MECHATRONICS SYSTEM INTEGRATION

MCTA 3203

MINI PROJECT:

WASHING MACHINE

SECTION 1

SEMESTER 2, 2024/2025

INSTRUCTOR:

ASSOC. PROF. EUR. ING. IR. TS. GS. INV. DR. ZULKIFLI BIN ZAINAL ABIDIN

DR. WAHJU SEDIONO

GROUP 9

NAME	MATRIC NO
AMER WAFDI BIN ZAINIZAM	2228703
AMALIN BALQIS BINTI RUYUSNI	2314684
FARHA QAISARA BINTI MOHD ZAHIDI	2315994

DATE OF SUBMISSION: 12 JUNE 2025

TABLE OF CONTENT **2**

1.0 COMPLIANCE TO PROJECT GUIDELINES

The mini project reflects a complete and functional embedded system design, incorporating essential elements of mechatronics, namely sensors, actuators, controllers, interfacing, and logical sequencing. The smart washing machine system was chosen as it effectively demonstrates the integration of multiple engineering domains including mechanical movement (motor control), electrical actuation (LEDs, buzzers), sensor signal processing (IR sensor and PixyCam), and digital interface (LCD, push buttons, and I2C communication).

The system is designed and implemented using three Arduino microcontrollers to reflect a distributed control approach. The master-slave architecture enhances modularity and reliability, which aligns with real-world industrial practices in embedded systems and automation. Each Arduino unit handles specific sub-tasks: the master oversees mode control, cycle execution, and LCD interface; one slave unit is dedicated to detecting white cloth using PixyCam; the other manages safety monitoring using an IR sensor and provides feedback via an LED.

In accordance with project criteria, the washing machine prototype supports:

- **Human-Machine Interaction:** Enabled through push buttons for power, mode selection, and cycle start/pause; and an LCD that continuously displays system status and wash phase.
- **Actuator Control:** A motor is activated according to wash phase logic using an H-bridge driver (L9110S), and a buzzer provides audible alerts.
- **Sensor Integration:** An IR sensor detects user presence and ensures the LED provides visual indication; a PixyCam detects white clothes that might need different treatment.

- **Cycle Sequencing:** The control logic enforces a clear wash cycle sequence—Wash → Rinse → Spin → Done—based on the selected mode.

Moreover, the system implements software features including interrupt-free polling logic, simple debounce handling with delays, and state transitions using enumerated types for clarity and expandability. By fulfilling both technical and educational objectives, the project meets and exceeds the expectations set out in the lab guidelines. The report also follows the documentation structure recommended, ensuring clarity in problem definition, design procedure, testing, results, and discussions.

In conclusion, the smart washing machine project not only adheres to the given project constraints and format but also serves as a robust platform for demonstrating advanced control strategies, modular design, and safety-focused automation, which are cornerstones of mechatronics system design.

2.0 DESIGN PROBLEM AND OBJECTIVES

Traditional washing machines, while effective in basic laundering tasks, often operate with limited intelligence and automation. These machines typically rely on manual input for every stage, from mode selection to wash duration and they lack the capacity to adapt based on the types of clothes or environmental conditions. For example, many machines cannot detect specific cloth types, such as delicate white fabrics that may require different handling or respond to the presence of a person nearby, which could be important for safety and interaction. As a result, users must be physically present to manage the washing process, increasing both user workload and the possibility of human error.

Furthermore, traditional designs often lack intuitive human-machine interaction. Visual indicators may be limited to a few LED lights or mechanical knobs, providing minimal feedback on the status of the washing cycle or errors. There is also typically no way for the machine to signal warnings, such as when inappropriate fabric is detected or if the user exceeds safe operational conditions.

This project addresses these limitations by implementing an intelligent and automated washing machine system that not only simplifies user interaction but also incorporates real-time monitoring and decision-making using sensors and microcontroller logic. The system is designed to autonomously manage wash modes and phases (Wash → Rinse → Spin → Done), detect human presence through an IR sensor, identify specific clothing items like white fabric using a PixyCam, and sound alerts using a buzzer when required. Through these enhancements, the system ensures improved safety, efficiency, and ease of use, elevating the washing machine from a basic appliance to a smart, sensor-based mechatronics solution.

The primary objective of this project is to design and implement a smart washing machine controller using microcontroller-based hardware, specifically Arduino Mega and Uno boards, to replicate the functionality of an intelligent laundry appliance. The project aims to go beyond basic motor control by incorporating intelligent decision-making, sensor feedback, and user interaction.

To achieve this, the system is built around multiple Arduino boards operating in a master-slave communication structure via the I2C protocol. The master controller is responsible for managing the main washing logic, which is selecting washing modes such as Cotton, Synthetic, Delicate, and Quick. While also handling user inputs via push buttons and displaying

system status and cycle phase on a 16x2 I2C LCD screen. This forms the main human-machine interaction interface.

A secondary microcontroller (slave) is tasked with handling safety and environment-based interactions. It uses an infrared (IR) sensor to detect nearby human presence and lights up an LED accordingly to indicate interaction readiness or possible interference with the machine's operation. Another slave board is dedicated to handling fabric-type detection through the PixyCam sensor. Specifically, it identifies white clothes that may require special treatment and triggers a buzzer alert to notify the user before proceeding with the wash cycle.

Additional components such as a load cell for weight sensing (future integration) and a motor driver for cycle-specific motor operation are also considered in the design. The motor runs during wash and spin phases and remains off during rinse and idle states. The entire system is developed to function cohesively, offering modular control, increased user safety, automated detection, and intuitive feedback, demonstrating the core principles of mechatronics in a real-world application.

3.0 DETAILS DESIGN DOCUMENTATION

The design and development of the smart washing machine project required careful selection and integration of both hardware and software components. This section outlines the technical composition of the system, detailing how each part contributes to the functionality of the overall setup. The project is built around modular microcontroller units, each handling a dedicated task, and is structured using a master-slave communication approach to reflect real-world industrial control systems.

3.1 HARDWARE COMPONENTS

These are the hardware components we used in the washing machine:

- Arduino Mega 2560
- 2 Arduino Uno
- PixyCam
- Passive Buzzer
- IR Sensor
- LED
- L9110S Driver
- LCD Display
- 3 Push Buttons
- Breadboard
- Jumper

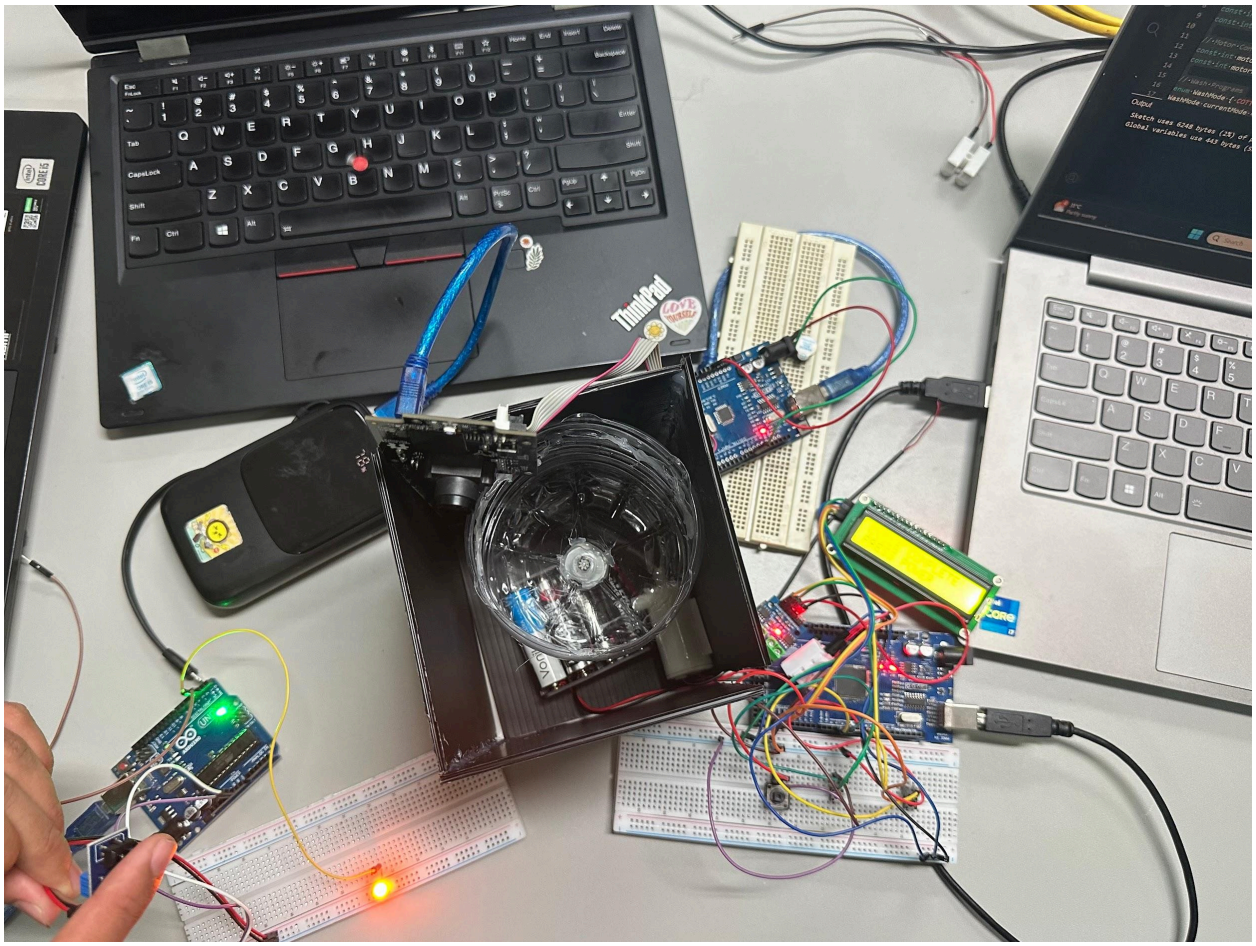

Figure 1: Hardware Setup

The hardware implementation of the smart washing machine system is based on a distributed microcontroller design using one Arduino Mega and two Arduino Uno boards. The Arduino Mega serves as the master controller, orchestrating the user interface, wash cycle logic, motor control, and communication with the slave devices. This board was selected for its increased number of digital and analog I/O pins and its higher memory capacity, which makes it ideal for managing multiple tasks concurrently.

The first Arduino Uno functions as a slave device dedicated to white cloth detection using a PixyCam. The PixyCam is an intelligent vision sensor capable of detecting color-coded signatures. In this setup, it is programmed to recognize white clothing (designated as signature 1). When such cloth is detected, the Uno activates a passive buzzer connected to it, producing an audible warning to indicate that a potentially delicate or inappropriate fabric has been placed inside the machine.

The second Arduino Uno acts as another slave controller, responsible for environmental awareness using an infrared (IR) sensor. This sensor is configured to detect human presence near the washing machine. Upon detection, an LED is turned on to indicate that someone is nearby, which can be useful for user feedback or to pause dangerous operations. This sensor also communicates detection status back to the master via the I2C protocol.

The LCD display used in this project is a 16x2 character display with an I2C interface, allowing for reduced wiring complexity. It displays current wash mode (Cotton, Synthetic, Delicate, or Quick), cycle phase (Wash, Rinse, Spin, Done), machine state (OFF, STANDBY, RUNNING, PAUSED, COMPLETE), and remaining time. It is the main user interface for real-time system monitoring.

To control the mechanical aspect of the washing operation, a DC motor is used to simulate the washing drum rotation. The motor is driven by an L9110S motor driver module, which is controlled by the Arduino Mega to switch the motor ON or OFF based on the current phase (only runs during WASH and SPIN). Basic push buttons are connected to the Mega for power control, mode selection, and start/pause functionality, allowing direct user interaction. These buttons are configured using internal pull-up resistors to simplify circuit design and improve reliability.

In summary, the system utilizes a rich combination of components, such as controllers, sensors, actuators, and displays, that are logically divided across three microcontrollers to achieve modularity, robustness, and ease of future scalability.

3.2 SOFTWARE ARCHITECTURE

The software architecture of the project is implemented across three distinct but interconnected Arduino programs, each responsible for specific functionalities. These programs are designed to work in coordination via I2C communication and real-time data sharing, following a modular software design approach.

Code 1, running on the first Arduino Uno slave, handles the PixyCam and buzzer module. The program initializes communication with the PixyCam sensor and continuously monitors for image blocks with a designated color signature (in this case, signature 1 representing white fabric). When such a block is detected, a boolean flag is set to trigger an audible warning using the buzzer connected to pin 9. The buzzer remains on as long as white fabric is detected, providing a safety alert to the user to remove or handle the item carefully. If no white fabric is present, the buzzer remains off, and normal operations can proceed.

Code 2, the main and most complex program, runs on the Arduino Mega master controller. This code is responsible for managing the washing machine's core states, which include OFF, STANDBY, RUNNING, PAUSED, and COMPLETE. Based on user input via push buttons, the program allows users to power on the machine, select wash modes, and start or pause the washing cycle. Once the cycle begins, the software automatically progresses through the predefined wash phases: WASH (50% of total time), RINSE (30%), SPIN (20%), and DONE. Each phase has a time duration associated with it, depending on the selected mode. This program also handles motor control, activating the motor during appropriate phases and updates the LCD display with current status and timing information. The use of enums and structured functions improves readability and scalability of the code.

Code 3, running on the second Arduino Uno slave, manages the IR sensor and LED module. The program reads the digital input from the IR sensor connected to pin 9 and, if human presence is detected, turns on an LED connected to pin 8. Additionally, this board acts as an I2C slave with address 8 and responds to requests from the master Arduino by sending the current IR sensor status. This allows the master to receive environment data and react accordingly if future expansions are made to pause operation for safety during human interaction.

All three pieces of code are optimized for responsiveness and non-blocking behavior using short delays and millis()-based timing where needed. Communication is kept efficient and simple, with the master only requesting minimal data from slaves and acting accordingly. This modular approach ensures that each Arduino handles a specific subsystem independently while contributing to the complete, intelligent behavior of the washing machine system.

4.0 IMPLEMENTATION OF CHAPTERS

The implementation phase of this smart washing machine project focused on building a fully functional prototype using embedded systems concepts and standalone Arduino modules. The project comprised three separate Arduino boards, which are one Arduino Mega and two Arduino Uno units, each programmed and operated independently due to the challenges faced during attempted integration. While the original plan included implementing master-slave communication using I2C or UART protocols to synchronize these microcontrollers, the integration attempt was unsuccessful due to communication errors and synchronization issues. Consequently, the design was revised to allow each Arduino to function autonomously, handling its own specific responsibilities within the system.

The Arduino Mega acted as the primary interface for the user. It was connected to a 16x2 I2C LCD and a set of push buttons that allowed users to power on the machine, select wash modes, and start or pause washing cycles. The LCD provided real-time visual feedback, including the current washing mode (such as Cotton, Synthetic, Delicate, or Quick), machine state (OFF, STANDBY, RUNNING, PAUSED, COMPLETE), and the active phase in the cycle (WASH, RINSE, SPIN, DONE). The Mega also controlled a DC motor via an L9110S motor driver, managing agitation during washing and spinning based on the selected mode and internal timing. The button inputs were handled with debounce logic to ensure accurate and reliable operation, preventing unintentional multiple triggers due to mechanical bounce.

The second component of the system involved a PixyCam module, which was connected to a standalone Arduino Uno. This board was responsible solely for object detection, specifically identifying white-colored fabrics using color signature tracking (signature 1). If the PixyCam

detected a white cloth in the field of view, possibly indicating delicate material, the Uno board triggered a passive buzzer to sound a warning. This function operated completely independently from the main control unit, providing an added safety and quality-assurance feature to alert users before starting a cycle that could potentially damage certain clothing.

A third Arduino Uno was tasked with handling an infrared (IR) proximity sensor and an LED indicator. The IR sensor was used to detect the presence of a person near the washing machine. When proximity was detected, the LED lit up as a simple visual alert. Although it did not directly influence the machine's operation, this feature enhanced the system's awareness of human presence and could be extended in future versions to include automatic safety responses, such as pausing motor activity during spin cycles if someone approaches.

Unlike fully integrated systems that use master-slave communication to synchronize multiple microcontrollers, this project took a modular and decentralized approach due to technical constraints. Each board was programmed using the Arduino IDE, and testing was done for each module in isolation to ensure individual reliability. While no inter-board data sharing occurred, the individual functionalities were effectively demonstrated and operated in parallel, contributing to the overall goal of the project.

Additional features such as a load cell and potentiometer were planned for weight measurement and simulation of timing or water level inputs. Although these components were not fully implemented in the final integrated prototype, initial tests showed that they could be used in future improvements. The potentiometer, when connected, allowed analog input to simulate variable wash durations and was useful for testing LCD updates. Similarly, weight

detection using the load cell was intended to alert the user if the laundry exceeded 1 kilogram, which could strain the motor.

In conclusion, although full hardware-software integration between boards was not achieved, the project successfully demonstrated key embedded system capabilities across separate modules. Each Arduino performed its assigned task reliably—whether controlling the washing cycle, monitoring for fabric types, or detecting human presence. This modular, standalone design provided flexibility, fault isolation, and ease of debugging, forming a solid foundation for future iterations of the smart washing machine with improved communication and system-level integration.

5.0 DISCUSSION

The development of the smart washing machine project provided an integrated way in which embedded systems concepts could be applied within an experiential and real-world environment. While working on the project, various challenges and learning experiences were achieved that affected the final destiny of the system. The initial plan was to develop a modular system in which multiple Arduino boards would interact using master-slave protocols such as I2C or UART. This exchange was intended to enable master control over all functions from sensor inputs, cycle control, and response from the actuators on one master board. But despite extended attempts, fail-safe integration of the Arduino Mega and the two Arduino Unos proved a failure due to synchronization errors and data transmission irregularities. This failure led to a fundamental design alteration in which every board was independently redesigned to function on its own, with its own sensors and actuators, without any interference from the other boards.

This independent-board approach, while not ideal for a fully integrated smart system, proved to be a practical and effective workaround. Each microcontroller successfully executed its respective task: the Arduino Mega managed user interaction, LCD feedback, motor control, and button responses; one Arduino Uno operated the PixyCam and triggered a buzzer alert upon detecting white cloth; the other Uno handled the IR sensor and illuminated an LED when human presence was detected. By isolating the functionality into distinct, independently functioning units, the team was able to demonstrate all major components of the system, even if they did not interact with each other in real time. This modular design also facilitated simpler debugging and more stable operation when testing, since failures in the logic of one board did not compromise other boards' function.

Another standout of the project was the real-time control and feedback loop offered by the Arduino Mega. The use of push buttons for selecting modes and controlling machine states closely mimics commercial washing machines, while the LCD provides clear user guidance throughout the wash cycle. The implementation of software debouncing for the buttons ensured more stable operation, reducing false triggers. Additionally, the cycle progression logic (Wash → Rinse → Spin → Done) effectively simulated the behavior of a real washing system, with motor activity mapped to appropriate phases. This showed a strong understanding of finite state machines and timing control in embedded systems.

The PixyCam integration added a smart feature that elevated the safety and efficiency of the prototype. Detecting white clothing before a cycle begins allows users to avoid damaging delicate fabrics or misusing machine settings. Though it functioned independently from the washing cycle logic, its real-time detection and alert via a passive buzzer highlighted the potential for intelligent fabric recognition in laundry systems. Similarly, the IR sensor and LED

module, while simple, contributed an additional layer of interaction—showing that the machine could sense environmental changes or human proximity, which could be used in future iterations to implement automatic safety responses.

In addition to the implemented modules, our team explored integrating advanced features such as weight detection and IoT connectivity. A load cell sensor with the HX711 module was initially planned to measure the weight of laundry in the drum, which would enable overload detection if the weight exceeded a threshold (1 kg). However, the HX711 module consistently failed to initialize and could not be detected by the microcontroller. After verifying the wiring and testing alternative code, we concluded that the module was likely faulty. As a result, the weight detection feature could not be realized in the final prototype.

We also attempted to develop an IoT interface using MIT App Inventor in combination with an ESP8266 WiFi module. The goal was to allow remote control of the washing machine, such as starting/stopping cycles and monitoring mode status via a mobile application. Even though the app interface was designed well, the ESP8266 module failed to establish stable serial communication with Arduino Mega. Repair efforts like baud rate adjustments, voltage level checks, and firmware resets were made, but no response was detected in the module. Ultimately, this feature was excluded from the final build due to hardware failure and time limitations.

A limitation of the project was the inability to combine these separate systems into a single, interconnected unit due to failed communication integration. This meant that although all functionalities were demonstrated, they operated in parallel rather than in a cohesive, centralized control system. Despite this, the project successfully met its key objectives and presented a strong foundation for further improvements. In future developments, efforts can be focused on

overcoming communication barriers, possibly by improving I2C address management, handling clock synchronization better, or exploring alternative communication protocols like SPI or wireless methods.

In summary, while the lack of integration presented a setback, the project was still able to effectively demonstrate the key features of a smart washing machine through a modular design. The lessons learned through attempted integration, individual module testing, and user interaction design have been invaluable in deepening the understanding of embedded systems engineering. The project stands as a functional and demonstrative prototype that successfully simulates smart automation in household appliances.

6.0 SAFETY

Safety plays a crucial role in the design and operation of any household appliance, especially one that involves electrical systems, mechanical movements, and user interaction, such as a washing machine. In this smart washing machine project, safety was considered in both the physical design and the control logic to prevent common hazards while promoting a secure user experience.

One of the core safety concerns in traditional and modern washing machines is the risk of injury from moving parts such as the rotating drum or motor. In commercial machines, if the lid is opened during a spin cycle, there is a danger of hands or clothing getting caught. While our prototype does not yet implement an automatic shutdown mechanism upon detecting human presence, we integrated an infrared (IR) sensor to detect motion or proximity. When someone is

near, an LED lights up, serving as a visual indicator that someone is close to the machine. This feature highlights the potential for future upgrades, such as pausing the motor automatically when human presence is detected, thereby improving safety.

Another common issue in washing machines is overloading, which can result in motor strain, belt damage, and excessive vibrations. To address this, our design included a load cell sensor to estimate the weight of the clothes in the drum. If the weight exceeds 1 kilogram, an alert system (such as a buzzer or display warning) can notify the user. Although the load cell was not fully functional in the final implementation, the system was structured to support such a feature for protecting internal components and ensuring efficient washing cycles.

Electrical hazards are also a major concern in any appliance using microcontrollers and external power supplies. Improper connections or voltage mismatches can lead to short circuits, overheating, or even electrical fires. In this project, all wiring was done using insulated jumper wires, and components like the Arduino Mega, Arduino Unos, motor drivers, and sensors were operated within their safe voltage and current ratings. The motor driver L9110S was chosen in part because of its built-in protection mechanisms, including current limiting and thermal shutdown features, to protect both the circuit and the user.

Another safety aspect considered was fabric damage and wash cycle compatibility. Using a PixyCam sensor, the system could detect white cloths, often associated with delicate fabrics. When a white item is detected, the system triggers a buzzer alert to warn the user. This can help prevent delicate items from being included in inappropriate wash cycles, reducing the risk of damage to the clothing.

User interface safety was also addressed through a button interface with software debounce logic. Without this logic, mechanical button bounces could cause unintended multiple state transitions, which might confuse users or even damage the system if certain operations were triggered repeatedly. By stabilizing input handling, the system ensured a safer and more predictable user experience.

Finally, all critical information about machine status, such as the current wash mode, phase (WASH, RINSE, SPIN), and operational state (OFF, STANDBY, RUNNING, COMPLETE) was clearly displayed on a 16x2 I2C LCD screen. This real-time feedback allowed users to interact with the machine more confidently and avoid interrupting active cycles, which could otherwise cause mechanical stress or data errors in the control system.

To sum up, while this project remains a prototype, it incorporates several important safety features and highlights common safety issues that would be critical in a fully operational smart washing machine. The IR sensor for human detection, buzzer alerts for delicate fabric recognition, load monitoring for overcapacity, and clear LCD feedback all contribute to a safer and smarter washing system. Future iterations could expand these features by adding emergency stop mechanisms, lid interlock sensors, waterproof casing, and enhanced overload protection to meet real-world safety standards.

7.0 CONCLUSION AND FURTHER WORKS

The smart washing machine project successfully demonstrates how embedded systems and sensor integration can transform traditional appliances into intelligent, user-friendly machines. Through the use of Arduino Mega and multiple Arduino Uno boards, the system effectively manages core washing functionalities such as cycle phase control, wash mode selection, user input via buttons, and dynamic status display through an LCD interface. The inclusion of PixyCam for white cloth detection and the IR sensor for human presence adds layers of automation and safety, making the system more adaptive and user-aware.

Although the project initially aimed for master-slave communication among the Arduinos, this integration was not fully achieved due to time constraints and technical challenges. Instead, each Arduino operated independently to handle its designated tasks, which still resulted in a functional and modular system. Despite this limitation, the prototype met many of its primary objectives, including real-time display updates, audible and visual alerts, motor control based on cycle phases, and user interaction through push buttons. These outcomes collectively demonstrate a practical implementation of mechatronic principles.

This project also highlighted key challenges, such as the complexity of inter-board communication and synchronization, particularly when multiple sensors and actuators are involved. The experience underlined the importance of thorough debugging and robust wiring practices, especially when dealing with multiple microcontrollers in a single system.

Looking ahead, several improvements and extensions can be pursued. Most important is to be able to properly integrate all the Arduino units via a proper master-slave communication protocol, either UART, I2C, or SPI. This would facilitate centralized control and data exchange

and minimize the architecture of the system. Second, the capacity of the load cell sensor can be used to its maximum to provide precise weight measurement and prevent overloading.

Other potential enhancements include the inclusion of a lid safety switch to automatically halt the cycle when the door is left ajar, the inclusion of waterproof enclosures for critical electronics, and the facilitation of the user interface with touch screen or voice command entry. Integration with mobile apps for remote monitoring and operation could also be undertaken, further enhancing the system's sophistication and convenience.

Lastly, the smart washing machine prototype is a sound basis for future enhancement. It illustrates the potential of embedded systems in creating smart, interactive, and secure appliances and provides hands-on learning experience in coding, circuit design, and system integration. With further refinement and advanced features, the project can grow into a working smart appliance waiting to be put into practical use.

APPENDICES