CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

EcoBin

Transforming the Way We Dispose

GROUP A-7

Aldrian Raffi Wicaksono	**2106653256**
Daffa Anis Fahrizi	**2106731365**
Rakha Argya Zahran	**2106733925**
M. Farrel Athaillah N.	**2106733875**

PREFACE

This project report has been prepared as a part of the final project requirements for the Cyber-Physical System studies at Universitas Indonesia. This report documents the culmination of our project titled "EcoBin: An Environmentally Friendly Automated Trash Bin System." and has been undertaken from the 1st of May 2023 to the 16th of May 2023 with the goal of innovating and revolutionizing daily waste disposal practices. It explores the development and implementation of an automated trash bin system that integrates a HC-SR04 distance sensor and an Arduino microcontroller. The EcoBin system aims to address the inefficiencies, inconveniences, and limited awareness surrounding waste management in our society.

The motivation behind the EcoBin project stems from the need to change and erase the inconveniences and inefficiencies posed by traditional waste disposal systems. Current methods of daily waste disposal in Indonesia often require manual operation, leading to potential hygiene and sanitary issues. Additionally, there is a lack of awareness among individuals about waste disposal and its impact on the environment. This project seeks to bridge these gaps by providing an eco-friendly alternative that automates the process of opening and closing the bin, enhancing convenience for users, and promoting waste processing awareness.

We recognize that this report may contain errors or mistakes, in the form of language, grammar, and content. Therefore, we humbly apologize for the imperfections of our project and welcome feedback and constructive criticisms, as we consider it a valuable opportunity for evaluation and improvement. It is our sincere hope that this final project report will serve as a valuable contribution to the field of cyber-physical systems.

Depok, May 15th, 2023

Group A-7

TABLE OF CONTENTS

CHAPTER 1 ..... **4**
INTRODUCTION **4**
1.1 PROBLEM STATEMENT 4
1.3 ACCEPTANCE CRITERIA 5
1.4 ROLES AND RESPONSIBILITIES 5
1.5 TIMELINE AND MILESTONES 5
CHAPTER 2 ..... **7**
IMPLEMENTATION **7**
2.1 HARDWARE DESIGN AND SCHEMATIC 7
2.2 SOFTWARE DEVELOPMENT 7
2.3 HARDWARE AND SOFTWARE INTEGRATION 8
CHAPTER 3 ..... **9**
TESTING AND EVALUATION **9**
3.1 TESTING 9
3.2 RESULT 9
3.3 EVALUATION 10
CHAPTER 4 ..... **11**
CONCLUSION **11**

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In this modern day world, the current day-to-day waste disposal systems heavily rely on manual operation, requiring individuals to physically open the trash bin lid, whether using their hands or their feet. This process not only consumes time, but also creates inconvenience and gives a suboptimal user experience, leading to discomfort and dissatisfaction. The manual operation required to open the bin lid can be cumbersome and unhygienic. Moreover, individuals may hesitate to dispose of waste properly if they perceive the process as inconvenient or unpleasant. Therefore, there is a need for a more comfortable and user-friendly waste disposal system.

1.2 PROPOSED SOLUTION

To address the aforementioned challenges, we propose the development of the EcoBin, an innovative and environmentally friendly automated trash bin system. The EcoBin will be powered by a distance sensor (HC-SR04) and an Arduino microcontroller, enabling automation and enhanced user experience. When a person approaches the bin, the distance sensor will detect their presence and send a signal to the Arduino. The Arduino will then activate a servo motor, automatically opening the bin lid. This automation eliminates the need for manual operation, significantly improving the efficiency and convenience of waste disposal.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. The HC-SR04 must be able to detect the presence of an object within its programmable proximity range of 10 centimeters.
2. The LED indicator must activate when the HC-SR04 sensor detects an object within its programmable proximity range of 10 centimeters.

3. The servo motor must be capable of moving and holding the EcoBin trash lid during the entire duration when the HC-SR04 detects an object within its proximity range.
4. The EcoBin trash lid must be capable of being opened downwards by the servo motor in response to the detection of an object by the HC-SR04 sensor.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Proteus Schematic Designer	Creating the schematic design of the hardware components using Proteus software and ensuring the components connections and functionality	Rakha Argya Zahran
Assembly and Arduino Developer	Developing the assembly code to control and implement the intended logic to the Arduino board along with its distance sensor, servo motor, and other hardware components.	Aldrian Raffi Wicaksono, Daffa Anis Fahrizi, Rakha Argya Zahran
Hardware Assembler	Physically assembling and connecting the hardware components such as the sensors, microcontrollers, jumper cables, etc.	Aldrian Raffi Wicaksono, Daffa Anis Fahrizi, M. Farrel Athaillah N.
Report Document, and Powerpoint Author	Create comprehensive reports and presentations	Aldrian Raffi Wicaksono, M. Farrel Athaillah N.

	documenting the project's description, progress, and outcomes	

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

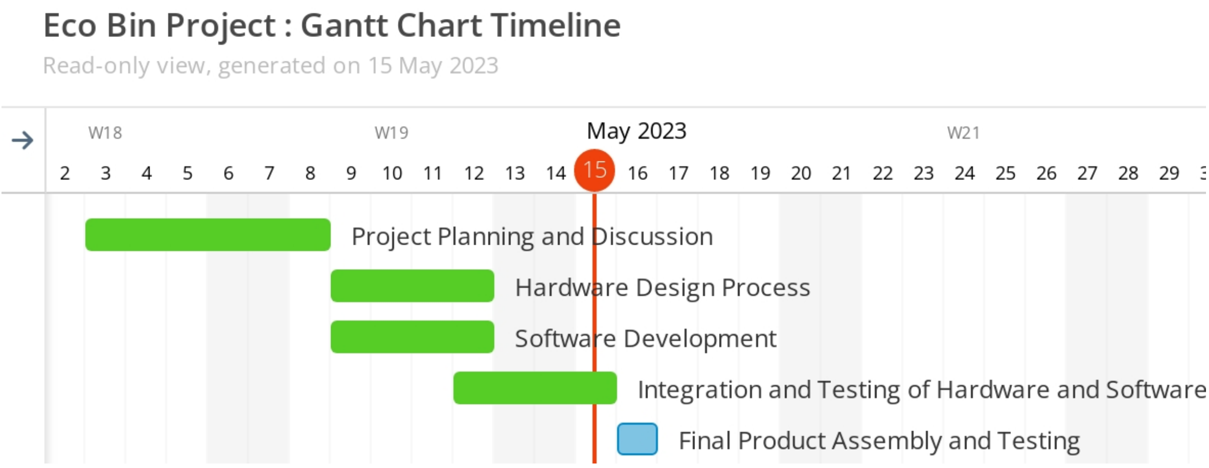


Fig 1. Project Timeline

- Project Planning and Discussion = 3rd - 8th of May 2023
- Hardware Design Process = 9th - 12th of May 2023
- Software Development = 9th - 12th of May 2023
- Integration and Testing of Hardware and Software = 12th - 15th of May 2023
- Final Product Assembly and Testing = 15th - 17th of May 2023

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

In the process of designing and compiling the hardware components used in the EcoBin project, we utilize various components with their unique purposes. Those components are as follows:

a. A mini breadboard base
b. 2x Arduino microcontroller
c. 1x HC-SR04 distance sensor
d. 1x Servo motor
e. 1x LED indicator
f. Trash bin
g. A set of jumper wires
h. 2 power source

Our design process utilized the Proteus software to create the design and schematics diagram. We placed and connected components such as the HC-SR04 sensor, servo motor, button, LED indicator, and power supply to ensure proper functionality and compatibility inside the Proteus simulator

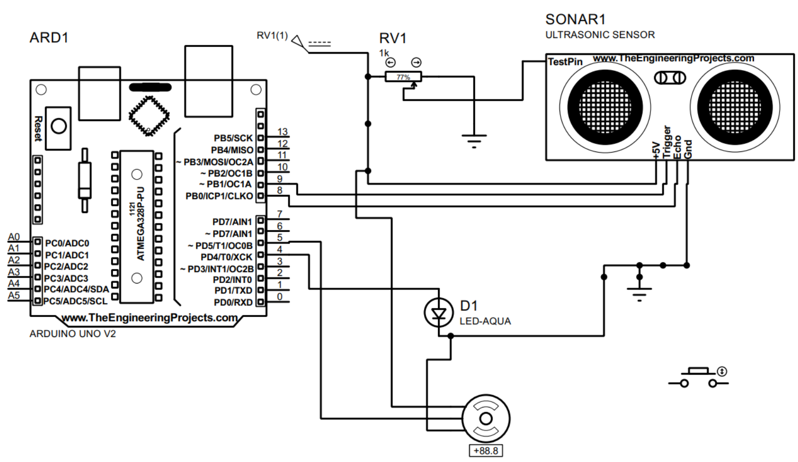


Fig 2. Proteus Schematics

Within Proteus, we incorporated libraries obtained from the internet for Arduino and the ultrasonic sensor. Prior to implementing the code, we converted the .S file into .hex format and imported it into Proteus. To simulate the distance measurement functionality of the ultrasonic sensor, we replaced it with a test pin connected to a variable potentiometer, allowing us to adjust the simulated distance.

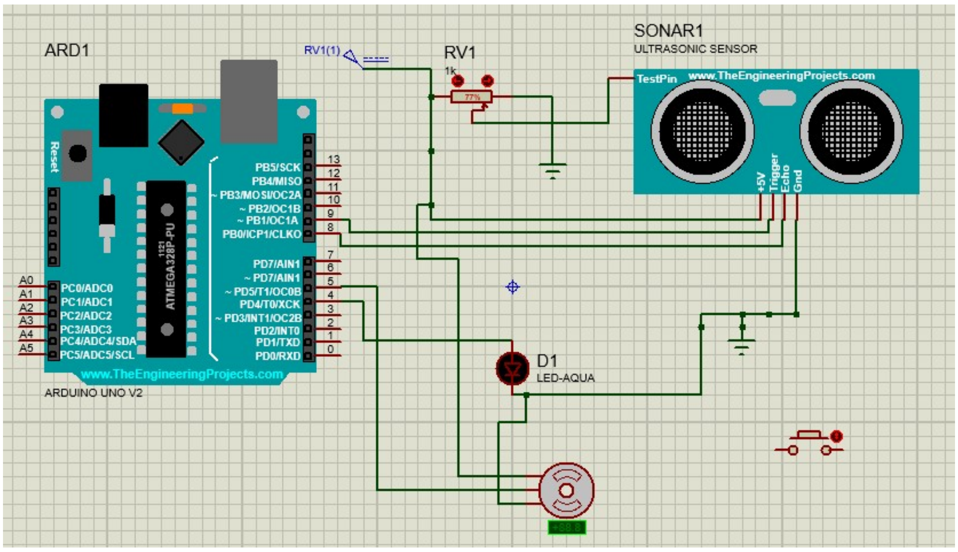


Fig 3. Proteus Design

2.2 SOFTWARE DEVELOPMENT

During the software development phase, we utilized the Arduino application for programming and uploading code to the Arduino Uno microcontroller used in our Ecobin project. This software development process involved writing and refining code to control the various components of the Ecobin system. To work with assembly code, the Arduino application can be installed by following the link provided: https://www.arduino.cc/en/software. This allowed us to leverage the powerful features and user-friendly interface of Arduino for efficient programming and development.

The core of the code comprises several subroutines that enable the control of essential components such as the HC-SR04 ultrasonic sensor, SG90 servo motor, button, and delay functionalities. These subroutines are designed to handle specific tasks and interactions between the hardware and software components. For instance, the main function serves as the program's central hub, executing a sequence of actions repeatedly. It includes calls to the

HC_SR04_sensor function for distance measurement and uses the RJMP instruction to return to the main function, ensuring a continuous loop of operations.

```
HC_SR04_sensor:
;--------------
    SBI DDRD, 4
    SBI   DDRB, 1         ;pin PB1 as o/p (Trigger)
    CBI   DDRB, 0         ;pin PB0 as i/p (Echo)
    ;-----------------------------------------------------------
    SBI   PORTB, 1
    RCALL delay_timer0
    CBI   PORTB, 1        ;send 10us high pulse to sensor
    ;-----------------------------------------------------------
    RCALL echo_PW         ;compute Echo pulse width count
    ;-----------------------------------------------------------
    RCALL byte2decimal    ;convert & display on MAX7219
    ;-----------------------------------------------------------
    RCALL delay_ms
    RET
```

```
;-----------------------
; Assembly Code
;-----------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"

;-----------------------
.global main
main:
RCALL HC_SR04_sensor
RJMP main

;===============================================================

HC_SR04_sensor:
;--------------
    SBI DDRD, 4
    SBI   DDRB, 1         ;pin PB1 as o/p (Trigger)
    CBI   DDRB, 0         ;pin PB0 as i/p (Echo)
    ;-----------------------------------------------------------
    SBI   PORTB, 1
    RCALL delay_timer0
    CBI   PORTB, 1        ;send 10us high pulse to sensor
    ;-----------------------------------------------------------
    RCALL echo_PW         ;compute Echo pulse width count
    ;-----------------------------------------------------------
    RCALL byte2decimal    ;convert & display on MAX7219
    ;-----------------------------------------------------------
    RCALL delay_ms
    RET
```

```
;==============================================================
echo_PW:
;-------
    LDI   R20, 0b00000000
    STS   TCCR1A, R20     ;Timer 1 normal mode
    LDI   R20, 0b11000101 ;set for rising edge detection &
    STS   TCCR1B, R20     ;prescaler=1024, noise cancellation ON
    ;-----------------------------------------------------------
lh1: IN    R21, TIFR1
    SBRS  R21, ICF1
    RJMP  lh1              ;loop until rising edge is detected
    ;-----------------------------------------------------------
    LDS   R16, ICR1L      ;store count value at rising edge
    ;-----------------------------------------------------------
    OUT   TIFR1, R21      ;clear flag for falling edge detection
    LDI   R20, 0b10000101
    STS   TCCR1B, R20     ;set for falling edge detection
    ;-----------------------------------------------------------
lh2: IN    R21, TIFR1
    SBRS  R21, ICF1
    RJMP  lh2              ;loop until falling edge is detected
    ;-----------------------------------------------------------
    LDS   R28, ICR1L      ;store count value at falling edge
    ;-----------------------------------------------------------
    SUB   R28, R16        ;count diff R22 = R22 - R16
    OUT   TIFR1, R21      ;clear flag for next sensor reading
    RET
```

```
byte2decimal:
;------------
      CLR   R26              ;set counter1, initial value 0
      CLR   R27              ;set counter2, initial value 0
      ;---------------------------------------------------------
lh70:  CPI   R28, 100         ;compare R28 with 100
Ret:  BRMI  lh80              ;jump when R28 < 100
      INC   R26              ;increment counter1 by 1
      SUBI  R28, 100         ;R28 = R28 - 100
      RJMP  lh70
      ;---------------------------------------------------------
lh80:  CPI   R28, 10          ;compare R28 with 10
      BRMI  dsp              ;jump when R 28 < 10
      INC R27 ;increment counter2 by 1
SUBI R28, 10 ;R28 = R28 - 10
RJMP lh80
;----------------------------------------------------------
dsp:
CPI R27, 3
BRSH mati_led ; If distance is greater than 30 cm, branch to no
CPI R27, 2
BRLO servo_led ; If distance is less than 10 cm, branch to servo
continue:
RET

; Kode nyalain LED
servo_led:
SBI PORTD, 4
RCALL servo_control ; Call the servo control subroutine
RJMP continue
```

```
; Kode matiin LED
mati_led:
CBI PORTD, 4
RJMP continue

; Servo control subroutine
servo_control:
SBI DDRD, 5 ; Set servo control pin (e.g., PD5) as output
LDI R30, 90  ; Set servo angle to 90 degrees ; Move the servo to a specific angle (adjust values as per servo specifications)
LDI R31, 0   ; Set upper byte of the servo angle to 0 (if necessary)

; Loop to generate servo control pulse
l1: SBI PORTD, 5   ; Set servo control pin HIGH
    RCALL delay_timer0  ; Delay for pulse width (adjust as per servo specifications)
    CBI PORTD, 5   ; Set servo control pin LOW
    RCALL delay_timer0  ; Delay for remaining pulse period (adjust as per servo specifications)
    DEC R30        ; Decrement servo angle by 1
    BRNE l1        ; Loop until the servo reaches the desired angle
RET

```

```
;==============================================================
; Delay subroutines
;==============================================================
delay_timer0: ;10 usec delay via Timer 0
;------------
CLR R20
OUT TCNT0, R20 ;initialize timer0 with count=0
LDI R20, 20
OUT OCR0A, R20 ;OCR0 = 20
LDI R20, 0b00001010
OUT TCCR0B, R20 ;timer0: CTC mode, prescaler 8
;---------------------------------------------------------
l0: IN R20, TIFR0 ;get TIFR0 byte & check
SBRS R20, OCF0A ;if OCF0=1, skip next instruction
RJMP l0 ;else, loop back & check OCF0 flag
;---------------------------------------------------------
CLR R20
OUT TCCR0B, R20 ;stop timer0
;---------------------------------------------------------
LDI R20, (1<<OCF0A)
OUT TIFR0, R20 ;clear OCF0 flag
RET

```

```
;==============================================================
delay_ms: ;delay 0.5s
LDI R21, 255
ls7 :LDI R22, 255
ls8 :LDI R23, 41
l9 :DEC R23
BRNE l9
DEC R22
BRNE ls8
DEC R21
BRNE ls7
RET
```

Within the HC_SR04_sensor function, instructions are implemented to set up the appropriate input/output (I/O) pins on the microcontroller, establishing the necessary connections with the HC-SR04 sensor. Following the pin configuration, a high pulse signal is sent to the sensor for precisely 10 microseconds. The echo_PW function is then invoked to calculate the duration of the received pulse from the HC-SR04 sensor, which provides the distance measurement. This value is further processed and displayed using the byte2decimal function, facilitating the conversion and presentation of the data in a user-friendly format.

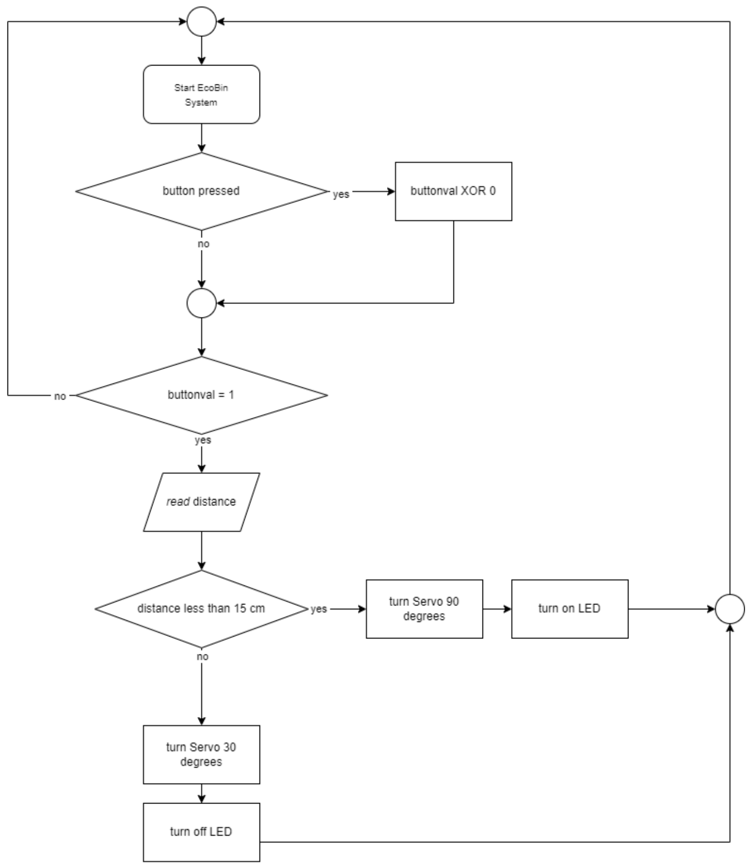


Fig 4. Program Flowchart

Additional subroutines, such as those for timer-based delays and delays in milliseconds, contribute to the functionality of the code. These subroutines ensure precise timing and synchronization between the various components of the Ecobin system. By implementing these control mechanisms, the code enables effective interaction with the HC-SR04 sensor, allowing for the appropriate response and manipulation of the LED and servo based on the detected distance. The software development process for the Ecobin project involved careful consideration of hardware integration, efficient code execution, and reliable performance to achieve the desired functionality and user experience.

2.3 HARDWARE AND SOFTWARE INTEGRATION

Upon the completion of the hardware and software development phases, the integration process of these components involves carrying out a set of crucial procedures, which includes:

1. Connecting the HC-SR04 distance sensor to the Arduino board:
   a. Connect the VCC pin of the sensor to the 5V port of the Arduino
   b. Connect the Ground pin of the sensor to the GND port of the Arduino
   c. Connect the trigger pin of the sensor to the DDRD1 port of the Arduino
   d. Connect the echo pin of the sensor to the DDRD0 port of the Arduino
2. Connecting the LED indicator to the Arduino board:
   a. Connect the cathode pin to the ground connection of the board
   b. Connect the anode pin to the DDRD4 port of the Arduino
3. Connecting the servo motor to the Arduino board:
   a. Connect the VCC pin of the motor to the 5V port of the Arduino
   b. Connect the Ground pin of the motor to the GND port of the Arduino
   c. Connect the PWM pin of the motor to the DDRB5 port of the Arduino

After successfully connecting and integrating all of the essential components as followed above, we have successfully completed the integration phase of the EcoBin project. The hardware and software components have been interconnected, allowing for extensive testing, refinement, and further development processes.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

The testing phase of our EcoBin project started by testing the integration of the software program and the hardware prototype. This trial's objective is to check and see whether or not the components used are suitable and sufficient for the project. After connecting the relevant hardware components with the program, our first trial resulted in our first error, which was the inability of the LED to indicate whether or not the HC-SR04 detected any object in its proximity. This was swiftly realized to be a software error where the program was stuck in a loop.

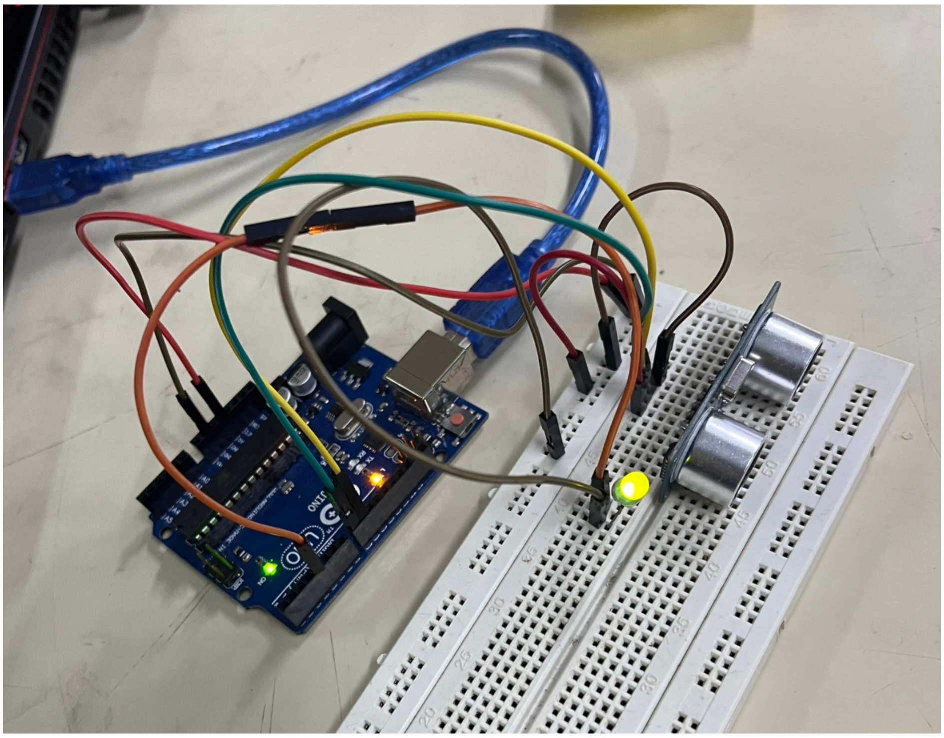


Fig 5. First Testing Trial: Continuous LED Illumination Error

After fixing the issue and polishing the hardware connections, we've encountered another problem, which is the lack of motion and movement from the servo motor after it's connected. After troubleshooting both the hardware and software side of the project, we've encountered where the problem lies, which is in the power requirements the hardware components needed.

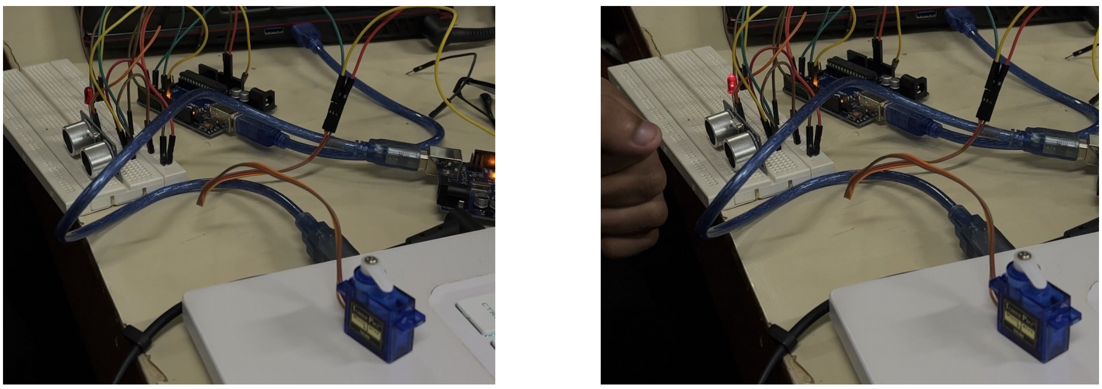


Fig 6 & 7. Second Testing Trial: Lack of Servo Motor Movement

To fix this problem, we've decided to properly use an external power supply for the second Arduino board which is connected to the servo motor. This solves our problem and moves the servo motor in accordance to its purpose.

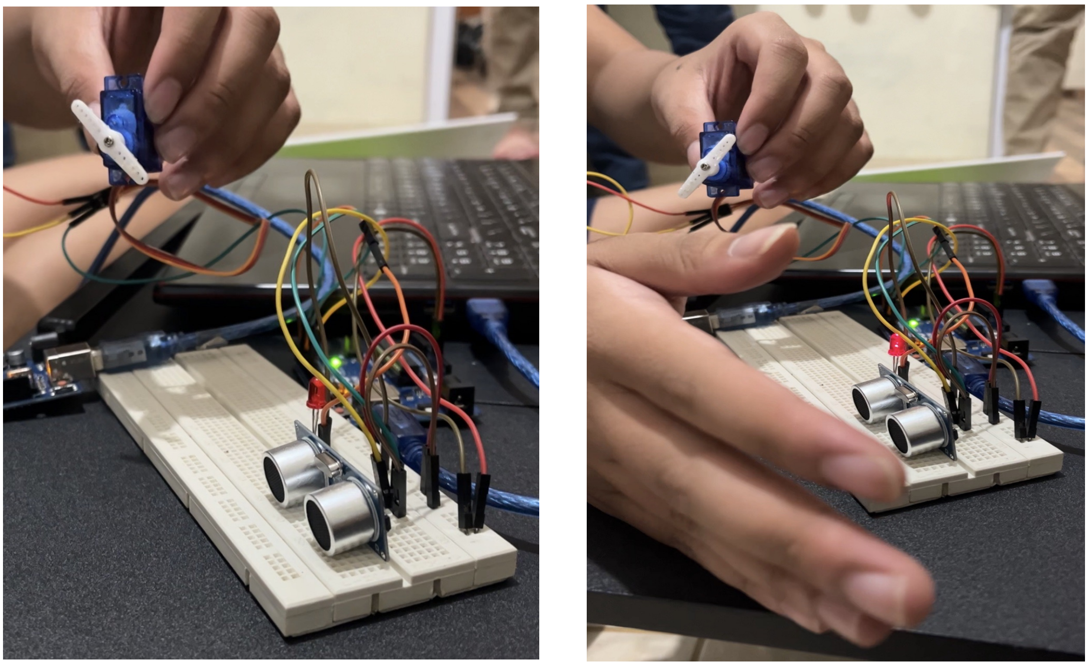


Fig 8 & 9. Third Testing Trial

In our third testing trial, we've successfully activated the servo motor to rotate in motion when the HC-SR04 sensor detects an object within its programmed proximity range. The LED indicator will also illuminate in response to the detection of an object from the sensor.

3.2 RESULT

To demonstrate the result of the EcoBin project, we've tested the functionality of the EcoBin electronic device by constructing and connecting the servo motor to a trash bin lid and test the practicality of said components.

Fig 10 & 11. Final Testing Result

3.3 EVALUATION

Upon evaluating the development process of the Ecobin project, we have identified an area that requires further improvement, specifically in servo control. Currently, our SG90 Servo is limited to moving in one direction and lacks the concept of degree control. This limitation arises from our team's limited competency in assembly programming, preventing us from finding a solution to enable the servo to turn in the reverse direction. Addressing this issue will be crucial to enhance the functionality and versatility of the Ecobin system.

Conversely, the HC-SR04 sensor has proven to perform exceptionally well throughout the project. It reliably measures distances and promptly detects objects within a range of less than 10 cm. The sensor efficiently sends signals to the Arduino microcontroller, enabling timely and accurate responses to the detected objects. Its reliable performance has contributed significantly to the overall functionality of the Ecobin project.

Moving forward, we recognize the importance of addressing the servo control limitation to optimize the Ecobin's capabilities. To overcome this challenge, we will focus on enhancing our assembly programming skills and exploring alternative servo control solutions. By developing a comprehensive understanding of servo control principles and leveraging our expertise in Arduino programming, we aim to enable bidirectional movement and precise degree control for the servo motor. This improvement will enhance the Ecobin's functionality and allow for more efficient lid opening and closing operations.

In conclusion, while the HC-SR04 sensor has exhibited excellent performance, we acknowledge the need for further development in servo control for the Ecobin project. By investing efforts into improving our assembly programming skills and finding innovative solutions, we are determined to overcome the servo's current limitations. This ongoing evaluation and enhancement process will contribute to realizing the project's objectives and delivering a more refined and efficient Ecobin system.

CHAPTER 4

CONCLUSION

The EcoBin project presents a practical solution for waste management in households and public spaces. The smart trash bin, equipped with sensors and a microcontroller, can revolutionize the traditional waste disposal system by promoting efficiency and sustainability. By using its HC-SR04 proximity sensors, the EcoBin intelligently detects the presence of individuals approaching and automatically opens its lid, providing a touchless and convenient experience for users to dispose their trash. This primary feature enhances hygiene and minimizes the risk of contamination / unsanitary physical contact, especially in public areas.

The EcoBin project showcases the potential of technology in creating smarter and more sustainable waste management systems. By streamlining daily waste disposal processes while also promoting recycling, the EcoBin contributes to a cleaner environment and a greener future.

Overall, the EcoBin project highlights the importance of innovative solutions in addressing environmental challenges. Through the combination of smart sensors and system automation. the EcoBin offers a practical and effective way to manage waste and encourages responsible waste disposal practices in our communities.

REFERENCES

[1] “Assembly via Arduino (part 21) - HC SR04 sensor,” YouTube, https://www.youtube.com/watch?v=dIIBG7Qz-_E (accessed May 16, 2023).

[2] Ultrasonic ranging module : HC-SR04 - electroschematics, https://www.electroschematics.com/wp-content/uploads/2013/07/HC-SR04-datasheet-version-2.pdf (accessed May 16, 2023).

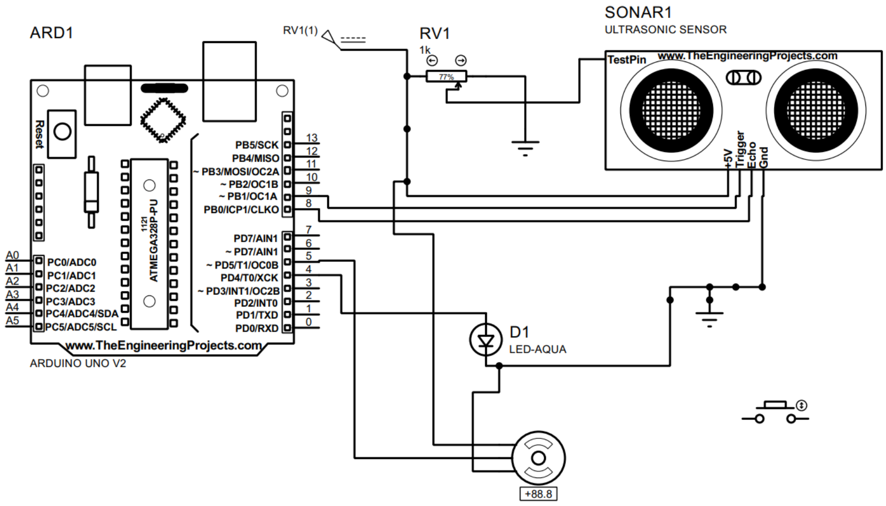
[3] “Servo Motor SG-90,” Components101, https://components101.com/motors/servo-motor-basics-pinout-datasheet (accessed May 16, 2023).

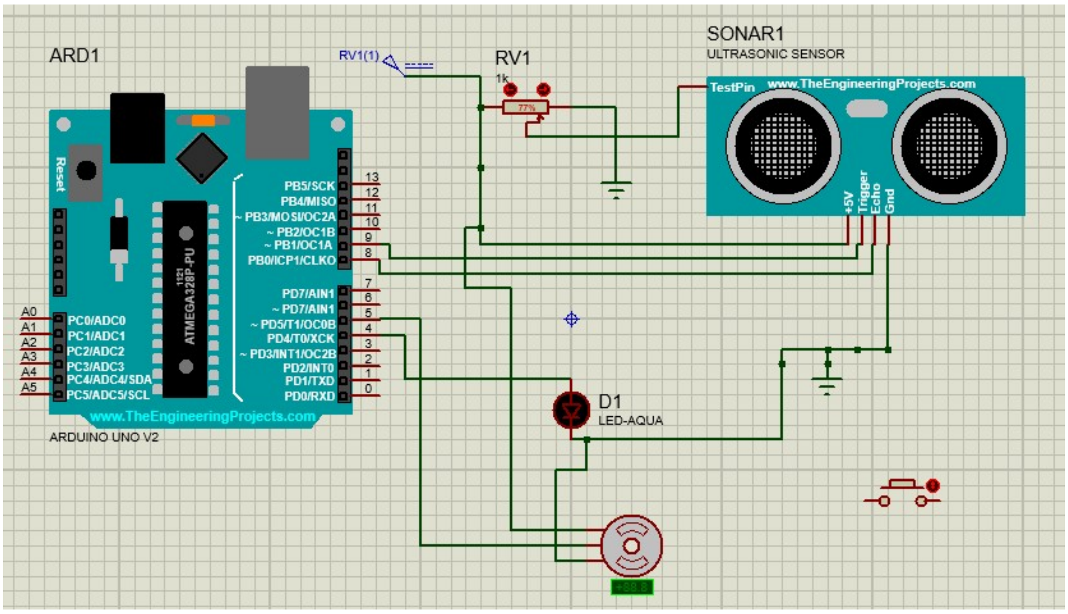
[4] Arduino, https://content.arduino.cc/assets/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (accessed May 16, 2023).

[5] "Assembly via Arduino - Programming HC-SR04 sensor," Anas Kuzechie Projects, https://akuzechie.blogspot.com/2021/12/assembly-via-arduino-programming-hc.html (accessed May 16, 2023).

APPENDICES

Appendix A: Project Schematic





Appendix B: Documentation