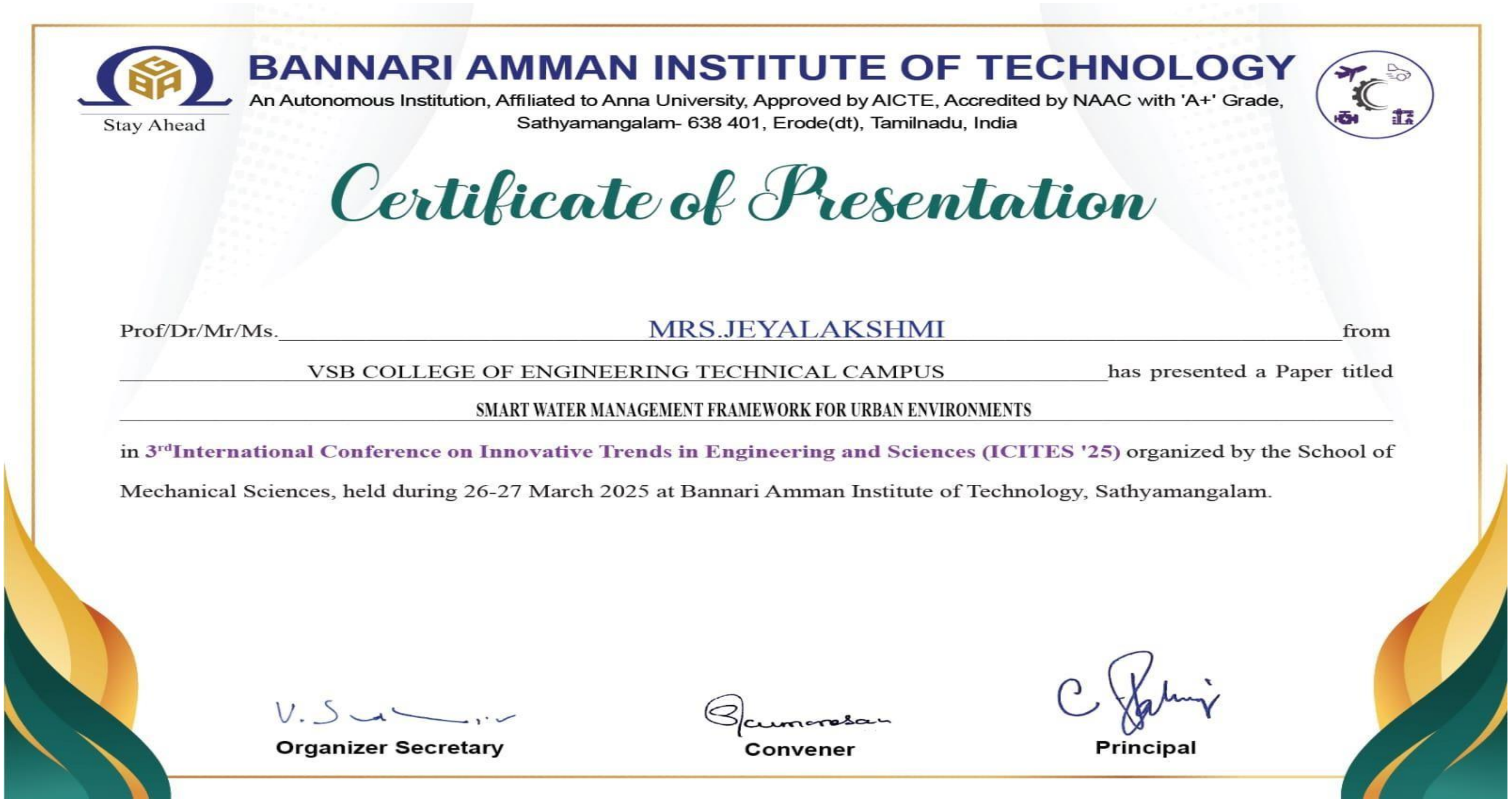
CHAPTER 9
REFERENCES

- Y. Lalle, M. Fourati, L. C. Fourati, and J. P. Barraca, ``Communication technologies for smart water grid applications: Overview, opportunities, and research directions," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107940.
- H. M. Yasin, ``IoT and ICT based smart water management, monitoring and controlling system: A review," *AJRCoS*, vol. 8, no. 2, pp. 4256, May 2021.
- M. Singh and S. Ahmed, ``IoT based smart water management systems: A systematic review,"

  Mater. Today, Proc., vol. 46, pp. 52115218, 2021.
- F. Jan, N. Min-Allah, and D. Dötegör, ``IoT based smart water quality monitoring: Recent techniques, trends and challenges for domestic applications,'' *Water,* vol. 13, no. 13, p. 1729, Jun. 2021.
- R. K. Singh, R. Berkvens, and M.Weyn, ``AgriFusion: An architecture for IoT and emerging technologies based on a precision agriculture survey,'' *IEEE Access*, vol. 9, pp. 136253136283, 2021.
- A. K. Verma, P. K. Singh, and R. K. Gupta, "IoT-based real-time water quality monitoring system in water treatment plants," *Heliyon*, vol. 10, no. 3, Mar. 2025, Art. no. e09876.
- S. Mukta, R. K. Gupta, and P. K. Singh, "IoT-based smart water quality monitoring system,"

  Internet of Things and Cyber-Physical Systems, vol. 5, pp. 123–135, 2021.
- S. Ismail, A. K. Verma, and R. K. Gupta, "IoT-based water monitoring systems: A systematic review," *Water*, vol. 14, no. 22, p. 3621, Nov. 2022.
- N. Bharath Kumar, P. Susheel Kumar, N. Pavan Kumar, and V. Chaitanya, "IoT-based smart water management system," *International Journal of Novel Research and Development*, vol. 10, no. 3, Mar. 2025.

- S. Ismail, A. K. Verma, and R. K. Gupta, "IoT-based water management systems: Survey and future research direction," *IEEE Access*, vol. 10, pp. 12345–12367, 2022.
- A. Pagano, D. Garlisi, F. Giuliano, T. Cattai, and F. Cuomo, "SWI-FEED: Smart water IoT framework for evaluation of energy and data in massive scenarios," *arXiv preprint arXiv:2404.07692*, Apr. 2024.
- M. E. Karar, M. F. Al-Rasheed, A. F. Al-Rasheed, and O. Reyad, "IoT and neural network-based water pumping control system for smart irrigation," *arXiv preprint arXiv:2005.04158*, May 2020.
- M. Bublin, H. Hirner, A.-M. Lanners, and R. Grosu, "Neuromorphic IoT architecture for efficient water management: A smart village case study," *arXiv preprint arXiv:2410.19562*, Oct. 2024.
- R. K. Singh, R. Berkvens, and M. Weyn, "AgriFusion: An architecture for IoT and emerging technologies based on a precision agriculture survey," *IEEE Access*, vol. 9, pp. 136253–136283, 2021.
- H. M. Yasin, "IoT and ICT-based smart water management, monitoring and controlling system: A review," *Asian Journal of Research in Computer Science*, vol. 8, no. 2, pp. 42–56, May 2021.



BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Stay Ahead

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade,
Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

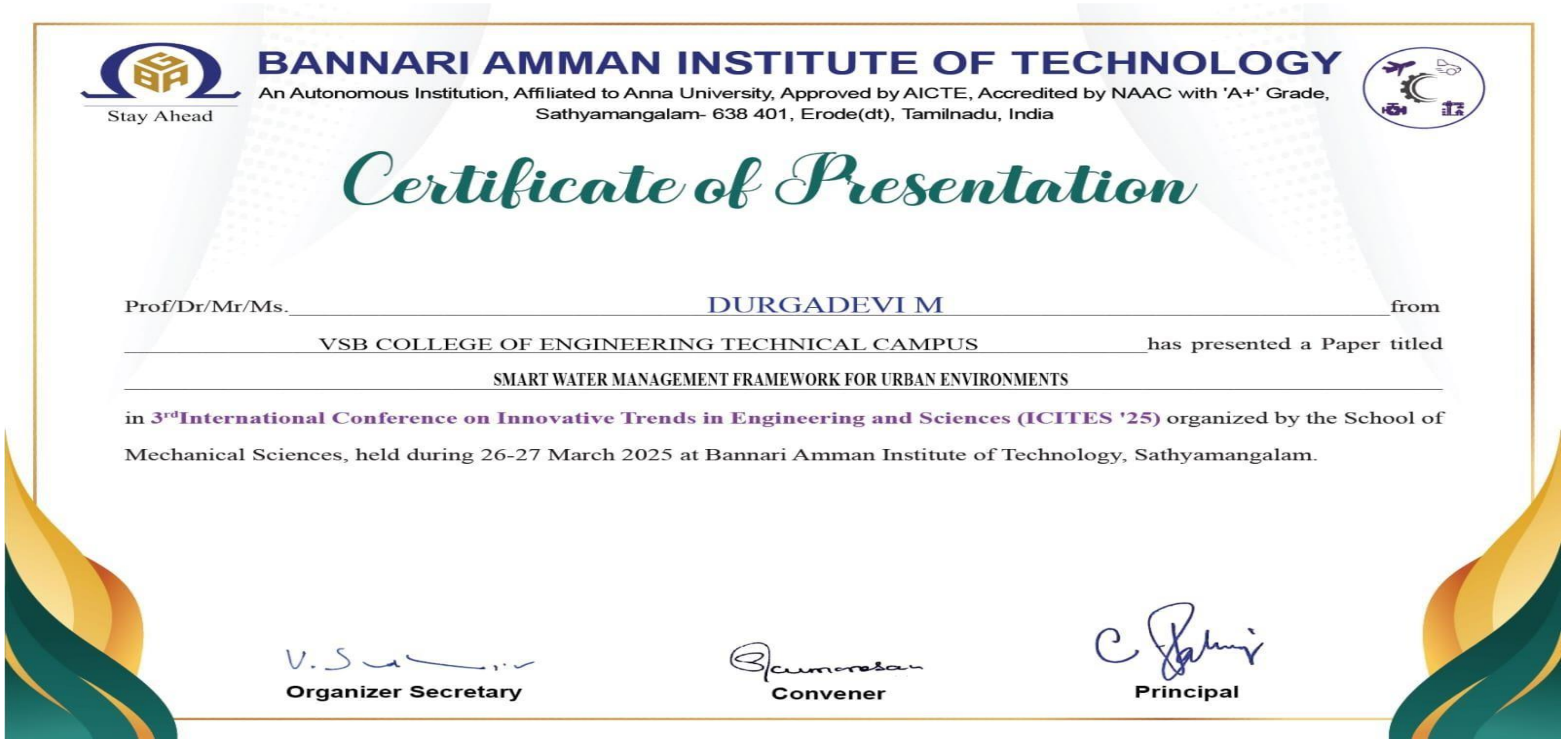
Certificate of Presentation

Prof/Dr/Mr/Ms. MRS.JEYALAKSHMI from VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS has presented a Paper titled SMART WATER MANAGEMENT FRAMEWORK FOR URBAN ENVIRONMENTS in 3rd **International Conference on Innovative Trends in Engineering and Sciences (ICITES '25)** organized by the School of Mechanical Sciences, held during 26-27 March 2025 at Bannari Amman Institute of Technology, Sathyamangalam.

[illegible]
Organizer Secretary

[illegible]
Convener

[illegible]
Principal



Stay Ahead

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade,
Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

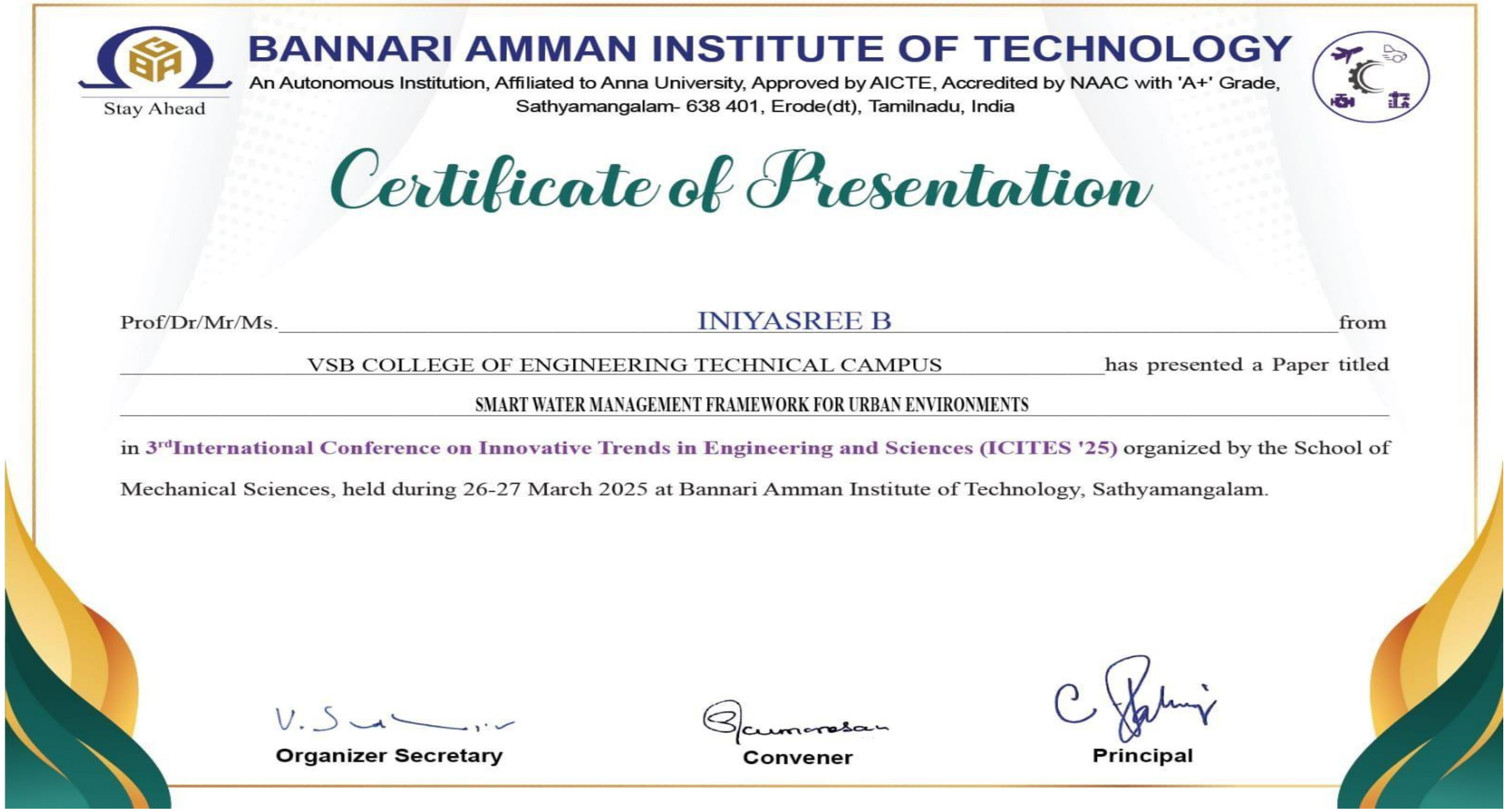
Certificate of Presentation

Prof/Dr/Mr/Ms. DURGADEVI M from VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS has presented a Paper titled **SMART WATER MANAGEMENT FRAMEWORK FOR URBAN ENVIRONMENTS** in **3rd International Conference on Innovative Trends in Engineering and Sciences (ICITES '25)** organized by the School of Mechanical Sciences, held during 26-27 March 2025 at Bannari Amman Institute of Technology, Sathyamangalam.

[illegible] **Organizer Secretary**

[illegible] **Convener**

[illegible] **Principal**



BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade,
Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

Stay Ahead

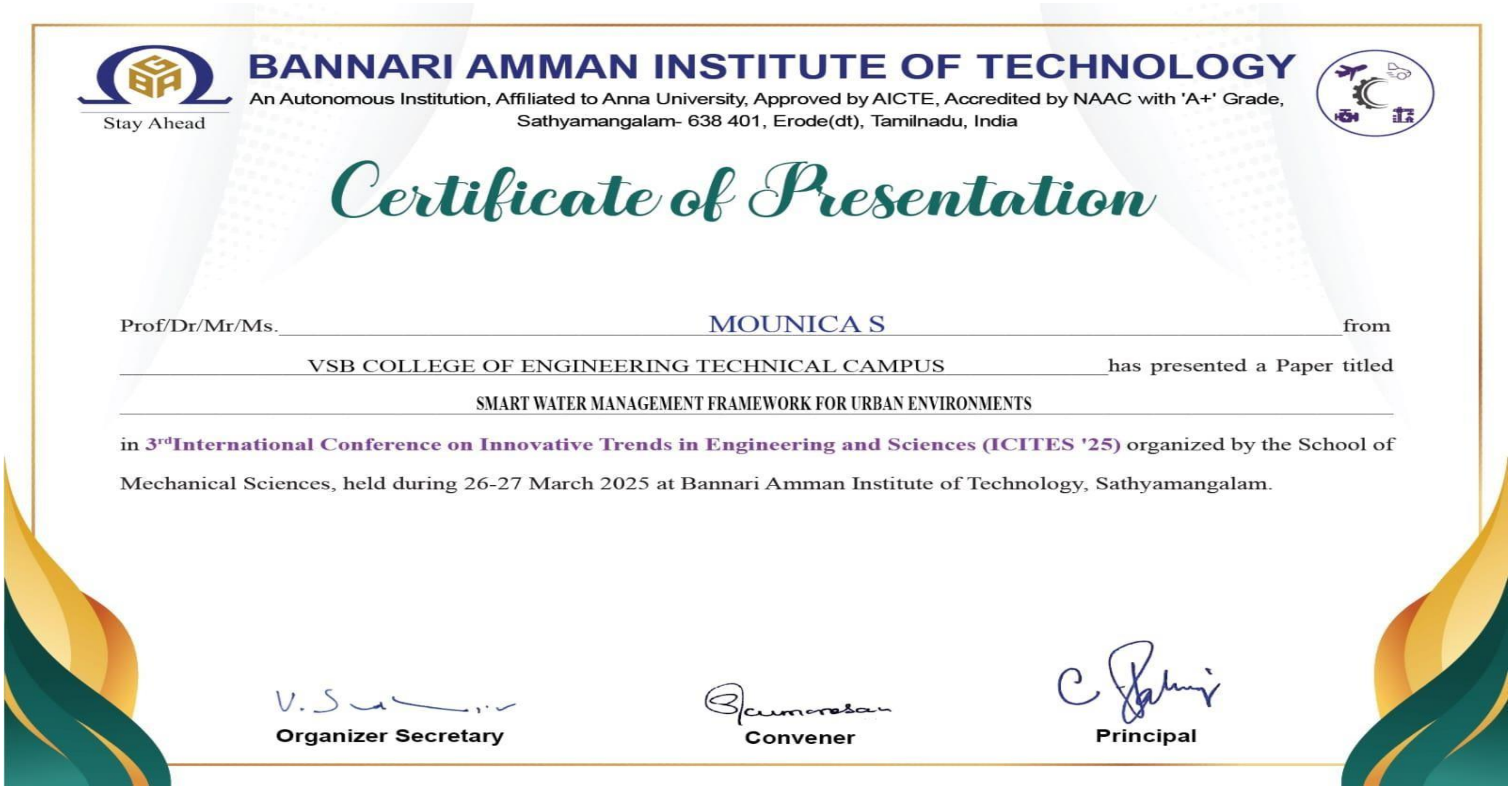
Certificate of Presentation

Prof/Dr/Mr/Ms. INIYASREE B from VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS has presented a Paper titled SMART WATER MANAGEMENT FRAMEWORK FOR URBAN ENVIRONMENTS in 3rd International Conference on Innovative Trends in Engineering and Sciences (ICITES '25) organized by the School of Mechanical Sciences, held during 26-27 March 2025 at Bannari Amman Institute of Technology, Sathyamangalam.

[illegible]
Organizer Secretary

[illegible]
Convener

[illegible]
Principal



BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Stay Ahead

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade, Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

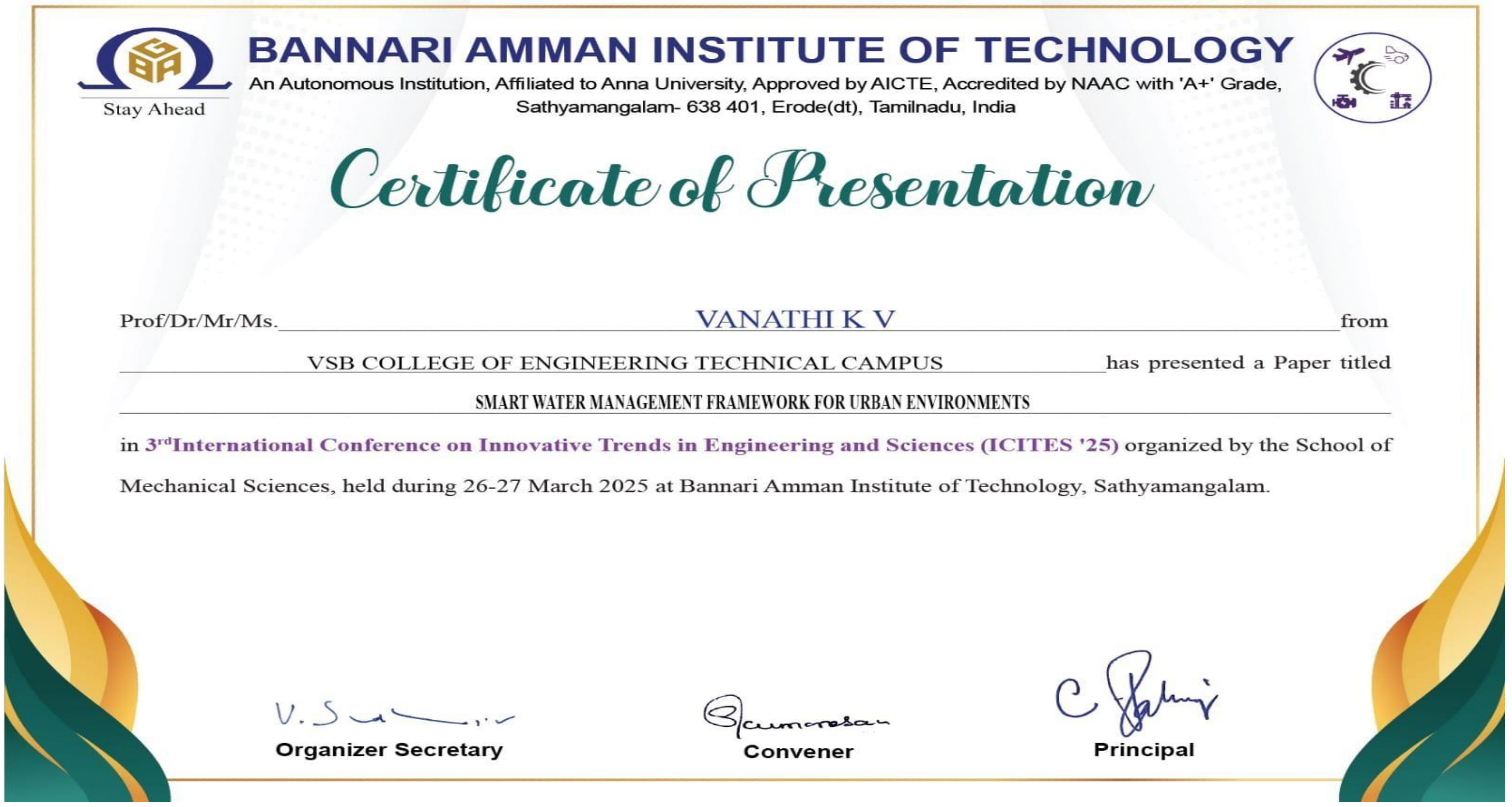
Certificate of Presentation

Prof/Dr/Mr/Ms. MOUNICA S from VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS has presented a Paper titled SMART WATER MANAGEMENT FRAMEWORK FOR URBAN ENVIRONMENTS in 3rd International Conference on Innovative Trends in Engineering and Sciences (ICITES '25) organized by the School of Mechanical Sciences, held during 26-27 March 2025 at Bannari Amman Institute of Technology, Sathyamangalam.

[illegible]
Organizer Secretary

[illegible]
Convener

[illegible]
Principal



BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Stay Ahead

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade, Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

Certificate of Presentation

Prof/Dr/Mr/Ms. VANATHI K V from VSB COLLEGE OF ENGINEERING TECHNICAL CAMPUS has presented a Paper titled SMART WATER MANAGEMENT FRAMEWORK FOR URBAN ENVIRONMENTS in 3rd International Conference on Innovative Trends in Engineering and Sciences (ICITES '25) organized by the School of Mechanical Sciences, held during 26-27 March 2025 at Bannari Amman Institute of Technology, Sathyamangalam.

[illegible]
Organizer Secretary

[illegible]
Convener

[illegible]
Principal