EcoBot: A Smartphone-Controlled Robot for Indoor Waste Collection

A Programming Embedded System presented to the faculty of

School of Engineering and Technology

In partial fulfillment of the requirements in the subject

PEMBEDS – PROGRAMMING EMBEDDED SYSTEMS

Leading to the degree of

Bachelor of Science in Information Technology

Presented by:

Abellar, Jonel John Darwin G.
Delos Santos, Adrian S.
Rabanal, Nathaniel Cedric G.
Ramirez, Levi Wesley F.

Submitted to:

Mr. Warren B. Panizales

Instructor

April 2025

Table of Contents

Abstract

Household cleaning can be tedious, especially when it involves picking up small waste items like paper scraps, food wrappers, or packaging. While traditional tools like brooms and vacuum cleaners are effective, they require manual effort and constant attention. To address this, the

project introduces **EcoBot**, a manually controlled indoor robot designed to assist with the collection of small household waste.

EcoBot is powered by an Arduino Uno R3 microcontroller and is equipped with a robotic grabber arm operated by servo motors. It features a built-in waste bin to store collected litter. The robot is controlled remotely through a custom Android application developed using Android Studio, allowing users to move and operate the robot directly from their smartphone.

This project focuses on enabling users to control and guide the robot around their home to collect scattered waste. The mobile app provides an intuitive interface for navigation and controlling the grabber mechanism.

By combining robotics and mobile app development, this project demonstrates a practical and affordable solution for smart home cleaning. It supports Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure by integrating accessible technologies to solve real-world household problems.

1. The Problem and Its' Background

Keeping a home clean and organized can be time-consuming, especially when it comes to regularly picking up small pieces of waste. This task can be inconvenient for people with busy lifestyles, physical limitations, or large living spaces.

To help address this issue, this project introduces **EcoBot**, a manually controlled robot that assists in collecting small indoor litter. Instead of relying on autonomous navigation, the robot is operated through a dedicated smartphone app developed in Android Studio. The app allows users to control the robot's movement and activate its grabber arm to pick up waste and store it in the onboard trash bin.

This solution offers an affordable and hands-on approach to home cleaning, combining robotics and mobile technology. It simplifies minor cleaning tasks while promoting innovation in smart home tools. The project also aligns with Sustainable Development Goal (SDG) 9 by applying simple, scalable technology to improve everyday living.

1.1. Introduction

The demand for smart home solutions continues to rise, driving innovation in the fields of robotics. One of the most repetitive household tasks is cleaning, particularly collecting small items like wrappers, paper scraps, or food packaging that can clutter floors. Although robotic vacuum cleaners exist, they often cannot handle solid waste or physically pick up scattered items.

This project presents **EcoBot**, a smartphone-controlled indoor robot designed to assist users in collecting small household waste. Unlike autonomous or suction-based robots, EcoBot is manually operated through a mobile application built in Android Studio. The app enables the user to navigate the robot, control its robotic arm, and collect visible waste with precision.

The robot uses affordable components such as the Arduino Mega 2560, servo motors, and dc motors. A built-in waste bin allows collected trash to be stored during operation. The intuitive Android app serves as the control interface, giving users full control of the robot's actions in real time.

By focusing on manual operation through a smartphone, this project remains cost-effective and practical for household use. It demonstrates how accessible technology can address everyday problems and contributes to Sustainable Development Goal 9: Industry, Innovation, and Infrastructure.

1.2. Statement of the Problem

Maintaining cleanliness in indoor environments, especially in larger households or homes with children and pets, can be a repetitive and time-consuming task. Small pieces of litter such as paper scraps, food wrappers, and other lightweight waste often go unnoticed or are ignored, gradually accumulating and affecting the hygiene and appearance of living spaces. Manually collecting these items can become burdensome, particularly for individuals with busy schedules, mobility limitations, or physical disabilities.

While current robotic cleaning solutions like vacuum robots are available in the market, they primarily rely on suction technology and are not capable of recognizing and picking up solid waste items that cannot be vacuumed. These devices also lack the ability to visually identify specific objects as trash, making them ineffective for diverse indoor cleaning tasks that involve object handling.

The problem this project aims to solve is the lack of an affordable indoor robot that can be easily controlled to collect and store small solid waste items. Existing systems often lack the ability to physically pick up litter using a robotic arm and do not provide a convenient, user-friendly way to assist with minor household cleaning tasks.

This project introduces a practical solution: a manually controlled robot equipped with a robotic arm and an onboard trash bin, designed to assist in collecting household litter. Operated via a smartphone application, the system allows users to guide the robot around indoor spaces and pick up waste using the robotic arm. By focusing on direct control, this project offers a more hands-on, flexible alternative to traditional cleaning tools, addressing the limitations of current robotic cleaning systems.

1.3. Objectives

This project aims to develop a smartphone-controlled robot equipped with a robotic arm and a built-in trash bin to assist in collecting small household waste. By combining Arduino-based hardware with a custom Android mobile application, the project promotes practical, low-cost solutions for smart home cleaning and supports sustainable living. The following objectives outline the key goals of this project:

- To design and build a manually controlled indoor robot capable of collecting small pieces of waste.
- To develop a functional robotic arm that can pick up and deposit waste into an onboard trash bin.
- To program and configure an Arduino Mega 2560 to control motors, and the robotic arm.

- To design and develop a custom Android application using Android Studio to remotely control the robot's movement and arm functions via Bluetooth.
- To ensure the system operates reliably within an indoor environment and responds accurately to user commands.

1.4. Scope and Limitation

Scope

This project will focus on developing a functional prototype of a manually controlled indoor cleaning robot equipped with a robotic arm and an onboard trash bin

- The system will use an Arduino Mega 2560 to manage motors and servos.
- The robot will be operated using a custom Android mobile application developed in Android Studio, allowing real-time manual control over movement and arm operation.
- The project aligns with Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure by demonstrating a low-cost, innovative approach to small-scale indoor waste management through smart technology and mobile integration.

Limitations

- The robot will not operate autonomously; it requires user control via the Android mobile application for navigation and trash collection.
- The robotic arm's strength and precision are limited by the type and torque of servo motors used, restricting the size and weight of waste it can handle.
- The system is designed for flat, indoor environments and may not perform well on uneven surfaces or in cluttered spaces.

1.5. Significance of the Project

Promoting Practical Indoor Waste Management

This project addresses a common household issue—manual collection of small litter such as wrappers, paper bits, and similar waste—by introducing a manually operated cleaning robot. By allowing users to control a robotic system through a mobile application, the project offers a hands-on yet innovative method to maintain indoor cleanliness without relying on traditional tools.

Empowering Users with Accessible Technology

Through the integration of affordable components like the Arduino Mega, Servo Motors, and DC Motors, this project makes robotic solutions more accessible to the public. It serves as a steppingstone for students, hobbyists, and developers interested in exploring robotics, control systems, and mobile app integration.

Encouraging Robotics in Everyday Applications

This system demonstrates how robotics can be applied beyond industrial or research settings and into everyday life. Although it is manually controlled, the project showcases the potential of combining mechanical design with user interfaces to create functional home-assistant tools.

Mobile-Controlled Robotics Education

By using Android Studio to build a companion mobile app, the project introduces mobile development as a complementary field to robotics. This dual integration of hardware and software can be valuable in STEM education and can inspire future enhancements like semi-autonomous control or sensor-based smart features.

Contribution to Sustainable Development Goals

While not fully autonomous, the project still supports **Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure** by promoting small-scale, innovative technology solutions for practical problems. It highlights how low-cost systems can contribute to smarter living environments and encourages environmentally conscious behavior through technology.

2. Hardware Requirements

This chapter outlines the essential hardware components required to build a manually controlled indoor waste-collecting robot. Each component has been selected based on functionality, compatibility with the Arduino Mega, and overall cost-efficiency to ensure the robot performs its tasks effectively.

The hardware requirements include the main controller board (Arduino Mega), a Bluetooth module for wireless communication with a smartphone, a power supply system, motor drivers and wheels for movement, servo motors for operating the robotic arm, and a built-in trash bin to store collected waste. Each component is described in detail along with its role in the overall system.

By understanding the function and integration of each hardware element, this chapter provides a comprehensive overview of the robot's physical design, which supports its core features—manual navigation via mobile app and basic mechanical trash collection using a robotic arm.

2.1. List of all Hardware Components Used

Component	Model/Brand (if any)
Arduino Mega clone	Arduino Mega (Original or Clone)
1x L298N Motor Driver Modules	L298N Dual H-Bridge
4x DC Gear Motors (3-6V)	Generic 3-6V Gear Motors
4x Wheels (Rubber/Plastic)	Compatible with DC motors
2x Servo Motors	SG90 or MG996R (stronger torque)
Robotic Arm Kit	Acrylic, Aluminum Arm Kit, or DIY
Scooper	Plastic or any lightweight materials

Chassis Frame	Reinforced plastic frame or DIY
Mini Trash Bin (Built-in)	Plastic or metal custom container
Rechargeable Battery Pack (12V)	12V 2000–3000mAh Li-ion/LiPo
Jumper Wires & Breadboard	Generic
18650 Cell Battery Holder Case	Generic
PKCell 3.7V Lithium-ion NMC 18650 Battery	For power
LM2596	Generic
18AWG Wire	Generic
Wireless Bluetooth Module	HC-05 or HC-06

2.2. Components Description

Component	Description
Arduino Mega 2560	Main microcontroller to control sensors, motors, and the robotic arm.
L298N Motor Driver Module	Controls the 4 DC motors (2 motors per driver module).
4x DC Gear Motors (3-6V)	Drives all four wheels, providing high torque for mobility and stability.
4x Wheels (Rubber/Plastic)	Enables the robot to move across different indoor floor surfaces.
2x Servo Motors	Used to control the joints of the robotic arm.
Robotic Arm	Mechanized arm structure for reaching and collecting trash.
Scooper	Picks up waste items and moves them into the trash bin.
Chassis Frame	The structural frame that supports all components and 3 lbs load.
Mini Trash Bin (Built-in)	Trash container to store the collected waste.

Rechargeable Battery Pack (12V)	Main power source for motors and control units.
DC-DC Buck Step Down Converter (LM2596)	Supplies safe power to Arduino and servos.
Jumper Wires & Breadboard	For wiring and prototyping connections between modules.

3. Software Requirements

This chapter outlines the essential software components and development tools required to build and operate a manually controlled waste-collecting robot system. It focuses on the platforms, programming environments, and libraries used to program the Arduino microcontroller and develop mobile applications.

The chapter covers the following:

- **Arduino IDE**: The primary environment used to write, upload, and debug code for the Arduino Mega. It handles the logic for motor control, servo movement, and Bluetooth communication.
- **C/C++ Language**: The programming language used within the Arduino IDE to control hardware components such as motors, servos, and sensors.
- **Android Studio**: The official IDE for Android development, used to create a custom mobile application that allows users to manually control the robot’s movement and robotic arm using Bluetooth.
- **Java/Kotlin**: Programming languages used in Android Studio to develop the mobile control interface.

- **Bluetooth Communication Libraries**: Used both in Arduino and the Android app to facilitate real-time wireless communication between the app and the robot.

3.1. Software Development Environment (Arduino IDE)

For this project, the **Arduino IDE version 1.8.19** was used as the primary development environment. This version provides a stable and user-friendly interface for writing, compiling, and uploading code to the **Arduino Mega** microcontroller.

Key features of this development environment include:

- Support for the C/C++ programming language
- Built-in serials monitor for debugging and real-time data monitoring
- Compatibility with external libraries for servo control and Bluetooth communication
- Integration with various Arduino boards and components

The version of Arduino IDE that the Developers use is Arduino IDE 2.3.6 version because this is the latest and stable and suitable for our development process.

3.2. Any necessary libraries for sensors or communication protocols

The following libraries were used in the Arduino IDE to enable communication with sensors, actuators, and modules essential to the project:

- **Servo.h**

  Used to control the servo motors in the robotic arm for precise positioning and movement.

- **SoftwareSerial.h**

  Enables serial communication on digital pins other than the default RX and TX, necessary for interfacing with the Bluetooth module (e.g., HC-05 or HC-06).

4. Circuit Design

This chapter presents the detailed circuit design of the indoor waste-collecting robot. It includes a schematic diagram that illustrates how all electronic components are interconnected, such as the Arduino Mega 2560, motor driver, servo motors for the robotic arm, Bluetooth module for wireless control, and power supply. The design ensures each component functions in harmony to enable smooth operation of the robot. This section also explains how signals are transmitted between components and how power is distributed throughout the system. By examining the circuit layout, readers will gain a clearer understanding of the electrical framework behind the robot's functionality.

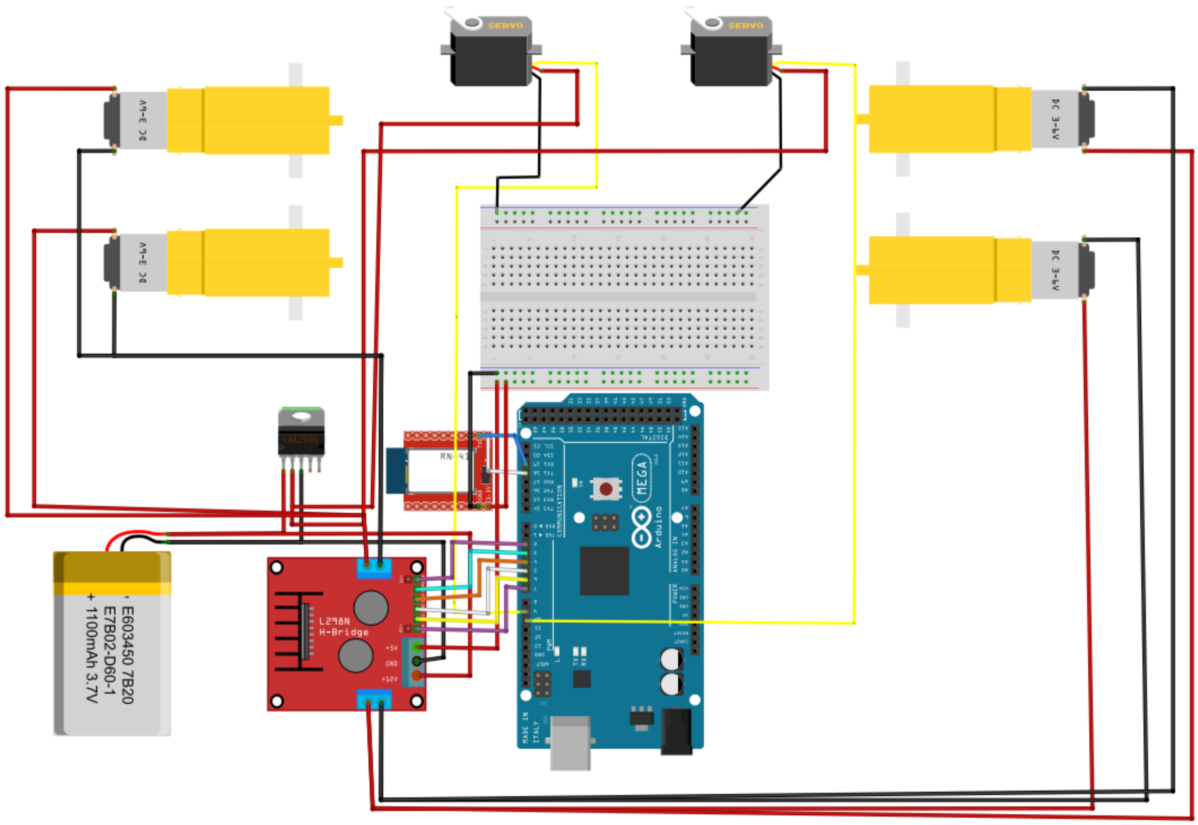

Figure 4.1 - Circuit Design

4.1. Schematic Diagram

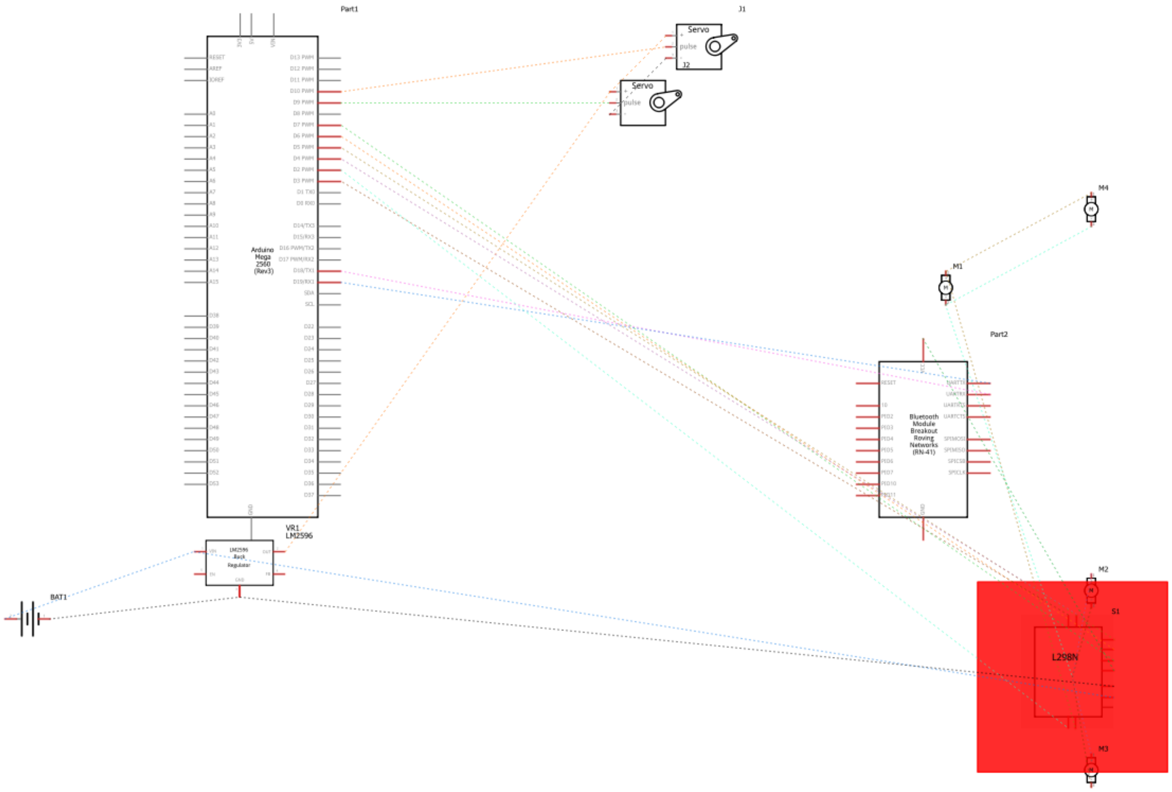

Figure 4.2 - Schematic Diagram

4.2. Connection Explanation

This section explains how each component in the system is connected and interacts with others in the Arduino-based robot setup:

Arduino Mega 2560

- Acts as the **central controller**.
- All components (motors, modules) are connected to it.
- It reads inputs (e.g., from Bluetooth) and controls outputs (e.g., servos, motors).

L298N Motor Driver Module

- Connected to **4 DC gear motors** (for wheel movement).
- Powered by a **battery pack** (connected to the motor driver's power terminals).
- **IN1–IN4 pins** connected to Arduino digital pins control motor direction.
- **EN pins** (Enable) control motor speed via PWM from the Arduino.
- Enables **forward, backward, and turning motions** of the robot.

DC Gear Motors (x4)

- Provide **locomotion**.
- Connected to the motor driver via output terminals (OUT1–OUT4).

- Controlled by Arduino through the motor driver.

Servo Motors (x2)

- Used to build the **robotic arm**.
- Each servo is connected to a separate **PWM-capable digital pin** on the Arduino.
- Powered via a separate VCC connected from the 12V battery to LM2596 and connected to a shared GND on the breadboard.
- Controlled for **lifting and dumping** trash into the onboard bin.

HC-05 Bluetooth Module

- Connects wirelessly to a **custom Android app** for manual control.
- **TX of HC-05 → RX of Arduino**, and **RX of HC-05 → TX of Arduino** (via voltage divider or level shifter if needed).
- Receives movement and arm control commands from the smartphone.

Battery Pack

- Powers the **L298N motor driver and LM2596,** which in turn powers the motors and servos.
- Also power the servos if current requirements are met (ensure enough current depending on the load).
- Arduino can be powered through the **VIN and GND** from the same pack.

Breadboard

- Used for **distributing power** (VCC and GND) to all motors and the Bluetooth module.
- Organizes the wiring and connections to keep the circuit tidy.

5. Arduino Code

This chapter presents the complete Arduino code used to control the smart indoor waste-collecting robot. It details the logic and functionality for the manual controls of the robot. Key components of the system include motor control for movement, and the servo-driven robotic arm mechanism for picking up trash. The code also features Bluetooth integration for manual control via a mobile device. Each function is carefully designed to coordinate mechanical actions, enabling the robot to perform efficiently within an indoor environment.

5.1. Complete Code Listing

```
#include <Servo.h>

#define ENA 7
#define IN1 6
```

```
#define IN2 5
#define ENB 2
#define IN3 4
#define IN4 3

Servo lifterLeft;
Servo lifterRight;

String inputString = "";
bool stringComplete = false;

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);

  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  lifterLeft.attach(9);
  lifterRight.attach(10);

  stopMotors();
}

void loop() {
  while (Serial1.available()) {
    char inChar = (char)Serial1.read();
    Serial.print(inChar);
    if (inChar == '\n') {
      stringComplete = true;
    } else {
      inputString += inChar;
    }
  }
```

```
  if (stringComplete) {
    processCommand(inputString);
    inputString = "";
    stringComplete = false;
  }
}

void processCommand(String command) {
  command.trim();
  Serial1.print("Received: ");
  Serial1.println(command);

  if (command.startsWith("F:")) {
    int speed = command.substring(2).toInt();
    moveForward(speed);
  } else if (command.startsWith("B:")) {
    int speed = command.substring(2).toInt();
    moveBackward(speed);
  } else if (command.startsWith("L:")) {
    int speed = command.substring(2).toInt();
    turnLeft(speed);
  } else if (command.startsWith("R:")) {
    int speed = command.substring(2).toInt();
    turnRight(speed);
  } else if (command.startsWith("S")) {
    stopMotors();
  } else if (command.startsWith("LFT:")) {
    int angle = command.substring(4).toInt();
    setLifterAngle(angle);
  }
}

void moveForward(int speed) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, speed);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENB, speed);
```

```
}

void moveBackward(int speed) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  analogWrite(ENA, speed);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENB, speed);
}

void turnLeft(int speed) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  analogWrite(ENA, speed);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENB, speed);
}

void turnRight(int speed) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, speed);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENB, speed);
}

void stopMotors() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, 0);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(ENB, 0);
}

void setLifterAngle(int angle) {
```

```
  lifterLeft.write(angle);
  delay(200);
  lifterRight.write(180 - angle);
}
```

5.2. Code Section Explanations

`setup()`

Initializes serial communication for both Bluetooth (`Serial1`) and the PC (`Serial`). Sets all motor control pins (ENA, `IN1`, `IN2`, ENB, `IN3`, `IN4`) as outputs. It also attaches the left and right lifter servos to their designated PWM pins (9 and 10), preparing them for movement. Finally, it calls `stopMotors()` to ensure the robot starts in a stopped state.

`loop()`

Continuously checks for incoming Bluetooth commands via `Serial1`. It reads characters into a buffer (`inputString`) until a newline character `\n` is received, signaling the end of a command. Once a full command is detected, it calls `processCommand()` to interpret and execute the instruction, such as moving the robot or adjusting the lifter.

`processCommand(String command)`

Parses and executes Bluetooth commands sent as strings. It trims whitespace and then matches the command prefix to determine what action to take:

- `"F:<speed>"`: Move forward at given speed (0–255)
- `"B:<speed>"`: Move backward
- `"L:<speed>"`: Turn left
- `"R:<speed>"`: Turn right
- `"S"`: Stop all motors

- `"LFT:<angle>"`: Move the lifting servos to a specific angle

Sends feedback through Bluetooth (`Serial1`) confirming the command received.

`moveForward(int speed)`

Drives both motors forward by setting their direction pins (`IN1`, `IN2`, `IN3`, `IN4`) appropriately and applying PWM speed to ENA and ENB.

`moveBackward(int speed)`

Drives both motors backward by reversing the direction pins and applying the specified speed.

`turnLeft(int speed)`

Turns the robot to the left by rotating the left motor backward and the right motor forward.

`turnRight(int speed)`

Turns the robot to the right by rotating the left motor forward and the right motor backward.

`stopMotors()`

Stops both motors by turning off all direction pins and setting PWM speed to 0 on both motors enable pins.

`setLifterAngle(int angle)`

Controls the lifter mechanism using two servo motors:

- Moves the **left servo** to the given `angle`
- Waits 200ms
- Moves the **right servo** to the **mirrored angle** (`180 - angle`), which helps achieve a symmetric lifting motion

This is used to raise, lower, or align the arms depending on the task (e.g., picking up or dropping trash).

6. Testing and Calibration

This chapter covers the procedures and results related to testing and calibrating the EcoBot's hardware and software components. It explains how motors and servos were tested to ensure accurate readings and proper functioning. Additionally, the chapter discusses adjustments made to motor speed and servo positions for smooth movement and precise trash handling. Overall, it ensures that the robot operates reliably and meets the desired performance standards.

6.1. Testing Procedures

This section outlines the step-by-step procedure used to test the assembled circuit and upload Arduino code to ensure proper functionality of the indoor trash-collecting robot.

Step 1: Initial Hardware Check

- Verify all connections on the breadboard or PCB, ensuring components such as the Arduino Mega, motors, Bluetooth module, and servo motors are properly wired.
- Check the power supply and battery connections for stable voltage output.

Step 2: Uploading the Arduino Code

- Connect the Arduino Mega 2560 to a computer via USB.
- Open the Arduino IDE (e.g., version 1.8.19).
- Load the final version of the program/code and verify for errors.
- Upload the code to the Arduino board successfully.

Step 3: Bluetooth Module Test

- Pair the Bluetooth module (e.g., HC-05) with the Android phone using the custom control app.
- Test communication by sending basic movement commands (e.g., forward, backward, left, right) from the app and observing the robot's response.

Step 4: Servo Motor Testing

- Observe whether the servo motors in the robotic arm move according to programmed sequences.
- Send control commands from the app to test picking and dropping gestures.

Step 5: Motor Driver and Movement Test

- Verify the robot's wheels respond to direction commands via the app.
- Test turning, speed control, and stopping features to ensure precise maneuverability.

Step 6: Trash Collection Test

- Place small test trash items in front of the robot.
- Use manual controls from the Android app to position the robot and activate the robotic arm to pick up and drop the item into the bin.

Step 7: Full System Integration Test

- Power on the entire system using the battery source.
- Use the Android app to test simultaneous movement and robotic arm control.

- Observe system behavior under typical operating conditions.

Step 8: Final Observations and Adjustments

- Record any issues such as response delay, incorrect motion, or servo misalignment.
- Adjust the code or mechanical structure accordingly.
- Retest after each change to confirm improvements.

7. Results and Discussion

In this chapter, you will find an analysis of the performance and outcomes of the EcoBot project during testing. It discusses how well the robot performed tasks such as movement and waste collection. The chapter highlights any challenges encountered, how the robot responded to different environments, and the effectiveness of manual mode. It also includes observations on motor control, and overall system reliability, providing insights into improvements and practical usability.

7.1. Observation and Findings

During the testing phase of the project, several important observations and results were noted. The Bluetooth communication between the Android application—developed in Android Studio and the robot, via the HC-05 Bluetooth module, was established successfully. The robot accurately responded to real-time commands such as forward, backward, left, right, and stop, with stable connectivity within a 5–10-meter indoor range. Manual navigation was smooth and responsive, allowing users to maneuver the robot toward trash effectively. The robotic arm, powered by servo motors, was able to pick up lightweight objects like paper and small plastic wrappers and deposit them into the onboard bin with reasonable accuracy. Adjustments were made to improve the arm's range of motion for better gripping precision.

System responsiveness was consistent, with all components including motors, servos, and the Bluetooth module working in synchronization. The power supply provided sufficient energy for continuous operation of around 20 to 30 minutes. The mobile app offered a user-friendly interface and allowed users to control both movement and robotic arm with ease. Visual feedback from the app confirmed successful command transmission to the robot.

Overall, the robot demonstrated reliable manual operation and effective indoor movement for waste collection tasks. However, it was found that the robotic arm performed best with lightweight items due to torque limitations. The system relies solely on Android devices for control, and compatibility was verified across multiple Android versions. These results confirm that the project prototype functions as expected and fulfills its core objectives under manual control.

7.2. Functionality Discussion

The project functions as an integrated manual waste collection system, where an Arduino-based robot is controlled via a custom-built Android application using Bluetooth communication. The robot is equipped with motors for movement, a robotic arm for picking up waste, and a built-in trash bin for storing collected items. Upon receiving commands from the Android app, the robot can move in different directions and operate the robotic arm to pick up lightweight trash, such as paper or plastic wrappers. This interaction between hardware and software allows for precise manual control, enabling users to navigate the robot toward waste and perform the collection task efficiently.

In real-world scenarios, this robot can be utilized in indoor environments such as schools, offices, homes, and hospitals to assist in maintaining cleanliness without requiring direct human contact with waste. It provides a safe and convenient way to collect trash in areas where sanitation and hygiene are important, or where manual collection might be difficult. Furthermore, this system has potential as a foundational platform for future enhancements, such as adding semi-autonomous features or IoT integration, making it a steppingstone toward more intelligent and sustainable robotic cleaning solutions.

7.3. Limitations and Improvements

One of the main limitations of the current project is its exclusive reliance on manual control through a mobile application. While this allows for user-directed operation, it limits the robot's independence and efficiency, especially in larger or more complex indoor environments. The robotic arm also has limited lifting capacity and precision due to the basic servo motors used, which may affect its ability to handle heavier or irregularly shaped waste. Additionally, the system is only compatible with Android devices, restricting its accessibility for users on other platforms.

In terms of improvements, a key future development would be the reintroduction of autonomous capabilities. This includes enabling the robot to navigate indoor spaces on its own using sensors for obstacle avoidance and potentially integrating a camera module for visual trash detection. These features would allow the robot to identify and collect waste without user intervention, significantly enhancing its practicality and application in real-world environments. Incorporating advanced microcontrollers, stronger actuators, and cross-platform mobile support would further increase the robot's functionality and usability.

8. Mobile Application Integration

The integration of a custom-built mobile controller application developed using Android Studio. This application serves as a user-friendly interface that allows users to remotely operate the robot via Bluetooth communication using the HC-05 module. The app features essential manual controls such as forward, backward, left, right, stop, and additional commands for adjusting speed and activating the trash-grabbing mechanism. Real-time command transmission ensures that the robot responds instantly to user input, enabling precise and responsive control. This chapter details the design of the app interface, the implementation of Bluetooth connectivity, and the programming logic behind each control function.

8.1. App Functionality

The mobile application developed for this project provides a range of essential features to facilitate real-time interaction with the robot. Designed with a simple and intuitive user interface, the app allows users to control the robot's movements—such as moving forward, backward, turning left or right, and stopping—through dedicated on-screen buttons. It also includes additional functions to adjust the speed of the motors, as well as adjusting the motion of the robot’s arm. The app connects to the robot wirelessly via Bluetooth, ensuring smooth command transmission using the HC-05 module. All actions initiated from the app are immediately reflected in the robot’s behavior, making the system responsive and efficient for remote control scenarios.

8.2. Communication Explanation

The communication between the Arduino and the mobile application is established through a Bluetooth wireless connection, specifically using the HC-05 Bluetooth module. This module acts as a serial communication bridge, allowing Arduino to send and receive data to and from the mobile app in real time. When the user interacts with the app’s control interface, commands are sent as simple character strings via Bluetooth. The Arduino continuously listens for incoming commands through its serial interface connected to the HC-05 module. Upon receiving a command, the Arduino processes it and executes the corresponding action, such as moving the robot or operating the grabber. This wireless communication protocol enables seamless and responsive control of the robot without the need for physical connections, enhancing user convenience and mobility.

8.3. User Manual

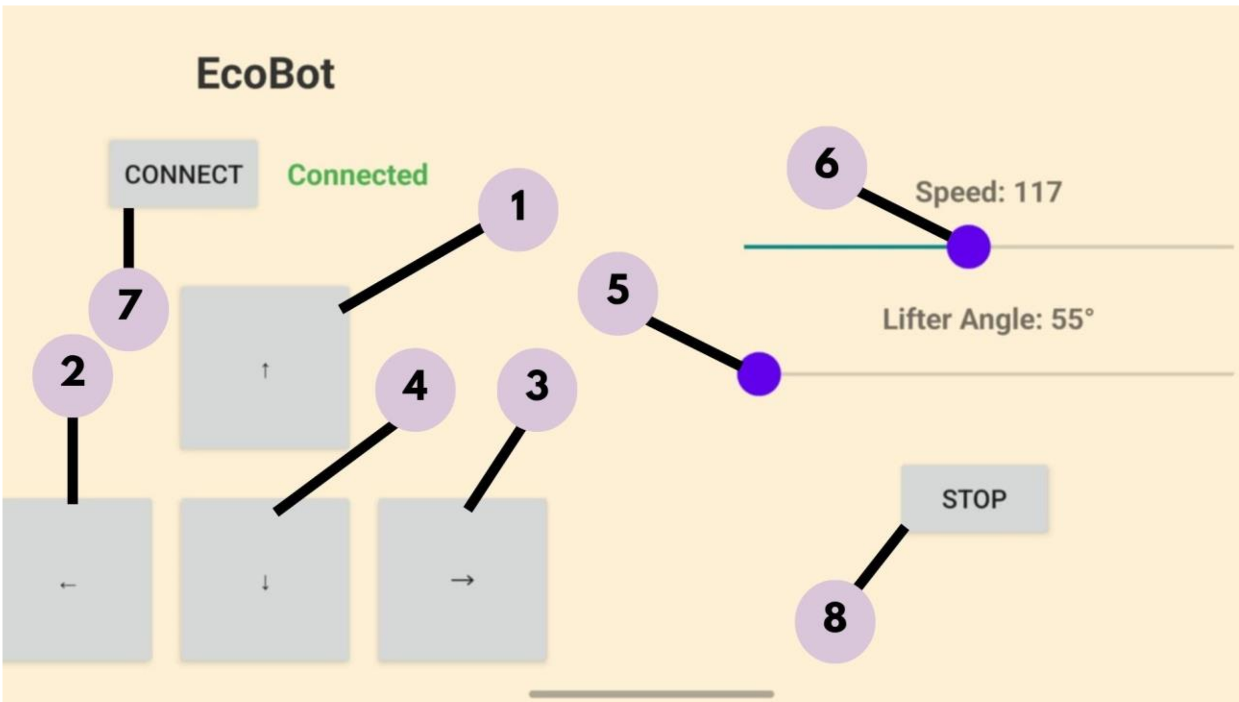

Figure 8.3 - User Manual

User manual

1. To moving forward.
2. To move left.
3. To move right.
4. To move backwards.
5. To adjust the arms.
6. To adjust the speed of the robot.
7. To connecting to Bluetooth
8. To stop any unwanted movements.

9. Conclusion

In this chapter, the conclusion summarizes the overall achievements and outcomes of the project. It highlights the successful development and integration of the EcoBot system, emphasizing how the combination of microcontroller and mobile app effectively addresses the problem of indoor waste collection. The chapter reflects on the challenges encountered, the solutions implemented, and the potential impact of the project on improving household cleaning efficiency. Finally, it may suggest future improvements or expansions to further enhance the robot's capabilities and usability.

9.1. Project Achievements

The project successfully developed a smart indoor waste collection robot capable of picking up small trash. It integrates motor control and a robotic grabber system to clean household spaces. Additionally, the project includes a mobile application that enhances user control and convenience. Overall, the EcoBot demonstrates effective routine cleaning tasks, improving efficiency and reducing manual effort.

9.2. Takeaways and Future Applications

Through this project, we gained valuable experience in integrating hardware and software to create a functional Arduino-based robot that can be manually controlled via a custom-built Android application. We learned how to interface various components such as servo motors, motor drivers, and Bluetooth modules with an Arduino Mega, as well as how to design and implement a mobile app using Android Studio for real-time robot control. This project also emphasized the importance of testing, debugging, and optimizing both hardware and software to ensure reliable performance.

In terms of future applications, this type of manually controlled robot has the potential to be used in household, industrial, and educational settings. For example, it can be adapted for waste collection in homes, factories, or offices, where space may be limited, or automation is not feasible. It can also serve as an educational tool to teach students about robotics, programming, and system integration. With further development, such as reintroducing autonomous features, adding more sensors, or enhancing mobility, this robot could evolve into a more advanced solution for smart cleaning and other practical tasks.

10. Appendix

This chapter present all the references used in the project. These are the code references and inspiration in this project, also shows the curriculum vitae of every project teams members.

10.1. Code References or additional resources

Narayan R, A. (2018, September 2). *Trash collecting robot with IoT technology*. TRASHBOT. https://iottrashbot.blogspot.com/2018/09/trash-collecting-robot-with-iot.html

10.2. References

MArobotic. (2021, October 18). Automatic and Manual Garbage Collector Robot | Application Control Garbage Collector Robot. https://www.youtube.com/watch?v=iagL5j2l-fA

10.4. Curriculum Vitae

ADRIAN S. DELOS SANTOS

STUDENT

#106 Pingkian 1 Central Tandang Sora Quezon City
+639060069683
Adriandelossantos265@gmail.com

OBJECTIVE

Motivated and enthusiastic junior software engineer with a passion for learning and applying emerging technologies. Eager to contribute to dynamic projects and collaborate with talented teams to develop innovative software solutions. Seeking a challenging position where I can leverage my skills in Programming, Front and Back End, UI/UX Design and Java to make meaningful contributions and further enhance my expertise in software development.

EDUCATION

Coronados School of Quezon City
1st Year- 3rd Year High School
(With High Honors)
2017-2020

Emilio Jacinto National High School
4th Year High School (graduated)
(With High Honors)
2020-2021

Emilio Jacinto National High School
Senior High School (graduated)
(With Honors)
2021-2023

National University - Fairview
Bachelor of Science in Information Technology
August 2023 - Present

EXPERIENCE

Freelance Encoder

September 2022 - January 2023

- Deed of Donation of property encoder

Intern in BPO Company

march 1 - march 30

- learning about the process of call center
- practicing mock call
- learning the right etiquette of a call center agent

Concentrix| March 1, 2023-March 30, 2023

- Learning new skills needed in the BPO industry
- Experienced answering calls from clients
- Experienced answering question in interview

Figure 10.4.1 - Delos Santos

NATHANIEL CEDRIC RABANAL

+(63) 9934501610

cedrabanal@gmail.com

21 Dao St. Northview 2 Filinvest 2 Batasan Hills Q.C.

SKILLS

- Microsoft Office Word
- Microsoft Excel
- Microsoft PowerPoint
- Communication Skills
- Fast Learner
- Computer Skills
- Adaptability
- Leadership
- Typing Skills
- Electrical Troubleshooting
- Java Programming
- Python Programming
- MySQL
- Cloud Computing
- Web Development
- Mobile Application Development

OBJECTIVES

A determined and fast-learner who desires to acquire new experiences and skills for the development of my career. An individual who seeks to be a part of the fast development of the company. A person who strives to improve the company. Has an exemplary knowledge on technologies.

EXPERIENCE

Barangay Batasan Hills | Barangay Office Staff

Feb 2023 - Mar 2023

- Worked as an internship to the Barangay Batasan Hills for the OJT of the school.
- Immersed to the community, helping people that requires help in the barangay. Helped my fellow barangay staff with troubleshooting of equipment.

EDUCATION

National University Fairview | 2023-Present
Currently taking Bachelor of Science in Information Technology with Specialization in Mobile and Internet Technologies

San Antonio De Padua Catholic School of Batasan Inc. | 2022-2023
Graduated Valedictorian

REFERENCES

Wilfred L. Gile LPT. from San Antonio De Padua Catholic School of Batasan Inc.
+(63) 947-737-0928

April A. Pancho LPT. from San Antonio De Padua Catholic School of Batasan Inc.
+(63) 910-004-2851

Sherwin Elacion from San Antonio De Padua Catholic School of Batasan Inc.
+(63) 995-257-6215

Figure 10.4.2 - Rabanal