

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vacek** Jméno: **Kryštof** Osobní číslo: **507198**
Fakulta/ústav: **Fakulta informačních technologií**
Zadávací katedra/ústav: **Katedra počítačových systémů**
Studijní program: **Informatika**
Specializace: **Počítačové inženýrství**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Dálkově ovládaný analogový zesilovač zvuku ovládaný s pomocí IR ovladače

Název bakalářské práce anglicky:

Remote controlled analog sound amplifier using an IR controller

Pokyny pro vypracování:

- 1) Prozkoumejte existující řešení.
- 2) Analyzujte technologii dálkového ovládání s pomocí IR ovladače.
- 3) Navrhněte vlastní zařízení založené na mikrokontroleru ATmega.
- 4) Zařízení bude splňovat tyto požadavky:
 - Ovládání bude prováděno, jak pomocí IR dálkového ovládání, tak na samotném zařízení s pomocí rotačního enkodéru a tlačítek.
 - Zařízení bude zobrazovat aktuální stav a nastavení zesilovače.
 - Zařízení bude umožňovat výběr mezi různými vstupy.
- 5) Navržené řešení zrealizujte a řádně otestujte.

Seznam doporučené literatury:

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Kubalík, Ph.D. katedra číslicového návrhu FIT

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

Datum zadání bakalářské práce: **21.01.2024** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: _____

Ing. Pavel Kubalík, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Pavel Tvrđík, CSc.
podpis vedoucí(ho) ústavu/katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Bakalářská práce

**DÁLKOVĚ OVLÁDANÝ
ANALOGOVÝ
ZESILOVAČ ZVUKU
OVLÁDANÝ S POMOCÍ
IR OVLADAČE**

Kryštof Vacek

Fakulta informačních technologií
Katedra počítačových systémů
Vedoucí: Ing. Pavel Kubalík, Ph.D.
16. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Kryštof Vacek. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Vacek Kryštof. *Dálkově ovládaný analogový zesilovač zvuku ovládaný s pomocí IR ovladače*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratek	ix
1 Úvod	1
2 Existující řešení	2
2.1 Zesilovač LTC audio ATM6500BT	2
2.2 Zesilovač Fonestar AS-1515	2
2.3 Marantz PM6007	3
3 Teorie	4
3.1 Dělení zesilovačů	4
3.1.1 Zesilovač třídy A	4
3.1.2 Zesilovač třídy B	4
3.1.3 Zesilovač třídy AB	5
3.1.4 Zesilovač třídy C	5
3.2 Mikrokontrolér AVR	5
3.2.1 Základní Typy	6
3.2.2 Parametry ATmega32	6
3.3 Dálkové ovládání pomocí infračerveného světla	7
3.3.1 Princip fungování	7
3.3.2 Metody přenosu dat	7
3.3.3 Samotná komunikace	8
3.3.4 Protokoly	8
3.3.5 Protokol NEC	8
3.3.6 Protokol RC5	9
3.3.7 Komponenty infračerveného ovladače	10
3.3.8 Vlastnosti infračervené komunikace	10
3.3.9 Vývoj a budoucnost	10
3.4 Sběrnice SPI	10
3.4.1 Spi rozhraní mikrokontroléru ATmega32	11
3.5 Sběrnice I2C	12
3.5.1 Adresace	12
3.5.2 Princip přenosu	13
3.5.3 I2C rozhraní mikrokontroléru ATmega32	14

4	Analýza	15
4.1	Displej pro výpis hodnot	15
4.2	Ovládání zesilovače na předním panelu	17
4.2.1	Rotačního enkodér	17
4.2.2	Princip inkrementálního rotačního enkodéru	18
4.3	Ovládání audia	19
4.4	Koncový zesilovač	21
4.5	Rádio	22
5	Návrh Řešení	24
5.1	Deska zesilovače	24
5.2	Aplikace	24
6	Deska zesilovače	26
6.1	Čelní panel	26
6.2	Digitální část	28
6.3	Analogová část	29
6.4	Shrnutí	30
7	Aplikace	31
7.1	Atmel studio	32
7.2	Knihovna pro I2C komunikaci	32
7.3	Knihovna pro MAX7219	32
7.3.1	Konstruktor	33
7.3.2	MAX7219ClearDisplay	33
7.3.3	MAX7219UpdateNumber	33
7.3.4	MAX7219UpdateAll	33
7.4	Knihovna pro rotační enkodér	34
7.4.1	readEncoder	35
7.4.2	getValue	35
7.5	Knihovna pro RDA5807	35
7.6	Knihovna pro oled displej	36
7.6.1	clearFullDisplay	37
7.6.2	printText	37
7.7	Knihovna pro TDA7303	37
7.8	Knihovna pro IR přijímač	38
7.9	Nahrání kódu do mikrokontroléru	39
8	Testování	41
9	Závěr	43
A	GIT repozitář	44
	Obsah příloh	48

Seznam obrázků

3.1	Pracovní třídy zesilovače: a) třída A, b) třída AB, c) třída B, d) třída C [6]	5
3.2	Popis pinů mikrokontroléru ATmega32 [11]	7
3.3	Dekódování modulovaného záření	8
3.4	Kódování pomocí protokolu NEC [13]	8
3.5	Průběh NEC protokolu [13]	9
3.6	Kódování pomocí protokolu RC5 [14]	9
3.7	Průběh RC5 protokolu [14]	9
3.8	Způsob komunikace s více zařízeními pomocí výběru uzlu signálem SS [16]	11
3.9	Možné schéma zapojení [18]	12
3.10	Počátek I2C komunikace [17]	13
3.11	Průběh I2C komunikace [18]	13
4.1	Příklad zapojení dvouznakového displeje řízeného přímo mikrokontrolérem	15
4.2	Schéma zapojení čipu MAX7219	16
4.3	Rozdíl metí inkrementálním a absolutním enkodérem [24]	17
4.4	Princip enkodéru s kontakty	18
4.5	Princip optického enkodéru	18
4.6	Schéma zapojení digitálního potenciometru [26]	19
4.7	Schéma zapojení čipu LM1036 [28]	20
4.8	Schéma zapojení čipu TDA7303 [29]	21
4.9	Vlevo zapojení 2x30W a vpravo zapojení 4x10W [30]	22
4.10	Popis pinů modulu rrd-102 [32]	22
4.11	Struktura oled displeje [33]	23
5.1	Blokové schéma desky	25
6.1	Schéma zapojení enkodérů	26
6.2	Schéma zapojení modulu VS1838 vlevo a oled displeje vpravo	27
6.3	Schéma zapojení čipu MAX7219	27
6.4	Schéma zapojení 7segmentových displejů	27
6.5	Schéma zapojení mikrokontroléru ATmega32	28
6.6	Schéma zapojení modulu rrd-102	28
6.7	Schéma zapojení audio čipu TDA7307	29
6.8	Schéma zapojení koncových zesilovačů TDA7377	29
6.9	3D náhled výsledné desky	30
7.1	Atmel studio - okno s výchozím programem	31
7.2	Tabulka pro převod na znaky pomocí BCD kódu	33
7.3	Ideální průběh enkodérů	34
7.4	Průběh enkodérů se záškrtami	34
7.5	Využití arduino desky jako externího programátoru	40
8.1	1 kHz na vstupu, zesílení nastaveno na hodnotu 56	41
8.2	1 kHz na vstupu, zesílení nastaveno na hodnotu 47	42

8.3	10 kHz na vstupu, zesílení nastaveno na hodnotu 47	42
-----	--	----

Seznam tabulek

3.1	Obsah registru SPCR [16]	11
3.2	Obsah registru SPSR [16]	12
3.3	Obsah registru TWCR [19]	14
7.1	Obsah registru 02h [31]	35
7.2	Obsah registru 04h [31]	36
7.3	Obsah registru 05h [31]	36
7.4	Obsah registru 07h [31]	36
7.5	Úvodní sekvence pro inicializaci oled displeje [33]	37
7.6	Adresy registrů čipu TDA7303 [29]	38

Seznam výpisů kódu

Chtěl bych především poděkovat vedoucímu práce panu Ing. Pavlu Kubalíkovi, Ph.D. za odborné vedení a vstřícný přístup. Dále bych chtěl poděkovat své rodině za podporu v době, kdy tato práce vznikala.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 16. května 2024

Abstrakt

Tato práce se zabývá návrhem analogového audio zesilovače řízeného mikrokontrolérem ATmega32. Hlavním úkolem byl návrh jak dekodéru pro příjem příkazů od dálkového ovládání pomocí infračerveného světla tak ovládání pomocí enkodérů nacházejících se na předním panelu zesilovače. Teoretická část se zaměřuje na výběr sběrnic pro vzájemnou komunikaci a na prozkoumání protokolů pro infračervené ovládání. Praktická část se posléze zaměřuje na výběr správných komponent, návrh plošného obvodu a nakonec na softwarovou implementaci. Výstupem této práce je plně funkční stavebnice audio zesilovače pro začátečníky. V příloze lze nalézt kompletní zdrojový kód programu pro mikrokontrolér a gerber soubory potřebné pro vyhotovení desky plošných spojů.

Klíčová slova analogový audio zesilovač, koncový zesilovač, infračervené dálkové ovládání, mikrokontrolér ATmega32, C++, DPS

Abstract

This thesis deals with the design of an analog audio amplifier controlled by an ATmega32 microcontroller. The main task was the design of both a decoder for receiving commands from a remote control using infrared light and control using encoders located on the front panel of the amplifier. The theoretical part focuses on the selection of buses for mutual communication and on the examination of protocols for infrared control. The practical part then focuses on the selection of the right components, the design of the circuit board and finally on the software implementation. The output of this work is a fully functional audio amplifier kit for beginners. In the attachment you can find the complete source code of the program for the microcontroller and the gerber files needed to make the printed circuit board.

Keywords analog audio amplifier, power amplifier, infrared remote control, microcontroller ATmega32, C++, PCB

Seznam zkratek

DMA	Direct memory access – přímý přístup do paměti
DPS/PCB	Deska plošných spojů
I2C	Inter-Integrated Circuit – sériová počítačová sběrnice
IR	infrared – infračervené světlo
OLED	Organic light-emitting diode – světelná dioda s organickou elektroluminiscenční látkou
RAM	Random access memory – paměť s přímým přístupem
RDS	Radio Data System – určený k přenosu doplňkových informací v sítích VKV FM
SPI	Serial Peripheral Interface – sériová počítačová sběrnice
USART	Synchronní / asynchronní sériové rozhraní
USB	Universal Serial Bus – universální sériová sběrnice



Kapitola 1

Úvod

Navzdory dnešnímu pokroku v oblasti audia, není možné zakoupit levný audio zesilovač, který by disponoval dostatečným počtem vstupů, výstupů a byl plně analogový. Samotné analogové zesílení je ale zapotřebí, například při zesilování výstupu z gramofonu. U digitálního zesilovače zde může docházet ke zkreslení zvuku.

Cílem této bakalářské práce je na základě průzkumu trhu navrhnout konkurence schopný analogový audio zesilovač. Práce dále analyzuje technologii dálkového ovládání s pomocí IR ovladače a komunikační sběrnice, které budou použity pro propojení komponent. Následně je zde proveden výběr komponent potřebných ke správné funkčnosti výrobku, návrh zapojení a na závěr softwarová implementace a test funkčnosti.

Ovládání zesilovače bude prováděno jak pomocí enkodérů přítomných na předním panelu zesilovače, tak i za pomoci infračerveného dálkového ovladače. Řídící jednotkou tohoto zesilovače bude mikrokontrolér ATmega32. Součástí návrhu bude i integrace rádio přijímače. K zobrazování hodnot nastavených na zesilovači budou použity 7segmentové displeje. K zobrazování frekvence rádia a textu bude použit oled panel.

Tento zesilovač bude možné používat například v domácnostech pro zesílení výstupu z různých rádií či CD přehrávačů. Nejvíce se ovšem bude hodit k zesílení výstupů z analogových přehrávačů, nejčastěji gramofonů. K zesilovači bude možné připojit prakticky jakékoliv reproduktory s impedancí od 2 do 4 Ohmů. Tento zesilovač navíc bude moct sestavit praktický úplný začátečník.

Existující řešení

Pro zobrazení současného stavu na trhu s audio zesilovači jsem si dovilil vybrat tři zástupce. Dva z nejnižší cenové kategorie jmenovitě Zesilovač LTC audio ATM6500BT a Fonestar AS-1515 a jeden z těch dražších konkrétně Marantz PM6007.

2.1 Zesilovač LTC audio ATM6500BT

- Cena: 2 652 Kč
- Na výstupu pouze jeden levý a pravý kanál a celkový výkon 50 W.
- Integrované rádio
- 2 vstupy pro rozhraní typu cinch, 2 vstupy typu jack a 1 usb vstup. [1]

Jako jeho největší nevýhodu bych zde označil možnost připojení pouze dvou reproduktorů. Tyto dva reproduktory lze navíc pouze připojit pomocí drátů. Chybí zde výstup typu jack. Vše je zde ovládáno tlačítky a pro zobrazení hodnot je zde jeden led display který dokáže zobrazit pouze předpřipravené znaky. [1]

2.2 Zesilovač Fonestar AS-1515

- Cena: 2 999 Kč
- Na výstupu pouze jeden levý a pravý kanál a celkový výkon 15 W.
- Integrované rádio
- 2 vstupy pro rozhraní typu cinch, 2 vstupy typu jack a 1 usb vstup. [2]

U těchto zesilovačů ale bohužel nelze zjistit, jestli jsou analogové, anebo digitální. Jako mínus už bych tedy označil neexistenci datasheetů. Ani zpracování není za tuto cenu kdoví jaké a absence výstupu typu jack je v dnešní době opravdu velké mínus, pokud má mít zesilovač širší použití. [1], [2]

2.3 Marantz PM6007

- Cena: 10 790 Kč
- Výstup pro 2 levé a 2 pravé reproduktory a celkový výkon 60 W.
- Obsahuje výstup typu jack
- Neobsahuje integrované rádio
- Má jak analogové vstupy, tak i digitální [3]

Jako velké plus bych samozřejmě označil tovární zpracování, ale to je kompenzováno vysokou cenou a uzavřeností systému. Marantz vyřešil ovládání analogového zesilovače motorizovanými potenciometry. To znamená, že se při přidávání nebo ubírání hlasitosti otáčí potenciometr motorem, zatímco při ovládání bez dálkového ovladače můžeme přímo otáčet potenciometrem. Toto má svá úskalí, jelikož se tyto motory musí spínat po stejně dlouhou dobu, aby se při každém stisku tlačítka na ovladači pootočily o stejný počet stupňů. Navíc je zde absence displeje pro zobrazení nastavených hodnot a díky chybějícímu podsvícení potenciometrů na dálku nelze jednoduše zjistit aktuální nastavení. [3]

3.1 Dělení zesilovačů

Zesilovače lze rozdělit na zesilovače napětí a výkonu:

- Zesilovače napětí zesilují signál od zdroje na požadovanou úroveň. Lze je rozlišovat na zesilovače s malou a velkou vstupní impedancí.
- Zesilovače výkonu musí na rozdíl od napěťového zesilovače zesílit i vstupní proud. Elektrický výkon zesilovače slouží k pohybu membrány reproduktoru. Tento pohyb se pak z části mění na akustický výkon. Lze nalézt zesilovače o výkonu od stovek mW až po zesilovače s výkonem několik kW. Koncový stupeň výkonových zesilovačů musí snést velké proudové zatížení. Musí zde být také dostatečný odvod ztrátového tepla. [4], [5]

Zesilovače lze také dělit dle pracovní třídy.

3.1.1 Zesilovač třídy A

Zesilovače třídy A trvale vedou proud na výstup bez ohledu na úroveň vstupního signálu. To znamená, že tranzistory ve výstupním stupni jsou neustále aktivní, což eliminuje přechodové zkreslení spojené s jinými třídami zesilovačů. Zesilovače třídy A jsou známy svou vysokou kvalitou zvuku a minimálním zkreslením, což je důsledek kontinuálního provozu výstupního stupně. Jednou z nevýhod třídy A je nízká účinnost, protože tranzistory jsou neustále aktivní, i když není přenášen žádný zvuk. To způsobuje, že zesilovače třídy A produkují více tepla než zesilovače jiných tříd. [4], [5]

3.1.2 Zesilovač třídy B

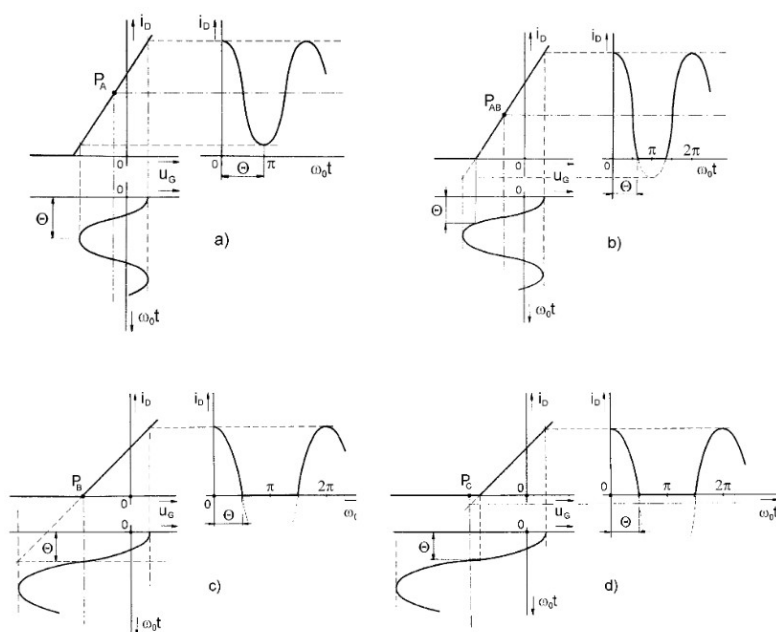
Zesilovače třídy B pracují na principu, kdy každý polovodič výstupního stupně je aktivní pouze při polovině vstupního signálu. Jedna polovina je aktivní pro kladný signál, zatímco druhá polovina je aktivní pro záporný signál. To pomáhá snížit spotřebu energie a tepelnou produkci ve srovnání s třídou A. Třída B je obecně efektivnější než třída A, protože tranzistory jsou aktivní pouze při přenášení signálu. Díky tomu, že tranzistory nejsou neustále aktivní, zesilovače třídy B produkují méně tepla než zesilovače třídy A. Tato třída je ale charakteristická přechodovým zkreslením při přechodu mezi kladným a záporným signálem. [4], [5]

3.1.3 Zesilovač třídy AB

Zesilovače třídy AB představují kompromis mezi zesilovači třídy A a třídy B, s cílem kombinovat výhody obou tříd. Tyto zesilovače využívají obě poloviny výstupního stupně, ale s tím rozdílem, že obě poloviny mohou být aktivní současně při malých signálech. To snižuje přechodové zkreslení, které je obvyklé u třídy B, a zároveň udržuje nižší spotřebu energie ve srovnání s třídou A. [4], [5]

3.1.4 Zesilovač třídy C

Tato třída je obvykle používána pro specifické aplikace, jako jsou například rádiové vysílače nebo některé speciální přenosové systémy. Tyto zesilovače jsou navrženy tak, aby byly v aktivním stavu pouze část časového cyklu signálu. To znamená, že jsou aktivní pouze během části vlny signálu, což snižuje průměrnou spotřebu energie a tepelnou produkci. Jsou obvykle velmi účinné, protože jsou aktivní jen krátký čas a poté jsou vypnuty. To znamená, že generují méně tepla než zesilovače tříd A nebo AB. Mohou mít vyšší harmonické zkreslení než jiné třídy, zejména při nižších úrovních signálu. [4], [5]



■ **Obrázek 3.1** Pracovní třídy zesilovače: a) třída A, b) třída AB, c) třída B, d) třída C [6]

3.2 Mikrokontrolér AVR

AVR odkazuje na rodinu mikrokontrolérů vyvinutých společností Atmel (nyní součástí Microchip Technology). Všeobecně je přijímáno, že zkratka AVR vznikla ze spojení "Alf and Vegard's RISC processor" (procesor s redukovanou sadou instrukcí Alf a Vegard), pojmenované podle dvou norských vývojářů, kteří tuto architekturu vytvořili. Mikrokontroléry AVR jsou známé pro svou jednoduchost, efektivitu a snadné použití, což je činí oblíbenými v široké škále aplikací. Využívají koncepci Harwardské architektury, to znamená, že mají oddělenou paměť pro program a pro data. Mikrokontroléry AVR používají architekturu s redukovanou sadou instrukcí (RISC), což zjednodušuje sadu instrukcí a umožňuje rychlejší a předvídatelnější vykonávání instrukcí. Obvykle obsahují paměť typu Flash pro ukládání programu, což umožňuje

snadné přeprogramování zařízení a paměť typu EEPROM pro ukládání nevolatilních dat. Tyto mikrokontroléry mají různé vestavěné periferie, jako jsou časovače, USART (univerzální synchronní/asynchronní přijímač/vysílač), SPI (sériové periferní rozhraní), I2C (Inter-Integrated Circuit) a další což umožňuje široké použití ve vestavných systémech. Mikrokontroléry AVR jsou navrženy tak, aby pracovaly s nízkou spotřebou energie, což je činí vhodnými pro bateriově napájené aplikace. Rodina AVR zahrnuje různorodou škálu mikrokontrolérů, od jednoduchých 8bitových zařízení po pokročilejší 32bitové modely. [7], [8]

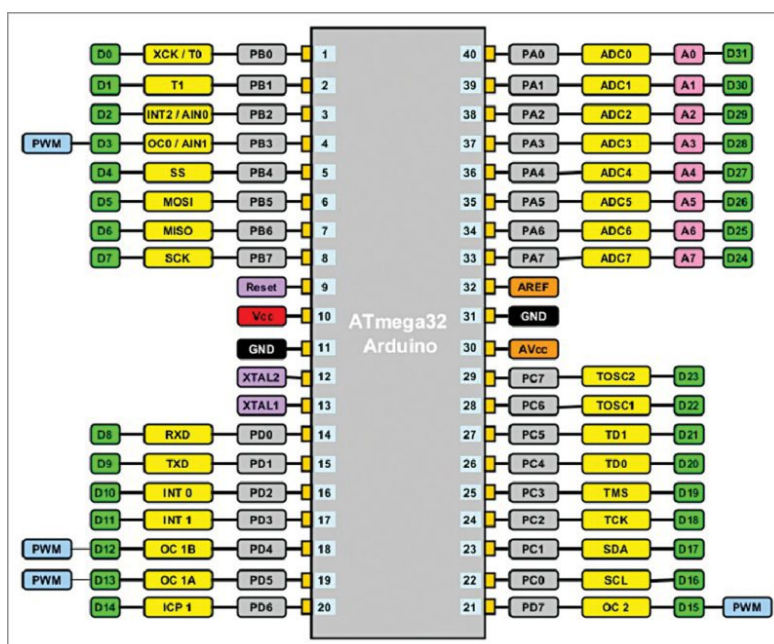
3.2.1 Základní Typy

- AT90 – řada, která už se dnes nevyrobí, náhradou může být řada ATmega nebo ATtiny
- ATtiny – využívají se v jednoduchých a malých elektronických obvodech
- ATmega – výkonné mikročipy, mají JTAG rozhraní, větší flash paměť a RAM, více integrovaných rozhraní.
- ATxmega – vlastnosti zvyšující výkon jako DMA, "Event System" a podpora kryptografie.
- ATmega pro speciální aplikace – ATmega se speciálními periferiemi jako LCD kontroléry, USB kontroléry, pokročilými PWM, CAN rozhraním, atd.
- FPSLIC™ (AVR s FPGA)
 - FPGA 5K – 40K hradel
 - SRAM pro programový kód, na rozdíl od ostatních AVR.
 - jádro AVR může běžet až na 50 MHz
- 32bitové AVR [8]

Některé mikrokontroléry typu ATmega se například používají v deskách arduino. Konkrétně jde o mikrokontrolér ATmega328p, ATmega2560 a ATmega32u4. Já jsem se rozhodl použít mikrokontrolér ATmega32 jelikož je relativně levný a obsahuje potřebný počet I/O pinů. Lze ho také koupit v provedení DIP, což umožňuje lehčí sestavení výsledného obvodu. Jde vlastně o mikrokontrolér ATmega32u4 bez vestavěného usb. [9]

3.2.2 Parametry ATmega32

- 24 digitálních vstupně výstupních pinů a 8 analogových
- podpora USART, I2C, SPI
- 32KB flash pro program, 1KB EEPROM a 2KB SRAM
- Frekvence až 16 MHz [10]



■ Obrázek 3.2 Popis pinů mikrokontroléru ATmega32 [11]

3.3 Dálkové ovládání pomocí infračerveného světla

Dálkové ovládání pomocí infračerveného světla je běžná technologie, která se používá k ovládání různých elektronických zařízení, jako jsou televize, audio zařízení, klimatizace a mnoho dalších. Tato metoda komunikace umožňuje uživatelům pohodlně ovládat svá zařízení na dálku, aniž by bylo nutné se jich fyzicky dotýkat. [12]

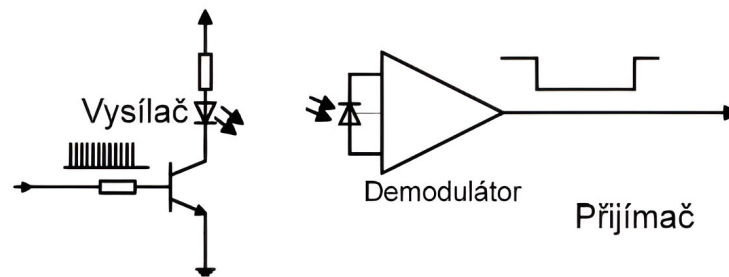
3.3.1 Princip fungování

Infračervená led dioda dokáže vyzařovat infračervené světlo. Tento druh záření je neviditelný pro lidské oko, protože má větší vlnovou délku, než viditelné světlo. Typicky má toto světlo vlnovou délku okolo 950nm. Naneštěstí infračervené světlo může produkovat spousta dalších zdrojů, než jen infračervená led dioda. Například: klasická žárovka nebo slunce. [12]

3.3.2 Metody přenosu dat

Nejlehčí metoda přenosu dat pomocí infračerveného světla by se zakládala na rozsvícení IR led diody, pokud chceme přenést logickou 1 a na zhasnutí pokud chceme přenést logickou 0. Každý takový bit, by se přenášel po nějakou časovou periodu, aby byl rozpoznatelný řetězec stejných bitů za sebou. Fakt že IR led dioda není jediná, kdo může vyzařovat infračervené světlo nám, ale tuto metodu zavrhuje. Mohlo by zde totiž docházet ke zkreslení dat například při rozsvícení v pokoji. [12]

Vylepšení této metody spočívá v modulaci infračerveného záření. Pokud totiž zmodulujeme náš světelný paprsek nějakou vysokou frekvencí, lze potom tento paprsek lehce filtrovat na straně přijímače. Postup je přitom velmi jednoduchý. Stačí po dobu vysílání dotyčného bitu s určitou frekvencí rozsvěcet a zhasínat IR led diodu. Typicky se používá frekvence 38 kHz. Tuto frekvenci lze také nazývat jako nosnou frekvenci signálu. [12]



■ **Obrázek 3.3** Dekódování modulovaného záření

3.3.3 Samotná komunikace

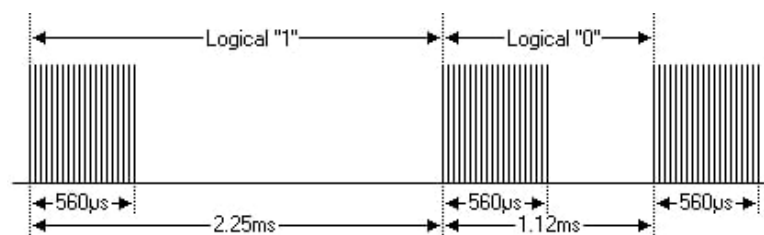
Prakticky každý může komunikovat pomocí Infračerveného světla. Není vyžadována žádná autentizace nebo autorizace mezi vysílačem a přijímačem. Každé zařízení lze ovládat tím ovladačem, který vysílá definované bitové sekvence. Díky tomu lze prodávat Univerzální ovladače. Tyto ovladače lze programovat již naprogramovaným ovladačem. Ten je na ně namířen a jsou na něm stisknuta tlačítka. Programovaný ovladač se ve chvíli jeho programování chová jako přijímač. Posléze jen opakuje bitové sekvence, které byly dříve přijaty pro jednotlivá tlačítka. [12]

3.3.4 Protokoly

Informace pomocí infračerveného světla lze přenášet pomocí vícero protokolů. Hlavní rozdíl mezi těmito protokoly spočívá ve způsobu přenosu logické 1 a 0 a nosnou frekvencí. Dvěma nej-používanějšími protokoly jsou Nec a RC5. [13], [14]

3.3.5 Protokol NEC

Protokol NEC byl vyvinut firmou NEC (Nippon Electric Company). Používá jednoduchý způsob kódování signálů, které jsou vysílány pomocí infračerveného světla. Typicky se v tomto protokolu používá binární kódování, kde jsou digitální hodnoty reprezentovány sérií krátkých a dlouhých pulzů. [13]



■ **Obrázek 3.4** Kódování pomocí protokolu NEC [13]

Základní charakteristiky protokolu NEC zahrnují:

- Kódy NEC jsou obvykle 32 bitů dlouhé a mají strukturu, která zahrnuje vlastní kód zařízení, kód příkazu, inverzní kód a stop bit.
- Opakování signálů: Každý příkaz se opakuje několikrát pro zajištění spolehlivosti přenosu.

- Startovací pulz: Každý přenos začíná krátkým startovacím pulsem, který slouží k synchronizaci.
- Adresa a příkaz: Následuje část kódu, která obsahuje informace o adrese zařízení a samotný příkaz, který má být vykonán.
- Bitová inverze: Jedná se o techniku, kdy příslušné bity jsou invertovány (změněny z 1 na 0 a naopak), což umožňuje detekci chyb.
- Používá modulační frekvenci 38 kHz, což znamená, že infračervené signály jsou modulovány s touto frekvencí. [13]

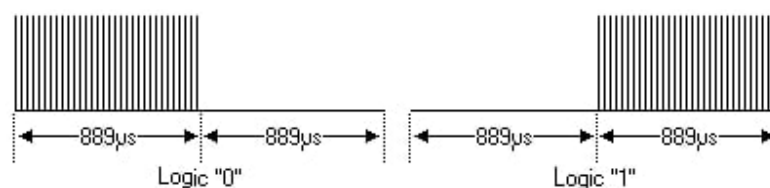


■ Obrázek 3.5 Průběh NEC protokolu [13]

3.3.6 Protokol RC5

Protokol RC5 byl vyvinut firmou Philips. Základní charakteristiky protokolu RC5 zahrnují:

- Nemá pevně stanovenou modulační frekvenci, což umožňuje větší flexibilitu v implementaci.
- Kódy RC5 jsou 14 bitů dlouhé a obsahují informace o adrese a příkazu. Neobsahují inverzní kód.
- Obsahuje 2 startbity a jeden toggle bit. Toggle bit je invertován při každém stisknutí tlačítka. Díky tomu lze vždy zjistit jestli bylo tlačítko znovu stisknuto, anebo jestli je stisknuto po delší dobu. [14]



■ Obrázek 3.6 Kódování pomocí protokolu RC5 [14]



■ Obrázek 3.7 Průběh RC5 protokolu [14]

3.3.7 Komponenty infračerveného ovladače

Ovladač pro svoji funkci potřebuje:

1. Mikrokontrolér: Řídí celkový chod dálkového ovladače a zpracovává stisknutá tlačítka
2. Infračervený vysílač: Generuje infračervené pulzy odpovídající každému tlačítku.
3. Baterie nebo zdroj energie: Poskytuje napájení pro provoz ovladače.
4. Tlačítka: Slouží k výběru různých funkcí na ovládaném zařízení. [12]

3.3.8 Vlastnosti infračervené komunikace

- Neviditelnost: Infračervené světlo není viditelné lidským okem, což umožňuje komunikaci bez jakéhokoli rušení pro uživatele.
- Omezený dosah: Infračervená komunikace má omezený dosah a je účinná pouze v přímé viditelnosti mezi ovladačem a zařízením.
- Jednoduchá implementace: Technologie infračerveného ovládání je relativně jednoduchá a levná na implementaci. [12]

3.3.9 Vývoj a budoucnost

S postupem technologie se objevují nové formy dálkového ovládání, jako je bezdrátové ovládání přes radiové vlny nebo pomocí internetu. Nicméně infračervené ovládání zůstává populární pro svou spolehlivost a jednoduchost. Výrobci se snaží zdokonalovat infračervené ovladače v oblasti energetické účinnosti a rozšířených možností konfigurace.

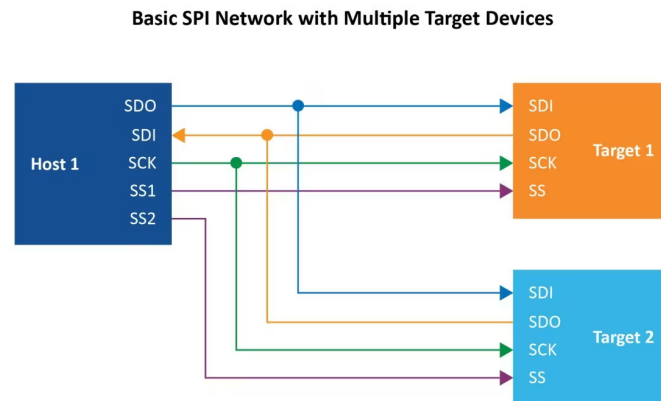
3.4 Sběrnice SPI

SPI je sériová sběrnice sloužící pro vzájemné propojení dvou či více komunikujících uzlů, přičemž jeden uzel obvykle vystupuje v roli takzvaného řadiče sběrnice (master), ostatní uzly pracují v režimu slave. Uzel, který pracuje jako master, obsahuje generátor hodinového signálu, který je rozveden do všech ostatních uzlů, čímž je umožněn zcela synchronní (navíc ještě obousměrný) přenos dat. Hodinový signál je rozváděn vodičem označovaným symbolem SCK. [15]

Dále jsou zde přítomny dva vodiče pro přenos dat:

- Datový vstup MISO (Master In, Slave Out): Po tomto vodiči jsou data posílána z slave obvodů do master obvodu.
- Datový výstup MOSI (Master Out, Slave In): Po tomto vodiči jsou data posílána z master obvodu do slave obvodů. [15]

Každý obvod typu slave má navíc vstup SS (Slave select) pro výběr obvodu. Je-li tento vstup neaktivní, SPI daného obvodu je také neaktivní a vodič MISO příslušného obvodu je ve vysoko impedančním stavu. Pro každý obvod typu slave existuje jeden vodič SS. Pomocí těchto vodičů vybíráme obvod typu slave se kterým chceme komunikovat. [15]



■ **Obrázek 3.8** Způsob komunikace s více zařízeními pomocí výběru uzlu signálem SS [16]

3.4.1 Spi rozhraní mikrokontroléru ATmega32

Procesor ATmega32 obsahuje rozhraní SPI a díky tomu lze pro posílání dat využít vnitřních registrů. Konkrétně obsahuje registry:

- SPCR (SPI Control Register) – Toto je vlastně hlavní registr. Obsahuje bity pro inicializaci a následné řízení SPI.
- SPSR (SPI Status Register) – Toto je stavový registr. Je určen ke čtení stavu jednotlivých linek (MISO a MOSI).
- SPDR (SPI Data Register) – Tento registr je použit pro čtení a posílání dat skrze SPI sběrnici [16]

■ **Tabulka 3.1** Obsah registru SPCR [16]

7. bit	Nastavení přerušení
6. bit	Zapnutí SPI
5. bit	Formát zasílání dat (MSB vs LSB)
4. bit	Určuje jestli je zařízení typu master nebo slave
3. bit	Nastavuje polaritu hodin
2. bit	Nastavuje, kdy jsou jednotlivé bity posílány. Jestli s náběžnou nebo spádovou hranou.
1. a 0. bit	Nastavují frekvenci hodin.

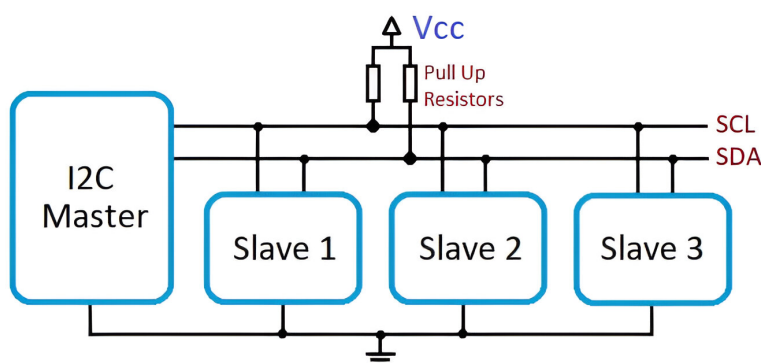
■ **Tabulka 3.2** Obsah registru SPSR [16]

7. bit	Příznak přerušení
6. bit	Příznak kolize tzn: Je aktivní pokud jsou data zapisována do datového registru, přestože předchozí přenos ještě nebyl dokončen.
5. – 1. bit	Rezervované bity
0. bit	Tento bit dokáže frekvenci hodin násobit dvěma. Tento bit je využíván pouze, pokud jsou nějaké speciální nároky na přenosovou rychlost.

Pro použití těchto registrů je potřeba využít příslušné piny mikrokontroléru Atmega32 určené právě pro SPI sběrnici. [16]

3.5 Sběrnice I2C

I2C je interní datová sběrnice sloužící pro přenos dat mezi jednotlivými integrovanými obvody. Na rozdíl od SPI sběrnice jsou potřeba pouze 2 vodiče na místo 3. Komunikace je ale díky tomu pouze poloduplexní (v jeden okamžik mohou data téct pouze jedním směrem). Zároveň nejsou potřeba žádné další vodiče pro výběr obvodu. Využívá vodiče SDA (serial data) a SCL (serial clock). Jednotlivé obvody jsou synchronizovány pomocí vodiče SCL. Přenosová rychlost je standardně 100 kHz. Pokud jsou vyšší nároky na přenosovou rychlost, může být zvýšena na 400 kHz nebo 1 MHz. Ne všechny obvody ale tuto vyšší rychlost podporují. Oba sběrnice vodiče musí být při nečinnosti v logické 1. Toho lze docílit přidáním pull-up rezistorů. [17], [18]



■ **Obrázek 3.9** Možné schéma zapojení [18]

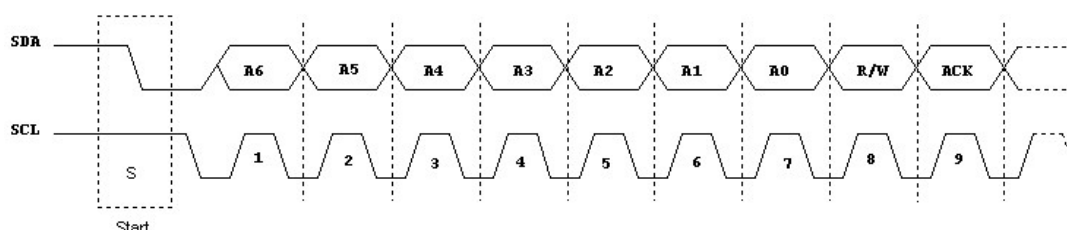
3.5.1 Adresace

Pro volbu zařízení komunikujícího pomocí sběrnice I2C používáme adresy. Díky tomu nemusíme vést selektující vodič ke každému obvodu. To především u mikrokontrolérů výrazně optimalizuje nároky na počet vstupně-výstupních pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově a v rozšířené verzi 10bitově. To umožňuje připojení 128 respektive 1024 čipů s různou adresou na jednu společnou sběrnici. V praxi jsou však tato čísla podstatně nižší, protože adresu čipu většinou nelze měnit plnými 7 (10) bity ale třeba jen třemi. U některých čipů jí nelze měnit vůbec. Takový čip může pak být na sběrnici přítomen pouze jeden. [17], [18]

3.5.2 Princip přenosu

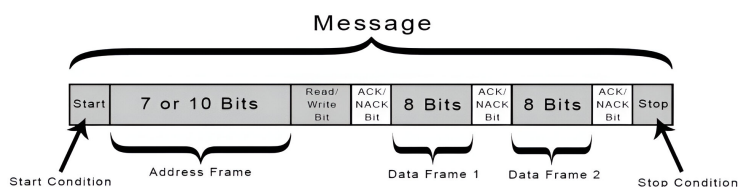
Jeden z integrovaných obvodů je nastaven do režimu master a ostatní do režimu slave. Veškeré řízení sběrnice má na starosti zařízení nakonfigurované do režimu master. V jednu chvíli může jako master pracovat pouze jediné zařízení, čímž je zaručeno, že na sběrnici nebude docházet ke kolizím. Ostatní uzly, které pracují v režimu slave, nemohou sběrnici řídit a dokonce ani nemají možnost samy žádat o vysílání či příjem dat. Master při jakémkoli přenosu generuje hodinový signál na vodiči SCL. Když jeden čip vysílá, přijímají všechny ostatní a pouze podle adresy určí, zda jsou data určena jim. [17], [18]

Komunikaci vždy zahajuje master a to tak, že sníží úroveň na datovém vodiči SDA na logickou nulu, zatímco SCL si po určitou dobu udržuje stav logické jedničky (tato doba je závislá na zvolené přenosové rychlosti). Tento stav, který se označuje termínem start bit, rozpoznávají všechny uzly připojené na sběrnici. Ihned po vyslání start bitu začne master vysílat adresu uzlu, se kterým si přeje komunikovat. Musí také určit, jestli chce do zařízení vysílat, anebo z něj přijímat. To určí R/W bitem který je součástí adresy. Standardně se k adresaci používá 7 bitů a poslední bit je R/W bit. Plná adresa tedy tvoří jeden byte. [17], [18]



■ Obrázek 3.10 Počátek I2C komunikace [17]

Každý bit musí mít ustálenou logickou hodnotu ve chvíli, kdy přijde náběžná hrana hodin. Po přenosu všech osmi bitů (sedm bitů adresy + osmý bit s určením směru přenosu dat) provede každý uzel slave porovnání přijaté adresy se svoji vlastní adresou. Vybraný uzel (jen on) totiž musí potvrdit, že se na sběrnici skutečně nachází. Potvrzení (bit ACK – acknowledge) se v devátém cyklu hodin pošle zpět na zařízení typu master. Příjem logické nuly značí, že adresovaný uzel skutečně existuje a je možné s ním zahájit komunikaci, logická jednička naopak může znamenat, že uzel s danou adresou buď neexistuje, nebo je z nějakého důvodu odpojen či prostě není z různých příčin připraven komunikovat. Samotná data se standardně posílají po jednom byte, za kterým následuje také potvrzení. Pokud chceme změnit datový tok mezi master obvodem a slave obvodem, musíme ukončit komunikaci a vybrat zařízení znovu se správně nastaveným READ/WRITE bitem. [17], [18]



■ Obrázek 3.11 Průběh I2C komunikace [18]

3.5.3 I2C rozhraní mikrokontroléru ATmega32

Procesor ATmega32 disponuje modulem pro I2C komunikaci. Pro správnou funkčnost je potřeba nastavit registry TWCR (control register) a TWBR (bit rate register). Registr TWBR nastavuje hodinovou frekvenci na vodiči SCL. Je do něj ukládán poměr mezi frekvencí krystalu a požadovanou frekvencí. [19]

■ **Tabulka 3.3** Obsah registru TWCR [19]

7. bit	I2C Interrupt Flag, je nastaven pokaždé, když se dokončí operace a proto se používá pro kontrolu dokončení přenosu. Vždy musí být na začátku přenosu softwarově resetován do logické 1.
6. bit	I2C Enable Acknowledge Bit
5. bit	I2C Start Condition
4. bit	I2C Stop Condition
3. bit	I2C Write Collision Flag
2. bit	I2C povoleno
0. bit	I2C přerušení povoleno.

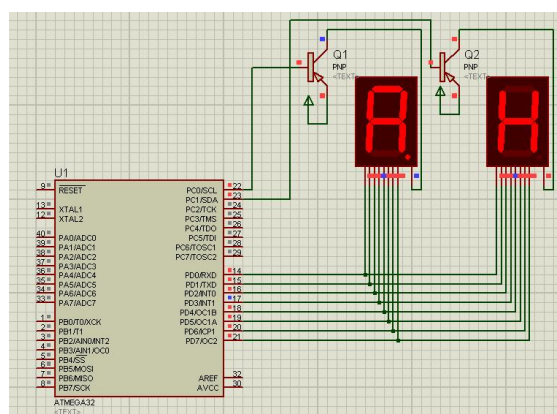
Kapitola 4

Analýza

Mikrokontrolér sám o sobě nemá žádný přínos. Pro správnou funkčnost je potřeba k mikrokontroléru připojit příslušná periferní zařízení. Mikrokontrolér pouze načítá data z periferních zařízení, ty posléze zpracuje a provede akce náležící k přijatým datům.

4.1 Displej pro výpis hodnot

Pro výpis hodnot zesilovače (hlasitost, hloubky, výšky, balance) jsem se rozhodl použít čtyři 7segmentové displeje, každý se dvěma možnými znaky. Tyto displeje mají společnou katodu. K řízení těchto displejů přímo z mikrokontroléru bych potřeboval 16 pinů. Samotný mikrokontrolér má 32 pinů a tím pádem by polovina možných pinů byla vynaložena na zobrazování hodnot. Rozhodl jsem se je tedy řídit nějakým jiným čipem, do kterého bych pouze posílal hodnoty k zobrazení.



Obrázek 4.1 Příklad zapojení dvouznakového displeje řízeného přímo mikrokontrolérem

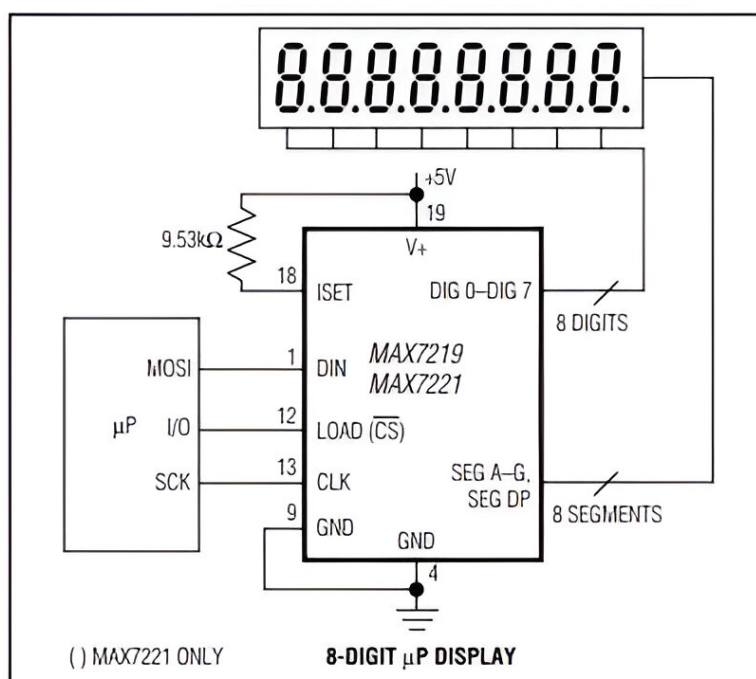
Prvotní záměr byl použit k řízení těchto displejů jednoduché posuvné registry 74HC595. Tyto registry komunikují pomocí SPI a dají se zapojovat do série. Zde ale vyvstalo několik problémů.

1. Při každém přepisu jedné hodnoty, by bylo potřeba přepisovat všechny displeje
2. Displeje by neustále svítily a měly by velkou spotřebu
3. K plné funkčnosti by bylo potřeba 5 těchto čipů, nebo 2 ale mikrokontrolér by byl zatížen neustálým obnovováním těchto displejů

Jako další mě napadlo použít čip TM1637 který je často používán pro hodinové displeje. Jeho nevýhodou je ale možnost obsluhovat pouze 6 znaků zároveň. Tudíž by byly potřeba dva, s tím že jeden by obsluhoval pouze 2 znaky. Tyto čipy ale bohužel nelze zapojovat do série. Pracují totiž na upravené I2C sběrnici. Nemají adresu a místo ní mají WE (write enable) pin. To znamená, že by byly potřeba celkově 4 piny (2 pro I2C sběrnici a dva pro selekci čipu). Program by následně musel rozlišovat, do kterého čipu se má zrovna zapisovat. [20]

Jako řešení jsem si tedy vybral čip MAX7219. Pro tento úkol představuje skvělou volbou jelikož:

- Pracuje na SPI sběrnici
- Dokáže simultánně zobrazovat na až 8 znacích
- Obnovovací frekvence znaku je 800 hz
- Obsahuje možnost nastavení jasu
- Obsahuje statickou RAM paměť, kam se na jednotlivé adresy zapisují hodnoty. Ty jsou následně vypisované na displej. To znamená, že se při změně hodnoty nemusí přepisovat všechny hodnoty ale pouze jedna na konkrétní adrese.
- Operační napětí je 5V [21]



■ **Obrázek 4.2** Schéma zapojení čipu MAX7219

[21]

4.2 Ovládání zesilovače na předním panelu

Pro ovládání zesilovače nelze použít klasické potenciometry, jelikož je potřeba měnit nastavení i pomocí ovladače. To u klasických potenciometrů nelze. Jediná možnost jak u nich změnit nastavení, je jejich manuální otočení. Tuto skutečnost lze obejít potenciometry s vestavěným motorem. Tyto potenciometry se používají u dražších zesilovačů. Například je používá již zmíněný Marantz PM6007 [3].

Příkladem je motorizovaný potenciometr PDB183-S425K-104A3. Mezi jeho hlavní nevýhody patří určitě cena. Tento potenciometr stojí 152kč, což pro zesilovač který má ostatním zesilovačům konkurovat mimo jiné cenou není úplně žádoucí. Mezi další nevýhody spadá i jejich ovladatelnost. Při otáčení skrze motor je nutné dodržovat jasně definovaný časový úsek, aby se otáčely vždy o stejný úhel. Mezi plusy samozřejmě spadá možnost použití prakticky jakéhokoliv schématu analogového zesilovače.

Motor se ovládá 4.5V a k chodu potřebuje 100 mA. ATmega32 dokáže na jednotlivých pinech produkovat maximálně 40 mA. K ovládání takového potenciometru je tedy potřeba další zdroj, který by musel být spouštěn pomocí mikrokontroléru. To vede k vyšší komplexitě obvodu a zároveň i k vyšší spotřebě. [22], [23]

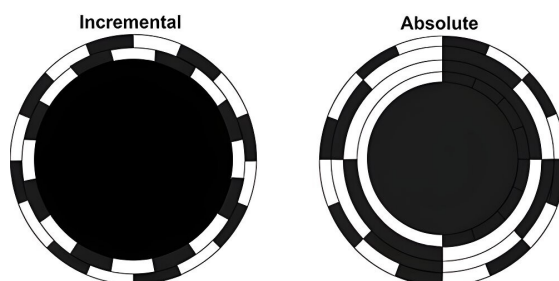
Jako druhá možnost se jeví ovládání pomocí tlačítek. Takové ovládání je ale v případě zesilovače méně komfortní. Pokud chce uživatel například rychle zvýšit hlasitost, musí tlačítko mnohokrát zmáčknout, anebo delší dobu držet. Proto jsem se rozhodl použít rotační enkodéry.

4.2.1 Rotačního enkodér

Rotační enkodéry jsou zařízení, která se obecně mohou použít v aplikacích, kde se měří délka, pozice, rychlost, úhel natočení nebo všechny veličiny, které na ně lze převést. Provádějí přeměnu mechanického pohybu na elektrické signály. Běžné rotační enkodéry se dělí na dvě skupiny:

- inkrementální – generují sled pulsů
- absolutní – produkují sled unikátních kódů [24]

Konstrukčně jednodušší inkrementální provedení při rotačním pohybu generuje pulsy, jejichž určitý počet odpovídá určité vzdálenosti, posuvu nebo natočení a závislost počtu pulsů na časový úsek odpovídá rychlosti. Výstupní signál je tedy relativní a informuje jen o pohybu, ne však sám o sobě o přesné pozici. Tu případně na základě výchozí pozice musí vyhodnocovat až externí měřicí jednotka (například mikrokontrolér). Konstrukčně složitější absolutní provedení již generují pro každou pozici natočení konkrétní číselnou hodnotu. Tedy v každém momentě je přesně definováno v jakém úhlu natočení se enkodér nachází. Nejčastěji se u těchto enkodérů používá kódování pomocí Grayova kódu. [24]

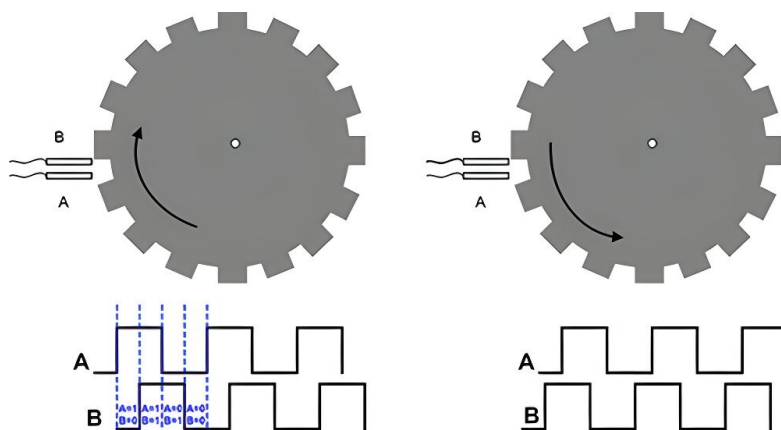


■ Obrázek 4.3 Rozdíl meti inkrementálním a absolutním enkodérem [24]

4.2.2 Princip inkrementálního rotačního enkodéru

Enkodér převádí pohyb na sekvence elektrických digitálních impulzů. Toho může docílit dvěma způsoby.

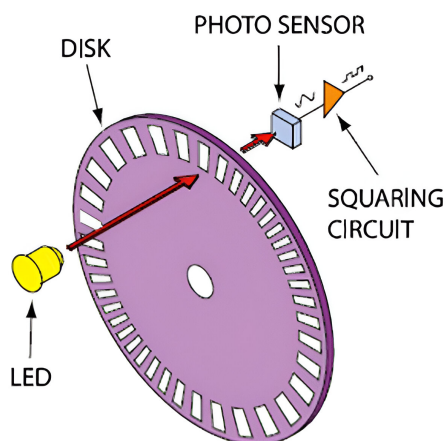
1. Způsob je užití kontaktů, které jsou při otáčení hřídele spínány a rozpínány. Takový enkodér se nazývá mechanický. Z počtu sepnutí kontaktů lze vyčíst o kolik se enkodér otočil. [24]



■ Obrázek 4.4 Princip enkodéru s kontakty

[24]

2. Způsob užívá led diodu a fototranzistor. Mezi led diodu a fototranzistor je umístěn disk, který má po svém obvodu otvory. Tento disk je spojen s hřídelí. Světlo z led diody může procházet diskem, pouze pokud se v jeho místě zrovna nachází otvor. Při otáčení tedy světlo střídavě osvětluje fototranzistor. Pokud je fototranzistor střídavě spínán a rozpínán, enkodér se otáčí. Z počtu sepnutí tranzistoru lze vyčíst o kolik se enkodér otočil. Takto fungující enkodér se nazývá optický.

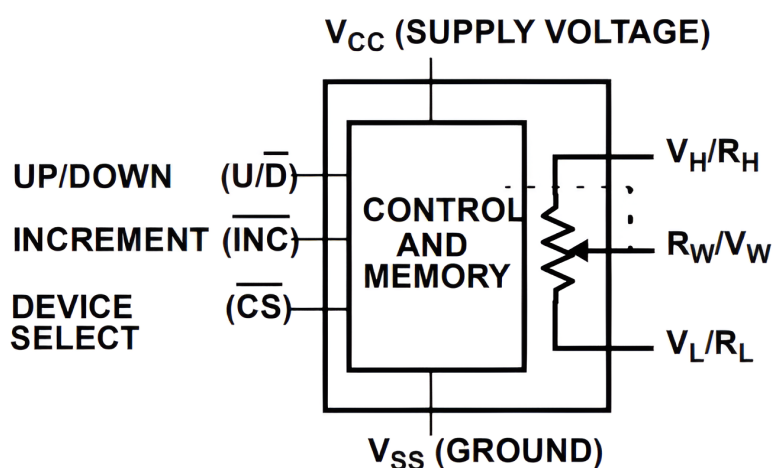


■ Obrázek 4.5 Princip optického enkodéru

Mechanický enkodér je zpravidla levnější ale za to méně spolehlivý a má menší rozlišovací schopnost. Já jsem se rozhodl použít mechanický enkodér, jelikož v mém případě je důležitější cena než rozlišovací schopnost. Spolehlivost mechanického enkodéru je pro tuto úlohu dostačující. Enkodéry jsou o dost levnější než motorizované potenciometry (152 Kč vs 18 Kč) a mají delší životnost (zhruba dvojnásobná). Oproti tlačítkům neztrácí zase tu možnost rychlého přenastavení. Enkodéry navíc fungují velice jednoduše. Pro obsluhu jednoho enkodéru jsou potřeba 2 piny mikrokontroléru. Jeden pro kontrolu jestli je enkodérem otáčeno a druhý na zjištění směru otáčení. Navíc mnou vybrané enkodéry obsahují také integrované tlačítko. To lze například použít na změnu vstupu zesilovače. Toto tlačítko potřebuje jeden pin. Tím pádem jsou potřeba tři piny na enkodér. Na pinech musí být nastaven pull-up rezistor jelikož se při pohybu piny propojují se zemí. Celkem jsou pro veškerou obsluhu zesilovače potřeba 4 enkodéry. [25]

4.3 Ovládání audia

Po výběru enkodéru jako primárního ovládání ale vyvstal problém, jak ovládat samotný obvod. Pro začátek jsem zvolil digitální potenciometry. Konkrétně jsem vybral typ X9C104S. Jejich použití je v celku snadné. Obsahují 3 piny, jejichž funkčnost je stejná jako u klasického potenciometru. Ovládají se třemi piny, jedním je určen čip, druhým směr a pomocí třetího je zvyšován, anebo snižován odpor podle nastaveného směru. [26]



■ **Obrázek 4.6** Schéma zapojení digitálního potenciometru [26]

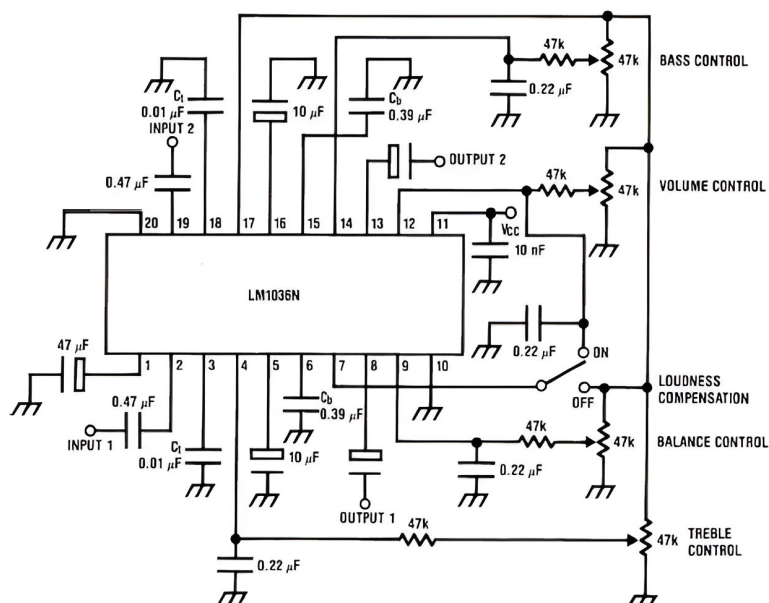
Výběr těchto digitálních potenciometrů určoval i výběr samotného obvodu pro předzesilovač. Můj prvotní nápad byl navrhnout tento předzesilovač okolo operačních zesilovačů typu TL072CP. Tyto zesilovače mají nízký šum a každý čip obsahuje dva operační zesilovače. Díky tomu se skvěle hodí pro návrh stereo zesilovače. U tohoto návrhu ale vyvstalo několik komplikací:

1. Pro správný chod by bylo potřeba symetrické napájení, což by vedlo k větší komplexnosti obvodu.
2. Navíc bych potřeboval 8 digitálních potenciometrů (4 pro každý kanál). U těchto digitálních potenciometrů jsem navíc narazil na velké odchylky. V datasheetu je uvedena 20% tolerance, což by vyústilo v nevyrovnanost kanálů. [26], [27]

Jako druhý pokus jsem zvolil audio procesor LM1036N. Mezi jeho výhody oproti užití standardních operačních zesilovačů patří jeho asymetrické napájení. Implementuje jak ovládání hlasitosti, tak i nastavení úrovně basů, výšek a nastavení stereo váhy. Tyto hodnoty jsou nastavovány

skrže velikost napětí přivedeného na dotyčné vstupy čipu. Bohužel maximální možné napětí je rovno 5,4 V. To znamená, že je pro nastavení například hlasitosti na maximum potřeba přivést na dotyčný pin čipu 5,4 V. Mikrokontroléry Atmega dokáží pracovat maximálně s 5V. Díky tomuto faktu jsem nemohl použít PWM výstupy k řízení audio procesoru a byl jsem znovu odkázán na digitální potenciometry. Aktuálně už bych ale potřeboval pouze 4. [28]

Největší nevýhoda tohoto audio procesoru ale tkví v jeho nedostupnosti. Tento audio procesor lze sehnat pouze v Číně např. na internetovém obchodě aliexpress. Texas instruments jej totiž v roce 2015 přestal vyrábět. Vyrábí se už jen čínské kopie a ty dle mých zkušeností nedosahují tak dobrých vlastností jako originál.



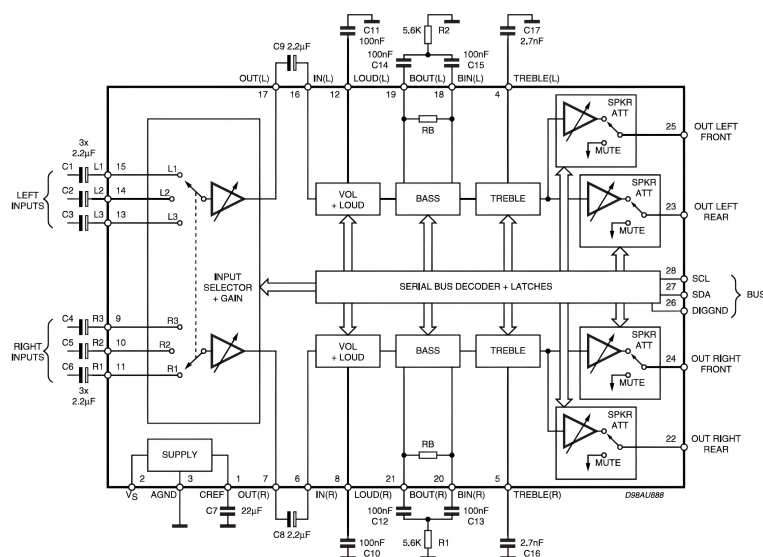
■ **Obrázek 4.7** Schéma zapojení čipu LM1036 [28]

Uvažoval jsem také nad čipem TDA1524. Tento čip je prakticky shodný s čipem LM1036N. Prakticky jediný rozdíl mezi nimi je stálá dostupnost tohoto čipu. Nutnost použití digitálních potenciometrů, díky jejich vysoké toleranci se mi ale zamlouvala čím dál tím méně.

Poté jsem narazil na audio procesor TDA7429. K ovládání tohoto audio procesoru je použita I2C sběrnice a tudíž nejsou potřeba žádné potenciometry. Tento procesor navíc nevyžaduje prakticky žádný další obvod. Tím pádem o dost zlehčuje návrh celkového obvodu. Bohužel už tento procesor, stejně jako procesor LM1036N, není vyráběn. Byl jsem tedy nucen sáhnout po trochu méně výkonném audio procesoru TDA7303. Ten je také řízen I2C sběrnicí. Tento audio procesor nabízí:

- Volbu mezi 3 vstupy
- 4 výstupy (levý přední, pravý přední, levý zadní, pravý zadní)
- Nastavení zesílení pro každý vstup
- Asymetrické napájení 6-10 V
- Krok nastavení hlasitosti 1.25 dB
- Nastavení basů a výšek -14 dB až +14 dB
- Možnost ztlumení každého výstupu zvlášť [29]

Lze jej koupit za 64 Kč v internetovém obchodě mouser.com. Nepotřebuje prakticky žádný vnější obvod. Díky ovládání skrze I2C sběrnici nepotřebuje ani digitální potenciometry. Tyto dva fakty návrh celého obvodu zjednodušují a zároveň také zlevňují. K ovládání celého audia nejsou potřeba kromě pinů I2C sběrnice žádné další piny na mikrokontroléru. [29]



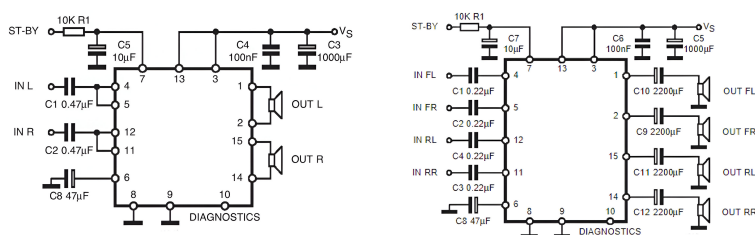
■ Obrázek 4.8 Schéma zapojení čipu TDA7303 [29]

4.4 Koncový zesilovač

Jelikož již vybraný TDA7303 má zesílení jen 11,25 dB, rozhodl jsem se do obvodu přidat ještě koncový zesilovač. Jako první jsem přemýšlel nad použitím zesilovače TDA2009. Tento čip jsem již jednou použil při výrobě zesilovače k počítači a byl jsem s ním velmi spokojen. Problém ale nastával v jeho samotném výkonu. Tento zesilovač má totiž výkon pouze 2x10 W. To je sice dostatek pro menší reproduktory umístěné blízko posluchače, ale na ozvučení větší místnosti to nemusí být dostatečné.

Mám také zkušenost se zesilovačem TDA7250. Tento zesilovač by se svým výkonem 100 W na reproduktor určitě stačil. Tohoto výkonu ale dosahuje pouze při symetrickém napájení ± 35 V. Díky nutnosti použití právě symetrického napájení, které by ztížilo návrh, jsem ho ale zavrhl.

Místo něj jsem vybral zesilovač TDA7377. Tento zesilovač lze napájet asymetrickým napájením o napětí 8 až 18V. Jeho výkon dosahuje 2x30 W při zapojení typu double bridge a 4x10 W při zapojení pro 4 vstupy. Obsahuje také stand-by mód při kterém má prakticky nulovou spotřebu. Tento mód lze mimo jiné používat k utlumení výstupů za běhu. TDA7377 stojí 68 Kč. K funkčnosti také nepotřebuje nijak velký obvod. [30]



■ Obrázek 4.9 V levo zapojení 2x30W a v pravo zapojení 4x10W [30]

4.5 Rádio

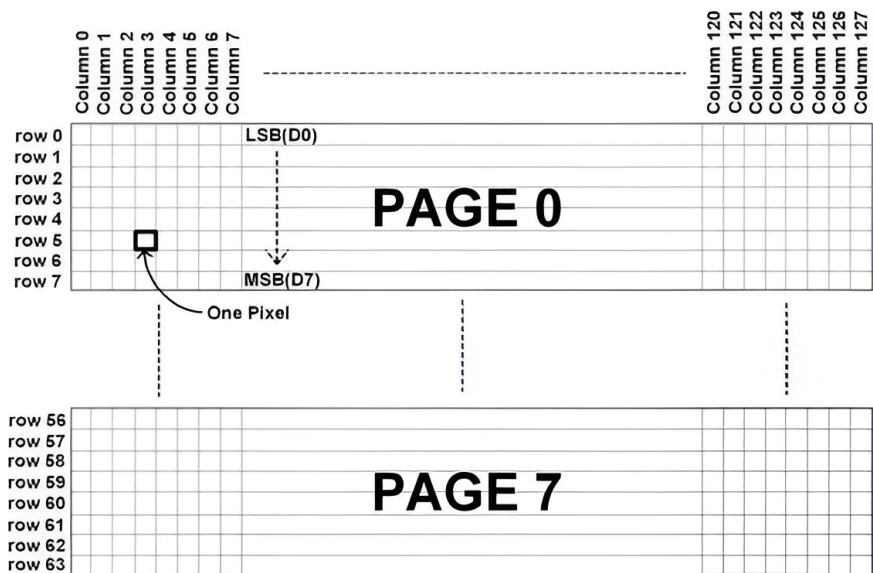
Pro integrované rádio, jsem se rozhodl použít modul rrd-102. Tento modul je velmi levný, stojí 28 Kč, ale plně dostačuje. Je napájen 3,3V a řízen pomocí I2C sběrnice. Tento modul již rovnou obsahuje samotný radiový čip RDA5807 spolu s krystalem o frekvenci 32,768 kHz. Podporuje rádiové frekvence od 50MHz do 115MHz. Zde v Česku používáme frekvence od 87MHz do 108MHz. Tento modul také dokáže přijímat RDS zprávy. Jeho výstupem je levý a pravý kanál se společnou zemí. [31]



■ Obrázek 4.10 Popis pinů modulu rrd-102 [32]

K zobrazování frekvence rádia jsem se rozhodl použít malý 0,96palcový oled displej. Tento displej je ovládaný pomocí I2C sběrnice. Na tomto displeji je možné zobrazovat i kratší texty. Toho lze využít například pro zobrazení popisu aktuálně zvoleného vstupu. Oled displej používá integrovaný obvod ssd1306. Tento obvod komunikuje přes i2c sběrnici. Displej má rozlišení 128*64 pixelů. Je rozdělen do 7 řádků a každý řádek je vysoký 8 pixelů, jak je znázorněno na obrázku 4.11. Výpis na displej probíhá po sloupci vysokém 8 pixelů. Pixel je rozsvícen, pokud je na jím odvozeném bitu logická 1.

Jakožto Infračervený přijímač pro dálkové ovládání jsem se rozhodl použít model VS1838B který stojí 6 Kč. Je vhodný pro příjem infračerveného záření o nosné frekvenci 38 KHz. Tento model lze napájet 5V. [34]



■ Obrázek 4.11 Struktura oled displeje [33]

Kapitola 5

Návrh Řešení

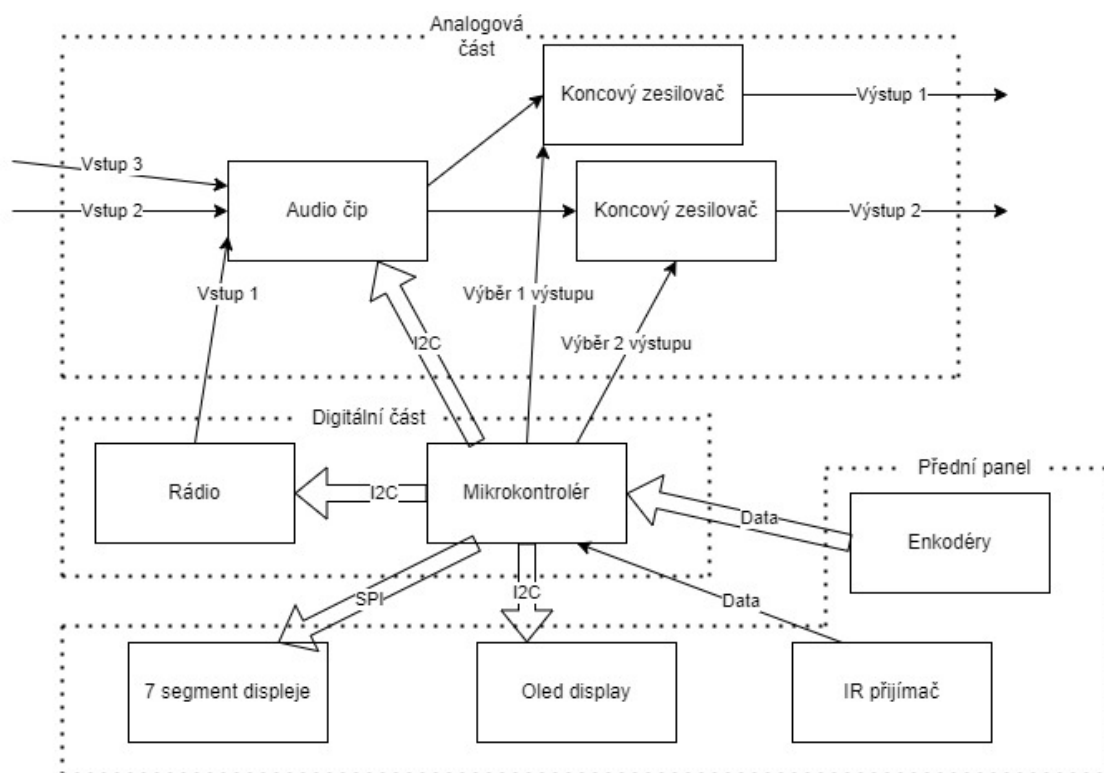
Celý zesilovač lze rozdělit na dvě části. Na Desku zesilovače, která zajišťuje úpravu vstupního signálu a zobrazuje hodnoty uživateli. A na aplikaci, která zajišťuje komunikaci mezi jednotlivými komponenty.

5.1 Deska zesilovače

Celou desku lze rozdělit na 3 části. Analogovou, digitální a na přední ovládací panel. Analogová část se bude starat o úpravu zvuku, digitální bude obsahovat mikrokontrolér ATmega32 a obvod potřebný k jeho funkčnosti. Bude obsahovat také rádio a převodník napětí pro dosažení 3,3 V potřebných pro čip rádia. Přední ovládací panel bude obsahovat Enkodéry, displeje a infračervený přijímač, pro příjem pokynů z ovladače. Na obrázku 5.1 je nakresleno blokové schéma desky s jejími důležitými bloky. Návrh desky bude proveden tak, aby bylo možné použít dvouvrstvého plošného spoje. Přední panel bude k desce připojen pomocí speciální sběrnice, která bude zároveň zajišťovat upevnění panelu kolmo ke zbytku desky.

5.2 Aplikace

Samotná aplikace vždy po spuštění inicializuje všechny komponenty do základního nastavení. Následně bude kontrolovat, jestli není otáčeno enkodérem nebo jestli není přijímán pokyn od ovladače. Samotná kontrola pokynů od ovladače bude kontrolována přes interrupt pin. Při příjmu pokynu ať už od ovladače nebo enkodéru provede akci k tomuto pokynu náležící. Aplikace bude k jednotlivým komponentám přistupovat pomocí I2C a SPI sběrníc. Aplikace bude umožňovat nastavení hlasitosti, úroveň hloubek, výšek, stereo váhy a dále výběr radio frekvence, vstupu a výstupu.



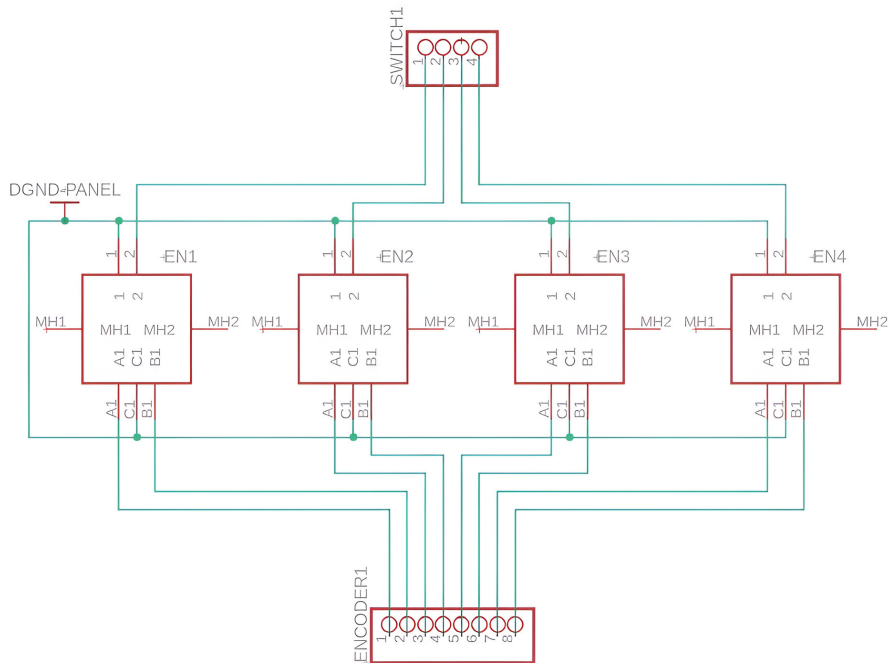
■ **Obrázek 5.1** Blokové schéma desky

Deska zesilovače

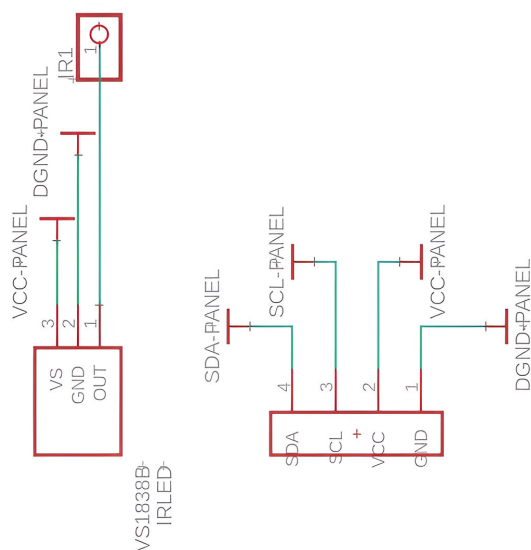
Pro návrh desky plošných spojů jsem se rozhodl použít program Fusion 360. Oproti mému prvnímu rozhodnutí použít program Eagle, zde lze zobrazit, jak bude osazená deska vypadat ve 3D. Tato funkce je opravdu důležitá pro návrh ovládacího panelu zesilovače. Program Fusion360 lze zakoupit u firmy autodesk a obsahuje i demo verzi, která je ovšem limitovaná velikostí navrhované desky.

6.1 Čelní panel

Na předním ovládacím panelu jsou přítomné enkodéry (obrázek 6.1), displej a přijímač infračerveného záření (obrázek 6.2). Tento panel je připojen k hlavní desce pomocí 20 pinů.

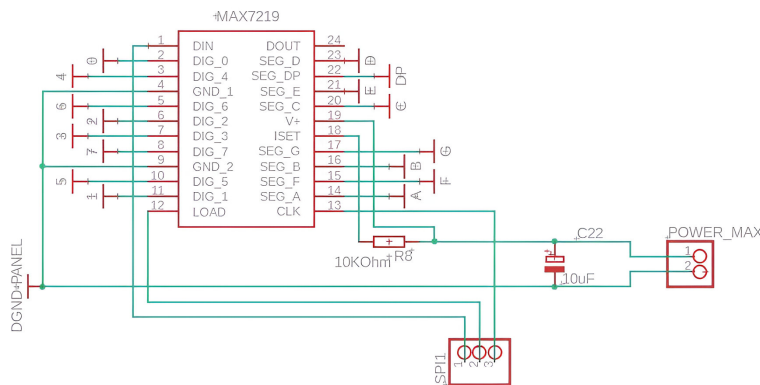


■ **Obrázek 6.1** Schéma zapojení enkodérů

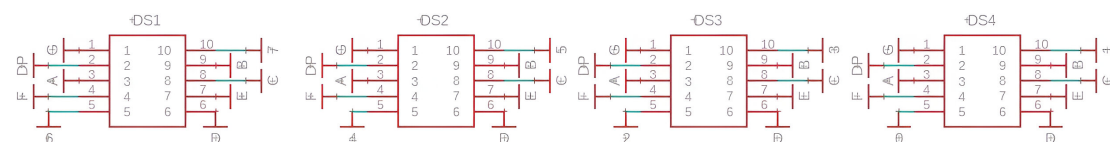


Obrázek 6.2 Schéma zapojení modulu VS1838 vlevo a oled displeje vpravo

Max7219 potřebuje ke správné funkci zdroj o napětí 5V a výkonu 5W. Proto je potřeba k němu přivést externí napájení. Pro stabilizaci napájení jsem se rozhodl použít elektrolytický kondenzátor. Jelikož lze do tohoto čipu pouze zapisovat, rozhodl jsem se použít pouze 3 vodiče z SPI sběrnice. Vodič MISO je nepotřebný. Celkové schéma zapojení čipu Max7219 lze nalézt na obrázku 6.3, zapojení 7segmentových displejů pak na obrázku 6.4.



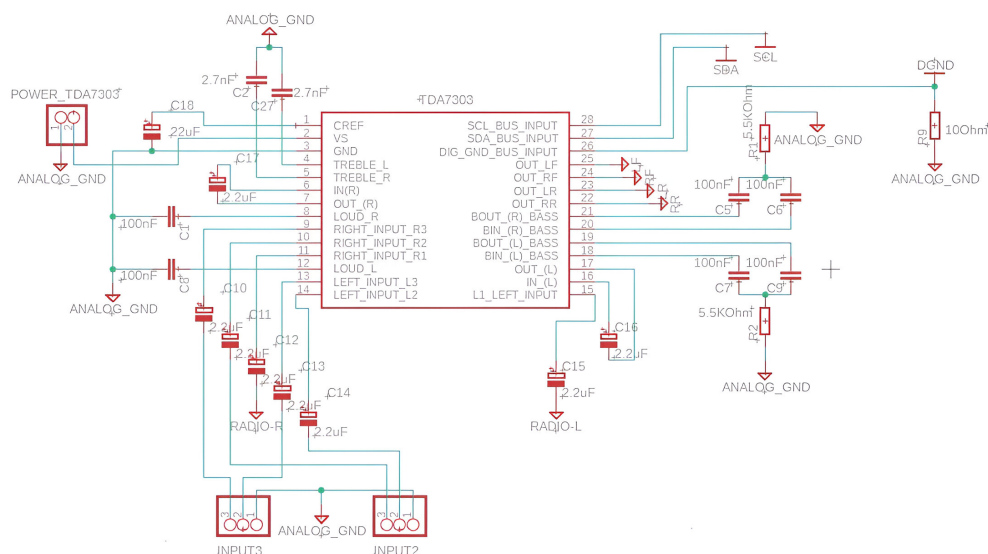
Obrázek 6.3 Schéma zapojení čipu MAX7219



Obrázek 6.4 Schéma zapojení 7segmentových displejů

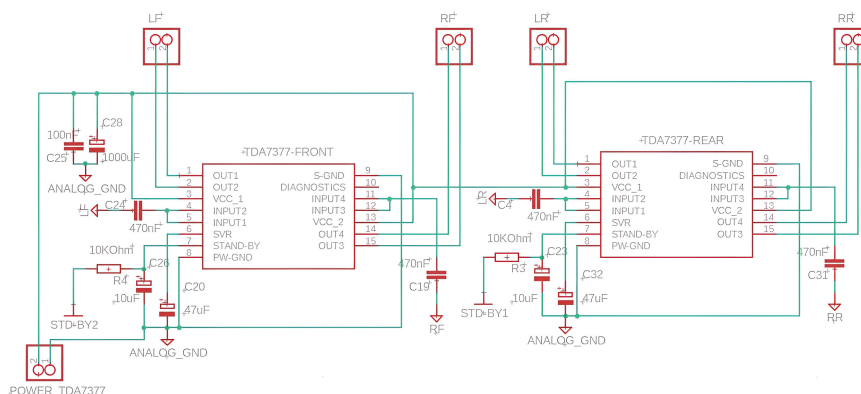
6.3 Analogová část

Analogová část obsahuje audio čip a koncové zesilovače. Jako první je zde audio čip TDA7303. Tento audio čip potřebuje na napájení 6 V až 10 V. Na vstupu lze vybírat mezi třemi kanály, přičemž na první kanál jsem interně připojil rádio. Prakticky celé schéma zapojení tohoto čipu je převzato z datasheetu [29]. K oddělení digitální země s analogovou je zde použit rezistor. Toto řešení zajistí pouze jeden přechod mezi těmito zeměmi. Pokud je k napájení všech částí zesilovače použit jeden zdroj a do všech částí je přivedena země z tohoto zdroje, nesmí být tento rezistor zapojen. Nebude tak zde docházet ke smyčkám. Celé schéma zapojení pro TDA7303 je na obrázku 6.7.



■ Obrázek 6.7 Schéma zapojení audio čipu TDA7307

TDA7303 obsahuje dva kanály na výstupu. Pro jejich plné využití jsem se rozhodl použít pro každý kanál koncový zesilovač TDA7377. Toto zapojení umožní kompletní vypnutí daného kanálu díky možnosti aktivování režimu stand-by na čipu korespondujícímu k danému kanálu. Potřebné napájecí napětí je 8 V až 18 V. Je zde použito zapojení double bridge. Schéma zapojení je převzato z datasheetu [30] a je na obrázku 6.8.

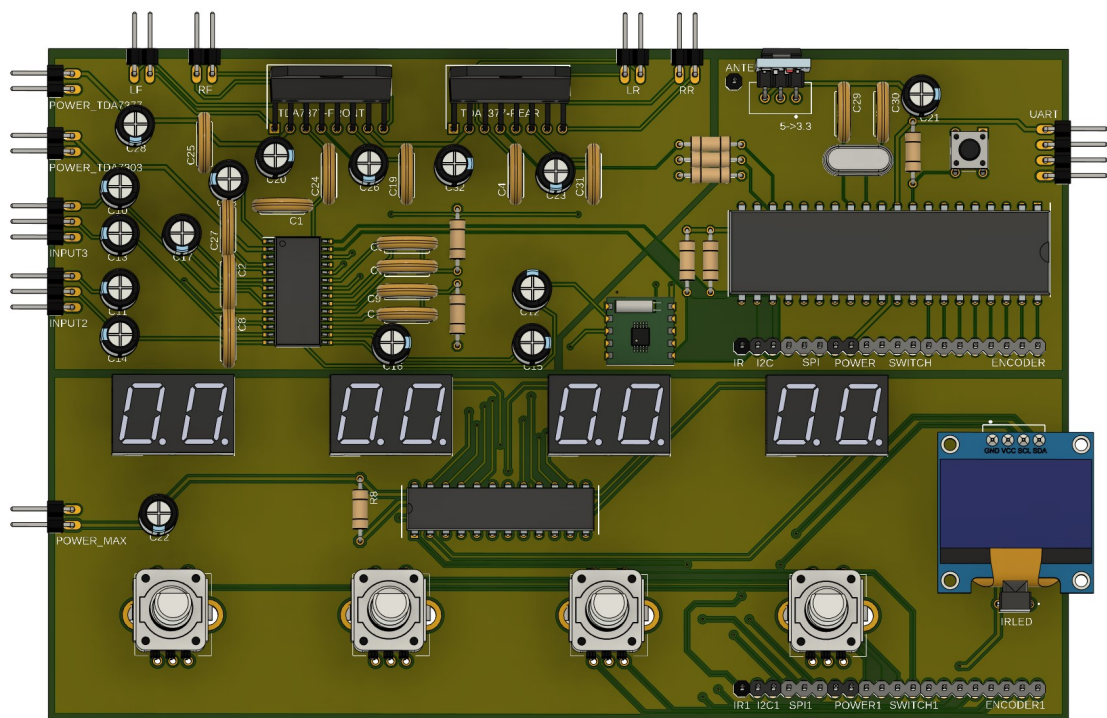


■ Obrázek 6.8 Schéma zapojení koncových zesilovačů TDA7377

6.4 Shrnutí

Celý obvod jsem navrhl jako jednu desku, která je zobrazena na obrázku 6.9. Jako hlavní důvod lze označit výrobní cenu. Tento způsob návrhu ovšem není úplně ideální, jelikož je nutné desku při montáži komponent rozříznout na dvě části. Tento řez musí být veden částí desky, která není použita pro vedení signálů.

Je také potřeba použít chlazení pro čipy TDA7377. K jejich chlazení lze použít hliníkový chladič, který lze k čipům přichytit pomocí otvoru v horní části čipu. K napájení jsem se rozhodl použít ATX spínaný zdroj s výstupem 24pinů. K těmto spínaným zdrojům lze koupit adaptér rozvádějící jednotlivá napětí ze zdroje na výstupní svorky jmenovitě 3,3 V, 5 V a 12 V. Já jsem si vybral adaptér, který disponuje i obvodem postaveným okolo čipu LM2596S. Čip LM2596S je spínaným napěťovým regulátorem. Napětí lze nastavit potenciometrem, který je součástí obvodu. Tento obvod dokáže produkovat napětí 4,5 V až 12 V. Je tedy ideální pro napájení čipu TDA7303 při nastavení 10V. Pro napájení jsem vybral ATX zdroj z důvodu jeho dostatečného výkonu, jehož standartní napájecí zdroje nedosahovaly.

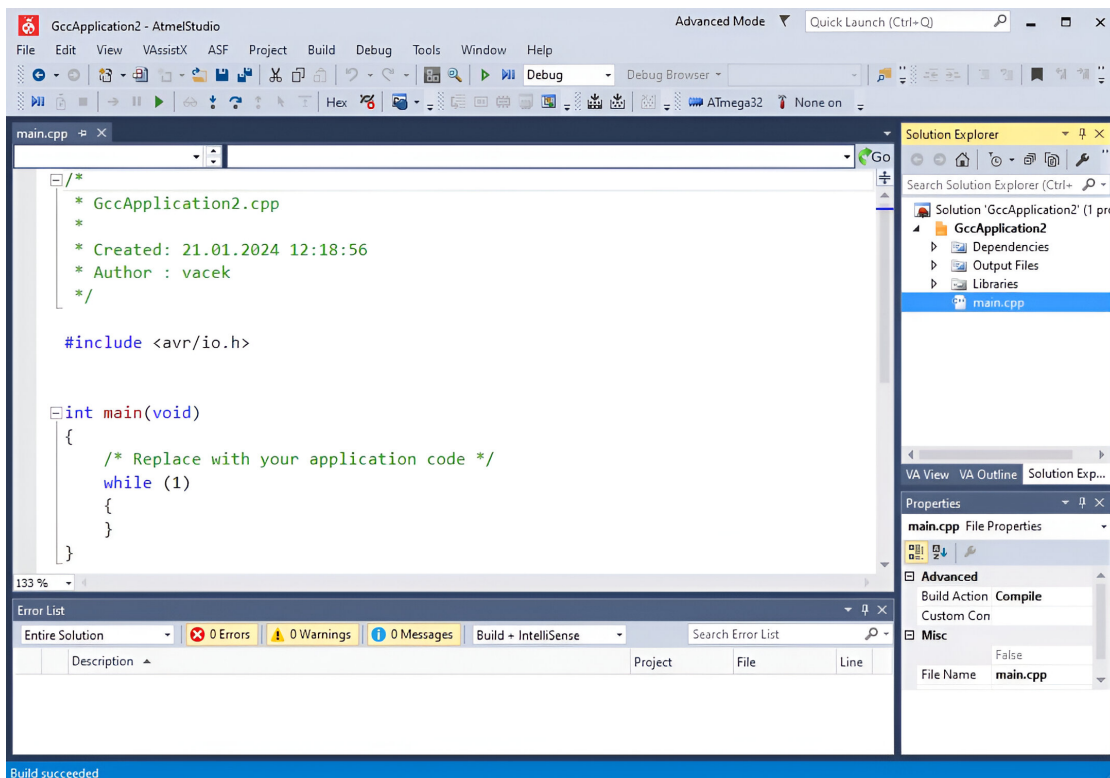


■ Obrázek 6.9 3D náhled výsledné desky

Kapitola 7

Aplikace

Žádný mikrokontrolér nebude plnit svou funkci, pokud nebude naprogramován. Pro psaní aplikace jsem si vybral Atmel studio. Studio disponuje potřebnými knihovnami pro programování AVR mikrokontrolérů. Lze zde psát programy v C/C++ a assembleru. Pro psaní nového programu je nutné vytvořit projekt. Já jsem se rozhodl vytvořit projekt v jazyce C++. Hlavní důvodem byla absence tříd v jazyce C, díky níž by se knihovny potom staly méně čitelnými. Atmel studio lze nainstalovat pouze na zařízení s operačním systémem Windows. Pro zařízení s operačním systémem Linux lze použít například Arduino IDE nebo Eclipse IDE.



Obrázek 7.1 Atmel studio - okno s výchozím programem

7.1 Atmel studio

V Atmel studiu lze dokonce vytvořit tzv. Arduino sketch (česky skica). V Atmel studiu lze tedy programovat úplně stejně jako v Arduino IDE se všemi Arduino knihovnami. Při tvorbě klasického projektu lze vybrat rovnou AVR čip který bude programován. Po vytvoření projektu je vytvořena složka projektu se souborem main. Pro psaní kódu pro mikrokontrolér je potřeba v programu nekonečná smyčka, jinak by kontrolér po chvíli dokončil program a došlo by k vypnutí.

7.2 Knihovna pro I2C komunikaci

Pro použití I2C komunikace jsem použil knihovnu `i2cmaster.h`. Jedná se o upravenou I2C knihovnu pro mikrokontrolér ATmega328p [35]. Na počátku programu je potřeba zavolat funkci `i2cMasterInit`, která akceptuje jako argument frekvenci I2C sběrnice. Metoda `i2cMasterStart` akceptuje adresu a mód. Zahajuje komunikaci se zařízením a nastavuje registr TWCR. Po zahájení komunikace lze používat metody:

- `i2cMasterWrite` – akceptuje jeden byte dat a zasílá ho do zařízení
- `i2cMasterReadAck` – přijímá jeden byte a následně zasílá potvrzení příjmu
- `i2cMasterReadNack` – přijímá jeden byte bez potvrzení

Po provedení veškerého přenosu lze použít metodu `i2cMasterStop`, která ukončí přenos nastavením registru TWCR. Pro ucelenou komunikaci lze používat metody:

- `i2cMasterSend` – akceptuje string a adresu, na kterou má zapsat
- `i2cMasterSendByte` – akceptuje byte a adresu, na kterou má zapsat
- `i2cMasterRecieve` – akceptuje adresu, ze které má přijímat

Tyto metody nabízejí kombinaci předešlých metod a jde o usnadnění psaní konečného kódu.

7.3 Knihovna pro MAX7219

Tento mikrocontroller má 14 registrů:

- 8 pro znaky (čísla), které jsou vypsány na příslušné displeje
- 1 pro nastavení módu dekodování znaků
- 1 pro nastavení intenzity svitu displejů
- 1 pro počet vypisovaných znaků (počet aktivních displejů)
- 1 pro zapnutí nebo vypnutí displejů
- 1 pro test čipu

Pomocí mód registru lze nastavit zobrazování čísel například pomocí BCD kódu. Díky tomu není potřeba tabulka pro převod čísel na 7segmentové znaky, mikrocontroller již má tento převod implementovaný tabulkou 7.2. Knihovna je napsána formou třídy, která si uchovává aktuálně vypsané hodnoty na displejích formou pole.

Table 5. Code B Font

7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6-D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

*The decimal point is set by bit D7 = 1

■ **Obrázek 7.2** Tabulka pro převod na znaky pomocí BCD kódu

7.3.1 Konstruktor

Konstruktor třídy Max7219 tedy spočívá v inicializaci SPI rozhraní a v nastavení registrů přítomných v čipu MAX7219. Jako parametry akceptuje intenzitu displeje a počet znaků. Metoda inicializuje komunikaci pomocí SPI sběrnice. Poté nastaví možnost zobrazování čísel pomocí BCD kódu, nastaví počet znaků, intenzitu svitu a umožní čipu vypisovat na displeje. Nakonec nastaví všechny zobrazené hodnoty na 0.

7.3.2 MAX7219ClearDisplay

Pro vymazání všech znaků je použita metoda MAX7219ClearDisplay. Tato metoda zapíše do všech znakových registrů hodnotu 0xF. Tato hodnota je v integrované převodní tabulce brána jako prázdný znak, tedy znak, u kterého nejsou aktivní žádné segmenty.

7.3.3 MAX7219UpdateNumber

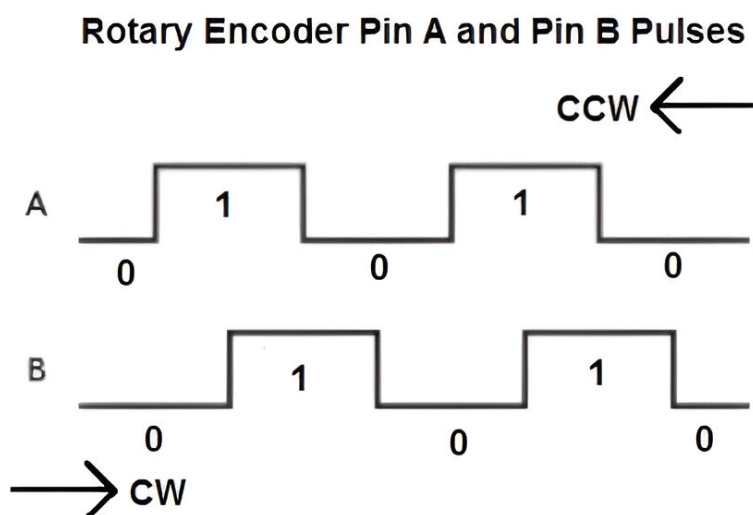
Max7219 umožňuje pro změnu jednoho konkrétního znaku na výstupu měnit pouze registr k němu náležící. Jelikož tento návrh používá 4 dvou znakové displeje, je potřeba měnit vždy dva znaky (desítkovou pozici a jednotkovou). Jednotlivé hodnoty zesilovače lze upravovat metodou MAX7219UpdateNumber. Tato metoda akceptuje jako argument pozici (číslo displeje). Tento displej poté smaže a vypíše na něm novou hodnotu přítomnou na dané pozici v poli vypsáních hodnot. Díky tomuto postupu není v případě hodnoty menší než 10 zobrazena 0 na desítkovém místě ale pouze jednotka.

7.3.4 MAX7219UpdateAll

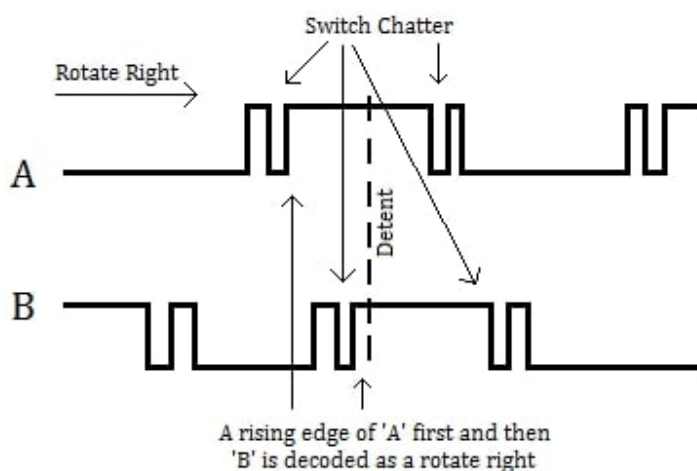
Pro aktualizaci všech hodnot na jednou lze použít metodu MAX7219UpdateAll. Tato metoda akceptuje na vstupu pole aktuálních hodnot o délce 4. Tyto hodnoty porovná s již zobrazenými hodnotami. Pokud se některá hodnota nerovná s již zobrazenou, vypíše jí. Tento přístup zamezí neustálému přepisování všech hodnot při změně jedné a tím pádem zrychlí průměrnou dobu výpisu.

7.4 Knihovna pro rotační enkodér

Výstupem enkodéru jsou ideálně dva obdélníkové signály fázově posunuté o 90°. Dle fázového posunu určíme směr, a dle počtu impulzů můžeme spočítat o kolik byl enkodér otočen (obrázek 7.3). Standardně se ale mohou na mechanickém enkodéru objevit zákmity, jak je znázorněno na obrázku 7.4.



■ **Obrázek 7.3** Ideální průběh enkodérů



■ **Obrázek 7.4** Průběh enkodérů se zákmity

Pro odstranění zákmitů jsem se rozhodl naprogramovat enkodér jako automat. Pro čtení a interpretaci dat z enkodéru lze použít knihovnu Encoder.h, Tato knihovna obsahuje třídu Encoder. Konstruktor třídy na vstupu přijímá port, ke kterému je enkodér připojen, dále pak 2 piny daného portu ke kterým jsou připojeny výstupní signály enkodéru. Spínací kontakty tohoto enkodéru spínají jeho výstupní piny se zemí. Proto je nutné na těchto pinech aktivovat pull-up rezistory, nebo je připojit k těmto kontaktům externě. Při inicializaci je také inicializován vnitřní stav a směr otáčení na nulu.

7.4.1 readEncoder

Pro čtení hodnot z enkodéru lze použít metodu `readEncoder`. Tato metoda obsahuje stavový automat, který reaguje na jednotlivé změny na výstupních signálech. Poté co je dokončeno otočení o jeden krok, je zapsán směr otočení.

7.4.2 getValue

Pro přečtení směru otočení lze použít metodu `getValue`. Tato metoda vrátí směr otočení a resetuje směr. Je nutné v každém cyklu programu provést jak metodu `readEncoder` pro obnovení směru, tak metodu `getValue` pro přečtení směru. Třída `encodér` neukládá počet již provedených otočení.

7.5 Knihovna pro RDA5807

Tento čip komunikuje pomocí I2C sběrnice. Podporuje frekvenci sběrnice až 400kHz. Nastavení se provádí pomocí 14 registrů umístěných na čipu. Pro správnou funkčnost v Evropě je potřeba nastavit registry 02h (tabulka 7.1), 04h (tabulka 7.2), 05h (tabulka 7.3), 07h (tabulka 7.4)

■ Tabulka 7.1 Obsah registru 02h [31]

15. bit	Nastavení výstupního modu (vysoká impedance nebo normální mód)
14. bit	Nastavení ztlumení
13. bit	Přepíná mezi mono/stereo
12. bit	Povoluje Bass boost
11. bit	Nastavuje, zda jsou referenční hodiny neustále dostupné
10. bit	Nastavuje vstupní mod referenčních hodin
9. bit	Nastavuje směr vyhledávání stanic
8. bit	Nastavení zda automatické vyhledávání stanic skončí při nalezení stanice a nebo prohledá celé spektrum.
7. bit	Nastavení zda automatické vyhledávání stanic skončí při dosažení konce spektra, anebo bude pokračovat od začátku.
6. – 4. bit	nastavení frekvence krystalu.
3. bit	Zapnutí příjmu rds zpráv
2. bit	Nastavení metody příjmu
1. bit	Nastavení resetu
0. bit	Zapnutí napájení

Pro naladění žádané frekvence je poté nutné vždy zapsat nové hodnoty do registru 03h. Pomocí vzorce je nejdříve potřeba spočítat hodnotu k zápisu odvozenou z žádané frekvence. Vzorec výpočtu pro frekvenční rozsah 87 – 108 MHz: $(\text{Frekvence} - 87) * 10$. Odvozená hodnota se zapisuje do 15. – 6. bitu. Navíc je potřeba zapsat do 4. bitu logickou jedničku pro povolení ladění [31]. Pro ovládání čipu RDA5807 jsem použil knihovnu `Radio.h`. Tato knihovna obsahuje třídu `Radio`. Konstruktor třídy nejdříve načte do registrů potřebná data pro inicializaci, která jsou uložena v poli. Poté načte do registrů data potřebná k naladění frekvence 87MHz. S rádiem lze poté interagovat s pomocí metod:

- `nextStation` – akceptuje jako parametr směr ladění a posouvá frekvenci o 100 kHz v zadaném směru.
- `selectedStation` – vrací nastavenou frekvenci v MHz.

■ **Tabulka 7.2** Obsah registru 04h [31]

14. bit	Nastavení přerušení při nalezení stanice. Pokud je tento bit aktivován, bude generovaný 0 pulz na pinu GPIO2 vždy když bude aktivováno toto přerušení.
13. bit	Nastavení módu RBDS
12. bit	Nastavení fronty pro RDS
11. bit	Nastavení procesu pro vyrovnání frekvenčního zkreslení
10. bit	Mazání RDS
9. bit	Nastavení softmute
8. bit	Nastavení zda automatické vyhledávání stanic skončí při nalezení stanice a nebo prohledá celé spektrum.
6. bit	Nastavení I2S rozhraní. I2S je standardní sériové rozhraní používané pro přenos digitálního zvuku mezi integrovanými obvody
5. – 0. bit	Nastavení I/O pinů

■ **Tabulka 7.3** Obsah registru 05h [31]

15. bit	Nastavuje interrupt. Pokud je nastaven do hodnoty 1 bude vygenerované přerušení trvat do chvíle než bude přečten registr 0CH. jinak bude přerušení trvat 5ms.
14. – 13. bit	Povolení RSSI seek modu. RSSI slouží k posouzení síly signálu od různých stanic.
11. – 8. bit	Nastavení minimální hodnoty SNR, kterou musí přijímač detekovat na dané frekvenci během hledání stanic, aby byla tato stanice považována za platnou. SNR (Signal-to-Noise Ratio) je poměr mezi silou signálu a úrovní šumu v přijímaném signálu.
7. – 6. bit	Nastavení anténního vstupu
5. – 4. bit	Nastavení pracovního proudu v LNA. LNA znamená "Low Noise Amplifier" a je často používán v kontextu rádiových přijímačů k označení zesilovače s nízkým šumem.
3. – 0. bit	Nastavení hlasitosti.

■ **Tabulka 7.4** Obsah registru 07h [31]

14. – 10. bit	Nastavení hladiny šumu, při které bude signál digitálně upraven
9. bit	Tento pin se nastavuje, pouze pokud je zvoleno frekvenční pásmo 65 – 75 MHz
7. – 2. bit	Stanovuje úroveň signálu nutnou k akceptování stanice při automatickém hledání
1. bit	Povoluje úpravu signálu při výskytu šumu
0. bit	Nastavení frekvenčního módu.

7.6 Knihovna pro oled displej

Pro komunikaci s tímto displejem jsem použil knihovnu Display.h. Knihovna obsahuje třídu Display. Při inicializaci tohoto displeje je potřeba zapsat inicializační sekvenci do command registru. Tato sekvence je popsána v tabulce 7.5.

■ **Tabulka 7.5** Úvodní sekvence pro inicializaci oled displeje [33]

Příkaz	Vysvětlení
0xAE	Vypnutí displeje (Display Off)
0xD5	Nastavení děličky hodin (Display Clock Divide Ratio/Oscillator Frequency)
0x80	Dělička = 0x80 (nastavení podle potřeby)
0xA8	Nastaví multiplexní poměr (Multiplex Ratio)
0x3F	Poměr = 0x3F (128x64 displej)
0xD3	Nastaví posun displeje (Display Offset)
0x00	Posun = 0
0x40	Nastaví startovní linku (Start Line)
0x8D	Charge Pump
0x14	povolení Charge Pump
0xA1	Nastaví přepínač segmentu
0xC8	Nastaví COM režim (COM Output Scan Direction)
0xDA	Nastaví konfiguraci hardwaru COM pinů
0x12	Alternativní konfigurace COM pinů
0x81	Nastaví kontrast
0xCF	Kontrast = 0xCF
0xD9	Nastaví přednastavené fáze displeje (Pre-charge Period)
0xF1	Hodnota přednastavené fáze displeje (nastavení podle potřeby)
0xDB	Nastaví úroveň VCOMH (VCOMH Deselect Level)
0x40	VCOMH = 0x40
0xA4	Nastaví výstupní režim (Output RAM to Display)
0xA6	Nastaví zobrazovací režim (Normal Display)
0xAF	Zapnout displej (Display On)

7.6.1 clearFullDisplay

Pro vymazání displeje lze použít metodu `clearFullDisplay`. Tato metoda maže displej bit po bitu, to znamená, že na každé místo v datové paměti displeje je zapsána 0.

7.6.2 printText

Pro displej jsem napsal speciální font obsahující čísla a malá písmena abecedy. Pro výpis textu na displej s pomocí vytvořeného fontu lze využít metodu `printText`. Tato metoda akceptuje string a vypíše tento string na displej.

7.7 Knihovna pro TDA7303

Čip TDA7303 také komunikuje pomocí I2C sběrnice. Obsahuje 8 registrů, jak je popsáno v tabulce 7.6. Pro nastavení registrů je potřeba místo znaků x v adrese zadat konkrétní hodnotu.

■ **Tabulka 7.6** Adresy registrů čipu TDA7303 [29]

Registr	Adresa
Nastavení hlasitosti	00xxxxxx
Výstup levý zadní	110xxxxx
Výstup pravý zadní	111xxxxx
Výstup levý přední	100xxxxx
Výstup pravý přední	101xxxxx
Přepínač vstupů	010xxxxx
Ovládání basů	0110xxxx
Ovládání výšek	0111xxxx

Pro komunikaci s tímto čipem jsem použil knihovnu TDA7313.h. Tato knihovna byla určena pro čip TDA7313 ale ten se prakticky neliší od zde použitého čipu TDA7303. Knihovna obsahuje třídu TDA7313. Jelikož nelze vyčíst z čipu jeho nastavení, je nutné, aby bylo veškeré nastavení uloženo v paměti mikrokontroléru.

V konstruktoru třídy probíhá počáteční nastavení čipu. Knihovna poté obsahuje metody pro nastavení vnitřních registrů čipu:

- `toggleInput` – rotuje mezi vstupy čipu
- `toggleOutput` – rotuje mezi výstupy čipu
- `toggleVolume` – snižuje/zvyšuje hlasitost v závislosti na argumentu
- `toggleBass` – nastavuje úroveň hloubek v závislosti na argumentu
- `toggleTreble` – nastavuje úroveň výšek v závislosti na argumentu
- `toggleOutput` – rotuje mezi výstupy čipu (pouze přední kanál, pouze zadní kanál, oba dohromady)
- `toggleBalance` – nastavuje stereo váhu v závislosti na argumentu

Poté obsahuje metody pro získání hodnot vnitřních registrů:

- `getInput`
- `getOutput`
- `getVolume`
- `getBass`
- `getTreble`
- `getOutput`
- `getBalance`

7.8 Knihovna pro IR přijímač

Výstup z infračerveného přijímače je při nečinnosti je v logické 1. Infračervený přijímač je připojený k pinu, který obsahuje vnější přerušení. Pro povolení vnějšího přerušení je potřeba správně nastavit registry mikrokontroléru: Registr GICR pro povolení vnějšího přerušení a registr MCUCR pro nastavení, kdy bude přerušení aktivováno. Možnostmi jsou náběžná, spádová nebo obě hrany.

Pro komunikaci s infračerveným přijímačem jsem upravil knihovnu `remote.h` [36]. Tato knihovna obsahuje metody `ResetIR`, `RemoteInit` a `GetRemoteCmd`. Na začátku je nutné zavolat metodu `RemoteInit`, která inicializuje vstupní přerušení na pinu `INT0` a časovač použitý k měření jednotlivých časových intervalů. Následně je vždy při příjmu dat pomocí přijímače provedeno přerušení, které provádí postupný příjem dat. Příjem dat je proveden ve formě stavového automatu. Knihovna obsahuje časovou toleranci 10% jelikož každý ovladač nemusí být tak přesný. Například levnější ovladače nepoužívají součástky s takovou přesností. Pokud se proces příjmu dat dostane do nedefinovaného stavu, je proveden reset příjmu dat. Po úspěšném příjmu dat a úspěšné kontrole dat, lze data přečíst pomocí metody `GetRemoteCmd`. Tato metoda vrátí buď nově přijatá data, anebo hodnotu 255. Tato hodnota indikuje, že žádná nová data přijata nebyla.

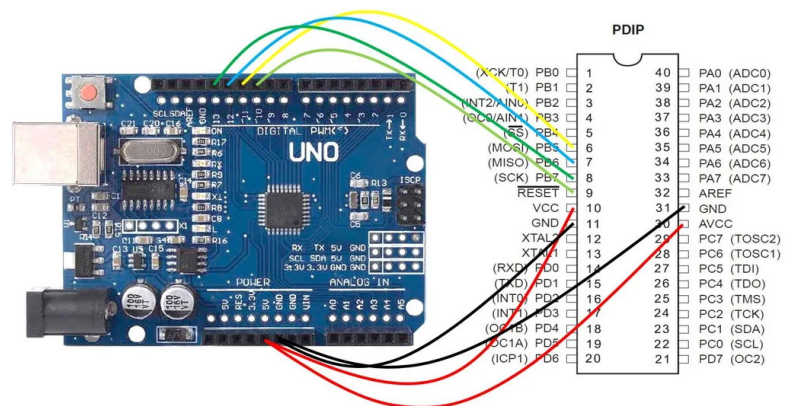
7.9 Nahrání kódu do mikrokontroléru

Kód lze do mikrokontroléru nahrát dvěma způsoby. Pro první způsob je potřeba nejprve uložit do mikrokontroléru speciální program, který se nazývá bootloader. Tento program umožňuje nahrávání nového programového kódu přes sériové rozhraní (například UART, USB, nebo SPI) bez použití externího programátoru. Využití bootloaderu pro ATmega32 má několik výhod:

- **Flexibilita:** Bootloader umožňuje aktualizaci firmware mikrokontroléru přímo z aplikace, což znamená, že nemusíte mít fyzicky připojený externí programátor k mikrokontroléru. To může být užitečné, pokud je mikrokontrolér umístěn v zařízení, kde není snadný přístup k programovacímu portu.
- **Jednoduchost aktualizace:** Bootloader umožňuje aktualizaci firmware bez nutnosti zásahu do fyzického zapojení mikrokontroléru. To může zjednodušit proces aktualizace firmware, zejména pokud se mikrokontrolér nachází v zařízení, které je již distribuováno.
- **Rychlost vývoje:** Bootloader umožňuje rychlou aktualizaci kódu během vývoje a testování nových funkcí nebo oprav chyb. To může zkrátit čas potřebný k vyvíjení nových verzí firmware.

Existuje několik bootloaderů dostupných pro ATmega32, včetně Bootloaderu Arduino (`optiboot`), který je populární pro Arduino desky. Jakmile je bootloader nahrán do mikrokontroléru, lze použít software pro nahrání kódu (například `avrdude` nebo Arduino IDE) k aktualizaci firmware přes sériové rozhraní podle specifikací bootloaderu. Použití bootloaderu vyžaduje určitou část paměti mikrokontroléru, což může ovlivnit dostupný prostor pro programový kód. Je tedy důležité zvážit tuto skutečnost při plánování velikosti vaší aplikace a využití bootloaderu.

Druhý způsob spočívá ve využití externího programátoru: Pro tento způsob je použita SPI sběrnice spolu s resetovacím pinem. Jako externí programátor lze použít Arduino desku (obrázek 7.5) ve které musí být nahrán Arduino ISP program, anebo modul k tomu přímo určený například `usbasp`. Tento způsob nepotřebuje žádnou další paměť na mikrokontroléru. Při každém nahrání kódu je potřeba čip připojit k externímu programátoru. Pro finální produkt se tedy více hodí druhý způsob.

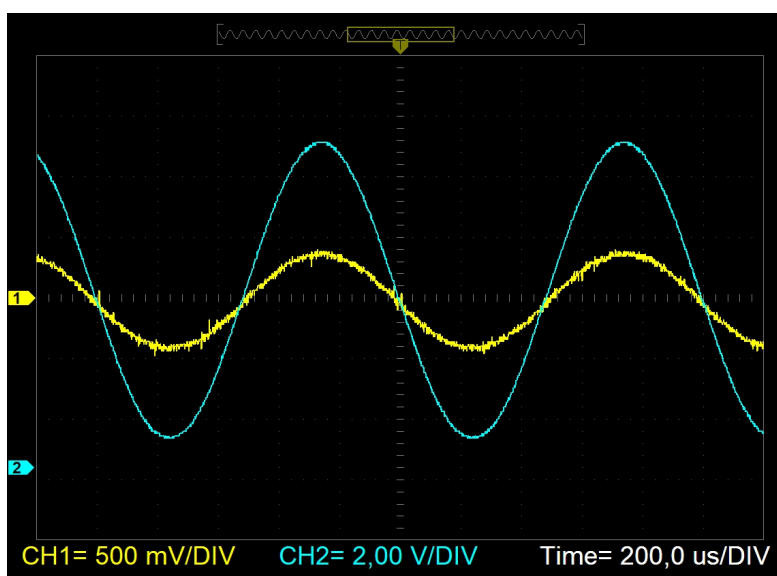


■ Obrázek 7.5 Využití arduino desky jako externího programátoru

Kapitola 8

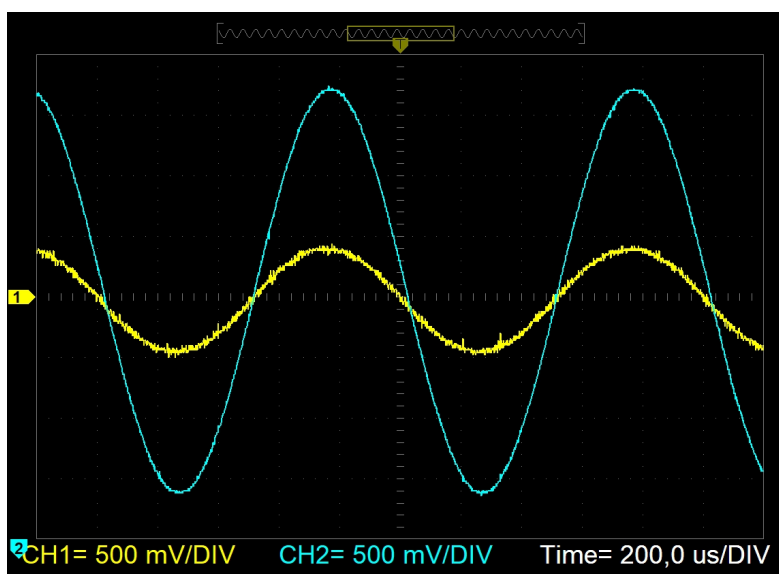
Testování

Pro otestování obvodu jsem připojil na zvukový vstup generátor sinusového napětí, na kterém jsem zvyšoval frekvenci. Zároveň jsem měnil úroveň hlasitosti na zesilovači. Jednotlivé průběhy jsem porovnával na dvoukanálovém osciloskopu, který jsem připojil na výstup zesilovače. Tímto způsobem jsem tedy zároveň testoval jak funkčnost ovládání, tak kvalitu zesílení signálu. Výsledná měření lze shlédnout na obrázcích 8.1, 8.2, 8.3. Vstupní signál je zbarven žlutě a výstupní signál modře. Z měření je patrné, že je výstup čistý, bez nějakého většího zkreslení oproti vstupu.

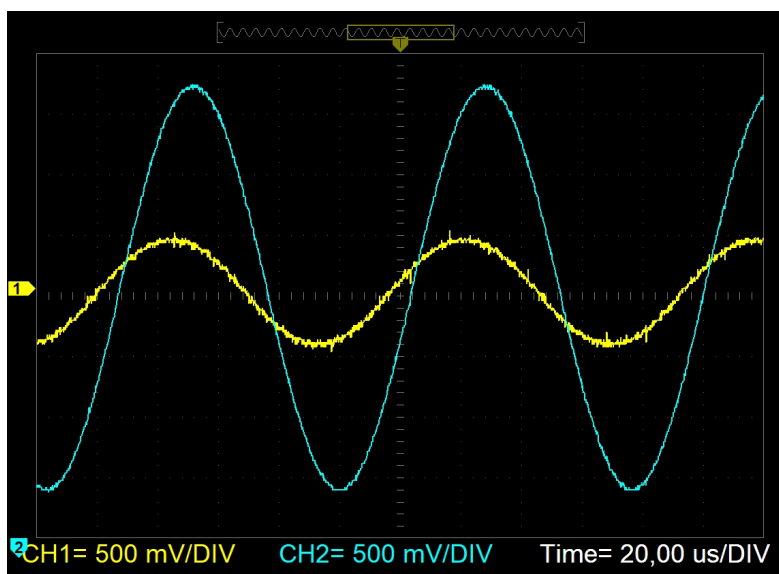


■ **Obrázek 8.1** 1 kHz na vstupu, zesílení nastaveno na hodnotu 56

Na obrázcích 8.1 a 8.2 je patrné, že při vstupní frekvenci 1 kHz prakticky neexistuje žádný fázový posun. Signál na výstupu je navíc čistý a neobsahuje žádné zkreslení.



■ **Obrázek 8.2** 1 kHz na vstupu, zesílení nastaveno na hodnotu 47



■ **Obrázek 8.3** 10 kHz na vstupu, zesílení nastaveno na hodnotu 47

Na tomto obrázku lze již pozorovat fázový posun. Je to dáno tím, že na frekvenci 10 kHz již působí filtr pro nastavení úrovně výšek. Toto ale na kvalitu výstupu nemá žádný vliv. Samotný výstupní signál je opět čistý a neobsahuje žádné zkreslení.



Kapitola 9

Závěr

Výsledkem této práce je plně funkční analogový audio zesilovač. Tento zesilovač je navíc jednoduchý na sestavení a tudíž se hodí i jako stavebnice pro lidi, kteří začínají s pájením a chtějí si vyrobit něco smysluplného, co poté budou moci denně používat. Cena se po spočtení cen všech použitých komponent a výroby desky bez zdroje pohybuje okolo 1100 Kč pokud jsou komponenty zakoupené v obchodech v České republice. Už při této ceně je tedy zesilovač více než konkurence schopný. Pokud by byly komponenty zakoupeny v Číně např. v internetovém obchodě Aliexpress je možno snížit cenu až na polovinu. Osobně jsem ve svém prototypu použil čipy jak z českých obchodů, tak i z čínských a nerozpoznal jsem rozdíl.

Při výběru a úpravě softwarových knihoven pro mě byla stěžejní jejich velikost po kompilaci a rychlost. Díky tomu je možné při tvorbě tohoto zesilovače použít mikrokontrolér ATmega16. Rozdíl oproti mikrokontroléru ATmega32 je pouze ve velikosti paměti (ATmega32 má 32 kB a ATmega16 má 16 kB) a v ceně. Lze tedy výslednou cenu ještě snížit použitím tohoto mikrokontroléru.

Cílem této bakalářské práce bylo na základě průzkumu trhu navrhnout konkurence schopný analogový audio zesilovač s dálkovým ovládním a ovládním pomocí enkodérů. Cíle této bakalářské práce tedy byly splněny. Zesilovač je cenově konkurence schopný a jak ukazují měření, nemá prakticky žádné zkreslení. Navíc si ho může sestavit doma prakticky každý. Zesilovač je nadále oproti ostatním vyjímečný svou otevřeností systému. Zdrojové kódy a schéma zapojení jsou volně dostupné. Každý kdo alespoň ovládá základy programování v C++ si může ovládní zesilovače přizpůsobit k obrazu svému. Navíc jsem na desce zhotovil výstup pro USART rozraní. Tudíž lze jednoduše s mikrokontrolérem po tomto rozhraní komunikovat. Nabízí se zde tedy jako rozšíření bluetooth modul, anebo připojení počítače a tím i možnost ovládní zesilovače přes bluetooth nebo internet.



Příloha A

GIT repozitář

Vše potřebné k vyhotovení zesilovače je volně dostupné na adrese <https://github.com/KrystofV/AnalogAudioAmplifier>

Bibliografie

1. A.S., Alza. *LTC audio ATM6500BT - HiFi zesilovač* — Alza.cz — alza.cz [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://www.alza.cz/ltc-audio-atm6500bt-d7859396.htm>.
2. A.S., Alza. *Fonestar AS-1515* — Alza.cz — alza.cz [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://www.alza.cz/fonestar-as-1515-d6229975.htm>.
3. A.S., Alza. *Marantz PM6007 stříbrno-zlatý* — Alza.cz — alza.cz [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://www.alza.cz/marantz-pm6007-stribrno-zlaty-d6158276.htm>.
4. SEDRA, Adel S; SMITH, Kenneth Carless. *Microelectronic circuits*. 5. vyd. New York: OXFORD UNIVERSITY PRESS, 2004. ISBN 0-19-514252-7.
5. ČERMÁK, Jindřich. *Kurs polovodičové techniky*. Praha: SNTL - nakladatelství technické literatury, 1976. ISBN 04-540-76.
6. HANUS, Stanislav; SVAČINA, Jiří. *Vysokofrekvenční a mikrovlnná technika přednášky*. Brno: FEKT VUT, 2002. ISBN 80-214-2222-X.
7. MYKLEBUST, Gaute. *The AVR microcontroller and C compiler co-design* [online]. Atmel Corporation, 2024 [cit. 2024-04-13]. Dostupné z: <https://docplayer.net/20751047-The-avr-microcontroller-and-c-compiler-co-design-dr-gaute-myklebust-atmel-corporation-atmel-development-center-trondheim-norway.html>.
8. DAGA, Akshay. *AVR Microcontroller : All You Need To Know- (Part 1/46)* [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://www.engineersgarage.com/avr-microcontroller-all-you-need-to-know-part-1-46/>.
9. GARBETT, Samuel L. *Exploring the Different Types of Arduino Microcontroller* [online]. 2022. [cit. 2024-04-13]. Dostupné z: <https://www.makeuseof.com/exploring-different-types-of-arduino-microcontroller/>.
10. *ATmega32 Datasheet* [online]. 2003. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/77378/ATMEL/ATMEGA32.html>.
11. BUREAU, EFY. *Arduino For Atmega32* [online]. 2016. [cit. 2024-04-13]. Dostupné z: <https://www.electronicsforu.com/electronics-projects/arduino-atmega32>.
12. BERGMANS, San. *IR Remote Control Theory* [online]. 2001, 2023-12-31. [cit. 2024-04-13]. Dostupné z: <https://www.sbprojects.net/knowledge/ir/index.php>.
13. BERGMANS, San. *NEC Protocol* [online]. 2001, 2023-12-31. [cit. 2024-04-13]. Dostupné z: <https://www.sbprojects.net/knowledge/ir/nec.php>.
14. BERGMANS, San. *Philips RC-5 Protocol* [online]. 2001, 2023-12-31. [cit. 2024-04-13]. Dostupné z: <https://www.sbprojects.net/knowledge/ir/rc5.php>.

15. MURPHY, Brendan. *What is the SPI Protocol and How to Debug SPI Communication?* [online]. 2020, 2023-12-31. [cit. 2024-04-13]. Dostupné z: <https://www.totalphase.com/blog/2020/07/what-is-spi-protocol-how-to-debug-spi-communication/>.
16. ATMEL MICROCHIP. *AVR151: Setup and Use of the SPI* [online]. 2016. [cit. 2024-04-13]. Dostupné z: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ApplicationNotes/ApplicationNotes/Atmel-2585-Setup-and-Use-of-the-SPI_ApplicationNote_AVR151.pdf.
17. SHARMA, Meetu; KHAN, Afia. *I2C Interface Timing Specifications and Constraints* [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://www.design-reuse.com/articles/54776/i2c-interface-timing-specifications-and-constraints.html>.
18. JUSTDOELECTRONICS. *Understand I2C Communication Protocol* [online]. 2023. [cit. 2024-04-13]. Dostupné z: <https://justdoelectronics.com/understand-i2c-communication-protocol/>.
19. DAGA, Akshay. *AVR ATmega32 TWI Registers* [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://www.engineersgarage.com/avr-atmega32-twi-registers/>.
20. ALLDATASHEET.COM. *TM1637 Datasheet* [online]. Titan micro electronics, 2024 [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1133627/TITAN/TM1637.html>.
21. ALLDATASHEET.COM. *MAX7219 Datasheet* [online]. Maxim electronics, 1997 [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/73745/MAXIM/MAX7219.html>.
22. ELECTRONICS, Mouser. *PDB183-S425K-104A3* [online]. 2024. [cit. 2024-04-13]. Dostupné z: <https://cz.mouser.com/ProductDetail/Bourns/PDB183-S425K-104A3?qs=pxDZ1BjcsCjHhV7oDQ0fZw%3D%3D>.
23. ALLDATASHEET.COM. *PDB183-GTR datasheet* [online]. Bourns Electronic Solutions, 2011 [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/521840/BOURNS/PDB183-GTR.html>.
24. PINI, Art. *How to use rotary encoders to quickly convert mechanical rotation into digital signals* [online]. DigiKey's North American Editors, 2018-12 [cit. 2024-04-13]. Dostupné z: <https://www.digikey.com/en/articles/how-to-use-rotary-encoders-convert-mechanical-rotation-digital-signals>.
25. *Rotary Encoder* [online]. 2016. [cit. 2024-04-13]. Dostupné z: <https://drive.google.com/file/d/19qLF55I4gEzkkV6Szd9oB6P018Gn08no/view?pli=1>.
26. RENESAS. *X9C104 Datasheet* [online]. 2009. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1058842/RENESAS/X9C104.html>.
27. TEXAS INSTRUMENTS. *TL072CP Datasheet* [online]. 1978. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/28807/TI/TL072CP.html>.
28. TEXAS INSTRUMENTS. *LM1036 Datasheet* [online]. 1995. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/836825/TI1/LM1036.html>.
29. STMICROELECTRONICS. *TDA7303 Datasheet* [online]. 2006. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/207468/STMICROELECTRONICS/TDA7303.html>.
30. STMICROELECTRONICS. *TDA7377 Datasheet* [online]. 1998. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/82979/STMICROELECTRONICS/TDA7377.html>.
31. RDA MICROELECTRONICS. *RDA5807M Datasheet* [online]. 2014. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1741404/RDA/RDA5807M.html>.

32. SOKOL, Sergey. *I2C FM Radio - Part 15 Microcontroller Basics* [online]. 2020. [cit. 2024-04-13]. Dostupné z: <https://www.circuitbread.com/tutorials/i2c-fm-radio-part-15-microcontroller-basics-pic10f200>.
33. SOLOMON SYSTECH. *SSD1306 Datasheet* [online]. 2008. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1179026/ETC2/SSD1306.html>.
34. *VS1838B Datasheet* [online]. n.d. [cit. 2024-04-13]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132465/ETC2/VS1838B.html>.
35. DENIS, GORIACHEV. *I2C AVR* [online]. 2018. [cit. 2024-04-13]. Dostupné z: https://github.com/denisgoriachev/i2c_avr.
36. AVINASH, GUPTA. *IR remote interfacing library for AVR series of microcontrollers* [online]. 2008. [cit. 2024-04-13]. Dostupné z: <https://extremeelectronics.co.in/code-libraries/using-ir-remote-with-avr-mcus-part-ii/>.

Obsah příloh

<code>src</code>	
├─ <code>impl</code>	zdrojové kódy implementace
├─ <code>thesis</code>	zdrojová forma práce ve formátu <code>L^AT_EX</code>
└─ <code>text</code>	text práce
├─ <code>thesis.pdf</code>	text práce ve formátu PDF