



**Najah National University
Faculty of Computer Engineering**

Hardware Graduation Project (GP 2)



By:
**Tala Zayed
Rayhan Shehada**

Supervisor
Dr. Emad Natsheh

Presented in partial fulfillment of the requirements for Bachelor
degree in Computer Engineering

Jun 2025

Acknowledgment

First and foremost, all praise and thanks are due to Allah, whose blessings, guidance, and mercy made this work possible. Our gratitude to everyone who supported us throughout the development of this project. We would like to sincerely thank our supervisor Dr. Emad Natsheh for his guidance and continuous support, Eng. Abdullah Hinawi for his valuable technical guidance and insightful feedback, and Mr. Alaa Zorba for his skilled craftsmanship and assistance in constructing the metal framework and mechanical components. Our appreciation also goes to our families and friends for their unwavering support, patience, and encouragement throughout this journey.

Disclaimer statement

This report was written by Tala Zayed and Rayhan Shehada at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

<i>Abstract</i>	7
<i>Chapter 1 Introduction</i>	8
2.1 <i>General background</i>	8
2.2 <i>Objectives</i>	8
2.3 <i>Overview</i>	9
<i>Chapter 2 Theoretical Background & Related works</i>	10
5.1 <i>Related works</i>	10
<i>Chapter 3 Methodology</i>	11
4.1 <i>Hardware Component</i>	11
4.2 <i>Hardware Implementation</i>	18
3.2.1 <i>System Overview</i>	18
3.2.2 <i>Wheelchair Implementation</i>	19
3.2.3 <i>Smart Home Implementation</i>	24
• Communication Between ESP32s Using MQTT through HiveMQ cloud broker	24
• Touchscreen-Based Input	26
• Voice Command Input	26
• Mobile App Control	27
• What is connect to each subscriber ESP32.....	Error! Bookmark not defined.
4.3 <i>Mobile Application Implementation</i>	27
4.4 <i>Development Tools and Languages</i>	27
4.5 <i>User Interface and Device Management Features</i>	27
4.6 <i>Voice Command Recording and Transcription</i>	28
4.7 <i>Publishing Commands via MQTT</i>	28
4.8 <i>Text-to-Speech Feedback and Bluetooth Playback</i>	28
3.4 <i>Constraints</i>	28
• DC Motors Synchronization	28
• Heavy Battery Weight	29
• Limitation of nRF Module – ESP32 Solution	30
• Internet issues and Hotspot Solution	30
• ESP32 Communication Over the Internet and HiveMQ Cloud	31
Chapter 4 <i>Results and Discussion</i>	33
4.1 <i>Smart Wheelchair Operation</i>	33
4.2 <i>Smart Home Operation</i>	33
• <i>Mobile Application Operation</i>	Error! Bookmark not defined.
<i>Chapter 5 Conclusions and Recommendation</i>	37

5.1	<i>Summary</i>	37
5.2	<i>Recommendations for Improvement</i>	37
5.3	<i>Lessons Learned</i>	37
5.4	<i>Future Work and Directions</i>	38
	<i>References</i>	39
	<i>Appendix</i>	40

Tables of Figures

Figure 1 : ESP32	11
Figure 2 : 18650 Li Ion Battery 3.7V	12
Figure 3 :LM2596 voltage regulator.....	12
Figure 4 : 2 channel relays	12
Figure 5 : Max9814 microphone module.....	13
Figure 6 : Bluetooth module	13
Figure 7 : 2.4 TFT Touch Screen.....	14
Figure 8 : Arduino Mega 2560.....	14
Figure 9 : Mechanical Joystick	15
Figure 10 : Momentary push button.....	15
Figure 11 :Brushless motor (hoverboard motor)	Error! Bookmark not defined.
Figure 12 : Brushless motor controller	16
Figure 13 : 36v Li Ion Battery	17
Figure 14 : DC Motor (موتور مستأحة سيارة)	17
Figure 15 : Dual H-Bridge Motor Driver L298N	17
Figure 16 : ECI12200 12V 20Ah.....	18
Figure 17 : Smart Home Implementation Structure.....	24

No table of figures entries found.

Abstract

When we first thought about our graduation project, one thing was clear: we wanted to create something that truly helps people. Thus, we will make a smart mobility chair designed to assist individuals with movement disabilities in completing their daily tasks more easily and smoothly.

The chair is primarily intended to enhance independence for individuals with disabilities by integrating assistive technologies. It features an automated control panel that allows users to manage various aspects of their home, including the air conditioner, fan, LED lights, alert system.

“Harmoni Move” is equipped with wireless communication, enabling seamless interaction between the chair and smart home devices. Additionally, the chair incorporates voice control for hands-free operation, an alert system for safety, and adaptive movement control to ensure smooth moving inside the house.

Beyond mobility, our system focuses on enhancing quality of life by providing intelligent automation, safety features, and user-friendly controls. Our goal is to contribute a meaningful project rather than simply submitting a graduation requirement, and this is our initial vision, and we may add or update some details.

Note: there is no projects have been done as the same exact idea, but similar ideas have been implemented.

Added Features:

- Wireless communication for seamless device control.
- Gesture and voice control for hands-free interaction
- Emergency alert system to enhance safety
- Automated adaptive home electronics based on the disabled person needs.

Chapter 1

Introduction

2.1 General background

The increasing number of disabled people and elderly with movement issues, grows the need for devices to support them and increase the quality of life. In the past decade, the Smart wheelchair becomes the most important mobility aid. Those chairs gained their importance from its safety level and ease of use and these devices are made with the needs of people with physical disabilities, they make it easy and natural for the user to interact with the intelligent wheelchair. The system lets people control and use the wheelchair with their own physical abilities, which gives them more freedom. The wheelchair also has a smart navigation system that can detect obstacles, recognize the environment, and avoid collisions. This makes it possible to move safely and quickly in a variety of settings [1].

Smart Wheelchair is a mechanical device designed with independent movement based on the commands from the human. This minimizes the needed effort from the human to push and drive the wheel. Along with obstacle detection system the wheelchair makes sure to minimize collision and accidents, to improve the life quality of the user. Smart wheelchairs have become very popular lately. These devices are used whether transportation, elderly with movements difficulty. While these is a vital aid for people with disabilities and those who lost their mobilities.

Various types of wheelchairs have been designed in the previous years, but the new models have been developed utilizing the trending technologies of the hardware and software such that It doesn't require the user to do much or make many adjustments [1], [2].

This project aims to build a smart wheelchair to support people with motion disabilities through joystick and 3 buttons for speed and support them in controlling home appliances in simple and easy way to support their independence and improve quality of life supporting various tasks such as turning on a light, adjusting the fan, or control any device, and moving safely through their own home. The project integrates smart home integration with mobility assistance in one unified system. The user will have the ability to control the movement of the chair using a joystick in all directions, and to make sure that the chair is safe in all environment obstacles detection has been added to avoid accident. Along with a touch screen attached to the chair the user can control all integrated appliances whether using the panel or using voice commands through voice sensor to make sure hands-free integration is supported as well. Mobile application has been designed as well to support the smart home integration in case the user is way from the chair using voice commands only.

Harmoni Move wants to go beyond traditional mobility aids by combining smart automation safety features, easy-to-use controls, and assistive technologies into one system. The project is a practical and socially useful solution that puts accessibility, comfort, and user freedom first.

2.2 Objectives

The main goal of this project is to create mechanical wheelchair system based on normal wheelchair makes it easier for people with severe physical disabilities to move around, be independent, and be safe. The system's goal is to give users a way to control both their movement

and their environment. It does this by combining smart home connectivity, intelligent control mechanisms, and a mobile app interface.

The specific objectives of the project are:

- Design a smart wheelchair equipped full directions joystick movement and 3 different speeds.
- Support Safety in the wheelchair by adding obstacles detection system.
- Integrate the chair with smart home devices using a touch screen or voice commands using wireless communication for real time interaction.
- Develop mobile application for smart home control using voice command to improve the user experience through responsive control, reliability, and ease of use.

2.3 Overview

In this section of the chapter introduction, an outline for the next chapters will be presented. The project report follows the following structure:

- Chapter 2 will present Theoretical Background and Previous Work for the research topic of smart wheelchair and supportive smart home for people with motion disabilities.
- Chapter 3 Methodology, this chapter outlines the methodology followed in the development of the Harmoni Move system, detailing the hardware and software components, system architecture, communication protocols, and control mechanisms.
- Chapter 4 analyses the research output in detail and presents the results.
- Chapter 5 discussion, interpret and compare the results. Clarify the features and limitations of the work.
- Chapter 6 concludes the topic and provides suggestions for further development.

Chapter 2

Theoretical Background & Related works

In this section, the literature review is investigated to analyse the current existing smart wheelchair and the integrated smart home and the various existing hardware and software technologies that can be utilized to improve the smart wheel with smart home integration for better safety, independence, user experience and quality of life.

5.1 Related works

Due to the importance of mobility aids the researchers have implemented many smart wheelchairs with various concepts and features to help people with disability in their daily life movement and activities.

C. F. Lung et al. proposed a low-cost wheelchair that is affordable for developing countries people, The design depends on set of sensors to support the obstacles detections and a joystick for control, and an interface for remote control movement and monitoring [3] While the same author improved his proposed wheelchair in the other paper [4], by adding speech recognition, camera and ultrasonic obstacles recognition.

While authors in proposed a design for a wheelchair that assist disabled people on mobility using joystick with focusing on some safety features such as anti-tip wheels and a seat harness [4].

While other researchers proposed phone-controlled wheelchairs such as C. F. Riman [5] who proposed an Android application that support voice command, gesture and touch screen input for navigation. Similarly, S.U. Khadilkar and N. Wagdarikar [6], [7] proposed a wheelchair system that is controlled using hand gestures and voice commands. It uses flex gloves for hand command and microphone for voice commands.

Recent improvements in smart wheelchair technologies have investigated a wide range of topics, including controlling the motion, like joysticks, eye-tracking, voice commands. Other work to add additional features like fall detection, monitoring vital signs and emergency alert. Also, various energy sources like solar panels. Although all this development on the fields there is still gaps. Many solutions are still too expensive and don't work for people with disabilities in both their upper and lower limbs, so a lot of people aren't getting the help they need. Mobility systems and smart home environments don't work together well in real life, which makes it harder for people to be independent in their daily lives.

Harmoni Move is a complete smart mobility solution that combines assistive robotics with smart home automation. It has a voice-controlled navigation, and adaptive controls for safe movement inside. It also integrated with many smart home devices, so people can easily interact with their surroundings. There is also a separate mobile app that lets users use voice commands to control home appliances. Harmoni Move fills in important gaps in current research by addressing control diversity, real-time safety and smart environment integration.

Chapter 3

Methodology

This chapter clarifies the methodology followed to design the Smart wheelchair and integrate it with Smart home. Including the chair hardware design, the smart home design, communication between them, and implementing Mobile application to support the Smart home functionalities.

4.1 Hardware Component

3.2.1 ESP32

ESP32 is a one of the most effective components used in implementing **Internet of Things (IoT)** applications. ESP32 a dual-core processor to support high processing power, multitasking and complex tasks handling. It has connection to Wi-Fi and Bluetooth to simplify communication with other devices. large number of input and output pins allow to have connections to multiple sensors and external devices.

Different programming languages and development frameworks can be used to program the ESP32. C++ is the most popular programming language. Using the Arduino IDE to write code in C++. Also, some modules have sensors built in, which makes it easier to add these sensing features to IoT projects [7].

We used 4 ESPs in our project, one of them is as publisher ESP which receives the commands from touch screen through serial communication created with Arduino, or commands came through HiveMQ cloud broker that came from mobile application, or commands came from voice sensor. And the 3 ESPs used as subscribers each of them is on a controller side. The communication between publisher ESP and subscribers ESP is done through the HiveMQ cloud broker.

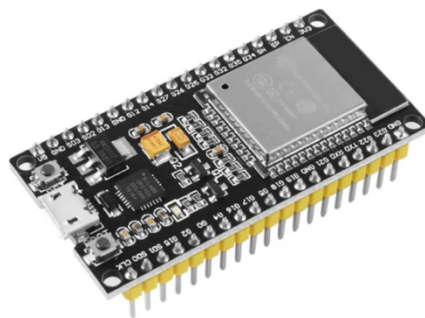


Figure 1 : ESP32

3.2.2 18650 Li Ion Battery 3.7V

18650 Li Ion Battery is rechargeable batteries which has a longer life cycle, higher energy efficiency, and a longer calendar life, in comparison to other types of rechargeable batteries.[8]

We used 2 batteries of 3.7v for each ESP to power it, then regulating the volts to be 5v instead of 7.4 (3.7+3.7).



Figure 2 : 18650 Li Ion Battery 3.7V

3.2.3 LM2596 voltage regulator

LM2596 voltage regulator is step-down buck converter, that provides changing output voltage through built in potentiometer that allows you to adjust the output voltage as you need[9].

We used it to connect to 7.4 voltage that came from 2 of 3.7v Li Ion Batteries and then adjust output voltage to be 5v to be as input voltage to ESP through cable.



Figure 3 :LM2596 voltage regulator

3.2.4 2 channel relays

It is a board that can be used to control high voltage, high current loads. The relays terminal (COM, NO, and NC) is being brought out with a screw terminal. It also comes with a LED to indicate the status of the relay[10].

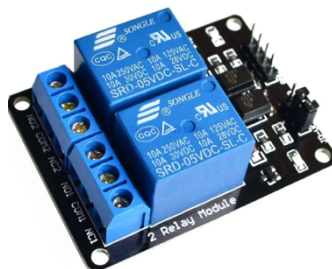


Figure 4 : 2 channel relays

3.2.5 Max9814 microphone module

It is a microphone amplifier module that has built in gain control with different values of gain 40db, 50db, 60db and low-noise microphone bias, it is suitable for a wide range of applications where audio amplification is needed such as portable recorders [11].

We used it for voice command, when user press the momentary push button the voice will be recorded through this microphone and go through ESP. For more voice clarity and noise removing we connected ceramic capacitor and electrolytic capacitor between (VCC/GND) and another capacitor on output.

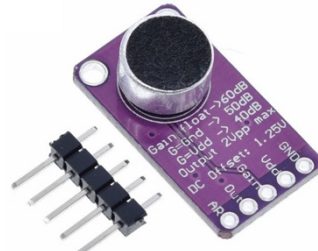


Figure 5 : Max9814 microphone module

3.2.6 Bluetooth module

It is a Receiver Board which is compact and cost-effective device that converts mp3 audios into speakers [12].

We used it to be connected to mobile Bluetooth while using mobile application to play feedback mp3 audio on our connected speaker with this module.



Figure 6 : Bluetooth module

3.2.7 2.4 TFT Touch Screen

It is a TFT touch screen used with Arduino, it has very convenient features, including the high resolution of 240*320 pixels, easiness in use, and ability to display 262000 different colors [[13].

We use it to display the devices from database and display (on, off) choices for each device.



Figure 7 : 2.4 TFT Touch Screen

3.2.8 Arduino Mega 2560

The ATmega2560 datasheet is the basis for the Arduino Mega 2560 microcontroller board. It has 54 digital input/output pins, 14 of which can be used as PWM (Pulse Width Modulation) outputs. It also has 16 analog inputs, 4 UARTs (hardware serial ports), a power jack, a 16 Mega Hertz crystal oscillator, an ICSP header, a USB connection, and a reset button. It has everything need to use the microcontroller; all to do is hook it up to a computer with a USB data cable or power it with an AC-DC adapter or battery.[14]

We use Arduino to connect touch screen with it, implement serial communication with publisher ESP and control the motors.

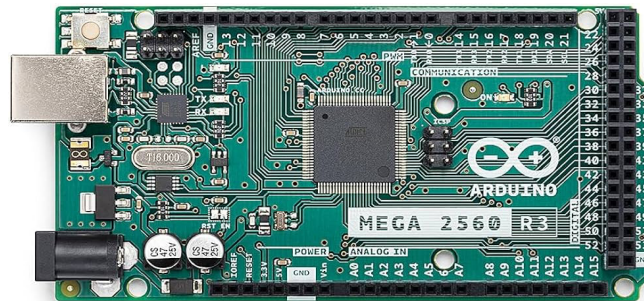


Figure 8 : Arduino Mega 2560

3.2.9 Industrial 4-way mechanical joystick switch (ABB MCB-10)

This project uses an ABB MCB-10 joystick, which is made for controlling direction. It lets you control things in four directions with mechanical switches that open and close circuits based on where the joystick is. The base unit has stacked contact blocks, each with numbered terminals that make it easy to find the right wires. You can wire these terminals to a microcontroller's digital inputs to find out which way something is moving. This kind of joystick works by closing a normally open contact when it is pushed in a direction (for example, connecting terminal 3 to 4). This sends a signal to the control system. This joystick sends binary (ON/OFF) signals instead of variable voltage like analog joysticks do. This makes it perfect for simple directional control logic in embedded systems [15].



Figure 9 : Mechanical Joystick



3.2.10 Momentary push button

Momentary push button is a special type of buttons, the circuit stays closed as long as the actuator remains pressed button and when release it become an opened circuit [16].

We use it as record button with max9814 microphone, and 3 buttons to control Speed.

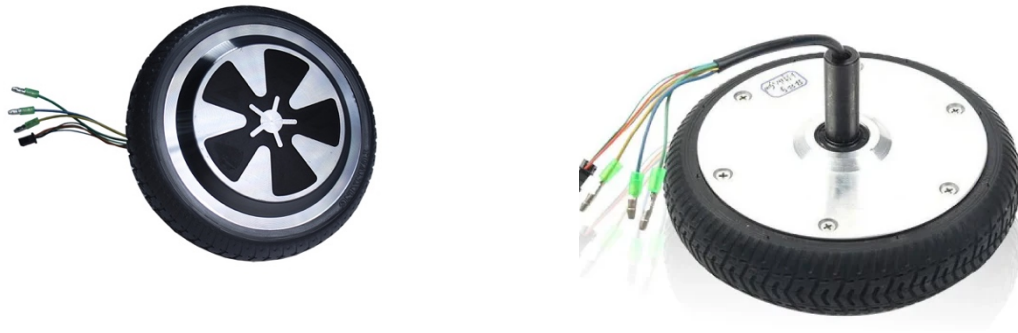


Figure 10 : Momentary push button

3.2.11 Brushless motor (hoverboard motor)

A brushless DC motor (BLDC) is a synchronous motor using a direct current electric power supply. It uses an electronic controller to switch DC currents to the motor windings, producing magnetic fields that effectively rotate in space and which the permanent magnet rotor follows. It has high power to weight ratio, high speed, high efficiency, and low maintenance [17].

Hoverboards use (BLDC) motors, to drive the wheels and provide propulsion. These motors are often integrated within the wheel hubs, offering a compact and efficient design. The motors work in conjunction with sensors and a control system to enable self-balancing and rider-controlled movement. The motor has 3 phases and 3 hall sensors.



3.2.12 Brushless motor controller

It is a controller or digital signal processor which receives the feedback from hall sensors to determine the position of rotor and precisely control the motor by sequentially energizing the appropriate motor windings [[18]. It has many wires we use throttle for controlling the speed, 3 phases, 3 hall sensors, VCC and GND.



Figure 11 : Brushless motor controller

3.2.13 36v Li Ion Battery

We used it for powering *hoverboard motors*.



Figure 12 : 36v Li Ion Battery

3.2.14 DC Motor (موتور مسآحة سيارة) (removed)

We used it for left and right movement.



Figure 13 : DC Motor (موتور مسآحة سيارة)

3.2.15 Dual H-Bridge Motor Driver L298N (removed)

It Is a motor driver that provides controlling direction and speed of DC motors, it can drive motors with voltage between 5v to 35v [19].

We used it to drive motor of left and right movement.

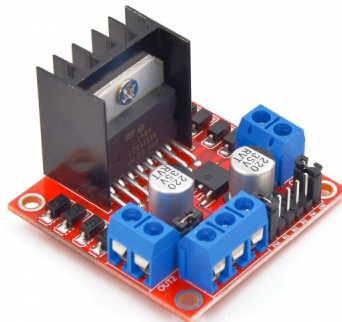


Figure 14 : Dual H-Bridge Motor Driver L298

3.2.16 ECI12200 12V 20Ah (removed)

We used it for powering left/right dc motor.



Figure 15 : ECI12200 12V 20Ah

4.2 Hardware Implementation

3.2.1 System Overview

We converted a normal wheelchair into a electric wheelchair with smart home system to support users with limited mobility. To make control more easy and comfortable, we designed a custom 3D-printed arm mounted on the side of the chair, this arm includes all the essential control modules in one place: a TFT touchscreen for home device control, a joystick for movement, three speed control buttons, a microphone (MAX9814) to capture voice commands, and a momentary push button to start voice recording.

The joystick allows the user to drive the wheelchair in all directions, the three speed buttons provide options to move on default speed or decrease and increase the speed, we also added ultrasonic sensors at the front of the wheelchair to detect obstacles, when something is too close, the system automatically stops.

The TFT touchscreen will display devices with ON and OFF buttons next to each one, the user can simply tap a button to control any connected device such as lights, fans, or any 220v AC appliance. We also added a voice control feature by microphone (MAX9814). When the user presses the push button, the system starts recording a voice command.

Along with hardware control, we developed a mobile application that supports both tap and voice-based commands. The app allows users to control home appliances remotely, manage devices, and receive spoken feedback after each action.

- 3.2.2 Wheelchair Implementation



First Trial: 2 DC Motors (Failed)

In our first trial to move the wheelchair, we designed rear metal bridge from scratch, then we've started by testing traditional relays to control the motors' directions, but they offered no speed adjustment. To improve the control speed, we moved to IBT_2 motor driver which gave us better handling of direction and speed. However, the motors still not synchronized properly, one motor often moved faster than the other. Applying multiple trials, the motors couldn't maintain consistent alignment.



We used a DC motor to drive the front bridge in the initial design. We mechanically mounted the bridge onto the front wheels and caused the bridge (highlighted with green) to turn the wheels to the left or to the right. When we turned the DC motor clockwise, it propelled the bridge forward and turned the wheels towards the right. Counterclockwise turning moved the wheels towards the left. We employed a timed strategy where the motor was instructed to rotate for 1 second for right turn and 1.2 seconds for left turn.

This solution proved to be erroneous and inconsistent because of mechanical timing lags and friction that imposed randomness in the wheel angles. The system was thus imprecise and could not accommodate consistent direction control.

To stay away from these limitations, we switched to using two brushless hoverboard motors, with each powering one side of the wheelchair. Instead of rotating the front wheels mechanically in left-right case, we achieved left and right movement by having the one motor spin forward and the other backward.



Second Trial: Brushless motors with brushless motor controller

**** photo of bridge and hoverboard motors**

After having synchronization issues with the DC motors, we move to **brushless motors, which is usually found in hoverboards or electric scooters** to improve performance and control, because these motors are powerful enough to support the weight and movement of a wheelchair. We mounted the motors on rear wheels of the chair, and we **designed a metal bridge** for them. The motors have three-phase power wires and hall sensors. the system is powered by a **36 lithium-ion** battery pack that provides a stable energy source to drive motors.



To control the brushless hoverboard motors, we used a **brushless motor controller**. This controller is utilizing both **three-phase motor connections** and **hall sensor feedback** to move the motor.



The controller includes multiple wires:

- 3 Phase Wires
- Hall Sensor Wires
- Battery Input
- Throttle Signal
- Learning Wire
- Cruise Pin
- 3-Speed
- L-Brake

We don't use all pins in our project, and we will discuss which pins used later.

A. Using learning pins (Failed)

Directions Implementation

In this trial we tried to implement directions using the learning pins of the mentioned controller, The setup included two relays per motor connected with digital pins on the Arduino which pulsed the learning wires to switch the direction mode.

We disconnect the learning wires in code to stop the motor, pulse them for forward or reverse and then enable the throttle again.

But the system behaved surprisingly at runtime. All joystick triggers reversed motor direction instead of keeping it in the desired direction. Even if the logic was specially timed and delays inserted, **the controller would often ignore the direction setting by joystick and change randomly**, this non-deterministic behavior rendered joystick control impossible since the system reversed on every press, no matter the intended command.

Speed Implementation

For this trial, we implemented speed control by transmitting PWM signals from the Arduino Mega to the two hoverboard motor controllers throttle pins. we used 2 pins from Arduino for transmitting PWM signals, and five speed levels were established in an array. Three buttons were provided for the user to regulate speed, one for speed increase, one for speed decrease, and one for reverting to the default level. The system dynamically changed the PWM output based on button input and reflected the speed level change on both motors via analogWrite function.

The PWM signals were stable, and the hoverboard controllers responded well to every speed adjustment, whether increasing, decreasing, or returning to default.

Obstacles Implementation

To implement collision detection system we used the HC-SR04 sensor to improve safety and prevent collisions. The sensor constantly monitors the distance in front of the wheelchair, and when it detects an object within 50 cm, the system immediately halts both motors by turning off the relay-controlled learning wires and sending the throttle PWM to zero. This overrides joystick input to prevent unwanted movement. But since the barrier is crossed and the joystick remains down, the system automatically continues to move in the previously commanded direction. Therefore, autonomous safety intervention is weighed against smooth user control without the necessity for the user to repeat commands on each stop.

Result of (Using learning pins)

- This experiment emphasized **the drawback of using learning pin for direction switching, while the speed and obstacles works correctly**. The hoverboard motor controller was not intended for dynamic in-motion direction toggling and thus compelled us to proceed to more stable alternatives for full bidirectional control.

B. Using swapping phases and hall sensors

Directions Implementation

we designed a system using two 4-channel mechanical relay modules to switch the forward direction wiring and the backward direction wiring. We wanted to dynamically switch connections between the motor's green and yellow phase wires and the controller's respective outputs, thereby reversing the direction of the motor by the actuation of the relay. The blue phase wire of the motor was left as is, directly attached to the controller because it was common in both modes and did not determine direction. During the forward configuration, the relays connected the green motor wire (GM1) to the green controller output (GC1), and the yellow motor wire (YM1) to the yellow controller output (YC1). In contrast, for the rearward configuration, the

wiring was deliberately reversed: the green motor wire (GM2) was wired to the yellow controller output (YC2), and the yellow motor wire (YM2) to the green controller output (GC2). This relay switching was accomplished using the COM contacts tied to the motor wires and the NO contacts tied to the controller outputs, with one set of relays energized at any moment depending on joystick direction inputs.

To ensure proper commutation and feedback alignment, we also extended this logic to the hall sensors, using additional relay channels to swap the green and yellow hall sensor wires during direction changes, while keeping the blue hall wire permanently connected. This guaranteed that the controller received correct hall signal sequences corresponding to the phase configuration in each direction.

While this configuration successfully enabled reliable directional control by allowing the motor to respond appropriately to joystick-based forward, backward, left, and right commands.

Speed Implementation

To be continued

Obstacles Implementation

To implement collision detection system, we used the HC-SR04 sensor to improve safety and prevent collisions. The sensor constantly monitors the distance in front of the wheelchair, and when it detects an object within 50 cm, the system immediately halts both motors by turning off the relay-controlled learning wires and sending the throttle PWM to zero. This overrides joystick input to prevent unwanted movement. But since the barrier is crossed and the joystick remains down, the system automatically continues to move in the previously commanded direction.

Therefore, autonomous safety intervention is weighed against smooth user control without the necessity for the user to repeat commands on each stop.

Result of (Using swapping phases and hall sensors)

To be continued

Pins/used?	Using learning pins and hall Sensors	Using switching phases (Relays)
3 Phases	Yes	Yes
Hall Sensor	Yes	Yes
Battery Input	Yes	Yes
Throttle Signal	Yes	Yes (No effect)
Learning Wire	Yes	No
Cruise Pin	No	No
3-Speed Selector	No	No
L-Brake Line	No	No

3D-printed control arm design



• 3.2.3 Smart Home Implementation

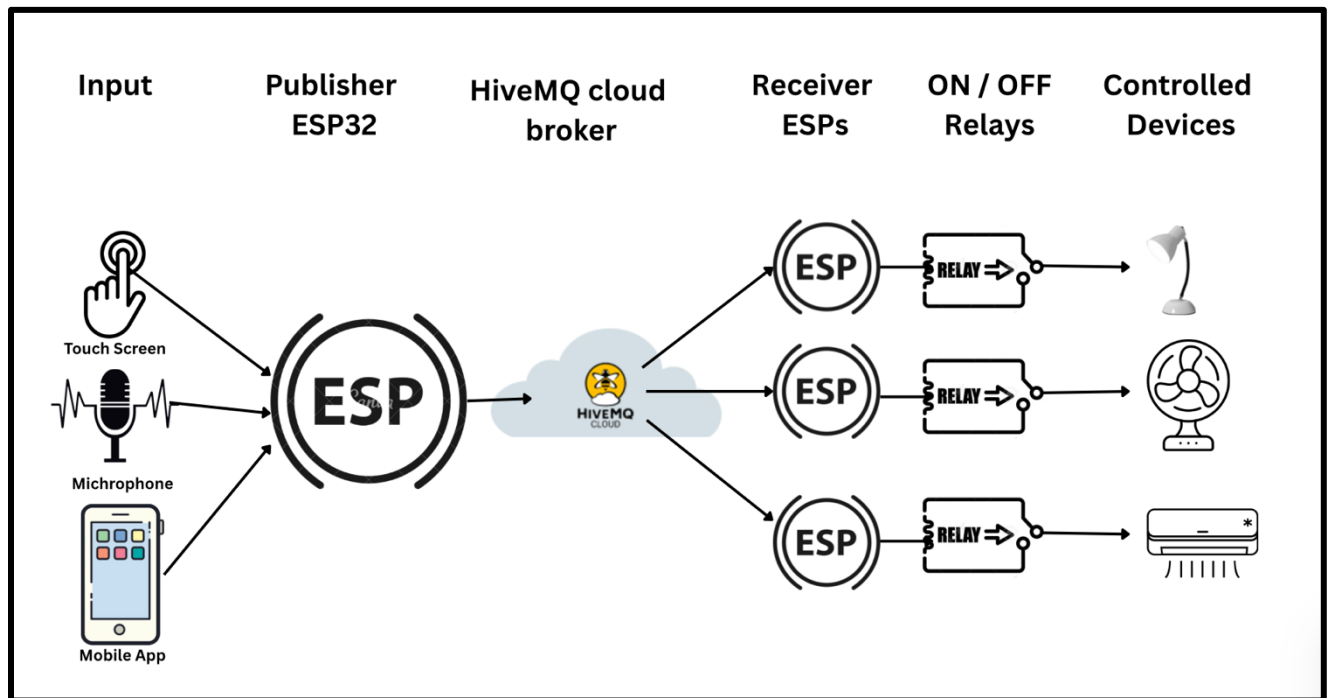


Figure 16 : Smart Home Implementation Structure.

Communication Between ESP32s Using MQTT through HiveMQ cloud broker

MQTT (Message Queuing Telemetry Transport) protocol is used in the communication between the publisher ESP32 and the other subscribers ESP32. MQTT has been chosen due to its compatibility with low power and low bandwidth devices, to achieve real time interaction and low delay. This communication is done over internet using HiveMQ Cloud Broker. The publisher ESP32 publishes commands to the smart devices within the system, where each device has its own

ESP32 to control its status, where a relay module switch it on/off based on the command received by the ESP32.

The flow starts with a command received by the publisher whether from the touchscreen, a voice input. It works to determine what is the targeted device, to send MQTT message including the topic which identify the receiver, payload that contains the command whether ON / OFF. For example, to turn on a fan, the publisher sends a message with the topic = controller# of fan and the payload = ON.

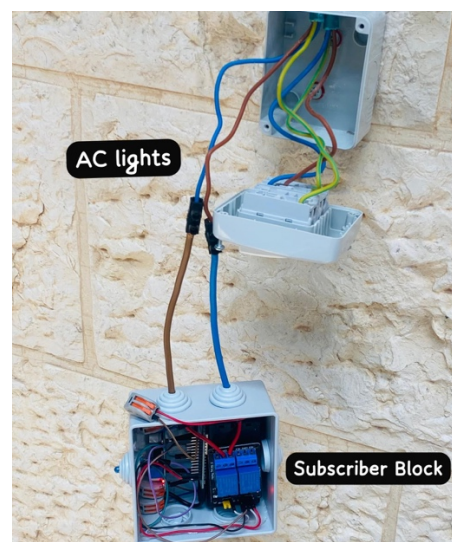
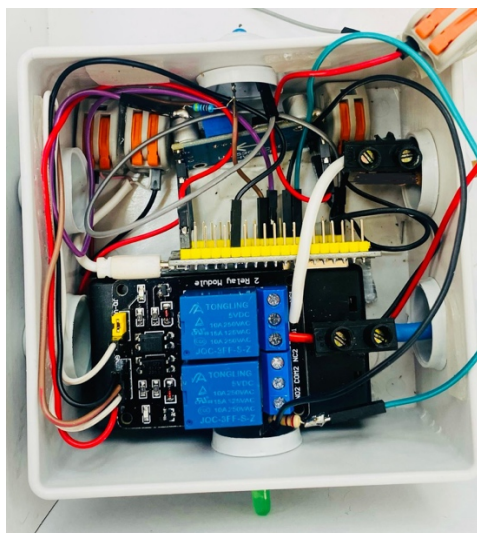
As all other ESP32s are connected to the HiveMQ broker and each one is listening for its assigned topic such that when it's topic publisher, only this ESP will receive the message and switch the relay to achieve the needed action.

The design of the system makes it modular, where each module is independent and adding devices will not affect the system by setup new ESP and assign unique topic. Also, the system is scalable, where increasing the number of devices will not add complexity to the system. Moreover, using the HiveMQ makes the system more reliable for real time communication as it clouds hosted. Supporting both local and remote communication.

Subscriber Blocks

Each subscriber ESP32 is used to control a certain home device using a specific hardware setup including:

- Two cells 3.7V Li-ion battery pack for power supply.
- LM2596 voltage regulator to regulate the voltage from 7.4 to 5V to be compatible with the ESP32.
- Power Cable to connect the voltage to the ESP32
- 2-channel relay module to control the high-voltage AC appliances
- Two LEDs for feedback where first to indicate the status of the ESP32 and the other the status of the device



Touchscreen-Based Input

The touchscreen input system structure contains an Arduino board connected to a TFT display, to allow user friendly interaction to control devices by just a tap. The touchscreen displays the list of the devices. This list is fetched by the published ESP32 publisher via an API over HTTP, then pass the response to the Arduino through Serial communication, which accordingly display it on the touch screen.

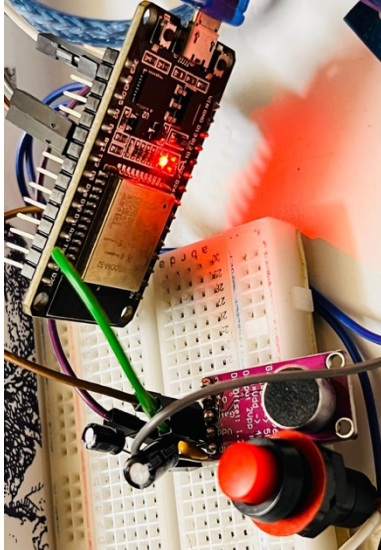
When the user clicks the switch button for one of the devices, the Arduino uses another serial connection using TX and RX pins to send the action to the ESP32 publisher. The ESP32 then processes the action, identifies the topic and the payload and pass it to the HiveMQ. As mentioned previously this message will be received by the subscriber ESP which listen to this topic which h accordingly will activate its relay to switch on/off the 220 AC device.



**** TC after finish**

Voice Command Input

The system allows hand free interaction using voice control command to make it more accessible and convenient. The ESP32 has push button to activate MAX9814 microphone to start record the audio and save it on MP3 format and convert to WAV file as final format. Then ESP32 send the audio file through an HTTP request to the OpenAI transcription API which is speech to text API to convert the voice to text for example “turn on the fan,”. The text command will be sent to specific HTTP API that will publish it to the HiveMQ with the identified topic and payload. The targeted subscriber ESP32 device will receive the message and act accordingly. In parallel with this, this text will be sent to another OpenAI command called interpret to return a response to this command in certain predefined format using prompt for example “the fan turned on” to provide audio feedback. And finally send this response to OpenAI text to speech command and play this sound to the user using Bluetooth audio module, for the example here a sound will be played saying “the fan turned on”. This spoken response makes the system more interactive.



** mic after finish

Mobile App Control

A mobile application has been implemented to support the system and allow the user to control the devices remotely even he is not on the chair. The app contains main button “Start Recording” to capture the voice input, then goes through the same flow mentioned above for the voice command input using microphone ending up with turning on / off the device and spoken feedback.

Note: We will discuss it more in the next page

4.3 Mobile Application Implementation

Specific mobile application was designed for this project to allow remote and interactive control for the smart home devices. It supports set of feature voice commands, speech feedback, and includes CRUD operations for the devices.

4.4 Development Tools and Languages

Frontend Framework: React Native (Expo)
Backend: Node.js + Express (with OpenAI API)
Language: JavaScript
IDE: Visual Studio Code

4.5 User Interface and Device Management Features

The App has a user friendly and simple interface, to support managing the devices through:

- Adding new device (device name and controller),
- Edit existing device.
- Delete existing device.

These updates are done using specific APIs that update the database and accordingly communicate the updated records to the touchscreen to ensure synchronization.

4.6 Voice Command Recording and Transcription

With a single simple button, the user will be able to record its command using the phone microphone send this to OpenAI's transcription API (Whisper) [20] to do a speech to text conversion and return the detected text, i.e., "turn on the fan". The detected text will be displayed on the screen to make confirmation.

4.7 Publishing Commands via MQTT

The transcribed text is sent to another backend endpoint that parsed the text to extract the following:

- The action ON / OFF
- MQTT topic (e.g., `fan`).

A message contains both action and topic are published to HiveMQ MQTT broker, to be received by the specific ESP to perform required action.

4.8 Text-to-Speech Feedback and Bluetooth Playback

For more interactivity, the system provides audio feedback after the action is performed. This audio feedback is generated using the OpenAI's Text-to-Speech API [21] and played back on a Bluetooth audio module.

3.4 Constraints

• DC Motors Synchronization

Car windshield wipers DC Motors have been initially used, but these motors rotate at different speed even on the same voltage. First, we've tried to switch from simple relays to IBT_2 motor drivers which allows controlling speed via PWM signals. However, this didn't achieve the accurate synchronization. So, we moved to hoverboards motors.



- **Heavy Battery Weight**

At the beginning of the project implementation 30 kg lead-acid battery was used. Due to its high weight, it affected the movement and the stability of the wheelchair of the project. To reduce this weight 12V 20A 12 kg 2 batteries was used for the DC Motors and then changed to only 2 lightweight 36V lithium-ion batteries for the hoverboard motors, this way we achieve the needed power level without affecting the project mobility.



- **Left and Write movement**

We initially used a DC motor to power a bridge mounted on the front to turn the wheels of the wheelchair left or right based on timed motor rotation (1 second to the right, 1.2 seconds to the

left). This approach was faulty due to mechanical lag and friction, causing inconsistent control of direction.

We replaced the initial configuration with two brushless hoverboard motors, one that governed one side of the chair. We rotated one motor clockwise and the other counterclockwise to generate smooth and stable turns without mechanical steering.



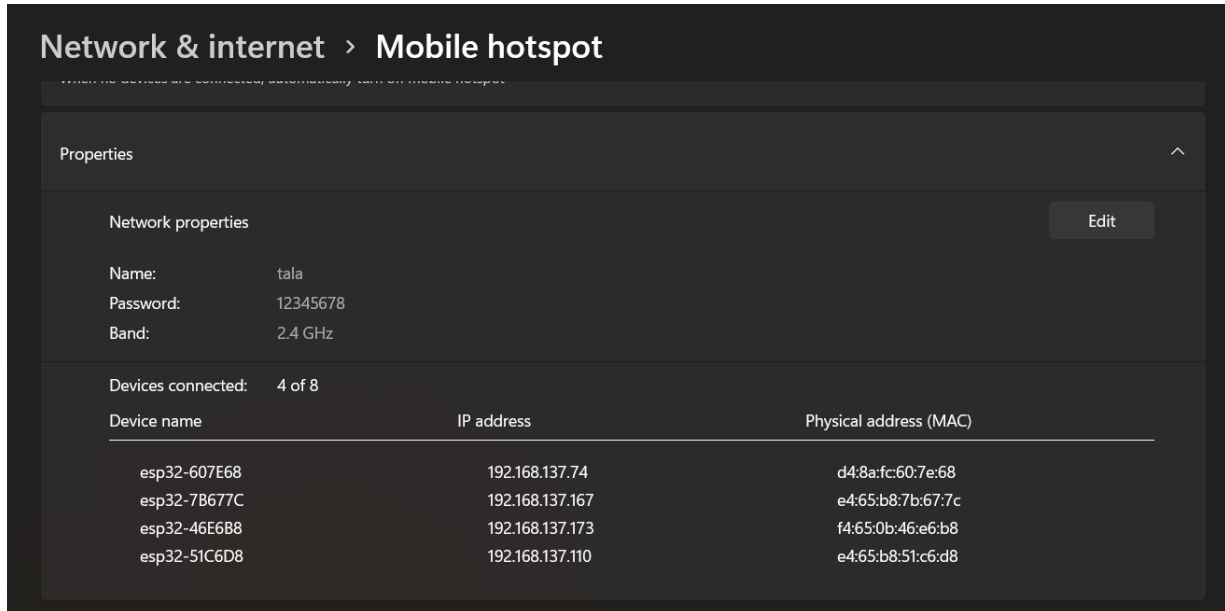
• Limitation of nRF Module – ESP32 Solution

We tried to use nRF24L01 modules for wireless communication between the Arduino (sender) and ESP32 (receiver). However, this caused a lot of communication issues including packet loss, limited range, and signal instability. To overcome these issues ESP32 modules were used sending and receiving over Wi-Fi.



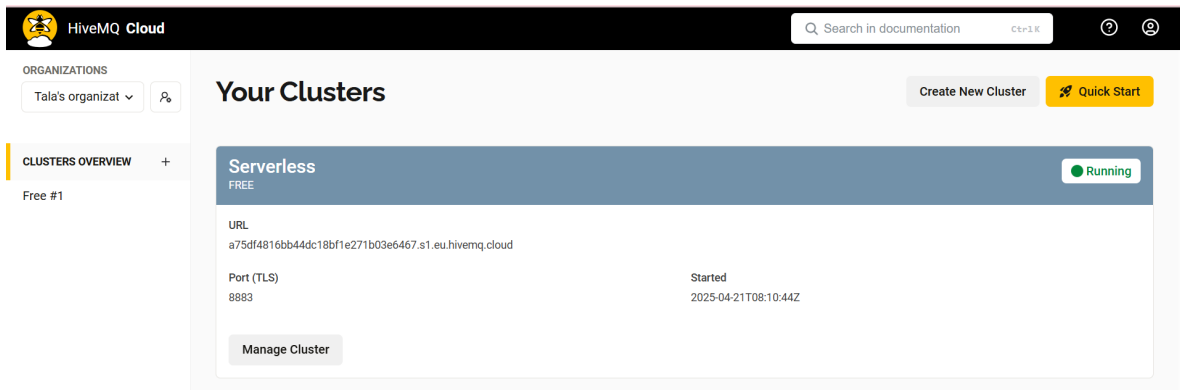
• Internet issues and Hotspot Solution

While working on the project in the university, we faced a challenge on connecting the ESP32 to Najah WIFI as it requires login credentials via an enterprise authentication protocol (802.1X with username and password). While ESP32 does not support this type of login. To handle this, mobile hotspot was used.



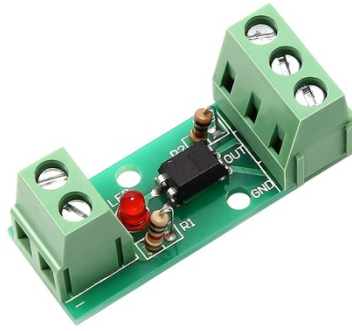
• ESP32 Communication Over the Internet and HiveMQ Cloud

The ESP32 modules have communicated over the internet using a HiveMQ Cloud. The first try was public MQTT broker was free, but it was unreliable and erratic. The second was Adafruit IO, which delayed real-time updates and introduced restrictions because of topic/feed structures. After examining these problems, we ultimately decided to use the HiveMQ Cloud broker, which offered secure communication, a free tier, and reliable MQTT support. This was a game-changer because it allowed our ESP32 units to communicate with each other over the internet in a consistent and seamless manner.



• Optocoupler

We initially used optocoupler for switching high-voltage AC devices, but the optocoupler was damaged (burned) since it was unable to handle the high voltage and current requirements of AC loads, so we move to relay.



Chapter 4

Results and Discussion

The implemented system was tested in two components: (1) wheelchair control using joystick input, safety sensors, and movement; and (2) smart home control using multiple input methods (voice, touchscreen, and mobile app). With a focus on device response and integration, this chapter examines the behavior, communication flow, and successful operation of both systems.

4.1 Smart Wheelchair Operation

The wheelchair movement was tested using joystick and confirmed smooth directional movement in all directions - forward, backward, left, and right. The obstacles detection functionality was tested as well, and it was able to detect the nearby obstacles and stop the chair avoiding any Collision or accident.

The chair was able to support people with disabilities, providing smooth movement with safety support which make it easier for them to be independent.

** chair after finish photo 1

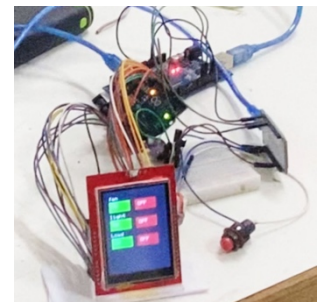
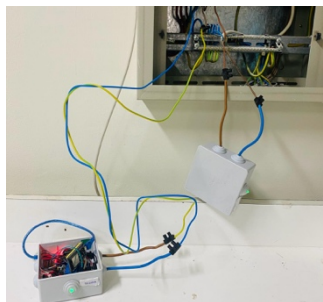
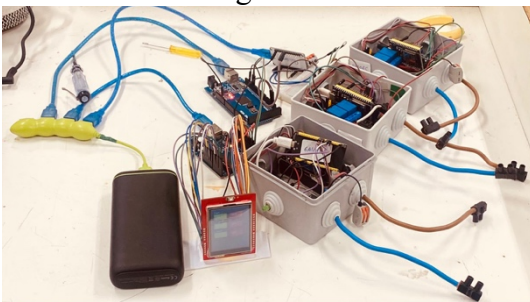
** chair after finish photo 2

For more details for results and testing see the demo.

4.2 Smart Home Operation

We tested the smart home system part in the university (??) we connected the boxes with the wires of each device from the (??) the green led is always on to indicate that box is working, the blue led is indicator if device on or off.

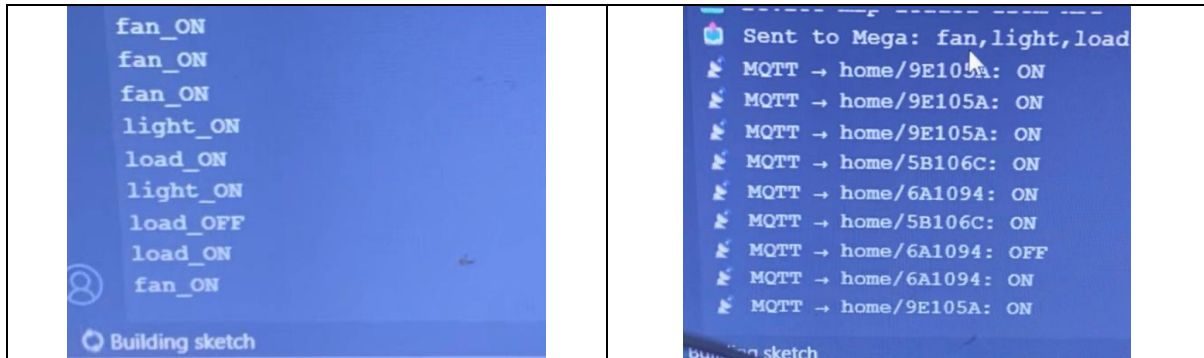
After that we moved touchscreen, Arduino, publisher ESP32, power bank with 3 cables in box of 3d control arm design.



** after finish photo (smart home focous)

- **Touchscreen Control Results**

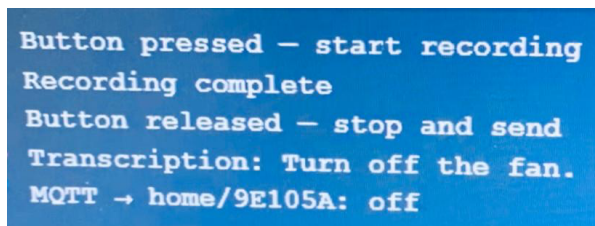
The touchscreen test results show its success in supporting the smart home control by displaying all available devices clearly, along with the switch on/off buttons which trigger the connected device immediately and correctly in all cases. Beside the LED indicators that shows the status of the device whether on / off. The system worked smoothly, and no failures or delays were observed during interaction.



The first image taken from Arduino serial monitor that connects to touchscreen and takes commands from it in format of device_on/off and send it through serial communication to ESP32, the second image taken from ESP32 serial monitor and shows that devices sent from ESP32 to Arduino and publish the commands to MQTT cloud to the specified topic with specified command.

- **Voice Control Results**

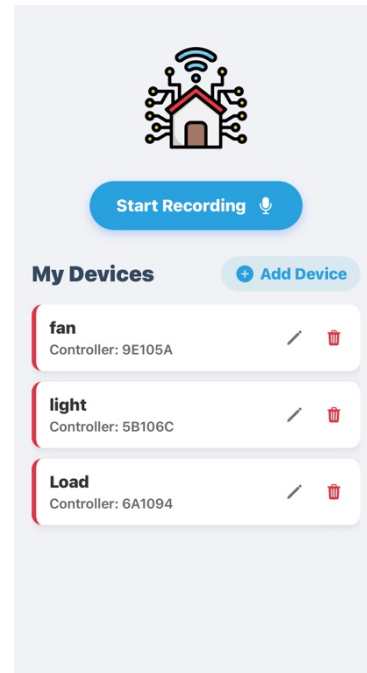
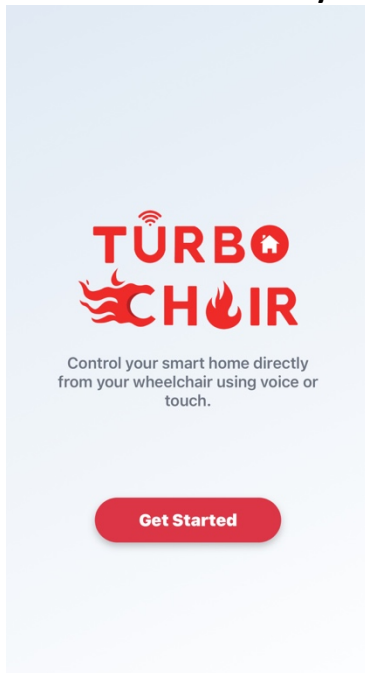
Voice command was tested, and user was able to record spoken commands, and the system was able to process the command and execute is directly with hands free interaction along with spoken feedback to make the system more interactive.



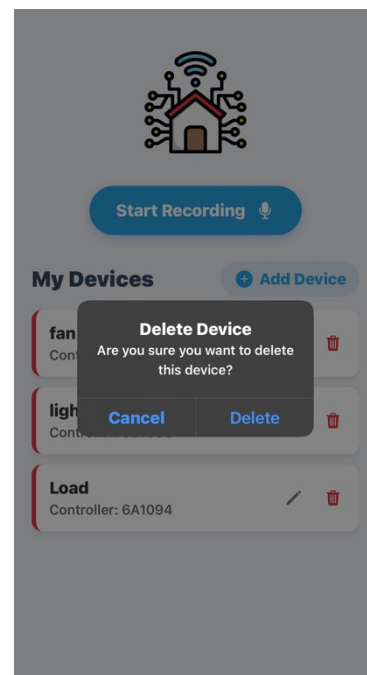
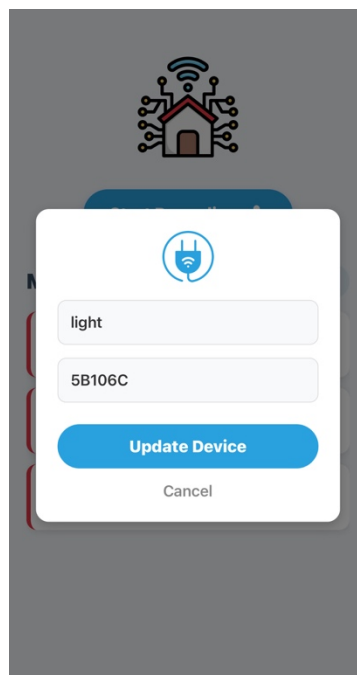
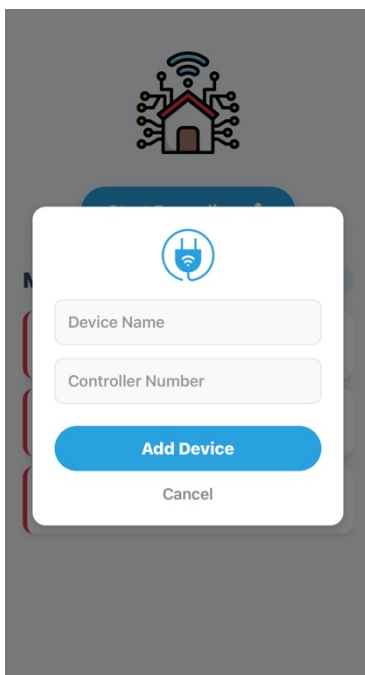
For more details for results and testing see the demo.

- **Mobile Application Operation**

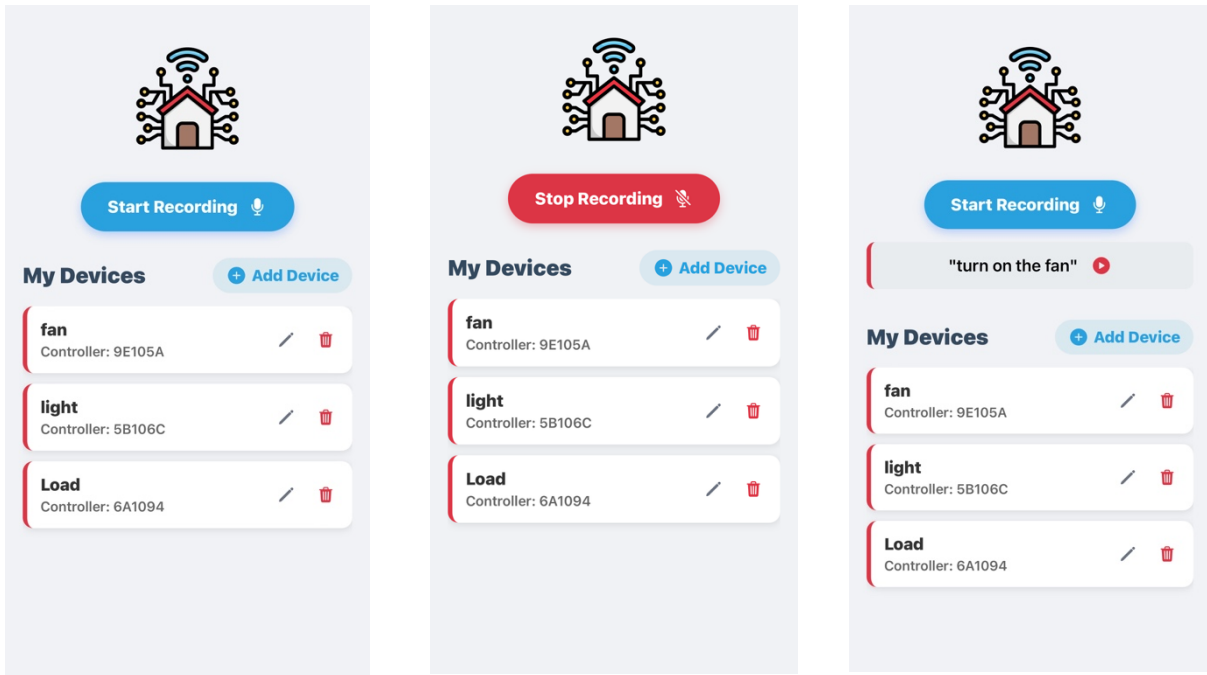
These are screenshots for welcome page and main page which displays the devices at the home of user and added manually.



The mobile application was tested and confirmed the app allows **add**, **edit**, and **delete** smart devices. When any update happens, it's communicated directly to the touch screen.



The mobile was tested for voice-based device control, the user was able to control the devices by recording audio, seeing the transcribed text, the command being executed on the device and spoken feedback.



Overall, the mobile app provided a smooth and responsive user experience for both local and remote control of smart devices.

Chapter 5

Conclusions and Recommendation

5.1 Summary

The Harmoni Move project shows how assistive technologies, the Internet of Things, and smart mobility systems can work together to make life better for people with severe physical disabilities. The system gives users more freedom, safety, and control over their environment by combining voice control, wireless communication, smart home automation, and a dedicated mobile app. At the end of this project a working prototype that meets the needs of real users was designed. This facilitates the motion for those peoples and allow them to control the house devices without need help from caregivers. The system's modular design lets it grow and change to meet the needs of different users and environments.

There were problems along the way, especially with calibrating sensors, keeping communication stable, and integrating hardware. However, the experience showed how important it is to design with the user in mind and test things all the time. The project not only met its technical goals, but it also had a real effect by making things easier to get to, move around, and live better.

In the end, Harmoni Move is a useful and inclusive option in the world of assistive technology. With more work, the system could become a full smart mobility ecosystem, with features like advanced obstacle detection, health monitoring. This would give users more freedom and confidence to live their lives.

5.2 Recommendations for Improvement

Improving the implemented smart wheelchair would strongly depends on feedback from disabled users who will try it and report any issue or concern to be considered on future work to guarantee user satisfaction. While based on the literature review and the study of this topic, there is a lot of gap areas that can be considered in improvement, including easier motion control and GPS tracking systems.

5.3 Lessons Learned

Working on this project has taught us a lot about user-centred design, IoT, embedded systems, and assistive technology. Starting with trying out different hardware parts like the ESP32, Arduino Mega, H-Bridge motor drivers, brushless motor controllers, DC motors, joysticks, and a 2.4" TFT touchscreen. Learning how to wire, power, make sure the voltage is compatible, map pins, control things in real time, and keep signals stable and program these parts.

We also learned about the ESP32's Wi-Fi features, MQTT protocol, and serial communication. And how to make sure that multiple microcontrollers can talk to each other reliably, as well as how to make data transfer between our devices and the Node.js backend as fast as possible.

Also, making the React Native mobile app was another important step, as was deploying the app and connecting it to real-time MQTT messages through the ESP32. Working with voice control, trying out voice-to-text processing APIs, and even testing joystick-based navigation logic. We had to deal with hardware noise, command misinterpretation, and signal delay along the way.

These problems made us add error handling, command confirmation, and user feedback systems.

5.4 Future Work and Directions

- Add camera-based vision for advanced obstacle detection.
- Integrate head movement controls for directional navigation.
- Support for Arabic language voice recognition.

References

- [1] E. Y. Kim, “Wheelchair Navigation System for Disabled and Elderly People,” *Sensors (Basel)*, vol. 16, no. 11, p. 1806, Nov. 2016, doi: 10.3390/S16111806.
- [2] V. K. Kumar and A. professor, “SMART ELECTRIC WHEEL CHAIR FOR HOME APPLIANCES CONTROL,” 2024, Accessed: Jun. 08, 2025. [Online]. Available: www.ijnrd.org
- [3] Lung C. F., Hsu C. M., and Lai C. S., “A Low-Cost Smart Wheelchair System for People with Severe Physical Disabilities,” *IEEE Trans Industr Inform*, vol. 14, no. 8, pp. 3529–3539, Aug. 2018.
- [4] Lung C. F., C. M. Hsu, and Lai and C. S., “A Novel Low-Cost Smart Wheelchair for Physically Disabled People,” *IEEE Trans Industr Inform*, vol. 13, no. 4, pp. 2054–2063, 2017.
- [5] C. F. Riman, “Multi-Controlled Wheelchair for Upper Extremities Disability,” *Journal of Mechatronics and Robotics*, vol. 2, no. 1, pp. 121–131, Jan. 2018, doi: 10.3844/jmrsp.2018.121.131.
- [6] S. U. Khadilkar and N. Wagdarikar, “Android phone controlled voice, gesture and touch screen operated smart wheelchair,” in *2015 International Conference on Pervasive Computing (ICPC)*, 2015, pp. 1–4. doi: 10.1109/PERVASIVE.2015.7087119.
- [7] D. Hercog, T. Lerher, M. Truntič, and O. Težak, “Design and Implementation of ESP32-Based IoT Devices,” *Sensors*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156739.
- [8] “Lithium-ion battery - Wikipedia.” Accessed: Jun. 09, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Lithium-ion_battery
- [9] “LM2596 Voltage Regulator Guide.” Accessed: Jun. 09, 2025. [Online]. Available: <https://www.ic-components.com/blog/LM2596-Voltage-Regulator-Guide.jsp>
- [10] “In-Depth: Interface Two Channel Relay Module with Arduino.” Accessed: Jun. 09, 2025. [Online]. Available: <https://lastminuteengineers.com/two-channel-relay-module-arduino-tutorial/>
- [11] “Electret Microphone Amplifier - MAX9814 with Auto Gain Control – Future Electronics Egypt.” Accessed: Jun. 09, 2025. [Online]. Available: https://store.future-electronics.com/products/electret-microphone-amplifier-max9814-with-auto-gain-control?srsId=AfmBOorgMVkGtV49IJB2jZILfYKJQmrRHpTo_By4TOaclUTaX_qf3koZ
- [12] “VHM-314 Bluetooth Audio Receiver Module - Alphontronic.” Accessed: Jun. 09, 2025. [Online]. Available: <https://alphontronic.lk/product/vhm-314-bluetooth-audio-receiver->

- module/?srsltid=AfmBOor0_eEstPzxp6EKQuIj3qB3DnJ2Ax51wVPBfjlq138yKWMWwmd4
- [13] “Arduino 2.4” Touch Screen LCD Shield Tutorial - Hackster.io.” Accessed: Jun. 09, 2025. [Online]. Available: <https://www.hackster.io/electropeak/arduino-2-4-touch-screen-lcd-shield-tutorial-fe6f05>
 - [14] “Mega 2560 Rev3 | Arduino Documentation.” Accessed: Jun. 09, 2025. [Online]. Available: <https://docs.arduino.cc/hardware/mega-2560/#suggested-libraries>
 - [15] S. A. Tafriahi, A. A. Ravankar, J. Victorio, S. Luces, and Y. Hirata, “A Novel Assistive Controller Based on Differential Geometry for Users of the Differential-Drive Wheeled Mobile Robots.”
 - [16] “Momentary Switch - Everything You Need to Know.” Accessed: Jun. 09, 2025. [Online]. Available: <https://www.langir.com/news/everything-about-momentary-switch/>
 - [17] “Brushless DC electric motor - Wikipedia.” Accessed: Jun. 09, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Brushless_DC_electric_motor
 - [18] “Exploring the Power and Precision of Brushless Motor Controllers - Ovaga Technologies.” Accessed: Jun. 09, 2025. [Online]. Available: <https://www.ovaga.com/blog/motor-control/exploring-the-power-and-precision-of-brushless-motor-controllers>
 - [19] “L298N Motor Driver - Arduino Interface, How It Works, Codes, Schematics.” Accessed: Jun. 09, 2025. [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/#h-h-bridge-dc-motor-control>
 - [20] “Speech to text - OpenAI API.” Accessed: Jun. 11, 2025. [Online]. Available: <https://platform.openai.com/docs/guides/speech-to-text>
 - [21] “Text to speech - OpenAI API.” Accessed: Jun. 11, 2025. [Online]. Available: <https://platform.openai.com/docs/guides/text-to-speech?lang=curl>

Appendix