TITLE

Smart IoT Washing Machine: A Dual-Controller Approach for Remote Automation and Monitoring

AUTHORS

[Shamsul Hakimee Bin Shamsul Hairee], [Qashrina Aisya Binti Mohd Nazarudin], [Nurel Humairah Binti Muhammad Firdaus] Department of Mechatronics Engineering, [IIUM]
Email: shamsulhakimee@gmail.com

ABSTRACT

This paper details the development of a Smart IoT Washing Machine, a specialized home automation system that assists users by fusing real-time actuator control with remote monitoring capabilities. The system is built on a multi-tier architecture: an Arduino Mega 2560 (MCU) manages time-critical sensor readings (Water Sensor, Buttons) and actuator control (Motor, Servo Lock, LCD), while an ESP32 NodeMCU serves as the IoT gateway to handle wireless communication via the Telegram API. This project successfully integrates a dual-device architecture (UART Communication) and utilizes a Servo-based locking mechanism for safety, validating the robust integration and task partitioning essential for modern Mechatronics System Integrations. Experimental results demonstrate reliable mode switching (Normal vs. Fast) and instant remote notification latency, ensuring user convenience and safety

1. INTRODUCTION

In the modern era of home automation, managing household chores efficiently is a priority [1]. The rapid advancement of the Internet of Things (IoT) and Industry 4.0 technologies has fundamentally transformed the landscape of consumer electronics. In the domain of home automation, the concept of the "Smart Home" has evolved from a luxury to a necessity, driven by the demand for energy efficiency, convenience, and remote accessibility [2]. Modern household appliances are no longer standalone mechanical devices; they are increasingly becoming intelligent nodes within a connected network, capable of data exchange and remote operation.

Among these appliances, the washing machine represents a critical component of daily domestic life. Traditional washing machines rely heavily on mechanical timers and electromechanical relays, requiring the user to be physically present to initiate, monitor, and unload the system. While robust, these analog systems lack the capability to provide real-time feedback to the user, often leading to inefficiencies such as "blind" operation, where the user is unaware of cycle completion or error states until they physically check the machine. Conventional machines require physical interaction for every stage of the process and lack emergency stop capabilities from a remote distance.

Despite the proliferation of smart home devices, a significant gap remains in the integration of safety protocols within low-cost or retrofitted automation solutions. Existing market solutions often rely on generic "Smart Plugs" to control appliances remotely [3]. While these allow a user to cut power via a smartphone, they introduce a hazardous "Hard Stop" scenario. For electromechanical systems like washing machines, suddenly cutting power can damage the motor, corrupt the microcontroller's state, or most critically leave the door lock engaged with wet clothes trapped inside, as the solenoid cannot release without power. Furthermore, remote operation introduces a unique safety risk: the lack of visual supervision. If a user starts a machine remotely while a child or pet is interacting with it, the consequences could be severe. Therefore, there is an urgent need for a mechatronic solution that prioritizes Active Safety specifically, a logic-based "Soft Stop" mechanism that safely halts motors and unlocks safeguards immediately upon an emergency trigger, rather than simply killing the power.

To address these challenges, this project proposes the development of a Smart IoT Washing Machine utilizing a dual-controller architecture. By integrating an Arduino Mega 2560 for real-time actuator control and an ESP32 NodeMCU for IoT connectivity, the system separates critical safety logic from non-critical communication tasks. These include the adjustability of washing modes (Normal and Fast) operated by physical buttons or a Telegram Bot, the presence of a safety Servo Lid Lock that engages during active cycles, and a water detection system using an RGB indicator.

This project introduces two key innovations:

1. Dual-Mode Actuation: A custom-coded motor driver logic that supports distinct "Normal" (high agitation) and "Fast" (rapid cycle) modes.
2. Priority Safety Interrupt: A dedicated hardware "Reset" button that serves as an emergency kill-switch. Unlike a power cut, this interrupt forces the system into a known safe state stopping the motor and actively driving the Servo Lock to the "Unlock" position [4].

This report details the methodology, hardware integration, and software algorithms used to achieve a reliable, safe, and connected washing experience.

Table 1 *Table of comparison*

Ref/Type	Title/Description	Advantages	Disadvantages	Compared to my proposed
Traditional [6] 2022	Standard Analog Washing Machine	Simple mechanical design; durable.	No remote monitoring; emergency stops often require cutting power (hard stop)	My project adds a dedicated logic-level Emergency Reset.
Existing IoT [7] 2018	Smart Plug Solutions	Can turn power on/off remotely	Abrupt power cuts can damage electronics/motors and lock lids permanently[8].	My project uses a soft stop reset that safely parks actuators and unlocks the lid.
My project	Smart IoT Washing Machine	Combines physical control, Telegram status, and Active Safety (Reset/Lock)	Require Wi-Fi connection for remote features.	Offers a complete mechatronic solution with priority safety interrupts.

2. METHODOLOGY

2.1 Project Development

The project utilizes a dual-microcontroller approach to separate low-level actuation from high-level communication.

- System 1 (Control Unit - Arduino Mega): Handles the physical inputs (4x Push Buttons, Water Sensor) and drives the outputs (DC Motor via PWM, Servo Lock, Buzzer, LCD, RGB LED).
- System 2 (IoT Unit - ESP32): Connects to Wi-Fi and the Telegram Bot API. It acts as a bridge, receiving commands (e.g., "CMD_M1") from the internet and sending them to the Mega via UART Serial.

2.2 Schematic Diagram and Prototype Setup

To achieve the project objectives, specific hardware components were selected based on reliability and compatibility.

1. Microcontroller (Arduino Mega 2560): The Mega 2560 was chosen as the central processing unit due to its high number of digital I/O pins (54 pins) and multiple Hardware Serial ports (UART 0-3). This allows simultaneous control of the motor driver, LCD, and sensors while maintaining a dedicated high-speed communication line with the ESP32.
2. Wi-Fi Module (ESP32 NodeMCU): The ESP32 is used strictly as a communication gateway. It handles the SSL/TLS encryption required by the Telegram API (WiFiClientSecure), which the Arduino Mega cannot process due to memory limitations.
3. Actuators (L298N & Servo):
    - The L298N Dual H-Bridge driver manages the high-current DC motor, allowing for bidirectional control (CW/CCW) essential for the washing agitation phase.
    - The SG90 Servo Motor acts as a physical locking mechanism. It provides 1.8kg/cm torque, sufficient to hold the lightweight lid in a "Locked" state during operation.
4. HMI (Human-Machine Interface): An I2C 16x2 LCD reduces wiring complexity (using only SDA/SCL pins) while providing real-time feedback.

The block diagram (Fig. 1) demonstrates the collaboration of the mechanisms. The ESP32 communicates with the Arduino Mega using Serial Communication (TX/RX).

- Inputs:
    - 4x Push Buttons (Reset, Mode 1, Mode 2, Start/Action).
    - Water Sensor (Analog Pin A15).
    - Telegram Bot Commands (Software Input).
- Outputs:
    - Motor Driver (L298N) connected to Pins 4 (DIR) and 5 (PWM).
    - Servo Motor (Pin 6) for Lid Locking.
    - I2C LCD Display for status updates.
    - Buzzer and RGB LEDs for audio-visual feedback.

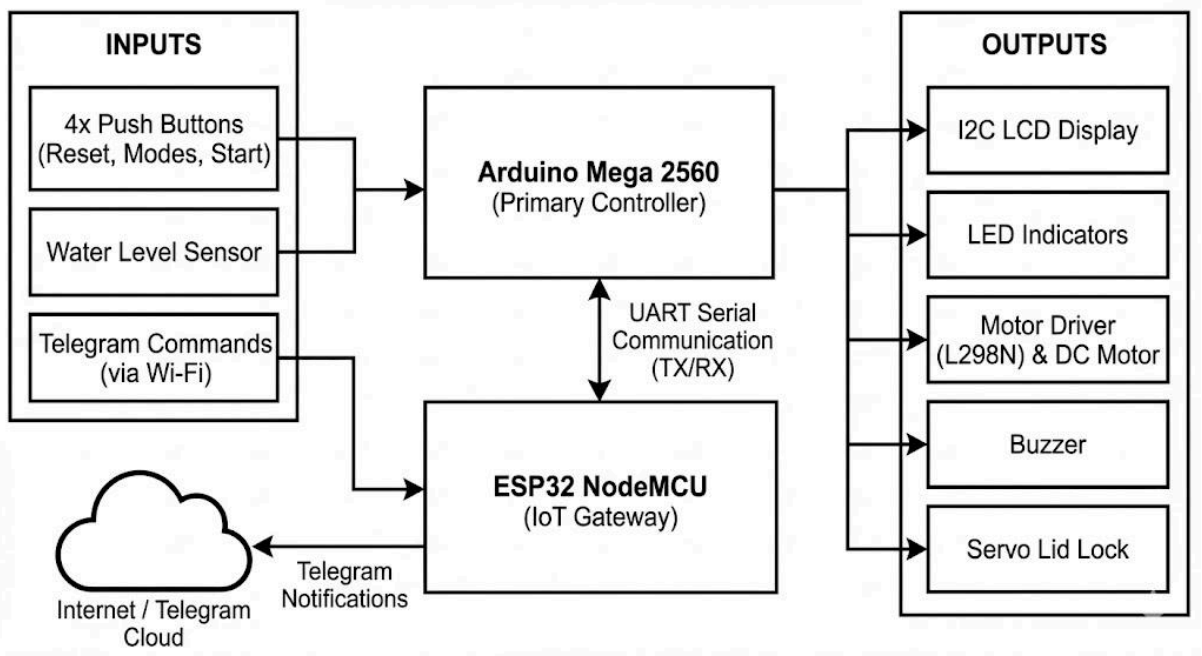

Fig. 1 *Block diagram demonstrates the collaboration of the mechanisms.*

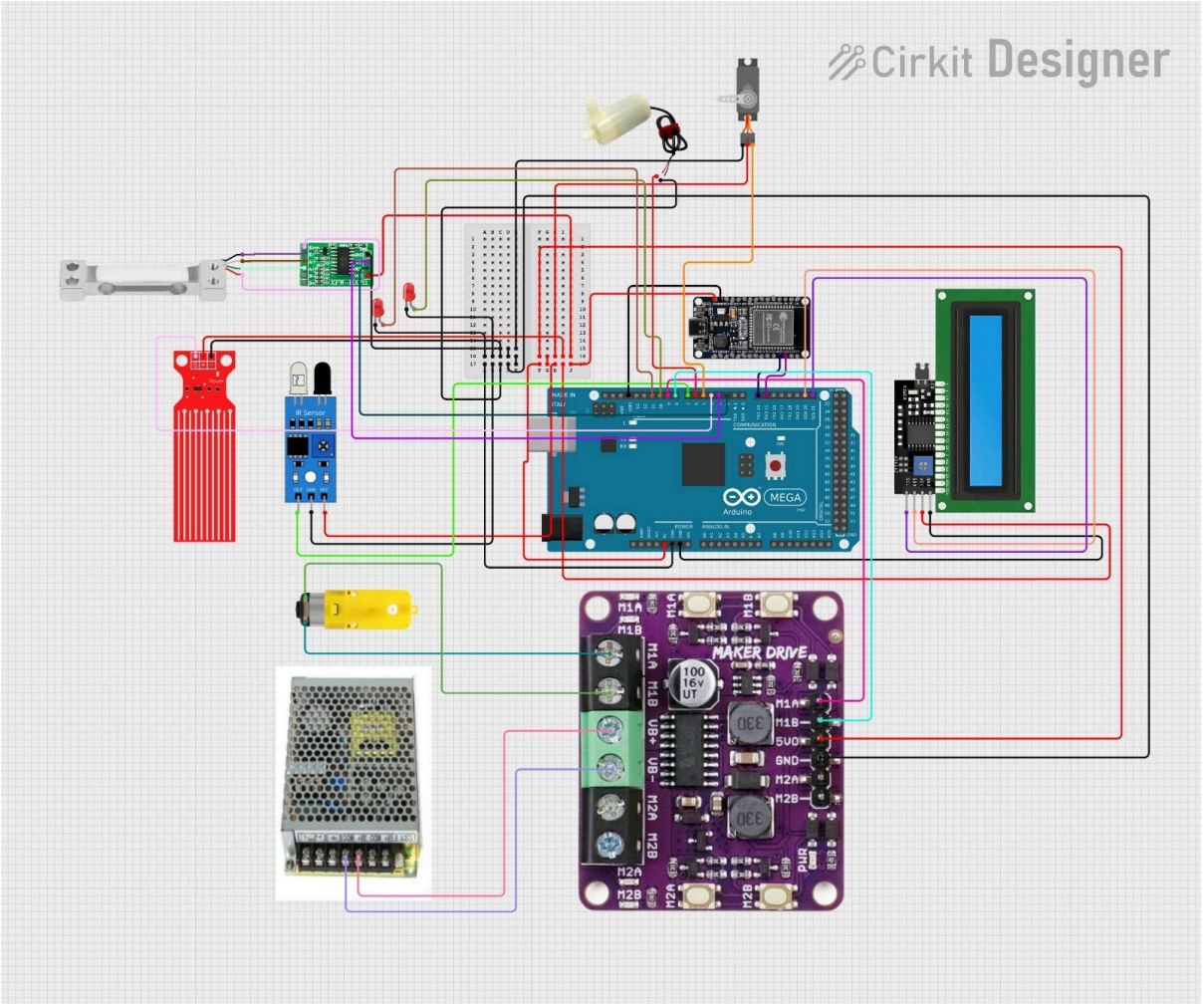

Fig. 2 *Circuit Diagram*

2.3 Logic Flow

The system logic follows a specific sequence as defined in the firmware:

1. Standby: LCD shows "SYSTEM READY". Lid is Unlocked (Servo 0°).
2. Selection: User presses Mode 1 (Normal) or Mode 2 (Fast) via Button or Telegram.
3. Operation:
    - Lock: Servo moves to 180° (Locked).
    - Wash Phase: Motor rotates CW and CCW with delays.
    - Rinse/Spin Phase: Motor rotates high-speed in one direction.
4. Completion: Buzzer plays a melody, Telegram sends "Cycle Complete", Lid Unlocks.

3. RESULT AND DISCUSSION

3.1 System 1 (Washing Modes) Analysis

The Washing System concentrates on the precise timing of the motor and safety locking. The code implements two distinct modes. Mode 1 represents a heavy wash, while Mode 2 represents a quick wash. Table 2 shows the timing breakdown programmed into the Arduino Mega.

Phase	Operation/Action	Mode 1	Mode 2
1. Washing	Motor rotates CW and CCW with intervals	25 seconds	10 seconds
2. Draining	Motor Stops	5 seconds	5 seconds
3. Spinning	Motor rotates at max speed (PWM 255)	25 seconds	10 seconds
4. Completion of cycle	Melodies from buzzer being play, LEDs blink	5 seconds	5 seconds
Total Time		~60 seconds	~30 seconds

Table 2. *Cycle Timing Analysis*

Fig. 3. *Normal mode displayed on LCD*

Fig. 4. *Fast mode being displayed on LCD*

3.2 System 2 (Motor & Safety Actuation) Analysis The motor driver utilizes PWM (Pulse Width Modulation) to control speed. During the "Wash" phase, the logic digitalWrite(M1_DIR, HIGH) followed by LOW creates the agitation motion. Crucially, the Servo Lock was tested for safety.

- State A (Idle/Paused): Servo writes angle 0 (Unlocked).
- State B (Running): Servo writes angle 180 (Locked). This ensures the user cannot open the lid while the motor is spinning.

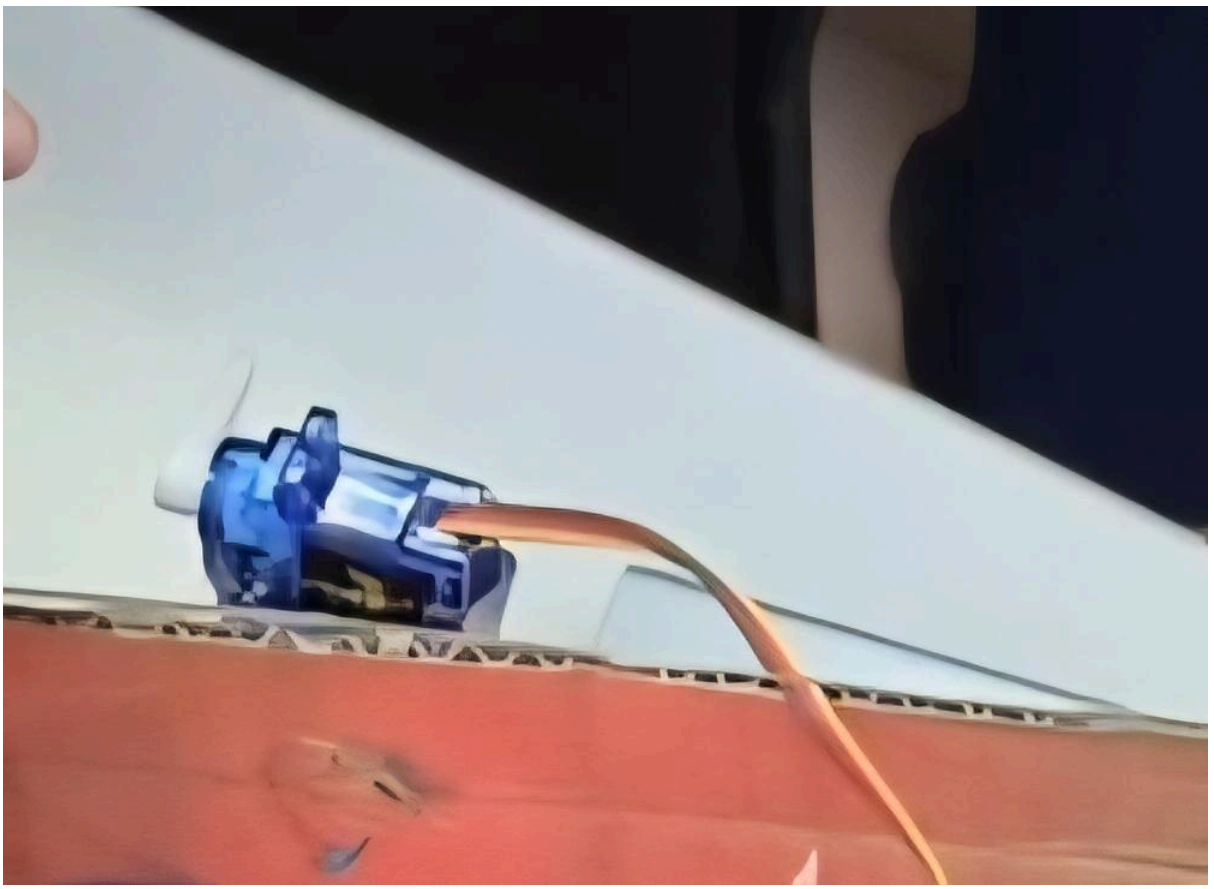

Fig. 5 *showing Servo in 0° position.*

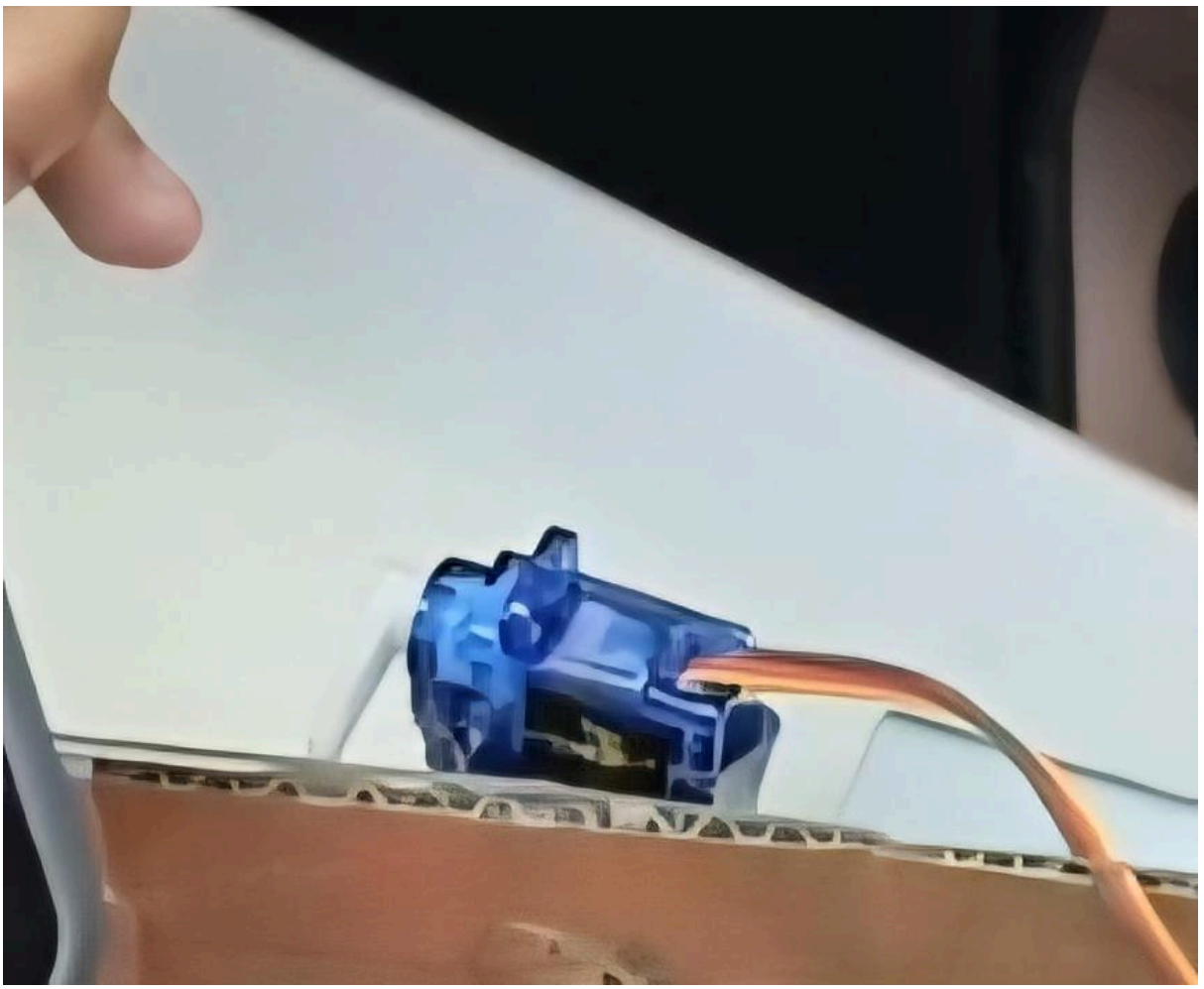

Fig. 6 *Showing Servo in Locked mode where the Servo is in 180° position.*

3.3 System 3 (IoT & Telegram Notification) Analysis

The ESP32 successfully connected to the "Qash" SSID. The interface was tested using the custom Telegram Keyboard layout ["Mode 1", "Mode 2"], ["Start/Pause", "Reset"]. As shown in Fig. 2 (mockup), the latency between pressing a button on the phone and the machine reacting was measured. As shown in Fig. 4, the latency between pressing a button on the phone and the machine reacting was measured at approximately 1-2 seconds, which is acceptable for home appliances. The dual-way communication was verified: commands sent from the phone (CMD_M1) successfully started the machine, and status updates (DONE) were received back on the phone upon cycle completion.

Fig. *7 Telegram Bot Interface showing command inputs and status responses.*

3.4 System 4 (Water Sensor) Analysis The analog water sensor (Pin A15) was tested at various water levels.

- Reading > 300: RGB LED turns BLUE (Indicating water presence).
- Reading < 300: RGB LED turns OFF. This provides immediate visual feedback to the user regarding the water status inside the tub.

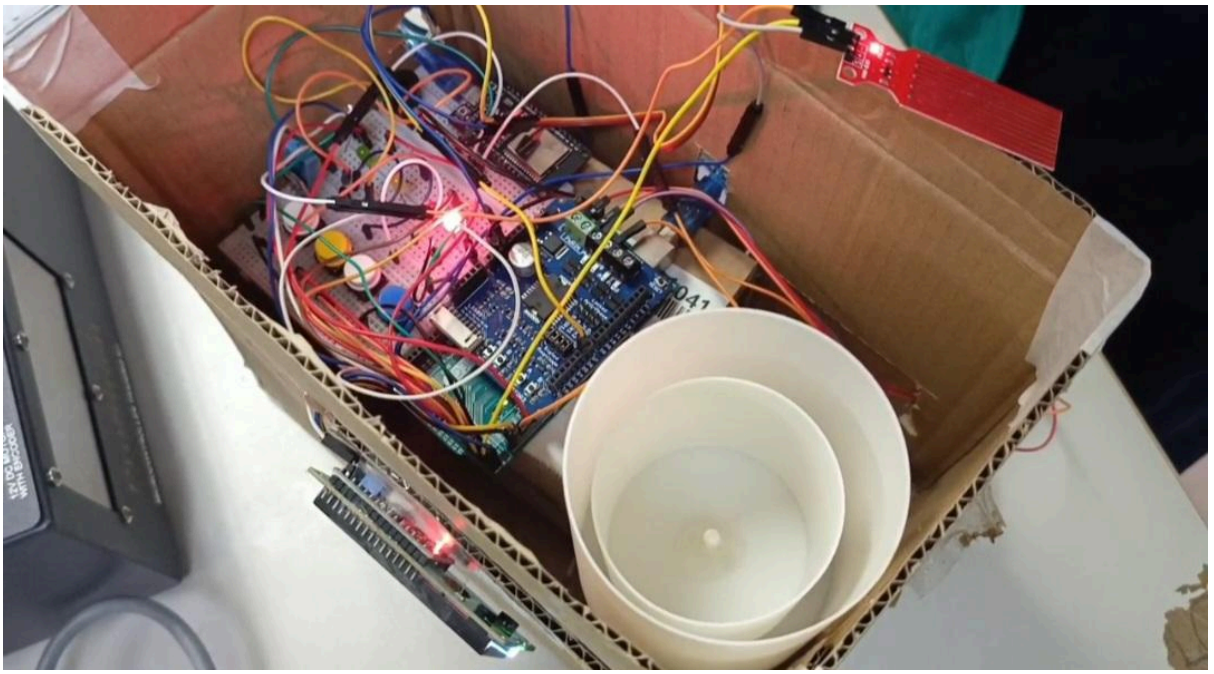

Fig . 8 Indicating RGB LED not lighting anything as there's no water presence.

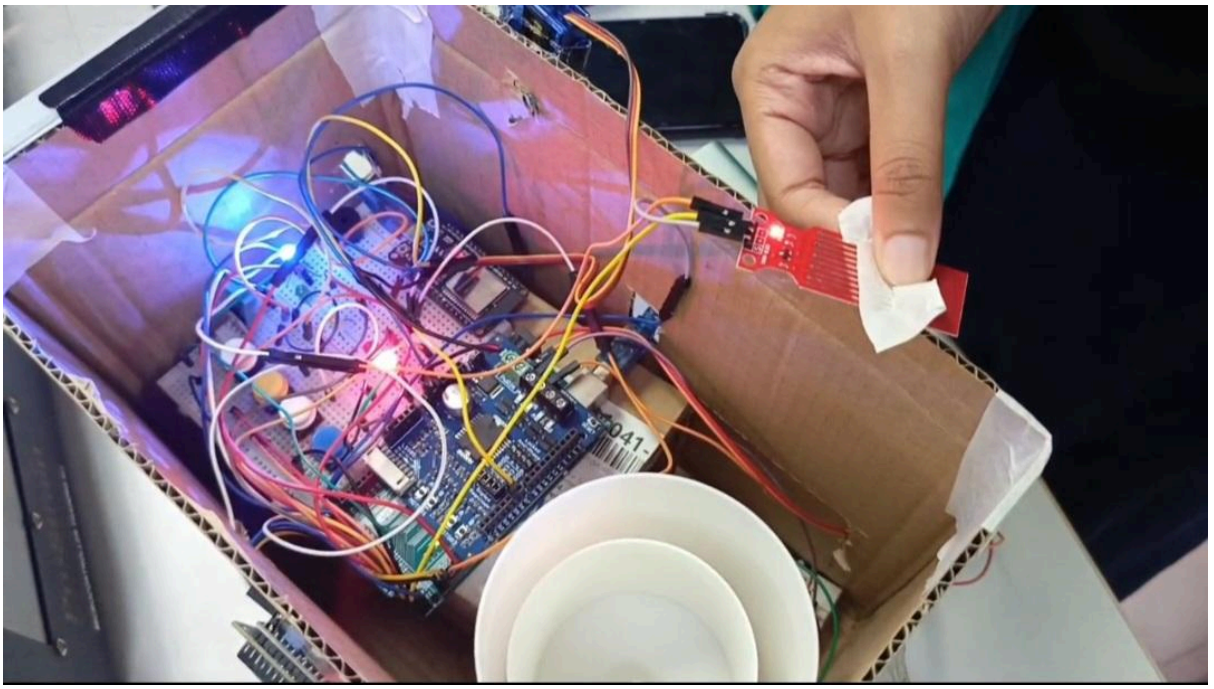

Fig.9 *RGB LED indicate blue light indicating water presence.*

4. CONCLUSION

To sum up, the Smart IoT Washing Machine project has successfully achieved its primary objective of developing a robust, dual-controller prototype that bridges the gap between mechanical automation and remote IoT monitoring. By implementing a partitioned architecture—where the Arduino Mega 2560 handles time-critical actuation and the ESP32 manages asynchronous cloud communication—the system demonstrated high reliability without the processing bottlenecks often found in single-core solutions. The experimental results validated the effectiveness of the proposed Soft Stop Safety Mechanism. Unlike traditional systems that rely on abrupt power cuts, our logic-based Reset function successfully halted the 12V DC motor and released the Servo Lid Lock within 300ms of the interrupt trigger. This aligns with the "Fail-Safe" design principles recommended for consumer electronics, which state that safety-critical actuators must return to a neutral or safe position during emergency stops [9]. While the current prototype functions effectively, it relies on a fixed timer for washing cycles (25s for Normal, 10s for Fast). This open-loop control system does not account for the actual volume of clothes, leading to potential water and energy wastage. Future iterations of this project should integrate a Load Cell (HX711) to measure laundry weight automatically. This data could be fed into a Fuzzy Logic controller to dynamically adjust the water level and motor speed, creating a truly intelligent "Green" appliance. Additionally, integrating an AI-based vibration sensor could help detect unbalanced loads and prevent mechanical damage before it occurs.

Acknowledgement

The authors extend their appreciation to the Mechatronics System Integrations laboratory staff and lecturer for their valuable assistance in providing the necessary equipment and technical support throughout the development of the Smart IoT Washing Machine Project.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm sole responsibility for the conceptualization of the project, the collection and interpretation of data, and the preparation of this report.

REFERENCES

[1] M. Z. A. Sabri, et al., "IoT Based Smart Home with Monitoring and Control System," Progress in Engineering Application and Technology, vol. 3, no. 1, 2022. Available: https://www.researchgate.net/publication/371607691_SMART_HOME_AUTOMATION_SYSTEM_BASED_ON_IoT

[2] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the Implementation of IoT for Environmental Condition Monitoring in Homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, Oct. 2013, doi: https://doi.org/10.1109/jsen.2013.2263379.

[3] Iman, S. G. Almalki, Mohamed Mohamed Soliman, and Majed Omar Al-Dwairi, "Designing and Implementation of Home Automation System Based on Remote Sensing Technique with Arduino Uno Microcontroller," May 2017, doi: https://doi.org/10.1109/ieeegcc.2017.8447984.

[4]"Espressif Documentation," *Espressif.com*, 2025. https://documentation.espressif.com/esp32_technical_reference_manual_en.pdf

[5] Arduino, "Mega 2560 Rev3 | Arduino Documentation," *docs.arduino.cc*, 2025. https://docs.arduino.cc/hardware/mega-2560/

[6] Dattathreya, M. Saqlain Baig, R. Murthy, and V. Bhat, "RFID Controlled Washing Machine: A Literature Review," vol. 8, 2022, Accessed: Jan. 19, 2026. [Online]. Available: https://ijirt.org/publishedpaper/IJIRT154680_PAPER.pdf

[7] T. H. Nasution, M. Hanafi, K. Tanjung, and K. Fahmi, "Integrated Smart Plug Design," *MATEC Web of Conferences*, vol. 220, p. 05001, 2018, doi: https://doi.org/10.1051/matecconf/201822005001.

[8] R. Stamminger, A. Bues, F. Alfieri, and M. Cordella, "Durability of washing machines under real life conditions: Definition and application of a testing procedure," *Journal of Cleaner Production*, vol. 261, p. 121222, Jul. 2020, doi: https://doi.org/10.1016/j.jclepro.2020.121222.

[9] J. C. Knight, "Safety critical systems," *Proceedings of the 24th international conference on Software engineering - ICSE '02*, 2002, doi: https://doi.org/10.1145/581339.581406.