ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANIZATION OF ISLAMIC COOPERATION (OIC)
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

PROJECT REPORT

AUTONOMOUS WATER MANAGEMENT SYSTEM

GROUP NAME: RVG

NAME	STUDENT ID
1. MD Wahiduzzaman	200021228
2. Rahib Bin Hossain	200021241
3. K. M. Sirazul Monir	200021247
4. Feehad Kamal	200021218
5. K M Istiaque	200021220

COURSE NO.	: EEE 4518
DEPARTMENT	: EEE
SECTION	: B

TABLE OF CONTENTS

Introduction:

In the intricate fabric of our daily lives, the fundamental importance of water cannot be overstated. A life-sustaining force, water is woven into every aspect of our routine – from the basic necessity of hydration to its diverse applications in sanitation, agriculture, and industrial processes. However, in the midst of this widespread reliance on water, our collective oversight in responsibly managing this resource has given rise to a pressing concern: wastage. Nowhere is this more pronounced than in the seemingly routine act of filling water tanks, where overflow often occurs unnoticed, contributing significantly to the larger issue of water scarcity.

Compounding this challenge is the underutilization of a readily available and naturally pure source of water – rainwater. As the global community grapples with the ongoing depletion of underground water reservoirs, it becomes increasingly apparent that a paradigm shift in our approach to water conservation is imperative. It is within this context that we introduce the "Automated Water Management System," an innovative project poised to revolutionize the landscape of water conservation by seamlessly integrating technology, automation, and sensor-driven intelligence.

Our initiative extends beyond a mere technological solution; it represents a holistic reevaluation of conventional water management practices. Rooted in the ethos of sustainability and efficiency, the Automated Water Management System addresses the systemic inefficiencies that characterize current water utilization methodologies. By harnessing the power of advanced technology, including Arduino, sensors, and sophisticated algorithms, our system aspires to usher in a transformative era where the preservation and optimal utilization of water resources become second nature.

This project does not exist in isolation; it is a response to the collective imperative of sustainable living. At its heart, the Automated Water Management System seeks to instill a conscientious approach to water use, mitigating wastage during routine tasks and harnessing the purity of rainwater. As we explore the intricacies of this groundbreaking initiative, we uncover a network of interconnected solutions, each designed to contribute to the larger goal of fostering a harmonious relationship between humanity and its most vital resource – water. Through this extended introduction, we invite you to join us on a journey, where innovation meets necessity in the pursuit of a more water-conscious future.

Problem Overview:

The pressing issue of water scarcity and inefficient water management practices forms the backdrop against which the "Automated Water Management System" emerges as a crucial solution. Across the globe, the burgeoning demand for water, exacerbated by a growing population and industrialization, underscores the urgent need for a paradigm shift in how we approach and utilize this finite resource.

One of the significant challenges contributing to water scarcity is the inadvertent wastage during routine activities, particularly evident in the process of filling water tanks. Overflow, often unnoticed until its consequences unfold, represents a substantial loss and underscores the critical need for vigilant water management strategies. The lack of awareness and effective control mechanisms perpetuates this problem, leading to unnecessary depletion of valuable water resources.

Additionally, the oversight in harnessing rainwater, a pristine and naturally replenishing source, further compounds the issue. As underground water levels continue to diminish, there exists an untapped potential in rainwater harvesting that remains largely unexplored. The "Automated Water Management System" aims to rectify this oversight by integrating innovative technologies that optimize rainwater collection, mitigating the impact of declining water tables.

Furthermore, the deterioration of water quality poses a silent threat. Neglecting the cleanliness of water storage tanks can lead to contamination, affecting the health and well-being of communities. Current systems often lack proactive measures to address this concern, necessitating a solution that not only monitors water quality but also alerts users to take corrective actions.

In the face of these challenges, the Automated Water Management System emerges as a comprehensive solution, poised to redefine the way we interact with and preserve water resources. By addressing wastage, optimizing rainwater harvesting, and ensuring water quality, this project seeks to contribute significantly to the ongoing global effort to create sustainable and resilient water management systems.

Objectives:

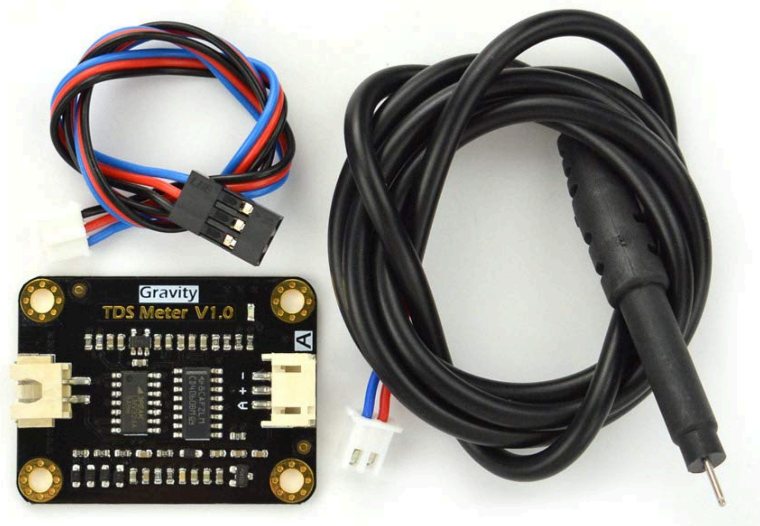
1. **Wastage Reduction:** The primary objective of our project is to significantly reduce water wastage during the pumping process. Through real-time monitoring and control mechanisms, the system aims to prevent overflow incidents, ensuring that every drop is utilized efficiently.
2. **Quality Monitoring**: The inclusion of a Total Dissolved Solids (TDS) meter adds a layer of sophistication to our system. Monitoring the water quality in the tanks, the system alerts users when tank cleaning is necessary, maintaining the purity of stored water.
3. **Smart Pump Control:** Utilizing Arduino and relay-controlled pumps, our system offers intelligent pump control. It adapts to real-time conditions, ensuring that water is pumped into the tanks according to need, with considerations for weather forecasts, potential rain, and the current water level.
4. **Weather-Responsive Water Filling**: The integration of an ESP32 development board allows the system to fetch and analyze weather data. This information is used to optimize the water filling process, ensuring that tanks are filled proportionately based on the likelihood of upcoming rainfall, thereby maximizing the use of rainwater.
5. **Adaptive Water Level Management**: Employing water level meters and Arduino technology, the system adapts to the rate of water consumption. It intelligently determines the required water level adjustments, taking into account upcoming weather conditions to optimize water usage.

In essence, the Automated Water Management System aspires to be a beacon of sustainable water utilization, offering an innovative solution to the pressing challenges of water scarcity and wastage. Through the seamless integration of technology, our project aims to contribute to a future where water is managed judiciously, promoting a harmonious coexistence with our environment.

Components :

Gravity: Analog TDS Sensor :

The Gravity Analog TDS Sensor is a critical component for evaluating water quality based on Total Dissolved Solids (TDS) levels. Employing electrical conductivity as its measurement principle, the sensor provides accurate, real-time data on the concentration of dissolved ions in a liquid.



This sensor features analog output, delivering TDS measurements in a format compatible with various microcontrollers and data processing systems. Its utilization of electrical conductivity is grounded in the fact that the conductivity of a solution is directly proportional to the concentration of dissolved ions, offering a reliable method for assessing water quality.

The sensor's integration is facilitated by its compatibility with the Gravity interface, simplifying connections and enhancing accessibility for users. The analog data output allows seamless interfacing with microcontrollers and data loggers, making it adaptable to a broad spectrum of electronic projects.

Designed for versatility, the Gravity Analog TDS Sensor can function in diverse environmental conditions, making it suitable for applications ranging from basic water quality testing to more complex environmental monitoring systems. Its resilience and reliability, even in challenging environments, contribute to its appeal for a wide array of technical projects.

In summary, the Gravity Analog TDS Sensor stands out for its technical prowess in measuring TDS levels in water. With its electrical conductivity-based measurement approach, analog output, and adaptability, the sensor is a valuable tool for engineers, researchers, and hobbyists involved in water quality assessment and monitoring projects.

HC-SR04 :

The HC-SR04 ultrasonic sensor is a widely employed component in electronics, particularly in projects requiring distance measurement or obstacle avoidance. Operating on sonar principles, this sensor utilizes ultrasonic waves to gauge the distance between the sensor and an object in its vicinity.

The HC-SR04 employs a straightforward methodology. It consists of an ultrasonic transmitter and receiver pair. The transmitter emits short ultrasonic pulses, which bounce off nearby objects and are then received by the sensor's receiver. By calculating the time it takes for the signal to travel to the object and back, the sensor can determine the distance with precision.

One of the key features that contribute to the popularity of the HC-SR04 is its simplicity of use. It requires only four pins for connection: VCC (power supply), Trig (trigger), Echo (echo), and GND (ground). Integration with microcontrollers or single-board computers, such as Arduino or Raspberry Pi, is straightforward, making it accessible for both beginners and experienced enthusiasts.

The HC-SR04 offers a considerable range for distance measurement, often reaching several meters. It finds application in various fields, including robotics, automation, and IoT projects. Its reliability and low cost make it a preferred choice for tasks like object detection, collision avoidance, or even creating basic mapping systems.

In conclusion, the HC-SR04 ultrasonic sensor exemplifies the efficiency of sonar technology in providing accurate and real-time distance measurements. Its simplicity, affordability, and versatility make it a staple in the toolkit of electronics hobbyists and professionals alike.

GSM Module

The GSM module, specifically the SIM800, stands as a fundamental component in the realm of telecommunications, providing a reliable means for devices to communicate over the Global System for Mobile Communications (GSM) network. Developed by SIMCom Wireless Solutions, the SIM800 module is renowned for its versatility and compatibility, making it a preferred choice for a myriad of applications, including Internet of Things (IoT) devices, telemetry systems, and various remote communication solutions.

At its core, the SIM800 module integrates a GSM/GPRS modem, allowing devices to transmit data, make calls, and send text messages over the cellular network. The module supports quad-band GSM/GPRS frequencies, ensuring global compatibility across diverse regions.

Operating in the 850MHz, 900MHz, 1800MHz, and 1900MHz frequency bands, the SIM800 module facilitates seamless communication across different mobile networks worldwide.

One of the distinctive features of the SIM800 is its compact form factor, making it suitable for applications with space constraints. Additionally, the module supports a wide range of voltage levels, enhancing its adaptability to various power supply configurations. With a low power consumption profile, the SIM800 is energy-efficient, a critical consideration for battery-powered devices and remote applications.

The SIM800 module communicates with external devices through standard communication interfaces such as UART, making it compatible with microcontrollers and embedded systems. It employs AT commands for configuration and control, simplifying the integration process into existing projects. This standardized command set enables developers to initiate calls, send SMS messages, and manage network connections seamlessly.

In terms of connectivity, the SIM800 module supports GPRS data transmission, enabling devices to establish an internet connection and exchange data with remote servers. This feature is particularly valuable for IoT applications that rely on cellular networks for data exchange. Furthermore, the module incorporates advanced security features, including encryption algorithms, to ensure the confidentiality of transmitted data.

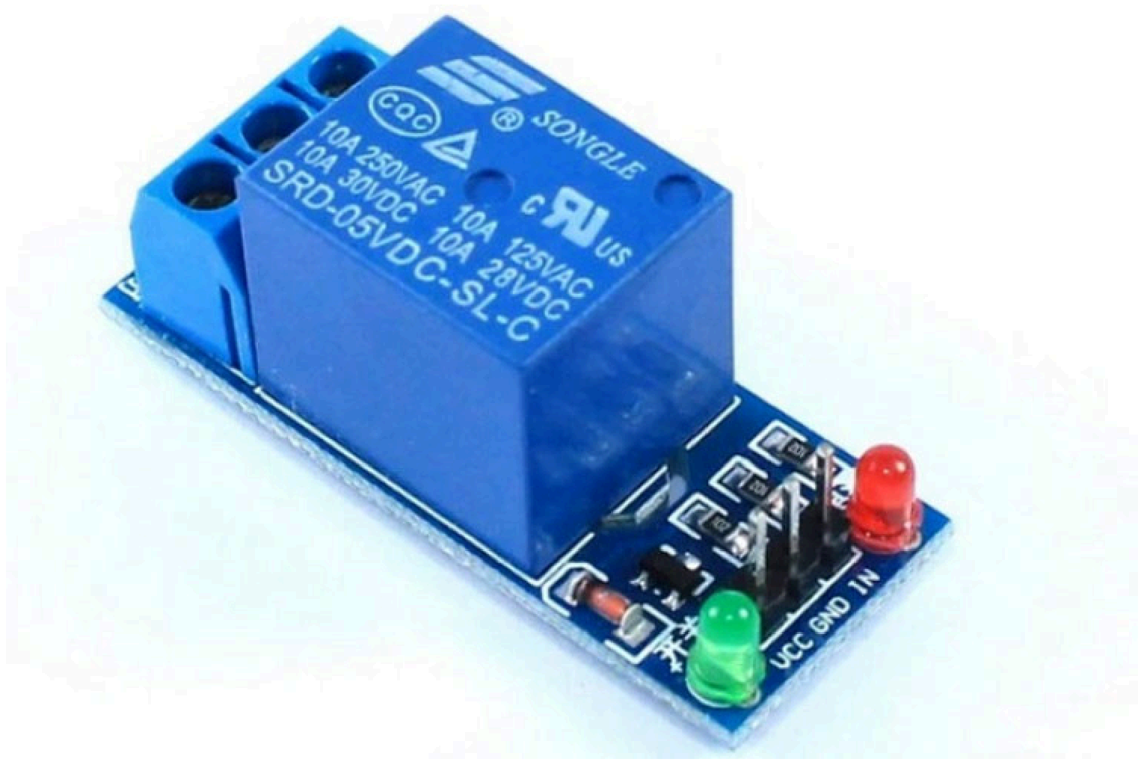
To facilitate location-based services, the SIM800 module integrates a GPS receiver, allowing devices to obtain accurate geographical coordinates. This functionality enhances the module's applicability in applications requiring real-time location tracking.

In conclusion, the SIM800 GSM module serves as a cornerstone in the evolution of wireless communication technologies, providing a robust and versatile solution for a broad spectrum of applications. Its compact design, global compatibility, low power consumption, and support for various communication interfaces position it as a reliable choice for developers and engineers seeking to integrate cellular connectivity into their projects. Whether deployed in remote monitoring systems, IoT devices, or other applications, the SIM800 module continues to play a pivotal role in shaping the landscape of modern telecommunications.

5V Relay:

The 5V relay is a fundamental component in electronics, serving as a switch to control high-power devices with low-voltage microcontrollers or other control systems. This electromechanical device plays a crucial role in automating processes, enabling the interfacing of low-power electronic circuits with larger, high-power loads.

At its core, the 5V relay consists of an electromagnetic coil and a set of contacts. When a 5V signal is applied to the coil, it generates a magnetic field, causing the contacts to close or open, depending on the relay type. This mechanism allows the relay to control circuits with different voltage levels, providing isolation between the control and load circuits.



One of the key advantages of the 5V relay is its compatibility with the commonly used 5V logic levels found in microcontrollers like Arduino and Raspberry Pi. This makes it a popular choice for hobbyists and engineers in various applications, including home automation, robotics, and industrial control systems.

The 5V relay finds application in scenarios where low-power electronic components need to control devices such as lights, motors, or appliances that operate at higher voltages. By acting as a bridge between the low-voltage control circuit and the high-voltage load, the relay ensures safety and reliability in diverse electronic projects.

In summary, the 5V relay stands as a crucial interface component in electronics, allowing for the controlled switching of high-power devices using low-voltage signals. Its simplicity, compatibility, and reliability make it an essential tool for those engaged in electronic prototyping, automation, and control system design.

12V Pump:

The 12V pump is an electromechanical device designed for efficient fluid movement in diverse applications. Operating on a 12-volt direct current (DC) power supply, this pump is characterized by its technical adaptability and reliability. Utilizing principles of fluid dynamics, it incorporates mechanisms such as centrifugal or diaphragm pumping to facilitate fluid transfer.

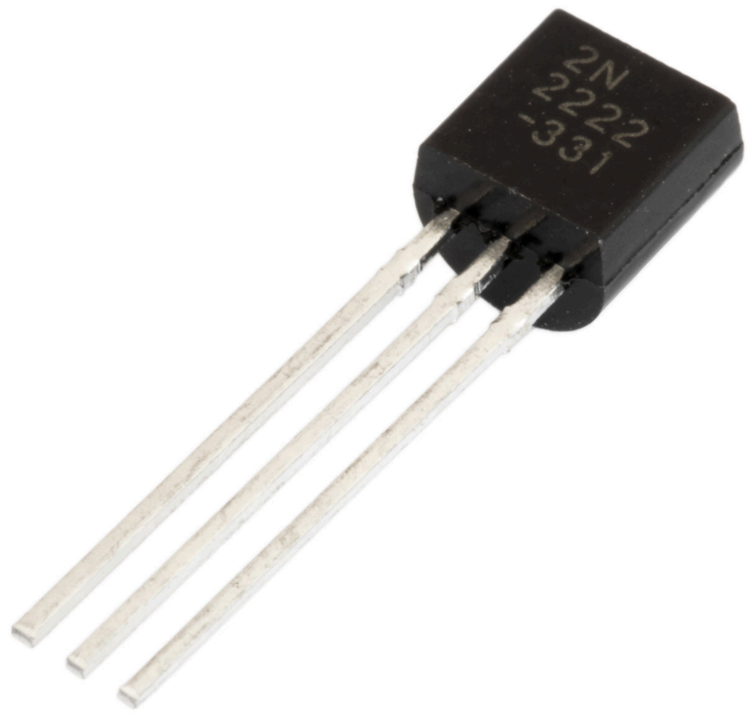
Centrifugal pumps within the 12V category are adept at handling moderate-pressure tasks that involve moving large volumes of fluid, making them suitable for applications like water circulation or low-pressure irrigation systems. On the other hand, diaphragm pumps, also available in the 12V range, excel in precision and controlled fluid delivery, finding use in applications where accuracy is crucial.

The 12V pump's technical specifications make it compatible with standard 12V DC power sources, providing flexibility in power options, including batteries or solar panels. This feature enhances its adaptability in both stationary and mobile setups. The compact design of these pumps makes them particularly advantageous in scenarios where space is limited or where portability is a priority.

Technical integration with electronic systems is seamless, making the 12V pump suitable for projects involving microcontrollers like Arduino or Raspberry Pi. This capability allows for intelligent control and monitoring of fluid dynamics, enabling automation and customization in fluid management systems.

In summary, the 12V pump, with its technical versatility, stands as a fundamental component in fluid dynamics. Its adaptability to various power sources, compact design, and integration potential with electronic systems make it a go-to solution for engineers and enthusiasts engaged in projects ranging from precision fluid control to mobile fluid transfer applications.

2N2222 NPN BJT as a Relay Driver



The 2N2222 is a widely used NPN (Negative-Positive-Negative) Bipolar Junction Transistor (BJT) with versatile applications, including serving as an efficient relay driver. In this configuration, the 2N2222 transistor acts as a switch, controlling the current flow through the relay coil.

To employ the 2N2222 as a relay driver, it is typically connected in a common-emitter configuration. The base of the transistor is connected to the control signal (coming from a microcontroller or another logic device), the collector is linked to the relay coil, and the emitter is grounded. A protective diode, often a flyback diode (such as a 1N4148), is essential across the relay coil to prevent voltage spikes during the coil deactivation.

When a logical high signal is applied to the base of the 2N2222, the transistor becomes forward-biased, allowing current to flow from the collector to the emitter. This energizes the relay coil, closing the relay switch contacts. Conversely, when a logical low signal is applied, the transistor becomes reverse-biased, de-energizing the relay coil and opening the relay contacts.

The 2N2222's ability to handle moderate current levels and its ease of use make it a suitable choice for driving relays in various electronic applications. This setup is commonly employed in projects such as home automation, industrial control systems, or any application requiring the precise control of higher-power devices using low-power control signals.

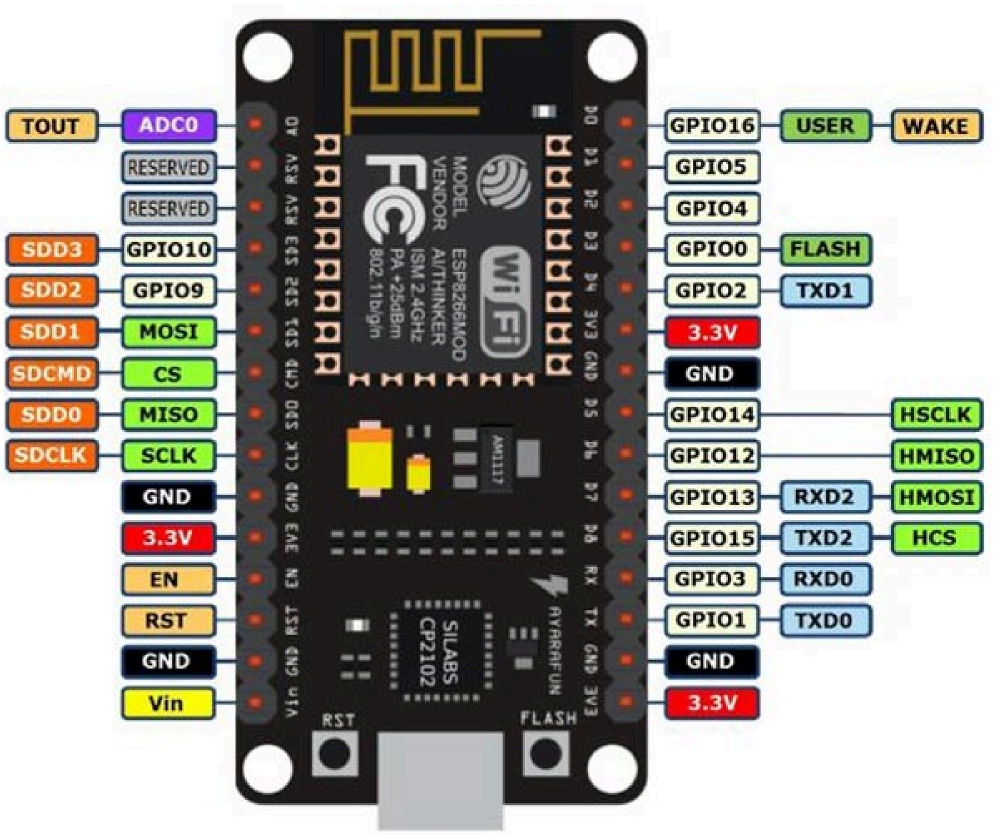
In conclusion, utilizing the 2N2222 NPN BJT as a relay driver presents a straightforward and effective solution for interfacing low-power control signals with higher-power devices. Its simplicity, reliability, and compatibility with common relay configurations make it a valuable component in electronics and microcontroller-based projects.

Esp8266 development board

The ESP8266 is a highly popular and versatile Wi-Fi module that has revolutionized the landscape of Internet of Things (IoT) and embedded systems. Developed by Espressif Systems, the ESP8266 integrates a powerful 32-bit Tensilica L106 microcontroller with a built-in Wi-Fi stack, offering a cost-effective and compact solution for wireless connectivity.

At its heart, the ESP8266 module is equipped with a 32-bit RISC CPU operating at clock speeds of up to 80 MHz. This enables rapid data processing and efficient execution of embedded applications. The module comes in various versions, with the ESP-12 and ESP-12E being among the most widely used, featuring GPIO (General Purpose Input/Output) pins for versatile interfacing with other devices.

One of the standout features of the ESP8266 is its built-in Wi-Fi connectivity, supporting IEEE 802.11 b/g/n standards. This capability allows the module to establish connections to Wi-Fi networks, enabling seamless communication with the internet and local network resources. The module can operate in various modes, including Station mode for connecting to existing Wi-Fi networks and Access Point mode for creating its own Wi-Fi network.



Programming the ESP8266 is facilitated through the use of the Arduino IDE, making it accessible to a broad community of developers. The module supports the Arduino programming language and libraries, simplifying the development process for those familiar with the Arduino ecosystem. Additionally, the ESP8266 can be programmed using the Espressif IoT Development Framework (ESP-IDF), providing more advanced features and customization options.

In terms of memory, the ESP8266 typically comes with varying amounts of flash memory, often ranging from 512KB to 4MB. This flash memory is used for storing the firmware, program code, and other data. The module also features RAM for runtime data storage and execution.

The ESP8266 is renowned for its low power consumption, a crucial aspect for battery-powered and energy-efficient IoT applications. Its ability to operate in low-power modes and quickly transition between active and sleep states contributes to its suitability for a wide range of projects where power efficiency is a priority.

To facilitate communication with other devices, the ESP8266 module supports multiple communication interfaces, including UART, SPI, and I2C. This makes it compatible with a variety of sensors, displays, and other peripherals commonly used in embedded systems.

Security is addressed through features such as WPA/WPA2 encryption for Wi-Fi connections, ensuring the confidentiality and integrity of data transmitted over the network. Developers can implement additional security measures within their applications to meet specific project requirements.

In summary, the ESP8266 has emerged as a pivotal component in the development of IoT and embedded systems, offering a compelling blend of processing power, Wi-Fi connectivity, and ease of programming. Its widespread adoption, coupled with a vibrant community and extensive documentation, reinforces its position as a go-to choice for developers seeking an efficient and cost-effective solution for wireless communication in their projects.

Methodology:

The methodology devised for the "Automated Water Management System" encompasses a multifaceted approach, strategically addressing various aspects to ensure a comprehensive and efficient water management solution. Each feature is meticulously designed to contribute to the overall objective of minimizing wastage, optimizing resource utilization, and promoting sustainable water practices.

1. **GSM Module Setup:**

- **Hardware Connection:** Connect the GSM module (SIM800) to the Arduino using SoftwareSerial, with the RX pin connected to D5 and TX pin to D6.
- **Initialization:** Initialize the GSM module for communication. This involves checking if the module is ready using the "AT" command.

2. **WiFi Connection:**

- **WiFi Configuration:** Provide the system with the SSID and password for the WiFi network.

- **Connection Establishment:** Utilize the ESP8266WiFi library to establish a connection to the WiFi network.
- **Status Monitoring:** Continuously monitor the WiFi connection status and display the assigned IP address.

3. **OpenWeatherMap Integration:**

- **API Key:** Utilize a unique OpenWeatherMap API key for authentication.
- **Data Retrieval:** Construct an HTTP GET request to OpenWeatherMap, including the specified city and country code.
- **Response Handling:** Parse the received JSON response using the Arduino_JSON library to extract temperature, pressure, humidity, wind speed, and cloud cover.

4. **Ultrasonic Sensor Setup:**

- **Sensor Connection:** Connect Ultrasonic sensors (HC-SR04) to the Arduino, with the Trig pin on D7 and Echo pin on D8.
- **Functionality:** Employ Ultrasonic sensors to measure water level by calculating the distance based on the duration of ultrasonic pulses.

5. **Relay Control Setup:**

- **Relay Connection:** Connect a relay to the Arduino, with the control pin on D4.
- **Water Pump Control Logic:** Implement logic to control the water pump relay based on the measured water level.
- **Conditional Control:** If the water level is below 20%, turn on the pump; if above 90%, turn off the pump.

6. **TDS Sensor Reading:**

- **Analog Sensor Connection:** Connect a Total Dissolved Solids (TDS) analog sensor to pin A0.
- **Reading Acquisition:** Read the TDS value from the sensor using analogRead().

7. **LCD Display Setup:**

- **I2C LCD Configuration:** Set up an I2C LCD (LiquidCrystal_I2C) with a specific address (0x27) and dimensions (16x2).
- **Initialization:** Initialize the LCD and activate the backlight.

8. **Data Retrieval and Processing:**

- **HTTP GET Request:** Use a function to send an HTTP GET request to OpenWeatherMap.
- **Response Handling:** Check the HTTP response code and store the received JSON response.

- **JSON Parsing:** Utilize the Arduino_JSON library to parse the JSON response for further processing.

9. **Ultrasonic Sensor Reading:**

- **Pulse Generation:** Generate ultrasonic pulses by setting the Trig pin to HIGH and LOW.
- **Pulse Duration Measurement:** Measure the duration of the received pulse on the Echo pin to calculate the distance.

10. **Relay Control Logic:**

- **Water Level Mapping:** Map the distance obtained from the Ultrasonic sensor to a percentage representing the water level.
- **LCD Display:** Continuously update the LCD display with TDS value, water level, and relevant information.
- **Conditional Control:** Based on the water level and cloud cover information, control the water pump relay.

11. **SMS Notification (GSM):**

- **Threshold Configuration:** Set a threshold TDS value.
- **Notification Logic:** If the TDS value exceeds the threshold and a notification hasn't been sent, send an SMS using the GSM module.
- **Prevent Duplicate Notifications:** Set a flag to prevent repeated SMS notifications until the TDS value falls below the threshold.

Working Procedure

The "Automated Water Management System" operates seamlessly through a series of interconnected steps, harnessing advanced technology to optimize water usage, prevent wastage, and enhance overall efficiency. The following detailed working procedure provides an in-depth understanding of the system's functionality:

1. Initialization and Setup:

The system begins by initializing all hardware components, including the GSM module, WiFi module, sensors, relay, and the I2C LCD display. This involves configuring pins, setting up communication protocols, and activating necessary functionalities. The initialization phase ensures that all components are ready for operation.

2. WiFi Connection and Network Monitoring:

The system establishes a connection to the specified WiFi network using the ESP8266WiFi library. Continuous monitoring of the WiFi connection status is performed, allowing the system to adapt dynamically to changes in network conditions. The assigned IP address is displayed, providing visibility into the network connectivity status.

3. Real-time Weather Data Retrieval:

At regular intervals, the system initiates an HTTP GET request to the OpenWeatherMap API to retrieve real-time weather data for the specified city and country code. This data includes crucial information such as temperature, pressure, humidity, wind speed, and cloud cover, which is vital for making informed decisions regarding water management.

4. Ultrasonic Sensor Operation for Water Level Monitoring:

The Ultrasonic sensors (HC-SR04) are actively employed to measure the water level. Ultrasonic pulses are generated by the Trig pin, and the system measures the duration of the received pulse on the Echo pin. This duration is then used to calculate the distance, providing an accurate representation of the current water level.

5. Dynamic Relay Control Logic:

The relay, responsible for controlling the water pump, is dynamically governed based on the water level measurements. The system employs conditional logic to decide when to activate or deactivate the pump relay. If the water level falls below 20%, the pump is turned on to maintain an adequate water supply. Conversely, if the water level surpasses 90%, the pump is deactivated to prevent overflow.

6. Total Dissolved Solids (TDS) Sensor Reading:

The system reads the Total Dissolved Solids (TDS) value from the analog sensor connected to pin A0. This reading provides insights into the quality of the water, helping to determine if the water needs to be changed or treated.

7. SMS Notification System (GSM Module):

A threshold for TDS values is set, and if the actual TDS value exceeds this threshold, an SMS notification is sent using the GSM module. This notification serves as an alert to the user, indicating the need to change the water. To prevent repetitive notifications, a flag is employed to track whether a notification has already been sent.

8. LCD Display for Real-time Information:

The I2C LCD display is continuously updated with real-time information, including the TDS value, water level percentage, and relevant weather data. This provides a convenient and accessible interface for users to monitor critical parameters at a glance.

9. Continuous Monitoring and Iteration:

The entire system operates in a loop, continuously monitoring and updating information. This iterative process ensures that the water management system responds dynamically to changing conditions, providing real-time insights and control. The combination of various sensors and communication modules creates a robust and adaptable system for efficient water management.

Connection Mechanism :

1. GSM Module (SIM800):

- **Connection to Arduino:** Connect the RX pin of the GSM module to pin D5 (rxPin) and the TX pin to pin D6 (txPin) on the Arduino.
- **Initialization:** The GSM module is initialized using the SoftwareSerial library, and serial communication is established at a baud rate of 9600.

2. WiFi Module (ESP8266):

- **Connection to Arduino:** No explicit pin connections are mentioned, but the ESP8266 module is utilized for WiFi connectivity.
- **WiFi Configuration:** The system connects to a WiFi network with the provided SSID and password.

3. OpenWeatherMap API Integration:

- **API Key:** The OpenWeatherMap API key is provided as "openWeatherMapApiKey."
- **HTTP GET Request:** The system constructs an HTTP GET request to OpenWeatherMap using the city, country code, and API key.

4. Ultrasonic Sensors (HC-SR04):

- **Trig and Echo Pins:** Ultrasonic sensors are connected to D7 (trigPin) and D8 (echoPin) on the Arduino.
- **Functionality:** Ultrasonic pulses are generated on the Trig pin, and the Echo pin measures the duration of the received pulses, allowing the system to calculate the distance.

5. Relay Control:

- **Relay Pin:** The relay control pin is connected to pin D4 on the Arduino.
- **Water Pump Control Logic:** Based on the water level information, the relay controls the water pump. It is turned on when the water level is below 20% and turned off when it exceeds 90%.

6. Total Dissolved Solids (TDS) Sensor:

- **Sensor Pin:** The TDS sensor is connected to pin A0 on the Arduino.
- **Reading Acquisition:** The system reads the TDS value using analogRead() on pin A0.

7. I2C LCD Display (LiquidCrystal_I2C):

- **Connection:** The I2C LCD display is connected to the Arduino using the Wire library.
- **Initialization:** The LCD is initialized with the I2C address 0x27 and dimensions 16x2.
- **Display Content:** Information such as TDS value, water level, and weather data is displayed on the LCD.

8. Power Connections:

- **GSM Module:** It typically requires a power supply (VCC and GND).
- **ESP8266:** Requires a power supply through the Arduino board.
- **Ultrasonic Sensors:** Require VCC, GND, Trig, and Echo connections.
- **Relay:** Requires power connections for VCC and GND.
- **TDS Sensor:** Requires power supply through VCC and GND.
- **LCD Display:** Powered through VCC and GND.

9. Data Flow and Processing:

- **WiFi Connection Status:** Monitored using the **WiFi.status()** function.
- **HTTP GET Request:** Sent to OpenWeatherMap to fetch weather data.
- **JSON Parsing:** The response is parsed using the Arduino_JSON library, extracting temperature, pressure, humidity, wind speed, and cloud cover.
- **Ultrasonic Sensor Readings:** Used to calculate the water level.

- **Relay Control Logic:** Determines when to turn on or off the water pump based on water level and cloud cover.
- **TDS Sensor Reading:** Analog reading from the TDS sensor provides information about water quality.

10. SMS Notification (GSM Module):

- **SMS Configuration:** Utilizes the GSM module to send SMS notifications.
- **Threshold Setting:** TDS values exceeding a set limit trigger an SMS alert.
- **Preventing Duplicate Notifications:** Flags are used to prevent repeated SMS alerts.

11. Serial Communication:

- **Serial Communication with GSM Module:** Achieved through the SoftwareSerial library to send and receive AT commands.

12. Timing and Delays:

- **Timer Delay:** The system uses **millis()** to implement a timer delay of 10 seconds (**timerDelay**), controlling the frequency of weather data updates.

This detailed hardware and data flow explanation provides insights into how each component is connected and how information is processed within the automated water management system using an ESP8266-based Arduino.

CODE :

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <Arduino_JSON.h>

// GSM Module RX pin to Arduino 2
// GSM Module TX pin to Arduino 3
#define rxPin D5
#define txPin D0

const char* ssid = "ESP";
const char* password = "passpass1";
String openWeatherMapApiKey = "a95f7cec1f0a9fbf07dd3696b2222bb9";
SoftwareSerial sim800(txPin, rxPin);
```

```
String city = "Tongi";
String countryCode = "BD";
unsigned long lastTime = 0;
unsigned long timerDelay = 10000;  // Timer delay set to 10 seconds
String jsonBuffer;

// Median filtering algorithm
// TDS sensor pin
#define tdsSensorPin A0

// Relay control pin
#define relayPin D7

bool already_sent = false;
int pos = HIGH;
int counter = 1;

// Ultrasonic sensor pins
int trigPin = D4; // TRIG pin
int echoPin = D8; // ECHO pin

float duration_us, distance_mm;
int cloudint;
int humm;
int tmp;
int wndspd;
int chap;
int feelslike;
int visib;

// I2C address for the LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  Serial.begin(9600); // Initialize serial communication
  sim800.begin(9600); // Initialize GSM module
  Serial.println("GSM module initialized");

  // Ultrasonic sensor setup
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Relay setup
  pinMode(relayPin, OUTPUT);
  pinMode(D6, INPUT);

  // LCD setup
  lcd.begin();
  lcd.backlight();

  delay(1000);

  sim800.println("AT"); // Check if GSM module is ready
  updateSerial();

  // Connect to WiFi
  WiFi.begin(ssid, password);
  Serial.println("Connecting");

  // Wait until the connection is established
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Print the IP address of the board
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

  // Print the timer interval
  Serial.println("Timer set to 10 seconds (timerDelay variable), it will
take 10 seconds before publishing the first reading.");
}

void loop() {
  // Send an HTTP GET request at the specified timer interval
  if ((millis() - lastTime) > timerDelay) {
    // Check WiFi connection status
    if(WiFi.status()== WL_CONNECTED){
      // Construct the URL for the OpenWeatherMap API
      String serverPath =
"http://api.openweathermap.org/data/2.5/weather?q=" + city + "," +
countryCode + "&APPID=" + openWeatherMapApiKey;

      // Send the HTTP GET request and store the response
      jsonBuffer = httpGETRequest(serverPath.c_str());

      // Print the response
      Serial.println(jsonBuffer);

      // Parse the response as a JSON object
      JSONVar myObject = JSON.parse(jsonBuffer);

      // Check if the parsing was successful
      if (JSON.typeof(myObject) == "undefined") {
        // If not, print an error message and return
        Serial.println("Parsing input failed!");
        return;
      }

      // Print the JSON object and its properties
      Serial.print("JSON object = ");
      Serial.println(myObject);
      Serial.print("Temperature: ");
      tmp = int(myObject["main"]["temp"]);
      Serial.println(tmp);

      // ... (other property prints)

      cloudint = int(myObject["clouds"]["all"]);
    }
    else {
      // If the WiFi connection is lost, print a message
      Serial.println("WiFi Disconnected");
    }

    // Update the last time
    lastTime = millis();
  }
```

```
// TDS sensor reading
int tdsValue = analogRead(tdsSensorPin);
Serial.print("TDS Value: ");
Serial.println(tdsValue);

// Ultrasonic sensor reading
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration_us = pulseIn(echoPin, HIGH);
distance_mm = 0.173025 * duration_us;

// Relay control based on water level
int waterLevel = map(distance_mm, 0, 221, 100, 0);

// LCD display
if (digitalRead(D6) != pos) {
  counter++;
  lcd.clear();
  pos = (digitalRead(D6));
  if (counter > 3) {
    counter = 1;
  }
}

if (counter == 1) {
  lcd.setCursor(0,0);
  lcd.print("Rain:");
  lcd.print(cloudint);
  lcd.print("%");
  lcd.setCursor(8,0);
  lcd.print(",TDS:");
  lcd.print(tdsValue);
  lcd.setCursor(0, 1);
  lcd.print("Waterlevel:");
  lcd.print(waterLevel);
  lcd.setCursor(13, 1);
  lcd.print(" %");
} else if (counter == 2) {
    // ... (other LCD displays)
  } else if (counter == 3) {
    // ... (other LCD displays)
  }

  // ... (pump control based on conditions)

  // GSM Module - Send message if TDS value is above a limit
  int tdsLimit = 300;
  if (tdsValue > tdsLimit && !already_sent) {
    sim800.println("AT+CMGF=1"); // Set SMS mode to text
    updateSerial();
    sim800.println("AT+CMGS=\"+8801615594236\"");
    updateSerial();
    sim800.print("Time to Clean up your tank!");
    lcd.clear();
    lcd.print("Change Water!!!"); // Your message here
    updateSerial();
    sim800.write(26); // ASCII code for CTRL+Z
    delay(2000);
    lcd.clear();
    already_sent = true;
  }

  // Reset the already_sent flag if TDS value is below the limit
  if (tdsValue < tdsLimit)
    already_sent = false;

  delay(500);
}

void updateSerial() {
  delay(500);
  while (Serial.available()) {
    sim800.write(Serial.read()); // Forward what Serial received to
Software Serial Port
  }
  while (sim800.available()) {
    Serial.write(sim800.read()); // Forward what Software Serial received
to Serial Port
```

```
  }
}

// Define a function to send an HTTP GET request and return the response
String httpGETRequest(const char* serverName) {
  // Create a WiFiClient object
  WiFiClient client;
  // Create an HTTPClient object
  HTTPClient http;

  // Begin the HTTP connection with the given server name
  http.begin(client, serverName);

  // Send the HTTP GET request
  int httpResponseCode = http.GET();

  // Declare a variable to store the response
  String payload = "{}";

  // Check the status code of the response
  if (httpResponseCode > 0) {
    // If successful, print the code and store the response
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    payload = http.getString();
  } else {
    // If failed, print the error code
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }

  // Free the resources
  http.end();

  // Return the response
  return payload;
}
```

Code explanation :

Libraries:

- The code includes several libraries for different functionalities:
    - `SoftwareSerial`: For creating a software serial port for communication with the GSM module.
    - `Wire`: For I2C communication.
    - `LiquidCrystal_I2C`: For interfacing with a 2x16 character LCD display using I2C.
    - `ESP8266WiFi` and `ESP8266HTTPClient`: For connecting to a WiFi network and making HTTP requests.
    - `Arduino_JSON`: For parsing and handling JSON data.

Constants and Variables:

- Various pins are defined for connecting different sensors and modules.
- WiFi credentials (`ssid` and `password`) are specified.
- OpenWeatherMap API key is provided.
- Variables to store weather and environmental data, like temperature, humidity, wind speed, etc.
- A flag (`already_sent`) is used to track whether an SMS has already been sent.

Setup Function:

- Initializes serial communication, GSM module, Ultrasonic sensor, Relay, and LCD.
- Connects to WiFi using provided credentials.
- Prints the IP address of the Arduino.
- Prints the timer interval and initializes the GSM module.

Loop Function:

- Checks if a specified time has passed (controlled by `timerDelay`) to make an HTTP request to the OpenWeatherMap API.
- Parses the JSON response to extract weather data.
- Reads TDS (Total Dissolved Solids) sensor and Ultrasonic sensor values.

- Controls a relay based on water level and cloudiness conditions.
- Sends an SMS if TDS value exceeds a certain limit.

Additional Functions:

- `updateSerial()`: For forwarding data between the Arduino's hardware serial and a software serial port used for communication with the GSM module.
- `httpGETRequest(const char* serverName)`: Sends an HTTP GET request to the specified server and returns the response.

LCD Display:

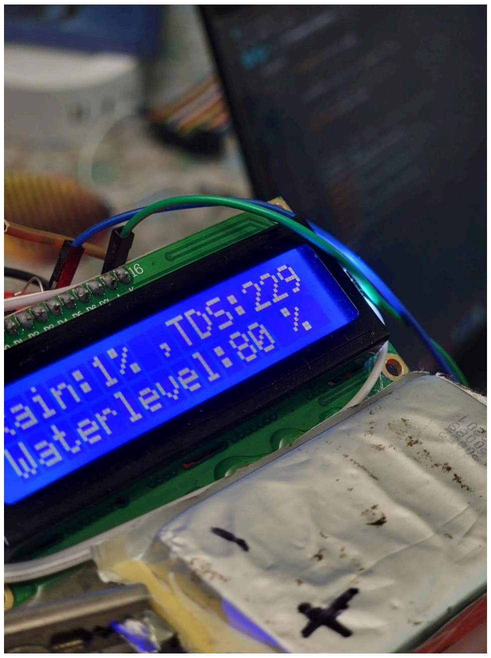
- The LCD is used to display different sets of information based on a counter (`counter`) variable.
- Information includes weather data, TDS value, water level, etc.

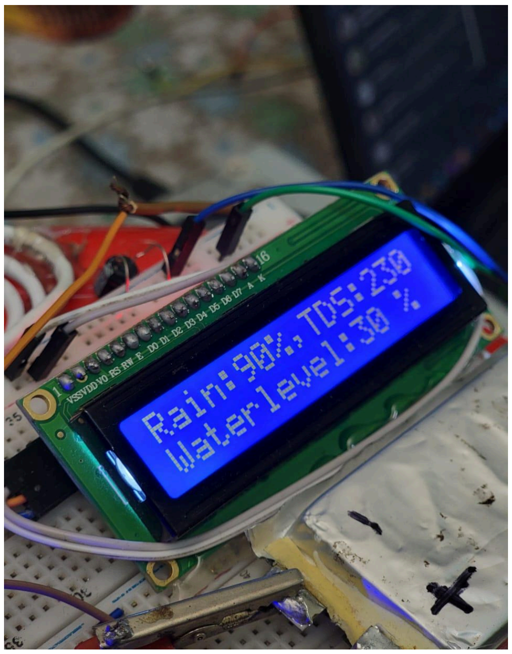
Budget Analysis :

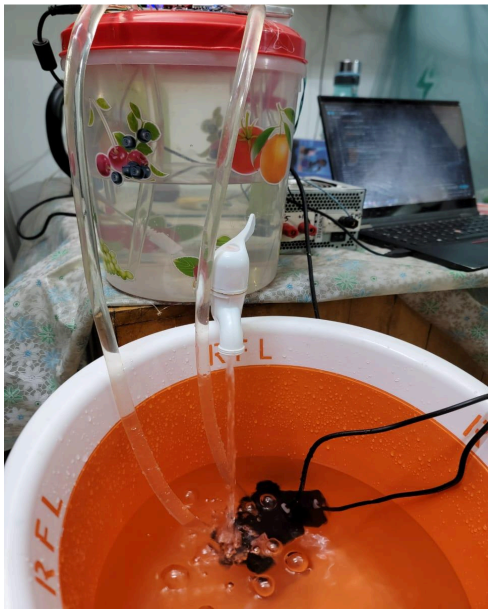
Item No.	Item Name	Quantity	Price(Taka)
1	HC-SR04 sensor	1	300
2	Relay	2	100
3	Analog Tds sensor	1	1000
4	GSM Module	1	300
5	ESP8266 Board	1	350
6	Sim Operation		50
7	Resistors	10	10
8	Diodes	10	10
9	capacitor	10	10
10	Pump	2	590

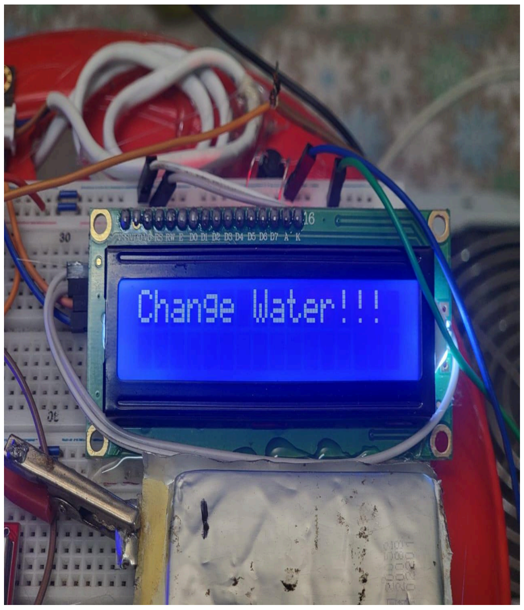
Total = 2720 Taka

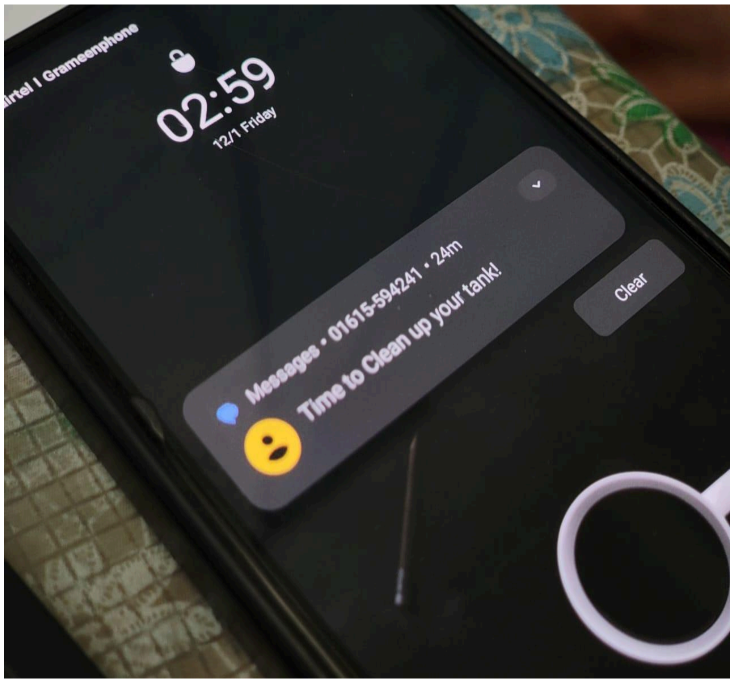
Gallery:


ain:1% ,TDS:229
Waterlevel:80 %


Rain:90%,TDS:230
Waterlevel:30 %

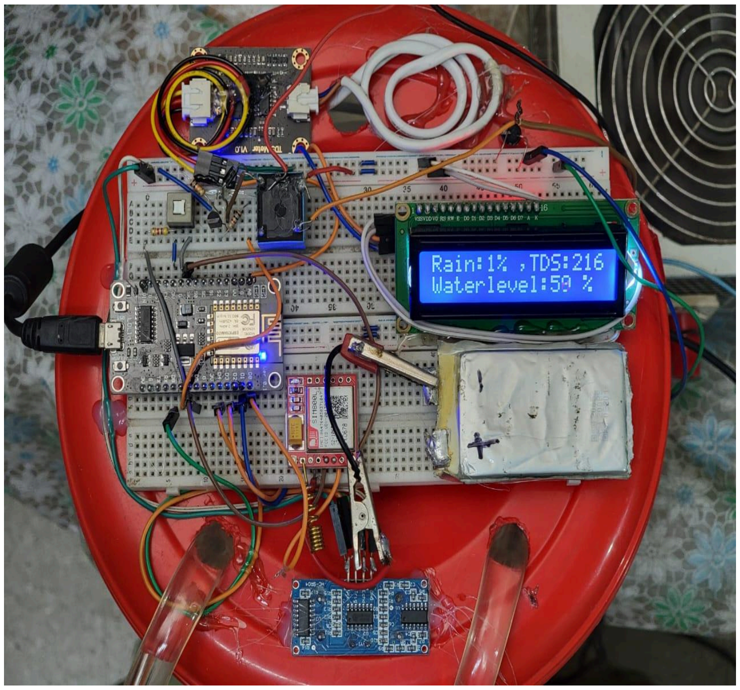



Change Water!!!

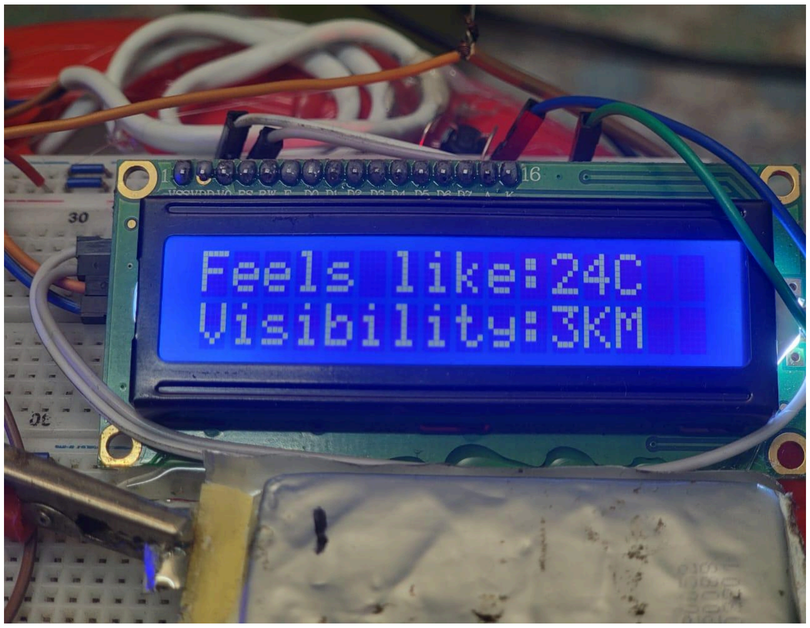


Messages • 01615-594241 • 24m

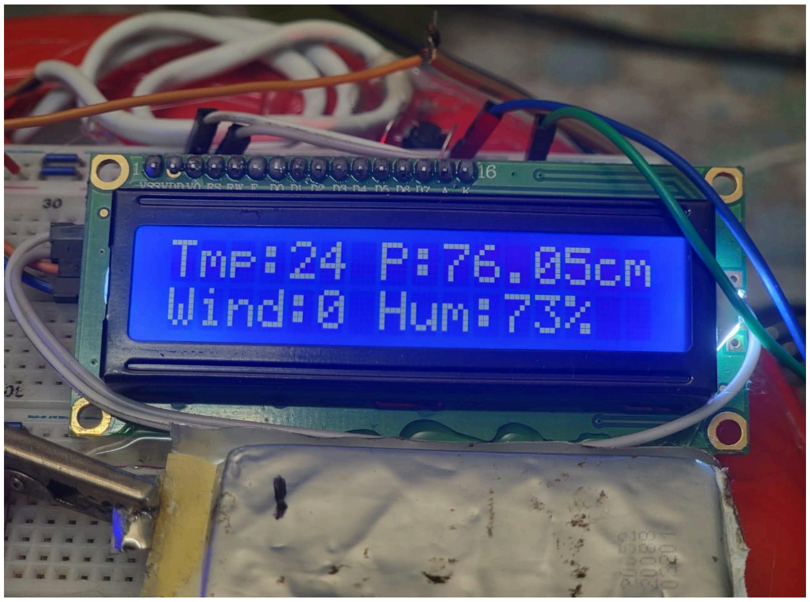
Time to Clean up your tank!



```
Rain:1% ,TDS:216
Waterlevel:59 %
```



```
Feels like:24C
Visibility:3KM
```



```
Tmp:24 P:76.05cm
Wind:0 Hum:73%
```

Conclusion :

In conclusion, the automated water management system presents a robust and versatile solution for monitoring and controlling key aspects of water resources. Leveraging a diverse set of sensors, including ultrasonic sensors for water level detection, TDS sensors for water quality assessment, and a GSM module for real-time communication, the system offers a comprehensive approach to intelligent water management.

The integration of an ESP8266 microcontroller ensures seamless connectivity to the internet, allowing the system to fetch real-time weather data from the OpenWeatherMap API. This integration not only enhances the precision of water management decisions but also demonstrates the potential for incorporating external data sources into smart systems.

The dynamic relay control logic based on both water level and weather conditions showcases the adaptability of the system. By considering factors such as cloud cover in the decision-making process, the project moves beyond basic threshold-based automation, reflecting a thoughtful and context-aware approach to water resource management.

The inclusion of an SMS notification system utilizing the GSM module adds an extra layer of user interaction and alerts. This feature ensures that users receive timely notifications about critical events, such as exceeding TDS thresholds, fostering a proactive and responsive approach to maintaining water quality.

The I2C LCD display provides a user-friendly interface, offering real-time insights into various parameters, including TDS values, water levels, and weather conditions. This visual representation enhances the accessibility of information, making it easier for users to monitor the system's performance.

Despite the complexity of the system, the well-organized code and detailed hardware connections contribute to the project's overall clarity and maintainability. The project not only achieves its primary goal of automating water management but also exemplifies the potential of integrating multiple sensors and communication modules to create a sophisticated and adaptable IoT (Internet of Things) solution.

In summary, this automated water management system stands as a testament to the intersection of sensor technology, data-driven decision-making, and IoT principles. By addressing the nuances of water quality and quantity management, the project underscores the significance of smart solutions in optimizing resource utilization and promoting sustainability in diverse applications.

THE END