



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mikulics** Jméno: **Štefan** Osobní číslo: **420040**
Fakulta/ústav: **Fakulta elektrotechnická**
Katedra/ústav: **Katedra elektrotechnologie**
Studijní program: **Elektrotechnika, energetika a management**
Studijní obor: **Technologické systémy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Řízení modelu

Název diplomové práce anglicky:

Model control

Pokyny pro vypracování:

Seznamte se s modelem v laboratoři T2:E1-3c.
Navrhněte rekonstrukci řízení modelu s prvky systému DCC XpressNet.
Vyberte komponenty a zabudujte je do modelu.
Upravte ovladač s vývojovou deskou STM Discovery Kit.
Proveďte oživení a ověření funkčnosti systému.

Seznam doporučené literatury:

Mikulics, Š: Bakalářská práce, ČVUT FEL, 2016
Pemer, J: Diplomová práce, ČVUT FEL, 2005
DCCWiki, Informační portál, www.dccwiki.com

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Karel Künzel, CSc., katedra elektrotechnologie FEL

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

Datum zadání diplomové práce: **16.01.2018**

Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **30.09.2019**

Ing. Karel Künzel, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra elektrotechnologie



Diplomová práce

Řízení modelu

Bc. Štefan Mikulics

Vedoucí práce: Ing. Karel Künzel, CSc.

Studijní program: Elektrotechnika, energetika a management, Magisterský

Obor: Technologické systémy

19. srpna 2018

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Karlu Künzelovi, CSc. za jeho cenné rady a poskytnutí všech prostředků potřebných k realizaci této diplomové práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Soběslavi dne 17. 8. 2018

.....

Abstract

The Diploma thesis deals with realization of a XpressNet driver that allows to control model railways.

The thesis follows up to the Bachelor thesis that deals with a prototype of XpressNet driver. The Diploma thesis includes a modernization of current model railway in the laboratory of T2:E1-3 to standards of XpressNet and creation of a full-working driver.

Key words

XpressNet bus driver, UART, RS485, model railway

Abstrakt

Diplomová práce se zabývá realizací ovladače sběrnice XpressNet, který umožňuje ovládání modelové železnice.

Práce navazuje na bakalářskou práci, která se zabývala vyrobením základního prototypu ovladače sběrnice. Součástí diplomové práce je modernizace modelu kolejiště v laboratoři T2:E1-3 katedry elektrotechnologie na standard XpressNet a zhotovení plně funkčního ovladače.

Klíčová slova

Ovladač sběrnice XpressNet, UART, RS485, model kolejiště

Obsah

1	Úvod	1
1.1	Cíl diplomové práce	1
1.2	Model kolejiště	1
1.3	Prototyp XpressNet ovladače	3
2	Modernizace kolejiště	5
2.1	Rozdíl mezi analogovým a digitálním řídicím systémem	5
2.2	Schéma DCC systému	6
2.3	Ovladač	6
2.4	Příkazová stanice	7
2.5	Lokodekodér	9
2.6	Souhrn modernizace	11
3	XpressNet	13
3.1	Formát komunikace	14
3.2	Časování XpressNet	15
3.3	Připojení XpressNet zařízení	16
3.4	Nevyžádané zprávy	17
3.5	Příklady vysílaných paketů	17
4	Hardware použitý v ovladači	19
4.1	STM32F429 - DISC1	19
4.2	Převodník TTL na RS-485	21
4.3	Spínaný regulátor z 12 V na 5 V	21
4.4	Inkrementální spínač s tlačítky	22
4.5	Optočlen	22
4.6	Plošné spoje	23
4.7	Pouzdro	25
4.8	Kompletní zařízení	26
5	Software ovladače	27
5.1	main.c	27
5.2	communication.c	28
5.3	pins.c	30
5.4	rs485.c	31
5.5	touch.c	31

5.6	gui.c	32
6	Obsluha ovladače	35
6.1	Popis ovladače	36
6.2	Vývojový diagram grafického rozhraní	36
6.3	Hlavní menu	37
6.4	Programovací režim	37
6.5	Nastavení	39
6.6	Inkrementální spínač s tlačítky	40
7	Ověření funkčnosti ovladače	41
7.1	Schéma testovacího pracoviště	41
7.2	Ověřování pomocí lokomotivy	42
7.3	Ověřování pomocí logického analyzátoru	43
7.4	Ověřování pomocí osciloskopu	49
8	Závěr	51
8.1	Modernizace kolejíště	51
8.2	Ovladač	51
8.3	Budoucí vývoj ovladače	52
A	Hlavní deska	55
A.1	Schéma zapojení	55
A.2	Výrobní data	57
B	Horní deska	59
B.1	Schéma zapojení	59
B.2	Výrobní data	61
C	Deska pro zdroj	63
C.1	Schéma zapojení	63
C.2	Výrobní data	65
D	Obsah příloženého CD	67
D.1	„Dokumentace“	67
D.2	„Plošné spoje“	67
D.3	„Pouzdro“	67
D.4	„Program“	67
D.5	„Text“	67

Seznam obrázků

1.1	Model kolejiště [10]	2
1.2	Schéma kolejiště [1]	3
1.3	Prototyp ovladače XpressNet s analyzátozem [1]	4
2.1	Schéma zapojení [3]	6
2.2	Příkazová stanice NanoX-S88	7
2.3	Blokové schéma zdroje pro NanoX-S88	8
2.4	Kompletní zdroj pro NanoX-S88	9
2.5	Lokodekodér	9
2.6	Schéma zapojení lokodekodéru	10
2.7	Schéma zapojení kolejiště	11
3.1	Asynchronní přenos [1]	13
3.2	Posloupnost paketu [1]	14
3.3	Konektory RJ-12 a DIN [1]	16
4.1	STM32F429 - DISC1 [16]	20
4.2	SN75176BP - Popis pinů [18]	21
4.3	LM2574 - Schéma zapojení [17]	22
4.4	Inkrementální spínač ZIPPY [4]	22
4.5	Schéma zapojení optočlenu	23
4.6	Hlavní deska seshora a zesponu	24
4.7	Horní deska seshora	24
4.8	Horní deska zesponu	25
4.9	Model vnitřku ovladače	25
4.10	Ovladač bez pouzdra	26
4.11	Kompletní ovladač	26
5.1	Vývojový diagram komunikace	29
6.1	Popis ovladače	35
6.2	Ovladač - Programovací režim	36
6.3	Ovladač - Hlavní nabídka a numerická klávesnice [1]	37
6.4	Ovladač - Programovací režim	38
6.5	Ovladač - Nastavení	40
7.1	Testovací pracoviště	42

7.2	Samostatné příkazové okno	43
7.3	Řada příkazových oken	43
7.4	Změna rychlosti a směru lokomotivy s pinem PF9	44
7.5	Změna rychlosti a směru lokomotivy s jinou adresou ovladače	44
7.6	Zastavení ovládané lokomotivy	45
7.7	Pozastavení ovládané lokomotivy	45
7.8	Zastavení všech lokomotivy	45
7.9	Oznámení o zastavení všech lokomotiv	46
7.10	Řada oznámení o zastavení všech lokomotiv	46
7.11	Čtení hodnoty CV registru	46
7.12	Zpráva o přechodu do servisního módu	47
7.13	Zaslání požadavku o odpověď příkazové stanice	47
7.14	Odpověď příkazové stanice	47
7.15	Zápis hodnoty do CV registru	48
7.16	Žádost o obnovení normálního provozu	48
7.17	Oznámení o obnovení normálního provozu	49
7.18	Sejmutý průběh RS-485 signálu	49
7.19	Sejmutý průběh DCC signálu	50

Seznam tabulek

2.1	Barevné značení lokodekodéru [2]	10
3.1	Význam 5. a 6. bitu [1]	15
3.2	Zapojení RJ-12 a DIN [1]	16
3.3	Příkazové okno [11]	17
3.4	Změna rychlosti a směru pro 27 rychlostních kroků [11]	18
3.5	Zastavení všech lokomotiv [11]	18
3.6	Oznámení o zastavení všech lokomotiv [11]	18
6.1	Tabulka na ovládání pomocí PLC [11]	40

Kapitola 1

Úvod

Diplomová práce se bude zabývat návrhem ovladače sběrnice XpressNet. XpressNet je sice sběrnice využívaná na modelová kolejiště, ale při pochopení této sběrnice člověk pochopí fungování průmyslových sběrnic, které pracují na podobném principu. Průmyslové sběrnice jsou v současnosti velmi důležitým nástrojem pro sběr, distribuci a vyhodnocení dat. Využívají se pro měření fyzikálních veličin pomocí senzorů nebo k řízení technologického systému.

1.1 Cíl diplomové práce

Cílem diplomové práce je vyvinout funkční ovladač sběrnice XpressNet, kterým se budou ovládat lokomotivy na kolejišti v laboratoři T2:E1-3 katedry elektrotechnologie. Při tvorbě tohoto ovladače se bude vycházet z mé bakalářské práce, kde se tento ovladač dostal do fáze prototypu. Jednalo se o ovladač a analyzátor, na kterých se simulovala komunikace standardu XpressNet. Tento prototyp nebylo možné připojit k reálnému systému XpressNet, a ani nebyl uživatelsky přívětivý. Bude tedy nutné nejen kompletně přepracovat program tohoto prototypu, ale také jeho hardwarové vybavení. Součástí této diplomové práce bude modernizace kolejiště v laboratoři T2:E1-3 katedry elektrotechnologie, protože obsahuje pouze analogové řízení lokomotiv a sensorový systém. Ani jedno z toho neodpovídá současnému digitálnímu řízení. Na konci této práce bude ovladač z hardwarové stránky zcela dokončen, ale ze stránky softwarové bude otevřen mnoha možnostem vylepšení a přizpůsobení podle uživatele. [1]

1.2 Model kolejiště

Během mé bakalářské práce se na modelu kolejiště nic nezměnilo, protože se tato práce zabývala hlavně sběrnicí XpressNet. Jak bylo už řečeno, aby bylo možné lokomotivy ovládat, tak je zde nutná modernizace kolejiště v laboratoři T2:E1-3 katedry elektrotechnologie. Modernizace se dotkne také lokomotiv, protože je nutné je osadit dekodéry, které pracují ve standardu Direct Command Control (DCC).

Kolejiště sestavil v roce 1988 Jan Štolba v rámci diplomové práce. Následně se kolejiště dále modifikovalo, ale nikde nebylo řádně zdokumentováno. V roce 2005 byl tento problém vyřešen diplomovou prací Jana Pernera, kde bylo kolejiště podrobně popsáno. Dále na kolejišti ve své diplomové práci Studie řízení technologického systému pracoval Ondřej Kuřík,

kteřý měl za úkol zhotovit jednoduchý ovladač XpressNet, ale kolejiště do digitální řízení převedeno nebylo. [1], [13]



Obrázek 1.1: Model kolejiště [10]

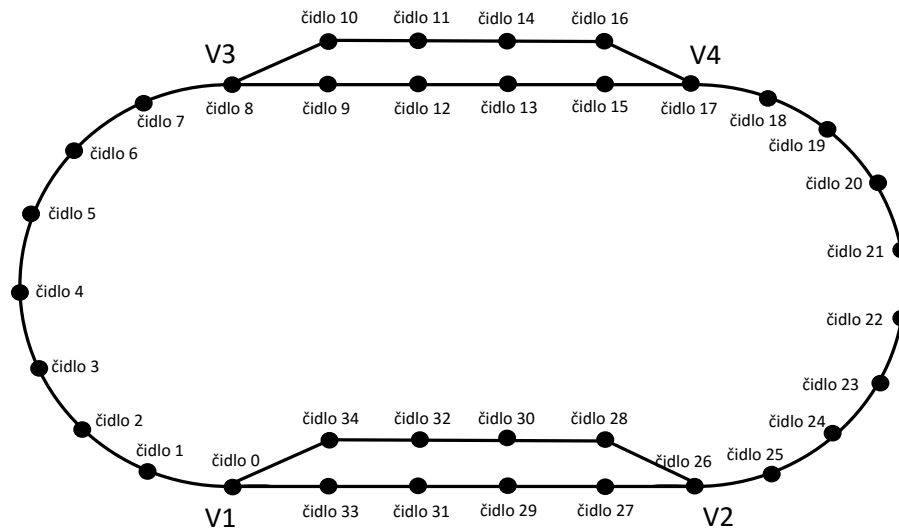
1.2.1 Současný stav modelu kolejiště

Dřevěná deska o rozměrech (10 x 120 x 200) cm je podkladem pro celé modelové kolejiště. Tvar kolejiště je ovál, který má zdvojené rovné části. Na tyto části se lokomotiva dostane pomocí 4 výhybek. Nad kolejemi se nachází 35 sensorových čidel. Schéma kolejiště včetně čidel je znázorněno na obrázku 1.2. Napájení je zajištěno pomocí 4 zdrojů: [10], [1]

- Napájecí zdroj pro výhybky
- Napájecí zdroj pro osvětlení kolejiště
- Napájecí zdroj 5 V
- Napájecí zdroj 12 V

1.2.2 Současný sensorový systém

Tento systém využívá soustavu fotosenzorů k určení aktuální pozice vlaku. Celkový počet fotosenzorů je 35 a je určen velikostí lokomotivy. S tímto počtem je vzdálenost senzorů mezi sebou 17,5 cm a tím není možné polohu vlaku ztratit. Jejich rozmístění bylo již zobrazeno na obrázku 1.2. [10], [1]



Obrázek 1.2: Schéma kolejiště [1]

1.2.3 Současné řízení modelu kolejiště

V současnosti se lokomotiva na kolejišti řídí pomocí infračerveného přenosu. Tento řídicí systém má tyto vlastnosti:

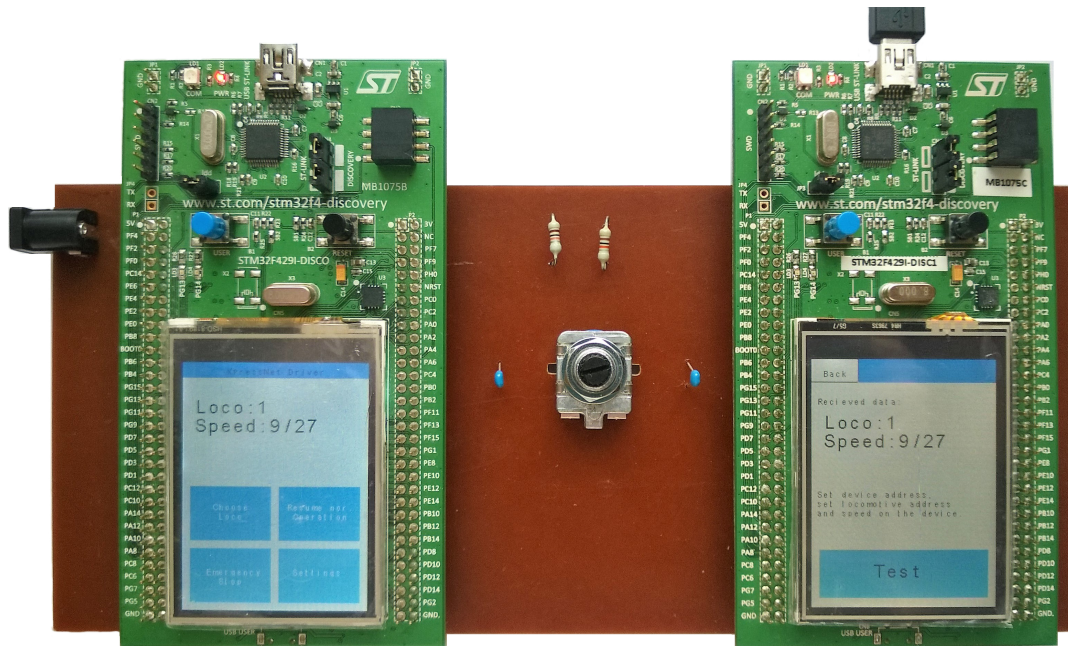
- Snímání polohy lokomotivy
- Řízení lokomotivy (jízda vpřed, jízda vzad, plynulý rozjezd a zastavení)
- Každá lokomotiva může být ovládána samostatně
- Možnost měnit směr jízdy lokomotivy výhybkami

Z pohledu řízení lokomotiv je tento systém velmi vydařený. Oproti klasickému analogovému řízení je zde největší výhodou možnost řídit lokomotivy samostatně a snímat jejich polohu. Problém nastává ve složitosti tohoto řízení a počtu senzorů. Model kolejiště je v tomto případě velmi jednoduchý a je zde potřeba 35 infračervených senzorů. V případě většího kolejiště by tento počet byl mnohem větší a celý systém byl velmi složitý. DCC je v porovnání s tímto řízením velmi jednoduchý, protože zde komunikace probíhá po kolejišti. [10]

1.3 Prototyp XpressNet ovladače

V mé bakalářské práci se vytvořil prototyp ovladače XpressNet s analyzátozem, jak je zobrazeno na obrázku 1.3. Jedná se o dva programovatelné kity STM32F429, které měly mezi sebou spojovací plošný spoj. K ovládání rychlosti se zde nacházel inkrementální spínač P-RE24. Tento prototyp posloužil hlavně k pochopení fungování sběrnice XpressNet, naučit se

programovat programovatelný kit STM32F429 a procvičit návrh plošného spoje. Prototyp ale nedokáže komunikovat s reálným systémem XpressNet. Jeden z problémů je, že komunikace probíhá ve standardu UART. Tento standard má jiné napěťové hladiny než RS-485, který využívá XpressNet. Dále napájení prototypu je 5 V, ale XpressNet má napájecí napětí 12 V. V neposlední řadě byl ovladač nepraktický, protože neměl ergonomii vhodnou ke klasickému používání. [1]



Obrázek 1.3: Prototyp ovladače XpressNet s analyzátozem [1]

Kapitola 2

Modernizace kolejiště

Jak bylo řečeno v první kapitole, současné kolejiště bude nutné modernizovat na digitální řídicí systém. V této kapitole bude vysvětleno, jak tento systém funguje, základní schéma tohoto systému a jaké komponenty jsou pro něj potřeba. Komponenty, které budou na modernizovaném kolejišti využity, zde budou podrobněji popsány.

2.1 Rozdíl mezi analogovým a digitálním řídicím systémem

V případě použití analogového systému řízení se lokomotiva ovládá pomocí regulátoru napětí. Při snižování napětí začne lokomotiva zpomalovat a při zvyšování napětí zrychlovat. Pokud chceme změnit směr jízdy stačí změnit polaritu napětí. Výhodou tohoto systému je jeho velká jednoduchost a k provozu nám stačí pouze regulační zdroj, který připojíme na kolejiště. Nevýhodou tohoto řešení je fakt, že regulujeme všechny lokomotivy, které napájí daný regulátor napětí.

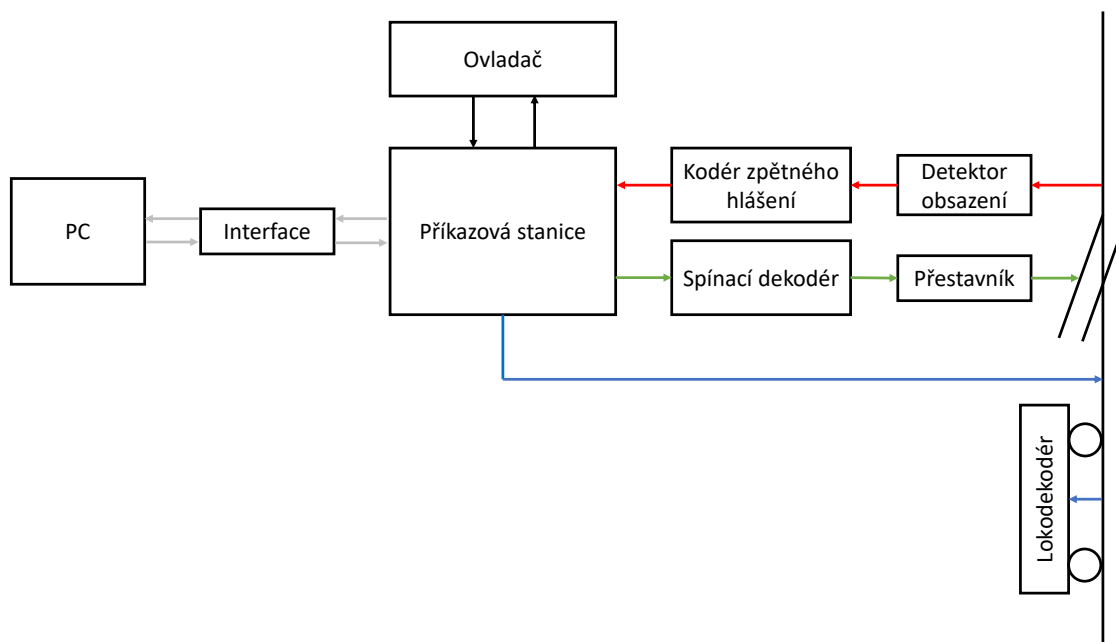
V současné době se využívá digitální řídicí systém, který vychází z myšlenky, že kromě napájení dokáže navíc posílat po kolejišti zakódované příkazy a informace. Tímto způsobem můžeme řídit jednotlivé lokomotivy, návěstidla, výhybky a další komponenty na kolejišti. Důležitou součástí tohoto řešení je dekodér, který musí být obsažen v každém komponentu kolejiště, který bude řízen. Hlavní výhody používání digitálního řídicího systému patří: [14]

- Možnost ovládání více lokomotiv a dalších zařízení na kolejišti
- Vlaky se chovají více realisticky - lze nastavit v dekodéru
- Možnost nastavit rychlostní kroky
- Ovládání speciálních funkcí lokomotiv - osvětlení, ozvučení
- Velké množství příslušenství

Mezi nevýhody patří samozřejmě vyšší náklady a složitější instalace oproti analogovému systému.

2.2 Schéma DCC systému

Základní komponenty pro instalaci DCC systému jsou příkazová stanice, ovladač a lokodekodér. Pomocí takového zapojení je možnost ovládat pouze několik lokomotiv. Pomocí ovladače se nastavuje příkaz, který se bude zasílat na kolejiště, a příkazová stanice ho převede na standard DCC. V případě složitějšího kolejiště je nutno zapojení doplnit ještě o spínací dekodéry a kodér zpětného hlášení. Spínací dekodéry slouží k ovládání výhybek, návěstidel nebo dalších komponentů. Kodér zpětného hlášení poskytuje zpětnou vazbu z kolejiště, například o pozici lokomotivy. V případě velmi složitých kolejišť, kde je potřeba využívat jízdních řádů pro jednotlivé lokomotivy, je vhodné použít rozhraní, pomocí kterého můžeme lokomotivy ovládat z PC. Vše je znázorněno na obrázku 2.1. Kvůli jednoduchosti modernizovaného kolejiště bude dostatečné základní zapojení. [3]



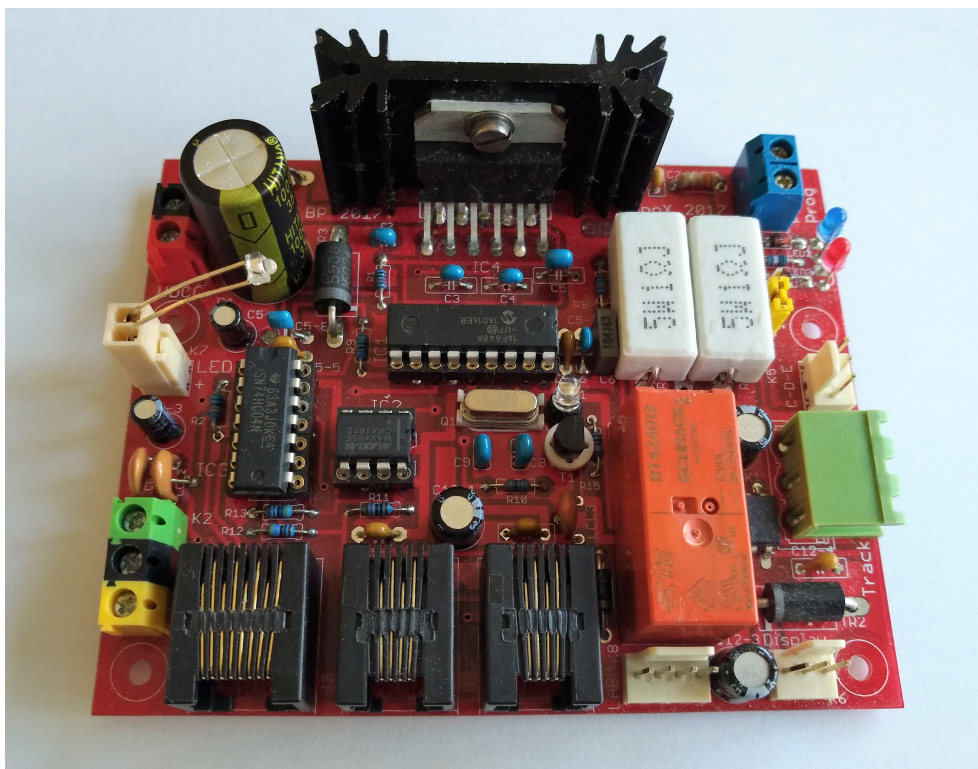
Obrázek 2.1: Schéma zapojení [3]

2.3 Ovladač

Ovladač bude fungovat na standardu XpressNet. V mé bakalářské práci jsem ovladač přivedl do fáze prototypu. Tento prototyp byl porovnán s originálním ovladačem Lenz LH100, ale nebyl zakomponován do kolejiště. Bude tedy nutné ovladač opatřit převodníkem TTL na RS485, konektorem RJ-12, stabilizátorem napětí a pouzdem, aby bylo možné ovladač pohodlně využívat a aby fungoval na jakékoli příkazové stanici, která podporuje standard XpressNet. Ovladač je zásadní částí této diplomové práce, a proto podrobnější informace budou další částí této práce.

2.4 Příkazová stanice

Současné kolejiště neobsahuje žádnou příkazovou stanici, která by dokázala vytvářet DCC signál, a proto bylo nutné o tuto stanici kolejiště rozšířit. Stanice lze zakoupit za cenu kolem 5000 Kč od společností Roco a Lenz. Obě stanice jsou velmi kvalitní a zaručují velkou spolehlivost na větší kolejiště. V případě kolejiště v laboratoři T2:E1-3 jsou tyto stanice zbytečně předimenzované, a proto byla vybrána příkazová stanice NanoX-S88.



Obrázek 2.2: Příkazová stanice NanoX-S88

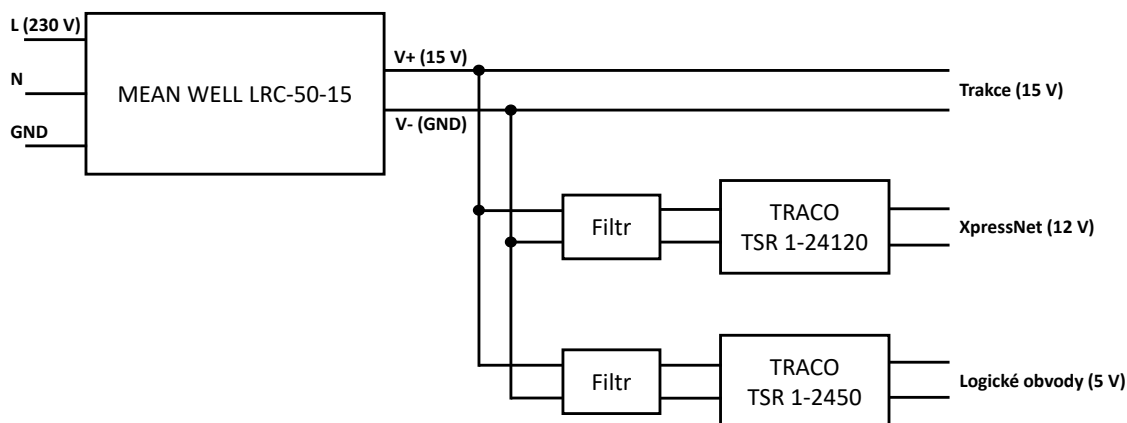
2.4.1 NanoX-S88

Původní stanice pochází od Paco Cañada, dále byla modifikována Jindrou Fučíkem, který na svých webových stránkách nabízí volně ke stažení návod ke složení, schémata a také .hex soubor, který se naprogramuje do procesoru PIC16F648A. Vylepšenou verzi této stanice prodává firma Digi-cz. Jelikož peněžní rozdíl mezi výrobou a zakoupením příkazové stanice byl minimální, tak tato stanice byla zakoupena. Příkazová stanice je znázorněna na obrázku 2.2. Stanice obsahuje sběrnici XpressNet a má tyto vlastnosti: [5], [9]

- Ovládání 16 lokomotiv s adresou 1 až 9999
- Ovládání 1024 výhybek a signálů
- Zapojení až 31 XpressNet ovladačů

- Podporuje 14, 28 a 128 rychlostních kroků, funkce FL a F1 až F12 pro každou lokomotivu
- Obsahuje konektor na programování a čtení DCC dekodérů
- Obsahuje konektor na kódér zpětného hlášení

2.4.2 Zdroj pro NanoX-S88



Obrázek 2.3: Blokové schéma zdroje pro NanoX-S88

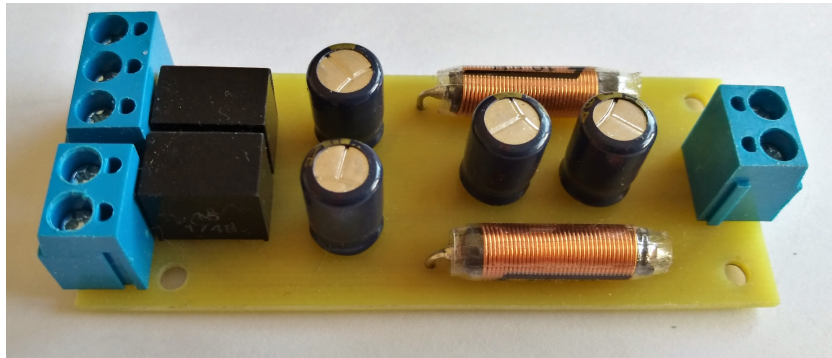
Pro napájení příkazové stanice jsou nutné 3 různé napájecí zdroje: [6]

- Trakční zdroj - 12 až 16 V
- Zdroj pro sběrnici XpressNet - 12 V
- Zdroj pro logické obvody příkazové stanice - 5 V

Dále je doporučeno, aby napájecí transformátor byl schopen poskytnout proud cca 3,5 A nebo více. Je to z důvodu nadproudové ochrany příkazové stanice, která vypíná v případě zkratu při 2,9 A. Zde se zase nabízí možnost zakoupit zdroj z webových stránek *digi.cz*, ale v tomto případě bylo mnohem výhodnější zdroj pro příkazovou stanici sestavit vlastní. [6]

Jako hlavní zdroj byl využit spínaný regulační zdroj MEAN WELL LRC-50-15. Tento zdroj může být napájen střídavým napětím 85 až 264 V a výstupní napětí je stejnosměrné 15 V, které je možno regulovat v rozsahu 13,5 V až 18 V. Zdroj zvládne výstupní proud až 3,4 A, a tím splňuje podmínku na sepnutí nadproudové ochrany. Celkový výkon je 51 W. Tento zdroj může sloužit bez jakýkoli úprav jako trakční zdroj.

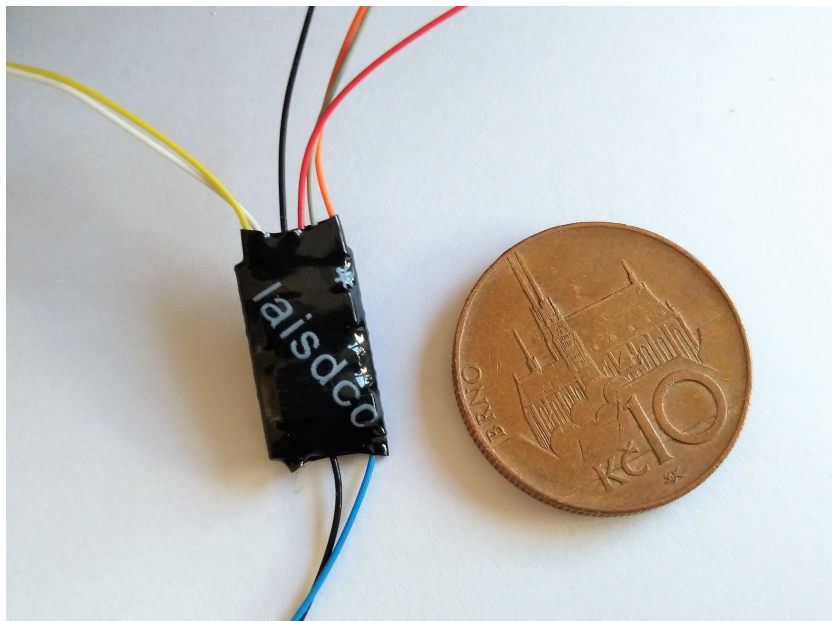
V případě napájení sběrnice XpressNet a logických obvodů příkazové stanice bylo nutné výstupní napětí zdroje upravit. K tomu byly využity dva regulační spínače od společnosti TRACO POWER - TSR 1-2450 a TSR 1-24120. Pro omezení rušivých vlivů je před spínače vložen vstupní PI filtr skládající se ze dvou kondenzátorů 4,7 μF a tlumivky 10 μH . Zjednodušené blokové schéma celého zdroje příkazové stanice je znázorněno na obrázku 2.3 a elektrické schéma je v příloze C. Finální podoba zdroje je zobrazena na obrázku 2.4.



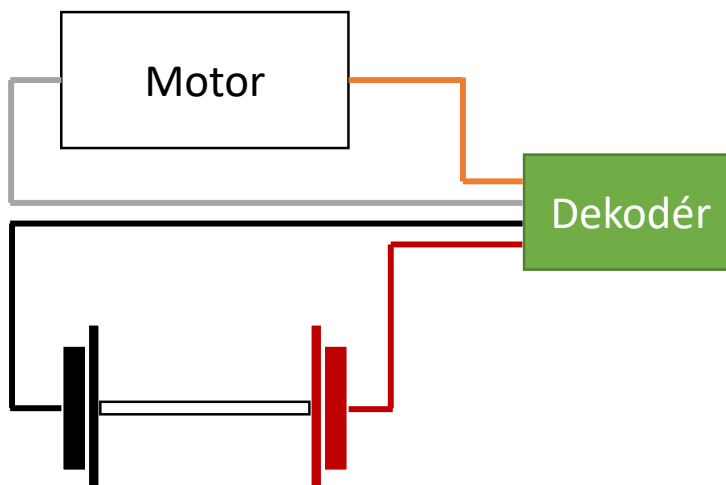
Obrázek 2.4: Kompletní zdroj pro NanoX-S88

2.5 Lokodekodér

Poslední částí základního zapojení je lokodekodér. Tento dekodér dokáže číst zakódované informace v signálu DCC a rozhoduje, jak se daná lokomotiva má chovat. Každý dekodér má vlastní adresu, vstup na DCC signál a výstupy na motory, osvětlení a další funkce lokomotivy. Rozměry takového dekodéru jsou asi 15x10x4 mm, protože se jedná o velmi malou součástku, tak jsem se rozhodl dekodér LaisDCC 6+2 z webových stránek dcdoma.eshop-zdarma.cz zakoupit. Lokodekodér je zobrazen na obrázku 2.5. [2]



Obrázek 2.5: Lokodekodér



Obrázek 2.6: Schéma zapojení lokodekodéru

Mezi základní vlastnosti dekodéru patří: [2]

- Kompatibilita s normou NEM (Normen Europäischer Modellbahnen)
- Možnost nastavení adresy 1 až 9999
- 14, 28 nebo 128 rychlostních kroků
- 4 nastavitelné funkce - základní nastavení: 2 vodiče jsou na přední a zadní světla, která se přepínají při změně jízdy
- Každá funkce může mít maximální odběr 100 mA

Vodiče lokodekodéru jsou označeny barevně podle standardu DCC. V tabulce 2.1 jsou popsány jednotlivé vodiče. V případě této diplomové práce bude zapojen pouze motor lokomotivy. Zapojení dekodéru je znázorněno na obrázku 2.6.

Tabulka 2.1: Barevné značení lokodekodéru [2]

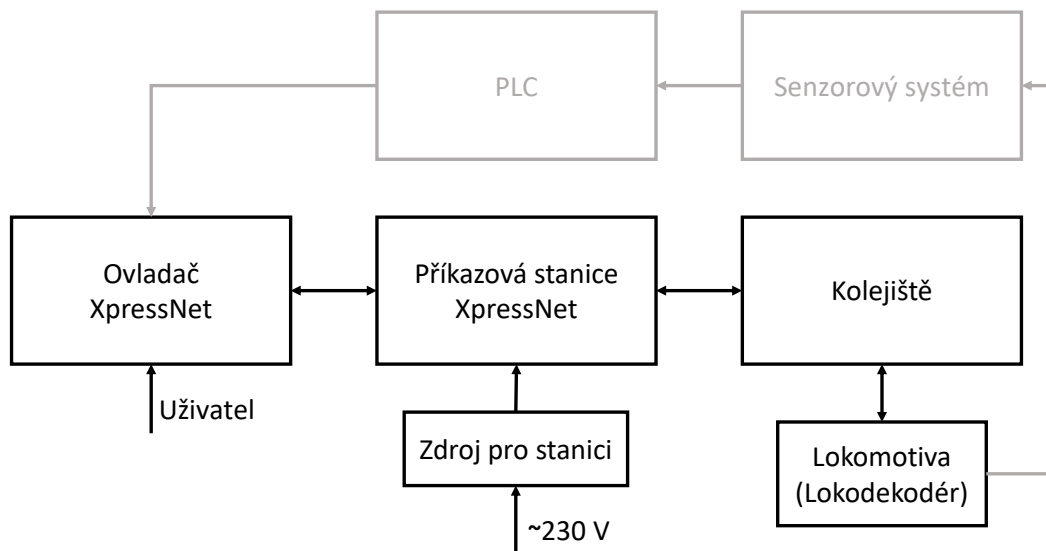
černá	Kolej - levá
červená	Kolej - pravá
modrá	Funkce (+)
šedá	Motor
oranžová	Motor
žlutá	Světla (vpřed) - F0
bílá	Světla (vzad) - F0
zelená	Funkce F1
fialová	Funkce F2

2.6 Souhrn modernizace

Na konci této kapitoly bych rád ještě shrnul celkovou modernizaci kolejiště. Zvolená koncepce je znázorněna ve schématu na obrázku 2.7. Komponenty, které jsou černé, jsou povinnou součástí systému XpressNet a šedé komponenty značí současný sensorový systém a PLC.

Modernizace zahrnuje zakoupenou stanici XpressNet, zdroj k příkazové stanici a zakoupený lokodekodér. Samozřejmě součástí celého systému je také ovladač, ale ten bude podrobně popsán v dalších kapitolách. V případě zapojení těchto komponent do kolejiště jsme již schopni ovládat lokomotivu.

Na kolejišti je již nainstalován sensorový systém a PLC - značeny šedě. Tento systém lze i nadále využívat pomocí signálů z PLC, které zadávají lokomotivám příkazy. Jedná se pouze o zapnutí lokomotivy a nastavení jejího směru. Tyto signály lze přivést do ovladače a převést je na standard XpressNet. Následná komunikace už probíhá v DCC. Z důvodu využívání sensorového systému bude na ovladači konektor, pomocí kterého lze ovladač propojit s PLC.



Obrázek 2.7: Schéma zapojení kolejiště

Kapitola 3

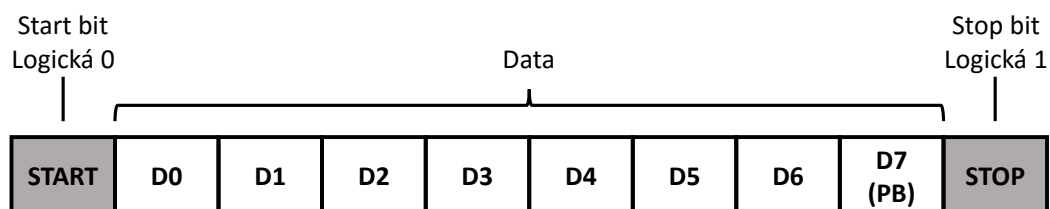
XpressNet

V této kapitole budou popsány základní vlastnosti XpressNetu, způsob komunikace a také příklady vysílaných paketů. XpressNet je síťový komunikační protokol používaný pro řízení modelových železnic. Tato sběrnice je využívána mezi ovladačem a příkazovou stanicí, která dále příkaz zakóduje do standardu DCC. Příkazová stanice dokáže v jeden okamžik komunikovat pouze s jedním ovladačem. K tomu napomáhá tzv. příkazové okno, které vysílá příkazová stanice a umožňuje tím ovladači zadat příkaz. Mezi hlavní výhody XpressNetu patří, že dokumentace je veřejně dostupná. Jsou v ní popsány všechny stavy, které mohou při komunikaci nastat. To umožňuje vytvořit si ovladač vlastní. [1], [11]

Komunikační protokol probíhá na standardu EIA RS-485, který využívá half duplex s jedním párem vodičů. Jedná se o asynchronní přenos, ke kterému není zapotřebí hodinového signálu. Příklad takové komunikace je znázorněn na obrázku 3.1. Komunikace začíná START bitem, který je v tomto případě logická 0. Následuje 8 datových bitů, kde poslední je paritní bit (PB), pomocí kterého se zjišťují chyby v komunikaci. Komunikace je ukončena STOP bitem, který nabývá hodnoty logické 1. [1], [11]

Vlastnosti komunikace pro XpressNet jsou: [1], [11]

- 1 start bit, 8-9 data bitů, 1 stop bit
- Přenosová rychlost 62 500 bit/s



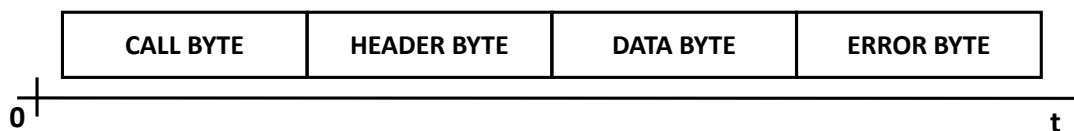
Obrázek 3.1: Asynchronní přenos [1]

3.1 Formát komunikace

Data posílaná mezi zařízeními XpressNet jsou ve formátu tzv. paketu. Tento paket se skládá ze čtyř bytů: [1]

- Call Byte
- Header Byte
- Data Byte
- Error Byte

Posloupnost paketu je zobrazena na obrázku 3.2. V případě, že vysílá příkazová stanice, komunikace začíná „Call Byte“, který je vždy pouze jeden. Pokud vysílá ovladač je tento byte vynechán. Po „Call Byte“ následuje „Header Byte“, který se zde vyskytuje také pouze jeden. Na „Header Byte“ navazuje „Data Byte“. Datových bytů zde může být více než jeden, záleží na konkrétním paketu. Komunikace je ukončena jedním „Error Bytem“, který se zde nachází z důvodu kontroly přenosu. [1], [11]



Obrázek 3.2: Posloupnost paketu [1]

3.1.1 Call Byte

Tento byte vysílá pouze příkazová stanice. Je rozlišen na několik druhů, které se rozpoznají pomocí 5. a 6. bitu. V tabulce 3.1 se nachází všechny možné typy „Call Byte“ podle těchto bitů.

Nejčastějším typem je příkazové okno, protože tímto způsobem dává příkazová stanice možnost ovladači, aby zadal svůj požadavek. V bitech 0 až 4 se nachází adresa ovladače, která může být v rozsahu 1 až 31. Pokud neprobíhá nějaká konkrétní komunikace, tak příkazová stanice posílá příkazová okna stále dokola se zvyšující se adresou ovladače. 7. bit je paritní bit.

Dalším častým typem „Call Byte“ je vysílání pro všechny ovladače. Celý byte může být buď s adresou ovladače, nebo bez ní. V případě, že se adresa nepřidává, je zpráva určena všem ovladačům. Jedná se o zprávy typu: normální provoz obnoven, nouzové zastavení, vypnutí napájení tratě a vstup do servisního módu. V druhém případě se jedná o zprávy, které jsou spojené se servisním módem. To jsou zprávy typu: datový byte nenalezen, příkazová stanice zaneprázdněná, odpověď na čtení CV registru atd.

Méně častými byty jsou potvrzení žádosti a zpětnovazební vysílání. Potvrzení žádosti příkazová stanice požaduje od ovladače, v případě chybného přenosu. Protože se jedná o

konkrétní ovladač, tak bity 0 až 4 jsou znova využité na adresu daného ovladače. Ovladač musí v tomto případě odpovědět na tuto žádost a komunikaci znovu opakovat. Zpětnovazební vysílání se v tomto případě nepoužívá vůbec, protože modernizované kolejiště nevyužívá žádné zařízení ve standardu DCC, které by příkazové stanici dávalo zpětnou vazbu. Tato zpětná vazba by byla dále zaslána všem ovladačům připojených do příkazové stanice. [11]

Tabulka 3.1: Význam 5. a 6. bitu [1]

6.bit	5.bit	Význam
0	0	Potvrzení žádosti
0	1	Zpětnovazební vysílání
1	0	Příkazové okno
1	1	Broadcastové vysílání

3.1.2 Header Byte

Tento byte se zde využívá jako hlavička, která obsahuje informace, podle kterých se předem rozpozná o jaký paket se bude jednat. Jedna z uložených informací je, kolik bude následovat datových bytů. Tato informace se nachází v dolních 4 bitech. Maximální počet následujících datových bytů je 15. Dále se v „Header Byte“ nachází informace o tom, zda se jedná o žádost, nebo identifikaci. Tato informace je uložena v horních 4 bitech. [11]

3.1.3 Data Byte

„Data Byte“ už bude obsahovat konkrétní data, která je potřeba přenášet. Může zde být uložena rychlost lokomotivy, adresa lokomotivy a mnohé další. K tomu, aby člověk mohl z datových bytů číst, je potřeba dokumentace XpressNet, kde jsou všechny možné pakety podrobně popsány. [11]

3.1.4 Error Byte

Poslední byte je „Error Byte“. Jedná se o kontrolu toho, jestli byla všechna data v pořádku přenesena. Tento byte obsahuje výsledek funkce XOR z „Header Byte“ a všech „Data Byte“. Při přijímání dat si zařízení XpressNet počítá vlastní „Error Byte“, které poté porovnává s tím přijatým. V případě, že se rovnají, tak je vše v pořádku, a komunikace může dále pokračovat. V případě, že se tyto dva byty nerovnají, tak dané zařízení odešle informaci o chybném přenosu. Přenos se poté bude znova opakovat. [11]

3.2 Časování XpressNet

Komunikace má danou přesnou časovou definici. To znamená, že ovladač XpressNet musí být schopen odpovědět příkazové stanici do 110 μs a příkazová stanice musí být schopná přijmout odpověď do 120 μs . V normálních podmínkách je rozmezí mezi příkazovými okny asi 400 až 500 μs , ale to není pro tuto diplomovou práci podstatné, protože v rámci diplomové

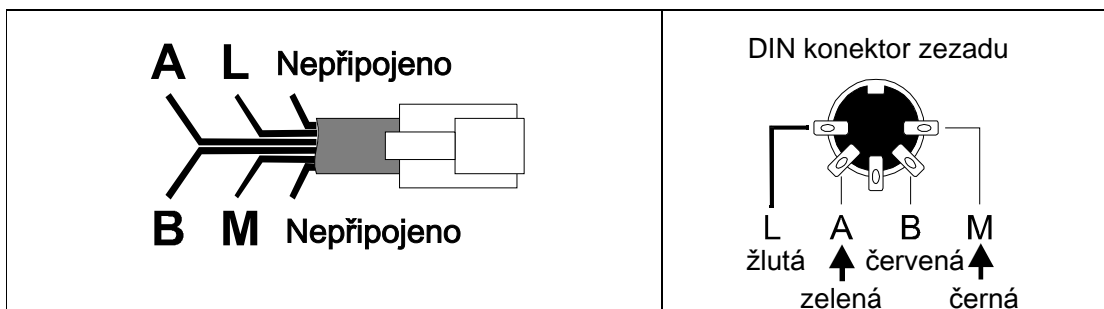
práce se jedná hlavně o realizaci ovladače. Již prototyp z mé bakalářské práce tuto podmínku s velkou rezervou splňoval a výsledný ovladač není výjimkou. Odpověď ovladače je cca 20 μ s. [1]

3.3 Připojení XpressNet zařízení

Standard XpressNet umožňuje zapojení pomocí dvou druhů konektorů. Jedná se o VIN konektor a RJ-12 konektor. V dnešní době se preferuje konektor RJ-12, a proto je využit také na ovladači i na příkazové stanici. Používají se 4 vodiče a jejich název a funkce jsou: [1]

- L – kladné napájecí napětí pro zařízení
- M – zem
- A – přijímací/vysílací vodič neinvertující
- V – přijímací/vysílací vodič invertující

Zapojení v obou konektorech je zobrazeno na obrázku 3.3 a v tabulce č. 3.2.



Obrázek 3.3: Konektory RJ-12 a DIN [1]

Tabulka 3.2: Zapojení RJ-12 a DIN [1]

Pin	RJ-12	DIN
Pin 1	"C"Kontrolní sběrnice	Nepřipojeno
Pin 2	Zem "M"	a Zem "M"
Pin 3	- RS-485 "B"	- RS-485 "B"
Pin 4	+ RS-485 "A"	+ RS-485 "A"
Pin 5	12 V "L"	12 V "L"
Pin 6	"D"Kontrolní sběrnice	Nepřipojeno

3.4 Nevyžádané zprávy

Jedná se o zprávy, které musí umět přijímat každé zařízení bez ohledu na příkazové okno. Mezi tyto zprávy patří: [1]

- Obnovení normálního provozu
- Vypnuté napájení tratě
- Nouzové zastavení
- Vstup do servisního módu
- Zpětnovazební vysílání
- Lokomotiva je ovládána jiným zařízením

Prvních pět zpráv je určeno pro všechna připojená zařízení, ale poslední dvě jsou určeny konkrétnímu zařízení, a proto musí obsahovat adresu zařízení. [1]

3.5 Příklady vysílaných paketů

Pro lepší představu se zde nachází několik základních paketů. Pakety, se kterými umí pracovat ovladač se nachází v dokumentaci programu, a všechny pakety se nacházejí v dokumentaci XpressNet. Dokumentace je velmi podrobná a přehledná. Obsahuje dvě části, které jsou věnovány paketům. Jedna část popisuje pakety, které vysílá příkazová stanice a druhá část popisuje pakety, které vysílá ovladač.

3.5.1 Příkazové okno

Příkazové okno vysílá pravidelně příkazová stanice a povoluje tím, aby ovladač mohl zadat svůj požadavek. Tabulka 3.3 znázorňuje formát příkazového okna. Pod písmeny A se nachází adresa ovladače, která se po každém vysílání mění v rozsahu od 0 do 31. Časová mezera mezi příkazovými okny je 150 μ s a v tomto okamžiku musí daný ovladač vyslat svůj požadavek. [11]

Tabulka 3.3: Příkazové okno [11]

	Call Byte
Bin	P10A AAAA
Hex	P+0x40+GA
Dec	P+64+GA

3.5.2 Změna rychlosti a směru lokomotivy

V případě, že se na ovladači změní rychlost, tak ovladač po přijetí příkazového okna odešle tento paket. To, že paket vysílá ovladač se rozezná jednoduchým způsobem a to tím, že tyto pakety začínají „Header Bytem“. Pakety vysílané z příkazové stanice začínají „Call Bytem“. Dále se v tomto paketu nachází byte „Identifikace“, který rozlišuje o jaký rychlostní krok se jedná. V tomto příkladě se jedná o 27 rychlostní stupeň. Rychlostní stupeň se nastavuje přímo v ovladači. Dále následují byty AH a AL. Jedná se o vyšší a nižší část adresy lokomotivy. Poslední datový byte obsahuje informaci o směru - R (0 je směr dopředu a 1 je směr dozadu) a o rychlosti - V. Paket je ukončen „Error Bytem“, který slouží jako kontrola, jestli komunikace proběhla v pořádku. Paket je zobrazen v tabulce 3.4. [11]

Tabulka 3.4: Změna rychlosti a směru pro 27 rychlostních kroků [11]

	Header B.	Identifikace	DB 1	DB 2	Data Byte 3	Error B.
Bin	1110 0100	0001 0001	AH	AL	R00VVVVV	XOR Byte
Hex	0xE4	0x11	AH	AL	RV	XOR Byte
Dec	228	17	AH	AL	RV	XOR Byte

3.5.3 Zastavení všech lokomotiv

Tento paket vyšle ovladač v případě stisknutí na tlačítko na displeji „Emergency Stop“. Jakmile tento paket příkazová stanice přijme, tak zastaví všechny lokomotivy na kolejišti a oznámí to všem ovladačům. Paket k zastavení všech lokomotiv je zobrazen v tabulce 3.5 a oznámení o zastavení všech lokomotiv je v tabulce 3.6. [11]

Tabulka 3.5: Zastavení všech lokomotiv [11]

	Header Byte	Error Byte
Bin	1000 0000	1000 0000
Hex	0x80	0x80
Dec	128	128

Tabulka 3.6: Oznámení o zastavení všech lokomotiv [11]

	Call Byte	Header Byte	Data Byte 1	Error Byte
Bin	0110 0000	1000 0001	0000 0000	0110 0001
Hex	0x60	0x81	0x00	0x81
Dec	96	129	0	129

Kapitola 4

Hardware použitý v ovladači

V této kapitole bude popsán všechnen použitý hardware, ze kterého je ovladač vyroben. Oproti mé bakalářské práci zde přibyly různé převodníky, aby ovladač správně komunikoval s příkazovou stanicí a dodržel standard XpressNet. Dále se zde nachází optočlen, který umožňuje komunikace s PLC. Tuto možnost žádný komerčně prodávaný ovladač neobsahuje.

Základ celého ovladače zůstal zachován. Byla znovu využita programovatelná vývojová deska STM32F429, protože stále splňuje požadavky na ovladač. Nyní už bude potřeba pouze jedna deska, a proto byla zvolena novější verze desky - DISC1.

4.1 STM32F429 - DISC1

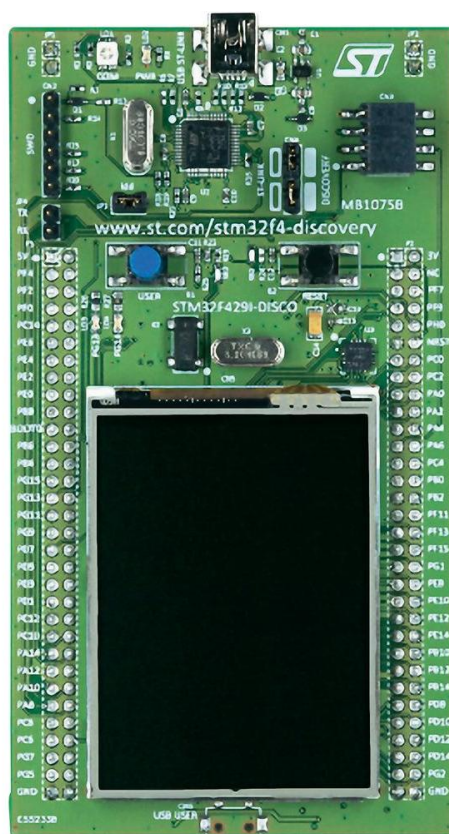
Jedná se o programovatelnou vývojovou desku od společnosti STMicroelectronics, která je zobrazena na obrázku 4.1. Mezi základní vlastnosti patří: [16], [1]

- ARM mikrokontrolér STM32F429ZIT6 s jádrem Cortex-M4
- Barevný TFT displej s úhlopříčkou 2,4" s QVGA rozlišením (320 x 240)
- Dotyková vrstva a její řadič
- Externí SDRAM paměť o velikosti 64 Mb
- 2 uživatelské LED, 1 uživatelské tlačítko a tlačítko Reset
- Programátor/debuger ST-LINK V2
- mbed - systém s možností cloudového připojení
- Virtuální COM port
- Podpora mass storage

Obvod má následující parametry:

- 32bitové jádro s jednotkou pro výpočet pohyblivé řádové čárky (FPU) s maximálním taktem 180 MHz

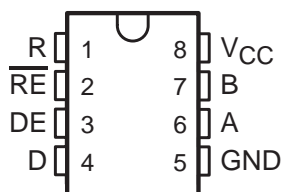
- 2 MB paměť FLASH, 256 + 4 kB paměť SRAM
- LCD-TFT řadič pro rozlišení až XGA (1024 x 768)
- Napájecí napětí 1,7 V až 3,6 V
- A/D převodník – až 24 kanálů
- 17 časovačů
- 3 sběrnice I2C, 6 sběrnic SPI
- 10/100 ethernet MAC
- Rozhraní pro kameru



Obrázek 4.1: STM32F429 - DISCO [16]

4.2 Převodník TTL na RS-485

Na desce STM4F429 se nacházejí pouze piny, které umí pracovat v hladině logické 1 a logické 0. Komunikace mezi ovladačem XpressNet a příkazovou stanicí probíhá pomocí standardu RS-485, kde napětí dosahuje kladných i záporných hodnot, proto zde bylo nutné přidat tento převodník. Byl vybrán integrovaný obvod Texas Instrumentals SN75176BP. Důležité také bylo, aby převodník zvládal rychlost komunikace XpressNet. V tomto případě má převodník zpoždění v řádech desítek ns, a proto bez problémů vyhovuje. [18]



Obrázek 4.2: SN75176BP - Popis pinů [18]

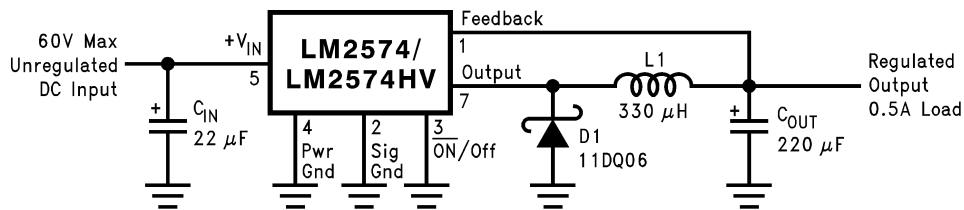
Na obrázku 4.2 se nachází popis pinů tohoto převodníku. Pin R znamená příchozí data v TTL, pin D znamená odchozí data v TTL. Negovaný pin RE a pin DE slouží k řízení datového toku. Pokud je na tyto dva piny přivedena logická 1, tak převodník převádí z TTL na RS-485. Pokud se na tyto piny připojí logická 0, tak převodník převádí z RS-485 na TTL. Na pinech A a B jsou připojeny vodiče, na kterých již probíhá komunikace ve standardu RS-485 a jsou zde přivedeny konektorem RJ-12. Piny Vcc a GND slouží k připojení napájení a země pro převodník. [18]

4.3 Spínaný regulátor z 12 V na 5 V

Standard XpressNet využívá napájení pro ovladače 12 V a deska STM32F429 potřebuje k napájení 5 V, a proto je nutné přidat zde napěťový regulátor. Pro tento účel byl vybrán LM2574N, který má tyto vlastnosti: [17]

- Výstupní napětí: 5 V
- Výstupní proud: 0,5 A
- Maximální napětí: 45 V
- Frekvence oscilátoru: 52 kHz
- Účinnost: 77 %

Tento regulátor je nutné doplnit vyhlazovacími kondenzátory a zpětnou vazbou. K tomu bylo využito referenční zapojení, které se nachází v dokumentaci tohoto regulátoru. Schéma tohoto zapojení se nachází na obrázku 4.3. [17]



Obrázek 4.3: LM2574 - Schéma zapojení [17]

4.4 Inkrementální spínač s tlačítky

Místo původního samotného inkrementálního spínače se zde nachází ZIPPY ANQ-I1B-O1B-S-Z. Jedná se o inkrementální spínač, který má po stranách 4 tlačítka a 1 tlačítko uprostřed. Tlačítka budou využita na rychlejší změnu adresy lokomotivy, zastavení lokomotivy a zapínání nebo vypínání speciálních funkcí lokomotivy. Celý inkrementální spínač má tyto vlastnosti: [4]

- Jmenovité napětí - 10 V
- Jmenovitý proud - 1 mA
- Počet pozic - 24



Obrázek 4.4: Inkrementální spínač ZIPPY [4]

Celý inkrementální spínač je napájen z desky napětím 3 V. V tomto případě nebylo nutné ke spínači nic dalšího přidávat, protože na STM32 lze nastavit vnitřní pull-down rezistor.

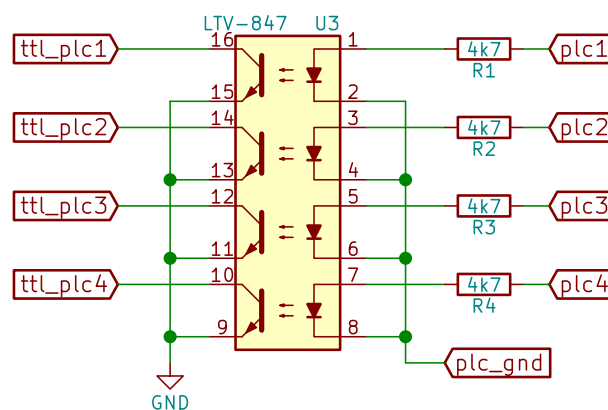
4.5 Optočlen

Ovladač bude také získávat informace z PLC. Aby nedošlo k poškození ovladače, je nutné mezi ovladačem a PLC galvanické oddělení. K tomuto účelu poslouží optočlen LITE

ON LTV847, který má tyto vlastnosti: [12]

- 4 optočleny s tranzistorem
- Izolační pevnost 5 kV

Schéma zapojení se nachází na obrázku 4.5. Jedná se o zapojení, které bylo využito v diplomové práci pana Pernera. Značky plc1 až plc4 jsou výstupy z PLC a plc_gnd je zem z PLC. Tyto výstupy jsou do ovladače přivedeny přes konektor RJ-45. Tento konektor byl využit z důvodu, aby nenastala záměna s konektorem XpressNet. Značky ttl_plc1 až ttl_plc4 jsou vstupy do STM32, GND značí zem ovladače.



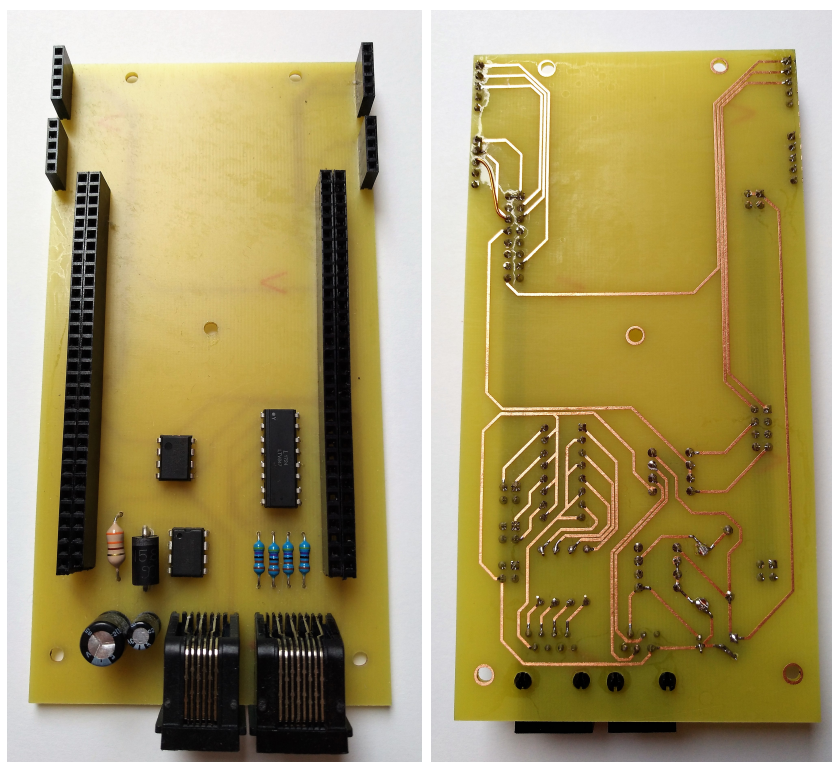
Obrázek 4.5: Schéma zapojení optočlenu

4.6 Plošné spoje

K propojení všech výše jmenovaných komponentů byly navrženy dva plošné spoje - hlavní deska a horní deska. Plošné spoje byly vyrobeny firmou Plošňáky.cz.

4.6.1 Hlavní deska

Hlavní deska slouží k usazení kitu STM32F429. Nachází se vše, co bylo výše vyjmenováno kromě inkrementálního spínače. Ve spodní části desky se nacházejí konektory vlevo RJ-12 a vpravo RJ-45. RJ-12 je využit pro napájení a komunikaci s příkazovou stanicí. Tento konektor byl vybrán s ohledem na standard XpressNet. Druhý konektor slouží k propojení s PLC. O něco výše se nacházejí regulátor, převodník, optočlen a k tomu přidružené komponenty zaručující jejich správnou funkčnost. V horní části desky se vyskytuje konektor, který propojuje hlavní desku s horní deskou. Aby bylo možné desku snadno připevnit do pouzdra, tak se zde nachází 5 děr na šrouby. Hlavní deska se zobrazuje na obrázku 4.6. Schéma a výrobní data se nachází v příloze A a na příloženém CD.



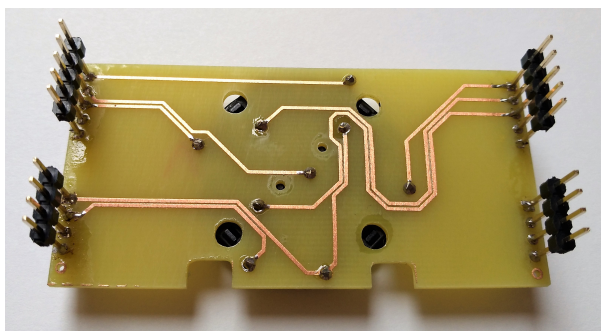
Obrázek 4.6: Hlavní deska seshora a zespodu

4.6.2 Horní deska

Horní deska obsahuje pouze inkrementální spínač a konektory k propojení s hlavní deskou. Tato deska byla navržena kvůli tomu, aby se inkrementální tlačítko mohlo umístit na výhodnější pozici. Kvůli tomu musela být kit STM32F429 lehce upravena. Byly odstraněny konektory CN2, CN3 a konektory CN4 a jumper JP3 byly přesunuty na druhou stranu kitu. Aby nebylo nutné odstraňovat také tlačítka USER a RESET, tak na desce se nacházejí dva výřezy. Deska je nachází na obrázkách 4.7 a 4.8. Schéma a výrobní data se nachází v příloze B a na přiloženém CD.



Obrázek 4.7: Horní deska seshora



Obrázek 4.8: Horní deska zespodu

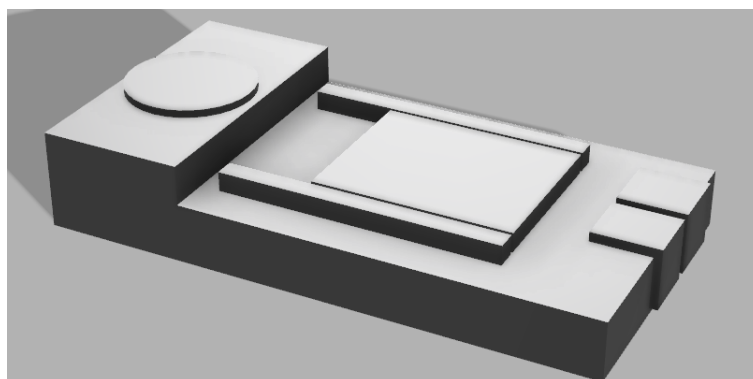
4.7 Pouzdro

Posledním a neméně důležitým prvkem ovladače je také jeho pouzdro, které jednotlivé komponenty ovladače ochrání a zvýší ergonomii. Protože vnitřní části ovladače měly velmi unikátní tvar, byl zvolen tisk pouzdra pomocí 3D tiskárny. V tomto případě bylo nutné celé pouzdro navrhnut v modelovacím programu. Výsledkem bylo pouzdro, do kterého vnitřní části ovladače zapadly naprosto přesně.

Pouzdro bylo modelováno v programu Autodesk Fusion 360, který nabízí studentům volně přístupnou verzi. Model byl uložen do formátu .stl, který je k nalezení na přiloženém CD. Pouzdro bylo vytištěno na 3D tiskárně v laboratoři T2:G1-123 katedry elektrotechnologie.

4.7.1 Model ovladače

Ještě před samotným modelováním pouzdra byl vyroben model vnitřní části ovladače, aby bylo možné jednoduše ověřit rozměry a správnost otvorů. Jednalo se pouze o velmi jednoduchý model, který obsahoval jen to nejdůležitější. Na obrázku 4.9 je tento model zobrazen. Model je k nalezení na přiloženém CD a může velmi dobře posloužit v případě, že se budoucí uživatel rozhodne navrhnout vlastní pouzdro.



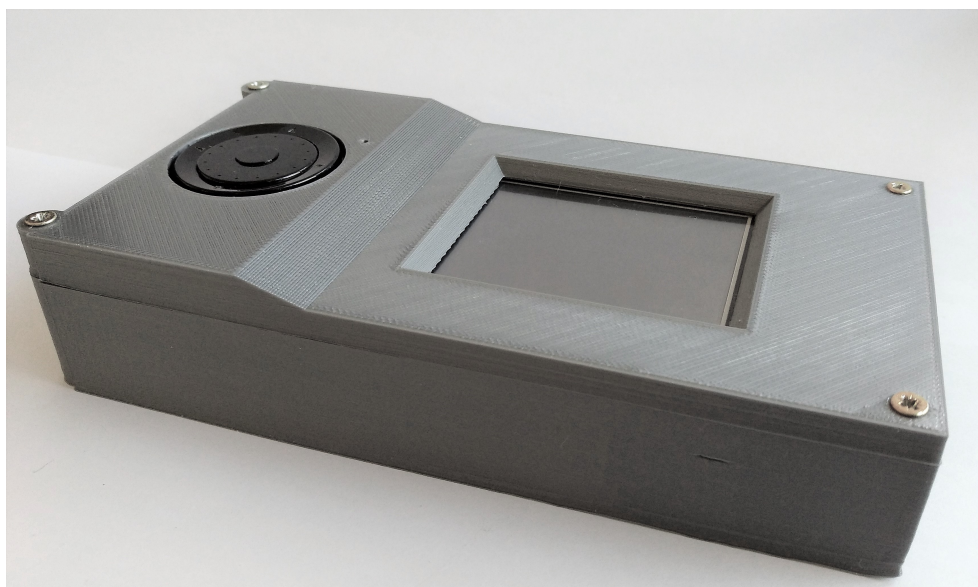
Obrázek 4.9: Model vnitřku ovladače

4.8 Kompletní zařízení

Na obrázcích 4.10 a 4.11 se nachází kompletní ovladač. Na prvním obrázku se jedná o vnitřek ovladače a na druhém je již ovladač v pouzdra.



Obrázek 4.10: Ovladač bez pouzdra



Obrázek 4.11: Kompletní ovladač

Kapitola 5

Software ovladače

V této kapitole bude podrobně popsán celý program tohoto ovladače. Kapitola bude rozdělena na jednotlivé .c soubory, kde budou popsány všechny nejdůležitější funkce a procedury. Kapitola je určena hlavně uživatelům, kteří by rádi program dobře pochopili a poté si ho mohli podle uvážení upravovat.

Program pro ovladač byl napsán v programovacím jazyce C. Bylo využito vývojové prostředí uVision Kel verze 5.0 ve Windows 10. Toto prostředí bylo výhodné hlavně díky velké podpoře programovatelných kitů od společnosti STMicroelectronics. Dále byly využity standardní knihovny na základní periférie desky. Podkladem pro tento program byl program z mé bakalářské práce, který byl kompletně refaktorován a byly zde přidány nové funkce. Program je nyní navíc podrobně okomentovaný způsobem, že z něj lze vygenerovat dokumentace pomocí DoxyGen. Tato dokumentace se nachází na přiloženém CD. Program je znovu pro větší přehlednost rozdělen na několik .c souborů. Ke každému .c souboru se zde nachází .h soubor (header file) obsahující prototypy funkcí, které jsou podrobně okomentované, a definice maker. Header soubory zde popsány nebudou. [1]

5.1 main.c

V tomto souboru se nachází hlavní funkce celého programu a to funkce main. Dále se zde nacházejí procedury, které se spouštějí v případě vnějších přerušení. Tyto přerušení mohou být vyvolány ze stisknutí tlačítka, příjmu dat přes UART, nebo příjmu logické hodnoty z PLC.

5.1.1 int main(void)

Na začátku této funkce probíhá konfigurace všech potřebných pinů, LCD, dotykové vrstvy a UARTu. Poté následuje nekonečná smyčka, ve které se nachází detekce dotyku na dotykové vrstvě a v případě zapnutého ovládní přes PLC kontrola pinů PB12, PB13, PD13 a PG3.

5.1.2 void EXTIx_IRQHandler(void)

Dále se zde nacházejí procedury, které se spouštějí v případě stisknutí tlačítka. Procedury *EXTI1_IRQHandler(void)* se spouští při stisknutí horního tlačítka. Procedura změny zna-

ménko u rychlosti a vykreslí novou rychlost na hlavní obrazovce. Procedura *EXTI2_IRQ_Handler(void)* patří levému tlačítku a při stisknutí se rychlost lokomotivy zmenší a znovu se vykreslí na hlavní obrazovku. *EXTI9_5_IRQHandler(void)* obsluhuje zbylá tlačítka. Tlačítko doprava zvýší rychlost lokomotivy, prostřední zastaví aktuálně ovládanou lokomotivu a spodní tlačítko pozastaví lokomotivu.

5.1.3 void UART5_IRQHandler(void)

Procedura *UART5_IRQHandler(void)* se spouští při příjmu dat přes UART. V proceduře se spustí další procedura z *communication.c*, která přijatá data uloží a následně je začne analyzovat.

5.2 communication.c

V tomto souboru se nachází celé jádro komunikace. Jsou zde procedury na analyzování dat a na odeslání určitého paketu. Konfigurace UART se nachází v souboru *rs485.c*, který bude popsán později. Pro lepší pochopení celé komunikace byl vytvořen vývojový diagram, který se nachází na obrázku 5.1.

5.2.1 void communication_control(void)

Tato procedura je zavolána při přerušení, které je vyvoláno příchodem dat na pin PD2. V případě, že se jedná o příkazové okno, tak ovladač zadá svůj požadavek. Pokud se o příkazové okno nejedná, tak data jsou uložena a poté analyzována procedurou *data_analyse(void)*.

5.2.2 void data_analyse(void)

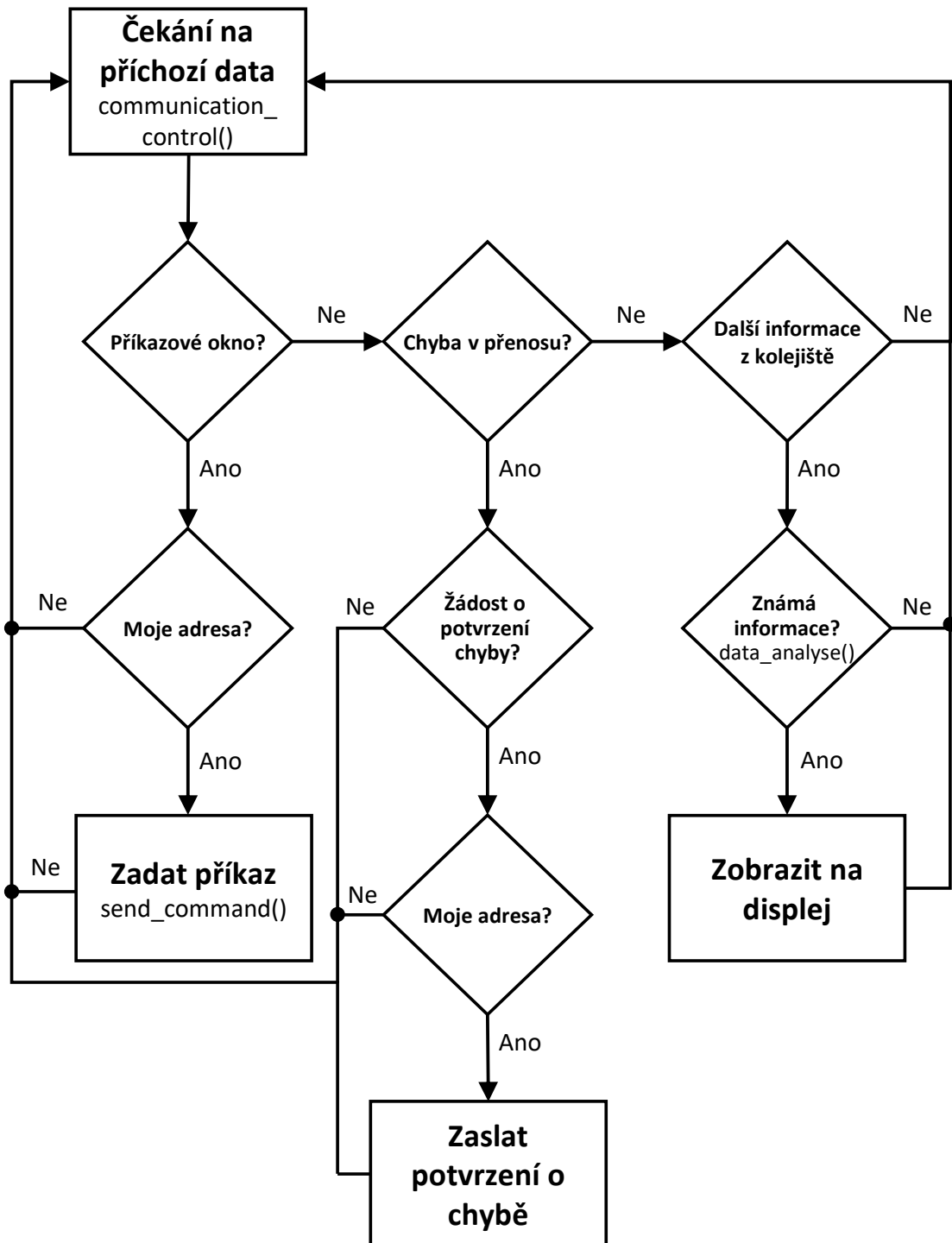
V případě, že jsou všechna přijatá data uložena, je spuštěna tato procedura, která data analyzuje. Na konci analýzy je zpráva zobrazena v horní části displeje. Jedná se o zprávy, které podle dokumentace XpressNet musí ovladač umět zobrazit (nouzové zastavení, vypnutí napájení tratě, zkrat atd.) a také o zprávy, které jsou spojeny se čtením a zápisem CV registrů (zapnutí servisního módu, obnovení normálního provozu, nenalezení registru atd.).

5.2.3 void send_command(void)

Tato procedura rozhoduje o tom, jaký příkaz se má poslat po přijetí příkazového okna. V této proceduře je velmi důležitá pomocná proměnná *cmd_help*, která pomocí podmínek *if* rozhodne, jaký příkaz se odešle. Tato proměnná se v průběhu programu mění podle toho, co na ovladači nastavujeme.

5.2.4 unsigned char error_byte_calculation(unsigned char data[10],char start, char end)

Tato funkce slouží k výpočtu „Error Byte“. Mezi parametry funkce patří *char data[10]*, kde se jedná o pole dat, ze kterého má být „Error Byte“ vypočítán. Dále jsou zde parametry



Obrázek 5.1: Vývojový diagram komunikace

char start a *char end*, které určují, jaké prvky pole *char data[10]* mají být součástí výpočtu. V některých případech bylo nutné první, nebo poslední prvek vynechat a pomocí těchto parametrů může být tato funkce univerzální.

5.2.5 Procedury k určitým paketům

Mezi tyto procedury patří *acknowledgement_response(void)*, *resume_operations_request(void)*, *emergency_stop(void)*, *CV_read_request(void)*, *service_mode_request(void)*, *CV_write_request(void)*, *locomotive_speed_direction(void)*. Všechny tyto procedury vysílají stejnojmenné pakety, které jsou k nalezení v dokumentaci XpressNet. Na začátku a na konci každé z těchto procedur se nachází nastavení piny PF9, aby se aktivoval mód vysílání v převodníku z TTL na RS-485.

5.3 pins.c

V tomto souboru se nachází konfigurace pinů pro inkrementální spínač, tlačítka a na vstupy z PLC. Také se zde nachází procedura, která kontroluje pohyb na inkrementálním spínači.

5.3.1 void incrementary_switch_config(void)

Tato procedura nakonfiguruje piny PD4 a PD5 pro inkrementální spínač. Jsou tedy nastaveny jako vstupy a na pull-down rezistor. Dále se zde nachází konfigurace přerušení pro pin PD4.

5.3.2 void incrementary_switch_control(int *max_speed, uint8_t *speed_changed)

Při změně na pinu PD4 je vyvoláno přerušení a tím je spuštěna tato procedura. Parametry procedury jsou ukazatel *int *max_speed*, který ukazuje na adresu maximální rychlosti a ukazatel *uint8_t *speed_changed*, který ukazuje na adresu proměnné, jestli byla rychlost změněna. Procedura si vždy pamatuje předchozí stav na pinech PD4 a PD5. Pomocí těchto stavů a aktuálního stavu pinů procedura určí, jestli se spínačem bylo točeno doprava, nebo doleva.

5.3.3 void plc_analyse_(void)

Tato procedura se využívá při zapnutí ovládání pomocí PLC, které je možné spustit v nastavení. Jedná se zde o hlídání hodnot na pinech PB12, PB13, PD13 a PG3. Podle aktuálních hodnot se tato procedura řídí tabulkou 6.1 a tím nastavuje rychlost a směr lokomotivám na adrese 1 a 2.

5.3.4 void buttons_config(void)

Tato procedura konfiguruje piny PC1, PC2, PF6, PF7 a PF8, které jsou propojeny s tlačítky u inkrementálního spínače. Piny jsou nastaveny jako vstupy a na pull-down rezistory. Dále je zde pro tyto piny aktivováno přerušení.

5.3.5 void plc_pins_config()

Poslední procedura z tohoto souboru konfiguruje piny PB12, PB13, PD13 a PG3, které jsou určeny pro příjem logické hodnoty z PLC. Piny jsou tedy nastaveny jako vstupy a na pull-up rezistory.

5.4 rs485.c

Soubor obsahuje konfigurace UART a pinu řízení toku, který je potřeba na ovládání převodníku TTL - RS-485.

5.4.1 void uart_config(void)

Tato procedura konfiguruje piny PC12 a PD5 a UART. Piny jsou nastaveny na alternativní funkci a na pull-up rezistory. Na těchto pinech se nachází UART5. UART5 je nakonfigurován na rychlost 62500 baud, délku slova 8 bitů a stop bit je logická 0. Dále je zde aktivováno přerušení pro tento UART.

5.4.2 void flow_control_config(void)

V této proceduře je nakonfigurován pin PF9, který řídí tok na převodníku TTL - RS-485. V případě, že ovladač potřebuje vysílat data, tak je na tento pin nastavena logická 1. Když jsou data přijímána je zde nastavena logická 0.

5.5 touch.c

V tomto souboru se nachází všechno k dotykovému displeji. Tento soubor nebyl oproti mé bakalářské práci z funkčního hlediska moc změněn. Bylo zde navíc přidáno ovládání pro nově vytvořenou obrazovku - programovací režim. Jinak se jednalo o drobné úpravy.

5.5.1 void touch_gui_config(void)

Tato procedura nakonfiguruje piny a rozhraní, které jsou potřebné pro dotykový displej. Dále se zde nachází kontrola, jestli byl dotykový displej správně nakonfigurován, pokud ne, tak se zobrazí chybová hláška. V tomto případě je nutné zařízení restartovat.

5.5.2 void touch_control(void)

Tato procedura je zavolána v případě dotyku na displeji. V této proceduře se nachází switch, který rozlišuje na jaké obrazovce se uživatel nachází. Podle toho zavolá proceduru, která náleží k dané obrazovce.

5.5.3 Procedury k určitým obrazovkám

Procedury *touch_menu(void)*, *touch_settings(void)*, *touch_programming_mode(void)* patří k obrazovkám hlavního menu, nastavení a programovacího módu. Procedury se liší pouze v jiném rozložení tlačítek, a proto jsou popsány dohromady.

5.5.4 void touch_keyboard(int* number, int max_number)

Tato procedura patří k obrazovce klávesnice. Je velmi podobná jako předchozí procedury, které patří k určitým obrazovkám, ale nachází se zde dva parametry, a proto je procedura popsána zvlášť. Mezi tyto parametry patří ukazatel *int* number*, který ukazuje na adresu čísla, které chceme na klávesnici zadat a poté změnit. Druhý parametr *int max_number* určuje maximální velikost čísla, která může být zadána. V případě, že zadané číslo je větší než maximum, neprovede se žádná změna a uživatel musí zadat číslo správně.

5.5.5 void write_keyboard_number(char number)

Jedná se o pomocnou funkci, která při stisknutí na klávesu při obrazovce klávesnice zobrazuje právě zadané číslo.

5.5.6 void delete_keyboard_number()

Jedná se o velmi podobnou proceduru jako byla poslední. Tato procedura při zmáčknutí tlačítka DEL vymaže zadané číslo na klávesnici.

5.5.7 void delay(void)

Tato procedura způsobuje zpoždění. Byla vytvořena, aby při klikání na obrazovku nevznikaly překlíky. Stejně využití má i u tlačítek, aby bylo zabráněno kliknutí více než jednou.

5.6 gui.c

V tomto souboru se nachází procedury určené ke grafickému uživatelskému rozhraní. Stejně jako u předchozího souboru se zde funkčně nic nezměnilo, byla zde přidána nová obrazovka - programovací režim a procedura vykreslení adresy lokomotivy.

5.6.1 void lcd_init_default(void)

Tato procedura nakonfiguruje vše potřebné k zobrazování na displeji. Jsou odtud volány další procedury, které jsou součástí standardní knihovny od STM. Kvůli větší přehlednosti v souboru main.c byly tyto procedury vloženy do jedné.

5.6.2 Procedury k určitým obrazovkám

Jedná se o procedury *gui_menu(void)*, *gui_settings(void)*, *gui_programming_mode(void)*, *gui_keyboard(void)*, které zajišťují zobrazování určitých obrazovek - hlavní nabídka, nastavení, programovací režim a klávesnice. Procedura *gui_settings_speed(void)* a *void gui_settings_plc(void)* slouží k zobrazení rychlostních kroků a k zapínání PLC režimu na obrazovce nastavení.

5.6.3 void gui_update_speed(void)

Tato procedura je volána v případě otočení inkrementálního spínače a následné změny rychlosti. Procedura zobrazí změněnou rychlost na hlavní obrazovce.

5.6.4 void gui_update_address(void)

Jedná se o velmi podobnou proceduru jako je předchozí. Je zde rozdíl v tom, že se zde zobrazuje změněná adresa, která se změní pomocí tlačítek k tomu určených.

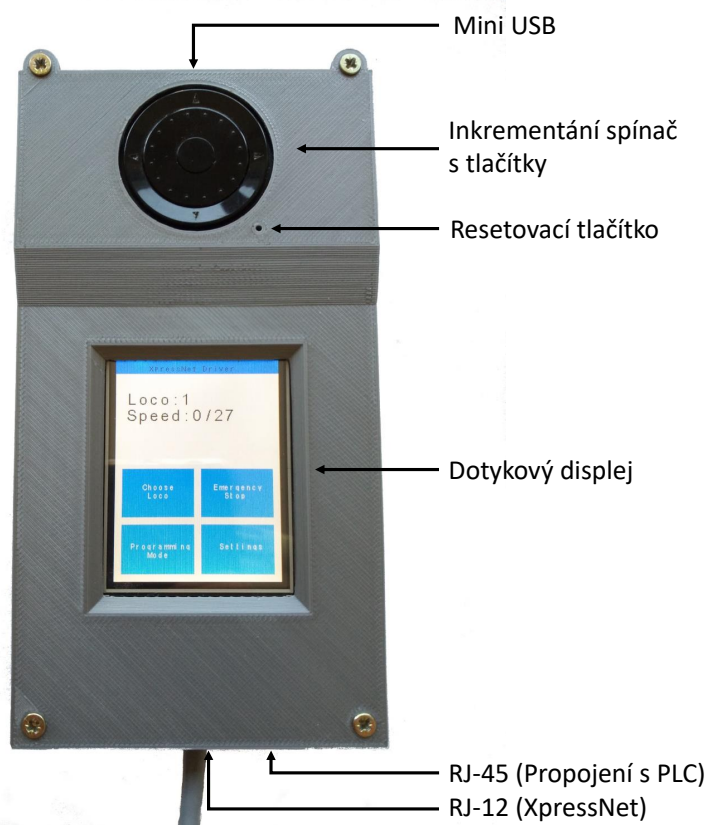
5.6.5 void gui_message(char text[])

Tato procedura je často volána při analyzování přijatých dat z příkazové stanice. V horní části obrazovky zobrazí text, který je zadaný v parametru *char text[]*.

Kapitola 6

Obsluha ovladače

Na začátku této kapitoly bude ovladač podrobně popsán z hardwarové stránky. Dále zde bude podrobně popsáno grafické uživatelské prostředí a veškerá nastavení ovladače. Na konci této kapitoly by měl být uživatel schopen ovladač bez problémů obsluhovat.



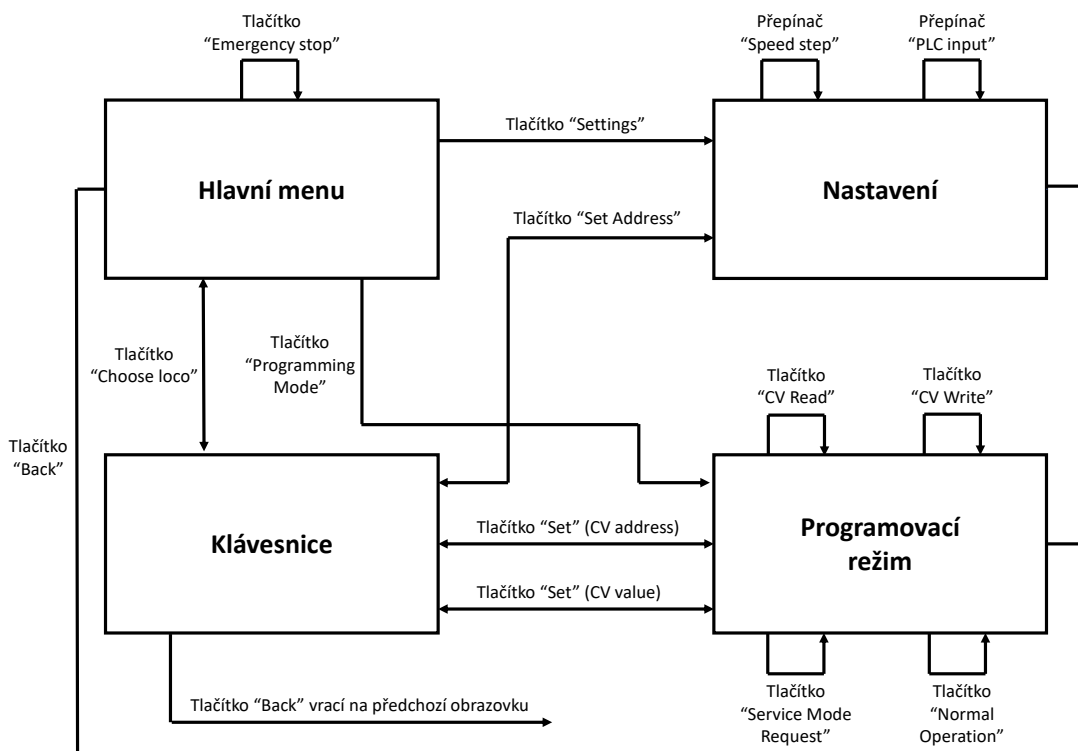
Obrázek 6.1: Popis ovladače

6.1 Popis ovladače

Na obrázku 6.1 je zobrazen ovladač s těmi nejdůležitějšími popisky. Na horním okraji ovladače se nachází mini-USB, pomocí kterého je možné ovladač naprogramovat. Na vrchní straně ovladače se vyskytují inkrementální spínač s tlačítky a dotykový displej, pomocí kterých ovladač ovládáme. V blízkosti inkrementálního spínače se nachází otvor, kde se s pomocí špendlíku může ovladač restartovat. Na dolním okraji ovladače jsou dva konektory. Menší z nich je konektor RJ-12, který slouží k propojení ovladače s příkazovou stanicí. Tímto konektorem neprobíhá pouze komunikace, ale ovladač je zde také napájen. Větší konektor je RJ-45, pomocí kterého lze ovladač propojit s PLC.

6.2 Vývojový diagram grafického rozhraní

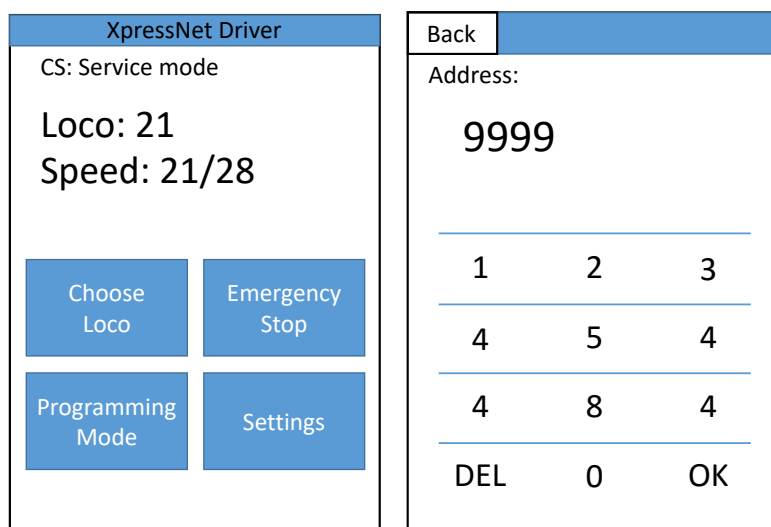
Grafické rozhraní se skládá se 4 základních obrazovek - hlavní menu, nastavení, programování režim a klávesnice. Propojení mezi jednotlivými obrazovkami je zobrazeno na obrázku 6.2.



Obrázek 6.2: Ovladač - Programovací režim

6.3 Hlavní menu

Po zapojení ovladače k XpressNet příkazové stanici se na obrazovce objeví hlavní nabídka. Ta obsahuje v horní části zprávy z příkazové stanice, adresu ovládané lokomotivy a rychlost ovládané lokomotivy. V dolní části hlavní nabídky se nachází 4 tlačítka. Tlačítko „Change address“ odkáže na klávesnici, kde je možné nastavit adresu ovládané lokomotivy. Po stisknutí tlačítka „Emergency Stop“ se zastaví všechny lokomotivy na kolejišti. Pod tlačítkem „Programming Mode“ se nachází programovací režim, který bude podrobněji popsán. Poslední tlačítko „Settings“ odkáže na nastavení ovladače. Hlavní nabídka a klávesnice jsou zobrazeny na obrázku 6.3.



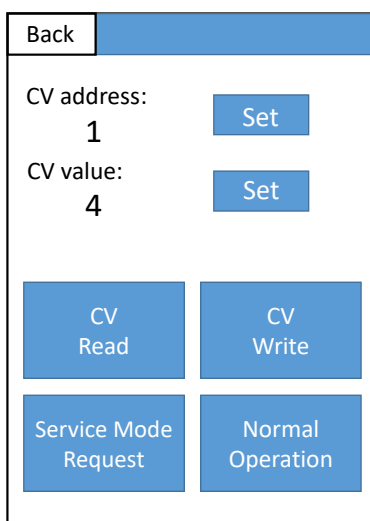
Obrázek 6.3: Ovladač - Hlavní nabídka a numerická klávesnice [1]

6.4 Programovací režim

Velmi důležitou součástí ovladače je programovací režim, pomocí kterého se nastavuje lokodekodér. Do tohoto režimu se přistoupí pomocí kliknutí na tlačítko „Programming Mode“. Zde je možné číst a zapisovat CV hodnoty dekodéru. Dále je nutné, aby lokomotiva s nastaveným dekodérem stála na oddělené programovací koleji.

Na obrázku 6.4 se nachází obrazovka programovacího režimu. V horní části obrazovky vlevo se nachází adresa a hodnota CV registru. Vpravo se nachází tlačítka nastavení adresy a hodnoty. V dolní části obrazovky se nachází 4 tlačítka: „Read CV“, „Write CV“, „Service mode Request“ a „Resume normal operation“.

V případě, že je potřeba přečíst CV registr, tak se nejdříve pomocí tlačítka „Set“ nastaví adresa CV registru. Poté se stiskne tlačítko „Read CV“, příkazová stanice se přepne do servisního módu, pokud v něm již nebyla, a poté lokomotiva lehce zavrčí. Servisní mód se pozná tak, že na příkazové stanici se rozsvítí zelená dioda. V případě, že se to nestane, tak se postup zopakuje. Nakonec se stiskne tlačítko „Service mode Request“, a tím se získá hodnota



Obrázek 6.4: Ovladač - Programovací režim

na zadané adrese CV registru. Na obrazovce se obnoví položka „CV value“ s novou hodnotou. Pokud byla špatně zadána adresa, nebo není možné zadanou adresu číst, zobrazí se hláška "Data Byte not found".

Pro zápis hodnoty do CV registru je nutné nastavit pomocí tlačítek „Set“ adresu CV registru a i jeho hodnotu. Poté se stiskne tlačítko „Write CV“ a lokomotiva lehce zavrčí. Zde příkazová stanice neposílá žádné potvrzení, jestli byl registr správně přepsán, a proto je vhodné registr po zápisu přečíst podle předchozího postupu.

Pokud se potřebujeme vrátit do normálního režimu, je nutné stisknout tlačítko „Resume normal operation“. Na příkazové stanici zhasne zelená dioda a je možné znovu ovládat lokomotivu.

6.4.1 Základní nastavení po instalaci dekodéru

Pokud se do lokomotivy nově nainstaluje dekodér, tak se lokomotiva postaví na programovací kolej a vyčíslí se hodnoty, které jsou nastaveny. Poté se provede reset tak, že se na CV8 zapíše hodnota 2. Pokud lokomotiva začne lehce vrčít, tak je vše v pořádku. Pokud se tak nestane, tak je nutné zkontrolovat zapojení dekodéru a hledat závadu. Nyní se lokomotiva může otestovat. Výchozí adresa dekodéru je 3. Na ovladači se nastaví adresa lokomotivy na hodnotu 3 a lokomotiva se rozpohybuje pomocí rychlostního kolečka. Pokud lokomotiva reaguje, tak vše je v pořádku a nachází se v základním nastavení. [8], [7]

6.4.2 Nastavení jízdních vlastností modelu

Aby se naplno využily výhody digitálního řízení, je vhodné podrobné nastavení motoru lokomotivy pomocí změn CV, mezi které patří tyto: [8], [7]

- CV2 - Zde se nachází minimální rozjezdové napětí. Výchozí hodnota je 0, rozsah 0-255, kde každých 20 odpovídá 1 V.
- CV3 - Zde se nachází rozjezdové zrychlení. Výchozí hodnota je 1, rozsah 1-255. Čím vyšší hodnota je nastavená, tím je rozjezd pomalejší.
- CV4 - Zde se nachází hodnota zpomalení/brzdění. Výchozí hodnota je 1, rozsah 1-255. Čím vyšší hodnota je nastavena, tím je větší dojezd lokomotivy.
- CV5 - Zde se nachází hodnota nejvyšší rychlosti modelu. Výchozí hodnota je 0 - při této hodnotě není aktivní, rozsah 1-255, kde každých 20 odpovídá 1 V. Čím vyšší hodnota, tím je vyšší maximální rychlost.
- CV6 - Zde se nachází hodnota střední rychlosti. Výchozí hodnota je 0 - při této hodnotě není aktivní, rozsah 1-255. Většinou se nastavuje 1/3 maximální rychlosti nastavené v CV5. Zrychlení bude postupné a lokomotiva se bude chovat realističtěji. Doporučená hodnota je 40.

6.4.3 Doporučené nastavení pro lokomotivy

Jízdní vlastnosti lokomotiv ovlivňuje kvalita převodu, typ motoru a také nastavení dekodéru. Pomocí dekodéru je mnoho možností, jak jízdní vlastnosti vylepšit. Dekodér navíc obsahuje funkci BEMF, která plní funkci zpětné vazby. Dekodér tímto způsobem udržuje stálou rychlost, i když se mění velikost zatížení. Zatížení ovlivňuje stoupání, klesání na trati, nebo počet zapojených vozů.

Na ukázkou se zde nachází nastavení pro elektrickou lokomotivu. Z dokumentace, která byla přiložena k dekodérům, je doporučené nastavení pro tento typ lokomotiv takové:

$$CV3 = 12, CV4 = 12, CV5 = 160, CV6 = 70, CV61 = 1$$

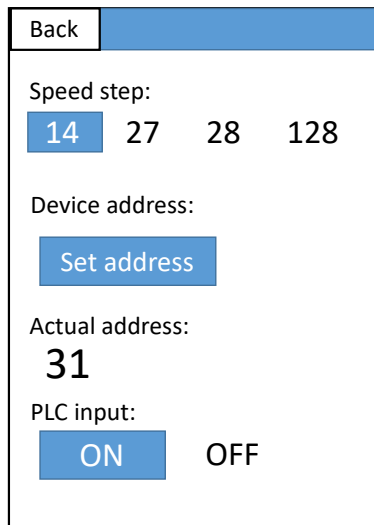
Při tomto nastavení se získá svižné zrychlení a brzdění, které odpovídá elektrickým lokomotivám. [8], [7]

6.5 Nastavení

Velká část nastavení zůstala oproti mé bakalářské práci velmi podobná. Stále se zde nachází zobrazení aktuální adresy ovladače, nastavení adresy ovladače a změna rychlostního kroku. Adresa ovladače může být v rozsahu od 1 do 31. Navíc zde přibyla možnost zapnout, nebo vypnout ovládání pomocí PLC. Grafické rozhraní nastavení je zobrazeno na obrázku 6.5.

6.5.1 Provoz pomocí PLC

V případě zapnutí PLC režimu se ovladač přepne do módu, kdy je ovládán pouze pomocí PLC. V tomto režimu není funkční inkrementální spínač a tlačítka, které provádějí jakékoli změny v rychlosti. Ovládání funguje podle tabulky 6.1, která vychází z diplomové práce Ing. Pernera. Optočlen invertuje logické hodnoty a tabulka již obsahuje invertované hodnoty, tedy hodnoty, které se dostanou přímo na STM32F429. V případě jízdy lokomotivy se rychlost na ovladači nastaví na 14 z maximální rychlosti 27. Obě lokomotivy mají na ovladači pevně



Obrázek 6.5: Ovladač - Nastavení

dané adresy - 1 a 2. Jedná se pouze o přípravu na ovládání pomocí PLC a tyto hodnoty bude potřeba po reálném testování ještě doladit. [13]

Tabulka 6.1: Tabulka na ovládání pomocí PLC [11]

Pin	Význam	Poznámka
PB12	Lokomotiva 1 - ON/OFF	log.0 - Jízda, log.1 - Stop
PB13	Lokomotiva 1 - Směr	log.0 - Dozadu, log.1 - Dopředu
PG3	Lokomotiva 2 - ON/OFF	log.0 - Jízda, log.1 - Stop
PD13	Lokomotiva 2 - Směr	log.0 - Dozadu, log.1 - Dopředu

6.6 Inkrementální spínač s tlačítky

V horní části ovladače se nachází inkrementální spínač se 4 postranními tlačítky a 1 tlačítkem uprostřed. Inkrementální spínač slouží k nastavování rychlosti lokomotivy na dané adrese. Pravým postranním tlačítkem se inkrementuje adresa lokomotivy a levým postranním tlačítkem se degraduje hodnota adresy lokomotivy. Prostřední tlačítko slouží k zastavení aktuálně ovládané lokomotivy. Horní tlačítko změní směr lokomotivy a poslední dolní tlačítko pozastaví lokomotivu - při opakovaném stisknutí se lokomotiva rozjede na předchozí rychlost.

Kapitola 7

Ověření funkčnosti ovladače

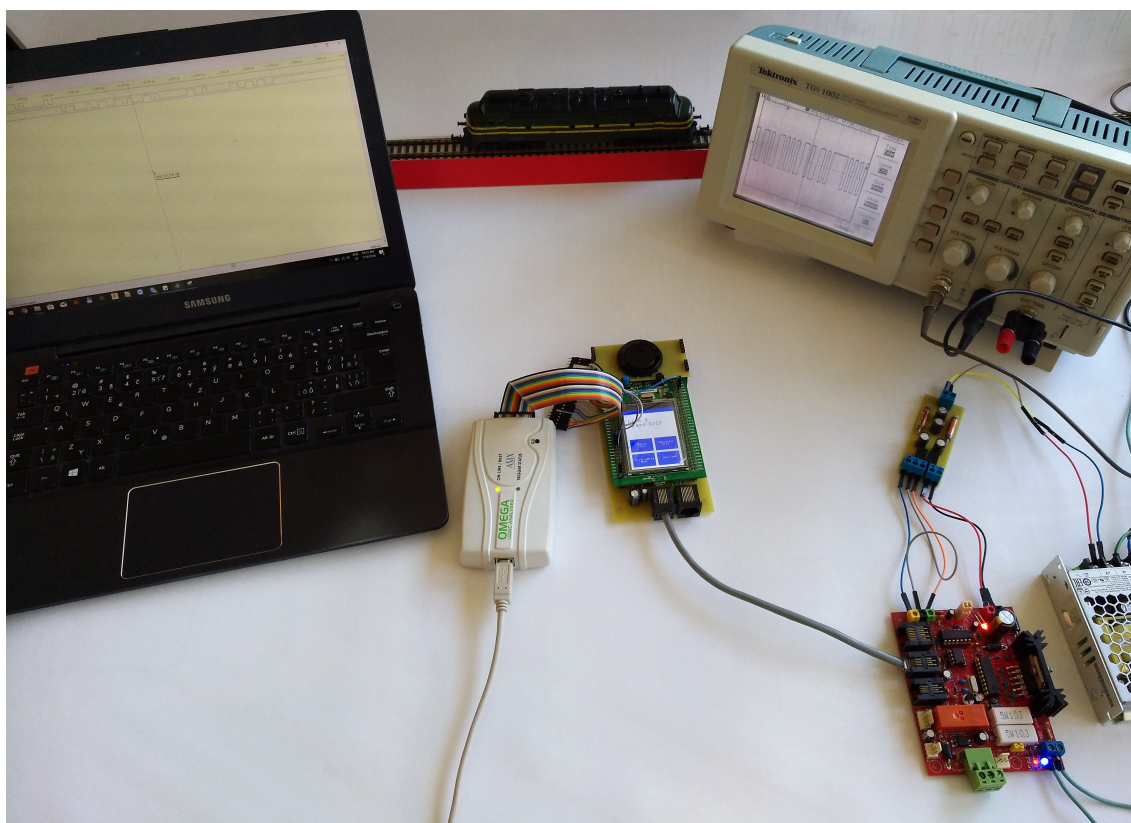
V této kapitole bude ověřena funkčnost ovladače. Ověřování probíhalo několika způsoby. Jeden ze způsobů bylo ověřovat, jestli se lokomotiva chová podle toho, co jí zadává ovladač. Pro podrobnější testování byl využit logický analyzátor, pomocí kterého se snímala komunikace mezi ovladačem a příkazovou stanicí. Pro ověření správných napěťových hladin byl využit osciloskop.

7.1 Schéma testovacího pracoviště

Ověřování probíhalo v místnosti T2:B3-443 katedry elektrotechnologie. Pracoviště je zobrazeno na obrázku [7.1](#). Seznam použitého zařízení a přístrojů je zde:

- Vyrobený ovladač XpressNet
- Příkazová stanice NanoX S88
- Vyrobený zdroj k příkazové stanici
- Testovací kolej
- Lokomotiva s dekodérem
- Logický analyzátor - ASIX Omega
- Osciloskop - Tektronix TDS 1002
- Izolovaný zdroj
- Notebook - Samsung 730U - Windows 10

Poznámka: Izolovaný zdroj byl využit z důvodu snímání diferenčního napětí. Využitý osciloskop je jinak uzemněn.



Obrázek 7.1: Testovací pracoviště

7.2 Ověřování pomocí lokomotivy

Díky jednoduchosti byl tento způsob ověření často využíván při vývoji a ladění ovladače. Při tomto ověřování bylo zapotřebí ovladače, lokomotivy, příkazové stanice a testovací koleje.

7.2.1 Ovládání lokomotivy

Na ovladači byly ověřeny všechny základní funkce, pomocí kterých se ovládá lokomotiva. Jednalo se o změnu rychlosti a směru, zastavení ovládané lokomotivy a zastavení všech lokomotiv na trati. Všechny tyto funkce fungovaly spolehlivě. Při tomto ověřování bylo také velkou výhodou to, že se testoval z pohledu koncového uživatele, a proto se ovladač tímto způsobem doladil k tomu, aby byl více uživatelsky přívětivý.

7.2.2 Programování dekodéru v lokomotivě

Podle postupu, který se nachází v předchozí kapitole v sekci 6.4 bylo ověřováno programování dekodéru. Lokomotiva stála na programovací koleji a ověřilo se čtení CV registru a zápis hodnoty do CV registru. Možnosti ověření programování dekodéru byly dvě. Jedna z

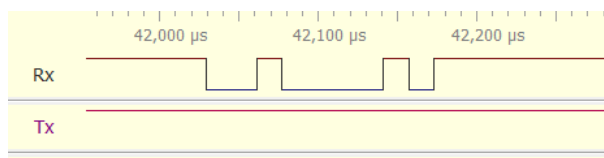
nich je nechat si vypsat hodnotu CV registru na displej a porovnat to s výchozími hodnotami v tabulce, která se nachází v manuálu k dekodéru. Druhá možnost je, že lokomotiva v době programování dekodéru lehce vrčí. Obě možnosti byly ověřeny a programovací režim na ovladači fungoval správně.

7.3 Ověřování pomocí logického analyzátoru

Ověřování probíhalo pomocí logického analyzátoru ASIX Omega, aplikace Sigma & Omega Logic Analyzers na notebooku s Windows 10. Kvůli lepší přehlednosti byla data snímána na programovatelném kitu STM32f429, kde se jedná o protokol UART. Proto na většině průběhů jsou snímány piny Rx a Tx. Rx je přijímací pin a Tx je vysílací pin. V případě RS-485 se jedná o diferenciální sběrnici, a proto se zde nelze rozlišit jaká data jsou přijímána, nebo vysílána. Na ověření sběrnice RS-485 byl využit osciloskop.

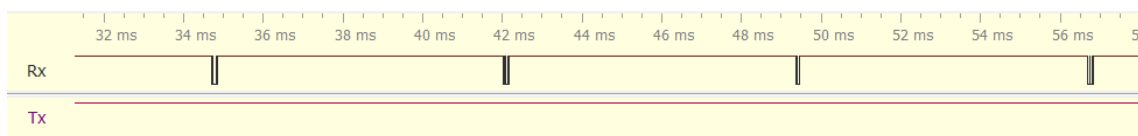
7.3.1 Příkazové okno

Jak bylo řečeno, tak příkazová stanice posílá pravidelně příkazová okna, kterými umožňuje zadat ovladačům jejich požadavek. V každém příkazovém okně se mění pouze adresa ovladače. Samostatné příkazové okno je zobrazeno na obrázku 7.2. Příkazové okno bylo již popsáno na konci kapitoly 3, ale i přesto bude toto příkazové okno rozebráno. V případě, že se zanedbá START bit a STOP bit, tak zbyde 0100 0010. Paritní bit je tedy 0, dále následuje 10, to identifikuje příkazové okno, a nakonec je adresa 00010. Jedná se tedy o příkazové okno pro ovladač, který má adresu 2.



Obrázek 7.2: Samostatné příkazové okno

Na obrázku 7.3 je znázorněna řada příkazových oken. Okna jsou od sebe vzdálena asi 7 ms a celkový počet příkazových oken je 32. Toto číslo také určuje maximální počet připojených ovladačů.

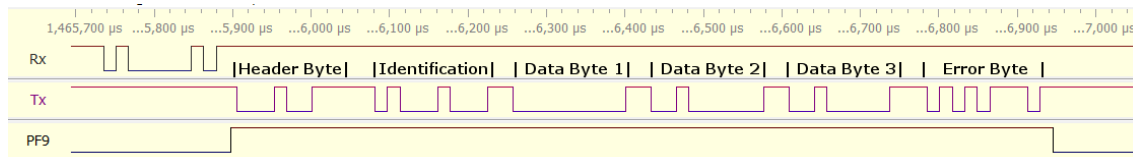


Obrázek 7.3: Řada příkazových oken

7.3.2 Změna rychlosti a směru lokomotivy

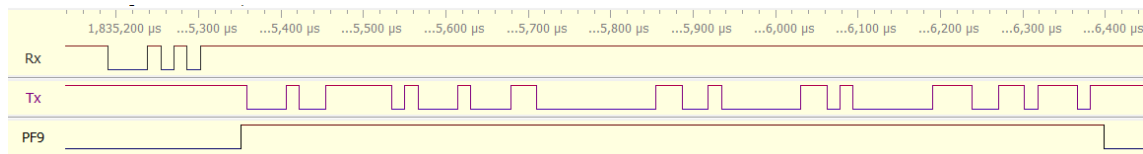
Zde se jedná o ten nejzákladnější paket, v případě, že se má lokomotiva uvést do pohybu. Základní popis tohoto paketu se nachází na konci kapitoly 3, podrobný popis se nachází v dokumentaci XpressNet. Na obrázku 7.4 je tento přenos sejmут.

Protože se jedná o složitější paket, tak bude rozebrán. Velmi důležité je dát pozor na to, že se data po sběrnici pohybují v opačném pořadí. V předchozím případě to nebylo jasně vidět, a proto na to upozorňuji až teď. Komunikace začne po přijetí „Call Byte“ na pinu Rx a ovladač odpovídá paket na pin Tx. „Header Byte“ je 1110 0100 a to značí, že se jedná o paket o změně rychlosti a směru lokomotivy. „Identification“ je 0001 0001, a proto se jedná o 27 rychlostních kroků. „Data Byte 1“ - 0000 0000 a „Data Byte 2“ - 0000 0010 značí adresu lokomotivy, a proto ovládaná lokomotiva má adresu 2. „Data Byte 3“ je 1000 0010 a to první bit znamená směr dopředu a zbytek značí rychlost 2. „Error Byte“ je 0111 0101 a to je XOR všech předchozích bytů. Dále je zde zobrazen pin PF9, který určuje směr toku dat pro převodník TTL na RS-485. V případě, že ovladač vysílá je tento pin v logické 1.



Obrázek 7.4: Změna rychlosti a směru lokomotivy s pinem PF9

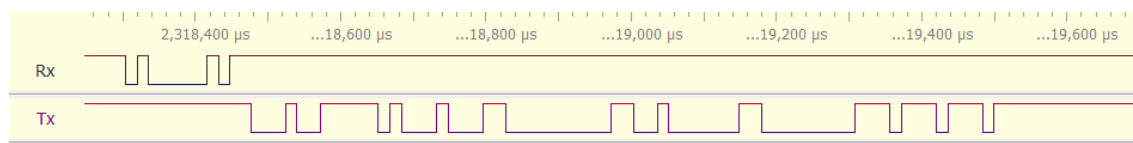
Ovladač musí odpovídat pouze na „Call Byte“, který mu je určený. Na obrázku 7.4 se jedná o ovladač adresy 1. Aby se ověřilo, zda-li funguje správně adresa ovladače, byl sejmут průběh, který je na obrázku 7.5. Zde má ovladač nastavenou adresu 20, a proto „Call Byte“ má v tomto případě tvar 1101 0100. V tomto bytu má adresa hodnotu 10100 a to je v desítkové soustavě 20. Zbytek paketu lze rozložit jako v předchozím příkladu.



Obrázek 7.5: Změna rychlosti a směru lokomotivy s jinou adresou ovladače

7.3.3 Zastavení ovládané lokomotivy

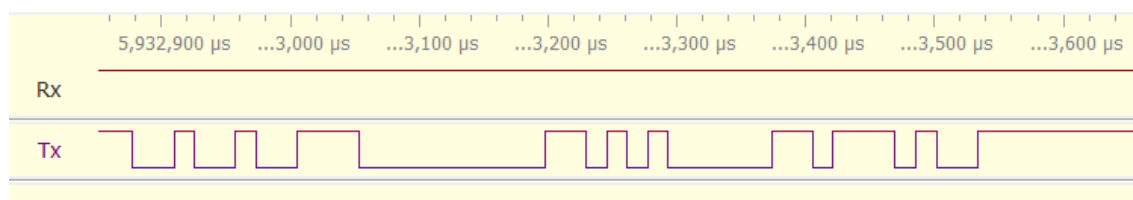
V tomto případě se jedná o stejný paket jako při změně rychlosti a směru lokomotivy. Je zde pevně nastavena rychlost na 0, tedy „Data Byte 3“ se rovná 0.



Obrázek 7.6: Zastavení ovládané lokomotivy

7.3.4 Pozastavení ovládané lokomotivy

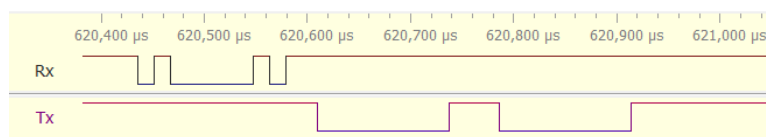
Při stisknutí dolního tlačítka u inkrementálního spínače se odešle posloupnost, která je zobrazena na obrázku 7.7. „Header Byte“ má hodnotu 1001 0001, „Data Byte 1“ - 0000 0000, „Data Byte 2“ - 0000 0010 a na konci se nachází „Error Byte“. Datové byty značí adresu lokomotivy a jedná se o lokomotivu adresy 2. Tato posloupnost pozastaví ovládanou lokomotivu.



Obrázek 7.7: Pozastavení ovládané lokomotivy

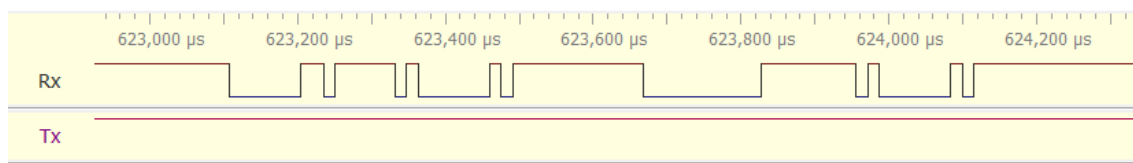
7.3.5 Zastavení všech lokomotiv

Při stisknutí tlačítka na displeji „Emergency stop“ ovladač vyšle paket, který je zobrazen na obrázku 7.8. „Header Byte“ má hodnotu 1000 0000 a „Error Byte“ nabývá hodnoty 1000 0000. Tyto dva byty odpovídají podle dokumentace XpressNet paketu „Stop all locomotives request (emergency stop)“, a proto žádost o zastavení všech lokomotiv je ve správném tvaru.



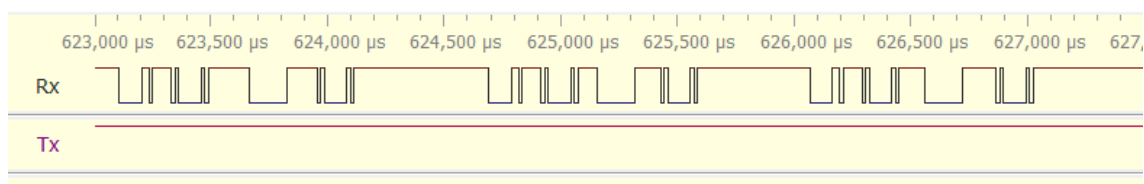
Obrázek 7.8: Zastavení všech lokomotiv

Pokud příkazová stanice tento paket v pořádku přijme, zastaví tak všechny lokomotivy na trati a pošle o tom zprávu všem připojeným ovladačům. Tato zpráva má tvar: „Call Byte“ má tvar 0110 0000, „Header Byte“ 1000 0001, „Data Byte 1“ 0000 0000 a poslední „Error Byte“ 0110 0001. Když se tato zpráva porovná s obrázkem 7.9, může se konstatovat, že ovladač tuto zprávu přijmul správně.



Obrázek 7.9: Oznámení o zastavení všech lokomotiv

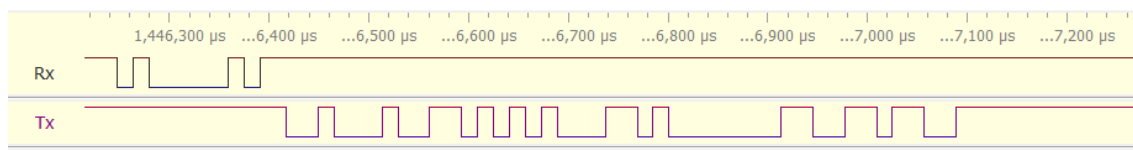
Příkazová stanice ale tuto zprávu posílá 3x za sebou, jak je zobrazeno na obrázku 7.10. Stanice se tímto ujišťuje, že všechny ovladače tuto zprávu obdrží v pořádku. Tento jev ještě nastává v případě zprávy o vypnutí napájení tratě.



Obrázek 7.10: Řada oznámení o zastavení všech lokomotiv

7.3.6 Čtení CV registru

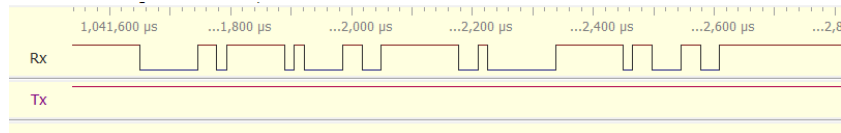
V případě čtení hodnoty z CV registru ovladač vysílá paket, který je zobrazen na obrázku 7.11. „Header Byte“ má hodnotu 0010 0010, „Data Byte 1“ má hodnotu 0001 0101. Tyto byty jsou dány dokumentací XpressNet. Dále pokračuje „Data Byte 2“, kde se nachází adresa CV registru, který má být přečten. V tomto případě 0000 0001 - adresa CV registru je 1. Poslední je „Error Byte“ 0011 0110 a to odpovídá XORu všech předchozích bytů. Celá tato posloupnost odpovídá „Direct Mode CV read request“ z dokumentace XpressNet.



Obrázek 7.11: Čtení hodnoty CV registru

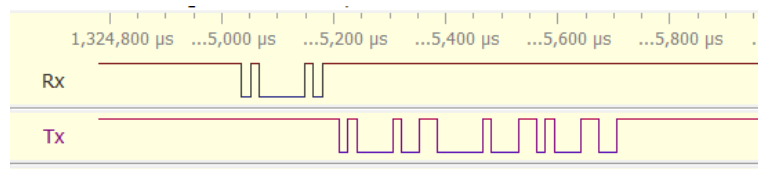
Po odeslání paketu jsou dvě možnosti, které mohou nastat. Pokud se ovladač nachází v normálním režimu, tak příkazová stanice odešle paket, který oznamuje, že se příkazová stanice přepnula do servisního módu. Poté si stanice připraví data, která byla přečtena z dekodéru. Druhá možnost je taková, že ovladač se již v servisním módu nachází, a proto žádný paket neodesílá a pouze si připraví data. Na obrázku 7.12 se nachází sejmutý průběh po odeslání žádosti o čtení z registru. „Call Byte“ je 0110 0000, „Header Byte“ 0110 0001, „Data Byte 1“ 0000 0010 a „Error Byte“ 0110 0011. Tato posloupnost podle dokumentace

XpressNet odpovídá paketu „Service Mode entry“, tím se dá snadno určit, že ovladač se nejdříve nacházel v normálním módu.



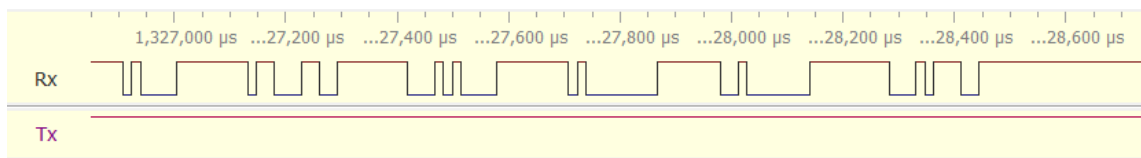
Obrázek 7.12: Zpráva o přechodu do servisního módu

Jak bylo řečeno, příkazová stanice si data pouze připraví, ale ovladači je sama od sebe neodešle. Aby bylo možné získat data, je potřeba odeslat paket „Request for Service Mode results“. Tomuto paketu odpovídá posloupnost, která se odešle po stisknutí tlačítka „Service Mode request“ v programovacím režimu. Posloupnost je zobrazena na obrázku 7.13. „Header Byte“ je 0010 0001, „Data Byte 1“ 0001 0000 a „Error Byte“ 0011 0001. Po přijetí předcho-



Obrázek 7.13: Zaslání požadavku o odpověď příkazové stanice

zího paketu příkazová stanice odešle data ve formátu, který odpovídá paketu „Service Mode response for Direct CV mode“ z dokumentace XpressNet. Na obrázku 7.14 je tento průběh zobrazen. Nyní proběhne jeho rozbor. Začátek komunikace začíná „Call Bytem“ 1110 0001, který oznamuje, že odpověď je pro ovladač adresy 00001, v desítkové soustavě 1. Následuje „Header Byte“ 0110 0011 a „Data Byte 1“ 0001 0100, které jsou pevně zadány v dokumentaci XpressNet. „Data Byte 2“ hodnoty 0000 00001 říká, že hodnota byla čtena z CV registru adresy 1. „Data Byte 3“ se nachází ve tvaru 0000 0010 a odpovídá to hodnotě uložené v CV registru na adrese 1. Hodnota registru je v desítkové soustavě 2. Paket ukončuje jako vždy „Error Byte“.

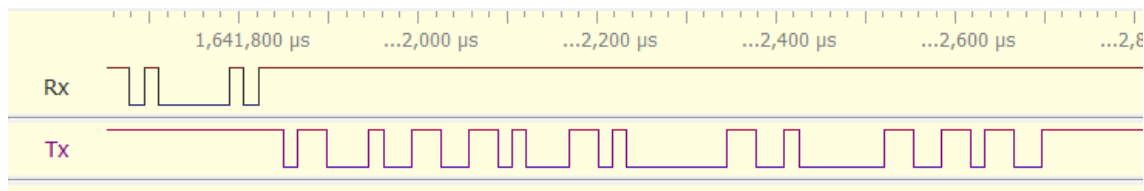


Obrázek 7.14: Odpověď příkazové stanice

7.3.7 Zápis do CV registru

Na obrázku 7.15 se nachází posloupnost, která je odeslána po stisknutí tlačítka „Write CV“ v programovacím módu. „Header Byte“ je 0010 0011 a „Data Byte 1“ 0001 0110. „Data Byte 2“ se nachází ve tvaru 0000 0001 a určuje, že se bude zapisovat do CV registru na adrese 1. „Data Byte 3“ je 0000 0010 a je v něm obsažena hodnota, která má být uložena do CV registru na adrese 1. Hodnota v desítkové soustavě je 2. Na konci se nachází „Error Byte“. Tento paket tedy požádá příkazovou stanici o to, aby do dekodéru zapsala hodnotu 2 na CV registru adresy 1. Celá tato posloupnost odpovídá „Direct Mode CV write request“, a proto je komunikace správná.

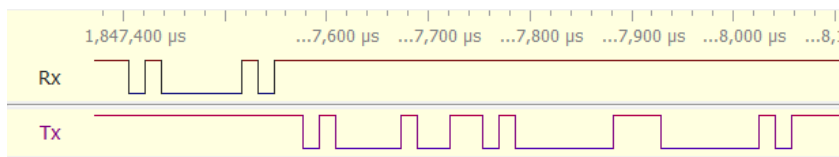
Příkazová stanice danou žádost provede a již neodesílá žádnou odpověď. V případě, že se ovladač před odesláním paketu nachází v normálním režimu, tak se příkazová stanice přepne do servisního módu a informuje o tom všechny ovladače.



Obrázek 7.15: Zápis hodnoty do CV registru

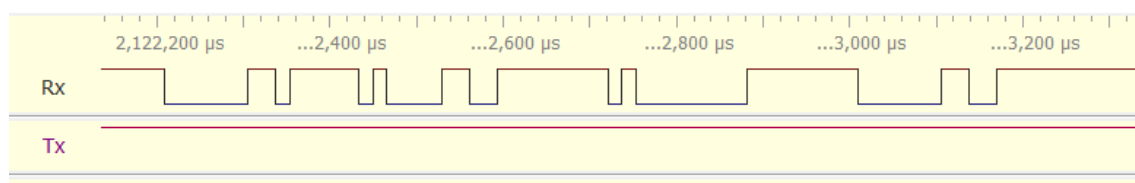
7.3.8 Obnovení normálního provozu

Pro ukončení programovacího módu je nutné zaslat příkazové stanici žádost o obnovení normálního provozu. Tento paket je na obrázku 7.16. „Header Byte“ nabývá hodnoty 0010 0001, „Data Byte 1“ hodnoty 1000 0001 a poslední „Error Byte“ má hodnotu 1010 0000. S porovnáním dokumentace XpressNet tato posloupnost odpovídá paketu „Resume operations request“, a proto je tento přenos v pořádku.



Obrázek 7.16: Žádost o obnovení normálního provozu

Pokud předcházející posloupnost správně přijme příkazová stanice, tak se přepne do normálního režimu a pošle o tom informaci všem připojeným ovladačům, jak je možno vidět na obrázku 7.17. „Call Byte“ je 0110 0000, „Header Byte“ 0110 0001, „Data Byte“ 0000 0001 a „Error byte“ má hodnotu 0110 0000. Podle dokumentace XpressNet se jedná o „Normal operation resumed“, a proto komunikace proběhla správně.



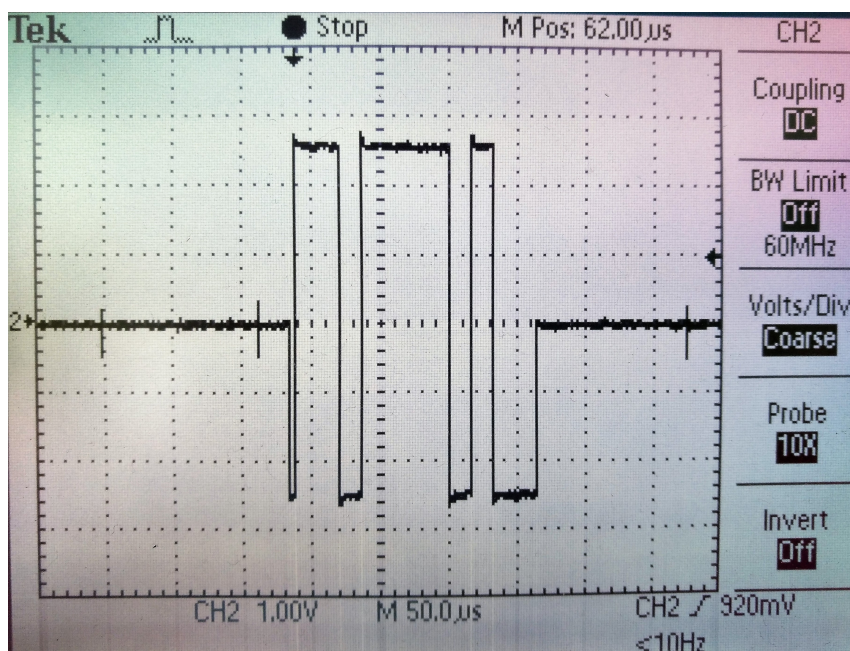
Obrázek 7.17: Oznámení o obnovení normálního provozu

7.4 Ověřování pomocí osciloskopu

Poslední metoda ověřování byla pomocí osciloskopu Tektronix TDS 1002. Zde se jednalo pouze o ověření napěťových úrovní na sběrnici RS-485 a DCC signálu. Ohledně správnosti komunikace mezi ovladačem a lokomotivou stačily dva předchozí kroky. V této části jsem chtěl hlavně ukázat, jak vypadají signály mezi jednotlivými komponenty ve skutečných napěťových hladinách a ne pouze v logických hodnotách.

7.4.1 RS-485

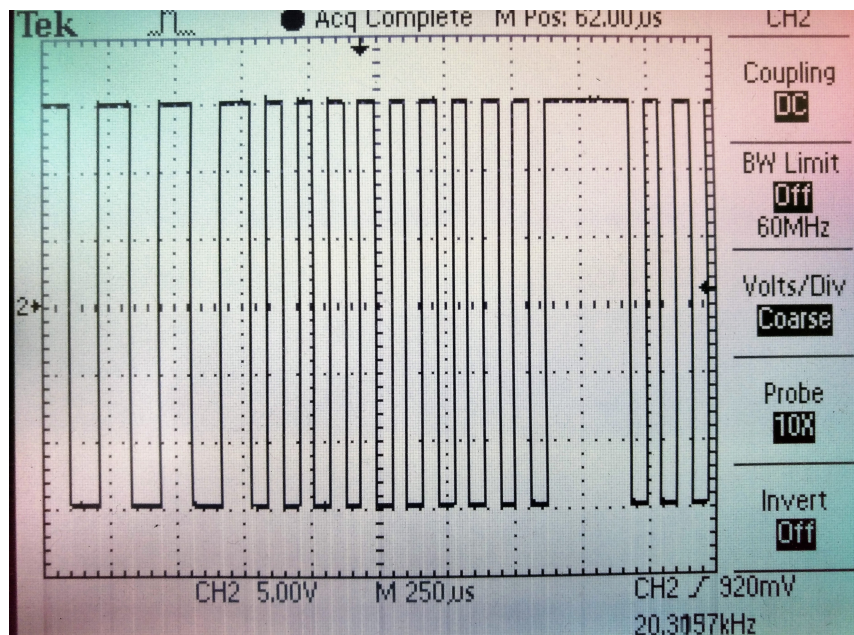
Jedná se o komunikaci mezi ovladačem a příkazovou stanicí, přesněji řečeno mezi převodníkem ovladače a příkazovou stanicí. Podle standardu RS-485 se rozlišuje logická 1 tak, že rozdíl napětí mezi vodiči je menší jak -200 mV, a v případě logické 0 je tento rozdíl větší jak 200 mV. Na obrázku 7.18 je vidět, že se zde dosahuje hodnot přibližně $-2,5$ V a $2,5$ V, a proto je jisté, že logické hodnoty jsou od sebe jasně odlišeny. [15]



Obrázek 7.18: Sejmутý průběh RS-485 signálu

7.4.2 DCC

Mezi příkazovou stanicí a lokomotivou se jedná o signál DCC. Logické hodnoty se zde rozlišují pulzně šířkovou modulací, jak je zobrazeno na obrázku 7.19. Napětí se pohybuje na hodnotách -15 V a 15 V a je určeno zdrojem MEAN WELL LRC-50-15, na kterém lze napětí ještě zvýšit až na 18 V. V případě kolejiště H0 je ale optimální hodnota napětí 15 V. [14]



Obrázek 7.19: Sejmутý průběh DCC signálu

Kapitola 8

Závěr

Tato práce se zabývala zhotovením ovladače XpressNet pro modelové kolejiště v laboratoři T2:E1-3 katedry elektrotechnologie. Součástí této diplomové práce byla také modernizace tohoto kolejiště.

8.1 Modernizace kolejiště

O modernizaci kolejiště pojednává kapitola 2. Na začátku této kapitoly byl popsán DCC systém a jeho porovnání s analogovým systémem. Dále jsou zde popsány všechny komponenty, které byly na kolejiště použity. Příkazová stanice byla zakoupena a byl k ní navržen napájecí zdroj, který jí dodává napětí 5 V, 12 V a 15 V. Lokodekodéry byly také zakoupeny a nainstalovány do lokomotiv. Celý systém byl ověřen s reálným ovladačem Lenz LH100 a fungoval bez problémů. Kolejiště a lokomotivy nyní odpovídají novým standardům, a proto se nyní jedná o velmi dobrý základ pro další vývoj na tomto kolejišti. Nabízí se zde přidání dalších funkcí do lokomotiv, ovládání výhybek, zpětná vazba z kolejiště atd. V těchto ohledech systém XpressNet a DCC nabízí mnohé.

8.2 Ovladač

Nejdůležitější částí této práce byl ale samotný ovladač, který se podařilo dostat do velmi slušného stavu. Když se vrátím do mé bakalářské práce, výsledek zde byl pouze prototyp ovladače, pomocí kterého jsem podrobně pochopil fungování sběrnice XpressNet, ale pro klasického uživatele byl naprosto nevhodný. Prototypu chyběl převodník TTL na RS-485, správný konektor na XpressNet a ani nebylo možné ho vložit do ergonomického pouzdra. I přes všechny tyto nevýhody byl tento prototyp výborný základ pro finální ovladač, hlavně z hlediska programu.

Program byl kompletně refaktorován, ale funkcionalita komunikace byla na velmi slušné úrovni již v prototypu a ve finálním ovladači stačily drobné úpravy. Bylo také nutné zde přidat nové funkce pro inkrementální spínač s tlačítky a pro přepínání toku na převodníku. Novinkou je zde také příprava na ovládání pomocí PLC, které již bylo nainstalované na kolejišti. Jedná se o režim, kdy ovladač získává informace přes konektor RJ-45, převede

je na standard XpressNet a vysílá do příkazové stanice. V programu byl psán ve formě, která odpovídá dokumentaci DoxyGen, a proto je možné tuto dokumentaci z programu vyexportovat. Tuto podrobnou dokumentaci velmi ocení všichni, kdo si budou chtít ovladač přizpůsobit podle svého.

Z hardwarového hlediska zde bylo nutné přidat převodník TTL na RS-485, regulátor napětí a optočlen, který slouží jako galvanické oddělení při propojení s PLC. Pro pohodlnější ovládání se zde nachází nový inkrementální spínač s tlačítky. Celou tuto hardwarovou část bylo nutné propojit a dát do takového tvaru, aby tyto vnitřní části ovladače dobře zapadly do pouzdra. Proto byly navrženy dva nové plošné spoje - hlavní deska a horní deska. Všechny komponenty až na inkrementální spínač se nachází na hlavní desce. Inkrementální spínač se nachází na horní desce. Samozřejmě zde byl také konektor RJ-12, který schvaluje standard XpressNet. Navíc se zde nachází konektor RJ-45, který slouží k propojení s PLC. Hardware byl vybrán, navržen a otestován, a proto byl tento úkol splněn.

Aby ovladač byl ergonomický, tak bylo namodelováno a vytisknuto také pouzdro. V této části práce se jednalo hlavně o možnost naučit se modelovat jednoduché objekty v Autodesk Fusion 360 a nahlédnout do světa 3D tiskáren.

Výsledkem toho všeho je funkční ovladač, který odpovídá standardům XpressNet a je vhodný pro každého uživatele. Ovladač byl podrobně otestován v kapitole č. 7. Testování bylo úspěšné a tím považují ovladač za funkční. Navíc se zde k systému XpressNet nachází příprava pro ovládání pomocí PLC, kterou žádný komerčně prodávaný ovladač nenabízí.

8.3 Budoucí vývoj ovladače

Jak bylo již řečeno, nad rámec této práce bylo v ovladači přidáno ovládání pomocí PLC. Z hardwarové stránky je zde pro toto ovládání všechno dokončeno, ale ze stránky softwarové je zde pouze základní funkcionality. To znamená, že lokomotivy mají pevně dané adresy a pouze jednu rychlost. Bylo by proto velmi vhodné, aby v rámci jiné práce bylo toto ovládání doplněno a řádně otestováno.

Literatura

- [1] Š. MIKULICS. *Ovladač sběrnice XpressNet*. ČVUT v Praze , FEL, 2005.
- [2] Administrátor stránek Zavavov.cz. *DCC loko dekodér* [online]. 2015. Dostupné z: <<http://www.zavavov.cz/cz/modelova-zeleznice/elektronika/93-dcc-loko-dekodeder/>>.
- [3] Bohuslav Partyk. *Základní schéma digitálního řízení modelové železnice* [online]. 2014. Dostupné z: <<http://www.masinky.info/2014/01/zakladni-schema-dcc/>>.
- [4] Conrad. *Enkodér do DPS Zippy ANQ-I1B-01B-S-Z* [online]. 2018. Dostupné z: <<https://www.conrad.cz/enkoder-do-dps-zippy-anq-i1b-01b-s-z-360-10-v-0-1-a-do-dps-1x-vyp-zap.k1094451>>.
- [5] Digi-cz. *Centrála NanoX* [online]. 2017. Dostupné z: <<https://www.digi-cz.cz/obchod/centrala-nanox/>>.
- [6] Digi-cz. *Centrála NanoX* [online]. 2017. Dostupné z: <<http://www.digi-cz.cz/obchod/zdroj-pro-centralu-nanox/>>.
- [7] Jiří Král. *CV List* [online]. 2018. Dostupné z: <[https://dccdoma.cz/_files/200001120-19aee1ba2b/CV%20list%20verze%201-2%2C3%20\(CZ\).pdf](https://dccdoma.cz/_files/200001120-19aee1ba2b/CV%20list%20verze%201-2%2C3%20(CZ).pdf)>.
- [8] Jiří Král. *Návod pro lokomotivní dekodéry LaisDcc* [online]. 2018. Dostupné z: <https://dccdoma.cz/_files/200001119-9a08e9b080/LaisDcc_Decoders_Manual_CZ_%20860014%2C21%2C18.pdf>.
- [9] Jindra Fučík. *NanoX-S88* [online]. 2018. Dostupné z: <http://www.fucik.name/masinky/nanox_opt/>.
- [10] KUŘÍK, O. *Studie řízení technologického systému, diplomová práce*. ČVUT v Praze , FEL, Praha, 2011.
- [11] Lenz electronic GMBH. *XpressNet Specification* [online]. Dostupné z: <<http://www.lenzusa.com/1newsite1/Manuals/xpressnet.pdf>>.
- [12] LITE-ON Optoelectronics. *Photocoupler Product Data Sheet LTV-816/827/847* [online]. 2017. Dostupné z: <<http://optoelectronics.liteon.com/upload/download/ds-70-96-0016/ltv-8x7%20series%20201610%20.pdf>>.

- [13] PERNER, J. *Řízení a vizualizace technologického systému, diplomová práce*. ČVUT v Praze , FEL, Praha, 2005.
- [14] Příspěvatelé DDCWiki.com. *Digital Command Control Advantages Over Direct Current* [online]. 2018. Dostupné z: <https://dccwiki.com/Digital_Command_Control_Advantages_Over_Direct_Current>.
- [15] Příspěvatelé Wikipedia.com. *RS-485* [online]. 2018. Dostupné z: <<https://en.wikipedia.org/wiki/RS-485>>.
- [16] STMicroelectronics N.V. *Discovery kit with STM32F429ZI MCU* [online]. 2015.
- [17] Texas Instrument. *LM2574x SIMPLE SWITCHER 0.5-A Step-Down Voltage Regulator* [online]. 2016. Dostupné z: <<http://www.ti.com/lit/ds/symlink/lm2574.pdf>>.
- [18] Texas Instrument. *SNx5176B Differential Bus Transceiver* [online]. 2015. Dostupné z: <<http://www.ti.com/lit/ds/symlink/sn65176b.pdf>>.

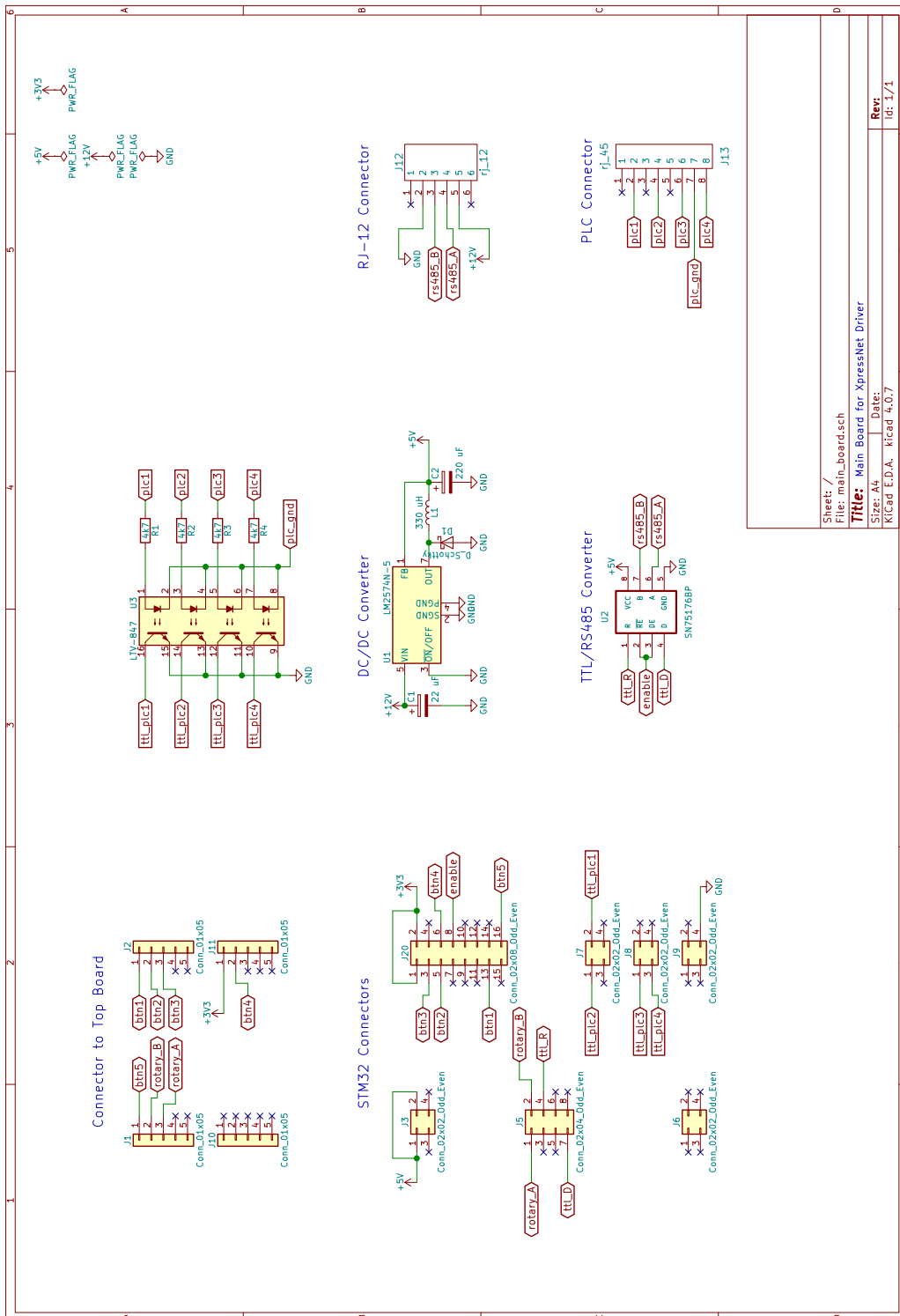
Příloha A

Hlavní deska

A.1 Schéma zapojení

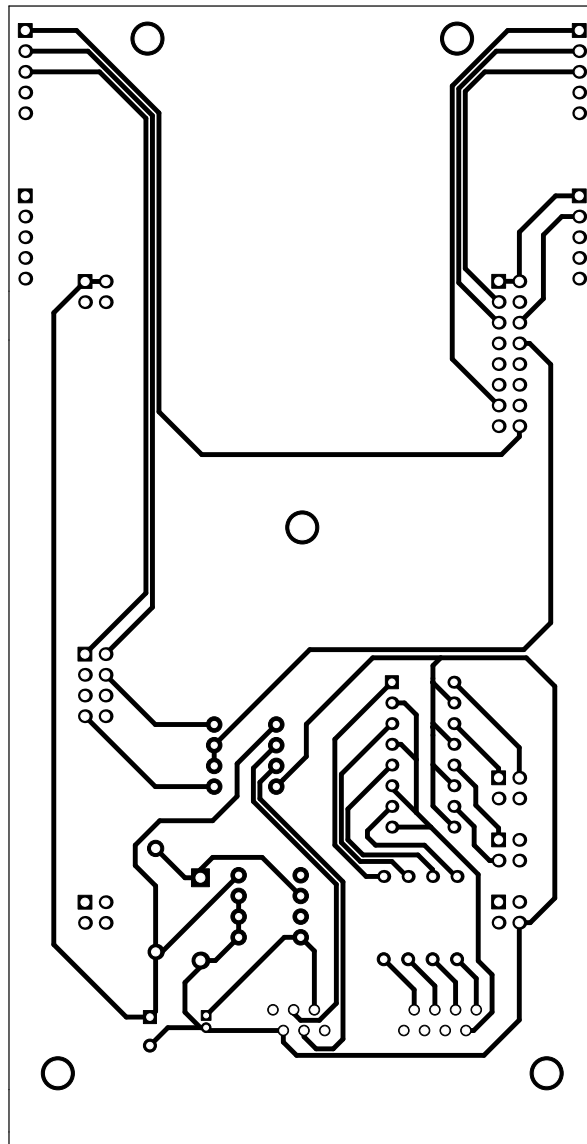
A.1.1 Seznam součástek

Značka	Typ/Hodnota
D1	Shottkyho dioda
C1	22 uF
C2	220 uF
R1	4k7
R2	4k7
R3	4k7
R4	4k7
U1	LM2574N
U2	SN75176BP
U3	LTV - 847
J1 - J11	Dutinková lišta - rozteč 2,54 mm
J12	RJ-12
J13	RJ-45
J20	Dutinková lišta - rozteč 2,54 mm

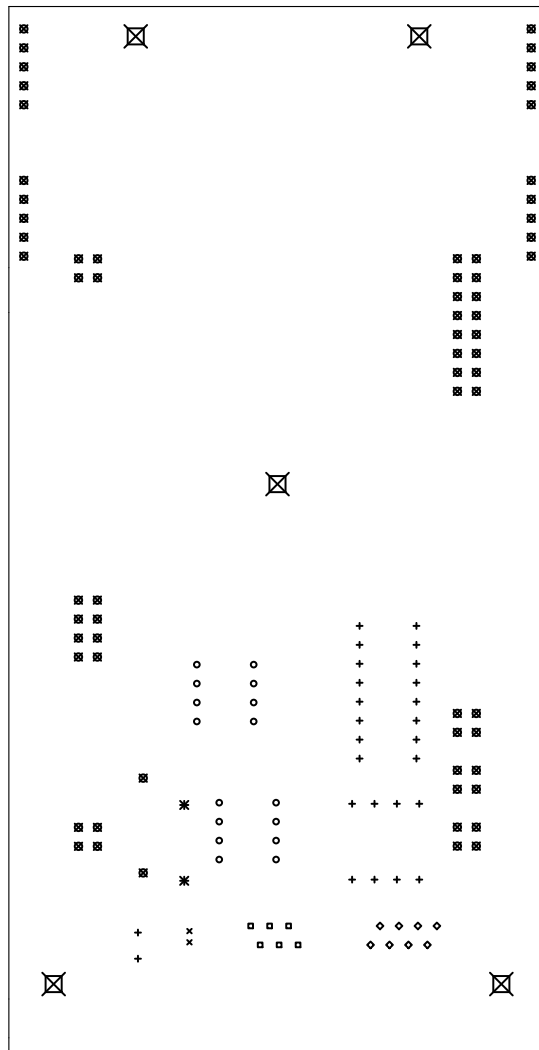


A.2 Výrobní data

A.2.1 Zadní strana - měď



A.2.2 Vrtý



Drill Map:

×	0.60mm	/	0.024"	(2 holes)
○	0.76mm	/	0.030"	(16 holes)
+	0.80mm	/	0.031"	(26 holes)
▪	0.89mm	/	0.035"	(6 holes)
◊	0.90mm	/	0.035"	(8 holes)
■	1.00mm	/	0.039"	(66 holes)
*	1.10mm	/	0.043"	(2 holes)
⊠	3.05mm	/	0.120"	(5 holes)

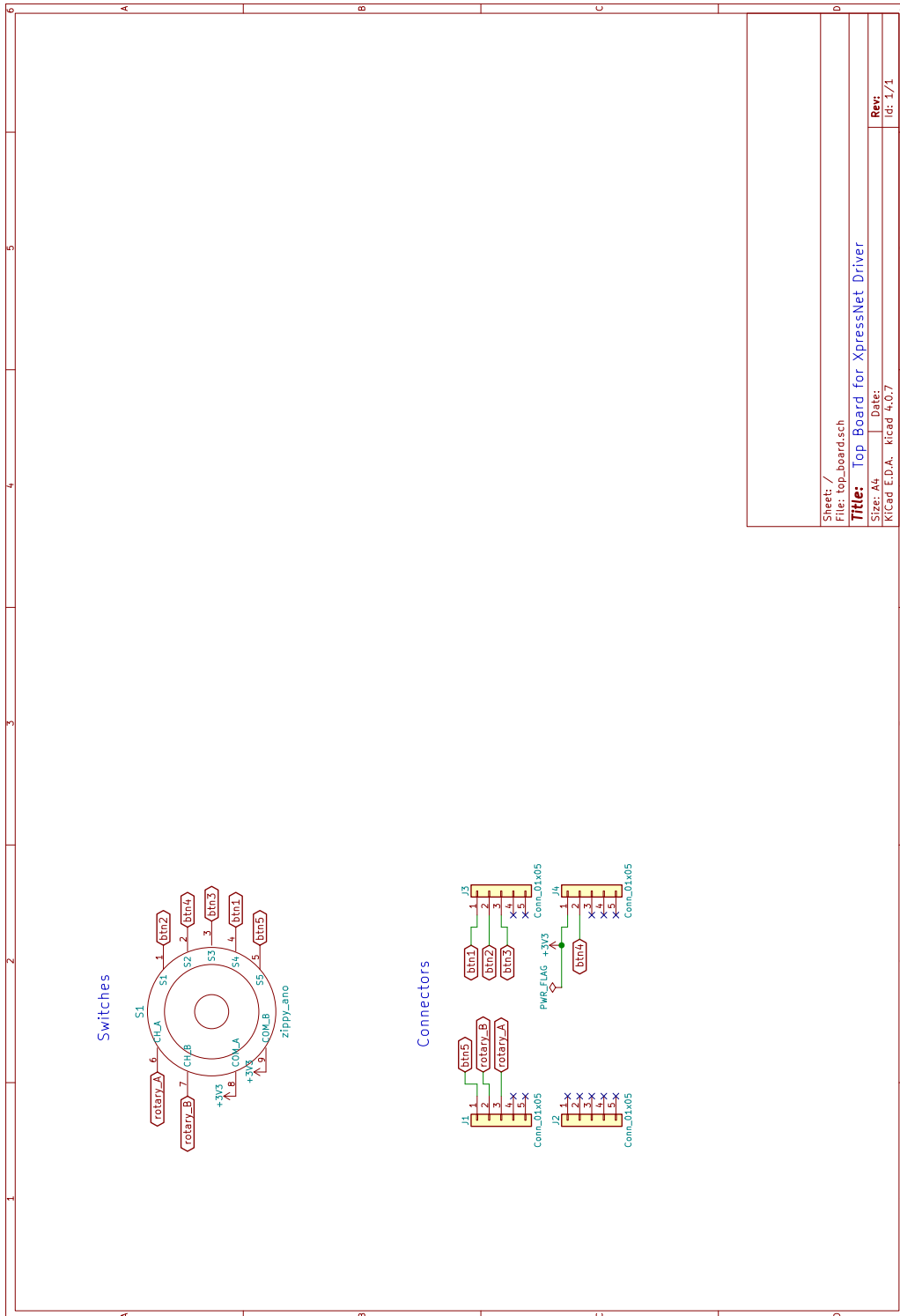
Příloha B

Horní deska

B.1 Schéma zapojení

B.1.1 Seznam součástí

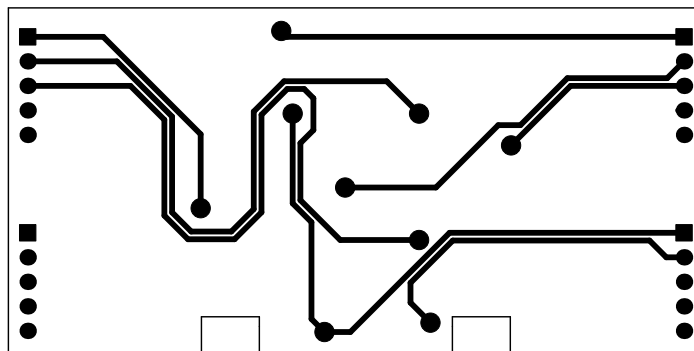
Značka	Typ/Hodnota
S1	ZIPPY ANQ-I1B-O1B-S-Z
J1	Dutinková lišta - rozteč 2,54 mm
J2	Dutinková lišta - rozteč 2,54 mm
J3	Dutinková lišta - rozteč 2,54 mm
J4	Dutinková lišta - rozteč 2,54 mm



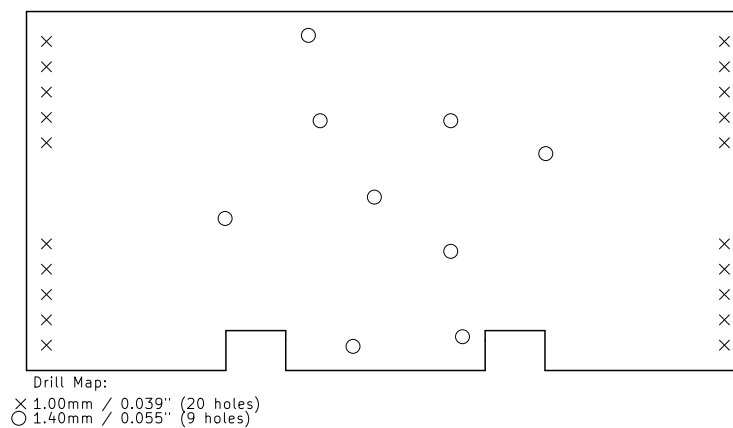
Sheet: /	
File: top_board.sch	
Title: Top Board for XpressNet Driver	
Size: A4	Date:
KiCad E.D.A.	KiCad 4.0.7
Rev:	Id: 1/1

B.2 Výrobní data

B.2.1 Zadní strana - měď



B.2.2 Vrtvy



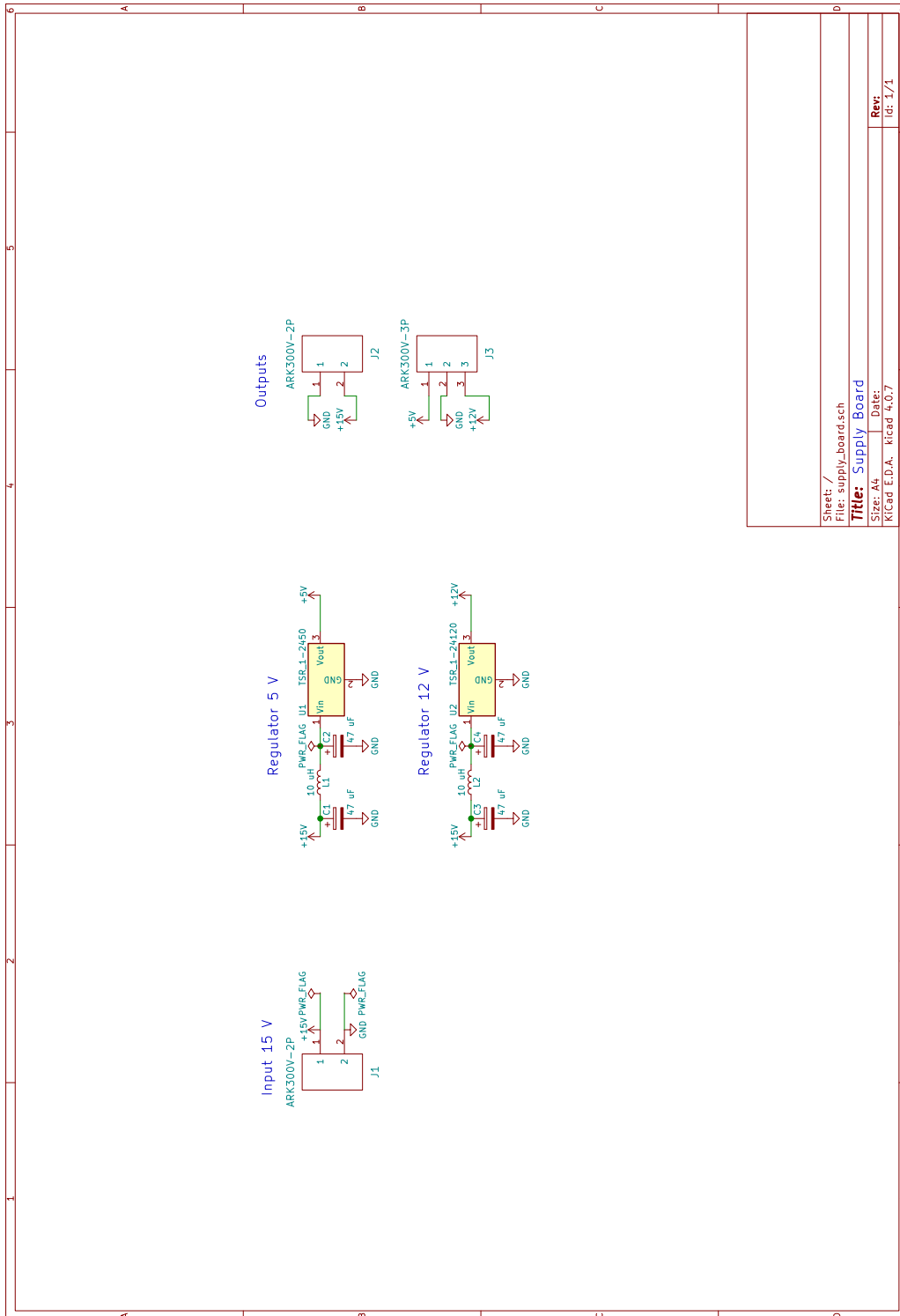
Příloha C

Deska pro zdroj

C.1 Schéma zapojení

C.1.1 Seznam součástek

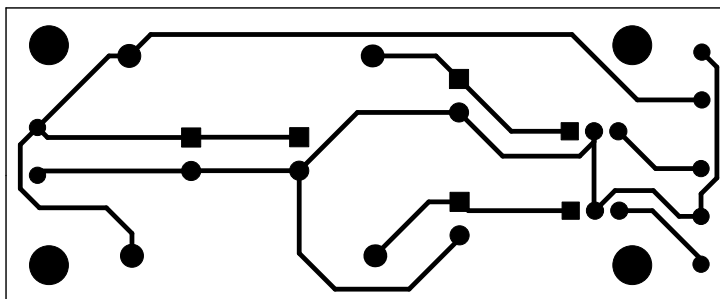
Značka	Typ/Hodnota
C1	47 uF
C2	47 uF
C3	47 uF
C4	47 uF
L1	10 uH
L2	10 uH
U1	TSR 1-2450
U2	TSR 1-24120
J1	ARK300V-2P
J2	ARK300V-2P
J3	ARK300V-3P



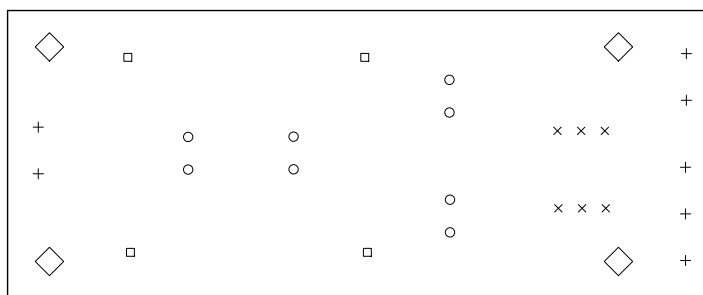
Sheet: /	
File: supply_board.sch	
Title: Supply Board	
Size: A4	Date:
KiCad E.D.A.	KiCad 4.0.7
Rev: 1/1	Id: 1/1

C.2 Výrobní data

C.2.1 Zadní strana - měď



C.2.2 Vrtvy



Drill Map:
× 0.80mm / 0.031" (6 holes)
○ 1.00mm / 0.039" (8 holes)
+ 1.10mm / 0.043" (7 holes)
□ 1.20mm / 0.047" (4 holes)
◇ 3.05mm / 0.120" (4 holes)

Příloha D

Obsah příloženého CD

D.1 „Dokumentace“

Tato složka obsahuje veškeré dokumentace - XpressNet, programovatelný kit STM atd.

D.2 „Plošné spoje“

Zde se nachází KiCad projekt pro horní desku, hlavní desku a desku pro zdroj k příkazové stanici.

D.3 „Pouzdra“

Zde je uložen projekt pouzdra pro program Autodesk Fusion 360 a soubor .stl, který slouží pro tisk pouzdra.

D.4 „Program“

Tato složka obsahuje program ovladače ve formě projektu pro Keil μ Vision. Také je zde dokumentace DoxyGen.

D.5 „Text“

V této složce se nachází kompletní text diplomové práce v projektu pro LaTeX.