WHEELCHAIR CONTROL THROUGH VOICE RECOGNITION USING ARDUINO

A MINI PROJECT REPORT

Submitted by

MONIKKA R	**(19CSR061)**
KEERTHANA S J	**(19CSR052)**
KAVEN KUMAR R	**(19CSR046)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE and ENGINEERING

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

ERODE 638 012

MAY 2022

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

ERODE 638012.

BONAFIDE CERTIFICATE

Certified that this project report **"WHEELCHAIR CONTROL THROUGH VOICE RECOGNITION USING ARDUINO"** is the bonafide work of **"MONIKKA.R(19CSR061),KEERTHANA.S.J(19CSR052),KAVEN KUMAR.R(19CSR046)"**who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr.M.Deepa Ph.D,
SUPERVISOR
Associate Professor,
Department of CSE,
Velalar College of Engineering and Technology,
Thindal,Erode-12

SIGNATURE
Dr.S.Jabeen Begum,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
Professor,
Department of CSE,
Velalar College of Engineering and Technology,
Thindal,Erode-12

Submitted for Semester Mini-Project viva-voce examination held on_____

INTERNAL EXAMINER

ACKNOWLEDGEMENT

On the glorious occasion of having consummated our project work, we would like to thank our honorable Secretary and Correspondent **Thiru.S.D.CHANDRASEKAR, B.A**., for providing sample facilities to complete our project.

We express our deep sense of gratitude and sincere thanks to our Principal **Dr. M. JAYARAMAN, B.E, M.E., Ph.D.,** for permitting us to undertake this project. We are indebted to our Dean **Prof.P.JAYACHANDAR, M.E.,** for giving us the opportunity and continuous inspiration to carry out this project.

We express our profound gratitude to our beloved Head of the Department **Dr.S.JABEEN BEGUM, M.E., Ph.D.,** for her valuable guidance and immensely profited by his comments and reviews.

Our sincere thanks our to Supervisor **DR.DEEPA.M, Ph.D,** Associate Professor for valuable guidance, advice and help rendered whenever we approached in times of need.

We express our sincere thanks to our Project Coordinator **Ms.C.KOTTEESWARI,B.E., M.Tech., (Ph.D.,)** Assistant Professor for his guidance to do our project successfully.

We express our sincere thanks to our Parents for their support to do our project successfully.

ABSTRACT

This project describes the design of a smart, motorized, voice-controlled wheelchair using embedded system. The proposed design supports voice activation system for physically differently abled persons incorporating manual operation. This p represents the “Voice-controlled Wheel chair” for the physically differently abled person where the voice command controls the movements of the wheelchair. The voice command is given through a cellular device having Bluetooth and the command is transferred and converted to string by the BT Voice Control for Arduino and is transferred to the Bluetooth Module HC-05 connected to the Arduino board for the control of the Wheelchair.

For example, when the user says, Go then chair will move in forward direction and when he says “Back” then the chair will move in backward direction and similarly “Left”, “Right” for rotating it in left and right directions respectively and “Stop” for making it stop. This system was designed and developed to save cost, time and energy of the patient. Ultrasonic sensor is also made a part of the design and it helps to detect obstacles lying ahead in the way of the wheelchair that can hinder the passage of the wheelchair.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The wheelchair is the most ubiquitous equipment used by people with lower limb disability. It enables them some degree of freedom in mobility and independence as opposed to those with both upper and lower limb disabilities. Most of the wheelchairs available in the market are manual in nature with some available with motorized option. Anything beyond that is custom made which is costly and not within the reach of most people. People with severe lower and upper disabilities have to resort to costly electronic controlled wheelchairs or be totally dependent on another person to move them around in their manual wheelchairs. Motorized wheelchairs controlled through joystick, softball, finger, tablet, chin and head are readily available at a high cost but most of them do not cater for those with upper limb disability. The advances in speech recognition technology have made it possible to control any electronics based device using voice command. This technology is capitalized for voice-controlled wheelchair to assist those with both upper and lower limb disabilities. A variety of voice-controlled wheelchairs have also been developed by other researchers [1-5]. The World Health Organization (WHO) has estimated that of the 75 million people who need assistive technology such as wheelchair, only 5% to 15% of those have access to one.

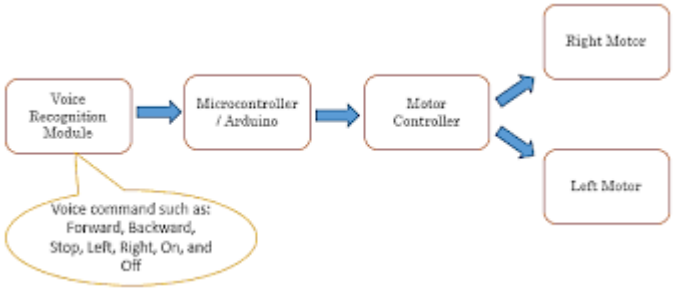


Figure 1.1: Working Model

1.2 OBJECTIVES

1.The main aim of this project is to help disabled people.

2. This is achieved by detecting the voice and moving forward and backward etc...

3. The movement of the wheel chair is controlled by Arduino processor using motor driver.

4. The aim is to improve and it will easy to move by voice controller.

1.3 APPLICATIONS

By designing and implementing a faster voice recognition system and a faster controller on the wheelchair, the voice controlled wheelchair may be safer to use by users. Work is in progress on voice recognition now voice recognition system are also using in vehicles, mobile apps are also introduced that do voice recognition and jets. Future scope of the project includes the ability to control various household devices through voice recognition like TV, air-condition, microwave oven etc.

1.4 TYPES OF DISABILITY

A report on disability by World Health Organization (WHO) states that around 15 percent of the world population are living with some kind of disability. Out of which about 2-4 percent had difficulties in functioning. United Nation Development Program (UNDP) estimated that around 80 percent of the disabled people live in developing countries. In India, the census 2011 which collected data for 8 disabilities, states that 20.5 percent of the disabilities lies in the movement [5]. The restriction in movement due to disability lead to the low self-esteem, stress, isolation, fear of abandoning, etc. The purpose of the proposed wheel chair is providing the multi-control operated wheel chair at a lower cost.

1.5 FEATURES

Features and System Specifications of Smart Wheel Chair The features of smart wheel chair are following: -

- ✓ Voice controlled automatic wheelchair.
- ✓ Smooth speed variations for patient comfort through Arduino Controlled DC Motor
- ✓ Facility to move in definite path

- ✓ Line follower to follow definite path.
- ✓ Soft start & Soft Stop
- ✓ It is also controlled by manual.
- ✓ Collision avoidance

CHAPTER 2
LITERATURE SURVEY

The idea of the project titled **"WHEEL CHAIR CONTROL THROUGH VOICE RECOGINITION USING ARDUINO"** gathered by analyzing many papers and reports. Below is the brief description of those papers and reports that have analyzed so far.

People with arms and hand impairment finds difficult to use a normal wheelchair as their hands are not capable of operating the normal wheelchair and cannot move it to any direction. Therefore, voice-controlled wheel chair is built to overcome the problems faced by such people and enable them to operate the wheelchair. The wheelchair will be operated using the voice commands through the given input. The Arduino will take care about all the directions the user wants. The instruction for each and every direction is written in the form of program in the Arduino itself.

Pramila Kupkar, Prajakta Pandit, Nikita Dhadhere and PP Jadhav, "Android controlled wheelchair", Imperial Journal of Interdisciplinary Research (IJIR) Volume-2 Issue-6 2016 stated that the people have disabilities with their hands, foot and lower extremities because of which they are unable to perform regular tasks. Many technologies are available to overcome this problem. To overcome this problem, there are several applications in the market which help handicapped people to perform their tasks.

Apsana S, Renjitha G Nair, "Voice Controlled Wheelchair using Arduino", International Advanced `Research Journal in Science, Engineering and Technology (IARJSET), Vol 3, Issue 3, August 2016 elaborated that one major challenge for these people, and also for a large number of amputees is locomotion, and it is impossible for these people to move around without assistance. However, many of those suffering from complete paralysis usually retain control of their eye movement, and this inspired us to develop an eye controlled electric wheel chair. Through the use of an eye tracker, different forms of wheelchair control are possible. This paper presents the de-sign and development of an eye-controlled wheelchair, an affordable and non-intrusive technique, that only uses the eye images captured by a small camera attached to the wheelchair.

Mr. Tarun Agrawal, "Review on Voice Recognition Module Working", International Journal of Advanced Research in Computer Science and Software Engineering, May 2014 proposed

a system which aids an assistance for physically handicapped ones those who are not able to move by themselves. It uses speech recognition by interfacing speech recognition kit (HM2007) with microcontroller and wheelchair. The system provides a Mic for the user to give commands HM2007 registers the commands and fed them to microcontroller. Motor driver drives the wheel chair according to the commands from microcontroller. Wheel Chair for Physically Disabled People with Voice &Eye control.

Ms S. D. Suryawanshi Mr. J. S. Chitode Ms. S. S. Pethakar, "Voice Operated Intelligent Wheelchair", 2013 introduced a concept useful for people with loco-motor disability. Here wheel chair is controlled by eye movement and voice commands. Eye movement is detected by using a head mounted camera. Corresponding output signals fed to motor which control wheel chair movement. Voice assistance is also used by this system. By giving commands the movement can be controlled. Voice Controlled Wheel chair system.

Kharka Bahadur Rai, Jeetendra Thakur, Nirmal Rai. Voice Controlled Wheel Chair Using Arduino, International Journal of Science, Technology & Management, June 2015 designed a system that operates on user's voice commands. This voice-controlled wheel chair helps them to drives the wheel chair without any one's help. This system can be controlled by user's simple voice commands. According to the direction specified in commands, wheel chair moves. Speech recognition is done here by using a speech recognition module. Design and Construction of Electric Drive –A smart system for disabled person with therapy

CHAPTER 3
EXISTING SYSTEM

3.1 INTRODUCTION

The Wheelchair is a vulnerable system used by aged and physically disabled people. By considering their requirements there are already many such systems in the market yet this our work is quite simpler and easy to use. Those complex control systems may be difficult to operate for ill persons. Hence eye gaze wheelchair provides the freedom to make their life easy and more convenient. This Eye gaze wheelchair does not require much energy and manpower to operate. The Camera captures the picture of an eye pupil of the patient on the wheelchair and sends these as input command for the DC motor which in turns moves the wheelchair accordingly through GPIO pins. The DC motor is used to perform the different operations such as to move the wheelchair left, right, forward, backward and stop. The advance level of Image Processing open computer vision library is used for Face and Eye detection. In general, there are many algorithms and applications for detecting these eye and face detection.

3.2 PROBLEM STATEMENT

A clinical survey indicates that 9-10% of severely disabled patients having difficulties find it impossible in using powered wheelchair in spite of having some training in handling and operating the wheelchair. This indicates that they are lacked of motor skill and strength and difficult to operate a sophisticated wheelchair function. Our system aims at user friendly product which requires no rigorous training.

3.3 METHODS AND SOFTWARE

Current input devices such as joystick, keyboard, mouse and other devices which are used for interaction with the automated devices require various hand movements. These input devices cannot be used by people suffering from physical disabilities. The existing devices and their disadvantages for handicap person are as categorized:

(1) Bio-potential based method which utilizes potential from user's body actions acquired by using special instrument. Instrument such as Electrooculography (EOG), Electromyography (EMG) , and

Electroencephalograph (EEG) ,Search coil can be used for measuring bio-potential. The search coil output can be used as sources of computer input for handicap person. Limitations of this method are poor gaze direction accuracy compared to video tracker, relatively costly.

(2) Voice Based method, which use user's voice as source input. Voice analysis is used to analyze user's voice and convert into digital data. This type of system is vulnerable against noise. Other voices which come from surrounding user may affect the system. Limitations of this method are it is less accurate on background noise, speaker variability, channel variability, speaking style, speed of speech, etc.

(3) Motion based method, utilizes other normal movement organs to operate computer input. Head, foot, and etc. can be used to control computer input. Limitation of this method is that it requires human effort to navigate like joystick, etc. Also, for a handicapped with any of the organs failed is not useful.

(4) Search coil method uses induced voltage with coil including in contact lenses attached to user 's eyes. Limitation of this method is burden to user, here measuring time is limited to approximately 30 to 60. They have limited Lifetime.

(5) Image Analysis method, utilizes camera to analyze user's desire and convert into digital data. Several image processing methods are used to analyze user's desire. The user's desire itself can be done by Gaze based analyze user's desire from user's gaze, Face based analyze user's desire from face expression, and the others. This method is costly.

3.3.1 Hardware

(1) USB cable: In computing, a USB cable is a serial communication physical interface through which information transfers in or out one bit at a time

(2) Arduino: Arduino is a simple microcontroller board and open-source development environment that allows you to make computers that drive both functional and creative projects a like.

(3) L298D allows DC motor to drive on either direction.

3.4 BASIC DEVICES

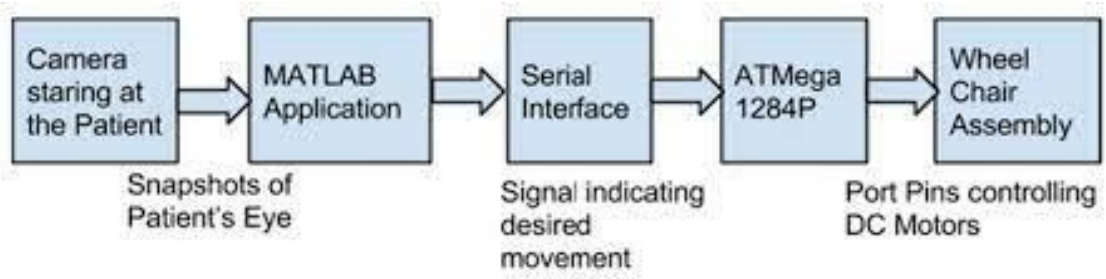


Figure 3.4 Schematic block diagram

3.4.1 WIRELESS CAMERA

Eye of the user is captured with a pin hole wireless camera which transmits the images to the Base station wirelessly.

Figure 3.4.1: Wireless camera

3.4.2 MICROCONTROLLER

- The microcontroller used in this model is Arduino. Arduino is a single-board microcontroller, intended to make building interactive object

- The system uses two microcontrollers. The Transmitting Microcontroller is connected to the processing unit.

- This microcontroller converts the information received from the processing unit into signals and transmits them wirelessly over to the receiving microcontroller attached to the wheelchair.

- The Receiving Microcontroller receives signals from the transmitting microcontroller wirelessly and accordingly initiates the movement in the required direction. This microcontroller is mounted on the wheelchair and is connected to the motor driver.

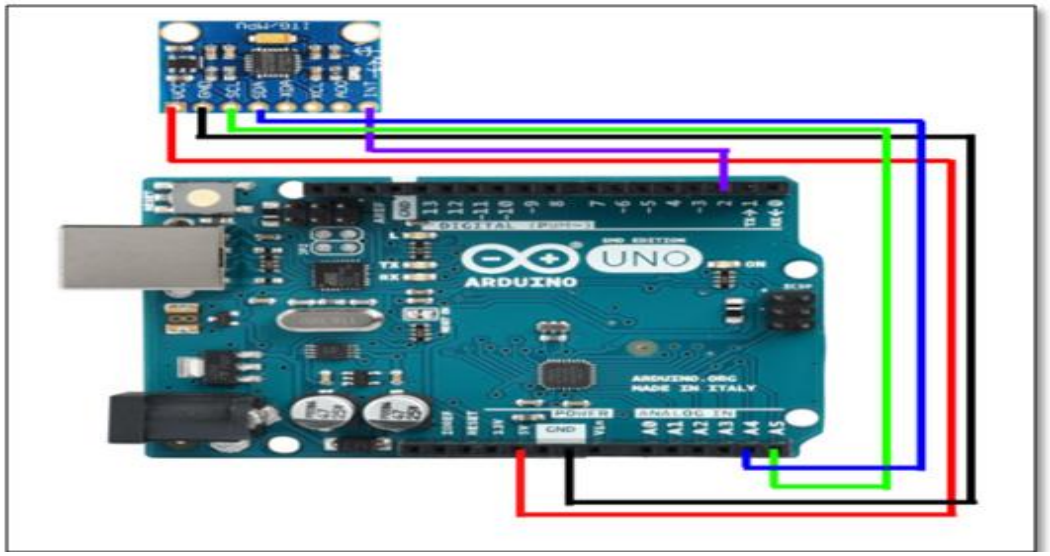


Figure 3.4.2: Microcontroller

3.4.3 OBSTACLE SENSORS

The wheelchair has been mounted with four ultrasonic sensors to avoid collision and damage to the user. The three sense monitor the forward, left and right directions. Ultrasonic sensors use electrical mechanical energy transformation to measure distance from the sensor to the target object.

Figure 3.4.3: Obstacle Sensor

3.4.4 MOTOR DRIVER

They provide the high current required to drive the motors. It is controlled by the Arduino. Motor drivers acts as an interface between the motors and the control circuits. Motor requires high amount of current whereas the controller circuit works on low current signals. So, the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.

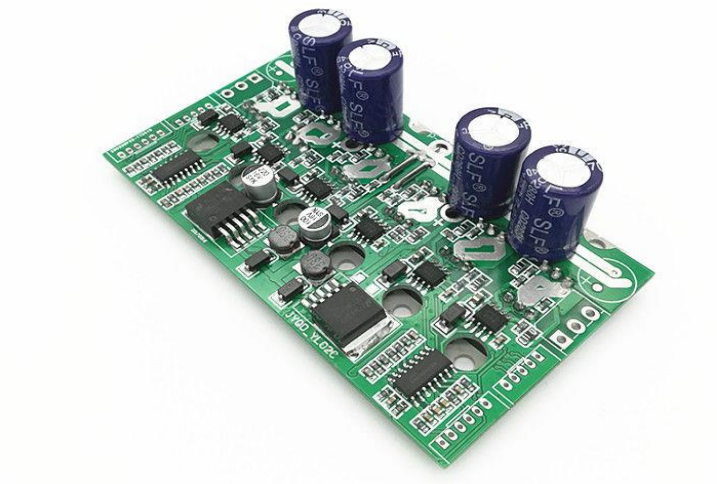


Figure 3.4.4: Motor Driver

CHAPTER 4

PROPOSED SYSTEM

4.1 OVERVIEW

Voice controlled wheelchair prototype was developed using a commercially available manual wheelchair to assist people with both upper and lower limb disabilities. An Arduino microcontroller processes the voice command from the speech recognition module and controls the motor movement of the wheelchair. Bluetooth module was also used to do away with messy wiring and an optional joystick command was also incorporated into the prototype design.

4.2 PROPOSED METHOD

In this project we are using Android Application and Voice Recognition System. But many of individuals with disabilities who need wheelchairs are satisfied with it, few members of the disabled community find it is difficult or impossible for operating a standard power wheelchair. This project is included in assistive technology. For handicapped and depended disable it is more independent, productive and enjoyable living.

There are four basic movements of a wheelchair to be applied by the user. The Five operations perform by the wheelchair are described as following:

1) Moving forward
2) Moving backward
3) Turning to the right
4) Turning to the left

4.3 RELATED WORKS

Several studies have concluded that the independent mobility or movement which is included powered wheel chair, manual wheelchair and walker access the benefit to all the disabled human beings, independent mobility increases vocational and educational opportunities, reduces dependence on other members, and promotes feelings of self-reliance and in dependability. Independent mobility plays a vital role in building the foundation for much early learning for young people. The lack of exploration and control often results into a

cycle of deprivation and lack of motivation that leads to learned helplessness. For aged people, independent movement is an important aspect of self-esteem and plays a vital role in "aging in place." Mobility difficulties led to the problem of activities of daily living (ADL) and instrumental ADL disabilities because of the need to move to accomplish many of these activities. The impaired mobility often results in reduced opportunities to have socialized policies, which leads to social isolation, and many mental problems. While the needs of many individuals with disabilities can be satisfied with traditional manual or self-automated wheelchairs, a segment of the disabled community finds it difficult or impossible to use wheelchairs independently.[5] The disabled population includes people with low vision, visual field reduction, spasticity, tremors, or cognitive deficits. These individuals' dependent on other people for mobility to push them in a manually handled wheelchair. To accommodate this population, several researchers have used technologies originally developed for Power wheelchairs have been designed of different ways, such as assuring collision free travel, aiding the performance of specific tasks (e.g., passing through doorways), and autonomously transporting the user between locations.

4.4 IMPLEMENTATION

4.4.1 IMPLEMENTATION PLATFORM

Hardware Requirement

- Arduino UNO.
- Motor driver
- Bluetooth Module HC-05.
- Wheel chair chassis.
- Battery 12 volts.
- 2 DC Motors (12 V 200 rpm).
- Android phone.
- Power supply.

Software Requirement

- Arduino IDE.
- Android Application.
- Coding language: Arduino programming (C).

4.5 IMPLEMENTATION DETAILS

Currently the prototype of the project has been developed.

The figure below is

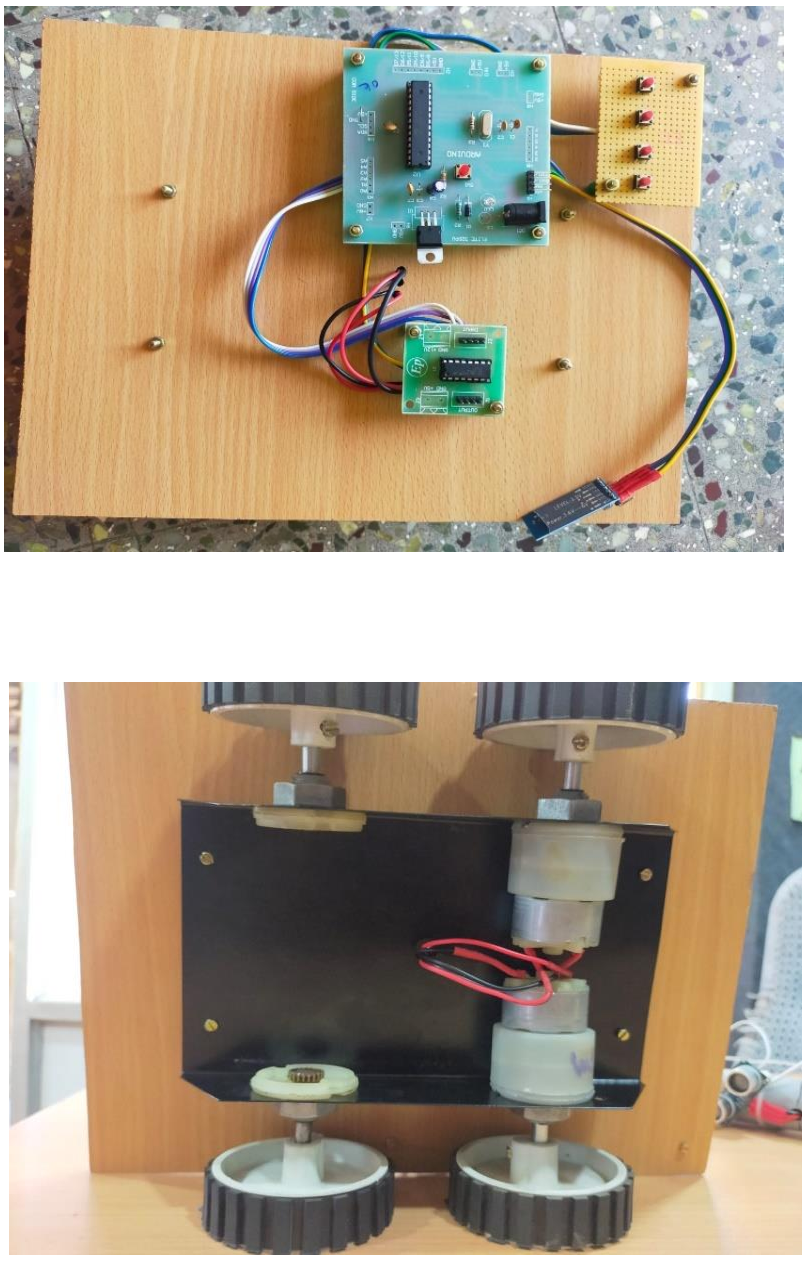


Figure 4.5 Implementation

4.6 FLOWCHART

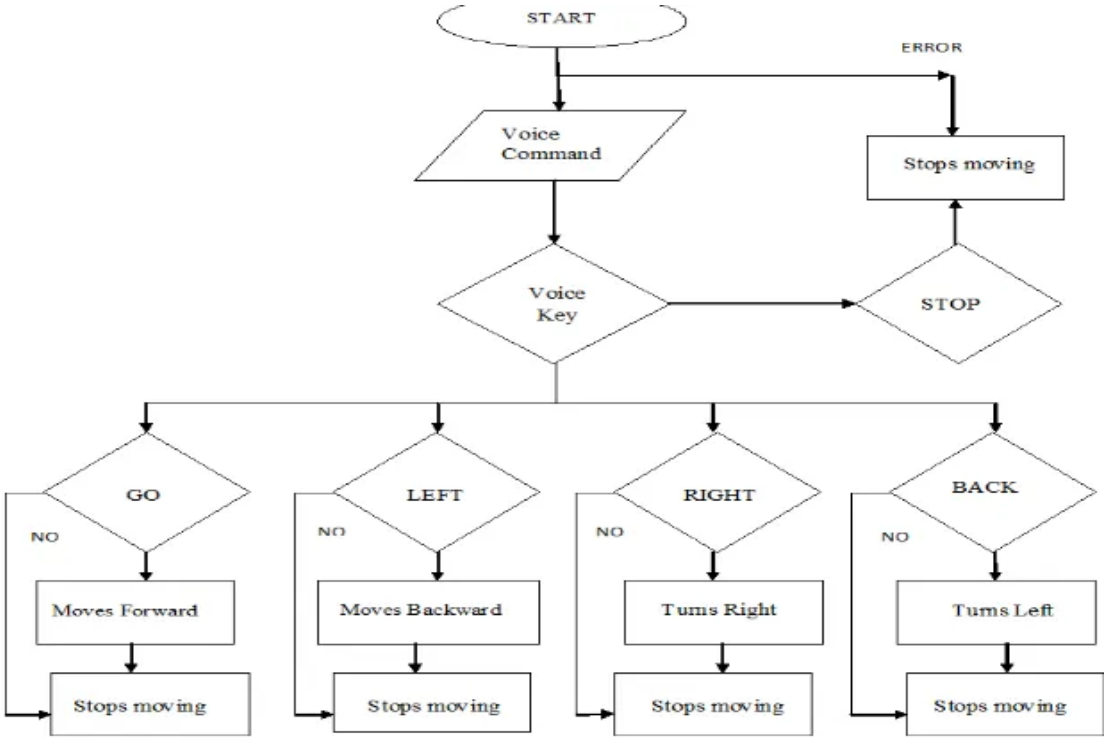


Figure 4.6 Flowchart

4.7 ADVANTAGES

This project describes the design and development of the motion control using voice recognition and graphical Android App for a wheelchair application. Objectives of Project:

- To develop a voice-controlled wheelchair system by using C for wheelchair control.
- To design and develop an android app through which we can operate a movement of wheel chair Using android device.
- To implement and use the voice-based system so that users voice as an input to control the mobility of wheelchair.
- Helps to implement movement for disabled people and aged people who can't move properly.
- Easy to drive with negligible efforts.
- Less Complexity and Hardware to mount.
- Can be mounted on the existing wheelchair.

- Wireless control helps to monitor the wheelchair.
- Reduces manpower and dependency on other human drive.
- Wheelchair is compact and economical.
- Provides easy movement for physically challenged people.
- Low power consuming and easy to operate the wheelchair.

4.7.1 Future Scope of project

- Voice recognition module is used to develop the voice recognition system. Voice recognition issues a Command to control the movement of wheelchair. For movement of wheelchair Microcontroller Atmega328 and DC motor circuit were built. For not to occur disorder during recognize the user voice, this system works in a quiet environment. Furthermore, the pronunciations accuracy must be ensured and the word-related (voice) the users voice must clear in short distance on microphone was essential in this innovation.
- Using gear box, we can produce high-speed moving wheelchair.
- PWM modulation can also increase speed.
- Solar Panel can also be used to charge the battery for power supply to the components required to drive the wheelchair.
- The wheelchair can also include the gesture feature to operate the wheelchair.
- Wheelchair only can function properly when the weight of the load for this system must be below 50 kilograms.

CHAPTER 5

SYSTEM DESCRIPTION

5.1 HARDWARE DESCRIPTION

5.1.1 ARDUINO UNO

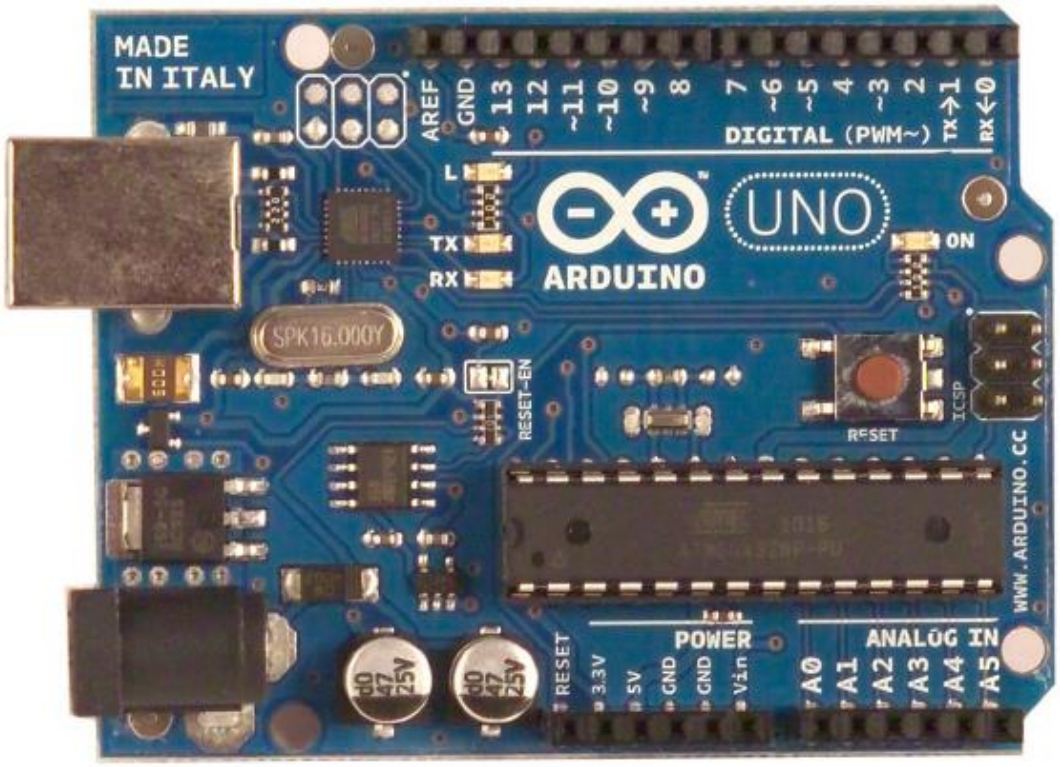


Figure 5.1.1:Arduino UNO Board

Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

Arduino boards can be purchased pre-assembled or as do-it-yourself kits. Hardware design information

is available for those who would like to assemble an Arduino by hand. It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced.

5.1.1.1 HISTROY

Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. At that time program students used a "BASIC Stamp" at a cost of $100, considered expensive for students. Massimo Banzi, one of the founders, taught at Ivrea.[2]

A hardware thesis was contributed for a wiring design by Colombian student Hernando Barragan. After the wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open-source community. The school eventually closed down, so these researchers, one of them David Cuartilla's, promoted the idea.[2]

A 3rd-party Arduino board with a RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are located at the top and the six analog input pins at the lower right.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles. Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the Lilypad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-

232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDI FT232. Some variants, such as the Arduino Mini and the unofficial Bo Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

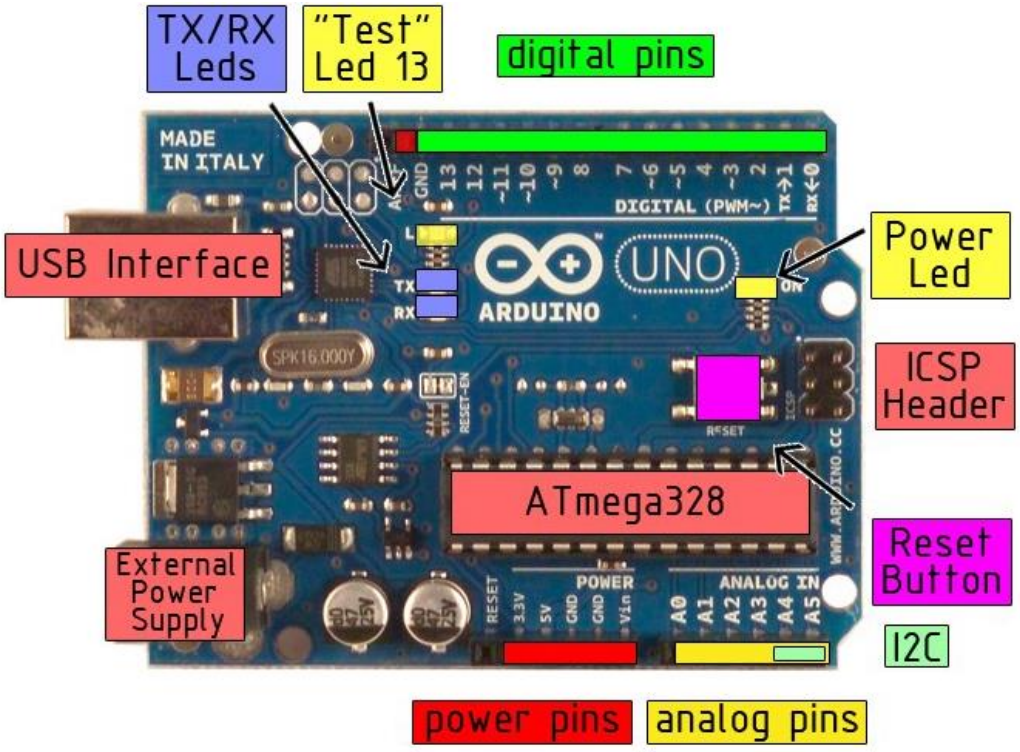


Figure 5.1.1.1: Arduino Uno board Description

5.1.1.2 POWER

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC to- DC adapter (wall -wart) or battery. The adapter can be connected by plugging a 2.1mm center -positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

THE POWER PINS ARE AS FOLLOWS:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7-12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.

3V3.A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins. The Arduino Nano, and Arduino-compatible Bare Bones Board and Bo Arduino boards may provide male header pins on the underside of the board to be plugged into solderless breadboards.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and may be used interchangeably. Many are the basic Arduino with the addition of common place output drivers, often for use in school-level education to simplify the construction of buggies and small robots. Others are electrically equivalent but change the form factor, sometimes permitting the continued use of Shields, sometimes not. Some variants use completely different processors, with varying levels.

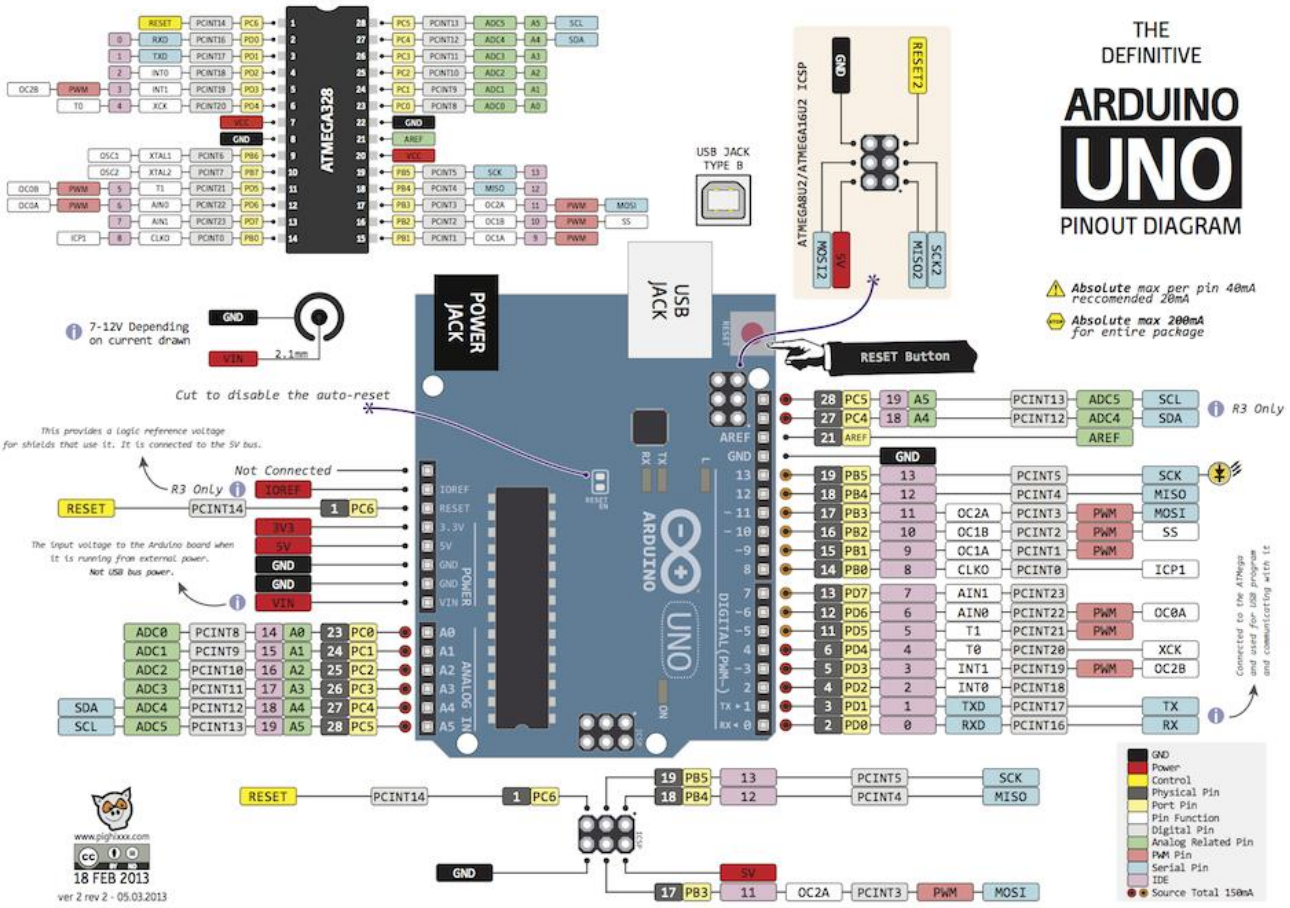


Figure 5.1.1.2: Arduino Uno Pin Diagram

5.1.1.3 MEMORY

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM

5.1.1.4 INPUT AND OUTPUT PINS

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e.1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library

.

There are a couple of other pins on the board:

- AREF Reference voltage for the analog inputs. Used with analog Reference
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

5.1.1.5 COMMUNICATION

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being

transmitted via the USB to serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus;
For SPI communication, use the SPI library

.

5.1.2 L293D - H BRIDGE MOTOR DRIVER

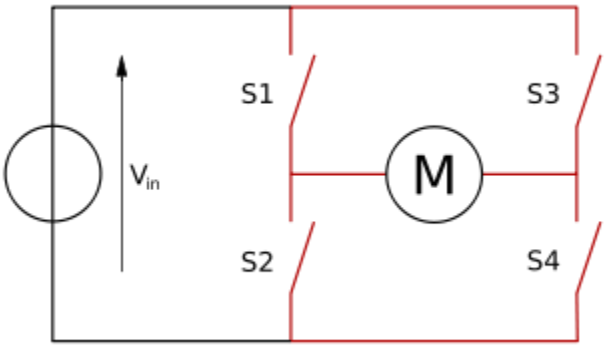


Figure 5.1.2: Motor Driver Diagram

An H bridge is an electronic circuit that enables a voltage to be applied across a load in either direction. These circuits are often used in robotics and other applications to allow DC motors to run forwards and backwards. H bridges are available as integrated circuits, or can be built from discrete components.

5.1.2.1 GENERAL

The term H bridge is derived from the typical graphical representation of such a circuit. An H bridge is built with four switches (solid-state or mechanical). When the switches S1 and S4 (according to the first figure) are closed (and S2 and S3 are open) a positive voltage will be applied across the motor. By opening S1 and S4 switches and closing S2 and S3 switches, this voltage is reversed, allowing reverse

operation of the motor. Using the nomenclature above, the switches S1 and S2 should never be closed at the same time, as this would cause a short circuit on the input voltage source. The same applies to the switches S3 and S4. This condition is known as shoot-through.

5.1.2.2 OPERATION

The H-bridge arrangement is generally used to reverse the polarity of the motor, but can also be used to 'brake' the motor, where the motor comes to a sudden stop, as the motor's terminals are shorted, or to let the motor 'free run' to a stop, as the motor is effectively disconnected from the circuit. The following table summarizes operation, with S1-S4 corresponding to the diagram above.

S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor free runs
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Shoot-through
0	0	1	1	Shoot-through
1	1	1	1	Shoot-through

Table 5.1.2.2: Operation L293D - H Bridge Motor Driver

5.1.2.3 CONSTRUCTION

A solid-state H bridge is typically constructed using opposite polarity devices, such as PNP BJTs or P-channel MOSFETs connected to the high voltage bus and NPN BJTs or N-channel MOSFETs connected to the low voltage bus. The most efficient MOSFET designs use N-channel MOSFETs on both the high side and low side because they typically have a third of the ON resistance of P-channel MOSFETs. This requires a more complex design since the gates of the high side MOSFETs must be driven positive with respect to the DC supply rail. However, many integrated circuit MOSFET drivers include a charge pump within the device to achieve this.

Alternatively, a switched-mode DC–DC converter can be used to provide isolated ('floating') supplies to the gate drive circuitry. A multiple-output flyback converter is well-suited to this application. Another method for driving MOSFET-bridges is the use of a specialised transformer known as a GDT (Gate Drive Transformer), which gives the isolated outputs for driving the upper FETs gates. The transformer core is usually a ferrite toroid, with 1:1 or 4:9 winding ratio. However, this method can only be used with high frequency signals. The design of the transformer is also very important, as the leakage inductance should be minimized, or cross conduction may occur. The outputs of the transformer also need to be usually clamped by Zener diodes, because high voltage spikes could destroy the MOSFET gates.

A common variation of this circuit uses just the two transistors on one side of the load, similar to a class AB amplifier. Such a configuration is called a "half bridge". The half bridge is used in some switched-mode power supplies that use synchronous rectifiers and in switching amplifiers. The half-H bridge type is commonly abbreviated to "Half-H" to distinguish it from full ("Full-H") H bridges. Another common variation, adding a third 'leg' to the bridge, creates a three-phase inverter. The three-phase inverter is the core of any AC motor drive. A further variation is the half-controlled bridge, where one of the high- and low-side switching devices (on opposite sides of the bridge) are replaced with diodes. This eliminates the shoot-through failure mode, and is commonly used to drive variable/switched reluctance machines and actuators where bi-directional current flow is not required.

A "double pole double throw" relay can generally achieve the same electrical functionality as an H bridge (considering the usual function of the device). An H bridge would be preferable to the relay where a smaller physical size, high speed switching, or low driving voltage is needed, or where the

wearing out of mechanical parts is undesirable. There are many commercially available inexpensive single and dual H-bridge packages, and L293x series are the most common ones. Few packages have built-in flyback diodes for back EMF protection.

5.1.2.4 OPERATION AS AN INVERTOR

A common use of the H bridge is an inverter. The arrangement is sometimes known as a single-phase bridge inverter. The H bridge with a DC supply will generate a square wave voltage waveform across the load. For a purely inductive load, the current waveform would be a triangle wave, with its peak depending on the inductance, switching frequency, and input voltage.

5.1.3 HC-05 BLUETOOTH MODULE WITH ARDUINO

Bluetooth is a technology for wireless communication. It is designed to replace cable connections. Usually, it connects small devices like mobile phones, PDAs and TVs using a short-range wireless connection. And it uses the 2.45Ghz frequency band. The connection can be point-to-point or multipoint where the maximum range is 10 meters. The transfer rate of the data is 1Mbps (or a maximum of 2Mbps).

Figure 5.1.3: Bluetooth Logo

5.1.3.1 INTERFACING BLUETOOTH WITH ARDUINO

We will create an Arduino-model where we turn on a LED light using Bluetooth commands. And we will that instruction using our Android Phone. In short, we will control our LED using a phone.
And via Bluetooth we will the ON/OFF instruction.

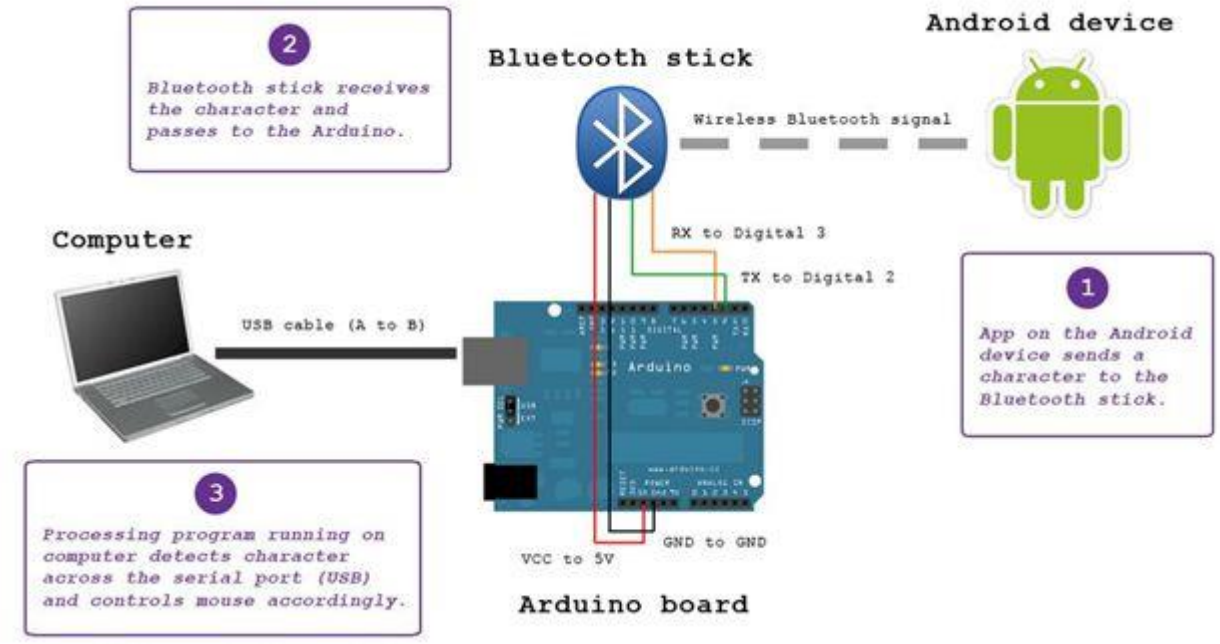


Figure 5.1.3.1: Bluetooth stick

So, what we need:

- Arduino Uno Board
- Bluetooth Module (HC-05)
- 1x 3.5mm LED
- 1k Ohm Resistor
- Jumper Wires

Most of the pre-requisites we have used already in my past articles, except for the Bluetooth Module HC-05.

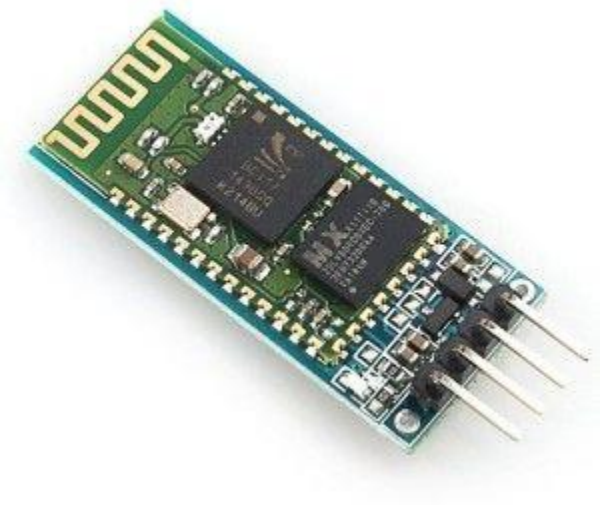


Figure 5.1.3.1: Bluetooth Module HC 05

It is a class-2 Bluetooth module with Serial Profile that can be used as Master or Slave. In Arduino, we will use Serial Communication for various purposes. Since it also supports Serial Communication and you can treat it as a replacement.

5.1.3.2 HC-05 SPECIFICATIONS

- 2.45Ghz **Frequency**
- Asynchronous **Speed** 2.1Mbps (max) .160Kbps
- **Security:** Authentication
- **Profile:** Bluetooth Serial Port
- **Power Supply:** +3.3 VDc
- **Working Temperature:** >20C
- **Cost:** Around INR 300

5.1.3.3 HC-05 DESCRIPTION

So, we have six (or four) leads in this module. But we will genuinely care about only four of them. Where, the two are for Vcc and GND.

Vcc= Power Supply (in other words 5V or 3.3V)

GND= Ground (in other words 0 volt)

And the next two leads are for RX (Receiving End) and TX (Transmitting End). From the basic idea, we can say the RX of the module will go to the TX of the Arduino UNO. In the same way, we connect the TX of the module with the RX of the Arduino UNO.

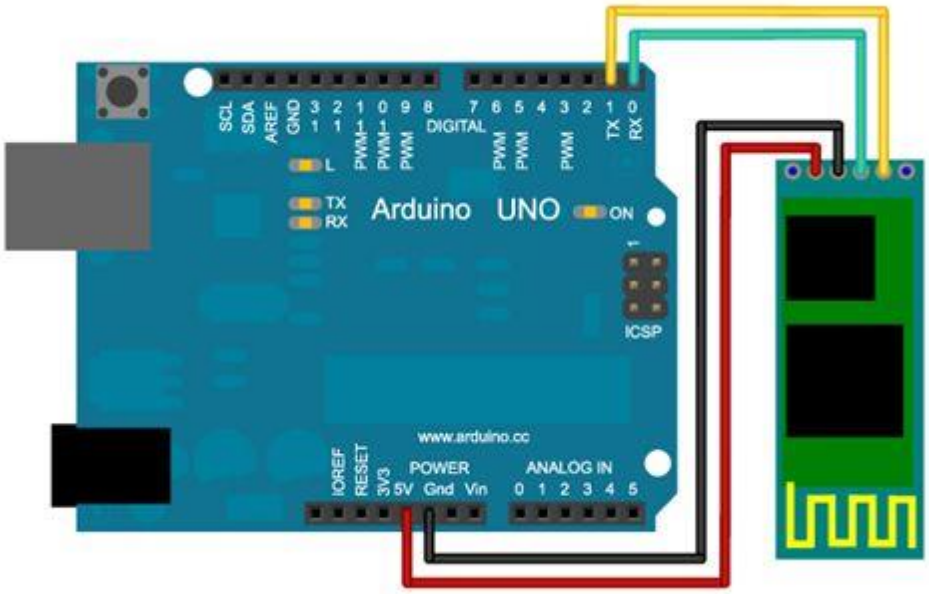


Figure 5.1.3.3: Arduino UNO

5.1.4 WORKING OF DC MOTOR

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

Let's start by looking at a simple 2-pole DC electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).

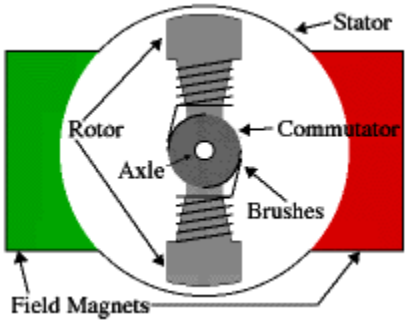


Figure 5.1.4: Internal View of DC Motor

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that Beamers will see), the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotors (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

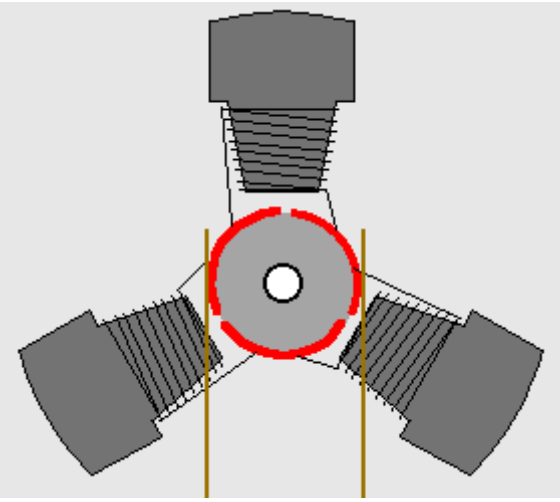


Figure 5.1.4: Coil Arrangements in DC Motor

A D.C. Motor is a machine which converts electrical energy into mechanical energy. Its location

is based on the principal that when a current carrying conductor is placed in the magnetic field, it experiences a mechanical force whose direction is given by Fleming's left-hand rule.

5.1.4.1 CONSTRUCTION

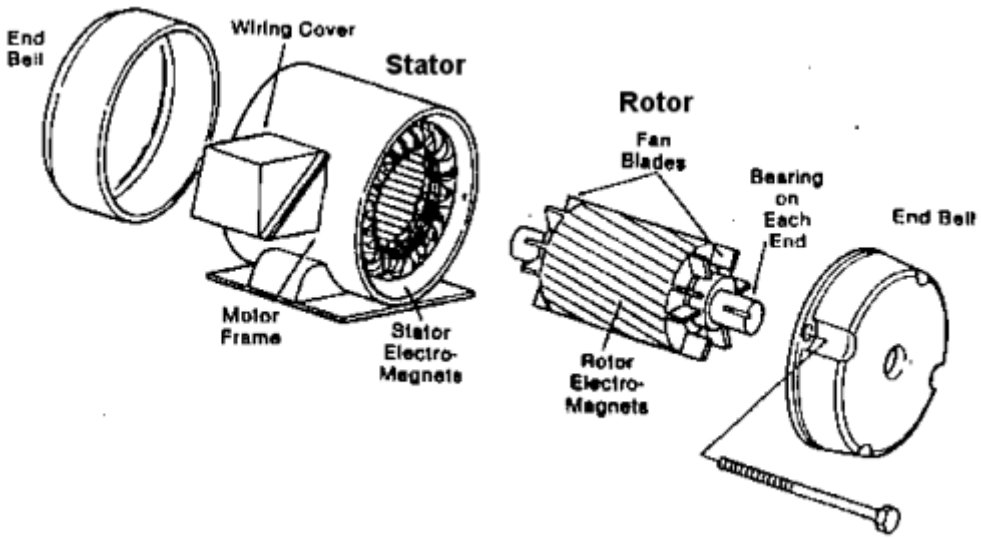


Figure 5.1.4.1: Construction of DC Motor

When its field magnets are excited & its armature conductors are supplied with current from the supply mains, they experience a force tending to rotate the armature. Armature conductors under N-pole are assumed to carry current downwards (plus) & those under S-pole are assumed to carry current upwards (minus). By applying Fleming's left – hand rule, the direction of the force on each conductor can be found. It is shown by small arrow placed above each conductor. It will be such that each conductor experiences a force which tends to rotate the armature in anti-clockwise direction. These forces collectively produce a driving torque which sets the armature rotating.

5.1.4.2 USES OR APPLICATIONS OF D.C MOTOR

- It is used whose high starting torque is required such as for driving hoists, cranes, trains etc.

- It is used where the motor can be directly coupled to load such as fan, where torque increases with speed.
- It is used where the speed has to be maintained approximately constant i.e. From no load to full load.
- It can be used to drive load at various speeds.

5.1.5 KEYPAD

The above decoder function can be implemented in software. The keypad could be interfaced with the 8051 as detailed below.

With the above configuration, an interrupt is generated on the INT0-bar line when a key is pressed. We will deal with how this works in a moment. Firstly, let's see how the keyboard is scanned.

The steps are:

- Scan row 1
- Scan row 2
- Scan row 3
- Scan row 4

Scanning a row is achieved by applying 0 V to the port pin for that row and 5 V to the other three rows, then scanning each individual column to see if one of them is LOW. If it is, then the key at the junction between the current row and column being scanned is the pressed key.

- Clear row 1, set other 3
- Scan column 1
- Scan column 2
- Scan column 3
- Scan column 4
- Clear row 2, set other 3
- Scan column 1
- Scan column 2

- Scan column 3
- Scan column 4
- Clear row 3, set other 3
- Scan column 1
- Scan column 2
- Scan column 3
- Scan column 4
- Clear row 4, set other 3
- Scan column 1
- Scan column 2
- Scan column 3
- Scan column 4

For example, let's say the key being pressed is key 6. When scanning the first row, P1.0 will be cleared while the other 3 rows (P1.1, P1.2 and P1.3) are set, as detailed in the diagram below.

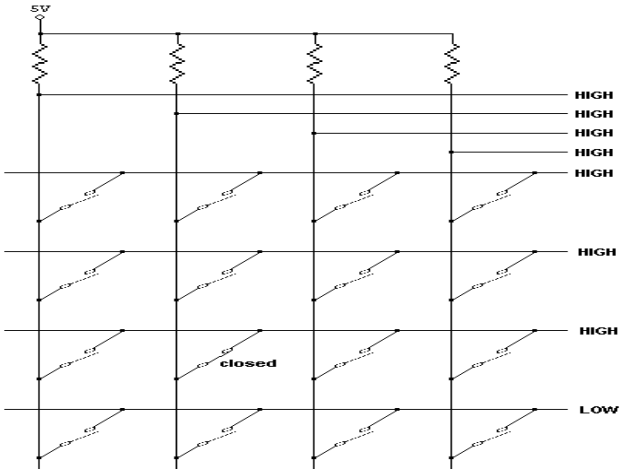


Figure 5.1.5: Key Pressing

Since no key in this row is closed there is no path for current through any of the pull-up resistors and all 4 columns (on P1.4 to P1.7) are HIGH. Therefore, the key pressed was not found while scanning row 1 .The diagram below illustrates scanning row 2.

The keypad is initialized in the main program; all rows are cleared. Therefore, when a key is pressed, since all rows are LOW, then one of the columns (the one containing the key that has been pressed) will be connected to 0 V. This logic 0 into the AND gate will result in a logic 0 out. Since the output of the AND gate is connected to INT0-bar, a key press will result in an external 0 interrupt.

If the first column is not 0, R0 is incremented (it now holds 1) and the next column is tested. If it is 0 then key 1 was pressed and this value is in R0 - therefore the key Found flag is set and the subroutine returns.

This is repeated for all columns until the key is found.

The external 0 interrupt flag (IE0) is then cleared because it may have been set as the key bounces and as we scanned the keyboard (remember, disabling an interrupt does not prevent the interrupt from occurring - a 0 on INT0 will still set the flag, IE0). If we did not clear this flag, once we again enable the external 0 interrupt the system would again vector to key Pressed ISR.

5.2 SOFTWARE REQUIREMENTS

5.2.1 ARDUINO IDE

- **Arduino IDE** is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.
- It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
- It is available for all operating systems i.e.MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code.
- A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
- Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
- The main code, also known as a sketch, created on the IDE platform will ultimately generate a

Hex File which is then transferred and uploaded in the controller on the board.

- The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
- This environment supports both C and C++ languages.

5.2.1.1 HOW TO DOWNLOAD ARDUINO IDE

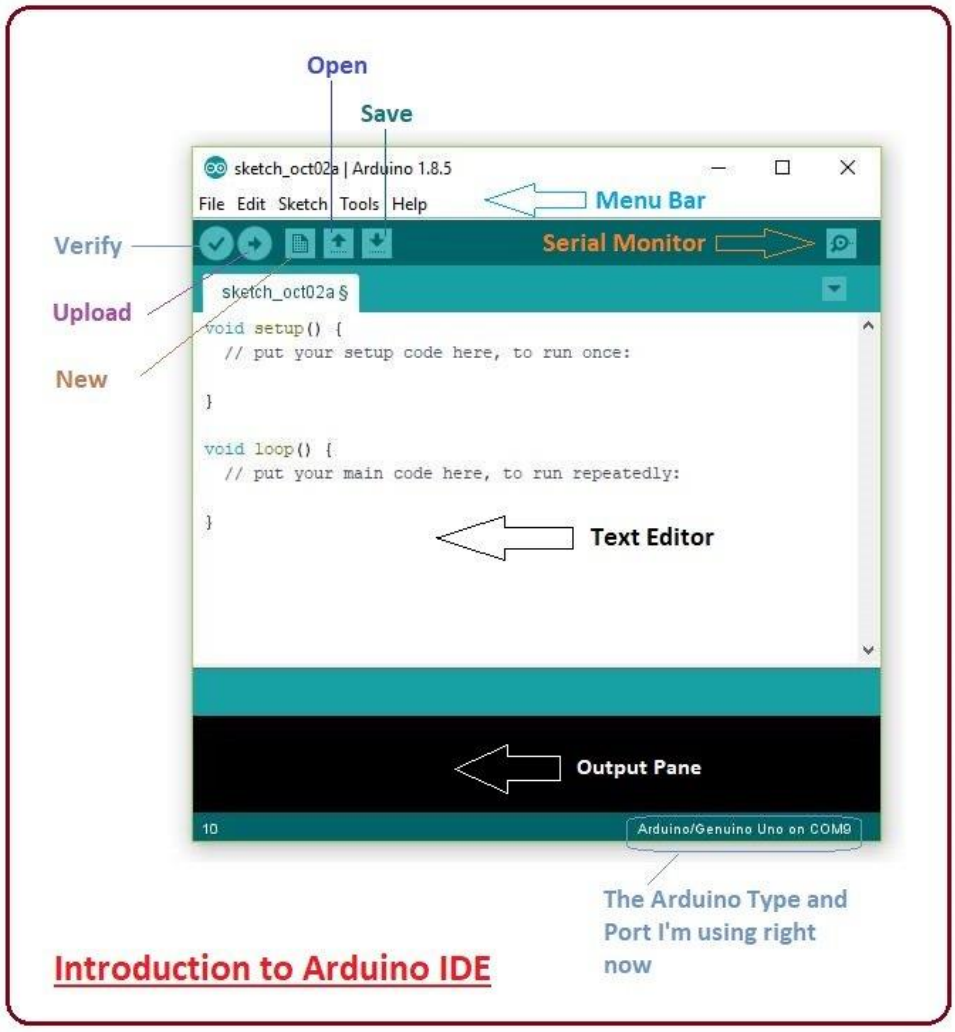
You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

- If you aim to download the Windows app version, make sure you have Windows 8.1 or Windows 10, as the app version is not compatible with Windows 7 or older version of this operating system.

The IDE environment is mainly distributed into three sections

1. **Menu Bar**
2. **Text Editor**
3. **Output Pane**

As you download and open the IDE software, it will appear like an image below:



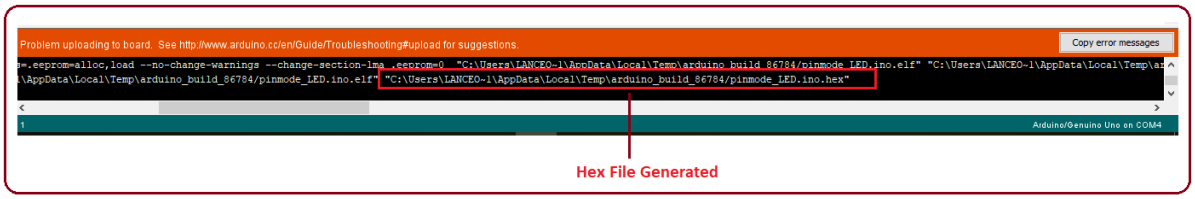
The bar appearing on the top is called **Menu Bar** that comes with five different options as follow:

- **File** – You can open a new window for writing the code or open an existing one. The following table shows the number of further subdivisions the file option is categorized into.

File	
New	This is used to open new text editor window to write your code
Open	Used for opening the existing written code
Open Recent	The option reserved for opening recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Default examples already stored in the IDE software
Close	Used for closing the main screen window of recent tab. If two tabs are open, it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program
Save as	It will allow you to save the recent program in your desired folder
Page setup	Page setup is used for modifying the page with portrait and landscape options. Some default page options are already given from which you can select the page you intend to work on
Print	It is used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editor page
Quit	It will quit the whole software all at once

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

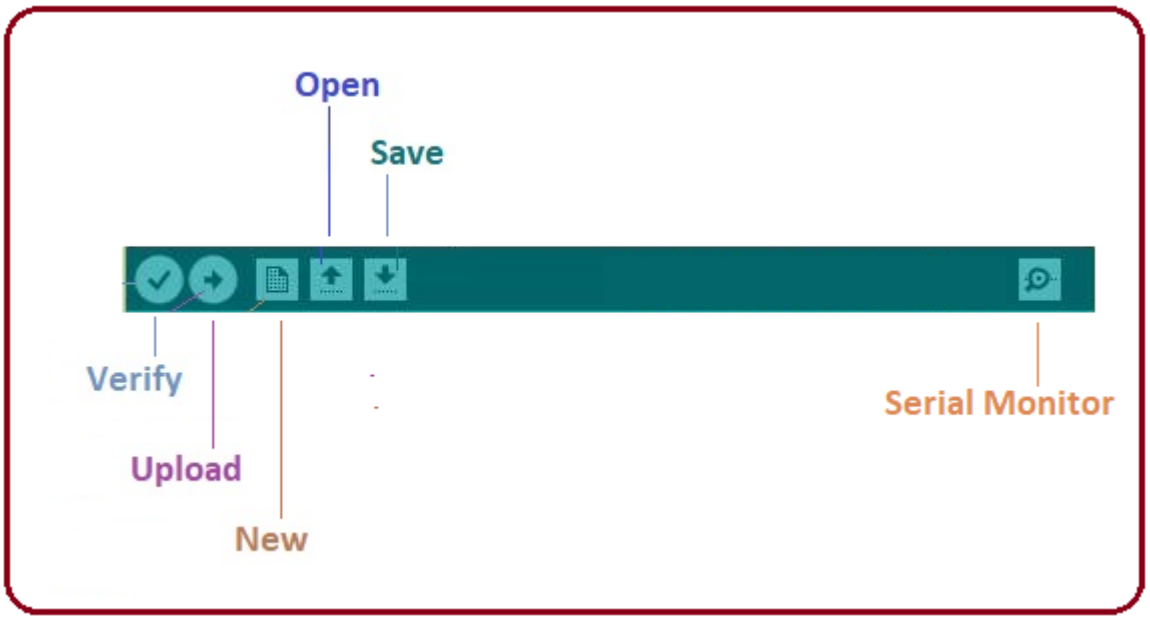
And at the end of the compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.



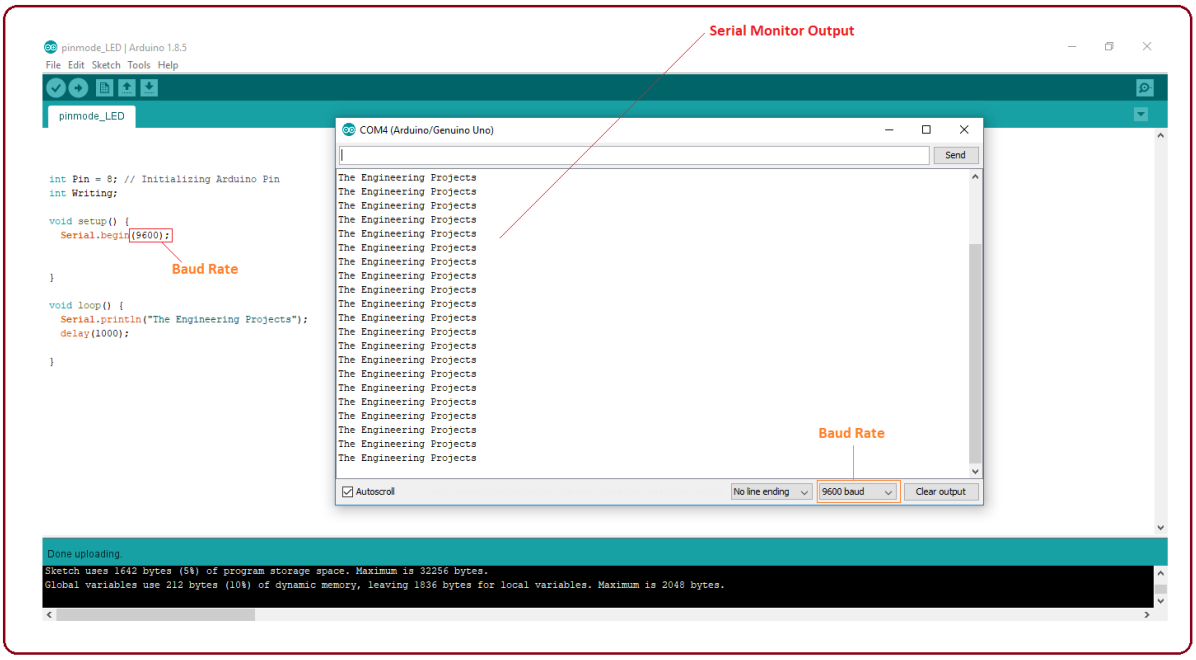
- **Edit –** Used for copying and pasting the code with further modification for font
- **Sketch –** For compiling and programming
- **Tools** – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.

- **Help –** In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

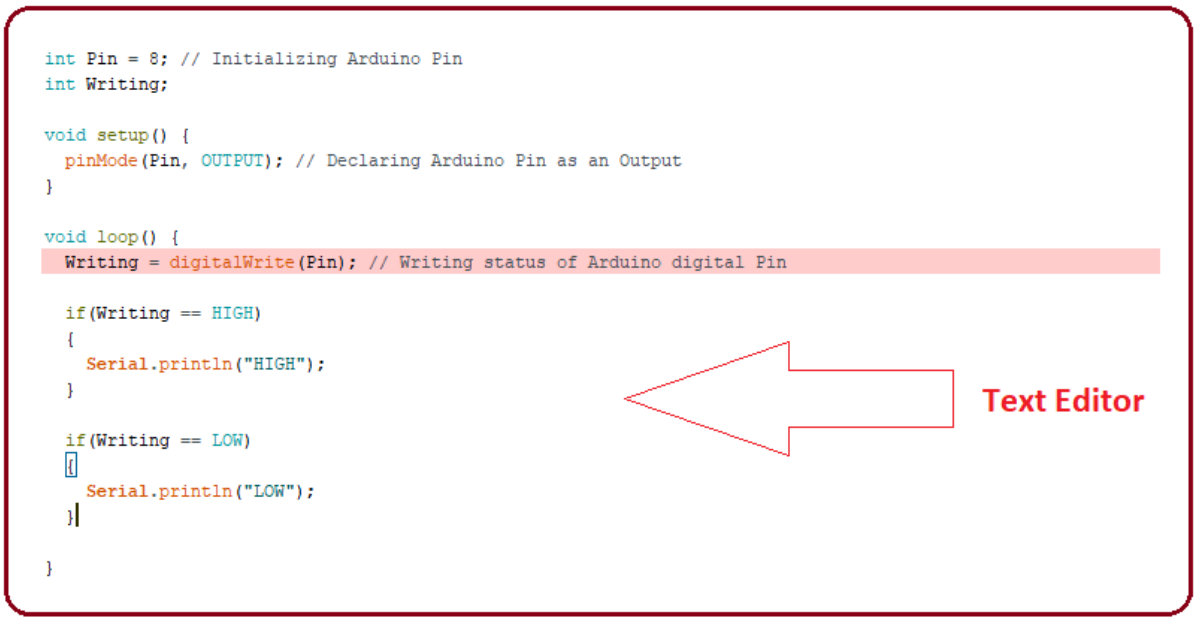
The **Six Buttons** appearing under the Menu tab are connected with the running program as follows.



- The checkmark appearing in the circular button is used to verify the code. Click this once you have written your code.
- The arrow key will upload and transfer the required code to the Arduino board.
- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a Serial Monitor – A separate pop-up window that acts as an independent terminal and plays a vital role in sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
- You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.



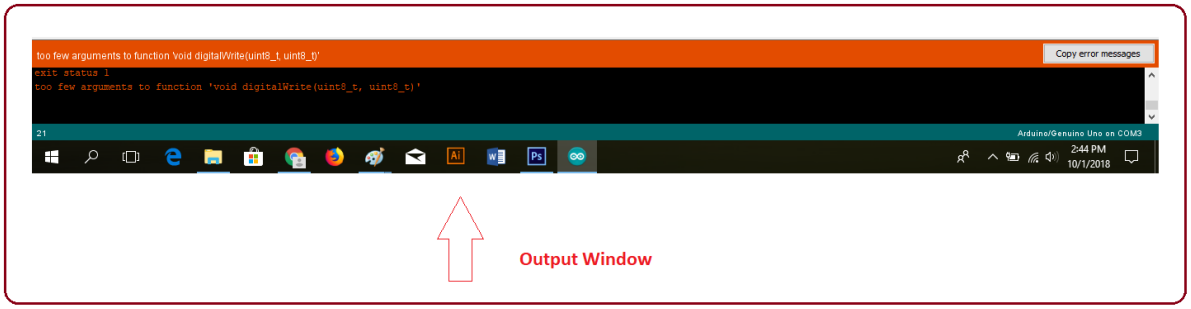
The main screen below the Menu bard is known as a simple text editor used for writing the required code.



The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors that occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.

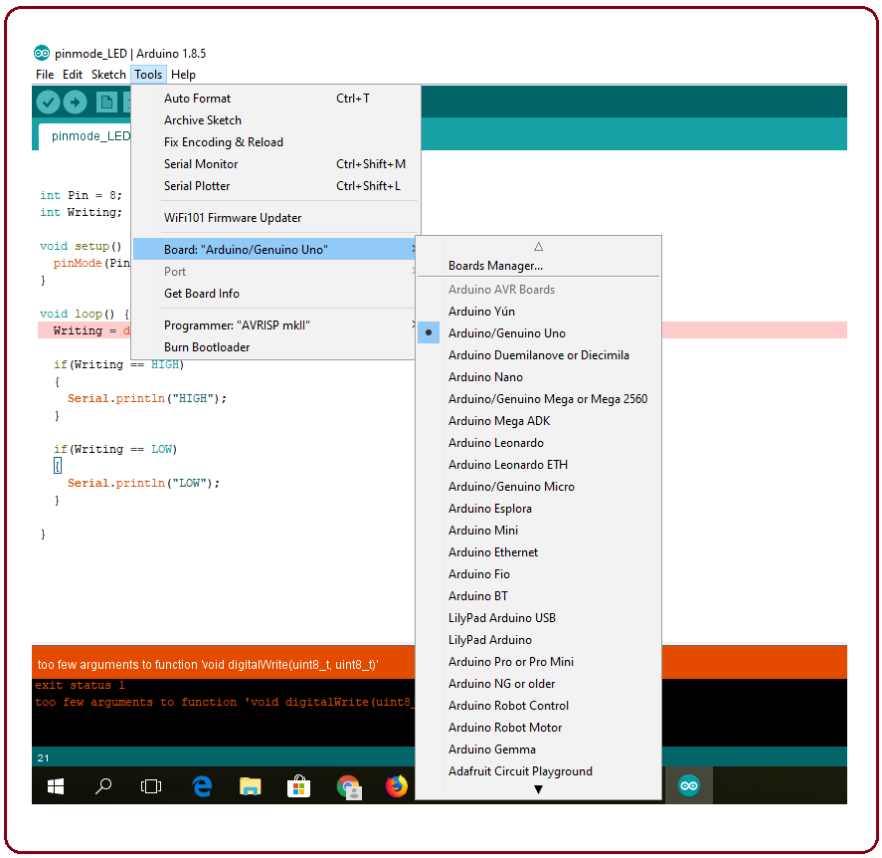
More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific

functions on the board.



5.2.1.2 ARDUINO LIBRARIES

- Libraries are very useful for adding extra functionality into the Arduino Module.
- There is a list of libraries you can check by clicking the Sketch button in the menu bar and going to Include Library.



- As you click the Include Library and Add the respective library it will be on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.h>

- Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from external sources.

5.2.1.3 MAKING PINS INPUT OR OUTPUT

The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W”. Writing it down with Digitalwrite or digitalwrite won’t be calling or addressing any function.

5.2.2 APPEDIX PROGRAM CODE

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);

int motor1_frd = 13;
int motor1_rev = 12;
int motor2_frd = 11;
int motor2_rev = 10;

#define frd_key A0
#define rev_key A1
#define left_key A2
#define right_key A3

char v1=0;

void motor_frd()
{
```

```
  digitalWrite(motor1_frd, HIGH);
  digitalWrite(motor1_rev, LOW);
  digitalWrite(motor2_frd, LOW);
  digitalWrite(motor2_rev, HIGH);
}

void motor_rev()
{
  digitalWrite(motor1_frd, LOW);
  digitalWrite(motor1_rev, HIGH);
  digitalWrite(motor2_frd, HIGH);
  digitalWrite(motor2_rev, LOW);
}

void motor_left()
{
  digitalWrite(motor1_frd, HIGH);
  digitalWrite(motor1_rev, LOW);
  digitalWrite(motor2_frd, HIGH);
  digitalWrite(motor2_rev, LOW);
}

void motor_right()
{
  digitalWrite(motor1_frd, LOW);
  digitalWrite(motor1_rev, HIGH);
  digitalWrite(motor2_frd, LOW);
  digitalWrite(motor2_rev, HIGH);
}
```

```
void motor_stop()
{
  digitalWrite(motor1_frd, LOW);
  digitalWrite(motor1_rev, LOW);
  digitalWrite(motor2_frd, LOW);
  digitalWrite(motor2_rev, LOW);
}


void setup()
{
  Serial.begin(9600);
  delay(100);
  mySerial.begin(9600);
  Serial.print("Voice Controlled WheelChair");
  mySerial.print("Voice Controlled WheelChair");
  pinMode(motor1_frd,OUTPUT);
  pinMode(motor2_frd,OUTPUT);
  pinMode(motor1_rev,OUTPUT);
  pinMode(motor2_rev,OUTPUT);

  pinMode(frd_key,INPUT_PULLUP);
  pinMode(rev_key,INPUT_PULLUP);
  pinMode(left_key,INPUT_PULLUP);
  pinMode(right_key,INPUT_PULLUP);


  digitalWrite(motor1_frd,LOW);
  digitalWrite(motor2_frd,LOW);
  digitalWrite(motor1_rev,LOW);
  digitalWrite(motor2_rev,LOW);
```

```
}

void loop()
{
  if(mySerial.available())
   {
    v1=mySerial.read();
    Serial.print(v1);
   }
    if(v1=='1')
    {
      v1=0;
      delay(200);
      motor_frd();
      delay(2000);
      motor_stop();
    }
   if(v1=='2')
    {
      v1=0;
      delay(200);
      motor_rev();
      delay(2000);
      motor_stop();
    }
     if(v1=='3')
    {
      v1=0;
      delay(200);
      motor_left();
```

```
  delay(2000);
  motor_stop();
}
 if(v1=='4')
{
  v1=0;
  delay(200);
  motor_right();
  delay(2000);
  motor_stop();
}

if(!digitalRead(frd_key))
{
  delay(300);
  Serial.println("Manual Mode: Forward Direction");
  mySerial.println("Manual Mode: Forward Direction");
  while(!digitalRead(frd_key))
  {
  motor_frd();
  }
  delay(500);
  motor_stop();
}

 if(!digitalRead(rev_key))
{
  delay(300);
  Serial.println("Manual Mode: Backward Direction");
  mySerial.println("Manual Mode: Backward Direction");
  while(!digitalRead(rev_key))
```

```
  {
  motor_rev();
  }
  delay(500);
  motor_stop();
}
 if(!digitalRead(left_key))
{
  delay(300);
  Serial.println("Manual Mode: Turning Left");
  mySerial.println("Manual Mode: Turning Left");
  while(!digitalRead(left_key))
  {
  motor_left();
  }
  delay(500);
  motor_stop();
}

 if(!digitalRead(right_key))
{
  delay(300);
  Serial.println("Manual Mode: Turning Right");
  mySerial.println("Manual Mode: Turning Right");
  while(!digitalRead(right_key))
  {
  motor_right();
  }
  delay(500);
  motor_stop();
}  }
```

CHAPTER-6

CONCLUSION

6.1 RESULTS

The project was tested for the movement of the wheel chair using trained voice after the design and development of the self-automated wheel chair with its various interfacing units.

• On the basis of two important aspects, firstly, on the accuracy of the voice system and secondly, wheelchair velocity by means of control commands this design is experimented. This would be implemented for disabled people. Firstly, the voice recognition system will be tested in a quiet room with only one single user. Every word was correctly recognized.

• For a next time, we will test it with a different user on whom the system was not trained. For example, words like “right” were recognized as “write” in this way about 5% errors occurred in this case.

• This was because the recognizer heard a different pronunciation. However, after the user had to speak the word a number of times the system had enough examples and properly determined what pronunciation the user speak of the word.

• There was no problem in correctly recognizing the words when the music was light but the recognizer found it difficult to recognize the user’s voice when we turned the volume high and often took commands from what it heard in the song.

6.2 CONCULSION

The wheelchair is controlled by the commands from the user as well as by the using a smartphone which is connected through the Wi-Fi module. This helps the disabled or the elderly people to move independently and thus eliminating the enslavement. Alerts are given if the person falls down from the wheelchair and stops when an obstacle is detected. The wheelchair also provides vibration therapy for faster recovery of the patient. The efficiency of voice command-based wheelchair can be improved by neural based algorithm.

REFERENCE

[1]https://www.google.com/search?q=wheelchair+control+through+voice+recognition+using+arduino+papers&client=ms-android
oppo&sxsrf=ALiCzsa2yB3WKFFVpaTqt6IukVLIh9q13w%3A1655011597234&ei=DXmlYsCADvWSseMPh6CIuAU&oq=wheelchair+control+through+voice+recognition+using+arduino+papers&gs_lcp=ChNtb2JpbGUtZ3dzLXdpei1zZXJwEAM6BwgjELADECc6BwgAEEcQsAM6BQghEKABOggIIRAeEBYQHToECB4QCkoECEEYAFCnE1iXQ2DNR2gBcAF4AIAB0wGIAYoKkgEFMC40LjOYAQCgAQHIAQnAAQE&sclient=mobile-gws-wiz-serp
[2]https://www.ijert.org/research/arduino-based-voice-operated-wheelchair-with-obstacle-detection-a-survey-IJERTCONV6IS13005.pdf
[3] https://www.researchgate.net/publication/319442825_Voice_Controlled_Wheelchair
[4]https://www.electroniclinic.com/voice-controlled-wheelchair-using-arduino-and-voice-recognition-module/
[5] https://ieeexplore.ieee.org/document/828895