Design Analysis of an IoT based Early Flood Detection and Alerting System

Okeke Remigius Obinna[1], Ehikhamenle Mathew[2]

[1]Department of Electrical/Electronic Engineering, University of Port Harcourt, Nigeria
Email: remigius.okeke@uniport.edu.ng
[2] Department of Electrical/Electronic Engineering, University of Port Harcourt, Nigeria
Email : mathew.ehikhamenle@uniport.edu.ng

Abstract— This study aims to develop and implement an Internet of Things (IoT) based early flood detection system that utilizes weather API parameters to trigger SMS alerts. The system employs an ESP32 microcontroller, integrated with sensing circuitry to measure water levels, and a GSM module responsible for sending timely SMS alerts to both residents and emergency response units in the event of critical flooding. Additionally, the system incorporates a liquid crystal display (LCD) to indicate the current system status and provide real-time weather parameter readings from the digital temperature and humidity module 11 (DHT11). To facilitate efficient monitoring and data access, the system is integrated with the Blynk mobile app, which offers unrestricted access to real-time water level and weather parameter data for both the controller chief and the forecasting unit.

I. INTRODUCTION

Flood is defined as the inundation of substantial volumes of water onto normally dry land, and is regarded as one of the most severe types of natural disasters. It is primarily triggered by heavy rainfall, the overflow of rivers or lakes, the absence of proper drainage systems, and the rapid melting of snow or the impact of storm surges resulting from tropical cyclones or tsunamis in coastal areas. Floods can vary significantly in magnitude, ranging from a few inches to several feet of water [1]. The consequences of flooding can be severe, leading to hardships, damage to crops, loss of life, and extensive destruction of both private and public property.

According to Eric Holdeman, an emergency management writer, there are five common types of floods: river flood, coastal flood, storm surge, inland flooding, and flash flood [2]. Some floods manifest suddenly and recede rapidly, while others develop gradually over several days or even months, owing to variations in size, duration, and affected area [3].

Nigeria faces a considerable risk of flooding, with this natural disaster being the most prevalent in the country. In 2012, Nigeria suffered a total loss of 16.9 billion US dollars, 363 fatalities, and displacement of 2.3 million people from their homes due to flooding, as reported by the National Emergency Management Agency (NEMA) [4]. The three primary forms of flooding in Nigeria are coastal, river, and urban flooding [5]. Factors contributing to flooding in Nigeria include poor or non-existent drainage systems, inadequate waste disposal and management, unregulated urbanization, weak implementation of planning laws, and corruption, as explained by Adaku Jane Echendu, an environment researcher in The Conversation Africa 2022 [6].

Given the devastating effects of floods, flood disaster management becomes crucial. This management process encompasses several phases, including preparedness,

prevention, response, and mitigation [7]. The preparedness phase focuses on minimizing damages through prompt and efficient response strategies, such as prediction and risk zone identification, which can be facilitated by the development of an early flood detection system. The prevention phase involves early forecasting, monitoring, and the formulation of contingency plans before or during an event. The response and mitigation phases encompass activities undertaken after a flood disaster, such as damage assessment, controlling floodwater movement, and relief management, as stated by Glago Frank in [7].

Flood detection and alerting systems play a vital role in predicting potential flood-prone areas and issuing warning messages via SMS to emergency response units and residents of those regions [8]. By leveraging these systems, the impact of flooding can be mitigated, human lives safeguarded, and socio-economic losses reduced.

Furthermore, floods represent a formidable natural disaster with severe consequences, causing extensive economic damage, loss of human lives, threats to other species, and property destruction. In Nigeria, flooding stands as the most common and impactful disaster. The implementation of flood detection systems can greatly aid in predicting flood occurrences, thereby enabling proactive measures to safeguard lives and mitigate socio-economic losses.

II. LITERATURE REVIEW

A series of reviews focusing on flood detection techniques were conducted in the past, often as part of research papers and technical reports covering various geographical regions.

Ankesh Suresh Patil et al [9] developed a flood monitoring system utilizing Arduino. The hardware components included Arduino Uno, water float sensors, an ultrasonic sensor, LED, GSM module, liquid crystal display, and buzzer. The system measured water-levels on roads using the ultrasonic and float sensors, and the data was processed by Arduino Uno and displayed on the LCD. In case of a flood, the GSM module sent SMS alerts to residents until water levels returned to normal. Limitations were identified with the ultrasonic sensor and ZP2058 float switch sensor used for water level measurement due to their restricted measurement range and potential inaccuracies.

Muhammad Ahmad Baballe [10] conducted a comparative study of flood detection systems utilizing Arduino Uno, an ultrasonic sensor, GSM module, LCD, and buzzer. Water levels were measured by the ultrasonic sensor, and when they exceeded the threshold, the

microcontroller triggered the GSM module to send SMS alerts to flood-prone areas. The limitation of the study was also the use of the ultrasonic sensor for water level measurement, with its limited range and susceptibility to false readings.

A. Vijaya Lakshmi et al [11] developed an IoT early flood detection and alerting system using Arduino Uno, an ultrasonic sensor, float sensor, flow sensor, and humidity sensor. These sensors were combined to predict floods and interfaced with Arduino Uno. In the event of a potential flood, authorities and nearby villages were notified via IoT with an alarm through a Wi-Fi module. The limitation was attributed to the use of the ultrasonic sensor for water level monitoring, with its constrained measurement range and susceptibility to false readings due to obstructions.

Ange Josiane Uwayisenga [12] created an IoT-based system for automated floodwater detection and early warning in East Africa. The system included an ultrasonic sensor, DHT22 temperature and humidity sensor, Arduino Uno microcontroller, GSM module, buzzer, and LCD. The ultrasonic sensor measured the distance to the water surface, while the DHT22 sensor measured temperature and humidity. The microcontroller processed the data and sent SMS notifications to alert local authorities, and the buzzer produced an alarm for the community near flood-prone areas. The limitation was related to the use of the ultrasonic sensor for water level monitoring, which had a limited range and could provide false readings when obstructed.

Jasmin Maurya [13] designed a flood avoidance system using IoT with Arduino Uno microcontroller, Wi-Fi module, DHT11 temperature and humidity sensor, water flow sensor, ultrasonic sensor, and LCD. The ultrasonic sensor detected water levels, while the DHT11 sensor measured temperature and humidity. The water flow sensor monitored water transfer. The system's data was displayed on the LCD, and IoT connectivity was provided through the Wi-Fi module. Limitations were identified with the use of the ultrasonic sensor for water level measurement, given its limitations in range and sensitivity to temperature variations.

Minakshi Roy et al [14] implemented a flood detector and water monitoring system using IoT with Arduino Wemos D1 R1, ultrasonic sensor, flow sensor, DHT11, float sensor, and liquid crystal display. The ultrasonic sensor measured water height, the flow sensor measured flow rate, and the DHT11 sensor measured temperature and humidity. The float sensor acted as a switch to send data to Arduino Wemos D1 R1, which triggered SMS alerts to an android application. Limitations were attributed to the use of the ultrasonic sensor and float

sensor for water level measurement, with the latter not precisely indicating water level height.

Wan Hassan et al [15] designed a flood indicator water level monitoring system with Arduino Uno, GSM module, ultrasonic sensor, and liquid crystal display. A float switch sensor determined water levels and sent data to Arduino Uno, which controlled the GSM module to send SMS alerts to authorities and the public. The limitation was the water measuring device, ZP2508 float switch sensors, which only detected rising water levels without precisely measuring height.

J.G. Natividad and J.M Mendez [16] built a flood monitoring and early warning system with an ultrasonic sensor, GSM module, and Arduino microcontroller. The ultrasonic sensor measured water distance, and when it exceeded the set threshold, the microcontroller triggered the GSM module to send warning alerts to response agencies. The limitation was the use of the ultrasonic sensor for water level measurement, which had limited range and was sensitive to temperature variations.

Yasir Hashim et al [17] designed a wireless flood monitoring system with an ultrasonic sensor for water level detection. GSM module and Bluetooth module were used for communication, and an MS-190 mini motor siren alerted locals during critical water levels. Limitations were attributed to the use of the ultrasonic sensor for water level measurement due to its limited detection range and sensitivity to obstructions.

Sukanth Behera and Saradiya Kishore Parida [18] developed an IoT-based flood monitoring and alerting system using Arduino Uno with rain sensor, water level sensor, and GPS. The Arduino Uno sent information or notification alerts to nearby villages and higher authorities based on sensor data. The limitation was the use of the water level sensor, which wasn't designed for submersion, leading to a short lifespan.

Dedi Satria et al [19] designed a google maps-based flood monitoring system with Arduino Uno, GSM module, ultrasonic sensor, and GPS U-Blox Neo 6 Module. Ultrasonic sensor measured water level, and GPS provided coordinates of the flood location to Arduino Uno, which sent data through SMS to an information system station. Limitations were related to the water level measuring device, the ultrasonic sensor, due to its limited detection range and sensitivity to temperature variations.

Amit Kumar et al [20] built an IoT-based early flood detection and precaution system with Arduino Uno, ultrasonic sensor, flow rate sensor, DHT11, ESP285 Wi-Fi module, and liquid crystal display. The ultrasonic sensor measured water level height, and data was uploaded to a cloud database for viewing on an android application.

Limitations were attributed to the use of the ultrasonic sensor for water level measurement, given its limited range and sensitivity to obstructions.

Edwin et al [21] designed a flood detector using Arduino Yun microcontroller, ultrasonic sensor, LED, solar power bank, and camera. The ultrasonic sensor measured flood height, and the camera provided live stream feeds. Limitations were identified with the ultrasonic sensor for water level measurement due to its limited range and susceptibility to obstructions.

III.   RESEARCH METHODOLOGY

An early flood detector is a comprehensive device designed for the prediction and detection of floods. It encompasses distinct units, including the sensing unit, processing unit, communication and display unit, and integration with the Blynk mobile app.

The sensing unit comprises a calibrated floater and a DHT11 module. The calibrated floater is equipped with a floater attached to a potentiometer, enabling precise water level measurements. Two thresholds are set within the floater, with the first threshold indicating a rising water level and the second threshold indicating a critical water level. The DHT11 module measures real-time weather parameters, including temperature and humidity, which are forwarded to the weather application programming interface (API) databank for analysis and weather predictions at specific geographical locations.

The communication and display unit includes a liquid crystal display, a buzzer, GSM module, and Blynk mobile app integration. The liquid crystal display presents essential information such as water level, temperature, and humidity values. The buzzer generates an alarm to alert operators at the control room when the water level surpasses the first threshold, signaling increased attention to rising water levels, though not necessarily indicating a flood situation. On the other hand, the SMS alert function, managed by the SIM800 GSM module, is triggered when the water level exceeds the second threshold, clearly indicating a flood. It initiates warning SMS messages to notify residents in the area, urging early evacuation to safeguard lives and valuable properties. Additionally, SMS alerts are sent to the emergency response unit, preparing them to respond promptly to potential life-threatening situations.

The system is powered by a 7.5-volt rechargeable direct current (DC) battery. To ensure the appropriate voltage supply of 5 volts for each component, a buck converter is utilized, effectively regulating the voltage. This precautionary measure safeguards the components

and circuit boards from potential failure and hazards, such as fire, that may arise due to excessive voltage.

Furthermore, the flood detector system is seamlessly integrated with the Blynk mobile app, providing the chief controller and forecast unit with unrestricted access to real-time data on water levels and weather parameters via their mobile phones. This integration facilitates faster rainfall predictions and overcomes geographical limitations,

allowing access to real-time water level and weather information from outside the control room.

This section presents the design methodology employed in this study, outlining the constituent units comprising the system. Furthermore, this section expounds upon the flowchart algorithm and presents the block and circuit diagrams of the work, adopting a Top-down approach for clarity and coherence. The overall system design is represented in the Diagram seen in fig. 1 below.
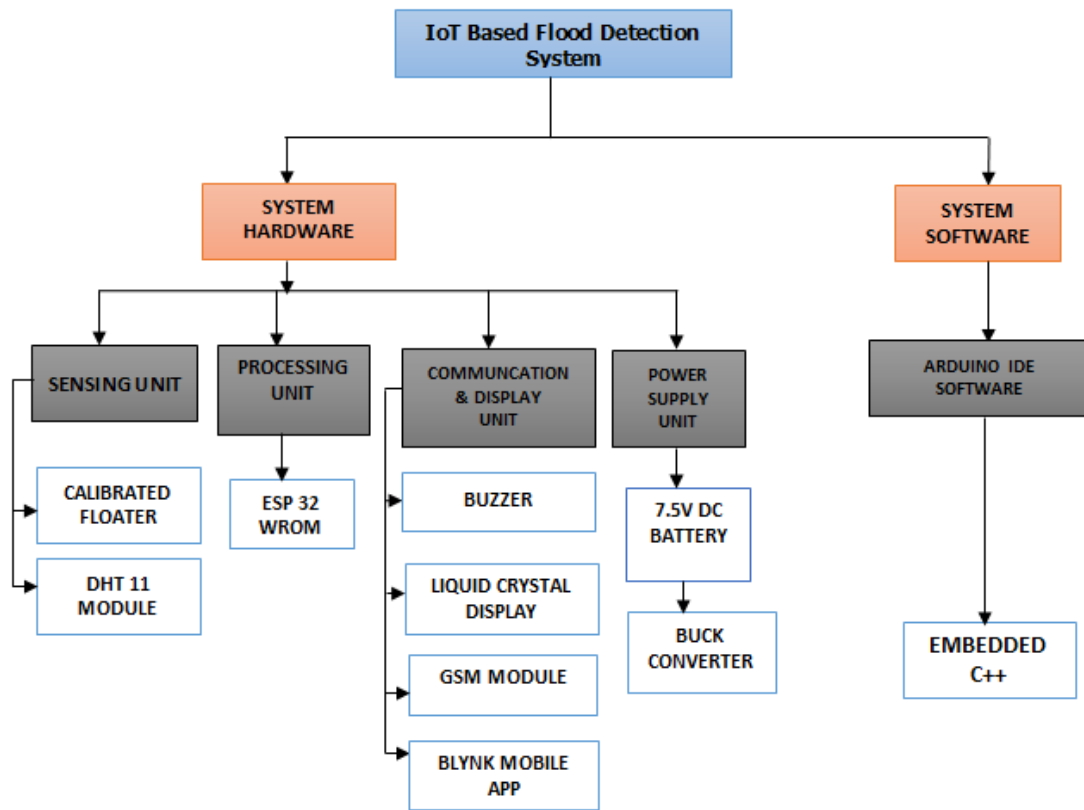

Fig. 1: Block Diagram of the IoT Flood Detection system.

1.1   Hardware (equipment) Used

1.   ESP 32 WROM microcontroller
2.   Floater
3.   10 Kilo ohms potentiometer
4.   DHT 11 module
5.   Buzzer
6.   16x2 liquid crystal display
7.   SIM 800 GSM module
8.   Smartphone with Blynk app installed
9.   7.5v rechargeable DC battery
10.   Plastic bowl
11.   Knockout box
12.   Switch
13.   Buck converter

From the datasheet, table 1 describes the common operating voltage required by the components and their respective current rating.

Table 1: Component Voltage and Current Specification

S/N	Component	Voltage rating (V)	Current rating(mA)
1	DHT 11	5	2.5
2	ESP 32	5	1200
3	Buzzer	5	20
4	LCD	5	1.5

5	SIM 800 GSM	5	1500

The following calculations were performed to determine the kind of power supply which depends on the total voltage and current required to efficiently power the circuit.

The determination of the total voltage required to power the circuit is based on the common operating voltage rating obtained from the datasheets of the individual components. As indicated in the component description provided table 1, each component necessitates a 5-volt input, thus establishing the minimum voltage required for the entire circuit as 5 volts. Given that the components are connected in parallel, the voltage supplied to each component remains consistent:

$V_{total} = V_{DHT11} = V_{ESP32} = V_{Buzzer} = V_{LCD} = V_{GSM} = 5$ volts.

To meet this voltage requirement, two 3.75-volt lithium batteries are utilized, chosen based on material availability. These batteries are connected in series, producing a total voltage of 7.5 volts. To ensure proper regulation of the voltage input to the various components, a buck converter is integrated into the power supply. This converter serves the crucial function of maintaining a stable 5-volt supply, enabling the efficient and safe operation of the circuit.

The total current required for the circuit is determined by summing the individual current ratings of the various components, as they are connected in parallel. Although the ESP 32 microcontroller features a 5V pin that could potentially power interfaced components, it was not directly utilized for this purpose to ensure optimal performance. This decision was based on the fact that the total current rating of the components surpasses the maximum output current of the microcontroller.

The batteries, connected in parallel, possess the same current rating. Specifically, each battery has a current rating of 3800mAh. This rating exceeds the total current required to power the circuit, ensuring sufficient capacity to effectively operate the system.

The calculation of the total current (I_{total}) is as follows:

$I_{total} = I_{DHT11} + I_{ESP32} + I_{Buzzer} + I_{LCD} + I_{GSM}$

$I_{total} = 2.5mA + 1200mA + 20mA + 1.5mA + 1500mA = 2724mA$

Considering the individual current ratings and the capacity of the batteries, they serve as suitable power sources for the circuit, enabling its effective functionality. The circuit diagram for the hardware is shown in fig. 2.
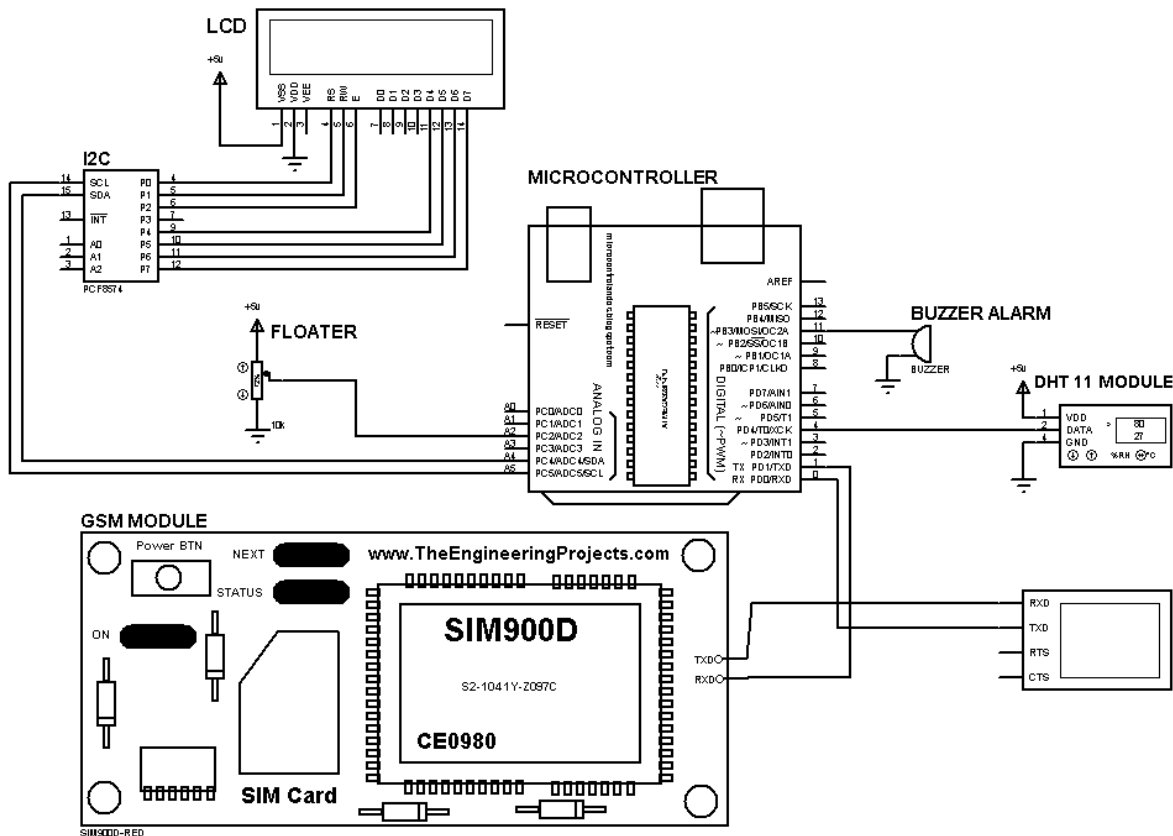
Fig. 2: Circuit Diagram of the System.

1.2 System Software

The ESP 32 WROM microcontroller utilized in this research was programmed using the embedded C++ programming language. C++ is a favored choice for developing software for microcontrollers due to its efficiency and extensive features that facilitate the creation of both simple and complex programs. Embedded C++ serves as a high-level programming language, whereas the native language understood by microcontrollers is machine language. To program the microcontroller using a high-level language, a compiler is essential to translate the high-level code into machine language. Consequently, an Integrated Development Environment (IDE) becomes crucial for this purpose.

An IDE is a software application that offers a comprehensive set of tools to assist programmers in writing, debugging, and maintaining software. These tools typically encompass a source code editor, a compiler or interpreter, a debugger, and various utilities for control, testing, and profiling. IDEs play a pivotal role in streamlining the development process, providing a cohesive environment for programming microcontrollers effectively. They can be tailored for specific purposes or used as general-purpose development platforms.

Arduino is a widely adopted open-source platform designed for creating electronics researches. The Arduino Software (IDE) is a free software application that operates on a computer, enabling users to write programs and subsequently upload them to the physical Arduino board. The IDE encompasses a text editor, compiler, and programmer, providing a comprehensive toolkit for programming convenience.

This section serves as a valuable reference for developers during the programming process. Moreover, the Arduino IDE can be enhanced to accommodate the ESP 32 Board through the Board Manager feature, allowing for seamless programming and code upload to the ESP microcontroller. By incorporating the ESP 32 Board, users can leverage its capabilities in conjunction with the familiar Arduino programming environment, enabling the development of sophisticated electronics projects.
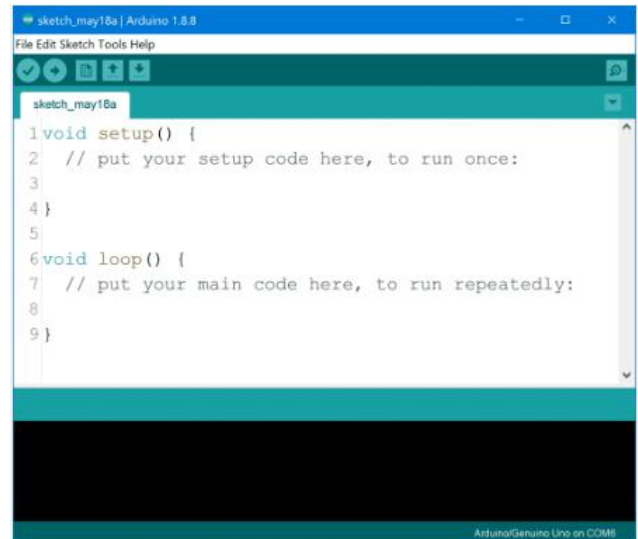

Fig. 3: Arduino IDE Interface

1.3 Flowchart Algorithm

The flowchart diagram in fig. 4 illustrates the step-by-step algorithm followed by the system. Upon activation, the sensors read water level, temperature, and humidity analogue values, which are subsequently transmitted to the microcontroller as inputs. The microcontroller then converts these analogue values into digital signals and outputs them on both the LCD and the Blynk app.

Next, the system measures the water height against two distinct thresholds. If neither threshold condition is met, the microcontroller continues to display the water level on the LCD. However, if the first threshold is met, the buzzer is triggered, and a caution message is displayed on the LCD every 2 seconds, accompanied by the water level, temperature, and humidity values.

In the event that the second threshold is met, the buzzer is again triggered, but this time a warning message is displayed on the LCD every 2 seconds, alongside the water level, temperature, and humidity values. Additionally, the system activates the GSM module to send a warning SMS to the residents of the affected area, the emergency response unit, and the government.
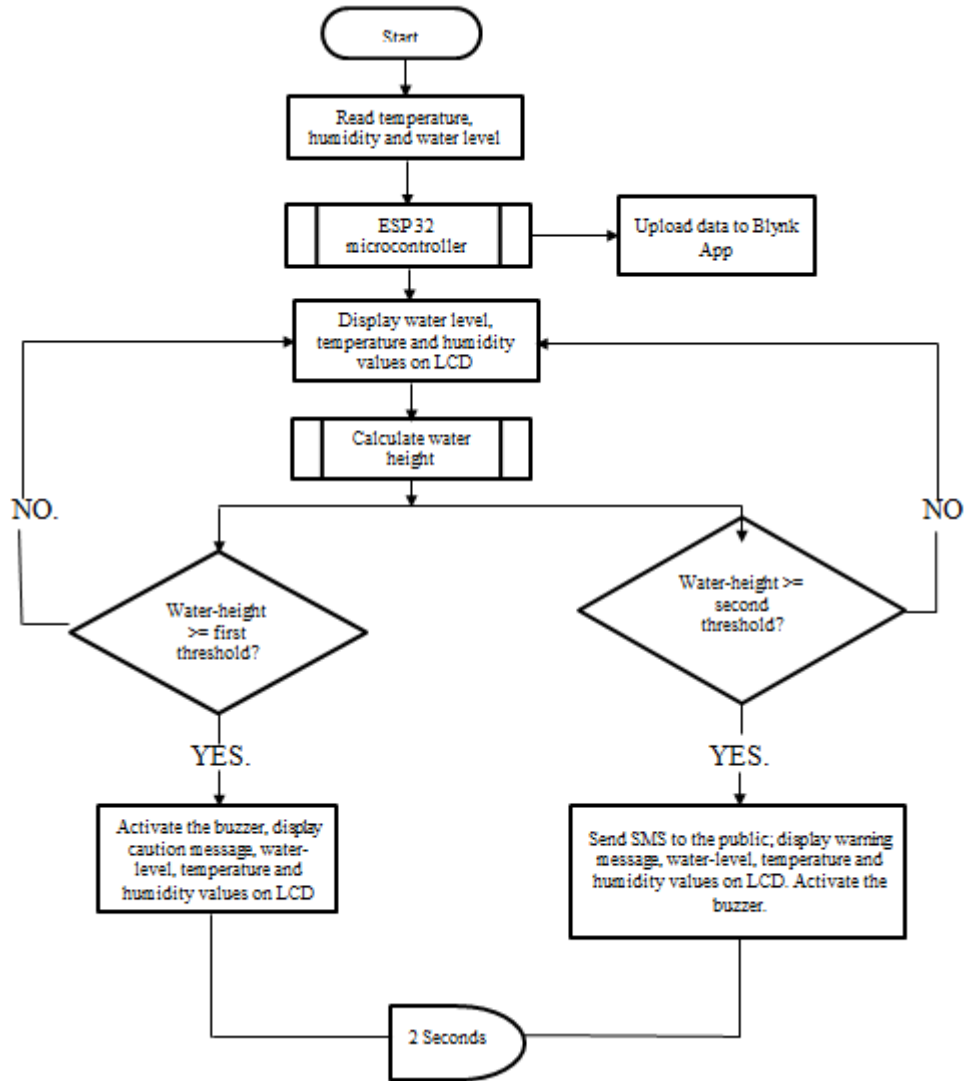
Fig.4: Flowchart Diagram

IV. RESULTS AND DISCUSSION

The proposed flood detector system underwent simulation to assess its viability and significance prior to the actual construction. The simulation process was carried out utilizing Proteus 8 Professional software as shown in the figure 6. The complete system development is partitioned into distinct hardware and software components. By constructing and testing individual segments of the system, the process achieved enhanced speed, efficiency, and minimized debugging time.

Prior to conducting the test, a thorough inspection of the battery was performed to ascertain if there was any physical damage. Subsequently, the batteries underwent individual testing, as well as testing while connected in series, to verify that they were capable of providing the necessary voltage and current required by the circuit. The battery test was conducted using a digital multi-meter as shown in fig. 5.
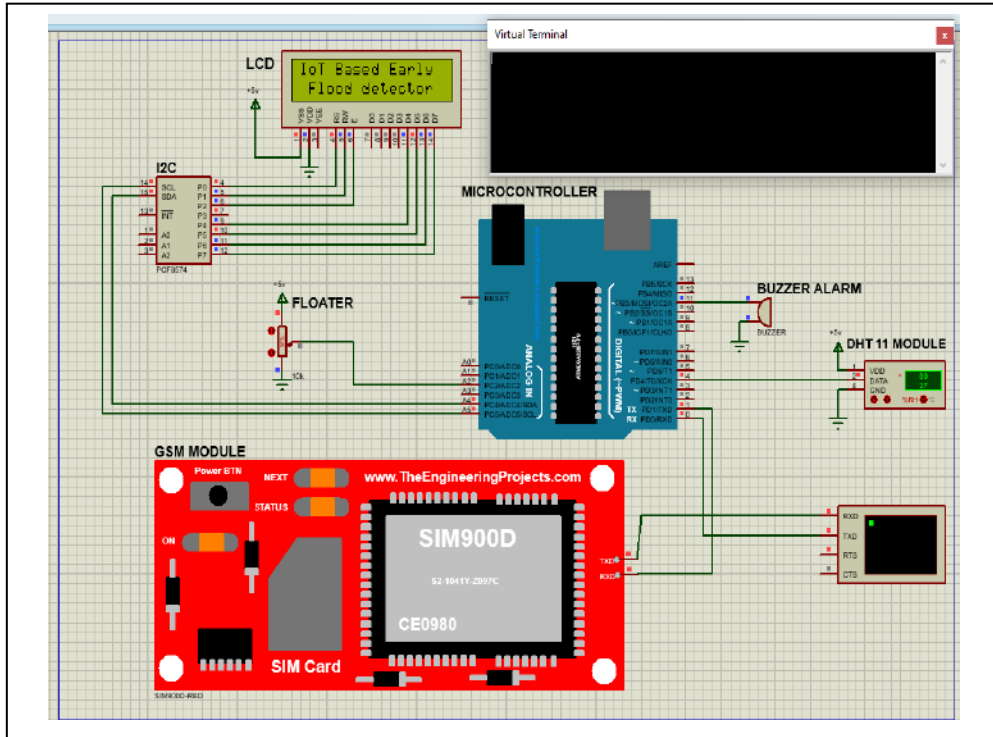

Fig. 5: Battery Test showing the Measured Voltage.

Fig. 6: Proteus Screenshot showing System simulation

Once the system is powered, it immediately measures water-level, temperature and humidity values. The initial threshold is activated when the water level surpasses 20 meters but remains below 44 meters. In such cases, a caution message will be exhibited on the LCD, and the buzzer alarm will be activated. Individual testing of each sensor was conducted to verify the accuracy of their water-level, temperature, and humidity readings, as well as to ensure their proper connection to the circuit. Additionally, the functionality of the sensor values and the warning messages displayed on the LCD was also examined through testing. Simulated and real test are shown in fig. 7.
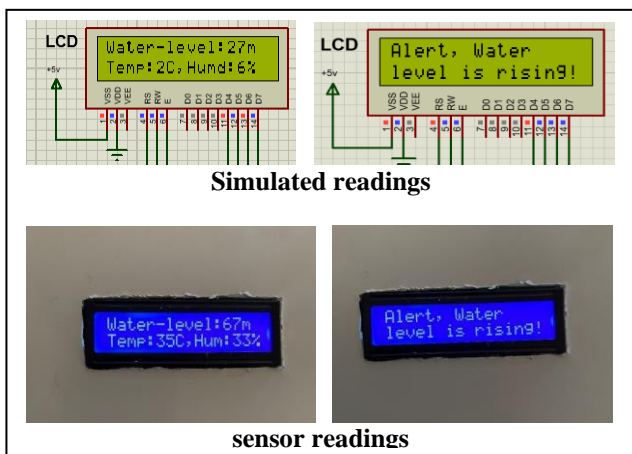

Fig. 7: 1ˢᵗ threshold results

Simulated Virtual Terminal (Screen) displaying the sent SMS.
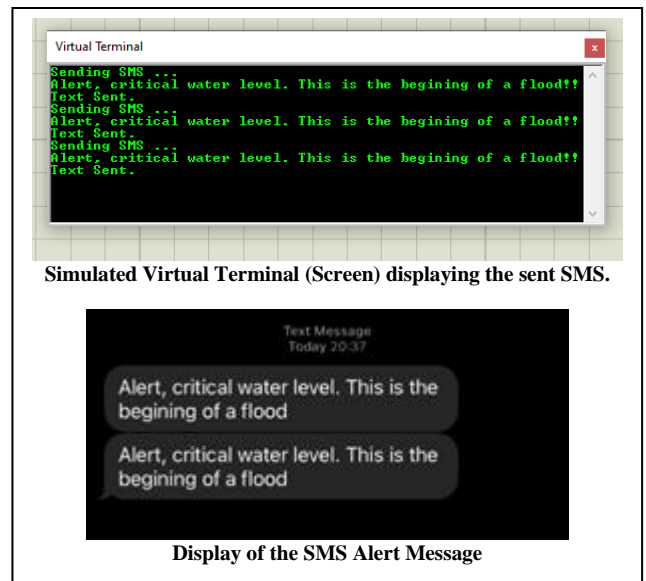

Display of the SMS Alert Message

Fig. 8: 2ⁿᵈ threshold results

The second threshold is reached when the water level rises to 45 meters and above. At this point, the system will activate the buzzer alarm, display a warning message on the LCD, and the GSM module will be utilized to send warning messages to both the emergency response units and the residents within that area. A comprehensive test was performed on the SIM800 GSM module to verify its

proper functioning in sending alert messages to the pre-programmed phone numbers when the second threshold value is triggered. Both simulated and real-time results are depicted in the accompanying figures. Both simulated and real test are shown in fig. 8.

The data synchronization of the sensor and floater values to the Blynk cloud were tested to ensure accurate display in the Blynk app.

Fig. 8: Displaying Sensor Values in the Blynk App

V.   CONCLUSION

This study presents the successful development and realization of a fully functional Internet of Things (IoT) based early flood detection and alerting system, integrating weather application programming interface (API) parameters. This prototype has demonstrated its efficacy in saving lives, protecting properties, and minimizing damages caused by flood disasters. To enhance flood management and mitigation efforts in Nigeria, it is imperative that responsible agencies, such as the Nigeria Hydrological Services Agency (NIHSA), the National Emergency Management Agency (NEMA), and other local government agencies and disaster management organizations across the states, equip themselves with this IoT flood detector and alerting systems. These systems enable effective real-time monitoring of water levels, facilitating proactive responses to potential flood events.

One notable contribution of this work to the existing body of research on flood detection systems lies in the incorporation of a calibrated floater for accurate water level measurement, SMS alert functionality, and real-time monitoring of water height, temperature, and humidity using the Blynk app.

Based on the observations made during the study, certain recommendations have been identified to further improve the system's performance. Firstly, integrating a solar panel with a battery can extend the system's operational duration, ensuring prolonged functionality even during extended periods of low sunlight. Secondly, the development of a third-party IoT app that provides water level data and weather forecasts to the general public without user restrictions, as opposed to the limited information sharing capabilities of the Blynk app, would enhance accessibility and outreach.

Moreover, the system's accuracy and efficiency can be enhanced by utilizing a custom-made spherical calibrated floater for water level measurement, as it offers better water displacement, thereby contributing to more precise readings.

In conclusion, the implementation of this IoT based early flood detection and alerting system, equipped with weather API parameters, offers significant potential to safeguard lives, protect properties, and mitigate flood-related damages. The integration of such systems by relevant agencies and the adoption of recommended improvements hold the promise of enhancing flood management practices and disaster response capabilities.

REFERENCES

[1] Hirsch, R. M., & Ryberg, K. R. (2012). Has the magnitude of floods across the USA changed with global CO_2 levels? *Hydrological Sciences Journal*, *57*(1), 1-9.

[2] Holdeman, E. E. (2023). Emergency Management's Journey with Technology. In *Disaster Management and Information Technology: Professional Response and Recovery Management in the Age of Disasters* (pp. 3-23). Cham: Springer International Publishing.

[3] Yang, H., Wang, H., Lu, J., Zhou, Z., Feng, Q., & Wu, Y. (2021). Full lifecycle monitoring on drought-converted catastrophic flood using sentinel-1 sar: A case study of poyang lake region during summer 2020. *Remote Sensing*, *13*(17), 3485.

[4] Danhassan, S. S., Abubakar, A., Zangina, A. S., Ahmad, M. H., Hazaea, S. A., Ishak, M. Y., & Zhang,

J. (2023). Flood Policy and Governance: A Pathway for Policy Coherence in Nigeria. *Sustainability*, *15*(3), 2392.

[5] Ajumobi, V. E., Womboh, S. B., & Ezem, S. B. (2023). Impacts of the 2022 flooding on the residents of Yenagoa, Bayelsa State, Nigeria. *Greener Journal of Environmental Management and Public Safety*, *11*(1), 1-6.

[6] Echendu, A. J. (2023). Human factors vs climate change; experts' view of drivers of flooding in Nigeria. *Natural Hazards Research*, *3*(2), 240-246.

[7] Glago, F. J. (2021). Flood disaster hazards; causes, impacts and management: a state-of-the-art review. *Natural hazards-impacts, adjustments and resilience*, 29-37.

[8] Subashini, M. J., Sudarmani, R., Gobika, S., & Varshini, R. (2021, February). Development of Smart Flood Monitoring and Early Warning System using Weather Forecasting Data and Wireless Sensor Networks-A Review. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 132-135). IEEE.

[9] Patel, A., Kumar, A., Singh, S., Rani, R., & Usmani, M. H. (2022). IOT Based Early Flood Detection and Avoidance. *Available at SSRN 4159183*.

[10] Baballe, M. A., & Abbati, Z. A. (2022). A Review of Flood Detection Systems. In *1st International Conference on Applied Engineering and Natural Sciences, Konya, Turkey May*.

[11] Lakshmi, R. A., Lakshmi, M. M., Swetha, P., & Prakash, T. D. (2020). FLOOD DETECTION AND EARLY WARNING SYSTEM.

[12] Uwayisenga, A. (2021). *IoT-based system for automated floodwater detection and early warning in the East Africa Region: a case study of Arusha and Dar es salaam, Tanzania* (Doctoral dissertation, NM-AIST).

[13] Maurya, J., Pant, H., Dwivedi, S., & Jaiswal, M. FLOOD AVOIDANCE USING IOT.

[14] Roy, M., Pradhan, P., George, J., & Pradhan, N. (2020). Flood Detection and water Monitoring system using IoT. *International Journal Of Engineering And Computer Science*, *9*(07), 25113-25115.

[15] Hassan, W. H. W., Jidin, A. Z., Aziz, S. A. C., & Rahim, N. (2019). Flood disaster indicator of water level monitoring system. *International Journal of Electrical and Computer Engineering*, *9*(3), 1694.

[16] Natividad, J. G., & Mendez, J. M. (2018, March). Flood monitoring and early warning system using ultrasonic sensor. In *IOP conference series: materials science and engineering* (Vol. 325, No. 1, p. 012020). IOP Publishing.

[17] Hashim, Y., Idzha, A. H. M., & Jabbar, W. A. (2018). The design and implementation of a wireless flood monitoring system. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(3-2), 7-11.

[18] Behera, S., & Parida, S. K. (2018). IoT-based Flood Monitoring and Alerting System using Arduino Uno. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *9*(3), 702-706.

[19] Satria, D., Yana, S., Munadi, R., & Syahreza, S. (2017). Prototype of Google maps-based flood monitoring system using Arduino and GSM module. *Int. Res. J. Eng. Technol*, *4*(10), 1044-1047.

[20] Kumar, M. A., Singh, V., Chaubey, V. K., Bisht, V. S., & Sharma, S. (2016). Iot Based Flood Detector and Early Precaution Method Instructor to Avoid Any Hazardous Event.

[21] Guadaña, R. R., Santos, R. A., De Guzman, E., Cuadra, V. S., De Luna, A. G., & Villanueva, C. (2016, May). FLood Detector System Using Arduino. In *Proceedings of 37th The IRES International Conference*.

[22] Mokwe U.V, Omijeh B.O, Eludoyin O.S, Okeke R.O (2023). Development of Flash flood warning System for Nkpolu Road, Rumuigbo Port Harcourt. Vol 8(4)