

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Marika Sobotková



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ZAŘÍZENÍ PRO AUDIOSTIMULACI

AUDIOSTIMULATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Marika Sobotková

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Sekora

BRNO 2016

Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Studentka: Marika Sobotková

ID: 164997

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Zařízení pro audiostimulaci

POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s principem audiostimulace s časováním využívané při neurologickém měření. 2) Navrhněte autonomní zařízení pro přehrávání stimulačních audiozáznamů v reálném čase. 3) Navrhněte protokol pro externí řízení zařízení (z počítače) tak, aby byl umožněn výběr záznamu, funkce "stop", "play", "pause" 4) Realizujte navržené zařízení. Zařízení musí umožňovat přehrávání audiozáznamů v kvalitě $f_{vz} = 44,1$ kHz, 16 bitů, stereo. 5) V rámci vyšetření ověřte funkci zařízení.

DOPORUČENÁ LITERATURA:

[1] CLARKSON, Peter M. Signal processing methods for audio, images and telecommunications. London: Academic Press, c1995, xvi, 436 s. Signal processing and its applications. ISBN 0121757900.

[2] PAXAL, Vendela Maria. Advanced digital signal processing algorithms for high speed applications: theory, design, and implementation. Trondheim: Norges Teknisk-Naturvitenskapelige Universitet, 1996, 80 s. Engineering. ISBN 00-294-6253-3.

Termín zadání: 8.2.2016

Termín odevzdání: 27.5.2016

Vedoucí práce: Ing. Jiří Sekora

Konzultant bakalářské práce: Ing. Martin Lamoš

prof. Ing. Ivo Provazník, Ph.D., předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem této bakalářské práce je návrh a realizace zařízení pro audiostimulaci, které bude využíváno v rámci vyšetření funkční magnetickou rezonancí, a seznámení s problematikou magnetické rezonance a stimulace. Celé zařízení je řízeno mikrokontrolérem z open-source platformy Arduino, modelová řada Due. Hardwarová realizace je provedena na nepájivém kontaktním poli, součástí je i návrh DPS modulu. Řídící část mikrokontroléru je reprezentována vývojovými diagramy. Algoritmus je navržen a realizován v jazyce vycházející z knihovny wiring náležející Arduino. Mikrokontrolér je řízen externě pomocí grafického uživatelského rozhraní vytvořeného v prostředí MATLAB.

KLÍČOVÁ SLOVA

audiostimulace, magnetická rezonance, Arduino, hardware, firmware, GUI

ABSTRACT

The aim of this bachelor thesis is to design a device for audio stimulation, which will be used during examination with functional magnetic resonance. The next aim is to familiarize readers with magnetic resonance and stimulation. Whole device is controlled by microcontroller from open-source platform Arduino, model Due. Hardware is based on a non-soldering contact field. There is also draft of printed circuit board. Software is represented by flowcharts. Algorithm is designed and realized in wiring Arduino's library (which belongs to Arduino). Microcontroller is controlled by graphical user interface made by MATLAB.

KEYWORDS

audio stimulation, magnetic resonance, Arduino, hardware, firmware, GUI

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Zařízení pro audiostimulaci“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autorky

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu bakalářské práce panu Ing. Jiřímu Sekorovi za odborné vedení, konzultace, trpělivost, podnětné návrhy k práci i vždy dobrou náladu. Také bych chtěla poděkovat Ing. Martinu Lamošovi za cenné informace z oblasti magnetické rezonance a audiostimulace.

Brno

.....

podpis autorky

OBSAH

Úvod	9
1 Magnetická rezonance a audiostimulace	10
1.1 Historie zkoumání mozku	10
1.2 Magnetická rezonance	11
1.2.1 Diagnostická metoda	11
1.2.2 Princip magnetické rezonance	11
1.2.3 Funkční magnetická rezonance	12
1.3 Stimulace	14
2 Návrh a realizace zařízení	15
2.1 Blokové schéma	15
2.2 Hardware	17
2.2.1 Hardwarová realizace	17
2.2.2 Deska plošných spojů	19
2.2.3 Způsob připojení k počítači	19
2.3 Firmware mikrokontroléru	21
2.3.1 Návrh řídicí části	21
2.3.2 Realizace řídicí části	25
2.4 Software grafického uživatelského rozhraní	29
2.4.1 Vývojový diagram grafického uživatelského rozhraní	29
2.5 Protokol pro externí řízení	32
3 Testování zařízení	34
Závěr	35
Literatura	36
Seznam symbolů, veličin a zkratk	38
A Schéma zapojení	40
B Seznam použitých součástek	41
C Návrh desky plošných spojů	42
D Obsah přiloženého CD	44

SEZNAM OBRÁZKŮ

2.1	Blokové schéma zařízení pro audiostimulaci	15
2.2	Vývojový diagram řídicího programu	22
2.3	Vývojový diagram kontroly SD karty	23
2.4	Vývojový diagram funkce PLAY	24
2.5	Grafické uživatelské rozhraní ovládacího panelu	29
2.6	Vývojový diagram grafického uživatelského rozhraní ovládacího panelu	30
A.1	Elektrické schéma zapojení	40
C.1	Osazovací výkres	42
C.2	Návrh desky plošných spojů	43

SEZNAM TABULEK

2.1	Ekvivalentní spojení mezi LCD a Arduinem	19
2.2	Základní hodnoty pro funkce PLAY, PAUSE, CONTINUE a STOP . . .	32
2.3	Příklad protokolu	33
B.1	Seznam použitých součástek	41

ÚVOD

Tato práce se věnuje zařízení pro audiostimulaci, které bude využíváné v rámci vyšetření funkční magnetickou rezonancí. Zařízení bude umožňovat spustit a přehrát jakýkoli předem zvolený audiostimul, který bude uložen na SD, případně μ SD kartě¹, která je součástí zařízení. Na uvedeném paměťovém mediu bude nahráno cca 200 skladeb a podnětů, přičemž obsluha zařízení si vždy vybere ten, který aktuálně potřebuje.

Zařízení bude externě řízeno počítačem, ke kterému bude připojeno přes USB port. Vstupní data jsou v HiFi kvalitě, tedy vzorkována $f_{vz} = 44100\text{Hz}$, 16 bitů a v nekomprimovaném formátu *wav*. Výstupní audio signál bude zachován stereo-fonní a v HiFi kvalitě.

Zařízení tvoří dvě hlavní části. První částí je část hardwarová, druhou částí je řízení mikrokontroléru. Zařízení je pro účely testování a odladění realizováno na nepá-jivém kontaktním poli, součástí je i návrh DPS modulu. Řídící část je reprezentována vývojovými diagramy a následným popisem samotné realizace.

Důležitou částí návrhu celého zařízení je protokol pro externí řízení definující základní příkazy, kterými je samotné zařízení ovládáno. Příkazy vycházejí ze čtyř funkcí, které byly definovány jako požadavky vyšetřujícího pracoviště. Konkrétně jde o funkce PLAY, PAUSE, CONTINUE a STOP.

¹SD karta (z ang. *Secure Digital*) je paměťová karta na principu flash paměti.

1 MAGNETICKÁ REZONANCE A AUDIOSTIMULACE

1.1 Historie zkoumání mozku

Zájem člověka o jeho nejvyšší řídicí centrum, mozek, byl vždy zahalen tajemstvím a mýty. V průběhu času se proměňovaly jednotlivé teorie, zpřesňovala lokalizace jednotlivých center nebo se lépe vysvětloval jejich význam. Ovšem ani v dnešní době nevíme o mozku vše. Však již Platón¹ ve své době mozek i samotné lidské poznání přirovnával k jeskyni.

Tři tisíce let před naším letopočtem bylo za nejdůležitější orgán lidského těla považováno srdce. Podle tehdejších lidí a filosofů v srdci sídlila duše člověka, a proto bylo takto označováno. Egypťanům vydržel tento názor až do doby počátků antického Řecka, tj. až do 4. století, kdy již Řekové „přesunuli“ duši do mozku. V té době se lidé snažili pochopit vztah mezi mozkiem a chováním člověka.

V 19. století byla vynalezena metoda kranioskopie, která dává do vztahu osobnost, inteligenci a další lidské vlastnosti a kvality s rozměrem lebky, tedy i s velikostí mozku. V té době Franz Joseph Gall, zakladatel kranioskopie, napsal, že mozek je tvořen dvěma základními substancemi, a to bílou a šedou hmotou mozkovou. Tento výrok byl první krokem k anatomii a poznatkům o mozku, jaké máme dnes.

Paul Broca² jako první lokalizoval některé důležité funkce v mozku a díky tomu je označován jako „otec neurologie“. Zjistil, jak mozek chápe řeč a kde se nachází centrum pro porozumění řeči. Za veškeré zásluhy je toto místo označeno jako Brocovo centrum. V této době se neurovědy staly respektovanou disciplínou.[1]

V dnešní době jde výzkum mozku velmi rychle dopředu a to díky moderním vyšetřovacím metodám, jako je EEG a ECoG, a zobrazovacím metodám, MRI a fMRI, CT, PET a SPECT. Pořád ale je ale v mozku, hlavně v jeho funkčnosti, mnoho neznámého a trochu i tajemného.

¹Platón, řecký filozof, 427 př. n. l. – 347 př. n. l.

²1824-1880, francouzský chirurg, patolog a anatom

1.2 Magnetická rezonance

1.2.1 Diagnostická metoda

Magnetická rezonance patří mezi moderní zobrazovací metody. Dělí se na klasickou, kdy jsou zobrazeny pouze struktury v lidském těle, hlavně měkké tkáně, a funkční magnetickou rezonanci, kdy lze i nepřímo mapovat neuronální aktivitu.

Funkční magnetická rezonance je vyšetřovací metoda využívaná za účelem hodnocení funkce zobrazovaných struktur, nejčastěji mozku. Funkcí se v širším slova smyslu mohou myslet různé „dynamické“ nebo „fyziologické“ parametry vyšetřované tkáně, jako je lokalizovaná neuronální a synaptická aktivita, perfuze tkání, chemická aktivita tkání, funkční konektivita mozkových oblastí nebo obsazení receptorů neurotransmitery.[2]

1.2.2 Princip magnetické rezonance

Magnetická rezonance je jednou z technicky nejsložitějších v praxi využívaných zobrazovacích metod. Fyzikální princip metody vychází ze sledování odezvy jader s nenulovým magnetickým momentem po interakci s vysokofrekvenčním elektromagnetickým vlněním. Měřená osoba, tedy i jádra všech prvků obsažená v těle, jsou v rámci vyšetření umístěna v silném magnetickém poli, které způsobuje vychýlení z „klidové“ osy rotace spinových momentů jednotlivých jader po směru nebo proti směru působení magnetické indukce. V rámci magnetické rezonance se používají velmi silná stacionární magnetická pole. Hodnota magnetické indukce se pohybuje v řádu jednotek tesla. U přístrojů pro měření na lidech indukce dosahuje hodnot od 1 T do 7 T.[3]

Každý prvek má své určité charakteristiky, jak se chová v silném vnějším statickém magnetickém poli. Mezi hlavní charakteristiky patří Larmorova precese a Larmorova frekvence. Jev nazývaný Larmorova precese popisuje rotační, neboli precesní, pohyb vektoru magnetického momentu kolem vektoru magnetické indukce. Vektor magnetického momentu rotuje s Larmorovou frekvencí.

Působením krátkým radiofrekvenčním (dále RF) pulzem, který je vyvolán RF cívkami, se vektor magnetizace překlopí o úhel alfa. Alfa popisuje i směr RF pulzu. Podmínkou je, aby frekvence RF pulzu byla v rezonanci s frekvencí precesního pohybu daného jádra. Z toho je jasné, že frekvence RF pulzu se rovná Larmorově frekvenci. RF pulz trvá omezenou a známou dobu a po ukončení pulzu dojde k náhodnému rozfázování precesního pohybu kolem vektoru magnetické indukce stacionárního magnetického pole s postupným návratem do původního směru. Doba po ukončení působení RF pulzu se nazývá relaxace a je určena dvěma relaxačními časy T_1 a T_2 , které jsou charakteristické pro každou tkáň. Čas T_1 popisuje dobu, za kterou dojde k obnově 63 % magnetizace původního směru. Čas T_2 ukazuje dobu,

z kterou dojde k poklesu magnetizace po RF na 37 % svého maxima při RF. Existuje pravidlo, že čas T2 je vždy kratší než čas T1.

Pro kódování pozice nestačí homogenní stacionární magnetické pole. Z toho důvodu jsou do systému přidány gradientní cívkky, které produkují lineárně proměnné magnetické pole v ose x . Toto pole se nazývá gradientní pole. Působením gradientního pole se získá výsledný signál, který je superpozicí více signálů o různých Larmorových frekvencích, který je snímán. Po nasnímání se aplikuje Fourierova transformace (dále FT), která převádí signál z časové domény do frekvenční. Díky FT dojde k separaci výsledného signálu na elementární signály v závislosti na jejich prostorové pozici, což je nutné k vytvoření obrazu. Intenzita signálu má vliv na kontrast ve výsledném obraze.[4]

1.2.3 Funkční magnetická rezonance

Předpoklad, že aktivita neuronu je navázána na zásobení živinami, jejichž nositelem je krev, je využíván při funkční magnetické rezonanci (dále fMRI). fMRI mapuje neuronální aktivitu nepřímo přes lokální změny oxygenace a perfuze mozkové kůry. Při fMRI se využívají dva principy mapování mozku a to změna perfuze a objemu krve v místě vyšší neuronální aktivity nebo změna poměru okysličené a neokysličené krve. V dnešní době převažuje druhý princip mapování.

Zvýšená aktivita neuronů má za následek zvětšení spotřeby energie i kyslíku. Kyslík je přenášen hemoglobinem³. Pokud hemoglobin kyslík obsahuje, jedná se o oxyhemoglobin, bez kyslíku jde o deoxyhemoglobin. Oxyhemoglobin a deoxyhemoglobin mají jiné magnetické vlastnosti, což znamená, že se jinak chovají v magnetickém poli. Deoxyhemoglobin je paramagnetický⁴, na rozdíl od oxyhemoglobinu i většiny tkání lidského těla, a funguje tak jako přirozená kontrastní látka. Paramagnetické látky nepatrně zesilují magnetické pole. Přítomnost deoxyhemoglobinu má tedy za následek tvorbu mikroskopických gradientů v magnetickém poli, které umožňují odlišit oxyhemoglobin od deoxyhemoglobinu a tím i určit funkční aktivitu určité části mozku.[5]

Funkční magnetická rezonance má v sobě veliký potenciál jak v oblasti výzkumu, tak hlavně v medicínské oblasti. V dnešní době je fMRI v neurologii brána spíše jako doplňková metoda z důvodu složitého zpracování signálů a častých artefaktů. Budoucnost fMRI je ve snímání v reálném čase. Po umožnění tohoto typu snímání

³Hemoglobin je červené krevní barvivo, jehož funkcí je přenos kyslíku a oxidu uhličitého mezi plírcemi a tkáněmi. Jedná se o protein složený ze 4 podjednotek a jednou z hlavních složek hemoglobinu je železo.

⁴Látky se rozdělují na diamagnetické, paramagnetické a feromagnetické podle chování ve vnějším magnetickém poli

bude využitelnost fMRI rozšířena o možnou lepší diagnostiku epilepsie, Alzheimerovy choroby i poškození mozku po cévní mozkové příhodě.[5]

1.3 Stimulace

Možností stimulace mozku je několik - obrazová, zvuková i audiovizuální. V této práci je pod drobnohledem pouze stimulace zvuková, neboli audiostimulace, kdy jsou využívány pouze zvukové podněty. Cílem stimulace je aktivace určité části mozku a snímání její odezvy.

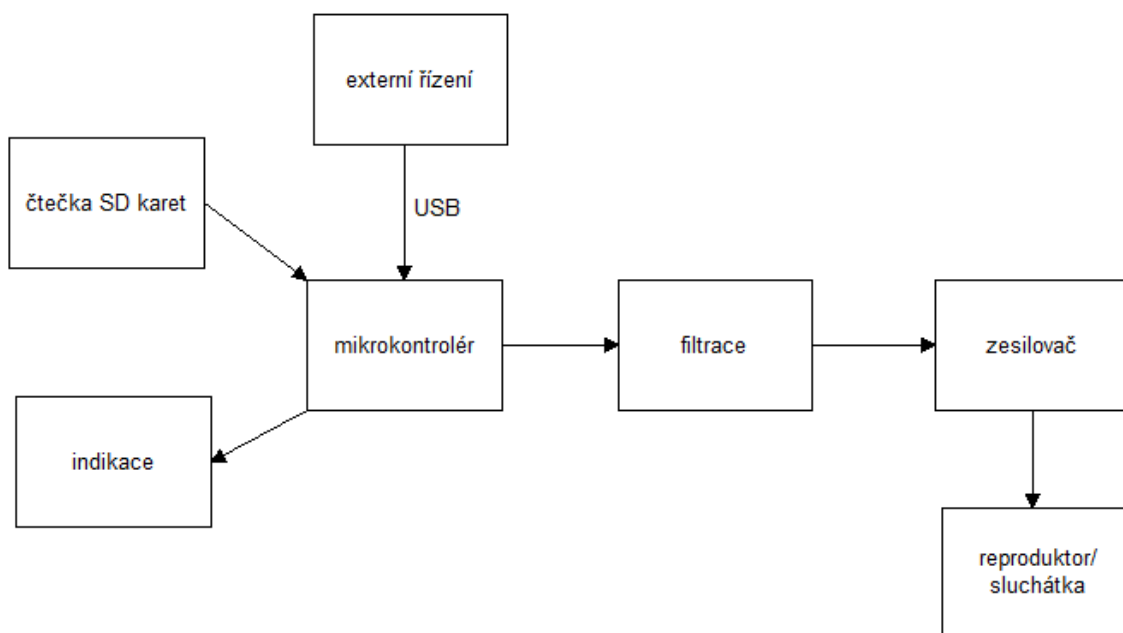
Při funkčním mapování mozku není žádná „základní úroveň“ neuronální aktivity, což komplikuje vyhodnocování a interpretaci výsledků fMRI. Z toho důvodu se při experimentech často střídají minimálně dvě stimulační podmínky, v elementárním případě klid a nějaké stimulace. Po nasnímaní lze děje porovnat a zhodnotit rozdíl mezi nimi. Pro každou stimulační podmínku se mozek snímá vícekrát.[6]

Stimulace je nejčastěji využívána neurology nebo výzkumnými pracovníky, kterým pomáhá odhalit nové poznatky o mozku. Při audiostimulaci se aktivuje primární sluchový kortex. Aktivace dalších sítí závisí na návrhu daného experimentu a na způsobu, jakým se daná audiostimulace využije. Například při pouštění různých typů hudby se sleduje vliv na emoční vnímání pacientů. Neurologové někdy využívají stimulaci jako doplňující vyšetření před neurochirurgickou operací v oblastech kritických pro řečové či motorické funkce. Vyšetření jim pomáhá k lepšímu prozkoumání těchto oblastí, díky kterému se mohou lépe těmito oblastem vyhnout nebo v ní být opatrnější.

2 NÁVRH A REALIZACE ZAŘÍZENÍ

2.1 Blokové schéma

Cílem blokového schématu je vytvoření myšlenkové mapy ukazující součásti hardwaru, návaznosti a spojení.



Obr. 2.1: Blokové schéma zařízení pro audiostimulaci

První blok je **čtečka SD karet**, která umožňuje čtení záznamů z SD karty. Na SD kartě jsou nahrány stimulační zvukové nahrávky. Používána je čtečka μ SD karet, která je součástí Arduino Ethernet Shieldu.

Nejdůležitějším blokem je **mikrokontrolér**. Mikrokontrolér byl vybrán z open-source platformy Arduino, modelová řada Due. V mikrokontroléru se provádí veškeré procesy včetně přehrávání a zastavení určitého souboru nebo i resetování.

Mikrokontrolér bude řízen blokem **externí řízení**. Řízením je ovládáno chování celého zařízení a slouží i k volbě skladby, která se má přehrát nebo zastavit. Řízení je jediným spojovacím článkem mezi uživatelem a samotným zařízením. Bloky mikrokontrolér a řízení jsou propojeny USB kabelem. Důvod využití právě USB rozhraní je uveden v části 2.2.3 na straně 19.

Indikace slouží pro indikaci stavu zařízení pomocí displeje.

Úprava vstupního signálu probíhá v bloku **filtrace**. Zde dochází k filtraci signálu a k převodu z digitálního na analogový. Z reproduktorů nebo sluchátek může vycházet pouze analogový signál. Z toho důvodu je nutný převod na D/A převodníku

(dále DAC).

Následuje blok **zesílení**, kde dochází ke konečné úpravě signálu, tj. k jeho zesílení. Zesilovač slouží k impedančnímu přizpůsobení výstupu pro připojenou zátěž, kterou v dané aplikaci představuje blok reproduktor/sluchátka

2.2 Hardware

První částí řešení je část hardwarová (dále HW). Do této části patří popis samotné HW realizace, návrh desky plošných spojů a způsob propojení HW se standardní periférií počítače.

2.2.1 Hardwarová realizace

Realizace elektrického obvodu je prováděna na nepájivém kontaktním poli. Nepájivé kontaktní pole bylo vybráno z důvodu snadné dostupnosti pole, jednoduchého zapojování součástek do obvodu a následné jednoduché výměny součástek při potřebě. Nepájivé pole zároveň eliminuje možnost případných chyb vzniklých např. při pájení obvodu.

Nejprve byl na nepájivém poli sestaven obvod pouze pro mono výstup, který sloužil pro kontrolu funkčnosti zařízení — kontrola funkcí PLAY, STOP, PAUSE a CONTINUE, kontrola správné funkčnosti LCD a součástek. V dalším kroku byl sestaven obvod pro stereo výstup. Finálním krokem realizace HW byl návrh desky plošných spojů pro stereo výstup, který je uveden v části 2.2.2 na straně 19.

Výběr mikrokontroléru

Prvotním krokem realizace, po seznámení se s problematikou audiostimulace při fMRI, byl výběr mikrokontroléru z modelových řad Arduino. Jako první byl vybrána modelová řada UNO. Jak se později ukázalo, součástí řady UNO je mikrokontrolér neumožňující přehrávání zvukových stimulů v požadované kvalitě[8], tedy $f_{vz} = 44100\text{Hz}$, 16 bitů a stereo. Z tohoto důvodu byla vybrána jiná, vyšší, řada Arduina, kontrétně Due. Due má v sobě zabudovaný 32bitový mikrokontrolér Atmel ARM Cortex, díky kterému je možné dosáhnout požadované (Hi-Fi) kvality. Druhým důvodem, proč byla vybrána řada Due, byly knihovny, které jsou součástí programu Arduino, a slouží k ovládání připojeného mikrokontroléru z řad Arduino. Pro práci se zvukovými záznamy pomocí Arduina se běžně využívají dvě knihovny a to *Audio* a *TMRpcm*. Knihovnu TMRpcm nebylo možné použít z důvodu schopnosti přehrávat záznamy pouze nižší kvality (8 bitů, $f_{vz} = 32000\text{Hz}$ a mono[7]) než jaká je požadována. Knihovna *TMRpcm* není součástí Arduina a je nutná její instalace. Knihovna *Audio* umožňuje přehrávat záznamy požadované kvality, ale je kompatibilní pouze s řadou Due.[9]

Součástí mikrokontroléru Arduino Due jsou 2 USB rozhraní nazvaná „Native USB Port“ a „Programming Port“¹. Pro práci s mikrokontrolérem a pro programování je možné využít oba piny, doporučován je „Programming Port“. Před pro-

¹ „Programming Port“ se nachází blíže k analogovým pinům označeným „A0—A11“

gramováním mikrokontroléru pomocí programu Arduino je nutné v programu zvolit vybranou modelovou řadu a v tomto případě i vybraný port.[10]

Zesílení a filtrace

Základním prvkem zapojení je nízkofrekvenční audio zesilovač, konkrétně typ LM386. Z důvodu požadovaného stereo výstupu jsou použity dva stejné zesilovače. Napájení zesilovače je řešeno společným napájecím napětím o velikosti 5 V, které je získáno z výstupu modulu Arduina. K výhodám tohoto zesilovače patří malé zkreslení (do 0,2 %). Díky nízkému výkonu není třeba zesilovač externě chladit, proto je často používán v přenosných audio přehrávačích. Parametry zvoleného zesilovače jsou pro danou aplikaci dostačující.

Připojením dalších prvků do obvodu lze měnit hodnotu zesílení. V tomto případě bylo zvoleno zesílení 20, které je dostačující. Jednoduchou změnou zpětné vazby lze v případě potřeby zvýšit zesílení až na hodnotu $A = 200$. [11]

Další důležitou částí je derivační RC článek, který je napojen na výstup z Arduina označený jako DAC. DAC piny označené DAC0, DAC1 slouží k vytvoření zvukového výstupu při použití Arduino knihovny *Audio*. [10] Derivační RC článek plní funkce horní propusti a odstraňuje stejnosměrnou složku signálu. Na samotný derivační článek je napojen zesilovač a na zesilovač reproduktor/sluchátka. Požadovaný je stereo výstup, a proto jsou použity dva derivační články, dva stejné zesilovače i dva reproduktory (případně sluchátka). Využívány jsou také oba DAC piny. Existence dvou nezávislých DAC pinů je další důvod k výběru mikrokontroléru Arduino Due.

Čtečka SD karet

Další důležitou částí je čtečka SD karet. Je použita čtečka μ SD karet, která je součástí Arduino Ethernet Shieldu. Pro přenos informací mezi čtečkou — Ethernet Shieldem a samotným Arduinem slouží Arduino pin číslo 4 a pin Ethernet Shieldu označený jako *SDCS*. Ethernet Shield je také napájen společným napájecím napětím 5 V. [12].

Indikace

Jako prvek sloužící k indikaci byl vybrán displej (dále LCD). Pro přenos dat jsou využity digitální piny 2, 3, 5, 6, 11, 12. Pin 4 je vynechán, protože funguje jako CS pin při komunikaci přes SPI s externím zařízením. Pro větší přehlednost jsou v tabulce 2.1 uvedeny ekvivalentní spojení mezi LCD a Arduinem. [13]

K pinu VO je připojen potenciometr, slouží k nastavování požadovaného jasu LCD.

Elektrické schéma zapojení, které je zde popsáno, je v příloze A na straně 40.

Tab. 2.1: Ekvivalentní spojení mezi LCD a Arduinem

LCD pin	Arduino pin
VSS	GND
VDD	5V
RS	12
RW	GND
E	11
D4	6
D5	5
D6	3
D7	2
A	5V
K	GND

2.2.2 Deska plošných spojů

Byl vytvořen návrh jednostranné desky plošných spojů (dále DPS) o rozměrech 100 mm×36 mm. Místo propojky byl použit rezistor $R3$ s hodnotou odporu 0Ω .

Osazovací výkres je na obrázku C.1 na straně 42. Návrh desky plošných spojů je na obrázku C.2 na straně 43. Také je na přiloženém CD.

Návrh plošného spoje byl vytvořen v programu EAGLE, verze 7.5.0.

2.2.3 Způsob připojení k počítači

Existuje více možných způsobů propojení HW se standardní periférií počítače. Je možné drátové i bezdrátové propojení. V tomto případě bylo vybráno drátové propojení. Bezdrátové, například WIFI nebo BLUETOOTH, nezaručuje přijetí všech odeslaných dat a je zde také možnost výpadků během přenosu signálu. V rámci drátového propojení je možné využít například USB rozhraní nebo Ethernet rozhraní, které by se nabízelo díky použitému Arduino Ethernet Shieldu obsahující právě Ethernet rozhraní. Ethernet rozhraní využívá dva protokoly, a to UDP a TCP. U UDP není zajištěno, že se veškerá odeslaná data dostanou ke koncovému uživateli, je však upřednostněna rychlost (přímota) doručení. S tímto nebývá u TCP protokolu problém, poněvadž tento kontroluje plné doručení všech datagramů, ale nedostatkem je nekonstantní zpoždění při přenosu dat.

Požadavkem zadání je konstantní hodnota zpoždění mezi požadavkem na spuštění vybraného audio podnětu a reakcí zařízení v podobě jeho spuštění. Z výše uvedených důvodů je jasné, že Ethernet rozhraní neumožňuje zaručení konstantnosti

zpoždění, a proto je využito USB rozhraní. USB rozhraní zajišťuje odeslání i přijetí všech dat ve stejném pořadí, v jakém byla odeslána, a s konstantním zpožděním. USB rozhraní je součástí modelu Due.

2.3 Firmware mikrokontroléru

Další částí řešení je firmware mikrokontroléru. V této části je řešena řídicí část, která je popsána vývojovým diagramem, na který navazuje samotný program.

2.3.1 Návrh řídicí části

Úkolem řídicí části je řídit a ovládat celé zařízení. Složitost a postup řídicího programu je uveden ve vývojových diagramech.

Na obrázku 2.2 na straně 22 je vývojový diagram popisující celý algoritmus.

Prvním blokem vývojového diagramu je blok **start**. Start označuje začátek samotného programu, kterému předchází načtení Arduino knihoven *SD*, *SPI*, *Audio* a *LiquidCrystal*.

Následují kroky kontroly SD karty, které jsou zobrazeny na obrázku 2.3 na straně 23.

Kontrola SD karty začíná blokem **!SD.begin(4)**, jehož funkcí je inicializace SD karty. Inicializace je provedena přes negaci, která je označena interpunkčním znaménkem „!“. Díky negaci je jako pozitivní výsledek označena chybná inicializace, která program vrátí před inicializací a inicializace pokračuje znovu až do negativního výsledku, kdy bude program pokračovat dalším krokem. Číslo 4 označuje pin, který slouží ke spojení se čtečkou SD karet.

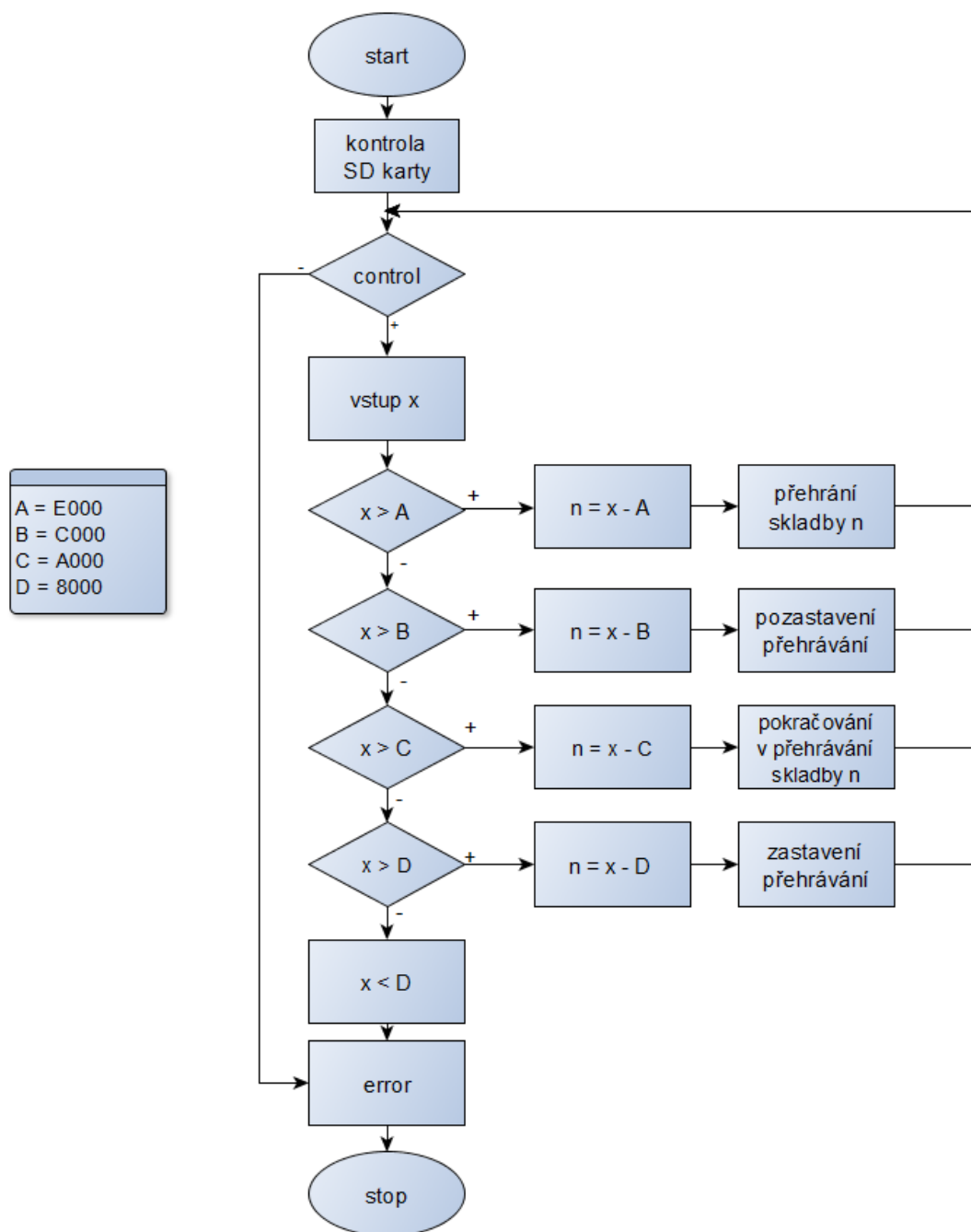
Dalším krokem je **for cyklus** pro i z oboru přirozených čísel od 1 do počtu skladeb, v tomto případě do 200, protože není přesně znám počet skladeb. Je znám pouze fakt, že jich bude maximálně 200.

Hodnota i vycházející z *for* cyklu určuje parametr následujícího bloku, jehož funkcí je kontrola existence dané skladby i ve formátu *wav*. Pokud skladba $i.wav$ existuje, do pomocné proměnné *control* se uloží číslo 1, v opačném případě 0.

Popis vývojového diagramu pokračuje popisem obecného diagramu na obrázku 2.2. V bloku označující **while** cyklus proměnné *control* je vnořena převážná část programu. V případě, že hodnota proměnné *control* je rovna jedné, program pokračuje do přerušení. Díky hodnotě 1 se *while* stává nekonečnou smyčkou, dokud nenastane změna v proměnné nebo nedojde k nějakému jinému zastavení. Za okolnosti, kdy se do podmínky dostane nula, program končí.

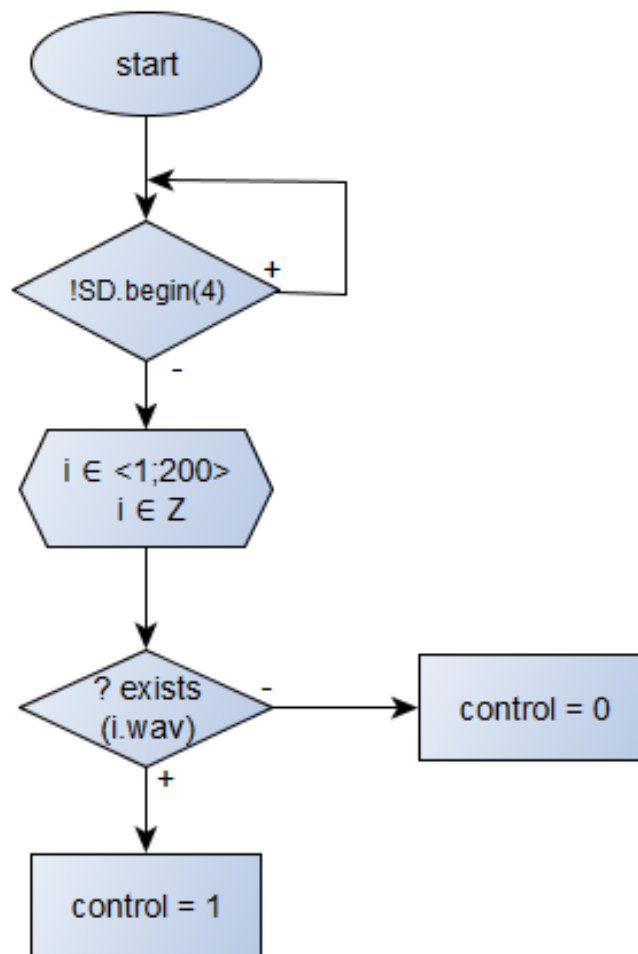
Blok **vstup X** je místem, kde se zadává 16bitové číslo v hexadecimální soustavě. V zadaném čísle bude zakódováno číslo skladby a funkce, která má nastat.

Funkce, která má nastat, se zjistí pomocí porovnání zadaného čísla x s číslem, které funguje jako indentifikátor pro danou funkci. Čísla jsou v hexadecimální soustavě a pro zjednodušení jsou nahrazena písmeny A, B, C a D. Pokud platí nerovnice



Obr. 2.2: Vývojový diagram řídicího programu

$x > A$, kde $A = E000$, dojde k nastavení programu na přehrávání. V případě platnosti nerovnice $x > B$ za předpokladu, že $B = C000$, program se přesune do části pozastavení přehrávání. Po pozastavení je někdy nutné i pokračování přehrávání a toto je zakomponováno v $x > C$, kde $C = A000$. Pro zastavení přehrávání je požadavek $x > D$, přičemž $D = 8000$. Porovnání jde postupně od nejvyššího čísla A po číslo



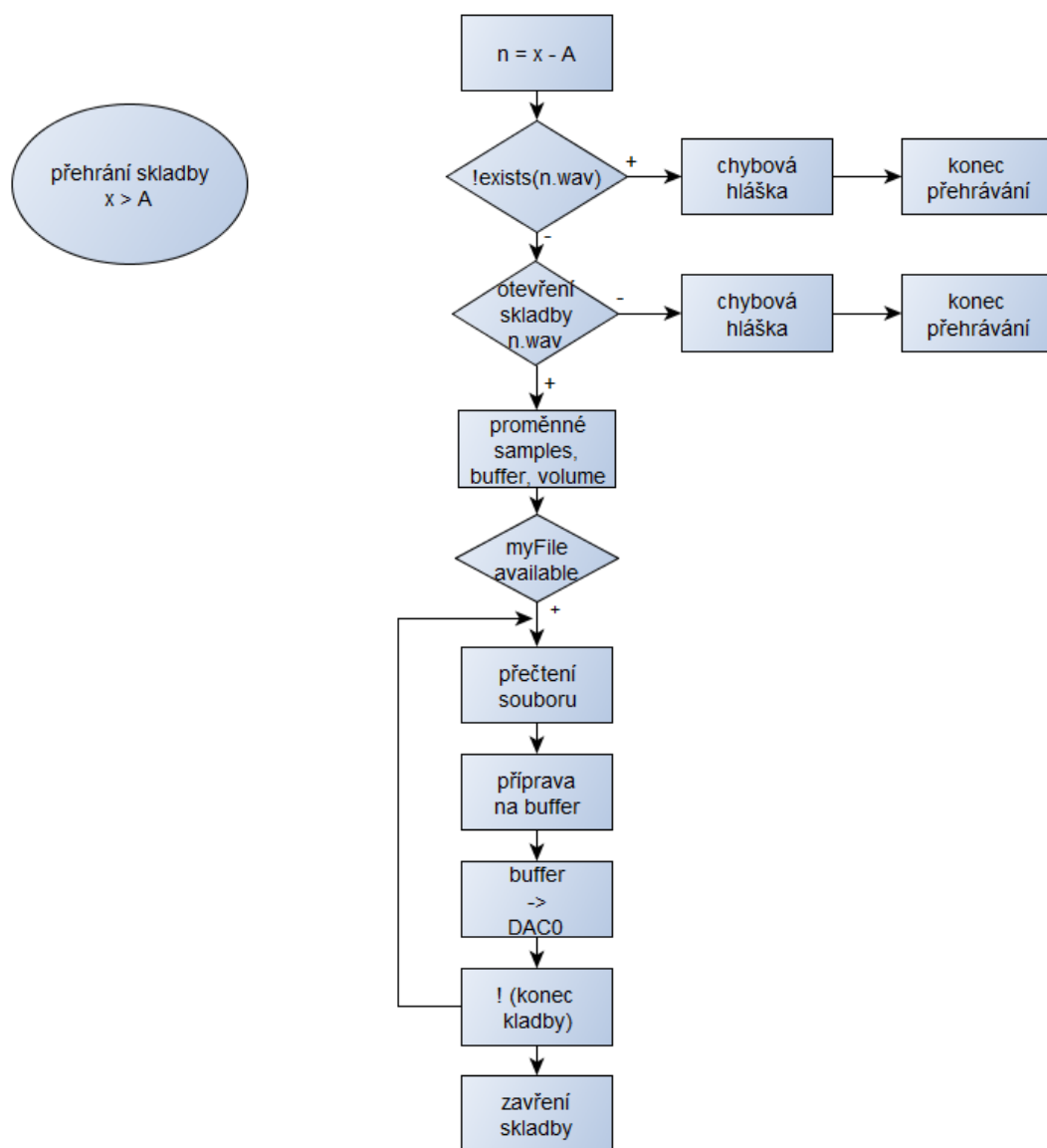
Obr. 2.3: Vývojový diagram kontroly SD karty

s nejmenší hodnotou D , což zajišťuje přiřazení ke správné funkci. Číslo skladby n se získá odečtením čísla popisujícího danou správně vybranou funkci od zadaného čísla x .

Na obrázku 2.4 na straně 24 je vývojový diagram² ukazující princip a způsob přehrávání vybrané skladby. Jak již bylo uvedeno, pro začátek přehrávání je nutné splnění podmínky $x > A$.

Prvním krokem je zjištění, která skladba se má přehrávat. Číslo n , které je identifikátorem každé skladby, se získá odečtením čísla A připadající na funkci PLAY od zadaného čísla x , tedy $(n = x - A)$.

²Ve vývojovém diagramu byly dlouhé názvy proměnných nahrazeny pouze jednotlivými znaky— x odpovídá proměnné InsertedNumber, n odpovídá proměnné TrackNumber a $n.wav$ odpovídá proměnné TrackTitle.wav.



Obr. 2.4: Vývojový diagram funkce PLAY

Následuje **kontrola**, zda soubor *n.wav* existuje. Kontrola je provedena přes negaci, což znamená, že pozitivní výsledek způsobí zobrazení chybového stavu a konec přípravy na přehrávání. Při negativním výsledku, tedy existenci skladby, algoritmus pokračuje dále.

Další příkaz slouží k **otevření** dané skladby *n*. Pokud se při tomto procesu vyskytne nějaká chyba a nedojde ke správnému otevření dané skladby, dojde opět k ukončení přehrávání a k zobrazení chybového stavu informující, jaký se vyskytl problém. V opačném případě pokračuje proces přehrávání uložením otevřené skladby

do proměnné *myFile*.

V dalším kroku jsou definovány **proměnné** *samples*, *buffer* a *volume*. Proměnná *buffer* slouží jako vyrovnávací paměť pro vyrovnání rozdílu mezi rychlostí přijímání dat a rychlosti jejich zpracování. Ve vyrovnávací paměti se zpracovává proměnná *samples*, jejíž hodnota je 1024 vzorků. *Samples* vyjadřuje počet vzorků, které mohou být najednou zpracovány. Hodnota proměnné *volume* je 1023.

Následuje **while** cyklus, jehož podmínkou je funkce *myFile.available*. Tento cyklus a jeho funkce podmiňuje to, že následující kroky algoritmu budou pokračovat do té doby, než skončí právě přehrávaná skladba *n*. Prvním krokem ve while cyklu je **přečtení souboru**, které je zakódováno ve funkci *Audio.read* s proměnnými *buffer* a jeho velikostí. Po přečtení následuje „**příprava na buffer**“, která je zapsána funkcí *Audio.prepare*. Tato funkce připravuje vzorky audio souboru do bufferu a nastavuje hlasitost. Z tohoto je jasné, že funkce *Audio.prepare* má 3 proměnné a to *buffer*, *samples* a *volume*.^[14]

Závěrečným krokem přehrávání je blok označený jako „**buffer->DAC0**“, který slouží pro přepsání zvukového signálu z buffru do digitálně-analogového převodníku. Tato funkce je zakomponována v *Audio.write*.^[15]

Posledním krokem je zavření právě přehrané skladby.

2.3.2 Realizace řídicí části

Realizace řídicí části navazuje na návrh řídicí části. Je tedy postavena na výše uvedených vývojových digramech.

Při sestavování řídicího softwaru mikrokontroléru Arduino je nutné definovat a načíst používané knihovny usnadňující práci např. s SD kartou nebo SPI zařízením. V tomto případě byly načteny 4 knihovny a to knihovna *SD*, *SPI*, *Audio* a *LiquidCrystal*. Knihovna *SD* slouží k ovládání čtečky SD karet, *SPI* pro ovládání externích zařízení propojených s mikrokontrolérem pomocí SPI, *Audio* pro načítání zvukových záznamů a jejich přehrávání, a *LiquidCrystal* pro ovládání LCD. Nezbytným počátečním krokem je také vyladění výstupu na 9600 baudů za sekundu pomocí příkazu *Serial.begin*^[16]. Nutná je také inicializace *Audio* knihovny pomocí *Audio.begin*, jejíž parametrem je dvojnásobek vzorkovacího kmitočtu, tedy 88200, a velikost vyrovnávací paměti^[17], a definování pinů propojených s LCD.

Dalším krokem byly kontroly skladeb uložených na SD kartě a inicializace samotné SD karty. Příkaz *SD.begin* kontroluje, zda je ve čtečce vložena SD karta, a zda je čtečka připojena ke správnému CS pinu. Pokud je vše v pořádku, program pokračuje, pokud ne, zobrazí se chybový stav a program se zastaví. Kontrola skladeb je prováděna pomocí for cyklu a příkazu *SD.exists*.

Z Arduino knihovny *SD* jsou využívány dva příkazy, konkrétně *SD.exists* a *SD.open*. Tyto příkazy fungují pouze v případě, že vstupní proměnná funkce je ve formátu *string*. Bylo by možné jednoduše napsat číslo dané skladby a její formát *wav* a toto celé uložit do proměnné typu *string*. V tomto případě by ale nebylo možné využít for cyklus pro kontrolu existence a správnosti skladeb na SD kartě. Zároveň by se také nedalo zjistit číslo skladby odečtením zadaného čísla od čísla definující danou funkci, protože tyto proměnné jsou uloženy ve formátu *integer*. Z tohoto důvodu se pracuje pouze s proměnnými typu *integer* a pouze před využitím výše uvedených dvou SD funkcí jsou převedeny do typu *string*. Dojde také k připojení koncovky *.wav*, aby se docílilo stejného zápisu, v jakém jsou skladby uloženy na SD kartě. Pro správnou funkci příkazů z knihovny *SD* je nutné zjistit formát SD karty a případně ji formátovat souborovým systémem FAT16 nebo FAT32.[18]

Knihovna *LiquidCrystal* slouží k ovládání LCD, konkrétně k zobrazování různých informací na displej. Byl vybrán LCD 16×2 znaků, tedy 2 řádky o 16 sloupcích. Nejprve bylo nutné definovat proměnnou *lcd*, která pak slouží k ovládání samotného displeje, a piny, se kterými je LCD propojen. Dále jsou definovány rozměry displeje pomocí příkazu *lcd.begin*. [19] Veškeré informace se zobrazují od levého horního rohu displeje, který má souřadnici [0,0]. Příkazem *lcd.setCursor* je možné změnit „počáteční“ bod, od kterého se zobrazují nové informace. Před nahráním nové informace je nutné displej vymazat pomocí příkazu *lcd.clear*, aby se informace nepřekrývaly. Tímto také dojde k nastavení pozice na výchozí hodnotu [0,0]. [20]

Nezbytnou součástí je také definování proměnných pro funkce, které byly definovány jako požadavky vyšetřujícího pracoviště. Jde o funkce *PLAY*, *PAUSE*, *CONTINUE* a *STOP*. Hodnoty těchto proměnných a postup, jak se těchto hodnot dosáhlo, je uveden v části 2.5 na straně 32.

Klíčová část programu je postavena na *while* cyklu, jehož součástí je definování proměnných daných funkcí a následně i samotné funkce. Pokud bude výsledek nějaké kontroly před *while* cyklem negativní, do proměnné *control* se uloží 0, v opačném případě číslo 1. Proměnná *control* je vstupním parametrem *while* cyklu. Pokud se tedy do *while* cyklu dostane hodnota proměnné 0, cyklus se zastaví a nedojde tedy k přehrání skladby ani k jakékoliv jiné funkci. V opačném případě, tedy pokud bude ve *while* cyklu hodnota 1, se cyklus stává nekonečným a „plnění“ příkazů algoritmu může pokračovat.

Pokud je v Arduinu nahrána určitá skladba a přehrávání není externě přerušeno např. posláním nového příkazu, dojde k opětovnému přehrávání dané skladby. Pro stejně kvalitní zvuk jako během prvního přehrávání je nutné vynulovat buffer. K vynulování bufferu lze použít tlačítko reset nacházející se na Arduino Due i na Arduino Ethernet Shieldu. Byla snaha o softwarové vynulování bufferu, ale z důvodu „nekompatibility“ datových typů to nakonec nebylo možné. Pokud v Arduinu žádná

skladba nahrána není, pouze se přehrává určitá skladba díky externímu ovládní, tlačítkem reset dojde k okamžitému přerušeni přehrávání a čekání na nový vstup.

Programové části

Níže je uvedena část kódu týkající se inicializace SD karty a kontroly typu skladby. Inicializace je prováděna příkazem *SD.begin(4)*, kde číslo 4 označuje linku pro výběr čipu. Inicializace je prováděna přes negaci označenou vykřičníkem. V případě chybné inicializace je do proměnné *control* typu integer uložena hodnota 0. Pokud inicializace proběhla v pořádku, program pokračuje na kontrolu typu skladeb.

```
Serial.print("Inicializace SD card...");
  if (!SD.begin(4)) {
    Serial.println("Inicializace selhala!");
    control = 0;
    return;
  }
  Serial.println("Inicializace dokončena.");
```

Kontrola typu skladeb začíná *for* cyklem od 1 do počtu skladeb uložených na SD kartě. Pro každé *Number* je v následující podmínce *if* provedena kontrola pomocí příkazu *SD.exists*. Je kontrolováno, zda daná skladba je v požadovaném typu *wav*. Pokud není, do předem definované proměnné *control* se uloží hodnota 0. V případě správného typu je uložena jednička jako hodnota proměnné *control*.

```
for(int Number = 1;Number<200;Number++) {
String Name = String(Number)+".wav";
  if (!SD.exists("Name.wav")) {
    control = 0;
  }
  else {
    control = 1;
  }
}
```

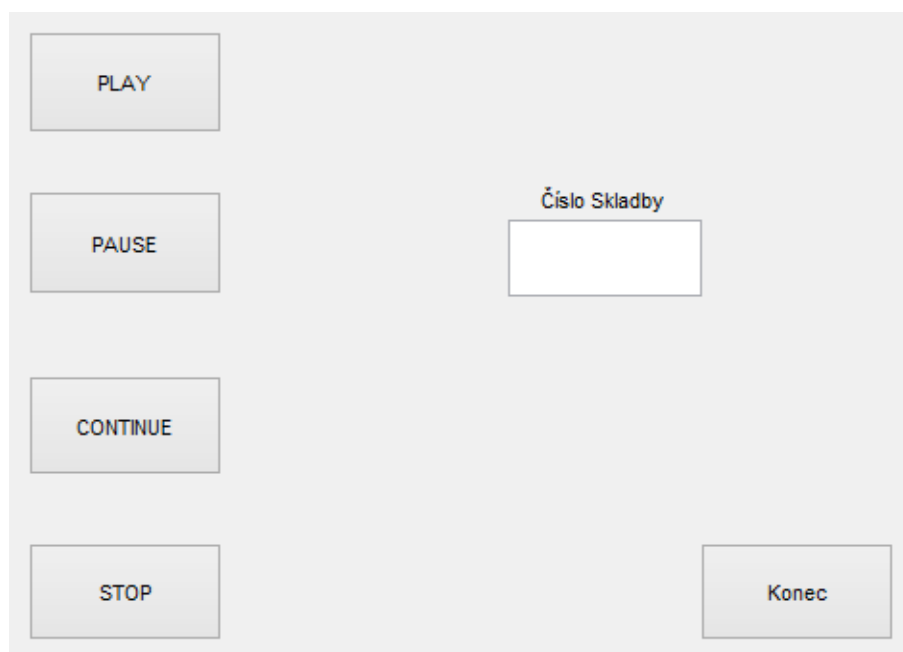
Níže je ukázka kódu pro přehrávání skladby. Přehrávání začíná podmínkou, ve které je porovnáváno zadané číslo *InsertedNumber* s *A*, což je hodnota pro funkci *PLAY*. Pokud je *InsertedNumber* větší než *A*, program pokračuje výpočtem čísla skladby a kontrolou existence dané skladby správného typu. Jestliže je vybraná

skladba nalezena, dochází k otevření skladby a k následným krokům nutným pro přehrávání. Proměnná *InsertedNumber* je typu integer, stejně tak proměnná *TrackNumber*. Před kontrolou a následným otevření vybrané skladby je nutné převést proměnnou *TrackNumber* do typu String a po přidání „koncovky“ *.wav* uložit do proměnné *TrackTitle* typu String. Audio podněty jsou typu *wav* a proto je nutné přidání „koncovky“ *.wav*

Podrobnější popis je uveden v části 2.3.1 na straně 23.

```
if (InsertedNumber>A); {
  TrackNumber = InsertedNumber-A;
  TrackTitle = String(TrackNumber)+".wav";
  if (!SD.exists("TrackTitle.wav")); {
    Serial.println("Skladba TrackTitle.wav nenalezena");
  }
  else {
    File myFile = SD.open("TrackTitle.wav");
    while (true) {
      samples = 1024;
      short buffer[samples];
      volume = 1023;
      while (myFile.available()) {
        myFile.read(buffer, sizeof(buffer));
        Audio.prepare(buffer, samples, volume);
        Audio.write(buffer, samples);
      }
    }
    myFile.close();
  }
}
```

2.4 Software grafického uživatelského rozhraní



Obr. 2.5: Grafické uživatelské rozhraní ovládacího panelu

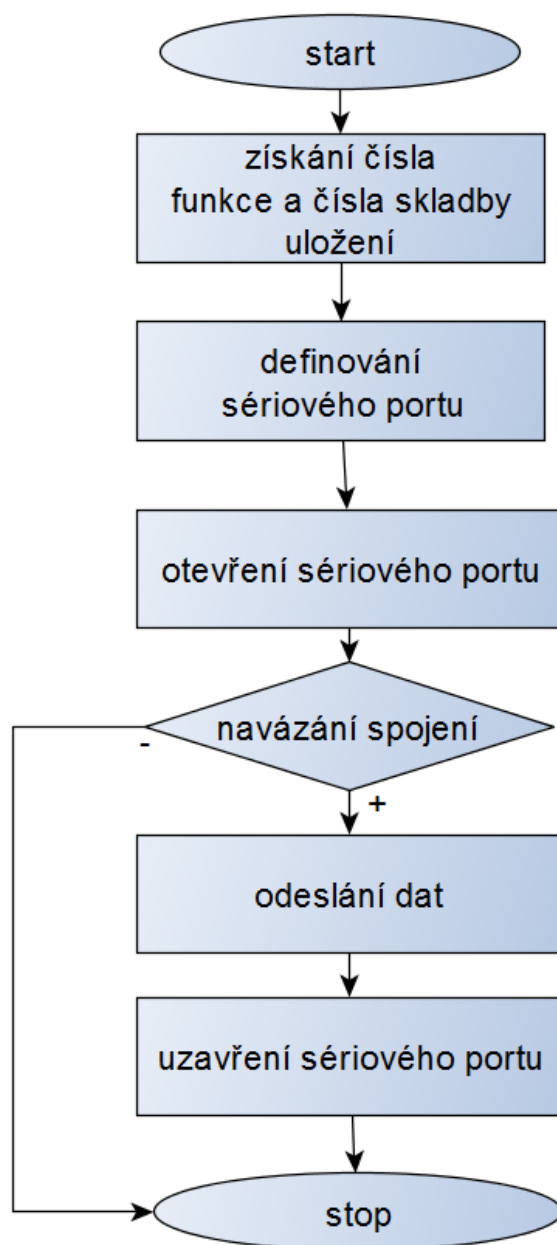
Pro požadovanou autonomnost zařízení bylo nutné ovládat zařízení a posílat příkazy z jiného „prostředí“ než je Arduino. K tomuto byl zvolen MATLAB[21]. V prostředí MATLAB bylo vytvořeno grafické uživatelské rozhraní (dále GUI), pomocí kterého je samotné zařízení ovládáno. Na obrázku 2.5 na straně 29 je náhled vytvořeného GUI.

Na obrázku je vidět 5 tlačítek, konkrétně *PLAY*, *PAUSE*, *CONTINUE*, *STOP* a *Konec*, a políčko umožňující zápis vybraného čísla skladby. Pro ovládání je nutné zapsat číslo vybrané skladby a stisknout tlačítko s názvem požadované funkce. Tlačítko *Konec* slouží k uzavření GUI.

2.4.1 Vývojový diagram grafického uživatelského rozhraní

Na obrázku 2.6 na straně 30 je vývojový diagram „tlačítek“ využívaných v GUI.

V rámci GUI jsou využívána 4 hlavní tlačítka a to *PLAY*, *PAUSE*, *CONTINUE* a *STOP*. Vývojový diagram je pouze jeden a to z toho důvodu, že všechny tlačítka pracují na stejném principu. Jediné obměny jsou v odesílaných datech do mikrokontroléru. Prvním krokem je získání čísla skladby zadaného uživatelem do konkrétního políčka a získání správného čísla funkce, které se určí stiskem tlačítka určující danou funkci. Tyto dvě hodnoty se uloží za sebe do datového řetězce v pořadí číslo



Obr. 2.6: Vývojový diagram grafického uživatelského rozhraní ovládacího panelu

funkce a číslo skladby. Dalším krokem je vytvoření globální proměnné *s*, do které se následně uloží parametry používaného portu pomocí příkazu *serial*. Je třeba zjistit, který sériový port (*com*) je vyhrazen pro daný mikrokontrolér³[22]. Následuje otevření sériového portu příkazem *fopen*, jehož parametrem je právě globální proměnná *s* se všemi uloženými parametry. Následuje *while* cyklus kontrolující navázání spojení s mikrokontrolérem. Pokud je spojení navázáno, následuje odeslání datového řetězce

³Lze zjistit příkazem „*a = arduino()*“

pomocí příkazu *fprintf*. Po odeslání řetězce do mikrokontroléru je ukončeno sériové spojení příkazem *fclose*.^[23] V mikrokontroléru dojde k sečtení přijatých čísel, čímž vnikne jedno číslo, se kterým poté pracuje podle pokynů uvedených v části 2.3.1 na straně 21.

Při vytváření GUI byly hodnoty zadávány v dekadické soustavě.

2.5 Protokol pro externí řízení

Protokol je určen pro externí řízení z počítače a definuje, jakým způsobem bude počítač (resp. obslužný program) zadávat povely hardwaru. V rámci řízení je použito 16bitové slovo, které je pro jednodušší popis rozděleno na 3 části.

První část tvoří pouze první bit, který bude mít vždy hodnotu jedna. Jedná se o úvodní bit (*start bit*), který určuje začátek nového příkazu .

Druhou část tvoří druhý a třetí bit. Těmito bity je určena funkce STOP, PLAY, PAUSE a CONTINUE. Pro přehrání skladby je použita dvojice bitů „11“, pro pozastavení přehrávání „10“, pro pokračování přehrávání „01“ a pro zastavení přehrávání „00“.

Třetí a poslední částí je 13 nižších bitů. Touto sérií bitů je již přímo určena skladba, která má být přehrána. Představuje tak binární hodnotu, která koresponduje s číslem skladby na paměťovém mediu.

Z toho důvodu, aby se vždy nemuselo zadávat celé 16bitové slovo, došlo k převodu čísel z binární do hexadecimální soustavy. Existují 4 možnosti funkce, co se se skladbou mohou stát. Proto existují i 4 základní hodnoty v hexadecimální soustavě, ze kterých se vychází. Vznikly převodem mezi soustavami s předpokladem, že není v systému žádná skladba. Základní hodnota pro PLAY je E000, pro PAUSE C000, pro CONTINUE A000 a pro poslední variantu STOP 8000. V rámci řízení je zadáno vždy jedno číslo, popisující vybranou funkci, navýšené o číslo skladby. Pro zjednodušení ovládání byla daná čísla nahrazena znakem, v tomto případě písmenem. „A“ je vyhrazeno pro funkci PLAY, „B“ pro funkci PAUSE, „C“ pro funkci CONTINUE a „D“ pro funkci STOP.

Pro přehlednost, a případné porovnání, jsou v tabulce 2.2 uvedeny základní hodnoty pro dané funkce v binární i hexadecimální soustavě. Součástí tabulky jsou také hodnoty funkcí v dekadické soustavě a to z toho důvodu, že zápis daných čísel v dekadické soustavě byl využíván v rámci GUI.

Tab. 2.2: Základní hodnoty pro funkce PLAY, PAUSE, CONTINUE a STOP

funkce	binární soustava	hexadecimální soustava	dekadická soustava
PLAY	1110 0000 0000 0000	E000	57344
PAUSE	1100 0000 0000 0000	C000	49152
CONTINUE	1010 0000 0000 0000	A000	40960
STOP	1000 0000 0000 0000	8000	32768

Tab. 2.3: Příklad protokolu

funkce	číslo skladby dekadická soustava	číslo skladby hexadecimální soustava	binární soustava	hexadecimální soustava
PLAY	21	15	1110 0000 0001 0101	E015
PLAY	54	36	1110 0000 0011 0110	E036
PLAY	166	A6	1110 0000 1010 0110	E0A6
PAUSE	21	15	1100 0000 0001 0101	C015
STOP	10	A	1000 0000 0000 1010	800A

Ve výše uvedené tabulce 2.3 je uveden příklad protokolu pro různé požadavky — tedy různé skladby a funkce. Pro přehlednost jsou uvedeny čísla vybraných skladeb v dekadické, ale i hexadecimální soustavě. Pro získání výsledné hodnoty v hexadecimální soustavě stačí pouze sečíst číslo vybrané skladby v hexadecimální soustavě se základní hodnotou funkce v hexadecimální soustavě uvedené v tabulce 2.2.

3 TESTOVÁNÍ ZAŘÍZENÍ

Poslední částí bakalářské práce bylo testování zařízení ve spolupráci se zadavatelem práce, výzkumným centrem CEITEC. Zařízení bylo otestováno pomocí grafického uživatelského rozhraní vytvořeného v prostředí MATLAB. Byla ověřena funkčnost a autonomnost zařízení a správnost navržených funkcí.

Zařízení je ovládáno příkazy posílanými přes sériový (com) port. Zařízení, podle požadavků, umožňuje výběr funkcí PLAY, STOP, PAUSE a CONTINUE a výběr konkrétní nahrávky uložené na SD kartě. Pro správnou funkčnost zařízení je nutná prodleva v délce přibližně 1 s, aby Matlab bezpečně otevřel daný port pro komunikaci s mikrokontrolérem Arduina.

S výzkumným centrem CEITEC bude probíhat další spolupráce spojená i s následným testováním. Konečným cílem spolupráce je kompatibilita navrženého zařízení se stimulačním softwarem E-prime, které CEITEC využívá v rámci vyšetření se stimulačními podněty. Přáním CEITEC je využívat jejich stimulační software, i když by se dal používat MATLAB, protože E-prime umožňuje vkládat do záznamu značky odpovídající začátku stimulačního podnětu, což je důležité pro následné zpracování a vyhodnocení dat.

Aktuálně je na CEITEC řešena implementace řídicího protokolu do prostředí E-prime a odstraňování problémů s komunikací E-prime — mikrokontrolér. V rámci implementace bylo testováno odstranění výše uvedené prodlevy, které ale nevede k řešení. Prodleva je patrně nutná, aby mikrokontrolér neztrácel příchozí data v době, kdy nemá port inicializován. Současně při delší prodlevě již patrně dochází k uzavření portu (tj. ukončení komunikace). Přestože samotná implementace nebyla součástí zadání práce, je nadále domluvena spolupráce, která povede k propojení zařízení bez použití GUI vytvořeného v prostředí MATLAB.

ZÁVĚR

Cílem bakalářské práce s názvem „Zařízení pro audiostimulaci“ bylo seznámení se s principem audiostimulace s časováním využívané při neurologickém měření. Tato část je popsána v kapitole 1 Magnetická rezonance a audiostimulace. Dalším cílem byl návrh a realizace samotného zařízení a návrh protokolu pro externí řízení zařízení počítačem. Tyto body jsou popsány v kapitole 2 Návrh a realizace zařízení. Posledním cílem bylo testování zařízení ve spolupráci se zadavatelem práce, výzkumným centrem CEITEC. Tento cíl je popsán v kapitole 3 Testování zařízení.

Pro ovládání zařízení jsou využívány čtyři funkce, které byly definovány jako požadavek vyšetřujícího pracoviště. Konkrétně jde o funkce PLAY pro přehrávání skladby, STOP pro zastavení přehrávání, PAUSE pro pozastavení přehrávání a funkci CONTINUE pro pokračování v přehrávání skladby od bodu pozastavení. Každé této funkci byla stanovena základní 16bitová hodnota, která klíčuje zvolenou funkci a pořadové číslo skladby.

Navržené zařízení je externě řízeno pomocí grafického uživatelského rozhraní vytvořeného v prostředí MATLAB. Zařízení, přesněji mikrokontrolér, je propojen s počítačem pomocí USB rozhraní. USB zaručuje odeslání a přijetí veškerých dat v pořadí, v jakém byla odeslána, a s požadovanou konstantní hodnotou zpoždění.

Zařízení pro audiostimulaci je postaveno na platformě Arduino Due s mikrokontrolérem Atmel ARM Cortex. Mezi další klíčové bloky zařízení patří čtečka SD karet, LCD, nízkofrekvenční audio zesilovač a příslušný filtr odstraňující stejnosměrnou složku signálu. Poslední dva zmíněné komponenty jsou zdvojeny ve stereo výstupu. Výstupem je stereofonní konektor typu JACK 3,5" umožňující připojení sluchátek. Zařízení je realizováno na nepájivém kontaktním poli. Součástí práce je návrh desky plošných spojů.

Výsledkem bakalářské práce je samostatně fungující zařízení pro audiostimulaci nezávislé na operačním systému, které umožňuje přehrávání audiozáznamů s konstantním zpožděním. Vzhledem k dosaženým výsledkům lze konstatovat, že zadání práce bylo splněno.

LITERATURA

- [1] TESAŘ, Michael *O psychologii.cz: Světové mozky o mozku* [online]. 2015, 27. 1. 2015 [cit. 2015-10-31]. Dostupné z URL: <<http://www.opsychologii.cz/clanek/146-svetove-mozky-o-mozku/>>.
- [2] HLUŠTÍK, Petr, David HORÁK, Roman HERZIG a Petr KAŇOVSKÝ. *Funkční zobrazování mozku pomocí magnetické rezonance v neurologii*. [pdf]. Neurologie pro praxi. 2008, 2008(9(2)) Dostupné z URL: <<http://www.solen.cz/pdfs/neu/2008/02/07.pdf>>.
- [3] NEZVAL, Jakub. *Nukleární magnetická rezonance: úvod do problematiky* [ppt]. 2015 [cit. 2015-10-06].
- [4] DOSTÁL, Marek. *Biomedicínská technika: Princip MR* [ppt]. 2015
- [5] CHLEBUS, Pavel, Michal MIKL, Milan BRÁZDIL a Petr KRUPA. *Funkční magnetická rezonance: Úvod do problematiky*. [pdf]. Neurologie pro praxi. 2008, 2005(3/2005): 6. Dostupné z URL: <http://fmri.mchmi.com/articles/chlebus_prehled.pdf>.
- [6] *fMRI experiment a jeho specifika* [online]. 2008, 11. 1. 2008 [cit. 2015-04-23]. Dostupné z URL: <http://fmri.mchmi.com/main_index.php?strana=15>.
- [7] *TMRh20/TMRpcm* [online]. GitHub, Inc., 2014 [cit. 2015-04-23]. Dostupné z URL: <<https://github.com/TMRh20/TMRpcm/wiki>>.
- [8] *Arduino UNO Genuino UNO. Arduino* [online]. 2016 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/main/arduinoBoardUno>>.
- [9] *Audio. Arduino* [online]. 2016 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/Audio>>.
- [10] *Arduino Due. Arduino* [online]. 2016 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/Main/ArduinoBoardDue>>.
- [11] *LM386 Datasheet (PDF) - National Semiconductor. Electronic Components Datasheet Search* [online]. 2000, 01 2000 [cit. 2015-12-28] Dostupné z URL: <<http://pdf1.alldatasheet.com/datasheet-pdf/view/8887/NSC/LM386.html>>.
- [12] *Arduino Ethernet Shield. Arduino* [online]. 2016 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/Main/ArduinoEthernetShield>>.

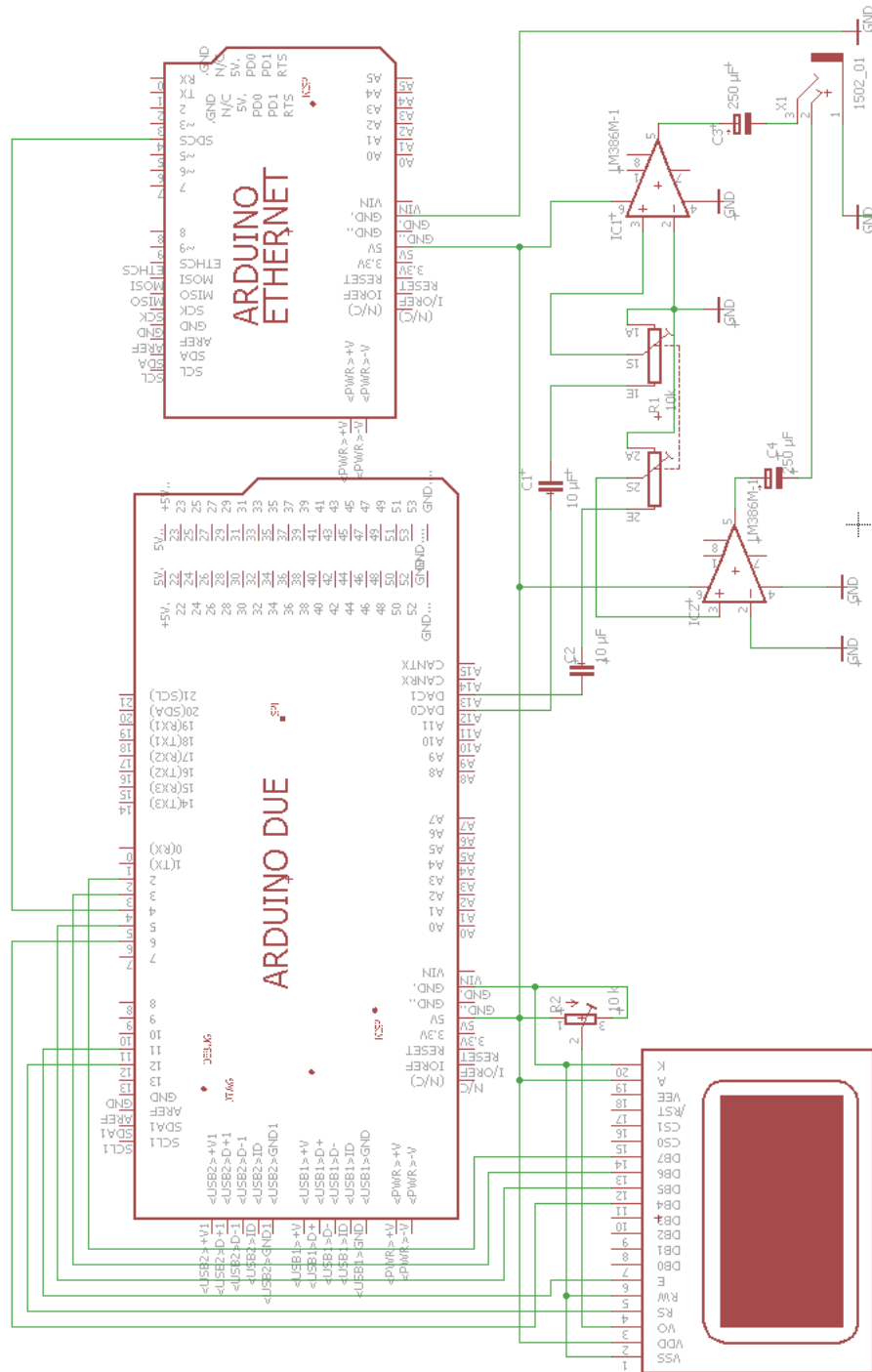
- [13] *"Hello World!"*. *Arduino* [online]. 2016 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/Tutorial/HelloWorld>>.
- [14] *AudioPrepare*. *Arduino* [online]. 2015 [cit. 2015-12-03]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/AudioPrepare>>.
- [15] *AudioWrite*. *Arduino* [online]. 2015 [cit. 2015-12-03]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/AudioWrite>>.
- [16] *SerialBegin*. *Arduino* [online]. 2016 [cit. 2016-05-15]. Dostupné z URL: <<https://www.arduino.cc/en/Serial/Begin>>.
- [17] *AudioBegin*. *Arduino* [online]. 2016 [cit. 2016-05-15]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/AudioBegin>>.
- [18] *SD Library*. *Arduino* [online]. 2015 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/SD>>.
- [19] *LiquidCrystal.begin*. *Arduino* [online]. 2016 [cit. 2015-05-19]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/LiquidCrystalBegin>>.
- [20] *LiquidCrystal Library*. *Arduino* [online]. 2015 [cit. 2015-04-23]. Dostupné z URL: <<https://www.arduino.cc/en/Reference/LiquidCrystal>>.
- [21] *Arduino Support from MATLAB* [online]. MathWorks, 2016 [cit. 2016-05-15]. Dostupné z URL: <<http://www.mathworks.com/hardware-support/arduino-matlab.html>>.
- [22] *Arduino Interface with MATLAB*. *All about Circuits* [online]. EETech Media, 2015 [cit. 2016-05-15]. Dostupné z URL: <<http://www.allaboutcircuits.com/projects/arduino-interface-with-matlab/>>.
- [23] *How To Send Data From MATLAB To Your Arduino*. *AllAboutEE* [online]. 2011 [cit. 2016-05-19]. Dostupné z URL: <<http://allaboutee.com/2011/07/04/how-to-send-data-from-matlab-to-your-arduino/>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

f_{vz}	vzorkovací kmitočet
EEG	Elektroencefalografie
ECoG	Elektrokortikografie
CT	Computed Tomography, počítačová tomografie
PET	Positron emission tomography, pozitronová emisní tomografie
SPECT	Single-Photon Emission Computed Tomography, jednofotonová emisní počítačová tomografie
UDP	User Datagram Protocol, síťový protokol
TCP	Transmission Control Protocol, síťový protokol
USB	Universal Serial Bus, sériová sběrnice počítače
DAC	Digital to Analog Convertor, digitálně analogový převodník

PŘÍLOHY

A SCHÉMA ZAPOJENÍ



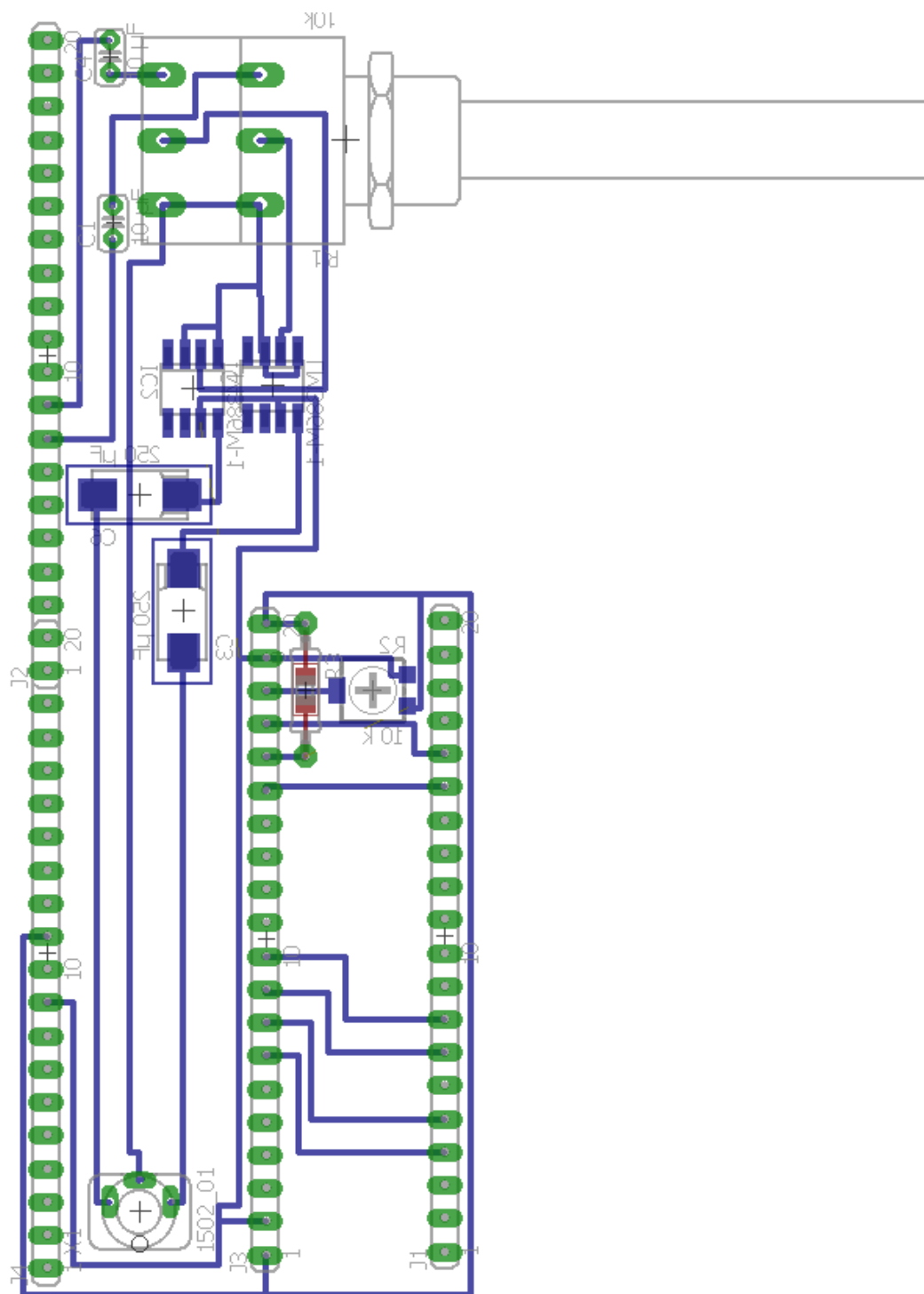
Obr. A.1: Elektrické schéma zapojení

B SEZNAM POUŽITÝCH SOUČÁSTEK

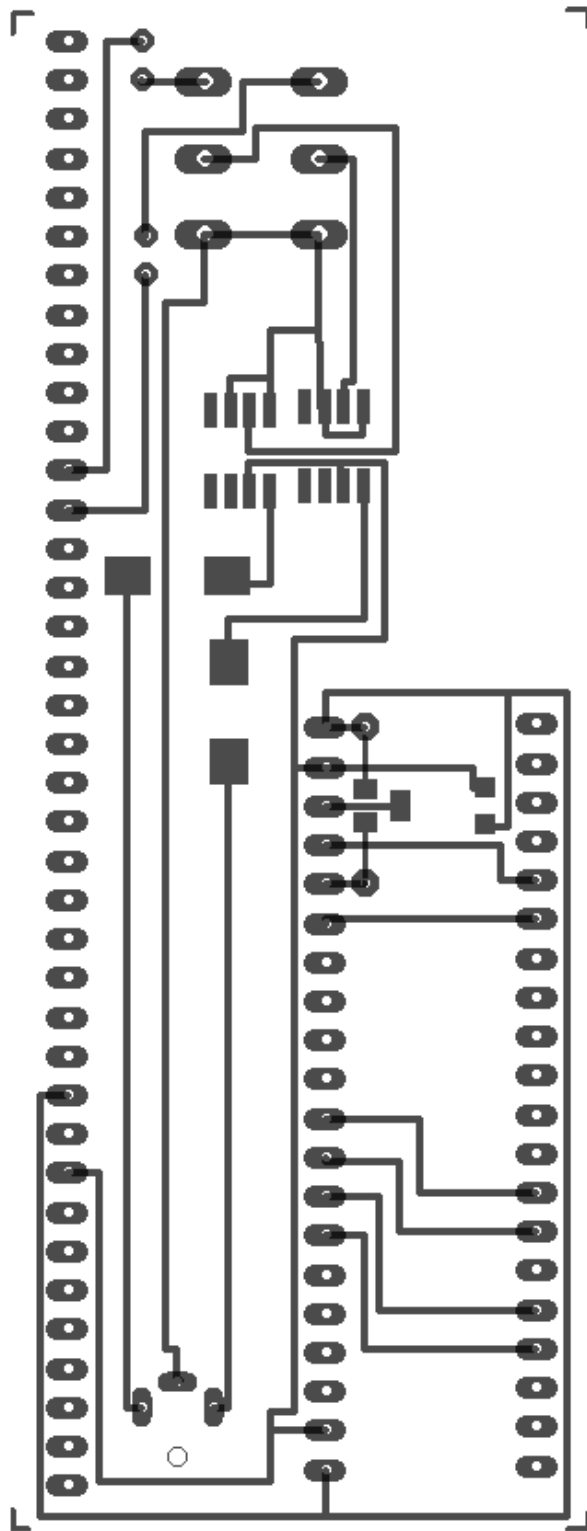
Tab. B.1: Seznam použitých součástek

součástka	hodnota	poznámka
IC1, IC2	LM386	zesilovač
C1, C2	$10\mu F$	16 V, elyt.
C3, C4	$220\mu F$	16 V, elyt
R1	$10k\Omega$	dvojitý potenciometr
R2	$10k\Omega$	potenciometr
X1	—	JACK 3,5"
R3	0Ω	drátová propojka, není ve schématu

C NÁVRH DESKY PLOŠNÝCH SPOJŮ



Obr. C.1: Osazovací výkres



Obr. C.2: Návrh desky plošných spojů

D OBSAH PŘILOŽENÉHO CD

Na přiloženém CD se nachází následující soubory:

- soubor s hlavním programem označený jako **HlavniProgram.ino**
- soubor s GUI označený jako **try_gui_1.m** a **try_gui_1.fig**
- elektrické schéma zapojení označené jako **SchémaZapojení.sch**
- návrh desky plošného spoje označený jako **DPS.sch** a **DSP.brd**

Hlavní program byl vytvořen v programu Arduino, verze 1.6.6.

Soubor s GUI byl vytvořen v programu MATLAB, verze R2015b.

Elektrické schéma zapojení a návrh desky plošných spojů byly vytvořeny v programu EAGLE, verze 7.5.0.