MINI PROJECT REPORT:
SMART WASHING MACHINE SYSTEM

GROUP 5

MCTA 3203

SEMESTER 1 2024/2025

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION: 16 JANUARY 2025

NO	NAME	MATRIC NUMBER
1.	MUHAMMAD AFIQ ADHAM BIN MOHD NADZRI	2227531
2.	MUHAMMAD AMIN BIN MOHAMAD RIZAL	2217535
3.	MUHAMMAD AFIQ BIN MOHD ASRI	2212541
4.	MUHAMMAD AKMAL BIN MOHD FAUZI	2214077
5.	MUHAMMAD AFIQ IKHWAN BIN NOR SHAHRIZAL	2215897

TABLE OF CONTENTS

INTRODUCTION

This project involves designing and developing a smart washing machine system prototype that integrates multiple sensors, actuators, and controllers to automate and enhance the washing process. The system utilizes a combination of Arduino Mega and Arduino Uno microcontrollers, alongside several components such as relays, motor drivers, ultrasonic sensors, and water level sensors. This project aims to create a functional prototype that demonstrates the capabilities of an intelligent washing machine with user-friendly features, safety measures, and efficient water management.

DESIGN PROBLEM AND OBJECTIVE

The main problem addressed by this project is the lack of automation and user interaction in traditional washing machines. This prototype aims to:

1. Automate water inflow and drainage using relays and pumps.
2. Detect user presence using an ultrasonic sensor to improve user interaction.
3. Ensure safety by using an infrared sensor to detect if the lid is open and triggering a buzzer as an alert.
4. Stop water inflow automatically when the desired water level is reached using a water level sensor.
5. Provide a user-friendly interface with push buttons and an LCD to select water temperature and start washing.
6. Integrate Arduino Mega and Uno microcontrollers for seamless communication and control of components.

The project also aimed to explore the integration of additional features such as Bluetooth connectivity, RFID authentication, and object detection using a Pixy camera.

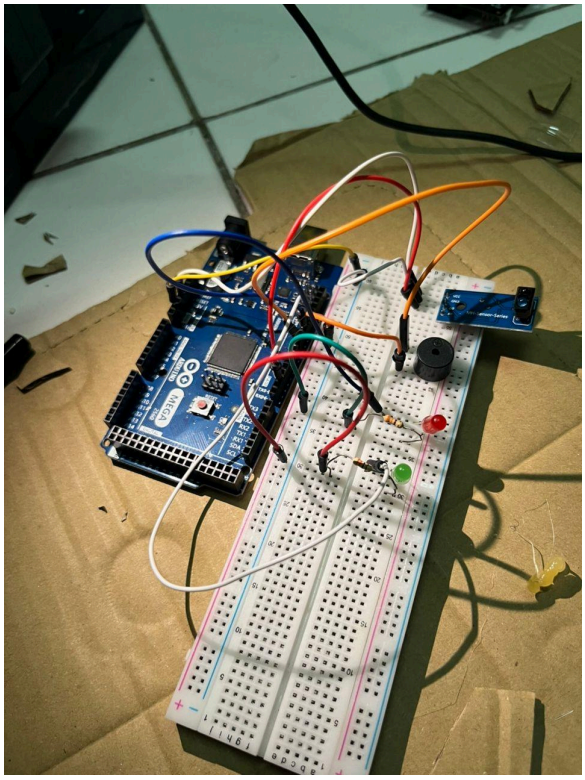
MATERIALS AND EQUIPMENT

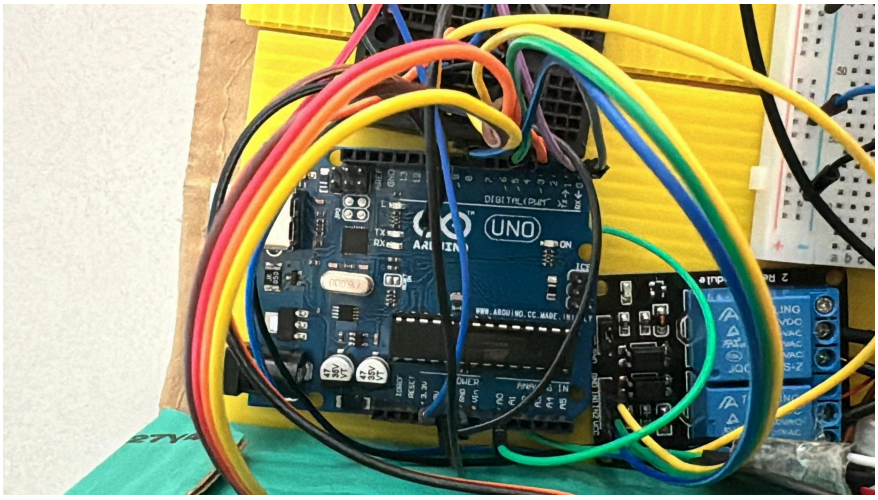
Our smart washing machine system consists of the following components:

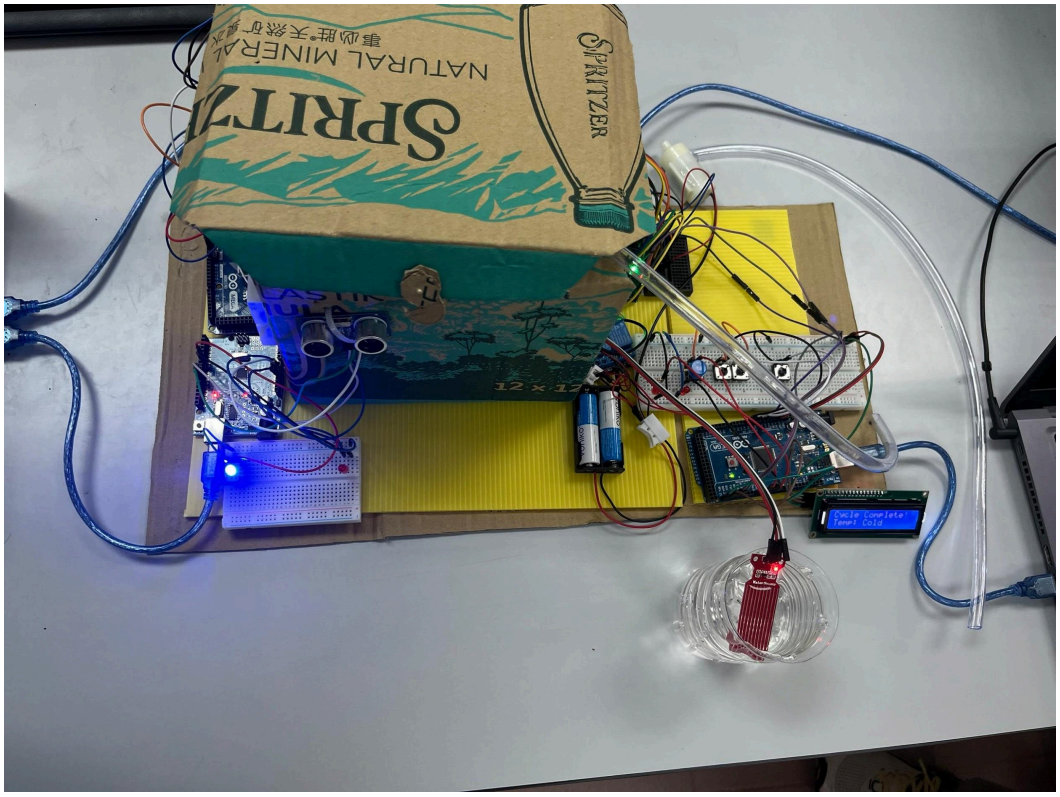
1. Microcontrollers:
    - Arduino Mega (main controller, sender).
    - Arduino Uno (receiver, integrated with Mega).
    - Two additional microcontrollers for demonstration purposes, are not integrated into the main system.
2. Sensors and Actuators:
    - Ultrasonic Sensor: Detects user presence to activate the washing machine.
    - Infrared Sensor: Monitors the lid's status; triggers a buzzer if the lid is open during operation.
    - Water Level Sensor (red): Measures the water level in the drum to stop the inflow pump automatically.
    - Relay Module (2-channel): Controls two pumps for water inflow and drainage.
    - Motor Driver (L9110): Drives a DC motor to rotate the drum.
    - Buzzer: Alerts the user if the lid is open.
3. User Interface:
    - Push the buttons to select water temperature and start the washing process.
    - LCD to show user selections and washing status.
4. Communication:
    - Arduino Mega acts as the sender, while Arduino Uno acts as the receiver.
5. Power Supply:

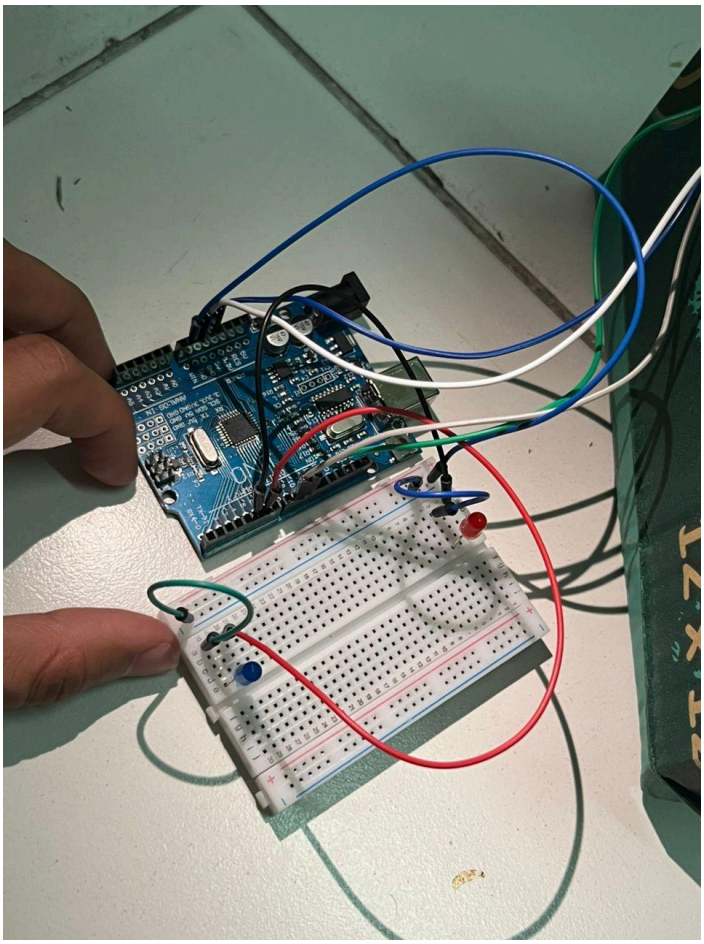
- The components are powered using an external power source, a battery for the pump, and a relay connection. Proper grounding and connections were ensured to avoid malfunctions.

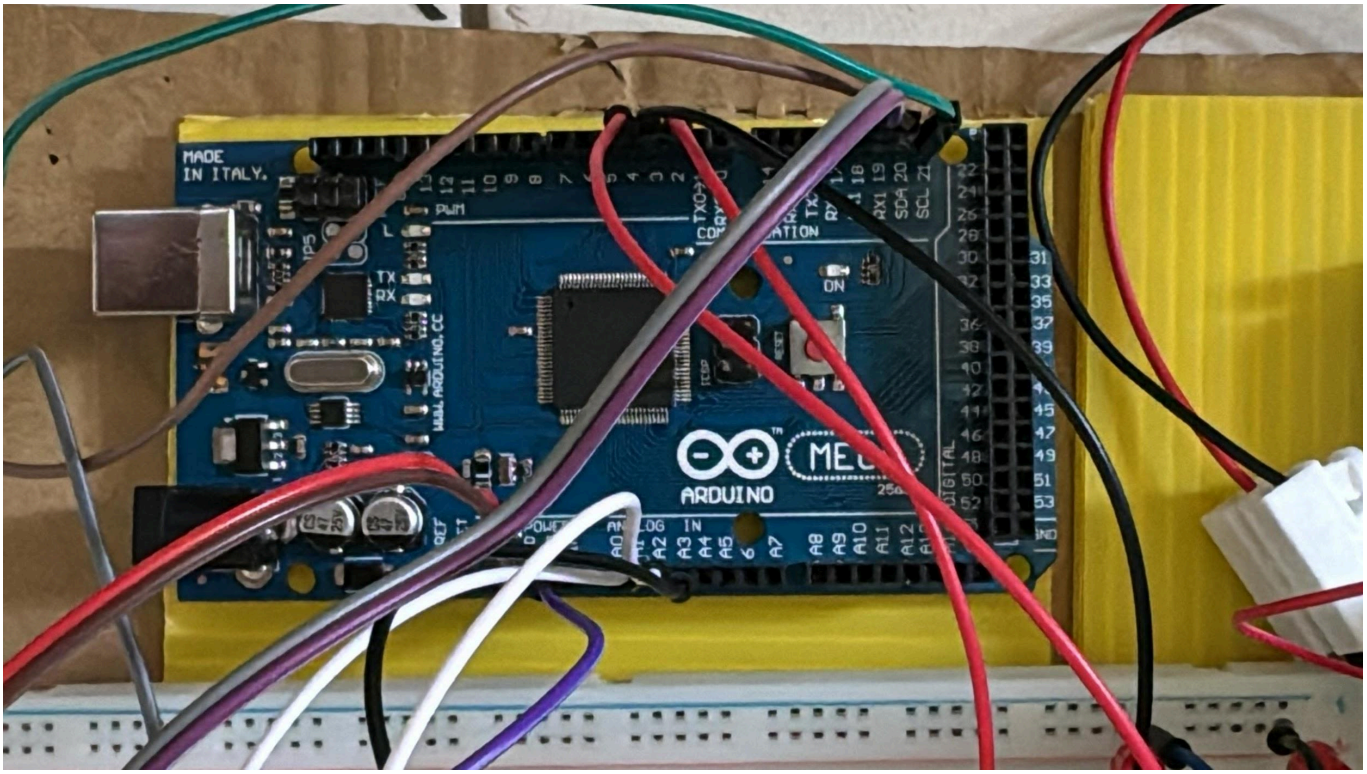
PROJECT SETUP






NATURAL MINERAL
SPRITZER
Cycle Complete
Temp: Cold




MADE
IN ITALY.
PWM
TX
RX
ON
ARDUINO

IMPLEMENTATION OF CHAPTER

1. Relay Module and Pump Control

The first step was to implement the relay module to control two pumps: one for water inflow and the other for drainage. The relay module was connected to the Arduino Mega, and initial tests were conducted to ensure that the relays could switch ON and OFF the pumps correctly. Each pump was tested individually to verify its operation, with the default state being OFF when the system powers on. The relays were programmed to activate based on the water level sensor and user inputs. This setup ensured that water inflow and drainage could be automated without manual intervention.

2. Motor Driver and Drum Rotation

The motor driver (L9110) was integrated to control the rotation of the washing drum. A DC motor was connected to the driver, and its speed and direction were controlled via the Arduino Mega. The motor was programmed to alternate between clockwise and counterclockwise rotations to simulate the washing action. Testing involved ensuring that the motor driver could handle the required load and operate smoothly without overheating. This feature was synchronized with the water inflow and drainage process to ensure proper washing cycles.

3. Ultrasonic Sensor for User Detection

The ultrasonic sensor was installed to detect the presence of a user near the washing machine. This sensor sends ultrasonic waves and measures the time taken for the waves to return, determining if someone is standing close to the machine. If a user is detected, the system becomes active and ready for operation. This feature adds an interactive aspect to the machine, ensuring that it

only activates when needed. The sensor was calibrated to avoid false triggers due to nearby objects.

4. Infrared Sensor and Safety Mechanism

An infrared sensor was used to detect whether the washing machine lid was open or closed. If the lid is open during operation, a buzzer is triggered to alert the user. This safety mechanism ensures that the washing process does not start or continue unless the lid is securely closed. The sensor was tested under various conditions to ensure reliability, and the buzzer was programmed to sound continuously until the lid was closed.

5. Water Level Sensor for Automation

A water level sensor was installed to monitor the water level in the drum. The sensor ensures that the inflow pump stops once the desired water level is reached, preventing overflows. The sensor was calibrated to provide accurate readings, and its output was linked to the relay controlling the inflow pump. Testing involved filling the drum to different levels to verify that the sensor could detect and stop the pump at the correct threshold.

6. Push Buttons and LCD Display

Push buttons and an LCD were integrated to allow users to select the water temperature and start the washing process. The buttons were programmed to send inputs to the Arduino Mega, which then displays the selected options on the LCD. The display also shows the washing status, such as "Filling Water," "Washing," or "Draining." This interface provides a simple and user-friendly way for users to interact with the machine. The LCD and buttons were tested to ensure proper communication with the microcontroller.

7. Communication Between Arduino Mega and Uno

The Arduino Mega and Uno were set up as the sender and receiver, respectively. Serial communication was established between the two microcontrollers to share data and coordinate operations. For instance, the Arduino Mega controls the overall process, while the Uno handles specific tasks such as managing the LCD and push buttons. Testing involved ensuring that data could be sent and received accurately without delays or errors.

8. Assembly and Integration

Once all components were individually tested, they were assembled into a single system. The wiring was carefully organized to avoid interference and ensure proper connections. The components were integrated with the Arduino Mega as the central controller. Debugging was conducted to resolve any issues with sensor readings, relay switching, or motor control. The final assembly was tested for functionality, ensuring that all parts worked together seamlessly.

CODING AND CONNECTION

Code for Sender (Arduino Mega)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// LCD setup
LiquidCrystal_I2C LCD(0x27, 16, 2);

// Button pins
const int hotButtonPin = A1;
const int mediumButtonPin = A2;
const int coldButtonPin = A3;
const int startButtonPin = 2;

// LED pins
const int washingLEDPin = 3;
const int rinsingLEDPin = 4;

// Variables
String selectedTemp = "";
int washTime = 0;
int rinseTime = 10; // Default rinsing time

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("Select Temp...");

  pinMode(hotButtonPin, INPUT_PULLUP);
  pinMode(mediumButtonPin, INPUT_PULLUP);
  pinMode(coldButtonPin, INPUT_PULLUP);
  pinMode(startButtonPin, INPUT_PULLUP);

  pinMode(washingLEDPin, OUTPUT);
  pinMode(rinsingLEDPin, OUTPUT);

  Serial1.begin(9600); // Communication with Arduino Uno
  Serial.begin(9600);  // Debugging
}

void loop() {
```

```
 // Temperature selection
 if (digitalRead(hotButtonPin) == LOW) {
   selected temp = "Hot";
  wash timee = 20; // Set wash time for hot
   delay(500);
 } else if (digitalRead(mediumButtonPin) == LOW) {
  selected tempp = "Medium";
   wash time = 15; // Set wash time for medium
   delay(500);
 } else if (digitalRead(coldButtonPin) == LOW) {
  selected tempp = "Cold";
  wash timee = 10; // Set wash time for cold
   delay(500);
 }

 // Display selected temperature
LCDd.setCursor(0, 1);
 LCD.print("Temp: ");
 LCD.print(selected temp);
 LCD.print("    ");

 // Send wash and rinse times to Uno
 if (digitalRead(start button in) == LOW) {
   Serial.println("Sending wash and rinse times to Uno...");
   Serial1.print(wash time); // Send wash time
   delay(5000);
   Serial1.print(",");       // Separator
   Serial1.println(rinseTime); // Send rinse time
   delay(1000);

   // Washing Process
   lcd.clear();
   lcd.print("Washing...");
   digitalWrite(washingLEDPin, HIGH); // Turn on washing LED

   // Countdown for washing
   for (int i = washTime; i > 0; i--) {
     lcd.setCursor(0, 1);
     lcd.print("Time Left: ");
     lcd.print(i);
     lcd.print("s    ");
     delay(1000);
   }
```

```
    digitalWrite(washingLEDPin, LOW); // Turn off washing LED

    // Rinsing Process
    lcd.clear();
    lcd.print("Rinsing...");
    digitalWrite(rinsingLEDPin, HIGH); // Turn on rinsing LED

    // 5-second delay between washing and rinsing
    delay(5000);  // 5-second delay before starting the rinse cycle

    for (int i = rinseTime; i > 0; i--) {
      lcd.setCursor(0, 1);
      lcd.print("Time Left: ");
      lcd.print(i);
      lcd.print("s   ");
      delay(1000);
    }
    digitalWrite(rinsingLEDPin, LOW); // Turn off rinsing LED

    // End of cycle
    lcd.clear();
    lcd.print("Cycle Complete!");
    delay(2000);
  }
}
```

Code for Receiver (Arduino Uno)

```
#include "Arduino.h"
#include "HX711.h"

// Pin definitions for load cell
#define DOUT 2  // Data pin for HX711
#define CLK  3  // Clock pin for HX711

// Pin definitions for relays (pumps)
const int relay1 = 7; // Pump 1 relay pin
```

```
const int relay2 = 8; // Pump 2 relay pin

// Pin definitions for Motor B
#define MOTOR_PIN_B1 9  // B-IB (PWM for speed control)
#define MOTOR_PIN_B2 10 // B-IA (direction control)

// Pin definitions for water level sensor
const int waterLevelPin = A0; // Analog pin for water level sensor

// Variables
HX711 scale;
int receivedWashTime = 0;
int receivedRinseTime = 0;
bool washStarted = false;
bool waterLevelReached = false;
int waterLevelThreshold = 500;  // Threshold for 85% water level (to be
calibrated)

void setup() {
  Serial.begin(9600);   // Communication with Arduino Mega

  // Initialize pins
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(MOTOR_PIN_B1, OUTPUT);
  pinMode(MOTOR_PIN_B2, OUTPUT);

  // Turn off all outputs initially
  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);
  digitalWrite(MOTOR_PIN_B1, LOW);
  digitalWrite(MOTOR_PIN_B2, LOW);

  // Initialize HX711
  scale.begin(DOUT, CLK);
  scale.set_scale(600);  // Example scale factor
  scale.tare();

  Serial.println("Receiver ready. Waiting for wash and rinse
times...");
}

void loop() {
```

```
  // Check if data is available from the Mega
  if (Serial.available() > 0) {
    String receivedData = Serial.readStringUntil('\n'); // Read the
incoming data
    int commaIndex = receivedData.indexOf(','); // Find the separator
    receivedWashTime = receivedData.substring(0, commaIndex).toInt();
// Parse wash time
    receivedRinseTime = receivedData.substring(0, commaIndex).toInt();
// Parse rinse time
    washStarted = true; // Mark the washing cycle to start
    Serial.print("Received Wash Time: ");
    Serial.println(receivedWashTime);
    Serial.print("Received Rinse Time: ");
    Serial.println(receivedRinseTime);
  }

  if (washStarted) {
    // Start the process
    startProcess(receivedWashTime, receivedRinseTime);
    washStarted = false; // Reset flag after cycle completion
  }
}

void startProcess(int washTime, int rinseTime) {
  // Step 1: Measure weight with the load cell
  float weight = scale.get_units(10); // Get weight, average over 10
readings
  Serial.print("Weight Measured: ");
  Serial.println(weight);

  // Check if the weight is greater than 100g
  if (weight > -20) {
    Serial.println("Weight is sufficient, proceeding with pump
activation...");

    // Step 2: Activate Pump 1 and start filling water
    Serial.println("Activating Pump 1...");
    digitalWrite(relay1, LOW); // Turn on Pump 1

    // Wait until water level reaches 85%
    while (analogRead(waterLevelPin) < waterLevelThreshold) {
      int waterLevel = analogRead(waterLevelPin);
      Serial.print("Current Water Level: ");
```

```
      Serial.println(waterLevel);
      delay(1000); // Check every second
    }

    digitalWrite(relay1, HIGH); // Turn off Pump 1 once water level
reaches 85%
    Serial.println("Water Level Reached 85%. Proceeding to
washing...");

    // Step 3: Start washing
    washCycle(washTime, rinseTime); // Start washing and rinsing cycle
  } else {
    Serial.println("Weight is insufficient, process will not start.");
  }
}

void washCycle(int washTime, int rinseTime) {
  Serial.println("Starting Wash Cycle...");
  // Activate Pump 1 (Water Intake)
  digitalWrite(relay1, HIGH); // Water intake pump (if necessary)
  digitalWrite(relay2, LOW);
  delay(5000); // Allow pump to run for 5 seconds
  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);

  // Start Motor (Washing)
  Serial.println("Motor Washing...");
  for (int i = washTime; i > 0; i--) {
    Serial.print("Washing Time Left: ");
    Serial.print(i);
    Serial.println("s");

    // Motor forward
    analogWrite(MOTOR_PIN_B1, 255); // Speed control
    digitalWrite(MOTOR_PIN_B2, LOW);
    delay(1000);
  }

  // Stop motor after washing
  digitalWrite(MOTOR_PIN_B1, LOW);
  digitalWrite(MOTOR_PIN_B2, LOW);
  Serial.println("Wash Cycle Completed.");
```

```
  // 5-second delay before starting the rinse cycle
  Serial.println("Waiting for 5 seconds before rinsing...");
  delay(5000); // 5-second delay between wash and rinse

  // Start Motor (Rinsing)
  Serial.println("Motor Rinsing...");
  for (int i = rinseTime; i > 0; i--) {
    Serial.print("Rinsing Time Left: ");
    Serial.print(i);
    Serial.println("s");

    // Motor reverse
    analogWrite(MOTOR_PIN_B2, 255); // Speed control
    digitalWrite(MOTOR_PIN_B1, LOW);
    delay(1000);
  }

  // Stop motor after rinsing
  digitalWrite(MOTOR_PIN_B1, LOW);
  digitalWrite(MOTOR_PIN_B2, LOW);
  Serial.println("Rinse Cycle Completed.");

  // Activate Pump 2 (Water Drain)
  Serial.println("Activating Pump 2...");
  digitalWrite(relay2, HIGH); // Run pump for draining
  digitalWrite(relay1, LOW);
  delay(5000); // Run pump for 5 seconds
  digitalWrite(relay2, HIGH);
  digitalWrite(relay1, HIGH);

  Serial.println("Cycle Complete.");
}
```

DISCUSSION

The washing machine prototype was developed to automate key processes such as water inflow, drainage, drum rotation, and user interaction while incorporating safety features. Overall, the project ran well, with most components functioning as intended. The integration of hardware and software demonstrated the potential of microcontrollers to handle complex systems. However, there were areas where the project fell short of its original plan, which impacted the overall performance and functionality.

Project Successes

The core functionalities of the washing machine worked effectively. The relay module successfully controlled the pumps for water inflow and drainage, with the water level sensor accurately detecting the desired water level to stop the inflow pump. The motor driver enabled smooth drum rotation, simulating the washing process by alternating between clockwise and counterclockwise directions. The ultrasonic sensor reliably detected user presence, activating the system only when someone was near, which added an interactive and energy-efficient aspect to the design. Additionally, the infrared sensor and buzzer combination ensured safety by preventing the machine from operating if the lid was open. These features demonstrated the feasibility of building a functional washing machine prototype using Arduino microcontrollers.

Challenges and Limitations

Despite the successes, some challenges prevented the project from fully meeting its initial goals. The **Bluetooth module** was planned to enable communication with a mobile app for additional control and monitoring, but it was not implemented due to software compatibility issues. This feature could have

enhanced the user experience by allowing remote operation, but the limitations in the code and testing time made it impractical to include.

The **RFID module**, which was intended to enhance user authentication, faced issues with precision and functionality. The RFID reader failed to detect tags reliably, which was likely due to calibration errors or hardware limitations. As a result, this feature was excluded from the final prototype.

The **Pixy camera**, which was supposed to assist in visual recognition tasks, was also omitted. The primary issues were related to its software library and compatibility with the Arduino system, as well as hardware problems with its connection cables. This component's exclusion meant that the project lacked advanced automation features that could have further distinguished it.

Performance Rating

Based on the original project plan, we rated the performance of our prototype **7/10**. While the core functionalities ran well and met the primary objectives, the missing components reduced the system's overall capability. The excluded features—Bluetooth, RFID, and Pixy camera—would have added significant value to the project by enhancing user interaction, security, and automation.

Safety

Safety measures were incorporated into the design to prevent accidents and ensure reliable operation. The infrared sensor ensures the lid is closed before starting the washing process, and the buzzer alerts the user if the lid is left open. Proper grounding of all electrical components was ensured to avoid short circuits or electrical hazards.

CONCLUSION

The washing machine prototype successfully demonstrated key functionalities such as water inflow and drainage, drum rotation, lid safety, and user interaction. Components like the relay module, motor driver, ultrasonic sensor, and water level sensor operated as intended, showcasing the potential of Arduino-based systems in automating everyday tasks. Despite certain limitations, such as the exclusion of the Bluetooth module, RFID, and Pixy camera due to software and hardware challenges, the project achieved most of its objectives and provided valuable insights into embedded systems design.

RECOMMENDATION

Future improvements include integrating the Bluetooth module for remote control, enhancing the reliability of the RFID system, and resolving software issues to utilize the Pixy camera for advanced automation. Adding water heating functionality, improving system robustness, and conducting comprehensive testing would further refine the prototype. These enhancements would transform the system into a more efficient, user-friendly, and marketable solution.

ACKNOWLEDGEMENTS

A special thanks goes out to Dr. Wahju Sediono and Dr. Zulkifli Bin Zainal Abidin, my teaching assistant, and my peers for their invaluable help and support in finishing this report. Their advice, feedback, and experience have greatly influenced the level of quality and understanding of this work. Their time, patience, and commitment to supporting my academic success are greatly appreciated.

STUDENT'S DECLARATION

Certificate of Originality and Authenticity

This is to certify that we are **responsible** for the work submitted in this report, that the original work is our own except as specified in the references and acknowledgment, and that the original work contained herein has not been untaken or done by unspecified sources or persons.

We hereby certify that this report has **not been done by only one individual** and **all of us have contributed to the report**. The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have **read** and **understand** the content of the total report and that no further improvement on the reports is needed from any of the individual contributors to the report.

We, therefore, agreed unanimously that this report shall be submitted for **marking** and this **final printed report** has been **verified by us**.

Signature:
Name: MUHAMMAD AFIQ ADHAM BIN MOHD NADZRI
Matric Number: 2227531

Read ☑
Understand ☑
Agree ☑

Signature:
Name: MUHAMMAD AMIN BIN MOHAMAD RIZAL
Matric Number: 2217535

Read ☑
Understand ☑
Agree ☑

Signature: afiq
Name: MUHAMMAD AFIQ BIN MOHD ASRI
Matric Number: 2212541

Read ☑
Understand ☑
Agree ☑

Signature:
Name: MUHAMMAD AKMAL BIN MOHD FAUZI
Matric Number: 2214077

Read ☑
Understand ☑
Agree ☑

Signature: Afiq
Name: MUHAMMAD AFIQ IKHWAN BIN NOR SHAHRIZAL
Matric Number: 2215897

Read ☑
Understand ☑
Agree ☑