

WHEEL CHAIR CONTROLLED BY HEAD MOTION USING ACCELEROMETER

Submitted in partial fulfilment of requirements for the award of

Bachelor of Engineering

in

Electronics and Communication Engineering

by

P.V.D.PRASAD (39130358)

R.JESWANTH(39130368)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

OCTOBER – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of RAMINENI JESWANTH (39130368), P.V.D. PRASAD (3913058) who carried out the project entitled **“Wheel Chair Controlled by Head Motion using Accelerometer”**

Internal Guide

Dr. M. R. EBENEZAR JEBARANI, M.E., Ph.D.,

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

RAMINENI JESWANTH (39130368), P.V.D.PRASAD(3913058) here by declare that the Project Report entitled “**Wheelchair Controlled by Head Movement**” done by us under the guidance of Dr. M.R.EBENEZER JEBARANI M.E., Ph.D., at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Electronics and communication Engineering.

DATE:Chennai

PLACE:23.04.23

1. 
2. 

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr.N.M.NANDHITHA, M.E.,Ph.D.**, Prof & Dean of school of Electrical and Electronics Engineering and **Dr.T.RAVI, M.E.,Ph.D.**, Heads of the Department, Department of Electronics and communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.M.R.EBENEZAR JEBARANI, M.E.,Ph.D.** for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and communication engineering who were helpful in many ways for the completion of the project.

TABLE OF CONTENTS

<i>CHAPTER NO</i>	<i>TITLE</i>	<i>PAGE NO.</i>
	ABSTRACT	6
1	INTRODUCTION	8
2	LITERATURE SURVEY	15
3	EXISTING SYSTEM AND PROPOSED SYSTEM	30
4	CONCLUSION AND FUTURE SCOPE	32

ABSTRACT:

The challenging problem faced by the paralyzed people is their independent mobility. They need an external help to perform their daily activities. Electric wheelchairs are designed to aid paraplegics. Unfortunately, these cannot be used by persons with higher degree of impairment, such as quadriplegics, i.e. persons that, due to age or illness, cannot move any of the body parts, except of the head. The main objective of this project is to provide an automated system for disabled people. The wheel chair will work based on the head movement of the user. The recognized gestures are used to generate motion control commands to the controller so that it can control the motion of the wheel chair according to the user intention. Design and development of Head motion-controlled wheelchair has been achieved using accelerometer sensors and microcontroller. The system is implemented practically and works well. The ACCELEROMETER senses the change in direction of head and accordingly the signal is given to microcontroller. Depending on the direction of the Acceleration, microcontroller controls the wheel chair directions like LEFT, RIGHT, FRONT, and BACK with the aid of DC motors.

Chapter 1

Introduction:

Quadriplegics are persons who are not able to use any of the extremities. The reasons for such decreased motion possibilities can be different: stroke, arthritis, high blood pressure, degenerative diseases of bones and joints and cases of paralysis and birth defects. Also, quadriplegia appears as a consequence of accidents or age. The patients with such severe disabilities are not able to perform their everyday actions, such as: feeding, toilette usage and movement through space. Depending on the severity of the disability, a patient can retain freedom of movement to a certain level by using different medical devices . Mobility has become very important for a good quality of life. Designing a system with independent mobility for such disabled people is our aim in this project.



Figure 1.1 – Clip view assumption of Wheelchair with Sensor Mounts

Problem Statement :

1. The developed wheelchair is best suited to meet the challenges of physically challenged people and also provides them with easy control of the wheelchair.
2. To analyze the EEG signal in terms of attention and meditation level by using their peak Microelectromechanical Systems (MEMS) based accelerometer sensor is used to direct the movement of the device.
3. The wheelchair may move forward, backward, left, and right directions based on the output signal of the MEMS device.
4. In addition, the wheelchair can also be controlled using a joystick and also with a mode switch that has been interfaced with the controller

Objectives:

1. With the advancement of science and technology innovative mechanisms adopted to provide assistance for physically challenged people.
2. In recent years, a growing interest in Brain-Computer Interfaces systems for medical and multimedia applications is observed.
3. BCI is a device that acts as a first mode to provide a direct interface between the human brain and computer.
4. The users just need to think of movement in order to drive the system. Therefore, the usage of BCI is one of the prominent devices for enabling the severely impaired user to control a 6 wheelchair.
5. The Mems accelerometer operates with head or hand movement, taking motion as an input signal for the movement of the wheelchair in a particular direction.
6. The third method for the movement of a wheelchair can be controlled manually by the joystick.

Scope of the Project:

A smart wheelchair aims to provide aid to handicapped and physically challenged persons by providing them with some sort of mobility that would greatly help them. A smart wheelchair consists of a major controller unit that allows the user to provide input in the form of a joystick or accelerometer or electroencephalogram (EEG). Brain-computer interface provides a communication method between physical devices and the human brain. The micro-electromechanical systems (MEMS) accelerometer can be used to effectively translate head movement into computer-interpreted signals. The controller unit then synthesizes the command and takes the required action so as to move the wheelchair to a particular position. A major task of designing a smart wheelchair control is to produce sufficient commands rapidly and accurately including left and right turn, forward and backward motions, stopping, acceleration and deceleration. The facilities provided by a typical wheelchair require people with disabilities to have intact manipulation abilities like using a joystick and operating the vehicle. However, many people with disabilities cannot manipulate mechanical devices and thus are unable to use a typical wheelchair.

Embedded system implementation

Introduction:

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, and store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc.

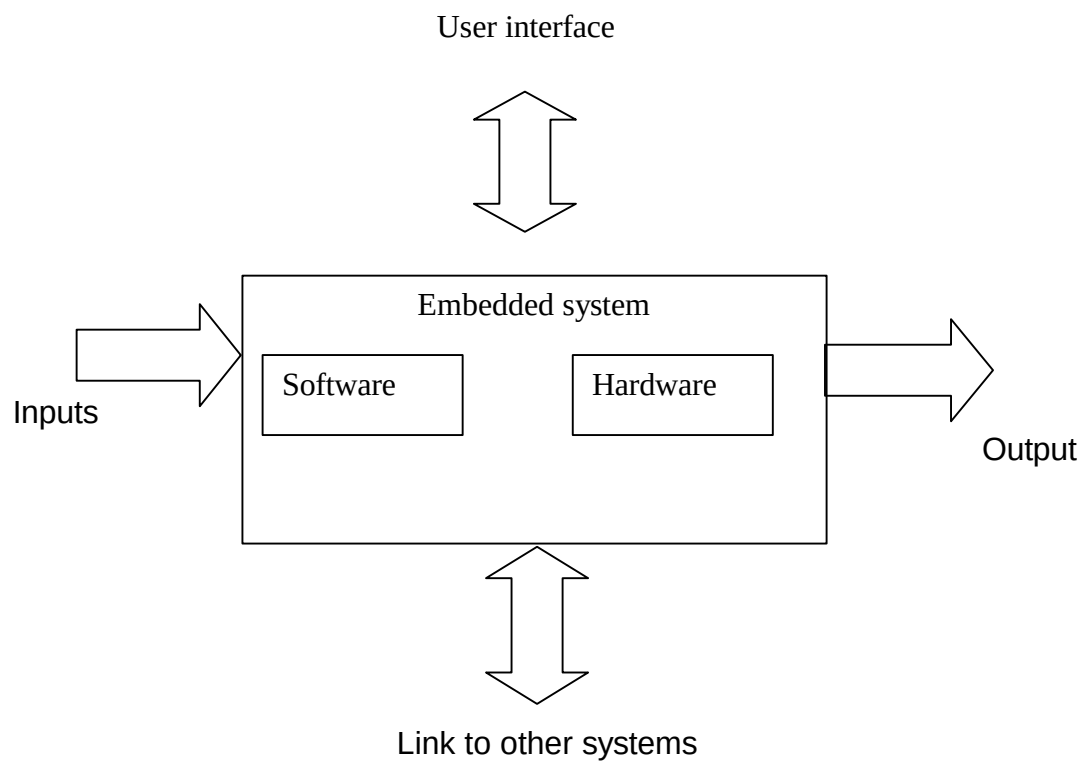
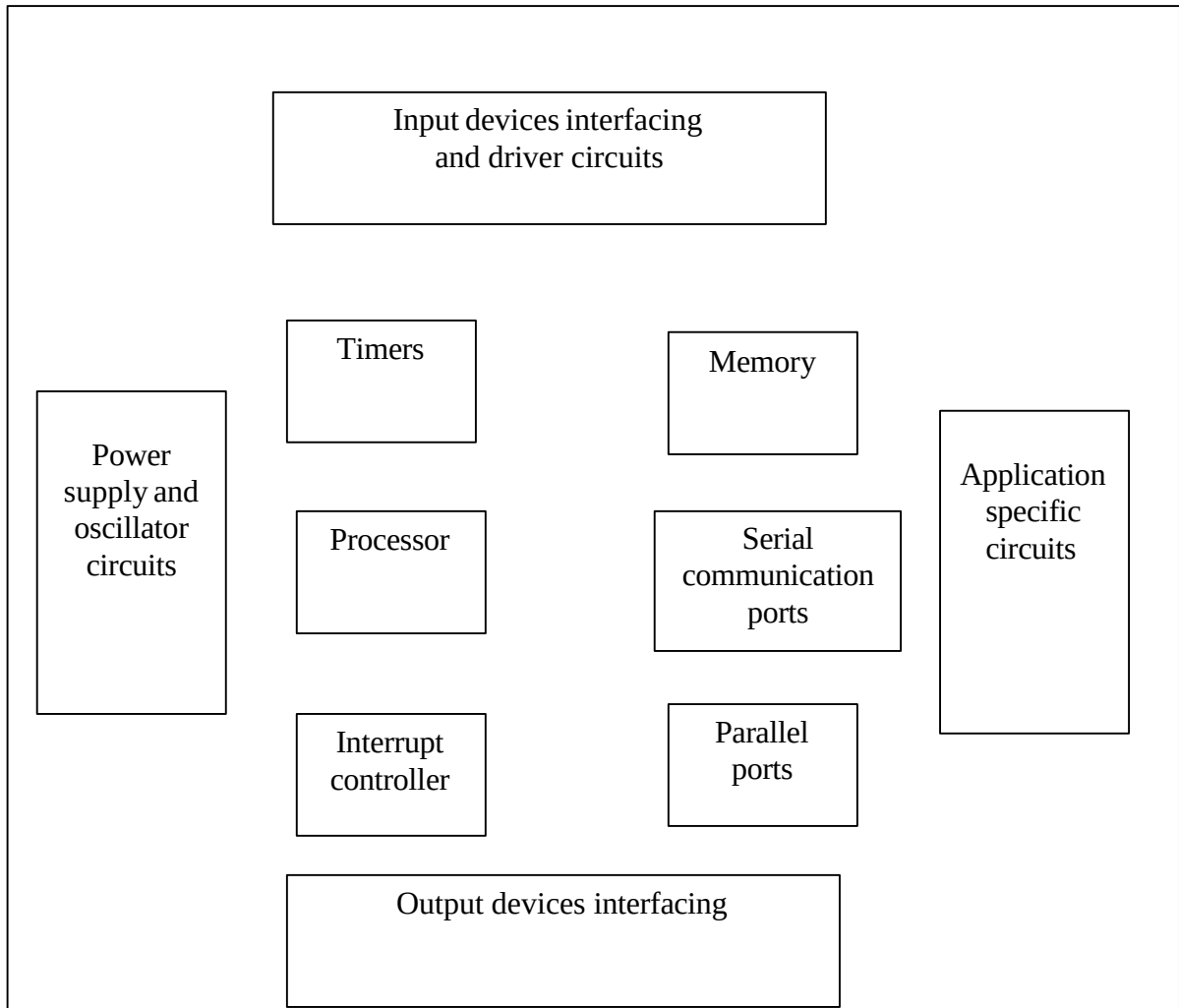


Fig: Overview of embedded system

Embedded system:

Embedded system includes mainly two sections, they are

1. Hardware
2. Software



Embedded System Hardware:

As with any electronic system, an embedded system requires a hardware platform on which it performs the operation. Embedded system hardware is built with a microprocessor or microcontroller. The embedded system hardware has elements like input output (I/O) interfaces, user interface, memory and the display. Usually, an embedded system consists of:

- Power Supply

- Processor
- Memory
- Timers
- Serial communication ports
- Output/Output circuits
- System application specific circuits

Embedded systems use different processors for its desired operation. Some of the processors used are

1. Microprocessor
2. Microcontroller
3. Digital signal processor

Microprocessor vs. Microcontroller

Microprocessor

- **CPU** on a chip.
- We can attach required amount of ROM, RAM and I/O ports.
- Expensive due to external peripherals.
- Large in size
- general-purpose

Microcontroller

- **Computer** on a chip
- fixed amount of on-chip ROM, RAM, I/O ports
- Low cost.
- Compact in size.
- Specific –purpose

Embedded System Software:

The embedded system software is written to perform a specific function. It is typically written in a high level format and then compiled down to provide code that can be lodged within a non-volatile memory within the hardware. An embedded system software is designed to keep in view of the three limits:

- Availability of system memory
- Availability of processor's speed
- When the system runs continuously, there is a need to limit power dissipation for events like stop, run and wake up.

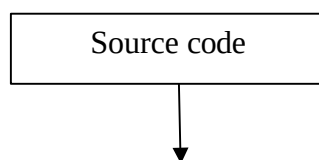
Bringing software and hardware together for embedded system:

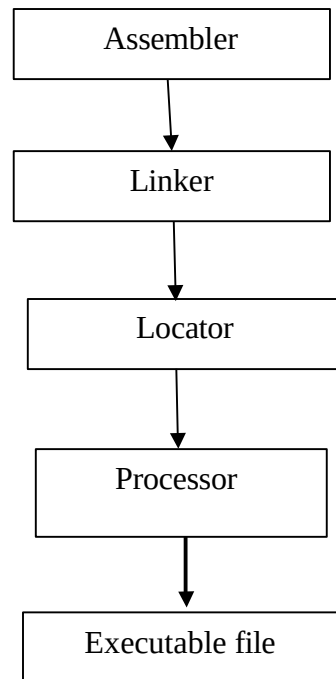
To make software to work with embedded systems we need to bring software and hardware together .for this purpose we need to burn our source code into microprocessor or microcontroller which is a hardware component and which takes care of all operations to be done by embedded system according to our code.

Generally we write source codes for embedded systems in assembly language, but the processors run only executable files. The process of converting the source code representation of your embedded software into an executable binary image involves three distinct steps:

1. Each of the source files must be compiled or assembled into an object file.
2. All of the object files that result from the first step must be linked together to produce a single object file, called the re-locatable program.
3. Physical memory addresses must be assigned to the relative offsets within the re-locatable program in a process called relocation.

The result of the final step is a file containing an executable binary image that is ready to run on the embedded system.





Flow of burning source code to processor

Applications:

Embedded systems have different applications. A few select applications of embedded systems are smart cards, telecommunications, satellites, missiles, digital consumer electronics, computer networking, etc.

Embedded Systems in Automobiles

- Motor Control System
- Engine or Body Safety
- Robotics in Assembly Line
- Mobile and E-Com Access

Embedded systems in Telecommunications

- Mobile computing
- Networking
- Wireless Communications

Embedded Systems in Smart Cards

- Banking
- Telephone
- Security Systems

Implementation flow:

Stage 1:

Considering the problems of existing methods and giving solution to that problem by considering the basic requirements for our proposed system

Stage 2:

Considering the hardware requirement for the proposed system

For this we need to select the below components:

1. Microcontroller
2. Inputs for the proposed system (ex: sensors, drivers etc..)
3. Outputs (ex: relays, loads)

Stage 3:

After considering hardware requirements, now we need to check out the software requirements. Based on the microcontroller we select there exists different software for coding, compiling, debugging. we need to write source code for that proposed system based on our requirements and compile, debug the code in that software .

After completing all the requirements of software and hardware we need to bring both together to work our system. For this we need to burn our source code into microcontroller, after burning our source code to microcontroller then connect all input and output modules as per our requirement.

Chapter 2

LITERATURE SURVEY

- Wheelchair Control by Head Motion Using Accelerometer Abhishek Gupta (2016) This system is an automatic head tilt movement controlled wheelchair that could operate

in any direction using head movements, i.e. Forward, Backward, Left and Right. It stops when the person does not tilt his head in any direction. In this paper, a microcontroller system that enables standard electric wheelchair control by head motion is developed. The project describes a wheelchair for physically disabled people developed using head motion and accelerometer sensor which is interfaced with DC motors. The prototype of the wheelchair is built using a PIC micro-Controller, chosen for its low cost, in addition to its features of easy erasing and programming.

- Wheelchair Control by Head Motion Aleksandar Pajkanović (2013)

Electric wheelchairs are designed to aid paraplegics. Unfortunately, these can not be used by persons with higher degree of impairment, such as quadriplegics, i.e. persons that, due to age or illness, can not move any of the body parts, except of the head. Medical devices designed to help them are very complicated, rare and expensive. In this paper a microcontroller system that enables standard electric wheelchair control by head motion is presented. The system comprises electronic and mechanic components. A novel head motion recognition technique based on accelerometer data processing is designed. The wheelchair joystick is controlled by the system's mechanical actuator. The system can be used with several different types of standard electric wheelchairs. It is tested and verified through an experiment performed within this paper.

- Head Motion Controlled Wheelchair V.Purushothaman (2019)

Traditional Wheelchairs though have certain limitations with the flexibility, heavy weight of the chair and limited functions. Tremendous developments have been made in the field of wheelchair technology. Be that as it may, even these noteworthy advancements couldn't help the quadriplegics to explore wheelchair freely. Medical gadgets intended to support the Paraplegic and Quadriplegic patients are exceptionally muddled, once in a while accessible and costly. We go for planning a straightforward financially effective programmed wheelchair utilizing MEMS technology for quadriplegics with head and neck versatility. The control system interprets the situation of the user's head into speed and directional control of the wheelchair. The system is divided into two main units: MEMS Sensor and programmed Arduino Micro Controller. The MEMS sensor senses the change in direction of head and likewise the signal is given to microcontroller. Depending on the direction of the Acceleration, microcontroller controls the wheel chair directions like LEFT, RIGHT, FRONT, and BACK with the aid of DC motors.

Chapter 3

EXISTING SYSTEM

In existing we are using wired robot which will have wires and some buttons to control a robot. For this kind of robots manual work is needed because one person need to go with the robot while operating as it is connected with robot.

Drawbacks of Existing System:

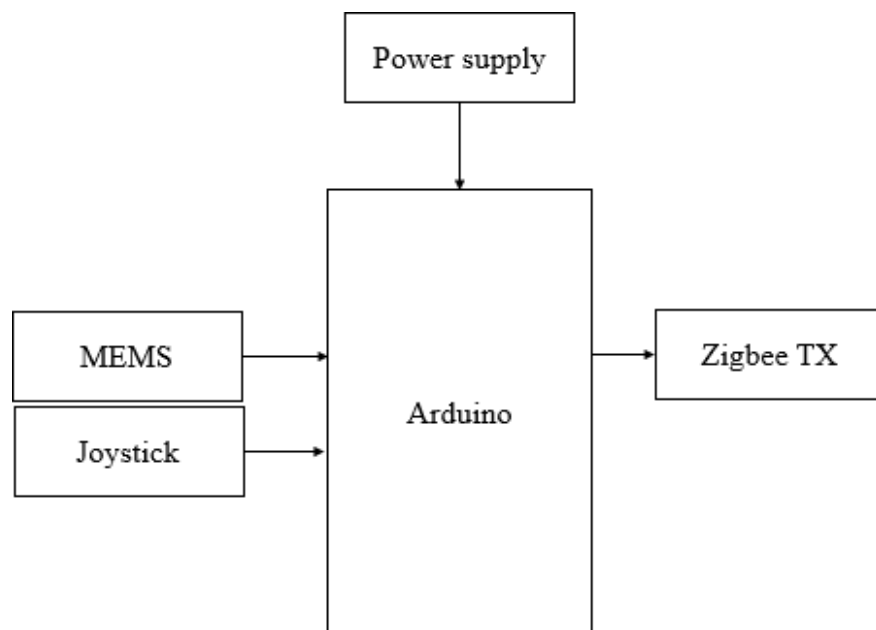
- Wired Systems
- Very Short Distance
- Manual assistance is required even if it is small distances also.

PROPOSED METHOD

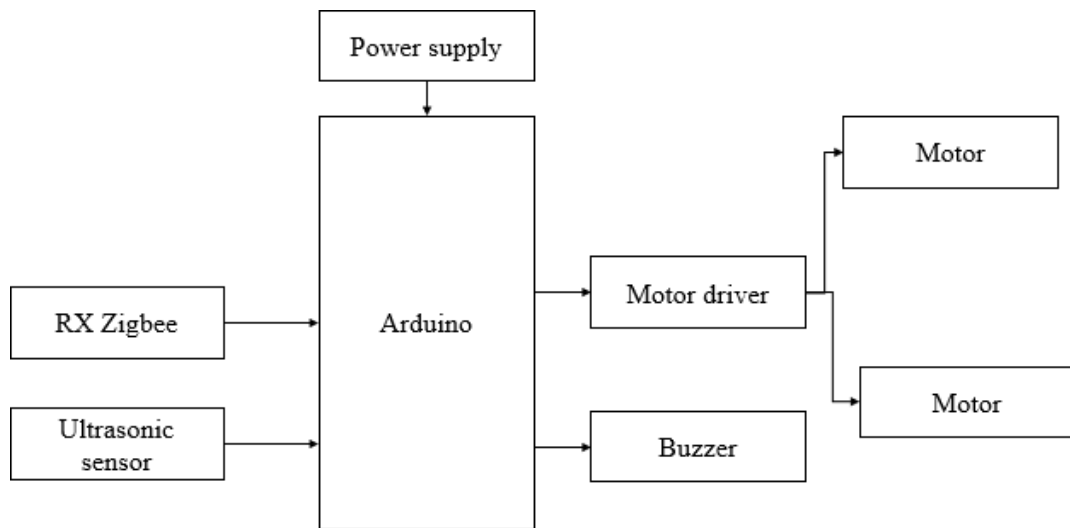
In the proposed method we are using two controllers as TX and RX. Bothe the TX and RX has TX Zigbee module and RX Zigbee module respectively. TX has Arduino, MEMS sensor and Zigbee TX Module, MEMS will detect the head motion and will transmit to the receiver side through Zigbee module. Based on the commands TX send the receiver will move the chair. Ultrasonic sensor will detect the obstacles, whenever the obstacle detected then buzzer will give the alerts.

Block diagram:

TX:



RX:



HARDWARE REQUIREMENTS

Arduino:

Arduino Uno is a very valuable addition in the electronics that consists of USB interface, 14 digital I/O pins, 6 analog pins, and Atmega328 microcontroller. It also supports serial communication using Tx and Rx pins.

There are many versions of Arduino boards introduced in the market like Arduino Uno, Arduino Due, Arduino Leonardo, Arduino Mega, however, most common versions are Arduino Uno and Arduino Mega. If you are planning to create a project relating to digital electronics, embedded system, robotics, or IoT, then using Arduino Uno would be the best, easy and most economical option.



Arduino Uno



Arduino Due



Arduino Leonardo



Arduino Mega

It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality.

The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and required some basic skills to learn it. It can be programmed using C and C++ language.

Some people get confused between **Microcontroller and Arduino**. While former is just an on system 40 pin chip that comes with a built-in microprocessor and later is a board that comes with the microcontroller in the base of the board, bootloader and allows easy access to input-output pins and makes uploading or burning of the program very easy.

Microcontroller



Arduino Board



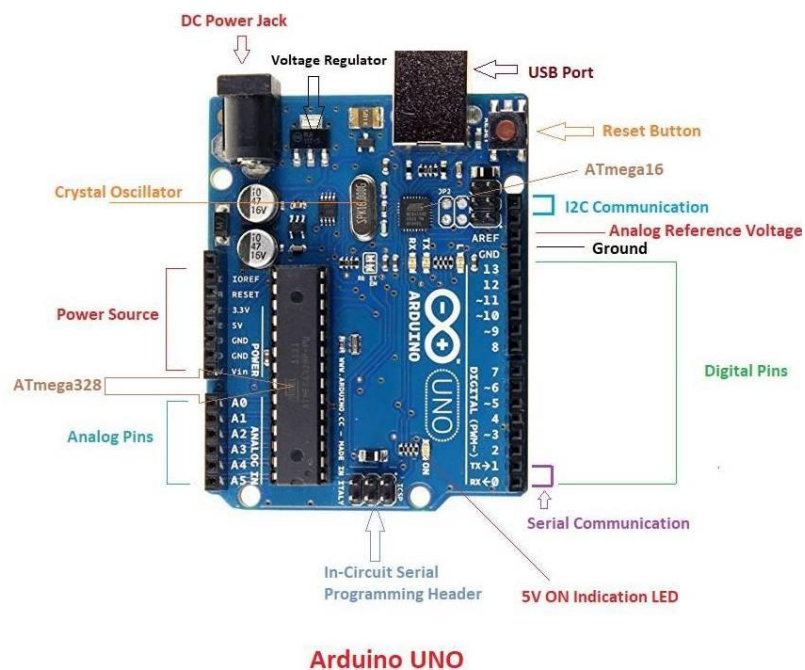
Microcontroller

While learning microcontroller requires some expertise and skills.

Nevertheless, we can say every Arduino is basically a microcontroller but not every microcontroller is an Arduino.

Introduction to Arduino

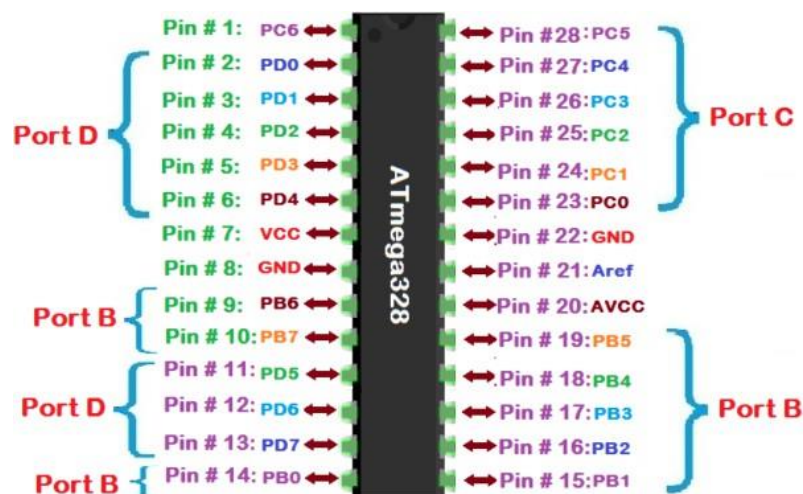
- **Arduino Uno** is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.
- First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.
- The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.
- It allows the designers to control and sense the external electronic devices in the real world



- This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. IDE is equally compatible with Windows, MAC or Linux Systems, however, Windows is preferable to use. Programming languages like C and C++ are used in IDE.
- Apart from USB, battery or AC to DC adopter can also be used to power the board.
- Arduino Uno boards are quite similar to other boards in Arduino family in terms of use and functionality, however, Uno boards don't come with FTDI USB to Serial driver chip.
- There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno are the most official versions that come with Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.
- When nature and functionality of the task go complex, Mirco SD card can be added in the boards to make them store more information.

Features of Arduino

- Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer.
- Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

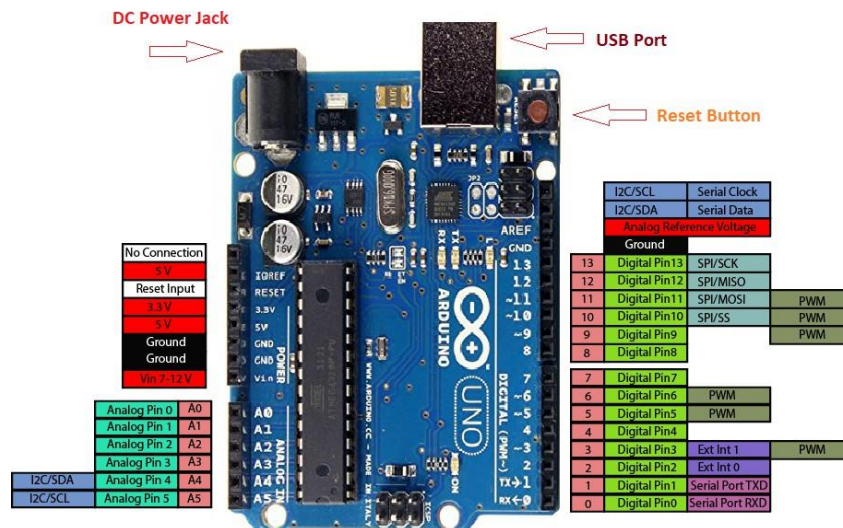


Atmega328 Microcontroller

- It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
- This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
- Reset pin is added in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
- There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.
- The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using `analogReference()` function and AREF pin.
- 13KB of flash memory is used to store the number of instructions in the form of code.
- Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

Arduino Pinout

- Arduino Uno is based on AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. Following figure shows the pinout of the Arduino Uno Board



Arduino Uno Pinout

Pin Description:

There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resistors useless and damages the device.

LED. Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

Vin. It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

5V. This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack.

USB supports voltage around 5V while Vin and Power Jack support a voltage ranges between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins, they result in bypassing the voltage regulation that can damage the board if voltage surpasses from its limit.

GND. These are ground pins. More than one ground pins are provided on the board which can be used as per requirement.

Reset. This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

IOREF. This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.

PWM. PWM is provided by 3, 5, 6,9,10, 11pins. These pins are configured to provide 8-bit output PWM.

SPI. It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.

AREF. It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

TWI. It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

Serial Communication. Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx pin is used to receive data while Tx pin is used to transmit data.

External Interrupts. Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

Arduino Uno Technical Specifications

Microcontroller	<u>ATmega328P</u> – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)

Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Communication and Programming:

Arduino Uno comes with an ability of interfacing with other other Arduino boards, microcontrollers and computer. The Atmega328 placed on the board provides serial communication using pins like Rx and Tx.

The Atmega16U2 incorporated on the board provides a pathway for serial communication using USB com drivers. Serial monitor is provided on the IDE software which is used to send or receive text data from the board. If LEDs placed on the Rx and Tx pins will flash, they indicate the transmission of data.

Arduino Uno is programmed using Arduino Software which a cross-platform application called IDE is written in Java. The AVR microcontroller Atmega328 laid out on the base comes with built-in boot loader that sets you free from using a separate burner to upload the program on the board.



Applications:

Arduino Uno comes with a wide range of applications. A larger number of people are using Arduino boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board.

- Embedded System
- Security and Defense System
- Digital Electronics and Robotics
- Parking Lot Counter
- Weighing Machines
- Traffic Light Count Down Timer
- Medical Instrument
- Emergency Light for Railways
- Home Automation
- Industrial Automation

There are a lot of other microcontrollers available in the market that are more powerful and cheap as compared to Arduino board. So, why you prefer Arduino Uno?

Actually, Arduino comes with a big community that is developing and sharing the knowledge with a wide range of audience. Quick support is available pertaining to technical aspects of any electronic project. When you decide Arduino board over other controllers, you don't need to arrange extra peripherals and devices as most of the functions are readily available on the board that makes your project economical in nature and free from a lot of technical expertise.

MEMS Sensor:

The term MEMS stands for micro-electro-mechanical systems. These are a set of devices, and the characterization of these devices can be done by their tiny size & the designing mode. The designing of these sensors can be done with the 1- 100-micrometer components. These devices can differ from small structures to very difficult electromechanical systems with numerous moving elements beneath the control of incorporated micro-electronics. Usually, these sensors include mechanical micro-actuators, micro-structures, micro-electronics, and micro-sensors in one package. This article discusses what is a MEMS sensor, working principle, advantages and it's applications

What is a MEMS Sensor?

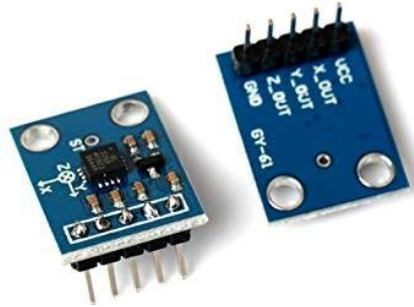
MEMS are low-cost, and high accuracy inertial sensors and these are used to serve an extensive range of industrial applications. This sensor uses a chip-based technology namely micro-electro-mechanical-system. These sensors are used to detect as well as measure the external stimulus like pressure, after that it responds to the pressure which is measured pressure with the help of some mechanical actions. The best examples of this mainly include revolving of a motor for compensating the pressure change.

The MEMS accelerometers can be divided into two important micro system architectures: piezo resistive and capacitive. Even though both of these two types of accelerometers possess internal proof masses which are excited by acceleration, the differences of these two architectures lie in the transduction mechanism which is used to the movement correlation of the internal proof mass to accelerate.

The Capacitive accelerometers possess a differential capacitor whose balance is disrupted by the proof mass movement. Piezo resistive accelerometers commonly rely on inducing, which attach the proof mass to the sensor which is used for identification of the movement of the mass.

Fujitsu successfully developed the 'FAR-S2AB' series, 3-axis Accelerometer, using state-of-the-art MEMS technology. This small and highly sensitive accelerometer can detect acceleration, inclination and vibration by measuring the motion in the x-, y-, and z-axis simultaneously. The MEMS 3-axis accelerometer consists of a Mass at the centre of the sensor's chip, which is suspended by 4 Beams doped with Piezo resistive material.

By sensing the mounting angle, the sensor can assist in compensating for the devices mounting angle, and therefore makes it possible to use ACCELEROMETER FACTSHEET MEMS 3-AXIS ACCELEROMETER normal SMD technology in high density boards, and also to realise the precise detection of the inclination angle. An interface IC within the sensor package also has temperature sensing and self-diagnosis functions.



Ultrasonic sensor

An ultrasonic sensor transmit ultrasonic waves into the air and detects reflected waves from an object. There are many applications for ultrasonic sensors, such as in intrusion alarm systems, automatic door openers and backup sensors for automobiles.

Accompanied by the rapid development of information processing technology, new fields of application, such as factory automation equipment and car electronics, are increasing and should continue to do so. Using its unique piezoelectric ceramics manufacturing technology developed over many years, Murata has developed various types of ultrasonic sensors which are compact and yet have very high performance. The information contained in this catalog will help you to make effective use of our ultrasonic sensors.



HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: $<15^{\circ}$
- Operating Current: $<15\text{mA}$

- Operating Frequency: 40Hz

HC-SR04 Ultrasonic Sensor - Working

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$\text{Distance} = \text{Speed} \times \text{Time}$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below



Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the

on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

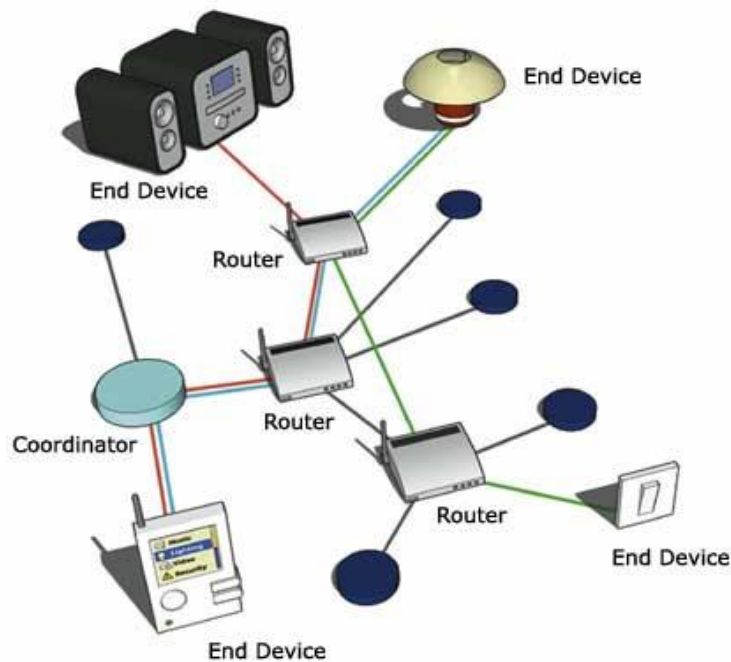
The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

Zigbee Module:

Zigbee is the most popular industry wireless mesh networking standard for connecting sensors, instrumentation and control systems. Zigbee, a specification for communication in a wireless personal area network (WPAN), has been called the "Internet of things." Theoretically, your Zigbee-enabled coffee maker can communicate with your Zigbee-enabled toaster. Zigbee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks. Zigbee and IEEE 802.15.4 are low data rate wireless networking standards that can eliminate the costly and damage prone wiring in industrial control applications. Flow or process control equipment can be place anywhere and still communicate with the rest of the system. It can also be moved, since the network doesn't care about the physical location of a sensor, pump or valve.

The Zigbee RF4CE standard enhances the IEEE 802.15.4 standard by providing a simple networking layer and standard application profiles that can be used to create interoperable multi-vendor consumer electronic solutions.

Zigbee targets the application domain of low power, low duty cycle and low data rate requirement devices. Figure below shows the example of a Zigbee network.



Zigbee is composed to become the global control/sensor network standard. It has been designed to provide the following features:

- Low power consumption, simply implemented
- Users expect batteries to last many months to years
- Bluetooth has many different modes and states depending upon your latency and power requirements such as sniff, park, hold, active, etc.; Zigbee/IEEE 802.15.4 has active (transmit/receive) or sleep
- Even mains powered equipment needs to be conscious of energy. Zigbee devices will be more ecological than its predecessors saving megawatts at it full deployment.

Wireless Standards

The demand for wireless solutions continues to grow and with it new standards have come forward and other existing standards have strengthened their position in the marketplace.

Comparison of Wireless Standards			
Wireless Parameter	Bluetooth	Wi-Fi	ZigBee
Frequency band	2.4 GHz	2.4 GHz	2.4 GHz
Physical/MAC layers	IEEE 802.15.1	IEEE 802.11b	IEEE 802.15.4
Range	9 m	75 to 90 m	Indoors: up to 30 m Outdoors (line of sight): up to 100 m
Current consumption	60 mA (Tx mode)	400 mA (Tx mode) 20 mA (Standby mode)	25-35 mA (Tx mode) 3 μ A (Standby mode)
Raw data rate	1 Mbps	11 Mbps	250 Kbps
Protocol stack size	250 KB	1 MB	32 KB 4 KB (for limited function end devices)
Typical network join time	>3 sec	variable, 1 sec typically	30 ms typically
Interference avoidance method	FHSS (frequency-hopping spread spectrum)	DSSS (direct-sequence spread spectrum)	DSSS (direct-sequence spread spectrum)
Minimum quiet bandwidth required	15 MHz (dynamic)	22 MHz (static)	3 MHz (static)
Maximum number of nodes per network	7	32 per access point	64 K
Number of channels	19	13	16

Zigbee Network Devices and their Operating Modes

Two types of devices can participate in a LR-WPAN: a full function device (FFD) and a reduced function device (RFD).

An RFD does not have routing capabilities. RFDs can be configured as end nodes only. They communicate with their parent, which is the node that allowed the RFD to join the network.

An FFD has routing capabilities and can be configured as the PAN coordinator. In a star network all nodes communicate with the PAN coordinator only so it does not matter if they are FFDs or RFDs. In a peer to-peer network there is also one PAN coordinator, but there are other FFDs which can communicate with not only the PAN coordinator, but also with other FFDs and RFDs.

There are three operating modes supported by IEEE 802.15.4: PAN coordinator, coordinator, and end device. FFDs can be configured for any of the operating modes. In Zigbee terminology the PAN coordinator is referred to as simply "coordinator." The IEEE term "coordinator" is the Zigbee term for "router."

How Zigbee Works

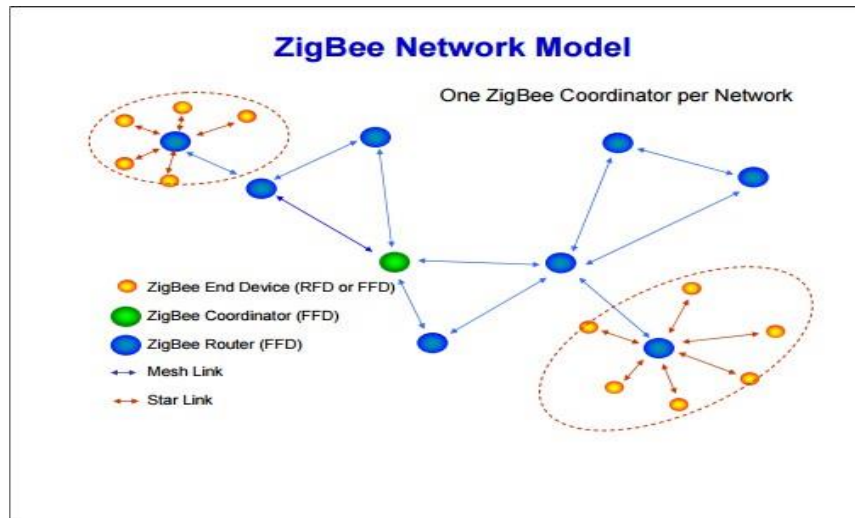
The Zigbee standard has the capacity to address up to 65535 nodes in a single network. However, the Zigbee Protocol defines three general types of nodes: Coordinators, Routers and End Device, with a requirement of one Coordinator per network. While all nodes can send and receive data, there are differences in the specific roles they play. An example of a Zigbee network is shown below in Figure.

Zigbee Coordinators (Network Creator) - are the most capable of the three node types. There is exactly one coordinator in each network and it is the device that establishes the network originally. It is able to store information about the network, including security keys.

Zigbee Routers (Multi-hop Routing) - act as intermediate nodes, relaying data from other devices.

Zigbee End Devices (Very low power) - can be low-power / battery powered devices. They have sufficient functionality to talk to their parents (either the coordinator or a router) and cannot relay data from other devices. This reduced functionality allows for the potential to reduce their cost.

To provide for low cost implementation options, the Zigbee Physical Device type distinguishes the type of hardware based on the IEEE 802.15.4 definition of reduced function device (RFD) and full function device (FFD). An IEEE 802.15.4 network requires at least one FFD to act as a network coordinator.



Zigbee uses two types of devices. Reduced-function devices (RFDs) are sensors that communicate with full-function devices (FFDs). FFDs are complex nodes that conform to the full 802.15.4 standard and can serve as routers. An end device gets its instructions from a Zigbee Coordinator. All Zigbee networks must include a coordinator, which is a full function device that manages the network.

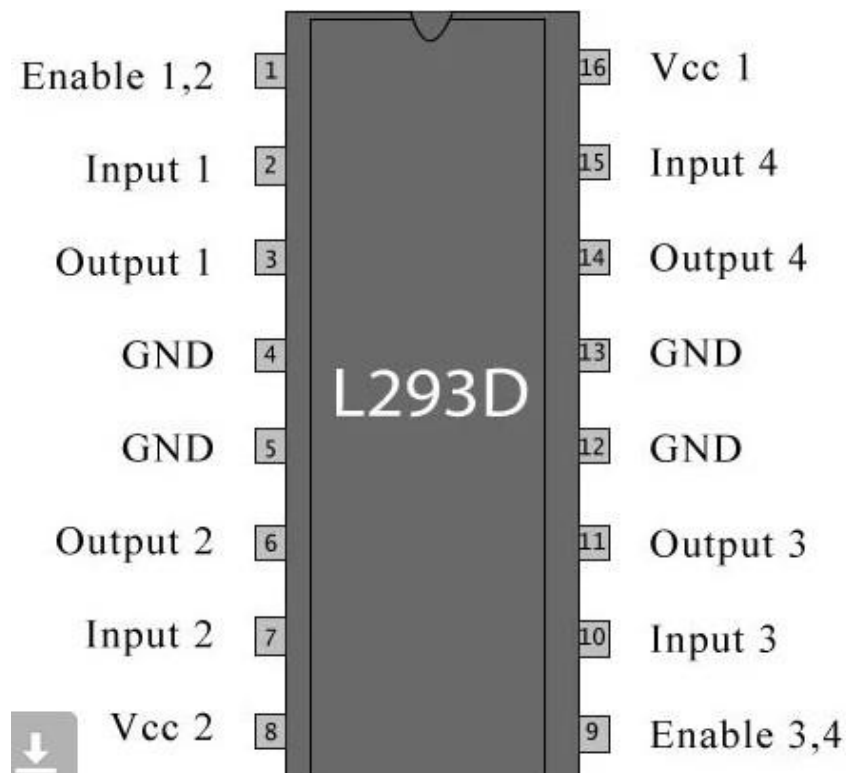
Zigbee technology provides static and dynamic star, cluster tree and mesh networking structures that allow large area network coverage, scalable networks and single point-of-failure avoidance.

If the end device is out of range of its coordinator, it can communicate through a router. The diagram shows what is called a mesh network. The term mesh is used because of the routers and coordinators have multiple communication path options. network may be extended through the use of Zigbee routers, Using local addressing you can configure simple networks of more than 65,000 (2¹⁶) nodes, thereby reducing address overhead.

Motor Driver:

A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors. The most commonly used motor driver IC's are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-

bridge. H-bridge is the simplest circuit for controlling a low current rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.



The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

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

Working of L293D

There are 4 input pins for l293d, pin 2, 7 on the left and pin 15, 10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.

In simple you need to provide Logic 0 or 1 across the input pins for rotating the motor.

L293D Logic Table.

Let's consider a Motor connected on left side output pins (pin 3, 6). For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.

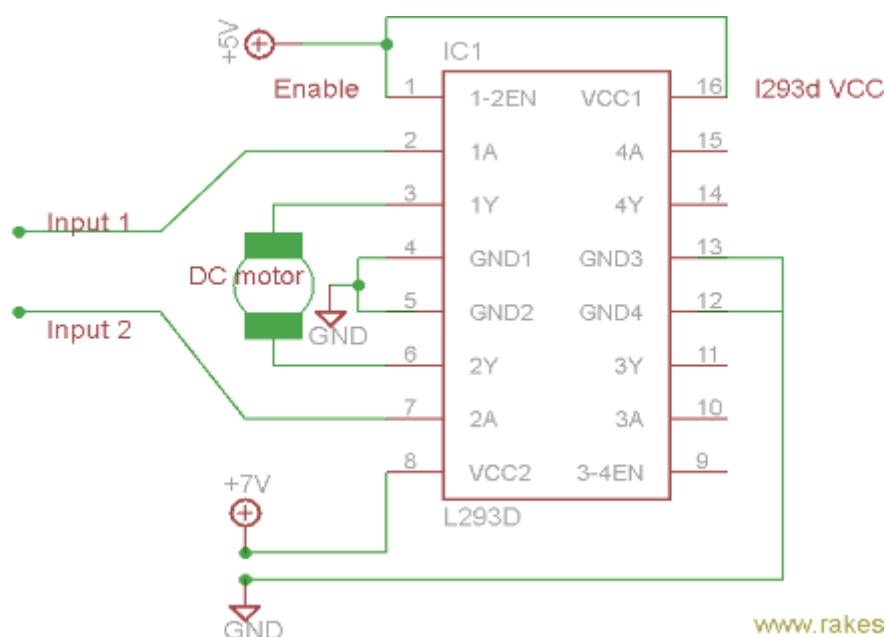
Pin 2 = Logic 1 and Pin 7 = Logic 0 | Clockwise Direction

Pin 2 = Logic 0 and Pin 7 = Logic 1 | Anticlockwise Direction

Pin 2 = Logic 0 and Pin 7 = Logic 0 | Idle [No rotation] [Hi-Impedance state]

Pin 2 = Logic 1 and Pin 7 = Logic 1 | Idle [No rotation]

Circuit Diagram for l293d motor driver IC controller



Voltage Specification

VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply). L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply.

The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d. VCC pin 16 is the voltage for its own internal Operation. The maximum voltage ranges from 5v and up to 36v.

What is a DC Motor?

A direct current (DC) motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current, and convert this energy into mechanical rotation.

DC motors use magnetic fields that occur from the electrical currents generated, which powers the movement of a rotor fixed within the output shaft. The output torque and speed depends upon both the electrical input and the design of the motor.

How DC motors work

The term ‘DC motor’ is used to refer to any rotary electrical machine that converts direct current electrical energy into mechanical energy. DC motors can vary in size and power from small motors in toys and appliances to large mechanisms that power vehicles, pull elevators and hoists, and drive steel rolling mills.

But how do DC motors work?

DC motors include two key components: a stator and an armature. The stator is the stationary part of a motor, while the armature rotates. In a DC motor, the stator

provides a rotating magnetic field that drives the armature to rotate.

A simple DC motor uses a stationary set of magnets in the stator, and a coil of wire with a current running through it to generate an electromagnetic field aligned with the centre of the coil. One or more windings of insulated wire are wrapped around the core of the motor to concentrate the magnetic field.

The windings of insulated wire are connected to a commutator (a rotary electrical switch), that applies an electrical current to the windings. The commutator allows each armature coil to be energised in turn, creating a steady rotating force (known as torque).

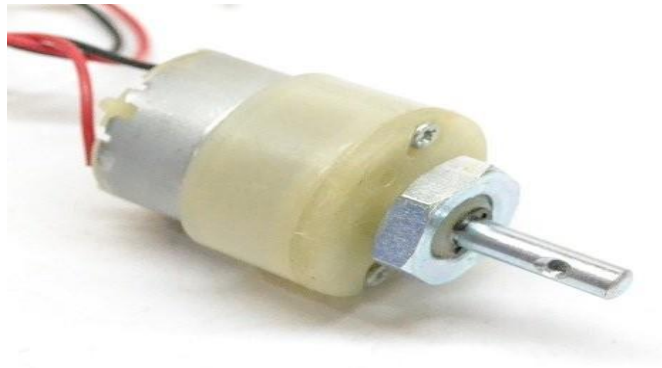
When the coils are turned on and off in sequence, a rotating magnetic field is created that interacts with the differing fields of the stationary magnets in the stator to create torque, which causes it to rotate. These key operating principles of DC motors allow them to convert the electrical energy from direct current into mechanical energy through the rotating movement, which can then be used for the propulsion of objects.

Who invented the DC motor?

This amazing piece of electrical equipment has revolutionised our lives in many ways, but who invented the DC motor? As with all major innovations, there are many people who had a role to play through the development of similar mechanisms.

In the US, Thomas Davenport is widely celebrated as the inventor of the first electric motor, and undoubtedly he was the first to patent a useable electric motor in 1837. Davenport, however, was not the first person to build an electric motor, with various inventors in Europe having already developed more powerful versions by the time Davenport filed his patent.

In 1834, Moritz Jacobi had presented a motor that was three times as powerful as the one Davenport would later patent, while Sibrandus Stratingh and Christopher Becker were the first to demonstrate a practical application for an electric motor, by running a small model car in 1835.



The first practical DC motor was invented some years later in 1886 by Frank Julian Sprague, whose invention led to the first motor powered trolley system in 1887, and the first electric elevator in 1892. Sprague's DC motor was a hugely significant development, leading to a variety of applications which would reshape the face of industry and manufacturing.

Types of DC Motors

So far, this guide has broadly explained how DC motors work, the history of these mechanisms, and what they look like. While the principles are the same across variants, there are actually several different types of DC motors, which offer specific advantages and disadvantages over each other.

Buzzer:

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play."



Buzzer Pin Configuration

Pin Number	Pin Name	Description
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 5V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

Buzzer Features and Specifications

- Rated Voltage: 6V DC
- Operating Voltage: 4-8V DC
- Rated current: <30mA
- Sound Type: Continuous Beep
- Resonant Frequency: ~2300 Hz
- Small and neat sealed package
- Breadboard and Perf board friendly

How to use a Buzzer

A **buzzer** is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customized with help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

Applications of Buzzer

- Alarming Circuits, where the user has to be alarmed about something
- Communication equipment's
- Automobile electronics
- Portable equipment's, due to its compact size

Joystick:

A joystick is an input device that can be used for controlling the movement of the cursor or a pointer in a computer device. The pointer/cursor movement is controlled by maneuvering a lever on the joystick. The input device is mostly used for gaming applications and, sometimes, in graphics applications.



SOFTWARE REQUIREMENTS

Arduino IDE:

Arduino IDE where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

Introduction to Arduino IDE:

- Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
- 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 easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
- 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.

How to install Arduino IDE:

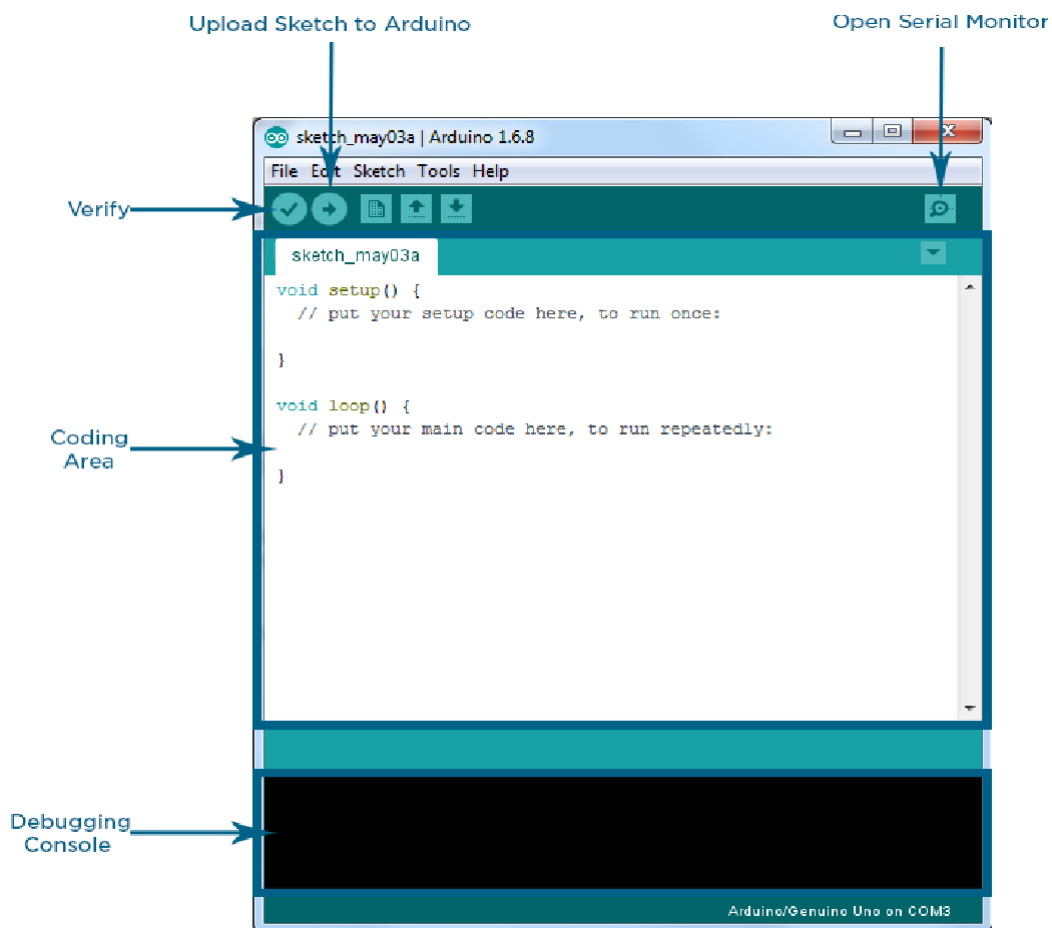
You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

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

The IDE environment is mainly distributed into three sections

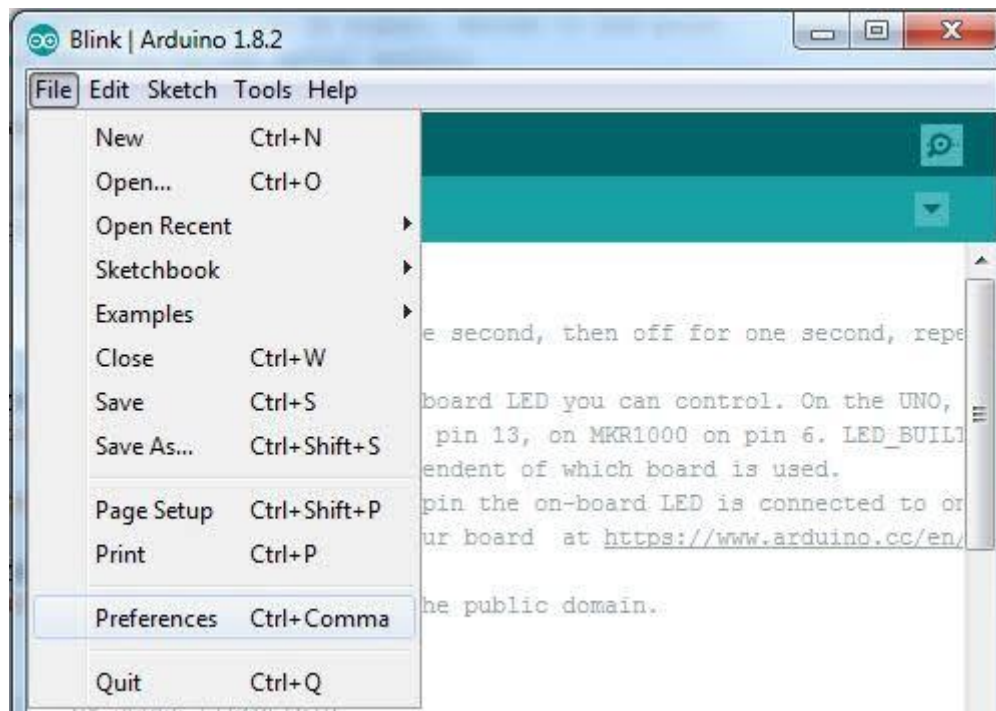
- **1. Menu Bar**
- **2. Text Editor**
- **3. Output Pane**

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

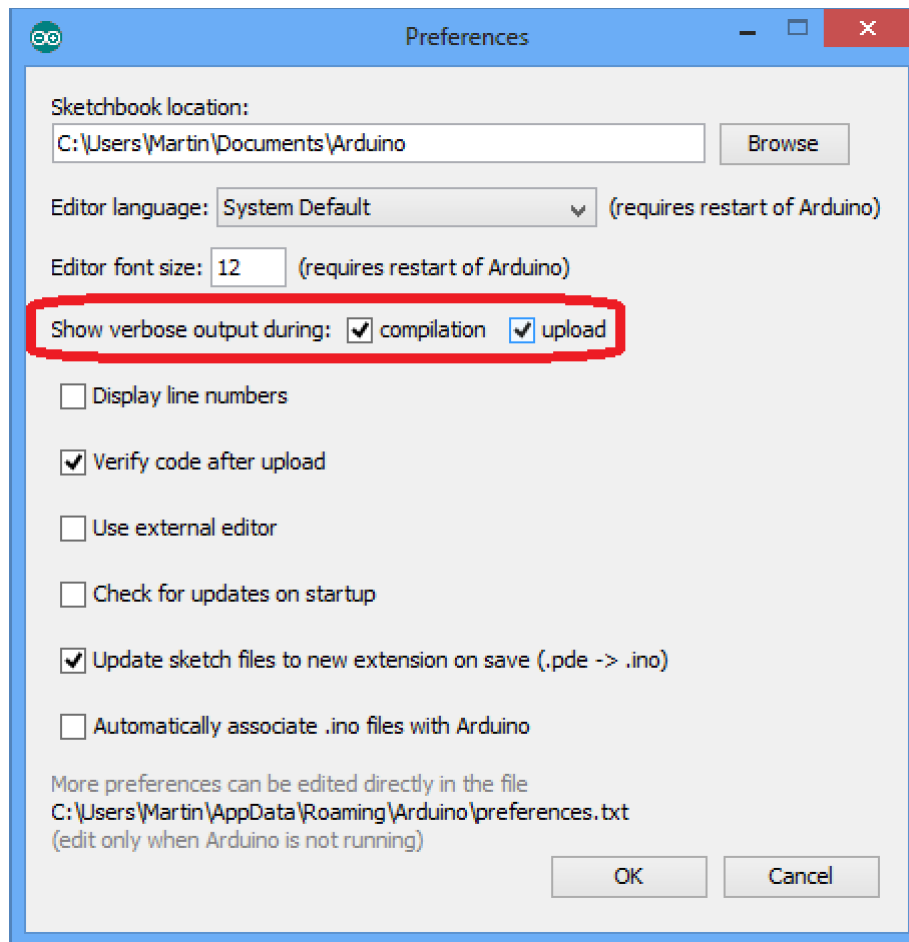


The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

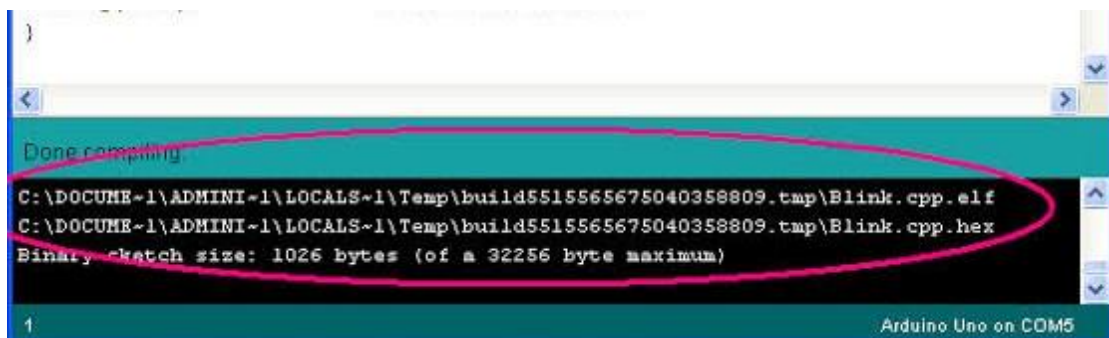
- **File** – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.



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.

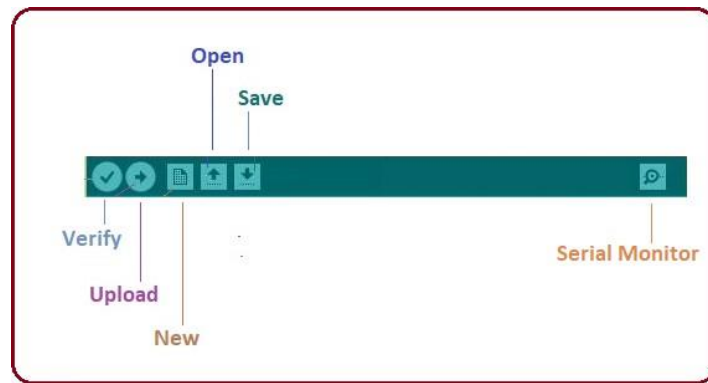


And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.



- **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.

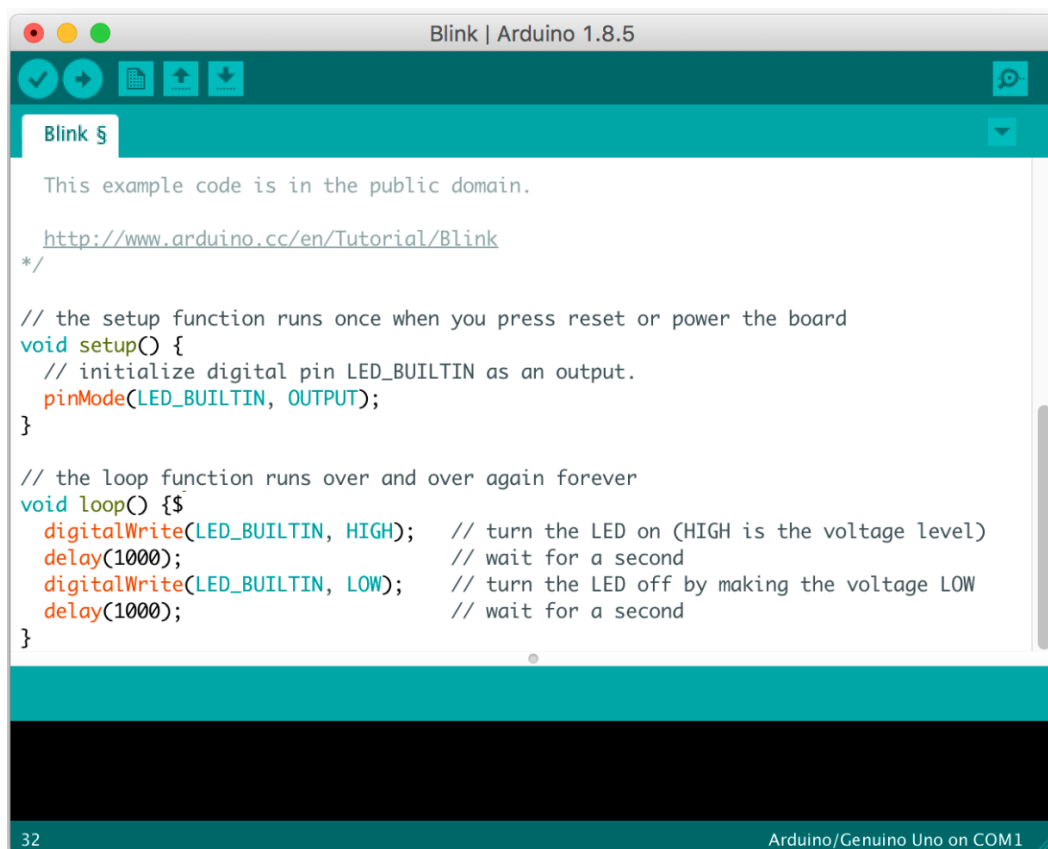
The **Six Buttons** appearing under the Menu tab are connected with the running program as follow.



- The check mark 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 for 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.



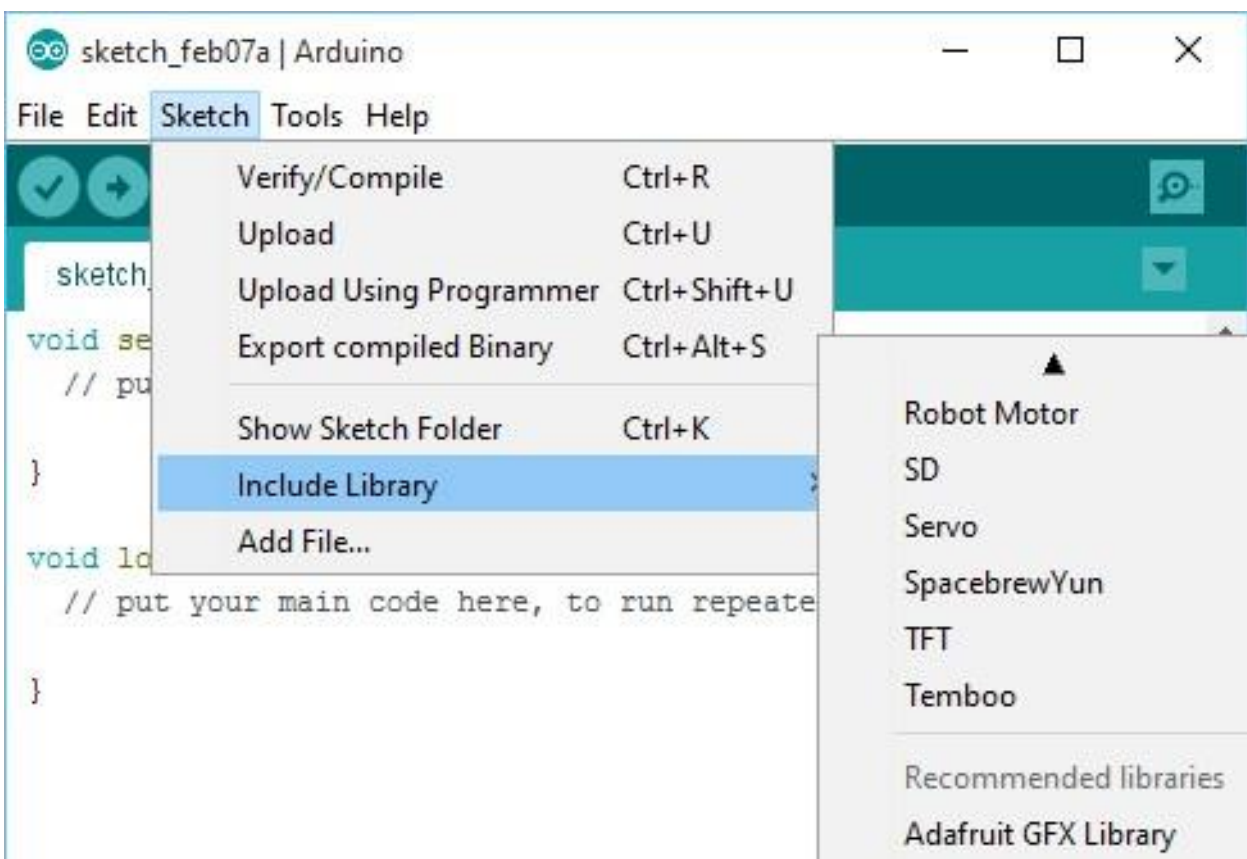
The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.



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.

Libraries:

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add 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 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 the external sources.

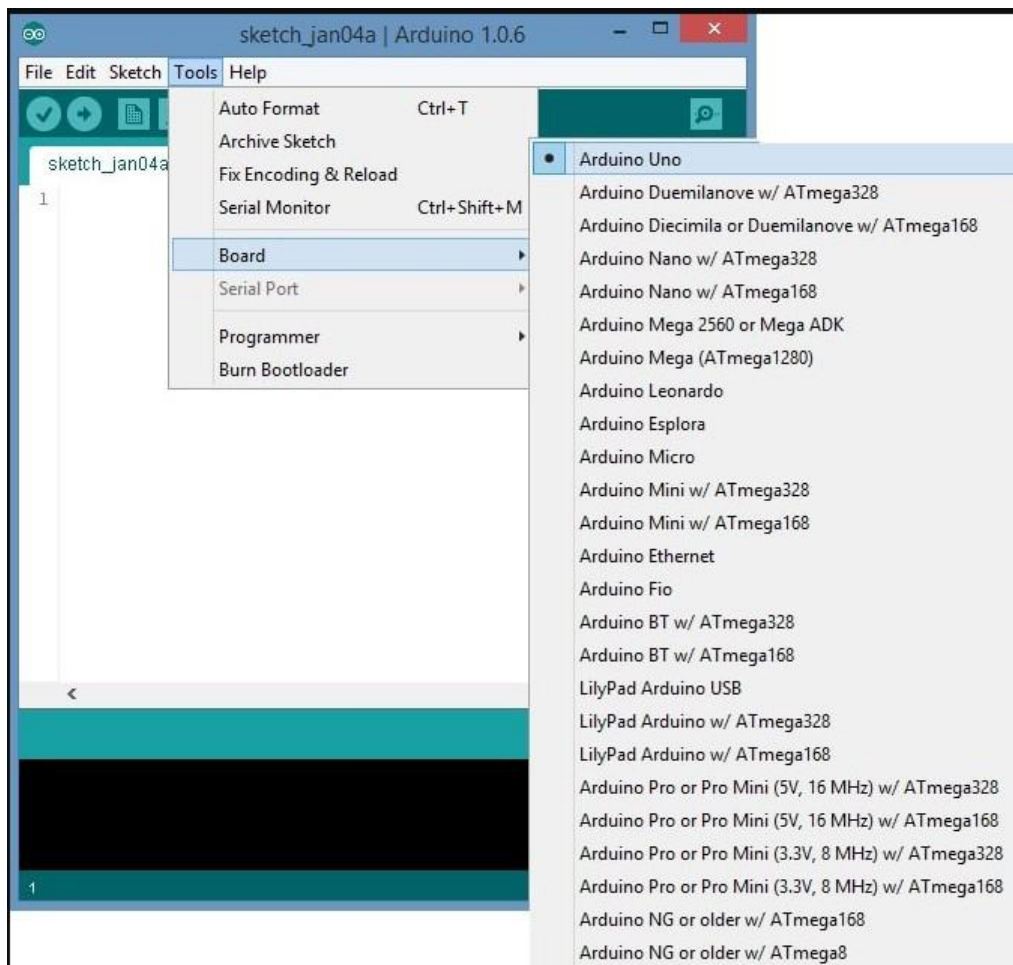
Making pins Input and output:

The digitalWrite 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.

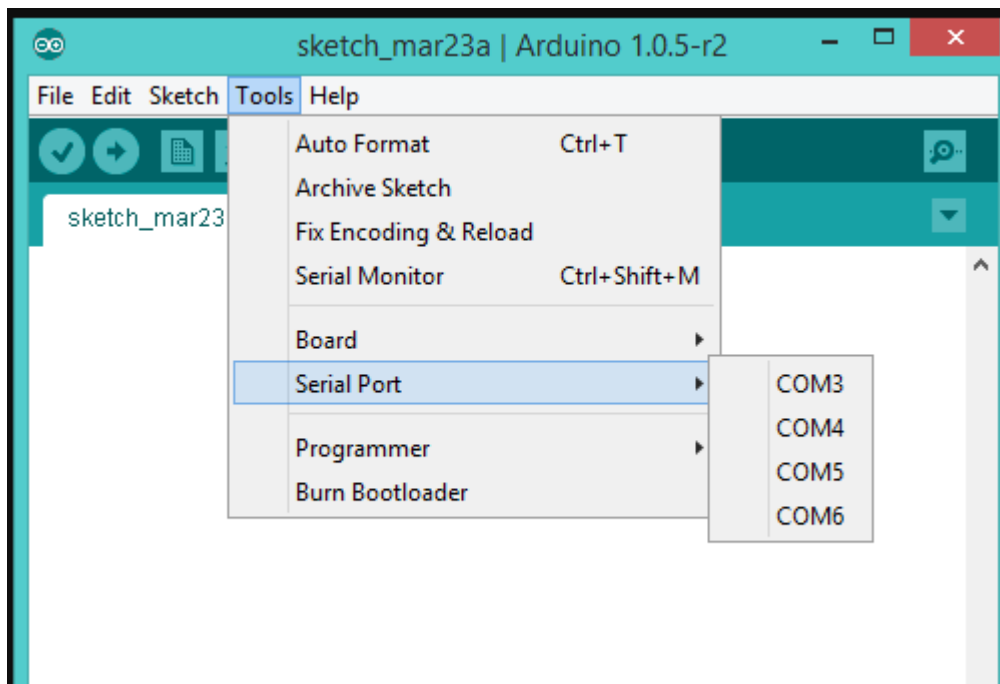
How to select the board:

In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the Tools on the Menu, it will open like the figure below.



- Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the ports section of the Windows Device Manager.

Following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with COM4 port at the right bottom corner of the screen.



- After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six button section or you can go to the Sketch section and press verify/compile and then upload.
- The sketch is written in the text editor and is then saved with the file extension .ino.

It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

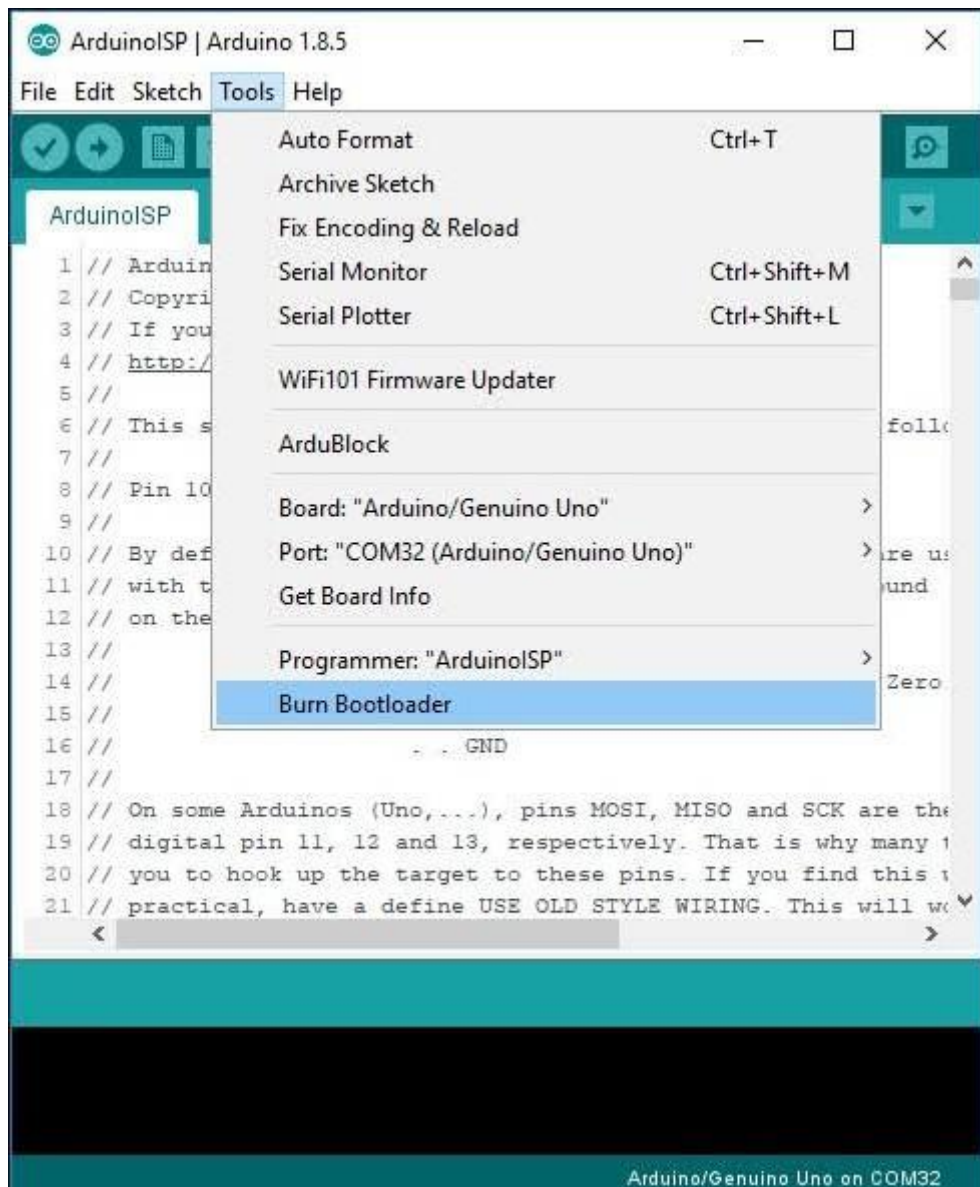
- Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

Note: The port selection criteria mentioned above is dedicated for Windows operating system only, you can check this [Guide](#) if you are using MAC or Linux.

- The amazing thing about this software is that no prior arrangement or bulk of mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment.

BootLoader:

As you go to the Tools section, you will find a bootloader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.



When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

ADVANTAGES & APPLICATIONS

Advantages:

- Reduces manual efforts
- Reduces time

Applications:

- Home

- hospitals,
- car systems
- industries

Chapter 4

Conclusion:

Using this proposed system, the signals were sent from the headset to the Arduino in order to instigate movements in the wheelchair based on the inputs from the brain. With the implementation of the above hardware architecture, the wheelchair could be controlled successfully. This wheelchair uses two motorsports' movements. Although this system is very raw, it is a step towards brain-controlled movement. The movement of the wheelchair will be solely configured to the signals generated by the mind. User-based or specific modules can be created thus generating a unique footprint. It uses upcoming and ever-evolving technology that will enable easy and manageable iterations. The components used are very low cost yet have an optimum performance level.

In this project a technique of head motion recognition is used to enable wheelchair control for quadriplegics. This system gives independent movement and a psychological advantage of being independent. To avoid physical hardship an accelerometer is used due to which the slight movement of head turns the wheelchair into the desired direction.

Future Scope:

In the future scope of the system, it would be better to design a cloud system to communicate directly from the headset to the wheelchair. In addition, improving accuracy is required by training more EEG data and testing the system on more subjects. An obstacle in the way could be detected automatically by the wheelchair forcing it to stop. The wheelchair could be integrated with head movements to control factors such as speed and brakes. Arduino efficiently controls drivers and drivers are able to communicate with motors. With more modifications, these 31 systems could be converted into a real system that can support the lives of amputees. This proposed wheelchair will potentially have an immense effect on the healthcare industry and the lives of disabled people. Some extra features can be added in the future to improve the functionality and practicality of the wheelchair

References:

- [1] U. Cortés, C. Urdiales, R. Annicchiarico, C. Barrué, A.B. Martinez, C. Caltagirone: Assistive Wheelchair Navigation: A Cognitive View, Studies in Computational Intelligence: Advanced Computational Intelligence Paradigms in Healthcare – 1, Vol. 48, 2007, pp. 165 – 187.
- [2] A. Landi: Update on Tetraplegia, Journal of Hand Surgery, Vol. 28, No. 3, June 2003, pp. 196 – 204.
- [3] N.I. Katevas, N.M. Sgouros, S.G. Tzafestas, G. Papakonstantinou, P. Beattie, J.M. Bishop, P. Tsanakas, D. Koutsouris: The Autonomous Mobile Robot SENARIO: A Sensor Aided

Intelligent Navigation System for Powered Wheelchairs, IEEE Robotics and Automation Magazine, Vol. 4, No. 4, Dec. 1997, pp. 60 – 70.

[4] G. Bourhis, O. Horn, O. Habert, A. Pruski: An Autonomous Vehicle for People with Motor Disabilities, IEEE Robotics and Automation Magazine, Vol. 8, No. 1, March 2001, pp. 20 – 28.

[5] A. Lankenau, T. Rofer: A Versatile and Safe Mobility Assistant, IEEE Robotics and Automation Magazine, Vol. 8, No. 1, March 2001, pp. 29 – 37.

[6] M. Mazo: An Integral System for Assisted Mobility (Automated Wheelchair), IEEE Robotics and Automation Magazine, Vol. 8, No. 1, March 2001, pp. 46 – 56.

[7] H.A. Yanco: Wheellesley: A Robotic Wheelchair System: Indoor Navigation and User Interface, Lecture Notes in Artificial Intelligence, Vol. 1458, Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing, Berlin, Springer, Germany, 1998, pp. 256 – 268.

[8] C.H. Kuo, H.H.W. Chen: Human-oriented Design of Autonomous Navigation Assisted Robotic Wheelchair for Indoor Environments, IEEE International Conference on Mechatronics, Budapest, Hungary, 03 – 05 July 2006, pp. 230 – 235.

[9] J.M. Detriche, B. Lesigne: Robotic system MASTER, European Conference on the Advancements of Rehabilitation Technology, Maastricht, Pays-Bas, Holland, 05 – 08 Nov. 1990, pp. 12 – 15.

[10] J.L. Jaffe: A Case Study: The Ultrasonic Head Controlled Wheelchair and Interface, OnCenter: Technology Transfer News, Vol. 1, No. 2, 1990.

[11] R.C. Simpson, S.P. Levine: Voice Control of a Powered Wheelchair, IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 10, No. 2, June 2002, pp. 122 – 125.

[12] P. Pino, P. Amoud, E. Brangier: A More Efficient Man Machine Interface: Fusion of the Interacting Telethesis and Smart Wheelchair Projects, 2nd International Conference on Knowledge-based Intelligent Electronic System, Adelaide, Australia, 21 – 23 April 1998, pp. 41 – 45. Wheelchair Control by Head Motion 151

[13] H.A. Yanco: Integrating Robotic Research: A Survey of Robotic Wheelchair Development, AAAI Spring Symposium on Integrating Robotic Research, Stanford, CA, USA, March 1998.

[14] M. Slavkovic, D. Jevtic: Face Recognition using Eigenface Approach, Serbian Journal of Electrical Engineering, Vol. 9, No. 1, Feb. 2012, pp. 121 – 130.

- [15] S. Mitra, T. Acharya: Gesture Recognition: A Survey, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 37, No. 3, May 2007, pp. 311 – 324.
- [16] L.R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol. 77, No. 2, Feb. 1989, pp. 257 – 286.
- [17] S.K. Pal, S. Mitra: Neuro-fuzzy Pattern Recognition: Methods in Soft Computing, John Wiley and Sons, NY, USA, 1999.
- [18] J. Liu, L. Zhong, J. Wickramasuriya, V. Vasudevan: uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications, Pervasive and Mobile Computing, Vol. 5, No. 6, Dec. 2009, pp. 657 – 675.

www.Arduino.org

www.beyondlogic.org

www.Wikipedia.org

www.elementzonline.com

www.elementztechblog.wordpress.com

www.msresearchworld.com