MECHATRONICS SYSTEM INTEGRATION

MINI PROJECT:
INTEGRATION ARDUINO WITH COMPONENTS AS MASTER AND SLAVE FOR WASHING MACHINE SIMULATION

GROUP NUMBER: A

PROGRAMME: MECHATRONICS

INSTRUCTOR :

ASSOC. PROF. DR. ZULKIFLI BIN ZAINAL ABIDIN

GROUP MEMBERS	MATRIC NO
AISYAH SOFEA BINTI OTHMAN	2115386
SITI ALIAA BINTI IBRAHIM	2112618
MUHAMMAD ADLI FAHMI BIN TAJUL ARIS	2113095
ETHAR ABDALLA ABDELKARIM OSMAN	2111282
NURAIN AINAA AQILAH BINTI ROSLI	2114560

DATE OF SUBMISSION: Monday, 22th January 2024

Table of Contents

Abstract

This study details a versatile washing machine system integrating automatic and manual modes through two Arduino boards. The potentiometer facilitates seamless user transitions between modes. In manual mode, operations, such as spinning (red LED, 10s), rinsing (blue LED, 5s), and completion (green LED with buzzer), are initiated by a pushbutton. A servo motor visually represents spinning, and a 7-segment display communicates the countdown timer. The automatic mode employs the "Arduino Bluetooth Control" app, utilising a Bluetooth serial module for user-friendly operation. Emphasising user experience, the design combines traditional and modern controls, enhancing adaptability, user interaction, and operational efficiency. Another experiment focuses on securing a washing machine through RFID technology, simulating a locked door mechanism triggered by RFID authentication, enhancing safety and preventing unauthorised access.

Introduction

This project aims to integrate Arduino as both Master and Slave along with some input and output components to simulate a washing machine. The project comprises two main parts: the door part, where the RFID reader (MFRC522) controls access, and the washing machine part, which involves the operation of the machine itself.

Door Part

The first part involves the integration of an RFID card reader as an input device, interacting with an LCD screen and a servo motor as output components. RFID technology is utilized for user identification and access control to simulate the washing machine door. Only users with the correct RFID cards are granted access, triggering the opening of the door through the servo motor. The LCD displays the access status, providing a tangible representation of granted access.

Washing Machine

The second part focuses on the washing machine's operation, allowing users to initiate the sequence either manually or wirelessly through bluetooth connection in their mobile phones. A push button on the master Arduino starts the washing machine sequence manually,

signalling the slave Arduino. The sequence includes a servo rotation (acting as impeller) during washing, red LED activation (10 seconds), blue LED activation during rinse (5 seconds), and green LED and buzzer activation to show that the wash has ended. The 7-segment display acts as the timer for each mode.

Material & Equipment

- 4 Arduino (2 for Door, 2 for Washing Machine)
- 2 Medium Size Breadboard
- 1 Small Size Breadboard
- Many jumpers
- 5 220 Ohm Resistors
- 3 LED (Red, Blue, Green)
- 1 Potentiometer
- 1 Servo Motor
- 1 7-Segment
- 1 LCD I2C
- 1 RFID RC 522
- 4 USB 2.0 Cable Type A/B (2 for Door, 2 for Washing Machine)
- 2 Laptop
- 1 Bluetooth Module (HC-06)
- 1 Buzzer

Experimental Setup

1.1 Door

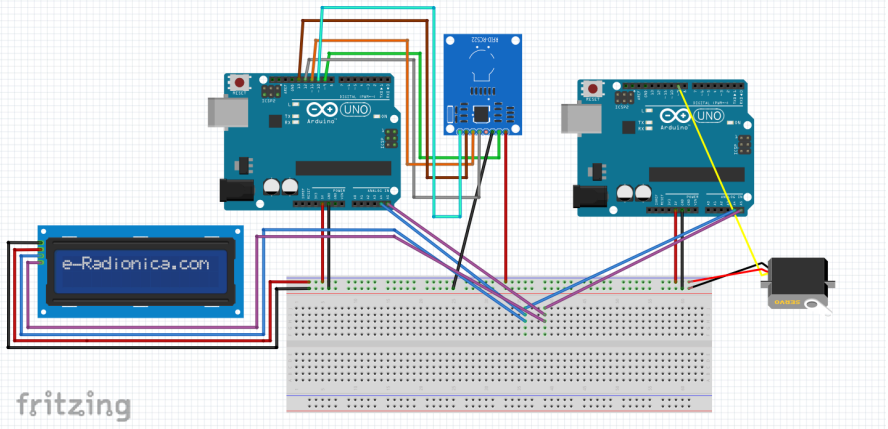


Figure 1.1 : Circuit Diagram of Washing Machine's Door

In the observed setup, FRID and LCD were placed on the same Arduino, referred to as the ‘master.’ Meanwhile, the servo motor was placed on another Arduino, referred to as the ‘salve.’ The integration between these two Arduino master-slave setups communicated using 12C communication. For the gate part, we set up 2 arduino into master-slave using I2C communication/protocol. All connections for RFID, LCD and servo are as below.

Master	**Slave**	**Arduino**	**LCD I2C**	**RFID RC 522**	**SERVO MOTOR**
5V	5V	5V	5V	5V	+supply
GND	GND	GND	GND	GND	-supply
SDA/A4	SDA/A4	A4	SDA	-	-
SCL/A5	SCL/A5	A5	SCL	-	-
		PIN 9		RST	PWM
		PIN 10		MOSI	
		PIN 11		MISO	
		PIN 12		SDA	
		PIN 13		SCK	

1.2 Washing Machine

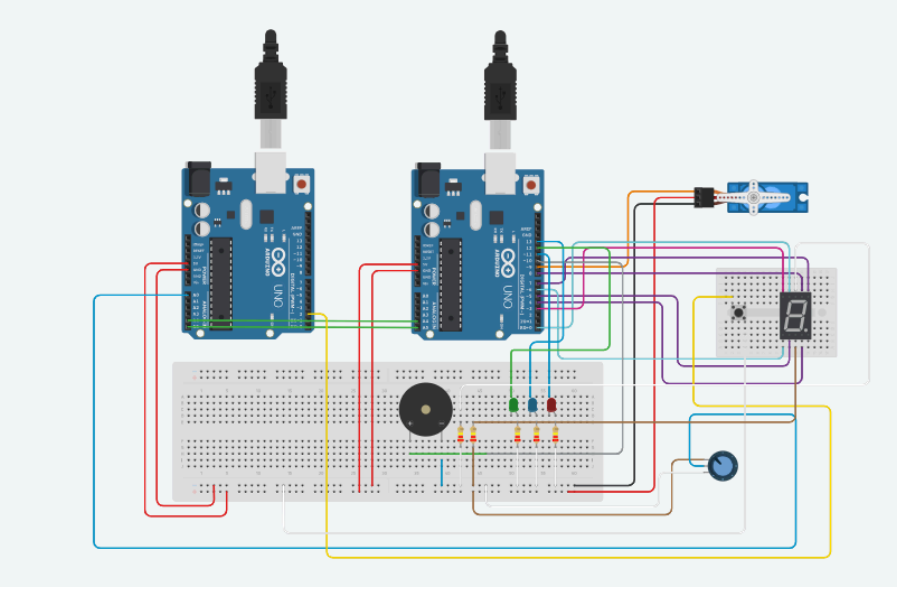


Figure 1.2 : Circuit Diagram of Washing Machine

For the washing part, we set up 2 arduino into master-slave using I2C communication/protocol. It is connected in a such that,

Master	Slave
5V	5V
GND	GND
SDA/A4	SDA/A4
SCL/A5	SCL/A5

Pin configuration at the Arduino for 'Master-Slave' I2C Communication

After that, all the components that we are gonna use were connected in accordance. However in the circuit diagram, since we use the 'Tinkercad' to illustrate it, it doesn't have bluetooth serial (HC-06) components and for bluetooth serial it is commonly used for serial communication hence it is connected to the RX and TX pin of the arduino. Firstly, the master were set up such that;

Components	Pin
Pushbutton	2
	GND
Potentiometer	A0
	5V
	GND
Bluetooth Serial (HC-06)	TX-1
	RX-0

Pin configuration at the Arduino for 'Master'

And then, the slave were setup such that;

Components	Pin
Servo Motor	GND

	5V	
	9	
Red LED	11	
	GND	
Blue LED	13	
	GND	
Green LED	12	
	GND	
Buzzer	10	
	GND	
7-Segment	*A*	8
	B	7
	C	4
	D	5
	E	6
	F	0
	G	3
Bluetooth Serial (HC-06)	VCC - 5V	
	GND - GND	
Resistors	- *Three 220 Ohm resistors are connected to each of the anode of LED and 2 of it to the common cathode of 7-segment*	

Pin configuration at the Arduino for 'Slave'

Methodology

Part 1 : Door Part

For the washing machine door section, our focus was on the method of opening the door. In this project, we utilised RFID to simulate the process of opening the washing machine door. In this simulation, when the designated RFID tag/card is scanned, the servo, functioning as the door, will rotate to indicate that the door has been opened, and the LCD will display 'door on.' The door could only be opened using a single RFID tag/card. The washing machine door would be locked during the washing process and could only be opened by using the designated RFID tag/card. This system can also be categorised under the safety section.

2.1 Arduino Coding

2.1.1 Master Code

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

#define SS_PIN 10  // Define the slave select pin for the RFID module
#define RST_PIN 9  // Define the reset pin for the RFID module

MFRC522 rfid(SS_PIN, RST_PIN); // Create an instance of the RFID library

LiquidCrystal_I2C lcd(0x27, 16, 2);  // I2C address 0x27, 16 columns and 2 rows

#define SLAVE_ADDRESS 4  // Replace with the actual slave address of the servo Arduino

void setup() {

  //Wire.begin();
  Serial.begin(9600);
  SPI.begin();       // Init SPI bus
  rfid.PCD_Init();    // Init RFID module

  lcd.begin();   // Init LCD
  lcd.backlight();    // Turn on the backlight

  Wire.begin();
```

```
  lcd.print("Place RFID tag");
}

void loop() {
  // Look for new RFID cards
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    lcd.clear();
    Serial.print("UID:");

    //Display UID on LCD
    for (byte i = 0; i < rfid.uid.size; i++) {
    lcd.print(rfid.uid.uidByte[i] < 0x10 ? " 0" : " ");
    lcd.print(rfid.uid.uidByte[i], HEX);
    }

    // Check if the UID matches your specified UID
    if (checkUID()) {
      lcd.setCursor(0, 1);
      //lcd.print("AG");
      lcd.print(" Door Open");
      //openDoor(); //call function to open the door
      sendDataToSlave("Door ON");  // Send data to the servo Arduino
      // Add your code to turn on the washing machine here
    } else {
      lcd.setCursor(0, 1);
      lcd.print("Access Denied");
    }

    delay(1000);  // Display the message for 2 seconds
    lcd.clear();
    lcd.print("Place RFID tag");
    delay(50);   // Delay to avoid multiple reads
  }
}

boolean checkUID() {
  // Replace this with your UID. You can find it using the example in the MFRC522 library.
  byte correctUID[] = {0xC3, 0x69, 0x16, 0x13};

  if (rfid.uid.size == sizeof(correctUID)) {
    for (byte i = 0; i < rfid.uid.size; i++) {
      if (rfid.uid.uidByte[i] != correctUID[i]) {
        return false;
      }
    }
    return true;
  }
  return false;
```

```
}

void sendDataToSlave(String data) {
  Wire.beginTransmission(SLAVE_ADDRESS);
  Wire.write(data.c_str());  // Send the data as a C-style string
  Wire.endTransmission();
}
```

2.1.2 Slave Code

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

Servo myServo;  // Create a Servo object

void setup()
{
  Wire.begin(4);                // join I2C bus with address #4
  Wire.onReceive(receiveEvent); // register event
  Serial.begin(9600);           // start serial for output

  myServo.attach(9);  // Attach the servo to pin 9 or replace with your pin
}

void loop()
{
  delay(100);
}

// Function that executes whenever data is received from the master
// This function is registered as an event, see setup()
void receiveEvent(int howMany)
{
  while (1 < Wire.available()) // Loop through all but the last
  {
    char c = Wire.read(); // Receive byte as a character
    Serial.print(c);      // Print the character
  }

  int x = Wire.read();  // Receive byte as an integer
  Serial.println(x);    // Print the integer

  // Check if x (received from master) is a specific value indicating the correct UID
  if (x == 0xC3, 0x69, 0x16, 0x13) {  // Replace 42 with the actual value you expect
```

```
    // Add your servo control code here
    myServo.write(90);  // Assuming 90 is the angle for the desired servo position
    delay(1000);        // Wait for the servo to reach the position

    // You can add more servo control code or perform other actions here

    // Move the servo back to the initial position
    myServo.write(0);   // Assuming 0 is the angle for the initial position
    delay(1000);
  }
}
```

Part 2 : Washing Machine

For the methodology for the washing machine system, our approach focused on the distinct modes of operation: automatic and manual. The system integration involves two Arduino boards and carefully chosen components to facilitate a smooth communication between these modes. A crucial element in this setup is the potentiometer, serving as the control interface for seamlessly toggling between automatic and manual control, thereby ensuring user flexibility.

In the manual mode, the initiation of washing machine operations—spinning (indicated by a red LED) for 10 seconds, rinsing (indicated by a blue LED) for an additional 5 seconds, and completion (indicated by a green LED) with a buzzer alarm—is triggered by pressing a pushbutton. Pertinent to that, the servo motor is used for visually representation of the spinning mode. Simultaneously, a 7-segment display visually communicates the countdown timer, enhancing user awareness during the manual washing process.

The automatic mode introduces a mobile application, "Arduino Bluetooth Control," replacing the traditional pushbutton with a more user-friendly interface. Central to this shift is the Bluetooth serial (HC-06) module, initiating washing machine operations via a button press within the mobile app. The design rationale prioritises user experience, emphasising specific components like the potentiometer and Bluetooth module to enhance user convenience, intuitive control, and seamless transitions between operational modes.

2.2 Arduino Coding

2.2.1 Master Code

```
#include <Wire.h>
#include <SoftwareSerial.h>
SoftwareSerial btSerial(0, 1);  // RX, TX pins for Bluetooth module

int sensorpin1 = A0;
int mode = 0;  // Variable to track the current mode
boolean btConnected = false;
char key;

const int buttonPin = 2;  // Pin to connect the push button
int countdown = 9;        // Initial countdown time in seconds

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);  // Set button pin as input with internal pull-up resistor
  pinMode(0, INPUT);
  pinMode(1, OUTPUT);
  btSerial.begin(9600);
  Wire.begin();  // Initialize I2C communication
  Serial.begin(9600);
  Serial.println("Master Arduino - Push Button");
}

void loop() {
  int value = analogRead(sensorpin1);

  Serial.println(value);
  // Determine mode based on potentiometer value
  if (value < 100) {
    mode = 0;
  } else {
    mode = 1;
  }
  // Perform actions based on the mode
  if (mode == 0) {
    // Actions for mode 0
    Serial.println("Mode 0: manual");
    // delay(1000);
    // Wait for the button press with debounce
    if (digitalRead(buttonPin) == LOW) {
      Serial.println("Button Pressed");
      Wire.beginTransmission(9);  // Slave address
      Wire.write(1);              // Send a byte indicating button press
      Wire.endTransmission();
      Serial.println("Pushbutton Value");
      delay(200);  // Debounce delay
    }
```

```
  //Check if the countdown has started
  // if (countdown > 0) {
  //   Serial.print("Time Left: ");
  //   Serial.println(countdown);
  //   delay(1000);  // Wait for 1 second
  //   countdown--;
  // }
  delay(1000);
 }

 else {
  // Actions for mode 1
  Serial.println("Mode 1: auto");
  if (btSerial.available()) {

    Serial.write(btSerial.read());


    if (Serial.available())
      btSerial.write(Serial.read());

    char koi = Serial.read();
    Serial.println(koi);

    key = koi;
    switch (key) {
      case 's':
        Wire.beginTransmission(9);  // Slave address
        Wire.write(1);              // Send a byte indicating button press
        Wire.endTransmission();
        Serial.println("Pushbutton Value");
        delay(200);  // Debounce delay
    }
  }delay(1000);
 }
}
```

2.2.2 Slave Code

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

const int servoPin = 9;
const int buzzerPin = 10;
const int redLedPin = 11;
const int greenLedPin = 12;
const int blueLedPin = 13;
const int segmentPins[] = {8, 7, 4, 5, 6, 0, 3}; // segments a, b, c, d, e, f, g
const int buttonPin = 2;  // Push button pin on master Arduino

Servo myservo;
```

```
bool motorRunning = false;
bool redLedBlinking = false;
bool blueLedOn = false;
bool sequenceStarted = false;
unsigned long lastButtonPressTime = 0;
unsigned long lastBlinkTime = 0;
int countdown = 9;  // Initial countdown (assume to be in minutes but we use seconds instead
because easy to read)
bool countdownEnabled = false;  // Flag to indicate if countdown is enabled
bool secondPhase = false;  // Flag to indicate if the second phase has started

void setup() {
  Wire.begin(9);
  Wire.onReceive(receiveEvent);
  pinMode(servoPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
  pinMode(greenLedPin, OUTPUT);
  pinMode(blueLedPin, OUTPUT);

  myservo.attach(servoPin);
  myservo.writeMicroseconds(1500);

  for (int i = 0; i < 7; i++) {
    pinMode(segmentPins[i], OUTPUT);
  }

  pinMode(buttonPin, INPUT_PULLUP);

  Serial.begin(9600);
  Serial.println("Please Press the Pushbutton");
}

void loop() {
  // Wait for the button press with debounce
  if (digitalRead(buttonPin) == LOW) {
    Serial.println("Button Pressed");
    Wire.beginTransmission(9);  // Slave address
    Wire.write(1);  // Send a byte indicating button press
    Wire.endTransmission();
   // Serial.println("Command Sent to Slave");
    delay(200);  // Debounce delay
  }

  if (sequenceStarted) {
    startSequence();
  }
}

void receiveEvent() {
  int command = Wire.read();  // Read the command from the master

  // Master has sent a command
  if (command == 1) {
```

```
    Serial.println("Laundry Start");
    sequenceStarted = true;
    countdownEnabled = true;  // Start the countdown
  }
}

void startSequence() {
  // Start the continuous rotation servo
  myservo.writeMicroseconds(2000);
  motorRunning = true;
  redLedBlinking = true;
  lastButtonPressTime = millis();
  Serial.println("Washing Start : 10 minutes ");

  while (motorRunning) {
    // Check if the motor has been running for 10 seconds
    if (millis() - lastButtonPressTime >= 10000) {
      // Stop the continuous rotation servo
      myservo.writeMicroseconds(1500);
      motorRunning = false;
      Serial.println("Washing Done");

      // Turn off the red LED
      digitalWrite(redLedPin, LOW);
      redLedBlinking = false;

      // Turn on the blue LED to signal rinse mode
      digitalWrite(blueLedPin, HIGH);
      Serial.println("Rinse On : 5 minutes ");
      blueLedOn = true;

      // Record the time the blue LED turned on
      unsigned long blueLedOnTime = millis();

// Continue with the rest of the sequence after the blue LED has been on for 5 seconds
      while (blueLedOn && millis() - blueLedOnTime < 5000) {
        // Optional: You can add other tasks here if needed
        // Update the countdown during the rinse mode
        countdown = 5 - ((millis() - lastButtonPressTime - 10000) / 1000);
        displayDigit(countdown);
        delay(1000); // Delay to update the countdown every second
      }

      // Turn off the blue LED
      digitalWrite(blueLedPin, LOW);
      blueLedOn = false;
      Serial.println("Rinse Finished");

      // Turn on the green LED
      digitalWrite(greenLedPin, HIGH);
      Serial.println("Laundry Done");

      // Turn on the buzzer
      digitalWrite(buzzerPin, HIGH);
```

```
      Serial.println("Yeay");

      // Wait for 5 seconds with the buzzer and green LED on
      delay(5000);

      // Turn off the buzzer, green LED
      digitalWrite(buzzerPin, LOW);
      digitalWrite(greenLedPin, LOW);
      Serial.println("Washing Machine Off");

      // Reset sequence flag
      sequenceStarted = false;
    }

    // Blink the red LED while the servo is running
    if (redLedBlinking && millis() - lastBlinkTime >= 500) {
      digitalWrite(redLedPin, !digitalRead(redLedPin));
      lastBlinkTime = millis();
    }

    // Update the countdown during the servo and LED sequence
    countdown = 9 - ((millis() - lastButtonPressTime) / 1000);
    displayDigit(countdown);
    delay(1000); // Delay to update the countdown every second
  }

  // Display the final countdown value
  displayDigit(countdown);
}

void displayDigit(int digit) {
  // Turn off all segments initially
  turnOff();

  // Define the segments for each digit
  const int digitSegments[][7] = {
    {1, 1, 1, 1, 1, 1, 0}, // 0
    {0, 1, 1, 0, 0, 0, 0}, // 1
    {1, 1, 0, 1, 1, 0, 1}, // 2
    {1, 1, 1, 1, 0, 0, 1}, // 3
    {0, 1, 1, 0, 0, 1, 1}, // 4
    {1, 0, 1, 1, 0, 1, 1}, // 5
    {1, 0, 1, 1, 1, 1, 1}, // 6
    {1, 1, 1, 0, 0, 0, 0}, // 7
    {1, 1, 1, 1, 1, 1, 1}, // 8
    {1, 1, 1, 1, 0, 1, 1}  // 9
  };

  // Display the segments for the given digit
  for (int i = 0; i < 7; i++) {
    digitalWrite(segmentPins[i], digitSegments[digit][i]);
  }
}
```

```
void turnOff() {
  // Turn off all segments
  for (int i = 0; i < 7; i++) {
    digitalWrite(segmentPins[i], LOW);
  }
}
```

Data Collection

3.1 Master Arduino (Washing machine)

The master Arduino serves as the control unit for the washing machine sequence. The initialization and setup involve configuring components such as the Bluetooth module, push button, analog sensor, and establishing serial communication at a rate of 9600 baud. Two operational modes are defined: Manual Mode (Mode 0) and Auto Mode (Mode 1). In Manual Mode, activated by a potentiometer value below 100, the system waits for a button press on Pin 2, signalling the start of the washing sequence. In Auto Mode, activated by a potentiometer value of 100 or above, the Arduino listens for commands from the Bluetooth module. Upon receiving specific commands, such as 's,' it triggers actions, including sending signals to the slave Arduino. The code also includes optional countdown displays during the washing and rinsing phases. Communication is facilitated through Bluetooth serial communication and I2C communication using the Wire library.

3.2 Slave Arduino (Washing machine)

The slave Arduino acts as the receiver for the washing machine control commands from the master Arduino. It is configured with pins for a servo motor, buzzer, LEDs, and a 7-segment display. The washing machine sequence is executed upon receiving the '1' command from the master Arduino. The sequence consists of three phases: Washing (10 minutes) with a blinking red LED, Rinsing (5 minutes) with a solid blue LED, and Completion with a solid green LED and sounding buzzer. A 7-segment display is employed to show the remaining time in seconds during the washing and rinsing phases. The slave Arduino logs the initiation of the washing machine sequence, the completion of each phase, the time and duration of each phase, and any errors or unexpected events that may occur during operation. The sequence execution ensures a controlled and automatic washing process based on the master Arduino's commands.

Data Analysis

The data analysis for the RFID-controlled door and washing machine system encompasses various aspects. Beginning with the door access logs generated by the master Arduino during RFID card authentication, the analysis focuses on identifying patterns in access events. This includes assessing trends in successful and unsuccessful access attempts over time, examining user behaviour, and ensuring the security of the system by analysing access denied events.

Simultaneously, the analysis extends to the washing machine control system. It involves examining the logs related to the activation and deactivation of the washing machine triggered by specific commands from the master Arduino. This includes evaluating the timing and duration of washing cycles and ensuring that the washing machine's operational states align with the commands sent by the master Arduino.Furthermore, the overall system performance is assessed by analysing the efficiency of data transmission between the master and slave Arduinos for both the door and washing machine control. Potential bottlenecks or communication issues that may impact the reliability of both functionalities are identified.

In summary, this comprehensive data analysis covers the entire system, including door access control and washing machine operation. It aims to provide insights into user interactions, security measures, and the overall performance of the RFID-controlled door and washing machine system.

Result

The integration between inputs and outputs to create a washing machine simulation was successfully established by using the communication of two Arduino microcontrollers acting as a Master and a Slave. The communication between 2 Arduino allows the inputs and outputs to communicate with each other and integrate to run a system.

In the first part of the experiment, integration of the RFID card reader as input with LCD and servo motor as outputs also was successfully achieved. RFID is used to identify users that were given access to use the washing machine. The RFID tags that have a unique

UID helps control access to simulate the washing machine door. Only users with the correct RFID card were granted the access to open the door of the washing machine. When users scan the card to the RFID scanner, LCD will display whether they were given access to it or the access was denied. The servo motor would turn, whenever the access was granted.

The second part is for the operation of the washing machine. For this project, a system that allows the machine to function either manually or automatically by turning the potentiometer was successfully created. For manuals, users can manually press the push button on the master Arduino to start the washing machine sequence by sending a signal from master Arduino to slave Arduino. The sequence includes the rotation of the servo for 9 seconds and red LED on during washing mode, blue LED on for 5 seconds during rinse mode and buzzer goes off, and green LED on when the all the process is done. The 7 segment works to display the timer of every mode.To operate the washing machine manually and automatically are exactly the same. However, the wireless input to start the washing machine via bluetooth module was successfully obtained in order for the washing to work automatically.

Discussion

The basis of this washing machine simulation is the effective integration of two Arduino microcontrollers, one of which functions as the master and the other as the slave. The coordinated implementation of a regulated washing process is made possible by this communication configuration, which facilitates smooth interaction between inputs and outputs.

RFID Technology Integration: In the initial phase of the project, the usage of RFID technology for access control and user identification worked well. The distinct UIDs of RFID tags made it easier to identify users and restricted access to the washing machine to those who were authorised. The RFID card reader, LCD, and servo motor worked together to provide a realistic simulation of a secure door system. Certain actions were triggered when users scanned their RFID cards; the servo motor turned on upon access confirmation, and the LCD showed the current access status.

Operation of Washing Machine: Both manual and automatic modes of operation were exhibited by the washing machine, which was controlled via master and slave Arduino

communication.By pushing a button on the master Arduino, which communicates with the slave Arduino, users can start the washing process manually. Servo rotation, LED signals, and buzzer alarms are all included in the sequence at various points. The washing process's complete completion shows how well the master-slave communication works to govern and coordinate a variety of outputs.

Bluetooth communication and potentiometer control: The potentiometer allowed users to easily switch between automatic and manual mode. The Arduino responded differently to values below 100 (Manual Mode) and over 100 (Auto Mode), depending on the potentiometer values. Through Bluetooth connectivity, the master Arduino receives commands from the "Arduino Bluetooth Control" smartphone application, enabling wireless input for autonomous operation. With the option to operate the washing machine remotely, this design improves user ease.

Opportunities for Learning and Difficulties: Nevertheless, there were some difficulties with the initiative. Numerous problems surfaced during installation, highlighting the significance of troubleshooting and debugging electronic circuitry. These difficulties fostered skill development and expertise in electronic system development by offering invaluable real-world experience. Even with the problems, the inputs and outputs are successfully integrated, setting the stage for more initiatives. The expertise gained also improves the capacity to manage intricate electronic systems.

In conclusion, the washing machine simulation project achieved its objective of integrating Arduino with components for effective washing machine control. The combination of RFID technology, servo motor control, potentiometer-based mode selection, and Bluetooth communication demonstrates the project's versatility and potential for practical applications in automated systems. The challenges encountered during the project contribute to the learning experience, preparing for future endeavours in electronic circuit design and implementation.

Recommendation

For future improvements, it is suggested to merge the code by combining the RFID-based door access system and the washing machine operation into a unified script. By merging these functionalities, the system can be more cohesive, enhancing maintainability and overall system integration. Moreover, implementation of a user profile system where

each user can store their preferences for washing machine settings could be very useful. This could include preferred wash cycles, temperatures, and spin speeds. The system could automatically adjust the settings based on the user's profile when they initiate a wash. Additionally, instead of just giving the option to start the wash wirelessly through bluetooth, we could add other options such as monitoring the timer wirelessly to allow for remote monitoring, which could be convenient for the user. A final recommendation would be to add a water sensor module that will allow the washing machine simulation to monitor water levels during each cycle, providing valuable insights into water consumption and potential leak detection.

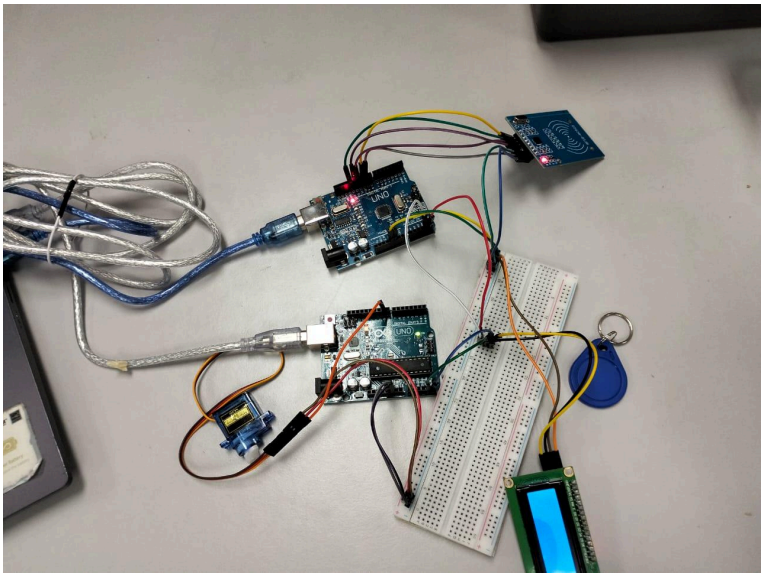
Conclusion

In conclusion, the objective to integrate Arduino with components to operate a washing machine was successfully achieved. We were able to integrate 2 Arduino and let them communicate with each other as Master and Slave. The effectiveness of RFID technology in user identification and control access has proven through the door part of this project. Moreover, integration of the servo motor acts as a representation to the real life system. Integration of input and outputs in the washing machine part has showcased the success of master and slave communication between Arduino. The inclusion of push button for manual function and bluetooth signal transfer for automatic function display the effectiveness of this project.

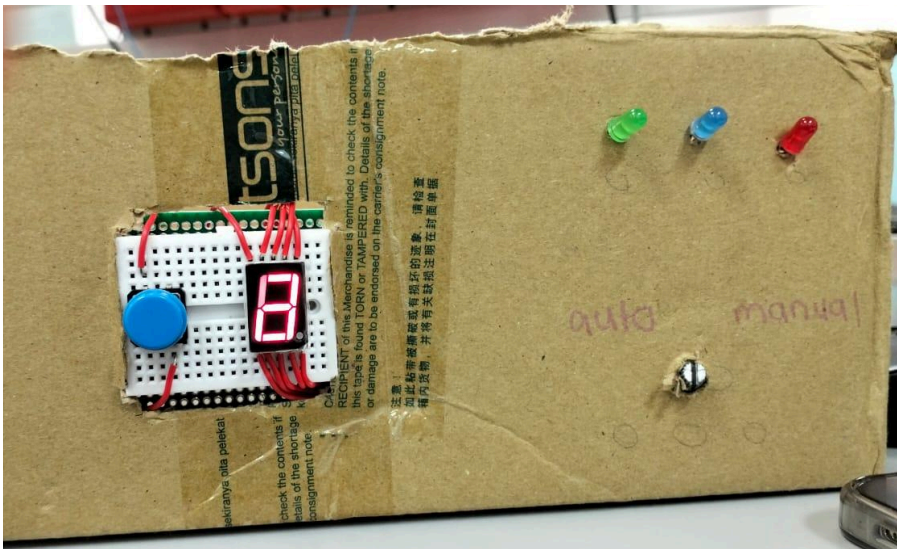
However, a number of issues occurred throughout this project, risking the functionality and performance of the system. It would enhance our system and provide us with practical experience in debugging and troubleshooting electronic circuits. We would be able to integrate several inputs and outputs into our ideas in the future with this experience, as well as by improving our abilities and expertise in this area.

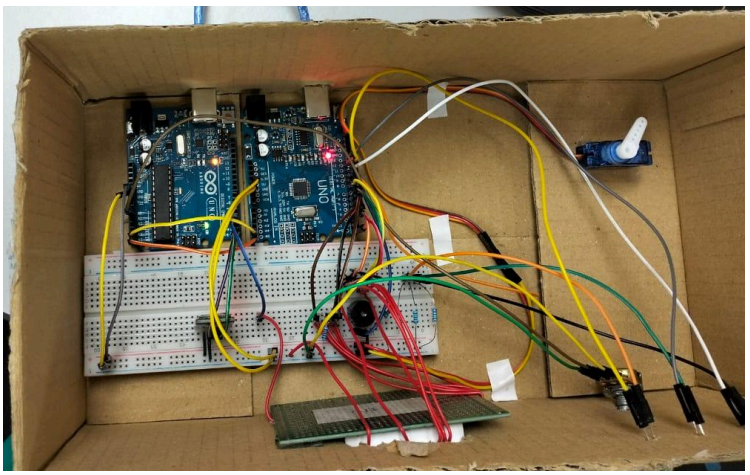
Appendix

Door Part



Washing Machine Part





Application used for the Automatic Washing Wachine

Arduino Bluetooth Controller - Apps on Google Play

Videos to Group A's Mini Project

Part 1: https://youtu.be/9bjdI62r7go

Part 2: https://youtu.be/Pl7b0fcwkIY

References

Zulkifli. (2014). Mechatronics Interfacing Lab Manual, (Rev. ed.). Unpublished Class Materials

Websites Link:

1. Arduino Website. February 8, 2018. Master Writer/Slave Receiver **https://wiki-content.arduino.cc/en/Tutorial/LibraryExamples/MasterWriter**

Acknowledgments

We would like to express our sincere gratitude to the individuals who provided invaluable assistance, guidance, and support during this experiment. First and foremost, we extend our appreciation to Assoc. Prof. Dr. Zulkifli Bin Zainal Abidin for his comprehensive instruction and mentorship throughout the mini project. Their insights, feedback, and enthusiasm played an important role in our understanding of Arduino programming.

Our fellow group members also deserve special acknowledgment for their collaboration and support. Our discussions, knowledge-sharing, and problem-solving sessions greatly enriched our understanding of this experiment's concepts and enhanced the overall learning experience. The collective contributions of our group members have not only enriched our learning experience but have also significantly contributed to the successful completion of this project.

Certificate of Originality and Authenticity

This is to certify that we are **responsible** for the work submitted in this mini project 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 **read** and **understand** the content of the report and no further improvement on the report is needed from any of the individual's contributions 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	AISYAH SOFEA OTHMAN	**Read** ✅
Matric Number	2115386	**Understand** ✅
Contribution	-Circuit connection -Coding - Abstract - Experiential Setup (door) - Methodology (door) - Material & Equipment	**Agree** ✅

Signature		
Name	SITI ALIAA BINTI IBRAHIM	**Read** ✅
Matric Number	2112618	**Understand** ✅
Contribution	-Circuit connection -Coding - Conclusion - Result	**Agree** ✅

Signature		
Name	ADLI FAHMI	**Read** ✅
Matric Number	2113095	**Understand** ✅
Contribution	-Circuit connection and design -Coding - Data analysis - Data collection	**Agree** ✅

Signature	*ethar*	
Name	ETHAR OSMAN	**Read** ✅
Matric Number	2111282	**Understand** ✅
Contribution	-Circuit connection -Coding - Introduction - Recommendation	**Agree** ✅

Signature	*ainaa*	
Name	NURAIN AINAA AQILAH BINTI ROSLI	**Read** ✅
Matric Number	2114560	**Understand** ✅
Contribution	- Experimental Setup : Washing Machine - Methodology : Washing Machine - Discussion - Circuit connection - Coding - Proofreading	**Agree** ✅