Contents

**APPROVAL FOR BINDING OF GRADUATION PROJECT REPORT……………………………………………………….1**

**GRADUATION PROJECT THESIS LOW-COST POTHOLE DETECTION MODEL BASED ON IOT AND ARTIFICIAL INTELLIGENCE………………………………………………….............................................................2**

**FIGURES………………………………………………………………………………………………………………………………………..4**

**ACKNOWLEDGEMENTS………………………………………………………………………………………………………………….5**

**ABSTRACT……………………………………………………………………………………………………………………………………..6**

**KEYWORDS**: Microcontroller, RF Module, Ultrasonic Sensor, Bluetooth Module**…………………………….6**

**CHAPTER 1 INTRODUCTION………………………………………………………………………………………………………..…7**

**1.1 INTRODUCTION……………………………………………………………………………………………………………….8**

**1.2 PROBLEM BACKGROUND…………………………………………………………….....................................8**

**1.3 PROBLEM SOLUTION……………………………………………………………………………………………………....9**

**1.4 GOALS AND OBJECTIVES………………………………………………………………………………………………..10**

**1.5 Hardware Requirements………………………………………………………………………………………………..10**

**1.6 WORK BREAKDOWN STRUCTURE & GANTT CHART…………………………………………………….….11**

**1.6.1 Work Breakdown Structure…………………………………………………………………………………………...11**

**1.6.2 Gantt Chart…………………………………………………………………………………………………………………...11**

[**CHAPTER 2 PROJECT ANALYSIS:……………………………………………………………………………………………….…12**](#_bookmark17)

**2.1** [**DEVELOPMENT METHODOLOGY:…………………………………………………..………………………………..…….13**](#_bookmark18)

**2.2**[**HARDWARE REQUIREMENTS SPECIFICATION………………………………………………………………….…..….13**](#_bookmark20)

[**2.2.1- Arduino UNO board:………………………………………………………………………….………………………………13**](#_bookmark21)

[**2.2.2- HC-05 Bluetooth Module:……………………………………………………………………………………………………...14**](#_bookmark23)

[**2.2.3- Motor Driver:………………………………………….…………………………………………………………………………15**](#_bookmark25)

[**2.2.4- Ultrasonic Sensor:…………………………………………….……………………………………………………………….17**](#_bookmark27)

**2.2.4.1 Features……………………………………………………………………………………………………..17**

**2.2.4.2 How Does it Work?...........................................................................................................18**

**2.2.4.3 HC-SR04 Ultrasonic Sensor Pinout……………………………………………………………………………………19**

**2.2.5- LCD1602 Module………………………………………………………………………………………………………………19**

**2.2.6- Arduino breadboard…………………………………………………………………………………………………………21**

[**2.2.7- Batteries:…………………………………………………………………………………………………………………………..23**](#_bookmark29)

[**2.2.8- Two Tire Wheel with two-DC Motor:…………………………………………………………………………………23**](#_bookmark31)

[**2.2.9- Movable Caster Wheel:…………………………………………………………………………………………………..…23**](#_bookmark33)

**2.2.10- Dupont Wire & JumperWire:*………………………………..……………………………………………..……...*…24**

**2.2.11- Arduino bluecontrol…………………………………………………………………………………………………………25**

[**SOFTWARE REQUIREMENTS SPECIFICATION……………………………………………………………………………..…27**](#_bookmark36)

[**2.3.1-Arduino IDE………………………………………………………………………………………………………………………..27**](#_bookmark37)

[**2.3.2-Domain Awardspace:………………………………………………………………………………………………………...27**](#_bookmark38)

[**2.3.3-Bot Telegram:…………………………………………………………………………………………………………………….27**](#_bookmark39)

[**2.3.4-Language C Arduino……………………………………………………………………………………………………………27**](#_bookmark40)

[**CHAPTER 3 PROJECT DESIGN:………………………………………………………………………………………………….…28**](#_bookmark43)

**3.1 Architecture Design:………………………………………………………………………………………………………………29**

**3.1.1 Circuit Diagram:………………………………………………………………………………………………………………….29**

**3.1.2 Architecture Design………………………………………………………………………………..29**

**3.2- Data flow…………………………………………………………………………………………………………………………..…30**

**3.3 Database Design………………………………………………………………………………………………………………….…31**

[**CHAPTER 4 PROJECT IMPLEMENTATION AND RESULT……………………………………………….……………….32**](#_bookmark52)

**4.1-Project Implementation………………………………………………………………………………………………33**

**4.2- Project Code…………………………………………………………………………………………………………..…34**

**4.3-Testing………………………………………………………………………………………………………………………………….38**

**4.3.1-** **The chair:…………………………………………………………………………………….......................................39**

**4.3.2-** **Real screen shots of the working project………………………………………………………………..………….40**

**4.4- Arduino BlueControl :………………………………………………………………………………………………………….41**

**4.4.1 Application………………………………………………………………………………………………………………………..42**

**4.4.2-Bluetooth………………………………………………………………………………………………………………………….43**

**4.4.3- Voice control…………………………………………………………………………………...................................43**

**Chapter 5 CONCLUSION………………………………………………………………………………………………………………45**

**5.1 Summary………………………………………………………………………………………………………………………………46**

**5.2 Limitations and Future Work…………………………………………………………………………………………………46**

**5.2.1 Limitations…………………………………………………………………………………………………………………………46**

**5.2.2 Future Work……………………………………………………………………………………………………………………….47**

Tables

[*Table 1: HC-SR04 Ultrasonic Sensor.………………………………………………………………………..20*](#_bookmark35)

[*Table 2: Cost of Hardware.……………………………………………………………………………………24*](#_bookmark72)

[*Table 3: REFERENCES……………………………………………………………………………………………45*](#_bookmark90)

Figures

[*Figure 1: WBS Diagram……………………………………………………………………………………………11*](#_bookmark14)

[*Figure 2: Time Gantt Chart ……………………………………………………………………………………….11*](#_bookmark16)

[*Figure 3: Arduino UNO board……………………………………………………………………………………14*](#_bookmark19)

[*Figure 4: HC-05 Bluetooth Module……………………………………………………………………………..15*](#_bookmark22)

[*Figure 5: Motor Driver……………………………………………………………………………………………16*](#_bookmark24)

[*Figure 6: Ultrasonic Sensor1……………………………………………………………………………………17*](#_bookmark26)

[*Figure 7: Ultrasonic Sensor2……………………………………………………………………………………18*](#_bookmark28)

[*Figure 8: Ultrasonic Sensor3…………………………………………………………………………………....19*](#_bookmark30)

[*Figure 9: I2C LCD1602*](#_bookmark30)*………………………………………………………………………..………………..20*

[*Figure10: Arduino breadboard……………………………………………………………………………….....22*](#_bookmark30)

[*Figure 11: Batteries ……………………………………………………………………………………………....23*](#_bookmark30)

[*Figure 12: Two Tire Wheel with two-DC Motor ………………………………………………………….…..23*](#_bookmark30)

[*Figure 13: Movable Caster Wheel…………………………………………………………………………..…..24*](#_bookmark32)

[*Figure 14: Dupont Wire…………………………………………………………………………………….……24*](#_bookmark34)

[*Figure 15: : Jumper Wire………………………………………………………………………………….…… 24*](#_bookmark46)

*Figure 16:Arduino blutcontrol………………………………………………………………………… .……...26*

*Figure 17: Circuit Diagram…………………………………………………………………………….….…..29*

*Figure 18: Sensor is on………………………………………………………………………………...……....30*

[*Figure 19: The smart wheelchair is on……………………………………………………………………….....30*](#_bookmark49)

[*Figure 20: data flow Diagram*](#_bookmark51) *………………………………………………………………………………......30*

*Figure 21: Database Design…………………………………………………………………………………......31*

[*Figure 22: chair……………………………………………………………………………………...………...38*](#_bookmark32)

[*Figure 23: wheelchair during operation………………………………………………………………………39*](#_bookmark34)

[*Figure 24: Data display screen ……………………………………………………………………………… 39*](#_bookmark46)

*Figure 25: chair while stopping****.****……………………………………………………………………………….40*

*Figure 26: installation****.****……………………………………………………………………………………….…40*

[*Figure 27: Arduino BlueControl…………………………..…………………………………………………...41*](#_bookmark49)

*Figure 28: Application……………………………………………………………………………*…...…….*42*

*Figure 29: HC\_05****.****……………………………………………………………………………………...............42*

*Figure 30: voice control stop ………………………………………..........................................................43*

*Figure 31: voice control backward……………………………………………………………………...…….43*

A**CKNOWLEDGEMENTS**

We would like to acknowledge the contributions of all members of our group. Our cooperation and teamwork were instrumental in the successful completion of this project.

We would also like to extend our sincere thanks to our supervisor “Dr. Mohammed Rizwan Sheikh,” who provided invaluable guidance and support throughout the entire process. His experience and dedication helped us overcome the challenges of this project and bring it to a successful conclusion.

Additionally, I want to express my appreciation for the support of our families and friends, who have provided encouragement and motivation throughout this journey.

Finally, we would like to thank Dr. Mohammed Rizwan Sheikh once again for providing the resources and opportunities that made this project possible. This graduation project has been a valuable learning experience, and we are grateful for that.

To allow the opportunity to apply what we have learned in a real environment.

Thank you all for your continued support and guidance.

ABSTRACT

For a physically handicapped person we developed a voice and gesture controlled wheelchair. The person who is handicapped or elderly can use this technology. This system is using voice and gesture. In voice system we use commands such as FORWARD, BACKWARD, LEFT, RIGHT, STOP. We recognized speech and getting a successful recognition rate of 99.03% to 98.3% with this we are also using gesture controlling movement of wheelchair for this we use ultrasonic sensor. The ultrasonic sensor is directly connected to the microcontroller ARDUINO which is connected to the BLUETOOTH (HT 05 MODULE) this IC is connected to the RF Transmitter module this transmits the data wirelessly. The motor gets the signal and then it will operate accordingly. The purpose of this system is to implement the direction control wheelchair with voice and gesture.

**KEYWORDS**: Microcontroller, RF Module, Ultrasonic Sensor, Bluetooth Module.

# **Chapter 1**

## **INTRODUCTION**

### **Introduction**

Smart wheelchair is a type of assistive technology that allows individuals with physical disabilities to control the movement and operation of their wheelchair using their voice. These innovative devices are designed to provide greater independence and mobility for individuals who may not be able to operate a traditional manual wheelchair.

Smart wheelchairs use advanced technology such as voice recognition software and sensors to interpret and respond to voice commands.

One of the key benefits of a smart wheelchair is that it eliminates the need for physical controls or manual operation, which can be challenging for individuals with limited mobility in their hands or arms. This technology provides greater freedom and independence for users, allowing them to move around and navigate their environment with ease.

Smart wheelchairs can be customized to meet the specific needs and preferences of individual users. The wheelchair is equipped with a voice recognition system that allows the user to control its movement, speed, and direction by speaking specific commands. The voice control system is designed to recognize a wide range of commands, including directional commands such as "go forward," "turn left," and "turn right," The system can also be programmed to recognize custom commands based on the user's specific needs and preferences.

### **Problem Background**

Although smart wheelchairs offer many advantages and can significantly improve the quality of life of people with mobility impairments, there are also some potential issues that may arise. Here are some examples of these problems:

1. Cost: A smart wheelchair may be more expensive than a conventional wheelchair due to the addition of sensors and other electronic components. This can make them less accessible to individuals who cannot afford the higher cost.

2. Reliability: Smart wheelchairs rely on electronic components to function properly, which can sometimes lead to reliability issues. For example, if a sensor malfunctions or the battery dies, the wheelchair may not operate as expected.

3. Complexity: Smart wheelchairs can be more complex than traditional wheelchairs, making them more difficult to operate and maintain. This challenge can be especially difficult for people with limited technical skills or intellectual disabilities.

4. Security: Smart wheelchairs may be vulnerable to hacking or other security threats, which could endanger the safety and privacy of the user.

5. Compatibility: The smart wheelchair may not be compatible with some environments or infrastructure, such as old buildings with narrow doors or uneven terrain.

While it is important to consider these issues, it is worth remembering that many of them can be addressed through proper design, testing, and maintenance. In addition, the benefits of smart wheelchairs may outweigh the potential problems for many individuals with mobility impairments.

### **Problem Solution**

To address the problems of the smart wheelchair, there are several solutions that can be implemented, including:

1. Affordable design: To make the smart wheelchair more affordable, designers can use cost-effective materials and components, and simplify the design without compromising functionality.

2. Redundancy: To improve the reliability of the smart wheelchair, designers can incorporate redundancy in critical components, such as backup batteries or sensors.

3. User-friendly interface: To make the smart wheelchair easier to use, designers can create a user-friendly interface that is easy to navigate, with clear instructions on how to operate the device.

4. Cybersecurity measures: To prevent cybersecurity threats, designers can implement security measures such as encryption, firewalls, and intrusion detection systems.

5. Terrain compatibility: To make the smart wheelchair more compatible with different terrains, designers can incorporate features such as adjustable height and suspension systems.

6. Regular maintenance and testing: To ensure the optimal performance of the smart wheelchair, regular maintenance and testing should be performed to detect any issues before they become major problems.

7. Technical support and training: To help users operate and maintain the smart wheelchair, technical support and training should be provided to ensure that they are familiar with the device and can troubleshoot any issues that arise.

Overall, by implementing these solutions, the smart wheelchair can become more accessible, dependable, and user-friendly, improving the quality of life for individuals with mobility impairments. It is also important for designers to continue researching and developing new technologies to further improve the functionality and performance of smart wheelchairs.

### **Goals and Objectives**

Ease of use: A smart wheelchair should be easy to use, even for people with limited mobility or dexterity. The wheelchair should be able to recognize a variety of voice commands and respond quickly and accurately.

Customization: A smart wheelchair should be customizable to meet the specific needs of the user. This may include adjusting the speed of the wheelchair, setting up custom voice commands, or configuring the wheelchair to work with other assistive devices.

Safety: A smart wheelchair should be designed with safety in mind. It should include features such as obstacle detection and avoidance, stability control, and emergency stop buttons to prevent accidents and ensure the user's safety.

Comfort: A smart wheelchair should be comfortable to use for extended periods of time. This may include features such as adjustable seating, cushioning, and armrests.

Durability: A smart wheelchair should be durable and able to withstand the rigors of daily use. It should be constructed from high-quality materials and designed to withstand bumps, scrapes, and other minor impacts.

Portability: A smart wheelchair should be portable and easy to transport. It should be lightweight and compact enough to fit in a car or other vehicle and should be easy to disassemble and reassemble for storage or transport.

### **Hardware Requirements**

1-Arduino UNO Board

* Arduino simulator (Arduino IDE)

2-HC-05 Bluetooth Module

3-Motor Driver

4-Ultrasonic Sensor

5-LCD1608

6-Batteries

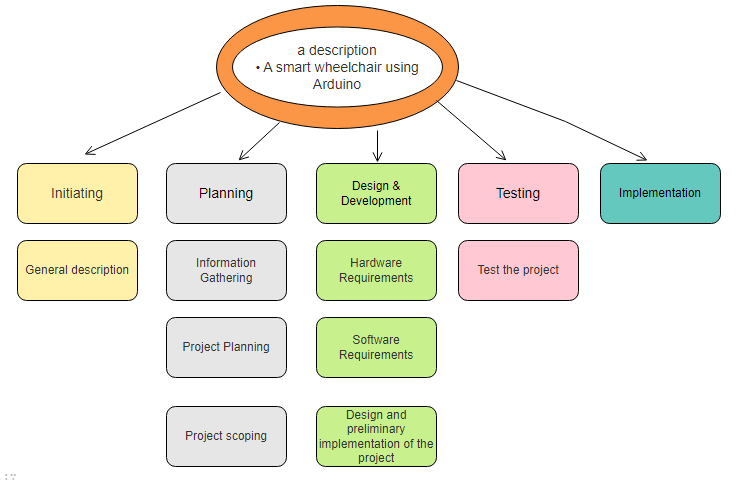
7-Two Tire Wheel with two-DC Motor

8-Movable Caster Wheel

9-Dupont Wire & Jumper Wire

### **Work Breakdown Structure & Gantt Chart**

##### 1.6.1 Work Breakdown Structure



***Figure 1: WBS Diagram***.

صورة تحتوي على لقطة شاشة, نص, برمجيات, خط

تم إنشاء الوصف تلقائياً**1.6.2 Gantt Chart**

***Figure 2: Time Gantt Chart***

**Chapter 2**

**PROJECT ANALYSIS**

**2.1 Development Methodology**

***Chapter 2***

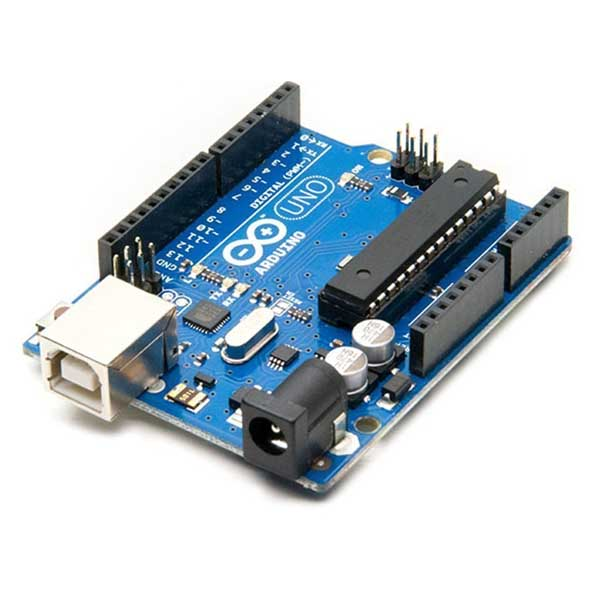
The goal of a smart wheelchair is to enhance the freedom of movement and independence of people with mobility impairments, and to improve their quality of life and well-being. The smart wheelchair relies on computing technologies, artificial intelligence, and robotics to improve its performance, maneuverability, and obstacle avoidance. The smart wheelchair can be controlled by the user independently via a console or controlled remotely through an application on a mobile phone or computer. The smart wheelchair has the ability to provide more freedom and flexibility to users, as they can reach places, they were not able to reach before, move easily in tight spaces and maneuver in crowded places. The smart wheelchair can also provide more comfort and luxury for users, as they can control sitting, lying and standing positions, and the chair can also provide additional features such as talking to other people or controlling lighting and air conditioning. In addition, the smart wheelchair can contribute to reducing the burden on caregivers and families, as users can rely more on themselves and reduce the need for assistance. In general, a smart wheelchair can help improve the quality of life for people with mobility impairments, increase their independence, and enable them to participate more in daily and community activities.

### **2.2Hardware Requirements Specification**

**2.2.1- Arduino UNO board:**

The Arduino Uno is a popular microcontroller board based on the ATmega328P microcontroller. It is widely used in electronics projects and prototyping due to its simplicity and versatility. Here's some information about the Arduino Uno board:

1. Microcontroller: The Arduino Uno is built around the ATmega328P microcontroller, which operates at 5 volts.
2. Digital I/O Pins: It has 14 digital input/output pins, of which 6 can be used as PWM (Pulse Width Modulation) outputs.
3. Analog Inputs: The board has 6 analog input pins, labeled A0 through A5. These pins can also be used as digital I/O pins.
4. Clock Speed: The ATmega328P runs at a clock speed of 16 MHz.
5. Power Supply: The Arduino Uno can be powered through a USB connection or an external power supply. The recommended input voltage range is 7 to 12 volts.
6. Programming: The Arduino Uno can be programmed using the Arduino IDE (Integrated Development Environment). It uses a simplified version of the C++ programming language.
7. Communication: The board has a USB interface, which can be used for uploading sketches (programs) and for serial communication with a computer. It also has an ICSP (In-Circuit Serial Programming) header for advanced programming and debugging.
8. Memory: The ATmega328P microcontroller on the Arduino Uno has 32 KB of flash memory for storing the program code, 2 KB of SRAM for variables and data, and 1 KB of EEPROM for non-volatile storage.
9. Shields: Arduino Uno is compatible with a wide range of shields, which are add-on boards that extend its functionality. Shields are available for various purposes, including motor control, wireless communication, display, and sensor integration.



***Figure 3: Arduino UNO board.***

**2.2.2- HC-05 Bluetooth Module:**

The HC-05 is a popular Bluetooth module that can be used to add wireless communication capabilities to your projects. Here's some information about the HC-05 Bluetooth module:

1. Bluetooth Version: The HC-05 module is based on Bluetooth 2.0 + EDR (Enhanced Data Rate) specification. It supports the Serial Port Profile (SPP), which allows it to act as a wireless serial communication link between devices.
2. Communication Range: The module has a typical communication range of approximately 10 meters (33 feet) in an open space. The range may vary depending on the environment and obstacles.
3. Operation Modes: The HC-05 module supports two main operation modes: Master mode and Slave mode. In Slave mode, it can be paired with a Bluetooth-enabled device such as a smartphone or a computer. In Master mode, it can initiate connections with other Bluetooth devices.
4. Serial Communication: The HC-05 module communicates with the microcontroller or other devices using a serial UART (Universal Asynchronous Receiver-Transmitter) interface. It has a transmit (TX) pin and a receive (RX) pin that can be connected to the corresponding UART pins of your microcontroller.
5. AT Commands: The module can be configured and controlled using AT commands. These commands allow you to change settings such as the Bluetooth device name, pairing password, baud rate, and more. The commands are sent over the UART interface by sending specific strings of characters.
6. Power Supply: The HC-05 module typically operates at 3.3 volts. It requires a power supply of 3.3V to 5V. Make sure to provide the appropriate voltage level based on your specific module.
7. Connection Pins: The HC-05 module usually has six pins: VCC (power supply), GND (ground), TX (transmit), RX (receive), STATE, and EN (enable). The STATE pin can be used to determine the module's current operating mode (Master or Slave). The EN pin is used to switch between AT command mode and data transmission mode.
8. Pairing and Security: The HC-05 module can be paired with other Bluetooth devices by entering a pairing password (also known as a PIN code). The default password is often "1234," but it can be changed using AT commands for increased security.

The HC-05 Bluetooth module is widely used in various applications such as robotics, home automation, wireless sensor networks, and more. It provides a convenient and straightforward way to establish wireless communication between devices using Bluetooth technology.



***Figure 4 HC-05 Bluetooth Module:***

**2.2.3- Motor Driver:**

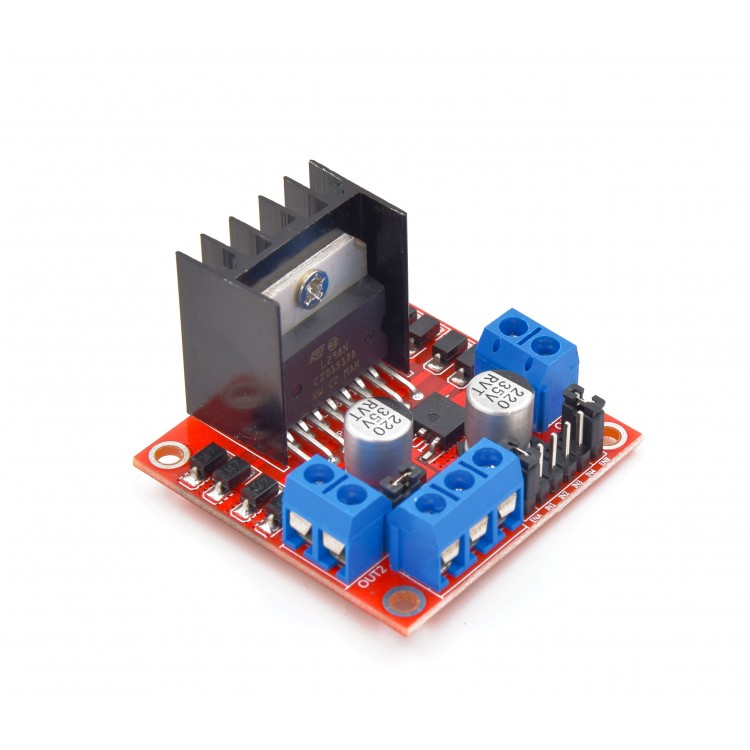
A motor driver is an important component in a smart wheelchair as it is responsible for controlling the movement of the motors that drive the wheels of the wheelchair. The motor driver receives signals from the microcontroller or other control circuitry and converts them into signals that control the speed and direction of the motors.

In a smart wheelchair, the motor driver needs to be able to control the movement of the wheelchair in a precise and responsive manner. It should be able to handle a range of inputs from sensors and other control signals and be able to adjust the speed and direction of the motors in real-time.

In addition, the motor driver in a smart wheelchair should be designed to handle the specific requirements of the wheelchair, such as the weight of the user, the terrain and surface on which it is being used, and any obstacles that may be encountered. It should also be designed to operate efficiently and reliably, with low power consumption and minimal heat generation.

There are several types of motor drivers that can be used in a smart wheelchair, including H-bridge motor drivers, motor driver boards, and integrated motor drivers. The choice of motor driver will depend on the specific requirements of the wheelchair and the type of motors being used.

1. Types of Motors: Motor drivers are designed to work with different types of motors, such as DC motors, stepper motors, and brushless DC motors. Each type of motor requires a specific driver circuit or module.
2. DC Motor Driver: A DC motor driver is used to control the speed and direction of a DC motor. It typically consists of power transistors (such as MOSFETs) or H-bridge circuits that can switch the motor's power supply on and off, controlling its rotation direction and speed. DC motor drivers often include features like current sensing, overcurrent protection, and braking functionality.
3. Stepper Motor Driver: Stepper motor drivers are specifically designed for controlling stepper motors, which are commonly used in applications requiring precise positioning or motion control. Stepper motor drivers send a sequence of electrical pulses to the motor's windings, causing it to step incrementally. They provide control overstep size, speed, and direction.
4. Brushless DC Motor (BLDC) Driver: BLDC motor drivers are used to control brushless DC motors, which offer higher efficiency and reliability compared to brushed DC motors. These drivers typically employ electronic commutation techniques, such as sensor-based or sensorless control, to synchronize the motor's phases and control its speed and direction.
5. Control Interfaces: Motor drivers can have various control interfaces, including analog voltage control, pulse-width modulation (PWM) control, and serial communication (such as I2C or SPI). The choice of control interface depends on the motor driver module and its compatibility with the controlling device.
6. Current and Voltage Ratings: Motor drivers are designed to handle specific current and voltage levels based on the motors they are intended to drive. It is crucial to select a motor driver that can handle the power requirements of your motor to ensure safe and reliable operation.
7. Protection Features: Many motor drivers include built-in protection features to safeguard against faults and ensure motor and driver longevity. These features may include overcurrent protection, overtemperature protection, under-voltage lockout, short-circuit protection, and reverse polarity protection.



***Figure 5: Motor Driver***

**2.2.4- Ultrasonic Sensor:**

The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this module comes with ultrasonic transmitter and receiver modules.

The following picture shows the HC-SR04 ultrasonic sensor.



***Figure 6: Ultrasonic Sensor1.***

### **2.2.4.1 Features**

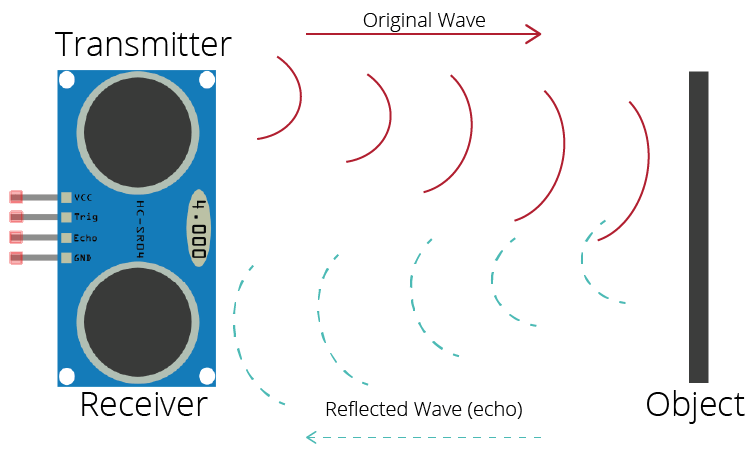
Here’s a list of some of the HC-SR04 ultrasonic sensor features and specs—for more information, you should consult the sensor’s datasheet:

* Power Supply: +5V DC
* Quiescent Current: <2mA
* Working Current: 15mA
* Effectual Angle: <15°
* Ranging Distance: 2cm – 400 cm/1″ – 13ft
* Resolution: 0.3 cm
* Measuring Angle: 30 degrees
* Trigger Input Pulse width: 10uS TTL pulse
* Echo Output Signal: TTL pulse proportional to the distance range
* Dimension: 45mm x 20mm x 15mm

### **2.2.4.2 How Does it Work?**

The ultrasonic sensor uses sonar to determine the distance to an object. Here’s what happens:

1. The ultrasound transmitter (trig pin) emits a high-frequency sound (40 kHz).
2. The sound travels through the air. If it finds an object, it bounces back to the module.
3. The ultrasound receiver (echo pin) receives the reflected sound (echo).



***Figure 7: Ultrasonic Sensor2.***

The time between the transmission and reception of the signal allows us to calculate the distance to an object. This is possible because we know the sound’s velocity in the air. Here’s the formula:

distance to an object = ((speed of sound in the air) \*time)/2

speed of sound in the air at 20ºC (68ºF) = **343m/s**

### **2.2.4.3** ultra**HC-SR04 Ultrasonic Sensor Pinout**

***Figure 8: Ultrasonic Sensor3.***

|  |  |
| --- | --- |
| VCC | Powers the sensor (5V) |
| Trig | Trigger Input Pin |
| Echo | Echo Output Pin |
| GND | Common GND |

Here’s the pinout of the HC-SR04 Ultrasonic Sensor.

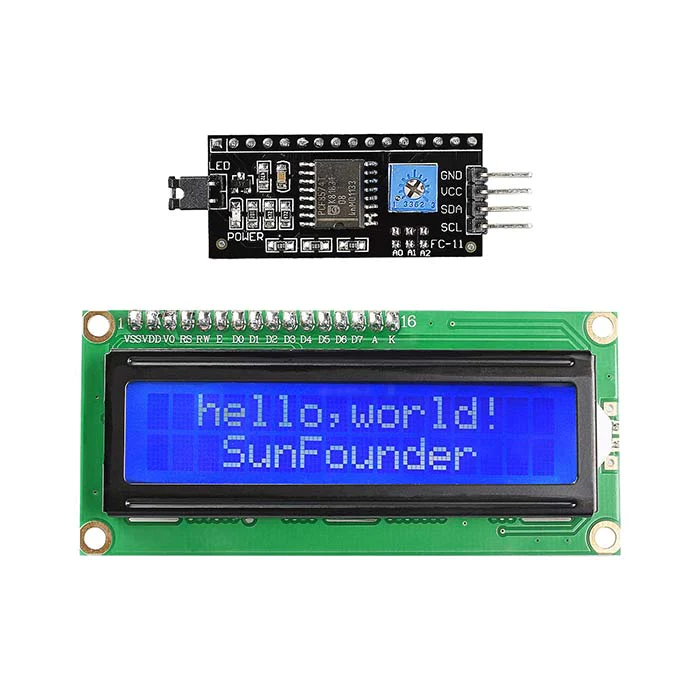
***Table 1: HC-SR04 Ultrasonic Sensor.***

**2.2.5- LCD1602 Module**

The I2C LCD1620 is a variant of the 16x2 LCD module that utilizes I2C (Inter-Integrated Circuit) communication protocol for interfacing with a microcontroller or other devices. It combines the standard 16x2 LCD display with an I2C backpack module, reducing the number of required pins for communication.

1. Display: The I2C LCD1620 module features a 16-character by 2-line display, similar to the standard 16x2 LCD module. It can display alphanumeric characters, symbols, and special characters.
2. Communication Interface: The module uses the I2C protocol to communicate with the controlling device. It includes an I2C backpack module that acts as an interface between the LCD module and the microcontroller. The backpack module typically includes an I2C-to-parallel converter chip, such as PCF8574 or PCF8574A.
3. I2C Address: The I2C LCD1620 module has a specific I2C address assigned to it. This address allows the microcontroller to communicate with the module over the I2C bus. The I2C address can be set or adjusted using hardware jumpers or solder pads on the backpack module.
4. I2C Communication: With the I2C backpack module, the module requires only two wires (SDA and SCL) for communication, along with the power (VCC) and ground (GND) connections. The I2C protocol allows for multiple devices to share the same bus using unique addresses.
5. Library and Code: To interface with the I2C LCD1620 module, you can use libraries specifically designed for I2C-based LCD modules. These libraries abstract the low-level I2C communication and provide convenient functions to control the display, write text, and set cursor positions. Popular libraries include LiquidCrystal\_I2C and Adafruit\_LiquidCrystal.
6. Power Supply: The module typically operates at a voltage of 5 volts. It requires a power supply connection (VCC) and a ground connection (GND) for proper operation.
7. Contrast Adjustment: The I2C LCD1620 module may include a potentiometer for adjusting the contrast of the displayed characters. By turning the potentiometer, you can optimize the contrast level for better visibility.
8. Backlight: Some I2C LCD1620 modules come with an integrated backlight, which can be controlled separately from the text display. The backlight can be turned on or off, or its intensity can be adjusted using software commands.

The I2C LCD1620 module simplifies the wiring and reduces the number of pins required for interfacing with the LCD. By utilizing the I2C protocol, it allows for easier integration with microcontrollers that have limited available I/O pins. However, it's essential to configure the I2C address correctly and use the appropriate library to communicate with the module effectively.

****

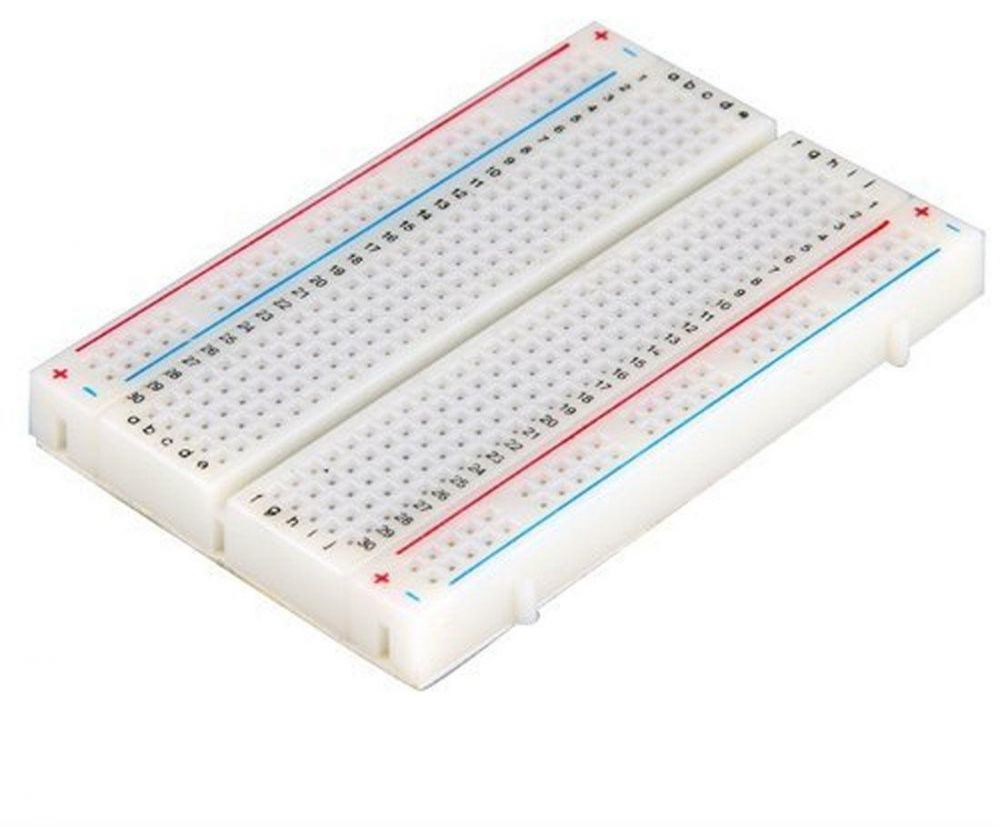
***Figure 9: I2C LCD1602***

**2.2.6-** **Arduino breadboard**

An Arduino breadboard refers to a breadboard used for prototyping and connecting Arduino components and circuits. A breadboard is a fundamental tool in electronics that allows you to build and test circuits without soldering. It consists of a rectangular plastic board with numerous holes arranged in rows and columns.

To use an Arduino breadboard effectively, follow these steps:

1. Gather the necessary components: This includes the Arduino board, electronic components such as resistors, capacitors, LEDs, sensors, and jumper wires. You will also need a power supply, such as a USB cable or a battery pack.
2. Place the Arduino board: Position the Arduino board in the desired location on the breadboard. Align the Arduino's pins with the appropriate rows on the breadboard, ensuring that the pins make contact with the metal strips inside the breadboard.
3. Connect power and ground: Insert jumper wires into the breadboard, connecting the 5V and GND pins of the Arduino board to the respective power rails on the breadboard. This provides power and a common ground reference for your circuits.
4. Connect components: Insert the desired electronic components into the breadboard, making connections as necessary. Use jumper wires to connect the component's pins to specific rows on the breadboard, creating the desired circuit connections.
5. Make use of breadboard columns: The columns on the breadboard are interconnected vertically, allowing you to easily establish electrical connections between components. Components connected to the same column are electrically connected, so you can place components in different rows within the same column to connect them.
6. Use jumper wires: Utilize jumper wires to establish connections between components and between the Arduino board and the breadboard. Strip the ends of the jumper wires and insert them into the appropriate holes on the breadboard to establish the necessary connections.
7. Test the circuit: Once you have connected all the components and made the necessary connections, connect the Arduino board to a power source (USB cable or battery pack) and upload the corresponding code to the Arduino using the Arduino IDE. This will allow you to test the circuit and observe the behavior of the connected components.



***Figure 10:*** ***Arduino breadboard***

**2.2.7- Batteries:**

Batteries are an essential component of a smart wheelchair as they provide the power needed to operate the motors, sensors, and other electronic components. The choice of battery depends on several factors, including the power requirements of the wheelchair, the expected operating time, and the weight and size constraints.

In addition to selecting the right battery, it is important to properly maintain and care for it to ensure optimal performance and lifespan. This includes following the manufacturer's recommendations for charging and discharging, avoiding extreme temperatures, and monitoring the battery's health and capacity over time.

Overall, batteries are a critical component of a smart wheelchair, and careful consideration should be given to their selection, maintenance, and integration into the overall wheelchair design. A properly selected and maintained battery can help ensure the reliability, safety, and performance of the smart wheelchair, and provide the user with the freedom and independence they need to navigate their environment.



***Figure 11: Batteries.***

**2.2.8- Two Tire Wheel with two-DC Motor:**

A two-wheeled smart wheelchair with two DC motors is a common configuration for motorizing a wheelchair. In this configuration, each wheel is driven by a separate DC motor, which allows for greater control and maneuverability of the wheelchair.

The DC motors are typically controlled by a motor driver circuit, which receives signals from the microcontroller or other control system and adjusts the speed and direction of the motors. The motor driver circuit can be programmed to provide precise control over the movement of the wheelchair, including speed, acceleration, and deceleration.

Overall, a two-wheeled smart wheelchair with two DC motors is a versatile and effective configuration for motorizing a wheelchair, providing greater control and maneuverability for the user, and improving their independence and quality of life.



***Figure 12: Two Tire Wheel with two-DC Motor***.

**2.2.9- Movable Caster Wheel:**

A movable caster wheel is a type of wheel that is commonly used in smart wheelchairs to provide increased maneuverability and stability. A caster wheel typically consists of a wheel mounted on a fork with a swivel joint, which allows the wheel to rotate freely in any direction.



***Figure 13: Movable Caster Wheel.***

**2.2.10- Dupont Wire & JumperWire:**

Dupont wire and jumper wire are two types of electrical wires that are commonly used in electronics projects, including those related to smart wheelchairs.

Dupont wire is a type of wire that is typically used to connect components on a breadboard or other prototyping board. It consists of a single insulated wire with a connector at each end, which can be inserted into the holes on a breadboard or other connector.

Jumper wire, on the other hand, is a type of wire that is used to create temporary connections between components. It typically consists of a thin wire with a connector at each end, which can be inserted into the pins on electronic components or other connectors.

Both Dupont wire and jumper wire are used to create electrical connections between components in a smart wheelchair, such as sensors, motors, and control systems. They are typically available in a variety of lengths, colors, and connector types, making it easy to customize the wiring for a specific project.

When working with Dupont wire and jumper wire, it is important to ensure that the connections are secure and that the wires are not exposed or touching each other, which can cause short circuits or other electrical issues. It is also important to use the appropriate gauge wire for the current being carried, and to avoid excessive bending or twisting of the wires, which can cause them to break or become damaged over time.



***Figure 14: Dupont Wire. Figure 15: Jumper Wi***

**2.2.11- Arduino bluecontrol**

To implement Bluetooth control using Arduino, you can use an Arduino board along with a Bluetooth module such as the HC-05 or HC-06. Here's a general outline of the steps involved:

1. Hardware Setup:

* Connect the Bluetooth module to the Arduino board. For example, connect the TX pin of the Bluetooth module to the RX pin of the Arduino board, and the RX pin of the Bluetooth module to the TX pin of the Arduino board. Make sure to also connect the module's VCC pin to a 5V power source and the GND pin to ground.

1. Software Setup:
   * Install the Arduino IDE (Integrated Development Environment) on your computer if you haven't already.
   * Open the Arduino IDE and create a new sketch.
   * Include the necessary libraries for Bluetooth communication. For example, if you are using the HC-05 module, you can include the SoftwareSerial library for serial communication with the module.
   * Set up the Bluetooth module by configuring the serial communication parameters (baud rate, parity, etc.) using appropriate commands. Refer to the documentation of your Bluetooth module for specific commands and configurations.
2. Bluetooth Communication:
   * Set up a serial communication interface between the Arduino board and the Bluetooth module. For example, you can use the Software Serial library to create a virtual serial port on specific Arduino pins.
   * Initialize the serial communication with the Bluetooth module using the appropriate baud rate.
   * Use the Serial.read() function to receive data from the Bluetooth module. This function will read any incoming data from the Bluetooth module and store it in a variable.
   * Process the received data based on your requirements. For example, you can define specific commands or data formats to control different functions or actions of your Arduino project. You can use conditionals (if-else statements) or switch-case statements to perform different actions based on the received commands.
   * Use the Serial.print() or Serial.write() functions to send data or feedback back to the Bluetooth device. This allows you to provide responses or send data from the Arduino back to the connected device.
3. Testing and Integration:
   * Upload the Arduino sketch to your Arduino board.
   * Pair your Bluetooth-enabled device (e.g., smartphone, tablet, or computer) with the Bluetooth module.
   * Use a Bluetooth terminal app on your device to establish a connection with the Bluetooth module.
   * Send commands or data from the Bluetooth terminal app to the Arduino and verify that the Arduino responds accordingly.



***Figure 16: Arduino blutcontrol***

|  |  |
| --- | --- |
| **Piece** | **Cost** |
| **Arduino UNO Board** | **48.96 SAR** |
| **HC-05 Bluetooth Module** | **45.93 SAR** |
| **Motor Driver** | **27 SAR** |
| **Ultrasonic Sensor**  **I2C-LCD1602** | **14.99 SAR**  **37 SAR** |
| **Batteries** | **37 SAR** |
| **Tow Tire Wheel with tow-DC Motor** | **35 SAR** |
| **Movable Caster Wheel**  **Dupont Wire & Jumpe Wire**  **Total** | **13.6 SAR**  **16.39 SAR**  **290.86 SAR** |

***Table 2: Cost of Hardware.***

**2.3 Software Requirements Specification**

**2.3.1-Arduino IDE:**

The Arduino Integrated Development Environment (IDE) is a software application that provides users with a platform to write, upload and debug code for their Arduino boards. Benefits:

a. User-friendly interface for beginners

b. Supports a wide range of boards and libraries

c. Simplifies the process of programming and uploading code to the board.

**2.3.2-Domain Awardspace:**

Awardspace is a domain registration and web hosting service that provides users with a platform to create and host their websites and database.

Benefits:

a. Free

b. Easy-to-use website builder

c. 24/7 support available.

**2.3.3-Bot Telegram:**

Telegram bots are automated programs that interact with users on the Telegram messaging platform. They can be used to perform various tasks, such as sending messages, answering questions, or providing information.

Benefits:

a. High level of customization

b. Wide range of functions and capabilities

c. Ability to interact with users in real-time.

**2.3.4-Language C Arduino:**

C is the programming language used to write code for the Arduino platform. It is a high- level language that is easy to learn and widely used in the electronics and robotics community.

Benefits:

a. Widely used in the electronics and robotics communities

b. Easy to learn for beginners

c. Supports a wide range of functions and libraries.

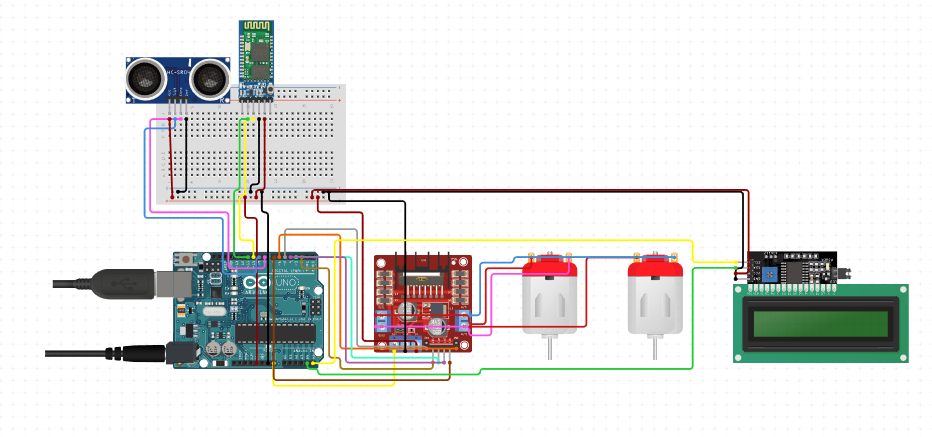
# Chapter 3

## PROJECT DESIGN

# **3.1 Architecture Design**

**3.1.1 Circuit Diagram**

A circuit diagram of an IoT project represents a graphical representation of the physical components of the project and the electrical connections between them. It helps in understanding the functionality and interconnections of the components used in the project.



***Figure 17: Circuit Diagram.***

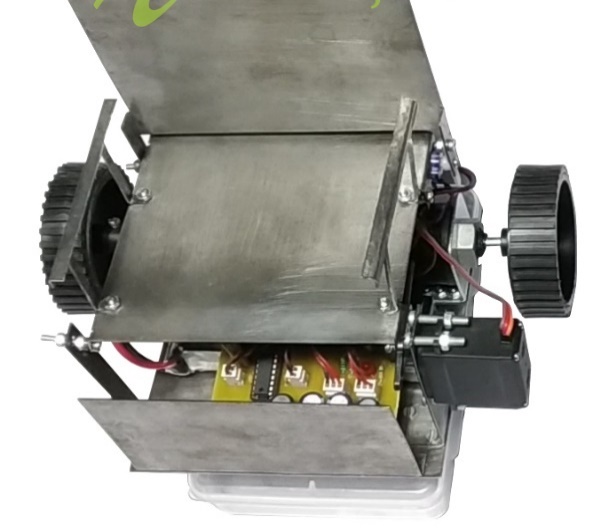
**3.1.2 Architecture Design.**

The smart wheelchair shall be designed in such a way as to meet the needs of the users and provide them with freedom and independence of movement and mobility, taking into account safety, security, efficiency, reliability and ease of use and maintenance. The system should also be designed in a way that facilitates the integration of the smart wheelchair with other smart devices and available smart services, such as smart phone applications, wearable devices, and smart home devices. APIs can be provided to help developers develop new applications and services that take advantage of the smart wheelchair's features.

Designing a smart wheelchair can improve the lives of people with disabilities, providing them with freedom, independence and inclusion in society.

صورة تحتوي على الأسلاك الكهربائية, كابل, الهندسة الإلكترونية, الإلكترونيات

تم إنشاء الوصف تلقائياً

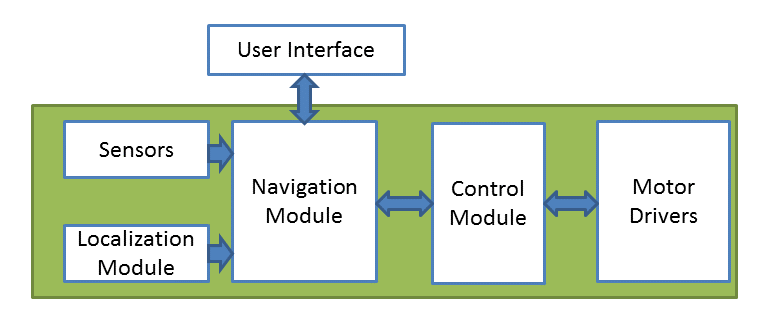


***Figure 18: Sensore is on*** ***Figure 19: The smart wheelchair is on***

**3.2- Data flow**

support disabled people. Smart wheelchairs extend the capabilities of standard power wheelchair by introducing control and navigational intelligence. A smart wheelchair normally consists of a standard. powered wheelchair base, a computer and sensors. The Smart wheelchair minimizes the physical and cognitive load required. in steering it. Smart Wheelchair is controlled by a computer. and a set of sensors. It based on techniques derived from mobile robotics research. A computer processes the sensor.

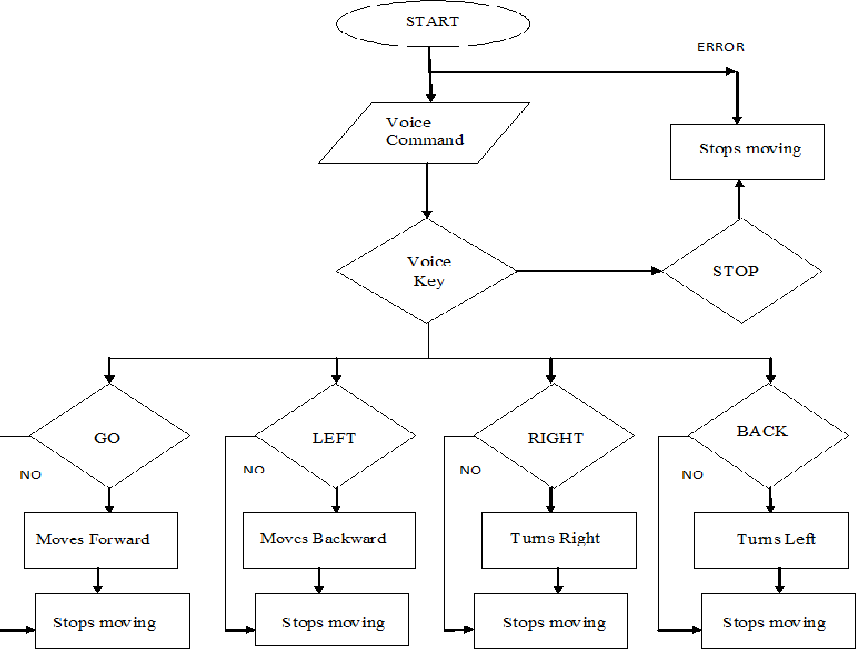
information and it generates the motor commands in an automatic way or with a shared control. The control module of smart wheelchair may consist of a standard wheelchair. joystick, speech-based control, facial expressions or even gaze control etc**.**



***Figure 20: data flow Diagram.***

**3.3 Database Design**

An embedded system is designed. The support included by the number of persons with physically incapacitated functions can account for this ring in the wheelchair. The voice command is sent through a cellular device that has bluetooth and the command is transmitted and converted into a string by the BT voice control for Arduino and transmitted to the bluetooth module SR-04, Arduino board to control the wheelchair. For example, when he makes him look right and right. This research was designed at a pre-patient cost, time and energy level. The ultrasonic control device, which appears on the counter ofthe cars waiting for you in the way of the wheelchair, also appears.



***Figure 21: Database Design.***

# **Chapter 4**

**PROJECT IMPLEMENTATION AND RESULT**

**4.1-Project Implementation**

Implementing a voice-controlled wheelchair involves integrating voice recognition technology with the wheelchair's control system. Here's an outline of the steps you can follow to implement a voice control feature for a wheelchair:

1. Voice Recognition System: Choose a suitable voice recognition technology or platform that can convert spoken commands into text or actionable commands. Popular options include cloud-based services like Amazon Alexa, Google Assistant, or local voice recognition libraries like PocketSphinx or CMU Sphinx.
2. Microphone and Audio Processing: Connect a microphone to capture the user's voice commands. Use an audio processing system to filter and amplify the audio signal, ensuring clear and reliable voice recognition.
3. Speech-to-Text Conversion: Send the captured audio data to the voice recognition system for speech-to-text conversion. The voice recognition system will process the audio and return the recognized text representing the user's spoken command.
4. Command Interpretation: Develop an algorithm or logic to interpret the recognized text command and map it to appropriate actions for controlling the wheelchair. Define a set of valid voice commands and their corresponding wheelchair actions, such as forward, backward, turning left or right, stopping, or adjusting speed.
5. Wheelchair Control System Integration: Integrate the voice control logic with the wheelchair's control system. This typically involves connecting the voice control module to the existing control circuitry, motor drivers, or microcontroller of the wheelchair.
6. Motor Control and Movement: Implement the necessary motor control mechanisms to execute the wheelchair's movements based on the interpreted voice commands. This may involve controlling the motor speed, direction, and turning angle.
7. Error Handling and Feedback: Implement error handling mechanisms to handle cases where the voice command is not recognized correctly or is ambiguous. Provide appropriate feedback to the user, such as audible prompts or visual indicators, to indicate the status of the voice recognition and the wheelchair's response.
8. Safety Features: Ensure the voice-controlled wheelchair includes necessary safety features, such as emergency stop buttons, collision detection sensors, and anti-tip mechanisms, to prioritize the user's safety during operation.
9. Testing and Refinement: Conduct thorough testing and user trials to validate the voice control functionality, optimize the recognition accuracy, and refine the user experience. Continuously improve the system based on user feedback and real-world usage.
10. Documentation and User Guide: Prepare comprehensive documentation and user guides that explain the voice control functionality, provide instructions for setup and calibration, and explain the available voice commands and their corresponding actions.

**4.2- Project Code**

**#include <NewPing.h>**

**#include <Wire.h>**

**#include <LiquidCrystal\_I2C.h>**

**#include <Ultrasonic.h>**

**Ultrasonic ultrasonic(9, 10);**

**int distance;**

**LiquidCrystal\_I2C lcd(0x27,30, 0);**

**String readvoice;**

**int k=0;**

**void setup() {**

**Serial.begin(9600);**

**pinMode(2,OUTPUT);**

**pinMode(3,OUTPUT);**

**pinMode(4,OUTPUT);**

**pinMode(5,OUTPUT);**

**lcd.init();**

**lcd.backlight();**

**lcd.setCursor(1, 0);**

**lcd.print("Smart wheelchair");**

**// Set up the LCD screen**

**lcd.init();**

**lcd.begin(16, 2);**

**lcd.backlight();**

**// Print text on the first line**

**lcd.setCursor(0, 0);**

**lcd.print("Smart wheelchair");**

**// Print text on the second line**

**lcd.setCursor(0, 1);**

**lcd.print("Distance CM=");**

**}**

**void loop() {**

**while (Serial.available())**

**{**

**delay(3);**

**char c = Serial.read();**

**readvoice += c;**

**}**

**if(readvoice.length() > 0)**

**{**

**Serial.println(readvoice);**

**if(readvoice == "forward")**

**{**

**digitalWrite(2, LOW);**

**digitalWrite(3, HIGH);**

**digitalWrite(4, HIGH);**

**digitalWrite(5, LOW);**

**k=1;**

**}**

**if(readvoice == "backward")**

**{**

**digitalWrite(2, HIGH);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, HIGH);**

**k=2;**

**}**

**if(readvoice == "left")**

**{**

**if (k==2)**

**{**

**digitalWrite(2, LOW); //back**

**digitalWrite(3, HIGH);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**delay(1000);**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**}**

**else**

**{**

**digitalWrite(2, LOW);**

**digitalWrite(3, HIGH);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**delay(1000);**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**}**

**}**

**if(readvoice == "right")**

**{**

**if (k==2)**

**{**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, HIGH); //right forward**

**digitalWrite(5, LOW);**

**delay(1000);**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**}**

**else**

**{**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, HIGH);**

**digitalWrite(5, LOW);**

**delay(1000);**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**}**

**}**

**if(readvoice == "stop" )**

**{**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**}**

**if(distance == 30)**

**{**

**digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**}**

**}**

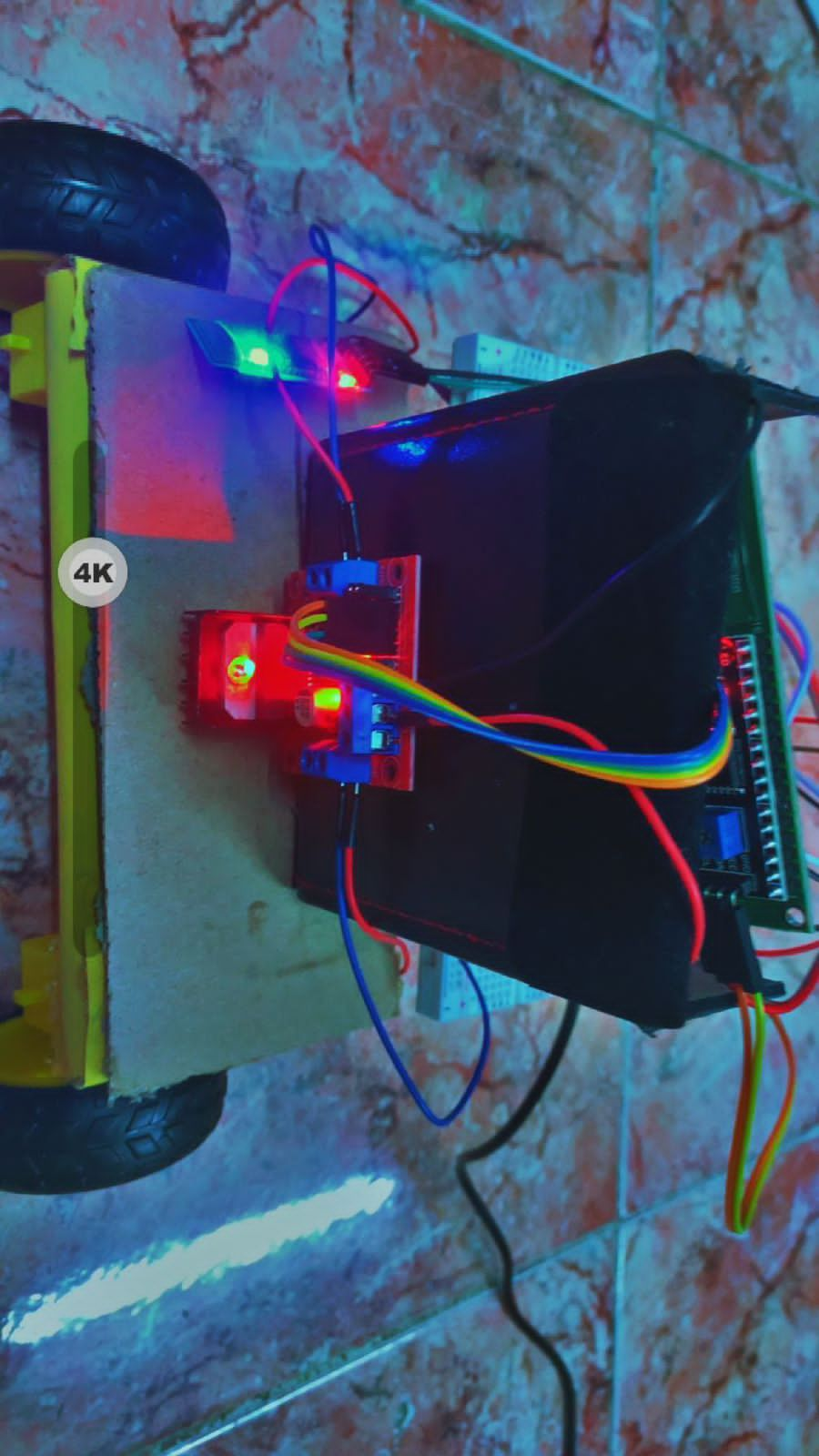
**readvoice="";**

**}**

**4.3-Testing**

In the testing phase, we tested the project in the experience of moving the chair forward, then to the right, then to the left, then standing by voice commands, and it was in different places. The experiment was empowering and successful.

First, we tested the chair with a forward-facing treadmill and it was great and we walked safely, then we ordered the voice commands to stop, then we tested walking to the right and then stopping and it was successful then we tested walking to the left then stopping and the experiment was successful then we tried to stop in front of the wall and it was successful



***Figure 22:*** *chair****.***

**4.3.1-** **The chair:**

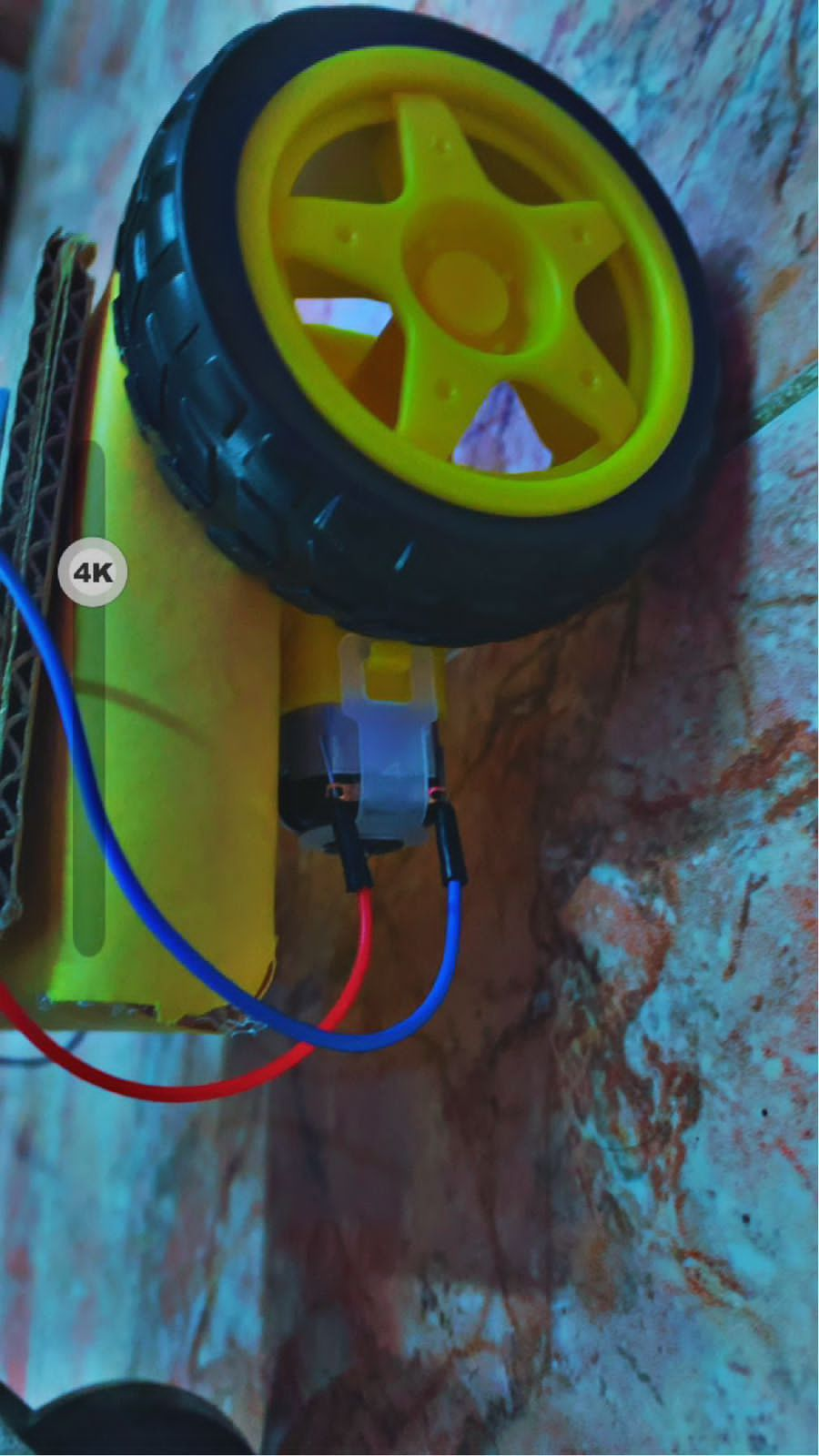


***Figure 23: wheelchair during operation***

**4.3.2-** **Real screen shots of the working project**



***Figure 24: Data display screen.***



***Figure 25: chair while stopping.***

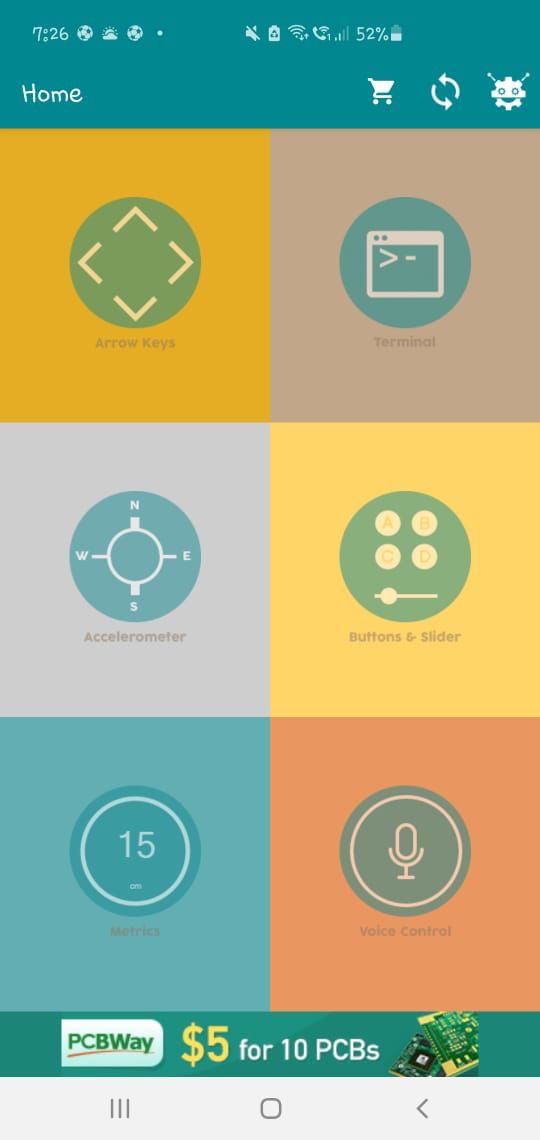
صورة تحتوي على الأسلاك الكهربائية, كابل, الهندسة الإلكترونية, الإلكترونيات

تم إنشاء الوصف تلقائياً

***Figure 26: installation.***

**4.4- Arduino BlueControl :**

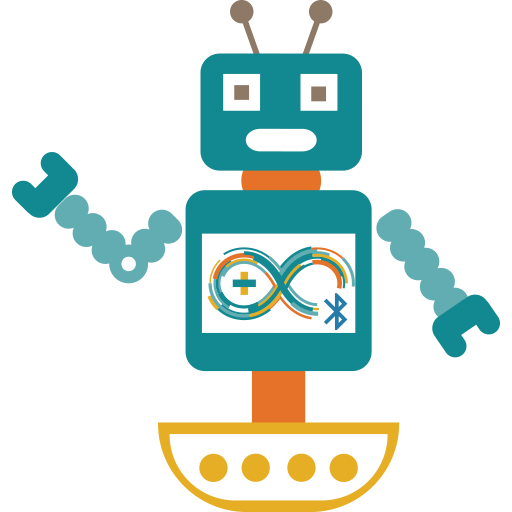
Arduino BlueControl is a program used to remotely control Arduino devices using Bluetooth technology. The program can be downloaded to Android smartphones, after which the phone can be used to control the Arduino devices connected by a Bluetooth module**.**



***Figure 27:*** Arduino BlueControl

**4.4.1 Application**

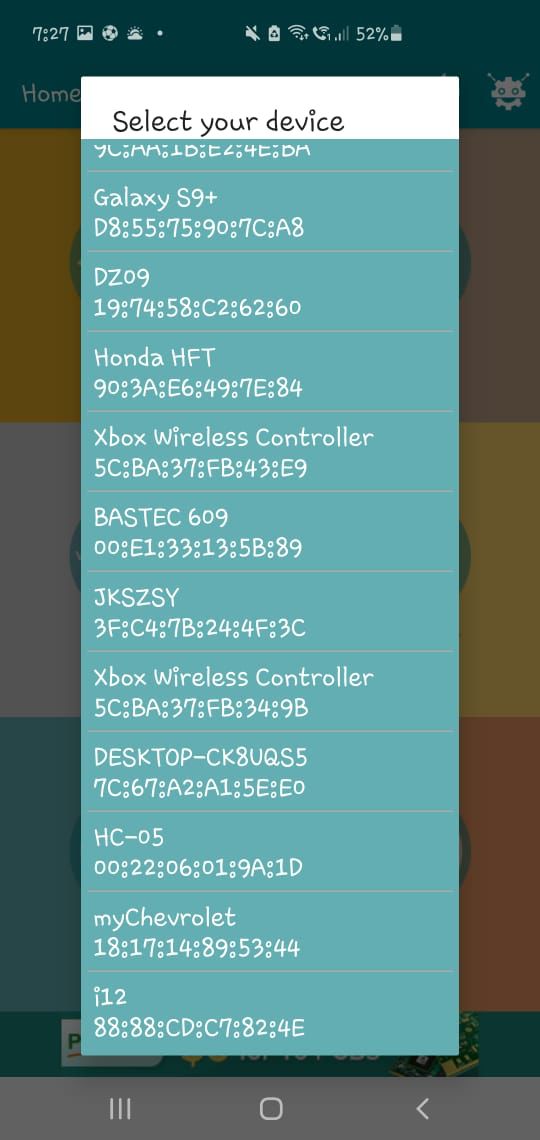
download and install. "Arduino bluecontrol" app on your smartphone and connect to "Smart. Wheelchair'!



***Figure 28: Application***

**4.4.2-Bluetooth**

Then we choose HC\_05 to connect to Bluetooth to control the use of the application



***Figure 29: HC\_05.***

**4.4.3- Voice control**

Voice control is a technology used in many devices and applications that enables users to control them using voice commands. Voice control relies on analyzing the sound and converting it into commands that the device can understand and execute.

Voice control can be used to send commands to move devices forward, backward, right, or left. To achieve this, modern voice control technologies such as voice analysis and voice command recognition are used.

صورة تحتوي على نص, لقطة شاشة, برمجيات, نظام التشغيل

تم إنشاء الوصف تلقائياً

***Figure 30: voice control stop. Figure 31: voice control backward.***

# 

# **Chapter 5**

CONCLUSION

**5.1 Summary**

The purpose of this project is to help people with special needs (people of determination) about the disabilities that led to their sitting and inability to walk or move freely; Using a wheelchair to provide a means of movement for people with mobility difficulties or physical disabilities. The model is used by voice commands to move to all four sides, right, left, forward, or backward. It is used via Bluetooth. Then it reads the data entered through the voice commands and then executes them. However, the warning message system indicates if there is a wall or an enclosed space or a wall and then changes direction. The results of this project aim to improve the lives of people with mobility disabilities and enable them to fully participate in daily life and achieve independence and personal freedom.

**5.2 Limitations and Future Work**

**5.2.1 Limitations**

The constraints we encountered during this project were significant and affected the overall scope and results of the project. Despite our best efforts, we faced challenges with time, design, and parts required for the job.

One of the main limitations was the lack of time. Our tight schedule made it difficult to fully test and improve our experiments, which affected the overall quality.

Although the smart wheelchair provides an important solution for people with mobility disabilities, it faces some limitations and challenges that prevent it from being widely used.

The cost of the smart wheelchair: as it requires advanced technology and high-quality materials, which makes it expensive, which hinders the ability of some to obtain it.

Size and weight: The smart wheelchair is heavier and bulkier than traditional chairs, which makes it difficult to move around in some cases, and it also requires more storage space.

Maintenance and operation: The smart wheelchair require maintenance, careful operation and great responsibility for the user, and may need special training to control and handle it.

Availability of infrastructure: The use of a smart wheelchair requires adequate infrastructure, such as sidewalks, corridors and wide doors, which is not available in all places.

In order to overcome these limitations and the challenges of the smart wheelchair, future work requires more research and development in this field, improving the design and technologies used, reducing costs, and providing the necessary support for users and those interested in it. It is expected that future work will contribute to further developing the smart wheelchair, improving its quality and providing it at more affordable prices, which will help improve the lives of people with mobility disabilities and enable them to fully participate in daily life.

**5.2.2 Future Work**

The future work of the smart wheelchair includes many aspects that can be improved and developed to achieve maximum benefit for users, and these aspects include:

Design improvement: The design of the smart wheelchair can be improved to make it more comfortable and easier to use, and its weight and size can be reduced to make it easier to carry and store. The design can also be improved to make it suitable for different needs and purposes.

Developing technologies: The technologies used in the smart wheelchair can be improved to make them more accurate and effective, and new technologies such as artificial intelligence and robotics can be developed to improve the performance of the smart wheelchair.

Expanding the scope of use: The scope of use of the smart wheelchair can be extended to different places and conditions and provides the necessary support for users to achieve freedom and independence of movement and movement.

Cost reduction: The cost of a smart wheelchair can be reduced by using less expensive materials, optimizing manufacturing and distribution processes, and providing financial support to users.

Raising awareness: It is possible to raise awareness of the benefits of a smart wheelchair and how to use it properly, and to provide the necessary training and education for users and those interested in it.

Market expansion: The smart wheelchair market can be expanded to different countries and regions, and the necessary support for marketing, distribution, maintenance and operation is provided.

Future work is expected to contribute to improving the quality of the smart wheelchair, expanding its use, achieving multiple benefits for people with mobility disabilities, improving their quality of life, and increasing their independence and freedom of movement.

Once these trials are completed, we intend to present the results of our work to the Ministry of Health. Not only will this allow us to share our findings with relevant stakeholders, but it will also help start the process of improving the quality of the smart wheelchair, expanding its use and preventing it from becoming bigger problems in the future. Through the use of artificial intelligence and extensive testing, we aim to make this project a leading model in the health field.

**REFERENCES**

|  |  |  |
| --- | --- | --- |
| **Software** |  | **Link** |
| **Arduino IDE** |  | <https://www.arduino.cc/en/software> |
| **SW-420** |  | [https://github.com/efduarte/pincello/blob/master/sensor-](https://github.com/efduarte/pincello/blob/master/sensor-vibration-sw-420.md) [vibration-sw-420.md](https://github.com/efduarte/pincello/blob/master/sensor-vibration-sw-420.md) |
| **MPU-6050** |  | [https://randomnerdtutorials.com/esp32-mpu-6050-](https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/) [accelerometer-gyroscope-arduino/](https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/) |
| **gy-gps6mv2** |  | <https://github.com/ahmadlogs/esp32-gps-webserver> |
| **jsn-sr04t** |  | [https://www.makerguides.com/jsn-sr04t-arduino-](https://www.makerguides.com/jsn-sr04t-arduino-tutorial/) [tutorial/](https://www.makerguides.com/jsn-sr04t-arduino-tutorial/) |
| **Connect with webserver** |  | <https://github.com/ahmadlogs/esp32-gps-webserver> |
| **Awardspace** |  | <https://www.awardspace.com/> |

***Table 3: REFERENCES.***