Charge for a whole day: Extending Battery Life for BCI Wearables using a Lightweight Wake-Up Command

Supplementary Material: The case for a wakeup command

Mohit Agarwal
Georgia Institute of Technology
magarwal37@gatech.edu

Raghupathy Sivakumar
Georgia Institute of Technology
siva@gatech.edu

BCI PLATFORMS

A typical BCI architecture consists of three core components: (i) an *electrode sensor array* placed on the scalp, (ii) a *hardware platform* to digitize, locally process and transmit the brainwaves, and (iii) an *algorithmic processing platform* to analyze and decode the received brainwaves in an application specific manner. Scalp electrodes provide a conductive medium for the signal to reach the hardware interface. The hardware platform includes AFEs/ASICs (Analog Front-End or Application Specific IC) for digital sampling, ADC (Analog to Digital Conversion), and noise suppression. A wearable device worn by users embeds the first two components responsible for acquisition, local processing and transmission of sensor data, and is also referred to as the "cap-end". The "mobile-end" serves as the algorithmic processing platform, and is typically either a smartphone or a computer.

Most of the commercially available BCI devices are application oriented e.g., Muse for meditation [1], Aurora for sleep analysis[2], etc. However, some of the devices are general-purpose and/or research grade devices e.g. Emotiv [3], OpenBCI [4]. Compared to medical or research-grade BCI devices, wearable BCI devices are inferior mostly in terms of signal quality, but are cheaper and easier to use. A list of all the available consumer devices in the market that cost less than $1000 is available[5]. Several of the available BCI hardware either do not perform well in terms of available signal quality and usability, or provide severely restricted access to the system design and raw EEG data. Moreover, most of them only provide an SDK to develop applications at the mobile-end with a non-programmable hard-coded firmware at the BCI cap. We use the OpenBCI platform as the representative BCI hardware for our study as it bundles all the required features (transparent hardware design

[1]https://choosemuse.com
[2]https://sleepwithaurora.com/
[3]https://emotiv.com
[4]https://openbci.com
[5]http://www.autodidacts.io/neurotech-hardware-roundup-eeg-bci-tdcs-neurofeedback/

CHI '20, April 25–30, 2020, Honolulu, HI, USA.
2020 Association of Computing Machinery.
ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376738

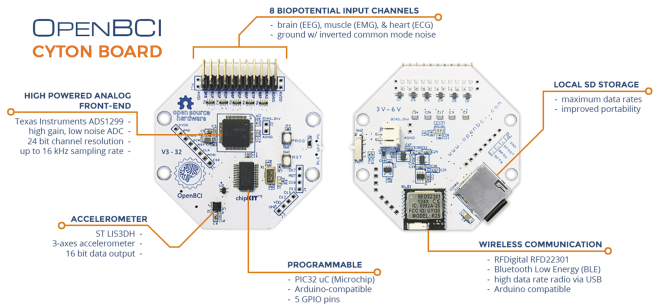


Figure 1. OpenBCI Architecture

and software algorithms along with full access to raw EEG data) in a single piece of hardware. However, we also experiment with the Muse headband to demonstrate the feasibility and extensibility of our analysis to the other available BCI platforms.

OPENBCI ARCHITECTURE

OpenBCI is an open-source, low-cost, programmable interface to access raw EEG signals. It has the capability to connect with upto 16 electrodes at a time, amplifying and digitizing the signals at 250Hz. As shown in Fig. 1, the architecture of the OpenBCI board consists of the three major components.

1. **Analog Front-End (ADS1299):** Designed and manufactured by Texas Instruments [6] for bio-signal measurements, this IC is responsible for digitizing and amplifying the EEG signals. It is a low-power, 8 channel, 24-bit ADC with built-in PGA (Programmable Gain Amplifier).
2. **Microcontroller (PIC32):** This Microchip PIC32 [7] Microcontroller is the central component of the OpenBCI board. It configures and coordinates with all the other ICs on the board to get data, arranges it, and transmits it to the radio module for forwarding to the "mobile unit". It is capable of executing instructions at 50MHz (default for OpenBCI is 40MHz). The program memory size and RAM is 128KB and 32KB respectively. *PIC32 enables the local processing on the OpenBCI board.*
3. **Radio (RFDuino):** It is a finger-tip sized, low-cost, radio module, enabled with a μC to transmit the sensor data to the mobile-end through Bluetooth Low-Energy (BLE). The OpenBCI uses RFD22301.

[6]http://www.ti.com/product/ADS1299
[7]https://www.microchip.com/wwwproducts/en/en557425

Settings	Default	Configurable	Power
μC Clock Rate (MHz)	40	6 - 80	20mA (@3.6V)
ADC Clock Rate (MHz)	2.048	No	120μW
ADC Channels	8	1 - 16	-
Data Rate(SPS)	250	250 - 16k	-
PGA	24x	1x-24x	-
Radio (RFDuino)	ON	ON/OFF	11.8mA (@3V)

*- the information is not available in the datasheet

Table 1. Potential Control Knobs in OpenBCI Board

Other components include an accelerometer (LIS3DH) and an SD card slot for 3-axis motion detection and external storage respectively.

POWER ANALYSIS

Macro Power Analysis

The OpenBCI board, in its default development form, requires 4 AA (1.5V, 2300mAh each) batteries. However, using 4 AA sized batteries is clearly not suitable for wearable devices due to weight and safety considerations (as the platform is being worn on a user's head). Thus, we first perform a power analysis on the platform as-is, and then extend the analysis for a typical wearable battery. Specifically, we use the battery specifications of an Apple Watch (250mAh, 3.8V, 0.94Wh) [8] and convert it into OpenBCI voltage requirements (6V, equivalent to 150mAh). To estimate the default battery life, we measure the current drawn in the hardware module and project the approximate battery life by assuming a constant voltage till the battery discharges [9]. Our experiments show that the battery life is only 3.42 hours. The advertised battery life for Emotiv EPOC+ (6 hrs on 680mAh) and Muse 2014 headset (5 hrs) also confirms our analysis of battery life for wearable BCI devices.

Micro Power Analysis

We now take a deeper look to identify the main source of power drain for a wearable BCI device. We identify the micro-components of the board along with their default settings, reconfigurability, and individual power requirements. We define "control-knobs" as those micro-components that are reconfigurable inside the OpenBCI board, and could possibly create a significant impact on battery life. We tabulate such micro-components in Table 1 [10] and explain them below.

- μC/ADS Clock rate: This represents the operating frequency of PIC32 microcontroller and ADS1299 IC respectively, which directly affects their processing speed. As we can see from Table 1, the power consumed by the ADS clock oscillator is very low but high for μC (Remember from the previous section that OpenBCI draws an average current of 43.78 mA). So, we consider μC clock rate as one of our control knobs.

[8] http://www.onefruit.co/blog/2015/06/29/how-big-is-the-42mm-apple-watch-battery/

[9] We turn off the accelerometer for this particular analysis.

[10] The list is not exhaustive.

Control Knob	Mean Power Deviation	Relative Importance
f	7.5554	62.90%
c	0.3804	3.17%
g	4.0757	33.93%

Table 2. Power Analysis

- ADC Channels: This denotes the form-factor of the device, i.e. the total number of channels from which EEG data is sampled simultaneously. Power consumption data per channel is not reported in the ADS 1299 datasheet, hence we consider this as our control knob for the power consumption analysis.
- Data Rate: The number of EEG samples recorded per second is known as the data rate. Following Nyquist Sampling Theorem, decreasing the data rate results in aliasing of the frequency components higher than (Data rate)/2. However, in the case of OpenBCI, it is set to 250SPS which is at its minimum value set by ADS1299.
- PGA (Programmable Gain Amplifier): PGA is an electrical amplifier with a controllable gain through external digital or analog signals. We consider PGA for the power consumption analysis.

Thus, for the power consumption analysis, our focus is on (i) μC clock rate (**f**), (ii) number of ADC channels (**c**), and (iii) programmable gain (**g**). The radio module will be turned off in the "low-power" mode, and hence we do not consider it for the power analysis.

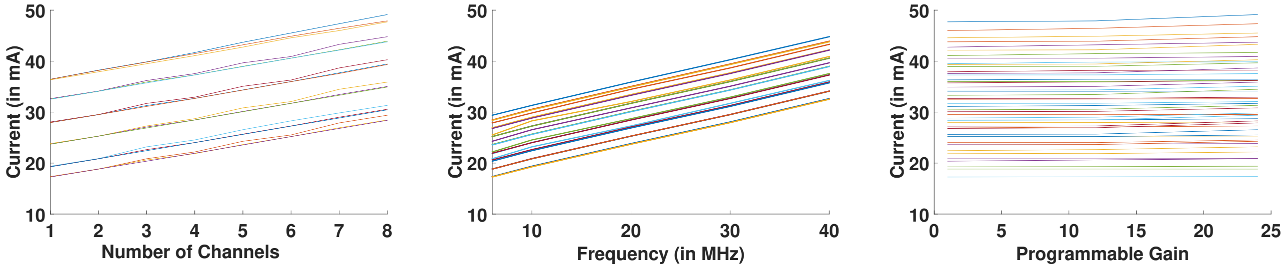
To evaluate the impact of each parameter on the OpenBCI battery life, we run an experiment to measure the average current drawn (in mA at constant voltage) for a specific ($\mathbf{f_i}$,$\mathbf{c_j}$,$\mathbf{g_k}$) from, $\mathbf{f_i} \in \{48,40,30,20,10,6\}MHz$, $\mathbf{g_j} \in \{24,12,1\}$ and $\mathbf{c_k} \in \{8,7,6,5,4,3,2,1\}$. For each (**f**,**c**,**g**), we take 5 snapshots and average them to reduce the measurement noise variations, and repeat for all such possible ($\mathbf{f_i}$,$\mathbf{c_j}$,$\mathbf{g_k}$) i.e.a total of 142 data points. Once we have the average power consumed for all permutations, we define a metric "average power deviation" to evaluate the impact of each knob on the battery life.

For $\mathbf{f_i}$, we calculate the average power deviation over the other two variables as,

$$\frac{1}{|I|}\sum_{j,k} Var_i(\mathbf{f_i},\mathbf{c_j},\mathbf{g_k}) \qquad (1)$$

i.e. we fix ($\mathbf{c_j}$,$\mathbf{g_k}$), and calculate the variance over all possible $\mathbf{f_i}$, and average over ($\mathbf{c_j}$,$\mathbf{g_k}$). We calculate a similar metric for $\mathbf{c_j}$ and $\mathbf{g_k}$ and report in Table 2 along with their percentage contribution.

Fig. 2 shows the relationship of power consumed with each control knob. The colored lines in each plot represents the different possible values of the free parameters (e.g. f and PGA in fig. 2(a)). From the trend and relative average power deviation, it can be clearly seen that PGA(**g**) has a very low impact on battery life. Hence, we maintain its default value (i.e. 24x) to keep the signal quality unaltered. As the trend of power consumption is linear with both **f** (validates the PIC32 claim of 0.5mA per MHz power drainage) and **c**, we fit a linear



(a). vs #Channels **(b). vs. frequency** **(c). vs. PG**

Figure 2. Impact of Different Control Knobs on Current Drawn, and hence Power Consumption

curve for power characteristics of OpenBCI,

$$\text{Current (mA)} = 0.4534 \times \mathbf{f} + 1.6615 \times \mathbf{c} + 12.8704 \quad (2)$$

The obtained R^2 statistic and p-values are 0.9994 and 0.0404 respectively for the above fit (eq. 2) which substantiates the goodness of the fit.

In the low-power mode (**f**=6MHz,**c**=1, radio=OFF) operation for 90% of the time [ref - wearable usage 10%], the estimated average current drawn will be 14.78mA, resulting in 10.14 hrs of battery life. This clearly shows that it is possible to achieve $3x$ improvement in the battery life provided the device is in the low-power mode when not actively used.