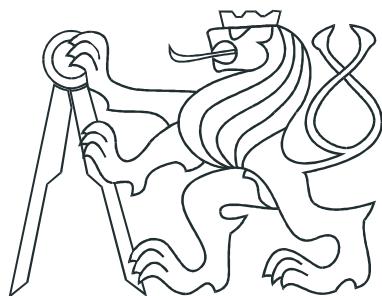


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

Řídicí modul modelové lokomotivy pro  
dopravní laboratoř FD ČVUT

Praha, 2011

Autor: Bc. Martin Hönig



## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady ( literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne

---

podpis

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Vítovi Fáberovi, Ph.D. za studijní materiály, konzultace a vedení v průběhu zpracování této diplomové práce. Děkuji Ing. Martinu Leso, Ph.D za možnost práce na velmi zajímavém projektu a za materiální podporu, bez které by nebyla realizace řešení možná. Dále děkuji Pavlovi Hübnerovi ml. za cenné konzultace během návrhu elektroniky a Ing. Karlovi Duškovi, Ph.D. za pomoc při osazování Bluetooth modulů na specializovaném pracovišti.

# **Abstrakt**

Cílem této diplomové práce je dokončení vývoje řídicího modulu pro modelové lokomotivy, které budou nasazeny v budovaném dopravním sále Fakulty dopravní ČVUT. Řídicí modul má za úkol převzít funkci komerčně vyráběných modulů s dodatečnou funkčností požadovanou pro jeho nasazení v dopravním sále. Jmenovitě je požadováno zpětnovazebně řídit otáčky motoru na požadované hodnotě ve všech možných režimech zatížení soupravy, sbírat a přepořídat data o poloze a ujeté vzdálenosti soupravy nadřazenému řídícímu systému a zastoupit funkci zvukového modulu. Pro oboustranný přenos je využita bezdrátová technologie Bluetooth. Práce má za úkol dle zadaných požadavků a dosavadních zkušeností (viz. (PAVLÍŠTA, D., 2007), (HÖNIG, M., 2008), (HÖNIG, M., 2009)) vybrat dostatečně výkonnou a vhodnou mikroprocesorovou platformu, která pokryje dodatečné požadavky na řídicí modul.

# **Abstract**

The goal of these diploma thesis is finishing development of control module for model locomotive, which will be operate on emerging transportation laboratory at CTU, Faculty of Transportation Sciences. Control module should take over a capabilities of commercially accessible control modules with some extra additional capabilities, which is required for operating at transportation laboratory. Namely it should be capable of feedback controlling of motor at desired value at all possible load, gather and report position of train and distance moved to superior system and take over a function of sound module. Bluetooth wireless technology is used for bilateral communication. Thesis should determine a sufficient performance microprocessor platform based on mentioned requirements and present experience (see. (PAVLÍŠTA, D., 2007), (HÖNIG, M., 2008), (HÖNIG, M., 2009)).

vložit originální zadání !!!!



# Obsah

|  |           |
|--|-----------|
| <b>Seznam obrázků</b>  | <b>ix</b> |
| <b>Seznam tabulek</b>  | <b>xi</b> |
| <b>1 Úvod</b>  | <b>1</b>  |
| 1.1 Přehled současného stavu . . . . .                               | 1         |
| 1.2 Návrh finálního řešení . . . . .                                 | 4         |
| <b>2 Řešení migrace na novou platformu</b>                           | <b>7</b>  |
| 2.1 Analýza požadavků na novou platformu . . . . .                   | 7         |
| 2.1.1 Definice hodnotících kritérií . . . . .                        | 7         |
| 2.1.2 Porovnání sledovaných parametrů . . . . .                      | 9         |
| 2.1.3 Nároky na výpočetní výkon . . . . .                            | 10        |
| 2.1.4 Diskuze nad porovnání kandidátů na novou platformu . . . . .   | 14        |
| 2.2 Migrace funkčního kódu na novou platformu . . . . .              | 15        |
| 2.2.1 Ověření splnění původní funkčnosti . . . . .                   | 16        |
| <b>3 Návrh a řešení zvukového modulu</b>                             | <b>17</b> |
| 3.1 Hardwarové řešení zvukového modulu . . . . .                     | 17        |
| 3.2 Softwarové řešení zvukového modelu . . . . .                     | 20        |
| 3.2.1 Modelování dle DP M. Hajného . . . . .                         | 20        |
| 3.2.2 Modelování dle Microsoft Train Simulator . . . . .             | 21        |
| 3.2.3 Modelování pro vlastní potřebu zvukového modulu . . . . .      | 24        |
| 3.3 Míchání více zdrojů signálu v digitální podobě . . . . .         | 24        |
| 3.4 Zvuková odezva emulace motoru na změnu rychlosti jízdy . . . . . | 27        |
| <b>4 Realizace nového řídicího modulu</b>                            | <b>29</b> |
| 4.1 Přepracovaná zdrojová část modulu . . . . .                      | 30        |

|  |   |           |
|--|---|-----------|
| 4.1.1  | Řešení se superkapacitorem . . . . .                            | 30        |
| 4.1.2  | Řešení s baterií . . . . .                                      | 34        |
| 4.1.3  | Výběr finálního řešení zálohování . . . . .                     | 35        |
| 4.2  | Implementace HCI rozhraní pro komunikaci s BT modulem . . . . . | 36        |
| 4.2.1  | HCI Command Packet . . . . .                                    | 37        |
| 4.2.2  | HCI Event Packet . . . . .                                      | 39        |
| 4.2.3  | HCI ACL/SCO Data Packet . . . . .                               | 40        |
| 4.2.4  | Proces vyjednání spojení mezi Bluetooth moduly . . . . .        | 42        |
| 4.3  | Testování alternativní odometrie . . . . .                      | 44        |
| 4.4  | Dodatečně přidané funkce . . . . .                              | 50        |
| 4.5  | Zástavba modulu do lokomotivy . . . . .                         | 51        |
| <b>5</b>   | <b>Závěr</b>  | <b>57</b> |
| <b>Literatura</b>                                    |   | <b>61</b> |
| <b>A Kompletní elektrické schema řídicího modulu</b> |   | <b>I</b>  |
| <b>B Seznam použitého software</b>                   |   | <b>IX</b> |
| <b>C Obsah přiloženého CD</b>                        |   | <b>XI</b> |

# Seznam obrázků

|      |  |    |
|------|--|----|
| 1.1  | Blokové schema stávajícího řídicího modulu . . . . .                     | 3  |
| 1.2  | Blokové schema navrhovaného přepracovaného řídicího modulu . . . . .     | 5  |
| 3.1  | Schéma zapojení zvukové části modulu . . . . .                           | 19 |
| 3.2  | Rekonstruované křivky rychlostního koeficientu přehrávaných vzorků zvuků | 23 |
| 3.3  | Rekonstruované křivky koeficientu zesílení přehrávaných vzorků zvuků . . | 23 |
| 3.4  | Simulace míchání dvou signálu v digitální podobě . . . . .               | 26 |
| 3.5  | Převodní charakteristika střídy PWM signálu na rychlosť jízdy . . . . .  | 28 |
| 4.1  | Simulace vybíjení superkapacitoru při výpadku napájení . . . . .         | 32 |
| 4.2  | Simulace nabíjení superkapacitoru při náběhu napájení . . . . .          | 33 |
| 4.3  | Schema zálohy napájení pomocí superkapacitoru . . . . .                  | 33 |
| 4.4  | Schema zálohy napájení pomocí baterie . . . . .                          | 34 |
| 4.5  | Schema komunikačního kanálu mezi zařízeními pomocí rozhraní HCI . . .    | 37 |
| 4.6  | Formát HCI Command Packetu . . . . .                                     | 38 |
| 4.7  | Formát HCI Event Packetu . . . . .                                       | 39 |
| 4.8  | Formát HCI Data Packetu . . . . .  | 41 |
| 4.9  | Porovnání údajů o rychlosti z různých zdrojů měření . . . . .            | 45 |
| 4.10 | Převodní poměr údajů o rychlosti v průběhu celého měření . . . . .       | 46 |
| 4.11 | Měření délky okruhu pomocí odometrie na bázi IR čidla . . . . .          | 48 |
| 4.12 | Měření délky okruhu pomocí odometrie na bázi BEMF . . . . .              | 48 |
| 4.13 | Schema detekce polarity napájení v kolejích . . . . .                    | 51 |
| 4.14 | Tělo lokomotivy po odvrtání a odfrézování . . . . .                      | 52 |
| 4.15 | Pohled zespodu na snímač IR balík bez plastového překrytu . . . . .      | 52 |
| 4.16 | Hotová zástavba kabeláže bez připojeného řídicího modulu . . . . .       | 53 |
| 4.17 | Hotová zástavba s osazeným řídicím modulem . . . . .                     | 54 |
| 4.18 | Pohled z boku - složení fotografie s a bez kapoty . . . . .              | 54 |
| 4.19 | Finální sestava lokomotivy s rozsvícenými hlavními světly . . . . .      | 55 |

|     |   |      |
|-----|---|------|
| A.1 | Schema zdrojové části řídicího modulu - část 1 . . . . .        | II   |
| A.2 | Schema zdrojové části řídicího modulu - část 2 . . . . .        | III  |
| A.3 | Schema procesorové části řídicího modulu - část 1 . . . . .     | IV   |
| A.4 | Schema procesorové části řídicího modulu - část 2 . . . . .     | V    |
| A.5 | Schema motorové budící části řídicího modulu - část 1 . . . . . | VI   |
| A.6 | Schema motorové budící části řídicího modulu - část 2 . . . . . | VII  |
| A.7 | Schema procesorové části řídicího modulu - část 3 . . . . .     | VIII |

# Seznam tabulek

|     |  |    |
|-----|--|----|
| 2.1 | Porovnání sledovaných parametrů mikroprocesorů . . . . .         | 11 |
| 2.2 | Periodicky volané bloky kódu . . . . .                           | 12 |
| 2.3 | Čas potřebný pro vykonání operace . . . . .                      | 14 |
| 4.1 | Formát příchozího datagramu od BT serveru . . . . .              | 41 |
| 4.2 | Význam nastavení rychlosti v příchozích datech . . . . .         | 42 |
| 4.3 | Význam nastavení světla a zvuku v příchozích datech . . . . .    | 42 |
| 4.4 | Formát standard. odchozího datagramu pro BT server . . . . .     | 42 |
| 4.5 | Formát urgentního odchozího datagramu pro BT server . . . . .    | 42 |
| 4.6 | Naměřené délky okruhů pomocí odometrie řídicího modulu . . . . . | 49 |



# Kapitola 1

## Úvod

Řídící modul je učen pro potřeby dopravního sálu budovaného v budové Fakulty dopravní ČVUT v Praze. Dopravní sál bude sloužit jako demonstrační pomůcka několika používaných zabezpečovacích systémů reálné železnice. Konkrétně se jedná o elektromechanický zabezpečovací systém (tzv. bubnový stroj), releové zabezpečovací zařízení AŽD 71 a nejnověji nasazované jednotné obslužné pracoviště dle základních technických požadavků ZTP 4 (elektronické stavědlo). Sál se sestává z kombinace obslužných stanovišť těchto zabezpečovacích systémů a modelového kolejistiště, které slouží jako živá vizualizace. Kolejiště o rozloze cca  $45\text{ m}^2$  je budováno v měřítku 1:87 (označovaném jako H0). Učebna by měla dále demonstrovat možnosti nasazení systému ETCS.

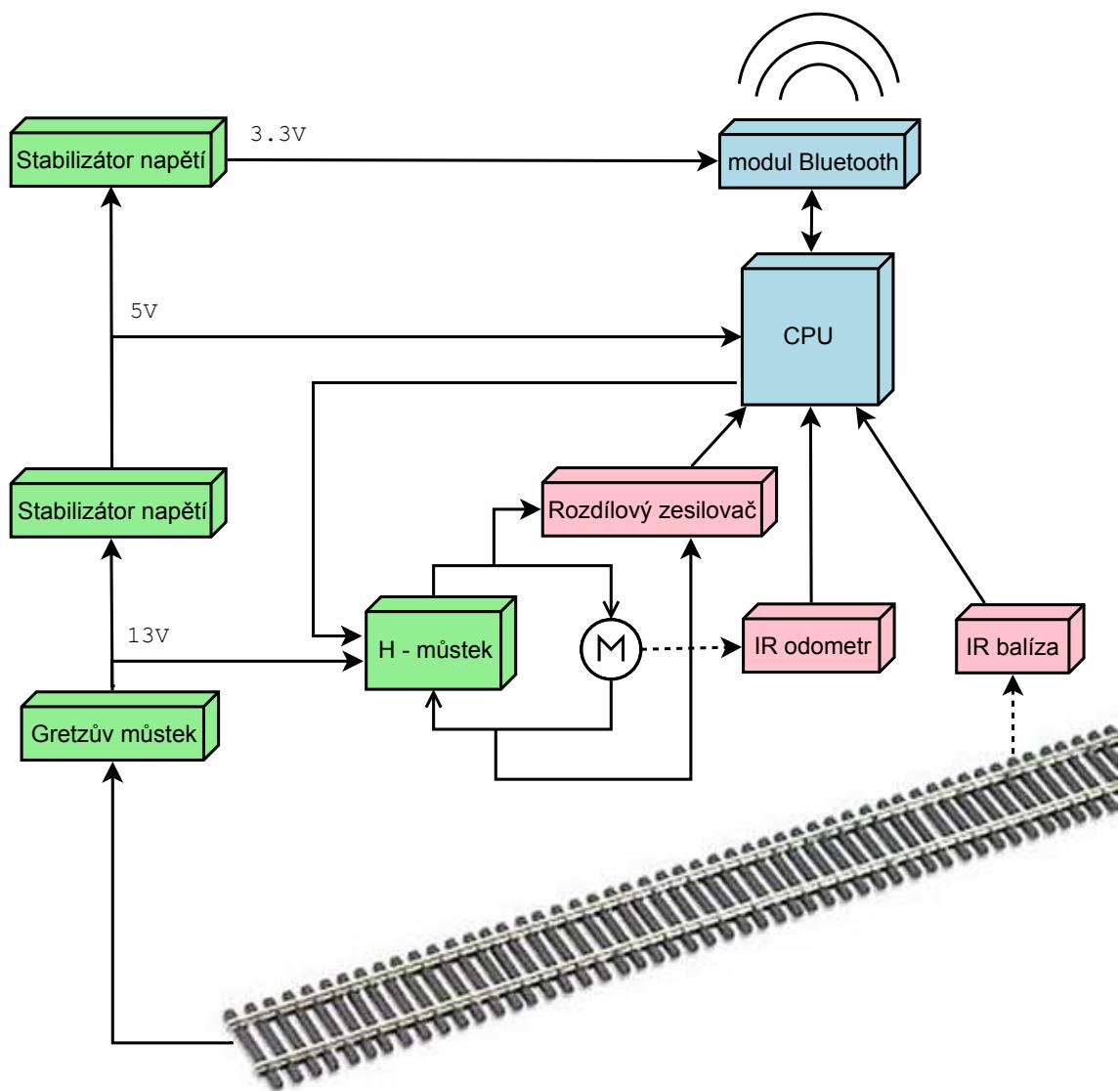
### 1.1 Přehled současného stavu

Lokomotivy oboustraně komunikují se servrem nadřízeného řídícího a zabezpečovacího systému pomocí bezdrátové technologie Bluetooth. Nadřazený systém ovládá rychlosť a světla lokomotivy. Lokomotivy hlásí nadřazenému systému aktuální rychlosť lokomotivy, ujetou vzdálenost lokomotivy a poslední detekovaný záhytný bod infrastruktury kolejistiště. Záhytné body jsou na známých místech infrastruktury a jsou realizovány jako vysílače v infračerveném spektru s kódováním Manchester.

Lokomotiva, respektive řídící modul, je napájen z kolejnic stejnosměrným napětím v rozsahu 12-14 V. Toto napětí je na modulu dvoucestně usměrněno a využíváno pro

buzení stejnosměrného motoru lokomotivy. Ostatní elektronika je připojen k stabilizovanému rozvodu napájení na úrovni 5 V. Řízení otáček motoru, sběr dat o projetých záhytných bodech infrastruktury a ujeté vzdálenosti, generování zvukových efektů zajišťuje mikroprocesor. Komunikace probíhá pomocí sériové linky, která je transparentně replicována modulem Bluetooth k nadřazenému systému na straně serveru. Řídicí modul je vybaven čidlem otáček motoru na bázi odrazivé infrazávory, analogovou částí pro měření zpětně indukovaného napětí motorem a přijímačem v infračerveném spektru snímající kódy záhytných bodů.

Stávající řešení je postaveno na 8 bitové platformě RISC AVR, konkrétně na čipech řady ATMega88/168. Tato platforma poskytuje při taktovací frekvenci 20 MHz výkon 20 MIPS. Do současné doby bylo implementováno a otestováno řešení, které zvládá řízení motoru, měření jeho otáček IR závorou a měřením zpětně indukovaného napětí motoru. Dále bylo odzkoušeno snímání a dekódování záhytných bodů infrastruktury a komunikace s testovacím modulem Bluetooth. Motivací pro přechod na výkonější platformu je integrace zvukového modulu, která by vytvořila kompletní náhradu komerčně prodávaných řešení se splněním přidaných požadavků, které komerční řídící moduly nepokrývají. Analýza požadavků a porovnání dostupných mikroprocesorů je jednou ze zásadních částí této práce. Blokové schema modulu je na obr. 1.1.



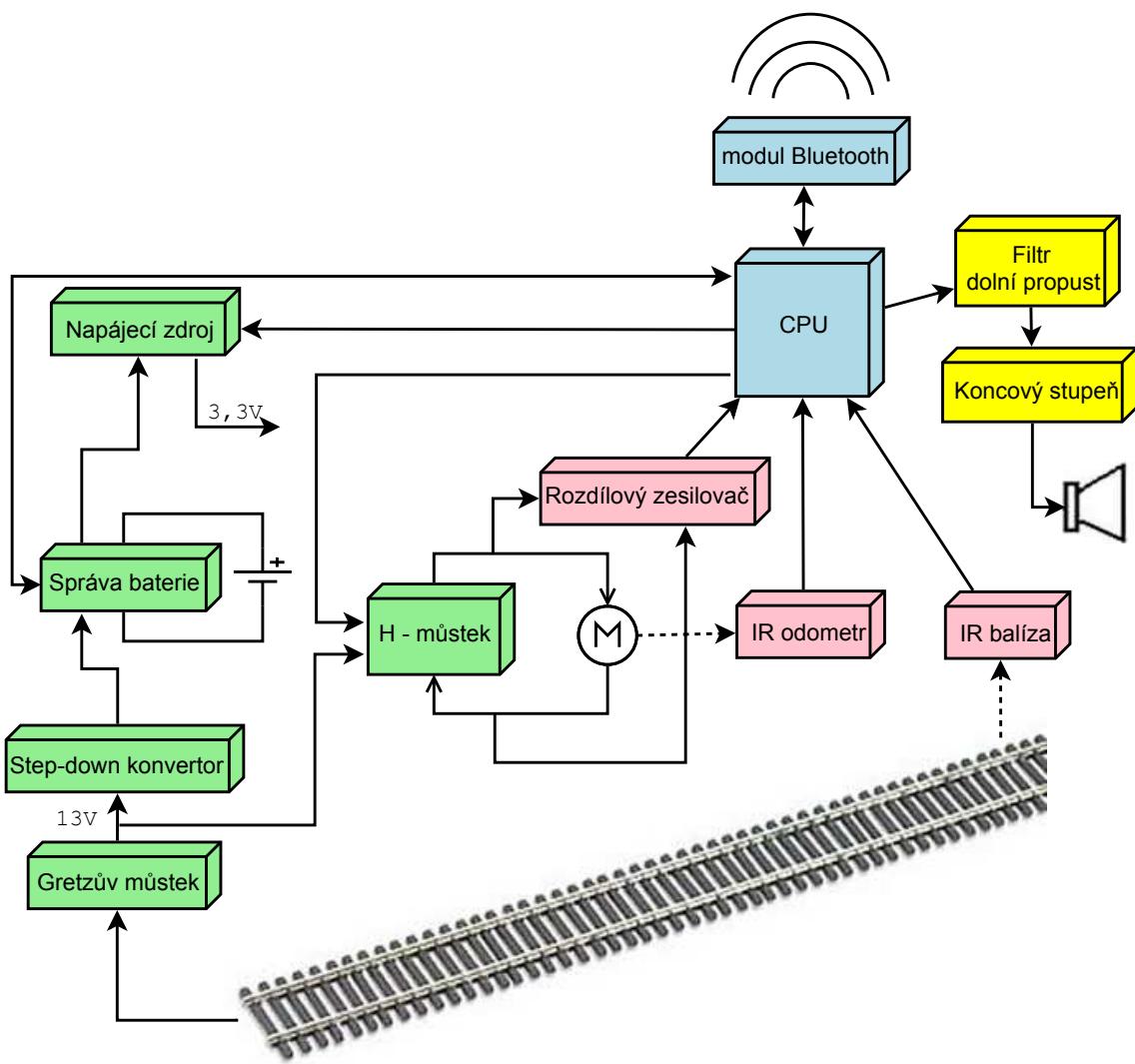
Obrázek 1.1: Blokové schema stávajícího řídicího modulu

## 1.2 Návrh finálního řešení

Řídicí modul je třeba dodatečně vybavit periferiemi, které dovolí integraci zvukového modulu. Jedná se o připojení externí paměti, která poskytne dostatek místa pro zvuková data za přiměřenou cenu. Nejvhodnějším řešením se ukazuje paměť flash s rozhraním SPI. Dále je třeba generovat audio signál pro reproduktor. Generování signálu je možné realizovat D/A převodníkem, ale mikroprocesory s integrovaným D/A převodníkem jsou poměrně drahé a příliš robustní pro cílovou aplikaci, takže je možné použít buďto malý externí převodník, nebo použít PWM mikroprocesoru pro generovaní signálu, který po externí filtraci bude zastupovat funkci D/A převodníku.

Vzhledem k dvojímu způsobu měření rychlosti motoru bude zvážena možnost odstranění snímače rychlosti motoru na infrazávory. To by umožnilo zjednodušení zástavby modulu do lokomotiv, protože by již nebylo třeba značkovat hřídel či setrvačník motoru a následně upevňovat infračervené čidlo odrazu. Odstranění tohoto čidla je však podmíněno otestováním dostatečné přesnosti odometrie získávané integrací, respektive sumací zpětně indukovaného napětí za známý časový úsek.

Další podstatnou změnou v řešení je použití dobíjecího akumulátoru nebo tzv. superkapacitoru místo stávajícího elektrolytického kondenzátoru. Ten vzhledem k poměrně enormní kapacitě a tedy i rozdílu mezi jeho a akumulátorem zabíral nezanedbatelný prostor a při zapnutí napájení způsoboval nepřijatelnou proudovou špičku. Kapacitor slouží pro pokrytí výpadků napájení, které jsou primárně způsobeny nečistotami na kolejnicích. Vytrvalejší zdroj energie s menším rozdílem umožní zástavbu reproduktoru a dále umožní zachovat komunikaci při delším výpadku napájení a v případě uvíznutí může nahlásit tento stav nadřízenému systému. Řešení by mělo také umožnit bezpečné zachování komunikace během průjezdu obratištěm, při kterém je třeba změnit polaritu napájení kolejnic před výjezdem z úseku smyčky obratiště, protože by jinak docházelo ke zkratu při průjezdu úseku na rozhraní kolejí s opačnou polaritou kolejnic. Blokové schema navrhovaného řešení řídicího modulu je uvedeno na obr. 1.2.



Obrázek 1.2: Blokové schema navrhovaného přepracovaného řídicího modulu



# Kapitola 2

## Řešení migrace na novou platformu

### 2.1 Analýza požadavků na novou platformu

Výběr nové platformy je třeba podložit analýzou současných požadavků. Požadavky jsou vyspecifikovány v následující podkap. 2.1.1. Pro splnění těchto požadavků je třeba definovat hodnotící kritéria, která budou schopna dát informaci o vhodnosti potenciálně zvažované nové platformě a pomohou tak objektivně rozhodnout mezi kandidáty.

#### 2.1.1 Definice hodnotících kritérií

Pro objektivní zhodnocení možností a výběr vhodné mikroprocesorové platformy je třeba definovat měřítka, kterými budeme hodnotit zvažované kandidáty. Základním a nezbytným požadavkem je výbava nutnými periferiemi pro realizaci funkčnosti požadované od řídicího modulu. Dalším měřítkem je fyzické pouzdro, ve kterém se daný čip vyrábí, protože to ovlivňuje použitelnost v omezeném prostoru lokomotivy a dále také ovlivňuje cenu tištěného spoje, protože je společně s modulem Bluetooth nejsložitějším obvodem na řídicím modulu a klade tím nejzásadnější požadavky na třídu přesnosti a nutnou technologii osazení tištěného spoje. Posledním a také velmi důležitým měřítkem je softwarová podpora produktu. Dnes vyráběné mikroprocesory jsou cenově a výkonově

velmi široce škálované a je tedy možno vybrat od různých výrobců několik velmi podobně vybavených čipů ve srovnatelné cenové relaci, ale je třeba také přihlédnout k dostupnosti kvalitního a cenově přijatelného vývojového prostředí.

Kromě těchto primárních kritérií je ještě vhodné přihlédnout k jistým sekundárním parametrům. Tyto parametry jsou sekundární, protože cílová aplikace buďto neklade zásadní nároky na zmíněné parametry, nebo jsou zmíněné parametry velmi vyrovnané u všech zvažovaných kandidátů a proto na ně není třeba klást důraz.

Definuji konkrétní požadavky na cílovou platformu.

- primární požadavky
  - dostatečný výpočetní výkon pro realizaci zvukového modelu
    - \* výpočetní výkon v celočíselné aritmetice
    - \* výpočetní výkon v plovoucí řádové čárce
  - požadované periferie
    - \* sériové RS-232 roshraní (podpora RTS/CTS řízení toku výhodou)
    - \* hardwarové PWM moduly s dostatečnou škálovatelností periody a strídy pro řízení motorového budiče
    - \* A/D převodník se čtyřmi multiplexovanými kanály
    - \* SPI/I2C rozhraní pro připojení externí paměti pro zvukový modul
    - \* dva časovače pro potřeby časování výpočtu regulační smyčky, dekódování signálu ze záhytných bodů kolejisti, časování odběru vzorků z A/D převodníku a dohled nad překročením časových limitů (ztráta komunikace, napájení, atd.)
    - \* dva nebo více volných vstupů s možností vyvolání externího přerušení (IR snímač otáček, detektor záhytných bodů kolejisti)
    - \* šest nebo více volných vstupně výstupních pinů pro obecné použití (světla, detekce polarity, reset Bluetooth modulu, ovládaní napájecího obvodu, apod.)
  - pouzdro čipu
    - \* velikost pouzdra
    - \* rozteč pinů pouzdra
  - vývojové prostředí

- \* vhodné debugovací rozhraní pro testování a ladění kódu (JTAG, Trace, debugWire)
  - \* použitelné vývojové prostředí dodávané k čipu, nebo prostředí podporující čip za dostupnou cenu
  - \* přehledná dokumentace, okomentované ukázkové projekty pro rychlé orientování a následnou migraci kódů
- sekundární požadavky
    - dostupnost čipu v maloobchodní síti
    - cena čipu
    - velikost programové paměti
    - referenční napětí pro A/D převodník
    - spotřeba jádra v aktivním stavu
    - přímý přístup do paměti - DMA

### 2.1.2 Porovnání sledovaných parametrů

Jako výkonější náhradu mikroprocesorové platformy jsem po průzkumu možností vybral tři vážné kandidáty. Jejich výběr byl ovlivněn dostupností vývojových kitů, které jsem dostal zapůjčeny od mého zadavatele projektu a mohl jsem se s nimi tedy blíže seznámit.

- ARM7TDMI - velmi rozšířené jádro 32 bitového mikroprocesoru RISC s Von Neumanovou architekturou, které je nabízeno mnoha výrobci. Jádro je na trhu více než deset let a vybudovalo si za tuto dobu rozsáhlou podporu komerčních výrobců i otevřené (tzv. open source) komunity
- Cortex-M3 - jedná se o pokročilou implementaci 32 bitového RISC jádra ARM7v s Harvardskou architekturou, která se prosazuje na poli levných a relativně výkonních vestavěných aplikacích
- TI TMS320 C28x - 32 bitové jádro s Harvardskou architekturou. Je nabízeno v malých pouzdrech s dostatečnou vybaveností periferiemi, má jádro častečně specializované pro DSP aplikace, ale oproti předchozím čipům nemá implementován přímý přístup do paměti, tzv. DMA.

Přehled sledovaných parametrů je přehledně zpracován do tab. 2.1. Tabulka maximálně pokrývá sledované parametry, ale některé detaily postihnout nemůže. Je vidět, že parametry jsou velmi vyrovnané, důvodem je velká škála verzí jednotlivých čipů. Ve srovnání jsou tedy vybrány čipy v malých pouzdrech s dostatečnou vybaveností.

### 2.1.3 Nároky na výpočetní výkon

Nezbytně nutným krokem pro správnou volbu platformy je také alespoň přibližná analýza výkonové náročnosti vykonávaného kódu. Na platformě AVR se dalo pro měření časové náročnosti částí kódu využít softwarového simulátoru AVR, který je součástí AVR studia. Simulátor zobrazuje přesný počet potřebných taktů hodin mikroprocesoru během vykonávaného kódu v jazyce C, respektive přeloženému ekvivalentu tohoto kódu do assembleru.

Pro zvažované cílové platformy takový nástroj bohužel neexistuje a proto jsem se musel uchýlit k jiné metodě měření. Je několik možností, jak odhadnout, nebo přímo experimentálně změřit čas nutný pro vykonání části kódu programu.

- teoretický výpočet pomocí analýzy přeloženého kódu do assembleru a vyhodnocení počtu a délky instrukcí
- měřením na fyzickém hardwaru
  - pomocí některého z interních časovačů
  - pomocí signalizace na některém externím pinu

Teoretický výpočet analýzy kódu v disassembly je možný, ale velice náročný, protože pro přesný výpočet by bylo nutné odhalit cykly v assembleru, vyhodnotit použité instrukce a jejich délku v hodinových cyklech, nebo alespoň odhadnout délku pomocí statistického průměru CPI (Cycle Per Instruction). Tento postup by zasloužil samostatný interpreter konkrétního assembleru kódu dle instrukční sady příslušné platformy, ruční zpracování by bylo velmi časově náročné a proto jsem se této metodě vyhnul.

Měření na fyzickém hardwaru jsem provedl pouze na procesoru s jádrem ARM7TDMI. Důvodem byla nejsnažší dostupnost tohoto procesoru. Udávaný výkon ostatních platform v jednotkách MIPS (počet instrukcí za vteřinu) srovnatelný, nebo mírně vyšší a jádro ARM7TDMI je podle dosavadních analýz možné brát jako nejméně výkonné,

|                      | ARM7TDMI<br>AT91SAM7S                            | Cortex-M3<br>LM3S2276                            | TI TMS320C2027           |
|----------------------|--|--|--------------------------|
| primární požadavky   |  |  |                          |
| Výp. výkon           | 0,9 MIPS/MHz                                     | 1,25 MIPS/MHz                                    | 1 MIPS/MHz               |
| RS-233               | ano (RTS/CTS)                                    | ano  | ano                      |
| PWM mod.             | ano (4x)   | ano (4x)   | ano (4x)                 |
| PWM frekv.           | clk/ $2^n$ /1-255                                | clk/ $2^n$                                       | clk/ $2^n$               |
| A/D kanály           | 8 (10 bit)                                       | 6 (10 bit)                                       | 16 (12 bit)              |
| SPI/I2C              | ano/ano  | ano/ano  | ano/ano                  |
| časovače             | ano (3+1)  | ano (3+1)  | ano (3)                  |
| ext. přerušení       | ano (2+)   | ano (2+)   | ano (2+)                 |
| I/O porty            | ano (5+)   | ano (5+)   | ano (5+)                 |
| pouzdro              | 64/48 pin LQFP                                   | 64 pin LQFP                                      | 48 pin PQFP/38 pin PSOP  |
| rozteč pinů          | 20 mil (0,5 mm)                                  | 20 mil (0,5 mm)                                  | 22 mil (0,56 mm)         |
| debug rozhr.         | JTAG/DBGU  | JTAG   | JTAG                     |
| IDE                  | Eclipse/CrossStudio<br>(IAR, Keil, CodeSourcery) | Eclipse/CrossStudio<br>(IAR, Keil, CodeSourcery) | Code Composer            |
| sekundární požadavky |  |  |                          |
| dostupnost           | GME, DigiKey, Farnell                            | DigiKey, Farnell                                 | verze v daném pouzdře ne |
| cena                 | 7\$  | 8,2\$  | -                        |
| spotřeba             | 33,3mA @ 55MHz                                   | 9,5mA @ 50MHz                                    | 70mA @ 40MHz             |
| A/D ref.             | 3V/ext.(max 3,3V)                                | 3V   | 3V                       |

Tabulka 2.1: Porovnání sledovaných parametrů mikroprocesorů

protože nemá oproti konkurentům hardwareovou děličku a vykazuje nejdelší latenci přerušovacího systému. Výkonostní analýza na jádru ARM7TDMI tedy poskytuje horní odhad časových intervalů.

Vydal jsem se pokud možno nejsnažší cestou a prováděl jsem měření času pomocí signalizace začátku a konce výpočtu bloku kódu pomocí externího pinu procesoru a tento signál jsem sledoval a měřil na osciloskopu. V cílové aplikaci nastávají některé události pravidelně, protože jsou cíleně vykonávány v pravidelných časových intervalech. Některé události jako například čítání pulzů od IR odometru nastávají neperiodicky a nelze je předvídat. V tab. 2.2 je uveden výčet nejčastěji prováděných operací s jejich intervalem opakování za jednotku času, respektive s jejich mezním minimálním intervalem mezi opakováním.

| operace                           | perioda opakování | priorita |
|-----------------------------------|-------------------|----------|
| měření zpětně generovaného napětí | 1 ms              | střední  |
| komunikace sériové linky          | 100 ms            | nízká    |
| výpočet akčního zásahu regulace   | 50 ms             | střední  |
| sledování signálu IR balíz        | 24,3 ms           | střední  |
| směšování zvuku na výstup         | 90,7 $\mu$ s      | vysoká   |
| čítání odometru                   | min 363 $\mu$ s   | střední  |

Tabulka 2.2: Periodicky volané bloky kódu

Všechny tyto předvídatelné procesy jsou řízeny časovači a vyvolávají přerušení běhu programu. Jejich prioritu lze zohlednit prioritou přerušení, kterou lze u všech procesorů programově nastavovat. Pro odhad řešitelnosti výpočetních problémů s dostupným výpočetním výkonem v čas je třeba změřit časové intervaly potřebné pro zpracování jednotlivých požadavků. Pro procesor AT91SAM7S jsou změřené časy potřebné pro vykonání jednotlivých operací uvedeny v tab. 2.3

Zatím jsem nebyl nucen využívat žádný pokročilejší nástroj pro plánování procesů, protože program bez problémů pracoval v jednom vlákně, kdy je vykonávána nekonečná smyčka přerušovaná požadavky od jednotlivých periferií. Na nové platformě je v oblasti plánování možné lépe rozvrhovat řešení úkolů, protože je možné uživatelsky nastavovat priority obsluh jednotlivých přerušení. Metoda, která se specializuje na rozvrhování více úkolů, které požadujeme vykonávat v reálném čase je tzv. "rate-monotonic" plánování, kdy všechny vlákna programu mají statickou prioritu pro vykonávání. Toto plánování má následující předpoklady:

- nejsou sdíleny společné prostředky
- nejzašší doba dokončení úkolů je shodná s periodou, přes kterou plánování analyzuje
- statické priority úkolů, úkol s nejvyšší prioritou je vykonán přednostně před všemi ostatními
- priority jsou přiděleny podle ”rate-monotonic” konvence (úkoly s kratší periodou/bližší deadline mají vyšší prioritu)
- čas potřebný pro tzv. přepnutí kontextu (přechod mezi řešením více úkolů) je v ideálním případě nulový, v praxi zanedbatelný oproti dálce vykonávaného kódu před dalším přepnutím kontextu

Pokud jsme schopni zaručit tyto předpoklady, tak matematický model této situace je řešitelný, pokud platí Liu-Leylandova teorém. Ten říká, že množina  $n$  periodicky se opakujících úkolů s různou délkou periody jejich opakování je řešitelná, pokud vždy platí

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq n \left( \sqrt[n]{2} - 1 \right),$$

kde  $C_i$  je čas potřebný pro řešení úkolu a  $T_i$  je perioda opakování tohoto úkolu. Je možné odvodit, že v limitním případě, kdy se počet procesů  $n$  blíží nekonečnu, je nutné, aby zatížení procesoru  $U \leq \ln 2 \approx 0,693$ . Je tedy možné obsloužit v čas všechny úkoly, pokud bude zatížení procesoru pod hranicí cca 69,3%. Zbylých cca 30% výpočetního času lze použít na časově nekritické úkoly, které nevyžadují splnění do přesného deadline (nejzašší čas pro dokončení operace).

V mé aplikaci lze dané předpoklady splnit. Program až na výjimky nesdílí prostředky, pouze v některých případech využívá některé společné proměnné, které si ale jednotlivé obsluhy přerušení vždy pouze čtou do své lokální kopie, aby byla hodnota proměnné zafixována do konce obsluhy přerušení. Rozdělení priorit je možné dodržet pomocí nastavení priorit obslužných rutin přerušení. Přepínání kontextu je v porovnání s dobou vykonávání kódu zanedbatelné, protože dokumentace garantuje obsluhu standardního přerušení (IRQ) za 42 hodinových cyklů a rychlého přerušení (FIQ) za 18 hodinových cyklů. To na cílové frekvenci 52 MHz představuje pouze  $0,808 \mu\text{s}$ , takže přepnutí kontextu je nezanedbatelné pouze u operace čítání pulzů z odometru, tam je četnost omezena maximálními otáčkami motoru a počtem značek na setrvačníku motoru a po vyčíslení se jedná o  $363\mu\text{s}$ .

| operace                           | doba potřebná pro obsluhu         |
|-----------------------------------|-----------------------------------|
| měření zpětně generovaného napětí | $36,6 \mu s$                      |
| komunikace sériové linky          | 3,7 ms (aktivní čekání)           |
| výpočet akčního zásahu regulace   | $120,4 \mu s$                     |
| směšování zvuku na výstup         | $14,2 \mu s$                      |
| sledování signálu IR balíz        | 1,28 ms                           |
| čítání odometru                   | $850 \text{ ns} + 870 \text{ ns}$ |

Tabulka 2.3: Čas potřebný pro vykonání operace

Provedu výpočet pro mojí úlohu plánování, kdy musí platit

$$U = \frac{36,6}{1000} + \frac{3,7}{100} + \frac{120,4}{50000} + \frac{1,28}{24,3} + \frac{14,2}{86,8} + \frac{1,72}{363} \leq 6 \left( \sqrt[6]{2} - 1 \right)$$

$$U = 0,2970 \leq 0,7348$$

Základní předpoklad řešitelnosti úloh v požadovaném čase je tedy splněn. Zbývající čas je k dispozici pro obsluhu časově nekritických úloh a plnění zvukové vyrovnávací paměti.

#### 2.1.4 Diskuze nad porovnáním kandidátů na novou platformu

Pokud vyhodnotím porovnání paramterů v tab. 2.1, tak v primárních parametrech jsou rozdíly pouze v hardwarovém řízení toku dat pro rozhraní RS-323, škálovatelnosti frekvence modulace PWM modulů a dostupném vývojové prostředí. Sekundární parametry se liší více. Pokud přihlédnu ke spotřebě, dostupnosti a vývojovému prostředí čipů série TMS320C2000, tak tuto možnost pokládám za nejméně vhodnou. Ačkoliv je tento čip je částečně specializován pro DSP aplikace a disponuje jednocyklovou instrukcí MAC (Multiply and Accumulate), tak jeho dostupnost v použitelném pouzdře, spotřeba a v neposlední řadě slabá podpora zastaralým vývojovým prostředím ho jednoznačně vyřazují z výběru.

Seriozní volbu jsem zúžil na výběr mezi jádrem ARM7TDMI a Cortex-M3. Toto dilema je velmi často diskutováno, protože výběr mezi jádrem ARM7 a Cortex-M3 řešilo už monoho vývojářů. Srovnání těchto jader je rozebráno v mnoha článcích a kromě

specificky definovaných kritérií pro mojí konkrétní aplikaci jsem se snažil nastudovat problematiku i obecněji, z pohledu celkové architektury jader. Velmi dobrým zdrojem je článek (LUNDGREN, A., 2009), jehož autor je dlouhodobým zaměstnancem firmy IAR, která vyvinula komerčně používané vývojové prostředí IAR Em-bedded workbench for ARM. Ve zkratce je zde vyzdvižen nejzásadnější rozdíl v podobě řadiče přerušení, tzv. NVIC - Nested Vector Interrupt Controller, který je na platformě Cortex-M3 integrovaný přímo v jádře a výrazně zkracuje dobu přerušení z 48 hodinových cyklů na 12. Dále tento integrovaný řadič přerušení harwarově řeší uložení kontextu, není třeba tuto proceduru provádět ručně, respektive direktivami pro překladač. Dále je zde diskutována cena čipů, která je v současné době vyrovnaná a podpora vývojových prostředí, která je v současné době také vyrovnaná. V případě použití čipu od Luminary Micro je přínosnou vlastností bezplatná softwarová podpora v podobě knihovny StellarisWare, která výrazně zjednodušuje práci s periferiemi a hlavně umožňuje snadnou migrovatelnost kódu mezi podobnými čipy, závisí pouze na přítomnosti používaných periferií na cílovém čipu migrace.

Ačkoliv je Cortex-M3 novějším, výkonějším, a v jistých ohledech pokročilejším jádrem oproti jádru ARM7TDMI, tak pro cílovou aplikaci nepředstavuje zásadně lepsí řešení oproti jádru ARM7TDMI, respektive jeho implementace čipem AT91SAM7S. Oba čipy poskytují dostatečný výpočetní výkon pro cílové nasazení. ARM7TDMI poskytuje oproti Cortex-M3 jádru řízení toku dat RTS/CTS a velmi důležitou přesnost nastavení frekvence modulů PWM. Jeho vyšší spořeba je snesitelnou cenou za zmíněné přednosti. Je třeba brát ohled na fakt, že porovnávaná výrobcem udávaná spotřeba je mezní (maximální) a to ve stavu, kdy jsou všechny periferie na čipu zapnuty, takže lze předpokládat, že spotřeba při reálném nasazení bude o něco nižší, než uvedená mezní hodnota.

## 2.2 Migrace funkčního kódu na novou platformu

Program vykonávaný mikrokontrolerem je od začátku vývoje psán v jazyce C je tedy částečně přenositelný na jiné platformy, pro které je dostupný překladač z jazyka C do instrukcí assembleru cílové platformy. Změnu zdrojového programu vyžadují části, které se provádí inicializace periferií a obsluhy přerušení. Vlastní inicializace periferií musela být kompletně přepracována. Na platformě AVR, respektive v prostředí AVR

Studia s vestavěným překladačem WinAVR se obsluha přerušení uvozovala klíčovým slovem `signal(interrupt_id)`, kde proměnná `interrupt_id` značí zdroj přerušení (seznam konstant označující zdroje přerušení je uveden v dokumentaci pro knihovnu AVR-libc). V prostředí Rowley CrossStudio for ARM se obslužné rutiny píší jako standardní funkce bez vstupních a výstupních proměnných, které se při inicializaci registrují v příslušném registru procesoru. Registraci, tedy zápis adresy obslužné funkce, který je volán při příslušném přerušení lze provádět ručně, nebo pohodlněji pomocí funkce knihovny CTL, která je součástí vývojového prostředí.

### 2.2.1 Ověření splnění původní funkčnosti

Jak již bylo zmíněno, hlavní prací při migraci na platformu ARM7 byla nutnost úpravy kódů, který inicializoval periferie. Po úspěšném zvládnutí úpravy konfigurace nových periferií jsem ověřil funkčnost nejdříve laboratorně s vývojovou deskou AT91SAM7X256-EK od firmy Atmel. K vývojové desce byl připojeny ovládané periferie pro ověření kompatibility řešení. Testy byly nutné hlavně z důvodu přechodu na 3,3 V logiku. Konkrétně se jednalo o spolehlivost ovládání budícího H můstku Si9986, který je určen pro ovládání pomocí TTL logiky. Dále bylo nutné otestovat nově použitý IR snímač GP1UE26RK0VF s tvarovačem. Ten, narozdíl od dříve používaného typu SFH5110, lze napájet napětím 3,3 V. Ukázalo se však, že při sníženém napájení na úroveň 3,3 V je snížena i citlivost snímače, který následně vyžadoval delší "burst" pro uvedení výstupu do úrovně logické nuly. To v praxi znamenalo, že došlo ke zkreslení signálu ve smyslu asymetrie doby trvání logických urovní. Bylo proto provedeno prodloužení doby vysílání logických úrovní z 16 na 20 impulzů na bit, to znamená při použité modulační frekvenci 36 kHz prodloužení z 0,444 ms na 0,555 ms. Toto opatření pomohlo eliminovat asymetrii na zanedbatelnou úroveň a nebylo tak třeba komplikovat příslušný software nutností reakce na různou dobu trvání log. úrovní. Následovaly experimenty testující realizovatelnost dodatečných funkčností a výroba prototypového tištěného spoje pro jízdní testy přímo na lokomotivě.

# Kapitola 3

## Návrh a řešení zvukového modulu

Pro generování akustického signálu bylo třeba vyřešit, jakým způsobem bude signál fysicky generován, jakým způsobem bude navázán na reproduktor a fyzicky jaký reproduktor je možné zastavět do lokomotivy.

### 3.1 Hardwarové řešení zvukového modulu

Pro generování signálu z procesoru, který nemá vlastní DA převodník, je možné použít externí DA převodník připojený na některé standardní komunikační rozhraní. Tyto DA převodníky jsou nabízeny často s rozhraním I<sup>2</sup>C, popřípadě s jeho mutací I<sup>2</sup>S. V návrhu bylo zvažováno použití obvodu CS4334 od firmy Cirrus Logic. Ten však vyžaduje přivádění poměrně rychlých externích hodin v řádech MHz a pro cílové použití má zbytečně velkou vzorkovací frekvenci zdrojového signálu.

Další řešení mohl poskytnout integrovaný obvod MAX9477 od firmy Maxim, který lze plnit daty pomocí I<sup>2</sup>S rozhraní, disponuje integrovaným koncovým stupněm. Tento obvod se však ukázal jako příliš robustní, protože měl poměrně velké pouzdro a byl koncipován na výrazně vyšší výstupní výkon než bylo třeba.

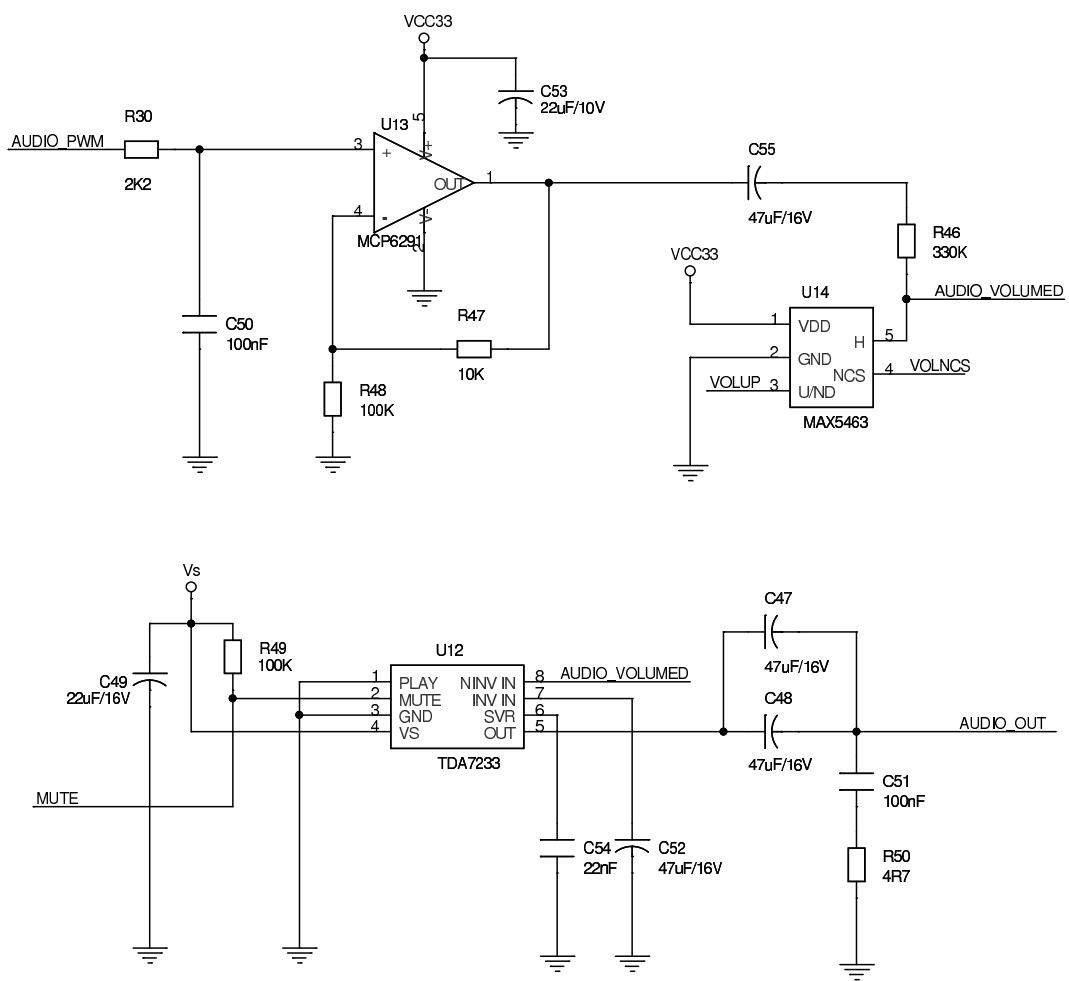
Nakonec bylo rozhodnuto použít techniku metodu filtrovaného PWM signálu pomocí dolní propusti. Tato technika je poměrně často využívána a firma Texas Instruments jí věnuje samostatnou aplikační nótu (MITCHELL, M., 2000). Metoda je však obecně použitelná na libovolném mikroprocesoru s dostatečně rychlým PWM výstupem. V prin-

cípu se jedná a využití modulovaného PWM signálu o několikanásobně vyšší frekvenci, než je vzorkovací frekvence výstupního audio signálu. Tento signál je následně pasivně filtrován pomocí RC článku. Výstupní hodnota je pak učena střídou PWM signálu. Tato metoda s použitím vybraného procesoru AT91SAM7S umožňuje zvolit v rámci běžných standardů libovolnou vzorkovací frekvenci výstupního audio signálu a také volbu rozlišení až 16 bitů. Vzhledem ke kvalitě dostupných zvukových dat, nutné kvalitě audio signálu, paměťové náročnosti a použitelné datové reprezentace byla zvolena vzorkovací frekvence 8 000 Hz s rozlišením 8 bit.

Signál je následně přiváděn do zesilovače, který dovoluje větší zatížitelnost výstupu, než samotný procesor. Protože koncové stupně mají poměrně velkou vstupní citlivost, je zesílení nastaveno na hodnotu 1/3. Pro regulaci hlasitosti je použit digitální potenciometr MAX5463 od firmy Maxim v miniaturním pouzdře SC70. Ten je ovládán dvouvodičově pomocí signálů negated chip select, dále jen NCS a up/negated down, dále jen U/ND. Digitální potenciometr inkrementálně posouvá jezdec přivedením sestupné hrany signálu U/ND s aktivovaným signálem NCS. Směr posuvu jezdce je dán úrovní signálu U/ND v době, kdy byl aktivován obvod signálem NCS. Podrobná funkce obvodu je popsána v jeho dokumentaci (MAXIM, 2005).

Jako koncový stupeň byly zvažovány různé obvody z řady TDA. Nakonec bylo rozhodnuto o použití obvodu TDA7233. Jedná se o jednokanálový (mono) zesilovač třídy AB s jmenovitým výstupním výkonem 1 W (při napájecím napětí 12 V a zátěží o jmenovitému odporu  $8 \Omega$  činí výstupní výkon až 1,9 W). Obvod má vstup MUTE pro absolutní ztlumení výstupního signálu. Při aktivaci tohoto vstupu se obvod zároveň uvede do režimu nízké spotřeby energie. Další informace je možné získat z dokumentace (ST MICROELECTRONIC, 2005).

Výsledné schema zapojení obvodu je uvedeno na obr. 3.1. Zapojení koncového stupně je dle doporučení výrobce. Zapojení bylo testováno s klasickým odporovým trimrem a vývojovou deskou. Testování digitálního potenciometru bylo prováděno až na prvním prototypu plošného spoje.



Obrázek 3.1: Schéma zapojení zvukové části modulu

## 3.2 Softwarové řešení zvukového modelu

Základ techniky modelování zvuku motoru v různých otáčkách byl čerpán z (HAJNÝ, M., 2006). Materiál se zabývá komplexním modelováním zvuku pro automobilový simulátor, tedy nejen modelováním zvuku motoru, ale i ruchů z vnějšího prostředí, jako je aerodynamický hluk, hluk od pneumatik, atd.. Modelování dieslového motoru lokomotivy je v principu shodná úloha jako modelování zvuku motoru automobilu. Pro svou práci jsem studoval především (HAJNÝ, M., 2006, kap. 7.4).

Druhým hlavním zdrojem poznatků mi byl reverse engineering konfiguračních souborů zvukových modelů železničního simulátoru Microsoft Train Simulator, dále jen MSTS. MSTS má poměrně silnou základnu příznivců, kteří vytvářejí 3D modely českých lokomotiv a vozů včetně jejich zvukových modelů. Po rozklíčování skriptů, které řídí přehrávaní jednotlivých zvuků během jízdy se ukázalo, že modelování zvuku motoru je velmi podobné s metodou v (HAJNÝ, M., 2006).

### 3.2.1 Modelování dle DP M. Hajného

Pro modelování zvuku motoru lze zvolit v zásadě dva přístupy:

- Syntéza zvuku z umělých signálů
- Syntéza zvuku skládáním nahraných reálných zvuků

Syntéza zvuku z umělých signálů by znamenala analýzu nejvýraznějších harmonických složek reálného signálu v celém spektru otáček motoru. Následně by bylo třeba approximovat funkce amplitud těchto harmonických složek v závislosti na otáčkoách motoru. Výsledný zvuk by se pak reprodukoval generováním a složením nejvýraznějších harmonických složek. Pro dosažení dostatečně věrného výsledku by bylo třeba použít poměrně velkého množství harmonických složek. Vzhledem k použití na jednočipovém mikroprocesoru nepřipadala z výkonového hlediska v úvahu.

Syntéza zvuku skládáním nahraných vzorků zvuků je oproti předchozímu řešení paměťové náročnější, ale výkonově méně náročná metoda. Syntéza zvuku motoru je prováděna skládáním zvukových smyček odpovídajících zvuku motoru v určitých diskrétních hodnotách otáček motoru. Pokud je dostatečné množství vzorků pro pokrytí celého spektra otáček, lze vzorky pouze míchat mezi sebou podle aktuálně požadovaných otáček motoru.

Funkce zesílení signálu, které lze vhodně použít pro překrývání zvukových smyček, jsou uvedeny v (HAJNÝ, M., 2006).

### 3.2.2 Modelování dle Microsoft Train Simulator

Zvukové modely v Microsoft Train Simulatoru (dále jen MSTS) jsou řešeny modifikací syntézy zvuku skládáním reálných zvuků z oddílu 3.2.1. Modifikace spočívá v použití chudší škály zvukových vzorků. Při malém počtu vzorků dochází k nespojitosti odezvy zvukového modelu na změnu otáček motoru, protože určitý rozsah otáček motoru je pokryt pouze jednou zvukovou smyčkou pro konstantní otáčky. Tento problém je řešen v MSTS buďto pomocí přehrávání vzorků různou rychlostí v rámci tohoto pásma otáček, nebo využitím předpokladu, že páka ovládající výkon motoru, respektive jeho otáčky, má jen omezené množství poloh, kterému je přizpůsoben počet vzorků modelu lokomotivy. Pokud je použito malé množství vzorků, tak je v rámci pásma otáček, kdy je využívána jediná zvuková smyčka, prováděno přehrávání smyčky rychleji, nebo pomaleji a je proto nutné provádět převzorkování signálu.

Chování zvukových modelů v MSTS je uloženo v souborech s příponou SMS (zkratka pro Sound Management System). Jedná se skripty s poměrně jednoduchou syntaxí. Každý soubor obsahuje hlavičku a dále je uvozen frází Tr\_SMS(). Uvnitř závorek jsou definice chování zvukového modelu, tzn. kdy a jak se mají pouštět jednotlivé zvuky náležící modelovanému vozidlu. Detailní popis významu skriptů jsem jsem získal od českého tvůrce zvukových modelu Petra Smutka, ale rozklíčování je možné najít i na internetu, viz. (HAGEN, R., 2002). Nejzajímavější částí byla část popisující chování motoru. Zde je uveden příklad skriptu, kde je definováno převzorkovaní signálu (rychlosť přehrávání) a zesílení signálu v závislosti na poloze řídící páky.

---

|                     |       |
|---------------------|-------|
| FrequencyCurve (    |       |
| Variable2Controlled |       |
| CurvePoints (       | 7     |
| 0.000               | 12025 |
| 0.010               | 12025 |
| 0.250               | 13000 |
| 0.300               | 13000 |
| 0.301               | 12025 |

```

0.350           12025
0.650           13000
)
Granularity ( 100 )
)

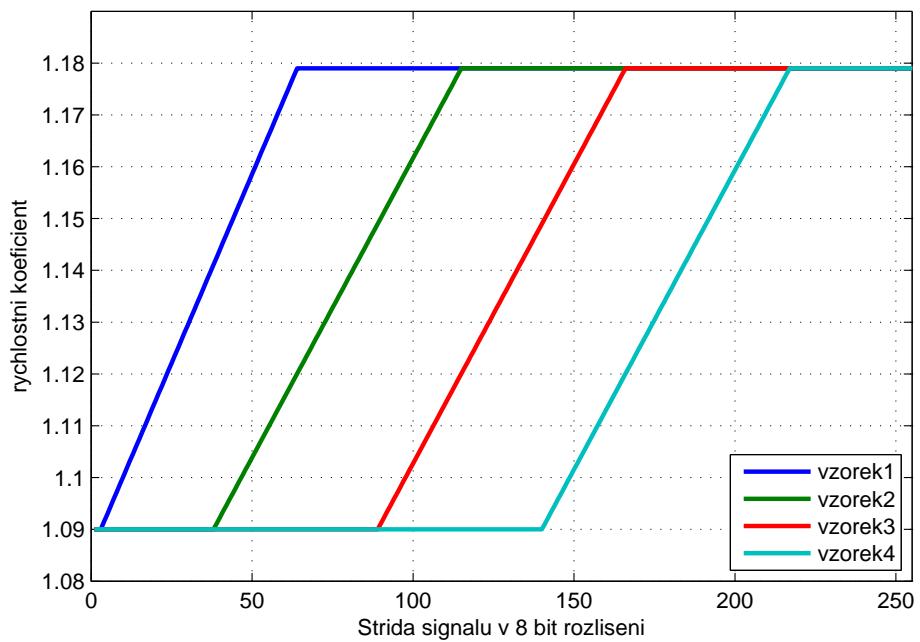
VolumeCurve(
    Variable2Controlled
    CurvePoints ( 12
        0.000           0.0
        0.010           0.3
        0.050           0.4
        0.150           0.45
        0.250           0.0
        0.350           0.0
        0.450           0.50
        0.550           0.60
        0.650           0.70
        0.750           0.80
        0.850           0.90
        0.950           1.0
    )
    Granularity ( 0.05 )
)

```

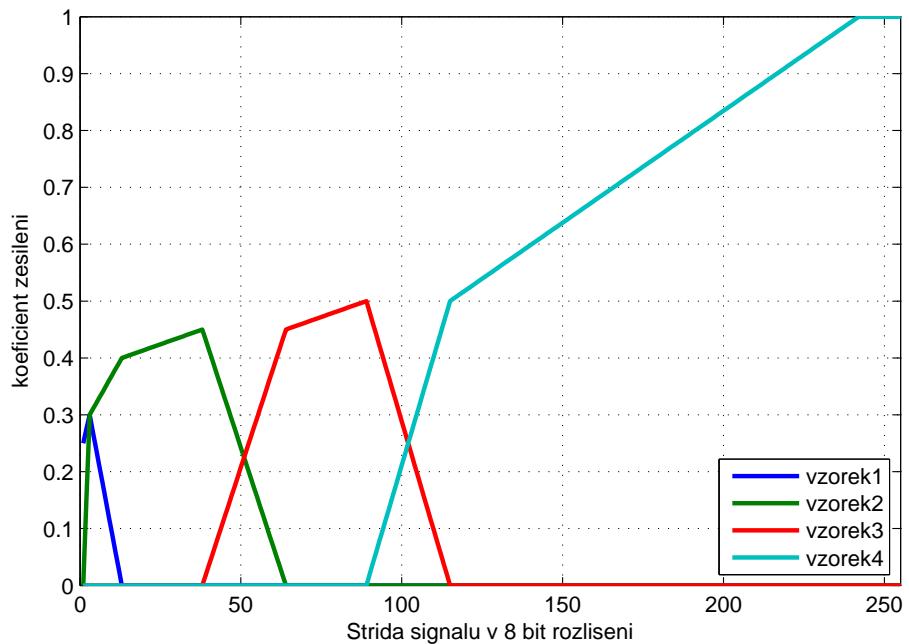
---

Pro lepší porozumění významu skriptu jsem křivky rekonstruoval v prostředí Matlab do grafu, viz. obr. 3.2 a obr. 3.3. Křivky rychlostního koeficientů ukazují lineární změnu rychlosti přehrávání vzorků v rozsahu, kdy je vzorek slyšitelný na výstupu. Křivky koeficientů zesílení naznačují, že první vzorek zvuku je používán převážně pouze pro simulaci volnoběhu motoru. Následující vzorky jsou využívány pro širší rozsahy otáček a speciálně poslední vzorek je využit v nejširším rozsahu, přičemž při přiblížování se k maximálnímu výkonu se narozdíl od přechozích vzorků zvyšuje hlasitost až k maximální úrovni.

Tento postup jsem reprodukoval pro cílové použití na mikroprocesoru, protože měl dostatečně nízkou výkonovou náročnost a zároveň přijatelnou paměťovou náročnost na množství zvukových dat.



Obrázek 3.2: Rekonstruované křivky rychlostního koeficientu přehrávaných vzorků zvuků



Obrázek 3.3: Rekonstruované křivky koeficientu zesílení přehrávaných vzorků zvuků

### 3.2.3 Modelování pro vlastní potřebu zvukového modulu

Pro testování zvukových modelů jsem používal nejdříve program Matlab a zvukové smyčky vytvořené lokální komunitou uživatelů MSTS. Tyto zvukové modely jsou velmi vhodné, protože jsou volně dostupné a mají rekonstruovat zvukové modely lokomotiv, které se pohybují na našich tratích, tedy především lokomotivy a motorová vozidla československé, případně ruské výroby.

Manipulace s rychlostí přehrávání zvukových smyček má jistá úskalí. Obecně lze měnit rychlosť v omezeném rozsahu. Při zrychlování zvuk degraduje, protože klesá počet vzorků na časový úsek, v extrémním případě až na mez, kdy se dostaneme do konfliktu se vzorkovacím teorém. Při zpomalování naopak houstne počet vzorků na časový úsek. To by teoreticky nemělo nijak negativně ovlivňovat kvalitu zvuku. Rozsah změny rychlosti přehrávání je však více omezen jevy, které souvisí s vnímáním zvuku posluchačem. Při změně rychlosti přehrávání smyčky dochází k posuvu frekvenčního spektra vzhůru, respektive níže, podle toho zda zrychlujeme či zpomalujeme přehrávání oproti originálu. Tento jev je pro posluchače zvláště patrný a nepříjemný, pokud takto manipulujieme s nahrávkou obsahující zpěv.

Změna rychlosti přehrávání zvukové smyčky uložené v digitální podobě přináší nutnost převzorkování signálu. Protože se rychlosť přehrávání mění řádově v jednotkách procent oproti originálu, je třeba při zachování konstantní rychlosťi vzorkování na výstupu dopočítávat hodnotu vzorků mezi původními známými hodnotami. Metod pro převzorkování je velké množství a především ty složitějsí, které se snaží zabránit výše uvedenému posuvu spektra, jsou často patentovány a střeženy svými vlastníky. Během testování v programu Matlab byla použita jednoduchá lineární interpolace vzorků. Touto metodou lze dle subjektivního poslechu měnit rychlosť přehrávané smyčky o  $\pm 12\%$  (při použité kvalitě vzorků dle kap. 3.1 a kvalitě reprodukce v cílovém zařízení). Na cílovém zařízení však bylo nutné se uchýlit k metodě prostého vybírání nejbližšího souseda.

## 3.3 Míchání více zdrojů signálu v digitální podobě

Při mísení zvukových signálu v otevřeném prostoru dochází ke sčítání tohoto vlnění. Pro smísení dvou, nebo více zvuků v digitální podobě není možné provést jednodu-

chou operací sčítání. Pokud jsou zvuková data v paměti mikroprocesoru uložena v rozlišení 8 bit, tedy v rozsahu 0 až 255, a stejně tak výstup z mikroprocesoru je D/A převodník se shodným rozlišením 8 bit, tak není možné tuto operaci provádět prostým součtem vzorků, protože výstup je shora omezen maximální hodnotou 255, respektive  $255 \cdot (V_{ref}/255)$ , kterou lze na D/A převodníku nastavit. Součet vzorků však produkuje hodnoty v rozsahu 0 až 510, které D/A převodník není schopen reprodukovat. Jako nejjednodušší řešení by se zdála normalizace zaznamenaných vzorků na poloviční hlasitost, tím bychom však redukovali kvalitu nahrávky, protože bychom stejně tak snížili dynamický rozsah na polovinu. Problém by se pak dále komplikoval, pokud bychom požadovali mísení obecně více než dvou signálů.

Pro vyřešení problému chceme realizovat algoritmus, který bude mísit signál A a B tak, že pokud bude signál B reprezentovat ticho, tak chceme na výstupu získat čistý signál A a naopak. Pokud budou oba signály nabývat nenulové amplitudy, chceme aby výsledkem operace byly hodnoty v rozsahu 0 až 255. Těmto požadavkům vyhovuje operace

$$Y = A + B - A \cdot B, \quad \text{kde} \quad A, B, Y \in \langle 0; 1 \rangle$$

respektive

$$Y = A + B - \frac{A \cdot B}{255}, \quad \text{kde} \quad A, B, Y \in \langle 0; 255 \rangle$$

Zvuková data jsou však uložena v lehce odlišné reprezentaci, kde ticho je reprezentováno hodnotou 128 a užitečný signál se mění v rozsahu 0 až 255. Tento algoritmus tedy nutné upravit do podoby

$$Y = 2 \cdot A \cdot B, \quad \text{kde} \quad A, B \in \langle 0; 0,5 \rangle, Y \in \langle 0; 255 \rangle$$

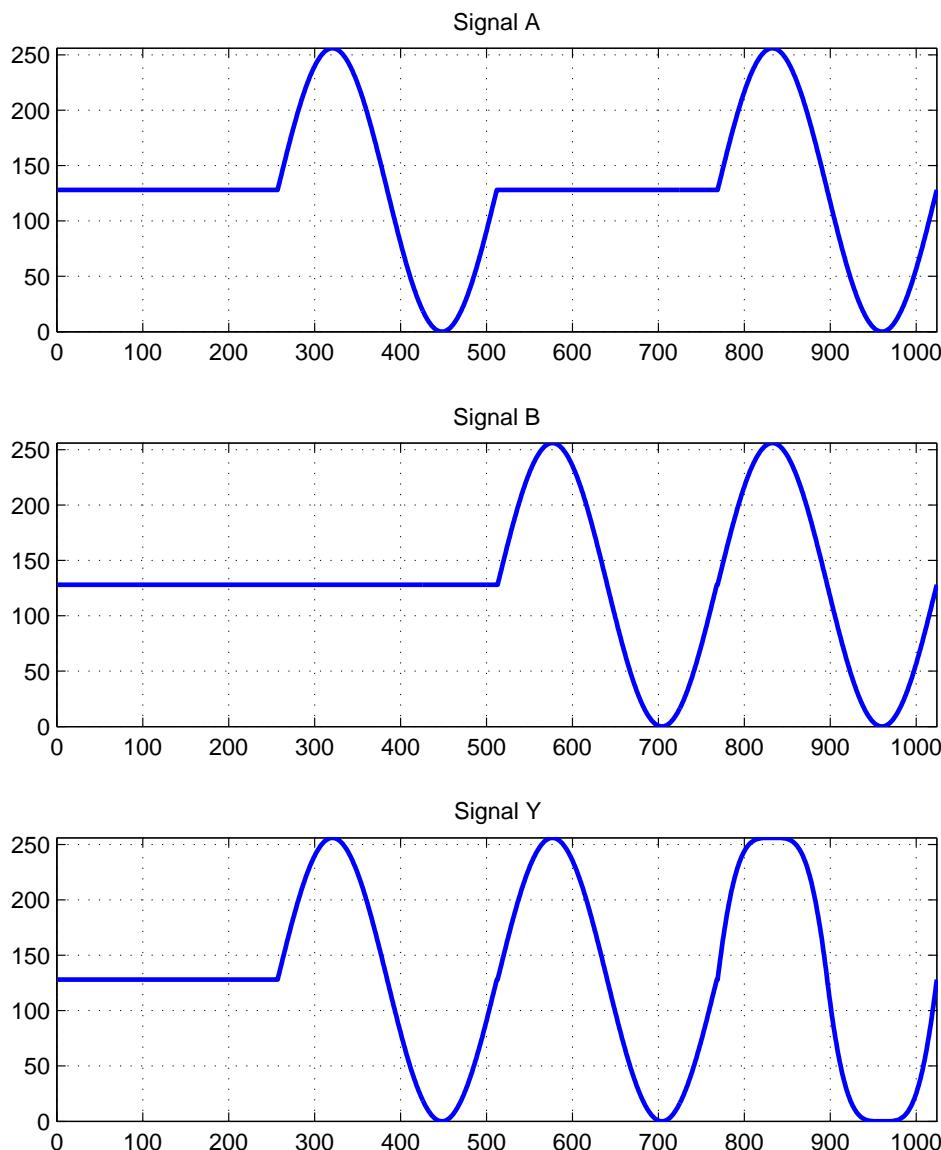
$$Y = 2 \cdot (A + B) - 2 \cdot A \cdot B - 1, \quad \text{kde} \quad A > 0.5 \text{ nebo } B > 0.5$$

respektive

$$Y = A \cdot B / 128, \quad \text{kde} \quad A, B \in \langle 0; 128 \rangle, Y \in \langle 0; 255 \rangle$$

$$Y = 2 \cdot (A + B) - A \cdot B / 128 - 256, \quad \text{kde} \quad A > 128 \text{ nebo } B > 128$$

Chování tohoto algoritmu je graficky znázorněno na obr. 3.4. Je vidět, že ticho v obou kanálech A i B generuje na výstupu také ticho. Pokud je na vstupu pouze jeden zdroj užitečného signálu, tak prochází nezměněn a pokud vstupují do výpočtu dva užitečné signály, tak dochází k dynamickému zatlumení tak, aby se signál udržel v přípustném rozsahu hodnot.



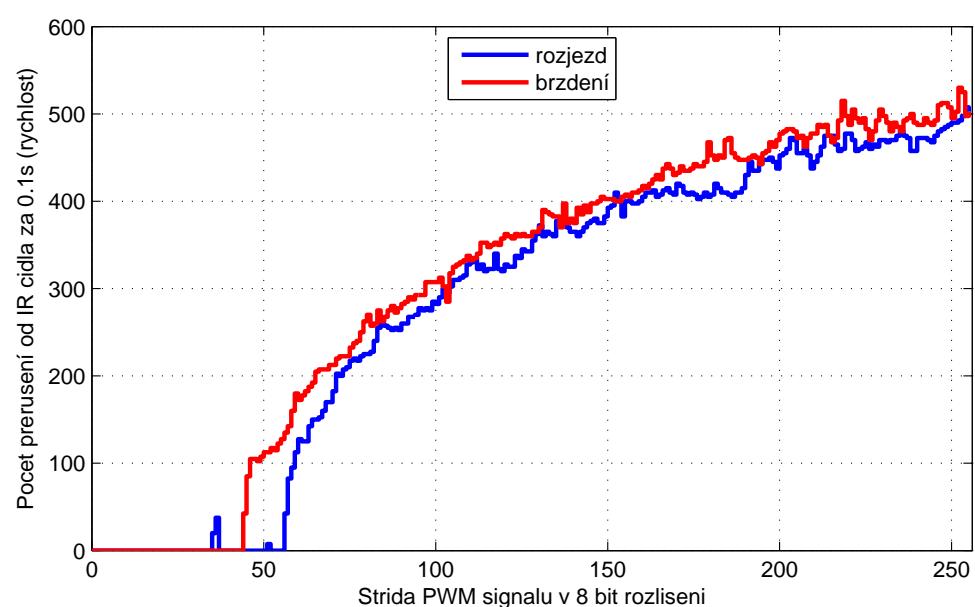
Obrázek 3.4: Simulace míchání dvou signálu v digitální podobě

### 3.4 Zvuková odezva emulace motoru na změnu rychlosti jízdy

V podkapitole 3.2.3 je popsán způsob, jakým je modelován zvuk motoru. Je třeba však zvolit veličinu, od které se bude odvozovat reprodukovaný zvuk. Vzhledem k tomu, že otáčky motoru úzce souvisí s požadovaným výkonem na motoru, měla by tedy být zvuková odezva odvozována od střídy PWM signálu přiváděného na motor. Tato vazba by zajistila, že i při jízdě konstantní rychlostí by při změně traťových podmínek (tzn. stoupání, klesání, nebo změna zapřažené soupravy a tím změněné zatížení lokomotivy) došlo k očekávané změně zvukového projevu. Při testu se však ukázalo, že tato jednoduchá vazba není možná, protože při rozjezdu byla odezva nepřirozená. Problémem je nelinearita v počátku převodní charakteristiky střídy PWM signálu na otáčky motoru (viz. obr. 3.5). To v znamená, že při pokusu o rozjezd, kdy je reprodukován zvuk volnoběžných otáček došlo k výrazné a nepřirozené skokové změně zvuku motoru.

Byla proto provedena úprava, kdy zvuková odezva motoru se odvodila od filtrované požadované rychlosti lokomotivy. Aby bylo docíleno požadovaného efektu změny odezvy při změně jízdních podmínek je prováděna korekce odezvy podle průměrované odchylky střídy PWM signálu od standardní charakteristiky.

Pokud bychom chtěli model ještě více přiblížit realitě, bylo by třeba zpozdit požadavky na rychlosť přiváděně do regulačního obvodu motoru tak, aby došlo k efektu předbuzení motoru před samotným rozjezdem. Tento postup však zatím nebyl vzhledem k testovacímu provozu implementován, protože by doslo k snížení ovladatelnosti soupravy především v případě nutnosti nouzového zastavení.



Obrázek 3.5: Převodní charakteristika střídy PWM signálu na rychlosť jízdy

# Kapitola 4

## Realizace nového řídicího modulu

Během migrace na novou platformu byla provedeny dvě, respektive tři iterace návrhového procesu. Tento proces se skládá z návrhu plošného spoje, zadáním do výroby, osazením, oživením elektroniky a následným testováním.

Během první iterace došlo k fatální chybě při návrhu, kdy byly díky nešťasnému dělení stran mezi výkresem mechanických rozměrů procesoru a tabulce okotovaných hodnot špatně navrženy dosedací plošky pro vývody procesoru. První prototyp tištěného spoje proto nemohl být plně osazen a otestován, došlo pouze k dílčímu testování subsystémů.

V druhém kole se podařilo již vyrobit plně osaditelný prototyp ve dvou verzích, jeden se zálohováním napájení pomocí superkapacitoru a druhý se zálohováním pomocí baterie. Bylo odhaleno několik drobných chyb ve zdrojové části obvodů, které byly oživovány jako první, ty se však podařilo odstranit pomocí drátových propojek. Hlavním cílem testování bylo ověření funkčnosti přepracované zdrojové části a rozhodnutí o finální variантě zálohování napájení.

Ve třetím kole byly opraveny chyby zjištěné při oživování druhého prototypu a byla vyrobena pouze vítězná verze z variant superkapacitor/baterie. Dále byl použit čtrnáctipinový konektor pro možnost snadného připojení modulu do předpřipravené zástavby v lokomotivě. Tím došlo k oddelení pracovních činností zástavby výzbroje do lokomotiv od výroby a oživení řídicích modulů. Nyní lze tyto činnosti provádět nezávisle na sobě.

## 4.1 Přepracovaná zdrojová část modulu

První změnou ve zdrojové části modulu je výměna lineárního stabilizátoru napětí L7805, s 5 V výstupem, za spínaný zdroj TPS62112 od firmy Texas Instruments se shodným výstupním napětím. Důvodem bylo mírné zahřívání původního stabilizátoru v důsledku poměrně velkého napěťového rozdílu mezi vstupem a výstupem stabilizátoru. Zahřívání nebylo nebezpečné pro obvod, protože má pouzdro vybavené dostatečnou chladičí plochou, ale hrozilo by poškození plastové karoserie lokomotivy při zakapotování. Druhou výhodou je pak lepší účinnost tohoto obvodu. Zapojení je realizováno dle doporučení výrobce, viz. (TEXAS INSTRUMENTS, 2009b).

Za tímto vstupním odvodem, který dodáva energii pro procesor, BT modul, měřící obvod zpětně indukovaného napětí a IR balízu, se řešení rodělo na dvě možné řešení zálohování napájení.

- řešení se superkapacitorem, který má předřazenou proudovou ochranu pro omezení nabíjecího proudu při zapnutí modulu
- řešení s jednočlánkovou Li-Pol baterií, se jmenovitým napětím 3,7 V. Správa nabíjení je řešena příslušným specializovaným obvodem

Na výstupu obou zálohovacích řešení je spínaný zdroj TPS63021 s přednastaveným výstupním napětím 3,3 V (tak též od firmy Texas Instruments). Tento spínaný zdroj je schopen pracovat v režimu "step-up" i "step-down" s automatickým výběrem funkce dle aktuálního napětí na vstupu obvodu. Tato funkčnost je nutná, protože vzhledem k úbytku napětí na diodách, případně na tranzistoru spínajícím záložní napájení, se vstupní napětí tohoto spínaného zdroje pohybuje v rozmezí nad i pod 3,3 V a nelze proto použít obvod pracující pouze v režimu "step-up", respektive v režimu "step-down". Zapojení je také provedeno dle doporučení výrobce, viz. (TEXAS INSTRUMENTS, 2010).

### 4.1.1 Řešení se superkapacitorem

Pro zálohu pomocí superkapacitoru byl použit BZ055A333ZSB od AVX Corporation s kapacitou 33mF na 5,5 V a udávaným vnitřním odporem pouhých 250 mΩ. Malý vnitřní odpor umožňuje velkou proudovou zatížitelnost. Rozměrově se jedná o kvádr s rozmiery

28x17x3,5 mm a vývody pro SMD technologii osazovaní(zkratka SMD je pro Surface Mounted Device, tedy povrchově montované zařízení).

Při návrhu řešení jsem provedl simulaci v prostředí Matlab, aby bylo možné odhadnout dobu, po kterou bude schopen tento kapacitor udržet v chodu 3,3 V větev napájení. Vycházel jsem z následujících předpokladů. Superkapacitor je v ideálním případě před výpadkem napájení nabít na napětí 5 V. Požadovaný výstupní proud zdroje na 3,3 V věti složen z několika složek.

- Bluetooth modul - špičkově, při příjmu i vysílání, max. 120 mA
- Procesor - při taktovací frekvenci 55 MHz, se všemi periferiemi zapnutými 33,3 mA
- Operační zesilovače - max 1,3 mA na obvod
- IR balíza - max. 0,4 mA bez vybuzení, 1,6 mA při vybuzení signálem

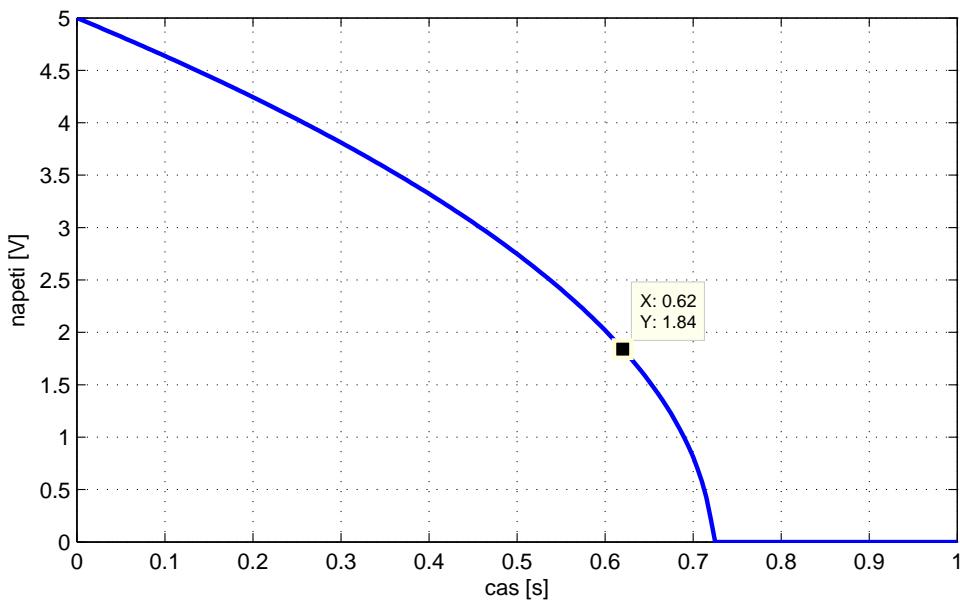
V součtu je tedy požadovaný výstupní proud v nejhorším případě 157,5 mA, budu tedy počítat se  $I_{out} = 160$  mA a  $U_{out} = 3,3$  V. Výpočet je proveden iteračně, kdy požaduji na výstupu konstantní výkon  $P_{out} = I_{out} \cdot U_{out} = 0,52$  W. Udávaný graf účinnosti v garantuje účinnost bezpečně nad 90%. Budu počítat orientačně s účinností  $\eta = P_{in}/P_{out} = 0,9$ . Je tedy třeba dodávat na vstupu zdroje  $P_{in} = P_{out}/\eta = 0,58$  W. Budu tedy iteračně počítat odebíraný proud na vstupu při známém napětí a výkonu jako  $I_{in} = P_{in}/U_{in}$ . Pokud tedy vyjdu z rovnic  $Q = I \cdot t$  a  $U = Q/C$ , budu iteračně řešit rovnici

$$I_i = \frac{P_{in}}{u_i}$$

$$u_{i+1} = u_i - \left( \frac{I_i \cdot \Delta t}{C} \right)$$

Vnitřní odpor superkapacitoru jsem zanedbal, protože na počátku procesu je na vnitřním odporu úbytek pouze  $P_{in}/U_0 \cdot R_{in} = (0,58/5) \cdot 0,25 = 0,029$  V, takže úbytek na vnitřním odporu superkapacitoru je opravdu zanedbatelný. Nechal jsem tedy vypočítat průběh napětí na superkapacitoru s velmi jemným časovým krokem  $\Delta t = 0,005$  s. Výsledek simulace je na obr. 4.1. V grafu je označen bod, kdy přestává pracovat spínaný zdroj 3,3 V věti. K výpadku napájení na 3,3 V věti tedy dojde v nejhorším případě za 0,62 s. To je dostatek času pro zjištění nebezpečného pádu napětí, když napětí je měřeno AD převodníkem každou ms a napětí na superkapacitoru je zjištováno v okamžiku, kdy je přiváděn výkon na motor, nebo minimálně jednou za 50 ms. Stejně tak se jedná o dostatečnou dobu na zaslání zprávy BT modulu o detekovaném pádu napájení, aby byl

nadřazený server uvědoměn o uváznutí soupravy na kolejisti. Server by měl mít i přehled o poloze uváznutí, protože lokomotiva do poslední chvíle zasílá informace o odometrii.

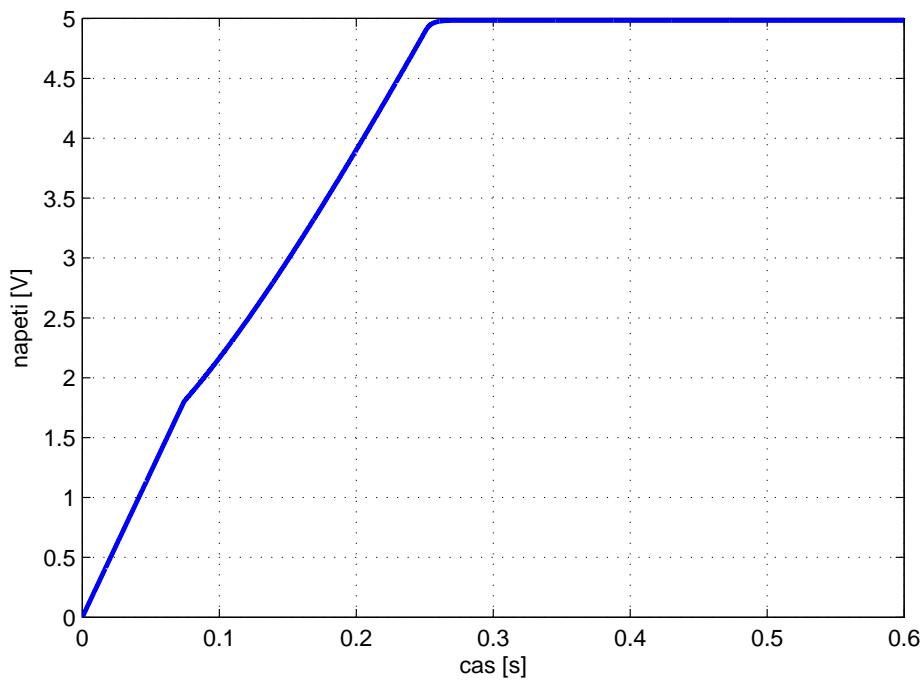


Obrázek 4.1: Simulace vybíjení superkapacitoru při výpadku napájení

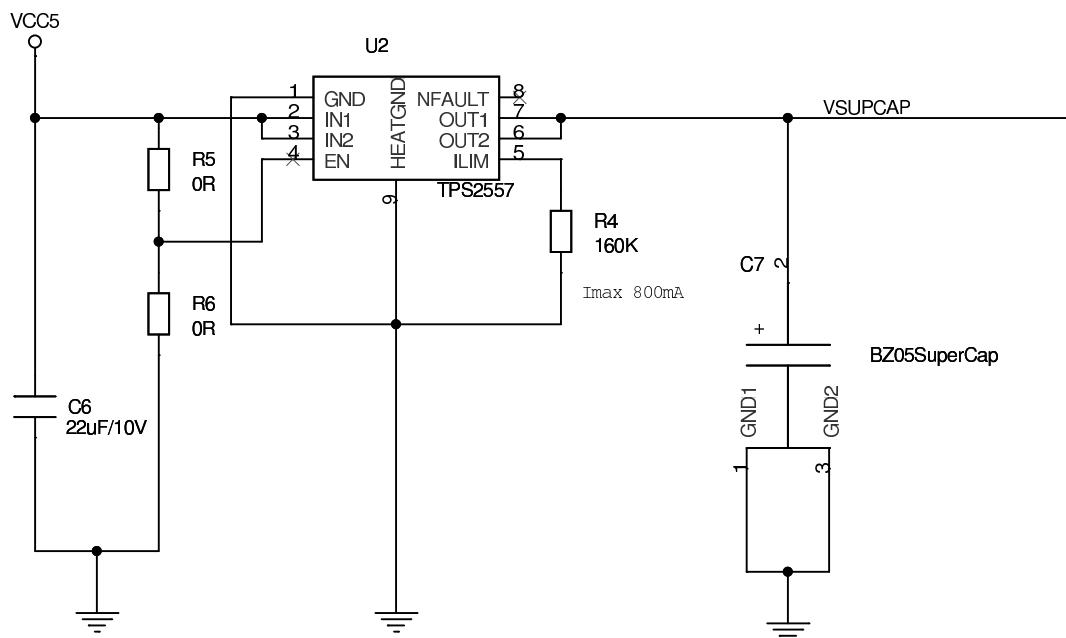
Pro kompletní přehled situace byl provedena i opačná simulace náběhu napájení při omezení nabíjecího proudu na  $I_{lim} = 0,8$  A. Nabíjecí proud byl omezen speciálním obvodem TPS2557 od firmy Texas Instruments, který dovoluje programovatelné omezení výstupního proudu pomocí externího odporu. Způsob řešení byl obdobný. Ve výpočtu je zahrnut i náběh spínaného zdroje při dosažení  $U_c = 1,8$  V. Výsledek je na obr. 4.2.

Nabíjení superkapacitoru je cca třikrát rychlejší než vybíjení, takže nehrozí během jízdy falešná detekce výpadku napájení, protože by musela být lokomotiva více než třetinu času izolována od kolejnic, respektive ještě nepříznivějším poměrem, protože výpadky by musely proniknout přes vstupní blokování a spínaný zdroj z vstupního napětí na 5 V větev. To je bezpečná rezerva, protože v takovém případě by mnohem dříve došlo k výpadku trakce a zastavení lokomotivy, tedy právě k opravdovému uváznutí.

Navržené schema zapojení se superkapacitorem je uvedeno na obr. 4.3. Nulové propojky R5 a R6 dovolují použít libovolnou verzi obvodu TPS2556/7. Stejně tak použití jedné z propojek (podle typu obvodu) lze oddělit 3,3 V větev napájení od zbytku zdrojové části a tím usnadňuje postupné oživování modulu. Hodnota odporu určujícího proudové omezení je dle dokumentace výrobce, viz. (TEXAS INSTRUMENTS, 2009a).



Obrázek 4.2: Simulace nabíjení superkapacitoru při náběhu napájení

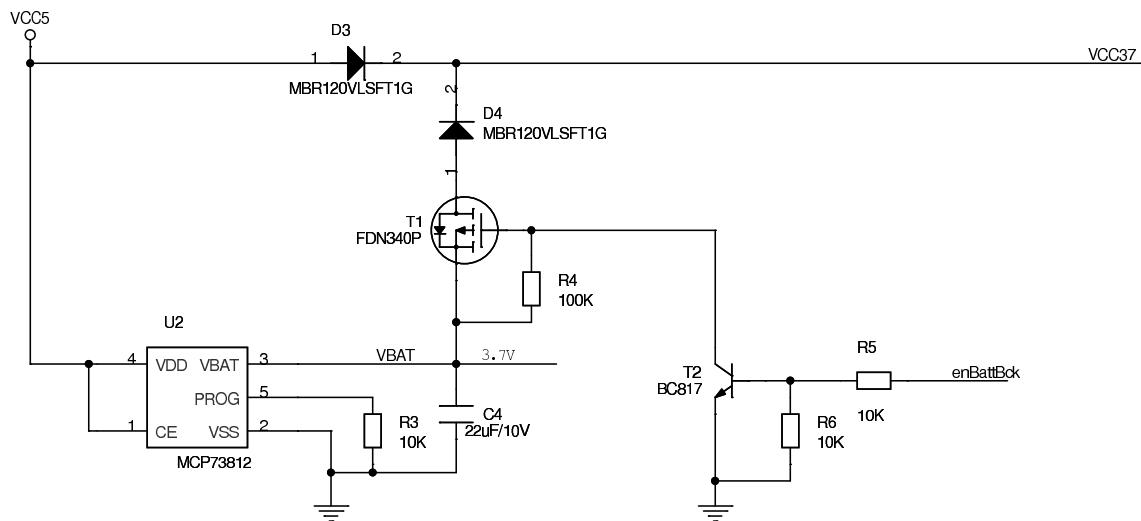


Obrázek 4.3: Schema zálohy napájení pomocí superkapacitoru

### 4.1.2 Řešení s baterií

Bateriová záloha se skládá z jednočlánkové Li-Pol baterie s jmenovitým napětím 3,7 V a jmenovitou kapacitou 110 mAh. Baterie má rozměry 30x15x9,5 mm. Nabíjení baterie je spravováno integrovaným obvodem MCP73812 od firmy Microchip. Tento obvod dobíjí baterii pomocí předprogramovaného konstantního proudu dokud není na jeho výstupu 4,2 V. Velikost nabíjecího proudu je doporučena volit jako 1 C, tedy jako jednonásobek kapacity v Ah. Tak lze dosáhnout nejkratšího času nabíjení bez degradace životnosti baterie. Podrobný popis funkce obvodu je uveden v jeho dokumentaci, viz. (MICROCHIP, 2007).

Dle teoretického předpokladu z kap. 4.1.1 je třeba na vstupu spínaného zdroje dodávat v nejhorším případě výkon  $P_{in} = 0,58$  W. To by při nominální hodnotě napětí na baterii znamenalo odběr 160 mA, to znamená hrubým odhadem výdrž cca 40 min. Protože není třeba, aby zálohování udržovalo 3,3 V větev napájení po tak dlouhou dobu a zároveň není žádoucí, aby spínaný zdroj 3,3 V větve vybíjel baterii za přípustnou mez její vybíjecí charakteristiky, byl zařazen mezi zdroj a baterii spínací tranzistor, který dovoluje procesoru v případě výpadku napájení po určité době řízeně přejít do vypnutého stavu. Dále je třeba ošetřit napájení tak, aby v případě běžné funkce byla baterie nabíjena obvodem MCP73812 a spínaný zdroj 3,3 V větve byl zásoben z 5 V napájení. Je třeba také zamazit toku proudu z 5 V větve směrem do baterie. Výsledné schema řešení je uvedeno na obr. 4.4.



Obrázek 4.4: Schema zálohy napájení pomocí baterie

Na schematu je vidět oddělení napájení z 5 V větve a z baterie pomocí dvou shottkyho diod D3 a D4 s nízkým přechodovým napětím cca 0,3 V. Aktivace záložního napájení z baterie je prováděna pomocí unipolárního tranzistoru T1 s P kanálem, který je ovládán procesorem přes bipolární tranzistor T2. Aktivace nabíjecího obvodu baterie U2 je spojeno s 5 V větví napájení, aby byl obvod aktivní vždy, když je modul v provozu. Napětí na baterii je pro účely diagnostiky sledováno pomocí AD převodníku na procesoru.

#### 4.1.3 Výběr finálního řešení zálohování

Zálohování napájení je motivováno nutností řízeného ukončení komunikace mezi BT serverem a řídícím modulem. U předchozích prototypů nebylo řešeno řízené vypnutí modulu a pokud došlo k neočekávanému výpadku napájení, BT server zaznamenal tuto ztrátu se značným zpožděním. Stějně tak je třeba zajistit překlenutí doby mezi přepólováním napájení z kolejí při pruzejdu obratištěm soupravy a ztráta komunikace s modulem je nežádoucí.

Během reálného testování se ukázalo, že skutečná výdrž napájení před kompletním výpadkem na 3,3 V věti je oproti odhadu delší, cca 2 s. Navzdoru tomuto faktu bylo rozhodnuto o použití bateriové zálohy. Ve prospěch bateriového řešení hovoří několik argumentů.

- vytrvalost - bateriová záloha poskytuje nesrovnatelně delší možnost provozu při výpadku napájení, a tím i bezpečnější řízené vypnutí modulu
- cena - superkapacitor stojí 16\$ a obvod omezující proud při náběhu stojí 2\$ oproti 2\$ za baterii a 0,68\$ za nabíjecí obvod baterie a dve diody po 0,5\$.
- dostupnost součástek a podpůrných obvodů - dostupnost obvodu TPS2556/7 není zcela bezproblémová, dosavadní exempláře byli získány jako vzorky přímo od výrobce. Vlastní superkapacitor není možné sehnat v rámci České republiky, použitý vzorek byl zakoupen v americkém obchodě. Oproti tomu Li-Pol baterie jsou bez problému dostupné ve velmi podobném provedení od několika různých výrobců a podpůrné obvody jsou také mnohem lépe dostupné.

Výhodou superkapacitoru je poněkud menší rozměr. Pevnou montáž na desku plošného spoje nelze považovat za výhodu, protože zástavba modulu je v každé lokomotivě značně

rozdílná a né vždy je méně flexibilní rozměrová konfigurace výhodou. Druhou výhodou superkapacitoru je menší množství podpůrných součástek, ale úspora místa na tištěném spoji by byla ve výsledku nepatrná.

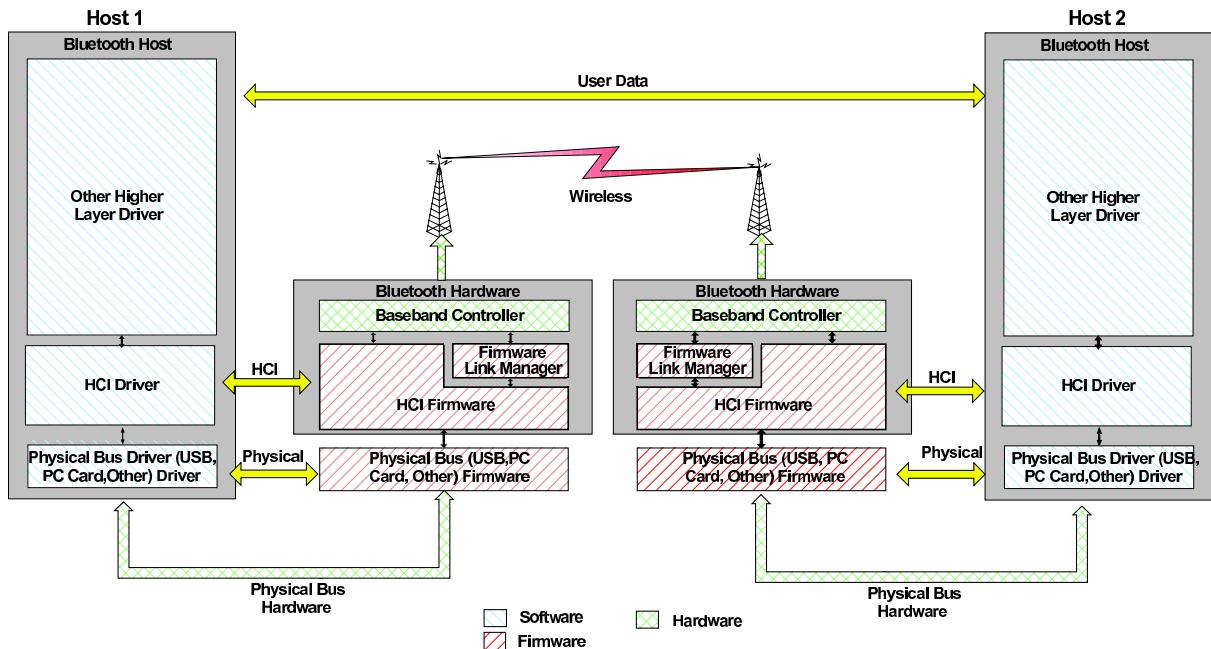
## 4.2 Implementace HCI rozhraní pro komunikaci s BT modulem

HCI rozhraní (Host Controller Interface) zpřístupňuje ovládací rozhraní pro ovládání kontroleru radiového modulu, linkového manažera a přístup k řídícím a stavovým registrům hardwaru. Jeho základní specifikace je určena dokumentem (BLUETOOTH SIG, 2001) a je společná pro všechny Bluetooth zařízení. Tato norma je veřejně dostupná na stránkách Bluetooth SIG (Special Interest Group). Konkrétní implementace se mírně liší podle použitého fyzického komunikačního rozhraní (USB, RS232, UART). Použitý modul WML-C40A od firmy BlueRadios respektuje tuto specifikaci a je možné s ním komunikovat pomocí rozhraními USB, nebo UART. Řídící modul využívá komunikaci pomocí rozhraní UART s hardwarovým řízením toku pomocí signálů RTS/CTS.

Pozice HCI rozhraní mezi koncovými zařízeními je naznačeno na obr. 4.5, přičemž jeden z hostů je BT server (postavený na platformě AVR, konkrétně ATmega162 a dále připojen do systému pomocí rozhraní CAN) a druhý je řídící modul lokomotivy. BT server je schopen simultálního spojení až se sedmi dalšími zařízeními v pozici "slave". Spojení mezi moduly je prováděno na základě fyzických MAC adres BT modulů. Při úspěšném propojení je spojení přidělen náhodně vygenerovaný "link key", který jednoznačně určuje aktuální spojení mezi dvěma hosty.

HCI rozhraní rozlišuje zprávy do několika kategorií:

- HCI Command Packet - řídící příkazy pro modul, slouží pro ovládání hardwaru BT modulu
- HCI ACL Data Packet - datový paket používaný při ACL spojení (Asynchronous Connection Link), určen pro paketově orientovanou asynchronní výměnu dat
- HCI SCO Data Packet - datový paket používaný při SCO spojení (Synchronous Connection Oriented), určen pro stálý datový tok např. hlasových dat



Obrázek 4.5: Schema komunikačního kanálu mezi zařízeními pomocí rozhraní HCI

- HCI Event Packet - paket pro asynchronně nastávající události

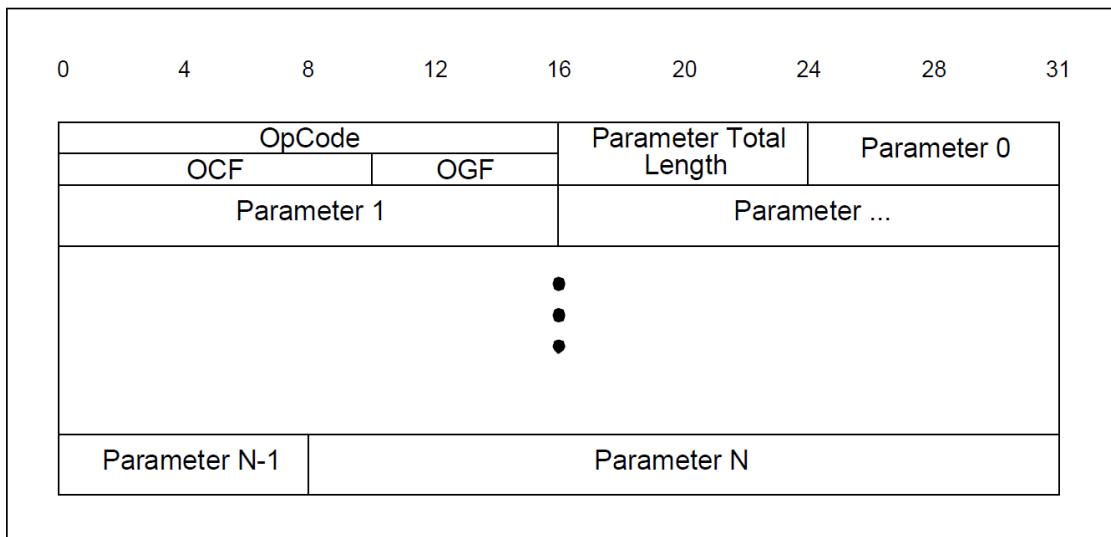
Fyzické rozhraní BT modulu je továrně nastaveno na 112000 baud/s, bez parity, jeden stop bit, RTS/CTS řízení toku dat. Změna nastavení je možná pomocí SPI rozhraní, ale není třeba, procesor řídícího modulu je nastaven shodně. BT modul komunikuje pomocí ustálených zpráv. Každá zpráva je uvozena kódem (0x01 až 0x04), rozlišujícím výše zmíněnou kategorii, do které zpráva spadá. Formát následujících dat za hlavičkou je pak určen podle kategorie zprávy.

V rámci cílového nasazení je použito jen několik nutných příkazů. Jejich popis je rozebrán v následujících podkapitolách.

### 4.2.1 HCI Command Packet

Struktura HCI Command Packetu je uvedena na obr. 4.6. První dva byty obsahují "OpCode", který je dělí na dvě části, 6 bitů OGF (OpCode Group Field) a 10 bitů OCF (OpCode Command Field). OGF určuje skupinu, do které příkaz patří a OCF

udává přímo kód příkazu z příslušné skupiny. Nejvyžší možná skupina OGF (0x3F) je rezervována pro speciální použití a je na libovůli každého výrobce. Následuje byte udávající počet následujících parametrů do konce paketu.



Obrázek 4.6: Formát HCI Command Packetu

Řídicí modul využívá následující příkazy:

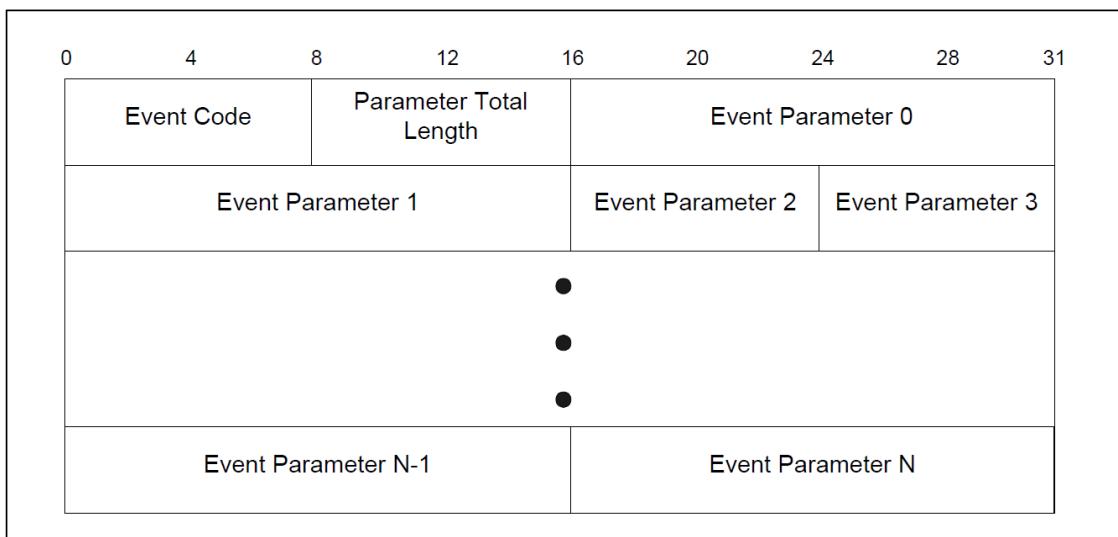
- OGF 0x01 - Host Controller & Baseband Commands
  - OCF 0x0C - Link Key Request Negative Reply - Slouží pro zaslání odpovědi na příchozí událost Link Key Request, kdy se snaží nějaké zařízení bezdrátově spojit s BT modulem. Tento příkaz zasílá negativní odpověď, tzn. že zařízení nejsou dosud spojena žádným klíčem (často se mluví o spárování zařízení). Vstupním parametrem je fyzická MAC adresa BT zařízení, kterému odpovídáme.
  - OCF 0x0D - PIN Code Request Reply - Slouží pro zaslání odpovědi na příchozí událost PIN Code Request. Vstupním parametrem je MAC adresa BT zařízení, délka PIN kódu a vlastní PIN kód.
- OGF 0x03 - Host Controller & Baseband Commands
  - OCF 0x03 - Reset - softwarový reset BT modulu
  - OCF 0x05 - Set Event Filter - filtr příchozích událostí, BT modul podle nastavení filtru přeposílá hostovi pouze události (Events), které projdou filtrem.

- OCF 0x13 - Change Local Name - změní jméno zařízení, které je viditelné zpod kterým se dodatečně identifikuje. Nové jméno zařízení je předáváno jako vstupní parametry v kódování UTF-8.
- OCF 0x1A - Write Scan Enable - podle parametru povoluje/zakazuje zjistitelnost BT modulu.
- OCF 0x20 - Write Authentication Enable - podle parametru povoluje/zakazuje nezabezpečené spojení.

Pokud není uvedeno jinak, vrací tyto příkazy jediný parametr, který se nazývá status. Pokud příkaz vrátí nulu, znamená to úspěšné provedení příkazu, číslo různé od nuly znamená neúspěch, význam lze dohledat v tabulce chybových kódů v dokumentaci (BLUETOOTH SIG, 2001, str. 766).

#### 4.2.2 HCI Event Packet

Struktura HCI Event Packetu je uvedena na obr. 4.7. První byte udává Event Code, tedy kód nastalé události. Následuje druhý byte, udávající počet následujících Event parametrů a zbytek paketu tvoří tyto parametry. Nejvyšší kód události je opět rezervován pro událost specifickou pro každého výrobce.



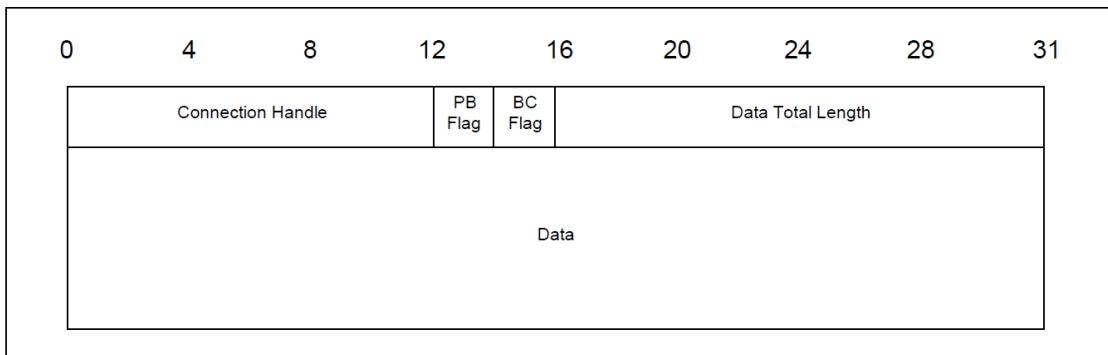
Obrázek 4.7: Formát HCI Event Packetu

Dokumentace (BLUETOOTH SIG, 2001) obsahuje plný výčet možných příchozích událostí. Řídicí modul reaguje pouze na určité očekávané typy příchozích událostí.

- Code 0x03 - Connection Complete Event - potvrzení spojení mezi BT moduly. Obsahuje informace o fyzické MAC adrese vzdáleného BT modulu, vygenerovaný identifikátor Connection Handle pro toto spojení a způsobu zabezpečení kanálu.
- Code 0x05 - Disconnection Complete Event - událost ztráty spojení s vzdáleným BT modulem. Obsahuje informaci o Connection Handle tohoto spojení a důvod ztráty spojení dle Error Listu.
- Code 0x0E - Command Complete Event - standardní odpověď na všechny Command Pakety, nese informaci o provedení/neprovedení přijmutého příkazu.
- Code 0x16 - Pin Code Request Event - vzdálený BT modul se pokouší vytvořit nové zabezpečené spojení pomocí PIN kódu.
- Code 0x17 - Link Key Request Event - vzdálený BT modul se pokouší spojit s BT modulem pomocí Link key (vytvořeném při případném přechozím spárování). Příchozí událost obsahuje jen MAC adresu vzdáleného BT modulu, je na uživateli, zda pro tuto adresu má uložen Link Key a podle toho odpoví na tuto událost.
- Code 0x18 - Link Key Notification Event - událost oznamující úspěšné spojení mezi BT moduly. Její součástí je Link Key, který může být použit při příštím spojení již bez PIN kódu.

#### 4.2.3 HCI ACL/SCO Data Packet

Struktura datových paketů je uvedena na obr. 4.8. Pro oba typy (ACL a SCO) datových paketů je struktura shodná. Prvních 12 bitů obsahuje "Connection Handle", což je již zmiňovaný jednoznačný identifikátor vygenerovaný pro spojení mezi dvěma BT zařízeními. Následují dva byty Packet Boundary Flag určující zda se jedná a první startovací paket vysílané zprávy, nebo některý z následujících. Další dva byty Broadcast Flag určují, zda se jedná o zprávu pro konkrétní zařízení, nebo zda se má provádět broadcast pro všechny zařízení typu slave v okolí (i ty co zatím pouze naslouchají), nebo zda se jedná o broadcast vysílání pouze pro slave zařízení již registrovaná do Piconetu vysílajícího zařízení.



Obrázek 4.8: Formát HCI Data Packetu

Pro komunikaci s BT servery se používají datové pakety typu ACL. Příchozí zprávy chodí periodicky z nadřízeného BT serveru dle dohodnutého formátu. V případě příliš dlouhé prodlevy považuje modul komunikaci za přerušenou a zastavuje lokomotivu. Řídicí modul generuje dva typy zpráv. Periodicky zasílá zpět svůj stav dle uživatelsky definovaného protokolu a v případě průjezdu přes IR balízu a úspěšném dekódování nové hodnoty odesílá zkrácenou zprávu o čísle balízy a stavu odometrie, protože odometrie je v okamžiku průjezdu přes balízu resetována.

Formát datagramů vyměňovaných mezi BT serverem a řídicími moduly jsou uvedeny v tabulkách 4.1, 4.4 a 4.5. Podrobný význam příchozích dat je dodefinován v tab. 4.2 a 4.3. Datagramy jsou zapouzdřené do pole Data v rámci datových paketů uvedených na obr. 4.8.

| význam                        | byte |                      |                |   |   |   |   |   |
|-------------------------------|------|----------------------|----------------|---|---|---|---|---|
|                               | 0    | 1                    | 2              | 3 | 4 | 5 | 6 | 7 |
| řídicí data                   | 0x00 | pož. rychlosť        | zvuk/osvětlení | - | - | - | - | - |
| reset řídicího modulu         | 0x01 | -                    | -              | - | - | - | - | - |
| inicializace řídicího modulu  | 0x02 | -                    | -              | - | - | - | - | - |
| přednastavení řídicího modulu | 0x03 | odometrie            | balíza         | - | - | - | - | - |
| povolení k jízdě              | 0x04 | ano (0x01)/ne (0x00) | -              | - | - | - | - | - |

Tabulka 4.1: Formát příchozího datagramu od BT serveru

| bit                    |   |   |   |   |   |                           |   |
|------------------------|---|---|---|---|---|---------------------------|---|
| 0                      | 1 | 2 | 3 | 4 | 5 | 6                         | 7 |
| hodnota pož. rychlosti |   |   |   |   |   | směr (0 - sudý/1 - lichý) |   |

Tabulka 4.2: Význam nastavení rychlosti v příchozích datech

| bit                  |                         |                     |                        |               |       |       |            |
|----------------------|-------------------------|---------------------|------------------------|---------------|-------|-------|------------|
| 0                    | 1                       | 2                   | 3                      | 4             | 5     | 6     | 7          |
| světla - přední bílá | světla - přední červená | světla - zadní bílá | světla - zadní červená | zvuk globálně | motor | hvizd | kom-presor |

Tabulka 4.3: Význam nastavení světla a zvuku v příchozích datech

| byte   |            |           |               |                          |                |   |   |
|--------|------------|-----------|---------------|--------------------------|----------------|---|---|
| 0      | 1          | 2         | 3             | 4                        | 5              | 6 | 7 |
| status | odome-trie | IR balíza | pož. rychlost | nasta-vení světla, zvuku | napětí baterie | - | - |

Tabulka 4.4: Formát standard. odchozího datagramu pro BT server

| byte   |            |           |   |   |   |   |   |
|--------|------------|-----------|---|---|---|---|---|
| 0      | 1          | 2         | 3 | 4 | 5 | 6 | 7 |
| status | odome-trie | IR balíza | - | - | - | - | - |

Tabulka 4.5: Formát urgentního odchozího datagramu pro BT server

#### 4.2.4 Proces vyjednání spojení mezi Bluetooth moduly

Pro lepší pochopení používaných zpráv je třeba popsat způsob, jakým je vyjednáváno spojení mezi BT moduly při navazování spojení. BT zařízení obecně umožňují použít pro zabezpečené spojení tzv. proces párování. Významem je navázání zabezpečeného spojení pomocí PIN kódu pouze jednou a použití vygenerovaného linkového klíče pro budoucí spojení, kdy bude možné se mezi spárovanými BT moduly spojit automaticky,

bez nutnosti ověřování pomocí PIN kódu. Tento způsob není pro komunikaci řídicího modulu s BT serverem použit, ale je nutné s ním počítat i při navazováním nového spojení.

Po zapnutí řídicího modulu proběhne nejprve inicializace BT modulu následujícími kroky:

- HW reset pomocí NRESET pinu BT modulu. Výrobce uvádí nutnost aktivace resetu na min. 5 ms.
- SW reset pomocí příslušného HCI příkazu. Procesor si tak ověří základní funkčnost BT modulu, protože musí dostat odpověď v podobě Command Complete Event.
- Změna jména BT modulu, kterým se identifikuje do okolí.
- Nastavení Event filtru na automatické přijmutí spojení od libovolného BT zařízení.
- Povolení zjistitelnosti BT zařízení pomocí příkazu Write Scan Enable.
- Nastavení požadavku na zabezpečené spojení pomocí příkazu Write Authentication Enable.

Inicializace vyžaduje potvrzování úspěšného provedení příkazů pomocí události Command Complete Event v nastaveném časovém limitu (500 ms). V opačném případě stavový automat skáče zpět na začátek a provádí HW reset BT modulu. Pokud proběhne inicializace úspěšně, stavový automat vyčkává na událost Link Key Request Event, tedy na pokus o příchozí spojení ze strany BT serveru. V případě příchodu této události dochází k následující výměně příkazů:

- IN - Link Key Request Event - pokus o spojení ze strany BT serveru, spojení je vždy oznámeno dotazem na linkový klíč pro MAC adresu, ze které dochází k pokusu o spojení.
- OUT - Link Key Request Negative Reply - řídicí modul odpovídá negativně, protože neprovádí párování s BT serverem a neukládá si proto Link Key pro další použití.
- IN - PIN Code Request Event - BT server se pokusí vytvořit spojení pomocí definovaného PIN kódu.
- OUT - PIN Code Request Reply - řídicí modul odpoví dohodnutým PIN kódem.

- IN - Connection Complete Event - v případě úspěchu je spojení potvrzeno událostí Connection Complete Event, která vrátí hodnotu Connection Handle platnou pouze pro aktuální relaci spojení.

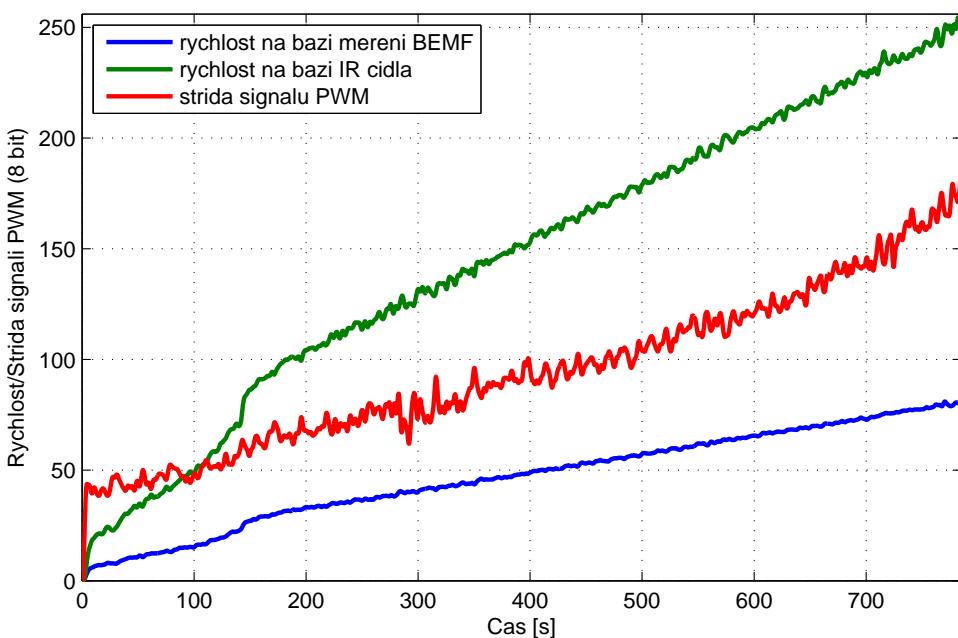
V případě narušení tohoto procesu neočekávanou událostí nebo vypršení časového limitu opět dojde k HW resetu a opakování celé procedury. Pokud se dojde k úspěšnému spojení, je zahájena komunikace mezi BT serverem a řídicím modulem pomocí datových paketů směrovaných pomocí MAC adresy a Connection Handle klíče. Komunikaci může narušit detekce výpadku spojení, kdy BT server neodpoví do časového limitu (1 s). Nebo dojde k výpadku napájení a řídicí modul lokomotivy se v případě neobnovení napájení řízeně odpojí a vypne.

### 4.3 Testování alternativní odometrie

Úspěšná realizace řízení rychlosti motoru pomocí zpětně generovaného napětí mělo za cíl kompletní nahradu měření rychlosti pomocí IR čidla, pracujícího na principu odrazivé optozávory. Důvodem je odstranění nutnosti zástavby čidla do rámu lokomotivy a výroba odrazivých/neodrazivých ploch na setrvačníku motoru. Tento postup se ještě více komplikuje v případě modelů regionálních souprav, které většinou nemají klasickou konstrukci s pětipólovým motorem se setrvačníkem jako klasické lokomotivy. Oproti tomu měření rychlosti pomocí zpětně indukovaného napětí na motoru nevyžaduje žádnou dodatečnou zástavbu do lokomotivy, protože měření je prováděno pomocí přívodů napětí k motoru. Protože lokomotivy nemají shodný převodový poměr mezi motorem a koly, je nutné provádět normování pro obě metody měření rychlosti. Pro plnohodnotnou nahradu bylo třeba otestovat, zda je přesnost odometrie založená na bázi zpětně indukovaného napětí na motoru srovnatelná s odometrií na bázi IR čidla.

Jako první test bylo provedeno měření převodního poměru mezi údaji o rychlosti v používaném rozsahu střídy signálu PWM, respektive v používaném spektru otáček. Měření pomocí zpětně indukovaného napětí nelze provádět do maximální střídy, protože se stoupající hodnotou střídy se zkracuje doba, po kterou je motor ve stavu vysoké impedance a tedy i počet naměřených vzorků za periodu. Aby bylo možné prohlásit měření rychlosti na bázi zpětně indukovaného napětí za ekvivaletní s referenčním měřením pomocí IR závory, je nutné, aby byl stálý převodní poměr mezi oběma způsoby měření

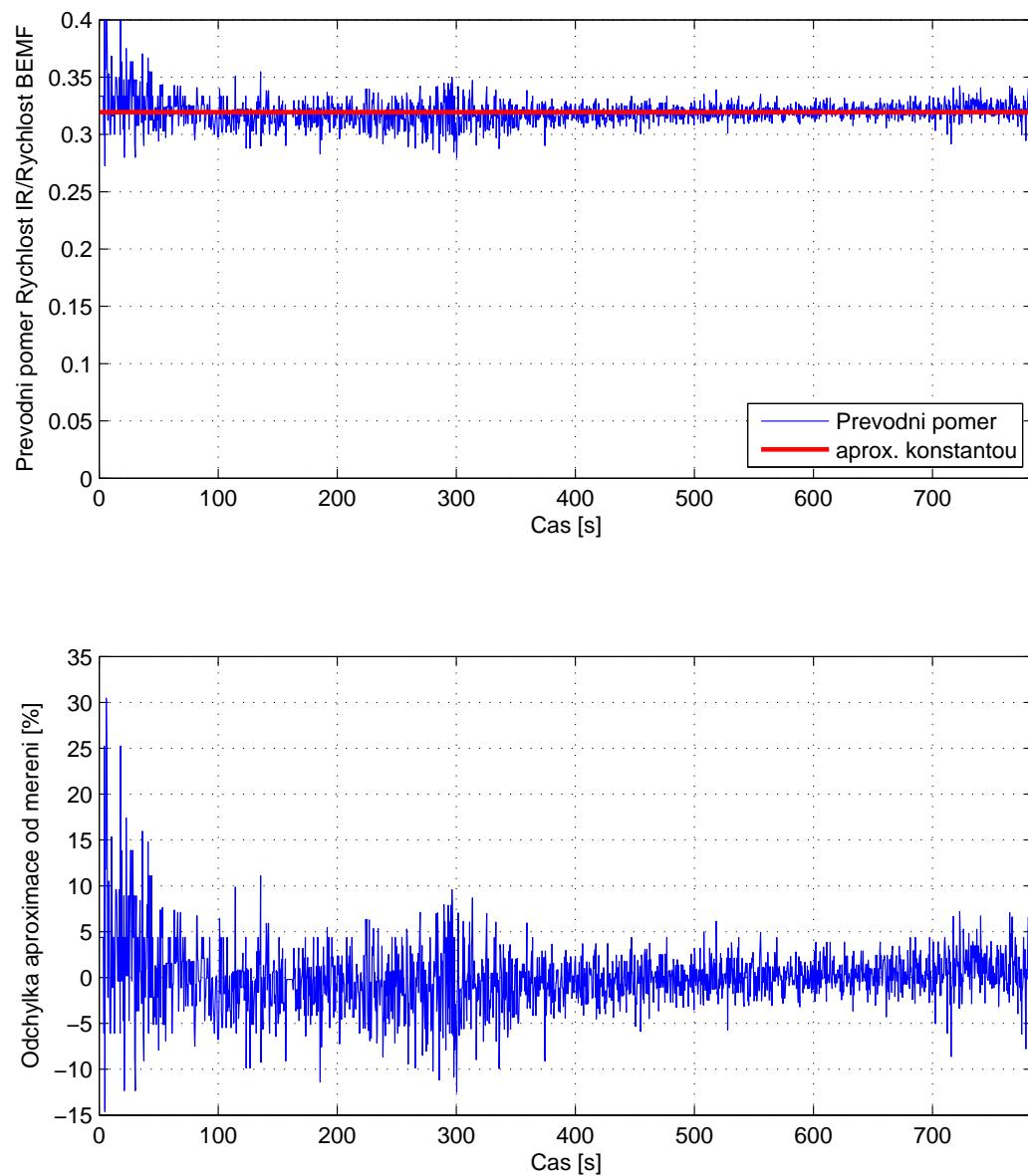
v pokud možno celém spektru otáček. Omezení měřící metody na bázi zpětně indukování napětí není až tak omezující, protože rychlosť jízdy, kdy se blíží střída signálu PWM k maximální hodnotě, je již nebezpečná z hlediska vykolejení soupravy. Měření je zaznamenáno na obr. 4.9, pro omezení zašumění jsem provedl průměrování dat plovoucím oknem o velikosti pěti vzorků.



Obrázek 4.9: Porovnání údajů o rychlosti z různých zdrojů měření

Toto měření dává pouze rámcovou představu o požadované svázanosti měřících metod. Pro exaktní porovnání je třeba vypočítat převodní poměr mezi veličinami měřené rychlosti v naměřeném rozsahu otáček. Graf vypočtených hodnot převodního poměru mezi otáčkami měřenými nepřímo A/D převodníkem a nepřímo IR závorou je vykreslen na obr. 4.10 společně s grafem odchylek. Hodnoty jsou vypočteny z čistých neprůměrovaných dat a proloženy approximací polynomem nultého řádu, tedy konstantou, pomocí metody nejmenších čtverců.

Approximace polynomem nultého řádu vyšla konstanta převodního poměru 0,3194. Pro zajímavost jsem provedl i approximaci pomocí polynomu prvního řádu. Výsledkem byla přímka definovaná jako  $y = k \cdot x + q$  s parametry  $k = 1,9813 \cdot 10^{-7}$  a  $q = 0,3193$ , tedy téměř naprostě shodný výsledek, protože sklon přímky je zanedbatelný. Chyba odhadu na obr. 4.10 je na začátku měření poměrně vysoká, to je způsobeno nízkým odstupem šumu



Obrázek 4.10: Převodní poměr údajů o rychlosti v průběhu celého měření

od užitečné hodnoty naměřené rychlosti při jízdě malou rychlostí. Celkově lze prohlásit, že chyba se drží v rozmezí  $\pm 10\%$ . Vzhledem k časovým konstantám řízené soustavy lze stlačit tento šum pomocí softwarového filtrování.

Jako poslední test jsem provedl jízdu po okruhu s jednou IR balízou. Po každém průjezdu lokomotivy přes balízu jsem změnil vysílaný kód. Tím jsem byl schopem detektovat průjezd jednoho okruhu a sledováním odometrie v PC jsem měl statistiku o změřené ujeté vzdálenosti za každé kolo. Naměřené údaje jsou uvedeny v tab. 4.6.

Měření je zpracováno obdobným způsobem. Provedl jsem odhad délky jednoho kola approximací polynomem nultého rádu a vyšetřil jsem odchylky naměřené délky od approximace. Výsledky jsou uvedeny na obr. 4.11 a obr. 4.12. Z měření ujeté vzdálenosti okruhu lze vyvodit velmi důležité informace o převodním poměru odometrie na skutečnou ujetou vzdálenost, protože délka okruhu je známá. Okruh má průměr 86,6 cm, to znamená obvod 272 cm. Lze tedy odvodit, že vztah pro výpočet ujeté vzdálenosti podle odometrie odvozené od IR čidla je

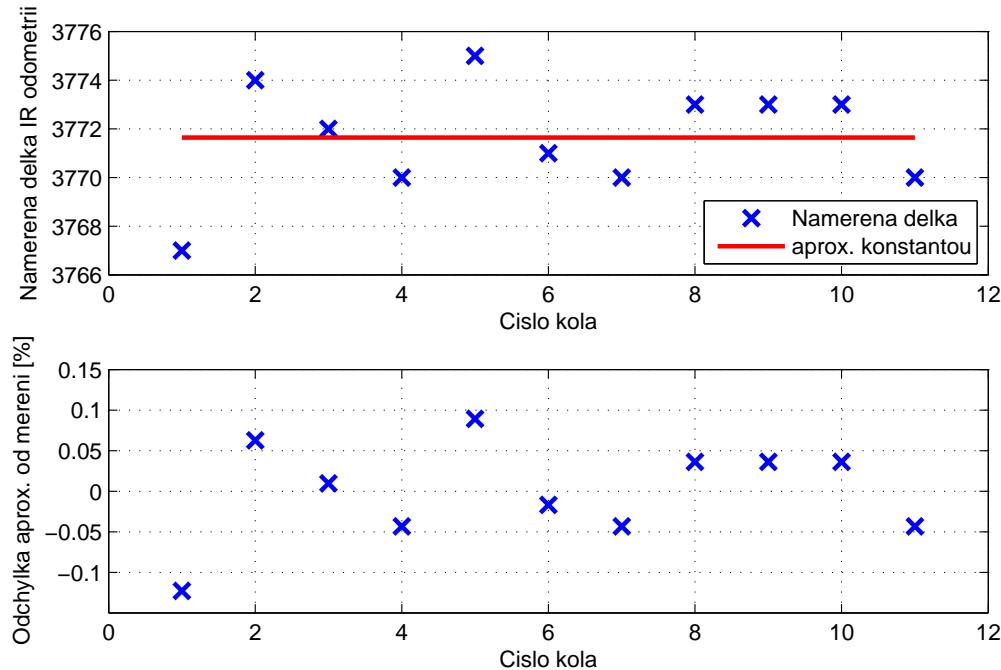
$$s_{IR} = k_{IR} \cdot odo_{IR} \quad \text{kde} \quad k_{IR} = \frac{272}{3771,6} = 0,0721$$

a vztah pro výpočet ujeté vzdálenosti podle odometrie od snímání zpětně indukovaného napětí motoru je

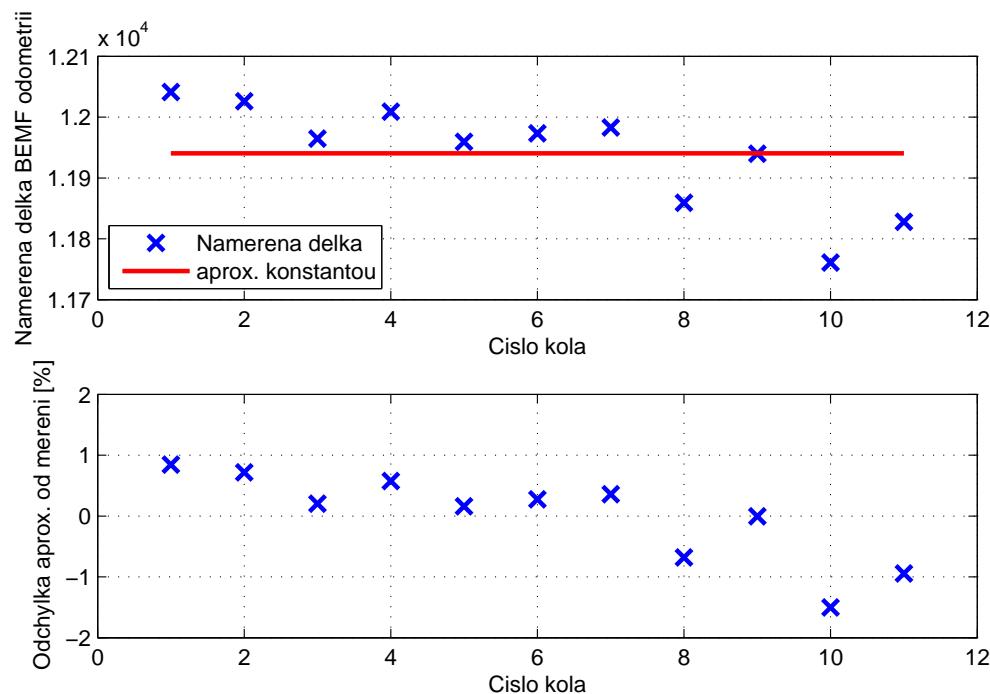
$$s_{AD} = k_{AD} \cdot odo_{AD} \quad \text{kde} \quad k_{AD} = \frac{272}{11940} = 0,0228$$

Dále lze dopočítat rozptyl hodnot ujeté vzdálenosti v reálných hodnotách vzdálenosti. Pokud rozptyl hodnot odometrie odvozené od IR čidla je dle grafu obr. 4.11 cca  $\pm 0,12\%$ , to znamená reálný rozptyl  $\pm(3771,6 \cdot 0,0012 \cdot k_{IR}) = 0,33$  cm. V případě odometrie odvozené od měření zpětně indukovaného napětí dle grafu obr. 4.12 rozptyl cca  $\pm 1,2\%$ , to znamená reálný rozptyl  $\pm(11940 \cdot 0,012 \cdot k_{IR}) = 3,26$  cm.

Odometrie na bázi zpětně indukovaného napětí vykazuje o řád nižší přesnost oproti stávajícímu řešení. V absolutních číslech to znamená stále dostačující přesnost, protože při každém průjezdu přes IR balízu dojde ke kalibraci a měření tedy probíhá relativně oproti poslední projeté IR balíze. Balízy jsou od sebe v kolejisti vzdáleny v rozmezí od jednoho do čtyř metrů, podle složitosti infrastruktury (tedy hustší pokrytí v problematických úsecích), to znamená, že ve stanicích, kde bude třeba zastavovat s vysokou přesností bude ujetá vzdálenost bezpečně nižší, než je délka testovacího okruhu a přesnost bude tedy minimálně stejná nebo lepší než vypočtených  $\pm 3,26$  cm.



Obrázek 4.11: Měření délky okruhu pomocí odometrie na bázi IR čidla



Obrázek 4.12: Měření délky okruhu pomocí odometrie na bázi BEMF

| kolo č. | naměřená hodnota z odometrie<br>na bázi IR čidla | naměřená hodnota z odometrie<br>na bázi BEMF |
|---------|--|--|
| 1       | 3767   | 12041  |
| 2       | 3774   | 12026  |
| 3       | 3772   | 11965  |
| 4       | 3770   | 12009  |
| 5       | 3775   | 11959  |
| 6       | 3771   | 11973  |
| 7       | 3770   | 11983  |
| 8       | 3773   | 11859  |
| 9       | 3773   | 11940  |
| 10      | 3773   | 11761  |
| 11      | 3770   | 11828  |

Tabulka 4.6: Naměřené délky okruhů pomocí odometrie řídicího modulu

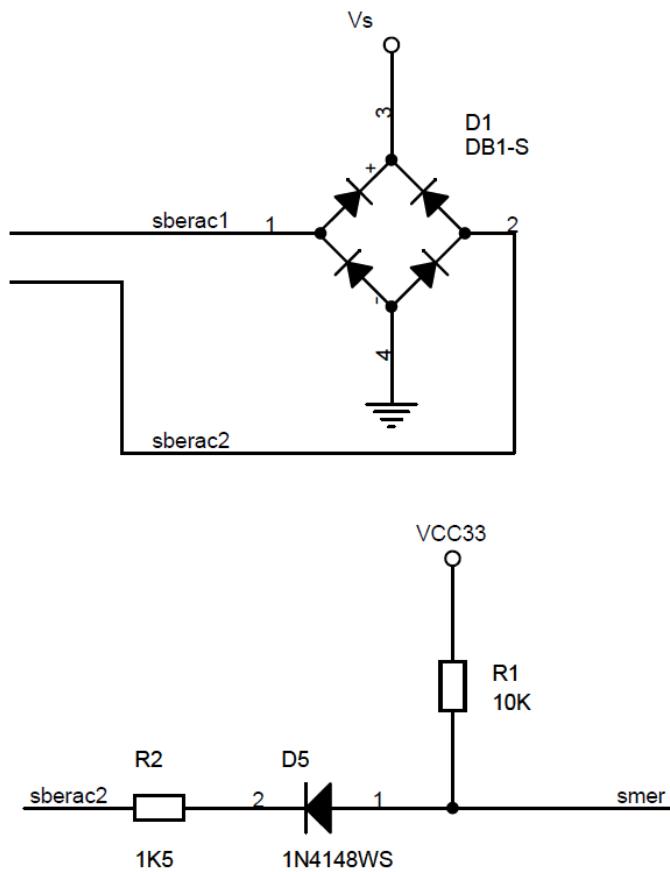
## 4.4 Dodatečně přidané funkce

Během vývoje modulu bylo implementováno několik drobných funkcí řídicího modulu, které většinou vyplynuly z nově použitých funkčních prvků, nebo z drobných změn v požadavcích na modul.

Byla připojena baterie na jeden z multiplexovaných vstupů AD převodníku procesoru, aby bylo možné sledovat aktuální napětí na baterii z důvodu ochrany před podpětím způsobeným vybitím, ale také z důvodu sledovaní životnosti baterie. Baterie je připojena k procesoru pomocí odporového děliče napětí, aby rozsah vstupního napětí na procesoru byl v rozmezí 0 V až  $U_{ref} = 3,3$  V. Dělič zatěžuje baterii odporem 200 k $\Omega$ , to znamená při nominální hodnotě napětí na baterii 3,7 V snesitelný stálý odběr proudu 18,5  $\mu$ A.

Pro detekci polarity napájení v kolejích bylo dříve využíváno optočlenu, který byl zapojen společně s omezovací diodou na sběrače lokomotivy. Optočlen pak v případě kladně polarizované diody stáhnul na svém výstupu úroveň napětí na logickou nulu, v případě závěrně polarizované vstupní diody byl výstup fixován pull-up odporem na úroveň logické jedničky. Nově je polarita zjišťována pomocí kombinace dvou odporů a jedné diody. Výřez ze schematu je vykreslen na obr. 4.13. Pokud je na sběrači kladná polarita napětí, je dioda D5 polarizována závěrně a na vstupu procesoru je vnučeno napětí 3,3 V pomocí pull-up odporu R1. Pokud je na sběrači opačná, je na sběrači před diodou záporné napětí vůči zemi, protože dochází k posuvu napětí země na diodě gretzova můstku. Protože přechodový odpor diody D5 není shodný s přechodovým odporem diod v gretzové můstku, je zařazen do obvodu ještě odpor R2, aby nedocházelo k zatěžování vstupu procesoru napětím nižším, než je jeho potenciál země. Řešení přineslo drobnou úsporu místa především z hlediska výšky nad povrchem tištěného spoje, protože běžně dostupné optočleny jsou většinou v poměrně robustních pouzdrech. Řešení přineslo pouze jednu drobnou komplikaci v podobě nutnosti SW filtrace detekovaného napájení, protože doba odbuzení optočlenu přirozeně filtrovala výpadky signálu při jízdě po špinavých kolejích.

Nově použitý procesor AT91SAM7S dovoluje provedení SW resetu pomocí registru RSTC\_CR (Reset Controller Control Register), není již nutné provádět přednastavení watchdog časovače v kombinaci s řízeným zacyklením programu za účelem provedení resetu procesoru.



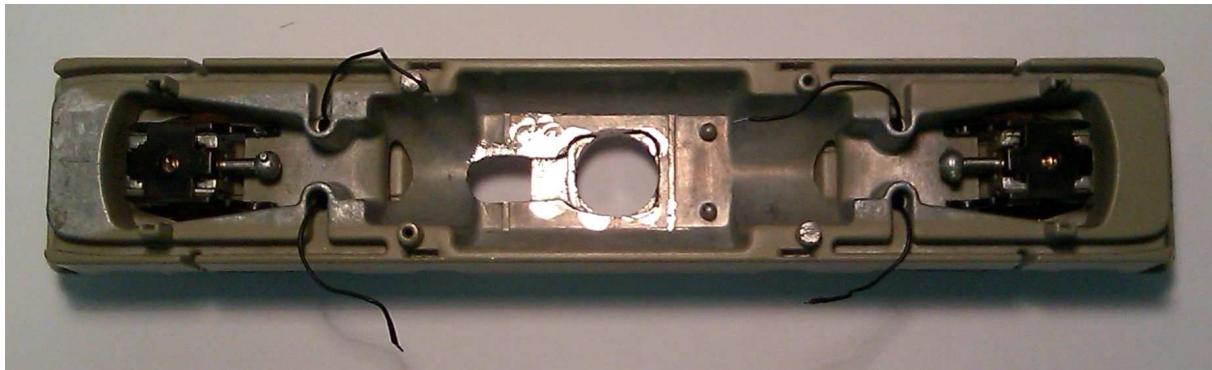
Obrázek 4.13: Schema detekce polarity napájení v kolejích

## 4.5 Zástavba modulu do lokomotivy

Finální prací na zprovoznění řídicího modulu byla zástavba do vlastních modelových lokomotiv. Pro ilustraci byla zdokumentována finální zástavba do lokomotivy ČSD 754 028-9, která byla jednou z nejméně prostorných lokomotiv pro cílovou zástavbu.

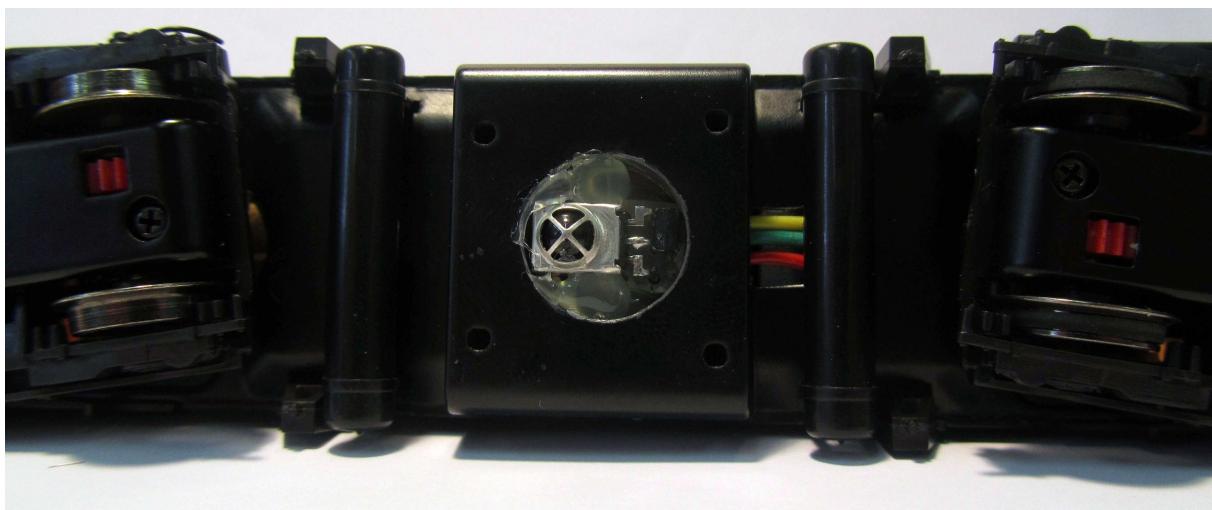
Lokomotivy určené pro zástavbu byly převážně od rakouské firmy Roco. Prvním krokem před každou zástavbou bylo kompletní odstrojení kovového těla lokomotivy, aby bylo možné bezpečně, bez poškození podvozků a dalších plastových částí zasáhnout do těla lokomotivy. Jednalo se především o vrtání do prostoru mezi podvozky lokomotivy pro vytvoření místa pro snímač IR balíz. Dále bylo třeba frézovacím nástavcem vyhloubit žlab pro vedení kabeláže od snímače kolem motoru na sdružený propojovací konektor. Ukázka předpřipraveného těla lokomotivy se zpětně osazenými podvozkami je na obr. 4.14. Tělo lokomotivy patří obdobnému typu lokomotivy (ČSD 753 193-2) se zcela shodnou

zástavbou.



Obrázek 4.14: Tělo lokomotivy po odvrtání a odfrézování

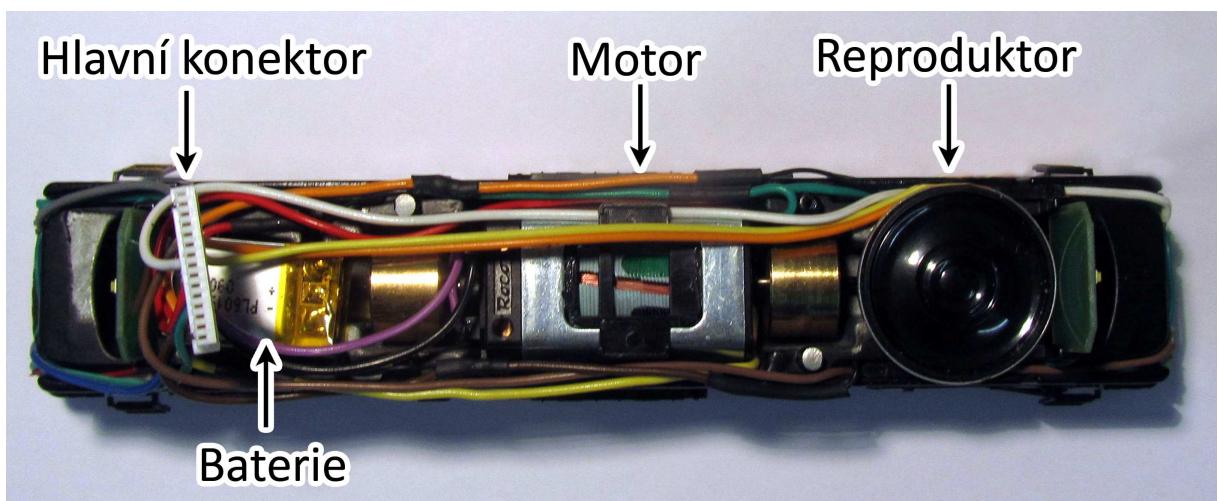
Po mechanickém zásahu do těla lokomotivy byla provedeno zástavba snímače IR balíz na malý odřezek univerzálního tištěného spoje. Následné bylo provedeno nastrojení těla lokomotivy podvozky a motorem. Tělo lokomotivy osazené snímačem je zdokumentováno na obr. 4.15. Každá lokomotiva měla mezi podvozky plastový překryt modelující kontejner reálné výzbroje, do kterého byl vyvrtán pouze průzor pro snímač, takže nebyl narušen vzhled lokomotivy.



Obrázek 4.15: Pohled zespodu na snímač IR balíz bez plastového překrytu

V zásadě byla ve vnitřním prostoru zanechána původní zástavba světel, které jsou

situovány do míst světlovodů umístěných na kapotách lokomotiv. Původní tištěné spoje s diodami byly zbaveny původní kabeláže, pokud byly v lokomotivě použity žárovky, byly osazeny na univerzální tištěné spoje. Následně bylo provedena rozvaha rozmístění tištěného spoje řídicího modulu, reproduktoru a záložní baterie. Kabeláž byla rozměřena přesně na míru a zapájena na své místo. Takto nazbrojené tělo lokomotivy je zdokumentováno na obr. 4.16.

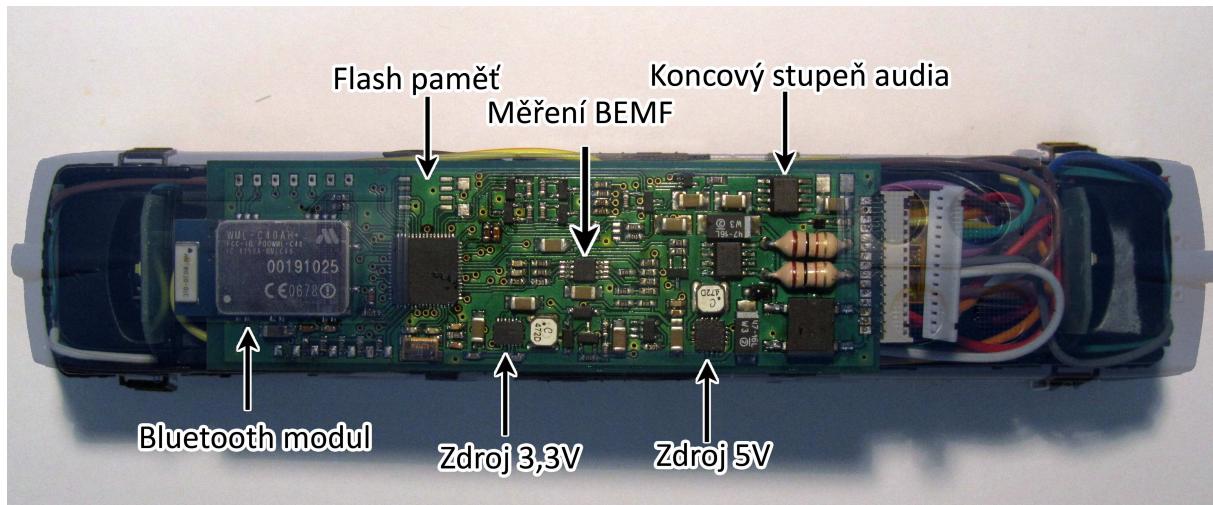


Obrázek 4.16: Hotová zástavba kabeláže bez připojeného řídicího modulu

U takto nazbrojených lokomotiv byla provedena kontrola funkčnosti kabeláže. Pro úplnost je uvedena ještě fotodokumentace s osazeným řídicím modulem (viz. obr. 4.17). Obrázek je opatřen stručnými popisky hlavních bloků řídicího modulu. Pro podrobný popis je třeba nahlédnout do výrobní dokumentace na přiloženém CD.

Předposlední obrázek je pohled z boku, kdy byla provedena fotomontáž snímku odkapotované a zakapotované lokomotivy pro představu o vnitřních nárocích na minimální rozměry řídicího modulu především ve směru ke stropu papoty (viz. obr. 4.18). Omezení na výšku je také patrné z profilu vertikálně zasazených tištěných spojů s diodou pro čelní svítily.

Na posledním obrázku (obr. 4.19) je zprovozněná lokomotiva s modulem. Výsledek je pro laika k nerozeznání od původního stavu před zástavbou elektroniky vlastní výroby.



Obrázek 4.17: Hotová zástavba s osazeným řídicím modulem



Obrázek 4.18: Pohled z boku - složení fotografie s a bez kapoty



Obrázek 4.19: Finální sestava lokomotivy s rozsvícenými hlavními světly



# Kapitola 5

## Závěr

Diplomová práce přinesla po hardwarové stránce finální řešení řídicího modulu, který je schopen spolehlivě řídit stejnosměrný motor lokomotivy za pomocí měření zpětně indukovaného napětí na motoru, obousměrně komunikovat pomocí bezdrátové technologie Bluetooth a standardního komunikačního rozhraní HCI, sbírat informace o IR značkách rozprostřených v infrastruktuře, sbírat informace o ujeté vzdálenosti pomocí integrace, respektive sumace informace o rychlosti za ekvidistantní časové úseky a generovat zvukovou kulisu v podobě emulace sirény a motoru lokomotivy. Drobné nedostatky řešení jsou výhradně softwarového charakteru a vyžadují dlouhodobější testování v reálných podmínkách provozu dopravní laboratoře.

V úvodu je uvedena motivace projektu, přednesení nedostatků současného stavu vývoje řídicího modulu a návrh na způsob řešení těchto nedostatků společně s požadovanou novou funkčností. Konkrétnější způsob řešení bylo třeba nejdříve podložit analýzou pozměněných požadavků na modul a vzhledem k záměru integrace zvukového modulu zvážit výkonovou náročnost. Tato analýza je provedena v podkap. 2.1. Analýza byla celkově cílena na 32 bitové platformy s výkonem cca 50 MIPS a jejím výsledkem bylo použití platformy s jádrem ARMv7TDMI, konkrétně jeho implementace od firmy Atmel v podobě AT91SAM7S. Na tuto platformu byl migrován dosavadní kód a byla ověřena původní funkčnost s pomocí vývojové desky AT91SAM7X-EK.

Po ověření splnění původní funkčnosti bylo přistoupeno k experimentům se zvukovým modulem. Bylo třeba získat základní představu o způsobu míchání více zvukových stop audio signálu do společného výstupu, o principu modelování zvuku motorů obecně. Dále bylo třeba navrhnout způsob, jakým bude procesor generovat audio signál a vybrat vhodný výkonový koncový stupeň. Po splnění těchto úkolů a úspěšném zprovoznění to-

hoto bloku bylo testováno vybavení procesoru externí pamětí typu Data Flash, která měla vyloučit nutnost použití procesoru s velkou interní pamětí (modely s 256 kB a 512 kB flash paměti). Možnost použití externí paměti zůstala zachována jako volitelná (je možné ji osadit na finální verzi řídicího modulu), protože se podařilo získat dostatečné množství procesoru typu AT91SAM7S256 s 256 kB paměti, která doposud pokrývala paměťové nároky na zvukový modul.

V kap. 4 je popsán postup celkové realizace řešení do kompaktní podoby použitelné v modelových lokomotivách. Hlavním úskalím bylo použití moderních součástek v pouzdrách typu QFN, které šetří místo na plošném spoji, ale vyžadují odlišný postup osazování oproti součástkám s klasickými vývody z pouzdra. Nejobtížnější bylo osazení Bluetooth modulu WML-C40A od firmy BlueRadios. Tyto moduly byly osazovány na specializovaném pracovišti Katedry elektrotechnologie, protože jejich radiová část vyžadovala dokonalé spojení se zemním potenciálem. To bylo vzhledem ke konstrukci modulu nemožné realizovat pomocí dostupného vybavení a prototypy osazené pomocí horkozdušné stanice a pájecí pasty vykazovali výrazně horší signálovou citlivost a dosah, protože nebylo fizicky možné dostatečně kvalitně uzemnit plošky č. 25 a 27, nacházející se v blízkosti keramické antény BT modulu.

Dále byla implementována finální verze komunikace pomocí modulů WML-C40A za pomoci standardizovaného rozhraní HCI (Host Command Interface). Doposud byly z testovacích důvodů využívány přeprogramované USB moduly od firmy Acer, postavené na chipsetu CSR (BCR417), které replikovaly transparentní seriovou linku mezi dvěma moduly. To podstatně zjednodušovalo ladění a vývoj řídicího modulu v ranné fázi vývoje a jízdních zkoušek, protože nebylo nutné disponovat Bluetooth serverem a CAN rozhraním, po kterém tento server komunikuje.

Bylo rozhodnuto pro finální řešení snímání otáček motoru pouze pomocí měření zpětně indukovaného napětí, protože bylo prokázáno, že regulace rychlosti jízdy je možná a následná odvozená odometrie poskytuje nižší, ale přesto dostatečnou přesnost pro cílové použití. Výhodou tohoto řešení je podstatné zjednodušení zástaveb řídicího modulu do modelových lokomotiv a souprav.

Pro zálohování napájení bylo rozhodnuto použít Li-Ion baterii, která poskytuje nesrovnatelně větší vytrvalost provozu při výpadku napájení oproti navrhovanému řešení se superkapacitorem.

Celkově byla realizace modulu souborem poměrně velkého množství dílčích problémů, které bylo třeba vyřešit a spojit v jeden kompaktní a funkční celek. Během dlouhodobé práce na projektu dopravního sálu Fakulty dopravní, respektive na této konkrétní části

jsem získal velké množství zkušeností z oblasti návrhu elektronických obvodů, jejich dimenzování, návrhu plošných spojů a vývoj aplikací na různých jednočipových platformách. S odstupem se ukázalo, že zvláště palčivým problémem v oblasti návrhu vlastní elektroniky je nestálost dostupnosti komponent a rychlosť, jakou některé komponenty zastávají a stávají se nedostupnými. Velkým pokrokem bylo nasazení moderních součástek v miniaturních pouzdrech, které umožnili realizovat výrazně pokročilou funkčnost při zachování rozměrů plošného spoje.



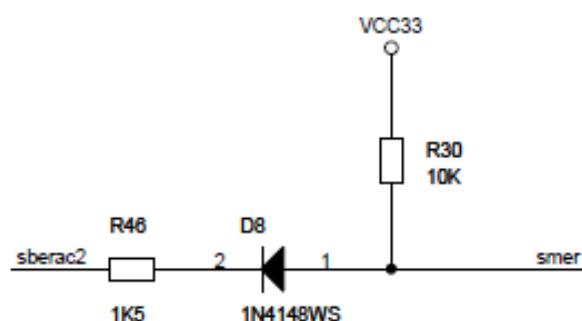
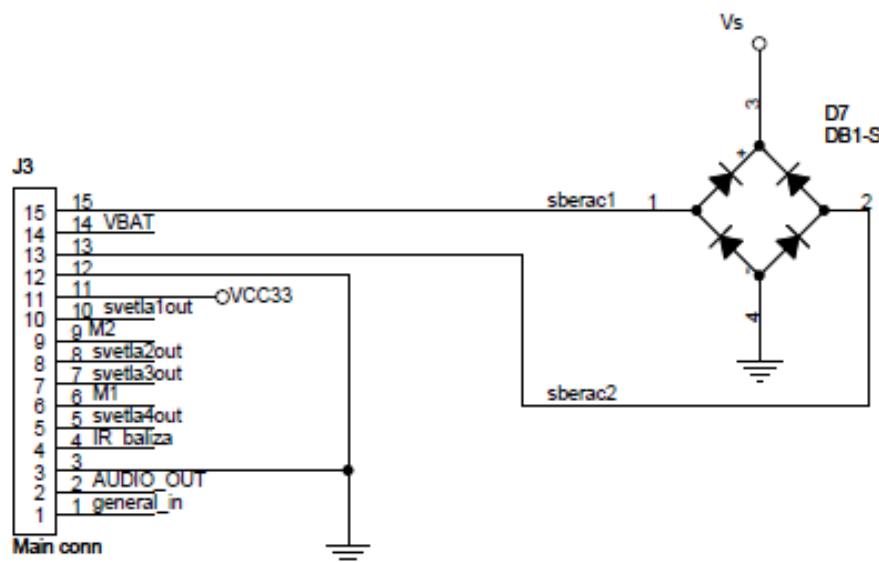
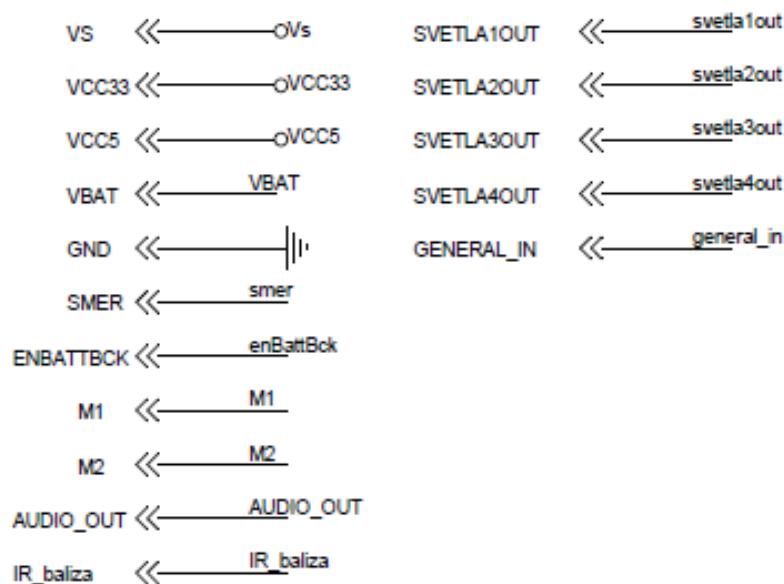
# Literatura

- BLUETOOTH SIG (2001), *Specification of the Bluetooth System ver. 1.1*, Bluetooth.org
- HAGEN, R. (2002), ‘Deciphering and understanding the .sms files in msts’, [http://www.msts.steam4me.net/tutorials/sms\\_doku.html](http://www.msts.steam4me.net/tutorials/sms_doku.html).
- HAJNÝ, M. (2006), *Diplomová práce – Zvukový funkční blok pro vozidlové simulátory*, ČVUT Praha
- HÖNIG, M. (2008), *Bakalářská práce – Řízení modelu lokomotivy*, ČVUT Praha
- HÖNIG, M. (2009), *Semestrální práce – Modelová lokomotiva - LQ regulace*, ČVUT Praha
- LUNDGREN, A. (2009), ‘Choosing between an arm7 and a cortex-m3 processor’, [http://www.edn.com/article/459352-Choosing\\_between\\_an\\_ARM7\\_and\\_a\\_Cortex\\_M3\\_processor.php](http://www.edn.com/article/459352-Choosing_between_an_ARM7_and_a_Cortex_M3_processor.php).
- MAXIM (2005), *MAX5460-MAX5468 32-Tap FleaPoT<sup>©</sup>, 2-Wire Digital Potentiometers*.
- MICROCHIP (2007), *MCP73811/MCP73812 Datasheet*.
- MITCHELL, M. (2000), *MSP430 - Using PWM Timer\_B as a DAC*, Texas Instrumenst
- PAVLÍŠTA, D. (2007), *Bakalářská práce – Řízení modelu lokomotivy*, ČVUT Praha
- ST MICROELECTRONIC (2005), *TDA7233/TDA7233D Datasheet*.
- TEXAS INSTRUMENTS (2009a), *TPS2556/TPS2557 Datasheet*.
- TEXAS INSTRUMENTS (2009b), *TPS6211x Datasheet*.
- TEXAS INSTRUMENTS (2010), *TPS63020/TPS63021 Datasheet*.

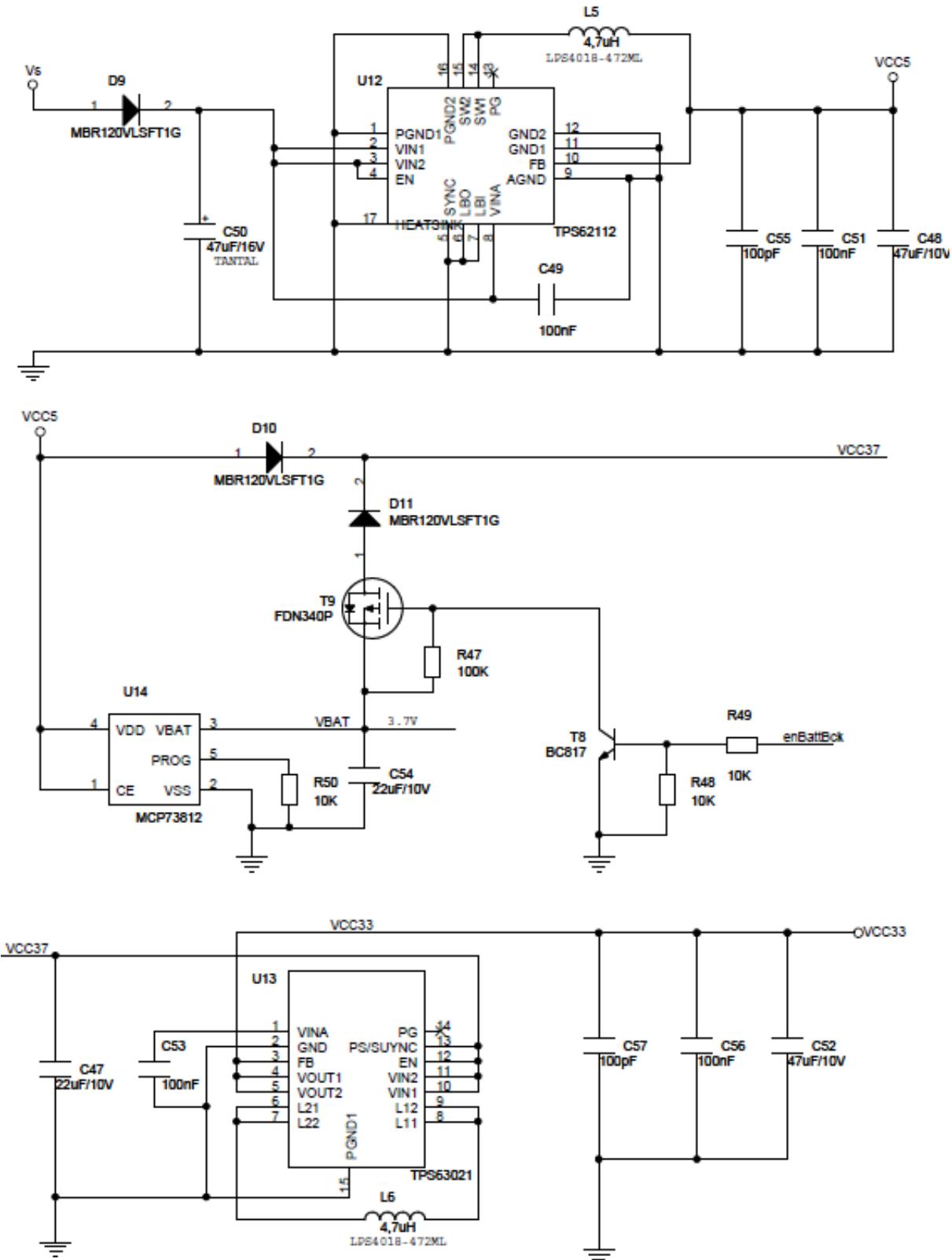


## Příloha A

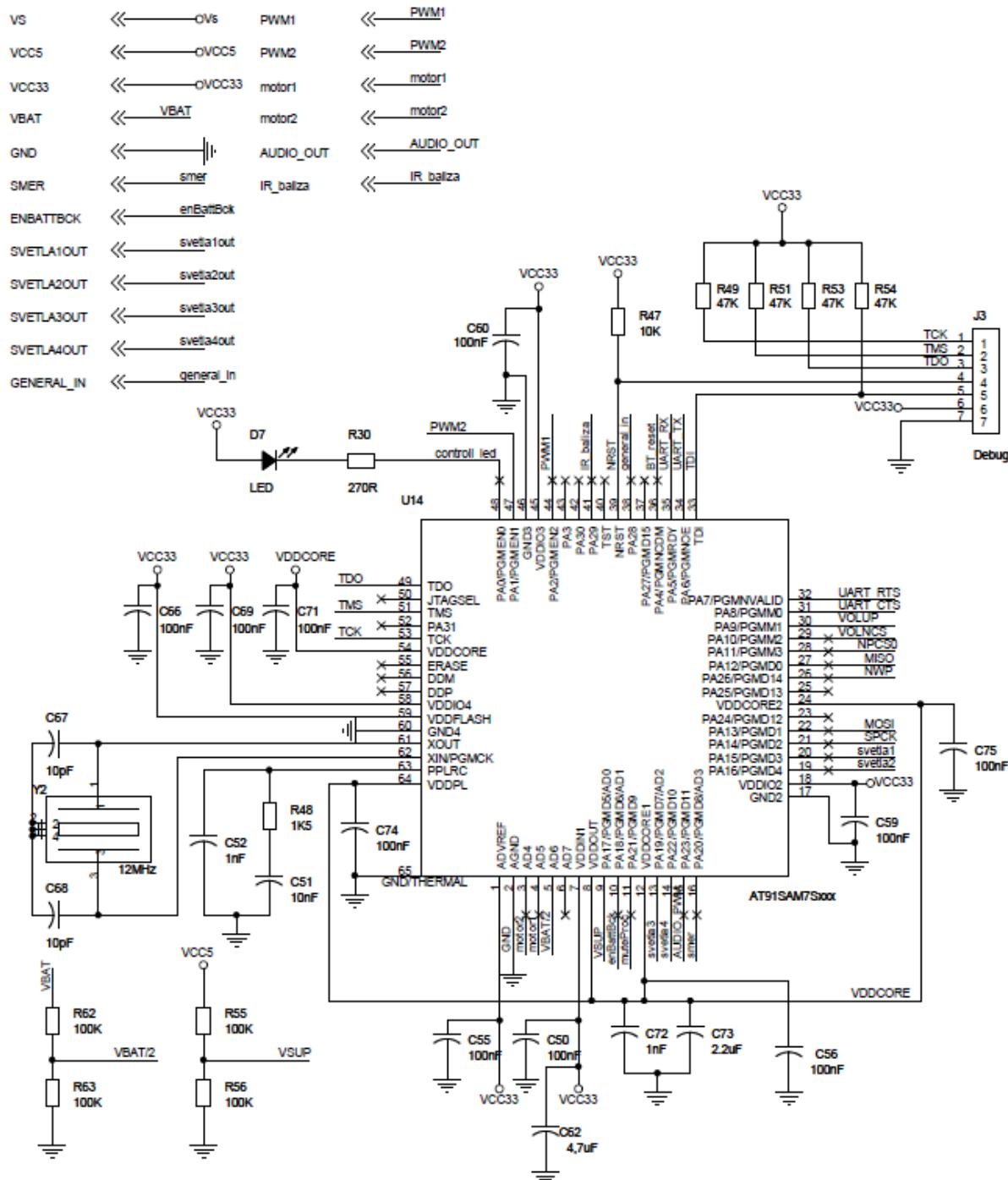
### Kompletní elektrické schema řídicího modulu



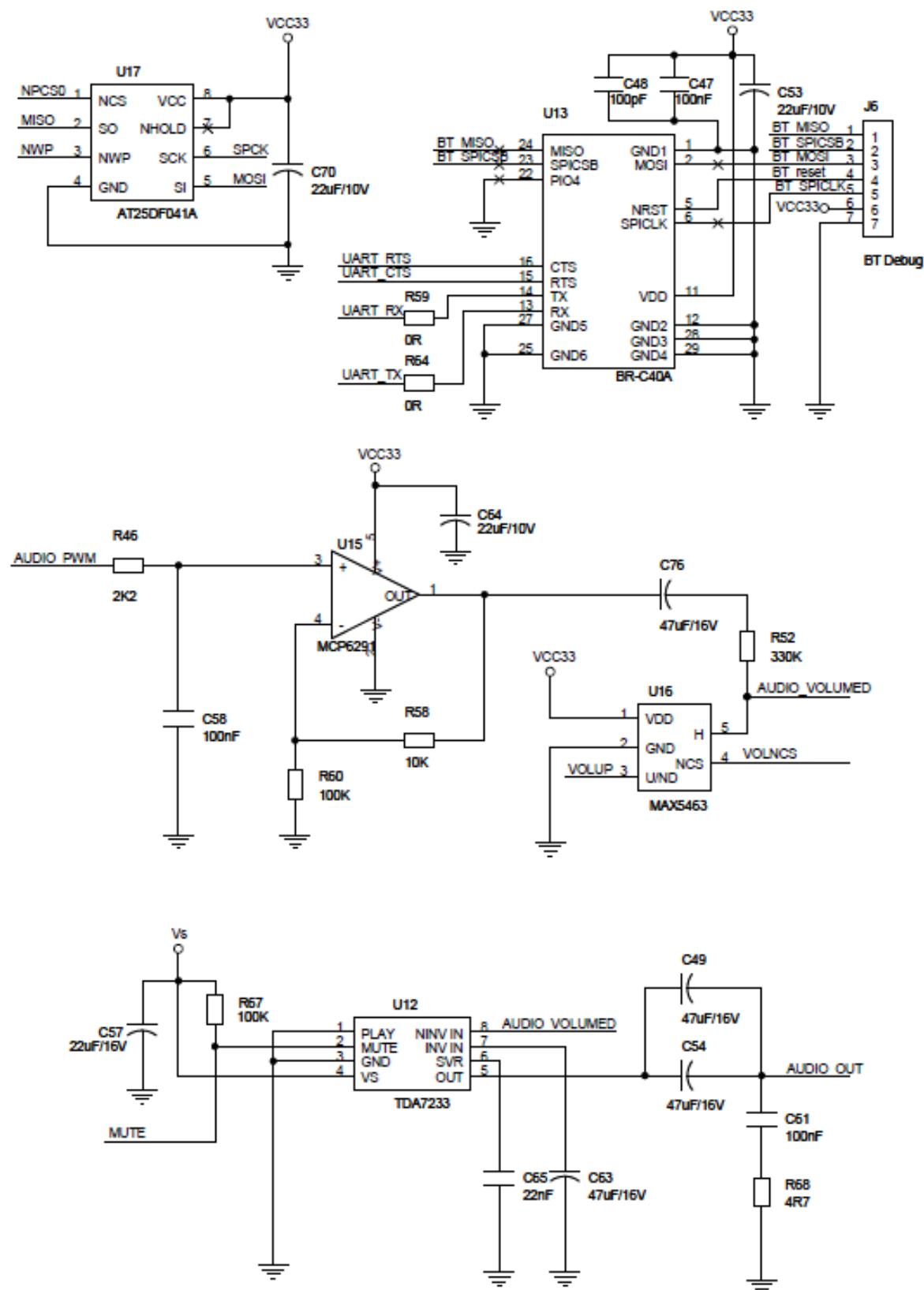
Obrázek A.1: Schema zdrojové části řídicího modulu - část 1



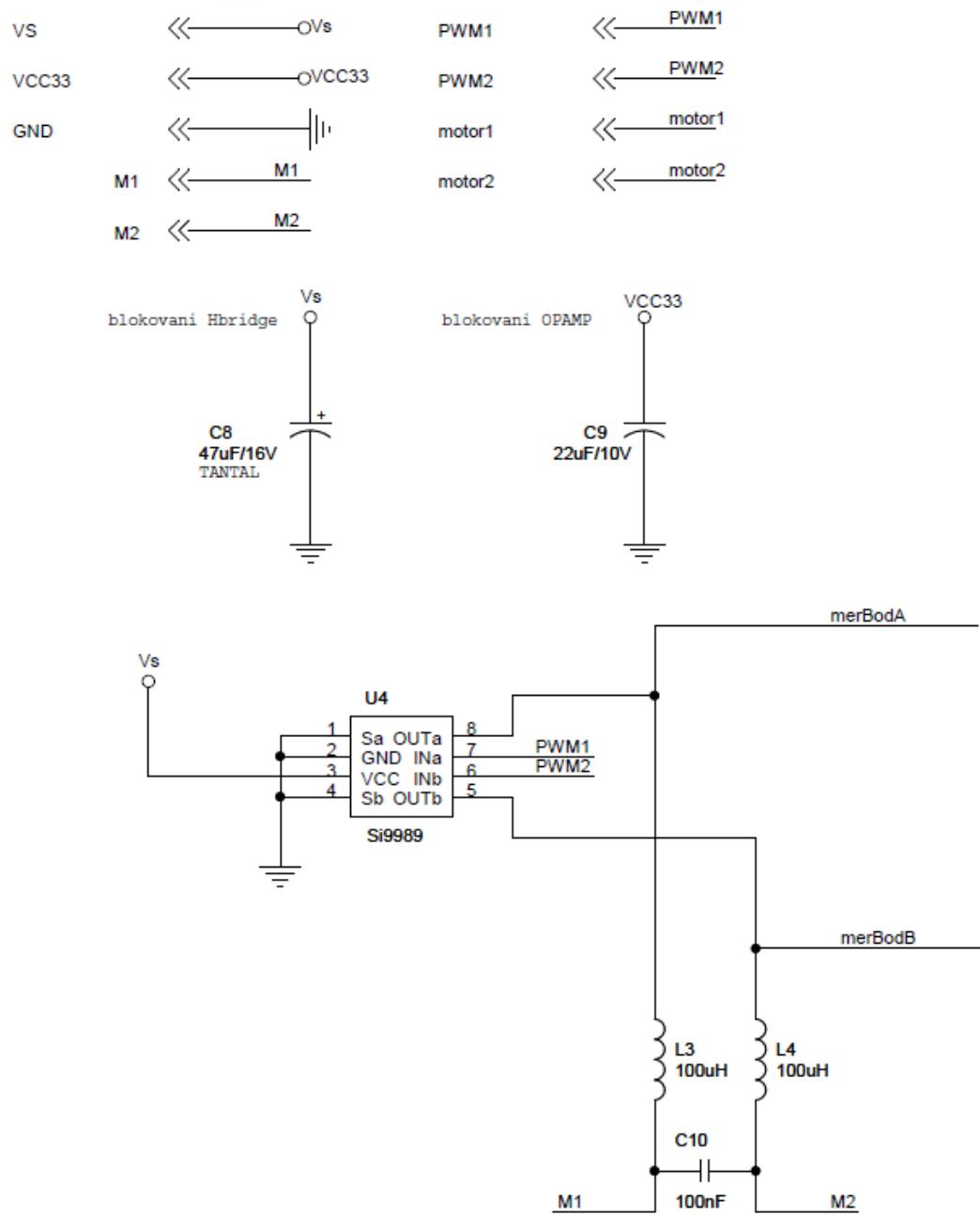
Obrázek A.2: Schema zdrojové části řídicího modulu - část 2



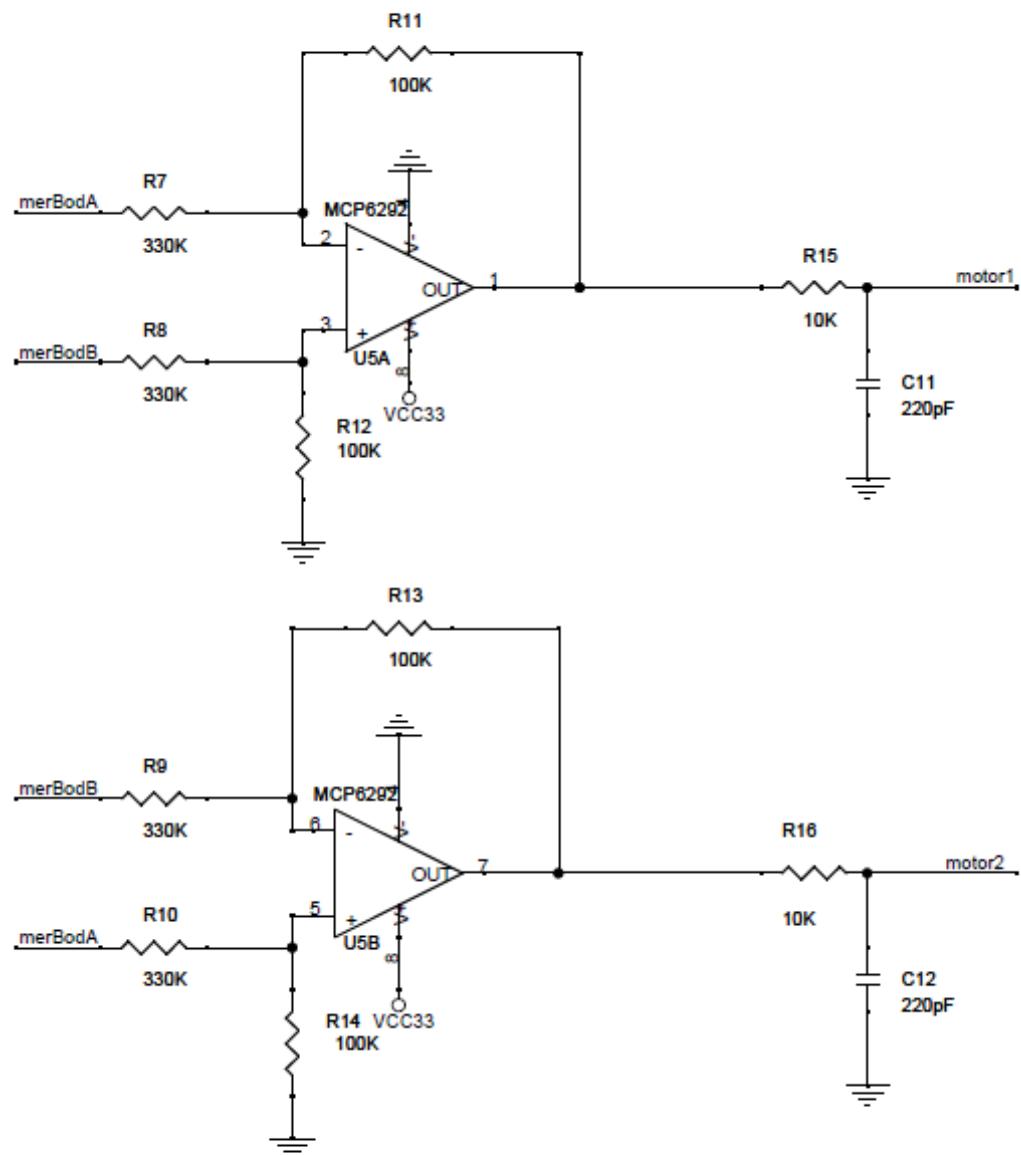
Obrázek A.3: Schema procesorové části řídicího modulu - část 1



Obrázek A.4: Schema procesorové části řídicího modulu - část 2

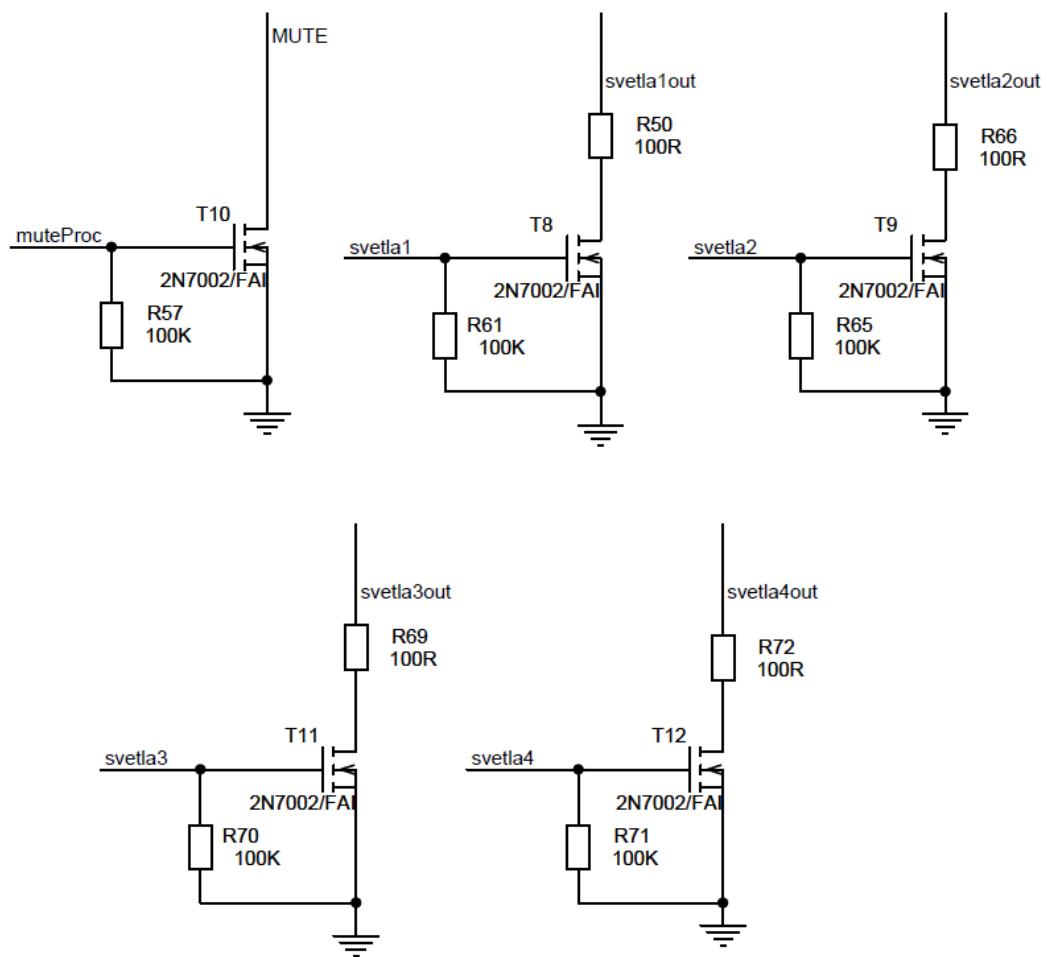


Obrázek A.5: Schema motorové budící části řídicího modulu - část 1



Obrázek A.6: Schema motorové budící části řídicího modulu - část 2

VIII PŘÍLOHA A. KOMPLETNÍ ELEKTRICKÉ SCHEMA ŘÍDICÍHO MODULU



Obrázek A.7: Schema procesorové části řídicího modulu - část 3

## Příloha B

### Seznam použitého software

- Adobe Acrobat Reader - prohlížení souborů .pdf, export obrázků z .pdf
- CAN Analyser - vyčítání dat z Bluetooth serveru během testování komunikace s řídicím modulem
- Gerbv - kontrola vygenerovaných výrobních dat prošných spojů
- GhostScript, GhostView - prohlížení souborů formátu .ps a jejich konverze do .pdf
- Mathworks Matlab 2008b - experimentování se zvukovými modely, generování grafů z naměřených hodnot
- MikTeX 2.8 - překladač zdrojového kódu L<sup>A</sup>T<sub>E</sub>X
- Open Office 3.2 - tvorba a převod obrázků do formátu .eps
- OrCAD 16 - návrh elektrických schemat a plošných spojů
- Rowley CrossStudio 2 for ARM - vývoj SW pro mikroprocesory s jádrem ARMX
- TeXnicCenter - editor zdrojového kódu pro L<sup>A</sup>T<sub>E</sub>X



# Příloha C

## Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy zdrojové kódy a záznamy z měření.

- Firmware: Obsahuje tři podadresáře s projekty Rowley CrossStudio pro různé verze vzniklých prototypů řídicího modulu
- Měření: Zdrojová data s naměřenými hodnotami, Matlab skripty generující grafy z dat
- Software: Volně dostupné verze software nutné pro zobrazení některých přiložených podkladů
- Výrobní dokumentace: Výrobní podklady pro tištěné spoje řídicího modulu
- Zvukový model: zvukové vzorky a Matlab skripty použité při rekonstrukci zvukového modulu a jeho simulaci