



Investigators of Smart-Grids and Renewable Energies for Electric Networks

<http://www.cs.csub.edu/~is-green/>

Phone:
(661) 654-6005

Fax:
(661) 654-6960

Email:
isgreen@cs.csubak.edu

California State University Bakersfield
Department of Computer & Electrical Engineering
& Computer Science

Mail Stop: 63 SCI
9001 Stockdale Highway
Bakersfield, California 93311-1022
Office: Science III 339



MSEIP-2

Summer Outreach Programs 2015

Tutored by:

Sheriff Sadiqbatcha

Brain Computer Interface For Power Industry Applications

General Overview

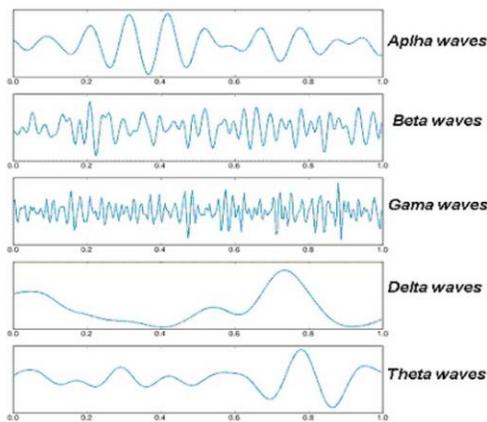
All executive and motor functions of the human body, whether voluntary or involuntary, has one common root. Mankind has always been fascinated by the intricacy of the human brain but has never been able to unlock its secrets. After all, tracing through a network of 100 billion neurons is an overwhelming task by any-ones standards. Over the years as new technologies emerge, doors have been opened in the path to better understanding the brain and its functional structure. Thus far solely been reserved for the medical field, recent advancements in safety and accessibility has made possible such technologies to be utilized in other applications. Electroencephalography (EEG) is one such technology, which is used to record electric potentials generated by neurons in the brain through external placement of electrodes on the scalp. Since electrode placement is peripheral and it solely functions as a medium of gathering data without interfering with any of the brains functionalities, EEG can be utilized without the supervision of a medical professional. This allows it to be implemented in a wide variety of applications without any concerns of health risks to the targeted user. Although EEG has a higher temporal resolution than functional Magnetic Resonance Imaging, it significantly lacks in its ability to deliver comparable spatial resolution. Moreover, possibilities of noise and inconsistencies in the signal caused by sources not associated with the brain has to be considered and addressed in-order for the signal to be viable. Offering such opportunities and challenges, EEG signals, provide reliable and yet costly interface for human-machine interactions.

Background

Electroencephalography is the measurement of electrical activity generated by neurons in the brain. A composite reading of the electrical signals generated by neurons in a given section of the brain is non-invasively measured by placing electrodes on the scalp. Strategic placement of

electrodes on different areas of the scalp and analysis of the signals yield a variety of information that relates to the current mental and emotional state of the brain. Ishihara and Yoshii found a distinct theta activity of EEG over frontal midline area during mental tasks, which is called Fm θ . It has been found that some electrical activity in the Fm θ area can be linked to a state of mental concentration.

The EEG signal obtained from the frontal lobe, when transformed from time domain to frequency domain using Fast Fourier Transform, provide us with clearly distinguishable distributions of frequencies linked to different mental and emotional states.



TYPE	FREQUENCY	MENTAL STATES
Delta	0.1Hz to 3Hz	Deep, dreamless sleep, non-Rem sleep, unconscious.
Theta	4Hz to 7Hz	Intuitive, creative, recall, fantasy, imaginary, dream.
Alpha	8Hz to 12Hz	Relaxed, but not drowsy, tranquil, conscious.
Low Beta	12Hz to 15Hz	Formerly SMR, relaxed yet focused, integrated.
Midrange Beta	16Hz to 20Hz	Thinking, aware of self & surroundings.
High Beta	21Hz to 30Hz	Alertness, agitation.
Gamma	30Hz to 100Hz	Motor functions, high mental activity.

Analysis of the data gathered from the Fm θ area in frequency domain enables us to measure one's relative level of alertness at any given time. Using this principle, a subject's attention level can be monitored in real time and used as a trigger or command to execute a predetermined task.

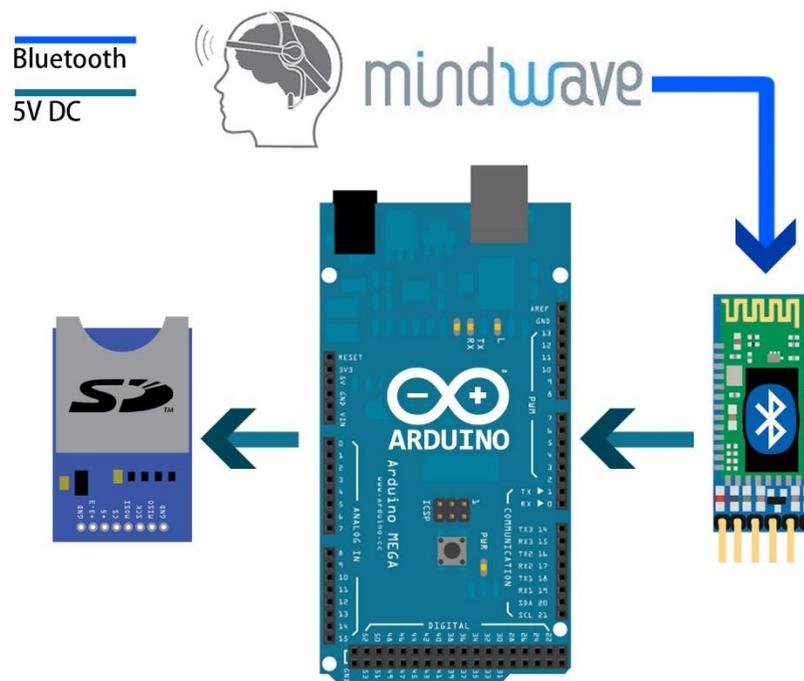
Hardware & Software Design

In this system the Mindwave Mobile headset by NeuroSky, Inc. is utilized to read the EEG signals. Not only is this device affordable in comparison to other consumer EEG sensors in the market, it is also very compact. Its wireless interface makes it ideal for wearable electronic applications. NeuroSky's eSense algorithm processes the EEG signals onboard and wirelessly transmits the calculated attention and meditation values through Bluetooth to a master device at the rate of 1Hz.

A HC-05 Bluetooth module, programmed with the MAC address and password of the Mindwave headset, acts as the master device that

automatically pairs with the headset once both devices are powered on. The data received by the HC-05 module is then transmitted to an Arduino microcontroller.

The Arduino is an open-source microcontroller board, incorporating an ARM processor, which can be configured by writing software using its dedicated development environment. In this system, the Arduino microcontroller is used to gather and analyze the data from the headset to be used in real-time or stored for future analysis.



Interfacing Mindwave Mobile to Arduino

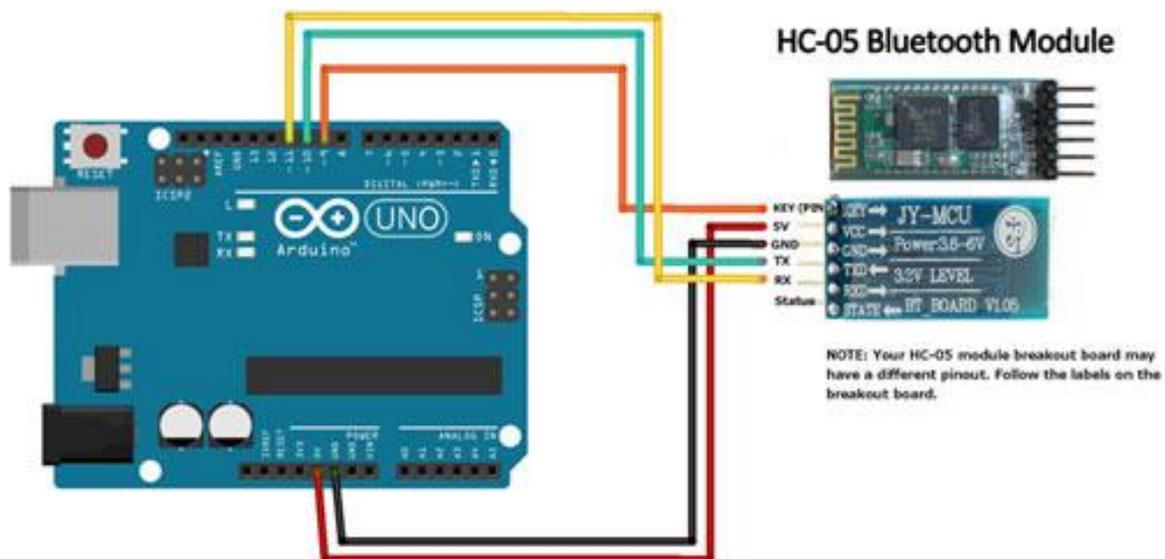
1. Pair the MindWave Mobile to your computer to acquire its MAC address
(Control panel → Hardware and Sound → Devices and Printers → Right click on the headset → Properties → Bluetooth - Then note your Unique Identifier Number)
2. Unpair the headset from your computer after noting the MAC address.
3. Configure the HC-05 Module with the MAC address and password of your headset.
4. Run the given startup code to test the system.

Configuring HC-05 Module

Hardware Required

- Arduino Board
- Mindwave Mobile Headset (Slave Device)
- HC-05 Bluetooth Module (Master Device)
- USB Cable for Arduino
- Breadboard
- Jumper wires

Connect the HC-05 module to the Arduino Uno as per the following diagram.



Press and Hold the push button on the HC-05 module for few seconds while powering on the Arduino. Release the button once both devices are power on. This procedure initiates Data Mode on the HC-05. Note: The LED on the HC-05 module should blink at low toggle (blink slowly) indicating that the device is on Data Mode, if it blink's at high toggle then power down the device and try again.

Copy the following code, paste it in the Arduino IDE and upload it to the board.

CODE

```
#include <SoftwareSerial.h>
```

```

SoftwareSerial BTSerial(10, 11); // RX | TX

void setup()
{
  pinMode(9, OUTPUT); // this pin will pull the HC-05 pin 34 (key pin) HIGH
                       // to switch module to AT mode

  digitalWrite(9, HIGH);
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400); // HC-05 default speed in AT command mode
}

void loop()
{
  // Keep reading from HC-05 and send to Arduino Serial Monitor
  if (BTSerial.available())
    Serial.write(BTSerial.read());

  // Keep reading from Arduino Serial Monitor and send to HC-05
  if (Serial.available())
    BTSerial.write(Serial.read());
}

```

Once the board is programmed open the serial monitor at 38400 baudrate and type in the following commands (color coded in orange) sequentially. Note: Do not use the quotation marks.

1. AT+NAME="PANTECH"
2. AT+UART="57600,0,0"
3. AT+ROLE="1"
4. AT+PSWD="1234"
5. AT+CMODE="0"
6. AT+BIND="XXXX,YY,ZZZZZ" (Mindwave Unique Number)
7. AT+IAC="9E8B33"

8. AT+CLASS="0"

9. AT+INQM="1,9,48"

You should get an "OK" after each command indicating that it has been successfully written to the device.

You have now successfully configured the HC-05 module to act as a master device that automatically pairs with the Mindwave Mobile headset once both devices are powered on.

Interfacing the Mindwave Mobile Headset

From this point we are only interested in the data transmitted from the HC-05 module to the Arduino, therefore the wire connecting the TX pin from the Arduino to the RXD pin on the HC-05 can be removed.

The following code parses the data from the headset and displays the attention and meditation values of the user on the serial monitor. You will use this code as the foundation and build upon it utilizing the skills you have acquired over the past week.

CODE

```
////////////////////////////////////  
// Arduino Bluetooth Interface with Mindwave  
////////////////////////////////////  
  
#define BAUDRATE 57600  
#define DEBUGOUTPUT 0  
  
#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()

{
  Serial.begin(BAUDRATE);      // USB
}

////////////////////////////////////
// Read data from Serial UART //
////////////////////////////////////
byte ReadOneByte()

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

#ifdef DEBUGOUTPUT
  Serial.print((char)ByteRead); // echo the same byte out the USB serial
                                // (for debug purposes)
#endif

  return ByteRead;
}

////////////////////////////////////
//MAIN LOOP//
////////////////////////////////////
void loop()

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

```

```

payloadLength = ReadOneByte();

if(payloadLength > 169) //Payload length can not be greater
                        //than 169

return;
generatedChecksum = 0;
for(int i = 0; i < payloadLength; i++)
{
payloadData[i] = ReadOneByte(); //Read payload into memory
generatedChecksum += payloadData[i];
}

checksum = ReadOneByte(); //Read checksum byte from
                          //stream
generatedChecksum = 255 - generatedChecksum;
//Take one's compliment of generated checksum

if(checksum == generatedChecksum)
{
    poorQuality = 200;
    attention = 0;
    meditation = 0;

    for(int i = 0; i < payloadLength; i++)
    {
        // Parse the payload
        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;
    } // switch
} // for loop

#if !DEBUGOUTPUT
    if(bigPacket)
    {

        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");

        // *** Add your code here ***

    }
#endif
    bigPacket = false;
}
else {
    // Checksum Error
} // end if else for checksum
} // end if read 0xAA byte
} // end if read 0xAA byte
}

```

USEFUL LINKS

[Arduino References](#)
[Mindwave Mobile](#)

Project Description

Good work! You have successfully interfaced the Mindwave Mobile headset with the Arduino Microcontroller. Now that you have the user's attention value, there's endless possibilities on how you can put it to use.

One application of this technology is its use in mind-controlled electric devices. As we know, any extended level of focused attention towards a specific object of interest can be traced using this system, therefore can be used as a command to execute the objects functionality i.e. turn a device on or off. This allows the system to be adapted as a control for certain electric devices, thereby can possibly grant disabled individuals further control of electronics in their home.

Another application of such technology can be realized in high risk industry. EEG sensors can be incorporated in the helmets of workers so that their attention values can be logged during their work day. The readings can be wirelessly transmitted to a central system allowing a supervisor to monitor the status of each individual worker in real time. Low attention level for an extended period of time can be detected and used to trigger a buzzer or notify the supervisor so that immediate actions can be taken to remove the worker from the high risk environment, thus tremendously reducing the chances of accidents.

In our case, considering the limited time frame and in order to cover a wide span of audience, let's consider building a fun arcade game. Remember we want to show that this technology isn't only reserved for the medical field, it versatile and affordable enough to be used for so many other applications. Here's the basics of the system, you may add additional features as you please. Use the skills you have acquired so far to interface two of these headsets to the same Arduino onboard an RC car. It's going to be a tug of war game where two users, standing on either side, will focus on the toy car in the middle. The person who concentrates more gets to pull the car towards them, the game is won when the car crosses the finish line on either side. Be creative, add more features like a system to keep score or to display the current attention level. Use your imagination, take pride in your work and have fun doing it. Good luck!