Smart Wheelchair Integrating Bluetooth and Health Monitoring System for Enhanced Mobility with Arduino Due

Neeraj Hampal
Electrical and Computer
Concordia University
n_hampal@live.concordia.ca

Raghav Mohindru
Electrical and Computer
Concordia University
r_mohind@live.concordia.ca

Pardheep Shankar
Electrical and Computer
Concordia University
p_shank@live.concordia.ca

Abstract— **This report is based on a project within the course COEN 6711 in Fall 2023. In a world rapidly advancing with smart technologies, there's an urgent need to enhance mobility for those with physical challenges, including the elderly. Recognizing this, we've unveiled an innovative motorized robot equipped with sophisticated wireless Bluetooth connectivity operated controls. Beyond just serving as a smart wheelchair solution, its applications extend to automated parking systems for vehicles and in-home transport. This state-of-the-art device is a boon for users, granting them freedom of movement without excessive physical effort. It's especially tailored for those visually or physically impaired, ensuring they navigate seamlessly. Its obstacle detection system prioritizes user safety by preventing collisions. Our mission is clear: deliver a comprehensive, yet budget-friendly robotic solution that champions autonomy and decreases the need for external support like caregivers. The authors acknowledge and declare hereby that this report is entirely based on their own work and that it has been written solely by the group members not having used any sources or support other than those acknowledged within the document. This report has not been submitted for any other academic award or evaluation.**

Keywords— Smart Technologies, Mobility, Physical Challenges, Elderly, Innovative, Motorized Robot, Wireless Bluetooth Connectivity, Automated Parking Systems, In-Home Transport, State-of-the-Art, Freedom of Movement, Visually Impaired, Physically Impaired, Obstacle Detection System, User Safety, Collision Prevention, Comprehensive Solution, Budget Friendly, Autonomy, External Support, Caregivers, Robotic Solution.

I. Introduction

In the years advancements, in technology have brought about significant changes in various aspects of our daily lives. However, when it comes to mobility aids like wheelchairs there has been innovation. This report focuses on a project called "Smart Wheelchair Integrating Bluetooth and Health Monitoring System " which combines accessibility, health monitoring and modern technology. The motivation behind this project is to improve the quality of life for individuals with mobility impairments by not providing them with mobility but also offering independence and health monitoring capabilities. At the heart of our wheelchair system lies the Arduino Due, a microcontroller board that coordinates the functions of sensors and communication modules. The wheelchair is equipped with sensors such as the HC SR04 sensor for detecting obstacles the DHT11 sensor for monitoring thermal stats of the patient and the MAX30100 pulse oximeter, for tracking vital health parameters. By integrating these sensors, we ensure a responsive user experience. Additionally, we have incorporated the HC 05 Bluetooth module to introduce connectivity enabling control and data transfer features. This project was conceived not as a response to address limitations in wheelchairs but also as an example of how embedded systems can create more adaptable and user centred mobility aids. The inclusion of health monitoring tools, into the wheelchair framework has the goal of meeting the needs of users who have additional health considerations. By integrating real-time health monitoring and automated navigation capabilities the wheelchair goes beyond its purpose. Becomes a versatile support system that fosters independence and well-being.

In the sections of this report, we will delve into the process of designing, addressing challenges and solutions presenting results from simulations and demonstrations as well as sharing our reflections on the outcomes of this project. Through this report, our aim is to provide an overview of our journey in developing a wheelchair that does not facilitate mobility but also represents progress, in merging health and technology within accessibility aids.

II. Literature Review

A. *Background and Related Work*

The concept of smart wheelchairs, integrating advanced technologies such as Bluetooth connectivity and health monitoring systems, represents a significant leap in assistive technologies, aimed at enhancing mobility and healthcare for individuals with disabilities.

Earlier research predominantly focused on traditional manual and electric wheelchairs. However, recent advancements have shifted towards the development of 'smart' wheelchairs that integrate various technologies to aid users with mobility challenges. These wheelchairs incorporate systems for gesture control, health monitoring, autonomous navigation, and environmental interaction. Studies like Mega lingam et al. explored Bluetooth-based wheelchair systems controlled by hand or finger gestures using an Android application [1]. This innovation provides a less physically demanding method of control, particularly beneficial for users with limited hand mobility. The implementation of gesture control signifies a user-centric approach to wheelchair design, focusing on ease of use and adaptability to different user needs. The integration of health monitoring systems in wheelchairs is a relatively new yet critical area of research. Papers like Sudha et al. have focused on designing wheelchairs that not only aid in mobility but also monitor the user's health through embedded sensors [2]. These systems track vital signs, providing valuable data for healthcare monitoring and emergency response. The concept of autonomous driving, as explored by Kim et al., has been adapted to smart wheelchairs, particularly in complex environments like hospitals. These wheelchairs use ultrasonic sensors and Wi-Fi/Bluetooth modules to navigate, avoid obstacles, and enhance patient mobility [3]. This innovation reflects a blend of robotic technology and healthcare, aiming to offer independent mobility solutions in structured environments like smart hospitals. The use of Arduino Due in these projects is particularly notable. Arduino platforms are revered for their flexibility, accessibility, and ease of integration with various sensors and modules. The application of Arduino Due in smart wheelchairs underscores its suitability for developing cost-effective, adaptable, and user-friendly assistive devices. This background and related work demonstrate a significant shift in wheelchair technology, from basic mobility aids to sophisticated systems integrating gesture control, health monitoring, and autonomous navigation. These innovations represent a convergence of healthcare, technology, and user-centered design, significantly enhancing the quality of life for individuals with mobility challenges. The use of platforms like Arduino Due further emphasizes the trend towards accessible and adaptable technology in the field of assistive devices.

B. Overview of microprocessors

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note section A-D below for more information on proofreading, spelling, and grammar.

Arduino Uno and Mega are capable enough of handling the project but usage of Due makes it a better choice due to power efficiency, more memory, and CPU capabilities making it an ideal choice to pick for real time simulations.
The three scholarly articles selected the utilize different microprocessors and microcontroller platforms, each contributing unique functionalities essential to the development of a smart wheelchair system.

“Autonomous Driving Robot Wheelchairs for Smart Hospitals” paper introduces a robot wheelchair with autonomous driving capabilities designed for use in smart hospitals. While the specific microprocessor or microcontroller used isn't explicitly mentioned, the wheelchair includes multiple ultrasonic sensors and Wi-Fi/Bluetooth modules for navigation and communication. The autonomous functionality implies the use of an advanced microcontroller or microprocessor capable of processing sensor data, controlling motors, and managing wireless communication. It is likely that a high-performance microcontroller or embedded system is used to handle these complex tasks [3].

In the research paper “Wireless Gesture-Controlled Wheelchair”, an Arduino Uno board is used as the primary microcontroller. Arduino Uno is a popular choice for prototyping due to its ease of programming, availability of inbuilt RX and TX pins for communication, and user-friendly interface. This board facilitates the interaction between the microcontroller and Bluetooth modules, making it an ideal choice for developing a gesture-controlled wheelchair system. The Arduino Uno controls the wheelchair's motors in response to signals sent from an Android application through Bluetooth, demonstrating its capability to handle wireless communication and motor control [1].

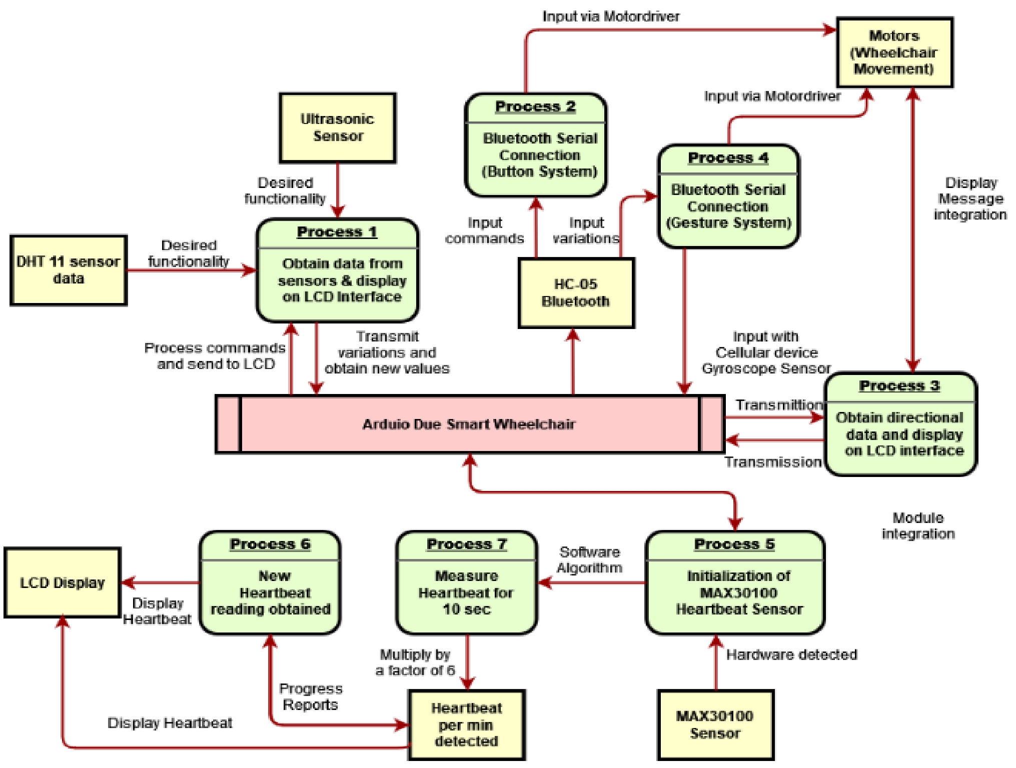
"A Smart Wheelchair for Aged People with Health Monitoring System" paper focuses on a smart wheelchair design that incorporates health monitoring features. It uses an Arduino Uno alongside a Raspberry Pi for enhanced processing and connectivity capabilities. The Arduino Uno is employed for its simplicity and effectiveness in sensor data processing and motor control, while the Raspberry Pi offers more advanced computing power for tasks like live broadcasting and cloud connectivity. The combination of these two platforms allows for comprehensive health data analysis and storage, as well as remote monitoring capabilities, making them well-suited for a health-monitoring wheelchair system [2].

In summary, these papers utilize a range of microcontrollers and processors, with Arduino Uno being a common choice due to its accessibility and ease of use. The inclusion of Raspberry Pi in one of the papers indicates a blend of simple control mechanisms with more complex data processing and communication tasks, highlighting the diverse technological requirements of advanced smart wheelchair systems.

The integration of microprocessors and microcontrollers, particularly the Arduino Due, plays a pivotal role in the development of the Smart Wheelchair project, which integrates Bluetooth and a Health Monitoring System. This overview examines how these components are crucial in achieving the project's objectives of enhanced mobility and health monitoring. Microprocessors and microcontrollers are the brains behind modern assistive technologies, including smart wheelchairs. They process inputs from various sensors, execute control algorithms, and manage output devices like motors and displays. In this project, the Arduino Due stands out for its powerful ARM Cortex-M3 microcontroller, offering significant processing capabilities, ample memory, and a range of input/output options. This makes it ideal for handling complex tasks such as processing sensor data, managing Bluetooth communication, and executing control commands for wheelchair movement. The Arduino Due manages Bluetooth modules to enable wireless communication. This allows for the implementation of remote-control features, where users can operate the wheelchair using a smartphone or other Bluetooth-enabled devices, enhancing the user's independence. A crucial aspect of the smart wheelchair is its ability to monitor the user's health. The Arduino Due collects data from various health sensors, such as heart rate monitors or temperature sensors. It processes this data in real-time, providing vital health insights and potentially alerting caregivers in case of emergencies. Microcontrollers process data from a range of sensors – ultrasonic for obstacle detection, accelerometers for stability, and gyros for directional data. This sensor integration is essential for the autonomous navigation capabilities of the wheelchair, ensuring safe and efficient movement. The Arduino Due also manages the user interface elements of the wheelchair. This includes displaying health data, battery status, and system alerts, as well as receiving input commands from the user or remote caregiver. The flexibility of the Arduino platform allows for customizable software, meaning that the wheelchair's behavior can be tailored to the specific needs and preferences of the user. Microcontrollers like the Arduino Due are designed to be energy-efficient, which is crucial for battery-powered devices like wheelchairs. Efficient processing ensures longer battery life, making the wheelchair more reliable and practical for everyday use. The ability of Arduino Due to process complex data, manage multiple inputs and outputs, and offer customizable software solutions makes them indispensable in the development of smart wheelchairs designed for enhanced mobility and health monitoring.

III. Model Description

The designed model for the "Smart Wheelchair Integrating Bluetooth and Health Monitoring System for Enhanced Mobility with Arduino Due" is a sophisticated, multi-functional assistive device aimed at enhancing the mobility and health monitoring capabilities for individuals with mobility impairments. This smart wheelchair combines cutting-edge technology with user-friendly features to create a comprehensive mobility solution. Hardware information is presented in this unit and Code explanation would be presented in Appendix (refer Appendix of report).



Input via Motordriver
Motors (Wheelchair Movement)
Process 2
Bluetooth Serial Connection (Button System)
Input via Motordriver
Ultrasonic Sensor
Process 4
Bluetooth Serial Connection (Gesture System)
Display Message integration
Desired functionality
Input commands
Input variations
Desired functionality
Process 1
Obtain data from sensors & display on LCD Interface
DHT 11 sensor data
HC-05 Bluetooth
Transmit variations and obtain new values
Process commands and send to LCD
Input with Cellular device Gyroscope Sensor
Process 3
Transmittion
Obtain directional data and display on LCD interface
Arduio Due Smart Wheelchair
Transmission
Module integration
Process 6
New Heartbeat reading obtained
Process 7
Measure Heartbeat for 10 sec
Software Algorithm
Process 5
Initialization of MAX30100 Heartbeat Sensor
LCD Display
Display Heartbeat
Multiply by a factor of 6
Hardware detected
Progress Reports
Heartbeat per min detected
MAX30100 Sensor
Display Heartbeat

Figure 1: Design Model Flowchart

A. *Arduino Due*

The Arduino Due is a powerful microcontroller board based on the 32-bit Atmel SAM3X8E ARM Cortex-M3 CPU, a significant step up from the more common 8-bit microcontrollers in other Arduino models. As the first ARM-based Arduino board, it offers increased processing power, making it ideal for projects that require handling more complex tasks, such as managing multiple sensors and controlling motor drivers simultaneously. What sets the Due apart is its extensive set of I/O interfaces. It boasts 54 digital input/output pins, 12 analog inputs, 4 UARTs (hardware serial ports), an 84 MHz clock speed, and USB OTG capabilities. The board's versatility makes it highly suitable for robotics and embedded system projects, where multifunctionality and higher computational needs are paramount.

The Due's 3.3V operating voltage is critical to note, as it differs from the 5V commonly used in other Arduino boards. This requires careful consideration when interfacing with various sensors and modules to avoid damaging the board. With features like DMA (Direct Memory Access), which allows for faster communication without CPU involvement, and DACs (Digital-to-Analog Converters), the Due is well-suited for sophisticated tasks like controlling a smart wheelchair.

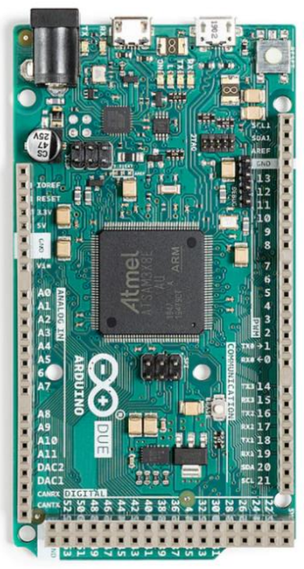


Figure 2: Arduino Due microcontroller used to develop model.

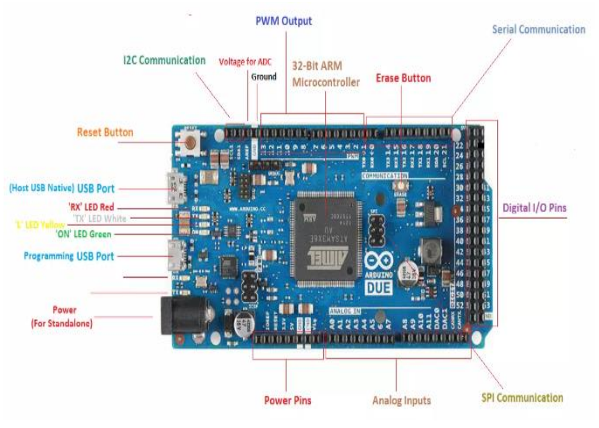


Figure 3: Arduino Due input and output ports details

B. *Ultrasonic Sensor*

The HC-SR04 Ultrasonic Sensor is a popular choice in robotics for distance measurement and object detection. It operates by emitting ultrasonic sound waves at a frequency of 40 kHz, which travel through the air and bounce back upon hitting an object. By measuring the time taken for the echo to return, the sensor calculates the distance to the nearest object. This sensor finds widespread use, where ensuring the user's safety by avoiding collisions is crucial. The HC-SR04's ability to provide non-contact distance measurements between 2cm and 400cm with a modest accuracy makes it suitable for this project.

TABLE 1: PIN CONFIGURATION OF ULTRASONIC SENSOR

Pin Number	Pin Name	Description
1	V_{cc}	The V_{cc} pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin must be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period which will be equal to the time taken for the US wave to return to the sensor.
4	Ground	This pin is connected to the Ground of the system.

The simplicity of the HC-SR04, combined with its low cost and ease of use, makes it an excellent choice for projects that require basic distance sensing capabilities. However, the sensor's effectiveness can vary based on environmental factors like temperature and humidity, as well as the reflectivity and size of the object being detected.

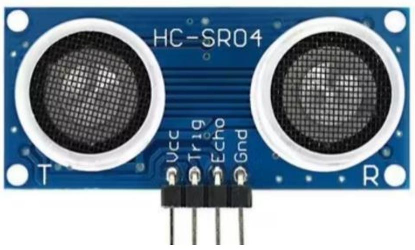


Figure 4: Ultrasonic Sensor (HC-SR04)

C. *Temperature Sensor*

The DHT-11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and outputs a digital signal on the data pin. This sensor is relatively simple to use but requires careful timing to grab data. The DHT-11 stands out for its ability to operate in a wide range of humidity (20-80%) and temperature (-40 to 80°C), albeit with lower accuracy compared to more sophisticated sensors. The sensor updates data every 2 seconds, making it suitable for applications where real-time changes in temperature and humidity are not critical. Pin configuration has Vcc(typically 3.3v), GND and the data pin to transmit data to the Arduino.

In the context of a smart wheelchair, the DHT-11 can provide valuable data regarding the environmental conditions surrounding the user. This information can be crucial for maintaining comfort and monitoring health, especially in temperature-sensitive conditions or for users with specific medical requirements.

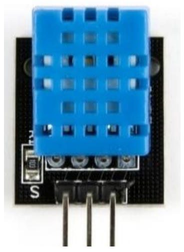


Figure 5: DHT-11 temperature sensor

D. *Heartbeat Sensor*

The MAX30100 is an integrated pulse oximetry and heart rate monitor sensor. It combines a red LED, an infrared LED, and a photodetector to measure the absorption of light in the user's bloodstream. By using these measurements, it can calculate the heart rate and blood oxygen saturation levels (SpO2). The sophistication of the MAX30100 lies in its ability to perform these measurements through the skin, making it non-intrusive and convenient for continuous monitoring. It's particularly valuable in health monitoring applications, such as a smart wheelchair, where keeping track of the user's cardiovascular health is essential. The ability to monitor heart rate and oxygen saturation on the go can provide critical data for users with heart conditions or respiratory issues. The MAX30100, however, requires careful calibration and algorithmic processing to provide accurate readings, as factors like movement and ambient light can affect its measurements. Its integration into a smart wheelchair system would demand robust software to process and interpret the sensor data reliably.

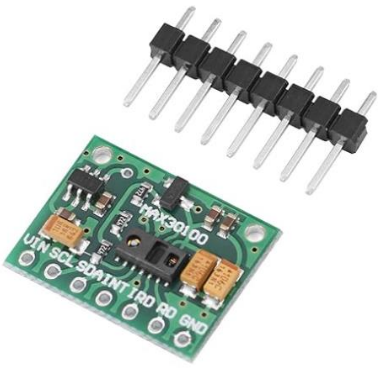


Figure 6: Heartbeat sensor used in the system.

E. Bluetooth Module

The HC-05 Bluetooth Module is a widely used solution for wireless communication in embedded systems. It operates on Bluetooth 2.0 technology and can function both as a master and a slave device, making it versatile for various applications. The module can be used to enable a wireless bridge between the microcontroller (like the Arduino Due) and a smartphone, computer, or other Bluetooth-enabled devices. In a smart wheelchair application, the HC-05 can facilitate real-time data transmission and control commands between the wheelchair and a handheld device. This functionality can enhance the user experience by allowing remote monitoring and control of the wheelchair, such as adjusting speed settings, monitoring battery levels, or even receiving health data directly on a user's smartphone. The HC-05 module's ease of use and its ability to operate over a considerable range (typically around 10 meters in open space) make it an excellent choice for short-range wireless communication in the smart wheelchair project.

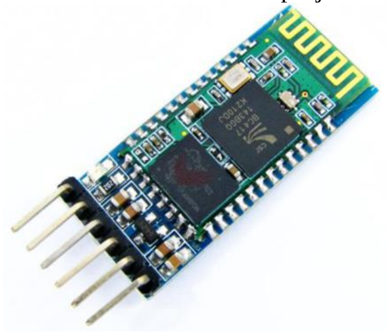


Figure 7: Bluetooth module used in the developed system

F. Motor Driver Sensor

The L293D is a dual H-bridge motor driver IC that allows DC motors to be driven in both directions. It's capable of driving two DC motors simultaneously or one stepper motor. The L293D can handle a supply voltage of up to 36V and a peak current of up to 600mA per channel. In the context of the smart wheelchair, the L293D is critical for controlling the wheelchair's motors. It allows the microcontroller to control the speed and direction of the motors, enabling maneuvers such as turning and reversing.
The pin configuration of the L293D IC is shown in the fig 8.

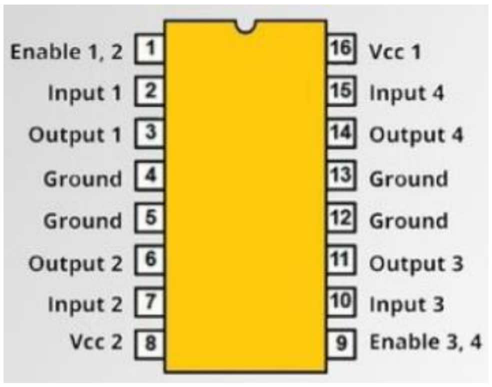


Figure 8: Pin Configuration of L293D IC

The IC features internal diodes for back EMF protection, an essential aspect when dealing with inductive loads like DC motors. The L293D's ability to manage high voltage and current makes it suitable for motor control applications where reliability and durability are paramount. Its integration into the wheelchair design would require careful planning to ensure that the motor specifications match the driver's capabilities and that adequate power supply arrangements are made.

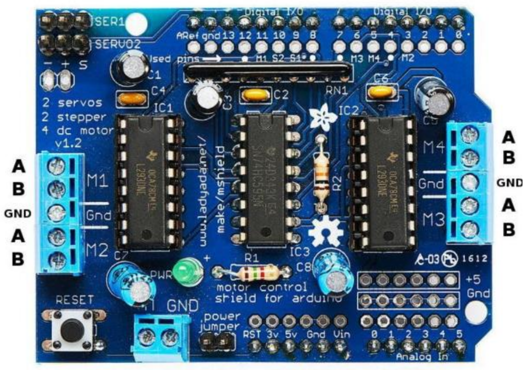


Figure 9: Motor driver sensor used in the system.

G. LCD LM016L

The LM016L is a standard 16x2 LCD display module used for displaying text and simple graphics. This type of display consists of 16 columns and 2 rows of characters, each character being formed by a matrix of 5x8 pixels. The LM016L operates on 5V and communicates with the microcontroller via a parallel interface. In a smart wheelchair system, the LM016L can serve as a user interface to display important information such as speed, battery level, temperature, heartbeat rate, and other health-related data. It provides a simple yet effective way for users to interact with and receive feedback from the wheelchair. The display's low power consumption and ease of interfacing with microcontrollers like the Arduino Due make it an attractive option for embedded systems. The LCD can be operated in either 8-bit or 4-bit mode, with the latter requiring fewer I/O pins from the microcontroller, a valuable consideration in complex projects like a smart wheelchair where I/O availability might be limited.

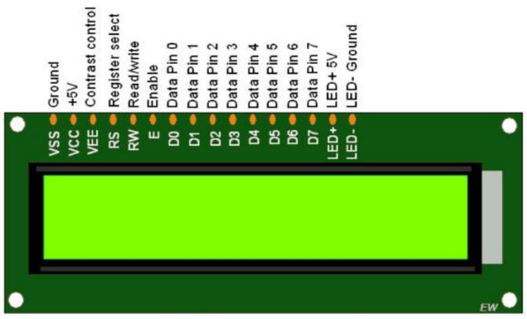


Figure 10: LM016L LCD display

IV. Experimental Architecture

This smart wheelchair combines cutting-edge technology with user-friendly features to create a comprehensive mobility solution.

A. *Hardware Architecture*

1. Arduino Due: Acts as the central processing unit of the wheelchair. It controls and coordinates all other components. Arduino Due is chosen for its higher processing power and multiple input/output ports, which are suitable for handling various sensors and communication modules.
2. Ultrasonic Sensors: Used for obstacle detection and avoidance. These sensors help in navigating the wheelchair safely by measuring the distance to obstacles in its path and sending this data to the Arduino Due for processing.
3. DHT11 Sensor: A basic, ultra-low-cost digital temperature and humidity sensor. It measures the ambient temperature and humidity, ensuring the comfort of the wheelchair user. The data collected is sent to the Arduino Due.
4. MAX30100 Sensor: A combined pulse oximeter and heart rate sensor. It measures the oxygen saturation (SpO2) levels and heart rate of the user, which is crucial for monitoring their health. The readings are sent to the Arduino Due.
5. HC-05 Bluetooth Module: Enables wireless communication. It can be used to pair the wheelchair with a smartphone or other Bluetooth-enabled device, allowing for remote control or data transmission.
6. L293D Motor Driver: Used to control the wheelchair's motors. The Arduino Due sends signals to the L293D to manage the speed and direction of the motors based on input from sensors or Bluetooth commands.
7. LCD LM016L: A display unit to show important information such as temperature, humidity, SpO2 levels, heart rate, and obstacle detection alerts. It is connected to the Arduino Due and displays data processed by the microcontroller.
8. Power supply: This project is aided with a 12v(1.3Ah) power supply and additionally added a power board which uses DC-DC converters to step down the input 12v supply to varied powers such as 9v, 5v and 3.3v along with GND pins to ground the electronics.

B. *Software Architecture*

The software architecture involves programming the Arduino Due to integrate all hardware components and ensure they work seamlessly together. Key aspects include:

1. Sensor Data Acquisition and Processing: The software continuously reads data from the ultrasonic, DHT11, and MAX30100 sensors. Algorithms are used to process this data, such as calculating the distance to obstacles, interpreting temperature and humidity readings, and analyzing heart rate and SpO2 levels.
2. Motor Control Logic: The software includes algorithms to control the wheelchair's motors through the L293D motor driver. It incorporates safety features, like stopping or changing direction if an obstacle is detected.
3. Bluetooth Communication Handling: The software manages Bluetooth communications via the HC-05 module. It can receive commands from a paired device to control the wheelchair or send health data to the device.
4. User Interface Management: The software controls what is displayed on the LCD screen, ensuring that the user receives real-time updates about their environment, health status, and wheelchair status.
5. Emergency Response System: Integration of an emergency alert system in the software that activates if critical health parameters are detected or in case of a malfunction.

V. Results

A. *Software Implementation*

We implemented the simulation of the proposed system using the Proteus Software. We implemented different scenarios of the project on the simulation as well as the hardware. There are various scenarios implemented for which we have included the respective screenshots and the behavior observed for the system. For the simulation we have used the Arduino Mega 250 board as the libraries for Arduino Due is not available for the Proteus Software. The connections of the various sensors such as the heartrate sensor are shown in the figures below.

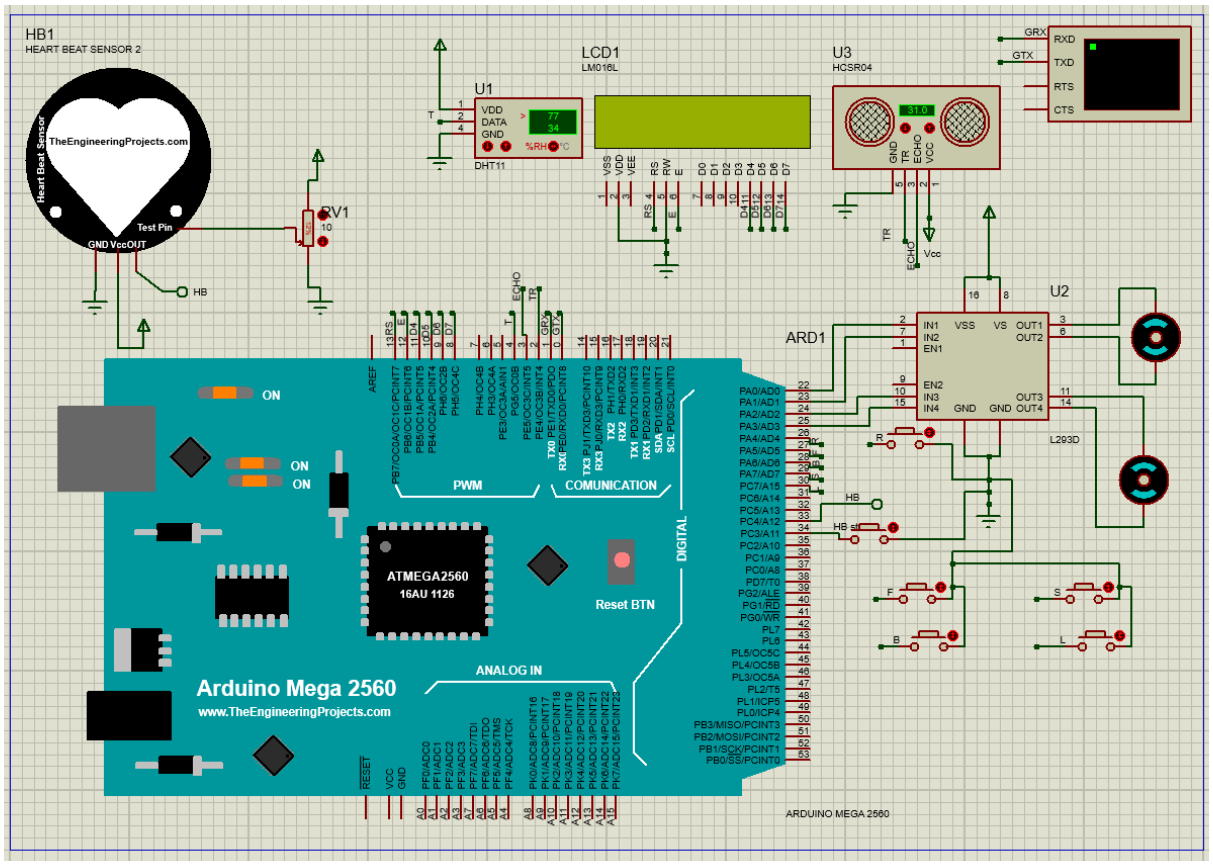


Figure 11: Proteus design of all functionalities.

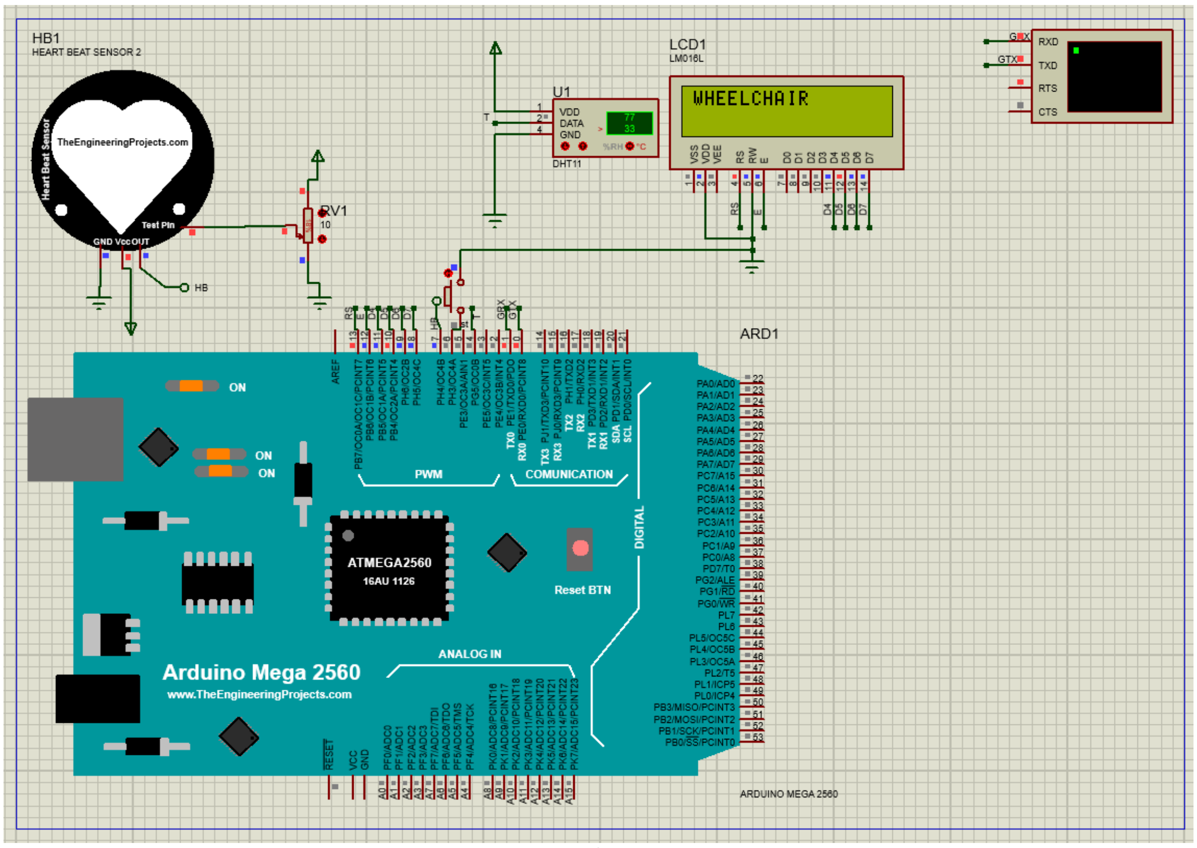


Figure 12: Proteus simulation of the healthcare system designed.

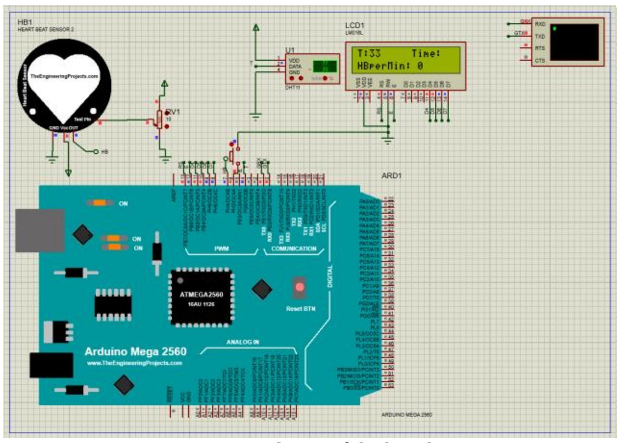


Figure 13: Simulation of the heartbeat sensor on the Proteus (sensor is off).

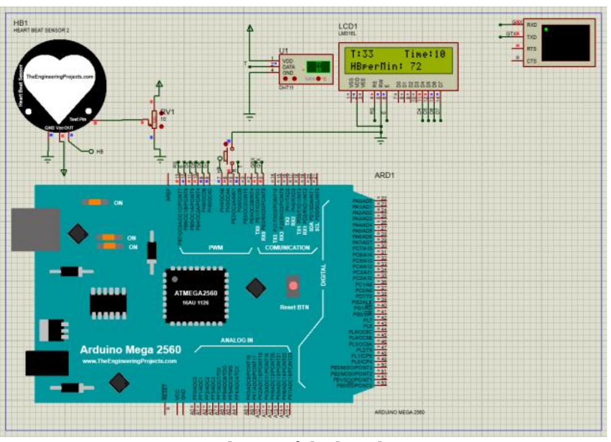


Figure 14: Simulation of the heartbeat sensor on the Proteus (sensor is on).

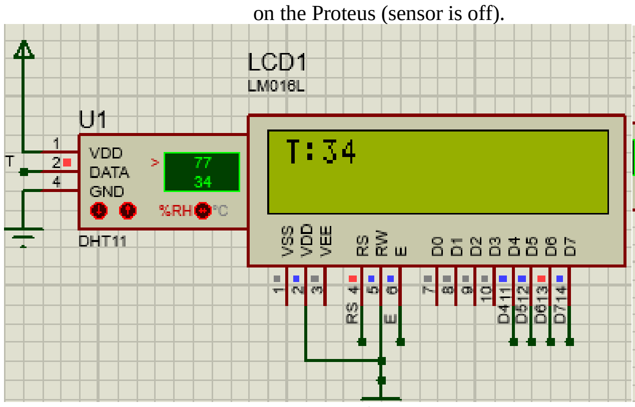


Figure 15: Simulation of the DHT11 sensor on the Proteus simulation

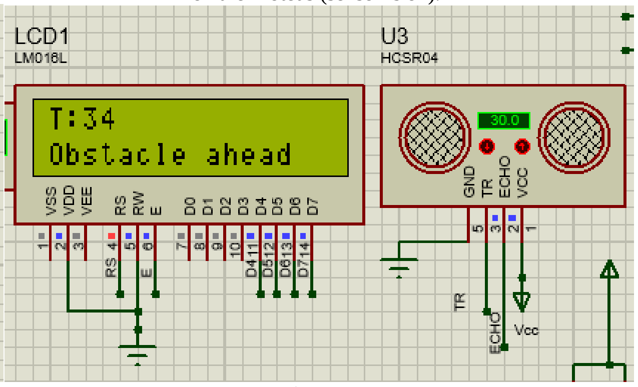


Figure 16: Simulation of the ultrasonic sensor on the Proteus simulation.

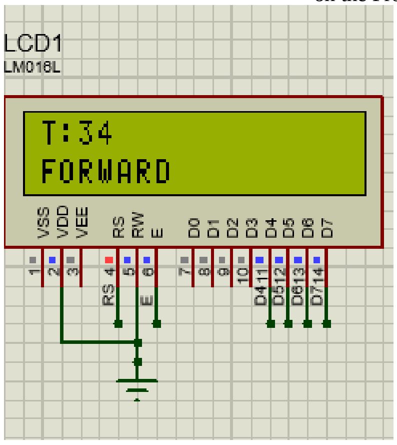


Figure 17: Simulation of Forward Direction

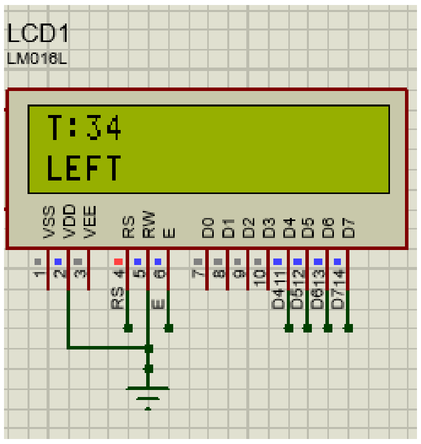


Figure 18: Simulation of Left Direction

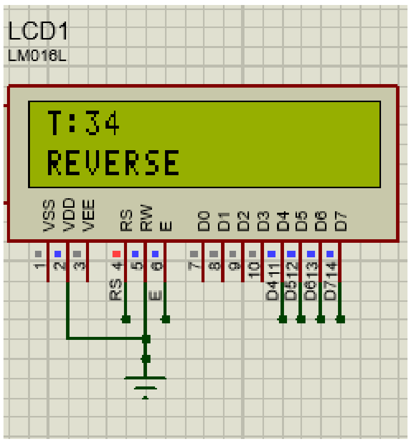


Figure 19: Simulation of the Reverse Direction

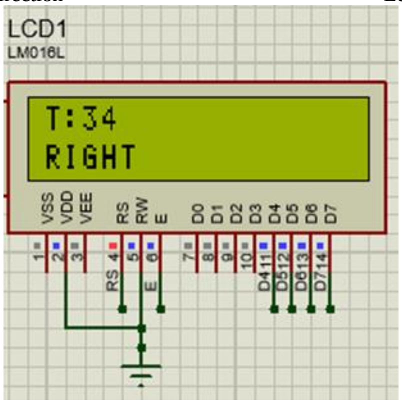


Figure 20: Simulation of the Right Direction

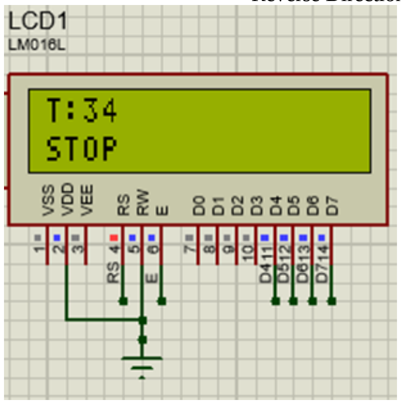


Figure 21: Simulation of the Stop command.

B. Hardware Implementation

For implementing the hardware, we have used the Arduino due board mounted on the wheelchair system utilizing the data collected from the various sensors such as heart rate and temperature sensor. Below in figures 19, 20 and 21 we have shown the hardware implemented for the proposed system.

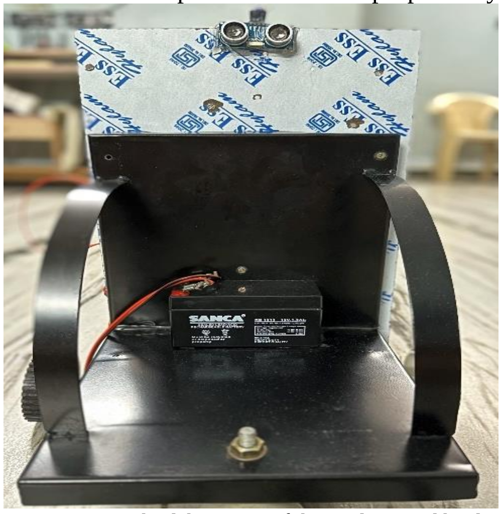


Figure 19: Front wheelchair view of the implemented hardware

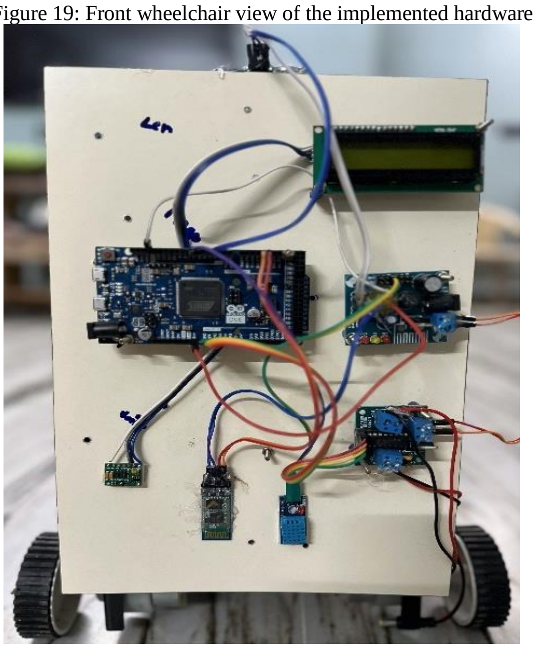


Figure 20: Back wheelchair view of the implemented hardware

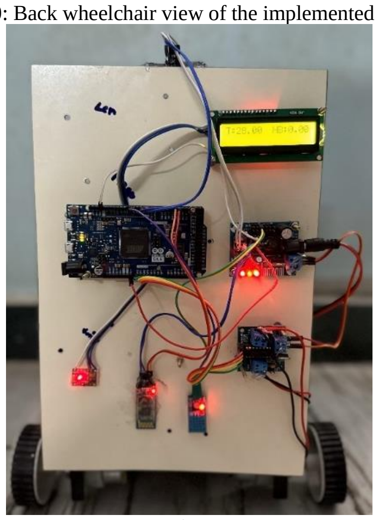


Figure 21: Working model of the hardware implementation.

VI. Future Scope

The Smart Wheelchair project integrating Bluetooth and health monitoring systems with Arduino Due presents numerous avenues for future development and enhancement. Here are some potential areas of future scope:

1. Artificial Intelligence (AI) and Machine Learning (ML) Integration: Implementing AI algorithms can significantly improve the wheelchair's navigation and obstacle avoidance capabilities. Machine learning can be used to learn from the user's behaviour and environment, offering more personalized and adaptive controls.

2. Internet of Things (IoT) Connectivity: Integrating the wheelchair with IoT platforms can enable real-time data monitoring and analysis. This can be used for remote health monitoring, where caregivers or medical professionals can access the user's health data remotely.

3. Advanced Health Monitoring Features: Incorporating more sophisticated health sensors, such as ECG monitors or blood pressure sensors, can provide a more comprehensive health overview of the user.

4. Voice Control Integration: Adding voice recognition technology can enhance user experience, especially for individuals with limited mobility or dexterity. Users could control the wheelchair and access its features using voice commands.

5. GPS and Navigation Systems: Integrating GPS for outdoor navigation can assist users in reaching their destinations more efficiently. This could be coupled with a mobile app to plan routes or receive directions.

6. Fall Detection and Emergency Response: Implementing a fall detection system using accelerometers and gyroscopes can alert caregivers or emergency services in the event of a fall or accident.

7. Autonomous Navigation Capabilities: Developing more advanced autonomous navigation capabilities using technologies like LiDAR or computer vision can enable the wheelchair to navigate more complex environments independently.

8. Battery Life Optimization and Solar Charging: Improving battery technology and integrating solar panels for charging can extend the wheelchair's operational time and reduce the need for frequent recharging.

9. User Interface (UI) Improvements: Enhancing the wheelchair's UI with touch screens or augmented reality (AR) displays can provide more intuitive and informative controls for the user.

10. Customization and Personalization: Offering customization options in software and hardware for individual user needs, such as adjustable seating,

11. personalized control settings, or modular design for easy upgrades.

12. Telepresence and Social Interaction Features: Incorporating features like a camera and screen can facilitate telepresence, allowing users to interact with others remotely, enhancing social engagement.
13. Integration with Smart Home Devices: Connecting the wheelchair with smart home systems can allow users to control home appliances and systems directly from the wheelchair.

By exploring these areas, the Smart Wheelchair project can evolve into a more advanced, user-friendly, and integrated mobility solution, enhancing the quality of life for its users.

VII. Limitations

While the Smart Wheelchair integrating Bluetooth and health monitoring systems with Arduino Due is an innovative project, it has several limitations that could impact its effectiveness and user experience. Here are some of the key limitations:

1. Hardware Constraints: The Arduino Due, while powerful for many applications, has limitations in processing speed and memory capacity compared to more advanced microcontrollers. This might restrict the implementation of more complex algorithms or the integration of additional sensors.

2. Battery Life and Power Management: The addition of multiple sensors, Bluetooth connectivity, and health monitoring tools can significantly drain the wheelchair's battery. Efficient power management and long battery life are crucial for the practical use of the wheelchair, especially for extended periods.

3. Bluetooth Connectivity Limitations: Bluetooth technology, although effective for short-range communication, has limitations in terms of range and be susceptible to interference. This could affect the reliability of the remote control and data transmission.

4. Sensor Accuracy and Reliability: The precision and reliability of sensors like the DHT11 and MAX30100 are critical. However, these sensors may have limitations in terms of accuracy and responsiveness, which could impact the health monitoring and navigation functionalities.

5. Size and Weight of the System: The integration of multiple components and systems could increase the size and weight of the wheelchair, potentially impacting its maneuverability and portability.

6. Complex User Interface: The system may have a relatively complex user interface, which could be challenging for some users, particularly the elderly or those with cognitive impairments.

7. Cost Implications: Integrating advanced technologies and multiple sensors can significantly increase the cost of the wheelchair, making it less accessible to a broader user base.

8. Environmental Limitations: The wheelchair’s performance might be affected by environmental factors like weather conditions, as some sensors may not function optimally in extreme temperatures or wet conditions.

9. Maintenance and Durability: The complexity of the system could lead to higher maintenance needs. Ensuring durability and easy maintenance is crucial for the long-term usability of the wheelchair.

10. Data Security and Privacy: With the integration of Bluetooth and health monitoring systems, ensuring the security and privacy of the user's health data is crucial. There is a risk of data breaches or unauthorized access.

11. Regulatory Compliance: Meeting health and safety standards and regulatory compliance for medical devices can be challenging, especially when integrating health monitoring features.

12. Software Stability and Updates: The software must be stable and reliable. Regular updates and bug fixes are necessary to ensure the system remains functional and up to date.

Addressing these limitations would be essential in further developing the Smart Wheelchair project to enhance its practicality, user-friendliness, and accessibility.

VIII. Comparative Analysis

The table below compares the unique aspects of our project with those of three referenced scholarly articles.

TABLE 2: COMPARATIVE ANALYSIS OF OUR PROJECT VS THREE REFERENCED SCHOLARLY ARTICLES

Feature	**Our Project**	**Autonomous Driving Robot Wheelchairs**	**Wireless Gesture-Controlled Wheelchair**	**Smart Wheelchair for Aged People**
Health Monitoring	Yes (Temperature, humidity, heart rate, SpO2) digital sensors	Not mentioned	No	Yes (Analog sensors)
Microcontroller	Arduino Due	Not specified	Arduino Uno	Arduino Uno and Raspberry Pi
Autonomous Navigation	Yes (Ultrasonic sensors for obstacle avoidance)	Yes	No	Not primarily

User Interface	LCD Display for real-time data display	Not specified	No	No
Comprehensive Mobility Solution	Yes (Beyond mobility aid), the Gyroscope of the phone is interfaced with Bluetooth.	Autonomous driving focus	Focus on gesture control	Health monitoring focus
Obstacle Detection Safety	Yes (HC-SR04 Ultrasonic Sensor)	Yes (Ultrasonic sensors)	No	Not primarily
Processing Power and Flexibility	High (Arduino Due)	Not specified	Moderate (Arduino Uno)	Moderate (Arduino Uno)

This comparative analysis illustrates our project's edge in several key areas. Notably, our integration of health monitoring tools sets our wheelchair apart, providing not just mobility but also health oversight. The use of Arduino Due represents an advancement in processing power and flexibility over the typical Arduino Uno microcontrollers used in similar projects. Furthermore, our approach to comprehensive mobility, combining both navigational autonomy and health monitoring, is unique in this field. The incorporation of Bluetooth connectivity and an interactive user interface also significantly enhances user engagement and convenience.

IX. Examining Important Questions

A. How our project can be turned into an autonomous one?

Our project was built based on it being the elemental first step in the direction of modern smart healthcare wheelchairs and already has the potential to be autonomous. But in the report, we will go over the ethical factors on why the control is to be with the patient operating the wheelchair rather than it being completely autonomous and technical future advancements to make it a completely autonomous one. The project is semi-autonomous with the use of an ultrasonic sensor that automatically prevents the falling of the patient by stopping the wheelchair by displaying "Obstacle detected". We can push our boundaries beyond one sensor and improve technical metrics in software and hardware departments to make it completely autonomous. Some of these are mentioned below:

1. *Sensors for Environmental Awareness:*
   - i. LIDAR/Radar: These sensors can provide 360-degree awareness of the wheelchair's surroundings, detecting obstacles, walls, and other objects in the environment rather than only the forward-facing ultrasonic sensors.
   - ii. Cameras: Implementing computer vision through cameras can help in recognizing specific objects, signs, or pathways.
2. *Advanced Navigation System*
   - *i.* GPS Integration: For outdoor navigation, a GPS module can be implemented and can guide the wheelchair to predefined destinations.
   - *ii.* Path Planning Algorithms: Implement algorithms like A* or Dijkstra for efficient pathfinding in complex environments.
   - *iii.* SLAM (Simultaneous Localization and Mapping): Essential for indoor navigation, allowing the wheelchair to build a map of its environment while keeping track of its location.
3. *Machine Learning for Adaptive Behavior*
   - i. Obstacle Avoidance: Train a machine learning model to recognize and avoid various obstacles and with neural networks and schema introductions we can integrate other apps such as Google Maps and integrate APIs into wheelchairs for better navigation and automatic motion capabilities linked with our hardware prototype.
   - ii. User Behavior Learning: The system can adapt to the user's preferences and habits, modifying its navigation strategy accordingly.
4. *User Interaction and Control Mechanisms*
   - i. Voice Commands: Allow users to control the wheelchair and set destinations using a voice module (E.g.: APR9600 module).
5. *Safety Features*
   - i. Fail-Safe Mechanisms: In case of system failure, the wheelchair should safely stop or switch to manual mode.
   - ii. Integration of SOS: Sends current live updates on the patient's health status and emergency treatments and medication suggestions to the nearby hospital to alert them in case of emergencies.
6. *Power Management and Efficiency*
   - i. Battery Life Optimization: Implement power-saving modes and efficient routing to extend battery life.
   - ii. Solar Panels (Optional): For outdoor use, solar panels can provide additional power.
7. *Connectivity and Remote Monitoring*
   - i. Wi-Fi: For real-time data transmission and remote monitoring by caregivers or medical professionals.
   - ii. Integration with Smart Devices with IoT: Enable the wheelchair to communicate with smart home devices for an integrated experience.
8. *Regulatory Compliance and Testing*
   - i. Compliance with Standards: Ensure the design meets all relevant safety and accessibility standards.
   - ii. Extensive Testing: Rigorous testing in various scenarios to ensure reliability and safety.
   - iii. Implementation Challenges
   - iv. Complexity and Cost: Developing a fully autonomous system can be complex and costly.

v. Reliability in Diverse Environments: Ensuring consistent performance in different environmental conditions.
vi. User Trust and Acceptance: Gaining the trust of users, especially in terms of safety and reliability.

B. How do you distribute and manage power among the various components during the voltage shift?

In our smart wheelchair project, managing and distributing power efficiently among various components is crucial for optimal performance and longevity. Here's how power management and distribution can be addressed in our project:

1. Power Board and Voltage Conversion

i. Central Power Source: A 12V battery with 1.3Ah serves as the central power source for the entire system.
ii. Power Board Utilization: The power board plays a vital role in stepping down the 12V supply to different voltage levels required by various components. This is achieved through voltage regulators or DC-DC converters.

2. Distribution of Power to Components

i. Arduino Due (3.3V): The Arduino Due is powered by 3.3V, which is lower than the standard 5V used by many other microcontrollers. To avoid complications, we fixed our input voltage to be 3.3V which is the same as the operating voltage of the Arduino Due. The power board steps down the 12V to 3.3V specifically for the Arduino Due, ensuring it receives the correct voltage without damage.
ii. Sensors (Ultrasonic, Temperature) and HC-05 Bluetooth Module (5V): These components require a 5V supply. The power board efficiently converts the 12V supply to 5V to power these components. This step-down process is critical as providing a higher voltage than required could damage these sensitive components.

3. Motor Driver L293D and Motors

i. The L293D motor driver is used to control the wheelchair's motors.
ii. It receives its power directly from the power board converter. This is essential as the motor driver needs to handle higher currents than the motors draw.
iii. The actual powering of the motors is managed by the L293D, which can handle the high current requirements of the motors while protecting the rest of the circuitry from these high loads.

4. Power Management Considerations

i. Efficiency: The step-down converters must be efficient to minimize heat generation and power loss.
ii. Thermal Management: Proper heat dissipation mechanisms, like heat sinks or cooling systems, should be in place to manage the heat produced by voltage conversion (Eg: Motor driver can heat up leading to real time prototype system testing errors).
iii. Protection Circuits: Incorporating protection circuits (like overvoltage and current protection) ensures the safety of the components and the user.
iv. Battery Life Optimization: Monitoring battery levels and implementing power-saving modes where possible can help extend the battery life.

5. Integration and Testing

i. Circuit Integration: Careful integration of the power board with the rest of the circuit is crucial. This includes proper wiring and ensuring stable connections.
ii. Testing Under Load: It's important to test the system under various load conditions to ensure the power distribution remains stable and efficient.

In summary, our project uses a centralized power board to step down a 12V battery to the required voltage levels for different components like the Arduino Due, sensors, HC-05 Bluetooth module, and the motor driver L293D. This approach ensures that each component receives the correct voltage for optimal operation while maintaining overall system efficiency and safety.

X. Acknowledgement

We would like to extend my sincere thanks to Dr. Fadi El-Hassan for his invaluable support and guidance throughout the duration of this project. His expertise and insightful perspectives have been fundamental in shaping the direction and success of our work. Additionally, we are grateful to the teaching assistant, Swathi Nagarajan, whose assistance and dedication have been instrumental in overcoming numerous challenges and enhancing our project's quality. Their combined efforts have not only contributed significantly to this project but have also enriched our learning experience.

XI. Conclusion

The development of the Smart Wheelchair project represents a significant step forward in the realm of assistive technology, blending advanced electronics with practical applications to enhance the quality of life for individuals with mobility impairments. Throughout this project, we have demonstrated how integrating Bluetooth connectivity and health monitoring systems with the Arduino Due microcontroller can lead to the creation of a highly functional and user-friendly mobility aid. Our efforts have culminated in a wheelchair that not only provides enhanced mobility but also offers vital health monitoring, ensuring the safety and well-being of its users. The incorporation of ultrasonic sensors, temperature and humidity sensors, and heart rate monitors, in conjunction with sophisticated control systems like the L293D motor driver and Bluetooth modules, showcases the potential of technology in addressing real-world challenges. Looking ahead, the possibilities for enhancing this project are boundless. The integration of AI and ML for improved navigation, the expansion into IoT connectivity for remote health monitoring, and the exploration of more advanced health monitoring features are just a few pathways for future development. The potential for voice control integration and autonomous navigation also opens new avenues for making the wheelchair even more user centric.

In conclusion, the Smart Wheelchair project stands as a testament to the power of technology in transforming lives. It underscores our commitment to innovation and our dedication to improving accessibility for those with physical limitations. As we continue to refine and enhance this project, we remain focused on our mission to deliver a solution that is not just a mobility aid but a comprehensive support system for its users.

REFERENCES

[1] Institute of Electrical and Electronics Engineers, *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS) : 6-7 Jan. 2017.*

[2] P. N. Sudha, S. V Ghorpade, and V. Sham Bhatt, “A Smart Wheelchair for Aged People with Health-Monitoring System,” 2022, doi: 10.35291/2454-9150.2021.0389.

[3] S. J. Kim, J. Park, Y. Na, and H. Won, “Autonomous Driving Robot Wheelchairs for Smart Hospitals,” in *2022 IEEE International Conference on Consumer Electronics-Asia, ICCE-Asia 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICCE-Asia57006.2022.9954833.

[4] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, “Brain-computer interface technology: a review of the first international meeting,” IEEE transactions on rehabilitation engineering, vol. 8, no. 2, pp. 164–173, 2000.

[5] R. Zhang, Y. Li, Y. Yan, H. Zhang, S. Wu, T. Yu, and Z. Gu, “Control of a wheelchair in an indoor environment based on a brain-computer interface and automated navigation,” IEEE transactions on neural systems and rehabilitation engineering, vol. 24, no. 1, pp. 128–139, 2015.

[6] H. Wang, X. Dong, Z. Chen, and B. E. Shi, “Hybrid gaze/EEG brain-computer interface for robot arm control on a pick and place task,” in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2015, pp. 1476– 1479.

[7] L.-D. Liao, C.-Y. Chen, I.-J. Wang, S.-F. Chen, S.-Y. Li, B.-W. Chen, J.-Y. Chang, and C.-T. Lin, “Gaming control using a wearable and wireless EEG-based brain-computer interface device with novel dry foam-based sensors,” Journal of neuroengineering and rehabilitation, vol. 9, no. 1, p. 5, 2012.

[8] K. LaFleur, K. Cassady, A. Doud, K. Shades, E. Rogin, and B. He, “Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface,” Journal of neural engineering, vol. 10, no. 4, p. 046003, 2013.

[9] J. KKevinand A. Subasi, “Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system,” Biomedical Signal Processing and Control, vol. 31, pp. 398–406, 2017. 30

[10] M. Pal and S. Bandyopadhyay, “Many-objective feature selection for motor imagery EEG signals using differential evolution and support vector machine,” in 2016 International Conference on Microelectronics, Computing, and Communications (MicroCom). IEEE, 2016, pp. 1–6.

[11] S. Bhattacharyya, P. Rakshit, A. Konar, D. Tibarewala, and R. Janarthanan, “Feature selection of motor image signals using firefly temporal difference q-learning and support vector machine,” in International Conference on Swarm, Evolutionary, and Memetic Computing. Springer, 2013, pp. 534–545.

[12] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup, “Transfer learning in braincomputer interfaces,” IEEE Computational Intelligence Magazine, vol. 11, no. 1, pp. 20–31, 2016.

[13] W.-L. Zheng, Y.-Q. Zhang, J.-Y. Zhu, and B.-L. Lu, “Transfer components between subjects foreign-based emotion recognition,” in 2015 International Conference on Affective Computing and Intelligent Interaction (ASCII). IEEE, 2015, pp. 917–922

APPENDIX

I. ARDUINO DUE HARDWARE SKETCH

```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN 6
#define DHTTYPE    DHT11
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;

LiquidCrystal_I2C lcd(0x27,16,2);
#define REPORTING_PERIOD_MS 1000

const int trigPin = 2;
const int echoPin = 3;

MAX30100 sensor;
int a;
PulseOximeter pox;
long duration;
int distance;
uint32_t tsLastReport = 0;

void onBeatDetected()
{
    Serial.println("Beat!");
}

void setup() {
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(A0,OUTPUT);
    pinMode(A1,OUTPUT);
    pinMode(A2,OUTPUT);
    pinMode(A3,OUTPUT);
    Serial.begin(9600);
    Serial1.begin(9600);

 dht.begin();
    sensor_t sensor1;
    dht.temperature().getSensor(&sensor1);
    dht.humidity().getSensor(&sensor1);
    delayMS = sensor1.min_delay / 1000;
    lcd.init();
    lcd.backlight();
    if (!pox.begin()) {
        Serial.println("FAILED");
        for(;;);
    } else {
        Serial.println("SUCCESS");
    }
    pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
    pox.setOnBeatDetectedCallback(onBeatDetected);
}

void loop() {
  pox.update();
  if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
      Serial.print("Heart rate:");
      Serial.print(pox.getHeartRate());
      Serial.print("bpm / SpO2:");
      Serial.print(pox.getSpO2());
      Serial.println("%");
      sensors_event_t event;
      dht.temperature().getEvent(&event);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("T: ");
      lcd.setCursor(2,0);
      lcd.print(event.temperature);

      lcd.setCursor(9,0);
      lcd.print("HB: ");
      lcd.setCursor(12,0);
      lcd.print(pox.getHeartRate());
      while(Serial1.available())
      {
        char t=Serial1.read();
        //Serial.println(t);
        if(t=='F')
        {
          Serial.println("FORWARD");
          digitalWrite(trigPin, LOW);
          delayMicroseconds(2);
          digitalWrite(trigPin, HIGH);
          delayMicroseconds(10);
          digitalWrite(trigPin, LOW);
          duration = pulseIn(echoPin, HIGH);
          distance= duration*0.034/2;
          pox.begin();
          pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);

          if(distance>15)
          {
            digitalWrite(A0,LOW);
            digitalWrite(A1,HIGH);
            digitalWrite(A2,LOW);
            digitalWrite(A3,HIGH);
            lcd.setCursor(0,1);
            lcd.print("FORWARD");
            pox.begin();

pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);
          }
          if(distance<15)
          {
            digitalWrite(A0,LOW);
            digitalWrite(A1,LOW);
            digitalWrite(A2,LOW);
            digitalWrite(A3,LOW);
          pox.begin();
          pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);

          }
        }
        if(t=='B')
        {
          digitalWrite(A0,HIGH);
          digitalWrite(A1,LOW);
          digitalWrite(A2,HIGH);
          digitalWrite(A3,LOW);
          lcd.setCursor(0,1);
          lcd.print("REVERSE");
          pox.begin();
          pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);
        }
        if(t=='R')
        {
          digitalWrite(A0,LOW);
          digitalWrite(A1,HIGH);
          digitalWrite(A2,HIGH);
          digitalWrite(A3,LOW);
          lcd.setCursor(0,1);
          lcd.print("RIGHT");
          pox.begin();
          pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);
        }
        if(t=='L')
        {
          digitalWrite(A0,HIGH);
          digitalWrite(A1,LOW);
          digitalWrite(A2,LOW);
          digitalWrite(A3,HIGH);
          lcd.setCursor(0,1);
          lcd.print("LEFT");
          pox.begin();
          pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);
        }
        if(t=='S')
        {
          digitalWrite(A0,LOW);
          digitalWrite(A1,LOW);
          digitalWrite(A2,LOW);
          digitalWrite(A3,LOW);
          pox.begin();
          pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

pox.setOnBeatDetectedCallback(onBeatDetected);

        }
      }
      tsLastReport = millis();
  }

}

void configureMax30100() {
  sensor.setMode(MAX30100_MODE_SPO2_HR);
  sensor.setLedsCurrent(MAX30100_LED_CURR_50MA,
MAX30100_LED_CURR_27_1MA);

sensor.setLedsPulseWidth(MAX30100_SPC_PW_1600US_16BITS)
;
  sensor.setSamplingRate(MAX30100_SAMPRATE_100HZ);
  sensor.setHighresModeEnabled(true);
}
end
```

II. Code Explanation

Key/Legend - Heartbeat Sensor codes
Ultrasonic Sensor codes
Bluetooth module codes
Temperature Sensor codes
LCD codes
Additional libraries

A. *Initial Setup and Definitions*

Library Inclusions: Wire.h, MAX30100_PulseOximeter.h, LiquidCrystal_I2C.h, Adafruit_Sensor.h, DHT.h, DHT_U.h are libraries included for interfacing with the respective sensors and the LCD display.
Pin Definitions: Pins for the DHT11 sensor (DHTPIN) and the ultrasonic sensor (trigPin, echoPin) are defined for interfacing with the Arduino Due.
Sensor Initialization: The MAX30100 pulse oximeter and DHT11 sensor are initialized. The LiquidCrystal_I2C lcd object is created for the LCD display.

Breaking down the Initial Setup and Definitions code segment
Include Statements and Definitions
#include <Wire.h>: Includes the Wire library for I2C communication.
#include "MAX30100_PulseOximeter.h": Includes the library for the MAX30100 pulse oximeter sensor.
#include <LiquidCrystal_I2C.h>: Includes the library for the I2C LCD display.
#include <Adafruit_Sensor.h>, #include <DHT.h>, #include <DHT_U.h>: Libraries for the DHT sensor for temperature and humidity.
#define DHTPIN 6: Defines the pin number to which the DHT sensor is connected.

#define DHTTYPE DHT11: Specifies the type of DHT sensor used (DHT11).
DHT_Unified dht(DHTPIN, DHTTYPE);: Creates a unified DHT sensor object.
LiquidCrystal_I2C lcd(0x27,16,2);: Initializes the LCD display with the I2C address 0x27, 16 characters wide and 2 lines.
const int trigPin = 2; const int echoPin = 3;: Defines the pins for the ultrasonic sensor.
MAX30100 sensor;: Creates an object for the MAX30100 sensor.
PulseOximeter pox;: Creates a Pulse Oximeter object for heart rate and oxygen saturation measurement.

B. *Main Setup Function (setup())*

Pin Mode Configuration: Sets the trigPin and echoPin for the ultrasonic sensor as output and input respectively.
Serial Communication: Initializes serial communication for debugging and data display on the serial monitor.
Sensor and Display Initialization: Initializes the DHT sensor, pulse oximeter, and LCD display. Error handling is included for the pulse oximeter (pox.begin()).

Breaking down setup function code segment

pinMode calls: Sets the pin modes for ultrasonic sensor pins and motor control pins.
Serial.begin(9600); Serial1.begin(9600);: Initializes serial communication.
dht.begin();: Initializes the DHT sensor.
lcd.init(); lcd.backlight();: Initializes and turns on the backlight of the LCD display.
pox.begin();: Initializes the pulse oximeter.
pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);: Sets the current for the IR LED of the pulse oximeter.
pox.setOnBeatDetectedCallback(onBeatDetected);: Sets a callback function that is called when a heartbeat is detected.

C. *Main Loop Function (loop())*

Pulse Oximeter Update: Regularly updates the pulse oximeter reading.
Sensor Readings and Display Updates:
Retrieves and displays temperature from the DHT sensor on the LCD.
Displays heart rate information from the pulse oximeter.
Bluetooth Communication Handling: Reads commands from Serial1 (Bluetooth module) and executes actions like moving forward, reversing, and turning, based on the received character ('F', 'B', 'R', 'L', 'S').
Motor Control: Controls the motors through digital pins (e.g., A0, A1, A2, A3) based on Bluetooth commands and ultrasonic sensor readings for obstacle avoidance.

Distance Measurement and Obstacle Avoidance
Ultrasonic Sensor Operation: Sends a pulse and measures the time taken for the echo to return, calculating the distance to an obstacle.
Conditional Motor Control: If an obstacle is detected within a certain distance (e.g., 15 cm), it stops or changes the wheelchair's direction.

Heartbeat Functions
onBeatDetected() Callback: A simple function that prints "Beat!" when a heartbeat is detected.
configureMax30100() Function: Configures various parameters of the MAX30100 sensor for accurate readings.

Breaking down loop and miscellaneous function code segment
pox.update();: Updates the pulse oximeter readings.

if (millis() - tsLastReport > REPORTING_PERIOD_MS) { ... }: Executes the enclosed code once every defined reporting period (1000ms).
Inside the if-statement:
Heart rate and SpO2 values are read from the pulse oximeter and printed on the Serial monitor.
Temperature is read from the DHT sensor and displayed on the LCD.
Bluetooth command handling ('F', 'B', 'R', 'L', 'S'): Controls the wheelchair's movement based on Bluetooth commands.
Ultrasonic sensors are used for distance measurement and obstacle avoidance.
Motor control logic to drive the wheelchair in different directions based on the commands received.

onBeatDetected Function
When a heartbeat is detected by the pulse oximeter, "Beat!" is printed to the Serial monitor.

configureMax30100 Function
Configures various parameters of the MAX30100 sensor for accurate readings.

XII. AVR PROTEUS CODE (ARDUINO MEGA 2560)

```
#include "dht11.h"
#include "LCD.h"
#include <TimerOne.h>
DHT dht;
char c;
#define D_temp 4
#define motor1_pin1 22
#define motor1_pin2 23
#define motor2_pin1 24
#define motor2_pin2 25
#define trigpin 2
#define echopin 3

int pos = 0;  // variable to store the servo
int HBSensor = 7; //   HB i/p
int HBCount = 0; //    HB start button
int HBCheck = 0;
int TimeinSec = 0;
int HBperMin = 0;
int HBStart = 6;
int HBStartCheck = 0;

void timerIsr()
{
  if(HBStartCheck == 1)
  {
      TimeinSec = TimeinSec + 1;
      lcd.setCursor(14,0);
      lcd.print(TimeinSec);
      lcd.print(" ");
  }
}

void setup()
{
  lcd.begin(16, 2);
  lcd.clear();
  Serial.begin(9600);
  pinMode(2, OUTPUT);//ULTRASONIC op
  pinMode(3, INPUT); //ULTRASONIC ip
  pinMode(27, INPUT_PULLUP); //F
  pinMode(28, INPUT_PULLUP); //B
  pinMode(29, INPUT_PULLUP); //S
  pinMode(30, INPUT_PULLUP); //L
  pinMode(26, INPUT_PULLUP); //R
  pinMode(motor1_pin1,  OUTPUT);      /*  Motor1
control pin 1 */
  pinMode(motor1_pin2,  OUTPUT);      /*  Motor1
control pin 2 */
  pinMode(motor2_pin1,  OUTPUT);      /*  Motor2
control pin 1 */
  pinMode(motor2_pin2,  OUTPUT);      /*  Motor2
control pin 2 */
  pinMode(HBSensor, INPUT); // HB i/p
  pinMode(HBStart,  INPUT_PULLUP);  //HB  start
Button
  Timer1.initialize(800000);
  Timer1.attachInterrupt( timerIsr );
  lcd.clear();
  lcd.setCursor(0, 0);
  delay(250);
  lcd.print("WHEELCHAIR");
  delay(1000);
  lcd.clear();
  lcd.setCursor(9,0);
  lcd.print("Time:");
  lcd.setCursor(0,1);
  lcd.print("HBperMin: 0");

}
//Function To Recieve Data From Bluetooth Device
and move
void loop()
{ //DHT11 reading
  dht.dht_read(D_temp);
  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(dht.temperature);
  //HB Sensor Reading
  if(digitalRead(HBStart) == LOW){HBStartCheck =
1;}
  if(HBStartCheck == 1)
  {
      if((digitalRead(HBSensor)   ==   HIGH)   &&
(HBCheck == 0))
      {
        HBCount = HBCount + 1;
        HBCheck = 1;
        lcd.setCursor(13,1);
        //lcd.print(HBCount);
        //lcd.print(" ");
      }
      if((digitalRead(HBSensor)   ==   LOW)   &&
(HBCheck == 1))
      {
        HBCheck = 0;
      }
      if(TimeinSec == 10)
      {
          HBperMin = HBCount * 6;
          HBStartCheck = 0;
          lcd.setCursor(10,1);
          lcd.print(HBperMin);
          lcd.print(" ");
          HBCount = 0;
          TimeinSec = 0;
      }
  }
//Ultrasonic detection
  long duration, distance;
  digitalWrite(2, LOW); //
  delayMicroseconds(2);
  digitalWrite(2, HIGH);
  delayMicroseconds(10);
  digitalWrite(2, LOW);
  duration = pulseIn(3, HIGH);
  distance = duration/58.2;
  if(distance < 30) {
    lcd.setCursor(0, 1);
    lcd.print("Obstacle ahead ");
    Serial.println("Obstacle ahead ");
    // Stop the motors
    digitalWrite(motor1_pin1, LOW);
    digitalWrite(motor1_pin2, LOW);
    digitalWrite(motor2_pin1, LOW);
    digitalWrite(motor2_pin2, LOW);
    delay(400);
    lcd.clear();
  }
  else
  {
    while  (Serial.available())  //Function  To
Recieve Data From Bluetooth Device and move
    {
      char c = (char)Serial.read();
      if (c == 'f')
      {
        Serial.print("FORWARD");
        lcd.setCursor(0, 1);
        lcd.print("FORWARD      ");
        digitalWrite(motor1_pin1, HIGH);
        digitalWrite(motor1_pin2, LOW);
        digitalWrite(motor2_pin1, HIGH);
        digitalWrite(motor2_pin2, LOW);

      }
      if (c == 'b')
      {
        Serial.print("REVERSE");
        lcd.setCursor(0, 1);
        lcd.print("REVERSE     ");
        digitalWrite(motor1_pin1, LOW);
        digitalWrite(motor1_pin2, HIGH);
        digitalWrite(motor2_pin1, LOW);
        digitalWrite(motor2_pin2, HIGH);
      }
      if (c == 's')
      {
        Serial.print("STOP");
        lcd.setCursor(0, 1);
        lcd.print("STOP      ");
        digitalWrite(motor1_pin1, LOW);
        digitalWrite(motor1_pin2, LOW);
        digitalWrite(motor2_pin1, LOW);
        digitalWrite(motor2_pin2, LOW);
      }
      if (c== 'l')
       {
        Serial.print("LEFT");
        lcd.setCursor(0, 1);
        lcd.print("LEFT      ");
        digitalWrite(motor1_pin1, HIGH);
        digitalWrite(motor1_pin2, LOW);
        digitalWrite(motor2_pin1, LOW);
        digitalWrite(motor2_pin2, HIGH);
      }
      if (c== 'r')
       {
        Serial.print("RIGHT");
        lcd.setCursor(0, 1);
        lcd.print("RIGHT     ");
        digitalWrite(motor1_pin1, LOW);
        digitalWrite(motor1_pin2, HIGH);
        digitalWrite(motor2_pin1, HIGH);
        digitalWrite(motor2_pin2, LOW);
      }

      delay(1000);
    }
    if (digitalRead(27) == 0)
    {
      Serial.print("FORWARD");
      lcd.setCursor(0, 1);
      lcd.print("FORWARD    ");
      digitalWrite(motor1_pin1, HIGH);
      digitalWrite(motor1_pin2, LOW);
      digitalWrite(motor2_pin1, HIGH);
      digitalWrite(motor2_pin2, LOW);

    }
    if (digitalRead(28) == 0)
    {
      Serial.print("REVERSE");
      lcd.setCursor(0, 1);
      lcd.print("REVERSE    ");
      digitalWrite(motor1_pin1, LOW);
      digitalWrite(motor1_pin2, HIGH);
      digitalWrite(motor2_pin1, LOW);
      digitalWrite(motor2_pin2, HIGH);
    }
    if (digitalRead(29) == 0)
    {
      Serial.print("STOP");
      lcd.setCursor(0, 1);
      lcd.print("STOP       ");
      digitalWrite(motor1_pin1, LOW);
      digitalWrite(motor1_pin2, LOW);
      digitalWrite(motor2_pin1, LOW);
      digitalWrite(motor2_pin2, LOW);
    }
    if (digitalRead(30) == 0)
    {
      Serial.print("LEFT");
      lcd.setCursor(0, 1);
      lcd.print("LEFT      ");
      digitalWrite(motor1_pin1, HIGH);
      digitalWrite(motor1_pin2, LOW);
      digitalWrite(motor2_pin1, LOW);
      digitalWrite(motor2_pin2, HIGH);
    }
    if (digitalRead(26) == 0)
    {
      Serial.print("RIGHT");
      lcd.setCursor(0, 1);
      lcd.print("RIGHT      ");
      digitalWrite(motor1_pin1, LOW);
      digitalWrite(motor1_pin2, HIGH);
      digitalWrite(motor2_pin1, HIGH);
      digitalWrite(motor2_pin2, LOW);
    }
  }
  delay(1000);
}
```