# INTRODUCTION

# 1. INTRODUCTION

Paralysis is usually caused by problems with the spinal cord that the brain uses to pass control messages to muscles. Therefore, patients who are suffering from this severe problem need a great deal of support to enhance their ability to carry out daily activities. As a result, this problem has an impact on a person's quality of life and adds a high cost for the residential care packages since another person is needed to serve patient and satisfy his needs. In reviewing the literature, researchers are focusing on finding a technology that could be used by individuals who are affected by this problem. The goal was to develop devices that could be steered by the electrical activity of the brain using external electrodes attached to the user's scalp. Brain computer interface technology was envisioned as a promising and useful strategy that could give patients who are severely physically disabled new abilities to interact with the world around them through their mental activity.Brain Computer Interface (BCI) is a process by which Humans can control the external devices using their brain waves.These interfaces are made using sensors which can record brain data either by invasive plantation or noninvasive plantation. Our Human Brain is highly complex and is made up of 100 billion neurons. There are many types of neurons in our brain such as motor neurons, sensory neurons. These neurons get fired up while generating a response for a particular stimuli and generate an electrical signal which is detected by the electrodes and can be used to control a number of devices. Home Automation is an area where BCI can be used and our entire house can be controlled simply by our brain. This technology would prove as a great boon for almost all people on the planet. Less energy would be wasted for performing menial tasks such as switching on the lights, ac's and other electrical appliances.This technology is breathtaking and has the potential to completely revolutionize and change our lives.

## 1.1 BACKGROUND

Nowadays, due to the impact of the inventions and innovations, various methods are adopted for the controlling of home appliances. Particularly, they are designed for the elderly people and also for the ill. In some cases many advanced systems are constructed for the physically disabled people. The main problem in all these systems is that they are not suitable for all kinds of people and under all conditions. Each system needs a specific design and device, which increases the cost of production and also less stability. Main objective of our project is to make a common system, which is used by all kind of people at all the situations for controlling devices around us.

Brain waves are one of the common resources of humans, which can be utilized effectively. In our project, the brain wave plays a vital role in controlling devices in the home environment. The brain waves are traced by means of Neurosky Mind wave mobile and are transferred using Bluetooth to the central automation system. The central system is made using Arduino and is programmed to control the all the appliances connected. The quality of the project is that, it can be used by all the people and can control all the devices. This also has a main application in the industrial automation system. The robotic arm and pulley rails are controlled using brain waves. This method increases the ease of use.

# LITERATURE SURVEY

# 2. LITERATURE SURVEY

## 2.1. EXISTING SYSTEM

Currently an external help is needed to assist the paralyzed patients. As their mental and physical strength is weaken, that result to lack of happiness. In a home with all sort of automation application, there will be too many remote control or monitoring terminal. Current smart home devices are usually a customized hybrid of one or more of these application for broader application.

# PROBLEM IDENTIFICATION & OBJECTIVES OF THE PROJECT

# 3. PROBLEM IDENTIFICATION & OBJECTIVES OF THE PROJECT

## 3.1. PROBLEM STATEMENT

In the present work we use the Brain Computer Interface technology to allow to the dependent persons the utilization of the basic elements of their house, such as turning on and turning off lamps, rolling up and down a roller shutter or switching on the heating system. For doing this it is necessary to automate these devices and to centralize its managing in a platform, which constitutes a demotics system

## 3.2. OBJECTIVES OF THE SYSTEM

Our project aims on controlling of hardware appliances using the electronics relay based switching circuit. The BCI is a direct communication pathway between the human brain and an external device, which help us to remove remote controlled home automation system. A Neurosky brainwave sensor is used to analyze the EEG signals.

- Disabled patient can control the home appliances by their own
- Remote controlled system is converted into brain computer interface
- Disable patient can alert or call anyone by using brain wave if there is any emergency

## 3.3. SYSTEM STUDY

Many environmental control systems were proposed and applied for people with disability to control their surroundings. Radio frequency identification and voice recognition are some of these systems. Those systems work well for people with motion disability while they will not work for people with voice or vision impairment. Other systems using human's physiological state were proposed. The author in proposed a BCI system to help disabled people to input phone numbers. The system is based on the steady-state visual evoked potential where twelve buttons are illuminated in front of the user at different rates. To this end, disabled people could input a phone number by gazing at those buttons.

Interaction between user's brain and computer was achieved through a number of ways: Visual Evoked Potentials (VEP), Slow Cortical Potentials, P300 potentials, N400 potentials, and Sensory Motor Rhythm (SMR). To this end, VEP refers to the electrical potential recorded from the visual cortex in response to stimulation of light; P300 is an event related potential (ERP, recorded in response to the occurrence of a discrete event, especially when the subject is actively engaged in the task of detecting the targets). This signal appears approximately 300ms after some infrequent stimuli and typically measured by the electrodes covering the parietal lobe.

Several techniques were used in the previous methods to extract and classify features form brain signals. Wavelet-based feature extraction algorithms were introduced in. Artificial Neural Network (ANN) has been used by for cortical control of arm prosthetics. Moreover, Power Spectral Density (PSD), Band Powers (BP), Adaptive Auto Regressive (AAR), were also used for feature extractions. A great variety of classification algorithms was also used to design BCI system.

### 3.3.1 TECHNICAL FEASIBILITY

For developing desktop application, we use software such as java and embedded C. The user needs to have knowledge of using computer. We are using Arduino for home automation and EEG headset for brain computer interface.

### 3.3.2 ECONOMIC FEASIBILITY

For developing desktop application, we use software such as java and embedded C. The user needs to have knowledge of using computer. We are using Arduino for home automation and EEG headset for brain computer interface.

### 3.3.3 SOCIAL FEASIBILITY

This project creates a great impact on society, since it can be used by all the people and also for various purposes. Home automation focuses on making it possible for older adults and people with disabilities to remain at home, safe and comfortable. Home automation is becoming a viable option for older adults and people with disabilities who would prefer to stay in comfort of their home rather than move to a healthcare facility. The transition to a healthcare facility can cause a lot of anxiety and home automation can either prevent or delay this anxiety. For the disabled, a smart home gives them opportunity for independence, which will help them to get confidence and determination. Smart home system will make it possible for family members to monitor their loved ones. Hence the impact on the Market is also high.

# PROBLEM ANALYSIS & DESIGN

# 4. PROBLEM ANALYSIS & DESIGN

## 4.1 MODULE DESCRIPTION

The system consists of three modules

- Headset Module
- Brain-Computer interface module
- Computer-Arduino module

### 4.1.1 HEADSET MODULE

EEG headset act as the input device. It maps the brain waves and provides us with the concentration values. These values are used for automation of house. Apart from concentration value it is also provide the eye blink value, meditation value etc. This is worn by the paralyzed patient by wrapping it around head.

### 4.1.2 BRAIN – COMPUTER INTERFACE MODULE

This is a desktop application developed in python. It acts as user interface. It contains a menu listing the devices to be controlled and the mouse pointer will keep on moving groom one devices to another. When the mouse pointer is over the required devices to be controlled, the user has to concentrate and when concentration exceeds a particular threshold, the required operation will be triggered by the program loaded in the arduino.

### 4.1.3 COMPUTER - ARDUINO MODULE

Arduino contains the program and it will trigger the operation depending on serial input from the computer. Relays are used for controlling the electronic devices.

## 4.2 BLOCK DIAGRAM

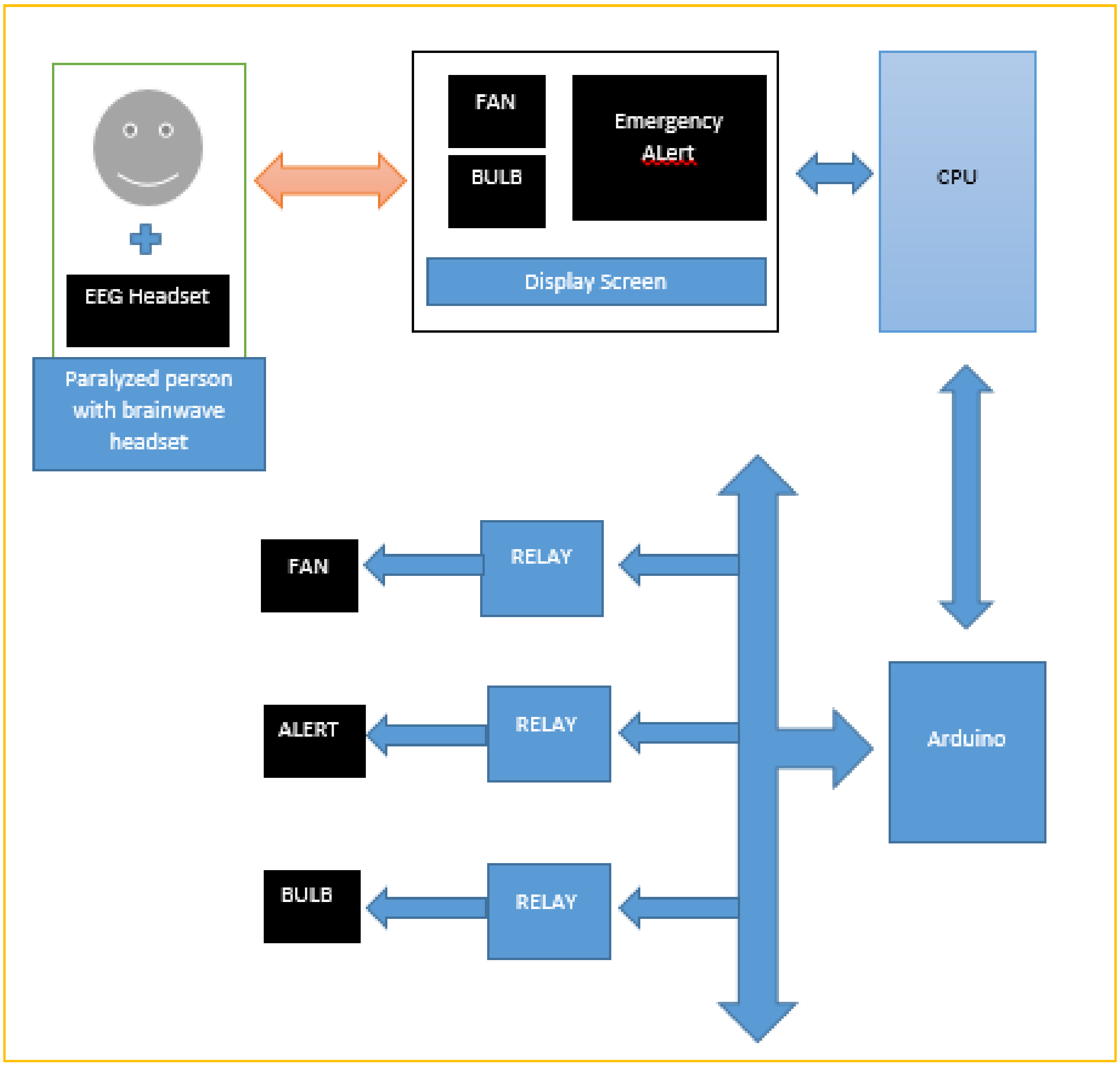


**Figure 4.1 Basic Block Diagram**

### 4.2.1 HEADSET MODULE

In this module an EEG headset is used for connect between brain waves and computer system

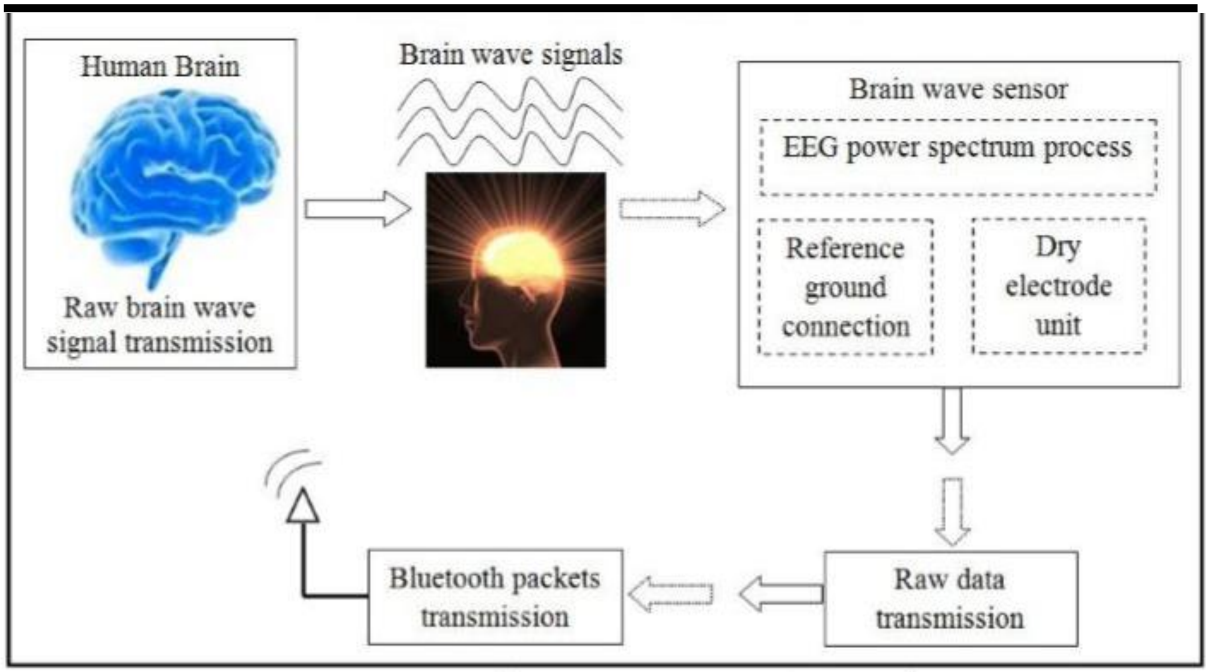


**Figure 4.2 : Headset Module**

#### 4.2.1.1 ELECTRODES

We will be using dry non-invasive scalp sensors to measure the electrical activity of our brain. Specific electrodes would be placed at the motor cortex area and the rest would be placed in many specific areas of the brain to measure brain activity.

#### 4.2.1.2 AMPLITUDE AND FILTER CIRCUIT

These signals along the scalp are very weak i.e., in micro volts and contain noise. Thus, we need to amplify and filter out the frequencies not useful to us, we can use an instrumentation amplifier to amplify the signals we get from electrodes.

One such instrumentation amplifier is AD620, in which we can vary the gain by varying the value or resistor. A High pass filter is designed having cut off frequency 7 Hz and a low pass filter is designed which will have a cut-off frequency at 31 Hz.

### 4.2.1.3 WIRELESS-TRANSMITTER

The wireless transmitter will transmit the signal which will be received by the receiver end. A Zig-bee Wireless transmitter is used to transmit the signals.

### 4.2.2 BRAIN-COMPUTER INTERFACE MODULE

Hence it will allow only frequencies from 8-31 Hz to pass. The basic noise in all the systems is at 60HZ which is due to power line interference. For, suppression or attenuation of this frequency we can design a notch filter, that will severe reduce gain of this frequency. Also an Analog to Digital converter will be required to feed the data to microcontroller. In this module brain computer interface is used to connect between home automation and brain sensor.

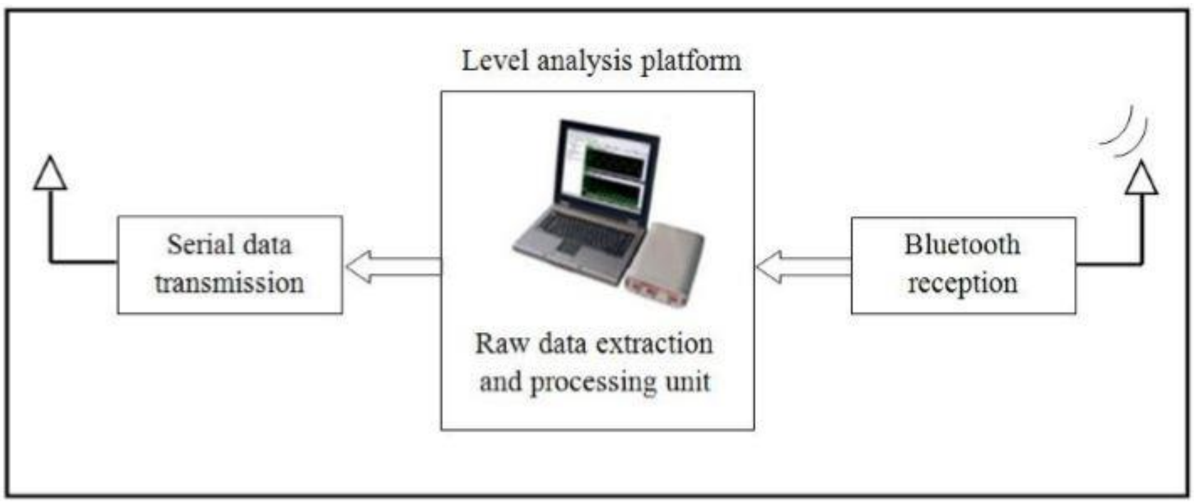


**Figure 4.3 Brain-Computer Interface Module**

### 4.2.2.1 USER-INTERFACE

The user interface will act as an interface which would let the user know which device is selected. It will basically let the user choose which device it wants to use. Apart from this the user interface will also be having many other details such as concentration level. The user interface will be completely customizable by the user. The user can add and name the devices which he wants to control. The user interface can be designed as a mobile application to make it more users friendly.

### 4.2.3 COMPUTER- ARDUINO MODULE

This module will deal with the controlling of hardware appliances using the electronics relay based switching circuits. Actual home appliances are connected to this circuit and the circuit will be then connected to the computer.

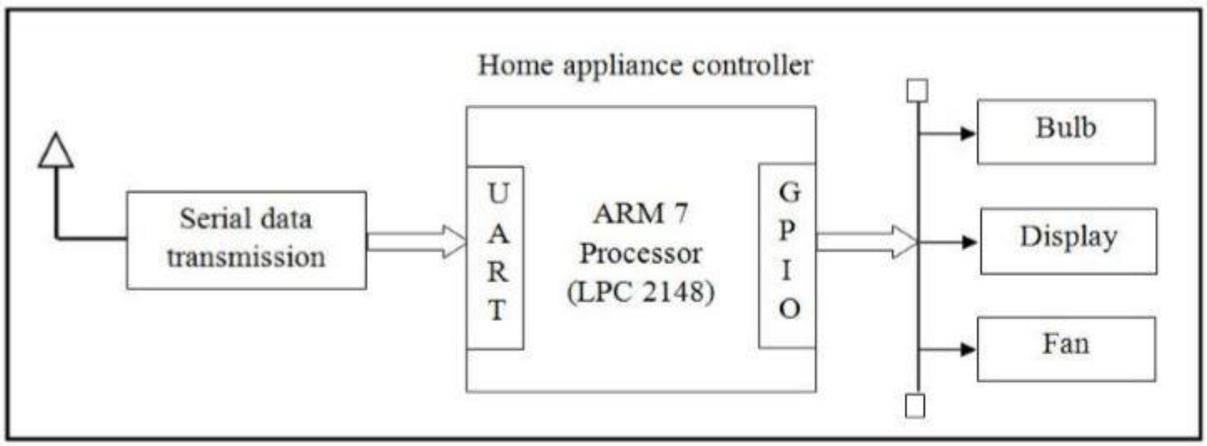


**Figure 4.4 Computer- Arduino Module**

#### 4.2.3.1 MICRO-CONTROLLER

After the amplification and the filtering process the microcontroller will process the signals and give the output according to the specific algorithm. The micro-controller will do the following operations:

1. Will take the digital signal data from the ADC and process it according to the working specified.
2. The microcontroller will be directly connected to the user interface which will display the data selected, and details such as concentration level.
3. The microcontroller will send the processed data to the wireless transmitter which will transmit it wirelessly.

#### 4.2.3.2 WIRELESS RECEIVER

The Receiver will receive the signals and feed it to the O/P Devices. The output devices can be anything lights, TV, fan etc.

### 4.3 FLOW CHART

After Switching on the Brainwave headset and the processor will initialize and the headset will starts sensing the neurons signals and after sensing the signals it will transfer them to through the Bluetooth to the system it enter into the Mat lab to check the attention and eye Blinking Levels.

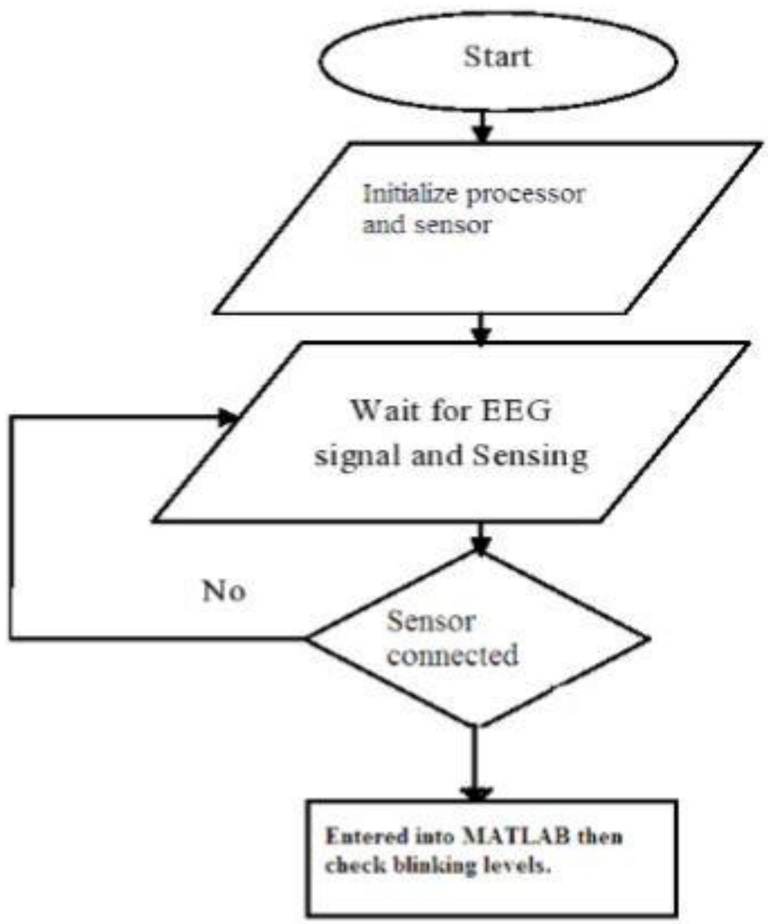


**Figure 4.5 Flow chart of Design flow**

The working of the system can be illustrated by following

The electrical signals that helps to communicate 2 neurons and that which forms the basis of whatever we do is never fully transferred from one neuron to another ,but some part of it escapes and reaches the scalp. This electrical neural signal can be observed and captured using right equipment's such as EEG electrodes. The electrodes will then send the signals to the amplifier and filter circuit wherein the signal is amplified and unwanted noise and signals are filtered out .The analogue signals are then converted into digital signals using an ADC converter. The microcontroller process the signals based on the following logic.

1. When the user imagines his right hand the left part of the motor cortex in the brain gets and the signals are detected .Over the motor cortex beta waves are associated with the muscle contractions that happen in isotonic movements and are suppressed prior to and during movement changes .So now the cursor will move in the right hand side of the user interface. The same can be done with the left hand, but the only difference will be is that when the user imagines his left hand the cursor will move to the left hand side.

2. Now if the user wants to select a particular device he will have to concentrate his mind. When a user concentrates his mind Beta waves (16-31Hz) are generated, the brain emits more than 18 Hz frequency and the frequency will get locked and that particular device will get selected. It must be noted that when the user moves the cursor to the left or to the right there will be a particular delay after every move by the cursor which the user can set. When a person relaxes alpha (8 – 15 Hz) waves are emitted. The above logic can also be implemented using machine learning, which will greatly reduce the error of the system.

# IMPLEMENTATION AND PRESENTATION OF DATA

# 5. IMPLEMENTATION AND PRESENTATION OF DATA

## 5.1. HARDWARE REQUIREMENTS

- EEG headset
- Arduino
- Bluetooth For Arduino
- Relay

### 5.1.1 ARDUINO

An Arduino (figure 5.1) board historically consists of an Atmel 8-, 16- or 32-bit AVR microcontroller (although since 2015 other makers' microcontrollers have been used) with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which let users connect the CPU board to a variety of interchangeable add-on modules termed shields.

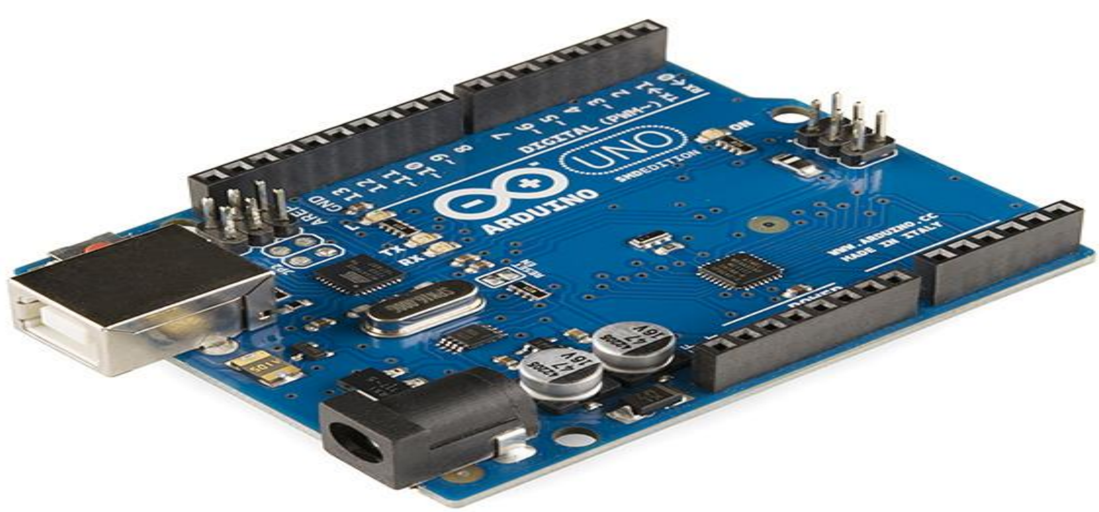


**Figure 5.1 Arduino**

Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an $I^2C$ serial bus—so many shields can be stacked and used in parallel. Before 2015, Official Arduinos had used the Atmel megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. In 2015, units by other producers were added. A handful of other processors have also been used by Arduino compatible devices. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external chip programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer. Currently, optiboot bootloader is the default bootloader installed on Arduino UNO.

At a conceptual level, when using the Arduino integrated development environment, all boards are programmed over a serial connection. Its implementation varies with the hardware version. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods, when used with traditional microcontroller tools instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

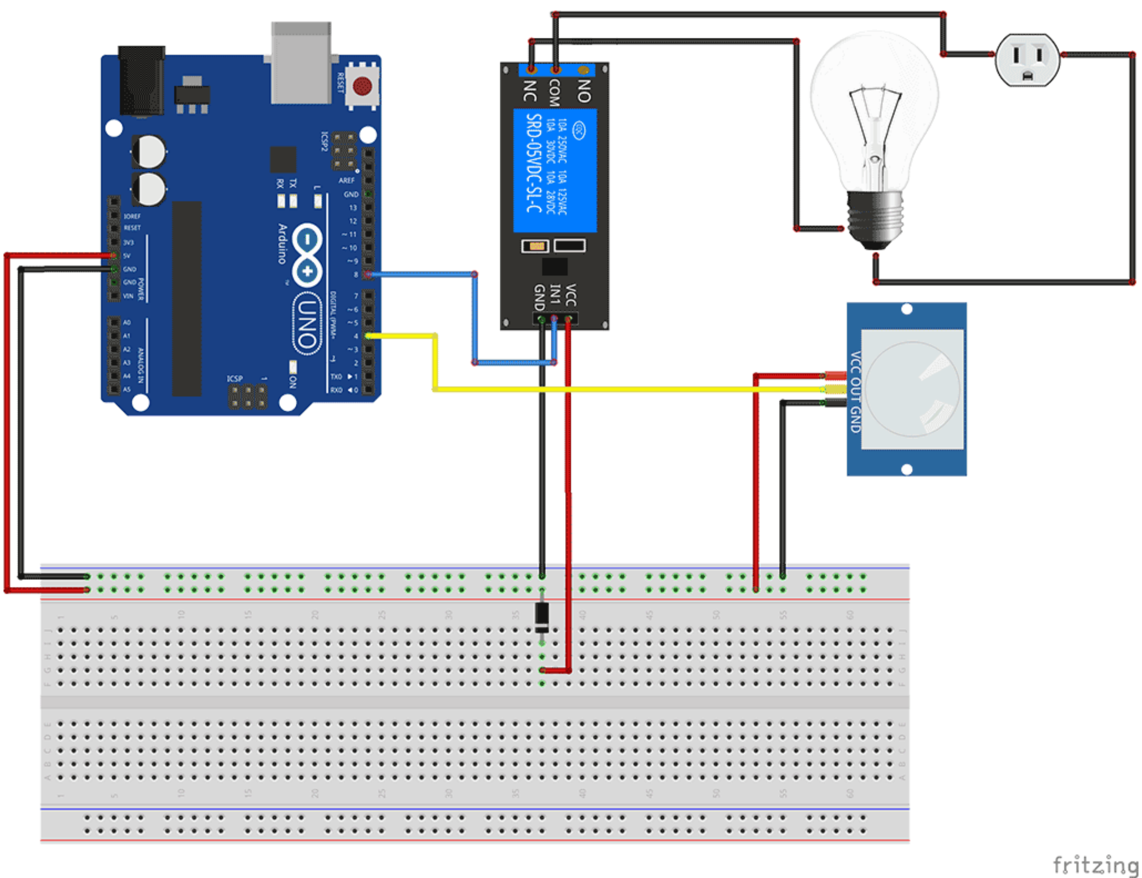


**Figure 5.2 Arduino Connection Diagram**

### 5.1.2 EEG HEADSET

**EEG Signals**

EEG signals can be collected with electrodes that are placed on the surface of the scalp. The most widely used electrodes are silver/silver chloride (Ag/AgCl) because they have low cost, low contact impedance, and relatively good stability. Furthermore, there are rather mature commercialized acquisition systems including the amplifier and EEG cap with integrated Ag/AgCl electrodes, which have been successfully applied in scientific research and clinical diagnosis. However, using Ag/AgCl electrodes requires removing outer skin layer and filling gel between electrodes and scalp (and thus, this kind of electrodes is also called “wet” electrodes). These operations take long time and are uncomfortable to users. To address these limitations of “wet” electrodes, some researchers have been exploring “dry” electrodes, which do not need to use gel and skin cleaning. The main disadvantage of existing dry electrodes is that the acquired EEG signals are worse than those acquired with conventional electrodes due to the increase of contact impedance. Some companies (such as Quasar, Emotiv Systems Inc., and NeuroSky Inc.) have been commercializing acquisition systems based on dry electrodes. Here we are using NeuroSky Brainwave headset. However, they are not yet mature, and some researchers have doubts about what physiological signals these systems actually acquire.

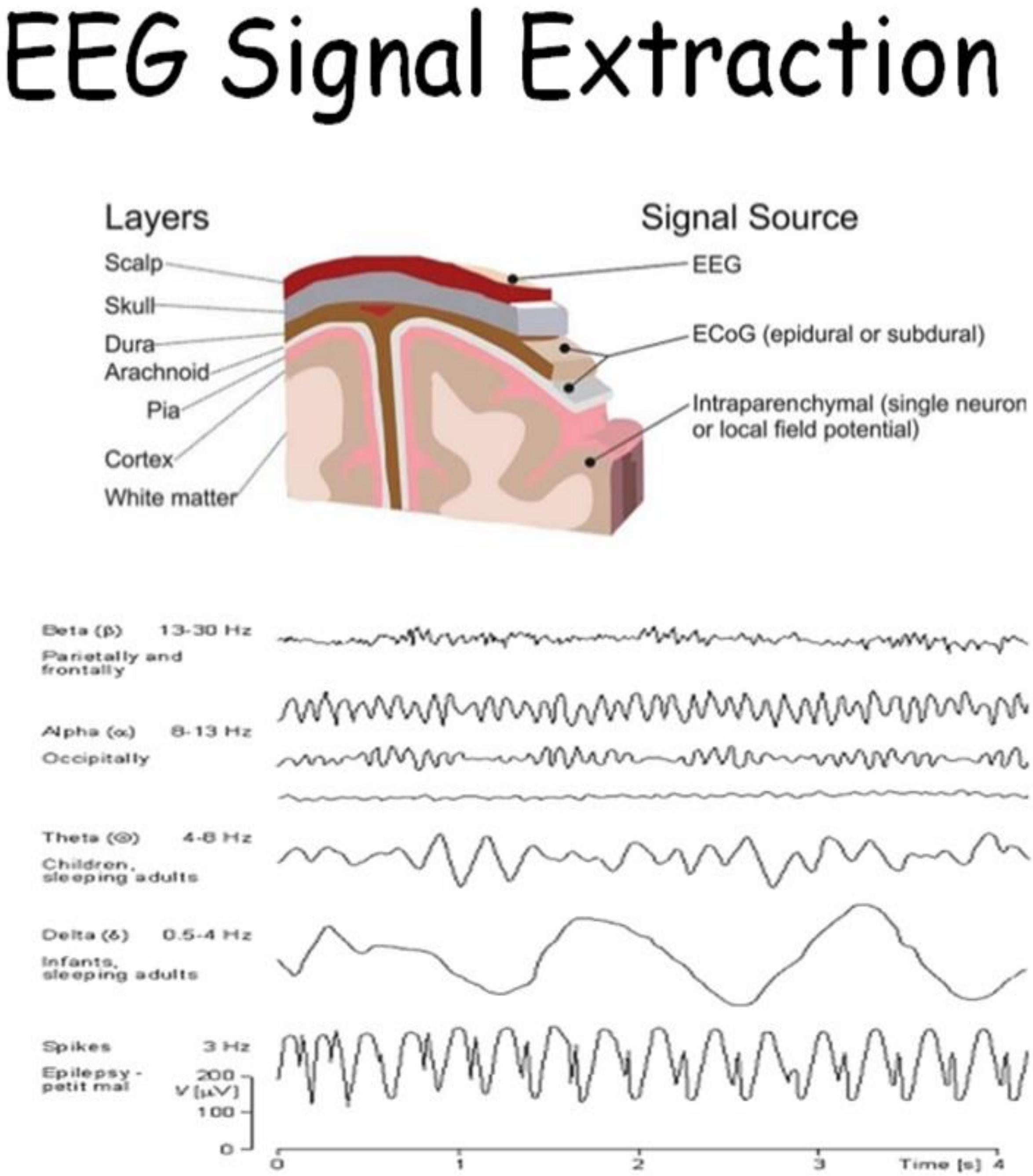


**Figure 5.3 EEG Signal Extarction**


**Figure 5.4: Brainwave Headset provided by NeuroSky**

The signals from our brain are taken by using the brain sensor shown in figure 5.4. EEG Sensor to Sense the Human brain, and it will be sensed by using the Brainwave Headset which is provided by NeuroSky i.e Dry electrode [5]. Technologies and those signals will be transferred by using Bluetooth which is there in the Brainwave headset, for this Brainwave headset we need to give power using a AAA battery .The Brainwave headset comes with Power switch, a sensor tip, flexible ear arm and a ground connection Ear clip. In this Headset we use Non-invasive sensor that won't cause any pain to the User who were the headset. After inserting 3 AAA batteries switch on the Brainwave headset automatically the LED indicator will blink in GREEN colour which shows the sensor is on.

**Brainwaves**

The last century of neuroscience research has greatly increased our knowledge about the brain and particularly, the electrical signals emitted by neurons firing in the brain. The patterns and frequencies of these electrical signals can be measured by placing a sensor on the scalp. The Mind Tools line of headset products contain Neurosky Think Gear technology, which quantify the analog electrical signals, commonly referred to as brainwaves, and exercise them into digital signals. The Think Gear technology then makes those computations and signals available to games and applications.

**eSense**

eSense is a NeuroSky's proprietary algorithm for representing mental states. To calculate eSense, the NeuroSky Think Gear technology intensifies the raw brainwave signal and removes the ambient noise and muscle movement. The eSense algorithm is then applied to the remaining signal, resulting in explicated eSense meter values. Please note that eSense meter values do not interpret an exact number, but instead describe ranges of activity. The eSense meters are a way to show how effectively the user is captivating Attention (similar to concentration) or Meditation (similar to relaxation).

**(a)Attention eSense:**

The eSense Attention meter shows 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.

**(b)Meditation eSense:**

The eSense Meditation meter shows 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 instantly result in an intensified effect 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 reduce activity by the active mental processes in the brain. It has long been an observed that closing one's eyes turns of 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.

**eSense Meter - Technical Description**

For each different type of eSense (i.e. Attention, Meditation), the meter value is reported on a relative eSense scale of 1 to 100. On this scale, a value between 40 to 60 at any given moment in time is considered “neutral” and is similar in notion to “baselines” that are established in conventional brainwave measurement techniques (though the method for determining a Think Gear baseline is proprietary and may differ from conventional brainwaves)

### 5.1.3 RELAY

We know that most of the high end industrial application devices have relays for their effective working. Relays are simple switches which are operated both electrically and mechanically. Relays consist of a n electromagnet and also a set of contacts. The switching mechanism is carried out with the help of the electromagnet. There are also other operating principles for its working. But they differ according to their applications. Most of the devices have the application of relays.

The main operation of a relay comes in places where only a low-power signal can be used to control a circuit. It is also used in places where only one signal can be used to control a lot of circuits. The application of relays started during the invention of telephones. They played an important role in switching calls in telephone exchanges. They were also used in long distance telegraphy. They were used to switch the signal coming from one source to another destination. After the invention of computers they were also used to perform Boolean and other logical operations. The high end applications of relays require high power to be driven by electric motors and so on. Such relays are called contactors

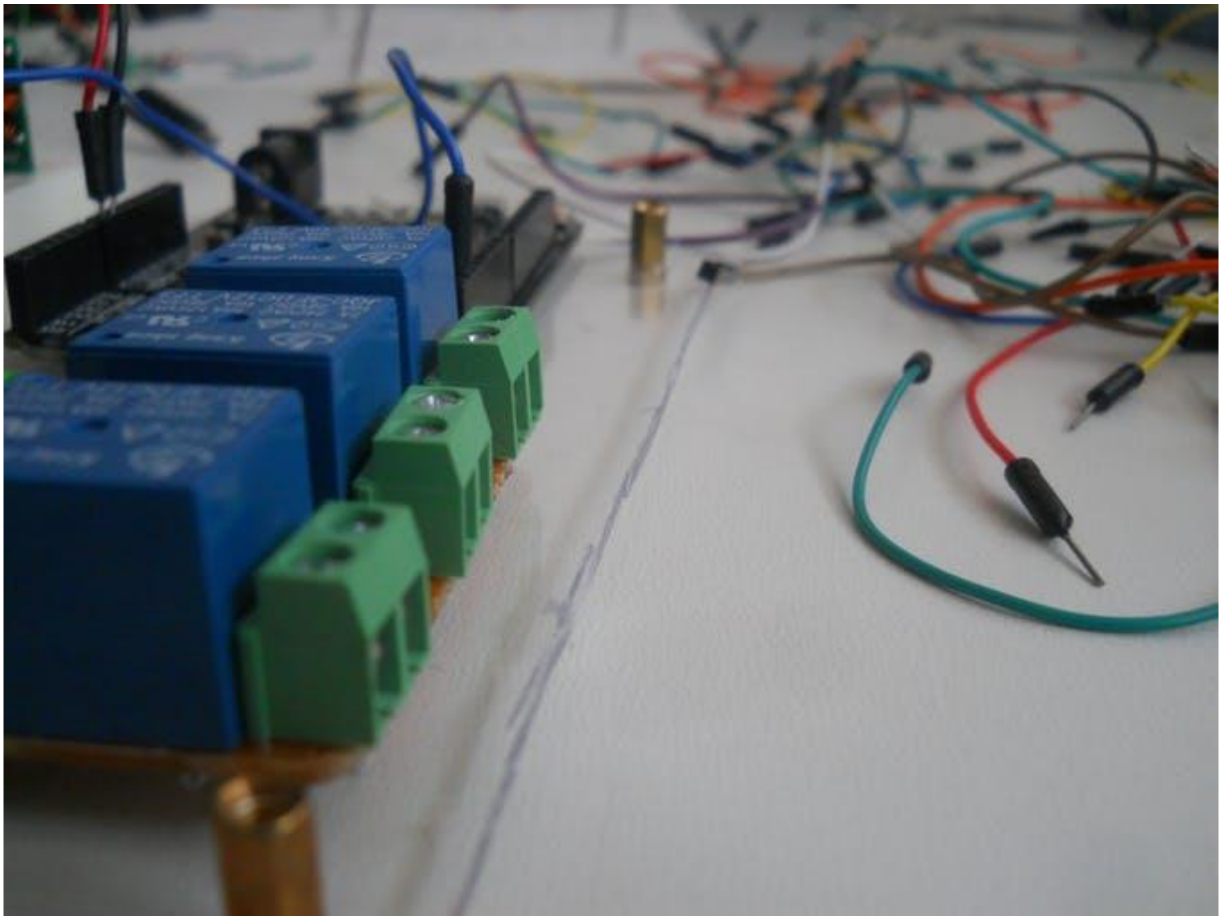

**Figure 5.5 Relay**

The basics for all the relays are the same. Take a look at a 4 – pin relay shown below. There are two colours shown. The green colour represents the control circuit and the red colour represents the load circuit. A small control coil is connected onto the control circuit. A switch is connected to the load. This switch is controlled by the coil in the control circuit. Now let us take the different steps that occour in a relay.

Relays have the exact working of a switch. So, the same concept is also applied. A relay is said to switch one or more poles. Each pole has contacts that can be thrown in mainly three ways. They are

- **Normally Open Contact (NO)** – NO contact is also called a make contact. It closes the circuit when the relay is activated. It disconnects the circuit when the relay is inactive.

- **Normally Closed Contact (NC)** – NC contact is also known as break contact. This is opposite to the NO contact. When the relay is activated, the circuit disconnects. When the relay is deactivated, the circuit connects.
- **Change-over (CO) / Double-throw (DT) Contacts** – This type of contacts are used to control two types of circuits. They are used to control a NO contact and also a NC contact with a common terminal. According to their type they are called by the names **break before make** and **make before break** contacts.
- Relays are also named with designations like

- **Single Pole Single Throw (SPST)** – This type of relay has a total of four terminals. Out of these two terminals can be connected or disconnected. The other two terminals are needed for the coil.
- **Single Pole Double Throw (SPDT)** – This type of a relay has a total of five terminals. Out f these two are the coil terminals. A common terminal is also included which connects to either of two others.
- **Double Pole Single Throw (DPST)** – This relay has a total of six terminals. These terminals are further divided into two pairs. Thus they can act as two SPST's which are actuated by a single coil. Out of the six terminals two of them are coil terminals.
- **Double Pole Double Throw (DPDT)** – This is the biggest of all. It has mainly eight relay terminals. Out of these two rows are designed to be change over terminals. They are designed to act as two SPDT relays which are actuated by a single coil.

You must note some factors while selecting a particular relay. They are

- Protection – Different protections like contact protection and coil protection must be noted. Contact protection helps in reducing arcing in circuits using inductors. Coil protection helps in reducing surge voltage produced during switching.

- Look for a standard relay with all regulatory approvals.
- Switching time – Ask for high speed switching relays if you want one.
- Ratings – There are current as well as voltage ratings. The current ratings vary from a few amperes to about 3000 amperes. In case of voltage ratings, they vary from 300 Volt AC to 600 Volt AC. There are also high voltage relays of about 15,000 Volts.
- Type of contact used – Whether it is a NC or NO or closed contact.
- Select Make before Break or Break before Make contacts wisely.
- Isolation between coil circuit and contacts

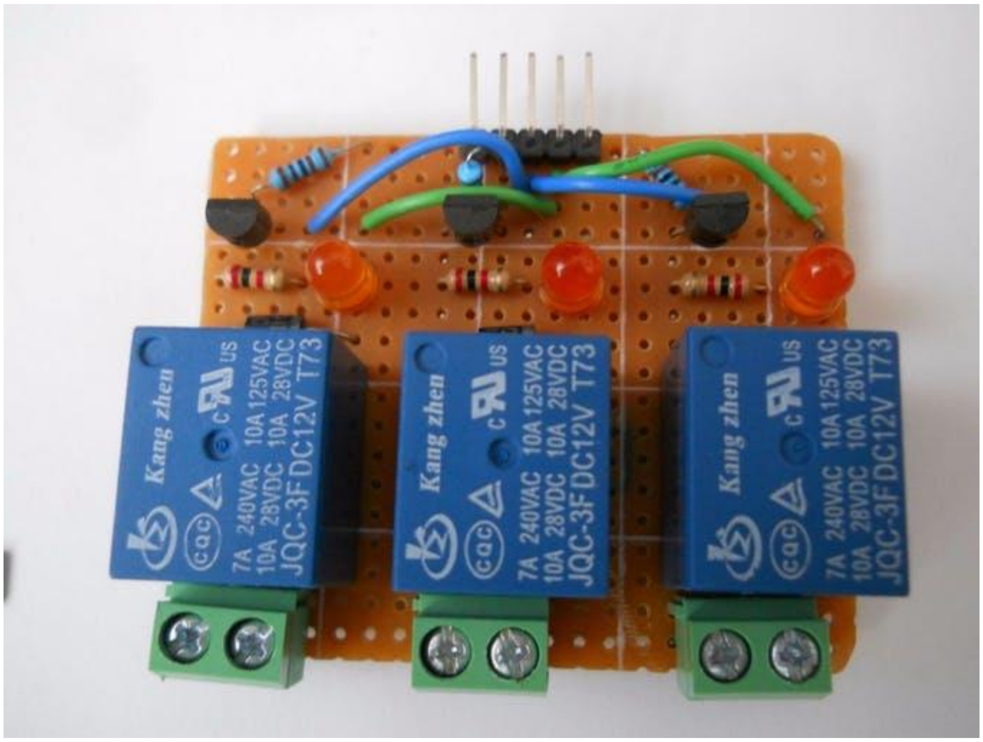


**Figure 5.6 3-channel relay**

### 5.1.4 ARDUINO BLUETOOTH

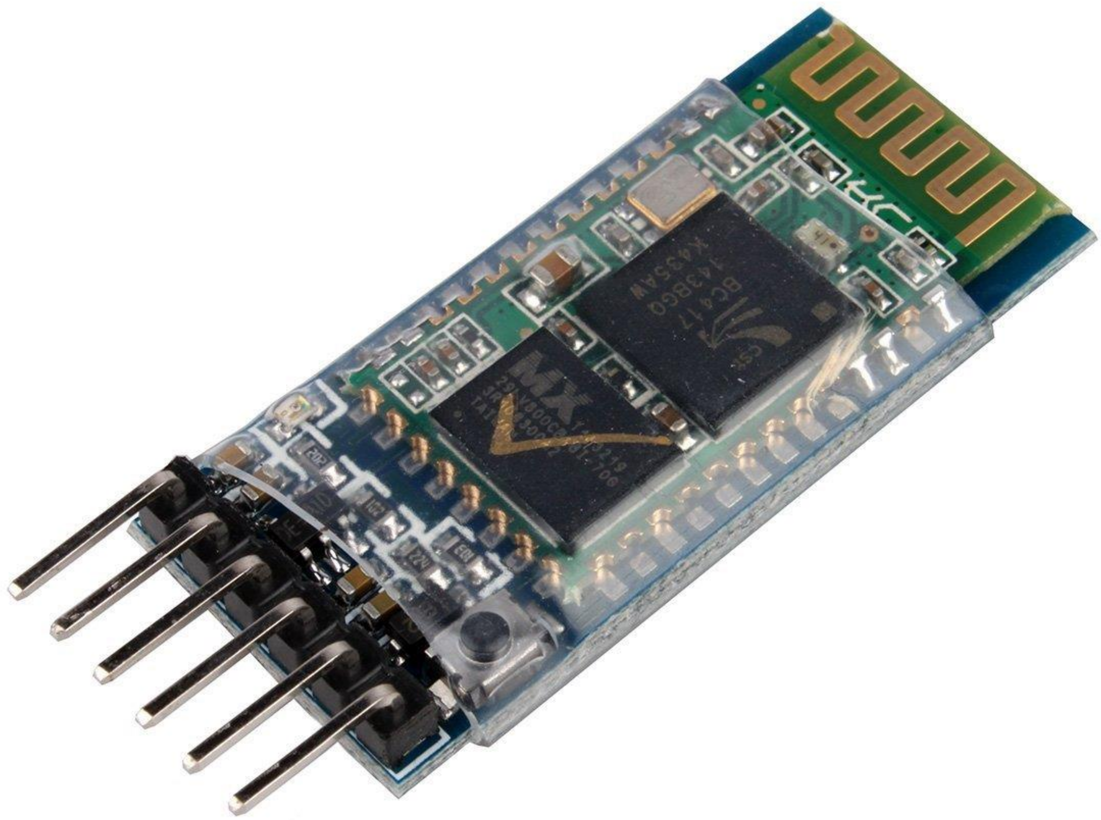

**Figure 5.7 Arduino Bluetooth**

The Arduino BT is a microcontroller board originally was based on the ATmega168, but now is supplied with the 328 (datasheet) and the Bluegiga WT11 bluetooth module (details and datasheet). It supports wireless serial communication over bluetooth (but is not compatible with Bluetooth headsets or other audio devices). It has 14 digital input/output pins (of which 6 can be used as PWM outputs and one can be used to reset the WT11 module), 6 analog inputs, a 16 MHz crystal oscillator, screw terminals for power, an ICSP header, and a reset button. It contains everything needed to support the microcontroller and can be programmed wirelessly over the Bluetooth connection.

The Arduino BT can be powered via the V+ and GND screw terminals. The board contains a DC-DC convector that allows it to be powered with as little as 2.5V, a maximum of 12V. Higher voltages or reversed polarity in the power supply can damage or destroy the board. The protection for reverse polarity connection is ONLY on the screw terminal.

The power pins are as follows:

- +VIN. The input voltage to the Arduino board (i.e. the same as the V+ screw terminal). You can supply voltage through this pin, or, if supplying voltage via the screw terminals, access it through this pin. Warning: The protection for reverse polarity connection is ONLY on the screw terminal, do not attach negative voltages to this pin. It will damage the board.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the screw terminal (2.5V - 12V) or the VIN pin of the board (2.5V-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- GND. Ground pins.

Bluetooth Communication

The Bluegiga WT11 module on the Arduino BT provides Bluetooth communication with computers, phones, and other Bluetooth devices. The WT11 communicates with the ATmega328 via serial (shared with the RX and TX pins on the board). It comes configured for 115200 baud communication. The module should be configurable and detectable by your operating system's bluetooth drivers, which should then provide a virtual com port for use by other applications. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board over this bluetooth connection. The board can also be reprogrammed using this same wireless connection.

## 5.2 SOFTWARE REQUIRMENTS

- Python Programing Language

### 5.2.1 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

## 5.3. TESTING

System testing is normally carried out in a planned manner according to the system test plan document. The system test plan identifies all testing-related activities that must be performed, specifies the schedule of testing, and allocates resources. It also lists all the test cases and the expected outputs for each test case. Here the modules are integrated in a planned manner.

### 5.3.1. FUNCTIONAL TESTING

Functional testing refers to tests that verify a specific action or function of the code. These are usually found in the code requirements documentation, although some development methodologies work from use cases or user stories. Functional tests tend to answer the question of "can the user do this" or "does this particular feature work".

### .5.3.2. STRUCTURAL TESTING

Structural testing is also called White box testing. This means a testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. White box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable.

### 5.3.3. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limiting type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing is an investigatory testing phase, where the focus is to have almost a destructive attitude and tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

### 5.3.4. TEST CASES

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not.

#### 5.3.4.1. Unit Test Cases

The software is being divided into different components and unit testing is performed on each of these modules. This section is repeated for all components.

#### 5.3.4.2. Integration Test Cases

Integration testing is a part of stress testing Involves integrating components to create a system or sub-system. May involve testing an increment to be delivered to the customer. In integration testing the test team has access to the system source code. The system is tested as components are integrated.

### 5.3.4.3. Validation Test Cases

This testing is done to see whether the integrated software is valid according to the user needs.

| Sl. No. | Test Case | Excepted Result | Test Result |
|---|---|---|---|
| 1 | On the click of light button on the desktop application | Lights are switched ON of the home automation | Successful |
| 2. | On the click of fan button on the desktop application | Fan are swithched ON | Successful |
| 3. | On the click of alarm button on the desktop application | Emergency Buffer is Switched ON | Successful |

**Table 5.1- Operations**

# CONCLUSION AND RECOMMENDATIONS

# 6. CONCLUSION AND RECOMMENDATIONS

Brain signals reflect the handled activities and controlling behaviour of the brain or the influence of the received information from other body parts either sensing or internal organs. Brain Computer Interfacing provides a channelling facility between brain and external equipment.

BCI applications have attracted the research community. Several studies have been presented in this paper regarding the growing interest in BCI application fields such as medical, organizational, transportation, games and entertainment, and security and authentication fields. It also demonstrates the various devices used for capturing brain signals.

These recording devices are divided into two main categories: invasive and non-invasive. Invasive category, which requires implanting surgery, is usually needed for critical paralyzed situations because of their higher accuracy rates achieved either spatially or temporally. On the other hand, the non-invasive category, as mentioned previously, has been widely spread in other application fields due to its advantages over the invasive one. Other challenges and issues posed as a result of utilizing brain signals have also been discussed along with some solutions offered by different algorithms at various BCI processing components.

# FUTURE WORKS SUGGESTED

# 7. FUTURE WORKS SUGGESTED

Computer application can be removed and direct contact of home appliances and brain wave can be implemented. Can add motion sensor, gas sensor, heat sensor which will alert the user in critical conditions. If configured then it can send a SMS based alert to assistant user. Disabled person can be given a robotic suit so that he can control the suit using brainwave and do things that ordinary peoples do.

# REFERENCES

# 8. REFERENCES

1.[Gentiletti, G. G., Taberning, C. B. y Acevedo, R. C. (2007). Interfaces Cerebro Computadora: Definición, Tipos y Estado Actual. Universidad Nacional de Entre Ríos, Paraná, Argentina.

2. http://arduino.cc/en/Main/ArduinoBoardUno

3. http://www.imserso.es/InterPresent1/groups/imserso/documents/binario/08_18idi.pdf. Ultimo acceso: 29 de Agosto de 2012.

4. http://www.NeuroSky.com/

5. Kollmann, A. y Simeoni, E. (2012). Exoesqueleto de Miembro Superior con Detección de Intención. Córdoba, Argentina.

6. Minguez, J. Tecnología de Interfaz Cerebro-Computador. Universidad de Zaragoza, España. 7. Minguez, J. Tecnología de Interfaz Cerebro-Computador. Universidad de Zaragoza, España. 8. NeuroSky® (2009). Brain Wave Signal (EEG) of NeuroSky®, Inc. Estados Unidos: autor. 9. Rodriguez, G., Fulgenzi, E. y Milstein, C. (2011). Día Mundial de la Esclerosis Lateral Amiotrófica (ELA). Argentina.

10. Sánchez, R. H. (2011). Brain Computer Interface (BCI) aplicado al control de dispositivos demóticos para incrementar la accesibilidad de las personas dependientes en el hogar digital y su entorno habitual Domo-BCI. Universidad de Valladolid.

11. http://www.universaldesign.com/

# APPENDIX

# APPENDIX I - CODE SNIPPETS

**Python Code For  Reading value from neurosky headband and processing it**

```
"""

Project             :
purpose             :       Reading value from neurosky
headband and processing it
developed on        :
last Modified on    :

"""

from NeuroPy import NeuroPy
import thread
import cursor
import time




tt =0                                                    # for
counting the number of times attention value increase
vv =0
running = True                                           # True
while the program is running
object1=NeuroPy("COM3")

thread.start_new(cursor.click,())                        #
Opening click() in cursor module in a thread
```

```
def brainwave():                                   # For
reading attention level from headband
    def attention_callback(attention_value):
        global vv,tt,running
        if running == False:
            return
        if tt>=2:
                cursor.initiate()                  #
calling initiate() in cursor module for mapping mouse
                tt=0



        print attention_value,                     #
Checking whether the value increased 5 times continuously
        if attention_value>vv and attention_value != 0 and
attention_value>35:
            vv=attention_value
            tt+=1
        elif attention_value<vv or attention_value==0:
            vv = attention_value
            tt=0
        return None

    try:
        if running==False:
            return

object1.setCallBack("attention",attention_callback) # call
function attention_callback when a new value for attention
is detected
```

```
        except:
            pass
        object1.start()



    thread.start_new(brainwave,())                  # opening
    the brainwave() in a thread to work in background


    tLastUpdate = 0
```

**Python Code For  Get signal from brainwave.py and using that to control mouse**

```
    """

    purpose           :       Get signal from brainwave.py and
    using that to control mouse

    """



    import win32api, win32con
    import time
    import tkMessageBox
    import Tkinter
    from Tkinter import *
    import thread
    import serial
```

```
x = 500
y =400                              # cursor y coordinate
box_x = x+10
box_y = y
                     # speed of mouse pointer

selection_delay=4

x_max = 1366
y_max = 768
brain_count = 0
previous_count = 0

arduino = serial.Serial('COM5', 9600, timeout=.1)
time.sleep(1) #give the connection a second to settle
var="1"

def initiate():                      # recive signal from
brainwave.py
    global brain_count
    brain_count+=1

def Single_Click(a,b):               # For singleclick in
appearing tkinter box

     win32api.SetCursorPos((a,b))

win32api.mouse_event(win32con.MOUSEEVENTF_LEFTDOWN,a,b,0,0
)
```

```
win32api.mouse_event(win32con.MOUSEEVENTF_LEFTUP,a,b,0,0)


def Light_One(root):
     global brain_count
     root.destroy()
     arduino.write(var)
     print "light one"
     box()
     return

def Close(root):
     global brain_count
     root.destroy()
     print "close"

     return

def Light_Two(root):
     global brain_count
     root.destroy()
     print "light two"
     box()
     return
def Fan(root):
    global brain_count
    root.destroy()
    print "fan"
    box()
    return
```

```
def mouse_on_box(root):              # positioning mouse
cursor on dialog box
    global x,y,brain_count,box_x,box_y,previous_count
    clicked = False
    while not clicked:
        win32api.SetCursorPos((box_x+60,box_y+50))
        if brain_count>previous_count:
            previous_count=brain_count
            clicked=True
            #root.destroy()
            Single_Click(box_x+60,box_y+170)
            return
        elif not clicked:
            time.sleep(selection_delay)
        win32api.SetCursorPos((box_x+60,box_y+90))
        if brain_count>previous_count:
            previous_count=brain_count
            clicked=True
            #root.destroy()
            Single_Click(box_x+60,box_y+50)
            print"hi"
            return
        elif not clicked:
            time.sleep(selection_delay)
        win32api.SetCursorPos((box_x+60,box_y+130))
        if brain_count>previous_count:
            previous_count=brain_count
            clicked=True
            #root.destroy()
            Single_Click(box_x+60,box_y+90)
            print"hi"
```

```
            return
        elif not clicked:
            time.sleep(selection_delay)
        win32api.SetCursorPos((box_x+60,box_y+170))
        if brain_count>previous_count:
            previous_count=brain_count
            clicked=True
            #root.destroy()
            Single_Click(box_x+60,box_y+130)
            return
        elif not clicked:
            time.sleep(selection_delay)

def box():                                 # for opening
dialogbox
    global x,y,brain_count,box_x,box_y,previous_count
    root = Tk()
    root.update_idletasks()
    root.resizable(0,0)
    size = tuple([180,170])



    root.wm_attributes("-topmost" , -1)
    box_x = x+10
    box_y = y

    if (x+10+130>1360):
        box_x = 1360-170
    if (y+170>760):
        box_y = 760-200
    root.geometry("%dx%d+%d+%d" % (size + (box_x, box_y)))
```

```
    b1 = Button(text = "LIGHT ONE",font=("Raleway",
10),command=lambda: Light_One(root), width=13)
    b1.pack()



    b2 = Button(text = "LIGHT TWO",font=("Raleway",
10),command = lambda: Light_Two(root),width=13)
    b2.pack()

    b3 = Button(text = "FAN",font=("Raleway", 10),command
= lambda: Fan(root),width=13)
    b3.pack()

    b4 = Button(text = "CLOSE",font=("Raleway",
10),command = lambda: Close(root),width=13)
    b4.pack()



    b1.place(x=30,y=10)
    b2.place(x=30,y=50)
    b3.place(x=30,y=90)
    b4.place(x=30,y=130)
    thread.start_new(mouse_on_box,(root,))



    root.mainloop()




def click():                              # calling
move_horizontal(), move_vertical, box() in order
```

```
global x,y,brain_count,previous_count
x = 500
y =400
while(1):
    if brain_count ==1:
            previous_count = brain_count
            box()
```

# APPENDIX II - SCREEN SHOTS

## A. DESKTOP APPLICATION

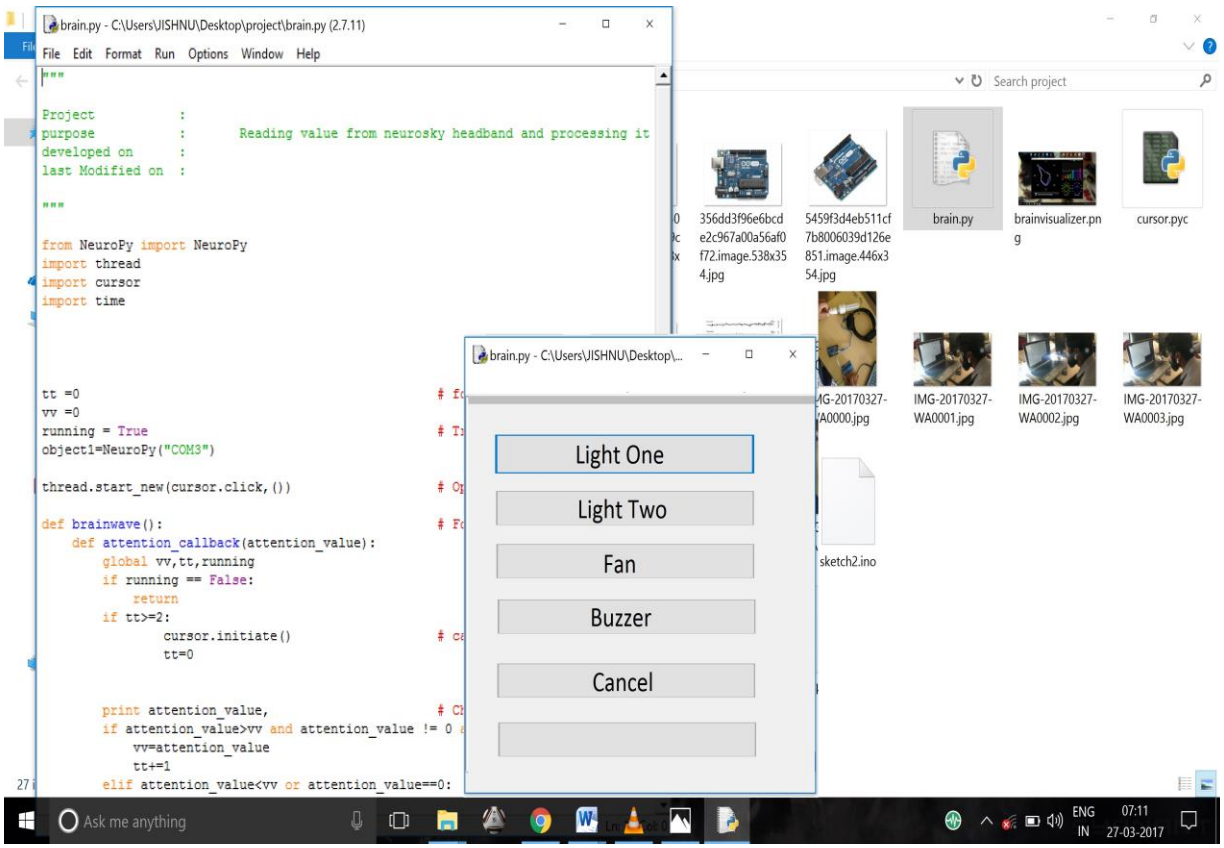


**Figure a: Desktop Application**

## B. BRAIN VISUALIZER


**Figure b: Brain Visualizer**

## C. HARDWARE

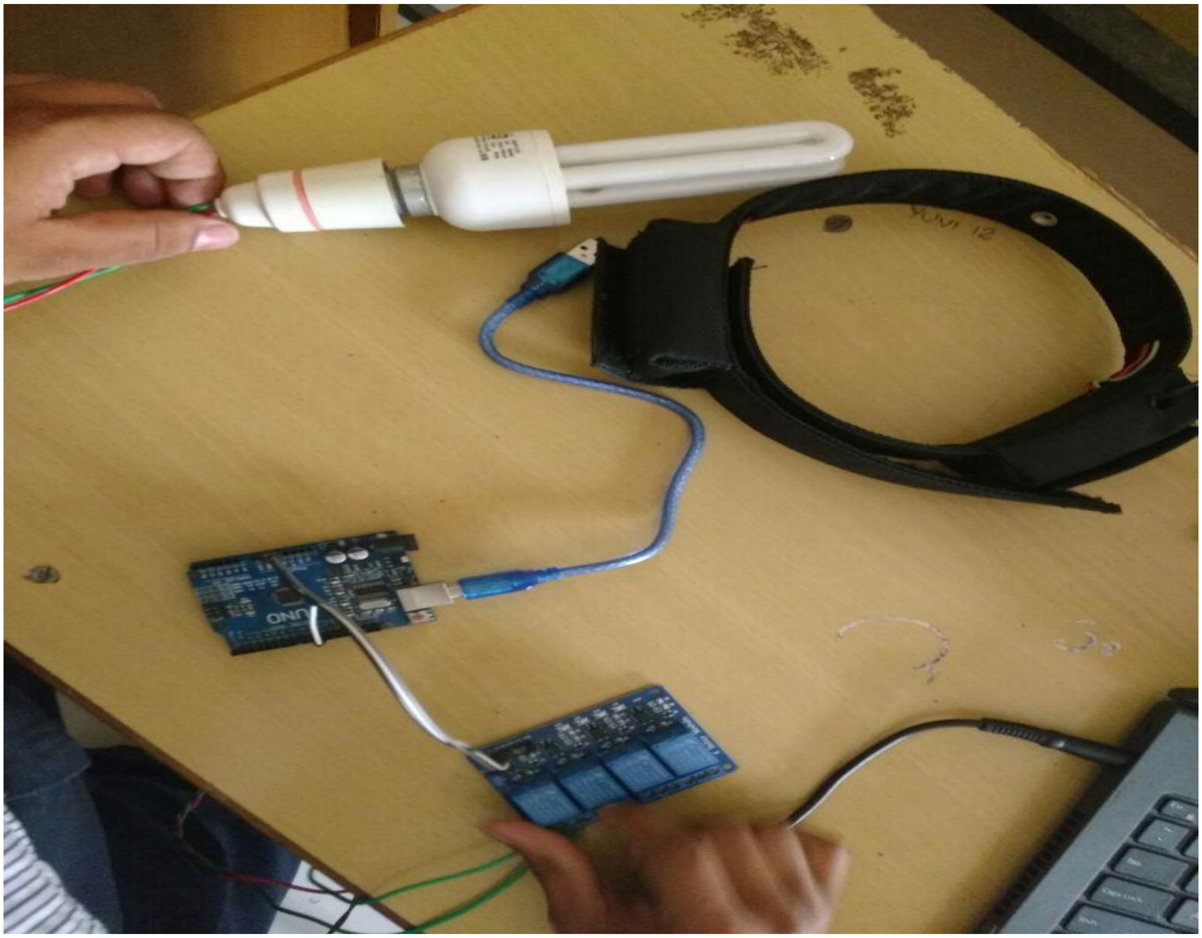

**Figure c: Hardware**

## D. HARDWARE CONNECTION

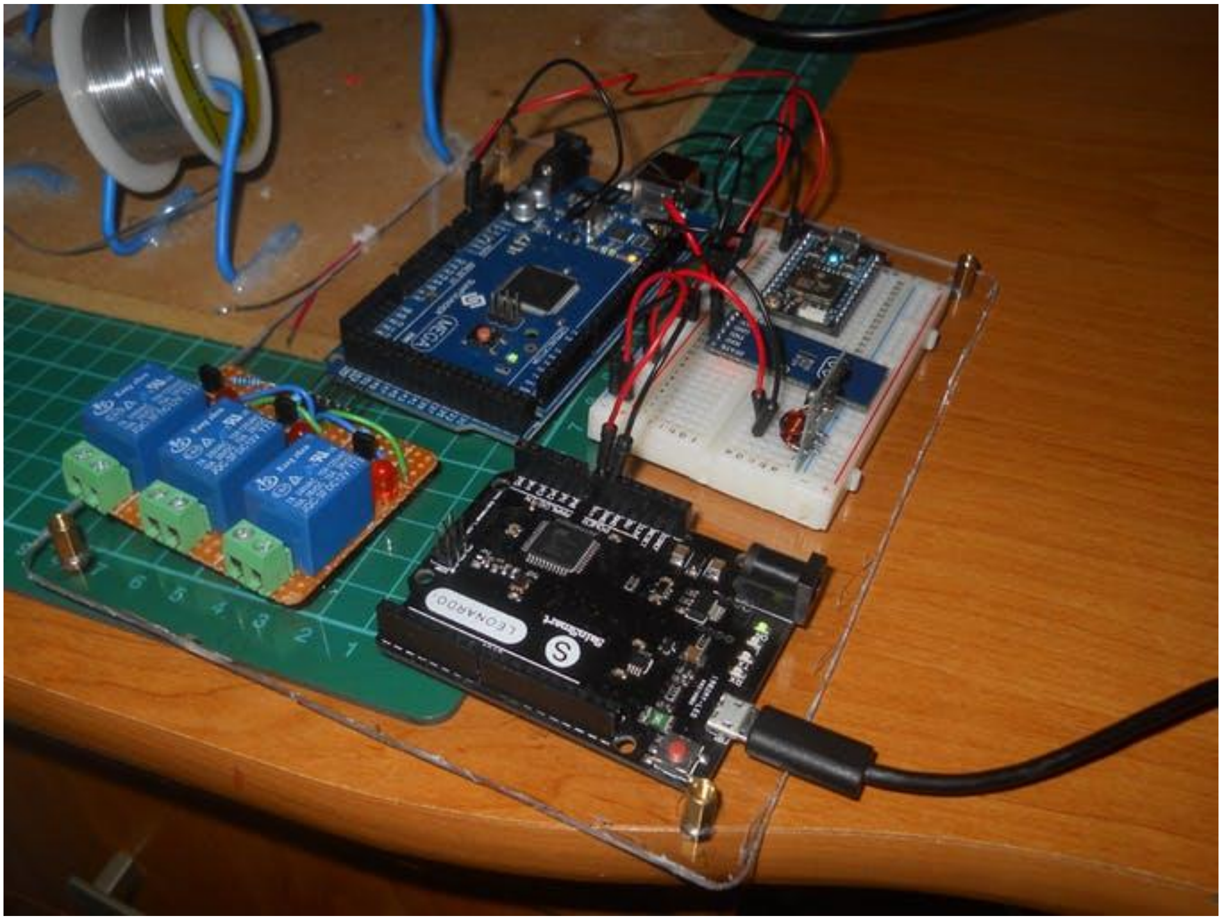

**Figure d: Hardware Connectiion**

## E. WORKING OF PRODUCT

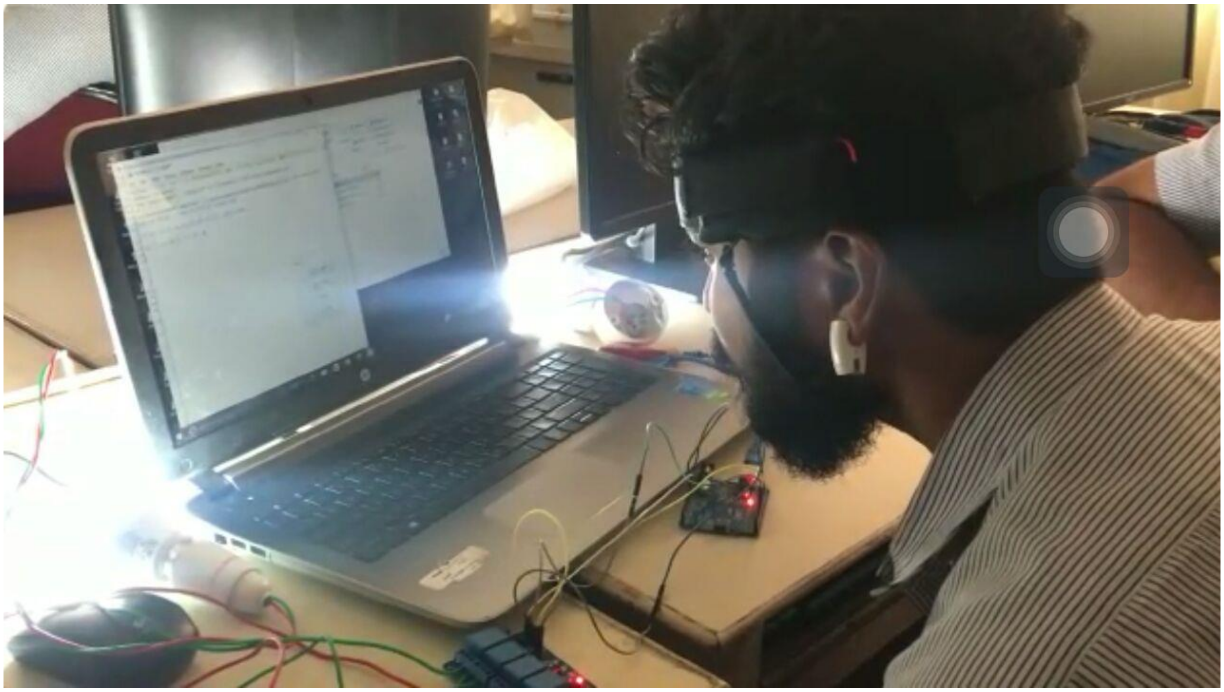

**Figure e: Working Of Product**