WIRELESS ROBOT CONTROL THROUGH HUMAN MACHINE INTERFACE USING EEG

A project report submitted in the partial fulfilment of degree of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

With specialization in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

P. Sainath - 1210413639
S. Sravya - 1210413653
N. B. Sri Harsha - 1210413637
M. Sai Sree Harsha - 1210413632

Under the Esteemed Guidance

Mr. K. Tarakeswara Rao, M.E

Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
GITAM INSTITUTE OF TECHNOLOGY
GITAM University
Estd. u/s 3 of UGC Act, 1956
Visakhapatnam - 530045

Department of Electronics and Communication Engineering

GITAM Institute of Technology

GITAM University

CERTIFICATE

This is to certify that the Project work entitled **"Wireless Mobile Robot Control through Human Machine Interface using Brain Signals"** is carried out by P. Sainath (1210413639), S. Sravya (1210413653), N.B. Sri Harsha (1210413637) and M. Sai Sree Harsha (1210413632) and is submitted in fulfilment of the requirements for the award of degree **Bachelor of Technology** in **Electronics and Communication Engineering**, **GITAM University**, Visakhapatnam

Project Guide:

Mr. K. Tarakeswara Rao
Assistant Professor,
Dept. of ECE,
GIT,
GITAM University.

Head of the Department:

Dr. V. Malleswara Rao
Professor,
Dept. of ECE,
GIT,
GITAM UNIVERSITY

DECLARATION

I hereby declare that the project entitled, "Wireless Mobile Robot Control through Human Machine Interface using Brain Signals" done by me, under the guidance of Prof Sri. K. TARAKESWARA RAO, has submitted in partial fulfilment of the requirements for award of degree of Bachelor of Technology in Electronics and communication Engineering. This work is carried out in the department Electronics and communication of Engineering, GITAM University, Visakhapatnam. This work has not been previously formed the basics for the award of any other degree or certificate.

BY

NAME	**REGD NO**	**SIGNATURE**
P. SAINATH	**1210413639**	
S. SRAVYA	**1210413653**	
N. B. SRI HARSHA	**1210413637**	
M. SAI SREE HARSHA	**1210413632**	

ACKNOWLEDGEMENT

I am very gratified to **Sri. K. Tarakeswara Rao**, Professor, ECE Department my project guide, for his constant and valuable guidance. His insight useful discussions have given confidence and inspiration to carry out project work successfully. I would like to express my gratitude to him.

I wish to express my gratitude to **Dr. V. Malleswara Rao**, Head of the Department, Electronics and communication Engineering, for his constant cooperation during the course of work.

I would like to thank our **Prof. K. Lakshmi Prasad**, Principal, GITAM UNIVERSITY for his encouragement and support to carry out this project successfully.

I would like to thank my project reviewer **Sri N.V.S.V Vijay Kumar**, Assistant Professor, for his suggestions and support in the tenure of my project.

I sincerely thank all the teaching and non-teaching staff in the Electronics and communication Engineering department for their sustained help in my pursuits.

ABSTRACT

Electroencephalogram based Brain-Computer Interface robot which helps as a powerful support for severely disabled people in their regular activities. This project proposes and implements a brain signal (mind) controlled robot to yield different movements .The key lies in the mapping of the EEG signal to the robot to achieve the objective. In this project we are developing a cost effective BCI robot that will help the physically challenged to lead an independent life with the help of their brain signals using non-invasive techniques. Mind-controlled mobile robot is a robot that uses an EEG-based Brain-Robot Interface (BRI) system that serves as a powerful aid for severely disabled people in their daily and professional life which does not uses a computer in between user and robot as in Brain-Computer interface (BCI), particularly to help them move deliberately by translating the brain signals into useful commands. Robot intelligence includes heart rate measurement, fire detection, collision avoidance and life detection systems for improving the safety of individuals.

This project describes about a brain controlled robot based on Brain–computer interfaces (BCI). BCIs are systems that can bypass conventional channels of communication (i.e., muscles and thoughts) to provide direct communication and control between the human brain and physical devices by translating different patterns of brain activity into commands in real time. With these commands a mobile robot can be controlled. Here the robot is self- controlled with the ultrasonic sensor. The intention of the project work is to develop a robot that can assist the disabled people in their daily life to do some work independent on others. Here, we are analysing the brain wave signals. Human brain consists of millions of interconnected neurons.

Robot is that device that can be remotely controlled using the user’s brain signals. This system uses BCI (Brain-Computer Interface) to provide communication between our brain and the robot. It uses an EEG (Electroencephalogram) headset to acquire data, classifies and interprets the data set on the hardware, and achieves desired commands on the robot based on the provided classification. The data is transferred through a Bluetooth module, while the commands are executed by Arduino. The purpose of the robot is to demonstrate the feasibility of applying in BCI application, as we will be shown in this project to controlling the robot.

TABLE OF CONTENTS

ABSTRACT **i**

LIST OF FIGURES **iv**

LIST OF TABLES **vi**

CHAPTER 1: INTRODUCTION 1

1.1 PROJECT DESCRIPTION 3
1.2 WHAT IS EEG? 4
1.3 ABOUT BCI 5
1.3.1 Working of BCI 7
1.3.2 History 7
1.3.3 Basic components of BCI 8
1.3.4 Types of BCI 9
1.4 ABOUT BRAIN WAVES 10

CHAPTER 2: **NEUROSKY MINDWAVE HEADSET** 17

2.1 INTRODUCTION 18
2.1.1 Setting up Your MindWave 19
2.1.2 Replacing the battery 19
2.1.3 Power 19
2.1.4 LED Light 20
2.1.5 Installing in Windows 20
2.1.6 Fitting of the MindWave 21
2.1.7 Thinkgear 22
2.1.8 eSense 23

CHAPTER 3 ARDUINO 25

3.1 WHAT IS ARDUINO? 26
3.1.1Microcontroller 26
3.1.2 Technical Specification 27
3.1.3 Pin specification 30

3.2ARDUINO TYPES 34
3.2.1 Features of Arduino 36
3.3 INSTALLING ARDUINO ON YOUR COMPUTER 36
3.4 SOFTWARE 38
3.5 APPLICATIONS OF ARDUINO 40
3.6 SUMMARY 41
CHAPTER 4: BLUETOOTH HC-05 MODULE 42
4.1 INTRODUCTION 43
4.1.1 Hardware features 43
4.1.2 HC-05 specifications 44
4.2 PIN DESCRIPTION 45
4.3 AT COMMANDS 47
4.4 SETTING UP CONNECTIONS 47
4.4.1 Interface module 48
4.5 CODE 56
CHAPTER 5: BUILDING THE ROBOT 68
5.1 INTRODUCTION 69
5.1.1 Tools and parts 70
5.2 BUILDING THE ROBOT 72
5.3 WORKING OF ROBOT 76
5.4 CODE 77

CONCLUSION 83
BIBILOGRAPHY 84

LIST OF FIGURES

Figure 1.1 Basics of BCI	6
Figure 1.2 Block Diagram of BCI	7
Figure 1.3 Basic flow of Signal	8
Figure 1.4 Types of BCI	9
Figure 1.5 Connection with Brain	10
Figure 1.6 Waves in Brain Signal	12
Figure 1.7 Graphical Representation of Brain Waves	13
Figure 1.8 State of Mind	15
Figure 2.1 MindWave Headset	18
Figure 2.2 Battery	19
Figure 2.3 On/Off Switch	20
Figure 2.4 Brain Waves in MindWave Application	24
Figure 3.1 Arduino UNO	26
Figure 3.2 Arduino UNO Pins	30
Figure 3.3 Types of Arduino	34
Figure 3.4 Basic Default Window of Arduino	37
Figure 3.5 Setup Procedure	38
Figure 3.6 COM Port Setup	39
Figure 3.7 Example Program Blink Code	39
Figure 3.8 Uploading Blink Program	40

Figure 4.1 MicroController – Bluetooth Module Connectivity	44
Figure 4.2 HC-05 & HC-06 modules	46
Figure 4.3 HC-05 wiring diagram	48
Figure 4.4 AT Commands in Serial Monitor	49
Figure 4.5 Determining MAC Address of MindWave	51
Figure 4.6 MAC Address of MindWave	52
Figure 4.7 Serial Monitor returning ‘OK’ Commands	54
Figure 4.8 Arduino, MindWave & LED	55
Figure 5.1 Mechanical Parts in Chassis	70
Figure 5.2 Picture of Assembled Chassis	72
Figure 5.3 Picture of Arduino UNO	73
Figure 5.4 Picture of L293D on Perf Board	73
Figure 5.5 Picture of HC-05	74
Figure 5.6 Circuit Diagram of Entire Setup	74
Figure 5.7 Code i.e., uploaded to Arduino	75
Figure 5.8 Operational Block diagram for Robot	76

LIST OF TABLES

Table 1.1 BrainWaves, Frequencies and Functions 11

Table 1.2 Details of Brain Waves 16

Table 2.1 OS Requirements 19

Table 2.2 LED Light States 20

CHAPTER 1

INTRODUCTION

The ultimate purpose of a direct brain computer interface (BCI) is to allow an individual with severe motor disabilities to have effective control over devices. A BCI system detects the presence of specific patterns in a person's ongoing brain activity that relates to the person's intention to initiate control. In our project we use a mind-controlled robot as a prototype for this idea. The bot is easy to use. You put on a headband and when you concentrate, the bot moves. Focus more and it goes faster. And it's a real robot too; it avoids edges so that it stays on the table.

The robot part was based on soccer bot from Make: Arduino Bots and Gadgets. We read the EEG with a NeuroSky MindWave. The early model had to use a computer as a gateway between Arduino and MindWave, because we were running the Mind Wave software and our own Python program on the computer. EEG is the recording of electrical activity of the brain from the scalp, produced by neurons firing in the brain. The brain cortex produces tiny electrical voltages (1–100 μV on the scalp). EEG doesn't read your thoughts, but it can tell your general state. For example, EEG can show if you are paying attention or meditating. The tiny voltages are easily masked by electrical noise from muscles and ambient sources. EEG currents are measured in microvolts (μV), which are millionths of a volt: 1 μV = 0.001 mV = 10^-6 V.

Noise from muscle and eye movement can be quite powerful compared to this. In normal buildings, the electrical main's current radiates a 50Hz or 60Hz electromagnetic field. In a laboratory setting, EEG is usually measured in a room that has less interference. At home, the EEG unit must filter out the troublesome signals.

The fundamental plan of BCI is to translate user created patterns of brain activity into corresponding commands. For brain mapping we use a technique electroencephalogram (EEG), which works on the electric pulses generated in the brain, by extracting these nerves we can track the state of the mind, for example attention, meditation. An electroencephalogram (EEG) gives us with different frequencies which can be further decoded as the states of mind. For the people with disabilities, it can be used to control a car or it can be implemented for anything which needs to be handled for movement through wheels. Products are available with which control through voice recognition or some sort of sensors. With a step ahead of them we are using brain to control the proceedings.

A mind-controlled robot is a robot that uses brain signal for its control via BRIs (hereafter, brain-controlled robots refer to Mind-Bot: An EEG-based mind-controlled mobile robots only). A special attention over mind-controlled mobile robotic systems is necessary than other mind-controlled devices because mobile robots require higher safety since they are used to transport very disabled people. The systems used to develop these kinds of robots need better performance parallel pronouncing its higher accuracy. Robot intelligence includes heart rate measurement, obstacle/collision avoidance, and fire detection and pronounces presence of life. All of these are considered as an additional safety techniques of Mind-Bot.

It is a fast growing emergent technology in which researchers aims to build a direct channel between human brain and computer. The brain accepts and control a mechanical devices as a natural part of its representation of the body.These computer brain interface are designed to restore the sensor functions, transmit sensory information to brain and stimulate the brain through artificially generated electrical signals

1.1 Project Description

The ultimate purpose of a direct brain computer interface (BCI) is to allow an individual with severe motor disabilities to have effective control over devices. A BCI system detects the presence of specific patterns in a person's on going brain activity that relates to the person's intention to initiate control.

In our project we use a robotic car as a prototype for this idea we will control the robotic car by the attention ratio of the Nerousky mindwave as the speed of the robotic car and we use the line detector to make the robotic car in the terminal or the road. We make all the system full wireless and we use the Bluetooth shield with arduino to connect the hardware (Robotic car) with Neruosky(Bluetooth dongle) and the Laptop.

We divide our project to many tasks to reach the final prototype:

- The first task was dealing with the brain signals by using the thinkgear packet and creates the interface of the brain signals (Alpha- beta- gamma- theta-ratio of attention and mediation).

- The second task was using Arduino to control the rate of light of LED because it is similar to control the dc motor and we use the attention ration as the rate of the light.
- The third task was building the robotic car and we make a small test to control it by a simple program in Arduino to move forward and take a delay and move to the right and take a delay and move to the left then take a delay to move again to forward
- The fourth task was connecting the Bluetooth shield in the Arduino to control it wireless.
- The fifth task was about a small program by the C# to control the robot by form of the direction as the keyboard and the controlling was full wireless.
- The final task was use the first program in the first task to control the speed of the dc motor by divide the speed to 3 parts (small-medium-high).

1.2 WHAT IS EEG?

An electroencephalogram is a measure of the brain's voltage fluctuations as detected from scalp electrodes.It is an approximation of the cumulative electrical activity of neurons.

An electroencephalogram (EEG) is a test used to evaluate the electrical activity in the brain. Brain cells communicate with each other through electrical impulses. An EEG can be used to help detect potential problems associated with this activity.The test tracks and records brain wave patterns. Small, flat metal discs called electrodes are attached to the scalp with wires. The electrodes analyze the electrical impulses in the brain and send signals to a computer, where the results are recorded.The electrical impulses in an EEG recording look like wavy lines with peaks and valleys. These lines allow doctors to quickly assess whether there are abnormal patterns. Any irregularities may be a sign of seizures or other brain disorders.

1.3 ABOUT BCI

Brain-computer interface (BCI) is collaboration between a brain and a device that enables signals from the brain to direct some external activity, such as control of a cursor or prosthetic limb. The interface enables a direct communications pathway between the brain and the object to be controlled. In the case of cursor control, for example, the signal is transmitted directly from the brain to the mechanism directing the cursor, rather than taking the normal route through the body's neuromuscular system from the brain to the finger on a mouse.

A BCI is a system that measures CNS activity and converts it into artificial output that replaces, enhances, supplements, or improves natural CNS output and thereby changes the on-going interactions between the CNS and its external or internal environment (CNS-Central Nervous System).The CNS is composed of the brain and the spinal cord and is differentiated from the peripheral nervous system (PNS), which is composed of the peripheral nerves and ganglia and the sensory receptors. The unique features of CNS structures are their location within the meningeal coverings (i.e., meninges), their distinctive cell types and histology, and their role in integrating the numerous different sensory inputs to produce effective motor outputs. In contrast, the PNS is not inside the meninges, does not have the unique CNS histology, and serves primarily to bring sensory inputs to the CNS and to carry motor outputs from it. CNS activity comprises the electrophysiological, neurochemical, and metabolic phenomena (such as neuronal action potentials, synaptic potentials, neurotransmitter releases, and oxygen consumption) that occur continually in the CNS.

A BCI output could **replace** natural output that has been lost to injury or disease. Thus, someone who cannot speak could use a BCI to spell words that are then spoken by a speech synthesizer. Or one who has lost limb control could use a BCI to operate a powered wheelchair.

A BCI output could **restore** lost natural output. Thus, someone with a spinal cord injury whose arms and hands are paralyzed could use a BCI to control stimulation of the paralyzed muscles with implanted electrodes so that the muscles move the limbs.

Or one who has lost bladder function from multiple sclerosis could use a BCI to stimulate the peripheral nerves controlling the bladder so as to produce urination.

A BCI output could **enhance** natural CNS output. Thus, someone engaged in a task that needs continuous attention over a long time (e.g., driving a car or performing sentry duty) could employ a BCI to detect the brain activity preceding breaks in attention and then produce an output (such as a sound) that alerts the person and restores attention. By preventing the periodic attentional breaks that normally compromise natural CNS output, the BCI enhances the natural output.

A BCI output could **supplement** natural CNS output. Thus, someone controlling cursor position with a standard joystick might employ a BCI to choose items that the cursor reaches. Or a person could use a BCI to control a third (i.e., robotic) arm and hand. In these examples, the BCI supplements natural neuromuscular output with another, artificial output.

Lastly, a BCI output might possibly **improve** natural CNS output. For example, a person whose arm movements have been compromised by a stroke damaging sensorimotor cortex might employ a BCI that measures signals from the damaged.

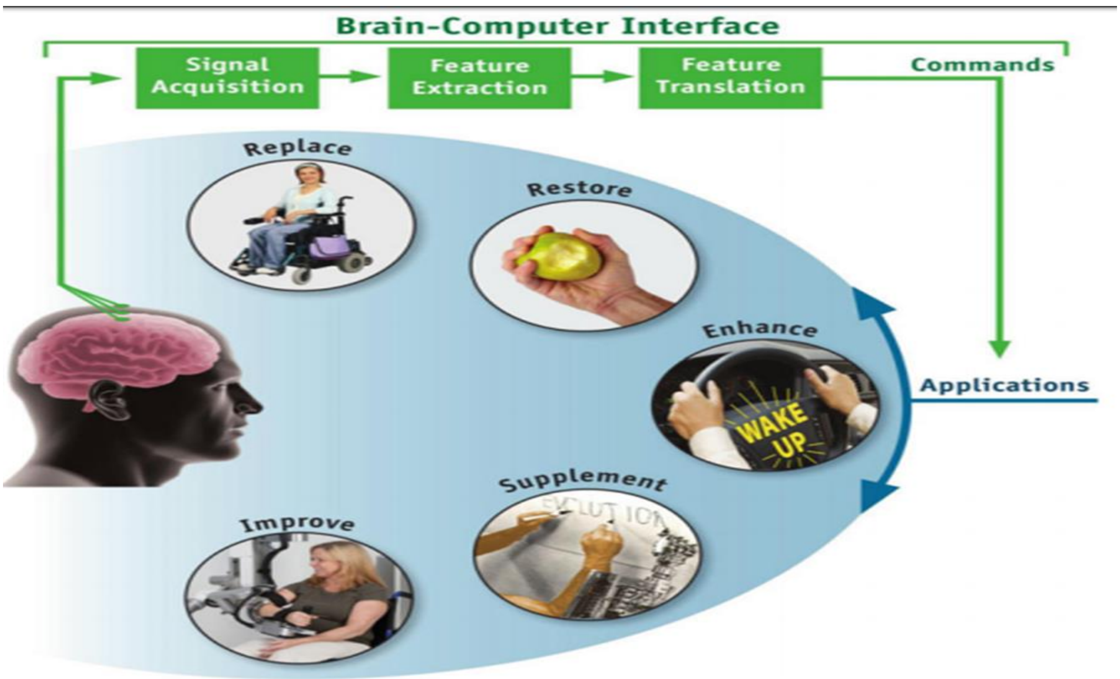


Fig 1.1: Basics of BCI

1.3.1 WORKING OF BCI:

The principle of BCI is based on to sense, transmit,analyse and appit the language of neurons.it consists of senso that is implanted in the cortex of the brain and th device analysize the brain signal.the signal generated by the brain are interepted and translated into cursor movement on the computer screen to control the coputer.

Objective: The goal is to develop fast and reliable connection between brain and computer which can be effectively used for disabled person move on their own.

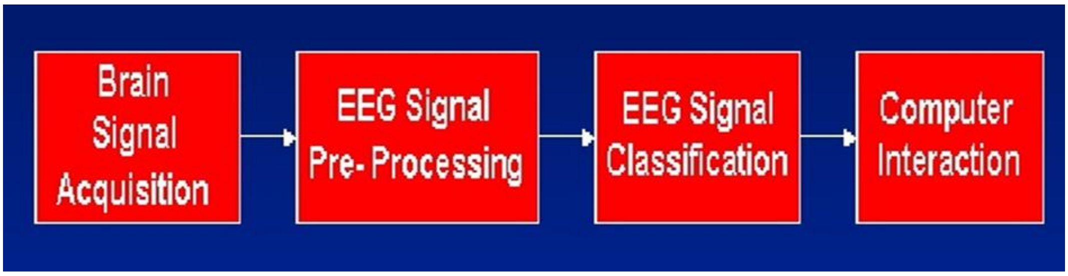


Fig 1.2: Block diagram of BCI

1.3.2 HISTORY:

Research on BCIs began in the 1970s at the University of California, Los Angeles (UCLA) under a grant from the National Science Foundation, followed by a contract from DARPA The papers published after this research also mark the first appearance of the expression *brain–computer interface* in scientific literature. In 1980s a report was given on control of a physical object, a mobile robot, using EEG signals A brain–computer interface (BCI), sometimes called a mind-machine interface (MMI), direct neural interface (DNI), or brain–machine interface (BMI), is a direct communication pathway between an enhanced or wired brain and an external device. BCIs are often directed at researching, mapping, assisting, augmenting, or repairing human cognitive or sensory-motor functions.

In 1924 Berger was the first to record human brain activity by means of EEG. By analyzing EEG traces, Berger was able to identify oscillatory activity in the brain, such as the alpha wave (8–12 Hz), also known as Berger's wave Berger's first recording device was very rudimentary. He inserted silver wires under the scalps of his patients.

These were later replaced by silver foils attached to the patients' head by rubber bandages. Berger connected these sensors to a Lippmann capillary electrometer, with disappointing results. More sophisticated measuring devices, such as the Siemens double-coil recording galvanometer, which displayed electric voltages as small as one ten thousandth of a volt, led to success.

Berger analyzed the interrelation of alternations in his EEG wave diagrams with brain diseases. EEGs permitted completely new possibilities for the research of human brain activities.

1.3.3 BASIC COMPONENETS OF BCI:

The 4 main steps involved in acquiring a brain signal are as follows:

1. Signal capture system
2. Signal processing system
3. Pattern recognition
4. Device control

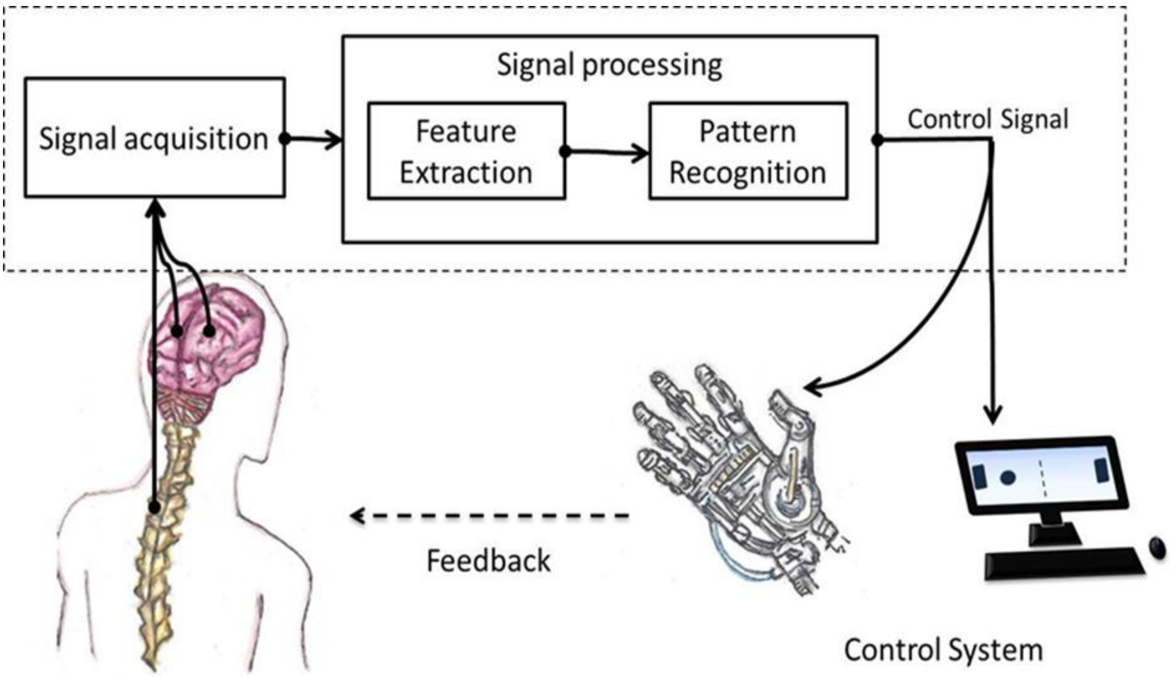


Fig 1.3: Basic flow of Signal

- The signal capture includes the electrodes and isolated electronic amplifiers.
- The signal processing includes the algorithm for linear prediction of signal
- Pattern recognition used to compose the neural network as to recognize which neuron are producing these signals.
- Interfaces are developed to enable control activity.

1.3.4 TYPES OF BCI:

They are of three types

1. Invasive
2. Non-invasive
3. semi-invasive

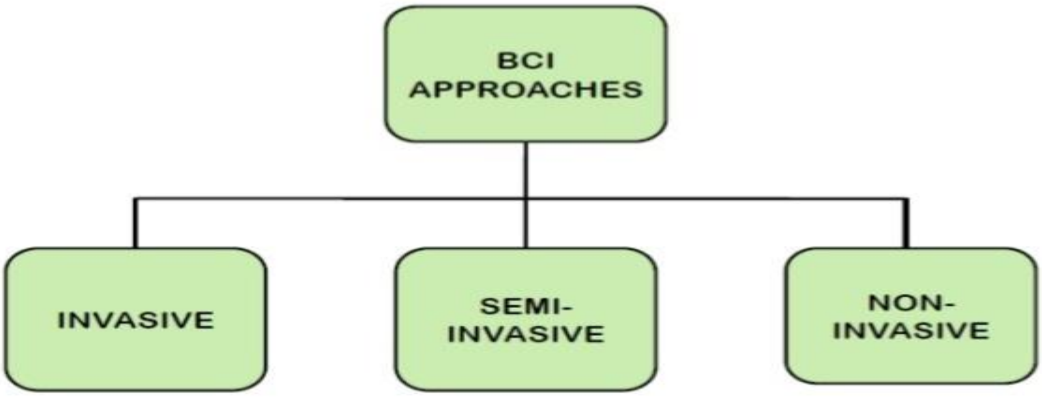


Fig 1.4: Types of BCI

Invasive: these are generally implanted in the brain through surgical procedure in the grey matter of the cortex area.

Non-invasive : they don't need any surgical procedures, the are just like wearable virtual reality devices

Semi-invasive: these are implemented inside the skull through surgical procedure but rest outside the grey matter of the brain

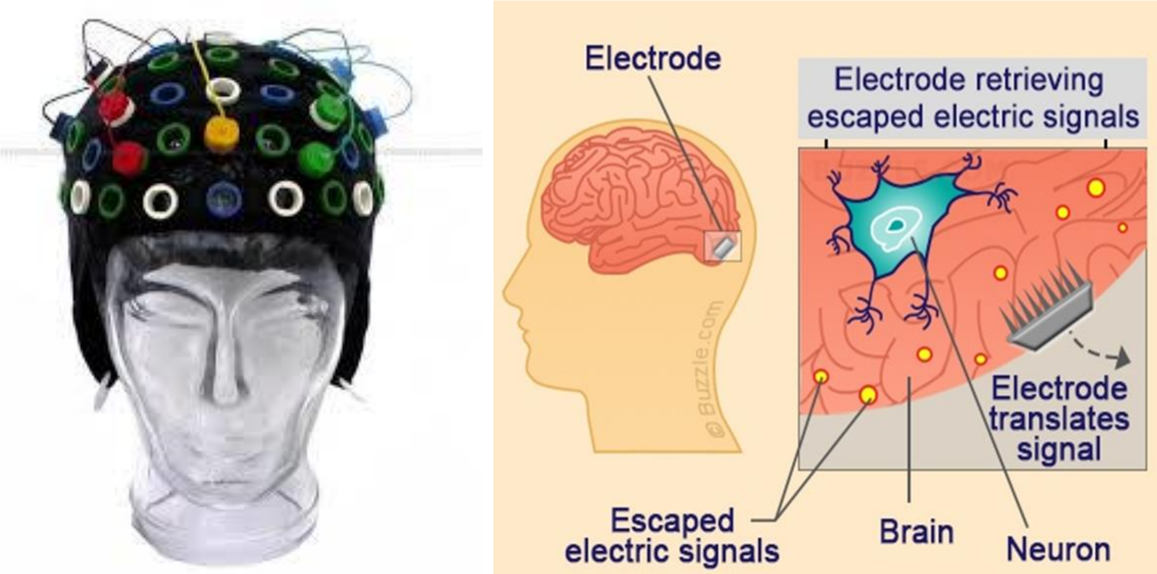


Fig 1.5: **Connection with Brain**

1.4 ABOUT BRAIN WAVES

Brain waves are generated by the building blocks of your brain -- the individual cells called neurons. Neurons communicate with each other by electrical changes. We can actually see these electrical changes in the form of brain waves as shown in an EEG (electroencephalogram).Brain waves are measured in cycles per second (Hertz; Hz is the short form). We also talk about the "frequency" of brain wave activity. The lower the number of Hz, the slower the brain activity or the slower the frequency of the activity.

Brainwaves are produced by synchronised electrical pulses from masses of neurons communicating with each other. Brainwaves are detected using sensors placed on the scalp. They are divided into bandwidths to describe their functions are shown below

1. Infra low---- <0.5Hz
2. Delta--------- upto 4 Hz
3. Theta--------- 4 to <8 Hz
4. Alpha ------- 8 to 13 Hz
5. Beta---------- >13 to 30 Hz
6. Gamma------ 30 to 100+ Hz

Delta: has a frequency of 3 Hz or below. It tends to be the highest in amplitude and the slowest waves. It is typical as the dominant beat in infants up to one year and in stages 3and 4 of sleep. It is typically most prominent frontally in grown-ups (e.g. FIRDA - Frontal Intermittent Rhythmic Delta) and posterior in children e.g. OIRDA – Occipital Intermittent Rhythmic Delta).

Theta: has a frequency of 3.5 to 7.5 Hz and is classified as "moderate" movement. It is perfectly typical in children upto13 years and in sleep however irregular in awake grown-ups. It can likewise be seen in generalized circulation in diffuse disorders, for example, metabolic encephalopathy or some instances of hydrocephalus.

Alpha: has a recurrence somewhere around 7.5 and 13 Hz.It is generally best found in the back districts of the head on every side, being higher in sufficiency on the overwhelming side. It shows up when shutting the eyes and unwinding, and vanishes when opening the eyes or alarming by any component (considering, figuring). It is the real cadence found in ordinary loose grown-ups.

Unconscious		**Conscious**		
Delta	**Theta**	**Alpha**	**Beta**	**Gamma**
0.5 – 4 Hz	4 – 8 Hz	8 – 13 Hz	13 – 30 Hz	30 – 42 Hz
Instinct	**Emotion**	**Consciousness**	**Thought**	**Will**
-Survival -Deep Sleep -Coma	-Drives -Feelings -Trance -Dreams	-Awareness of the body -Integration of feelings	-Perception -Concentration -Mental Activity	-Extreme Focus -Energy -Ecstasy

Table 1.1: Brainwaves, Frequencies Aad Functions

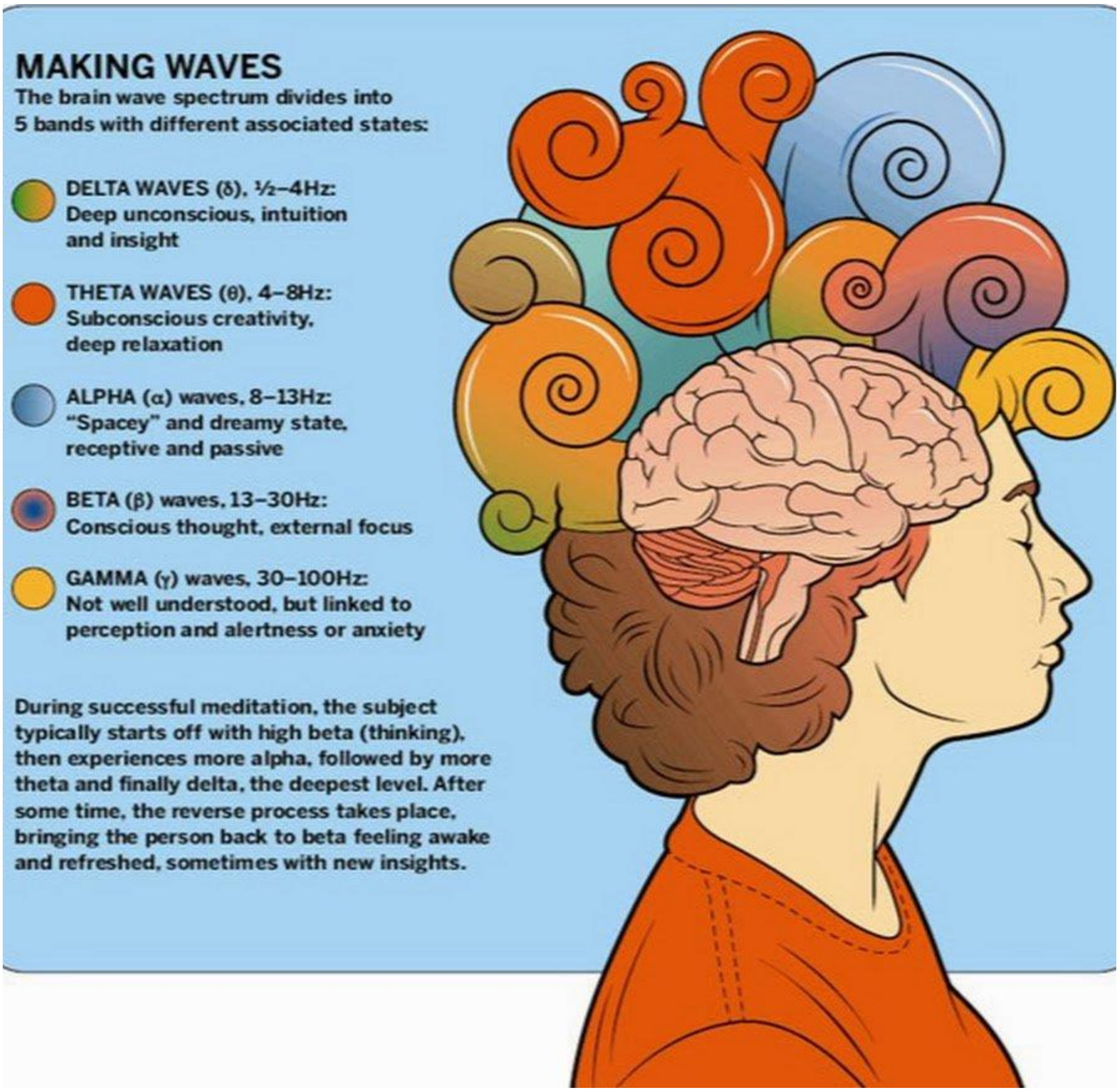


Fig 1.6: Waves In Brain Signal

Beta: Beta movement is "quick" action. It has a recurrence of 14 and more prominent Hz. It is generally seen on both sides in symmetrical conveyance and is most clear frontally. It is emphasized by narcotic trance-like medications particularly the benzodiazepines and the barbiturates. It might be non-attendant or lessened in.

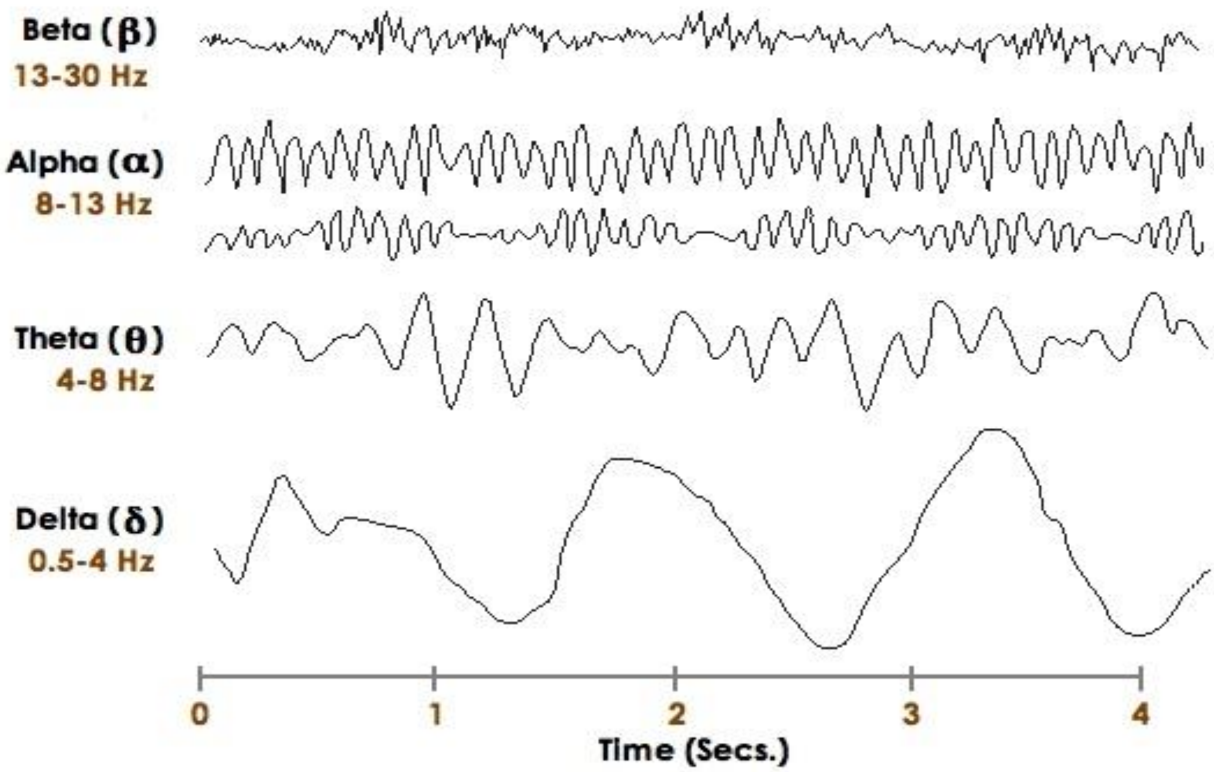


Fig 1.7: Graphical Representation Of Brain Waves

INFRA-LOW (<.5HZ)

Infra-Low brainwaves (also known as Slow Cortical Potentials), are thought to be the basic cortical rythms that underlie our higher brain functions. Very little is known about infra-low brainwaves. Their slow nature make them difficult to detect and accurately measure, so few studies have been done. They appear to take a major role in brain timing and network function.

DELTA WAVES (.5 TO 3 HZ)

Delta brainwaves are slow, loud brainwaves (low frequency and deeply penetrating, like a drum beat). They are generated in deepest meditation and dreamless sleep. Delta waves suspend external awareness and are the source of empathy. Healing and regeneration are stimulated in this state, and that is why deep restorative sleep is so essential to the healing process.

THETA WAVES (3 TO 8 HZ)

Theta brainwaves occur most often in sleep but are also dominant in deep meditation. It acts as our gateway to learning and memory. In theta, our senses are withdrawn from the external world and focused on signals originating from within. It is that twilight state which we normally only experience fleetingly as we wake or drift off to sleep. In theta we are in a dream; vivid imagery, intuition and information beyond our normal conscious awareness. It’s where we hold our ‘stuff’, our fears, troubled history, and nightmares.

ALPHA WAVES (8 TO 12 HZ)

Alpha brainwaves are dominant during quietly flowing thoughts, and in some meditative states. Alpha is ‘the power of now’, being here, in the present. Alpha is the resting state for the brain. Alpha waves aid overall mental coordination, calmness, alertness, mind/body integration and learning.

BETA WAVES (12 TO 38 HZ)

Beta brainwaves dominate our normal waking state of consciousness when attention is directed towards cognitive tasks and the outside world. Beta is a ‘fast’ activity, present when we are alert, attentive, engaged in problem solving, judgment, decision making, and engaged in focused mental activity.

Beta brainwaves are further divided into three bands;

1. Lo-Beta (Beta1, 12-15Hz) can be thought of as a 'fast idle, or musing.
2. Beta (Beta2, 15-22Hz) is high engagement or actively figuring something out.
3. Hi-Beta (Beta3, 22-38Hz) is highly complex thought, integrating new experiences, high anxiety, or excitement.

Continual high frequency processing is not a very efficient way to run the brain, as it takes a tremendous amount of energy.

GAMMA WAVES (38 TO 42 HZ)

Gamma brainwaves are the fastest of brain waves (high frequency, like a flute), and relate to simultaneous processing of information from different brain areas. It passes information rapidly, and as the most subtle of the brainwave frequencies, the mind has to be quiet to access it. Gamma was dismissed as 'spare brain noise' until researchers discovered it was highly active when in states of universal love, altruism, and the ‘higher virtues’. Gamma is also above the frequency of neuronal firing, so how it is generated remains a mystery. It is speculated that Gamma rhythms modulate perception and consciousness, and that a greater presence of Gamma relates to expanded consciousness and spiritual emergence.

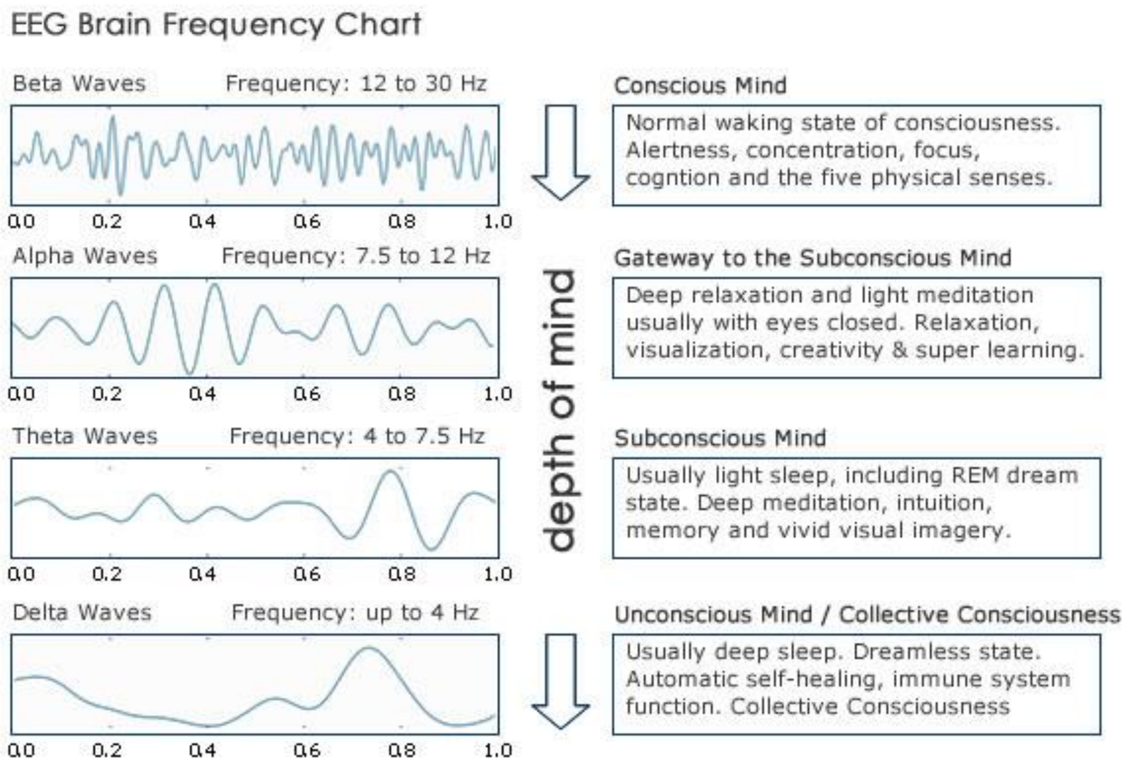


Fig 1.8: State Of Mind

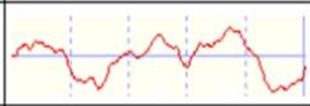
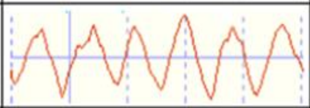
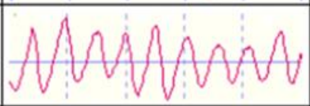
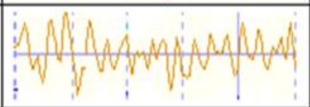
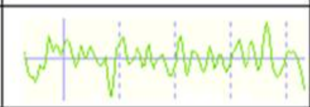
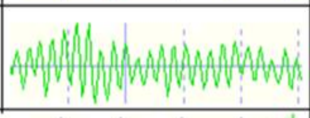
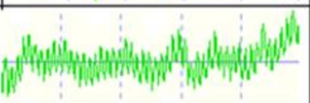
EEG Bands				
Band	Picture	Description	Positive	Negative
Delta 0-1.5hz		Non-Rhythmic slow, Glial cells control it	Sets up communication networks in the brain	Too much seen in brain injury, learning problems
Delta 2-3hz		Rhythmic slow	Empathy, calm, sleep	Brain Idling: Seen in trauma, poor focus and self control
Theta 4-7hz		Rhythmic slow	Emotional processing, drowsy	Brain Idling: Seen in poor focus and self control
Alpha 8-11hz		Rhythmic slow	Awake, calm, unfocussed	Brain Idling: poor focus, self control, anxiety
Beta I 12-15hz		Fast wave; SMR sensory-motor rhythm	Brain Active: Alert, calm, focussed	Anxiety, agitation, can't calm down, also poor focus and self control
Beta II 15-18hz		Fast wave	Brain Active: Sharp focus, energetic	Anxiety, agitation, can't calm down, also poor focus and self control
Beta III 18-25hz		Very fast wave	Brain extremely active, vigorous physical activity	Anxiety, agitation, can't slow down, mind racing, also poor focus and self control
Gamma 38-42hz		Very fast wave	Intense brain activity, high level processing	Too much = brain can't switch off; too little = can't focus

Table 1.2: Details Of Brain Waves

CHAPTER -2

NEUROSKY MINDWAVE HEADSET

2.1 INTRODUCTION

NeuroSky, Inc. is a manufacturer of Brain-Computer Interface (BCI) technologies for consumer product applications, which was founded in 2004 in Silicon Valley, California The company adapts electroencephalography (EEG) and electromyography (EMG) technology to fit a consumer market within a number of fields such as entertainment (toys and games), education, automotive, and health. NeuroSky technology allows for low-cost EEG-linked research and products by using inexpensive dry sensors; older EEGs require the application of a conductive gel between the sensors and the head. The systems also include built-in electrical “noise” reduction software/hardware, and utilize embedded (chip level) solutions for signal processing and output

The NeuroSky MindSet is a brainwave sensing headset which uses a ‘medical grade’ probe to capture brain patterns and translate them into stuff you can do with a computer.

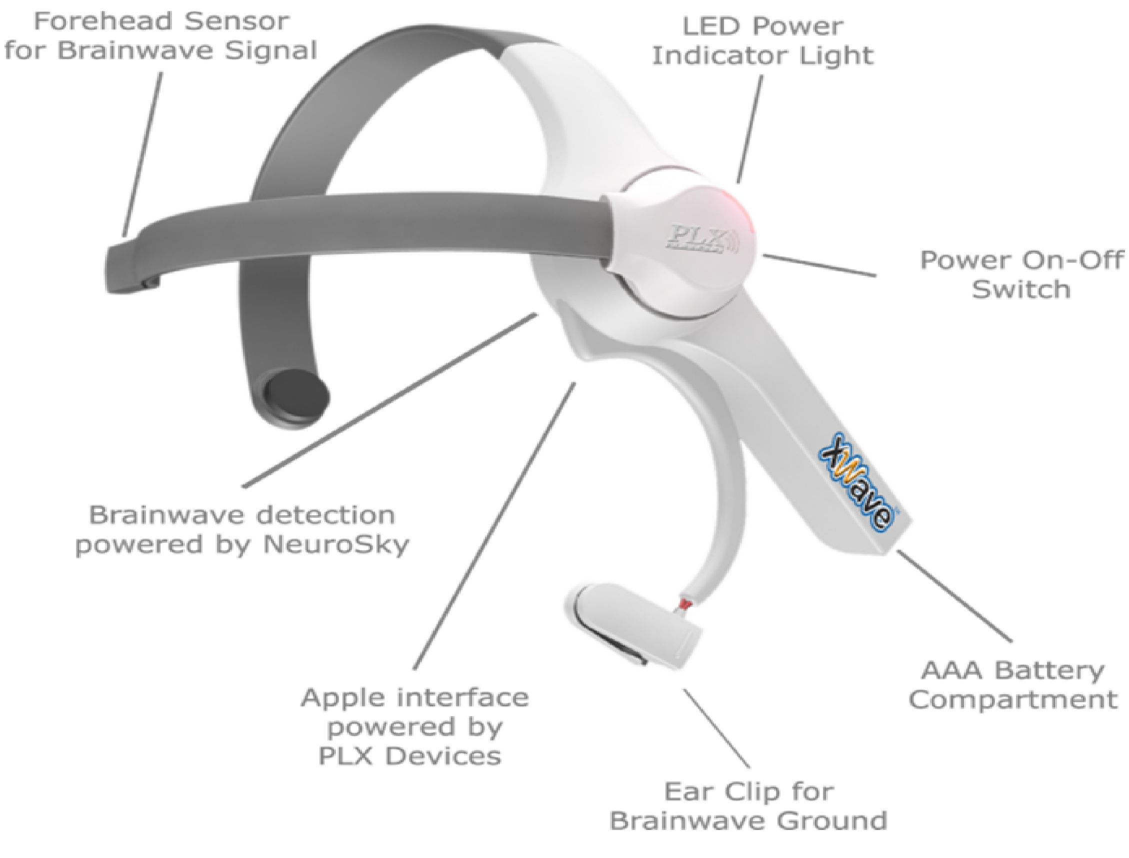


Fig 2.1: Mindwave Headset

2.1.1 Setting up your Mind Wave:

Minimum System Requirements:

	PC	Mac
Operating system	Windows XP/Vista/7	Mac OS X 10.5.8 or later
Processor	CoreDuo or equivalent	
Memory	1GB or more	
Video	DirectX 9.0 or greater	Intel GMA900 or greater
Hard disk	1GB free disk space	
USB	An available USB port	

Table 2.1: OS Requirements

2.1.2 Replacing The Battery:

MindWave headset requires 1 AAA battery to operate. To install or replace the battery, slide open the battery cover. Remove any existing battery within and replace with a new AAA battery

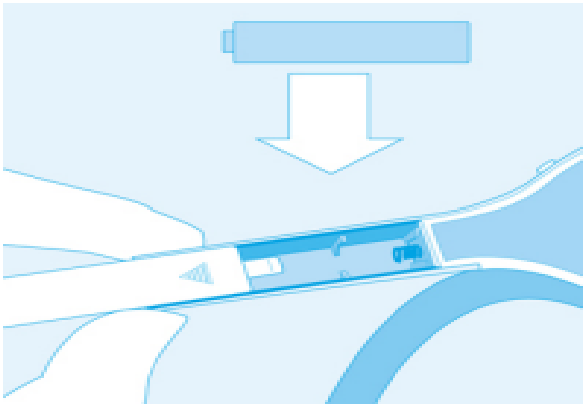


Fig 2.2: Battery

2.1.3 Power:

To power on the MindWave, slide the switch to the ON position. To turn the MindWave off, slide the switch back to the OFF position. While the MindWave is powered on, the LED light on the side of the headset will be turned on. If the MindWave has a low battery, the LED light will flash to indicate low battery status.

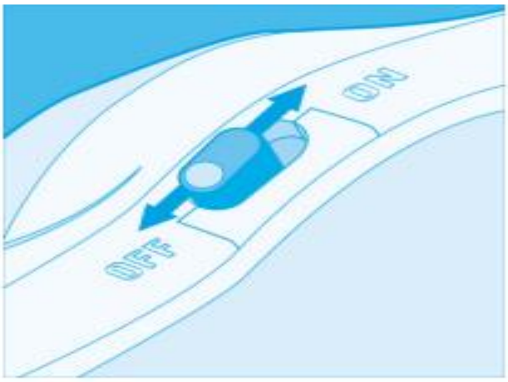


Fig 2.3: ON/OFF Switch

2.1.4 LED Light:

MindWave's LED light has has two colors: red and blue. Refer to the chart to see what state the MindWave is in.

Light color	Blinking	MindWave State
Off		Powered off.
Red	No	Powered on but not connected
Blue	No	Powered on and connected
Red or Blue	Yes	Low battery

Table 2.2: LED Light States

2.1.5 Installing in Windows:

Windows XP/Vista/7

1. Insert the supplied MindWave Application Disc into your disc drive.

2. Run the MindWave-Setup.exe installer if prompted or double-click on MindWave-Setup.exe

3. Follow the installer's on-screen instructions.

4. Insert the MindWave Wireless USB adapter when prompted (If you already plugged it in earlier, unplug and replug it into a different USB port).

5. MindWave Manager will launch. Follow the on-screen instructions.

6. Select Scan Automatically for MindWaves in range (recommended) and click Continue

7. MindWave Manager will scan for your MindWave. If found the MindWave will display your MindWave’s ID.

8. Your MindWave is now paired.

2.1.6 Fitting the MindWave:

MindWave is more than your average headset. It has the ability to use your brainwaves for exciting new applications.

Important: In order to take full advantage of these functions and features of the MindWave, the MindWave must be properly worn.

1. Orient the MindWave with the forehead Sensor Arm on your left hand side. Rotate the Sensor Arm from its base by about 90 degrees. It can be rotated slightly more if necessary to get proper ёt and comfort.
2. the overhead band of the MindWave is adjustable and can be extended to various sizes. Put on the MindWave. If the sensor does not make contact with the forehead or if it is not comfortable, remove the MindWave to readjust the overhead band and the forehead Sensor Arm. The forehead Sensor Arm is flexible and should arch inwards
3. Allow the rubber ear hoop to rest behind your left ear, and then clip the earclip onto your earlobe.
4. Make sure the two metal contacts on the inside of both sides of the earclip make skincontact with your earlobe or ear. Move any hair or obstructions (such as jewelry) out of the way. Readjust the earclip as necessary to make proper contact with the skin of your ear. You may need to squeeze the ear clip against your ear for a few moments.
5. Adjust the forehead Sensor Arm of the headset so that the Sensor Tip makes contact with the skin of your forehead. This Sensor Tip must maintain steady skin contact in order to properly measure your brainwaves. The Sensor Tip should be comfortable, yet stay firmly in position. Keep hair away from the sensor – the sensor must be able to directly contact the skin at all times. Make up, dead skin, or debris can interfere with the connection. Scratch or wipe the obstruction away if you have trouble obtaining a clean signal.

6. this is how the MindWave should look when properly worn. During usage, if you are not receiving a signal, repeat the steps above to make minor adjustments to ensure the sensor and contacts have proper skin contact.

Note: If you are having a problems with the signal quality, try sitting still for a few seconds. Note that talking can sometimes interfere with the signal quality as well. If this does not work, check that your head is not within a few feet of a strong electrical device (like a laptop adapter or an electrical outlet).

2.1.7 ThinkGear:

ThinkGear is the technology inside every NeuroSky product or partner product that enables a device to interface with the wearers' brainwaves. It includes the sensor that touches the forehead, the contact and reference points located on the ear pad, and the on-board chip that processes all of the data. Both the raw brainwaves and the eSense Meters (Attention and Meditation) are calculated on the ThinkGear chip.

2.1.8 eSense:

eSense is a NeuroSky's proprietary algorithm for characterizing mental states. To calculate eSense, the NeuroSky ThinkGear technology amplifies the raw brainwave signal and removes the ambient noise and muscle movement. The eSense algorithm is then applied to the remaining signal, resulting in the interpreted eSense meter values. Please note that eSense meter values do not describe an exact number, but instead describe ranges of activity. The eSense meters are a way to show how effectively the user is engaging Attention (similar to concentration) or Meditation (similar to relaxation). Like exercising an unfamiliar muscle, it may take some time to gain full proficiency with each of the eSense meters. In many cases, people tend to be better at one eSense than the other when they first begin. We recommend trying different tactics until you are successful with one. Once you see a reaction on the screen from your efforts, you will be able to duplicate the action more easily with additional practice.

Generally, Attention can be controlled through a visual focus. Focus on a singular idea. Try to "funnel" your concentration and focus your train of thought towards pushing up the meter. Other suggestions include picking a point on the screen to stare at or imagining the action you are trying to accomplish happening. For example, look at the Attention eSense meter and imagine the dial moving towards higher numbers.

For Meditation, it typically helps to try to relax yourself. Connect to a sense of peace and calm by clearing your mind of thoughts and distractions. If you are having difficulty engaging Meditation, close your eyes, wait a number of seconds, and then open your eyes to see how the meter has responded.

If you have trouble at first in controlling your eSense meter levels, be patient. Try different techniques and practice. Also be sure to read and try to understand the Technical Description in order to get a better idea about how eSense actually works under the hood.

- **ATTENTION eSense:** The eSense Attention meter indicates the intensity of a user's level of mental "focus" or "attention", such as that which occurs during intense concentration and directed (but stable) mental activity. Its value ranges from 0 to 100. Distractions, wandering thoughts, lack of focus, or anxiety may lower the Attention meter level. See eSense Meter – General Information for details about interpreting eSense levels in general.
- **MEDITATION eSense:** The eSense Meditation meter indicates the level of a user's mental "calmness" or "relaxation". Its value ranges from 0 to 100. Note that Meditation is a measure of a person's mental states, not physical levels, so simply relaxing all the muscles of the body may not immediately result in a heightened Meditation level. However, for most people in most normal circumstances, relaxing the body often helps the mind to relax as well. Meditation is related to reduced activity by the active mental processes in the brain. It has long been an observed effect that closing one's eyes turns off the mental activities which process images from the eyes. So closing the eyes is often an effective method for increasing the Meditation meter level. Distractions, wandering thoughts, anxiety, agitation, and sensory stimuli may lower the meditation meter levels. See eSense Meter – General Information for details about interpreting eSense levels in general.

Fig 2.4: Brain Waves In Mindwave Application

CHAPTER-3

ARDUINO

3.1 WHAT IS ARDUINO?

Arduino is an open source electronics prototyping platform based on flexible, easy-to-use hardware and software. It’s intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments.

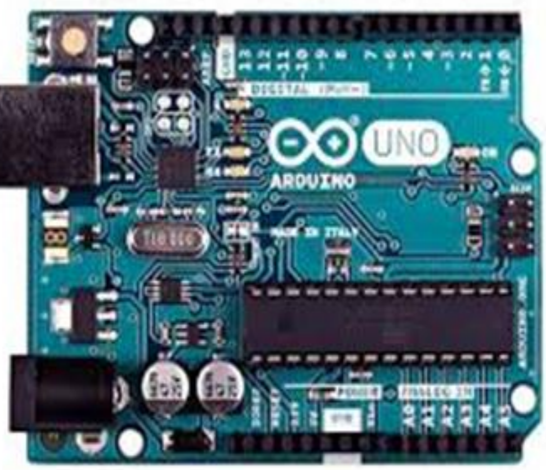


Fig 3.1 Arduino Uno

It’s an open-source physical computing platform based on a microcontroller board,and a development environment for writing software for the board. In simple words,Arduino is a small microcontroller board with a USB plug to connect to your computer and a number of connection sockets that can be wired up to external electronics, such asmotors,relays,light sensors, laser diodes,loudspeakers,microphones, etc. They can either be powered through the USB connection from the computer or from a 9V battery. They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently

3.1.1 Microcontroller:

Microcontroller can be described as a computer embedded on a rather small circuit-board. To describe the function of a microcontroller more precisely it is a single chip that can perform various calculations and tasks and send/receive signals from other devices via the available pins.

Precisely what tasks and communication with the world it does, is what is governed by what instructions we give to the Microcontroller. It is this job of telling the chip what to do, is what we refer to as programming on it.However, the uC by itself, cannot accomplish much; it needs several external inputs: power, for one; a steady clock signal, for another. Also, the job of programming it has to be accomplished by an external circuit. So typically, a uC is used along with a circuit which provides these things to it; this combination is called a microcontroller board. The Arduino Uno that you have received is one such microcontroller board. The actual microcontroller at its heart is the chip called Atmega328

The advantages that Arduino offers over other microcontroller boards are largely in terms of reliability of the circuit hardware as well as the ease of programming and using it.

3.1.2 Technical specifications:

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)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB

EEPROM	1 KB
Clock Speed	16 MHz

Power:

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

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

• 5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

• 3.3V: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

• GND: Ground pins

Memory:

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output:

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

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication 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 analogReference() 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 analogReference().
- 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.

3.1.3 Pin specification:

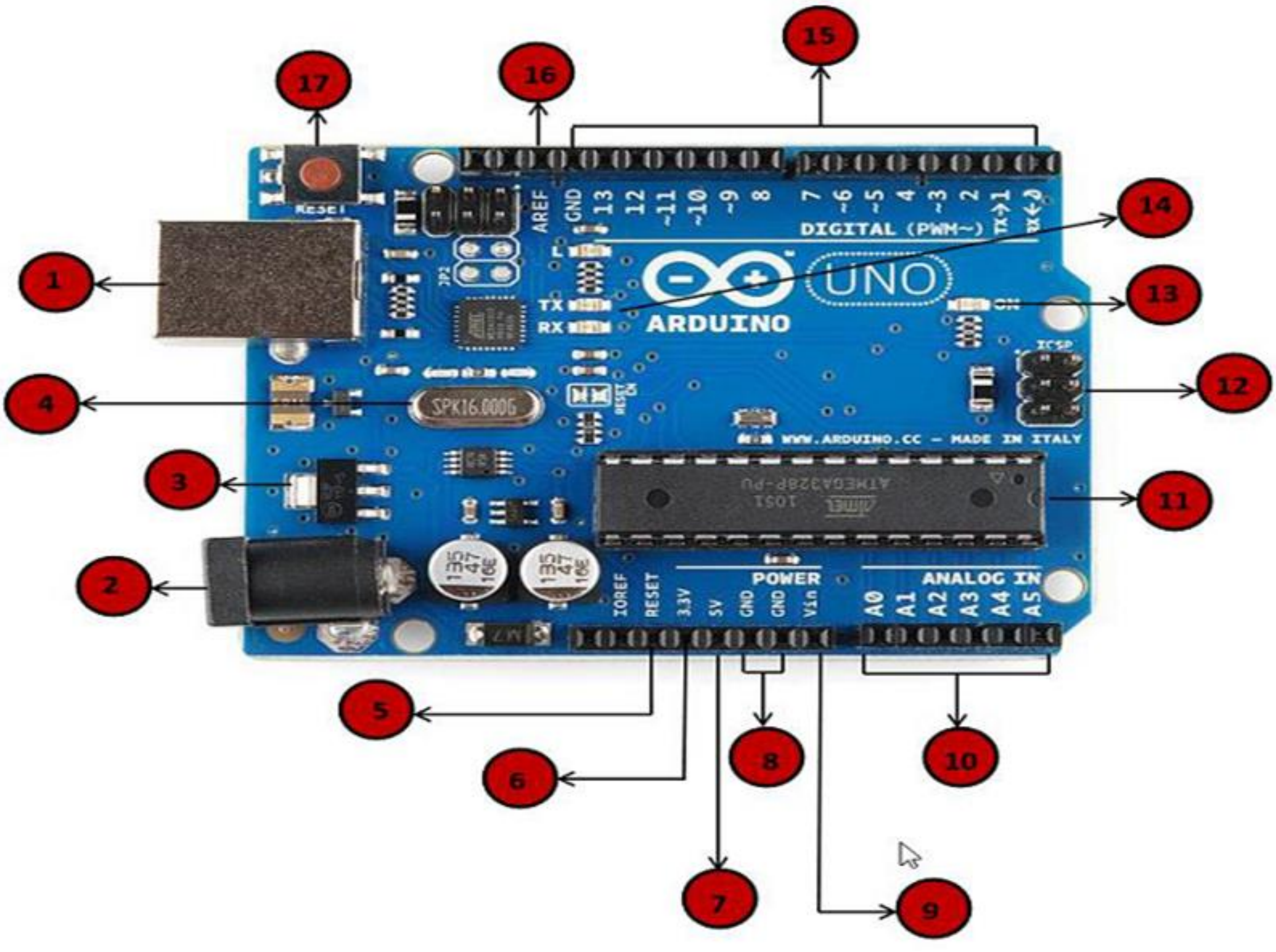


Fig 3.2 Arduino Uno Pins

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).

2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

	• GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
7	**Analog pins** The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
8	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
9	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
10	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to

	indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
11	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
12	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
13	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.2 ARDUINO TYPES

There have been many revisions of the USB Arduino, some of them are:

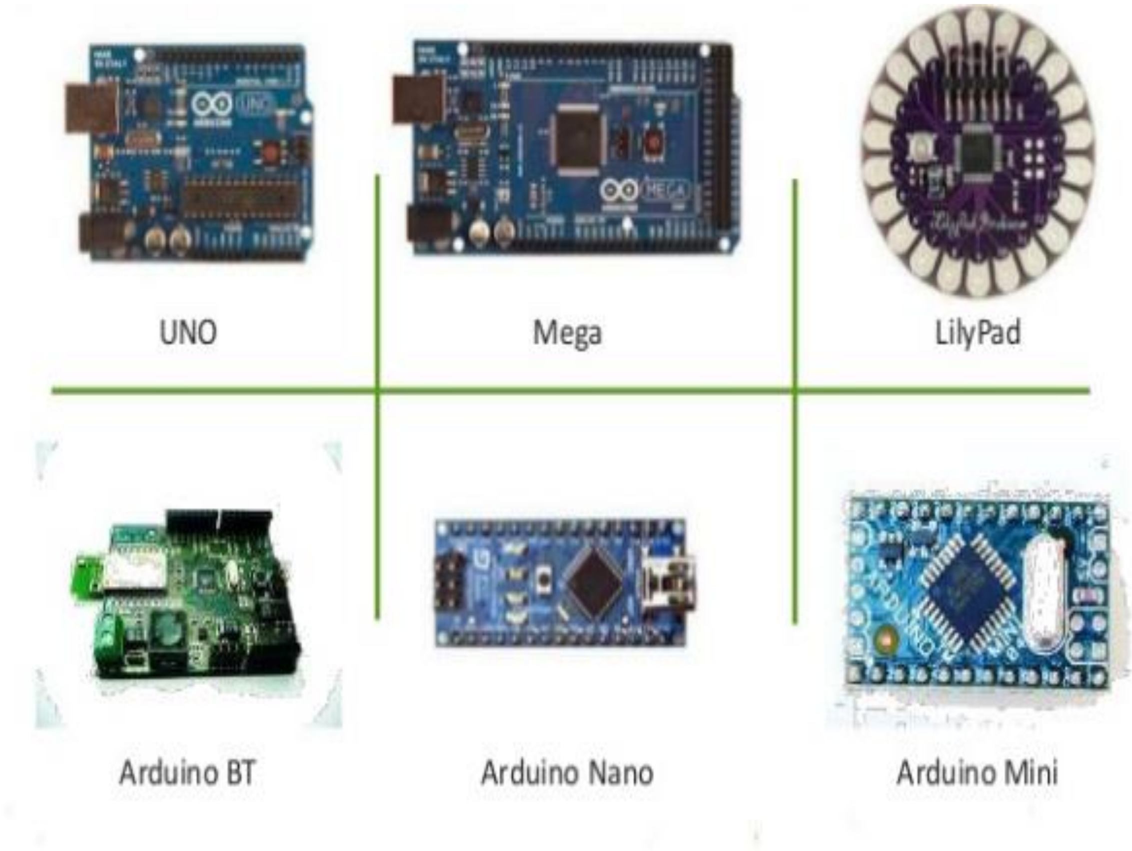


Fig 3.3: Types Of Arduino

Arduino UNO:

This is the latest revision of the basic Arduino USB board. It connects to the computer with a standard USB cable and contains everything else you need to program and use the board. It can be extended with a variety of shields: custom daughter-boards with specific features. It is similar to the Duemilanove, but has a different USB-to-serial chip the ATMega8U2,and newly designed labelling to make inputs and outputs easier to identify.

Arduino Mega 2560:

A larger, more powerful Arduino board. Has extra digital pins, PWM pins, analog inputs, serial ports, etc. the version of the Mega released with the Uno, this version features the Atmega2560, which has twice the memory, and uses the ATMega 8U2 for USB-to-serial communication. ARDUINO-Infinity exist.

Arduino Duemilanove:

The Duemilanove automatically selects the appropriate power supply (USB or external power), eliminating the need for the power selection jumper found on previous boards. It also adds an easiest to cut trace for disabling the auto-reset, along with a solder jumper for re-enabling it.

Note: around March 1st, 2009, the Duemilanove started to ship with the ATmega328p instead of theATmega168.

Arduino Fio:

An Arduino intended for use as a wireless node. Has a header for an XBee radio, a connector for a LiPo battery, and a battery charging circuit.

LilyPad Arduino:

A stripped-down, circular Arduino board designed for stitching into clothing and other fabric/flexible applications. Needs an additional adapter to communicate with a computer.

Arduino Diecimila:

The main change in the Arduino Diecimila is that it can be reset from the computer, without the need to physically press the reset button on the board. The Diecimila uses a low dropout voltage regulator which lowers the board's power consumption when powered by an external supply (AC/DC adapter or battery). Are settable poly fuse protects your computer's USB ports from shorts and surges. It also provides pin headers for the reset line and for 3.3V.There is a built-in LED on pin 13. Some blue Diecimila boards say "Prototype – Limited Edition" but are in fact fully-tested production boards (the actual prototypes are red)

Arduino Extreme

The Arduino Extreme uses many more surface mount components than previous USB Arduino boards and comes with female pin headers. It also has RX and TX LEDs that indicate when data is being sent to or from the board

Arduino Mini 04

On this version of the Arduino Mini, two of the pins changed. The third pin became reset (instead of ground) and fourth pin became ground (instead of being unconnected). These boards are labelled "Mini 04"

3.2.1 Features of the Arduino (IDE):

• Simple to use

• Open Source (Free)

• Programming style similar to C

• You can download it from www.arduino.cc

3.3 INSTALLING ARDUINO ON YOUR COMPUTER

Download Arduino Integrated Design Environment (IDE) here (Most recent version: 1.6.5): https://www.arduino.cc/en/Main/Software

This is the Arduino IDE once it's been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.

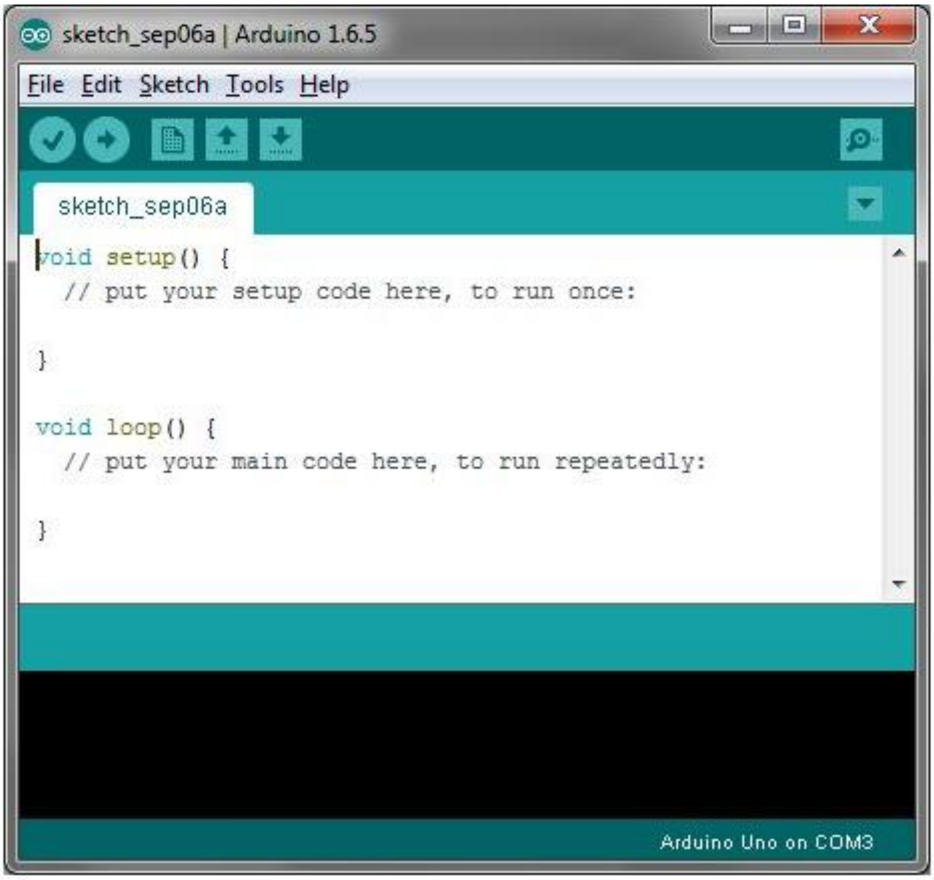
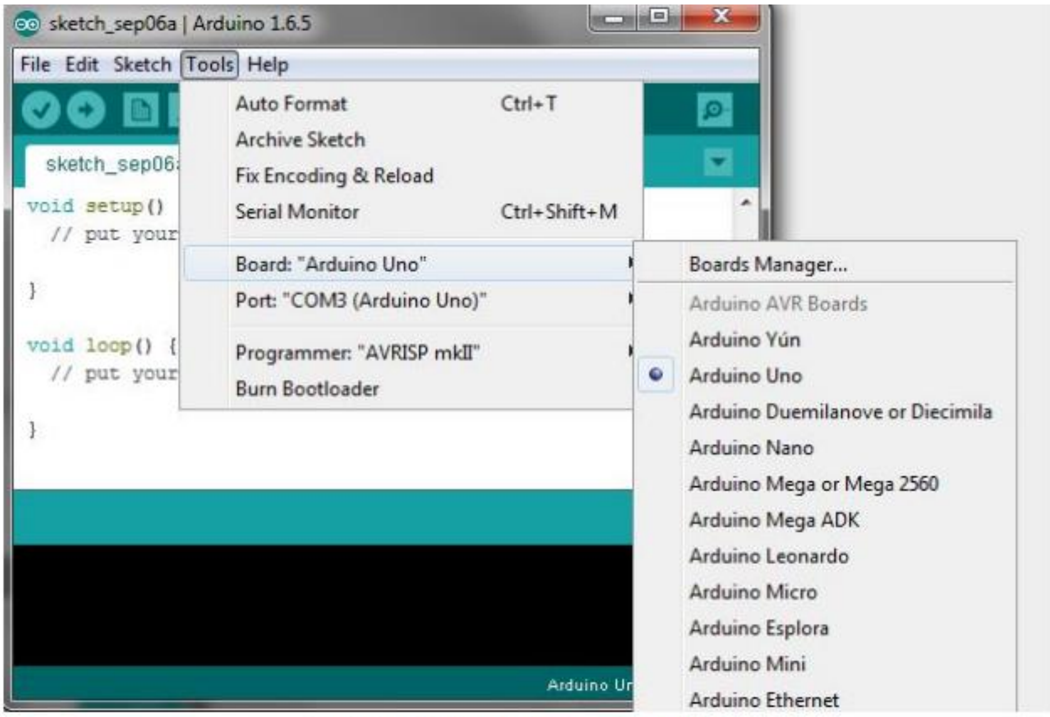


Fig3.4 : Basic Default Window Of Arduino

To program the Arduino board :

- you must first download the development environment from here www.arduino.cc/en/Main/Software.
- Choose the right version for your operating system.
- Download the file and double-click on it to open it ; on Windows or Linux this creates a folder named arduino-[version], such as arduino-1.0
- Drag the folder to wherever you want it: your desktop, your Program Files folder (on Windows), etc. On the Mac, double-clicking it will open a disk image with an Arduino application (drag it to your Applications folder).
- Now whenever you want to run the Arduino , you'll open up the arduino (Windows and Linux) or Applications folder (Mac), and double-click the Arduino icon. Don't do this yet, though; there is one more step.

- Now you must install the drivers that allow your computer to talk to your board through the USB



Arduino IDE: Board Setup Procedure

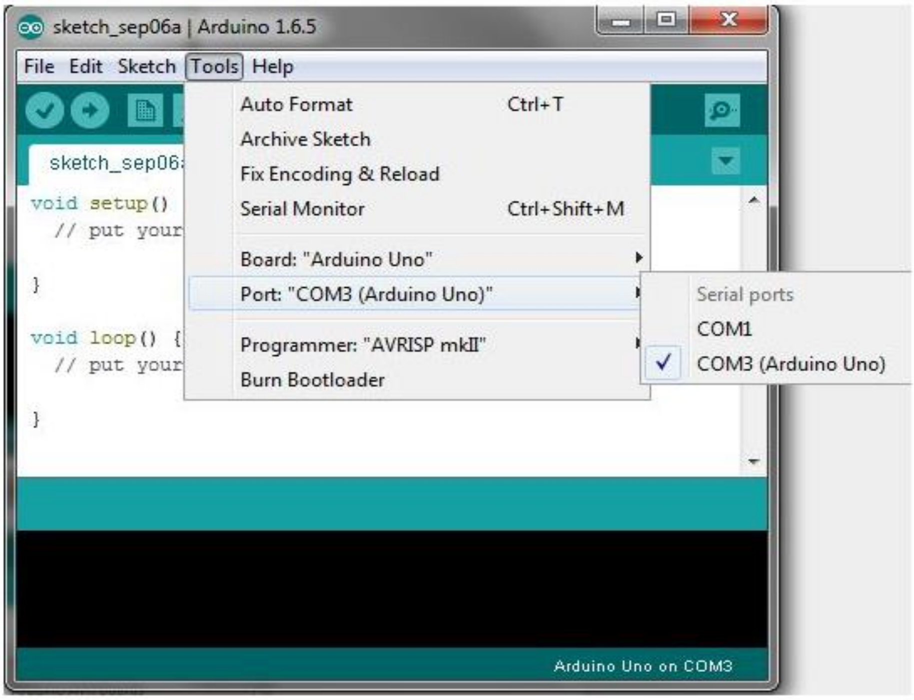
Fig3.5: Set Up Procedure

3.4 Software

The software used by the arduino is Arduino IDE.he Arduino IDE is a cross-platform application written inJava,and is derived from the IDE fortheProcessing programming languageand the Wiringproject. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes acode editor with features such assyntax highlighting,brace matching,and automatic indentation,and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to editmakefilesor run programs on acommand-line interface.Althoughbuilding on command-line is possible if required with some third-party tools such asIno. The Arduino IDE comes with aC/ C++ library called "Wiring" (from the project of the samename), which makes many common input/output operations much easier. Arduino programs arewritten in C/C++, although users only need define two functions to make a runnable program:

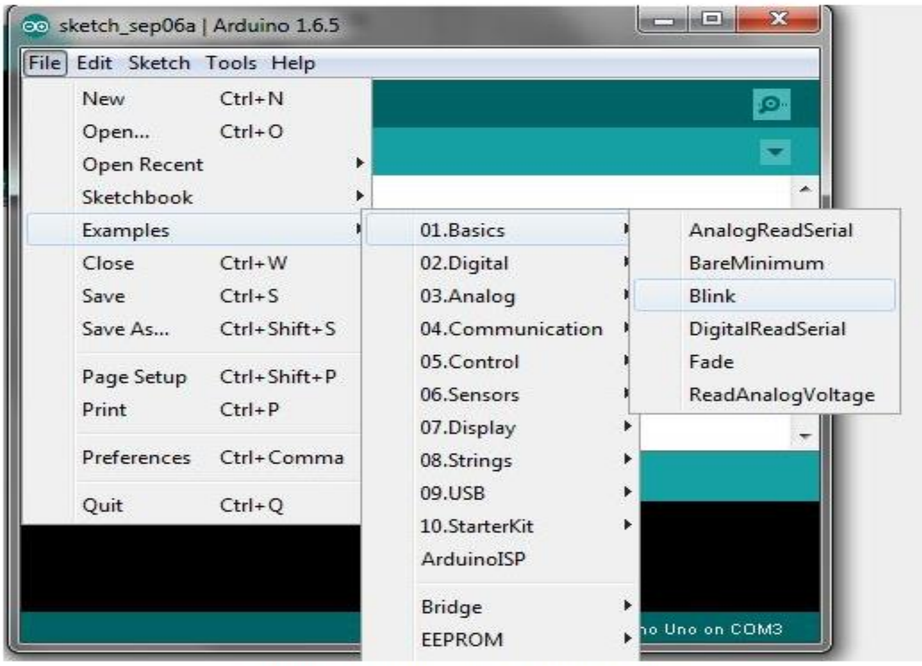
- setup( )– a function run once at the start of a program that can initialize settings

- loop( ) – a function called repeatedly until the board powers off



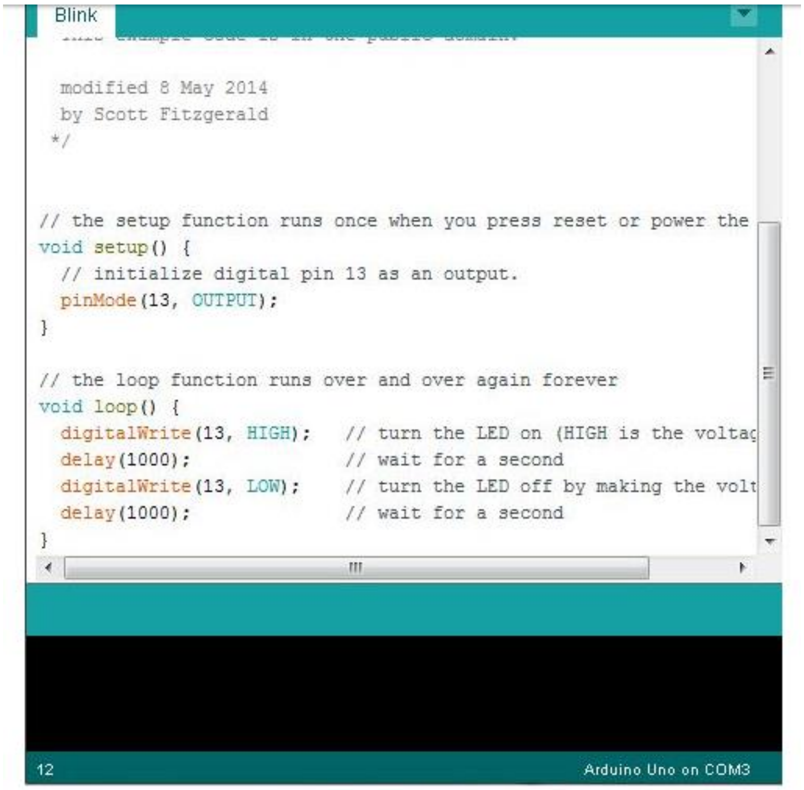
Arduino IDE: COM Port Setup

Fig3.6 : Com Port Set Up



Arduino IDE: Loading Blink Sketch

Fig3.7: Example Program Blink Code



Arduino IDE: Uploading Blink

Fig3.8: Uploading Blink Program

3.5 APPLICATIONS OF ARDUINO

Arduino was basically designes to make the process of using electronics inmultidisciplinary projects more accessible.It is intended for artists,designers,hobbyists,andanyone interested in creating interactive objects or environments.Arduino can sense theenvironment by receiving input from a variety of sensors and can affect its surroundings bycontrolling lights,motors,and other actuators.because of these features,arduino finds extensiveapplication in various fields.Arduino projects can be stand-alone or they can communicate withsoftware running on a computer.Arduino received an Honarary Mention in Digital Communication section of the 2006 ArsElectronica PrixArduino is used by all class of people in a different way.some students use it in theirprojects,some using arduino for fun,some went out to become entreupreuners.This only showshow useful is this tiny device

ARDUINO is spreading rapidly across the globe.Arduino is actually an open source hardwareproject that can be pro grammed to read temperatures, control a motor, and sense touch.the Arduino is both a cute, blue micro controller platform that fits nicely in the palm of yourhand and an expanding community of developers who support it, distributed across two dozencoun tries, four continents, and counting.

Thousands of projects have been done worldwide using this tiny little device.some of which to mention are:

- Simple room temperature readout
- Interactive real-time auditory feedback system
- GPS receiver Module
- Ultrasonic Sensor
- Infrared detectors
- SONAR
- Various sensor projects like
- Keypad security code
- Sensor tube for heart monitor
- Pulse rate monitor

3.6 Guide Summary

1. Download and install Arduino IDE (https://www.arduino.cc/en/Main/Software)

2. Plug in your Arduino Board

3. Select the proper board in the IDE (Tools>Boards>Arduino Uno)

4. Select the proper COM port (Tools>Port>COMx (Arduino Uno))

5. Open the “Blink” sketch (File>Examples>Basics>01.Blink)

6. Press the Upload button to upload the program to the board

7. Confirm that your board is working as expected by observing LED

CHAPTER -4

BLUETOOTH HC-05 MODULE

4.1 INTRODUCTION

It is an easy to use **Bluetooth SPP (Serial Port Protocol) module**, designed for transparent wireless serial connection setup .The HC-05 Bluetooth Module can be used in a Master or Slave configuration .By default the setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections .Master module can initiate a connection to other devices, making it a great solution for wireless communication.

The Bluetooth module which is used mostly is the HC-05 Bluetooth Module. This module is cheap, readily available, and quick to acquire. It is also designed to ensure compatibility with Arduino microcontrollers. Furthermore, it is highly customizable, such as allowing its name to be changed, modifying its access password, and switching from master mode to slave mode. This last feature is important because it will enable future development of the Sensor Interface to add wireless Bluetooth sensor reading functionality. To communicate with Bluetooth sensors, the module should be in Master mode. However, when communicating with the Android device, the module should be in Slave mode. This Bluetooth module’s ability to switch between these 2 modes will allow the Sensor Interface to communicate with Bluetooth sensors as wired sensors.

4.1.1 Hardware features:

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

4.1.2 HC-05 Specification:

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm

First thing you need to do is identify your module. It can be either HC05 or HC06. Both the modules are same in functionality except the pin-out. Also HC05 can act as both master and slave whereas HC06 functions only as slave. It's hard to differentiate between the two only by seeing. One probable way would be checking the back of the breakout board. If it has "JY-MCU" written on the back, it's probably a HC06. ours has "ZS-040" written and it is a HC05.

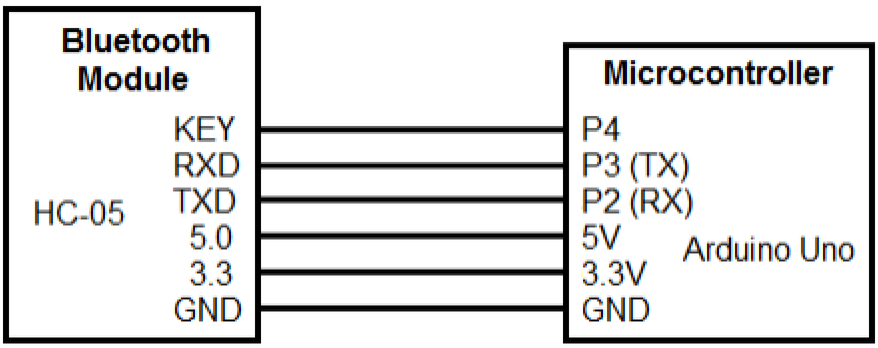


Fig 4.1: Microcontroller - Bluetooth Module Connectivity

Above Figure depicts the connections between the Bluetooth module and the microcontroller. The module has both a 5.0V input pin and a 3.3V input pin. Although the module may operate correctly with only one of these supply pins connected, both are connected to the microcontroller’s output voltage pins to guarantee proper functionality. The Bluetooth module’s RX pin must connect with the microcontroller’s TX pin, and its TX pin must connect to the microcontroller’s RX pin. The Arduino Uno’s default TX serial pin is P1 and the default RX pin is P0. However, these were not used for a couple of reasons. Most importantly, when P1 and P0 were used as TX and RX pins, the Bluetooth module would not communicate with the microcontroller. This meant that different pins had to be used. Fortunately, an Arduino library exists to change digital pins to RX/TX pins, called Software Serial. Using this library, P3 was changed to a TX pin and P2 was changed to an RX pin. This decision has the added benefit of freeing up P1 and P0. This is beneficial because the Arduino has a useful debugging and monitoring feature called Serial Monitor. Using this, the Arduino can write messages to a computer screen when connected to a computer. However, this feature only works when P1 and P0 are not in use. Therefore, by using P3 and P2 for TX and RX instead of P1 and P0, testing and debugging was a much easier and faster process. The Bluetooth module has one more input pin, labeled KEY. This pin is used for changing its mode between Master mode and Slave mode, as well as customizing its other.

4.2 PIN DESCRIPTION

The HC-05 module has 6 pins. They are as follows:

ENABLE: When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e., the module remains on and communication also takes place.

VCC: Supply Voltage 3.3V to 5V

GND: Ground pin

TXD & RXD: These two pins acts as an UART interface for communication

STATE: It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device signal goes low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes high. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

BUTTON SWITCH: This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

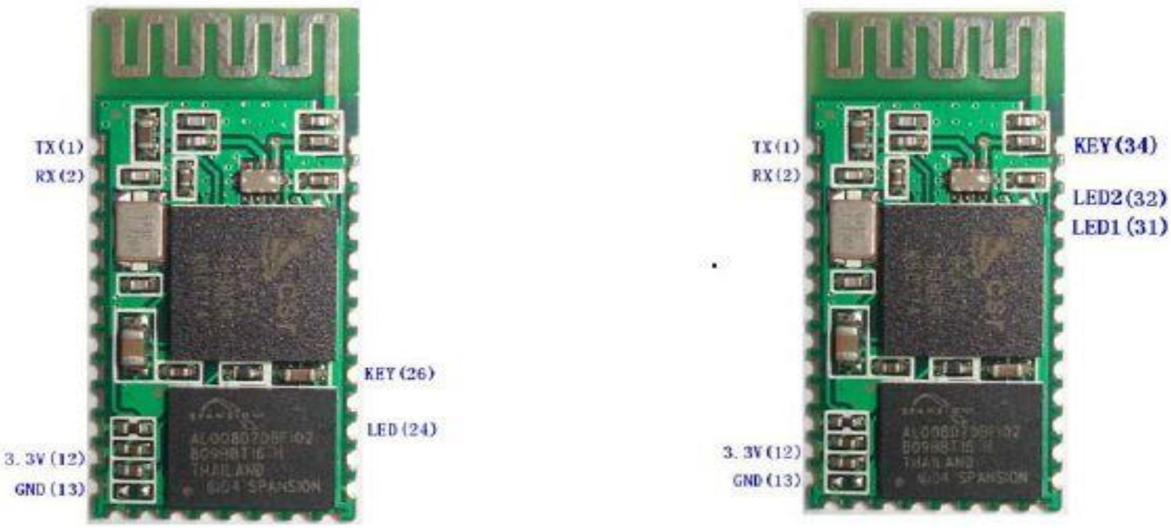


Figure 1 HC-06 Figure 2 HC-05

Fig 4.2 HC-05 & HC-06 Modules

PINS: The pins found on the breakout board are,

KEY: This pin has to be pulled high to enter AT mode.

RXD : Serial input pin.

TXD: Serial output pin.

Once you identified the module, now it's time to enter the AT mode.

4.3 AT Commands

Type "AT" (without the quotes) on the serial monitor and press enter. if "OK" appears then everything is all right and the module is ready to take command. Now you can change the name of the module, retrieve address or version or even reset to factory settings. To see the default name, type AT+NAME. The name will be prompted, by default it is HC-05 or JY_MCU or something like that. To change the name just type AT+NAME=your desired name.

Here is an important note, if the key pin is not high, i.e. not connected to Vcc while receiving AT commands(if you did not solder the wire and released it after the module entered AT mode), it will not show the default name even after giving right command. But you can still change the name by the command mentioned above. To verify if the name has really changed, search the device from your pc/mobile. The changed name will appear. To change baud rate, type AT+UART=desired baud rate. Exit by sending AT+RESET command.

4.4 SETTING UP CONNECTION

For HC05:

Once the module is in AT mode, open Arduino.IDE. Go to tools>serial port>select the com port your USB to TTL converter is connected to (to find out, go to device manager of your pc>ports(COM &LPT)) . Now open the serial monitor. The bt module is now communicating at a baud rate of 38400. So change the baud rate to 38400 at bottom right corner of the serial monitor. Also change "no line ending " to "both NL & CR" found just beside the baud

Hardware Connections:

As we know that Vcc and Gnd of the module goes to Vcc and Gnd of Arduino. The TXD pin goes to RXD pin of Arduino and RXD pin goes to TXD pin of Arduino i.e.,(digital pin 0 and 1).The user can use the on board Led. But here, led is connected to digital pin 12 externally for betterment of the process.

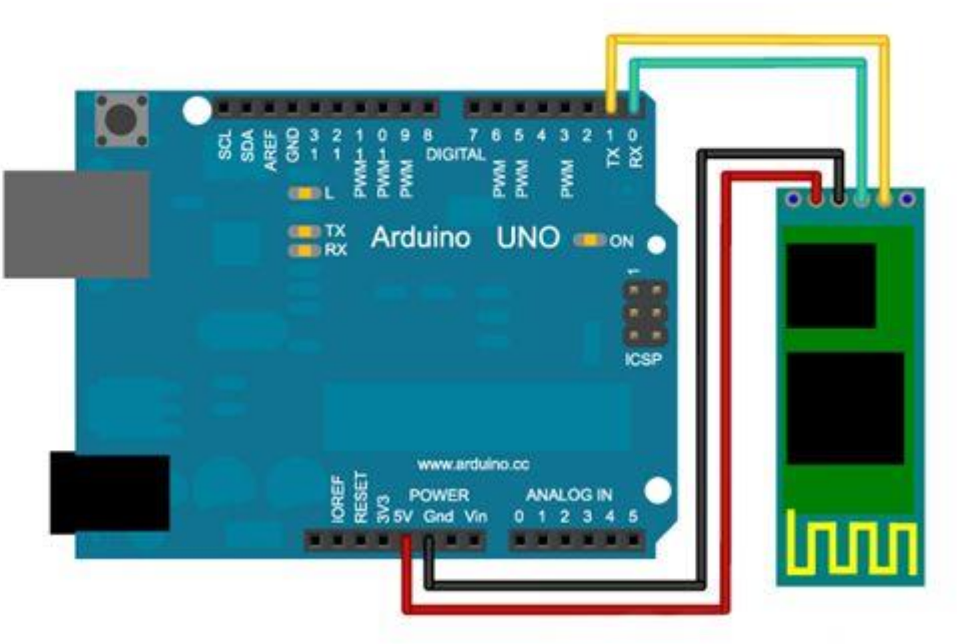


Fig 4.3: HC-05 & Arduino Interfacing

Note: HC-05 can be connected either to 5V or 3.3V.

4.4.1 Interface module:

First part "Connection to configure HC-05 module"

In this first post, as the title indicates, we will learn how to configure our HC-05 module using the Arduino board and later connect it to the NeuroSky MindWave Mobile diadem.

To start the configuration of the module using the Arduino, (in this example we will use the Arduino UNO board), it is necessary to connect previously the circuit shown below:.

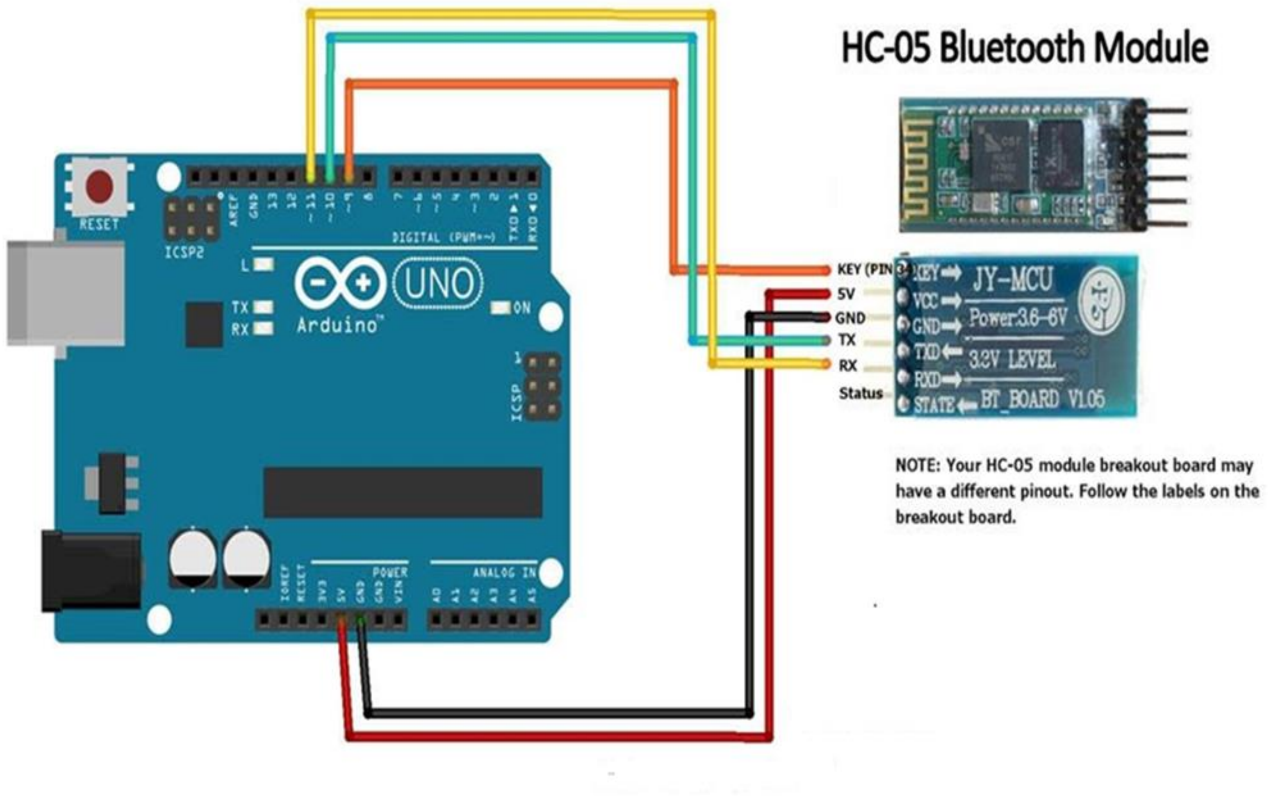


FIG 4.4: HC-05 Wiring Diagram

Once the circuit is armed, the "Sketch" must be loaded, which will allow us to enter the configuration mode of the HC-05, known as AT mode, through the Arduino IDE. When the Sketch and the armed circuit have been successfully loaded, we proceed to put the HC-05 in AT mode (without turning the Arduino off or on the PC-USB), for this it is only necessary to disconnect the voltage cable (VCC) from The Arduino board a moment and reconnect it, we will also notice that the Bluetooth LED will turn on and off every two seconds. If it all works out well in the Arduino IDE / tools / serial monitor, we change the speed values to "9600 baud" and "Both Nl and CR". At this point we will see the screen "Enter AT commands: If we send the AT command, we must reply OK.

Second part "Configuration of the HC-05 module"

Fig 4.4: AT Commands In Serial Monitor

Now we can start with the HC-05 device configuration. The following are the basic commands that we must know to configure our Bluetooth device:

Know the name of the module. AT + NAME ?

Change the name, in the following example it is changed to "MINOMBRE". AT + NAME = MINOMBRE

A very important fact the HC-05 functions as slave and master.

- As a slave: wait for another device to connect.
- As master: the HC-05 connects to another device.

Know how it is configured with the command: AT + ROLE?

- If you answer "0" it is in slave mode.

- If you answer "1" it is in master mode. By default it comes in slave mode.

- Switch to slave mode: Command: AT + ROLE = 0
- Change to master mode: Command: AT + ROLE = 1

Know the password of the HC-05 Command: AT + PSWD? By default comes the "1234" or the "0000".

Shows the configuration with the Arduino board, by default it is configured at a speed of 9600. Command: AT + UART?

Part Three "Configuration of the HC-05 module for connection to the MindWave headset"

To connect the bluetooth module to the Neurosky MindWave headset it is necessary to know the mac address of our MindWave NeuroSky headset, so we must connect it to a device which can be a computer or a cell phone.

To check the address in windows 7 we can go to home / devices and printers, we should have the device appear, this should have been added previously, otherwise it is necessary to go to start / add bluetooth device and select the diadem.

Fig 4.5: Determining Mac Address Of Mindwave

For the mindwave version move the address appears in hexadecimal. In this example it appears as 20: 68: 9d: c2: 24: 46 which debugging it is left as

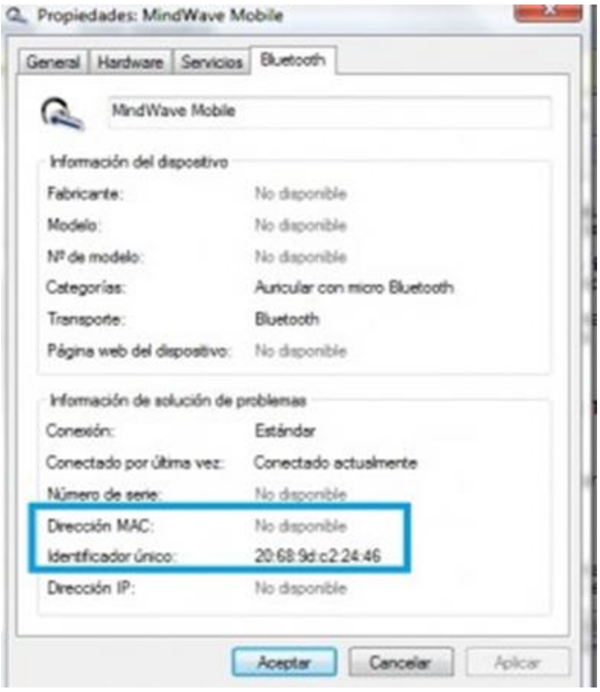


Fig 4.6: Mac Address Of Mindwave

"20689DC22446".At this point we will return to commands seen in the second part of this tutorial. The first thing is to proceed to configure the Bluetooth module as master mode, since this time we occupy that the HC-05 is connected to another device (in this case we occupy it to connect to the diadem), we enter the Arduino IDE with the Sketch , (Provided in the second part of this document),

Command: AT + UART = 57600,0,0 // We change the speed at which the module operates, originally we had it at 9600 and changed it to 57600.

To connect our headset to the HC-05, pay attention to the following steps: Set the HC-05 module as master mode Command: AT + ROLE = 1

We change the password so that they can connect, for this case of the Mindwave diadem it is necessary that the password is (0000). AT + PSWD = 0000

We prepare the module to connect to a specific address,

- // 0 = Device-specific address.
- // 1 = to any device that is available.

For this case it will be a specific address. Command: AT + CMODE = 0. The Bluetooth is connected to the Mac address of the slave, the leading zeros are omitted. The MAC address, as mentioned above, comes in a hexadecimal value which is debugged in this example as "20689DC22446" and must be entered in the format ####, ##, !!!!!!. Example is as follows: Command: AT + BIND = 2068,9D, C22446.

Now comes a part in which we must pay close attention we will execute the following commands in the following order:

- Command: AT + INIT \ R \ N Initialize SPP (Can not repeatedly initialize)
- Command: AT + IAC = 9E8B33 \ R \ N INQUIERE GENERAL INQUIERE ACCESS CODE
- Command: AT + CLASS = 0 \ R \ N CHECK ALL DEVICES TYPES
- Command: AT + INQM = 1,9,48 INQUIRE MODE: RSSI, MAX NUMBER 9, TIMEOUT 48
- Command: AT + INQ \ R \ N CHECK. Everything must dial OK.
- After this we can pair our devices with the following command: AT + PAIR = 2068,9D, C22446
- Finally we introduce the following command to finish connecting the devices. Command: AT + LINK = 2068,9D, C22446

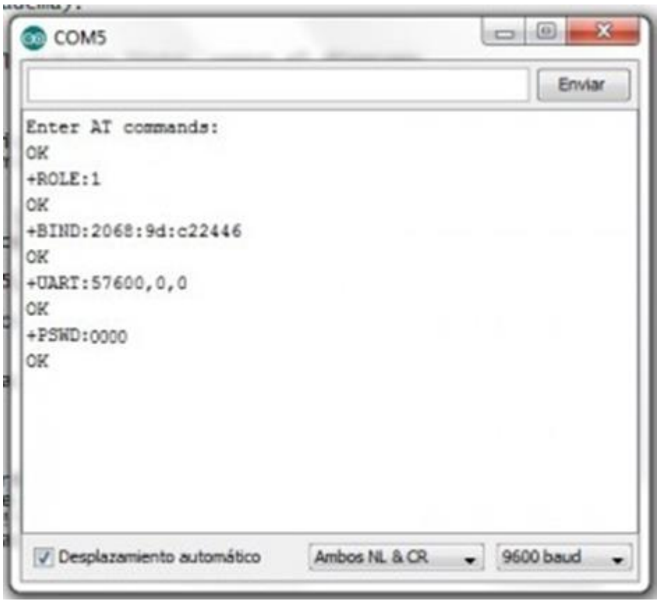


Fig 4.7: Serial Monitor Returning 'OK' Commands

Now disconnect the key cable and restart the module.

Part Four "Connection HC-05 through Arduino to MindWave NeuroSky diadem"

Now, we successfully configured the Bluetooth module to automatically connect to mindwave when they are both switched ON.

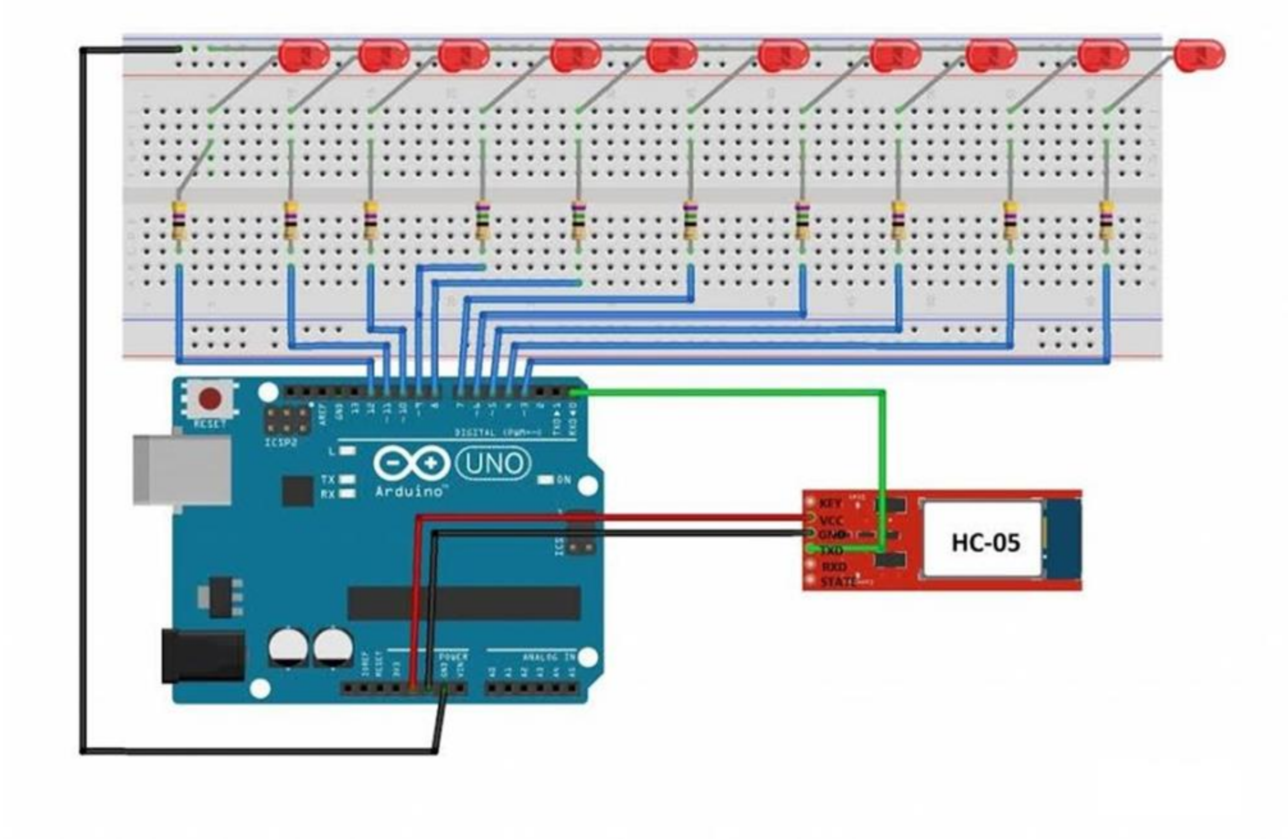


Fig 4.8: Arduino, Mindwave & Led

To ensure that it works properly let us verify their connection by verifying the attention and meditation parameters of our brain signals. When there is a poor connection no LED will blink, whenever attention level increases the LED's glow one by one in series according to our attention.

To make above concept operational let us upload code into Arduino UNO microcontroller to verify:

4.5 CODE

```
#define LED 13

#define BAUDRATE 57600

#define DEBUGOUTPUT 0


#define GREENLED1  3

#define GREENLED2  4

#define GREENLED3  5

#define YELLOWLED1 6

#define YELLOWLED2 7

#define YELLOWLED3 8

#define YELLOWLED4 9

#define REDLED1    10

#define REDLED2    11

#define REDLED3    12


#define powercontrol 10


// checksum variables

byte generatedChecksum = 0;

byte checksum = 0;

int payloadLength = 0;

byte payloadData[64] = {0};

byte poorQuality = 0;

byte attention = 0;
```

```
byte meditation = 0;

// system variables
long lastReceivedPacket = 0;
boolean bigPacket = false;

// Microprocessor Setup //
void setup()
{
	pinMode(GREENLED1, OUTPUT);
	pinMode(GREENLED2, OUTPUT);
	pinMode(GREENLED3, OUTPUT);
	pinMode(YELLOWLED1, OUTPUT);
	pinMode(YELLOWLED2, OUTPUT);
	pinMode(YELLOWLED3, OUTPUT);
	pinMode(YELLOWLED4, OUTPUT);
	pinMode(REDLED1, OUTPUT);
	pinMode(REDLED2, OUTPUT);
	pinMode(REDLED3, OUTPUT);
	pinMode(LED, OUTPUT);
	Serial.begin(BAUDRATE);

}
```

// Read data from Serial UART //

```
byte ReadOneByte()
{
        int ByteRead;
        while(!Serial.available());
        ByteRead = Serial.read();

       #if DEBUGOUTPUT
        Serial.print((char)ByteRead;
       #endif
        return ByteRead;
}
```

//MAIN LOOP//

```
void loop()
 {
  // Look for sync bytes
  if(ReadOneByte() == 170)
{
    if(ReadOneByte() == 170)
 {

      payloadLength = ReadOneByte();
      if(payloadLength > 169)
          return;
```

```
generatedChecksum = 0;
for(int i = 0; i < payloadLength; i++)
{
        payloadData[i] = ReadOneByte();
        generatedChecksum += payloadData[i];
 }
checksum = ReadOneByte();
generatedChecksum = 255 - generatedChecksum;
  if(checksum == generatedChecksum)
 {
  poorQuality = 200;
  attention = 0;
  meditation = 0;

  for(int i = 0; i < payloadLength; i++)
 {
       switch (payloadData[i])
        {
                case 2:
                  i++;
                  poorQuality = payloadData[i];
                  bigPacket = true;
                  break;
                case 4:
```

```
          i++;
          attention = payloadData[i];
          break;
        case 5:
          i++;
          meditation = payloadData[i];
          break;
        case 0x80:
          i = i + 3;
          break;
        case 0x83:
          i = i + 25;
          break;
        default:
          break;
    }
   }

#if !DEBUGOUTPUT

   if(bigPacket)
  {
        if(poorQuality == 0)
          digitalWrite(LED, HIGH);
        else
```

```
 digitalWrite(LED, LOW);
Serial.print("PoorQuality: ");
Serial.print(poorQuality, DEC);
Serial.print(" Attention: ");
Serial.print(attention, DEC);
Serial.print(" Time since last packet: ");
Serial.print(millis() - lastReceivedPacket, DEC);
lastReceivedPacket = millis();
Serial.print("\n");

switch(attention / 10)
{
	case 0:
		digitalWrite(GREENLED1, HIGH);
		digitalWrite(GREENLED2, LOW);
		digitalWrite(GREENLED3, LOW);
		digitalWrite(YELLOWLED1, LOW);
		digitalWrite(YELLOWLED2, LOW);
		digitalWrite(YELLOWLED3, LOW);
		digitalWrite(YELLOWLED4, LOW);
		digitalWrite(REDLED1, LOW);
		digitalWrite(REDLED2, LOW);
		digitalWrite(REDLED3, LOW);
		break;
```

```
case 1:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, LOW);
        digitalWrite(YELLOWLED1, LOW);
        digitalWrite(YELLOWLED2, LOW);
        digitalWrite(YELLOWLED3, LOW);
        digitalWrite(YELLOWLED4, LOW);
        digitalWrite(REDLED1, LOW);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
case 2:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, LOW);
        digitalWrite(YELLOWLED2, LOW);
        digitalWrite(YELLOWLED3, LOW);
        digitalWrite(YELLOWLED4, LOW);
        digitalWrite(REDLED1, LOW);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
```

```
case 3:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, LOW);
        digitalWrite(YELLOWLED3, LOW);
        digitalWrite(YELLOWLED4, LOW);
        digitalWrite(REDLED1, LOW);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
case 4:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, LOW);
        digitalWrite(YELLOWLED4, LOW);
        digitalWrite(REDLED1, LOW);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
```

```
case 5:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, HIGH);
        digitalWrite(YELLOWLED4, LOW);
        digitalWrite(REDLED1, LOW);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
case 6:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, HIGH);
        digitalWrite(YELLOWLED4, HIGH);
        digitalWrite(REDLED1, LOW);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
```

```
case 7:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, HIGH);
        digitalWrite(YELLOWLED4, HIGH);
        digitalWrite(REDLED1, HIGH);
        digitalWrite(REDLED2, LOW);
        digitalWrite(REDLED3, LOW);
        break;
case 8:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, HIGH);
        digitalWrite(YELLOWLED4, HIGH);
        digitalWrite(REDLED1, HIGH);
        digitalWrite(REDLED2, HIGH);
        digitalWrite(REDLED3, LOW);
        break;
```

```
case 9:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, HIGH);
        digitalWrite(YELLOWLED4, HIGH);
        digitalWrite(REDLED1, HIGH);
        digitalWrite(REDLED2, HIGH);
        digitalWrite(REDLED3, HIGH);
        break;
case 10:
        digitalWrite(GREENLED1, HIGH);
        digitalWrite(GREENLED2, HIGH);
        digitalWrite(GREENLED3, HIGH);
        digitalWrite(YELLOWLED1, HIGH);
        digitalWrite(YELLOWLED2, HIGH);
        digitalWrite(YELLOWLED3, HIGH);
        digitalWrite(YELLOWLED4, HIGH);
        digitalWrite(REDLED1, HIGH);
        digitalWrite(REDLED2, HIGH);
        digitalWrite(REDLED3, HIGH);
        break;
}
```

```
          }
      #endif
       bigPacket = false;
        }
    }
  }
}
```

CHAPTER – 5
BULIDING ROBOT

5.1 INTRODUCTION

Before programming anything, we'll build the chassis for the robot. Basically it's a traditional rover robot structure with four dc motors. To make it suitable for mind-controlling needs, we'll add a line detector and motor driver kit to communicate the motors with arduino. We use a solder less bread-board and Bluetooth shield for the Arduino, to make the system full wireless and communicate the ardunio with Neruosky by Bluetooth .We can adding components and wirers easy to arduino after we put the Bluetooth shield.

Here’s how all the major components will work together to create a working robot:

Arduino: This is the brains of the project. It is essentially a small embedded computer With a brain (a microcontroller), as well as header pins that can Connect to inputs (sensors) and outputs (actuators).

2WD Chassis: This holds everything together. It’s essentially the platform for the Robot.

Motors: These are motors that can be connected to (Motor Driver L293D) and then connect the motor driver to the Arduino. Arduino communicates with them by sending pulses to control speed and direction.

Motor Driver kit: This kit used to connect the 2 dc motors with arduino to communicate the speed and direction.

Line Detector: With the line detector, your robot will avoid a black line, which makes it stay in the arena (helpful for keeping it from falling off a table).

Mind Wave: Mind Wave measures your brainwaves and transmits the results for the Arduino. We have to add a Bluetooth shield to connect the arduino with Mind wave by Bluetooth dongle. We want to connect it directly to Arduino instead of a computer’s USB port

5.1.1 Tools and Parts:

Here parts and tools needed to make the robot. Feel free to improvise if you don't find the exact matches. We list the

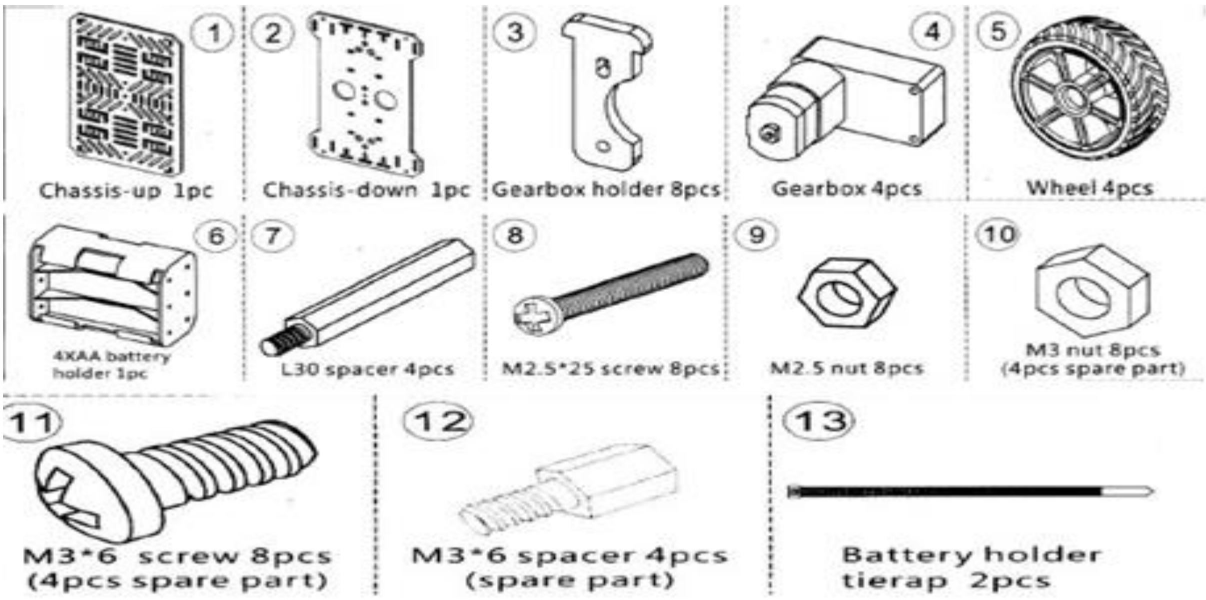


Fig 5.1: Mechanical Parts In Chassis

1. Rechargeable battery (we used a 4 AA battery energy cell 2500mAh 4.8v).
2. Chargeable battery Energizer 5000mAh 9v for the motor driver kit.
3. Battery holder for Energizer 9v.
4. Nerousky Mind wave.
5. Line-detecting sensor.
6. Connection wire for the line-detecting sensor.
7. Small solderless breadboard.
8. Perf Board.
9. Arduino Uno.
10. Motor driver 2 channel (we used L293D).
11. Bluetooth shield for Arduino (HC-05).
12. Ball Caster

13. Jumper Wires.
14. Battery holder tie rap.

Gearbox dc motor: Dc motor motors will be moving the wheels of our robot. The most usual type of Dc motors have limited rotation. They are used when you need to turn the motor to a specific angle. In our robot, we only need to control speed and direction. And, of course, the motor needs to be able to turn freely. Continuous rotation Dc motors are made for this. Almost any Dc motor can be modified to continuous rotation, but it’s easier to buy a ready-made version. The Parallax (Futaba) continuous rotation Dc motor is perfect for our needs has an external potentiometer adjustment screw, which allows identical. It centering of two Dc motors effortlessly. You’ll notice how handy this is later when we program the movements for the robot.

Motor Driver:

Driver section consists motor driver and two DC motors. Motor driver is used for driving motors because arduino does not supply enough voltage and current to motor. So we add a motor driver circuit to get enough voltage and current for motor. Arduino sends commands to this motor driver and then it drive motors.

5.2 BUILDING THE ROBOT

Now we will implement the main body of the robot the chassis it's easy to implement it just follow the instruction below and read it carefully.

Step 1: Assemble the chassis of the bot.

Fig 5.2: Picture Of Assembled Chassis

Step 2: Connect the motors to the chassis.

Step 3: Attaching Arduino

Before attaching the Arduino to the robot, cover the bottom of the Arduino with a tie rap to attach it with upper chassis but be carefully from the short circuits that could happen if the Arduino touched metal parts of the boot. We put the arduino in the center of the upper chassis of the robot.

Fig 5.3: Picture Of Arduino Uno

Step 4: Connect the motors to Motor Driver such as L293D.

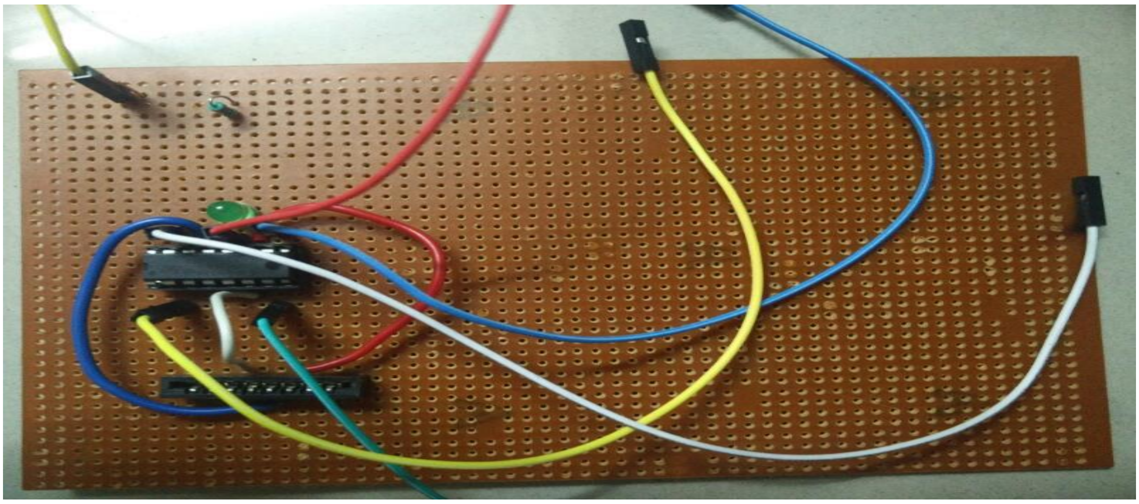


Fig 5.4: Picture Of L293d On Perf Board

Step 5: Connect Bluetooth module to Arduino as shown in the previous chapter.

Fig 5.5: Picture Of HC-05

Step 6: Make all the connections as shown in the below figure and for less confusion solder the wires on a perf board.

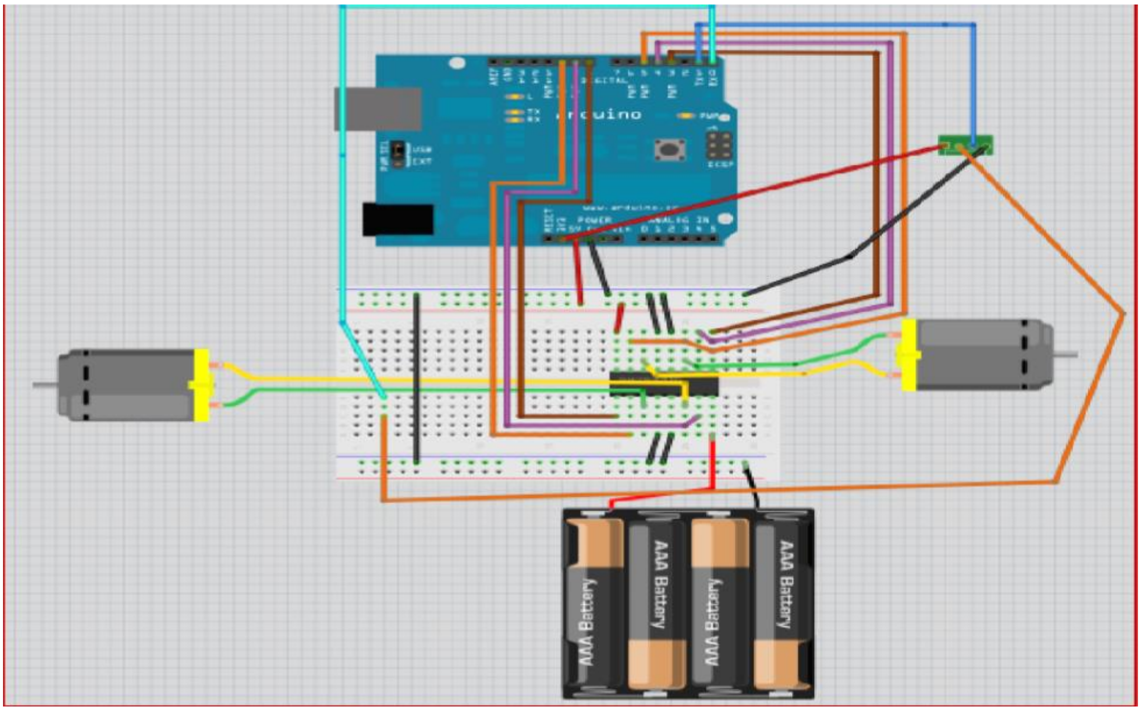


Fig 5.6: Circuit Diagram Of Entire Setup

Step 7: Load the Arduino Program into Arduino.

Fig 5.7: Code i.e., Uploaded To Arduino

Step 8: Switch ON the MindWave and it will directly pair to the Bluetooth Module.

5.3 WORKING OF ROBOT

Mind-Controlled robot is constructed using the Mindwave EEG headset, Arduino UNO, Bluetooth Module (HC-05) and a motor driver (L293D). The brain activity of the user is read using the EEG headset. The motor controller was used for controlling the direction and the speed of the servo motors used for the robot. The brain activity recorded by the Mindwave headset was transmitted to the Arduino Uno using a Bluetooth module.

The Attention value transmitted by the Mindwave headset is used to control the robot. The Mindwave headset transmitted the attention value as a stream of bytes. The format of each packet was <Number of bytes sent> <Type> <Value> <checksum>, where the Number of bytes sent signifies the total number of bytes sent in the packet, Type denotes the type of value transmitted (attention, meditation etc.,) by the headset, Value is the actual value and the checksum is used for validating if the packet is read in full. The Arduino is programmed to extract the attention value from the bytes transmitted. If the attention value read by the headset is above a particular threshold, the robot moves forward. The Speed of the servo motors is controlled based on the attention value returned by the Mindwave headset. So if the user is fully focusing, thus having an attention value of 1, the speed of the motor will be at its maximum.

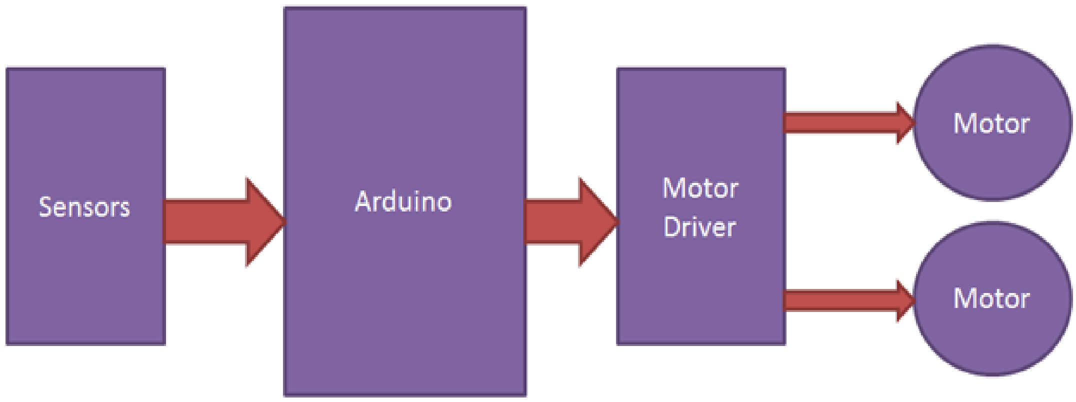


Fig 5.8: Operational Block Diagram For Robot

5.4 CODE

```
#define BAUDRATE 57600

#define DEBUGOUTPUT 0

#define LM1  3

#define LM2  4

#define RM1  7

#define RM2  8

#define powercontrol 10

// checksum variables

byte generatedChecksum = 0;

byte checksum = 0;

int payloadLength = 0;

byte payloadData[64] = {0};

byte poorQuality = 0;

byte attention = 0;

byte meditation = 0;

// system variables

long lastReceivedPacket = 0;

boolean bigPacket = false;


// Microprocessor Setup //

void setup()

{

        pinMode(LM1, OUTPUT);

         pinMode(RM1, OUTPUT);
```

```
        pinMode(LM2, OUTPUT);
        pinMode(RM2, OUTPUT);
        Serial.begin(BAUDRATE);
}
```

// Read data from Serial UART //

```
byte ReadOneByte()
{
        int ByteRead;
        while(!Serial.available());
        ByteRead = Serial.read();

       #if DEBUGOUTPUT
        Serial.print((char)ByteRead);
       #endif
        return ByteRead;
}
```

//MAIN LOOP//

```
void loop()
{
       if(ReadOneByte() == 170) {
       if(ReadOneByte() == 170) {
       payloadLength = ReadOneByte();
       if(payloadLength > 169)
         return;
```

```
 generatedChecksum = 0;
 for(int i = 0; i < payloadLength; i++)
 {
         payloadData[i] = ReadOneByte();
         generatedChecksum += payloadData[i];
}
 checksum = ReadOneByte();
 generatedChecksum = 255 - generatedChecksum;
 if(checksum == generatedChecksum)
 {
   poorQuality = 200;
   attention = 0;
   meditation = 0;
   for(int i = 0; i < payloadLength; i++)
  {
            switch (payloadData[i]) {
            case 2:
            i++;
            poorQuality = payloadData[i];
            bigPacket = true;
            break;
            case 4:
            i++;
            attention = payloadData[i];
            break;
```

```
                case 5:
                i++;
                meditation = payloadData[i];
                            break;
                case 0x80:
                i = i + 3;
                break;
                 case 0x83:
                i = i + 25;
                 break;
                default:
                 break;
                }
            }
#if !DEBUGOUTPUT
      if(bigPacket)
         {
               Serial.print("PoorQuality: ");
               Serial.print(poorQuality, DEC);
               Serial.print(" Attention: ");
               Serial.print(attention, DEC);
               Serial.print(" Meditation: ");
               Serial.print(meditation, DEC);
               Serial.print(" Time since last packet: ");
               Serial.print(millis() - lastReceivedPacket, DEC);
```

```
lastReceivedPacket = millis();
Serial.print("\n");
switch(attention / 20)
    {
        case 0:
            digitalWrite(LM1,LOW);
            digitalWrite(LM2,HIGH);
            digitalWrite(RM1,LOW);
            digitalWrite(RM2,HIGH);
            break;
        case 1:
            digitalWrite(LM1,LOW);
            digitalWrite(RM1,LOW);
            digitalWrite(LM2,LOW);
            digitalWrite(RM2,LOW);
            break;
        case 2:
            digitalWrite(LM1,LOW);
            digitalWrite(LM2,LOW);
            digitalWrite(RM1,LOW);
            digitalWrite(RM2,LOW);
            break;
        case 3:
            digitalWrite(LM1,HIGH);
            digitalWrite(LM2,LOW);
```

```
                    digitalWrite(RM1,HIGH);
                    digitalWrite(RM2,LOW);
                    break;
                case 4:
                    digitalWrite(LM1,HIGH);
                    digitalWrite(LM2,LOW);
                    digitalWrite(RM1,HIGH);
                    digitalWrite(RM2,LOW);
                    break;
                 case 5:
                    digitalWrite(LM1,HIGH);
                    digitalWrite(LM2,LOW);
                    digitalWrite(RM1,HIGH);
                    digitalWrite(RM2,LOW);
                    break;

             }
         }
        #endif
             bigPacket = false;
      }
      }
  }
}
```

CONCLUSION

In the educational uses:

•In this project we can use voice recognition to send the function of movement by speech and use Neruosky to control the speed.

•Use the Emotive EBook to control the directions of the robot by using malty sensors to make all the directions and the speed by the brain signals.

• Use the brain computer interface to control the robotic Arm by Neruosky or Emotive EBook.

In the entertainment uses:

•Make an intelligence video games

•By using malty sensors you can make a brain browser.

BIBLIOGRAPHY

Books:

- M. Benzi, "Getting started with Arduino", O'Reilly, 2009.
- T.Karvinen,K.Karvinen, "Make a Mind-Controlled Arduino Robot", O'Reilly, 2012
- M.Kurz, W.Almer, F.Landolt, "Brain Computer Interface" 2006.
- "Arduino basic C review", Nuno Alves college of Engineering,Western New England University.
- K.Crowley, A.Sliney, I.Pitt, D.Murphy," Evaluating a Brain-Computer Interface to Categorise Human Emotional Response", IEEE, 2010
- Kristín Guðmundsdóttir,"Improving Players' Control Over The NeuroSky Brain Computer-Interface",School of Computer Science,2011

Websites:

- www.arduino.cc
- http://www.neurosky.com/
- http://www.oreilly.com.
- http://my.safaribooksonline.com./
- http://en.wikipedia.org/wiki/Brain%E2%80%93computer_interface#EEG
- http://www.seeedstudio.com/wiki/Grove_-_Line_Finder
- http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth
- http://www.electrical-neuroimaging.ch/publications/grave_esann2009.pdf