

# MindWave and Arduino

## Features

- Interfaces with the MindWave with the RF USB dongle for stand-alone operation
- Acquiring the Attention and Meditation eSense values from the data stream.
- Displaying your Attention level using LEDs

## Introduction

This step-by-step guide is intended to quickly get a typical Arduino board user up and running with the MindWave. This guide will show how to setup the RF USB module quickly for communication with the MindWave and gives example code to parse the output stream. The parsed output stream can then be used to perform various tasks such as to control the speed of a motor, to light up LEDs, and to control the position of a servo motor, etc.

## Interfacing to the MindWave

Interfacing to the MindWave requires the user to hack the USB dongle. The following describes easiest way to quickly get up and running using an Arduino board. The Arduino is sold by a vendor listed in the sources section. The MindWave and USB dongle are available at the NeuroSky Store.

## Required Materials

This application requires the following materials:

- MindWave
- Arduino Diecimila or Duemilanove
- RF USB dongle
- Arduino development environment
- Avago HDSP-4832 10-Element LED bar graph array
- 4 75 $\Omega$  resistors
- 6 74 $\Omega$  resistors
- Breadboard
- SparkFun Level Converter
- 12V Power Source (included with Arduino Board)

## Section 4 – Detailed Instructions

### System Overview

The system uses the Arduino's hardware UART for both USB communication and Bluetooth communication. After the system is up and running, the Arduino may be detached from the computer and run independently.

The basic programming flow is as follows:

1. Connect the dongle to the Arduino board..
2. Connect the LED array to the Arduino board.
3. Download the provided example code into the Arduino board using the Arduino development environment.
4. Power up the whole system and observe the Attention eSense value on the LED array.

## Detailed Instructions

### Connecting the dongle to the Arduino Board

Connecting the dongle to the Arduino Board requires removing the plastic covering from the dongle and soldering wires to 4 of the pins.

1. Remove the plastic covering from the USB dongle.
2. Once the cover is off, you should see the board shown below (Note: Bottom View Orientation)

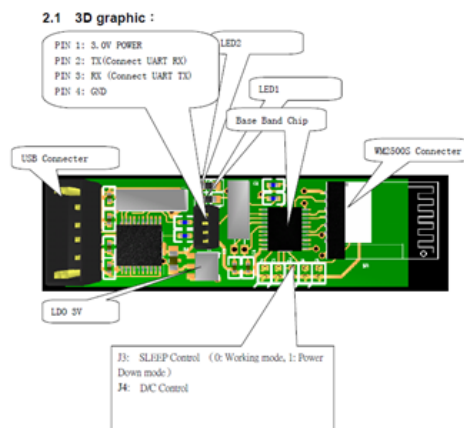


Figure 1: Dongle Schematic

3. Find the 4 pins, located in the center of the board.
4. Starting at the end with the ‘ + ’ sign the pin assignments are:
  - (a) Pin 1 (starting at +): 3.0 V Power
  - (b) Pin 2 : TX

## Section 4 – Detailed Instructions

- (c) Pin 3 : RX
  - (d) Pin 4 : Ground
5. Solder wires to these 4 pins, leaving enough wire to make the necessary adjustments.
  6. Connect the boards together, according to the schematic shown below

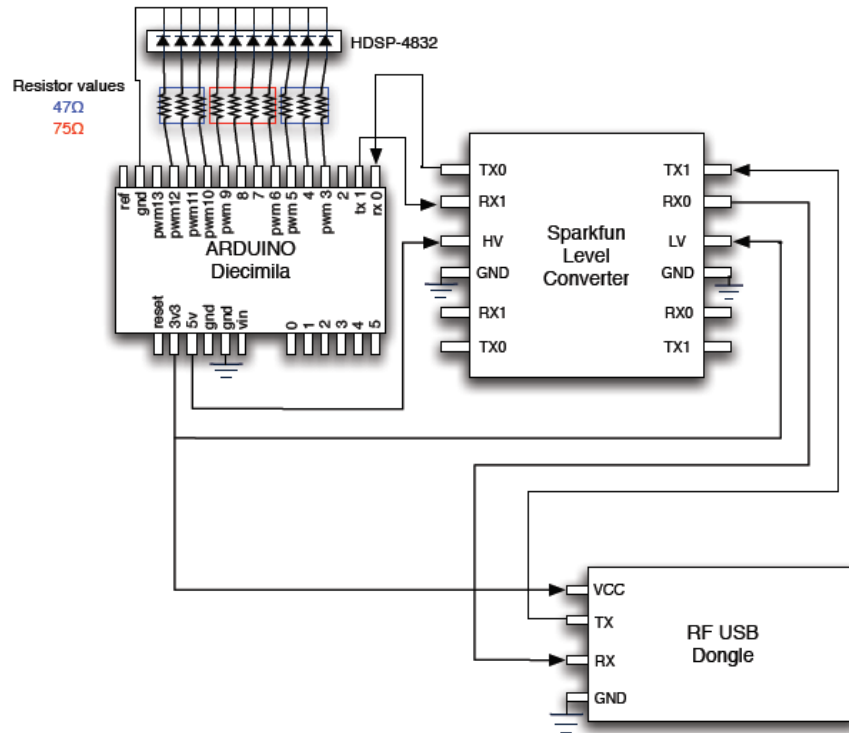


Figure 2: Circuit Schematic

## Cutting Traces on the dongle

In order for the dongle to properly receive the TX signal from the Arduino, two traces on the board must be cut. Looking at the dongle from the top, there are two traces leading from the TX and RX pins to the CH340 chip. With an X-Acto knife, cut these two traces. After the traces have been cut, check the connections with a DMM to ensure that the CH340 chip (pins 3 & 4) is no longer connected to the TX and RX pins.

## Powering the Arduino

The Arduino may be powered by USB, by battery, or by an external power supply. Please see the Arduino hardware page here <http://arduino.cc/en/Main/Hardware> for more information.

## Getting started with the Arduino development environment

Please see Arduino Getting Started guide here <http://arduino.cc/en/Guide/HomePage>

## Arduino Code Description

Arduino code is based on the C/C++ language and requires very little else to get started. All the details or regular microprocessor development such as processor clock speed, programming speed, clock sources, etc. are all initialized by the Arduino core. With this in mind, the Arduino system will still be able to perform much of the required tasks with no problem at all. The simplicity of using the microprocessor peripherals combined with the C language along with the sample code provided provides for very quick interface to the MindSet. The user is encouraged to extend the Arduino code by appending code after the indicated section where the packet has been checked for a valid checksum.

Thereafter, the user may use the eSense values to do one of many things possible. Some of which is to control the brightness of an LED, the height of an object, etc.

The bulk of the code is used for parsing the packet data. For further information, see the Mind-Set Communication Protocol document in the MindSet Development tools (MDT). The MDT is available at the NeuroSky Store for free. <http://store.neurosky.com>

## Syncing the Arduino Board

1. Open Mind\_Wave.pde
2. Tools > Board > Arduino Diecimila
3. In Control Panel > Select Device Manager
  - (a) Look in “Ports” to see which COM port the USB Serial Port is connected to
4. Tools > Serial Port > Select the corresponding COM port

## Running Mind\_Wave.pde

1. Once you have synced the program and your board, disconnect the TX and RX lines coming out of the Arduino
2. File > Upload to I/O Board
3. Once the program displays “Done Uploading”, you can disconnect the USB serial port and run the Arduino from independent power source.
4. Reconnect TX and RX lines.
5. Power up the Arduino Board via power source
6. Arduino should have green PWR LED lit and dongle should have red LED lit
7. Turn on Mindwave Headset
8. Hit “Reset” button to pair the MindWave to the dongle.
9. Once connected MindWave LED and dongle LED will both turn blue.
10. Enjoy observing your Attention eSense value on the LED array.

## Switching to Attention/Meditation Mode

The default eSense reading in Mind\_Wave.pde is attention. To switch the program to measure your meditation level, requires changing a few lines of code. Inside the pde file, locate the code shown below.

## Section 4 – Detailed Instructions

```
#if !DEBUGOUTPUT

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

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);
  if(attention > 40)
    digitalWrite(GREENLED1, HIGH);
  else
    digitalWrite(GREENLED1, LOW);

  Serial.print(" Time since last packet: ");
  Serial.print(millis() - lastReceivedPacket, DEC);
  lastReceivedPacket = millis();
  Serial.print("\n");

  switch(attention / 10) {
  case 0:
```

Figure 3: MindWave.pde

Inside these lines of code, replace “attention” with “meditation” to operate in meditation. If you are switching back to attention mode, replace “meditation” with “attention”.

## Material Sources

The Level Converter and Arduino board can be obtained from:

Sparkfun Electronics  
6175 Longbow Drive, Suite 200  
Boulder, CO 80301  
<http://www.sparkfun.com>

More information about the Arduino can be found here: <http://www.arduino.cc>

**Corporate Address**

NeuroSky, Inc.  
125 S. Market St., Ste. 900  
San Jose, CA 95113  
United States  
(408) 600-0129

Questions/Support: <http://support.neurosky.com>  
or email: [support@neurosky.com](mailto:support@neurosky.com)

Community Forum: <http://developer.neurosky.com/forum>

Information in this document is subject to change without notice.

Reproduction in any manner whatsoever without the written permission of NeuroSky Inc. is strictly forbidden. Trademarks used in this text: eSense™, ThinkGear™, Mind-Kit™, NeuroBoy™ and NeuroSky® are trademarks of NeuroSky, Inc.

**Disclaimer:** The information in this document is provided in connection with NeuroSky products. No license, express or implied, by estoppels or otherwise, to any intellectual property rights is granted by this document or in connection with the sale of NeuroSky products. NeuroSky assumes no liability whatsoever and disclaims any express, implied or statutory warranty relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or non-infringement. In no even shall NeuroSky be liable for any direct, indirect, consequential, punitive, special or incidental damages (including, without limitation, damages for loss of profits, business interruption, or loss of information) arising out of the use of inability to use this document, even if NeuroSky has been advised of the possibility of such damages. NeuroSky makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. NeuroSky does not make any commitment to update the information contained herein. NeuroSky's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.