

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**ZÍSKÁVÁNÍ DAT
Z BCI ZAŘÍZENÍ MUSE**

2017

**LUCIE
PILSOVÁ**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pilsová** Jméno: **Lucie** Osobní číslo: **420555**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Strojírenství**
Studijní obor: **Informační a automatizační technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Získávání dat z BCI zařízení Muse

Název bakalářské práce anglicky:

Data reading from BCI device Muse

Pokyny pro vypracování:

- zajistit přenos dat z Muse na server
- vytvořit jednoduchý server, který bude data ukládat do databáze
- vytvořit webové rozhraní pro čtení dat

Seznam doporučené literatury:

KRAJČA, K., and Jitka MOHYLOVÁ. Číslíkové zpracování neurofyziologických signálů. 1. vyd. Praha: ČVUT, 2011. 168 s. ISBN 978-80-01-04721-7.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Matouš Cejnek, U12110.3


Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

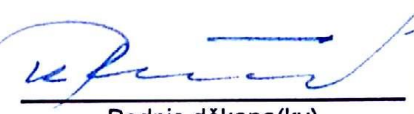
Datum zadání bakalářské práce: **19.04.2017**

Termín odevzdání bakalářské práce: **16.06.2017**

Platnost zadání bakalářské práce:


Podpis vedoucí(ho) práce


Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

19.4.2017

Datum převzetí zadání



Podpis studentky

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího diplomové práce jako jejího spoluautora.

Souhlasím také s případnou publikací výsledků bakalářské práce nebo její podstatné části, pokud budu uvedena jako její spoluautor.

V Praze dne

.....

podpis

Poděkování

Ráda bych tímto poděkovala vedoucímu své bakalářské práce, panu Ing. Matouši Cejnkovi za všestranné vedení, připomínky a cenné rady, které mi velice pomohly při zpracování této práce.

Abstrakt

Cílem teoretické části bakalářské práce je seznámení se s problematikou snímání biologických signálů, zejména elektroencefalografie, na níž potom staví metoda komunikace mozku s počítačem, tzv. brain-computer interfacing. Práce uvádí přehled dostupných BCI pro komerční účely. Praktická část se poté zaměřuje na získání a vykreslení dat ze zařízení Muse Headband vybraného na základě rešerše v teoretické části.

Klíčová slova: brain-computer interface, EEG, Java, JavaScript, Python, server, Muse

Abstract

The main target of the theoretical part of this thesis is an introduction of topic related to sensing biological signals, mainly using electroencephalography, which forms the basis of the brain-computer interfacing. This work shows the overview of commonly available BCIs for commercial purposes. The practical part of this thesis is focused on gaining and visualising EEG data obtained from the Muse Headband device, which has been chosen according to the theoretical part.

Key words: brain-computer interface, EEG, Java, JavaScript, Python, server, Muse

Citace

PILSOVÁ, Lucie. *Získávání dat z BCI zařízení Muse*. Praha, 2017. Bakalářská práce. České vysoké učení technické v Praze, Fakulta strojní. Vedoucí práce Matouš Cejnek.

Seznam tabulek

Tabulka 1: Technické údaje Aurora Dreamband [14].....	7
Tabulka 2: Parametry Melon Headband [16].....	9
Tabulka 3: Specifikace iFocusBand [19].....	10
Tabulka 4: Specifikace Muse Headband [21].....	11
Tabulka 5: Specifikace Star Wars Force Trainer.....	11
Tabulka 6: Specifikace Neurosky MindWave [23].....	12
Tabulka 7: Specifikace zařízení MindFlex.....	12
Tabulka 8: Specifikace zařízení MyndPlay.....	14
Tabulka 9: Specifikace XWave a XWave Sport.....	15
Tabulka 10: Předplatné licence softwaru Pure.EEG.....	16
Tabulka 11: Specifikace Emotiv Insight.....	17
Tabulka 12: Specifikace Emotiv EPOC+.....	18
Tabulka 13: Přehled headsetů k 3D tisku [40].....	20
Tabulka 14: Specifikace Board Kitů OpenBCI [42].....	20

Seznam obrázků

Obr. 1: Schéma montáže EEG elektrod [4].....	3
Obr. 2: Zjednodušené schéma BCI systému [5].....	4
Obr. 3: Ukázka moderního provedení stolu pro hru Mindball [11]	6
Obr. 4: Ukázka použití prototypu pro lucidní snění [14]	7
Obr. 5: Ukázka komunikace aplikace s uživatelem [15].....	8
Obr. 6: Schéma Melon Headband [18].....	9
Obr. 7: Avatar v aplikaci Melon Band[18]	9
Obr. 8: Schema připojení Muse Headband k telefonu [20].....	10
Obr. 9: Příklad použití MyndPlay.	13
Obr. 10: Headset MyndPlay [29]	13
Obr. 11: Schéma první verze zařízení XWave[30]	14
Obr. 12: Schema XWave Sport [31]	15
Obr. 13: Modely EPOC+ a Insight [33]	16
Obr. 14: 3D Vizualizace signálů z mozku[35]	17
Obr. 15: Nosná část OpenBCI k tisku na 3D tiskárně [38]	19
Obr. 16: Doporučená baterie Adafruit nebo Sparkfun 500 mAh s nabíječkou [39]	19
Obr. 17: Schéma umístění elektrod dle mezinárodního standardu 10/20 [44 s.3]	23
Obr. 18: Screenshot prvního úspěšného měření s Muse	24
Obr. 19: Stromová struktura projektu v Android Studio.....	25
Obr. 20: Soubor xml a jeho překlad do grafické podoby	26
Obr. 21: Nové proměnné v MainActivity.java.....	27
Obr. 22: Timer task	28
Obr. 23: Pomocné fronty	29
Obr. 24: Formátování dat k odeslání	29
Obr. 25: Schéma komunikace se serverem	30
Obr. 26: Použití SimpleHTTP Server a SocketServer	31
Obr. 27: Uložení dat ze zásobníku do souboru	32
Obr. 28: Ukládání do souboru .csv	32
Obr. 29: Vytvoření json_data	33
Obr. 30: Nová data do zásobníku	34
Obr. 31: Poslední část kódu v metodě do_POST()	34

Obr. 32: Metoda do_GET()	35
Obr. 33: Ukázka čárového grafu jqPlot [54]	36
Obr. 34: Schéma ajaxování	37
Obr. 35: Zjednodušená ukázka kódu pro ajaxování	38

Seznam použitých zkratek

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
BCI	Brain-computer Interface
CSS	Cascading Style Sheets
CSV	Comma-separated Values
ECG	Electrocardiography
EDR	Enhanced Data Rate
EEG	Electroencefalography
EMG	Electromyography
EOG	Electrooculography
FAQ	Frequently Asked Questions
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LE	Low Energy
LED	Light Emitting Diode
SDK	Software Development Kit
TCP	Transmission Control Protocol
USB	Universal Serial Bus
XML	eXtensible Markup Language

Obsah

1	Úvod.....	1
2	Teoretická část.....	1
2.1	EEG	1
2.1.1	Základní pojmy	1
2.1.2	Elektroencefalografie	1
2.1.3	Získání signálu	2
2.1.4	Pozice elektrod	2
2.1.5	Zesílení	3
2.1.6	Rušení.....	3
2.2	BCI.....	4
2.2.1	Základní princip a rozdělení.....	4
2.2.2	Počátky neinvazivního BCI založeném na EEG	5
2.3	Dostupná BCI na trhu	5
2.3.1	Zábavní BCI	6
2.3.2	BCI s čipy NeuroSky.....	11
2.3.3	Emotiv	15
2.3.4	OpenBCI.....	18
2.3.5	OpenEEG	21
3	Praktická část.....	22
3.1	Muse Headband 2016	22
3.1.1	Technické specifikace	22
3.1.2	Umístění elektrod	23
3.2	Android SDK a jeho nutná modifikace	23
3.2.1	Aplikace TestLibMuse	23
3.2.2	Projekt v AndroidStudio.....	24
3.3	Přijímání, zpracování a ukládání dat pomocí Pythonu	29

3.3.1	SimpleHTTPServer	30
3.3.2	Příjem dat a uložení do souboru	31
3.3.3	Úprava metody do_GET().....	34
3.4	Vykreslování grafů pomocí JavaScript.....	35
3.4.1	jqPlot	35
3.4.2	Ajax	36
4	Závěr.....	39
	Zdroje	41
	Přílohy	46

1 Úvod

Dnešní doba nabízí mnoho možností jak naslouchat svému tělu. V nabídce jsou různé krokoměry, chytré hodinky, trackery a další fitness pomůcky. Mezi ně patří, zatím stále netradiční, zařízení pro měření mozkové aktivity. S těmito elektroencefalografy jsme se donedávna mohli setkat pouze v nemocnicích a výzkumných zařízeních. Nyní se na trhu objevují i takové výrobky, které mohou být užitečné v komerční sféře. Na zmapování trhu jsem se zaměřila v teoretické části.

Pro účely praktické stránky bakalářské práce bylo vybráno zařízení Muse. Cílem byla realizace přenosu dat a jejich zobrazení v uživatelském rozhraní. Původním záměrem bylo hlouběji se zaměřit na samotnou analýzu dat, případnou aplikaci transformací, hodnocení a verifikace měřených dat pomocí vybraného zařízení. Po zakoupení BCI ale nastala hlavní komplikace a tou byla absence knihoven pro jazyk Python. Tato práce tedy řeší problém samotného přenosu dat a zároveň také i jejich vykreslení. Text praktické části tak nabízí průřez celého procesu, ukazuje jak grafický výstup, tak i funkce, které by za normálních okolností zůstaly skryté v použitých knihovnách.

2 Teoretická část

2.1 EEG

2.1.1 Základní pojmy

Signál – Lze si jej představit jako nosič informace nebo jako fyzikální kvantum vykazující určité změny v čase, prostoru apod. Matematicky může být signál popsán funkcí n -proměnných. V případě této práce jsou sledovány tzv. biologické signály. Jedná se o signály, které se šíří z živého organismu a jsou následně zařízením snímány a dále zpracovávány.

Zpracování signálu – Jedná se o získávání určitých informací, které jsou přenášeny signálem. Tyto informace je třeba vizuálně, matematicky nebo statisticky rozdělit na dílčí signály, které vypovídají o různých fyziologických pochodech v organismu. Je třeba brát v úvahu náhodné jevy, šum, ale i periodicitu typickou pro děje ve sledovaném subjektu [1, s. 8].

2.1.2 Elektroencefalografie

Elektroencefalografie je metoda určená ke sledování neurofyziologické aktivity mozku, zejména v jeho povrchových strukturách. Nejčastěji se této metody užívá v lékařství pro diagnostiku epilepsie, spánkových poruch a dalších onemocnění souvisejícími s fyzickým,

ale i psychickým stavem pacienta. Základním nástrojem pro měření jsou elektrody umístěné podle přesných schémat na povrchu lebky. Snímanou veličinou je pak napětí (řády μV), které se mění s časem. Napětí nemá stálou frekvenci ani amplitudu, můžeme za určitých okolností předpokládat jen částečnou stacionaritu v konkrétních úsecích. [2, s. 219]

Elektrické potenciály mozku poprvé zaznamenal německý vědec Hans Berger. Jako při mnoha zásadních objevech v naší historii byly v roce 1929 Bergerovy závěry přijaty vědeckou obcí s velkou dávkou nedůvěry a skepticismu. Berger sám se pohyboval na hraně vědy a určitého okultismu (věřil například ve spojení mysli mezi sourozenci, což nebylo vědecky nikdy prokázáno). Jeho výzkum dále převzal tým z univerzity v Cambridge v čele s Lordem Edgarem Adrianem, který v roce 1934 potvrdil Bergerovy teorie a dále je také rozvíjel. [3]

2.1.3 Získání signálu

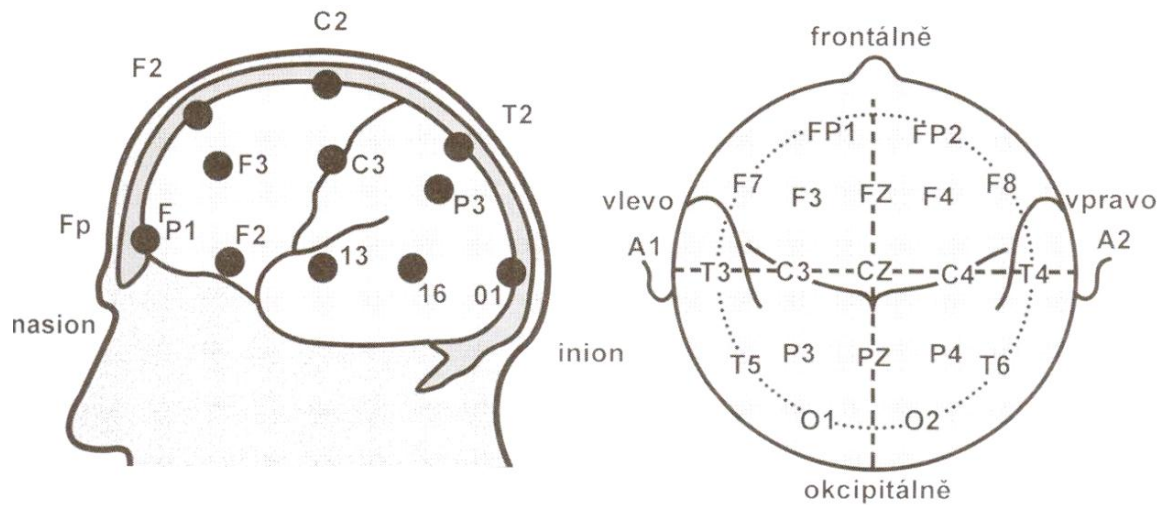
Mozková aktivita je snímána přístrojem zvaným elektroencefalograf, který na výstupu vykresluje výchylky na papír nebo do počítače. Snímací elektrody se umísťují buď na povrch hlavy (tzv. neinvazivní metoda, nejčastější) nebo přímo na povrch mozku. Aby mohl být signál vyhodnocován, musí projít řadou úprav. V 50. letech se vědci začali zabývat matematickým popisem signálu a to nejprve pomocí Fourierových transformací. Získaný signál je ve skutečnosti neperiodický a pomocí různých aproximačních modelů je třeba jej popsat. S nástupem využití výpočetní techniky je práce s EEG mnohonásobně jednodušší. EEG můžeme zpracovat podle tří hledisek, která s sebou samozřejmě nesou určité výhody, ale i nevýhody. Analyzujeme buď na základě amplitud, kdy ale neuvažujeme vlnovou délku, ani lokální periodičnost, nebo naopak analyzujeme periodu (vln. délku) na daných úsecích. Třetím způsobem je rozbor frekvence a oddělování jejích složek od sebe. Ty se pak analyzují nezávisle na sobě. [1, s. 7]

2.1.4 Pozice elektrod

Jako každé měření podléhá i elektroencefalografie mnoha standardům. Za zmínku stojí mezinárodní systém 10-20. V roce 1958 jej vytvořil Herbert H. Jasper. Tato norma uvádí, jak správně umísťovat elektrody na hlavu subjektu, a pojmenovává jednotlivé pozice, které ukazuje Obr. 1. V každé pomyslné linii jsou elektrody rozloženy v 10, 20, 20, 20, 20 a 10 % její celkové délky.

Každá elektroda nese písmenné a číslicové označení: Fp – frontopolární, F – frontální, P – parietální, T – temporální, O – okcipitální a C – centrální. Lichá čísla značí levou, sudá čísla

pravou hemisféru. Rozlišuje se 8 párových elektrod a 3 nepárové, tzv. vertexové, ty se značí písmenem z.



Obr. 1: Schéma montáže EEG elektrod [4]

Existují také měřicí systémy využívající hustěji osazených elektrod, a proto se zavádějí i jiná rozlišení například 10-5 [4].

2.1.5 Zesílení

Data jsou nyní zpracovávána výhradně pomocí počítače. Záznam na papír nemá z ekologických ani ekonomických náhledů žádný význam. Pro převedení spojitého signálu je použit analogově-digitální převodník (nejčastěji v diferenčním zapojení), který převede úseky na posloupnost vzorků v pravidelných časových intervalech. Dochází k tzv. vzorkování signálu, kde každý vzorek je shromážděn v jedné z 2^n úrovní (n je rovno počtu bitů A/D převodníku). [1, s. 17]

2.1.6 Rušení

Pro správnou analýzu je nutné eliminovat veškeré možné zdroje zkreslujícího signálu. Nežádoucí vlivy nazýváme tzv. artefakty. Ty mohou být dvojího původu:

- a) Fyzikální (technické) – rušení spotřebiči zapojenými do sítě (tzv. síťový brum), šum elektronických spotřebičů apod.
- b) Biologické – tep, elektrické signály při stahování svalů a další [1, s. 14]

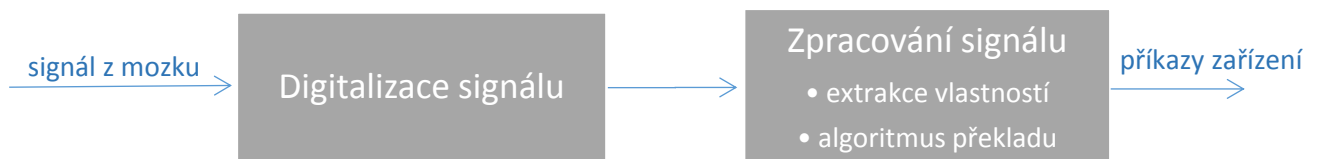
Artefakty lze eliminovat už samotným umístěním přístroje v budově (odrušená laboratoř), pozicí elektrod a později odstranit v počítači (například odstranění rušení od synapsí srdce odečtením elektrokardiogramu, který se nahrával souběžně při měření EEG).

2.2 BCI

2.2.1 Základní princip a rozdělení

BCI, z anglického brain-computer interface (v překladu rozhraní mozek-počítač nebo rozhraní mozek-stroj), je technologie, která za pomoci EEG snímače dále zpracovává elektromagnetické signály z mozku. BCI tedy pracuje se záznamem mozkové aktivity a dále je z něj schopné předat informace a příkazy do koncového zařízení.

BCI systém umožňuje převod mozkové aktivity na digitální informaci, se kterou lze následně pomocí počítače pracovat. U neinvazivního BCI se využívá EEG přístroje, který je celý umístěn na povrchu lebky. Tato skutečnost může vyvolat značné nepřesnosti, avšak zápory jsou vyváženy faktem, že při výzkumu i praktickém použití není třeba provádět žádné chirurgické zásahy do těla uživatele. Dále je třeba vytvořit vhodný algoritmus, který převede EEG na data vhodná k řízení cílového objektu a bude správně pracovat i při použití různými uživateli [5]. Takové algoritmy jsou velmi výpočetně náročné, vycházejí z principu neuronových sítí a využívají Markovova rozhodovacího procesu [6].



Obr. 2: Zjednodušené schéma BCI systému [5]

V BCI dochází k:

- snímání dat – to je realizováno pomocí EEG zařízení, které měří napěťové potenciály neuronů v mozku.
- oddělení nežádoucích artefaktů a filtraci – vhodným zapojením, případně odečtem známých rušivých signálů dochází k vyčištění signálu. Je také nutné uvažovat tlumení vlivem průchodu signálu přes lebku a kůži.
- parametrizaci – předzpracovaný signál je rozdělen na parametry, které pak slouží k provedení další analýzy pro uskutečnění práce s koncovým zařízením.
- klasifikaci – klasifikátor dle získaných parametrů rozpozná jejich účel a generuje další příkazy.

- vykonání příkazů – příkazy z klasifikátoru jsou pak vykonány tzv. akčním členem (ovládání externího zařízení – kurzoru myši, klávesnice, robotické protézy končetiny apod.)

Zařazena musí být také zpětná vazba, pomocí níž lze v reálném čase sledovat reakce na aktuální mozkovou aktivitu [7].

2.2.2 Počátky neinvazivního BCI založeném na EEG

Jak již bylo zmíněno, neinvazivní BCI v sobě skrývají velký potenciál k využití i pro komerční účely. Zcela reálným se tak může stát ovládání nejrůznějších zařízení pouhou myšlenkou. Jako první se snažili o realizaci takového zařízení vědci Farwell a Donchin v roce 1988, i když zatím jen jako prostředek pro komunikaci s pacienty, kteří byli trvale upoutáni na lůžko a nevykazovali žádnou fyzickou aktivitu, pomocí níž by mohli komunikovat [8]. Farewell a Donchin se zaměřili na tzv. vlnu P300, která v EEG signalizuje náhlé rozhodnutí [9].

Další výzkumné týmy v rámci univerzit po celém světě se neustále snaží vyvíjet nová BCI. Spojen s konkurenčním týmem z Tübingenu navrhl v roce 2003 Jonathan Wolpaw z Wadsworthova centra pro zdraví tzv. BCI2000, což je zařízení umožňující svou stavbou implementaci dalších vylepšení. To se stalo zásadním krokem pro další vývoj, protože bylo možné kombinovat různé signály, ale i způsoby jejich zpracování, zapojit výstupní zařízení a to vše použít ve spojení s různými počítačovými operačními systémy [10].

2.3 Dostupná BCI na trhu

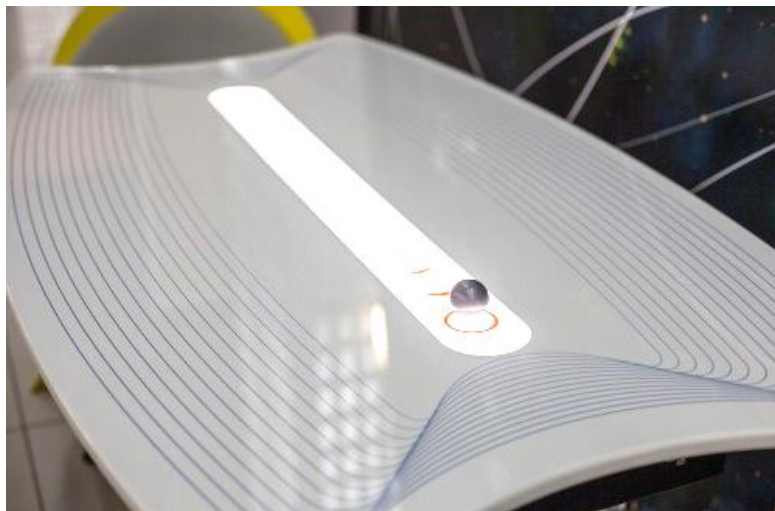
BCI pro komerční účely jsou k dostání v mnoha podobách. Lze je ale rozdělit do dvou skupin, a to na jednoúčelová a univerzální. Do první skupiny můžeme zařadit zařízení, která jsou primárně určena k tréninku vlastní mozkové aktivity a soustředění. Takováto BCI fungují pouze v návaznosti na výrobcem dodávanou aplikaci, pomocí níž má pak uživatel možnost sledovat svou mozkovou aktivitu. V herním průmyslu je možné se také setkat s BCI, která jsou určená pro hraní her jako je například nejstarší Mindball nebo Star Wars Force Trainer. Tato rozhraní ale nelze účinně použít pro výzkum.

Pro další zkoumání a účely této bakalářské práce je nutné hledat v druhé skupině BCI, která svou stavbou umožňují relevantní snímání EEG signálu a také poskytují data ve formátu použitelném k pozdějšímu zpracování.

2.3.1 Zábavní BCI

Mindball

Průkopníkem v oblasti komerčních BCI se stala Švédská společnost Interactive Productline. V roce 2003 byl vyvinut první prototyp Brainball (později Mindball). Hra je založena na jednoduchém principu, ale vyžaduje maximální pozornost hráčů. Ti se snaží soustředit na pohyb míčku a přetlačují jeden druhého. Uživatelé mají na hlavě připevněnu čelenku s elektrodami umístěnými na čelní straně hlavy, data z elektrod jsou zpracována počítačem, který ovládá pohyb míčku na stole (Obr. 3).



Obr. 3: Ukázka moderního provedení stolu pro hru Mindball [11]

Nyní je k dostání verze hry pro děti i pro dospělé, zapojit se může i více hráčů. V nabídce je i trenažér nebo počítačová hra bez fyzicky přítomného stolu. Mozkovou aktivitu lze v reálném čase sledovat na počítači a ukládat pro pozdější porovnání [12].

Aurora Dreamband

Aurora Dream je zařízení v podobě textilní čelenky (Obr. 4) s elektrodami umístěnými na čele. Aurora Dream bylo vyvinuto skupinou iwinks, a umožňuje snímání spánku a vyhodnocování jeho kvality. Uživatel má možnost sledovat průběh svého spánku, nastavit si šetrný budík a také relaxovat za svitu LED o specifických vlnových délkách, které napomáhají dosažení stavu lucidního snění¹.

¹ „Lucidní snění je zvláštní způsob prožívání snu, ve kterém jste si plně vědomi toho, že právě sníte. Přestože se v danou chvíli nacházíte uprostřed svého snu, zcela jasně si uvědomujete, že Vaše fyzické tělo ve skutečnosti leží v posteli a odpočívá [13].“



Obr. 4: Ukázka použití prototypu pro lucidní snění [14]

Projekt byl spuštěn 12. prosince 2013 a byl financován příspěvkem ze serveru kickstarter.com. Od té doby byla iWinks schopna přejít od prototypu na sériovou výrobu, vyvinout rozhraní pro počítače (Borealis) i mobilní telefony (iWinks App). Specifikace shrnuje následující tabulka.

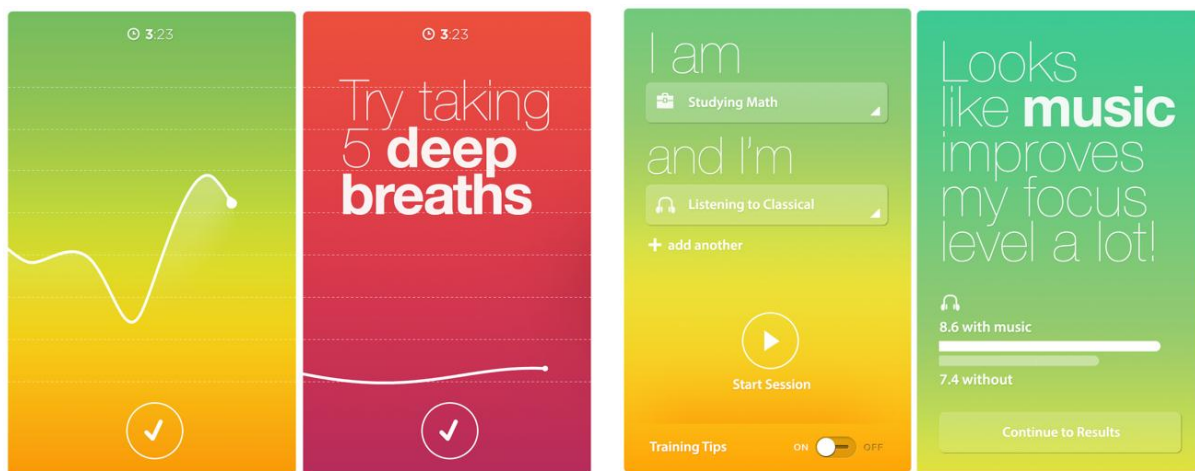
Tabulka 1: Technické údaje Aurora Dreamband [14]

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) Pohyb očí (EOG) Svalové stahy (EMG) Tep (ECG)	Třiosý akcelerometr Třiosý gyroskop	USB	USB Bluetooth	1	Zdarma, data z biosenzoru, snímače polohy a pohybu, okolní teplota, ovládání LED	\$299

Melon Headband

Melon Headband, jak už sám název napovídá, je dalším výrobkem na trhu v podobě čelenky, ve které jsou umístěny čtyři elektrody, které snímají mozkovou aktivitu na čele uživatele. Oficiálně první výrobek byl dokončen a odeslán k zákazníkovi 19. července 2014. Projekt byl opět financován přes server kickstarter.com, kdy první žádost o příspěvky skupina Melon uveřejnila 14. května 2013.

Melon Headband slouží k mentálnímu tréninku uživatele, zlepšuje jeho soustředění a zvyšuje jeho studijní či pracovní výkony. K zařízením je třeba mít nainstalovanou aplikaci (pro iOS, nebo Android), která nabízí mnoho možností, jak analyzovat vlastní mozek.



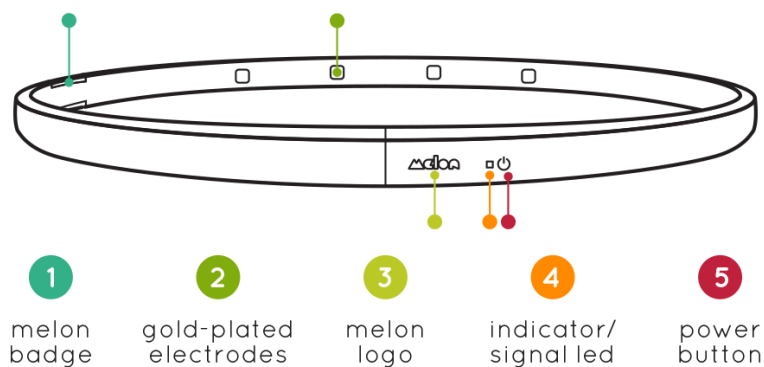
Obr. 5: Ukázka komunikace aplikace s uživatelem [15]

Aplikace nejen že snímá aktivitu mozku v době, kdy máte nasazenou čelenku, ale získaná data také analyzuje za pomoci vstupních informací, které v průběhu dne uživatel zapisuje. Pokud je uživatel v práci a poslouchá určitý typ hudby, zaznamená tento fakt a aplikace sama vyhodnotí, zda uvedený žánr hudby byl pro zákazníka vhodný při plnění úkolů. Dále si lze pomocí aplikace trénovat soustředění hraním logických her, které se také soustředí na zlepšení koncentrace a trpělivosti spotřebitele. Aplikace během této činnosti doporučí i vhodné druhy relaxace, jako například: „Zkuste se nyní pětkrát zhluboka nadechnout.“ a ihned zobrazí křivky značící míru soustředění před a po provedení nádechu (Obr. 5).

Na trh bylo uvedeno několik desítek kusů tohoto výrobku, ale nyní nelze přes oficiální stránky Melon žádný další koupit. Dochází údajně k optimalizaci a testování. Firma se nebrání ani spolupráci s externími vývojáři a nabízí vývojářský balíček, který zájemce po dohodě obdrží [16].

Podle [16] používají zesilovací i filtrovací protokoly od korporace NeuroSky, stejně tak jako vestavěný čip, což je v [17] uvedeno chybně.

Melon se dle svých FAQ nebrání ani spolupráci na akademické úrovni, flexibilně zodpovídají dotazy a uveřejňují detaily pro uživatele.



Obr. 6: Schéma Melon Headband [18]

Tabulka 2: Parametry Melon Headband [16]

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) algoritmus pro měření soustředění	Tříosý akcelerometr (bude zabudován v nové verzi)	USB	micro USB, bluetooth 4.0	4	Dostupné, návod od výrobce „How to hack Melon“	\$149

iFocus Band

Na rozdíl od Melon Band používá čelenka iFocus Band k tréninku soustředění vizualizaci v podobě avatara (Obr. 7). Zařízení bylo vyvinuto v roce 2009 Australským inženýry Henrym a Grahamem Boultonovými. Jejich hlavním cílem je zvýšit výkon a soustředění sportovců. Své poznatky testovali v roce 2013 na americké golfové PGA Tour, kde zaznamenali úspěchy ve spolupráci s golfistou Jasonem Dayem [19].



Obr. 7: Avatar v aplikaci Melon Band[18]

iFocus Band působí svým vzhledem poněkud nemoderně, i design aplikace nepůsobí, že by sledovala nejnovější trendy. Oproti Melon Band je iFocus ale již delší dobu na trhu

a může se pochlubit prokázanými výsledky zlepšení výkonnosti sportovců v mnoha odvětvích. Nabízí také konkrétní podobu SDK a přístup k surovým EEG datům.

Tabulka 3: Specifikace iFocusBand [19]

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) algoritmus pro měření soustředění	-	USB	USB, bluetooth	3	Dostupné, možnost zpracování „raw EEG“, Win, iOS	\$500

Muse

V roce 2014 společnost InteraXon představila svůj první výrobek z oblasti BCI pod názvem Muse. Dle oficiálních propagačních materiálů se jedná o fitness pomůcku, která má snadnou cestou dopomoci člověku relaxovat. Dalším přínosem je s tím spojené snížení stresu a zvýšení výkonu a pozornosti při každodenních činnostech. K Muse Headband patří samozřejmě mobilní aplikace (Obr. 8), která obsahuje i tutoriál. Samotné používání je deklarováno jako zcela intuitivní, je jen třeba dodržet následující body:

- Připojení k aplikaci – Dochází k němu přes Bluetooth, jakmile je zařízení spárováno, Muse se při zapnutí snaží automaticky spojit s posledním zařízením ve své paměti.
- Tutoriál a kalibrace – Aplikace provede uživatele správným nasazením a umístěním elektrod na hlavě. Před každou „session“ (sezením) musí aplikace pořídit tzv. „brain snapshot“ (mozkový snímek), který je využíván jako výchozí stav a od něhož se odvíjejí další měření.
- Sezení – Samotné sezení je vlastně meditací. Uživatel se má snažit zklidnit svou mysl a medítovat. Aplikace dává zpětnou vazbu zvukovou stopou větru. Pokud je slyšet silný vítr, je nutné se více soustředit na zklidnění mysli.



Obr. 8: Schema připojení Muse Headband k telefonu [20]

Aplikace pak nabízí dlouhodobé vedení statistiky výsledků a zlepšení. Takto se může zařízení zdát velmi jednoúčelové, ale InteraXon je velmi vstřícný také k vývojářům. Nabízí k produktu tři softwarové nástroje, a to MuseIO pro živý stream dat z Muse, MuseLab pro vizualizaci, nahrávání a filtrování dat a MusePlayer, který je vhodný k převodu dat z formátu .muse do univerzálního tabulkového formátu .csv nebo pro práci s MATLAB. Surová EEG data jsou k dispozici v jednotlivých vrstvách alfa až gama, přičemž MuseLab umožňuje filtraci artefaktů a používá ke zpracování FFT (Fast Fourier Transform).

Tabulka 4: Specifikace Muse Headband [21]

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) Srdeční aktivita (ECG)	3-D akcelerometr	USB	USB, bluetooth 4.0 LE	5	Dostupné, možnost „raw EEG, ECG, accelerometer“ Přehledné SDK	\$249

2.3.2 BCI s čipy NeuroSky

Společnost NeuroSky se sídlem v Silicon Valley byla založena v roce 2004 a zaměřuje se na výrobu čipů pro snímání signálů EEG a ECG. NeuroSky podporuje nově vznikající projekty v oblasti konstrukce tzv. wearable devices (zařízení vhodná k nošení) poskytováním svých biosenzorů a zpracovávacích algoritmů.

Star Wars Force Trainer

Výrobek patří do kategorie hraček, ale je nezbytné jej uvést, protože byl na trhu jedním z prvních, který využíval technologie zpracování EEG signálu pro komerční účely. Force trainer ve své první podobě (vydán společností Uncle Milton na podzim 2009) sloužil pro zvedání míčku pomocí mysli. V praxi to znamenalo, že čip v headsetu vyhodnotil EEG signál a odesílal příkazy ventilátoru, který zdvihal míček a ten následně „levitoval“ [22].

Druhá a nyní stále dostupná verze Star Wars Force Trainer II funguje na stejném principu, ale umožňuje hráči pohybovat holografickou simulací promítaného předmětu [23].

Tabulka 5: Specifikace Star Wars Force Trainer

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) beta vlny	-	AAA baterie	bluetooth	Verze I: 1 Verze II: 3	nedostupné	\$109.99

MindSet a MindWave

Společnost NeuroSky také prodává své vlastní výrobky v podobě headsetu. Ten se skládá z pohyblivého ramena, které si uživatel dle své fyziologie umístí na čelo nad své levé oko, a z klipu, který se připíná na ušní lalůček pro sledování tepové frekvence [24].

Od roku 2007, kdy přišel na svět první prototyp MindSet, bylo představeno několik produktových řad, přičemž současná nese název MindWave. Existují varianty pro použití s počítačem, ale i mobilem či tabletem. NeuroSky nabízí mnoho aplikací pro děti i dospělé, nejde tedy jen o hraní her, ale je možné provádět i analýzu svého soustředění nebo stresu. Na rozdíl od Star Wars Force Trainer rozeznává Mind Wave signály alfa a beta (zdali i gama a delta není upřesněno). Data lze extrahovat pro vlastní zpracování zakoupením balíčku NeuroView, který spolupracuje s MATLAB, a to v ceně \$499.99 [25].

Důležitý je fakt, že algoritmus na potlačení šumu je nastaven na síťovou frekvenci 60 Hz (USA), a proto by v našich zeměpisných šířkách zařízení nefungovalo správně (frekvence v síti 50 Hz) [23].

Tabulka 6: Specifikace Neurosky MindWave [23]

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) Srdeční aktivita (ECG)	-	AAA baterie	Bluetooth 2.1	1	Ano, za poplatek „raw EEG .csv “	\$99.99 (bez aplikací)

MindFlex

Na podzim 2009 vydala společnost Mattel herní zařízení Mind Flex skládající se z čelenky s elektrodami snímajícími EEG a herní konzole. Na základě míry soustředění hráče se pak roztočí na hrací ploše ventilátor, jenž zvedá do vzduchu pěnový míček, který má za úkol překonat všechny překážky. Jedná se o jednoúčelové zařízení, které je ale dalším krokem k přivedení technologie BCI mezi širší veřejnost [26].

Tabulka 7: Specifikace zařízení MindFlex

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) Srdeční aktivita (ECG)	-	3 x AAA baterie	-	1	nedostupné	\$79.99

MyndPlay

MyndPlay BrainBand je dalším zařízením (na trhu od roku 2011) pracujícím s čipy NeuroSky, a snaží se velmi netradiční formou zaujmout celé spektrum uživatelů. MyndPlay je zároveň speciálně upravený videopřehrávač, který umožňuje diváku v reálném čase ovlivňovat děj filmu. Při přehrávání videa se v určitých pasážích objeví upozornění a ukazatel soustředění. Je pak na uživateli, aby udržel svou míru koncentrace a ovlivnil děj. Například pokud se bude divák dostatečně soustředit, a ukazatel zůstane v daných mezích, hlavní hrdina vyhraje právě probíhající souboj [27].

Praktické použití v reklamě - pokud se zákazník dostatečně soustředí, plechovka se otevře. Pokud soustředění dosáhne horního limitu, plechovka exploduje (Obr. 9). V úvahu připadá i neúspěch, a plechovka zůstane zavřená [28].



Obr. 9: Příklad použití MyndPlay.

K samotnému headsetu je dostupná aplikace pro vytvoření interaktivního videa, a také balíček nástrojů pro výzkum. Tento balíček lze zakoupit i u mateřské společnosti NeuroSky.



Obr. 10: Headset MyndPlay [29]

Tabulka 8: Specifikace zařízení MyndPlay

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) Srdeční aktivita (ECG)	-	Věštavěný akumulátor, dobíjení micro USB	Bluetooth	1	Ano, „raw EEG“ při koupi Research Toolkit	\$249

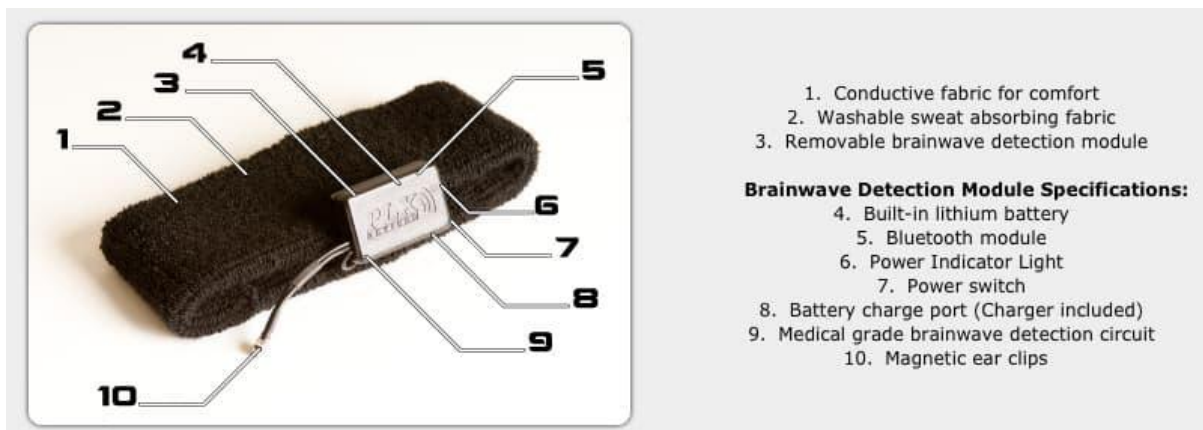
XWave

XWave, zařízení od společnosti PLX, je nápadně podobné MindWave od NeuroSky (Obr. 11). XWave také vychází z technologie NeuroSky a první verze byla uvedena na trh v roce 2010. Je to další z řady wearable devices, které zákazníkovi umožňuje aktivně se podílet na tréninku a sledování svého soustředění.



Obr. 11: Schéma první verze zařízení XWave[30]

Tento model byl o dva roky později následován modernější variantou, která se zcela zaměřila na trh s chytrými doplňky pro sportovce. Čelenka ze snadno omyvatelného materiálu byla schopna odolat agresivním složkám potu a zároveň svým designem nepůsobila nápadně jako zatím všechna dostupná zařízení (Obr. 12).



Obr. 12: Schema XWave Sport [31]

Na oficiální stránkách PLX není dnes žádná zmínka o XWave, poslední kusy nabízí už jen obchodníci s elektronikou na serveru e-bay. Podle článku [31] nabízí PLX prostředí pro vývojáře a kompletní přístup k datům ze senzorů. Více informací nebylo uvedeno.

Tabulka 9: Specifikace XWave a XWave Sport

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG) Srdeční aktivita (ECG)	-	AAA baterie	Bluetooth 2.1	1	pouze pro iOS „raw EEG .csv “	\$100
Mozková aktivita (EEG)	-	USB (Li-ion baterie)	Bluetooth	1	Dostupné (bližší informace neznámé)	\$99

2.3.3 Emotiv

Společnost Emotiv byla založena v roce 2011 a nabízí dvě varianty BCI. Starší z dvojice Emotiv EPOC (nové verze nazvány EPOC+) je určena především pro akademické účely, naopak Emotiv Insight nabízí běžným uživatelům možnost nahlédnout do chování svého mozku a procvičit své soustředění (Obr. 13).



Obr. 13: Modely EPOC+ a Insight [33]

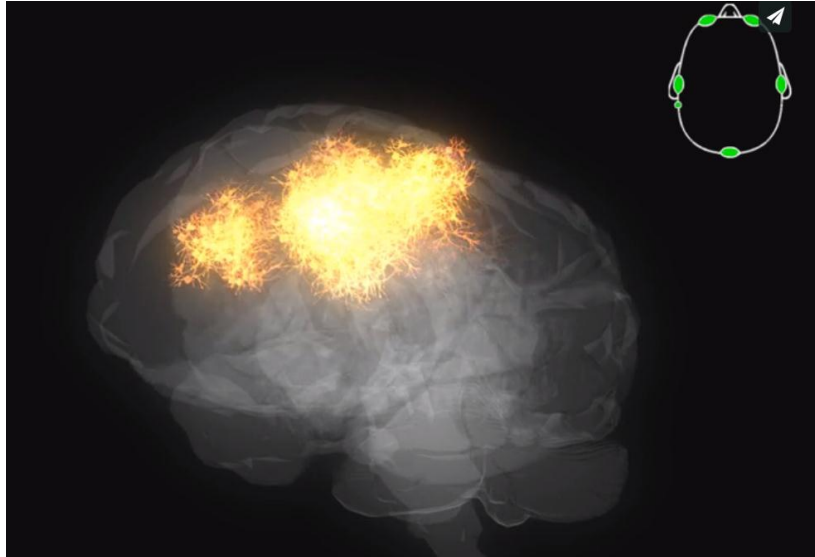
V sekci software nabídky Emotiv jsou dostupné zdarma hry a aplikace pro základní trénink koncentrace a relaxace. Má-li zákazník zájem o prostředí Pure.EEG pro práci s EEG signálem (případně údaji z gyroskopu a akcelerometru), musí zaplatit poplatek. Výši ukazuje následující tabulka platná k 21.5.2016 [33]:

Tabulka 10: Předplatné licence softwaru Pure.EEG

Záznamů za měsíc	50	100	250	251+
Měsíční předplatné	\$49	\$75	\$149	dohodou
Roční předplatné	\$499	\$749	\$1499	dohodou

Emotiv Insight

Emotiv Insight v kategorii headsetů (v podobě čelenky) disponuje jednoznačně největším počtem elektrod. Má také dva referenční senzory. EEG signál je oproti zařízením s jednou elektrodou měřen daleko přesněji. Jednou z aplikací dostupných pro PC je také 3D Brain Visualizer umožňující modelovat vzruchy odehrávající se v mozku a interpretovat je na obecném 3D modelu mozku (Obr. 14).



Obr. 14: 3D Vizualizace signálů z mozku[35]

Emotiv Insight je schopen detekovat různé výrazy tváře:

- mrknutí
- zavřené levé/pravé víčko
- zamračení
- zvednuté obočí (překvapení)
- úsměv
- zaťaté zuby

a také emoční stavy:

- okamžité vzrušení, překvapení
- dlouhotrvající vzrušení, překvapení
- stres
- uvolnění, relaxace

Tabulka 11: Specifikace Emotiv Insight

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG)	3D akcelerometr, 3D gyroskop, 3D magnetometr	Li-Pol 480mAh	Bluetooth SMART 4.0 LE	5 + 2 referenční	Ano	\$299

Emotiv EPOC+

Druhým zařízením od společnosti EMOTIV je neuroheadset EPOC+, který nabízí nadstandardní možnosti analýzy EEG signálu. Výrobce sám doporučuje EPOC+ pro vědecké účely. Přístup k surovým EEG datům je umožněn pouze přes předplacené rozhraní Pure.EEG.

Oproti Insight, rozpoznává EPOC navíc pohled očí vlevo a vpravo, smích nebo dokonce pousmání se zvednutím koutku vlevo či vpravo [33].

V roce 2010 vytvořil programátor Cody Brocious v rámci projektu Cody's Emokit open-source knihovnu pro čtení dat přímo ze zařízení EPOC. Funkčnost kódu ale již není potvrzena s nástupem modelu EPOC+ [36].

Tabulka 12: Specifikace Emotiv EPOC+

Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
Mozková aktivita (EEG), Svalová aktivita (EMG)	3D akcelerometr, 3D gyroskop, 3D magnetometr	Li-Pol 680mAh	Bluetooth SMART 4.0 LE	14 + 2 referenční	Ano	\$799

2.3.4 OpenBCI

Open BCI je projekt byl založen v roce 2011 profesorem Joelem Murphym, který působil na fakultě MFA designu a technologie na The New School of Parsons. Zde se na praktické hodině setkal s nadšeným studentem Conorem Russomanem a spolu začali vytvářet vlastní BCI. Nejprve začali se zařízením Mindflex a jeho rozkódování a převedení do rozhraní v jazyce Python. Vydali se tak na cestu poznávání dostupných BCI zařízení, která však často nespĺňovala jejich požadavky. Jak je patrné z výčtu zařízení výše, má většina z nich nedostatečný počet a fixní polohu elektrod, software i hardware je velice drahý a ne zcela přátelský k vývojářům. Dvojice se tak rozhodla vytvořit univerzální BCI dostupné univerzitám i středním školám. Parsons a Russomano věří, že úspěchu a pokroku je možné dosáhnout jedině sdílením informací a rozšířením technologie mezi drobnější výzkumná zařízení než jen držet know-how v uzavřeném vědeckém týmu [37].

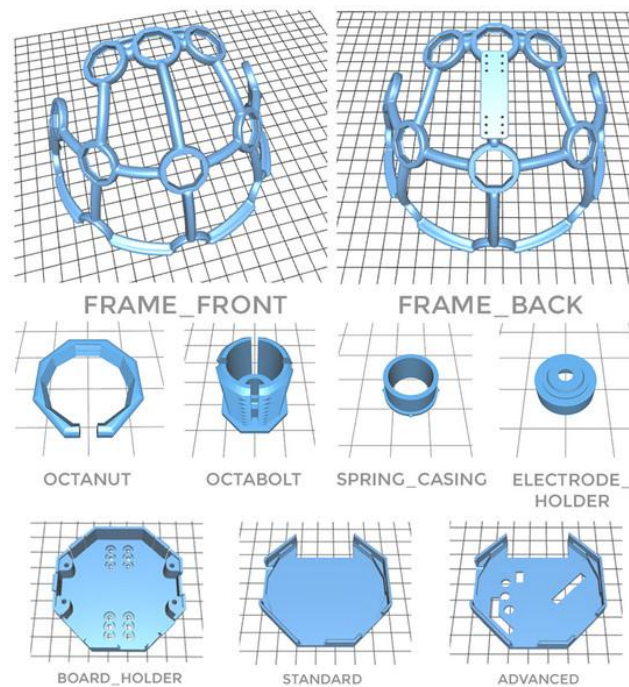
Headset

Open BCI se skládá ze dvou částí:

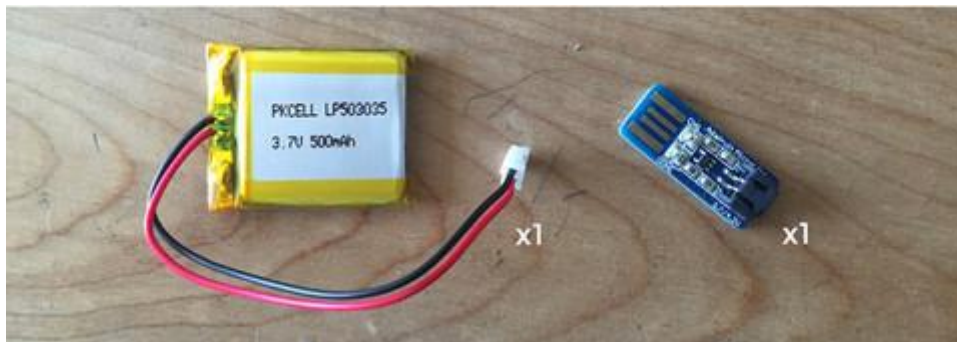
- headsetu – obsahuje otvory pro umístění elektrod a připojení základní desky
- základní desky – její výběr se řídí počtem požadovaných elektrod

a ty jsou doplněny kabely, šrouby a dalším příslušenstvím potřebným k samotné montáži. Headset je vyráběn podle plánů na 3D tiskárně, lze jej zakoupit buď v e-shopu OpenBCI a nebo si zakoupit plány a vytisknout si headset samostatně (Obr. 15). V nabídce je také balíček

příslušenství, které lze samozřejmě pořídit v obchodě s elektrotechnikou. Balíček „Print It Yourself“ je určen pro zákazníky, kterým se nechtějí součástky po jedné shánět. Součástí není Li-Ion baterie, kterou je nutné pořídit u externích výrobců zvlášť (Obr. 16).



Obr. 15: Nosná část OpenBCI k tisku na 3D tiskárně [38]



Obr. 16: Doporučená baterie Adafruit nebo Sparkfun 500 mAh s nabíječkou [39]

Headset prošel několika vývojovými stadii. První verzí byl tzv SpiderClaw 3000, který se ale neosvědčil při použití na různých velikostech lebky uživatelů. Nejaktuálnějším modelem je tak Ultracortex Mark III „Nova“ a „Supernova“ pro použití se suchými elektrodami. Ultracortex Mark III se liší pouze počtem pozic pro umístění elektrod. Na serveru kickstarter.com je nyní možné přispět a tím si i zařídít předobjednání nejnovějšího modelu Ultracortex Mark IV, který předplatitelé obdrží na podzim 2016 v předstihu před zveřejněním plánů na GitHub [40].

Tabulka 13: Přehled headsetů k 3D tisku [40]

Headset	Slotů pro elektrody	Cena za plány	Cena za tisk	Cena za tisk a sestavení
Ultracortex Mark III „Nova“	21	Dostupné soubory.stl na GitHub bez baterie a součástek	\$499.99 bez baterie	\$799.99 bez baterie
Ultracortex Mark III „Supernova“	29	Dostupné soubory.stl na GitHub bez baterie a součástek	\$699.99 bez baterie	\$1099.99 bez baterie
Ultracortex Mark IV předobjednání	61	\$299.99 bez baterie a součástek	\$399.99 bez baterie	\$599.99 bez baterie

Základní desky

Podle počtu elektrod rozlišujeme 3 typy základních desek, které se připevňují k plastovému headsetu v zadní části hlavy. Novinkou je cenově dostupnější Ganglion, jehož cílovou skupinou by měly být převážně školní experimentální zařízení. Ganglion bude vydán spolu s Ultracortex Mark IV na podzim 2016 [41].

Tabulka 14: Specifikace Board Kitů OpenBCI [42]

Název	Biosenzory	Snímače pohybu	Napájení	Připojení	Počet elektrod	SDK	Cena
32bit Board Kit (8-channel)	EEG EMG EKG	3D akcelerometr	2 x Li-Ion 580mAh	Bluetooth 4.0 LE	8 (suché)	Ano	\$499.99
R&D Kit (16-channel) Daisy	EEG EMG EKG	3D akcelerometr	2 x Li-Ion 580mAh	Bluetooth 4.0 LE	16 (suché)	Ano	\$899.99
Ganglion Board Kit (4-channel)	EEG EMG EKG	3D akcelerometr	?	Bluetooth 4.0 LE	4 (suché)	Ano	\$99.99

2.3.5 OpenEEG

OpenEEG je projekt, který nabízí návody a odkazy pro domácí výrobu BCI. OpenEEG chápe, že zařízení dostupná na trhu mohou být zcela mimo finanční možnosti nadšenců v oblasti neurofeedbacku. Základním zařízením, které si uživatel sám vytvoří je ModularEEG, které lze v závislosti na ceně součástek vyrobit za cenu \$200 - \$400. ModularEEG se skládá ze dvou (rozšířením až na 16 se zabýval německy student Sebastian Holtermann) nebo více EEG zesilovačů a základní desky (uživatel si vybere ze seznamu doporučených externích prodejců), která komunikuje s počítačem přes standardní sériový kabel [42].

Samotný rozdíl je popsán autory konkurenčního OpenBCI na [43]: „Řečí techniky, OpenBCI má 24 bitový ADC (analogově-digitální převodník) oproti OpenEEG, které má převodník 10 bitový. To je velký rozdíl v dynamickém rozsahu, který umožňuje OpenBCI měřit malé signály (EEG), větší signály (ECG na zápěstí) a ještě silnější signály (ECG na hrudi, EOG nebo EMG) bez ořezu a dalších ztrát. Při měření s OpenEEG se může stát, že při umístění elektrody na zápěstí změříte nezpracovatelně malé signály, naopak při měření srdeční aktivity přímo na hrudníku budou signály mimo rozsah, a v počítači dojde k jejich ořezu. Ve výsledku je tedy OpenEEG limitováno svým dynamickým rozsahem, který je šitý na míru pouze měření EEG. Open BCI se svým širokým dynamickým rozsahem může být použit v aplikacích pro snímání více druhů biosignálů než jen EEG.“

3 Praktická část

Získávání informací, zejména konkrétnějších technických specifikací bylo velmi náročné. Vzhledem k úplné absenci tištěných zdrojů bylo nutné internetové zdroje i vícekrát prověřovat a někdy nešlo informace ověřit vůbec, a proto nebyly pro svou nejednoznačnost vůbec uvedeny.

Pro výzkum jsou jednoelektrodová zařízení nevhodná a soustředí se pouze na snímání jednoho typu kanálu EEG.

Z hlediska počtu elektrod jsou nejlepšími OpenBCI R&D Kit a Emotiv EPOC+. Zařízení od OpenBCI umožňuje měřit nejen EEG, ale i EKG a další, a to na různých místech lebky, protože ze pozice elektrod měnit podle potřeby, a případně je přepojit i na jiné části těla. Emotiv EPOC+ je naopak zaměřen pouze na mozkovou aktivitu a je vázán na software, který si uživatel musí paušálně platit. Sečtou-li se ale náklady na výrobu OpenBCI a koupi čipu, a porovnájí s náklady spojenými s pořízením EPOC+ pohybuje se cena na podobné výši \$1000.

Pokud není potřeba zcela univerzální zařízení, vhodnými kandidáty byly Emotiv Insight (5+2 referenční elektrody), Muse Headband (5 elektrod) a novinka od OpenBCI Ganglion (4 elektrody). Výhodou OpenBCI Ganglion je dostupnost softwaru bez poplatku, stejně tak je tomu i u Muse Headband. Ceny za pořízení těchto BCI nepřesahují \$500. Z této skupiny bylo rozhodnuto zakoupit Muse Headband, které bylo doručeno na fakultu v listopadu 2016.

Očekávanými nevýhodami bylo umístění elektrod, avšak velkým lákadlem bylo přehledně zpracovaná vývojářská příručka a knihovny pro Python.

3.1 Muse Headband 2016

Při prvním testování Muse bylo zjištěno, že nám byla doručena nejnovější verze 2016. Po instalaci ovladačů a zprovoznění dongle pro bluetooth LE připojení k počítači se ovšem ukázalo, že dostupné aplikace pro práci se signálem a surovými daty z Muse jsou kompatibilní pouze s verzí 2014. Software development kit 2016 byl dostupný pouze pro MAC nebo ve formě Android aplikace, která umožňovala zobrazení dat v reálném čase na displeji zařízení s Androidem v 6.0 (API 23). Tímto nastala změna v plánovaném postupu. V první řadě bylo třeba zajistit samotný přenos dat ze zařízení Android do počítače.

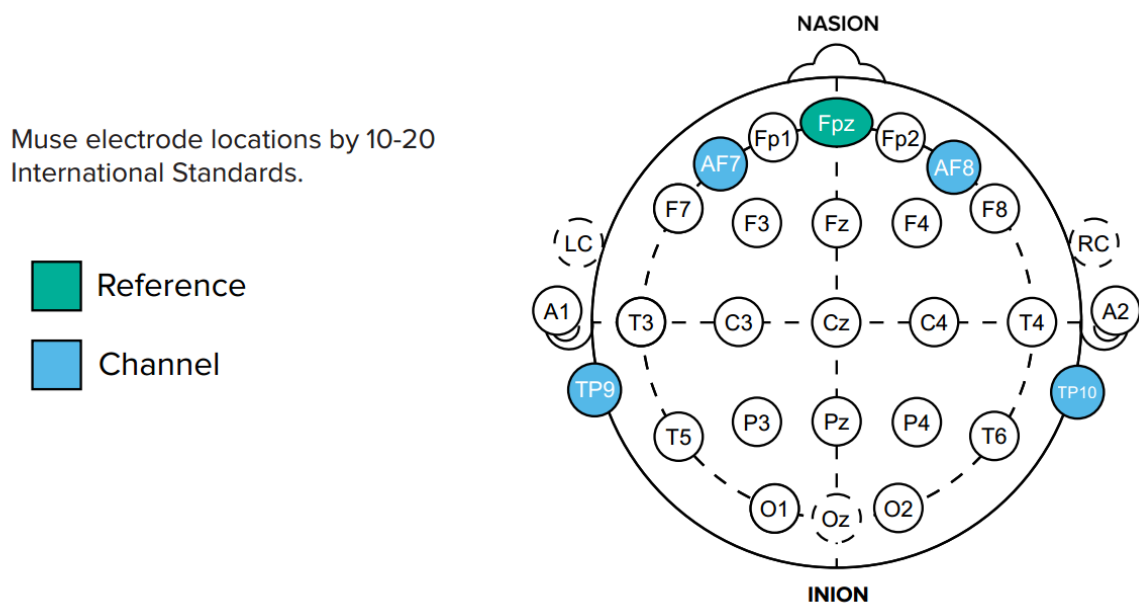
3.1.1 Technické specifikace

V druhé modelové řadě Muse byly provedeny četné změny. Byl odstraněn druhý micro USB port, který umožňoval připojení dalších modulů (například měření EKG), došlo ke zlepšení

kapacity baterie a také snížení spotřeby díky použití bluetooth LE. Došlo také k navýšení frekvence vzorkování EEG. Další rozdíly lze najít v Příloze 3.

3.1.2 Umístění elektrod

Dvě měřicí elektrody jsou umístěny na čelní straně hlavy (stříbrné kontakty), ve středu pak leží tzv. referenční elektroda. Další dvě, pryžové, se pak umísťují za uši a tím také celý headset drží svou pozici (Obr. 17).

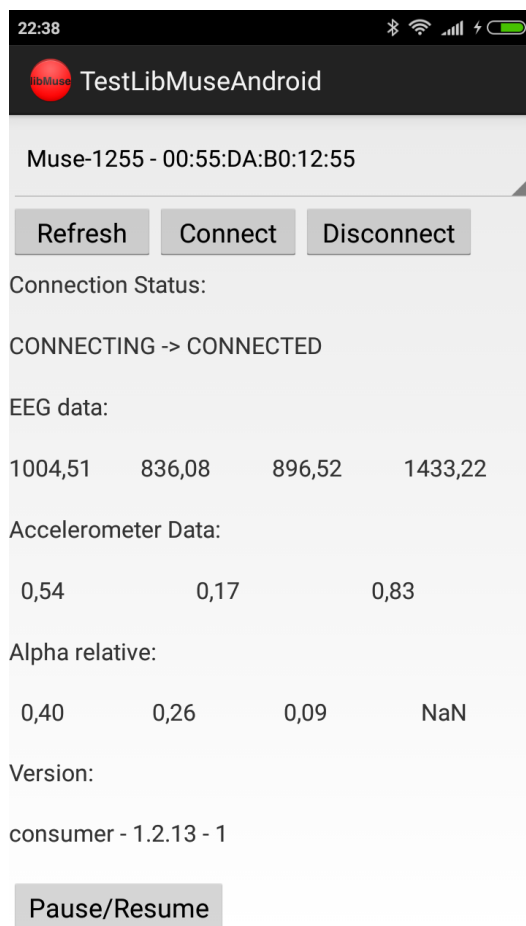


Obr. 17: Schéma umístění elektrod dle mezinárodního standardu 10/20 [44 s.3]

3.2 Android SDK a jeho nutná modifikace

3.2.1 Aplikace TestLibMuse

Na stránkách Muse v sekci pro vývojáře [45] byl doporučen postup instalace zkušební aplikace TestLibMuse (Obr. 18). K vývoji bylo použito Android Studio. Je nutné podotknout, že spuštění aplikace není možné v simulaci mobilního zařízení v tzv. emulátoru, protože nedisponuje možností užití bluetooth. Pro testování byl použit smartphone Xiaomi Redmi 3S s verzí Android 6.0.1 a MIUI 8.1.1.0.



Obr. 18: Screenshot prvního úspěšného měření s Muse

3.2.2 Projekt v AndroidStudio

SDK bylo tedy napsáno v objektovém jazyce Java. Zjednodušeně se projekt rozděluje na layout `.xml` a kódovou `.java` část, která se v průběhu odkazuje na jednotlivé prvky, které jsou viditelné uživateli na obrazovce zařízení.

Vzhledem k téměř nulovým zkušenostem bylo velmi obtížné zpočátku porozumět samotnému SDK a jeho struktuře. Snahou tak bylo kód efektivně modifikovat a nezasahovat do ostatních fungujících částí. Do původního `.xml` souboru byl přidán text box pro zadání adresy serveru, kam budou data posílána, jinak nebylo třeba grafickou stránku aplikace nijak měnit. Ve výsledku fungovala jen jako prostředník pro přenos dat, a tak nebylo nutné aplikaci po této stránce více vylepšovat.

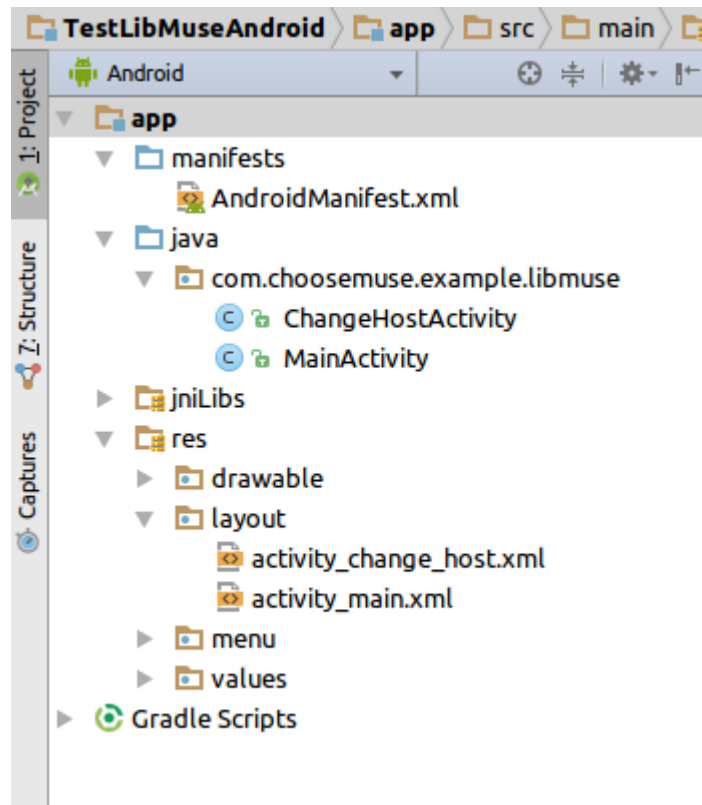
Pro práci se sítí bylo nejprve nutné přidat do souboru `AndroidManifest.xml` povolení pro síťové připojení. Dále stojí za zmínku následující čtveřice řešení:

Hierarchie v rámci projektu aplikace

Na následující stromové struktuře (Obr. 19) je vidět stavba celého projektu v Android Studio.

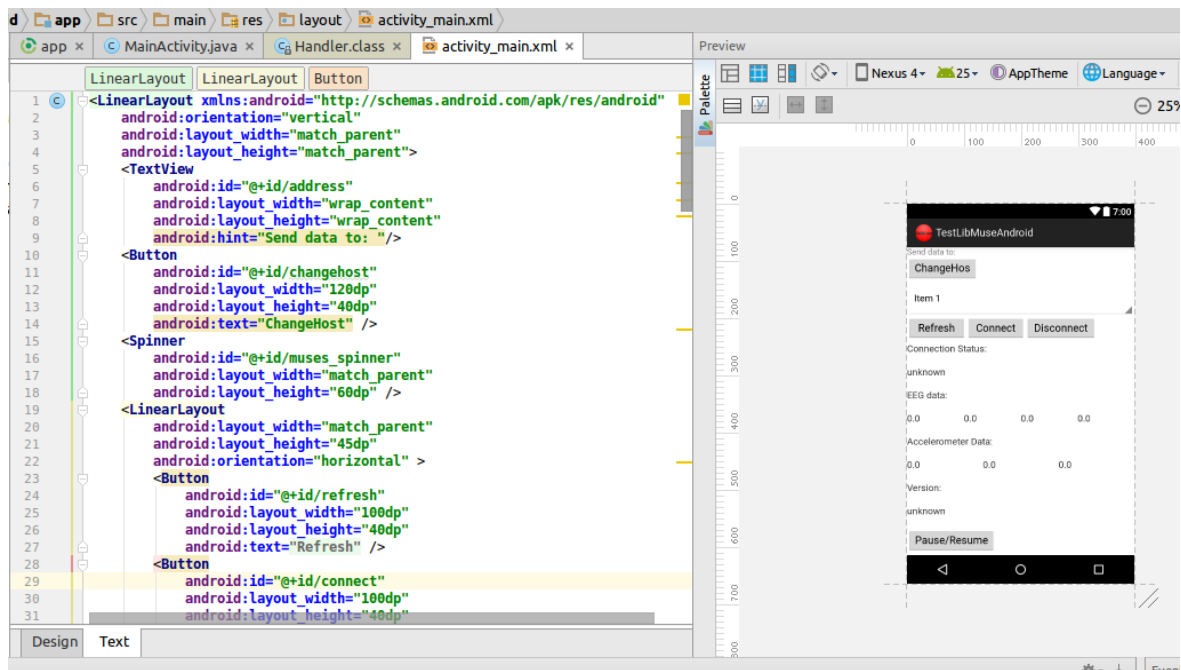
Pro potřeby bakalářské práce byl modifikován soubor `AndroidManifest.xml`

a `MainActivity.java`.



Obr. 19: Stromová struktura projektu v Android Studio

Pro rozšíření funkčnosti byl pak přidána aktivita `ChangeHostActivity.java`, která se uživateli zobrazí v podobě nového okénka pro zadání adresy serveru pro příjem dat, a soubor layoutu (rozložení) `activity_change_host.xml` (Obr. 20). Práce se značkovacím jazykem xml byla velmi intuitivní hlavně proto, že je velmi podobný např. html.



Obr. 20: Soubor xml a jeho překlad do grafické podoby

Pomocné proměnné v MainActivity

Do již hotového projektu, který zastává funkcí zatím jediného dostupného SDK na webových stránkách Muse, bylo tedy nutné přidat takový kód, jež zachytí data, která jsou nyní pouze zobrazována, a odešle je pomocí http požadavku na server. Původně měla být surová EEG data dostupná přímo z knihovny pro Python, a proto byla práce v jazyce Java nemilým překvapením. Potřebné teoretické znalosti o Java byly získány z [46], praktické pak z dotazů na fóru StackOverflow [47].

Nejjednodušším řešením bylo data odesílat formou POST na server. V úvahu připadalo použití Java knihovny JSON, avšak toto se ukázalo jako nefunkční (vizte níže). V konečném důsledku byly přidány následující proměnné, jak ukazuje Obr. 21.

```

private TextView address;
private final String DEFAULT_HOST="http://147.32.114.72:8000";
private final int CHANGE_HOST_CODE=123;
private Timer myTimerGrabData=null;
private final int CONNECTION_TIMEOUT=3000; //ms
private final int UPDATE_FREQUENCY=3000; //ms
private final int UPDATE_PER_SECND=128;
private int counter=0;
private double[] x=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];
private double[] y=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];
private double[] z=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];
private double[] a1=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];
private double[] a2=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];
private double[] a3=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];
private double[] a4=new double[UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)];

```

Obr. 21: Nové proměnné v MainActivity.java

Za zmínku stojí proměnná `DEFAULT_HOST`, ve které je uložena IP adresa, která je pro testování nejčastěji použita a je uživateli na začátku nabídnuta jako tzv. hint text v textovém poli. Pro posílání dat byl vytvořen vlastní časovač pro sběr dat `myTimerGrabData`. Pokud server nebude odpovídat je nastavena hodnota 3 s (3000 ms) jako hraniční pro zastavení odesílání dat. Frekvence odesílání dat byla experimentálně stanovena na 3 s. Pokud by se data odesílala rychleji, metoda `addToQueueEEG()` by takový nápor nezvládala a bylo by nutné použít jiné řešení formou kontinuálního toku (streamu) dat. Číslo `UPDATE_PER_SECOND` je nastaveno na vzorkovací frekvenci 128 Hz. Muse samo o sobě dokáže vzorkovat i 256 Hz, ale opět by docházelo k přehlcení požadavku POST, a chybám z přetečení dat. Proto byl zvolen nejbližší možný kompromis v podobě zmenšení frekvence ukládání vzorků. Všechny proměnné jsou označeny klíčovým slovem `private` (zavedená konvence tak, aby nemohly být tyto pracovní hodnoty měněny zvnějšku), a některé navíc slovem `final`, které zajišťuje, aby nedošlo k přepsání během provozu aplikace. Naopak změny musí být povoleny u pomocné proměnné `counter`, která zajišťuje, že se po době

$$\text{UPDATE_PER_SECOND} \cdot \frac{\text{UPDATE_FREQUENCY [ms]}}{1000} \quad (1)$$

odejde požadavek POST na server. Následuje deklarace polí pro ukládání desetinných hodnot z akcelerometru (x, y, z) a kanálů EEG, kde `a1` odpovídá pozici TP9, `a2` AF7, `a3` AF8 a `a4` TP9.

Timer a odesílání

Metoda `schedule(TimerTask task, long delay, long period)` časovače (timeru) slouží k provádění specifického úkonu v daném, opakovaném časovém intervalu. Toho bylo

využito právě pro ukládání jednotlivých vzorků vždy po 1/128 s a také odesílání dat tehdy, když nabyde pomocný counter hodnoty 384 (128*3000/1000). Zpoždění (delay) tedy udává dobu odkladu, než se provede odeslání, a mezičas (period) je interval mezi spouštěními opakujícího se úkolu (Obr. 22).

```
myTimerGrabData = new Timer();
myTimerGrabData.schedule(() -> {
    addToQueueACCEL();
    addToQueueEEG();
    counter++;
    if(counter==UPDATE_PER_SECND*(UPDATE_FREQUENCY/1000)) {
        Log.i("Thread","TimetThread: "+Thread.currentThread().getId());
        if(dataTransmission)
            sendDataToServer();
        counter = 0;
    }
}, 0, 1000/UPDATE_PER_SECND);
}
```

Obr. 22: Timer task

Vždy v čase 1/128 s (vzorek) se provede task timeru (v novém vlákně – threadu, protože operace jako je odesílání dat na server nemohou probíhat v hlavním UI vlákně) – tedy přidá se hodnota z akcelerometru a EEG, inkrementuje se pomocný counter a zkontroluje se, nenabyl-li hraniční hodnoty pro odeslání dat (zároveň se kontroluje, je-li zahájen přenos dat). Pokud nastala chvíle, kdy je dosaženo 384 vzorků, data se odešlou a counter je potřeba vynulovat.

Ukládání a formátování dat k odeslání

V SDK je přítomna konstrukce switch/case, která ukládá vzorky podle typů paketů do polí accelBuffer a eegBuffer v intervalech rovných vzorkování 256 Hz. Jak bylo zmíněno dříve, není možné pracovat s lepším vzorkováním než 128 Hz. Proto byly vytvořeny pomocné objekty-fronty na ukládání dat, a to addToQueueACCEL() a addToQueueEEG(). V nich bylo také původně zamýšleno tvořit JSON objekty pro odeslání dat (Obr. 23). Avšak použití JSON nebylo možné, nejpravděpodobněji kvůli rychlosti a množství dat, které bylo třeba zapsat. Tím sice odpadla možnost jednoduchého zanesení časového údaje (timestamp), nakonec se však ukázalo, že není potřeba, protože data chodí v daném intervalu, který lze snadno vypočítat.

```

private void addToQueueACCEL() {
    x[counter]=accelBuffer[0];
    y[counter]=accelBuffer[1];
    z[counter]=accelBuffer[2];
    /*try {
        double a1=accelBuffer[0];
        double a2=accelBuffer[1];
        double a3=accelBuffer[2];
        final JSONObject data = new JSONObject();
        data.put("type", 1);
        data.put("tm", System.currentTimeMillis());
        data.put("x", a1);
        data.put("y", a2);
        data.put("z", a3);
        packetQueue.put(data);
    } catch (JSONException e) {
        e.printStackTrace();
    }*/
}

private void addToQueueEEG() {
    a1[counter]=eegBuffer[0];
    a2[counter]=eegBuffer[1];
    a3[counter]=eegBuffer[2];
    a4[counter]=eegBuffer[3];
}

```

Obr. 23: Pomocné fronty

Pro odeslání byl tedy vytvořen řetězec `dataString` za pomoci cyklu `for` a jednoduché podmínky. Nebylo jisté, zda skript na straně serveru bude schopen efektivně rozdělit (parsovat) příchozí data pomocí knihovny JSON. Ukázalo se ale, že příchozí data lze bez problémů parsovat do seznamu (Python list).

```

// Prepare data
String dataString = "{\"data\":[";
for (int i = 0; i < UPDATE_PER_SECND * (UPDATE_FREQUENCY / 1000); i++) {
    if (i == 0)
        dataString += "[";
    else
        dataString += ",";
    dataString += x[i] + "," + y[i] + "," + z[i] + "," + a1[i] + "," + a2[i] + "," + a3[i] + "," + a4[i] + ",";
}
dataString += "]}";

```

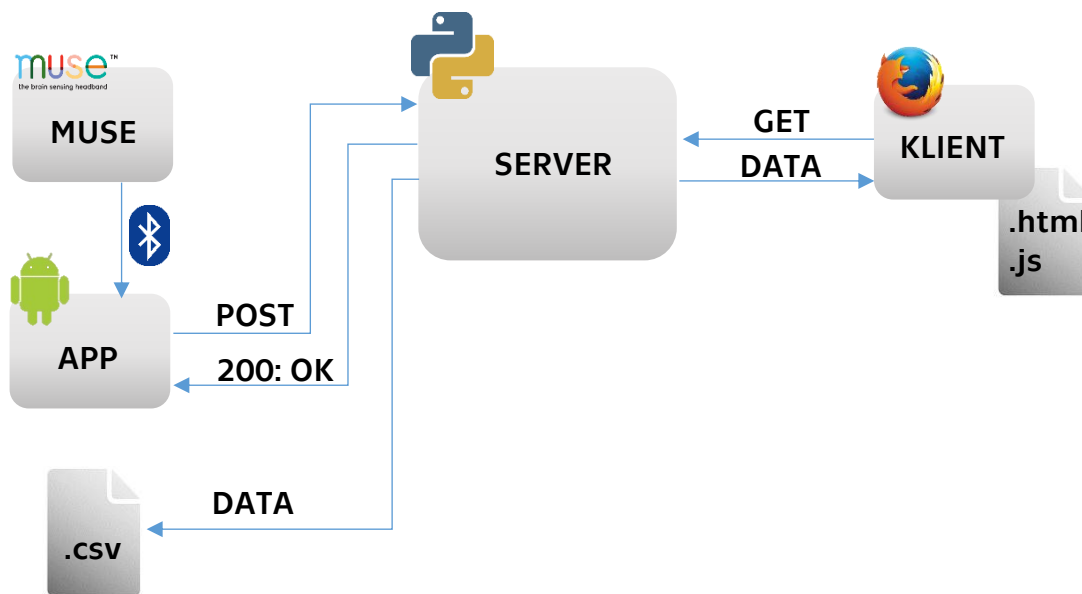
Obr. 24: Formátování dat k odeslání

Dokud je pomocná proměnná `i` menší než konečný počet vzorků, projde se podmínkou, která rozliší začátek, nebo střed zápisu dat. Po doběhnutí cyklu se na konec `dataString` zapíše uzavření jedné datové řady.

3.3 Přijímání, zpracování a ukládání dat pomocí Pythonu

Nyní jsou data již přenesena z Muse přes bluetooth do mobilního zařízení, které pomocí POST odesílá data serveru. Ten data přijme, a pokud nedojde k žádné chybě, odešle odpověď s kódem

200. Server přijímaná data postupně odkládá do tabulkového souboru. Klient, v tomto případě prohlížeč, po otevření adresy serveru zobrazuje html soubor, který v sobě nese zároveň kód v jazyce JavaScript, pomocí něhož se poté dynamicky vykresluje v prohlížeči graf. Tento postup graficky znázorňuje Obr. 25.



Obr. 25: Schéma komunikace se serverem

3.3.1 SimpleHTTPServer

V jazyce Python je pro jednoduché serverové aplikace dostupný modul `SimpleHTTPServer` a jeho handler `SimpleHTTPRequestHandler`, který obsluhuje reakce na dané události (příjem dat apod.). V základu obsahuje metodu `do_HEAD()`, která odpovídá na request (dotaz) typu HEAD a odesílá zpět tzv. hlavičku. Druhou připravenou metodou je `do_GET()`, která se, pokud není uživatelem určeno jinak, nasměruje do téhož adresáře, kde se nachází zdrojový kód. V této složce pak hledá soubor `index.html` nebo `index.htm`. Pokud se ve složce nenachází, vypíše se stromová strukturu obsahu složky. Pokud má metoda nahlížet do uživatelem nastavené složky, ve které se ale nenachází žádný soubor, nelze nic vypsát ani do stromové struktury a je vráceno chybové hlášení s kódem `404: Nenalezeno` [48].

Pro vytvoření serveru je vhodné využít knihovnu `SocketServer`. Podle [49] se jedná o framework² pro síťové servery, zjednodušující práci programátora tak, že nemusí složitě vymýšlet jak server vůbec zprovoznit, což umožňuje více se zaměřit na vývoj samotné aplikace.

² Podle [50] je framework souborem nástrojů, knihoven, zvyklostí a ověřených postupů, které mají v obecné formě podobu opakovaně použitelných modulů.

Základní struktura serveru je na Obr. 26. Je potřeba nastavit číslo portu (např. 8000, který spadá do rozsahu čísel portů přenosu dat protokolem TCP). Pro obsluhu událostí je použit buď základní `SimpleHTTPRequestHandler`, nebo se vytvoří potomek této metody, jak bude ukázáno v další podkapitole. V tomto konkrétním případě se jedná o přenos dat a proto je použita třída `SocketServer.TCP(adresa, handler)`. V parametru `adresa` je poté udána IP adresa počítače, na němž server běží, a číslo portu ve formátu `(IP_adresa, PORT)`. `TCPServer` poté volá jednu ze dvou metod `handle_request()`, nebo `serve_forever()`, pro obsluhu jednoho, nebo mnoha requestů.

```
1 import SimpleHTTPServer
2 import SocketServer
3
4 PORT = 8000
5
6 Handler = SimpleHTTPServer.SimpleHTTPRequestHandler
7
8 httpd = SocketServer.TCPServer("", PORT, Handler)
9
10 print "Serving at port", PORT
11 httpd.serve_forever()
```

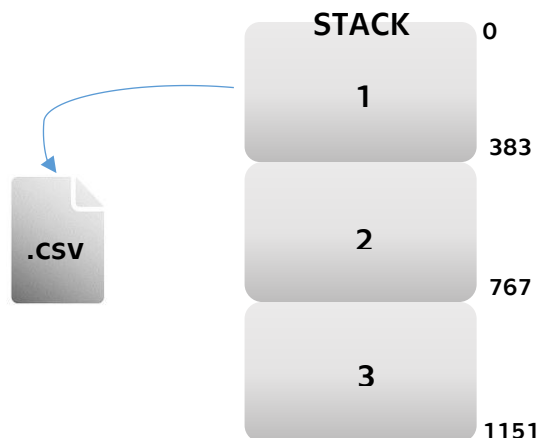
Obr. 26: Použití SimpleHTTP Server a SocketServer

3.3.2 Příjem dat a uložení do souboru

Pro příjem dat bylo třeba vytvořit nový request handler s názvem `MyRequestHandler`. Ten dědí od svého rodiče mj. metody `do_HEAD()` a `do_GET()`. Pro požadovanou funkci serveru je potřeba vytvořit ještě metodu `do_POST()`. Ta má tři funkce: převést data z formátu JSON do pole typu `numpy array`, uložit data do zásobníku a odložit část z nich do tabulkového souboru.

Aby byl kód univerzální a fungoval i pro úplně první spuštění serveru, je zásobník (`stack`) naplněn nulami. Ty jsou postupně nahrazeny naměřenými hodnotami.

První krok tohoto procesu je na Obr. 27. Horní třetina zásobníku (`stack`), realizovaného pomocí pole, se uloží do souboru.



Obr. 27: Uložení dat ze zásobníku do souboru

Uložení probíhá pomocí modulu `csv` a jeho metody `writer()`. Soubor se nejdříve otevře (pokud neexistuje, tak se vytvoří) a cyklem `for` a funkcí `range()`, nebo `xrange()` se generují jednotlivé řádky souboru. Tyto dvě funkce se od sebe liší způsobem, jakým generují číselnou posloupnost. `Range()` tvoří čísla v paměti v datovém typu `list` (seznam). `Xrange()` vrací sice stejné hodnoty, které by byly v seznamu od `range()`, ale jeho data jsou objektem, a neskladují se všechny současně. Výhodou `xrange()` jsou tedy značně menší nároky na paměť, což se hodí při generování velkých rozsahů. Stejným způsobem, cyklem `for` a `range()`, se vytvoří sedm sloupců odpovídajících třem signálům z akcelerometru a čtyřem z EEG. `Writer` (zapisovatel) zapisuje řetězce oddělené čárkou, a nový řádek odděluje pomocí `\r\n`. Zápis v kódu je na Obr. 28.

```

1 import SimpleHTTPServer
2 import SocketServer
3 import numpy as np
4 import cgi
5 import csv
6 import json
7 from datetime import datetime
8
9 stack = np.zeros((1152,7))
10 now = datetime.now.strftime('%Y-%m-%d_%H-%M-%s')
11
12 class MyRequestHandler(SimpleHTTPServer.SimpleHTTPRequestHandler):
13
14     def do_POST(self):
15         global stack
16         with open('data_%s.csv' % now, 'ab') as csvfile:
17             writer=csv.writer(csvfile)
18             for i in xrange(383):
19                 writer.writerow([stack[i,j] for j in range(7)])
20

```

Obr. 28: Ukládání do souboru `.csv`

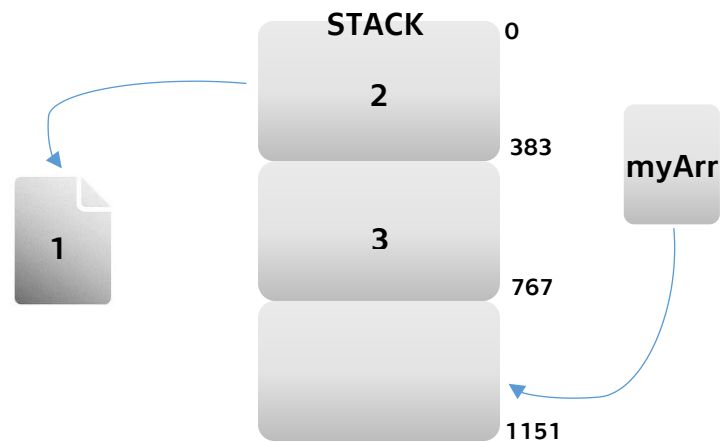
Samotný příjem dat probíhá pomocí modulu cgi (Common Gateway Interface) na řádcích 21-23 v kódu na Obr. 29.

```
1 import SimpleHTTPServer
2 import SocketServer
3 import numpy as np
4 import cgi
5 import csv
6 import json
7 from datetime import datetime
8
9 stack = np.zeros((1152,7))
10 now = datetime.now.strftime('%Y-%m-%d_%H-%M-%s')
11
12 class MyRequestHandler(SimpleHTTPServer.SimpleHTTPRequestHandler):
13
14     def do_POST(self):
15         global stack
16         with open('data_%s.csv' %now,'ab') as csvfile:
17             writer=csv.writer(csvfile)
18             for i in xrange(383):
19                 writer.writerow([stack[i,j] for j in range(7)])
20
21         length = int(self.headers.getheader('content-length'))
22         json_data = cgi.parse_qs(self.rfile.read(length),
23                                 keep_blank_values=1).keys()[0]
24
```

Obr. 29: Vytvoření `json_data`

Do pomocné proměnné se zapíše délka příchozích dat nacházející se v headeru (hlavičce) `content-length` a převede se na číslo. Poté je třeba rozparsovat přijatá data, k tomuto účelu slouží funkce `parse_qs(qs, keep_blank_values)`. Jejím vstupem je `qs`, neboli query string (řetězec dotazu, resp. vlastní data) a parametr `keep_blank_values` určuje, zdali se mají prázdné hodnoty převádět také, nebo ignorovat, jakoby vůbec nebyly přeneseny [51]. CGI vrací data ve formě slovníku (dictionary). Data se v něm nacházejí na nulté pozici, a jsou uloženy do `json_data`. Hodnoty ještě stále nejsou nijak přístupné, a proto bylo vhodné použít modul `json`, který umí efektivně data rozparsovat. Do proměnné `data` jsou pak přiřazeny hodnoty odpovídající klíči `data`. Pro snadnější manipulaci byl takto vzniklý seznam (list) převede na tzv. numpy pole (array) `myArr`.

V posledním kroku se poté provede nové zapsání do spodní třetiny zásobníku. Realizuje se tak funkce LIFO (last in, first out) jak znázorňuje schéma na Obr. 30.



Obr. 30: Nová data do zásobníku

Podle [52] ovlivňuje EEG data offset v hodnotě 800 μV . Proto je při manipulaci s daty v zásobníku odečten. Dojde k výměně dat a poté odeslání odpovědi (kód 200) v hlavičce (Obr. 31).

```

27     myArr=np.asarray(data)
28     myArr[:,3:]-=800
29     stack[:768,:]=stack[384:,:]
30     stack[768:,:]=myArr
31
32     self.send_response(200)
33     self.send_header("Content-type", "text/html")
34     self.end_headers()
35     self.wfile.write("")

```

Obr. 31: Poslední část kódu v metodě do_POST()

3.3.3 Úprava metody do_GET()

Požadavek GET přichází ze strany klienta, prohlížeče, každé tři vteřiny (provedení vysvětluje následující kapitola). Pro server je důležité, že adresa požadavku pro poskytnutí dat je zakončena /values. Zásobník stack je nejprve převeden pomocí funkce `numpy.char.mod(a, data)` tak, aby obsahoval místo čísel řetězce. Pomocí parametru `a` jde provádět úpravy čísel (např. počet desetinných míst). V tomto případě je místo parametru `a %f`, což značí, že zdrojová data jsou typu float (s plovoucí des. čárkou, a nezaokrouhlují se). Pro vykreslení do grafu se použije pouze první třetina zásobníku, tedy od 0. prvku do 383. včetně. Jednotlivé řádky zásobníku se převedou pomocí funkce `list()` na seznam (list), kde jsou data odděleny v rámci jednoho indexu (řádku) čárkami. Funkce `a.join(data)` pak

sloučí data do jednoho řetězce a také použije jako oddělovač čárku (to však není nutné, za parametr a můžeme volit libovolný znak). Pro odpověď klientovi je třeba ještě celý pomocný seznam `temp` převést na jeden řetězec, a to se provede opakováním funkce `join`, kde se jednotlivé buňky sloučí, oddělí středníkem a uloží do `out`. Data jsou pak odeslána pomocí `wfile`, což je objekt, který se chová podobně jako soubor, do kterého lze psát nebo z něj číst [49]. Zde je použit jako odpověď klientovi, tedy `self.wfile.write(out)`. Ve všech ostatních případech (`else`) vrací metoda pouze soubor `index.html`. Celý postup je na následujícím Obr. 32.

```
38 def do_GET(self):
39     if self.path.startswith("/values"):
40         """Values."""
41         stack_str = np.char.mod('%f', stack)
42         temp = [",".join(list(stack_str[:384,i])) for i in range(7)]
43         out = ";".join(temp)
44
45         self.send_response(200)
46         self.send_header("Content-type", "text/html")
47         self.end_headers()
48         self.wfile.write(out)
49     else:
50         """Serve a GET request."""
51         f = self.send_head()
52         if f:
53             self.copyfile(f, self.wfile)
54             f.close()
```

Obr. 32: Metoda `do_GET()`

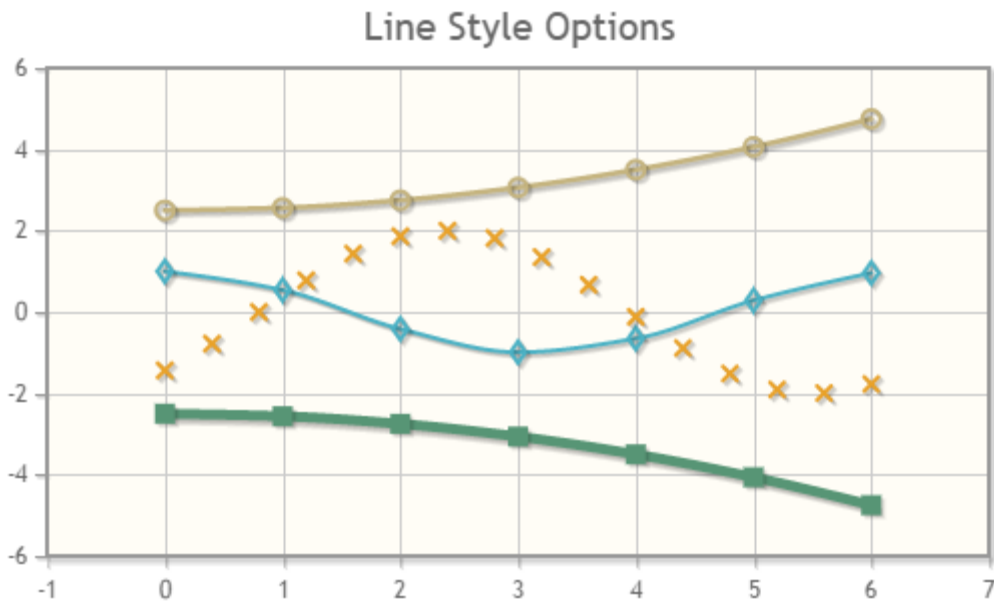
3.4 Vykreslování grafů pomocí JavaScript

JavaScript je jazyk, který je, ač tomu název přímo nenapovídá, zcela nezávislý jazyk, a s Javou má pouze podobnou syntaxi. JavaScript se používá pro dynamické úpravy webových stránek a zapisuje se přímo do html kódu. Spolu s ním je také JavaScript vykonáván až v klientském prostředí (prohlížeči) [53].

3.4.1 jqPlot

Na internetu je široká nabídka předpřipravených grafů, jejichž JavaScript kód lze jednoduše vložit do webových stránek, aniž by bylo třeba znát přesnou syntaxi. Každý graf nebo funkce pak závisí na zdrojových souborech, které musí být na začátku html souboru deklarovány. Všechny vycházejí z knihovny z jQuery, která usnadňuje dynamickou manipulaci s kaskádovým formátovacím stylem CSS, obsluhuje události v HTML apod.

Pro účely této práce byl vybrán jqPlot. Jedná se o opensource nabízející velké množství typů základních i speciálních interaktivních grafů. K vykreslení dat z Muse postačí jeden z obvyklejších grafů, a tím je čárový graf (line chart) [54].



Obr. 33: Ukázka čárového grafu jqPlot [54]

Graf je schopný přijímat datové řady a ukládá je do proměnné `response`. Jak bylo uvedeno v kapitole o Pythonu, data přichází jako řetězec. Je tedy potřeba provést reverzní postup k přechodnému slučování. `Response` je tedy rozdělena do pole, kde oddělovačem je středník. Jednotlivé datové řady jsou pak utvořeny rozštěpením jednotlivých indexů pole `columns`.

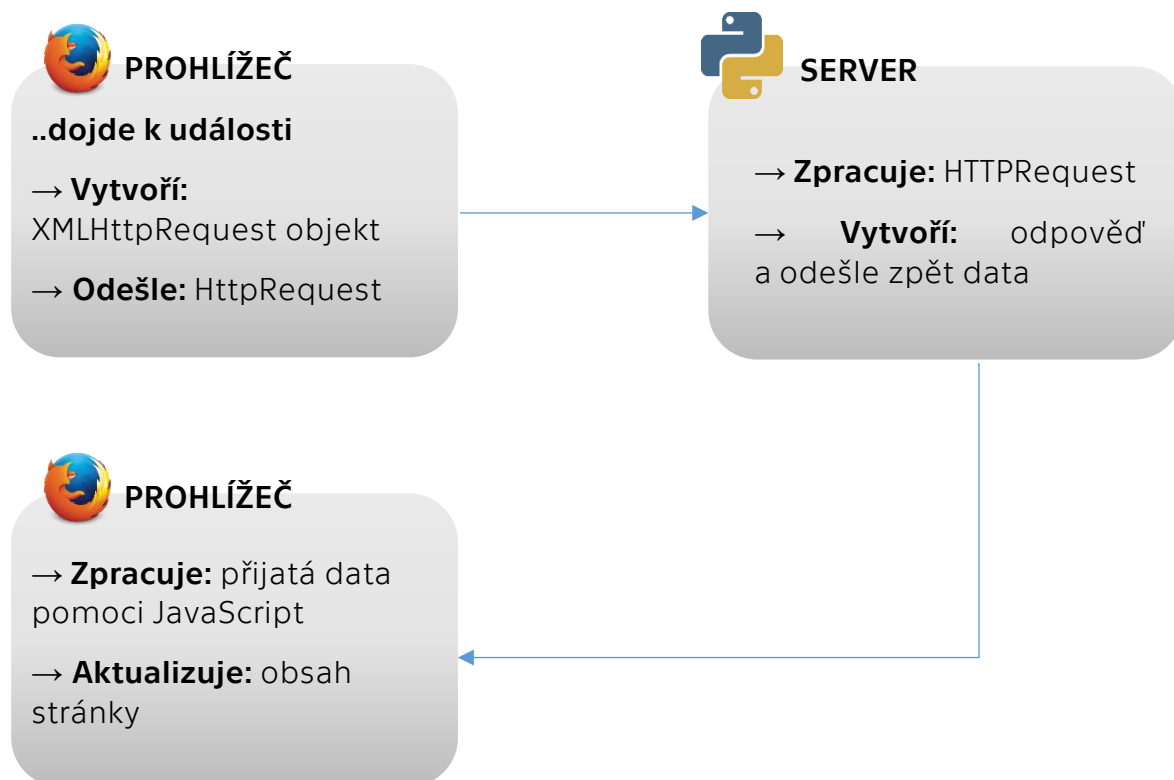
V nastavení zobrazení grafu bylo zakázáno vyhlazení a taktéž zobrazování označování pozice bodu. Povoleny byly naopak popisky os a legenda. Je užitečné, že hodnoty legendy lze dynamicky měnit. Pro účely této práce bylo však dosazeno pouze pole, kde jsou popisky statické. Možnost dynamického zásahu nicméně ukazuje na velký potenciál práce s grafy v JavaScript.

JavaScript musí být provázán s html kódem. To se děje pomocí identifikátoru `id` prvků html na které se JavaScript odkazuje, např. takto: `<div id = "plot2" class = "c1"></div>`.

3.4.2 Ajax

Aby se data v grafu dynamicky měnila, v případě odesílání dat z Muse je potřeba obnovit po třech sekundách, musí klient posílat požadavek v pravidelných intervalech. První možností je, že bude uživatel stránku obnovovat stiskem klávesy F5. To je velmi nepohodlné. Stránka samotná by se dokázala aktualizovat také bez přispění uživatele, ale stejně by docházelo ke zbytečnému znovuvypisování neměnných částí html stránky. Řešením je takzvané ajaxování (ajaxing).

AJAX neboli Asynchronous Javascript and XML nejjednodušeji znázorňuje následující schéma na Obr. 34.



Obr. 34: Schéma ajaxování

Funkce ajax obsahuje atributy `type`, `url`, `success` a `complete` (a další jako například `async` nebo `dataType`). V `type` je definován typ požadavku, v tomto případě `GET`, `url` specifikuje podobu požadavku – `values`. Na toto reaguje metoda `do_GET()` v Python skriptu. `Success` (neboli úspěch) pak provede příslušné operace po zdárném přijetí dat (zde vykreslení grafů). Když se dokončí vykreslování, přejde ajaxování do fáze `complete` (dokončeno), kde je teprve dovoleno provést požadavek k získání nových dat [55]. Příklad kódu je na Obr. 35.


```

1   var columns, x, y, z, tp9, af7, af8, tp10;
2   var interval = 3000;
3
4   function doAjax() {
5     $.ajax({
6       type: "GET",
7       url: 'values',
8       success: function(response){
9         //přijetí dat a porcování datových řad
10        $(document).ready(function(){
11          $.jqplot.config.enablePlugins = true;
12          ACCplot = $.jqplot ('plot1', [x, y, z ], {
13            //vlastnosti grafu a nastavení
14          });
15
16          EEGplott = $.jqplot ('plot2', [tp9, af7, af8, tp10], {
17            //vlastnosti grafu a nastavení
18          });
19        });
20      };
21      complete: function(response){
22        setTimeout(doAjax, interval);
23      }
24    });

```

Obr. 35: Zjednodušená ukázka kódu pro ajaxování

4 Závěr

V teoretické části práce uvádím přehled dostupných zařízení, pro úplnost i těch, která se pro účely této práce nehodí. Jedná se zejména o jednoelektrodová zařízení, která se soustředí pouze na snímání jednoho typu kanálu EEG tak, aby sloužila spíše jako fitness pomůcka. Proto jsem tato zařízení vyjmula z výběru hned na začátku.

Rozhodla jsem se stanovit tři kritéria pro výběr vhodného BCI, a to: počet elektrod, dostupnost SDK a cena.

Z hlediska počtu elektrod byly nejlepšími OpenBCI R&D Kit a Emotiv EPOC+. Zařízení od OpenBCI umožňuje měřit (nejen) EEG, na různých místech lebky. Emotiv EPOC+ je ale vázán na software, který si uživatel musí paušálně platit. Náklady na pořízení jsou u obou BCI srovnatelné, cena by se pohybovala ve výši \$1000. Kvůli ceně jsem proto tato dvě zařízení ne zvolila.

Rozhodla jsem se tedy vybrat ne zcela univerzální (ve smyslu omezeného počtu elektrod a jejich fixní pozice) zařízení. Vhodnými kandidáty byli Emotiv Insight (5+2 referenční elektrody), Muse Headband (4 elektrody) a novinka od OpenBCI Ganglion (4 elektrody). Výhodou OpenBCI Ganglion je dostupnost softwaru bez poplatku, stejně tak je tomu i u Muse Headband, kde ale SDK vypadalo daleko lákavěji. Byly přístupné knihovny pro Python pro příjem i ukládání dat. Vývojářská dokumentace vypadala velmi přívětivě, přehledně a na fóru pružně reagoval tým výrobců na dotazy. Proto jsem se na základě konzultací s vedoucím práce rozhodla pořídit Muse Headband (\$500).

Když Muse dorazilo, zjistila jsem, že se jedná o nejnovější verzi, která má sice mnoho hardwarových výhod oproti Muse 2014, ale k této novince existovala pouze předváděcí Android aplikace, která zobrazovala data na displeji mobilního zařízení v reálném čase. Pokud jsem tedy chtěla s prací pokračovat, bylo nutné změnit postup a začít řešit, jak data vůbec přenést do počítače.

Přenos je realizován v jazyce Java metodou POST, která ale po zpětném uvážení není pro přenos takového objemu a hustoty dat vhodná. Proto je také zpracování a zobrazení dat jen „polovičně real-time“, neboť dochází ke zpoždění tři sekundy. Ty byly zvoleny na základě pozorování tak, aby byla data odesílaná bez chyb (přetečení apod.).

Původní Android aplikace byla modifikována, aby mohl uživatel zadat adresu serveru, který bude data přijímat. V části neviditelné pro uživatele pak dochází k odesílání naměřených

hodnot. Odesílání lze pozastavit, a pokud server tři sekundy neodpovídá, odesílání dat se přeručí.

Běh serveru byl vyřešen pomocí dvou knihoven (modulů) SimpleHTTPServer a SocketServer. Jsou to hojně užívané moduly a na internetu se nachází mnoho tutoriálů. Server předává data dál prohlížeči a zároveň ukládá data do souboru, který je opatřen časovou značkou a nachází se v téže složce na disku jako server.

Webové rozhraní jsem vytvořila jako jednoduchou html stránku, ve které zároveň běží kód v jazyce JavaScript. Pro vykreslení grafů jsem vybrala opensource nástroj jqPlot, který nabízí nepřehledné množství volně dostupných skriptů pro kreslení grafů. Vzhledem k tomu, že jqPlot se nacházel mezi nejlepšími deseti nástroji pro kreslení grafů, mě velmi překvapila jeho nedokonalá dokumentace. Pro další aplikace bych jej spíše nedoporučila. Jedinou věcí, kterou se nepodařilo vyřešit, je dynamická změna hodnot na ose x. S příchodem každé série dat začínají popisky vždy od nuly. Čárový graf u jqPlot tuto modifikaci neumožňuje. V Příloze 2 a 3 jsou ukázána vzorová měření, dále je přiložena tabulka technických specifikací Muse.

Pro maximální využití aplikace by bylo do budoucna vhodné vyřešit odesílání dat např. formou kontinuálního toku (streamu) dat, a zvolit efektivnější pomůcku k tvorbě grafů (například chart.js nebo Flot).

Zdroje

- [1] KRAJČA, Vladimír a Jitka MOHYLOVÁ. *Číslicové zpracování neurofyziologických signálů*. 1. vyd. V Praze: České vysoké učení technické, 2011. ISBN 978-80-01-04721-7.
- [2] HRAZDIRA, Ivo a Vojtěch MORNSTEIN. *Lékařská biofyzika a přístrojová technika*. 1. vyd. Brno: Neptun, 2001. ISBN 80-902896-1-4.
- [3] MILLET, David. The Origins of EEG: The Seventh Annual Meeting 2002. *International Society for the History of the Neurosciences* [online]. Los Angeles, California, USA, 2002 [cit. 2016-03-14]. Dostupné z: <http://www.bri.ucla.edu/nha/ishn/ab24-2002.htm>
- [4] POKORNÝ, Jan. Elektroencefalografie [online][prezentace].[cit. 2016-03-14]. Dostupné z: <http://fbmi.cvut.cz/files/nodes/657/public/EEG.pdf>
- [5] ŠVEJDA, Jaromír, Roman ŽÁK a Roman JAŠEK. Zpracování mozkové aktivity v BCI systémech. *Trilobit: Odborný vědecký časopis*. Zlín: Univerzita Tomáše Bati, **2012**(1). ISSN 1804-1795
- [6] STATSOFT. *Úvod do neuronových sítí*. Praha, 2013. Dostupné z: http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_linky.pdf
- [7] KNAPOVSKÝ, Martin a Michal ŠMRHA. *Brain-computer interface*. ČVUT Praha, 2013. Dostupné také z: https://nit.felk.cvut.cz/~dark/Vyuka/AST2013/2013_05_07_utery/knapovsky_smrha_-_bci/vypracovani.pdf
- [8] FARWELL, L.A. a E. DONCHIN. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*. 1988, **70**(6), 510-523. DOI: 10.1016/0013-4694(88)90149-6. ISSN 00134694. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/0013469488901496>
- [9] BOČKOVÁ, Martina. *Kognitivní procesy v mozku ve světle intracerebrální registrace indukované a evokované aktivity*. Brno, 2010. Dostupné také z: https://is.muni.cz/th/15525/lf_d/disertacni_prace.txt. Disertační práce. LF MU Brno.
- [10] SCHALK, Gerwin, Dennis MCFARLAND, Thilo HINTERBERGER a Jonathan WOLPAW. BCI2000: a General-Purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering*. New York, Albany, 2004, **51**(6), 1034-1043.

- [11] Interactive Productline. Customers version of Mindball Game Core. [fotografie]. *Mind Ball* [online]. Dostupné z: http://www.mindball.se/image/cache/data/products/game_core_customers_6-500x500.jpg. Formát: 500 x 500.
- [12] Mindball Game. *Interactive Productline* [online]. Nacka (Sweden), 2014 [cit. 2016-05-06]. Dostupné z: <http://www.mindball.se/mindball-game>
- [13] Co je lucidní snění? *Lucidní snění* [online]. 2015 [cit. 2016-05-06]. Dostupné z: <http://www.lucidnisneni.cz/>
- [14] iWinks - Dream Band. *iWinks* [online]. San Diego: iWinks LLC, 2016 [cit. 2016-05-06]. Dostupné z: <http://www.iwinks.org/aurora/>
- [15] Melon. Melon learns about and displays your focus for you. Get personalized tips (optional) when your focus dips too low [Obrázek]. *Kickstarter* [online]. Dostupné z: https://ksr-ugc.imgix.net/assets/000/589/158/ff8615f9eab982a24d4faa4ea9cb8398_original.png?w=680&fit=max&v=1368514380&auto=format&lossless=true&s=e94821320e1f3aa08a007879f9e2513f
- [16] Melon - a headband and app to measure your focus by Melon - kickstarter. *Kickstarter* [online]. Los Angeles, 2015 [cit. 2016-05-08]. Dostupné z: <https://www.kickstarter.com/projects/806146824/melon-a-headband-and-mobile-app-to-measure-your-fo/description>
- [17] Comparison of consumer brain–computer interfaces. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-08]. Dostupné z: https://en.wikipedia.org/wiki/Comparison_of_consumer_brain%E2%80%93computer_interfaces
- [18] Melon. Melon features [Obrázek]. *ThinkMelon* [online]. Dostupné z: https://support.thinkmelon.com/hc/en-us/article_attachments/202515467/zen_bandparts.gif
- [19] Focus Band|FAQ. *Focus Band - Mind Training Headset, proven, intuitive & rugged* [online]. Windaroo, Australia: T 2 Green Pty Ltd, 2015 [cit. 2016-05-08]. Dostupné z: <http://www.ifocusband.com/#!/faq/c1oql>
- [20] InteraXon. Connecting Muse To Your Mobile Phone [fotografie]. Muse [online]. Dostupné z: http://storage.googleapis.com/ix_chooseuse/uploads/assets/howItWorks_step_11.jpg
- [21] InteraXon [online]. Muse: The Brain Sensing Headband. 2016. [cit. 2016-05-08]. Dostupné z: https://storage.googleapis.com/ix_downloads/technical_specifications/Technical%20Specifications%2C%20Validation%2C%20and%20Research%20Use%20-%202016_v11.pdf
- [22] NeuroSky Store: EEG Headsets. *NeuroSky: EEG - ECG - Biosensors* [online]. San Jose, California [cit. 2016-05-11]. Dostupné z: <http://store.neurosky.com/>

- [23] VILMUR, Pete. Mysterious Force Trainer Toy Explained. In: *StarWars.com* [online]. 2009 [cit. 2016-05-11]. Dostupné z: <https://web.archive.org/web/20090211122524/http://starwars.com/vault/collecting/20090209b.html>
- [24] MindWave. *NeuroSky: EEG - ECG - Biosensors* [online]. San Jose, California [cit. 2016-05-11]. Dostupné z: <http://store.neurosky.com/pages/mindwave>
- [25] Research Tool. *NeuroSky: EEG - ECG - Biosensors* [online]. San Jose, California [cit. 2016-05-11]. Dostupné z: <http://store.neurosky.com/products/mindset-research-tools>
- [26] MindFlex. *NeuroSky: EEG - ECG - Biosensors* [online]. San Jose, California [cit. 2016-05-19]. Dostupné z: <http://store.neurosky.com/collections/hardware/products/mindflex>
- [27] BrainBandXL & MyndPlay Pro Bundle. *MyndPlay Ltd* [online]. London, 2016 [cit. 2016-05-19]. Dostupné z: <http://myndplay.com/products.php?prod=9>
- [28] MYNDPLAY [online]. *Creating Interactive Video Content for the MyndPlay Platform*. 2011. [cit. 2016-05-19]. Dostupné z: <http://myndplay.com/PDF/Createfinal.pdf>
- [29] MYNDPLAY. MyndPlay brainband [fotografie]. Gadgets and Gizmos [online]. Dostupné z: <http://cdn.gadgetsandgizmos.org/wp-content/uploads/2011/12/MyndPlaybrainband.jpg>
- [30] PLX. XWave Headset [fotografie]. Amazon.com [online]. Dostupné z: http://lh6.ggpht.com/_drWrR8Q5Jvw/TP4RpTSC0vI/AAAAAAAAAJ7c/3Wr0MYjQCXM/XWave-ins1.jpg
- [31] PLX. The XWave Sport components [fotografie]. Gizmag [online]. Dostupné z: http://img-2.gizmag.com/xwave_sport-3.jpg?auto=format%2Ccompress&fit=max&h=670&q=60&w=1000&s=cebb0acc71ba45cf0ca0c7c5bf0e9972
- [32] QUICK, Darren. Workout brain and body with the XWave Sport brainwave interface headband: Wearables. In: *Gizmag: New and Emerging Technology News* [online]. Carlton, Australia: Gizmag Ptd Lty, 2012 [cit. 2016-05-19]. Dostupné z: <http://www.gizmag.com/xwave-sport/23093/>
- [33] EMOTIV: Comparison. *EMOTIV - Brainwear: Wireless EEG Technology* [online]. San Francisco, USA, 2016 [cit. 2016-05-22]. Dostupné z: <http://emotiv.com/comparison/>
- [34] EMOTIV: Pure. EEG. *EMOTIV - Brainwear: Wireless EEG Technology* [online]. San Francisco, USA, 2016 [cit. 2016-05-22]. Dostupné z: <http://emotiv.com/product/emotiv-pure-ee/>
- [35] EMOTIV. 3D Brain Visualiser [video]. *EMOTIV - Brainwear: Wireless EEG Technology* [online]. Emotiv, 2015. [cit. 2016-05-22]. Dostupné z: <http://emotiv.com/the-science/>
- [36] BISHOP, Bryan. Emotiv EPOC EEG Headset Hacked. In: *Humanity +* [online]. Los Angeles, California, 2010 [cit. 2016-05-22]. Dostupné z: <http://hplusmagazine.com/2010/09/13/emotiv-epoc-eeeg-headset-hacked>
- [37] MURPHY, Joel a Connor RUSSOMANNO. OpenBCI: An Open Source Brain-Computer Interface For Makers. *Kickstarter* [online]. Brooklyn, New York, 2013 [cit. 2016-05-22].

Dostupné z: <https://www.kickstarter.com/projects/openbci/openbci-an-open-source-brain-computer-interface-fo>

[38] OpenBCI. Printable headset parts [fotografie]. OpenBCI [online]. Dostupné z: http://cdn.shopify.com/s/files/1/0613/9353/files/PrintedParts_grande.jpg?14956560392368527132

[39] OpenBCI. Recommended battery and charger [fotografie]. OpenBCI [online]. Dostupné z: http://cdn.shopify.com/s/files/1/0613/9353/files/batteryOptions_large.jpg?1557338893038188661

[40] MURPHY, Joel a Connor RUSSOMANNO. OpenBCI: Docs. *GitHub: How People Build Software* [online]. Brooklyn, New York, 2014 [cit. 2016-05-22]. Dostupné z: <https://github.com/OpenBCI/Docs/blob/master/headware/01-Ultracortex-Mark-III-Nova.md>

[41] OpenBCI: Shop. *OpenBCI Shop: Featured Products* [online]. Brooklyn, New York, 2016 [cit. 2016-05-22]. Dostupné z: <http://shop.openbci.com/collections/frontpage>

[42] The Modular EEG. *OpenEEG: EEG for the rest of us!* [online]. 2003 [cit. 2016-05-22]. Dostupné z: <http://openeeg.sourceforge.net/doc/modeeg/modeeg.html>

[43] MURPHY, Joel a Connor RUSSOMANNO. OpenBCI: An Open Source Brain-Computer Interface for Makers. *Kickstarter* [online]. Brooklyn, New York, 2013 [cit. 2016-05-22]. Dostupné z: https://www.kickstarter.com/projects/openbci/openbci-an-open-source-brain-computer-interface-fo/description#project_faq_76925

[44] INTERAXON. *Technical specifications, validation and research use*. Toronto, 2017. Dostupné také z: <http://developer.choosemuse.com/hardware-firmware/hardware-specifications>

[45] Android | Getting Started. *Muse Developers* [online]. Toronto: InteraXon, 2016 [cit. 2017-04-24]. Dostupné z: <http://developer.choosemuse.com/android/getting-started-with-libmuse-android>

[46] PECINOVSKÝ, Rudolf. *Java 8: úvod do objektové architektury pro mírně pokročilé*. Praha: Grada Publishing, 2014. Knihovna programátora (Grada). ISBN 978-80-247-4638-8.

[47] STACK OVERFLOW. *Stack Overflow: Newest Questions* [online]. 2017 [cit. 2017-05-01]. Dostupné z: <http://stackoverflow.com/questions>

[48] SimpleHTTPServer — Simple HTTP request handler. *Python: Welcome to Python* [online]. Beaverton, USA: Python Software Foundation, 2017 [cit. 2017-06-10]. Dostupné z: <https://docs.python.org/2/library/simplehttpserver.html#module-SimpleHTTPServer>

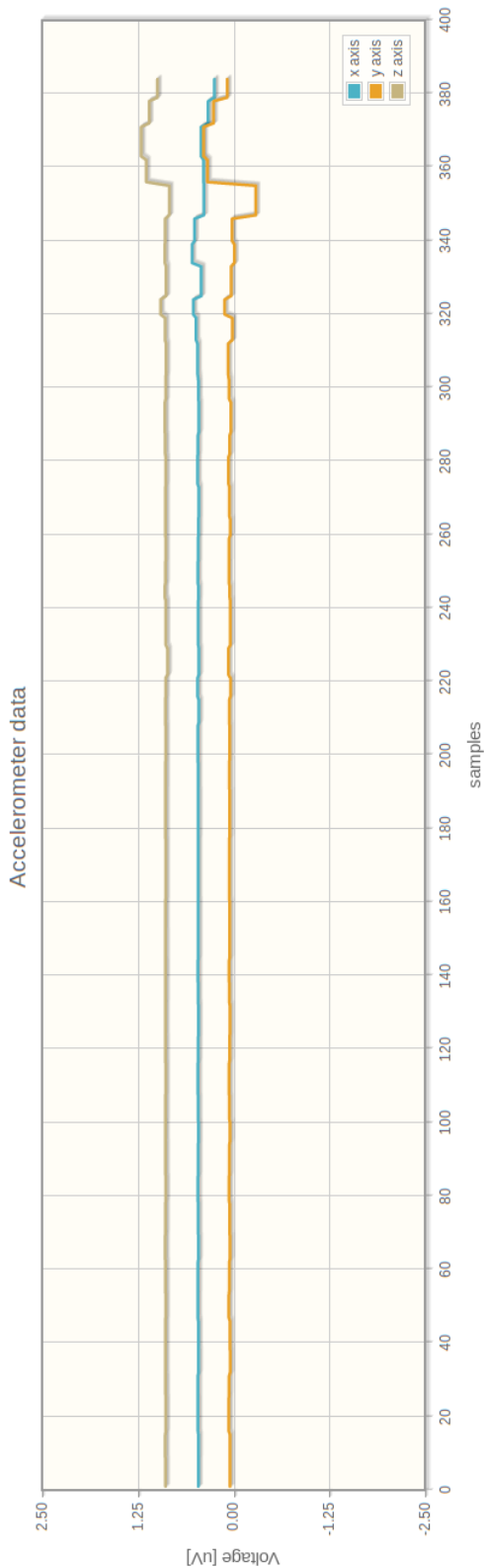
[49] Socket Server – a framework for network servers. *Python: Welcome to Python* [online]. Beaverton, USA: Python Software Foundation, 2017 [cit. 2017-06-10]. Dostupné z: <https://docs.python.org/2/library/socketserver.html>

- [50] CROFT, Jeff. Frameworks for Designers. a *List Apart: For People Who Make Websites* [online]. 2007 [cit. 2017-06-11]. ISSN 1534-0295. Dostupné z: <https://alistapart.com/article/frameworksfordesigners>
- [51] cgi — Common Gateway Interface. *Python: Welcome to Python* [online]. Beaverton, USA: Python Software Foundation, 2017 [cit. 2017-06-13]. Dostupné z: <https://docs.python.org/2/library/cgi.html?highlight=cgi#module-cgi>
- [52] EEG readings amplitudes: Muse Forum. *Tech Support: Muse Forum* [online]. Toronto: InteraXon, 2017 [cit. 2017-06-14]. Dostupné z: <http://forum.choosemuse.com/t/eeg-readings-amplitudes/347>
- [53] Javascript - úvod. *Jak psát web* [online]. Praha: Janovský, 2016 [cit. 2017-06-14]. Dostupné z: <https://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [54] Line Charts and Options. *JqPlot: Charts and Graphs for jQuery* [online]. Leonello, 2011 [cit. 2017-06-14]. Dostupné z: <http://www.jqplot.com/examples/line-charts.php>
- [55] AJAX introduction. *W3Schools: Online Web Tutorials* [online]. Refsnes Data, 2017 [cit. 2017-06-14]. Dostupné z: https://www.w3schools.com/xml/ajax_intro.asp

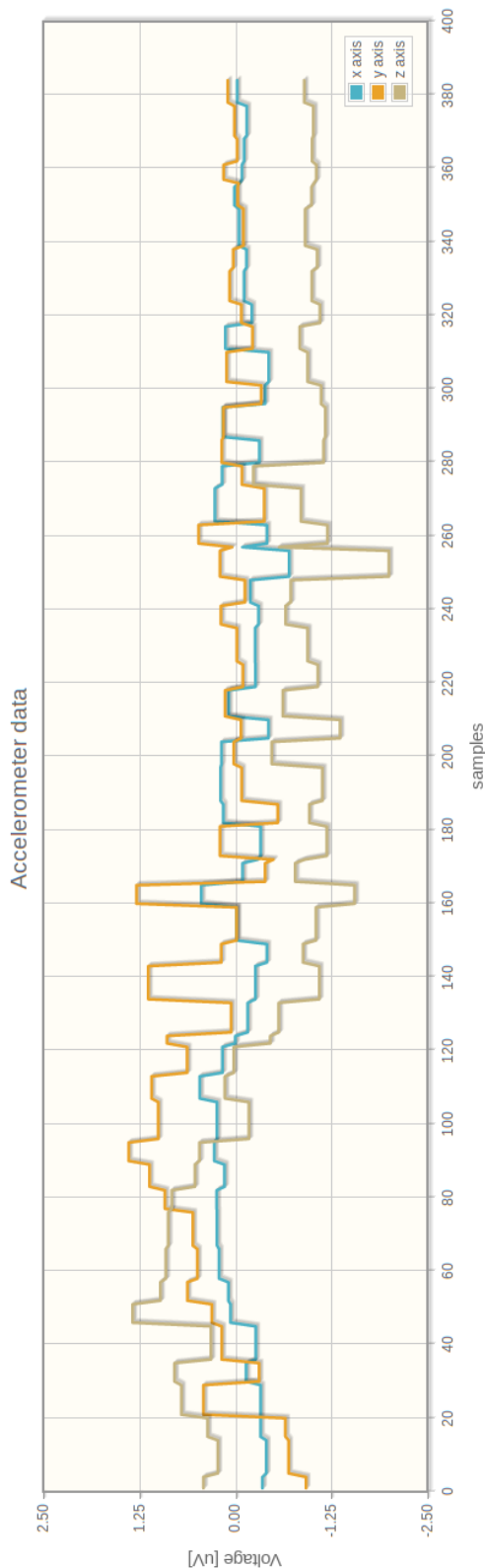
Přílohy

Příloha 1: Příklad měření – data z akcelerometru (klid vs. pohyb)

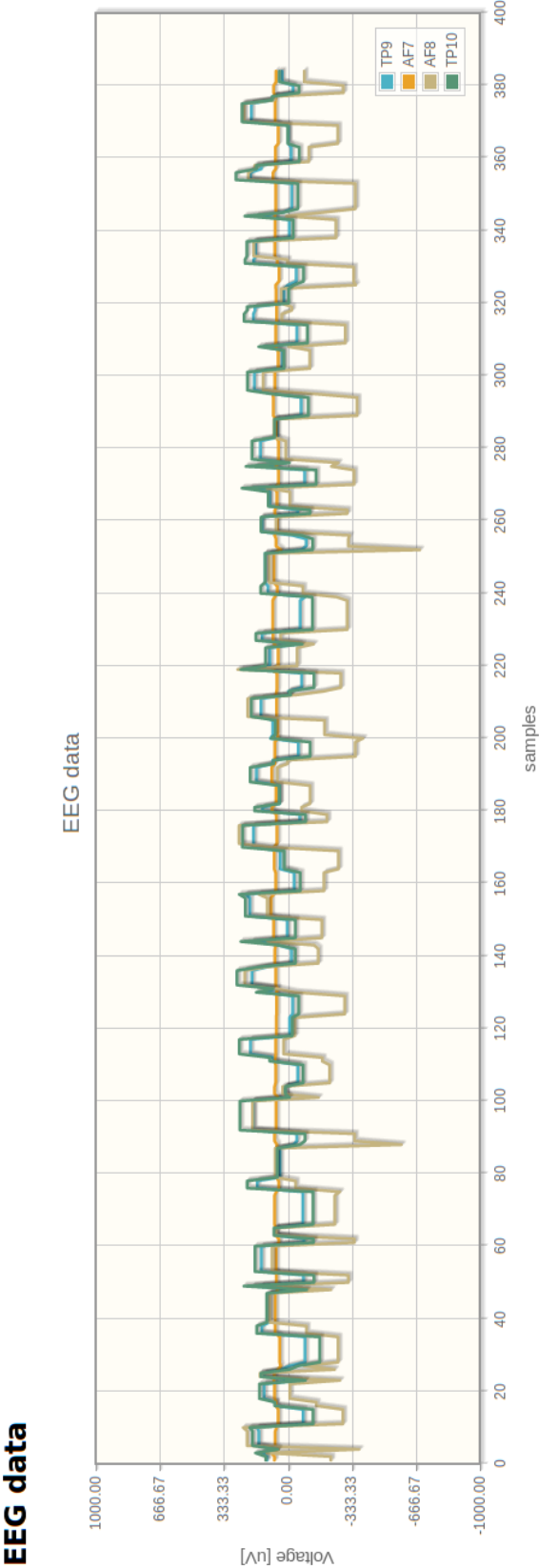
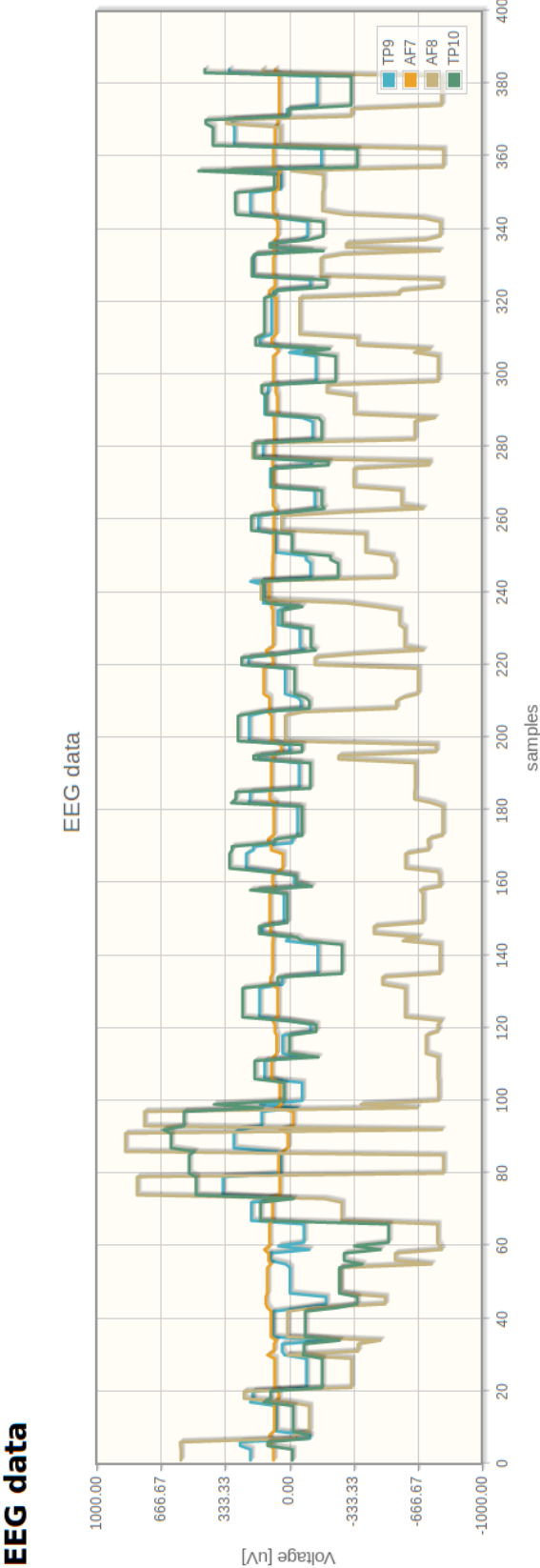
Data z akcelerometru



Data z akcelerometru



Příloha 2: Příklad měření – data z elektrod (před a po ustálení)



Příloha 3: Porovnání parametrů Muse 2014 a 2016 [44 s. 4]

Parametr	Muse 2014	Muse 2016
Bezdrátové připojení	Bluetooth 2.1 s EDR	Bluetooth 4.0 LE
EEG – počet kanálů	4-6	4-5
EEG – vzorkování [Hz]	220 nebo 500	256
EEG – rozlišení [b/snímek]	10 nebo 16	12
Pozice referenční elektrody	FPz (CMS/DRL)	FPz (CMS/DRL)
Pozice měřicích elektrod	TP9, AF7, AF8, TP10	TP9, AF7, AF8, TP10
Baterie	Li-Ion až 5 hodin	Li-Ion až 10h
Nabíjecí čas [h]	2,5	2,5 pomalé nabíjení 1,5 rychlé nabíjení
Materiál frontálních elektrod	Stříbro	Stříbro
Materiál temporálních elektrod	Silikon - pryž	Silikon - pryž
Hmotnost [g]	61	60
Minimální obvod hlavy [cm]	52	52
Maximální obvod hlavy [cm]	60	60
Akcelerometr – počet os	3	3
Akcelerometr – vzorkování [Hz]	500	52
Akcelerometr – rozlišení [b/snímek]	10	16
Akcelerometr rozsah [G]	+/- 2	+/- 4
Gyroskop	Žádný	+/- 1000 °/s
Vstupní rozsah	2mV peak to peak AC	2mV peak to peak AC
Potlačení šumu	Zpětnovazební pasivní elektroda DRL s 2 uV (RMS) šumovou citlivostí	Zpětnovazební pasivní elektroda DRL s 2 uV (RMS) šumovou citlivostí
	50 nebo 60 Hz síťový šum	Žádné potlačení síťového šumu

Muse App kompatibilita	iOS, Android	iOS, Android
Nástroje pro výzkum	MuseIO, MuseLab a Muse Player	Deklarovány ale k dubnu 2017 nevydány
	Pro Windows, Mac OS, Linux	Pro Windows, Mac OS, Linux
LibMuse kompatibilita	iOS, Android	iOS, Android
Počet Micro-USB portů	2	1