



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 Issue: V Month of publication: May 2022

DOI: <https://doi.org/10.22214/ijraset.2022.43160>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Sensor Based Wheelchair for Crippled by Speech Recognition

Raj Singh¹, Sanjeev Gupta², Jeetesh³, Sahil Singh⁴, Abhishek Yadav⁵

^{1, 2, 3, 4, 5}Department of Mechanical Engineering, KIET Group of Institutions, Ghaziabad, 201206, Uttar Pradesh, India

Abstract: According to research there are about 6 million populations in the world who are paralyzed and needs a wheelchair for their mobility. Earlier the wheelchairs had to be moved and be externally supported by any person. To help overcome this “joystick-controlled wheelchairs” are developed. But in regular use, these joystick-controlled wheelchairs became difficult to use. Especially in the case of paralyzed people, the use of joystick became more difficult due to the hard buttons and unidirectional use of the joysticks. To overcome these problems, we’ve tried to develop a “gesture-controlled wheelchair” which can be moved with a slight tilt of the hand. This can be used in both hands and can be controlled to come to the user from a distance. The current work is implemented with Arduino based devices such as Arduino UNO processors and programmed through Arduino IDE.

Keywords: Sensor wheelchair, Arduino NANO, L293D, PCB, Bluetooth Module, Arduino IDE

I. INTRODUCTION

This project proposes an integrated approach to real time detection, tracking and direction recognition of Voice, which is intended to be used as a human-robot interaction interface for the intelligent wheelchair. This paper is to demonstrate that accelerometers can be used to effectively translate Voice into computer interpreted signals. For gesture recognition the accelerometer data is calibrated and filtered. The accelerometers can measure the magnitude and direction of gravity in addition to movement induced acceleration. To calibrate the accelerometers, we rotate the device “sensitive axis with respect to gravity and use the resultant signal as an absolute measurement. Integrating a single chip wireless solution with a MEMS accelerometer would yield an autonomous device small enough to apply to the fingernails, because of their small size and weight. Accelerometers are attached to the fingertips and back of the hand. Arrows on the hand show the location of accelerometers and their sensitive directions, that the sensitive direction of the accelerometer is in the plane of the hand. The Voice-based wheelchair is suitable for the elderly and the physically challenged people who are unfortunate to have lost ability in their limbs due to paralysis or by birth or by old age.

This sensor finds the tilt and makes use of the accelerometer to change the direction of the wheelchair depending on tilt. For example, if the tilt is to the right side, then the wheelchair moves in right direction or if the tilt is to the left side, then the wheelchair moves in left direction. Wheelchair movement can be controlled in Forward, Reverse, and Left and Right direction along with obstacle detection using ultrasonic sensor. Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. One among the technologies, which had greater developments, is the MEMS ACCELEROMETER SENSOR. These had greater importance than any other technologies due its user-friendly nature. MEMS ACCELEROMETER SENSOR based devices can be easily reachable to the common man due to its simpler operation.

A. Development Board

A microprocessor development board is a printed circuit board containing a microprocessor and the minimal support logic needed for an electronic engineer or any person that wants to become acquainted with the microprocessor on the board and to learn to program it. It also served users of the microprocessor as a method to prototype applications in products.

B. Circuit Design

A circuit is any loop through which matter is carried. For an electronic circuit, the matter carried is the charge by electronics and the source of these electrons is the positive terminal of the voltage source. When this charge flows from the positive terminal, through the loop and reaches the negative terminal, the circuit is said to be completed. However, this circuit consists of several components which affects the flow of charge in many ways. Some may provide hindrance to the flow of charge, some simple store or dissipate charge. Some require external source of energy, some supply energy

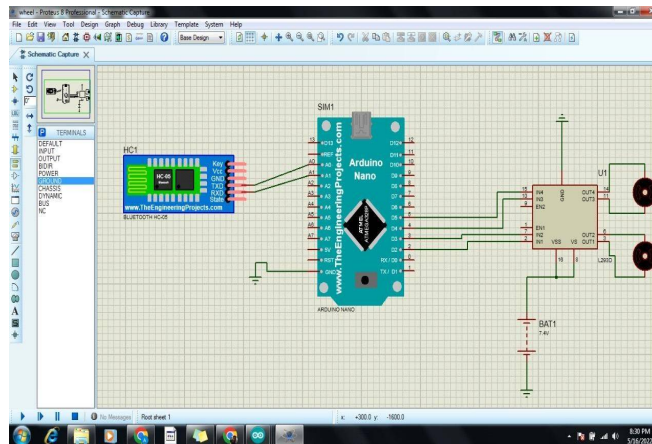


Figure-Circuit diagram

C. Block Diagram

The block diagram for whole project is given where power supply is given at initial from starting, rectifier and regulator are used then a display LCD is attached to them, and Arduino UNO with Bluetooth module and wheelchair are connected, and UNO receives the gesture from any of the four explained in diagram.

D. Hardware Required

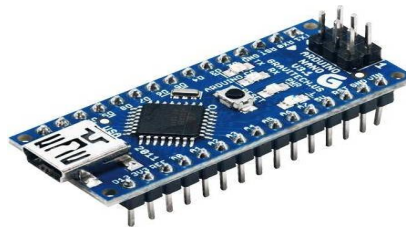
Arduino Nano, BLUETOOTH MODULE HC05, Mobile Device, DC Gear MOTOR, Resistors, Capacitors, Push Button, Charging Connector, 1200MAH Battery, Castor Wheel, Simple Wheel, 3D Printed Chassis, Cables & Connectors, Motor Driver, PCB, Adapter/Power Supply

E. Coding

Python is a beginner-friendly programming language that is used in schools, web development, scientific research, and in many other industries. Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and with Raspberry Pi lets you connect your project to the real world.

It can be done either on raspberry pi or by Arduino UNO, but we will prefer Arduino UNO.

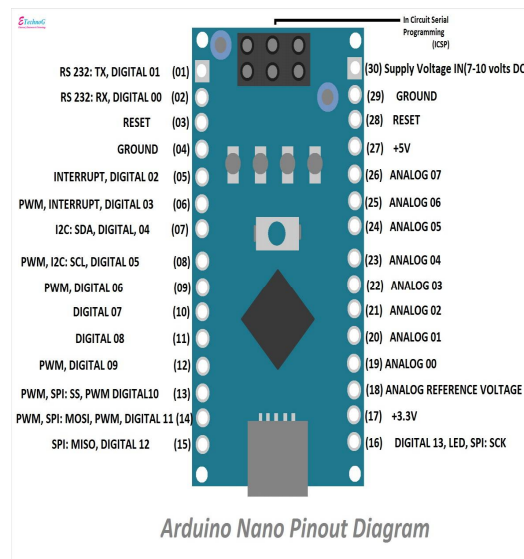
F. Arduino Nano



Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the Arduino language, inspired by the Processing language and used with a modified version of the Processing IDE. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go.

G. Arduino NANO Pinout Diagram

Arduino is an open-source hardware platform that is readily available for hobbyists & enthusiasts across the globe to build projects. It comes with an ATMEGA microcontroller that processes the data and facilitates the proper working of the IOT system



H. Features

- ATmega328P Microcontroller is from 8-bit AVR family
- Operating voltage is 5 V
- Input voltage (VIN) is 7V to 12V
- Input / Output Pins are 22
- Analog I / P pins are 6 from A0 to A5
- Digital Pin are 14
- KB SRAM is 2 KB
- EEPROM is 1 KB
- CLK speed is 16 MHz
- Size of the printed circuit board is 18 X 45mm
- Supports three communications like SPI, IIC, & USART

I. Bluetooth Module

Bluetooth module is utilized as UART RS232 serial converter module. It can efficiently exchange the UART data through the wireless Bluetooth, without complex piece design or profound learning in the Bluetooth programming stack. This module can be utilized as a part of Master or Slave Mode and simple switchable between these two modes, by default Slave mode is designed. To setup Wireless Serial Communication, HC-05 Bluetooth Module is most requesting and prominent because of its low cost and to a great degree high highlights.



Figure- Bluetooth module

J. PCB

A printed circuit board (PCB) mechanically supports and electrically connects electronic components or electrical components using conductive tracks, pads and other features etched from one or more sheet layers of copper laminated onto and/or between sheet layers of a non-conductive substrate. Components are generally soldered onto the PCB to both electrically connect and mechanically fasten them to it.



Figure-PCB

K. Capacitor Diodes and resistance

A diode is a two-terminal electronic component that conducts current primarily in one direction (asymmetric conductance); it has low (ideally zero) resistance in one direction,

The resistor is a passive electrical component to create resistance in the flow of electric current. In almost all electrical networks and electronic circuits they can be found. The resistance is measured in ohms. An ohm is the resistance that occurs when a current of one ampere passes through a resistor with a one volt drop across its terminals



Figure-Capacitor diode and resistance

L. L293D Motor Driver

A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors. The most commonly used motor driver IC's are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.



Figure - L293D Motor

II. HOW TO RUN PROJECT

Switch on the power supply of voice module to provide sufficient voltage for proper working

Switch on the interrupt switch and give your speech command as per requirement of movement of wheelchair to voice module.

After given the speech command switch off the interrupt switch to avoid any other noise interference to voice module from error.

Voice module generate hex code correspond to input speech. Ex, if we speak forward then output of voice module is 02h.

That code is applied to input port 1 of the AT89C52 micro controller.

Micro controller generate correspond code ex02h (0010) which is given to both motor driver to drive the motor in forward direction.

These steps are repeated for all commands

Table 1: Input and Output port for wheelchair motion

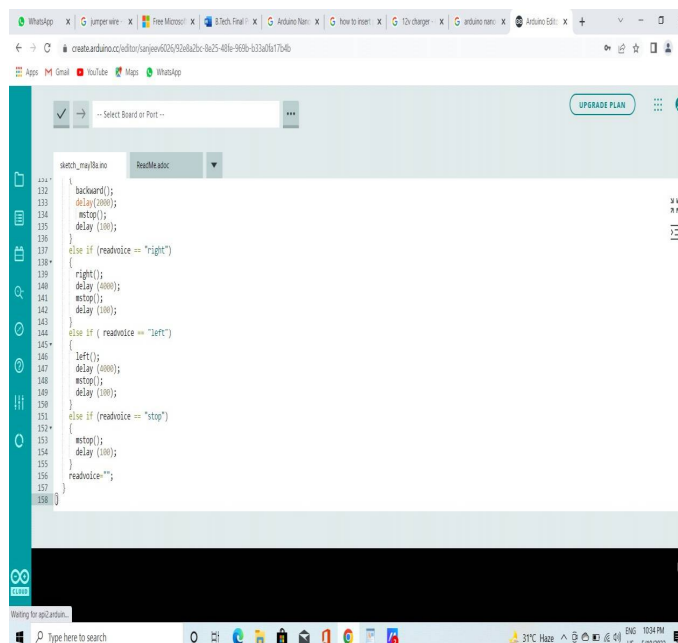
INPUT	INPUT AT PORT1	OUTPUT AT PORT2	WHEELCHAIR MOTION
02	0010	0A 1010	FORWORD
08	1000	05 0101	BACKWORD
04	0100	02 0010	LEFT
06	0110	08 1000	RIGHT
05	0101	03 0011	STOP

Table 2: List of Components

Name of components	Value
RESISTOR	10E
CAPACITOR	33pF
IC	AT89C52
WHEEL CHAIR	-
SOCKETS	40 PINS AT89C52
VOICE RECOGNIZATION MODULE	HM 2007
MISCELLANEOUS	CRYSTAL
MOTORS	DC GEAR MOTOR DRIVING CIRCUIT

III. CODING

We used Arduino IDE Software for programming to control our wheelchair



```

131 void backword()
132 {
133   digitalWrite(10, LOW);
134   digitalWrite(11, HIGH);
135   delay(1000);
136   digitalWrite(10, HIGH);
137   digitalWrite(11, LOW);
138   delay(1000);
139   digitalWrite(10, LOW);
140   digitalWrite(11, HIGH);
141   delay(1000);
142   digitalWrite(10, HIGH);
143   digitalWrite(11, LOW);
144   delay(1000);
145   digitalWrite(10, LOW);
146   digitalWrite(11, HIGH);
147   delay(1000);
148   digitalWrite(10, HIGH);
149   digitalWrite(11, LOW);
150   delay(1000);
151   digitalWrite(10, LOW);
152   digitalWrite(11, HIGH);
153   delay(1000);
154   digitalWrite(10, HIGH);
155   digitalWrite(11, LOW);
156   delay(1000);
157   digitalWrite(10, LOW);
158   digitalWrite(11, HIGH);
159   delay(1000);
160   digitalWrite(10, HIGH);
161   digitalWrite(11, LOW);
162   delay(1000);
163   digitalWrite(10, LOW);
164   digitalWrite(11, HIGH);
165   delay(1000);
166   digitalWrite(10, HIGH);
167   digitalWrite(11, LOW);
168   delay(1000);
169   digitalWrite(10, LOW);
170   digitalWrite(11, HIGH);
171   delay(1000);
172   digitalWrite(10, HIGH);
173   digitalWrite(11, LOW);
174   delay(1000);
175   digitalWrite(10, LOW);
176   digitalWrite(11, HIGH);
177   delay(1000);
178   digitalWrite(10, HIGH);
179   digitalWrite(11, LOW);
180   delay(1000);
181   digitalWrite(10, LOW);
182   digitalWrite(11, HIGH);
183   delay(1000);
184   digitalWrite(10, HIGH);
185   digitalWrite(11, LOW);
186   delay(1000);
187   digitalWrite(10, LOW);
188   digitalWrite(11, HIGH);
189   delay(1000);
190   digitalWrite(10, HIGH);
191   digitalWrite(11, LOW);
192   delay(1000);
193   digitalWrite(10, LOW);
194   digitalWrite(11, HIGH);
195   delay(1000);
196   digitalWrite(10, HIGH);
197   digitalWrite(11, LOW);
198   delay(1000);
199   digitalWrite(10, LOW);
200   digitalWrite(11, HIGH);
201   delay(1000);
202   digitalWrite(10, HIGH);
203   digitalWrite(11, LOW);
204   delay(1000);
205   digitalWrite(10, LOW);
206   digitalWrite(11, HIGH);
207   delay(1000);
208   digitalWrite(10, HIGH);
209   digitalWrite(11, LOW);
210   delay(1000);
211   digitalWrite(10, LOW);
212   digitalWrite(11, HIGH);
213   delay(1000);
214   digitalWrite(10, HIGH);
215   digitalWrite(11, LOW);
216   delay(1000);
217   digitalWrite(10, LOW);
218   digitalWrite(11, HIGH);
219   delay(1000);
220   digitalWrite(10, HIGH);
221   digitalWrite(11, LOW);
222   delay(1000);
223   digitalWrite(10, LOW);
224   digitalWrite(11, HIGH);
225   delay(1000);
226   digitalWrite(10, HIGH);
227   digitalWrite(11, LOW);
228   delay(1000);
229   digitalWrite(10, LOW);
230   digitalWrite(11, HIGH);
231   delay(1000);
232   digitalWrite(10, HIGH);
233   digitalWrite(11, LOW);
234   delay(1000);
235   digitalWrite(10, LOW);
236   digitalWrite(11, HIGH);
237   delay(1000);
238   digitalWrite(10, HIGH);
239   digitalWrite(11, LOW);
240   delay(1000);
241   digitalWrite(10, LOW);
242   digitalWrite(11, HIGH);
243   delay(1000);
244   digitalWrite(10, HIGH);
245   digitalWrite(11, LOW);
246   delay(1000);
247   digitalWrite(10, LOW);
248   digitalWrite(11, HIGH);
249   delay(1000);
250   digitalWrite(10, HIGH);
251   digitalWrite(11, LOW);
252   delay(1000);
253   digitalWrite(10, LOW);
254   digitalWrite(11, HIGH);
255   delay(1000);
256   digitalWrite(10, HIGH);
257   digitalWrite(11, LOW);
258   delay(1000);
259   digitalWrite(10, LOW);
260   digitalWrite(11, HIGH);
261   delay(1000);
262   digitalWrite(10, HIGH);
263   digitalWrite(11, LOW);
264   delay(1000);
265   digitalWrite(10, LOW);
266   digitalWrite(11, HIGH);
267   delay(1000);
268   digitalWrite(10, HIGH);
269   digitalWrite(11, LOW);
270   delay(1000);
271   digitalWrite(10, LOW);
272   digitalWrite(11, HIGH);
273   delay(1000);
274   digitalWrite(10, HIGH);
275   digitalWrite(11, LOW);
276   delay(1000);
277   digitalWrite(10, LOW);
278   digitalWrite(11, HIGH);
279   delay(1000);
280   digitalWrite(10, HIGH);
281   digitalWrite(11, LOW);
282   delay(1000);
283   digitalWrite(10, LOW);
284   digitalWrite(11, HIGH);
285   delay(1000);
286   digitalWrite(10, HIGH);
287   digitalWrite(11, LOW);
288   delay(1000);
289   digitalWrite(10, LOW);
290   digitalWrite(11, HIGH);
291   delay(1000);
292   digitalWrite(10, HIGH);
293   digitalWrite(11, LOW);
294   delay(1000);
295   digitalWrite(10, LOW);
296   digitalWrite(11, HIGH);
297   delay(1000);
298   digitalWrite(10, HIGH);
299   digitalWrite(11, LOW);
300   delay(1000);
301   digitalWrite(10, LOW);
302   digitalWrite(11, HIGH);
303   delay(1000);
304   digitalWrite(10, HIGH);
305   digitalWrite(11, LOW);
306   delay(1000);
307   digitalWrite(10, LOW);
308   digitalWrite(11, HIGH);
309   delay(1000);
310   digitalWrite(10, HIGH);
311   digitalWrite(11, LOW);
312   delay(1000);
313   digitalWrite(10, LOW);
314   digitalWrite(11, HIGH);
315   delay(1000);
316   digitalWrite(10, HIGH);
317   digitalWrite(11, LOW);
318   delay(1000);
319   digitalWrite(10, LOW);
320   digitalWrite(11, HIGH);
321   delay(1000);
322   digitalWrite(10, HIGH);
323   digitalWrite(11, LOW);
324   delay(1000);
325   digitalWrite(10, LOW);
326   digitalWrite(11, HIGH);
327   delay(1000);
328   digitalWrite(10, HIGH);
329   digitalWrite(11, LOW);
330   delay(1000);
331   digitalWrite(10, LOW);
332   digitalWrite(11, HIGH);
333   delay(1000);
334   digitalWrite(10, HIGH);
335   digitalWrite(11, LOW);
336   delay(1000);
337   digitalWrite(10, LOW);
338   digitalWrite(11, HIGH);
339   delay(1000);
340   digitalWrite(10, HIGH);
341   digitalWrite(11, LOW);
342   delay(1000);
343   digitalWrite(10, LOW);
344   digitalWrite(11, HIGH);
345   delay(1000);
346   digitalWrite(10, HIGH);
347   digitalWrite(11, LOW);
348   delay(1000);
349   digitalWrite(10, LOW);
350   digitalWrite(11, HIGH);
351   delay(1000);
352   digitalWrite(10, HIGH);
353   digitalWrite(11, LOW);
354   delay(1000);
355   digitalWrite(10, LOW);
356   digitalWrite(11, HIGH);
357   delay(1000);
358   digitalWrite(10, HIGH);
359   digitalWrite(11, LOW);
360   delay(1000);
361   digitalWrite(10, LOW);
362   digitalWrite(11, HIGH);
363   delay(1000);
364   digitalWrite(10, HIGH);
365   digitalWrite(11, LOW);
366   delay(1000);
367   digitalWrite(10, LOW);
368   digitalWrite(11, HIGH);
369   delay(1000);
370   digitalWrite(10, HIGH);
371   digitalWrite(11, LOW);
372   delay(1000);
373   digitalWrite(10, LOW);
374   digitalWrite(11, HIGH);
375   delay(1000);
376   digitalWrite(10, HIGH);
377   digitalWrite(11, LOW);
378   delay(1000);
379   digitalWrite(10, LOW);
380   digitalWrite(11, HIGH);
381   delay(1000);
382   digitalWrite(10, HIGH);
383   digitalWrite(11, LOW);
384   delay(1000);
385   digitalWrite(10, LOW);
386   digitalWrite(11, HIGH);
387   delay(1000);
388   digitalWrite(10, HIGH);
389   digitalWrite(11, LOW);
390   delay(1000);
391   digitalWrite(10, LOW);
392   digitalWrite(11, HIGH);
393   delay(1000);
394   digitalWrite(10, HIGH);
395   digitalWrite(11, LOW);
396   delay(1000);
397   digitalWrite(10, LOW);
398   digitalWrite(11, HIGH);
399   delay(1000);
399 }

```

Fig: Screenshot of C Code on Arduino IDE used for Voice controlled Wheelchair

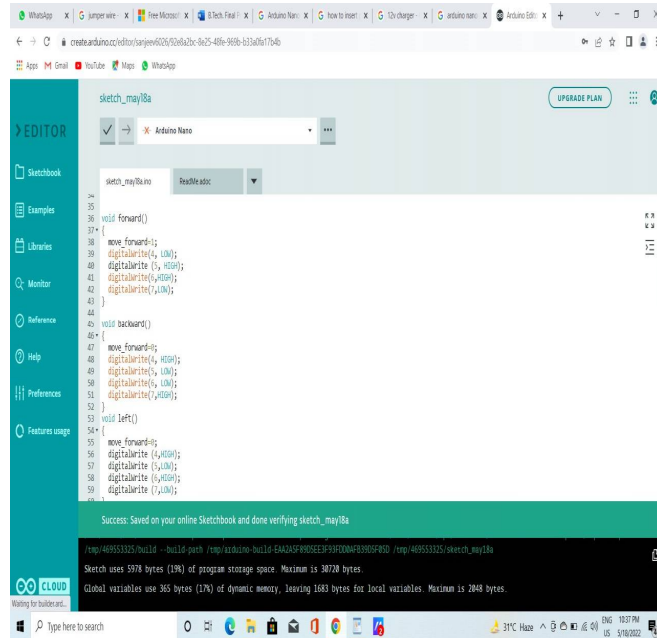


Fig: Screenshot of testing C Code on Arduino IDE used for Voice controlled Wheelchair

IV. SOME PICTURES OF PROJECT

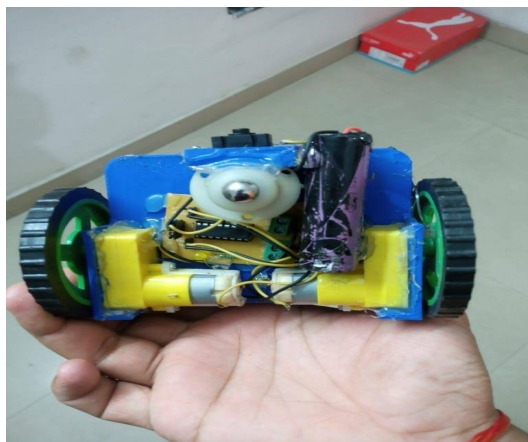
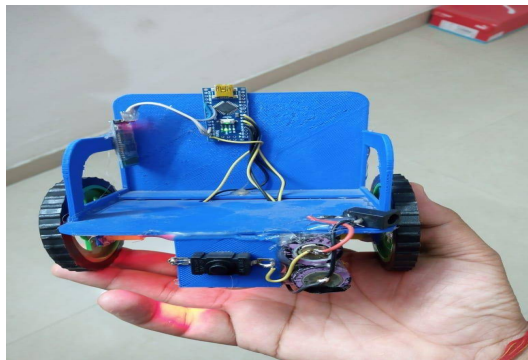


Figure-Project Pictures

V. RESULTS AND DISCUSSION

After a detailed study of various methods used to improve the effectiveness of wheelchair used by crippled people, we move to a wheelchair is fully capable of carrying the load of a person and moving in accordance with the gesture or speech given by the person who is using the wheelchair. Certain improvisation and improvement can be done to make the wheelchair more reachable to those whose whole body is paralyzed. Certain eyes gesture or brain signals reader can be imparted on the wheelchair system to make it better

REFERENCES

- [1] S. D. Suryawanshi, J. S. Chitode, S. S. Pethakar, 2013, Voice Operated Intelligent Wheelchair.”
- [2] Kharka Bahadur Rai, Jeetendra Thakur, Nirmal Rai, Volume No.04, Issue No. 06, June 2015 voice-controlled wheelchair using arduino,”
- [3] Simpson RC, Levine SP, IEEE Trans Neural System Rehabilitation Eng. 2000, 122-125,” Voice control of a powered wheelchair powered wheelchair
- [4] UCHIYAMA Saburo and TEIIMA Noriyuki, "Effective operation of power wheelchair by using speech recognition symem", Proceedings of 75th Annual conference of Japan Society of Mechanical Engineers, pp. 92-93, May. 1998
- [5] S.-Y. Cho, A. Winod and K. Cheng, "Towards a Brain-Computer Interface based control for next generation electric wheelchairs", Power Electronics Systems and Applications 2009. PESA 2009. 3rd International Conference, 2009.
- [6] U. I. F. Qidwai, "Arabic speech-controlled wheelchair: A fuzzy scenario", Information Sciences Signal Processing and their Applications (ISSPA) 2010 10th International Conference, 2010.
- [7] R. Akmeliawati, F. Tis and U. Wani, "Design and development of a hand-glove controlled wheelchair", Mechatronics (ICOM) 2011 4th International Conference, 2011



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)