LABORATORY REPORT

Mini Project: Washing Machine

MCTA 3203

GROUP H

PROGRAMME: MECHATRONICS ENGINEERING

GROUP MEMBERS:	MATRIC NO:
1. SYEDA SAMIA	2123536
2. ATHIRAH HAZIQAH BINTI BADERULHISHAM	2210240
3. SITI HEIDI AMIRA BINTI AZRUL	2111970

DATE OF SUBMISSION: Thursday, 13th June 2024

ABSTRACT

This mini project explores the possibility of automating a traditional washing machine using an Arduino microcontroller.The goal is to replace the original control unit with a low-cost, user-programmable system based on Arduino. The report details the design, development, and implementation of this automated washing machine system.This report aims to contribute to the growing field of DIY (Do-It-Yourself) appliance automation and demonstrate the potential of Arduino for controlling complex systems like washing machines.

Table of Contents

INTRODUCTION

This project explores the automation of a traditional washing machine using an Arduino microcontroller, aimed at replacing the original control unit with a versatile, user-programmable system. Washing machines are essential household appliances that have traditionally relied on fixed control mechanisms, limiting customization and adaptability. By integrating Arduino, an open-source electronics platform known for its flexibility and ease of use, this project aims to demonstrate how DIY (Do-It-Yourself) appliance automation can revolutionize home appliances.

Arduino enables the creation of intelligent systems capable of reading sensor inputs and controlling various outputs such as motors, displays, and actuators. This capability allows for the development of a washing machine system that can be tailored to specific user needs, enhancing functionality and user experience. The project utilizes components like RFID sensors, servo motors, LCD displays, and ultrasonic sensors to automate washing and drying cycles based on load size and user inputs.

The hypothesis driving this experiment is that an Arduino-based control system can effectively replace conventional washing machine controls, offering a cost-effective and adaptable solution. The objectives include designing a robust hardware setup, developing intuitive software logic for cycle automation, and implementing safety features such as door locking mechanisms and error detection systems.

Through this exploration, we aim to contribute to the field of DIY appliance automation by providing insights into the design, development, and implementation of Arduino-based washing machine automation. By leveraging Arduino's capabilities, this project demonstrates the potential for enhancing household appliances with customizable, programmable solutions that cater to modern consumer needs.

PROCEDURE

<u>Materials And Equipment:</u>

- 1 x Arduino Mega
- 1 x USB cable
- 50 x Jumper wires
- 1 x resistor
- 1 x Tap module
- 1 x 9g micro servo motor
- 1 x RGB SMD
- 1 x Rotary encoder module
- 1 x Pushbutton
- 2 x Breadboard
- 1 x RFID RC522 RF IC card sensor module
- 1 x RFID tag
- 1 x Ultrasonic sensor HC-SR04
- 1 x Active piezo buzzer
- 1 x 4x4 matrix keypad

- 1 x LCD display screen 16x2 I2C

Experimental Setup:

1. Connect the GND and 5V pins of the Arduino to the breadboard for use for the components.
2. Connection of RFID sensor to Arduino:
    a. 3.3V → 3.3V on Arduino
    b. RST pin → D9
    c. GND → GND
    d. MISO → D50
    e. MOSI → D51
    f. SCK → D52
    g. SDA → D10
3. Connection of tap module sensor to Arduino:
    a. SIG → A0
    b. Middle pin → 5V
    c. Negative side pin → GND
4. Connection of RGB SMD sensor to Arduino:
    a. Negative pin → GND
    b. R pin → D13
    c. G pin → D12
    d. B pin → D11
5. Connection of rotary encoder to Arduino:
    a. GND → GND
    b. + pin → 5V
    c. CLK → D19
    d. DT → D18
    e. SW → D6
6. Connection of pushbutton to Arduino:
    a. Pin 1 → GND
    b. Pin 2 → D5
7. Connection of piezo buzzer to Arduino:
    a. - pin → GND
    b. SIG → D22
8. Connection of ultrasonic sensor to Arduino:
    a. VCC → 5V
    b. Trig → D3
    c. Echo → D2
    d. GND → GND
9. Connection of 4x4 matrix keypad to Arduino:
    a. R1 → D37
    b. R2 → D36
    c. R3 → D35
    d. R4 → D34
    e. C1 → D33
    f. C2 → D32

    g. C3 → D31
    h. C4 → D30
10. Connection of servo motor to Arduino:
    a. Brown pin → GND
    b. Red pin → 5V
    c. Yellow pin → D8
11. Connection of LCD display to Arduino:
    a. VCC → 5V
    b. GND → GND
    c. SDA → D20
    d. SCL → D21
12. Connect the Arduino to the PC using the USB cable.

Below are schematics of the circuit which were used to verify the results and for simulation purposes.

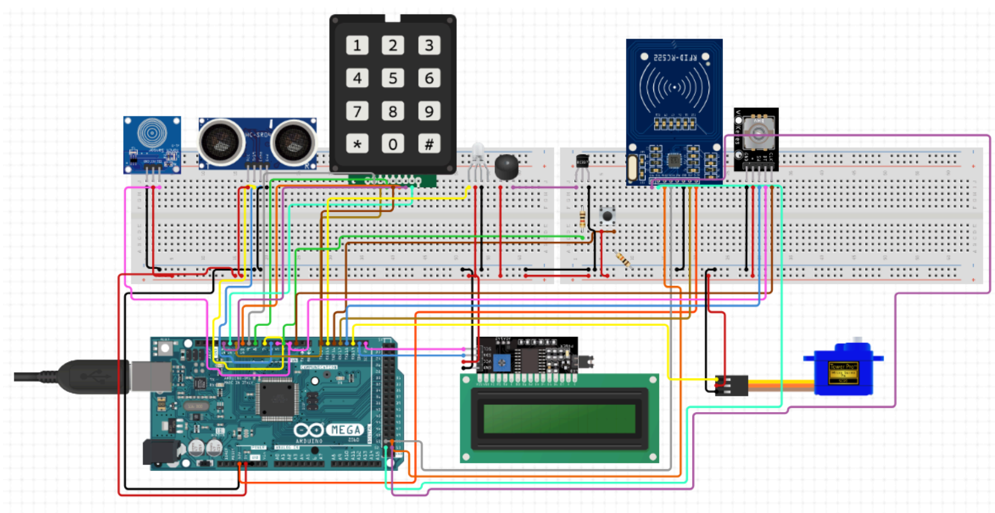

Fig. 1. Components connected to the Arduino

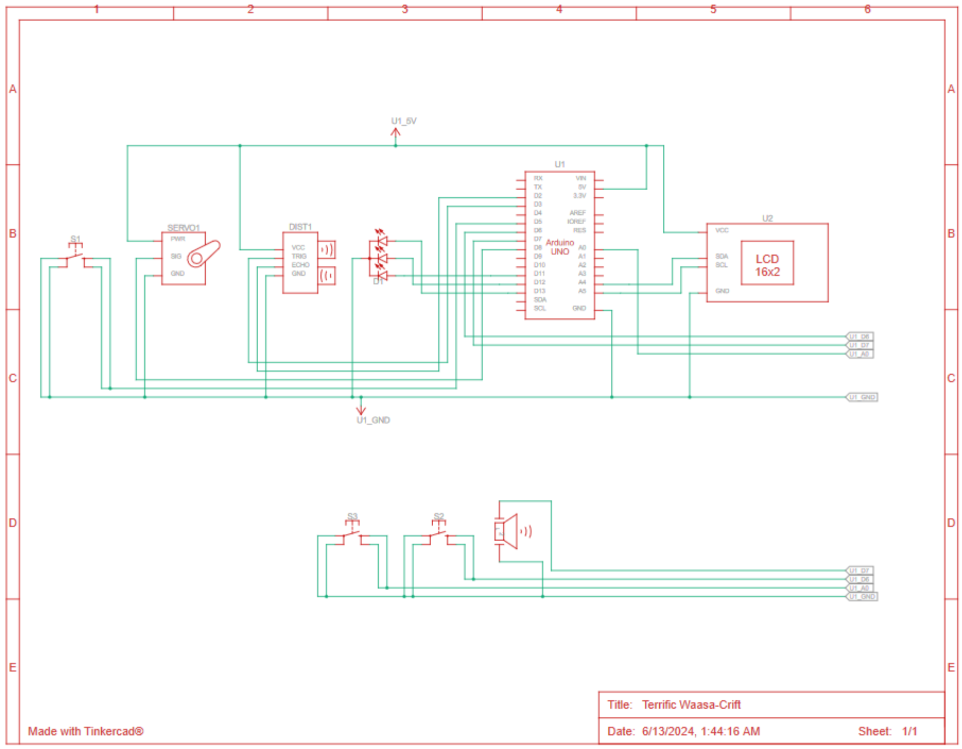

Fig. 2. Circuit diagram excluding RFID, keypad and rotary encoder.

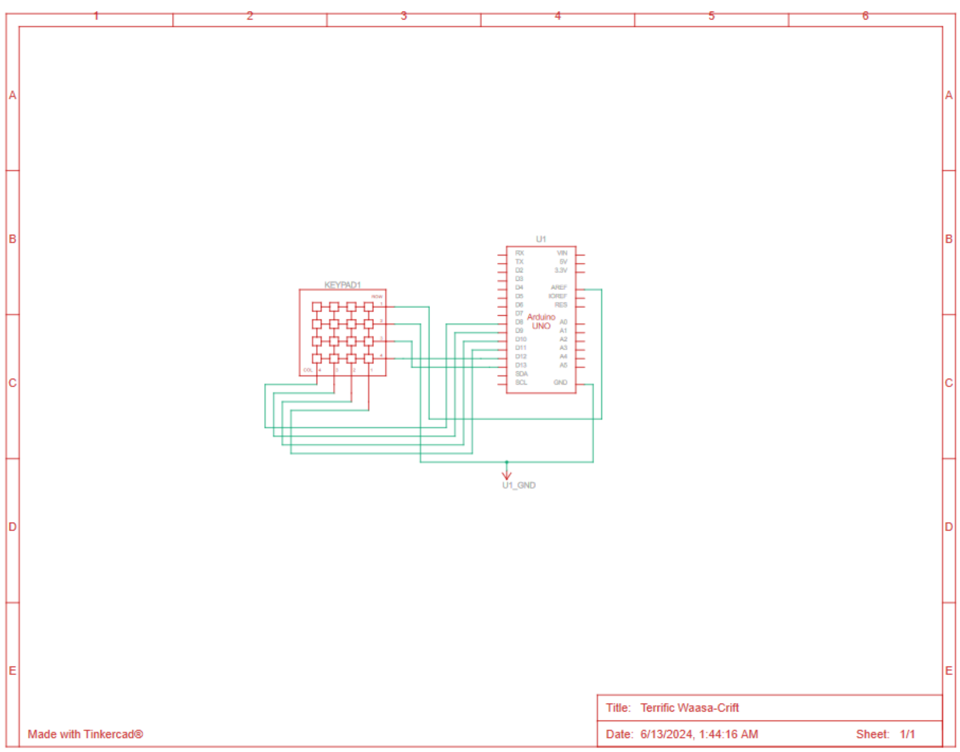

Fig. 3. Circuit diagram of keypad with Arduino.

Methodology:

1. Working principle of the washing machine prototype:
    a. The card is scanned on the RFID sensor.
    b. RGB will turn red.
    c. The display will ask the user to load their clothes into the machine.
    d. The display will ask the user to close the door of the washing machine.
    e. The user will close the door which is simulated by the tap module.
    f. The display will ask the user to choose between a washing cycle and a drying cycle.
    g. When pushbutton 1 is pressed the display will state: “Washing….”.
    h. RGB turns blue.
    i. Washing cycle begins and lasts according to the ultrasonic sensor values.
    j. Pump will add soapy water according to the ultrasonic sensor values.
    k. Motor will rotate back and forth continuously.
    l. Valves open and a pump will activate to remove remaining soapy water.
    m. Pump adds clean water.
    n. Motor rotates back and forth continuously.
    o. Valves open and the pump activates to remove water.
    p. LCD states: “Drying…”.
    q. RGB turns purple.
    r. Drying cycle begins.
    s. Motor will rotate clockwise continuously for a while.
    t. Motor stops then rotates counterclockwise continuously.
    u. Steps ‘s’ and ‘t’ repeat until time is up according to the ultrasonic sensor values.
    v. The display will print “Take out clothes”.
    w. Piezo buzzer will play a tone.
    x. RGB turns green indicating finished operation.
    y. When pushbutton 2 is pressed steps ‘p’ to ‘w’ are enforced.
2. Extra feature included: A piano for the user to enjoy playing while waiting for the laundry to be done.
3. Connect the components as stated in the experimental setup.
4. Test each component’s code individually to be certain they work and verify they are compatible with the Arduino.
5. Ensure proper communication between components such as I2C for LCD display.
6. Test the connection between components like the pushbutton interacting with the LCD display.
7. Install the needed libraries in Arduino:
    a. Keypad
    b. LiquidCrystal_I2C
    c. Servo
    d. Wire
    e. SPI
    f. MFRC522
8. Write a code consisting of the steps in the working principle and the extra feature.
9. Optimize the Arduino code for efficiency, clarity and responsiveness.
10. Upload the code to the Arduino via USB cable.
11. Test prototype.

Arduino Code:

```
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Keypad.h>

char keys[4][4] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
MFRC522 mfrc522(10, 9);   // Create MFRC522 instance
Servo myServo;
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int button = 5;
#define RED_PIN 13
#define GREEN_PIN 12
#define BLUE_PIN 11
#define CLK 19
#define DT 18
#define SW 6
const int tapPin = A0;
const int buzzerPin = 22;
const int trigPin = 3;
const int echoPin = 2;
byte rowPins[4] = {37, 36, 35, 34};
byte colPins[4] = {33, 32, 31, 30};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, 4, 4);

const int noteC = 261;
const int noteD = 294;
const int noteE = 329;
const int noteF = 349;
const int noteG = 392;
const int noteA = 440;
const int noteB = 493;
const int noteC_High = 523;
```

```
// Define the melody for the song (e.g., "Twinkle Twinkle Little Star")
int melody[] = {
  noteC, noteC, noteG, noteG, noteA, noteA, noteG,
  noteF, noteF, noteE, noteE, noteD, noteD, noteC,
  noteG, noteG, noteF, noteF, noteE, noteE, noteD,
  noteG, noteG, noteF, noteF, noteE, noteE, noteD,
  noteC, noteC, noteG, noteG, noteA, noteA, noteG,
  noteF, noteF, noteE, noteE, noteD, noteD, noteC
};

// Define the note durations: 4 = quarter note, 8 = eighth note, etc.
int noteDurations[] = {
  4, 4, 4, 4, 4, 4, 2,
  4, 4, 4, 4, 4, 4, 2,
  4, 4, 4, 4, 4, 4, 2,
  4, 4, 4, 4, 4, 4, 2,
  4, 4, 4, 4, 4, 4, 2,
  4, 4, 4, 4, 4, 4, 2
};

void setup() {
  Serial.begin(9600);
  SPI.begin();                // Init SPI bus
  mfrc522.PCD_Init();         // Init MFRC522
  lcd.init();    // Initialize the LCD with the I2C address
  lcd.backlight();  // Turn on the backlight
  pinMode(tapPin, INPUT);
  pinMode(button,INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT);
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);
  pinMode(CLK, INPUT);
  pinMode(DT, INPUT);
  pinMode(SW, INPUT_PULLUP);
  myServo.attach(8);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.println("System initialized. Waiting for RFID card...");
}
```

```
void loop() {
  char key = keypad.getKey();
    if (key) {
    if (key == 'D') {
      playSong();
    } else {
      playNote(key);
    }
    Serial.println(key);
  }

  if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
    Serial.println("RFID card detected.");
    delay(50);
    setRGBColor(0, 225, 0); //red
    lcd.clear();
    lcd.print("Load clothes");
    delay(3000);
    lcd.clear();
    lcd.print("Close the door");
     while (!digitalRead(tapPin)) {
      delay(100); // Wait until tap module senses door closed
    }
    Serial.println("Tap detected");
    delay(100);
    lcd.clear();
    lcd.print("What service?");
    lcd.setCursor(0, 1);
    lcd.print("1:Wash 2:Dry");
  }

  if (digitalRead(SW) == LOW) {
     Serial.println("pot pressed");
     delay(50);
     startDrying();
     signalCompletion();
  } else if (digitalRead(button) == LOW) {
     Serial.println("bottun pressed");
     delay(50);
     startWashing();
     signalCompletion();
  }
```

```
}

void startWashing() {
  lcd.clear();
  lcd.print("Washing...");
  setRGBColor(0, 0, 255);  // Set RGB to blue
  Serial.println("Washing started.");
  int cycles = determineCycles();
  for (int i = 0; i < cycles; i++) {
    myServo.write(0);
    delay(200);
    myServo.write(180);
    delay(200);
    myServo.write(0);
    delay(1000);
    for (int j = 0; j < 5; j++) {
      myServo.write(0);
      delay(200);
      myServo.write(180);
      delay(200);
    }
  delay(1000);
  startDrying();
  }
}

void startDrying() {
  lcd.clear();
  lcd.print("Drying...");
  setRGBColor(0, 200, 225); //purple
  int cycles = determineCycles();
  for (int i = 0; i < cycles; i++) {
    for (int j = 0; j < 3; j++) {
      myServo.write(0);
      delay(200);
      myServo.write(180);
      delay(200);
    }
  }
}

  int determineCycles() {
```

```
  long duration;
  int distance;

  // Send a 10us pulse to the trigger pin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the echo pin and calculate the duration
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance <= 5) {
    return 3; // If distance is less than or equal to 5 cm
  } else if (distance <= 20) {
    return 2; // If distance is between 6 cm and 20 cm
  } else {
    return 1; // If distance is more than 21 cm
  }
}

void signalCompletion() {
  lcd.clear();
  lcd.print("Take out clothes");
  for (int i = 0; i < 3; i++) {
    digitalWrite(buzzerPin, HIGH);
    delay(1000);
    digitalWrite(buzzerPin, LOW);
    delay(1000);
  }
  delay(2000);
  setRGBColor(255, 0, 0);  // Set RGB to green
  lcd.clear();
  lcd.print("Thank you");
  delay(2000);
  lcd.clear();
```

```
}

void setRGBColor(int red, int green, int blue) {
  analogWrite(RED_PIN, red);
  analogWrite(GREEN_PIN, green);
  analogWrite(BLUE_PIN, blue);
}

void playNote(char key) {
  switch (key) {
    case '1': tone(buzzerPin, noteC, 200); break;
    case '2': tone(buzzerPin, noteD, 200); break;
    case '3': tone(buzzerPin, noteE, 200); break;
    case 'A': tone(buzzerPin, noteF, 200); break;
    case '4': tone(buzzerPin, noteG, 200); break;
    case '5': tone(buzzerPin, noteA, 200); break;
    case '6': tone(buzzerPin, noteB, 200); break;
    case 'B': tone(buzzerPin, noteC_High, 200); break;
    case '7': tone(buzzerPin, noteC / 2, 200); break; // Lower octave C
    case '8': tone(buzzerPin, noteD / 2, 200); break; // Lower octave D
    case '9': tone(buzzerPin, noteE / 2, 200); break; // Lower octave E
    case 'C': tone(buzzerPin, noteF / 2, 200); break; // Lower octave F
    case '*': tone(buzzerPin, noteG / 2, 200); break; // Lower octave G
    case '0': tone(buzzerPin, noteA / 2, 200); break; // Lower octave A
    case '#': tone(buzzerPin, noteB / 2, 200); break; // Lower octave B
    default: break;
  }
}

void playSong() {
  int melodyLength = sizeof(melody) / sizeof(melody[0]);
  for (int i = 0; i < melodyLength; i++) {
    int noteDuration = 1000 / noteDurations[i];
    tone(buzzerPin, melody[i], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(buzzerPin);
  }
}
```

DATA COLLECTED

```
Distance: 7 cm - 2 cycles
Distance: 3 cm - 3 cycle
Distance: 4 cm - 3 cycle
Distance: 3 cm - 3 cycle
Distance: 3 cm - 3 cycle
Distance: 6 cm - 2 cycles
Distance: 8 cm - 2 cycles
Distance: 8 cm - 2 cycles
Distance: 10 cm - 2 cycles
Distance: 13 cm - 2 cycles
Distance: 17 cm - 2 cycles
Distance: 175 cm - 1 cycles
Distance: 121 cm - 1 cycles
```

Fig. 4. Data from ultrasonic sensor

DATA ANALYSIS

When the distance measured from the ultrasonic sensor is less than 5 cm then the serial monitor will print “3 cycles” which means that the washing and drying cycles will repeat 3 times and when the distance is between 6 cm and 20 cm then the washing and during cycles will repeat twice and last but not least when the distance is more than 21 cm then the cycles will repeat once.

RESULTS

The washing machine prototype starts when the RFID sensor detects a card, the RGB will turn on and turn red while the LCD display will ask the user to load their clothes and after 2 seconds it’ll ask the user to close the door and that is simulated by tapping the tap module. Subsequently, the display will ask the user to choose a process, the washing process or the drying process can be chosen by pressing the pushbutton or the rotary encoder respectively. Granted that the user presses the pushbutton, the display will state “Washing...” and the RGB will turn blue. Soon after, the servo motor will rotate back and forth for a second and stop, then after a delay of 1 second the motor will rotate back and forth again but this time for 2 seconds, this cycle, will repeat twice if the ultrasonic measures the distance to be between 6 to 20 cm, and once if it is more than 21 cm, however, if the distance is less than 5 cm then it will repeat three times. Afterwards, the LCD screen will display “Drying…” while the RGB turns purple. As the next step, servo motor will rotate back and forth for 1.2 seconds, this cycle length is also controlled by the ultrasonic sensor, so it will repeat twice if the ultrasonic measures the distance to be between 6 to 20 cm, and once if it is more than 21 cm, however, if the distance is less than 5 cm then it will repeat three times then stop. Last but not least, the screen will display “Take out clothes” and a tone will play by the buzzer. After a delay of 2 seconds, the RGB will turn green to indicate the completed process and the display will state “Thank you” for 2 seconds. On the other hand, if the rotary encoder is pressed, the prototype will play out the drying cycle which is according to the values of the ultrasonic sensor then the screen will display "Take out clothes" and a tone will play by the buzzer. After a delay of 2 seconds, the RGB will turn green to indicate the completed process and the display will state "Thank you" for 2 seconds.

Here is the link to the video of the prototype working:

https://github.com/Chrysanthly/Group-H/blob/main/MiniProject/WashingMachinePrototype.mp4

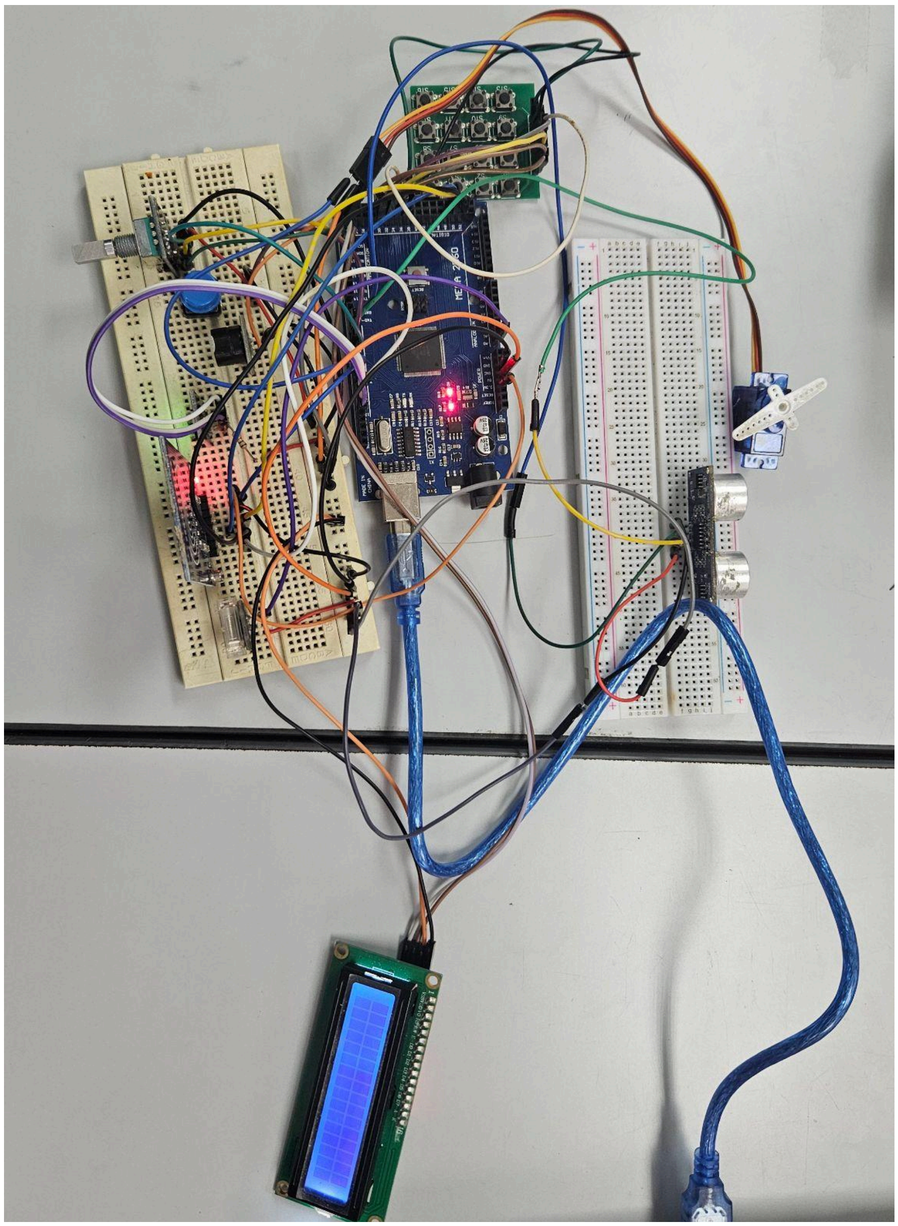

Fig. 5. Washing machine prototype

DISCUSSION

In products that involve consumers in every step, including proper user interface is crucial since it is customers who will decide the sales of the product. As such, washing machines are one of those products that rely heavily on their users. For our prototype I made sure to make the process easy to follow and simple, by using a single LCD display and adding simple codes to ask the users to follow the steps and ensure proper usage of the machine. Furthermore, by adding the tone at the end of the process we made the prototype accessible for visually impaired users.

As seen in the results section, a rotary encoder was utilized instead of a pushbutton. This allows for future projects or prototypes to use it to make the user interface more interactive and engaging, similar to commercial washing machines that make use of a knob to help users choose the desired setting for the load. Moreover, to simulate the pump and valve working we added a delay after the motor stops spinning. Actual pumps and valves were not added to the prototype for the lack of resources and time. For the drying cycle, using a different motor that is able to rotate continuously at high speeds would be more appropriate to better simulate the working principle of the washing machine drying. The reason washing machines use high speed continuous rotations is because that is better for removing as much water as possible.

As for the part containing the data collected and analysis of that data, the reason the cycle repeats only once when the distance measured is more than 21 cm, and repeats twice when the distance is between 6 and 20 cm, and finally repeats three times when the distance is less than 5 cm; is because if we imagine the ultrasonic sensor to be at the roof inside the machine then whenever the distance is low it would mean that the amount of clothes in the machine is high and the other way around is correct. Furthermore, the machine shouldn't wash and dry all amounts of clothes with the same power, energy and time length, as that would use unnecessary energy and time when a lesser amount is doing the work as intended. This method ensures that the machine adjusts its washing and drying cycles based on the load size, optimizing energy and time usage.

Further discussion can explore the potential for integrating additional sensors and functionalities to enhance the prototype's efficiency and user-friendliness. For instance, incorporating a weight sensor could provide a more accurate measure of the load size, improving the washing and drying cycle's optimization. Additionally, integrating a temperature sensor could help in adjusting the water temperature dynamically based on the fabric type and load size, ensuring better washing results and fabric care.

Another aspect to consider is the safety features of the prototype. While the current design includes basic safety mechanisms, future iterations could incorporate more advanced features such as water leakage detection, motor overheating protection, and emergency stop functionality. These features would make the washing machine safer for home use and prevent potential accidents or damages.

Moreover, the implementation of a more sophisticated user interface, possibly with a touchscreen display, could provide users with more options and settings to choose from, enhancing their control over the washing process. This could include settings for different fabric types, stain levels, and washing intensities, making the washing machine more versatile and user-friendly.

In terms of DIY appliance automation, this project demonstrates the potential of Arduino for controlling complex systems like washing machines. It opens up possibilities for hobbyists and makers to create customized and automated solutions for their household needs. This project can serve as a foundation for further exploration and innovation in the field of DIY appliance automation, inspiring others to experiment and improve upon the design.

SAFETY

The current prototype lacks a feature to lock the door while the machine is running, which presents a significant safety risk. In the worst-case scenario, if the door were to be unlocked by a child during operation, it could lead to catastrophic consequences, such as water spillage, electrical hazards, and potential injury from moving parts. Therefore, implementing a robust door locking mechanism that activates during the washing and drying cycles is essential. This mechanism should ensure that the door remains securely locked until the machine has completed its cycle and all moving parts have come to a stop. Additionally, the outer drum of a washing machine should be supported by a heavy-duty spring. This is crucial because, during operation, the machine can experience significant vibrations and shaking. The spring helps absorb these forces, preventing excessive movement and reducing the risk of damage to the machine's internal components. Without such a damping system, the vibrations could lead to structural damage, component failure, and potential safety hazards over time.

Furthermore safety features that should be considered are adding more sensors, emergency stopping and many more. First and foremost, the machine should be able to detect water leakage. To illustrate further, incorporating sensors to detect any water leakage from the machine so if a leak is detected, the system should automatically shut down and alert the user to prevent water damage and electrical hazards. Secondly, protecting motors from overheating by implementing temperature sensors to monitor the motor's temperature. If the motor overheats, the system should stop the operation and alert the user to prevent potential fires or damage to the motor. Moreover, adding an easily accessible emergency stop button that allows the user to immediately halt the machine's operation in case of an emergency. Furthermore, automatic shutdown on error detection. By programming the Arduino to detect errors or malfunctions, such as motor failure or sensor issues. Upon detecting such errors, the system should automatically shut down and display an error message on the LCD screen to inform the user.

By integrating these additional safety features, the washing machine prototype can be made significantly safer and more reliable, reducing the risk of accidents and extending the machine's lifespan. Implementing these enhancements will also ensure compliance with standard safety regulations and increase user confidence in the automated washing machine system.

CONCLUSION

In conclusion, this project successfully demonstrated the feasibility and advantages of automating a traditional washing machine using an Arduino microcontroller. By replacing the original control unit with Arduino-based automation, we achieved customizable washing and drying cycles controlled by user inputs and sensor data. The experiment supported the hypothesis that Arduino can effectively enhance washing machine functionality through flexible programming and sensor integration.

The main findings highlight the potential of DIY appliance automation to provide cost-effective, user-customizable solutions for household appliances. This approach not only improves user experience by allowing tailored settings but also enhances safety with features like door locking mechanisms and error detection systems. Looking forward, the broader implications of this experiment extend to the realm of smart home technology, where Arduino's adaptability could revolutionize how appliances interact with users and their environments. By enabling consumers to modify and enhance their appliances according to specific needs, Arduino-based automation contributes to sustainable living practices and technological empowerment.

In essence, this project underscores Arduino's role in advancing DIY appliance automation and sets a foundation for future innovations in home appliance design and functionality.

RECOMMENDATIONS

Implementing a pump and a heating valve would improve and make the prototype a more realistic washer,the pump can electrically remove and add water when needed while the water heater valve could add warmer water when needed since higher temperatures help clean clothes better. Adding a thermostat to measure the temperature of the incoming water will help with the implementation of the heater valve. Moreover, using a stepper motor would result in faster rotation speeds and would allow the drum to rotate continuously clockwise or counterclockwise. A WIFI module is a good choice to make the machine an IOT device and let users monitor their cycles and processes in real time. For good measure, exchanging the ultrasonic sensors with photoelectric sensors will help improve the accuracy of the product. Last but not least, adding more washing and drying options for different kinds of clothes and materials.

REFERENCES

Woodford, C. (2022, January 14). *Clothes washing machines: How they work*. Explain that Stuff. https://www.explainthatstuff.com/washingmachine.html

McCallum, B. (2024, January 7). *How does a washing machine work: All types explained*. Oh So Spotless. https://ohsospotless.com/how-does-a-washing-machine-work/

APPENDICES

Fig. 6. Display stating "Load clothes"

Fig. 7. Display stating "Close the door"

Fig. 8. Display asking user to choose a service

Fig. 9. Display stating “Washing…”

Fig. 10. Display stating “Drying…”

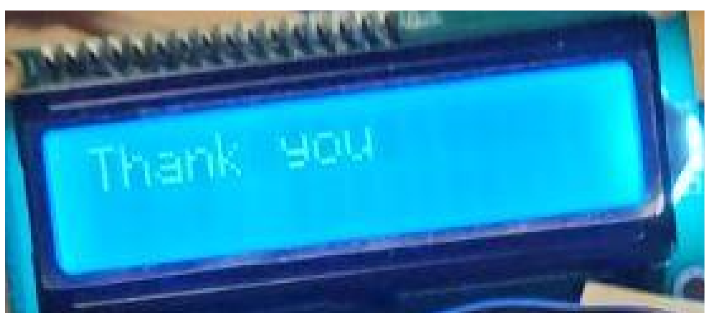

Fig. 11. Display stating “Thank you”

Fig. 12. Distances measured by ultrasonic sensor

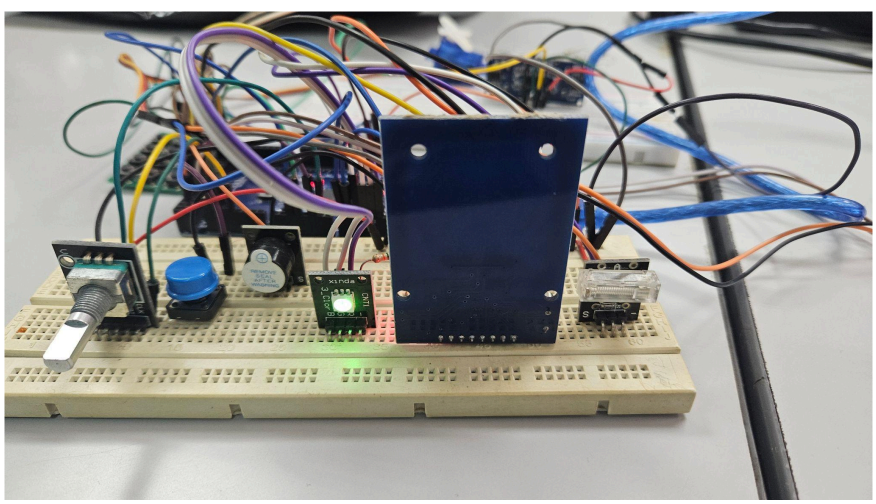

Fig. 13. The prototype from the front

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Zulkifli for his invaluable guidance and support throughout the execution of this experiment. His expertise and insightful feedback greatly enhanced my understanding of the principles underlying Arduino-based electronics. Additionally, for his assistance in providing constructive suggestions for improvement.

STUDENT'S DECLARATION

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 acknowledgement, and that the original work contained herein have 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 no further improvement on the reports is needed from any of the individual's 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.

<table>
<tr><td rowspan="3">Name: Syeda Samia
Matric Number: 2123536
Contribution: Introduction. Procedure, Data collected, Data analysis, Results, Discussion, Safety, Conclusion, Recommendations</td><td>Read</td><td>✓</td></tr>
<tr><td>Understand</td><td>✓</td></tr>
<tr><td>Agree</td><td>✓</td></tr>
<tr><td rowspan="3">Name: Athirah Haziqah bt Baderulhisham
Matric Number: 2210240
Contribution: Abstract, Introduction</td><td>Read</td><td></td></tr>
<tr><td>Understand</td><td></td></tr>
<tr><td>Agree</td><td></td></tr>
<tr><td rowspan="3">Name: Siti Heidi Amira Binti Azrul
Matric Number: 2111970
Contribution:</td><td>Read</td><td></td></tr>
<tr><td>Understand</td><td></td></tr>
<tr><td>Agree</td><td></td></tr>
</table>