ACKNOWLEDGEMENT

I would like to take this opportunity to express my heartfelt gratitude to all those who have supported and assisted me throughout the completion of this individual project, the Water Level Tracking Dashboard.

Primarily, I extend my sincere appreciation to my supervisor, Mr. Shyam Sundar Khatiwada for their guidance, expertise, and invaluable feedback. Their continuous support and encouragement have been instrumental in shaping the project and ensuring its successful completion.

I would like to acknowledge the assistance and encouragement provided by my friends and family. Their unwavering belief in my abilities and their constant motivation have been a source of inspiration throughout this project.

I am grateful to Deerwalk Institute of Technology (DWIT College) for providing the necessary resources and facilities for the project.

I would like to express my thanks to Mr. Saroj Dhakal who participated in the testing phase of the Water Level Tracking Dashboard. His willingness to provide feedback and insights greatly contributed to the refinement and improvement of the system.

Lastly, I would like to acknowledge the contributions of all the researchers and developers in the field of IoT, whose work and advancements have laid the foundation for projects like mine.

Once again, I am sincerely thankful to all those who have played a part, big or small, in the completion of this project. Your support and assistance have been invaluable.

Bipashree Aryal
Roll No.: 6-2-1175-55-2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IoT: Internet of Things

WLTD: Water-Level Tracking Dashboard

ESP32: Espressif Systems-32

HCSR04: High-Conductance Ultrasonic Sensor

NPR: Nepalese Rupee

IDE: Integrated Development Environment

Wi-Fi: Wireless Fidelity

IO: Input/Output

RTDB: Real-Time Database

API: Application Programming Interface

CHAPTER 1: INTRODUCTION

1.1. Overview

The Water Level Tracking Dashboard project was initiated due to personal experiences and concerns regarding the hassle of manually controlling water pumps and the decreasing groundwater levels during winter seasons. The project aims to address the problem of inefficient water level management in household and commercial water tanks.

The primary objective of the project is to develop a smart solution that simplifies water level management and optimizes the utilization of precious groundwater resources. providing real-time insights into water consumption patterns, users can make informed decisions regarding their water usage.

The scope of the project focuses on creating a user-friendly dashboard that accurately tracks water levels and offers real-time motor status. While the project aims to enhance water management practices, it is important to note that it does not inherently decrease water consumption. Rather, it empowers users with valuable information to better manage their water usage and promote conscious water management practices.

1.2. Problem Statement

The current problem lies in the inefficient management of water levels in household and commercial water tanks, compounded by the hassle of manually controlling water pumps. This issue arises due to the lack of convenient and automated solutions that provide real-time monitoring of water levels. The existing systems require constant manual intervention, making it challenging for users to effectively manage their water resources. The evidence of this problem can be seen in the frustrating and time-consuming process faced by individuals in maintaining optimal water levels in tanks and the resulting wastage or shortage of water.

The Water Level Tracking Dashboard project proposes a solution to address this problem by developing a smart system that automates the control of water pumps and offers real-time monitoring of water levels. By incorporating advanced sensors and microcontroller technology, the project enables users to remotely monitor the water levels in their tanks. The system will provide accurate and timely information on water levels, empowering users to make informed decisions and take proactive measures to manage water efficiently.

1.3. Objectives

i. Develop a Water Level Tracking Dashboard that automates water pump control based on real-time water level data, simplifying water level management in household and commercial water tanks.
    a. Integrate an ESP32 Microcontroller and HCSR04 Sonar Module for accurate water level monitoring.
    b. Implement a Relay system for automatic execution of the water motor respective to the water level in the tank.

ii. Provide insights into water consumption patterns through the Water Level Tracking Dashboard, enabling users to make informed decisions regarding their water usage.
    a. Display real-time levels on the dashboard.

iii. Enhance water management practices and promote efficient utilization of groundwater resources.

iv. Alleviate the risks caused when people manually inspect water levels in their tanks by climbing beside the tanks.

v. Completely stop the wastage of water caused due to overflow of water tanks.

vi. Offer a cost-effective and user-friendly solution for homeowners and tenants in Nepal, addressing the market void for affordable smart water level management systems.

1.4. Scope and Limitation

1.4.1. Scope

i. The project focuses on the development of a Water Level Tracking Dashboard using the ESP32 Microcontroller and HCSR04 Sonar Module.

ii. The dashboard will provide real-time monitoring of water levels in household and commercial water tanks.

iii. The System will automatically turn the Motor on/off respective to the amount of water in the tank.

1.4.2. Limitations

i. The project's scope is limited to monitoring the water level and fill the water automatically when needed. It does not include mechanisms to directly reduce water consumption.

ii. The accuracy of the water level measurements may be affected by external factors such as temperature changes, tank material and sensor calibration.

iii. The project assumes a stable internet connection for remote access and control of the Water Level Tracking Dashboard.

1.5. Report Organization

This Project Report is Organized as follows:

i. Preliminary Section:

   This Section consists of the Title Page, Abstract, Acknowledgement, Table of Contents, List of Figures, List of Tables, and the List of Abbreviations.

ii. Introduction Section:

   This Section contains the Overview, Problem Statement, Objectives, Scope, and the Limitation of the project.

iii. Background Study and Literature Review Section:

   In this section, the Background Study and the Literature Review are performed before the selection of the project is discussed.

iv. System Analysis:

   This section consists of the Requirement Analysis (Functional and Non-Functional), Feasibility Analysis (Technical, Operational, Economic), Data Modeling, and System Design.

v. Implementation and Testing:

   The Implementation of the project and the testing conducted on the project along with the Test Cases, Input Data and Output Expected are discussed in this section.

vi. Conclusion and Future Recommendation:

   The Outcome of the Project, Conclusion, and the Future Recommendations to further improve the scope of the project are discussed in this section.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1. Background Study

The HCSR04 Sonar Module [1] is an ultrasonic sensor that calculates distance by measuring the time it takes for ultrasonic waves to travel and return. By emitting ultrasonic waves and detecting their reflections, the module can determine the distance to an object. This functionality is essential for accurately measuring the water level in a tank, as the module can be positioned at the top of the tank to measure the distance to the water surface.

The ESP32 microcontroller [2], known for its Wi-Fi capabilities, can be used to post data to a Firebase Realtime Database [3]. Firebase provides cloud-based services, including a Realtime Database that allows for the storage and retrieval of data in real-time. By establishing a Wi-Fi connection and utilizing the Firebase API [4], the ESP32 can send HTTP POST requests containing relevant data. This data is stored in the Firebase Realtime Database, enabling users to access and monitor the water level information in real-time through the Water Tracking Level Dashboard.

The combination of the HCSR04 Sonar Module's distance calculation capabilities and the ESP32's data posting functionality to Firebase Realtime Database forms the foundation of the Water Level Tracking Dashboard project. These technologies enable accurate water level measurements and seamless data communication, facilitating efficient monitoring and control of water levels in tanks.

2.2. Literature Review

After conducting both internet and market research, it was discovered that there are no "Smart" solutions accessible in the market. Existing methods employ a Static On/Off Flow Control Valve, which turns on or off the Inlet Pipe dependent on the level of the water, but this does not manage the condition of the electric motor, resulting in long-term damage to the costly component.

Additional investigation in prominent online retailers yielded no results. Ivent Solutions Limited, situated in New Zealand, sells a comparable product named Smart-Water Online™. Which provides identical functionality to the Water Level Tracking Dashboard but costs roughly $435 for the starting pack alone, which is a staggering 29-Times the price of the hardware that the Water Level Tracking Dashboard utilizes [5].

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1. IMPLEMENTATION

After evaluating diverse options, it was determined that the Raspberry Pi [6], while capable, is expensive and computationally overkill for the project's scale.

The Arduino Uno [7], on the other hand, is a reliable microcontroller but lacks built-in Wi-Fi or Bluetooth capabilities. To make it suitable for IoT-based projects, the addition of a wireless module is required. However, this not only increases the cost but also introduces technical complexities.

Considering these factors, the ESP32 [2] microcontroller emerged as the ideal choice for this project. It offers a great package for the price, with built-in wireless features that eliminate the need for additional modules. The ESP32 provides the necessary capabilities for seamless data communication and connectivity [8].

However, flashing the code to the ESP32 can be a tedious process. It has two modes: Download mode, used for flashing data to the chip, and Hardware mode, used for utilizing the ESP32 as a HID (Human Interface Device) Class device. One notable advantage of the ESP32 over the Arduino Uno is its ability to function as a HID device without requiring additional hardware.

During the code flashing process, it is necessary to hold the boot button on the ESP32 module until the computer establishes a connection with the chip. However, this inconvenience can be addressed by placing a 10µF capacitor between the Enable (EN) and Ground pins of the ESP32 module.

4.1.1. Tools Used

i. **Hardware Components:** ESP32 Microcontroller, HCSR04 Sonar Module, 2-Channel Relay Module, Breadboard, Jumper Wires, Micro-USB Cable.
ii. **Backend Code:** Arduino Programming Language (A Variation of C++).
iii. **Frontend Code:** Flutter (A Framework of Dart Programming Language).
iv. **Database Platform:** Google Firebase Real-Time Database.
v. **IDEs/Code Editors:** Arduino IDE, Visual Studio Code.
vi. **Serial Monitoring Software:** Open Serial Port Monitor.
vii. **Diagrams and Charts:** Draw.io, TeamGantt, Circuit-Diagram.org.
viii. **Documentation:** Microsoft Word, Microsoft PowerPoint.

4.1.2. Implementation Details of Modules

The Implementation of Modules in WLTD can be differentiated into multiple layers as discussed below:

a. Hardware Implementation:
   - **Sensor Integration:** Connection between the HCSR04 Sonar Module and the ESP32 Microcontroller.
   - **Pump Control Interface:** Interfacing between the ESP32 with the Relay Module to enable automated functionalities.

b. Communication Implementation:
   - **Wi-Fi Module:** Utilizing the built-in Wi-Fi capabilities of the ESP32 along with the Wi-Fi Library for connection to the internet and subsequently the Firebase Database.
   - **Firebase Integration:** Interfacing the ESP32 with the Firebase Realtime Database (RTDB) to store and retrieve water level data and pump control status using the Firebase-ESP32 Library.

c. User Interface Implementation:
   - **Mobile Application:** Providing a platform for the users using Flutter to provide an interactive platform for monitoring the water levels and the pump control options.
   - **Real-Time Updates:** Using Flutter's Real-Time Listeners to receive live updates whenever the water level changes.

4.2. TESTING

4.2.1. Test Cases for Unit Testing

S/N	Test Case	Test Input	Expected Outcome	Test Result
1	Sensor Reading	Simulated Water Level Data	Correct water reading is captured	PASS
2	Pump Control	System turns the pump On.	Pump state changes to ON	PASS
		System turns the pump Off.	Pump state changes to OFF	PASS
3	Data Synchronization	New water level recorded	Real-Time update of water level on UI	PASS
4	Database Connectivity	Valid Database Credentials	Successful connection to the Firebase RTDB	PASS
		Invalid Database Credentials	Connection error message displayed	PASS

Table 2: Test Cases for Unit Testing WLTD

4.2.2. Test Cases for System Testing

S/N	Test Case	Test Input	Expected Outcome	Test Result
1	Sensor Reading	Place the sensor in a water tank with known water levels	Water Level data is accurately captured and displayed.	PASS
		Verify the displayed water level readings with the actual water levels.	Water level readings match the actual levels	PASS
2	Pump Control	The Water Level is below the Minimum Threshold	Pump state changes to ON	PASS
		Water Level is above the Maximum Threshold	Pump state changes to OFF	PASS
3	Data Synchronization	Simulate changes in water level data	Real-Time update of water level on UI	PASS

Table 3: Test Cases for System Testing WLTD

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION

5.1. LESSON LEARNT/OUTCOME

During the Water Level Tracking Dashboard project, several valuable lessons were learned that contributed to the overall development and implementation process. These lessons provided insights and guidance for future projects of a similar nature.

Firstly, the importance of selecting the right microcontroller for the project was highlighted. Through the evaluation of different options such as Raspberry Pi, Arduino Uno, and ESP32, it was realized that the ESP32 microcontroller proved to be the most suitable choice. It offered a good balance between cost, functionality, and ease of use, with its built-in wireless capabilities and affordability. This experience emphasized the significance of considering specific project requirements and constraints when selecting hardware components.

Additionally, the process of flashing code to the ESP32 highlighted the need for thorough testing and troubleshooting. Understanding the different modes of the ESP32, such as the download mode and hardware mode, was crucial in ensuring successful code deployment. The discovery of using a 10µF capacitor to address the boot button holding issue demonstrated the importance of resourcefulness and creative problem-solving when encountering technical challenges.

Lastly, the project highlighted the significance of thorough testing at different stages. From unit testing the individual modules to system testing the overall functionality, conducting comprehensive tests helped identify and rectify issues early on. The importance of a robust testing strategy became evident in ensuring the reliability, accuracy, and performance of the Water Level Tracking Dashboard.

In conclusion, the lessons learned throughout the project encompassed hardware selection, troubleshooting, and rigorous testing. These insights will undoubtedly guide future endeavors, emphasizing the significance of thoughtful decision-making, adaptability, and continuous improvement in achieving successful project outcomes.

5.2. CONCLUSION

In conclusion, the Water Level Tracking Dashboard project has successfully tackled the challenge of water level management in household and commercial settings. By leveraging the ESP32 microcontroller and the HCSR04 Sonar Module, this project has offered a valuable solution for monitoring water levels and controlling water pumps.

Through iterative prototyping and comprehensive testing, the project demonstrated the feasibility and effectiveness of using the ESP32 microcontroller as a cost-efficient and feature-rich platform. Its built-in wireless capabilities and affordability make it an ideal choice for implementing IoT-based water level tracking systems.

The successful implementation of the Water Level Tracking Dashboard project highlights the potential of IoT technologies in addressing real-world challenges. It offers a practical and affordable solution for water level monitoring and management, benefiting homeowners and industries in Nepal and beyond. The project's findings and insights can serve as a foundation for future endeavors in smart water management, fostering a more sustainable approach to water resource utilization.

REFERENCES

[1] ElecFreaks, "HC-SR04 Ultrasonic Module User Guide," [Online]. Available: https://www.elecfreaks.com/blog/post/hc-sr04-ultrasonic-module-user-guide.html.

[2] E. Systems, "Arduino-ESP32 Libraries," 2020. [Online]. Available: https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFi.h.

[3] Google Inc., "Firebase Documentation," Google Inc., [Online]. Available: https://firebase.google.com/docs. [Accessed 03 04 2023].

[4] Mobizt, "Firebase-ESP32," 19 04 2019. [Online]. Available: https://github.com/mobizt/Firebase-ESP32.

[5] "Smart Water Online," 16 02 2021. [Online]. Available: https://smartwateronline.com.

[6] The Pi Foundation, "Raspberry Pi Documentation," [Online]. Available: https://www.raspberrypi.com/documentation/. [Accessed 23 03 2023].

[7] Arduino Inc., "Arduino Docs," [Online]. Available: https://docs.arduino.cc/. [Accessed 23 03 2023].

[8] J.-L. Aufranc, "CNX-Software," [Online]. Available: https://www.cnx-software.com/2020/03/24/know-the-differences-between-raspberry-pi-arduino-and-esp8266-esp32/. [Accessed 21 03 2023].

[9] A. N. R. L. Steven Sachio, "CORE," 14 12 2018. [Online]. Available: https://www.core.ac.uk.

[10] F. Dev, "Flutter Documentation," [Online]. Available: https://docs.flutter.dev/.

[11] M. M. A. K. V. M. Shweta Karad, "IoT Based Real Time Water Monitoring System for Smart City," *International Journal of Innovative Science and Research Technology,* p. 251, 2018.

[12] S. C. Priya Jayakrishnan, "Water Level Monitoring System Using IoT," *International Research Journal of Engineering and Technology (IRJET),* p. 1817, 2017.

[13] Espressif Systems, "ESP32 Technical References," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. [Accessed 22 03 2023].