

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

# Bakalářská práce

Ověření možnosti realizace jednoduchého stimulátoru  
pro neuroinformatické experimenty na  
platformách Raspberry PI a Cubieboard

Místo této strany bude  
zadání práce.

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 28. června 2018

Radim Panuška

## **Abstract**

The verifying the feasibility of a simple stimulator for neuroinformatic experiments on the Raspberry and Cubieboard platforms. At the Department of Informatics and Computer Science there is research in the field of neuroinformatics. For this research a stimulators to implement ERP experiments are needed. The objective of this thesis is to check the features of new platforms for the creation of a stimulator for ERP experiments. These new platforms offer the opportunity to create a stimulator using one device, all in one. By arguing this option is to design a stimulator according to the requirements of neuroinformatic groups and to implement this stimulator in C/C++. The function of developed application should be verified.

## **Abstrakt**

Na fakultě informatiky a výpočetní techniky probíhá výzkum v oblasti neuroinformatiky. Pro tento výzkum jsou potřeba stimulátory pro provádění ERP experimentů. Cílem této práce je prozkoumat možnosti nových platforem pro vytvoření stimulátoru pro ERP experimenty. Tyto nové platformy nabízejí možnost vytvořit stimulátor s použitím jednoho zařízení, tj. vše v jednom. Pro potvrzení této možnosti je úkolem navrhnout stimulátor podle požadavků neuroinformatické skupiny a implementovat tento stimulátor v jazyku C/C++. Na vytvořeném stimulátoru je potřeba otestovat, zda jeho vlastnosti splňují požadavky kladené na stimulátory pro neuroinformatiku.

## Poděkování:

Rád bych touto cestou poděkoval Ing. Pavlu Mautnerovi, Ph. D. za velmi cenné připomínky, rady, vstřícnost při konzultacích a v neposlední řadě za trpělivost, kterou mi věnoval během vedení mé bakalářské práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
<b>2</b>	<b>Teoretická část</b>	<b>10</b>
2.1	Popis Raspberry PI . . . . .	10
2.2	Popis Cubieboard . . . . .	13
2.3	Popis SW pro vybrané mikropočítače: . . . . .	15
2.4	Srovnání obou platforem: . . . . .	15
2.5	Popis knihovny SDL . . . . .	16
2.5.1	Možnosti SDL . . . . .	16
2.6	Knihovna Wiring PI . . . . .	17
2.6.1	Stručný popis základních funkcí Wiring PI . . . . .	17
2.6.2	Specifické funkce pro Raspberry PI . . . . .	18
2.6.3	Další možnosti a rozšíření WiringPi . . . . .	18
2.7	Elektroencefalografie a ERP . . . . .	19
2.7.1	Elektroencefalograf . . . . .	19
2.7.2	Rozdělení mozkových vln dle frekvence: . . . . .	20
2.7.3	Chyby v EEG signálu: . . . . .	20
2.8	Evokované potenciály: . . . . .	21
2.8.1	Základní ERP komponenty . . . . .	21
2.9	Dostupné open stimulátory a jejich vlastnosti . . . . .	23
2.9.1	PsychoPy . . . . .	23
2.9.2	OpenSesame . . . . .	27
2.9.3	Závěrečné hodnocení . . . . .	31
<b>3</b>	<b>Praktická část</b>	<b>32</b>
3.1	Test vhodnosti RPI + Raspbian + wiringPi pro stimulátor . . . . .	32
3.1.1	Test Raspbian - Raspberry PI 2 . . . . .	33
3.1.2	Řešení nevyhovujícího chování Raspbianu . . . . .	33
3.1.3	Test Raspbian - raspberry ver. 3B+ . . . . .	36
3.1.4	Příprava raspberry PI, úpravy operačního systému . . . . .	37
3.1.5	Výběr platforem pro testování stimulátoru . . . . .	37
3.2	Požadavky na stimulátor . . . . .	38
3.2.1	Rizika při návrhu . . . . .	39
3.3	Návrh stimulátoru . . . . .	40
3.3.1	Návrh grafického uživatelské rozhraní . . . . .	40
3.3.2	Návrh spouštěcího programu . . . . .	42

3.4	Implementace stimulátoru . . . . .	42
3.4.1	Implementace GUI . . . . .	43
3.4.2	Implementace spouštěcího programu experimentu . .	44
3.4.3	Implementace generátoru scénářů . . . . .	46
3.5	Ověření funkčnosti stimulátoru . . . . .	49
3.5.1	Testování délky kritických fází stimulace . . . . .	49
3.5.2	Ověření provedením experimentu . . . . .	50
<b>4</b>	<b>Závěr</b>	<b>53</b>
	<b>Literatura</b>	<b>54</b>
<b>A</b>	<b>Seznam obrázků</b>	<b>55</b>
<b>B</b>	<b>Seznam tabulek</b>	<b>57</b>
<b>C</b>	<b>Příloha A - seznam použitých pinů na GPIO</b>	<b>59</b>
<b>D</b>	<b>Příloha B - stručný návod k aplikaci</b>	<b>60</b>

# 1 Úvod

Od devadesátých let minulého století došlo k velmi výraznému zvýšení v počtu publikací týkajících se neuroinformatických experimentů. Z našeho pohledu je velmi zajímavá metoda zkoumání mozkové aktivity pomocí stimulatoru a EEG. Touto neinvazivní metodou se mohou zabývat i vědečtí pracovníci, kteří nemají medicínské vzdělání. Zároveň i cena technického vybavení pro tyto experimenty je zlomkem ceny vybavení pro jiné metody PET (pozitronová emisní tomografie).

Pro tyto experimenty, kterými se zabývá skupina pracovníků na Katedře informatiky a výpočetní techniky, jsou potřeba stimulatory. V dnešní době již existují volně dostupné softwarové produkty pro tyto stimulatory dostupné pro mnoho platform. Zadání této práce pochází z doby, kdy tyto produkty nebyly ještě tolik rozšířené a bylo zajímavé zabývat se myšlenkou na sestavení vlastního produktu.

Avšak i v dnešní době je stále zajímavé zabývat se jednoduchým produktem, který umožní jednoduché sestavení scénáře experimentu a umožní jeho provedení.

Neuroinformatická skupina na fakultě KIV se zabývala tvorbou stimulatoru na platformě Arduino. Tato platforma velmi dobře splňuje požadavky na měření a prezentaci v reálném čase. Není však možné pracovat s multimediálním obsahem. Kvůli hardwarovým omezením této platformy.

Tato možnost se objevila s příchodem mini/mikro počítačů typu Raspberry nebo Cubieboard. Tyto počítače v sobě spojují vlastnosti klasického počítače, tj. plnohodnotný operační systém Linux, s možností hardwarového připojení signálů na tzv. GPIO port. Výhodou je přímý přístup k pinům tohoto portu.

Nevýhodou těchto počítačů je samozřejmě nižší výkon oproti klasickým stolním počítačům, či notebookům. Vzhledem k nárůstu výkonosti těchto platform v posledních letech, se zde objevuje možnost, že by tyto platformy mohly mít dostatek výkonu k tvorbě a provádění ERP experimentů.

Cílem práce je ověřit tyto možnosti a vyzkoušet je návrhem a vytvořením



jednoduchého stimulatoru v jazyce C, dle požadavků neuroinformatické skupiny. Zároveň je potřeba ověřit, zda stimulator vyhovuje požadavkům kladeným na neuroinformatické stimulatory.

V teoretické části své práce jsem se zabýval teorií týkající se experimentů v neuroinformaci, druhy komponent popsaných u událostmi evokovaných potenciálů. Dále jsem uvedl i možnosti současných volně dostupných stimulatorů.

Praktická část se zabývá návrhem a implementací vlastního jednoduchého stimulatoru a dosažených výsledků při ověřování vhodnosti mikro počítačů pro tuto oblast.

## 2 Teoretická část

### 2.1 Popis Raspberry PI

Jedná se o počítač formátu velikosti kreditní karty. První verze počítače byla vyvinuta nadací Raspberry PI v letech 2006 - 2008. Důvodem k vývoji tohoto počítače byla snaha vytvořit levnou variantu počítače pro výuku práce s počítačem. Pro tento účel byl použit procesor Broadcom BCM2835. Jedná se o procesor původně určený pro settop boxy, u novější verze se jedná o procesor BCM2836 a BCM2837. Dalším ze záměrů bylo vytvořit počítač, na kterém poběží open source operační systém a který bude možno bez větší námahy znovu nainstalovat, pokud během zkoušení a učení dojde k poškození instalace operačního systému.

Již od začátku byl tento nápad velmi kladně přijat širokou komunitou. Výsledkem byl velký zájem a sta tisíce objednávek (k dnešnímu dni byly vyrobeny miliony kusů Raspberry PI). Platforma se nadále vyvíjí a objevují se jak výkonnější varianty (např. Raspberry PI 3B+) tak i menší varianty (např. Raspberry PI Zero). Počítače nacházejí široké uplatnění jako multimediální centra, různé ovládací počítače (například pro řízení dronů, robotů atd.). I když se následně objevily klony (Orange PI, Banana PI), které mají lepší HW parametry, Raspberry vždy vítězí díky masovému rozšíření a z toho vyplývající podpoře.

Raspberry PI existuje v několika různě výkonných verzích. V současné době jsou na trhu verze Raspberry PI 2B+, Raspberry PI 3 a Raspberry PI 3B+.

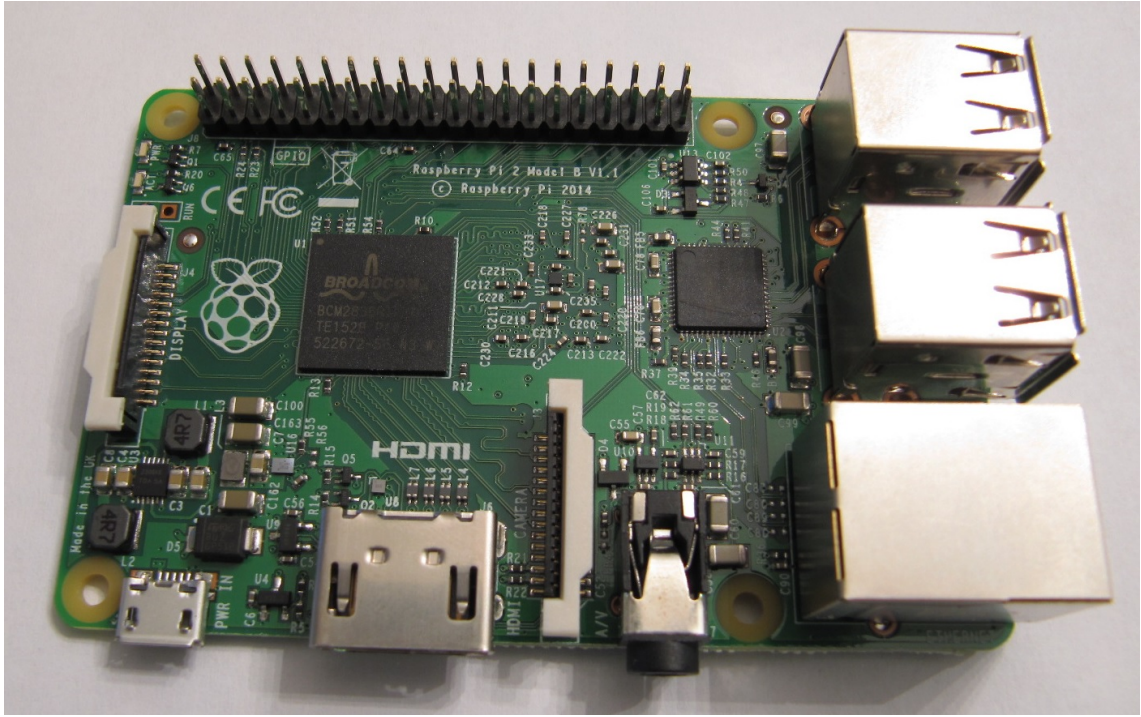
<b>Parametr</b>	<b>Raspberry PI 2</b>	<b>Raspberry PI 3B+</b>
Typ procesoru	BCM 2836	BCM 2837
- frekvence procesoru [MHz]	900	1400
- počet jader procesoru	4	4
Paměť RAM	1 GB DDR2	1 GB DDR2
Rychlost Ethernet rozhraní	100Mbit/s	1GB - (omezení 300Mbit/s)
WIFI	Ne	802.11b/g/n/ac
Bluetooth	Ne	v 4.2
Počet USB 2.0	4	4
Počet GPIO pinů	40	40
Rozhraní pro kameru	Ano	Ano
Rozhraní pro display	Ano	Ano
HDMI výstup	Ano	Ano
Audio out	3.5 mm JACK	3.5 mm JACK
Parametr	85 x 56 mm	85 x 56 mm
Cena	1029 Czk	979 Czk

Tabulka 2.1: Tabulka parametrů Raspberry

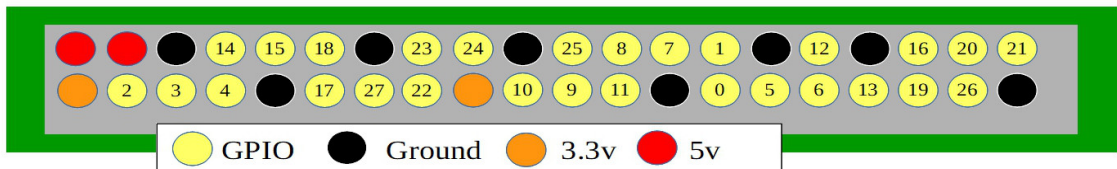
Jak vyplývá ze srovnávací tabulky modely 2 a 3B+ se od sebe liší hlavně typem použitého procesoru a možnostmi bezdrátové komunikace.

Od začátku byl velkou výhodou těchto počítačů konektor GPIO(General Purpose Input/Output)[3]. GPIO pracuje s napětím 3.3 a není 5v tolerantní. Protože je konektor přímo připojen k vývodům procesoru, připojení 5 v vstupu způsobí zničení mikropočítače. Práce s jednotlivými GPIO bude popsána v sekci o knihovně WiringPi.

Mikropočítače jsou dále vybaveny sériovým rozhraním pro display a rozhraním pro připojení kamery, která je dodávána jako příslušenství k Raspberry PI. Tato rozhraní nebudou pro stimulátor použita. Vzhledem k tomu, že novější verze je nyní levnější a výkonnější, doporučuji pro simulátor použít Raspberry PI 3B+. Já provedu testy na obou modelech pro případ, že by někdo měl k dispozici starší verzi. Použití verze Raspberry PI 1 B+ nedoporučuji s ohledem na to, že tato verze nemá dostatečný výkon.



Obrázek 2.1: foto Raspberry PI 2



Obrázek 2.2: Popis GPIO konektoru

<b>Parametr</b>	<b>Cubieboard 1</b>	<b>Cubieboard 2</b>
Typ procesoru	Allwinner A10	Allwinner A20
- frekvence procesoru [MHz]	1000	1000
- počet jader procesoru	1	2
Paměť RAM	1 GB DDR3	1 GB DDR3
Rychlost Ethernet rozhraní	100Mbit/s	100Mbit/s
WIFI	Ne	Ne
Bluetooth	Ne	Ne
Počet USB 2.0	2	2
Počet GPIO pinů	96	96
Rozhraní pro kameru	Ne	Ne
Rozhraní pro display	pomocí GPIO	pomocí GPIO
HDMI výstup	Ano	Ano
Audio out	3.5 mm JACK	3.5 mm JACK
Parametr	104 x 60 mm	104 x 60 mm
Cena	již se neprodává	již se neprodává

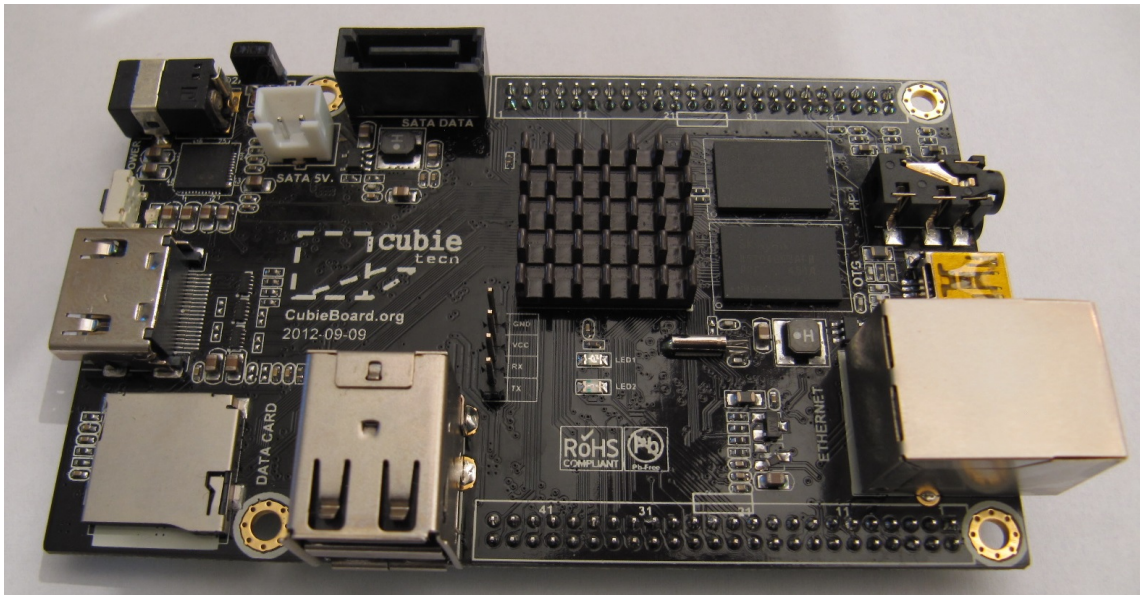
Tabulka 2.2: Tabulka parametrů Cubieboard

## 2.2 Popis Cubieboard

Cubieboard je mikropočítač založený na procesorech Allwinner. Cubieboard je zaměřen na segment HTPC (pc pro domácí kino), proto Cubieboard umožňuje připojení 2.5"pevného disku.

Podrobný popis GPIO konektoru je v příloze 1. GPIO piny pracují s napětím 3.3 V

Výběr verze Cubieboard: podobně jako u Raspberry PI jsem si pro realizaci vybral výkonnější desku, vzhledem k tomu, že méně výkonná deska již není v prodeji. Existují i novější varianty, které však nejsou v ČR k dostání a je nutné je objednat na některém ze zahraničních eshopů(přibližná cena je cca 100USD).



Obrázek 2.3: foto Cubieboard 2

Parametr	Raspberry PI	Cubieboard 2
Portované operační systémy	Raspbian(Debian) Ubuntu Mate Snappy Ubuntu Core Windows 10 IOT CORE OSMC OPENELEC PINET RISC OS	Lubuntu Android 4.2.x Fedora 19 Debian Cubian Embedded Linux
Podpora Open GL instalace SDL	OpenGL ES 1.1/2.0 SDL 2	OpenGL ES 1.1/2.0 SDL 2

Tabulka 2.3: Srovnávací tabulka SW možností Raspberry vs. Cubieboard

## 2.3 Popis SW pro vybrané mikropočítače:

Při dalším zkoumání se již zaměřím pouze na vybrané verze Raspberry PI (uvedené informace se týkají všech verzí od 2B+ a vyšších) a Cubieboard 2.

Jako operační systém pro vytvoření zkušební aplikace jsem vybral Raspbian / Lubuntu. Pro práci s multimediálním obsahem jsem vybral knihovnu SDL 2, protože používá OpenGL ES, které je na RPI ( tuto zkratku budu nadále používat pro Raspberry PI 2) akcelerováno.

## 2.4 Srovnání obou platforem:

RPI výhody:

- velmi široká komunita a rozsáhlá podpora operačních systémů a knihoven
- poloviční cena
- výkonnější procesor

Cubieboard výhody:

- vyšší rychlost přenosu dat v paměti
- rozhraní SATA (pro naše účely nepotřebné)

- větší počet GPIO

Při srovnávání podpory obou mikropočítačů je zřejmá převaha na straně RPI, které je masově rozšířené a komunita kolem tohoto produktu je mnohonásobně větší, než u Cubieboard. Výhody, které má Cubieboard oproti RPI nejsou pro tvorbu simulátoru rozhodující. Jediná relevantní výhoda by mohla být vyšší rychlost přenosu dat v paměti. Proto pro primární vývoj simulátoru použijí RPI.

## 2.5 Popis knihovny SDL

Je zkratkou pro Simple DirectMedia Layer [6], což je multiplatformní knihovna určená pro práci s audio zařízením, klávesnicí, herními ovladači a grafickým hardwarem pomocí OpenGL (OpenGL je rozhraní pro programování aplikací pro práci s 2D a 3D grafickými objekty). SDL knihovna je distribuována pod licencí zlib. Zlib je free licence, která umožňuje volně používat a distribuovat tuto knihovnu. SDL je napsána v jazyku C a umí pracovat i s C++. Dále jsou k dispozici propojení pro několik dalších jazyků, např. C# nebo Python.

### 2.5.1 Možnosti SDL

#### Audio

Knihovna poskytuje funkce pro přehrávání 8-bit a 16-bit audio, mono/stereo/5.1surround. Audio je spuštěno nezávisle ve vlastním vlákně. Knihovna dále poskytuje součást `SDL_mixer` která umožňuje kompletní ovládání audio výstupu.

#### Video

Knihovna může být použita společně s OpenGL nebo samostatně pro 3D grafiku. Dále podporuje akcelerovanou 2D grafiku. U 2D grafiky umožňuje rotace objektů, prolínání obrázků, zvětšování a další. Dále umožňuje vytvářet a spravovat více oken.

#### Detekce vstupních událostí

SDL podporuje vstup z klávesnice, myši, herních ovladačů a multidoteková gesta.



## Ostatní funkce

Protože se jedná o komplexní rozhraní je výčet všech možností velmi dlouhý. Proto uvedu jen některé další:

- podpora vytváření vláken
- podpora struktur pro souběh vláken
- časovače
- přístup k souborů
- zjišťování informací o procesoru

## podporované operační systémy

- Windows - používá Win32 rozhraní
- Mac OS X - používá Cocoa rozhraní a Core Audio pro zvuk
- Linux - používá X11, ALSA, OSS and PulseAudio rozhraní
- iOS - používá UIKit a Core Audio
- Android - používá JNI pro video a zvuk

## 2.6 Knihovna Wiring PI

Jedná se o knihovnu, která umožňuje pracovat s GPIO konektorem [7]. Knihovna je napsána v jazyce C. Podporuje procesory BCM2835, BCM2836 (použit v Raspberry PI2) a BCM2837 (použit v Raspberry PI3). Knihovna je šířena pod GNU licencí. Knihovna se inspirovuje zapojením u platformy Arduino. Knihovna byla rozšířena i na jiné platformy, než je Raspberry. Knihovna je dostupná na GitHubu. Součástí knihovny je i možnost nastavovat GPIO pomocí utilit z příkazové řádky.

### 2.6.1 Stručný popis základních funkcí Wiring PI

- PinMode - nastavuje pro určitý pin typ jeho chování. PIN může fungovat jako vstupní, výstupní nebo PWM.
- pullUpDownControl - nastavuje pull-up nebo pull-down mód na daném PINu.

- digitalWrite - zapisuje hodnoty 0 nebo 1 na určený PIN.
- pwmWrite - zapisuje hodnotu do PWM registru
- digitalRead - vrací hodnotu na určitém PINu.
- analogRead/analogWrite - vyžadují dodatečný analogový modul

### 2.6.2 Specifické funkce pro Raspberry PI

- pwmSetMode/pwmSetRange/pwmSetClock - funkce pro nastavení PWM
- piBoardRev - vrací číslo revize počítače Raspberry PI
- wpiPinToGpio - mapování čísla fyzického pinu na číslo pinu používané WiringPi
- physPinToGpio - mapování čísla fyzického pinu na číslo pinu na P1 konektoru

### 2.6.3 Další možnosti a rozšíření WiringPi

- časovače v milisekundách a mikrosekundách
- funkce pro zastavením běhu programu na určitou dobu
- funkce pro změnu priority programu
- funkce pro práci s vlákny
- knihovna pro práci se seriovým portem
- knihovna pro práci SPI portem
- knihovna pro práci I2C rozhraním
- knihovna pro nastavování PWM na GPIO
- knihovna pro vytváření tónů na GPIO

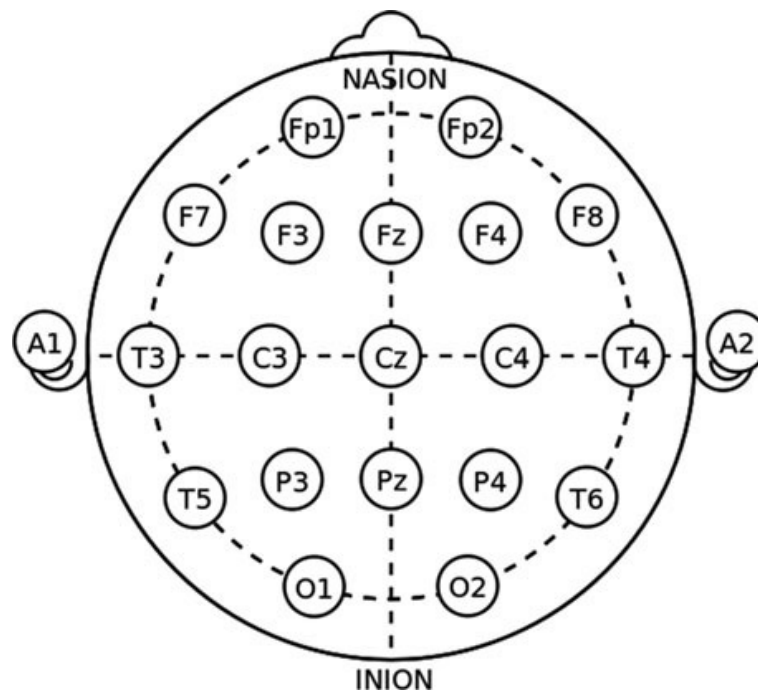
Dále jsou zde rozšíření pro ovládaní displejů a nebo rozšiřující desky gertboard. Tato knihovna je jednou z ukázek rozsáhlé podpory od Raspberry PI komunity.

## 2.7 Elektroencefalografie a ERP

Elektroencefalografie je neinvazivní vyšetřovací metoda používající k vyšetřování mozku změny polarizace neuronů. Záznam těchto změn probíhá pomocí přístroje EEG. Tato metoda byla poprvé představena v roce 1924 psychiatrem Hansem Bergerem a výsledky publikovány v roce 1929[2]. Díky rozvoji výpočetní techniky prošla i elektroencefalografie velkým vývojem. Dnes je používána v psychiatrii pro odhalení chorob jako je epilepsie, encefalitida nebo různých poruch vývoje nervové soustavy u dětí.

### 2.7.1 Elektroencefalograf

Přístroj, který slouží k měření elektroencefalogramu. To je signál, který vznikne zesílením signálu z diferenčních zesilovačů kam je přiváděn signál z jednotlivých elektrod umístěných na kůži lebky vyšetřovaného subjektu. Výsledkem je rozdíl mezi referenční elektrodou, která je umístěna na místě, kde se neočekává žádný EEG signál. Tato elektroda se většinou umísťuje na ušní lalůček. Encefalograf obsahuje soustavu snímacích elektrod, soustavu diferenčních zesilovačů a počítač pro záznam získaného signálu. Soustava elektrod je na lebce rozmístěna v určitém schématu.



Obrázek 2.4: Rozmístění elektrod systém 10-20 [1]

Označení elektrod vychází z jejich umístění na lebce a anglického označení, skupina

- F - frontal - čelní lalok
- C - central - střední skupina
- P - parietal - temenní lalok
- T - temporal - spánkový lalok
- O - occipital - týlní lalok

Lichá čísla značí levou a sudá pravou hemisféru. Elektrody A1 a A2 jsou referenční. Poloha jednotlivých elektrod je na lebce fixována pomocí čepice. Snímané signály z kůže jsou o velikosti mikrovoltů. Z tohoto důvodu je nutné zajistit dokonalý přenos signálu mezi kůží a elektrodou. K tomu napomáhá elektrovodivá pasta. Signály se zracovávají každý zvlášť, tj. každému kanálu přísluší zesilovač.

### **2.7.2 Rozdělení mozkových vln dle frekvence:**

- Delta vlny 0 - 3,5 Hz - vyskytují se v hlubokém spánku a v dětství
- Delta vlny 4 - 7 Hz - vyskytují se v během usínání
- Delta vlny 8 - 13 Hz - vyskytují se v bdění se zavřenýma očima
- Delta vlny 14 - 30 Hz - vyskytují se při duševní aktivitě

### **2.7.3 Chyby v EEG signálu:**

Vzhledem k tomu, že zaznamenáváme změny napětí a amplitudou o velikosti desítek mikrovoltů, je měření velmi náchylné k poruchám. Tyto poruchy mohou způsobit problémy při dalším zpracování dat během vyhledávání evokovaných potenciálů. Chyby nazýváme artefakty.

Příčiny artefaktů:

- od vnějších zdrojů elektrického napětí - interference s nimi
- od vyšetřovaného subjektu - mrknutí, pohyb očí
- od EEG zařízení

Většinu artefaktů má tvar odlišný od mozkové aktivity a jsou proto v záznamu rozeznatelné. Před další zpracování signálu je potřeba je zohlednit.

## 2.8 Evokované potenciály:

V angličtině je používána zkratka ERP (event related potentials) jsou odezvy mozku na vnější podněty odvozené z EEG záznamu. Komponenty můžeme z EEG záznamu odvodit buď prostým průměrováním nebo složitějšími metodami. První ERP záznam byl publikován v roce 1939. V té době kvůli absenci pokročilé výpočetní techniky byly možnosti výzkumu velmi omezené. K výraznějšímu pokroku došlo až roku 1964, kdy byla popsána CNV (contingent negative variation) ERP komponenta. Tato komponenta byla objevena při experimentu, kdy testovanému subjektu byl vyslán varovný signál ve dvou různých časových intervalech před target stimulem. Na záznamu bylo identifikováno rozsáhlé záporné napětí na předních elektrodách, které bylo vyhodnoceno jako známka přípravy subjectu na target stimul.

Tento objev podnítil další výzkum a další rok byla objevena P3 komponenta. Následujících patnáct let byly objeveny mnohé další komponenty. Jak ukazuje graf v [erp Tutorial] počet publikací se v dalších letech významně zvyšoval až na téměř 6 tisíc v letech 2007-2008.

### 2.8.1 Základní ERP komponenty

#### Zrakové komponenty

##### C1 - zraková

Objevuje se na P elektrodách. Její polarita se může měnit. je generována v oblasti V1 (primární vizuální cortex). Je záporná pro stimuly v horní oblasti vidění a kladná pro stimuly ve spodní oblasti. Spojuje se většinou s kladnou P1 komponentou. Začíná typicky 40-60mS po stimulu a vrcholí 80-100 mS po stimulu. Je velmi citlivá na kontrast a prostorovou frekvenci stimulu.

##### P1 - zraková

Následuje po C1 komponentě a částečně ji překrývá. Začíná 60 - 90 ms a vrcholí 100 - 130 ms po stimulu. Její zpoždění závisí na kontrastu stimulu. Stejně jako C1 je citlivá na změny ve stimulu.

##### N1 - zraková

Následuje po P1 komponentě. Existuje několik N1 vizuálních subkomponent. Nejdříve vrcholí 100 - 150 ms po stimulu na anterior elektrodách a následují dvě N1 komponenty s typickým vrcholem 150 - 200 ms po stimulu. Jedna na parietálním kortexu a druhá z laterálního occipital kortexu. Obě

komponenty jsou ovlivněny prostorovou poyorností subjektu. Navíc laterální ccipital subkomponenta je větší, když subjekt vylučuje stimuly namísto jejich hledání.

## **P2 - zraková**

Následuje po N1 vlně na anterior a centrálních elektrodách. Je větší u target stimulů a je umocněna pokud jsou target stimuly poměrně občasně. Anterior P2 komponenta má tuto vlastnost podobnou jako P3 komponenta. Na rozdíl od P3 komponenty tento úkaz u anterior P2 efekt nastává pro jednoduché stimuly.

## **N170 a kladný vertex potenciál**

Reakce na tvář způsobuje negativnější potenciálem než reakce na stimul bez tváře. Vyskytuje se nad pravou hemisférou. Vrcholí v průměru 170 ms po stimulu.

## **Sluchové komponenty**

Některé komponenty se vyskytují 10 ms po stimulu. Nazývají se BERs(brainstem evoked responses) nebo ABRs(auditory brainstem responses). Jsou velmi užitečné pro vyšetřování sluchových onemocnění u kojenců.

## **N1 - sluchová**

Stejně jako u vizuální komponenty má i sluchová řadu subkomponent. Přední centrální komponenta vrcholí 75 ms a je způsobována sluchovým kortexem. Potenciál vertexu vrcholí 100 ms a třetí pocházející laterální částí vrcholí 150 ms po stimulu. N1 kopomenta je citlivá na pozornost.

## **MMN - sluchová**

MMN(mismatch negativity) - vyskytuje se u subjektů které byli vystaveny řadě stejných stimulům, která je občas narušena rozdílným stimulem. Například řada tónů se stejnou frekvencí, která je občas přerušena tónem s jinou frekvencí. Objevuje se na centrální střední čáře a vrcholí 160 - 220 ms po stimulu. Vyskytuje se, když je subjekt stimulům vystaven při jiné činnosti.

## **Komponenta typu N2**

Je vyvolána opakujícím se nontarget stimulem. Je umocněna občasným presentováním jiných stimulů v opakující se sérii.

### **P3**

Je vyvolána neočekávanou změnou frekvence nebo hlasitosti tónu. Známe dva druhy této komponenty a to P3a a P3b. P3b je někdy též označována jako P300 komponenta. Amplituda této komponenty je umocněna soustředění subjektu na úkol. Zvyšuje se také spolu se složitostí úkolu. Objevuje se o target stimulu. Doba výskytu závisí na složitosti úkolu, který subjekt řeší.

### **N400**

Jedná se o komponentu spojenou s řečí. Záporná vlna v oblasti centrálního a parietální kortexu. Je vyvolána po objevení se nečekaného slova ve větě. Zpravidla ke konci věty. Bývá také vyvolána nějakým druhem nečekané změny, např. při vykreslování čáry.

## **2.9 Dostupné open stimulátory a jejich vlastnosti**

V posledních letech se podařilo portovat na Raspberry PI otevřené softwareové stimulátory. Jedná se o OpenSesame a PsychoPY. Oba produkty lze na Raspberry nainstalovat z repozitáře najednou pomocí příkazu :

```
apt-get install opensesame
```

Jedná se v obou případech o rozsáhlé softwareové balíky, proto zde uvedu pouze jejich stručný popis.

### **2.9.1 PsychoPy**

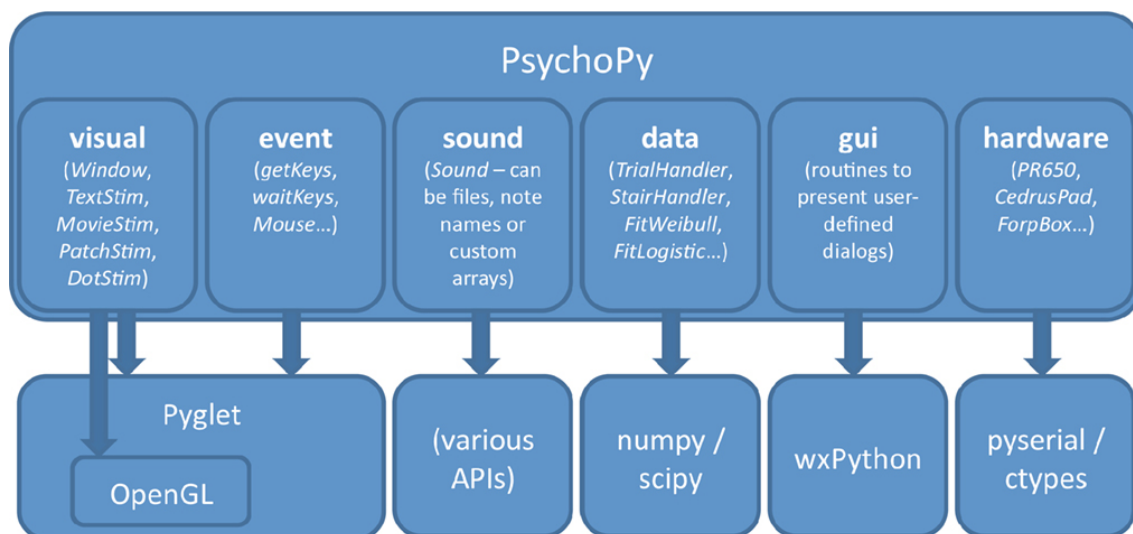
Jedná se o knihovnu napsanou v jazyce Python. PsychoPy podporuje OpenGL, aby bylo možné pomocí něho psát velké množství různých experimentů. PsychoPy je vyvíjeno od roku 2003 a je aktivně podporováno, tj. uživatelé mohou hlásit chyby a navrhopvat různá zlepšení. Silnou stránkou tohoto řešení je to, že je založeno na jazyce Python a umožňuje psát skripty pro ERP experimenty v tomto jazyku. Zároveň je možné používat i knihovny tohoto jazyka nezávisle na platformě, jako jsou pyglet, wxPython a jumpy. Výhodou tohoto jazyka je i silná podpora pro analýzu dat. Proto je možné provádět i všechny analýzy získaných výsledků.

Další z výhod je možnost použití hardwarově akcelerované grafiky. Toto umožňuje tomuto produktu používat ke stimulaci nejenom statické stimuly, ale i animace či videosekvence. Zároveň jsou možné i odezvy na pohyby myši. Stimuly mohou být ovlivněny předchozí odpovědí testovaného subjektu.

Hardwarová přenositelnost je hlavním důvodem, proč jsem informace o PsychoPy zařadil do mé práce. Umožňuje vyzkoušet tento stimulátor na Raspberry PI. Mimo této platformy je možné PsychoPy spustit i na dalších platrofmách jako jsou Windows, Mac a další. PsychoPy je dodáváno spolu s integrovaným prostředím PsychoPyIDE. Toto prostředí má pomoci nováčkům s tvorbou prvních experimentů a odstítnit je od nutnosti instalace jazyka Python a potřebných knihoven.



Produkt obsahuje toto uspořádní modulů:



Obrázek 2.5: PsychoPy modules [5]

**PsychoPY podporuje tyto vizuální stimuly:**

### Windows

Stimuly je možné zobrazovat v celoobrazovkovém režimu, nebo v okně. Produkt podporuje i vícemonitorové uspořádání, tj. na každém monitoru může být umístěné jiné okno. Toto umožňuje zobrazovat binokulární stimuly. To znamená zobrazení rozdílných stimulů pro každé oko. Pro zobrazování je použita technika dvojího bufferování. To znamená, že stimul je v prvním kroku vykreslen do skrytého okna a potom přepnut do aktivního okna. To umožňuje velmi rychlé zobrazení stimulu.

### PatchStim

Jeden z nejpoužívanějších stimulů, umožňuje zobrazovat obrázky nebo různé textury. Podporuje velké množství operací s obrázky jako jsou rotace, posuny nebo prolínání.

### TextStim

Slouží k zobrazování textu. Umožňuje používat vyhlazování nebo jakékoliv fonty nainstalované na prezentačním počítači. Je kompatibilní s Unicode, tj. podporuje i naše národní znaky.

## **Sound**

Umožňuje jednoduchým způsobem vytvářet a přehrávat zvuky. Nabízí buď tvorbu tónů pomocí frekvence a doby trvání, nebo přehrávání souborů typu mpg nebo wav. Přehrávání je možné i ve formátu stereo, což umožňuje provádět celou řadu audio experimentů.

## **DotStim**

Jedním z možných stimulů je náhodné pole teček. Pozice jednotlivých teček se mohou měnit. Pozice teček se mohou měnit podle různých pravidel. Některé mohou stát, zatímco jiné se pohybují v různých směrech.

## **Reakce na odpovědi**

Pro experimenty, které potřebují zaznamenávat odpovědi od testovaného subjektu, PsychoPy podporuje širokou řadu vstupů, jako jsou klávesnice, myš či různé herní ovladače, nebo zvláštní zařízení. Je možné zaznamenávat stisk jednotlivých kláves, nebo pozici kurzoru myši a stisku jejích tlačítek.

## **Vstupně/výstupní zařízení**

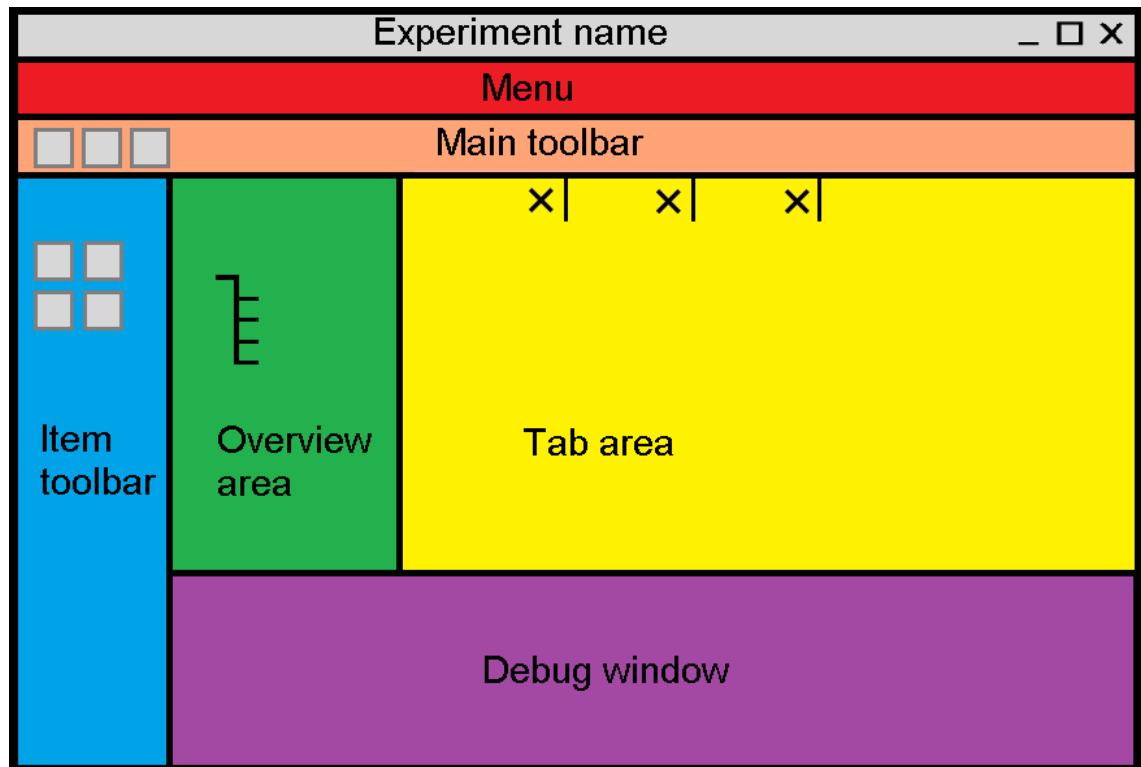
Produkt umožňuje jednoduše přistupovat k USB zařízením, nebo paralelnímu portu pomocí emulátorů. Paralelní port je často používán k zasílání kódů do měřicího EEG počítače. Tyto kódy potom slouží k rozpoznání typu stimulu, který je právě zobrazován, aby mohla být naměřená reakce správně vyhodnocena. PsychoPy podporuje i různé komunikační protokoly pro zařízení používána pro měření.

## **Časování při experimentu**

Jak již bylo řečeno, PsychoPy se snaží používáním doublebufferingu co nejvíce zkrátit dobu zobrazování stimulu na obrazovce. Zároveň je ošetřen i souběh ostatních vláken, která čekají na dokončení zobrazování stimulu a jsou tak synchronizována. V těchto případech velmi pomáhá zpracování obrazových dat procesorem grafické karty. Zobrazovací frekvence je závislá na zobrazovacím zařízení a výkonu grafické karty. Může se pohybovat až přes 100Hz.

## 2.9.2 OpenSesame

Jedná se o další z open source produktů. Poslední verze byla vydána 5. března 2018. Podobně jako PsychoPy podporuje programování v jazyce Python. Obsahuje integrované IDE pro návrh experimentů.



Obrázek 2.6: Základní schéma IDE openSesame [4]

Stručný popis jednotlivých částí:

- Overview are - obsahuje jednotlivé kroky experimentu. Po kliknutí na určitý krok, se jeho detaily otevřou v oblasti zvané Tab area.
- Tab area - detail u vybranému kroku experimentu
- Item toolbar - ikony s výběrem jednotlivých komponent a také s nej-používanějšími komponentami

Tento produkt má propracovaný návrhář experimentů , který podporuje tyto velké množství komponent. Jedná se zejména o:

- průběhy a struktury
- odpovědi

- prezentace stimulů
- formuláře
- sledování pohybu očí
- podpora jazyka Python
- pokročilé funkce
- podpora mnoha zařízení

### **Průběhy a struktury**

Produkt podporuje tvorbu nezávislých proměnných. Dále je možné použít pseudonáhodný generátor. Možnost vytváření cyklů, sekvencí včetně jejich přerušování. Pro pseudonáhodný výběr je možné náhodně třídit řádky nebo sloupce vnitřních tabulek, které jsou potom použity jako podklad pro experiment. Proměnné je možné načítat ze souborů .xlsx nebo csv. Tok instrukcí je možné řídit buď sekvenčně, nebo paralelně. To je užitečné například na čtení odpovědí z klávesnice během promítání stimulu.

### **Odpovědi**

je možné načítat z těchto zařízení:

- klávesnice
- myš
- joystick
- tlačítkový ovladač
- SR ovladač
- nahrávání zvuku

U klávesnice je možné určit správné odpovědi, možné odpovědi a názvy tlačítek. Dále je možné nastavovat timeout pro odpověď.

Podobně jako u klávesnice je i u myši možné nastavit hodnoty pro tlačítka. Dále správné a povolené odpovědi. A v neposlední řadě je možné nastavit timeout. Navíc oproti klávesnici je možné nastavit oblasti a souřadnice zájmu pro zaznamenávání polohy kurzoru myši.

Joystick podporuje pohyb os nebo trackballu. Navíc je zde implementována

detekce pohnutí.

Jako vstupní zařízení je možné použít mnoho tlačítkových ovladačů. Je zde možnost emulace stisků kláves klávesnice nebo emulace joysticku.

Specialitou je SR vstupní zařízení, které bylo přímo vyvinuto pro sběr odpovědí v psychologických experimentech. Originální verze má 5 tlačítek a 5 světel. K připojení se používá sériový port. V dnešní době většinou USB sběrnice s emulací seriového portu.

U nahrávání zvuku je možné nastavit pouze začátek a konec nahrávání.

## Prezentace stimulů

je možné vybírat z těchto stimulů:

- textové stimuly
- vizuální stimuly
- zvukové stimuly
- video

Textové stimuly umožňují HTML formátování. Dále volbu fontu ze skupiny cca 7 fontů, nastavení velikosti a stylu.

Vizuální stimuly se dělí na dvě skupiny:

- statické obrázky
- položky se zpětnou vazbou

Základní rozdíl je v tom, že statické obrázky se připravují před spuštěním experimentu, zatímco položky se zpětnou vazbou se připravují během experimentu. S tím je nutné počítat při návrhu, protože vygenerování položek se zpětnou vazbou je pomalejší. Nedoporučuje se položky se zpětnou vazbou používat pro časově kritické experimenty.

Zvukové stimuly podporují přehrávání souborů typu .wav nebo .ogg. Parametry umožňují například nastavení doby přehrávání bez ohledu na skutečnou délku souboru. Dále je k dispozici základní syntetizér, který umožňuje volbu tvaru vlny, čas náběhu na plnou hlasitost, balance mezi levým a pravým kanálem.

Video přehrávání je založeno na přehrávači VLC. Pokud není nainstalovaný je možné plugin stáhnout ze stránek Opensesame.

## **Formuláře**

Formuláře slouží k získávání odpovědí z dotazníků. Umožňují klasické checkboxy, obrázky, tračítka s obrázkem, vstup textu, nebo známkovací stupnici.

## **Sledování pohybu očí**

Je založeno na Python knihovně PyGaze. Podporovaná zařízení jsou EyeLink a EyeTribe. EyeLink potřebuje instalaci dalšího software. Podporované funkce jsou kalibrace, čekání na mrknutí a další.

## **Podpora jazyka Python**

Umožňuje doplnění experimentů pokud nejsou k dispozici požadované funkce přes grafické rozhraní. V tomto případě je možné doplnit do experimentu kód jazyka Python.

## **Podpora backend systémů**

Backend systémem rozumíme softwareovou vrstvu, která pracuje se vstupními nebo výstupními zařízeními. Díky podpoře jazyka Python můžeme využívat celou řadu jeho knihoven. OpenSesame je nezávislý na backendech a to umožňuje výběr některého z backendů.

V současné době jsou podporovány tyto systémy:

- legacy
- psycho
- droid
- xpyriment

Backend se dá zvolit ve vlastnostech experimentu.

## Podpora systému Android

Experimenty je možné spouštět na zařízeních se systémem Android. Podporovaná jsou zařízení s verzí Androidu 2.2 a vyšší. V tomto případě je vyžadovaným backend systémem droid. Nedají se používat zařízení synth a sampler.

### 2.9.3 Závěrečné hodnocení

U obou systémů se jedná o velmi pokročilé systémy s širokými možnostmi. Podmínkou pro plné využití těchto systémů je znalost programování v jazyku Python. I když lze tyto systémy provozovat na platformě Raspberry, není to úplně bez problémů. Produkt PsychPy si stěžoval při inicializačním testu na latency paměťové karty. Testy PsychoPy pak končily chybou.

Navíc jsem nikde nenašel popsanou podporu pro GPIO. To může znamenat, že produkty je možné provozovat na Raspberry, ale nejsou plně využity jeho výhody k nimž GPIO nesporně patří.

## 3 Praktická část

### 3.1 Test vhodnosti RPI + Raspbian + wiringPi pro stimulátor

Operační systému typu Linux mezi něž patří i Raspbian mají prediktivní plánování úloh a obecně nejsou vhodné pro realtime aplikace. Prediktivní plánování úloh se vyznačuje tím, že jádro operačního systému provádějící simultanní multitasking pomocí přidělování času jednotlivým procesům v určitých cyklech, pozastavuje a opětovně spouští procesy bez jejich "vědomí".

Protože stimulátor bude mimo vysílání stimulů i zaznamenávat reakci na stimul je potřeba, aby řešení splňovalo určité požadavky na dobu odezvy během měření. Existuje zde riziko, že stimulátor v sekce čekání na odezvu bude přerušen a opět spuštěn v době kdy odezva již nebude na vstupu stimulátoru. Tím bychom tuto odezvu nezachytili.

Dále je zde riziko nedodržení požadované přesnosti na změření doby odezvy, pokud by proces měřící odezvu byl uspán na dobu delší než 1 milisekunda. Stejně riziko platí i pro proces nastavující kód na GPIO. Pokud dojde k zastavení programu v místě, kde je spuštěno zobrazování stimulu, může dojít k pozdnímu nastavení kódu stimulu na GPIO, což povede ke zkreslení měření intervalu odezvy na EEG. Provedl jsem sérii měření, abych zjistil v jakém rozsahu bude kolísat doba provádění stejného kódu při opakovaném provádění.

Testování jsem provedl jednoduchým programem pomocí použití funkce , která měří čas v mikrosekundách a zjišťuje, jako dlouho trval cyklus padesáti přečtení stavu určitého GPIO pinu. Po té je čas změřen a zařazen do jednoho z rozsahu časů ve výsledné tabulce. Tabulka ilustruje jaké je riziko, že proces bude během odečítání reakce z GPIO odstaven na příliš dlouhou dobu. Všechna měření byla prováděna při počtu opakování zkušebního cyklu 200 mil. Doba trvání testu byla 600 sec.



Délka běhu v mikrosekundach	1. měření CPU bez zátěže	2. měření CPU bez zátěže	3. měření 50% zátěž CPU	4. měření 100% zátěž CPU
< 10	199 mil.	199 mil.	199 mil.	199 mil.
10 - 25	35.9 tis.	35.1 tis.	263.2 tis.	88.7 tis.
25 - 50	10.4 tis.	9.0 tis.	10.4 tis.	10.1 tis.
50 - 100	485	718	981	1 422
100 - 250	47	2	681	819
250 - 500	6	26	323	1 753
500 - 1 000	23	13	8	138
1 000 - 2 000	2	0	12	177
2 000 - 4 000	3	4	26	167
4 000 - 8 000	3	0	10	16
8 000 - 16 000	1	0	5	213
16 000 - 32 000	0	0	1	2
> 32 000	0	0	0	0

Tabulka 3.1: Test odezvy na OS Raspbian - Raspberry PI 2

### 3.1.1 Test Raspbian - Raspberry PI 2

Výsledky testování:

Zátěž byla vytvořena jednoduchým programem s nekonečným cyklem. Rozbor naměřených hodnot: naměřené hodnoty ukazují, že aplikace není schopna zaručit měření s přesností 1mS, protože v určitém počtu případů překročí doba měření 1 ms.

### 3.1.2 Řešení nevyhovujícího chování Raspbianu

Možné varianty a jejich výsledky:

- použití upraveného jádra Linuxu pro real-time aplikace (jádro Linuxu je soubor programů, které jsou spuštěny v chráněném režimu a zajišťují provádění operací s vstupně výstupními zařízeními atd.)
- vyhrazení jádra procesoru pro běh aplikace
- kombinace obou

Délka běhu v mikrosekundach	1. měření CPU bez zátěže	2. měření CPU bez zátěže	3. měření 50% zátěž CPU	4. měření 100% zátěž CPU
< 10	196 mil.	196 mil.	196 mil.	196 mil.
10 - 25	474 tis.	3.88 mil.	2.87 mil.	2.88 mil.
25 - 50	3.98 mil.	46.3 tis.	879 tis.	815 tis.
50 - 100	32.3 tis	9.01 tis.	14.5 tis.	14.3 tis.
100 - 250	3.30 tis.	122	35.2 tis.	39.2 tis.
250 - 500	2	3	1.12 tis.	780
500 - 1 000	1	0	0	6
1 000 - 2 000	1	0	5	2
2 000 - 4 000	0	0	37	46
4 000 - 8 000	0	0	27	17
8 000 - 16 000	0	0	341	317
16 000 - 32 000	0	0	13	19
maximalni doba (v mikrosekundach)	6	26	30.1 tis.	30.1 tis.

Tabulka 3.2: 1. upravené jádro Linuxu pro real-time aplikace RPI 2

Délka běhu v mikrosekundach	1. měření CPU bez zátěže	2. měření CPU bez zátěže	3. měření 50% zátěž CPU	4. měření 100% zátěž CPU
< 10	199 mil.	199 mil.	199 mil.	199 mil.
10 - 25	43.1 tis.	43.1 tis.	43.6 tis.	43.2 tis.
25 - 50	976	995	582	859
50 - 100	267	269	242	261
100 - 250	0	0	0	0
250 - 500	0	0	0	0
maximalni doba (v mikrosekundach)	78	26	81	83

Tabulka 3.3: 2. vyhrazené jádro procesoru pro běh aplikace RPI 2

<b>Délka běhu v mikrosekundach</b>	<b>1. měření CPU bez zátěže</b>	<b>2. měření CPU bez zátěže</b>	<b>3. měření 50% zátěž CPU</b>	<b>4. měření 100% zátěž CPU</b>
< 10	199 mil.	199 mil.	199 mil.	199 mil.
10 - 25	183	196	192	214
25 - 50	42.6 tis.	42.8 tis.	42.1 tis.	42 tis.
50 - 100	1 667	1 768	1 959	2 072
100 - 250	0	0	0	0
250 - 500	0	0	0	0
maximalni doba (v mikrosekundach)	82	82	84	89

Tabulka 3.4: vyhrazené jádro procesoru RT kernel Linuxu RPI 2

Délka běhu v mikrosekundach	1. měření CPU bez zátěže	2. měření CPU bez zátěže	3. měření 50% zátěž CPU	4. měření 100% zátěž CPU
< 10	199 mil.	199 mil.	199 mil.	199 mil.
10 - 25	3 359	2 172	4 326	3 229
25 - 50	1171	477	526	861
50 - 100	260	88	97	187
100 - 250	43	7	85	104
250 - 500	15	1	23	47
500 - 1 000	1	7	12	65
1 000 - 2 000	1	8	20	21
2 000 - 4 000	0	0	6	57
4 000 - 8 000	1	0	3	16
8 000 - 16 000	0	1	1	322
16 000 - 32 000	0	0	0	20
> 32 000	0	0	0	0

Tabulka 3.5: normální běh aplikace RPI 3B+

### 3.1.3 Test Raspbian - raspberry ver. 3B+

Výsledky testování:

Zátěž byla vytvořena stejným způsobem jako pro raspberry PI 2

Rozbor naměřených hodnot:

I když jsou hodnoty naměřené bez zatížení výrazně lepší než hodnoty u Raspberry PI 2, stále ani s touto výkonnější verzí není aplikace schopna zaručit měření s přesností 1mS, protože v určitém počtu případů překročí doba měření 1 ms.

Řešení číslo 1, tj. upravené jádro Linuxu pro real-time aplikace tzv. RT kernel, nebylo v době zkoušení dostatečně stabilní. Protože stabilita je velmi důležitá pro výsledek experimentu, vyloučil jsem pro Raspberry PI 3B+ tuto variantu. Ze stejného důvodu byla vyloučena i varianta 3, tj. vyhrazené jádro s použitím RT kernelu. Později by obě varianty mohly být vyzkoušeny, pokud k tomu bude důvod.

Diskuze naměřených hodnot:

Raspberry PI3B+ díky novějšímu procesoru a zvýšené frekvenci vykazuje lepší výsledky přibližně o jednu časovou skupinu. Stejně jako u Raspberry PI 2 jsem vybral variantu vyhrazených dvou jader pro běh experimentu a měření odezvy.

Délka běhu v mikrosekundach	1. měření CPU bez zátěže	2. měření CPU bez zátěže	3. měření 50% zátěž CPU	4. měření 100% zátěž CPU
< 10	199 mil.	199 mil.	199 mil.	199 mil.
10 - 25	1790	4025	939	808
25 - 50	277	522	157	149
50 - 100	3	10	5	3
100 - 250	0	0	0	0
250 - 500	0	0	0	0
maximalni doba (v mikrosekundach)	97	70	61	61

Tabulka 3.6: 2. vyhrazené jádro pro běh aplikace RPI 3B+

### 3.1.4 Příprava raspberry PI, úpravy operačního systému

Pro správnou funkci aplikace je nutno udělat nastavení v konfiguraci operačního systému, aby byla dvě jádra procesoru vyhrazena pro spouštění experimentu. Toto se provede úpravou souboru `/boot/cmdline.txt` přidáním parametru

`isolcpus =2,3`

Toto nastavení zabrání operačnímu systému používat třetí a čtvrté jádro.

### 3.1.5 Výběr platformem pro testování stimulátoru

Jak je vidět v teoretické sekci, platforma cubieboard nenabízí takovou podporu jako produkty Raspberry. Od doby vzniku zadání této práce navíc došlo k výraznému zhoršení dostupnosti této platformy v České republice.

Když jsem hledal možnosti pořízení novějších modelů cubiebord, našel jsem pouze jeden e-shop, který uvádí na svých stránkách cca 4 roky starou verzi cubieboard 2, ale není možné ji zakoupit. Novější verze nejsou v českých nabízené.

Z těchto důvodů jsem se rozhodl pokračovat v testování na dvou různých typech Raspberry PI, z nichž ten novější je k dostání v několika e-shopech. Starší typ znamenal velký posun ve využitelnosti oproti svému předchůdci a

je dostupný v bazarech. Navíc jeho vybavení ukazovalo, že by měl dostačovat na provádění experimentů.

## 3.2 Požadavky na stimulátor

Na vytvořeném stimulátoru bude možno provádět ERP experimenty se zvukovými a statickými obrazovými stimuly s frekvencí maximálně 10 stimulů za sekundu. Experimenty budou mít předem daný pevný scénář a nebude možno reagovat na odezvy subjektu. Nebude možné použít ani animací či videa.

### Vstup:

- obrazový a zvukový stimul
- požadovaná pravděpodobnost stimulu
- druh stimulu - target/nontarget
- délka stimulu
- délka přestávky po stimulu
- kód stimulu, který bude nastaven na GPIO
- doba po kterou nebude měřena odpověď

Požadavky na výstup stimulátoru: vysílání vizuálních a zvukových stimulů s frekvencí 1snímek/sec. Vyslání vizuálního stimulu bude realizováno pomocí standardního monitoru, zvukové stimuly budou realizovány pomocí sluchátek/výstupu pro sluchátka. Po každém vyslání stimulu budou nastaveny určené GPIO piny na stav high (3.3 v). Dále bude spuštěno měření doby případné reakce s přesností 1ms.

Ze zadání vyplývá, že výsledná aplikace musí být schopná pracovat s multi-mediálními soubory a s určenou přesností schopna měřit dobu odezvy. Testy potvrdily možnost využít standardní instalaci Raspbianu(verze Linuxu určená pro RPI vycházející s distribuce Debian), to umožňuje pro uživatelské rozhraní použít X server.

Cílem je navrhnout a vytvořit stimulátor řádově jednodušší než jsou konkurenční produkty, které většinou používají pro tvorbu scénářů různé programovací jazyky vlastní definice. Protože ne každý vědec v oboru neurologie ovládá programování v těchto pseudojazycích bude navrhovaný stimulátor scénář generovat podle pevného modelu na základě zadaných údajů.

### 3.2.1 Rizika při návrhu

#### **Problém:**

Jak bylo výše zmíněno, operační systém pro dané hardwareové platformě je založen na operačním systému linux, který používá prediktivní plánování procesů po časových kvantech. Nevýhodou tohoto uspořádání je, že časová kvanta mohou překročit dobu 1 milisekundy, což je požadovaná přesnost a v tomto případě nelze zaručit dodržení požadované přesnosti.

#### **Řešení:**

- použití reálnového kernelu pro operační systém - řešení realtime systému není příliš vhodné vzhledem k omezenému výběru těchto systémů pro raspberry a také vzhledem k omezené podpoře multimédií, síťových technologií a ovladačů pro danou platformu v porovnání s operačním systémem raspbian.
- spuštění experimentu a měření odezvy na dvou rezervovaných jádrech procesoru

### **Problém:**

Nelze sestavit libovolný experiment. Pevná struktura generátoru experimentů má pouze omezenou možnost na ovlivnění výsledku pomocí vstupních parametrů.

### **Řešení:**

Nemá řešení, je to daň za jednoduchost stimulatoru. V případě potřeby by bylo možné aplikaci rozšířit o další typy generátorů scénářů. Kód pro spouštění scénáře může zůstat stejný. Pokud by bylo nutné řídit experiment pomocí odpovědí uživatele, bude nutné změnit i kód sloužící pro běh experimentu.

## **3.3 Návrh stimulatoru**

Stimulator bude rozdělen na dvě základní části:

- grafické uživatelské rozhraní
- program pro spuštění experimentu

Měření odezvy a délky měření za účelem kontroly dodržení požadovaných parametrů bude realizováno separátním procesem, který se bude spouštět na vyčleněném jádru procesoru.

### **3.3.1 Návrh grafického uživatelského rozhraní**

Grafické uživatelské rozhraní bylo navrhováno s ohledem na příslušnost jednotlivých údajů k částem experimentu, tj. parametry platné pro celý experiment jako jsou:

- název souboru
- časový interval pro dead zone - časový interval ve kterém není odpověď považována za platnou, ale za náhodnou, protože k odpovědi dochází v čase, kdy lidský mozek není ještě schopen reagovat. Jedná se s největší pravděpodobností o náhodnou odpověď a nebo o opožděnou odpověď na předchozí stimul.
- časový interval pro kód stimulu - čas po který je na výstupním portu počítače vysílán uživatelem definovaný kód stimulu



Tyto informace jsou obsaženy v horní třetině hlavní obrazovky. Střední část obrazovky je tvořena tabulkou, která obsahuje jednotlivé stimuly/skupiny stimulů používané v experimentu. Pro zjednodušení práce byla navržena tato adresářová struktura pro jednotlivé experimenty:

- kořenový adresář obsahuje jednotlivé vygenerované scénáře pro daný experiment a výsledky spuštěných experimentů
- podadresáře obsahují obrazový a zvukový soubor, oba soubory mají stejný název. Očekávané typy souborů jsou BMP pro obrázek a WAV pro zvuk.
- adresář obsahující nontarget stimuly (je pouze jeden pro celý experiment) může obsahovat více souborů.

Ukázky obrazove se nachází v příloze D v dokumentaci k aplikaci.

Jednotlivé řádky tabulky je možné měnit pomocí tlačítek:

1. EDIT - umožní editaci aktivního řádku
2. INSERT - umožní vložení nového řádku
3. DELETE - maže aktivní řádek

Úprava položek scénáře bude probíhat v samostatném formuláři. Po jeho vyplnění se přehledová tabulka aktualizuje. Řádky jsem seřadil dle položky target sestupně, což zajistí řazení třídy nontarget stimulů až na konec. Toto je očekávané v návrhářích scénářů.

Přehledová tabulka bude obsahovat tyto položky:

1. druh stimulu - Y = target stimul, N = nontarget stimul
2. adresář obsahující obrazový a zvukový stimul
3. pravděpodobnost se kterou se má být stimul obsažen ve scénáři
4. doba trvání stimulu (v milisekundách)
5. délka pauzy následující po stimulu (v milisekundách)
6. vertikální souřadice obrazového stimulu, 0 je nahoře
7. horizontální souřadice obrazového stimulu, 0 je vlevo
8. hlasitost přehrávání zvukového stimulu

Následuje sekce pro generování/spouštění scénáře.

1. název souboru pro vygenerování scénáře
2. tlačítko pro vygenerování scénáře
3. tlačítko pro spuštění scénáře

Spodní sekce je určena pro operace spojené s definicí experimentu.

1. tlačítko pro otevření definice existujícího experimentu
2. tlačítko pro založení definice nového experimentu
3. tlačítko pro uložení experimentu

Zde vlož obrázky aplikace a udělej popis GUI.

### 3.3.2 Návrh spouštěcího programu

Bude se jednat se o samostatný program napsaný v jazyce c s použitím knihoven SDL 2.0 a Wiring PI. Program používá pouze jedno vlákno, aby nedocházelo k nežádoucím souběhům během spouštění experimentu. Funkcí program je načtení scénáře, jeho spuštění a zaznamenávání odezvy uživatele. Vstupem je soubor s danou logikou, který byl vytvořen v GUI pomocí návrháře scénářů.

Při návrhu této části aplikace je nutné brát ohled na jednoduchost a rychlý běh programu. I když je raspberry PI2 taktováno na frekvenci 900 MHz, tak vzhledem k ARM procesoru není jeho výkon srovnatelný s procesory používanými v noteboocích, či stolních počítačích(např. Intel I3, I5 ..). Proto je kód programu navržen pro běh na dvou jádrech procesoru a optimalizován na vyžití dob určené pro čekání na ukončení stimulu nebo na ukončení pauzy mezi stimuly.

Program bude očekávat jeden vstupní parametr a to název souboru se scénářem experimentu. Název výstupních souborů je generován automaticky. Soubory jsou ukládány do adresáře, kde je uložen scénář experimentu.

## 3.4 Implementace stimulátoru

Vytvoření stimulátoru jsem dělal v prostředí Code::Blocks 10.05 na raspberri PI 2.0. Grafickou podobu jsem vytvořil v prostředí QT verze 4.8. Prostředí Code::Blocks 10.05 je součástí distribuce operačního systému Raspbian. Prostředí QT verze 4.81 je možné nainstalovat z repositáře.

### 3.4.1 Implementace GUI

GUI obsahuje tyto třídy: Addialog - třída slouží k zadávání podrobností pro jednotlivé stimuly, nebo, v případě non target stimulů, pro jejich skupinu. Tato třída je zároveň použita pro editování údajů p stimulech. Třída obsahuje definici get a set funkcí pro práci s jednotlivými prvky formuláře. Protože jsou všechna data obsažena v položkách formuláře nejsou třeba žádné interní proměnné.

Main - třída slouží pouze pro vytvoření objektu aplikace, vytvoření a zobrazení hlavního okna. Na závěr třída spouští aplikaci.

MainWindow - jedná se o hlavní okno aplikace, které poskytuje všechna data o experimentu a umožňuje spustit scénář. Protože cele zpracování (s výjimkou provádění scénáře) probíhá v této metodě nejsou potřeba žádné metody pro vstup a výstup dat. Rozhraní proto obsahuje pouze události týkající se stisknutí tlačítek na formuláři.

ExperimentItem - obsahuje datovou strukturu používanou pro generování scénáře pro spuštění experimentu. Třída obsahuje tyto vlastní proměnné:

- stimulNr - číslo stimulu
- pictureIndex - indexové číslo obrázku
- posX - horizontální pozice umístění stimulu na obrazovce
- posY - vertikální pozice umístění stimulu na obrazovce
- soundFile - název souboru se zvukem
- stimuluCode - řetězec obsahující kód stimulu, který je nastaven na GPIO
- gpioOutputInterval - doba po kterou je stimuluCode prezentován na GPIO
- soundVolume - hlasitost přehrávání zvukového stimulu
- intervalMilisec - doba trvání stimulu
- pauseMilisec - doba mezi stimuly
- deadzoneMilisec - doba po kterou by odpověď neměla být vyhodnocována

- pictureFile - název souboru s obrazovým stimulem
- target - příznak Y/N, zda se jedná o target stimul

Všechny proměnné jsou private a typu QString. Vzhledem k velikosti scénářů a tomu, že výstup je realizován do textového souboru, není potřeba optimalizovat typ jednotlivých položek podle jejich skutečného rozsahu. Proměnné jsou přístupné pouze pomocí public metod get a set.

GUI dále obsahuje dva formuláře, které přísluší třídám QDialog a MainWindow.

### 3.4.2 Implementace spouštěcího programu experimentu

V programu jsou použity tyto struktury pro reprezentaci dat:

#### test\_polozka

položka slouží pro vytvoření jednosměrného seřazeného seznamu položek spouštěcího scénáře. Obsahuje tyto proměnné:

- INT step\_nr - obsahuje číslo porádku kroku, je důležitá pro spojení stimulu a odpovědi ve výstupním souboru
- INT stimul\_nr - číslo stimulu v rámci množiny stimulů
- INT picture\_index - slouží k propojení s množinou obrazových stimulů
- INT pos\_x - X pozice obrazového stimulu na obrazovce
- INT pos\_y - Y pozice obrazového stimulu na obrazovce
- CHAR[150] sound\_file - název souboru se zvukem
- CHAR[8] stimul\_code - kód stimulu ve dvojkové soustavě
- INT sound\_volume - hlasitost zvukového stimulu
- INT interval\_milisec - doba trvání stimulu
- INT pause\_milisec - pauza mezi stimuly
- INT deadzone\_milisec - doba kdy odpověď není považována za platnou
- INT output\_code - ??
- INT input\_code - očekávaná odpověď

- INT gpio\_output\_interval - doba nastavení kódu stimulu na GPIO
- test\_polozka \*next - ukazatel na další položku scénáře

### **pictures\_polozka**

položka obsahuje data potřebná pro pole obrazových stimulů načtených knihovnou SDL. Obsahuje tyto proměnné:

- INT picture\_nr - číslo obrázku
- CHAR[150] picture\_file - název souboru s obrazovým stimulem
- INT picture\_index - slouží k propojení s množinou obrazových stimulů

Program obsahuje tyto části:

- sekci načtení scénáře spuštěného experimentu
- vytvoření jednotlivých stimulů na plátno
- zobrazení stimulu
- výpočet zbývajících času do konce zobrazení stimulu
- vytvoření procesu pro měření odezvy
- pauza mezi stimuly a její výpočet

Jednotlivé funkce a obsah těchto částí je podrobně vysvětlen v části týkající se realizace.

### **Načtení scénáře spuštěného experimentu a načtení stimulů na plátno**

Během zkoušení prototypu jsem zjistil, že načtení obrazového stimulu před jeho zobrazením trvá až 30 milisekund. Toto omezuje použití stimulátoru, proto jsem přistoupil k optimalizaci kódu, která spočívá v načtení všech obrázků před spuštěním experimentu. Proto má soubor dvě sekce, první obsahuje seznam všech obrazových stimulů a jejich indexu.

Druhá část obsahuje scénář ve kterém jsou použity indexy obrazových stimulů. Tímto se zkrátila doba potřebná k provedení jednoho kroku scénáře a to umožní případné zkrácení doby trvání stimulu, tj. stimulátor je schopen zobrazovat více stimulů za jednu sekundu.

## Zobrazení stimulu

Probíhá funkcí knihovny SDL 2.0. Doba trvání je cca .

## Výpočet zbývajících času do konce zobrazení stimulu

pro dodržení délky trvání jednotlivých stimulů a přestávek mezi nimi jsem zohlednil čas spotřebovaný na zobrazení stimulu a nahrání zvuku do funkce na přehrávání při výpočtu času čekání na uběhnutí celkové doby stimulu.

## Vytvoření procesu pro měření odezvy

je realizování pomocí funkce fork(). Tento proces je spuštěn na dedikovaném jádře procesoru. Cílem je předejít případným výpadkům procesu, který měří odezvu, při náhodném nárůstu zatížení ostatními procesy operačního systému. Odezva je měřena cyklickým odečítáním stavu určených pinů GPIO. Během odečítání jsou měřeny doby běhu jednotlivých cyklů a zjišťováno, zda nebyla překročena požadovaná odezva.

## pauza mezi stimuly a její výpočet

dobu pauzy mezi stimuly jsem se snažil využít pro přípravu dalšího stimulu. Proto je výsledná pauza, stejně jako u doby trvání stimulu, vypočtena se zohledněním trvání těchto operací.

## Měření odpovědi - zhodnocení

Z naměřených dat vyplývá, že při stisknutí mikropínače dochází k zámkům, které se jeví jako opakované stisknutí s velmi krátkou periodou (cca 1 milisekunda). Tyto technicky způsobené odpovědi je možné softwarově odfiltrvat po ukončení experimentu. Jako maximální hodnotu pro počet stisků za sekundu jsem nastavil 50 stisků. To znamená, že odpovědi, jejich doba od poslední odpovědi je kratší než 50 milisekund budou vymazány.

### 3.4.3 Implementace generátoru scénářů

S ohledem na požadovanou jednoduchost a nízké zatížení procesoru počítače, je celý scénář detailně vygenerován před spuštěním na základě hodnot zadaných přes GUI při definici experimentu. Generátor vytváří scénář podle pevných pravidel se zohledněním zadaných parametrů.

Tato pevná pravidla jsou:

- target stimuly nikdy nejsou dva za sebou
- mezi každými dvěma target stimuly je náhodné množství nontarget stimulů

Tato pravidla není možné ovlivnit nastavením parametrů. Pro porušení těchto pravidel je nutné úprava kódu generátoru.

Funkce generátoru:

- příprava dat pro generování scénáře
- výstup první sekce souboru
- cyklus generování stimulů

### **Příprava dat pro generování scénáře**

V této části dochází ke kompletaci všech dat potřebných pro scénář. Pro potřeby přípravy dat jsem nadefinoval třídu "experimentitem" která slouží k uchování hodnot potřebných pro generování obou částí scénáře. Na začátku je potřeba projít všechny scénáře a vyhledat obrazové stimuly.

K vyhledávání je použita funkce QDir ze které jsou pomocí vlastnosti entryList filtrovány pomocí přípon. Adresář s nontarget stimuly je zařazen na konec seznamu položek. Pokud se adresář s nontarget stimuly dostane na vstup dříve, je vynechán, jeho pozice v tabulce stimulů uložena a použita po projití všech řádek tabulky.

### **Výstup první sekce souboru**

Zde je uložen seznam všech stimulů s jejich indexem a příznakem target. V další sekci souboru jsou místo názvů souboru již použity pouze jejich indexy. Příznak target je později použit pro kontrolní výpočet procent zastoupení všech stimulů v adresáři. Tyto hodnoty umožňují kontrolu, zda při generování nedošlo k chybě.

### **Cyklus generování stimulů**

Scénář je generován pomocí cyklu, který má stejný počet opakování jako je celkový počet target stimulů. Při každém průběhu se počítají hodnoty určující rozsah pro náhodné číslo určující kolik bude nontarget stimulů.

Nejdříve si vypočteme střed intervalu pro výběr náhodného počtu nontarget stimulů:

$$\text{NonTargetPivot} = \text{Počet target stimulů} / \text{počet nontarget stimulů}$$

Dále potřebujeme vypočítat poloměr intervalu, který jsem zvolil jako polovinu středu zaokrouhlenou na celá čísla dolů.

### **Příklad:**

Celková pravděpodobnost target stimulů = 20%

Pravděpodobnost nontarget stimulů = 80%

Střed intervalu = 4

Poloměr intervalu = 2

Interval pro počet nontarget stimulů = <3 , 6>

K určení náhodného čísla je volána funkce, která používá systémové zařízení /dev/urandom. Toto číslo je následně mapováno na potřebný rozsah pomocí dělení modulo velikostí rozsahu. Výběr target stimulu probíhá z množiny stimulů všech tříd. Počet stimulů v jednotlivých třídách se po výběru třídy snižuje. To zajistí, že v případě opakovaného výběru jedné třídy se snižuje její zastoupení v celkovém součtu a tím i pravděpodobnost jejího dalšího výběru.

Náhodnost je nutná pro vyloučení možnosti, že by si sledovaný subjekt osvojil frekvenci target stimulů a odpovídal na základě očekávání namísto skutečného pozorování. Například, pokud by mezi target stimuly byly vždy tři non target stimuly a subjekt měl odpovídat ANO/NE na otázku je to target stimul, dojde během prvních několika desítek stimulů k naučení této frekvence a subjekt bude odpovídat každý čtvrtý stimul ANO, aniž by sledoval jaký stimul je zobrazený.

Toto můžu potvrdit, z vlastní zkušenosti, kdy jsem tento jev na sobě pozoroval při experimentu, kdy náhodný počet nontarget stimulů měl malé rozpětí. V mém případě to bylo 3-5 nontarget stimulů a po prvních několika minutách jsem měl snahu předpovídat jaký stimul přijde namísto soustředění se na skutečný stimul.



Měřený stimul	Raspberry PI 2 OS Wheezy verze 7 v[ms]	Raspberry PI 2 OS Stretch verze 9 v[ms]	Raspberry PI 3B+ OS Stretch verze 9 v[ms]
doba prvního stimulu	60	12	8 mil.
doba ostatních stimulů	38	10	6

Tabulka 3.7: Doba zobrazení obrazového stimulu

### Zhodnocení fungování generátoru scénářů

Zhodnocení není úplně jednoduché, přesnost v daném kroku je silně ovlivněna rozpětím počtu nontarget stimulů. To je nyní v kódu nastaveno na +- polovinu střední hodnoty průměrného počtu nontarget stimulů. Pokud máme možný počet nontarget stimulů 3-7 stimulů je možná odchylka pro požadovanou hodnotu při 100. stimulu +-2

## 3.5 Ověření funkčnosti stimulátoru

Ověření funkčnosti bylo provedeno v první řadě pomocí experimentu a dále pomocí testování délky trvání jednotlivých úseků běhu programu stimulátoru. Pro správnou funkci stimulátoru je důležitá délka odezvy v určitých fázích stimulace. Testování bylo provedeno na dvou verzích Raspberry PI. Jednalo se o starší Raspberry PI 2 a dále o nejnovější dostupnou verzi Raspberry PI 3B+.

### 3.5.1 Testování délky kritických fází stimulace

Jedná se o fáze jako je:

- doba zobrazení obrazového stimulu
- doba od konce zobrazení do nastavení kódu na GPIO konektoru
- doba odečítání stavu odpovědi na GPIO

#### Doba zobrazení obrazového stimulu

Jedná se o nejproblematictější úsek. Je nejnáročnější na hardware. V přiložené tabulce jsou popsány časy získané opakovaným měřením z různých dnů.

Měřený stimul	Raspberry PI 2 OS Wheezy verze 7 [mikrosekundach]	Raspberry PI 2 OS Stretch verze 9 [mikrosekundach]	Raspberry PI 3B+ OS Stretch verze 9 v [mikrosekundach]
rozsah	7 - 64	5 - 6	2 - 4

Tabulka 3.8: Doba od konce zobrazení do nastavení kódu na GPIO konektoru

Nejlépejších výsledků dosáhlo Raspberry PI 3B+. Zajímavý je vliv použitého operačního systému. Se starší verzí operačního systému dosahovalo Raspberry PI 2 hodnot, které již nebyly vyhovující pro měření experimentů. S novějším systémem jsou hodnoty zcela vyhovující. Rozdíl u nového systému mezi Raspberry PI 2 a Raspberry PI 3B+ odpovídá rozdílu frekvencí procesorů.

Při dalším testování jsem, ale narazil na problémy se stabilitou tohoto nového systému. Pro plnohodnotné využívání stimulatoru je potřeba odhalit příčinu těchto potíží. Na starší verzi systému byla aplikace zcela stabilní.

### Doba od konce zobrazení do nastavení kódu na GPIO konektoru

U všech měření se tato doba pohybovala hluboko pod požadovanou 1 milisekundou.

### Doba odečítání stavu odpovědi na GPIO

Tato hodnota je nejméně důležitá. Přesto je v průběhu programu kontrolováno překročení 1 ms. Pro tuto hodnotu nebyly prováděny žádné testy. Během zkoušení aplikace se tato hodnota u nejpomalejší varianty pohybovala do 100 mikrosekund.

## 3.5.2 Ověření provedením experimentu

Ověření bylo provedeno pomocí experimentu pro zachycení P3 komponenty. Měření proběhlo na Katedře informatiky a výpočetní techniky v učebně UU403. Z technických důvodů bylo toto testování provedeno pouze na modelu Raspberry PI 2.

### Použité pomůcky:

- stimulátor na raspberry PI 2
- zvukotěsná kabina

- EEG měřicí počítač
- snímací čepice

### **Popis testovacího scénáře**

Pro tento experiment byl použit scénář se dvěma target stimuly a skupinou nontarget stimulů. Jako stimuly byla použita písmena. Testovaný subjekt měl na úkol počítat target stimuly.

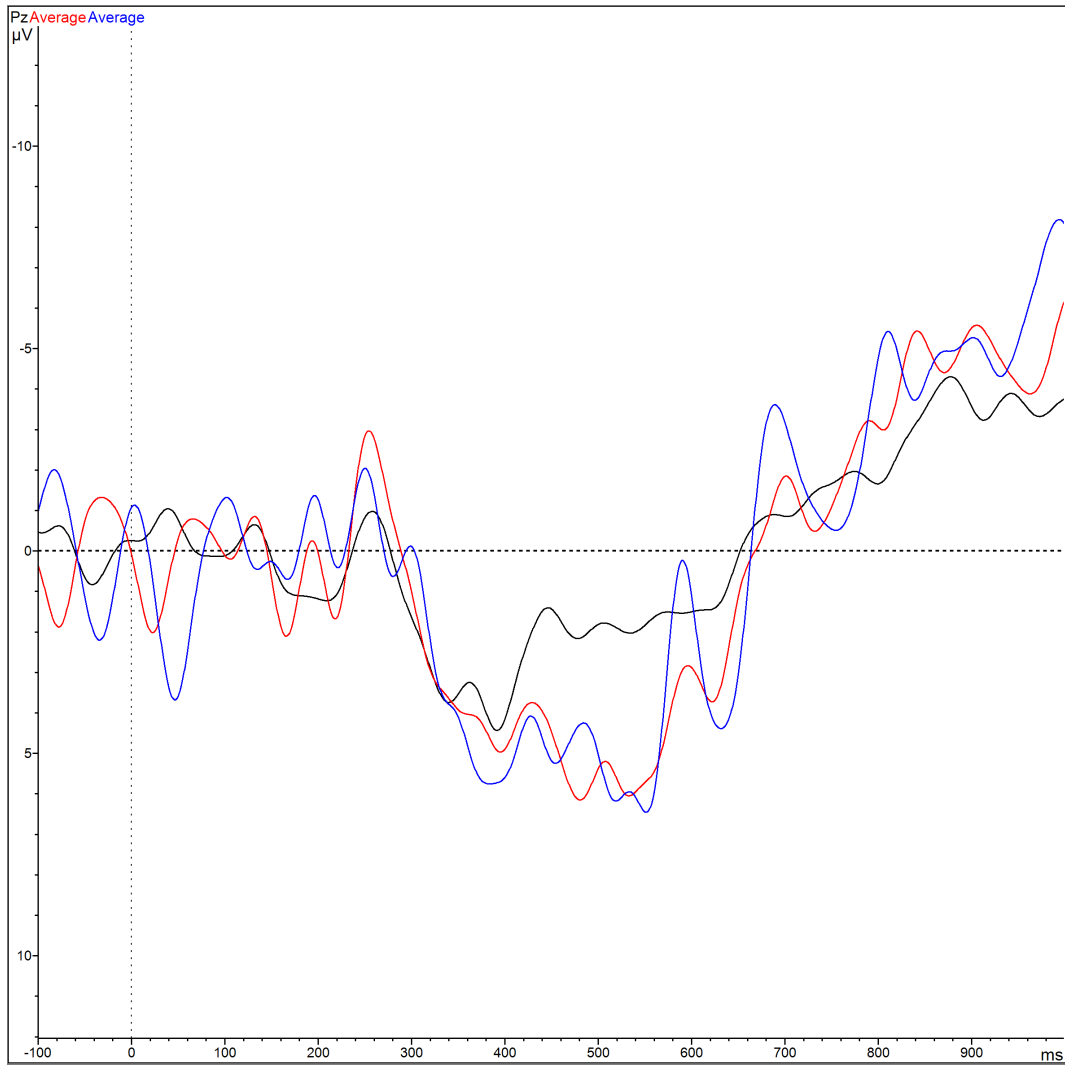
### **Nastavení stimulů:**

- 10% první target stimul
- 20% druhý target stimul
- 70% skupina nontarget stimulů

Konkrétní scénář byl vygenerován generátorem scénářů. Stimulátor každou z těchto skupin označí na výstupu jiným kód, aby bylo možno v záznamu EEG rozlišit okamžik, kdy došlo k promítnutí stimulu subjektu.

Výsledkem měření bylo zobrazení P300 komponenty a jejích subkomponent. Měření sloužilo pouze pro ověření technické funkce stimuluátoru. Proto bylo provedeno pouze jedno měření.

- Popis grafu:
- černá je standardní stimul - nejčetnější
  - červená deviantní stimul - 2. nejčetnější vlna P3b vrcholí někde kolem 480 ms
  - modrá - distraktor - nejméně četný stimul



Obrázek 3.1: zachycení komponenty P3

## 4 Závěr

Během své bakalářské práce jsem získal základní znalosti z neuroinformatiky a neuroinformatických experimentů. Dále jsem se seznámil s novými možnostmi, které nabízejí nové platformy na trhu mikropočítačů. Vyzkoušel jsem si i práci v prostředí QT, které nebylo vyučováno během mého studia.

Prvotní cíl, tj. ověřit vhodnost nových platform mikropočítačů pro neuroinformatické experimenty a ověřit jejich schopnosti na vlastním stimulátoru se podařilo splnit. Problémem zůstává nízká stabilita nových verzí operačního systému, která se ukázala jako nevhodnější, vzhledem k dobré podpoře hardwareové akcelerace grafiky.

Vzhledem k velkému počtu uživatelů a dobré podpoře platformy Raspberry PI, věřím, že tyto nedostatky se podaří v brzké době odstranit.

Platforma Cubieboard se ukázala jako nevhodná, vzhledem ke své nízké dostupnosti. Navíc poslední verze této platformy se více zaměřují na produkty v oblasti domácího kina a přesouvají se do oblasti počítačů pro domácí kina. To má za následek opouštění rozhraní GPIO u těchto počítačů. Pro naše experimenty je GPIO zásadní, protože umožňuje vytvořit stimulátor na jediném zařízení.

Dosažené výsledky splňují požadavky na stimulátory v oblasti neuroinformatiky a ukazují, že je za jistých okolností používat Raspberry PI jako stimulátor.

Do budoucna by bylo možné rozšířit možnosti navrženého stimulátoru o další schémata experimentů, aby se podařilo pokrýt co nejvíce druhů experimentů. Dále by bylo vhodné navrhnout systém šablon pro jednotlivé typy experimentů.

# Literatura

- [1] *Neuronal Correlation Parameter and the Idea of Thermodynamic Entropy of an N-Body Gravitationally Bounded System* [online]. Ioannis Haranas, 2017. [cit. 2018/06/24]. Picture 10-20 system. Dostupné z: [https://www.researchgate.net/figure/international-10-20-system-for-EEG\\_fig19\\_320121487](https://www.researchgate.net/figure/international-10-20-system-for-EEG_fig19_320121487).
- [2] LUCK, S. J. *An Introduction to the Event-Related Potential Technique(COGNITIVE NEUROSCIENCE)*. The Mit Press, 2005. ISBN 978-0262621960.
- [3] *Raspberry PI Documentation* [online]. [cit. 2018/05/24]. Raspberry PI,Wikipedia. Dostupné z: <https://www.raspberrypi.org>.
- [4] *OpenSesame Documentation* [online]. OpenSesame. [cit. 2018/06/26]. OpenSesame. Dostupné z: <http://osdoc.cogsci.nl/>.
- [5] PEIRCE, J. W. *Generating stimuli for neuroscience using PsychoPy* [online]. Nottingham Visual Neuroscience, School of Psychology, University of Nottingham, Nottingham, UK, 2009. [cit. 2018/06/25]. PsychoPy article. Dostupné z: <https://www.frontiersin.org/articles/10.3389/neuro.11.010.2008/full>.
- [6] *Simple DirectMedia Layer* [online]. [cit. 2018/06/27]. SDL Documentation. Dostupné z: <https://wiki.libsdl.org/FrontPage>.
- [7] *OpenSesame Documentation* [online]. [cit. 2018/06/23]. WiringPi. Dostupné z: <http://wiringpi.com/>.

# A Seznam obrázků

# Seznam obrázků

2.1	foto Raspberry PI 2 . . . . .	12
2.2	Popis GPIO konektoru . . . . .	12
2.3	foto Cubieboard 2 . . . . .	14
2.4	Rozmístění elektrod systém 10-20 [1] . . . . .	19
2.5	PsychoPy modules [5] . . . . .	25
2.6	Základní schéma IDE openSesame [4] . . . . .	27
3.1	zachycení komponenty P3 . . . . .	52
D.1	Hlavní okno aplikace . . . . .	60
D.2	Okno pro zadávání a úpravy stimulů . . . . .	61



## B Seznam tabulek

# Seznam tabulek

2.1	Tabulka parametrů Raspberry . . . . .	11
2.2	Tabulka parametrů Cubieboard . . . . .	13
2.3	Srovnávací tabulka SW možností Raspberry vs. Cubieboard .	15
3.1	Test odezvy na OS Raspbian - Raspberry PI 2 . . . . .	33
3.2	1. upravené jádro Linuxu pro real-time aplikace RPI 2 . . . .	34
3.3	2. vyhrazené jádro procesoru pro běh aplikace RPI 2 . . . .	34
3.4	vyhrazené jádro procesoru RT kernel Linuxu RPI 2 . . . . .	35
3.5	normální běh aplikace RPI 3B+ . . . . .	36
3.6	2. vyhrazené jádro pro běh aplikace RPI 3B+ . . . . .	37
3.7	Doba zobrazení obrazového stimulu . . . . .	49
3.8	Doba nastavení kódu na GPIO . . . . .	50

# C Příloha A - seznam použitých pinů na GPIO

## Vstupní PINY

`digitalRead(7)*16+digitalRead(12)*8+digitalRead(16)*4+digitalRead(20)*2+digitalRead(21);`

- PIN7 - hodnota odpovědi 16
- PIN12 - hodnota odpovědi 8
- PIN16 - hodnota odpovědi 4
- PIN20 - hodnota odpovědi 2
- PIN21 - hodnota odpovědi 1

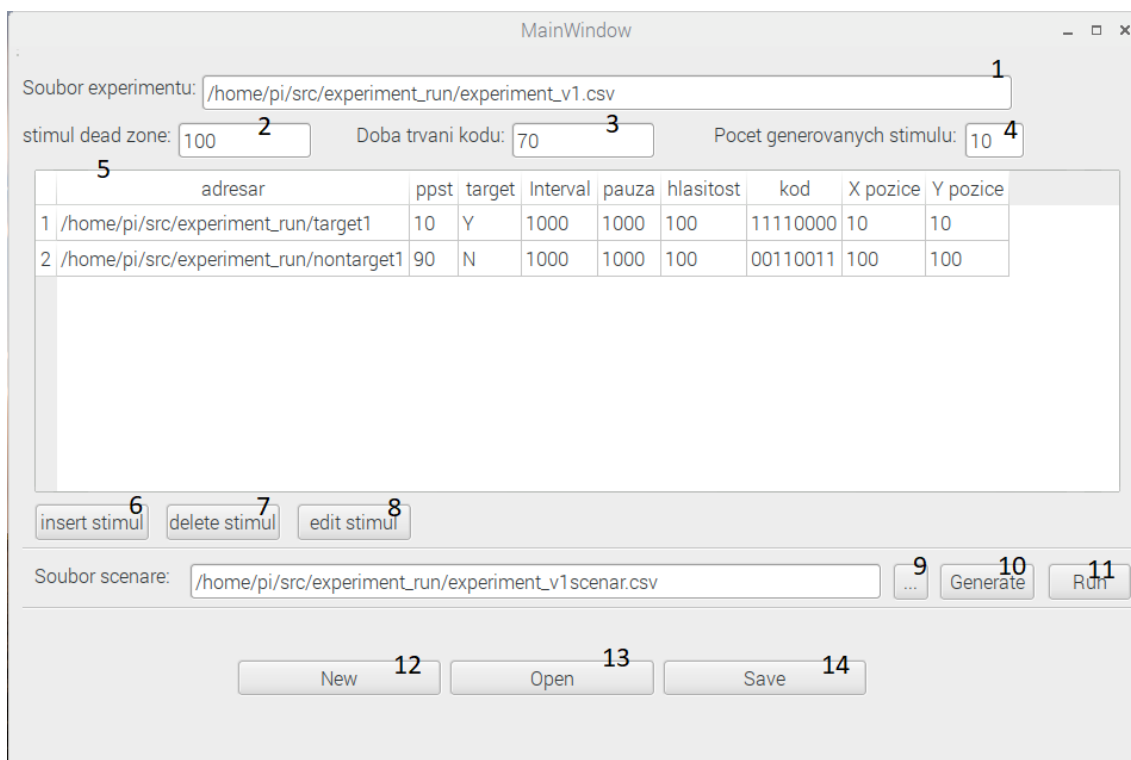
Logovaná hodnota je součtem pinů nastavených na 1.

## Výstupní PINY

- PIN10 - index pozice v kódu stimulu 8
- PIN9 - index pozice v kódu stimulu 7
- PIN11 - index pozice v kódu stimulu 6
- PIN5 - index pozice v kódu stimulu 5
- PIN6 - index pozice v kódu stimulu 4
- PIN13 - index pozice v kódu stimulu 3
- PIN19 - index pozice v kódu stimulu 2
- PIN26 - index pozice v kódu stimulu 1

Číslování jde od leva do prava.

# D Příloha B - stručný návod k aplikaci

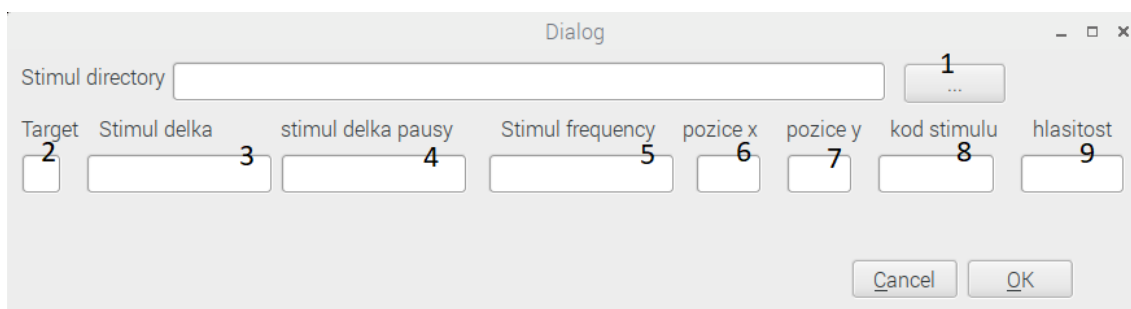


Obrázek D.1: Hlavní okno aplikace

Význam jednotlivých ovládacích prvků hlavního okna:

1. soubor experimentu - zobrazuje soubor ve kterém je uložen aktuální experiment, při vytváření nového může být prázdný
2. stimul dead zone - interval po který odpověď není platná
3. doba trvání kódu - interval jak dlouho je výstupní kód stimulu nastaven an GPIO
4. počet generovaných stimulu
5. tabulka obsahující seznam stimulu aktuálního experimentu

6. EDIT - umožní editaci aktivního řádku
7. INSERT - umožní vložení nového řádku
8. DELETE - maže aktivní řádek
9. tlačítko pro výběr souboru scénáře, který bude použit pro spuštění nebo generování
10. generování nového scénáře
11. spuštění aktuálního scénáře
12. vytvoření nového experimentu
13. otevření uloženého experimentu
14. uložení vytvořeného experimentu



Obrázek D.2: Okno pro zadávání a úpravy stimulů

Okno pro úpravy a vkládání nových stimulů:

1. tlačítko pro výběr adresáře, kde se stimul nachází
2. druh stimulu - Y = target stimul, N = nontarget stimul
3. doba trvání stimulu (v milisekundách)
4. délka pauzy následující po stimulu (v milisekundách)
5. pravděpodobnost se kterou se má být stimul obsažen ve scénáři v procentech
6. vertikální souřadice obrazového stimulu, 0 je nahoře
7. horizontální souřadice obrazového stimulu, 0 je vlevo
8. kód stimulu, který bude nastaven na GPIO během přehrávání stimulu
9. hlasitost přehrávání zvukového stimulu

## **Mazání obrazovky mezi stimuly**

Pro mazání obrazovky mezi stimuly slouží soubor blankscreen.bmp, který musí být přítomen v adresáři, ze kterého je scénář spouštěn a mít stejnou velikost jako největší stimul.

## **Adresářová struktura aplikace**

- BIN - obsahuje přeložené kódy dvou programů, GUI - (erp-test) a aplikace pro spouštění experimentu (experiment\_run)
- ERP\_TEST - obsahuje zdrojové kódy ke grafickému rozhraní
- ERP\_TEST\_RELEASE - obsahuje soubor Makefile pro překlad grafického rozhraní
- EXPERIMENT\_RUN\_SRC - obsahuje zdrojové kódy ke grafickému rozhraní
- EXPERIMENT\_RUN\_RELEASE - obsahuje soubor Makefile pro překlad experiment\_run
- LIBS - obsahuje knihovny potřebné pro překlad