udan University of Science & Technology

College of Postgraduate studies

Design of Voice Based Smart Wheelchair System

تصميم نظام كرسي مدولب ذكي مبني علي التحكم الصوتي

A Thesis Submitted In Partial Fulfillment Of The Requirements For The Degree of M.Sc. In Mechatronics Engineering

Prepared by:

Mohammed Omer Mohammed Abdelrahman

Supervised by:

Dr. Ala Eldin Abdallah Awouda

October 2018

الايـــــــــــــة

بسم الله الرحمن الرحيم

قال تعالى :

(وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ ۖ وَسَتُرَدُّونَ إِلَىٰ عَالِمِ الْغَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ)

(التوبة. ١٠٥)

DEDICATION

To the angel in our life , to the meaning of love and the meaning of compassion and dedication , my smile which is the mystery of my existence and the secret of my success ,To the heart as pure whiteness…

My Mother Granule

To who taught me to give without waiting, to who I carry their names proudly. I ask God to extend his age to see the harvest has come after a long and tomorrow and forever...

My Father

To these who have shown me what is the most beautiful of life, to those who remind me of hope when I need it….

My Brothers, Sisters and Friends

Acknowledgement

Words are not enough to thank my teacher, supervisor and guide Dr. Ala Eldin Abdallah Awouda for his excellent guidance, patience, frank criticism, care and constructing comments in preparing the thesis and I am indebted to him forever.

I would like to express my sincere gratitude to TCH. Ashraf for his constant support and motivation. It would never have been possible for me to take this work to completion without his incredible support and encouragement.

I benefited greatly from many fruitful discussions with my work space mates: Altayeb, Alnour, Hassan and M.fateh Elrahman. I cannot forget there valuable help and motivation.

I owe many thanks to my work colleagues, those who always support and give full attention for me to solve my problem. They always help me in exchanging any ideas and give the enjoyable studying environment. Their friendships are invaluable to me.

ABSTRACT

The main purpose of this project is to design a motorized, voice controlled wheelchair by using speech recognition module

The objective of this project is to facilitate the movement of people who are disabled or handicapped and elderly people who are not able to move well. The result of this design will allow certain people to live a life with less dependence on others.

The design supports voice activation system for physically disabled persons. Arduino microcontroller and speaker dependent voice recognition processor have been used to support the navigation of the wheel chair. GPS and GSM modules have been used for surveillance purposes. The direction and velocity of the chair are controlled by pre-defined voice commands. The wheelchair does not respond to a false speech command. It proved a good working in clear and noisy environments with safe movement.

The results of this project show that this project can be used for future research works and to design excellence innovation that meets market need and public interest.

المستخلص

الهدف الرئيسي من هذا المشروع هو تصميم كرسي مدولب مبني علي التحكم الصوتي عن طريق استخدام اداة تعررف صوتية.

أهداف المشروع هي تسهيل حركة الأشخاص المعاقين والمشلولين وكبار السن اللذين لا يمكنهم التحرك بسهولة. النتائج من هذا المشروع سوف تسمح لكثير من الأشخاص ان يعيشوا حياتهم باعتماد أقل علي الاخرين.

التصميم يدعم نظام صوتي التنشيط للأشخاص المعاقين. تم استخدام ميكروكنترولر (اردوينو) ومعالج تعرف صوتي للتوجيه والتحكم بالكرسي المدولب. كما تم استخدام نظام الملاحة العالمي مع نظام اتصالات لأغراض المراقبة. اتجاه وسرعة الكرسي يتم التحكم بهما عن طريق اوامر صوتية مسبقة التعريف.

الكرسي لا يستجيب لأوامر صوتية خاطئة. ولقد اثبت انه يعمل جيدا في بيئة هادئة وضوضائية مع حركة امنة.

نتائج هذا المشروع أظهرت أنه يمكن استخدام هذا المشروع في الأبحاث المستقبلية لتصميم ابتكار ممتاز يلاقي احتياجات السوق والاهتمام العام.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	DESCRIPTION
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read only Memory
GND	Ground
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MIC	Microphone
PWM	Pulse width Modulation
RXD	Receive Data
SMS	Short Message Service
SIM	Subscriber Identity Module
SR	Speech recognition
TXD	Transmit Data
USB	Universal Serial Bus

CHAPTER ONE
INTRODUCTION

1.1 Overview

The idea of using voice activated technology for controlling the motion of the wheelchair is to prove that it can be a unique concept that would stand apart from the rest of the average projects. The use of this new technology in conjunction with a mechanical system in order to simplify everyday life would spark interest in an ever growing modern society. Many people with disabilities do not have the dexterity necessary to control a joystick on an electrical wheelchair. This can be a great for the quadriplegics who is permanently unable to move any of the arms or legs. They can use their wheelchair easier only using voice commands. The aim of this study is to implement an interesting application using small vocabulary word recognition system. The methodology adopted is based on grouping a microcontroller with a speech recognition development kit for isolated word from a dependent speaker. The resulting design is used to control a wheelchair for a handicapped person based on the vocal command. It therefore involves the recognition of isolated words from a While the needs of many individuals with disabilities can be satisfied with power wheelchairs, some members of the disabled community find it is difficult or impossible to operate a standard power wheelchair. This project could be part of an assistive technology. It is for more independent, productive and enjoyable living limited vocabulary. In order to gain in time design, tests have shown that it would be better to choose a speech recognition kit and to adapt it to the application. There are five options for basic motions of a wheelchair to be applied by the user. The five conditions of the wheelchair can be described as the following:

i. Moving forward to the front of the user

ii. Moving backward to the back of the user

iii. Turning to the right

iv. Turning to the left

v. Static or stop condition.

1.2 Problem Statements

Independent mobility is crucial for development of physical, cognitive, communicative and social skill for physically impaired people. The high price of the electric wheel chairs. This project is thus aimed at the development of more sophisticated control scheme for electric powered wheelchair.

The main problem of the wheelchair is that cannot be used by disabled person, so the type of artificial aid needed by a disabled person in order to move about depends, to a large extent, on the level of his incapacity. So no need to handles behind the seat to allow it to be pushed by another person.

1.3 Proposed Solution

The challenge for engineering is to provide safe and effective mobility in a dynamic environment. Through thoughtful research and design, power wheelchair control will progress along safe and effective pathways towards providing users independent and self-guided mobility. This project will give the severely disabled people an innovative solution to control their wheelchair using voice interfacing.

1.4 Project Significances

User interface is an important component of any product handle by the human user. The concept of the design is to make a voice activated wheelchair, which can replace the use of a joystick.

In the past decades GUI (Graphical User Interface), Keyboard, Keypad, Joystick is the dominating tools for Interaction with machine. Now from them SR (Speech recognition) system is one of the interesting tool to the researchers for interaction with machine. The reason draws attention to the researcher, because people are used to communicate with a natural language in the social context. So this technology can be widely-accepted to the human user fairly and easily. But for the wheelchair application more researches and more analysis have to be done. This is because this will include the human safety and more over this kind of application is very new especially in Sudan. Thus the project is significant because:

i. Speech processing can be done in real time and has long been considered as a natural to assist powered wheelchair user.

ii. Many disabled people exist in today's world and require help in order to overcome physical challenges. Thus this project will provide an alternative to the disabled in controlling the motion of the wheelchair using their voices.

iii. The efficiency of using voice controlled wheelchair can be identified.

1.5 Objectives

The main aim is to design a voice controlled system to allow disabled people for acting.

The objectives are:

A. To propose and develop a wheelchair control system using voice recognition system.

B. To simulate the proposed system.
C. To practically implement the system.

1.6 Research Methodology

The design involves incorporation of a voice recognition module which provides voice store and recognition features. Inclusion of this module along with GSM and GPS module will progress along safe and effective pathways towards providing users independent and self-guided mobility

The voice commands are recorded and stored in the memory and the database using VR module and Arduino software. The Arduino Software allows user to specify which voice command for which movement, also to add a mobile number for surveillance.

Arduino microcontroller is used as main processing unit to read data from the VR module and start processing, leading the motors accordingly using L293D Motor driver module, and update the information received from the GPS and send it to the mobile number using GSM.

1.7 Thesis organization

Thesis is summarized in five chapters:

Chapter 1: Introduction explain problem statement, proposed solution and objectives.

Chapter 2: Literature review highlight previous work and explain system component.

Chapter 3: System design and explain how it work.

Chapter 4: simulation, result and discussion.

Chapter 5: Conclusion and recommendation.

CHAPTER TWO
LITERATURE REVIEW

2.1 Overview

The idea of using voice activated technology for controlling the motion of the wheelchair and home automation are to prove that it can be a unique concept that would stand apart from the rest of the average projects. The use of this new technology in conjunction with a mechanical system in order to simplify everyday life and it would spark interest in an ever growing modern society. Many people with disabilities do not have the dexterity necessary to control a switch on an electrical wheelchair. This can be a great for the quadriplegics who is permanently unable to move any of the arms or legs. They can use their wheelchair easier only using voice commands and also they can control home appliances. The aim of this study is to implement an interesting application using small vocabulary word recognition system. The methodology adopted is based on grouping a microprocessor with a speech recognized development kit for isolated word from a dependent speaker. The resulting design is used to control a wheelchair and home appliances for a handicapped person based on the vocal command. In order to gain in time design, tests have shown that it would be better to choose a speech recognition kit and to adapt it to the application.

2.2 Previous studies

Research from University of Notre Dame, 2000, suggests that the current power wheelchair control interfaces used may not, be adequate to provide truly independent mobility for substantial number of person with disabilities.The Respondents to the survey reported on average that approximately ten percent of the patients trained to operate a power wheelchair cannot use the chair upon completion of training for activities of daily living or can do so only with extreme difficulty [1].

Wheelchairs have evolved very little over the past 1000 years. Most of the design changes have occurred within recent decades as shown in the following outline of wheelchair history.

In the 6th Century A.D. - Earliest recording of a wheelchair; a Chinese engraving picturing a man in a chair with three wheels [2]. Also in the 16th Century A.D. - Wheelchairs were well-developed in Europe and commonly found in drawings and literature. After that the first wheelchair patent was issued in the United States in 1869 [3].

In 1903 - An electrically-driven wheelchair operating on a 12-volt battery and a 3/8 horsepower motor was used to give people rides. At the time it was not used for handicapped mobility but it did pave the way for future developments [2].

Compact wheelchairs were developed using metal tubing instead of the traditional bulky wood components in 1909. World War I - The first electric wheelchairs were used for the handicapped. A battery and motor were applied to existing wheelchairs with a simple one-speed on/off switch.

In 1937 - The patent for a wheelchair with a folding X-brace frame was issued to two engineers named Everest and Jennings. Though previous chairs had been foldable top-to-bottom, the side-to-side folding position of the cross frame allowed the drive wheels to remain in place. This basic concept is still the standard for manual wheelchairs today [4].

The first patent was issued for an electric wheelchair in 1940 [4].

In 1950 - Sam Duke received a patent for a releasable add-on power drive applied to manual wheelchair (the unit was actually permanently fitted to the chair with U-bolts) [2].

1960's - Folding wheelchairs were commonly fitted with electric drives.
The drive units were still very heavy and quite difficult to put on and take off. At that point both joystick and steering column mechanisms were available [2].
1970's - Wheelchair frames made of aircraft quality aluminum were introduced to the market and started a revolution of ultra-light wheelchairs. The technology has aided in the reduction of the overall weight of many types of wheelchairs [4].
1980's - Most electric wheelchairs on the market were still bulky, heavy, and required a special vehicle for transportation. The power components of the chair were integrated into the frame which has been strengthened to support them [4].
1990's - The popular electric wheelchairs on the market are foldable though they require removal of at least the leg rests and batteries. The Katalavox speech-recognition control system can be used by quadriplegics to control their power wheelchair. The commands are combined to emulate the movements of a joystick. This voice controlled wheelchair was not been commercialized but it is customized for individual used.
2000's – The use of joystick, head or chin control and sip and puff control for severely disabled people are recognized. There are also other interfacing used like eye gaze; tongue pad; head, hand, foot switch controls.

2.3 Wheelchair Components

There are a number of possible driving wheel configurations (front wheel drive, rear wheel drive and mid wheel drive) which affect the characteristics of the chair in different situations, with turning while driving being the most complex. Further features can be added to assist the user such as lights, actuators and wireless links. The heart and brains of the powered wheelchair is in the controller as it provides both a conduit for the power to the motors and controls the overall system. The typical powered wheelchair user is disabled in a way that means they rely almost totally on the software contained within this controller to provide safe and reliable performance. This reliance is the same for every user, no matter the ability, preference, or operating environment.

The general features of the wheelchair are as the following.

i. The user interface. It can be a programmable joystick, however many other methods of control are possible (sip and puff, scanning, head movement, etc.) as power wheelchairs have sophisticated electronics to control their motors.

ii. The seating and postural support. Some power wheelchair models have features like power stand, power recline, power tilt, and power elevation.

iii. Power wheelchairs come with more tire and powerbase options.

iv. Prices of the mobility rises depending on the features it has.

v. Power wheelchairs have a variable type speed control knob so we can set the speed from 1 to 5+ mph. It can accept a 4 point tie-down in a motor vehicle which could tend to make them safer as a seat in a motor vehicle.

vi. Batteries for most power wheelchairs are gel cell sealed batteries which are approved for transport. Most electric wheelchairs are equipped with 2 gel cell 12 volt batteries capable of going 15 to 20 miles on a full charge over level terrain. These sealed batteries are approved for transport.

vii. Very few power wheelchairs are breakdown for placement in the trunk of a car. Vehicles most always need to be adapted with a lift or ramps because they are too heavy.

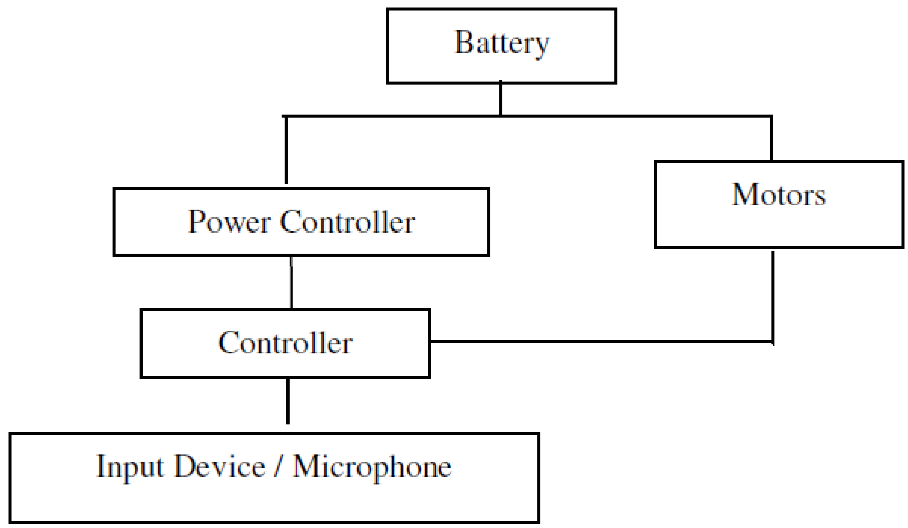


Figure (2.1) Basic Components of Electrical Power Wheelchair

2.4 Related Researches

Several researchers have considered using human voice to control powered wheelchairs.

Naturally, a wheelchair voice control system should operate reliably for a large number of users, reduce the physical requirements; and if avoiding the need to move on one or more road extremities, should assist a user in maintaining well the chair position. However, the voice's limited bandwidth makes it difficult to adjust frequently the wheelchair's velocity, and also a voice input system may fail to identify a speaker. Thus, voices interface has yet to become commercially viable for wheelchair control; rather its use is normally suggested in combination with a navigation assistance system for obstacle identification and avoidance in the wheelchair's path [5].

The example of other research doing the Voice Recognition is done by acquiring the Microsoft SDK 5.1 software development kit, which has the necessary capabilities. The team will be using the Speech Recognition (SR) engines provided by the SDK to interpret voice commands. The control algorithms for smooth movement are an incremental system, thus the current state of the velocity parameters (V?) will be updated based on the specific voice commands. This allows the developer to use a small number of commands to create a fluid and flexible control motion, while also maintaining a short training period and ease of control for the user.

Speech recognition systems were first used by severely disabled individuals with normal speech. The goal was to promote independence whereby SR was used to convert human speech signals into effective actions. Frequently, speech is the only remaining means of communication left for these individuals. The first voice activated wheelchair with an environmental control unit (ECU) was developed in the late 1970s at Rehabilitation Medicine in New York.

The user could operate multiple items including the telephone, radio, fans, curtains, intercom, page-turner and more. A group of individuals with cerebral palsy rated the wheelchair as superior to breath control systems because it eliminated the need for scanning, allowing the user quicker access by directly selecting the desired function with voice.

The first voice activated power wheelchair was used by a young Norwegian law-student in 1984. It enabled him to attend his classes without the help of an attendant. The wheelchair was customized for him using Katalavox speech recognition system.

There are some other researches have been done to upgrade the wheelchair using assistive technology. A good research example is done by S. Rao and R. Kuc from Yale University. They built and analysis a prototype of an intelligent wheelchair which is equipped with ultrasonic range sensors and a Motorola 68HC11.

The prototype was designed to detect fall-offs and objects in its path, thereby addressing the needs of visually impaired persons who has been confined to wheelchairs.

2.5 Speech Recognition

Speech recognition is the process of converting an acoustic signal, captured by micro- phone or a telephone, to a recognized command or word.

There two important part of in Speech Recognition - i) Recognize the series of sound and ii) Identified the word from the sound. This recognition technique depends also on many parameters - Speaking Mode, Speaking Style, Speaker Enrollment, Size of the Vocabulary, Language Model, Perplexity, Transducer etc. There are two types of Speak Mode for speech recognition system - one word at a time (isolated-word speech) and continuous speech. Depending on the speaker enrolment, the speech recognition system can also divide - Speaker dependent and Speaker independent system. In Speaker dependent systems user need to be train the systems before using them, on the other hand Speaker independent system can identify any speaker's speech. Vocabulary size and the language model also important factors in a Speech recognition system. Language model or artificial grammars are used to confine word combination in a series of word or sound. The size of the vocabulary also should be in a suitable number. Large numbers of vocabularies or many similar sounding words make recognition difficult for the system.

The digital processing capabilities of microcontrollers have enabled voice control to penetrate embedded systems. These new microcontrollers, sometimes called embedded digital signal processors or DSP controllers, have sufficient performance for real-time speech processing, and they integrate almost all needed control peripherals on one piece of silicon.

Voice control implies that the system will recognize only a limited command set, not fluent speech. That limitation markedly decreases memory and performance requirements compared with those needed for fluent-speech recognition.

2.6 Voice Recognition Module

In this project VR module, voice recognition processor is chosen. It can be programmed words for both speaker dependent and speaker independent. Moreover it is easy to program and low cost. The module could recognize voice. It receives configuration commands or responds through serial port interface. With this module, we can control the car or other electrical devices by voice.

Figure (2.2) VR module

The module can store 15 pieces of voice instruction. Those 15 pieces are divided into 3 groups, with 5 in one group. First we should record the voice instructions group by group. After that, we should import one group by serial command before it could recognize the 5 voice instructions within that group. If we need to implement instructions in other groups, we should import the group first. This module is speaker independent. If your friend speaks the voice instruction instead of you, it may not identify the instruction. The speaker independence requires strictly good MIC. The MIC we supply is not good enough for it to be speaker-independent.

2.6.1 VR Parameters

• Voltage: 4.5-5.5V

• Current: <40mA

• Digital Interface: 5V TTL level UART interface

• Analog Interface: 3.5mm mono-channel microphone connector + microphone pin interface

• Size: 30mm x 47.5mm

• Recognition accuracy: 99% (under ideal environment)

This module can be configured by sending commands via serial port. Configuration will be not erased after powered off.

Its interface is 5V TTL. The serial data format: 8 data bits, no parity, 1 stop bit. The default baud rate is 9600 and baud rate can be changed.

Command format is "Head + Key". "Head" is a 0xaa, and "Key" is as follows:

- Delete Group 1 - send hex AA 01
- Delete Group 2 - send hex AA 02
- Delete Group 3 - send hex AA 03
- Delete All Groups - send hex AA 04
- Record Group 1 - send hex AA 11
- Record Group 2 - send hex AA 12
- Record Group 3 - send hex AA 13
- Import Group 1 - send hex AA 21
- Import Group 2 - send hex AA 22
- Import Group 3 - send hex AA 23
- Query the recorded group - send hex AA 24

To modify the serial baud rate to 38400, you need to send command: 0xaa35. If successful, it will return "Baud: 38400 \n"(in Common Mode) or 0xcc (in Compact Mode). The baud rate is set to 38400.

The main difference between Compact Mode and Common Mode is the returning message. Common Mode response is long string but Compact Mode response is a byte. For example, after sending 0xaa04 to delete all the contents of the 3 groups, in Common Mode it will return "All Groups Deleted! \ n", but in Compact Mode it will return a concise bytes such as 0xcc which means a successful operation.

For the first-time use, we need to do some configuration:

1. Select the serial baud rate (default 9600)
2. Select the communication mode: Common Mode or Compact Mode
3. Recording five instructions of the first group (or 2nd or 3rd as required)
4. Import the group you need to use (only recognize 5 instructions within one group at the same time)

After all the setting above, you can speak or send voice instruction to it. If identified successfully, result will be returned via serial port in the format: group number + command number. For example, return Result: 11 (Compact mode returns 0x11) means identified the first command of group 1.

If voice instruction is recorded, each time after you power it on, you need to import the group before letting it identify voice instructions.

2.6.2 LED Indicators

Recording stage:

1. Record indication: D1 (RED) flashes 3 times within the 600ms, then off for400ms, and then flashes quickly for 4 times within 600ms. Now the recording indication is over.

2. Begin to speak: D1 (RED) is off for 400ms, and then is on. Voice during the time while D1 (RED) is on will be recorded by this module.
3. Recording a voice instruction successfully for the first time: D1 (RED) off, D2 (ORANGE) on for 300ms.
4. Recording a voice instruction successfully for the first time: D1 (RED) off, D2 (ORANGE) on for 700ms.
5. Recording failure: D2 (ORANGE) flashes 4 times within the 600ms. In cases that voice instructions detected twice don't match, or the sound is too large, or there is no sound, recording will fail. You need to start over the recording process for that instruction.

In waiting mode, D2 (ORANGE) is off, and D1 (RED) is on for 80ms every other 200ms, fast flashing. In this mode, it doesn't recognize voice command, only waiting for serial commands.

In identification stage, D2 (ORANGE) is off, and D1 (RED) is on for 100ms every other 1500ms, slow flashing. In this stage, this module is processing received voice signal, and if matching, it will send the result immediately via serial port.

2.6.3 Recording

Before using it, we have to record voice instructions. Each voice instruction has the maximum length of 1300ms, which ensures that most words can be recorded. Once you start recording, you can't stop the recording process until you finish all the 5 voice instructions recording of one group. Also, once you start recording, the previous content of that group will be erased. In recording stage, this module doesn't reply to any other serial commands.

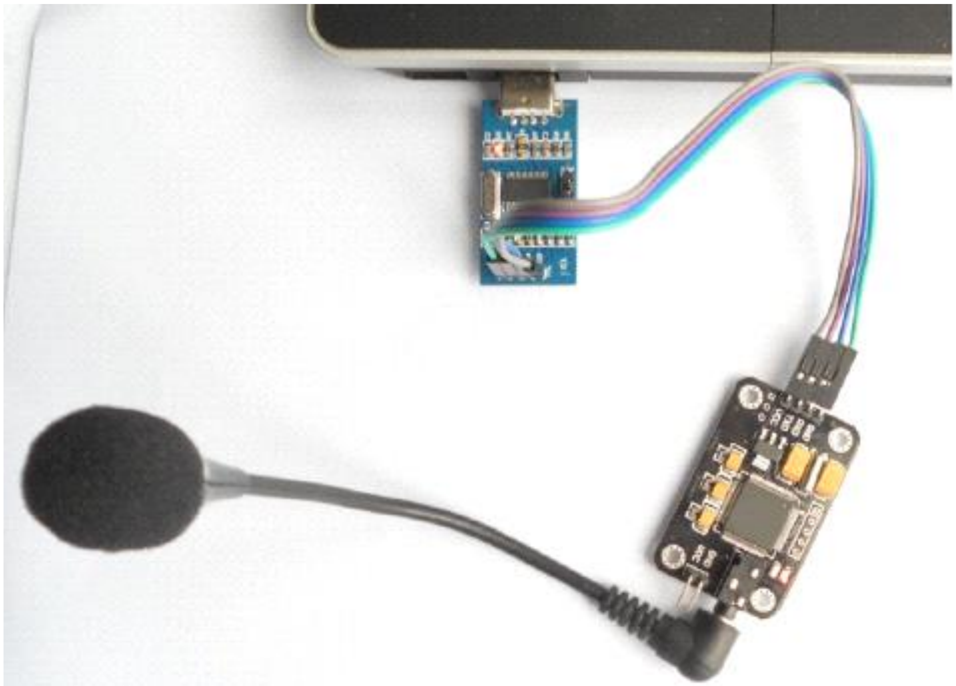


Figure (2.3) Recording Tools

LED will flash to indicate state.

Serial port setting:

- Baud rate: 9600
- Parity bit: None
- Data bit: 8
- Stop bit: 1
- Send format: Hex
- Receive format: Char

2.7 Arduino Microcontroller

Arduino is open Source electronic prototyping platform based on flexible easy to use hardware and software. It is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible.

The hardware consists of a simple 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 [6].

2.7.1 Arduino hardware

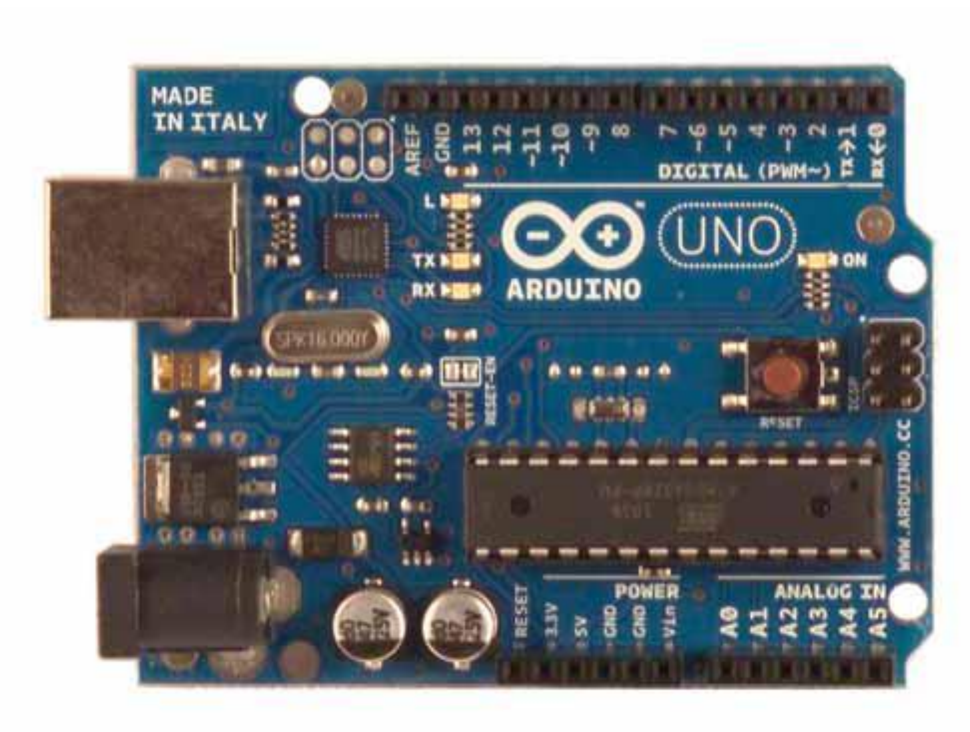


Figure (2.4) Arduino Hardware

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended):7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analogy Input Pins: 6
- DC Current per I/O Pin: 40mA

• DC Current for 3.3V Pin: 50mA

• Flash Memory: 32 KB (ATmega328)

• SRAM: 2 KB (ATmega328)

• EEPROM: 1 KB (ATmega328)

• Clock Speed: 16 MHz

The Arduino board is a small-form microcontroller circuit board. At the time of this writing, a number of Arduino boards exist:-

• Arduino Uno

• Arduino Leonardo

• Arduino Lily Pad

• Arduino Mega

• Arduino Nano

• Arduino Mini

• Arduino Mini Pro

• Arduino BT

Figure (2.5) Arduino Family

	Processor	Processor Voltage	Supply Voltage	Flash	SRAM	Digital I/O Pins	PWM Pins	Analog Inputs	Hardware Serial Ports	Dimensions	Shield Compatibility	Notes and Special Features
Uno	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Excellent (most will work)	
Uno Ethernet	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Very Good (some pin conflicts)	Has Ethernet Port. Requires FTDI cable to program.
Mega	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	
Mega ADK	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	Works with Android Development Kit.
Leonardo	16MHz Atmega 32U4	5v	7-12v	32Kb	2.5Kb	20*	7	12*	1	2.1"x2.7" 53x75mm	Fair (many Pinout Differences)	Native USB capabilities. USB Micro B programming port.
Due	84MHz ARM SAM3X8E	3.3v	7-12v	512Kb	96Kb	54	12	12	4	2.1"x4" 53x102mm	Poor (voltage and pinout differences)	Fastest processor. Most memory. 2-channel DAC. USB micro B programming port. Native micro AB port.
Micro	16MHz Atmega 32U4	5v	5v	32Kb	2.5Kb	20*	7	12*	1	0.7"x1.9" 18x49mm	N/A	Smallest board size. Native USB capabilities
Flora	8MHz Atmega 32U4	3.3v	3.5-16v	32Kb	2.5Kb	8*	4	4*	1	1.75" dia 44.5mm dia	N/A	Sewable Pads. Fabric-friendly design. Native USB Capabilities
DC Boarduino	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. Requires FTDI cable for programming
USB Boarduino	16MHz Atmega 328	5v	5v (USB)	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. USB Mini B programming port.
Menta	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	Excellent (most will work)	Mint-Tin Size and Prototyping Area. Requires FTDI cable for programming.

Table (2.1) Compare between the different types of Arduino

2.7.2 Arduino Software

Arduino microcontrollers are programmed using the Arduino IDE (Integrated Development Environment). Arduino programs, called “sketches”, are written in a programming language similar to C and C++.

Every sketch must have a setup () function (executed just once) followed by a loop () function (potentially executed many times); add “comments” to code to make it easier to read. Many sensors and other hardware devices come with prewritten software line for sample code, libraries (of functions).

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in Liquid Crystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download [7].

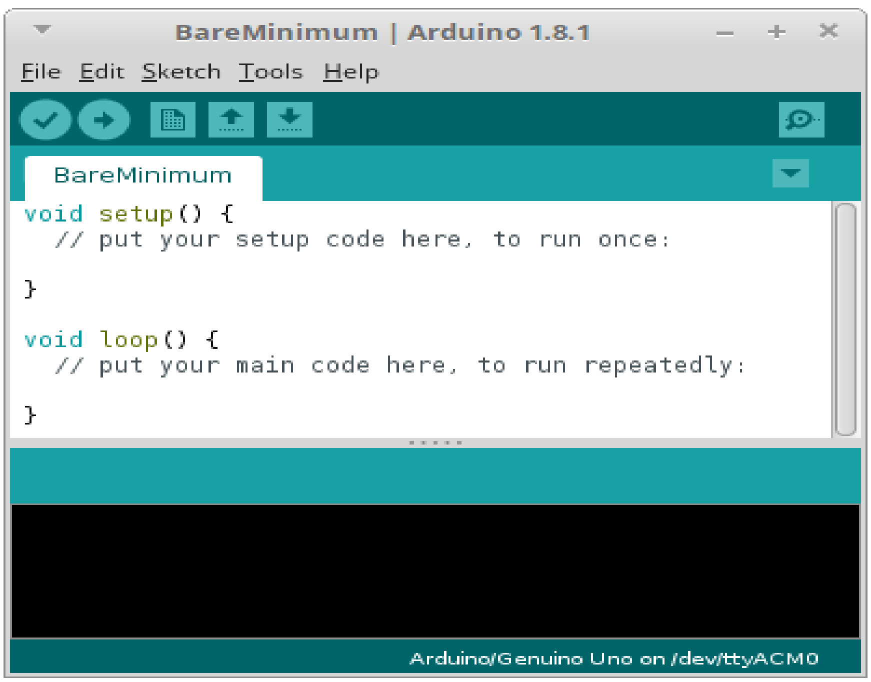


Figure (2.6) Arduino IDE

2.7.3 Arduino UNO

The Arduino UNO is microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM output), 6 analogy input, a 16 MHZ crystal oscillator, AUSB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller, simply connect it to a computer with USB cable or power it with an AC-to- DC adapter or battery to get started.

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. We don't advise it.

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

- **GND.** Ground pins.

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write (), and digital Read () 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.

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

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.

- **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 [8].

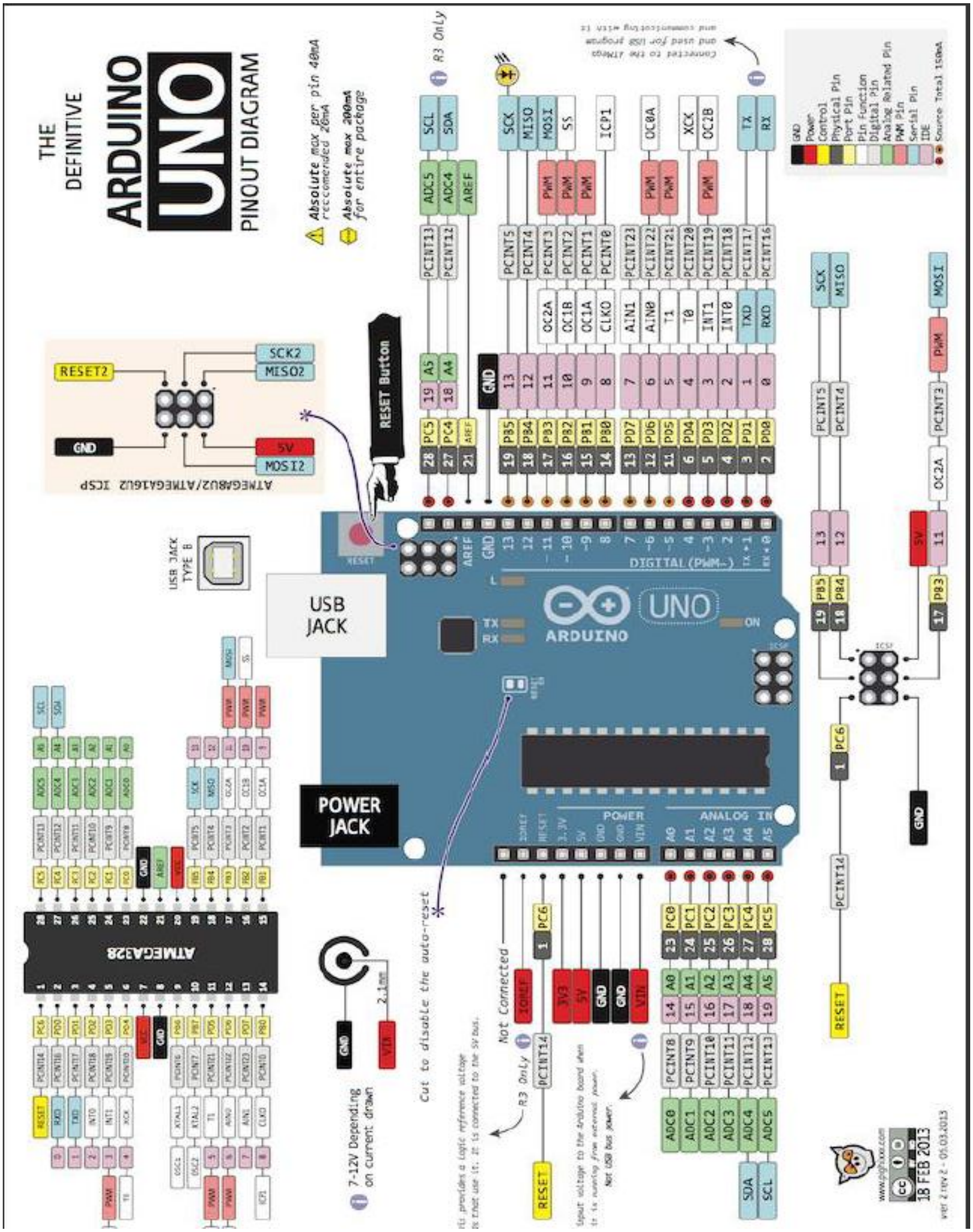


Figure (2.7) Arduino UNO

2.8 Global Positioning System (GPS)

The Global Positioning System (GPS) is a satellite-based navigation system made up of at least 24 satellites. GPS works in any weather conditions, anywhere in the world, 24 hours a day, with no subscription fees or setup charges.
The U.S. Department of Defense (USDOD) originally put the satellites into orbit for military use, but they were made available for civilian use in the 1980s.

2.8.1 GPS operation

GPS satellites circle the Earth twice a day in a precise orbit. Each satellite transmits a unique signal and orbital parameters that allow GPS devices to decode and compute the precise location of the satellite. GPS receivers use this information and trilateration to calculate a user's exact location. Essentially, the GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a user's position and display it electronically to measure your running route, map a golf course, find a way home or adventure anywhere.

2.8.2 GPS module

A GPS navigation device, or simply GPS is a GPS receiver module piece of hardware that you add to other piece of hardware (e.g. car head unit, Arduino even your computer) to give it the possibility to receive information from GPS satellites and then to calculate the device's geographical position. Using suitable software, the device may send SMS, display the position on a map, and it may offer directions.

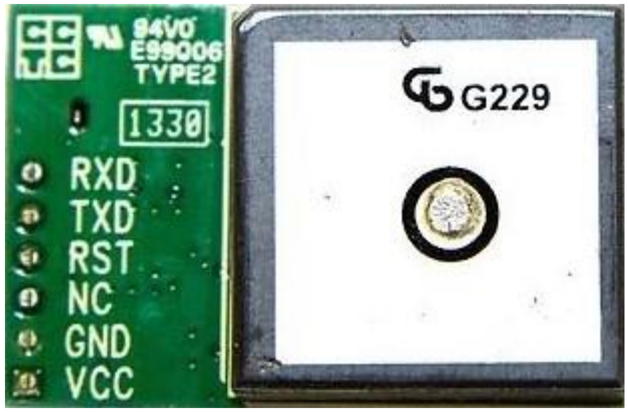


Figure (2.8) GPS module

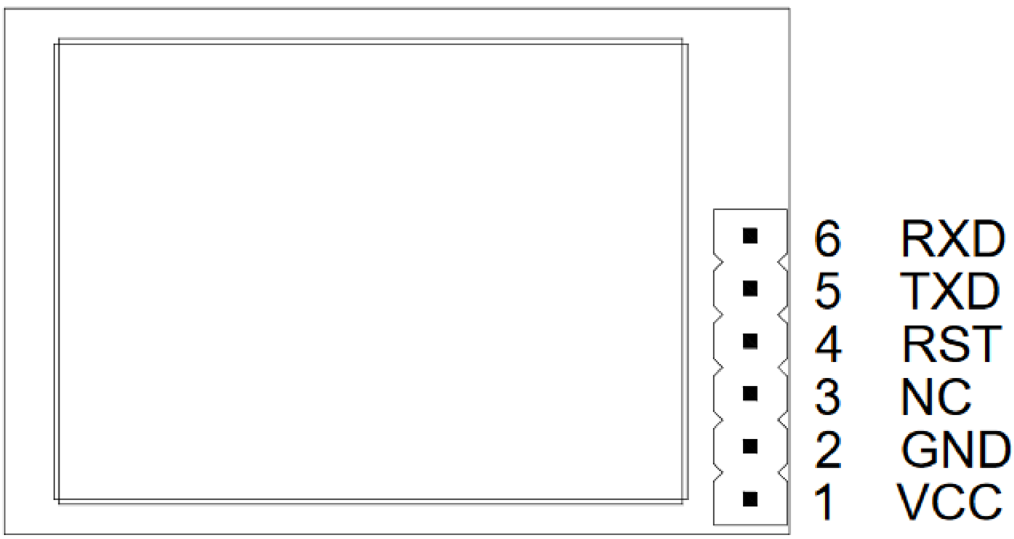


Figure (2.9) GPS pin description

- **Application :**
  - LBS (Location Based Service)
  - Vehicle navigation system
  - PND (Portable Navigation Device)
  - GPS mouse and Bluetooth GPS receiver
  - Timing application

Table no 2.2. GPS pin assignment

Pin No.	Pin name	Description	Remark
1	VCC	Module Power Supply	VCC: 5V±5%
2	GND	Module Power ground	Reference Ground
3	NC	Not open	Leave open
4	RST	Module Reset	
5	TXD	Receive Data	Leave Open in not used
6	RXD	Transmit Data	Leave Open in not used

2.9 Global System for Mobile communications GSM

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services. GSM supports voice calls and data transfer speeds of up to 9.6 kbps, together with the transmission of SMS (Short Message Service).

A GSM modem is a special type of modem that accepts a SIM card and operates over a subscription to a mobile operator just like as a mobile phone. GSM modem is a wireless modem which sends and receives data through radio waves.

A GSM modem requires a SIM card from a wireless carrier in order to operate Just like as a GSM mobile phone. GSM modem support standard AT commands as well as extended set of AT commands. With the standard AT commands and extended AT commands, you can do things like:

• Sending SMS message

• Reading, Writing and Deleting SMS massage

• Monitoring the signal strength

• Reading, Writing and Searching phonebook entries

• Real time clock

GSM operates in the 900MHz and 1.8GHz bands in Europe and the 1.9GHz and 850MHz bands in the US. GSM services are also transmitted via 850MHz spectrum in Australia, Canada and many Latin American countries. The use of harmonized spectrum across most of the globe, combined with GSM's international roaming capability, allows travelers to access the same mobile services at home and abroad. GSM enables individuals to be reached via the same mobile number in up to 219 countries.

Terrestrial GSM networks now cover more than 90% of the world's population. GSM satellite roaming has also extended service access to areas where terrestrial coverage is not available [9].

2.9.1 GSM Features

· Quad Band GSM/GPRS: 850 / 900 / 1800 / 1900 MHz

· Built in RS232 to TTL or vice versa Logic Converter (MAX232)

· Configurable Baud Rate

· SMA (Sub Miniature version A) connector with GSM L Type Antenna

· Built in SIM (Subscriber Identity Module) Card holder

· Built in Network Status LED

· Inbuilt Powerful TCP / IP (Transfer Control Protocol / Internet Protocol) stack for internet data transfer through GPRS (General Packet Radio Service)

· Audio Interface Connectors (Audio in and Audio out)

· Most Status and controlling pins are available

· Normal Operation Temperature: -20 °C to +55 °C

· Input Voltage: 5V to 12V DC.

2.9.2 Hardware Description of GSM module

Figure (2.10) GSM module

- **SIM Com SIM900A GSM Module:**

This is actual SIM900 GSM module which is manufactured by SIM Com. Designed for global market, SIM900 is a quad-band GSM/GPRS engine that works on frequencies GSM 850MHz; EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM900 features GPRS multi slot class 10/ class 8 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4. With a tiny configuration of 24mm x 24mm x 3mm, SIM900 can meet almost all the space requirements in User’s applications, such as M2M, smart phone, PDA and other mobile devices.

Figure (2.11) SIM Com SIM900

- **SIM (Subscriber Identity Module) Card Slot:**

This on board SIM card slot provides User functionality of insert a SIM (GSM only) card of any service provider. Process of inserting and locking SIM card into SIM card slot is given in this manual. While inserting in and removing out SIM card from SIM card slot, User needs to take precaution that power supply should be OFF so that after making Power supply ON it wills be easy to reinitialize with SIM for this module.

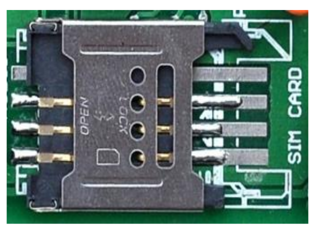


Figure (2.12) Card Slot

- **Indicator LEDs:**

Indicator LEDs just used to indicate status accordingly. These are three LEDs represents Power On/Off Status, Network Status and Module On/Off Status respectively. Power LED will keep on until the power supply is enable to this board. Network Status LED will show whether inserted SIM card successfully connected to service provider's Network or not, in short signal strength. Module On/Off indicator LED will show status of GSM module's power on/off.

Figure (2.13) Indicator LEDs

- **RXD, TXD and GND pins:**

These pins are used to connect devices which need to be connected to GSM module through USART (Universal Synchronous Asynchronous Receiver and Transmitter) communication. Devices may be like Desktop or Laptop Computer System, Microcontrollers, etc. RXD (Receive Data) should be connected to TXD (Transmit Data) of other device and vice versa, whereas GND (Ground) should be connected to other device's GND pin to make ground common for both systems.

Figure (2.14) RXD, TXD and GND pins

2.10 Motors and Motor driver

A motor is defined as an electric or mechanic device that can create a motion. While interfacing with the controller; some of the motors like DC motor, stepper motor and brushless dc motor may require a driver IC or driver circuit. DC motor is a type of motor that can convert DC into a mechanical power.

A motor driver is a little current amplifier; 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.

Motor drivers chosen by filtering the results by various attributes: by Maximum Output Current (15 mA, 1 A, 1.2 A, 1.5 A,…), Maximum Supply Voltage (46 µV, 36 V, 52 V, …) and Number of Outputs (from 1 to 12). [10].

2.10.1 L293D Motor driver

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. Dual H-bridge Motor Driver integrated circuit (IC).

- **L293D Pin Diagram and Configuration:**

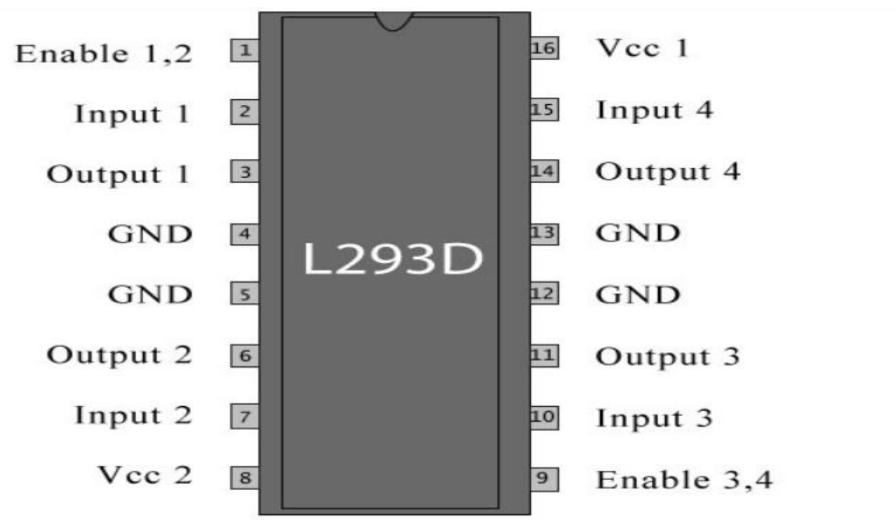


Figure 2.15: L293D Pin Diagram

Table 2.3. L293D pin assignment

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)
9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

CHAPTER THREE

SYSTEM DESIGN

3.1 Introduction

This chapter is about describing the flow chart, block diagram and the description of the system**.**

3.2 System Design

The main part of the design is to control the motion of the wheelchair. There are four types of motions are considered, moving forward, moving in reverse direction, moving to the left and moving to the right. For the speed, the user may use slow or fast speed. Slow speed is important as the user want to move in short distance or approaching an object.

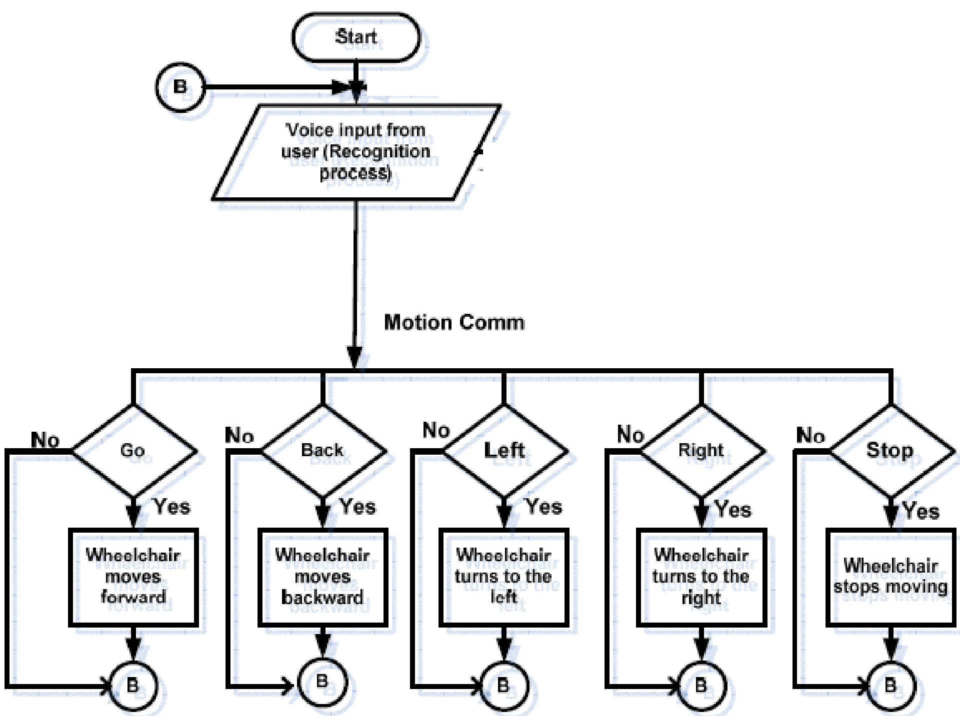


Figure 3.1: Flowchart for the motion controlled wheelchair using voice

3.3 Description of the System

The system has being installed using the following components:-

- VR module. (Simulated by Virtual terminal in the simulation circuits).
- 2Micro controllers Arduino Uno. (One Arduino Uno cannot be programmed for three transmitter-receiver devices).
- Motor driver.
- 2 Motors.
- GPS module.
- GSM module.

3.4 Signal movement

The system is designed of two circuits as follow:-

1\ The first circuit:

- VR module RXD pin is connected to the Arduino in pin PD1\TXD, and the VR module TXD pin to the Arduino pin PD0\RXD.

The driver is used to connect the motors with the Arduino by connecting the inputs pins [IN1, IN2, EN1, EN2, IN3, IN4] with pins of Arduino [PD7, PB0, PD3, PD5, PB1, PB2] and the outputs of the driver are connected to the motors. The pins [OUT1, OUT2] are connected to motor NO.1, and [OUT3, OUT4] to motor NO.2.

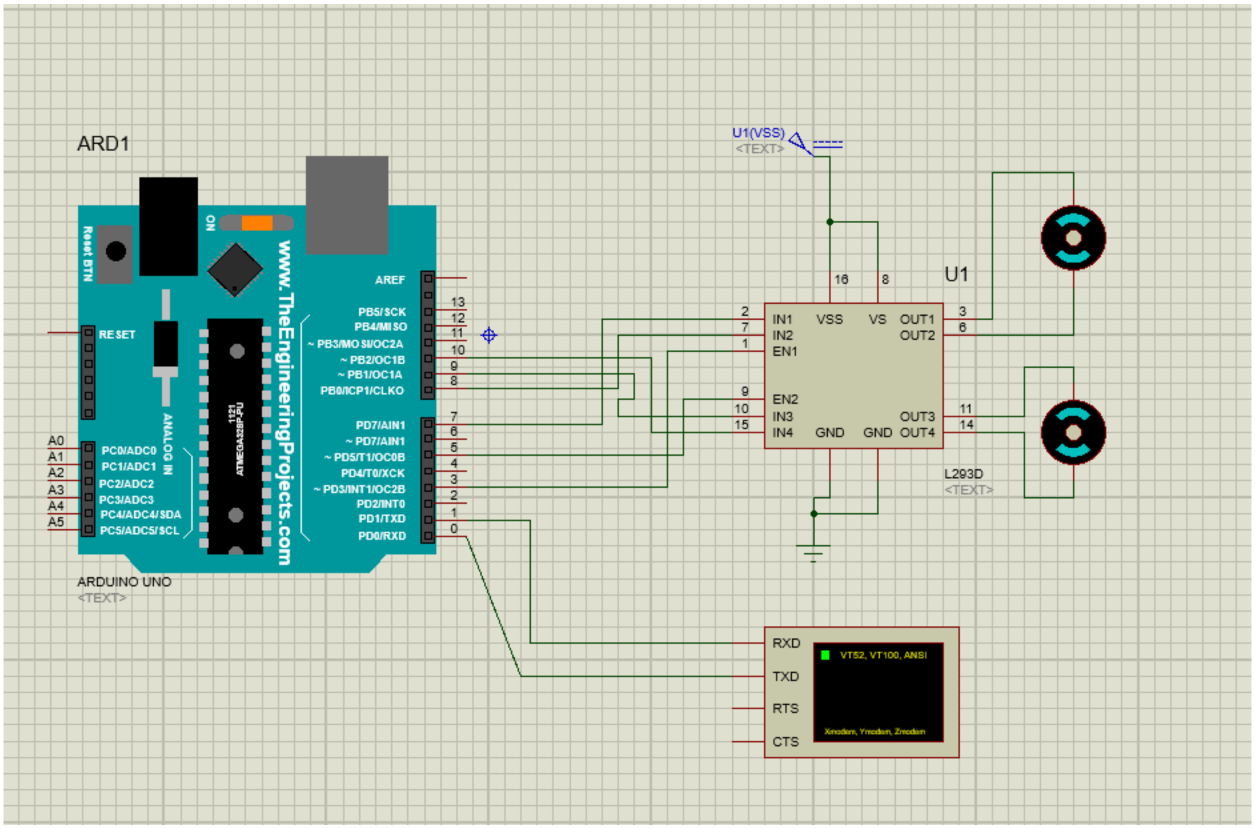


Figure 3.2: circuit diagram of the system “Part one”

2\ The second circuit:

- GSM module RXD pin is connected to the Arduino in pin PD1\TXD, and the TXD pin to the Arduino pin PD0\RXD.
- GPS module RXD pin is connected to the Arduino in pin PD2, and the TXD pin to the Arduino pin PD3.

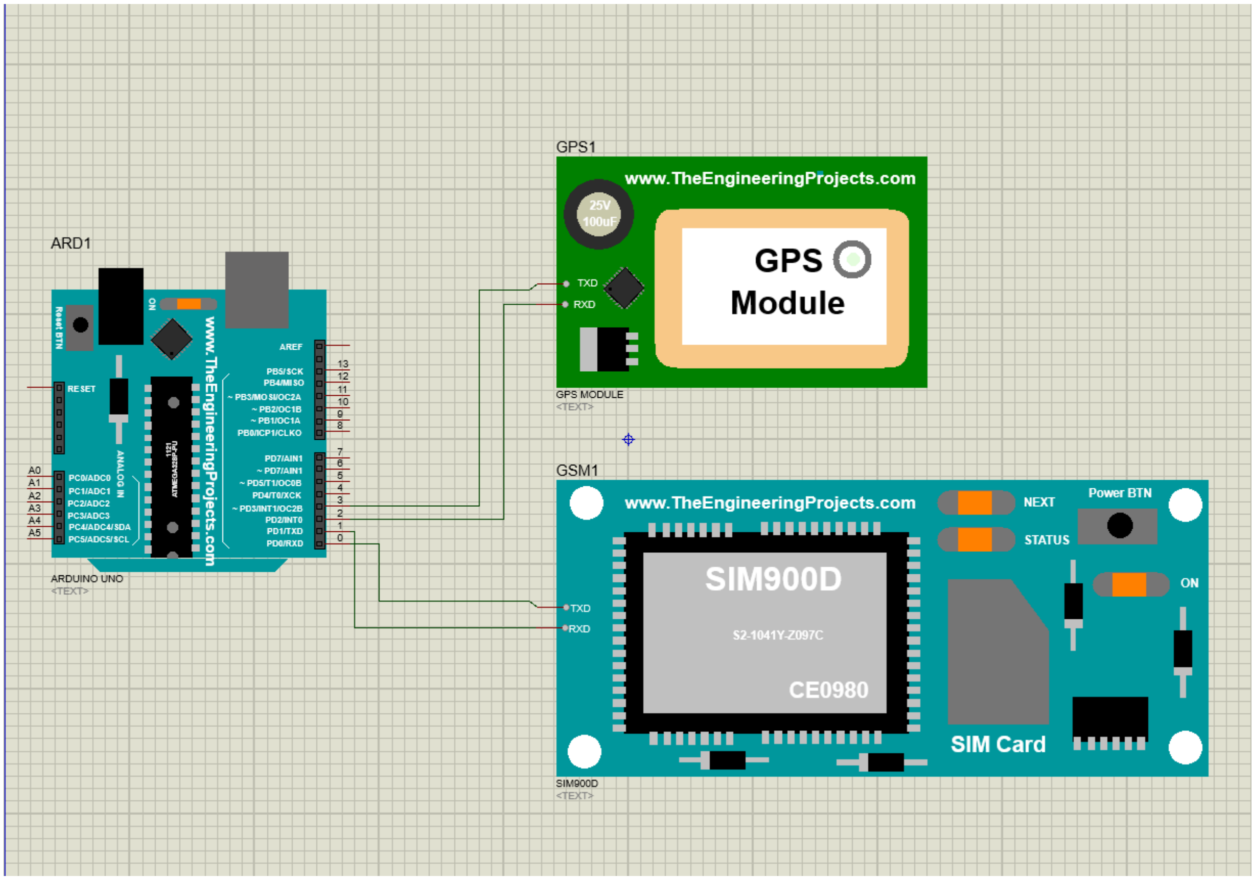


Figure 3.3: circuit diagram of the system "Part Two"

The system starts by applying the supply voltage to the speech recognition circuit. The system will be in stand by condition which the LED on circuit recognition board will be turned on.

The direction and speed of wheelchair depend on the user. Forward command the wheelchair move in forward direction. For the reverse direction the opposite movement of wheel rotation will occur. The left command will make right wheel moves forward and left wheel stops. The right command makes left wheel moves forward and right wheel stops.

In this system, by assigning the word command stop the rotation of both motors will stop. The wheelchair system will go back to the stand by condition or end the whole system by turning off the power supply of the speech recognition board. The voice commands used are as the table 3.1 below.

Table 3.1: Voice command

Command	Condition
Forward	Moving straight to the forward
Backward	Moving straight in the backward
Right	Turning to the right
Left	Turning to the left
stop	No motion/wheelchair stops

3.5 Speech recognition board

The step begins by pressing the word number which wants to train on the keypad. The circuit can be trained to recognize up to 15 words. It can be used any numbers between 1 and 15. In another way if the 15 word vocabulary is not desirable, it can configure the circuit for the word vocabulary as this configuration usually provides better recognition accuracy.

In this project, second method has been chosen because only have five commands are needed to execute.

It is started by pressing the number "0x11" to train word. When the number is activated, on the keypad the red LED will turn off. The chosen number is displayed on the serial display.

Before we replace the display board with interfacing circuit, the confirmation need to take by looking at digital display from board for make sure that the right commands was trained. Then we start training the voice commands. Then, the user should say the specific command clearly to the microphone. LED will blink off momentarily showing that the word is trained.

Finally, one by one of the commands which are list in the table 3.1 are speak out with different number. For each of them, the LED should blink off momentarily; this is a sign that the word has been accepted.

The chip provides the following error codes: 43 for word too loud, 46 for word too noise and 41 for word no match.

There is also another way to change or erasing words. The words can easily be changed by overwriting the original word. For instance suppose word two was the word “back” and if we want to change it to the word “reverse”. By pressing “6” then saying the word “reverse” into the microphone then the word had been trained. If we wish to erase the word without replacing it with another word.

In this project the memory used be train to the speech recognition are as table 3.2.

Table 3.2: The memory used for storing commands

Serial display	**Voices**
0x11	Forward
0x12	Back
0x13	Right
0x14	Left
0x15	stop

3.6 Electronic Circuit Development

Developing the circuit for the voice recognition circuit is easy as the speech recognition kit is used. In this kit, the VR version. 2 processor is already assembled with the input and output port, memory chip and the digital display. The instructions given by the supplier must be followed carefully so that the system can work properly.

CHAPTER FOUR

RESULTS AND DISCUSSION

After the design and development parts are completed, some testing and analysis are done. This includes testing on the accuracy of the system.

4.1 Accuracy for Speech Recognition Circuit

Condition 1: silent area

This experiment was conducted in a room which is in quiet condition to affect the result of the experiment. Experiment purpose is to find out the accuracy of the VR speech recognition circuit in different conditions.

Five trials were done to the circuit base on the commands listed at the table 4.1.

Command conditions:

- 1 is recognized.
- 0 is not recognized.

Table 4.1: The result in silent area

Trial / command	1	2	3	4	5	Total
Forward	1	1	0	1	1	4
Back	1	1	1	1	0	4
Right	1	1	1	1	1	5
Left	1	1	1	1	1	5
stop	1	0	1	1	1	4

From table 4.1, there are 22 over 25 commands recognized by the VR v.2.0 speech recognition circuit. The percentage of the accuracy of speech recognition circuit in silent condition is 88%. Calculation for percentage is shown as below.

Accuracy = 22/25 X100% = 88%

Condition 2: noisy area

The testing is done outside of the quiet room where it is considered as natural environment. From this testing, the results are as table 4.2.

Table 4.2: The result in noisy area

Trial / command	1	2	3	4	5	Total
Forward	0	1	0	1	1	3
Back	1	1	1	1	0	4
Right	1	1	0	0	1	3
Left	1	0	1	1	1	4
stop	0	0	1	1	1	3

From table 4.2, there are 17 over 25 commands recognized by the speech recognition circuit. The percentage of the accuracy of VR speech recognition circuit in silent condition is 68%. Calculation for percentage is shown as below.

Accuracy = 17/25 X100% = 68%

From the result, we can find out that the speech recognition circuit accuracy is less when assign the commands in the noisy area. That means the voice controlled wheelchair system has less control when in the noisy condition, but still working.

4.2 System Implemention from the simulation

There are five cases for the VR and one case for GPS and GSM when implementing the system, LCD screen was used to describe what is happening.

4.2.1 VR case (1)

When the simulation is run and the user enter (0x11) to the screen of the virtual terminal. As a result (forward) message will appear on the screen of the LCD.

Simultaneously the Arduino will set the driver pins (EN1 and EN2) as an output and digital write (IN1, HIGH); (IN2, LOW); (IN3, HIGH); (IN4, LOW);

The voltage will be set to the corresponding value: 5V for HIGH, 0V (ground) for LOW.

The motors will move in forward direction as shown in Figure (4.1).

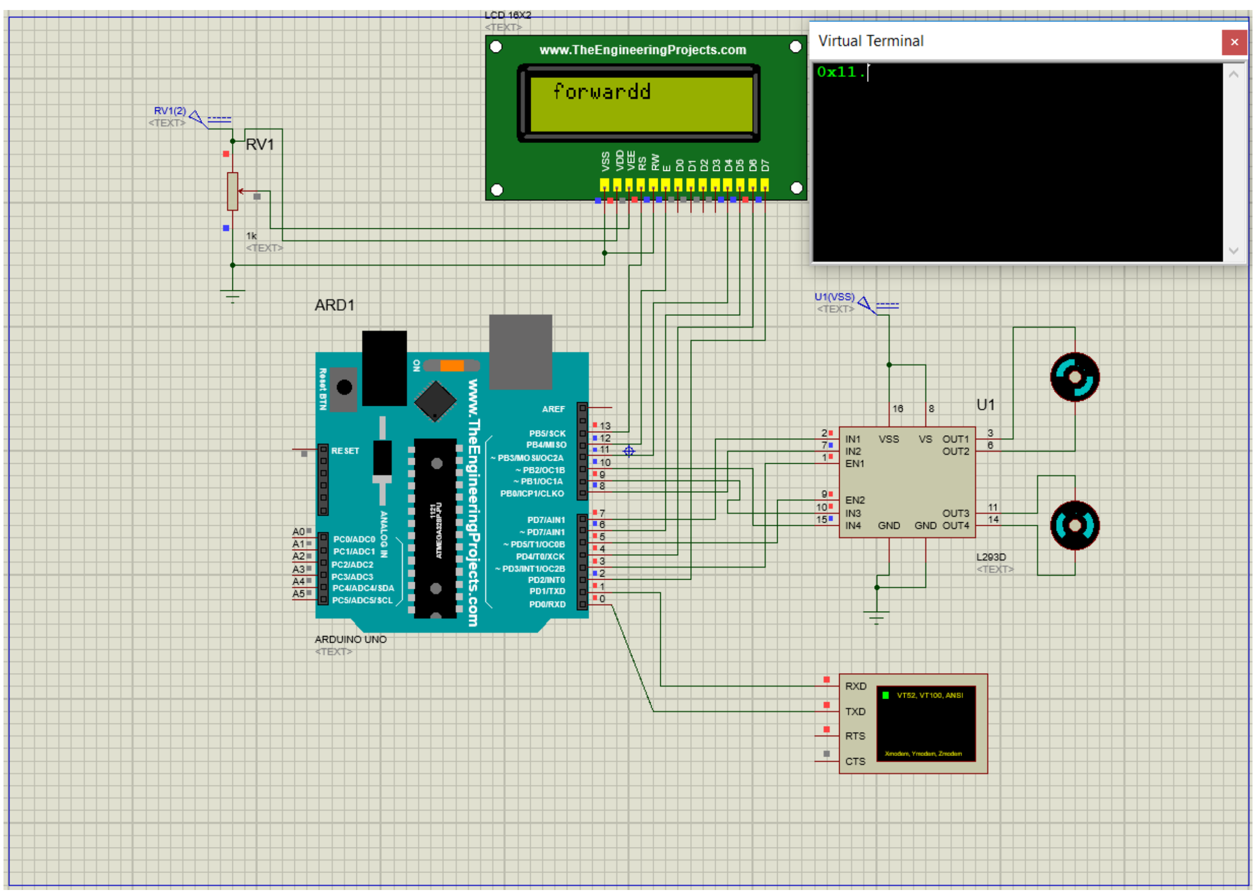


Figure 4.1: forward condition

4.2.2 VR case (2)

When the simulation is run and the user enter (0x12) to the screen of the virtual terminal. As a result (backward) message will appear on the screen of the LCD.

Simultaneously the Arduino will set the driver pins (EN1 and EN2) as an output and digital write (IN1, LOW); (IN2, HIGH); (IN3, LOW); (IN4, HIGH);

The voltage will be set to the corresponding value: 5V for HIGH, 0V (ground) for LOW.

The motors will move in the reverse direction and the opposite movement of wheel rotation will occur as shown in Figure (4.2).

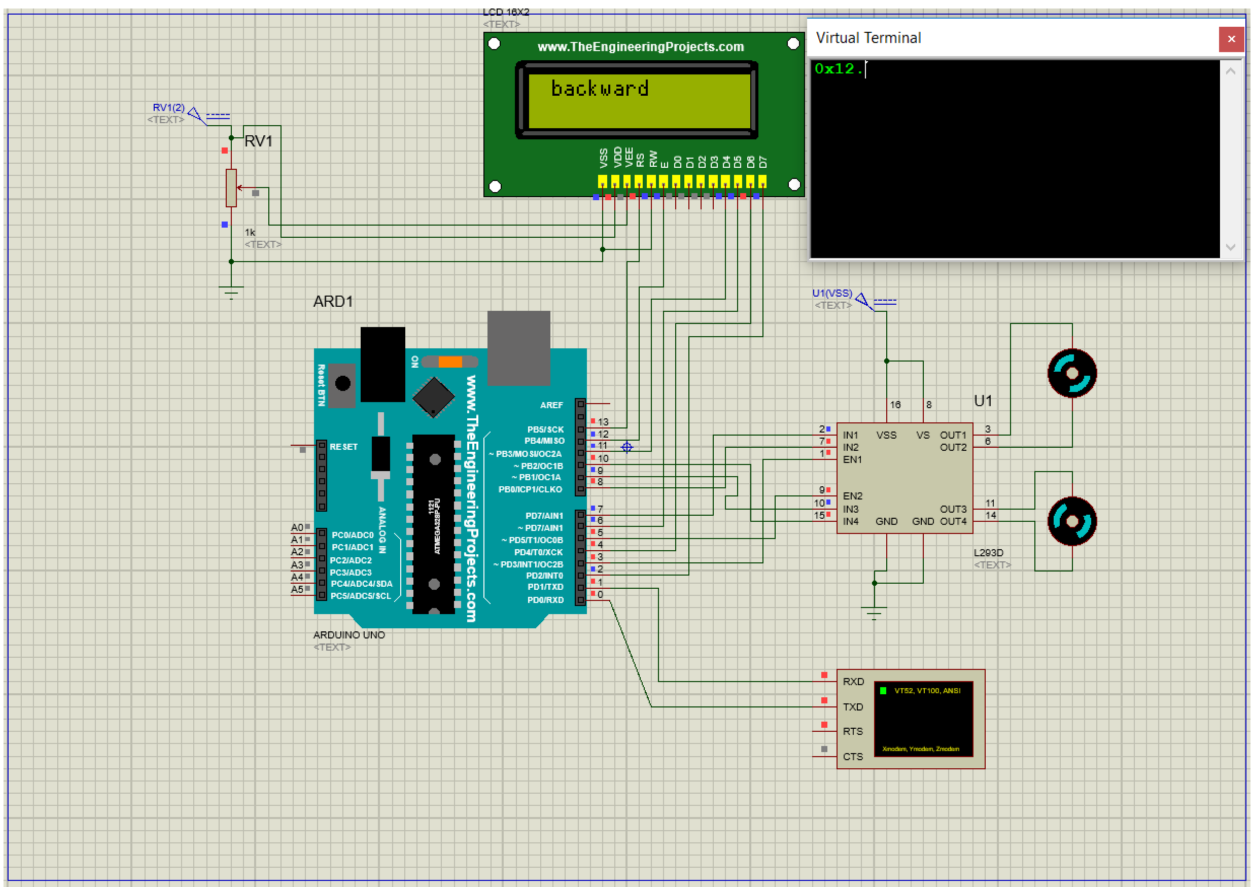


Figure 4.2: backward condition

4.2.3 VR case (3)

When the simulation is run and the user enter (0x13) to the screen of the virtual terminal. As a result (right) message will appear on the screen of the LCD. Simultaneously the Arduino will set the driver pins (EN1 and EN2) as an output and digital write (IN1, HIGH); (IN2, LOW); (IN3, LOW); (IN4, LOW);

The voltage will be set to the corresponding value: 5V for HIGH, 0V (ground) for LOW.

The left motor will move forward and right motor will stop as shown in Figure (4.3).

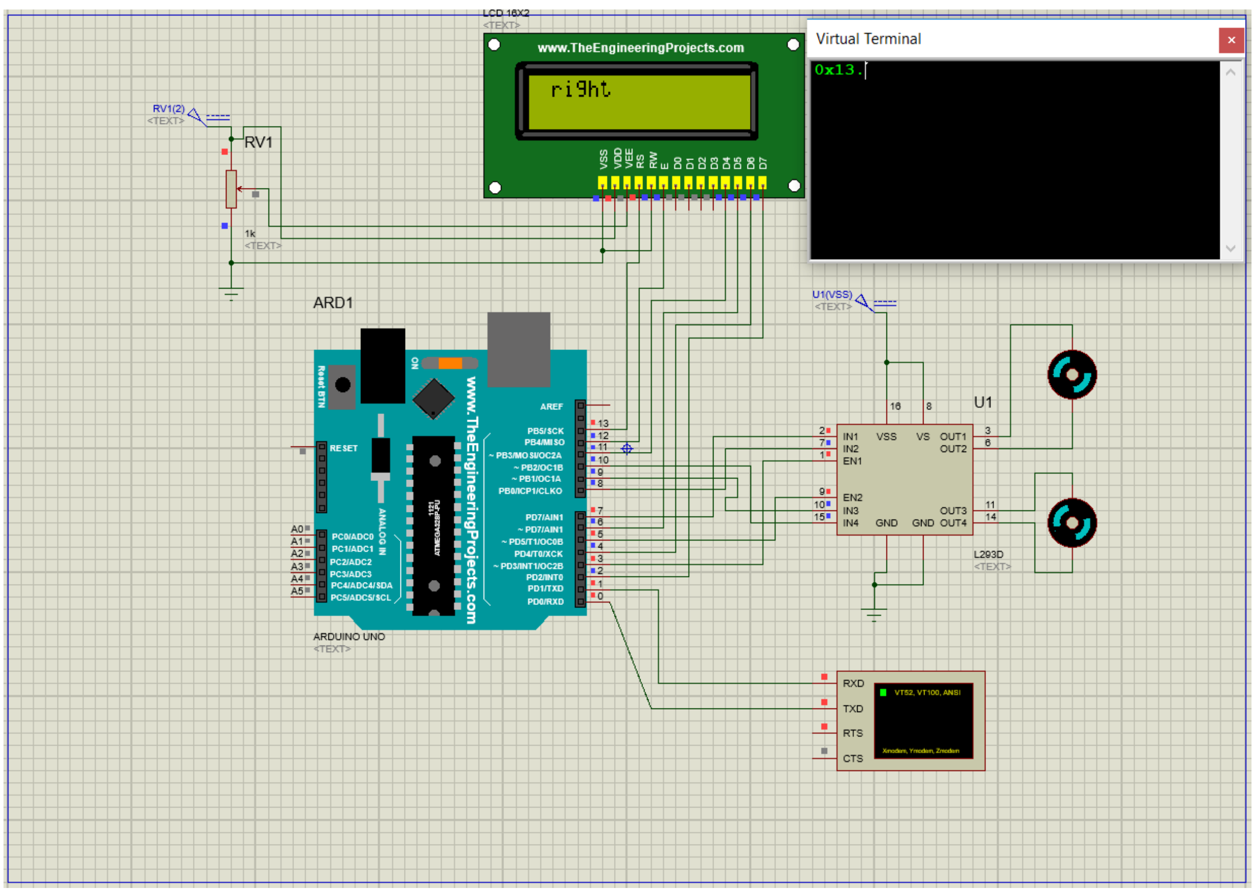


Figure 4.3: right condition

4.2.4 VR case (4)

When the simulation is run and the user enter (0x14) to the screen of the virtual terminal. As a result (left) message will appear on the screen of the LCD. Simultaneously the Arduino will set the driver pins (EN1 and EN2) as an output and digital write (IN1, LOW); (IN2, LOW); (IN3, HIGH); (IN4, LOW);

The voltage will be set to the corresponding value: 5V for HIGH, 0V (ground) for LOW.

The left motor will stop and right motor will rotate forward as shown in Figure (4.4).

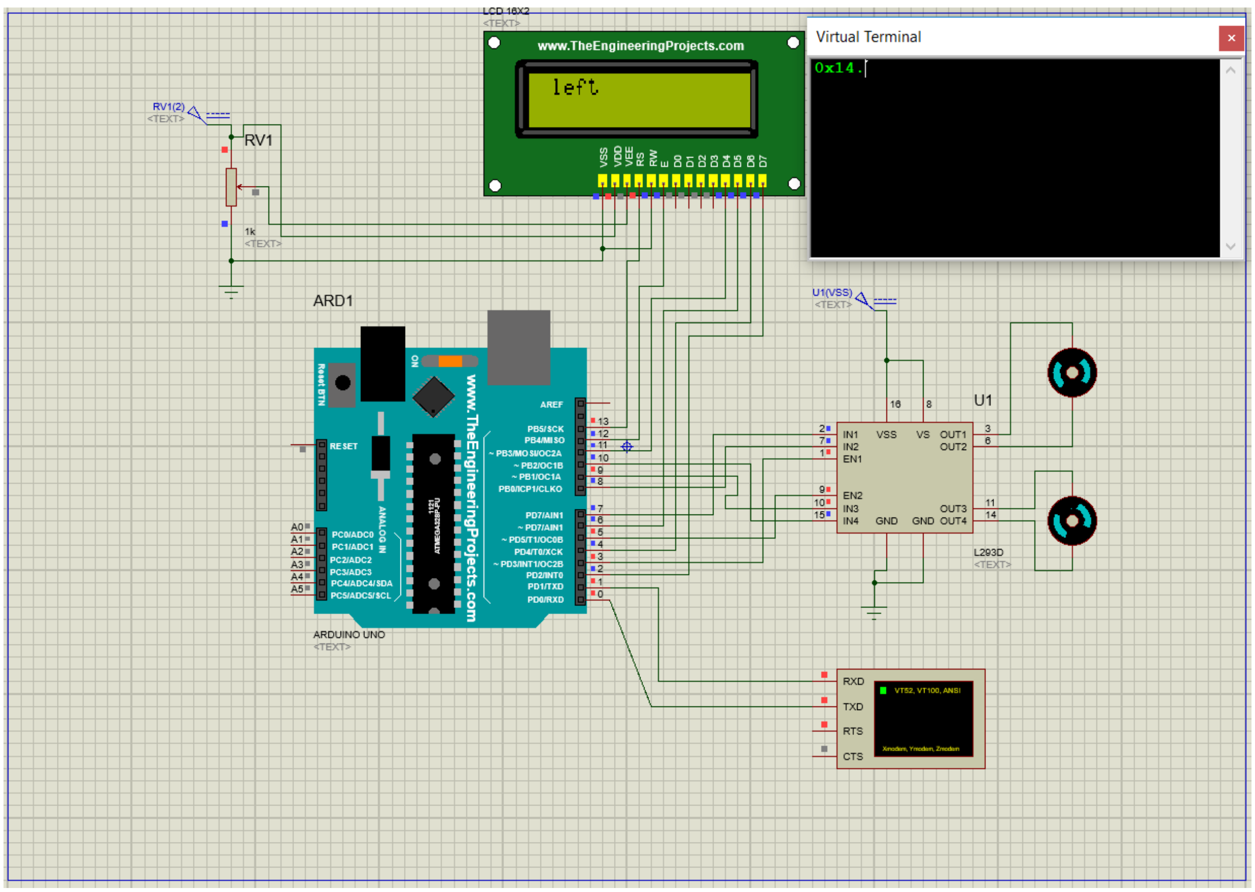


Figure 4.4: left condition

4.2.5 VR case (5)

When the simulation is run and the user enter (0x15) to the screen of the virtual terminal. As a result (stop) message will appear on the screen of the LCD. Simultaneously the Arduino will set the driver pins (EN1 and EN2) as an output and digital write (IN1, LOW); (IN2, LOW); (IN3, LOW); (IN4, LOW);

The voltage will be set to the corresponding value: 5V for HIGH, 0V (ground) for LOW.

The rotation of both motors will stop. The wheelchair system will go back to the stand by condition as shown in Figure (4.5).

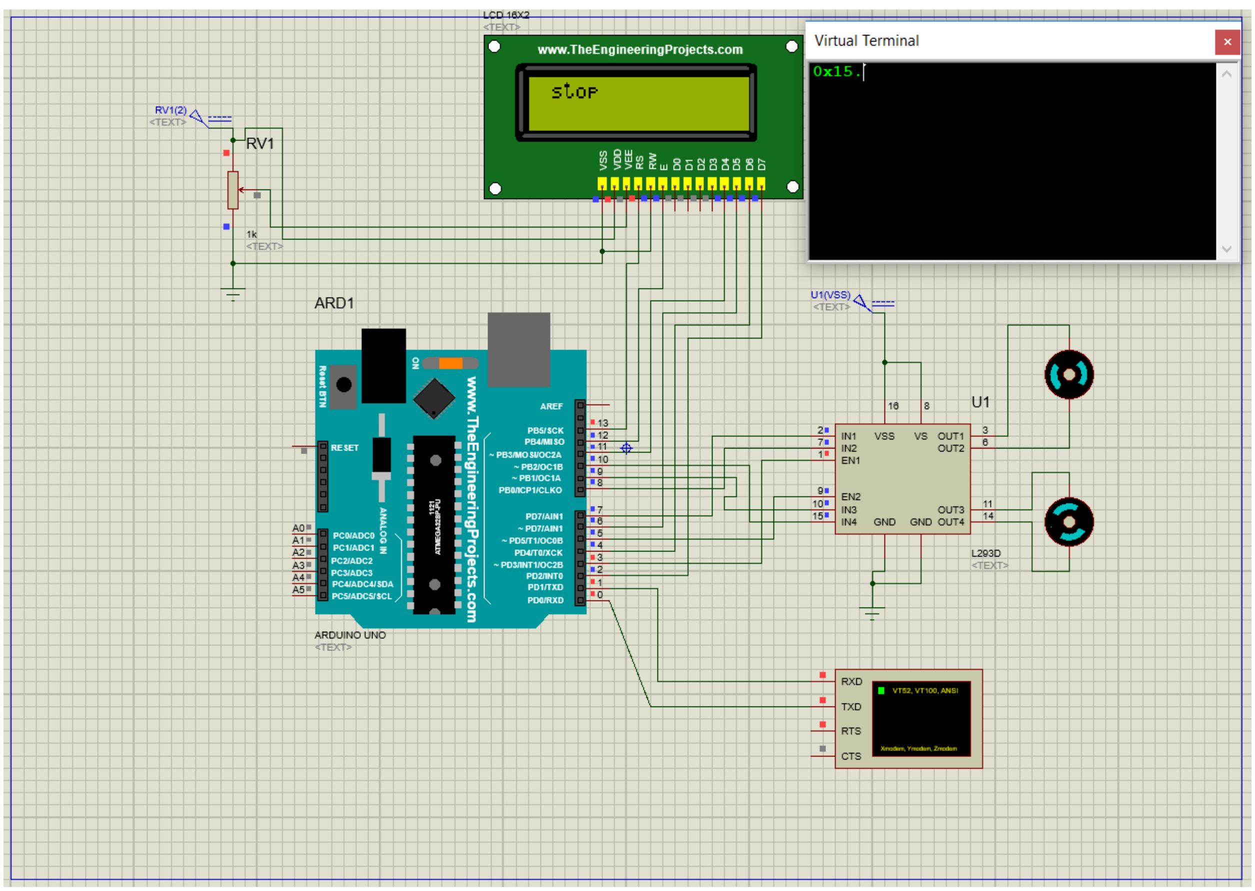


Figure 4.5: stop condition

4.2.6 GPS and GSM case

When the simulation is run the Arduino receives the location (longitude and latitude) of the user every certain time from the GPS and send it via message through the GSM to a recipient's mobile number with country code. The virtual terminal screen was used to display the AT commands as shown in Figure (4.6).

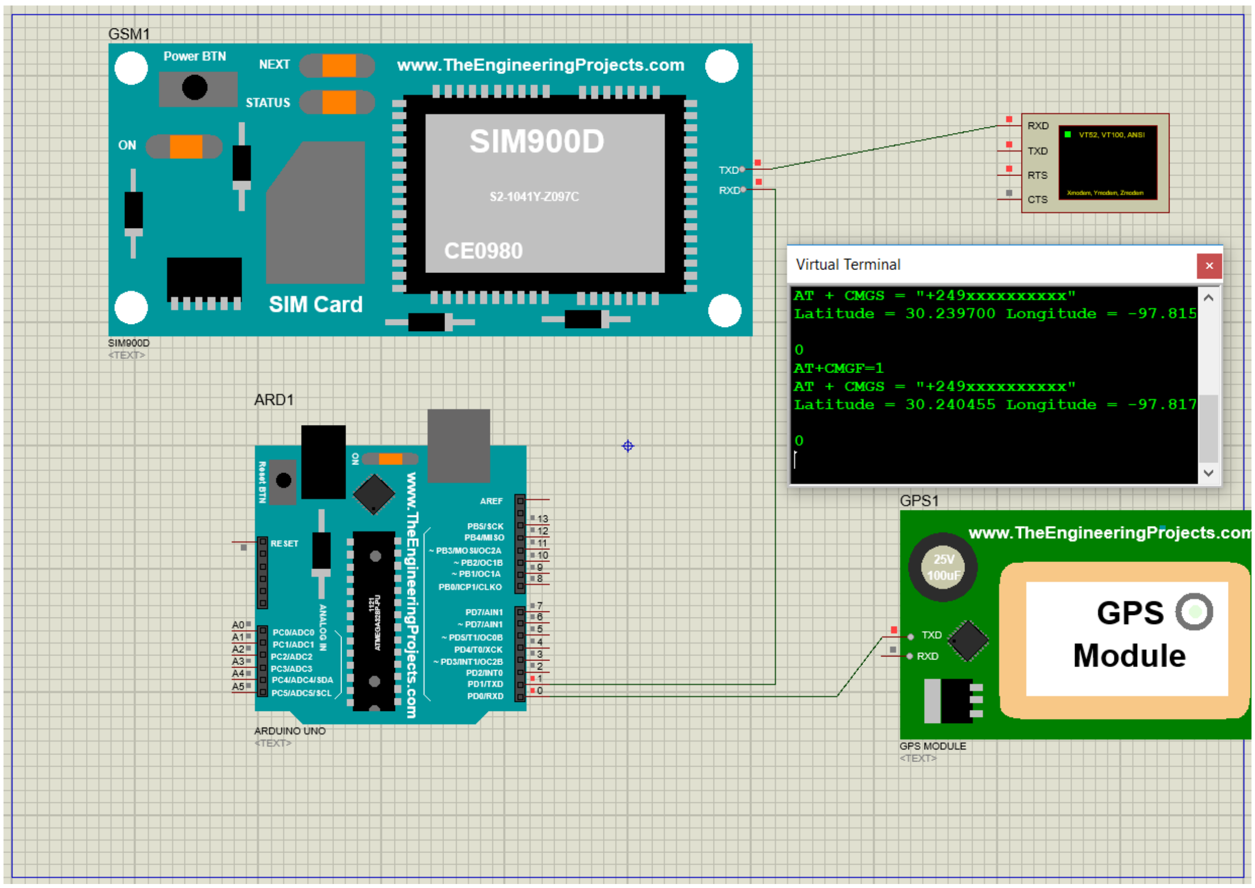


Figure 4.6: GPS and GSM

4.3 Hardware implementation

The hardware implementation has done in both mechanical & electronics parts.

4.3.1 Mechanical assembly

These components include the chair frame, right, left and front wheels, left and right DC motors, ball bearing for the front wheel, nuts and bolts. First the front wheel was put in place. Ball bearings, nuts and bolts were used. After that the right and left wheel ware affixed with the DC motors and mounted to the frame using nuts and bolts.

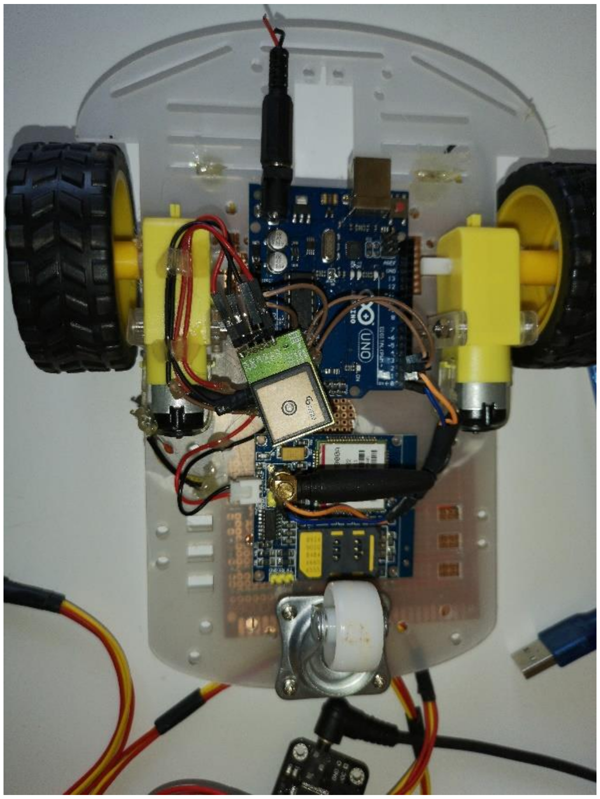


Figure 4.7: Mechanical parts

4.3.2 Electronic assembly

The electronic section implementation of the project is subdivided into several sub-circuits namely:

Battery Charging circuitry, Power Supply circuitry, Microcontrollers circuitry, VR module circuitry, [GPS, GSM] modules circuitry and Motors Driver circuitry.

All electronic implementation was carried out using a mini Vero board and all components where soldered onto the Vero board.

After the individual system components making up the design have been tested individually both through simulation and physically, the various sub systems were incorporated together and the final full system testing was carried out. Several persons of varying sounds were used to test how the overall system responded to various voice sounds and the system responded optimally.

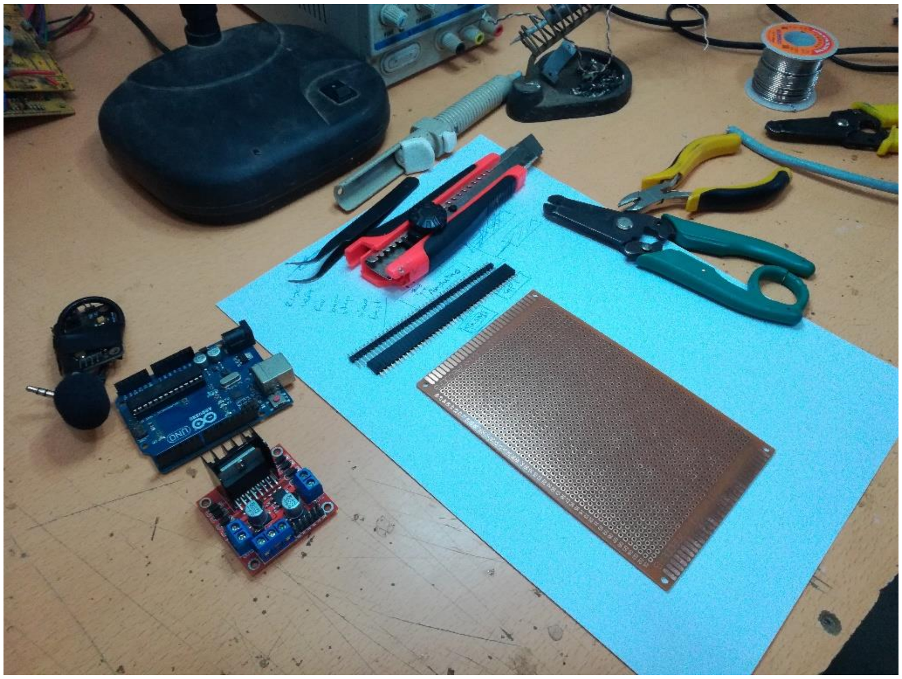


Figure 4.8: Electronic parts part 1

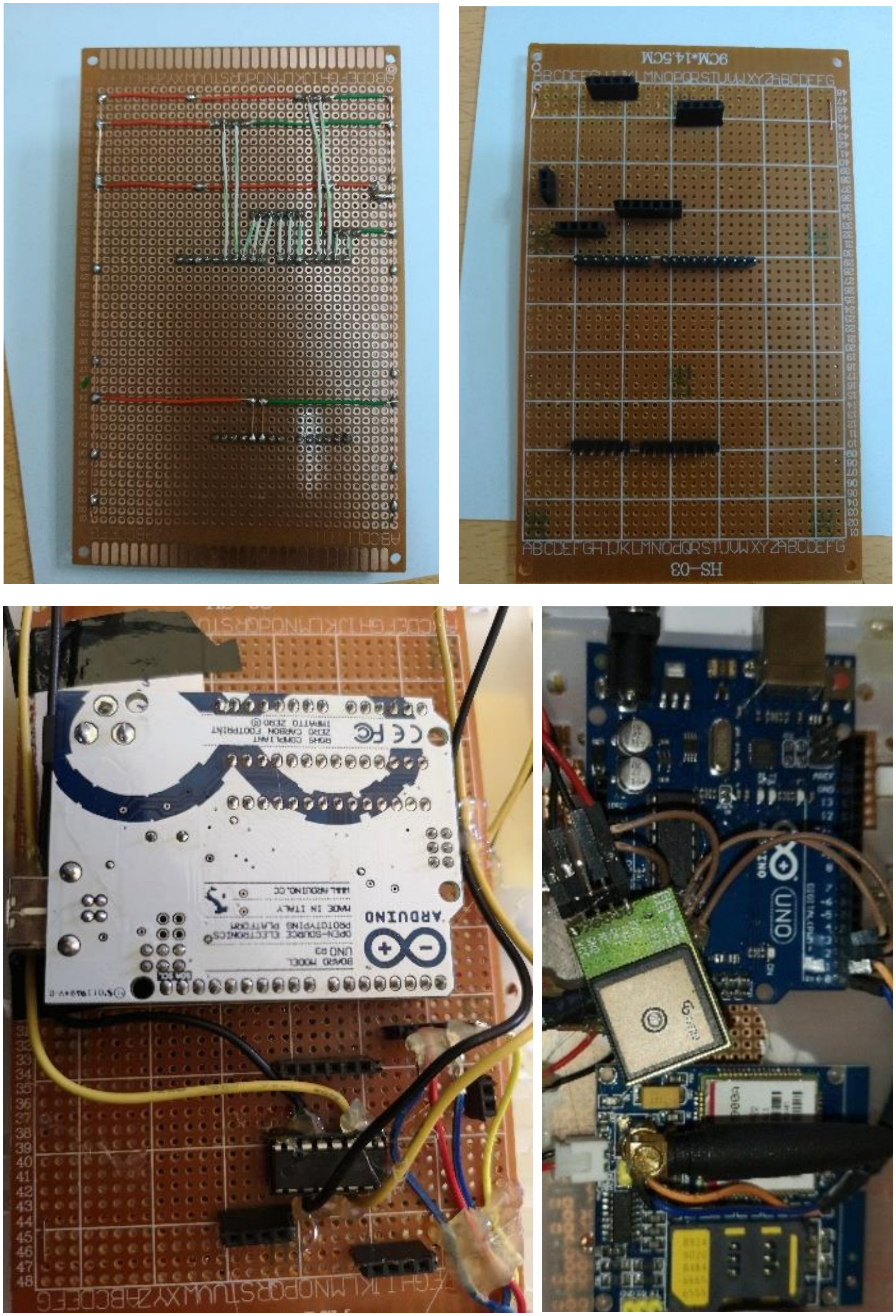


Figure 4.9: Electronic parts part 2

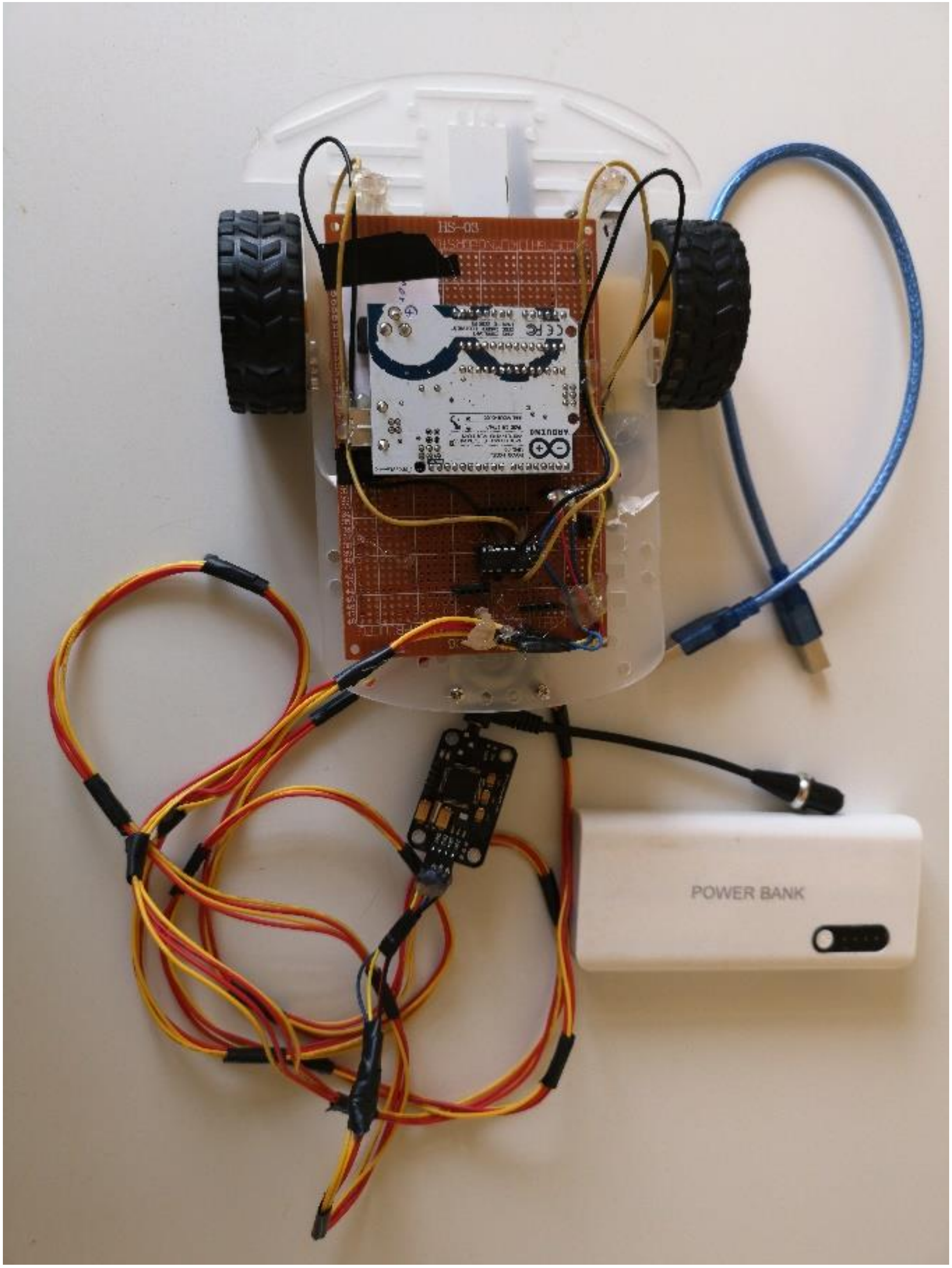


Figure 4.10: A real image for the design

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

As a conclusion, the objectives for this project were covered and achieved. This is done by implementing voice recognition processor chip for acquiring and distinguishing the command for controlling the motion of a wheelchair. The speed and direction of the wheelchair now can be selected using the specified commands. Thus the only thing needed to ride the wheelchair is to have voice. Besides that, the development of this project is done with less cost and affordable.

The design not only reduce the manufacture cost compare with present market one but also will give great competitive with other types of electrical wheelchair. However there are some improvements should be done to make it more reliable. This is outlined in the recommendation part. Improving this system, will directly enhance the life style of the disable people in the community. Lastly, this kind of system could contribute to the evolution of the wheelchair technology.

5.2 Recommendations

This project still has many improvements that should be done to improve it accuracy and reliability. There are some suggestions for the future research and development.

i. Adding the signal conditioning part which is consisting of a filter circuit. In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range.

ii. To apply sensors for security purpose. There so many types of sensors are available. However, many researches and testing with different algorithms have to be done in order to make it successful.

iii. Designing a controller to control the front wheels so that they will be self-centered each time the wheelchair stops.

REFERENCES

[1] Linda Fehr, MS. W. Edwin Langbein, Steven B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities," *Journal of Rehabilitation Research and Development,* 2000.

[2] H. L.KAMENETZ, "A Brief History of the Wheelchair," *Journal of the History of Medicine and Allied Sciences,* 1969.

[3] D.Hotchkiss, GROUND SWELL ON WHEELS, San Francisco, 1993.

[4] R. Hobso, "Concerning knowledge of mental states," *Journal of Medical Psychology,* 1990.

[5] Q.P.Ha,T.H.Tran and G. Dissanayake, "A Wavelet and Neural Network-based voice interface system for wheelchair control," *Intelligent System Technologies and Applications,* 2005.

[6] Rupinder Saini , Narinder Rana, "Comparison of Various Biometric Methods," *International Journal of Advances in Science and Technology (IJAST) ,* 2014.

[7] James Flaten, MN Space Grant Consortium, Christopher Gosch , Chad Serba, Introduction to Arduino.

[8] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis , "arduino," 2000. [Online]. Available: https://www.arduino.cc/.

[9] V.Ramya , B. Palaniappan , V. Sumathi, "GSM Based Embedded System for Remote Laboratory Safety Monitoring and Alerting," *International Journal of Distributed and Parallel Systems (IJDPS),* 2012.

[10] "futureelectronics," [Online]. Available: https://www.futureelectronics.com/.

Appendix A:

Arduino IDE

```
//Motor A
int enA = 10;
int in1 = 9;
int in2 = 8;
//Motor B
int enB = 5;
int in3 = 7;
int in4 = 6;
byte com = 0; //reply from voice recognition
//This will run only one time.
void setup()
{
Serial.begin(9600);
 pinMode(enA, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
delay(2000);
Serial.write(0xAA);
Serial.write(0x37);
delay(1000);
Serial.write(0xAA);
```

```
Serial.write(0x21);
}
void loop() // run over and over again
{
while(Serial.available())
{
com = Serial.read();
switch(com)
{
case 0x13:
{
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        analogWrite(enA, 10);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
        analogWrite(enB, 10);
}
break;

case 0x14:
{
digitalWrite(in1, LOW);
 analogWrite(enA, 10);
  analogWrite(enB, 10);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
```

```
  digitalWrite(in4, HIGH);
}
break;

case 0x11:
{
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
}
break;
case 0x12:
 {
        digitalWrite(in1,LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
  }
 break;
 case 0x15:
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
```

```
}
 break;
}
}
}
```

Appendix B:

VR manual

Voice Recognition Module

Speak to control (Arduino Compatible)

Introduction

The module could recognize your voice. It receives configuration commands or responds through serial port interface. With this module, we can control the car or other electrical devices by voice.

This module can store 15 pieces of voice instruction. Those 15 pieces are divided into 3 groups, with 5 in one group. First we should record the voice instructions group by group. After that, we should import one group by serial command before it could recognize the 5 voice instructions within that group. If we need to implement instructions in other groups, we should import the group first. This module is speaker independent. If your friend speaks the voice instruction instead of you, it may not identify the instruction. Please note that speaker independence requires strictly good MIC. The MIC we supply is not good enough for it to be speaker-independent.

Technica

Parameters

- *Voltage: 4.5-5.5V*
- *Current: <40mA*
- *Digital interface: 5V TTL level UART interface*
- *Analog interface: 3.5mm mono-channel microphone connector + microphone pin interface*
- *Size: 30mm x 47.5mm*
- *Recognition accuracy: 99% (under ideal environment)*

Serial Command

This module can be configured by sending commands via serial port. Configuration will be not erased after powered off.

Its interface is 5V TTL. The serial data format: 8 data bits, no parity, 1 stop bit. The default baud rate is 9600 and baud rate can be changed.

Command format is "Head + Key". "Head" is a 0xaa, and "Key" is as follows:

Key (HEX format)	Description	Respond in Common Mode	Respond in Compact Mode
0x00	Enter into "Waiting" state	"Waiting! | n" : successful "ERROR! | n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x01	Delete the instructions of group 1	"Group1 Deleted ! | n" : successful "ERROR! | n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x02	Delete the instructions of group 2	"Group2 Deleted ! | n" : successful "ERROR! | n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x03	Delete the instructions of group 3	"Group3 Deleted ! | n" : successful "ERROR! | n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x04	Delete the instructions of all the 3 groups	" All Groups Deleted !|n " : successful "ERROR! | n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x11	Begin to record instructions of group 1	"ERROR! | n" : Instruction error "START | n" : Ready for recording, you can speak now "No voice | n" : no voice detected "Again | n" : Speak the voice instruction again. Do not speak until getting the START message "Too loud | n" : Too loud to record "Different | n" : voice instruction confirming failed. Voice for the second chance is different with the first one. "Finish one | n" : recording one voice instruction successfully "Group1 finished! | n" : finish recording group 1	0xe0 : Instruction error 0x40 : Ready for recording, you can speak now 0x41 : no voice detected 0x42 : Speak the voice instruction again. Do not speak until getting the START message 0x43 : Too loud to record 0x44 : voice instruction confirming failed. Voice for the second chance is different with the first one. 0x45 : recording one voice instruction successfully 0x46 : finish recording group 1
0x12	Begin to record instructions of group 2	"ERROR! | n" : Instruction error "START | n" : Ready for recording, you can speak now "No voice | n" : no voice detected "Again | n" : Speak the voice instruction again. Do not speak until getting the START message "Too loud | n" : Too loud to record "Different | n" : voice instruction confirming failed. Voice for the second chance is different with the first one. "Finish one | n" : recording one voice instruction successfully "Group2 finished! | n" : finish recording group 2	0xe0 : Instruction error 0x40 : Ready for recording, you can speak now 0x41 : no voice detected 0x42 : Speak the voice instruction again. Do not speak until getting the START message 0x43 : Too loud to record 0x44 : voice instruction confirming failed. Voice for the second chance is different with the first one. 0x45 : recording one voice instruction successfully 0x47 : finish recording group 2
0x13	Begin to record instructions of group 3	"ERROR! | n" : Instruction error "START | n" : Ready for recording, you can speak now "No voice | n" : no voice detected "Again | n" : Speak the voice instruction again. Do not speak until getting the START message "Too loud | n" : Too loud to record	0xe0 : Instruction error 0x40 : Ready for recording, you can speak now 0x41 : no voice detected 0x42 : Speak the voice instruction again. Do not speak until getting the START message

<table>
<tr><td></td><td></td><td>"Different | n" : voice instruction confirming failed. Voice for the second chance is different with the first one.
"Finish one | n" : recording one voice instruction successfully
"Group3 finished! | n" : finish recording group 3</td><td>0x43 : Too loud to record
0x44 : voice instruction confirming failed. Voice for the second chance is different with the first one.
0x45 : recording one voice instruction successfully
0x48 : finish recording group 3</td></tr>
<tr><td>0x21</td><td>Import group 1 and be ready for voice instruction</td><td>"Group1 Imported !\n" : Successful
"ERROR! | n" : Instruction error
"Import failed !\n" : Importing voice group failed</td><td>0xcc : Successful
0xe0 : Instruction error
0xe1 : Importing voice group failed</td></tr>
<tr><td>0x22</td><td>Import group 2 and be ready for voice instruction</td><td>"Group2 Imported !\n" : Successful
"ERROR! | n" : Instruction error
"Import failed !\n" : Importing voice group failed</td><td>0xcc : Successful
0xe0 : Instruction error
0xe1 : Importing voice group failed</td></tr>
<tr><td>0x23</td><td>Import group 3 and be ready for voice instruction</td><td>"Group3 Imported !\n" : Successful
"ERROR! | n" : Instruction error
"Import failed !\n" : Importing voice group failed</td><td>0xcc : Successful
0xe0 : Instruction error
0xe1 : Importing voice group failed</td></tr>
<tr><td>0x24</td><td>Query the recorded group</td><td>"Used group:0\n" : No group is recorded
"Used group:1\n" : Group 1 is recorded
"Used group:2\n" : Group 2 is recorded
"Used group:3\n" : Group 3 is recorded
"Used group:12\n" : Group 1 and Group 2 are recorded
"Used group:13\n" : Group 1 and Group 3 are recorded
"Used group:23\n" : Group 2 and Group 3 are recorded
"Used group:123\n" : All the 3 groups are recorded
"ERROR! | n" : Instruction error</td><td>0x00 : No group is recorded
0x01 : Group 1 is recorded
0x02 : Group 2 is recorded
0x04 : Group 3 is recorded
0x03 : Group 1 and Group 2 are recorded
0x05 : Group 1 and Group 3 are recorded
0x06 : Group 2 and Group 3 are recorded
0x07 : All the 3 groups are recorded
0xe0 : Instruction error</td></tr>
<tr><td>0x31</td><td>Change the baud rate to 2400bps</td><td>"Baud: 2400\n" : Successful
"ERROR! | n" : Instruction error</td><td rowspan="7">0xcc : successful
0xe0 : Instruction error</td></tr>
<tr><td>0x32</td><td>Change the baud rate to 4800bps</td><td>"Baud: 4800\n" : Successful
"ERROR! | n" : Instruction error</td></tr>
<tr><td>0x33</td><td>Change the baud rate to 9600bps</td><td>"Baud: 9600\n" : Successful
"ERROR! | n" : Instruction error</td></tr>
<tr><td>0x34</td><td>Change the baud rate to 19200bps</td><td>"Baud: 19200\n" : Successful
"ERROR! | n" : Instruction error</td></tr>
<tr><td>0x35</td><td>Change the baud rate to 38400bps</td><td>"Baud: 38400\n" : Successful
"ERROR! | n" : Instruction error</td></tr>
<tr><td>0x36</td><td>Switch to Common Mode</td><td>"Common Mode\n" : Successful
"ERROR! | n" : Instruction error</td></tr>
<tr><td>0x37</td><td>Switch to Compact Mode</td><td>"Compact Mode\n" : Successful
"ERROR! | n" : Instruction error</td></tr>
<tr><td>0xbb</td><td>Query version information</td><td>Version information</td><td>No respond</td></tr>
</table>

If you want to modify the serial baud rate to 38400, you need to send command: ***0xaa35****. If successful, it will return "Baud: 38400 | n"(in Common Mode) or* ***0xcc*** *(in Compact Mode). The baud rate is set to 38400.*

The main difference between Compact Mode and Common Mode is the returning message. Common Mode response is long string but Compact Mode response is a byte. For example, after sending ***0xaa04*** *to delete all the contents of the 3 groups, in Common*

*Mode it will return **"All Groups Deleted! | n"**, but in Compact Mode it will return a concise bytes such as **0xcc** which means a successful operation.*

For the first-time use, we need to do some configuration:

1. *Select the serial baud rate (default 9600)*
2. *Select the communication mode: Common Mode or Compact Mode*
3. *Recording five instructions of the first group(or 2nd or 3rd as required)*
4. *Import the group you need to use (only recognize 5 instructions within one group at the same time)*

*After all the setting above, you can speak or send voice instruction to it. If identified successfully, result will be returned via serial port in the format: group number + command number. For example, return **Result: 11** (Compact mode returns 0x11) means identified the first command of group 1.*

If voice instruction is recorded, each time after you power it on, you need to import the group before letting it identify voice instructions.

LED

Recording stage:

1. *Record indication: D1 (RED) flashes 3 times within the 600ms, then off for400ms, and then flashes quickly for 4 times within 600ms. Now the recording indication is over.*
2. *Begin to speak: D1 (RED) is off for 400ms, and then is on. Voice during the time while D1 (RED) is on will be recorded by this module.*
3. *Recording a voice instruction successfully for the first time: D1 (RED) off, D2 (ORANGE) on for 300ms.*
4. *Recording a voice instruction successfully for the first time: D1 (RED) off, D2 (ORANGE) on for 700ms.*
5. *Recording failure: D2 (ORANGE) flashes 4 times within the 600ms. In cases that voice instructions detected twice don't match, or the sound is too large, or there is no sound, recording will fail. You need to start over the recording process for that instruction.*

Waiting mode:

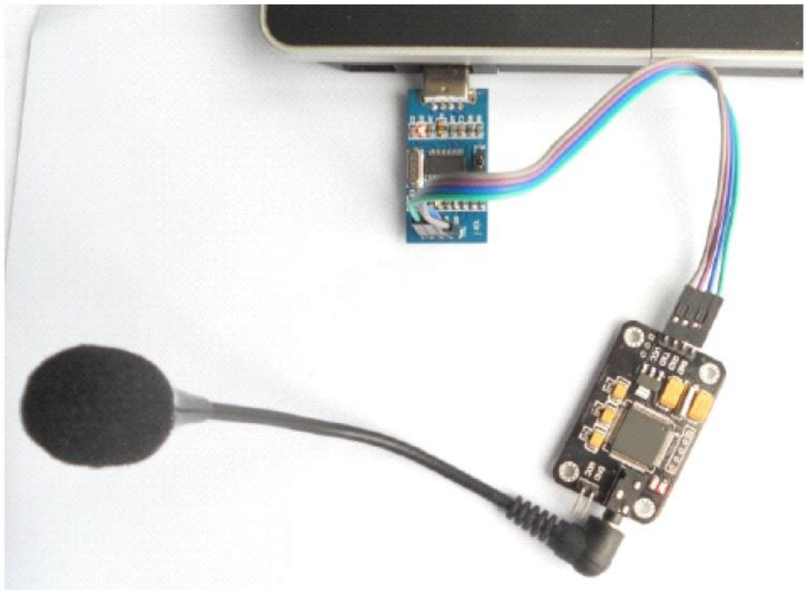
In waiting mode, D2 (ORANGE) is off, and D1 (RED) is on for 80ms every other 200ms, fast flashing. In this mode, it doesn't recognize voice command, only waiting for serial commands.

Recognition stage:

In identification stage, D2 (ORANGE) is off, and D1 (RED) is on for 100ms every other 1500ms, slow flashing. In this stage, this module is processing received voice signal, and if matching, it will send the result immediately via serial port.

Recording

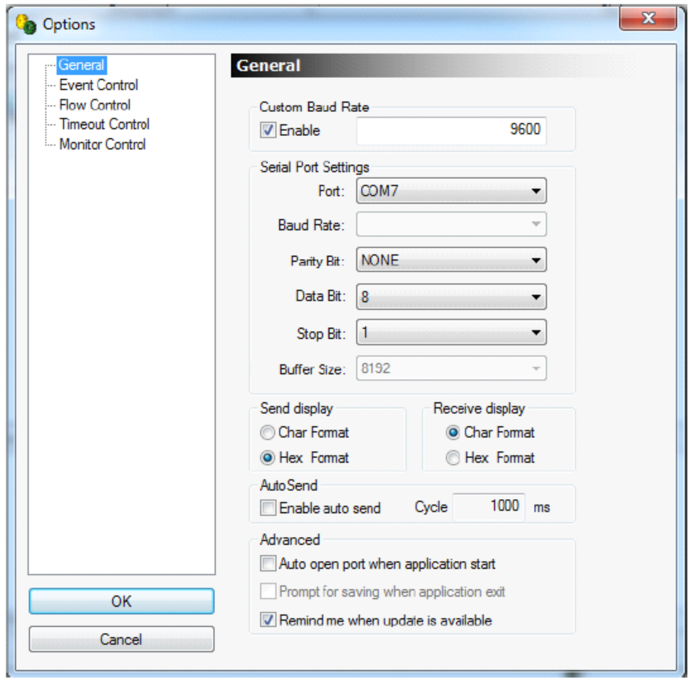
Before using it, we have to record voice instructions. Each voice instruction has the maximum length of 1300ms, which ensures that most words can be recorded. Once you start recording, you can't stop the recording process until you finish all the 5 voice instructions recording of one group. Also, once you start recording, the previous content of that group will be erased. In recording stage, this module doesn't reply to any other serial commands.



LED will flash to indicate state. Please refer to the LED part.

First, you need a serial tool. Here we use AccessPort (Download page).

1. *Serial port setting:*
    - *Baud rate: 9600*
    - *Parity bit: None*
    - *Data bit: 8*
    - *Stop bit: 1*
    - *Send format: Hex*
    - *Receive format: Char*



Send commands