

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

DIGITÁLNÍ OVLADAČ HLASITOSTI S GRAFICKOU INDIKACÍ

DIGITAL VOLUME CONTROL POD WITH GRAPHIC DISPLAY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Karel Schilhab

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Hanák, Ph.D.

BRNO 2021



Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Karel Schilhab

ID: 203628

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Digitální ovladač hlasitosti s grafickou indikací

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a vyrobte ovladač hlasitosti s digitálním řízením a grafickým zobrazením aktuálního stavu. Hlasitost se bude nastavovat inkrementálním enkodérem a ovladač si ji musí pamatovat i při odpojení napájení. Kromě toho musí ovladač řídit i režim spánku výkonového zesilovače (standby) a dočasné ztlumení (mute). Aktuální stav a hlasitost musí být zobrazena na malém displeji (e-paper nebo OLED). Informace o hlasitosti bude do zesilovače předávána nízkošumovým analogovým signálem v rozsahu 0 až 1 V. Pro realizaci vyberte vhodnou krabičku a mikrokontrolér, nejlépe od firmy Atmel. Při návrhu zohledněte spotřebu energie a snažte se, aby vyrobený vzorek ovladače vypadal esteticky.

DOPORUČENÁ LITERATURA:

[1] BRTNÍK, Bohumil a David MATOUŠEK. Mikroprocesorová technika: práce s mikrokontroléry řady ATMEL AVR ATXmega A4. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-406-4.

[2] MATOUŠEK, David. Práce s mikrokontroléry ATMEL. 2. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-7300-209-4.

Termín zadání: 1.2.2021

Termín odevzdání: 31.5.2021

Vedoucí práce: Ing. Pavel Hanák, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

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

Bakalářská práce řeší návrh a konstrukci digitálního ovladače hlasitosti s grafickou indikací, funkcemi Mute a Standby. Součástí návrhu je mikrokontrolér, OLED displej, D/A převodník, inkrementální enkodér a stabilizátor napětí. Komunikace v zařízení probíhá pomocí sběrnice I2C. Zastavení hlasitosti je realizováno změnou napětí v rozsahu od 0 do 1 V. Cílem práce je navrhnout a vyrobit obvod, napsat zdrojový kód a zařízení zabudovat do krabičky.

Klíčová slova

Mikrokontrolér, inkrementální enkodér, D/A převodník, digitální ovladač hlasitosti.

Abstract

The bachelor thesis deals with the design and construction of a digital volume control with graphical indication, Mute and Standby functions. The design includes a microcontroller, OLED display, D / A converter, incremental encoder and voltage stabilizer. Communication in the device takes place via the I2C line. Stopping the volume is realized by changing the voltage in the range from 0 to 1 V. The aim of the work is to design and manufacture the circuit, write the source code and build the device into the box.

Keywords

Microcontroller, incremental encoder, D / A converter, digital volume control.

Bibliografická citace

SCHILHAB, Karel. *Digitální ovladač hlasitosti s grafickou indikací* [online]. Brno, 2021 [cit. 2021-05-17]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/130559>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Pavel Hanák.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Karel Schilhab*

VUT ID studenta: *203628*

Typ práce: *Bakalářská práce*

Akademický rok: *2020/21*

Téma závěrečné práce: *Digitální ovladač hlasitosti s grafickou indikací*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: 22. května 2021

podpis autora

Poděkování

Děkuji vedoucímu mé bakalářské práce panu Ing. Pavlu Hanákovi, Ph.D. za odbornou pomoc a trpělivost při zpracování mé bakalářské práce.

V Brně dne: 22. května 2021

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	9
ÚVOD	10
1. VÝBĚR KOMPONENTŮ	11
1.1 MIKROKONTROLÉR	11
1.2 DISPLEJ	12
1.3 D/A PŘEVODNÍK	13
1.4 INKREMENTÁLNÍ ENKODÉR	13
1.5 STABILIZÁTOR NAPĚTÍ.....	14
1.6 NAPĚŤOVÁ REFERENCE	15
2. I2C KOMUNIKACE.....	16
2.1 I2C KOMUNIKACE S DISPLEJEM	16
2.2 I2C KOMUNIKACE S D/A PŘEVODNÍKEM	17
3. PERIFERIE MIKROKONTROLÉRU ATMEGA 328P.....	18
3.1 VSTUPNĚ/VÝSTUPNÍ PORTY	18
3.2 EXTERNÍ PŘERUŠENÍ.....	18
3.3 ČASOVAČE	19
3.4 TWI	20
4. NÁVRH SCHÉMATU DIGITÁLNÍHO OVLADAČE HLASITOSTI.....	21
4.1 SEZNAM SOUČÁSTEK	24
5. NÁVRH DESKY PLOŠNÉHO SPOJE.....	25
6. PROGRAM PRO MIKROKONTROLÉR	27
6.1 POUŽITÉ KNIHOVNY	27
6.1.1 <i>avr/io.h</i>	27
6.1.2 <i>util/delay.h</i>	27
6.1.3 <i>i2cmaster.h</i>	27
6.1.4 <i>avr/interrupt.h</i>	27
6.1.5 <i>avr/eeprom.h</i>	27
6.2 FUNKCE MAIN	27
6.2.1 <i>Definice vstupů a výstupů</i>	27
6.2.2 <i>Nastavení periférií</i>	28
6.2.3 <i>Inicializace displeje</i>	28
6.2.4 <i>Hlavní smyčka programu</i>	28
6.3 NASTAVENÍ HODNOTY VOLUME	31
6.4 FUNKCE PRO OVLÁDÁNÍ DISPLEJE	33
6.5 FUNKCE PRO OVLÁDÁNÍ D/A PŘEVODNÍKU	33
6.6 ZOBRAZENÍ HODNOT NA DISPLEJI.....	34
7. KOMPLETACE DIGITÁLNÍHO OVLADAČE HLASITOSTI.....	37
8. ZÁVĚR.....	40

LITERATURA.....	41
SEZNAM SYMBOLŮ A ZKRATEK	42
SEZNAM PŘÍLOH.....	43

SEZNAM OBRÁZKŮ

4.1	Schéma zapojení displeje.	22
4.2	Schéma zapojení enkodéru a spínače.	23
5.1	Rozmístění součástek na desce plošných spojů strana Top.	25
5.2	Rozmístění součástek na desce plošných spojů strana Bottom.	26
5.3	Deska plošných spojů strana Top.	26
5.4	Deska plošných spojů strana Bottom.	26
6.1	Vývojový diagram přepínání módů.	31
6.2	Zobrazení nastavené hodnoty volume.	36
6.3	Zobrazení symbolu Mute.	36
6.4	Zobrazení symbolu Standby.	36
7.1	Deska plošných spojů strana Top.	37
7.2	Deska plošných spojů strana Bottom.	37
7.3	Osazená deska plošných spojů strana Top.	37
7.4	Osazená deska plošných spojů strana Bottom.	38
7.5	Krabička Hammond Manufacturing 1551KBK s připravenými otvory.	38
7.6	Digitální ovladač hlasitosti umístěný v krabičce.	39
7.7	Hotový digitální ovladač hlasitosti.	39

SEZNAM TABULEK

4.1	Seznam součástek.	24
-----	------------------------	----

ÚVOD

Zadáním bakalářské práce bylo vytvořit digitální ovladač hlasitosti řízený mikrokontrolérem. Samotné nastavování mělo být prováděno inkrementálním enkodérem a požadovaný výstup z ovladače mělo být stejnosměrné napětí o hodnotě od nula do jednoho voltu a signál pro zapnutí zesilovače Standby. Ovladač měl také obsahovat grafickou indikaci nastavení zesílení, funkce Mute a funkce Standby. Ovladač má být umístěn na kraji počítačového monitoru, tak aby displej byl umístěn na pohledové straně krabičky a ovládací prvky byly z boku krabičky. Na celkové provedení krabičky jsou kladeny vysoké vizuální nároky.

K nastavování hlasitosti u většiny audiozařízení se standardně používají otočné nebo tahové potenciometry s logaritmickou charakteristikou. Použití těchto potenciometrů je jednoduché a jedná se o poměrně levnou součástku. Problém nastává, je-li zapotřebí ovládat více zařízení najednou, například více kanálů jednoho zesilovače, nebo je-li zapotřebí ovládat jedno zařízení více ovladači, a to například u ovládání na panelu přístroje a zároveň dálkovým ovladačem. Pro ovládání jednoho přístroje více ovladači lze využít takzvaný motorizovaný fader, který změní svou polohu v závislosti na pokynu jiného ovládacího prvku. Tyto motorizované fadery jsou hojně využívány u digitálních mixážních pultů. Pro ovládání více zařízení najednou se dá použít takzvaný tandemový potenciometr, ale ten je pro šest kanálů příliš veliký a jeho připojení by vyžadovalo mnoho stíněných vodičů. Krabička na takový ovladač by byla příliš velká a nebylo by možné její umístění na počítačový monitor. Z tohoto důvodu je lepším řešením využití digitálního nastavování hlasitosti, které je v tomto případě menší a může ovládat neomezený počet kanálů. [1]

Tento konkrétní digitální ovladač hlasitosti má udávat nastavovanou hodnotu zesílení pomocí napětí. Součástí ovládaného šesti kanálového zesilovače je obvod, který nastavuje zesílení zesilovače podle vstupního napětí. Při lineární změně napětí na tomto obvodu se hodnota mění logaritmicky. Nastavování napětí dodávaného digitálním ovladačem se u tohoto konkrétního zesilovače musí pohybovat v rozmezí od nuly do jednoho voltu. Při nastavení na nula voltů bude zesílení minimální a při nastavení na jeden volt bude zesílení nastaveno na maximální.

Cílem bakalářské práce je sestavit funkční prototyp digitálního ovladače hlasitosti s grafickou indikací, který bude bez dalších úprav použitelný pro ovládání výkonového zesilovače.

1. VÝBĚR KOMPONENTŮ

Na začátku návrhu je zapotřebí vybrat součástky, které budou vyhovovat pro konstrukci ovladače hlasitosti s grafickou indikací.

1.1 Mikrokontrolér

Hlavním komponentem digitálního ovladače je mikrokontrolér. Na trhu je mnoho výrobců a typů mikrokontrolérů. Asi nejrozšířenějším výrobcem na trhu je firma Mikrochip, další jsou například Texas Instruments nebo Infineon Technologies. Mikrokontrolér je integrovaný obvod, který má funkci mikropočítače. Jedná se o hlavní komponent velkého množství elektronických zařízení. Mikrokontrolér je konstruován tak, aby byl schopný pracovat bez velkého množství podpůrných komponentů. V mikrokontroléru bývají umístěny paměti, logické a analogové vstupy a výstupy, komunikační sběrnice (nejčastěji SPI a I2C), a velké množství dalších podpůrných obvodů.

Z historického hlediska se mikrokontroléry dělí na dvě architektury, a to Von Neumannovu architekturu a Harvardskou architekturu.

Von Neumannova architektura je založena na používání jedné paměti pro program i ukládání dat. U této architektury je tedy vše ukládáno na jedno místo a není zapotřebí rozlišovat instrukcemi přístup k programu a k datům. Další výhodou je, že k přenášení dat stačí pouze jedna datová sběrnice, ovšem tyto mikrokontroléry jsou pomalejší právě z důvodu přenášení dat po jedné sběrnici.

Harvardská architektura je oproti tomu tvořena jednotlivými paměťmi rozdělenými pro program a pro ukládaná data. Mikrokontroléry této architektury obsahují více sběrnic. Výhodou těchto mikrokontrolérů je jejich rychlost, protože je možné číst data a instrukce zároveň v jeden čas.

U moderních architektur již nelze s přesností určit, jestli se jedná o Von Neumannovu architekturu nebo Harvardskou architekturu, protože z důvodu stupně integrace se paměť mikrokontroléru jeví jako jedna a ve skutečnosti je složena z několika nezávislých pamětí a sběrnic.

Pro aplikaci v digitálním ovladači hlasitosti není zapotřebí složitý procesor. Pro tuto konstrukci byl vybrán mikrokontrolér AVR s označením ATMEGA328P-AU od společnosti Mikrochip. Je využíván v mnoha aplikacích jako velice levný a jednoduchý mikrokontrolér. Velice známým použitím je použití na platformě Arduino. Tento mikrokontrolér má osmibitové jádro Harvardské architektury RISC. Architektura RISC je architektura založena na snížení instrukcí, jedná se o instrukce, které jsou modifikované pro rychlý tok informací. ATMEGA328P-AU disponuje třemi paměťmi, a to EEPROM (1 kB), SRAM (2 kB) a Flash (32 kB). Paměť Flash je určena pouze k ukládání napsaného programu a po naprogramování mikrokontroléru se již nemění. Data, se

kterými mikrokontrolér pracuje se ukládají na paměť SRAM, tato paměť pracuje pouze je-li mikrokontrolér napájen a v případě výpadku napájení jsou data smazána. Pro případ uložení zpracovaných dat slouží paměť EPROM, která uchovává zpracovaná data uložená i po odpojení napájení, a při opětovném zapnutí jsou data opět přístupná. Tento mikrokontrolér také disponuje možností vnějšího přerušení programu na 24 pinech. Pro komunikaci je možné využít rozhraní debugWIRE, I2C, SPI x2 a UART. Tento mikrokontrolér také disponuje vlastním rezonátorem, z tohoto důvodu již není nutné pro jeho funkci zapojovat externí krystal a s ním spojené kondenzátory proti zákmitu.[2], [3]

1.2 Displej

Další důležitou součástí je displej. V zadání práce je navrhnut e-paper nebo OLED displej.

E-paper je založen na principu pohybujících se bílých mikrokapslí, které jsou umístěny v tenké vrstvě tmavého viskózního roztoku. Při přivedení impulzu se tyto částice pohnou směrem k přední stěně displeje a pixel se jeví jako bílý, při přivedení impulzu opačné polarity částice změni svou polohu a pixel je černý. Uvedený viskózní roztok má tak vysokou hustotu, že bez přivedení signálu displej nemění svou podobu a poslední zobrazení na něm zůstane i několik let. Tento displej nepotřebuje žádné podsvícení, ale jeho podoba nevypadá příliš atraktivně.

OLED displej pracuje na principu LED diod, jedna vrstva displeje tvoří anodu a druhá vrstva tvoří katodu. Jedna z těchto vrstev je kovová a druhá je tvořena průhledným materiálem (pohledová strana displeje). Mezi elektrodami se nachází vrstva organické látky. OLED displeje existují s aktivní maticí (označované taky jako AMOLED), nebo pasivní maticí (označované jako PMOLED).

Displeje pracující na principu pasivní matrice jsou jednodušší a používají se převážně na aplikace, kde je zapotřebí zobrazit pouze text, nebo kde je zapotřebí jednoduchý grafický displej. Jejich funkce je založena na principu matrice složené z mřížky navzájem překřížených vodičů, kde je v každém překřížení připojená elektroda. Každé toto překřížení představuje jeden pixel. Na anody a katody se napětí přivádí pomocí multiplexových přepínačů. Displej produkuje světelná záření a nepotřebuje tudíž žádné podsvícení, a je viditelný i bez vnějšího zdroje světla. Jas displeje je možné ovlivnit velikostí proudu impulzu přivedeného na daný pixel. U většiny displejů je ovšem možné nastavovat jas pouze celého displeje. OLED displeje s pasivní maticí jsou vyráběny v alfanumerické verzi a v grafické verzi. Alfa numerická verze má již předdefinované pole pro psaní znaků, zatímco u grafické verze jde zapínat jednotlivé pixely nezávisle na jejich poloze.

Displeje používající aktivní maticí jsou vhodné pro náročnější aplikace, jako je například zobrazování videí nebo jiné grafické aplikace. U těchto displejů je každý pixel samostatně řízen dvěma tranzistory. Displeje s aktivní maticí mají ostřejší vykreslování

obrazu, vyšší vykreslovací frekvenci a také mají nižší spotřebu než displeje s pasivní matricí. Tyto displeje jsou ovšem dražší z důvodu složitější struktury.

Pro digitální ovladač hlasitosti byl zvolen grafický OLED displej s pasivní matricí od značky Electronic assembly s označením EAW096016-WALW. Tento displej má zobrazovací pole s rozlišením 96 x 16 pixelů a rozměry zobrazovací plochy 21,1 mm x 3,5 mm. Na displeji není zapotřebí zobrazovat žádné složité grafické obrazce, a proto bude tento displej na ovladač hlasitosti dostačující. Tento displej komunikuje s mikrokontrolérem přes sběrnici I2C.[4], [5], [6]

1.3 D/A převodník

D/A převodník je součástka zajišťující převod digitálního signálu na analogový. D/A převodníky se dnes používají téměř ve všech zařízeních pracujících se zvukem. Jejich komunikace může být zajištěna různými typy rozhraní, nejčastější jsou ovšem rozhraní SPI a rozhraní I2C. D/A převodníky se vyrábějí v různých variantách rozlišení. Pro převod zvukových vzorků se používají převodníky s rozlišením 16 bitů až 32 bitů. Pro ovládání hlasitosti stačí převodník s menším rozlišením. D/A převodníky mají také různé maximální vzorkovací frekvence, to ovšem pro jednoduchou aplikaci v digitálním ovladači není zapotřebí řešit. D/A převodníky jsou implementovány jako integrované obvody a jsou často konstruovány jako několik převodníků v jednom pouzdře, zde mluvíme o vícekanálových D/A převodnících.

Pro digitální ovladač hlasitosti byl použit jednocanálový dvanáctibitový D/A převodník, vyrobený firmou Microchip s označením MCP4726A0T-E/CH. Tento D/A převodník byl vybrán z důvodu stejné komunikace jako používá pro komunikaci displej, a to prostřednictvím rozhraní I2C. Pro aplikaci by byl dostačující i převodník s menším bitovým rozlišením například čtyř bitů, ale ten by mohl nastavovat pouze 16 úrovní hlasitosti, vzhledem k nepatrnému rozdílu ceny byl zvolen převodník s větším rozlišením, u kterého bude možné mnohem přesnější nastavení.[7], [8]

1.4 Inkrementální enkodér

Enkodéry jsou elektronické součástky nahrazující standardně používané potenciometry. Enkodéry existují ve dvou typech, a to jako absolutní enkodéry a inkrementální enkodéry.

Absolutní enkodéry jsou jakousi digitální variantou klasických potenciometrů. Mají pevně daný počet otáček a ke každé poloze je jim přiřazen jednoznačný kód, podle kterého mikrokontrolér pozná polohu, ve které se právě enkodér nachází. Na počtu poloh enkodéru závisí bitová velikost přenášených dat.

Inkrementální enkodéry se dále dají rozdělit na mechanické enkodéry a optické enkodéry. Optické enkodéry se používají například pro snímání otáček motoru, u nejrůznějších pojezdů a servomotorů. Mechanické enkodéry jsou oproti nim levnější, ale mívají menší rozlišovací schopnost a nejsou tak spolehlivé. Mechanické enkodéry se

využívají hlavně jako panelové ovladače. Mechanické enkodéry disponují dvojicí spínačů, které při spínání generují dva obdélníkové signály a ty mají posunutou fázi o devadesát stupňů. Podle tohoto fázového posuvu lze určit, zdali je hmatníkem otáčeno po směru nebo proti směru hodinových ručiček. Pro indikaci míry otočení bývají enkodéry vybavené zarážkami, přes které enkodér při otáčení přecvakává. Mechanické inkrementální enkodéry se vyrábějí s různým počtem sepnutí na otočku a s různým počtem zarážek.

Enkodéry jsou vyráběny ve více verzích, horizontální nebo vertikální, pro montáž technologií THT nebo SMD a v mnoha případech doplněné o spínač pod hřídelí pro ovládání další funkce zařízení, nebo s podsvícenou hřídelí.

Pro digitální ovladač byl vybrán enkodér od výrobce Alps s označením EC11B15242AE, jedná se o horizontální enkodér určený pro montáž THT a závitem pro uchycení do panelu. Enkodér s touto montáží byl vybrán z důvodu uchycení celé desky spojů tímto enkodérem do krabičky. [9]

1.5 Stabilizátor napětí

Stabilizátor napětí je součástka využívaná pro snížení nebo stabilizaci napětí. Výstupní napětí stabilizátoru není ovlivněno odebíraným proudem, vstupním napětím ani okolní teplotou. Stabilizátory napětí se dělí na několik typů, a to lineární parametrický stabilizátor, lineární zpětnovazební stabilizátor a spínaný zpětnovazební stabilizátor.

Lineární parametrický stabilizátor je nejčastěji tvořen Zenerovou nebo lavinovou diodou. Jedná se o nejjednodušší stabilizátor, protože k jeho realizaci stačí pouze Zenerova dioda a rezistor. Lineární parametrický stabilizátor lze zdokonalit přidáním tranzistoru zapojeného jako emitorový sledovač, který reguluje napětí podle Zenerovy diody.

Lineární zpětnovazební stabilizátor pracuje na principu porovnávání výstupního napětí stabilizátoru s referenčním napětím umístěným v pouzdře stabilizátoru. Regulačním prvkem tohoto stabilizátoru je tranzistor v zapojení emitorového sledovače a je řízený operačním zesilovačem, který porovnává obě napětí. Tento stabilizátor jde realizovat i za pomoci pouze diskrétních součástek, a to dvěma tranzistory, Zenerovou diodou a rezistory zapojenými jako napěťový dělič.

Spínaný zpětnovazební stabilizátor funguje podobně jako lineární zpětnovazební stabilizátor s tím rozdílem, že tranzistor je pouze v poloze sepnuto nebo rozepnuto a změnou střídý (poměrem sepnuto-rozepnuto) určuje velikost výstupního napětí.

Každý typ stabilizátoru má při práci úbytek napětí, tento úbytek závisí na úbytku na jednotlivých součástkách ve stabilizátoru. Jednotlivé typy stabilizátorů napětí se liší úrovní úbytku napětí. [10]

1.6 Napět'ová reference

Při provozu zařízení se jeho napájecí napětí může v čase měnit. Když je zapotřebí mít na výstupu přesnou hodnotu, je nutné, aby v obvodu byla reference, se kterou lze danou hodnotu porovnat.

Napět'ová reference je součástka (nebo celý obvod), na které je pevně daná hodnota napětí, která se nesmí měnit v závislosti na zátěži nebo teplotě. Její využití je hlavně v napájecích zdrojích, A/D a D/A převodnících.

2. I2C KOMUNIKACE

Mikrokontrolér v této aplikaci využívá ke komunikaci s displejem a s D/A převodníkem sériové sběrnice I2C. Jedná se o sběrnici používanou pro komunikaci mezi jednotlivými komponenty jednoho zařízení. S touto komunikací přišla společnost Philips, původně byla využívána pouze touto firmou. Nyní se sběrnice I2C využívá u velkého počtu zařízení pro komunikaci mezi mikrokontroléry a dalšími součástkami, jako jsou například A/D a D/A převodníky, sériové paměti, obvody pro video a audio. Výhodou této komunikace je, že komunikuje pouze pomocí dvou vodičů označených jako SDA a SCL. Jedno zařízení, většinou mikrokontrolér, je nastaveno jako MASTER a ostatní zařízení v zapojení jsou nastavené jako SLAVE. MASTER se od ostatních komponentů liší tím, že na vodiči SCL vysílá hodinový signál řídící ostatní součásti. Princip funkce komunikace I2C spočívá v adresování dat pro jednotlivé komponenty. Jedno zařízení vysílá data a všechna ostatní zařízení tyto data přijímají, podle nastavené adresy stanoví, jestli jsou data určená pro ně. Standardně jsou adresy dány sedmibitovou informací, ale pro složitější aplikace jde nastavovat adresy i deseti bity. To znamená, že v řetězci může být zapojeno 128 zařízení pro sedmibitovou komunikaci nebo 1024 zařízení při desetibitové komunikaci. Ve skutečnosti jde ovšem připojit zařízení méně, protože některá zařízení již mají předdefinované adresy, které se volí menším počtem bitů. Přenos dat je složený z více celků. Nejprve jsou oba signály (SDA a SCL) nastaveny na logickou jedničku. Pro zahájení přenosu se SDA přepne do nuly, tím je spuštěn přenos a SCL začíná pracovat. Odesílání dat probíhá po jednom bitu, při každém nastavení SCL do nuly je odeslán jeden bit. Zařízení, pro které jsou data odeslána, po přijetí jednoho bajtu posílá potvrzovací zprávu Ack. Přenos probíhá, dokud není ukončen nastavením SDA do logické jedničky při SCL v logické jedničce.

2.1 I2C komunikace s displejem

Displej pro komunikaci využívá vestavěný řadič SSD1306B, který jde maximálně použít pro maticový panel o rozměru 128 x 64 pixelů. Tento řadič má hardwarově volitelné komunikační rozhraní I2C, SPI, paralelní komunikaci 8080 a paralelní komunikaci 6800. Při komunikaci pomocí rozhraní I2C komunikuje řadič pomocí pomocného bitu adresy SA0, datového signálu SDA a hodinového signálu SCL. Datové a hodinové cesty musí být připojené k pull-up rezistorům. Pro inicializaci se používá pin RES. Adresa slave řadiče SSD1306B je „b0111100“ nebo „b0111101“ v závislosti na tom, jestli je pin SA0 nastaveno na logickou nulu nebo logickou jedničku. Režim zápisu je spuštěn přepnutím bitu R/W na logickou nulu. Potvrzovací signál je odeslán po přijetí jednoho bajtu, a to včetně adresy slave a R/W bitu. Potvrzovací bit je definován stažením sběrnice SDA na logickou nulu během vysoké úrovně hodinového impulsu souvisejícího s potvrzením. Po přenosu adresy slave může být poslán kontrolní bajt nebo datový bajt. Kontrolní bajt je

složen z bitů Co a D/C#, po nichž následuje šest nul. Pokud je bit Co nastaven na logickou nulu, bude přenos následujících informací obsahovat pouze datové bajty. Bit D/C# určuje, jestli další datový bajt bude příkaz nebo data. Pokud je bit D/C# nastaven na logickou nulu, bude následující datový bajt příkaz, ale pokud je bit D/C# nastaven na logickou jedničku, bude následující datový bajt obsahovat data určená k zapsání na GDDRAM. Ukazatel adresy sloupce GDDRAM se automaticky pokaždé zvýší o jednu. Režim zápisu se ukončí po použití podmínky pro zastavení.[6]

2.2 I2C komunikace s D/A převodníkem

D/A převodník komunikuje s mikropočítačem následujícím způsobem:

Mikropočítač posílá START sekvenci, čímž zahajuje komunikaci. Dále posílá první bajt, který obsahuje sedmibitovou adresu převodníku a R/W bit. Adresa použitého převodníku je složena ze čtyř pevně určených bitů „1100“, za nimi následují další tři bity, určené při výrobě převodníku. Je tedy možné použít hned osm převodníků s různými adresami na jedné sběrnici I2C. Informace o posledních třech bitech adresy je ukryta v názvu převodníku, konkrétně vybraný převodník MCP4726A0T-E/CH má adresu „1100000“. R/W bit pak určuje, zda budeme číst, nebo zapisovat data do registru převodníku.

Za bajtem obsahujícím adresu a R/W bit následují další dva bajty. První bajt obsahuje dva příkazové bity, určující, co se má s dalšími daty provést. V případě zápisu do registru DA převodníku jsou oba bity v logické nule. Dále následují dva „power down“ bity, jejich nastavením lze přerušit činnost převodníku a šetřit tak spotřebu. Výstup převodníku se pak připojí na interní „pull down“ rezistor. Pro nepřetržitý převod jsou oba bity nastaveny rovněž do logické nuly. Poslední čtyři bity druhého bajtu a třetí bajt pak náleží dvanáctibitové hodnotě pro převod napětí. Na konci mikropočítač vyšle STOP sekvenci, kterou ukončí.[8]

3. PERIFERIE MIKROKONTROLÉRU ATMEGA 328P

3.1 Vstupně/výstupní porty

Ke komunikaci mikrokontroléru s okolními obvody slouží vstupně výstupní porty. Atmega 328P disponuje třemi osmibitovými porty PORTB, PORTC a PORTD. Každý z těchto portů je možné ovládat třemi registry.

Registr DDRx slouží k nastavení směru portů. Jednotlivými bity portů je možné nastavit piny jako vstupní, a to zápisem logické 1, nebo jako výstupní zápisem logické 0 do registru DDRx. Například pokud nastavíme registr na $DDRB = 0b00001111$, tak budou první čtyři piny vstupní a další čtyři piny výstupní.

Pokud jsou piny mikrokontroléru nastaveny jako vstupní, lze jejich stav číst pomocí registru PINx. Je-li na pin připojeno napájecí napětí, v registru se objeví hodnota logické 1, při připojení k zemi pak hodnota logická 0.

V případě že je pin kontroléru nastaven registrem DDRx jako výstupní, k jeho řízení se využívá registr PORTx. K nastavení hodnoty napájecího napětí na výstup zapisujeme do registru hodnotu logické 1 a pro nastavení pinu na 0 V nastavujeme logickou 0. Pokud je pin nastaven registrem DDRx jako vstupní, je možné pomocí registru PORTx nastavit na vstupu interní pull-up rezistor o hodnotě asi 20 – 50 k Ω , a to zápisem logické 1 do tohoto registru.

Vstupně výstupní porty jsou také využívány ostatními perifériemi mikrokontroléru, jako jsou například A/D převodníky, výstupy čítačů, nebo komunikační periférie (TWI, SPI, USART). Při aktivaci těchto periférií dojde k převzetí řízení příslušných pinů.

3.2 Externí přerušení

Mikrokontrolér umožňuje přerušení běhu programu při změně stavu určitého definovaného pinu a vykonat krátký kód tzv. obsluhu přerušení. Atmega 328P má dva vstupy pro externí přerušení, a to piny INT0 a INT1. Pro tyto vstupy je možné nastavit generování přerušovacího vektoru pro nízkou úroveň, vzestupnou hranu, sestupnou hranu, nebo jakoukoli změnu úrovně.

Dále mikrokontrolér obsahuje vstupní piny PCINT0 – 23. Jedná se o piny, které je možné maskovat jako tři samostatné skupiny PCINT0 – 7, PCINT8 – 14, PCINT16 – 23 a povolené vstupy pak vyvolávají přerušení při jakékoliv změně úrovně. Není tedy možné filtrovat pouze sestupné nebo vzestupné hrany.

Pro povolení těchto přerušovacích vstupů slouží registr PCICR, kde prvním až třetím bitem povolujeme nebo zakazujeme jednotlivé skupiny pinů PCINT. Chceme-li, aby přerušení vyvolával pouze jeden nebo více určitých vstupů, je nutné je povolit v maskovacím registru. Pro každou skupinu pinů existuje jeden registr PCMSK0, PCMSK1, PCMSK2.

Pokud tedy chceme aktivovat vstup PCINT22 pro externí přerušení, je nutné nastavit do maskovacího registru PCMSK2 = 0b01000000 a přerušení povolit zápisem do registru PCICR = 0b00000100. Pro povolení přerušování od jakéhokoliv zdroje je ještě nutné nastavit bit I ve status registru mikrokontroléru. Což je v Atmel Studiu možné provést voláním funkce sei().

3.3 Časovače

V případě Atmega 328P se jedná o modul čítače se dvěma nezávislými porovnávacími jednotkami s možností vytvářet PWM signál na dvou výstupních pinech. Časovače je možné využít pro odměřování času mimo jádro mikrokontroléru. Je tedy možné stále provádět kód a měřit čas nezávisle, což dává značnou výhodu proti použití funkce delay, kdy je pro čekání využíváno jádro kontroléru a není v tu chvíli možné provádět další kód. Další možností je využití komparátorů s porovnávacími registry ke generování periodického signálu na výstupy.

Atmega 328P obsahuje dva 8bitové a jeden 16bitový časovač. Každý z těchto časovačů je možné napájet různými kmitočty hodinového signálu a využívat je k nezávislému měření času, nebo k vytváření až šesti různých periodických průběhů na výstupech kontroléru.

8bitový časovač 0, je možné nastavit stejně jako i ostatní dva časovače do šesti pracovních režimů. Jedná se o normální mód, kdy čítač čítá nahoru až do přetečení, dále rychlý a fázově korigovaný PWM režim oba se dvěma možnými rozsahy čítání, a nakonec režim CTC, kdy čítač reaguje na shodu s nastavenou hodnotou v registru OCR0 a vynuluje čítací registr. Módy časovače je možné nastavit bity WGM00, WGM01 a WGM02, které se nacházejí v registrech TCCR0A a TCCR0B. V registru TCCR0A je dále možné přiřazovat výstupy kontroléru k čítačům.

Před samotný čítač je možné zařadit 10bitovou předděličku hodinového signálu. Jjí nastavení je možné bity CS00, CS01 a CS02 v registru TCCR0B. Hodinový signál z mikrokontroléru je možné dělit 8, 64, 256 nebo 1024, dále je možné připojit externí hodinový signál na pin T0 a čítat na jeho náběžnou, nebo sestupnou hranu.

Registr TCNT0 slouží jako čítací registr, je však také možné do něj zapisovat libovolné hodnoty programově. Dále jsou zde dva registry OCR0A a OCR0B, do kterých je zapisována hodnota sloužící k porovnávání s registrem TCNT0 a generování výstupního signálu.

Registrem TIMSK0 je možné povolit přerušení, a to při přetečení registru TCNT0, a při shodě s registry OCR0A a OCR0B. Přerušení je výhodné použít právě při odměřování časového intervalu, kdy se nejprve nastaví počáteční hodnota do čítacího registru a po načítání registru dojde po přesně stanoveném časovém intervalu k jeho přetečení a vyvolání přerušení.

3.4 TWI

Kvůli licenčním právům na název I2C nazývá Atmel tento protokol TWI (Two wire interface). Komunikaci I2C lze na tomto mikrokontroléru realizovat dvěma způsoby.

Programovým řešením tzv “bit-banging” I2C, kdy softwarem mikrokontroléru řídíme časování a odesílání jednotlivých bitů podle tohoto protokolu, tento způsob komunikace nedosahuje příliš velkých datových rychlostí, ale výhodou může být možnost zvolit si pro SCK a SDA libovolné dva piny mikrokontroléru. Kromě toho je také možné využít více pinů a komunikovat po více sběrnících I2C.

Hardwarovým řešením, kde používáme periférii mikroprocesoru, sloužící přímo k účelu komunikace pomocí protokolu I2C. Mikropočítač ATmega328P má jedno hardwarové rozhraní TWI a dvojice pinů SCK a SDA je pevně připojena na piny PC4 a PC5. Toto rozhraní je možné řídit pomocí pěti registrů, a lze díky němu dosahovat velkých datových rychlostí, při malé zátěži jádra kontroléru.

Pro nastavení rychlosti komunikace slouží registr TWBR, datová rychlost je pak dána vztahem:

$$f = (\text{frekvence_CPU}) / (16 + 2 \cdot TWBR \cdot 4^{TWPS})$$

Kde TWPS je předdělička hodin pro TWI, tu je možné konfigurovat v registru TWSR. V registru TWSR dále najdeme informační bity, které říkají, ve kterém stavu se celá periferie TWI nachází. Registr TWAR slouží k určení adresy Slave, ten ale není v tomto případě využit, protože využíváme mikropočítač v roli Master.

Samotné řízení komunikace se pak provádí pomocí registru TWCR. Pomocí jeho bitů lze například zapínat celý modul TWI, vysílat START sekvenci nebo STOP sekvenci atd. Makra jednotlivých bitů jsou uložena v knihovně `avr/io.h`, díky které je práce s modulem TWI usnadněna. Poslední registr TWDR slouží k zápisu samotných dat, které mají být po sběrnici vyslány, nebo pro příjem dat, v závislosti na stavu periferie.[3]

4. NÁVRH SCHÉMATU DIGITÁLNÍHO OVLADAČE HLASITOSTI

Schéma celého zařízení je přiložené v příloze.

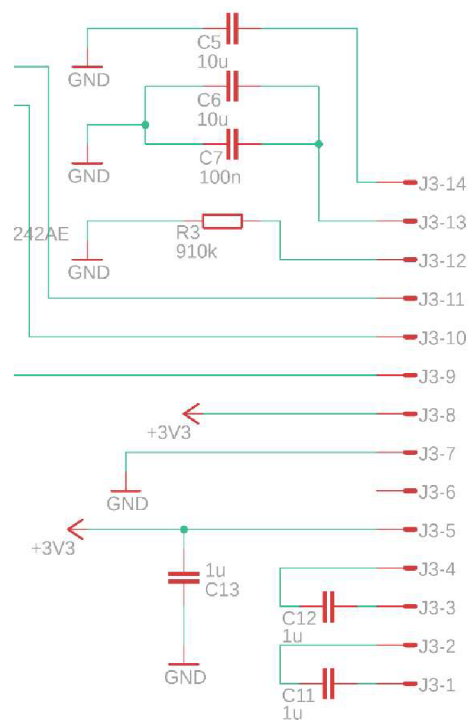
Návrh schématu a také desky plošného spoje byl proveden v programu Eagle 9.6.2.

Ovladač bude k zesilovači připojen pomocí kabelu se čtyřmi vodiči, tyto vodiče budou sloužit k přívodu napájecího napětí o hodnotě 5 V, k vedení ovládacího napětí do zesilovače, které bude nastavováno v rozmezí od 0 V do 1 V v závislosti na požadovaném zesílení zesilovače, k vedení ovládacího napětí, které bude zesilovač zapínat a poslední bude sloužit jako zemní vodič.

Na vstupu napájecího napětí se nachází feritová perla, která je určena k odstranění vysokofrekvenčního napětí, vysokofrekvenční napětí se uvnitř feritového jádra přemění na teplo a nepronikne tak do zařízení.

Hlavním komponentem zařízení je mikrokontrolér ATMEGA 328P-AU, tento konkrétní mikrokontrolér je pro montáž SMD a je umístěn v pouzdře TQFP-32, jedná se o poměrně malé pouzdro, ale stále zapojitelné ručním pájením.

Další podstatnou součástí je displej Electronic assembly EAW096016-WALW, k jehož připojení slouží pohyblivý přívod obsahující 14 pinů. Pro správnou funkci displeje je zapotřebí na piny připojit pomocný obvod navržený výrobcem (obr. 4.1), který slouží k nastavení displeje. Jedná se o kondenzátory C5, C6, C7, C11 a C12 a rezistor R3. Kondenzátor C11 zapojený mezi piny 1 a 2, a kondenzátor C12 zapojený mezi piny 3 a 4 jsou součástí nábojové pumpy, vestavěné v displeji, která dodává napájecí napětí 7 V pro OLED matici. Na pin číslo 5 je připojené napájecí napětí 3,3V a stabilizační kondenzátor C13, který je napojen na zem. Jedná se o napájecí napětí pro nábojovou pumpu. Pin 7 a 8 slouží jako hlavní napájecí piny řadiče SSD1306B OLED displeje. Pin 7 je připojen na zem a pin 8 na napájecí napětí 3,3 V. Pin 9 je RESET obvodu, ten je připojen přímo na jeden z výstupů mikrokontroléru. Piny 10 a 11 slouží ke komunikaci po lince I2C pin 10 je připojen na SCL a pin 11 na SDA. Mezi pin 12 a zem je připojen rezistor R3, který slouží k nastavení referenčního proudu pro segmenty OLED matice, čímž je určen jas displeje. Tento proud nesmí být větší než 12,5 μ A. Na pinu 13 je výstupní napětí pro OLED segmenty, které je podle doporučení výrobce stabilizováno pomocí paralelní kombinace kondenzátorů C6 a C7. A na pinu 14 se nachází kondenzátor C5 připojený na zem, sloužící ke stabilizaci výstupního napětí vestavěné nábojové pumpy.



Obr. 4.1 Schéma zapojení displeje.

Ovládací napětí na výstup dodává D/A převodník MCP4726A0T-E/CH. Tento převodník je stejně jako displej připojen přes sběrnici I2C. Kromě napájecího napětí převodník ke své správné funkci potřebuje referenční napětí, které mu je dodáno pomocí napěťové reference LMV431BIMF/NOPB, která má referenční napětí 1,24 V a v obvodu je zapojena sériově k rezistoru R4 připojenému na napětí 3,3 V.

Mikrokontrolér, displej i D/A převodník jsou napájeny napětím 3,3 V, stejné napájecí napětí je důležité pro správnou funkci komunikace po rozhraní I2C. Toto napětí zajišťuje v obvodu napěťový stabilizátor s označením P7333-33SAG-7, jedná se o regulátor s nízkým úbytkem napětí a to 0,3 V. Napájecí vstup mikrokontroléru musí být ošetřen proti zákmitům, to je realizováno pomocí kondenzátoru C10.

Pro správnou funkci komunikace I2C je zapotřebí kromě propojení kontaktů SDA a SCL displeje, převodníku a mikrokontroléru ještě napájení této komunikace, to je zajištěno připojením vedení SDA přes rezistor R1 a vedení SCL přes rezistor R2 na napájecí napětí 3,3 V.

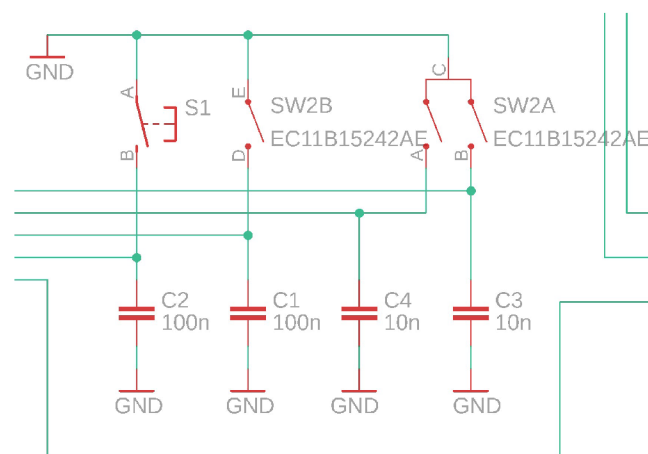
Signál pro zajištění ovládní funkce Standby bude 0 V pro vypnutí napájecího zdroje zesilovače a 5 V pro jeho zapnutí, o tuto funkci se v obvodu stará unipolární mosfet tranzistor Q1 jehož Gate je připojen přes R7 na výstup mikrokontroléru PC3, Sours je připojen na zem a Drain přes rezistor R6 na napětí 3,3 V.

Ovládní bude prováděno pomocí enkodéru EC11B15242AE, spínače v enkodéru a samostatného spínače S1 (obr. 4.2). Spínače i enkodér jsou zapojeny na zem a druhá

strana kontaktů je připojena na vstupy mikrokontroléru s aktivovanými vnitřními pull-up rezistory. Aby nedocházelo k záskmitům jsou na spínačích kondenzátory C2 a C1 a na enkodéru C3 a C4, které tvoří RC článek se zabudovanými pull-up rezistory. Doba nabíjení těchto kondenzátorů je dána rovnicí:

$$\tau = R \cdot C$$

Pull-up rezistory v mikrokontroléru mají hodnotu 20 – 50 k Ω . Původně navržené kondenzátory měly hodnotu 100 nF. Později bylo zjištěno, že při rychlém otáčení hřídeli enkodéru se nestíhají kondenzátory nabíjet na prahové napětí potřebné k detekci mikrokontrolérem, čímž docházelo k chybám v podobě vynechání některých ovládacích impulzů. Tato chyba byla odstraněna zmenšením kondenzátorů C3 a C4 na hodnotu 10 nF, čímž se čas potřebný k nabíjení kondenzátorů zmenšil desetkrát.



Obr. 4.2 Schéma zapojení enkodéru a spínače.

Pro programování zařízení je ve schématu programovací konektor, který je připojený na napájecí napětí, na zem a s mikrokontrolérem je spojený na kontakty MOSI, MISO, SCK, RESET. Tento pin musí být pro normální funkci obvodu ve stavu logické 1. K jeho dosažení slouží externí rezistor R5 připojený na kladné napájecí napětí. Při programování je vstup RESET stažen do logické 0 pomocí programátoru.[3], [6], [8]

4.1 Seznam součástek

Tabulka 4.1 Seznam součástek

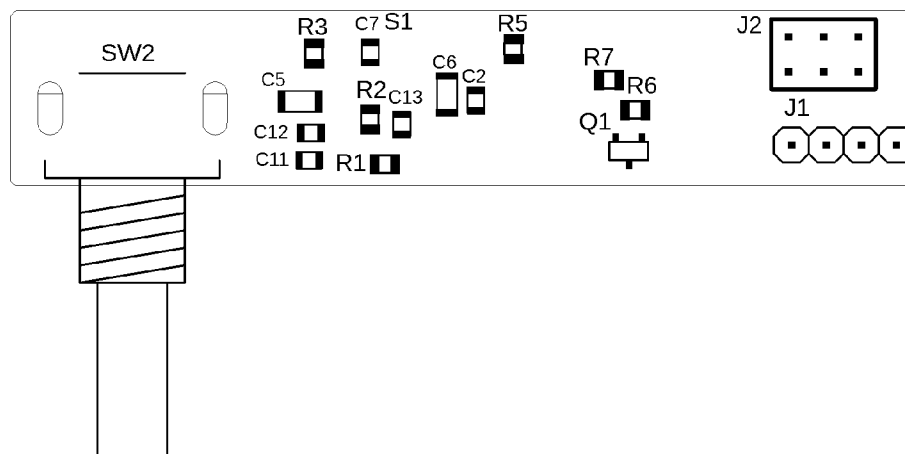
Označení	Součástka	Hodnota	Pouzdro
IC1	Mikrokontrolér	ATMega328P-AU	TQFP32
IC2	Stabilizátor napětí	AP7333-33SAG-7	SOT23
IC3	Napěťová reference	LMV431BIMF/NOPB	SOT23
IC4	D/A převodník	MCP4726A0T-E/CH	SOT23-6
S1	Mikrospínač		
Q1	Tranzistor N-MOSFET	BSS138	SOT23
R1	Rezistor	4,7 k Ω	0805
R2	Rezistor	4,7 k Ω	0805
R3	Rezistor	910 k Ω	0805
R4	Rezistor	470 Ω	0805
R5	Rezistor	10 k Ω	0805
R6	Rezistor	1 k Ω	0805
R7	Rezistor	470 Ω	0805
C1	Kondenzátor	100 nF	0805
C2	Kondenzátor	100 nF	0805
C3	Kondenzátor	10 nF	0805
C4	Kondenzátor	10 nF	0805
C5	Kondenzátor	10 μ F	1206
C6	Kondenzátor	10 μ F	1206
C7	Kondenzátor	100 nF	0805
C8	Kondenzátor	100 nF	0805
C9	Kondenzátor	100 nF	0805
C10	Kondenzátor	100 nF	0805
C11	Kondenzátor	1 μ F	0805
C12	Kondenzátor	1 μ F	0805
C13	Kondenzátor	1 μ F	0805
FB1	Feritová perla	BLM31AJ601SN1L	1206
J2	Konektor		AVR-HDR-6

5. NÁVRH DESKY PLOŠNÉHO SPOJE

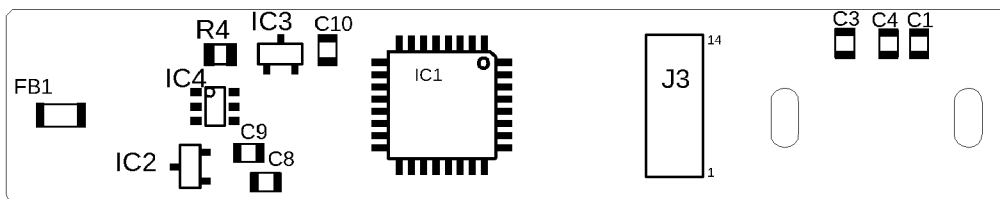
Z důvodů uspořené místa byla většina součástek vybrána pro montáž SMD a deska plošných spojů byla navrhována jako dvoustranná. Pro displej, který je napájený přímo na desce bylo zapotřebí vytvořit knihovnu a navrhnout patici odpovídající přesným rozměrům patice na displeji, aby bylo možné displej zapájet, dále byla upraveny knihovny pro enkodér, který je montovaný v horizontální poloze, ale u výrobce enkodéru je k dispozici pouze knihovna s paticí pro vertikální montáž. Pro montáž mikrosplínače byly na desce navrženy pouze pájecí plošky a spínač byl zapojen pomocí pohyblivých přívodů. Na desce byly také vytvořeny pájecí plochy pro připojení propojovacího kabelu k zesilovači, a patice pro šesti-pinový programovací konektor, která je osazena dvouřadou kolíkovou lištou.

Na desce bylo důležité přesné umístění enkodéru a patice displeje z důvodu vizuální podoby hotového ovladače, ostatní součástky byly na desce rozmístěny tak aby deska měla co nejmenší rozměry (obr. 5.1, obr. 5.2) a spoje na desce byly co nejkratší.

Deska plošného spoje byla navržena pro umístění do krabičky od výrobce Hammond Manufacturing s označením 1551KBK. Tato krabička má venkovní rozměr 80 mm x 40 mm x 20 mm. Z důvodů umístění displeje na desce bylo zapotřebí navrhnout desku takových rozměrů, aby se vešla do krabičky na výšku, výsledná velikost desky je 66 mm x 13 mm.

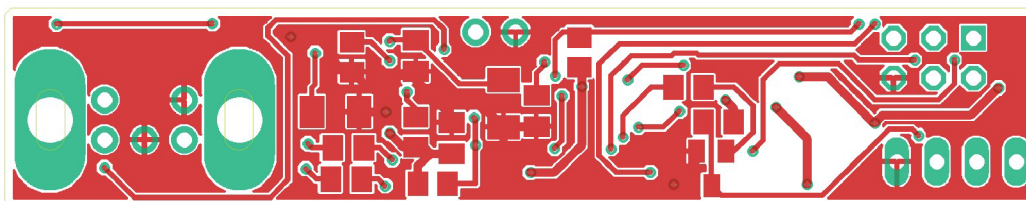


Obr. 5.1 Rozmístění součástek na desce plošných spojů strana Top.

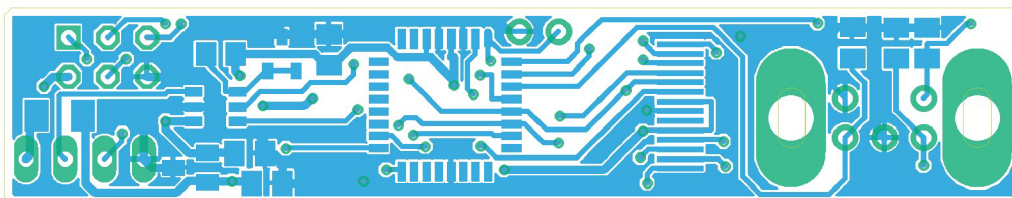


Obr. 5.2 Rozmístění součástek na desce plošných spojů strana Bottom.

Z důvodu malého rozměru plošného spoje a poměrně velkého množství cest na malé ploše je použita šířka čar pouze 0,012 inch, což odpovídá šířce 0,3048 mm pro logické cesty a 0,02 inch, což odpovídá šířce 0,508 mm pro napájecí rozvody. Tato šířka čar by měla být naprosto dostačující, protože obvodem nebudou protékat velké proudy. Prázdná místa na desce byla vyplněna polygonem spojeným s vodičem GND (obr. 5.3, obr. 5.4).



Obr. 5.3 Deska plošných spojů strana Top.



Obr. 5.4 Deska plošných spojů strana Bottom.

6. PROGRAM PRO MIKROKONTROLÉR

Program pro mikrokontrolér byl navrhnut v programu Atmel Studio 7.0, který je přímo určený pro tvorbu programu pro mikrokontroléry společnosti Atmel. Tento program byl použit i k nastavení mikrokontroléru a nahrání programu. Pro připojení k počítači byl použit jednoduchý USB programátor komunikující přes rozhraní ISP.

6.1 Použité knihovny

6.1.1 avr/io.h

Tato knihovna slouží k ovládání hardwaru mikroprocesoru, konkrétně k definování I/O registrů.

6.1.2 util/delay.h

Funkce v této knihovně slouží k ovládání funkce delay (čekání programu). Po jejím použití je možné zadávat skutečné časy v milisekundách nebo nanosekundách místo počtu cyklů. Pro správnou funkci knihovny je zapotřebí aby byla v programu nadefinována frekvence taktování mikrokontroléru. Nadefinování se provede příkazem `#define F_CPU 8000000UL` pro frekvenci taktování 8MHz.

6.1.3 i2cmaster.h

Tato knihovna slouží pro komunikaci pomocí rozhraní I2C.

6.1.4 avr/interrupt.h

Tato knihovna obsahuje nástroje pro přerušení programu. Pro přerušení programu je v programu použit příkaz `ISR(INT0_vect)`.

6.1.5 avr/eeprom.h

Tato knihovna obsahuje soubory potřebné k zápisů a zpracování dat na EEPROM paměti.

6.2 Funkce main

6.2.1 Definice vstupů a výstupů

```
DDRD |= (1 << PORTD4);  
PORTD |= 0b00001111;  
DDRC |= (1 << PORTC3);
```

V prvním řádku je definován pin PD4 jako výstup, jedná se o reset displeje. Na druhém řádku jsou na pinech PD0 - PD3 zapnuty pull-up rezistory, tyto piny jsou využity jako vstup pro enkodér a mikrospínač. Na třetím řádku definován pin C3 jako výstup, zde se jedná o signál standby. Ostatní používané piny se automaticky přiřadí k používaným

periferiím při jejich aktivaci.

6.2.2 Nastavení periferií

```
PCICR |= (1 << PCIE2);  
PCMSK2 |= (1 << PCINT16);  
  
TCCR0A = 0;  
TCCR0B = 0b00000101;  
TIMSK0 = 1<<TOIE0;
```

Na prvním řádku je nastaveno přerušování programu na pinech PCINT16 – 23. Na druhém řádku je poté povoleno pouze přerušování na pinu PCINT16. Následně je nastaven čítač na normální mód, jeho vstupní hodinový signál je nastaven předděličkou kmitočtu na frekvenci 7812 kHz, a je nastaveno povolení přerušování při přetečení čítacího registru.

6.2.3 Inicializace displeje

Nejprve je zapotřebí displej restartovat, to je provedeno na prvních čtyřech řádcích kódu. Poté nastává samotná inicializace displeje, nastavení skutečné šířky displeje na 16 pixelů, zapnutí interní nábojové pumpy, nastavení konfigurace pinů COM, zapnutí displeje a nastavení směru vykreslování sloupců a řádků na displej.

```
PORTD &= 0b11101111;  
_delay_ms(100);  
PORTD |= 0b00010000;  
_delay_ms(100);  
  
i2c_init();  
_delay_ms(200);  
  
command(0xA8); //set multiplex ratio...  
command(0x0F); //...to 16?1  
command(0x8D); //charge pump setting...  
command(0x14); //...Enable charge pump  
command(0xDA); //set COM pins hardware configuration...  
command(0x02); //...to sequential COM pin configuration  
command(0xAF); //set display on  
command(0xC8); //set COM output direction (remap)  
command(0xA1);
```

6.2.4 Hlavní smyčka programu

V hlavní smyčce programu je v prvním kroku provedena kontrola, zda jsou stisknutá tlačítka mute nebo standby.

Nejprve se kontroluje tlačítko standby, v případě že je stisknuté, je provedena kontrola, zda je hodnota proměnné swStandby rovna 0, v případě že je, přepíše se tato hodnota na jedna a hodnota proměnné Standby se zneguje, běží-li smyčka znovu a tlačítko je stále zmáčknuté je hodnota proměnné swStandby rovna 1 a již se nic nestane. Jeli tlačítko puštěno, hodnota proměnné swStandby se změní opět na 0. Při dalším stisknutí tlačítka dojde opět k negaci proměnné Standby.

Podobným způsobem je přepínána i funkce mute, zde jsou použity proměnné swMute a Mute. Rozdílem je podmínka, která má na starosti provést změnu stavu proměnné Mute

pouze je-li hodnota proměnné Standby rovna nule, tím je ošetřeno, aby nedocházelo ke změnám nastavení, pokud je zesilovač vypnutý.

```
if(!(PIND & (1 << PIND2))){ //Standby tlačitko
    if(swStandby == 0){
        swStandby = 1;
        Standby = !Standby;
    }
}
else{
    swStandby = 0;
}

if(!(PIND & (1 << PIND3))){ //Mute tlačitko
    if(swMute == 0){
        swMute = 1;
        if(!Standby)
            Mute = !Mute;
    }
}
else{
    swMute = 0;
}
```

Poté nastává v programu část, kdy jsou přepínány jednotlivé módy, běžný provoz, mute a standby (obr. 6.1).

```
switch (mod)
{
    case 0:
        _delay_ms(10);
        bar(count);
        writeDacReg(count * 1.449);
        PORTC &= !(1 << PORTC3);
        prevCount = count;
        if(Mute)
            mod = 1;
        if(Standby)
            mod = 2;
        break;
}
```

V case 0 je nejprve volaná funkce, která zobrazí hodnotu na displeji. Následuje funkce, která nastavuje hodnotu D/A převodníku, protože proměnná count, která slouží k nastavení hlasitosti je od 0 do 69, je zapotřebí hodnotu vynásobit konstantou 1,449. Tím docílíme toho, že rozsah hodnot posílaných na D/A převodník je od 0 do 100, napětí na převodníku je poté 0 až 1 V. Následně je na pin PC3 poslána logická nula, tím je na konektor standby přivedeno napětí 5 V. Poté je hodnota proměnné count uložena do proměnné prevCount. Následně je příkazy if kontrolováno, zda nebylo zmáčknuto tlačítko mute nebo standby, v případě že ano, je přepnuto na mod 1 nebo 2.

```
case 1:
    _delay_ms(10);
    symbol(0);
    writeDacReg(0);
    PORTC &= !(1 << PORTC3);
    count = prevCount;
    if(!Mute)
```

```

        mod = 0;
    if (Standby)
        mod = 2;
    break;

```

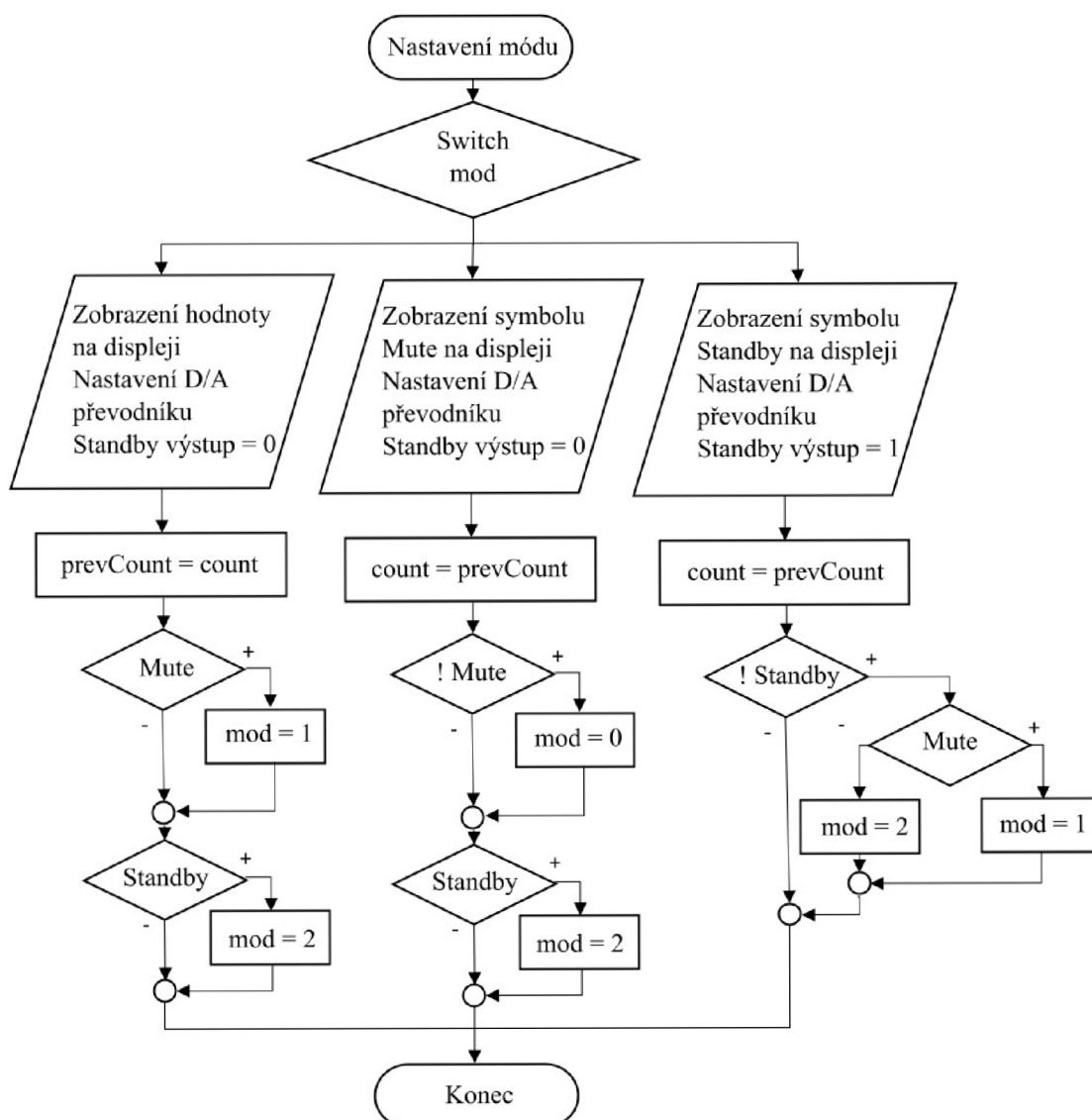
V case 1 je volaná funkce symbol, která zobrazuje na displeji symbol mute (v případě že je parametrem 0) a symbol standby (v případě že je parametrem 1). Následně je D/A převodníku poslána nula. Na pin PC3 je stejně jako v case 0 posílána logická nula. Do proměnné count se nahrává hodnota prevCount, hodnota prevCount je uložena v case 0 a v case 1 a case 2 ji není možné změnit, proto i kdyby došlo ke změně hodnoty count v režimu mute nebo standby, po návratu do stavu běžné funkce (case 0) bude hodnota count stejně jako při jejím opuštění. Poté opět následují příkazy if, které mají za úkol zkontrolovat změnu stavu tlačítek a přepnout na jiný case.

```

        case 2:
            _delay_ms(10);
            symbol(1);
            writeDacReg(0);
            PORTC |= (1 << PORTC3);
            count = prevCount;
            if (!Standby) {
                if (Mute) {
                    mod = 1;
                }
                else {
                    mod = 0;
                }
            }
            break;
    }

```

V case 2 je volaná funkce symbol, která zobrazí hodnotu na displeji, Následuje posláni hodnoty nula na D/A převodník. Poté se pin PC3 přepne na logickou 1, tím je na výstupu ovladače sloužící jako standby přivedeno napětí 0 V. Do proměnné count se ukládá hodnota proměnné prevCount jako v case 1. Poté opět následují příkazy if, zde je ovšem nejprve kontrolováno, zda hodnota proměnné Standby je v logické nule, je-li, je kontrolován stav proměnné Mute a poté je přepnuto na case 0 nebo 1. Když je proměnná Standby rovna jedné, zůstane v provozu case 2.



Obr. 6.1 Vývojový diagram přepínání módů.

6.3 Nastavení hodnoty volume

Pro nastavení hodnota volume je použita proměnná count. Ta je nastavována při vyvolání obsluhy přerušení. Toto přerušení je vyvoláno změnou stavu na pinu PD0, na který je připojen jeden ze spínačů enkodéru. Podmínkou if se kontroluje směr točení enkodéru. V případě, že je po změně stavu pinu PD0 na PD0 opačná hodnota, než na pinu PD1, jedná se o otočení enkodérem proti směru hodinových ručiček. V tomto případě je od proměnné count odečítaná proměnná inkrement. V opačném případě se proměnná zvyšuje o inkrement. Dochází také ke kontrole, zda-li je hodnota proměnné mezi 0 až 69.

```

ISR(PCINT2_vect)
{
cli();
    if(((PIND & (1 << PIND0)) && !(PIND & (1 << PIND1))) || (!(PIND &
(1 << PIND0)) && (PIND & (1 << PIND1)))){
        count = count - inkrement;
        if(count >= 69){
            count = 69;
        }
        if(count <= 0){
            count = 0;
        }
    }
    else{
        count = count + inkrement;
        if(count >= 69){
            count = 69;
        }
        if(count <= 0){
            count = 0;
        }
    }
    TCNT0 = 0;
    inkrement = 3;
    eepromWrite = 100;
    sei();
}

```

Po změně hodnoty proměnné count je čítací registr TCNT0 nastaven do nuly, inkrement je nastaven na hodnotu 3 a proměnná eepromWrite je nastavená na 100. Jakmile dojde k přetečení čítače, což je 30,5 ms po vynulování čítacího registru, dojde ke změně hodnoty proměnné na jedna. Tímto se mění rychlost nastavování hodnoty volume, je-li čas mezi změnami hodnoty enkodéru menší než 30,5 ms, mění se hodnota proměnné count o tři, je-li čas větší, mění se pouze o jedna. Čímž je umožněno jemnější nastavení hlasitosti.

Při každém přetečení se hodnota proměnné eepromWrite sníží o jedna, dokud nedosáhne hodnoty eepromWrite = 1, v tento moment dojde k zapsání hodnoty count na paměť eeprom a proměnná eepromWrite se nastaví na nula. Tím je zajištěno, že se ukládání nebude stále opakovat, ale hodnota count se uloží až 3 s po poslední změně hodnoty. Tato hodnota je po spuštění přístroje přečtena z eeprom a nahrána do proměnné count.

```

ISR(TIMERO_OVF_vect)
{
    inkrement = 1;
    if(eepromWrite == 1){
        eeprom_update_byte (( uint8_t *) 64, count);
        eepromWrite = 0;
    }
    if(eepromWrite > 0)
        eepromWrite--;
}

```

6.4 Funkce pro ovládání displeje

```
void command(unsigned char com){
    i2c_start_wait((ADR_OLED<<1)+I2C_WRITE);
    i2c_write(0x00);
    i2c_write(com);
    i2c_stop();
}
```

Tato funkce slouží k zasílání příkazů do displeje. Nejprve je poslána adresa displeje a s ní je poslán Write bit, který určuje, že se do displeje bude zapisovat. Následně je do displeje zaslán kontrolní byte, který na sedmém místě obsahuje D/C#, tento bit určuje, zda budou následovat data (logická jedna) nebo příkazy (logická nula). Následující byte obsahuje samotný příkaz, který do funkce vstupuje jako parametr. Poté je komunikace zastavena.

```
void dataByteSend(unsigned char data, int pocetbyte){
    i2c_start_wait((ADR_OLED<<1)+I2C_WRITE);
    i2c_write(0x40);
    for(int i = 0; i < pocetbyte; i++){
        i2c_write(data);
    }
    i2c_stop();
}
```

Tato funkce má na starosti odesílání dat do displeje. Nejprve je odeslána adresa a kontrolní byte, s tím rozdílem, že bit D/C# je v tomto případě logická jedna, což určuje, že následující data budou zobrazena na displeji. Následuje cyklus, ve kterém posíláme do displeje databyte, ten je možné do displeje odeslat několikrát za sebou, pomocí parametru pocetbyte. Po odeslání dat je komunikace ukončena.

6.5 Funkce pro ovládání D/A převodníku

Nejprve je zapotřebí provést konfiguraci D/A převodníku. K tomu slouží vlastní funkce writeDacConf(). Komunikace je zahájena stejně jako u displeje, a to vysláním adresy D/A převodníku spolu s write bitem, následuje konfigurační byte. Tím nastavíme, aby referenční napětí bylo bráno z referenčního vstupu D/A převodníku, nastavení normálního provozu výstupního zesilovače a zesílení zesilovače 1x.

```
char writeDacConf(){
    i2c_start_wait((ADR_DAC<<1)+I2C_WRITE);
    i2c_write(0b10010000);
    i2c_stop();
    return 0;
}
```

Další funkce slouží k zápisu dat do registru D/A převodníku. Vstupním parametrem funkce jsou data v rozsahu 0 – 100, která je před odesláním zapotřebí přepočítat pro zvolenou napěťovou referenci 1,24 V, tak aby výstupní napětí bylo v rozsahu 0 – 1 V. Protože je D/A převodník dvanáctibitový, je zapotřebí data rozdělit do dvou bajtů. Nejprve jsou vysílány vrchní čtyři bity a hned poté zbylé bity.

```
char writeDacReg(unsigned int data){
    data = (4095.0/REFNAP)*data;
    char dataLow, dataHigh;
    dataLow = data & 0b11111111;
    data = data >> 8;
    dataHigh = data & 0b00001111;

    i2c_start_wait((ADR_DAC<<1)+I2C_WRITE);
    i2c_write(dataHigh);
    i2c_write(dataLow);
    i2c_stop();
    return 0;
}
```

6.6 Zobrazení hodnot na displeji

```
void clear(){
    unsigned char page = 0xB0;
    for(int j = 0; j < 2; j++){
        command(0x00);
        command(0x10);
        page = 0xB0 + j;
        command(page);

        dataByteSend(0x00, 96);
    }
}
```

Tato funkce slouží ke smazání displeje, to je provedeno posláním nulových hodnot na jednotlivé řádky displeje. Nejprve je nastaven ukazatel displeje na začátek řádku, pomocí příkazů 0x00 a 0x10, a na levý sloupec je 96x poslána hodnota 0, v druhém průchodu cyklu for je vynulována druhá polovina displeje. Nulová hodnota reprezentuje zhasnuté pixely.

Další funkce slouží k vykreslování nastavené hodnoty hlasitosti (obr. 6.2). Vstupním parametrem je proměnná pocet, která určuje počet zobrazovaných čárek. Čárky jsou z důvodu přehlednosti soustředěny do bloků po třech, maximální hlasitost je reprezentována 23 plnými bloky. Nejprve je nastaven ukazatel displeje na začátek řádku, poté je do levého sloupce vykreslen první a druhý řádek reprezentující začátek rámečku. Následně jsou v cyklu vykresleny celá bloky s prázdnými mezerami mezi bloky jeden pixel. Poté jsou vykresleny zbylé čárky a zbytek displeje je vykreslen prázdným rámečkem.

```
void bar(int pocet){
    uint8_t pocetBloku = pocet/3;
```

```

command(0x00);
command(0x10);
command(0xB0);

dataByteSend(0xFF, 1);
dataByteSend(0x01, 1);
for(int i = pocetBloku; i > 0; i--){
    dataByteSend(0xFD, 3);
    dataByteSend(0x01, 1);
}
dataByteSend(0xFD, pocet % 3);
dataByteSend(0x01, 92 - (pocet + pocetBloku));
dataByteSend(0xFF, 1);

```

To samé je následně provedeno i u pravé části displeje, pouze s úpravou hodnot pro zrcadlové vykreslení displeje.

```

command(0x00);
command(0x10);
command(0xB1);

dataByteSend(0xFF, 1);
dataByteSend(0x80, 1);
for(int i = pocetBloku; i > 0; i--){
    dataByteSend(0xBF, 3);
    dataByteSend(0x80, 1);
}
dataByteSend(0xBF, pocet % 3);
dataByteSend(0x80, 92 - (pocet + pocetBloku));
dataByteSend(0xFF, 1);
}

```

Funkce symbol slouží k zobrazení symbolu mute (obr. 6.3) a standby (obr. 6.4) na obrazovce. To, jaký symbol bude na obrazovce vykreslen, určuje vstupní proměnná Sym. Nejprve je na displej v levém sloupci vykresleno 38 prázdných řádků. Následuje cyklus for který po jednotlivých řádcích vykresluje 19 řádků dat zapsaných v poli vytvořeného symbolu, poté následuje vypsání 39 prázdných řádků. V pravém sloupci displeje je také nejprve vykresleno 38 prázdných řádků a následuje cyklus for, který vykreslí zbylé řádky dat z pole symbolu a následně jsou vypsány opět prázdné řádky.

```

void symbol(int Sym){

    command(0x00);
    command(0x10);
    command(0xB0);

    dataByteSend(0x00, 38);

    for(int i = 0; i < 19; i++){
        if(Sym){
            dataByteSend(StandbySymbol[i], 1);
        }
        else{
            dataByteSend(MuteSymbol[i], 1);
        }
    }

    dataByteSend(0x00, 39);
}

```



```

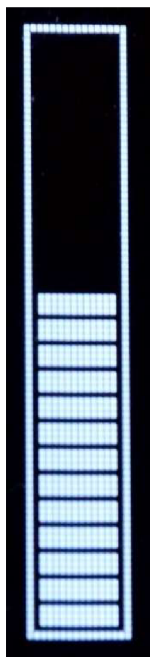
command(0x00);
command(0x10);
command(0xB1);

dataByteSend(0x00, 38);

for(int i = 19; i < 38; i++){
    if(Sym){
        dataByteSend(StandbySymbol[i],1);
    }
    else{
        dataByteSend(MuteSymbol[i],1);
    }
}
dataByteSend(0x00, 39);
}

```

Podoba zobrazovaných symbolů je následující:



Obr. 6.2 Zobrazení nastavené hodnoty volume.



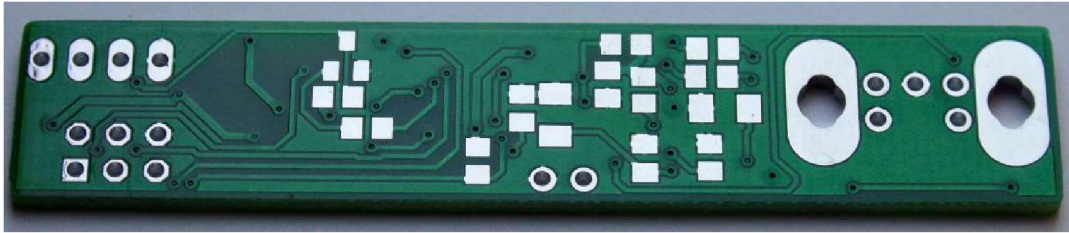
Obr. 6.3 Zobrazení symbolu Mute.



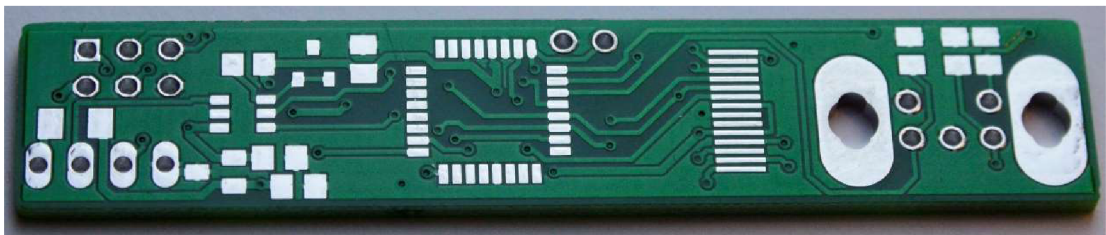
Obr. 6.4 Zobrazení symbolu Standby.

7. KOMPLETECE DIGITÁLNÍHO OVLADAČE HLASITOSTI

Výroba DPS byla realizována firmou JLPCB. Jedná se o dvouvrstvý plošný spoj o tloušťce 1,6 mm, s prokovenými otvory s nanesenou nepájkivou maskou (obr. 7.1, obr. 7.2).

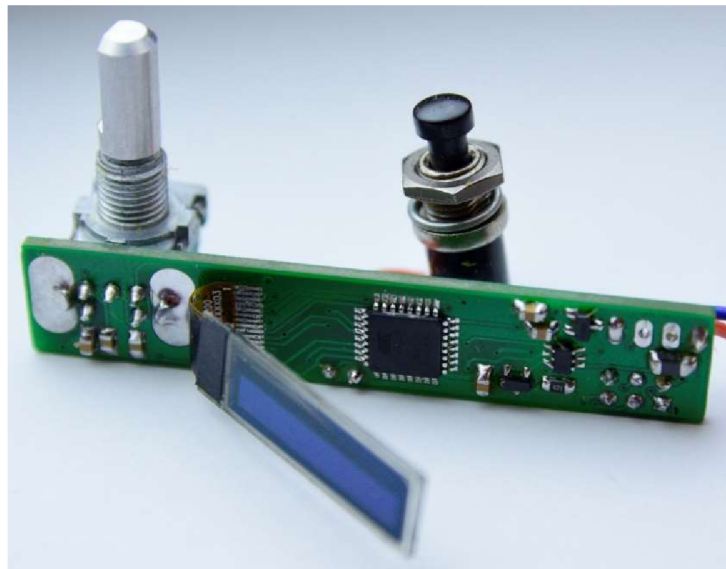


Obr. 7.1 Deska plošných spojů strana Top.

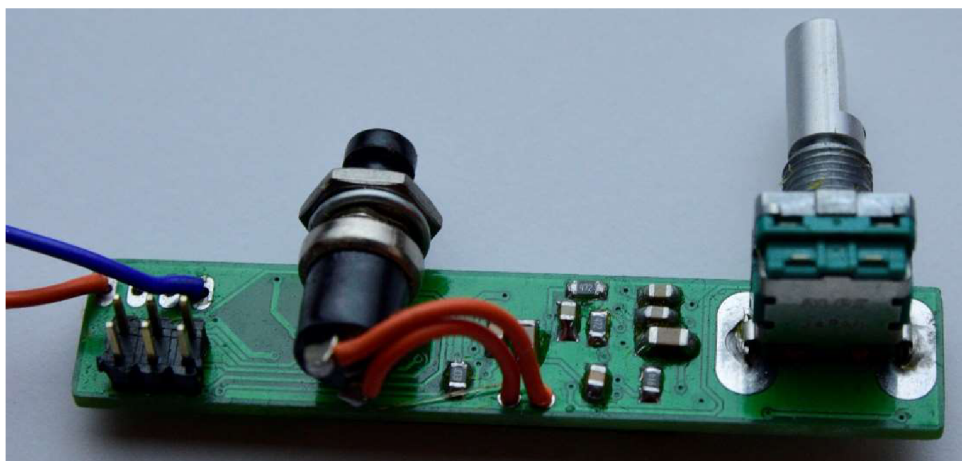


Obr. 7.2 Deska plošných spojů strana Bottom.

Vyrobené DPS byly následně ručně osazeny všemi součástkami (obr. 7.3, obr. 7.4). Obvod byl vyzkoušen a do mikrokontroléru byl nahrán program.



Obr. 7.3 Osazená deska plošných spojů strana Top.



Obr. 7.4 Osazená deska plošných spojů strana Bottom.

Aby bylo možné zařízení namontovat do vybrané krabičky, bylo zapotřebí do krabičky navrtat díry na enkodér, mikrosínač a přívodní kabel a proříznout otvor pro displej (obr. 7.5).



Obr. 7.5 Krabička Hammond Manufacturing 1551KBK s připravenými otvory.

Pro přívodní kabel byla použita gumová průchodka s venkovním průměrem 6,3 mm a vnitřním 4 mm. Kabel byl proti vytržení zajištěn stahovacím páskem, který nelze po kabelu posouvat, a tudíž nelze kabel vytrhnout. DPS drží pouze na enkodéru, který je přišroubován na stěnu krabičky. Aby se displej v krabičce nehýbal, je k DPS přilepený pomocí oboustranné lepicí pásky (obr. 7.6).



Obr. 7. 6 Digitální ovladač hlasitosti umístěný v krabičce.



Obr. 7. 7 Hotový digitální ovladač hlasitosti

8. ZÁVĚR

V první části bakalářské práce byly vybrány potřebné součástky pro sestavení digitálního ovladače hlasitosti, následně byl proveden kompletní návrh schématu a poté desky plošných spojů. V další části byl vytvořený kompletní zdrojový kód pro digitální ovladač s grafickou indikací, který nastavuje výstupní napětí na D/A převodníku, spíná tranzistor pro ovládání režimu standby, počítá hodnotu nastaveného volume a zobrazuje na displeji stav volume, režimy mute a standby.

Dalším krokem bylo sestavení prototypu digitálního ovladače, nahrání programu, odzkoušení a změření všech funkcí. Výstupní napětí pro ovládání volume se nastavuje v rozmezí 0 – 1 V a výstup standby má napětí 0 V při spuštění režimu standby a napětí 5 V při jeho vypnutí. Při nastavování hodnoty volume nejprve kontrolér vynechával některé polohy enkodéru, což bylo špatným výběrem kondenzátorů, zmenšením těchto kondenzátorů 10x bylo docíleno bezproblémové funkce nastavování hodnot volume.

Krabička pro umístění zařízení byla ručně navržena a byl do ní vyřiznut otvor na displej, jelikož je otvor malý, jsou na něm vidět i nepatrné nepřesnosti.

LITERATURA

- [1] KEFAUVER, Alen a David PATSCHKE. Fundamental of digital audio. New Edition. Wisconsin 53562: A-R Editions, 2007. ISBN 978-0-89579-611-0.
- [2] MATOUŠEK, David. Práce s mikrokontroléry ATMEL AT89C2051. Praha: BEN - technická literatura, 2002. ISBN 80-7300-048-2.
- [3] Atmel [online katalogový list]. ATmega328P. ©2015 [cit. 20.5.2021]. Dostupné online z: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [4] OLED. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-5-20]. Dostupné z: <https://cs.wikipedia.org/wiki/OLED>
- [5] Electronic Assembly [online katalogový list]. Micro OLED 0.84" - 96x16 dots. ©2020 [cit. 20.5.2021]. <https://www.mouser.com/datasheet/2/127/W096016-XA-1275487.pdf>
- [6] SOLOMON SYSTECH [online katalogový list]. SSD1306B. ©2012 [cit. 20.5.2021]. Dostupné online z: https://www.lcd-module.com/fileadmin/eng/pdf/zubehoer/SSD1306B_1.1.pdf
- [7] D/A převodník. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2019 [cit. 2021-5-20]. Dostupné z: https://cs.wikipedia.org/wiki/D/A_p%C5%99evodn%C3%ADk.
- [8] Mikrochip [online katalogový list]. MCP4706/4716/4726. ©2011-2012 [cit. 20.5.2021]. Dostupné online z: <https://ww1.microchip.com/downloads/en/DeviceDoc/22272C.pdf>
- [9] PAŠTYKA, Jan a Ondřej HRUŠKA. Rotační enkodér [online]. Katedra měření, 2017 [cit. 2021-5-20]. Dostupné z: https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/rotary_encoder/Rotary_Encoder.pdf. Dokumentace. České vysoké učení technické v Praze Fakulta elektrotechnická.
- [10] NEČÁSEK, Sláva. Radiotechnika do kapsy: Kapitola Stabilizátory napětí. Praha 1: SNTL, 1981. ISBN 04-537-81.
- [11] BUMBA, Jiří. Programování mikroprocesorů: praktický návod nejen pro mikroprocesory PIC. Brno: Computer Press, 2011. ISBN 978-80-251-2838-1.
- [12] BOUREK, Jiří. Programování mikroprocesor AVR v jazyce C [online]. 2007 [cit. 2021-5-20]. Dostupné z: <http://noel.feld.cvut.cz/vyu/scs/prezentace2007/avr-libc/avr-libc.pdf>.

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

DPS	Deska plošných spojů
GND	Ground
I2C	Inter-Integrated Circuit
OLED	Organic light-emitting diode
SMD	Surface mount device
SPI	Serial Peripheral Interface
THT	Through-hole technology
TWI	Two Wire Interface
UART	Universal asynchronous receiver-transmitter

Symboly:

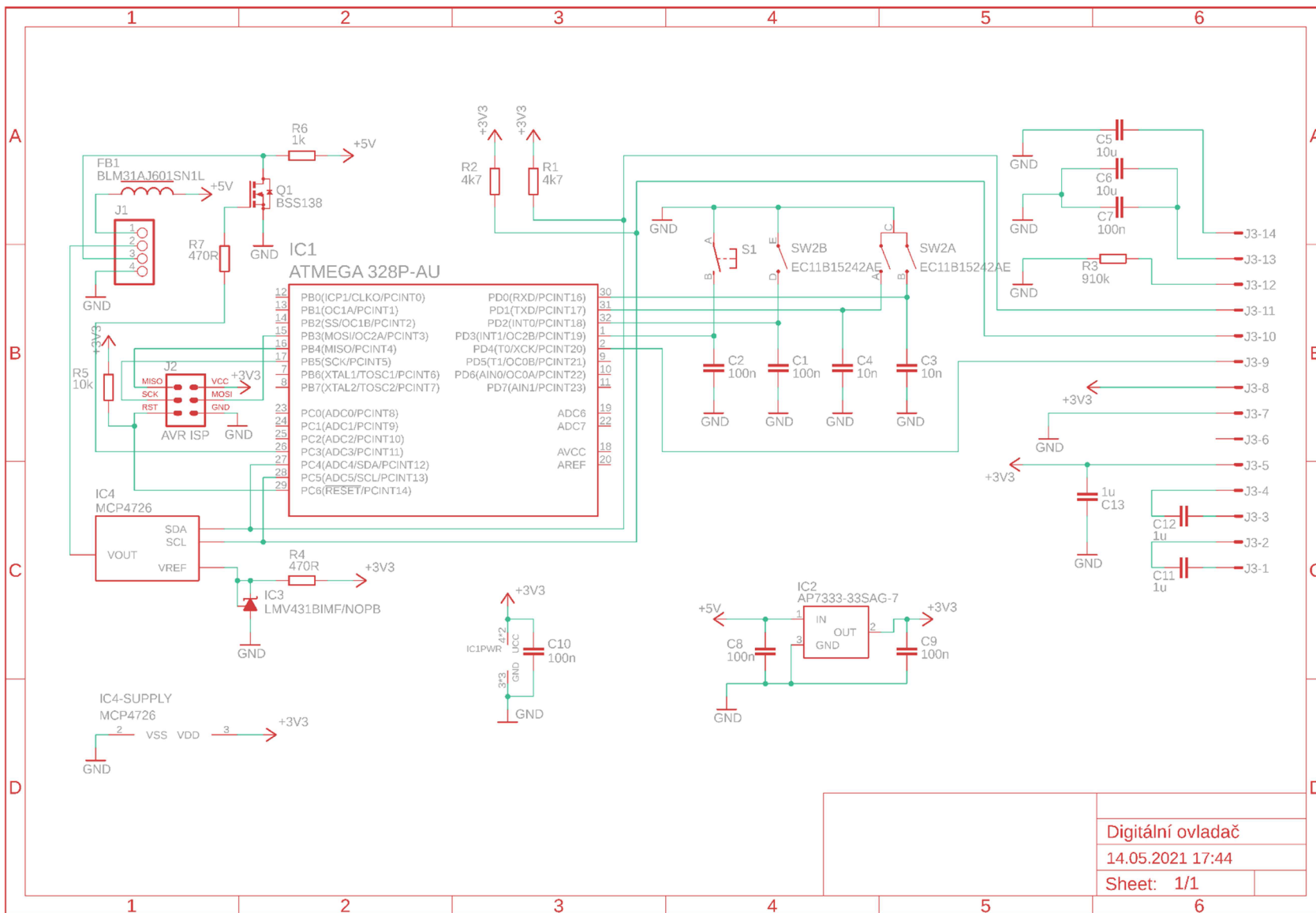
U	napětí	(V)
I	proud	(A)
τ	časová konstanta	(s)
R	odpor	(Ω)
C	kapacita	(F)
f	frekvence	(Hz)

SEZNAM PŘÍLOH

PŘÍLOHA A - SCHÉMA ZAPOJENÍ..... 44

PŘÍLOHA B - ZDROJOVÝ KÓD PROGRAMU JE ULOŽEN V DIGITÁLNÍ PODOBĚ

Schéma zapojení



Digitální ovladač

14.05.2021 17:44

Sheet: 1/1