e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

# SMART WHEELCHAIR PROTOTYPE

**Dr. M N Jayaram*1, Pooja M Nagali*2, Samantha Menezes*3, Chethan Kumar Ks*4, N Yeshwanth*5**

*1HOD And Associate Professor, ECE Department, JSS Science And Technology University, Mysore, Karnataka, India.

*2,3,4,5Student, ECE Department, JSS Science And Technology University, Mysore, Karnataka, India.

## ABSTRACT

Smart wheelchair prototype model is developed and it includes key, voice, gesture features along with obstacle detection and the speed control. The wheelchair prototype model can be controlled by using the application on the smartphone which is developed. Key feature is used when a person can see but cannot speak. Voice feature is used when a person can speak but has lost hands or legs in war and has got delicate fingers. Gesture can be used when a person can move wrist but can't speak or can't see. Hence key, voice, gesture and obstacles features are developed to overcome these problems. The application on the smartphone and the wheelchair are connected with the help of Bluetooth module. Motor driver circuit, ultrasonic sensor and Arduino atmega 2560 are used to achieve the expected result. The features can be used on the real wheelchair without any modification but ultrasonic sensor, wireless connectivity module and the motors needs to be changed. This paper gives an overview of the prototype developed and the features achieved.

**Keywords:** Bluetooth Module, Ultrasonic Module, Atmega 2560 Microcontroller, Motor Driver Circuit, PWM.

## I. INTRODUCTION

A wheelchair is a manually operated or power-driven device designed primarily for use by an individual with a disability. Most manual wheelchairs have push handles at the top of the back to allow manual propulsion by helping people. In the case of a user-propelled manual wheelchair, the controlling and driving of the wheelchair is done by disabled users. But the hand work is more straining as compared to legs. Long interval use of manual wheelchairs leads to lower physical capability of the user; hence it is not advisable.

A smart wheelchair typically consists of a standard wheelchair, microcontrollers and a collection of sensors. Till today, Smart wheelchairs are developed that provide navigation help to the user in a number of different ways, such as assuring accident-free travel, helping the performance of definite tasks, and autonomously transporting the user between locations. The purpose of a smart wheelchair is to reduce or eliminate the user's responsibility to drive a wheelchair. This paper presents a brief wheelchair prototype design.

## II. METHODOLOGY

The major blocks of the project are as shown in the block diagram of the project as shown.

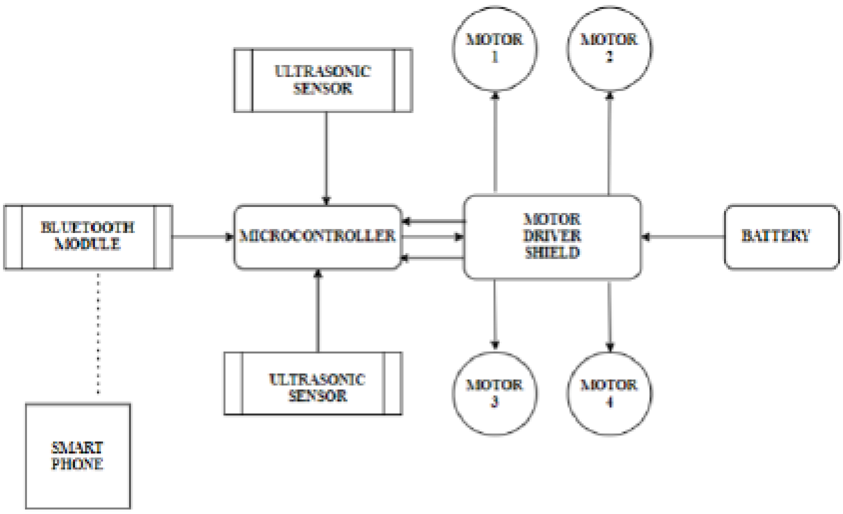


**Fig 2.1:** Proposed Block Diagram

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

User needs to pair the Bluetooth module on the wheelchair with the Bluetooth present on the smart phone for all the features, then they can control the movement of the wheelchair by keypad, voice, gesture-controlled features along with obstacle detection. The speed can be controlled for all the features using PWM concept.

**Key controlled feature**

The keypad control can be done by pressing the keys on the smartphone through the app. It has five keys forward, reverse, right, left and stop. The code is written on the IDE and when the key pressed on the app becomes equal to the key which is being read by the microcontroller then according to the selected key on the app the bot will move in that direction and speed control is achieved with the help of motors and motor driver circuit and the microcontroller.

**Voice**

The voice control can be done by giving voice commands on the smartphone through the app. It has five commands forward, reverse, right, left and stop. The code is written on the IDE and the voice command is read and processed on the app. If the command processed on the app and the command which is written in the IDE are compared once the strings are equal the bot moves in that direction and speed control is achieved with the help of motors and motor driver circuit and the microcontroller.

**Gesture**

The gesture control can be done by using the accelerometer on the smartphone and the app on the smartphone. It has five keys forward, reverse, right, left and stop. The code is written on the IDE and when the inclination becomes equal to or greater than the set inclination for each of the directions then the bot moves in that direction and speed control is achieved with the help of motors and motor driver circuit and the microcontroller.

**Key, voice, gesture, obstacle controlled wheelchair prototype operations**

All the 3 features are integrated in a single app. The app has voice recognition module, Bluetooth connectivity module, accelerometer module, clock along with other keys. For key, voice, gesture forward, reverse, left, right and stop along with speed control are successfully verified. The ultrasonic is fitted on front and back of the chassis. Desired distance is set. In our case it is 35 cm. If there is an obstacle present at a distance lesser than the set distance the bot starts moving and stands still at the same position until obstacle is moved.

## III. MODELING AND ANALYSIS

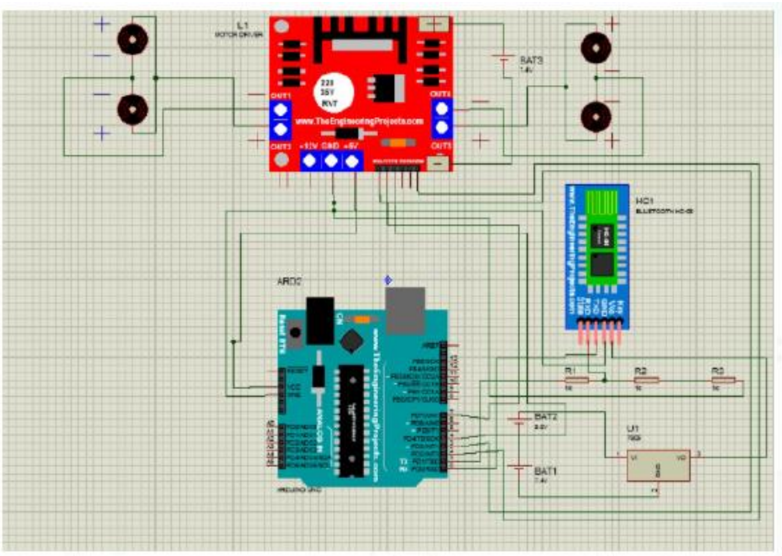

**Fig 3.1** Circuit design without ultrasonic sensors

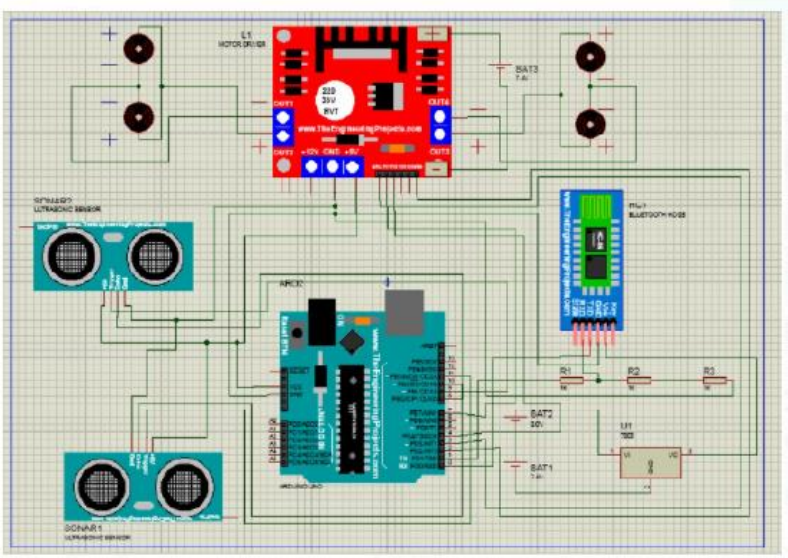

**Fig 3.2** Circuit design with ultrasonic sensors

The microcontroller used is Atmega 2560. The microcontroller is interfaced with the Bluetooth HC-05 module. The transmitter and receiver of the microcontroller is cross connected with the transmitter and receiver of the Bluetooth module. As HC-05 Bluetooth module has 3.3 V level for RX/TX, we make use of voltage regulator 7805 which takes 7v input and gives 5v output. Then a potential divider network is used to connect to the transmitter and receiver network. Ultrasonic sensors are used, one is connected to front and one is at the back. Motor driver L293D is interfaced with the Arduino microcontroller. The enable pins of the motor driver are connected to the microcontroller to control the speed and direction of rotation of the motors. Four BO motors are attached by shorting the terminals on the left side and by shorting the terminals on the right side. For using the application, the Bluetooth has to be paired with the Bluetooth of the smartphone.

**Software working**

**KEYPAD CONTROLLED** : MIT app is assigned for keypad controlled operation. It has five keys. Forward, Backward, Turn Left, Turn Right and Stop


**Fig 3.3:** Keypad Controlled

(a) Forward: Forward key is pressed on the app and upon pressing 'F' is read by the microcontroller. In the C++ code, Serial.read() function keeps checking for ASCII values and stores them in state variables. When (state == 'F') then all four motors rotate in clockwise direction.

(b) Backward: Reverse key is pressed on the app and upon pressing 'B' is read by the microcontroller. When (state == 'B') then all four motors rotate in an anticlockwise direction.

(c) Stop: When the stop key is pressed on the app and upon pressing 'S' is read by the microcontroller. When (state == 'S') all four motors will stop rotating.

(d) Turn Left: When the Turn Left key is pressed on the app and upon pressing 'L' is read by the microcontroller. When(state == 'L') two motors at right move forward. Left two motors will stop.

(e) Turn Right: When the Turn Right key is pressed on the app and upon pressing 'R' is read by the microcontroller. When (state == 'R') two motors at left rotate clockwise. Right two motors will be idle.

**VOICE CONTROLLED**: Speech recognition module is used on the MIT app. Microphone of the smartphone is used to recognize the speech and microcontroller processes, validate the data and rotate the motor.


**Fig 3.4:** Voice Controlled

(a) Forward : microphone checks if the same is spoken by the user . If yes , all four motors are rotated forward.
(b) Backward : microphone checks if the same is spoken by the user . If yes , all four motors are rotated backwards.
(c) Stop : microphone checks if the same is spoken by the user . If yes , all four motors are halted.
(d) Turn Left : microphone checks if the same is spoken by the user . If yes , two motors at right move forward , two motors at left remain halted.
(e) Turn Right : microphone checks if the same is spoken by the user . If yes , two motors at left rotate clockwise, two motors at right remain idle.

**KEY, VOICE, OBSTACLE AND GESTURE** all features are integrated to form a single app.

KEY CONTROLLED

(a) Forward : Forward key is pressed on the app and upon pressing 'F' is read by the microcontroller. In the C++ code, Serial.read() function keeps checking for ASCII values and stores them in state variables. When (state == 'F') then all four motors rotate in clockwise direction.
(b) Backward : Reverse key is pressed on the app and upon pressing 'B' is read by the microcontroller. When (state == 'B') then all four motors rotate in an anticlockwise direction.
(c) Stop : When the stop key is pressed on the app and upon pressing 'S' is read by the microcontroller. When (state == 'S') all four motors will stop rotating.
(d) Turn Left : When the Turn Left key is pressed on the app and upon pressing 'L' is read by the microcontroller. When (state == 'L') two motors at right move forward. Left two motors will stop.
(e) Turn Right : When the Turn Right key is pressed on the app and upon pressing 'R' is read by the microcontroller. When (state == 'R') two motors at left rotate clockwise. Right two motors will be idle.

VOICE CONTROLLED

(a) Forward : microphone checks if the same is spoken by the user . If yes , all four motors are rotated forward.
(b) Backward : microphone checks if the same is spoken by the user . If yes , all four motors are rotated backwards.
(c) Stop : microphone checks if the same is spoken by the user . If yes , all four motors are halted.
(d) Turn Left : microphone checks if the same is spoken by the user . If yes , two motors at right move forward , two motors at left remain halted.

(e) Turn Right : microphone checks if the same is spoken by the user . If yes , two motors at left rotate clockwise , two motors at right remain idle.

OBSTACLE

While moving forward and backward if the distance between the chassis and the external object was lesser than 35cm then the motors stopped rotating immediately. When the motors stopped rotating that means there was an object at the front or behind the robot. This is only for prototypes. Else a different sensor needs to be used for obstacle detection to obtain greater range.

GESTURE

Accelerometer on the phone is used with the app to control the wheel on the chassis by rotating the motors. In all the features the motor speed was controlled by making use of PWM concept and key, voice, obstacle and gesture features were obtained by making use of the software and the hardware.


**Fig 3.5:** Key , Voice, Obstacle and Gesture Controlled

## IV. RESULTS AND DISCUSSION

Implementation of Keypad controlled is as follows

**Fig 4.1 :** Stop Key pressed **Fig 4.2:** Bot at rest

**Fig 4.3** : Forward key pressed

**Fig 4.4** : Bot moves forward until it detects obstacle


**Fig 4.5** : Backward key pressed

**Fig 4.6 :** Bot moves backward until it detects obstacle

**Fig 4.7**: Turn Left key pressed

**Fig 4.8:** Bot turns Left


**Fig 4.9:** Turn Right key pressed

**Fig 4.10:** Bot turns Right

Implementation of voice controlled is as follows


**Fig 4.11 :** command for bot to stop

**Fig 4.12 :** Bot at rest


**Fig 4.13 :** command to move forward

**Fig 4.14 :** Bot moves forward until it detects obstacle


**Fig 4.15 :** command to move backward

**Fig 4.16 :** Bot moves backward until it detects obstacle


**Fig 4.17 :** command to turn Left

**Fig 4.18** : Bot turns Left

**Fig 4.19 :** command to turn Right

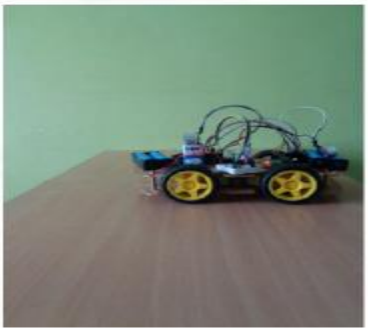

**Fig 4.20 :** Bot turns Right

Implementation of gesture controlled is as follows

**Fig 4.21:** Gesture for Bot to halt

**Fig4.22:** Bot at rest

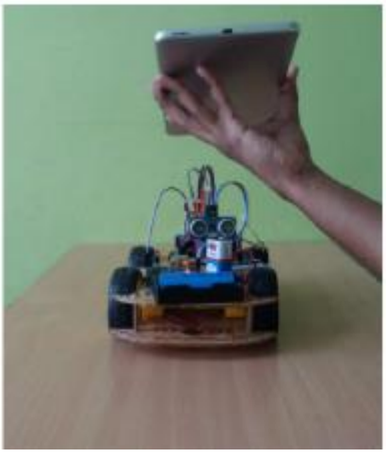

**Fig 4.23:** Gesture for Bot to move forward

**Fig 4.24 :** Bot moves forward until it detects obstacle

**Fig 4.25:** Gesture for Bot to move backward

**Fig 4.26 :** Bot moves backward until it detects obstacle

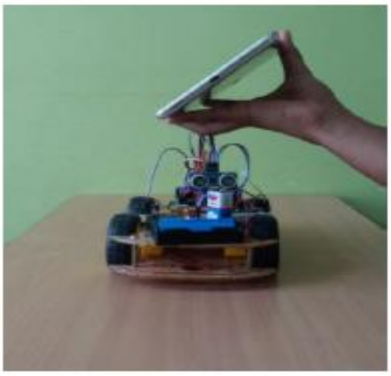

**Fig 4.27:** Gesture for Bot to turn Left

**Fig 4.28 :** Bot turns Left

**Fig 4.29**: Gesture for Bot to turn Right

**Fig 4.30:** Bot turns Right

## V. CONCLUSION

Till today, a considerable amount of work has been done in the field of smart wheelchairs but small amount of attention has been given to user centric wheelchair design. Most of the prototypes have been designed without considering user expectations. The prototype which we have designed does not compromise on user comfort. Usually, such designs have inexpensive hardware and software which are not affordable for most potential users. We have overcome this by integrating sensors into a limited space. The prototype can mainly be controlled by a smart phone which is an added advantage. The user interface is designed in such a way that people of any age group would not have a problem using our application. he inexpensive and lightweight sensors reduce the overall cost of the entire system and make it affordable. The weight of the system is further reduced by using light weight batteries and chassis. The system was successfully implemented to control the movement of a wheelchair in five directions left, right, forward, backward and stop by using our smartphone. Three modes of controlling were successfully implemented: gesture, voice and keypad along with obstacle detection.

## VI. REFERENCES

[1] Premangshu Chanda, Pallab Kanti Mukherjee, Subrata Modak."Gesture controlled robot using arduino and android". Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, West Bengal, India.

[2] Muddappa AP, Sangeetha AS, Jason M, Sahana B. “Automated wheelchair with gesture control". UG Scholar, Department of EEE, Vidyavardhaka college of Engineering, Mysuru, India.

[3] M.Prathyusha, K. S. Roy, Mahaboob Ali Shaik." Voice and Touch Screen Based Direction and Speed Control of Wheel Chair for Physically Challenged Using Arduino".Department of ECE, KL University, Guntur, India.

[4] Deepak Kumar Lodhi, Ritakshi Gupta, Manoj Kumar Pandey." Smart electronic wheelchair using Arduino and Bluetooth module".Department of electronics and Communication engineering, Amity University, Haryana, India.

[5] Tan kian Hou, Yagasena, Chelladurai. \Arduino based voice controlled wheelchair", School of Engineering, Faculty of Science and Technology, Quest International University, Malaysia.

[6] D Anjaneyulu, B Ajanta Reddy, Siva Kumar." An intelligent wheelchair controller using keyboard and accelerometer".Department of Electronics and Communication Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India.

[7] Krunal Bansod, Kushal Asarkar, Mandar Topre, Vikrant Raj." Hand gesture controlled wheelchair".International journal of scientic and technology research volume 5, april 2016.