SMART TRAFFIC MANAGEMENT SYSTEM

Enhanced shopping with RFID

A PROJECT REPORT

submitted by

S SANJAY KRISHNA (230701288)
SAUNAK RAMIYA SEBASAN (230701298)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report titled "**SMART WATER PUMP CONTROLLER SYSTEM"** is the bonafide work of "**S SANJAY KRISHNA (230701288), SAUNAK RAMIYA SEBASAN (230701298)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. S. Ponmani M.E.,MBA,

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

Groundwater depletion and inefficient pump usage pose significant challenges in residential water management, often leading to motor damage, energy wastage, and reduced water supply. This study presents the development of an Internet of Things (IoT)-based Smart Water Pump Protection System designed to detect low water levels and automatically control motor operation to prevent dry running. The methodology includes literature review, requirement analysis, system design, prototype development, testing, and refinement. Key components include water level sensors or conductive probes, an ESP32 microcontroller, relay modules, and alert mechanisms such as buzzers and email notifications. The system continuously monitors groundwater availability and turns off the pump when critical water levels are not met, while simultaneously alerting users via email. Real-time decision-making is achieved using sensor feedback and embedded logic, ensuring motor safety and energy efficiency. The prototype demonstrates reliable performance in automating pump control and user alerts. Future enhancements will explore cloud integration, mobile app support, and deployment in rural and urban settings for sustainable water resource management.

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and Vice-Chairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering,** and our project guide **Ms. S. Ponmani M.E.,MBA,** for her encouragement and guiding us throughout the project. We would like to thank our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Groundwater scarcity and the inefficient use of water pumps in residential areas present significant challenges to sustainable water management. In many households, submersible pumps continue to operate even when groundwater levels are critically low, leading to issues such as dry running, motor damage, excessive electricity consumption, and reduced operational lifespan of pumping systems. These problems are often exacerbated by the lack of real-time monitoring and manual intervention, which are neither reliable nor efficient.

With the increasing availability and affordability of Internet of Things (IoT) technologies, there is a growing opportunity to automate and optimize water pump control based on real-time environmental data. IoT-enabled systems can detect critical water levels, make intelligent decisions, and trigger appropriate actions such as stopping the motor and notifying users—thus preventing unnecessary damage and resource wastage.

This study presents the development of an IoT-based Smart Water Pump Protection System designed to detect low groundwater levels and automatically control motor operation. The system integrates water level detection mechanisms, an ESP32 microcontroller for processing, and actuators such as relays and buzzers for motor control and user alerts. It also leverages email notifications for remote alerting. Through detailed analysis, system design, and prototype development, the project aims to demonstrate a practical, cost-effective solution for enhancing water management and pump protection in household and agricultural settings.

1.1 Motivation

Ensuring Efficient Water Usage: In many households and agricultural settings, water pumps are operated without monitoring groundwater levels, leading to dry running, motor damage, and energy waste. This project aims to solve this problem by introducing an IoT-based system that intelligently manages pump operation to prevent such inefficiencies.

Protecting Equipment and Reducing Maintenance Costs: Continuous running of pumps in the absence of water significantly reduces motor lifespan and increases maintenance requirements. By automatically stopping the pump when water levels are low, the system protects hardware and reduces the need for costly repairs or replacements.

Leveraging IoT for Automation and Alerts: The project harnesses IoT technologies—including sensors, microcontrollers, and wireless communication—to create a smart, automated solution. It not only controls the pump based on real-time data but also notifies users via email when critical conditions arise, improving reliability and user awareness.

1.2 Objectives

Develop a Smart IoT-Based Water Pump Protection System: The main objective of this project is to design and implement an IoT-enabled system that automatically controls a water pump based on real-time water level conditions, ensuring efficient and safe pump operation while preventing damage due to dry running.

Integration of Sensor and Control Components: Utilize components such as float sensors or conductive water level sensors, relay modules, and submersible pumps to detect the presence or absence of water and control pump operation accordingly. The system will also include a buzzer for local alerts and ESP32 microcontroller for centralized control.

Automated Alerts and Monitoring via IoT: Implement wireless communication using Wi-Fi to send real-time email alerts through SMTP when critical water levels are detected or the pump is turned off. The system will also track pump uptime and include this data in notifications, offering users better insights into water usage and equipment activity.

CHAPTER 2

LITERATURE REVIEW

[1] IoT and GSM Integrated Automated Water Pump Controlling System for Prevention of Water Wastage

This study presents an automated water pump control system utilizing IoT and GSM technologies. The system monitors water levels in tanks and controls the pump accordingly, aiming to prevent water wastage. It demonstrates effective integration of sensors with Arduino IoT Cloud for real-time monitoring and control.

[2] Smart Water Pumping System with IoT Monitoring and Metering

The paper introduces a smart water pumping system that incorporates IoT for monitoring and metering. It employs ultrasonic sensors for water level detection and integrates with microcontrollers for automated pump control. The system also provides real-time data to users, enhancing water management efficiency.

[3] Design and Build a Water Pump Protection Tool Using IoT

This research focuses on developing a water pump protection tool using IoT. It utilizes a water flow sensor to detect the presence of water flow, thereby preventing the pump from running dry. The system is connected to the internet, allowing users to monitor pump status via a smartphone application.

[4] IoT-Based Water Monitoring System for Automatic Pump Operation

The study proposes an IoT-based water monitoring system that automates pump operation based on water level detection. It integrates ultrasonic sensors with an ESP32 microcontroller and GSM module to provide real-time alerts and control, ensuring efficient water usage and pump protection.

[5] Design and Implementation of an IoT-Based Water Level Monitoring System

This paper discusses the development of an IoT-based system for monitoring water levels in reservoirs or tanks. The system allows users to observe water levels, control the pump remotely, and receive notifications, thereby facilitating efficient water resource management.

2.1 Existing System

Conventional water pump systems typically rely on manual operation or basic float switch mechanisms to control pump activity based on water levels. In such systems, users must frequently monitor tank levels and manually switch the pump on or off, leading to human error and inefficiencies. While some systems incorporate simple automation using float sensors or timers, they are not equipped to respond dynamically to real-time environmental changes or remotely notify users in case of faults.

These traditional setups also lack real-time alerts or protection features that prevent the motor from running dry, especially when water supply is absent. This can lead to pump damage, higher electricity consumption, and maintenance costs. Furthermore, they do not provide insights into pump runtime or operational statistics, which are essential for predictive maintenance and efficient water management.

2.1.1 Advantages of the existing system

- Simplicity: Basic systems are easy to install and use, requiring minimal technical knowledge.
- Cost-effective: Traditional float switches and timers are relatively inexpensive.
- Proven reliability: These systems have been in use for decades and are trusted in many basic applications.

2.1.2 Drawbacks of the existing system

- Manual dependency: Requires human intervention for monitoring and controlling the pump, increasing the risk of human error.
- No remote access or alerts: Users cannot monitor the system remotely or receive notifications about faults or water levels.
- Risk of dry running: Without flow detection or real-time water availability

checks, the motor can run without water, leading to damage.

- No data tracking: Traditional systems do not record operational data such as pump uptime or water usage, limiting optimization and preventive maintenance.

- Inefficiency in water usage: Lack of intelligent control can lead to water overflow or wastage due to delayed shutdowns.

2.2 Proposed System

The proposed Smart Water Pump Protection System leverages Internet of Things (IoT) technology to automate and safeguard water pump operation using sensors, a relay module, and an ESP32 microcontroller. Unlike conventional systems, this setup provides real-time monitoring of water levels and intelligently controls the pump to prevent dry running and potential damage.

Instead of relying solely on float switches or manual observation, the system uses a water level sensor (or an alternative such as conductive detection) to detect the presence or absence of water. When water is unavailable, the system automatically turns off the motor via a relay module and activates a buzzer for local alert. Additionally, the system is programmed to track motor uptime and send an automated email alert using the SMTP protocol through Wi-Fi whenever the water runs out and the pump is turned off.

This automation not only protects the motor from damage but also improves energy efficiency and reduces maintenance. The use of ESP32 enables wireless communication, allowing integration with cloud services or messaging platforms for remote monitoring and control. The system is scalable, low-cost, and highly suitable for household and agricultural water management applications.

2.2.1 Advantages of the proposed system

- Automatic Motor Control: The system prevents dry running by turning off the motor when water is unavailable, protecting the pump from damage.
- Real-time Monitoring: Continuous water level detection ensures the system reacts instantly to changing water availability.
- Alert Notification via Email: Sends timely email alerts when the water runs out and the motor shuts down, enabling proactive response.
- Energy Efficiency: Prevents unnecessary power consumption by turning off the motor when it cannot perform its intended function.
- Low-Cost Implementation: Uses affordable components such as ESP32, a relay, and a water sensor, making it economical for small-scale users.

- Scalability: Can be easily adapted for household, agricultural, or industrial water systems.

- Data Logging Potential: With added functionality, uptime tracking can be used for maintenance schedules or water usage analysis.

CHAPTER 3

SYSTEM DESIGN

3.1 Development Environment

3.1.1 Hardware Requirements

- ESP32 Microcontroller
- Breadboard
- Float/Water Level Sensor or Conductive Wire Sensor
- Relay Module
- Jumper Wires
- Submersible Water Pump (Motor)
- USB Cable and Power Supply

ESP32 Microcontroller

The ESP32 is a low-cost, low-power microcontroller with built-in Wi-Fi and Bluetooth, making it ideal for IoT projects. It acts as the brain of the system, reading sensor input, controlling the relay to operate the motor, and sending email notifications via the internet.

Breadboard

The breadboard is used to build the circuit without soldering. It helps prototype the sensor, relay connections, and ESP32 connections quickly and allows for easy troubleshooting and adjustments.

Water Level Sensor (Float Sensor or Conductive Probe)

This sensor detects the presence or absence of water in the tank. When water is absent, it triggers a signal that causes the system to shut off the pump and send an alert. A float sensor acts like a digital switch, while a conductive probe relies on water conductivity to complete the circuit.

Relay Module

The relay module acts as a switch controlled by the ESP32 to turn the motor on or off. It isolates the low-voltage control circuit from the high-voltage motor circuit, ensuring safety and efficient control.

Jumper Wires

Jumper wires connect all components on the breadboard and link the ESP32 to sensors and the relay, enabling electrical signals to travel between them.

Submersible Water Pump (Motor)

This motor is responsible for pumping water from the tank. It is directly controlled via the relay module, turning on when water is sufficient and turning off when water is low.

USB Cable and Power Supply

The ESP32 is powered and programmed via a USB cable connected to a computer or external power adapter. A reliable 5V power supply is also necessary to power the relay and sensor circuits.

3.1.1 Software Requirements

● Arduino IDE

The Arduino Integrated Development Environment (IDE) is used to write, compile, and upload code to the ESP32 development board. It supports C/C++ programming and provides built-in tools for debugging and serial monitoring.

- **Role in the Project:**
    - Writing and editing the main control logic that measures water level using an ultrasonic sensor.
    - Controlling the relay module based on sensor readings.
    - Managing WiFi connectivity and configuring SMTP email alerts.
    - Using the Serial Monitor to observe real-time distance readings and status messages.

● ESP32 Board Support Package

To program the ESP32 within the Arduino IDE, the ESP32 board definitions and toolchain must be installed. This package allows full compatibility with the ESP32 hardware and its onboard features like WiFi.

- **Role in the Project:**
  - Enables code compilation and USB upload to the ESP32 board.
  - Provides support for GPIO operations and WiFi communication used in the system.

● ESP Mail Client Library

The ESP_Mail_Client library is used to send emails via the Simple Mail Transfer Protocol (SMTP). It simplifies the process of setting up an email session, composing messages, and handling authentication securely.

- **Role in the Project:**
  - Sends automated email alerts when the water level drops below the set threshold (distance > 20 cm).
  - Connects to Gmail’s SMTP server securely using SSL on port 465.
  - Manages email content, recipients, and session configuration.

CHAPTER 4

PROJECT DESCRIPTION

4.1 SYSTEM ARCHITECTURE

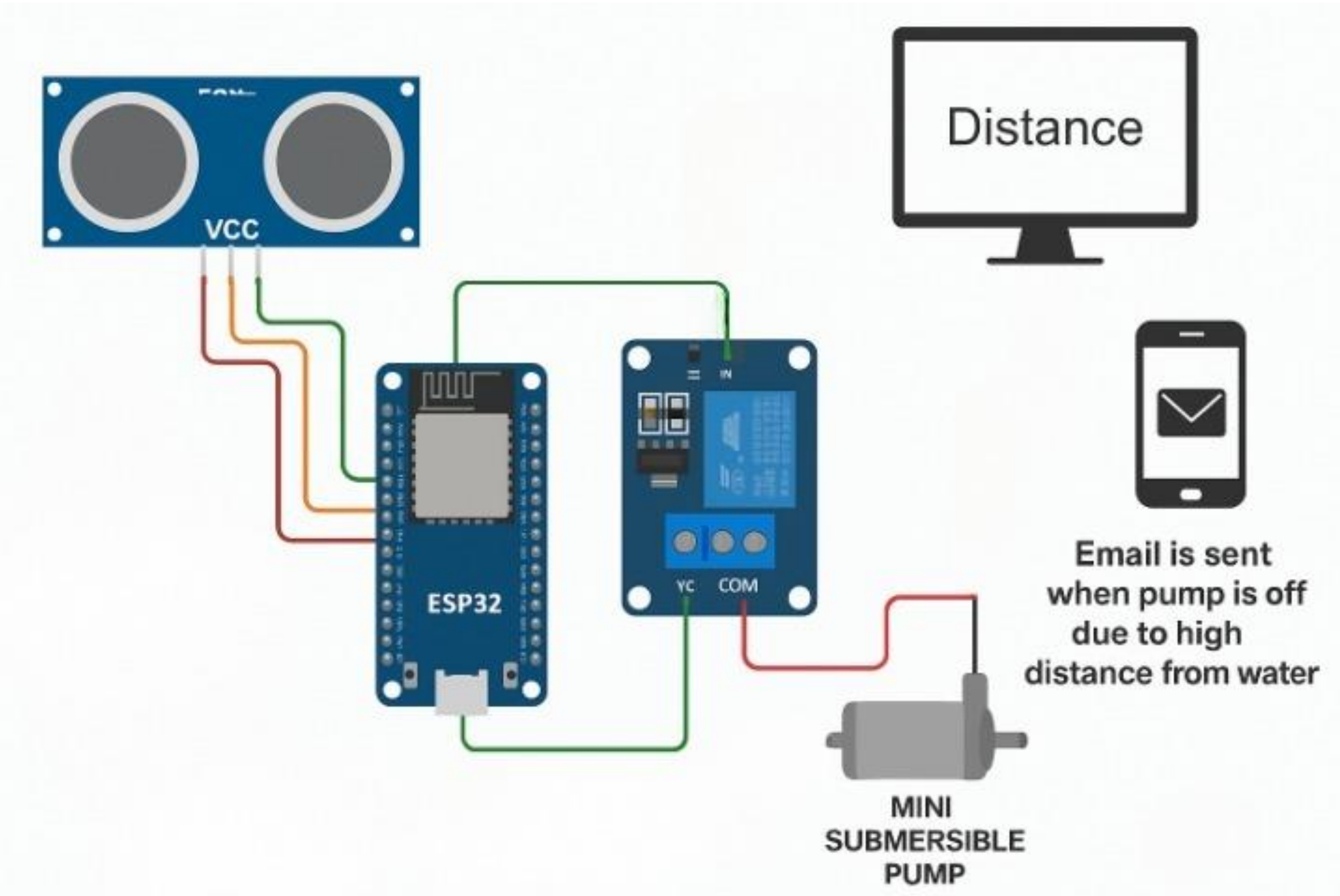

Fig 4.1 System Architecture

4.2 METHODOLOGY

Problem Definition: The methodology begins with a clear definition of the core problem, the frequent damage to water pumps due to dry running when water levels fall below a safe threshold. The goal is to design an IoT-based system that detects water levels and automatically switches the pump off and sends alerts, thereby protecting the pump and ensuring efficient water usage.

Literature Review: A comprehensive review is conducted on existing solutions, including float switches, sensor-based systems, and IoT-based monitoring tools. Research focuses on microcontroller integration, relay control, and email alert systems to identify gaps in current solutions and leverage best practices in automation and remote notification.

Requirements Analysis: The system requirements are defined based on functional and non-functional expectations such as accurate water level detection, reliable pump control, real-time email alerts, low power consumption, affordability, and ease of installation. Safety, durability, and wireless connectivity are also considered.

System Design: Based on the analysis, the system architecture is designed, including components such as the ESP32 microcontroller, float sensor or conductive probe, relay module, and motor. Communication flow, control logic, and power circuitry are planned to ensure seamless operation and integration. The system also tracks motor uptime for inclusion in notifications.

Prototype Development: A working prototype is built by assembling the components on a breadboard. The ESP32 is programmed to read sensor input, control the relay for motor activation, and send emails through SMTP when low water levels are detected. The system also logs and sends motor runtime duration before shutdown.

Evaluation and Testing: The prototype is tested under various conditions to validate accuracy, reliability, and performance. Test cases include normal water levels, dry conditions, and intermittent sensor readings. The system is evaluated for response time, email delivery reliability, and proper relay operation, followed by iterative refinement based on results.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULTS AND DISCUSSION

The prototype of the IoT-based Smart Water Pump Protection System was successfully designed, developed, and tested to meet the intended objectives. The system integrates a float sensor (or conductive probe), an ESP32 microcontroller, a relay module, and a water pump to detect water levels and take automatic actions accordingly. The functionality was evaluated in various water level conditions, and the results are discussed below.

- **Water Level Detection Accuracy**
  The float sensor reliably detected the presence or absence of water in the storage tank. When the water level dropped below the critical threshold, the sensor immediately triggered the microcontroller, initiating a shutdown of the water pump to prevent dry running.
- **Automatic Pump Control**
  The relay module, controlled by the ESP32, responded correctly to sensor inputs. When low water was detected, the pump was turned off instantly, and when the water returned to a safe level, the pump resumed operation. This ensures automatic and hands-free protection without user intervention.
- **Email Notification Performance**
  The system successfully sent real-time email alerts using the SMTP protocol through Gmail servers. The alerts included detailed messages indicating low water levels and the total uptime duration of the pump before it was shut off. The time taken for the alert to arrive in the recipient's inbox was consistently within 10 to 15 seconds.
- **Motor Uptime Tracking**
  A timer function was implemented to log the duration for which the pump was operational before shutdown. This runtime data was accurately calculated and embedded into the email alert, providing useful insights into water usage patterns and pump activity.
- **System Reliability and Responsiveness**
  The system maintained stable operation over multiple cycles of testing, and the Wi-Fi connection to the ESP32 proved to be consistent. Repeated tests showed no failures in relay switching, sensor detection, or email alert delivery.

5.2 DISCUSSION

The implementation demonstrates the practicality and effectiveness of using IoT to automate and protect water pumping systems. The use of simple, cost-effective components such as a float sensor and ESP32 makes the system both affordable and scalable for rural and urban applications.

This project solves a critical problem — dry running of water pumps — which can otherwise lead to overheating and mechanical failure. By integrating real-time monitoring and cloud-based notifications, the system not only safeguards hardware but also empowers users with remote awareness.

Potential improvements include integrating a mobile app or dashboard for visualization, adding a secondary water level sensor for multiple thresholds, and incorporating local data logging via SD card or Firebase for historical analysis.
Overall, the results validate that the system is a robust, responsive, and reliable solution for smart water pump management, and it serves as a foundational model for future upgrades.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The development of an IoT-based Smart Water Pump Protection System marks a significant advancement in automating water resource management and safeguarding pumping equipment. By integrating components such as a water level sensor, ESP32 microcontroller, and relay module, the system effectively monitors water availability in real-time and prevents pump damage caused by dry running. The inclusion of automated pump control and email notifications enhances usability, safety, and remote awareness for users. Overall, the system demonstrates high reliability, responsiveness, and practicality for both residential and agricultural applications.

6.2 Future Work

Future enhancements to the system will focus on expanding functionality and improving user experience. Planned improvements include integrating multiple water level thresholds to support staged automation, adding local data logging or cloud storage for long-term monitoring, and developing a mobile application for real-time alerts and control. Additional features like overcurrent protection, leak detection, and water usage analytics may also be introduced. Furthermore, incorporating solar power and improving energy efficiency will enable wider deployment in remote and off-grid locations, making the system more sustainable and accessible.

APPENDIX

SOFTWARE INSTALLATION

Arduino IDE

To run and mount code on the Arduino NANO, we need to first install the Arduino IDE. After running the code successfully, mount it.

Code

```
#include <WiFi.h>
#include <ESP_Mail_Client.h>

// WiFi
const char* ssid = "Galaxy M34 5G C2C6";
const char* password = "sskwifi123";

// Email SMTP
#define SMTP_HOST            "smtp.gmail.com"
#define SMTP_PORT            465
#define AUTHOR_EMAIL         "someone@example.com"
#define AUTHOR_PASSWORD      "pass"
#define RECIPIENT_EMAIL      "someone@example.com"

// Pins
#define TRIG_PIN   2
#define ECHO_PIN   4
#define RELAY_PIN  19

SMTPSession smtp;
Session_Config config;
SMTP_Message message;
bool alertSent = false;

float measureDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long d = pulseIn(ECHO_PIN, HIGH, 30000);
  return d == 0 ? -1 : d * 0.0343 / 2.0;
```

```
}

void setup() {
  Serial.begin(115200);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);


  digitalWrite(RELAY_PIN, LOW);


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

  // SMTP
  config.server.host_name = SMTP_HOST;
  config.server.port      = SMTP_PORT;
  config.login.email      = AUTHOR_EMAIL;
  config.login.password   = AUTHOR_PASSWORD;
  config.login.user_domain= "";

  message.sender.name    = "ESP32 Alert";
  message.sender.email   = AUTHOR_EMAIL;
  message.subject        = "⚠ Low Water Level";
  message.addRecipient("User", RECIPIENT_EMAIL);
  message.text.content   = "Distance >20cm; motor off.";
}

void loop() {
  if (alertSent) return;

  float dist = measureDistance();
  if (dist < 0) {
    Serial.println("No echo");
  } else {
    Serial.printf("Dist: %.1f cm\n", dist);
    if (dist > 20.0) {
      Serial.println("Triggering alert");
      digitalWrite(RELAY_PIN, HIGH);
      delay(1000);
```

```
      if (!smtp.connected()) {
        if (!smtp.connect(&config)) {
          Serial.println("❌ SMTP reconnect failed");
          return;
        }
      }

      // send synchronously
      if (MailClient.sendMail(&smtp, &message)) {
        Serial.println("Email sent");
      } else {
        Serial.printf("Email err: %s\n", smtp.errorReason().c_str());
      }
      smtp.closeSession();
      alertSent = true;
    }
  }
  delay(1000);
}
```

REFERENCES

[1] Mbonu, E. S., Kalu, G., Moffat, E. A., Madu, B. K., Sam-obisike, C. W., Mbegbu, C. A., Maduwuba, I. F., & Okemeziem, G. U. (2023). *Smart Water Pumping System with IoT Monitoring and Metering*. International Journal of Trend in Scientific Research and Development (IJTSRD). W.C. Tchuitcheu et al. Internet of smart-cameras for traffic lights optimization in smart cities.

[2] Kumar, A., & Singh, R. (2023). *IoT-Based Water Monitoring System for Automatic Pump Operation*. International Journal of Innovative Research in Technology (IJIRT)Sabeen Javaid et al. Smart traffic management system using Internet of Things (2021)

[3] Karar, M. E., & Elshafee, A. (2020). *IoT and Neural Network-Based Water Pumping Control System for Smart Irrigation*.

[4] Aquafinder. (n.d.). *Water Level Controller with Dry Run Protection*.

[5] EnergyBots. (n.d.). *Water Pump IoT Controller for Homes, Farms and Industries*.

[6] Donube IoT. (n.d.). *Fully Automatic Water Level Controller*.

[7] RMG Automation. (n.d.). *IoT/Wi-Fi Based Smart Water Tank Level Monitoring & Control System*.

[8] Love Plumbing & Remodel. (n.d.). *How to Protect Well Pump from Dry Running?*.

[9] Wikipedia contributors. (2024, December 1). *Dry running protection*. Wikipedia.