

**Systém audiovizuální stimulace s EEG zpětnou
biologickou vazbou**

**SYSTEM OF AUDIOVISUAL STIMULATION
WITH EEG BIOFEEDBACK**

DIPLMOVÁ PRÁCE

Student: Bc. David Fiřt

Vedoucí diplomové práce: Ing. Karel Hána Ph.D.

2009

Univerzita Karlova v Praze

1. lékařská fakulta

Studijní program: Specializace ve zdravotnictví

Studijní obor: Zdravotnická technika a informatika

ID studijního oboru: 5345T018

Bc. David Fiřt

**System audiovizuální stimulace s EEG zpětnou biologickou
vazbou**

Typ závěrečné práce

Magisterská

Vedoucí závěrečné práce: Ing. Karel Hána Ph.D.

Datum práce

Praha, 27.5.2009

Prohlášení:

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje.

V Praze, 27.5.2009

David Fiřt

Identifikační záznam:

David, Fiřt. Systém audiovizuální stimulace se zpětnou biologickou vazbou. [SYSTEM OF AUDIOVISUAL STIMULATION WITH EEG BIOFEEDBACK]. Praha, 2009. 163 s., 13 příl., 7 tabulek, 54 obr.. Magisterská. Univerzita Karlova v Praze, 1. lékařská fakulta, Společné pracoviště biomedicínského inženýrství ČVUT a UK / FBMI 2009. Vedoucí závěrečné práce Ing. Karel Hána Ph.D .

FIŘT, David. Systém audiovizuální stimulace se zpětnou biologickou vazbou. [s.l.], 2009. 163 s. , 2 schémata, 6 fotografií, 1 graf, 3 výpisy programů, 1 CD-ROM . Univerzita Karlova v Praze, 1. lékařská fakulta. Vedoucí diplomové práce Ing. Karel Hána Ph.D.

Abstrakt

Název: Systém audiovizuální stimulace s EEG zpětnou biologickou vazbou

Autor: David Fiřt

Vedoucí: Ing. Karel Hána Ph.D.

Abstrakt:

V diplomová práci je popsáno sestavení přístroje pro audiovizuální stimulaci s EEG zpětnou vazbou z běžně dostupných hotových komponent, kterými jsou EEG zesilovač, AD převodník, převodník USB/UART, přístroj pro AVS a osobní počítač. Dále je zde popsán program speciálně napsaný pro tuto aplikaci v jazyce Delphi, který realizuje načítání hodnot EEG signálu z AD převodníku, analýzu vstupních dat EEG pomocí Fourierovy transformace včetně grafické prezentace a zpětně ovlivňuje mozkovou činnost pomocí přístroje pro audiovizuální stimulaci vybaveného technologií AudioStrobe.

Klíčová slova

audiovizuální stimulace, EEG, zpětná biologická vazba

Abstract

Title: System of audiovisual stimulation with EEG biofeedback

Autor: David Fiřt

Leader: Ing. Karel Hána Ph.D.

Abstract:

In thesis be described to make-up apparatus audiovisual stimulation with EEG backward structure from readily available ready component that the are EEG amplifier, AD inverter, inverter USB/UART, apparatus for AVS and personal computer. Next is here describe programme specially written for this application in language Delphi that the implements counting values EEG signal from AD inverter, analysis enter data EEG by the help of Fourier transformation inclusive graphic presentation and by return works cerebral activity by the help of apparatus for audiovisual stimulation equipped technology AudioStrobe.

Keywords

audiovisual stimulation, EEG, biofeedback

Poděkování

Rád bych poděkoval všem, kteří se zasloužili na vzniku této diplomové práce, především vedoucímu diplomové práce Ing. Karlovi Hánovi Ph.D. za vedení projektu a za podnětné připomínky, svým rodičům za materiální a morální podporu a také lidem, kteří se na internetu nezištně dělí o své znalosti, zkušenosti i projekty, které vytvořili.

Obsah

1. Úvod.....	- 13 -
2. Cíl práce	- 15 -
3. Fyziologie zrakového a sluchového ústrojí.....	- 16 -
3.1. Zrakové ústrojí.....	- 16 -
3.1.1. Stavba a funkce oka.....	- 16 -
3.1.2. Zraková dráha	- 18 -
3.2. Sluchové ústrojí	- 19 -
3.2.1. Stavba a funkce sluchového ústrojí.....	- 19 -
3.2.2. Sluchová dráha.....	- 21 -
3.3. Senzorické oblasti mozkové kůry	- 22 -
4. Audiovizuální stimulace	- 23 -
4.1. Podmínky pro audiovizuální stimulaci	- 23 -
4.2. Formy stimulace	- 24 -
4.2.1. Visuální stimulace	- 24 -
4.2.2. Zvuková stimulace	- 24 -
4.3. Fázování stimulačních impulsů	- 26 -
4.3.1. Synchronní fázování.....	- 26 -
4.3.2. Asynchronní fázování.....	- 26 -
4.3.3. Laterální fázování.....	- 27 -
4.4. Technologie AudioStrobe.....	- 27 -
4.5. Využití AVS přístrojů	- 27 -
4.5.1. Relaxace.....	- 27 -
4.5.2. Sport a regenerace	- 28 -
4.5.3. Učení se a zlepšení všech duševních výkonů	- 28 -
4.5.4. Zábava a rozvoj fantazie.....	- 28 -

4.6. AVS přístroj Orion.....	- 29 -
5. Elektroencefalogram (EEG).....	- 31 -
5.1. Vznik mozkové aktivity	- 31 -
5.2. Parametry EEG.....	- 31 -
6. Fourierova transformace	- 35 -
6.1. Podstata Fourierovy transformace	- 35 -
6.2. Fourierova transformace pro spojité signály	- 35 -
6.3. Fourierova transformace pro diskrétní signály	- 36 -
6.3.1. Rychlá Fourierova transformace	- 36 -
6.3.1.1. Algoritmus rychlé Fourierovy transformace.....	- 37 -
6.3.1.2. Praktická aplikace rychlé Fourierovy transformace.....	- 40 -
6.4. Váhovací okna	- 41 -
7. Systém audiovizuální stimulace s EEG zpětnou biologickou vazbou.....	- 43 -
7.1. Výběr komponent	- 43 -
7.2. Zapojení komponent	- 44 -
7.3. Modular EEG.....	- 47 -
7.3.1. Snímací elektrody.....	- 48 -
7.3.2. EEG zesilovač	- 48 -
7.3.3. Frekvenční filtr.....	- 50 -
7.3.4. DRL zesilovač.....	- 51 -
7.3.5. Mikroprocesor - AD převodník	- 51 -
7.3.6. Optické oddělovače	- 54 -
7.3.7. Převodník úrovní TTL/sériové rozhraní.....	- 54 -
7.3.8. Napájení modulů	- 55 -
7.3.9. Technické parametry Modular EEG	- 56 -
7.3.10. Oživení Modular EEG.....	- 57 -
7.4. Převodník USB/sériové rozhraní	- 58 -

7.4.1. Vlastnosti modulu UMS3	- 58 -
7.4.2. Zapojení UMS3	- 60 -
7.5. Programová podpora projektu	- 61 -
7.5.1. Struktura programů	- 61 -
7.5.2. Program Mexický příliv	- 63 -
7.5.2.1. Popis programu Mexický příliv	- 63 -
7.5.2.2. Mapování souborů	- 64 -
7.5.2.3. Popis formuláře programu Mexický příliv	- 65 -
7.5.2.4. Vývojový diagram pro program Mexický příliv	- 74 -
7.5.2.5. Popis vývojového diagramu programu Mexický příliv	- 82 -
7.5.2.6. Popis vývojového diagramu programu Mexický příliv - Procedura Načítání a práce s AVS	- 85 -
7.5.3. Program FFTnew	- 89 -
7.5.3.1. Popis programu FFTnew	- 89 -
7.5.3.2. Vývojový diagram programu FFTnew	- 91 -
7.5.3.3. Popis vývojového diagramu programu FFTnew	- 93 -
7.5.4. Program Generátor	- 96 -
7.5.4.1. Popis programu Generátor	- 96 -
7.5.4.2. Vývojový diagram programu Generátor	- 97 -
7.5.4.3. Popis vývojového diagramu programu Generátor	- 99 -
7.6. Výsledky stimulace	- 102 -
7.7. Vyhodnocení výsledků	- 105 -
8. Závěr	- 106 -
9. Seznam použité literatury	- 108 -
10. Seznam obrázků	- 111 -
11. Seznam tabulek	- 113 -
12. Seznam příloh	- 114 -

13. Přílohy - 115 -

1. Úvod

Antropologové objevili, že šimpanzi chodili desítky kilometrů daleko jen proto, aby viděli záblesky slunce na vodní hladině nebo slyšeli zvuk vodopádu. Zjistili, že sledováním určitých zvukových a světelných vjemů si savci udržují psychosomatickou rovnováhu. Lidé intuitivně využívají působení přírodní stimulace již nejméně přes desetitisíc let. Pozorování plápolajícího ohně, hukot vody či rytmické bubnování nás uklidňuje, při zrychlení rytmu bubnování se naopak objevuje pocit vybuzení mysli. Stav radostné euforie dokázal u davu lidí vyvolat Ptolemaios, který nechal sluneční paprsky probleskovat kruhem kolovrátku.

V průběhu minulých dvou století se mnoho vědců i lékařů snažilo prozkoumat vliv vnější stimulace na lidskou mysl nebo různé formy stimulace použít k léčení pacientů. Do počátku dvacátého století nebylo možné tyto poznatky žádným vědeckým způsobem zdokumentovat, až v roce 1924 německý psychiatr Hans Berger sestavil první encefalograf a objevil tím existenci mozkových vln. Bylo zjištěno, že princip frekvencí je pro náš mozek základním principem fungování. Jinou frekvenci vysílá při bdění, jinou ve spánku a jinou při bolesti.

V letech 1940 - 1944 zdokonalil anglický neurofyziolog Walter Grey princip audiovizuální stimulace. Svým objevem efektu napodobení frekvence zjistil, že lidský mozek reaguje na pulsující světlo a zvuk tak, že přeladí svojí frekvenci. Postupně s rozvojem elektroniky bylo vyrobeno množství vizuálních a audio stimulačních přístrojů. Největšího rozvoje v oboru audiovizuální stimulace však došlo teprve v 80. letech minulého století, kdy se i v této oblasti začali používat mikroprocesory a tím bylo možné použít zrakovou i sluchovou stimulaci najednou v synchronizované podobě.

Klasické AVS přístroje postupně mění frekvence podle předem zvoleného programu, ale nepočítají s tím, že uživatele může něco vyrušit nebo prostě, že začne přemýšlet o problémech každodenního života, čímž se poruší synchronizace mozkových vln s frekvencí vysílanou AVS přístrojem. Toto se snaží vyřešit navržený přístroj v této diplomové práci použitím EEG zpětné vazby u audiovizuální stimulace. Využito je přitom vlastnosti mozku, který se snadno synchronizuje s externí frekvencí s malým rozdílem vůči své vlastní, nežli s frekvencí velmi vzdálenou. Přístroj přizpůsobuje frekvenci audiovizuální stimulace aktuálním frekvencím mozkových vln

tak, aby byl rozdíl těchto frekvencí minimální a přitom se pomalu snaží přeladit mozek na uživatelem danou relaxační úroveň.

2. Cíl práce

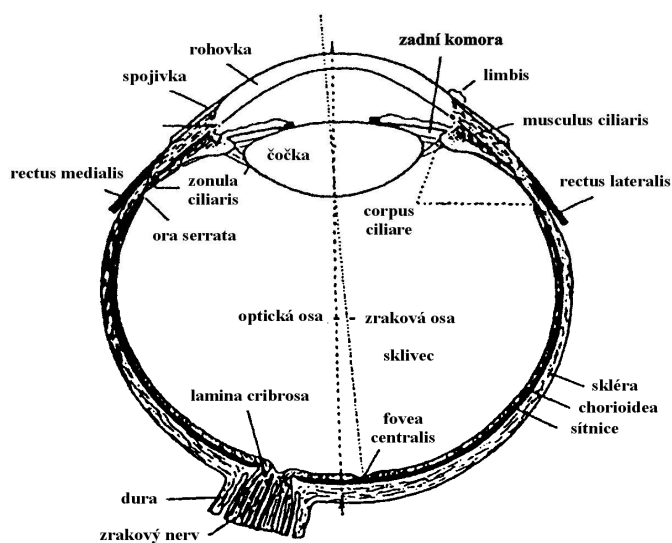
Cílem této diplomové práce je navrhnutí a sestavení přístroje pro audiovizuální stimulace s EEG zpětnou biologickou vazbou. Jako jednotlivé části budou použity sériově vyráběné komponenty pro zrychlení stavby zařízení. Zařízení bude tvořit EEG zesilovač, AD převodník, přístroj pro audiovizuální stimulaci a počítač s programem vytvořeným speciálně pro tuto aplikaci. Při volbě jednotlivých komponent bude třeba dbát na jejich elektrotechnické parametry, ale i možnosti zapojení do funkčního celku. Ovládací program bude vytvořen tak, aby uživatel mohl na počátku nastavit důležité parametry pro správnou analýzu měřených EEG signálů a jejich vyhodnocování. Průběh audiovizuální stimulace se zpětnou EEG vazbou bude zdokumentován ve výstupním souboru tak, aby bylo možné ověřit správnost fungování celého zařízení.

3. Fyziologie zrakového a sluchového ústrojí

3.1. Zrakové ústrojí

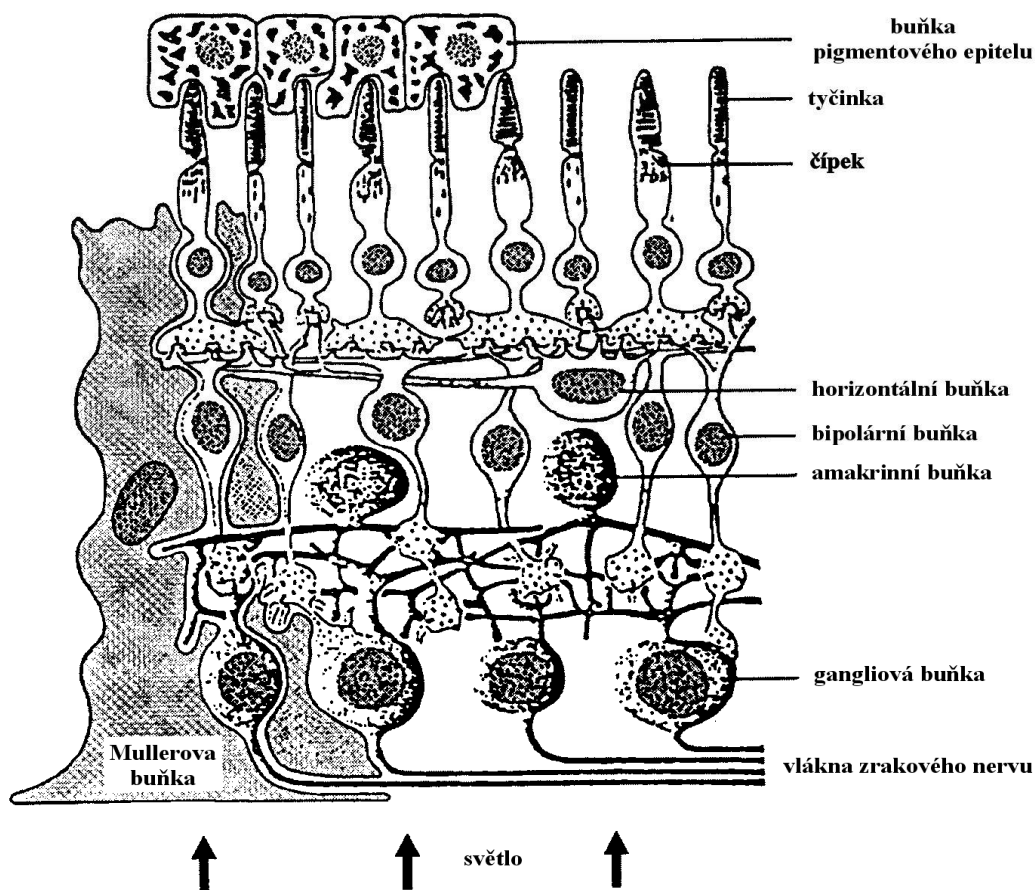
3.1.1. Stavba a funkce oka

Zrakem vnímáme světlo v rozsahu cca 400-750 nm vlnové délky elektromagnetického vlnění. Zrakem získáváme přibližně 60-65 % všech našich informací. U audiovizuální stimulaci je vizuální stimulaci také dominantní složkou stimulace. Oko je tak naším nejvýznamnějším smyslovým orgánem[3].



Obr.1. Řez pravého oka [1]

Světelný paprsek, pronikající optickým aparátem oka a inertním sklivcem, dopadá na sítnici (retinu), která představuje vlastní receptorovou strukturu oka. Skládá se z fotosenzitivních receptorů, kterými jsou tyčinky a čípky. Na ně jsou napojeny bipolární buňky, jejichž výběžky mají spojení s gangliovými buňkami sítnice. Neurity gangliových buněk opouštějí bulbus jako zrakový nerv v místě zvaném slepá skvrna (fovea caeca). Zrakový nerv je tvořena výstupem asi 1 milionu nervových vláken gangliových buněk sítnice z očního bulbu. Představuje tak nosič zrakových informací z asi 130 miliónů tyčinek a 6 milionů čípků.



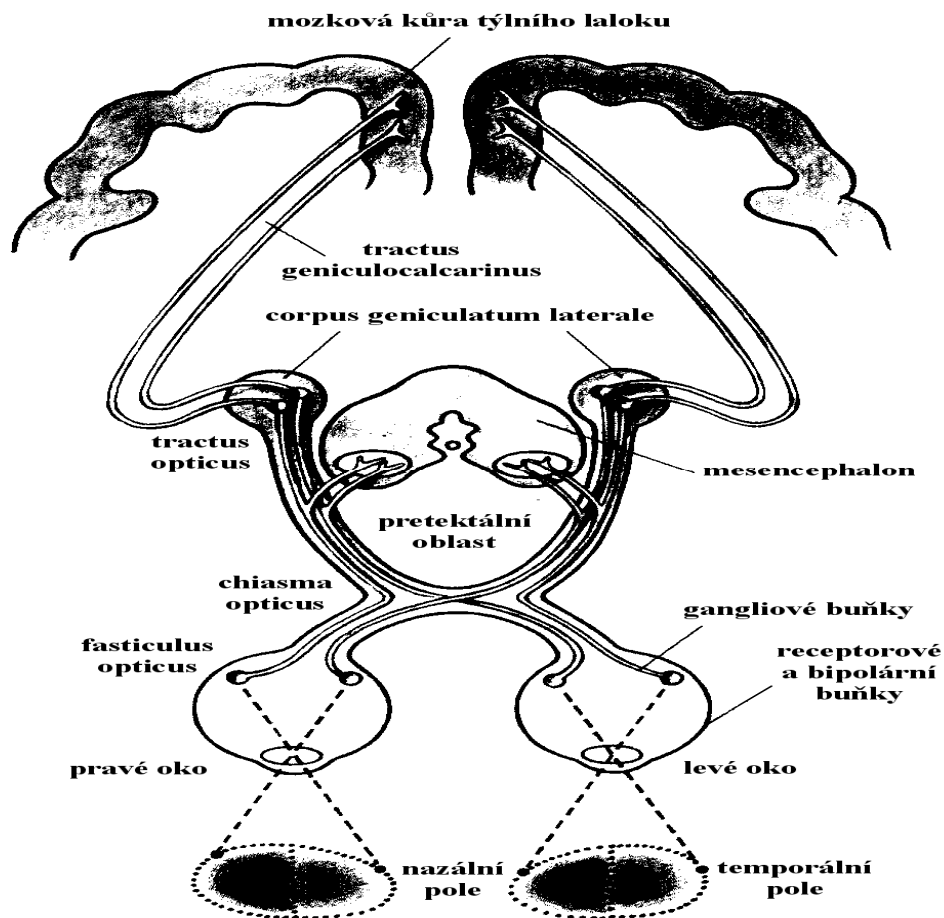
Obr.2. Zjednodušené schéma sítnice [1]

Tyčinky a čípky nejsou v sítnici uloženy pravidelně. V žluté skvrně (fovea centralis) jsou přítomny pouze čípky. Je to místo nejostřejšího vidění, protože na 1 čípek se napojuje 1 neuron. Nedochozí zde k divergenci ani konvergenci informace. Čípky jsou schopny vidět pouze za světla a barevně. Směrem od fovea centralis do periferie sítnice ubývá čípků a relativně přibývá tyčinek. Tyčinky umožňují vidění za šera, černobíle a neostře.

Podstatou barevného vidění je existence tří pigmentů v terčících čípků. Tyto fotolabilní pigmenty reagují na tři různé vlnové délky spektra. Existují tak 3 druhy čípků s různými zrakovými pigmenty, které maximálně absorbují v určitých oblastech viditelného spektra a zprostředkují tak vnímání barev různých odstínů podle toho, v jakém poměru a s jakou účinností je dráždí příslušný podnět. Existují tak čípky pro barvu červenou (absorpce v rozsahu 750-600 nm), zelenou (550-500 nm) a modrou (500-400 nm). Tyčinky mají pigment citlivý jen na vlnovou délku 500 nm.

K podráždění tyčinky stačí energie 1 fotonu, ale k podráždění čípku je třeba několik stovek fotonů. Znamená to, že tyčinky jsou citlivější než čípky.

3.1.2. Zraková dráha



Obr.3. Zraková dráha [2]

Na spodině lebeční se vlákna z mediálních polovin sítnice, tj. přenášející obrazy z vnějších polovin zorných polí každého bulbu, kříží v oblasti chiasma opticum, kdežto z mediálních polovin zorných polí pokračují nezkříženě. Do levé hemisféry mozku tak pokračují obrazy z levých polovin sítnic, tj. z pravých polovin zorných polí, zatímco u pravé hemisféry je tomu naopak. V chiazmatu se z optické dráhy odpojují vlákna do suprachiasmatického jádra, jehož funkcí je koordinace cirkadiánní rytmicity. Z chiazmatu pokračuje optická dráha v obou hemisférách jako tractus opticus do ipsilaterálního specifického jádra talamu - corpus geniculatum laterale (CGL). Cestou však vysílá odbočky do čtverohrbolí středního mozku (colliculi superiores), jež synchronizuje rychlé pohyby očí a koordinuje je s pohyby těla, a do pretektální arey, která je odpovědná za nepodmíněné reflexní pohyby očí. Z CGL vycházející axony vytvářejí radiatio optica a vedou do primární zrakové korové

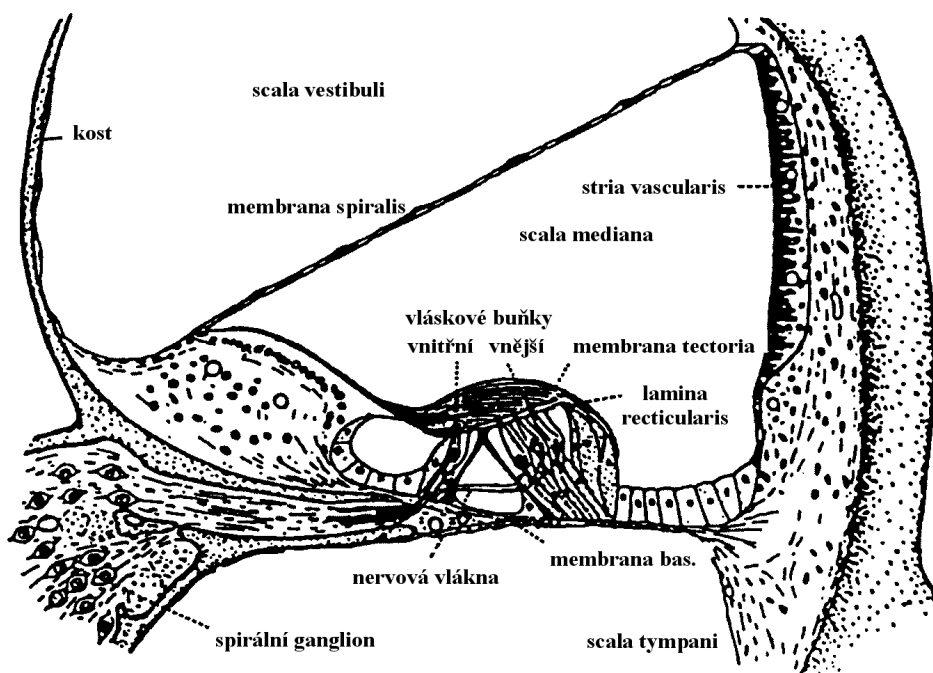
oblasti v okcipitálním laloku (Brodmannova area 17, zraková oblast I). Area 17 je uspořádána do sloupců, ve kterých je analyzován tvar a orientace obrysů a pohyb. Uprostřed sloupců jsou struktury, kterým říkáme čepy, kde je kódována barva. Z primární zrakové oblasti dále vjem postupuje do sekundární zrakové oblasti (Brodmannova area 18, zraková oblast 2) a do vzdálenějších asociačních zrakových oblastí. Dráha probíhá jednak dorzální cestou do parietální kůry (odpovědná za lokalizaci, pohyb a poznání předmětu), jednak ventrální cestou do temporálního laloku (analýza barev a tvaru).[2]

3.2. Sluchové ústrojí

3.2.1. Stavba a funkce sluchového ústrojí

Sluch má největší citlivost ze všech smyslů. Zvuky slyšíme od 16 Hz do 20000 Hz. Tato hranice je ovšem velmi relativní, protože s postupujícím věkem se výrazně snižuje. Největší citlivost pro vnímání zvuků je v oblasti lidské řeči a hudby v rozmezí 1000 až 5000 Hz.

Sluchové ústrojí dělíme na zevní, střední a vnitřní ucho. Zevní ucho je tvořeno boltcem a zvukovodem, který je ukončen bubínkem. Bubínek (membrana tympani) kmitá ve stejné frekvenci, s jakým kmitočtem přichází zvukový stimulus a předává tuto frekvenci dál, a to prostřednictvím tří na sebe navazujících kůstek středního ucha. Jde o kladívko, kovádlíku a třmínek. Impulsy se přenáší, aniž by se zkreslily, na oválné okénko. Prostor středního ucha je spojen s nosohltanem Eustachovou tubicí. To je velmi důležité, protože tímto kanálem je vyrovnáván tlak ve vnitřním uchu s tlakem atmosférickým a nedochází k deformaci bubínku.

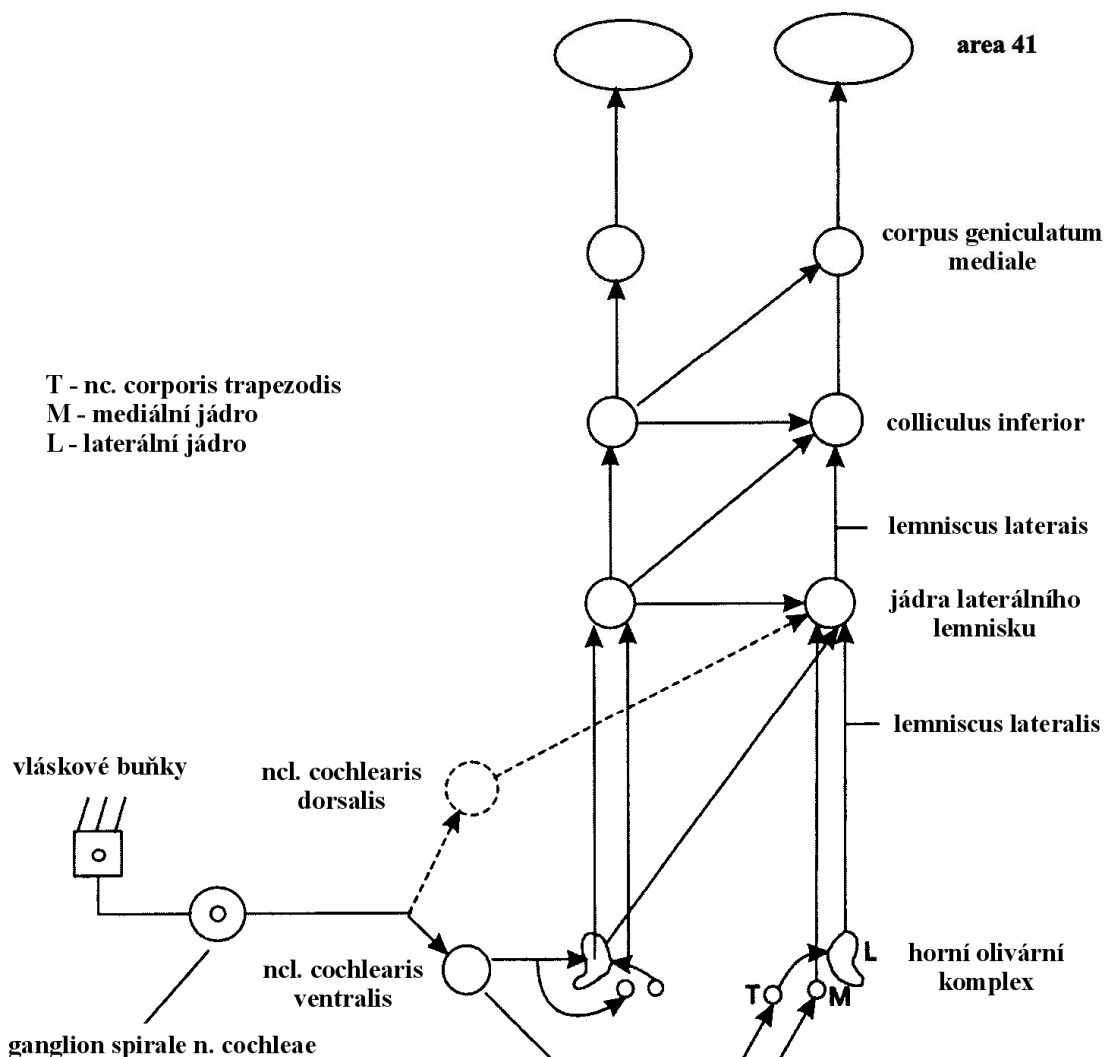


Obr.4. Scala media a Cortiho orgán [1]

Samotné vnitřní ucho má složitější strukturu. Vlastní sluchový orgán je uložen v hlemýždi vnitřního ucha ve skalní kosti. Hlemýžď tvoří tři kanálky, stočené do spirály. Scala tympani a scala vestibuli jsou vyplněny perilymfou a scala media endolymfou. Vibrace z oválného okénka se dál šíří prostředím perilymfy ve scala vestibuli, přenáší se membránou spiralis do scala media a rozechvívá totožnými frekvencemi bazilární membránu. Na té jsou umístěny vláskové buňky sluchového ústrojí (Cortiho orgánu), které se aktivují. [3]

Čím je vyšší tón, tím je vibrace lokalizovaná blíže k bázi hlemýžďe, čím je naopak hlubší, tím je lokalizována blíže k vrcholu hlemýžďe. Jde o tzv. tonotopii - o lokalizaci různých frekvencí (výšek tónů) na různých místech bazilární membrány. Receptorový potenciál vzniká při ohybu cilií senzoričkových vláskových buněk a to vzhůru nebo dolů. Při pohybu vzhůru vzniká depolarizace, při dolním pohybu hyperpolarizace. Při pohybu cilií se otevírají nebo zavírají na principu pružin iontové kanály.

3.2.2. Sluchová dráha



Obr.5. Zjednodušené schéma sluchové dráhy [1]

Neurity aferentních neuronů vedou z Cortiho orgánu do anteroventrálního, posteroventrálního a dorzálního kochleárního jádra. V těchto třech jádrech je jejich aference uspořádána v různém stupni komplexnosti podle kmitočtů. Laterální útlum zde zesiluje kontrast, tj. potlačuje šum. V horní olivě a nucleus accessorius se porovnává intenzita a časový průběh podnětů. Další přepojovací stanicí je jádro laterálního lemnisku a dále colliculi inferiores. Přes thalamus se aferentní informace nakonec dostanou do primární sluchové kůry, která je obklopena sekundárními sluchovými oblastmi. Úlohou těchto center je analýza komplexních zvuků,

krátkodobá paměť při porovnávání tónů, útlum nevhodných motorických reakcí, zaměření pozornosti na sluchové podněty a mnoho dalších. [2]

3.3. Senzorické oblasti mozkové kůry

Dráždění periferních nervů jednotlivými podněty (působení světelných záblesků na sítnici, dráždění kochleárních nervových zakončení krátkými zvukovými impulsy, atd.) vyvolá odpovědi v příslušných korových oblastech. V recepční oblasti mozkové kůry se jejich příchod projeví náhlým zvýšením elektrické aktivity v této oblasti. Experimentálně bylo ověřeno, že mozek velmi citlivě a rychle reaguje na jakoukoliv změnu ve vnějším prostředí. Pochody, které souvisejí se skutečným rozpoznáváním vnějších zrakových, sluchových a jiných podnětů, začínají se zpožděním asi 300 ms [3].

4. Audiovizuální stimulace

Audiovizuální stimulace působí na naladění mozku tím způsobem, že stimulační přístroj programově generuje světelné záblesky a zvukové rytmy, které pomocí zrakového a sluchového ústrojí vysílají pulsy do zrakových a zvukových mozkových center. Tato centra kopírují příchozí frekvence a postupně je přenášejí do ostatních oblastí mozku, čímž dochází ke změně stavu vědomí jedince.

4.1. Podmínky pro audiovizuální stimulaci

Aby byl výsledek stimulace co nejefektivnější, musí být splněna řada požadavků jak na straně přístroje, tak i na straně stimulované osoby.

Přístroj musí vysílat synchronně světelné i zvukové stimulační pulsy v přesných hodnotách frekvence po zvolenou dobu. Hodnoty těchto frekvencí se synchronně mění v potřebném rozsahu pásma a v závislosti na účelu aplikace. Rozsah frekvenčního pásma pro vizuální stimulaci je 0,5 - 40 Hz, tedy frekvenčního rozsahu bioelektrické činnosti mozku klinicky zdravého dospělého člověka. Intenzita světelných záblesků se musí dát měnit v rozsahu 0 - 400 luxů. U generovaných tónů by mělo být frekvenční rozmezí alespoň 60 - 900 Hz s možností manuálně měnit tóninu zvuků a hlasitost v rozsahu 0 - 100 decibelů. Stimulovaná osoba by měla mít určitou vůli se účinkům stimulace podřídit a vědomě vyloučit nebo potlačit všechny ostatní smyslové podněty ze svého okolí. Existují však skupiny osob, které by bez konzultace s odborným lékařem neměly AVS přístroj používat. Jsou to lidé s diagnostikovanou duševní nebo nervovou a psychiatrickou poruchou, lidé po vážném úraze hlavy nebo páteře a lidé trpící jakýmkoliv záchvatovitými stavy, např. epilepsií. Dále by AVS přístroj neměly používat těhotné ženy. Pro zdravé děti do zhruba 6-10 let věku nemá obvykle použití žádný přínos. Horní věková hranice není omezena, u osob s vážným onemocněním je doporučena konzultace s lékařem. AVS přístroje se nedoporučují používat souběžně s alkoholem, psychofarmaky nebo drogovými látkami [4].

4.2. Formy stimulace

Při audiovizuální stimulaci se využívá pouze zvukové a vizuální stimulace. Oba typy stimulací jsou schopny samy o sobě přeladit mozek na stimulační frekvenci.

4.2.1. Vizuální stimulace

Při vizuální stimulaci dochází k rychlému dostavení účinků stimulace, což je způsobeno tím, že zrakové centrum zabírá značnou část kůry mozkové a informace vedená nervovým kanálem se dostává do obou polovin mozku současně a dvojmo.

Pro zrakovou stimulaci se dnes používají tři druhy stimulačních brýlí. Jsou to standardní brýle s LED diodami, brýle pro otevřené oči a brýle Lumasoft.

Standardní brýle s LED diodami vypadají téměř stejně jako brýle sluneční, s tím rozdílem, že blikající diody jsou nainstalovány z vnitřní strany tmavých skel. Obvykle se instalují čtyři diody uprostřed každého skla. Tyto brýle jsou určeny k užívání se zavřenými očima a mohou být použity k jakémukoliv stimulačnímu programu. Barvy používaných diod jsou červená, jantarová, modrá, zelená a bílá.

Brýle pro otevřené oči se podobají brýlím na čtení, s tím rozdílem, že na rámu obrouček jsou nainstalovány světelné diody. Uživatel může současně užívat stimulační program a kontrolní přístroj zpětné vazby, zejména při procvičování v hladině beta. Tyto speciální brýle uživateli umožňují přijímat světelnou stimulaci s otevřenými očima a současně pozorovat monitor počítače, na němž jsou zobrazeny vzorce jeho mozkových vln z přístroje zpětné vazby. Další možností je použití stimulace během práce nebo studia.

Brýle Lumasoft jsou třetí možností. Ke stimulaci využívají malé výbojky s bílým světlem. Mezi výbojkami a očima uživatele je umístěno stínítko. Jejich užití je shodné jako u brýlí s LED diodami.

4.2.2. Zvuková stimulace

U zvukové stimulace je doba přeladění frekvence delší než u vizuální stimulace z důvodu pomalejšího přenosu signálu do sluchového centra, ve kterém dochází k vyhodnocení signálů v obou polovinách mozku odděleně. Zvukové stimulační impulsy mají nejčastěji tvar sinusovky, pily nebo pravoúhlé vlny.

K účinným typům stimulačních akustických modulovaných signálů patří bílý šum, klapot, izochronní tóny, šedý nebo hnědý šum, monaurální rytmy, binaurální rytmy a duální binaurální rytmy [5]. Všechny z nich lze poměrně snadno vytvořit digitálně elektronicky.

Každý z těchto zvuků má z fyziologického i psychologického hlediska výhody i nevýhody.

Klapot je fyziologicky nejúčinnější. Podobá se tupému bubnování v jediné tónině, zpravidla mezi 80-100 Hz a v cílové frekvenci 1-40 Hz. Jeho nevýhoda spočívá v jeho psychologické nelíbivosti - většina lidí jej špatně snáší. Z tohoto důvodu se v přístrojích AVS neinstaluje.

Bílý šum je velmi rychlým a nahodilým střídáním zvukových frekvencí v rozpětí 30-1000 Hz a jeho účinek na naladění není ani cílený, ani bezprostřední. Je vlastně formou smyslové deprivace sluchu tím, že poskytuje neurčitou a zmatečnou informaci. Mozkové centrum sluchu na takový podnět reaguje útlumem a tak podpoří celkový útlum CNS. Většina lidí na bílý šum reaguje psychologicky velmi dobře. Šedý a hnědý šum jsou segmenty bílého šumu, kdy jsou záměrně určitá frekvenční pásma bílého šumu odfiltrována. Zvláštní fyziologické náležitosti těchto šumů jsou doposud velmi málo prozkoumány, a tudíž jejich využití při stimulaci je velmi výlučné.

Izochronní nebo pulzní tóny existují ve třech tónových charakteristikách jako sinusové hudební tóny, triangulární tóny a pravoúhlé tóny. Tento typ zvuku má menší fyziologickou účinnost než klapot, avšak jeho poslech je většinou lidí příjemný. Jeho nevýhodou je, že plnohodnotně působí jen v rozmezí 80-180 Hz. Izochronní a pulzní tóny jsou nejčastěji využívány v asynchronním režimu stimulace, kdy jeden stimul pro jedno ucho je generován o milisekundy dříve než druhý.

Monaurální rytmy vznikají vysíláním dvou tónů různých frekvencí (např. 140 a 150 Hz), které jsou mixovány do jediného zvuku a vysílány pro obě uši současně. Jejich fyziologická účinnost je poměrně nízká, a proto jsou v posledních letech využívány málo, a to buď jako doprovodný mix k šumu nebo v generátorech psychoakustické stimulace poslouchané z reproduktoru.

Binaurální rytmy jsou zvukové stimuly, které vznikají přímo a pouze v mozku. Jejich příjem bez stereo sluchátek nevyvolá potřebný účinek. Generátor vysílá pro jedno ucho zvuk na nosné frekvenci, např. 200 Hz, a v témže okamžiku pro

druhé ucho jiný zvuk na offset frekvenci, např. 210Hz. Sluchové centrum pak vyhodnotí příjem jako rozdíl obou frekvencí (10 Hz) a na tuto rozdílovou frekvenci se přeladí. Binaurální rytmy jsou fyziologicky velmi účinné, avšak jejich cílová stimulační frekvence má individuálně horní okraj okolo 18-22 Hz. Vyšší rozpětí už mozek nestihne asimilovat a vnímá je jako dva různé tóny, to znamená, že se jejich specifický účinek zcela vytrácí. Největší výhodou binaurálních rytmů je, že působí daleko pod i nad prahem slyšitelnosti. Binaurálními rytmy jsou vybaveny všechny modely stimulatorů s audio výstupem na sluchátka [I1].

Charakteristika duálních binaurálních rytmů je naprosto shodná s jednoduchými. Rozdíl spočívá v tom, že generátor vysílá pro každé ucho zvlášť mix dvou hudebních tónů. Například pro stimulační frekvenci 4 Hz jsou pro pravé ucho generovány tóny o frekvenci 128 a 256 Hz a pro levé ucho tóny o frekvenci 128+4 a 256+4 Hz. Duální binaurální rytmus poskytuje bohatší a plnější zvuk, který má většinou i výraznější psychodynamický efekt a zejména osobám s kultivovaným sluchem přináší i silnější prožitek.

4.3. Fázování stimulačních impulsů

4.3.1. Synchronní fázování

Při synchronním fázování stimulace vysílají generátory AVS přístroje záblesky světla i akustické impulsy do obou očí a uší ve stejném okamžiku. Fyziologicky danou předností této stimulace je naladit mozkové vlny na danou stimulační frekvenci s relativní přesností a stabilitou výsledku. Z psychologického hlediska je synchronní stimulace většině uživatelů příjemnější než ostatní typy stimulace.

4.3.2. Asynchronní fázování

Při asynchronním fázování jde v jednom okamžiku zrakový impuls do levého oka a sluchový do pravého ucha a v dalším okamžiku jsou současně vyslány impulsy stranově obrácené. Asynchronní stimulace je ve svém okamžitém účinku mnohem razantnější než synchronní. Dokáže důrazněji a rychleji naladit mozkové vlny na danou stimulační frekvenci. Tato přednost se proto využívá v průběhu vstupní fáze programu a často při přechodech o několik Hz výše či níže během dalšího běhu

programu. Zjevnou nevýhodou této stimulace je, že tvoří vyšší harmonické, které jsou násobkem základní nosné frekvence, a které v mozkových centrech rezonují současně s nosnou frekvencí. Vznik vyšších harmonických zejména podporují zrková centra, která při asynchronní stimulaci přijmou v bleskovém sledu z jednoho celého impulsu čtyři oddělené stimuly, což samo o sobě v těchto centrech vyvolává zmatečnou reakci, která není žádoucí. Vyšší harmonické jsou průvodním jevem i u synchronní stimulace, ale jsou tak slabé, že většinou nezpůsobí problémy.

4.3.3. Laterální fázování

Při laterálním asymetrickém fázování jsou dva impulsy vysílány pro levé smyslové receptory a vzápětí dva pro pravé.

4.4. Technologie AudioStrobe

Je schopnost některých typů AVS přístrojů dekodovat z externího zdroje zvuku (např. CD s přírodní hudbou se stopou Audistrobe) digitální stopu, která synchronizuje světelné záblesky. Frekvence této stopy je na prahu slyšitelnosti a to na frekvenci 19 200 Hz, takže uživatel ji neslyší, ale přístroj ji bez problémů dekoduje.

4.5. Využití AVS přístrojů

4.5.1. Relaxace

Programy pro relaxaci slouží k uklidnění, uvolnění a dosažení duševní rovnováhy. Účelem je osvěžení a načerpání nové síly. Při hluboké relaxaci lze prožít výrazné harmonické a citové zážitky a stavy vědomí, které byly uživateli doposud neznámé. Neschopnost dosažení stavu relaxace může mít za následek přivolání či zhoršení chronických i akutních degenerativních chorob mozku a nervové soustavy, soustavy endokrinních žláz, hladkého i kosterního svalstva, kosterního skeletu a kloubů, srdce a krevního oběhu, dýchacího ústrojí, zažívacího a vylučovacího ústrojí, metabolického systému a pochopitelně i psychiky. Účinná relaxace není jen jednou z cest k odstranění nebo zlepšení zdravotních potíží, ale je i významným faktorem prevence, sloužícím k upevnění kondice a zkvalitnění každodenního života [7].

4.5.2. Sport a regenerace

Používání AVS přístrojů vede ke stimulaci mysli, která výrazně ovlivňuje výkonnost. V případě sportovců existují dvě možnosti využití AVS přístrojů. Použití před výkonem umožňuje vybudit na určitou dobu smyslovou i tělesnou výkonnost a zredukovat předstartovní stres. Hluboké uvolnění před výkonem předchází i možným sportovním úrazům. Po sportovním výkonu je stimulace přístroji součástí rehabilitace.

4.5.3. Učení se a zlepšení všech duševních výkonů

Schopnost učit se není ani tak závislá na inteligenci, jako na momentálním duševním naladění. Nejlepších výsledků v učení je dosahováno při naladění mozku na hladinu alfa. V hladině alfa totiž člověk přijímá, třídí a vybavuje si informace několikrát rychleji než v hladině beta. Kromě samotného učení zlepšuje používání AVS přístroje všechny duševní výkony jako jsou paměť, soustředění i IQ.

4.5.4. Zábava a rozvoj fantazie

Rozvoj kreativity do jisté míry poskytuje většina programů AVS přístrojů, ať už to jsou programy zaměřené na uvolnění, učení či zdraví. Působení AVS přístrojů na člověka stimuluje výrazně i pravou, tedy citovou, intuitivní hemisféru, která je sídlem kreativity.

4.6. AVS přístroj Orion

Pro tuto diplomovou práci byl použit přístroj Orion od americké společnosti Syntec Systems International, Inc.. Důvodem použití tohoto přístroje bylo mé vlastnictví a podpora technologie AudioStrobe.



Obr.6. Přístroj pro AVS Orion

Součásti přístroje:

- vlastní tělo přístroje
- světelné brýle s 2x 4 bílými LED diodami
- stereofonní sluchátka

Přístroj obsahuje:

- manuální ovládání hlasitosti
- manuální ovládání intenzity záblesků
- výstupní konektor pro světelné brýle
- vstupně/výstupní konektory pro audio signály - 2x
- konektor pro externí napájení, při jeho nevyužití je přístroj napájen z 3 kusů baterií typu AA
- tlačítko Start/Stop pro spuštění/zastavení stimulace

- tlačítko Mód/Pauza pro volbu uživatelského módu nebo spuštění pauzy během stimulace
- tlačítko Nahoru/Dolů pro volbu stimulačního programu nebo v manuálním režimu pro nastavení frekvence
- spínač ON/OFF pro zapnutí/vypnutí přístroje

Módy přístroje:

- 23 přednastavených programů pro stimulaci
- U1 - AudioStrobe - použití technologie AudioStrobe s využitím externího zdroje zvuku obsahující AudioStrobe stopu (CD)
- U2 - Manual Control - možnost manuálního nastavení stimulační frekvenci v rozsahu 1 - 30 Hz, která je po celou dobu sezení konstantní
- U3 - Color Pulse - transformuje externí zvuk z audio vstupu na světelné pulsy
- U4 - Micro Pulse - transformuje externí zvuk, který je snímán vestaveným mikrofonem na světelné pulsy [I2]

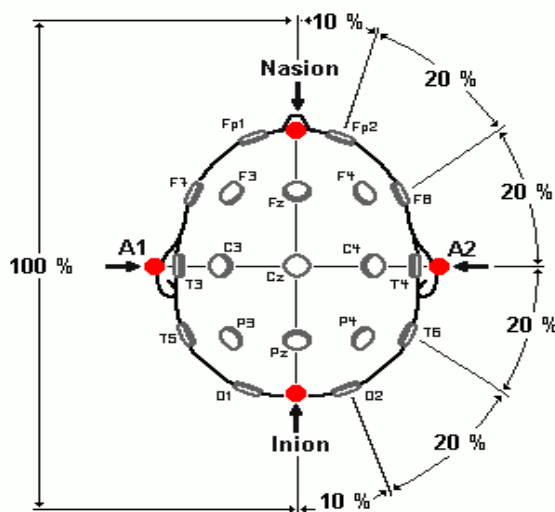
5. Elektroencefalogram (EEG)

5.1. Vznik mozkové aktivity

Elektroencefalogram je sumační záznam elektrické aktivity v CNS. Vzhledem k platnosti Coulombových zákonů zaznamenáváme aktivitu především povrchových struktur mozku. Podkorové struktury se na ní sice také podílejí, ale jejich přínos je o celé řády menší. Amplituda potenciálů naměřených z povrchu kůže lebky je v desítkách μV , tedy o tři řády menší než membránový potenciál, který se měří v mV. Zdrojem EEG aktivity jsou hlavně, díky svému trvání a četnosti, postsynaptické potenciály excitačního i inhibičního typu. Podstatně méně se na ní podílejí akční potenciály, které jsou sice větší, ale mnohem kratší a nevyskytují se tak často jako potenciály postsynaptické. Pro genezi EEG jsou zvláště významné neurony pacemakerového typu spontánně produkující oscilační výboje, dále inhibiční interneurony a zpětnovazebná spojení představující základní princip funkce oscilátoru [1].

5.2. Parametry EEG

Technicky se záznam EEG aktivity provádí jako porovnání potenciálů dvou bodů na kůži lebky - bipolární záznam nebo jako měření rozdílu elektrického potenciálu mezi aktivním bodem mozkové tkáně proti bodu s nulovým potenciálem (např. na ušním boltci) - unipolární záznam.

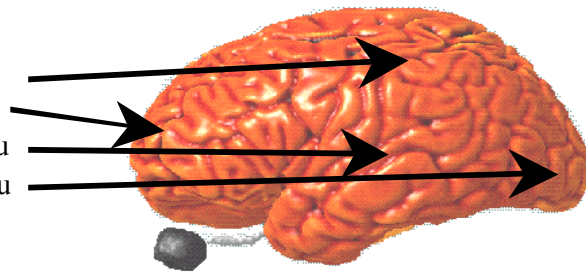


Obr.7. Systém 10-20 [I3]

Označení elektrod

Elektrody začínající písmenem:

- „P“ se nachází na parietálním laloku
- „F“ se nachází na frontálním laloku.
- „T“ se nachází na temporálním laloku
- „O“ se nachází na okcipitálním laloku



Obr.8. Označení elektrod [14]

Bioelektrický signál se získává z elektrod umístěných ve většině případů na kožním povrchu vlasaté části hlavy. Jejich umístění a označení je dáno mezinárodní normou (tzv. systém 10/20, či Montrealskou konvencí). Dán je i způsob vzájemného propojení elektrod při bipolárním způsobu záznamu. Konvence také říká, že pokud vlna nebo hrot, směřuje vzhůru od izometrické linie, je označován jako negativní.

Snímání se provádí buď přímo z vodivého povrchu lebky (elektroencefalografie) nebo peroperačně z povrchu mozkové kůry, resp. měkkých mozkových obalů (elektrokortikografie) či z jiných povrchově dostupných struktur mozku, nebo i z hlubokých mozkových struktur přístupných jen stereotakticky (obvykle v průběhu neurochirurgických operací).

Záznamy EEG jsou sice individuálně variabilní, ale lze v nich určit některé charakteristiky typické jak pro jednotlivce, tak i pro člověka jako druh. Jde o převládající frekvence vln za určitých stavů bdění a spánku, jejich amplituda a konečně i grafoelementy specifické pro určité fyziologické i patologické projevy CNS jako jsou výboje či vřetena. Základní typy aktivit neboli rytmů EEG jsou charakterizovány frekvencí, amplitudou a tvarem vln a případně i stupněm jejich synchronie mezi různými oblastmi mozkové kůry.

V EEG záznamu se hodnotí frekvence a amplitudy vln. Obecně platí, že čím je větší frekvence, tím menší je amplituda vlny. Základními vlnami v EEG u dospělého člověka jsou vlny alfa, beta, gama a delta. Každý typ má určitou funkční charakteristiku [2].

Delta aktivita:

Frekvence 0,5-4 Hz, amplituda 20-200 μ V.

Pásmo delta je stavem velmi výrazného útlumu všech funkcí. Nalézáme se v ní během bezesného spánku, případně během bezvědomí způsobeného nemocí či úrazem.

V tomto stavu dochází k hluboké regeneraci všech životních funkcí a k hromadění energetických rezerv. Proces látkové přeměny probíhá pomaleji, ale zato velmi důkladně.

Théta aktivita:

Frekvence 4-8 Hz, amplituda 5-100 μ V.

Pásmo théta je stavem výrazného útlumu všech funkcí. Mysl ani tělo nereagují samy o sobě na žádné smyslové podněty. Často dochází k samovolnému uvolnění obrazových informací z podvědomé paměti a jejich selektivní prezentaci v mysli (sny).

Je charakteristická pro ospalost, usínání, hluboké uvolnění, ale i při poruchách pozornosti a lehké mozkové dysfunkci (LMD).

Alfa aktivita:

Frekvence 8-13 Hz, amplituda 5-100 μ V.

Pásmo alfa je stavem prostým jakéhokoli napětí a mysl nereaguje na podněty z vnějšího prostředí, ani se nezabývá soustředěným myšlením. Je to stav odpočinku těla při plné bdělosti, který je doprovázen i podmíněn příjemnými až slastnými pocity v návaznosti na zvýšenou tvorbu a vylučování přirozených opiátů z endokrinního aparátu. Setrvání v hladině alfa posiluje a regeneruje všechny životní funkce a vylučuje, až na vzácné výjimky, souběžnou přítomnost i rozvoj jakéhokoliv onemocnění.

Náš mozek produkuje elektrický proud této frekvence během bdění se zavřenýma očima, při tělesné a duševní relaxaci. Mizí při otevření očí a zaměřené pozornosti. V hladině alfa se spouštějí autosanační, sebohojivé síly těla, zlepšuje se schopnost učení.

Beta aktivita

Frekvence 14-30 Hz, amplituda 2-20 μV .

Pásma beta charakterizuje vědomé smyslové soustředění na naše okolí, připravenost reagovat, akceschopnost, menší či větší míra napětí a stav náladových změn až podrážděnosti. V hladině beta, vyšší či nižší, jsme po většinu dne a když používáme motorický aparát k pohybu.

Nižší pásma (do 18Hz) se objevují v běžném stavu bdělosti.

Do nejvyšších hodnot hladiny beta se dostaneme v souvislosti se značným úsilím soustředěním se na náročný úkol, ve vypjatých stresových situacích psychické i fyzické povahy, při psychoneurotických poruchách a chorobách.

Setrvávání v hladině beta nad 22-24Hz představuje pro celý organismus energetickou zátěž a způsobuje zhoršující se schopnost a kvalitu přeměny a regenerace tkání.

Pásma vyšší (kolem 30Hz) se pak objevují při podráždění, trémě, úzkosti, nebo vysoce náročných životních situacích.

Gama aktivita

Frekvence 30-60 Hz, amplituda 2-10 μV .

Pásma gama je nejvyšší frekvenční pásmo EEG. Je nestabilní aktivitou a epizody konstantní frekvence mají trvání 100 - 300 ms s intervalem 15 - 30 ms. Nejde tedy o klasický kontinuální rytmus. Každá epizoda gama rytmu je spojena s výraznou negativitou stejnosměrného potenciálu mozku.

6. Fourierova transformace

Pro analýzu frekvencí a amplitud v EEG signálu pomocí výpočetní techniky se používá Fourierova transformace, jejímž autorem je Jean Fourier. Pro následující kapitoly byly použity literární a internetové zdroje [10],[11],[12],[15],[16],[17],[18].

6.1. Podstata Fourierovy transformace

Každý signál je možno reprezentovat v jeho časové oblasti, ale také v duální frekvenční oblasti. Podstatou Fourierovy transformace je, že funkci v časové oblasti přetransformujeme do frekvenční oblasti a získáme tak její frekvenční spektrum. Fourierova transformace vychází z předpokladu, že každý signál lze vyjádřit jako superpozici nekonečně mnoha sinusových signálů. Výsledkem Fourierovy transformace jsou Fourierovy koeficienty, které nám říkají, jak daný sinusový průběh přispívá k celkovému signálu. Hlavním cílem Fourierovy transformace je tedy najít harmonické složky a jejich amplitudy.

6.2. Fourierova transformace pro spojitě signály

Fourierova transformace vyjádřená Fourierovým integrálem

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (1)$$

který existuje jen, jestliže je funkce $x(t)$ absolutně integrovatelná

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty \quad (2)$$

kde $\omega = 2\pi f$

Zpětná Fourierova transformace je potom definována vztahem

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega)e^{j\omega t} d\omega \quad (3)$$

6.3. Fourierova transformace pro diskrétní signály

Diskrétní Fourierova transformace nepracuje se spojitým signálem, ale se vzorky signálu snímanými v diskrétních časových okamžicích. Jsou určeny ke zpracování na počítačích. Vzorky jsou snímány s časovým odstupem Δt . Diskrétní Fourierova transformace se používá pro výpočet spektra libovolného signálu.

Vztah pro přímou diskrétní Fourierovu transformaci

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \quad (4)$$

kde $k=0,1,\dots,N-1$

Zpětná Fourierova transformace je potom

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N} \quad (5)$$

kde $n=0,1,\dots,N-1$

Tato transformace pracuje se vstupní komplexní řadou. Lze však vynulovat imaginární část, naplnit reálnou část a tak pracovat jen s reálným signálem. Výsledný signál je opět komplexní řadou, která se dále převádí do polárních souřadnic. Díky převedení do polárních souřadnic můžeme vypočítat amplitudu (MAG) a fázový posun (PHA)

$$MAG(k) = \sqrt{ReX(k)^2 + ImX(k)^2} \quad (6)$$

$$PHA(k) = \arctan\left(\frac{ImX(k)}{ReX(k)}\right) \quad (7)$$

6.3.1. Rychlá Fourierova transformace

Také označovaná jako FFT (Fast Fourier Transform). Tento algoritmus pro výpočet diskrétní Fourierovy transformace byl publikován v roce 1965 J. W. Cooleyem a J. W. Turkeyem. Hlavní výhodou FFT je její značná rychlost oproti DFT. Pro výpočet N hodnot pomocí DFT je potřeba provést N^2 komplexních násobení a

$N(N-1)$ komplexních sčítání, takže doba pro výpočet je dána přibližně časem pro výpočet N^2 operací. Naproti tomu doba výpočtu FFT je pouze $N/2 \cdot \log_2(N)$.

6.3.1.1. Algoritmus rychlé Fourierovy transformace

Algoritmus s decimací v čase je nejčastěji používaným FFT algoritmem. Je navržen pro délku transformace $N = 2^m$, kde m je přirozené číslo. Tyto algoritmy využívají periodičnosti a symetrií komplexní exponenciály. Tato exponenciála se pak označuje otáčecí činitel a označuje se W_N :

$$W_n = e^{\frac{-j2\pi}{N}} \quad (8)$$

Potom vztah pro DFT lze přepsat na

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad (9)$$

kde $k=0,1,\dots,N-1$

Posloupnost $x(n)$ rozdělíme na dvě posloupnosti, jednu obsahující pouze sudé členy $x_1(n)=x(2n)$ a druhou obsahující pouze liché členy $x_2(n)=x(2n+1)$, kde n je v intervalu $0 \dots (N/2)-1$.

Potom platí

$$\begin{aligned} X(k) &= \sum_{n=0}^{\frac{N}{2}-1} x(2n)W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)W_N^{(2n+1)k} = \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_1(n)W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_N^{2nk} \end{aligned} \quad (10)$$

Pomocí vztahu $W_N^{2kn} = W_{N/2}^{kn}$ [10] můžeme předchozí vztah upravit jako

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{\frac{N}{2}-1} x_1(n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n)W_{N/2}^{nk} = \\
 &= X_1(k) + W_N^k X_2(k)
 \end{aligned}
 \tag{11}$$

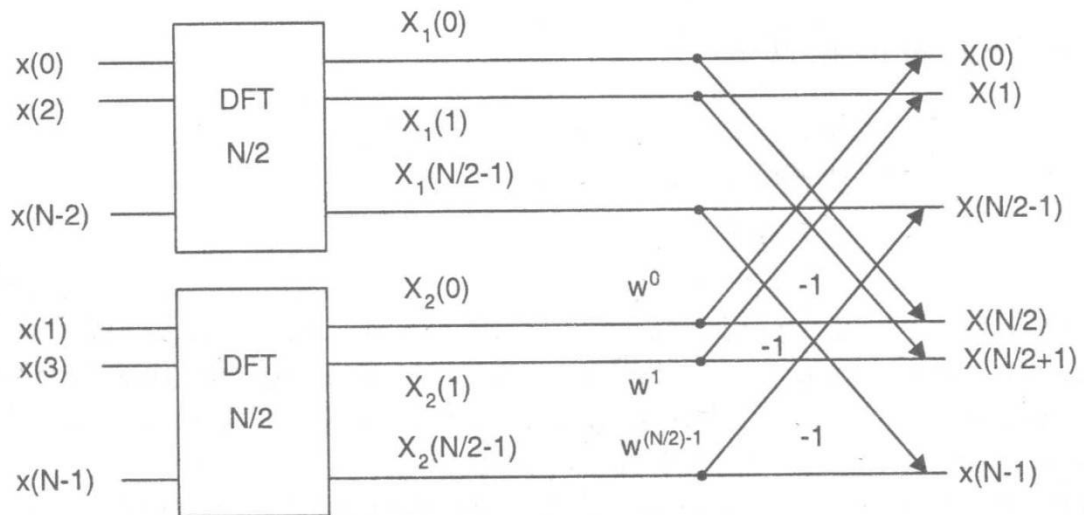
kde $k=0,1,\dots,N-1$

Vztah (11) je součet dvou $N/2$ bodových DFT, aplikovaných na vybrané posloupnosti sudých a lichých členů signálů, přičemž $X_2(k)$ je násobena mocninami W_N . Protože jsou $X_1(k)$ a $X_2(k)$ periodické s periodou $N/2$ a W_N^k je periodická s periodou N , lze vztah (11) rozepsat jako

$$\begin{aligned}
 X(k) &= X_1(k) + W_N^k X_2(k) \\
 X(k + N/2) &= X_1(k) - W_N^k X_2(k)
 \end{aligned}
 \tag{12}$$

kde $k=0,1,\dots,(N/2)-1$

Vztah (12) lze graficky znázornit jako obr. 9.



Obr.9. Algoritmus DIT FFT - první krok [10]

Rozdělením původní N bodové posloupnosti na dvě $N/2$ bodové posloupnosti se ušetří 50 % operací. Místo N^2 potřebujeme nyní pouze $N^2/2$ operací. Další úspory lze

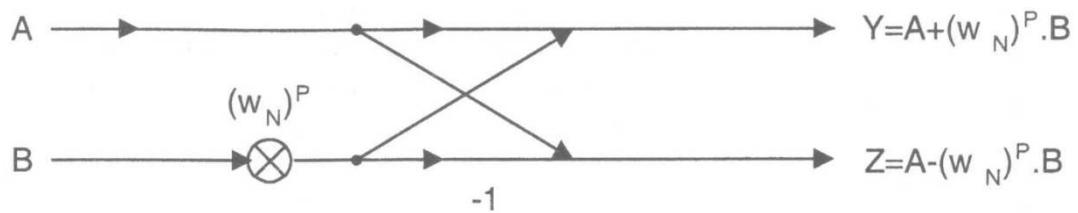
dosáhnout dalším dělením obou $N/2$ posloupností stejným způsobem jako byl signál rozdělen poprvé. Výsledkem jsou čtyři $N/4$ bodové DFT:

$$\begin{aligned} X_1(k) &= U(k) + W_N^{2k}V(k), \\ X_1(k + N/4) &= U(k) - W_N^{2k}V(k), \end{aligned} \tag{13}$$

$$\begin{aligned} X_2(k) &= Y(k) + W_N^{2k}Z(k), \\ X_2(k + N/4) &= Y(k) - W_N^{2k}Z(k), \end{aligned} \tag{14}$$

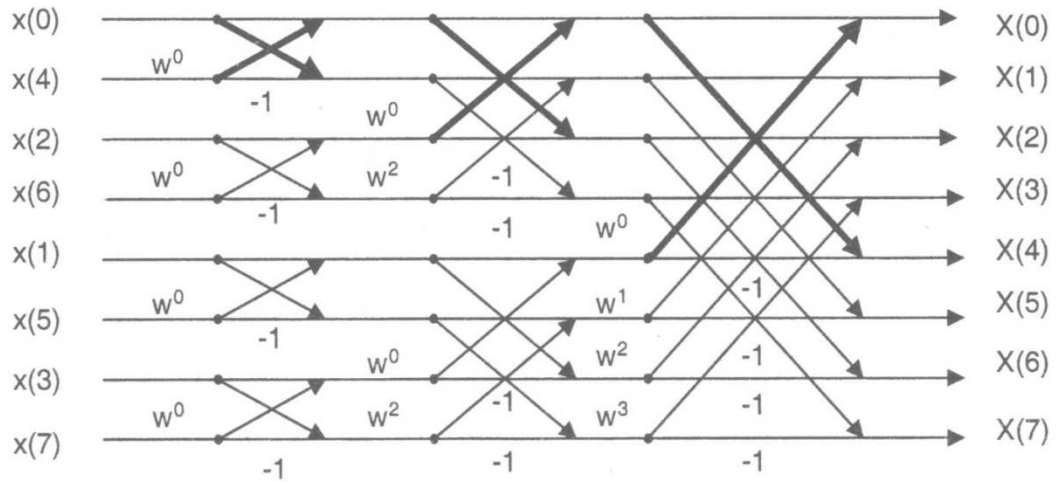
kde $k=0,1,\dots,(N/4)-1$

Vstupní posloupnosti $U(k)$ a $V(k)$ získáme pomocí DFT sudých a lichých členů $x_1(n)$, posloupnosti $Y(k)$ a $Z(k)$ pomocí sudých a lichých členů $x_2(n)$. Opakováním tohoto postupu se dostaneme až k základní dvojici rovnic, která vyjadřuje dvoubodovou DFT. Znázornění těchto rovnic v grafu signálových toků se podle tvaru nazývá motýlek DFT.



Obr.10. Motýlek DIF FFT [10]

Na obr.11. je znázorněno schéma algoritmu osmibodové FFT. Na vstupu jsou složky vektoru x v bitově invertovaném pořadí $x_0, x_4, x_2, x_6, x_1, x_5, x_3, x_7$, na výstupu jsou složky vektoru X . Graf obsahuje celkem 3 svislé sloupce motýlků, které odpovídají třem krokům rychlého algoritmu, v každém sloupci jsou čtyři motýlci.



Obr.11. Schéma algoritmu osmibodové DIT FFT [10]

V obecném případě bude sloupců $M = \log_2 N$, v každém bude $N/2$ motýlků. Budeme tedy rychlou Fourierovu transformaci počítat v M krocích vnějšího cyklu pro $m=1, \dots, M$. Pro pevné m provedeme operace naznačené v $N/2$ motýlcích, celkem tedy musíme realizovat $M \cdot N/2$ motýlků. V každém motýlku se provádí jedno komplexní násobení a dvě komplexní sčítání.

6.3.1.2. Praktická aplikace rychlé Fourierovy transformace

Vstupem pro algoritmus rychlé Fourierovy transformace je N hodnot číslicového signálu a výstupem N hodnot komplexních koeficientů spektra na normovaných frekvencích k/N , tj. na reálných frekvencích $k \cdot f_s / N$. Spektrum je periodické s periodou f_s , tzn. pro $k > N$ dostaneme tytéž hodnoty. Hodnoty koeficientů pro $N/2 < k < N$ jsou komplexně sdružené s prvními $N/2$ hodnotami a netřeba je proto počítat. Pokud vybraných N vzorků signálu netvoří jednu periodu (v praxi se to stává téměř vždy), jsou výsledné hodnoty pouze aproximací spektra, zatíženou různými chybami, zejména tzv. „rozmazání spektra“ (ve spektru se objeví neexistující složky). Nejlepším řešením je okénkovací funkce (window function). Výřez části signálu se provede násobením signálu obdélníkovou (resp. trojúhelníkovou, Hammingovou) funkcí.

6.4. Váhovací okna

V praxi se používá několik základních druhů oken, ve skutečnosti jich existuje velmi mnoho. Každé okno má různé vlastnosti a je vhodné pro jinou aplikaci. Všechny vztahy pro okna zde uvedené jsou v tzv. kauzální formě (tj. pro nezáporná n) [16].

Obdélníkové (jinak taky pravoúhlé či Dirichletovo) okno (Obr.12a) je nejjednodušším oknem, je popsáno vztahem

$$w(n) = 1 \quad (15)$$

kde $n=0,1,\dots,N-1$

Jak je vidět ze vztahu (15) toto okno pouze ořezává signál na N vzorků, ale jinak plně zachovává jeho tvar.

Hammingovo okno (Obr.12b) je v praxi jedno z nejpoužívanějších oken. Jeho předpis je

$$w(n) = 0,54 - 0,46 \cdot \cos\left(\frac{2\pi}{N} \cdot n\right) \quad (16)$$

kde $n=0,1,\dots,N-1$

Hanningovo (Hannovo nebo raised cosine) okno (Obr.12c) je speciální případ sinového okna. V praxi je opět velmi používané. Je popsáno vztahem

$$w(n) = 0,5 - 0,5 \cdot \cos\left(\frac{2\pi}{N} \cdot n\right) \quad (17)$$

kde $n=0,1,\dots,N-1$

Dalším oknem je trojúhelníkové (Barletovo) okno (Obr.12d) s předpisem

$$w(n) = \frac{2n}{N} \quad (18)$$

kde $n=0,1,\dots,(N/2)-1$

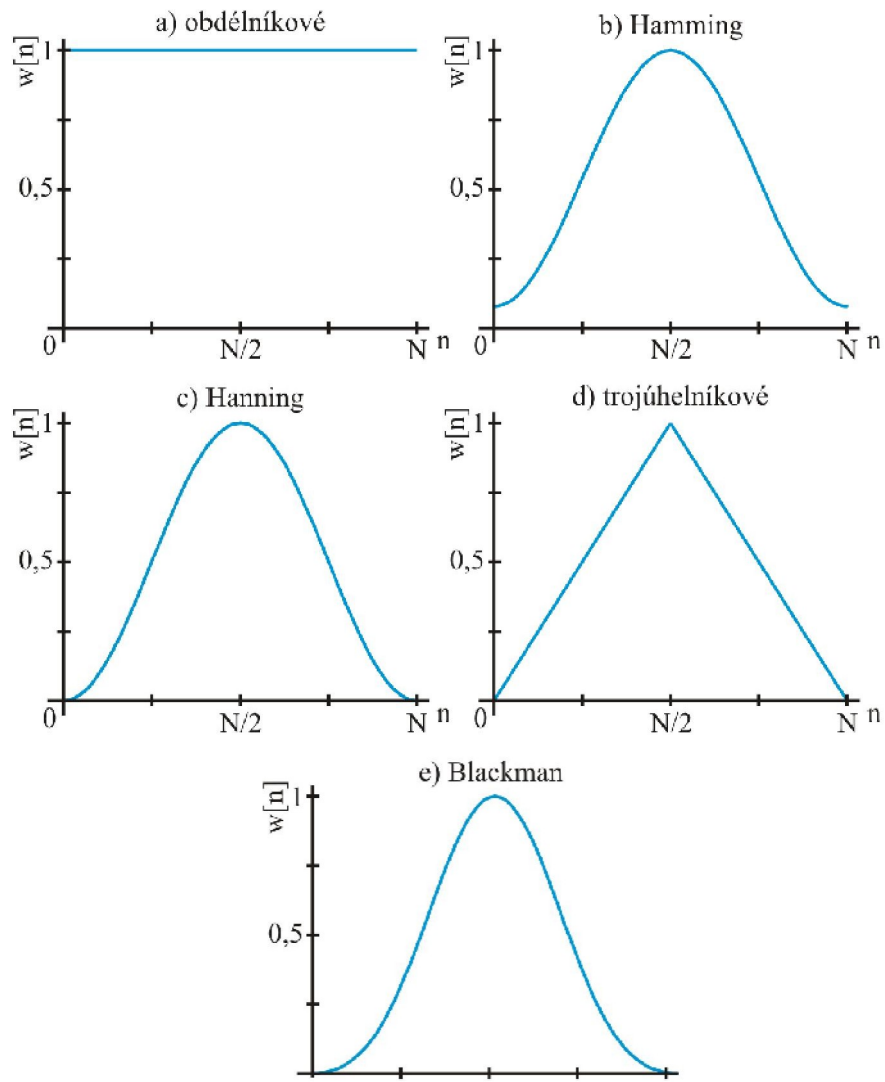
$$w(n) = 2 - \frac{2n}{N} \quad (19)$$

kde $n=N/2,\dots,N-1$

Posledním zde zmíněným oknem je Blackmanovo okno (Obr.12e) , které je popsáno vztahem

$$w(n) = 0,42 - 0,5 \cdot \cos\left(\frac{2\pi}{N} \cdot n\right) + 0,08 \cdot \cos\left(\frac{2\pi}{N} \cdot 2n\right) \quad (20)$$

kde $n=0,1,\dots,N-1$



Obr.12. Průběhy váhovacích oken [I6]

7. System audiovizuální stimulace s EEG zpětnou biologickou vazbou

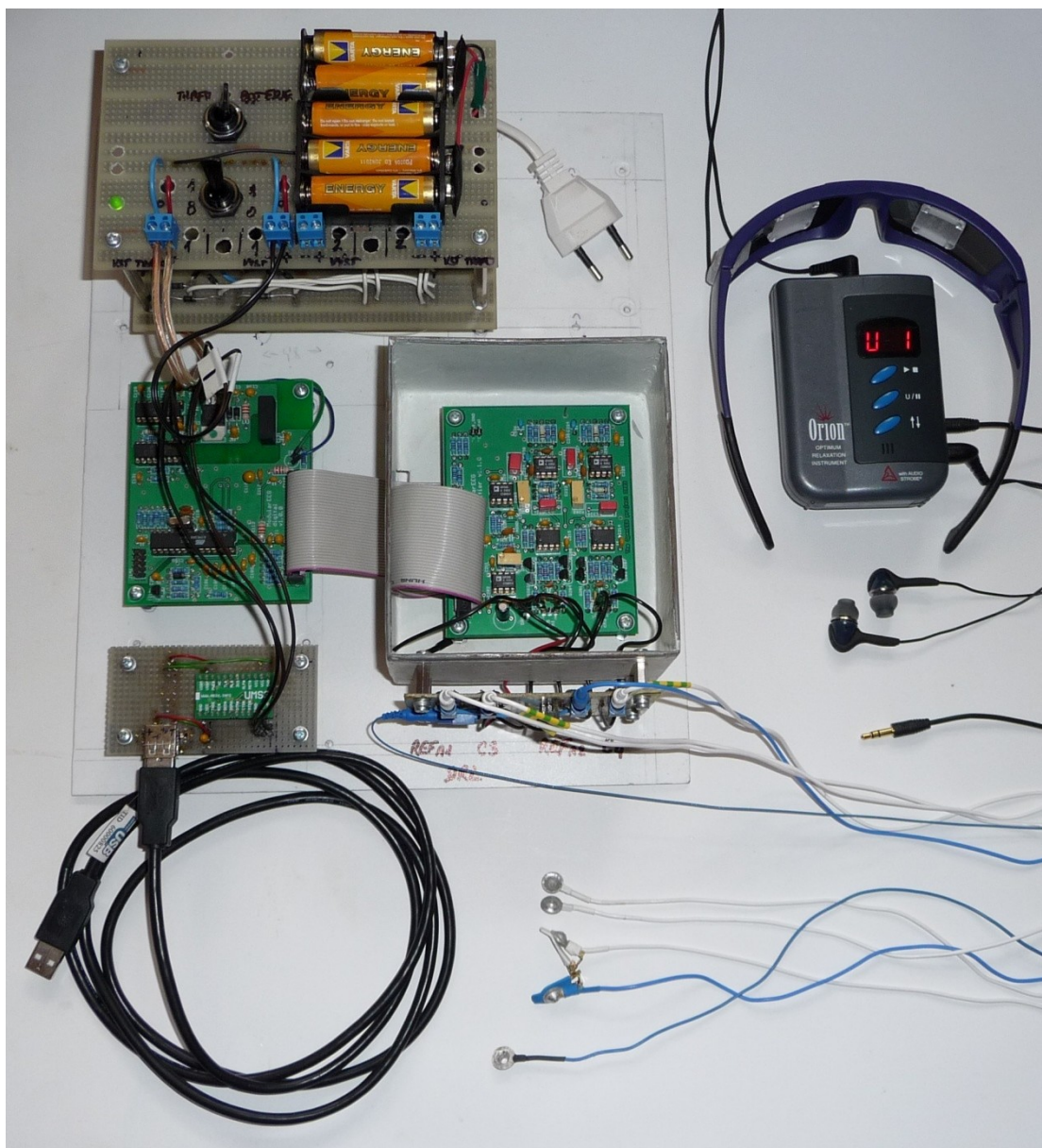
7.1. Výběr komponent

Pro realizaci systému audiovizuální stimulace s EEG zpětnou biologickou vazbou jsem se rozhodl vybrat následující komponenty:

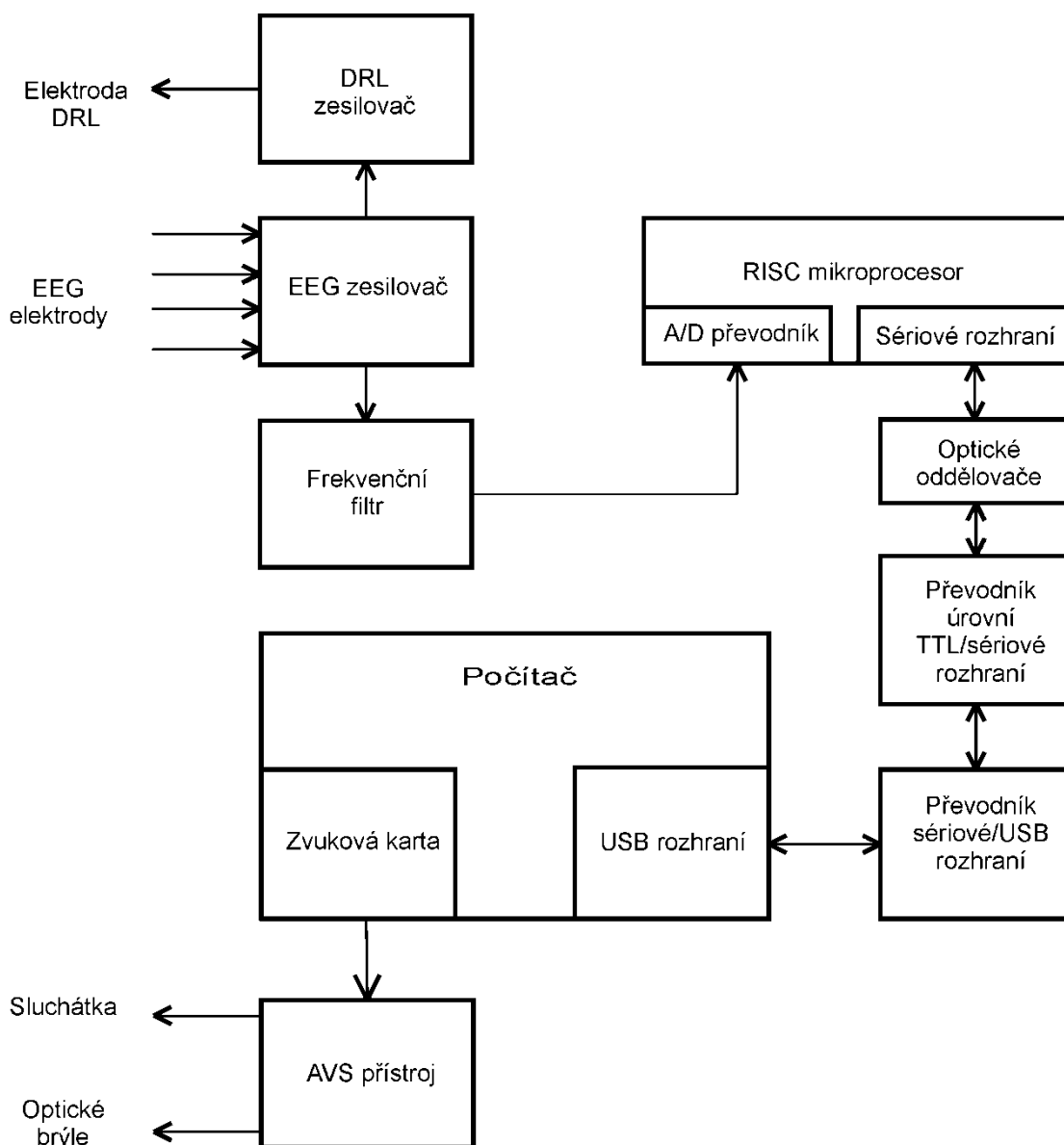
- Modul 2 kanálového EEG zesilovače, který vyrábí firma Olimex podle návrhu otevřeného projektu OpenEEG - Modular EEG, na kterém se podílejí jak obyčejní nadšenci EEG, tak doktoři z oboru psychologie a psychiatrie. Prostřednictvím internetu je to opravdu mezinárodní projekt. Hlavní důvod proč jsem použil tento modul bylo, že jsem jej měl k dispozici, má dobré parametry a to, že na stránkách projektu [I9] je bohatá dokumentace a velké diskusní fórum.
- Modul mikroprocesoru-AD převodníku, který je v podstatě mezistupeň mezi modulem EEG zesilovače a osobním počítačem. Pochází ze stejného zdroje a platí o něm vše popsané u modulu zesilovače.
- Převodník sériového rozhraní na USB jsem zvolil výrobek od firmy ASIX s názvem UMS3 pro jeho skvělé parametry a snadné použití v uživatelských aplikacích.
- AVS přístroj je přístroj s označením Orion od firmy Synetic Systems International, Inc. dodávaný do České republiky firmou Galaxy. Použil jsem jej, protože s ním mám zkušenosti a podporuje technologii AudioStrobe.
- Osobní počítač je srdcem celé aplikace. Musí obsahovat zvukovou kartu a USB konektor, což je dnes běžné vybavení. Doporučuji alespoň dvojjádrový procesor s frekvencí 2 GHz a operační paměť 1 GB a více (dle operačního systému). Celá softwarová část běžící na osobním počítači je naprogramována v jazyce Delphi7 Personal, ve kterém v současné době programuji všechny své aplikace po přechodu z jazyku Pascal. Aplikace běží bez problémů pod operačním systémem Windows XP i Vista.

7.2. Zapojení komponent

Na obr. 13 je reálné zapojení systému audiovizuální stimulace s EEG zpětnou biologickou vazbou včetně EEG elektrod a AVS přístroje, Jediné technické zařízení, které tu není zobrazeno je osobní počítač do něhož se připojuje USB kabel a konektor Jack.



Obr.13. Celkové zapojení systému audiovizuální stimulace s EEG zpětnou biologickou vazbou včetně EEG elektrod a AVS přístroje



Obr.14. Blokové schéma zapojení systému audiovizuální stimulace s EEG zpětnou biologickou vazbou

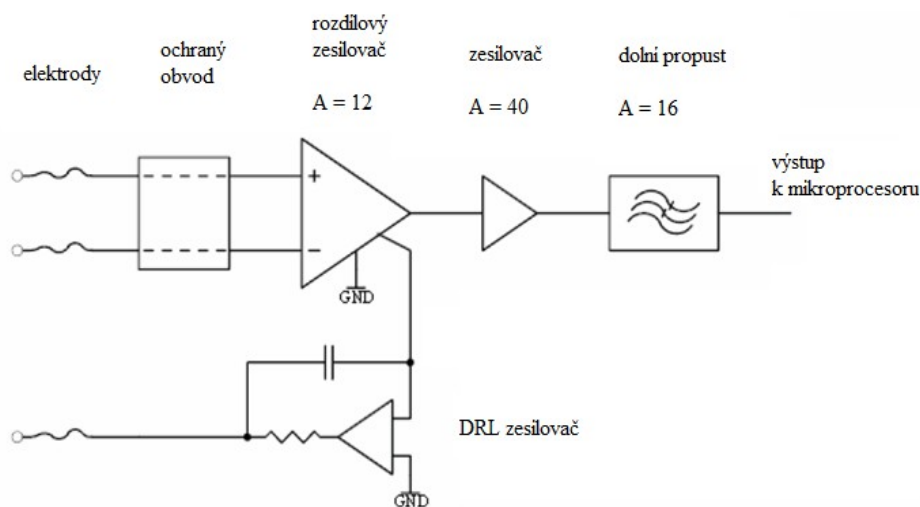
S popisem zapojení začneme u uživatele, který má na hlavě přichycené čtyři elektrody (2 snímací a 2 referenční v unipolárním zapojení), které snímají EEG signál z povrchu hlavy. Signál je přiveden do EEG zesilovače, který jej zesílí z napětí v řádu μV na jednotky voltů. Ze zesíleného signálu jsou odfiltrovány rušivé složky elektromagnetického rušení pomocí frekvenčního filtru. Uživatel má na sobě přidělanou ještě jednu elektrodu, většinou na pravé noze, ale dá se přichytit i na ruku se stejnou funkcí. Je to elektroda vedoucí z DRL zesilovače. Ten slouží k potlačení souhlasných rušivých signálů, a tak výrazně přispívá k menšímu zarušení EEG

signálu vycházejícího ze zesilovače. Zesílený signál je dále digitalizován pomocí AD převodníku, který je v této aplikaci součástí mikroprocesoru. Mikroprocesor posílá dál digitalizovaný signál směrem k počítači prostřednictvím sériového rozhraní. Optické oddělovače jsou zařazeny kvůli větší bezpečnosti uživatele i proto, aby snížily rušení šířené od počítače. Dále jsou data převedena z TTL úrovní signálu na úroveň signálu sériového rozhraní. Sériové konektory jsou dnes již zastaralé a některé moderní počítače je už dokonce ani nemají. Toto řeší poslední obvod před počítačem, který je převodníkem mezi sériovým rozhraním a USB rozhraním. USB kabel z tohoto modulu vede do počítače, kde je provedena poslední část zpracování naměřeného signálu a to jeho analýza. K tomu slouží program napsaný speciálně pro tuto aplikaci. Jeho výsledkem je grafické znázornění naměřených dat. Program také z analyzovaných dat vypočítává nejvhodnější frekvenci pro zpětnovazebnou stimulaci pomocí přístroje AVS tak, aby tato frekvence byla rovna aktuální dominantní frekvenci EEG snímaného signálu a přitom o kousek (maximálně o 2 Hz) posunuta blíže cílové relaxační frekvenci. Informaci jakou frekvenci má AVS přístroj použít pro vizuální stimulaci, předává počítač pomocí zvukového signálu dvojitých binaurálních rytmů, ve kterých je zakódována na prahu slyšitelnosti (technologie AudioStrobe).

7.3. Modular EEG

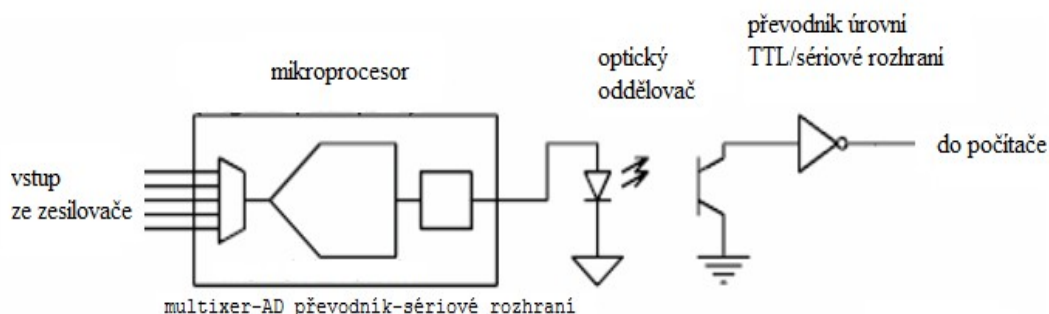
Modular EEG je otevřený internetový projekt věnovaný měření a analýze EEG signálů. Autory popisovaného zapojení jsou ing. Joerg Hansmann a Andreas Rob [110].

Projekt se skládá ze dvou modulů. První je modul 2 kanálového EEG zesilovače sloužící k předzpracování (zesílení a filtraci) EEG signálu. Obsahuje ochranný obvod, rozdílový zesilovač, neinvertující zesilovač, dolní propust a DRL zesilovač.



Obr.15. Blokové schéma modulu EEG zesilovače pro jeden kanál

Druhý modul je modul mikroprocesoru-AD převodníku, který slouží k digitalizaci zesíleného EEG signálu a odeslání naměřených hodnot do osobního počítače prostřednictvím sériového rozhraní.



Obr.16. Blokové schéma modulu mikroprocesoru-AD převodníku

7.3.1. Snímací elektrody

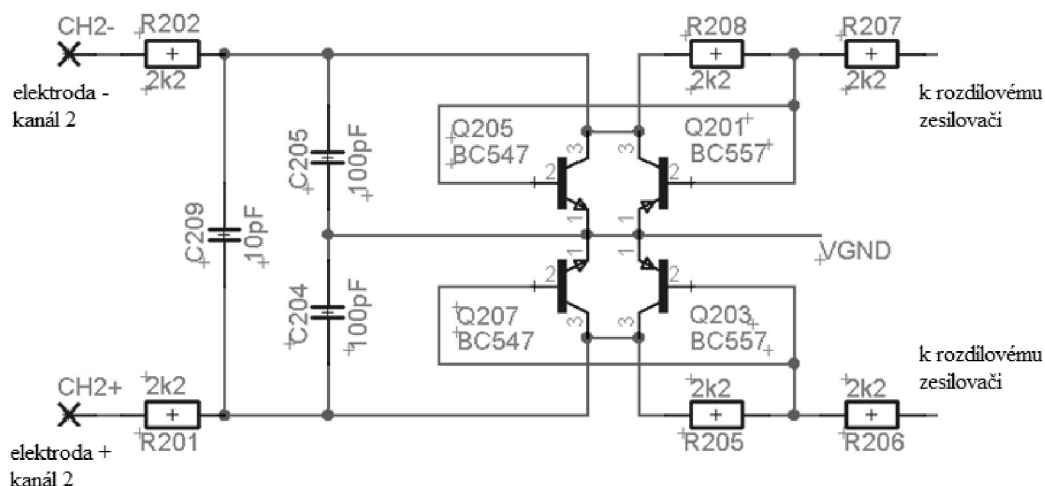
Pro snímání EEG signálu byly použity dvě cínové kalíškové elektrody umístěné C3 a C4 (systém 10-20) a dvě cínové ušní elektrody umístěné A1 a A2 v unipolárním zapojení snímání. U prvního kanálu tvoří elektroda umístěná na levém ušním lalůčku (A1) referenční elektrodu a elektroda umístěná na skalpu nad levou hemisférou (C3) snímací elektrodu. U druhého kanálu tvoří elektroda umístěná na pravém ušním lalůčku (A2) referenční elektrodu a elektroda umístěná na skalpu nad pravou hemisférou (C4) snímací elektrodu. Snímací místa byla zvolena stejně jako u aplikace EEG Biofeedback od firmy Alien, která používá snímané EEG signály pro ovládání hry běžící na počítači. S tímto biofeedbackem mám osobní zkušenosti, takže vím, že snímání na těchto umístěních je velmi efektivní.

7.3.2. EEG zesilovač

EEG signál snímaný elektrodami na povrchu hlavy má velikost jen několik málo mikrovoltů a je potřeba jej zesílit několik tisíckrát kvalitním zesilovačem, aby bylo možné jeho další zpracování.

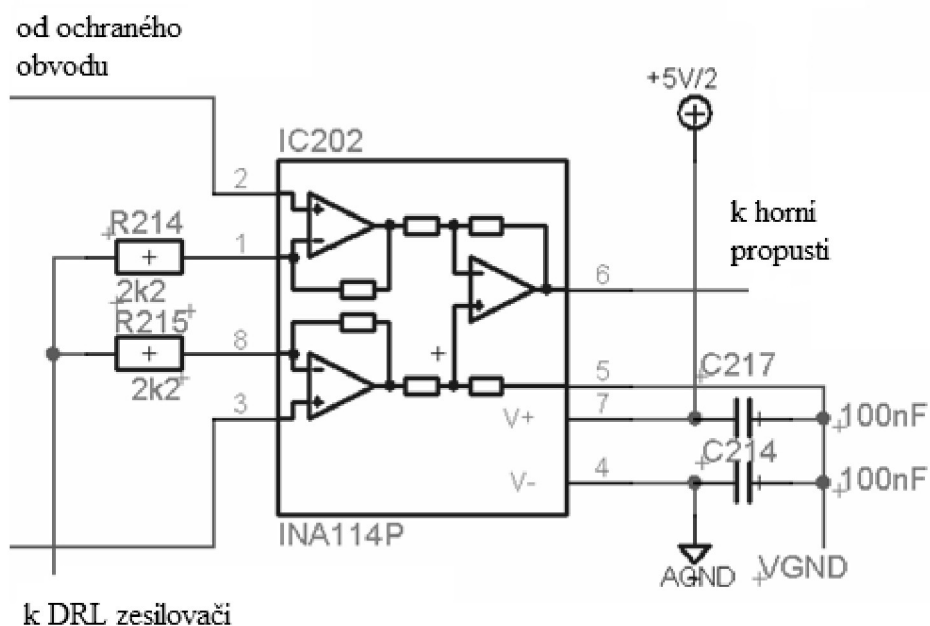
Celý zesilovač je uložen v kovové krabici spojené s aktivní zemí VGND. Tím je zapojení chráněno před elektrostatickým nábojem, ale také hlavně před elektromagnetickým rušením z okolí.

Na počátku zapojení se nachází ochranný obvod. První část obvodu tvoří kondenzátory (C209, C204, C205), které slouží k potlačení radiofrekvenčních signálů vstupujících do systému přes kabely elektrod. Druhou část tvoří tranzistory (Q202, Q204, Q206 Q208), které zamezují vstupu napětí většího než 0,7 V a zároveň omezí průchod proudu uživatelem v případě zkratu mezi výkonovou částí a vstupem zesilovače.



Obr.17. Ochranný vstupní obvod

První stupeň zesilovače tvoří rozdílový zesilovač, který zesílí rozdíl obou vstupních napětí (U_{REF1} a U_{C3} resp. U_{REF2} a U_{C4}) 12 krát.



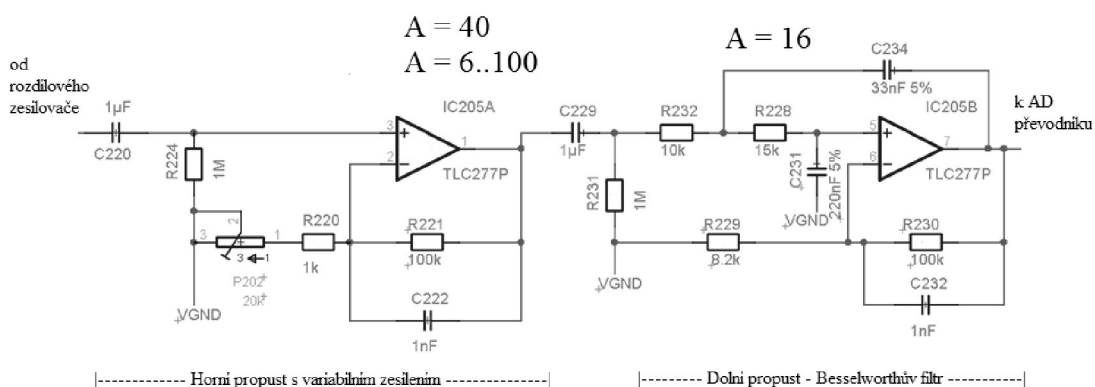
Obr.18. Rozdílový zesilovač

Poté je signál zesílen přibližně 40 krát v druhém zesilovacím stupni, který tvoří neinvertující zesilovač. Součástí druhého stupně je také horní propust. Ta

odstraňuje stejnosměrnou složku signálu, protože některé materiály elektrod jako zlato nebo ocel, jsou polarizovatelné. To znamená, že se na povrchu elektrody může nahromadit elektrický náboj o relativně velkém stejnosměrném napětí v řádu několika set milivoltů. Teoreticky, kdybychom zesílili vstupní signál o napětí 200 mV 480 krát, dostali bychom na výstupu zesilovače napětí o velikosti 96 V. To není možné dopustit, protože maximální napětí na výstupu má být ideálně do 2,5 V. Výstupní signál by uvázl v maximální vysoké nebo nízké úrovni, obvykle +/- 2,5 V a neobsahoval by žádný EEG signál, proto horní propust s $f_M = 0,16$ Hz řeší tento problém.

7.3.3. Frekvenční filtr

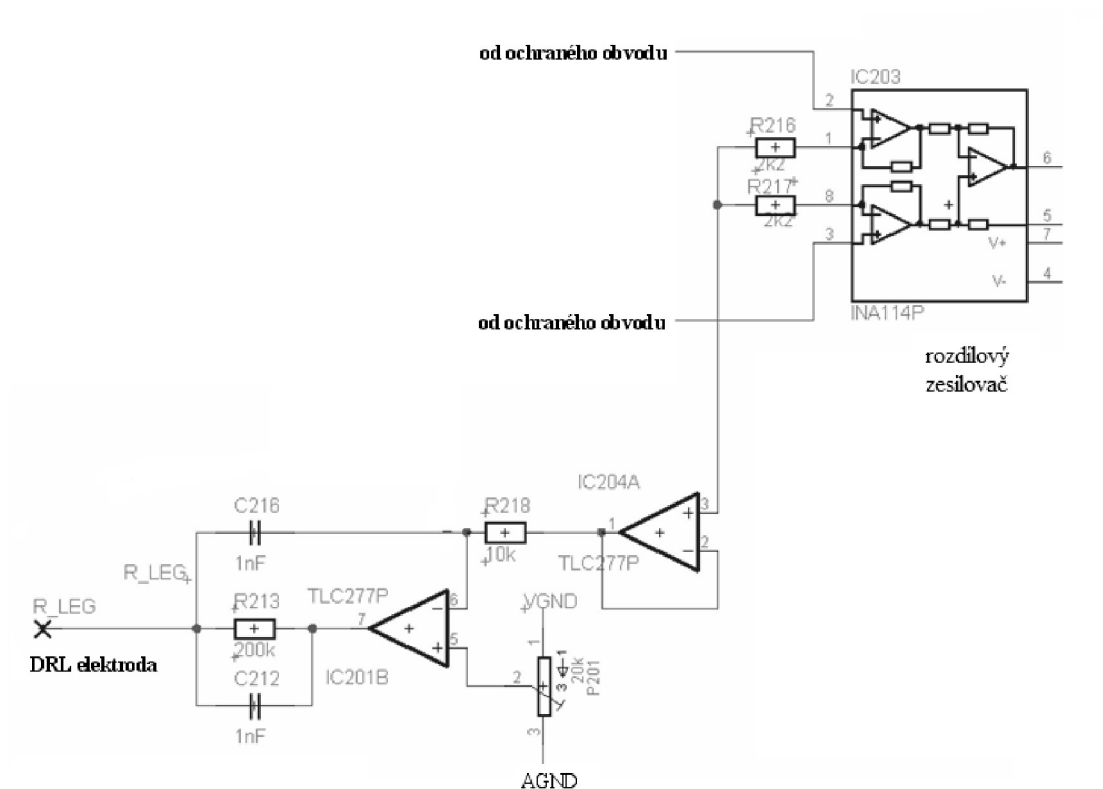
Protože snímaný signál má velmi nízkou amplitudu, velmi snadno se ponoří do elektromagnetického šumu, jehož hlavní složku tvoří síťové rušení o frekvenci 50 Hz pocházející z elektroinstalace v našem okolí. Proto je kromě zesílení signálu třeba odfiltrování těchto rušivých složek dolní propustí (Besselworthův filtr s $f_M=59$ Hz). Bohužel toto zapojení je navrženo pro americkou síťovou frekvenci 60 Hz a odstranění tohoto nežádoucího šumu je řešeno až při počítačové analýze signálu. Pro naše podmínky by mělo být $f_M=49$ Hz nebo ještě lépe $f_M=39$ Hz, protože frekvence nad 39 Hz jsou pro náš účel nezajímavé. V rámci filtrace je provedeno i 16 násobné zesílení signálu. Frekvenční přenosová charakteristika filtrů modulu EEG zesilovače je uvedena v příloze P9.



Obr.19. Schéma zapojení horních a dolních propustí

7.3.4. DRL zesilovač

Vedle signálového zesilovače a filtru obsahuje zapojení ještě třetí zesilovač, který směřuje "v opačném směru" než zbytek zapojení. DRL zesilovač slouží k redukci rušivých souhlasných signálů jako je síťová frekvence 50 Hz. Zapojení tohoto zesilovače bylo dřív používáno jen u měření EKG, kdy DRL výstup byl připojen k pravé noze.



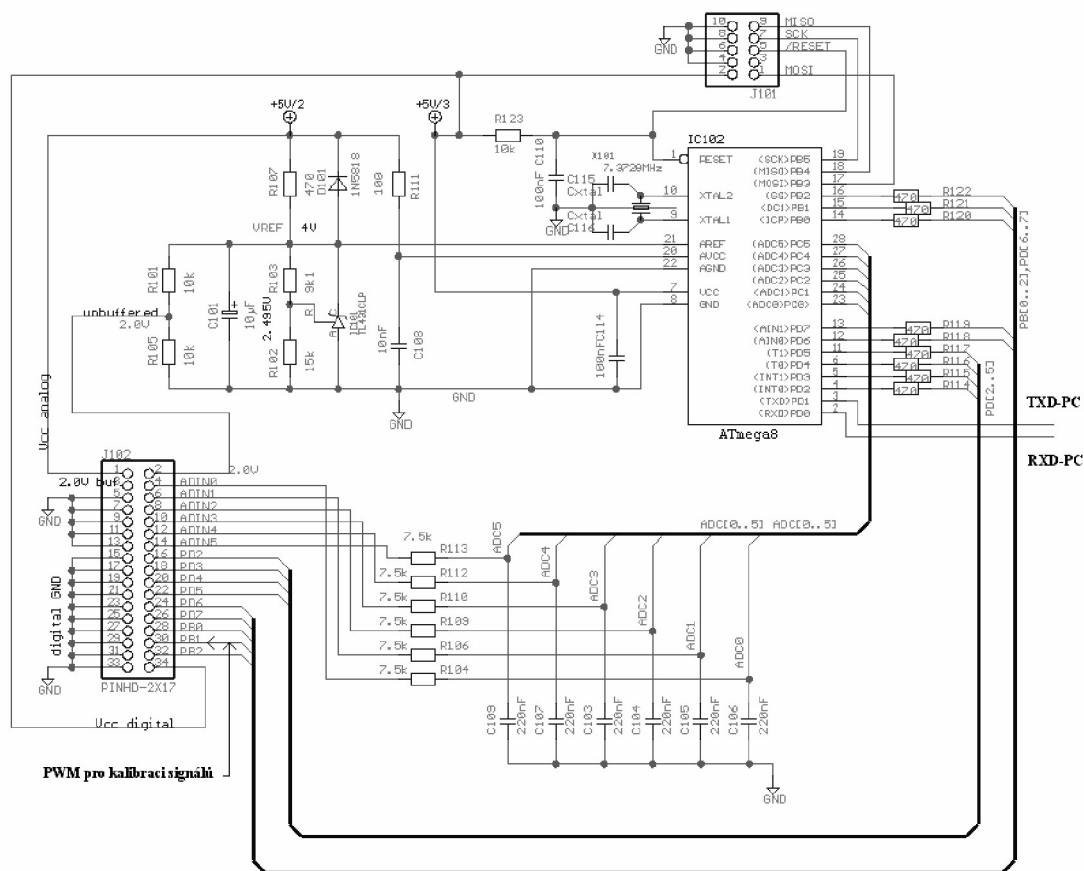
Obr.20. Schéma zapojení DRL zesilovače

DRL výstup nahrazuje zemnicí elektrodu, která se používala u starších verzí EEG návrhů a může zmenšit hlavní části šumu až 100 krát, než kdyby byl použit samotný zesilovač bez tohoto zapojení.

7.3.5. Mikroprocesor - AD převodník

Po filtrování je analogový signál připraven pro převod na digitální pomocí analogově číslicového převodníku, který je v našem případě umístěn uvnitř mikroprocesoru Atmega8. Po převodu posílá mikroprocesor počítači digitalizovaný EEG signál přes sériové rozhraní (RXD,TXD).

Součástí zapojení je i oscilační krystal generující pro procesor taktovací frekvenci 7,3729 MHz a zdroj referenčního napětí 4 V pro AD převodník.



Obr.21. Schéma zapojení mikroprocesoru

Autoři návrhu zvolili mikroprocesor ATmega8 od firmy Atmel z těchto důvodů:

- Je dostatečně rychlý pro danou aplikaci
- Umožňuje zároveň analogově-digitální převod 6 kanálů s rozlišením 10 bitů
- Programování je velmi snadné

Mikroprocesor již od výrobce modulu obsahoval firmware, který byl vytvořen pro projekt OpenEEG - Modular EEG. Existuje několik modifikací firmware, které se liší hlavně datovým přenosovým protokolem pro odesílání naměřených dat do počítače. Přenosový protokol P2 je nejběžnější protokol, používá 17 bajtů pro jeden přenosový cyklus a byl použit v této aplikaci.

Tab.1. Přenosový protokol P2:

Číslo bajtu	Význam
1.	synchronizační bajt, hodnota vždy \$a5
2.	synchronizační bajt, hodnota vždy \$5a
3.	označuje verzi protokolu, hodnota vždy 0x02
4.	číslo přenášeného datového rámce, hodnota na počátku 0x00 a po každém rámci dat se zvětší o 1, při hodnotě 255 se vrací na hodnotu 0, slouží pro kontrolu správného načítání do počítače
5.	formát 000000bb, kde bb jsou 2 horní bity hodnoty kanálu 1
6.	8 spodních bitů hodnoty kanálu 1
7.	formát 000000bb, kde bb jsou 2 horní bity hodnoty kanálu 2
8.	8 spodních bitů hodnoty kanálu 2
9.	formát 000000bb, kde bb jsou 2 horní bity hodnoty kanálu 3
10.	8 spodních bitů hodnoty kanálu 3
11.	formát 000000bb, kde bb jsou 2 horní bity hodnoty kanálu 4
12.	8 spodních bitů hodnoty kanálu 4
13.	formát 000000bb, kde bb jsou 2 horní bity hodnoty kanálu 5
14.	8 spodních bitů hodnoty kanálu 5
15.	formát 000000bb, kde bb jsou 2 horní bity hodnoty kanálu 6
16.	8 spodních bitů hodnoty kanálu 6
17.	bajt s digitálními vstupy (PD2..PD5) ve spodních 4 bitech

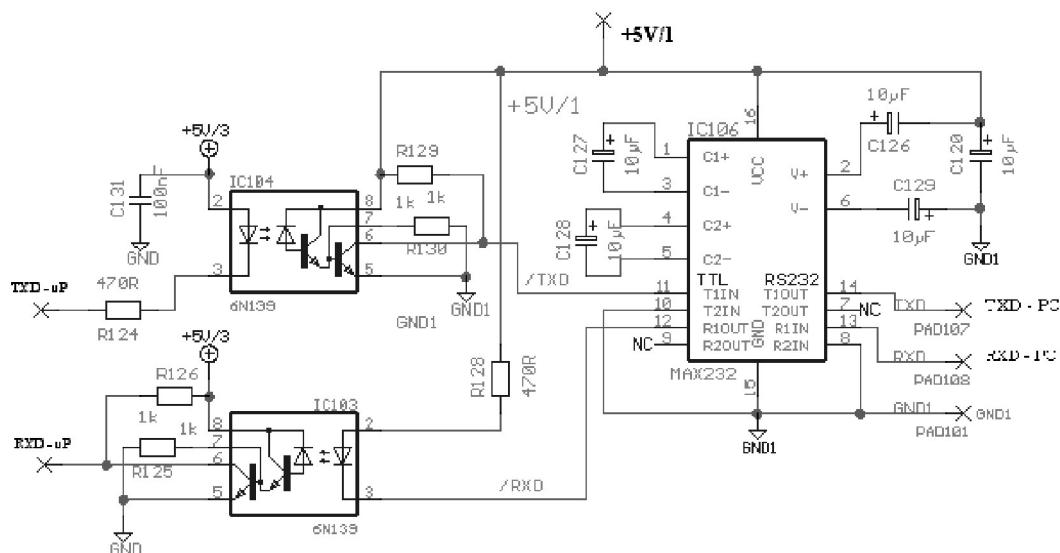
V následující tabulce jsou uvedeny parametry sériového přenosu z mikroprocesoru. Takto musí být nastaven i sériový příjem na počítači, přestože se celé zapojení připojuje k počítači přes USB rozhraní, počítač vnímá vstupní zařízení díky převodníku UMS3 a jeho ovladačům jako sériové rozhraní RS-232.

Tab.2. Parametry sériového přenosu mezi počítačem a mikroprocesorem:

Přenosová rychlost	57600 baudů
Velikost dat	8 bitů
Parita	Ne
Počet stop bitů	1
Regulace toku dat	Ne

7.3.6. Optické oddělovače

Pro zamezení elektrického rušení směrem od počítače a pro elektrické oddělení obvodů, je vstupní i výstupní datová linka oddělena pomocí optočlenů.



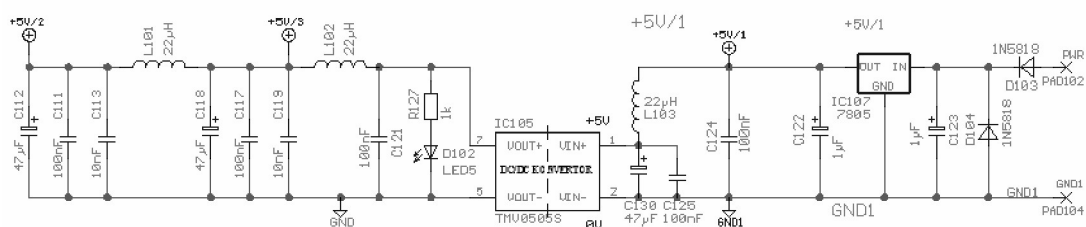
Obr.22. - Schéma zapojení optických oddělovačů a převodníku úrovní TTL/sériové rozhraní

7.3.7. Převodník úrovní TTL/sériové rozhraní

Za optočleny je zapojen obvod MAX232, který mění napětí TTL úrovní (0-5V) na napětí sériového rozhraní (+/-12V).

7.3.8. Napájení modulů

Doporučené napájení obvodů je 9 - 12 voltů realizované z bateriových článků. V mém případě je napájení řešeno z 5 kusů bateriových článků typu AA (7,5 V). Toto napětí je dostatečně velké na to, aby bylo stabilizováno na napětí 5 V. Mé zapojení obsahuje i možnost síťového napájení, ale tato varianta byla určena hlavně pro zkušební provoz (ladění zapojení), aby nebylo potřeba používat drahé baterie. Použití baterií kromě bezpečnosti má i výhodu ve snížení síťového brumu v obvodu. Důležitým prvkem napájecího obvodu je izolační člen DC/DC konvertor, který slouží k ochraně uživatele před přepětím.



Obr.23. Schéma zapojení napájecího obvodu

Napájecí napětí má několik úrovní:

+5V/1 - neizolované regulované napětí 5V

GND1 - neizolovaná zem

+5V/3 - izolované napětí 5V pro digitální část

+5V/2 - izolované a filtrované napětí 5V pro analogovou část, včetně zesilovačů

GND - digitální zem

VGND - virtuální zem, o napětí 2V vůči AGND

AGND - GND v digitální části zapojení

7.3.9. Technické parametry Modular EEG

Tab.3. Proudové a napěťové parametry zapojení

Proudový odběr - deska s mikroprocesorem	56 mA
Proudový odběr - celé zařízení (s 2 aktivními kanály)	73 mA
Vstupní maximální napěťový rozsah	+/-256 μ V
Napěťová izolační pevnost	2500V (1 minutu)
Stálá napěťová izolační pevnost	480V

Poznámky:

Napěťová izolační pevnost DC/DC konvertoru použitého v napájecím obvodu je 3000 V po dobu jedné minuty, déle než jedna minuta 1100 V.

Napěťová izolační pevnost u optického oddělovacího členu je 2500V po dobu jedné minuty, déle než jedna minuta 480V.

Tab.4. Parametry EEG zesilovače

Napěťové zesílení	7812,5
Horní propust	2. řádu, $f_m = 0,4$ Hz
Potlačení souhlasných signálů (DRL) pro $f=50/60$ Hz	> 40dB CMRR

Poznámky:

U hodnoty napěťového zesílení je uváděna typická hodnota, ve skutečnosti záleží na přesnosti použitých součástek. Celkové zesílení lze také měnit pomocí odporových trimrů P202, P203. V mém zapojení mi vyšlo napěťové zesílení obou kanálů $A_{u1}=9500$, $A_{u2}=9500$.

Tab.5. Parametry AD převodníku (Atmel ATmega8)

Počet kanálů	6
Rozlišení	10 bitů
Integrální nelinearita	0,5 LSB
Absolutní přesnost	+/- 2 LSB
Doba převodu	13-260 μ s

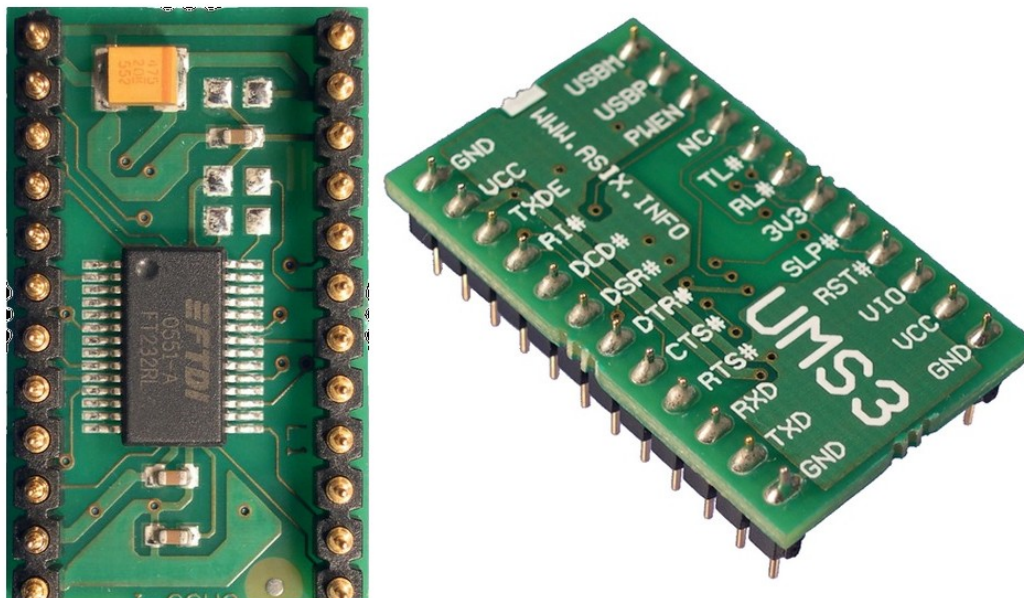
7.3.10. Oživení Modular EEG

Zprovoznění Modular EEG není složité, je shrnuto v následujících třech krocích.

1. Obě desky propojíme pomocí 34 žilového kabelu, který se používá jako datový kabel k IDE diskům v počítačích. Propojíme konektory RXD, TXD, GND mezi modulem UMS3 a deskou AD převodníku.
2. Po zapojení napájení a zapojení k počítači pomocí USB kabelu, je třeba nastavit zesílení druhého zesilovacího stupně. K tomu budeme potřebovat program Electricguru volně stažitelný z internetu, který zobrazuje snímaný signál. Zapojíme výstupy U_CAI na vstup "+" 1. kanálu a Cal_GND na "-" 1. kanálu a pomocí odporového trimru P203 nastavíme zesílení tak, aby se maximální rozkmit zobrazovaného kalibračního signálu v programu Electricguru zobrazoval na hodnotách 262 a 762. Poté zapojíme kalibrační signál na vstupy 2. zesilovače a provedeme to samé jako u prvního kanálu, jen regulační trimr je tentokrát odporový trimr P202.
3. Posledním krokem je nastavení ořezávání signálu pomocí DRL. Spojíme všechny signálové vstupy a zapojíme je na výstup DRL. Digitální multimetr zapojíme mezi výstup DRL a výstup VGND. Otáčíme jezdcem odporového trimru P201 tak dlouho, až multimetr nebude ukazovat žádné napětí (maximálně desetiny mV). Tím je nastavení dokončeno.

7.4. Převodník USB/sériové rozhraní

Jako převodník mezi sériovým rozhraním a USB rozhraním je použit modul UMS3, který je založen na integrovaném obvodu FT232RL firmy FTDI.[I12]



Obr.24. Modul UMS3 (vlevo ze spodu, vpravo z vrchu)

7.4.1. Vlastnosti modulu UMS3

- možnost plného hardwarového řízení toku dat
- přenos dat rychlostí až 1 MBd (pro RS232)
- napájení z USB sběrnice nebo z aplikace napětím 3,3 V až 5,25 V
- komunikace s aplikací v napěťových úrovních 1,8 V - 5,25 V
- vyrovnávací paměť 256 bajtů pro příjem / 128 bajtů pro vysílání
- protokol USB 1.1, USB 2.0 kompatibilní, Full Speed
- identifikace pomocí jména zařízení, sériového čísla zařízení aplikace
- požadovaný proudový odběr, sériové číslo modulu a další náležitosti týkající se USB se do paměti modulu programují přes rozhraní USB pomocí utility MPROG volně stažitelné na internetu
- každý ze signálů sběrnice UART je možné invertovat do jeho negativní podoby

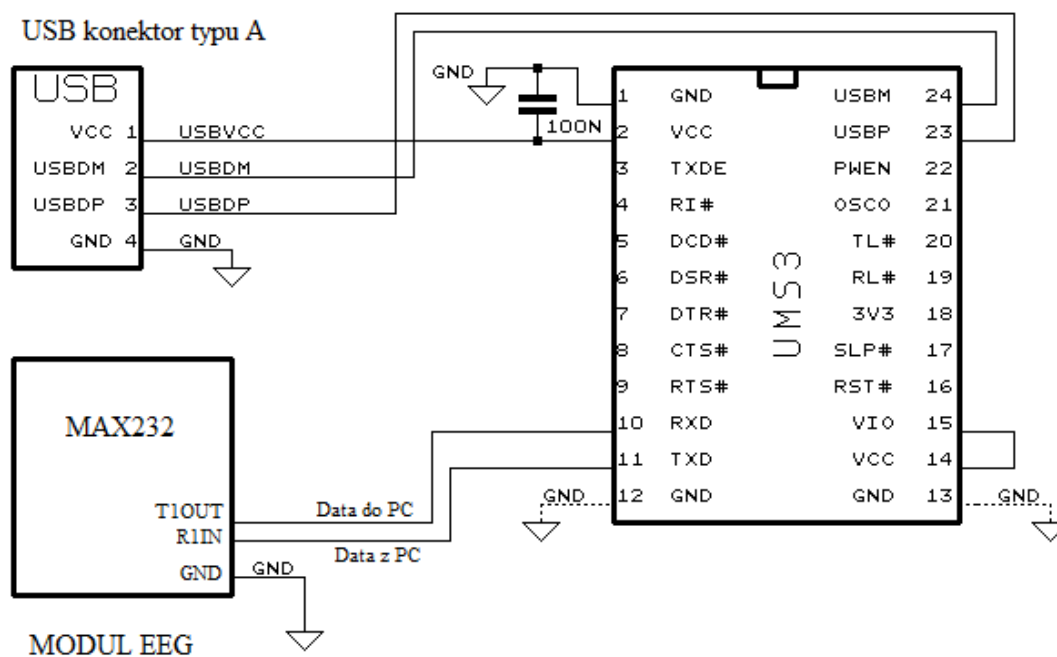
Tab.6. Popis pinů UMS3

Číslo pinu	Označení	Popis
1	GND	Zem
2	VCC	Napájení +3,3 V až 5,25 V
10	RXD	Data přijímaná z aplikace do PC
11	TXD	Data vysílaná z PC do aplikace
12	GND	Zem
13	GND	Zem
14	VCC	Napájení +3,3 V až 5,25 V
15	VIO	Napájení +1,8 V až +5,25 V výstupní
23	USBP	USB datový signál plus (I/O)
24	USBM	USB datový signál mínus (I/O)

Poznámka: V této tabulce jsou uvedeny jen ty piny, které byly aktivně použity v mém zapojení.

7.4.2. Zapojení UMS3

Zvolil jsem zapojení, které nepotřebuje externí napájení a je napájeno přímo z USB sběrnice. Odebíraný proud nesmí překročit 100 mA., což toto zapojení splňuje, protože napájíme v podstatě jen modul, který odebírá maximálně 15 mA. Jako konektor USB jsem zvolil typ A.



Obr.25. Schéma zapojení modulu UMS3

7.5. Programová podpora projektu

Programy, které budou v následujících kapitolách popsány jsem napsal speciálně pro tuto aplikaci v programovacím jazyce Delphi 7. Pracuje bez problémů pod operačním systémem Windows XP Professional i 64 bitovou verzí Vista Business. Při tvorbě programů jsem se snažil, aby bylo vytvořeno přátelské a přehledné prostředí pro uživatele i když tento program slouží hlavně k otestování sestavného přístroje a dalšímu zkoumání použití EEG zpětné vazby s využitím AVS. Bez jakýchkoliv úprav je aplikace použitelná jen pro zapojení výše popsané.

7.5.1. Struktura programů

Celá aplikace se skládá v podstatě ze tří současně běžících programů nazvaných Mexický příliv, FFTnew a Generátor. Místo jednoho programu bylo zvoleno několik současně běžících programů proto, aby operace vyžadující více času pro svoje provedení nebrzdily provádění jiných procesů, které mohou běžet paralelně. Takto pracují programy současně, což moderní operační systémy a procesory velmi dobře podporují. Programy si mezi sebou vyměňují jen data nebo ovládací informace a jinak pracují zcela nezávisle na sobě.

Hlavní program nazvaný Mexický příliv (verze 024) je hlavní program celé aplikace a je to také jediný program, který uživatel spouští a ovládá. Ostatní dvě aplikace se spouštějí samy v minimalizovaném módu a samy se i ukončují. Jsou ovládány z hlavního programu, ze kterého získávají i potřebná data pro svou funkci.

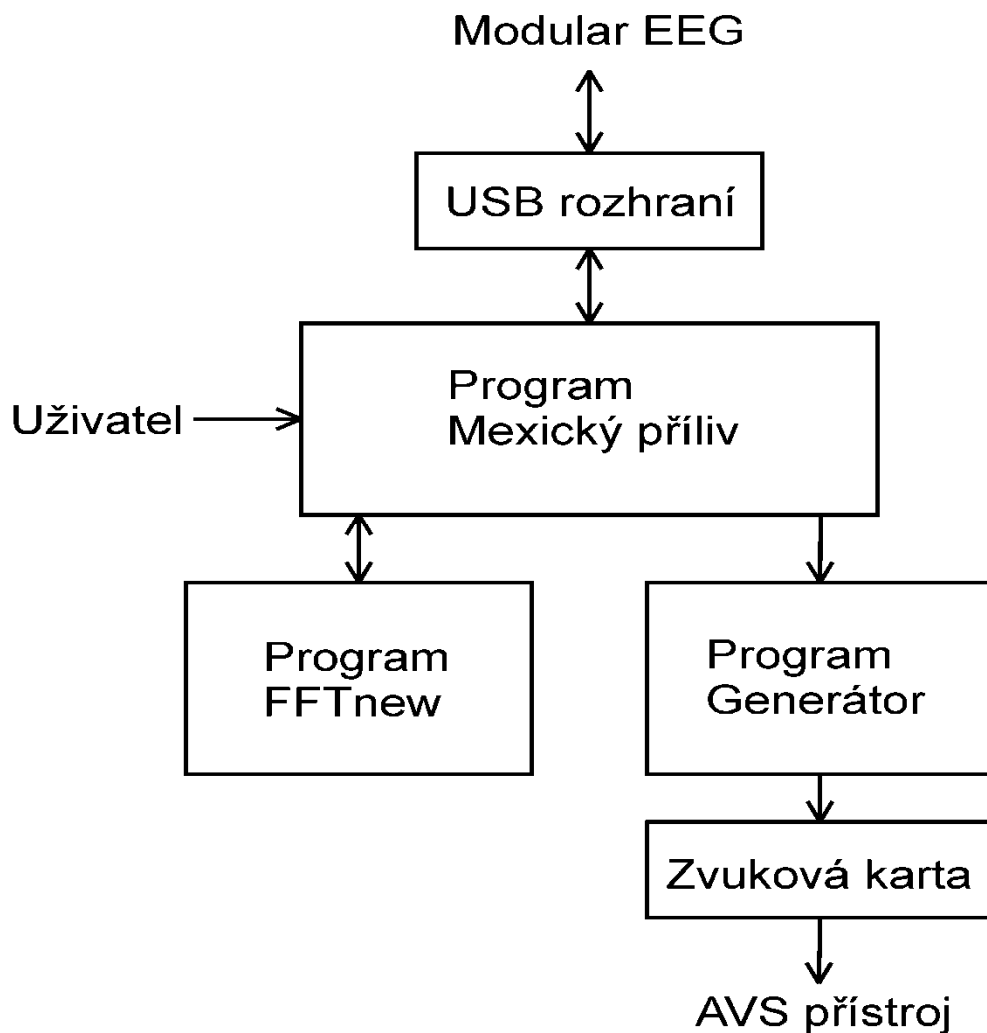
Hlavní program umožňuje nastavit uživateli všechny potřebné údaje pro chod celého projektu. Načítá data přes USB rozhraní z modulu AD převodníku, posílá je aplikaci počítající FFT, zobrazuje graficky vypočítané hodnoty a vypočítává vhodnou frekvenci pro AVS stimulaci a tuto informaci posílá programu generující zvukový signál přes zvukovou kartu počítače.

První z podpůrných programů se jmenuje FFTnew (verze 014) a jak název napovídá slouží k výpočtu Fourierovy transformace. Výpočet FFT je časově náročnější a proto pro její výpočet byl zvolen samostatný program.

Druhý podpůrný program nazvaný Generátor (verze 003) slouží k plynulému generování a přehrávání dvojitých binaurálních rytmů se složkou zvuku pro technologii AudioStrobe. Samostatnost tohoto programu byla zvolena proto, aby bylo

zajištěno plynulé generování zvuku bez nepříjemných přerušování při navazování jednotlivých přehrávaných úseků.

Na následujícím obrázku je blokově znázorněna provázanost jednotlivých aplikací včetně technických návazností.



Obr.26. Blokové schéma propojení všech programů včetně technických návazností

7.5.2. Program Mexický příliv

7.5.2.1. Popis programu Mexický příliv

Ke spuštění programu není potřeba žádná instalace. Stačí celý adresář zkopírovat na pevný disk.

Po spuštění programu Mexický příliv se načtou ze souboru nastavení všech polí formuláře tak, jak byla nastavena při minulém ukončení programu. Pokud spouštíte program poprvé jsou nastaveny ve všech polích výchozí hodnoty. Objeví se okno programu a pokud je nalezeno zařízení UMS3 spustí se v minimalizovaném okně program FFTnew. V opačném případě se objeví chybové hlášení a program je ukončen. Začnou se načítat data z AD převodníku, která jsou předzpracována a uložena na disk pro program FFTnew, který s nimi provede rychlou Fourierovu transformaci a uloží je zpět na disk. Odtud si je načte znovu program Mexický příliv a zobrazí grafické průběhy dvou zvolených kanálů EEG signálů. Vykreslí se i frekvenční spektrum pro zobrazované EEG signály. Vypočítá se frekvence s maximální amplitudou a zobrazí se vedle grafů. Cyklus načítání a zobrazování se opakuje stále dokola, dokud není kliknuto na tlačítko "Konec programu". V této chvíli se nacházíme v základním zobrazovacím módu, který je určen k uživatelskému nastavení, jak budou zpracovávána a vyhodnocována naměřená data. Ve chvíli, kdy se uživatel rozhodne ukončit program, pošle program zprávu aplikaci FFTnew, aby ukončila činnost, uloží se nastavení polí formuláře a program se ukončí.

Kliknutím na tlačítko "Start práce s AVS" se dostaneme do módu, který je určen pro audiovizuální stimulaci se zpětnou EEG vazbou. Jsou zablokována všechna pole pro nastavení parametrů výpočtu dat a jejich zpracování. V minimalizovaném okně je spuštěn program Generátor, který generuje zvuk duálních binaurálních rytmů se stopou AudioStrobe přes zvukovou kartu. Dále se načítají data z AD převodníku, ukládají se ke zpracování, provádí se FFT a načítají se a vykreslují průběhy EEG signálů a jejich frekvenční spektra. Kromě frekvence s maximální amplitudou, se začínají počítat i dominantní frekvence z frekvenčních spekter zvolených pracovních kanálů. Z těchto dominantních frekvencí je vyhodnocena celková dominantní frekvence, která je použita pro výpočet nejlepší stimulační frekvence pro AVS. Vypočítaná stimulační frekvence je poslána jako zpráva aplikaci Generátor, která ji použije k výpočtu generovaného zvuku. V hlavním okně se v tomto módu zobrazují

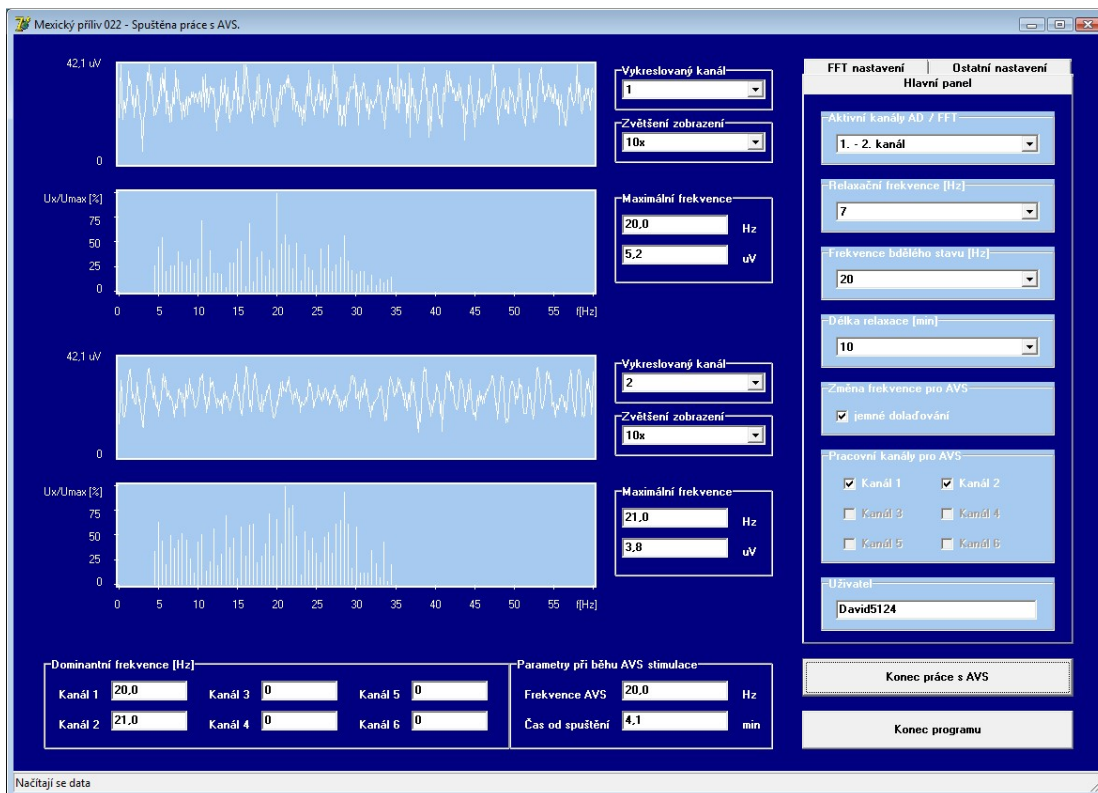
kromě vypočítaných dominantních frekvencí i stimulační frekvence na AVS přístroji a čas od spuštění stimulace. Do dokumentu pro hodnocení stimulace je ukládán čas měření, maximální a dominantní frekvence pro aktivní kanály, celková dominantní frekvence a stimulační frekvence. Po překročení doby stimulace se posílá zpráva programu Generátor, aby ukončil svou činnost. Ve sluchátkách se místo duálních binaurálních rytmů ozve každých patnáct sekund krátké pípnutí oznamující uživateli, že stimulace je ukončena. Toto pípnutí trvá do té doby, dokud uživatel neklikne na tlačítko "Konec práce s AVS". Program se poté vrátí do základního módu.

7.5.2.2. Mapování souborů

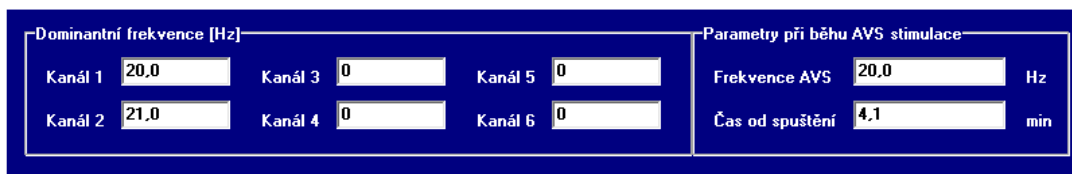
V textu uvádím, že jedna aplikace pošle druhé aplikaci zprávu s informacemi. Tento popis je trochu zjednodušený, ale na druhou stranu jednoduše vystihuje podstatu věci. Ve skutečnosti obě aplikace využívají mapování souborů pomocí programového rozhraní WinApi. Tímto způsobem lze sdílet určenou část paměti více aplikacemi a vyměňovat si tak data mezi aplikacemi, což je normálně ve Windows nemožné.

Jedna aplikace je serverem (v našem případě vždy program Mexický příliv) a druhá klientem (programy FFTnew, Generátor). Aplikace typu server vytvoří pomocí API společný paměťový prostor o potřebné velikosti (v našem případě 1 kB). V případě, že se aplikace server rozhodne poslat data aplikaci klient, uloží je do vytvořeného datového prostoru a pošle zprávu s jedinečným identifikačním názvem všem běžícím aplikacím. Aplikace klient dostane tuto zprávu a podle identifikátoru zjistí, že je určena pro ní. Tato zpráva je pro aplikaci klient povel, aby si ze společného paměťového prostoru načetla data. Těmito daty je pro aplikaci FFTnew řetězec "Konec" pro její ukončení. Aplikaci Generátor je tímto způsobem posílána stimulační frekvence nebo řetězec "Konec" pro její ukončení.

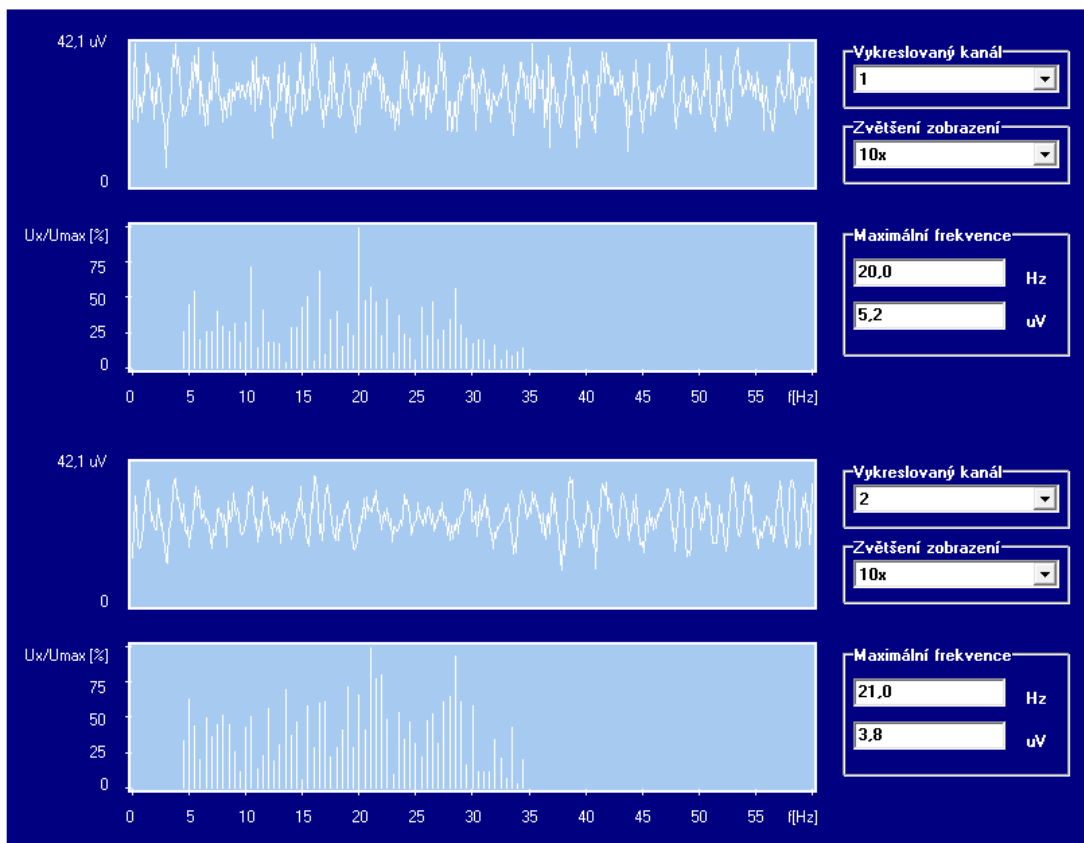
7.5.2.3. Popis formuláře programu Mexický příliv



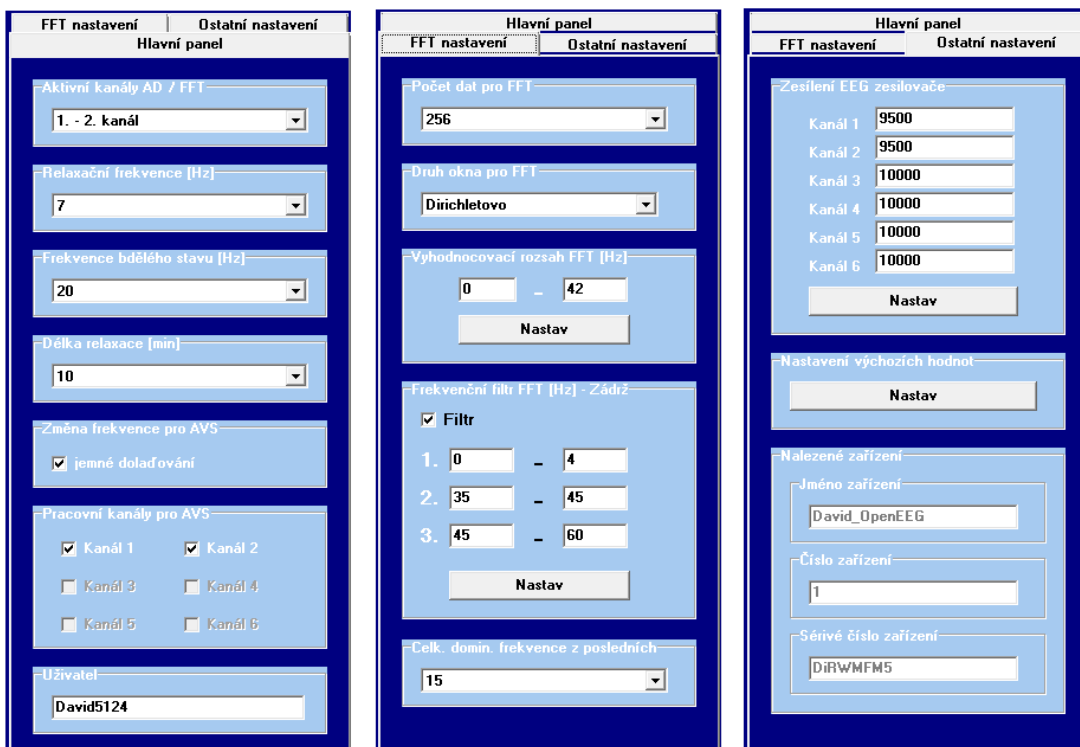
Obr.27. Okno aplikace Mexický příliv



Obr.28. Detail spodní části okna aplikace Mexický příliv



Obr.29. Detail grafické části okna aplikace Mexický příliv



Obr.30. Záložky v aplikaci Mexický příliv

Nyní bude následovat vysvětlení jednotlivých komponent formuláře programu Mexický příliv.

Graf - EEG signál

Nachází se na dvou místech v levé části formuláře.

Zobrazuje EEG signál, který již prošel Fourierovou transformací. Pokud jste si zvolili frekvenční filtrování v záložce FFT nastavení, uvidíte již signál zbavený rušivých složek.

Graf - Frekvenční spektrum EEG signálu

Nachází se na dvou místech v levé části formuláře vždy pod grafem zobrazujícím EEG signál, ze kterého je pomocí Fourierovy transformace toto frekvenční spektrum počítáno.

Znázorňuje velikost amplitud jednotlivých frekvencí EEG signálu v procentuálním vyjádření, kde hodnotu 100 % má frekvence s maximální amplitudou.

Výběrové pole - Vykreslovaný kanál

Nachází se na dvou místech formuláře vždy napravo od grafu vykreslovaného EEG signálu, se kterým souvisí.

Slouží k volbě kanálu nalevo se zobrazujícího EEG signálu a frekvenčního spektra, které se nachází pod ním. Je možné si zvolit z kanálu 1 - 6.

Výběrové pole - Zvětšení zobrazení

Nachází se na dvou místech formuláře vždy napravo od grafu vykreslovaného EEG signálu, se kterým souvisí.

Pomocí tohoto pole je možné si zvolit zvětšení nalevo se zobrazovaného EEG signálu od 1 do 1000 násobného zvětšení.

Informační pole - Maximální frekvence

Nachází se na dvou místech formuláře vždy napravo od grafu frekvenčního spektra EEG signálu, se kterým souvisí.

Zobrazuje frekvenci s maximální amplitudou a její amplitudu v mikrovoltech.

Informační pole - Dominantní frekvence

Nachází se v levé spodní části formuláře.

Zobrazuje dominantní frekvenci pro každý aktivní kanál EEG signálu. Dominantní frekvence je zobrazována až po kliknutí na tlačítko "Spuštění práce s AVS".

Informační pole - Frekvence AVS

Nachází se ve středu spodní části formuláře.

Zobrazuje aktuální stimulační frekvenci AVS přístroje. Zobrazí se až po kliknutí na tlačítko "Spuštění práce s AVS".

Informační pole - Čas od spuštění

Nachází se ve středu spodní části formuláře.

Zobrazuje čas od spuštění audiovizuální stimulace.

Tlačítko - Start práce s AVS

Nachází se vpravo ve spodní části formuláře.

Po kliknutí na toto tlačítko začíná audiovizuální stimulace s výjimkou, kdy jste změnili některé z editačních polí vyžadující po editaci kliknutí na tlačítko "Nastav" a vy jste tak neučinili. V tomto případě se vypíše chybové hlášení a vy musíte provést nastavení (barva konfliktního pole je podbarvena červeně), jinak nespustíte stimulační cyklus. Ve stimulačním cyklu se dál načítají data, zobrazují se grafy EEG signálů a jejich frekvenční spektra, zobrazují se maximální frekvence. Navíc se začíná zobrazovat i dominantní frekvence všech aktivních kanálů, zobrazuje se čas od spuštění stimulace a frekvence na AVS přístroji.

Po kliknutí na tlačítko "Start práce s AVS" se změní nápis na tlačítku na "Konec práce s AVS". V této chvíli je možné kliknutím na tlačítko předčasně ukončit stimulaci. Kliknutím na toto tlačítko také ukončujete po uplynutí relaxační doby cyklus stimulace a vracíte se zpět do zobrazovacího módu programu.

Tlačítko - Konec programu

Nachází se vpravo ve spodní části formuláře.

Po kliknutí na toto tlačítko se ukončí program Mexický příliv i program FFTnew. Program vám nepůjde ukončit ve chvíli, kdy se nacházíte ve stimulačním módu programu. Tehdy musíte nejprve ukončit tento mód kliknutím na tlačítko "Konec práce s AVS".

Záložky Hlavní panel, FFT nastavení, Ostatní nastavení

Nacházejí se v pravé části formuláře.

Jedná se o tři panely s nastavovacími poli. Přepínání se provádí v horní části kliknutím na příslušný název.

Záložka Hlavní panel obsahuje:

- výběrové pole Aktivní kanály
- nastavení pro relaxaci: pole Relaxační frekvence, Frekvence bdělého stavu, Délka relaxace
- nastavení související s výpočtem stimulační frekvence AVS přístroje: pole Změna frekvence pro AVS a Pracovní Kanály pro AVS.
- editační pole Uživatel.

Záložka FFT nastavení tvoří pole pro nastavení parametrů, podle kterých pracuje Fourierova transformace, či přesněji řečeno souvisejících s výpočtem FFT. Na této záložce naleznete:

- výběrová pole Počet dat pro FFT, Druh okna pro FFT, Celková dominantní frekvence z posledních
- editační pole Vyhodnocovací rozsah FFT, Frekvenční filtr FFT

Záložka Ostatní nastavení obsahuje:

- skupinu editačních polí Zesílení EEG zesilovače
- Tlačítko - Nastavení výchozích hodnot
- informační panel - Nalezené zařízení

Výběrové pole - Aktivní kanály AD/FFT

Nachází se na záložce Hlavní panel v horní části.

Slouží k nastavení počtu zpracovávaných EEG signálů, jak odesílaných FFT, tak FFT zpracovávaných. Je možné si zvolit 1 - 6 kanálů, s možností vždy jen

jednoho nebo dvojic, či prvních čtyř nebo všech. Bez konfliktů si zvolíte více či méně než máte skutečných EEG aktivních kanálů, ale doporučuji zvolit přesně tolik kolik máte skutečně. Ve chvíli, kdy jste zvolili, že máte aktivní jen jeden kanál, již nedokážete ve volbě Vykrusovaný kanál zvolit jiný než první i kdyby jste druhý kanál měli skutečně zapojený, to samé platí pro nastavení polí v Pracovních kanálech pro AVS. Při volbě více kanálů než používáte zase FFT zpracovává nepotřené data a tím se prodlužuje celý proces analýzy signálů.

Výběrové pole - Relaxační frekvence

Nachází se na záložce Hlavní panel v horní části.

Používá se pro nastavení cílové relaxační frekvence, to jest frekvence, na kterou se vás snaží AVS přístroj postupně naladit.

Výběrové pole Frekvence bdělého stavu

Nachází se na záložce Hlavní panel v horní části.

Slouží k nastavení frekvence bdělého stavu, na kterou vás přístroj postupně naladí z aktuální relaxační frekvence po uplynutí relaxační doby.

Výběrové pole Délka relaxace

Nachází se na záložce Hlavní panel ve střední části.

Slouží k nastavení délky relaxace (stimulace).

Potvrzovací pole Změna frekvence pro AVS

Nachází se na záložce Hlavní panel ve střední části.

Pomocí zaškrtnutí políčka Jemné doladování si můžete zvolit, jestli chcete menší rozdíl mezi celkovou dominantní frekvencí a stimulační frekvencí. Při jemném doladování je rozdíl celkové dominantní a stimulační frekvence v rozmezí 0,5-1,4 Hz. V případě, že celková dominantní frekvence je blízká cílové frekvenci ($\pm 0,5$ Hz), potom stimulační frekvence je rovna cílové frekvenci. Pokud políčko Jemné doladování zůstane nezaškrtnuté, potom se rozdíl mezi celkovou dominantní frekvencí a stimulační frekvencí pohybuje v rozmezí 1-1,9 Hz. V případě, že celková dominantní frekvence je blízká cílové frekvenci (± 1 Hz), potom stimulační frekvence je rovna cílové frekvenci.

Skupina potvrzovacích polí - Pracovní Kanály pro AVS

Nachází se na záložce Hlavní panel ve spodní části.

Je možné si zde zvolit data, kterých kanálů chcete použít pro výpočet stimulační frekvence. Je možné si zvolit jen kanály v rozmezí, které jste si zvolili v poli Aktivní kanály.

Editační pole - Uživatel

Nachází se na záložce Hlavní panel ve spodní části.

V současné době není toto pole využíváno, jeho význam byl při pokusu zjistit nejlepší způsob pro výpočet dominantních frekvencí, kdy se používalo pro identifikaci uložených a znovu načítaných dlouhodobých průměrných či maximálních hodnot jednotlivých harmonických.

Výběrové pole - Počet dat pro FFT

Nachází se na záložce FFT nastavení v horní části.

Umožňuje nastavit počet dat zpracovávaných Fourierovou transformací pro jeden kanál a také určuje počet dat, pro jeden kanál, načtených z AD převodníku v jednom cyklu. Je možno si zvolit z hodnot 128, 256, 512 a 1024.

Výběrové pole - Druh okna pro FFT

Nachází se na záložce FFT nastavení v horní části.

Nabízí výběr jedno ze čtyř váhovacích oken pro předzpracování signálu před FFT. Nabízenými okny jsou Dirichletovo (obdélníkové), Hanningovo, Hammingovo a Blackmanovo okno.

Skupina editačních polí - Vyhodnocovací rozsah FFT

Nachází se na záložce FFT nastavení v horní části.

Jsou zde dvě editační pole pro nastavení frekvenčního rozsahu, ve kterém vrací FFT data frekvenčního spektra. Toto rozmezí také limituje frekvence, mezi kterými bude vyhledána frekvence s maximální amplitudou daného kanálu a jen z frekvencí v tomto rozmezí bude počítána dominantní frekvence. Pod editačními poli pro volbu frekvenčního rozmezí je tlačítko "Nastav", na které je po zvolení hodnot nutno kliknout, jinak změna nebude provedena. Na nutnost kliknutí vás upozorní

změna barvy celého pole na tmavě červenou. Po kliknutí na tlačítko je kontrolováno, zda hodnota v levém poli je větší než v pravém. Jestliže není splněna tato podmínka, vypíše se chybové hlášení a barva pole zůstane červená. V případě, že kontrola proběhla v pořádku nastaví se hodnoty a pole změní barvu na původní.

Skupina editačních polí - Frekvenční filtr FFT

Nachází se na záložce FFT nastavení ve střední části.

Úplně nahoře v tomto poli se nachází potvrzovací pole Filtr, které značí, jestli vůbec chcete frekvenční filtraci prováděnou na frekvenčním spektru po výpočtu FFT použít. Pokud je políčko zaškrtnuto používají se tři frekvenční rozmezí, jejichž hodnoty jsou uvedeny v níže zobrazených polích. Je nutno zde zdůraznit, že se jedná o frekvenční rozsahy frekvenčních zádrží. Pod editačními poli pro volbu frekvencí je tlačítko "Nastav", na které je po zvolení hodnot nutno kliknout, jinak změna nebude provedena. Na nutnost kliknutí vás upozorní změna barvy celého pole na tmavě červenou. Po kliknutí na tlačítko je v každém rozsahu kontrolováno, zda hodnota v levém poli je větší než v pravém. Jestliže není splněna tato podmínka, vypíše se chybové hlášení a barva pole zůstane červená. V případě, že kontrola proběhla v pořádku nastaví se hodnoty a pole změní barvu na původní. Pokud by jste chtěli zvolit jen jedno frekvenční rozmezí, je nutno toto rozmezí napsat ve stejné podobě do všech tří frekvenčních rozsahů.

Výběrové pole - Celková dominantní frekvence z posledních

Nachází se na záložce FFT nastavení ve spodní části.

Zde je možné si zvolit z kolika posledních celkových dominantních frekvencí bude vyhodnocována aktuální stimulační frekvence.

Skupina editačních polí - Zesílení EEG zesilovače

Nachází se na záložce Ostatní nastavení v horní části.

Je zde 6 editačních polí pro nastavení skutečného napěťového zesílení kanálů EEG zesilovače. Hodnota zesílení je použita pro výpočet velikosti amplitudy maximální frekvence. Pod editačními poli pro volbu zesílení je tlačítko "Nastav", na které je po zvolení hodnot nutno kliknout, jinak změna nebude provedena. Na nutnost kliknutí vás upozorní změna barvy celého pole na tmavě červenou. Po kliknutí na

tlačítko je kontrolováno, zda hodnota některého pole není rovna nule. Jestliže není splněna tato podmínka, vypíše se chybové hlášení a barva pole zůstane červená. V případě, že kontrola proběhla v pořádku nastaví se hodnoty a pole změní barvu na původní.

Tlačítko - Nastavení výchozích hodnot

Nachází se na záložce Ostatní nastavení ve střední části.

Slouží k nastavení všech polí formuláře na výchozí hodnoty, tak jak byli nastaveny před prvním použitím.

Skupina informačních polí - Nalezené zařízení

Nachází se na záložce Ostatní nastavení ve spodní části.

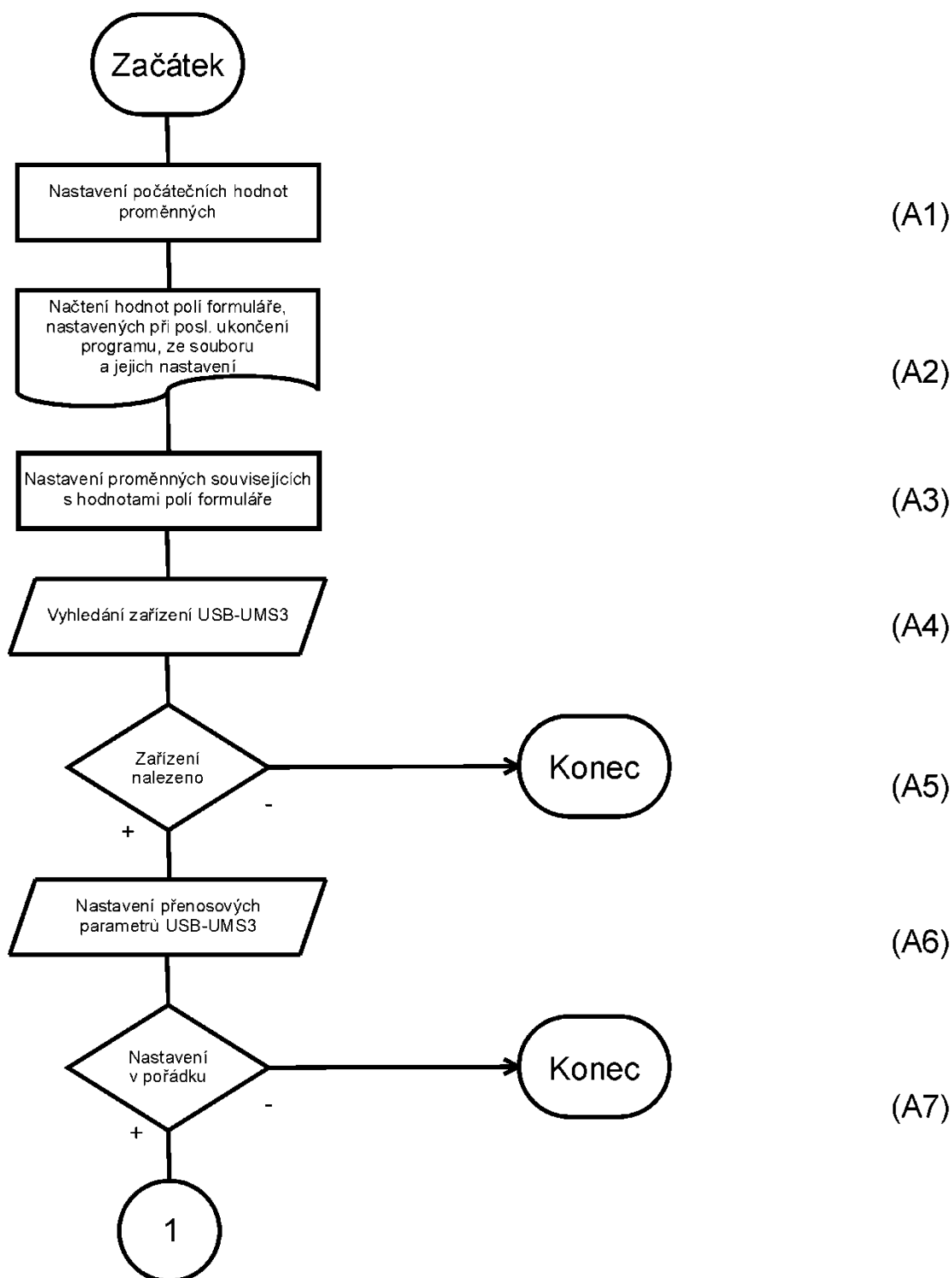
Na tomto panelu se nachází tři informační pole o nalezeném zařízení UMS3. Těmito informacemi je jméno zařízení, číslo zařízení a sériové číslo zařízení. Tyto pole jsou pouze informační.

Stavový řádek formuláře

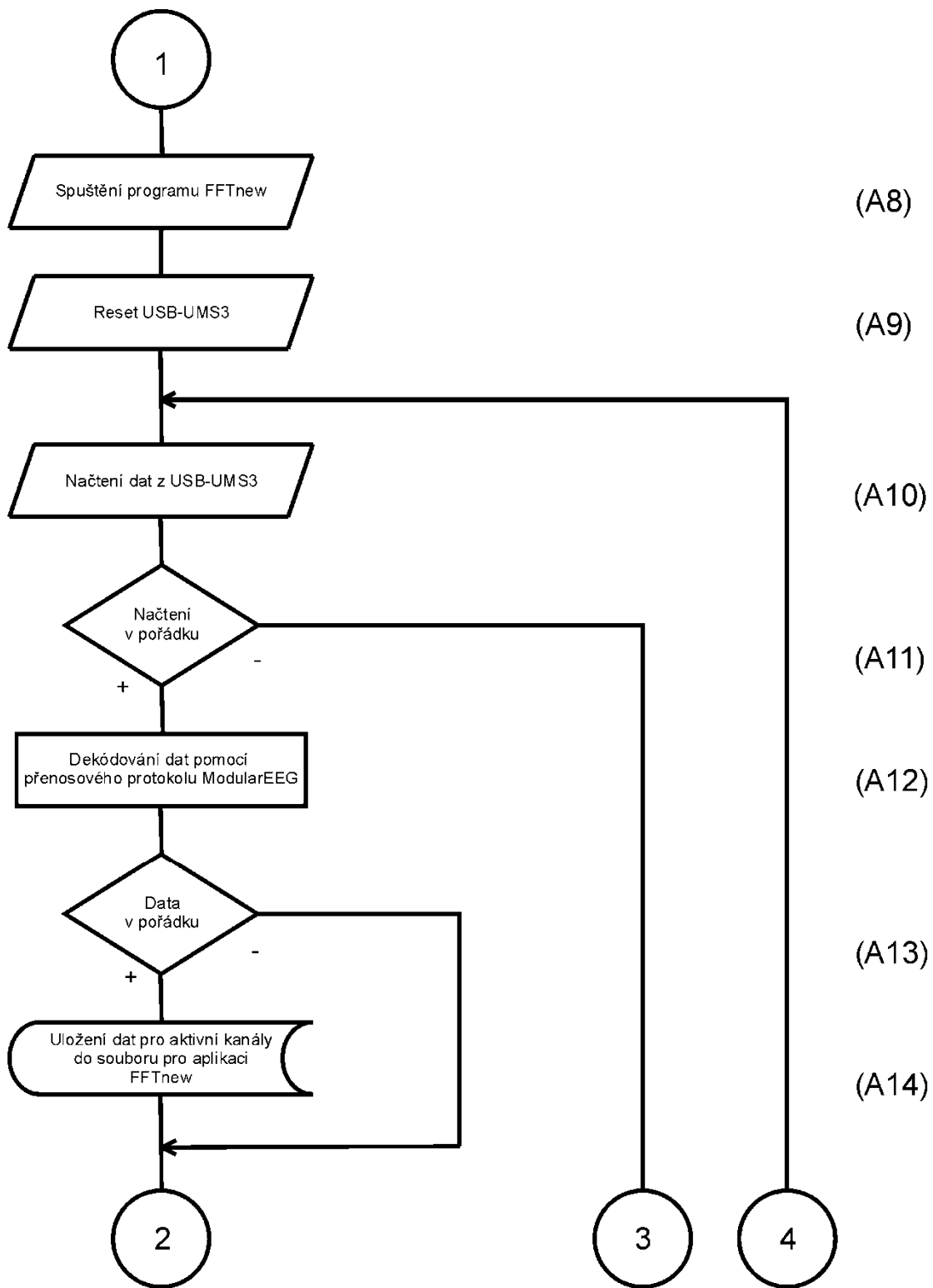
Nachází se ve spodní části formuláře.

Informuje uživatele, co v této chvíli program provádí za operaci.

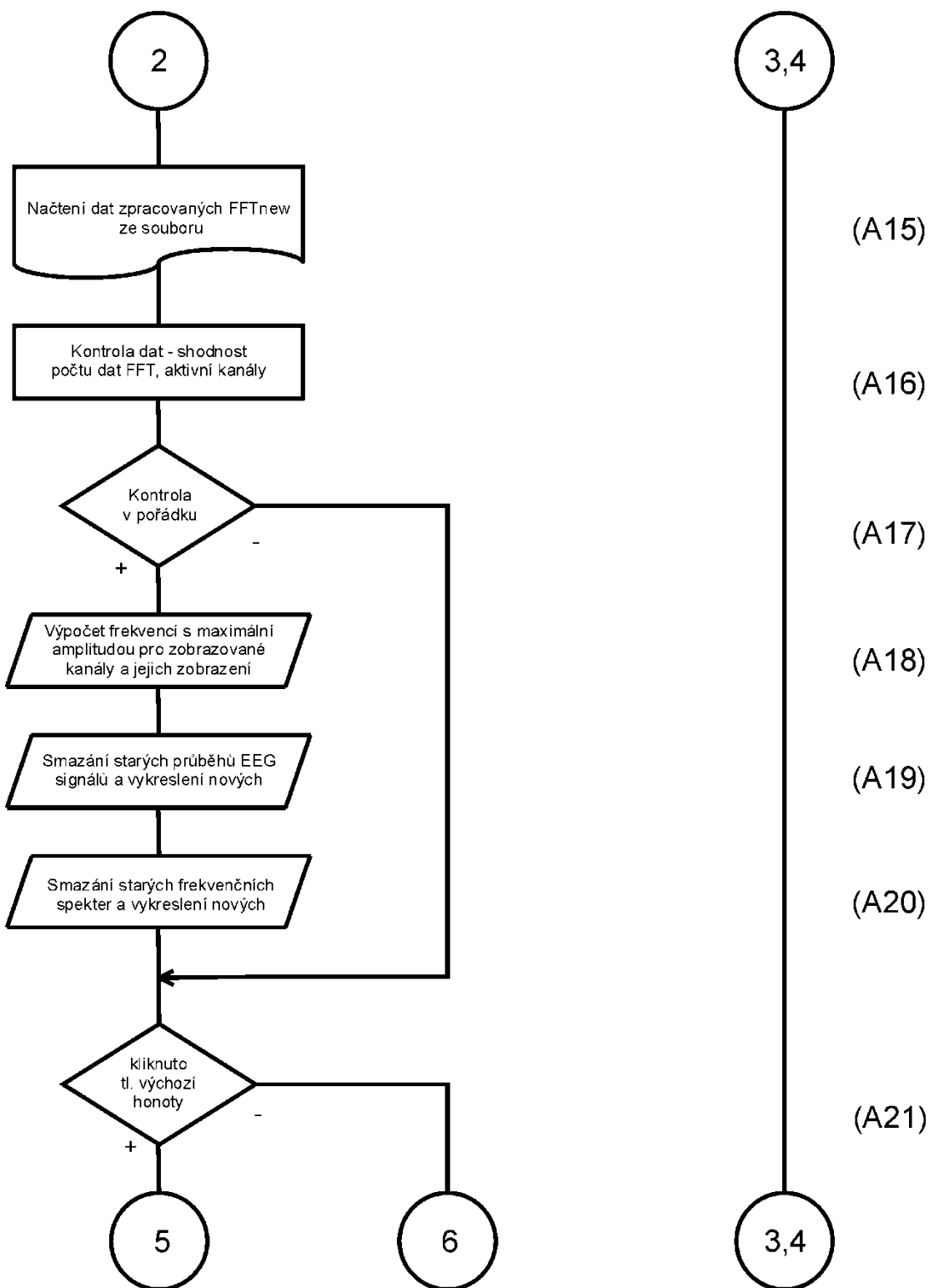
7.5.2.4. Vývojový diagram pro program Mexický příliv



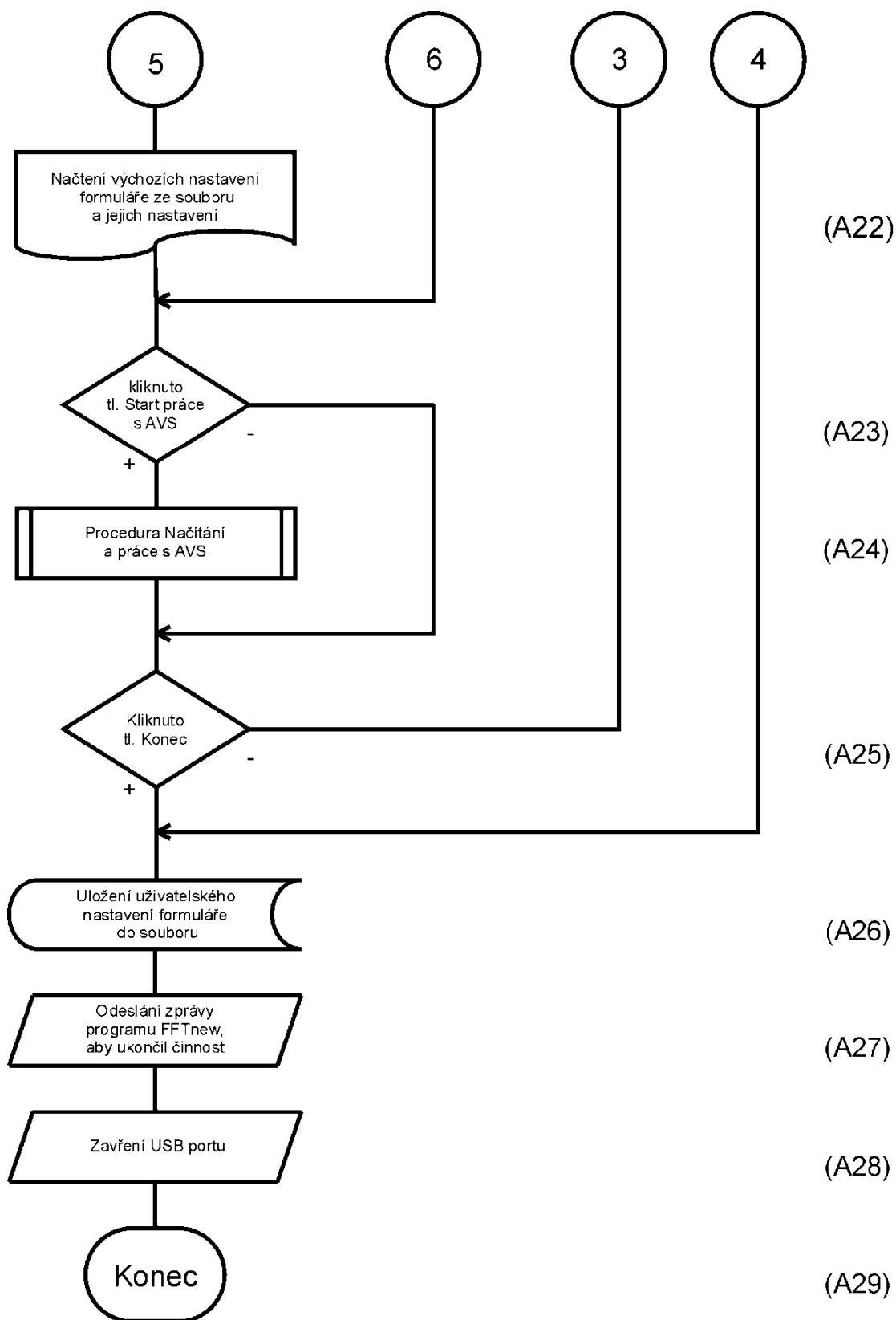
Obr.31. Vývojový diagram programu Mexický příliv strana 1



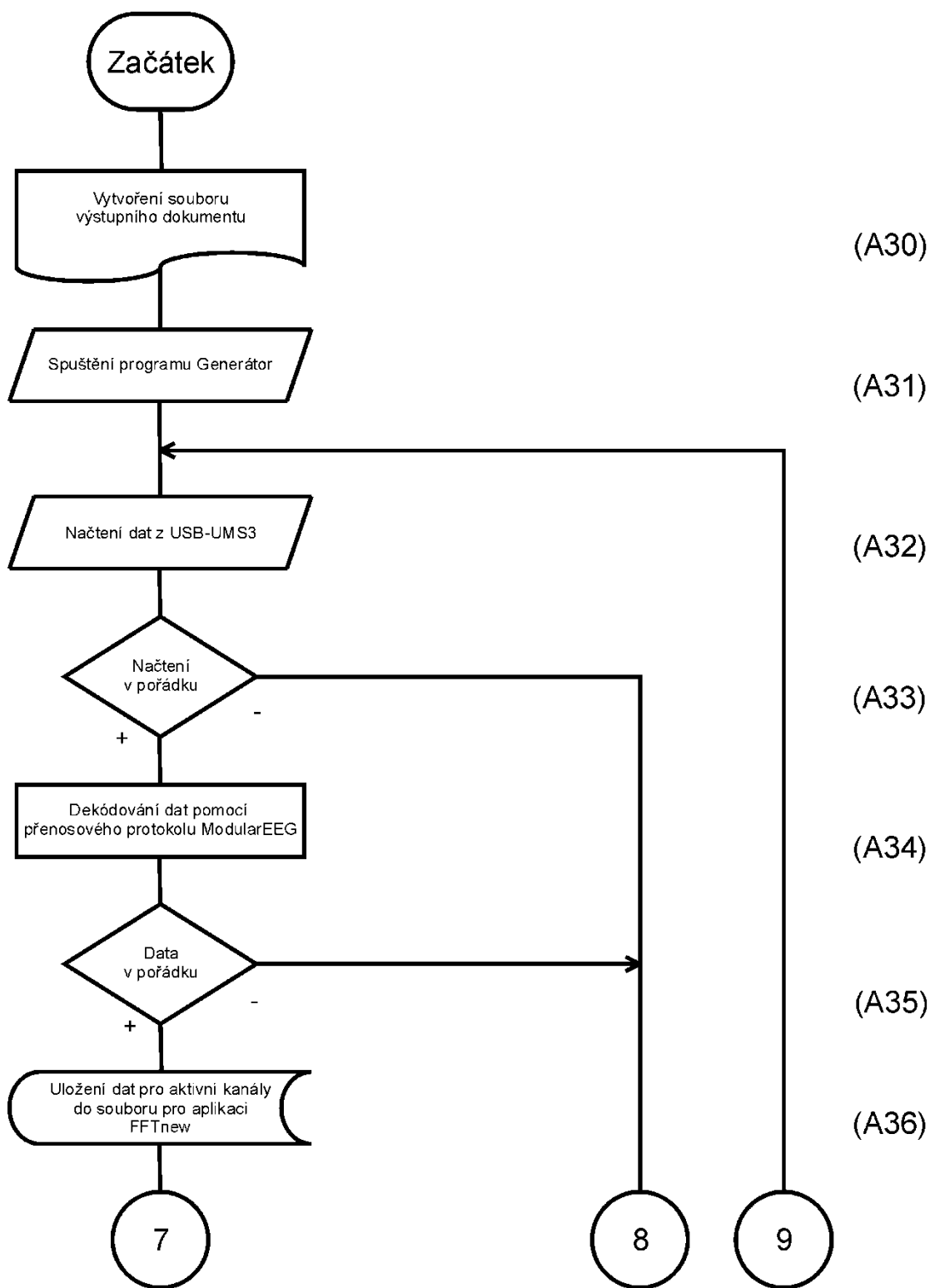
Obr.32. Vývojový diagram programu Mexický příliv strana 2



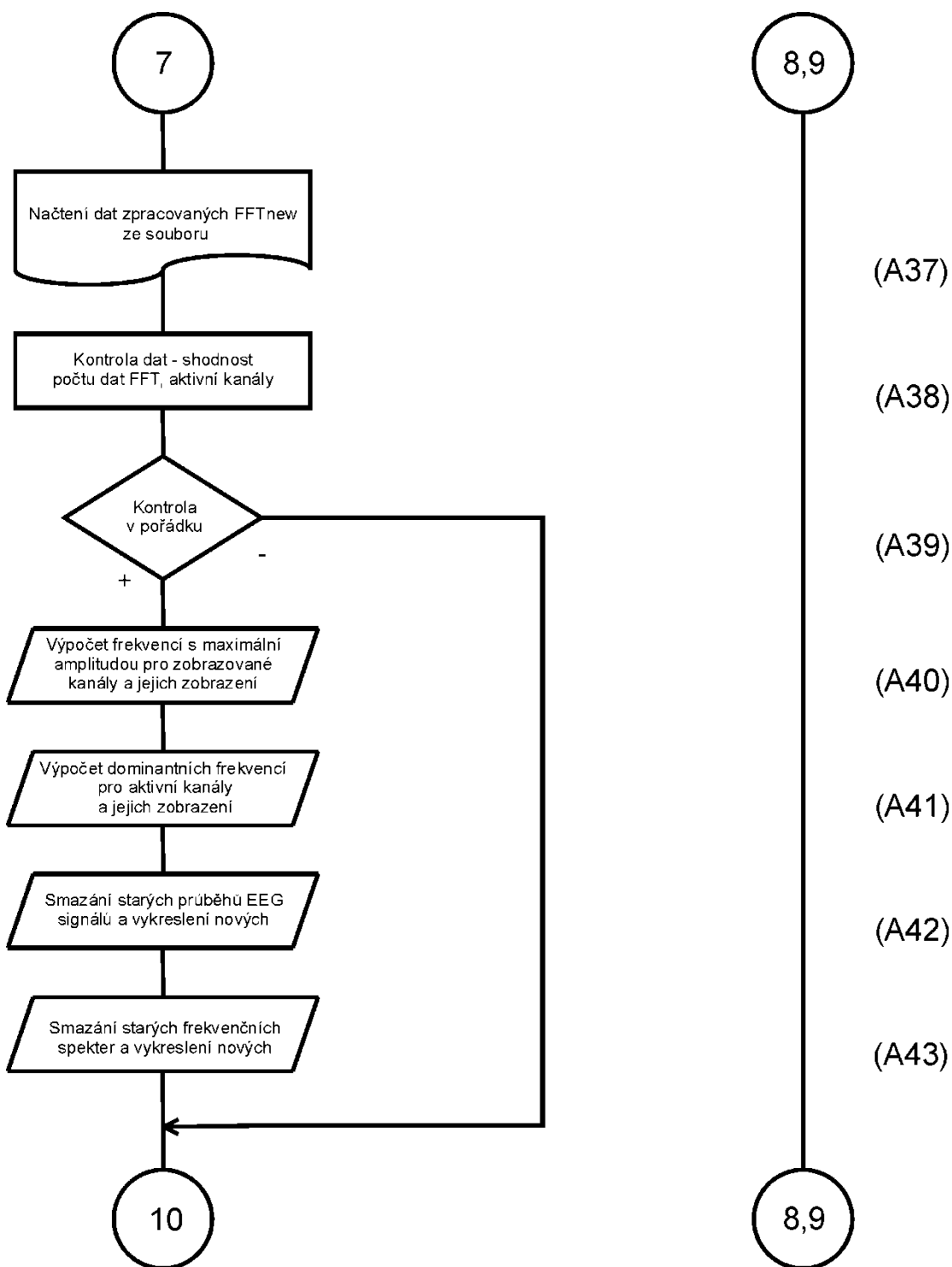
Obr.33. Vývojový diagram programu Mexický příliv strana 3



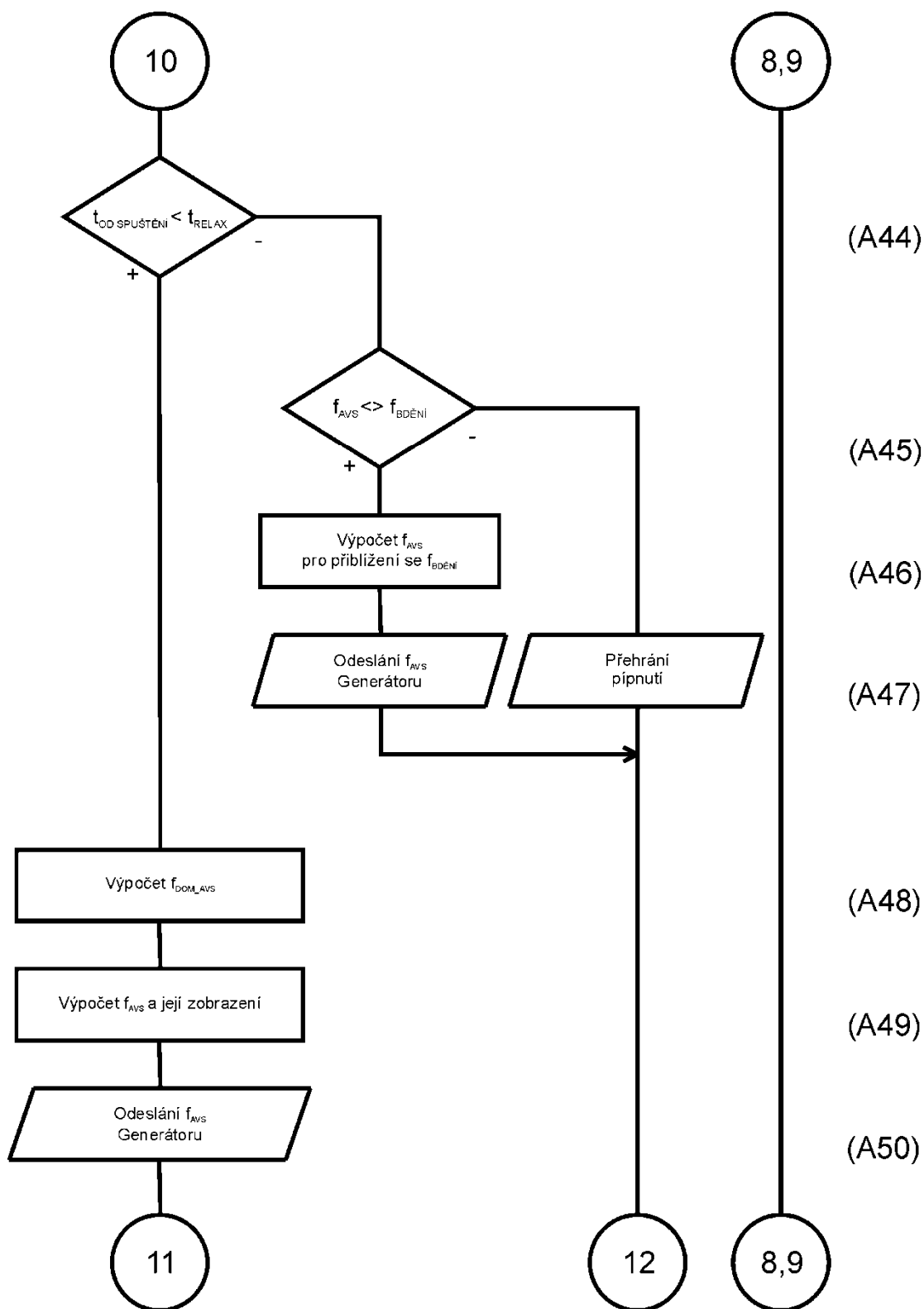
Obr.34. Vývojový diagram programu Mexický příliv strana 4



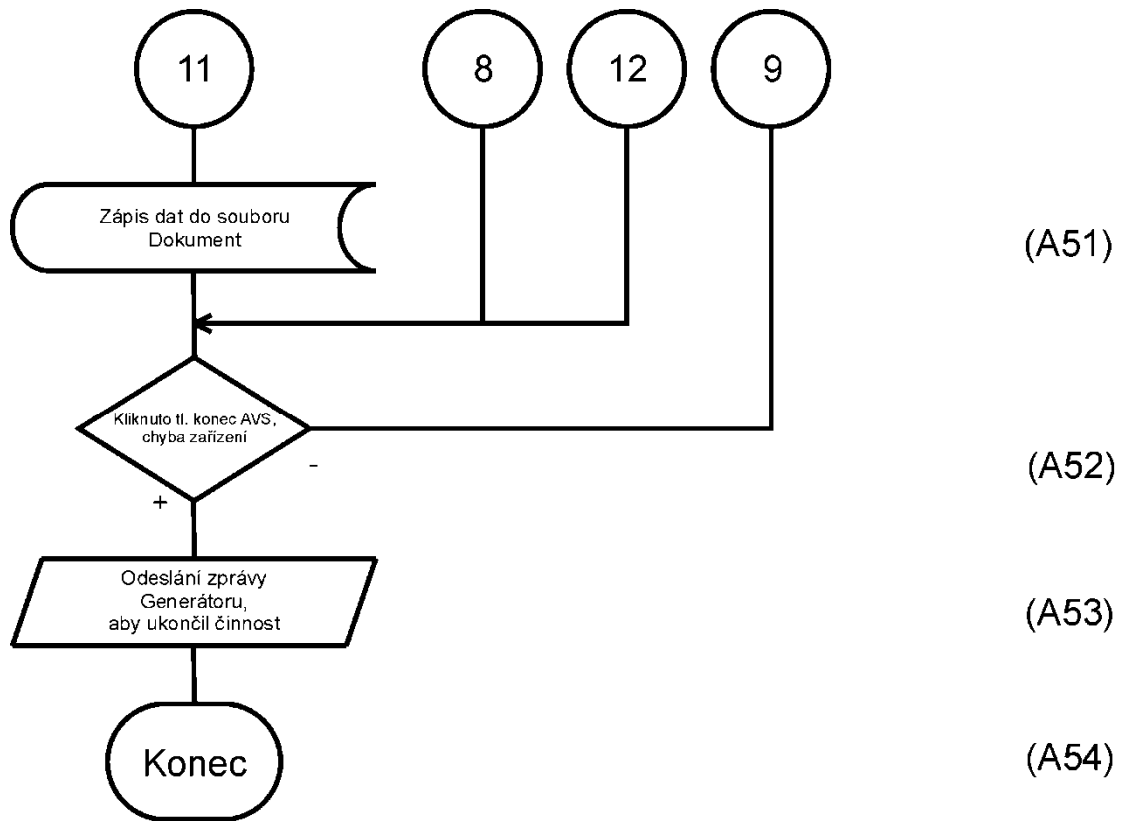
Obr.35. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 1



Obr.36. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 2



Obr.37. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 3



Obr.38. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 4

7.5.2.5. Popis vývojového diagramu programu Mexický příliv

A1

- nastavení konstant a proměnných
- nastavení příznaků chyb a stavů budoucích dějů

A2

- načtení ze souboru hodnot všech polí formuláře, jak byly nastaveny při minulém ukončení programu a jejich nastavení

A3

- nastavení všech číselných proměnných, se kterými program počítá, převodem z polí formuláře do číselných proměnných
- převedení slovních nastavení polí formuláře na proměnné typu řetězec, platí pro ty, které se budou posílat programu FFTnew (př. typ váhovacího okna)
- nastavení blokace výběru pracovních kanálů pro AVS, které nejsou v rozmezí aktivních kanálů (myšleno AD převodníku)

A4, A5

- vyhledání, jestli se mezi zařízeními připojenými na USB nachází zařízení UMS3, hledání se provádí podle názvu zařízení, který jsem dříve nastavil utilitou Mprog
- pokud je zařízení UMS3 nalezeno, nastaví se do proměnných číslo zařízení a sériové číslo, čímž je zařízení jednoznačně identifikovatelné a volatelné
- pokud zařízení není nalezeno, zobrazí se chybové hlášení a program se ukončí

A6,A7

- nastaví se přenosové parametry na zařízení UMS3 pro sériový přenos, viz popis zařízení UMS3
- pokud při nastavení dojde k chybě zařízení, vypíše se chybové hlášení a program se ukončí

A8

- pomocí programového rozhraní WinApi je spuštěn program FFTnew v minimalizovaném zobrazení

A9

- počáteční resetování zařízení UMS3

A10

- na tomto místě začíná cyklus načítání a vyhodnocování dat, který je možné ukončit kliknutím na tlačítko "Konec"
- načítají se EEG data v počtu potřebném pro FFT z úložiště v UMS3, což velmi zrychluje činnost celého programu, oproti tomu, kdyby program načítal data postupně, jak je převádí AD převodník

A11

- pokud došlo k chybě zařízení UMS3 při načítání dat neprovádí se žádné další operace a program bude po uložení uživatelského nastavení ukončen

A12, A13, A14

- po načtení jsou data dekódována podle protokolu P2, viz kapitola Modular EEG
- pokud čísla jednotlivých paketů načtených dat nejdou za sebou, data se nezpracovávají ani neposílají FFT
- data se v novém praktičtějším formátu ukládají na disk, kde jsou připravena pro zpracování v programu FFTnew
- při ukládání dat se vynechávají data z neaktivních vstupů AD převodníku a jsou přidány informace pro výpočet FFT (počet dat FFT, první a poslední aktivní kanál, rozmezí uživatelsky důležitých frekvencí F_{min} a F_{max} , rozmezí frekvenčních filtrů (prováděných v FFT), název váhovacího okna)

A15

- načtení dat z programu FFTnew ze souboru, těmito daty jsou EEG signály a jejich frekvenční spektra pro aktivní kanály, počet dat pro výpočet FFT, první a poslední aktivní kanál

A16, A17

- kontrola zda hodnoty (počet dat FFT, první a poslední kanál) načtené ze souboru jsou shodné s hodnotami, které se v současné chvíli používají v hlavním programu, pokud nejsou shodné, nenačtou se žádná data do pracovních polí, nebudou graficky znázorněna ani se neprovede výpočet frekvencí s max. amplitudou

A18

- pro oba zobrazované kanály jsou nalezeny frekvence s maximální amplitudou z frekvenčního spektra a vypočítají se jejich amplitudy, tyto hodnoty se vypíší vedle grafů
- výpočet frekvence z čísla harmonické složky:
 $Koef = 1 / (N / f_{VZORKOVACÍ\ AD})$, kde N je počet dat pro výpočet FFT
 $f = n * Koef$, kde n je číslo harmonické
- výpočet amplitudy frekvence v μV
 $U_{\text{řs}} = ((4 * V * (U_{\text{RefNapetiAD}} / D)) / A_{u_{\text{kan}}}) * 1000000;$
kde
V je velikost harmonické složky
D je maximální hodnota AD převodníku+1 (v našem případě 1024 (0..1023))
 $A_{u_{\text{kan}}}$ - napěťové zesílení kanálu EEG zesilovače

A19, A20

- smazání starých grafů EEG signálů a jejich frekvenčních spekter
- vykreslení nových grafů, frekvenční spektra jsou zobrazena v procentuálním zobrazení, kde frekvence s maximální amplitudou je rovna 100 procent

A21, A22

- zjištění, jestli v průběhu tohoto programového cyklu uživatel neklikl na tlačítko "Nastavení výchozích hodnot", pokud ano, načtou se ze souboru všechna výchozí nastavení jednotlivých polí formuláře a nastaví se na tyto hodnoty
- nastaví se také hodnoty proměnných vycházejících z hodnot těchto polí

A23, A24

- zjištění, jestli v průběhu tohoto programového cyklu uživatel neklikl na tlačítko "Start práce s AVS", pokud ano je spuštěna procedura "Načítání a práce s AVS", bude popsána ve zvláštním algoritmu

A25

- zjištění, jestli v průběhu tohoto programového cyklu uživatel neklikl na tlačítko "Konec", jestliže ne program provede další cyklus od bodu A10

A26, A27, A28, A29

- jestliže si uživatel přeje ukončit program uloží se nastavení formuláře do souboru

- aplikace pošle programu FFTnew pomocí funkce mapování souborů zprávu, aby ukončila činnost
- provede se zavření USB portu
- aplikace se ukončí

7.5.2.6. Popis vývojového diagramu programu Mexický příliv - Procedura Načítání a práce s AVS

A30

- vytvoření výstupního souboru Dokument, do kterého se budou ukládat výsledky průběhu relaxace s AVS, název souboru bude Dokument + datum + hodina a minuta spuštění stimulace, soubor je ve formátu csv pro snadnou konverzi do programu Excel

A31

- pomocí programového rozhraní WinApi je spuštěn program Generátor

A32

- na tomto místě začíná cyklus načítání a vyhodnocování dat, výpočet frekvence pro AVS, cyklus je možné ukončit kliknutím na tlačítko "Konec práce s AVS" nebo se ukončí sám při chybě zařízení USM3
- načítají se EEG data v počtu potřebném pro FFT z úložiště v UMS3

A33

- pokud došlo k chybě zařízení UMS3 při načítání dat, neprovádí se žádné další operace a po odeslání zprávy programu Generátor, aby ukončil činnost, se procedura ukončí

A34, A35, A36

- po načtení jsou data dekodována podle protokolu P2, viz. kapitola Modular EEG
- pokud čísla jednotlivých paketů načtených dat nejdou za sebou, data se nezpracovávají ani neposílají FFT
- data se v novém praktičtějším formátu ukládají na disk, kde jsou připravena pro zpracování v programu FFTnew
- při ukládání dat se vynechávají data z neaktivních vstupů AD převodníku a jsou přidány informace pro výpočet FFT (počet dat FFT, první a poslední

aktivní kanál, rozmezí uživatelsky důležitých frekvencí F_{min} a F_{max} ,
rozmezí frekvenčních filtrů (prováděných v FFT , název váhovacího okna)

A37

- načtení dat zpracovaných programem FFTnew ze souboru, těmito daty jsou EEG signály a jejich frekvenční spektra pro aktivní kanály, počet dat pro výpočet FFT, první a poslední aktivní kanál

A38, A39

- kontrola zda hodnoty (počet dat FFT, první a poslední kanál) načtené ze souboru jsou shodné s hodnotami, které se v současné chvíli používají v hlavním programu, pokud nejsou shodné, nenačtou se žádná data do pracovních polí, nebudou graficky znázorněna ani se neprovede výpočet frekvencí s max. amplitudou a dominantních frekvencí

A40

- pro oba zobrazované kanály jsou nalezeny frekvence s maximální amplitudou z frekvenčního spektra a vypočítají se jejich amplitudy (výpočet stejný jako v hlavní části programu), tyto hodnoty se vypíší vedle grafů

A41

- výpočet dominantních frekvencí z frekvenčního spektra pro aktivní kanály, kdy je v každém kanálu dominantní frekvence rovna frekvenci s maximální amplitudou
- dominantní frekvence jsou zobrazeny a uloženy do pole, ve kterém je archivováno posledních 1-20 (dle nastavení uživatele) dominantních frekvencí pro každý kanál

A42, A43

- smazání starých grafů EEG signálů a jejich frekvenčních spekter
- vykreslení nových grafů, frekvenční spektra jsou zobrazena v procentuálním zobrazení, kde frekvence s maximální amplitudou je rovna 100 procent

A44

- zjištění, jestli je dosažena doba relaxačního sezení, pokud není je proveden až bod A48

A45, A46, A47

- jestliže je frekvence AVS na přístroji jiná než frekvence bdělé hladiny zvýší- sníží se frekvence pro AVS o 1 Hz blíže frekvenci bdělé hladiny, tato frekvence se pošle aplikaci Generátor pomocí funkce mapování souborů
- jestliže frekvence AVS na přístroji je rovna frekvenci bdělé hladiny, pak se přehraje každých 15 sekund pípnutí signalizující uživateli, že může ukončit relaxaci, v této chvíli se v prvním průběhu před pípnutím ukončuje aplikace Generátor
- dále se neprovádějí žádné další body, opakuje se jen tento cyklus od bodu A32 a čeká se až uživatel klikne na tlačítko "Konec práce s AVS"

A48

- v poli, ve kterém jsou archivovány všechny dominantní frekvence z minulých cyklů (1-20) viz. bod A41, se vyhledá ze všech zvolených pracovních kanálů dominantní frekvence s nejvyšší frekvencí, nalezená frekvence je frekvence f_{DOM_AVS}

A49

- vypočítá se hodnota f_{avs} , parametry pro její určení jsou ovlivněny stimulačními frekvencemi AVS přístroje, kde frekvenční rozmezí stimulace je 1-30 Hz s krokem 1 Hz
- jestliže $f_{DOM_AVS} < 1$ potom $f_{avs} = 1$
- jestliže $f_{DOM_AVS} > 22$ potom $f_{avs} = 30$
- jestliže $f_{DOM_AVS} > f_{CÍLOVÁ}$, potom $f_{avs} = I_{konv} - 1$, kde I_{konv} je nejbližší stimulační frekvence AVS přístroje menší než f_{DOM_AVS} , v případě, že je uživatelem preferováno jemnější rozhodování je f_{avs} rovna I_{konv} , jestliže je splněna podmínka, že $f_{DOM_AVS} - I_{konv} > 0,5$.

V případě, že se po tomto výpočtu změní výraz $f_{avs} > f_{CÍLOVÁ}$ na nepravdu, pak $f_{avs} = f_{CÍLOVÁ}$

- jestliže $f_{DOM_AVS} < f_{CÍLOVÁ}$, potom $f_{avs} = I_{konv} + 1$, kde I_{konv} je nejbližší stimulační frekvence AVS přístroje větší než f_{DOM_AVS} , v případě, že je uživatelem preferováno jemnější rozhodování je f_{avs} rovna I_{konv} , jestliže je splněna podmínka, že hodnota $I_{konv} - f_{DOM_AVS} > 0,5$.

V případě, že se po tomto výpočtu změní výraz $f_{avs} < f_{CÍLOVÁ}$ na nepravdu, poté $f_{avs} = f_{CÍLOVÁ}$

- jestliže $f_{\text{DOM_AVS}} = f_{\text{CÍLOVÁ}}$, potom $f_{\text{avs}} = f_{\text{CÍLOVÁ}}$

A50

- pomocí funkce mapování souborů je poslána aplikaci Generátor hodnota f_{avs}

A51

- zápis dat do souboru Dokument, těmito daty je čas měření, frekvence f_{DOM} pro všechny aktivní kanály, frekvence $f_{\text{DOM_AVS}}$ a frekvence f_{avs}

A52

- opakuje se cyklus od bodu A32, pokud uživatel neklikl na tlačítko "Konec práce s AVS" nebo nenastala chyba zařízení UMS3

A53

- pomocí pomocí funkce mapování souborů WinApi je poslána zpráva aplikaci Generátor, aby ukončila činnost, platí v případě, že nebyla již ukončena při překročení relaxační doby

A54

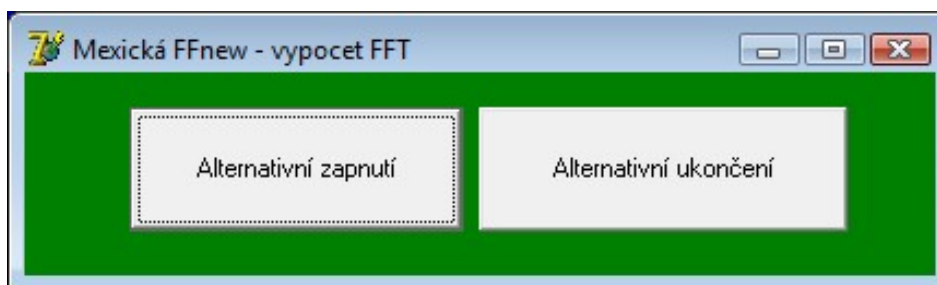
- ukončení této procedury a návrat do hlavního programového cyklu (A25)

7.5.3. Program FFTnew

7.5.3.1. Popis programu FFTnew

Program FFTnew byl postaven na programové jednotce, která pro výpočet používá standardní algoritmus FFT s decimací v čase. Tuto jednotku jsem převzal ze zdroje [I7] a jen jsem ji aplikoval do svého programu. K programové knihovně jsem proto přistupoval jako ke programové knihovně, kdy zadáte vstupní data a dostanete výstupní data a vůbec nevíte co se děje mezitím. Proto ve vývojovém diagramu ani v jeho popisu nebude popisován obsah této jednotky. Princip FFT s decimací v čase je popsán v kapitole o Fourierově transformaci. Správnost výpočtu FFT jsem si samozřejmě ověřil použitím jiného programu počítajícího FFT. Při zadání stejných vstupních dat byly výstupní data shodná. Druhým způsobem kontroly bylo, že jsem zadal taková vstupní data, u kterých jsem znal výsledky. I v tomto případě program fungoval správně.

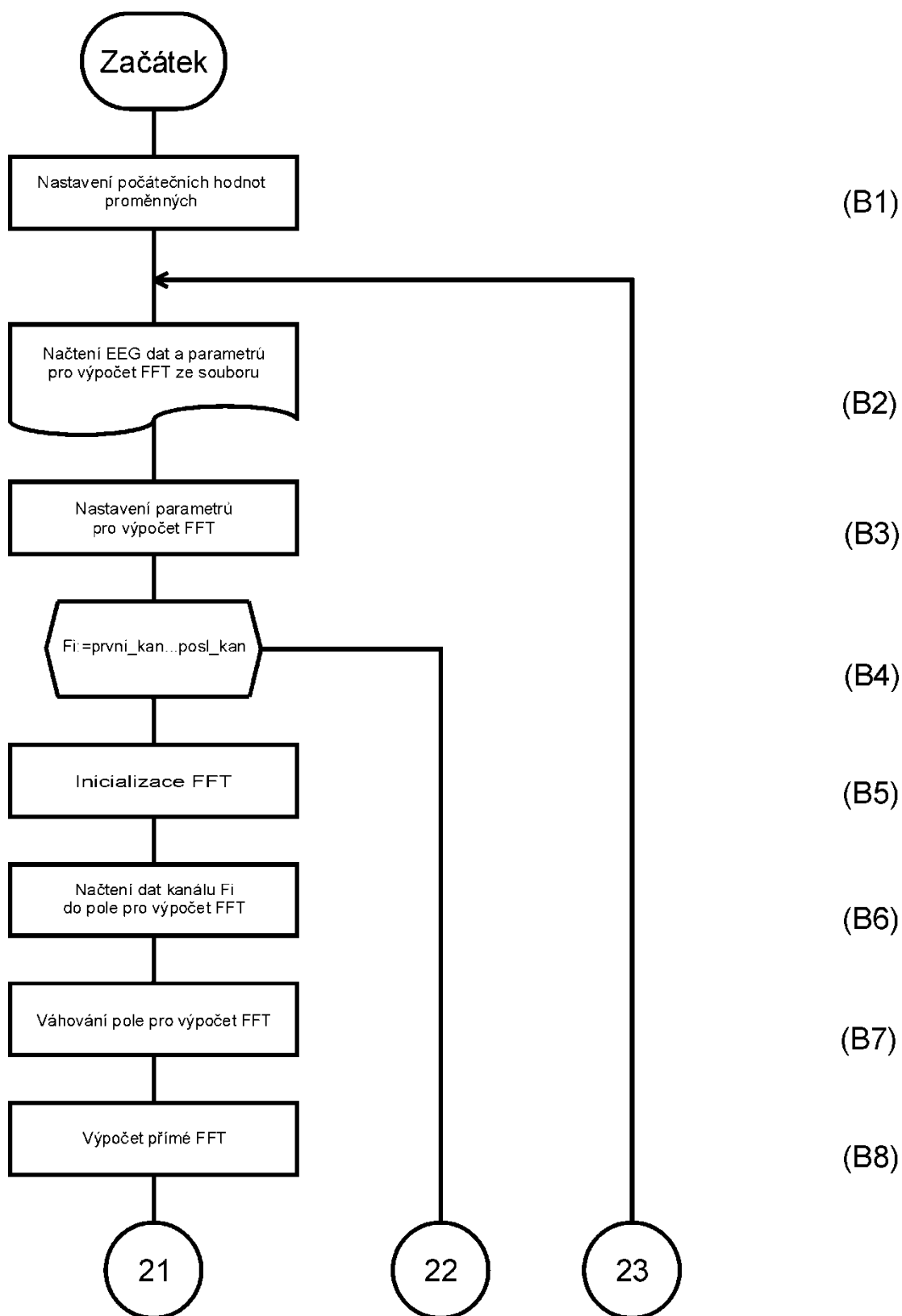
Aplikace FFTnew je automaticky spouštěna a ukončována programem Mexický příliv. Po svém startu se okno minimalizuje, protože uživatel v této aplikaci nic nenastavuje. Všechny nastavované parametry jsou obsaženy v okně aplikace Mexický příliv v záložce "Nastavení FFT" a jsou posílány spolu s daty. Přesto aplikace obsahuje dvě tlačítka. Tlačítko "Alternativní zapnutí" umožňuje spustit aplikaci v případě, že jste ji omylem ukončili nebo ji automaticky nespustila aplikace Mexický příliv (ještě se mi nestalo). Druhým tlačítkem je tlačítko "Alternativní ukončení", které umožňuje ukončení aplikace v libovolném okamžiku nebo kdyby ji neukončila aplikace Mexický příliv (také se ještě nestalo).



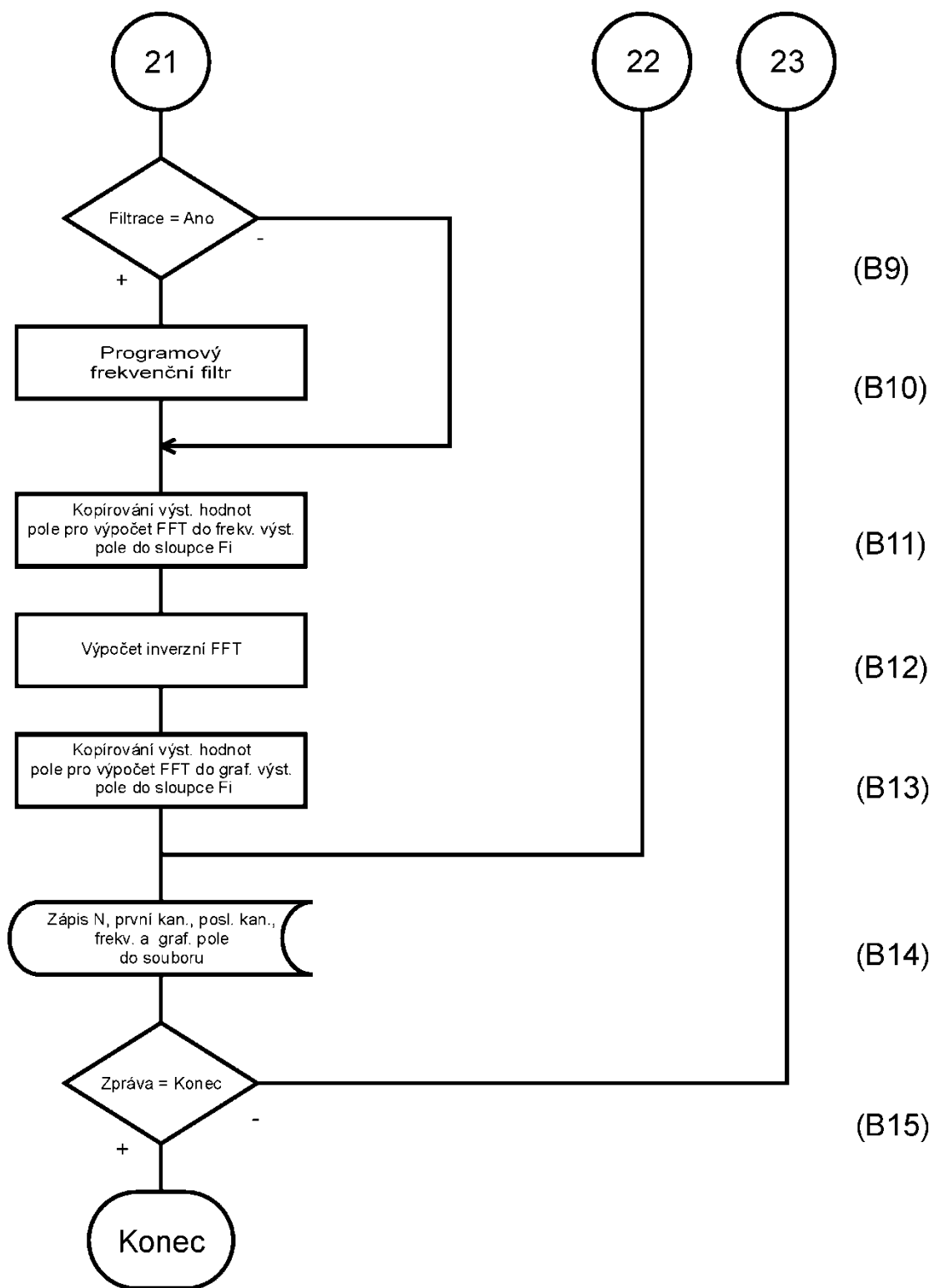
Obr.39. Okno aplikace FFTnew

Aplikace načte předzpracovaná data EEG signálů, které jsou uložena na disku aplikací Mexický příliv. Před těmito daty jsou v souboru uloženy také informace, které potřebuje aplikace znát pro svou činnost. Proveďte se okénkovací funkce a poté přímá FFT. Dále se provede s výsledky frekvenční filtrace, pokud je požadována. Poslední krok je inverzní FFT. Výsledky, to jest frekvenční spektrum a filtrované průběhy EEG signálů jsou uloženy do souboru pro zobrazení a další analýzu v programu Mexický příliv. Cyklus od načítání dat až po jejich uložení se provádí do té doby, než aplikace Mexický příliv nepošle zprávu k ukončení aplikace.

7.5.3.2. Vývojový diagram programu FFTnew



Obr.40. Vývojový diagram programu FFTnew strana 1



Obr.41. Vývojový diagram programu FFTnew strana 2

7.5.3.3. Popis vývojového diagramu programu FFTnew

B1

- nastavení konstant, nastavení počátečních hodnot proměnných, vynulování pole pro ukládání hodnot EEG signálu ze souboru

B2, B3

- zde začíná cyklus, který se opakuje do té doby, dokud uživatel neukončí program Mexický příliv
- ze souboru se načtou předzpracovaná data EEG signálů v rozsahu aktivních kanálů hlavního programu
- dále se načtou ze souboru parametry, podle kterých se počítá FFT, jsou to :
počet dat FFT, první a poslední aktivní kanál, minimální a maximální frekvenční rozsah, ve kterém budou po provedení FFT data dál zpracovávána, hodnoty frekvenčních filtrů (zádrží), druh váhovacího okna
tyto hodnoty se nastaví do pracovních proměnných

B4

- v tomto bodu začíná cyklus, který provede FFT a IFFT pro každý aktivní kanál

B5

- provede se inicializace FFT pro N dat

B6

- načtou se data ze vstupního pole F_i -tého kanálu do reálné části pole pro výpočet FFT, imaginární část tohoto pole se vynuluje

B7

- provede se váhování reálných částí hodnot pole pro výpočet FFT podle zvoleného váhovacího okna v programu Mexický příliv
- k dispozici jsou Hammingovo, Hanningovo, Dirichletovo a Blackmanovo okno
- pro Hanningovo okno se provede
pro $i=0...(N-1)$
 $\text{pole}[i].r := \text{apole}[i].r * (0.5 - 0.5 * \cos((2 * \pi / N) * i))$
- pro Hammingovo okno se provede
pro $i:=0...(N-1)$

$\text{pole}[i].r := \text{pole}[i].r * (0.54 - 0.46 * \cos(2 * \pi / N * i)) ;$

- pro Blackmanovo okno se provede
pro $i := 0 \dots (N-1)$
 $w := 2.0 * \pi * i / N ;$
 $t := bc1 - bc2 * \cos(w) + bc3 * \cos(2.0 * w) - bc4 * \cos(3.0 * w) ;$
 $\text{apole}[i].r := \text{pole}[i].r * t / bc1 ;$
kde $bc1 = 0,35875$, $bc2 = 0,48829$, $bc3 = 0,14128$, $bc4 = 0,01168$
- pro Dirichletovo okno se neprovádí žádná akce

B8

- provede se výpočet přímé FFT pro N prvků, výsledky jsou vráceny ve stejném poli

B9, B10

- pokud uživatel zvolil, že chce provést frekvenční filtraci při Fourierově transformaci, pak se u každé položky pole pro výpočet FFT přepočítá číslo harmonické složky na frekvenci a jestliže tato frekvence je v rozsahu frekvenční zádrže určené uživatelem, tak se položka pole dané harmonické složky vynuluje (reálná i imaginární část), to samé se provede pro položku pole N mínus daná harmonická, kde N je počet dat FFT
- výpočet frekvence z čísla harmonické složky:
 $\text{Koeff} = 1 / (N / f_{\text{VZORKOVACÍ AD}})$, kde N je počet dat pro výpočet FFT
 $f = n * \text{Koeff}$, kde n je číslo harmonické

B11

- vypočítané frekvenční spektrum se uloží do sloupce výstupního pole frekvenčního spektra pro daný kanál

B12

- provede se inverzní FFT, výsledek je znovu v poli určeném pro výpočet FFT

B13

- výsledek inverzní FFT se zkopíruje do sloupce výstupního pole grafického zobrazení pro daný kanál
- zde končí cyklus, který se provádí pro první až poslední kanál

B14

- do výstupního souboru se zapíše data pro program Mexický příliv: počet dat pro FFT, první a poslední kanál, poloviční hodnota počtu dat pro FFT, koeficient pro výpočet frekvence z harmonické složky, poté následují data pole grafického výstupu pro první až poslední aktivní kanál a nakonec jsou uložena data z pole frekvenčního spektra pro první až poslední aktivní kanál

B15

- pokud od programu Mexický příliv nepřišla pomocí funkce mapování souborů zpráva s povelom ukončit aplikaci, opakuje se cyklus od bodu B2

7.5.4. Program Generátor

7.5.4.1. Popis programu Generátor

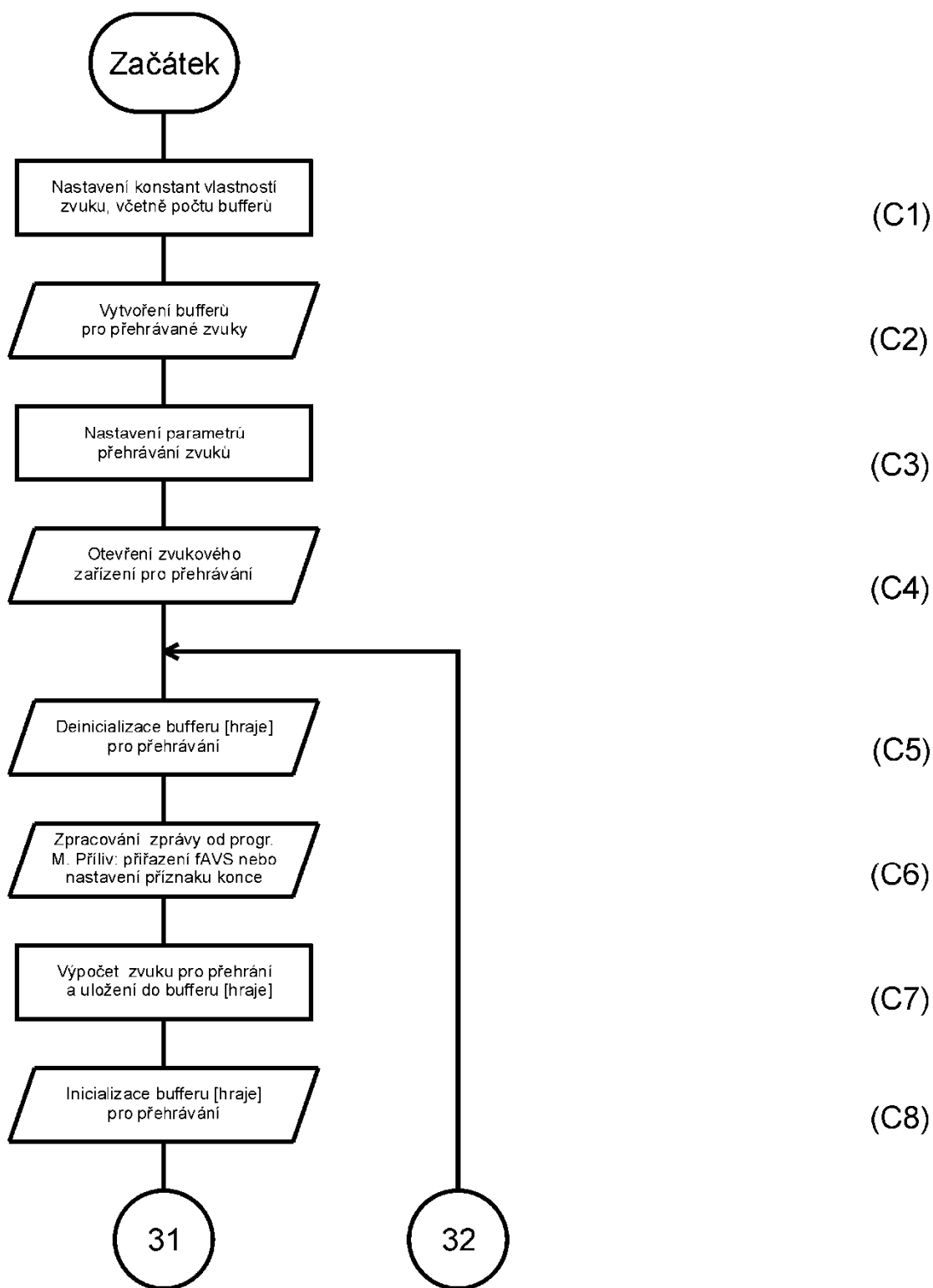
Aplikace Generátor je automaticky spouštěna a ukončována programem Mexický příliv. Po svém startu se okno minimalizuje, protože uživatel v této aplikaci nic nemusí nastavovat. Okno aplikace má sice dvě editační okna pro nastavení frekvence prvního a druhého signálu pro dvojité binaurální rytmy, ale nastavené hodnoty jsou nejpoužívanější. Upozorňuji, že frekvence v těchto polích by neměly být větší než 1500 Hz, protože nad tuto hodnotu, již mozek tyto zvuky nevnímá jako binaurální rytmy. Okno dále obsahuje dvě tlačítka. Tlačítko "Alternativní zapnutí/stop" umožňuje spustit/zastavit generování zvuků. Druhým tlačítkem je tlačítko "Alternativní ukončení", které umožňuje ukončení aplikace v libovolném okamžiku nebo kdyby ji neukončila aplikace MP (ještě se nestalo).



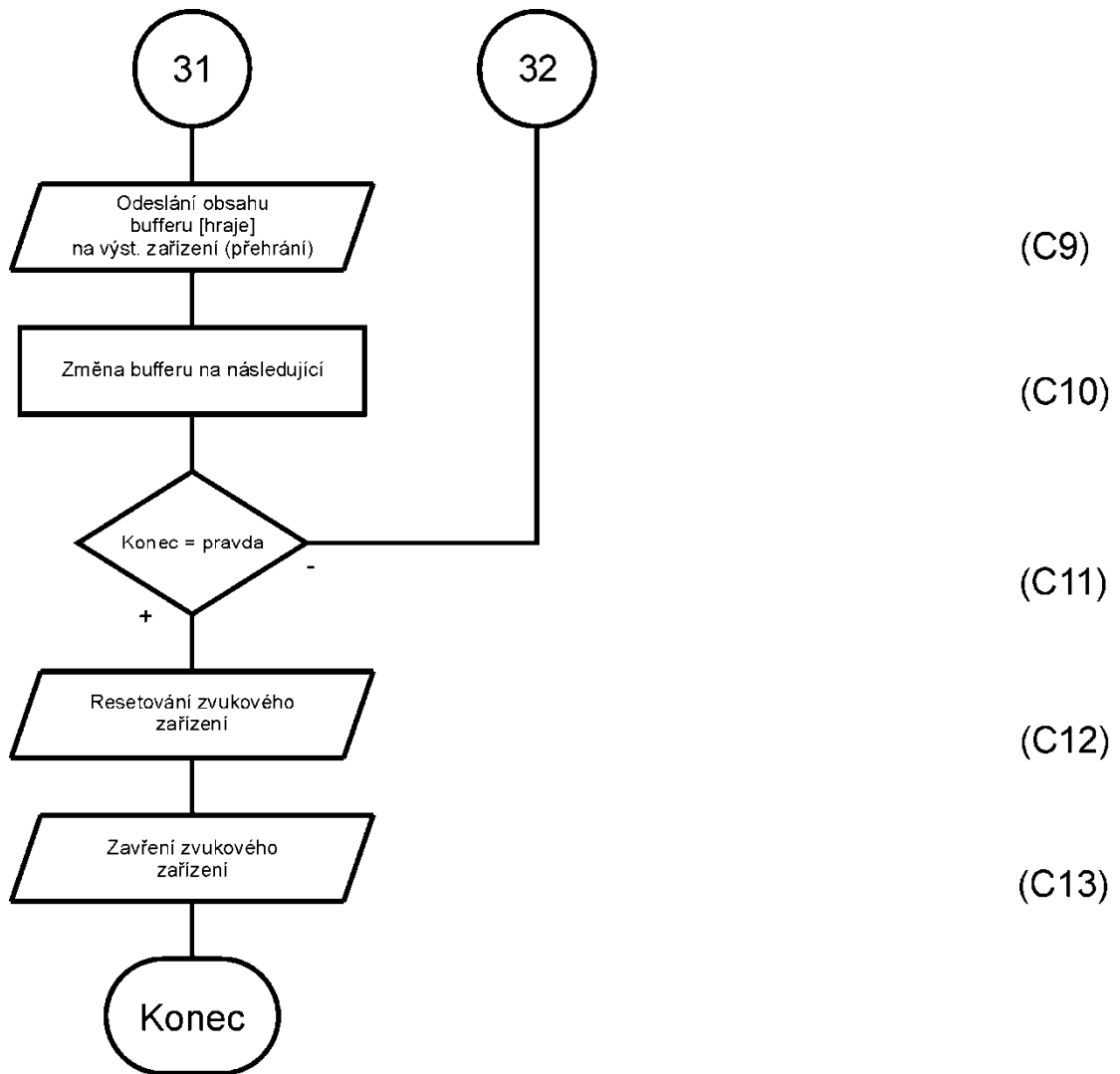
Obr.42. Okno aplikace Generátor

Aplikace Generátor generuje dvojité binaurální rytmy a signál AudioStrobe, pro jejichž výpočet je použita stimulační frekvence vypočítaná programem Mexický příliv. Program funguje na principu střídání bufferů, kdy do jednoho zapisujeme data zvukového signálu a obsah druhého je v té samé chvíli přehráván zvukovou kartou. Poté se funkce obou bufferů vymění. Použitím tohoto principu a využitím přímo funkcí programového prostředí WinApi pro práci se zvukem nedochází při přehrávání zvuku k nepříjemnému praskání mezi navazujícími částmi přehrávaných zvuků. Jedinou informací, kterou program potřebuje pro správný výpočet generovaných zvuků je stimulační frekvence, kterou posílá aplikace Mexický příliv pomocí funkce mapování souborů. Stejnou cestou dostane aplikace také povel ke svému ukončení.

7.5.4.2. Vývojový diagram programu Generátor



Obr.43. Vývojový diagram programu Generátor strana 1



Obr.44. Vývojový diagram programu Generátor strana 2

7.5.4.3. Popis vývojového diagramu programu Generátor

C1

- nastaví se konstanty vlastností přehrávaného zvuku a počet bufferů
- vzorkovací frekvence = 48 000
- počet kanálu = 2 (1-mono, 2-stereo)
- počet vzorku v bufferu = 48 000
- rozlišení vzorků = 2 (16 bitové)
- počet bufferů = 2
- polovina minimální amplitudy signálu AdioStrobe = 4000 (max. signálu = $65535/2$)
- polovina maximální amplitudy signálu AdioStrobe = 16000
- polovina amplitudy 1. signálu pro dvojité binaurální rytmy = 8000
- polovina amplitudy 2. signálu pro dvojité binaurální rytmy = 8000
- frekvence signálu AudioStrobe = 19200
- počáteční frekvence 1. signálu = 220
- počáteční frekvence 2. signálu = 440

C2

- vytvoří se buffery pro přehrávané zvuky, pro kontinuální přehrávání je potřeba mít alespoň 2, v našem případě jsou vytvořeny 2 buffery
- velikost bufferu = počet_vzorků_v_bufferu * počet_kanálu * rozlišení_vzorků

C3

- nastaví se parametry přehrávání zvuku
- počet bitů ve vzorku = $8 * \text{rozlišení_vzorků}$
- formát zvuku = WAV PCM
- kanálů = počet_kanálu
- vzorků za sekundu = vzorkovací_frekvence
- vyřizované bloky = počet_vzorků * rozlišení_vzorků
- průměrně bajtů za sekundu = počet_kanálu * rozlišení_vzorků * vzorkovací_frekvence

C4

- otevře se zvukové zařízení pro přehrávání s parametry v bodě C3

C5

- zde začíná cyklus přehrávání a tvoření zvuků
- provede se deinicializace jednoho z bufferů pro přehrávání, to znamená, že nyní do něj můžeme bezkonfliktně zapsat data nového přehrávaného zvuku nebo jeho části

C6

- přiřazení frekvence f_{AVS} , která přišla pomocí funkce mapování souborů z aplikace Mexický příliv, do proměnné G_{frekv} nebo jestliže přišel řetězec "Konec" nastaví se příznak ukončení programu a proměnná G_{frekv} se nastaví na nulovou hodnotu

C7

- vytvoří se zvukový signál, který vznikne sečtením tří zvukových složek
- první složku tvoří sinusový průběh signálu AudioStrobe o frekvenci 19200 Hz, jehož amplituda se lineárně mění od hodnoty $2 * \text{polovina_minimální_amplitudy_signálu_AudoStrobe}$ po hodnotu $2 * \text{polovina_maximální_amplitudy_signálu_AudiStrobe}$ a zpět, s frekvencí G_{frekv} .
- druhou složku tvoří sinusový průběh o amplitudě $2 * \text{polovina_amplitudy_1.signálu}$ a frekvenci $\text{frekvence_1.signálu}$ pro pravý kanál a u levého kanálu je frekvence rovna $\text{frekvence_1.signálu} + G_{frekv}$
- třetí složku tvoří sinusový průběh o amplitudě $2 * \text{polovina_amplitudy_2.signálu}$ a frekvenci $\text{frekvence_2.signálu}$ pro pravý kanál a u levého kanálu je frekvence rovna $\text{frekvence_2.signálu} + G_{frekv}$
- vytváří se vždy průběh o délce 1 sekundy
- všechny tři složky se pro každý kanál uloží do bufferu pro přehrávání

C8

- buffer, do kterého jsme uložili data se stává aktivním pro přehrávání zvuku

C9

- odešle se obsah bufferu na výstupní zvukové zařízení, čímž prakticky dojde k přehrávání zvuku

C10

- změní se buffer se kterým nyní chceme pracovat, tzn.. ten druhý, který není přehráván

C11

- jestliže příznak konce nemá hodnotu pravda, provede se další cyklus tvorby a přehrání zvuku od bodu C5

C12, C13

- provede se resetování zvukového zařízení a zavření zvukového zařízení, poté se program ukončí

7.6. Výsledky stimulace

Výsledky stimulace jsou ukládány v souboru Dokument. Následující tabulka obsahuje ukázkou výpisu o délce přibližně jedné minuty.

Jednotlivými sloupci jsou:

- Čas - doba od spuštění stimulace v minutách
- fMaxZK - frekvence s maximální amplitudou zvoleného zobrazovaného signálu
- fDomK - dominantní frekvence jednotlivých aktivních kanálů
- fDom - celková dominantní frekvence
- fAVS - frekvence audiovizuální stimulace

Tab.7. Výpis části souboru Dokument:

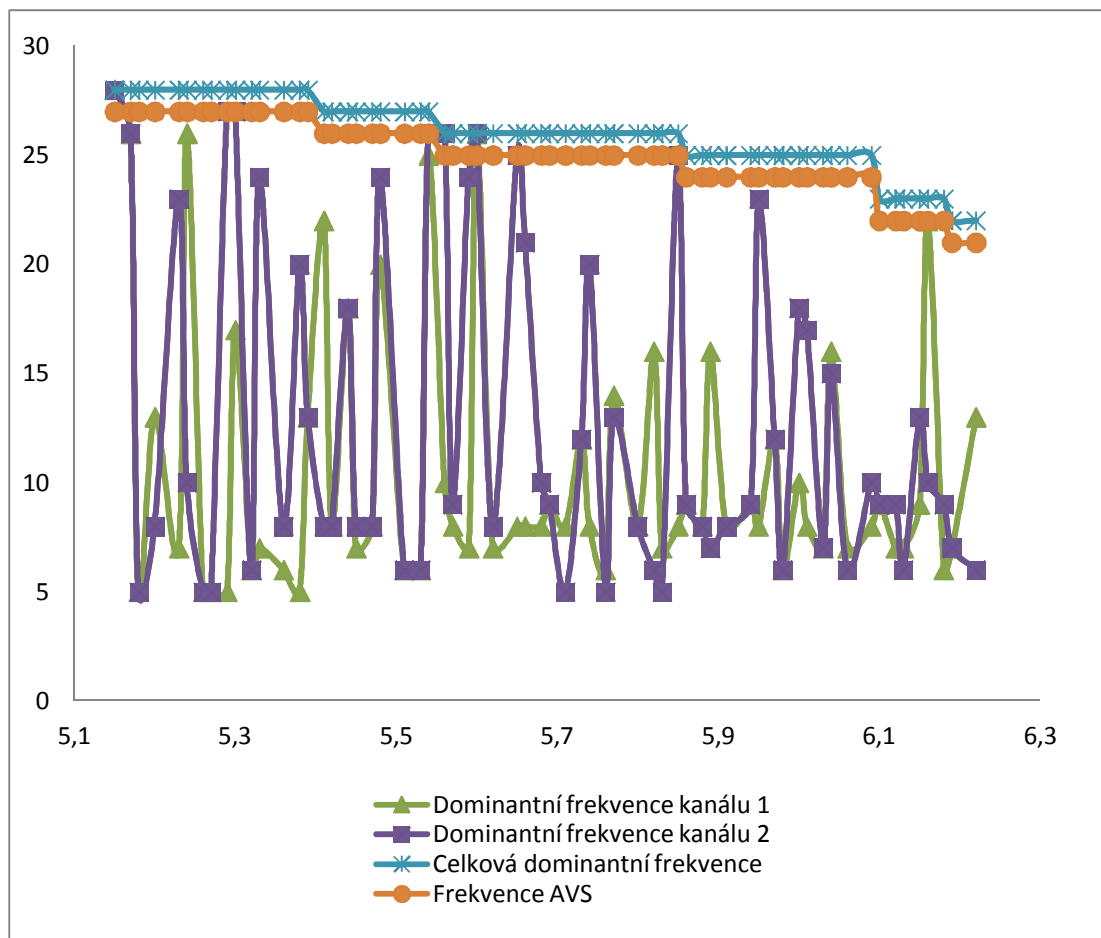
čas[min]	fMaxZK1[Hz]	fMaxZK2[Hz]	fDomK1[Hz]	fDomK2[Hz]	fDom[Hz]	fAVS[Hz]
5,15	28	28	28	28	28	27
5,17	26	26	26	26	28	27
5,18	5	5	5	5	28	27
5,2	13	8	13	8	28	27
5,23	7	23	7	23	28	27
5,24	26	10	26	10	28	27
5,26	5	5	5	5	28	27
5,27	5	5	5	5	28	27
5,29	5	27	5	27	28	27
5,3	17	27	17	27	28	27
5,32	6	6	6	6	28	27
5,33	7	24	7	24	28	27
5,36	6	8	6	8	28	27
5,38	5	20	5	20	28	27
5,39	13	13	13	13	28	27
5,41	22	8	22	8	27	26
5,42	8	8	8	8	27	26
5,44	18	18	18	18	27	26
5,45	7	8	7	8	27	26
5,47	8	8	8	8	27	26
5,48	20	24	20	24	27	26
5,51	6	6	6	6	27	26
5,53	6	6	6	6	27	26
5,54	25	26	25	26	27	26
5,56	10	26	10	26	26	25

čas[min]	fMaxZK1[Hz]	fMaxZK2[Hz]	fDomK1[Hz]	fDomK2[Hz]	fDom[Hz]	fAVS[Hz]
5,59	7	24	7	24	26	25
5,6	26	26	26	26	26	25
5,62	7	8	7	8	26	25
5,65	8	25	8	25	26	25
5,66	8	21	8	21	26	25
5,68	8	10	8	10	26	25
5,69	9	9	9	9	26	25
5,71	8	5	8	5	26	25
5,73	12	12	12	12	26	25
5,74	8	20	8	20	26	25
5,76	6	5	6	5	26	25
5,77	14	13	14	13	26	25
5,8	8	8	8	8	26	25
5,82	16	6	16	6	26	25
5,83	7	5	7	5	26	25
5,85	8	25	8	25	26	25
5,86	9	9	9	9	25	24
5,88	8	8	8	8	25	24
5,89	16	7	16	7	25	24
5,91	8	8	8	8	25	24
5,94	9	9	9	9	25	24
5,95	8	23	8	23	25	24
5,97	12	12	12	12	25	24
5,98	6	6	6	6	25	24
6	10	18	10	18	25	24
6,01	8	17	8	17	25	24
6,03	7	7	7	7	25	24
6,04	16	15	16	15	25	24
6,06	7	6	7	6	25	24
6,09	8	10	8	10	25	24
6,1	9	9	9	9	23	22
6,12	7	9	7	9	23	22
6,13	7	6	7	6	23	22
6,15	9	13	9	13	23	22
6,16	22	10	22	10	23	22
6,18	6	9	6	9	23	22

Stimulace byla provedena s těmito parametry:

- Počet aktivních kanálů - 2
- Relaxační frekvence - 7 Hz
- Frekvence bdělého stavu - 20 Hz
- Celková doba relaxace (stimulace) - 10 minut
- Změna frekvence pro AVS - Jemné doladování - Ano
- Pracovní kanály - 1, 2
- Počet dat pro FFT - 256
- Druh okna pro FFT - Hanningovo
- Vyhodnocovací frekvenční rozsah FFT - 0-42 Hz
- Frekvenční zádrže - 0-4 Hz, 35-45 Hz, 45-60 Hz
- Výpočet celkové dominantní frekvence z posledních - 15

Grafické znázornění vybraného úseku stimulace



Obr.45. Grafické znázornění výsledků stimulace

7.7. Vyhodnocení výsledků

Z výsledků ukládaných do souboru Dokument je patrné, že program i hardwarová část projektu jako celek pracují správně, to co by bylo třeba vylepšit je výpočet dominantní frekvence pro jednotlivé kanály. Bohužel se mi v průběhu více jak tří měsíčního experimentování s různými možnostmi výpočtu dominantní frekvence nepodařilo nalézt žádné uspokojivé řešení.

8. Závěr

Prvním krokem bylo navrhnutí a sestavení přístroje pro audiovizuální stimulaci s EEG zpětnou biologickou vazbou. Jako základní koncept zapojení jsem se rozhodl použít EEG zesilovač, AD převodník se sériovým výstupem dat a osobní počítač, na kterém poběží program pro analýzu EEG signálů a výpočet nejvhodnější stimulační frekvence. Ta je předávána AVS přístroji pomocí technologie AudioStrobe. Největší předností tohoto konceptu je možnost snadno měnit analyzační program podle dosažených poznatků během testování zařízení a ušetření použití dalších komponent (např. jednočipového procesoru) při využití osobního počítače, který dnes vlastní prakticky každý.

Jako komponenty hardwarové části byly zvoleny modul EEG zesilovače s horní a dolní propustí, modul AD převodníku umístěného v mikroprocesoru se sériovým rozhraním RS232, které je navíc opticky odděleno od počítače kvůli rušení a modul převodníku UART/USB rozhraní, protože většina moderních počítačů již má jen USB porty namísto sériových. Všechny tyto komponenty jsou sériově vyráběny, takže je stačí jen mezi sebou zapojit.

Použití technologie AudioStrobe pro komunikaci mezi počítačem a AVS přístrojem byla použita pro jednoduchost, se kterou lze zařízení zapojit, stačí Jack konektor zapojit do zvukové karty počítače a druhý konektor kabelu do AVS přístroje. Není tedy jakkoliv třeba zasahovat do AVS přístroje, ani sestavovat vlastní stimulační přístroj. Stačí jakýkoliv AVS přístroj, který podporuje technologii AudioStrobe.

Druhou část projektu, řekl bych i stěžejní, tvoří tři programy běžící na osobním počítači. Byly naprogramované v jazyce Delphi 7 speciálně pro tuto aplikaci. Prvním je program Mexický příliv. Je to jediný program, který uživatel spouští a obsluhuje. Slouží k nastavení všech potřebných parametrů analýzy signálů a stimulace, načítání zesílených digitalizovaných EEG signálů, grafickému zobrazení těchto signálů a jejich frekvenčních spekter, výpočtu nejlepší stimulační frekvence. Druhým programem je program FFTnew počítajícím FFT. Třetím programem je program Generátor tvořící a přehrávající dvojité binaurální rytmy se složkou AudioStrobe. Všechny tři programy běží současně a jen si vyměňují mezi sebou potřebná data. Použitím několika programů oproti jednomu je dosaženo vyšší rychlosti běhu celé aplikace.

Při tvorbě uživatelského prostředí programů jsem kladl důraz nejen na možnost nastavení všech důležitých parametrů, ale i na přátelské a přehledné prostředí. Programy jsou stabilní a bez problémů běží pod operačním systémem Windows XP a Vista.

Pro zkušenějšího programátora mohou být vytvořené programy inspirací, nejen pro svoji jedinečnost jako celek, ale i jednotlivé části se dají odděleně použít pro rozličné aplikace. Jako příklad bych uvedl program nazvaný Generátor, který generuje signál pro AVS přístroj využívající technologii AudioStrobe. S malými úpravami a se sestavením zapojení s asi pěti součástkami lze vyrobit funkčně plnohodnotný AVS přístroj za zlomek ceny prodávaných přístrojů. Jako druhý příklad bych uvedl programovou jednotku nazvanou UMS3, která slouží ke komunikaci se stejnojmenným modulem. Na stránkách výrobce, ani na internetu jsem nenašel plně fungující ovládací program či jednotku psanou v jazyce Delphi pro tento modul. Takže toto je první verze, o které vím.

Z výsledků testování zařízení vyplývá, že hardwarová i softwarová část pracují správně. Jediným slabým místem je výpočet dominantních frekvencí pro jednotlivé EEG signály. Bohužel všechny vyzkoušené metody výpočtu se ukázaly jako neuspokojivé. Nakonec byla zvolena metoda, kdy dominantní frekvence je frekvence s maximální frekvencí z frekvenčního spektra a celková dominantní frekvence je největší frekvence z posledních X vypočítaných dominantních frekvencí.

V průběhu tvorby a testování zařízení jsem nabyl přesvědčení, že zavedení zpětné EEG biologické vazby u audiovizuální stimulaci má velký potenciál a je dobré pokračovat tímto směrem. Jak se ukázalo, náš mozek velmi dobře a rychle reaguje na stimulační frekvenci blízkou aktuální frekvenci EEG.

9. Seznam použité literatury

- [1] - Stanislav Trojan, lékařská fyziologie,
Praha: Grada Publishing, 2003, ISBN 80-247-0512-5
- [2] - Richard Rokyta, Fyziologie
Praha: nakladatelství ISV, 2008, ISBN 80-86642-47-X
- [3] - Jindřich Mourek, Fyziologie - učebnice pro studenty nezdr. oborů
Grada Publishing, 2005, ISBN 80-247-1190-7
- [4] - Michael Landgraf, Stavby vědomí & audiovizuální stimulace
Praha : Galaxy 2000, ISBN 80-902870-0-X
- [5] - J. M. Valuch, Neurotechnologie, mozek a souvislosti
Praha : Galaxy 1998, ISBN 80-238-2446-5
- [6] - Jan Valuch, Průvodce světem AVS přístrojů - psychowalkmanů
Praha : Galaxy Czech 2007, ISBN 978-80-903202-1-5
- [7] - Jana Zahradníková, Diplomová práce Testování účinnosti psychowalkmana
České vysoké učení technické v Praze - Fakulta elektrotechnická,
rok obhajoby 2003
- [8] - Svatoš, J. Biologické signály 1- Geneze, zpracování a analýza.
Praha: České vysoké učení technické, 1998. ISBN 80-01-01822-9.
- [9] - Ernst Niedermeyer, Fernando Lopes da Silva,
Electroencephalography: Basic Principles, Clinical Applications, and Related
Fields
Lippincott Williams & Wilkins, 2004, ISBN 0781751268, 9780781751261.
- [10] - V. Hlaváč, M. Sedláček, Zpracování signálů a obrazů.
Praha : Vydavatelství ČVUT, 2002, ISBN 80-01-03110-1
- [11] - Uhlíř, Jan. Číslíkové zpracování signálů. 2. vyd.,
Praha: Vydavatelství ČVUT, 2002, ISBN 80-01-02613-2
- [12] - Čížek, Václav. Diskrétní Fourierova transformace a její použití,
Praha : SNTL, 1981
- [13] - Kainka, Burkhard; Využití rozhraní PC pod Windows, Ostrava: HEL, 2000.
ISBN 80-86167-13-5
- [14] - David Matoušek, USB prakticky s obvody FTDI 1. díl
Praha: Ben 2003, ISBN 80-7300103-9

- [15] - Michal Procházka, Mojmír Strakoš, Borland Delphi průvodce vývojáře
Praha: Mobil media 2002, ISBN 80-86593-31-2
- [16] - L. Svoboda, P. Voneš, T. Konšal, M. Mareš, 1001 tipů a triků pro Delphi
Praha: Computer Press 2001, ISBN 80-7226-529-6
- [I1] - Interstate Industries, Inc./Monroe Products, Změněné stavy vědomí poslechem
CD s Hemi-Sync
[online] 2004, cit. - 17.5.2009
<http://www.bezvaportal.cz/hemi-sync-proces/>
- [I2] - Syntetic Systems International, Inc., User Manual AVS Orion
[online] 2000, cit. - 17.5.2009
<http://www.mindplace.com/productsLS.htm#>
- [I3] - Lazina med., Positioning of EEG electrodes according to the international
10-20 electrode placement System
[online] 2004, cit. - 10.5.2009
<http://www.lazina.com/english/1020sys.htm>
- [I4] - Jiří Slavík, EEG Workshop
[online] 2002, cit. - 1.5.2009
<http://www.volny.cz/slavij/DokumentaceHTML/Dokumentace.htm>
- [I5] - Aplikovaná fyzika, Mini, Diskrétní Fourierova transformace, Univerzita
Palackého v Olomouci
[online] 11.12.2003, cit. - 3.5.2009
<http://apfyz.upol.cz/ucebnice/down/mini/fourtrans.pdf>
- [I6] - Aleš Doležal, Programové vybavení pro frekvenční analýzu zvukového signálu
[online] 2006, cit. - 2.5.2009
https://www.stag.utb.cz/apps/stag/dipfile/index.php?download_this_unauthorized=3320
- [I7] - Hála Vojtěch, Zdrojové texty pro rychlou Fourierovu transformaci v jazyce
Pascal
[online] 22.4.2000, cit. - 2.5.2009.
<http://hyperkrychle.cz/fftpas.html>

- [I18] - Oto Gregořík, Indikátor rojové nálady včelstva
[online] 2007, cit. - 2.5.2009
http://uai.fme.vutbr.cz/szz/2007/DP_Gregorik.PDF
- [I19] - Project Open EEG, různí autoři
[online] 2003, cit. - 13.5.2009
<http://openeeg.sourceforge.net/>
- [I10] - Dan Griffiths, Nelo, Jim Peters, Andreas Robinson, Jack Spaar and Yaniv Vilnai, The ModularEEG Design
[online] 2003, cit. - 13.5.2009
http://openeeg.sourceforge.net/doc/modeeg/modeeg_design.html
- [I11] - Christoph Veigl, Microprocessor based Design for Biomedical Applications
[online] 2007, cit. - 15.5.2009
http://embsys.technikum-wien.at/staff/veigl/MDBA/Projects/Project_Example_Uart_ISR.pdf
- [I12] - Asix s.r.o., UMS3 převodník USB - UART
[online] 2007, cit. - 17.5.2009
http://www.asix.cz/a6_ums3.htm
- [I13] - David Majda, Sdílení paměti pomocí mapování souborů
[online] 19.7.2002, cit. 18.5.2009
<http://www.builder.cz/art/homepage/filemapping1.html>
- [I14] - Hekrhy, Diskusní fórum - Rutinu na generování tónů
[online] 28.8.2008, cit. 18.5.2009
<http://forum.zive.cz/viewtopic.php?f=922&t=1047295>

10. Seznam obrázků

- Obr.1. Řez pravého oka
- Obr.2. Zjednodušené schéma sítnice
- Obr.3. Zraková dráha
- Obr.4. Scala media a Cortiho orgán
- Obr.5. Zjednodušené schéma sluchové dráhy
- Obr.6. Přístroj pro AVS Orion
- Obr.7. Systém 10-20
- Obr.8. Označení elektrod
- Obr.9. Algoritmus DIT FFT - první krok
- Obr.10. Motýlek DIF FFT
- Obr.11. Schéma algoritmu osmibodové DIT FFT
- Obr.12. Průběhy váhovacích oken
- Obr.13. Celkové zapojení systému audiovizuální stimulace s EEG zpětnou biologickou vazbou včetně EEG elektrod a AVS přístroje
- Obr.14. Blokové schéma zapojení systému audiovizuální stimulace s EEG zpětnou biologickou vazbou
- Obr.15. Blokové schéma modulu EEG zesilovače pro jeden kanál
- Obr.16. Blokové schéma modulu mikroprocesoru-AD převodníku
- Obr.17. Ochranný vstupní obvod
- Obr.18. Rozdílový zesilovač
- Obr.19. Schéma zapojení horních a dolních propustí
- Obr.20. Schéma zapojení DRL zesilovače
- Obr.21. Schéma zapojení mikroprocesoru
- Obr.22. Schéma zapojení optických oddělovačů a převodníku úrovní TTL/sériového rozhraní
- Obr.23. Schéma zapojení napájecího obvodu
- Obr.24. Modul UMS3 (vlevo ze spodu, vpravo z vrchu)
- Obr.25. Schéma zapojení modulu UMS3
- Obr.26. Blokové schéma propojení všech programů včetně technických návazností
- Obr.27. Okno aplikace Mexický příliv
- Obr.28. Detail spodní části okna aplikace Mexický příliv
- Obr.29. Detail grafické části okna aplikace Mexický příliv

- Obr.30. Záložky v aplikaci Mexický příliv
- Obr.31. Vývojový diagram programu Mexický příliv strana
- Obr.32. Vývojový diagram programu Mexický příliv strana 2
- Obr.33. Vývojový diagram programu Mexický příliv strana 3
- Obr.34. Vývojový diagram programu Mexický příliv strana 4
- Obr.35. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 1
- Obr.36. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 2
- Obr.37. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 3
- Obr.38. Vývojový diagram programu Mexický příliv - Procedura Načítání a práce s AVS strana 4
- Obr.39. Okno aplikace FFTnew
- Obr.40. Vývojový diagram programu FFTnew strana 1
- Obr.41. Vývojový diagram programu FFTnew strana 2
- Obr.42. Okno aplikace Generátor
- Obr.43. Vývojový diagram programu Generátor strana 1
- Obr.44. Vývojový diagram programu Generátor strana 2
- Obr.45. Grafické znázornění výsledků stimulace
- Obr.46. Schéma zapojení: Dvoukanálový EEG zesilovač
- Obr. 47. Schéma zapojení: Modul - Izolovaný mikroprocesor/AD převodník
- Obr.48. Fotografie celkového zapojení zařízení
- Obr.49. Fotografie Modular EEG zesilovač EEG
- Obr.50. Fotografie Modular EEG mikroprocesor/AD převodník
- Obr.51. Fotografie Modular EEG zesilovač EEG v kovové krabici
- Obr.52. Fotografie síťového napájení pro testování zapojení
- Obr.53. Fotografie modulu UMS3
- Obr.54. Frekvenční přenosová charakteristika filtrů modulu EEG zesilovače

11. Seznam tabulek

Tab.1. Přenosový protokol P2

Tab.2. Parametry sériového přenosu mezi počítačem a mikroprocesorem

Tab.3. Proudové a napěťové parametry zapojení

Tab.4. Parametry EEG zesilovače

Tab.5. Parametry AD převodníku (Atmel ATmega8)

Tab.6. Popis pinů UMS3

Tab.7. Výpis části souboru Dokument

12. Seznam příloh

Příloha P1 Schéma zapojení: Dvoukanálový EEG zesilovač

Příloha P2 Schéma zapojení: Modul - Izolovaný mikroprocesor/AD převodník

Příloha P3 Celkové zapojení zařízení

Příloha P4 Fotografie Modular EEG zesilovač EEG

Příloha P5 Fotografie Modular EEG mikroprocesor/AD převodník

Příloha P6 Fotografie Modular EEG zesilovač EEG v kovové krabici

Příloha P7 Fotografie síťového napájení pro testování zapojení

Příloha P8 Fotografie modulu UMS3

Příloha P9 Frekvenční přenosová charakteristika filtrů modulu EEG zesilovače

Příloha P10 Výpis zdrojového kódu programu Mexický příliv

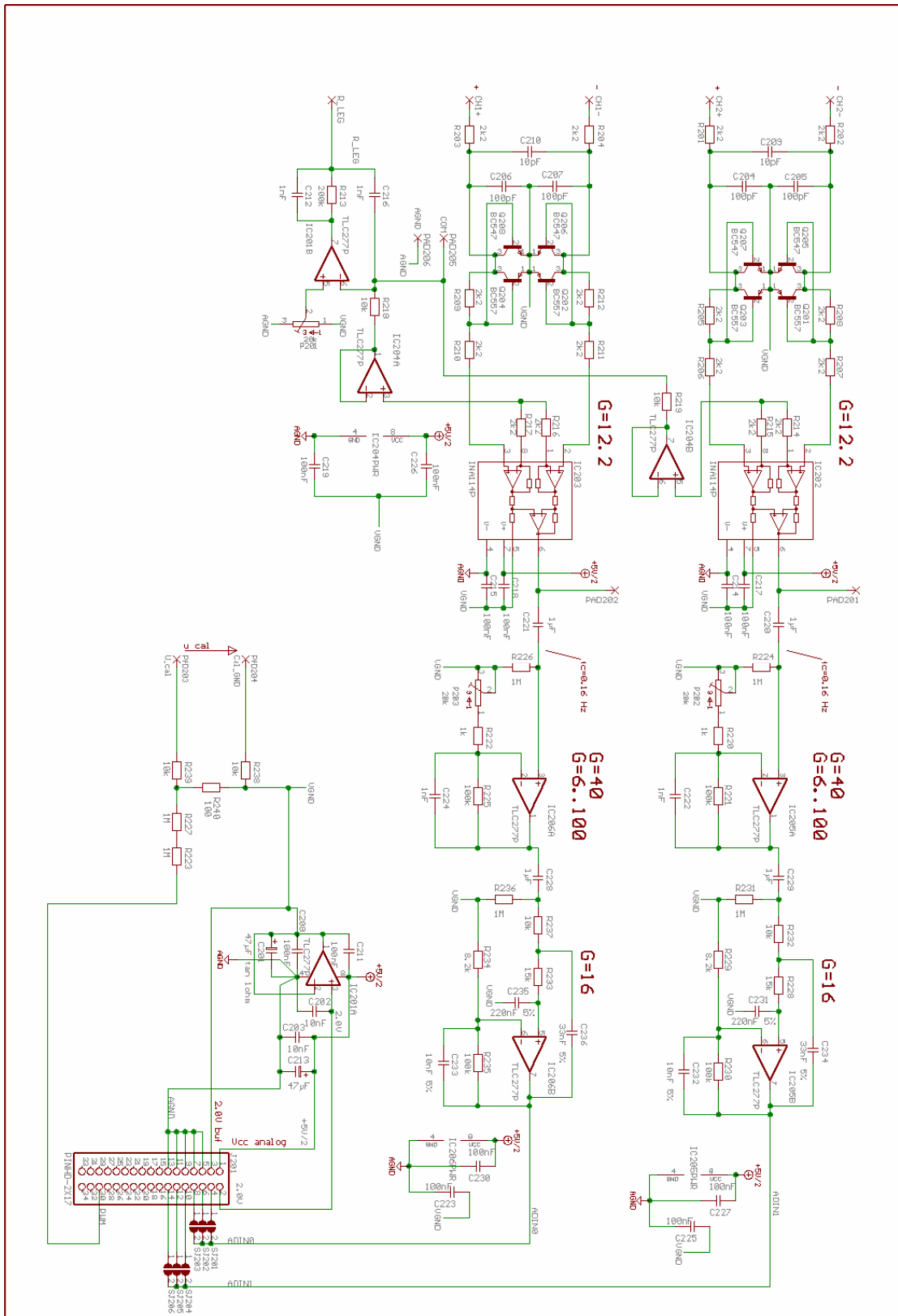
Příloha P11 Výpis zdrojového kódu programu FFTnew

Příloha P12 Výpis zdrojového kódu programu Generátor

Příloha P13 CD s programy Mexický příliv, FFTnew, Generátor

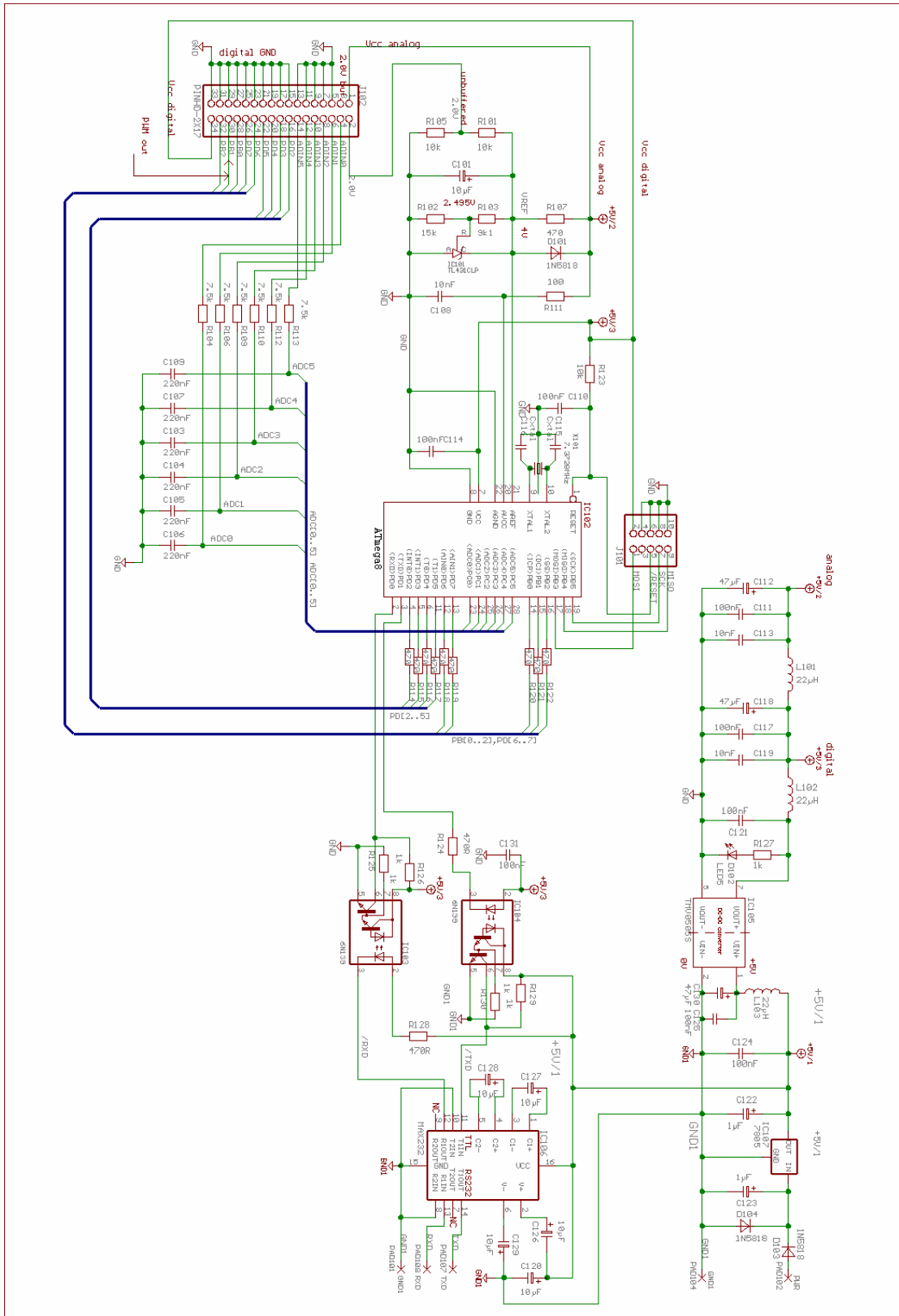
13. Přílohy

Příloha P1 Schéma zapojení: Dvoukanálový EEG zesilovač



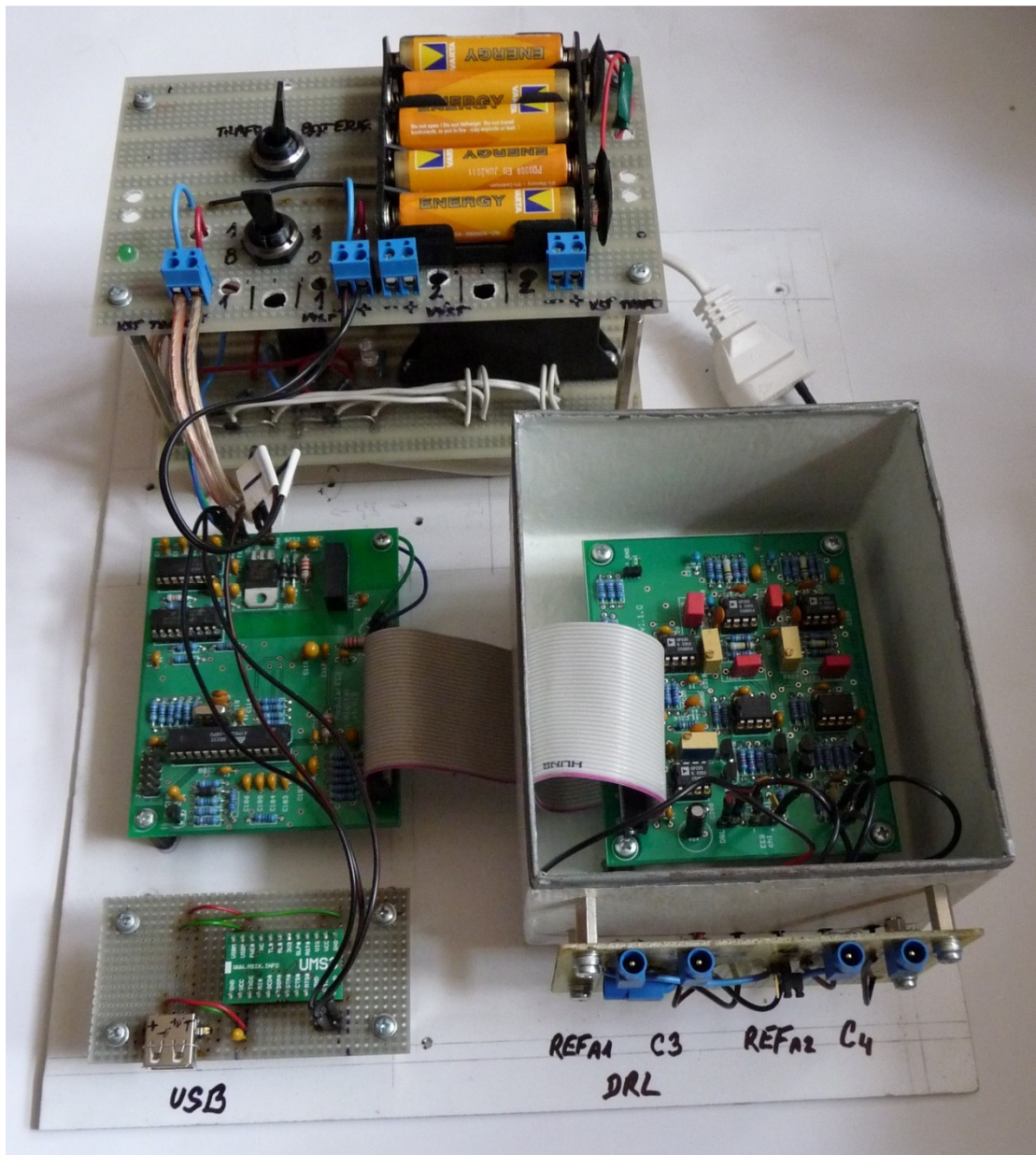
Obr.46. Schéma zapojení: Dvoukanálový EEG zesilovač

Příloha P2 Schéma zapojení: Modul - Izolovaný mikroprocesor/AD převodník



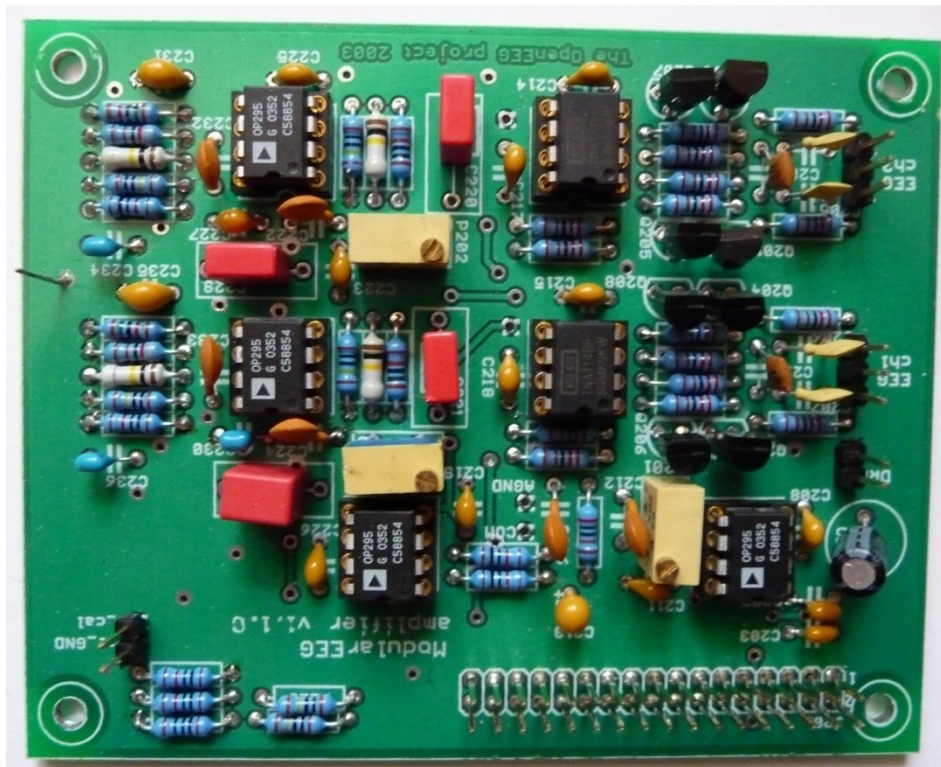
Obr. 47. Schéma zapojení: Modul - Izolovaný mikroprocesor/AD převodník

Příloha P3 Celkové zapojení zařízení



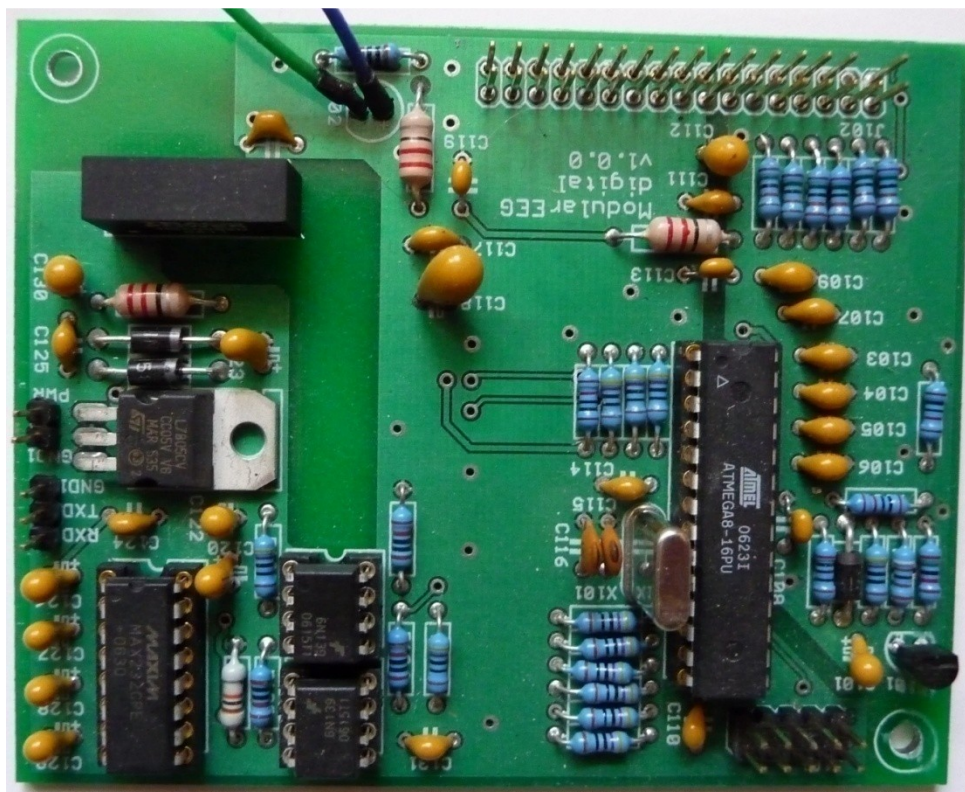
Obr.48. Fotografie celkového zapojení zařízení

Příloha P4 Fotografie Modular EEG zesilovač EEG



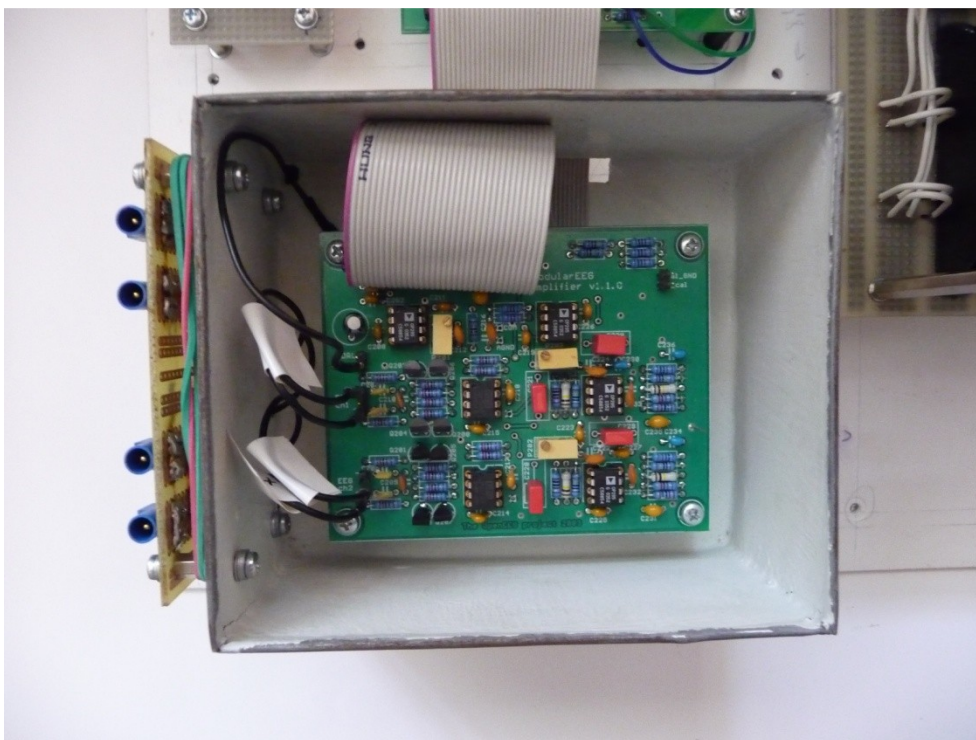
Obr.49. Fotografie Modular EEG zesilovač EEG

Příloha P5 Fotografie Modular EEG mikroprocesor/AD převodník



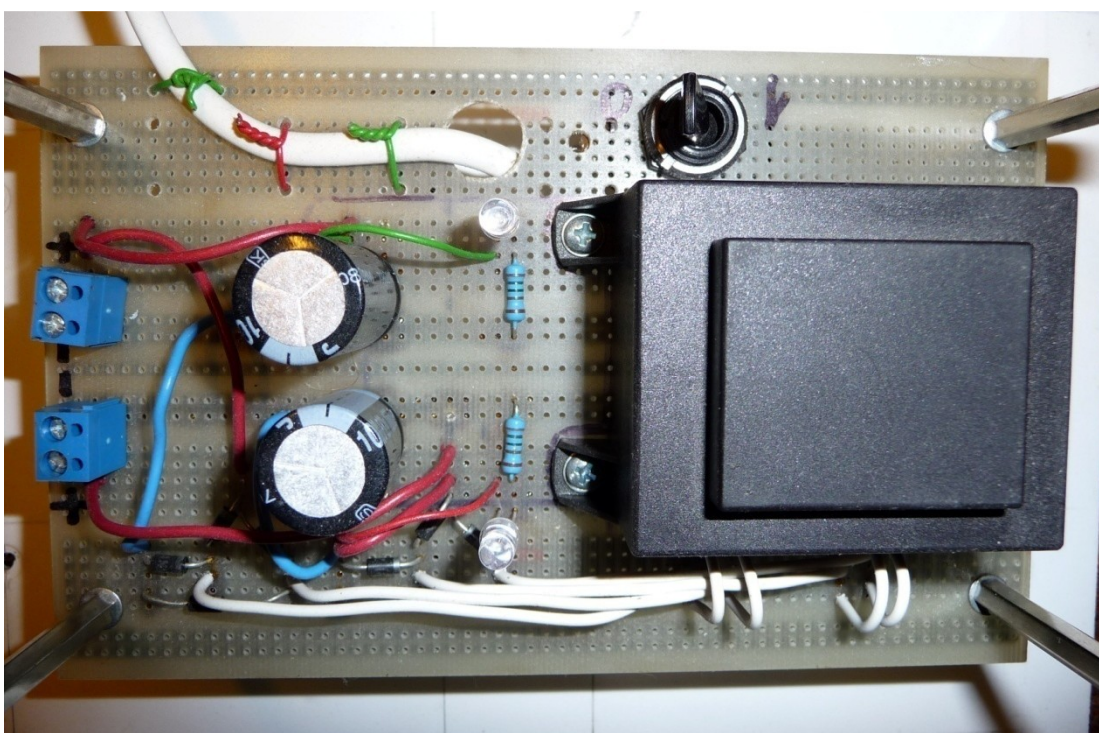
Obr.50. Fotografie Modular EEG mikroprocesor/AD převodník

Příloha P6 Fotografie Modular EEG zesilovač EEG v kovové krabici



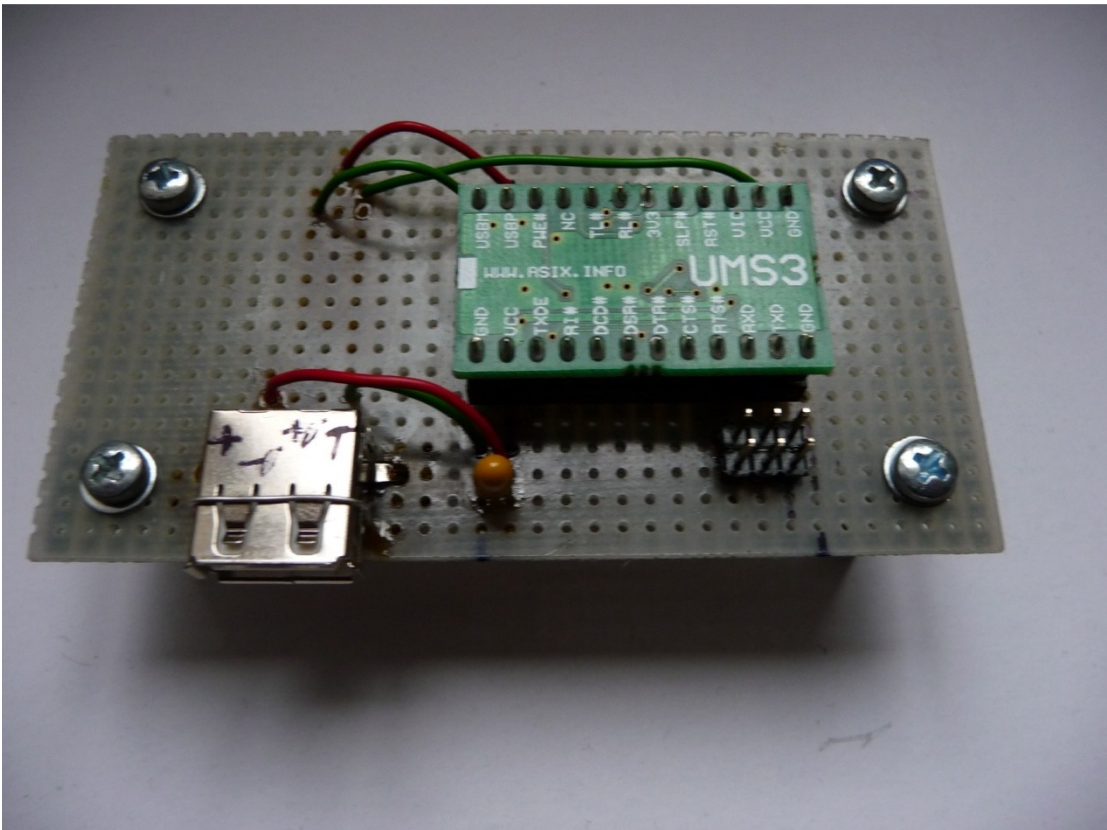
Obr.51. Fotografie Modular EEG zesilovač EEG v kovové krabici

Příloha P7 Fotografie síťového napájení pro testování zapojení

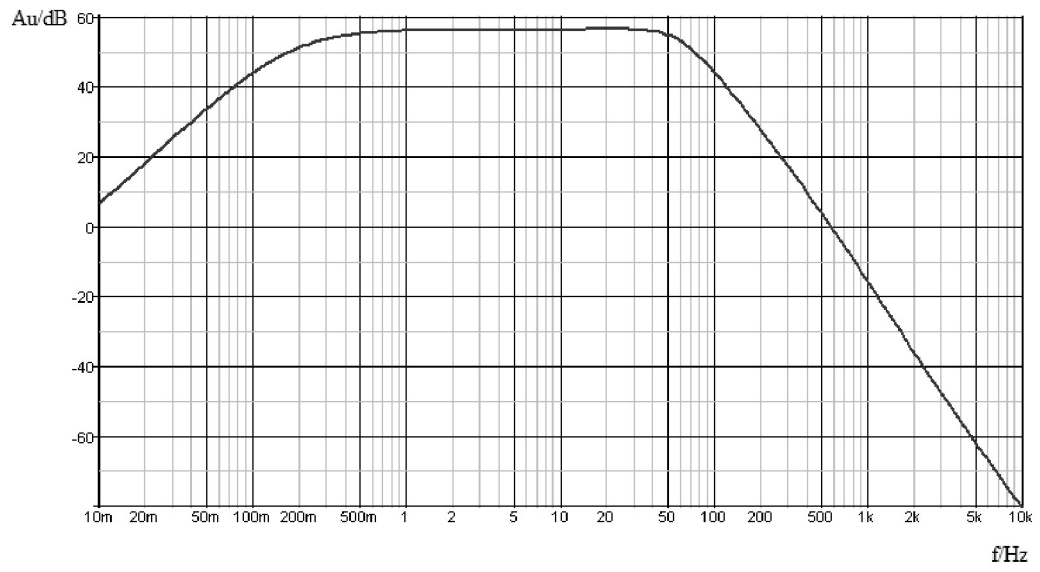


Obr.52. Fotografie síťového napájení pro testování zapojení

Příloha P8 Fotografie modulu UMS3



Obr.53. Fotografie modulu UMS3

Příloha P9 Frekvenční přenosová charakteristika filtrů modulu EEG zesilovače

Obr.54. Frekvenční přenosová charakteristika filtrů modulu EEG zesilovače

Příloha P10 Výpis zdrojového kódu programu Mexický příliv

```

unit MexicoZ;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Mask, MPlayer, ComCtrls;
type
  THlavniForm1 = class(TForm)
    PCZalozky: TPageControl;
    TSHlavni: TTabSheet;
    TSFFT: TTabSheet;
    TSOstatni: TTabSheet;
    GBZes: TGroupBox;
    LblKan1: TLabel;
    LblKan2: TLabel;
    LblKan3: TLabel;
    LblKan4: TLabel;
    LblKan5: TLabel;
    LblKan6: TLabel;
    BtnNastavZesileni: TButton;
    EdtZesilKan1: TMaskEdit;
    EdtZesilKan2: TMaskEdit;
    EdtZesilKan3: TMaskEdit;
    EdtZesilKan4: TMaskEdit;
    EdtZesilKan5: TMaskEdit;
    EdtZesilKan6: TMaskEdit;
    ImgGraf1: TImage;
    ImgFrekv1: TImage;
    ImgGraf2: TImage;
    ImgFrekv2: TImage;
    LBLPopis1: TLabel;
    LblPopG1: TLabel;
    LBLPopis2: TLabel;
    LBLPopis3: TLabel;
    LBLPopis4: TLabel;
    LBLPopis5: TLabel;
    LBLPopis6: TLabel;
    LblPopG16: TLabel;
    LblPopG5: TLabel;
    LblPopG4: TLabel;
    LblPopG3: TLabel;
    LblPopG2: TLabel;
    MediaPlayer1: TMediaPlayer;
    BtnNactiData: TButton;
    BtnKonec: TButton;
    GBGRKanal1: TGroupBox;
    CBkanalGraf1: TComboBox;
    GBGrRozsah1: TGroupBox;
    CBGrafRozsah1: TComboBox;
    GBGrDomFr1: TGroupBox;
    LblHz1: TLabel;
    LbluV1: TLabel;
    EdtDomFrekHarm1: TEdit;
    EdtDomFrekAmpli1: TEdit;
    GBGRKanal2: TGroupBox;
    CBkanalGraf2: TComboBox;
    GBGrRozsah2: TGroupBox;
    CBGrafRozsah2: TComboBox;
    GBGrDomFr2: TGroupBox;
    LblHz2: TLabel;
    LbluV2: TLabel;
    EdtDomFrekHarm2: TEdit;
    EdtDomFrekAmpli2: TEdit;
    OpenDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    GBAkKan: TGroupBox;
    CBAktivniKanal: TComboBox;
    GBNastavpuvod: TGroupBox;
    BtnNastavVychozHod: TButton;
    GBPracKan: TGroupBox;
    ChBPracKan1: TCheckBox;
    ChBPracKan2: TCheckBox;
    ChBPracKan3: TCheckBox;
    ChBPracKan4: TCheckBox;
  end;

```

```
ChBPracKan5: TCheckBox;
ChBPracKan6: TCheckBox;
GBDomFrekv: TGroupBox;
EdtDomFrekv1: TEdit;
EdtDomFrekv5: TEdit;
EdtDomFrekv2: TEdit;
EdtDomFrekv3: TEdit;
EdtDomFrekv4: TEdit;
EdtDomFrekv6: TEdit;
LblDomFr1: TLabel;
LblDomFr2: TLabel;
LblDomFr4: TLabel;
LblDomFr3: TLabel;
LblDomFr6: TLabel;
LblDomFr5: TLabel;
GBInfoHl: TGroupBox;
LblAktFrekvAVS: TLabel;
EdtAktFrekvAVS: TEdit;
Label1: TLabel;
EdtCasOdZac: TEdit;
LblPopHz: TLabel;
LblPopmin: TLabel;
StBStavRadek: TStatusBar;
GBSoZmen: TGroupBox;
GBNalezeneZarizeni: TGroupBox;
GBJmenoZar: TGroupBox;
EdtJmenoZarizeni: TEdit;
GBCisloZar: TGroupBox;
EdtCisloZarizeni: TEdit;
GBSerCisloZar: TGroupBox;
EdtSerCisloZarizeni: TEdit;
GBPocetD: TGroupBox;
CBPocetDat: TComboBox;
GBOkno: TGroupBox;
CBOkno: TComboBox;
GBFrekvRoz: TGroupBox;
LblPom4: TLabel;
GroupBox3: TGroupBox;
EdtRozMin: TMaskEdit;
EdtRozMax: TMaskEdit;
BtnNastavFrekvRozsah: TButton;
GBFiltr: TGroupBox;
Lb1: TLabel;
Lb2: TLabel;
Lb3: TLabel;
LblPom1: TLabel;
LblPom2: TLabel;
LblPom3: TLabel;
ChBFiltr: TCheckBox;
GroupBox1: TGroupBox;
EdtFiltrMin1: TMaskEdit;
EdtFiltrMin2: TMaskEdit;
EdtFiltrMin3: TMaskEdit;
EdtFiltrMax1: TMaskEdit;
EdtFiltrMax2: TMaskEdit;
EdtFiltrMax3: TMaskEdit;
BtnNastavFiltr: TButton;
MEdtSouUzivatelMaxima: TMaskEdit;
GBDeRe: TGroupBox;
CBDelkaRelax: TComboBox;
GBDoFr: TGroupBox;
CBCilovaFrekv: TComboBox;
GBFrBd: TGroupBox;
CBFrekvBdeStavu: TComboBox;
GBJeDol: TGroupBox;
CHBJemneDolad: TCheckBox;
LblNapetiZobrKan1: TLabel;
LblNapetiZK1_0: TLabel;
LblNapetiZK2_0: TLabel;
LblNapetiZobrKan2: TLabel;
GBFrekvPosl: TGroupBox;
CBPosledFrekv: TComboBox;
procedure BtnKonecClick(Sender: TObject);
//procedure BtnNactiDataClick(Sender: TObject);
procedure BtnKonecNacClick(Sender: TObject);
```

```

procedure CBAktivniKanalyChange(Sender: TObject);
procedure CBPocetDatChange(Sender: TObject);
procedure CBDelkaRelaxChange(Sender: TObject);
procedure FormPaint(Sender: TObject);
procedure CBkanalGraf1Change(Sender: TObject);
procedure CBkanalGraf2Change(Sender: TObject);
procedure CBOknoChange(Sender: TObject);
procedure ChBFiltrClick(Sender: TObject);
procedure BtnNastavFrekvRozsahClick(Sender: TObject);
procedure BtnNastavFiltrClick(Sender: TObject);
procedure BtnNastavZesileniClick(Sender: TObject);
procedure CBGrafRozsah1Change(Sender: TObject);
procedure CBGrafRozsah2Change(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure EdtRozMinChange(Sender: TObject);
procedure EdtRozMaxChange(Sender: TObject);
procedure EdtFiltrMin1Change(Sender: TObject);
procedure EdtFiltrMax1Change(Sender: TObject);
procedure EdtFiltrMin2Change(Sender: TObject);
procedure EdtFiltrMax2Change(Sender: TObject);
procedure EdtFiltrMin3Change(Sender: TObject);
procedure EdtFiltrMax3Change(Sender: TObject);
procedure EdtZesilKan1Change(Sender: TObject);
procedure EdtZesilKan2Change(Sender: TObject);
procedure EdtZesilKan3Change(Sender: TObject);
procedure EdtZesilKan4Change(Sender: TObject);
procedure EdtZesilKan5Change(Sender: TObject);
procedure EdtZesilKan6Change(Sender: TObject);
procedure BtnNastavVychozHodClick(Sender: TObject);
procedure BtnNactiDataClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure CBPosledFrekvChange(Sender: TObject);
private
  procedure NactiDataZFFT;
  procedure VykresliGraf(ktery:word);
  procedure VykresliFrekv(ktery:word);
  procedure ramecekgraf(ktery:word);
  procedure PoSpusteni;
  procedure NastavPrvniPosledniKan;
  procedure NacteniHodnotForm;
  procedure UlozeniHodnotForm;
  procedure ramecekfrekv(ktery:word);
  procedure NastavVychoziHodnoty;
  procedure NacitaniaAVS;
  procedure ImpulsNahoruAVS;
  procedure ImpulsDoluAVS;
  procedure PosliFrekv(SE1RetFrekv:string);
  function PrevedExtendStr81(HpomPrevodExtend:extended):string;
  function PrevedExtendStr82(HpomPrevodExtend:extended):string;
  function PrevedStrProGenerator81(HpomPrevodExtended:extended):string;
  procedure NastavEnableFalse;
  procedure NastavEnableTrue;
  procedure PosliKonecFFT;
  procedure prepocetDatFFT_cast1;
  procedure prepocetDatFFT_cast2;
  { Private declarations }
public
  { Public declarations }
end;
var
  HlavniForm1: THlavniForm1;

implementation
uses USB_UMS3;
const HRefNapetiAD=4;
      Hnx=1024;
      HMaximKan=6;
      SE1MsgId = 'Message_FrekvStrobe';
      SE1MappingId = 'MappingStrobe';
      SE2MsgId = 'Message_FFT';
      SE2MappingId = 'MappingFFT';

var
  HChybazarizeni,HPrizZmenaNastaveni,PrizFatalChyba,HPrizChybaDat,Hprizerror,

```

```

priztik1,prizchybycteni,HprizSpusteniAplAVS,HprizSpusteniAplikace,
HPrizNastaveniVychHodnot,HprizPocatNast,HprizPocatNastHod,Hpriztlkon,
HpriztlkonecNacitani,HPrizZmenaRozsahFrekv,HPrizZmenaFrekvFiltr,
HPrizZmenaZesileni,HPrizPrvniPrubeh,HprizNacitaniaAVS,
HPrizZmenaUzivatelMaxima:boolean;
SE1HMessage,SE2HMessage,SE1HMap,SE2HMap: THandle;
vystDokument,vstfrekvFFT,vstNastaveniForm, souHarmMaxim:textfile;
GrafZvetKan1,GrafZvetKan2,HposuvMaxim,vyberkanal1,vyberkanal2,VstHdatFFT,
VstHdatFFTPul,Hi,Hj,HdatFFT,hlaska,HsirkaHarm,Hk,HPocetVyhodFrekv,VstHAktKan,
HPocetAktKan,Hprvnikan,Hposledkan,VstHPrvniKan,VstHPosledniKan,prikon,
DelkaRelax,SE1Frekv:word;
HPomVstHMaxAmplZobrKan,VstHXKcoef,HStaryPomKcoef1,HStaryPomKcoef2,
HpomExtended:extended;
VstHmaxHarm:array[1..HMaximKan] of extended;
VstHmaxAmpl:array[1..HMaximKan] of extended;
poleMaxFrekvenci:array[1..HmaximKan,1..20] of extended;
VstHgrafKan:array[1..HmaximKan,1..Hnx] of extended;
VstHmaxHarmZobrKan:array[1..2] of extended;
VstHmaxAmplZobrKan:array[1..2] of extended;
VstHFrekvKan:array[1..HmaximKan,1..(Hnx div 2)-1] of extended;
HZesilKan:array[1..6] of word;
HHlaseniChyb,HZarNo,HZarSerNo,HZarJmeno,HIAktAdresStr,HStrDruhOkna,
HStrFrekvMin,HStrFrekvMax,HStrPrizFiltr,HStrFiMin1,HStrFiMin2,HStrFiMin3,
HStrFiMax1,HStrFiMax2,HStrFiMax3,HpomString,HStrvyberkan1,
HStrvyberkan2:string;

```

```
{SR *.dfm}
```

```

procedure THlavniForm1.FormPaint(Sender: TObject);
begin
  if HprizSpusteniAplikace=true then PoSpusteni;
end;

```

```

procedure THlavniForm1.PoSpusteni;
begin
  HprizSpusteniAplikace:=false; // priznak kvuli udalosti paint
  HlavniForm1.Refresh;
  HPrizZmenaRozsahFrekv:=false;
  HPrizZmenaFrekvFiltr:=false;
  HPrizZmenaZesileni:=false;
  HprizNacitaniaAVS:=false;
  Hpriztlkon:=false;
  HpriztlkonecNacitani:=false;
  HPrizNastaveniVychHodnot:=false;
  HPrizPrvniPrubeh:=true;
  HPrizZmenaNastaveni:=true;
  PrizFatalChyba:=false;
  HPrizChybaDat:=false;
  HPrizZmenaUzivatelMaxima:=false;
  HIAktAdresStr:= (ExtractFilePath(Application.ExeName));
  assignfile(vstFrekvFFT,HIAktAdresStr+'Pracovni\FFT_Vysledky.dat');
  assignfile(vstNastaveniForm,HIAktAdresStr+'Pracovni\Nastaveni_Form.naf');
  ramecekGraf(1);
  ramecekGraf(2);
  ramecekFrekv(1);
  ramecekFrekv(2);
  EdtDomFrekAmpli1.Text:=0;
  EdtDomFrekHarm1.Text:=0;
  EdtDomFrekAmpli2.Text:=0;
  EdtDomFrekHarm2.Text:=0;
  HprizPocatNastHod:=true;
  NacteniHodnotForm;
  HprizPocatNastHod:=false;
  HdatFFT:=strtoint(CBPocetDat.Text); {KONDAC = POCET DAT PRO FFT}
  vyberkanal1:=strtoint(CBkanalgraf1.text); {byte}
  vyberkanal2:=strtoint(CBkanalgraf2.text); {byte}
  HStrvyberkan1:=CBkanalgraf1.text;
  HStrvyberkan2:=CBkanalgraf2.text;
  NastavPrvniPosledniKan;
  HStrDruhOkna:=CBOkno.text;
  HStrFrekvMin:=trim(EdtRozMin.text);
  HStrFrekvMax:=trim(EdtRozMax.text);
  if ChBFiltr.Checked=true then HStrPrizFiltr:=Filtr1'
    else HStrPrizFiltr:=Filtr0';
  HStrFiMin1:=trim(EdtFiltrMin1.text);

```

```

HStrFiMin2:=trim(EdtFiltrMin2.text);
HStrFiMin3:=trim(EdtFiltrMin3.text);
HStrFiMax1:=trim(EdtFiltrMax1.text);
HStrFiMax2:=trim(EdtFiltrMax2.text);
HStrFiMax3:=trim(EdtFiltrMax3.text);
HZesilKan[1]:=strtoint(trim(EdtZesilKan1.text));
HZesilKan[2]:=strtoint(trim(EdtZesilKan2.text));
HZesilKan[3]:=strtoint(trim(EdtZesilKan3.text));
HZesilKan[4]:=strtoint(trim(EdtZesilKan4.text));
HZesilKan[5]:=strtoint(trim(EdtZesilKan5.text));
HZesilKan[6]:=strtoint(trim(EdtZesilKan6.text));
case CBGrafRozsah1.ItemIndex of
  0:GrafZvetKan1:=1;
  1:GrafZvetKan1:=2;
  2:GrafZvetKan1:=5;
  3:GrafZvetKan1:=10;
  4:GrafZvetKan1:=20;
  5:GrafZvetKan1:=50;
  6:GrafZvetKan1:=100;
  7:GrafZvetKan1:=200;
  8:GrafZvetKan1:=500;
  9:GrafZvetKan1:=1000;
end;
case CBGrafRozsah2.ItemIndex of
  0:GrafZvetKan2:=1;
  1:GrafZvetKan2:=2;
  2:GrafZvetKan2:=5;
  3:GrafZvetKan2:=10;
  4:GrafZvetKan2:=20;
  5:GrafZvetKan2:=50;
  6:GrafZvetKan2:=100;
  7:GrafZvetKan2:=200;
  8:GrafZvetKan2:=500;
  9:GrafZvetKan2:=1000;
end;
HPocetVyhodFrekv:=strtoint(CBPosledFrekv.Text);
for Hi:=1 to 6 do
  begin
    case Hi of
      1:if (HPrvniKan<=Hi) and (Hi<=HPosledKan)
        then ChBPracKan1.enabled:=true
        else ChBPracKan1.enabled:=false;
      2:if (HPrvniKan<=Hi) and (Hi<=HPosledKan)
        then ChBPracKan2.enabled:=true
        else ChBPracKan2.enabled:=false;
      3:if (HPrvniKan<=Hi) and (Hi<=HPosledKan)
        then ChBPracKan3.enabled:=true
        else ChBPracKan3.enabled:=false;
      4:if (HPrvniKan<=Hi) and (Hi<=HPosledKan)
        then ChBPracKan4.enabled:=true
        else ChBPracKan4.enabled:=false;
      5:if (HPrvniKan<=Hi) and (Hi<=HPosledKan)
        then ChBPracKan5.enabled:=true
        else ChBPracKan5.enabled:=false;
      6:if (HPrvniKan<=Hi) and (Hi<=HPosledKan)
        then ChBPracKan6.enabled:=true
        else ChBPracKan6.enabled:=false;
    end; //konec case
  end; // konec for
  LblNapetiZobrKan1.caption:=PrevedExtendStr81(((HRefNapetiAD/GrafZvetKan1)/
    HZesilKan[VyberKanal1])*1000000)+' uV';
  LblNapetiZobrKan2.caption:=PrevedExtendStr81(((HRefNapetiAD/GrafZvetKan2)/
    HZesilKan[VyberKanal2])*1000000)+' uV';
  for Hi:=1 to HMaximKan do VstHmaxHarm[Hi]:=0;
  for Hi:=1 to HmaximKan do VstHmaxAmpl[Hi]:=0;
  for Hi:=1 to HmaximKan do begin
    for Hj:=1 to Hnx do VstHgrafKan[Hi,Hj]:=0;
    end;
  for Hi:=1 to 2 do VstHmaxHarmZobrKan[Hi]:=0;
  for Hi:=1 to 2 do VstHmaxAmplZobrKan[Hi]:=0;
  for Hi:=1 to HmaximKan do begin
    for Hj:=1 to ((Hnx div 2)-1) do VstHFrekvKan[Hi,Hj]:=0;
    end;
  HChybarizeni:=false;
  OtevriPort(HHlaseniChyb,HChybarizeni,HZarNo,HZarSerNo,HZarJmeno);

```

```

if HChybazarizeni=true then showmessage(HHlaseniChyb)
else begin
    EdtCisloZarizeni.Text:=HzarNo;
    EdtJmenoZarizeni.Text:=HzarJmeno;
    EdtSerCisloZarizeni.text:=HZarSerNo;
    StBStavRadek.Panels[0].text:=HHlaseniChyb;
end;
if HChybazarizeni=false then NastavPrenosPortu(HHlaseniChyb,HChybazarizeni);
if HChybazarizeni=true then showmessage(HHlaseniChyb)
else StBStavRadek.Panels[0].text:=HHlaseniChyb;
if HChybazarizeni=true
then
begin
showmessage('Nastal problém komunikace s periférním zařízením.
Prosím přesvědčte se, že je správně zapojeno a znovu spusťte program');
ZavriPort;
application.terminate;
end
else
begin
Hlaska := Winexec (PChar(HIAktAdresStr+'FFTnew.exe'), sw_showNormal);
VymazVstupniZasobnik;
ResetujFTDI;
repeat
StBStavRadek.Panels[0].text:='Nacitaji se data';
NactiDataDoSouboru(HStrFrekvMin,HStrFrekvMax,HstrPrizFiltr,
HStrFiMin1,HStrFiMin2,HStrFiMin3,HStrFiMax1,HStrFiMax2,HStrFiMax3,
HStrDruhOkna,HDatFFT,HPrvniKan,HPosledkan,HHlaseniChyb,
HChybazarizeni);
StBStavRadek.Panels[0].text:=HHlaseniChyb;
if HChybazarizeni=true
then
begin showmessage('Chyba zarizeni pri cteni');
Hpriztlkon:=true;
end
else
begin
HPrizChybaDat:=false;
StBStavRadek.Panels[0].text:='Načitají se data z FFT';
NactiDataZFFT;
StBStavRadek.Panels[0].text:='Kreslí se grafy';
if HPrizChybaDat=false
then
begin
begin
prepocetDatFFT_cast1;
EdtDomFrekHarm1.Text:=PrevedExtendStr81
(VstHMaxHarmZobrKan[1]);
EdtDomFrekHarm2.Text:=PrevedExtendStr81
(VstHMaxHarmZobrKan[2]);
HPomVstHMaxAmplZobrKan:=((4*VstHMaxAmplZobrKan[1]*
(HRefNapetiAD/1024))/HZesilKan[vyberkanal1])*1000000;
EdtDomFrekAmpli1.Text:=PrevedExtendStr81
(HPomVstHMaxAmplZobrKan);
HPomVstHMaxAmplZobrKan:=((4*VstHMaxAmplZobrKan[2]*
(HRefNapetiAD/1024))/HZesilKan[vyberkanal2])*1000000;
EdtDomFrekAmpli2.Text:=PrevedExtendStr81
(HPomVstHMaxAmplZobrKan);
ramecekGraf(1);
VykresliGraf(1);
ramecekGraf(2);
VykresliGraf(2);
ramecekFrekv(1);
VykresliFrekv(1);
ramecekFrekv(2);
VykresliFrekv(2);
end;
if HPrizNastaveniVychHodnot=true then NastavVychoziHodnoty;
if HprizNacitaniaAVS=true then NacitaniaAVS;
end; // konec else
application.processmessages;
until Hpriztlkon=true;
end;
UlozeniHodnotForm;
PosliKonecFFT;
ZavriPort;

```

```

application.Terminate;
end;

procedure THlavniForm1.BtnNactiDataClick(Sender: TObject);
begin
if HprizSpusteniAplAVS=false
then
begin
begin
if (HPrizZmenaRozsahFrekv=true) or (HPrizZmenaFrekvFiltr=true) or
(HPrizZmenaZesileni=true)
then
begin
begin
if HPrizZmenaRozsahFrekv=true then showmessage('Změnili jste hodnoty
pro frekvenční rozsah, ale nenastavili jste je, prosím použijte
tlačítko Nastav pod těmito hodnotami');
if HPrizZmenaFrekvFiltr=true then showmessage('Změnili jste hodnoty
pro frekvenční filtry, ale nenastavili jste je, prosím použijte
tlačítko Nastav pod těmito hodnotami');
if HPrizZmenaZesileni=true then showmessage('Změnili jste hodnoty
pro zesílení EEG zesilovačů, ale nenastavili jste je, prosím
použijte tlačítko Nastav pod těmito hodnotami');
end
else HprizNacitaniaAVS:=true;
end
else HpriztlkonecNacitani:=true;
end;

procedure THlavniForm1.NacitaniaAVS;
var
HprizChybaDokum,HpriKonecGeneratoru,HprizChybaMaxim:boolean;
indexPosuvuCasu,Hrok,HMesic,Hden,HHodina,HMinuta,HSec,HMsec:word;
akttik1,akttik2,akttikPoc,akttikOdZac,akttikZmenaZvuk:longword;
aktfrefkvAVS,pomAktfrefkvAVS,domFFTdown,domFTTU,ikonv,FrekvBdeStavu,
frefkvdomfft,cilovafrekv:extended;
HretfrefkvAvs,HretStaraFrefkvAvs,HretDokument,Hretvst:string;
begin
HlavniForm1.Caption:='Mexický příliv 024 - Spuštěna práce s AVS.';
HprizSpusteniAplAVS:=true;
HpriztlkonecNacitani:=false;
HprizChybaDokum:=false;
HPrizChybaDat:=false;
HChybazarizeni:=false;
HprizNacitaniaAVS:=false;
indexPosuvuCasu:=0;
HprizChybaMaxim:=false;
HpriKonecGeneratoru:=false;    ///
HretStaraFrefkvAvs:='30,0';
akttikZmenaZvuk:=(GetTickCount);
NastavEnableFalse;
BtnNactiData.Caption:='Konec práce s AVS';
HposuvMaxim:=1;
for Hi:=1 to HmaximKan do
begin
for Hj:=1 to 20 do poleMaxFrekvenci[Hi,Hj]:=0;
end;
assignfile(souHarmMaxim,HIAktAdresStr+'Pracovni'+
trim(MEdtSouUzivatelMaxima.text)+'.dat');
Application.ProcessMessages;
//nastaveni souboru s vysledky
decodedate(date,Hrok,HMesic,Hden);
decodeTime(time,HHodina,HMinuta,HSec,HMsec); //word
HretDokument:='Dokument_'+inttostr(Hrok);
if Hmesic<10 then HretDokument:=HretDokument+'0';
HretDokument:=HretDokument+inttostr(Hmesic);
if HDen<10 then HretDokument:=HretDokument+'0';
HretDokument:=HretDokument+inttostr(HDen)+'_';
if HHodina<10 then HretDokument:=HretDokument+'0';
HretDokument:=HretDokument+inttostr(HHodina);
if HMinuta<10 then HretDokument:=HretDokument+'0';
HretDokument:=HretDokument+inttostr(HMinuta);
assignfile(vystDokument,HIAktAdresStr+'Vysledky'+HretDokument+'.csv');
StBStavRadek.Panels[0].text:='Plnění proměnných';
FrefkvBdeStavu:=strtoint(CBFrefkvBdeStavu.text);
cilovafrekv:=strtoint(CBCilovaFrefkv.text);
aktfrefkvAVS:=30;    /// =30 Hz

```



```

DelkaRelax:=strtoint(CBDelkaRelax.Text);
StBStavRadek.Panels[0].text:='Nastavení výst. dokumentu';
try
rewrite(vystDokument);
  try
    write(vystDokument,'čas[s:]);
    for Hi:=1 to 2 do write(vystDokument,('fMaxZobrKan'+
      inttostr(Hi)+'[Hz:]);
    for Hi:=Hprvnikan to Hposledkan do write(vystDokument,('fK'+
      inttostr(Hi)+'[Hz:]);
    write(vystDokument,'fDom[Hz:]);
    write(vystDokument,'fAVS[Hz:]);
    writeln(vystDokument);
  finally
end; // finally
except
  on e:EInOutError do begin showmessage('Chyba zapisu do souboru
    Dokument');
    HprizChybaDokum:=true;
  end;
end; // except
Application.ProcessMessages;
akttik1:=(GetTickCount);
repeat
  akttik2:=(GetTickCount);
  StBStavRadek.Panels[0].text:='Čekejte prosím '+PrevedExtendStr81((10000-
    (akttik2-akttik1)/1000))+ ' s';
  Application.ProcessMessages;
  Application.ProcessMessages;
until (akttik2-akttik1)>10000; // cekani 10s
if (HprizChybaDokum=false) or (HprizChybaMaxim=false) then
begin
  Hlaska := Winexec (PChar(HIAktAdresStr+'Generator.exe'), sw_showNormal);
  akttikPoc:=(GetTickCount);
  repeat //globalni repeat
    akttik1:=(GetTickCount);
    akttikOdZac:=(akttik1-akttikpoc);
    EdtCasOdZac.Text:=PrevedExtendStr81((akttikOdZac/60000));
    StBStavRadek.Panels[0].text:='Načítají se data';
    NactiDataDoSouboru(HStrFrekvMin,HStrFrekvMax,HStrPrizFiltr,HStrFiMin1,
      HStrFiMin2,HStrFiMin3,HStrFiMax1,HStrFiMax2,HStrFiMax3,HStrDruhOkna,
      HDatFFT,HPrvniKan,HPosledkan,HHlaseniChyb,HChybarizeni);
    StBStavRadek.Panels[0].text:=HHlaseniChyb;
    if HChybarizeni=true
    then
    begin
      if HpriKonecGeneratoru=false
      then
      begin
        HpriKonecGeneratoru:=true;
        PosliFrekv('Konec');
      end;
      showmessage('Chyba zařízení USB-UMS3 při čtení');
    end
    else
    begin
      HPrizChybaDat:=false;
      StBStavRadek.Panels[0].text:='Načítají se data z FFT';
      NactiDataZFFT;
      prepocetDatFFT_cast1;
      if HpriKonecGeneratoru=false then prepocetDatFFT_cast2;
      StBStavRadek.Panels[0].text:='Kresli se grafy';
      if HPrizChybaDat=false
      then
      begin
        EdtDomFrekHarm1.Text:=PrevedExtendStr81(VstHMaxHarmZobrKan[1]);
        EdtDomFrekHarm2.Text:=PrevedExtendStr81(VstHMaxHarmZobrKan[2]);
        HPomVstHMaxAmplZobrKan:=((4*VstHMaxAmplZobrKan[1]*
          (HRefNapetiAD/1024))/HZesilKan[vyberkanal1])*1000000;
        EdtDomFrekAmpli1.Text:=PrevedExtendStr81(HPomVstHMaxAmplZobrKan);
        HPomVstHMaxAmplZobrKan:=((4*VstHMaxAmplZobrKan[2]*
          (HRefNapetiAD/1024))/HZesilKan[vyberkanal2])*1000000;
        for Hi:=Hprvnikan to Hposledkan do
        begin
          case Hi of

```

```

1:if ChBPracKan1.checked=true
  then EdtDomFrekv1.Text:=PrevedExtendStr81(VstHMaxHarm[Hi])
  else EdtDomFrekv1.Text:='0';
2:if ChBPracKan2.checked=true
  then EdtDomFrekv2.Text:=PrevedExtendStr81(VstHMaxHarm[Hi])
  else EdtDomFrekv2.Text:='0';
3:if ChBPracKan3.checked=true
  then EdtDomFrekv3.Text:=PrevedExtendStr81(VstHMaxHarm[Hi])
  else EdtDomFrekv3.Text:='0';
4:if ChBPracKan4.checked=true
  then EdtDomFrekv4.Text:=PrevedExtendStr81(VstHMaxHarm[Hi])
  else EdtDomFrekv4.Text:='0';
5:if ChBPracKan5.checked=true
  then EdtDomFrekv5.Text:=PrevedExtendStr81(VstHMaxHarm[Hi])
  else EdtDomFrekv5.Text:='0';
6:if ChBPracKan6.checked=true
  then EdtDomFrekv6.Text:=PrevedExtendStr81(VstHMaxHarm[Hi])
  else EdtDomFrekv6.Text:='0';
end;
end;
ramecekGraf(1);
VykresliGraf(1);
ramecekGraf(2);
VykresliGraf(2);
ramecekFrekv(1);
VykresliFrekv(1);
ramecekFrekv(2);
VykresliFrekv(2);
if (akttikOdZac/60000)>DelkaRelax
then
begin
if (akttikOdZac/60000)>(DelkaRelax+(0.25*indexPosuvuCasu))
then // [min]
begin
indexPosuvuCasu:=indexPosuvuCasu+1;
StBStavRadek.Panels[0].text:='Probíhá návrat k bdělé hladině
vědomí';
if FrekvBdeStavu=AktFrekvAVS
then
begin
if HpriKonecGeneratoru=false then
begin
HpriKonecGeneratoru:=true;
PosliFrekv('Konec');
akttik1:=(GetTickCount);
repeat
akttik2:=(GetTickCount);
StBStavRadek.Panels[0].text:='Čekajte prosím '+
(PrevedExtendStr81((15000-(akttik2-akttik1))/1000))+
's';
Application.ProcessMessages;
until (akttik2-akttik1)>15000;
end;

mediaplayer1.DeviceType:=dtautoselect;
mediaplayer1.filename:=(ExtractFilePath
(Application.ExeName))+'\Zvuky\Gong.wav';
mediaplayer1.open;
mediaplayer1.play;
end;
if FrekvBdeStavu>AktFrekvAVS
then
begin
AktFrekvAVS:=AktFrekvAVS+1;
HretFrekvAvs:=PrevedExtendStr81(aktfrekvAVS);
EdtAktFrekvAVS.Text:=HretFrekvAvs;
PosliFrekv(PrevedStrProGenerator81(aktfrekvAVS));
end;

if FrekvBdeStavu<AktFrekvAVS
then
begin
AktFrekvAVS:=AktFrekvAVS-1;
HretFrekvAvs:=PrevedExtendStr81(aktfrekvAVS);
EdtAktFrekvAVS.Text:=HretFrekvAvs;

```

```

        PosliFrekv(PrevedStrProGenerator81(aktfrekvAVS));
    end;
end;
end //// konec THEN posuvu k bdeni pri prekročení času relaxace
else
begin
StBStavRadek.Panels[0].text:='Výpočet frekvence pro AVS.';
frekvdomFFT:=0;
for Hi:=(Hprvnikan) to (Hposledkan) do
begin
case Hi of
1:if ChBPracKan1.checked=true then
begin
for Hj:=1 to HPocetVyhodFrekv do
if poleMaxFrekvenci[Hi,Hj]>frekvdomFFT then
frekvdomFFT:=poleMaxFrekvenci[Hi,Hj];
end;
2:if ChBPracKan2.checked=true then
begin
for Hj:=1 to HPocetVyhodFrekv do
if poleMaxFrekvenci[Hi,Hj]>frekvdomFFT then
frekvdomFFT:=poleMaxFrekvenci[Hi,Hj];
end;
3:if ChBPracKan3.checked=true then
begin
for Hj:=1 to HPocetVyhodFrekv do
if poleMaxFrekvenci[Hi,Hj]>frekvdomFFT then
frekvdomFFT:=poleMaxFrekvenci[Hi,Hj];
end;
4:if ChBPracKan4.checked=true then
begin
for Hj:=1 to HPocetVyhodFrekv do
if poleMaxFrekvenci[Hi,Hj]>frekvdomFFT then
frekvdomFFT:=poleMaxFrekvenci[Hi,Hj];
end;
5:if ChBPracKan5.checked=true then
begin
for Hj:=1 to HPocetVyhodFrekv do
if poleMaxFrekvenci[Hi,Hj]>frekvdomFFT then
frekvdomFFT:=poleMaxFrekvenci[Hi,Hj];
end;
6:if ChBPracKan6.checked=true then
begin
for Hj:=1 to HPocetVyhodFrekv do
if poleMaxFrekvenci[Hi,Hj]>frekvdomFFT then
frekvdomFFT:=poleMaxFrekvenci[Hi,Hj];
end;
end; // konec case
end; // konec for
if cilovafrekv<frekvdomfft
then
begin
if (0<=frekvdomfft) and (frekvdomfft<31)
then
begin
Ikonv:=30+1; //20+1
repeat
Ikonv:=Ikonv-1;
until Ikonv<=FrekvDomFFT;
if CHBJemneDolad.Checked=true
then
begin
if FrekvDomFFT-Ikonv>0.5 then AktFrekvAVS:=Ikonv
else AktFrekvAVS:=Ikonv-1;
end
else AktFrekvAVS:=Ikonv-1;
if cilovafrekv>AktFrekvAVS then AktFrekvAVS:=Ikonv;
if AktFrekvAVS<1 then AktFrekvAVS:=1;
end;
if 31<=frekvdomfft then AktFrekvAVS:=30;
end;
if cilovafrekv>frekvdomfft
then
begin
if (0<=frekvdomfft) and (frekvdomfft<=30)

```

```

then
begin
Ikonv:=0;
repeat
Ikonv:=Ikonv+1;
until Ikonv>=FrekvDomFFT;
if CHBJemneDolad.Checked=true
then
begin
if Ikonv-FrekvDomFFT>0.5 then AktFrekvAVS:=Ikonv
else AktFrekvAVS:=Ikonv+1;
end
else AktFrekvAVS:=Ikonv+1;
if cilovafrekv<AktFrekvAVS then AktFrekvAVS:=Ikonv;
if AktFrekvAVS>30 then AktFrekvAVS:=30;
end;
if 30<frekvdomfft then AktFrekvAVS:=30;
end;
if cilovafrekv=frekvdomfft then AktFrekvAVS:= cilovafrekv;
HretFrekvAvs:=PrevedExtendStr81(aktfrekvAVS);
EdtAktFrekvAVS.Text:=HretFrekvAvs;
if HretFrekvAvs<>HretStaraFrekvAvs
then
begin
PosliFrekv(PrevedStrProGenerator81(aktfrekvAVS));
HretStaraFrekvAvs:=HretFrekvAvs;
end;
akttik1:=(GetTickCount);
StBStavRadek.Panels[0].text:='Zápis dat do souboru dokument.';
write(vystDokument,(PrevedExtendStr82(akttikOdZac/60000))+'););
for Hi:=1 to 2 do write(vystDokument,
(PrevedExtendStr81(VstHMaxHarmZobrKan[Hi]))+'););
for Hi:=Hprvnikan to Hposledkan do
begin
case Hi of
1:if ChBPracKan1.checked=true
then write(vystDokument,
(PrevedExtendStr81(VstHMaxHarm[1]))+'););
else write(vystDokument,'0;');
2:if ChBPracKan2.checked=true
then write(vystDokument,
(PrevedExtendStr81(VstHMaxHarm[2]))+'););
else write(vystDokument,'0;');
3:if ChBPracKan3.checked=true
then write(vystDokument,
(PrevedExtendStr81(VstHMaxHarm[3]))+'););
else write(vystDokument,'0;');
4:if ChBPracKan4.checked=true
then write(vystDokument,
(PrevedExtendStr81(VstHMaxHarm[4]))+'););
else write(vystDokument,'0;');
5:if ChBPracKan5.checked=true
then write(vystDokument,
(PrevedExtendStr81(VstHMaxHarm[5]))+'););
else write(vystDokument,'0;');
6:if ChBPracKan6.checked=true
then write(vystDokument,
(PrevedExtendStr81(VstHMaxHarm[6]))+'););
else write(vystDokument,'0;');
end;
end;
write(vystDokument,(PrevedExtendStr81(frekvdomfft))+'););
write(vystDokument,(PrevedExtendStr81(aktfrekvAVS))+'););
writeln(vystDokument);
end; // else relax doba
end; //KONEC PRO HPrizChybaDat
end; //KONEC PRO HChybaZarizeni
Application.ProcessMessages;
until (HpriztlkonecNacitani=true) or (HChybaZarizeni=true)
or (HpriztChybaMaxim=true);
end; //KONEC PRO prizChybaDokum
closefile(vystDokument);
aktfrekvAVS:=0;
EdtAktFrekvAVS.Text:=PrevedExtendStr81(aktfrekvAVS);
EdtCasOdZac.Text:='0,0';

```

```

EdtDomFrekv1.text:='0';
EdtDomFrekv2.text:='0';
EdtDomFrekv3.text:='0';
EdtDomFrekv4.text:='0';
EdtDomFrekv5.text:='0';
EdtDomFrekv6.text:='0';
if HpriKonecGeneratoru=false
then
begin
HpriKonecGeneratoru:=true;
PosliFrekv('Konec'); /// odeslani Konce na generator
end;
HpriztlkonecNacitani:=false;
HprizSpusteniAplAVS:=false;
NastavEnableTrue;
BtnNactiData.Caption:='Start práce s AVS';
StBStavRadek.Panels[0].text:='';
HlavniForm1.Caption:='Mexický příliv 024 - kompatibilní s FFTnew 014 a
Generator 02 a vyšší';
end;

procedure THlavniForm1.PosliFrekv(SE1RetFrekv:string);
var
SE1P: PChar;
begin
if SE1HMap <> 0 then
begin
SE1P := MapViewOfFile(SE1HMap, FILE_MAP_WRITE,
0, 0, Length(SE1RetFrekv) + 1);
StrCopy(SE1P, PChar(SE1RetFrekv));
UnmapViewOfFile(SE1P);
SendMessage(HWND_BROADCAST, SE1HMessage, 0, 0);
end;
end;

Procedure THlavniForm1.VykresliFrekv(ktery:word);
var
HnasKoef,HpomerAmpli,HPomKoef:extended;
begin
HlavniForm1.Canvas.pen.Mode:=pmcopy;
HlavniForm1.Canvas.pen.style:=psSolid;
HlavniForm1.Canvas.Pen.Color:= clwhite;
HlavniForm1.Canvas.pen.width:=1;
case ktery of
1: begin
Hnaskoef:= HlavniForm1.ImgFrekv1.width/60;
HPomKoef:=VstHxkoef*Hnaskoef;
for Hi:=1 to (VstHdatFFTPul-1) do
begin
if VstHMaxAmplZobrKan[1]=0 then HpomerAmpli:=1
else HpomerAmpli:=1-
(VstHFrekvKan[vyberkanal1,Hi]/
VstHMaxAmplZobrKan[1]);
HlavniForm1.Canvas.moveto(HlavniForm1.ImgFrekv1.left+
round(Hi*HPomKoef),HlavniForm1.ImgFrekv1.top+
HlavniForm1.ImgFrekv1.height ); {x,y,dword}
HlavniForm1.Canvas.lineto(HlavniForm1.ImgFrekv1.left+
round(Hi*HPomKoef),HlavniForm1.ImgFrekv1.top+
round(HpomerAmpli*HlavniForm1.ImgFrekv1.height ));
end;
end;
2: begin
Hnaskoef:= HlavniForm1.ImgFrekv2.width/60;
HPomKoef:=VstHxkoef*Hnaskoef;
for Hi:=1 to (VstHdatFFTPul-1) do
begin
if VstHMaxAmplZobrKan[2]=0 then HpomerAmpli:=1
else HpomerAmpli:=1-
(VstHFrekvKan[vyberkanal2,Hi]/
VstHMaxAmplZobrKan[2]);
HlavniForm1.Canvas.moveto(HlavniForm1.ImgFrekv2.left+
round(Hi*HPomKoef),HlavniForm1.ImgFrekv2.top+
HlavniForm1.ImgFrekv2.height ); {x,y,dword}
HlavniForm1.Canvas.lineto(HlavniForm1.ImgFrekv2.left+
round(Hi*HPomKoef),HlavniForm1.ImgFrekv2.top+

```

```

        round(HpomerAmpli*HlavniForm1.ImgFrekv2.height) );
    end;
end;
end; // konec case
end;

procedure THlavniForm1.VykresliGraf(ktery:word);
var
    HpomPocatekGrpole:word;
    HPomNasY,HPolOsY,HStredGrY,HPomNasX,HVysly:extended;
begin
    HpomPocatekGrpole:=(VstHdatFFT div 2)-64;
    HlavniForm1.Canvas.pen.Mode:=pmcopy; {pmlblack, pmwhite}
    HlavniForm1.Canvas.pen.style:=psSolid; {plna cara}
    HlavniForm1.Canvas.Pen.Color:= clwhite;
    HlavniForm1.Canvas.pen.width:=1; {sirka cary v pixelech}
    case ktery of
    1:begin
        HPomNasY:=(HlavniForm1.Imggraf1.height/1024);
        HPolOsY:=HlavniForm1.Imggraf1.height/2;
        HStredGrY:=HlavniForm1.Imggraf1.top+HPolOsY;
        HPomNasX:=HlavniForm1.Imggraf1.width/VstHdatFFT;
        HVysly:=VstHGrafKan[vyberkanal1,1]-512;
        Hvysly:=Hvysly*HPomNasY * GrafZvetKan1;
        if Hvysly>HPolOsY then Hvysly:=HPolOsY;
        if Hvysly<((-1)*HPolOsY) then Hvysly:=((-1)*HPolOsY);
        HlavniForm1.Canvas.moveto (HlavniForm1.Imggraf1.left,
            round(HStredGrY-HVysly));
        for Hi:=2 to VstHdatFFT do
            begin
                HVysly:=VstHGrafKan[vyberkanal1,Hi]-512;
                Hvysly:=Hvysly*HPomNasY * GrafZvetKan1;
                if Hvysly>HPolOsY then Hvysly:=HPolOsY;
                if Hvysly<((-1)*HPolOsY) then Hvysly:=((-1)*HPolOsY);
                HlavniForm1.Canvas.lineto(HlavniForm1.Imggraf1.left+
                    round(HPomNasX*Hi),round(HStredGrY-HVysly));
            end;
        LblNapetiZobrKan1.caption:=PrevedExtendStr81(((HRefNapetiAD/
            GrafZvetKan1)/HZesilKan[vyberkanal1])*1000000)+' uV';
        end;
    2:begin
        HPomNasY:=(HlavniForm1.Imggraf2.height/1024);
        HPolOsY:=HlavniForm1.Imggraf2.height/2;
        HStredGrY:=HlavniForm1.Imggraf2.top+HPolOsY;
        HPomNasX:=HlavniForm1.Imggraf2.width/VstHdatFFT;
        HVysly:=VstHGrafKan[vyberkanal2,1]-512;
        Hvysly:=Hvysly*HPomNasY * GrafZvetKan2;
        if Hvysly>HPolOsY then Hvysly:=HPolOsY;
        if Hvysly<((-1)*HPolOsY) then Hvysly:=((-1)*HPolOsY);
        HlavniForm1.Canvas.moveto (HlavniForm1.Imggraf2.left,
            round(HStredGrY-HVysly));
        for Hi:=2 to VstHdatFFT do
            begin
                HVysly:=VstHGrafKan[vyberkanal2,Hi]-512;
                Hvysly:=Hvysly*HPomNasY * GrafZvetKan2;
                if Hvysly>HPolOsY then Hvysly:=HPolOsY;
                if Hvysly<((-1)*HPolOsY) then Hvysly:=((-1)*HPolOsY);
                HlavniForm1.Canvas.lineto(HlavniForm1.Imggraf2.left+
                    round(HPomNasX*Hi),round(HStredGrY-HVysly));
            end;
        LblNapetiZobrKan2.caption:=PrevedExtendStr81(((HRefNapetiAD/
            GrafZvetKan2)/HZesilKan[vyberkanal2])*1000000)+' uV';
        end;
    end; // konec case
end;

procedure THlavniForm1.BtnKonecClick(Sender: TObject);
begin
    if (HPrizZmenaRozsahFrekv=true) or (HPrizZmenaFrekvFiltr=true) or
        (HPrizZmenaZesileni=true)
    then
        begin
            if HPrizZmenaRozsahFrekv=true then showmessage('Změnili jste hodnoty
                pro frekvenční rozsah, ale nenastavili jste je, prosím použijte tlačítko
                Nastav pod těmito hodnotami');
        end;
    end;
end;

```

```
if HPrizZmenaFrekvFiltr=true then showmessage('Změnili jste hodnoty
pro frekvenční filtry, ale nenastavili jste je, prosím použijte tlačítko
Nastav pod těmito hodnotami');
if HPrizZmenaZesileni=true then showmessage('Změnili jste hodnoty
pro zesílení EEG zesilovačů, ale nenastavili jste je, prosím použijte
tlačítko Nastav pod těmito hodnotami');
end
else Hpriztlkon:=true;
end;
```

```
procedure THlavniForm1.BtnKonecNacClick(Sender: TObject);
begin
  HpriztlkonecNacitani:=true;
end;
```

```
procedure THlavniForm1.NastavPrvniPosledniKan;
begin
```

```
  case CBAktivniKanaly.ItemIndex of
```

```
    0: begin Hprvnikan:=1;
           Hposledkan:=2;
           HPocetAktKan:=2;
        end;
```

```
    1: begin Hprvnikan:=1;
           Hposledkan:=4;
           HPocetAktKan:=4;
        end;
```

```
    2: begin Hprvnikan:=1;
           Hposledkan:=6;
           HPocetAktKan:=6;
        end;
```

```
    3: begin Hprvnikan:=3;
           Hposledkan:=4;
           HPocetAktKan:=2;
        end;
```

```
    4: begin Hprvnikan:=5;
           Hposledkan:=6;
           HPocetAktKan:=2;
        end;
```

```
    5: begin Hprvnikan:=1;
           Hposledkan:=1;
           HPocetAktKan:=1;
        end;
```

```
    6: begin Hprvnikan:=2;
           Hposledkan:=2;
           HPocetAktKan:=1;
        end;
```

```
    7: begin Hprvnikan:=3;
           Hposledkan:=3;
           HPocetAktKan:=1;
        end;
```

```
    8: begin Hprvnikan:=4;
           Hposledkan:=4;
           HPocetAktKan:=1;
        end;
```

```
    9: begin Hprvnikan:=5;
           Hposledkan:=5;
           HPocetAktKan:=1;
        end;
```

```
   10: begin Hprvnikan:=6;
           Hposledkan:=6;
           HPocetAktKan:=1;
        end;
```

```
  end; // konec case
end;
```

```
procedure THlavniForm1.CBkanalGraf1Change(Sender: TObject);
begin
```

```
  ramecekGraf(1);
```

```
  ramecekFrekv(1);
```

```
  EdtDomFrekAmpli1.Text:='0';
```

```
  EdtDomFrekHarm1.Text:='0';
```

```
  if (CBkanalGraf1.ItemIndex>HposledKan-1)
```

```
  then CBkanalGraf1.ItemIndex:=HposledKan-1;
```

```
  if (CBkanalGraf1.ItemIndex<HPrvniKan-1)
```

```
  then CBkanalGraf1.ItemIndex:=HPrvniKan-1;
```

```

vyberkanal1:=CBkanalGraf1.ItemIndex+1;      //zobr kanal 1
HStrvyberkan1:=CBkanalgraf1.text;
HPrizZmenaNastaveni:=true;
end;

procedure THlavniForm1.CBkanalGraf2Change(Sender: TObject);
begin
ramecekGraf(2);
ramecekFrekv(2);
EdtDomFrekAmpli2.Text:='0';
EdtDomFrekHarm2.Text:='0';
if (CBkanalGraf2.ItemIndex>HposledKan-1)
then CBkanalGraf2.ItemIndex:=HposledKan-1;
if (CBkanalGraf2.ItemIndex<HPrvniKan-1)
then CBkanalGraf2.ItemIndex:=HPrvniKan-1;
vyberkanal2:=CBkanalGraf2.ItemIndex+1;      //zobr kanal 2
HStrvyberkan2:=CBkanalgraf2.text;
HPrizZmenaNastaveni:=true;
end;

procedure THlavniForm1.CBAktivniKanalyChange(Sender: TObject);
begin
HPrizZmenaNastaveni:=true;
ramecekGraf(1);
ramecekGraf(2);
ramecekFrekv(1);
ramecekFrekv(2);
EdtDomFrekAmpli1.Text:='0';
EdtDomFrekHarm1.Text:='0';
EdtDomFrekAmpli2.Text:='0';
EdtDomFrekHarm2.Text:='0';
EdtDomFrekv1.Text:='0';
EdtDomFrekv2.Text:='0';
EdtDomFrekv3.Text:='0';
EdtDomFrekv4.Text:='0';
EdtDomFrekv5.Text:='0';
EdtDomFrekv6.Text:='0';
NastavPrvniPosledniKan;
if (CBkanalGraf1.ItemIndex>HposledKan-1)
then CBkanalGraf1.ItemIndex:=HposledKan-1;
if (CBkanalGraf1.ItemIndex<HPrvniKan-1)
then CBkanalGraf1.ItemIndex:=HPrvniKan-1;
vyberkanal1:=CBkanalGraf1.ItemIndex+1;      //zobr kanal 1
HStrvyberkan1:=CBkanalgraf1.text;
if (CBkanalGraf2.ItemIndex>HposledKan-1)
then CBkanalGraf2.ItemIndex:=HposledKan-1;
if (CBkanalGraf2.ItemIndex<HPrvniKan-1)
then CBkanalGraf2.ItemIndex:=HPrvniKan-1;
vyberkanal2:=CBkanalGraf2.ItemIndex+1;      //zobr kanal 2
HStrvyberkan2:=CBkanalgraf2.text;
for Hi:=1 to 6 do
begin
case Hi of
1:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
ChBPracKan1.enabled:=true;
ChBPracKan1.checked:=true;
end
else begin
ChBPracKan1.checked:=false;
ChBPracKan1.enabled:=false;
end;
2:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
ChBPracKan2.checked:=true;
ChBPracKan2.enabled:=true;
end
else begin
ChBPracKan2.checked:=false;
ChBPracKan2.enabled:=false;
end;
3:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
ChBPracKan3.checked:=true;
ChBPracKan3.enabled:=true;
end
else begin
ChBPracKan3.checked:=false;

```



```

        ChBPracKan3.enabled:=false;
    end;
4:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
    ChBPracKan4.checked:=true;
    ChBPracKan4.enabled:=true;
end
else begin
    ChBPracKan4.checked:=false;
    ChBPracKan4.enabled:=false;
end;
5:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
    ChBPracKan5.checked:=true;
    ChBPracKan5.enabled:=true;
end
else begin
    ChBPracKan5.checked:=false;
    ChBPracKan5.enabled:=false;
end;
6:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
    ChBPracKan6.checked:=true;
    ChBPracKan6.enabled:=true;
end
else begin
    ChBPracKan6.checked:=false;
    ChBPracKan6.enabled:=false;
end;

end; //konec case
end; // konec for
end;

procedure THlavniForm1.CBPocetDatChange(Sender: TObject);
begin
    HDatFFT:=strtoint(CBPocetDat.Text); {KONDAC = POCET DAT PRO FFT}
    ramecekGraf(1);
    ramecekGraf(2);
    ramecekFrekv(1);
    ramecekFrekv(2);
    HPrizZmenaNastaveni:=true;
end;

procedure THlavniForm1.CBDelkaRelaxChange(Sender: TObject);
begin
    DelkaRelax:=strtoint(CBDelkaRelax.Text);
end;

procedure THlavniForm1.ramecekgraf(ktery:word);
begin
    HlavniForm1.Canvas.pen.Mode:=pmcopy; {pmlblack, pmwhite}
    HlavniForm1.Canvas.pen.style:=psSolid; {plna cara}
    HlavniForm1.Canvas.Pen.Color:= clwhite;
    HlavniForm1.Canvas.pen.width:=2; {sirka cary v pixelech}
    HlavniForm1.Canvas.Brush.color:=clSkyBlue;
    case Ktery of
        1:HlavniForm1.Canvas.rectangle(Imggraf1.left-2,Imggraf1.top-2,
            (Imggraf1.left+Imggraf1.width)+4,(Imggraf1.top+Imggraf1.height)+4);
        2:HlavniForm1.Canvas.rectangle(Imggraf2.left-2,Imggraf2.top-2,
            (Imggraf2.left+Imggraf2.width)+4,(Imggraf2.top+Imggraf2.height)+4);
    end;
end;

procedure THlavniForm1.ramecekFrekv(ktery:word);
var
    igrf:word;
begin
    HlavniForm1.Canvas.pen.Mode:=pmcopy;
    HlavniForm1.Canvas.pen.style:=psSolid;
    HlavniForm1.Canvas.Pen.Color:= clwhite;
    HlavniForm1.Canvas.pen.width:=2;
    HlavniForm1.Canvas.Brush.color:=clSkyBlue;
    case Ktery of
        1:HlavniForm1.Canvas.rectangle(HlavniForm1.ImgFrekv1.left-2,
            HlavniForm1.ImgFrekv1.top-2,(HlavniForm1.ImgFrekv1.left+
            HlavniForm1.ImgFrekv1.width)+4,(HlavniForm1.ImgFrekv1.top+
            HlavniForm1.ImgFrekv1.height)+4);
        2:HlavniForm1.Canvas.rectangle(HlavniForm1.ImgFrekv2.left-2,

```

```

    HlavniForm1.ImgFrekv2.top-2,(HlavniForm1.ImgFrekv2.left+
    HlavniForm1.ImgFrekv2.width)+4,(HlavniForm1.ImgFrekv2.top+
    HlavniForm1.ImgFrekv2.height)+4);
end;
HlavniForm1.Canvas.pen.width:=1;
case Ktery of
1:begin
    for igraf:=0 to 12 do
    begin
        HlavniForm1.Canvas.moveto(HlavniForm1.ImgFrekv1.left+
        round(igraf*(HlavniForm1.ImgFrekv1.width/12)),
        HlavniForm1.ImgFrekv1.top+HlavniForm1.ImgFrekv1.height+1);
        HlavniForm1.Canvas.lineto(HlavniForm1.ImgFrekv1.left+
        round(igraf*(HlavniForm1.ImgFrekv1.width/12)),
        HlavniForm1.ImgFrekv1.top+HlavniForm1.ImgFrekv1.height+6);
    end;
    for igraf:=0 to 4 do
    begin
        HlavniForm1.Canvas.moveto((HlavniForm1.ImgFrekv1.left-1),
        (HlavniForm1.ImgFrekv1.top+
        round(igraf*(HlavniForm1.ImgFrekv1.height/4))));
        HlavniForm1.Canvas.lineto((HlavniForm1.ImgFrekv1.left-6),
        (HlavniForm1.ImgFrekv1.top+
        round(igraf*(HlavniForm1.ImgFrekv1.height/4))));
    end;
end;
2:begin
    for igraf:=0 to 12 do
    begin
        HlavniForm1.Canvas.moveto(HlavniForm1.ImgFrekv2.left+
        round(igraf*(HlavniForm1.ImgFrekv2.width/12)),
        HlavniForm1.ImgFrekv2.top+HlavniForm1.ImgFrekv2.height+1);
        HlavniForm1.Canvas.lineto(HlavniForm1.ImgFrekv2.left+
        round(igraf*(HlavniForm1.ImgFrekv2.width/12)),
        HlavniForm1.ImgFrekv2.top+HlavniForm1.ImgFrekv2.height+6);
    end;
    for igraf:=0 to 4 do
    begin
        HlavniForm1.Canvas.moveto((HlavniForm1.ImgFrekv2.left-1),
        (HlavniForm1.ImgFrekv2.top+
        round(igraf*(HlavniForm1.ImgFrekv2.height/4))));
        HlavniForm1.Canvas.lineto((HlavniForm1.ImgFrekv2.left-6),
        (HlavniForm1.ImgFrekv2.top+
        round(igraf*(HlavniForm1.ImgFrekv2.height/4))));
    end;
end;
end;
end;

procedure THlavniForm1.CBOKnoChange(Sender: TObject);
begin
    HStrDruhOkna:=CBOKno.text;
    HPrizZmenaNastaveni:=true;
end;

procedure THlavniForm1.ChBFiltrClick(Sender: TObject);
begin
    if ChBFiltr.Checked=true then HStrPrizFiltr:='Filtr1'
        else HStrPrizFiltr:='Filtr0';
    HPrizZmenaNastaveni:=true;
end;

procedure THlavniForm1.BtnNastavFrekvRozsahClick(Sender: TObject);
begin
    if (strtoint(trim(EdtRozMin.text))<=strtoint(trim(EdtRozMax.text))) and
        (strtoint(trim(EdtRozMin.text))<=42) and
        (strtoint(trim(EdtRozMax.text))<=60)
    then
    begin
        HStrFrekvMin:=trim(EdtRozMin.text);
        HStrFrekvMax:=trim(EdtRozMax.text);
        HPrizZmenaRozsahFrekv:=false;
        GBFrekvRoz.Color:=clSkyBlue;
        HPrizZmenaNastaveni:=true;
    end
end

```

```

else
begin
if (strtoint(trim(EdtRozMin.text))>60) or
(strtoint(trim(EdtRozMax.text))>60)
then showmessage('Chyba - maximální frekvence je 60 Hz')
else showmessage('Chyba - Minimální frekvence je větší než maximální.');
```

```

end;
end;

procedure THlavniForm1.BtnNastavFiltrClick(Sender: TObject);
var
HmistiPriz:boolean;
begin
HmistiPriz:=true;
if strtoint(trim(EdtFiltrMin1.text))<=strtoint(trim(EdtFiltrMax1.text))
then
begin
HStrFiMin1:=trim(EdtFiltrMin1.text);
HStrFiMax1:=trim(EdtFiltrMax1.text);
end
else
begin
showmessage('Chyba: Filtr 1 - Minimální frekvence je větší než maximální');
HmistiPriz:=false;
end;
if strtoint(trim(EdtFiltrMin2.text))<=strtoint(trim(EdtFiltrMax2.text))
then
begin
HStrFiMin2:=trim(EdtFiltrMin2.text);
HStrFiMax2:=trim(EdtFiltrMax2.text);
end
else
begin
showmessage('Chyba: Filtr 2 - Minimální frekvence je větší než maximální');
HmistiPriz:=false;
end;
if strtoint(trim(EdtFiltrMin3.text))<=strtoint(trim(EdtFiltrMax3.text)) then
begin
HStrFiMin3:=trim(EdtFiltrMin3.text);
HStrFiMax3:=trim(EdtFiltrMax3.text);
end
else
begin
showmessage('Chyba: Filtr 3 - Minimální frekvence je větší než maximální');
HmistiPriz:=false;
end;
if HmistiPriz=true
then
begin
HPrizZmenaFrekvFiltr:=false;
GBFiltr.Color:=clSkyBlue;
ChBFiltr.Color:=clSkyBlue;
HPrizZmenaNastaveni:=true;
end;
end;

procedure THlavniForm1.BtnNastavZesileniClick(Sender: TObject);
begin
if (strtoint(trim(EdtZesilKan1.text))=0) or
(strtoint(trim(EdtZesilKan2.text))=0) or
(strtoint(trim(EdtZesilKan3.text))=0) or
(strtoint(trim(EdtZesilKan4.text))=0) or
(strtoint(trim(EdtZesilKan5.text))=0) or
(strtoint(trim(EdtZesilKan6.text))=0)
then showmessage
('Zesilovač nemůže mít nulové zesílení, prosím opravte zesílení')
else
begin
HZesilKan[1]:=strtoint(trim(EdtZesilKan1.text));
HZesilKan[2]:=strtoint(trim(EdtZesilKan2.text));
HZesilKan[3]:=strtoint(trim(EdtZesilKan3.text));
HZesilKan[4]:=strtoint(trim(EdtZesilKan4.text));
HZesilKan[5]:=strtoint(trim(EdtZesilKan5.text));
HZesilKan[6]:=strtoint(trim(EdtZesilKan6.text));
HPrizZmenaZesileni:=false;

```

```

    GBZes.Color:=clSkyBlue;
  end;
end;

procedure THlavniForm1.CBGrafRozsah1Change(Sender: TObject);
begin
  ramecekGraf(1);
  ramecekGraf(2);
  ramecekFrekv(1);
  ramecekFrekv(2);
  case CBGrafRozsah1.ItemIndex of
    0:GrafZvetKan1:=1;
    1:GrafZvetKan1:=2;
    2:GrafZvetKan1:=5;
    3:GrafZvetKan1:=10;
    4:GrafZvetKan1:=20;
    5:GrafZvetKan1:=50;
    6:GrafZvetKan1:=100;
    7:GrafZvetKan1:=200;
    8:GrafZvetKan1:=500;
    9:GrafZvetKan1:=1000;
  end;
end;

procedure THlavniForm1.CBGrafRozsah2Change(Sender: TObject);
begin
  ramecekGraf(1);
  ramecekGraf(2);
  ramecekFrekv(1);
  ramecekFrekv(2);
  case CBGrafRozsah2.ItemIndex of
    0:GrafZvetKan2:=1;
    1:GrafZvetKan2:=2;
    2:GrafZvetKan2:=5;
    3:GrafZvetKan2:=10;
    4:GrafZvetKan2:=20;
    5:GrafZvetKan2:=50;
    6:GrafZvetKan2:=100;
    7:GrafZvetKan2:=200;
    8:GrafZvetKan2:=500;
    9:GrafZvetKan2:=1000;
  end;
end;

procedure THlavniForm1.NactiDataZFFT;
var Hpomret:string;
begin
  repeat
    Hprizerror:=false;
    try
      reset(vstFrekvFFT);
      try
        readln(vstFrekvFFT,HPomret);
        VstHdatFFT:=strtoint(HPomret);
        readln(vstFrekvFFT,HPomret);
        VstHPrvniKan:=strtoint(HPomret);
        readln(vstFrekvFFT,HPomret);
        VstHPosledniKan:=strtoint(HPomret);
        readln(vstFrekvFFT,HPomret);
        VstHdatFFTPul:=strtoint(HPomret);
        readln(vstFrekvFFT,HPomret);
        VstHxkoef:=(strtoint(HPomret))/10000;
      if (VstHdatFFT<>HdatFFT) or (VstHPrvniKan<>HPrvniKan) or
        (VstHposledniKan<>HPosledKan)
      then HPrizChybaDat:=true
      else
        begin
          HPrizChybaDat:=false;
          for Hi:=VstHPrvniKan to VstHPosledniKan do
            begin
              for Hj:=1 to VstHdatFFT do
                begin
                  readln(vstFrekvFFT,HPomRet);
                  VstHGrafKan[Hi,Hj]:=(strtoint(HPomRet))/10000;
                end;
            end;
          end;
        end;
    end;
  until Hprizerror=false;
end;

```



```

end;
VstHMaxHarm[Hi]:=HFMaxHarm*VstHxkoef;
VstHMaxAmpl[Hi]:=HFmaxAmpl;
poleMaxFrekvenci[Hi,HposuvMaxim]:=HFmaxHarm*VstHxkoef;
if HposuvMaxim=HPocetVyhodFrekv then HposuvMaxim:=1
    else HposuvMaxim:=HposuvMaxim+1;
end;
end;

procedure THlavniForm1.FormCreate(Sender: TObject);
begin
    HprizSpusteniAplikace:=true;
    HprizSpusteniAplAVS:=false;
    SE1HMessage := RegisterWindowMessage(SE1MsgId);
    SE1HMap:= CreateFileMapping($FFFFFFFF, nil,PAGE_READWRITE, 0, 1024,
        SE1MappingId);
    SE2HMessage := RegisterWindowMessage(SE2MsgId);
    SE2HMap:= CreateFileMapping($FFFFFFFF, nil,PAGE_READWRITE, 0, 1024,
        SE2MappingId);
end;

procedure THlavniForm1.EdtRozMinChange(Sender: TObject);
begin
    if HprizPocatNastHod=false
    then
        begin
            HPrizZmenaRozsahFrekv:=true;
            GBFrekvRoz.Color:=clMaroon;
        end;
end;

procedure THlavniForm1.EdtRozMaxChange(Sender: TObject);
begin
    if HprizPocatNastHod=false
    then
        begin
            HPrizZmenaRozsahFrekv:=true;
            GBFrekvRoz.Color:=clMaroon;
        end;
end;

procedure THlavniForm1.EdtFiltrMin1Change(Sender: TObject);
begin
    if HprizPocatNastHod=false
    then
        begin
            HPrizZmenaFrekvFiltr:=true;
            GBFiltr.Color:=clMaroon;
            ChBFiltr.Color:=clMaroon;
        end;
end;

procedure THlavniForm1.EdtFiltrMax1Change(Sender: TObject);
begin
    if HprizPocatNastHod=false
    then
        begin
            HPrizZmenaFrekvFiltr:=true;
            GBFiltr.Color:=clMaroon;
            ChBFiltr.Color:=clMaroon;
        end;
end;

procedure THlavniForm1.EdtFiltrMin2Change(Sender: TObject);
begin
    if HprizPocatNastHod=false
    then
        begin
            HPrizZmenaFrekvFiltr:=true;
            GBFiltr.Color:=clMaroon;
            ChBFiltr.Color:=clMaroon;
        end;
end;

procedure THlavniForm1.EdtFiltrMax2Change(Sender: TObject);

```

```
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaFrekvFiltr:=true;
GBFiltr.Color:=clMaroon;
ChBFiltr.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtFiltrMin3Change(Sender: TObject);
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaFrekvFiltr:=true;
GBFiltr.Color:=clMaroon;
ChBFiltr.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtFiltrMax3Change(Sender: TObject);
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaFrekvFiltr:=true;
GBFiltr.Color:=clMaroon;
ChBFiltr.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtZesilKan1Change(Sender: TObject);
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaZesileni:=true;
GBZes.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtZesilKan2Change(Sender: TObject);
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaZesileni:=true;
GBZes.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtZesilKan3Change(Sender: TObject);
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaZesileni:=true;
GBZes.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtZesilKan4Change(Sender: TObject);
begin
if HprizPocatNastHod=false
then
begin
HPrizZmenaZesileni:=true;
GBZes.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtZesilKan5Change(Sender: TObject);
begin
if HprizPocatNastHod=false
```

```

then
begin
  HPrizZmenaZesileni:=true;
  GBZes.Color:=clMaroon;
end;
end;

procedure THlavniForm1.EdtZesilKan6Change(Sender: TObject);
begin
  if HprizPocatNastHod=false
  then
  begin
    HPrizZmenaZesileni:=true;
    GBZes.Color:=clMaroon;
  end;
end;

Procedure THlavniForm1.NacteniHodnotForm;
var
  HMistRet:string;
begin
  Hprizerror:=false;
  try
    reset(vstNastaveniForm);
    try
      readln(vstNastaveniForm, HMistRet);
      CBAktivniKanalY.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBKanalGraf1.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBGrafRozsah1.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBKanalGraf2.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBGrafRozsah2.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBDelkaRelax.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBCilovaFrekv.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBFrekvBdeStavu.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBJemneDolad.Checked:=true
        else ChBJemneDolad.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBPracKan1.Checked:=true
        else ChBPracKan1.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBPracKan2.Checked:=true
        else ChBPracKan2.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBPracKan3.Checked:=true
        else ChBPracKan3.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBPracKan4.Checked:=true
        else ChBPracKan4.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBPracKan5.Checked:=true
        else ChBPracKan5.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBPracKan6.Checked:=true
        else ChBPracKan6.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      CBPocetDat.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      CBOkno.ItemIndex:=strtoint(HMistRet);
      readln(vstNastaveniForm, HMistRet);
      EdtRozMin.Text:=HMistRet;
      readln(vstNastaveniForm, HMistRet);
      EdtRozMax.Text:=HMistRet;
      readln(vstNastaveniForm, HMistRet);
      if HMistRet='True' then ChBFiltr.Checked:=true
        else ChBFiltr.Checked:=false;
      readln(vstNastaveniForm, HMistRet);
      EdtFiltrMin1.Text:=HMistRet;

```



```

readln(vstNastaveniForm, HMistRet);
EdtFiltrMin2.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtFiltrMin3.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtFiltrMax1.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtFiltrMax2.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtFiltrMax3.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
CBPosledFrekv.ItemIndex:=strtoint(HMistRet);
readln(vstNastaveniForm, HMistRet);
EdtZesilKan1.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtZesilKan2.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtZesilKan3.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtZesilKan4.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtZesilKan5.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
EdtZesilKan6.Text:=HMistRet;
readln(vstNastaveniForm, HMistRet);
MEdtSouUzivatelMaxima.Text:=HMistRet;
finally
  closefile(vstNastaveniForm);
end; // finally
except
  on e:EInOutError do Hprizerror:=true;
end; // except
end;

Procedure THlavniForm1.UlozeniHodnotForm;
begin
  Hprizerror:=false;
  try
    rewrite(vstNastaveniForm);
    try
      writeln(vstNastaveniForm, inttostr(CBAktivniKanal.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBKanalGraf1.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBGrafRozsah1.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBKanalGraf2.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBGrafRozsah2.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBDelkaRelax.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBCilovaFrekv.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBFrekvBdeStavu.ItemIndex));
      if ChBJemneDolad.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      if ChBPracKan1.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      if ChBPracKan2.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      if ChBPracKan3.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      if ChBPracKan4.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      if ChBPracKan5.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      if ChBPracKan6.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      writeln(vstNastaveniForm, inttostr(CBPocetDat.ItemIndex));
      writeln(vstNastaveniForm, inttostr(CBOKno.ItemIndex));
      writeln(vstNastaveniForm, EdtRozMin.Text);
      writeln(vstNastaveniForm, EdtRozMax.Text);
      if ChBFiltr.Checked=true then writeln(vstNastaveniForm, 'True')
        else writeln(vstNastaveniForm, 'False');
      writeln(vstNastaveniForm, EdtFiltrMin1.Text);
      writeln(vstNastaveniForm, EdtFiltrMin2.Text);
      writeln(vstNastaveniForm, EdtFiltrMin3.Text);
      writeln(vstNastaveniForm, EdtFiltrMax1.Text);
      writeln(vstNastaveniForm, EdtFiltrMax2.Text);
      writeln(vstNastaveniForm, EdtFiltrMax3.Text);
      writeln(vstNastaveniForm, inttostr(CBPosledFrekv.ItemIndex));

```

```

writeln(vstNastaveniForm,EdtZesilKan1.Text);
writeln(vstNastaveniForm,EdtZesilKan2.Text);
writeln(vstNastaveniForm,EdtZesilKan3.Text);
writeln(vstNastaveniForm,EdtZesilKan4.Text);
writeln(vstNastaveniForm,EdtZesilKan5.Text);
writeln(vstNastaveniForm,EdtZesilKan6.Text);
writeln(vstNastaveniForm,MEdtSouUzivatelMaxima.Text);
finally
  closefile(vstNastaveniForm);
end; // finally
except
  on e:ElnOutError do Hprizerror:=true;
end; // except
end;

procedure THlavniForm1.BtnNastavVychozHodClick(Sender: TObject);
begin
  HPrizNastaveniVychHodnot:=true;
end;

Procedure THlavniForm1.NastavVychoziHodnoty;
begin
  HPrizNastaveniVychHodnot:=false;
  assignfile(vstNastaveniForm,HlAktAdresStr+
    'Pracovni\VychoziNastaveni_Form.naf');
  HprizPocatNastHod:=true;
  NacteniHodnotForm;
  HprizPocatNastHod:=false;
  ramecekGraf(1);
  ramecekGraf(2);
  ramecekFrekv(1);
  ramecekFrekv(2);
  HDatFFT:=strtoint(CBPocetDat.Text); {KONDAC = POCET DAT PRO FFT}
  vyberkanal1:=strtoint(CBkanalgraf1.text); //zobr kanal 1
  vyberkanal2:=strtoint(CBkanalgraf2.text); //zobr kanal 2
  HStrvyberkan1:=CBkanalgraf1.text;
  HStrvyberkan2:=CBkanalgraf2.text;
  NastavPrvniPosledniKan;
  HStrDruhOkna:=CBOkno.text;
  HStrFrekvMin:=trim(EdtRozMin.text);
  HStrFrekvMax:=trim(EdtRozMax.text);
  if ChBFiltr.Checked=true then HStrPrizFiltr:='Filtr1'
    else HStrPrizFiltr:='Filtr0';
  HStrFiMin1:=trim(EdtFiltrMin1.text);
  HStrFiMin2:=trim(EdtFiltrMin2.text);
  HStrFiMin3:=trim(EdtFiltrMin3.text);
  HStrFiMax1:=trim(EdtFiltrMax1.text);
  HStrFiMax2:=trim(EdtFiltrMax2.text);
  HStrFiMax3:=trim(EdtFiltrMax3.text);
  HZesilKan[1]:=strtoint(trim(EdtZesilKan1.text));
  HZesilKan[2]:=strtoint(trim(EdtZesilKan2.text));
  HZesilKan[3]:=strtoint(trim(EdtZesilKan3.text));
  HZesilKan[4]:=strtoint(trim(EdtZesilKan4.text));
  HZesilKan[5]:=strtoint(trim(EdtZesilKan5.text));
  HZesilKan[6]:=strtoint(trim(EdtZesilKan6.text));
  case CBGrafRozsah1.ItemIndex of
    0:GrafZvetKan1:=1;
    1:GrafZvetKan1:=2;
    2:GrafZvetKan1:=5;
    3:GrafZvetKan1:=10;
    4:GrafZvetKan1:=20;
    5:GrafZvetKan1:=50;
    6:GrafZvetKan1:=100;
    7:GrafZvetKan1:=200;
    8:GrafZvetKan1:=500;
    9:GrafZvetKan1:=1000;
  end;
  case CBGrafRozsah2.ItemIndex of
    0:GrafZvetKan2:=1;
    1:GrafZvetKan2:=2;
    2:GrafZvetKan2:=5;
    3:GrafZvetKan2:=10;
    4:GrafZvetKan2:=20;
    5:GrafZvetKan2:=50;
    6:GrafZvetKan2:=100;
  end;

```

```

7:GrafZvetKan2:=200;
8:GrafZvetKan2:=500;
9:GrafZvetKan2:=1000;
end;
HPocetVyhodFrekv:=strtoint(CBPosledFrekv.Text);
for Hi:=1 to 6 do
begin
case Hi of
1:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
        ChBPracKan1.enabled:=true;
        ChBPracKan1.checked:=true;
        end
    else begin
        ChBPracKan1.checked:=false;
        ChBPracKan1.enabled:=false;
        end;
2:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
        ChBPracKan2.checked:=true;
        ChBPracKan2.enabled:=true;
        end
    else begin
        ChBPracKan2.checked:=false;
        ChBPracKan2.enabled:=false;
        end;
3:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
        ChBPracKan3.checked:=true;
        ChBPracKan3.enabled:=true;
        end
    else begin
        ChBPracKan3.checked:=false;
        ChBPracKan3.enabled:=false;
        end;
4:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
        ChBPracKan4.checked:=true;
        ChBPracKan4.enabled:=true;
        end
    else begin
        ChBPracKan4.checked:=false;
        ChBPracKan4.enabled:=false;
        end;
5:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
        ChBPracKan5.checked:=true;
        ChBPracKan5.enabled:=true;
        end
    else begin
        ChBPracKan5.checked:=false;
        ChBPracKan5.enabled:=false;
        end;
6:if (HPrvniKan<=Hi) and (Hi<=HPosledKan) then begin
        ChBPracKan6.checked:=true;
        ChBPracKan6.enabled:=true;
        end
    else begin
        ChBPracKan6.checked:=false;
        ChBPracKan6.enabled:=false;
        end;

end; //konec case
end; // konec for
for Hi:=1 to HmaximKan do VstHmaxHarm[Hi]:=0;
for Hi:=1 to HmaximKan do VstHmaxAmpl[Hi]:=0;
for Hi:=1 to HmaximKan do
begin
for Hj:=1 to Hnx do VstHgrafKan[Hi,Hj]:=0;
end;
for Hi:=1 to 2 do VstHmaxHarmZobrKan[Hi]:=0;
for Hi:=1 to 2 do VstHmaxAmplZobrKan[Hi]:=0;
for Hi:=1 to HmaximKan do
begin
for Hj:=1 to ((Hnx div 2)-1) do VstHfrekvKan[Hi,Hj]:=0;
end;
HPrizZmenaNastaveni:=true;
assignfile(vstNastaveniForm,HIAktAdresStr+'Pracovni\Nastaveni_Form.naf');
end;

function THlavniForm1.PrevedExtendStr81(HpomPrevodExtend:extended):string;

```

```

var
  HpomPrevodString:string;
  HpomPrevodI:word;
begin
  str(HpomPrevodExtend:8:1, HpomPrevodString);
  for HpomPrevodI:=1 to length(HpomPrevodString) do
    if HpomPrevodString[HpomPrevodI]='.' then
      HpomPrevodString[HpomPrevodI]='';
    HpomPrevodString:=trim(HpomPrevodString);
    PrevedExtendStr81:=HpomPrevodString;
  end;

function THlavniForm1.PrevedStrProGenerator81(HpomPrevodExtended:extended)
:string;
var
  HpomPrevodString:string;
  HpomPrevodExtended2:extended;
  HpomPrevodI:word;
begin
  str(HpomPrevodExtended:8:1, HpomPrevodString);
  for HpomPrevodI:=1 to length(HpomPrevodString) do
    begin
      if HpomPrevodString[HpomPrevodI]='.' then
        begin
          HpomPrevodString[HpomPrevodI]:=HpomPrevodString[HpomPrevodI+1];
          HpomPrevodString[HpomPrevodI+1]='';
        end;
      end;
    HpomPrevodString:=trim(HpomPrevodString);
    PrevedStrProGenerator81:=HpomPrevodString;
  end;

function THlavniForm1.PrevedExtendStr82(HpomPrevodExtend:extended):string;
var
  HpomPrevodString:string;
  HpomPrevodI:word;
begin
  str(HpomPrevodExtend:8:2, HpomPrevodString);
  for HpomPrevodI:=1 to length(HpomPrevodString) do
    if HpomPrevodString[HpomPrevodI]='.' then
      HpomPrevodString[HpomPrevodI]='';
    HpomPrevodString:=trim(HpomPrevodString);
    PrevedExtendStr82:=HpomPrevodString;
  end;

procedure THlavniForm1.NastavEnableFalse;
begin
  GBZes.enabled:=false;
  GBakKan.enabled:=false;
  GBNastavpuvod.enabled:=false;
  GBSoZmen.enabled:=false;
  GBPocetD.enabled:=false;
  GBOkno.enabled:=false;
  GBFrekvRoz.enabled:=false;
  GBfiltr.enabled:=false;
  GBDeRe.enabled:=false;
  GBDoFr.enabled:=false;
  GBFrBd.enabled:=false;
  GBJeDol.enabled:=false;
  GBFrekvPosl.enabled:=false;
end;

procedure THlavniForm1.NastavEnableTrue;
begin
  GBZes.enabled:=true;
  GBakKan.enabled:=true;
  GBNastavpuvod.enabled:=true;
  GBSoZmen.enabled:=true;
  GBPocetD.enabled:=true;
  GBOkno.enabled:=true;
  GBFrekvRoz.enabled:=true;
  GBfiltr.enabled:=true;
  GBDeRe.enabled:=true;
  GBDoFr.enabled:=true;
end;

```

```

    GBFrBd.enabled:=true;
    GBJeDol.enabled:=true;
    GBFrekvPosl.enabled:=true;
end;

procedure THlavniForm1.PosliKonecFFT;
var
    SE2P: PChar;
    SE2RetOdesilany:string;
begin
    SE2RetOdesilany:='Konec';
    if SE2HMap <> 0 then
    begin
        SE2P := MapViewOfFile(SE2HMap, FILE_MAP_WRITE,0, 0,
            Length(SE2RetOdesilany) + 1);
        StrCopy(SE2P, PChar(SE2RetOdesilany));
        UnmapViewOfFile(SE2P);
        SendMessage(HWND_BROADCAST, SE2HMessage, 0, 0);
    end;
end;

procedure THlavniForm1.ImpulsNahoruAVS;
begin
end;

procedure THlavniForm1.ImpulsDoluAVS;
begin
end;

procedure THlavniForm1.FormDestroy(Sender: TObject);
begin
    CloseHandle(SE1HMap);
    CloseHandle(SE2HMap);
end;

procedure THlavniForm1.CBPosledFrekvChange(Sender: TObject);
begin
    HPocetVyhodFrekv:=strtoint(CBPosledFrekv.Text);
end;

end.

unit USB_UMS3;
interface
uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,Dialogs, StdCtrls, ComCtrls, ExtCtrls, Mask;

procedure VymazVstupniZasobnik;
procedure ResetujFTDI;
procedure NastavPrenosPortu(var DyVystHlaseni:string;
    var VystCHybaZarizeni:boolean);
procedure NactiDataDoSouboru(DyHStrFrekvMin,DyHStrFrekvMax,DyHStrPrizFiltr,
    DyHStrFiMin1,DyHStrFiMin2,DyHStrFiMin3,DyHStrFiMax1,DyHStrFiMax2,
    DyHStrFiMax3,DyHStrDruhOkna:string;DyPocetDatFFT,DyPrvniKanal,
    DyPosledniKanal:word;var DyVystHlaseni:string;VystCHybaZarizeni:boolean);
procedure OtevriPort(var DyVystHlaseni:string;var VystCHybaZarizeni:boolean;
    var DyZarNo:string;var DyZarSerNo:string;var DyZarJmeno:String);
procedure ZavriPort;

implementation
uses D2XXUnit;
const dynx=1024;

var
    DevicePresent : Boolean;
    Selected_Device_Serial_Number,Selected_Device_Description : String;

Procedure SetUpBaudRate(FTDIstr : String);
Begin
    If FTDIstr = '300' then FT_Current_Baud := FT_BAUD_300 else
    If FTDIstr = '600' then FT_Current_Baud := FT_BAUD_600 else
    If FTDIstr = '1.200' then FT_Current_Baud := FT_BAUD_1200 else
    If FTDIstr = '2.400' then FT_Current_Baud := FT_BAUD_2400 else

```

```

If FTDIstr = '4.800' then FT_Current_Baud := FT_BAUD_4800 else
If FTDIstr = '9.600' then FT_Current_Baud := FT_BAUD_9600 else
If FTDIstr = '19.200' then FT_Current_Baud := FT_BAUD_19200 else
If FTDIstr = '38.400' then FT_Current_Baud := FT_BAUD_38400 else
If FTDIstr = '57.600' then FT_Current_Baud := FT_BAUD_57600 else
If FTDIstr = '115.200' then FT_Current_Baud := FT_BAUD_115200 else
If FTDIstr = '230.400' then FT_Current_Baud := FT_BAUD_230400 else
If FTDIstr = '460.800' then FT_Current_Baud := FT_BAUD_460800 else
If FTDIstr = '921.600' then FT_Current_Baud := FT_BAUD_921600 else
FT_SetupError := True;
end;

Procedure SetUpDataBits(FTDIstr : String);
Begin
If FTDIstr = '7 bits' then FT_Current_DataBits := FT_DATA_BITS_7 else
If FTDIstr = '8 bits' then FT_Current_DataBits := FT_DATA_BITS_8 else
FT_SetupError := True;
end;

Procedure SetUpStopBits(FTDIstr : String);
Begin
If FTDIstr = '1 bit' then FT_Current_StopBits := FT_STOP_BITS_1 else
If FTDIstr = '2 bits' then FT_Current_StopBits := FT_STOP_BITS_2 else
FT_SetupError := True;
end;

Procedure SetUpParity(FTDIstr : String);
Begin
If FTDIstr = 'None' then FT_Current_Parity := FT_PARITY_NONE else
If FTDIstr = 'Odd' then FT_Current_Parity := FT_PARITY_ODD else
If FTDIstr = 'Even' then FT_Current_Parity := FT_PARITY_EVEN else
If FTDIstr = 'Mark' then FT_Current_Parity := FT_PARITY_MARK else
If FTDIstr = 'Space' then FT_Current_Parity := FT_PARITY_SPACE else
FT_SetupError := True;
end;

Procedure SetUpFlowControl(FTDIstr : String);
Begin
If FTDIstr = 'None' then FT_Current_FlowControl := FT_FLOW_NONE else
If FTDIstr = 'RTS/CTS' then FT_Current_FlowControl := FT_FLOW_RTS_CTS else
If FTDIstr = 'DTR/DSR' then FT_Current_FlowControl := FT_FLOW_DTR_DSR else
If FTDIstr = 'X-On/X-Off' then FT_Current_FlowControl := FT_FLOW_XON_XOFF else
FT_SetupError := True;
end;

Function HexToByte( FTDIstr1 : String; Var FTDIHexVal : Byte ) : Boolean;
Var FTDIstr2 : String;
Begin
FTDIstr2 := UpperCase(FTDIstr1);
FTDIHexVal := 0;
If (( FTDIstr2[1] in ['0'..'9'] ) or ( FTDIstr2[1] in ['A'..'F'] )) and
(( FTDIstr2[2] in ['0'..'9'] ) or ( FTDIstr2[2] in ['A'..'F'] ))
then
Begin
Result := true;
If ( FTDIstr2[1] in ['0'..'9'] )
then FTDIHexval := Ord(FTDIstr2[1])-Ord('0')
else FTDIHexval := Ord(FTDIstr2[1])-Ord('A')+10;
FTDIHexval := FTDIHexval * 16;
If ( FTDIstr2[2] in ['0'..'9'] )
then FTDIHexval := FTDIHexval + Ord(FTDIstr2[2])-Ord('0')
else FTDIHexval := FTDIHexval + Ord(FTDIstr2[2])-Ord('A')+10;
End
else Result := false;
end;

procedure NastavPrenosPortu(var DyVystHlaseni:string;
var VystCHybaZarizeni:boolean);
var
SouBaudRate,SouDatabits,SouStopBits,SouParity,SouFlowControl,SouRTSON,
SouDTRon,SouEventon,SouErroron,SouXONVal,SouXOFFVal,SouEventVal,SouErrorVal,
SouReadTimeOut,SouWriteTimeOut,FTDIstrX:string;
begin
VystCHybaZarizeni:=false;

```

```

SouBaudRate:='57.600';
SouDatabits:='8 bits';
SouStopBits:='1 bit';
SouParity:='None';
SouFlowControl:='None';
SouRTSON:='RTS1';
SouDTRON:='DTR1';
SouEventOn:='Event0';
SouErrorOn:='Error0';
SouXONVal:='11';
SouXOFFVal:='13';
SouEventVal:='00';
SouErrorVal:='00';
SouReadTimeOut:='300';
SouWriteTimeOut:='300';
FT_SetupError := False;
SetUpBaudRate(SouBaudRate);
SetUpDataBits(SouDatabits);
SetUpStopBits(SouStopBits);
SetUpParity(SouParity);
SetUpFlowControl(SouFlowControl);
if SouRTSON='RTS1' then FT_RTS_On := true
    else FT_RTS_On := false;
if SouDTRON='DTR1' then FT_DTR_On := true
    else FT_DTR_On := false;
if SouEventOn='Event1' then FT_Event_On := true
    else FT_Event_On := false;
if SouErrorOn='Error1' then FT_Error_On := true
    else FT_Error_On := false;
FTDIStRX := SouXONVal;
if Not HexToByte(FTDIStRX,FT_XON_Value ) then FT_SetupError := True;
FTDIStRX := SouXOFFVal;
if Not HexToByte(FTDIStRX,FT_XOFF_Value ) then FT_SetupError := True;
FTDIStRX := SouEventVal;
if Not HexToByte(FTDIStRX,FT_EVENT_Value ) then FT_SetupError := True;
FTDIStRX := SouErrorVal;
if Not HexToByte(FTDIStRX,FT_ERROR_Value ) then FT_SetupError := True;
Set_USB_Device_BaudRate;
Set_USB_Device_DataCharacteristics;
Set_USB_Device_FlowControl;
Set_USB_Device_RTS;
Clr_USB_Device_DTR;
Set_USB_Device_Chars;
Set_USB_Device_TimeOuts(strtoint(SouReadTimeOut),strtoint(SouWriteTimeOut));
VystChybaZarizeni:=FT_SetUperror;
if FT_SetUperror then DyVystHlaseni:='Chyba Konfigurace zařízení. '
    else DyVystHlaseni:='Konfigurace zařízení v pořádku. ';
end;

procedure NactiDataDoSouboru(DyHStrFrekvMin,DyHStrFrekvMax,DyHstrPrizFiltr,
    DyHStrFiMin1,DyHStrFiMin2,DyHStrFiMin3,DyHStrFiMax1,DyHStrFiMax2,DyHStrFiMax3,
    DyHStrDruhOkna:string;DyPocetDatFFT,DyPrvniKanal,DyPosledniKanal:word;
    var DyVystHlaseni:string;VystChybaZarizeni:boolean);
var
    NaPortStatus : FT_Result;
    DyTotal,DyI,DyQ_Bytes,DyPocetNacitDat,Dyjk,Dyik,DyCisloSpravNacit,DyPomPoradi,
    DyPoradi,DyRadPole,DyVstCi,DyCisloKan,DyAktTik2,DyAktTik1:longword;
    DyVstPole:array [1..7,1..(DyNx+5)] of word;
    DyVstBuffer : Array[1..FT_In_Buffer_Size] of Byte;
    DyPrizPocatNacit,DyPrizSpravNacit,prizChybyZarizeni,Dyprizerrors:boolean;
    DySoubVyst:textfile;
    DyAktAdresStr:string;
begin
    DyVystHlaseni:='';
    VystChybaZarizeni:=false;
    prizChybyZarizeni:=false;
    DyVstCi:=0;
    DyPocetNacitDat:=((DyPocetDatFFT*17)+(5*17));
    FT_Enable_Error_Report := false; // vypnutí Error-Report
    DyTotal := 0;
    DyAktAdresStr:= (ExtractFilePath(Application.ExeName))+'\Pracovni\';
    AssignFile(DySoubVyst,DyAktAdresStr+'OpenEEG_NactenaData.dat');
    Repeat
        Application.ProcessMessages;
    until NaPortStatus := Get_USB_Device_QueueStatus;
end;

```

```

If NaPortStatus <> FT_OK
then // Device no longer present ...
begin
  DyVystHlaseni:=DyVystHlaseni+'Chyba nacistani. ';
  prizChybyZarizeni:=true;
end
else
begin
if FT_Q_Bytes>(DyPocetNacitDat-DyTotal)
then DyQ_Bytes:=(DyPocetNacitDat-DyTotal)
else DyQ_Bytes:=FT_Q_Bytes;
if DyQ_Bytes > 0
then
begin
  DyI := Read_USB_Device_Buffer(DyQ_Bytes);
  for Dyik:=1 to DyI do DyVstBuffer[Dyik+Dytotal]:=FT_In_Buffer[Dyik-1];
  DyTotal := DyTotal + DyI;
end;
end; //konec else
Until (DyTotal>=DyPocetNacitDat) or (NaPortStatus<>FT_OK);
DyVystHlaseni:=DyVystHlaseni+'Načteno Bajtů : '+IntToStr(DyTotal)+'.';
if prizChybyZarizeni=false
then
begin
  DyPrizPocatNacit:=true;
  DyPrizSpravNacit:=true;
  DyCisloSpravNacit:=0;
  DyPomPoradi:=1;
  DyPoradi:=1;
  DyRadPole:=0;
  Dyik:=0;
  repeat
  Dyik:=Dyik+1;
  DyVstCi:=DyVstBuffer[Dyik];
  case DyPomPoradi of
  1: if DyPrizPocatNacit=true
  then
  begin
  if DyVstCi=165 then DyPoradi:=DyPoradi+1
  else DyPoradi:=1;
  end
  else
  begin
  if DyVstCi=165 then DyPoradi:=DyPoradi+1
  else DyPrizSpravNacit:=false;
  end;
  2: if DyPrizPocatNacit=true
  then
  begin
  if DyVstCi=90 then DyPoradi:=DyPoradi+1
  else DyPoradi:=1;
  end
  else
  begin
  if DyVstCi=90 then DyPoradi:=DyPoradi+1
  else DyPrizSpravNacit:=false;
  end;
  3: if DyPrizPocatNacit=true
  then
  begin
  if DyVstCi=2 then DyPoradi:=DyPoradi+1
  else DyPoradi:=1;
  end
  else
  begin
  if DyVstCi=2 then DyPoradi:=DyPoradi+1
  else DyPrizSpravNacit:=false;
  end;
  4: begin
  DyPoradi:=DyPoradi+1;
  DyRadPole:=DyRadPole+1;
  if DyPrizPocatNacit=true
  then
  begin
  DyPrizPocatNacit:=false;

```



```
DyCisloSpravNacit:=DyVstCi;
end
else
begin
if DyCisloSpravNacit=255
then DyCisloSpravNacit:=0
else DyCisloSpravNacit:=DyCisloSpravNacit+1;
if DyCisloSpravNacit<>DyVstCi
then
begin
DyPrizSpravNacit:=false;
DyVystHlaseni:=DyVystHlaseni+
'Chyba správnosti načítání - shody rámců. ';
end;
end;
end;
end;
5,7,9,11,13,15: begin
DyPoradi:=DyPoradi+1;
DyCisloKan:=(DyPomPoradi-3) div 2;
case DyVstCi of
0:DyVstPole[DyCisloKan,DyRadPole]:=0;
1:DyVstPole[DyCisloKan,DyRadPole]:=256;
2:DyVstPole[DyCisloKan,DyRadPole]:=512;
3:DyVstPole[DyCisloKan,DyRadPole]:=768;
end;
end;
6,8,10,12,14,16: begin
DyPoradi:=DyPoradi+1;
DyCisloKan:=(DyPomPoradi-3) div 2;
DyVstPole[DyCisloKan,DyRadPole]:=
DyVstPole[DyCisloKan,DyRadPole]+DyVstCi;
end;
17: begin
DyPoradi:=1;
DyVstPole[7,DyRadPole]:=DyVstCi;
end;
end; // konec case
DyPomPoradi:=DyPoradi;
until (DyRadPole=(DyPocetDatFFT+1)) or (Dyik>=(((DyPocetDatFFT+1)*17)+18))
or (DyPrizSpravNacit=false);
if (Dyik>=(((DyPocetDatFFT+1)*17)+18)) and (DyRadPole<>(DyPocetDatFFT+1))
then
begin
DyPrizSpravNacit:=false;
DyVystHlaseni:=DyVystHlaseni+'Zapnuta korekce. ';
end;

if DyPrizSpravNacit=true
then
begin
repeat
Dyprizerrors:=false;
try
rewrite(DySoubVyst);
try
writeln(DySoubvyst,DyHStrFrekvMin);
writeln(DySoubvyst,DyHStrFrekvMax);
writeln(DySoubvyst,DyHStrPrizFiltr);
writeln(DySoubvyst,DyHStrFiMin1);
writeln(DySoubvyst,DyHStrFiMin2);
writeln(DySoubvyst,DyHStrFiMin3);
writeln(DySoubvyst,DyHStrFiMax1);
writeln(DySoubvyst,DyHStrFiMax2);
writeln(DySoubvyst,DyHStrFiMax3);
writeln(DySoubvyst,DyHStrDruhOkna);
writeln(DySoubvyst,inttostr(DyPocetDatFFT));
writeln(DySoubvyst,inttostr(DyPrvniKanal));
writeln(DySoubvyst,inttostr(DyPosledniKanal));
for Dyjk:=DyPrvniKanal to DyPosledniKanal do
begin
for Dyik:=1 to DyPocetDatFFT do
begin
writeln(DySoubvyst,inttostr(DyVstPole[Dyjk,Dyik]));
end;
end;
end;
end;
```

```

    finally
    closefile(DySoubVyst);
end; // finally
except
on e:ElnOutError do Dyprizerrors:=true;
end; // except
Application.ProcessMessages;
until Dyprizerrors=false;
DyVystHlaseni:=DyVystHlaseni+'Data zapsana do souboru. ';
end;
end; // konec chyba zarizeni
VystCHybaZarizeni:=prizChybyZarizeni;
end;

procedure OtevriPort(var DyVystHlaseni:string;var VystCHybaZarizeni:boolean;
var DyZarNo:string;var DyZarSerNo:string;var DyZarJmeno:String);
var
prizChybyZarizeni:boolean;
DyIm : longword;
OtDeviceIndex : DWord;
OtPortStatus : FT_Result;
begin
DyVystHlaseni:="";
VystCHybaZarizeni:=false;
prizChybyZarizeni:=true;
FT_Enable_Error_Report := true;
DevicePresent := False;
GetFTDeviceCount;
OtDeviceIndex := 0;
if FT_Device_Count > 0
then
begin
DyIm:=0;
repeat
DyIm:=DyIm+1;
DyZarNo:=IntToStr(DyIm); //i= cislo zarizeni
GetFTDeviceSerialNo( OtDeviceIndex );
DyZarSerNo:=(FT_Device_String);
GetFTDeviceDescription ( OtDeviceIndex );
DyZarJmeno:=(FT_Device_String);
OtDeviceIndex := OtDeviceIndex + 1;
until (DyIm=FT_Device_Count) or (DyZarJmeno='David_OpenEEG');
if DyZarJmeno='David_OpenEEG'
then
begin
OtDeviceIndex := OtDeviceIndex - 1;
DyVystHlaseni:=DyVystHlaseni+'Nalezeno zarizeni '+DyZarJmeno+'.';
prizChybyZarizeni:=false;
end;
end; // konec if
if prizChybyZarizeni=true
then DyVystHlaseni:=DyVystHlaseni+'Nebylo nalezeno periferni zarizeni.'
else
begin
Selected_Device_Serial_Number := DyZarSerNo;
Selected_Device_Description := DyZarJmeno;
end; // konec else
if prizChybyZarizeni=false
then
begin
if Open_USB_Device_By_Serial_Number(Selected_Device_Serial_Number) <> FT_OK
then
begin
OtPortStatus := Close_USB_Device;
OtPortStatus := Open_USB_Device;
if OtPortStatus <> FT_OK
then
begin
prizChybyZarizeni:=true;
DyVystHlaseni:=DyVystHlaseni+'Zarizeni nelze otevrit.';
end;
end;
end;
end;
VystCHybaZarizeni:=prizChybyZarizeni;
end;
end;

```

```
procedure ZavriPort;
begin
  close_USB_Device;
end;

procedure VymazVstupniZasobnik;
begin
  Purge_USB_Device_In;
end;

procedure ResetujFTDI;
begin
  Reset_USB_Device;
end;

end.
```

Příloha P11 Výpis zdrojového kódu programu FFTnew

```

unit FFTnewZ;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TFFTForm1 = class(TForm)
    BtnAltZapnuti: TButton;
    BtnAltKonec: TButton;
    procedure mywindow;
    procedure FormPaint(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BtnAltZapnutiClick(Sender: TObject);
    procedure BtnAltKonecClick(Sender: TObject);
    procedure FormDestroy(Sender: TObject);

  private
    procedure PoSpusteniFFT;
    procedure WndProc(var CL2Message: TMessage); override;
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FFTForm1: TFFTForm1;

implementation
uses cplx, fourier;
const FMaxKanal=6;
      Fnx=1024;
      CL2MsgId = 'Message_FFT';
      CL2MappingId = 'MappingFFT';

var
  Fprizkonce,HprizSpusteniAplikaceFFT:boolean;
  Fapole:array[0..(Fnx-1)]of complex;
  Fnm,Fnm2:longint;
  FTypOkna,FSouPrizKonce:string;
  FTik1,Ftik2:longword;
  CL2HMessage: THandle;
  CL2HMap: THandle;

{SR *.dfm}
procedure TFFTForm1.FormPaint(Sender: TObject);
begin
  if HprizSpusteniAplikaceFFT=true
  then
    begin
      FFTForm1.WindowState:=wsminimized;
      PoSpusteniFFT;
    end;
end;

procedure TFFTForm1.BtnAltZapnutiClick(Sender: TObject);
begin
  PoSpusteniFFT;
end;

procedure TFFTForm1.PoSpusteniFFT;
var
  Fkanal,FPrvniKan,FPosledniKan:byte;
  FmaxAmpl,FAktAmpl,Fxkoef,FcasNacitani,Fnaskoef,Fpomerampli:extended;
  Fii,Fji,FmaxHarm,FminimFrekv,FmaximFrekv,FiMin1,FiMin2,FiMin3,FiMax1,FiMax2,
  FiMax3:word;
  Fprizerror:boolean;
  Fretvst,Fstrverzekanalu,FAktAdresStr,FRetFiltr:string;
  FSouvst,FSouvyst:textfile;
  Fpolevst:array[1..FMaxKanal,0..(Fnx-1)] of word;

```

```

Fgrapole:array[1..FMaxKanal,0..(Fnx-1)] of extended;
FPoleZobrAktFrekv:array[1..FMaxKanal,1..((Fnx div 2)-1)] of extended;
begin
HprizSpusteniAplikaceFFT:=false;
FSouPrizKonce:="";
FFTform1.Caption:='Mexická FFnew - nastaveni promeny';
Fprizkonce:=false;
FAktAdresStr:=(ExtractFilePath(Application.ExeName))+'\Pracovni\';
AssignFile(FSouVst,FAktAdresStr+'OpenEEG_NactenaData.dat');
AssignFile(FSouVyst,FAktAdresStr+'FFT_Vysledky.dat');
for Fii:=1 to FMaxKanal do
begin
for Fji:=0 to (Fnx-1) do Fpolevst[Fii,Fji]:=0;
end;
repeat
for Fii:=1 to FMaxKanal do
begin
for Fji:=1 to ((Fnx div 2)-1) do FPoleZobrAktFrekv[Fii,Fji]:=0;
end;
for Fii:=1 to FMaxKanal do
begin
for Fji:=0 to (Fnx-1) do Fgrapole[Fii,Fji]:=0;
end;
Application.ProcessMessages;
FFTform1.Caption:='Mexická FFnew - nacteni dat ze vstupniho souboru';
repeat
Fprizerror:=false;
try
reset(FSouVst);
try
readln(FsouVst,Fretvst);
Fminimfrekv:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
Fmaximfrekv:=strtoint(Fretvst);
readln(FsouVst,FRetFiltr);
readln(FsouVst,Fretvst);
FiMin1:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FiMin2:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FiMin3:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FiMax1:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FiMax2:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FiMax3:=strtoint(Fretvst);
readln(FsouVst,FTypOkna);
readln(FsouVst,Fretvst);
Fnm:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FPrvniKan:=strtoint(Fretvst);
readln(FsouVst,Fretvst);
FPosledniKan:=strtoint(Fretvst);
for Fii:=FPrvniKan to FPosledniKan do
begin
for Fji:=0 to (Fnm-1) do
begin
readln(FsouVst,Fretvst);
Fpolevst[Fii,Fji]:= strtoint(Fretvst);
end;
end;
finally
closefile(FsouVst);
end; // finally
except
on e:EInOutError do Fprizerror:=true;
end; // except
Application.ProcessMessages;
until Fprizerror=false;
Fnm2:=Fnm div 2;
FcasNacitani:=Fnm/256;
Fkcoef:=1/FcasNacitani;
Application.ProcessMessages;
FFTform1.Caption:='Mexická FFnew - vypocet FFT';

```

```

for Fkanal:=FPrvniKan to FPosledniKan do
begin
  fftInit(Fnm);
  for Fii:=0 to (Fnm-1) do
  begin
    Fapole[Fii].r:=Fpolestvst[Fkanal,Fii];
    Fapole[Fii].i:=0;
  end;
  mywindow;
  FFT(Fapole,direct);
  Application.ProcessMessages;
  FFTform1.Caption:='Mexická FFnew - prepocet na skutečne frekvence!';
  if FRetFiltr='Filtr1' then
  begin
    for Fii:=1 to (Fnm2-1) do
    begin
      if ( ((Fkcoef*Fii)>=FiMin1) and ((Fkcoef*Fii)<=FiMax1) ) or
        ( ((Fkcoef*Fii)>=FiMin2) and ((Fkcoef*Fii)<=FiMax2) ) or
        ( ((Fkcoef*Fii)>=FiMin3) and ((Fkcoef*Fii)<=FiMax3) )
      then
      begin
        Fapole[Fii].r:=0;
        Fapole[Fii].i:=0;
        Fapole[Fnm-Fii].r:=0;
        Fapole[Fnm-Fii].i:=0;
      end;
    end;
    Application.ProcessMessages;
    for Fii:=1 to (Fnm2-1) do
    begin
      if ((Fii*Fkcoef)<=FmaximFrekv) and ((Fii*Fkcoef)>=FminimFrekv) and
        ((Fii*Fkcoef)<60)
      then
      begin
        FAktAmpl:=( sqrt ( (Fapole[Fii].r*Fapole[Fii].r)+
          (Fapole[Fii].i*Fapole[Fii].i) ) )/Fnm;
        FPoleZobrAktFrekv[Fkanal,Fii]:=FAktAmpl;
      end
      else FPoleZobrAktFrekv[Fkanal,Fii]:=0;
    end;
    FFT(Fapole,inverse);
    for Fii:=0 to (Fnm-1) do Fgrapole[Fkanal,Fii]:=(Fapole[Fii].r);
  end;
  FFTform1.Caption:='Mexická FFTnew - vypis frekvenci do vystupniho souboru!';
  Application.ProcessMessages;
  repeat
  Fprizerror:=false;
  try
  rewrite(Fsouvyyst);
  try
  writeln(Fsouvyyst,inttostr(Fnm));
  writeln(Fsouvyyst,inttostr(FPrvniKan));
  writeln(Fsouvyyst,inttostr(FPosledniKan));
  writeln(Fsouvyyst,inttostr(Fnm2));
  writeln(Fsouvyyst,inttostr(round(Fkcoef*10000)));
  for Fii:=FPrvniKan to FPosledniKan do
  begin
    for Fji:=0 to (Fnm-1) do writeln(Fsouvyyst,
      inttostr(round(Fgrapole[Fii,Fji]*10000) ) );
  end;
  for Fii:=FPrvniKan to FPosledniKan do
  begin
    for Fji:=1 to (Fnm2-1) do writeln(Fsouvyyst,
      inttostr(round(FPoleZobrAktFrekv[Fii,Fji]*10000) ) );
  end;
  finally
  closefile(Fsouvyyst);
  end; // finally
  except
  on e:ElnOutError do Fprizerror:=true;
  end; // except
  Application.ProcessMessages;
  until Fprizerror=false;
  Application.ProcessMessages;

```

```

until FSouPrizKonce='Konec';
FFTform1.Caption:='Mexická FFTnew 011';
application.Terminate;
end;

procedure TFFTForm1.mywindow;
const
  Fbc1=0.35875;
  Fbc2=0.48829;
  Fbc3=0.14128;
  Fbc4=0.01168;
  Fnpi=3.141592653589793238;
var
  Fc:char;
  Fi:longint;
  Ft,Fw,Fw1:extended;
begin
  Fc:=FTypOkna[3];
  case Fc of
    'n': begin //Hanningovo'
      for Fi:=0 to (Fnm-1) do
        begin
          Fapole[Fi].r:=Fapole[Fi].r*( 0.5-0.5*cos( (2*Fnpi/Fnm)*Fi) );
        end;
      end;
    'a': begin //Blackmanovo'
      for Fi:=0 to (Fnm-1) do
        begin
          Fw:=2.0*Fnpi*Fi/Fnm;
          Ft:=Fbc1-Fbc2*cos(Fw)+Fbc3*cos(2.0*Fw)-Fbc4*cos(3.0*Fw);
          Fapole[Fi].r:=Fapole[Fi].r*Ft/Fbc1;
        end;
      end;
    'm': begin ///Hammingovo'
      for Fi:=0 to (Fnm-1) do
        begin
          Fapole[Fi].r:=Fapole[Fi].r*( 0.54-0.46*cos( (2*Fnpi/Fnm)*Fi) );
        end;
      end;
    else begin /// 'Obdélníkové'
      {zadna akce}
    end;
  end;{konec case}
end;

procedure TFFTForm1.WndProc(var CL2Message: TMessage);
const
  CL2MaxTextLength = 1024;
var
  CL2P: PChar;
  CL2StrPrijmany:string;
begin
  with CL2Message do
    if Msg = CL2HMessage
    then
      begin
        if CL2HMap <> 0
        then
          begin
            CL2P := MapViewOfFile(CL2HMap, FILE_MAP_READ,0, 0, CL2MaxTextLength);
            CL2StrPrijmany:=CL2P;
            FSouPrizKonce:= CL2StrPrijmany;
            UnmapViewOfFile(CL2P);
          end;
          Result:=0;
        end
      else
        inherited;
    end;
end;

procedure TFFTForm1.FormCreate(Sender: TObject);
begin
  HprizSpusteniAplikaceFFT:=true;
  CL2HMessage := RegisterWindowMessage(CL2MsgId);
  CL2HMap:= OpenFileMapping(FILE_MAP_READ,False, CL2MappingId);

```

```
end;

procedure TFFTForm1.BtnAltKonecClick(Sender: TObject);
begin
  FSouPrizKonce:='Konec';
end;

procedure TFFTForm1.FormDestroy(Sender: TObject);
begin
  CloseHandle(CL2HMap);
end;

end.
```


Příloha P12 Výpis zdrojového kódu programu Generátor

```

unit GeneratorZ;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, MMSystem, Mask;

type
  Tfrm_main = class(TForm)
    btn_start: TButton;
    GBFrekv1: TGroupBox;
    MEdtFrekv1: TMaskEdit;
    GBFrekv2: TGroupBox;
    MEdtFrekv2: TMaskEdit;
    btn_konec: TButton;
    procedure FormCreate(Sender: TObject);
    procedure btn_startClick(Sender: TObject);
    procedure MEdtFrekv2Change(Sender: TObject);
    procedure MEdtFrekv1Change(Sender: TObject);
    procedure btn_konecClick(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormPaint(Sender: TObject);

  private
    { Private declarations }
    procedure generuj;
  public
    { Public declarations }
    procedure WndProc(var CLIMessage: TMessage); override;
  end;

var
  frm_main: Tfrm_main;

implementation

{$R *.dfm}

const
  SAMP_FREKV = 48000; //vzorkovací frekvence
  SAMP_KANALU = 2; //pocet kamalu, 1-mono 2-stereo
  SAMP_VZORKU = 48000; //8192; //pocet vzorku v bufferu
  SAMP_BPS = 2; //rozliseni vzorku, 16bit
  SAMP_BUFFERU = 2; //pocet bufferu
  GMinAmplAS=4000; //1600
  GMaxAmplAS=16000; //16000
  GAmplsig1=8000;
  GAmplsig2=8000;
  npi=3.1415926535897932385;
  FrekvAStrobe=19200;
  CLIMsgId = 'Message_FrekvStrobe';
  CLIMappingId = 'MappingStrobe';

var Gjed:longword;
  GSmerpilaUp,CL1prizSpusteniAplikaceFFT,Gprizkonce:boolean;
  GDelkaPulPila,GPrirPila,GAmplAS,GFrekv,L1,L2,L3,R1,R2,R3,GFrekSig1,
  GFrekSig2:extended;
  CL1HMessage: THandle;
  CL1HMap: THandle;
  CL1Frekv:string;

procedure Tfrm_main.Generuj;
var
  i, hraje, f: Integer;
  fi: Double;
  b: PSmallint;
  h: array[0..SAMP_BUFFERU-1] of WAVEHDR;
  wo: HWAVEOUT;
  wf: PCMWAVEFORMAT;
begin
  CL1prizSpusteniAplikaceFFT:=false;

```

```

fi := 0.0; //fazovy posun
hraje := 0; //cislo bufferu, ktery zrovna hraje
for i := 0 to SAMP_BUFFERU-1 do
begin
  FillChar(h[i], sizeof(h[i]), 0);
  h[i].dwBufferLength := SAMP_VZORKU*SAMP_KANALU*SAMP_BPS;
  h[i].dwLoops := 1;
  GetMem(h[i].lpData, h[i].dwBufferLength);
end;
wf.wBitsPerSample := 8*SAMP_BPS;
wf.wFormatTag := WAVE_FORMAT_PCM;
wf.wf.nChannels := SAMP_KANALU;
wf.wf.nSamplesPerSec := SAMP_FREKV;
wf.wf.nBlockAlign := SAMP_KANALU*SAMP_BPS;
wf.wf.nAvgBytesPerSec := SAMP_KANALU*SAMP_BPS*SAMP_FREKV;
waveOutOpen(@wo, WAVE_MAPPER, @wf, 0, 0, CALLBACK_NULL);
for i := 0 to SAMP_BUFFERU-1 do h[i].dwFlags := h[i].dwFlags or WHDR_DONE;
repeat
if ((h[hraje].dwFlags and WHDR_DONE) = WHDR_DONE)
then
begin
  b := Pointer(h[hraje].lpData);
  waveOutUnprepareHeader(wo, @h[hraje], sizeof(h[hraje]));
  FillChar(h[hraje], sizeof(h[hraje]), 0);
  h[hraje].lpData := Pointer(b);
  h[hraje].dwBufferLength := SAMP_VZORKU*SAMP_KANALU*SAMP_BPS;
  h[hraje].dwLoops := 1;
  if CL1Frekv='Konec'
  then
  begin
    Gprizkonce:=true;
    Gfrekv:=0;
  end
  else Gfrekv:=(strtoint(CL1Frekv))/10;
  GDelkaPulPila:=(SAMP_FREKV/GFrekv)/2;
  GPrirPila:=(GMaxAmplAS-GMinAmplAS)/GDelkaPulPila;
  Gjed:=1;
  GAmplAS:=GMinAmplAS;
  GSmerPilaUp:=true;
  for i := 0 to SAMP_VZORKU-1 do
  begin
    R2 := (GAmplsig1 * Sin(2 * nPi * GFrekSig1 * I / SAMP_FREKV));
    L2 := (GAmplsig1 * Sin(2 * nPi * (GFrekSig1+GFrekv) * I / SAMP_FREKV));
    R3 := (GAmplsig2 * Sin(2 * nPi * GFrekSig2 * I / SAMP_FREKV));
    L3 := (GAmplsig2 * Sin(2 * nPi * (GFrekSig2+GFrekv) * I / SAMP_FREKV));
    L1 := (GAmplAS * Sin(2 * nPi * FrekvAStroke * I / SAMP_FREKV));
    R1 := (GAmplAS * Sin(2 * nPi * FrekvAStroke * I / SAMP_FREKV));
    if I>=(GJed*GDelkaPulPila)
    then
    begin
      Gjed:=Gjed+1;
      if GSmerPilaUp=true then GSmerPilaUp:=false
      else GSmerPilaUp:=true;
    end;
    if GSmerPilaUp=true then GAmplAS:=GAmplAS+GPrirPila
    else GAmplAS:=GAmplAS-GPrirPila;

    b^ := Trunc(L1+L2+L3);
    Inc(b);
    b^ := Trunc(R1+R2+R3);
    Inc(b);
  end;
  waveOutPrepareHeader(wo, @h[hraje], sizeof(h[hraje]));
  waveOutWrite(wo, @h[hraje], sizeof(h[hraje]));
  hraje := (hraje+1) mod SAMP_BUFFERU;
end;
Application.ProcessMessages;
until Gprizkonce=true;
waveOutReset(wo);
for i := 0 to SAMP_BUFFERU-1 do waveOutUnprepareHeader(wo, @h[i], sizeof(h));
waveOutClose(wo);
for i := 0 to SAMP_BUFFERU-1 do FreeMem(h[i].lpData);
application.terminate;
end;

```

```
procedure Tfrm_main.WndProc(var CL1Message: TMessage);
const
  CL1MaxTextLength = 1024;
var
  CL1P: PChar;
begin
  with CL1Message do
    if Msg = CL1HMessage
    then
      begin
        if CL1HMap <> 0
        then
          begin
            CL1P := MapViewOfFile(CL1HMap, FILE_MAP_READ,
              0, 0, CL1MaxTextLength);
            CL1Frekv:=CL1P;
            UnmapViewOfFile(CL1P);
            if CL1Frekv='Konec'
            then application.Terminate
            else frm_main.caption:= 'Audio Strobe '+
              (floattostr(((strtoint(CL1Frekv))/10)))+ ' Hz';
          end;
          Result:=0;
        end
      else inherited;
    end;
end;

procedure Tfrm_main.FormCreate(Sender: TObject);
begin
  Gprizkonce:=false;
  CL1prizSpusteniAplikaceFFT:=true;
  GFrekSig1:=220;
  GFrekSig2:=440;
  CL1Frekv:='300';
  CL1HMessage := RegisterWindowMessage(CL1MsgId);
  CL1HMap:= OpenFileMapping(FILE_MAP_READ,False, CL1MappingId);
end;

procedure Tfrm_main.btn_startClick(Sender: TObject);
begin
  Generuj;
end;

procedure Tfrm_main.MEdtFrekv2Change(Sender: TObject);
begin
  GFrekSig2:=strtoint(trim(MEdtFrekv2.text));
end;

procedure Tfrm_main.MEdtFrekv1Change(Sender: TObject);
begin
  GFrekSig1:=strtoint(trim(MEdtFrekv1.text));
end;

procedure Tfrm_main.btn_konecClick(Sender: TObject);
begin
  Gprizkonce:=true;
end;

procedure Tfrm_main.FormDestroy(Sender: TObject);
begin
  CloseHandle(CL1HMap);
end;

procedure Tfrm_main.FormPaint(Sender: TObject);
begin
  if CL1prizSpusteniAplikaceFFT=true
  then
    begin
      Frm_main.WindowState:=wsminimized;
      Generuj;
    end;
end;

end.
```